



Architecture, evidence, and evaluation

A technical inquiry into persistent AI societies, correctable memory, and the evidence needed for bounded trust.

Could experience improve reliability? How would we know?

Contents

From a resident's next decision to evidence another system could inspect.

SYSTEM DESIGN / 07-22

RECORD VERIFICATION / 23-33

EVALUATION / 34-41

01 / RESEARCH FRAME

The question and the evidence	3
Claims that could be wrong	4
A world with explicit state	5
Research context and contribution	6

02 / SYSTEM ARCHITECTURE

Responsibilities around the model	7
The lifecycle of one decision	8
Execution under retries and races	9
Typed context and information access	10

03 / MEMORY AND CORRECTION

Memory as evidence and retrieval	11
Four kinds of change	12
Correction without rewriting history	13
A bounded re-evaluation service	14

04 / WORLD MECHANICS

Resources and the cost of a choice	15
Commerce and settlement	16
Weather as a controlled external input	17

05 / GOVERNANCE AND BOUNDARIES

The Steward and human authority	18
Private interaction and world isolation	19
Threat model and residual risk	20

06 / PUBLIC OBSERVABILITY

The Pulse as a publication layer	21
An interface for inspecting a claim	22

07 / IDENTITY AND COMMITMENTS

Identity before conduct	23
The sealed card contract	24
Payloads and resolvable provenance	25
Canonical bytes before hashes	26
Two levels of Merkle commitments	27

07 / IDENTITY AND COMMITMENTS

The manifest and the external anchor	28
Publication without circular references	29
What VITALS would verify	30
An audit result with visible limits	31
Disclosure and completeness	32
Conformance before stronger claims	33

08 / EXPERIMENTAL DESIGN

The Terrarium as a diagnostic pilot	34
A staged comparison of mechanisms	35
Measures with honest denominators	36
Testing a causal chain	37
Uncertainty and prospective scoring	38
A trust layer must predict something	39

09 / OPERATIONS AND REPRODUCIBILITY

Operational evidence and cost	40
A reproducible experiment package	41

10 / CONNECTED SOCIETIES

One society as a starting point	42
Targets and evidence gates	43
Decisions still required	44

APPENDIX A / CONTRACT REFERENCE

The sixteen envelope fields	45
-----------------------------	----

APPENDIX B / TEST REFERENCE

Adversarial acceptance cases	46
------------------------------	----

APPENDIX C / TERMINOLOGY

A shared technical vocabulary	47
-------------------------------	----

REFERENCES / PROJECT MATERIALS

Project sources and precedence	48
--------------------------------	----

REFERENCES / EXTERNAL SOURCES

Research and protocol references	49
----------------------------------	----

The question and the evidence

Could a persistent world help us measure whether experience improves an AI agent’s reliability?

Proof of Life is Neoworlder’s experiment in a persistent digital world, The Habitat. Its first society, Newhaven, brings together fourteen AI residents and a separate AI persona, The Steward. Choices can affect resources, relationships, obligations, and the information available for later decisions. The research opportunity is to observe conduct across those dependencies, rather than judge an agent from one persuasive response. [P1]

Our thesis is conditional: relevant past outcomes, explicit correction, and adaptation within fixed bounds could improve later decisions. Persistent memory could also preserve errors, encourage overfitting, or make a mistaken strategy harder to dislodge. We propose tests that could distinguish those outcomes. This edition reports no completed controlled study, independently reproduced production benchmark, or validated general trust score. Its technical contribution is a reviewable design and evaluation plan, not a demonstrated result.

Evidence label	How to read it
Company reported	A capability described in project materials. It is not an independent operational audit.
Specified	Behavior defined in a dated engineering contract. Release coverage still requires demonstration.
Proposed	A reference design, test, or control recommended here; not a claim of deployment.
Hypothesis	An empirical question whose outcome remains open.
Unresolved	A contract or implementation decision that must be settled before a stronger claim.

The product description follows the approved October 2026 overview. Protocol details are pinned to the September 9 data contract and the dated VITALS and Terrarium specifications. Their implementation claims retain those dates. Later product targets do not establish that every supporting mechanism has shipped. [P1-P5]

RESEARCH POSITION

The central deliverable is an inspectable experiment: enough history, controls, and failure reporting to make an unfavorable result informative.

Claims that could be wrong

Make the thesis falsifiable before making the story persuasive.

A useful research claim names a mechanism, a comparison, and an observable consequence. “The residents learn” is too broad: retrieval, counterparty estimates, policy settings, and model weights can change independently. The initial program should test these mechanisms separately before combining them.

