

Lithi: Crowdsourced AI compute, powered by people

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

Abstract

Lithi Compute connects spare capacity on Macs people already own to work submitted by customers' existing AI agents. Locally, the Lithi Compute Kit prepares permitted input, encrypts it before transfer, and submits approved work within a quoted limit. The Lithi Agent Network divides jobs into bounded pieces, limits how much of a source any one owner can read, runs deterministic algorithms, native models, or generative models, and returns checked results with exceptions and cost. The customer's agent retains the goal, context, final assembly, and decision to use each returned result. Signed recipes meter admitted input and accepted output in tokens; generation speed describes one phase of generative work, alongside input processing, checking, and delivery. Half of eligible settled task revenue forms the provider pool, while monthly reports distinguish completed work, payments, qualified electricity measurements or estimates, and estimated emissions. This paper follows an illustrative 1,000-document translation from funding to payout and relates accepted work to Mac-owner earnings, with calculator-based illustrations by hardware configuration and available hours.

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? Owners share spare capacity, your agent sends suitable work, and the network returns checked results.

Lithi works like a smart grid for computing, matching spare capacity on Macs people already own to demand from customers' AI agents. It routes each piece to an eligible Mac, preferring nearby Macs when permitted. Capacity grows as more owners connect.

Through the **Lithi Compute Kit** on your machine, your existing agent delegates bounded work while keeping the overall goal, context, and final review. Agents such as Claude Code, Codex, and Cursor connect over the Model Context Protocol (MCP); applications can use the CLI, SDKs, or REST. The path has four steps:

1. **Define:** Your agent sets the goal, permitted input, and limits.
2. **Prepare and quote:** The Kit prepares and encrypts permitted input, obtains a quote, and submits approved work within the funded limit.
3. **Execute and check:** The Lithi Agent Network runs the pieces and returns accepted results with exceptions.
4. **Review:** Your agent validates, assembles, and uses or rejects each returned result.

A Mac can contribute without generating long-form text: supported operations may extract document structure, transform media, or run a native model to return compact labels and predictions. Generative operations produce translations or structured responses. These paths perform different physical work, so their useful throughput and electricity consumption are measured separately.

Figure 1 shows the exchange. Anyone with a supported Mac can contribute, starting with one personal machine, and choose their hours or pause. Provider Macs receive neither customer account credentials nor local identity maps; task decryption keys follow a separate, assignment-bound release path (Section 10). Monthly reports track completed work, payments, electricity, and estimated emissions. They distinguish direct meter readings from calibrated device profiles so energy and carbon claims rest on explicit evidence.

A running example anchors the paper: translating 1,000 documents into French (Section 13) under illustrative workload, profile, timing, and power assumptions.

The paper makes four contributions: an agent-first interface for bounded work; a per-owner exposure bound; a home-cell model that accepts each result once through recovery; and linked records for accepted work, payments, electricity, and emissions.

2 Who does what

Who decides, and who keeps the record? The customer's agent, global services, the home cell, and Mac owners have different responsibilities.

Your agent owns the goal, permitted source, local preparation, and final use of results. Global services manage access, credit, spending limits, and the workflow's current home. The home cell assigns work, tracks attempts, and records accepted results. Mac owners decide which devices participate and when. Figure 2 shows their relationship.

A **task** is the customer's goal, a **workflow** is the connected stage plan that carries it out, and a **run** is one execution of that workflow. Global services assign each workflow to one **home cell**, tracked by a home-assignment version called an **epoch**, but cannot accept a Mac's result in that cell's place. Storage access does not grant decryption. The customer's agent cannot target a specific Mac or alter an owner's policy. Customer work, provider participation, and platform administration use separate permissions (Table 1; Appendix A).

3 Choose an outcome and a supported path

How do you choose work that Lithi can run? Start with the result you need, then select a supported path that produces it.

For example, ask for French text with the same document IDs and terminology, and name the quality requirements, destination, deadline, and spending limit. Reading a PDF, translating its text, and

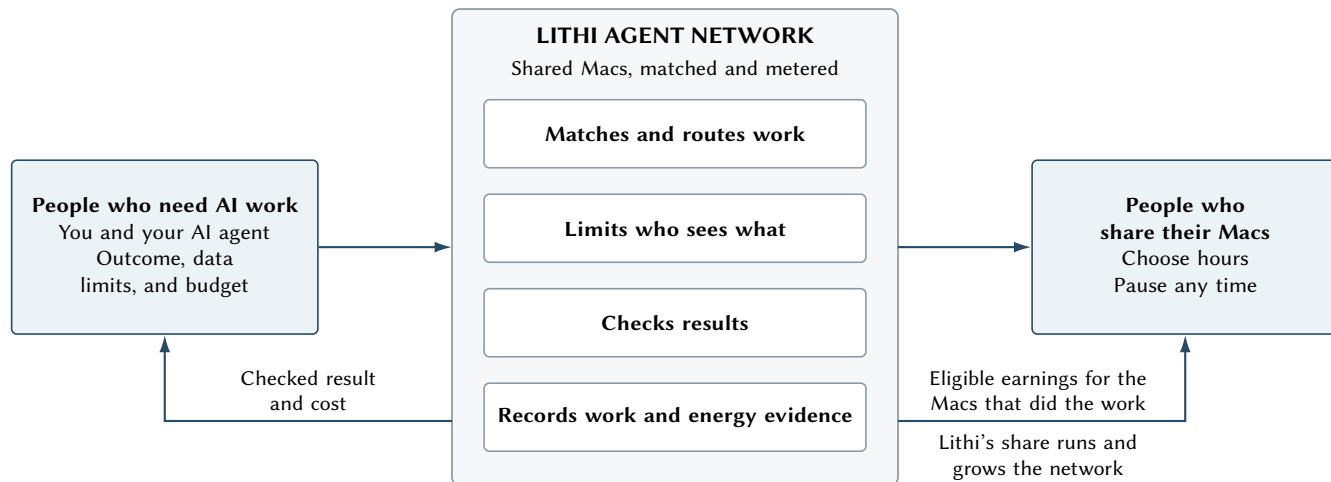


Figure 1: Customers send permitted work, and Mac owners choose when to contribute. The network returns checked results and records eligible contributions.

Table 1: Key terms used in the main explanation.

Term	Meaning
Source group	Related content that must stay together.
Block	Prepared content encrypted together before transfer.
Shard	Whole blocks bundled into one assignable piece of work.
Attempt; lease and fence	One Mac's try at a shard, bounded by a time limit (lease) and attempt generation number (fence).
Home cell; epoch	The cell that owns assignments and accepted results, and its current home-assignment version (epoch).
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.

recreating its layout are separate steps, even when one workflow connects them.

Every service follows three stages (Figure 3). A **reader** converts supported input into one of five representations: text, records, visuals (images, pages, and frames), audio, or tensors (numerical arrays used by models). An **operation** performs the work, and a **writer** produces the requested output format. The Kit verifies formats, model release, limits, and checks during planning; quote and admission recheck metering and eligible Mac capacity before execution. Unsupported paths stop with a clear reason and next action. Table 2 summarizes the seven operation families, which share the five-tool Work interface (Section 4) across the 46 customer capabilities in Appendix B.

The admitted operation determines how an assigned Mac processes input (Table 3). Deterministic processing applies declared rules, native model processing produces typed observations or predictions, and generative processing reads a rendered prompt and

generates a response. Every path carries its own output and checking obligations.

Results carry source IDs, coverage, uncertainty, and abstentions, where a check declines to answer. Speaker labels and face observations do not establish a person's identity. Workflows can filter items, run parallel branches, and join outputs by stable IDs under declared rules. If pieces remain unfinished, the chosen completion policy decides whether to return a partial result, wait, or stop.

Example: reviewing changed code. Your coding agent sends a fixed revision, its diff, and permitted context. Lithi returns source-linked findings and unresolved items. The coding agent checks those references before editing; a changed revision needs a new comparison. Repository credentials, edits, builds, tests, and final acceptance stay with that agent. Findings and patch suggestions do not prove that code passed its tests. Small changes and context that cannot fit may be better kept local.

4 Connect agents, keys and automations

How does your agent connect, and what may it spend? It uses the Kit through one interface, with scoped access and limits you set.

Agents connect through MCP; applications and workflow tools use the CLI, SDKs, or REST. These routes share services, access rules, quotes, and results. Each host keeps its own connection and delegation controls [1–4, 6]. The Kit confirms the goal and account access, then loads the guidance the task needs [5]. Table 4 shows its four parts.

The Work interface has five tools: `lithi.quote` obtains an exact quote; `lithi.run` sends one job; `lithi.batch` sends independent requests; `lithi.status` checks progress; and `lithi.result` retrieves the checked result. Administration and reporting have separate permissions.

Before sending, the agent records the permitted input, required context, checks, destination, completion policy, deadline, and spending ceiling. It saves an identity for each action so a lost response can recover that action. An expired quote needs a fresh quote and approval; adding credit does not revive it.

