Machine agency infrastructure architecture draft

Cisco Caceres · 21 October 2026 Pacific · Architecture draft

Reliable machine agency requires more than a model response. A worker must receive an authorized task, execute within its permissions and resources, retain state, produce evidence, and recover or escalate when the task fails. This draft describes interfaces for connecting those responsibilities across a portfolio of separately developed systems. The components do not currently constitute one integrated autonomous platform.

The architectural question is: which contracts and controls allow independently operated models, agent clients, compute tools, and physical interfaces to cooperate without mistaking delivery, agreement, or a successful process exit for successful work?

Components and demonstrated boundaries

Implementation, a tested interface, and successful end-to-end agency are different evidence levels. The following inventory describes source and existing qualification records; it does not report a new integrated-system experiment.

Component	Existing responsibility	Evidence boundary
Echo	Member addressing, durable messaging, shared tasks, fenced task leases, operator-scoped shared repository-path leases, advisory local file holds, and coordination interfaces	Current-source offline tests cover overlapping-claim exclusion across daemons, disjoint claims, owned worktrees with serial integration, and lease persistence across daemon/broker restarts. These tests do not establish live worker reachability or coding-task benefit.
Model gateways and routers	Common provider interfaces and model-selection infrastructure	Provider support and routing must be qualified per implementation and endpoint. A common API does not establish equivalent models, tool execution, or compatibility with every agent client.
TuneHarness	Guarded experiment and compute workflows, checks, cost records, evaluation, and teardown	Completed experiments retain their declared environments and methods. Provisioning controls do not establish a shared budget across arbitrary external workers.
Bootscry	Console observation and keyboard/ mouse control through KVM tooling	Hardware evidence includes successful operations and failures requiring intervention. Console control does not establish an independent power-control or universally unattended recovery capability.
Acoustic systems	Speech recognition, synthetic-speech analysis, and related perception experiments	AMBIE's published benchmarks evaluate third-party models. Noise or Voice's internal diagnostics do not establish general synthetic-speech detection or speaker authentication.
Verifier research	Selection, stopping, error dependence, and cost-aware evaluation	The engineering pilot, authored examples, and replay tools are separate from the proposed confirmatory study. Judge agreement is not ground truth.
Applications and persistent assistants	User-facing tasks, domain state, and retrieval	Application permissions, memory, and deployment evidence require their own review. Persistent coordination records are not equivalent to semantic agent memory.
ThermoCog	Exploration of physical and energy constraints on computation	Design-stage work, with no demonstrated computational substrate or measured energy advantage.

The existing Echo integration documentation supplies runnable CLI examples, a CI task adapter, MCP interoperability boundaries, and a generic text-inference adapter. It also distinguishes documented clients from individually qualified runtime combinations. See integration contracts and the architecture and boundaries.

Connection status

Connection	Current evidence	Remaining boundary
Echo CLI/MCP to local coordination state and daemon	Current registered-command checks launch the configured Claude, Codex, and agy MCP commands; each exposes 20 tools and four resources. Offline source tests cover scoped path leases and worktree integration.	Registration and fixture checks do not establish an interactive client round-trip, live worker reachability, deployment of this source revision, or task-success benefit.
Echo AI client to a compatible inference endpoint	Text completion/streaming adapter, fixture tests, and recorded representative live checks	Tool dispatch and optional server/model capabilities require separate qualification.
Python model gateway to providers	Routing, alias, cache, and usage components in source	Current deployment and complete streaming cost accounting are unqualified.
Echo to the Python model gateway	A compatible endpoint makes an adapter feasible	This connection is proposed; compatible protocols alone are not a tested edge.
TuneHarness babysitter to local inference, Claude SDK, and CLI tools	Implemented wiring and mocked boundary tests	This is direct model access, not demonstrated gateway integration. The CLI gate permits `exec` and `ssh`; it is not a read-only operating-system sandbox.
TuneHarness training/export to serving	Documented pipeline and receipt-gated deployment components	Individual artifacts do not establish composition with coordination and physical-control systems.
Bootscry MCP to its KVM library	Adapter implementation and mock-backed stdio tests; separate hardware sessions are documented	Those evidence sets do not establish a model-driven MCP hardware session by their combination.
Echo to TuneHarness or Bootscry, and the complete portfolio loop	Proposed composition	The scoped source audit found no explicit operative cross-project adapters; wiring elsewhere would require its own evidence.

Coordination and execution boundaries

Echo is a coordination component alongside workers and tools, not a mandatory route through which every model, GPU, or hardware operation already passes. Model access, coordination, compute, and physical control can have separate adapters and authority boundaries. An architecture drawing must label a proposed adapter differently from an implemented or tested connection.

