Lithi: Crowdsourced AI compute, powered by people

Ollie Hawkins
Lithi Technologies LLC
Boston, USA
ollie@lithi.ai

Abstract

Lithi Compute connects the computing needs of individuals, teams, and businesses to a parallel AI network powered by the Macs people already own. Customers continue using their preferred AI agents, including Google Gemini, Claude, ChatGPT, GitHub Copilot, Cursor, or custom applications. In a compatible host, a parent agent turns your goal into bounded jobs for helper agents called subagents. Each helper uses scoped access and an approved spending allocation. Lithi runs suitable batch stages while the parent keeps planning and final review. On the customer's machine, the Lithi Compute Kit prepares permitted files locally. It replaces selected personal information and sensitive identifiers with consistent aliases, a privacy step called pseudonymization. It then encrypts approved blocks before transfer. Original files and private lookup tables never leave the customer's environment.

Across the Lithi Agent Network, each workflow has one coordination unit, called its home cell. It assigns bounded pieces of work called shards. These shards run in parallel across eligible Macs. As demand and Mac availability change, feedback controls compare waiting shards with awake, eligible Macs and independent checking capacity. This paces new assignments within customer spending limits, privacy caps, and each Mac owner's approved hours. Local pseudonymization, encryption, and per-owner disclosure caps protect customer data. The cap covers all Macs controlled by that owner. Under a 5% exposure allowance, full-source execution and checking need at least 40 known owners overall. Each source group's checker has a different owner from its executor. One owner's share stays within its allowance across all its Macs. Customers also choose where their shards run. Local Community Matching prioritizes eligible Macs within 500 miles of the customer's chosen area. It expands outward when task size or deadlines require more capacity, within approved jurisdictions and transfer rules. Global Mixed Matching disperses work across eligible owners within approved jurisdictions. Security separates protecting the provider's Mac from bounding customer disclosure. Signed task workers run inside Apple's macOS App Sandbox with restricted file and network access. They accept data for registered operations, not customer programs. Ordinary model execution still uses readable data in CPU and GPU memory. The execution boundary requires session-scoped keys and encrypted transfers; it does not establish confidentiality from a provider-controlled operating system. Every shard passes its required independent checks before at-most-once acceptance.

Mac owners and business fleet operators use Lithi Connect to contribute chosen hours, including overnight and off-peak windows. This turns idle personal and office Macs into a shared circular compute grid. Half of eligible settled distributable task revenue goes to the provider pool for accepted execution and checking. Invitation rewards come from Lithi's share and help grow local participation. Tokens measure input and output under the operation's fixed counting rules. Customer pricing is $0.05 per million admitted input tokens and $0.10 per million accepted output tokens. Normal platform retries are included. In a worked 1,000-document translation comparison, delegating to Lithi reduces full-task cost by 78.9%, from $35.00 to $7.396. Each month, customers receive a dated impact report. It reconciles completed work, contributor payouts, and electricity records across all participating Macs. Coverage includes execution, checking, failed attempts, and attributable overhead. The report shows contributing Macs, electricity used, and estimated emissions. It distinguishes qualified measurements, calibrated estimates, and missing coverage.

Keywords

crowdsourced compute, Apple silicon, AI agents, distributed inference, owner exposure, fault recovery, result verification, compute accounting, electricity and emissions reporting

1 What Lithi Compute does

How can ordinary Macs help finish your AI work without exposing your whole project? Your agent delegates repeatable bulk work to Macs people and businesses already own. The local Lithi Compute Kit removes unused permitted fields, replaces selected identifiers with aliases, and encrypts approved input before transfer.

The network divides a batch into bounded pieces called shards and limits what each owner can receive across all their Macs. Related records or pages form a source group, the unit used to track disclosure. With a 5% exposure allowance, one owner receives at most 50 of a 1,000-document batch's source groups. Twenty executing owners and twenty different checking owners provide one valid 40-owner allocation. Owners can execute one source group and check another within the same allowance; each group's checker has a different owner from its executor.

Customers choose **Local Community Matching** near their chosen area or **Global Mixed Matching** across eligible worldwide regions. Local matching starts within 500 miles and widens its search with task size and deadline. Like a smart grid for computing, the network matches suitable work to available capacity and returns checked results. Current readings of waiting work and available Macs guide placement and capacity decisions within the approved permissions, budget, and checking rules.

For example, a team gives its preferred AI agent the goal of translating 1,000 support documents into French. The customer sets permitted sources, a deadline and a spending limit. The parent organizes the work and supervises any helper agents. The Kit prepares approved input locally and keeps the private alias map local. Lithi returns checked results and accounting records for the parent to reconcile. Section 13 follows timing, charges and electricity.

The Kit prepares and submits the approved handoff from the environment holding the permitted files. The path has four steps:

- **Step 1, Define.** State the outcome, permitted sources, required context, acceptance checks, deadline, and spending limit. Then assess whether delegation is useful.
- **Step 2, Prepare and approve.** Prepare the input locally, replace selected identifiers with aliases, and encrypt approved content. Approve file transfer and spending separately.
- **Step 3, Run and check.** Lithi assigns bounded shards to eligible Macs in parallel and runs the required checks.
- **Step 4, Review and assemble.** Your agent receives checked files or structured data and joins them by stable source ID. It reviews coverage, exceptions, and charges before use.

Mac owners and business fleet operators use Lithi Connect to choose working hours, including overnight and off-peak windows. They can pause participation at any time. Security answers two questions: what can customer work reach on the Mac, and what customer content can its owner see? Apple's macOS App Sandbox restricts the worker's access to system resources. Lithi's worker policy excludes personal files, accounts, and general network access; tasks supply data for registered operations. On the customer side, local preparation removes unused fields and replaces selected identifiers before encryption. Owner-wide disclosure limits bound the assigned slice, which becomes readable during execution. Process isolation separates applications, while confidentiality from a hostile operating system requires a stronger boundary. Returned results stay untrusted data until the customer's agent validates and reviews them. Receiving a file never runs it automatically.

Figure 1 shows the exchange.

The paper brings together bounded agent handoffs, per-owner disclosure limits, home-cell acceptance and recovery, and reconciled work, payment, and electricity records.

This architecture brings together 18 months of development, from March 2025 to September 2026. Owner-approved hours give existing personal and business Macs another productive use, connecting customer demand with income for contributors. When regional preferences and local availability align, provider allocations support people and businesses contributing in that region.

Clear responsibilities make that exchange possible.

2 Who does what

Every workflow is a partnership with separate responsibilities. Your agent sets the goal, selects permitted files, and reviews the final output. The local Kit prepares the handoff. Global services check account access and credit and choose a coordinating home cell. That cell manages shard assignments, independent checks, and acceptance. Mac owners choose when their computers participate. Customer work, provider administration, and billing keep separate permissions. Your agent cannot pick a specific Mac or change an owner's policy. Assigned Macs receive approved input rather than access to your workspace or account keys. Appendix A gives the detailed permission tables.

The workflow begins with local preparation.

3 Local preparation and bounded disclosure

What actually leaves your machine? Your workspace stays local. The Kit runs alongside your files to prepare a stripped-down, permitted working copy. It removes unused permitted fields and replaces selected personal identifiers, such as names or account numbers, with consistent aliases. This is called pseudonymization: aliases preserve relationships while their private lookup map stays local. It then encrypts approved blocks before upload. Original files and the private alias lookup table remain in your source environment. Assigned Macs read only the prepared portions authorized for their work.

3.1 Keep the meaning

The Kit pins the source version and validates qualified input profiles. Apache Parquet and Apache Arrow organize structured data by column; other profiles cover JSONL, CSV, PDF, media, and archives. The Kit preserves required IDs, types, order, and relationships. Where appropriate, it uses deterministic readers, rule-based parsers that reproduce the same result for the same input and profile. Appendix B gives the format and merge rules.

Preparation can remove unused permitted fields and repeated instructions while retaining required evidence. Oversized context receives a fit decision rather than silent truncation. The Kit prepares and pseudonymizes input locally without a Lithi usage charge. Per-owner sharding limits each assigned Mac to its prepared portion; original files and the private lookup table stay local.

3.2 Remove or replace identifiers

Separate secret and personal-information checks cover the composed request, including instructions, schemas and examples. The Kit returns eligible, review-required or blocked; unresolved review stays local and approval cannot override prohibited disclosure.

Where permitted, the Kit removes unused fields and replaces selected identifiers with consistent aliases. In this example, the customer's name becomes a bracketed alias while the case identifier stays unchanged:

Local source

Example customer requests a refund for `{case_id}`.

Prepared input

[customer alias] requests a refund for `{case_id}`.

`[customer alias]` is an explanatory placeholder rather than a production token format. The private lookup map stays local. Restoration follows the output policy, and uncertain matches require review, as Section F.1 explains. Surrounding free text can still identify a person, so this is pseudonymization rather than full anonymization.

Tool responses contain counts and references by default. Plaintext previews need separate permission because printed content may enter a hosted agent's context. Remote plaintext conversion requires an approved processing stage.

3.3 Approve what leaves

Transfer approval permits uploading the exact encrypted artifact, a sealed file identified by its content fingerprint, to its destination. A changed file requires a new identity and approval. Quote approval separately permits spending on the quoted work.

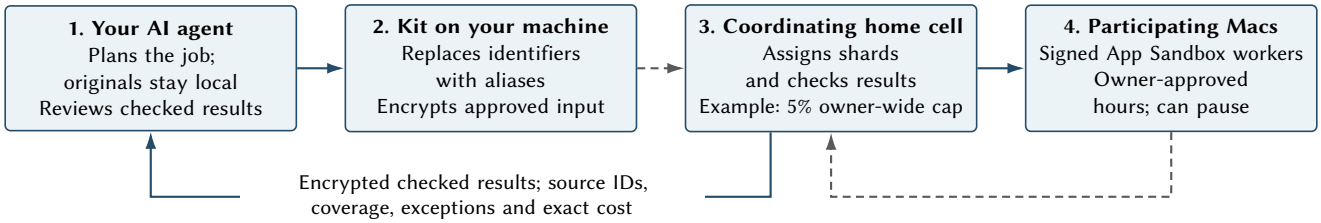


Figure 1: How a job flows from your agent to checked results. The local Kit replaces selected identifiers with aliases and encrypts approved input. The home cell coordinates bounded shards across sandboxed Macs, then returns results, source IDs, coverage, and cost for review.

The prepared input now needs a supported processing path.

4 Choose an outcome and a supported path

Start with the required result, permitted data, checks, and deadline. A workflow connects processing stages; each stage's capability and approved profile fix its inputs, processing method, limits, and checks.

Keep full-project context and local tools with your agent. Compare its current path with the Lithi quote and retained planning, tools, and review. The Kit summarizes supported usage locally; raw account readings remain local [31, 32]. Section 15 compares full cost and outcomes.

4.1 Four workflow patterns

Your agent supplies the outcome, sources, relationships, and policy. The Kit and network compile and adopt the plan using four combinable patterns:

- **Parallel batch:** Apply the same supported operation to independent items, such as ticket labeling or document translation.
- **Staged pipeline:** Pass checked output into the next operation: extract document text, analyze or translate it, then assemble the result.
- **Parallel branches and join:** Process different aspects of a source concurrently, then join results locally by stable source ID or timestamp.
- **Parallel work with a declared merge:** Combine partial counts, sums, or scores while preserving their meaning. This map-and-reduce pattern uses declared reducers, explained in Section 8.3.

Ready windows pace work as eligible execution and checking slots become available. For a product video, one branch uses `video.scenes`; another uses `speech.transcribe`, then `text.translate`. The agent joins checked results locally by timestamp. The separate `speech.subtitle` operation produces subtitles directly from qualified speech audio.

4.2 Capability domains and bulk work

Seven domains organize the 46 capabilities: **agent and batch work**, **documents and PDFs**, **text and language**, **speech and audio**, **images and vision**, **video**, and **model verification**. Appendix B follows this order for exact identifiers, inputs, outputs, and checks.

Batch work beyond code. Fix the question, evidence, result format, and rubric. The `agent.bulk_work` capability organizes bounded items under that recipe; `agent.candidate_set` evaluates declared alternatives against references and the frozen rubric. Generation, evaluation, and acceptance remain separate stages.

For ticket review, supply permitted text and a fixed label set. Supported stages return source-linked findings and exceptions for parent review. The parent controls account updates. Table 1 shows other combinations.

Table 1: People in different roles give their agents goals and delegate suitable stages. These illustrative workflows use permitted evidence, admitted profiles and their own quotes; the parent reviews returned coverage and exceptions.

Person and goal	Helper work through Lithi	Returned work and review
RevOps lead, B2B services: enrich 50,000 permitted company records	Prepare evidence and schema; run parallel classification or rubric stages through `agent.bulk_work` or `text.classify`	Source-linked attributes, labels and exceptions; the parent governs CRM updates
Healthcare quality analyst: find themes in approved feedback	Prepare de-identified or synthetic input; use an admitted analysis profile and Global Mixed Matching within permitted jurisdictions	Findings and coverage for professional review; dispersion remains separate from de-identification
Localization producer: localize licensed product videos	Run scene and transcription branches, then translation; join checked results locally by timestamp	Translated cues and scene observations for media review
Software architect: assess a 400-file refactor	Review pinned source and diffs as static data; evaluate declared candidate fixes	Source-linked findings; the parent keeps edits, builds, tests and final acceptance

Healthcare and routing. A quality analyst asks for themes in approved de-identified feedback. After the organization's required privacy assessment, the parent selects an admitted profile and Global Mixed Matching within permitted jurisdictions. The data must also satisfy the profile's category restrictions. Routing disperses the prepared pieces; de-identification is assessed separately [49, 50]. Returned findings receive professional review.

For code review, supply a pinned revision, its diff—the changed lines—and permitted context. Macs process it as static text and syntax, never compiling or running customer code. The host retains credentials, edits, builds, tests, and acceptance; findings inform review rather than establishing a test pass.

Worked language-model profile. Section 13 illustrates 9,600 input tokens, 1,200 output tokens, and 10,800 combined per call. These budgets cover the complete prompt and returned items; the chosen operation's profile supplies actual admission limits.

4.3 Four local execution paths

A qualified reader prepares input; the profile fixes the operation and runtime, and checks verify its returned result. Placement selects an eligible Mac. Figure 2 and Table 2 show these paths.

Core ML is Apple's on-device model framework [29]. Lithi's `ml.coreml` operation verifies a qualified model specification on CPU. MLX is Apple's open framework for Apple silicon [30], using unified memory shared by the CPU and graphics processor. MLX processes decrypted input as readable text, tokens, and model values in that memory [37]. The approved Gemma 4 12B model has about 12 billion parameters, the learned values shaping its responses. Its 4-bit weights use reduced storage precision to lower memory use. Appendix B pins the model, tokenizer, and template.

A customer selects a registered operation and supplies typed data. Loading a dynamic library adds executable code inside the same process; it creates no new isolation boundary [38]. Workers therefore exclude customer-supplied plugins, shell commands, and native libraries. Required symbols establish adapter compatibility, not trust in its code.

Table 2: Four execution paths on a participating Mac. Each path uses its qualified runtime and checks to preserve the requested meaning, structure, and source coverage.

Processing path	Example work	Acceptance checks
Deterministic: rule-based	PDF extraction, media conversion or deduplication under defined rules	Output, meaning and coverage against the rules
Apple-native: macOS frameworks	Qualified vision, speech, sound or natural-language analysis	Typed results, uncertainty and task checks
Core ML: CPU verification	Verify a qualified model specification on CPU	Verification result for that specification
MLX generative: local language model	Translation or structured text	Required meaning, evidence, structure and coverage

Macs differ in chip, unified memory, and available headroom. Table 12 spans 16 GB to 512 GB configurations. The complete runtime and working set must fit one Mac; Section 8 explains call budgets.

Discovery supplies supported profiles and codecs. Planning checks the reader, release, and limits; quote and admission recheck metering, required checks, and capacity. Unsupported requests receive a reason and next action.

Results carry source IDs, coverage, uncertainty, and abstentions—checks that decline an answer when evidence is insufficient. Speaker turns and face observations describe structure within the catalog's identity limits. The completion policy governs partial delivery, waiting, or stopping.

With the processing path selected, the agent connects through the scoped Work interface.

5 The agent economy: goals, helpers and budgets

How can your agent complete a goal through bounded paid work? Here, an agent economy means software agents arranging and accounting for work under authority supplied by their customers. Lithi connects that demand with eligible computing capacity contributed by Mac owners.