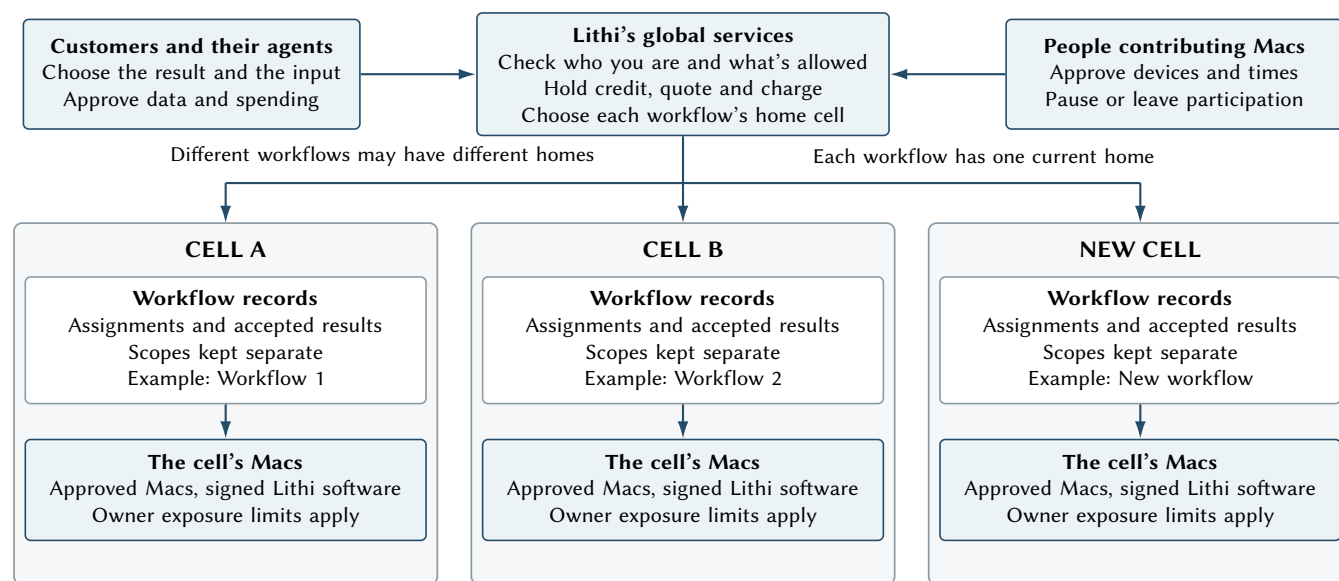


Figure 2: Each workflow has one home cell with separate work records. Shared services and software remain common dependencies.

A scoped **Work key** submits and reads permitted tasks; it cannot create credentials, raise limits, buy credit, save cards, or request refunds. Account administration, key issuance, resources, and limits require separate **Management access**, while purchases and refunds require an authorized billing person. Provider Macs receive short-lived execution permits, never customer Work keys.

When a parent agent delegates to subagents (Figure 4), the host bounds nesting, concurrency, time, calls, and iterations; Lithi runs only the Compute work they submit. Per-child isolation requires separate Work keys and connections. Children sharing a connection share one identity and task budget. A key's ceiling is a spending cap: two \$6 requests cannot both receive full grants under a \$10 remaining ceiling, and adding credit or subagents cannot raise policy limits.

Within an approved policy, agents may quote, submit, check status, and evaluate unattended. A changed destination, broader disclosure, unresolved submission, exhausted allowance, or security hold halts continuation. The parent reconciles every child's results, failures, coverage, exceptions, and Lithi costs alongside its own model and tool expenses.

Integrations prepare approved sources and deliver accepted artifacts to approved destinations. Source versions or completed snapshots identify paid input; a moving directory cannot. Credentials remain limited to the approved resource and expiry, and endpoint restrictions exclude private and loopback addresses. Each scheduled occurrence has a saved identity: redelivery recovers that run, while deliberate reprocessing needs new approval. A due time is not a guaranteed start.

5 Prepare and protect your input

What leaves your machine? The Kit preserves what the task needs, removes what it is not permitted to use, and encrypts approved

input before transfer. Original files and the identity map stay with you.

5.1 Keep meaning

Preparation fixes the source version, checks format and resource limits, and preserves required structure, types, order, and relationships. Document ID 000123 remains a string rather than the number 123, and related items stay together across rows or files (Appendix B). Where a deterministic reader can extract structure directly, it avoids asking a generative model to reconstruct it. Local preparation can remove irrelevant permitted fields and deduplicate repeated instructions, but never discards required evidence to force a task to fit. It remains separate from paid work assigned to provider Macs.

5.2 Limit disclosure

The Kit checks secrets separately from personal information and inspects the composed request, including instructions, schemas, and examples. It returns eligible, review-required, or blocked. Unresolved review stays local, and approval cannot override prohibited disclosure.

Replacing a name with an alias reduces direct identification, but other text may still identify the person. This is pseudonymization, not anonymization. In the translation example, a name and email can be removed while the issue text remains and its identity map stays local.

Anything a tool prints may enter a cloud-hosted agent's context. The Kit therefore returns counts and references by default; a content preview needs separate permission. Remote plaintext conversion must be an approved processing stage.

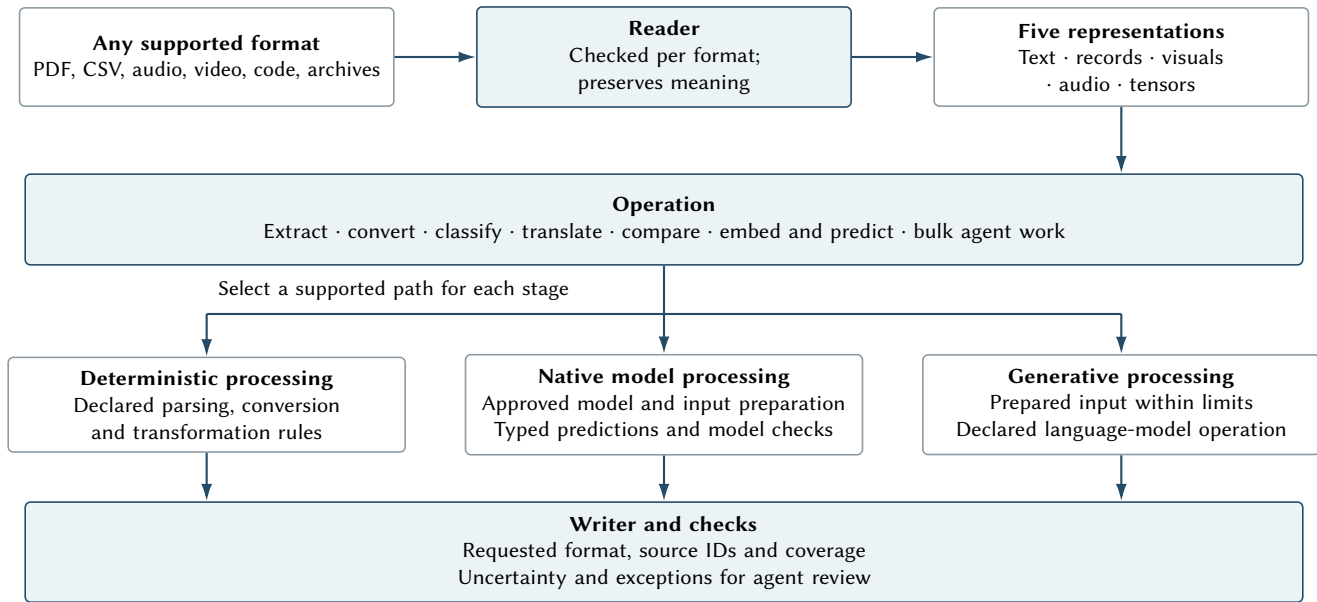


Figure 3: Supported files follow one path: read the input, perform the operation and return the requested output with its checks.

5.3 Approve transfer

Preparation records what was inspected and transformed, explicitly distinguishing “no findings” from “not inspected.” Sealing occurs locally: transfer approval authorizes uploading the exact encrypted artifact to its destination, while quote approval separately authorizes spending.

6 Quote, fund and approve

What does the work cost, and when is credit used? Tokens set the price, an authorized person funds the account, and approved work uses credit within its limits.

Customer Compute costs \$0.05 per million input tokens and \$0.10 per million output tokens. A signed metering recipe fixes each content type’s count; a provider’s own count cannot replace it. Equation (1) gives the charge. The worked task’s 6 million input and 1 million output tokens give a \$0.40 planning total.

$$C_{\text{USD}} = \frac{0.05I + 0.10O}{1,000,000}. \quad (1)$$

Here I and O are admitted input and accepted output token counts under the signed metering recipe (Appendix D), and C is the US-dollar charge. Text input counts the complete rendered prompt, including instructions, context, and required chat-template control tokens [20]; other content follows its released media or data profile. Metering standardizes pricing across representations without implying language-model generation. Pages, images, minutes, and Mac-hours carry no separate fees. Normal platform retries, default checks, delivery, and repair never charge twice for the same logical work. Quotes use planned counts; recognition uses accepted work. 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.

Customers buy \$25 to \$10,000 of credit for one account. Each dollar of service subtotal provides one dollar of entitlement, excluding separately disclosed tax. Credit is not a cash-withdrawal balance. Purchases do not raise account, project, key, or period limits; automatic reload is disabled, so an agent stops for a new human-approved purchase when needed.

Lithi reserves the quote’s maximum from existing credit, consumes credit for accepted usage, and releases only the amount shown to be unused. Uncertain obligations remain reserved. Stages and subagents share the original maximum. Table 5 separates collection, admission, acceptance, and recognition.

