Sensitive Data, Shared Cloud

Planning a P3–P4 Secure Research Environment for HIPAA and FERPA Data on Google Cloud



Chuck Forsyth

UCR Research Computing, University of California, Riverside

Version 1.2, November 2026

Public edition

A plan and the evidence behind it. Nothing described here has been built.

Cite: C. E. Forsyth III. 2026. *Sensitive Data, Shared Cloud: Planning a P3–P4 Secure Research Environment for HIPAA and FERPA Data on Google Cloud.* Zenodo. doi:10.5281/zenodo.23890331

Companion paper: doi:10.5281/zenodo.23890329

About this report. We in UCR Research Computing have written a plan for a shared Google Cloud environment for research data that the University of California classifies at Protection Level 3 or 4, including HIPAA protected health information and FERPA student education records. The environment has not been built. It exists as a written plan, infrastructure code that has been validated and planned offline but never applied, image build scripts, runbooks and templates. In this report I check that plan, claim by claim, against what peer universities publish, against the regulations and UC policy, against Google's documentation, and against selected research literature.

Scope. Controlled Unclassified Information and other data that legally require NIST SP 800-171 are out of scope. UCR runs a separate CUI enclave for that data.

This edition. This is the public edition. I have generalized identifiers, live-system observations and some operational detail from the internal edition. Project, group, bucket and service-account names in code excerpts are stand-ins (written like `<project>`). Campus systems, offices and roles appear by function: "the campus research cluster", "the campus security office", "the systems team", "the accountable data owner", "a data steward". Weaknesses in the design are described by kind, with an example a reader can picture, and not by configuration line. The plan and its working documents are not public, so where I rely on them I describe the design in my own words and cite nothing. I wrote it for people at other universities building something similar. The views and readings here are my own and do not represent the University of California, Riverside or the University of California. The campus security office, systems team and leadership have not reviewed or approved this report, and nothing in it is UC policy or a commitment by UCR.

How I used AI. The plan, the design decisions and the conclusions are mine and my team's. I used AI writing agents (Claude Sonnet 5.5) to draft chapters from my brief and Claude Opus 5.5 to assemble and edit them, and AI reviewer panels to check the drafts against primary sources. I checked the primary sources behind every correction and am responsible for every claim. The evidence base also includes eleven AI-assisted research reports, which I treated as leads and checked against primary sources; claims from outside UCR carry public citations, or are labeled as reported where I could not confirm them. Appendix A gives the details.

Contents

Executive Summary **1**

1 Why UCR Needs a P3–P4 Environment **5**
1.1 UC protection levels and the research data that lands in them . . . 5
1.2 The gap: no campus place for P3-P4 compute . . . 6
1.3 Why the shared open cluster cannot host P3-P4 data . . . 6
1.4 Scope and routing . . . 8
1.5 The decisions D1 to D14 in plain words . . . 9
1.6 How the plan reached version 3 . . . 11
1.7 The funding question . . . 11
1.8 Design considerations . . . 12

2 How Peer Universities Run Secure Research Environments **14**
2.1 Method and what counts as verified . . . 14
2.2 Two hosting patterns, and how many peers use each . . . 15
2.3 How researchers get in, and what the gate checks . . . 17
2.4 Who operates the environment, and who pays . . . 17
2.5 Data classes, and where the CUI line falls . . . 20
2.6 UK and European practice as a lens . . . 20
2.7 Related frameworks, and what they leave out . . . 22
2.8 Where the AI-assisted research reports and the institutions' pages disagree . . . 23
2.9 How the plan compares . . . 23
2.10 Design considerations . . . 24

3 The Rules: HIPAA, FERPA and UC IS-3 **25**
3.1 HIPAA . . . 25
3.2 FERPA . . . 28
3.3 UC IS-3, P3 and P4 . . . 30
3.4 Data use agreements and what providers add . . . 32
3.5 Out of scope: CUI and NIST SP 800-171 . . . 32
3.6 HIPAA, FERPA and IS-3 P4 compared . . . 33
3.7 Shared responsibility . . . 33
3.8 Design considerations . . . 36

4 Google Cloud Building Blocks **37**

4.1 The blocks in one view . . . 37
4.2 Projects and organization policies . . . 37
4.3 A network with no internet . . . 41
4.4 VPC Service Controls . . . 42
4.5 Identity-Aware Proxy, OS Login and the VM itself . . . 44
4.6 Artifact Registry remote repositories . . . 45
4.7 Cloud Storage . . . 45
4.8 Cloud Logging . . . 45
4.9 BAA coverage . . . 46
4.10 Frameworks around the blocks . . . 47
4.11 Documented gotchas, collected . . . 48
4.12 Design considerations . . . 48

5 Architecture of the Planned Environment 50
5.1 Two projects, one boundary . . . 50
5.2 The network: no way out except Google's APIs . . . 52
5.3 Identity and access . . . 53
5.4 The per-study module . . . 54
5.5 Images and the boot sequence . . . 55
5.6 Reaching RStudio and the desktop . . . 57
5.7 Software and logging in the architecture . . . 57
5.8 What the offline plans contain . . . 58
5.9 How the design changed from v1 to v3 . . . 58
5.10 The fifteen pilot tests . . . 59
5.11 Where the architecture stands . . . 60
5.12 Design considerations . . . 60

6 Tenancy and Isolation: One Shared Project or One per Study 62
6.1 What a Google Cloud project does and does not isolate . . . 62
6.2 How the plan separates studies inside one project . . . 62
6.3 The case for a project per study . . . 65
6.4 Three options side by side . . . 66
6.5 Perimeters and the limits of VPC Service Controls . . . 70
6.6 The customer-managed key debate . . . 70
6.7 What data providers actually say . . . 72
6.8 Identity as the commingling path . . . 72
6.9 Shared file systems and a future cluster . . . 73
6.10 Conditions that would move a study into its own project . . . 73
6.11 Design considerations . . . 75

7 Data In and Data Out **76**
7.1 The planned path in and out . . . 76
7.2 The perimeter and the tunnel . . . 78
7.3 Copy-out from inside the VM . . . 78
7.4 Checking outputs . . . 81
7.5 Trained models: the export gap . . . 83
7.6 End of study: soft delete and destruction . . . 84
7.7 Residual-risk acceptance . . . 85
7.8 Design considerations . . . 85

8 Software Without Internet and Supply-Chain Risk **87**
8.1 What the plan specifies today . . . 87
8.2 Why the live mirror is the weakest part of the design . . . 88
8.3 Two incidents that can be verified . . . 89
8.4 What scanners can and cannot do . . . 90
8.5 Four policies compared . . . 91
8.6 Tools for each ecosystem . . . 93
8.7 Weak points in the plan's own supply chain . . . 94
8.8 A recommended policy for UCR . . . 95
8.9 Request process and turnaround . . . 97
8.10 Design considerations . . . 97

9 The Researcher's Experience **99**
9.1 What the published evidence says . . . 99
9.2 The planned first day . . . 100
9.3 The connection path and its timers . . . 103
9.4 Working conditions inside the VM . . . 105
9.5 Friction points and what the plan does about them . . . 106
9.6 Measuring the experience in the pilot . . . 106
9.7 Design considerations . . . 109

10 Logging, Monitoring and Incident Response **111**
10.1 What the plan would record . . . 111
10.2 The log project, the sink and retention . . . 113
10.3 Alerts . . . 116
10.4 Review, SIEM and detection services . . . 118
10.5 Incident response and breach clocks . . . 119
10.6 Proposed tabletop exercises . . . 120
10.7 Design considerations . . . 123

11 Operating Model, People and Process **124**
11.1 Roles and the accountability matrix . . . 124
11.2 The life of a study . . . 125
11.3 The runbooks: triggers, owners and evidence . . . 129
11.4 Teardown with evidence . . . 131
11.5 Student education records at intake . . . 132
11.6 Staffing: what the evidence supports . . . 134
11.7 Design considerations . . . 136

12 Cost and Funding **138**
12.1 Method and price sources . . . 138
12.2 Cost of the three VM templates . . . 139
12.3 Platform costs that belong to no study . . . 141
12.4 The funding question . . . 141
12.5 Design considerations . . . 145

13 GPUs, HPC and AI on Sensitive Data **147**
13.1 The GPU template as planned . . . 147
13.2 Open-weight language models inside the study VM . . . 148
13.3 Managed AI under the BAA . . . 151
13.4 What peers and the University say . . . 152
13.5 Confidential GPUs . . . 153
13.6 Phase 2: a secure Slurm cluster . . . 155
13.7 Design considerations . . . 156

14 The Plan Compared: Lessons, Risks and Roadmap **158**
14.1 The plan against peers and the literature on twelve dimensions . . . 158
14.2 What the two review panels changed . . . 158
14.3 A checklist for inherited defaults . . . 161
14.4 Corrections to sources and to the plan's documents . . . 162
14.5 Candidate changes for version 4 . . . 164
14.6 The risk register . . . 166
14.7 Roadmap and decision gates . . . 169
14.8 Decisions for the research computing director and leadership . . . 170
14.9 Open questions . . . 171
14.10 Design considerations . . . 172
14.11 Conclusion . . . 172

References **174**

A Methods and Sources **192**

A.1 The subject: a plan, frozen . . . 192
A.2 Evidence gathering . . . 192
A.3 Writing and editing, and how I used AI . . . 193
A.4 What the checking found . . . 193
A.5 Limits . . . 193
A.6 Where the material is kept . . . 193

List of Figures

1.1 Rows in the research computing team's working list of IS-3 controls for a P3-P4 compute environment, by heading and by the level at which the row applies (counted from the list; the list is a working analysis, not a campus security office determination). Rows marked "all levels" apply at P4 as well and include the rows the list assigns to P2 and above; the HIPAA-specific rows are in the same group. 7

1.2 Intended routing of research data by type, protection level and destination. The dashed box marks the environment I examine here; it exists only as a plan. The routing follows the plan's scope decision and the routing rule for medical school data. 9

2.1 Hosting of the 19 offerings in Table 2.1, as stated on each institution's own page (read 2026-11-01). Rows whose pages do not name a platform are counted separately, not assigned to either side. 15

3.1 FERPA routes the intake would record for a study using student education records. Drawn from the text of 34 CFR 99.30, 99.31 and 99.35; the determination itself belongs to the campus FERPA officer. 30

3.2 Controls in the working extract of IS-3 for a P3-P4 compute environment, by group and by the level at which each applies. Counts are of the 71 table rows in the extract, not of the 63 its text states. The extract is a working analysis by the research computing team, not a determination by the campus security office. 31

4.1 Google Cloud building blocks as the foundation and secure-VM Terraform declare them. Counts come from the offline foundation plan (100 resources). The perimeter is dashed because the offline plan creates none. The dashed arrow is the project-level log sink from the secure project to the log project's bucket. 38

5.1 Overall architecture of the planned environment, drawn from the foundation and secure-VM Terraform. Solid arrows are access paths (administrators act on the secure project; the deployer, which administrators may act as, also has write access to the state storage in the log project); the dashed arrow is the cross-project log sink. Three study VMs are shown as in the pilot example. 51

5.2 Excerpt from the offline foundation plan (with computed-at-apply attributes omitted). The project identifier is a stand-in; the plan was produced with Terraform 1.12.2 and Google provider 8.9.0 and was never applied. 53

5.3 Planned resources by Terraform type, from the offline plans (foundation: 100 resources; three-study pilot: 59). Both plans used placeholder values and were never applied; every action in both is a create. 54

5.4 Boot sequence with the fail-closed login gate as the design intends it. As written, the status can survive a reboot, so the login service can start before the first box runs and honor the previous boot's `OK`; a boot-scoped status removes that window. 56

6.1 Three tenancy layouts for the planned environment. Dashed lines are VPC Service Controls perimeters: one around the shared project in A, optionally one per study in B, and one covering the environment folder in C, which new study projects join automatically when created in the folder. Panel A is drawn from the planned foundation and per-study module. Panels B and C are proposals, not part of the plan. 69

7.1 Planned data path for one study, drawn from the per-study module, the release runbook and the audit rules. Steps 1 to 3 are Cloud Storage calls (a write, a read and a write) and are visible in Data Access logs; only step 4 raises an alert. Step 5 and the dashed routes are outside every technical log: the guest audit rules cover some ways of copying files but not all, and Chapter 10 found no alert on them. 78

8.1 The planned software path (solid boxes) and the additions proposed below (dashed fill). In the plan, the PyPI, Docker Hub and Yum remotes reach the internet directly through the registry once upstream access is allowed; the proposal moves Python and OS packages to curated standard repositories filled by the builder and keeps the remotes as a fallback until the pilot decides. 95

9.1 Planned steps before a researcher's first login (swimlane, planned process only; no step has been run and the plan sets no duration for any of them except removal of leavers). The study group is created at step 4 from the signed DSP; people added later follow steps 5 and 6. . . 101

9.2 Planned first day on a study VM (swimlane, planned process only). The dashed arrow marks that the package request and the export request are independent; the container rebuild is shown only to place it. The plan sets no turnaround for steps 5 and 7; the container target is written as "TBD, e.g. 2 business days." 101

9.3 Proposed tunnel command for the wrapper: the plan's tunnel command plus client keepalives. Placeholders are in capitals. `-N` opens no shell, which is also why shell timeouts do not apply to it. 104

9.4 Mock-up: a proposed `p4 connect` session. The wrapper does not exist, and the messages, option names and the retry advice are illustrative. Whether the researcher could be told why a login was refused depends on the gate's message, which the plan has not specified. 105

10.1 Planned log flow, drawn from the planned configuration. Solid arrows exist in the offline plan; dashed boxes and arrows are optional or undecided (the SIEM topic is created only when a SIEM is enabled, and the plan names no SIEM). The offline plan contains one sink to the log bucket and no perimeter, so the egress question for the sink is untested. Of the nine alert policies, eight are in the secure project and one is in the log project. 113

11.1 Lifecycle of a study under the plan, with the role that acts at each stage and the record each stage leaves. Drawn from the plan's lifecycle list, the RACI and the runbooks. The build stage is a one-time platform stage. 127

12.1 Left: monthly list-price cost of one VM per template, at continuous operation and at 40% uptime, split by component (Iowa, on demand, prices read 2026-11-01; CIS fee at the low value). Right: monthly cost of a subsidy that covers VMs, disks, snapshots and buckets for 5, 15 or 30 studies, for one CPU VM per study, and for an assumed mixed fleet, meaning a guessed mix of templates rather than a forecast: 4 CPU and 1 GPU VM for 5 studies; 11 CPU, 3 GPU and 1 high-memory for 15; 21 CPU, 6 GPU and 3 high-memory for 30. Each is shown at continuous operation and at 40% uptime. No demand data exist. Platform fixed costs are excluded. Source: `figs/ch12_cost_calc.py`. 140

13.1 Weight memory of four Qwen3 models at three precisions against the 24 GB of an L4 (dashed) and an assumed 21.6 GB budget (dotted). Data from Table 13.1; script `figs/ch13_memory.py`. 149

13.2 A proposed path for open-weight model files into an offline study VM. Nothing here is in the plan: it combines the plan's builder-outside pattern for images with the read-only bucket roles in the `secure-vm` module. Shaded boxes are review steps by staff. The VM reaches the bucket and nothing else; the researcher reaches the server only through the same tunnel used for RStudio. 150

14.1 Proposed roadmap with six decision gates. Phases follow section 7 of the plan; the gates split its final step. Gate order is a proposal: each gate depends on the results of the one before. 170

List of Tables

1.1 Why an open shared cluster does not fit a P3-P4 boundary, generalized from the research computing team's gap analysis. . . . 8

1.2 Decisions D1 to D14 of plan v3, restated in plain words. Status for all rows: decided in the plan; none implemented. . . . 10

2.1 Peer offerings for high-risk research data: platform, data classes and operator, as stated on the institution's own page (read 2026-11-01). "ns" means the page does not state the item. Sources are cited in the first column. . . . 16

2.2 Access model, cost to the researcher and onboarding gate for the offerings in Table 2.1. Prices are as published on the pages cited there, in US dollars, and are not adjusted. "Not found" means I did not find the item on the pages I read, not that none exists. Verified: Y means the institution's own page was read for this row; Y* means only a meeting-minutes document was read. . . . 18

2.3 Reference environments and standards from the UK, Europe, Australia and one US university, as recorded in the cited abstracts or documentation pages. The records say nothing about operating experience. . . . 22

2.4 Claims in the AI-assisted research reports compared with the institutions' current pages (read 2026-11-01). . . . 23

3.1 HIPAA, FERPA and IS-3 P4 compared. Sources: eCFR text of 45 CFR 160 and 164 and 34 CFR 99 (current through 2026-10-29); UCOP IS-3 pages; working extract of IS-3 controls. 33

3.2 Proposed shared-responsibility matrix for the planned environment. The plan's RACI is draft v3 and unsigned; "open" marks rows where the plan itself has no agreed owner. . . . 35

4.1 Platform and network blocks: function, planned use, BAA listing and main gotcha. . . . 39

4.2 Access, software, storage and governance blocks: function, planned use, BAA listing and main gotcha. . . . 40

4.3 Documented limits that bear on the plan, the source, and what the pilot should check. . . . 48

5.1 The three VM templates in plan section 9 and the module defaults. Disk sizes are defaults; the pilot example overrides the highmem data disk to 2000 GB. . . . 55

5.2 Planned resources by function, from the offline foundation and pilot plans. . . . 58

5.3 Principal design changes by plan version and the review finding that drove each. . . . 59

6.1 Where Google Cloud applies the controls that bear on study separation. All rows from Google documentation read on 2026-11-01. . . . 63

6.2 Layers that separate Study A from Study B in the planned shared project, from the per-study module and foundation configuration, and who holds the power to cross each layer. From the offline plan. . . . 64

6.3 Claims from three AI-generated research reports and the working gap analysis, checked against the sources named. The reports are model-generated and treated as leads. 67
6.4 The three tenancy options compared. The plan is option A. Entries on data-provider acceptability reflect the agreements that could be read (Section 6.7); none of them names a cloud project. 68

7.1 Copy-out routes from a study VM, the control the plan's files show, and the residual risk. "Logged" means a log source exists, not that anyone is alerted. Routes 4 and 5 depend on pilot tests 9 and 10, which have not run. 79
7.2 Output-checking rules, with the plan's release checklist beside them. ONS and CMS values are quoted from their documents; the plan's default of 11 is its own choice and is not taken from either. "None" means the checklist has no corresponding item. 82

8.1 Incidents used here: what the primary records establish and what remains unverified. . . . 89
8.2 Four admission policies for software entering a P3-P4 enclave. Ratings are qualitative and apply to the plan's context. 92

9.1 Timers that can end or interrupt a researcher's session. Rows 1 and 2 are Google documentation; rows 3 to 6 are in the plan's image and startup files; row 7 is a Google admin setting whose UCR value is unknown. 104
9.2 Friction points, planned mitigation and residual friction. Residual items are inferences from the plan's files unless a source is cited. 107
9.3 Proposed pilot metrics. No target values exist in the plan; the targets column is deliberately empty until the pilot has produced baselines. All are proposed. 108

10.1 List-price cost of holding 1 TiB of log data for 72 months, us-central1 where regional. Prices: Cloud Logging $0.50 per GiB ingested (first 50 GiB per project per month free; 30 days of storage included) and $0.01 per GiB per month beyond 30 days [1]; Cloud Storage regional Standard $0.0200 and Archive $0.0012 per GiB per month, from the hourly rates on the pricing page [2]. Free allotments, retrieval fees and the Archive 365-day minimum duration are ignored. 115
10.2 The nine planned alert policies, in the order of the offline foundation plan. "Notifies": Security is the campus security office's queue, Systems the systems team's. "Runbook" names the plan's runbook that applies; none of the log-based policies has a response step beyond the text in its documentation field. 116
10.3 Events with no planned alert. "Evidence" states what I verified; the proposals are mine and untested. 117
10.4 SIEM options for the planned environment. "Verified" means read in Google's documentation on 2026-11-01. 118
10.5 Notification clocks that could apply to a study in the planned environment. "Verified" means read in the primary text on 2026-11-01. Which row applies depends on data source, role and counsel's reading. 121
10.6 Proposed tabletop scenarios. "Tests" names the plan element or evidence gap each scenario exercises. 122

11.1 Condensed RACI from the plan (draft v3, to be signed). Role labels: Owner, the accountable data owner for the environment; Security, the campus security office; Officer, the security officer; Systems, the systems team; Computing, the research computing team; Steward, the study's data steward. Exactly one party is accountable in each row; "[?]" marks an assignment the plan has not settled. Rows were merged only where the accountable party was the same. . . . 126

11.2 How DSP content becomes module settings in the plan. The right column names variables in the planned `secure-vm` module and its labels. . . . 128

11.3 The twelve runbooks in the plan: trigger, owner and the evidence the plan says each run leaves. "No template" means the plan names the record but supplies no form. Security is the campus security office; Systems is the systems team. All runbooks are drafts. . . . 130

11.4 Proposed FERPA additions to the intake checklist, with where each answer would be recorded and which platform setting it drives. The determinations belong to the campus FERPA officer and the security office; the research computing team only records them. . . . 133

11.5 Arithmetic on assumed inputs, not a forecast: annual hours of export review by number of active studies. Low case: 14.2 releases per study per year at 30 minutes (review-derived, unverified). High case: 104 releases per study per year (two per week) at 60 minutes (assumed). FTE at 2,080 hours. . . . 135

12.1 Monthly list-price cost of one study VM by template, USD, Iowa (us-central1), on demand, 730 hours, prices read 2026-11-01. CIS fee at the low value ($0.0223 per hour; none on the GPU template). Snapshots and buckets use the assumptions in Section 12.1. Components are rounded to the nearest dollar; totals are computed from the unrounded rates and then rounded, so a row can differ by $1 from the sum of its displayed parts (high-memory: 1,530.24 + 16.28 + 105.00 + 54.12 = 1,705.65, shown as 1,706). The last column is SCC Premium at the project-level rate of $0.0071 per vCPU-hour, not included in the totals [3]. Script: `figs/ch12_cost_calc.py`. . . . 139

12.2 Platform costs outside any single study, USD per month at list price, prices read 2026-11-01. Quantities in italics are my assumptions. "Owner" records who the plan or the funding rule would have to name; none is named today. . . . 142

12.3 Monthly cost of a subsidy covering VMs, disks, snapshots and buckets (USD, list price, CIS fee at the low value, Iowa, prices read 2026-11-01). Platform fixed costs (Table 12.2) and staff time are excluded. Mixes: 4 CPU and 1 GPU (5 studies); 11 CPU, 3 GPU, 1 high-memory (15); 21 CPU, 6 GPU, 3 high-memory (30). Source: `figs/ch12_cost_calc.py`. . . . 143

12.4 Funding options for the shared environment. Dollar figures are from Tables 12.1, 12.2 and 12.3. . . . 145

13.1 Open-weight Qwen3 models against one 24 GB L4. Weights are decimal gigabytes (parameters times bytes per parameter; 4-bit adds 10% overhead). "Room" is the cache that remains after weights inside an assumed 21.6 GB budget, expressed as tokens at the model's bf16 KV size. Arithmetic, not measurement; license and parameter counts from the model repositories read on 2026-11-01. . . . 149

13.2 Published AI-tool policies for sensitive data, from institution pages read on 2026-11-01. Harvard's and Michigan's levels are their own scales. . . . 153

13.3 AI options for P3-P4 data and the conditions each would need. Status describes the evidence I found; none is built. . . . 154

13.4 Isolation options for phase 2 storage and where the plan stands. Costs use $200 per month per instance (Basic HDD 1 TiB; $197 at list, Chapter 12). 156

14.1 The plan compared with verified peer practice and the literature on twelve dimensions. The last column is my reading, not a rating by the campus security office. 159

14.2 Changes driven by the two review panels, with the plan version in which each landed. . . . 160

14.3 How each register row was found, and whether finding it needed a document from outside the repository (a vendor, upstream or regulatory page, a price page or an agreement). No = nothing in the repository implies the defect; Yes = deciding it needs a fact stated only in an outside document. The classification is mine. The two repository-only panels' accepted findings were fixed before v3, so they have no open rows; R01 and R02 echo the first panel's process list, and a few rows (for example R12 and R35) touch points that a single raw panel review raised but its verdict did not carry forward. 161

14.4 Inherited-default checklist: one row per kind of defect, the document that settles it, a generic check, and whether a review of the repository alone could find it. Repo-only?: No = nothing in the repository implies the defect; deciding it needs a fact stated only in an outside document. Type-level; not a description of our configuration. 162

14.5 Corrections found while checking: statements in reports, in the plan's companion documents and in the plan text. Report and gap-analysis items already in Table A.1 are grouped. 163

14.6 Candidate changes for a version 4. P = before the pilot; D = before first P3–P4 data; L = later. S-numbers are from the plan comparison; N-numbers are mine. 165

14.7 Risk register for the planned environment. L = likelihood, I = impact (H, M, L). Type: D = decision or document, C = code or configuration, T = pilot test. 166

A.1 Corrections to literature reviews and the gap analysis 194

Executive Summary

UC Riverside has no shared campus place to compute on non-CUI research data that the University of California classifies at Protection Level 3 or 4, outside the medical school's own systems for its clinical data. That population covers HIPAA protected health information received under data use agreements, FERPA student education records, large sets of personal information, and data an IRB or a contract places at P3 or P4. Our answer in UCR Research Computing is a plan, now at version 3, for a shared environment on Google Cloud. It exists as documents, Terraform that validates and plans offline (100 resources for the shared foundation, 59 for a three-study pilot), image scripts, runbooks and templates. None of it has been built.

I checked that plan, claim by claim, against peer universities' own pages, the regulations and UC policy, Google's documentation and selected research literature. Most of what I found is specific to our plan, and Chapter 14 lists it. Four findings travel to any campus building something similar, so I put them first.

Four lessons for other campuses

1. The defects that matter most are inherited defaults. The problems with the highest stakes were not in our own logic. They were in what we inherited from vendors and open-source projects: which API name a service perimeter has to list, which options only a paid edition of an application honors, the storage soft delete that keeps "deleted" data for a week, and package mirrors that pass through whatever upstream publishes. Each of these is settled by a page outside the repository. Two single-shot AI review panels that saw only the repository found none of them; AI agents given the same code plus the vendor and upstream pages found them (Chapter 14). The comparison is confounded, since the finders had the documents and no human repository-only review was run, but the practical advice holds: give reviewers the vendor pages along with the code.

2. In a shared project, study isolation is decided in the administrator plane. The researcher path in our plan separates studies well: one VM, one group and one set of buckets per study. The exposure sits with the administrators and the deployment identity, which reach every study in the project (Chapter 5). On the documented behavior, per-study customer-managed keys do not change that: from Google's documentation, one Compute Engine service agent per project holds the key role for every study's key. I have not tested this (Chapter 6).

3. A folder-level perimeter makes project-per-study tenancy cheaper to add than it looks. Google adds a project created in a perimeter folder to that perimeter automatically. Adding a study project to such a folder therefore needs no perimeter or policy edit, at roughly twice the shared design's per-study resources (about 40 against about 20, an estimate rather than a Terraform plan) and a larger one-time build. In return each study gets the isolation a project boundary brings (Chapter 6).

4. FERPA's studies exception and faculty-initiated research. The exception in 34 CFR 99.31(a)(6) covers studies done for or on behalf of the educational agency, for three named purposes. On my reading, the studies exception does not cover faculty-initiated research unless the institution commissions or adopts the study;

the campus FERPA officer decides. IRB approval does not fill the gap, so intake has to record which route authorizes each student-records study (Chapter 3).

Where the plan holds up and where it has gaps

I compared the plan with practice on twelve dimensions (Chapter 14). On hosting, identity and the operating model it is consistent with what peer pages describe. On the network, data out and logging it is consistent with Google's guidance, but no peer page says enough to compare against. On tenancy it departs from the one Google Cloud peer, defensibly but without a written rationale. It has gaps on five: data in, software admission, cost consistency, student records, and GPUs and AI. Tenancy (one shared project, or a folder of study projects) remains open.

Peers. Of the 19 peer offerings I read, eleven state their hosting: six are cloud-hosted and five on premises. Only Stanford's Nero is documented as running on Google Cloud; UC San Diego's Sherlock lists Google Cloud among the clouds it supports. Peers that describe onboarding put a human review of the data, agreement or classification before a project is created, as our intake does. Nero gives each requesting team its own project, and no peer page describes one shared project for all studies. Most subsidized peers run on premises and most cloud peers pass costs through; Johns Hopkins SAFE Desktop, cloud-based and free to 100 GB, is the exception (Chapter 2).

Rules. The 2024 rewrite of the HIPAA Security Rule is still a proposal, with final action listed for July 2027. IS-3 already requires most of what it would add at P3 and above, so we should track each proposed item as met or as a recorded gap. Scan cadence, an annual penetration test and an annual compliance audit have no owner or schedule yet. FERPA is in scope but absent from the plan. FERPA has no breach-notification rule of its own; notice duties come from state law and the agreements (Chapter 3).

Google Cloud. Every data-bearing service the plan uses is on Google's HIPAA BAA list. OS Login, VM Manager, Organization Policy and billing budgets are not named there, and the campus security office has to decide how to treat administrative services. Our mirrors for operating-system packages rely on a Preview feature, and Google advises against pre-release offerings with PHI. VPC Service Controls protects Google API calls, not network paths, so the perimeter and the firewall are both needed. Read against Google's pages, our perimeter has three kinds of gap: a product protected by several APIs needs all of them in the perimeter, the network ranges must match the ranges Google publishes, and every administrator rule should carry the campus access level once it exists (Chapter 4).

Isolation. The CMS and ResDAC agreements I read restrict linkage and require need-to-know access, but neither bans co-hosting outright (NIH controlled-access data goes to UCR's CUI enclave in any case). We should still ask each provider at intake and keep written triggers that would move a study into its own project (Chapter 6). The deployer problem behind lesson 2 is set out in Chapter 5.

Login gate. The fail-closed login gate is meant to close at the start of every boot. As written, the order of boot-time steps leaves a window: if a step hangs before the gate is reset, the previous boot's pass state stands and logins stay open. The fix is a status that cannot outlive a boot, such as one kept in memory-backed storage that is empty at every start (Chapter 5).

Data out. The `gui_only` mode relies on session restrictions that only the paid RStudio edition (Posit Workbench) honors; open-source RStudio ignores them. Teardown could report that nothing remains while Cloud Storage's seven-day soft delete still holds copies, and the Department of Education's guidance says deletion alone is not destruction. The export form has no rule for trained models (Chapter 7).

Software. Two 2026 releases of PyTorch Lightning on PyPI were compromised and ran on import; a live mirror would have served them to the next user. I recommend an admission rule (a quarantine window, a curated allowlist for the strictest studies, and R pinned at 4.4.0 or later) over reliance on scanners (Chapter 8).

Researchers and operations. The best primary evidence on researcher experience is about approval delay: a UK evaluation found more than a third of academic respondents waited over six months. The planned tunnel wrapper (not yet written) sets no client keepalives, so a dead tunnel hangs instead of failing fast, and the one-hour IAP idle limit is untested. The idle check misses federated users' processes, so VMs may stop under running work. None of the fifteen pilot tests measures researcher time or satisfaction (Chapter 9). Keeping raw logs for six years is our choice; HIPAA names six years for documentation, not logs. One planned alert watches the wrong log, so it would stay silent, although the records still reach the audit log (Chapter 10). The RACI gives every activity one accountable party, but five rows depend on an accountable data owner the plan has not named and seven on the systems team accepting roles it has not yet accepted. I found no verified staffing ratio, so the pilot should measure effort (Chapter 11).

Cost. Recomputed at list price, the CPU template matches the plan ($355 a month running continuously), the GPU template is about 4% above the plan's $910 ($947), and the high-memory template is about a fifth too low ($1,652 against $1,290). Idle stop at 40% uptime cuts the CPU template from $383 to $203 a month (47%), less than half, because disks, snapshots and buckets still bill while the VM is stopped. The shared platform's cloud bill is small (roughly $6 to $102 a month before Security Command Center and any shared file system), so the funding question is about staff time and risk. The plan's "possibly subsidized" decision (under the campus no-charge tier) conflicts with a project-pays model in which P3–P4 projects pay their own costs, and the Terraform's $2,000 monthly budget is a placeholder marked TBD (Chapter 12).

GPUs and AI. An 8B open-weight model such as Qwen3-8B runs at full precision on the plan's single L4 GPU; a 32B model fits only at 4-bit with a short context. The L4 cannot run as a Confidential VM, no GPU quota has been requested for the secure project, and managed AI under the BAA needs campus security office approval before use (Chapter 13).

Fixes and decisions before first data

The risk register in Chapter 14 has thirty-five items. Fourteen close with a decision, a signature or a document edit. Eighteen need code or configuration, and three need pilot tests. Eight are rated high on both likelihood and impact. Eight rows are v3 defects the claims panel found while checking this book, among them the boot-check ordering, the reach of the deployment identity and the coverage of inactivity detection; Chapter 14 gives their fixes as version 4 candidates.

Six decisions only leadership can make

Testing will not settle these; the research computing director and campus leadership have to.

1. Funding: the plan's "possibly subsidized" position against the rule that projects pay through a chart

of accounts (COA) and a memorandum of understanding (MOU), and who pays for shared staff, the penetration test and incident response.

2. Tenancy: keep the shared project (D3) with written triggers that would move a study out, or adopt a folder of per-study projects under one perimeter (Option C).
3. Software admission: live mirrors, quarantine, or an allowlist.
4. The RStudio edition for `gui_only` studies: Workbench, desktop only, or an accepted risk.
5. Whether to reuse public open-source landing-zone code for the foundation. UCR's CUI enclave stays a separate service with its own scope either way.
6. Whether to evaluate Globus as an ingress path.

Before the pilot, I would add FERPA to scope and intake, name the accountable data owner and get the systems team's written acceptance of its roles. The image work comes next: make the login gate's status boot-scoped, widen the idle check, point the alerts at the right log, and fix the package-mirror configuration. The perimeter needs its API list, network ranges and administrator access level checked against Google's pages, and the log sink needs its rules corrected. Set soft delete and extend the teardown check, add client keepalives, pin R, verify the build script's downloads, and add a cross-study negative test and researcher-time measures to the pilot. Before the first P3–P4 data we need the campus security office's risk treatment plan and the security officer's acceptance (including D4), a decision on who may act as the deployer, an enforced perimeter, the locked log bucket, a tabletop exercise and the penetration test.

Chapter 1

Why UCR Needs a P3–P4 Environment

UCR researchers already have places to compute on public and low-risk data, and one place to compute on Controlled Unclassified Information (CUI). What they do not have is a campus place to compute on data that University of California policy classifies at Protection Level 3 or 4 (P3 or P4) when that data is not CUI: health records received from outside hospitals under a data use agreement, student education records, large sets of personal information, and research data that an Institutional Review Board or a contract designates as P4. A researcher who receives such data today has to find a home for it case by case. Our plan exists because the campus has no shared answer. It is a draft (version 3, frozen at a single repository revision) for a shared secure environment on Google Cloud. Nothing in it has been built: the Terraform has been validated and planned offline with placeholder values, the image scripts have never run, and no cloud resource has been created for it.

Below I set out the choices we made and the reasons the plan gives for them, including one inconsistency in our funding position that the plan leaves open.

1.1 UC protection levels and the research data that lands in them

The University of California classifies institutional information into four Protection Levels. P1 is minimal-risk, essentially public information. P2 covers information not specifically protected by law or contract but not meant for public use. P3 (moderate) covers information whose unauthorized disclosure could lead to small to moderate fines, penalties or civil actions, and it explicitly includes lower-risk items that, when combined, represent increased risk. P4 (high) covers information whose compromise could result in significant fines, penalties, regulatory action, or civil or criminal violations, where statutory, regulatory and contractual obligations are the main drivers [4]. The system-wide policy that attaches controls to these levels is BFB-IS-3, Electronic Information Security, together with its supporting standards [5].

The UC Protection Level Classification Guide lists examples at each level [6]. Three entries matter for the population this plan serves. Student education records, which are the records FERPA governs, are listed at P3. Protected health information (PHI), medical records and patient records are listed at P4, as are Social Security numbers, individually identifiable genetic information about human subjects, information with contractual requirements for P4-level protection, and research information classified as P4 by an IRB. A further P4 entry covers personally identifiable information when it appears in large sets and contains a comprehensive set of information about a person. The practical consequence for a student-records study is that the level depends on content: an extract of ordinary education records sits at P3, unless it is identifiable human subject research data or carries student disability, medical or financial-aid information, all of which the guide lists at P4, while the same extract joined to Social Security numbers, or assembled into a comprehensive per-person file, lands at P4 by the guide's own entries. In UC practice the accountable data owner and the local information security lead make the final call for a given dataset, so we treat P3 and P4 as one population and build to the stricter level.

HIPAA reaches UC research through a narrower door than many researchers assume. According to the UC Office of the President, UC is a HIPAA hybrid entity with a single health care component made up of medical centers, clinics, student health and health plans, and the research function sits outside that component;

research health data that is not tied to health care services is therefore not itself subject to the Privacy and Security Rules at UC [7]. Such data is still P4 under IS-3, and California's Confidentiality of Medical Information Act, the Information Practices Act and the Common Rule still apply. HIPAA obligations arrive at a UCR project through the data source's paperwork: a business associate agreement, a data use agreement for a limited data set, or a data use agreement from an outside covered entity. The terms in those documents (often a promise to meet Security Rule safeguards, a breach notice window, and a bar on subcontractors without approval) become binding on the study. Chapter 3 covers the legal text of HIPAA, FERPA and IS-3; whatever the legal route, the data sources and UC policy set the same technical bar.

The Security Rule itself is under revision. HHS published a proposed rule in the Federal Register on 6 January 2025 (90 FR 898) titled "HIPAA Security Rule To Strengthen the Cybersecurity of Electronic Protected Health Information," with comments closing on 7 March 2025 [8]. As of 1 November 2026 the rule is still a proposal: the current eCFR text is unchanged, and the Unified Agenda lists the rule under Long-Term Actions with a projected final action date of July 2027 (Chapter 3) [9]. We build toward the proposal (mandatory MFA and encryption, asset inventory, tested restores) because IS-3 already requires most of it at P3 and above.

1.2 The gap: no campus place for P3-P4 compute

The plan's stated purpose is one sentence: give UCR researchers with P3-P4 data, including PHI from outside covered entities under a BAA or data use agreement, a place to compute, because no other campus option exists. The same section adds that the environment might be subsidized under the no-charge tier of the campus research cluster, a shared Slurm cluster on Google Cloud, and that cost and build effort should stay low by sharing infrastructure across studies. The funding clause is examined in Section 1.7.

The gap is not a lack of cloud accounts. A researcher can open a Google Cloud project, and UCR has an executed Google Cloud business associate agreement. Google's own HIPAA page describes compliance as a shared responsibility, states that Google Cloud supports HIPAA compliance within the scope of a BAA, and says the customer remains responsible for evaluating its own compliance [10]. A signed BAA therefore lets PHI sit on covered services; it does not supply segmentation, access review, log review, change control or disposal evidence. Those are the controls IS-3 asks for, and an individual lab is poorly placed to run them. Our working list of controls for a P3-P4 compute environment has 71 numbered rows grouped under nine headings (governance, people, access, encryption, network, operations and logging, development, suppliers, and media and incident handling). The document's own text says 63 controls; its tables hold 71 rows, and the figure counts rows. Figure 1.1 shows how many of them apply at each level.

Many of these rows are process rather than technology: background checks for administrators with P3 access, a training gate before access, signed acceptable-use or data security plans, quarterly access recertification, a risk assessment or a risk treatment plan approved by the security officer, and evidence that controls are in place and current. A shared environment makes sense for exactly this reason. The process controls cost roughly the same for one study as for ten, so one reviewed platform that each study's data security plan can inherit is cheaper than ten bespoke ones. Our idea is simple: approve and operate the platform once, and let each study's plan record only what differs.

1.3 Why the shared open cluster cannot host P3-P4 data

The campus research cluster is a shared Slurm cluster on Google Cloud, and it is the closest existing resource to what a P3-P4 researcher needs: batch compute, GPUs, and an operating team. The obvious first proposal is

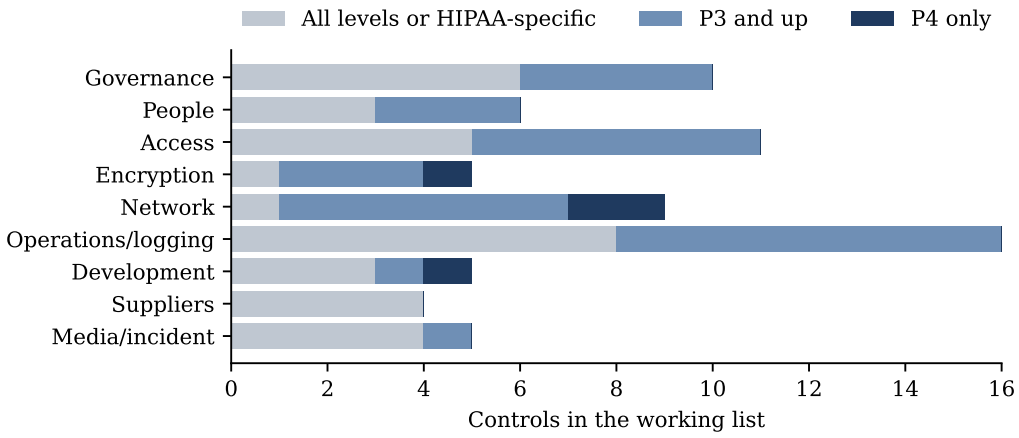


Figure 1.1: Rows in the research computing team's working list of IS-3 controls for a P3-P4 compute environment, by heading and by the level at which the row applies (counted from the list; the list is a working analysis, not a campus security office determination). Rows marked "all levels" apply at P4 as well and include the rows the list assigns to P2 and above; the HIPAA-specific rows are in the same group.

to harden it. Our own gap analysis checked that option against the control list and rejected it. The reasons come from what the cluster is for. It was built as an open, shared, P1-P2 system, and the features that make it useful to every lab are the features that fail the P3-P4 list.

Table 1.1 summarizes the conflicts the analysis recorded. I describe them at the level of design properties of an open shared cluster; they say what such a cluster is for, and they are not a security assessment of any running system.

The shared file system is the deepest conflict, because the cluster's usefulness comes from every node seeing the same home, application and scratch directories, and that sharing is the opposite of the isolation a data use agreement usually demands. Open egress comes next: researchers install packages and pull containers from public registries every day, and a single outbound path serves both that purpose and any copy-out. AI tooling is a newer problem. Assistants that read job output and logs are useful on an open cluster; on a P4 boundary, job output would become input to an AI service that no one has approved for that data. Then there is the operating model. The model under which P3-P4 work is meant to run separates the team that builds the research tooling from the teams that administer the systems and review the logs; an open cluster run by one team does not have that separation.

The analysis also records what already works: nodes have no public IP addresses, access runs through Identity-Aware Proxy and OS Login with campus single sign-on, the operating system is supported through 2029, the infrastructure code is under version control, and Google encrypts data at rest by default. Its conclusion is that retrofitting the main cluster to P4 would mean removing what makes it useful to everybody (internet, shared software installs, portals, AI tools) and putting every P1 user under P4 rules. It recommends a separate environment instead, and I agree.

The gap analysis is ours, and the plan departed from it

Two cautions apply to the gap analysis itself. First, it is our own working document and not a determination by the campus security office, which has not signed off on its findings or on its control list. Second, the analysis recommended something different from what the plan then specified. It proposed one hardened cluster per study, each in its own project with customer-managed encryption keys, on the argument that

Table 1.1: Why an open shared cluster does not fit a P3-P4 boundary, generalized from the research computing team's gap analysis.

Design property of the cluster	Why it conflicts with P3-P4
Every P1-P2 user and lab shares nodes, file systems and scratch	Not a segmented network holding only P3+ resources; no need-to-know separation between labs
Shared network file system for home, apps, scratch and shared data, separated by Unix permissions	Root on any node reaches all of it; the shared protocol was chosen for convenience, not in-transit protection
Open outbound internet access for software installs	Software installs and data copy-out share one path
Logging, monitoring and key custody designed for open (P1-P2) use	The P3-P4 rows for data-access records, protected log copies, log review, security testing, key custody and API perimeters apply only above P2
Interactive portals, notebooks and desktops	Extra access paths into what would be a P4 boundary
Users install software freely from public sources	No change control on software that touches P4 data
Operating and people processes sized for an open service	Separation of duties, administrator screening, training gates and access recertification are P3-P4 requirements the open service was never designed to carry
Shared billing	P3-P4 projects may be expected to pay their own way on a separate account

> data use agreements often forbid commingling studies. The plan (decisions D3, D4 and D5) chose one shared environment with Google's default encryption and individual virtual machines first. We changed course by decision, recorded in the plan; Chapter 6 weighs the reasons and the counter-arguments.

1.4 Scope and routing

The plan covers UC IS-3 P3-P4 research data: HIPAA PHI received under data use agreements, FERPA student education records, large personal information sets, IRB-classified P3 and P4 data, and contractual P4. It does not cover everything sensitive that UCR researchers handle, and the boundaries matter because researchers need to know where each kind of data should go.

1.4.1 UCR's separate CUI enclave

UCR already runs a separate CUI enclave. Google Cloud's public account of the work says UCR partnered with Google Public Sector to build it with Stellar Engine, Google's published infrastructure-as-code foundation, on infrastructure designed to support NIST SP 800-171 and CMMC Level 2 [11]. The enclave serves a different legal population: work that must meet NIST SP 800-171 or CMMC.

The CUI enclave is the sibling service for that population and the routing target for it. It is not a venue for the P3-P4 environment in this book, and the plan does not build on it. Several of the AI-assisted research reports run during this study describe the CUI enclave's foundation as the vehicle for the P3-P4 design; that is wrong, and I don't repeat it. Where the two services share something, such as the habit of expressing a secure foundation as reviewed infrastructure code, the similarity is incidental and the services have separate scopes, separate operating teams and separate approval paths.

One routing rule surprises researchers. NIH controlled-access genomic data, such as data obtained through

dbGaP, falls on the NIST SP 800-171 side of the line, although it is not CUI. NIH notice NOT-OD-24-157, effective 25 January 2025, states that NIH expects Approved Users to keep controlled-access data on institutional and third-party systems that meet NIST SP 800-171 and to attest that their institution complies; the expectation is written into new and renewed Data Use Certifications [12]. A researcher with such data therefore goes to the CUI enclave even though the data is also highly sensitive health information and would be P4 under the UC guide, which lists individually identifiable genetic information at that level [6]. A study can sit on both sides of the line: genomic data under NIH terms in the CUI enclave and a separate clinical or education data set under a data use agreement in the P3-P4 environment, with intake deciding which data goes where.

1.4.2 Medical school clinical data

Clinical data that originates in the medical school stays with the medical school's own solutions, by data source. The planned routing treats this as settled: clinical data originating in the medical school stays with the medical school's IT group, and PHI from outside hospitals into a study outside the medical school goes to the secure environment this plan describes. I make no claim about how the medical school or other campus units operate their own systems beyond what public sources show.

1.4.3 The routing diagram

Figure 1.2 puts these rules in one picture. The diagram describes intended routing; the P3-P4 environment in the middle row does not yet exist, and until it does, the first row of the intake decision for such data has no destination.

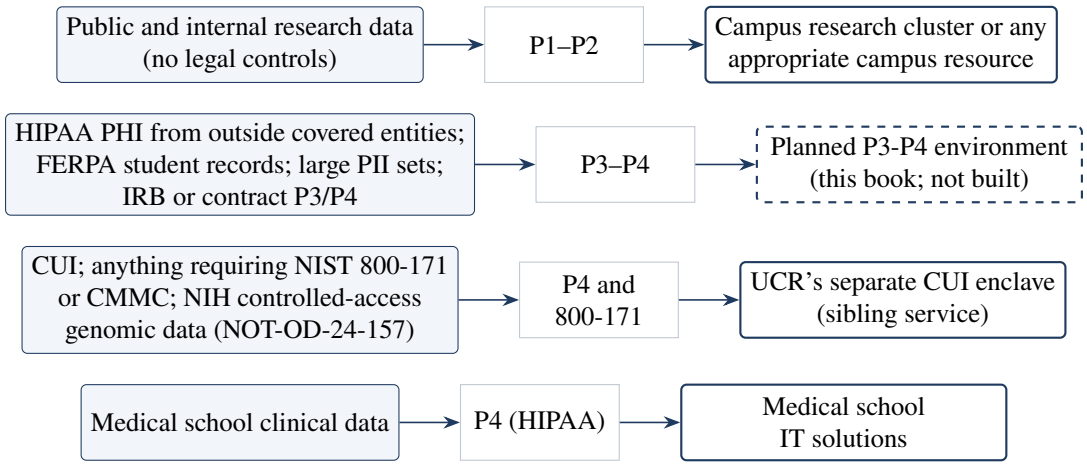


Figure 1.2: Intended routing of research data by type, protection level and destination. The dashed box marks the environment I examine here; it exists only as a plan. The routing follows the plan's scope decision and the routing rule for medical school data.

1.5 The decisions D1 to D14 in plain words

The plan records fourteen decisions by research computing leadership (D9 follows the first review panel and D14 the second, both described below). Table 1.2 restates each in plain words with the chapter that examines it. The decisions are the plan's own; Chapter 14 asks whether they are right.

Table 1.2: Decisions D1 to D14 of plan v3, restated in plain words. Status for all rows: decided in the plan; none implemented.

ID	Decision in plain words	Examined in
D1	UCR has a signed Google Cloud BAA. Only services the BAA covers may touch PHI.	Ch. 3, 4
D2	Duo multifactor authentication is enforced at campus sign-in to Google, and the plan relies on that for the IS-3 MFA requirement; no code in the plan enforces it.	Ch. 3, 9
D3	One shared environment serves all studies, with no project or cluster per study.	Ch. 6
D4	Google's default encryption at rest, no customer-managed keys, recorded as a risk the security officer must accept.	Ch. 6
D5	Individual secure virtual machines first; a secure cluster only if demand appears.	Ch. 13
D6	All secure VMs go in one dedicated Google Cloud project, separate from the campus research cluster's project.	Ch. 5, 6
D7	Reuse the campus research cluster's infrastructure (repository, tooling, image ideas) only where doing so opens no path between the two zones.	Ch. 5
D8	Funding: possibly subsidized under the campus no-charge tier; open.	Ch. 12; Section 1.7
D9	Adopt all recommendations of the first review panel (plan v2).	Ch. 14
D10	Offer RStudio Server over IAP and a Linux desktop over IAP, the desktop through a native RDP client.	Ch. 9
D11	A small separate log project, administered by the campus security office with no compute in it, holds audit logs and the Terraform state.	Ch. 10
D12	State the copy-out risk instead of hiding it: the default shell mode accepts it (prohibited, logged, accepted by the security officer), and a `gui_only` mode with no shell, no file copy and no clipboard serves studies whose agreement forbids copy-out.	Ch. 7
D13	Outside collaborators receive UCR affiliate accounts under the same data security plan; IAM stays limited to the campus domain.	Ch. 9, 11
D14	Adopt all recommendations of the second review panel (plan v3).	Ch. 14

D3, D4 and D12 shape the plan more than the rest. With D3 and D4 we traded per-study isolation and key custody for lower cost and effort, and recorded both as risks for the security officer to accept. D12 says a user with a shell can copy data out; it names that risk and offers a mode without a shell. All three are on the list of items to be agreed in writing before any PHI arrives.

1.6 How the plan reached version 3

The plan went through three versions in one day. Version 1 contained the VM templates and the foundation Terraform. A six-lens review panel of a lightweight language model reviewed it (one model, single-shot reviews, each finding checked by hand against the files before we accepted it), and the combined verdict scored it "not ready" on all six lenses while calling the design sound and the code buggy, with the process and ownership layer missing. Version 2 added baked images, boot checks, a separate log project, a deployer service account, alerts, a budget and a billing export, and the RStudio and desktop access paths. A ten-lens panel then reviewed version 2, found that most new bugs were in the newly added and never-run code, and pressed for plain statements of what SSH access can prevent and of how data is disposed of. Version 3 added a fail-closed login gate, locked audit rules, a teardown script that writes evidence, restore objectives, an export release checklist, the `gui_only` mode, and a rebuilt table of roles.

These panels are model reviews. They find defects in code and gaps in documents, and we checked each finding by hand, but they are not independent expert review and they did not involve the campus security office, the systems team or the security officer. Chapter 5 describes what the panels changed in the design and Chapter 14 assesses what remains untested.

The plan separates what the code would build from what remains process. Its built-versus-process table lists fifteen areas of IS-3 coverage; in fourteen of them it names process work still needed, in three (pentest, supplier confirmation, people) the built column is empty, and in a fourth (MFA) it says that nothing in the code enforces the control. The process work it names includes the campus security office's acceptance of the shared boundary, evidence of Duo enforcement, the security officer's acceptance of default encryption, weekly review of administrator activity, a patch cadence, restore objectives and testing, an export release record, a pentest, confirmation that the services used are on Google's BAA list, training and access recertification, and second-person change review. The plan also lists nine agreements that must be signed before any PHI: the role table signed by the research computing team, the systems team, the campus security office, the security officer and the accountable data owner; an incident and breach matrix; the risk treatment plan with the security officer's acceptances; a per-study agreement gate; an access process; retention and disposal; restore objectives; export release rules; and a funding model. Its open items include the choice of SIEM, the systems team's commitment to operate and be on call, and who the accountable data owner is (Chapter 14 lists them).

1.7 The funding question

Our planning record holds two funding positions, and we cannot follow both. The plan's purpose section and decision D8 say the environment might be centrally subsidized under the shared cluster's no-charge tier, and D8 is marked open. The gap analysis instead works from a project-pays model: P3-P4 projects pay all of their costs through a chart of accounts (COA), a memorandum of understanding (MOU) and a separate billing account.

The two positions imply different designs. Under a subsidy, the platform's fixed costs and some per-study compute would be absorbed by a campus program, and the plan's billing export and per-study labels exist so that the cost of the subsidy is visible. Under a COA/MOU rule, each study would pay for its own virtual

machines, disks, snapshots and log storage, and someone still has to pay for the shared parts: the foundation and log projects, the campus security office's logging and review time, incident response, and the pentest the plan lists before the first P4 data. The plan itself lists, among the agreements needed before PHI, a funding statement covering the scope of any subsidy, a per-study ceiling, who pays above it, and who funds incident response, pentest and log storage.

I name the inconsistency here and leave the choice to leadership. Chapter 12 computes per-VM costs from published prices and shows how each funding model changes the arithmetic.

If we pick a funding rule

If the COA/MOU rule governs, D8 should be rewritten, the subsidy arithmetic in the plan's cost section becomes a statement of what each study would be invoiced, and the shared costs need an owner before the first study signs. If a no-charge-tier subsidy governs, the plan should say which studies qualify and what happens when the subsidy is exhausted, and the project-pays rule needs an exception on record. Until one of these is chosen, a researcher asking what the environment would cost cannot be given an answer.

1.8 Design considerations

1. **State the population in one sentence at intake.** The scope rule (P3-P4 data that is not CUI, with HIPAA PHI and FERPA records as the main cases) should appear on the intake form together with the routing diagram, so that a researcher with dbGaP data or medical school clinical data is redirected before any design work starts.
2. **Decide the classification owner for mixed data sets.** The UC guide places ordinary student education records at P3, but identifiable human subject research data, student disability, medical or financial-aid information, Social Security numbers and comprehensive PII sets at P4, so a student-records study cannot default to P3. The plan should say who confirms the level for a given study (the accountable data owner with the unit information security lead (UISL)) and record the decision in the study's data security plan.
3. **Ask the campus security office to review the gap analysis and the control list.** The list has 71 rows although its text says 63, and it has not been reviewed by the campus security office. A reviewed and renumbered list would become the shared baseline for the risk treatment plan.
4. **Resolve D8 against the COA/MOU rule before any study is onboarded.** Either document an exception to the rule or rewrite D8. Name the owner of the shared costs (foundation, log project, security review time, pentest) in either case.
5. **Give the plan a destination in the interim.** Until the environment exists, researchers with P3-P4 data from outside covered entities have no campus option. The intake process should say what such a researcher is told today, including whether an individual-lab project under the BAA is acceptable with campus security office review.
6. **Keep the CUI enclave relationship to routing only.** Intake, documents and training should describe the two services as siblings with separate scopes, and any shared practice (for example reviewed infrastructure code) should be documented as a convention, not as a dependency.
7. **Record why the plan departed from the gap analysis.** The analysis recommended per-study projects with customer-managed keys; D3 and D4 chose the opposite. A short decision record that states the reasons and the conditions that would reverse them (taken up in Chapter 6) would let the campus security office and the security officer judge the risk acceptance on the same footing as the designers.

8. **Treat the panel reviews as defect-finding, not assurance.** Plan the human reviews that the panels did not provide: the systems team on operability, the campus security office on controls, and the security officer on risk acceptance.

Chapter 2

How Peer Universities Run Secure Research Environments

Universities that host sensitive research data have converged on a small set of shapes. A central research computing group, usually with a health-sciences or security partner, runs a separate environment for the data that campus policy rates as high risk. Researchers reach it through a gate: a data risk review, an IRB or data use agreement check, or a project request that a human approves. Inside, the work happens on a managed desktop, a Slurm cluster behind a VPN, or a virtual machine, with software installed by staff and data leaving only by a reviewed route. The shapes differ in where they run (a campus data center or a public cloud), who pays (central subsidy, chargeback, or a mix) and how much of the control set is shared across projects. Those three differences matter most for our plan, which would run one shared environment for P3 and P4 data on Google Cloud.

I set the plan beside what peer institutions publish about their own environments, and beside UK and European practice, which has the most developed written standards. I read every institutional fact in the tables on that institution's own web page on 1 November 2026, or label it *reported*. Public pages show what an institution says it offers, not how well the service works, and they say nothing about staffing, incident history or audit results.

2.1 Method and what counts as verified

The peer set is not a systematic sample. I seeded it from the institutions named in one AI-assisted research report on peer university enclaves, plus two further reports on peer benchmarking and secure Slurm clusters, and then checked each institution on its own pages. A different seed would have produced a different list; in particular, campuses with good services but little web documentation are under-represented. Those reports name about twenty institutions and give detailed profiles of them, with core counts, storage volumes and prices. The reports cite campus pages that I could not match to the claims in several cases, so I treated every profile as a lead. For each institution I fetched the current service page, grepped it for the claim, and kept the claim only if the page said it. Where a campus page blocked scripted requests (Michigan's page returned an access challenge to `curl`), I read it through a page-extraction tool and say so in the bibliography. Where I found no usable page, the institution appears in the text as "reported" or not at all.

Fourteen US institutions have a row in Tables 2.1 and 2.2: Stanford, Harvard, Yale, Duke, Michigan, Penn, Johns Hopkins, NYU, Columbia, Utah, UC Berkeley, UC San Diego, UCLA and UCSF. Five more (WashU, Purdue, Indiana, UC Irvine and UC Davis) are discussed in the text only, because their pages describe a CUI-only service, a consulting program or a mapping table, or could not be read. UC Davis is the one campus on the seed list that I could not verify: its pages returned a block page to scripted requests, so the report's statement that it deploys Azure subscriptions through a service called AggieCloud stays unconfirmed.

A row records an *offering* (Stanford has two, Duke has four, UCSF has two), so the 14 institutions produce 19 rows. UCSF's research-computing page also lists CoreHPC, its general high-performance computing environment, which it says supports P1 to P4 data [13]. It has no row because the page presents it as the campus's general cluster rather than a service for sensitive data, and it does not state hosting, so it would not

change the hosting split. Duke's Research Compute Cluster has a row because Duke's page lists it among its environments for sensitive data. I record "hosting" only when the page says where the systems run. A page that describes a "protected virtual network space" or a "virtual server environment" without naming the platform counts as not stated, even when the report claims otherwise.

2.2 Two hosting patterns, and how many peers use each

On the verified rows, on-premises and cloud-hosted environments both exist in every size, and the largest group of pages says nothing about hosting. Of the 19 offerings, 11 state their hosting. Six are cloud-hosted: Stanford Nero (Google Cloud), the Duke School of Medicine Azure subscription, the UCSF AWS Secure Enterprise Cloud, UC San Diego's Sherlock (AWS, Azure, Google Cloud or the San Diego Supercomputer Center data center), the Johns Hopkins SAFE Desktop ("cloud-based") and the NYU Secure Research Data Environment (which uses cloud computing resources, in the page's wording). Five are on-premises: Stanford Carina (an "on-prem" environment on the Stanford Research Computing private cloud), the Duke Research Compute Cluster ("on prem"), the Harvard FASSE cluster (servers in an access-controlled data center), the Utah Protected Environment (hardware in a surveilled data center) and the UCLA High Compliance Environment ("traditional (on-premise)" network, compute and storage). The other eight rows (Table 2.1) do not name a platform on the pages I read. Figure 2.1 shows the split.

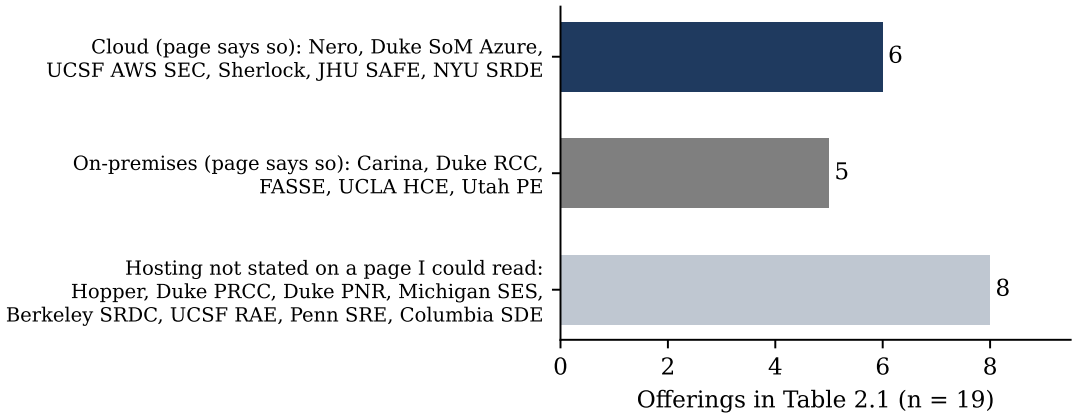


Figure 2.1: Hosting of the 19 offerings in Table 2.1, as stated on each institution's own page (read 2026-11-01). Rows whose pages do not name a platform are counted separately, not assigned to either side.

The count does not support a statement that peers are moving to the cloud, and it does not support the opposite. It supports a narrower one: a cloud-hosted environment for high-risk research data has precedent at several large universities, and only one of them, Stanford's Nero, is documented as running on Google Cloud. Penn's cloud page states that its GCP offering is "in development," and Sherlock lists Google Cloud among the clouds it supports. The peer-enclaves report describes Johns Hopkins SAFE as built on Azure with Citrix and Penn's enclave as AWS-based. The Johns Hopkins page says "cloud-based" and nothing more, and Penn's page names no provider, so both details remain reported.

Cloud-hosted P3-P4 environments have peer precedent, but only Nero is documented as running on Google Cloud

Six of the eleven offerings that state their hosting are cloud-hosted and five are on-premises. Only Stanford Nero is documented as running on Google Cloud, with a per-project model and native Google services; Sherlock lists Google Cloud among the clouds it supports. Our platform is not unusual in kind. Our single shared project for all studies is: Nero gives each requesting team its own Google Cloud project, and no peer page describes one shared project for all studies.

Table 2.1: Peer offerings for high-risk research data: platform, data classes and operator, as stated on the institution's own page (read 2026-11-01). "ns" means the page does not state the item. Sources are cited in the first column.

Institution and offering	Hosting	Data classes named	Operator
Stanford Nero [14, 15]	Google Cloud	"High-risk" data, including NIH controlled-access data	Research Computing with the School of Medicine
Stanford Carina [16, 17]	On-premises (private cloud)	PHI and other high-risk data	Research Computing with the School of Medicine
Harvard FASSE [18]	Data center (on-premises)	Level 3 data under DUAs and IRBs; *not* rated for Level 4	FAS Research Computing
Yale Hopper [19]	ns	ePHI, NIH controlled-access data, CUI, other sensitive data	Research Computing with Health Sciences IT
Duke PRCC [20]	ns	Identifiable Duke Health patient data	ns
Duke PNR [20]	ns	Sensitive and regulated data; no Duke Health patient data; 800-171	ns
Duke SoM Azure [20]	Azure	PHI including Duke Health patient data; 800-171 and 800-53	ns
Duke RCC [20]	On-premises ("on prem")	Sensitive data; 800-171 but not 800-53	ns
Michigan SES [21]	ns ("virtual server environment")	Up to "high," including CUI	Advanced Research Computing, ITS
Penn SRE [22]	ns	ePHI, NIH controlled-access genomic data, CUI, "High"	ISC Cloud Solutions with Research Services
Johns Hopkins SAFE Desktop [23]	Cloud ("cloud-based")	PII and PHI	Central IT
NYU SRDE [24]	Cloud	Moderate and high risk; 800-171 intent	Research Technology Services
Columbia SDE [25]	ns	ns	CUIT Research Services
Utah Protected Environment [26, 27]	Data center (on-premises)	HIPAA PHI and PII	CHPC
UC Berkeley SRDC [28]	ns	P4, including PHI and CUI	Research IT with campus partners
UC San Diego Sherlock [29]	AWS, Azure, Google Cloud or SDSC data center	HIPAA, FISMA, CUI, FERPA	San Diego Supercomputer Center
UCLA HCE [30]	On-premises	P4 focus, including 800-171, DFARS, PCI	Digital and Technology Solutions
UCSF RAE [13, 31]	ns	P1 to P4	UCSF IT
UCSF AWS SEC [13]	AWS	P1 to P4	ns

2.3 How researchers get in, and what the gate checks

Table 2.2 lists how each offering is reached, who pays, and what must happen before a project exists. The strongest regularity is the gate. Every offering whose page describes onboarding puts a review before the data lands (Berkeley's page names a classification consultation). Stanford requires a Data Risk Assessment by its Information Security Office and University Privacy Office for each new dataset, and exempts datasets already covered by an assessment (the Stanford Medicine research data repository and Population Health Sciences) [15]. Harvard requires its research computing group, under the Harvard Research Data Security Policy (HRDSP), to review the DUA, data transfer agreement or IRB documents before a FASSE project is created and before any data is copied to the cluster, and sends researchers to the research office (OVPR) first [18]. NYU requires IRB approval or a waiver and lists the SRDE owner as the same person as the IRB protocol's PI [24]. UCLA conducts a risk assessment on each application before accepting it [30]. Yale grants access per project on a PI's request, and the PI must then approve each added account [19]. Our intake checklist and signed Data Security Plan (DSP) come at the same step.

After the gate, the access models split by data class and discipline. Windows desktops are common where the users are clinicians and social scientists: UCSF RAE offers Windows virtual machines [31], Johns Hopkins SAFE Desktop is a Windows virtual desktop [23], Utah runs a Windows server in its Protected Environment [26, 27], and Berkeley's SRDC offers Linux and Windows VMs [28]. Slurm clusters with Open OnDemand desktops sit at the campuses with a computational-science user base (Carina, FASSE, Hopper, Utah). Stanford's comparison page lists the same analytical tools in both of its offerings: Python, R, RStudio, SAS and Stata (the last two with proof of an existing license) [17]. The plan documents I read describe RStudio Server and an XFCE desktop on Rocky Linux VMs and do not mention SAS, Stata or a Windows image, so a researcher whose workflow depends on those tools has no stated path. That is a question for the pilot, not a defect in the design.

Several pages describe controls that the plan would have to match or consciously decline. Michigan keeps administrator rights for machines inside an enclave with its own staff, takes software requests by ticket, requires the researcher to supply licensed software media, and reserves the right to reject software that looks like a risk to the enclave. Michigan also states that restricted data must not leave the enclave unless the faculty sponsor consents [21]. Harvard turns off interactive job allocation and sends users to its virtual desktop instead [18]. Yale states that it deactivates accounts for owners who fail to renew annual training or who violate security measures, and deactivates whole projects whose PI misses the quarterly account review; the listed violations include taking photographs of screens [19]. That last item is the analog hole that Chapter 7 treats as a residual risk: a peer writes it into the rules for the account, which makes it an account-termination offense without making it technically preventable. Utah isolates the Protected Environment on separate subnets and VLANs, places VMs in project-specific groups, and describes the environment as "shared" with groups isolated by Unix access controls and other controls [27]. That is a shared platform with per-project separation, the model we chose in decision D3.

Peers that describe onboarding put a human review before data lands

Stanford, Harvard, NYU, UCLA and Yale each put a human review of the data, the agreement or the project ahead of project creation; Duke's pages name intake forms and Berkeley offers a classification consultation. The review draws on the IRB, the DUA or a campus risk office, and the environment is configured to match. Our intake checklist and signed DSP sit at the same point in the sequence.

