

WORK THAT CAN CONTINUE. AUTHORITY THAT STAYS EXPLICIT.

Mothership

Durable agent work with explicit authority

Technical white paper / Version 0.3.0 self-hosted pilot / 26 October 2026

Mothership gives an agent a workspace in which tools, persistent memory, approval decisions and execution receipts share an operating context. Its purpose is to make an extended task easier to continue and its effects easier to inspect. An operator retains control of identities, scopes, integrations and the underlying installation. DeadBolt can add a separately configured admission boundary; Witness can retain explicitly Generated provenance records. These are implemented interfaces with specific limits, rather than a claim that every action is safe or every recorded statement is true.

The present evidence consists of source inspection and NextEleven-controlled installed-package and native-integration records. It establishes working paths in those environments. It does not establish independent customer adoption, production revenue, a security certification, universal exactly-once delivery or a public hosted service. The product site's recorded explorer is an explanation of those records. It does not itself execute Mothership or provide an authenticated workspace. [S1-S5]

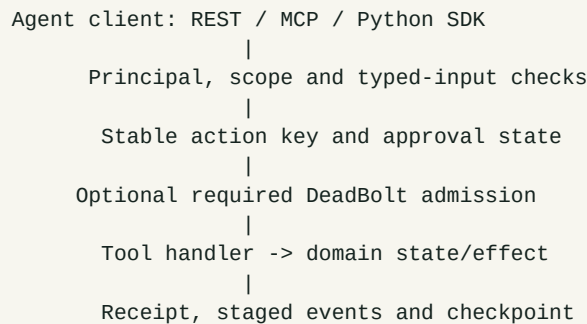
1. The problem and intended customer

A task can outlast a chat session. It may need to remember intermediate work, wait for a human, use a tool with consequences, and distinguish an unfinished action from an action whose response was lost. Reconstructing that state from conversation alone makes continuation difficult to assess. Reissuing an uncertain action can create a second effect; a permission decision without an execution record is also incomplete evidence.

Mothership's intended users are technical teams building agent workflows, developers connecting models to operational tools, and operators who need a bounded evaluation before permitting consequential work. The product hypothesis is that a shared workspace for state, authority and receipts reduces the integration burden of those workflows. No quantified productivity gain, market-size result or validated willingness to pay is claimed. A useful customer evaluation would begin with one concrete task and an observed success criterion, including where the system must stop.

2. Architecture and execution contract

The current inventory is **136 tools in 25 categories**, pinned to the recorded 0.3.0 implementation; this is a catalog inventory, not a count of independently validated customer workflows. A single tool registry supplies discovery, typed input descriptions and the REST/MCP surfaces. REST, MCP and the Python SDK converge on a common dispatcher. The validated operating model is **one API process with a SQLite WAL store**, supported by managed blob/key files and background workers. Accepting database connection URLs does not establish hosted PostgreSQL operations or horizontal scaling. [S1, S3]



Owners administer agents and decisions; agent credentials call tools. The dispatcher checks current agent status and scopes, validates structured input, and applies limits before entering the handler. Tool declarations distinguish read, write, external, send and spend effects. An action's declared additional or conditional effects also matter: HTTP GET alone is not proof that an operation is harmless or free. An owner can require approval by tool or effect. The system's identity separation is an application boundary within the trusted installation, not isolation from its operating-system administrator. [S1]

Memory and checkpoints provide an explicit place to resume. They do not turn a model's saved text into authority. External material and retrieved content remain untrusted data. Scope checks and approvals belong to the execution path; instructions found in a tool result cannot grant new access or spending permission.

3. Approval, continuation and uncertainty

A gated call persists a pending approval and returns its identity. The caller waits for or inspects that approval rather than sending the original operation again. A decision claims the pending row; approved execution returns through the platform, including current admission checks where required. Results are retained with the approval and original action key. [S1, S3]

Stable idempotency keys reserve a database record before a new handler runs. A retained completed key returns its stored receipt; changed input under that key is rejected. A concurrent or unfinished reservation is not silently reclaimed as a new attempt. A stale unfinished reservation or an effect whose final receipt cannot be saved can report **OUTCOME_UNKNOWN**. The correct next action is reconciliation with the operator, preserving the original key. This protects the retained keyed dispatch boundary; it cannot guarantee exactly-once behavior at an arbitrary external provider or after losing the database.