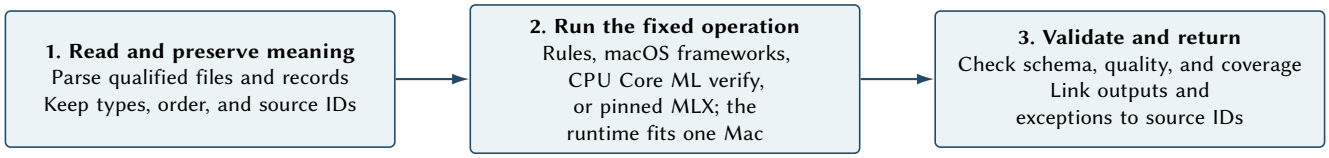


Figure 2: Every admitted stage reads, processes, and checks its input. A qualified reader preserves structure and source IDs; one Mac runs the fixed processing path within local memory. Required checks validate the declared output before it returns with coverage, uncertainty, and exceptions.

5.1 Connect your agent and follow the work cycle

The Kit runs where permitted sources live. Choose a supported connection; host and account controls govern its tools and delegation [1–4, 6, 34, 35].

Table 3: Three ways to connect to the local Kit: agent tools, terminal commands, or application interfaces. Client support varies; pricing, scoped permissions, and approval rules remain the same.

Route	Use
MCP: Model Context Protocol	Compatible AI agents call the scoped Work tools.
CLI: command-line interface	Approved terminal, shell, batch, and scheduled integrations.
Language clients / REST APIs	Software development kits provide client helpers; REST APIs use HTTP requests. TypeScript, Python, and .NET applications use the supported portions.

Copy prompt guides assessment; the customer approves connection changes. After access approval, **Get my API key** opens the protected credential step. The client loads the secret from its approved store, outside prompts and conversations. Connection, transfer, work approval and admission retain separate permissions. The setup guide and Appendix A explain the handoff [5].

Python offers selected helpers and .NET a wire client; preparation and delivery use qualified paths. Caller-side pandas, Polars or PyArrow can produce qualified Parquet or Arrow files, or admitted records. Offline preparation and later approved transfer are separate; network APIs require approved connectivity.

Five Work tools cover the execution cycle:

- **`lithi.quote`** — returns a signed price ceiling and qualified capacity information, without reserving credit or admitting work.
- **`lithi.run`** — submits a bounded job.
- **`lithi.batch`** — submits independent requests that can run in parallel.
- **`lithi.status`** — reports progress across stages.
- **`lithi.result`** — retrieves checked outputs or their references, coverage and exceptions.

The agent checks the profile, prepares permitted input and requests a quote. An authorized person or existing policy approves its bounds. The agent submits through `lithi.run` or `lithi.batch`, monitors progress and retrieves results. The Kit validates delivery and restores permitted aliases; the parent reconciles coverage, exceptions and cost.

Fixed metered counts give reproducible tariff calculations. Quotes also reflect current release, capacity and expiry. Required artifact transfers retain separate approval; quoting admits nothing and reserves no credit.

5.2 Give helpers scoped access and budgets

The parent plans workstreams and dependencies, spawning helper agents where its host supports them. Four responsibilities keep that delegation bounded:

- **Customer and account management:** approve the goal, permitted data, limits and policy; provision credentials and funding through their authorized management and billing access.
- **Parent and host:** assign jobs and spending allocations; control context, concurrency, checkpoints and stopping rules.
- **Helper:** use its assigned connection and Work key within applicable limits.
- **Lithi:** enforce funding and admission, coordinate bounded execution and checking, and record accepted usage.

An authorized issuer provisions each independently managed helper's Work key and protected connection where the host supports separate bindings. The key grants API permissions and expiry; spend policy and shared task limits govern spending. More helpers or credentials create no credit.

For a $10 Lithi task ceiling, the parent might allocate up to $4 to each of two helpers and leave $2 unallocated. These are planning allocations within configured limits. Lithi holds the quoted maximum when it admits the work. Accepted usage and unresolved holds reduce the remaining allowance. The parent tracks its own model and tool costs separately.

Requests retain their approved funded task identity under account, project, key and period limits. Separate tasks still share applicable account/project limits. Work permissions grant no key-issuance, billing or limit-management authority. Provider Macs receive execution permits. Figure 3 shows the division.

5.3 Complete and reconcile approved work

Within approved policy, the parent and helpers can quote, submit, monitor and evaluate unattended. The host governs their context,

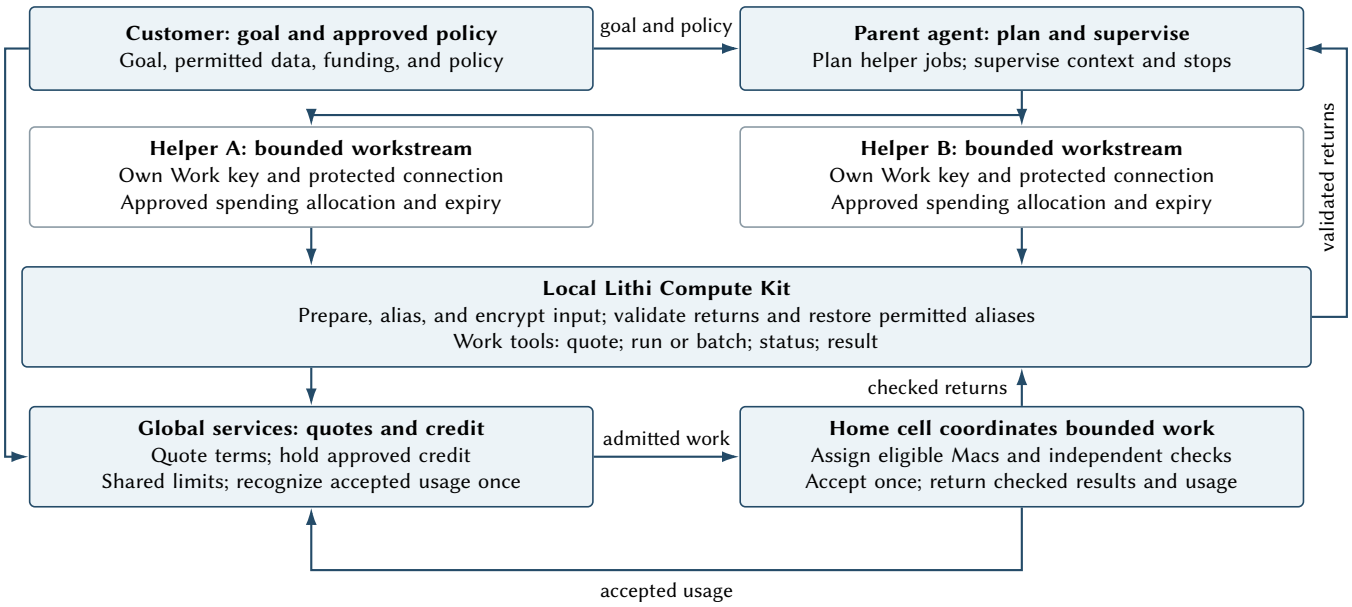


Figure 3: A parent turns a customer's goal into bounded helper jobs. Protected Work-key connections and approved spending allocations reach the local Kit, while global services enforce the funded task and account limits. The Kit validates returned delivery before the parent reconciles results, exceptions and full cost.

dependencies and stopping rules; Lithi handles bounded network execution.

For a new workflow, the parent can use a permitted qualification batch under the same privacy and checking rules. It reviews returned evidence before committing the larger batch. Trial costs enter the full-cost comparison.

Quotes provide commercial terms and forecasts; accepted outputs, coverage and checking records provide completion evidence. The parent reconciles every required outcome, unresolved item and cost before closing the goal. After restart, it resumes the saved plan and original request identities. Section 11 covers recovery; Section A.4 keeps occurrence and destination rules.

Section 15 compares full-task cost. Eligible settled Lithi revenue supports the provider pool, linking agent-directed work with contributed hours on existing Macs. The next section follows the price and approval boundary.

6 Quote, fund and approve

What does the work cost, and when is credit used? A quote states the maximum charge. Approval permits the work; admission reserves credit. Accepted usage consumes that credit.

For language-model work, tokens are the text pieces a model reads or writes. Customer Compute costs $0.05 per million admitted input tokens and $0.10 per million accepted output tokens. A signed metering recipe fixes each representation's count; an executing Mac cannot replace it with its own count. For the dollar charge, multiply the input count by 0.05 and the output count by 0.10. Add the products, then divide by one million, as Equation (1) shows. Host-agent calls, other tools, and review enter the full-task comparison in Section 15.

$$C_{\mathrm{USD}} = \frac{0.05I + 0.10O}{1{,}000{,}000}. \tag{1}$$

Here I and O are the signed recipe's admitted input and accepted output counts, defined in Appendix D. Language-model input includes the complete rendered prompt: instructions, context, and chat-template control tokens counted with the pinned tokenizer [20]. That pin fixes the text-splitting algorithm and vocabulary, so the same canonical input receives the same count on every Mac.

Other recipes count canonical records, image or frame patches, decoded audio duration and tensor shapes under released profiles. Billing tokens count commercial usage, not all physical work; pages, images, audio minutes and Mac-hours carry no separate fees. Normal retries, default checks, delivery and repair do not charge twice for the same logical work. Included independent checks are funded from the task's provider pool under the signed role-allocation policy in Section 14.3. They add physical work and electricity while the customer's logical work is recognized once. A separately requested review stage follows its own quoted scope. Quotes use planned counts and recognition uses accepted work; Appendix D distinguishes the records.

Four decisions remain distinct:

- **Estimate:** a planning calculation that reserves nothing, uploads nothing, and starts nothing.
- **Signed quote:** the account-bound work, maximum price, and expiry Lithi will accept.

- **Execution approval:** permission from an authorized person or existing policy for that work.
- **Admission:** confirmation that the approved work can start within its requirements.

An expired quote needs a fresh quote and approval; adding credit does not revive it.

Table 4 separates collection, admission, acceptance, and recognition.

Table 4: Four separate decisions protect the task's credit. Collection creates credit, admission reserves it, the home cell accepts checked work, and global recognition consumes accepted usage once.

Boundary	Committed result
Collection	A verified purchase issues account credit once; no execution or earnings.
Global admission	Holds the original maximum from eligible credit and grants bounded cell authority.
Home cell	Accepts checked logical work and commits usage evidence. It does not debit global credit.
Global recognition	Consumes accepted usage from the original hold once; releases only the proven unused remainder.

Account, project, key and period limits continue to govern funded work. Lithi reserves the quoted maximum, recognizes exact accepted usage and releases the proven unused remainder; unresolved obligations retain their holds. Stages and subagents share that maximum, and display rounding preserves the exact charge and limit. Appendix D gives metering and reconciliation rules; the billing guide explains account procedures.

Once funded and approved, the workflow needs eligible capacity.

7 Home cells, regional placement and growth

How does the network respond when demand rises or Macs become unavailable? One home cell coordinates each workflow and accepts checked results. Placement uses current eligible capacity; changes to capacity need separate authority.

7.1 Follow a bounded feedback loop

Capacity policy requires six steps, shown in Figure 4:

1. **Measure.** Track waiting shards, eligible execution slots, and independent checking capacity.
2. **Compare with a target.** A target is the operating level allowed by policy, such as a maximum queue wait or concurrency ceiling.
3. **Recommend a bounded change.** Calculate a limited step from current readings and recent history.
4. **Check authority and safety.** Check permissions, jurisdiction, owner exposure, checking capacity, and downstream resource limits.
5. **Apply through an actuator.** An actuator is the separate component authorized to change a live setting. It applies only an approved step.
6. **Confirm by read-back.** Read-back checks live records for confirmed capacity before another adjustment.

The recommendation, request, and confirmed result need separate records. An unresolved change must reconcile before another adjustment. The calculator's recommendation carries no authority to change a queue ceiling or provision a cell.

Table 5 connects readings to their decisions.

Table 5: Five questions connect waiting work with qualified capacity. Local checks protect each Mac; home-cell and admission rules govern shared work, owner exposure, and checking. Capacity recommendations remain separate from authority to act.

Question	What governs the answer
Can this Mac accept another shard?	Lithi Connect checks owner hours, pause, power, heat, and available memory. The home cell separately reserves the owner's disclosure allowance across all their Macs.
How much work may proceed?	A ready window bounds the shards released to execution within slot, independent-checker, and queue limits.
Whose work gets capacity?	Customer fairness governs access to shared capacity; placement ranks eligible Macs. Owner-exposure limits separately protect privacy.
Should a worker ceiling increase?	Recent arrivals, completions, waits, eligible Mac supply, and downstream spare capacity inform a bounded recommendation.
Is another home cell needed?	Repeated pressure across separate peak periods and several resources supports a recommendation. Provisioning still needs authority and qualification.

Mac capacity, cell coordination capacity, and independent checking capacity are separate limits. A higher worker ceiling or another cell helps only with enough eligible Macs and independent checking owners.

Respond to sustained load. Five safeguards limit repeated reactions to brief spikes:

- **Smooth readings.** An exponentially weighted moving average, or EWMA, blends recent arrival and completion counts with history.
- **Use different increase and decrease rules.** This is hysteresis: increases need sustained pressure; decreases need a longer healthy period.
- **Wait between changes.** A cooldown gives a confirmed adjustment time to take effect.
- **Hold at a full boundary.** Backpressure paces releases when downstream capacity is full.
- **Check spare capacity.** Headroom is the capacity remaining in checking, storage, or coordination. Recommendations preserve policy reserve requirements.

Stale or incomplete readings hold a capacity change. These safeguards constrain reactions; stability for a workload needs its own evidence.

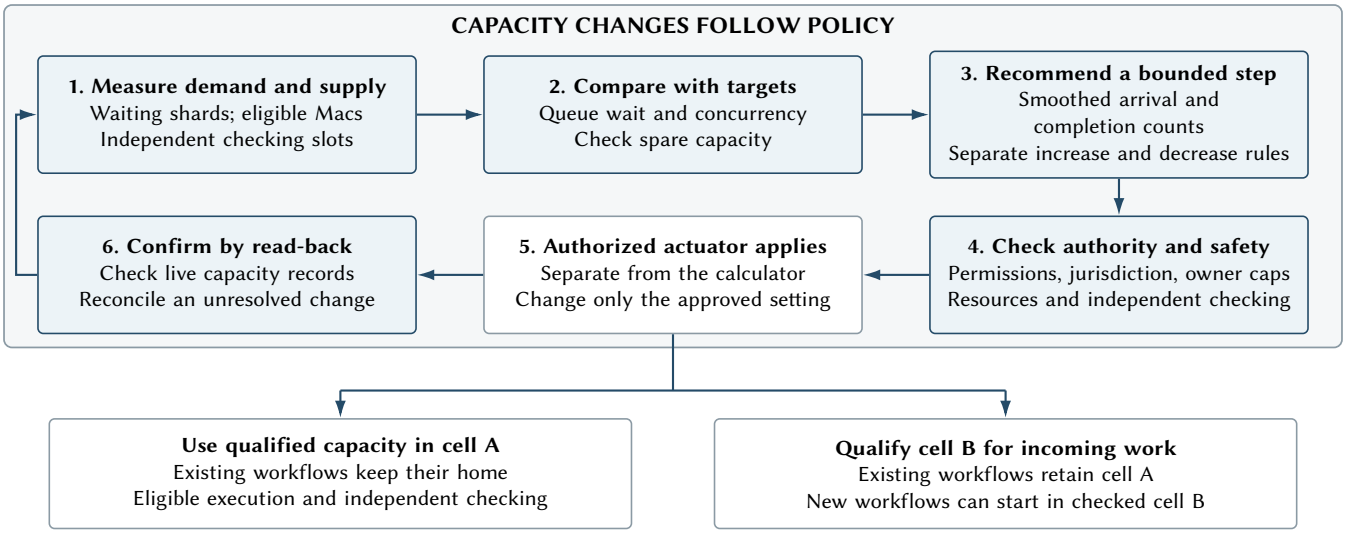


Figure 4: Capacity policy separates measuring, recommending, and acting. A bounded recommendation passes authority and safety checks before a separate actuator may apply it; read-back confirms the result before another adjustment. Qualified capacity can serve the current cell or incoming workflows in a new cell while existing workflows retain their home.