Example: funding and settling credit. A team buys \$100 of credit, excluding tax, producing one card charge and one paid invoice. Reserving \$0.40 for the task leaves \$99.60 available. When 990 documents are accepted, Lithi consumes \$0.396 and keeps \$0.004 reserved for the ten unresolved documents. If reconciliation later closes them without further billable usage, releasing \$0.004 makes \$99.604 available.

Spending uses exact amounts; display rounding never changes a charge or limit. Cancellation requests, quote expiry, and queue completion cannot close an in-flight obligation by themselves. A funding outage stops new reservations while existing grants retain their finite limits. Identical prepared input under the same recipe

Table 2: Seven operation families over five representations. Each returns its actual output and the checks that apply to it.

Operation family	What it does	Works on
Extract	Pull text, fields, tables, frames, or scenes from a source	text, visuals, audio
Convert and clean	Change format, resolution, or quality without changing meaning	visuals, audio
Classify and detect	Label items or find things in them	any
Translate, transcribe, and describe	Turn one language or medium into another	text, audio, visuals
Compare, rank, and evaluate	Score, order, or choose between candidates under a fixed rubric	any
Embed and predict	Produce vectors or model predictions	text, tensors
Bulk agent work	Run many independent items, including code review, and return findings	text, records

Table 3: Processing paths and what establishes a useful result.

Processing path	Example work	What establishes a useful result
Deterministic	Supported parsing, conversion, or transformation	Correct output, preserved meaning where required, and complete coverage
Native model	Supported speech, vision, or classification	Qualified model output, uncertainty, and task-specific checks
Generative	Translation, assessment, or structured generation	Required meaning, evidence, structure, and output coverage

Table 4: The four parts of the Kit, all using the same execution contract.

Module	Responsibility
Setup	Connects the agent to the selected account and supported interface, then confirms the 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	Chains local steps and Compute steps with checkpoints the agent owns; it never chooses Macs.

Table 5: Separate authorities, one funded work identity.

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.

gives identical counts. Excluded content is not admitted input; accepted output has a positive count or an explicit zero-output rule.

7 Home cells and growth

Where does a workflow live, and how does capacity grow? Each workflow has one home cell, which assigns its work and records accepted results.

A cell serves many workflows while keeping their scopes and work records separate. Lithi records each workflow's current home and checks the cell's permission to act. Capacity grows by adding eligible Macs or opening another cell for new work (Figure 5).

Growth must fit memory, transfer, owner allowances, and checking capacity. Spare processors cannot replace a missing checker or expired permission. Region constrains processing location; a customer's locality preference guides selection among eligible Macs.

If demand exceeds capacity, Lithi can pause new work or prepare another cell. Missing measurements do not establish spare capacity. Running workflows keep their home during ordinary growth; an approved recovery can move them. Before a cell retires, its unfinished work, reserved credit, and retained files are resolved.

Within a home cell, each 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. When capacity, checking, or permissions fall short, Lithi pauses new admission and reports status and next actions without exposing private content or device locations. Suspected compromise follows Section 12.

8 Split work without losing meaning or privacy

What is a shard? Sharding divides a large job into pieces that Lithi can assign separately. Your agent supplies the outcome, input, and limits; Lithi works out the pieces.

The Kit keeps related input together and encrypts it into blocks on your machine. Lithi's planner packs whole blocks into bounded **shards**, using sizes, dependencies, and declared limits without opening the content. It assigns each shard to an eligible Mac after checks against the owner's exposure allowance.

In the translation example, each document stays whole and forms one shard: 1,000 documents give 1,000 shards. Smaller items can share a shard when their combined input, required context, and planned output fit the operation's profile. A Mac can process several

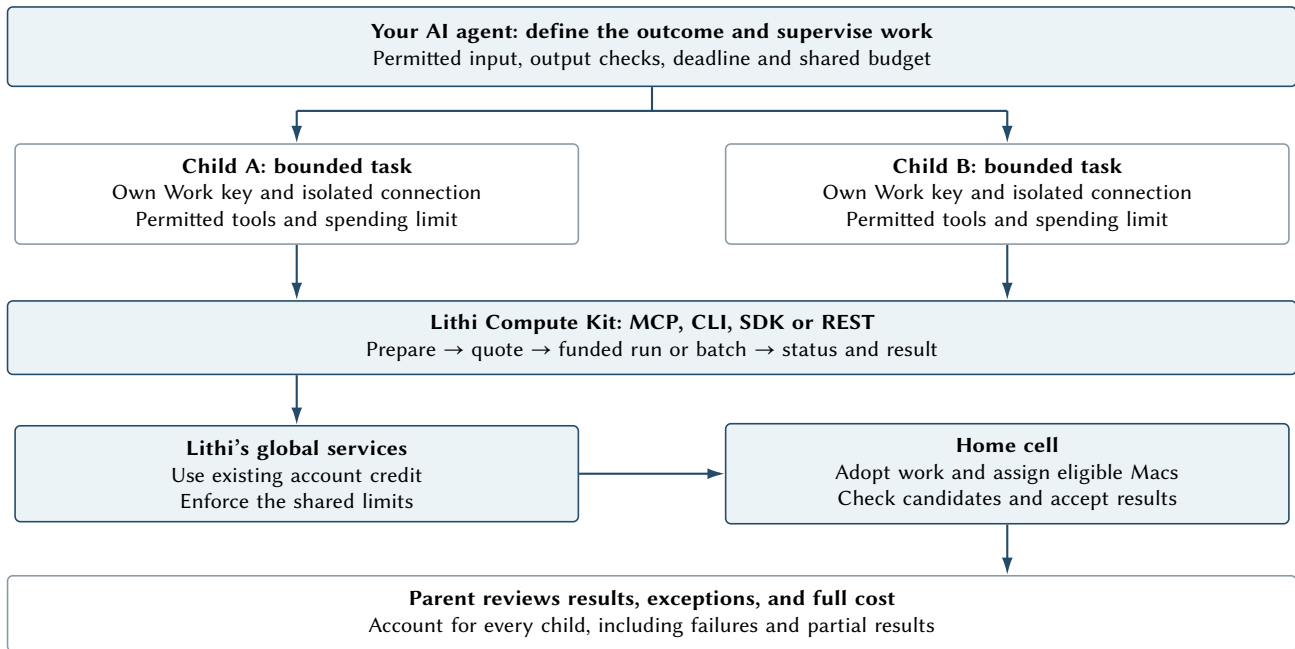


Figure 4: Your agent can delegate to subagents while sharing one task budget. The Kit connects their permitted work to the network.

shards over time. If an attempt fails, another eligible Mac can retry the same shard without changing its source identity.

Lithi checks the source, dependencies, and completion rules before adopting the plan. It rejects cycles, missing pieces, and pieces assigned twice. Checking only the total would miss a missing piece and a duplicate that cancel numerically, so coverage checks inspect the pieces themselves.

The running example assumes a per-call model profile of at most 9,600 input tokens, 1,200 output tokens, and 10,800 combined. Input and output ceilings are checked independently and against the combined cap; the output allowance is not deducted a second time from input. A workflow can process larger sources across many bounded shards. Native and deterministic operations enforce their own admitted format, resource, and metering profiles; current discovery supplies actual limits.

Packing accounts for both input and output. Many short classifications can share a shard because their responses are compact; translation packs fewer items to leave room for full output. Fixed instructions and required context count in each rendered call where they appear. The compiler preserves semantic groups, returning a split or refusal rather than truncating required content when an item cannot fit.

Reducers combine pieces into an answer under the same fit and disclosure rules, using several bounded stages when needed. Exact means combine counts and sums. If one piece has one value summing to 10 and another has three summing to 270, the whole mean is $(10 + 270) / (1 + 3) = 70$. Averaging their means, 10 and 90, wrongly gives 50. Ranking, deduplication, and joins likewise need declared cross-piece rules.

An owner can control several Macs, so the exposure allowance applies across all of them. At 5% over 1,000 documents, one owner may read at most 50, however many Macs they run. 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. Equation (2) counts distinct groups in either set, counting overlap once.

$$|E \cup R| \leq A. \quad (2)$$

Derivatives, overlap, checking, and reducer input refer to the same source population. Repacking, key changes, and migration cannot reset the history; reservations precede key release. Production alone requires at least $\lceil |G|/A \rceil$ distinct owners; full-source checking by an independent owner requires at least $\lceil 2|G|/A \rceil$. Retries and resource limits can require more. This bound constrains known owners, not collusion or false identities (Section 18). An indivisible group exceeding processing or disclosure limits returns no fit unless an alternative is approved.

9 Planning and completion forecasts

When will the work finish? Lithi estimates when the required end-point will be reached: accepted results, or confirmed delivery when required. Waiting, execution, checking, and delivery all contribute.

Figure 6 combines the quoted, then approved plan with eligible capacity and timing evidence from comparable runs. A step waits until its input is ready, a resource slot is free, and every prerequisite has finished. Parallel branches overlap rather than adding their durations.