2.4 Who operates the environment, and who pays

Table 2.2: Access model, cost to the researcher and onboarding gate for the offerings in Table 2.1. Prices are as published on the pages cited there, in US dollars, and are not adjusted. "Not found" means I did not find the item on the pages I read, not that none exists. Verified: Y means the institution's own page was read for this row; Y* means only a meeting-minutes document was read.

Offering	Access model	Cost to researcher	Gate	Verified
Nero	SSH, JupyterHub, Google Cloud console and APIs	Billed to the PI's project account at Google list rates less at least 18%; list example about $290 per month for 8 vCPU and 30 GB; network ingress and egress fees waived [32]	Data Risk Assessment; PTA number	Y
Carina	SSH, Slurm, Open OnDemand, NoVNC	Free; 1 TB per project; extra 10 TB units at $100 per month [33]	Data Risk Assessment	Y
FASSE	VPN, then Open OnDemand desktop or SSH; interactive `salloc` disabled	Not found	HRDSP review of DUA and IRB	Y
Hopper	Yale VPN only, even on campus; web portal, SSH	No service-unit charge from 2026-07-01; storage beyond the no-cost allocation $5.30 per TiB per month	PI project request; annual training	Y
Duke PRCC	ns	Fee schedule (amounts not read)	Request for PRCC services	Y
Duke PNR	ns; "no high performance compute"	No cost for standard configuration, storage and tools; extra compute and storage for a fee	Research and Innovation intake form	Y
Duke SoM Azure	ns	Fee (amounts not read)	ServiceNow request	Y
Duke RCC	HPC	Fee schedule (amounts not read)	ServiceNow request	Y
Michigan SES	Virtual servers; researchers get no admin rights	$7.00 per GB RAM, $3 per GB GPU RAM, $15.41 per TB storage, each per month (cost recovery)	Not found	Y
Penn SRE	ns	Cloud Solutions rates are direct billed at provider on-demand rates; SRE price not found	Request through ISC	Y
JHU SAFE	Windows virtual desktop; collaborators with a JHED ID	Free to 100 GB for JHU faculty and staff with a JHED ID [23]	IRB and data trust criteria	Y
NYU SRDE	Secure virtual environments	Not found	IRB approval or waiver; PI eligibility; a named Data Steward	Y
Columbia SDE	Not found	Not found	A named Data Security Officer; central DSO service now offered	Y*
Utah PE	Open OnDemand, Windows server, VM farm, HPC cluster	Not found	Project request	Y
Berkeley SRDC	HPC and Linux or Windows VMs	Baseline allowance at no cost (VMs capped at 150,000 core hours per year); condo purchase of nodes at cost over a 5-year lifecycle [34, 35]	Consultation to determine classification	Y
Sherlock	Secure desktops, compute, storage	Service agreement or institutional MOU; fee for service	Contract	Y
UCLA HCE	VDI to segmented workloads	No cost to campus for accepted projects; 1 TB standard storage	Risk assessment per application	Y
UCSF RAE	Windows remote desktop with Duo MFA; no GPUs	Free tier (100 GB shared); Premium dedicated VMs $300, $700 or $1,300 per month	ServiceNow request	Y
UCSF AWS SEC	AWS	Pay for what is used	ns	Y

2.4.1 Operators

Peer environments are rarely run by one unit. Stanford's Nero and Carina are described as collaborations between the School of Medicine and Research Computing [14, 16]. Yale's Hopper is "jointly supported" by its research computing center and Health Sciences IT [19]. Penn's enclave is operated by its cloud solutions group "in partnership with" the Office of Research Services [22]. Berkeley's SRDC was "developed and operated by RTL and a coalition of partners" (RTL is Research, Teaching, and Learning) [28]. Johns Hopkins' SAFE Desktop is operated by central IT, with the Johns Hopkins Medicine Data Trust Council's criteria shaping its use [23]. UC San Diego's Sherlock is a unit of the San Diego Supercomputer Center that sells secure-cloud services as a provider, with institutional MOUs as one contract form [29, 36]. Usually a technology group runs the platform and a compliance or privacy group approves projects. Our RACI draws the same line between the systems team and the campus security office.

Peers have solved the steward problem in different ways. Columbia's committee minutes record a new "Data Security Officer as a Service" for its Secure Data Enclave: research teams that cannot name a qualified DSO from their own staff may engage a trained central IT employee to fill the role [25]. The plan assigns a data steward per study to release exports. A study team with no suitable steward is a realistic case, and Columbia's solution is a central pool. NYU requires a Data Steward, a faculty member or university employee, for each SRDE project, "solely responsible for the transport of data" [24], which is the plan's per-study data steward under the same name. Indiana University announced a different model in 2020, SecureMyResearch, which offered consulting and a "cookbook" of recipes for common research uses, some of them pre-approved for data such as HIPAA PHI [37]. It is a program that routes researchers to approved platforms, not an enclave. The UK literature adds a third observation, from a 2026 talk on TRE infrastructure staffing: running a TRE draws on a wide range of roles beyond software developers, including information governance specialists, project managers and public-engagement staff [38]. It comes from a single program and says nothing about headcount.

2.4.2 Cost to the researcher

Of the 14 institutions, seven offer a no-cost baseline for at least one offering: Stanford Carina, Yale Hopper (no service-unit charge since 1 July 2026), Berkeley SRDC (a baseline allowance), UCLA HCE (no cost to campus for accepted projects), Duke PNR (standard configuration), UCSF RAE (a free shared tier) and Johns Hopkins SAFE Desktop (free to 100 GB). Five charge the researcher or the researcher's unit: Stanford Nero, Duke PRCC, SoM Azure and RCC, Michigan SES, UCSF RAE Premium and AWS SEC, and Sherlock. Harvard, NYU, Columbia, Utah and Penn (for its enclave) did not state a price where I looked. The counts are by offering, so some institutions appear on both sides. The picture is mixed: two of the on-premises offerings with stated hosting are free at the point of use (Carina, UCLA HCE) and one charges a fee (Duke RCC); most cloud-hosted offerings with a stated price pass costs through, and Johns Hopkins SAFE Desktop, free to 100 GB, is the exception. UCSF RAE and Duke PNR also have a free tier, but their pages do not state hosting.

The cloud-hosted offerings show the pass-through model most clearly. Stanford bills Nero projects to the PI's project-task-award number at a negotiated discount of at least 18% from published Google Cloud rates, waives ingress and egress fees, and states list prices that are comparable in scale to the per-VM figures in the plan's cost templates (Chapter 12 recomputes those from published prices) [32]. UCSF Premium charges flat monthly fees per dedicated VM, between $300 and $1,300 [31]. Michigan's rates are cost recovery per gigabyte of RAM and terabyte of storage [21]. The peer-enclaves report described Michigan's secure enclave as including a free base tier of 80,000 CPU hours, 16 GB of RAM and storage. The current SES page lists only cost-recovery rates and no free tier. I did not check whether the Research Computing Package covers other Michigan services.

Peers that subsidize mostly run their own hardware

Our open funding question (D8 against the project-pays rule, Section 1.7) has both answers in use at peers. Peers that subsidize and state their hosting are mostly on-premises (Carina, UCLA HCE); the exception is Johns Hopkins SAFE Desktop, a cloud-based Windows desktop that is free to 100 GB. Hopper and SRDC do not state their hosting. My guess is that the marginal cost to those institutions is already sunk, but the pages don't say so. Most peers on public cloud (Nero, Duke SoM Azure, UCSF AWS SEC, Sherlock) pass costs through, and Stanford softens the bill with a negotiated discount and waived transfer fees. For an environment on Google Cloud, pass-through is the closer precedent.

2.5 Data classes, and where the CUI line falls

Most peers bundle several regimes into one service. Yale Hopper accepts ePHI, NIH controlled-access data and CUI [19]. Michigan SES is suitable up to "high," including CUI [21]. Penn's enclave lists ePHI, NIH controlled-access genomic data, CUI and its "High" class [22]. Berkeley's SRDC lists PHI and CUI [28]. Stanford Nero states it is suitable for NIH controlled-access data [15], and NYU's SRDE intends to meet NIST SP 800-171 [24]. Bundling lets one operator build one control set and one intake path. The cost is that the control set must be the strictest one, and the service inherits the audit burden of the strictest regime.

Not every peer bundles. Harvard's FASSE is rated for Level 3 and its page says in plain terms that it is not rated for Level 4 and that the research computing group does not provide a Level 4 environment, directing researchers to the university-level research computing office [18]. WashU lists a separate "NIST 800-171 Access-Controlled Environment" for NIH controlled-access data [39], and Purdue's Weber cluster is described as the specialty system for export-controlled data or data under NIST SP 800-171 [40]. Duke's pages separate a protected network for non-patient data (which names 800-171) from an Azure subscription and a compute cluster for patient data [20]. Both patterns exist at peers: a combined service, and a split in which a distinct service carries the 800-171 workload.

NIH controlled-access genomic data has needed NIST SP 800-171 since 25 January 2025, and approved users must attest that their institution complies [12]. We route that data, and other CUI, to UCR's separate CUI enclave, the sibling service described in Chapter 1. Peers that bundled regimes show why the routing line has operational weight: a researcher with both a HIPAA dataset and a dbGaP dataset needs two homes at UCR, and intake must say so.

UC campuses present a mixed picture. UCLA's HCE targets P4 and 800-171 together and is on-premises [30]. UCSF states that RAE and its AWS environment can both hold P1 to P4 data [13]. Berkeley's SRDC is explicitly built for P4 [28]. UC Irvine's information security page maps data types to environments by protection level, and it marks Sherlock (the UC San Diego service) as an acceptable environment for P4 items such as Social Security numbers and IRB-classified P4 research data, and for student education records (which it classes as P3), while directing PHI to the campus health security team [41]. The same page lists a campus Secure Research Environment of UCI's own, which it marks as a consult-first option for those items. The only UC page I read that mentions Google Cloud for this purpose is Sherlock's list of supported clouds. No UC campus page describes a shared Google Cloud project for P3 and P4 data, so UCR would not be following a UC precedent there.

2.6 UK and European practice as a lens

UK and European practice supplies the vocabulary and the written standards that US campus pages mostly lack. I read the Zenodo and arXiv records at abstract level, and the documentation pages in full where I cite

them.

2.6.1 The Five Safes and SATRE

The Five Safes framework organizes disclosure risk into five dimensions: safe people, safe projects, safe settings, safe data and safe outputs [42]. Mapped onto the plan, safe people are the signed DSP and training gate, safe projects are the intake review, safe settings are the no-internet project with Identity-Aware Proxy access, safe data is the minimum-necessary and classification step, and safe outputs are the data steward and release record. The mapping is my own and is useful mainly as a checklist for what an intake form should ask.

SATRE, the Standard Architecture for Trusted Research Environments, is a UK community specification developed with more than 60 organizations and built on four principles: usability, maintaining public trust, observability and standardization [43]. The DARE UK description says version 1.0 compiled best practice for TRE operations, that organizations in academia, the NHS and industry use it to assess their environments, that it is part of European efforts for the European Health Data Space, and that version 2.0 adds federation and control of AI and machine-learning outputs [44]. I did not score the plan against SATRE's full capability list, and I propose that scoring as a pilot task: SATRE is the only written specification I found that I could hand to an auditor from outside UCR.

2.6.2 Reference environments

Table 2.3 summarizes what the literature records for each non-US environment. The Turing Data Safe Haven is the nearest public analogue to the plan's engineering approach: it is described as an open-source, reproducibly deployable, cloud-based TRE built from infrastructure-as-code tools, with a shared model for sensitivity tiers and a default set of technical measures for each tier [45, 46]. Its documentation states that access requires multi-factor authentication, that researchers can read data only from inside the environment, and that outbound internet access is blocked at the higher tiers; the deployment uses Azure [47]. TREEHOOSE, a Dundee project, offers an open-source infrastructure-as-code TRE toolkit for public cloud, with the argument that public cloud removes capital cost and reduces maintenance [48]. Our literature screen labeled TREEHOOSE as AWS-based; the record's abstract does not name a provider, so I do not repeat that.

SANE and KeyPoint bear most directly on design choices. SANE's two variants exist because some data providers will not let a researcher see the data at all, and the plan's `gui_only` mode is a partial answer to the same pressure for studies whose DUA forbids copy-out [49]. KeyPoint's abstract describes enforcing strict role and project separation at the virtual desktop layer to remove "inadvertent opportunities for data linkage or masquerading data ingress and data egress via different projects" [52]. That sentence is the best published statement of the risk in a shared environment where one person works on several studies, and the plan answers it with one VM and one group per study rather than with a persona mechanism. Chapter 6 takes up that choice. The Minnesota precedent [53] is useful for a different reason: it describes a university shared controlled-access cloud that grew out of an HPC center's needs for on-demand, long-running and container-based work, and its security design was written to satisfy NIH controlled-access requirements of 2018, which has since tightened to 800-171.

Finally, UK secure data facility practice treats approved and trained researchers, time-limited projects and statistical disclosure control checks before release as the baseline [54]. The plan carries all three: training gate, an end date on each study that drives teardown, and a data steward with a release record.

Table 2.3: Reference environments and standards from the UK, Europe, Australia and one US university, as recorded in the cited abstracts or documentation pages. The records say nothing about operating experience.

Environment or standard	What the record states	Relevance to the plan
Turing Data Safe Haven [45–47]	Open-source, IaC-deployed cloud TRE; sensitivity tiers with default technical measures; MFA required; data readable only inside; outbound internet blocked at higher tiers	Closest analogue to a Terraform-built per-tier environment, on Azure rather than Google Cloud
TREEHOOSE [48]	Open-source IaC toolkit for deploying a TRE in public cloud; argues cloud lowers capital and maintenance cost	Same argument as the plan's cost rationale
SURF SANE [49, 50]	Container-like analysis environment in which the data provider keeps control; *tinker* variant lets the researcher see the data, *blind* variant has the researcher submit an algorithm that the provider approves	Resembles the plan's shell and `gui_only` modes in intent; the blind variant has no counterpart
KeyPoint (Australia) [51, 52]	One national TRE on OpenStack, with institutional identity federation and personas that keep strict role and project separation at the virtual desktop; Guacamole and Leostream for desktops	Evidence for a single multi-tenant platform with identity-enforced separation (decision D3)
Minnesota Stratus [53]	Locally hosted OpenStack and Ceph cloud, subscription-based, with a "walled garden" security design for NIH controlled-access data	US university precedent for a shared secure cloud; on-premises, not public cloud
SATRE [43, 44]	Specification built with 60+ organizations; v2.0 adds federation and AI/ML output control	Checklist for scoring the plan

2.7 Related frameworks, and what they leave out

Three published frameworks describe environments like ours, and each places our plan differently. As far as their published text goes, none of them addresses the findings that matter most here.

The Turing Data Safe Haven assigns each project to one of five sensitivity tiers, 0 to 4, and applies default technical controls per tier [47]. Our P3–P4 population sits at tiers 3 and 4; Turing's own documentation says its software does not currently support tier 4. On the network our plan matches tier 3: outbound internet is blocked at tiers 2 and 3. On software it does not. Turing's tier 3 allows only a pre-agreed allowlist of CRAN and PyPI packages and needs both the investigator and a referee to approve code brought in, while our live mirrors resemble its tier 2, where anything on CRAN or PyPI is available through a proxy or mirror [47]. Read that way, our software admission sits a tier below our data, which is the gap Chapter 8 addresses.

SATRE's output-management capabilities bear on our data-out findings. One requires a documented policy for outputs that cannot be checked by hand and names trained model weights as the example, because a model trained on sensitive records may have memorized some of them [55]. Our export form has no rule for trained models (Chapter 7), so SATRE would have flagged that gap. The Five Safes (above) work as an intake checklist but say nothing about how a setting is built.

NIST SP 800-223 gives a reference architecture for high-performance computing security that divides a system into four function zones: access, management, high-performance computing (compute) and data storage [56]. Our plan has two projects, not four zones. The secure project holds the access path, the compute and the study buckets, and the log project holds the audit evidence and Terraform state. The management plane, meaning the administrators and the deployment identity, acts directly on the secure project. The 800-223 vocabulary makes that visible: in our design, management and compute share one boundary, and only the evidence is split out. The publication is written for clusters with login nodes and parallel file systems, so the mapping is loose.

As far as their published text goes, none of the three frameworks I read addresses vendor-default verification or admin-plane isolation in a shared cloud project, the class of defect that drove most of our version 4 list. They do not tell you which API names a cloud perimeter must list, which application options only a paid edition honors, how long a storage service keeps deleted objects, or whether a deployment identity in a shared cloud project can act as any tenant's identity. Those answers live in vendor and upstream documentation, and in the cloud provider's identity model, which a framework written at the level of zones, tiers or capabilities does not reach (Chapter 14).

2.8 Where the AI-assisted research reports and the institutions' pages disagree

The reports often describe institutions as earlier years' pages or third-party summaries did, and the current pages differ on cost, scope and system names. Table 2.4 lists the differences, which bear on how far to trust the profile details I could not check.

Table 2.4: Claims in the AI-assisted research reports compared with the institutions' current pages (read 2026-11-01).

Institution	Report states	Current page states
Harvard FASSE	Built for Level 4 and 5 data (CUI, FISMA, FedRAMP, ePHI)	Level 3 only; not rated for Level 4; the group provides no Level 4 environment [18]
Stanford Nero	Standard instance about $100 per month; storage about $40 per TB per month	Example VM about $290 per month at list; object storage about $20 per TB per month; discount of at least 18% [32]
Michigan	Armis2 decommissioned and consolidated; secure enclave has a free base tier	A knowledge-base article modified 2026-06-11 still names the Armis2 and Lighthouse clusters; SES is priced by cost recovery [21, 57]
Yale Hopper	Optional paid priority tier available now	"In the coming months" priority partitions will be added [19]
Duke	PRCC replaces PACE; RAPID VMs for protected data	PRCC is listed as a fee-based environment for identifiable patient data; a "PACE" request form appears only for a separate population-health data service; RAPID VMs are described as general research-computing services [20, 58]
Yale SRDE	Archival name for decommissioned systems	Not found on Yale's page; an NYU service carries the name SRDE [19, 24]

The Michigan row is the least certain: the article lists the two clusters on a help page for cluster login, which shows they are referenced in current documentation, and does not by itself prove the systems are in production. All I can say is that the report's claim of decommissioning is not confirmed. The Yale row records that the name SRDE belongs to NYU's service in current pages, so any report text that attaches it to Yale should be discarded. Everything else in the reports' peer profiles (core counts, petabyte figures, user counts, 5 GB export directories, Utah's block-rate prices, Penn's facility specifications) is unverified and I do not use it.

2.9 How the plan compares

Peers support the plan's main shape. A central group runs the platform while a privacy or security office approves projects, an intake gate comes before data, Utah separates projects by group, network and access control on shared hardware, and most cloud-hosted offerings pass costs through. Four questions stay open. Precedent on Google Cloud is thin: one documented peer. Peers offer Windows desktops and SAS or Stata where researchers need them, and our plan mentions neither. Most peers with a subsidy run campus-hosted

systems, and the one cloud-hosted subsidy I found (Johns Hopkins SAFE) is capped at 100 GB. And we name a data steward per study, as NYU does, with no fallback for a study that has no qualified candidate; Columbia uses a central pool.

No peer page I read mentions an external security review of its own environment; our plan has had two internal model panels. Peer pages describe service, not assurance, so this is a limit on my evidence, not a claim that peer environments lack audits.

2.10 Design considerations

1. **Describe the gate as a first-class design element.** Peers put a human review before project creation. The plan's intake checklist should say what the reviewer decides (classification, agreement terms, routing to the CUI service, FERPA exception where it applies) and who signs, so that the gate is auditable.
2. **Score the plan against SATRE during the pilot.** SATRE is the only written specification I found that an outside assessor could use. Treat the score as evidence for the campus security office and for data providers, and record the capabilities the plan declines (for example federation and the blind-analysis mode).
3. **Decide the software question early.** Ask pilot researchers whether they need SAS, Stata, SPSS or a Windows desktop. Peers with clinical and social-science users offer them. If the answer is yes, decide whether the plan adds a Windows image or states that such studies belong elsewhere.
4. **Resolve D8 against the pass-through precedent.** Most peers on public cloud charge the researcher. Record the funding rule in the plan, with any subsidy limited to named items such as shared platform operation, so that the per-study bill is predictable.
5. **Plan a data-steward pool.** Columbia's central DSO service addresses the case where a study team has no suitable steward. Decide whether central IT or the campus security office would provide a fallback, and what training and separation of duties apply.
6. **Write the routing line into intake.** Peers either bundle CUI and PHI or split them. UCR splits. The intake form should ask about NIH controlled-access data and CUI first, and send those studies to the sibling CUI service, so that a researcher with both kinds of data is told at the start that two homes are needed.
7. **Record peer rules the plan would enforce differently.** Yale's account-termination list includes photographing screens, Michigan withholds administrator rights from researchers, Harvard disables interactive job allocation. For each, state whether the plan matches it, and if not, why.
8. **Seek peer evidence the web cannot give.** Ask Stanford Research Computing about Nero's staffing, onboarding time and incidents, and ask UC peers how they handle P3-P4 requests that their enclaves cannot take. Public pages do not carry that information.

Chapter 3

The Rules: HIPAA, FERPA and UC IS-3

Three different kinds of rule reach a study that holds protected health information (PHI) or student education records, and they do not reach it the same way. A statute binds an entity because of what the entity is: a covered entity or business associate under the Health Insurance Portability and Accountability Act (HIPAA), or an educational agency or institution that receives federal education funds under the Family Educational Rights and Privacy Act (FERPA). A university policy binds data because of how it is classified: under the University of California policy IS-3, the same record is P3 or P4 wherever it sits. A contract binds a study because a data provider wrote terms into a data use agreement (DUA) or data sharing agreement. A shared research environment has to satisfy whichever of these applies to each study inside it, so I had to sort out which rule produces which requirement before judging the design.

The three also differ in how specific they are. The HIPAA Security Rule lists safeguards and the regulation text can be quoted. FERPA says almost nothing about technology: its regulation governs who may receive a record, for what purpose, under what written terms, and when the record must be destroyed. IS-3 sits between them and is the most operational: it gives P3 and P4 control lists that a campus security office can audit. I worked from the primary texts as retrieved from the electronic Code of Federal Regulations (eCFR, current through 2026-10-29), the Federal Register, UC and NIH. None of this is legal advice; where a question turns on how counsel or the UC privacy office reads a rule, I say so.

3.1 HIPAA

3.1.1 Who is regulated, and how HIPAA reaches UCR research

The Security Rule applies to covered entities (health plans, clearinghouses and most health care providers) and to their business associates [59]. The University of California is a hybrid entity. According to the UC Office of the President (UCOP), the Regents designated UC a HIPAA hybrid covered entity in 2002 with a single health care component (medical centers, clinics, health care providers, health plans and student health centers), and "the research function is excluded from HIPAA coverage at UC." UCOP adds that research health information not associated with a health care service is not subject to the Privacy and Security Rules, and that other state and federal laws govern it [7]. So HIPAA reaches these studies through the data source's agreement, a DUA or a BAA, and not because UCR is a covered entity for its research. For a study run by a UCR faculty member outside the health care component, HIPAA does not attach by default.

It attaches through the data source. Three routes recur. First, a covered entity may disclose PHI for research with an Institutional Review Board or privacy board waiver of authorization under 45 CFR 164.512(i), or with individual authorization under 164.508. Second, it may disclose a limited data set to a recipient that signs a data use agreement meeting 45 CFR 164.514(e)(4), which must establish the permitted uses, who may use or receive the data, and a promise not to use or further disclose the information other than as the agreement permits, to use appropriate safeguards, to report any use or disclosure the agreement does not allow, to bind any agent to the same restrictions, and not to identify the information or contact the individuals [60]. Third, a covered entity may write its own DUA with HIPAA-style terms. In each route the Security Rule's text does not directly bind a UCR researcher outside the health care component, but the contract imports the standard, usually as a promise to meet Security Rule safeguards plus a breach-notice window. The exception

is a fourth route that Chapter 1 names: if a UCR unit signs a business associate agreement with an outside covered entity, UCR is that entity's business associate for the data, and the Security Rule (164.302) and the business associate breach clock (164.410) bind it directly. Whether a given research arrangement makes UCR a business associate is a question for the UC privacy office.

The Google Cloud business associate agreement (BAA) sits awkwardly in this picture, and I can name the awkwardness but not resolve it. A business associate agreement is the written assurance a covered entity must obtain before a business associate creates, receives, maintains or transmits electronic PHI on its behalf (45 CFR 164.308(b)(1) and 164.314(a)) [61]. Our plan assumes an executed Google Cloud BAA. Whether Google is a business associate of a covered entity when the data holder is a UCR research unit outside the health care component is a question for the UC privacy office and counsel. The plan's decision D1 ("only BAA-covered services touch PHI") takes the conservative course: the BAA is the contract that gives the strongest available assurance about Google's handling, whichever way the legal role is characterized.

3.1.2 What the Security Rule requires

The rule is organized as administrative (45 CFR 164.308), physical (164.310) and technical (164.312) safeguards, plus organizational requirements for business associate contracts (164.314) and a documentation standard (164.316). Each standard has implementation specifications that are either Required or Addressable. "Addressable" does not mean optional: the covered entity or business associate must assess whether the specification is reasonable and appropriate and, if not, document why and adopt an equivalent measure. The points that bear on the plan, quoted or closely paraphrased from the eCFR [61]:

- Risk analysis and risk management are Required: "Conduct an accurate and thorough assessment of the potential risks and vulnerabilities to the confidentiality, integrity, and availability of electronic protected health information" (164.308(a)(1)(ii)(A)). Information system activity review is also Required: regularly review "audit logs, access reports, and security incident tracking reports" (164.308(a)(1)(ii)(D)).
- Technical safeguards require unique user identification (Required), emergency access procedures (Required), automatic logoff (Addressable) and encryption and decryption (Addressable) (164.312(a)). Audit controls must "record and examine activity in information systems that contain or use electronic protected health information" (164.312(b)). Transmission security includes encryption "whenever deemed appropriate" (Addressable, 164.312(e)).
- Disposal is Required: policies for "the final disposition of electronic protected health information, and/or the hardware or electronic media on which it is stored" (164.310(d)(2)(i)).
- Documentation, including each required action, activity or assessment, must be kept for 6 years from creation or last effective date, whichever is later (164.316(b)(2)(i)).

I read two of these provisions in ways that affect the plan. The 6-year period governs documentation the subpart requires, such as policies and risk analyses. Whether raw audit logs count as such documentation is arguable, and the Security Rule text sets no log retention period; our proposal to keep logs for 6 years (Chapter 10) is a policy choice for the campus security office and records management, not something the regulation text dictates. Decision D4 is not a HIPAA problem under either the current rule or the proposal: data is still encrypted at rest by Google's default encryption, and neither text requires customer-managed keys. Key custody is an IS-3 and contract question, taken up below.

3.1.3 NIST SP 800-66 Revision 2

NIST published SP 800-66 Revision 2, *Implementing the HIPAA Security Rule: A Cybersecurity Resource Guide*, in final form on 2024-02-14 [59]. It is guidance, not regulation. It describes itself as practical guidance for regulated entities of any size, and it maps Security Rule standards to NIST Cybersecurity Framework subcategories and to SP 800-53 Revision 5 controls. That mapping is what we can use: UC's IS-3 control list and Google's published control mappings can both be read against the same SP 800-53 vocabulary, which gives the campus security office a way to show that an IS-3 evidence set also addresses a Security Rule standard. The plan does not currently carry such a crosswalk.

3.1.4 The 2024 proposed Security Rule: status

The Office for Civil Rights (OCR) at HHS issued a notice of proposed rulemaking (NPRM) on 2024-12-27, published in the Federal Register on 2025-01-06 as 90 FR 898 (RIN 0945-AA22), with comments due 2025-03-07 [8, 62]. HHS's fact sheet lists the proposals the plan cares about: remove the distinction between required and addressable specifications, with limited exceptions; require a technology asset inventory and a network map reviewed at least every 12 months; require multi-factor authentication with limited exceptions; vulnerability scanning at least every six months and penetration testing at least every 12 months; network segmentation; written procedures to restore certain systems and data within 72 hours; a compliance audit at least every 12 months; and annual verification by business associates of their technical safeguards [62].

As of 2026-11-01 the rule is proposed, not final, and three sources agree. The eCFR text of 45 CFR 164.312, current through 2026-10-29, still labels automatic logoff and encryption as Addressable [61]. The reginfo.gov Unified Agenda entry for RIN 0945-AA22 lists the NPRM as published 2025-01-06, places the rule in "Long-Term Actions," and gives a final action date of 07/00/2027 [9]. A July 2026 law-firm analysis (a practitioner source) reports that HHS moved the final date from May 2026 to July 2027 and reads the new date as a sign that HHS still intends to finalize at least some of the proposed amendments, while calling all such timelines aspirational [63]. A Federal Register search for RIN 0945-AA22 on 1 November 2026 returned the 2025 proposed rule and no final rule.

The 2024 HIPAA Security Rule rewrite is still a proposal, so we should build toward it without citing it as law

The current Security Rule keeps Addressable specifications, including encryption. The NPRM would make them Required and add MFA, segmentation, 72-hour restore, and annual audit and testing. July 2027 is a projection, and nobody knows how much of the proposal will be finalized. Every NPRM item above should already be met or have a recorded gap, which is not a large ask, because IS-3 already requires most of them at P3 and above (Figure 3.2).

3.1.5 Breach notification

Under 45 CFR 164.404(b), a covered entity must notify affected individuals "without unreasonable delay and in no case later than 60 calendar days after discovery of a breach" of unsecured PHI. The duty is to notify without unreasonable delay; 60 days is the outer limit, not a grace period. A breach is treated as discovered on the first day it is known, or by exercising reasonable diligence would have been known, to any workforce member or agent other than the person committing it (164.404(a)(2)). A business associate must notify the covered entity on the same outer limit of 60 calendar days (164.410(b)) [61]. For UCR, the 164.410 clock applies only where UCR is a business associate; a limited data set recipient's reporting duty and clock come from its data use agreement. Data use agreements commonly set shorter windows. The NIH controlled-access program, for example, requires notice to the NIH Incident Response Team and related offices within 24 hours

of when an incident is identified [64] (that data routes to UCR's separate CUI enclave; it is cited here only as an example of a short contractual clock). Because a DUA clock can run out long before the 60-day outer limit, the plan's requirement that an incident and breach matrix be signed before any PHI arrives (Section 3.7.2) matters more than the HIPAA number.

California law adds a second clock that is not specific to health data. Civil Code section 1798.82(a)(2), as published on the Legislature's site, requires disclosure to a California resident "within 30 calendar days of discovery or notification of the data breach," with allowance for law enforcement needs and for determining the scope of a breach [65]. That section is written for an "individual or business" that owns or licenses computerized personal information. Section 1798.29, the parallel section for state agencies, is the obvious candidate for a UC campus, and the UC classification guide names both [6]. I read subdivision (a) of 1798.29, which requires disclosure "in the most expedient time possible and without unreasonable delay" and gives no day count [66]; I did not read its definitions, and Chapter 10 quotes the same text. Whether UC counts as an "agency" under 1798.29, and so which California clock applies to a campus, is a question for counsel.

3.2 FERPA

3.2.1 What FERPA protects and what it omits

"Education records" are records that are "directly related to a student" and "maintained by an educational agency or institution or by a party acting for the agency or institution" (34 CFR 99.3) [67]. The default is that personally identifiable information (PII) from an education record may not be disclosed without the consent required by 99.30. Section 99.31(a) lists the conditions under which consent is not needed. FERPA has no security rule. The Department of Education's student privacy office states that FERPA "does not require educational institutions to adopt specific security controls" while advising institutions to "take appropriate steps to safeguard student records" [68]. A FERPA environment is therefore specified by the exception that authorizes the disclosure, the written agreement that exception requires, and whatever technical terms the disclosing party adds. IS-3 supplies the technical floor on the UC side.

3.2.2 The routes that do not need consent

Four provisions matter for research computing. I quote them from the eCFR as current on 2026-10-29 [67].

School officials, 99.31(a)(1). Disclosure is allowed to other school officials, including teachers, "whom the agency or institution has determined to have legitimate educational interests." An outside party can be a school official if the institution has outsourced an institutional service or function to it, and the outside party "(1) Performs an institutional service or function for which the agency or institution would otherwise use employees; (2) Is under the direct control of the agency or institution with respect to the use and maintenance of education records; and (3) Is subject to the requirements of Sec. 99.33(a) governing the use and redisclosure of personally identifiable information from education records." This is the route by which a cloud provider can handle student records for the institution. The institution "must use reasonable methods to ensure that school officials obtain access to only those education records in which they have legitimate educational interests" (99.31(a)(1)(ii)); an institution that uses no physical or technological access controls must ensure that its administrative policy is effective. For our plan, the second sentence is the one that helps: per-study groups, per-study buckets and the logged access path are the kind of technological controls the provision contemplates.

Studies, 99.31(a)(6). Disclosure is allowed "to organizations conducting studies for, or on behalf of, educational agencies or institutions to: (A) Develop, validate, or administer predictive tests; (B) Administer student aid programs; or (C) Improve instruction." The disclosure may be made only if three conditions hold (99.31(a)(6)(iii)). The study "is conducted in a manner that does not permit personal identification of parents and students by individuals other than representatives of the organization that have legitimate interests in the information"; the information "is destroyed when no longer needed for the purposes for which the study was conducted"; and the disclosing agency enters into a written agreement with the organization that specifies the purpose, scope and duration of the study and the information to be disclosed; requires use of the PII only for the stated purposes; requires conduct of the study without personal identification by anyone other than representatives of the organization with legitimate interests; and requires destruction of all PII when no longer needed and "specifies the time period in which the information must be destroyed." The regulation also says the agency is not required to initiate a study or endorse its conclusions (99.31(a)(6)(iv)), and that "organization" includes Federal, State and local agencies and independent organizations (99.31(a)(6)(v)).

Audit and evaluation, 99.31(a)(3) and 99.35. Authorized representatives of the Comptroller General, the Attorney General, the Secretary of Education and State and local educational authorities may receive PII for an audit or evaluation of Federal or State supported education programs. The State or local authority must designate a non-employee representative in a written agreement that designates the entity, specifies the PII, states that the purpose is an audit or evaluation, describes the activity with enough specificity to show it falls within the exception and how the PII will be used, requires destruction when no longer needed, specifies the destruction period, and establishes policies to protect the PII from further disclosure and unauthorized use (99.35(a)(3)). For a UCR team this route matters when a state agency, not a school, is the data holder.

De-identified release, 99.31(b). An institution or a party that received records under Part 99 may release records "after the removal of all personally identifiable information" if it has made a reasonable determination that a student's identity is not personally identifiable, taking into account other reasonably available information, whether through single or multiple releases. This is the route that moves a dataset out of FERPA scope, and it is the basis for any output release from the environment (Chapter 7).

Redisclosure is limited by 99.33(a): the institution may disclose PII "only on the condition that the party to whom the information is disclosed will not disclose the information to any other party without the prior consent of the parent or eligible student," and the receiving party's officers, employees and agents may use it "only for the purposes for which the disclosure was made." The disclosing institution must also keep a record of each request and disclosure with the student's records (99.32(a)).

On my reading, the FERPA studies exception covers studies for or on behalf of the educational agency, not any research a faculty member chooses to do

Section 99.31(a)(6)(i) names three purposes: predictive tests, student aid programs and improving instruction, and the study must be conducted "for, or on behalf of" the educational agency or institution. On my reading of 99.31(a)(6), the studies exception does not cover faculty-initiated research unless the institution commissions or adopts the study; the campus FERPA officer decides. IRB approval does not supply the missing authority, because the Common Rule and FERPA ask different questions. The routes that remain for such work are student consent under 99.30, de-identified data under 99.31(b), or an institutional function performed by someone who qualifies as a school official under 99.31(a)(1). The environment can enforce a written agreement's terms; it cannot make a study fit an exception.

The Department of Education and UCR's registrar have issued no interpretation I could find on this point. Whether a particular study fits is for the campus FERPA officer to determine, and intake should record the

answer instead of assuming it (Section 3.7.2). Figure 3.1 lays out the decision that intake would record.

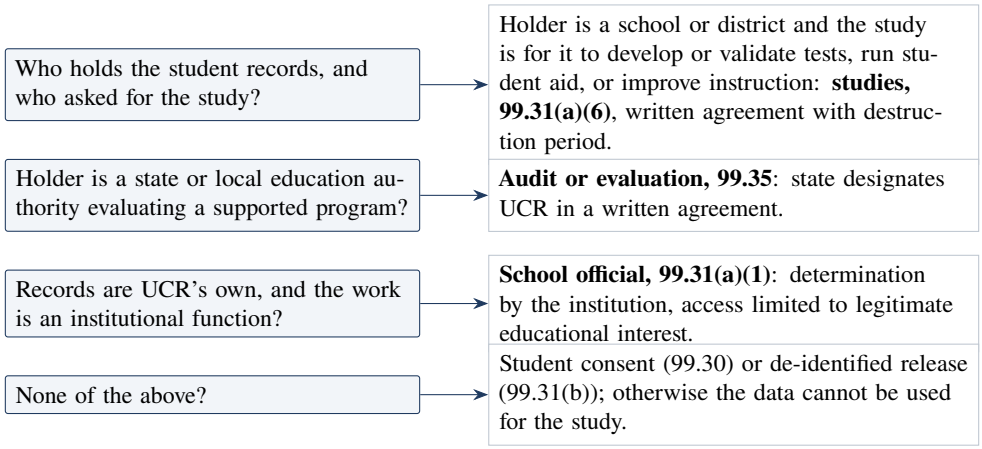


Figure 3.1: FERPA routes the intake would record for a study using student education records. Drawn from the text of 34 CFR 99.30, 99.31 and 99.35; the determination itself belongs to the campus FERPA officer.

3.2.3 Breach and enforcement

Part 99 has no breach notification section; I searched the current text of the whole part and found none. The Department of Education's data breach response checklist agrees: "FERPA itself does not contain specific breach notification requirements," and the protection comes from the prohibition on unauthorized disclosure and the recordation of disclosures [69]. Notice duties for a FERPA record therefore come from state law (the California section above) and from the agreement. Where a data set includes federal student-aid information, two more duties apply: the institution's Gramm-Leach-Bliley Act (GLBA) safeguards obligations, and its duty to report breaches of that information to the Department's Federal Student Aid office [68]. The UC guide puts GLBA-covered student financial-aid information at P4 [6], so such a study would also move up a level.

Enforcement runs through the Department. The Office of the Chief Privacy Officer investigates complaints (99.60). If an institution does not comply within the time the Department sets, the Secretary may "withhold further payments," issue a complaint to compel compliance through a cease and desist order, or terminate eligibility for funding (99.67(a)). If an outside party violates the destruction requirement of 99.31(a)(6)(iii)(B), the originating institution "may not allow the third party found to be responsible" further access to PII for at least five years (99.67(c)) [70]. The Supreme Court held in *Gonzaga University v. Doe* that the relevant FERPA provisions create no personal rights enforceable under 42 U.S.C. 1983 [71]. For a UCR study, then, the practical consequences are losing a data provider and putting institutional funding eligibility at risk, which is why providers insist on destruction evidence.

3.3 UC IS-3, P3 and P4

3.3.1 Classification

IS-3 is UC's systemwide electronic information security policy; the classification standard defines four protection levels, and UC has published a Protection Level Classification Guide that lists example information types by level [4, 6]. P4 is described as information whose unauthorized disclosure or modification "could result in significant fines, penalties, regulatory action, or civil or criminal violations," with statutory, regulatory

and contractual obligations as “major drivers” [4]. The guide, which I read in the version at the UCOP link, places these items at P4: Protected Health Information, health information, medical records and patient records; Social Security numbers; personally identifiable information “when contained in large sets and when containing a comprehensive set of information about a person”; human subject research data with individual identifiers, “particularly identifiers listed in law”; sensitive identifiable human subject research data and research covered by a Certificate of Confidentiality or the Common Rule; disability or other medical information collected from students to provide services; student financial-aid information covered by the Gramm-Leach-Bliley Act (GLBA); information with contractual requirements for P4-level protection; and research information classified as P4 by an IRB or otherwise required to be stored or processed in a high-security environment. Student education records are listed at P3 and student special services records at P3 [6].

So student education records are P3 as an information type, but a research extract reaches P4 under other entries in the same guide: when it is identifiable human subject research data (particularly with identifiers listed in law) or research under a Certificate of Confidentiality or the Common Rule, when it holds student disability or medical information or federal student-aid information, when it carries SSNs or is a large comprehensive PII set, or when its agreement requires P4 handling. The accountable data owner and the unit information security lead make the classification decision, and the campus security office determines it in the plan’s RACI. That is why I write P3-P4 throughout: the same environment would hold both, and the P3-only control differences are small enough that the plan applies the P4 set everywhere.

3.3.2 The control list

I used a working extract of the IS-3 and supporting-standard controls relevant to a P3-P4 compute environment, organized in nine groups, A to I (governance, people, access, encryption, network, operations and logging, development, suppliers, media and incident). Its text says 63 controls; the table actually contains 71 rows, a discrepancy Chapter 1 already notes, and the campus security office has not reviewed the list. Figure 3.2 counts the 71 rows by the level the extract assigns. Of the 71, 33 apply at every level or at P2 and above, 32 at P3 and above, four at P4 only, and two are HIPAA-specific supplier controls.

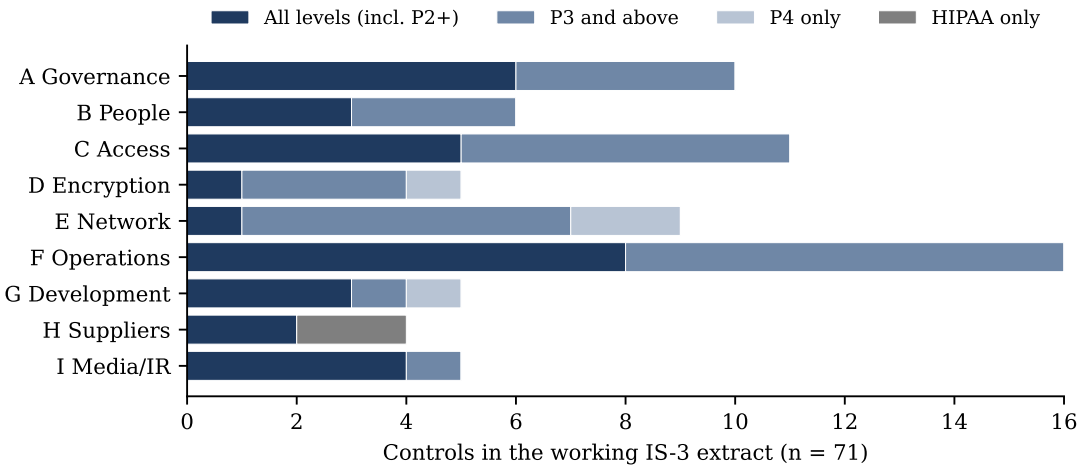


Figure 3.2: Controls in the working extract of IS-3 for a P3-P4 compute environment, by group and by the level at which each applies. Counts are of the 71 table rows in the extract, not of the 63 its text states. The extract is a working analysis by the research computing team, not a determination by the campus security office.

The four P4-only rows are encryption at rest on any electronic media (D2), routing all network access to P4 through secure access control points (E2), the stricter rules for P4 access-control devices (E5) and a penetration test at least every three years and after major change (G2). The two HIPAA-specific rows are the BAA requirement for PHI and the rule that only BAA-covered Google services touch PHI. So P3 versus P4 matters less for the build than the label suggests; it changes the encryption and pentest posture and the network access path, which the plan treats uniformly.

3.4 Data use agreements and what providers add

In a data provider's agreement the abstract rules turn into obligations someone can check. Four kinds of agreement will reach our intake, and each adds terms the environment has to honor.

HIPAA data use agreements and BAAs. The terms of a limited data set DUA are fixed in part by regulation (164.514(e)(4), summarized above). Beyond those, the covered entity commonly adds Security Rule language, a breach-notice window, restrictions on subcontractors and destruction on termination. The plan treats "no subcontractors without approval" as a reason to check Google's role and to list the sub-services in use.

NIH controlled-access genomic data. Notice NOT-OD-24-157, effective 2025-01-25, states that NIH expects Approved Users of NIH controlled-access data to maintain it on systems that meet NIST SP 800-171, an expectation written into new and renewed Data Use Certifications; users attest that their institution is compliant, and a user who chooses a third-party IT system or cloud service provider must give NIH an attestation that the third-party system complies. Non-U.S. users who cannot attest to 800-171 may attest to ISO/IEC 27001/27002 [12]. The ISO/IEC alternative is limited to non-U.S. users in the notice's text. Incident notice to NIH is required within 24 hours of identification [64]. Data of this kind does not belong in this environment (Section 3.5).

CMS and other federal health data. Federal health data agreements, such as those of the Centers for Medicare & Medicaid Services, can restrict commingling of data across agreements and set fixed destruction deadlines. I did not verify specific terms against CMS documents here and treat them as leads (Chapter 6 reads the CMS agreement). The design question is real regardless: if a provider's DUA forbids a shared boundary, a shared project fails that study. Chapter 6 takes up that decision.

State education agency agreements. State agreements can add terms such as named-user lists, U.S.-only storage, encryption, destruction certificates, short breach windows and bans on training commercial AI models on student data. These are terms to expect; I did not read them from actual agreements. The 99.35(a)(3) and 99.31(a)(6)(iii)(C) agreement elements above are the part that is regulation.

3.5 Out of scope: CUI and NIST SP 800-171

Controlled Unclassified Information (CUI) and any data whose legal or contractual terms require NIST SP 800-171 or Cybersecurity Maturity Model Certification are outside this environment. UCR already runs a separate CUI enclave for that data, scoped to NIST SP 800-171 and CMMC Level 2 [11]. NIH controlled-access genomic data governed by NOT-OD-24-157 routes there. UCR's separate CUI enclave is the sibling service and the routing target; the two services have different scopes, and the planned P3-P4 environment makes no claim to meet 800-171 and would not be presented to a data provider as meeting it.

The intake checklist should therefore ask one early question of every study: does any agreement, award term or data provider require NIST SP 800-171 or an attestation to it. A yes ends the intake and starts a referral.

3.6 HIPAA, FERPA and IS-3 P4 compared

Table 3.1 sets the three side by side on six points. Where a cell rests on the working extract of IS-3 rather than on the policy text itself, the extract is the source.

Table 3.1: HIPAA, FERPA and IS-3 P4 compared. Sources: eCFR text of 45 CFR 160 and 164 and 34 CFR 99 (current through 2026-10-29); UCOP IS-3 pages; working extract of IS-3 controls.

	HIPAA	FERPA	IS-3 P4 (UC)
Who is regulated	Covered entities and business associates. UC is a hybrid entity; research is outside its health care component, so HIPAA reaches studies through the DUA or BAA [7].	Educational agencies and institutions receiving federal education funds, and parties acting for them or receiving records under Part 99 [67].	All UC institutional information and IT resources by classification, regardless of the law that applies [4].
Vendor contract	Business associate agreement required (164.308(b), 164.314(a)). Limited data set recipients sign a DUA (164.514(e)).	No BAA equivalent. An outsourced party must be under direct control of the institution and bound by 99.33(a) to be a school official; studies and audit routes need written agreements.	UC Data Security Appendix (DS) or an approved equivalent for suppliers, and the UC BAA where PHI is involved [5].
Technical safeguards	Standards with Required and Addressable specifications: access control, audit controls, integrity, authentication, transmission security (164.312); encryption is Addressable today.	None specified. "Reasonable methods" so officials reach only records with a legitimate educational interest (99.31(a)(1)(ii)).	Itemized: MFA, segmentation, encryption in transit and at rest on any media, logging with a separate protected copy, hardened baselines, authenticated scans, pentest every 3 years.
Breach notification	Individuals without unreasonable delay, outer limit 60 calendar days from discovery (164.404(b)); business associate to covered entity on the same outer limit (164.410); DUAs often shorter.	No FERPA breach rule. State law and the agreement set the clock; disclosures must be recorded (99.32) [69]. GLBA and Federal Student Aid reporting where student-aid data is involved.	Prompt report to the security officer, who informs the privacy officer; DUA and BAA notice windows must be met.
Destruction	Required policies for final disposition of ePHI and media (164.310(d)(2)(i)); documentation kept 6 years (164.316).	Studies: destroy when no longer needed, with a stated period in the agreement (99.31(a)(6)(iii)). Audit route: same (99.35(a)(3)(iii)-(iv)).	Institutional Information Disposal Standard; for cloud, deletion evidence.
Enforcement	HHS Office for Civil Rights; civil money penalties under 160.404, with amounts adjusted annually in 45 CFR 102 (current figures not retrieved) [72].	Department of Education; withhold funds, cease and desist, terminate eligibility; five-year bar on a third party that breaches destruction terms (99.67). No private right enforceable under section 1983 [71].	Internal: policy exceptions only through the security officer with compensating controls; compliance review and audit; consequences of external law and contract arrive through those instruments.

3.7 Shared responsibility

3.7.1 Google as provider under the BAA

Google's own HIPAA documentation states that "complying with HIPAA is a shared responsibility between you and Google," that the customer must determine whether it is a covered entity or business associate and whether it needs a BAA, and that customers are responsible for ensuring that the environments they build "are properly configured and secured" [10]. The same page says the BAA covers Google Cloud's entire infrastructure and then lists the covered products. That list, as retrieved on 2026-11-01, includes Compute Engine, Persistent Disk, Cloud Storage, Filestore, Artifact Registry, Cloud Logging, Cloud Monitoring, Cloud DNS, Virtual Private Cloud, VPC Service Controls, Identity-Aware Proxy, Cloud Key Management Service, Secret Manager and Security Command Center. It instructs customers to disable unsupported services when working with PHI and says that if a service cannot be disabled it must not be used for PHI, and not to use pre-GA offerings with PHI.

Two details on that page matter for us. The strings "OS Login" and "OS Config" do not appear on the retrieved page, although the plan's list of services for the campus security office to confirm names OS Config and OS Login. They may be covered as parts of Compute Engine, but the page I read does not say so, and the plan's request that the security office confirm the list is warranted. The page also advises against putting PHI or credentials in resource metadata, because logs might capture it, and says Artifact Registry artifact names appear in logs and are visible to users with the reader role. A naming convention that keeps study identifiers and participant-derived strings out of bucket names, VM labels and container names would follow from that advice; the plan's module uses study identifiers in resource names, and those identifiers would need to be non-identifying.

3.7.2 The matrix

Table 3.2 assigns each obligation to one of three parties: Google as provider under the BAA; the platform operators, meaning the systems team, the campus security office (with the security officer for risk acceptance and breach notification) and the research computing team as builder; and each study's principal investigator (PI) and team. It follows our draft RACI and is a proposal. The RACI is a draft awaiting signatures from the research computing team, the systems team, the campus security office, the security officer and the environment's accountable data owner, and some rows below are marked open where the plan marks them so.

FERPA needs three additions to the plan

Our plan names HIPAA and does not name FERPA. Its decisions D1 and D2 settle the Google BAA and campus Duo MFA, and its section on agreements to be signed before any PHI includes an incident and breach matrix with notice clocks per DUA and BAA. FERPA needs three additions: an intake field for the exception that authorizes each student-records study; the destruction date taken from the written agreement, so that the teardown evidence can be matched to it; and a row for FERPA in the breach matrix, which would state the state-law clock and the agreement's clock because Part 99 has none.
Among the process items in the plan's coverage table are security-office acceptance of the shared boundary, the Duo evidence, the BAA services check, the training and background-check gate, and the pentest before first P4 data. Duo enforcement on every ucr.edu account that would use IAP is stated as a decision (D2) and as an item to confirm with the systems team; I take it as stated, with the evidence outstanding.

Table 3.2: Proposed shared-responsibility matrix for the planned environment. The plan's RACI is draft v3 and unsigned; "open" marks rows where the plan itself has no agreed owner.

Obligation	Google (provider)	Platform operators (systems, security, research computing)	Study PI and team
Physical and infrastructure security	Data centers, hardware, hypervisor, service availability.	Obtain and review assurance reports (IS-3 15.2.1; gap-analysis row H3).	None.
Contract and agreements	Offers the BAA; sets covered services.	UCR holds the BAA; security office confirms covered services; UC Appendix DS for non-BAA terms.	Reads the DUA; discloses its terms at intake.
Classification and risk assessment	None.	Security office determines level; one Risk Treatment Plan for the shared pattern, accepted by the security officer.	Supplies data description; accountable data owner for study data.
Project, network and perimeter configuration	Provides the controls (org policy, VPC, VPC-SC, IAP).	Research computing team writes, systems team applies under change ticket, security office reviews and watches perimeter dry-run.	None.
Identity and MFA	Federated sign-in works with campus identity.	The campus identity service enforces Duo on ucr.edu sign-in; evidence for IS-3 MFA is open until the systems team supplies it.	Uses own account; no sharing.
Per-study access list	None.	Systems team owns groups and applies tickets; removals within one business day.	Requests adds and removals, signs addendum, completes training, joins quarterly review.
Images, patching and baselines	Provides base images and patch services.	Systems team rebuilds and patches; research computing team maintains scripts; patch cadence open.	Requests software via research computing intake.
Logging and log review	Produces audit logs for covered services.	Security office administers the log project and reviews weekly; security officer owns retention and bucket lock.	Reports anomalies; trains on logged conduct.
Encryption	Default encryption at rest; TLS in transit on its services.	Security office recommends and the security officer accepts or rejects default keys (decision D4, pending).	None.
Data intake and agreement fit	None.	Security office checks DUA for shared environment and copy-out; research computing intake form.	Provides DUA; states exception for FERPA data (open: not in plan v3).
Export and output release	None.	Platform provides buckets, alert on read.	PI accountable; a data steward releases and records.
Incident detection and notice	Notifies customer of incidents in its scope as the BAA provides (I did not read the BAA terms).	Security office accountable for response; security officer accountable for notice to data sources; matrix to be signed before PHI.	Reports suspected incidents at once; cooperates.
Destruction at end of study	Deletes resources on request.	Systems team tears down with evidence; security office signs off.	Confirms completion; meets the DUA or FERPA period.
FERPA exception determination	None.	Security office and FERPA officer advise; no field in plan v3 (open).	Identifies exception and agreement owner.
Funding	Bills the billing account.	Accountable data owner of the environment; subsidy question open (Chapter 12).	Pays through its own chart of accounts unless a decision states otherwise.

3.8 Design considerations

1. **Add FERPA to the plan's scope and intake.** Capture, per student-records study, the authorizing exception (school official, studies, audit or evaluation, de-identified, consent), the written agreement, its named data fields, and the destruction deadline. Ask the FERPA officer to approve the determination, and refuse intake of a study that cannot name one.
2. **Make the study-fit test explicit.** State in the intake guide that the studies exception covers studies for or on behalf of the agency, and that IRB approval is a separate requirement. This protects both the PI and the institution from a disclosure that never had authority.
3. **Ask counsel and the UC privacy office two questions before any PHI.** Is Google a business associate for research data held outside the UC health care component, and which California breach statute applies to a UC campus acting as the data holder? The plan relies on the BAA either way, and the answer shapes who notifies whom.
4. **Sign the incident and breach matrix with real clocks.** Include HIPAA's "without unreasonable delay" duty and its 60-day outer limit, the California section counsel says applies (1798.82's 30 days or 1798.29's "most expedient time possible"), the 164.410 clock where UCR is a business associate, any NIH or other federal agreement clock for data that is not controlled-access genomic data (that data goes to UCR's separate CUI enclave), and each provider's DUA window. Use the shortest as the operating clock for each study and record who calls whom.
5. **Build to the NPRM, cite it as a proposal.** Keep a tracker of the NPRM items (asset inventory and network map, MFA, 6-month scans, annual pentest, 72-hour restore, annual audit) against the plan, and re-check the Unified Agenda entry for RIN 0945-AA22 at each review. Do not describe any of them as current law in a data security plan.
6. **Confirm Google services against the BAA list in writing.** Include Compute Engine's OS Login and OS Config features, Filestore, Pub/Sub, VPC Service Controls and anything the pilot adds, and re-confirm at each pilot gate because the list changes.
7. **Keep identifiers out of resource metadata.** Fix a study naming convention that cannot carry participant-derived or sensitive strings into bucket names, labels, container names or alert text, since logs may record them.
8. **Treat the routing question as the first question at intake.** Any required attestation to NIST SP 800-171, any CUI marking and any NIH controlled-access dataset ends the intake and starts a referral to UCR's separate CUI enclave.
9. **Ask the campus security office to review the 71-row control list and the NIST SP 800-66r2 mapping.** A reviewed, renumbered list and a one-page crosswalk from IS-3 controls to Security Rule standards would let one evidence set serve both.
10. **Decide what the plan does when a provider's DUA forbids a shared boundary.** Name the clause that triggers a per-study project in advance (Chapter 6), rather than discovering it when the first CMS-type agreement arrives.

Chapter 4

Google Cloud Building Blocks

The planned environment is assembled from a short list of Google Cloud products, and almost all of its security properties come from how those products are combined rather than from any one of them. A project is the unit of billing, quota, IAM and audit. An organization policy is a constraint that holds even when a template is wrong. A VPC with no default route removes the internet from the VM's world, and a private DNS zone points Google's API names at an address range that only reaches services a perimeter can police. Identity-Aware Proxy (IAP) and OS Login replace the open SSH port and the key file. Buckets, log buckets and Artifact Registry repositories carry the data, the evidence and the software. For each block I note what Google documents, which parts of our Terraform use it, whether Google lists it under the Business Associate Agreement (BAA), and the documented limits the pilot will run into.

I took product behavior from Google's own documentation, read on 2026-11-01. Those pages change, so each claim names the page it rests on. Because the Terraform has never been applied, "the plan uses" a block means it is written to. Where a documented limit and a plan choice might collide, I leave it as a question for the pilot. Where Google's page and the Terraform settle the point on reading, I call it a defect in plan v3.

4.1 The blocks in one view

Figure 4.1 stacks the blocks as the Terraform declares them. The top layer is the resource hierarchy: the plan puts its two projects in a dedicated folder in the campus resource hierarchy, and the VPC Service Controls perimeter belongs to an organization-level access policy, which is why the foundation Terraform leaves it out until an access policy identifier is supplied. The offline foundation plan contains 100 resources and none of them is a perimeter. The second layer is the two projects. The third holds the project-level guardrails, the network and the software mirrors in the secure project, and the audit bucket, state bucket and billing dataset in the log project. The bottom layer is the per-study module: one VM with a data disk, three buckets, one service account, one snapshot schedule and their IAM bindings (19 or 20 resources).

Tables 4.1 and 4.2 map each block to what it does, how the plan uses it, whether Google's HIPAA page lists it, and the main gotcha. The BAA column records a lookup against Google's list of supported products (page text dated 2025-09-17 for the Google Cloud list, retrieved 2026-11-01) [73]. "Listed" means the product name appears there. "Not listed" means the name does not appear; it does not mean Google has said the product is prohibited, and Section 4.9 explains why that difference matters for control-plane services.

4.2 Projects and organization policies

A Google Cloud project is where resources, quotas, billing links and IAM bindings live, and every audit log entry names one. The plan treats the project as the boundary for P4 work and creates two: a secure project that holds every study VM, and a small log project with no compute that the campus security office administers. Both are created with `auto_create_network = false`, so neither starts with the default network and its permissive rules.

Organization policy constraints sit above IAM. They restrict what can be configured regardless of who asks. The plan sets fourteen at the project level: ten boolean constraints, three deny-all list constraints and a

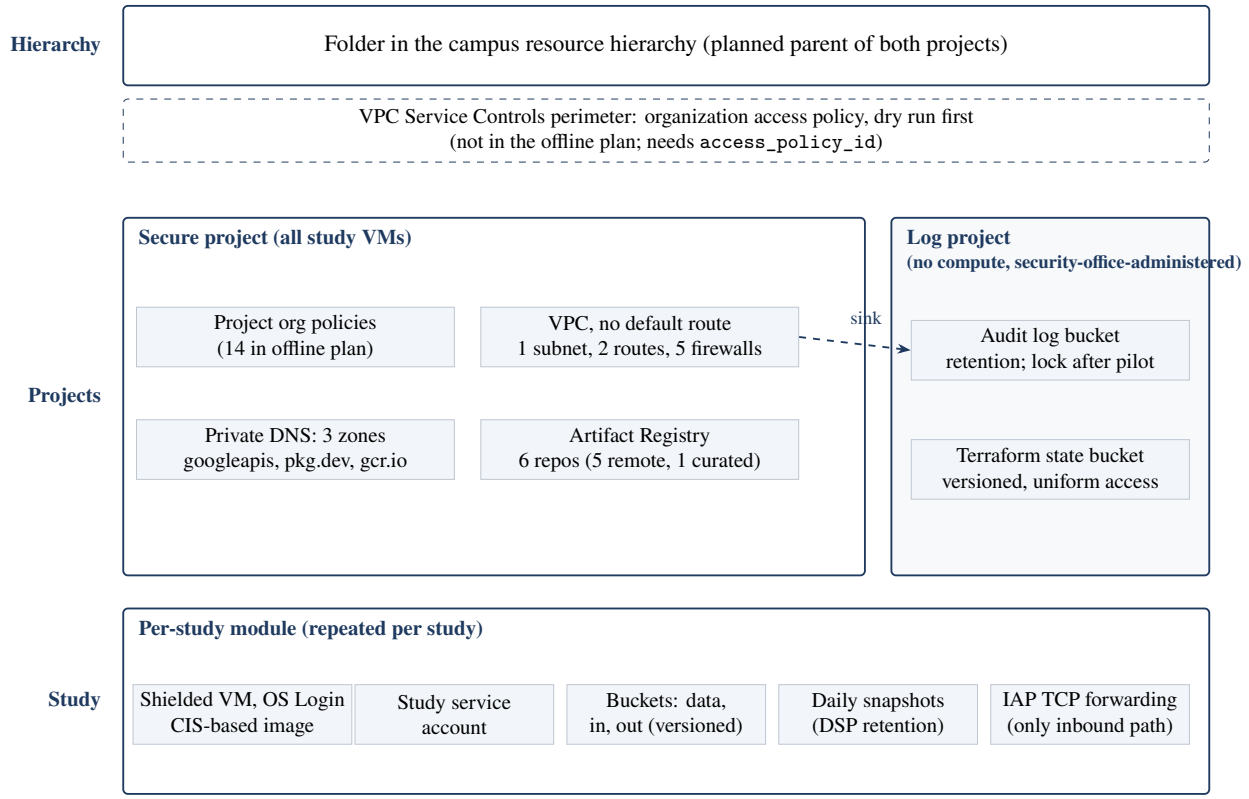


Figure 4.1: Google Cloud building blocks as the foundation and secure-VM Terraform declare them. Counts come from the offline foundation plan (100 resources). The perimeter is dashed because the offline plan creates none. The dashed arrow is the project-level log sink from the secure project to the log project's bucket.

Table 4.1: Platform and network blocks: function, planned use, BAA listing and main gotcha.

Block	What it does	How the plan uses it	BAA (checked)	Gotcha
Projects and Resource Manager	Unit of billing, quota, IAM, audit scope	Secure project and log project, `auto_create_network = false`	Listed (Cloud Resource Manager API)	Project-level policies only; org-level controls need org admin rights
Organization policy	Constraints enforced on resources in a project, folder or organization	14 project-level policies (OS Login, Shielded VM, no external IP, no SA keys, uniform bucket access, locations)	Not listed	Does not stop later changes by an admin who holds the right IAM role (see Shielded VM)
VPC, routes, firewall	Private network; ingress and egress rules with logging	Own VPC, default route deleted, IAP-only ingress, egress only to the restricted VIP	Listed (VPC)	Connection tracking expires after 10 minutes without a packet
Private Google Access, restricted VIP	VMs without external IPs reach Google APIs on `199.36.153.4/30`	Route and egress rule to the /30; private DNS maps names to it	Listed (VPC)	Only VPC-SC-supported APIs; only HTTP-based protocols; Google's table lists a second range for direct connectivity, which routes must also cover
Cloud DNS private zones	Resolves names inside the VPC	Zones for `googleapis.com`, `pkg.dev`, `gcr.io`	Listed	Other domains Google lists (gstatic, pki.goog, run.app) have no zone in the plan
VPC Service Controls	Perimeter around Google API calls to named services	Dry-run perimeter, 11 restricted services, two ingress rules	Listed	Protects API calls, not SSH or network paths; admin APIs and consoles break; needs org rights
Cloud Logging	Stores and routes logs; log buckets with retention and lock	Data Access logs on all services; project sink to a locked-after-pilot bucket in the log project	Listed	Locking is irreversible; logs can carry resource metadata
Cloud Monitoring	Metrics and alert policies	9 alert policies, 3 notification channels	Listed	A channel pointing outside the organization is a documented perimeter exfiltration path

Table 4.2: Access, software, storage and governance blocks: function, planned use, BAA listing and main gotcha.

Block	What it does	How the plan uses it	BAA (checked)	Gotcha
Identity-Aware Proxy (TCP forwarding)	Authenticated tunnel to a VM port without a public IP, from `35.235.240.0/20`	Only inbound path; SSH carries RStudio and RDP forwards	Listed (IAP)	Not for bulk transfer; one-hour inactivity disconnect
OS Login	IAM-controlled Linux accounts and SSH access	Required by policy; metadata SSH keys blocked	Not listed by name	Its 2FA uses Google 2-Step Verification, separate from campus SSO MFA
Shielded VM	Secure Boot, vTPM, integrity monitoring	All three enabled in the module; policy requires Shielded images	Feature of Compute Engine (listed)	Secure Boot can be disabled after creation despite the policy
Compute Engine, Persistent Disk	VMs, disks, snapshots, images	One VM per study; daily snapshots	Listed (both)	GPU VMs cannot live-migrate; images are still your patching duty
CIS Hardened Images	Vendor images configured to CIS Benchmarks, sold through partner listings	Level 1 Rocky 9 base for the CPU image	Not applicable (third-party image)	Image project path and licensing terms not yet verified
Artifact Registry remote repositories	Pull-through cache for public repositories	PyPI, Docker Hub and three YUM remotes; one standard repo for curated containers	Listed	YUM and Apt remotes are Preview; no CRAN or conda; upstream access is denied inside a perimeter by default
Cloud Storage	Buckets with IAM-only access, versioning, retention policy	Three buckets per study; uniform access, public access prevention, versioning	Listed	Uniform access locks in after 90 days; versioning and default soft delete keep deleted PHI copies
Assured Workloads	Folder-level control packages (residency, personnel access, key management)	Not used	Listed	Needs an organization; package fit for P3-P4 untested
Cloud Workstations	Managed browser or local-IDE workstations	Not used; a managed alternative to raw VMs	Listed	VPC-SC use needs a private cluster and extra APIs in the perimeter

resource-location allowlist limiting resources to the `us-central1` locations. All fifteen constraint names the Terraform uses appear in Google's constraint reference [74]. The boolean group covers OS Login, Shielded VM, serial port access, default network creation, nested virtualization, service account key creation, automatic Editor grants to default service accounts, uniform bucket-level access, public access prevention and public Cloud SQL addresses. The deny-all group blocks external VM IP addresses, VPC peering and IP forwarding. A domain-restriction policy is written but stays off until a directory customer identifier is supplied.

These policies apply to one project, because organization-level changes need org or folder administrator rights that we deliberately kept from the deployer account. A future project created outside the Terraform would not inherit them. A constraint also governs resource configuration at creation and update time, and its description can contain exceptions. Google's Shielded VM constraint requires new instances to use Shielded images with Secure Boot, vTPM and integrity monitoring, and the same entry says Secure Boot can be disabled after creation [74]. The plan gives the systems team an instance-administration role on the project, which would let an administrator change a VM's shielded configuration. The audit trail would show the change, and the plan has an alert on IAM changes and on the boot-time check, but no alert specifically on an instance's shielded configuration. I list it as a design consideration below.

4.3 A network with no internet

The plan's network is the most distinctive block. The Terraform deletes the VPC's default routes at creation (the `delete_default_routes_on_create` setting), so no route to the internet exists. The subnet has Private Google Access turned on. Two routes with the default internet gateway as next hop are added back: one to the restricted range (`199.36.153.4/30`) and one to the IAP range (`35.235.240.0/20`) so that IAP return traffic has a path. Five firewall rules follow: allow ingress on TCP 22 from the IAP range to tagged VMs, deny all other ingress, allow egress on TCP 443 to the restricted range, allow the metadata server, and deny all other egress. The two deny rules and the IAP allow rule log; the egress allow rules do not.