7.2 Match and grow within approved limits

Before ranking Macs, the home cell checks jurisdiction, runtime release, memory, bandwidth, owner exposure, and independent checking capacity. The complete runtime must fit one Mac's shared CPU and graphics memory. Placement follows the customer's choice:

- **Local Community Matching:** Prioritize eligible Macs within 500 miles of the chosen area. Task size, independent-owner requirements, or the deadline can require a wider search within approved jurisdictions and transfer rules.
- **Global Mixed Matching:** Distribute shards across eligible regions and independent owners. Each Mac contributes work that fits its released profile.

Both modes preserve the approved data, jurisdiction, execution, disclosure and checking boundaries. Global Mixed Matching disperses work among eligible owners within those boundaries. More independent local owners provide more regional options. Section 8 defines the disclosure bound; Section 9 connects eligible slots to deadlines.

Growth first uses qualified capacity within a cell's limits, then can prepare another cell for incoming workflows. Existing workflows retain their home. Running jobs keep their approved model, metering, privacy, and checking rules; placement can change within those rules. New defaults apply to new jobs.

A restart recovers in the same cell. An approved move advances the home-assignment version, called an epoch; authority checks reject the old epoch. Before retirement, unfinished work, reserved credit, and retained files must reconcile. Appendix A defines replay and transaction rules.

Capacity, checking, or permission shortfalls pause new admission with status and next actions. Status reports keep customer content and device locations private. Suspected compromise follows Section 12.

Placement begins with bounded pieces of work.

8 Split work without losing meaning or privacy

What is a shard? A shard is a bounded piece of work containing whole input blocks, required context, an operation and limits. Lithi assigns it to an eligible Mac. Your agent supplies the source, outcome and policy limits; the network adopts the execution plan.

8.1 Related input stays together

The Kit keeps related input in source groups and encrypts it into blocks on your machine. Lithi packs whole blocks and required context into shards. The planner uses sizes and dependencies without opening the content.

Independent shards may run in parallel across eligible Macs. Dependent stages wait for their required inputs and checks.

Each assigned Mac runs the complete required local algorithm, native framework or model. Sharding divides data and task work, not model weights. One Mac may process many shards over time; a retry keeps the same shard identity.

8.2 Each call must fit

Each model-based shard fits one Mac's available unified memory: weights, the key-value or KV cache, input, reserved output, and runtime buffers. The cache stores earlier tokens' attention state for reuse during generation.

Count the complete rendered prompt against the input ceiling, including instructions, context, and template control tokens. Reserve output within its own and combined ceilings; do not deduct it again from the input allowance. The illustrative profile permits 9,600 input, 1,200 output, and 10,800 combined tokens per call. Discovery supplies actual limits.

Pack short items together only when their evidence and combined answers fit, preserving related source groups and required context. Resource and owner-disclosure limits also apply.

Illustrative packing. Sixteen records at 200 tokens each contribute 3,200 record tokens. Adding 1,600 tokens for shared instructions gives 4,800 input tokens. Sixty output tokens per record require 960 tokens; another 240 for structure brings the output reservation to 1,200. The call therefore uses 6,000 combined tokens. Output space binds first in this example; actual packing uses the pinned tokenizer and complete rendered call.

Items, calls, shards, workflows, and Macs are different units. A shard can carry several items; a Mac can process several shards. Items sharing a prompt share its context. The compiler fits complete required evidence or returns a permitted partition or no-fit decision. Native and deterministic stages follow their own format, resource, and metering profiles.

8.3 Join without losing meaning

Stable source IDs let your agent assemble results and identify gaps. Before adopting a plan, Lithi checks its dependencies and exact source coverage. It rejects cycles, omitted items and duplicated IDs; a simple total could miss an omission and a duplicate that cancel numerically.

Reducers are merge steps that combine partial shard results under declared rules. They obey the same fit and disclosure limits, using several bounded stages when needed. Counts, sums, ranking, deduplication and joins must preserve their meaning across shards. Appendix B gives the merge rules.

8.4 Limit what one owner receives

Privacy limits follow the owner across all their Macs. One person or business may control several devices, so Lithi enforces the exposure allowance A across the verified owner. Under a 5% allowance on a 1,000-document job, an owner may receive keys for at most 50 source groups. Full-source execution and checking require at least 40 known owners overall. Twenty executing owners and twenty different checking owners form one valid allocation. A source group's checker belongs to a different owner from its executor. An owner may execute one group and check another; both count toward the same owner-wide allowance. Freeze the source-group population G and allowance A at admission. Let E contain groups already made readable to that owner and R its unresolved release reservations. Their combined distinct-group count must stay within the owner's allowance. Count a group appearing in both sets only once. Equation (2) gives this bound.

$$|E \cup R| \le A. \tag{2}$$

Derivatives, overlap, checking, and reducer input refer to the same source population. Repacking, key changes, and migration cannot reset that history. Reservations precede key release. Execution alone needs the source-group count divided by the allowance, rounded up: $\lceil |G|/A \rceil$ owners. Independent full-source checking doubles the source-group count before division and rounding: $\lceil 2|G|/A \rceil$ owners. Retries and resource limits can require more. Exposure follows the recorded owner across enrolled Macs. Device-bound admission, server-controlled placement and independent checking support that boundary. Its assurance still depends on establishing owner identity; payout verification alone does not establish independent owners. Section 18 retains the collusion and false-identity limits.

With the work divided, we can estimate its finish time.

9 Planning and completion forecasts

A whole-job forecast estimates acceptance or requested delivery from the plan, eligible capacity and comparable timing evidence. It includes waiting, processing, checking, retries and delivery, with dependencies and resource limits governing overlap.

The forecast estimates what the current plan can finish. Operational controls govern how much work may proceed within its permissions. Figure 5 shows the forecasting inputs. A generative step first reads its prompt, a phase called prefill. It then writes the response one token at a time, called decoding. Its duration combines both phases and occupied overhead. Overhead includes parsing, transfer, startup, and validation; native operations use their own units. The step starts when its input is ready, a Mac slot is free, and every prerequisite has finished. Adding duration to that start gives the finish time. Equation (3) expresses these dependencies.

$$\begin{aligned} d_v &\approx \frac{I_v}{r_{\text{in}}} + \frac{O_v}{r_{\text{out}}} + d_{\text{overhead},v}, \\ s_v &\ge \max(r_v,\ a_v,\ f_{\text{pred},v}), \\ f_v &= s_v + d_v. \end{aligned} \tag{3}$$

Here d is duration, s and f are start and finish, r is input-ready time, and a is the next slot. P(v) lists prerequisites; $f_{\text{pred},v}$ is their latest finish and is omitted when there are none.

For Local Community Matching, the home cell widens the eligible search beyond 500 miles when local slot availability would push the forecast past the deadline. The wider search preserves the approved jurisdiction, disclosure and checking requirements.

Timing records distinguish waiting, transfer, execution, checking and delivery. The 95th-percentile duration, called p95, is the time within which 95% of comparable runs finish. Adding stage p95 values does not establish a job p95 [15]; insufficient comparable evidence can mean no justified estimate. A forecast states its method, coverage and freshness and cannot relax the deadline, budget or checks. With the illustrative 20-Mac schedule in Table 9, 105 minutes of work and a ten-minute reserve give a latest planned start of 07:05 for a 09:00 deadline.

Recorded waits, transfers, execution, checking, and delivery times can inform later forecasts for the same operation and comparable Mac profiles. Their scope, coverage, and freshness must qualify first. Physical-attempt timing records remain distinct from accepted-outcome labels.

That forecast guides the authorized execution path.

10 Run on shared Macs and accept results

The home cell offers work only after the eligibility, jurisdiction, capacity and exposure checks in Section 7. It reserves the owner's allowance, and the Mac checks its local policy, resources, and lease before claiming during approved hours. A lease is time-bounded permission for that attempt; an expired lease cannot authorize

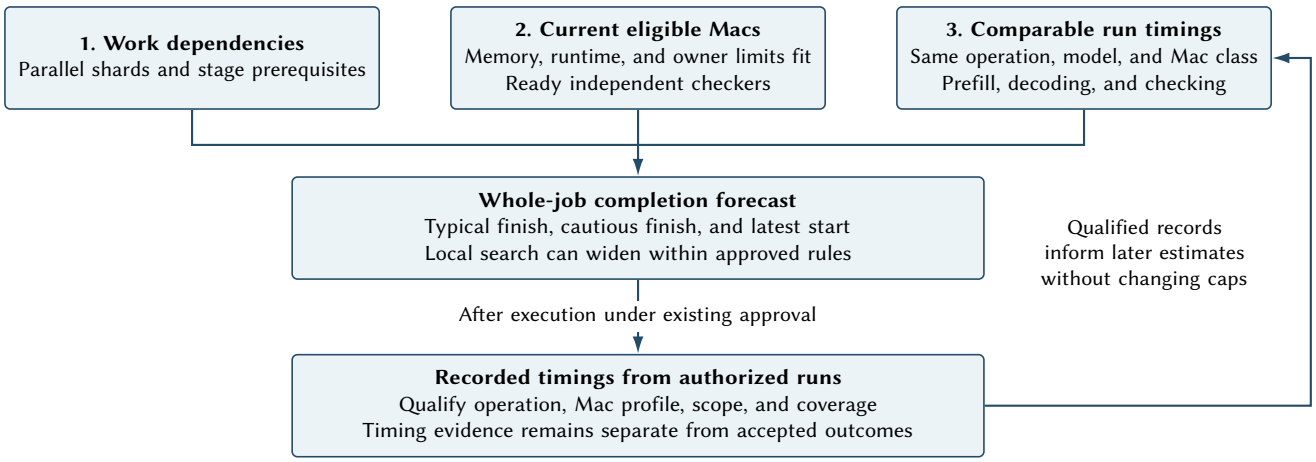


Figure 5: The planner estimates completion from stage dependencies, eligible execution and checking slots, and comparable timings. Within approved rules, a local search can widen beyond 500 miles when the deadline needs more capacity. Qualified recorded timings inform later estimates without changing spending or capacity limits.

more work. The customer approves the task, the owner permits participation and the cell authorizes the attempt. A returned result remains a candidate until checked.

10.1 Separate storage links from key custody

Encrypted input, intermediate and final objects live in cell-scoped storage and are fetched through short-lived links. Anyone holding a link can fetch its ciphertext, the encrypted data, until expiry. Links carry no decryption keys.

The execution boundary requires the isolated task worker to generate its task private key within the active executor session. The Sidebar interface and transport broker carry ciphertext, public-key metadata, and restricted status, rather than task plaintext or usable private keys. This is a process-isolation requirement, not hardware protection against the Mac's administrator. Candidate output is encrypted for its authorized recipients before leaving the worker, including separately authorized checking and customer delivery.

Before releasing keys, the custodian checks the customer account, executing device, worker session, software release, pinned model, task, attempt, lease, fence, expiry, and result recipient together. Current home-cell identity and epoch also bind that authority. Owner exposure is reserved before placement commits, dispatch, and key release. A valid transport signature cannot authorize a substituted recipient key, different worker session, or replayed permit.

Customers choose who releases those keys:

- **Managed custody:** Lithi's key service holds and releases keys for authorized attempts, so it could technically open the input.
- **Customer-held custody:** the Kit on your machine holds and releases keys to authorized Macs. Lithi's service has no input-decryption authority through that release path, and the Kit stays online while work runs.

The assigned executor reads its permitted plaintext, the input unlocked for that assignment, in either mode. The following access map separates required process handling from the host-memory limit. Appendix C explains the cryptographic methods.

Table 6: Who can read which part of a task. The execution boundary requires ciphertext outside the task worker, while authorized executors read their assigned slices. These process rules do not exclude a provider-controlled operating system.

Who	Authorized access and limits
The customer's machine (Kit)	Everything permitted. Original files and the private alias map stay local.
Storage and the network	Encrypted blocks, sizes and timing. Storage links convey no decryption key.
Lithi's planner and home cell	Block references, sizes, limits and bounded check records, rather than task content.
Lithi's key service	Managed custody permits input-key access; customer-held custody does not grant that authority.
Sidebar interface, transport broker and task storage	Required handling: ciphertext and restricted metadata. Task private keys stay session-scoped; application storage excludes plaintext and usable keys.
Assigned task executor	Reads its assigned slice in CPU/GPU memory. A host controlling that execution environment may inspect or retain it.
Other Macs and owners	Normal key release excludes unassigned content. The cap covers known owners, not collusion or compromised shared controls.

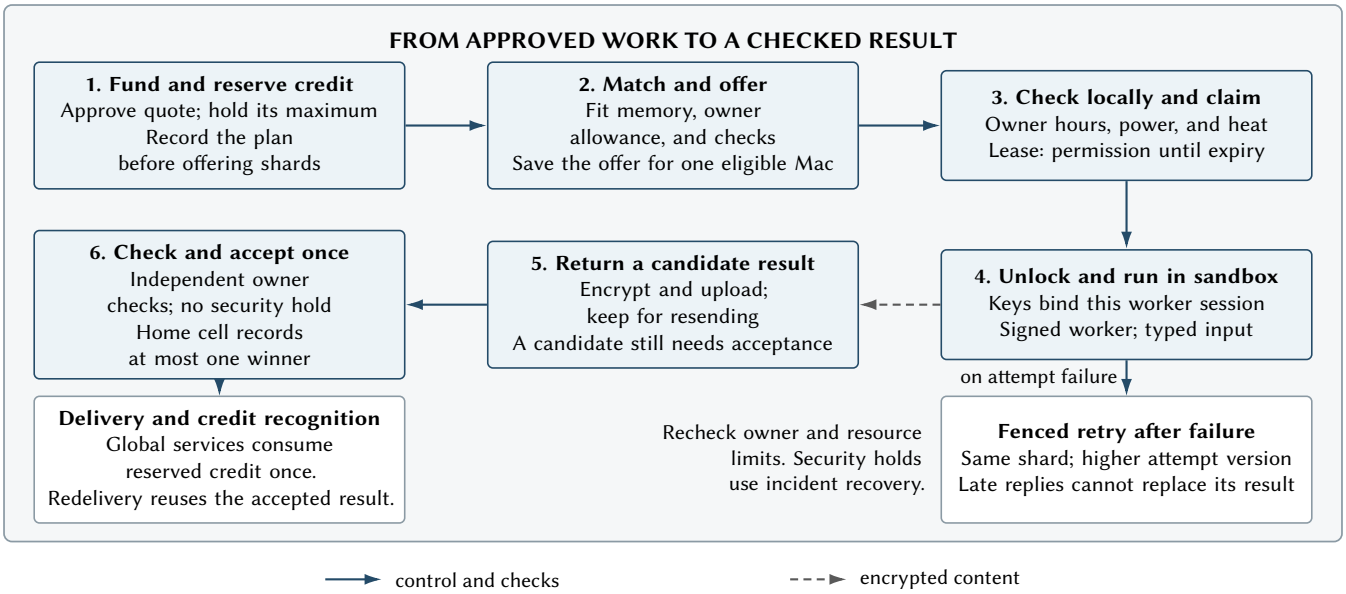


Figure 6: A shard moves from approved funding to one accepted result. After a timed claim, key release checks the current assignment and worker session; the sandboxed executor reads its slice and returns an encrypted candidate. Independent checks precede acceptance, with delivery and global credit recognition recorded separately.

10.2 Protect the provider's Mac and isolate execution state

Lithi Connect checks permission, schedule, power state, foreground activity, thermal headroom, memory, storage, and active work before claiming a shard. The attempt fixes its operation, model, tokenizer, instructions, and limits. Apple's macOS App Sandbox restricts process access [11]; Lithi's execution boundary requires narrower privileges than the personal desktop application:

- **Keep personal state outside the worker.** Permit verified, read-only model assets and narrowly scoped task storage. Exclude the owner's documents, keychain, mail, browser sessions, and payment credentials. A broad shared app-group container must not expose personal application state to task workers.
- **Separate inference from network transfers.** The inference worker has no general network access. A restricted broker performs approved encrypted transfers. Interprocess communication, or IPC, uses typed commands with size limits. Signed peer checks authenticate the calling process.
- **Treat inputs as data.** Signed readers parse documents, media, and code diffs for registered operations. Customer content grants no authority to run a shell, load an executable plugin, or select an arbitrary native library. Public model reuse preserves those permissions.
- **Dispose of private run state.** Task keys, private key-value caches, token arrays, and intermediate buffers belong to the active task. Cleanup waits for native and GPU work to finish, clears owned buffers, and recycles workers where justified. Application logs and caches exclude plaintext. Clearing one variable or ending a process does not prove erasure of all copies, GPU allocations, swap, or host-captured data.