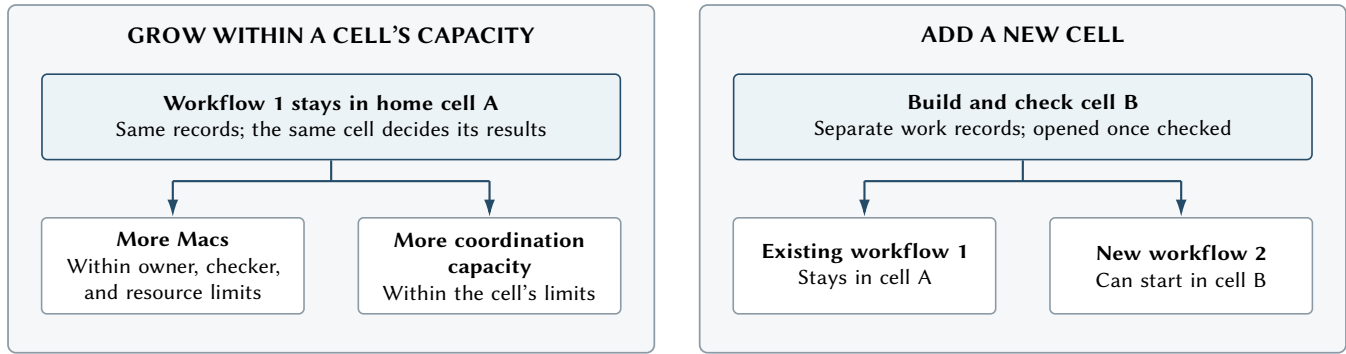


Figure 5: Add capacity within a cell's limits, then open another cell for new work. Existing workflows keep their home.

For a generative step v , occupied duration d includes processing I rendered input tokens at the input-processing rate, generating O tokens at the generation rate, and other occupied time such as transfer, startup, and local validation [21]. These are per-attempt physical quantities, not extra billable tokens. Independent checking remains a separate step. Equation (3) connects that duration to start s and finish f , given input ready time r , the next resource slot a , and prerequisite set $P(v)$; omit the prerequisite term when the set is empty.

$$\begin{aligned}
 d_v &\approx \frac{I_v}{r_{\text{in}}} + \frac{O_v}{r_{\text{out}}} + d_{\text{overhead},v}, \\
 s_v &\geq \max\left(r_v, a_v, \max_{u \in P(v)} f_u\right), \\
 f_v &= s_v + d_v.
 \end{aligned} \tag{3}$$

Generation speed describes only the generation term. It does not measure file parsing, media decoding, tokenizer encoding, or native model execution. Native operations use measured durations over their admitted workloads, such as records or frames, alongside token counts. Whole-job forecasts retain the separate waiting, checking, and delivery stages.

Timing records distinguish waiting, transfer, claim, startup, execution, checking, finalization, and delivery. A p95 time covers 95% of runs in its declared population; adding stage p95 values does not establish a job p95 [15]. A cautious estimate states its method, coverage, and freshness. Sparse evidence can mean no justified estimate.

The customer sees the estimate, deadline, latest start, and validity. Using the illustrative 20-Mac schedule in Table 9, 105 minutes of remaining work and a ten-minute reserve give a latest planned start of 07:05 for a 09:00 deadline. Forecast updates preserve earlier records and cannot relax the deadline, budget, or checks.

10 Run on shared Macs and accept results

How does a piece reach a Mac, and when does it count? The home cell offers it, the Mac claims it, keys release for that assignment, and checks determine acceptance.

The customer approves the task, the owner permits participation, and the home cell authorizes each attempt. Eligibility covers permission, identity, jurisdiction, encryption, runtime, resources, file

access, deadline, exposure, and checking capacity. Ranking follows eligibility. Locality is an opt-in preference among suitable Macs, not a distance guarantee.

Once the cell records an assignment and reserves the owner's exposure allowance, the Mac verifies its local policy, resources, current permission, and lease before claiming work (Figure 7). Standing owner policy permits claims without manual approval of each shard; being online alone grants no execution permission. Encrypted input, intermediate, and final objects live in cell-scoped storage and are fetched through short-lived links. Anyone holding a link can use it until expiry, so links carry no decryption keys. The Mac receives its shard keys and a signed execution permit through a separate path bound to that assignment and attempt.

10.1 Choose who holds the keys

In **managed custody**, Lithi's key service holds and releases keys, so Lithi could technically open the input. In **customer-held custody**, the Kit on your machine releases them; Lithi's service cannot open the input, and the Kit stays online while work runs. Storage access and key release are separate permissions. Table 6 shows the access limits, and Appendix C explains the cryptographic methods and key flow.

10.2 Run within the owner's limits

The app checks owner permission, power, activity, temperature, memory, storage, and active work. Encrypted connections and restricted workers limit what each process can reach [11]. Task content cannot grant shell, mail, payment, or owner-account access. The job specifies its model, tokenizer, instructions, and limits; a mismatched Mac refuses it rather than switching models.

Public model files and approved public instructions can be reused to shorten startup. Private input, working memory, and output remain separate for each account and run. Inputs, outputs, temporary files, keys, and records follow their stated retention. Revocation stops future authorized actions but cannot recall disclosed plaintext, and restoring a backup cannot revive revoked access.

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

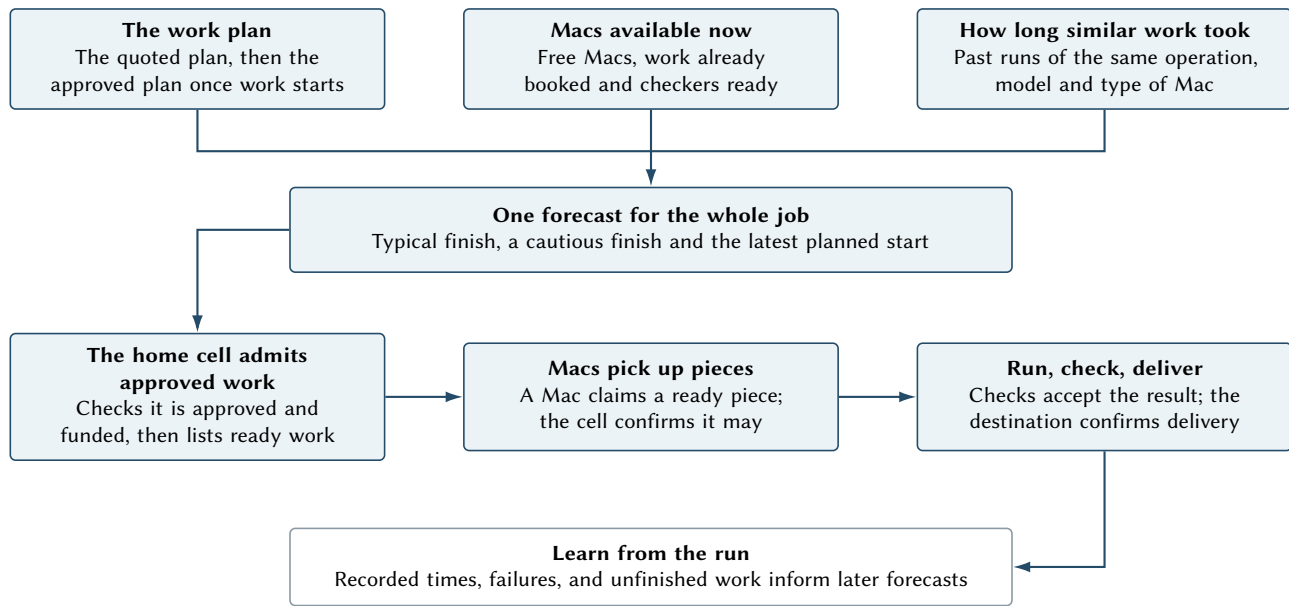


Figure 6: Whole-job forecasts combine the work plan, eligible capacity, and timing evidence. They remain estimates.

Table 6: Authorized content access while work runs, with its important limits.

Who	Authorized access and limits
The customer's machine (Kit)	Everything permitted. The original files and identity map never leave.
Storage and the network	Encrypted blocks, their sizes and timing; no content
Lithi's planner and home cell	Block references, sizes, limits, and check results; no content
Lithi's key service	In managed custody, it can open input keys; in customer-held custody, it cannot
The assigned Mac	Receives its assigned content. A privileged host can inspect that plaintext and may retain information already disclosed.
Other Macs and owners	Receive no keys for unassigned content through the normal release path. The cap applies to known owners, not collusion or compromised shared controls.

IDs; semantic checks assess meaning and required quality. 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. Passing the configured checks permits acceptance, not certainty that an AI answer is mistake-free.

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.

Before restoring permitted local fields, output validation checks integrity, IDs, schema, and representation. It rejects traversal, unexpected files, and unapproved overwrites. Returned scripts, macros, and formulas gain no execution permission, and your agent validates delivered files before using them.

11 Recover the step that failed

What happens when a step fails? Recovery repeats that step under the original job's limits.

Transport losses replay the original request identity; failed execution assigns a new fenced attempt for the same shard after rechecking permissions, exposure, retry limits, deadline, and budget (Table 7). Replacements cannot widen access, alter destinations, or lower checks. Stale attempts cannot overwrite an accepted result, and normal retries or artifact redelivery add no customer charge; speculative concurrent copies require explicit permission. Zero credit does not block authorized access to retained results, link renewal, or safe cancellation. An uncertain action keeps its reserved credit and capacity until reconciliation against its saved record; changing the original request's meaning remains a conflict.