4.3.1 Private Google Access and the restricted VIP

Private Google Access lets a VM with no external IP address reach Google APIs. Google documents two virtual IP ranges: `private.googleapis.com` at `199.36.153.8/30` and `restricted.googleapis.com` at `199.36.153.4/30`. The restricted range reaches only APIs that VPC Service Controls supports and blocks the rest; the private range reaches most Google APIs whether or not the perimeter supports them. Google's guidance is to choose the restricted range when only supported services are needed [75]. The same page lists the constraints the plan inherits. Both ranges carry only HTTP-based protocols over TCP, so ICMP and other protocols do not work. Routes must use the default internet gateway as next hop, and Google states that such traffic stays inside Google's network despite the name. Some services offer direct connectivity that bypasses the Google Front Ends, and for those Google says routes and firewall rules must allow egress to `34.126.0.0/18`.

Google's configuration table for `restricted.googleapis.com` lists routes and egress firewall rules to both `199.36.153.4/30` and `34.126.0.0/18`, and recommends adding the second range [75]. Routes and egress rules have to cover every range Google documents for the restricted path, not only the familiar one, so I would check ours against that table before the dry run. What stays open is whether any service the plan uses relies on direct connectivity; the page read does not list them. The pilot's tests with fake data would show it, as long as a failed call is distinguishable from a perimeter denial in the logs.

4.3.2 Private DNS for `googleapis.com`, `pkg.dev` and `gcr.io`

Private Google Access also needs DNS. Google's instructions are to create a private zone for `googleapis.com` with an A record for `restricted.googleapis.com` pointing at the four addresses in the /30 and a wildcard CNAME to it. Other domains used by Google services need their own handling, and the page names `*.gcr.io`, `*.gstatic.com`, `*.pkg.dev`, `pki.goog` and `*.run.app` among them [75]. The VPC Service Controls product page makes the Artifact Registry case explicit: for repositories on the `pkg.dev` domain, DNS for `*.pkg.dev` must map to the private or restricted range [76]. The plan's network configuration creates three private zones and the records for them. It covers `googleapis.com`, `pkg.dev` and `gcr.io`, which is the Artifact Registry requirement, and leaves out the other names.

4.3.3 Firewall connection tracking

Google's firewall documentation states that response traffic is allowed while the connection-tracking state is active, and that a rule's tracking state is considered active if at least one packet is sent every 10 minutes [77]. Any long-lived session that goes quiet for ten minutes can lose its return path. For this plan the long-lived sessions are SSH tunnels carrying RStudio and RDP. The startup script sets `ClientAliveInterval 300` and `ClientAliveCountMax 3` in sshd, so the server would send a keepalive probe every five minutes, inside the ten-minute window. Two behaviors remain to test. IAP's own documentation says it disconnects sessions after one hour of inactivity and that `gcloud compute start-iap-tunnel` attempts to re-establish a dropped tunnel [78]; it does not say whether SSH keepalive packets count as activity for that timer. And an idle RStudio session behind a tunnel that silently times out is a researcher-experience problem before it is a security one (Chapter 9).

4.4 VPC Service Controls

VPC Service Controls draws a perimeter around projects and restricts the Google APIs named in it. A caller outside the perimeter cannot read a protected bucket even with valid credentials, and a caller inside cannot copy data to a bucket outside. Perimeters work on API requests. Google's OS Login entry says plainly that SSH connections to VM instances are not protected by VPC Service Controls [76], so the perimeter and the firewall cover different exits. Dry-run mode logs requests that would violate the perimeter without denying them [79], which is how the plan intends to start.

The plan's perimeter lists eleven restricted services (storage, compute, logging, monitoring, Artifact Registry, IAP, OS Login, OS Config, Pub/Sub, BigQuery and the AI Platform API) and enables the "VPC accessible services" restriction to `RESTRICTED-SERVICES`. It has two ingress policies, one for administrators and one meant to let only UCR identities reach IAP. As written, the perimeter checks identity and not network or device. No campus access level exists yet, and once one does, every ingress rule should carry it.

VPC Service Controls protects Google API calls, not network paths, so the plan needs the perimeter and the firewall together

Google's product page for VPC-SC states that SSH connections to VM instances are not protected, that hierarchical firewalls are unaffected by perimeters, and that VPC peering operations do not enforce perimeter restrictions [76]. The plan answers each gap with a separate control: deny-all egress except the restricted range, an organization policy denying peering, and IAP as the only inbound path. When I see "perimeter" on a diagram I ask which exits it covers. For API-based exfiltration (a copy to another project's bucket with valid credentials) the perimeter is the control. For a socket opened from a VM, the

firewall is.

4.4.1 Supported and unsupported products

Google publishes a supported-products page with the service name, launch status and limitations for each integration [76]. Several entries bear directly on this plan.

- *Cloud Shell* is not supported. Google says VPC-SC treats it as outside service perimeters and denies access to protected data, with an exception when a device meeting the access level starts it. Cloud Shell is on the BAA list, which shows that the list and the perimeter list are independent.
- *Compute Engine* is supported, with limits: SSH-in-browser is not supported inside the perimeter and the `gcloud` CLI is the alternative; the interactive serial console does not support the restricted VIP. The plan disables the serial console by policy.
- *Cloud Storage* is supported, but the Cloud Storage page in the Google Cloud console is not reachable if the API is protected, unless an ingress rule or access level allows the user.
- *Artifact Registry* is supported, and access to upstream sources outside a perimeter is denied by default; an administrator allows it per project and location with `gcloud artifacts vpcsc-config allow` [80]. The plan's Terraform comment records the same step and marks it for verification in dry-run.
- *Identity-Aware Proxy for TCP* is listed with separate entries for its administrative API and its tunnel usage API. Google notes that only the usage API of IAP for TCP can be protected by a perimeter, and that using IAP for TCP inside a perimeter requires DNS entries for `tunnel.cloudproxy.app` and `*.tunnel.cloudproxy.app` to point to the restricted VIP.
- *Cloud Monitoring* is supported, with a documented caveat: a notification channel pointing outside the organization can combine with custom metrics and alert policies to send data outside the perimeter.
- *Cloud Logging* is supported. Google states that project-level sinks to a protected destination need no ingress rule.
- *VM Manager* is supported, and Google names the several APIs that must all be in the perimeter to protect it fully.
- *Cloud Workstations* is supported when the Compute Engine API is also restricted, the cluster is private, and the Storage, Container Registry and Artifact Registry APIs are accessible inside the perimeter [81].

Our perimeter against Google's list: one fix now, four dry-run checks

When I read our perimeter beside this page I found one settled defect and four things for the dry run. (1) A perimeter list has to name the exact API each access path calls. Think of a product with a control-panel API and a separate API for actually using it: put only the control panel in the perimeter and day-to-day use is governed by IAM alone. Our list needs a pass against Google's names before the dry run. (2) A product protected by several APIs needs all of them in the perimeter, and ours does not yet list all of them for every product it names. (3) A log sink that crosses the perimeter edge needs an explicit egress decision; Google's page does not settle the case of a protected source writing to an unprotected destination. (4) Every ingress rule should carry the campus access level once it exists. (5) Private DNS has to cover every name a product needs inside a perimeter, which matters for tunnels only if someone opens them from inside the VPC. Items (2) to (5) belong on the plan's "verify in dry-run" list; item (1) is a correction to make before the dry run starts.

4.5 Identity-Aware Proxy, OS Login and the VM itself

4.5.1 IAP TCP forwarding

IAP TCP forwarding lets a user reach a VM port without a public IP. The VM's firewall must allow ingress from `35.235.240.0/20`, which Google says contains all addresses IAP uses for TCP forwarding, and the user needs the IAP-secured Tunnel User role (`roles/iap.tunnelResourceAccessor`) [78]. The plan grants that role to the study group and to the systems team. Under "known limitations" Google writes that TCP forwarding is not intended for bulk transfer of data and that IAP reserves the right to rate-limit users abusing it, and that sessions disconnect after one hour of inactivity. That fits what we already do: data move through buckets, not through the tunnel, and SFTP is off. A report's more specific claim of a throttle on bulk transfer is not in Google's text, which says only that IAP reserves the right to rate-limit.

4.5.2 OS Login and its second factor

OS Login ties Linux accounts and SSH access to IAM. The roles are `roles/compute.osLogin` (no sudo) and `roles/compute.osAdminLogin` (sudo). Google states that an instance using OS Login cannot use metadata-based SSH keys, because the guest agent ignores them [82]. The plan enforces OS Login by policy and also sets `block-project-ssh-keys`.

OS Login has its own two-factor option. Setting the metadata key `enable-oslogin-2fa` to `TRUE` (in addition to `enable-oslogin`) requires users to complete 2-Step Verification, and Google says users must have 2-Step Verification configured on their Google account or domain or access is denied; the page adds that 2-Step Verification is not enforced on service account users [83]. That is a Google-side factor. It is separate from the Duo prompt UCR enforces at campus single sign-on. The plan does not set `enable-oslogin-2fa`, and its Terraform comment gives the reason: UCR's MFA is Duo at the SAML sign-in that IAP already requires, and the plan marks this "verify in pilot". It was a deliberate choice, and the Data Security Plan should say so. The pages I read do not say whether Google 2-Step Verification would add a second prompt or be satisfied by the federated sign-in, so I assume neither; pilot test 1 checks for the prompt.

4.5.3 Shielded VM, images and patching

Shielded VM offers Secure Boot, a virtual TPM and integrity monitoring. Google's page describes Secure Boot as helping prevent malicious code loading early in boot, and measured boot as checking the bootloader, kernel and boot drivers [84]. The module enables all three. For the GPU variant the open question is whether our NVIDIA driver stack loads under Secure Boot. The module sets Secure Boot for every template, so a failure would show up on the first GPU pilot VM (pilot test 2), not in a plan.

For the base image the plan names a CIS Hardened Image Level 1 for Rocky Linux 9 for the CPU template. CIS lists Rocky Linux 9 Level 1 among its hardened images and says they are available through Google Cloud's marketplace, where CIS is a partner [85]. The image-build script refers to a CIS image project path and family that I did not verify, and CIS's page does not state price or licensing terms. The cost question belongs in Chapter 12; the verification belongs on the first run of the build script, which has never run. The GPU variant starts from the Rocky Linux 9 NVIDIA-optimized image family in the `rocky-linux-accelerator-cloud` project (publisher and support terms not verified) and applies OpenSCAP CIS Level 1 remediation at build time. In both cases the VM has no internet, so every package and agent is baked in at image build and patched by rebuilding, with VM Manager (OS Config) handling inventory and patch jobs against the Artifact Registry mirrors.

4.6 Artifact Registry remote repositories

Artifact Registry stores packages and container images. A *remote repository* is a pull-through cache of an upstream public source. Google lists preset upstreams for Docker (Docker Hub), Go, Maven Central, npm and Python (PyPI), and, in Preview, OS package repositories for Apt (Debian, Ubuntu) and Yum (CentOS variants, Rocky, EPEL). Custom upstream URLs are accepted for Docker, npm, Maven and Python, and must use HTTPS [86]. Google's overview also describes use cases: cached copies stay available when the upstream is down, and remote repositories used with virtual repositories can reduce dependency-confusion risk.

This list sets what the plan can mirror. The plan creates a PyPI remote, a Docker Hub remote and three Yum remotes (Rocky 9 BaseOS, Rocky 9 AppStream and EPEL 9), plus one standard repository for curated containers. It states that CRAN, Bioconductor and conda are not covered by remote repositories and handles them as containers built outside the perimeter. That matches Google's lists: none of those three upstreams appears among the presets. One configuration defect shows on reading. A remote repository's upstream path is relative to the preset base URL, so a path that repeats part of that base points at a directory that does not exist, much as `cd docs/docs` fails when you are already in `docs`. One of our mirrors has this defect, and packages the images need come from it, so images built against the mirror as written would lose them (Chapter 8). Two gotchas follow from the same page. The Yum remotes are in Preview, and Google's HIPAA guidance says not to use pre-GA offerings with PHI unless a notice says otherwise [73]; no PHI would flow through a package cache, but the campus security office may still ask. And a remote repository inside a perimeter cannot reach its upstream until an administrator allows it, which has to be done once the perimeter is enforced. Chapter 8 takes up what mirroring everything does to supply-chain risk.

4.7 Cloud Storage

Each study gets three buckets: working data, an "in" bucket for landing files, and an "out" bucket for approved exports. The module enables uniform bucket-level access, public access prevention and object versioning, and a lifecycle rule deletes noncurrent versions after a configurable number of days.

Uniform bucket-level access removes object ACLs so that IAM alone controls access. Google recommends it, and notes that it cannot be disabled once it has been active for 90 consecutive days [87]. Versioning keeps prior object generations, and Google's HIPAA page recommends enabling it to preserve archives and recover after accidental deletion [73]. The same feature keeps copies of overwritten or deleted PHI, which is why the module's lifecycle rule exists and why the teardown script deletes every object version. Bucket Lock applies a retention policy to a bucket so that objects cannot be deleted or replaced until they reach a defined age; locking a retention policy is irreversible, and until it is locked the policy can be increased, decreased or removed [88]. The study buckets do not use a retention policy, since a study's data must be deletable at the end of the agreement; the plan's irreversible lock is reserved for the audit log bucket. Within the perimeter, the Cloud Storage console page is unreachable unless an ingress rule allows it [76], so day-to-day work is by tool, not by browser. Soft delete is on by default with a seven-day retention, during which deleted or overwritten objects can be restored [89]; the Terraform sets no soft-delete policy (Chapter 7).

4.8 Cloud Logging

Cloud Logging keeps entries in log buckets, with retention that can be configured and buckets that can be locked [90]. Admin Activity audit logs are always on. The plan turns on Data Access logs (admin read, data read, data write) for all services in the secure project, and routes audit, auditd, RStudio, xrdp, syslog, VPC flow and firewall logs by a project-level sink to an audit log bucket in the log project, with 2,190 days of

retention by default (about six years; six calendar years is 2,192, Chapter 10). The log project carries no compute, and only the security office's group holds project-level roles there. Two other identities reach into it: the sink's writer identity writes log entries, and the deployer identity can write to the Terraform state storage, which also stores teardown evidence. The audit log bucket itself stays restricted to the security office; the deployer's reach into the evidence store is the deployer problem in Section 5.3.

Google's page on log buckets warns that locking is irreversible. It locks the retention policy with the bucket, and the bucket cannot be deleted until every entry has met the retention period; the page suggests adding a project lien to prevent accidental deletion of a project that holds a locked bucket [90]. Two consequences for the plan follow. A locked six-year bucket means the log project outlives the secure project by years, and nobody has yet written down who owns that project's billing and closure. And Google's HIPAA page notes that audit logs never include resource data contents or query results but might capture resource metadata, so PHI should not go into resource names, labels or alert text [73]. The plan puts a study identifier in bucket names and labels; the Data Security Plan template should say that identifiers must not be participant-derived. Chapter 10 covers retention periods and the alert set.

4.9 BAA coverage

The design assumes an executed Google Cloud BAA, and plan decision D1 says that only BAA-covered services touch PHI. The plan's IS-3 table lists the check of which services are covered as an item still to do. Google's HIPAA page lists the covered products and states that the BAA covers Google Cloud's entire infrastructure (all regions, zones, network paths and points of presence) and the products named [73]. A lookup against that list, retrieved 2026-11-01, gives the following.

- *Listed:* Compute Engine, Persistent Disk, Cloud Storage, Artifact Registry, Cloud Logging, Cloud Monitoring, Identity-Aware Proxy, VPC Service Controls, Virtual Private Cloud, Cloud DNS, Pub/Sub, Identity and Access Management, Access Context Manager, Cloud Resource Manager API, Cloud Asset Inventory, Assured Workloads, Cloud Workstations, Cloud NAT, Cloud Shell, Security Command Center, BigQuery and Cloud KMS.
- *AI products, under renamed entries:* the page no longer lists "Vertex AI" by that name. It lists AI Platform Training and Prediction, Vertex AI Workbench instances, Generative AI on Agent Platform and Gemini Enterprise Agent Platform (the page does not say how these entries map to the `aiplatform.googleapis.com` API the plan's perimeter names, so the mapping stays unverified). A check of which entry applies to a specific model or feature belongs in Chapter 13 and should be repeated before any PHI use.
- *Not found by name:* Organization Policy Service, OS Login, VM Manager (OS Config), Cloud Billing and billing budgets. Each of these is control-plane or administrative. None would carry PHI in the plan's design.

The HIPAA page's own instruction is to disable unsupported Google Cloud services when working with PHI, and if a service cannot be disabled, not to use it for PHI [73]. That reads as a data-handling rule. A service that holds no PHI, such as the organization policy engine, is a different question from one that processes it. Even so, the plan needs the campus security office and counsel to decide whether services not named on the list but used only to configure the environment are acceptable; the lookup shows the question exists and does not answer it. The page also says not to use pre-GA offerings with PHI unless a notice or the offering's terms say otherwise, which connects to the Preview status of the Yum remotes above.

Every data-bearing service the plan uses is on Google's BAA list; the services that are not named are administrative, and that distinction needs a security-office decision

Compute Engine, Persistent Disk, Cloud Storage, Artifact Registry, Cloud Logging, Cloud Monitoring, IAP, VPC Service Controls, Cloud DNS and Pub/Sub all appear on the list. OS Login, VM Manager, the organization policy service and billing budgets do not appear by name. The plan's own table leaves this check open; my lookup closes the first half. It does not close the second, which is a policy judgment.

4.10 Frameworks around the blocks

4.10.1 Assured Workloads

Assured Workloads applies predefined control packages to folders. Google lists capabilities including regional and regulatory data boundaries, personnel data access controls, encryption key management controls, compliance updates and monitoring for violations. Regulatory packages include Healthcare and Life Sciences Controls (with or without US support) for HIPAA and HITRUST, alongside FedRAMP, CJIS, IL2, IL4, IL5, ITAR and IRS Publication 1075 [91]. A separate page lists the products each control package supports [92]. Assured Workloads itself is on the BAA list. The plan does not use it. A HIPAA-oriented package is a plausible candidate for a later version, but nothing in the sources read here says what a P3-P4 project would gain over the plan's own project-level policies, and a package would add organization-level setup. The question belongs on the roadmap, not in the pilot.

4.10.2 Public landing-zone frameworks

UCR's separate CUI enclave, aimed at NIST 800-171 and CMMC Level 2, was built on Google Public Sector's published infrastructure-as-code foundation [11]. That enclave is the sibling service to which this environment routes CUI and the NIH controlled-access data that now requires 800-171 (Chapter 1). The planned P3-P4 environment is a separate design, with its own Terraform, scope and approvals.

Public landing-zone frameworks of this kind, and the Cloud Foundation Fabric code they derive from, are still a useful reference and possibly a source of code. Their typical stages (bootstrap with remote state and audit sinks; folders, projects and roles; networking; and key management with policy lockdowns) correspond to groups of files in the plan's foundation Terraform. The correspondence also exposes where designs differ: a framework built for CUI usually adds customer-managed keys for all storage, while plan decision D4 chooses Google's default encryption and records it as a security-office risk acceptance; and it usually builds hub-and-spoke networking, while the plan has one VPC with no shared network and no peering. Whether to start P3-P4 from such a framework, from Cloud Foundation Fabric, or from our hand-written Terraform is for the systems team and the campus security office to decide. I would weigh the maintenance burden and the team's existing familiarity, and ask what we lose by importing the CUI enclave's controls into a design with different needs.

4.10.3 Cloud Workstations

Cloud Workstations provides managed development environments with a browser-based IDE or a local IDE connection, defined by a workstation configuration that applies on the next start [93]. It is on the BAA list and supported by VPC Service Controls, with the conditions above: a private cluster, no public IPs, the Compute Engine API restricted alongside Workstations, and the Storage, Container Registry and Artifact Registry APIs VPC-accessible; if image streaming is used, Google's VPC-SC page for the product adds the Container

File System API to the restricted services and an egress rule, including one to Google's preconfigured-IDE image project [81]. The plan uses raw VMs because it needs RStudio Server and a Linux desktop, and because the module bakes the fail-closed login gate into the image. Workstations remains a candidate for a future code-only workflow where a managed environment is enough, and for comparison with what peers run (Chapter 2).

4.11 Documented gotchas, collected

Table 4.3 collects the documented limits our design can run into, with the source for each and what the pilot should check.

Table 4.3: Documented limits that bear on the plan, the source, and what the pilot should check.

Documented limit	Source	What to check in the pilot
Firewall tracking state active only if a packet is sent at least every 10 minutes	Firewall documentation [77]	An idle RStudio tunnel survives past 10 and 60 minutes with the sshd keepalives set
IAP disconnects after one hour of inactivity; not for bulk transfer	IAP documentation [78]	Whether keepalives count as activity; what a 2 GB upload through the tunnel does
Restricted VIP supports only VPC-SC-supported APIs and HTTP-based TCP; Google's table lists a second range for direct connectivity that routes and egress rules must also cover	Private Google Access page [75]	Cover every documented range, or show from dry-run and flow logs that no call is dropped
Other Google domains (gstatic, pki.goog, run.app) need their own DNS handling	Same page	Whether any tool in the image resolves one of them
Cloud Shell, SSH-in-browser and the Storage console page do not work inside a perimeter	VPC-SC supported-products page [76]	Researchers and admins use `gcloud` from a laptop; admin identities need an ingress path
Artifact Registry upstream access is denied inside a perimeter until allowed	Remote repository page [80]	Run the allow step at enforcement; confirm a pull after enforcement
Notification channels outside the organization are a perimeter exfiltration path	VPC-SC supported-products page	Channels limited to UCR addresses; alert text carries no PHI
Secure Boot can be disabled after creation despite the policy	Organization policy reference [74]	Alert or periodic check on shielded configuration
Locked log bucket cannot be deleted until entries age out	Logging buckets page [90]	Lien on the log project; owner of its billing and closure
Soft delete keeps deleted objects restorable for 7 days by default	Soft delete page [89]	Teardown evidence states the soft-delete setting; re-check after the window
Some products expose separate configuration and usage APIs; the perimeter must restrict the one actually used	VPC-SC supported-products page [76]	Check each restricted service name against the calls the workload makes, before the dry run
A remote repository's upstream path is relative to the preset base URL	Remote repository page [80]	Check each mirror path; first image build pulls every needed package from the mirrors

4.12 Design considerations

1. **Record the BAA lookup and decide on unlisted control-plane services.** Put my dated lookup into the plan's supplier row, and ask the campus security office and counsel whether OS Login, VM

Manager, organization policy and billing budgets, which carry no PHI, are acceptable. Repeat the lookup whenever the Google list's "last updated" date changes.

2. **Correct the settled perimeter and network defects before the dry run.** Check every restricted service name against the API each access path actually calls, and attach ingress rules to those names; cover every range Google documents for the restricted path in routes and egress rules; and write every ingress rule so that it carries the campus access level once it exists.
3. **Add the unsettled perimeter items to the dry-run checklist:** the remaining supporting APIs for patch management, egress for the log sink, and whether any service uses direct connectivity. Decide before enforcement whether UCR identities alone are acceptable or a device or network access level is required.
4. **Run pilot test 2 (Secure Boot with the GPU image) first.** A driver stack that cannot load under Secure Boot would change the module defaults for the GPU template.
5. **Alert on shielded-configuration changes,** or accept that the organization policy can be bypassed by an instance admin and document the reasoning with the audit trail as the compensating control.
6. **Document the OS Login 2FA decision** in the Data Security Plan: Duo at SAML sign-in is the factor, Google's 2-Step Verification is not required, and pilot test 1 confirms that IAP sign-in cannot be reached without the Duo prompt.
7. **Verify the CIS image path and its licensing terms** before the first image build, and carry the cost into Chapter 12.
8. **Decide the foundation lineage.** Choose between the hand-written Terraform, Cloud Foundation Fabric and a public landing-zone framework, with the systems team and the campus security office, and write down which CUI-enclave controls (CMEK, hub-and-spoke) are intentionally not carried over.
9. **Name an owner for the locked log project** (billing, lien, closure), and add an instruction that study identifiers in resource names and labels must not derive from participants.
10. **Check every mirror's upstream path** against the preset base URL and confirm a metadata fetch through each mirror before the first image build.
11. **Keep the Preview status of the Yum remotes visible** in the software chapter's risk register, and watch for general availability.

Chapter 5

Architecture of the Planned Environment

The plan specifies one shared environment for all P3-P4 studies, built from two Google Cloud projects and a small set of Terraform modules. A secure project holds every study virtual machine (VM), the network they sit in, the software mirrors and the alerts. A separate log project, administered by the campus security office, holds the audit logs, the Terraform state and the billing export. Each study receives one VM, one service account and three storage buckets from a single module call. Researchers would reach the VM through Identity-Aware Proxy (IAP) after campus single sign-on with Duo, and would work in a shell, in RStudio Server, or in a Linux desktop, all carried over the same SSH tunnel.

None of this exists as a running system. What does exist is the plan document (draft v3), Terraform that has been validated and planned offline with placeholder values and never applied, image-build scripts that have never run, and the records of two review panels that changed the design between v1 and v3. The architecture below is as specified in those artifacts. Every statement of behavior is a statement about what the design would do, and several of them are explicitly marked in the plan as untested until a pilot with fake data.

5.1 Two projects, one boundary

The plan treats the project as the boundary. A dedicated secure project, placed in a dedicated folder in the campus resource hierarchy, carries the project-level organization policies, the VPC Service Controls perimeter and the data-access audit logs, so that everything inside it is P3-P4 and the campus security office has a single audit scope. We deliberately do not reuse the campus research cluster's project. A project built for open P1-P2 use is configured for convenience, not for a P4 boundary, and we would rather leave the shared cluster as it is than retrofit it. It also gives the secure project its own billing view, so that the cost of any subsidy would be visible.

Decision D11 added the second project after the first review panel. The log project has no compute. It holds the audit log bucket (6-year retention, with an irreversible retention lock to be applied only after pilot acceptance), the versioned Terraform state bucket, teardown evidence and the billing export dataset. At project level only a security-office group has roles there; neither the research computing team nor the systems team's compute administrators do. Two other identities can reach into it: the sink's writer identity can write log entries, and the deployer identity has write rights on the Terraform state storage, which also holds teardown evidence. The audit log bucket itself is administered only by the security office. The purpose, taken from items 4.4 and 4.11 of the UC Event Logging Standard, is that no one who can administer the VMs can erase the record of what they did. For the audit logs the design meets it. For the Terraform state and the teardown evidence it does not, for the reason set out in Section 5.3.

Figure 5.1 draws the whole arrangement from the Terraform. The one-time foundation is applied by an organization or folder administrator, because it needs rights that the routine deployer identity deliberately lacks. Afterwards a deployer service account applies per-study modules. We intend the systems team to act as it only under an approved change; nothing in the code enforces that (Section 5.3).

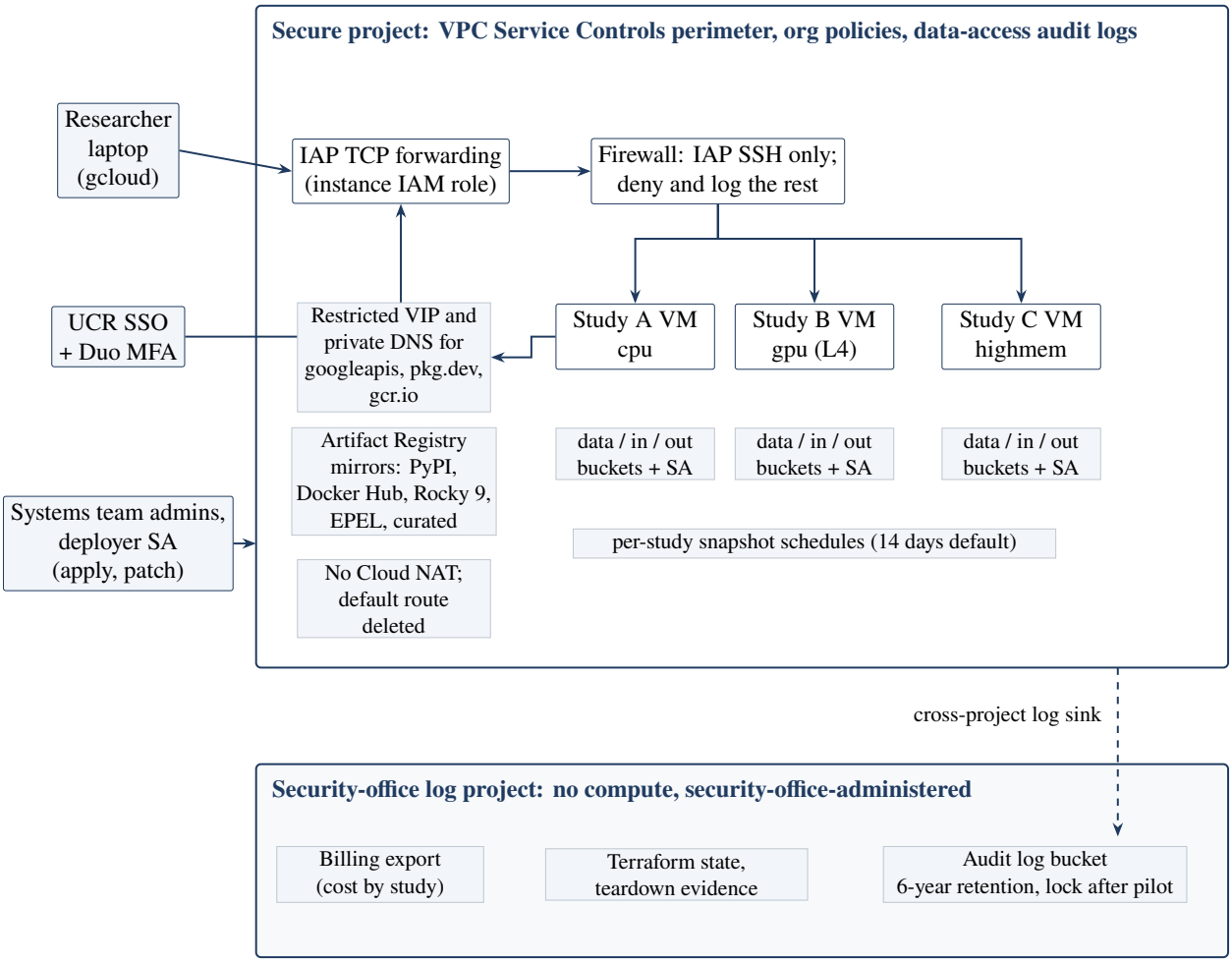


Figure 5.1: Overall architecture of the planned environment, drawn from the foundation and secure-VM Terraform. Solid arrows are access paths (administrators act on the secure project; the deployer, which administrators may act as, also has write access to the state storage in the log project); the dashed arrow is the cross-project log sink. Three study VMs are shown as in the pilot example.

5.2 The network: no way out except Google's APIs

The foundation network is a single custom-mode VPC with one subnet in us-central1, created with `delete_default_routes_on_create` so that the default internet route never exists. There is no Cloud NAT. The only routes are to two destinations: Google's restricted API range (199.36.153.4/30) and the range IAP uses for TCP forwarding (35.235.240.0/20) [94, 95]. The design intent, stated in the Terraform header, is that a VM cannot reach the internet even if a firewall rule is wrong, because no route exists to carry the packet.

Five firewall rules sit on top. One allows SSH from the IAP range to VMs tagged `p4-secure`. One allows egress on TCP 443 to the restricted range, and one allows egress to the instance metadata address. Two deny rules (ingress and egress, priority 65000) cover everything else and are logged. The ingress deny has source 0.0.0.0/0, which also covers other VMs in the subnet, so one study's VM would be unable to open a connection to another's; the first review panel had claimed otherwise and the claim was rejected after checking the rule. Whether that holds in practice is pilot test 3, which also checks that IAP still works with the default route deleted.

Name resolution is part of the same design. Three private DNS zones send `googleapis.com`, `pkg.dev` and `gcr.io` to the restricted virtual IP, so that API calls, Artifact Registry and container pulls land on the range that VPC Service Controls understands. Google documents the restricted range as reachable only for services supported by the perimeter feature and not announced to the internet [94, 96]. A practical consequence for the user experience is that firewall connection tracking stays active only while at least one packet crosses every ten minutes [97], so an idle SSH tunnel could be dropped; the sshd settings in the boot script (`ClientAliveInterval 300`, `ClientAliveCountMax 3`) address the server side, and Chapter 9 takes up the client keepalives.

5.2.1 Organization policies as a second layer

Beyond the network, the foundation enforces project-level organization policies so that a mistake in a module cannot silently weaken the project. The offline plan creates 14 of them: ten boolean constraints (OS Login required, Shielded VM required, no default network, serial ports disabled, nested virtualization disabled, no service-account key creation, no automatic Editor grant to default service accounts, uniform bucket-level access, public access prevention, and no public Cloud SQL addresses), three deny-all constraints (external IPs, VPC peering, IP forwarding) and one resource-location restriction to us-central1. The Terraform also contains a domain-restricted-sharing policy so that only `ucr.edu` identities can be granted roles, but it is created only when the directory customer identifier is supplied, and that identifier is an open item. In the offline plan with placeholder values the policy is therefore absent, which is why the plan lists 14 policies rather than 15. Figure 5.2 shows one resource block from that plan.

The VPC Service Controls perimeter shows the same pattern. The Terraform defines a perimeter that starts in dry-run mode (violations are logged and nothing is blocked), with an ingress rule intended to limit IAP to listed UCR groups (Chapter 4 checks its service names against Google's list), and enforces it only after a pilot week without unexpected violations. It also requires an organization access policy identifier and organization-level rights. Because that identifier is still an open item, the offline plan contains no perimeter resource at all. The 100 planned resources therefore describe the network, policies, mirrors, logging and alerting, and not the perimeter that the plan calls the control for data leaving through Google APIs.

```
  # google_org_policy_policy.boolean["compute.requireOsLogin"] will be created
  + resource "google_org_policy_policy" "boolean" {
      + name   = "projects/<project>/policies/compute.requireOsLogin"
      + parent = "projects/<project>"
      + spec {
          + rules {
              + enforce = "TRUE"
            }
        }
    }
...
Plan: 100 to add, 0 to change, 0 to destroy.
```

Figure 5.2: Excerpt from the offline foundation plan (with computed-at-apply attributes omitted). The project identifier is a stand-in; the plan was produced with Terraform 1.12.2 and Google provider 8.9.0 and was never applied.

The offline plan contains no perimeter and no domain restriction, so the headline count of 100 resources understates what is still to be configured

Two controls the plan treats as central, the VPC Service Controls perimeter and the `ucr.edu`-only IAM policy, are written in Terraform but are skipped in the offline plan because the organization identifiers they need are open items. A reader who counts resources as a measure of readiness would overstate how much of the design has been exercised. The remaining work includes obtaining those identifiers, running the perimeter in dry-run, and enforcing it.

5.3 Identity and access

Access is layered. Sign-in uses the campus identity provider with Duo MFA (decision D2); the Terraform enforces nothing about MFA, and the plan's own coverage table marks it as inherited from campus single sign-on with the evidence to come from a campus identity-service enforcement policy and pilot test 1. After the first panel we removed the OS Login two-factor metadata, because that feature uses Google 2-Step Verification and UCR users may not have it; the design relies on Duo at the IAP sign-in instead, and the pilot has to confirm that.

Each study has a Google group of everyone on its signed Data Security Plan (DSP), owned by the systems team. The module grants that group three things, each scoped to that study's VM or service account: `roles/compute.osLogin` on the instance (login without sudo), `roles/iap.tunnelResourceAccessor` on the instance, and `roles/iam.serviceAccountUser` on the VM's own service account. The last is required for OS Login on a VM that has a service account attached and grants the right to run as that account rather than to mint tokens for it; the first panel flagged it as high risk and the claim was rejected after checking.

Administrators form a separate group. The foundation gives it instance administration, OS admin login, IAP tunnel access and patch execution at project level, and no project-wide `serviceAccountUser`, so that an administrator would not be able to start a VM as a study's identity. The security-office group receives read-only security roles (log viewer, private log viewer, IAM security reviewer, compute viewer, inventory viewer) in the secure project and administrative roles only in the log project. The deployer undoes part of that, and this is the one place I set the problem out in full. The deployer service account holds broad project-level rights so that it can apply the per-study module, and the whole administrator group may act as it. Administrators can therefore reach study identities indirectly through the deployment identity, so "admins cannot act as a study identity" holds only by process (an approved change and a second person's review), not

by code. The same identity can write to the storage in the log project that holds Terraform state and teardown evidence. Put together, the people who run the VMs could also edit the proof that a study's data were destroyed. One identity reaches across every tenant and into the evidence store, and that is a separation-of-duties problem. I would fix it in two moves: limit who may act as the deployer to a small break-glass group, with an alert on every use, and move teardown evidence to a locked bucket that only the security office controls. The first makes the study-identity statement checkable; the second keeps the duty of running VMs apart from the duty of keeping evidence. Later chapters refer back here as "the deployer problem".

5.4 The per-study module

One call of the `secure-vm` module creates everything a study needs. In the pilot example, each call produces between 19 and 20 resources; three calls produce the 59 resources of the pilot plan. Figure 5.3 counts the planned resources by type for both offline plans.

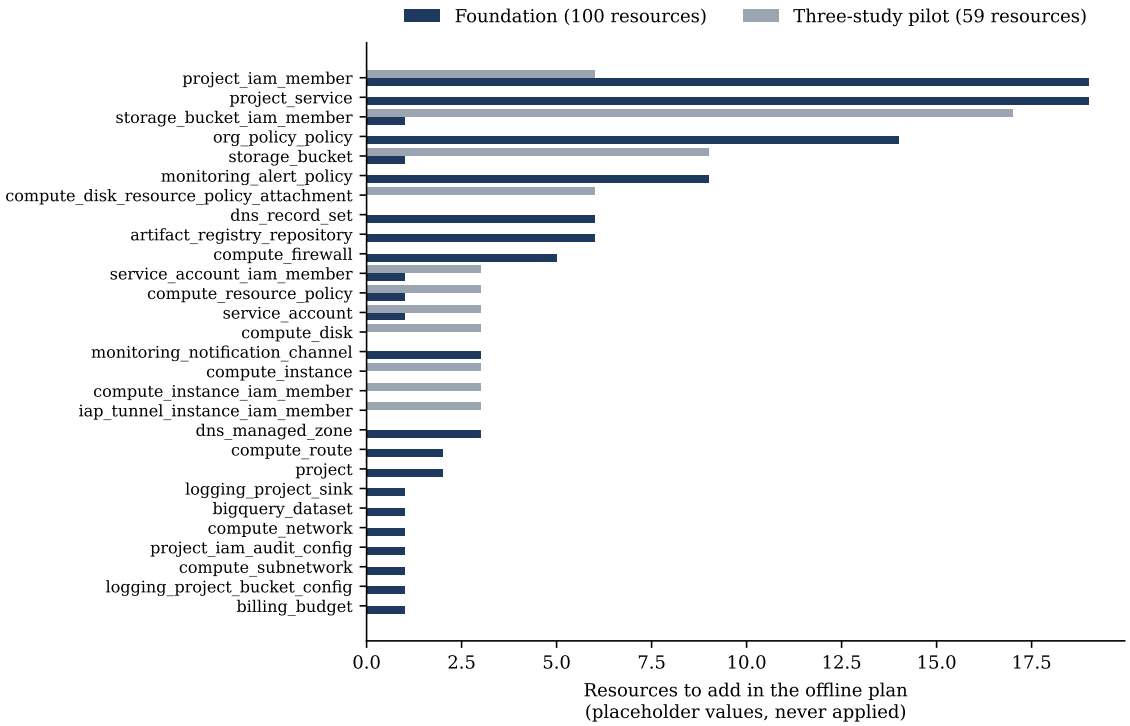


Figure 5.3: Planned resources by Terraform type, from the offline plans (foundation: 100 resources; three-study pilot: 59). Both plans used placeholder values and were never applied; every action in both is a create.

5.4.1 Compute

The VM has no external IP, Shielded VM settings (Secure Boot, vTPM, integrity monitoring), OS Login required, project SSH keys blocked, the serial port disabled and deletion protection on. It runs as a per-study service account with the `cloud-platform` scope, with actual access limited by IAM on that account rather than by OAuth scopes. Three templates (Table 5.1) set machine type and disks; every size can be overridden per study. GPU instances cannot live-migrate, so the module sets maintenance behavior to terminate for them.

Table 5.1: The three VM templates in plan section 9 and the module defaults. Disk sizes are defaults; the pilot example overrides the highmem data disk to 2000 GB.

Template	Machine	Size	Disks (boot + data)	Image family	Pilot example
cpu	n2-standard-8	8 vCPU, 32 GB	50 GB + 500 GB	p4-cpu-rocky9	R/survey, RStudio only
gpu	g2-standard-16	16 vCPU, 64 GB, NVIDIA L4	100 GB + 1 TB	p4-gpu-rocky9	imaging, 240-minute idle stop
highmem	n2-highmem-32	32 vCPU, 256 GB	50 GB + 1 TB	p4-cpu-rocky9	genomics, gui_only, desktop

5.4.2 Storage

Each study gets three buckets: `data`, `in` and `out`. All have uniform bucket-level access, enforced public-access prevention, versioning and a lifecycle rule that deletes noncurrent object versions after 30 days by default, because old versions of a file are copies of the data. The VM's service account has full object control on `data`, read-only on `in` and create-only on `out`. The study group can upload to and list `in`, so researchers can confirm what they have sent. Only a named data-steward group can read `out`, and each such read appears in the data-access logs and would raise an alert to the security office. Version 1 had given the VM service account full control on all three buckets; the first panel's finding on this is the origin of the current split. Chapter 7 examines what this arrangement does and does not prevent.

A per-study snapshot schedule (daily, retention from the DSP with a 14-day default) covers both disks, and a label carrying the DSP's end date is intended to drive the teardown reminder. The teardown script that deletes VMs, disks, snapshots, every object version, buckets and the service account, and writes an evidence record, exists in the design and has not been run.

5.4.3 Study modes

Two switches shape what a researcher can do. The `gui` list turns RStudio Server and the XFCE desktop on or off per VM. The `access_mode` value is `shell` by default or `gui_only`, which replaces any command a regular user runs over SSH with a helper that only sets a password, while still allowing the tunnel to carry browser and desktop traffic. The plan states that shell mode permits an authorized user to copy data to a laptop through SSH even with the SFTP subsystem disabled, that this is prohibited by the DSP and logged, and that the security officer would have to accept it as residual risk (D12). `gui_only` is the option for studies whose data use agreement forbids copy-out, and what remains there is screenshots and retyping.

5.5 Images and the boot sequence

The VMs have no internet at first boot, so the plan bakes everything into images instead of installing at startup. A build script runs on a temporary builder VM that does have internet, outside the secure subnet, and the resulting disk becomes an image family in the secure project. The CPU variant starts from a CIS Hardened Image Level 1 for Rocky Linux 9; the GPU variant starts from the Rocky Linux 9 NVIDIA-optimized image family (NVIDIA 580 drivers) in the `rocky-linux-accelerator-cloud` project, whose publisher and support terms were not verified, and applies OpenSCAP CIS Level 1 remediation at build time. Both bake in the Ops Agent and its configuration, auditd rules ending in `-e 2` (which locks the rules until reboot), a 15-minute shell timeout, package repository files that point only at the Artifact Registry mirrors with public repositories disabled, R and RStudio Server bound to localhost, XFCE and xrdp bound to localhost with

clipboard, drive, printer and audio channels refused, the SFTP subsystem disabled, the idle-stop timer, and the login gate. A CIS evidence scan report is kept with each image record. The build script has never been run.

This design is the main change that the first panel produced. Four of its six lenses independently noted that v1 deleted the default route but left package repositories pointing at public Rocky mirrors and installed the Ops Agent from a Google download host that is not on the restricted range, with failures hidden by errors that were ignored. v2 moved all of that into images and shrank the startup script to per-VM setup plus verification.

5.5.1 The fail-closed login gate

The v2 startup script wrote P4-OK or P4-FAIL to Cloud Logging, which an alert could catch, but researchers could still log in to a VM whose audit daemon was down. The second panel's security lens identified this, and v3 added an authentication-stack hook that makes the check a gate. The startup script, which runs at every boot and is written to be repeatable, is intended to close the gate by writing a failing status early in the boot. It then writes the SSH configuration (port forwarding only to localhost 8787 and 3389, a login banner), switches RStudio and xrdp on or off from instance metadata, applies `gui_only` if set, and runs the checks. Only if every check passes does it mark the status as passing. A small hook, installed in the login stacks for SSH, the remote desktop and RStudio, refuses any user who is not in the OS Login administrator group unless the status file starts with `OK`.

The checks are specific: the audit daemon and the Ops Agent are active; the audit rules report as locked; the SFTP subsystem is off; the gate is present in the SSH login stack; the shell timeout is set; the data disk is mounted; for desktop VMs the xrdp clipboard and drive channels are off; and nothing other than sshd listens on an address other than localhost. Figure 5.4 shows the sequence.

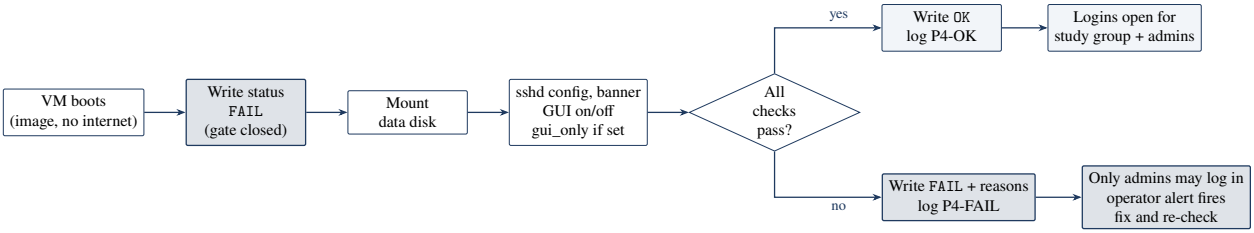


Figure 5.4: Boot sequence with the fail-closed login gate as the design intends it. As written, the status can survive a reboot, so the login service can start before the first box runs and honor the previous boot's `OK`; a boot-scoped status removes that window.

The gate is meant to start closed: the script writes a failing status early, so a crash after that point leaves researchers locked out, not let in. The ordering leaves a window, though. As written, the status file can survive a reboot, and a startup script runs only when the guest environment gets around to starting it. The login service can come up first and accept researchers against the previous boot's `OK`, and a step that hangs before the close holds that window open. Moving the failure write to the first line of the script does not close it on its own, because the login service can still start before the script does. A boot-scoped status does: keep the status in memory-backed storage that is empty at every boot, or bind it to the current boot identifier, so that a status from an earlier boot never counts. The gate is also evaluated only at login. It does not end sessions already open, and the checks run at boot, so a control that breaks later is not detected until the next boot or re-check; the plan refers to a re-check but has no separate periodic check. Pilot test 6 (break auditd on purpose and confirm that SSH, RStudio and desktop logins are refused while an administrator can still enter)

is the evidence the plan relies on.

The login gate turns a monitoring signal into an access control, and its first test is a fake-data pilot

v2 would have alerted on a failed boot check while still admitting researchers. v3 would refuse non-administrator logins until the checks pass (once the boot-time window above is closed), which converts an observed fault into a blocked path and makes the failure visible to the researcher as well as to the operators. The mechanism is specified as a short startup script and a small login hook that have not yet run on a real VM.

5.6 Reaching RStudio and the desktop

The browser and desktop tools add no firewall rule, load balancer or extra VM. RStudio Server listens on 127.0.0.1:8787 and xrdp on 127.0.0.1:3389, and the researcher's laptop runs one command that opens an IAP-backed SSH tunnel with two local port forwards [95]. The server permits forwarding only to those two ports. A browser then opens a localhost address for RStudio and a native RDP client opens localhost for the desktop. Getting to either still requires campus sign-in, the study group and a password; because OS Login accounts have no local password, each user would run a helper once over SSH to set a per-VM password used by RStudio and xrdp only. The helper sets it for the caller and refuses administrators. Whether this works for OS Login users, who may lack a local shadow entry, is part of pilot test 9.

The desktop moved to a native RDP client in v3. v2 had a browser-based Guacamole desktop, and the second panel's images and desktop lenses found that the build script never fetched the application archive, never created the directory it installed into and shipped a placeholder shared password, which would have failed the IS-3 unique-accounts expectation. Guacamole is deferred until it has per-user authentication. With a native client, xrdp's own configuration refuses the clipboard and drive channels, so the client's settings cannot re-enable them; the same restriction removes paste-in, so code would come in through the `in` bucket or through git over the mirrors. Pilot test 9 checks that the clipboard and drive channels are in fact blocked.

5.6.1 Idle stop

Cost in the plan's own estimate is driven by VMs left running, so a systemd timer powers a VM off after a configurable number of idle minutes (default 120, zero to disable; the GPU example uses 240). A VM is meant to count as busy if there is a login session, an RStudio session, a desktop session, any process owned by a regular user (intended to catch detached tmux and nohup jobs, including I/O-bound jobs under 10% CPU), load above 10% of cores, or a keep-running file on the data disk. The second panel found that v2's version, which looked only at logins and CPU, could stop quiet jobs; v3's process check was the fix. As written, though, v3's check misses the identity provider's users, so a detached job or an open tunnel probably would not keep the VM awake (the user-ID gap in Chapter 9). Each stop is logged with its reason, and disks persist. Whether it spares a running RStudio job and a tmux job is pilot test 12.

5.7 Software and logging in the architecture

Five Artifact Registry remote repositories proxy PyPI, Docker Hub, Rocky 9 BaseOS, Rocky 9 AppStream and EPEL 9, and a sixth, standard repository holds the curated containers (six repositories in the offline plan). Google's remote repositories cache upstream packages, and inside a perimeter they reach the upstream only after an administrator allows it [80]. As written, one mirror's upstream path would not resolve against

Google's preset base URL (Chapter 4). CRAN, Bioconductor, conda and the PyTorch wheel index are not mirrored. The plan covers them with three curated containers (R, PyTorch and bioinformatics) built by the research computing team outside the perimeter and pushed to the curated repository; a new R package is a request. An allowlist egress proxy is held back as a fallback only if the security office agrees. Chapter 8 evaluates this trade-off.

On the logging side, the foundation turns on admin-read, data-read and data-write audit logs for all services and routes a defined filter (audit logs, auditd, RStudio, xrdp, syslog, VPC flow logs and firewall logs) through a cross-project sink into the log project's retention bucket [98]. Alert policies go to a systems-team queue and a security-office queue (three channels, one of them in the log project): boot check failed, login refused by the gate, disk above 90%, IAM change, logging-config change, audit-config change, sudo use and export-bucket read, plus one policy that watches the log project itself. The alert filters match both text and JSON payload forms, since the second panel noted it was unverified where the Ops Agent places the message; pilot test 7 is meant to settle it. A monthly budget with 50, 90 and 100% thresholds and a BigQuery billing export complete the foundation. Chapter 10 treats detection and response.

5.8 What the offline plans contain

Table 5.2 summarizes the planned resources by role. All were produced by planning with placeholder values (project identifier, billing account, folder and group names are stand-ins), and every action in both plans is a create. The plan document's text says the foundation plan has 101 resources; the plan file I regenerated has 100, and nothing in the repository explains the difference.

Table 5.2: Planned resources by function, from the offline foundation and pilot plans.

Function	Count	Main resource types
Foundation plan (100 resources)		
Enabled APIs and IAM bindings	38	19 `project_service`, 19 `project_iam_member` (admins, security office in both projects, deployer, sink writer)
Organization policies	14	`org_policy_policy`
Alerting and budget	14	9 alert policies, 3 notification channels, 1 budget, 1 billing dataset
Network and DNS	18	1 network, 1 subnet, 2 routes, 5 firewalls, 3 DNS zones, 6 DNS records
Software mirrors	6	`artifact_registry_repository`
Projects, logging, state	10	2 projects, audit config, log bucket, sink, state bucket and its IAM binding, deployer service account and its IAM binding, 1 default snapshot policy
Pilot plan (59 resources, three studies)		
Per study (19 to 20 each)	59	VM, data disk, 2 snapshot attachments, snapshot policy, 3 buckets, 5 to 6 bucket IAM bindings, service account, 2 project IAM bindings (log and metric writer), 3 instance-level or service-account grants

5.9 How the design changed from v1 to v3

The plan was reviewed twice by panels of single-shot model reviewers (six lenses on v1, ten on v2), and each finding was checked against the actual files before being accepted; both verdicts list claims that were rejected as wrong about Google Cloud mechanics, and the second notes that five of the Terraform lens's

twelve findings were rejected. These panels are an internal quality check and not an independent assessment: the reviewers did not run any code. Table 5.3 gives the principal changes and the finding that drove each.

Table 5.3: Principal design changes by plan version and the review finding that drove each.

Version	Change in the plan	Finding that drove it
v1	Shared secure project, no-internet VPC, mirrors, per-study VM, SA and buckets; startup script installed software and ran checks that could fail silently	(baseline, reviewed by six-lens panel)
v2	Baked images with Ops Agent, auditd, mirror configuration and CIS hardening; startup reduced to setup and checks	First boot could not install anything: default route deleted, repos pointed at public mirrors, errors hidden (4 of 6 lenses)
v2	VM service account roles split: data admin, `in` read, `out` create-only; steward group reads `out`	Full object control on all three buckets
v2	Separate security-office-administered log project holding audit logs, state and billing export; deployer service account; systems team applies under approved change	Log bucket in the same project could be deleted by project owners; who applies Terraform
v2	Dropped project-wide `serviceAccountUser` for admins; dropped OS Login 2FA metadata	Admin blast radius (4 of 6 lenses); Duo SSO versus Google 2-Step Verification
v2	Curated containers for R, PyTorch and conda; RStudio Server and a desktop over IAP; idle-stop; budget and billing export; P4-OK/FAIL status	No CRAN, Bioconductor or conda access; no GUI for survey and R users; cost driven by always-on VMs
v3	Fail-closed PAM login gate on sshd, xrdp and RStudio	Boot check only alerted, researchers could still log in
v3	Locked audit rules (`-e 2`); alert filters match text and JSON payloads; logging and audit-config change alerts in both projects	Root could alter rules; unverified payload field; log tampering detection
v3	Desktop moved from Guacamole to native RDP through the tunnel; GUI password helper	Guacamole could not install as written and used a shared password
v3	Teardown script with evidence; noncurrent-version lifecycle; per-study snapshot retention; end-date label; RTO 1 business day and RPO 24 hours	Data disposal was not built; no restore target
v3	SFTP off; `gui_only` access mode; copy-out residual risk stated; affiliate accounts for outside collaborators	Authorized users can copy out over SSH (stated by several lenses); outside collaborators locked out by the `ucr.edu` policy

5.10 The fifteen pilot tests

The plan ends with fifteen acceptance tests for a pilot run on fake data. Every pilot-test number in this book refers to this list. None of the tests has run.

1. *Login.* Campus SSO with Duo works through IAP with no Google 2-Step Verification prompt, and an OS Login user has no sudo.
2. *GPU.* Secure Boot and the NVIDIA driver boot under the required Shielded VM settings, and a CUDA test runs.
3. *Network.* No internet; the mirrors work; IAP works without a default route; no VM-to-VM connections.
4. *Perimeter.* A week in dry run, then enforce and repeat test 3.

5. *Buckets.* The `in`, `out` and steward permissions behave as designed, an export read raises the alert, and the noncurrent-version lifecycle works.
6. *Fail closed.* Break a boot check: researcher logins are refused while an administrator can still get in, and the fix reopens logins.
7. *Alerts fire.* The status and privilege alerts arrive, and a change to the log sink alerts the security office.
8. *Audit rules* stay locked until reboot.
9. *RStudio and desktop.* The GUI password helper works, clipboard and drive redirection are blocked, and the service ports listen on localhost only.
10. *`gui_only`.* The shell is replaced by the password helper, copy commands fail, the tunnel still works and RStudio download is off.
11. *SFTP* clients cannot connect.
12. *Idle stop.* An idle VM stops; active jobs and the keep-running marker keep it up.
13. *Restore.* A study restores from its own snapshots within the recovery time objective and cannot attach another study's disk.
14. *Upload.* A 200 GB upload into `in` is timed and the result published.
15. *Teardown.* A dry run, then the real run, with the evidence record written.

5.11 Where the architecture stands

Coherent on paper, untested in practice

The architecture is internally coherent and several of its choices (the deleted default route, the fail-closed gate, a separate log project) are visible in the Terraform and scripts rather than only in prose. The gaps are equally concrete. The perimeter and domain restriction are absent from the offline plan; the 101 versus 100 resource count in the plan text needs correcting; MFA is inherited rather than built; reading the code against its intent shows three v3 weaknesses (the boot-time window in the login gate, the deployer problem, and idle stop's user-ID coverage); and the first-boot, GPU Secure Boot (test 2), IAP-with-no-default-route (test 3) and password-helper (test 9) behaviors are all tagged "verify in pilot" by the plan itself. The environment would also depend on the systems team accepting a role (apply, patch, rebuild images, on-call) that the plan marks as unconfirmed. The review panels found real code and document errors. They did not run anything, and they do not replace the pilot or the security office's own review.

5.12 Design considerations

1. Report plan status by control, not by resource count. A readiness table that separates "in Terraform, planned offline", "in Terraform, skipped for missing identifiers" and "process only" would stop the 100-resource figure from being read as coverage.
2. Obtain the access policy and directory customer identifiers early, so that the perimeter and domain restriction can be planned and tested in dry-run during the pilot rather than appearing after it.
3. Fix the deployer problem as described in Section 5.3: a break-glass group with an alert on every use, and teardown evidence in a locked bucket the deployer cannot write.

4. Make the gate status boot-scoped (memory-backed storage that is empty at every boot, or a status bound to the boot identifier), so that the login service can never honor a previous boot's `OK`; then break a boot step in pilot test 6 and confirm that only an administrator can log in.
5. Add a periodic re-check of the boot controls (not only at boot) and an alert if the status file changes from OK, then decide whether the gate should also terminate sessions or only block new ones.
6. Treat the pilot acceptance list as a gate with a recorded result per test, including the tests the plan tags "verify": Duo with no Google 2-Step Verification (test 1), Secure Boot with the NVIDIA driver (test 2), IAP with no default route (test 3), and the GUI password helper (test 9).
7. Specify keepalive settings in the researcher wrapper that fit within the ten-minute connection-tracking window, and test a long-running tunnel idle for an hour.
8. Reconcile the 101-resource statement in the plan with the regenerated plan, and regenerate and archive the plan output with each release of the Terraform.
9. State in the Risk Treatment Plan which controls are enforced by code, which are enforced only if the researcher or administrator follows a process, and which are accepted residual risks (shell-mode copy-out, screenshots, Google default encryption).

Chapter 6

Tenancy and Isolation: One Shared Project or One per Study

The plan puts every P3-P4 study into one Google Cloud project. Decision D3 says "one shared environment for all studies, not a project or cluster per study," D6 puts all secure VMs in one dedicated project, and D4 keeps Google's default encryption at rest and records the absence of customer-managed keys as a risk acceptance still pending, with the campus security office responsible and the security officer accountable. The stated reason is cost and effort: the plan wants to share infrastructure across studies so that the build stays small. Separation between studies is then carried by objects inside the project: one Google group, one service account, one VM, one snapshot schedule and three buckets per study, each bound by resource-level permissions.

The working analysis of the campus research cluster that preceded the plan recommended the opposite for a cluster: a separate project per study, with customer-managed keys, on the premise that data use agreements usually forbid co-mingling and that teardown is cleaner. Three of the AI-generated research reports consulted for this book repeat that view with stronger language, and one calls the shared project non-compliant and audit-deficient. I take D3 and D4 as our position and give the counter-argument in its best form, checking each claim on both sides against Google's documentation and the primary agreements I could read. The disagreement turns out narrower than the reports suggest. Several of the reports' most dramatic attack paths do not apply to the plan as written, a few of their claims are wrong, and a smaller set of real exposures remains. Most of those sit in the administrator plane and the log project, not in the researcher path. I then compare three layouts on equal terms: our shared project, a project per study, and a folder holding one project per study under one folder-level perimeter, which Google's documentation makes cheaper to run than earlier drafts assumed.

6.1 What a Google Cloud project does and does not isolate

The project is the unit at which Google applies most of the controls the plan relies on. Table 6.1 shows, from Google's documentation, which of the controls that bear on study separation sit at the project, below it or above it.

Two facts in the table decide most of the argument. First, inside one project, separation of studies depends on resource-level bindings and on the discipline of never granting a data-bearing role at project level. Second, a VPC Service Controls perimeter does not separate resources that sit inside the same perimeter. It controls what crosses the perimeter edge. That applies equally to one project with many studies and to many projects placed in one shared perimeter, a point taken up in Section 6.5.

6.2 How the plan separates studies inside one project

Everything that separates one study from another in our design comes from the per-study module. Read with the foundation's project-level bindings, it gives the layers in Table 6.2. The last column names who could cross each layer, because that is what an assessor would ask.

Table 6.1: Where Google Cloud applies the controls that bear on study separation. All rows from Google documentation read on 2026-11-01.

Control	What the documentation says	Scope
IAM allow policy	The effective policy of a resource is the union of the policy set on it and the policies inherited from its parents (organization, folder, project) [99]. A role granted on the project therefore reaches every resource in it. Some resources (Compute Engine instances, Cloud Storage buckets, service accounts) also take their own bindings.	Hierarchy, plus per resource
Firewall rules	Rules apply per VPC network. Targets can be network tags or service accounts. Any principal allowed to edit an instance can change its tags; Google advises target service accounts where strict control is needed [77].	Network (project, or host project of a Shared VPC)
VPC Service Controls	A perimeter protects services used by its member projects, folders or VPC networks. Projects created in or moved into a folder that is a perimeter member join that perimeter automatically. A VM inside a perimeter can read and write a bucket in the same perimeter [100–102]. A project is protected by only one regular perimeter in enforced mode (and one in dry run) [102, 103].	Project, folder or network
Organization policy	Set on an organization, folder or project and enforced on that resource and its children [104]. The plan's resource-location restriction and VM constraints are project-level.	Hierarchy
Quotas	A quota restricts how much of a resource a project can use, including API calls [105].	Project
Billing	A billing account says who pays and can be linked to one or more projects [106]. Cost by study inside a project needs labels.	Project to account
Default encryption	Google owns and manages the keys; customers cannot view them or review key usage logs. Customer-managed keys can be viewed, rotated, destroyed and logged [107].	Per resource, key per customer

Table 6.2: Layers that separate Study A from Study B in the planned shared project, from the per-study module and foundation configuration, and who holds the power to cross each layer. From the offline plan.

Layer	Mechanism in the planned configuration	Who could cross it
Login	OS Login and IAP tunnel roles granted to the study group on that instance only; OS Login required by organization policy; metadata SSH keys blocked per VM	Systems-team administrators (project-level administrative login and tunnel access)
Identity of the VM	One service account per study; the right to act as it granted to the study group on that account only; no project-wide user role for administrators	The deployer identity (which can run VMs as any study identity), and so every administrator, because the administrator group may act as the deployer
Data buckets	Three buckets per study; bucket-level roles for the study's service account and groups; uniform access, public access prevention	Deployer (broad project-level storage rights); anyone granted a project-level storage role
Disks and snapshots	Per-study snapshot schedule; labels carry the study name	Administrators with project-level instance administration; deployer
Network	One VPC; deny-all ingress and egress rules on all VMs; no route out except Google APIs and IAP	Whoever edits firewall rules; the rules apply to the whole network, not per study
API perimeter	One perimeter around the project (dry-run first; absent from the offline plan)	Not applicable inside the perimeter
Logs and evidence	One audit bucket in the log project, one sink, study name in labels and resource names; Terraform state and teardown evidence in a second bucket in the same project	The campus security office for the audit bucket; the deployer, and so any administrator acting as it, can write to the state and evidence storage

The researcher path is narrow. A study group member holds roles on one instance, one service account and the inbound bucket, and the module gives that group no role that would let it create instances, disks or snapshots. The attack paths in the reports that start with a researcher snapshotting another study's disk or attaching another study's service account therefore need permissions the plan does not hand to researchers. Attaching a service account to a VM requires permission to act as that account [108], and the plan withholds that right at project level from administrators on purpose.

The administrator plane is wide. The admin group holds instance administration at project level, so it reaches the instances, and through them the disks, of every study. On top of that sits the deployer problem from Section 5.3: administrators can reach study identities indirectly through the deployment identity, held back only by process. In a project per study the same administrators would still be able to reach each project, so the roles would not disappear. What would change is the scope of one mistaken grant and the amount of work an assessor needs to prove who can reach one study's data.

Evidence is the third difference. To answer "who can reach Study A's data" in the shared project, an assessor has to merge the project policy with the bucket, instance and service-account policies, because the effective policy is the union of inherited and local bindings [99]. In a project holding one study, one policy listing answers most of the question. That is a real audit-cost difference, and it is a documentation and tooling cost more than a security gap.

In the plan the cross-study exposure sits in the administrator and deployer roles, not in the researcher path

The module gives researchers resource-level roles on their own study and no permission to create or snapshot disks or attach service accounts. The roles that span all studies are the systems team's administrator group and the deployer identity, plus any project-level grant someone makes by mistake later. That plane decides how separate the studies really are. Either we limit it with controls an assessor can see (a scheduled check that no data-bearing role exists at project level, alerts on IAM change, deny policies), or the case for per-study projects gets stronger.

6.3 The case for a project per study

The reports argue for per-study projects on six grounds, and Table 6.3 checks the specific claims behind them.

Smaller blast radius for a mistake. A wrong grant in Project A reaches nothing in Project B because IAM inheritance stops at the project. Folder-level or organization-level grants still cross projects, so the argument holds only if no study-bearing role is granted above the study project. One of the AI-generated reports and the working analysis both use this argument.

Smaller assessment scope. One AI-generated report argues that an assessor will treat a shared project as one boundary. For a Controlled Unclassified Information (CUI) assessment under NIST SP 800-171 that may be true, but CUI is routed to the UCR's separate CUI enclave and is out of scope here (Chapter 1). The relevant frameworks for this environment are UC IS-3, the HIPAA Security Rule and data use agreements, none of which names a project. The HIPAA technical safeguards require access "only to those persons or software programs that have been granted access rights" and mechanisms that record activity [109]. They set outcomes and leave the boundary choice to the entity. The FERPA rule on outsourced school officials requires "reasonable methods" so that officials see only the records in which they have a legitimate interest, and names technological access controls as one such method [110]. A single shared project with resource-level controls can meet either text. Whether a campus security analyst or a data provider would accept it is a separate

question.

Independent quotas and incident containment. Quotas apply per project [105], so a heavy job in one project cannot exhaust another project's API quota. Containment follows the same shape: one project can be isolated or shut down without touching another study. The plan's per-VM controls give partial containment in the shared project (stop one VM, remove one group) but not for project-wide problems.

Clean disposal. Deleting a project removes every resource in it. Google's page on projects adds that soft-deleted projects count against the project quota and are fully deleted after 30 days [111], which is a timing fact a destruction certificate would have to state. The plan's teardown script instead lists and deletes every resource that carries the study's label and then checks that none remain. That works only for resources that carry the label. A disk someone created by hand last spring, without the label, would be missed by both the deletion and the check, and the evidence record would still say nothing remains.

Billing attribution. A billing account can be linked to one or more projects [106], so per-study projects allow per-study billing accounts if the chart of accounts needs them. The shared project has one billing account and cost-by-study through labels and a billing export. This touches the funding question: the project-pays position in the planning record (Section 1.7) is that P3-P4 projects pay through a chart of accounts, a memorandum of understanding and a separate billing account, and a single shared project can honor that only by internal chargeback.

Customer-managed keys per study. The reports pair per-study projects with a per-study key so that destroying one key renders one study's data unreadable. Section 6.6 treats this separately.

Checked, the debate looks different. Where the reports describe a researcher moving laterally, the plan's own roles block the path. Where they describe encryption keys as the barrier, Google's documentation points the other way for a shared project. Where they cite a UC rule, the standard that could be read says less than reported. What survives is the argument about administrator reach, assessment effort and the provider-specific agreements that have not yet been read.

6.4 Three options side by side

Option A is the plan: one secure project, one log project. Option B is a project per study, each a full copy of the foundation with its own network, organization policies and perimeter, vended by Terraform. Option C is the folder layout that one of the reports proposes: one folder for the environment holding a small platform project (Shared VPC host, software mirrors) and one service project per study, with organization policies, the log sink and the VPC Service Controls perimeter set once on the folder. Google documents that projects created in or moved into a folder that is a perimeter member join that perimeter automatically, without a change to the perimeter configuration [102]. Organization policies set on a folder are likewise enforced on its children [104]. A new study project in Option C therefore inherits the guardrails the plan now applies by hand to one project, and adding a study needs no perimeter or policy edit, though at about twice the shared design's per-study resources (Table 6.4 and the arithmetic below). Figure 6.1 draws the three.

Table 6.4 sets the options against six criteria. The cost and effort row combines a count from the real artifacts with judgment; I explain the count below the table. Each column assumes the option is built well: in particular, that administrator roles in B and C are granted per project and not on the folder, since a folder-level administrator grant would give C the same administrator reach as A.