Check hardware evidence separately. A signature proves possession of a key. A software key holder can also sign a claim that the key is hardware-backed. Independent evidence must establish that provenance before the claim affects admission.

Apple's Managed Device Attestation supplies hardware-backed identity and selected device-posture evidence on supported Apple-silicon Macs [39]. Verification requires the expected Apple certificate chain, freshness checks, and binding to the connecting device [40]. Its supported enrollment and refresh rules do not supply an unlimited per-request boot oracle or attest the complete MLX execution state.

Apple also documents Secure Enclave ML-KEM-768 key operations on macOS 26 [41]. Hardware key custody and protection of ordinary MLX memory are different properties. A qualified key mechanism must preserve the selected protocol and recipient binding; it is not a drop-in replacement for X-Wing.

Require new authority after restart. The session boundary requires retiring task keys when the executor ends and retaining only ciphertext and content-free recovery metadata. Resuming work requires a fresh assignment and key exchange, checked against server records outside the provider's snapshot control. FileVault protects locked storage from parties without its unlocking credentials; it does not exclude the Mac's own authorized administrator [42].

The lifecycle comparison keeps each guarantee beside its condition.

Table 7: How shutdown and recovery affect task security. Current server authority can reject old permissions and prevent new disclosures. It cannot erase plaintext already captured during a run.

Scenario	Protection and boundary
Inspect storage after shutdown	Ciphertext-only task storage protects artifacts if usable keys and plaintext were never persisted or captured. Application cleanup alone does not prove the state of OS-managed swap or dumps.
Restore an older app-data snapshot	Current server-side fences, sessions, and epochs reject stale authority. A restarted executor needs a fresh assignment and key exchange.
Change boot security or the OS	Future sensitive key release requires the trustworthy evidence demanded by policy. Missing, stale, or invalid evidence refuses release; device attestation does not prove every application or boot property.
Copy plaintext during execution	Expiry, revocation, or reboot can block future authority, but cannot recall an already copied key or readable result.
Control the executing kernel	Ordinary App Sandbox cannot establish secrecy from that kernel. Pseudonymization and owner caps limit disclosed content. Recovery mode alone does not establish access to a previous session's live memory.

10.3 Check and accept once

A returned result is a candidate until its identity, input, release, and permission are checked. Structural checks assess fields, types and IDs under the operation's declared rules, including canonical comparisons where required. For generative work, valid answers need not use identical wording. The checking policy evaluates the candidate against the permitted source, protected placeholders, glossary and rubric, returning structured findings, uncertainty and any abstentions. Different-owner checking uses another owner's Mac with separately authorized input. Sampled checks report their coverage, while disagreements, abstentions, and missing coverage remain visible.

A candidate that meets the configured checks can be accepted.

The home cell records at most one accepted result per shard and links its usage to that result. Cancellation and acceptance resolve against the same work record: cancellation recorded first prevents later acceptance and charging; acceptance recorded first remains in the accounting history.

Just as customer inputs stay data on provider Macs, returned results remain untrusted data on the customer's machine. Before restoring permitted local aliases, the Kit validates cryptographic integrity, the authorized result recipient, source IDs, exact schema types, and representation. A valid signature establishes the claimed signer, not the truth or safety of the returned content. It blocks path traversal, tricks that try to write outside the approved folder, along with unexpected file types and unapproved overwrites. Returned patches, scripts, macros, and spreadsheet formulas gain no execution authority. Receiving or accepting a result never runs it automatically. Your agent validates and reviews delivered files before applying or using them.

Where the workflow permits, the Kit validates and writes accepted files or structured data in the approved workspace. Your agent can inspect the manifest, coverage and bounded exceptions without replaying the whole batch through its conversation. Required semantic review remains part of the task and its cost.

The same rules continue through attempt failure and recovery.

11 Recover the step that failed

Recovery repeats only the failed step under the original limits. A replacement attempt receives a higher version number, called a fence. That fence blocks delayed effects from an older attempt from replacing the current result; physical work can still repeat. Normal retries and authorized redelivery add no customer charge. Uncertain actions keep their holds until reconciliation, and speculative copies require explicit policy. Zero credit still permits authorized retained-result access and safe cancellation. Table 8 identifies the recovery for each failure.

Table 8: Recover the failed step while keeping the same logical work identity. Attempts may repeat physically, but saved acceptance and usage records prevent another charge for the same accepted work.

Failure	Permitted recovery
Submission reply lost	Look up or repeat the original request using the same request identity.
Work message lost	Resend the saved message for the same work.
Mac sleeps or fails	Recheck limits and assign a new attempt at the same shard.
Result acknowledgment lost	Resend the saved encrypted result without running the work again.
Required check fails	Preserve the exception; retry or stop only under the declared policy.
Delivery destination fails	Retain the accepted artifact and retry the authorized delivery.
Cell loses permission to act	Block its old assignments and reconcile trusted records before moving the work.

Suspected compromise calls for a different kind of recovery.

12 Contain a breach

Controls outside a suspect cell pause affected assignments and key releases. Unaffected cells can continue only while shared controls remain trusted; a compromised shared dependency can require a wider stop.

Figure 7 shows breach recovery. Independent trusted evidence supports accepted results, and the old cell stays fenced. Reconciliation carries financial obligations, each owner's exposure history E, and unresolved release reservations R into authorized recovery. In the illustration, trusted records retain 700 translations without charging again and leave 300 for a clean cell. A compromised Mac

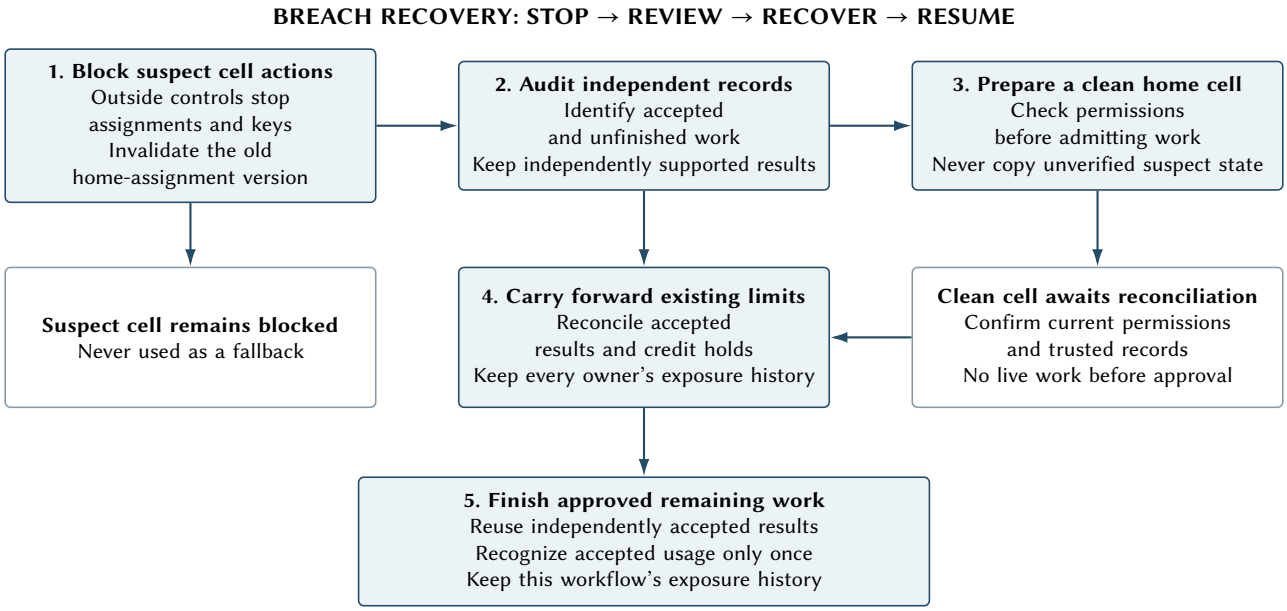


Figure 7: Outside controls pause suspect work while trusted evidence identifies accepted and unfinished results. A clean replacement receives reconciled permissions, credit obligations, and owner exposure history. Only approved unfinished work resumes; migration resets neither disclosure caps nor accepted charges.

stays excluded until separately repaired and qualified. An executor restart requires fresh session authority and key exchange; restoring local files cannot restore server-side permission. Revocation stops future authorized releases and assignments, while previously disclosed plaintext remains outside recall. Appendix F explains the remaining risks.

These rules come together in the next 1,000-document translation example.

13 The 1,000-document task, end to end

How do quoting, sharding, checking and accounting fit together? This section applies the rules from Sections 6, 8 and 9 to the hypothetical translation batch introduced in Section 1 (Figure 8). It reconciles the whole task and each role under stated timing, wall-power and allocation assumptions.

13.1 Assumptions

- **Workload and packing:** 1,000 support documents of about 600 words each, one whole document per shard, preserving IDs, placeholders, glossary terms and structure.
- **Metered tokens:** 6,000 input tokens including instructions and context, and 1,000 output tokens per shard. The assumed profile permits 9,600 input, 1,200 output and 10,800 combined; actual limits come from discovery. The planned totals are 6 million input and 1 million output tokens: a $0.40 quote.
- **Occupied time:** Assume 100 seconds of translation and 20 seconds of independent checking per document, totaling 2,000 Mac-minutes with eligible resources ready. Generation produces the full translation; checking can read source and candidate and emit a shorter verdict, reducing decoding work. These durations are illustrative; Equation (3) also accounts for input processing and occupied overhead. Local preparation and delivery add five assumed minutes.
- **Power boundary:** An assumed 40 W occupied-time-weighted mean for each translation and checking role across the mixed fleet. Individual Macs and operations draw different power; this calculation covers whole-Mac wall input during the occupied slots.
- **Revenue allocation:** 50% of eligible settled accepted revenue forms the provider pool, with assumed 80/20 weights for translation and checking.

13.2 Preparation and approval

The customer's parent agent keeps the glossary decisions, overall plan, and final exception review. It checks the source, destination, two-hour deadline, $0.40 budget, and completion checks before choosing delegation and scoped subagents. First, it compares the bounded translation stage with the current path. The following English instruction and French translation illustrate a protected case identifier and glossary term. For example, given "Restart the Lithi desktop app before opening case `{case_id}`." and a glossary mapping "case" to a support ticket, the French output is "Redémarrez l'application de bureau Lithi avant d'ouvrir le ticket `{case_id}`." Verification checks meaning, terminology, document ID, and placeholder preservation. The instruction is translated rather than executed.

Locally, the Kit prepares and seals the 1,000 documents into encrypted blocks. The adopted plan packs them into 1,000 one-document shards under the $0.40 quote, using Equation (1).

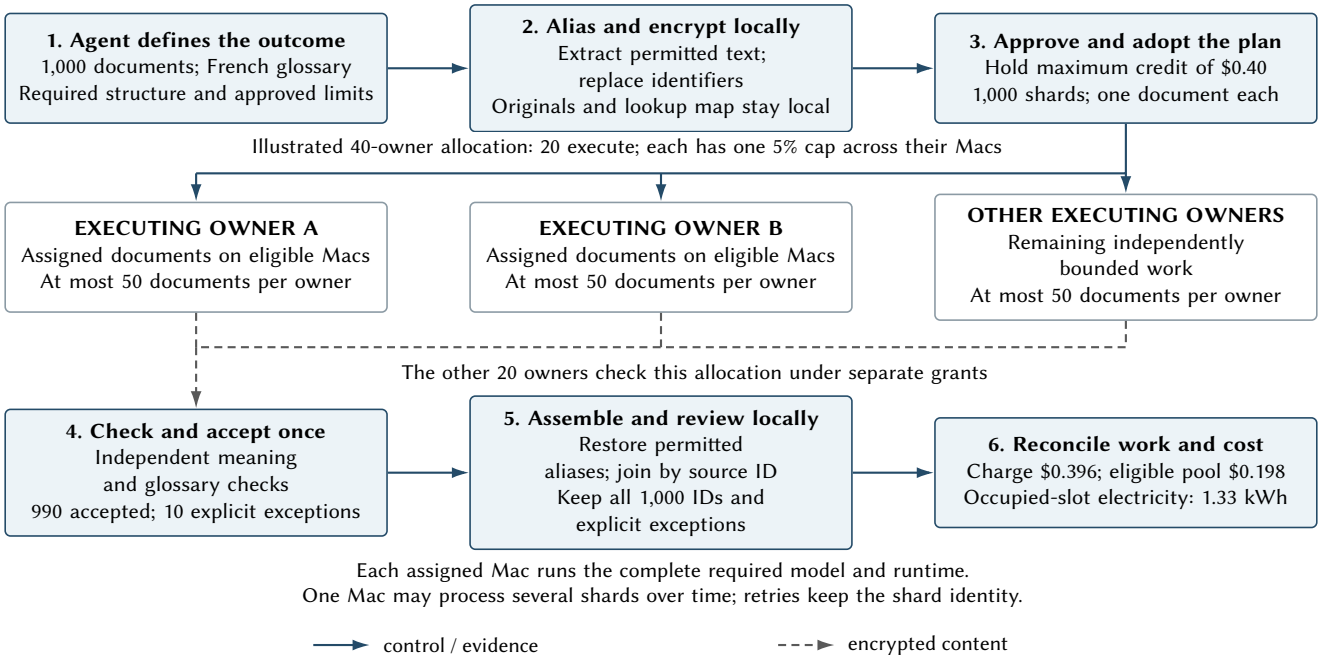


Figure 8: The illustrative 1,000-document batch fans out after local pseudonymization and encryption, then joins by source ID. At the 5% allowance, this 40-owner allocation uses 20 executing and 20 different checking owners. Other schedules can share roles across different source groups, with timing, acceptance, cost and electricity calculated under their stated assumptions.

Once funded, the home cell assigns shards to eligible Macs, as Figure 8 shows. A 5% owner exposure allowance permits 50 source groups per owner: translation requires at least 20 distinct owners, and independent full-source checking raises the overall minimum to 40. Each document's checker has a different owner from its executor; roles can differ for other documents. The checker reads the permitted source, glossary, candidate, and rubric within its profile. Default checking is included in the quote.

13.3 Time

At 120 combined seconds per document, the task totals 2,000 Mac-minutes, or 33 hours and 20 minutes of occupied Mac time. The exposure allowance requires multiple owners. Table 9 illustrates translation and checking time across 20, 50 and 100 concurrent Macs. With 100 concurrent slots, 1,000 shards are processed over time in about 20 minutes.

Table 9: How parallel slots shorten the illustrative batch's turnaround. Its 2,000 Mac-minutes equal 33 hours and 20 minutes serially, or about 20 minutes with 100 concurrent slots under the stated scheduling assumptions.

Macs working at once	Owners across both stages	Translation and checking
20	40	About 1 hour 40 minutes
50	50	About 40 minutes
100	100	About 20 minutes

These schedules assume homogeneous stages and eligible resources ready. With 20 concurrent Macs, 20 owners translate 50 documents each and 20 different owners check them. With 50 or 100 concurrent Macs, owners can execute and check disjoint sets within their allowance. In the illustrative 100-owner schedule, each owner translates ten documents and checks ten different documents: twenty source groups within the fifty-group allowance. Preparation

and delivery add five assumed minutes. Transfer delays, switching and retries can extend completion; Section 9 forecasts actual workflows.

If an owner pauses. Consider a separate variation on this batch: one translating owner pauses after 30 of their 50 documents have been checked and accepted. Local checks withhold new claims, and the Mac's availability updates the home cell's view of supply.

An unfinished attempt follows its existing completion or cancellation rules. If its lease expires before acceptance, recovery uses a higher attempt fence (Section 11). Another eligible executing owner can take the 20 unfinished documents only with room under the owner-wide 50-source-group allowance. Checking still uses independent owners; the 30 accepted translations stay recorded.

The forecast reflects the remaining eligible Macs. The $0.40 quote ceiling, privacy rules, and checking standards remain fixed. This interruption changes the timing assumptions in Table 9 rather than inheriting its illustrative finish time.

13.4 Accepted results

Suppose 990 shards pass the declared checks and are accepted, while ten remain unresolved after their attempted translation and checking.

The parent joins accepted outputs by document ID, restores only permitted fields, and reports 990 documents as partial delivery. Translation does not automatically recreate a PDF's layout.