12 Contain a breach

What if a cell or Mac is compromised? Lithi blocks affected assignments and key releases, checks trusted records, and resumes unfinished work only after recovery and permissions are confirmed.

Separate cell records and permissions allow unaffected cells to continue when shared controls remain trusted. Compromised shared services or software may require a wider shutdown. Lithi can hold a suspect result, exclude one Mac, or pause a cell; a Mac going offline is not itself evidence of attack.

Figure 8 shows the response. Controls outside the suspect cell enforce the pause; the old cell remains blocked. A result marked

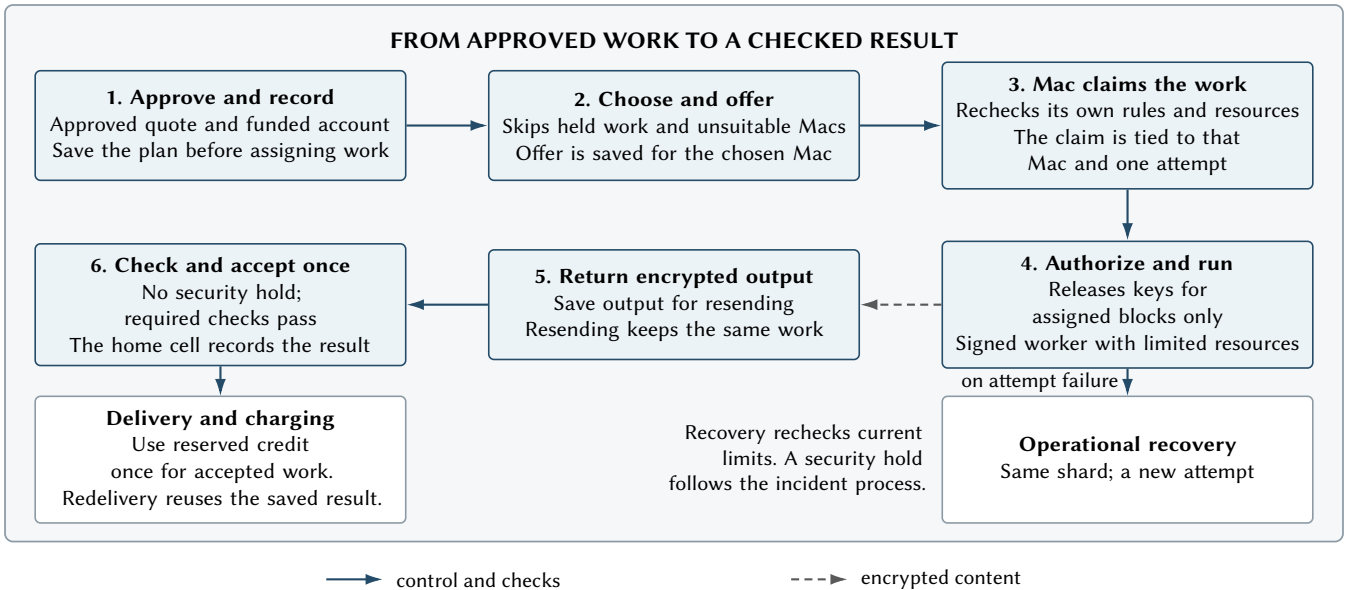


Figure 7: The Mac claims approved work, returns an encrypted candidate and receives acceptance only after the required checks. Delivery and charging follow separately.

Table 7: Recovery changes only the state that actually failed.

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.

accepted in suspect records needs independent support. Reconciliation preserves accepted results, unfinished obligations, reserved credit, and each owner’s exposure history before the work moves.

Table 8 gives the customer consequences. **Recovery after 700 accepted results** is a separate illustration: trusted records support 700 translations and leave 300 unfinished. Lithi keeps the 700 and runs only the remaining 300 in a clean cell, reviewing unsupported results separately.

Replacing a cell does not repair a compromised Mac. It stays excluded until checked and repaired. Already disclosed content cannot be recalled, and an unreachable Mac remains unconfirmed. Appendix F summarizes the remaining risks.

Table 8: What happens to your work during recovery?

Part of the work	What happens
Workflows in other cells	Unaffected cells can keep running when shared controls remain trusted
Results already accepted	Kept when independent, trusted records support them
Unfinished work	Resumed only after safe recovery and current permissions are confirmed
Customer charges	Accepted work is not charged again; uncertain charges stay held for reconciliation
Owners’ exposure history	Preserved across recovery; it never resets
Content already disclosed	Cannot be recalled; implicated Macs are excluded pending separate repair

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

What does one complete path look like? Your agent prepares 1,000 document translations, Lithi assigns their shards, and the agent assembles checked outputs while accounting for every exception.

The following are illustrative assumptions for this task:

- **Workload:** 1,000 documents of about 600 words each, translated into French while preserving terminology, structure, and source identity.
- **Profile and tokens:** one document per shard, with 6,000 input tokens including instructions and context, and 1,000 output tokens. The assumed profile permits 9,600 input, 1,200 output, and 10,800 combined; actual operation limits come from discovery.
- **Time and capacity:** 100 seconds for translation and 20 seconds for independent checking per document, with eligible

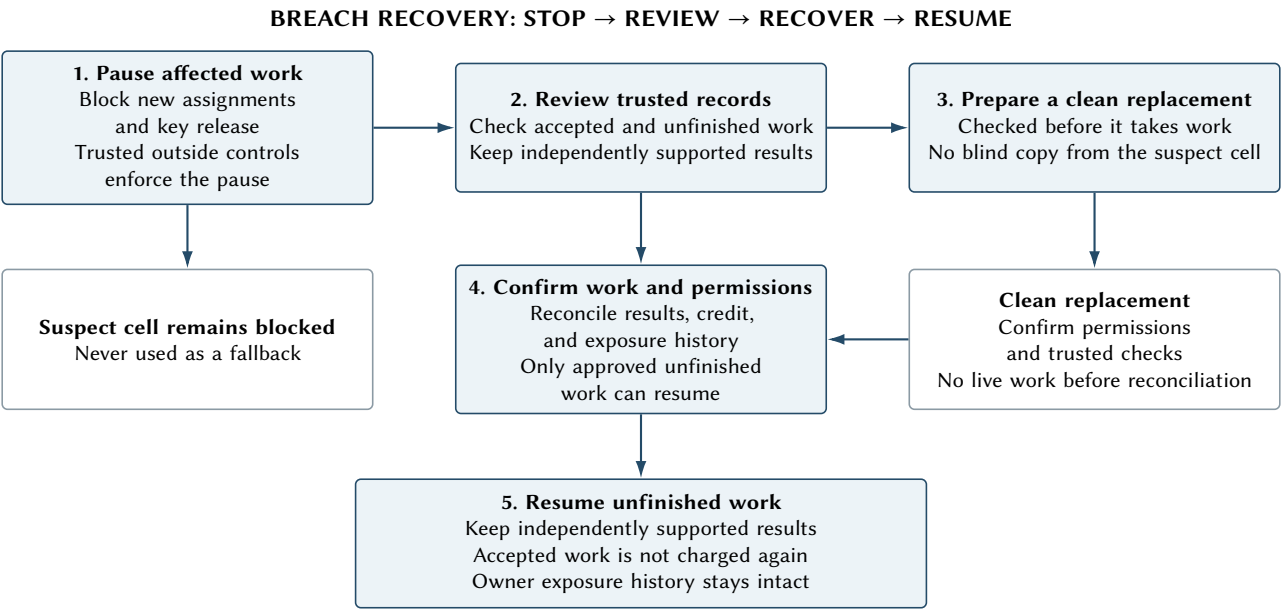


Figure 8: Affected work pauses while trusted records and a clean replacement are checked. Only approved unfinished work resumes.

resources ready. Preparation and delivery add five assumed minutes.

- **Electricity:** 40 W per working Mac during translation and checking.
- **Allocation:** the accepted charge is eligible for revenue sharing, with an assumed 80/20 split between translation and checking roles.

The customer’s parent agent fixes the source, destination, deadline, budget, and completion checks, optionally delegating submission and review to scoped subagents. For example, given “Restart the Lithi desktop app before opening case {case_id}.” and a glossary mapping “case” to a support ticket, the formal French output is “Redémarrez l’application de bureau Lithi avant d’ouvrir le ticket {case_id}.” Verification confirms that meaning, terminology, document ID, and placeholder are preserved. The instruction is translated, not executed.

Locally, the Kit prepares and seals the 1,000 documents into 1,000 shards and obtains the \$0.40 quote (Equation (1)). Once funded, the home cell assigns shards to eligible Macs (Figure 9). A 5% owner exposure allowance permits 50 documents per owner: translation requires at least 20 distinct owners, and independent full-source checking raises the minimum to 40. The checker reads the permitted source, glossary, candidate, and rubric within its profile. Default checking is included in the quote.

At 120 combined seconds per document, the task totals 2,000 Mac-minutes, or 33 h 20 min of serial work. The exposure allowance precludes running it on a single owner’s Mac. Table 9 illustrates two-stage turnaround across 20, 50, and 100 concurrent Macs. With 20 Macs, 20 owners translate 50 documents each and 20 different owners check them; with 50 or 100 Macs, owners can translate and check disjoint sets within their allowance. Preparation and delivery add five assumed minutes; transfer delays, switching, and retries can increase elapsed time.