Table 6.3: Claims from three AI-generated research reports and the working gap analysis, checked against the sources named. The reports are model-generated and treated as leads.

Claim	What the check found	Result
Project-level roles inherit to every study resource	Matches Google's IAM documentation [99]. The plan keeps researcher roles at resource level; administrators hold project-level compute roles	Verified
VPC-SC does not separate resources inside one perimeter	Google's overview gives an in-perimeter VM reading an in-perimeter bucket as normal behavior [100]. One report's own recommended macro perimeter around all study projects has the same property	Verified; cuts against that report's design
A stolen VM token gives lateral access to all study buckets	Each study's service account holds bucket-level roles on its own buckets only; the project-level roles for VM service accounts are log and metric writing	Not applicable to the plan as written
A researcher can attach another study's service account to their VM	Requires permission to act as that account [108] and permission to create or edit instances. The plan gives researchers neither; the deployer holds project-wide service-account user	Applies to the deployer, not researchers
Project-wide SSH keys reach every VM	OS Login is required by organization policy, and Google says an instance using OS Login ignores metadata keys [82]; the module also blocks project keys per VM	Largely addressed
A study identity can snapshot and mount another study's disk	Needs disk create and snapshot permissions that the plan gives to administrators and the deployer, not to study groups. Not tested here	Administrator-plane exposure
Per-study CMEK stops that mount because the user lacks key access	Google's disk documentation has the project's Compute Engine service agent hold the key role, one agent per project [112]. In a shared project that one agent would hold every study's key role	Does not hold in a shared project (follows from Google's CMEK page)
Default encryption gives no key-use log; CMEK does	Google's comparison table: logging of administrative and data access to keys is available for customer-managed keys and not for default encryption [107]	Verified
Destroying a CMEK key crypto-shreds data instantly	Google recommends a minimum scheduled-destruction period of 30 days for keys [113], so a recommended configuration is not instant	Partly verified
UC IS-3 section 8.2 requires CMEK for P4	The UC key-management standard says private keys protecting institutional information are classified P4, P4 key access must be recorded, and keys must not be reused for separate information. It does not mention cloud providers or CMEK [114]. I did not find a section 8.2 in it	Not supported by the standard read
Noisy-neighbor quota exhaustion; VPC-SC attribute limits of 50 to 100 per access level	Per-project quotas are documented [105]. The numeric limits were not found in the pages read	Numbers unverified
Folder-level perimeters are inherited by new projects (one report)	Google's folder-membership page: projects added to a folder configured in a perimeter join it automatically, with no change to the perimeter configuration; a project's own explicit assignment takes precedence [102]	Verified; folder membership is not supported in perimeter bridges
CMS prohibits commingling absolutely and requires destruction in 90 days	The CMS and ResDAC texts read bar linking beneficiary records to other data without CMS authorization and limit access to a need-to-know basis. The 90-day rule applies to shipped physical media [115, 116]	Partly verified; "absolute" not found
Most UC campuses require CMEK for P4 (gap analysis)	No source was cited in the gap analysis and none was found	Unverified

Table 6.4: The three tenancy options compared. The plan is option A. Entries on data-provider acceptability reflect the agreements that could be read (Section 6.7); none of them names a cloud project.

Criterion	A. Shared project (plan)	B. Project per study	C. Folder, project per study
Isolation boundary	Resource-level IAM, per-study service account, firewall deny rules, host-level controls; one perimeter	Project IAM boundary; own network and quotas; optional perimeter per study for API-level separation	Project IAM boundary and own quotas per study; shared network through the Shared VPC host; one folder perimeter, so no API-level separation between studies (as in A)
Blast radius of a wrong grant	A project-level grant reaches all studies; resource-level grants reach one	A project-level grant reaches one study; folder or organization grants reach all	A project-level grant reaches one study; a folder-level grant reaches all, as in B
Blast radius of administrator error or compromise	All studies, through project-level instance and storage roles	One study per project role; the platform and vending identities still span all	One study per project role if administrator roles are granted per project; the platform, vending and folder administrators still span all
Audit evidence	One project policy plus per-resource policies to merge; logs filtered by study label; one assessment scope	One policy and one log set per study; evidence for one DUA is a bounded set	One project policy and one log stream per study, plus the folder policies as an inherited layer to explain
Cost and overhead	About 20 added resources per study in the offline pilot plan; one network, one set of mirrors, one sink	Up to about 100 resources per study if the foundation is repeated unchanged; a vending pipeline to build and own	Platform, mirrors, network, org policies and perimeter set once on the folder; each study adds its project, API enablement, a budget, a Shared VPC attachment and the module (roughly 40, an estimate not yet planned)
Admin effort	Lowest to build; grows with studies as each adds resource-level bindings to review	Highest to build; steady per study once vended; many projects to patch, monitor and audit	Middle to build (folder rights, Shared VPC, a vending module); per study close to A, since the perimeter and policies need no edit; more projects to monitor
Data-provider acceptability	Unknown until each agreement is read; a provider that requires separation or its own key fails it	Meets a separation or own-key requirement without exception handling	Meets a separate-project requirement; a separate-perimeter requirement needs that study moved to its own perimeter (its explicit assignment overrides the folder)
Teardown and disposal evidence	Label-driven script; checks only labeled resources; logs remain in the shared locked bucket	Project deletion with a 30-day soft-delete window; logs still need a disposal answer	As B; a removed project also leaves the folder perimeter automatically
Billing	One account; chargeback by label and export	Per-study accounts possible	Per-study accounts possible

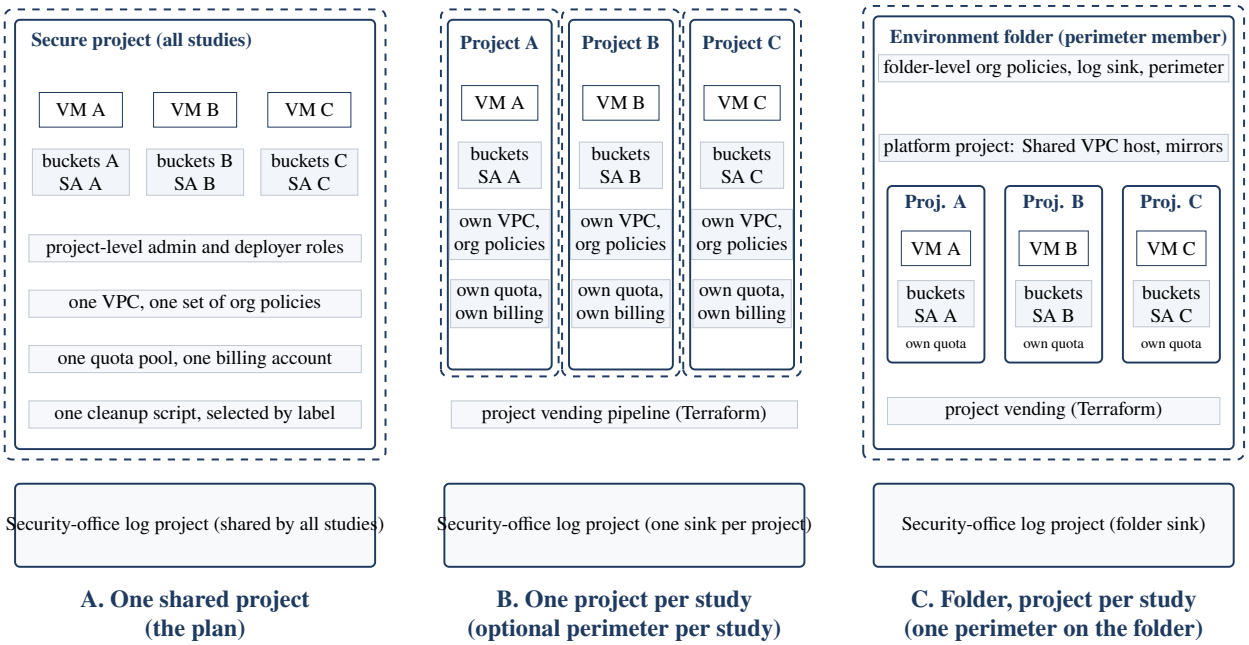


Figure 6.1: Three tenancy layouts for the planned environment. Dashed lines are VPC Service Controls perimeters: one around the shared project in A, optionally one per study in B, and one covering the environment folder in C, which new study projects join automatically when created in the folder. Panel A is drawn from the planned foundation and per-study module. Panels B and C are proposals, not part of the plan.

The resource arithmetic uses the offline plans. The foundation plan has 100 resources, and by project attribute 82 of them sit in the secure project, including six Artifact Registry repositories for software mirrors. Repeating everything but the mirrors per study project would be 76 resources, plus the project, a budget and the 19 or 20 resources of the study module, which comes to about 100 per study. That is an upper bound: a vending module would share mirrors, probably the network through a Shared VPC host [117], and possibly the alert policies. It shows the order of magnitude, roughly five times the per-study footprint of the shared design if nothing is shared. The module itself already takes the project and subnetwork as inputs, but the foundation takes a single project identifier and would have to be reshaped as a reusable module. Option C sits between the two. Its per-study additions, estimated from the same plan and not produced by a Terraform plan, are the project, about 15 enabled APIs, a budget, a Shared VPC service-project attachment, four or five project-level administrator bindings and the 19 or 20 module resources: roughly 40, about twice the shared design's per-study footprint. The network, mirrors, organization policies, alert policies and perimeter would be written once, at folder or platform-project level, and would need no edit when a study is added. The one-time cost is higher than A's: folder-level rights for organization policy and the perimeter (which the plan's notes say the deployer deliberately lacks), a Shared VPC host, and a vending module. I did not price any design; Chapter 12 covers cost.

Where Option C stands against the plan. A and C share more than the framing "shared project or project per study" suggests. Both have one network, one set of mirrors, one perimeter and one log destination, and in both a study is added by a Terraform module call that needs no perimeter or policy edit. What C adds is the project boundary: a project-level grant reaches one study, quotas and billing separate by default, an assessor can answer "who can reach Study A" from one project policy plus the folder's, disposal can use project deletion, and a study that hits a trigger in Section 6.10 can be given its own perimeter or key project without leaving the environment. What C costs is a larger one-time build that needs folder-level rights the

plan's deployer deliberately lacks, a Shared VPC host to run, roughly twice the per-study resources, and more projects to patch, monitor and include in access reviews. Some things C does not change. Members of one folder perimeter are not separated from each other at the API layer, a person on two studies still links them, and if the systems team's administrator roles are granted on the folder instead of per project, the administrator plane spans every study exactly as it does in A. The deployer problem and the perimeter service-name defect (Chapter 4) would carry into C unchanged.

We still hold D3, the shared project, chosen for the smallest build and the fewest objects to run, and the comparison does not overturn it. It does change the alternative leadership is choosing against. Option B, a full foundation per study, was a costly fallback; Option C is a design that could be adopted from the start or moved to later, with study modules that already take the project as an input. Whether to accept A's administrator reach and single policy scope in exchange for the smaller build, or pay C's one-time cost for a project boundary per study, is a decision for the campus security office, the security officer and the systems team, and belongs in the leadership decision on D3 (Chapter 14).

6.5 Perimeters and the limits of VPC Service Controls

The plan's perimeter lists the secure project's number as its only resource and restricts a defined set of services. Two points follow from Google's pages. A project is protected by only one regular perimeter in enforced mode [102, 103], so a study project in a shared perimeter cannot also sit in a per-study one. And a perimeter does not separate members from each other. One report's recommended design wraps all study projects in one "macro" perimeter and then credits the perimeter with isolation between them. Per Google's overview that protection does not exist inside a perimeter, so isolation between those projects would come from the project IAM boundary alone [100]. A design that wanted API-level separation between studies would need a perimeter per study, with ingress and egress rules for anything that had to cross, and the number of rules would grow with the number of studies.

Google documents folder membership in perimeters: projects created in or moved into a folder that is a perimeter member join that perimeter automatically, and leave it when removed from the folder. When a project and one of its ancestor folders sit in different perimeters, the lowest explicit assignment wins, so a project's own assignment takes precedence over the folder's [102]. The report was right on this point; version 1.0 of this book called the claim probably wrong, and that was a mistake. The page lists limits: folder membership does not work with perimeter bridges, it does not cover perimeters configured with allowed service patterns, and, through a known issue, placing a VPC network's project explicitly in a dry-run perimeter removes the enforced protection it would inherit from its folder. The last one matters for Option C's dry-run phase, because its platform project holds the Shared VPC network.

This changes the cost of Option C more than its security. The perimeter can name the environment folder once, and a new study project needs no perimeter edit. It does not give API-level separation between studies, for the reason above: members of one perimeter are not separated from each other. A study that needed that separation could be given its own perimeter by explicit assignment, which overrides the folder, at the cost of ingress and egress rules for whatever crosses. The two perimeter fixes in Chapter 4 (name the exact API each access path calls; give every ingress rule the campus access level once it exists) apply to all three layouts.

6.6 The customer-managed key debate

On D4, Google's documentation is more even-handed than the reports. Google's CMEK page compares default encryption with customer-managed keys on several rows. Customers own and manage CMEK keys

and can view their metadata, control rotation, apply organization policies to them, log administrative and data access to them and get logical data separation through encryption, and the page marks default encryption as not supporting regulatory requirements for customer-managed keys [107]. Under default encryption, Google owns the keys, customers cannot view them or review usage logs, and data from several customers is typically protected by shared key-encryption keys. That is the substance of the counter-argument, and it is a fair one if a provider's agreement or a regulator asks for key custody.

Google's recommended-practices page frames CMEK as the choice where you want a cryptographic boundary around data, and lists the capabilities it gives as owning the keys, controlling location and lifecycle, selective deletion for off-boarding (crypto-shredding), unique keys that form a cryptographic boundary, logged key use and meeting regulation that requires any of these [113]. The same page recommends a separate key project per environment folder, a minimum scheduled-destruction period for keys of 30 days, and notes that finer key granularity costs more in key versions and automation.

Cryptographic separation needs per-study projects. In a shared project, the Compute Engine service agent is one principal per project, and Google's disk documentation has that agent hold the encrypt and decrypt role on the key [112]. With several study keys in one project that single agent would hold all of them. Google's CMEK page states that a requester does not need access to the key as long as the service agent holds the encrypter and decrypter role [107], so a principal who obtained the permissions to snapshot or attach another study's disk would not need the key role itself. This follows from Google's documentation; I have not tested it, and a cross-study attach test belongs in the pilot. It means the report's claim that per-study keys block cross-study disk mounts holds at best in a project per study and is weak in the shared project. On the documented behavior, in one shared project, per-study customer-managed keys do not isolate studies from each other, because one service agent holds every key. CMEK in our layout would add key-use logging, a destruction control and an answer for providers about key custody, and nothing more.

Crypto-shredding is not instant. The FERPA report calls key destruction an instant, auditable route to the certificate of destruction. Google's guidance recommends a 30-day minimum scheduled-destruction window for keys, so a destruction record would carry a wait of that order [113]. That is similar in timing to the 30-day soft-delete window for a deleted project. The plan's teardown script, by contrast, deletes the resources and records what remained. It would need a way to show deletion of snapshots and object versions, and that would carry similar timing questions.

What UC's own standard says. The UC key-management standard classifies private keys that protect institutional information at Protection Level 4, requires that use of keys protecting P4 information be recorded so that it is auditable, and says never to reuse a key to encrypt another set of unrelated or separate information [114]. The text addresses "workforce members handling private keys," so whether it reaches keys that Google holds inside default encryption is an interpretation question. If the campus security office reads it to reach them, D4 conflicts with the audit clause, and the reuse clause reads as pointing toward per-study keys. If it reads it not to reach them, D4 stands. The plan lists D4 as a risk acceptance that is still pending. The report's "section 8.2" did not appear in this standard, so the claim that IS-3 requires CMEK is unverified here.

The counter-argument for CMEK is therefore strongest as a provider and security-office question and weakest as an isolation argument. If the plan adopted CMEK at all, the cleaner shape is Option B or C with a key per study in a separate key project, because that combination gives the cryptographic boundary that the key alone cannot give in a shared project. Google's page recommends one key project per environment folder [113], which fits Option C's folder directly.

6.7 What data providers actually say

The strongest argument for per-study projects would be a provider whose agreement forbids co-hosting. I read the primary texts that could be reached.

The NIH document for users of controlled-access data (updated 2024-07-25, about 470 words) sets NIST SP 800-171 as the standard and asks users of third-party systems to give the repository an attestation [118]. It contains no requirement about hosting several studies together. NIH controlled-access data routes to UCR's separate CUI enclave in any case (Chapter 3), so one report's discussion of dbGaP sharing groups does not bear on this environment.

The CMS Research Identifiable File policy guide and the data use agreement are more specific. The agreement bars linking beneficiary-level records to any other source without express CMS authorization, limits access within the organization and its collaborators to the minimum data and number of people needed (need-to-know), and requires destruction of original, derivative and backup files by the expiration date with an attestation of destruction [116]. The policy guide sets a 90-day destruction rule for physical media shipped by CMS, requires data in an approved environment to be destroyed on closure of the DUA unless another active DUA covers it, bars reuse of data from a collaborating organization, and recommends access through CMS's own virtual research data center [115]. None of this uses the word project, and I found no absolute ban on co-hosting. The linking clause and the destruction clause would shape how a shared project is run. Study A's researchers must not be able to join CMS data with Study B's data, and destruction must reach backups, which for the plan means snapshots and object versions. A provider whose own environment is the recommended route may also never send physical data to UCR at all. The CMS texts were read for one data type. Other federal, state and commercial providers were not read, and their terms may differ.

FERPA's redisclosure rule makes the disclosure conditional on the receiving party not disclosing the information to any other party without consent [119]. Whether a systems-team administrator or a researcher on another study counts as another party depends on how the agreement names the people and roles and is for the campus FERPA officer. The plan's one-group-per-study design maps naturally onto a named-user list, which Chapter 3 notes is a common agreement term. HIPAA has no boundary rule beyond the outcomes cited above.

In short, no agreement read here requires a separate cloud project. That is a statement about the texts read, not a finding that no provider will. The working analysis's premise that agreements usually forbid co-mingling is not borne out by the CMS texts, which restrict linkage and access. An agreement that says the data must be kept "physically or logically separate" would be a trigger, and the intake checklist would need a field to catch it (Section 6.10).

6.8 Identity as the commingling path

The Australian KeyPoint trusted research environment has a presentation on "managing identities to enforce strict project separation at the virtual desktop," which reports that it associates roles with a federated identity through what it calls personas, to eliminate "inadvertent opportunities for data linkage or masquerading data ingress and data egress via different projects" [52]. A companion abstract describes a self-contained vault architecture, vault-specific directory instances and complete separation of research activities [51]. They describe separation per project inside a shared platform, with a vault per project, and they do not describe sharing at the level of a single cloud project. They are better read as support for the idea that one operating team can run many separated environments than as support for one shared boundary.

For our plan, the person is the link between studies. A researcher who sits on two studies' groups holds a login to two VMs and a laptop, and in shell mode the plan accepts that copy-out is possible but prohibited and

logged (decision D12). Separate projects would not stop the same person from moving data between studies. Only the agreements, the named-user list and the gui-only mode do that. The intake step should therefore ask whether anyone on the study also works on another P3-P4 study with a data provider that restricts linkage, and the DSP should say what that person is allowed to do. Stanford's Nero pages describe a shared platform to which a team is given a "Nero GCP project" and bills costs to the PI's account [120]. Nero, the one verified Google Cloud peer, therefore gives each requesting team its own Google Cloud project rather than one shared project; its pages do not say whether projects are per protocol or per team, and the report's statement of a project per protocol is not confirmed on the pages read.

6.9 Shared file systems and a future cluster

The plan's phase 2 sketches a small Slurm cluster in the same secure project with one shared Filestore and a directory per study, and says that study separation on the shared file system would be by Unix permissions only, which the campus security office would have to accept. Here our own text matches the report's concern. The working analysis of the campus research cluster lists a shared file system with Unix-permission isolation as one of the reasons that cluster does not fit P3-P4 data, because one administrator or root user on any node sees everything. That analysis then recommends a cluster, a Filestore and a key per study in a project of its own. Read together, they disagree: phase 2 carries forward the shared design the gap analysis rejected for clusters.

One AI-generated report adds that POSIX access lists drift, that mapping identities to numeric user and group identifiers can collide when people leave, and that file-level access cannot be reconstructed from audit logs. These are plausible and match common experience, but they are the report's claims and I did not verify them against Filestore documentation. The reasoning that holds without them is simpler. Cloud Storage bucket IAM separates studies by identity at the API, with a data-access log entry per object read. A shared POSIX tree separates them by operating-system permissions on machines that administrators can become root on. If phase 2 goes ahead, the safer plan is one Filestore instance per study, or a project per study cluster as the gap analysis proposed, and the plan's rough cost for the smallest instance (about $200 a month, the plan's own estimate, not checked against the price list) shows why that choice has a price tag.

6.10 Conditions that would move a study into its own project

The plan does not say what would make a study leave the shared project. The conditions below are my proposals for the intake step. Each is a fact that can be checked when a study applies, and any one of them would move that study to its own project, or reopen D3 for all studies if several occur in the first year. Where the environment already has a perimeter folder (Option C), that move is a project created in the folder; without one, it is a project vended outside the shared design (Option B).

1. **Separation language in the agreement.** The data use agreement, data sharing agreement or business associate agreement requires the data to be physically or logically separate from other data, forbids co-hosting with other sponsors' data, or requires a dedicated environment.
2. **Key custody requirement.** A provider or the security officer requires a customer-managed key, or key destruction as evidence of disposal, for that study. A per-study key is only useful in its own project (Section 6.6).
3. **A linkage clause that the platform cannot enforce.** The agreement restricts linking its data to other data, and the study's team also works on a second study whose data could be linked. Separate projects do

not stop a shared person, so this trigger pairs with a named-user restriction in the DSP.

4. **Data location or service differences.** The study needs a region other than us-central1 (the plan restricts locations at project level) or a Google service that the shared perimeter's allowed-service list excludes. Both are project-level settings.
5. **Different administrators.** The provider or the campus security office requires named administrators, a different background-check regime, or exclusion of the systems team's administrator group from that study's data.
6. **Different perimeter or access path.** The study needs ingress rules that the shared perimeter should not carry, for example a path for an outside collaborator beyond affiliate accounts, or per-study API separation.
7. **Separate billing account.** The chart of accounts or the funder requires the study's cloud costs on a distinct billing account that label-based chargeback cannot satisfy.
8. **Log disposal.** The agreement requires the study's logs to be destroyed with the data, which a locked shared audit bucket cannot do (Chapter 10).
9. **Quota or availability.** The study needs GPU or API quota that would crowd out others in the shared project, or a studied outage shows one study affecting another.
10. **An assessor or security-office finding.** The campus security office recommends against, or the security officer declines, the pending risk acceptance of D3, or an audit, incident review or pilot test shows that cross-study access is possible through a role that cannot be removed.
11. **Scale of administration.** The number of concurrent studies makes the per-study resource-level review unmanageable. The plan gives no number, and the campus security office and the systems team should choose one before the first study starts, with a review of D3 at that count.

Triggers 1, 2, 5 and 10 are the likeliest in the first year on the evidence available, because they come from agreements and from the campus security office. Triggers 4, 6, 7, 8 and 9 come from design settings that the plan fixes at project level, which is why the project is the natural escape hatch.

D3 is defensible, and Option C is a real alternative

D3 and D4 are decided, but neither risk acceptance has been given; the plan lists both as pending, with the campus security office responsible and the security officer accountable. The planned configuration separates researchers well at resource level and spans all studies through the administrator plane, including the deployer problem (Section 5.3). The fifteen pilot tests (Section 5.10) include a VM-to-VM network check (test 3), an outbound-bucket read check (test 5) and a restore that cannot attach another study's disk (test 13), but none asks whether a member of one study's group can read another study's bucket, which a three-study pilot could test directly. The foundation takes one project identifier, so a move to Option B or C requires restructuring the foundation. The module is project-agnostic, which helps, and Google's folder membership for perimeters means Option C would not need a perimeter edit per study. The plan has no intake field for a provider's separation language and no written reversal conditions. The working analysis's own recommendation for clusters points the other way, and its premise that "most UC campuses require CMEK for P4" has no source. Where I come out: D3 is defensible for the VM pattern on cost. Option C is a real alternative leadership should weigh. D4 is open for the security office and officer. Phase 2 shouldn't inherit the shared design without a fresh decision.

6.11 Design considerations

1. **Write a decision record for D3 and D4.** State the reasons (cost, build effort), the evidence checked here, Option C as the alternative considered, and the reversal conditions in Section 6.10. Ask the campus security office to recommend and the security officer to accept or reject D4 explicitly, with a reading of the UC key-management standard in writing.
2. **Add intake questions that detect triggers.** Ask for any separation, dedicated-environment, key-custody, log-disposal, location and linkage language in each agreement, and whether any team member is on another P3-P4 study. Make a yes answer route to the campus security office before a VM is built.
3. **Control the administrator plane.** Narrow the deployer's project-level roles where the module allows, apply the deployer fix in Section 5.3, and add a scheduled check that no data-bearing role exists at project level. Consider IAM deny policies as a guardrail. Alert on any project-level grant of a storage or compute role.
4. **Add cross-study negative tests to the pilot.** Alongside test 13, use the three pilot studies to check that a member of one study's group cannot list or read another study's buckets, log in to its VM, or use its service account, and that an administrator cannot start a VM as a study's identity.
5. **Target firewall rules by service account.** Network tags can be edited by anyone who can edit the instance, so a per-study tag is a weak selector. Where a rule needs to differ by study, target the service account.
6. **Produce a per-study access report.** A script that merges project, bucket, instance and service-account policies for one study would give an assessor the single listing that a project per study provides, and would be reusable in access reviews.
7. **Tighten teardown evidence.** Make the teardown script check that no unlabeled disk, snapshot or bucket exists in the project, and state the 30-day or key-destruction timing in the destruction record for any study that uses project deletion or key destruction.
8. **Settle log disposal.** Decide whether an agreement requiring log destruction would be answered by per-study log buckets (not locked, shorter retention) or by a project per study, before locking the shared bucket.
9. **Decide the study-count review point.** The campus security office and the systems team should pick the number of concurrent studies at which D3 is reviewed, and the plan should record it.
10. **Prototype Option C before it is needed.** Turn the foundation into a folder-level platform plus a study-project module, and test one study project in a folder perimeter with a Shared VPC host in dry run (watching the documented issue with a network project in a dry-run perimeter), so that a trigger, or a leadership choice of C, leads to a bounded task and not to a redesign. Pair the prototype with the experience of the team that runs UCR's separate CUI enclave with project vending.
11. **Do not carry the shared design into phase 2 by default.** If a cluster is built, evaluate one Filestore instance per study or a project per cluster, and record the campus security office's view of Unix-permission separation before the design is fixed.
12. **Read more provider agreements.** Obtain the actual agreement text for the first expected data providers (including at least one state agency and one commercial source), and the campus FERPA officer's view of administrator access to education records, before the first study is accepted.

Chapter 7

Data In and Data Out

In our plan, data enter a study by one route, a researcher uploading files from a laptop to an inbound bucket, and leave by one approved route, a data steward downloading files from an outbound bucket after reading them against a six-item, one-page release record. Between those two buckets, firewalls, identities and logs do the work. At the way out a person has to decide that a table or a figure no longer identifies anyone. A researcher with a shell could also copy data to a laptop, and the plan records that as a residual risk (D12). Below I follow data along that path, check each control against the plan's files, and compare the release checklist with the rules that national statistics agencies and the U.S. Centers for Medicare and Medicaid Services (CMS) apply.

The pilot tests that would confirm these controls have not run. Where I say that a control "would" work, the evidence is a configuration file or a vendor document, and where I say a control does nothing, the evidence is the vendor's own source or documentation. Sources I could not read are marked *not verified*.

7.1 The planned path in and out

7.1.1 Ingress through the inbound bucket

In our plan, each study would receive three Cloud Storage buckets from the per-study module: a working-data bucket, an inbound bucket and an outbound bucket. The study's service account holds object administration on the working bucket, object viewing only on the inbound bucket, and object creation only on the outbound bucket. The study group (everyone on the signed Data Security Plan, DSP) can create, list and read objects in the inbound bucket. Among study roles, only the named steward group can read the outbound bucket (the deployer can too; see the deployer problem in Section 5.3), and every read of an object in an outbound bucket would raise an alert to the campus security office. The offline foundation plan turns on Data Access audit logs for all services, which is what makes bucket reads and writes visible [121]. Each bucket has uniform bucket-level access, enforced public-access prevention, versioning, and a lifecycle rule that deletes noncurrent versions after 30 days by default (Chapter 5).

The plan names one ingress method, a bucket upload from the researcher's laptop, and leaves the method to the researcher guide. Pilot test 14 would time a 200 GB upload from campus.

The data reach the inbound bucket through a laptop. A provider that ships files by download link or portable media lands them on an unmanaged device before the upload, and the plan has no rule for that laptop. Nothing checks what arrives: the files I searched (plan, runbooks, images, Terraform) contain no checksum comparison, malware scan or inventory step for the inbound bucket, so a corrupted or mislabeled file would first be noticed by a researcher. And the study group can read the inbound bucket, so any member can read back from their own laptop what any member uploaded. The VM's service account cannot write to the inbound bucket, so results cannot flow that way, but the bucket is a standing copy of the source data that outlives its usefulness once the data are loaded to the working bucket. The module has no rule that clears the inbound bucket after ingestion.

The perimeter may block our own upload and release paths

The upload and download paths cross the planned perimeter, and I found no ingress rule that would admit them. The rules are written for administrators and for IAP; none names the storage calls that study members and stewards would make, and an ingress rule only admits the calls it names. An enforced perimeter would be expected to refuse the laptop upload and the steward's download, because both are storage API calls from outside the perimeter. The offline plan contains no perimeter at all, so no test has shown this either way. It is the first thing the dry-run week should confirm (Chapter 4, Chapter 6).

7.1.2 Globus as an optional ingress path

Practitioner reports describe Globus as a common campus route for large transfers of protected data (not verified here), and our plan comparison lists evaluating it as recommendation S8. I verified only Globus's own statements, which are narrower than those reports. Globus requires a high assurance collection when protected health information (PHI) is disclosed to its service, and requires the institution to hold a business associate agreement with the University of Chicago [122]. High assurance policy requires sign-in with a specific identity provider and re-authentication after an administrator-set timeout, and audit logs are retained under the subscriber's own policy [123]. The user guide tells users never to enter sensitive data in input fields such as transfer labels, collection names, descriptions or group names, and not to put it in file names or directory paths unless the files are on a high assurance collection [122].

What I could not establish is whether UCR holds a Globus subscription with high assurance features or the BAA, and whether a Globus endpoint can run in a project with no internet route and no Cloud NAT. A Globus Connect Server endpoint would need to talk to Globus services; I did not verify the network requirements, so the compatibility with the plan's network is *not verified* and is a design question for the systems team. If an endpoint could run only with an outbound path, it would be the first exception to the no-internet design and would belong in its own perimeter review.

7.1.3 Egress through the data steward

The planned release runbook has five steps. The researcher puts files in an outbound folder on the VM and asks the steward. The VM copies them to the outbound bucket. The steward reviews them against the one-page export release record. The steward downloads them and records the release (who, what, why, date). The campus security office receives the export-read alert and compares it with the record in the weekly review. Figure 7.1 draws the path with the points where an event is logged or alerted, and the copy-out routes the plan does not claim to close.

The alert fires on every read of an outbound object, so it cannot tell a review from a release; the campus security office matches the alert to the record by hand in the weekly review, and the record is a paper form with no machine link to the object. The steward downloads the files in order to review them, which means unreviewed output, possibly disclosive, lands on the steward's device. The plan does not say whether the steward is on the DSP, trained for the data, or independent of the study team. And nothing records what a steward has already released, so cumulative releases are invisible; this matters for the rule in 34 CFR 99.31(b), discussed below.

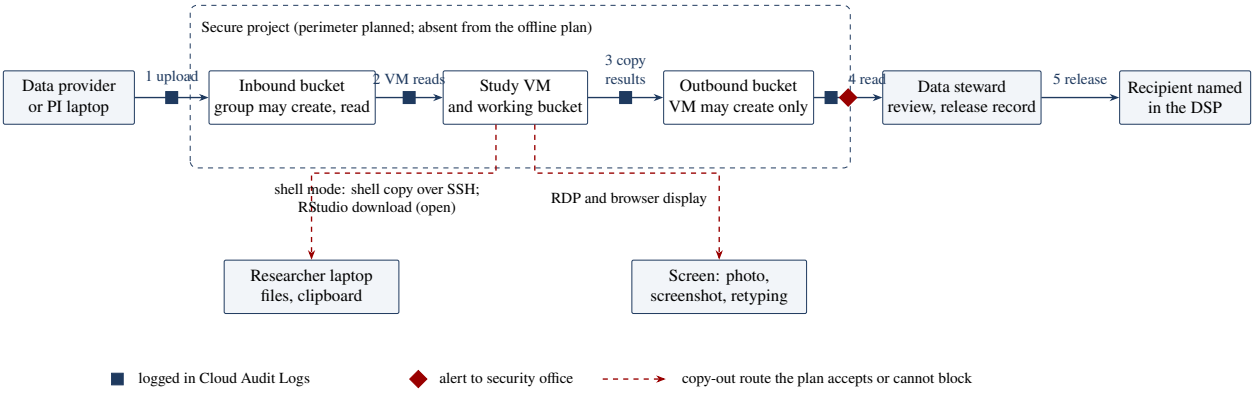


Figure 7.1: Planned data path for one study, drawn from the per-study module, the release runbook and the audit rules. Steps 1 to 3 are Cloud Storage calls (a write, a read and a write) and are visible in Data Access logs; only step 4 raises an alert. Step 5 and the dashed routes are outside every technical log: the guest audit rules cover some ways of copying files but not all, and Chapter 10 found no alert on them.

7.2 The perimeter and the tunnel

The plan states that a VPC Service Controls perimeter would stop data leaving through Google APIs to buckets in other projects, even with valid credentials. That is the technical backstop for the VM-to-API route: a researcher on the VM who tried to copy to a personal bucket would hit the missing internet route first, and the perimeter second. The perimeter is dry-run first and is absent from the offline plan, so for now the controls on that route are the missing route and the VM service account's IAM bindings, which grant roles on that study's buckets only.

For data moving through the tunnel, Google's text is short: IAP TCP forwarding is not intended for bulk transfer of data, and IAP reserves the right to rate-limit users abusing the service [95]. A report in the evidence set claims a specific throttle; Google's page does not say that, and I treat any throughput limit as unknown until pilot test 14 measures it. The plan's decision to move data by bucket and to leave the tunnel for interactive traffic is consistent with Google's statement. It also means that the tunnel is not the intended copy-out route in shell mode, only a usable one, which is the difference between a control and a deterrent.

7.3 Copy-out from inside the VM

The plan's position on copy-out is decision D12: the residual risk is stated, shell mode accepts it (prohibited, logged, accepted by the security officer), and `gui_only` mode is for studies whose data use agreement (DUA) forbids copy-out. The desktop and RStudio design document says what each mode does and does not stop. Those statements can be checked against the image files. Table 7.1 lists each route I found, the control the files show, whether the route is logged, and what remains.

7.3.1 Shell mode and the SSH routes

The image's SSH server configuration disables the SFTP subsystem, so drag-and-drop clients cannot connect. The planned design notes that the default copy mode of recent OpenSSH clients also fails, because OpenSSH 9.0 moved file copy onto the SFTP protocol, while legacy modes still work and are audited. The OpenSSH 9.0 release notes confirm the switch [124]. The claim is accurate, and it shows what the control is: the SFTP block does not stop a copy, it removes the easy tools. Anyone with an interactive shell can move bytes to

Table 7.1: Copy-out routes from a study VM, the control the plan's files show, and the residual risk. "Logged" means a log source exists, not that anyone is alerted. Routes 4 and 5 depend on pilot tests 9 and 10, which have not run.

Route	Control in the files	Logged?	Residual risk
1. Graphical and default file-transfer clients over SSH	SFTP subsystem disabled in the image's SSH server configuration; recent OpenSSH clients use SFTP for file copy by default [124]	Failed sessions appear in SSH server logs only if shipped (Chapter 10)	Older clients and legacy transfer modes still copy; the control is a side effect of a client default
2. Shell copy over the SSH session (shell mode)	DSP prohibition and login banner; the plan states it is accepted as residual risk (its RACI and D12 make the security officer the accepting party)	Some copy methods audited, not all; no alert	Deliberate copy by an authorized user; detection only by weekly search
3. Extra port forwards	Local forwarding limited to the two service ports	Not separately	None beyond the two forwarded services, which are routes themselves
4. RDP clipboard, drives, printers, audio	Static channels for clipboard, drives and audio turned off at the server; session-level restrictions not set	xrdp session logs shipped	The granular clipboard restrictions are not set; whether channel settings alone block every path is pilot test 9
5. RStudio file download, plot export, console copy	A download-blocking option written for `gui_only` studies only	RStudio session logs shipped	Option is commercial-edition only and ignored by the free edition (source); browser copy of text has no control
6. Google APIs to an outside bucket	No internet route; VM service account limited to the study's buckets; perimeter planned	Data Access logs (if the call reaches Google)	Depends on the perimeter being enforced
7. The release path itself	Steward review and release record	Read alert; no log of what was released	Steward error, collusion, or a model or table that passes review but discloses
8. Screen, camera, retyping	DSP signature, banner	None	Cannot be blocked technically, as the plan says
9. Administrator access	`gui_only` excludes administrators from the study path; sudo alert and weekly review	Admin Activity and sudo log	Administrators are trusted, with review after the fact

the laptop through the same SSH channel with ordinary utilities. The plan does not claim otherwise. Audit coverage of copy activity is partial, and the weekly review has to find the entries because no alert uses them (Chapter 10).

A log of file copies is evidence for an investigation and a deterrent for people who read the banner. It is not prevention. D12 says as much, and the question for a data owner is whether prohibition plus detection satisfies its terms, which is a question the intake checklist needs to ask for every DUA.

7.3.2 The xrdp desktop

xrdp reads channel settings from its server configuration, and the planned configuration turns the clipboard, drive, audio and related channels off at the channel level. The design expects that the client's own settings cannot restore the channels, and pilot test 9 checks whether the channel setting blocks the clipboard in both directions. An AI-generated research report recommends a more granular approach: since xrdp 0.9.19, the release notes say that both inbound and outbound clipboards can be restricted for text, files or images [125], and `sesman.ini` has `RestrictOutboundClipboard` and `RestrictInboundClipboard`, each accepting `all`, `none` or a list of `text`, `file` and `image`. It also has `EnableFuseMount`, which when set to false disables drive redirection and copying of files [126]. The report's suggested configuration, outbound clipboard off and inbound on, is a usability trade: it would let a researcher paste a code snippet or a schema into the desktop (Chapter 9) and still block anything leaving by clipboard.

The plan blocks the clipboard in both directions at the channel level and sets none of the `sesman.ini` options. On the version: the image takes xrdp from EPEL, EPEL 9 carries xrdp 0.10.6.1 as of 2026-11-01, and its `sesman.ini` manual documents all three options, so the granular settings would be available to a build made today. The build does not pin the version, so the image record should log it. (This assumes the package mirror resolves; Chapter 8 discusses why mirror configuration must be checked against the upstream's actual layout.) The granular clipboard restrictions are not set; whether channel settings alone block every path is *not verified*, and exactly what pilot test 9 is for. Setting both layers (channel off, granular restrictions on, FUSE mount off) costs one file edit and gives two independent settings for the pilot to test.

7.3.3 RStudio and the `gui_only` lockdown

`gui_only` is the plan's answer for studies whose DUA forbids copy-out. In the planned design, a non-administrator's SSH command is replaced by the GUI-password helper, local port forwarding is kept for the tunnel, and, for these studies, RStudio's session configuration is set to block file downloads and uploads. The design itself marks the option names as to be verified on open-source RStudio in the pilot.

The verification can be done by reading. Posit documents `allow-file-downloads` and `allow-file-uploads` in the Workbench `rsession.conf` reference, its commercial server product [127]. The open-source session code contains a check that, for options in this family (including `allow-shell`, `allow-file-downloads` and `allow-file-uploads`), logs a warning that the option "was set, but it is not supported in this edition of RStudio and will be ignored" [128].

The `gui_only` lockdown of RStudio file download does nothing on the free edition

The two options the design relies on are Workbench options. In the open-source build the accessor for each of these options returns true whatever the file says, and the session logs that the option is ignored [128]. In `gui_only` mode the SSH route is closed and the xrdp channels are off, but the Files pane download, the plot export and the built-in terminal in the browser would remain available on RStudio studies. The plan's claim for `gui_only`, "no file download," therefore holds for the desktop and not for

RStudio. Pilot test 10 would confirm it. Until then, a DUA that forbids copy-out should not be placed in `gui_only` with RStudio on the strength of the option.

Two alternatives would close the gap without buying Workbench. The first is to disable RStudio for `gui_only` studies and use the desktop only. The second is to keep RStudio and treat its download button as part of the accepted residual risk, stating that in the DSP. The first costs the researchers a convenient interface (Chapter 9) and the second weakens what `gui_only` promises. A third path, a front end that wraps RStudio in a proxy refusing download requests, adds a component the plan deliberately avoided. Chapter 14 carries the choice into the risk register. The pilot's third study sets `gui_only` with both RStudio and the desktop; it is the study that would exercise this.

7.3.4 The analog hole

No technical control stops a person reading a figure from the screen and typing it elsewhere, or photographing the display. The plan states that neither mode stops this and that the DSP signature carries it. What a steward-reviewed output path does, and a no-review path does not, is give a legitimate route for results. The practical measure is how fast the legitimate route is. The plan sets no turnaround target for releases, and a slow steward queue is a reason for people to stop using it.

7.4 Checking outputs

7.4.1 What the plan asks the steward to check

The release record (a template for one form per release) has six checklist items: aggregate or summary results only, unless the DSP or DUA allows record-level data; no direct identifiers, expressed as the HIPAA Safe Harbor list of 18 categories; small cells suppressed below a threshold the record leaves blank, with a default of 11 unless the DUA says otherwise; no free-text fields that could hold identifiers; figures and tables that cannot be combined to re-identify a person; and a destination and purpose allowed by the DSP or DUA. The decision is "released" or "returned for changes" with a signature and a date.

This is the Five Safes "safe outputs" dimension in its simplest form (Chapter 2) [42]. Where output checking is a staffed function, the rules are more detailed.

7.4.2 Rules in use elsewhere

Table 7.2 sets the plan's checklist beside rules I could read at their source. The Office for National Statistics (ONS) Secure Research Service sets a threshold of 10 data subjects: unweighted counts below 10, and statistics derived from groups below 10, must be suppressed, and "counts ranging from zero to nine, inclusive, are not permitted." It also forbids record-level data (a threshold of 1) under any circumstances, forbids class disclosure and dominance, and requires that suppression cannot be reversed by differencing [129]. Its guidance defines dominance in two ways: the largest entity above 43.75% of the total, and/or the entities other than the largest two below 12.5% of the largest. It adds that other trusted research environments may use other definitions and that dominance is rare and hard to spot at the checking stage [129].

CMS states a different rule. Its cell size suppression policy says no cell containing a value of 1 to 10 can be reported directly, a value of zero does not violate the policy, and no cell can be reported that allows a value of 1 to 10 to be derived from other reported cells, including through percentages [130]. It applies to documents (manuscript, table, chart) made with CMS research identifiable files and limited data sets. The U.S.

Department of Education's disclosure avoidance material names minimum cell size and controlled rounding as the main tabular methods and leaves the size as an unspecified "X" [131, 132].

Table 7.2: Output-checking rules, with the plan's release checklist beside them. ONS and CMS values are quoted from their documents; the plan's default of 11 is its own choice and is not taken from either. "None" means the checklist has no corresponding item.

Rule	ONS Secure Research Service (Feb. 2025) [129]	CMS and other sources	Plan's checklist
Small cells	Unweighted counts and derived statistics below 10 data subjects suppressed; zeros 0 to 9 included	CMS: cells with 1 to 10 not reported, zero allowed [130]; Education: minimum cell size "X" [131]	Counts below a blank; default 11 unless the DUA says otherwise
Secondary disclosure (differencing)	Required: suppression must not be reversible by totals or differencing	CMS: no cell derivable from other reported information	One generic item (figures and tables cannot be combined)
Dominance	Largest entity above 43.75% of total, or all but the top two under 12.5% of the largest, not permitted	Not in CMS text read	None
Record-level data	Never permitted in outputs	DUA-specific	"Aggregate only unless the DSP or DUA allows"
Direct identifiers	Covered by general rules	HIPAA Safe Harbor: 18 categories, plus no actual knowledge of re-identification [133]	Safe Harbor list copied into the form
Student data	Not applicable	34 CFR 99.31(b): release after removing PII and a reasonable determination, "whether through single or multiple releases," taking account of other reasonably available information [67]	None
Cumulative releases	Not stated in the rules I read	FERPA text above covers multiple releases	One record per release; no running log
Code, logs, notebooks, free text	Rules by file type; output must advance the project	Not addressed	Free-text fields only
Trained models	Not covered	GRAIMATTER and related papers (next section)	None

Our default threshold is stricter than ONS in one respect (it suppresses 10, which ONS permits) and compatible with CMS, which suppresses 1 to 10. Read literally, "counts below 11" would also suppress zeros, which the ONS rule requires and CMS does not; the form should say whether zeros are in or out, since the two data providers it might meet disagree. The HIPAA Safe Harbor line is correct in intent but simplifies the regulation. Section 164.514(b)(2) removes names, geography smaller than a state with a limited exception for the first three digits of a ZIP code in populous areas, all date elements except year, ages over 89 (aggregable into one category), and fifteen further categories of identifiers, and then requires that the covered entity has no actual knowledge that the remaining information could identify a person [133]. The passage I read gives identifier categories and an expert-determination alternative, and does not state a count threshold for aggregate tables. The count thresholds come from data providers and practice, not from the Safe Harbor list.

The second gap is for student data, which is P3 by default and P4 in several cases (Chapter 11). The

checklist cites only HIPAA. FERPA studies in the plan would hold student education records, and 34 CFR 99.31(b) allows release of de-identified records only after a reasonable determination, taking into account other reasonably available information and whether disclosure arises through single or multiple releases [67]. A per-release form with no running log cannot show that the cumulative releases from one study stay under that test. The third gap is dominance, which the form has no item for and which would matter for any table of money, hospital charges or institutional counts.

7.4.3 Tooling and the human checker

The plan assumes a person reads each file. Output checking in trusted research environments has produced a body of tools for assisting that person. The SACRO toolkit, as its paper's abstract describes it, adds a Python package over common analysis tools that identifies potentially disclosive outputs against commonly used tests, applies optional mitigation, reports the reasons, and produces summary documents for staff; it supports Python, R and Stata and is designed to assist human checkers, not replace them [134]. A consensus statement from the same project sets principles for using automation in output checking [135]. I did not install or test either, and I read the consensus statement only at record-description level.

For this plan the relevant point is that the researchers use R and Python on the VM, and a package that runs the common disclosure tests while they work would give the steward a log of what was checked, matching the plan's own preference for a record. The software chapter (Chapter 8) is where a request to add such a package would be handled, and since the VM has no internet it would arrive through the curated-container route.

The release path is a procedure, not yet a control

The release path would work as a procedure and not yet as a control. The steward's authority is the form, the form is a template whose default threshold is the plan's own choice, and the evidence that a release was approved is a paper record that the campus security office matches by hand to a read alert. What is missing is a set of definitions, since the machinery exists: who may be a steward (on the DSP, trained for that data, not the person whose paper is being released); where review happens (ideally in place, so unreviewed output does not travel to a steward's laptop); what each DUA's cell rule is, copied into the record at intake; and a running log of what has gone out. None of this needs new infrastructure. It needs the template and the intake checklist to say it.

7.5 Trained models: the export gap

The plan's pilot includes a GPU study, and the GPU template (Chapter 13) exists so that researchers can train models on sensitive data. The release record has no line for a model. Its first item asks for "aggregate or summary results only," and a set of trained weights is neither. A researcher who trains a model on P4 data and wants to share it would be asking the steward to approve a file that no checklist item describes.

The literature says why this is hard. A 2021 paper on disclosure of machine-learning models from trusted research environments states that few such environments support automated model building, that trained models may leak personal training data once released, that the extent is not well understood, and that guidelines for safe disclosure of models did not exist in these environments [136]. The GRAIMATTER green paper, funded under a UK DARE sprint, offers draft recommendations for disclosure control of trained models and says that trusted research environments "do not have mature processes and controls" against the added risks [137]. I did not review the recommendations themselves, and they are drafts that their authors expected to evolve.

Without the technical detail, the plan can still set a policy at the right level. A model is a release of

derived data from the training set, and a steward cannot judge its risk by reading the file. Options, in order of increasing permissiveness: release no weights (only evaluation metrics and predictions on non-sensitive test data); release weights only to recipients inside the DSP, so that the model never leaves the environment's rules; release weights from a model of a type and training setup that the data owner has approved in writing; or release weights after a documented attack-based assessment of the kind the GRAIMATTER recommendations describe. Each depends on what the DUA allows, and the intake checklist is the place to ask. The same reasoning applies to a fine-tuned open-weight language model; Chapter 13 treats those. Intermediate checkpoints, optimizer states, embeddings and saved datasets are in the same class, as are R objects and notebooks that embed the data they were computed from. A steward given the current checklist has no basis to refuse any of these, or to accept them.

7.6 End of study: soft delete and destruction

The planned teardown procedure is written for IS-3 disposal and for the DUA obligation to return or destroy data, and it is run by the systems team under a change ticket. It lifts deletion protection, deletes the VM and disks, deletes every snapshot carrying the study label, removes every object including noncurrent versions, deletes the buckets and the study's service account, and writes an evidence record with the inventory before and after and a statement that nothing remains when no labeled resource is left.

Cloud Storage soft delete is enabled by default for all buckets that support it, with a default retention of seven days, during which a deleted or overwritten object cannot be permanently deleted and can be restored [89]. The planned configuration sets no soft-delete policy, so a study's three buckets would use Google's default. The teardown procedure does not mention soft delete and does not check for soft-deleted objects. Whether a recursive all-versions delete bypasses the soft-delete period, and whether the lifecycle rule's deletion of noncurrent versions also enters it, I did not verify; the page I read describes the mechanism but I did not test either command.

Teardown could report that nothing remains while soft-deleted copies of study data still exist

The procedure's closing check lists resources by study label, and soft-deleted objects are not labeled resources. With a default seven-day period, deleted data from a P4 study could remain restorable for a week after the evidence record says nothing is left. The same applies to the 30-day lifecycle rule meant to remove old object versions (the second panel's finding that old versions are PHI copies): a pilot test that confirms an overwritten version is gone after the window would pass while a soft-deleted copy lingers. The fix is a bucket-level soft delete policy of zero days for study buckets, or an explicit wait and re-check step, and a line in the evidence record that states the soft-delete setting.

Whether seven days matters depends on the DUA's wording. The U.S. Department of Education's Privacy Technical Assistance Center (PTAC) guidance on data destruction states that deletion of a digital record is typically insufficient because the data are preserved and can be undeleted, and that file deletion, disk formatting and "one way" encryption are not effective ways to dispose of sensitive data [138]. The guidance classifies sanitization as clear, purge or destroy under NIST SP 800-88. A cloud customer cannot physically sanitize the provider's media, so what a DUA's destruction clause requires of a cloud study is a question for the data owner and counsel; I make no claim about what any clause says. The conservative reading is that a recoverable-for-a-week state is a different thing from the destruction the clause describes, and the teardown evidence should not describe it as the same. Chapter 6 discusses customer-managed keys; because the same guidance lists "one way" encryption among the ineffective methods, key destruction would not settle the question without the data owner's agreement.

The script retains the audit logs about the study, as it states, which is the right choice for the log project and a conflict to resolve in the DUA if the agreement requires destruction of "all derived records" (Chapter 10). The evidence record sits in the Terraform state bucket, which is not locked (Chapter 10). And the release records for each study are paper or file artifacts with no stated retention; they are the only proof of what left, and should be filed with the DSP.

7.7 Residual-risk acceptance

D12 asks the security officer to accept the copy-out risk of shell mode. The plan names the people who are the risk (authorized users), the routes (Table 7.1) and the mitigations (prohibition, logging, the banner). For acceptance to be informed it needs more.

It should be per study, not per platform. A DUA that forbids copy-out cannot be satisfied by a platform-level acceptance that says copy-out is possible and prohibited; that study needs `gui_only`, with the RStudio finding resolved. A DUA that is silent can use shell mode. The DSP template should have a field for this, and the plan's DSP-to-module table already maps "DUA forbids copy-out" to `access_mode = gui_only`.

The acceptance should also state the detection capability, because a prohibited-and-logged route that nobody reviews is not a mitigation. Chapter 10 found that the copy-out audit records have no alert, and that the review step exists only in the weekly runbook. The acceptance record should say what is searched, how often, and by whom.

And it should state the review conditions: the accepted risk would be revisited after the first incident, after any change to the allowed routes, and at the annual platform review. These conditions are not in the plan today.

Our strongest mode rests on other people's defaults

Three controls on the data path depend on defaults of other people's software, not on settings the plan makes: scp's use of SFTP since OpenSSH 9.0, the xrdp channel behavior of the installed version, and RStudio's treatment of Workbench options. Reading the sources settles two of them: the OpenSSH switch is real but depends on the client's version, and open-source RStudio ignores the download options. The xrdp version from EPEL supports the granular settings, but whether the channel settings block the clipboard is still for pilot test 9. Together they decide whether the `gui_only` promise holds. What the plan has not yet done is confirm that its strictest mode does what the design document says.

7.8 Design considerations

1. Add a storage ingress rule for study members and stewards to the perimeter design, or confirm in the dry-run week that none is needed, before the perimeter is enforced. Test the laptop upload and the steward download specifically.
2. Define the ingress procedure: where files arrive from the data provider, who may touch them, a checksum or inventory step on the inbound bucket, and when it is cleared after loading. Treat the provider-to-laptop hop as part of the data path in the DSP.
3. Decide whether Globus is in scope. Establish whether UCR holds a high assurance subscription and the BAA with the University of Chicago, and whether an endpoint can run without an internet route. Record the result as a decision, not an assumption.

4. Fix the `gui_only` claim for RStudio. Either disable RStudio in `gui_only`, or state in the DSP that RStudio download remains open, or test an alternative. Run pilot test 10 against RStudio first.
5. Set xrdp restrictions in two layers: channels off in `xrdp.ini` and the granular clipboard and `EnableFuseMount` options in `sesman.ini`, and record the installed xrdp version in the image record. Decide, with researchers, whether inbound paste is allowed.
6. Extend the release record with dominance, differencing and zero handling; with FERPA's reasonable-determination test for student data; with a running log of prior releases for the study; and with the data owner's cell rule copied from the DUA at intake.
7. Require a trained model, checkpoint, embedding or saved object to be named in the DSP as a possible output, with a default of no release and an approval path. Treat GRAIMATTER-type recommendations as a starting point to be read in full before the pilot's GPU study produces anything.
8. Specify the steward: on the DSP, trained for the data, independent of the author of the output, and reviewing in place where possible. Link the release record to the object (name and generation) and to the read alert.
9. Set soft delete for study buckets to match the DUA's destruction terms, extend the teardown script to check and record it, and have the evidence record state what it does not cover.
10. Write the residual-risk acceptance per study, with the detection method and review frequency stated, and ask the data owner in each DUA whether prohibition plus detection satisfies copy-out terms.
11. Measure the release path in the pilot: time from request to release, and the number of returns for changes. A slow route teaches researchers to look for another one.

Chapter 8

Software Without Internet and Supply-Chain Risk

A virtual machine with no default route cannot run `pip install`, `install.packages()` or `conda install` the way a researcher's laptop does, and almost every analysis workflow in genomics, survey statistics and machine learning begins with one of those commands. The plan removes the internet from the study VMs and replaces it with a short list of internal software sources: Artifact Registry remote repositories for Python, Docker Hub and three Rocky and EPEL RPM repositories, one standard repository for curated containers, and packages baked into VM images by a builder that sits outside the perimeter. The design is sound as plumbing. Whether it is sound as a security control depends on a second question that plumbing does not answer: what does the environment admit when a researcher asks for software, how fast, and who looked at it first?

That question matters more in 2026 than it seemed to when we chose live mirrors. Public package registries have become a target for credential-stealing code that runs when a package is imported, and research packages are among the victims. I read our software path against the primary records of two incidents I could verify (a compromise of the PyTorch Lightning packages on PyPI and a deserialization flaw in the R language), against the documentation of the tools we use and the ones we did not choose, and against the small research literature on package-malware detection. Chapter 4 describes the Artifact Registry features and Chapter 5 places the software path in the architecture.

8.1 What the plan specifies today

The plan's software section is a table of six needs and how each is met. Operating-system packages come from `dnf` against Artifact Registry remotes for Rocky 9 BaseOS, AppStream and EPEL 9. Python comes from `pip` against an Artifact Registry PyPI remote, configured system-wide at image build. Container pulls go through a Docker Hub remote, with `apptainer` installed from EPEL and a login profile that points users at the Docker Hub remote and the curated repository. R, CRAN and Bioconductor are not mirrored at all: the plan ships R in the image, provides a curated container `p4-r`, and treats a new R package as a request to the research computing team, which would rebuild the container outside the perimeter. PyTorch with CUDA is handled the same way (`p4-torch`), because the PyTorch wheel index is not among Artifact Registry's preset upstreams, and conda and bioconda tools go into `p4-bio`. The request path is intake to the research computing team, a rebuild, a push to the curated repository and a target turnaround that the plan leaves as "TBD, e.g. 2 business days".

The offline Terraform plan contains six Artifact Registry repository resources: the PyPI remote, the Docker Hub remote, three Yum remotes and the standard curated repository. There is no virtual repository, no cleanup policy, no repository-level IAM resource and no Artifact Registry resource beyond those six. The plan records the upstream-access step (the registry's VPC Service Controls allow setting) and marks it to be verified in dry run; Chapter 4 covers that step.

The opposite situation holds on an open shared HPC cluster: users can install anything (`uv`, `pixi`, container pulls from Docker Hub), which means there is no change control on software that would touch P4 data, against the IS-3 expectation that software installs and configuration changes are approved through

change management. The software path in the plan is the answer to that problem. Its test is whether it gives the campus security office an approval point that means something.

8.2 Why the live mirror is the weakest part of the design

An Artifact Registry remote repository is a pull-through cache. Google's documentation says the first request for a package version makes the registry download it from the upstream and cache it, and later requests for that version are served from the cache; a request for an artifact the upstream lacks fails [139]. Two properties follow from that description. First, the repository does not decide what is admissible: any name and version that exists upstream at the moment of the first request is eligible. Second, the first researcher to ask for a new release is the one who causes it to be fetched. The pages I read (the remote-repository overview, the creation guide and the virtual-repository overview) document no minimum release age, quarantine period, allowlist or malware-scan step for remote repositories [86, 139, 140]. That is a statement about what those pages say, not a claim that no such feature exists. The systems team should ask Google, and the answer belongs in the campus security office's review.

Google does document controls that narrow the exposure. Virtual repositories give clients a single endpoint, and the documentation says they can help prevent dependency-confusion attacks by letting an administrator give a private standard repository priority over a public remote [140]. A Connector repository mode (Preview) proxies without caching and is described as useful where full auditability of the upstream is required [86]. Inside a VPC Service Controls perimeter, the registry's calls to upstream sources are denied by default; the project-level setting has two values, deny and allow, and an unset value means deny [141]. That last property is useful: it means the live mirror's reach to the internet is an explicit, auditable administrator decision, and that the decision can be taken per project and per location. It does not limit which packages the upstream can serve once the setting is allow.

- *Formats.* Artifact Registry lists Python, Docker, Yum, Apt, npm, Maven, Go, Ruby and Conda among its formats, with Ruby and Conda marked Preview, and its preset upstreams for remote repositories do not include CRAN or Bioconductor [139, 142]. The plan's statement that CRAN, Bioconductor and conda are outside what remote repositories cover therefore matches Google's pages; whether a custom conda upstream is possible in Preview was not established.
- *Preview status of the Yum remotes.* The RPM configuration page marks use of a remote Yum repository as Preview, subject to Google's pre-GA terms [143]. Google's HIPAA guidance says not to use pre-GA offerings with PHI unless a notice says otherwise [73]. No PHI would pass through a package cache, so the rule may not apply, and the exposure to the Preview terms is "as is" support with no service commitment. The security office should record whether it accepts a pre-GA dependency in the patching path of a PHI system; that is a policy call, not a technical one. Section 8.8 proposes a way to remove the question.
- *Authentication.* Google's Python guidance describes access tokens, the keyring library or password authentication for Artifact Registry Python repositories, and says tokens are valid for 60 minutes [144]. The planned client configuration names only the index; the build installs no credential helper, and the planned configuration grants no study identity read access to the repositories. Every mirror needs a named reader identity and a credential path, or the first pull fails. Whether the VMs can pull from the mirrors, and with which identity, is something the first run of pilot test 3 ("mirrors work", Section 5.10) would settle. I list it in the design considerations because the offline plan would never show the failure.

8.2.1 What a live mirror does not defend against

A live mirror is close to the opposite of a quarantine. When an attacker publishes a malicious release of a popular package using stolen maintainer credentials, the mirror serves it to the next researcher who runs `pip install -U`. When an attacker registers a package name that a coding assistant tends to invent, the mirror resolves that name because it exists upstream. When a legitimate package is yanked or fixed, whether the cached copy persists is a property of the registry that the pages read do not state. The incidents in the next section show the first of these cases at a scale that matters to research computing.

8.3 Two incidents that can be verified

One AI-generated research report on software supply chains that I consulted describes a wave of attacks on research packages in 2025 and 2026, with named actors, campaigns and package counts. Most of those details could not be confirmed against primary records and are not repeated here as fact. The two incidents below rest on records I checked directly on 2026-11-01. Table 8.1 separates what was confirmed from what was not.

Table 8.1: Incidents used here: what the primary records establish and what remains unverified.

Incident	Confirmed in a primary record	Not confirmed
PyTorch Lightning packages on PyPI, versions 2.6.2 and 2.6.3	OSV lists the compromise as PYSEC-2026-508 (aliases CVE-2026-44484 and a GitHub advisory) with affected versions 2.6.2 and 2.6.3 and a fixed version 2.6.4, and as MAL-2026-3201 for the `lightning` package [145, 146]. The vendor advisory text inside the record is dated 2026-04-30 and recommends pinning 2.6.1 and rotating credentials. PyPI's release listing shows 2.6.1 uploaded 2026-01-30 and 2.6.4 uploaded 2026-05-20, and no 2.6.2 or 2.6.3 [147]. A vendor write-up dated 2026-04-30 describes code in `__init__.py` that runs on import and downloads a JavaScript runtime [148].	The upload timestamps of the malicious files (removed from PyPI), so the length of the exposure window is unknown. The attribution to named groups and the details in the AI-generated report beyond what the OSV record and the vendor write-up state.
R deserialization flaw CVE-2024-27322	OSV: arbitrary code execution when a crafted RDS file or R package is interacted with, on R from 1.4.0 up to but not including 4.4.0 [149].	The report's severity score and its description of package metadata files as an attack path; the OSV text names "R package" but does not describe the mechanism.
Bioinformatics PyPI packages in a June 2026 wave	OSV malware records for `ensmallen` 0.8.101 and `dynamo-release` 1.5.4 (and three other packages in the report's list) describe an infostealer executed with a downloaded JavaScript runtime at Python startup [150, 151].	The report's count of 19 packages and 37 wheels. A vendor URL in OSV's references describes six bioinformatics packages. The OSV malware record is dated 2026-06-06 while a second OSV entry says the malicious releases were published 2026-06-08; the dates disagree and were not resolved.

8.3.1 The Lightning compromise

The OSV malware record for the `lightning` package says the compromised versions inject code that starts automatically when the module is imported, downloads a legitimate JavaScript runtime, and runs an infostealer.

It lists credentials from files, process memory, cloud metadata endpoints and command-line tools such as `gh` and `gcloud`, and says the code tries to spread using whatever credentials it finds [146]. The vendor write-up adds that the injected code lives in the package's `__init__.py`, starts a background thread before any legitimate code loads, and that the stolen data was encrypted and sent to public GitHub repositories [148]. The report's claims about a bundled JavaScript runtime are thus supported; the actor names are not.

None of what follows depends on the unverified details. The malicious code ran on import, not at install time, so controls that only block build scripts at install (such as pip's option to forbid source distributions, which exists to avoid running arbitrary code during installation [152]) would not have stopped it. The target list includes cloud metadata endpoints and `gcloud`, which on a Google Cloud VM or builder means the service account token. And the fixed version arrived 20 days after the advisory date, which means a researcher who pinned to the last clean release (2.6.1) was safe while one who ran an upgrade was not.

The Lightning compromise would have reached a live mirror's next user, and the plan's builder is also in the blast radius

The compromised releases executed on import and went after cloud credentials. With live remote repositories, the first `pip install` that resolved to 2.6.2 or 2.6.3 inside the enclave would have fetched and cached it. The plan's `p4-torch` container is built outside the perimeter by the research computing team; a build during the exposure window would have run the code on the builder, where a project service account with push rights to the curated repository is the credential most worth stealing. No document I read gives the length of the exposure window, so I can't say whether a seven-day minimum age would have prevented either case. I can say that no control in our plan would have.

8.3.2 The R flaw and the version pin

OSV's record for CVE-2024-27322 says R is affected from 1.4.0 up to but not including 4.4.0 [149]. The consequence for the plan is a version floor: the image and every R-bearing container must ship R 4.4.0 or later. The planned build installs R from EPEL, and a query of the Fedora package database on 2026-11-01 reports R 4.6.1 in the EPEL 9 release repository [153], so a build today would satisfy the floor. The build does not check, and the curated `p4-r` container has no recipe yet, so the floor holds only because of what EPEL happens to carry today. The exposure is wider than the interpreter, too: R stores package metadata and saved objects in serialized form, researchers routinely receive `.rds` files from collaborators and data providers, and the CVE text covers an RDS file as well as a package. The floor matters most for exactly the files a P4 study would handle.

Three additions the two incidents call for

Our software path answers the IS-3 change-control finding with mirrors and containers but does not yet say what makes an admission decision. The two verified incidents suggest three concrete additions: an automated assertion that every image and container ships R 4.4.0 or later (a failing build, not a note in a document); an admission rule for Python packages that does not depend on the first researcher's `pip install`; and separate credentials for building and for promoting artifacts into the curated repository. None of this is in the planned configuration or build today.

8.4 What scanners can and cannot do

An obvious response to the incidents is to scan packages before they enter the enclave. The AI-generated report proposes a pipeline with vulnerability databases, behavioral heuristics and signing, run in a connected

staging zone. Scanning of known vulnerabilities is mature: Posit Package Manager, for instance, reports known vulnerabilities for CRAN, Bioconductor and PyPI packages from the OSV database, can block affected packages, and says plainly that it does not itself scan for vulnerabilities, relying on OSV [154]. A vulnerability feed tells a staff member that a package has a known advisory. It does not tell them that a package published yesterday is malicious, which is the situation of the first hours of the Lightning case. Malware detection for new releases is a different problem, and the recent literature gives it a measured answer.

Three papers bear on it. CHASE, a multi-agent LLM system for dissecting malicious PyPI packages, reports 98.4% recall and a 0.08% false positive rate on a dataset of 3,000 packages (500 malicious, 2,500 benign) with a median analysis time of 4.5 minutes per package [155]. PYPILINE reports 96.7% precision, 99.6% recall and an F1 of 98.1%, an improvement of 5.7 to 21.6 percentage points in F1 over baseline tools [156]. MTPkgScan takes the opposite approach: it generates behavior-preserving variants of malicious PyPI packages and records a fault whenever a scanner flags the original and misses the variant [157]. The abstract describes the method and gives no rates, so all I can say is that the authors built a framework because such misses occur, and that a scanner's score on known samples is not its score against an adversary who rewrites the sample.

Those datasets are 17 percent malicious (CHASE's 500 of 3,000), while a real registry feed is overwhelmingly benign. The figures are reported results on the authors' datasets and are not a forecast for UCR's volume. As arithmetic, a 0.08% false-positive rate applied to 10,000 benign releases flags about eight of them for human review; whether a queue of that size is tolerable depends on volume that UCR has not measured. The point for design is narrower than the numbers: detectors raise the cost of getting malicious code through, a human or an age gate still decides, and a detector's recall on yesterday's attack is an unreliable guide to tomorrow's.