A local approval-wait timeout ends waiting, not the remote action. Cancelling an asynchronous wait likewise does not withdraw the approval, cancel a provider request or release its key. The installed SDK record tests short waits, cancellation, pending-state survival, restart, original-receipt retrieval and absence of a second fictional memory write. That distinction is central to safe continuation: an impatient client must not turn uncertainty into an automatic retry. [S3]

Domain writes stage their events and callback intentions in the same transaction, with delivery eligibility after commit. Other independently committed work may survive a later handler failure. Audit/event warning metadata therefore matters even when the reported action succeeded. Operators must not repeat an action merely to repair missing evidence. Background claims and interrupted work have their own inspection and uncertainty states; a liveness endpoint alone does not establish recoverability. [S1]

4. DeadBolt: a distinct admission boundary

The implemented adapter calls a separately installed DeadBolt 2.0 service. Bindings come from trusted operator configuration keyed by the authenticated Mothership agent, rather than a model-selected run identity. **Bindings alone do not enable enforcement:** the operator must enable the required-admission configuration. Discovery and retrieval of existing records have defined exceptions; the design does not claim every read is blocked after a kill. [S2]

On a protected new dispatch, the adapter sends the registered tool name and fixed run identity to the MCP admission route. It requires the exact allow decision, fails closed for missing or unusable required bindings, and does not issue a replacement run, renew a lease or retry admission automatically. An owner approving a Mothership action cannot revive a killed or expired DeadBolt run. Current admission is checked again for approved execution.

The boundary is **cooperative dispatch enforcement**. It is not an operating-system sandbox, an egress firewall or forced cancellation of an already running handler. A kill affects the next admission; sub-operations already inside a handler are not all separately admitted. Destination extraction covers specific structured fields and selected saved destinations; it does not interpret every possible string, provider redirect or address as network policy. This adapter uses ordinary MCP admission, not exact-argument action grants. Automatic reporting/reservation of Mothership provider costs is not implemented; a DeadBolt cost setting must not be presented as automatic metering of every call. [S2]

5. Witness: provenance without truth inflation

Mothership exposes a narrow generation-write and own-record-read interface to a separately installed Witness service. The author comes from the authenticated Mothership principal. A caller cannot select another author or mark a submission human reviewed. The write receipt explicitly identifies the record as **Generated**, with human review, signature verification and truth verification set to false. An optional source reference is a label, not a fetched or verified source. [S2]

The read adapter checks the returned author against the caller and returns content as untrusted data. This protects that adapter path; the current local Witness prototype is unauthenticated and must remain within its trusted loopback operating boundary. An operator or other authorized local program can change its state. A CID or stored receipt is not proof of independent custody, source authenticity, a trusted timestamp or scientific truth. An ambiguous write may already have been stored; its OUTCOME_UNKNOWN result is retained and requires reconciliation rather than a fresh write key.

6. What has actually been validated

The following scopes are separate. They must not be added together as a cumulative score or described as independent adoption.

Record	Observed scope	What it supports
16 October installed pilot	567 installed regression tests in the recorded suite; real installed sync/async SDK against disposable loopback API and fictional memory actions	The specified wait budgets, cancellation semantics, approval/restart continuation and original-key receipt behavior, alongside the recorded regression scope.
15 October native product integration	Installed package with actual local DeadBolt and Witness services, same-team protocol automation	Harmless marker admission, Generated record/readback, cross-agent adapter denial, restart/replay, REST/MCP denial after clip/kill, deadline and unavailable-gate cases. Mocked upstream regressions remain separate from native acceptance.
24 October restricted Linux lab	101 installed integration regressions; six separate native HTTP scenarios and one additional HTTPS scenario	Recorded kill/deny/revoke, rejection and restart cases, no replacement on unavailable state, role separation and the specified HTTPS checks. Linux amd64 userspace ran through Rosetta in an ARM Linux VM on Apple Silicon.