Table 9: Illustrative translation and checking times for 2,000 Mac-minutes.

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

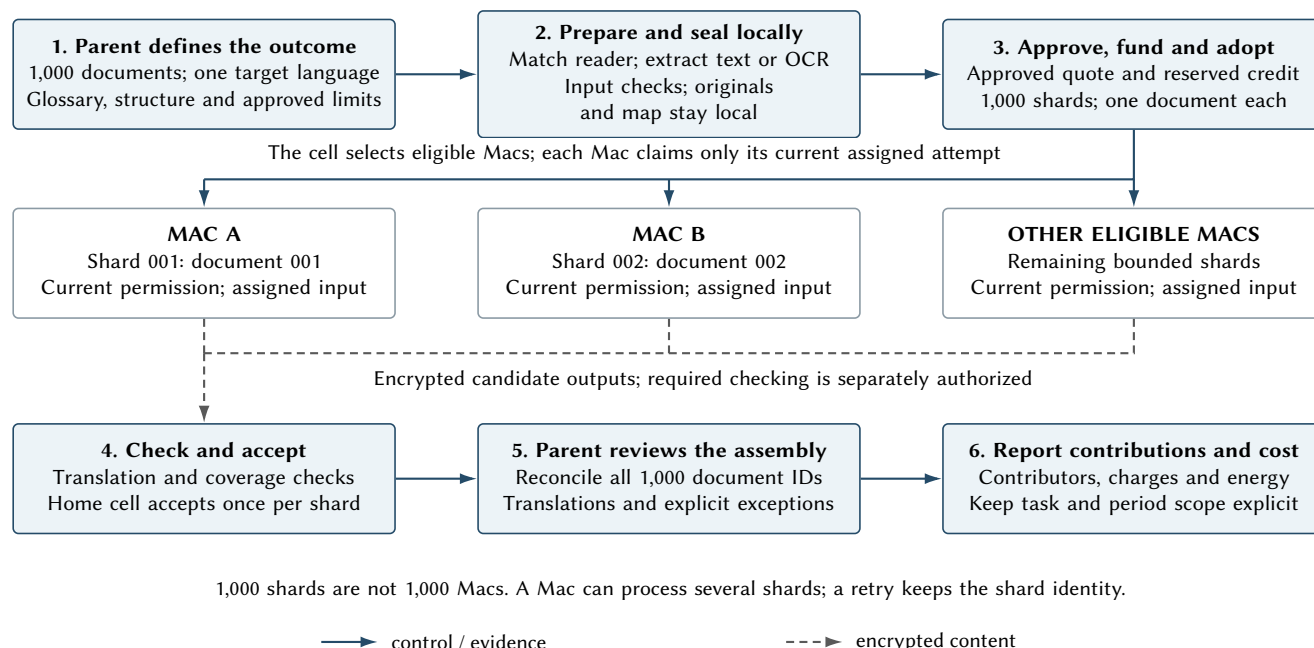


Figure 9: Illustrative translation flow: one document per shard, checked results and explicit exceptions. Shards and participating Macs are different counts.

Partial delivery. 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.

Table 10: Illustrative partial delivery: reserved credit, accepted usage and contributions.

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/checking electricity	About 1.3333 kWh at an assumed 40 W per working Mac; other activity is not estimated here

Accepted translation and checking work creates the provider pool shown in Table 10. The signed allocation policy divides that pool among the Macs that contributed. A Mac's monthly earnings reflect its accepted contributions across many jobs. The Connect

earnings calculator provides a separate planning illustration based on the Mac, chip, memory and available hours, with estimated electricity included (Appendix E.2).

Allocations are not cash payouts or profit, and payment processing fees apply to the credit purchase, not each task. The parent reconciles task-level files, coverage, exceptions, delivery, and full cost. An indivisible contract clause exceeding the assumed 9,600-token input limit returns *no fit* rather than losing meaning through truncation.

14 The people sharing their Macs

What does an owner agree to, and what do they earn? Owners choose their Macs and hours, review the permissions, and consent to participation. Only qualified Macs receive assignments.

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, pause, or withdraw. Sleep, logout, connectivity, foreground use, and heat affect availability; overnight permission cannot make a sleeping Mac work. Pausing stops new assignments while existing attempts follow their completion or cancellation rules. Signed updates preserve consent. Withdrawal reconciles remaining work and removes only Lithi-owned components.

Half of eligible settled task revenue forms the provider pool, which the signed allocation policy divides among the Macs whose execution and checking contributed to accepted work. An individual

Mac receives its weighted share of that pool rather than half of the whole task charge. Lithi uses its share for acquisition, support, coordination, checking, and payment operations; owners do not find or bill customers.

An owner who invites another Mac earns 10% of that Mac's eligible earnings, capped at 5% of task revenue and paid from Lithi's share. The invited owner keeps their full allocation.

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.

Equation (4) distinguishes the provider pool from an individual Mac's allocated earnings.

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

Here D is eligible settled distributable revenue for task j , P is its provider pool, and w is Mac m 's allocation weight. G is gross allocated earnings. 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.

What different Macs could earn. The Connect calculator combines your Mac's family, chip and memory with the hours you make it available. It scales the hardware planning figure to those hours and includes estimated electricity to show an illustrative monthly amount before tax. You can adjust the hours and electricity price for your circumstances [22].

Table 18 gives representative configurations under the calculator's stated assumptions. Actual earnings come from accepted work your Mac helps complete, once the customer has paid. Available work varies. Hardware planning figures already represent owner earnings assumptions; the provider-pool percentage is not deducted again.

Chip capability and available memory help determine which workloads fit a Mac's execution profile. A Mac can contribute by processing documents, running supported native models or generating responses. Input-heavy work can return a compact result after processing substantial permitted input, so output generation speed describes only part of its contribution.

15 Compare results and improve the network

How do you compare methods, and how does Lithi improve? Compare outcomes against the same completion and quality requirements using full task cost. Test network changes before approving them.

A configuration includes the agent model, instructions, guidance, tools, task policy, and stopping rules. Use the same population, quality requirement, deadline, and endpoint for each comparison. Equation (5) adds its costs; all terms refer to configuration c over that same task.

$$\begin{aligned} C_c &= C_{\text{host}} + C_{\text{Lithi}} + C_{\text{tools}} \\ &\quad + C_{\text{local/review}} + C_{\text{setup/evaluation}}. \end{aligned} \quad (5)$$

Host cost covers the agent's model calls. Tool cost covers external tools, local/review cost covers local work and review, and setup/evaluation covers preparing and assessing the comparison. Keep those categories separate. Include unsuccessful calls and trials, and count recognized Lithi consumption rather than unspent credit purchases.

With S counting outcomes that meet the endpoint, Equation (6) gives the value ratios.

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

Unknown costs stay unknown, and subscription use is not automatically cash saved. Appendix E illustrates the accounting and explains when an observed difference supports an improvement claim. Lower cost alone cannot establish preserved quality; an allowed partial result remains partial. Exported run and cost records show Lithi's contribution, not an isolated causal effect.

Better local preparation can reduce unnecessary tokens, repeated calls, and avoidable failures, lowering both the customer's bill and the single-task provider pool. Owner value improves only when higher accepted throughput, better electricity efficiency, and sufficient demand outweigh that per-task reduction. Inflating token counts without increasing required, accepted work creates no useful value.

Lithi's improvement engine proposes a small change to a permitted setting. The experiment service compares current and proposed settings and evaluates the outcomes. A separate release authority approves changes; the proposer cannot approve its own. Home cells keep execution and acceptance, and existing jobs retain their original settings (Figure 12).

Each comparison records its settings, assigned groups, scope, and budget before work starts. Outcomes include failures, cancellations, and unresolved work; retries do not create independent samples. Consent, identity, eligibility, security, retention, required checks, and payouts remain constraints. A comparison grants no training rights and cannot create credit or change the tariff. Inconclusive evidence stays inconclusive, and private customer recipes need separate permission for wider reuse.

16 Report contribution, electricity and emissions

What can a report say about work, electricity and emissions? It identifies completed work, contributors, electricity, estimated emissions, and payments, each with its own scope and coverage.

Electricity accounting includes execution, checking, failed attempts, and other activity within the declared reporting boundary. Qualified meters or calibrated profiles supply labeled measurements or estimates. CPU time, temperature, and charger rating do not establish whole-Mac energy. Device load and grid input are different boundaries; adding both double counts.