8.4.1 Slopsquatting

A separate risk comes from the researchers' own tools. A 2026 replication of a USENIX Security 2025 study reports that the earlier work found package-name hallucination rates from 5.2% for commercial models to 21.7% for open-source models; the replication measured 4.62% to 6.10% across five 2026 code models on 199,845 prompts, and identified 127 package names that all five invented, of which 53 remained registrable after the registries' defenses [158]. An attacker can register such a name and wait. Inside an enclave with a live PyPI mirror, a researcher who pastes an assistant's `requirements.txt` would have the registry resolve the squatted name. Inside an enclave where admission is by request, the same name would reach a human first. The study's finding is evidence about the pressure on a mirror-everything policy, not about UCR's researchers, who may use assistants less or more than the study's prompts.

8.5 Four policies compared

The options sit on a line from "anything upstream" to "only what staff built". Table 8.2 compares four of them on security, researcher friction and staff effort. The ratings are my qualitative judgments from the sources above and from the structure of each option. The plan's own figures for turnaround are targets, and the AI-generated report's turnaround and storage figures are generic benchmarks and model outputs that are not used here.

The options are not exclusive, and our plan already combines the first and last. But our weakest row (mirror everything) governs the ecosystem with the most frequent and most dangerous releases (PyPI), while its strongest row (curated containers) governs the ecosystems where Artifact Registry has no remote (CRAN, conda). The security ordering and the ecosystem risk ordering run in opposite directions.

Table 8.2: Four admission policies for software entering a P3-P4 enclave. Ratings are qualitative and apply to the plan's context.

Policy	How software enters	Security	Researcher friction	Staff effort
Mirror everything (the plan's Python and Docker Hub remotes today)	Any upstream package is fetched on first request by whoever asks	Weakest. No admission decision; a malicious release or squatted name is served as soon as it is requested. Reach to upstream is one auditable switch	Lowest. Researchers use `pip` and `docker` as on a laptop	Low to run; high on the day of an incident, when someone must find and purge what was fetched
Mirror with quarantine	A mirror (live or batch) holds new releases for a minimum age and optionally scans them before release	Moderate to strong against fast-detected attacks. Unknown detection latency sets the value of the window; does nothing for a package malicious from its first release	Low for older versions; waits for new ones. Researchers need a way to ask for an exception	Moderate. Needs a component that Artifact Registry's remotes do not document, built or bought, plus review of exceptions
Curated allowlist	Researcher requests a package; staff review, pin and import it with its dependencies	Strong. Every package has a named approver and a version	High. Each new package waits; dependency trees multiply requests	High and recurring. Staff time scales with the number of requests and the number of dependencies per request
Curated containers (the plan's `p4-r`, `p4-torch`, `p4-bio`)	Staff build signed images outside the perimeter; VMs pull images only	Strong for the contents of the image; weak if researchers can then install into a running container or if the builder is compromised	Low for common stacks; high for anything outside them. New package means a rebuild	Moderate. Builds, signing, scanning, rebuilds for each security release

8.5.1 What peers report

The peer evidence is thin and is covered in Chapter 2. Where peer pages describe software in enclaves, they describe ticket-based processes: one university's enclave takes software requests by ticket and reserves the right to reject software, and administrators keep rights on machines inside the enclave (Chapter 2, which cites the institution's own page). That description is of a process, not a quantified turnaround, and none of the peer pages checked published a measured turnaround for software requests. Our "2 business days" is therefore a target with no external anchor, and I treat it as a hypothesis for the pilot to test.

8.6 Tools for each ecosystem

8.6.1 R: Posit Package Manager or a curated container

The plan treats R as a container problem. The alternative that fits an R-heavy campus is Posit Package Manager (PPM). Its air-gapped mode is documented: a downloader runs on a system with internet access and copies CRAN, Bioconductor, PyPI and Open VSX data to the offline server, whose version must match the downloader's [159]. The documentation gives storage figures (at least 250 GB for a typical installation and at least 170 GB for CRAN alone, before binary packages) and warns that an update whose requirements file omits packages already served makes the repository stop serving them silently, so the file must be kept cumulative [159]. PPM evaluates CRAN, Bioconductor, PyPI and Open VSX each business day and publishes snapshots, and repositories can be pinned to a date for reproducibility [160]. The date pin is also an admission control: a repository frozen at a snapshot a few weeks old contains no release newer than that date, and the researcher cannot choose otherwise. PPM supports blocking by package name, version and license, with full blocking at the product's Advanced tier [154]. It is a commercial product; I did not establish UCR's price or whether campus already holds a license, and Chapter 12 would carry the figure.

The same documentation shows what PPM changes. A snapshot-pinned PPM inside the perimeter would give R users `install.packages()` without a ticket for anything in the snapshot, which a curated container cannot do, and it would age the whole CRAN universe by construction. It also adds a server to patch and a transfer process, and a package inside the snapshot is not vetted, only old. For a P4 environment the likely use is a snapshot a few weeks behind, with the R floor enforced and `.rds` files from outside the enclave handled as untrusted input.

8.6.2 Python: where an age rule can live

Two client tools document a minimum-age filter. uv's `exclude-newer` limits candidates to files uploaded before a date or before a duration such as `1 week` or `P7D`, comparing against each file's upload time, and it can be set at system level in `/etc/uv/uv.toml` [161, 162]. pip has an `--uploaded-prior-to` option that takes a datetime or a duration such as `P3D`, and the documentation says it is only effective for indexes that provide upload-time metadata [163]. That metadata is the `upload-time` field of the Simple API standardized in PEP 700, where the field is optional [164]. I did not establish whether an Artifact Registry PyPI remote passes upload time through to the client; that needs a test.

A client-side age rule is better than none and weaker than a server-side one: it lives in a file the researcher can override, whether by an environment variable, a command-line flag or a project file. It would be a useful second layer on a mirror that enforces its own policy; it is not an admission control by itself. pip also has a hash-checking mode that rejects any file whose hash is not listed, which is the right mechanism for a curated, locked environment [152].

8.6.3 Conda and bioconda

The plan puts conda and bioconda tools in `p4-bio`. A conda mirror is a configuration exercise in conda itself, which documents channel and default-channel keys for pointing clients at a mirrored location [165]. conda-store adds a service that builds shared environments from declarative specifications and, in its documentation, generates lockfiles and other artifacts so that a given specification can be reproduced [166]. Its server and worker components mean a new service to run and patch. For UCR's scale the mirror question is less pressing than the build question: who may add a conda tool to `p4-bio`, what pins it, and how a rebuild is triggered when a tool has a security release. Artifact Registry's own Conda format is Preview and was not tested for remote use [142].

8.6.4 Containers: signing and what Apptainer does and does not give

Apptainer's security documentation describes why it exists for shared systems: containers run as the invoking user, and by default runs without setuid using unprivileged user namespaces where the platform supports them [167]. Its signing documentation says SIF images can be signed and verified with PGP keys by default, that the signature travels inside the image file, that multiple signatures can document an approval process, and that administrators of privileged installations can restrict execution to signed images with an Execution Control List [168]. Two limits matter. Verification proves that an image is the one a signer produced; it does not say the signer's content was safe, so a signature means as much as the review that came before it. And the Execution Control List is described for privileged installations, while the plan's default would be an unprivileged one, so the enforcement path (a wrapper, a policy on the VM, or both) has to be designed. Pulling `docker://` images converts OCI layers into SIF on the VM [169], which means a researcher with a path to the Docker Hub remote can turn any public image into something runnable. The plan's login profile points users at the Docker Hub remote as a convenience, and the residual risk is that the convenience becomes the policy. If curated containers are the control, the VM-side path to the Docker Hub remote is a hole in it.

8.7 Weak points in the plan's own supply chain

The image build is where the enclave's software enters. Reading the planned build and configuration, I found these gaps:

- Direct downloads during the build. Vendor installers fetched over the network and run without a checksum or signature check give whoever controls that path code that runs as root on the builder and ends up in every image. Every direct download needs a pinned digest or a verified signature.
- Package signatures versus metadata signatures. Checking package signatures but not repository metadata is the right choice behind a proxy, because signatures are made upstream and a cache cannot forge them, but metadata is then unsigned in transit through the cache, so a compromised cache could serve stale metadata that points at older, vulnerable versions. The distribution's signing keys have to be present in the image; one AI-generated report makes the same point.
- The builder VM has internet access by design, in a separate project or a subnet with NAT. The Lightning payload went after cloud metadata credentials. The builder's service account is therefore the identity to minimize, and the person or process that pushes to the curated repository should not be the one that ran `pip install` on the open internet.
- There is no signing step, no software bill of materials, no digest pinning for `p4-*` containers and no record of what was in an image beyond the CIS scan report the plan keeps. A curated repository

referenced by mutable tags lets an image change under the same name.

- Mirror paths must match the upstream's real layout. A remote repository's path is appended to the preset upstream base [86], so a path that repeats part of the base resolves to nothing. Because `dnf` treats an unreachable enabled repository as an error by default, one wrong mirror path can break every later install in the build as well as updates on study VMs. A metadata fetch against each remote belongs in the dry run, and the item is carried to the revision list (Chapter 14).
- Builder access across the perimeter. If the build switches to the secure project's mirrors before it finishes installing, a builder outside the perimeter must read from inside it, which needs both a read grant and, under an enforced perimeter, an ingress rule. Either keep the public repositories until the end of the build, or grant and test the path.

Each of these is a design gap in a build that has never run; none is an observation about a live system.

8.8 A recommended policy for UCR

I recommend five changes and a decision gate, for review by the campus security office and the systems team; the research computing team would estimate the staff effort. Figure 8.1 shows the planned software path and the proposed additions.

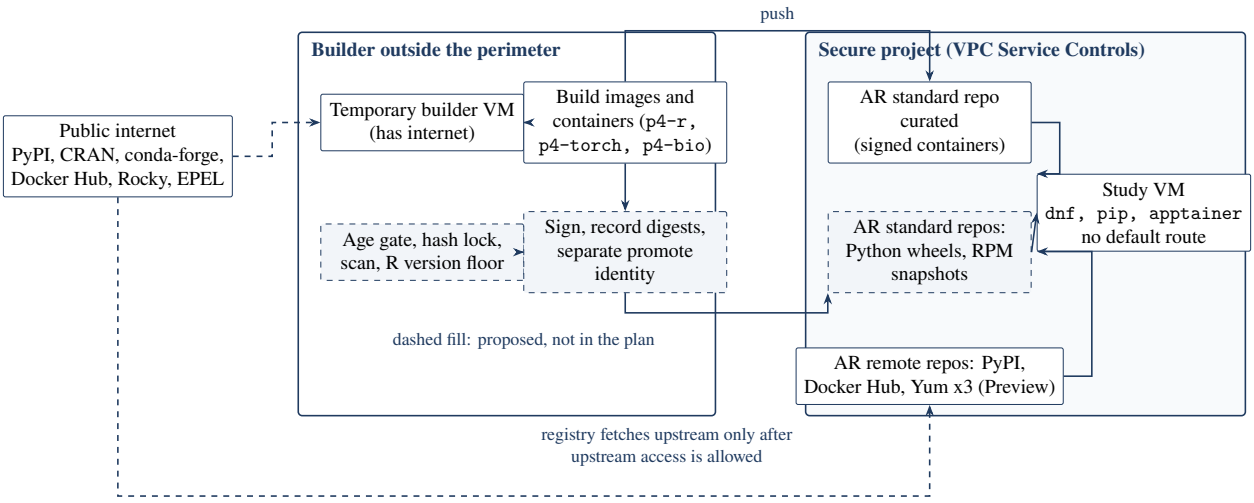


Figure 8.1: The planned software path (solid boxes) and the additions proposed below (dashed fill). In the plan, the PyPI, Docker Hub and Yum remotes reach the internet directly through the registry once upstream access is allowed; the proposal moves Python and OS packages to curated standard repositories filled by the builder and keeps the remotes as a fallback until the pilot decides.

1. A version floor that fails the build. The image build and each container build should run `R -version` and fail if the result is below 4.4.0 [149]. A matching boot-time or login-time check would catch an image that was rebuilt from an older base. Researchers should be told that RDS files from outside the enclave are untrusted input regardless of the version.

2. Python through an admission rule, not a live remote. The plan's PyPI remote should not be the default path for P4 studies. Two designs meet the aim. In the first, the builder resolves each study's dependency

request into a hash-locked set of wheels, applies a minimum age, and imports the files into a standard (not remote) Artifact Registry Python repository that VMs read. In the second, the remote stays and a system-wide client-side age rule is added for pip and uv, which is weaker for the reasons given above and depends on a test of upload-time metadata. The first is more work and gives an approver and a record per package; the second is a stopgap. The age should be a parameter chosen by the campus security office with the detection-latency question in view: in Chapter 14 I suggest 7 to 14 days, which is my own starting point and not an evidence-based value. A researcher who needs a newer version asks, and the research computing team reviews the release notes and the diff before it is admitted.

3. OS packages from snapshots, which also removes the Preview question. The patching path currently depends on three Yum remotes marked Preview. A standard Yum repository filled by the builder from Rocky and EPEL snapshots would give the same `dnf` experience from a repository type that the RPM page does not mark as Preview in the text I read (the status of standard Yum repositories should be confirmed with Google), and it makes the patch cadence an explicit decision: each snapshot is imported, tested on the pilot VM and then released, which is also what IS-3 expects of software change control. If the campus security office accepts the Preview terms for a package cache that carries no PHI, the remotes can remain; the decision should be recorded in the Risk Treatment Plan with its basis [73].

4. Signed containers with VM-side enforcement. Every `p4-*` image should be signed on a system separate from the builder, referenced by digest and not by tag, and accompanied by a bill of materials and the scan results. The VMs should verify before running. The VM-side path to the Docker Hub remote should be closed for P4 studies (the login profile's pointer to it would go), so that curated containers are the only container source. If a study needs a public image, it is requested like any other package.

5. A request process with a published service target and a measured one. The plan has intake and a target. It needs a ticket type for each class of request (new R package, new Python package or version, new container, OS package, model weights per Chapter 13), a named approver per class (the research computing team for R and Python, the campus security office consulted when egress would change), a default admission test (age, license, maintainer history, diff against the prior version, scan result), and a record that ties each admitted artifact to the request. Turnaround should be measured from the first pilot day: the pilot already includes a time measurement for a 200 GB upload and none for software requests. The target in the plan should be replaced by the pilot's measured median and 90th percentile.

Decision gate. Before the first PHI study, the pilot (fake data) should show: the VM can pull from each configured source with its own identity; upstream access is denied before the allow step and a pull works after it; a package newer than the age floor is refused; a deliberately unsigned image fails verification; an image with R below 4.4.0 fails the build; and a yanked package that was cached is removed from the mirror within a stated time.

The admission rule should not depend on who asks first, and no scanner supplies that

The scanner papers report strong results on their own datasets, and one exists specifically because scanners miss behavior-preserving rewrites. The two incidents were found after release, not before. A policy that gives every new release an age, every new name a human, and every admitted artifact a record does not depend on a detector's recall; a detector then raises the cost of getting past the policy.

For UCR the choice is between a research environment where `pip install` behaves as on a laptop and

one where a package request has a cost and an approver. Our design already accepts that cost for R and conda, where Artifact Registry has no remote, and avoids it for Python and Docker Hub, where it has. The recommendation reverses that asymmetry for P4 studies: the ecosystems with the most frequent releases and the best-documented 2026 incidents get the stronger gate. The price is researcher friction on new Python packages. Whether the plan's turnaround target can hold for the stacks UCR researchers use is an empirical question that the pilot answers and that the campus research advisory body should see before the policy is fixed.

8.9 Request process and turnaround

A package request in this design has a lifecycle: researcher submits a ticket naming the package, version and purpose; the research computing team resolves dependencies on the builder; the admission test runs; the research computing team (and the campus security office if egress changes) approves; the builder imports and signs; the study's VM image or repository is updated; the ticket records the digests. Each step has a cost that should be measured separately. The transitive dependency tree is the multiplier. A scientific Python package can pull in dozens of dependencies. As a rough check I resolved `scanpy` for Python 3.11 with `uv pip compile` on one build machine and got 45 pinned packages; `pytorch-lightning` gave 26. These counts are a point observation on one day and one platform, not a typical value, and are used only to show that a request is a set and not a single package. The design question for staff is whether each of the 45 needs its own approval, or whether the request is approved as a locked set and the diff against previously admitted packages is the unit of review. The second is the only form that scales.

Container-based delivery shortens the path for common stacks and lengthens it for the rest: a new R package means a rebuild of `p4-r`, a re-sign and a re-pull on every VM that uses it. PPM or a package snapshot would remove that step for R at the price of a server. The cost of each choice is staff time, and Chapter 11 treats who owns it.

8.10 Design considerations

1. Treat the software path as the enclave's change-control point for IS-3 and write the admission rule down: what each ecosystem requires before an artifact is visible to VMs, who approves, and what record is kept. Today the plan names a request but not a rule.
2. Add a build-time assertion that every image and R-bearing container ships R 4.4.0 or later, and a login-time check of the version, because the floor holds only by the current contents of EPEL [149, 153]. Handle `.rds` files from outside the enclave as untrusted input.
3. Decide, with the campus security office, a minimum age for Python and R packages and who may grant an exception. Do not adopt my 7 to 14 day suggestion without a reason; ask what detection latency the campus security office is protecting against, and record the answer.
4. Move P4 studies' Python path from the live PyPI remote to a standard repository filled by the builder with hash-locked wheels, or document why the live remote plus a client-side age rule is acceptable. Test whether the Artifact Registry PyPI remote passes upload-time metadata to pip and uv.
5. Confirm with Google whether Artifact Registry remotes offer any age, allow or scan control, and whether a yanked upstream package is removed from the cache. Record the answers; if none exist, treat the remote as a pull-through cache with no admission function.

6. Resolve the Yum Preview question by one of two routes: accept the remotes with a recorded security-office decision, or use builder-filled snapshot repositories. Confirm the launch stage of standard Yum repositories at the same time. Check each remote's path against the upstream layout and test a metadata fetch against all three remotes in the dry-run week.
7. Close the loop on authentication: specify which identity pulls from each repository, grant it read access in the infrastructure code, and add credential handling to the pip, dnf and Apptainer configurations. The offline plan contains none, and pilot test 3 should be extended to each source.
8. Separate build, sign and promote. The builder should hold no standing credentials, and the identity that can write to the curated repository should not run package installs from the internet. Add checksums or signature verification to every direct download in the build.
9. Sign every `p4-*` image, pin by digest, keep a bill of materials, and enforce verification on the VM; close the VM-side path to the Docker Hub remote for P4 studies.
10. Evaluate Posit Package Manager as a snapshot-pinned R source if the request volume for R packages is high, and establish whether campus already holds a license.
11. Measure software request turnaround and the number of dependencies per request during the pilot, and replace the "TBD" in the plan with the median and 90th percentile.
12. Add the controls to the pilot acceptance list: refused too-new package, failed unsigned image, failed build below the R floor, and cache purge after a yank. Add the Lightning case as a tabletop scenario for Chapter 10: a researcher reports that an installed package may be malicious, and the team must state what was fetched, by which study, and which credentials were reachable.

Chapter 9

The Researcher's Experience

Our plan gives a researcher one tunnel command, RStudio Server in a browser, an optional Linux desktop over a native remote-desktop client, a one-time per-VM password, a request process for software and a steward-mediated export. Those are the replacements for what the controls take away. The controls (no internet route, a login gate, no clipboard, curated software, a reviewed export path) each remove something a researcher would otherwise do on a laptop or a departmental server, and if the replacement isn't workable the work moves somewhere less safe. Most of what the plan says about this experience is intention. The environment does not exist, the `p4 connect` wrapper that would hide the tunnel command has not been written, and the plan itself marks the password helper, the idle stop and the Duo login as behaviors to be verified in a pilot with fake data.

Published evidence on researcher experience in secure environments is thin and mostly from the United Kingdom, where trusted research environments (TREs) are a national institution; I use what I could verify and say where the evidence stops. Chapter 5 describes the architecture the researcher meets, Chapter 7 the export rules behind the last step, and Chapter 8 the software process behind the package request.

9.1 What the published evidence says

9.1.1 Waiting for access is the recurring complaint

The most useful primary source I found is the interim evaluation of Administrative Data Research UK (ADR UK), prepared by Oxford Insights and Lateral Economics for the Economic and Social Research Council (ESRC) and published in November 2024 [170]. It drew on 109 survey responses (68 from researchers) and interviews. Its findings on access time are specific: more than a third of academic survey respondents reported that it took more than six months to access data; most respondents said access took at least four to six months, counting application and the time for data owners to provide data for linkage; interviewees described access taking upwards of a year for new projects, and delayed project starts. Of the TREs asked for their own timing, only one, SAIL in Wales, could supply a figure: an average of 8.4 weeks from the start of an application to data access between 2019 and 2024, excluding two pandemic years. On the other side of the ledger, just over half of surveyed researchers were satisfied overall with how easy it is to access data through TREs, remote access was singled out as helpful, and the evaluation states that once access is granted, most researchers find the TREs easy to use. Among respondents who could compare, more than 70% were more satisfied with ADR UK TRE support services than with alternatives [170].

Two cautions apply. First, these delays are dominated by data-owner approval, linkage and accreditation, which at UCR would correspond to the DUA, DSP and security-office review before the environment exists, and not to the login path. The evidence says little about how fast a researcher can get a working workspace once approvals exist, which is the part the research computing team and the systems team control. Second, a figure that circulates in one of the AI-generated research reports I consulted, that nearly a third of applicants to ESRC-funded TREs waited more than 12 months, is not in the primary source. The ADR UK evaluation says "more than six months" for more than a third of respondents, and "upwards of a year" only for some interviews. The ESRC has run a separate TRE user experience survey, but the public page located lists no results [171]. I treat the 12-month figure as unverified and don't use it.

The best primary evidence concerns how long approval takes, not how hard the environment is to use, and it does not support the "one-third wait over 12 months" figure

The ADR UK interim evaluation reports that more than a third of academic respondents waited more than six months for data, that SAIL averaged 8.4 weeks from application to access, and that researchers who got access mostly found the TREs easy to use. The 12-month claim in an AI-generated report was not found there. For the UCR pilot, the measurable and controllable interval is from an approved Data Security Plan to a researcher's first working session, and no published figure was found for that interval.

9.1.2 Other published sources

A presentation on the Safe Data Access Professionals' network notes that access to controlled data ranges from on-site only, through remote-execution portals, to remote desktop, and that facilities rely on trained, approved researchers and checked outputs [54]. A 2026 conference presentation on research software engineers (RSEs) in UK TREs argues that building and running these platforms draws on a wide set of roles, including cloud infrastructure management, statistical and anonymization expertise and user-interface design, many of them underappreciated [38]; the implication for UCR is that support for researchers is a staffed activity and not a by-product of the build, a point Chapter 11 takes up. A German research-data-infrastructure paper presents a desktop-as-a-service approach on the argument that data is often too large or too restricted to move and that a desktop avoids local setup and maintenance [172]; this is the same trade the plan makes by offering an XFCE desktop. None of the three reports measurements the plan could borrow.

One further abstract bears on a gap in the plan. Raab and colleagues note that some UK groups have proposed low-fidelity synthetic data, clearly labeled as not real, so that researchers outside a TRE can test code and explore structure before entering it, and they propose four checks before release: labeling, a disclosure evaluation, structural similarity to the real data, and documentation of differences [173]. The plan has no equivalent. Every hour of code debugging done inside the enclave is an hour spent in the slowest, most restricted working conditions we offer. One AI-generated report describes a synthetic-data onboarding practice at a US university, but that institution's practice was not confirmed on its own pages and is not relied on.

9.1.3 Claims not used

The same report also offers figures I could not trace to a primary source, so I don't use them: a satisfaction percentage from a university-wide Australian survey presented as TRE evidence, a 6,300-person survey with 28 responses, a 9-to-14-month onboarding baseline attributed to a consortium, ticket-deflection percentages from enterprise IT sources, remote-desktop latency figures from vantage-point measurements, and a 24-to-48-hour export target said to match a Scottish safe haven. Each may be true; none was verified, and none is needed for a plan that has not yet measured its own numbers. The plan's own targets are few: leavers removed within one business day, restore within one business day, and a software request turnaround written as "[TBD, e.g. 2 business days]". It sets no target for onboarding time, first login or export review.

9.2 The planned first day

I assembled the sequence below from the plan's runbooks (R1, R2, R5), the intake checklist, the add-person addendum, the desktop and RStudio design note and the planned configuration and image build. None of it has been run. Figures 9.1 and 9.2 draw it as swimlanes: the first covers the steps before a researcher touches the environment, the second the first day on the VM.

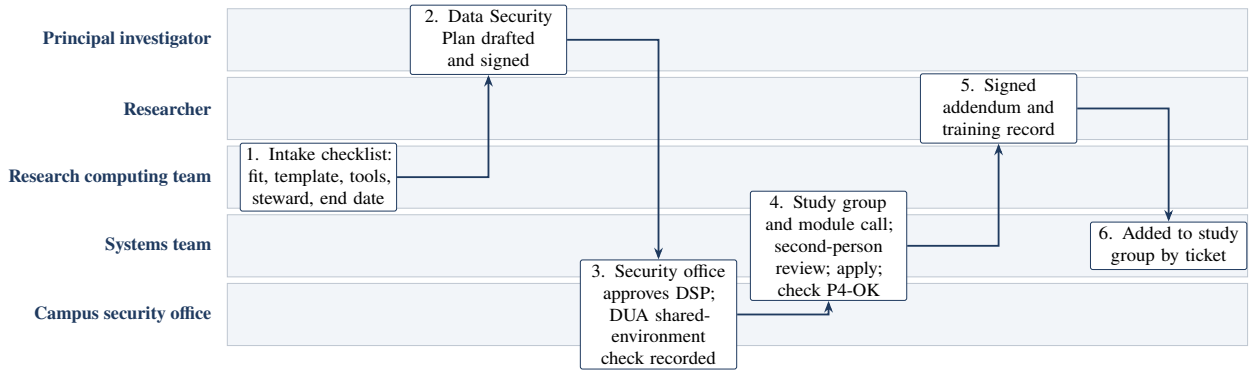


Figure 9.1: Planned steps before a researcher's first login (swimlane, planned process only; no step has been run and the plan sets no duration for any of them except removal of leavers). The study group is created at step 4 from the signed DSP; people added later follow steps 5 and 6.

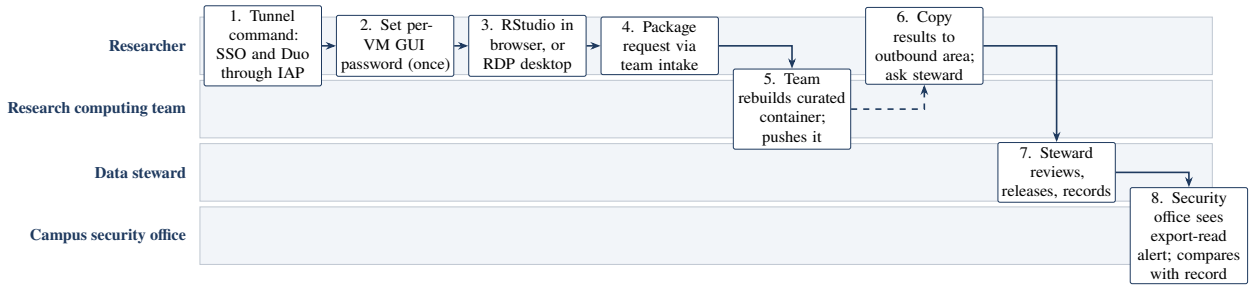


Figure 9.2: Planned first day on a study VM (swimlane, planned process only). The dashed arrow marks that the package request and the export request are independent; the container rebuild is shown only to place it. The plan sets no turnaround for steps 5 and 7; the container target is written as "TBD, e.g. 2 business days."

9.2.1 Intake and the signed plan

A study starts with an intake conversation with the research computing team. The checklist asks for the security office's classification (P3 or P4, with CUI routed to UCR's separate CUI enclave and medical-school clinical data routed to the medical school, as set out in Chapter 1), whether each data use agreement allows a shared, cloud-hosted, UCR-operated environment, whether it forbids copy-out (which would set `gui_only` mode), whether outside collaborators are involved, the VM template, data size, tools wanted, the named data steward, the end date and the funding source. The principal investigator then completes the DSP, the research computing team drafts its technical sections from a template, the campus security office approves, and the systems team onboards the study under runbook R1: create the study group and steward group, add the module call, obtain a second-person review, apply, confirm the VM reports P4-OK, and send the researcher guide. From the researcher's side this is a period of paperwork whose length the plan does not estimate. The ADR UK evidence suggests this is where delay accumulates.

9.2.2 Being added

Each person who will touch the data signs the addendum ("I will not copy study data off the secure VM by any means; results leave only through the data steward") with the PI, attaches a training record, and the systems team adds them to the study group on a ticket. The group carries the study's grants: OS Login without sudo and IAP tunnel access on that VM, the right to act as the VM's service account, and upload and read on the study's inbound bucket. This one step is the access control, so its speed is also a researcher-experience variable: a postdoctoral researcher who joins a running study waits for a ticket. The plan's only stated clock is removal within one business day.

9.2.3 Login through IAP with Duo

The researcher runs one `gcloud compute ssh` command with the `--tunnel-through-iap` flag, which the planned per-study module emits as an output. Google documents `gcloud compute ssh` as a thin wrapper around `ssh` that handles authentication and translates the instance name into an address [174]. Sign-in goes through the campus identity provider, where Duo is enforced (UCR fact, decision D2). The plan does not build the MFA step; it inherits it, and pilot test 1 is meant to confirm that Duo, with no Google 2-Step Verification, is enough for OS Login. Two more things need checking in the pilot. Google's OS Login documentation says that users who need SSH access from the Google Cloud CLI must hold the OS Login role at the project level, or a project-level role containing `compute.projects.get` in addition [175]; the module grants the role on the instance only, so the first real login may fail for a reason that has nothing to do with Duo. And if the campus Google Workspace tenant sets a Google Cloud session length, which Google's admin documentation says can be set between one and 24 hours and applies to the `gcloud` tool, researchers would see a re-authentication prompt on that cycle [176]. Whether UCR sets one is unknown.

If the boot checks have not passed, the fail-closed gate refuses the login (Chapter 5). The researcher would see a refusal, and the plan names no user-facing message or support path for it; the alert goes to the systems team and the campus security office, not to the researcher.

9.2.4 Setting the GUI password

OS Login accounts have no local password, and RStudio Server and xrdp sign in with a password. On first use the researcher therefore runs `p4-set-gui-password` over the authenticated SSH session, choosing a password of at least 14 characters; a root helper sets it only for the caller and refuses administrators. The

password is per VM and used only by RStudio and the desktop. The cost is a second credential for a task that already needed Duo. The plan flags that setting a password for an OS Login user may need a local shadow entry that such users might lack (open item 8, pilot test 9); if that fails, the design needs a different sign-in for RStudio, and the note lists Posit Workbench with single sign-on as a later, paid option. In `gui_only` studies, plain `ssh` to the VM runs this helper and nothing else, so over SSH the researcher gets no shell; a study with the desktop still has a terminal inside the VM.

9.2.5 RStudio or the desktop over the tunnel

With the tunnel running, the researcher opens `http://localhost:8787` for RStudio Server or points a native RDP client at `localhost:3389`. The server limits forwarding to those two ports. RStudio sessions suspend after 15 idle minutes (a running R job is not suspended) and the configuration requires sign-in again after 15 minutes of inactivity with no persistent sign-in. The desktop is XFCE with a browser that can reach only local pages because the VM has no internet route, a terminal, a file manager and an editor; clipboard, drive, printer and audio channels are refused by the server, so the client's settings cannot re-enable them. The same refusal means no paste into the desktop, so code and small inputs arrive through the inbound bucket or through git against the mirrors. GPU-accelerated 3D in the desktop is explicitly out of scope for the first version.

9.2.6 Requesting a package

There is no live CRAN, Bioconductor, conda or PyTorch index. Python comes from a PyPI mirror, system packages from Rocky and EPEL mirrors, and everything else from three curated containers (`p4-r`, `p4-torch`, `p4-bio`); a new R package is a request through the research computing team's intake, which rebuilds the container outside the perimeter and pushes it, with a target turnaround the plan writes as "[TBD, e.g. 2 business days]" and lists as open item 9. Chapter 8 evaluates the supply-chain side. For the researcher, the point is that the first time an analysis needs a package the container lacks, work stops for however long the request takes, and that the mirrors do not help with the most common case in R.

9.2.7 Requesting an export

Under runbook R5 the researcher places files in an outbound folder, a command copies them to the study's outbound bucket, and the named data steward reviews them against the release record template: aggregate results only, no direct identifiers, small cells suppressed (default fewer than 11 unless the DUA says otherwise), and consistent with what the DSP and DUA allow. The steward downloads the files, records the release, and the campus security office receives the export-read alert and compares it with the record in its weekly review. The plan gives no review turnaround, no rule for model weights and no way for the researcher to see queue status. Chapter 7 treats those gaps; here they matter because export is the step at which a researcher's schedule meets another person's calendar.

9.3 The connection path and its timers

A dropped tunnel is the likeliest day-to-day irritant, and it is shaped by several independent timers that were set by different people for different reasons. Table 9.1 lists the ones I could establish from documentation or from the plan's files.

Table 9.1: Timers that can end or interrupt a researcher's session. Rows 1 and 2 are Google documentation; rows 3 to 6 are in the plan's image and startup files; row 7 is a Google admin setting whose UCR value is unknown.

#	Timer	Documented behavior	Consequence for the plan
1	VPC firewall connection tracking	Return traffic is allowed while tracking state is active, which requires at least one packet every 10 minutes [97]	An idle tunnel with no packets could lose return traffic; periodic packets prevent it
2	IAP TCP forwarding	Disconnects sessions after one hour of inactivity; `start-iap- tunnel` attempts to reestablish a dropped tunnel [95]	Not stated whether SSH keepalive packets count as activity, or whether a reestablished tunnel keeps an SSH session alive
3	SSH server keepalive probe	Server probes every 5 minutes; about 15 minutes without a reply ends the session	Server probes also generate traffic at less than the 10-minute interval
4	Shell idle timeout	Interactive shells end after 15 idle minutes	Applies to shells only; a tunnel-only session has no shell
5	RStudio session and sign-in	Suspends after 15 idle minutes; sign-in required again after 15 idle minutes	Running R jobs are not suspended; idle browsers re-prompt
6	Idle stop	VM powers off after 120 idle minutes by default (240 in the GPU example)	The process check's user range probably does not cover OS Login accounts, so a tunnel or a detached job may not count as activity
7	Google Cloud session length	Admin-set, 1 to 24 hours, applies to `gcloud` [176]	Possible re-authentication prompt on that cycle

The plan's server-side probes (row 3) already send a packet every five minutes, which is inside the ten-minute tracking window, so the firewall is probably not the weak point as long as the client is reachable. The weak points are the client (a laptop that sleeps, a home or campus network that drops idle connections) and the IAP hour. Meanwhile the tunnel command the planned module emits has no client-side keepalive options. The OpenSSH client sends no keepalive messages by default (`ServerAlive- Interval` defaults to 0), and with a nonzero interval it disconnects after the interval times `ServerAliveCountMax` (default 3) if the server stops answering [177]. A wrapper that sets both makes a dead tunnel fail fast, so the wrapper can restart it, instead of hanging until the researcher notices a frozen browser tab.

9.3.1 A proposed `p4 connect` wrapper

The plan lists a `p4 connect <study>` wrapper for Mac and Windows as a pilot deliverable and says laptops need the `gcloud` CLI. The wrapper does not exist yet. Google describes `start-iap-tunnel` as opening a local tunnel through which another process, such as SSH or RDP, can connect [178]. Figure 9.3 shows the command it would run, which extends the module's tunnel command with keepalive and fail-fast options, and Figure 9.4 shows a mock-up of the session. Project and zone values are placeholders. The intervals shown (60 seconds, five misses) are starting values to be tested, not recommendations backed by measurement.

```
gcloud compute ssh STUDY-VM --project PROJECT --zone ZONE --tunnel-through-iap -- \
  -N -L 8787:localhost:8787 -L 3389:localhost:3389 \
  -o ServerAliveInterval=60 -o ServerAliveCountMax=5 \
  -o ExitOnForwardFailure=yes
```

Figure 9.3: Proposed tunnel command for the wrapper: the plan's tunnel command plus client keepalives. Placeholders are in capitals. `-N` opens no shell, which is also why shell timeouts do not apply to it.

Three design questions sit behind the mock-up. The wrapper should print the one-time password step only when needed, which means discovering whether a password has been set without exposing a way to

```
$ p4 connect survey-demo
p4: gcloud found; sign in with UCR SSO + Duo (browser opens)
p4: opening IAP tunnel, keepalive 60 s x 5, restart on failure
p4: tunnel up
      RStudio  http://localhost:8787     (per-VM password)
      Desktop  localhost:3389            (RDP client, if enabled)
p4: first time on this VM?  p4 connect survey-demo --set-password
p4: Ctrl-C closes the tunnel

p4: link lost (no keepalive reply); restarting in 5 s ...
p4: tunnel up again; RStudio session should resume

$ p4 connect survey-demo            # a second scenario
p4: login refused: this VM has not passed its security checks.
    The systems team has been alerted. Retry in 15 minutes or open a ticket.
```

Figure 9.4: Mock-up: a proposed `p4 connect` session. The wrapper does not exist, and the messages, option names and the retry advice are illustrative. Whether the researcher could be told why a login was refused depends on the gate's message, which the plan has not specified.

enumerate users. It should not log or cache anything beyond what `gcloud` already does. And it adds a software artifact that the systems team or the research computing team must version, sign and support on Mac, Windows and Linux laptops, most of them unmanaged; that is a staffing item and not only a script.

9.4 Working conditions inside the VM

9.4.1 A stopped VM and who can start it

The idle stop powers a VM off after a configurable period without sessions, user processes or load, and the disks persist. This saves money (Chapter 12), but when I read the module's IAM bindings, the study group holds OS Login, IAP tunnel access and the service-account role, and none of these starts a stopped instance; only the administrator group holds instance administration. A researcher returning on Monday to a VM stopped on Friday would therefore appear to need a ticket. I found this by reading the planned configuration; it has not been tested. Google's Compute Engine IAM reference lists a start permission for instances; granting a narrow role containing it on that one instance to the study group, or exposing a start action through the wrapper, would be a design change to consider with the campus security office. A related question is what keeps the VM awake. The design intends to count any process owned by a regular user, to catch detached `tmux` and `nohup` jobs, but as written it misses the identity provider's users: the idle check only counts users in the range a local install would use, and the identity provider hands out IDs far outside it [179]. Some detached jobs and idle tunnels would then not keep the VM awake, and quiet jobs could be powered off under the researcher; interactive sessions and an explicit keep-running marker would still hold it up. I found this by reading, not running: inactivity detection must cover the identity provider's user-ID range, and pilot test 12 should run with a real OS Login account and a detached job.

9.4.2 Latency and graphics

No verified latency figure for this path is available; the figures in the AI-generated report are not used. The path crosses the campus network, Google's network, IAP and a TCP tunnel to a region (us-central1) a considerable distance from Riverside, and Google's documentation says IAP TCP forwarding is not intended for bulk data transfer and that Google reserves the right to rate-limit abusive use [95]. For typing in RStudio

this is likely to be tolerable; for a desktop session with large windows or image browsing it is the open question. The mitigation in the plan is the native RDP client (the second panel's change from a browser desktop), which makes graphics adaptation the client's job. The pilot should record round-trip time and subjective responsiveness for each interface, from campus and from home, and should not assume the same answer for both.

9.4.3 Moving code and data in

With no clipboard, no SFTP and no internet, work begins by getting code and data into the VM. Researchers upload data to the inbound bucket (pilot test 14 measures a 200 GB upload from campus), pull code with git through whatever mirrors allow, and install packages from the mirrors or containers. The plan does not say how a researcher outside the perimeter reaches the storage API when the perimeter is enforced; the planned ingress rule covers only the tunnel, and this is a question for Chapter 7 and the dry-run. The practical consequence is that a researcher's first week is dominated by logistics that a laptop user never thinks about.

9.4.4 GPU work

The GPU template is one L4 on a `g2-standard-16` machine, and a researcher would see an ordinary Linux VM with the NVIDIA driver and a curated PyTorch container; GPU instances cannot live-migrate, so host maintenance terminates them and a running job would be lost, which researchers should be told. The shared cluster's GPU quotas belong to that cluster's project; the secure project is separate, so whether it has L4 quota at all is a question for the systems team before the first GPU study (Chapter 13).

9.4.5 Collaborators from other institutions

Decision D13 gives outside collaborators UCR affiliate accounts under the same DSP, and the `ucr.edu`-only policy stays. The alternative Google documents for users from another organization, a dedicated external-user role granted at the organization level [175], is not used. For the researcher, an affiliate account means a UCR-managed identity, Duo enrollment and the same addendum and training as anyone else. The plan lists the affiliate-account process as open item 10, and whether affiliates can enroll in Duo and sign in through the same SSO path is not shown in any file read. The collaborator's lead time is therefore unknown, and for multi-institution studies it may exceed the VM's.

9.5 Friction points and what the plan does about them

Table 9.2 lists the friction points in the design, what the plan does to reduce each, and what remains. "Plan mitigation" means something specified in the plan or its files; none has been tested.

9.6 Measuring the experience in the pilot

The plan's fifteen pilot tests (Section 5.10) are acceptance tests: they establish that controls work. None measures a researcher's time or opinion, and only test 14 produces a number a researcher would feel (the upload time). A secure environment can pass all fifteen and still be avoided. The pilot should therefore add a small measurement plan, with the caveat that the plan names three demonstration studies and the first real pilot study is not yet identified (open item 11). With a handful of users, results should be reported as individual intervals and quotations, not as averages or percentages.

Table 9.2: Friction points, planned mitigation and residual friction. Residual items are inferences from the plan's files unless a source is cited.

Friction	Mitigation in the plan	Residual friction
Approvals before day one	Intake checklist, DSP templates, runbook R1	No target time; depends on DUA and security-office review; published delays are mostly here [170]
Joining a running study	Signed addendum, ticketed group add (R2)	Ticket latency; training prerequisite; no stated clock
Installing `gcloud` and signing in with Duo	`p4 connect` wrapper (pilot deliverable)	Wrapper not written; unmanaged laptops; Windows support; possible project-level role need [175]
Second password for RStudio and desktop	One-time helper, 14 characters, per VM	Two credentials; helper may not work for OS Login users (open item 8); no reset procedure beyond rerunning it
Dropped tunnels	Server probes every 5 minutes; proposed client keepalives	IAP one-hour inactivity and reconnect behavior not specified for SSH [95]; laptop sleep
Packages not in the image	Mirrors, three curated containers, request process	Each new R, conda or CUDA need is a request; turnaround is TBD
No clipboard, no drag and drop, no SFTP	Code via inbound bucket or git over mirrors	Cannot paste; routine file handling is slower; `gui_only` adds restrictions
VM stopped by idle stop	Disks persist; 120-minute default; process check (user range likely misses OS Login users)	Study group appears unable to start a stopped VM; detached jobs may be stopped mid-run
Refused login (gate closed)	Alert to the systems team and security office; admins can still log in	No researcher-facing message or support path specified
Export wait	Steward runbook R5, checklist, small-cell default	No turnaround or queue visibility; no model-weights rule
Latency and desktop graphics	Native RDP client; XFCE; no 3D in v1	No measurement; region distance; IAP not intended for bulk transfer [95]
Code development on real data	None	All debugging happens inside the restricted environment; synthetic data not planned [173]
Outside collaborators	Affiliate accounts under the same DSP (D13)	Process open (item 10); Duo enrollment for affiliates unconfirmed

Table 9.3: Proposed pilot metrics. No target values exist in the plan; the targets column is deliberately empty until the pilot has produced baselines. All are proposed.

Metric	Definition (start to end)	Data source
Time to onboard a study	Security-office approval of the DSP to module applied and P4-OK logged	Approval record, change ticket, Cloud Logging
Time to add a person	Signed addendum received to group membership change	Ticket, group audit log
Time to first login	Group add to first successful IAP sign-in and SSH session; report first RStudio page load separately	Audit and IAP logs in the log project; researcher diary
First-week failures	Count of login refusals, password-helper failures, tunnel drops and idle-stop surprises, with cause	Alerts, wrapper output pasted by users, support tickets
Package request turnaround	Request received to package usable in the container	Intake ticket timestamps
Export turnaround	Researcher's request to release record signed; also count of rejected or revised requests and reasons	Steward release record, export-read alert time
Connection quality	Round-trip time, and a three-level rating of responsiveness, per interface and location (campus, home)	Short form after each of the first few sessions
Satisfaction	A short standard usability questionnaire plus open questions (what did you work around, what would you stop using) at week 1 and at the end of the pilot	Survey; interviews with every pilot user

Time intervals should be taken from system records where they exist and from the researcher only where they do not, because a researcher remembers waiting longer than the log shows. The satisfaction survey should be accompanied by interviews with every pilot user, since small samples hide the one blocking problem that matters. And the pilot should record workarounds explicitly: the most informative finding would be any place a researcher reached for a tool outside the environment, because the published concern about secure environments is that friction pushes work elsewhere. The ADR UK survey is itself a reminder that response rates on researcher surveys can be limited, so interviews matter.

What the researcher path still lacks

Our plan has the pieces of a researcher path: a one-command tunnel, RStudio and desktop access, a documented password step, request processes for software and export, and a stated residual-risk position that tells researchers what is prohibited. It does not yet have the pieces that make the path tolerable day to day: a wrapper, a researcher-facing message when the gate closes, a way to start a stopped VM, a named turnaround for software and export, a plan for code development without real data, and a staffed support role. Several of the plan's own open items (8, 9, 10, 11) are researcher-experience items. Reading the planned configuration and image build against Google's OS Login and IAP documentation, I would make three checks at the first pilot logins: that a study-group member can sign in with `gcloud compute ssh` although the OS Login role is granted on the instance and not at project level, that the study group can start a stopped VM (it holds no start permission today), and that the idle stop counts OS Login users' processes so it does not stop a VM under a running job. The environment would also be judged against the existing alternatives researchers use today, which are departmental servers and laptops; the plan does not describe how it would compare.

9.7 Design considerations

1. Provide a low-fidelity synthetic dataset, labeled as such and checked as in Raab et al. [173], outside the perimeter, so that code is developed and debugged before it enters the environment; decide who builds it and who checks disclosure.
2. Write and publish target intervals for onboarding, adding a person, software requests and export review only after pilot baselines exist, then commit to them; name an owner for each interval.
3. Build the `p4 connect` wrapper with client keepalives and fail-fast options, test it on Mac, Windows and Linux laptops, and decide who versions, signs and supports it.
4. In the pilot, test: whether SSH keepalives count as IAP activity; what happens to an SSH session when `gcloud` reestablishes the tunnel; a tunnel idle for more than an hour; laptop sleep and resume.
5. Test `gcloud compute ssh` for a study-group member holding only the instance-level roles in the module, and fix the grants if the project-level requirement applies.
6. Make inactivity detection cover the identity provider's full user-ID range, then test with a real OS Login account that a detached job keeps the VM awake and that a forgotten tunnel does not; define a researcher path to start a stopped VM (a narrow instance-level start right or a wrapper action) with the campus security office's agreement.
7. Specify the message and support path a researcher sees when the fail-closed gate refuses a login, so that a control failure is a ticket and not a mystery.
8. Decide the plan for researchers who cannot make the GUI-password helper work, and write a password-reset procedure.

9. Add researcher-facing items to the pilot acceptance list: time to first login, a timed package request and a timed export request with fake data, using the metrics in Table 9.3.
10. Confirm early that affiliate accounts can enroll in Duo and use the same sign-in path, and estimate the lead time for a collaborator from another institution.
11. Fund a named researcher-support role for the pilot and decide how its workload is measured (tickets, hours), since the RSE literature treats such roles as part of running a TRE and not as overhead [38].

Chapter 10

Logging, Monitoring and Incident Response

Our plan turns on Data Access audit logging for every service in the secure project, and that one setting is what lets us say later who read which object. The logs have two readers: an analyst who has hours to decide whether something has gone wrong, and a sponsor, data provider or regulator who will ask months later who touched which records and what we did about it. HIPAA ties both readers to the same evidence. The Security Rule requires mechanisms that "record and examine activity in information systems that contain or use electronic protected health information," procedures to "regularly review records of information system activity," and procedures to identify, respond to and document security incidents [180, 181]. The Breach Notification Rule then presumes that an impermissible use or disclosure is a breach unless the entity shows a low probability of compromise through a documented assessment of four factors, and it puts the burden of proof on the entity [182, 183]. One of the four factors is whether the information was actually acquired or viewed. Only logs can answer that. FERPA has no breach rule at all (Section 10.5), so for student records the logs serve the agreement, state law and UC policy instead.

The plan has four layers: those Data Access audit logs; guest telemetry from the Linux audit daemon and the Ops Agent; a cross-project sink into a retention-locked bucket in a log project administered by the campus security office; and nine alert policies that email two queues. Behind them sit a weekly review runbook (R11) and a three-line incident runbook (R9). The foundation's offline plan contains the audit configuration, the log bucket, the sink and the nine alert policies. I read each layer against Google's documentation, the HIPAA text and UC policy, asking where the evidence would be thin on the day it is needed. The control-plane trail is the strongest part of the design, the record of what happens to data on a VM disk is the weakest, and the incident process is the least developed.

10.1 What the plan would record

10.1.1 Control-plane records

Cloud Audit Logs writes up to four kinds of entry for each project. Admin Activity entries record configuration and metadata changes, are always written, and cannot be configured, excluded or disabled, even if the Logging API is turned off. Data Access entries record reads of configuration and of user data and writes of user data; except for BigQuery they are off by default because of volume, they are written to the project whose data was accessed, and they cost money. System Event entries record Google-initiated changes. Policy Denied entries record denials caused by security policy, such as a VPC Service Controls violation; they are on by default and cannot be disabled, though exclusion filters can drop them [184]. Google's page also states that the entries Cloud Audit Logs writes are immutable.

The plan enables Data Access logging for all services in the secure project with one project-level audit configuration covering `ADMIN_READ`, `DATA_READ` and `DATA_WRITE`; the offline plan creates it. For Cloud Storage, Google lists getting object data, getting object metadata and listing objects as `DATA_READ`, and creating or deleting objects as `DATA_WRITE` [121]. The export-read alert and the steward's reconciliation of releases both depend on those entries. Reading them takes a role that a plain Logs Viewer lacks: Google's page says Logs Viewer cannot see Data Access entries in the `_Default` bucket and names Private Logs Viewer for that purpose [184]. The planned configuration grants the security office both roles on the secure

project and grants the compute administrators neither.

Where entries sit by default is a second layer of protection that the plan does not have to build. Admin Activity and System Event entries always go to the `_Required` bucket of the project where they originate, which has a fixed 400-day retention that cannot be changed, and Data Access entries go to `_Default` (30 days unless changed) [90, 185]. The plan's sink copies a chosen set of entries to the log project (next section), so there would be two copies of the administrative trail with different custodians for at least 400 days.

Audit logs have a limit that the four-factor assessment will run into. They record API calls to Google services, and Google's HIPAA page notes that they do not include resource data contents [73]. A Data Access entry can show that a principal read an object from a study bucket. It cannot show what a researcher did with files on the VM's data disk, and in the plan most working data sits on that disk, not in a bucket. For disk data, the evidence of "viewed" is whatever the guest records.

10.1.2 Guest telemetry

The audit rules baked into the image watch changes to the account, group, privilege and remote-login configuration, and record execution of a small set of file-transfer tools under a single copy-out key. The rule set ends by locking the audit configuration; the `auditctl` manual states that in this mode changes are audited and denied and the configuration can be changed only by rebooting [186]. The boot check confirms that the audit daemon is running in locked mode and fails the VM's check otherwise (Chapter 5). The Ops Agent configuration collects the audit log, the RStudio server logs and the remote-desktop logs in one pipeline. Google's documentation says the agent collects the general file-based syslog by default [187], so the boot-check messages, which are written to syslog, flow without extra configuration.

Reading these files together, I see three limits.

One alert watches the wrong place. One planned alert's log source does not match where the operating system writes the events: the alert looks in one log file while the distribution's default logging setup writes those events to another [188]. The records still reach the audit log, which the plan ships, but as written nothing would alert on them. Every audit alert needs a test that triggers it on the real image. The fix is to collect the right source or base the alert on audit records; pilot test 7 would confirm either. Chapter 14 carries the item in its register.

The audit log arrives as raw text. The guest audit log is sent line by line with no parser. The copy-out audit signal has no alert or scheduled review step. The same holds for the audit records described above.

No rule watches reads of study data. The rules record that a copy tool ran. They do not record which files a process opened on the data volume. Auditing every file read on an analysis volume would be noisy, so the omission is reasonable, but for data on disk the evidence for factor three is the session record (who was logged in, when, from which tunnel), the copy-tool events and the absence of any route off the VM.

10.1.3 Network logs

The subnet enables flow logs at five-second intervals with full sampling and all metadata, and the firewall rules log matches. Both feed the sink filter. Google classifies VPC Flow Logs and firewall rule logs as vended network logs with their own ingestion price ($0.25 per GiB on the pricing page) and says they are also subject to network telemetry pricing [1]. With no internet route and IAP-only ingress the volume should be small, but the plan has not measured it. The pilot should record GiB per day by log type before anyone quotes a storage figure.

10.2 The log project, the sink and retention

10.2.1 Flow of records

Figure 10.1 draws the path from the VM and from Google's control plane to the bucket held by the security office, as the planned configuration describes it.

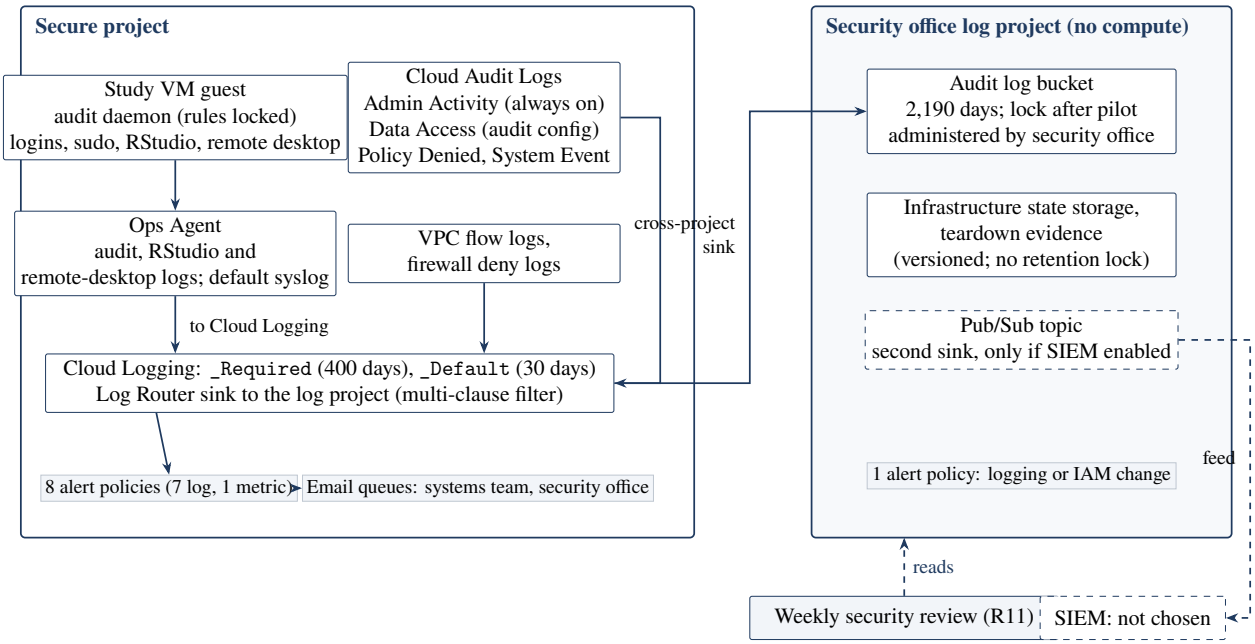


Figure 10.1: Planned log flow, drawn from the planned configuration. Solid arrows exist in the offline plan; dashed boxes and arrows are optional or undecided (the SIEM topic is created only when a SIEM is enabled, and the plan names no SIEM). The offline plan contains one sink to the log bucket and no perimeter, so the egress question for the sink is untested. Of the nine alert policies, eight are in the secure project and one is in the log project.

10.2.2 Cross-project behavior

Google's sink documentation says a sink's destination can sit in a different resource from the sink, and gives the example of routing a project's entries to a log bucket in another project. The sink's writer identity needs the Logs Bucket Writer role on the destination [189]. The plan grants that role to the sink's writer identity on the log project. Because the destination is a bucket, not the project itself, the destination project's own sinks do not reroute the entries, and the destination's `_Required` bucket stores only entries that originate there [189]. The plan's one sink therefore does what it is meant to do on paper. Two questions remain for the pilot and are shared with Chapter 4: whether the sink needs an egress rule when the perimeter is enforced (Google says project-level sinks to a protected destination need no ingress rule, and the log project is outside the perimeter as drawn), and whether the optional Pub/Sub sink, which crosses the same boundary, behaves the same way.

Our separation of duties follows the UC Event Logging Standard's intent. The standard says that, when possible, administrators of resources at P3 and above must not have permission to erase, deactivate or modify logs of their own activity (section 4.11), and it expects a SIEM feed as required and scoped by the location's chief security officer (section 4.9) [190]. The compute administrators hold no project-level role in the log project, the audit log bucket is administered only by the security office, and the security office has write rights

there but only read rights in the secure project. That separation is not complete, because of the deployer problem in Chapter 5: the deployment identity reaches both the secure project and the storage in the log project that holds teardown evidence. The plan answers with alerts on IAM, logging and audit-config changes in both projects, which make tampering visible after the fact; they do not prevent it. That stance makes the alert filters (Section 10.3) part of the control.

10.2.3 Retention, lock and what HIPAA requires

The log bucket is a Cloud Logging bucket, not a Cloud Storage bucket, so Google's rules for log buckets apply: retention is configurable from 1 to 3,650 days (30 by default), shortening a retention period leaves a 7-day grace period before expired entries are deleted, and locking is irreversible: it locks the bucket against updates including retention, and the bucket cannot be deleted until every entry has met its retention period [90]. The console cannot lock a log bucket; the command line or the API must be used, which in the plan means a lock flag in Terraform. Google's page suggests a project lien to prevent deletion of a project that contains a locked bucket [90]; the planned configuration contains none. The plan sets 2,190 days and leaves the bucket unlocked until pilot acceptance, with the security officer accountable for the lock. Until then the security office could shorten or delete the bucket, which is acceptable for fake-data pilot logs.

45 CFR 164.316(b)(2)(i) requires a covered entity or business associate to retain *the documentation required by paragraph (b)(1)* for six years from the date of its creation or the date it last was in effect, whichever is later [191]. Paragraph (b)(1) covers the written policies and procedures and a written record of any action, activity or assessment that the Security Rule requires to be documented. A policy, a risk analysis, a weekly review record and an incident record are documentation in that sense. Raw log entries are not named. The audit-controls standard and the activity-review specification state no retention period [180, 181]. The breach subpart imports the administrative requirements of the privacy rule's documentation provision (164.414(a)), which carries its own six-year retention and the burden-of-proof record for any notification or no-breach determination [183]. One AI-assisted literature review states that HHS penalizes entities that cannot produce historical logs, and concludes that six years of locked raw logs is the practical standard; I could not confirm that claim from a primary source. UC's standard sends the question elsewhere: logs are retained by the records schedule set by the data's accountable owner, contractual obligations, litigation holds and regulatory requirements [190]. Six years of raw logs is therefore a conservative design default for the security office and records management to confirm, not a quotation from the regulation.

I would fix two details before the lock. Six calendar years from 2026-11-01 span 2,192 days (two leap days), not 2,190; irreversible settings are the place to be exact. And the lock sets a floor that conflicts with a possible data use agreement clause. Audit entries can carry resource names, labels and object paths (Google's HIPAA page says audit logs might capture resource metadata, and warns against putting PHI in resource names or alerting configurations [73]). If a DUA requires destruction of all copies of derived data on termination, a locked log full of object names is a conflict that cannot be resolved later by deleting anything. The countermeasure is upstream: study identifiers in bucket names and labels must not derive from participants, and the DSP template should say so.

Six years of raw logs is a design choice; the regulation names six years for documentation

45 CFR 164.316(b)(2)(i) fixes the retention of documentation, such as policies, assessments, review records and incident records, at six years. It does not name raw system logs, and the audit-controls and activity-review provisions state no period. The plan's 2,190-day locked bucket is defensible as a conservative default because a breach can be discovered long after the event and the entity carries the burden of proof, but the lock cannot be undone. The review records, incident files and teardown evidence

that *are* documentation have no tamper-resistant home in the plan.

The log project contains one Cloud Logging bucket (locked after pilot) and one Cloud Storage location for infrastructure state, which has versioning but no retention policy or lock. Teardown evidence is written to that same storage, within the deployer's reach (Chapter 5). The weekly review record in R11 has no stated location at all. A Cloud Storage retention policy, locked on a separate evidence bucket, would give the review records, the teardown evidence and the incident extracts the property the log bucket has [88]. Google's page notes that locking a Cloud Storage retention policy applies a lien to the project automatically, and that the person removing the lien should be someone with legal or compliance accountability outside the IT department [88].

10.2.4 Cost of keeping logs

Table 10.1 compares three ways to hold one tebibyte (1,024 GiB) for 72 months at list prices read on 2026-11-01. The arithmetic is mine; the volume is unknown, so the table is a per-TiB unit for scaling after the pilot measures real volume, not a forecast.

Table 10.1: List-price cost of holding 1 TiB of log data for 72 months, us-central1 where regional. Prices: Cloud Logging $0.50 per GiB ingested (first 50 GiB per project per month free; 30 days of storage included) and $0.01 per GiB per month beyond 30 days [1]; Cloud Storage regional Standard $0.0200 and Archive $0.0012 per GiB per month, from the hourly rates on the pricing page [2]. Free allotments, retrieval fees and the Archive 365-day minimum duration are ignored.

Destination	Ingest, once	Storage, 72 months	Notes
Cloud Logging bucket (plan)	$512	$727	Searchable in Logs Explorer for the whole period; retention lock available. Storage is charged for the 71 months beyond the included 30 days. Total about $1,239.
Cloud Storage Standard	none stated	$1,475	Files, not searchable in place; no advantage over a log bucket for this use.
Cloud Storage Archive	none stated	$88	Cheapest; entries must be exported back before analysis; retention lock available.

The table leaves out a cost the plan creates. Audit entries and guest logs are written to the secure project's `_Default` bucket by the system sink, and the plan's sink copies the same entries to the log bucket. By my reading of the pricing page, ingestion into a log bucket is charged where the bucket sits and routing itself is free, so the same entry is ingested twice. An exclusion filter on the `_Default` sink would avoid that, at the price of losing the 30-day local copy that the security office's read-only roles would otherwise use. The pilot's billing export (planned in the log project) is the way to settle it.

Gaps to close in log retention before first PHI

Our plan has the right structure for tamper resistance: a separate log project, no compute-admin project role in it, a retention-locked bucket and alerts on changes to both projects. Gaps to close before first PHI: confirm 2,190 days (or another figure) with the security office and records management; decide who owns the log project's billing and closure for years after the secure project is retired; add a lien; create a locked evidence bucket for review records, teardown evidence and incident extracts, out of the deployer's reach; point the privilege-use alert at the right log and confirm it on the actual image; and measure log volume and double-ingestion cost in the pilot. The plan is also silent on what happens to the locked bucket if a DUA demands destruction of derived data.

10.3 Alerts

The planned configuration defines notification channels for the systems team's queue and the security office's queue (and a third for the log project) and nine alert policies. Eight of the nine are log-based: seven in the secure project and one that watches the log project. The ninth is a metric threshold on disk use. All nine are email alerts; the eight log-based policies have a five-minute notification rate limit and auto-close after 1,800 seconds, the minimum Google permits [192], while the disk policy sets neither. The display names and routing in Table 10.2 come from the offline plan.

Table 10.2: The nine planned alert policies, in the order of the offline foundation plan. "Notifies": Security is the campus security office's queue, Systems the systems team's. "Runbook" names the plan's runbook that applies; none of the log-based policies has a response step beyond the text in its documentation field.

Display name	Trigger	Notifies	Runbook
P4 audit_config_change	Project IAM policy change whose delta alters audit configuration (Data Access settings)	Security	R11, R9 if no ticket
P4 boot_check_failed	Boot-check failure message in the system log	Systems and Security	R8
P4 disk over 90%	Agent disk-used metric over 90% for 10 minutes on any secure VM	Systems	none named
P4 export_read	Object read on a study's export bucket	Security	R5 reconciled in R11
P4 iam_change	IAM policy changes in the secure project	Security	R11, R9 if no ticket
P4 logging_config_change	Sink create, update or delete; bucket update or delete; CMEK or settings update, in the secure project	Security	R11, R9 if no ticket
P4 log project: logging or IAM change	IAM, sink, bucket or Cloud Storage bucket deletion or permission change in the log project	Security (own channel)	R9
P4 login_gate_refused	Login-gate refusal message (a researcher was refused because the VM check is not OK)	Systems	R8
P4 sudo_use	Message in the system log	Security	R11 weekly review

Routing differs from the plan text in one place. The plan text sends the boot-check alert to the systems team's queue; the configuration sends it to both queues. Two recipients is the safer reading, and the document and the code should agree.

Filter strictness. Google's query language distinguishes exact equality, substring match and regular expressions [193]. Google's audit pages show that project-level, instance-level and bucket-level permission changes are recorded under differently formed method names [194], so an alert written as an exact match on one form will miss the others. Each resource type that carries a permission needs its own match and its own test. Because the deployment path holds broad storage and IAM rights, a change to who can read a study's export bucket is within its reach. Whether the `iam_change` alert fires on each of project, bucket and instance IAM changes is a pilot test the plan does not list (its test 7, in Section 5.10, covers the status and privilege alerts and sink changes).

Group changes are outside the trail. Study, administrator and steward group memberships live in the identity provider, so directory group changes are outside the project audit trail. The weekly review would need the directory's group-membership audit log, which the plan does not mention.

Rate limiting also suppresses. A repeated event produces one email per five minutes, so a review that counts emails undercounts events; the review has to query the logs.

Alert text must not carry PHI. Google's HIPAA page says not to place PHI in alerting configurations [73]. The alert documentation strings in the plan contain none. I did not verify what an emailed alert carries from a matched log entry, and command lines in system log messages can include file names; the pilot should read the first real alert email before the queue addresses are fixed. Chapter 4 notes that a notification channel pointing outside the organization is a documented exfiltration path under VPC Service Controls; the two queue addresses should stay on ucr.edu.

10.3.1 What the nine policies do not cover

Table 10.3 lists events that matter to this design and have no alert. The Compute Engine audit page confirms that Shielded VM, metadata and service-account changes are written as Admin Activity entries [194], so each can be matched with a log-based policy.

Table 10.3: Events with no planned alert. "Evidence" states what I verified; the proposals are mine and untested.

Event	Why it matters here	Evidence and proposed check
Shielded VM configuration change	Shielded VM settings can change after creation, and no alert watches for it (Chapter 4).	Shielded VM configuration updates are an Admin Activity method in the Compute Engine API. One log-based policy to the security office.
Instance metadata or service account change	The boot check reads the study's mode settings from instance metadata, and the startup script itself is metadata. An administrator who edits them could weaken a `gui_only` VM at the next boot.	Metadata and service-account changes are Admin Activity methods. Add to the same policy.
Sink delivery failure	Silent logging failure is a commonly cited audit failure mode. The log-project watch catches IAM and config changes, not other causes such as quota or a perimeter block.	Google documents sink error notifications [189]; none is wired to a queue. Check in the pilot, with the perimeter in dry run.
Agent or audit daemon stops after boot	The login gate evaluates at boot and at login only; the plan has no periodic re-check. A stopped Ops Agent leaves the VM running with no telemetry.	A metric-absence alert is a documented Cloud Monitoring condition type [195]; not planned.
Policy Denied entries	VPC Service Controls dry-run violations and denials are the pilot's main evidence for enforcing the perimeter.	Policy Denied entries are on by default [184]; the plan says to watch them but has no alert or saved query.
Copy-out audit hits and org policy, firewall or snapshot changes	Copy-tool use is the plan's one guest signal for copy-out; policy, firewall and snapshot changes alter the boundary or the evidence.	Guest events arrive as raw text. Add a saved query reviewed weekly, then decide which hits merit a live alert.

The first row is the Shielded VM gap from Chapter 4, written as a ticket: one policy, one filter, one queue. Its absence leaves the compensating control in the same place as the log bucket's: the Admin Activity entry exists and is retained, and nobody is told when it appears. Whether the plan should alert or accept that and write down the reasoning, as Chapter 4 says, is a question for the security office. A weekly review that reads only the plan's four alert streams would never see it.