Accepted documents arrive as files with source identities and coverage records. The parent agent reviews those files and resolves exceptions; Section 15 includes its retained planning and review calls in the full-task cost.

13.5 Electricity

The 20-concurrent-Mac schedule uses 40 distinct Macs and owners across two stages. The physical shares are 5/6 translation and 1/6 checking; the separate 80/20 revenue weights allocate the provider pool. Attempts on unresolved documents still consume electricity. The full-precision reconciliation is in Appendix D.

Table 10: Electricity for the illustrative translation and checking batch. An assumed 40 W occupied-time mean gives 1.33 kWh across both roles. The rows separate time, power, and energy.

Role and Macs	Time per Mac	Electricity
Translation (20 Macs)	83 min 20 sec	55.56 Wh mean per Mac; 1.11 kWh total
Checking (20 Macs)	16 min 40 sec	11.11 Wh mean per Mac; 0.22 kWh total

Together these roles consume approximately 1.33 kWh, excluding activity outside the occupied slots.

13.6 Charges and allocations

Table 11: Reconcile partial delivery without treating a hold as a charge. The example reserves $0.40, recognizes $0.396 of accepted usage, and allocates an eligible provider pool separately.

Line	Value
Held at admission	$0.40, the quote maximum
Recognized charge	$0.396 for 990 documents: 5.94M input and 0.99M output tokens; $0.004 stays reserved while obligations are unresolved
Provider pool, half	$0.198: $0.1584 to translating Macs and $0.0396 to checking Macs under the assumed 80/20 role weights
Lithi's share before operating costs	$0.198
Mac time	2,000 Mac-minutes, including attempts on the ten unresolved documents
Execution and checking electricity	About 1.3333 kWh across 2,000 Mac-minutes under the stated 40 W wall-power assumption; activity outside those slots is excluded

Once the ten unresolved items close without further accepted usage, reconciliation releases the unused $0.004 to the account's available credit. The $0.396 charge remains recorded for the 990 accepted translations. Section D.1 reconciles the balance.

Section 15 compares full task cost for this 1,000-document population using a separate 24-hour deadline and 95% completion floor: $7.396 with Lithi against $35.00 for the agent alone. Task allocations are separate from cash payouts and monthly earnings (Section 14; Appendix E.2). Payment-processing fees apply to the credit purchase rather than each task. The parent reconciles delivered files, source coverage, exceptions and full cost before using the result.

The task's allocations lead us to the Mac owners who provided its capacity.

14 The people and businesses sharing their Macs

What does an owner agree to, and what do they earn? Owners can use Lithi Connect directly or begin with the separate provider **Copy prompt** in a supported agent. The agent reviews the intended provider account, permitted Macs, working times and available management tools, then prepares changes for approval. Owners review permissions and consent to participation. Only qualified Macs receive assignments.

The Lithi Provider CLI, `lithi-provider`, supports discovery, plans, management and read-back through commands exposed by the approved build. It begins with version and capability checks. Provider access is separate from customer Compute credentials and does not authorize customer spending. The provider setup guide explains the handoff.

Table 12: Monthly and annual planning illustrations for 12 available hours a day, after estimated electricity and before tax. Memory and runtime fit determine eligible work; actual earnings depend on accepted contributions and settled revenue.

Mac	Chip	Memory	Monthly illustration	Over 12 months
MacBook Air	M4	16GB	$43.15	$517.80
MacBook Air	M4	32GB	$65.65	$787.80
MacBook Pro	M4 Pro	24GB	$105.10	$1,261.20
MacBook Pro	M4 Max	64GB	$172.60	$2,071.20
Mac mini	M4	16GB	$49.06	$588.72
Mac mini	M4 Pro	48GB	$120.10	$1,441.20
Mac Studio	M4 Max	64GB	$194.84	$2,338.08
Mac Studio	M3 Ultra	512GB	$285.73	$3,428.76
iMac	M4	16GB	$54.45	$653.40

14.1 Choose when to participate

Lithi Connect uses signed, sandboxed workers for customer data and registered operations. The worker boundary requires less access than the personal desktop application: approved models and task storage, with personal files, accounts, and general networking excluded. Customers supply data, rather than programs to install or execute. Section 10.2 explains these requirements and the separate host-memory limit.

Customer credit and provider enrollment are independent: owners do not buy credit to contribute, and customers need not enroll a Mac. A shared installer or invitation does not transfer device identity or consent. Setup identifies the device, release, permissions, hours, and timezone; selection, consent, enrollment, and qualification remain distinct.

Owners can permit unattended claims during approved hours, including overnight or off-peak windows, pause, or withdraw. Before claiming work, Lithi Connect checks approved hours, pause state, power, heat, and available resources. When local conditions restrict participation, the Mac reports its availability and withholds new claims. Placement uses current eligible capacity; existing attempts follow their completion or cancellation rules. Signed updates preserve consent. Withdrawal reconciles remaining work and removes only Lithi-owned components.

14.2 Manage a selected fleet

The same worker boundary applies to every fleet Mac. Task permissions exclude company files outside the run, corporate intranets, shared personal application state, and customer-supplied commands. IT deployment does not widen those permissions. A customer using its own or an explicitly trusted managed fleet changes which administrators it trusts; fleet management alone does not create confidential execution memory.

A provider organization selects Macs it is authorized to manage and begins with a fixed pilot, verifying it before expanding to another reviewed group. Multiple personal Macs do not require a business account. Existing approved tools can deliver verified software. Secure Shell, abbreviated SSH, provides remote command-line login. Mobile Device Management, or MDM, configures company Macs centrally. The provider plan records the intended account, targets, and policy.

The organization's existing MDM performs its deployment actions [33]. Each Mac still needs valid provider authorization, its own device identity, required permissions and runtime qualification. Installation, enrollment, qualification and current availability are separate results. Roles can permit device administration without exposing earnings or payout controls. Fleet reports retain every selected target, including unavailable Macs and those awaiting action. Remote changes remain subject to local checks and applicable managed policy.

Adding Macs does not multiply an owner's source-exposure allowance or create independent checking owners. Section 8's bound continues across devices controlled by that verified owner. The fleet guide covers setup and management.

14.3 How earnings are allocated

Work arranged by customer agents feeds the provider pool, connecting bounded task budgets with computing hours that people and businesses choose to contribute. Half of eligible settled distributable revenue forms the provider pool. The signed policy divides it among Macs whose execution and checking produced accepted results. An inviting owner earns 10% of the invited Mac's eligible earnings. That reward comes from Lithi's share and is capped at 5% of task revenue. The invited owner keeps their full allocation.

More independent owners in a region expand local execution and cross-owner checking options, helping more regional work fit inside the local matching area. Lithi handles customer acquisition, billing, support and coordination from its share. Allocated revenue reports each owner's contribution; electricity and other operating costs determine net return.

Separate invitation example. If an invited Mac earns $1 from eligible settled work, the inviting owner earns $0.10 from Lithi's share and the invited owner keeps the full $1. The 5% task-revenue cap applies. This is separate from the $0.396 translation-task illustration.

The provider pool equals half of eligible settled distributable task revenue. Each Mac's gross allocation sums its policy-weighted

share of those pools across tasks. Equation (4) distinguishes the shared pool from an individual Mac's allocated earnings.

$$\begin{aligned} P_j &= 0.5D_j, \\ G_m &= \sum_j w_{m,j} P_j. \end{aligned} \tag{4}$$

Here D_j is eligible settled distributable revenue for task j, P_j is its provider pool, and $w_{m,j}$ is Mac m's allocation weight. For each allocated pool, the signed policy's weights are non-negative and sum to one. G_m is that Mac's gross allocated earnings across tasks.

Earnings statements identify accepted work and the owner's share once the customer has paid. Earned, payable, and paid amounts remain separate; fractions of a cent carry forward. Payouts use verified, settled funds after declared taxes, refund reserves, and customer obligations are covered.

14.4 What the calculator illustrates

Planning illustrations. The calculator illustrates the owner's share for selected Mac configurations, available hours, and electricity price. This example uses 12 available hours a day over a 30-day month. Annual figures multiply the monthly illustration by twelve. Actual earnings follow demand, eligible accepted contributions, and settled payment.

Each figure assumes 12 available hours a day (360 hours in a 30-day month) and electricity at $0.1835 per kWh, the US average residential price, June 2026 (EIA Electric Power Monthly, Table 5.3). It is in US dollars, after estimated electricity and before tax.

Table 12 uses the Lithi Connect calculator [22] and dated EIA price [23].

The calculator starts with a hardware planning figure for 240 available hours and scales it to the hours selected. Family and chip-class premiums and memory adjustments are planning assumptions, rather than measured throughput rates. Electricity uses 80% of the stated power reference: Apple's published maximum power for desktops and bundled-adapter bounds for notebooks. Adapter ratings include charging and do not measure working draw. Monthly figures are rounded to cents; the 12-month illustration is the monthly figure multiplied by 12. Displayed estimates have a zero lower bound. Actual statements report recorded work and payments.

Apple's desktop power references state their measurement conditions [24] [25] [26]. Notebook bounds use the applicable adapter references [27] [28]. The full calculator guide lists every available illustration and explains how to use other hours or your own electricity price. Hardware compatibility, workload qualification and an available earnings illustration are separate checks.

Table 12 spans 16 GB configurations through 512 GB workstations. Chip capability and available headroom decide which released workloads fit each Mac. Smaller systems support extraction, native analysis, and checking. More memory fits larger working sets and supported concurrency. Each Mac runs a complete runtime. Input-heavy work can return labels, vectors, or extracted fields. Accepted results, cost, coverage, and completion time describe the contribution.

For customers, the corresponding question is the full cost of accepted work.

15 Compare results and improve the network

Does delegation improve value? Compare the full cost of reaching the same required result. Keep the source population, output requirements, acceptance checks, deadline, and completion policy fixed. Record preparation, execution, checking, delivery, coverage, elapsed time, cost, and electricity. Operational controls react within approved settings; experiments compare permitted alternatives under fixed task requirements. Quality, safety, permissions, and owner protection remain constraints when comparing cost and speed.

A configuration c fixes the host agent model, instructions, guidance, tools, task policy, and stopping rules. Full cost adds host-agent calls, Lithi usage, other tools, local preparation and review, and setup and evaluation. Equation (5) keeps these costs within the same task boundary.

$$\begin{aligned} C_c = {} & C_{\text{host},c} + C_{\text{Lithi},c} + C_{\text{tools},c} \\ & + C_{\text{local/review},c} + C_{\text{setup/evaluation},c}. \end{aligned} \tag{5}$$

Host cost covers agent-model calls. Lithi cost counts recognized usage rather than unspent credit purchases. Tool cost covers external services. Local preparation and exception review form one cost component. Setup and evaluation cover preparing and assessing the comparison. Count unsuccessful calls and trials in their incurred category. Included platform retries never become another customer charge.

Table 13 illustrates a reference host-model API comparison for translation over 24 hours with a 95% completion floor. Configuration A processes 6 million input and 1 million output tokens on the host model. At $3 per million input and $15 per million output, its model charge is $33.00. Configuration B incurs $0.396 for 990 documents on Lithi and assumes $5.00 in host planning and coordination. Both add $2.00 for local setup and review. Ratios use exact totals before rounding.

For each configuration, count accepted outcomes that meet the declared deadline and quality floor. Full task cost is C_c and the accepted-outcome count is S_c. Dividing cost by outcomes gives cost per accepted outcome. Reversing the ratio gives accepted outcomes per dollar, as Equation (6) shows. The first ratio needs a positive outcome count; the second needs a positive cost.

$$\begin{aligned} \text{Cost per accepted outcome} &= \frac{C_c}{S_c} \quad (S_c > 0), \\ \text{Accepted outcomes per dollar} &= \frac{S_c}{C_c} \quad (C_c > 0). \end{aligned} \tag{6}$$

Table 13: Full-task costs for 1,000 translated documents under the worked assumptions and reference API rates. Keeping planning and review with the host while delegating bulk translation lowers cost by 78.9%, from $35.00 to $7.396, with about 4.9 times as many checked outcomes per dollar.

Metric	Agent alone	Agent plus Lithi
Agent model calls	$33.00	$5.00
Lithi usage	—	$0.396
Review and setup	$2.00	$2.00
Full task cost	$35.00	$7.396
Checked / assigned	960 / 1,000	990 / 1,000
Outcomes per dollar	27.4	133.9
Cost per checked document	$0.0365	$0.0075

Under this reference baseline, delegating bulk translation lowers full-task cost by 78.9%, from $35.00 to $7.396. The assumed accepted outcomes produce 133.9 outcomes per dollar with Lithi and 27.4 with the host model alone, about 4.9 times as many. A parent agent can use this method to compare work allocations under an approved budget.

In this illustration, Lithi usage accounts for $0.396 of the full-task cost. Half of that eligible settled revenue provides the $0.198 provider pool; host-model and local review costs remain separate components.

Report cash cost and subscription allowance in their actual billing pools. Provider terms remain separate; missing prices and evidence stay unknown. Include failed trials, local review, and external costs. Compare coverage and quality at the same endpoint. Removing unnecessary tokens lowers both the bill and provider pool while preserving the required work.

Test a change before approving it. Compare a permitted setting with the current baseline under fixed task requirements.

A separate authority approves release; running jobs keep pinned settings. Failures and unresolved work count, with retries retained in their original sample group. Consent, privacy, payout, and tariff constraints stay fixed. Comparison creates no training rights. Figure 12 and Section E.4 explain the release boundary.

The same work records also support electricity and emissions reporting.

16 Report contribution, electricity and emissions

Existing Macs perform useful work during owner-approved hours. Impact reports connect that delegated work with contributor allocations and qualified electricity records for the same period. Local matching can support regional participation; Global Mixed Matching spreads eligible participation across permitted regions. Reports reconcile past activity; operational control decisions use fresh resource and queue readings.

Each month, customers receive a dated report covering every participating Mac within the declared physical boundary. This includes execution, independent checking, failed attempts, and attributable overhead, even where no output is accepted. Electricity intervals link to work and reconcile with task allocations and network totals. The report shows contributing Macs and owners, electricity used, and estimated emissions. Shared interval energy is allocated once. Measurements, calibrated estimates, and missing coverage stay distinct; missing readings never count as zero.

16.1 Where did the number come from?

Qualified meters supply measurements; calibrated device profiles supply labeled estimates. CPU time, temperature, and charger ratings alone do not establish whole-Mac energy. Calculator assumptions remain separate from operational records.

16.2 What boundary does it cover?

Whole-Mac wall AC input and internal device DC load have different boundaries; never sum them. Occupied-slot wall power includes other device activity. Incremental task-energy claims require a declared idle or foreground baseline and separate evidence.

16.3 What activity is included?

Multiply each interval's average power in watts by its duration in seconds. Sum the products and divide by 3.6 million for electricity in kilowatt-hours. Equation (7) applies this rule. Section 13 counts occupied translation and checking slots and excludes activity outside those slots.

$$E_{\mathrm{kWh}} = \frac{\sum_k P_k \, \Delta t_k}{3{,}600{,}000}. \tag{7}$$

Compare electricity using accepted results under the same task and quality requirements. Keep failed-attempt energy, unresolved work, missing intervals, and the measurement boundary visible. Lower energy with less completed work may be worse. Input-processing and generation rates explain performance but do not replace accepted work as the denominator.

Multiply each interval's electricity in kilowatt-hours by its regional, hourly grid factor. The factor expresses grams of carbon dioxide equivalent per kilowatt-hour.

Equation (8) sums these products into estimated grams of carbon dioxide equivalent.

$$M_{\mathrm{gCO_2e}} = \sum_k E_{\mathrm{kWh},k} \, \mathrm{CI}_k. \tag{8}$$

The report records each grid factor's source, method, version, uncertainty, and coverage.

Software Carbon Intensity, abbreviated SCI, combines operational and allocated hardware emissions per declared unit of work [14]. An avoided-emissions comparison needs a matched baseline. Section E.5 links hypothetical participation, distributed memory, decoding capacity, and electricity consumption.

Report earnings and qualified electricity for the same scope and period. Neither proves the other.