Equation (7) sums power P in watts over interval durations in seconds. Each interval's energy is its power multiplied by its duration, converted to kWh. Missing intervals stay missing. The

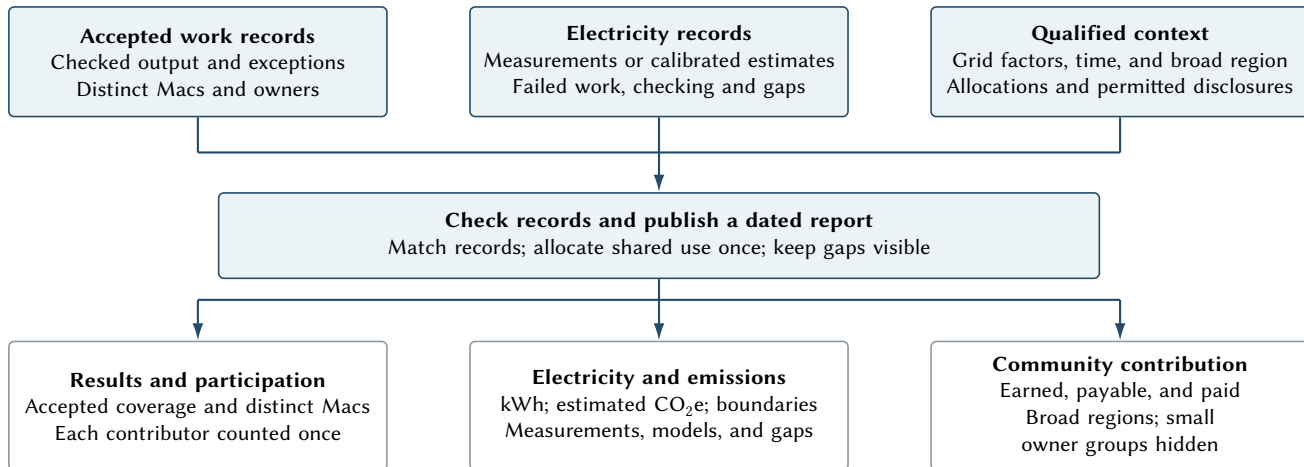


Figure 10: Reports keep completed work, electricity, estimated emissions and provider payments separate, with their own coverage.

worked task's 1.33 kWh covers its assumed execution and checking only.

$$E_{\text{kWh}} = \frac{\sum_k P_k \Delta t_k}{3,600,000}. \quad (7)$$

Electricity efficiency compares accepted results across routes meeting the same task and quality requirements. A path using less electricity while leaving more work unresolved may be worse overall. Comparisons therefore retain accepted coverage, failed-attempt energy, missing intervals, and the measurement boundary. Input-processing and generation rates help explain performance; neither replaces accepted work as the denominator.

Operational emissions apply each interval's regional, hourly grid factor I , in grams of CO₂e per kWh (Equation (8)). The report states the factor's method, version, uncertainty, and coverage.

$$M_{\text{gCO}_2\text{e}} = \sum_k E_{\text{kWh},k} I_k. \quad (8)$$

Software Carbon Intensity (SCI) also accounts for embodied emissions from hardware across a declared system boundary [14]. Operational electricity alone is not a full footprint. Reusing hardware or preferring nearby Macs cannot by itself establish avoided emissions.

Reports identify recorded earnings and qualified electricity over the same scope and period. The calculator's planning assumptions remain separate from calibrated measurements and modeled operational estimates. Device load and wall input represent different boundaries and are never summed together.

Figure 10 and Table 11 keep these evidence types separate. Physical retries can add electricity without another accepted outcome. Count distinct Macs, owners, accepted-work contributors, and paid participants separately; overlapping child contributions count once in the task total.

Regional reports require complete ownership coverage and at least five independent owners per released broad region, with suppression where small groups could be inferred. They expose neither

Table 11: Seven customer questions, with separate evidence for each answer.

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 CO ₂ e, 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.

household addresses nor a live device map. Dated reports preserve separate dates for work, energy, usage, and cash. Later evidence creates a new revision; it does not silently rewrite the earlier one.

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.

18 Limitations

Data suitability. Executing Macs can inspect their assigned plaintext. Encryption protects transfer and storage, not execution memory. Exposure limits apply to known owners and do not establish protection against collusion or false identities.

Workload suitability. 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.

Worked examples. Workload, profile, timing, cost, power and role-weight figures are illustrative unless labeled as measurements. The earnings calculator uses a hardware planning curve, available hours, stated power references and an electricity price; it rounds displayed amounts to cents. These assumptions are not measured hardware rates or promised work. Measurements apply only to their stated workload, runtime, hardware and checks.

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

19 Conclusion

Lithi connects the spare capacity of Macs people already own. Your agent sets the goal and limits; the Kit prepares permitted input; and the network uses supported deterministic, native-model, and generative paths to return accepted results with exceptions and cost. Owners choose when their Macs participate and share in eligible settled revenue from accepted work. The Connect calculator shows how Mac family, chip, memory and available hours shape an illustrative monthly earnings figure, including estimated electricity.

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.openai.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>

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

A Work terms and recovery

Table 12 distinguishes the goal, submitted work, physical attempts, and delivery. Recovery preserves the original work while recording any new attempt. An identifier alone grants no permission.

Table 12: What each work term means during recovery.

Term	Meaning during recovery
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	One Mac's physical try at a shard, bound to a time limit (lease) and attempt generation (fence). Replacing the Mac 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 with a fixed, immutable 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 can be planned, waiting, active, accepted, or partial; failures, cancellation, expiry, and holds follow its completion rule. Shards remain accounted for when unresolved. Acceptance, delivery, and payment have separate progress, so a recoverable delivery failure cannot undo accepted work.

B Service outputs and checks

This catalog covers all 46 customer-callable capability identities across the seven operation families; four additional dev.* build and test manifests are restricted to internal tooling. The admitted operation and release determine whether execution uses a deterministic algorithm, a native model, or a generative model, along with its resource limits. Current discovery determines available profiles and limits; a listed identity alone does not establish live availability. Large source collections are processed across bounded shards rather than inside a single model call.

B.1 Preparation obligations

On your machine, the Kit fixes the source version and checks its format and resource limits. Typed records preserve IDs, integer precision, decimal scale, nulls, and timestamps. JSON, JSONL, CSV, Parquet, and Arrow use different profiles; unsupported nesting or conversions are not silently flattened [12, 13]. Documents preserve reading order and tables. Supported OCR reports source regions and uncertainty.

Workbooks keep formulas separate from cached values; slides retain their order and permitted notes. Media preparation preserves selected tracks, languages, channels, and timebases. Storage boundaries do not define task boundaries: related records can cross Parquet row groups, so preparation keeps their required groups together before encryption.

B.2 Capability catalog

Families can reuse a capability. A display label creates no new service, and an output digest alone does not complete it. Table 13 lists each identity's input, output, and checking subject.

Table 13: All 46 customer capability identities: input, output, and acceptance subject.

Capability identity	Required input	Returned output	Required checks
agent.bulk_work	Bounded items, pinned source or diffs, or recorded diagnostics	Typed results or source-linked findings with exceptions	Revision and line anchors, required context, rubric, schema, and full coverage
agent.candidate_set	Declared candidates or designs, references, and fixed rubric	Comparison, selection, disagreement, and abstentions	Candidate and source identity, evidence, and coverage; no automatic patch application
audio.classify	Qualified audio signal	Timestamped audio observations	Declared classes and source alignment
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 transcript and timebase	Subtitle artifact	Cue order, text, timebase, and encoding
speech.translate	Admitted speech and locale pair	Aligned translated output	Locale pair and declared output modality
speech.synthesize	Permitted text, voice, and synthesis profile	Speech audio artifact	Voice permission, language, and decoding
document.extract	Qualified document representation	Source-linked text and tables	Coverage, citations, and omissions
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
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
text.classify	Permitted bounded text	Closed-label records	Uncertainty, rubric, and item coverage
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
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

Table 13: All 46 customer capability identities: input, output, and acceptance subject. (continued)

Capability identity	Required input	Returned output	Required checks
video.transcode	Qualified container and codec tracks	Encoded video and tracks	Decode, duration, and synchronization
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
ml.coreml	Qualified model or package and typed inputs	Typed predictions or measured report	Exact operation, model, features, runtime, and tolerance

C Encryption and key access

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

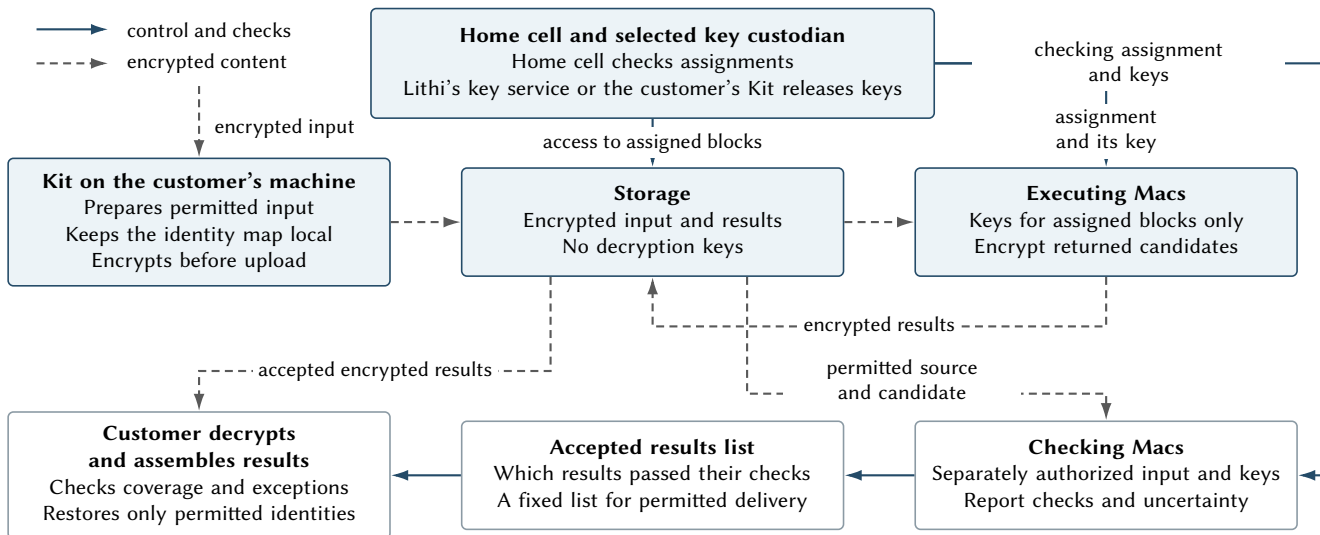


Figure 11: Storage holds encrypted content. The selected custodian releases keys for authorized execution and checking; the customer assembles accepted results.