10.4 Review, SIEM and detection services

10.4.1 Weekly admin-activity review

Runbook R11 has the security office review sudo-use, IAM-change, logging-change and export-read alerts each week against change tickets and export records, note exceptions, and keep the review record. It answers two regulatory asks at once: the HIPAA requirement to regularly review activity records, and the UC standard's requirement that privileged-user actions at P3 and above be logged and reviewed by a peer, "based on risk" [181, 190]. It is meant to stop logs piling up in cold storage with nobody reading them. Three details are unspecified. The review reads alerts, not logs, so anything outside the nine filters is invisible (Table 10.3). The plan does not say who covers absences in the security office. And the record has no stated home, which is the evidence-bucket question raised in the retention discussion. Three added queries would close most of the gap: Data Access reads by principal and volume, Policy Denied and IAP denials, and the guest copy-out key.

10.4.2 SIEM options

The UC Event Logging Standard requires units at P3 and above to send log data to a SIEM as required and scoped by the location's chief security officer [190]. The plan lists the SIEM as undecided and offers one mechanism: a Pub/Sub topic in the log project fed by a second sink, created only when a SIEM is enabled. Google's sink documentation names Pub/Sub as the destination for third-party tools such as Splunk or Datadog [189]. Table 10.4 sets out the options on the evidence read. I do not know which SIEM, if any, the campus security office runs, and the table does not assume one.

Table 10.4: SIEM options for the planned environment. "Verified" means read in Google's documentation on 2026-11-01.

Option	What the evidence says	Open questions
Campus or enterprise SIEM via the Pub/Sub topic	The plan's Terraform creates the topic and sink when enabled. Pub/Sub is the documented path to third-party tools [189].	Which SIEM; who runs the collector inside or outside the perimeter (Pub/Sub is a restricted service in the plan's perimeter list); message retention; who pays.
Google Security Operations (SecOps)	Google's BAA page lists SecOps SIEM and SOAR as covered [73]. Cloud Audit Logs can be ingested by a direct, real-time Cloud Logging filter or by routing to Cloud Storage for scheduled pull [196].	Pricing was not read. Direct ingestion starts when the filter is created. I did not confirm that Ops Agent guest logs are accepted on the direct path; the Cloud Storage route may be needed.
No SIEM at launch; native alerts and weekly review	The Event Logging Standard's SIEM requirement is scoped by the location's chief security officer [190].	A written scoping decision, and a date by which it is revisited.

10.4.3 Security Command Center

The pricing page for Security Command Center (SCC) lists three tiers (the tiers page adds a Standard-legacy column): Standard at no additional charge, Premium, and Enterprise, which is marked deprecated and scheduled to shut down on 2027-05-21 with automatic move to Premium [197]. The tier comparison lists Event Threat Detection (which monitors Cloud Logging for threats such as malware, cryptocurrency mining and data exfiltration) and Virtual Machine Threat Detection as Premium and Enterprise features, and not Standard [197]. SCC appears on Google's HIPAA covered list [73]. Premium can be activated at project or

organization level. At organization level it is a fixed-price subscription (5% of the larger of committed spend or projected run rate, with a minimum of $15,000 a year) or pay-as-you-go. At project level it is pay-as-you-go only, with listed rates of $0.0071 per Compute Engine vCPU-hour and small per-operation rates for Cloud Storage [3]. The plan's cost note calls SCC Premium "unpriced"; the price can now be bounded. Three pilot VMs of 8, 16 and 32 vCPUs running around the clock for a 730-hour month would carry about $290 of Compute Engine charge at the project-level rate (56 vCPUs times 730 hours times $0.0071), before Cloud Storage operations and with idle stop lowering it. That is an upper bound for compute under the listed rate, not a quote, and I did not establish whether a project-level activation depends on any organization-level setting. One AI-assisted literature review calls SCC Premium required for P4; I found no UC or HIPAA text that says so, so it is a risk decision for the security officer.

10.5 Incident response and breach clocks

10.5.1 Process

UC's Incident Response Standard puts the decision to notify affected individuals or regulators with the Location's Lead Location Authority, who consults leadership, counsel and compliance and privacy, and requires significant incidents to be escalated under the UC Cyber Incident Escalation Protocol (I did not read the protocol itself). It also requires that the Location's incident response plan be tested at least once a year, with results documented, gaps identified and action items tracked to completion, and says that responding to a real incident does not count as a test [198]. HIPAA's Security Rule requires procedures to identify and respond to suspected or known incidents, to mitigate harmful effects and to document incidents and their outcomes [181]. NIST's current incident response publication (SP 800-61 Revision 3, April 2025) and its exercise guide (SP 800-84) are the usual references for building and exercising the plan [199, 200]. SATRE, the UK specification for trusted research environments, includes a mandatory item that a TRE log, track and resolve deviations, incidents and audit findings [55].

The plan's runbook R9 reads, in full, that the security office is accountable and the systems team responds, and that responders should contain (stop the VM, keep disks and snapshots as evidence), notify per the matrix, preserve logs and hold a post-incident review. That is an outline, and it hides two conflicts. First, snapshots are the evidence, yet the per-study schedule defaults to 14 days of retention and the teardown script deletes all snapshots; an incident found on day 12 of a study's last week needs a hold step that stops both. Second, evidence preservation can collide with a DUA's destruction clause, and the plan does not say who decides. A review of this runbook by the security office, counsel and the privacy office belongs with the matrix the plan requires before PHI.

10.5.2 Alerts, knowledge and the clock

HIPAA treats a breach as discovered on the first day it is known, or by exercising reasonable diligence would have been known, to any workforce member or agent other than the person committing it [201]. A real-time alert delivered to a queue that nobody reads over a weekend could be argued to be knowledge that reasonable diligence would have produced. That reading is mine, not case law, and counsel should confirm it. The design consequence does not depend on it: the alert queues need an owner on duty or an explicit statement of response hours, and the plan marks on-call and after-hours escalation as open.

10.5.3 Clocks

Table 10.5 lays the clocks side by side. Chapter 3 sets out the rules; the table adds timing and what starts each clock. Non-HIPAA, non-state-law periods are reported or contractual and vary by agreement.

The table shows why the matrix the plan requires before any PHI matters more than any single number. The HIPAA clock is the longest and the one least likely to bind: a DUA window of 24 to 72 hours, where one exists, will pass in a different decade of an investigation. California's 30 days sits between them and depends on how counsel reads the two sections for a campus. For FERPA data, the only external clocks are state law and the agreement, which is why the plan's breach matrix needs a FERPA row that records both. The operating rule for each study should be the shortest applicable clock, starting from the earliest moment anyone on the team knew, with a named person responsible for writing the start time down; Chapter 3 reaches the same rule.

Logs can rebut a HIPAA breach presumption only if Data Access and guest records exist and arrive

The entity must show low probability of compromise to avoid notification, and carries the burden of proof [182, 183]. Data Access logs are off by default and the plan turns them on in Terraform. For bucket data they can show which object a principal read. For data on a VM disk they cannot, and the guest rules record copy-tool use and session identity, not file reads. If the sink is down for three weeks, the logs for that period do not exist anywhere but the VM, and the presumption stands.

Open items before PHI for incident evidence

The environment would answer most breach-assessment questions for bucket data and for control-plane changes. It would answer fewer for disk data, and the gap is stated here so the Data Security Plan and the DUA review can account for it (for example, by relying on the absence of any route off the VM, the export-bucket trail and the agreements). Open items before PHI: a tested privilege-use and login alert path; alerts for Shielded VM, metadata and sink-failure events; an on-call rota in the security office or stated response hours for the alert queues; a hold procedure that stops snapshot expiry and teardown; a rewritten R9 reviewed by the security office, counsel and the privacy office; and a signed matrix with a FERPA row. Which California statute applies, whether UCR acts as a business associate for these studies and the weight of the UC escalation protocol are questions for campus counsel and the privacy officer.

10.6 Proposed tabletop exercises

UC's standard requires the Location's incident response plan to be tested at least once a year [198]; our plan doesn't yet say how this environment takes part, and it schedules no exercise of its own. I propose a tabletop as a gate before first PHI. The scenarios in Table 10.6 are proposals, written to exercise the specific design choices discussed above. Scenario 1 adapts a general scenario from one AI-assisted literature review (a phished researcher whose session is used to stage clinical data in an export bucket), and scenarios 2 to 8 are mine. Each should be run with fake data in the pilot, with a security analyst, a systems team engineer, the research computing lead, the privacy officer, counsel, a PI and a data steward in the room, and the campus FERPA officer for scenarios that involve student records. Each ends with a written record of the exercise, gaps found and action items tracked to closure, which is what the UC standard asks of a test.

Table 10.5: Notification clocks that could apply to a study in the planned environment. "Verified" means read in the primary text on 2026-11-01. Which row applies depends on data source, role and counsel's reading.

Regime	Who notifies whom	Clock and trigger	Status
HIPAA, covered entity	Covered entity to affected individuals; to HHS; to media when more than 500 residents of a state are involved	Without unreasonable delay and no later than 60 calendar days after discovery (164.404(b), 164.406(b)). HHS: with the individual notice if 500 or more people, otherwise within 60 days after year end (164.408) [201, 202].	Verified
HIPAA, business associate	Business associate to covered entity	Same outer limit of 60 calendar days after discovery (164.410(b)) [203].	Verified; UCR's role is a counsel question (Chapter 3)
HIPAA limited data set DUA	Recipient to the data provider	The regulation requires the DUA to oblige the recipient to report any use or disclosure not provided for by the DUA of which it becomes aware. It sets no number of days (164.514(e)(4)) [60].	Verified; the provider sets the clock
Typical DUA or state agreement	Recipient to provider or agency	AI-assisted literature reviews state 24 to 72 hours, and one states 72 hours for a New York State Department of Health agreement.	Reported, unverified
NIH controlled access	Approved user to NIH incident team and others	Within 24 hours of when the incident is identified [64].	Verified; such data routes to UCR's separate CUI enclave, so this row is a benchmark only
California Civil Code 1798.82	Business that owns or licenses data to California residents; maintainer to owner	Disclosure within 30 calendar days of discovery or notification, subject to law-enforcement and scope-determination delays ((a)(2)). A maintainer notifies the owner "immediately following discovery" ((b)). More than 500 residents: sample notice to the Attorney General within 15 calendar days of notifying consumers ((f)). A HIPAA covered entity is deemed to meet the content rules of (d) by complying with HITECH 13402(f), but is not exempted from the rest of the section ((e)) [65].	Verified (subdivisions read)
California Civil Code 1798.29	Agency that owns or licenses data	"Most expedient time possible and without unreasonable delay" in subdivision (a), the only part I read [66].	Subdivision (a) read; whether UC is an "agency" under it, and the rest of the section, not checked
FERPA	None in Part 99	No breach-notification section in 34 CFR Part 99 (searched the full text). Disclosures must be recorded under 99.32 [204]. Department guidance says FERPA itself has no specific breach-notification requirements [69]. Duties come from state law and the agreement.	Verified by absence
UC internal	Unit to the location's chief security officer; Lead Location Authority decides external notice	Escalation per the UC Cyber Incident Escalation Protocol, not read [198].	Policy; clock not verified

Table 10.6: Proposed tabletop scenarios. "Tests" names the plan element or evidence gap each scenario exercises.

#	Scenario	Key injects	Tests
1	Phished researcher; the attacker uses the researcher's tunnel and RStudio session and stages study data in the export bucket.	Spike of reads; steward asks about an unexpected request; the DUA is a limited data set with a short window.	Export-read alert, steward reconciliation, containment (stop VM, revoke access), which clock applies.
2	Insider copies data over a remote shell; the DUA forbids copy-out.	Copy-out audit hits found in a weekly search; no alert fired.	Guest telemetry, the copy-out residual risk the security officer accepted, `gui_only` decision.
3	An administrator changes a VM's security configuration; the change is logged but not alerted, and is found at the weekly review.	Admin Activity entry present, no alert.	Table 10.3 rows 1 and 2; whether admin actions are reviewed within the clock.
4	The log sink stops delivering for three weeks (IAM change or perimeter block); discovered at review.	Gap in the log bucket; a PHI incident falls in the gap.	Sink-failure monitoring; four-factor assessment without logs; secure-project `_Default` and `_Required` copies as fallback.
5	The data steward releases a file with small cells or identifiers.	Recipient is outside UCR; DUA names a provider contact.	Release record, recall, multi-party notice.
6	A person is on two studies' groups by mistake, or retains access after leaving.	Found at quarterly review; access logs show reads.	Group-membership audit trail, the one-business-day removal target, commingling terms in the DUA.
7	A package from a mirror runs code in a study VM (Chapter 8); a snapshot is the evidence and the study ends in two days.	Snapshot retention 14 days; teardown script deletes all snapshots; DUA requires destruction.	Hold procedure versus destruction clause; who decides.
8	The deployment identity is misused to delete a study's VM, disks and snapshots and to edit the teardown evidence.	The deployer problem (Chapter 5); evidence storage is not locked.	Restore (RTO one business day, RPO 24 hours), evidence integrity, `iam_change` coverage.

10.7 Design considerations

1. **Fix the evidence home before the lock.** Create a locked Cloud Storage evidence bucket in the log project for review records, teardown evidence and incident extracts, apply a lien under legal or compliance ownership, and keep infrastructure state separate. Decide who owns the log project's billing and closure for the retention period.
2. **Confirm the log retention figure with the records schedule,** and set it exactly (2,192 days or more for six calendar years). State in the Risk Treatment Plan that the figure is a design default and why, citing 164.316(b)(2)(i) for documentation and the UC standard for logs. Ask counsel what a DUA destruction clause means for locked logs, and forbid participant-derived identifiers in bucket names and labels in the DSP template.
3. **Make the privilege-use and login alerts work on the real image.** Point each audit alert at the log source the operating system actually uses, or base it on the audit records, and add a pilot test that triggers each alert, including `iam_change` for project, bucket and instance IAM, `audit_config_change` and `export_read`.
4. **Add one log-based policy for Shielded VM, metadata and service account changes,** routed to the security office, or write down the reasoning for accepting the gap. Add a metric-absence policy for agent heartbeat so that a stopped Ops Agent is noticed between boots.
5. **Wire sink-error notifications to the security office's queue** and add a saved query for Policy Denied and IAP denials, reviewed weekly during the dry-run period and after enforcement.
6. **Widen the weekly review from alerts to queries,** and move group-membership audit logs from the directory into the review. Record where each review record is kept, who covers absences and what counts as an exception.
7. **Measure before choosing a tier.** Run the pilot long enough to record GiB per day by log type, test the double-ingestion effect of the secure project's `_Default` sink, and then decide on exclusion filters, a Cloud Storage Archive copy and an SCC Premium project-level activation (upper bound near $290 a month for the three pilot VMs at list rates) as separate decisions.
8. **Record the SIEM decision in writing,** even if it is "none at launch," with the scoping required by the Event Logging Standard, a review date and the open perimeter question for the Pub/Sub path.
9. **Rewrite R9 into a procedure** with a hold step that stops snapshot expiry and teardown, a decision owner for evidence versus destruction, a clock log, named roles and response hours for the alert queues, and a FERPA row in the breach matrix. Have the security office, counsel and the privacy office review it before any PHI.
10. **Run the first tabletop as a pre-PHI gate** using scenarios 1, 3, 4 and 7, then run the full set within the first year and each year after, tracking action items to closure as UC's standard requires.

Chapter 11

Operating Model, People and Process

Our plan names eight roles and twelve runbooks, and every row of its responsibility matrix (RACI) has exactly one accountable party. Terraform can make a VM that refuses logins when its audit daemon is down, but it cannot decide whether a new postdoc may be added to a study, notice that a graduate student left in June, or read an export for small cells. Most of the controls the plan lists as "process still needed" in its coverage table are jobs that one of those roles would do on a schedule and leave evidence of. So two questions matter: who is accountable for each recurring job, and what record would show that it was done. The plan answers the first with the matrix and the second with templates and a teardown script.

The roles are proposed, the runbooks are drafts, and the plan itself marks several assignments, including the systems team's acceptance of operational ownership, as to be confirmed. I read the operating model as a proposal. People appear only as roles, and the roles are named by function: the principal investigator (PI) completes, maintains and submits the Data Security Plan (DSP); the research computing team guides the process and writes only the technical sections; the campus security office reviews and approves; the systems team would build and operate the environment, a role the plan proposes and that has not yet been accepted.

11.1 Roles and the accountability matrix

11.1.1 The cast

Eight roles appear in the matrix. The research computing team writes and maintains the Terraform, the images and the curated containers, runs intake, drafts the technical sections of each DSP and writes the researcher guide; it would hold no administrator role on study VMs and no role in the log project, so the people who write the code cannot also operate it or erase its record. The systems team would apply Terraform as a deployer service account under a change ticket, patch, rebuild images, restore, own every study's Google groups, remove leavers and run teardown. The security office would approve the platform as one Risk Treatment Plan (RTP), administer the log project, run the weekly administrator-activity review, scan, commission the penetration test, approve each DSP and own incident response. The security officer would sign the risk acceptances, the log retention period, the irreversible lock on the audit bucket and breach notification. Two roles come from IS-3 itself: the accountable data owner for the environment (proposed as a delegate for research IT, not yet named) and the unit information security lead (UISL) named on each DSP. The PI and the per-study data steward, who reviews and releases exports, complete the list.

Two features of the cast matter later. The accountable data owner is accountable for five rows of the matrix and has not been named; the plan lists the owner's identity among its open items. The data steward is the only role that exists once per study, and the checklist says the steward should not be the person producing the results "if possible," which in a small lab may mean a colleague with a full-time job of their own.

11.1.2 The matrix, condensed

The source matrix has 30 activity rows and eight role columns. Table 11.1 merges related rows that share the same accountable party, giving 23 rows, and keeps the plan's letters: A for accountable, R for does the work,

C for consulted, I for informed. I checked the source by script: each of the 30 original rows has exactly one A. Counted by rows, the security office is accountable for eight, the systems team for seven, the accountable data owner, the security officer and the PI for five each. Counted by rows in which a role does the work (R or A/R), the security office has 13, the systems team 10, the research computing team five, the PI four, the UISL two and the steward one. Row counts say nothing about effort, but they show where the plan concentrates the weekly work: in the security office.

Three choices in the matrix matter. First, accountability for adding a person sits with the PI, but the work and the technical control sit with the systems team, and the check that the agreement permits the person sits with nobody (Section 11.5). Second, accountability for the export release sits with the PI, who is also the person whose results are being released. The steward is responsible, and the security office is only informed by an alert and a weekly comparison. The PI is the accountable party for the DSP in general and so the choice is consistent with UCR practice, but it means the plan has no accountable party who is independent of the study for the one control that stands between the data and the outside world; the separation rests on the steward being a different person from the analyst. Third, the matrix has no rows for several jobs that the runbooks assume: appointing and replacing the data steward, owning the Google groups (stated in a companion document, not in the matrix), running the HR separation feed that removes leavers, operating the quarterly restore test, and deciding when a hold stops teardown during an incident. The plan's own statement that the RACI must be signed by the research computing team, the systems team lead, the security office, the security officer and the accountable data owner before any PHI is the point at which these rows should be added.

The RACI gives each of 30 activities one accountable party, but five rows depend on an accountable data owner the plan has not named, and seven rows depend on the systems team accepting a role it has not yet accepted

The source matrix satisfies the one-accountable-party rule in every row, which is the property that matters when something goes wrong. The rule only protects the institution if the named parties have agreed. The accountable data owner is accountable for platform design, second-person review of foundation changes, software requests, funding and the researcher guide, and the plan lists the owner's identity as open. The systems team is accountable for seven operating rows, and the plan marks its operational role as to be confirmed and its commitment as an open item. Until both are settled in writing, the matrix describes an intention.

11.2 The life of a study

Our lifecycle has ten steps (intake, classification, DSP, approval, onboarding, adding and removing people, quarterly review, change events, releases, teardown). Figure 11.1 regroups them into seven stages and shows who acts at each. The build stage is drawn dashed because it happens once for the platform, before any study, and because the RTP and risk acceptances that the security office and the security officer give there are what let each later DSP inherit controls instead of describing them again.

11.2.1 Intake and classification

Intake is a conversation with the research computing team that fills a one-page checklist: study and PI, data sources and agreements (DUA, BAA, IRB number), the security office's classification (P3 or P4), whether each agreement allows a shared, cloud-hosted, UCR-operated environment, whether it forbids copy-out by researchers (which sets `access_mode = gui_only`), whether outside collaborators are involved, the

Table 11.1: Condensed RACI from the plan (draft v3, to be signed). Role labels: Owner, the accountable data owner for the environment; Security, the campus security office; Officer, the security officer; Systems, the systems team; Computing, the research computing team; Steward, the study's data steward. Exactly one party is accountable in each row; "[?]" marks an assignment the plan has not settled. Rows were merged only where the accountable party was the same.

#	Activity	A	R	C or I
1	Platform design and code (Terraform, images)	Owner	Computing	C: Systems, Security
2	Second-person review of foundation changes	Owner	Systems, Security	I: Computing
3	Apply Terraform; onboard a study (module call)	Systems	Systems	I: Computing, Security, PI
4	Patching, image rebuilds, restores	Systems	Systems	I: Security
5	Alerts: boot check, login gate, disk	Systems	Systems	I: Security
6	Alerts on IAM, logging, sudo, export read; weekly review; scans; pentest	Security	Security	C: Systems
7	Log project administration	Security	Security	
8	Platform Risk Treatment Plan	Officer	Security	C: Computing, Systems, Owner
9	Risk acceptance (D3, D4, copy-out); log retention; bucket lock	Officer	Security	C: Computing, Systems, Owner
10	Incident response	Security	Systems, UISL	C: Computing; I: Officer, Owner, PI
11	Breach notification to data sources	Officer	Security	C: PI; I: Owner
12	Data classification	Security	Security	C: Computing, PI, UISL
13	DSP study sections	PI	PI	C: Computing, UISL
14	DSP technical sections	PI	Computing	C: Systems, Security
15	DSP approval; DUA shared-environment and copy-out check	Security	Security, PI (DUA)	C: Computing
16	Add a person (signed addendum and training)	PI	Systems	C: Computing, UISL; I: Security
17	Remove a person (within 1 business day)	Systems	Systems, PI (notify)	I: Security, UISL
18	Quarterly access review	PI	UISL	C: Systems, Security (spot-check)
19	Export release	PI	Steward	I: Security (alert)
20	Software and container requests	Owner	Computing	C: Systems, Security (if egress changes)
21	Teardown and evidence	Systems	Systems, PI (confirm)	C: Computing, Security (sign-off); I: UISL
22	Funding or subsidy (D8) and its liability	Owner [?]	Computing	
23	Researcher guide and support intake	Owner	Computing	C: Systems

	1 Intake	2 DSP	3 Build (once)	4 Onboard	5 Operate and recertify	6 Export	7 Teardown
PI	requests; names steward, end date, people	completes, signs and submits the DSP		signs addendum for each new person	confirms members quarterly; asks for adds	accountable for each release	confirms data returned or destroyed
Research computing	intake checklist	drafts technical sections only	writes Terraform, images, runbooks	researcher guide	software and container requests		consulted
Security office and officer	classifies the data	checks DUA; approves DSP	RTP; officer accepts risks; pilot review	informed	weekly review; spot-check; scans	matches export-read alert to record	signs off
Systems team			applies foundation under ticket	R1: groups, module call, P4-OK check	patch, restore, remove leavers		R10: runs teardown script
Data steward						reviews, downloads, records	
Evidence kept	intake record, security office classification	signed DSP, DUA check	offline plan, review record, signed RACI	ticket, addendum, training, P4-OK log	access review, review record, patch and restore records	release record, export-read log	evidence file, signed record in DSP

Figure 11.1: Lifecycle of a study under the plan, with the role that acts at each stage and the record each stage leaves. Drawn from the plan's lifecycle list, the RACI and the runbooks. The build stage is a one-time platform stage.

VM template, expected data size and duration, tools and software needs, the named data steward, end date, snapshot retention and funding source. The first two lines of the checklist are routing rules. Data that legally requires NIST SP 800-171, which includes NIH controlled-access data since 2025-01-25 [12], goes to UCR's separate CUI enclave, the sibling service for controlled unclassified information; clinical data that originates in the medical school goes to the medical school's own solutions. Chapter 1 sets out the scope and Chapter 9 describes the first day from the researcher's side.

11.2.2 The Data Security Plan

The DSP is the unit of approval. The platform's central operating idea is that the security office approves the secure VM pattern once, as one RTP, and each study's DSP then says that it runs on the service, inherits the platform's controls and records only what differs. The plan supplies a template for the technical sections (computing environment, access control, data handling, incident reporting) in which inherited controls are marked as such and the study fills in template, tools, access mode, data sources, minimum necessary, retention, steward and end date. Table 11.2 lists the translation from DSP field to module setting, which is the interface between the paperwork and the code: the signed DSP is the input to the systems team's ticket.

The division of labor on the DSP follows the UCR practice stated in the plan: the PI owns it, the research computing team facilitates and writes the technical sections, the security office approves, and the systems team is the IT service provider. The security office's approval is described as mostly a fit test (does the study match the pattern?) together with two checks that are specific to the study: whether each agreement allows a shared environment and whether it forbids copy-out. Signers are the PI, every person with data access and the UISL; the security office approves, and the security officer signs only when the office flags the study.

The paperwork carries a cost that the plan does not estimate. A study cannot start until the DSP is

Table 11.2: How DSP content becomes module settings in the plan. The right column names variables in the planned `secure-vm` module and its labels.

DSP section	Module or platform setting
Researchers with data access	Study group membership (owned by the systems team)
Compute needs	Template (cpu, gpu, highmem) and machine or disk overrides
Analysis tools	`gui` list (RStudio, desktop)
Who releases results	The study's steward group
Data sources and transfer in	The inbound bucket; upload method in the researcher guide
Retention and end date	`snapshot_retention_days`, `noncurrent_version_days`, `end_date` label
DUA forbids copy-out	`access_mode = gui_only`
Study, PI and DSP identifiers	Labels `study`, `pi`, `dsp_id` (also used for cost reporting)

approved and every person listed has completed training and signed. Chapter 9 cites a UK evaluation in which over a third of academic respondents waited more than six months for access; the plan sets no target for DSP turnaround, and the pilot tests measure none.

11.2.3 Onboarding, change events and the first weeks

Runbook R1 starts from a DSP approved by the security office with complete signatures, a recorded DUA check, an end date, a named steward and training records for every person. The systems team creates the study group and the steward group (with the systems team as owner, not the PI), adds a module call with the template, access mode, end date, snapshot retention, steward group and labels, has a second person review it, applies it, checks that the VM reports P4-OK in Cloud Logging, sends the researcher guide and records the onboarding in the DSP. Two practices stand out. Making the systems team the owner of the groups means a PI cannot add a person by editing a group. And the module takes its inputs from the signed DSP, so the DSP becomes the authoritative record of who may have access.

R12 lists the events that change a DSP after approval: a new person, a leaver, a new data source or DUA amendment (the security office re-reviews before the data arrives), an extension (a DSP amendment with a new end date, then a label update), a new tool that needs internet or a new container (a software request, with the security office consulted if egress changes), a multi-site collaborator (a UCR affiliate account under the same DSP, decision D13), and a compute change (a ticket, with no DSP change unless data handling changes). The table is short and complete as written. What is missing is a trigger that nobody has to remember: the plan relies on the PI or the research computing team noticing the event. The one automatic trigger specified is the `end_date` label, which the module sets from the DSP and the runbook says will drive a reminder to the PI 30 days before the end. The module and the foundation carry the label, but in the planned configuration nothing yet sends the reminder; the label is an input for a reminder that someone still has to build.

11.2.4 Operate and recertify

Day-to-day work falls into five kinds: access changes, patching, restores, alerts and reviews. Access changes work as follows. A PI who wants to add a person files a ticket that references the signed addendum and the training record; the systems team adds the person to the group; the security office learns of it through the group or IAM change. The addendum is a one-page form in which the new person says they have read the DSP, will not copy study data off the VM by any means, and will report suspected incidents to the security office immediately, signed by the person and the PI. Removal has a clock of one business day from notice, marked for the security office to confirm. The triggers are leaving UCR, leaving the study, withdrawal of the

addendum and the quarterly review; the steps are removal from the group, ending active sessions on the VM and recording the date in the DSP; and a feed from HR separation data is meant to remove leavers from all groups without waiting for a PI. HIPAA frames the same job as a termination procedure for access to ePHI, which the Security Rule lists as addressable, and as a workforce-security policy that is required [181]. The feed does not exist yet; it is a request to the systems team.

The quarterly review exports each group's membership, the PI confirms each person, the UISL signs and the result is filed in the DSP record; anyone not confirmed is removed under the removal runbook. The security office spot-checks, though the plan does not say how many studies or how it picks them.

One of the AI-assisted literature reviews I commissioned lists five areas in which university research enclaves are said to fail audits (log review, recertification and deprovisioning, media sanitization, contingency testing, encryption in transit inside the enclave). The list is the review's own, drawn from sources I could not trace, and its figures for notification and restoration windows rest on proposed HIPAA rule text that has not been finalized (Chapter 3). It is still a reasonable checklist of where recurring human work tends to lapse, and four of its five areas correspond to runbooks R11, R3 and R4, R10 and R7 and R9. The fifth, encryption in transit, is a design property that Chapter 5 covers.

11.3 The runbooks: triggers, owners and evidence

Table 11.3 lists the twelve runbooks with their trigger, the party who does the work (and, where it differs, the accountable party from the matrix), and the evidence the runbook or its template says the run leaves. The evidence column is the one to read. HIPAA requires documentation of the policies and procedures it asks for, kept for six years from creation or last effective date [191], and the UC logging standard expects review records. A runbook whose evidence is undefined is a runbook whose run cannot later be shown.

Reading the table as an auditor would, only three of the twelve runbooks (R2, R5, R10) have a form. Of the nine without one, R1, R3, R4 and R12 at least name where the record goes (the ticket or the DSP record), while R6 to R9 and R11 also name no place to keep it. R9 and R11 are the two that matter most for the HIPAA requirement to "regularly review records of information system activity" and to document incident response [181]. Chapter 10 reaches the same conclusion about the weekly review and the incident procedure. Retention of these records is also unspecified. The log project's audit bucket would hold audit logs for the agreed period, but the review records, the access reviews and the teardown forms sit in the DSP record, whose own retention and storage the plan does not describe. For HIPAA-regulated studies that is a six-year documentation question.

11.3.1 Patching and image rebuilds

R6 combines two cycles. A VM Manager patch job would run monthly across all VMs in the secure project, and a monthly image rebuild (on a builder outside the secure network) would produce a new image family that new VMs use; existing VMs are patched in place. The consequence is that an old study's VM is the product of the image that was current at its onboarding plus patches, and the plan does not say how drift between the two would be detected. A critical vulnerability is to be fixed within seven days, a figure marked for confirmation by the systems team. Two operating details are missing. The runbook has no maintenance window or notice to the PI, although a patch reboot would interrupt long jobs. Nothing guards against a patch reboot, and the idle stop may treat active researchers as idle (Chapter 9), so long jobs have two ways to be cut off. And VM Manager (OS Config) is one of the control-plane services that Google's BAA page did not list by name when checked for Chapter 4; whether that matters is a question for the security office and counsel, not an operating decision. The image build has open issues recorded in Chapter 8.

Table 11.3: The twelve runbooks in the plan: trigger, owner and the evidence the plan says each run leaves. "No template" means the plan names the record but supplies no form. Security is the campus security office; Systems is the systems team. All runbooks are drafts.

Id	Name	Trigger	Owner	Evidence produced
R1	Onboard a study	Approved, fully signed DSP	Systems	Second-reviewed module call and ticket; P4-OK log entry; onboarding note in the DSP record (no template)
R2	Add a person	PI ticket citing signed addendum and training	Systems	Ticket; signed addendum (template); training record; group change in the audit log
R3	Remove a person	Leaver, left the study, addendum withdrawn, review outcome; 1 business day	Systems	Removal date written in the DSP record; ticket (no template)
R4	Quarterly access review	Calendar, every quarter	PI and UISL; Security spot-check	Membership export; PI confirmation; UISL signature in the DSP record (no template)
R5	Export release	Researcher request to the steward	Steward	Export release record (template, one per release); export-read entry in the data-access log
R6	Patch and image rebuild	Monthly; critical CVE within 7 days [confirm]	Systems	VM Manager patch job result; CIS evidence report per image (no template for the patch record)
R7	Restore	Data loss or VM loss; quarterly test on a demo study	Systems	Restore ticket; P4-OK after restore; test record (no template)
R8	Fail-closed VM	Alert: boot check failed or login gate refused	Systems	Alert; boot-check status; reboot and OK entry (no template)
R9	Incident	Suspected incident	Security accountable; Systems responds	Preserved disks, snapshots and logs; notifications; post-incident review (no template)
R10	Study end and teardown	30 days before `end_date`; PI and Security confirm end	Systems with PI and Security	Evidence file kept in the log project; teardown evidence form signed by Systems and Security; attached to the DSP
R11	Weekly admin-activity review	Weekly	Security	Review record with exceptions noted (no template, location unspecified)
R12	DSP change events	New person, leaver, new source or DUA amendment, extension, new tool, collaborator, compute change	Research computing intake, then Security	Amended DSP; re-review record (no template)

11.3.2 Restores

The objectives are one business day to bring a study VM back and no more than 24 hours of data loss, which matches the daily snapshot schedule. R7 creates disks from the latest snapshots of that study only, attaches them to a new VM from the same module call, confirms P4-OK and informs the researcher, with a quarterly restore test on a demo study. The restriction to one study's snapshots depends on the runbook being followed; pilot test 13 would check that a restore cannot attach another study's disk. HIPAA's contingency-plan standard makes a data backup plan and a disaster recovery plan required, and periodic testing of contingency plans addressable [181]. The snapshot retention default of 14 days sets the oldest state that can be restored, and a DUA that requires shorter retention would shorten it. An RTO is a promise to a PI only if there is a person on duty: the plan lists the systems team as on-call primary, marked for confirmation, and the matrix has no deputy.

11.4 Teardown with evidence

Ending a study is the step where a shared environment most easily falls short, because everything in it is persistent by design: snapshots, object versions and logs all outlive the VM. The plan's response is the combination of runbook R10, the teardown script and the evidence form. It is the most developed process in the plan.

11.4.1 The procedure

Thirty days before the end date, a reminder goes to the PI. An extension is a DSP amendment. At the end, the PI confirms that data has been returned or destroyed as the agreement requires and that the final export is done. The systems team then runs the script with the project, study, zone and change ticket, first as a dry run and then for real. The script inventories the study's VMs, disks, snapshots and buckets (selected by the study label) into an evidence file; lifts deletion protection and deletes the VMs; deletes any remaining disks and every snapshot; deletes the study's snapshot schedule; removes every object version in every bucket and then the bucket; deletes the study's service account; re-lists the labeled resources; records whether anything remains and how much; appends a note that audit logs are kept; copies the file to storage in the log project; and exits with an error if anything remains. The systems team then removes the module call and applies, attaches the evidence to the DSP, and the security office signs off. The evidence form adds checks that the script does not perform: that the PI confirmed the data's return or destruction with a date, that the final exports are recorded, that the study and steward groups are deleted and that audit logs are retained, not deleted.

The design has the right shape. Deletion is scripted, so it is repeatable; the dry run is reviewed; the inventory before and after makes the claim "nothing left" checkable against a record; a change ticket ties the run to an authorization; and both the systems team and the security office sign. The script names IS-3 disposal (8.3.2) and the DUA return-or-destroy terms as its purpose, and it has never been run; its first test is pilot test 15.

11.4.2 What the evidence would and would not show

Reading the script and form against the regulations, I have six questions. Each is about the design, not a live system.

What counts as the study's resources. The final check looks only for resources carrying the study label. A resource created without the label, or one that a researcher or administrator made by hand, would not

be inventoried or deleted, and the check would still report that nothing remains. The same is true of an unlabeled disk snapshot made by hand. The check proves absence of labeled resources, which is weaker than the sentence on the form ("VMs, disks, snapshots, buckets incl. all versions").

Soft delete. Cloud Storage soft delete is on by default for buckets, with a default retention of seven days during which a deleted object or bucket can be restored [205]. The script deletes buckets and object versions but does not address soft delete, so for a period after teardown the data may still be recoverable by a project administrator. If a DUA names a destruction deadline, that deadline would need to include the soft-delete period or the study's buckets would need soft delete disabled before deletion. Because of the soft-delete window (Chapter 7), the DSP's destruction date and the teardown date are not the same date.

Failure handling. Some deletion steps are best-effort and ignore errors. The final inventory would still catch remaining labeled disks, snapshots or buckets, but a deletion that is outside the inventory, such as the service account, could fail silently.

Where the evidence sits. The record is kept alongside the infrastructure state in the log project, in storage that is versioned but not retention-locked, and the dry run keeps no record. Chapter 10 makes the same observation: the evidence of destruction sits where the least protected log project data sits. The matter is not abstract, because the form is what a data provider would be shown.

No cryptographic erasure. One AI-assisted literature review recommends crypto-shredding with per-study customer-managed keys as the destruction method. The plan decided against customer-managed keys (D4, a risk the security officer must accept on the security office's recommendation, pending), so deletion is the only mechanism available, and the plan relies on Google's deletion for the physical media. Whether that satisfies a given DUA's destruction clause is a question for the provider and counsel, and a reason to ask for the wording at intake.

Logs and destruction. The script's note that audit logs about the study are kept for the agreed period is correct under the HIPAA documentation rule and the plan's six-year retention. It sits uneasily with an agreement that requires destruction of everything derived from the data. Chapter 10 recommends forbidding participant-derived identifiers in resource names and labels so that the logs hold no identifying content, and asking counsel what a destruction clause means for locked logs. The HIPAA text itself requires policies for the final disposition of ePHI and the media it is stored on [206]; whether log entries about access to ePHI are themselves ePHI is a legal question the plan has not asked.

Six small fixes for teardown

Teardown is the plan's most complete process: a runbook, a dry-run script, a before-and-after inventory, a signed form and a named test. The remaining gaps are specific. The reminder before the end date is only a label; the final check sees only labeled resources; soft delete extends the life of deleted objects by seven days unless changed; the evidence record sits in unlocked storage; group deletion and the PI's attestation depend on the form and not the script; and no one has asked whether a locked log counts against a destruction clause. None needs new technology. Each is a line in the script, the form or the DSP template.

11.5 Student education records at intake

The intake checklist was written around P4 health data. It asks whether each agreement allows a shared environment and whether it forbids copy-out, but it has no field for the questions that student education records raise. UC classifies student education records as P3 [4], but the same guide puts several kinds of student data at P4: sets containing items such as SSNs, human-subject research data with individual identifiers,

research covered by a Certificate of Confidentiality or the Common Rule, and disability or medical information collected from students to provide services. An identified student-records research extract can therefore be P4 without SSNs, and intake should not default student data to P3. FERPA offers no breach rule and no security rule; what it requires is that disclosure of personally identifiable information from an education record meet one of the conditions in 34 CFR 99.31 and, in several of them, be governed by a written agreement that sets how long the data may be kept [207, 208]. Chapter 3 reads the exceptions. For operations, the consequence is five questions to add to intake, summarized in Table 11.4.

Table 11.4: Proposed FERPA additions to the intake checklist, with where each answer would be recorded and which platform setting it drives. The determinations belong to the campus FERPA officer and the security office; the research computing team only records them.

Question	Why it matters	Recorded in or drives
1. Which FERPA exception authorizes this disclosure?	The route decides what the agreement must say: school official (99.31(a)(1)), studies for or on behalf of the institution (99.31(a)(6)), audit or evaluation by a state authority's representative (99.35), written consent (99.30), or de-identified data (99.31(b)) [207, 208]. On my reading, the studies exception does not cover faculty-initiated research unless the institution commissions or adopts the study; the campus FERPA officer decides.	DSP data-handling section; security office classification record
2. What is the destruction date or period in the agreement?	The studies exception requires destruction when the information is no longer needed for the study's purpose; the audit exception requires the agreement to specify the destruction period [207, 208].	`end_date`, `snapshot_retention_days`, `noncurrent_version_days`; teardown schedule including soft delete
3. Does the agreement name the individuals who may access the data?	Agreements may restrict access to named users, so an added person might need an amendment, not only a PI addendum. This is reported practice, not read agreements.	Study group; DSP people list; addendum check in R2
4. Does the provider forbid mixing its data with other providers' data, or hosting it in a shared environment?	A state agency or district may object to co-residence with other data. The terms vary and none was verified; a "yes" is a trigger to consider a dedicated project (Chapter 6).	Security office DUA check; decision record per study
5. Is any of the data NIH controlled-access (such as dbGaP) or CUI?	These require NIST SP 800-171 and route to the separate CUI enclave, not this environment [12].	Intake routing; no DSP here

The first question has the most consequence for design, and it is the one the research computing team and the security office are least placed to answer. The plan's DSP template presumes that the study has authority to hold the data. For health data, the authority is a signed agreement with a covered entity or a limited data set DUA. For student records, the authority is a FERPA exception, and an agreement that does not fit one is not made valid by IRB approval. Intake should therefore end with a recorded determination from the campus FERPA officer and the security office and should not proceed to DSP drafting without it.

The second question links the agreement to platform settings. The destruction period in the agreement should become the `end_date` label and set the snapshot retention and the noncurrent-version window, not the plan's defaults of 14 and 30 days if the agreement requires shorter. Because deleted objects survive soft delete

for seven days by default and because restore points persist until the snapshot schedule's retention expires, the destruction date should be the last day on which any copy might remain, and the teardown date should be set earlier by that margin.

The third question exposes a gap in R2. The runbook for a new person requires a signed addendum and training. It does not require anyone to check whether the agreement restricts who may be named. Where it does, the addendum is necessary but not sufficient, and the person cannot be added until the data provider has agreed. Adding a line to R2 ("agreement permits this person, checked by the PI, recorded in the ticket") costs nothing.

The fourth question belongs to the tenancy argument in Chapter 6. The plan's decision D3 puts all studies in one shared project. A provider whose terms forbid commingling is a case the plan handles only as an open item, with no stated rule. The operating model should record at intake whether a provider has said anything on the subject, so that the answer exists before the first such study arrives.

The fifth question completes the routing rule on the existing checklist. NIH controlled-access data and other material that legally requires NIST SP 800-171 goes to the separate CUI enclave. Intake in this environment simply records the routing and the reason.

11.6 Staffing: what the evidence supports

The plan gives no estimate of operating effort. Its build estimate is three to four weeks; its requests to the systems team and the security office are lists of recurring duties with no hours attached. I found no verified staffing ratio for a shared secure research environment of this kind; the numbers that circulate come from a model, not from measurement.

11.6.1 A literature review's model, labeled as a model

One AI-assisted literature review proposes four staffing blocks (platform and DevSecOps engineering, security and compliance, output checking, and onboarding support), each with an FTE range, and sizes output checking at 1.0 to 1.5 FTE per 50 active projects. It derives that figure from assumptions: two exports per project per week, 30 minutes per review, 40 hours per FTE. The same review cites a ratio of one platform engineer per 20 to 30 studies by adapting a general DevOps ratio that came from practitioner blog posts, and a research-administration ratio from a labor-cost consortium. A second review sizes an enclave for 800 researchers at roughly 320 projects and 2.3 FTE of output checking, using a request rate per project that it attributes to a UK data bank's 2023 figures and a productivity of 2,000 requests per FTE per year that it attributes to an arXiv paper. The arXiv identifier that review cites for that figure points to a different, unrelated paper, so the productivity claim has no source I could check. I did not verify the data bank's request counts either. I don't use any of these figures as fact.

11.6.2 Effort drivers, with arithmetic that can be checked

What can be said is which quantities drive effort, and how sensitive the answer is to the unknown ones. Table 11.5 computes the annual hours of export review for three study counts under two assumptions at each end: a low case (14.2 releases per project per year, which is the figure one review derives from a UK data bank and which I did not verify, with 30 minutes each) and a high case (104 releases per year, which is two per week as assumed in the other review, with 60 minutes each). FTE is hours divided by 2,080. The two cases differ by a factor of about 15, and no measurement in this book narrows the gap.

Table 11.5: Arithmetic on assumed inputs, not a forecast: annual hours of export review by number of active studies. Low case: 14.2 releases per study per year at 30 minutes (review-derived, unverified). High case: 104 releases per study per year (two per week) at 60 minutes (assumed). FTE at 2,080 hours.

Active studies	Low case hours	Low case FTE	High case hours	High case FTE
3 (pilot size)	21	0.01	312	0.15
10	71	0.03	1,040	0.50
50	355	0.17	5,200	2.50

The pilot has three studies, where export review is a task and not a job. At 50 studies it could be either a part-time duty or two and a half positions. The plan's design makes the steward a member of the study's own team, so the cost is spread across labs and is not a central staffing line; that is cheaper for the institution and slower for the researcher whose steward has other work. Whether to centralize is a decision for when volume is known, and the pilot should count releases and minutes per release to find out.

The other drivers don't reduce to arithmetic, so I list them by how they scale.

Fixed costs of the platform, independent of study count. Monthly patching and image rebuilds, keeping the Terraform and images current, weekly administrator-activity review, quarterly restore test, annual RTP review, the penetration test every three years, the SIEM decision and its operation, on-call, and incident response practice. These sit with the systems team and the security office and are the most likely to be paid for from existing staff. They have a minimum size of a person-fraction regardless of whether there are 3 or 30 studies.

Per-study costs. Intake and DSP drafting (research computing team, with security office review), onboarding, additions and removals, quarterly access reviews, software requests, teardown and its evidence. These scale with study count and turnover. The number of additions and removals per study per year depends on lab turnover, which the plan does not estimate.

Per-request costs. Export reviews, software and container requests (each new R package is a container rebuild, Chapter 8) and restore events. These scale with research activity.

Sources of variability. DSP turnaround and the number of review cycles; how many studies use `gui_only`; how many have DUAs that need individual review; and the rate of alerts, which the pilot can measure.

11.6.3 What the literature adds

Two Zenodo presentations bear on this. One describes the UK's experience building and linking trusted research environments and notes that the work needs more than software engineers: managing cloud infrastructure, understanding statistical tests and the limits of anonymization, specifying metadata, and, alongside, information governance specialists, project managers and community managers [209]. The other describes the UK Safe Data Access Professionals network, set up so that organizations that provide secure access to controlled data can share practice, and notes that outputs undergo disclosure checks before release [210]. Neither gives a ratio. They support a qualitative point: operating a secure environment draws on governance and statistical skills as well as engineering, and peer networks exist to share that expertise. The plan's operating model has the engineering and security roles well defined and the governance and statistical role (the steward) thinly defined.

No verified staffing ratio exists for this kind of environment, so the plan should measure effort in the pilot and not borrow a ratio

The staffing figures in circulation are one literature review's assumptions, built on a per-project request rate and a per-FTE productivity figure that I could not trace to a primary source. Varying the two unknown inputs across plausible ranges changes the export-review workload at 50 studies by a factor of about 15 (Table 11.5). The pilot's fifteen acceptance tests measure whether controls work and none measures how long operating them takes. Adding time-to-complete for onboarding, a release review, a removal and a restore would turn the next staffing discussion from a model into a measurement.

Where staffing will break first

The plan lists duties for the systems team and the security office but assigns no hours, and the security office carries the most rows in the matrix (13 as responsible, eight as accountable). A three-study pilot will not stress this. A fifty-study service might, and the single-point roles are the ones to watch: one security analyst for the weekly review, one systems on-call primary, and one data steward per study. The plan does not name deputies or hand-offs for any of them. The decisions that would change the load most are whether stewards are central or per-lab, whether the HR separation feed is automated, and whether DSP approval takes days or weeks.

11.7 Design considerations

1. **Get the RACI signed with the missing rows added.** Before any PHI or student records, add rows for appointing and replacing data stewards, owning the Google groups, running the HR separation feed, running the quarterly restore test, and placing a hold on teardown during an incident. Name the accountable data owner, and record the systems team's acceptance of its operating role and the seven rows in writing.
2. **Separate accountability for export release from the study.** Decide whether the security office or another party independent of the PI should be accountable for the export control, or record that the plan accepts the PI as accountable with the steward as the independent check, and say what would trigger a central steward pool.
3. **Supply a form and a home for every run.** Add templates for the patch record, restore test, fail-closed event, incident and weekly review (R6 to R9, R11), and simple record formats for onboarding, removal, the quarterly review and DSP changes (R1, R3, R4, R12); decide where review and DSP records are stored, and state their retention against the HIPAA six-year documentation rule.
4. **Build the reminders.** Implement the 30-day end-date reminder (a scheduled job that reads the `end_date` label and opens a ticket), an alert for studies past their end date, and a quarterly calendar entry for access reviews. A label is not a reminder.
5. **Harden teardown evidence.** Make the final check inventory by project and not only by label for compute, disk, snapshot and bucket resources created in the study's zone or named with its prefix; address soft delete explicitly (disable it or add its period to the destruction date); do not let any deletion step fail silently; write the evidence to a locked evidence bucket (Chapter 10); and add group deletion and the PI attestation to the script's checklist or to its output.
6. **Add the FERPA questions to intake.** Use the five questions of Table 11.4, end intake with a recorded determination from the campus FERPA officer and the security office, and keep the DSP template's destruction wording tied to the agreement's date.

7. **Check named-user terms in R2.** Add to the add-a-person runbook a check that the agreement permits the person, recorded in the ticket, and ask each data provider at intake whether it restricts named users.
8. **Give patching a maintenance window.** Specify notice to PIs, how long jobs are protected, and what happens to a GPU job during a patch reboot; add a check for drift between the current image family and patched VMs.
9. **Name deputies.** For the weekly review, the on-call role, the data steward and the quarterly access review, write down who covers absence and for how long, and how coverage is recorded.
10. **Measure effort in the pilot.** Record the elapsed and working time for DSP approval, onboarding, an added person, a removal, a release review, a software request, a restore and a teardown, and the number of alerts per week. Use those measurements, not a ratio from a model, for the first staffing request, and revisit it at 10 studies.

Chapter 12

Cost and Funding

The plan carries one paragraph on cost. It gives three per-VM figures (about $355, $910 and $1,290 a month for the CPU, GPU and high-memory templates running around the clock), a subsidy arithmetic for 5, 15 and 30 studies, a statement that the foundation and log projects are "small" (under $100 a month), and a note that Security Command Center Premium is unpriced. The note calls itself a panel estimate to be checked against the billing catalog, so I checked it. I recomputed each template from Google's published list prices for Iowa (us-central1), on demand, over a 730-hour month, as read on 2026-11-01, and looked at what changes when VMs stop while idle. Nothing here is a quote. UCR's negotiated rates, any credits and Google's own price changes will move every figure. The calculation script sits with the chapter so anyone can rerun it.

The harder question is who pays. Decision D8 says the environment may be subsidized under the campus no-charge tier. The stated rule says P3-P4 projects pay all of their costs through a chart of accounts (COA), a memorandum of understanding (MOU) and a separate billing account (Chapter 1). Those give different bills to different people. The second also raises questions our Terraform doesn't answer yet: how one shared project produces per-study invoices, and which costs belong to no study. I price both positions and the federal cost principles that bear on a recharge. The choice stays with research computing and campus leadership.

12.1 Method and price sources

Every unit price comes from a Google page read on 2026-11-01. VM prices are the Iowa rows of the N2 and G2 tables: $0.388472 per hour for `n2-standard-8`, $2.096224 for `n2-highmem-32` [211] and $1.147208 for `g2-standard-16`, which includes its one NVIDIA L4 [212]. Balanced persistent disk is $0.000136986 per GiB-hour ($0.10 per GiB-month), and standard snapshot storage in us-central1 is $0.000068493 per GiB-hour ($0.05 per GiB-month) [213]. Cloud Storage Standard comes to $0.02 per GiB-month, the rate Chapter 10 uses [2]. The templates' machine types and disk sizes are read from the module: CPU is 8 vCPU with a 50 GB boot disk and a 500 GB data disk, GPU is 16 vCPU with an L4, a 100 GB boot disk and a 1 TB data disk, and high-memory is `n2-highmem-32` with a 50 GB boot disk and a 1 TB data disk. All disks are `pd-balanced`.

The calculation excludes sustained-use and committed-use discounts, so it is a list-price result. Google's page says N2 machines earn sustained-use discounts of up to 20% and that accelerator-optimized machines, which include G2, do not [212, 214]. At 24-hour operation an N2 VM would therefore cost up to about $57 less for the CPU template and $306 less for the high-memory template than the figures below. A VM that runs only part of the month earns less. I left the discount out because the lower bound depends on how Google meters it across a project, and I did not model that.

Four inputs are my assumptions, not published numbers, and I flag each where it appears. Snapshots hold half of the provisioned disk, times 1.3 for 14 days of daily change. Each study's buckets hold as much data as its data disk. The idle-stop uptime is 40% of the month, which is the plan's own figure and about 292 hours. Audit-log volume, which drives the largest variable platform cost, is unmeasured (Chapter 10), so the platform table shows two illustrative volumes.

The CIS Hardened Image fee needs more care, because I could not find a published price for the instance sizes in the plan. CIS's knowledge-base page says the software charge is $0.01 to $0.06 per hour depending

on core size, billed only while the instance runs [215]. The Marketplace listing for the Rocky Linux 9 Level 1 image shows a default usage fee of $16.06 a month for an `e2-small` running 24 hours a day for 30 days [216], which is about $0.0223 an hour. I used $0.0223 an hour for the low case and $0.06 for the high case, and charged neither to the GPU template, which the plan builds from Google's Rocky 9 NVIDIA image. One of the AI-assisted research reports prices the fee "per vCPU-hour" at $0.02 to $0.06; CIS's page says per instance-hour by core size, so that unit is not supported.

12.2 Cost of the three VM templates

Table 12.1 shows the monthly list-price bill for one study VM of each template, with the CIS fee at its low value, and the same bill at 40% uptime. A sustained 24-hour month for the CPU template comes to $383. Compute is 74% of it; disks, snapshots and buckets together are $83. The GPU template is $1,003 and the high-memory template $1,706.

Table 12.1: Monthly list-price cost of one study VM by template, USD, Iowa (us-central1), on demand, 730 hours, prices read 2026-11-01. CIS fee at the low value ($0.0223 per hour; none on the GPU template). Snapshots and buckets use the assumptions in Section 12.1. Components are rounded to the nearest dollar; totals are computed from the unrounded rates and then rounded, so a row can differ by $1 from the sum of its displayed parts (high-memory: 1,530.24 + 16.28 + 105.00 + 54.12 = 1,705.65, shown as 1,706). The last column is SCC Premium at the project-level rate of $0.0071 per vCPU-hour, not included in the totals [3]. Script: `figs/ch12_cost_calc.py`.

Template	Machine	VM	CIS	Disks	Snap. + buckets	Total 24x7	Total 40%	SCC 24x7
CPU	`n2-standard-8`	284	16	55	28	383	203	41
GPU	`g2-standard-16`	837	0	110	56	1,003	501	83
High-memory	`n2-highmem-32`	1,530	16	105	54	1,706	778	166

With the high CIS fee ($0.06 an hour) the CPU template is $410 at 24 hours and $214 at 40% uptime, and the high-memory template is $1,733 and $789. The fee is the least certain price in the table, and at the high value it is $44 a month on a CPU VM running continuously. I'd settle it with CIS or through the Marketplace listing before we publish a rate.

The plan's CPU estimate holds; its high-memory estimate is about a fifth too low

On the plan's stated basis (VM, disks and CIS fee, 24 hours a day, no snapshots or buckets) the recomputed figures are $355 for the CPU template, $947 for the GPU template and $1,652 for the high-memory template, against the plan's $355, $910 and $1,290. The CPU figure matches to the dollar. The GPU figure is 4% higher than the plan's. The high-memory figure in the plan is below the `n2-highmem-32` VM charge alone ($1,530), so it cannot be a list-price number for the template as written.

The gap probably comes from a different machine size or disk in the estimate; I couldn't tell which. The plan's subsidy arithmetic uses only the CPU figure, so the error does not reach those totals; it would reach any rate card that copied the high-memory number.

One peer figure is close to this calculation. Stanford's Nero page lists a compute instance with 8 vCPUs, 30 GB of RAM and 100 GB of disk at about $290 a month without discounts [32]. The same shape on Google's list prices (an `n2-standard-8` running all month plus a 100 GB balanced disk) is $294 before any CIS fee. Stanford does not say which machine type it priced, so the agreement suggests continuous list pricing rather than proving it. The page adds that Stanford's negotiated discount is at least 18% and that ingress and egress fees are waived, neither of which UCR has stated for this environment.

12.2.1 Idle stop and uptime

The CPU, GPU and high-memory templates default to stopping when no session, no RStudio or desktop session and no heavy process has been seen for 120 minutes (240 for the GPU example in the pilot), checked every five minutes by a timer in the image. A stopped instance stops compute charges but, in Google's words, you continue to incur charges for disks, snapshots and external addresses that remain [217]. The CIS fee is billed only while the instance runs [215]. Storage is therefore a floor: $83 a month on the CPU template however little it is used.

Figure 12.1 shows the effect. At 40% uptime the CPU template falls from $383 to $203, a reduction of 47%, not a half, because $83 does not move. The high-memory template falls from $1,706 to $778 (54%) because compute is a larger share. The 40% figure is generous for an analysis workload. The cost and funding AI-assisted research report assumes about 40 hours of active analysis a week, which is 160 to 175 hours a month, or 22 to 24% of a 730-hour month; the script gives the CPU template at $149 for that uptime. The report's figure is itself an assumption. The pilot has no test that records hours of use, so none of these uptimes is measured. The billing export, which the plan creates a dataset for, would give actual figures within the first month.

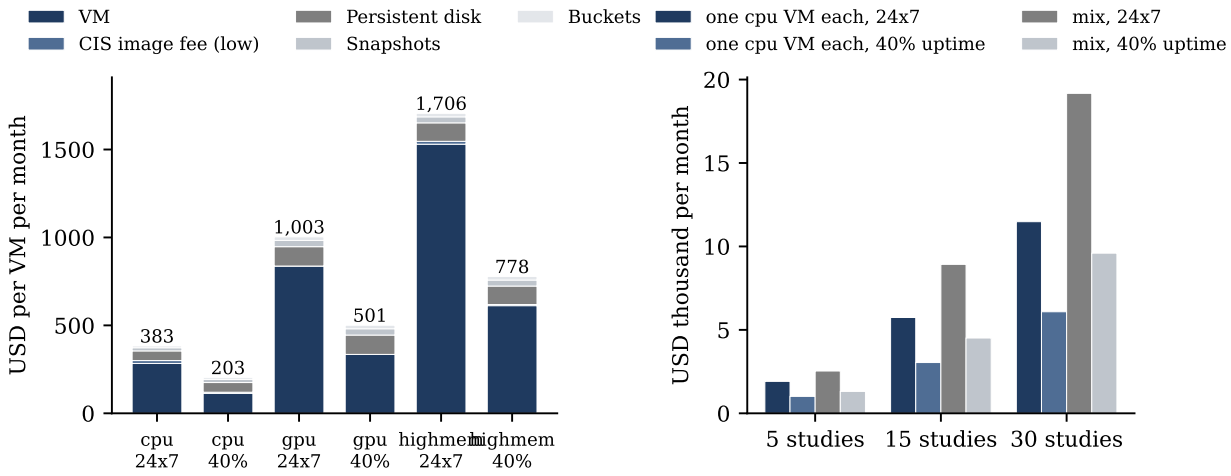


Figure 12.1: Left: monthly list-price cost of one VM per template, at continuous operation and at 40% uptime, split by component (Iowa, on demand, prices read 2026-11-01; CIS fee at the low value). Right: monthly cost of a subsidy that covers VMs, disks, snapshots and buckets for 5, 15 or 30 studies, for one CPU VM per study, and for an assumed mixed fleet, meaning a guessed mix of templates rather than a forecast: 4 CPU and 1 GPU VM for 5 studies; 11 CPU, 3 GPU and 1 high-memory for 15; 21 CPU, 6 GPU and 3 high-memory for 30. Each is shown at continuous operation and at 40% uptime. No demand data exist. Platform fixed costs are excluded. Source: `figs/ch12_cost_calc.py`.

Idle stop has a cost the arithmetic does not show. A VM that has stopped must be started before a researcher can work, and the researcher-experience chapter (Chapter 9) treats that wait as part of the login path. A subsidy or rate schedule that depends on 40% uptime also depends on researchers accepting the delay and on the idle detector working, which pilot test 12 checks. As written it may not work: because of the user-ID gap in Chapter 9, a detached job would probably not keep a VM awake and a VM could stop under running work. Uptime would then be lower than 40%, and the saving larger, for the wrong reason.

12.3 Platform costs that belong to no study

Table 12.2 lists the costs of the foundation and log projects and of the shared services around them. The priced items are small. The three private DNS zones in the foundation are $0.20 a month each at the rate for the first 25 zones [218]. Artifact Registry storage is $0.10 per GiB-month after a free half gibibyte [219]; 50 GiB of mirrored packages and containers would cost about $5. Alerting is free today; from no sooner than 1 September 2027 Google will charge $0.35 a month per metric reference [1]. Eight of the plan's nine alert policies are log-match conditions and the ninth (disk full) is a metric threshold, so the charge would be about $0.35 to $1 a month. The Terraform state bucket is negligible.

Log storage is the item that grows with time. Chapter 10 established the price of logs: $0.50 per GiB ingested after the first 50 GiB a project receives each month, and $0.01 per GiB-month beyond 30 days in a log bucket [1]. The volume is unmeasured. At 20 GiB a month the whole platform stays inside the free allowance and the 72-month retention would add about $14 a month by the end; at 100 GiB a month ingestion costs $25 and storage reaches about $71 a month after six years. Chapter 10 found that the plan's sink probably ingests each entry twice, once into the secure project's default bucket and once into the log project, so the real ingestion line may be higher; each project has its own free allowance.

The lower half of the table matters more. The GCP bill of the shared platform is of the order of tens of dollars a month. SCC Premium, if the campus security office decides it is needed, is the only shared line over $100 at list price, and Chapter 10 bounded it from Google's published rates [3]. The costs that matter are the unpriced rows, which are people's time and one-off work. The plan names several of them in its list of agreements required before PHI (a funding statement covering who pays for incident response, the pentest and log storage), but puts no figure on any.

The shared platform's cloud bill is small, so the funding fight is about staff time and risk, not the GCP invoice

Priced at list, the foundation and log projects cost roughly $6 to $102 a month before SCC Premium and any Filestore cluster. At 5, 15 or 30 studies the per-study share of that is dollars. The shared costs that would decide a rate are security review time, systems operation, steward effort, the pentest and incident response, and the plan prices none of them. Any funding model that quotes only GCP prices understates what the service costs UCR.

The cost and funding AI-assisted research report offers a model that tries to price these. For 20 studies it puts the isolated project-per-study design at $23,331 a month and the shared design at $6,915, and for 50 studies $58,178 and $9,584, with a ratio of engineering staff to studies behind the difference. I treat these as the report's own model, not as findings. By the report's table, $15,000 of the $16,417 monthly difference at 20 studies (91%) is the labor line, which rests on an assumed $125 hourly rate and assumed hours per study (8 hours for each isolated project against 40 hours for the central team). A further $608 comes from Cloud NAT gateways at $32 a month each, which the plan does not use: its network has no Cloud NAT and no internet route. That leaves roughly $800 a month of the difference from cloud prices. The model therefore says little about GCP economics. It says that if per-study projects need eight staff hours each every month and a shared project needs none beyond a fixed team, the shared design is cheaper. Chapter 6 weighs tenancy on its other merits. The price check adds one fact: the per-project fixed charges at list price (DNS, registry, alerts) are too small to push the decision either way.

12.4 The funding question

Table 12.2: Platform costs outside any single study, USD per month at list price, prices read 2026-11-01. Quantities in italics are my assumptions. "Owner" records who the plan or the funding rule would have to name; none is named today.

Item	USD per month	Basis and open question
Cloud DNS, 3 private zones	0.60	[218]
Artifact Registry storage	about 5	*50 GiB* mirrored packages and containers [219]
Monitoring alert policies	under 1 (from Sept 2027)	8 log-match policies and 1 metric threshold; alerting is free until no sooner than 1 September 2027 [1]
Audit log ingestion and storage	0 to 25 ingest; 14 to 71 storage by year six	*20 and 100 GiB per month*, unmeasured; possible double ingestion (Chapter 10)
Security Command Center Premium	up to about 290 for three 24-hour pilot VMs	Project-level rate, compute only; activation scope unconfirmed (Chapter 10); the table's "SCC" column gives it per VM
Filestore for a Phase 2 cluster	about 197	Basic HDD, 1 TiB minimum, at the Iowa (us-central1) list rates: $196.84; the plan says about $200 [220]
Billing budget, billing export dataset	not priced	Budgets and the export are described on Google's budget and billing pages; I did not find a charge for budgets and did not price the export dataset [221]
Total of priced items, excluding SCC and Filestore	about 6 to 102	Consistent with the plan's "under $100" except at the high log volume near the end of six years' retention; the alert charge starts no earlier than September 2027
Security review, alert triage and log reading	not priced	Owner: campus security office. Staff time
Platform operation, patching and image rebuilds	not priced	Owner: systems team. Staff time
Data steward effort	not priced	Owner: per study (Chapter 7, Chapter 11)
Penetration test before first P4 data	not priced	Plan lists it among agreements and tests
Incident response and notification	not priced	Plan asks who funds it

12.4.1 What the two positions say

The plan says the environment is possibly subsidized under the campus no-charge tier, with the decision open, and that a separate billing view would show what the subsidy costs. Its Terraform sets a monthly budget with alerts at 50, 90 and 100% and carries a placeholder default of $2,000 that its own description marks to be decided. The research computing team's gap analysis records the rule that P3-P4 projects pay all costs through a COA, an MOU and a separate billing account. Chapter 1 named the inconsistency, and I can't resolve it here; I can price both sides.

Under the second position the unit of billing is a study. The plan's module puts a `study` label on each VM, data disk, bucket and snapshot policy, and the foundation creates a BigQuery dataset for a billing export keyed on that label. Google's billing documentation lists labels among the fields of the standard usage cost export [222]. A monthly per-study statement could be built from it. Three gaps remain for the pilot to test. The export itself is switched on once in the Billing console, not by Terraform. Labels on snapshot storage and on log ingestion may not carry the study name, and I did not confirm that they do. And everything outside a study (Artifact Registry, DNS, logs, SCC, Filestore) has no study label at all and would have to be allocated by a rule.

A separate billing account has its own mechanics. A Google Cloud project is billed to the Cloud Billing account it is linked to, and moving it requires unlinking and relinking [223]. A separate account for all P3-P4 projects is therefore a property of the secure project, not of each study. Per-study invoicing within it is a chargeback process run by whoever manages the account, using the labels. The plan's budget is also an alerts-only budget: Google's page distinguishes alerts-only budgets from spend-cap budgets, and the Terraform creates the former [221]. A per-study ceiling that actually stops spending would need another mechanism, such as the end date already carried as a label, or an alert that the systems team acts on.

12.4.2 What a subsidy would cost

Table 12.3 gives the monthly cost of subsidizing the VMs, disks, snapshots and buckets of 5, 15 and 30 studies, using the plan's own simplifying case (one CPU VM per study) and a mixed fleet. The mixes are my assumptions to show sensitivity to GPU and high-memory use, not forecasts. The CPU case at continuous operation reproduces the plan's figures to within 9%: $1.9 thousand, $5.7 thousand and $11.5 thousand against the plan's $1.8 thousand, $5.3 thousand and $10.7 thousand, the difference being the snapshots and buckets that the plan's estimate omitted.

Table 12.3: Monthly cost of a subsidy covering VMs, disks, snapshots and buckets (USD, list price, CIS fee at the low value, Iowa, prices read 2026-11-01). Platform fixed costs (Table 12.2) and staff time are excluded. Mixes: 4 CPU and 1 GPU (5 studies); 11 CPU, 3 GPU, 1 high-memory (15); 21 CPU, 6 GPU, 3 high-memory (30). Source: `figs/ch12_cost_calc.py`.

Studies	CPU each, 24x7	CPU each, 40%	Mix, 24x7	Mix, 40%	Mix, 40%, per year
5	1,914	1,014	2,534	1,312	15,744
15	5,741	3,042	8,925	4,511	54,132
30	11,482	6,085	19,174	9,597	115,162

The range is wide. For 15 studies the subsidy is $3.0 thousand a month if every study uses one CPU VM at 40% uptime and $8.9 thousand if the mix includes GPUs and a high-memory machine running continuously, a factor of three. Chapter 13 discusses GPU demand. One GPU study at continuous operation costs about as much as two and a half CPU studies, and that a subsidy without a per-study ceiling lets one study set the bill.