Three boundaries require explicit contracts:

1. **Assignment to execution.** A durable task or message expresses intent. The worker must separately accept ownership and demonstrate that execution began. Lease and fence checks in the coordination service do not automatically prevent an expired worker from changing an external system; that system or its adapter must enforce the relevant authority.

2. **Execution to evidence.** Exit status, a model's explanation, and a claimed verification command have different meanings. An acceptance check must observe the resulting artifact or state independently. Worker-reported receipts remain reports until their required evidence is reconciled.
3. **Evidence to recovery.** A retry must preserve identity, prior expense, and the disposition of any previous action. An uncertain outcome is not permission to repeat an irreversible operation. Recovery can require human intervention rather than another model call.

Cryptographic coordination identities and role permissions help control coordination operations. They do not confer operating-system isolation on a child process. Echo's default local file holds are advisory. A separate shared repository-path lease is acquired when a task claim includes an operator-assigned `repo_scope` and canonical repository-relative paths: the daemon reserves the complete path set in JetStream before committing the claim, and binds the reservation to the task, member, attempt, fence, and lease. Exact and parent/child path overlaps contend across daemons; disjoint paths may proceed concurrently. A successful terminal task transition releases the reservation, while uncertain outcomes remain reserved until their recovery bound.

These shared leases coordinate compliant Echo claims using the same scope. They do not prove that a checkout is the named repository, prevent direct filesystem writes or raw NATS mutations, sandbox a worker, or authorize Git ref changes. Echo-created worktrees isolate coding attempts, but competing changes to the same files can still conflict; an integration owner must select, test, and apply a candidate serially.

On 21 October 2026 Pacific, the bounded offline coordination gate passed with a worktree integration smoke and cross-daemon tests for overlap exclusion, disjoint claims, serial integration, and reservation recovery after daemon and broker restart. A separate registration check launched the configured Claude, Codex, and agy MCP commands and discovered their tools and resources. No live worker trial ran in that environment, so live cross-vendor coordination and coding-task improvement remain unqualified.

Proposed integration contracts

Each connection needs an explicit task identity, allowed actions, source/version commitments, resource limits, result schema, cancellation semantics, and failure disposition. A cross-component identity mapping must not silently equate a broker member, model account, operating-system user, and application customer.

Cost accounting must include generation, verification, repeated context, failures, retries, and the declared compute or execution costs. Local execution is not economically free; hardware and electricity can remain unmeasured and must be identified as such. Missing usage stays unknown. Independent project ledgers do not establish enforcement of a portfolio-wide budget.

An action receipt should distinguish requested, admitted, started, completed, verified, failed, and uncertain states. Persisting a receipt does not prove that an external effect happened exactly once. The adapter must define idempotency or a reconciliation procedure appropriate to the action.

The proposed control-plane role is therefore coordination of authorized work with inspectable evidence and declared resource accounting. It is a research direction, not a claim that the portfolio replaces an operating-system kernel or implements a complete hosted agent platform.

A bounded integration study

A useful first integration test should connect only the components needed by a declared task. Adding speech, physical control, or rented GPUs to a software-only task would introduce confounders without establishing their value.

The prospective path is task assignment, worker acceptance, model/tool action, artifact receipt, independent acceptance check, and a declared recovery or escalation. Qualification should first use authored actions and isolated owned resources. Model-driven evaluation follows serving identity, tokenizer/request accounting, data rights, and budget qualification. This proposal authorizes no calls or compute acquisition and changes no frozen verifier protocol.

The comparison must give a simpler coordination baseline the same task information, allowed tools, confinement, model resources, total budget, and acceptance checks. Measure final accepted tasks, duplicated or refused actions, lost state, interventions, known expense, missing usage, and recovery time. A fault arm must specify when the fault is injected and retain tasks that never reach that point. Public fixtures, model capability, recovery mechanics, and fresh-task performance remain separate results.

Echo already has a narrower recovery and coding-comparison protocol. Its retained synthetic diagnostics establish no coordination benefit. A new integration study must identify its own question and controls rather than reinterpret those outcomes as proof of a general operating environment.

Research contributions and publication gates

An architecture paper can contribute a precise contract inventory and testable failure taxonomy before it contributes an empirical performance result. Its value depends on making integrations and missing guarantees inspectable. Claims of improved reliability, useful autonomy, or correctness per unit compute require controlled results with complete denominators and accounting.

The verifier research remains a separate model-method question about inference-time decision information under fixed budgets. This architecture can supply operational requirements and experimental infrastructure; using Echo to run a study does not make Echo part of its scientific treatment or establish novelty in the verifier method.

Before public release, complete the source-backed connection matrix, review the authority and evidence semantics, and distinguish existing qualification from proposed tests. Before claiming an integrated platform, demonstrate the declared end-to-end path and failure behavior. Before claiming frontier-model transfer, evaluate the relevant models and tasks under a separately frozen protocol.