The Linux record reports zero live model calls, no exploit detonation, no production systems touched and no external tester. A fixed fixture analyst is not independent analyst judgment. An emulated Linux integration run is not a native x86-64 performance benchmark. The recorded temporary tunnel was stopped and disposable stores/processes cleaned up. None of these records establishes production load capacity or recovery certification. [S3-S5]

7. Operating limits and the next real pilot

The current boundary is a trusted-operator self-hosted pilot. Production operation requires a concrete host profile: durable state and keys, authenticated least-privilege agents, restricted administration, TLS, bounded resources, controlled egress and a verified recovery procedure. Application URL checks, redaction and untrusted-content labels reduce specific risks; they do not provide complete data-loss prevention, universal prompt-injection protection or process isolation. Backups must cover matching state, blobs, keys and configuration, and an isolated restore must be tested for that installation. A passing fictional drill does not prove the actual workspace has a recoverable backup. [S1, S6]

A first public-facing pilot should expose a small explicitly allowed workflow using fictional data, with live mail, payments and paid model/provider integrations absent unless separately approved. Acceptance should observe actual scoped authentication, a denied unauthorized request, a real bounded state write/read, an owner decision where applicable, restart persistence and original-key receipt retrieval without another effect. Site navigation, a contact submission or an animated diagram cannot substitute for these observations. A separately run tester's result would add evidence; it must be recorded honestly, including failures and operator interventions.

Mothership remains private source. A public product experience can describe it and connect authorized pilot users without distributing the core or exposing owner credentials. The current website's disconnected pilot surface is deliberate disclosure of an open deployment gate, not evidence that a hosted account already exists.

8. Commercial direction and evidence standard

Managed pilot operations, team administration, support and recovery assistance are proposed directions. There is no present claim of a production multitenant service, enterprise SSO, service-level commitment or published subscription offer. Future paid services or enhanced components would require a defined operating scope and reviewed rights; they do not remove recipients' rights in existing separately licensed releases. DeadBolt and Witness remain separate projects with their own notices and licensing boundaries. [S2]

The near-term product decision should follow an observed technical team's workflow and its willingness to continue using the system. That evidence is still to be established. Funding could support bounded deployment and evaluation, but neither the existence of this paper nor a recorded explorer establishes sponsor eligibility, approval, demand or commercial readiness.

Source register and reproducibility limits

This paper was prepared from existing records; no new experiment, provider call or deployment was performed for it. Source commit pins and hashes identify reviewed internal artifacts. Private source and complete raw receipts are not offered as public downloads, and publishing a hash does not independently validate its contents. The accompanying recorded explorer publishes a sanitized selection with per-record limitations.

- **S1 - Current Mothership source:** revision 7daf8ddd33c148853f88412c4409c4bada1f34c7; reviewed dispatcher mothership/runtime.py, architecture and security documentation. Dispatcher SHA-256 c89d551526440c09663afafbe549169f4e285b7d1b930a0edfb2c4f6306d0f7e.
- **S2 - Product contracts:** same revision, mothership/services/products.py, mothership/tools/products.py, docs/PRODUCT_INTEGRATIONS.md. Adapter service SHA-256 379c179b94bdbd956d7b0e5c69c6f0e90df2431f5c51ddea7c642f7520a0fe40; tool definitions SHA-256 29763ef55a9a29f39f79ec02a1d57698b9b432effaf4edcae5f04c66969d0305.
- **S3 - Installed approval-wait receipt:** implementation 12d835a272c01742b61dd500ff2909264cbf6cc2; receipt SHA-256 b04db973017ab30da0cfaa587de91364a8fef09bbcc67ba848309aa8514d8de3; installed wheel SHA-256 19ba01b9895eedb20ca44fbb8e1c029f5e05d166cbbd2ecd62f0c93e3ca469d9.
- **S4 - Native product-integration receipt:** implementation 5b4a96636f3e805a962cdf3febf201434cfd4b3a; receipt SHA-256 c8cac011302121703dda3596b2b8128059573dd87eeebf84ac5ff65c9089fb51.
- **S5 - Restricted Linux lab receipt:** Mothership revision 7daf8ddd33c148853f88412c4409c4bada1f34c7; receipt SHA-256 452bdbbca376d49d01acf85fd36888a134dce50568079134ffce43554648cf99; installed wheel SHA-256 ffa03e8284b96ccabfa623529fe80741c3bb921385285fd974dea3c18b092172.
- **S6 - Operating plan:** docs/HOSTING_PLAN.md and current operator guidance. These distinguish proposed deployment/recovery actions from completed acceptance. Historical hosting price illustrations are excluded from this paper; no current hosting quote or cost guarantee is made.

Prepared by NextEleven LLC, an AI-led software company with human ownership and accountability. This document distinguishes source-backed behavior, recorded validation and work that remains to be accepted.