12.4.3 Peer funding models

Chapter 2 reports that peers that subsidize a baseline and state their hosting mostly run on-premises systems, and that most of the cloud-hosted offerings it could verify pass costs through (Johns Hopkins SAFE Desktop, a cloud-based desktop that is free to 100 GB, is the exception) [23]. Stanford bills Nero to the PI's project-task-award number at list less at least 18% [32]. UCSF's RAE Premium charges a flat monthly fee per dedicated VM: $300 for 4 CPUs and 16 GB, $700 for 8 CPUs and 32 GB or more, $1,300 for 16 CPUs and 64 GB or more, with hosting not stated on the page [31]. Michigan's Secure Enclave Services page lists cost-recovery rates of $7.00 per GB of RAM, $3 per GB of GPU RAM and $15.41 per TB of storage per month [21]; an 8-CPU VM with 32 GB would be $224 a month for memory before storage. These are not like-for-like. They do show that the 8-CPU, 32 GB template at $383 continuous and $203 at 40% uptime is inside the range peers publish, and that peers price the GPU separately. The figures are peers' published prices, not their costs, and whether any of them recovers staff time is not stated on the pages.

12.4.4 Federal cost principles

If a P3-P4 study charges its cloud use to a federal award, two parts of the Uniform Guidance matter. 2 CFR 200.413(b) says costs that would otherwise be indirect may be treated as direct if directly related to a specific award, and gives as examples "services rendered by specialized facilities, cybersecurity, integrated data systems" and other institutional service operations [224]. A cloud VM purchased for one study fits that language more easily than a shared platform's fixed costs. What UC treats as direct is for the campus research-administration office to decide.

If the campus runs the service as a specialized service facility and bills awards for it, 2 CFR 200.468 applies. The section covers "highly complex or specialized facilities" and names computing facilities. Material costs "must be charged directly to the applicable Federal awards based on actual usage of the services on the basis of a schedule of rates or established methodology that" does not discriminate between federal and other activities, including internal use, and "is designed to recover only the aggregate costs of the services," normally the direct costs plus an allocable share of indirect costs, with rates adjusted at least every two years and considering the over- or under-applied amounts of earlier periods [225]. Three consequences follow for the design. A subsidy that covers some users but not others is permitted only if the rate schedule itself is the same for federal and non-federal users; a free tier for internal users and a charge to awards is the pattern the text excludes. A rate must be tied to usage, which fits metered VM hours better than a flat fee. And fixed platform costs may be recovered only as part of the aggregate cost base, with a reconciliation step. NIH adds that budgets for data management and sharing must not include infrastructure already in institutional overhead and that costs may not be charged as both direct and indirect [226]. If part of the platform were funded from F&A, charging the same part to grants would double-charge; the campus would need to know which.

The cost and funding AI-assisted research report states three federal points that the current text does not support as written. It describes a break-even "corridor" of plus or minus 15% as a requirement of 2 CFR 200.468; the current text has no such figure [225]. It gives a $5,000 equipment threshold for capital items, while the current definition in 2 CFR 200.1 is the lesser of the recipient's capitalization level and $10,000 [227]. And it calls the 26% cap on administrative costs statutory; the cap is regulatory, in 2 CFR 200 Appendix III, section C.8.a [228], and it limits the administrative component of F&A, not a recharge rate, so it does not bear on the pricing here. Cloud usage is also an operating expense, not equipment, so the equipment threshold is unlikely to matter here.

12.4.5 The options side by side

Table 12.4 sets out five ways to fund the service. It describes consequences, not a recommendation. The choice belongs to research computing and campus leadership, with the campus research-administration office on the federal rules and the campus security office on whether any shared cost should be a security-office budget item.

Table 12.4: Funding options for the shared environment. Dollar figures are from Tables 12.1, 12.2 and 12.3.

Option	Who pays what	What it needs	Open questions
A. Full pass-through (COA/MOU, as stated)	Each study pays its VMs, disks, snapshots and buckets, and a share of shared costs, from its own COA.	Per-study labels and billing export; a rate or allocation rule for shared costs; an MOU template; a monthly statement.	Who funds the shared costs before the first study signs; what rate covers staff time; a researcher without a grant cannot pay.
B. Full subsidy (no-charge tier, as D8 reads)	Campus program pays all GCP costs: $1.0 to $2.5 thousand a month at 5 studies, $3.0 to $8.9 thousand at 15, $6.1 to $19.2 thousand at 30.	A named funding source and exception to the stated rule; a per-study ceiling; a way to stop spending.	What happens when the subsidy is spent; which studies qualify; whether the project-pays rule needs an exception on record.
C. Platform central, studies pass through	Campus funds shared costs; each study pays its own VM, disk and storage.	Study labels (already planned); an owner for shared costs.	Peers' subsidized baselines, mostly on premises, resemble this; the shared GCP cost is small, so the central share is mostly staff time.
D. Capped subsidy	Campus pays up to a per-study monthly or lifetime ceiling; the study pays above it.	A ceiling; an enforcement mechanism (the plan's budget only alerts); a rule for GPU and high-memory use.	Which ceiling; who informs the PI; how one GPU study is kept from consuming the program.
E. Campus recharge rate, if run as a specialized service facility under 2 CFR 200.468	The facility publishes a rate schedule; awards and other funds pay it equally; surplus or deficit is reconciled.	Recharge-center status or a decision that it is not needed; a rate method; biennial rate review.	Whether UCR treats a pass-through cloud purchase as a service center at all; how fixed costs enter the rate.

Four things to settle before the first study is onboarded

Our cost note is a usable first estimate for the CPU template; it needs correcting for the high-memory template and for snapshots and buckets, and it has no figure for anyone's time. Before the first study we need (1) a choice among the options above; (2) a named owner for shared costs; (3) a billing-export check that each cost line carries a study label; and (4) a rate or allocation statement that the campus research-administration office has read against 2 CFR 200.413 and 200.468. None of these is decided yet, and every figure here is an Iowa list price from 2026-11-01, not UCR's contract terms.

12.5 Design considerations

1. **Settle D8 against the COA/MOU rule before any study is onboarded**, and record whichever way it goes. The cost of leaving it open is that a researcher who asks what the service costs cannot be answered. If a subsidy is chosen, document the exception to the rule and name its funder.

2. **Correct the plan's cost paragraph.** Replace the high-memory figure (about $1,290) with a recomputed one (about $1,650 on the plan's basis, $1,706 with snapshots and buckets), say whether SUD is assumed, and add snapshots and buckets to each template. Label every figure with the price date.
3. **Name an owner and a budget for the unpriced shared costs**: security review, systems operation, steward effort, pentest, incident response and SCC if used. Estimate staff time from the pilot, not from another institution's ratio, and treat the AI-assisted research report's FTE and labor figures as hypotheses.
4. **Measure before quoting a rate.** The pilot should record VM hours by template, audit-log GiB per day by type, snapshot size and bucket growth. A rate quoted from the 40% assumption is a guess; the billing export gives the real figure within a month.
5. **Test per-study allocation in the pilot.** Check that VM hours, disks, snapshots, buckets and log ingestion each appear in the export with the `study` label, enable the export before the first VM exists, and decide how unlabeled costs are shared.
6. **Add a spending limit if a ceiling is promised.** The budget in the Terraform sends alerts. If a per-study ceiling is part of any option, say what the systems team does when a threshold is reached and who is told.
7. **Price GPU and high-memory studies separately.** One continuous GPU VM costs about $1,000 a month at list. A ceiling, a larger rate or an approval step for these templates avoids one study setting the program's bill.
8. **Settle the CIS fee** for the sizes in the plan, either from the Marketplace listing at deployment or from CIS. It is the only template price not published in a table, and the $0.0223 to $0.06 hourly range spans $16 to $44 a month on each of the two CIS-based templates running continuously, a difference of about $28.
9. **Ask the campus research-administration office two questions in writing.** Does the service need to be a recharge facility under 2 CFR 200.468, or may studies charge cloud use directly under 200.413? And does any part of the platform sit in the indirect cost pool, so that charging it again would double-charge?
10. **Ask Stanford how Nero's discount and waived transfer fees were negotiated.** If UCR's Google agreement already carries a similar discount, the figures here fall by that share; if it does not, the question of whether to seek one belongs to the campus Google account team, not to this plan.

Chapter 13

GPUs, HPC and AI on Sensitive Data

The plan gives a study one way to use a GPU: a single virtual machine with one NVIDIA L4 and 24 GB of GPU memory, started from the same locked-down pattern as the CPU templates. Nothing else about accelerated computing is built. There is no model-weight path, no position on managed AI services, no confidential-computing option and, apart from four lines in section 4.4, no cluster. Yet the questions that arrive first from researchers with P3-P4 data are about AI. They want to summarize clinical notes, code open-ended survey answers, run an imaging model, or ask a large language model (LLM) about records that may not leave the project. The campus has written guidance on which AI tools may touch which protection level, and the University of California legal office has written a stricter general rule [229, 230]. So the real question is what our one GPU template can do for that demand, and what else we'd need.

Three arrangements cover almost every request. The model can run inside the study's own virtual machine (VM), with weights carried in beforehand and no network path out. The model can be a managed service in the same Google Cloud project, reached over Google's private API path under the business associate agreement (BAA). Or the data can go to an outside service, which for P3-P4 data is the arrangement that policy at the University of California (UC) and at peers does not permit without a negotiated contract. I spend most of my time on the first two, because the third is closed for P3-P4 data. Facts about Google products were read on Google's documentation pages on 2026-11-01; claims about peer institutions come from each institution's own page.

13.1 The GPU template as planned

The module's `gpu` template is a `g2-standard-16` with 16 vCPU, 64 GB of memory, one L4, a 100 GB boot disk and a 1 TB data disk, started from the image family `p4-gpu-rocky9`. Google's machine-family page confirms the shape: the G2 series has the Ada Lovelace L4 with 24 GB of GDDR6 memory, a `g2-standard-16` carries one, and the series scales to 4 and 8 GPUs at `g2-standard-48` (96 GB total) and `g2-standard-96` (192 GB) [231]. The same page gives the L4 about 300 GB/s of memory bandwidth and lists bfloat16 and FP8 support. Those two numbers set what an LLM can do on it, and Section 13.2.1 turns them into arithmetic.

Four details of the Terraform and image script matter for sensitive work. First, the module sets `on_host_maintenance` to `TERMINATE` for any GPU template, because GPU instances cannot live-migrate; the offline plan for the pilot's `imaging-demo` VM shows that value together with Secure Boot, vTPM and integrity monitoring switched on (offline plan). A host maintenance event would therefore stop a running job. Second, the image script bakes the NVIDIA 580 driver from a Google-provided Rocky Linux 9 accelerator image family, then applies CIS Level 1 remediation with OpenSCAP. The script has never run, and I did not check that the named base family exists. Third, pilot test 2 is the only GPU test: the VM must boot with Secure Boot and the driver under the Shielded VM organization policy, and `nvidia-smi` and a PyTorch CUDA check must pass in the curated PyTorch container. Whether a signed driver module loads under Secure Boot in the image the script would produce is exactly what that test measures, and nothing offline can show it. Fourth, the module hard-codes a single accelerator of type `nvidia-l4`. Google documents G2 machine types as having GPUs attached by type, so whether an explicit accelerator block is accepted, and whether a

machine-type override to a four-GPU shape works, is another thing only a pilot apply can show.

The plan's idle-stop timer also matters more for GPUs than for other templates. The script counts a VM as busy if anyone is logged in, an RStudio or desktop session exists, any process runs under a regular user's ID, CPU load exceeds 10 percent of core count, or a keep-running file exists. GPU utilization is not checked. The user-process rule was meant to catch a researcher's detached training job, but the user-ID gap described in Chapter 9 means such a job probably does not count, and a GPU VM could stop in the middle of a run. An inference server started as a system service would not count either, and an idle-looking VM running a model for a study's collaborators would be powered off after the idle window. The pilot's GPU example sets the window to 240 minutes.

On capacity, the only facts I have are about the other project. The campus research cluster's own regional GPU quota includes L4 in volume, A100 40 GB in small numbers and no H100. Quota is granted per project, and the secure project would be a separate one (decision D6, Chapter 1). Nothing in the plan or the Terraform requests GPU quota for it, and Chapter 9 reached the same conclusion. Those quotas still say something useful about what the campus can obtain from Google: L4 is available in volume, A100 40 GB in small numbers, and the parts that matter for confidential computing (H100) not at all. Section 13.5 comes back to that.

Three things before the first GPU study

We have a GPU VM checked only on paper, with no GPU quota, no model-weight path and no stated position on managed AI. Before the first GPU study we'd need a quota request for the secure project; a decision on how model weights and curated containers reach an offline VM (Section 13.2.2); and a written answer from the campus security office on the managed-AI row of Table 13.3. The idle-stop rule also needs to count GPU work.

13.2 Open-weight language models inside the study VM

13.2.1 What fits on a 24 GB card

The question that decides whether a local model is practical is memory. A model's weights occupy the number of parameters times the bytes per parameter: 2 bytes at bf16, 1 at 8-bit, about 0.5 at 4-bit. On top of the weights, the serving engine needs a key-value (KV) cache that grows with the number of tokens in flight, plus working memory. I used the four Qwen3 models that Qwen publishes under the Apache-2.0 license and checked the parameter counts and license on the model repositories on 2026-11-01. Total parameters come from each repository's safetensors metadata (8.19, 14.77, 30.53 and 32.76 billion), and the license field reads `apache-2.0` for the four base models and for the FP8, AWQ and GPTQ-Int4 repositories that exist [232–235]. The layer and KV-head counts come from each repository's `config.json`.

Table 13.1 shows the arithmetic. I assumed that 90 percent of the 24 GB, or 21.6 GB, is available to weights and cache, which leaves room for CUDA context and activations; the vLLM engine exposes the share as a setting, but the 90 percent is my assumption, not a measured value. The 4-bit column adds 10 percent to cover quantization scales and unquantized embedding layers, also an assumption. The KV cache per token is $2 \times$ layers $\times$ KV heads $\times 128 \times 2$ bytes for a bf16 cache, using the head dimension of 128 in each configuration. For Qwen3-8B that is $2 \times 36 \times 8 \times 128 \times 2 = 147{,}456$ bytes, or 144 KiB per token. Figure 13.1 plots the weight columns against the card.

Table 13.1: Open-weight Qwen3 models against one 24 GB L4. Weights are decimal gigabytes (parameters times bytes per parameter; 4-bit adds 10% overhead). "Room" is the cache that remains after weights inside an assumed 21.6 GB budget, expressed as tokens at the model's bf16 KV size. Arithmetic, not measurement; license and parameter counts from the model repositories read on 2026-11-01.

Model (Apache-2.0)	Params (B)	bf16 (GB)	8-bit (GB)	4-bit (GB)	KV (KiB/ tok.)	Fits on one L4
Qwen3-8B	8.19	16.4	8.2	4.5	144	Yes at all three precisions. At bf16, 5.2 GB remain: about 35,000 tokens of cache.
Qwen3-14B	14.77	29.5	14.8	8.1	160	Not at bf16. At 8-bit, 6.8 GB remain (about 41,700 tokens); at 4-bit, 13.5 GB (about 82,000).
Qwen3-30B-A3B (MoE, 3.3B active)	30.53	61.1	30.5	16.8	96	Only at 4-bit, since every expert must be resident. 4.8 GB remain (about 48,900 tokens).
Qwen3-32B	32.76	65.5	32.8	18.0	256	Only at 4-bit, and tight: 3.6 GB remain, about 13,700 tokens. A 32,768-token context needs 8.6 GB of cache and does not fit.

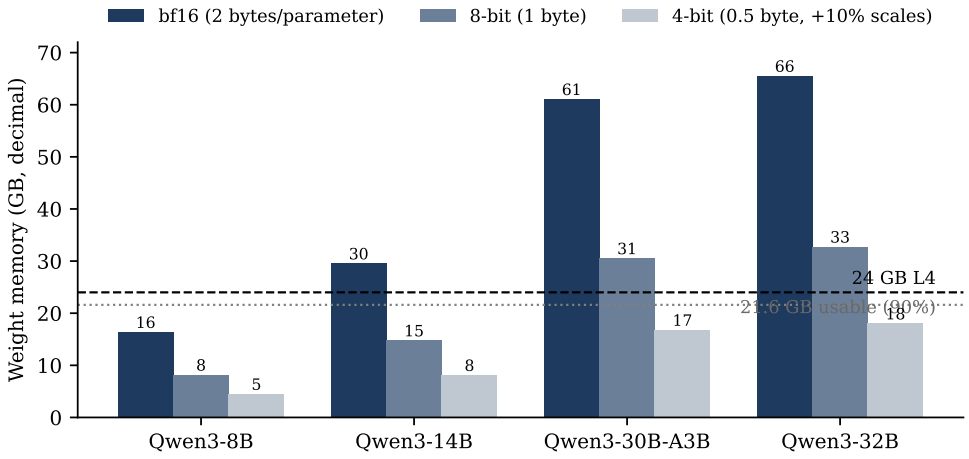


Figure 13.1: Weight memory of four Qwen3 models at three precisions against the 24 GB of an L4 (dashed) and an assumed 21.6 GB budget (dotted). Data from Table 13.1; script `figs/ch13_memory.py`.

An 8B model runs at full precision on an L4; a 32B model fits only at 4-bit and with a short context

Of the four Qwen3 sizes, only the 8B model fits at bf16. The 14B model needs 8-bit or 4-bit weights, the 30B-A3B mixture-of-experts model needs 4-bit even though only 3.3B parameters are active per token, and the 32B model at 4-bit leaves room for roughly 13,700 tokens of cache. Larger models call for the multi-GPU shapes (`g2-standard-48` has 96 GB across four L4s) or for the A100 class.

Two further checks follow from the specifications. The engine must support the quantization format on the card: vLLM's documentation lists AWQ, GPTQ, FP8 and other formats and shows AWQ, GPTQ and the Marlin kernels as compatible with the Ada generation to which the L4 belongs [236]. Qwen publishes an FP8 version of each of these models and a 4-bit version (AWQ for the 8B, 14B and 32B models, GPTQ-Int4 for 30B-A3B) under the same license, so no one at UCR would need to quantize anything inside the enclave. Second, token speed on a single stream is limited by memory bandwidth, since each generated token reads the weights once. At 300 GB/s [231], a ceiling follows from dividing by the weight bytes: about 18 tokens per second for the 8B model at bf16, about 20 for the 14B model at 8-bit, and about 17 for the 32B model at 4-bit. The mixture-of-experts model reads only its active experts, roughly 1.8 GB per token at 4-bit, so its ceiling is nearer 160. These are upper bounds that ignore the cache reads and routing overhead. Batching raises total throughput without raising a single user's speed. Quality loss from quantization is not computed here, and any claim that a 4-bit 32B model beats an 8B model at bf16 for a given clinical task needs a test on the study's own data. The pilot should measure tokens per second and task accuracy for one model.

13.2.2 Getting weights in without a network

An offline VM cannot download weights. The air-lock pattern is the one the plan already uses for software: build outside, review, and let the enclave pull only from curated stores (Chapter 8). Our plan says nothing about model weights, so the path below is my proposal. Figure 13.2 draws one version of the path.

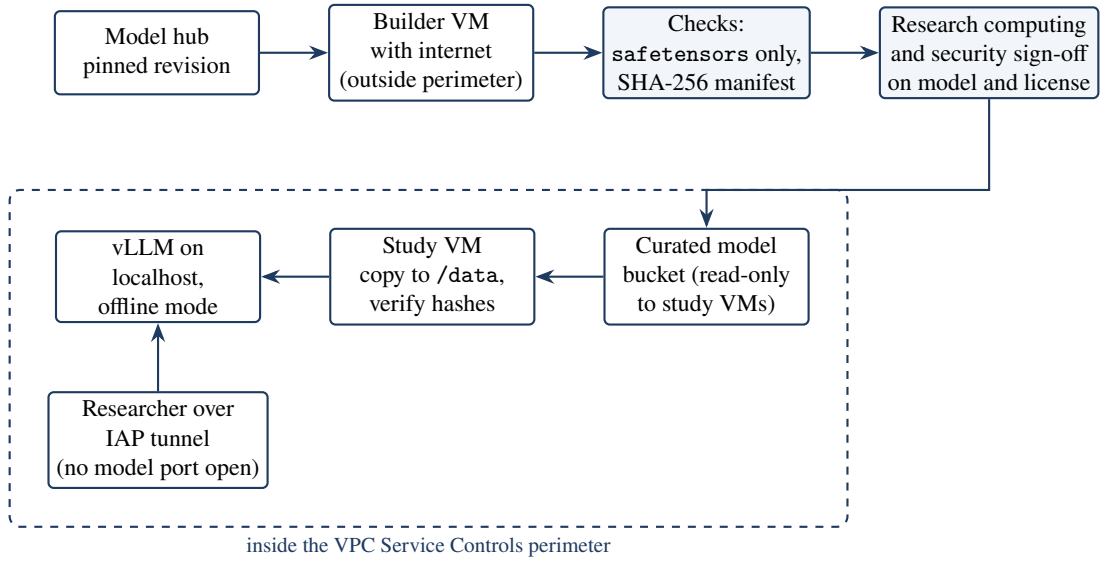


Figure 13.2: A proposed path for open-weight model files into an offline study VM. Nothing here is in the plan: it combines the plan's builder-outside pattern for images with the read-only bucket roles in the `secure-vm` module. Shaded boxes are review steps by staff. The VM reaches the bucket and nothing else; the researcher reaches the server only through the same tunnel used for RStudio.

The steps rest on documented behavior. The Hugging Face client has an offline setting: if `HF_HUB_OFFLINE`

is set, no HTTP calls go to the hub, only cached files are read, and a missing file raises an error [237]. The vLLM project collects usage statistics by default and documents opt-outs through `VLLM_NO_USAGE_STATS`, `DO_NOT_TRACK` or a file in the user's home directory [238]. In a VM with no route out the report would fail anyway, and the safer position is to set the opt-out in the image so that a future change to the network does not turn it on. On the file format, the safetensors project describes pickle-based checkpoints as unsafe because loading them can run arbitrary code, and describes its own format as a way to load files from an untrusted source without that risk [239]. A policy that admits only safetensors files closes the weight-file version of the supply-chain problem in Chapter 8, though it does not address the Python code that accompanies some model repositories, which the review step must refuse or read.

Where the files rest is a design choice with a cost. The study's `in` bucket already exists, and the VM's service account can read it but not write it. Placing a 16 GB to 66 GB model there for each study copies large objects to every project's buckets and puts staff-supplied files in a bucket that study members can also write to. A single curated bucket, readable by all study VMs and writable only by the platform administrators, is simpler and resembles the plan's curated container repository. It is a new shared object, so it brings the cross-study questions of Chapter 6 with it, in a mild form because the content is public model files with no study data. The review step also has to hold up against the limits of malware scanners reported in Chapter 8: a hash manifest proves that the file in the enclave equals the file reviewed, and nothing about what the file does.

13.2.3 Serving a model on a VM that holds PHI

Run inside the VM, the model sends no data to a third party, and none of the vendor-contract logic in the UC legal alert applies to it [230]. The alert is written about providing data to AI companies, and the campus security office may still treat the model as software that needs an inventory entry and a source. Several controls remain the study's and the platform's.

The server should listen on the loopback interface and be reached by the researcher through the tunnel's existing forwarded ports or an SSH forward, never on a port that other VMs in the subnet could reach. The vLLM security page recommends network isolation and describes which endpoints an API key protects and which it does not; it also documents prefix-cache timing as a side channel in multi-tenant deployments and a cache-salting option [240]. A VM that serves one study has no cross-study tenants, but a study with several collaborators of different clearance would have the multi-user case inside one VM. Request logging should be configured so prompts do not appear in log files that the Ops Agent ships to the log project, since the log bucket is locked for years (Chapter 10); I did not verify what vLLM logs by default and the build should check it.

The model itself is a new kind of artifact. Weights tuned on PHI are derived from it and could reproduce training records, so they would be P4 objects whose export goes through the steward path of Chapter 7. Fine-tuning needs more memory than inference (gradients and optimizer state), and I did not compute it. The systematization of knowledge on privacy-aware LLMs in healthcare organizes the threats by phase (preprocessing, fine-tuning and inference), which matches the split here [241]. An LLM given code-execution tools inside the VM acts with the researcher's rights, so it adds no path out of the enclave but can write and run code, and a coding assistant can suggest packages that do not exist, the slopsquatting risk of Chapter 8.

13.3 Managed AI under the BAA

The design assumes an executed Google Cloud BAA. Google's HIPAA page states that the BAA includes Google Cloud's entire infrastructure and the listed products [10]. Chapter 4 noted that the page no longer lists "Vertex AI" by that name. What counts is the list as the page shows it now.

In the Google Cloud supported-services list, the page names *Generative AI on Agent Platform*, *Gemini Enterprise Agent Platform*, *AI Platform Training and Prediction*, *Vertex AI Workbench instances*, *Agent Search on Gemini Enterprise Agent Platform* and, in a separate Workspace list dated 2026-10-28, *Gemini App* (excluding Gemini in Chrome). The Google Cloud list carries the date 2025-09-17, while the Workspace list on the same page is dated 2026-10-28; the naming on the page and on Google's current Vertex documentation is the newer Agent Platform one, so the date should be rechecked before anyone relies on the list. Model Garden does not appear by name. The page's best-practice table for the Agent Search entry says to use regional APIs and resource locations for PHI, and the page as a whole says to disable unsupported services when working with PHI, and not to use pre-GA offerings with PHI unless a notice says so [10]. So the answer to "is Vertex under the BAA?" is a qualified yes for the entries named, with three open points that the page cannot settle. Whether a Gemini call through `aiplatform.googleapis.com`, the service the plan's perimeter lists, is the entry named *Generative AI on Agent Platform* is a reasonable reading that Google's page does not state in those words. Whether a particular model version is generally available matters, because new model versions often start in preview and the page excludes pre-GA offerings. And the page says nothing about third-party models offered in Model Garden, some of which may be served by the model's developer under its own terms rather than by Google. I found nothing on Google's page that settles this, so I treat it as unverified and recommend that only Google-served models be considered until Google answers in writing.

The perimeter side is better documented. Google's page on VPC Service Controls for Vertex says that, with the service inside a perimeter, training data, created models, online inference requests, batch inference results and Gemini models cannot leave the perimeter, and that public access to the service is blocked for callers who are not allowed in; one listed limitation is that live fetching of URL context is disabled so that exfiltration is blocked [242]. I verified only the URL-context case, not other grounding features. The plan's foundation Terraform already puts `aiplatform.googleapis.com` in the perimeter's restricted services, sends `*.googleapis.com` to the restricted virtual IP through private DNS, and allows egress only to that address. It does not enable the Vertex API in the secure project: the enabled-services list has fifteen entries and none is `aiplatform`. As with all of the perimeter, offline planning skips the perimeter resource until an access policy identifier is set (Chapter 5). A Vertex call from a study VM would therefore need the API enabled, an IAM role for the study's service account, a regional endpoint and the dry-run results, and it would need a decision about whether that service account's role covers every study or only studies that ask.

The campus position is available too. UCR's campus IT guidance says that P4 data requires campus IT approval before any AI tool is used, and that the tools usable with consultation include Vertex AI and Google AI Studio when tied to a Google Cloud project; for P3 it lists Vertex AI as permissible [229]. A campus news item reports that the university's Google agreement prevents use of uploaded content for model training and that even P4 data may be processed with campus IT consultation [243]. Both describe campus tools used from a campus account, not a model called from a P4 study VM. The Gemini web application, even where Google's BAA list names it, is a destination outside the perimeter, and moving PHI from the enclave into it is an egress event that the steward path of Chapter 7 would have to approve.

13.4 What peers and the University say

Table 13.2 gathers the policies I could read on institution pages. They mostly agree on the general shape: general-purpose AI tools with a campus contract top out at P3 or Level 3, and the highest level goes to special platforms approved case by case. Michigan is the exception: its campus AI platform is approved for PHI.

Two things matter for us. First, Harvard Medical School puts its Level 4 path inside environments run for that purpose (the Longwood Cluster and HMS Azure AI, by special request). Michigan instead approves its campus AI platform, which fronts Azure OpenAI, for PHI, while Michigan Medicine excludes the

Table 13.2: Published AI-tool policies for sensitive data, from institution pages read on 2026-11-01. Harvard's and Michigan's levels are their own scales.

Institution	What the page states
University of California, Office of the General Counsel (March 2024)	Without properly negotiated agreements, units may not give generative AI companies personal information, including financial or medical information of employees, students and patients, even if de-identified. Contracts for tools that handle regulated data must include a data security appendix and, for PHI, a BAA appendix [230].
UC Riverside campus IT	P4 needs campus IT approval before any AI tool is used. Listed for P3: Agent Builder, Google AI Studio (with a Google Cloud project), Vertex AI, and others. Tools without a campus contract: P1 data only [229].
UCLA	Most campus tools are limited to P1-P3. FERPA and HIPAA data may not be entered unless approved in a secure environment [244].
UC Irvine	Campus ZotGPT, Copilot and Gemini: P1-P3. Non-campus ChatGPT, Copilot, Gemini and Claude: P1 only. P4 is not acceptable for any listed service by default [245].
Harvard (HUIT and HMS)	Listed tools are approved to Level 3 and not for Level 4 data "including HIPAA, FERPA, and PHI" [246]. At HMS, Level 4 tools are special requests: HMS Azure AI and the Longwood Cluster; no AI tool is approved for Level 5 [247].
University of Michigan	Central IT AI services, which include campus-hosted open-source models, include the safeguards required by HIPAA and may be used with PHI [248]. On the Michigan Medicine page, the Llama and Claude models offered through that service are not approved for ePHI [249]. Third-party AI without a university contract: low-sensitivity data only [250].

campus-hosted Llama models from ePHI use. Neither pattern is a model served from inside a secure research project, so that arrangement has no verified peer precedent. Second, the UC rules are written around contracts with AI vendors, and the BAA is what turns Google's AI services into candidates; the University's own text gives no rule for a model that the university runs itself, so that is a question for the campus security office.

Offer the local model first

Our decision D1 ("only BAA-covered services touch PHI") can be applied to AI only after someone maps the Google page's service names to the calls a researcher would make. An enterprise-grade managed model is plausible for P3 and, with security-office approval, for P4. A local model is available without any contract question but costs a GPU VM and staff effort to stage weights. I'd offer the local route first, because it needs nothing from Google beyond what we already use.

13.5 Confidential GPUs

Confidential computing encrypts memory in use and isolates it from the host. Google's documentation pairs it with GPUs on two machine families: A3 High with NVIDIA H100 under Intel TDX, and G4 with NVIDIA RTX PRO 6000 under AMD SEV [251, 252]. The G2 machine type with the L4 does not appear on either page, so the planned template cannot be made confidential. Google's configuration page lists the A3 High GPU zones as `europe-west4-c`, `us-central1-a` and `us-east5-a`. The machine-type page says that single-GPU A3 High shapes are available only as Spot or Flex-start VMs, and the Confidential VM page notes live migration is not supported and records a GPU attestation fault that could report a mismatched measurement, with Google and NVIDIA working on a fix [231, 252]. The campus research cluster's lack of H100 quota says nothing about the secure project but shows the campus does not hold that capacity today.

On cost in performance, an arXiv study of confidential LLM inference reports that on H100 confidential

Table 13.3: AI options for P3-P4 data and the conditions each would need. Status describes the evidence I found; none is built.

Option	Data path	Conditions	Status
Open-weight model in the study VM	Data and model stay on the study's disk; no third party	Weights admitted by the airlock of Figure 13.2; safetensors only; loopback serving; prompt logging off; outputs and tuned weights exported only by the steward; security-office inventory entry	Feasible on L4 for 8B at bf16 and 14B to 32B quantized (Table 13.1). Plan has no weight path.
Vertex AI, Google-served Gemini, in the study project under the BAA	Request goes over Google's private API path to Google's service in a US region; stays inside the perimeter by Google's description	BAA executed; API enabled and perimeter enforced after dry-run; regional endpoint; GA model; security office and counsel accept the Google page's naming; per-study IAM	BAA entry named but not mapped to this call. Perimeter and API not yet set.
Third-party models through Vertex Model Garden	Possibly served by the model's developer under its own terms	Written confirmation from Google that the BAA covers the specific model and tier	Unverified. Not recommended until confirmed.
Campus Gemini or NotebookLM from a campus account	Browser session outside the perimeter	Campus IT approval; the data would have to leave the enclave first	Not a study-VM option; egress approval needed.
External consumer and commercial APIs without a UC agreement	Data goes to the vendor	None satisfy UC policy for P3-P4	Prohibited by the UC legal alert and campus guidance [229, 230].
AI assistants on the open shared cluster	Can read job output and logs	Not available	The plan bars access from these tools; the data flow is not approved for P3-P4.

GPUs throughput fell by 4 to 8 percent, with smaller penalties at larger batch and input sizes, and that CPU enclaves (TDX and SGX) added under 10 percent throughput and under 20 percent latency overhead for Llama 2 models of 7B, 13B and 70B parameters [253]. I read the abstract only. A Trusted CI webinar abstract frames confidential high-performance computing as a research effort toward trusted-execution hardware built for HPC [254], which I read as a sign, not evidence, that confidential HPC is still a research topic rather than a production option for a P4 cluster.

The more basic question is whether confidential computing answers the plan's threat model. It protects data in use from the cloud operator and the hypervisor. The Google BAA already treats Google's infrastructure as covered, and the exposures the plan's reviewers and Chapter 6 identified sit with administrators who hold roles inside the project and with researchers, who log in to the guest. A confidential VM does not change who can log in to the guest. If the security office's risk assessment names the cloud operator as an adversary, the confidential option moves from academic to relevant; I found no such statement in the plan. For the first edition I would record the option, note that the planned L4 template cannot use it, and revisit it if a study's agreement or the security office requires it.

13.6 Phase 2: a secure Slurm cluster

13.6.1 What the plan says, and the demand it might meet

Section 4.4 of the plan allows a small Slurm cluster in the same secure project, only if a study needs one: its own small controller and login nodes, exclusive nodes, one shared Filestore of 1 TiB at the minimum (the plan's estimate is about $200 a month; Chapter 12 computes $197 at list) with a directory per study, separate from the campus research cluster's controller. The plan then states that study separation on that Filestore is by Unix permissions only and that the security office must accept that; an open item asks whether phase 2 is ever needed. No Terraform exists for it.

The demand that would justify a cluster is queueing for many users and jobs, multi-node work and job arrays. A single study that wants more GPUs does not need Slurm: the four-L4 and eight-L4 G2 shapes give 96 and 192 GB inside one VM [231], which is a template override. Peers show two models. Harvard Medical School offers a Level 4 cluster for AI workloads, by special request [247], and Chapter 2 records Slurm-based enclaves at other campuses. The research literature shows the other: a poster from a UK cloud trusted research environment proposes giving each researcher an on-demand Slurm environment, with, in the authors' words, full data and network isolation from other projects, with cost savings compared with an always-on physical environment [255]. The UK national AI resource project describes building a trusted research environment on a national GPU system and states the two-sided problem: such environments lack scale, and supercomputers lack the isolation to take sensitive data [256]. A third paper evaluates Kubernetes alongside batch scheduling on the UK Isambard-AI machine for a trusted research environment and for a vLLM-based multi-tenant model-hosting service, and states that Kubernetes does not replace batch scheduling [257].

13.6.2 The shared file system question

Chapter 6 identified that the plan's phase 2 carries forward the shared Unix-permission design that was judged unsuitable for P3-P4 on the campus research cluster. Google's documentation narrows the choices. The Filestore tier page lists a Basic HDD tier, which is a legacy tier, from 1 TiB, serving NFSv3 only, and zonal and regional tiers that serve NFSv3 and NFSv4.1 [258]. Google's access-control page says that with NFSv4.1, Kerberos can be used to secure access, and that otherwise access is controlled by Linux NFS options and IAM for instance operations [259]. Google's HIPAA page lists Filestore among covered products and

advises IP-based access control to limit which Compute Engine VMs and GKE clusters can reach an instance [10]. The plan's Basic HDD choice therefore cannot use Kerberos at all. Earlier planning assumed that NFSv4.1 with Kerberos is not available on Filestore; Google's current pages say that NFSv4.1 and Kerberos are available, at least on the zonal and regional tiers (the price of those tiers was not checked).

With NFSv3 the server trusts the user and group numbers that a client presents, a well-known property of the protocol. Instance-level IP controls decide which VMs mount the share; they cannot distinguish one directory from another. So a root user on any cluster node can read every study's directory, and exclusive nodes do not change that. Slurm accounts and partitions limit who can submit jobs, which is a scheduling control and not a data control. The foundation Terraform's restricted-services list has no Filestore API, so the cluster's storage would also sit outside the perimeter definition as written.

Table 13.4: Isolation options for phase 2 storage and where the plan stands. Costs use $200 per month per instance (Basic HDD 1 TiB; $197 at list, Chapter 12).

Option	What separates studies	Open questions
One shared Filestore, directory per study (the plan)	Unix permissions inside one export; clients mounted by IP	Security-office acceptance, as the plan states; root on any node sees all; no Kerberos on the tier named
One Filestore per study in the shared project	Separate instance and, by IP rules, separate client sets	About $200 per month each, so five studies is about $1,000; does not stop a shared administrator; tier choice decides NFSv4.1
Cluster per study in its own project	Project boundary, own perimeter, own keys	The recommended pattern for clusters; restructures the foundation (Chapter 6); fixed head-node cost per study
Ephemeral cluster per study with node-local disks and bucket staging	No shared file system; data in per-study buckets and on node disks	Fits the on-demand model in the cloud TRE poster [255]; slower to set up; needs a staging tool that is not in the plan

The administrator plane matters as much as the file system. A cluster adds a controller on which a Slurm administrator can start jobs as any user, and the plan's roles for systems-team administrators (Chapter 6) would need an extension to say who holds that power. Slurm accounting and the controller's logs would have to flow to the security-office-only log project. The plan's boot-time gate (Chapter 5) was written for single VMs: a compute node that joins the cluster would need the same gate or an equivalent, and the shared login node needs thought about which study's group it admits.

Phase 2 is a design question, not a defect

Phase 2 as written copies the shared-file-system design that is unsuitable for P3-P4 clusters, and chooses a Filestore tier that cannot use Kerberos. No cluster code exists yet, so this is something to settle before anyone writes it. The status today is no demand shown, no design agreed and no recorded security-office acceptance of Unix-permission separation.

13.7 Design considerations

1. **Request GPU quota for the secure project** (L4 in `us-central1`) and record the number. The campus research cluster's quota does not transfer, and the first GPU study should not discover this at apply time.
2. **Add a model-weights path to the plan.** Choose between per-study copies in the `in` bucket and one

curated, platform-written bucket; set a rule of safetensors files only, pinned revisions, a SHA-256 manifest and research computing and security-office sign-off for each model and license (Figure 13.2).

3. **Offer the local route first.** Publish a short list of admitted models with their measured fit: Qwen3-8B at bf16 and Qwen3-14B at 8-bit on one L4 are the comfortable ones from Table 13.1. Use the pilot to measure tokens per second and accuracy on a synthetic task before promising larger models.
4. **Test the GPU template beyond test 2.** Add checks that the explicit accelerator block is accepted on a G2 type, that the driver loads under Secure Boot from the built image, that the idle-stop timer counts OS Login users' jobs and does not power off a VM running a job or serving a model, and that a host maintenance event is survivable.
5. **Bake the offline settings into the image:** the offline hub flag, the vLLM statistics opt-out, loopback-only serving and no prompt text in shipped logs. Verify what vLLM logs by default.
6. **Ask the campus security office and counsel for a written position on managed AI.** The question to put to Google is which listed entry covers a Gemini call to `aiplatform.googleapis.com`, which model versions count as generally available, and whether any third-party model in Model Garden is covered. Until then keep third-party models out of scope.
7. **Decide the status of tuned weights and model outputs** as P4 artifacts that leave only through the steward path (Chapter 7), and give the steward a checklist for model files.
8. **Record that confidential GPUs are out of scope for the first edition** unless the security office's threat model names the cloud operator, and note that the L4 template cannot be made confidential.
9. **Hold phase 2 until a demand and a storage design exist.** If a cluster is built, compare the four options in Table 13.4, put Filestore in the perimeter definition, choose a tier that supports NFSv4.1 with Kerberos if a shared file system is kept, and define who holds Slurm administrator rights.
10. **Name a trigger for phase 2.** For example: more than a stated number of concurrent GPU users on one study, or a study that needs more than one node. Single-study GPU growth should use larger G2 shapes first.
11. **Correct the planning assumption** that Filestore cannot offer NFSv4.1 with Kerberos.

Chapter 14

The Plan Compared: Lessons, Risks and Roadmap

Version 3 of our plan is a design with validated Terraform that has never been applied, image scripts that have never run, and a set of runbooks and templates. Set beside what peer universities publish, what the rules require and what the literature says, its structure has precedent: the project boundary, the identity path, the audit sink and the data steward all appear in peer descriptions or vendor guidance. A few controls (the fail-closed login gate, locked audit rules, a written residual risk for copy-out) go further than any peer page I read describes, though the login gate needs its boot-time ordering fixed before it fails closed. The exposures are elsewhere. Some are things the plan doesn't address yet: student education records, trained models, providers' separation language and the cost of people. Others are defaults we inherited from vendors and open-source projects, which showed up only when I read the Terraform and scripts next to their documentation.

No number in the register is a measurement.

14.1 The plan against peers and the literature on twelve dimensions

Table 14.1 summarizes the comparison, with each row pointing to the chapter that holds the evidence. "Peers" means only the offerings whose own pages I read (Chapter 2). That set was seeded from institutions named in one AI-assisted report and checked on their own pages; it is not a systematic sample. Where peer pages say little, the table says so. "Literature includes standards and papers.

The structural choices (project boundary, no internet, campus identity, steward-mediated export) have precedent, and the weaknesses in them are specific rather than structural. The plan is weakest where it inherits defaults from other people's software and services: RStudio editions, package registries, soft delete, Google's pre-GA offerings. Each needs a fact from outside the repository, and neither review panel had one (Lesson 1 in the Executive Summary).

The structure has precedent; the exposures are omissions and unchecked defaults

On three of the twelve dimensions (hosting, identity, operating-model shape) the plan is consistent with what peer pages describe. On three more (network, data out, logging) it is consistent with vendor and published guidance, but no peer page says enough to compare against. On tenancy it departs from the one Google Cloud peer, defensibly but without a written rationale. It has gaps on five: data in, software admission, cost consistency, student records, and GPUs and AI. Most close with a decision, a document edit or a configuration line, and the register in Section 14.6 counts them.

14.2 What the two review panels changed

Before I started, the plan went through two single-shot panels of model reviewers (six lenses on version 1, ten on version 2), each run through a campus AI gateway. Each finding was checked against the files before it was accepted, and some were rejected. These were internal checks by AI reviewers, not independent audits;

Table 14.1: The plan compared with verified peer practice and the literature on twelve dimensions. The last column is my reading, not a rating by the campus security office.

Dimension	Plan (v3)	Peers (verified pages)	Literature and rules	Reading
Hosting and boundary	One dedicated secure project plus a log project run by the security office; no internet route; perimeter planned in dry-run first (Ch. 5)	Of eleven offerings that state hosting, six cloud and five on-premises; only Stanford Nero is documented on Google Cloud (Ch. 2)	Five Safes and SATRE treat the environment as one of several safeguards (Ch. 2)	Consistent with peer pages; little Google Cloud precedent
Network and API perimeter	Deleted default route, restricted VIP, private DNS, VPC-SC (Ch. 4)	Peer pages rarely describe network design	VPC-SC protects API calls, not network paths (Ch. 4)	Consistent with vendor guidance; no peer comparator; perimeter not yet in any plan output
Identity and access	Campus SSO with Duo, IAP, OS Login, per-study groups, affiliate accounts for outsiders (Ch. 5)	Peers that describe onboarding put a human review before data lands (Ch. 2)	Published friction concerns approval waits more than tooling (Ch. 9)	Consistent with peer pages; waiting time unmeasured
Tenancy	One shared project; per-study groups, accounts and buckets; default encryption (Ch. 6)	The one verified Google Cloud peer (Nero) gives each requesting team its own project; no peer page describes one shared project; provider agreements read name no cloud project	Counter-arguments favor project per study and customer keys; no source requires either; a folder perimeter (Option C) makes per-study projects cheaper than assumed (Ch. 6)	Departs from the one Google Cloud peer; defensible, cost-driven; reversal conditions undocu-mented
Data in	Upload to a per-study bucket; Globus not planned (Ch. 7)	Not described in peer pages read	Globus has a high-assurance mode and needs a BAA; fit with a no-internet network unverified (Ch. 7)	Gap: no ingress procedure
Data out	Data steward, checklist, release record, threshold default 11, written copy-out residual risk (D12) (Ch. 7)	Not described in detail	Output checking is a human decision; trained models need separate rules (Ch. 7)	Consistent with published guidance; no peer comparator; gaps on models, zeros, dominance
Software	Live mirrors for PyPI and Docker Hub, curated containers, R from EPEL (Ch. 8)	Ticket-based requests; right to reject software (Ch. 8)	Verified 2026 incidents argue for an admission rule; scanners are limited (Ch. 8)	Gap: no admission decision
Logging and response	Data Access logs, cross-project sink, nine alerts, locked audit rules (Ch. 10)	Not described in detail	Six years is the regulation's figure for documentation, not raw logs (Ch. 10)	Consistent with vendor guidance; no peer comparator; several coverage gaps
Operating model	RACI with one accountable party per row; staffing not stated (Ch. 11)	Operated by research computing with a school or central IT (Ch. 2)	No verified staffing ratio exists (Ch. 11)	Consistent with peer pages in shape; data owner and

Chapter 5 gives the version history. Table 14.2 lists the changes that most altered the design. The first panel scored every lens "not ready" (security 4, the others 3 or below). The second scored compliance 5, DSP process 4, images and scripts 2 and the rest 3. Most first-panel design issues were fixed by then, and the new code (image build, browser tools, alerts, log project) carried most of the new bugs.

Table 14.2: Changes driven by the two review panels, with the plan version in which each landed.

Finding	Change in the plan	Version
First boot could not install anything with the default route deleted	Images baked outside the perimeter with the logging agent, audit rules and repository settings; startup script reduced to mount and checks	v2
Startup script hid failures	Failure status and alert; then a fail-closed login gate that refuses non-admin logins	v2, v3
VM service account held broad rights on all buckets	Narrow roles per bucket; named data-steward group	v2
Plan overclaimed controls (locked log bucket, training, review)	Built-versus-process table; entries marked process where nothing is built	v2, v3
Administrators had project-wide service-account use	Project-wide service-account use dropped; deployer account applies study modules	v2
Log bucket in the same project as compute	Separate log project administered by the security office (D11)	v2
No CRAN, Bioconductor or conda path	Curated containers first (R, PyTorch, bioinformatics)	v2
No graphical option	RStudio Server and a desktop over IAP (D10); native RDP in place of Guacamole	v2, v3
Desktop could not install as written; shared password	Guacamole dropped for now; per-user GUI password helper	v3
Audit rules changeable by root	Immutable audit rule set	v3
No data disposal	Teardown script and runbook with evidence; noncurrent-version lifecycle; per-study snapshot retention	v3
Copy-out by authorized users	Residual risk stated (D12); SFTP off; GUI-only mode; steward checklist	v3
RACI ambiguous on risk acceptance	One accountable party per row; data-owner and security-officer columns; sign-off block	v3
Outside collaborators blocked by domain policy	Affiliate accounts under the same DSP (D13)	v3

The panels also showed their limits. Both verdicts record rejected claims: in the second, five of twelve Terraform-lens items were wrong about Google Cloud mechanics, and in the first about a third of the security lens's high-severity items were wrong. And neither verdict contains any of the defect classes that turned up later: perimeter service lists that must name every API a feature uses, the release status of mirror formats, edition limits of an open-source tool, storage soft-delete defaults, missing ingress rules. Each needs a document from outside the repository, which a single-shot reviewer working from the repository doesn't have. The comparison is confounded: the reviewers who found these defects (the claims panel, and my own drafting checks with AI agents) had the vendor and upstream pages, and no human review of the repository alone was run, so the result says what a reviewer without the documents misses, not how a human would do. A later claims panel on my draft found eight more v3 defects by reading the code against Google's pages and the upstream distribution sources (rows R06, R10, R12 and R31 to R35).

Table 14.3 sorts all thirty-five register rows by how they were found. Twenty-one needed an outside document, and none of those twenty-one appears as a finding in either repository-only verdict.

Table 14.3: How each register row was found, and whether finding it needed a document from outside the repository (a vendor, upstream or regulatory page, a price page or an agreement). No = nothing in the repository implies the defect; Yes = deciding it needs a fact stated only in an outside document. The classification is mine. The two repository-only panels' accepted findings were fixed before v3, so they have no open rows; R01 and R02 echo the first panel's process list, and a few rows (for example R12 and R35) touch points that a single raw panel review raised but its verdict did not carry forward.

Found by	Outside document needed	Register rows	Rows
Plan's open-items list (also in the v1 panel's process list), sharpened while checking	No	R01, R02	2
My reading of the plan and code while drafting	No	R03, R05, R09, R11, R15, R20, R23, R28, R30	9
My primary-source check while drafting, against a vendor, upstream or regulatory page	Yes	R04, R07, R08, R13, R14, R16, R17, R18, R19, R21, R22, R24, R25, R26, R27, R29	16
Claims panel on the draft, reading the code	No	R12, R31, R35	3
Claims panel on the draft, reading against Google's or upstream pages	Yes	R06, R10, R32, R33, R34	5
Repository-only AI panels on v1 and v2 (verdicts)	—	No open rows; accepted findings fixed before v3 (Table 14.2). Rejected: 5 of 12 Terraform-lens items (v2); about a third of security-lens high-severity items (v1)	0
Total	Yes 21, no 14		35

Supply primary documentation with the code

Two single-shot AI review panels that saw only the repository found none of the inherited-default defects; AI agents given the same code plus the vendor and upstream pages found them. Twenty-one of the thirty-five register rows needed a vendor, upstream or regulatory page, and none of them appears in either repository-only panel's verdict. For the next review I'd hand reviewers the vendor pages for every setting along with the code, and add one lens whose only job is to check each setting against its vendor's page.

14.3 A checklist for inherited defaults

The defects that mattered most in our plan came from defaults set by someone else. Table 14.4 turns them into a checklist another campus can run against its own enclave code. Each row is a kind of defect, not our configuration. The last column says whether reading the repository alone could have caught it. For nine of the ten the answer is no: nothing in the repository implies the defect, and deciding it needs a fact stated only in an outside document. The tenth, boot-time check ordering, was visible in the code, and both panels missed it.

Table 14.4: Inherited-default checklist: one row per kind of defect, the document that settles it, a generic check, and whether a review of the repository alone could find it. Repo-only?: No = nothing in the repository implies the defect; deciding it needs a fact stated only in an outside document. Type-level; not a description of our configuration.

Defect class	Document that settles it	Generic check to run	Repo-only?
Perimeter service names versus the APIs each access path calls	Vendor's supported-products page for the perimeter [76]	For each access path, list every API it calls and confirm each one is restricted and has an ingress rule	No
Restricted-endpoint routes and egress versus the vendor's range table	Vendor's private-access configuration page [75]	Diff routes, DNS and egress rules against every range the vendor lists for the restricted endpoint	No
Edition-gated application options	Product's configuration reference and source [127, 128]	Confirm that each lockdown option is honored by the edition actually deployed, not only documented for the paid one	No
Storage soft delete versus teardown and destruction evidence	Storage soft-delete page [89]	Set the soft-delete policy deliberately; make the final teardown check list soft-deleted objects, not only labeled resources	No
Identity-provider user-ID ranges versus inactivity detection	Identity provider's user-ID documentation [179]	Compare the ID range the idle check counts with the range the provider assigns; test with a real federated account and a detached job	No
Operating-system default log layout versus alert log sources	Distribution's default logging configuration [188]	For each alert, find the file or stream the OS writes the event to and confirm it is shipped and matched	No
Boot-time check ordering versus login-service start	The image's own service ordering	Confirm the gate is closed before any login service starts and that no status survives a reboot; force a failing step	Yes in principle (visible in the code; both panels missed it)
Live package mirrors versus an admission policy	Vulnerability records for the ecosystem [146]; registry remote-repository docs [80]	Write the admission rule (age quarantine, allowlist, minimum patched version) and check each mirror against it	No
Pre-GA products in the PHI path versus BAA guidance	Vendor's HIPAA page and the product's launch stage [73, 143]	List the launch stage of every product and feature in the data and patching paths; record the security office's ruling on each pre-GA one	No
Per-study keys versus shared service agents	Vendor's customer-managed key documentation [112]	Identify the principal that holds each key's encrypt and decrypt role; if one agent holds them all, keys do not separate studies	No

14.4 Corrections to sources and to the plan's documents

While drafting, I checked claims against primary sources and found errors in the AI-assisted research reports, in the plan's companion documents and in the plan text itself. Appendix A (Table A.1) lists the corrections to outside sources. Table 14.5 adds the ones about the plan's own documents and a few outside items the appendix doesn't carry. These are statements that are wrong or unsupported. Plan defects, which are design or configuration questions, are in the register (Section 14.6).

Table 14.5: Corrections found while checking: statements in reports, in the plan's companion documents and in the plan text. Report and gap-analysis items already in Table A.1 are grouped.

Statement and where	What the check found	Chapter
Gap analysis: "63 controls"	The tables hold 71 rows	1, 3
Plan: foundation plan has 101 resources	The offline plan has 100 (pilot 59)	5
Gap analysis recommends a cluster per study and customer keys, saying most UC campuses require CMEK for P4	Contradicts D3, D4 and D5; the CMEK claim has no source	6
Gap analysis: Duo and MFA to be determined	D2 records Duo as enforced; the evidence for IS-3 is what remains open	3
Link to the UC IS-3 PDF	The address returns 404; cite the policy index or policy page	3
Peer-enclaves report: peer facts	FASSE is Level 3 only; Nero is about $290 per month and about $20 per TB; SRDE is NYU's; Yale Hopper priority tier is future; Michigan free tier not on its page; TREEHOOSE abstract names no provider (Table A.1)	2
Multi-tenancy report: NIH accepts equivalents; Google's published foundation is the per-grant vehicle	ISO/IEC 27001 only for non-US users who cannot attest; that foundation built the sibling CUI enclave	1, 3
Gap analysis: HIPAA Security Rule rewrite "slipped or dropped"	Still a proposal; the Unified Agenda lists final action 07/2027	3
Gap analysis: CMS bars co-hosting; 90-day destruction	Terms restrict linkage and require destruction; no absolute ban found; 90 days applies to shipped media	6
Multi-tenancy report: VPC-SC isolates studies from each other	A perimeter around one project does not separate studies inside it	6
Plan default cell threshold 11 versus the ONS rule of 10	No error: 11 is the plan's own choice; the text should say so	7
Researcher-experience report: one third waited over 12 months	Not found; ADR UK reports over a third waited over 6 months	9
Data ingress/egress report arXiv citation; logging report "six years of locked raw logs"	The identifier is an unrelated paper; 164.316 covers documentation retention	7, 10
Cost and funding report claims	CIS fee is per instance-hour; no ±15% corridor in 200.468; equipment threshold is now the lesser of the capitalization level and $10,000; the 26% administrative cap is in Appendix III to Part 200 (section C.8.a), not statute	12
Gap analysis: Filestore cannot do Kerberos with NFSv4.1	Zonal and regional tiers can; Basic HDD is NFSv3 only	13
GPU and AI report on managed AI coverage	The BAA list names Filestore, Batch, Cluster Director and Managed Lustre; Model Garden is not named	13
Plan: only the security office has roles in the log project	True of the audit log bucket, not of the whole project; see the deployer problem in Section 5.3 (R12)	5, 10
Plan: the idle check "catches detached jobs"	As written it misses the identity provider's users (R32)	9

14.5 Candidate changes for version 4

The plan comparison I prepared before writing proposed ten changes, S1 to S10. I added more as I read the Terraform and scripts while drafting. Table 14.6 gives each a priority: *P* before the fake-data pilot, *D* before the first P3 or P4 data enters, *L* later. They are suggestions for the research computing director and our partners.

Table 14.6: Candidate changes for a version 4. P = before the pilot; D = before first P3–P4 data; L = later. S-numbers are from the plan comparison; N-numbers are mine.

Item	Change	Prio.	Chapter
S1	Add FERPA to the plan: exception, agreement destruction date and named users at intake, a breach-matrix row, a test that a study fits the studies exception	D	3, 11
S2	Software admission: pin R 4.4.0 or later with a failing build (P); quarantine window on mirrored PyPI, CRAN and conda, and an allowlist for GUI-only studies (D)	P, D	8
S3	Export rule for trained models: default no export without review	D	7
S4	Keepalives in the researcher connect command; bulk data through buckets, not tunnels	P	9
S5	Intake question on provider separation language; route dbGaP and CUI to the sibling enclave	D	6
S6	Align D8 with the COA/MOU rule, or record the decision to depart from it	D	12
S7	Tabletop exercise as a gate before PHI	D	10
S8	Evaluate Globus as an optional ingress path	L	7
S9	Ask the sibling enclave team whether early stages of its published foundation can supply this one (bootstrap, state, sinks, folders)	P	4
S10	Keep the shared project and default encryption, or adopt the folder layout (Option C) if leadership chooses it; write the counter-argument and reversal conditions either way	D	6
N1	Dry-run week confirms the perimeter's remaining open points (storage ingress for members and data stewards, sink path, any other violations)	P	4, 7
N2	Decide the RStudio edition; until then treat GUI-only mode as unproven and run pilot test 10 for RStudio and the remote desktop	P	7
N3	Image build: integrity checks for installers, a registry read binding for VMs, the missing R container recipe, digest pinning	P	8
N4	Add cross-study negative tests to the pilot; target firewall rules by service account	P	6
N5	Add researcher metrics (time to first login, package and export turnaround) to the pilot	P	9
N6	Teardown: soft-delete policy, unlabeled-resource check, clear ingest storage after use, lock the evidence	D	7, 11
N7	Gate re-check during sessions; alert on Shielded VM or Secure Boot change	D	5, 10
N8	Log fixes: sink path versus perimeter, deletion protection on the log project, a six-year retention that counts leap days, group-change alerts	D	10
N9	Export checklist: zero handling, 99.31(b) test, dominance item, release log	D	7
N10	Correct the cost text; replace the $2,000 placeholder budget with a decided amount; decide how a per-study ceiling is enforced	D	12
N11	Request GPU quota for the secure project; add a model-weights path; enable and bound the managed-AI service before any managed-AI use	D	13
N12	Hold phase 2 until demand and a storage design exist	L	13
N13	Make the gate's status boot-scoped, so a failed or hung step leaves the login gate closed and no earlier pass carries over	P	5
N14	Separate duties: limit who may act as the deployer to a break-glass group with an alert on each use; keep the deployer away from the evidence store	D	5, 6, 10
N15	Widen the idle check to the identity provider's users; run pilot test 12 with a real federated account and a detached job	P	9, 12, 13
N16	Check every upstream mirror address against the upstream source before the first image build, so required packages install	P	4, 8
N17	Cover every address range Google lists for the restricted endpoint in routes and egress rules	P	4
N18	Name in the perimeter every API each feature uses, and write the ingress rules for them	P	4, 5
N19	Apply the campus access level to every perimeter ingress rule once it exists	D	4
N20	Point each audit alert at the log the operating system actually writes, or at the audit records	P	10

Thirteen of the thirty items are priority P (S4, S9, N1 to N5, N13, N15 to N18 and N20), S2 is split between P and D, fourteen are D and two are L. N13 to N20 are the eight v3 defects that the claims panel found. Six of the eight are P, because otherwise the pilot would test the wrong script or image. Most P items

are test or configuration work that fits in the plan's three-to-four-week build estimate. The D items are mostly documents and decisions.

14.6 The risk register

Table 14.7 lists thirty-five risks. Each is a defect or open question in the plan or its Terraform, written as a design question and stated as a kind of weakness rather than a configuration line. Likelihood (L) and impact (I) are my judgments on a three-point scale (H, M, L), with no measurement behind them; they exist to order the work. Owners are generic roles. The type column says what closes each risk: *D* a decision or document edit, *C* a code or configuration change, *T* a pilot test that may or may not find a problem. Sources for each row are in the chapter listed.

Table 14.7: Risk register for the planned environment. L = likelihood, I = impact (H, M, L). Type: D = decision or document, C = code or configuration, T = pilot test.

ID	Risk	L	I	Owner	Mitigation	Type	Ch.
R01	No risk treatment plan yet; the security officer has not accepted the shared boundary (D3), default encryption (D4) or the copy-out residual risk (D12)	M	H	Security officer, security office	One Risk Treatment Plan with a decision record and reversal conditions; written reading of the UC key-management standard	D	6, 11
R02	Accountable data owner unnamed; the systems team has not accepted its operating rows; the RACI lacks lifecycle rows (steward appointment, group ownership, separation feed, restore test, teardown hold); one reminder exists only on paper	H	H	Research computing director, systems team	Name the accountable data owner; add the rows; record acceptance in writing; implement or drop the reminder	D	11
R03	D8 (possibly subsidized) conflicts with the COA/MOU rule; no owner of shared costs	H	M	Research computing director, leadership	Decide and record; name the funder of staff, pentest and incident response	D	12
R04	Cost text is stale: high-memory estimate about a fifth low, GPU about 4% off, platform cost omitted; the budget amount is a $2,000 placeholder; budget alerts cannot enforce a per-study ceiling	H	M	Research computing team	Recompute from price pages with dates; replace the placeholder; choose an enforcement route	D	12
R05	Neither offline plan contains the service perimeter or the domain restriction (deferred until identifiers exist)	H	H	Systems team, security office	Obtain the identifiers; plan and run dry-run in the pilot	C	5
R06	A product protected by several APIs needs all of them in the perimeter; one access path is not yet covered	H	M	Systems team	For each access path, list the APIs it calls from Google's supported-products page, restrict them and write ingress rules for them; confirm in dry-run	C	4
R07	No ingress rule admits storage API calls from study members and data stewards; enforcement may block upload and export	H	H	Systems team	Test laptop upload and steward download in dry-run; add rule or confirm none needed	T	7
R08	Log sink writes to a project outside the perimeter with no egress rule	M	M	Systems team	Test in dry-run; add the rule or move the sink path	C	10

ID	Risk	L	I	Owner	Mitigation	Type	Ch.
R09	Evidence store not yet tamper-resistant: log storage unlocked, no deletion protection on the log project, retention computed short of six years, teardown evidence in unlocked storage	M	H	Security office	Add deletion protection; correct the retention; lock after pilot; lock or copy evidence	C	10
R10	One alert watches a different log from the one the operating system writes (records still reach the audit log); IAM-change matching too narrow; duplicate ingestion; group changes need the directory log; alert recipients unsettled	H	M	Security office, systems team	Base each alert on a log that is shipped; widen matching; remove the duplicate; add a directory-log alert; settle recipients	C	10
R11	No alert on a change to instance integrity settings while administrators hold instance-level rights	M	M	Security office	Add alert on integrity-setting changes; review at the weekly check	C	10
R12	Administrators can reach study identities and the evidence store indirectly through the deployment identity; our separation statements hold only by process (the deployer problem, Section 5.3)	H	H	Systems team, security office	Limit deployer use to break-glass with an alert; keep evidence in a locked store the deployer cannot write; consider deny policies	C	5, 6, 10
R13	Cross-study separation untested; firewall targeting relies on a label that instance editors can change	M	H	Research computing team	Add negative tests to the pilot; target firewall rules by service account	T	6
R14	Phase 2 copies the shared Unix-permission file system the gap analysis rejected, on a tier without Kerberos	L	H	Research computing team, security office	Hold phase 2; evaluate one instance per study or a cluster project; choose an NFSv4.1 tier	D	13
R15	Fail-closed gate evaluated at boot and login only; no periodic re-check	M	M	Research computing team	Add timed re-check and status-change alert; decide whether to end sessions	C	5
R16	Researcher path: no client keepalives, the provider's inactivity drop untested, no message on gate refusal, no way for the study to restart a stopped VM, no pilot metric for researcher time or satisfaction	H	M	Research computing team	Add keepalives and a refusal message; give the study a start path; add metrics to the pilot	C	9
R17	The open-source edition of the browser IDE ignores download and shell restrictions, so its GUI-only lockdown likely does nothing	H	H	Research computing team, leadership	Decide the edition; until then do not offer it for studies that forbid copy-out	D	7
R18	Some finer remote-desktop redirection options are left at their defaults (the main clipboard and drive channels are refused); copy-out audit coverage is partial	M	H	Research computing team	Set the options; widen audit coverage; verify in pilot test 9	T	7

ID	Risk	L	I	Owner	Mitigation	Type	Ch.
R19	Export form treats “under 11” as suppressing zeros (ONS yes, CMS no), has no 99.31(b) test, dominance item or release log	M	M	Data steward, security office	Copy each agreement’s cell rule into the record; add items and a log	D	7
R20	No rule for releasing trained models, though a GPU study is in the pilot	M	H	Security office, data steward	Default no export of models trained on P3–P4 data without review	D	7
R21	Teardown completeness: the final check can miss soft-deleted copies and unlabeled resources; ingest storage is never cleared and stays readable by members; delete errors are suppressed	H	H	Systems team, security office	Set soft delete; extend the final check to unlabeled resources; clear ingest storage; fail on delete errors	C	7, 11
R22	Live mirrors of public package registries give no admission decision; the language runtime version is not pinned	H	H	Security office, research computing team	Quarantine window, allowlist for GUI-only studies, failing build below a minimum patched version	C	8
R23	Image build has integrity and signature verification gaps, lacks one container recipe and a registry read binding for VMs	M	M	Research computing team	Add checks, recipe and binding; pin digests; separate build and promote identities	C	8
R24	Some mirror repository formats are pre-GA or unsupported; Google advises against pre-GA services with PHI	M	M	Security office, research computing team	No PHI would pass through a package cache, so the rule may not apply; the security office records whether it accepts a pre-GA dependency in the patching path; if not, batch-import snapshots into standard repositories	D	8
R25	D1 reads “only BAA-covered services”, but several control-plane services are not named on the list	M	M	Security office, counsel	Restate D1 as data-bearing services; ask counsel about control-plane items	D	4
R26	FERPA absent from the plan: no exception field, no destruction date, no breach-matrix row	H	H	Security office, FERPA officer	Add five intake questions and a matrix row (Table 11.4)	D	3, 11
R27	Provider agreements may require separation the shared project cannot give; intake does not ask	M	H	Security office, research computing team	Intake question; route to the security office before a VM is built; read first providers’ text	D	6
R28	An unused foundation resource duplicates one the module creates and inflates the resource count	H	L	Research computing team	Remove or use it; regenerate and archive plan output	C	5
R29	GPU: no quota requested for the secure project; the managed-AI service is restricted by the perimeter but not planned; L4 cannot be confidential; idle stop ignores GPU use	M	M	Research computing team, systems team	Request quota; decide managed-AI use; add GPU check to idle stop	D	13
R30	No tabletop exercise before first PHI; notice clocks per agreement unverified	M	H	Security office, privacy office	Run the exercises in Chapter 10 before first data	D	10

ID	Risk	L	I	Owner	Mitigation	Type	Ch.
R31	The gate is intended to close early in boot, but the ordering leaves a window: a failed or hung step, or a login service that starts before the script, can find the previous boot's pass status	M	H	Research computing team	Make the status boot-scoped (empty at every boot, or tied to the boot ID); test a forced failure in pilot test 6	C	5
R32	The idle check misses the identity provider's users; as written, tunnels and detached jobs probably do not keep a VM awake, and VMs may stop under running jobs	H	M	Research computing team	Widen the check; test with a real federated account and a detached job	C	9, 12, 13
R33	An upstream mirror address that does not resolve would leave images without required packages	H	M	Research computing team	Correct it; check each remote before the first image build	C	4, 8
R34	Routes and egress rules for Google's restricted endpoint must cover every range Google publishes for it; ours have not been checked against the full list	M	M	Systems team	Check against the table and add any missing route and egress rule; watch the dry-run for direct-connectivity failures	C	4
R35	Every ingress rule should carry the campus access level once it exists; as written, not every rule would pick it up	H	M	Systems team, security office	Reference the access-level setting in every ingress rule	C	4

Counting the rows by type gives 14 decision or document risks (D), 18 code or configuration risks (C) and three pilot tests (T). So fourteen of the thirty-five can close without any engineering: they need a decision, a signature or a document edit. Eight rows are rated high on both likelihood and impact. Three are decisions (R02, R17, R26), four are configuration (R05, R12, R21, R22) and one is a pilot test (R07). Rows R31 to R35, with R06, R10 and R12, carry the eight v3 defects that the claims panel found. R06 moved from a pilot test to a configuration change because Google's documentation settles it. The risks most likely to stall the first study are owners and decisions (R01 to R04, R17, R26), not Terraform.