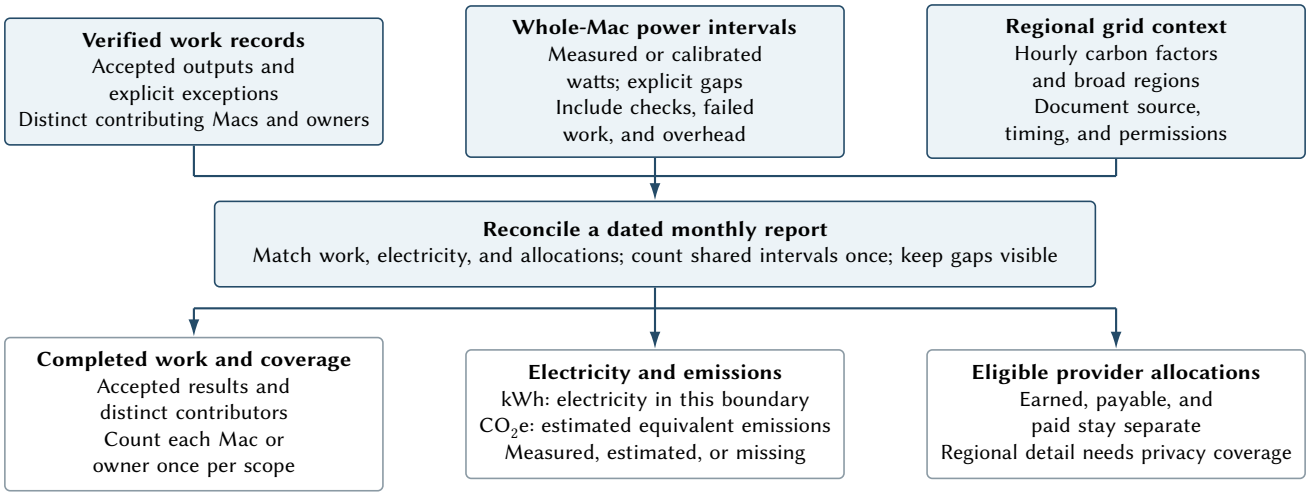


Figure 9: The monthly report reconciles completed work, contributor earnings, and electricity within one declared period and boundary. It combines accepted records, all relevant Mac intervals including checking and failed attempts, and regional grid factors. Measurements, calibrated estimates, missing coverage, and payment states stay distinct.

Figure 9 and Table 14 separate these evidence types. Retries can consume electricity without accepted output. Count distinct Macs, owners, accepted-work contributors, and paid participants separately. Count overlapping child contributions once per task.

Table 14: Seven questions answered by a dated impact report. Completed work, electricity, estimated emissions, and provider payments use their own evidence and coverage, so one total cannot stand in for another.

Customer question	Reported answer and qualification
What finished?	Accepted output, document coverage, checks, and exceptions.
Who contributed?	Distinct Macs and owners for defined populations; privacy-safe regional ranges when permitted.
How much electricity?	Qualified kWh, boundary, measurement or model provenance, and coverage.
What emissions?	Estimated CO2e, grid factors, uncertainty, and missing lifecycle and infrastructure components.
What reached providers?	Earned, payable, and paid records separately; never infer cash from participation.
How do I verify it?	Report period and revision, calculation methods, coverage, and file integrity checks.
How do I compare value?	Compare accepted coverage and full cost, then qualified electricity and the owner's allocation over the same boundary and period.

Regional reports need complete ownership coverage and at least five independent owners per released broad region. Suppress small-group inferences, keep household addresses private, and publish no live device map.

Keep separate dates for work, energy, usage, and cash. Later evidence creates a dated revision rather than rewriting the earlier report.

These design choices build on earlier work in distributed and volunteer computing.

17 Related work

Volunteer computing showed that idle personal computers can support distributed scientific work. SETI@home and Folding@home used donated machines [17, 18], and BOINC generalized public-resource computing [19]. Lithi focuses on paid work from a customer's existing agent, per-owner exposure limits, home-cell acceptance and recovery, and linked work, payment, and energy records.

Protecting data from the machine's own operator requires a different platform boundary. Apple's Private Cloud Compute combines hardware-rooted software measurements, restricted executable releases, removal of general privileged runtime access, and per-boot storage-key protection on its Apple-silicon nodes [43–45]. These properties come from the platform as a whole. A consumer Mac application, sandbox, or dynamically loaded framework does not reproduce that operator-exclusion model.

These distinctions guide the choice of execution environment.

18 Limitations

Assigned plaintext and host memory. Ordinary MLX execution handles readable text, tokens, and model state in unified CPU/GPU memory. Process isolation and ephemeral keys restrict intended handling; they do not make that memory confidential from a provider-controlled kernel. Local preparation replaces selected identifiers, while surrounding text may remain identifying. Owner-wide caps bound disclosure to known owners and do not defeat collusion or false identities.

Workload fit. The caller's context, tool access and usage restrictions, as well as current operations and profile limits, decide

what can run. More Macs cannot make unsupported or indivisible over-limit work fit. Tightly synchronized workloads may be better run elsewhere.

Fallible checks. Configured checks can miss errors; unresolved items and required human review remain visible.

Illustrative figures. Unless labeled as measurements, task timings, wall-power figures, role weights and calculator outputs (Section 14.4; Appendix E.2) are planning illustrations. They do not establish measured hardware performance or promised income. Measurements apply only to their declared workload, runtime release, hardware and acceptance checks.

Participation and reporting coverage. Work depends on demand and current eligibility. Electricity and emissions reports depend on coverage and declared boundaries; missing evidence stays visible.

Choose a stronger boundary when required. If a task cannot disclose even its prepared slice to an unrelated provider, select a trust model that meets that requirement. The alternatives below change either the trusted administrator or the computation itself; they are not extra guarantees supplied by ordinary sharding.

Table 15: Three ways to strengthen the execution trust boundary. Each requires its own qualification; none follows merely from encrypting input around ordinary MLX inference.

Execution model	Security benefit	Condition or trade-off
Customer-owned or trusted managed workers	Keeps readable input within the chosen administration.	Trust in that fleet's administrators and physical security remains.
Confidential CPU/GPU execution	A qualified hardware boundary can exclude specified host operators from in-use data.	Attestation, launch policy, secret release, and the precise backend threat model must agree [46].
Cryptographic computation	Fully homomorphic encryption computes over ciphertext; secure multiparty computation distributes secrets across participants.	Both require different algorithms and workload measurements. Multiparty security also depends on its declared corruption and collusion model [47, 48].

Those boundaries keep the model useful when choosing which work to delegate.

19 Conclusion

Lithi Compute lets customers give their preferred AI agents goals and delegate suitable batch stages to existing Macs. In compatible hosts, parent agents coordinate bounded helpers under scoped access and approved budgets. Local preparation, bounded disclosure and independent checking return source-linked results the parent can review and reconcile. Half of eligible settled distributable task revenue supports the people and businesses contributing those computing hours. Bounded feedback matches demand with qualified capacity within customer limits and each Mac owner's control. Dated work, payout and electricity records make the value of that shared computing visible to its participants.

References

[1] Model Context Protocol. Specification (2026-07-28). https://modelcontextprotocol.io/specification/2026-07-28
[2] Anthropic. Claude Code subagents and delegation controls. https://code.claude.com/docs/en/sub-agents
[3] OpenAI. Codex Model Context Protocol connections. https://learn.chatgpt.com/docs/extend/mcp?surface=cli
[4] Cursor. Model Context Protocol. https://cursor.com/docs/mcp
[5] Agent Skills. Specification and progressive disclosure. https://agentskills.io/specification
[6] OpenAPI Initiative. OpenAPI Specification. https://spec.openapis.org/oas/latest.html
[7] NIST. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (SP 800-38D). https://csrc.nist.gov/pubs/sp/800/38/d/final
[8] IETF. Hybrid Public Key Encryption (RFC 9180). https://www.rfc-editor.org/rfc/rfc9180
[9] D. Connolly, P. Schwabe and B. Westerbaan. X-Wing: General-Purpose Hybrid Post-Quantum KEM. Internet-Draft, revision 06. Reference revision, not a final RFC. https://www.ietf.org/archive/id/draft-connolly-cfrg-xwing-kem-06.html
[10] IETF. HMAC-based Extract-and-Expand Key Derivation Function (RFC 5869). https://www.rfc-editor.org/rfc/rfc5869
[11] Apple Developer Documentation. Protecting user data with App Sandbox. https://developer.apple.com/documentation/security/protecting-user-data-with-app-sandbox
[12] Apache Parquet. Logical types. https://parquet.apache.org/docs/file-format/types/logicaltypes/
[13] Apache Arrow. Columnar format. https://arrow.apache.org/docs/format/Columnar.html
[14] Green Software Foundation. Software Carbon Intensity specification. https://sci.greensoftware.foundation/
[15] J. Dean and L. A. Barroso. 2013. The Tail at Scale. Communications of the ACM 56(2), 74-80. DOI 10.1145/2408776.2408794. https://research.google/pubs/the-tail-at-scale/
[16] W. Hoeffding. 1963. Probability Inequalities for Sums of Bounded Random Variables. Journal of the American Statistical Association 58(301), 13-30. https://doi.org/10.1080/01621459.1963.10500830
[17] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky and D. Werthimer. 2002. SETI@home: An Experiment in Public-Resource Computing. Communications of the ACM 45(11), 56–61. https://doi.org/10.1145/581571.581573
[18] M. Shirts and V. S. Pande. 2000. Screen Savers of the World Unite! Science 290(5498), 1903–1904. https://doi.org/10.1126/science.290.5498.1903
[19] D. P. Anderson. 2004. BOINC: A System for Public-Resource Computing and Storage. Fifth IEEE/ACM International Workshop on Grid Computing, 4–10. https://doi.org/10.1109/GRID.2004.14
[20] Hugging Face. Chat templating documentation. Checked 2026-10-26. https://huggingface.co/docs/transformers/main/chat_templating
[21] MLX. Prompt processing and generation measurements (mlx_lm.generate). Checked 2026-10-26. https://github.com/ml-explore/mlx-lm/blob/main/mlx_lm/generate.py
[22] Lithi. Connect earnings calculator and methodology. Checked 2026-10-27. https://lithi.ai/connect/docs/estimate-earnings
[23] U.S. Energy Information Administration. Electric Power Monthly, Table 5.3. June 2026 residential electricity price. Checked 2026-10-27. https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=table_5_03
[24] Apple. Mac mini power consumption and thermal output information. Checked 2026-10-27. https://support.apple.com/en-us/103253
[25] Apple. Mac Studio power consumption and thermal output information. Checked 2026-10-27. https://support.apple.com/en-us/102027

[26] Apple. iMac power consumption and thermal output information. Checked 2026-10-27. https://support.apple.com/en-us/109513
[27] Apple. MacBook Air (13-inch, M4, 2025) technical specifications. Checked 2026-10-27. https://support.apple.com/en-us/122209
[28] Apple. MacBook Pro power adapters and charging. Checked 2026-10-27. https://support.apple.com/en-us/109509
[29] Apple Developer Documentation. Core ML: on-device model execution. Checked 2026-10-28. https://developer.apple.com/documentation/coreml
[30] MLX. A framework for machine learning on Apple silicon. Checked 2026-10-28. https://ml-explore.github.io/mlx/build/html/index.html
[31] OpenAI. ChatGPT pricing and plan usage limits. Checked 2026-10-29. https://learn.chatgpt.com/docs/pricing#what-are-the-usage-limits-for-my-plan
[32] Anthropic. How do usage and length limits work? Checked 2026-10-29. https://support.claude.com/en/articles/11647753-how-do-usage-and-length-limits-work
[33] Apple Developer Documentation. Device Management. Checked 2026-10-29. https://developer.apple.com/documentation/devicemanagement
[34] Gemini CLI. MCP servers with Gemini CLI. Checked 2026-10-30. https://geminicli.com/docs/tools/mcp-server/
[35] GitHub Docs. Extending GitHub Copilot Chat with Model Context Protocol (MCP) servers. Checked 2026-10-30. https://docs.github.com/en/copilot/how-tos/copilot-in-your-ide/customize-copilot/extend-copilot-with-tools-and-context/extend-copilot-chat-with-mcp
[36] S. R. Howard, A. Ramdas, J. McAuliffe, and J. Sekhon. 2021. Time-uniform, nonparametric, nonasymptotic confidence sequences. The Annals of Statistics 49(2), 1055–1080. https://arxiv.org/abs/1810.08240
[37] MLX. Unified memory: shared CPU and GPU access. Checked 2026-11-03. https://ml-explore.github.io/mlx/build/html/usage/unified_memory.html
[38] Apple. Dynamic Library Programming Topics: introduction. Checked 2026-11-03. https://developer.apple.com/library/archive/documentation/DeveloperTools/Conceptual/DynamicLibraries/000-Introduction/Introduction.html
[39] Apple Platform Security. Managed Device Attestation for Apple devices. Checked 2026-11-03. https://support.apple.com/guide/security/managed-device-attestation-sec8a37b4cb2/web
[40] Apple Platform Deployment. Managed Device Attestation: enrollment, validation and freshness. Checked 2026-11-03. https://support.apple.com/guide/deployment/managed-device-attestation-dep28afbde6a/web
[41] Apple Developer Documentation. SecureEnclave.MLKEM768. macOS 26 availability. Checked 2026-11-03. https://developer.apple.com/documentation/cryptokit/secureenclave/mlkem768
[42] Apple Platform Security. Volume encryption with FileVault in macOS. Checked 2026-11-03. https://support.apple.com/guide/security/volume-encryption-with-filevault-sec4c6dc1b6e/web
[43] Apple. Private Cloud Compute: Hardware Root of Trust. Checked 2026-11-03. https://security.apple.com/documentation/private-cloud-compute/hardwarerootoftrust
[44] Apple. Private Cloud Compute: No Privileged Runtime Access. Checked 2026-11-03. https://security.apple.com/documentation/private-cloud-compute/noprivilegedaccess
[45] Apple. Private Cloud Compute: Software Foundations. Checked 2026-11-03. https://security.apple.com/documentation/private-cloud-compute/softwarefoundations
[46] NVIDIA. Confidential Computing: trust boundaries, attestation and policy-controlled secret release. Checked 2026-11-03. https://docs.nvidia.com/confidential-computing/overview
[47] Scaling up FHE-based Privacy-Preserving ML: Higher Throughput, Longer Inputs for LLama-3-8B. 2026. arXiv:2601.18511. Checked 2026-11-03. https://arxiv.org/abs/2601.18511
[48] MP-SPDZ. Protocols and security models: corruption and collusion assumptions. Checked 2026-11-03. https://mp-spdz.readthedocs.io/en/latest/readme.html
[49] U.S. Department of Health and Human Services, Office for Civil Rights. Guidance Regarding Methods for De-identification of Protected Health Information. Checked 2026-11-04. https://www.hhs.gov/hipaa/for-professionals/special-topics/de-identification/index.html
[50] Information Commissioner's Office. Pseudonymisation guidance. Checked 2026-11-04. https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/data-sharing/anonymisation/pseudonymisation/

The appendices provide supporting detail: work terms, customer capabilities, encryption, billing, comparisons, and privacy and security limits.

A Work terms, roles and delegation

A.1 Work terms

Table 16 defines input, logical work, physical attempts and financial authority. Recovery preserves the original work while recording any new attempt. An identifier alone grants no permission.

Table 16: The paper's work units and permission boundaries in one place. Each definition states what the unit represents and what recovery must preserve.

Term	Meaning and recovery rule
Source group	Related content that must stay together.
Block	Prepared content encrypted together before transfer.
Mac	An eligible computer that runs the complete required local runtime and can process many shards over time.
Exposure allowance	The permitted source-group disclosure to one known owner across their Macs.
Check; acceptance	Assessment under the operation's rules; recording a qualifying result once.
Task	The customer's goal and shared limits. Several runs cannot multiply its budget.
Workflow and run	The connected steps and one execution of them. Recovery preserves the run.
Job or batch	One submitted request or a fixed group of requests. Repeating a submission recovers the original work.
Stage	One operation and its dependencies within the workflow.
Shard	One bounded piece of work. A retry keeps the same responsibility and source coverage.
Attempt, lease, and fence	A Mac's physical try has a lease, its time-bounded authority, and a fence, its attempt version. A replacement advances the fence; resending its result does not.
Home cell and epoch	The workflow's current execution cell and home-assignment version. Approved migration advances the epoch once and blocks stale writers.
Accepted result	A checked result recorded once for its shard. Accepted, partial, and unresolved work remain distinct; a late or repeated submission cannot add another acceptance.
Artifact	A saved input or output file identified by its content fingerprint. A changed file has a different identity.
Delivery	Sending an accepted artifact to its approved destination. Prepared, sent, and acknowledged are separate states; a delivery failure does not undo acceptance.
Scheduled occurrence	One intended time or source event. Repeating its message recovers that run; deliberate reprocessing needs new approval.
Financial hold	Credit or settlement value reserved until its obligation is reconciled; uncertainty cannot release it.
Security hold	A pause on affected work or key release until trusted evidence and current permissions support continuation.
Work credential	Scoped customer access to permitted Compute actions, subject to expiry and account policy.
Provider access	Separate authority to manage permitted provider participation, not customer spending.
Execution permit	Short-lived authority for a provider Mac to perform an assigned execution.