Table 14 names the public cryptographic methods and their roles [7–10].

Table 14: Public cryptographic methods and their roles.

Method	Role
AES-256-GCM	Encrypts each block with a fresh 32-byte key, 12-byte nonce, and 16-byte authentication tag [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)	Combines classical X25519 and post-quantum ML-KEM-768 key encapsulation; the cited revision is an Internet-Draft, not a final RFC [9].
HKDF-SHA256	Derives domain-separated keys for distinct cryptographic purposes [10].

The home cell verifies the active assignment, recipient, lease, fence, and epoch before releasing keys, authorizing execution and checking separately. Section 10.1 and Table 6 detail custody and memory limits. Revocation blocks new authorized key openings; it cannot recall plaintext or keys already copied.

D Metering, billing records and collection

Signed metering recipes fix how admitted input and accepted output convert to tokens (Table 15). Internal ledgers track integer nano-dollars, or billionths of a US dollar, so display rounding never alters admission or settlement. Purchases, usage, and refunds remain separate records.

Table 15: Five representations, one counting rule each.

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 16 distinguishes the three work quantities.

Table 16: Three distinct work quantities tracked by the ledger.

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

D.2 Record the credit purchase

A verified purchase produces one paid invoice. Delayed invoice generation or delivery retries the document without charging again.

D.3 Record accepted usage

The monthly statement records accepted usage, purchases, holds, consumption, corrections, and carried credit for an exact period. Its prepaid portion has zero amount due because funded credit already covered the consumption. Finance distinguishes prepayment from

later service use so importing both documents does not duplicate an expense or payable.

D.4 Reconcile refunds and uncertain payments

Refunds concern eligible unused value from the original purchase. Lithi holds that value, reconciles the approved refund, and links its credit note without refunding twice. Pending refunds and disputes keep their holds; they neither erase delivered work nor create negative usage. Invoice and statement access remains available at zero credit.

Confirmed payment funds the account once. Repeated notifications cannot add extra credit. If the outcome is uncertain, Lithi checks the original payment rather than requesting another charge; closing a browser does not establish payment failure. Provider pay-outs remain separate from customer purchases. Revenue allocation and the invitation example are explained in Section 14, with the task figures in Table 10.

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

Table 17 illustrates full-task accounting for a 24-hour translation endpoint with a 95% completion floor. Configuration A runs all 6 million input and 1 million output tokens on the host agent’s model at \$3 and \$15 per million (\$33.00). Configuration B delegates translation to Lithi (\$0.396 for 990 accepted documents) and assumes \$5.00 in host-agent calls for planning and coordination. Both add \$2.00 for local setup and review. Value ratios use exact totals before display rounding.

Table 17: Illustrative full-task costs and assumed checked outcomes.

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

E.2 Input-heavy work and Mac earnings

A supported native or deterministic path can process substantial permitted input and return a compact result. For example, 1 million billable input tokens and 1,000 billable output tokens across admitted shards cost \$0.0501 under Equation (1). Assuming the full charge is eligible, that task contributes \$0.025050 to the provider pool. These task amounts describe admitted usage; the monthly illustrations below cover contributions across jobs.

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 18 uses the same functions as the Connect calculator [22]. The dated electricity-price assumption comes from EIA [23]. These are planning illustrations, not measured income or promised work.

Table 18: Illustrative monthly earnings, after estimated electricity and before tax. Available hours scale the illustration; accepted work determines actual earnings.

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

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-adaptor 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.

When comparing generative and supported native or hybrid routes, fix the source population, required output, quality checks, deadline, and completion rule in advance. Compare only routes meeting that endpoint. Record every stage, returned artifact, accepted and unresolved unit, physical attempt, full cost, and electricity. Include local Kit preparation in the workflow comparison even though it creates no provider earnings.

E.3 Decide whether the outcome improved

Documents from one source family form one evaluation group, and retries stay with that group. Execution shards do not create statistical independence. A separate hypothetical comparison uses 1,000 independent source-family groups per method, with mean scores of 0.96 and 0.99 (Table 19).

Table 19: Hypothetical independent-group comparison: an observed gain does not establish an improvement.

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

Each group contributes an equally weighted score Y between 0 and 1, such as the fraction delivered and checked in time; document weighting needs its own rule. With fixed sample sizes, planned comparisons, and one analysis time, Hoeffding’s inequality [16] gives Equation (9).

$$\hat{\Delta} = \hat{\mu}_B - \hat{\mu}_A, \quad (9)$$

$$\varepsilon_a = \sqrt{\frac{\ln(4K/\alpha)}{2n_a}}, \quad a \in \{A, B\}.$$

Each hatted mean averages n independent groups; $\hat{\Delta}$ subtracts A’s mean from B’s. K counts planned comparisons and α bounds family-wise interval error. With 1,000 groups per method, $K = 1$,

Table 20 lists permitted comparison dimensions and the invariants that remain fixed across every trial.

Table 20: Examples of permitted improvements and the limits that remain fixed.

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.

and $\alpha = 0.05$, each allowance is about 4.68 percentage points. The interval adds and subtracts their sum, 9.36 points, from the observed difference and clips to $[-1, +1]$. Here the three-point gain gives $[-6.36, +12.36]$ percentage points and remains inconclusive.

For independent bounded outcomes under this fixed design, the chance that any planned interval misses its true difference is at most α . This coverage guarantee assigns no probability to the fixed true difference after observing the interval.

Claiming an improvement requires the entire confidence interval to lie above a predeclared threshold δ while meeting every quality and safety floor. Testing cost reduction at preserved quality requires a separate non-inferiority margin.

Fix groups, success criteria, and stopping time before testing; count every assigned unit. Bound B’s missing scores least favorably and A’s most favorably; a changing conclusion stays inconclusive. Retries add no samples. Repeated looks, adaptive candidates, and reused groups need a method accounting for that design.

E.4 Test and approve network changes

Figure 12 follows one proposed network change from a fixed comparison through testing, accounting, evaluation, and independent release approval. Home cells retain acceptance. Monitoring and retained findings guide later questions without widening the original evidence or permissions.

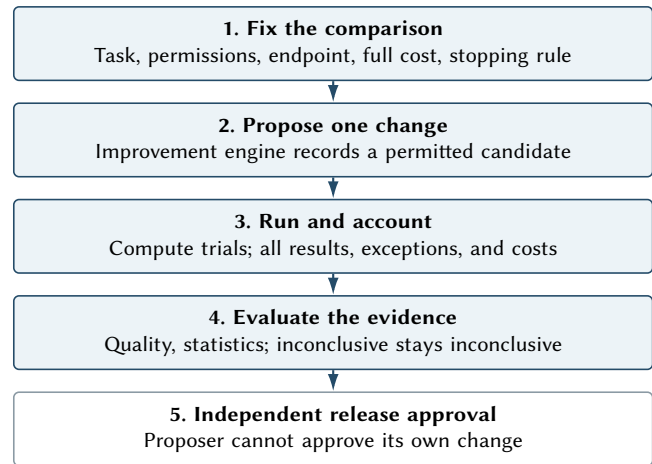


Figure 12: Independent approval follows testing. New jobs use approved settings; existing jobs keep theirs. Monitoring, rollback, and scoped findings guide later comparisons.

Table 20: Examples of permitted improvements and the limits that remain fixed. (continued)

Question	What Lithi may compare	What stays fixed
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.

F Privacy and security limits

Table 21 summarizes the risks, protections, and remaining limits for normal work and recovery.

Table 21: Privacy and security risks, protections, and remaining limits.

Risk	What limits it	What remains
Unsafe input	Supported formats, input checks, and restricted workers limit what a task can reach.	A signature or scan cannot rule out every parser defect.
Access on an executing Mac	Local preparation, owner reading allowances, and separate shard keys limit readable input.	A privileged owner of an assigned Mac can inspect its plaintext memory.
Invalid or compromised permission	Scoped API identities, cell epochs, leases, and attempt fences reject stale or forged authority.	Compromised shared signers or software can require a wider shutdown.
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.

Automated rule verification. Disposable PostgreSQL, Redis/BullMQ, and mocked-payment suites check that stale fences or late submissions cannot replace a committed winner; parallel reservations across multiple Macs respect an owner's exposure allowance A ; concurrent recognition debits a hold once without overspending; expired leases and lost queue signals recover idempotently from committed records and outboxes; and stale cell-registration epochs or interrupted payment replies reconcile without duplicate authority or charges. These checks concern execution and accounting invariants, not model quality or fleet completion times.