Three kinds of unfinished work

The register doesn't say the plan is unsafe. It says three kinds of work are unfinished. Some decisions only the security officer, the research computing director and leadership can make. Some configuration is unproven until a dry-run week and the fifteen-test pilot have run. And a short list of facts about other people's software has to be checked against each vendor's page. The plan already lists most of the first kind as open items (security-office acceptance, the accountable data owner, funding, systems-team commitment). It doesn't yet list the second and third; Section 14.7 places them among the pilot tests.

14.7 Roadmap and decision gates

The plan estimates three to four weeks for build steps 1 to 6 and puts the security review, enforcement, log lock, penetration test and first study together in step 7. Figure 14.1 spreads step 7 across six gates, each depending on the one before. I don't propose calendar dates; they depend on the capacity of the security office and the systems team, which I don't know. The order has reasons. The Risk Treatment Plan needs the pilot results, because three of its claims (shared boundary, fail-closed gate, copy-out mode) rest on tests. Enforcing

the perimeter follows a clean dry-run week. The log bucket lock is irreversible for its retention period, so it waits for the pilot's teardown test and the retention decision. The penetration test follows enforcement, so it tests the system as it would run.

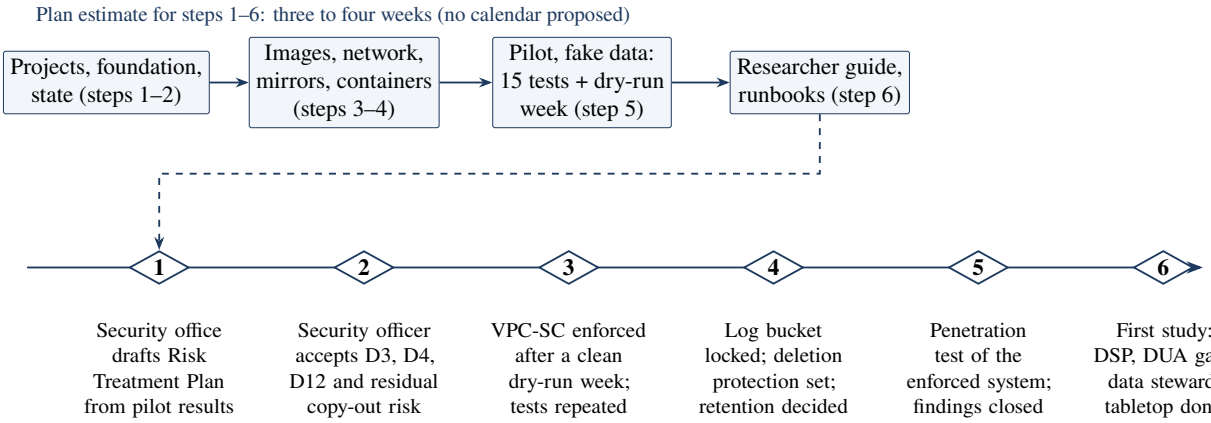


Figure 14.1: Proposed roadmap with six decision gates. Phases follow section 7 of the plan; the gates split its final step. Gate order is a proposal: each gate depends on the results of the one before.

Every gate needs evidence, and the fifteen pilot tests (Section 5.10) supply most of it. Tests 1, 3, 4 and 14 inform the perimeter gate. Tests 5, 7 and 15 inform the lock and the retention decision. Tests 6 and 8 to 11 inform the Risk Treatment Plan's claims about the gate, the audit rules and copy-out. Test 2 shows whether the GPU image boots at all, and tests 12 and 13 inform operating cost and restore. Four checks are missing from the fifteen and should be added: cross-study negative tests (R13), the storage ingress check (R07), a measured researcher path (R16) and an RStudio edition check (R17). I'd record a gate as passed only with a named approver and the test results attached.

Most of the first study's critical path is decisions

The fifteen tests plus four additions would confirm or refute the register's eighteen configuration risks and three test risks. They can't close the fourteen decision risks. If the decisions in Section 14.8 wait until after the pilot, gate 2 becomes the longest wait, because it needs the accountable data owner, the funding answer and the RStudio edition at once.

14.8 Decisions for the research computing director and leadership

Six decisions belong to the research computing director, a campus advisory body or campus leadership, and no test can settle them.

Funding: D8 or the COA/MOU rule. The plan's D8 says the service is "possibly subsidized" by the campus no-charge tier. The rule that P3–P4 projects pay their own costs through a chart of accounts, an MOU and a separate billing account says the opposite (Chapter 12). The two cannot both be policy. The GCP invoice is small; the open cost is staff time for security review, systems operation, data stewards, penetration testing and incident response. A decision names who pays each of those.

Shared project, folder of study projects, or per-study triggers. D3 chooses one shared project, and that remains the plan's position. Chapter 6 lists conditions that would force a study into its own project (a provider

that requires it, a security-office finding, a study-count review point). It also sets out a real alternative, Option C: one folder for the environment holding a small platform project and one project per study, with organization policies, the log sink and one VPC Service Controls perimeter set on the folder. Google documents that projects created in or moved into a perimeter folder join the perimeter automatically [102], so adding a study needs no perimeter or policy edit (at roughly twice the shared design's per-study resources, an estimate), and a study gets a project IAM boundary without a copy of the foundation. Leadership would choose among three paths: keep D3 with written triggers and a study count at which it is reviewed; adopt Option C before the foundation is applied, while the change is cheap; or keep D3 for the pilot and decide on Option C at the review point.

Mirror policy. Live mirrors, mirrors with a quarantine window, or a curated allowlist (Chapter 8) trade researcher waiting against admission control. The choice sets staff load: the more admission control, the more recurring requests for the research computing team to handle.

RStudio edition. Open-source RStudio Server ignores the file-download and shell restrictions that the plan's GUI-only mode relies on; the options exist in Posit Workbench (Chapter 7). Three paths are open: license the edition that has the controls (I didn't research the cost), accept that RStudio is a shell-mode option only, or offer the desktop only for studies that forbid copy-out. Each changes which studies the service can take.

Reusing the sibling enclave's foundation. UCR's separate CUI enclave was built on Google's published infrastructure-as-code foundation. Whether its early stages could supply the bootstrap, state and log sinks for this environment is a question for the team that runs that enclave (S9). The sibling remains a routing target for 800-171 data; the question is only whether the foundation can reuse its pieces, and it has to be answered before the foundation is hand-written and applied.

Globus. Globus supports high-assurance collections for PHI and needs a BAA; whether UCR has a subscription and whether an endpoint can run in a project with no internet route are unverified (Chapter 7). Leadership decides whether to spend the effort to find out or keep bucket upload as the only path.

14.9 Open questions

The plan's own twelve open items stand: SIEM choice and scanner reach, security-office acceptance, systems-team commitment, the accountable data owner, funding, project and billing identifiers, perimeter and domain identifiers, the GUI password approach, container turnaround, affiliate accounts, the first pilot study, and whether phase 2 is ever needed. I'd add these, none of which I can answer:

- Does the security officer, through the security office's risk treatment plan, accept Google default encryption for P4 on the shared project (D4), and does any named provider require customer-managed keys? (Chapter 6)
- Which Google service entry covers each control-plane service the plan uses, and does counsel treat them as in scope of the BAA? (Chapter 4)
- Can an authenticated scanner reach private VMs, and which SIEM will hold the sink? (Chapters 10, 11)
- How long does DSP approval take at UCR? The published evidence is about approval waits, not tool friction. (Chapter 9)

- Does the campus FERPA officer read administrator access to education records as consistent with the exception a given study uses? (Chapters 3, 11)
- Does a locked log count against a destruction clause in a provider's agreement? (Chapter 11)
- Can affiliate accounts enroll in Duo and use the same sign-in path? (Chapter 9)

14.10 Design considerations

1. **Adopt the register as the plan's open-items list.** Replace section 13 of the plan with the thirty-five rows, give each an owner and a date, and report status by risk, not by resource count.
2. **Close the decision risks first.** Fourteen of the thirty-five rows need no engineering. Get the security officer, the research computing director and the systems team to decide R01 to R04, R17 and R26 before the pilot starts. Fix the v3 defects that would make the pilot test the wrong thing (R31 to R33 and R10's alert source) before the first image is built.
3. **Add the four missing tests to the pilot list** (cross-study negatives, storage ingress, researcher time, RStudio edition) and record a result and an approver for each of the fifteen existing ones.
4. **Treat the dry-run week as a design review of the perimeter.** Fix the API list, the address-range coverage and the access level on every ingress rule first (R06, R34, R35), then test storage ingress and the sink path.
5. **Run the inherited-default checklist** (Table 14.4) against v4 before the first apply, and give any future review panel the vendor pages along with the code.
6. **Keep corrections separate from defects.** Fix the 71-row and 100-resource statements in the plan documents, and cite the policy page instead of the dead IS-3 PDF link.
7. **Write down the gate order and the reasons for it,** including that the log lock waits for the retention decision, and have the security officer sign each gate.
8. **Record the departures from peer practice** (shared tenancy rather than Option C or a project per study, live mirrors, no Globus, default encryption) as decisions with reversal conditions, so a later reader can tell a choice from an omission.
9. **Revisit the register after the pilot.** Replace the likelihood and impact judgments with test results and measured times, and retire the rows the tests close.

14.11 Conclusion

If I could get one thing decided first, it would be the accountable data owner. Five rows of our responsibility matrix point at that person, and gate 2 can't pass without one. Funding and the RStudio edition come right behind. They get easier once someone owns the data side.

Before anyone builds an image, I'd fix the defects that would make the pilot test the wrong system. The login gate has to start every boot closed, with nothing carried over from the last boot. The idle check has to count the user IDs our identity provider actually hands out; otherwise it stops VMs under running jobs and makes our uptime look better than it is. And each alert has to read the log the operating system really writes. None of these is hard to fix. Two of the three, the idle check's user-ID range and the alert's log source, showed up only beside the identity provider's and the distribution's pages; the third, the gate's boot ordering,

was visible in the code and both panels missed it. Reading code next to someone else's documentation is the lesson I'd pass on to anyone building one of these.

I found no gap that requires abandoning the VM-per-study, no-internet pattern. Tenancy is still open: one shared project, or a folder of study projects. I'd like leadership to choose before the foundation is applied, while changing course is cheap.

What I'm least sure of is staffing. I couldn't find a verified ratio of staff to studies anywhere, and the cloud bill is too small to tell us what the service really costs; the people are the cost. So in the pilot I want everyone involved to log their hours on security review, image rebuilds, data-steward checks and tickets. A few months of that log will be a better answer than any number I could have put here.

References

[1] Google Cloud. Google Cloud Observability pricing. https://cloud.google.com/products/observability/pricing, 2026. List prices read 2026-11-01.

[2] Google Cloud. Cloud Storage pricing. https://cloud.google.com/storage/pricing, 2026. Accessed 2026-11-01.

[3] Google Cloud. Security Command Center pricing. https://cloud.google.com/security-command-center/pricing, 2026. List prices read 2026-11-01.

[4] University of California Office of the President. Classification of Information and IT Resources. https://security.ucop.edu/policies/institutional-information-and-it-resource-classification.html, 2026. Accessed 2026-11-01.

[5] University of California Office of the President. Systemwide IT Policies (includes BFB-IS-3, Electronic Information Security). https://security.ucop.edu/policies/it-policies.html, 2026. Accessed 2026-11-01.

[6] University of California Office of the President. UC Protection Level Classification Guide. https://security.ucop.edu/files/documents/uc-protection-level-classification-guide.pdf, 2026. Accessed 2026-11-01.

[7] University of California Office of the President. Health Insurance Portability and Accountability Act of 1996 (HIPAA), Ethics, Compliance and Audit Services. ucop.edu, 2026. URL https://www.ucop.edu/ethics-compliance-audit-services/compliance/hipaa/. Accessed 2026-11-01.

[8] Department of Health and Human Services. HIPAA security rule to strengthen the cybersecurity of electronic protected health information (proposed rule, 90 FR 898). https://www.federalregister.gov/documents/2025/01/06/2024-30983/hipaa-security-rule-to-strengthen-the-cybersecurity-of-electronic-protected-health-information, 2026. Accessed 2026-11-01.

[9] Office of Information and Regulatory Affairs. Unified Agenda entry, RIN 0945-AA22: HIPAA Security Rule to Strengthen the Cybersecurity of Electronic Protected Health Information. reginfo.gov, 2026. URL https://www.reginfo.gov/public/do/eAgendaViewRule?pubId=202510&RIN=0945-AA22. Stage: Long-Term Actions; final action 07/00/2027. Accessed 2026-11-01.

[10] Google Cloud. HIPAA compliance on Google Cloud. https://cloud.google.com/security/compliance/hipaa, 2026. Accessed 2026-11-01.

[11] Google Cloud. How UC Riverside is securing the path to federal grants with Google Public Sector. https://cloud.google.com/blog/topics/public-sector/a-new-standard-for-research-how-uc-riverside-is-securing-the-path-to-federal-grants-with-google-public-sector, 2026. Accessed 2026-11-01.

[12] National Institutes of Health. NOT-OD-24-157: Implementation update for data management and access practices under the genomic data sharing policy. https://grants.nih.gov/grants/guide/notice-files/NOT-OD-24-157.html, 2024. Accessed 2026-11-01.

[13] UCSF Research Cybersecurity. Analyze and Compute Your Large Datasets. https://secureresearch.ucsf.edu/analyze-and-compute-your-large-datasets, 2026. Accessed 2026-11-01.

[14] Stanford Research Computing. Nero GCP: Powerful, Secure Computing for High-Risk Data. https://docs.nero.stanford.edu/, 2026. Accessed 2026-11-01.

[15] Stanford Research Computing. Who Can Use Nero? Nero GCP User Documentation. https://docs.nero.stanford.edu/nero-who.html, 2026. Accessed 2026-11-01.

[16] Stanford Research Computing. Carina Computing Platform. https://docs.carina.stanford.edu/, 2026. Accessed 2026-11-01.

[17] Stanford Research Computing. Carina vs. Nero GCP. https://docs.carina.stanford.edu/prem-vs-cloud, 2026. Accessed 2026-11-01.

[18] Harvard FAS Research Computing. FASSE cluster (FAS secure environment). https://docs.rc.fas.harvard.edu/kb/fasse/, 2026. Accessed 2026-11-01.

[19] Yale Center for Research Computing. Hopper. https://docs.ycrc.yale.edu/clusters/hopper/, 2026. Accessed 2026-11-01.

[20] Duke University. Determine Compute and Data Storage Solutions (myResearchPath). https://myresearchpath.duke.edu/topics/determine-compute-and-data-storage-solutions, 2026. Accessed 2026-11-01.

[21] University of Michigan ITS. Secure Enclave Services (SES). https://its.umich.edu/advanced-research-computing/cloud/secure-enclave-services, 2026. Accessed 2026-11-01; the page blocks scripted `curl` requests and was read through a page-extraction tool.

[22] University of Pennsylvania Information Systems and Computing. Cloud Solutions (including the Secure Research Enclave). https://isc.upenn.edu/services/backup-storage-platforms/cloud-solutions, 2026. Accessed 2026-11-01.

[23] Johns Hopkins University Sheridan Libraries. Secure Storage Choices, Protecting Human Subject Identifiers. https://guides.library.jhu.edu/protecting_identifiers/StorageChoices, 2026. Accessed 2026-11-01.

[24] NYU Research Technology Services. Secure Research Data Environment (SRDE): Eligibility and Accounts. https://services.rt.nyu.edu/docs/srde/getting_started/eligibility_accounts/, 2026. Accessed 2026-11-01.

[25] Columbia University Shared Research Computing Policy Advisory Committee. Minutes of the Spring 2026 Meeting. https://research.columbia.edu/sites/research.columbia.edu/files/content/ORI/SRCPAC/Minutes_SRCPAC%20Meeting_Spring%202026.pdf, 2026. Accessed 2026-11-01.

[26] University of Utah Center for High Performance Computing. Protected Environment at CHPC. https://www.chpc.utah.edu/resources/ProtectedEnvironment.php, 2026. Accessed 2026-11-01.

[27] University of Utah Center for High Performance Computing. Protected Environment Frequently Asked Questions. https://www.chpc.utah.edu/documentation/pefaq.php, 2026. Accessed 2026-11-01.

[28] UC Berkeley Research IT. Secure Research Data and Compute (SRDC) at Berkeley. https://docs-research-it.berkeley.edu/services/srdc/, 2026. Accessed 2026-11-01.

[29] San Diego Supercomputer Center Sherlock Cloud Solutions. Sherlock Cloud Solutions: Solutions. https://sherlock.sdsc.edu/solutions, 2026. Accessed 2026-11-01.

[30] UCLA Digital and Technology Solutions. High Compliance Environment (HCE). https://dts.ucla.edu/products-services/security/high-compliance-environment, 2026. Accessed 2026-11-01.

[31] UCSF IT. Research Analysis Environment (RAE). https://it.ucsf.edu/service/rae, 2026. Accessed 2026-11-01.

[32] Stanford Research Computing. Nero GCP Cost. https://docs.nero.stanford.edu/gcp-cost.html, 2026. Accessed 2026-11-01.

[33] Stanford Research Computing. Carina Cost. https://docs.carina.stanford.edu/cost, 2026. Accessed 2026-11-01.

[34] UC Berkeley Research IT. Faculty Computing Allowance on SRDC. https://docs-research-it.berkeley.edu/services/srdc/srdc_fca/, 2026. Accessed 2026-11-01.

[35] UC Berkeley Research IT. Condo Service on SRDC. https://docs-research-it.berkeley.edu/services/srdc/srdc_condo/, 2026. Accessed 2026-11-01.

[36] San Diego Supercomputer Center Sherlock Cloud Solutions. Sherlock Cloud Solutions: Value and Pricing. https://sherlock.sdsc.edu/value, 2026. Accessed 2026-11-01.

[37] Indiana University IT News. Introducing SecureMyResearch. https://news.iu.edu/live/news/30912-introducing-securemyresearch, 2020. Accessed 2026-11-01.

[38] Simon Li and Oore-oluwa Ayinde. How many RSEs (and related roles) does it take to change a lightbulb in a cage? How RSEs are shaping UK TRE infrastructure. Zenodo, presentation, 2026. Accessed 2026-11-01.

[39] Washington University in St. Louis Research Infrastructure Services. NIST 800-171 access-controlled environment. https://ris.wustl.edu/systems/nist-800-171-access-controlled-environment/, 2026. Accessed 2026-11-01.

[40] Purdue Rosen Center for Advanced Computing. Compute: Weber. https://www.rcac.purdue.edu/compute/weber, 2026. Accessed 2026-11-01.

[41] UC Irvine Information Security. Protect Research Data. https://www.security.uci.edu/how-to/protect-research-data/, 2026. Accessed 2026-11-01.

[42] James Bailie and Ruobin Gong. The Five Safes as a Privacy Context, 2025. Version 2, 2026-09-01. Accessed 2026-11-01.

[43] Christian Cole, Hari Sood, Simon Li, Katie Oldfield, Matt Craddock, and Nel Swanepoel. SATRE: Standard architecture for trusted research environments. Zenodo, version 2.0.1, 2026. Accessed 2026-11-01.

[44] DARE UK. Standardised Architecture for Trusted Research Environments (SATRE). https://dareuk.org.uk/trevolution/standardised-architecture-for-trusted-research-environments-satre/, 2026. Accessed 2026-11-01.

[45] Martin T. O'Reilly. The Turing Data Safe Haven: An open, scalable, reproducibly deployable, cloud-based Trusted Research Environment for working safely with sensitive data. Zenodo, presentation, 2023. Accessed 2026-11-01.

[46] Alan Turing Institute. The Turing Data Safe Haven documentation, version 5.2.1. https://data-safe-haven.readthedocs.io/en/v5.2.1/index.html, 2026. Accessed 2026-11-01.

[47] Alan Turing Institute. Technical controls, Turing Data Safe Haven documentation, version 5.2.1. https://data-safe-haven.readthedocs.io/en/v5.2.1/design/security/technical_controls.html, 2026. Accessed 2026-11-02.

[48] Simon Li, Chuang Gao, James Sutherland, Emily Jefferson, and Christian Cole. TREEHOOSE: an open source, cloud-based trusted research environment. Zenodo, poster, 2022. Accessed 2026-11-01.

[49] Lucas van der Meer, Steven Claeyssens, and Martin Brandt. SANE: Secure ANalysis environment at SURF cloud. Zenodo, presentation, 2022. Accessed 2026-11-01.

[50] Sharing Sensitive Data Using SANE. Zenodo, presentation, 2023. Accessed 2026-11-01.

[51] Peter Marendy. KeyPoint: A universal trusted research environment for sharing, accessing and analysing sensitive data. Zenodo, presentation, 2023. Accessed 2026-11-01.

[52] KeyPoint: Managing identities to enforce strict project separation at the virtual desktop. Zenodo, presentation, 2024. Accessed 2026-11-01.

[53] Evan F. Bollig, Graham T. Allan, Benjamin J. Lynch, Yectli A. Huerta, Mathew Mix, and Edward A. Munsell. Leveraging OpenStack and Ceph for a Controlled-Access Data Cloud, 2018. Accessed 2026-11-01.

[54] Beate Lichtwardt, John Sanderson, and James Scott. Secure Data Facility Professionals Networks: what's in it for your TRE? Zenodo, presentation, 2022. Accessed 2026-11-01.

[55] SATRE project. SATRE specification, standard architecture for trusted research environments. https://satre-specification.readthedocs.io/en/stable/specification.html, 2026. Items 1.2.09 and 1.2.10 read. Accessed 2026-11-01.

[56] Yang Guo, Ramaswamy Chandramouli, Lowell Wofford, Rickey Gregg, Gary Key, Antwan Clark, Catherine Hinton, Andrew Prout, Albert Reuther, Ryan Adamson, Aron Warren, Purushotham Bangalore, Erik Deumens, and Csilla Farkas. High-Performance Computing Security: Architecture, Threat Analysis, and Security Posture. NIST Special Publication 800-223, National Institute of Standards and Technology, February 2024. Accessed 2026-11-02.

[57] University of Michigan ITS. Access or Request Support for Advanced Research Computing (knowledge-base article 12164). https://teamdynamix.umich.edu/TDClient/30/Portal/KB/PrintArticle?ID=12164, 2026. Accessed 2026-11-01; article modified 2026-06-11.

[58] Duke Office of Information Technology. Research Computing. https://rc.duke.edu/protected-network-research, 2026. Accessed 2026-11-01.

[59] Jeffrey A. Marron. Implementing the Health Insurance Portability and Accountability Act (HIPAA) Security Rule: A Cybersecurity Resource Guide. Special Publication 800-66 Revision 2, National Institute of Standards and Technology, 2024. URL https://csrc.nist.gov/pubs/sp/800/66/r2/final. Final 2024-02-14. Accessed 2026-11-01.

[60] U.S. Department of Health and Human Services. 45 CFR 164.514, other requirements relating to uses and disclosures of protected health information (limited data sets and data use agreements). Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-45/section-164.514. Accessed 2026-11-01.

[61] U.S. Department of Health and Human Services. Security and Privacy, 45 CFR Part 164, Sections 164.308, 164.310, 164.312, 164.314, 164.316, 164.404 and 164.410. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-45/part-164. Text current through 2026-10-29. Accessed 2026-11-01.

[62] U.S. Department of Health and Human Services. HIPAA security rule notice of proposed rulemaking to strengthen cybersecurity for electronic protected health information: Fact sheet. HHS.gov, 2024. URL https://www.hhs.gov/hipaa/for-professionals/security/hipaa-security-rule-nprm/factsheet/index.html. Dated 2024-12-27. Retrieved by text extraction; direct command-line requests returned HTTP 403. Accessed 2026-11-01.

[63] Adam H. Greene, Rebecca L. Williams, and Apurva Dharia. HHS updates rulemaking timeframes. Davis Wright Tremaine, Privacy and Security Law Blog, 2026. URL https://www.dwt.com/blogs/privacy--security-law-blog/2026/07/hhs-updates-hipaa-rulemaking-timeframes. Practitioner source, 2026-07-08. Retrieved by text extraction; direct command-line requests returned HTTP 403. Accessed 2026-11-01.

[64] National Institutes of Health. NIH controlled-access data repositories implementation guidebook. grants.nih.gov, 2026. URL https://grants.nih.gov/sites/default/files/flmngr/NIH-CADR-Implementation-Guidebook.pdf. Incident notification within 24 hours of identification. Accessed 2026-11-01.

[65] State of California. California Civil Code section 1798.82. California Legislative Information, 2026. URL https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?lawCode=CIV&sectionNum=1798.82. Subdivision (a) only was read. Retrieved by text extraction; direct command-line requests returned HTTP 403. Accessed 2026-11-01.

[66] California Legislature. California Civil Code section 1798.29. https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?lawCode=CIV&sectionNum=1798.29, 2026. Subdivision (a) read; definitions not read. Retrieved by text extraction. Accessed 2026-11-01.

[67] U.S. Department of Education. Family Educational Rights and Privacy, 34 CFR Part 99, Sections 99.3, 99.30-99.33 and 99.35. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-34/subtitle-A/part-99. Text current through 2026-10-29. Accessed 2026-11-01.

[68] U.S. Department of Education, Student Privacy Policy Office. Data Security: K-12 and Higher Education. studentprivacy.ed.gov, 2026. URL https://studentprivacy.ed.gov/data-security-k-12-and-higher-education. Accessed 2026-11-01.

[69] U.S. Department of Education, Privacy Technical Assistance Center. Data Breach Response Checklist. studentprivacy.ed.gov, 2012. URL https://studentprivacy.ed.gov/sites/default/files/resource_document/file/checklist_data_breach_response_092012_0.pdf. Accessed 2026-11-01.

[70] U.S. Department of Education. 34 CFR 99.60 and 99.67, enforcement procedures. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-34/part-99/subpart-E/section-99.67. Accessed 2026-11-01.

[71] Supreme Court of the United States. Gonzaga University v. Doe, 536 U.S. 273 (2002), syllabus. Legal Information Institute, 2002. URL https://www.law.cornell.edu/supct/html/01-679.ZS.html. Accessed 2026-11-01.

[72] U.S. Department of Health and Human Services. 45 CFR 160.404, amount of a civil money penalty. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-45/part-160/subpart-D/section-160.404. Accessed 2026-11-01.

[73] Google Cloud. HIPAA compliance on google cloud. https://docs.cloud.google.com/security/compliance/hipaa, 2026. Supported-products list dated 2025-09-17; Accessed 2026-11-01.

[74] Google Cloud. Organization policy constraints. https://docs.cloud.google.com/resource-manager/docs/organization-policy/org-policy-constraints, 2026. Accessed 2026-11-01.

[75] Google Cloud. Configure Private Google Access. https://docs.cloud.google.com/vpc/docs/configure-private-google-access, 2026. Accessed 2026-11-01.

[76] Google Cloud. Supported products and limitations, VPC Service Controls. https://docs.cloud.google.com/vpc-service-controls/docs/supported-products, 2026. Accessed 2026-11-01.

[77] Google Cloud. VPC firewall rules. https://docs.cloud.google.com/firewall/docs/firewalls, 2026. Last updated 2026-10-26 UTC; Accessed 2026-11-01.

[78] Google Cloud. Use IAP for TCP forwarding. https://docs.cloud.google.com/iap/docs/using-tcp-forwarding, 2026. Accessed 2026-11-01.

[79] Google Cloud. Dry run mode for service perimeters. https://docs.cloud.google.com/vpc-service-controls/docs/dry-run-mode, 2026. Accessed 2026-11-01.

[80] Google Cloud. Create remote repositories, Artifact Registry. https://cloud.google.com/artifact-registry/docs/repositories/remote-repo, 2026. Accessed 2026-11-01.

[81] Google Cloud. Cloud Workstations and VPC Service Controls. https://docs.cloud.google.com/workstations/docs/vpc-service-controls, 2026. Accessed 2026-11-01.

[82] Google Cloud. About OS Login. https://docs.cloud.google.com/compute/docs/oslogin, 2026. Accessed 2026-11-01.

[83] Google Cloud. Set up OS Login. https://docs.cloud.google.com/compute/docs/oslogin/set-up-oslogin, 2026. Accessed 2026-11-01.

[84] Google Cloud. Shielded VM. https://cloud.google.com/security/products/shielded-vm, 2026. Accessed 2026-11-01.

[85] Center for Internet Security. CIS hardened images on google cloud platform. https://www.cisecurity.org/cis-hardened-images/google, 2026. Accessed 2026-11-01.

[86] Google Cloud. Remote repository overview, Artifact Registry. https://docs.cloud.google.com/artifact-registry/docs/repositories/remote-overview, 2026. Accessed 2026-11-01.

[87] Google Cloud. Uniform bucket-level access. https://docs.cloud.google.com/storage/docs/uniform-bucket-level-access, 2026. Accessed 2026-11-01.

[88] Google Cloud. Bucket Lock. https://docs.cloud.google.com/storage/docs/bucket-lock, 2026. Accessed 2026-11-01.

[89] Google Cloud. Soft delete. https://docs.cloud.google.com/storage/docs/soft-delete, 2026. Accessed 2026-11-01.

[90] Google Cloud. Configure log buckets, Cloud Logging. https://docs.cloud.google.com/logging/docs/buckets, 2026. Accessed 2026-11-01.

[91] Google Cloud. Assured Workloads overview. https://docs.cloud.google.com/assured-workloads/docs/overview, 2026. Accessed 2026-11-01.

[92] Google Cloud. Assured Workloads supported products by control package. https://docs.cloud.google.com/assured-workloads/docs/supported-products, 2026. Accessed 2026-11-01.

[93] Google Cloud. Cloud Workstations overview. https://docs.cloud.google.com/workstations/docs/overview, 2026. Accessed 2026-11-01.

[94] Google Cloud. Configure Private Google Access. https://cloud.google.com/vpc/docs/configure-private-google-access, 2026. Accessed 2026-11-01.

[95] Google Cloud. Use IAP TCP forwarding. https://cloud.google.com/iap/docs/using-tcp-forwarding, 2026. Accessed 2026-11-01.

[96] Google Cloud. VPC service controls overview. https://cloud.google.com/vpc-service-controls/docs/overview, 2026. Accessed 2026-11-01.

[97] Google Cloud. VPC firewall rules. https://cloud.google.com/firewall/docs/firewalls, 2026. Accessed 2026-11-01.

[98] Google Cloud. Route logs to supported destinations. https://cloud.google.com/logging/docs/export/configure_export_v2, 2026. Accessed 2026-11-01.

[99] Google Cloud. Use resource hierarchy for access control. https://docs.cloud.google.com/iam/docs/resource-hierarchy-access-control, 2026. Accessed 2026-11-01.

[100] Google Cloud. Overview of VPC Service Controls. https://docs.cloud.google.com/vpc-service-controls/docs/overview, 2026. Accessed 2026-11-01.

[101] Google Cloud. Service perimeter details and configuration. https://docs.cloud.google.com/vpc-service-controls/docs/service-perimeters, 2026. Accessed 2026-11-01.

[102] Google Cloud. Folder membership in service perimeters. https://docs.cloud.google.com/vpc-service-controls/docs/folder-membership, 2026. Page last updated 2026-10-28. Accessed 2026-11-02.

[103] Google Cloud. Scoped policies (Access Context Manager). https://docs.cloud.google.com/access-context-manager/docs/scoped-policies, 2026. Accessed 2026-11-01.

[104] Google Cloud. Understanding organization policy constraints. https://docs.cloud.google.com/resource-manager/docs/organization-policy/understanding-constraints, 2026. Accessed 2026-11-01.

[105] Google Cloud. Cloud Quotas overview. https://docs.cloud.google.com/docs/quotas/overview, 2026. Accessed 2026-11-01.

[106] Google Cloud. Cloud Billing concepts. https://docs.cloud.google.com/billing/docs/concepts, 2026. Accessed 2026-11-01.

[107] Google Cloud. Customer-managed encryption keys (CMEK). https://docs.cloud.google.com/kms/docs/cmek, 2026. Accessed 2026-11-01.

[108] Google Cloud. Service account permissions: impersonate and attach. https://docs.cloud.google.com/iam/docs/service-account-permissions, 2026. Accessed 2026-11-01.

[109] U.S. Department of Health and Human Services. 45 CFR 164.312, Technical safeguards. https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164/subpart-C/section-164.312, 2026. Accessed 2026-11-01.

[110] U.S. Department of Education. 34 CFR 99.31: Under what conditions is prior consent not required to disclose information? https://www.ecfr.gov/current/title-34/subtitle-A/part-99/subpart-D/section-99.31, 2026. Accessed 2026-11-01.

[111] Google Cloud. Create projects. https://docs.cloud.google.com/resource-manager/docs/creating-managing-projects, 2026. Accessed 2026-11-01.

[112] Google Cloud. Protect resources with Cloud KMS keys (Compute Engine). https://docs.cloud.google.com/compute/docs/disks/customer-managed-encryption, 2026. Accessed 2026-11-01.

[113] Google Cloud. Recommended practices for using CMEKs. https://docs.cloud.google.com/kms/docs/cmek-best-practices, 2026. Accessed 2026-11-01.

[114] University of California Office of the President. UC encryption key and certificate management standard. https://security.ucop.edu/files/documents/policies/encryption-key-and-certificate-management-standard.pdf, 2020. Page footer last updated 2020-02-10; accessed 2026-11-01.

[115] Centers for Medicare and Medicaid Services. Research Identifiable File Data Use Agreement Policy Guide. https://www.cms.gov/files/document/research-identifiable-file-data-use-agreement-policies.pdf, 2026. Accessed 2026-11-01.

[116] Research Data Assistance Center (ResDAC). Research Identifiable File Data Use Agreement. https://resdac.org/sites/default/files/2026-05/RIF-Data-Use-Agreement.pdf, 2026. Accessed 2026-11-01.

[117] Google Cloud. Shared VPC. https://docs.cloud.google.com/vpc/docs/shared-vpc, 2026. Accessed 2026-11-01.

[118] National Institutes of Health. NIH security best practices for users of controlled-access data. https://grants.nih.gov/sites/default/files/flmngr/NIH-Security-BPs-for-Users-of-Controlled-Access-Data.pdf, 2024. Updated 2024-07-25; accessed 2026-11-01.

[119] U.S. Department of Education. 34 CFR 99.33, Limitations on redisclosure. https://www.ecfr.gov/current/title-34/subtitle-A/part-99/subpart-D/section-99.33, 2026. Accessed 2026-11-01.

[120] Stanford Research Computing. Overview of Nero Google Cloud Platform (GCP). https://docs.nero.stanford.edu/gcp-overview.html, 2026. Accessed 2026-11-01.

[121] Google Cloud. Cloud Audit Logs with Cloud Storage. https://docs.cloud.google.com/storage/docs/audit-logging, 2026. Accessed 2026-11-01.

[122] Globus. Sensitive Data Transfer and Sharing User Guide. https://docs.globus.org/guides/overviews/security/user-guide-sensitive-data/, 2026. Accessed 2026-11-01.

[123] Globus. Managing Protected Data. https://www.globus.org/protected-data, 2026. Accessed 2026-11-01.

[124] OpenSSH project. OpenSSH 9.0 release notes. https://www.openssh.com/txt/release-9.0, 2022. Accessed 2026-11-01.

[125] Neutrino Labs. xrdp v0.9.19 release notes. https://github.com/neutrinolabs/xrdp/releases/tag/v0.9.19, 2026. Accessed 2026-11-01.

[126] Neutrino Labs. xrdp sesman.ini configuration file and manual page, development branch. https://raw.githubusercontent.com/neutrinolabs/xrdp/devel/sesman/sesman.ini.in, 2026. Accessed 2026-11-01.

[127] Posit. rsession.conf, Workbench documentation, release 2026.10.1. https://docs.posit.co/ide/server-pro/admin/reference/rsession_conf.html, 2026. Accessed 2026-11-01.

[128] Posit. RStudio source, SessionOptions.cpp. https://raw.githubusercontent.com/rstudio/rstudio/main/src/cpp/session/SessionOptions.cpp, 2026. Main branch. Accessed 2026-11-01.

[129] Office for National Statistics. SRS output checking guidance: Statistical disclosure control. https://www.ons.gov.uk/file?uri=/aboutus/whatwedo/statistics/requestingstatistics/secureresearchservice/gettingyourresearchoutputsapproved/srsoutputcheckingguidance.february2025.pdf, 2025. February 2025 edition. Accessed 2026-11-01.

[130] Centers for Medicare and Medicaid Services. CMS cell suppression policy. https://www.hhs.gov/guidance/document/cms-cell-suppression-policy, 2020. Issue date 2020-01-01; version date 2017-05-08. Read through a text extractor; a direct fetch returned HTTP 403. Accessed 2026-11-01.

[131] U.S. Department of Education, Privacy Technical Assistance Center. Data De-identification: An Overview of Basic Terms. https://studentprivacy.ed.gov/sites/default/files/resource_document/file/data_deidentification_terms_0.pdf, 2013. Document code PTAC-GL, October 2012, updated May 2013. Accessed 2026-11-01.

[132] U.S. Department of Education, Student Privacy Policy Office. Disclosure Avoidance. https://studentprivacy.ed.gov/content/disclosure-avoidance, 2026. Accessed 2026-11-01.

[133] U.S. Department of Health and Human Services. 45 CFR 164.514(b), implementation specifications: requirements for de-identification of protected health information. https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164/subpart-E/section-164.514, 2026. Text current through 2026-10-29. Accessed 2026-11-01.

[134] Richard J. Preen, Maha Albashir, Simon Davy, and Jim Smith. A multi-language toolkit for the semi-automated checking of research outputs, 2022. Read at abstract level. Accessed 2026-11-01.

[135] Katie Oldfield, Layla Robinson, Pete Stokes, and Eilidh Guthrie. Consensus Statement on the use of Automation when Checking the Disclosure Risk of Research Outputs. Zenodo, 2023. Read at record-description level. Accessed 2026-11-01.

[136] Esma Mansouri-Benssassi, Simon Rogers, Jim Smith, Felix Ritchie, and Emily Jefferson. Machine Learning Models Disclosure from Trusted Research Environments (TRE), Challenges and Opportunities, 2021. Read at abstract level. Accessed 2026-11-01.

[137] Emily Jefferson, James Liley, Maeve Malone, Smarti Reel, Alba Crespi-Boixader, and Xaroula Kerasidou. GRAIMATTER green paper: Recommendations for disclosure control of trained machine learning (ML) models from trusted research environments (TREs), 2022. Read at abstract level. Accessed 2026-11-01.

[138] U.S. Department of Education, Privacy Technical Assistance Center. Best Practices for Data Destruction. https://studentprivacy.ed.gov/sites/default/files/resource_document/file/Best%20Practices%20for%20Data%20Destruction%20(2019-3-26).pdf, 2019. Page dated March 2019. Accessed 2026-11-01.

[139] Google Cloud. Create remote repositories, Artifact Registry. https://docs.cloud.google.com/artifact-registry/docs/repositories/remote-repo, 2026. Accessed 2026-11-01.

[140] Google Cloud. Virtual repositories overview, Artifact Registry. https://docs.cloud.google.com/artifact-registry/docs/repositories/virtual-overview, 2026. Accessed 2026-11-01.

[141] Google Cloud. VPCSCConfig, Artifact Registry REST reference. https://docs.cloud.google.com/artifact-registry/docs/reference/rest/v1/VPCSCConfig, 2026. Accessed 2026-11-01.

[142] Google Cloud. Supported formats, Artifact Registry. https://docs.cloud.google.com/artifact-registry/docs/supported-formats, 2026. Accessed 2026-11-01.

[143] Google Cloud. Configure a VM to access a Yum repository, Artifact Registry. https://docs.cloud.google.com/artifact-registry/docs/os-packages/rpm/configure, 2026. Remote Yum use marked Preview; Accessed 2026-11-01.

[144] Google Cloud. Configure authentication to Artifact Registry for Python package repositories. https://docs.cloud.google.com/artifact-registry/docs/python/authentication, 2026. Accessed 2026-11-01.

[145] Open Source Vulnerabilities. PYSEC-2026-508: Compromise of PyTorch Lightning PyPI package versions. https://osv.dev/vulnerability/PYSEC-2026-508, 2026. Record read through api.osv.dev; Accessed 2026-11-01.

[146] Open Source Vulnerabilities. MAL-2026-3201: Malicious code in lightning (PyPI). https://osv.dev/vulnerability/MAL-2026-3201, 2026. Accessed 2026-11-01.

[147] Python Package Index. pytorch-lightning release metadata (JSON API). https://pypi.org/pypi/pytorch-lightning/json, 2026. Upload times read for versions 2.6.1 to 2.6.4; Accessed 2026-11-01.

[148] Ilyas Makari. Popular PyTorch Lightning package compromised by Mini Shai-Hulud. https://www.aikido.dev/blog/pytorch-lightning-pypi-compromise-mini-shai-hulud, 2026. Vendor blog post dated 2026-04-30; Accessed 2026-11-01.

[149] Open Source Vulnerabilities. CVE-2024-27322: deserialization of untrusted data in R. https://osv.dev/vulnerability/CVE-2024-27322, 2026. Accessed 2026-11-01.

[150] Open Source Vulnerabilities. MAL-2026-5315: Malicious code in ensmallen (PyPI). https://osv.dev/vulnerability/MAL-2026-5315, 2026. Accessed 2026-11-01.

[151] Open Source Vulnerabilities. MAL-2026-5274: Malicious code in dynamo-release (PyPI). https://osv.dev/vulnerability/MAL-2026-5274, 2026. Accessed 2026-11-01.

[152] Python Packaging Authority. Secure installs, pip documentation. https://pip.pypa.io/en/stable/topics/secure-installs/, 2026. Accessed 2026-11-01.

[153] Fedora Project. mdapi record for package R in EPEL 9. https://mdapi.fedoraproject.org/epel9/pkg/R, 2026. Reported version 4.6.1, release repository; Accessed 2026-11-01.

[154] Posit. Package security, Posit Package Manager. https://packagemanager.posit.co/__docs__/admin/security/package.html, 2026. Accessed 2026-11-01.

[155] Takaaki Toda and Tatsuya Mori. CHASE: LLM agents for dissecting malicious PyPI packages, 2026. Abstract read; Accessed 2026-11-01.

[156] Siyuan Pang, Yepeng Yao, Zhengwei Jiang, Zijing Fan, Haozhe Li, and Baoxu Liu. PYPILINE: Malicious PyPI package detection via suspicious API knowledge and agent workflow, 2026. Abstract read; Accessed 2026-11-01.

[157] Himadri Chowdhury, Hatem Ahmed, and Joseph Khoury. MTPkgScan: A metamorphic testing framework for package malware scanners. Zenodo record, 2026. Record abstract read; Accessed 2026-11-01.

[158] Aleksandr Churilov. The range shrinks, the threat remains: re-evaluating LLM package hallucinations on the 2026 frontier-model cohort, 2026. Abstract read; Accessed 2026-11-01.

[159] Posit. Air-gapped Package Manager. https://packagemanager.posit.co/__docs__/admin/appendix/air-gapped-installs.html, 2026. Accessed 2026-11-01.

[160] Posit. Repositories and sources, Posit Package Manager. https://packagemanager.posit.co/__docs__/admin/repositories.html, 2026. Accessed 2026-11-01.

[161] Astral. Settings, uv documentation: exclude-newer. https://docs.astral.sh/uv/reference/settings/, 2026. Accessed 2026-11-01.

[162] Astral. Configuration files, uv documentation. https://docs.astral.sh/uv/concepts/configuration-files/, 2026. Accessed 2026-11-01.

[163] Python Packaging Authority. pip install, pip documentation. https://pip.pypa.io/en/stable/cli/pip_install/, 2026. Accessed 2026-11-01.

[164] Paul Nygren and Dustin Mueller. PEP 700: Additional fields for the Simple API for package indexes. https://peps.python.org/pep-0700/, 2026. Accessed 2026-11-01.

[165] conda contributors. Mirroring channels, conda documentation. https://docs.conda.io/projects/conda/en/stable/user-guide/configuration/mirroring.html, 2026. Accessed 2026-11-01.

[166] conda-store contributors. conda-store concepts. https://conda.store/conda-store/explanations/conda-store-concepts, 2026. Accessed 2026-11-01.

[167] Apptainer. Security in Apptainer. https://apptainer.org/docs/user/latest/security.html, 2026. Accessed 2026-11-01.

[168] Apptainer. Signing and verifying containers. https://apptainer.org/docs/user/latest/signNverify.html, 2026. Accessed 2026-11-01.

[169] Apptainer. Support for Docker and OCI containers. https://apptainer.org/docs/user/latest/docker_and_oci.html, 2026. Accessed 2026-11-01.

[170] Jasmine Kendall, Livia Martinescu, Sulamaan Rahim, and Gene Tunny. Interim evaluation of Administrative Data Research UK. https://www.ukri.org/wp-content/uploads/2024/11/ESRC-121124-InterimEvaluationAdministrativeDataResearchUK.pdf, 2024. Oxford Insights and Lateral Economics for ESRC; accessed 2026-11-01.

[171] UK Research and Innovation. Trusted Research Environment User Experience Survey (ESRC). https://engagementhub.ukri.org/esrc-1/royal-tre-test/consult_view/, 2026. Survey landing page; no results published there. Accessed 2026-11-01.

[172] Michael Scherle, Rafael Gieschke, Isabela Mocanu, et al. Open Source Platform for Scalable Multi-Purpose Virtual Desktop Infrastructures. Zenodo, 2025. Abstract-level reading.

[173] Gillian M. Raab, Sophie McCall, and Liam Cavin. Four checks for low-fidelity synthetic data: recommendations for disclosure control and quality evaluation. arXiv, 2025. Abstract-level reading.

[174] Google Cloud. gcloud compute ssh. https://cloud.google.com/sdk/gcloud/reference/compute/ssh, 2026. Accessed 2026-11-01.

[175] Google Cloud. Set up OS Login: required IAM roles. https://cloud.google.com/compute/docs/oslogin/set-up-oslogin, 2026. Accessed 2026-11-01.

[176] Google Workspace Admin Help. Set session length for Google Cloud services. https://support.google.com/a/answer/9368756, 2026. Accessed 2026-11-01.

[177] OpenBSD Project. ssh_config(5): ServerAliveInterval and ServerAliveCountMax. https://man.openbsd.org/ssh_config, 2026. Accessed 2026-11-01.

[178] Google Cloud. gcloud compute start-iap-tunnel. https://cloud.google.com/sdk/gcloud/reference/compute/start-iap-tunnel, 2026. Accessed 2026-11-01.

[179] Google Cloud. Manage OS Login in an organization. https://cloud.google.com/compute/docs/oslogin/manage-oslogin-in-an-org, 2026. States that custom POSIX UIDs must be between 1001 and 60000, or between 65535 and 2147483647; default UIDs are randomly generated. Accessed 2026-11-02.

[180] U.S. Department of Health and Human Services. 45 CFR 164.312, technical safeguards. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-45/section-164.312. Text current through 2026-10-29. Accessed 2026-11-01.

[181] U.S. Department of Health and Human Services. 45 CFR 164.308, administrative safeguards. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-45/section-164.308. Text current through 2026-10-29. Accessed 2026-11-01.

[182] U.S. Department of Health and Human Services. 45 CFR 164.402, definitions (breach, unsecured protected health information). Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-45/section-164.402. Text current through 2026-10-29. Accessed 2026-11-01.

[183] U.S. Department of Health and Human Services. 45 CFR 164.414, administrative requirements and burden of proof. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-45/section-164.414. Text current through 2026-10-29. Accessed 2026-11-01.

[184] Google Cloud. Cloud Audit Logs overview. https://docs.cloud.google.com/logging/docs/audit, 2026. Accessed 2026-11-01.

[185] Google Cloud. Cloud Logging quotas and limits, log retention periods. https://docs.cloud.google.com/logging/quotas, 2026. Accessed 2026-11-01.

[186] Linux man-pages project. auditctl(8), assist controlling the kernel's audit system. https://man7.org/linux/man-pages/man8/auditctl.8.html, 2026. Accessed 2026-11-01.

[187] Google Cloud. Ops Agent configuration. https://docs.cloud.google.com/stackdriver/docs/solutions/agents/ops-agent/configuration, 2026. Accessed 2026-11-01.

[188] Rocky Enterprise Software Foundation. rsyslog.conf, rsyslog package source for Rocky Linux 9 (branch r9). https://git.rockylinux.org/staging/rpms/rsyslog/-/raw/r9/SOURCES/rsyslog.conf, 2026. Default configuration separates authentication messages from the general system log. Accessed 2026-11-02.

[189] Google Cloud. Configure and manage sinks. https://docs.cloud.google.com/logging/docs/export/configure_export_v2, 2026. Accessed 2026-11-01.

[190] University of California Office of the President. UC event logging standard, sections 4.4 to 4.12. https://security.ucop.edu/files/documents/policies/event-logging-standard.pdf, 2020. Last updated 11/04/2020 per the document. Accessed 2026-11-01.

[191] U.S. Department of Health and Human Services. 45 CFR 164.316, policies and procedures and documentation requirements. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-45/section-164.316. Text current through 2026-10-29. Accessed 2026-11-01.

[192] Google Cloud. Monitor your logs: log-based alerting policies. https://docs.cloud.google.com/logging/docs/alerting/log-based-alerts, 2026. Accessed 2026-11-01.

[193] Google Cloud. Logging query language. https://docs.cloud.google.com/logging/docs/view/logging-query-language, 2026. Accessed 2026-11-01.

[194] Google Cloud. Compute Engine audit logging information. https://docs.cloud.google.com/compute/docs/audit-logging, 2026. Accessed 2026-11-01.

[195] Google Cloud. Types of alerting policies and conditions. https://docs.cloud.google.com/monitoring/alerts/types-of-conditions, 2026. Lists alerting on metric absence. Accessed 2026-11-01.

[196] Google Cloud. Ingest Google Cloud data to Google SecOps. https://docs.cloud.google.com/chronicle/docs/ingestion/cloud/ingest-gcp-logs, 2026. Accessed 2026-11-01.

[197] Google Cloud. Security Command Center service tiers. https://docs.cloud.google.com/security-command-center/docs/service-tiers, 2026. Accessed 2026-11-01.

[198] University of California Office of the President. UC information security incident response standard. https://security.ucop.edu/files/documents/policies/incident-response-standard.pdf, 2019. Last updated 8/21/2019 per the document; sections 2.7, 2.9. Accessed 2026-11-01.

[199] National Institute of Standards and Technology. SP 800-61 revision 3, incident response recommendations and considerations for cybersecurity risk management: A CSF 2.0 community profile. https://csrc.nist.gov/pubs/sp/800/61/r3/final, 2025. Published April 2025. Accessed 2026-11-01.

[200] National Institute of Standards and Technology. SP 800-84, guide to test, training, and exercise programs for IT plans and capabilities. https://csrc.nist.gov/pubs/sp/800/84/final, 2006. Published September 2006. Accessed 2026-11-01.

[201] U.S. Department of Health and Human Services. 45 CFR 164.404, notification to individuals. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-45/section-164.404. Text current through 2026-10-29. Accessed 2026-11-01.

[202] U.S. Department of Health and Human Services. 45 CFR 164.408, notification to the Secretary. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-45/section-164.408. Text current through 2026-10-29. Accessed 2026-11-01.

[203] U.S. Department of Health and Human Services. 45 CFR 164.410, notification by a business associate. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-45/section-164.410. Text current through 2026-10-29. Accessed 2026-11-01.

[204] U.S. Department of Education. Family Educational Rights and Privacy, 34 CFR Part 99. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-34/part-99. Text current through 2026-10-29; searched for "breach" with no match; section 99.32 read. Accessed 2026-11-01.

[205] Google Cloud. Soft delete, Cloud Storage documentation. cloud.google.com, 2026. URL https://cloud.google.com/storage/docs/soft-delete. Page states soft delete is on by default with a 7-day default retention. Accessed 2026-11-01.

[206] U.S. Department of Health and Human Services. 45 CFR 164.310, physical safeguards. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-45/section-164.310. Text current through 2026-10-29. Accessed 2026-11-01.

[207] U.S. Department of Education. 34 CFR 99.31, under what conditions is prior consent not required to disclose information. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-34/part-99/subpart-D/section-99.31. Text current through 2026-10-29. Accessed 2026-11-01.

[208] U.S. Department of Education. 34 CFR 99.35, disclosure of information for Federal or State program purposes. Electronic Code of Federal Regulations, 2026. URL https://www.ecfr.gov/current/title-34/part-99/subpart-D/section-99.35. Text current through 2026-10-29. Accessed 2026-11-01.

[209] Simon Li and Oore-oluwa Ayinde. How many RSEs (and related roles) does it take to change a lightbulb in a cage? How RSEs are shaping UK TRE infrastructure. Zenodo presentation, 2026. URL https://zenodo.org/records/22666945. Read at abstract level. Accessed 2026-11-01.

[210] Beate Lichtwardt, John Sanderson, and James Scott. Secure Data Facility Professionals Networks: what's in it for your TRE? Zenodo presentation, 2022. URL https://zenodo.org/records/6631341. Read at abstract level. Accessed 2026-11-01.

[211] Google Cloud. Compute Engine pricing: general-purpose machine types (N2), Iowa (us-central1). https://cloud.google.com/products/compute/pricing/general-purpose, 2026. Accessed 2026-11-01.

[212] Google Cloud. Compute Engine pricing: accelerator-optimized machine types (G2), Iowa (us-central1). https://cloud.google.com/products/compute/pricing/accelerator-optimized, 2026. Accessed 2026-11-01.

[213] Google Cloud. Disk and image pricing. https://cloud.google.com/compute/disks-image-pricing, 2026. Accessed 2026-11-01.

[214] Google Cloud. Sustained use discounts. https://cloud.google.com/compute/docs/sustained-use-discounts, 2026. Accessed 2026-11-01.

[215] Center for Internet Security. Hardened Images Pricing Questions. https://cisecurity.atlassian.net/wiki/spaces/CSKB/pages/1139541045/Hardened+Images+Pricing+Questions, 2024. Knowledge-base page dated 2024-01-25; accessed 2026-11-01.

[216] Center for Internet Security. CIS hardened image level 1 on rocky linux 9, google cloud marketplace listing. https://console.cloud.google.com/marketplace/product/cis-public/cis-hi-l1-on-rocky-linux-9, 2026. Accessed 2026-11-01; estimate table for the default configuration.

[217] Google Cloud. Stop and start a compute instance. https://cloud.google.com/compute/docs/instances/stop-start-instance, 2026. Accessed 2026-11-01.

[218] Google Cloud. Cloud DNS pricing. https://cloud.google.com/dns/pricing, 2026. Accessed 2026-11-01.

[219] Google Cloud. Artifact Registry pricing. https://cloud.google.com/artifact-registry/pricing, 2026. Accessed 2026-11-01.

[220] Google Cloud. Filestore pricing. https://cloud.google.com/filestore/pricing, 2026. Accessed 2026-11-01; the region selector state of the extracted page could not be confirmed.

[221] Google Cloud. Create, edit, or delete budgets and budget alerts. https://cloud.google.com/billing/docs/how-to/budgets, 2026. Accessed 2026-11-01.

[222] Google Cloud. Export Cloud Billing data to BigQuery. https://cloud.google.com/billing/docs/how-to/export-data-bigquery, 2026. Accessed 2026-11-01.

[223] Google Cloud. Enable, disable, or change billing for a project. https://cloud.google.com/billing/docs/how-to/modify-project, 2026. Accessed 2026-11-01.

[224] Office of the Federal Register. 2 CFR 200.413, Direct costs. https://www.ecfr.gov/current/title-2/section-200.413, 2026. Electronic Code of Federal Regulations, current text accessed 2026-11-01.

[225] Office of the Federal Register. 2 CFR 200.468, Specialized service facilities. https://www.ecfr.gov/current/title-2/section-200.468, 2026. Electronic Code of Federal Regulations, current text accessed 2026-11-01.

[226] National Institutes of Health. Supplemental Information to the NIH Policy for Data Management and Sharing: Allowable Costs for Data Management and Sharing (NOT-OD-21-015). https://grants.nih.gov/grants/guide/notice-files/NOT-OD-21-015.html, 2020. Released 2020-10-29; accessed 2026-11-01.

[227] Office of the Federal Register. 2 CFR 200.1, Definitions (equipment). https://www.ecfr.gov/current/title-2/section-200.1, 2026. Electronic Code of Federal Regulations, current text accessed 2026-11-01.

[228] Office of the Federal Register. 2 CFR Part 200, Appendix III, Indirect (F&A) Costs Identification and Assignment, and Rate Determination for Institutions of Higher Education. https://www.ecfr.gov/current/title-2/part-200, 2026. Section C.8.a, Limitation on Reimbursement of Administrative Costs (26% of modified total direct costs). Accessed 2026-11-02.

[229] UCR Information Technology Solutions. Generative AI at UCR (archived copy of the campus page). https://webarchive.ucr.edu/its.ucr.edu/ai.html, 2026. Accessed 2026-11-01.

[230] University of California Office of the General Counsel. Legal Alert: Artificial Intelligence Tools. https://www.ucop.edu/ethics-compliance-audit-services/_files/compliance/ai/ai-alert.pdf, 2024. March 2024; accessed 2026-11-01.

[231] Google Cloud. Accelerator-optimized machine family and GPU machine types (G2, A2, A3, G4). https://cloud.google.com/compute/docs/gpus, 2026. Accessed 2026-11-01.

[232] Qwen Team. Qwen3-8B model card. https://huggingface.co/Qwen/Qwen3-8B, 2026. Accessed 2026-11-01; parameter counts from the repository's safetensors metadata.

[233] Qwen Team. Qwen3-14B model card. https://huggingface.co/Qwen/Qwen3-14B, 2026. Accessed 2026-11-01.

[234] Qwen Team. Qwen3-30B-A3B model card. https://huggingface.co/Qwen/Qwen3-30B-A3B, 2026. Accessed 2026-11-01.

[235] Qwen Team. Qwen3-32B model card. https://huggingface.co/Qwen/Qwen3-32B, 2026. Accessed 2026-11-01.

[236] vLLM Project. Quantization (supported formats and hardware compatibility). https://docs.vllm.ai/en/latest/features/quantization/index.html, 2026. Accessed 2026-11-01.

[237] Hugging Face. Environment variables (huggingface_hub). https://huggingface.co/docs/huggingface_hub/en/package_reference/environment_variables, 2026. Accessed 2026-11-01.

[238] vLLM Project. Usage Stats Collection. https://docs.vllm.ai/en/latest/usage/usage_stats.html, 2026. Accessed 2026-11-01.

[239] Hugging Face. safetensors: a simple and safe way to store and distribute tensors. https://github.com/huggingface/safetensors, 2026. Accessed 2026-11-01.

[240] vLLM Project. Security. https://docs.vllm.ai/en/latest/usage/security.html, 2026. Accessed 2026-11-01.

[241] Mohoshin Ara Tahera, Karamveer Singh Sidhu, Shuvalaxmi Dass, and Sajal Saha. SoK: Privacy-aware LLM in healthcare: Threat model, privacy techniques, challenges and recommendations. *arXiv preprint*, 2026. URL https://arxiv.org/abs/2601.10004.

[242] Google Cloud. VPC service controls with Vertex AI. https://cloud.google.com/vertex-ai/docs/general/vpc-service-controls, 2026. Accessed 2026-11-01.

[243] Inside UCR. UCR community urged to use Google AI tools. https://insideucr.ucr.edu/stories/2026/03/20/ucr-community-urged-use-google-ai-tools, 2026. Accessed 2026-11-01.

[244] UCLA Digital and Technology Solutions. AI use and recommendation guide. https://dts.ucla.edu/initiatives/ai/ai-use-recommendation-guide, 2026. Accessed 2026-11-01.

[245] UC Irvine Information Security. Use Artificial Intelligence Chatbots Securely. https://www.security.uci.edu/how-to/ai-chatbot/, 2026. Accessed 2026-11-01.

[246] Harvard University Information Technology. AI tools. https://huit.harvard.edu/ai/tools, 2026. Accessed 2026-11-01.

[247] Harvard Medical School IT. Generative AI guidelines. https://it.hms.harvard.edu/about/policies-and-guidelines/generative-ai-guidelines, 2026. Accessed 2026-11-01.

[248] University of Michigan ITS. ITS AI services. https://its.umich.edu/computing/ai, 2026. Accessed 2026-11-01.

[249] University of Michigan Safe Computing. U-M GPT, U-M Maizey, and U-M GPT Toolkit Services (Michigan Medicine). https://safecomputing.umich.edu/michigan-medicine/employee-resources/using-gpt-at-michigan-medicine, 2026. Accessed 2026-11-01.

[250] University of Michigan Safe Computing. Artificial Intelligence and U-M Institutional Data. https://safecomputing.umich.edu/protect-the-u/safely-use-sensitive-data/AI-and-UM-Data, 2026. Accessed 2026-11-01.

[251] Google Cloud. Create a Confidential VM instance with GPU. https://cloud.google.com/confidential-computing/confidential-vm/docs/create-a-confidential-vm-instance-with-gpu, 2026. Accessed 2026-11-01.

[252] Google Cloud. Confidential VM supported configurations. https://cloud.google.com/confidential-computing/confidential-vm/docs/supported-configurations, 2026. Accessed 2026-11-01.

[253] Marcin Chrapek, Marcin Copik, Etienne Mettaz, and Torsten Hoefler. Confidential LLM Inference: Performance and Cost Across CPU and GPU TEEs. *arXiv preprint*, 2025. URL https://arxiv.org/abs/2509.18886.

[254] Chenghong Wang. Trusted CI Webinar: Towards Practical Confidential High-Performance Computing. Presentation, Zenodo, 2025. URL https://doi.org/10.5281/zenodo.15723505.

[255] Parminder Singh Reel, Jackson Aaron, Susan Krueger, Christian Cole, Alexander Doney, and Douglas Steele. Designing Machine Learning Experiments using SLURM within a Cloud Trusted Research Environment. Poster, Zenodo, 2023. URL https://doi.org/10.5281/zenodo.13921331.

[256] Jim Madge, Matt Craddock, James Hetherington, Martin O'Reilly, and Tom Young. Harnessing the power of AIRR supercomputers for trusted research. Presentation, Zenodo, 2025. URL https://doi.org/10.5281/zenodo.17968190.

[257] Jake Watson, Sadaf R Alam, Christopher Woods, Abdelwahab Kawafi, Thomas Green, and Ian Johnson. Multi-tenant Kubernetes Use Cases for AI, Secure Computing and Data Services, and More. *arXiv preprint*, 2026. URL https://arxiv.org/abs/2608.00742.

[258] Google Cloud. Filestore service tiers. https://cloud.google.com/filestore/docs/service-tiers, 2026. Accessed 2026-11-01.

[259] Google Cloud. Filestore access control. https://cloud.google.com/filestore/docs/access-control, 2026. Accessed 2026-11-01.

Chapter A

Methods and Sources

Here is how I put the book together, what each kind of source can and can't support, and where the raw material is kept, so a reader can check any claim or repeat the work.

A.1 The subject: a plan, frozen

I examined one design: version 3 of our plan for a shared P3–P4 secure research environment on Google Cloud, with its Terraform, image build scripts, runbooks and templates. I froze the plan at a single repository commit before writing began, and every chapter works from that frozen copy. I checked the Terraform with `terraform validate` and planned it offline with placeholder values (100 resources for the foundation, 59 for a three-study pilot). Nothing was applied to any cloud account, and no image was built. Statements about how the environment "would" behave are readings of the code and documents, not observations of a running system.

The plan had already been through two internal review panels before I started (six lenses on version 1, ten on version 2), run as single-shot model reviews through a campus AI gateway. Their verdicts, and the changes they drove, are used in Chapters 5 and 14. They are internal checks by AI reviewers, not independent audits.

A.2 Evidence gathering

AI-assisted literature reviews. On 1 November 2026 I commissioned eleven reports from an AI-assisted literature review tool, one per question: peer university enclaves; Google Cloud reference architectures; control frameworks; data ingress and egress; researcher experience; offline software supply; logging, monitoring and incident response; cost and funding; multi-tenancy and isolation; GPUs, HPC and AI on sensitive data; and FERPA. Each ran at depth 2 and breadth 3 (a parent run with up to three child tasks; one run spawned two), 43 sessions in total. I archived every session as full JSON alongside the reports, with the prompts as sent, and treated the reports as leads only. They state unverified figures in the same confident, cited style as verified ones, many of their citations are placeholder markers or search-engine redirect links, and my checks found specific errors in them (Table A.1).

Literature search. A rerunnable script of mine queried the arXiv and Zenodo APIs with 31 queries across ten topics (trusted research environments, cloud TRE architecture, output checking, governance, HIPAA research security, secure HPC, offline software supply, researcher usability, LLMs on sensitive data, audit logging). It returned 481 records, which were scored by keyword and recency and screened by hand to 32. I read these at abstract level, and the chapters cite them only for what their abstracts state.

Primary sources. Claims that would change the plan or a conclusion were checked against primary sources: the electronic Code of Federal Regulations for FERPA (34 CFR Part 99) and HIPAA (45 CFR Parts 160 and 164), the Federal Register and the Unified Agenda for the HIPAA Security Rule proposal, NIH notices, University of California policy pages and the UC Protection Level Classification Guide, Google Cloud

documentation and price pages, the OSV vulnerability database, vendor documentation (Posit, xrdp, OpenSSH, Globus), and each peer institution's own web pages. Where a page could not be retrieved or did not say what a report claimed, the chapter labels the claim as reported or leaves it out.

Real artifacts. The only "screenshots" in the book are real tool output: excerpts of the offline `terraform plan` and resource counts drawn from its JSON. Diagrams were drawn from the Terraform and the plan. Any figure that imitates a user interface is captioned as a mock-up. Data charts were computed from published prices or the plan's own numbers, with the scripts kept beside the figures.

A.3 Writing and editing, and how I used AI

The plan, the design decisions and the conclusions are mine and my team's. I used AI writing agents (Claude Sonnet 5.5) to draft chapters from my brief and Claude Opus 5.5 to assemble and condense them, and AI reviewer panels to check the drafts against primary sources. I checked the primary sources behind every correction and am responsible for every claim.

The chapters were drafted in batches from one shared brief that set the sources, the verification and scope rules, and the conventions; a five-reviewer claims panel then checked the assembled draft against primary sources, and Table 14.3 shows how each register row was found.

A.4 What the checking found

Checking against primary sources corrected the research reports and the plan in several places. Table A.1 lists the corrections to outside sources; the plan defects are in Chapter 14.

A.5 Limits

The book has the limits of its method. Peer profiles rest on public web pages, which can lag practice or overstate it. Several sites blocked automated retrieval and were read through a text extractor. Literature was read at abstract level, and the chapters cite it only for what the abstracts state; I do not repeat this caveat at each citation. Prices are list prices on the dates stated, without UCR's negotiated discounts or credits. No part of the environment has been built or tested, so every statement about its behavior is a prediction from code, which the pilot tests in Chapter 14 are meant to confirm or refute. The chapters were drafted by AI agents working from my brief and checked by me and by reviewer panels. The review-method comparison in Chapter 14 is confounded: the reviewers who found the inherited-default defects had the vendor and upstream pages, and no human repository-only review was run. Errors that remain are mine.

A.6 Where the material is kept

The evidence folder holds the prompts, the eleven reports, all 43 archived sessions, the literature search script and its raw and screened results, the verification log, the synthesis and the plan comparison, and the frozen plan. The book folder holds the chapter sources, per-chapter bibliographies, figure scripts, the offline plan artifacts, the shared writer brief, and the review records. I keep both folders.

Table A.1: Corrections to AI-assisted literature reviews and the working gap analysis found by primary-source checks. Each row names the claim and what the primary source says.

Claim	Primary source	Chapter
A public CUI reference build is the vehicle for this P3–P4 design	That build is UCR's separate CUI enclave (Google blog and documentation)	1, 4
Harvard FASSE is rated Level 4/5	Harvard's page: Level 3 only	2
Stanford Nero costs about $100/month and $40/TB	Stanford's page: about $290/month and about $20/TB	2
"SRDE" is a Yale system	SRDE is NYU's service name	2
NIH accepts 800-171 "or equivalent" frameworks	NOT-OD-24-157: ISO 27001/27002 only for non-US users who cannot attest	3
UK applicants: a third waited over 12 months	Not found; ADR UK evaluation: over a third waited over 6 months	9
Six years of locked raw logs is required by HIPAA	45 CFR 164.316(b)(2) governs documentation retention	10
Cited arXiv 2311.02642 for output-checking facts	That identifier is an unrelated paper	7
CMS forbids co-hosting across agreements; 90-day destruction	CMS terms restrict linkage; the 90-day rule covers shipped media	6
Filestore cannot use Kerberos with NFSv4.1	Supported on zonal and regional tiers; Basic HDD is NFSv3 only	13
CIS image fee is per vCPU-hour; ±15% corridor in 200.468	Per instance-hour; no such figure in the eCFR text	12
The 26% administrative cap is statutory	It is regulatory: 2 CFR 200 Appendix III, section C.8.a	12