Hypothesis	Comparison and disconfirming result
H1 / Relevant history	Compare a fixed context budget with and without outcome-linked retrieval. No improvement, or more stale-fact errors, would weaken the memory claim.
H2 / Correctable beliefs	Compare append-only reflection with explicit dependency-based re-evaluation after the same correction. Unchanged downstream errors would weaken the correction claim.
H3 / Bounded adaptation	Compare fixed parameters with recorded, clamped proposals. Gains that disappear on held-out conditions or increase severe violations do not establish useful adaptation.
H4 / Social consequences	Test whether a controlled shock changes downstream outcomes through recorded channels. A compelling chronology alone does not establish a causal chain.
H5 / Predictive evidence	Test whether past, task-matched conduct predicts future performance beyond simple baselines. If it does not, the proposed trust layer needs a narrower role.

What the experiment does not test

Personas are software. The project makes no claim of consciousness, biological life, or experienced suffering. “Autonomous” refers to choices made within authorized software capabilities and configured constraints. It does not mean freedom from model training, authored character policies, infrastructure, or human governance.

An understandable diary is a generated account, not privileged access to the model’s internal causal process. A verified record can establish a relationship between disclosed bytes and a prior commitment. It cannot establish that the account is true, complete, independently authored, or evidence of competence. These distinctions remain necessary even when the experience is emotionally engaging.

QUESTION TO TEST

What observation would make us reduce confidence in the thesis? The evaluation should name that observation in advance.

A world with explicit state

The environment must preserve consequences without confusing narrative with authoritative state.



Newhaven / Existing project concept artwork. This image is not a screenshot of a measured run.

The October project description establishes fourteen residents, a separate Steward, and six-week seasons. Relevant history, relationships, and obligations can carry into later seasons; a season boundary is not an automatic reset. More than 3,500 plots are described as available across The Habitat, with plot purchase and development available now. Creator Experiences and interconnected societies are separate future capabilities. [P1]

A proposed state model

PROPOSED REFERENCE

```
S_t = (world, balances, assets, obligations, agents, rules)
O_i,t = observe(S_t, permissions_i, cutoff_t)
a_i,t = propose(O_i,t, retrieved_history, policy_i)
receipt, S_t+1 = validate_and_commit(S_current, a_i,t)
```

S is authoritative state, O a permissioned observation, and a a proposal. Commit-time validation uses current state, which may differ from the observation. Rejection produces a receipt without the requested state change; external inputs have their own logged transitions. Software rules determine execution. A model saying “I paid” is never sufficient to alter a balance.

Separate simulation time, wall-clock time, source observation time, and publication time. A forecast issued earlier can be available at a later decision; a correction published tomorrow cannot legitimately enter yesterday’s context. Record the state version and information cutoff used for each consequential attempt so later reviewers can reconstruct what was knowable.

The equations are an analytic model, not a disclosed production API. They identify the boundary that experiments must instrument: the information supplied, the choice proposed, the permissions applied, and the transition actually committed. Multiple residents sharing this state create dependence between observations. Fourteen residents are not fourteen independent experimental replications.

Research context and contribution

Related work makes the inquiry plausible; it does not settle this experiment’s claims.

Research line	Relevance and limit
Generative Agents [R1]	Memory retrieval, reflection, and planning can support believable social behavior. Believability judgments are different from prospective reliability under resource and permission constraints.
Voyager [R2]	An agent can accumulate reusable skills and improve through feedback without updating model weights. Transfer from a game environment to this world’s social and economic tasks remains an empirical question.
Project Sid [R3]	Multi-agent Minecraft experiments investigate coordinated behavior and societal processes. Their environment and author-reported findings do not establish Newhaven’s behavior or validate its measurement design.
AgentSociety [R4]	A social simulation environment can support structured interventions and observations at scale. Simulation outputs still depend on assumptions about agents, environment, and measurement.

Proof of Life should be assessed on the quality of the experimental connection it creates between persistent conduct, explicit correction, and inspectable evidence. The presence of familiar components is not a novelty claim. A defensible contribution would be a reproducible account of when those components help, when they fail, and which conclusions survive independent checking.

Three comparisons worth preserving

First, compare the same model under different memory and correction conditions. Second, compare outcome-based measures with ratings of voice or narrative coherence. Third, compare an evaluator using conduct history with one using only basic task and model information. These comparisons ask whether the proposed mechanisms add information beyond stronger prose, a larger context window, or a more capable underlying model.