Acceptance, delivery and financial settlement have separate progress records. Unresolved work and its outstanding obligations remain accounted for until reconciliation under the declared completion policy.

A.2 Kit modules

The Kit has four parts that share the execution contract.

Table 17: Four local Kit modules prepare and manage the same authorized handoff. Preparation, connection, submission, and caller-owned checkpoints keep distinct responsibilities under one execution contract.

Module	Responsibility
Setup	Discovers supported clients, previews connection changes, summarizes usage locally and helps assess the decision before approved setup.
Data	Reads the input with the right adapter, keeps related records together, removes what isn't needed, and keeps the identity map local.
Core	Seals the prepared input, gets the quote and approvals, sends the work, and validates results.
Workflows	Preserves caller-owned checkpoints and declared stages while the host advances the plan; it never chooses Macs.

A.3 Home-cell responsibilities

Global services assign each workflow to one home cell; owners control participation.

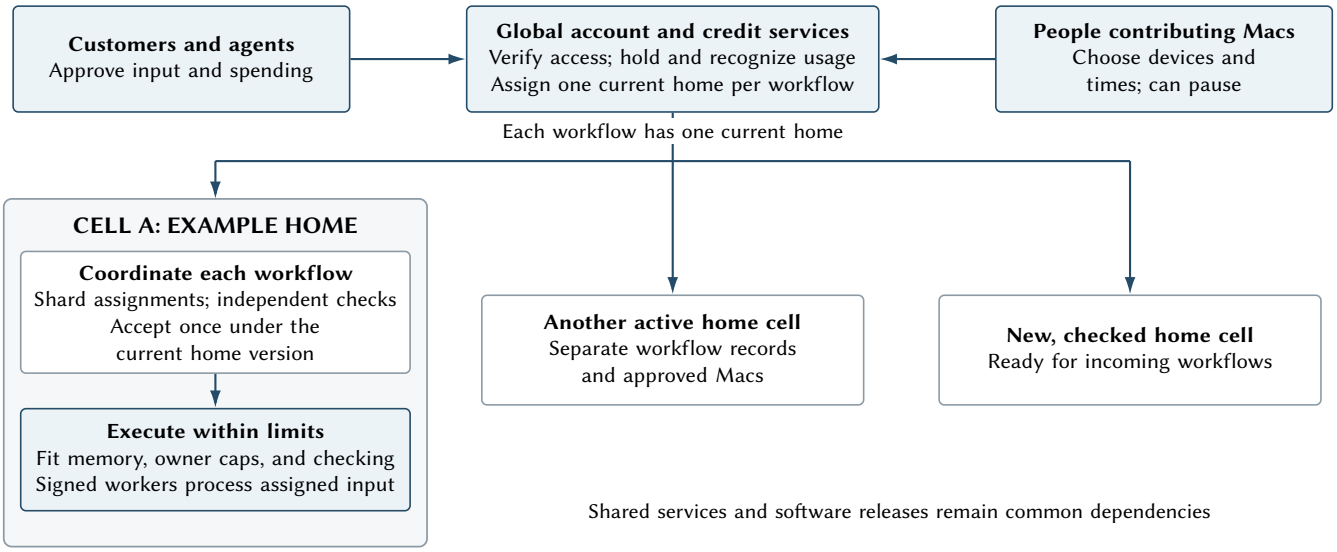


Figure 10: Global services verify account authority, hold credit, and assign a workflow to one active home cell. That cell coordinates its eligible Macs, independent checks, and acceptance records. Separate workflow scopes preserve authority while shared software and global services remain common dependencies.

Each home-cell state change and its outgoing queue signal commit together in one short database transaction. If a queue message is lost, the surviving record replays it under the same work identity. A trusted restart recovers in place; an approved migration advances the workflow's home epoch and fences off the old cell's authority for that workflow.

A.4 Scoped delegation

Section 5 and Figure 3 follow the goal-to-result cycle. Work keys bind API permissions and expiry; shared funded task identity and spend policy govern limits. Labels create no credit. Independently managed helpers use separate keys and connections where hosts support them; provider Macs use execution permits.

Approved source versions or completed snapshots identify paid input. Each scheduled occurrence keeps its original identity for redelivery; deliberate reprocessing needs new approval. Delivery uses approved destinations and resource-scoped credentials. Private-network and loopback endpoints are excluded to prevent internal network probing; loopback refers to the local computer itself.

The host owns context, nesting, checkpoints, time, calls, iterations, concurrency and stopping rules. The Kit has no independent scheduler or retry daemon. Changed destinations, broader disclosure, unresolved submissions, exhausted allowance or security holds pause continuation until reconciliation. The host resumes its saved plan, checking results, coverage, exceptions and full cost. Revocation restricts future actions; accepted work and obligations remain recorded and reconcile.

B Service outputs and checks

This catalog groups 46 customer capability identities into the seven domains introduced in Section 4. Four additional build and test manifests are internal. The domain labels organize the lookup; each operation's approved profile determines the work it can perform.

Each admitted release fixes its reader, representation, runtime, limits, metering and checks. Current discovery determines available profiles. Table 18 fixes the model, tokenizer and prompt-template identities for the MLX path; a different release needs its own admitted profile.

Table 18: The exact model, tokenizer, and template files fixed by the language-model release. Matching these pins keeps execution and input counting consistent across assigned Macs.

Release component	Identity
Language model	`mlx-community/gemma-4-12B-it-4bit`
Model revision	`73bcf09092aa277861d5a191b989b666f7f32e8f`
`tokenizer.json` SHA-256	`cc8d3a0ce36466ccc1278bf987df5f71db1719b9ca6b4118264f45cb627bfe0f`
`tokenizer_config.json` SHA-256	`fc1384a911d2c9860ac07bc3ceafff20bff26695991744b7dbe5e1e4522bfa57`
`chat_template.jinja` SHA-256	`36e3a42e5cf14cd0020e72d92e1fdd9970f59b82170e421f0cbe1bb42bead3f0`

Table 19 is the domain index. Its examples describe useful operations across text, records, visuals, audio, and tensors; a workflow can combine capabilities from several domains.

Table 19: Seven catalog domains connect common tasks with exact capability names. The examples describe operations; the selected profile supplies the execution path, limits, and required checks.

Catalog domain	Identifiers and count	Typical work
Agent and batch work	`agent.*`; 2 capabilities	Apply fixed recipes to bounded items; compare declared alternatives.
Documents and PDFs	`document.*`; 5 capabilities	Extract text, tables, and layout; read scans; transform qualified PDFs.
Text and language	`text.*`; 8 capabilities	Classify, extract entities, deduplicate, identify language or sentiment, translate, rank, and embed.
Speech and audio	`speech.*` and `audio.*`; 9 capabilities	Transcribe, distinguish speaker turns, subtitle, translate or synthesize speech; classify, clean, and normalize audio.
Images and vision	`image.*` and `vision.*`; 16 capabilities	Convert and transform images; return object, region, pose, caption, and comparison observations.
Video	`video.*`; 5 capabilities	Compose or transcode video; return keyframes, motion observations, and scene timelines.
Model verification	`ml.coreml`; 1 capability	Verify a qualified model specification on the CPU.

B.1 Joining shard outputs

A declared reducer combines partial results rather than simply concatenating them. Counts and sums produce exact means; ranking, deduplication, and joins need their declared cross-piece rules. Merge stages preserve source identity and the same disclosure limits.

B.2 Preparation obligations

On your machine, the Kit fixes the source version and checks format and resource limits. Document ID `000123` remains a string, not `123`. The manifest records which inspections ran, distinguishing inspected with no findings from not inspected:

- **Typed records:** JSON, JSONL, CSV, Parquet and Arrow profiles preserve IDs, integer precision, decimal scale, nulls and timestamps. Related records can cross row groups, so required groups stay together before encryption; unsupported nesting or conversions are not silently flattened [12, 13].
- **Documents:** PDF and scans preserve reading order and tables; supported OCR reports regions and uncertainty. Workbooks distinguish formulas from cached values, and slides retain order and permitted notes.
- **Media:** Admitted image, audio and video profiles govern codecs, tracks, language, channels, timebases and coordinates.
- **Archives and repositories:** Admitted members retain stable relative paths and revision identity. An extension alone does not admit a decoder or codec.

B.3 Capability catalog

A workflow may reuse a capability at several stages. Each approved profile fixes the reader, representation, runtime, limits, metering, and checks. Table 20 groups the 46 identities in the same domain order as Table 19. Read each input, returned result, and checking subject together; current discovery supplies precise profile limits. The returned artifact and check records establish the outcome; an output digest alone is incomplete.

Table 20: All 46 capability identities, grouped under seven catalog domains. Read each capability's required input, returned result, and acceptance checks together. The code identifiers remain exact.

Capability identity	Required input	Returned output	Required checks
Agent and batch work · 2			
`agent.bulk_work`	Prepared records or text from documents, pinned diffs, or recorded diagnostics; required context and declared rubric	Checked records or source-linked findings with coverage and exceptions	Source and item identity, required context, rubric, declared format, and coverage; code findings retain revision and line anchors
`agent.candidate_set`	Declared candidate outputs, references, and a fixed rubric	Comparison, selection, disagreement, and abstentions	Candidate and source identity, evidence, and coverage; no automatic patch application
Documents and PDFs · 5	Qualified document representation	Source-linked text and tables	Coverage, citations, and omissions
`document.extract`			
`document.layout`	Qualified document or page representation	Reading-order and layout structure	Page and region identity and structure
`document.ocr`	Qualified scanned pages or images	OCR text and regions	Language, uncertainty, and page correspondence
`document.pdf.extract`	Qualified PDF and parser profile	PDF extraction	Bounded parsing, source pages, and active content
`document.pdf.transform`	Qualified PDF and declared transformation	Transformed PDF	Actual file, reopening, and intended fidelity
Text and language · 8	Permitted bounded text	Closed-label records	Uncertainty, rubric, and item coverage
`text.classify`			
`text.deduplicate`	Text records and equality or similarity rule	Groups and representatives	Stable IDs and declared equality or similarity
`text.embed`	Permitted text and admitted embedding profile	Vectors	Dimensions, dtype, model, and item correspondence
`text.entities`	Permitted bounded text	Entity observations and spans	Source offsets and sensitive-output policy
`text.language`	Permitted bounded text	Language observations	Confidence where defined; not translation support
`text.rerank`	Query and candidates with permitted evidence	Ordered candidates	Frozen rubric, ties, and candidate identities
`text.sentiment`	Permitted bounded text	Sentiment observations	Declared labels and uncertain cases
`text.translate`	Admitted text and locale pair	Aligned translated text	Locales, protected terms, completeness, and meaning
Speech and audio · 9	Qualified audio signal	Timestamped audio observations	Declared classes and source alignment
`audio.classify`			
`audio.cleanup`	Qualified audio signal	Cleaned audio artifact	Decode, duration, and preservation
`audio.normalize`	Qualified audio signal	Normalized audio artifact	Level policy, clipping, and duration
`speech.transcribe`	Admitted speech and language profile	Timed transcript	Language, text coverage, and uncertain spans
`speech.transcribe_native`	Admitted native speech profile	Native transcript	Language, OS profile, and actual text
`speech.diarize`	Qualified speech or recording	Speaker turns	Overlap and uncertainty; not person identity
`speech.subtitle`	Qualified speech audio and subtitle profile	Subtitle artifact	Cue order, timebase, text, and encoding
`speech.translate`	Admitted speech and locale pair	Aligned translated output	Locale pair and declared output modality

Table 20: All 46 capability identities, grouped under seven catalog domains. Read each capability's required input, returned result, and acceptance checks together. The code identifiers remain exact. (continued)

Capability identity	Required input	Returned output	Required checks
`speech.synthesize`	Permitted text, voice, and synthesis profile	Speech audio artifact	Voice permission, language, and decoding
Images and vision · 16 `image.convert`	Qualified image decoder and encoder pair	Target-format image	Decoder, encoder, dimensions, color, and orientation
`image.transform`	Qualified image plus declared transform	Image derivative	Declared transform and source coordinates
`vision.aesthetics`	Qualified images or frames	Defined aesthetic observations	Declared model and rubric; not universal quality
`vision.animal_pose`	Qualified images or frames	Animal pose coordinates	Class, model, coordinate space, and confidence
`vision.background`	Qualified images or frames	Background mask or derivative	Alpha, geometry, and declared interpretation
`vision.barcode`	Qualified images or frames	Decoded barcode and region	Source region; decoded links are not actions
`vision.body_pose`	Qualified images or frames	Body pose coordinates	Coordinate convention and uncertainty
`vision.caption`	Permitted images or frames	Image-grounded caption	Unsupported details and uncertainty
`vision.classify`	Qualified images or frames	Image labels	Declared classes, identity, and uncertainty
`vision.faces`	Permitted images under exact face-data policy	Permitted face observations	Exact data policy; no implicit identification
`vision.hand_pose`	Qualified images or frames	Hand landmarks	Coordinates and reference comparison
`vision.objects`	Qualified images or frames	Objects and regions	Reference comparison, location, and uncertainty
`vision.qa`	Permitted image and bounded question	Image-grounded answer	Bounded question and uncertainty; no new tools
`vision.saliency`	Qualified images or frames	Saliency representation	Spatial correspondence and output meaning
`vision.segment`	Qualified images or frames	Masks or segmentation labels	Geometry, classes, and source correspondence
`vision.similarity`	Qualified image pair or set	Similarity output	Model, metric, and compared identities
Video · 5 `video.compose`	Qualified clips, timeline, and assets	Composed video	Corpus scope, timeline, tracks, and decoding
`video.keyframes`	Qualified video track	Frames with source times	Frame identities, timestamps, and image profile
`video.motion`	Qualified frame sequence	Motion observations	Coordinate convention and timebase
`video.scenes`	Qualified video	Scene timeline	Ordered intervals and source coverage
`video.transcode`	Qualified container and codec tracks	Encoded video and tracks	Decode, duration, and synchronization
Model verification · 1 `ml.coreml`	Qualified model specification	Model-verification output	Admitted verify operation, model, and CPU-only runtime

C Encryption and key access

The Lithi Compute Kit encrypts permitted input before transfer. Storage access and permission to release keys remain separate, as Figure 11 shows.

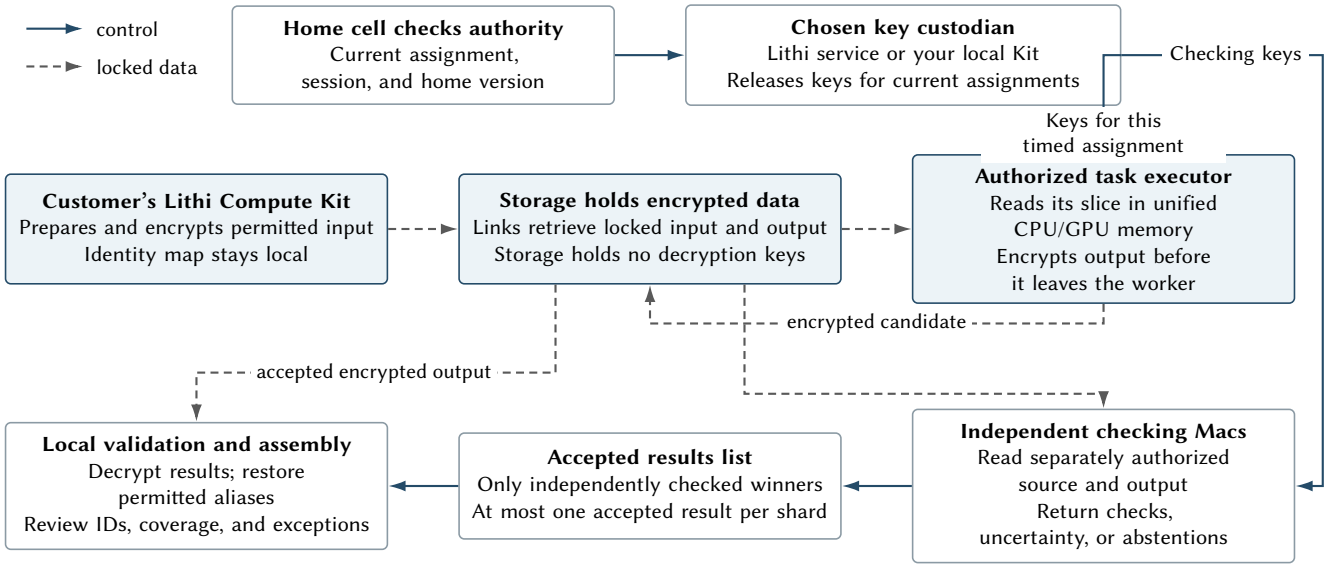


Figure 11: Encrypted blocks and decryption keys follow separate paths. The custodian binds release to the current assignment and worker session; the execution boundary requires ciphertext outside that worker. The assigned slice becomes readable during execution, so process isolation does not establish secrecy from the provider's operating system.

Table 21 names the public cryptographic methods and their roles [7–10].

Table 21: How encrypted blocks and their keys receive different protection. The methods provide payload encryption, recipient key establishment, and purpose-separated key derivation; custody determines who can authorize decryption.

Method	Role
AES-256-GCM	Encrypts each block with a fresh 32-byte key. The 12-byte nonce must not repeat for the same key; the 16-byte authentication tag is a tamper-detection value checked during decryption [7].
HPKE (RFC 9180)	Seals each shard key to its assigned recipient and wraps the ordered block keys beneath it [8].
X-Wing (draft-06)	Establishes shared key material by combining classical X25519 key agreement with ML-KEM-768. ML-KEM-768 provides post-quantum key encapsulation, designed to resist attacks using quantum computers. AES encrypts the payload. The cited revision is an Internet-Draft, not a final RFC [9].
HKDF-SHA256	Derives purpose-specific keys from secret key material. Distinct contextual labels separate the keys used for different cryptographic roles [10].

Key release checks the complete assignment described in Section 10.1, including account, device, worker session, release, model, task, attempt, lease, fence, expiry, and recipient. Current cell identity and epoch remain part of that authority. Execution and checking receive separate authorization after the owner-exposure reservation.

X-Wing defines shared-secret establishment, not hardware key custody. A software CryptoKit private key, including a key reconstructed from an imported seed, does not establish non-exportability. The execution boundary requires fresh task keys inside the executor and excludes usable keys from UI state, transport, and persistent task storage. Secure Enclave ML-KEM-768 offers a separate hardware key mechanism on supported systems [41]; protocol compatibility and lifecycle need their own qualification.

Software expiry checks and revocation prevent future authorized releases. Server-side fencing rejects stale state after snapshot restoration. Neither control can invalidate key material or plaintext a hostile host already copied. Table 6 preserves the distinction between required process handling and host-memory exposure.

D Metering, billing records and collection

Table 22 specifies how signed recipes count each representation under the tariff in Equation (1). Internal ledgers track holds, recognized usage and balances in integer nano-dollars, or billionths of a US dollar. Display rounding changes neither admission nor settlement. Purchases, usage and refunds remain separate records.

Table 22: Five representations share the customer tariff but use different signed counting rules. Text uses the pinned tokenizer; other representations follow their admitted record, patch, duration, or shape recipes.

Representation	Input counted as	Accepted output counted as
Text, code, and Markdown	Released tokenizer over the complete admitted rendering, including required instructions, context, and template	Released tokenizer over accepted text, not compressed byte length
Records and typed data	Canonical typed content under the admitted schema	Canonical returned content, preserving types and identity
Visuals: images, pages, and frames	Patches after the recipe's deterministic resize and resolution rules	Output patches, or text tokens when the result is text
Audio	Decoded duration under the released audio profile	Transcript text or the released audio-output profile
Tensors and vectors	Admitted shape and processing profile	The released tensor or vector-output profile

D.1 Count admitted work

Counts follow the content the operation actually consumes. A document contributes extracted text and page visuals only when those visuals are read; video contributes its admitted frames and audio. Archives and repositories contribute admitted members without a container surcharge. Content already embedded in a prompt is not counted again as a separate input attachment.

Accepted output is the actual returned content under its metering recipe, not a digest or receipt describing it. Every accepted output resolves to a positive count or an explicit deterministic zero-output rule; missing output never counts as zero. Counts bind to the pinned tokenizer, template, canonicalization, and metering profile. Identical canonical input under the same profile gives the same count regardless of executor speed or target earnings. Table 23 distinguishes the three work quantities.

Table 23: Why physical activity and customer billing need separate totals. Platform retries increase resource and electricity records while at-most-once recognition prevents duplicate charges for the same accepted work.

Quantity	Purpose
Physical work	Every attempt, including retries and checking; used for resource and electricity accounting
Accepted work	Logical work behind accepted results
Billable work	Accepted, customer-authorized work after excluding platform duplication

Example: the account balance for Section 13. Funding $100 of credit, excluding tax, and applying Table 11's $0.40 hold leaves $99.60 available. Recognizing the $0.396 charge and closing the remaining obligation without further billable usage releases unused credit, leaving $99.604 available.

At the assumed 40 W role means, multiply 40 watts by 5,000 translation seconds per Mac and divide by 3.6 million. This gives 0.0555556 kWh mean per translating Mac, or 1.111111 kWh across 20 Macs. Checking uses 1,000 seconds per Mac, giving 0.0111111 kWh per Mac and 0.222222 kWh across 20 Macs. The occupied-slot total is 1.333333 kWh.

See the billing guide for purchase, invoice, reload and refund procedures.

D.2 Funding and reconciliation

Funding, accepted usage, reserved credit, refunds and provider payouts retain separate records. Repeated payment events credit the original purchase once; refunds reconcile eligible unused value under the original identity. Uncertain payments, refunds and disputes keep their holds until resolved. Access to retained invoices and statements remains available at zero credit. New reservations require current funding authority, while existing grants retain their finite limits during an outage.

The billing and invoice guide explains credit purchases, paid invoices, prepaid usage statements and refund documents. These documents distinguish prepayment from service use so importing them preserves one expense and payable history.

Automated rule verification. Disposable PostgreSQL, Redis/BullMQ, and mocked-payment suites check that stale attempts cannot replace committed winners; parallel reservations respect owner allowance A; and recognition consumes held credit once without overspending. Expired leases, lost messages, stale epochs, and interrupted payment replies recover from committed records and outboxes. An outbox saves pending messages with related state in one transaction. Recovery is idempotent: repeating a committed action creates no extra acceptance, authority, or charge. These suites verify execution and accounting; separate evaluations assess model quality and fleet performance.

Security acceptance requirements. Accepting an executor boundary requires evidence that forged hardware claims, substituted recipient keys, mismatched worker sessions, and replayed or expired permits are rejected. Crash, reboot, and snapshot tests must preserve fresh-authorization requirements. Synthetic customer markers must be searched for in UI state, broker logs, persistent caches, and other locations outside the intended worker boundary. Hostile payload tests must exercise access to provider files, credentials, local networks, and unrestricted code execution. Test records identify the signed build, device, policy, exercised path, and observed result. Source inspection or a passing software test does not establish confidentiality from a provider-controlled operating system.

E Comparing configurations

A lower bill and a better result are different findings. Compare the full cost of meeting the same task requirements, then assess whether the evidence supports an improvement.

E.1 Compare the full cost

See Section 15 for the full-cost comparison.

E.2 Input-heavy work and Mac earnings

Under Equation (1), 1 million billable input tokens and 1,000 billable output tokens cost $0.0501. If the full charge is eligible and settled, its provider pool is $0.025050 under Equation (4). This task allocation is distinct from the monthly hardware illustrations in Section 14.4.

Section 15 compares input-heavy work on the same full-cost and quality basis, including local preparation.

E.3 Decide whether the outcome improved

Use independent source-family groups with equally weighted scores Y between zero and one. Retries and shards stay within their group. Fix sample sizes, planned comparisons, and one analysis time. Hoeffding's inequality bounds uncertainty for independent outcomes within known limits, without assuming a bell-curve distribution [16]. Equation (9) uses it to compare mean group scores. Add and subtract the sum of both allowances around the difference in means.

$$\widehat{\Delta} = \hat{\mu}_B - \hat{\mu}_A, \qquad \varepsilon_a = \sqrt{\frac{\ln(4K/\alpha)}{2n_a}}, \qquad a \in \{A, B\}. \tag{9}$$

The hatted means are average group scores; n counts independent groups per method, K planned comparisons, and α family-wise interval error. With 1,000 groups per method, $K = 1$ and $\alpha = 0.05$, each allowance is about 4.68 percentage points. The hypothetical three-point gain therefore remains inconclusive. Table 24 shows the worked comparison.

Table 24: A hypothetical comparison between independent groups, including its uncertainty allowance. An observed gain supports improvement only when the declared statistical test and quality conditions are met.

Measure	Value
Independent groups	1,000 per method
Method A (current)	Mean group score 0.96
Method B (proposed)	Mean group score 0.99
Observed gain	+3 percentage points
Uncertainty interval	About −6.36 to +12.36 percentage points
Decision	Inconclusive; improvement not established

Improvement requires the entire interval above a predeclared threshold δ and all quality and safety floors. A non-inferiority comparison asks whether a candidate falls behind the baseline by no more than a predeclared acceptable margin. A cheaper processing path still has to satisfy that quality rule. Count every assigned group and bound missing scores conservatively.

Repeatedly checking a fixed-sample interval and stopping when it looks favorable can exceed its declared error rate. A confidence sequence is an uncertainty bound designed to stay valid across repeated looks under its stated assumptions [36]. Sequential methods, adaptive candidates, and reused groups require a method justified for that sampling design.

E.4 Test and approve network changes

Operational controls react inside an approved policy. The slower experimental loop asks whether a permitted setting should change. Figure 12 follows one candidate through a fixed comparison, evaluation, and separate release approval. Home cells retain acceptance.

The comparison records assigned groups, accepted coverage, quality, deadline performance, full cost, and qualified electricity evidence. Missing outcomes follow the declared analysis rules. Statistical evidence informs the release decision; it grants no deployment authority. After release, monitoring admits qualified findings for the next proposal and retains rollback. Running jobs keep their approved settings.

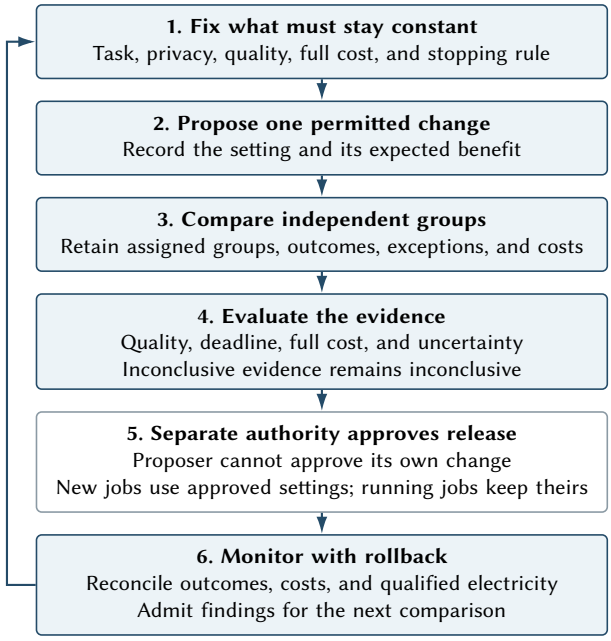


Figure 12: A permitted setting change is compared across independent source-family groups under fixed task, privacy, and quality rules. Statistical evidence informs a separate release authority; running jobs retain their approved settings. Monitoring and rollback produce qualified findings for later proposals.

Table 25 lists permitted comparison dimensions and the invariants that remain fixed across every trial.

Table 25: What an approved comparison may change and what it must preserve. Better packing, timing, or processing choices stay within the fixed privacy, quality, model, resource, and billing rules.

Question	What Lithi may compare	What stays fixed
Useful context	Approved context selection within the operation's limits.	Required evidence, privacy, related content, output structure, and input/output limits.
Structured output	Approved instructions or supported output settings.	Model identity, required output, independent checking, and safety rules.
Packing efficiency	Whole-block packing within admitted bounds.	Source identity, complete coverage, and each owner's exposure allowance.
Faster startup	Reuse of public model files and approved public instructions.	Account separation and permission to reuse private content.
Parallel work and delivery	Supported concurrency and partial delivery.	Resource limits, owner hours, file integrity, and approved destination.
Checking and recovery	Permitted scheduling and retries.	Required checks, independent assessment, one accepted result, and billing rules.
Processing route	Supported native, deterministic, or hybrid routes meeting the same endpoint.	Required meaning, output, coverage, quality, privacy, and acceptance rules.

E.5 Participation scenario

Assume a hypothetical base of 100 million M-series Macs, 15% participation and 10 offered hours per participant per day. For an admitted generative profile, set an assumed output-decoding rate of 40 tokens per second of occupied decoding time. One decoding hour yields 144,000 output tokens; ten full decoding hours yield 1.44 million. These are physical output-decoding units; input prefill and signed media-meter recipes use their own units.

Table 26: A hypothetical participation scenario links offered slots, distributed memory, decoding capacity, and electricity. Every total follows the stated fleet, availability, occupancy, and power assumptions; it is not a measured network result.

Quantity	Assumption or derivation
Participants	15 million from 100 million Macs at 15% participation
Offered capacity	150 million Mac-hours per day; 54.75 billion per year; 6.25 million 24-hour-average offered slot equivalents
Aggregate installed unified memory	360–480 PB, assuming 24–32 GB per participant; distributed across separate Macs, with each shard fitting one Mac
Raw decoding ceiling	21.6 trillion output tokens per day; 7.884 quadrillion per year at 40 tokens per second and full offered-time decoding
Illustrative 5/6 generation allocation	18 trillion output tokens per day; 6.57 quadrillion per year with 5/6 of offered time assigned to generation
Electricity consumed	At an assumed 40 W occupied-time-weighted whole-Mac mean and 100% occupancy of offered hours: 6 GWh per day, 2.19 TWh per year and 250 MW averaged over 24 hours

These figures describe offered slots and decoding-time capacity; the 5/6 fraction borrows the translation example's generation/checking time split. Equation 3 adds input processing and task overhead, while occupancy, decoding duty factor and accepted yield determine completed output; Mac-hours describe internal scenario capacity.

F Privacy and security limits

Table 27 pairs risks with their remaining limits. Both routing modes retain approved data, jurisdiction and disclosed execution boundaries. Geographic dispersion does not establish anonymity; de-identification requires a separate assessment [49, 50].

Table 27: How the design limits each privacy or security risk, alongside the exposure that remains. Read both columns together when deciding what content and custody mode to approve.

Risk	What limits it	What remains
Unsafe input	Typed registered operations, bounded IPC, and narrow worker privileges restrict customer work.	Signatures, symbol checks, and scans cannot rule out every parser defect or make loaded code an isolation boundary.
Access on an executing Mac	Local aliases, owner-wide caps, and separately authorized shard keys limit readable content.	Ordinary execution exposes its slice to the host trust boundary; process isolation and wiping do not exclude a hostile kernel.
Invalid or compromised permission	Account, device, session, recipient, lease, fence, and current cell authority bind key release.	Key possession or a signed hardware flag alone does not establish independent hardware provenance.
Metadata exposure	Limited diagnostics and aggregated reports reduce routine disclosure.	Content size, timing, and permitted context can still reveal information.
External sources and destinations	Approved addresses and limited credentials restrict access.	External services have their own trust and availability limits.
Shared software or services	Reviewed releases and controls outside an affected cell support isolation.	A shared compromised dependency can affect more than one cell.
Restart or snapshot rollback	Fresh session authority and server-side fences prevent local snapshots from reviving access.	Boot posture requires qualified evidence; a new authorization cannot erase previously captured plaintext.

F.1 Scoped pseudonymization and restoration

Aliases use keyed, domain-separated derivation of reviewed entity IDs within tenant, purpose and dataset scope. The private map stays local and version-stable across retries. Equal names do not merge different people; ambiguous mentions need review. Reverse mappings stay outside provider manifests, ordinary agent context and logs. Restoration requires item-scoped permission; wrong-scope requests are refused.