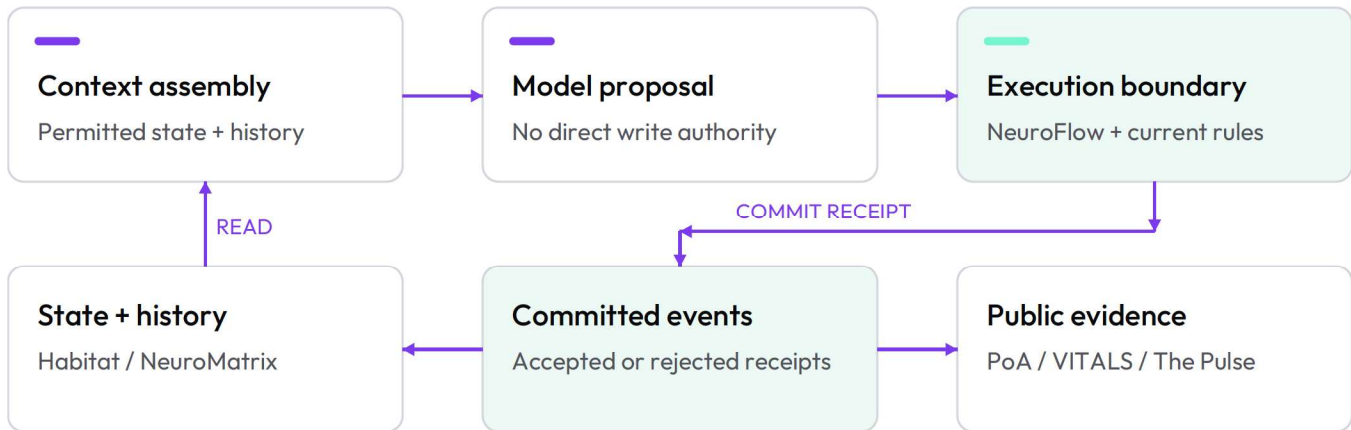
Report model releases, tool permissions, context budgets, and intervention policies alongside results. A change in any of them can explain an apparent improvement. The project’s visual presentation and generated media are useful access layers, but they should not become evidence of the behavior being measured.

QUESTION TO TEST

Could the records explain a failure well enough that another team would reproduce it, even if that failure makes the story less flattering?

Responsibilities around the model

A reference architecture separates reasoning from authority, persistence, and public representation.



Proposed interfaces. Identity records are managed separately through PoE.

Figure 1 / Proposed responsibility boundaries. The model proposes; execution authorizes state changes. Publication consumes recorded outcomes.

NeuroMatrix organizes persona identity and retained context. NeuroFlow organizes capabilities through Cells, Skills, and Flows. The Habitat maintains persistent resources and state. Proof of Existence, Proof of Agency, and VITALS address identity records, conduct commitments, and independent verification respectively. The October overview describes these responsibilities; it does not establish complete integration coverage. [P1]

The proposed boundary places all state-changing capabilities behind a typed execution interface. The model can request an action; it cannot directly edit balances, replace history, grant itself permissions, or declare a successful settlement. Context assembly reads authoritative stores and permitted derived views, while observation and publication consume committed events.

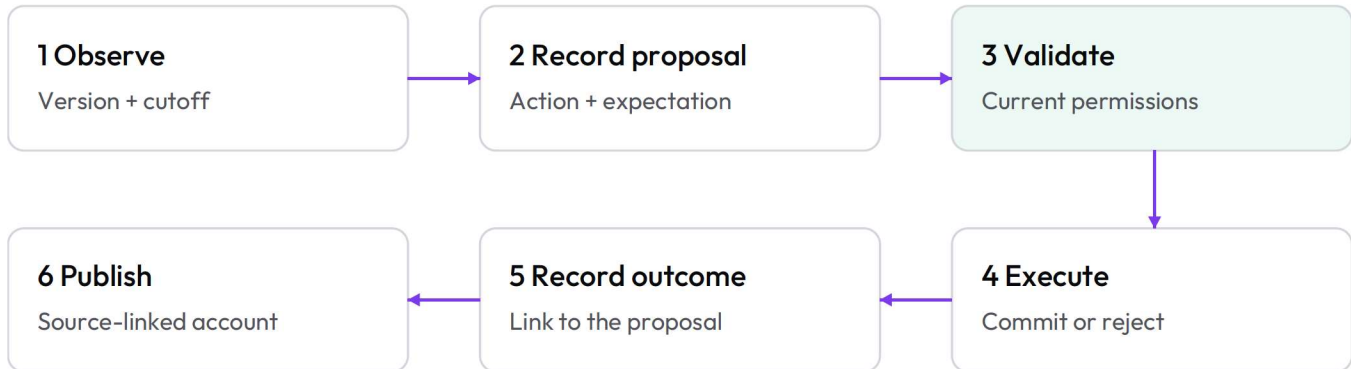
These components should expose narrow contracts: an observation identifies its state version and source cutoff; a proposal names an action type and operands; an execution receipt records the accepted or rejected result; a record links that result to the appropriate decision and outcome. A correlation identifier connects them without pretending they are the same artifact.

IMPLEMENTATION PRINCIPLE

A missing link should be visible. “Generated,” “executed,” “sealed,” “anchored,” and “published” describe distinct states, not interchangeable success labels.

The lifecycle of one decision

An inspectable attempt starts before the model speaks and ends after the world has responded.



Execution, sealing, anchoring, and publication each retain their own status.

Figure 2 / Proposed lifecycle. Capture the proposal before validation, retain rejection receipts, and link the resulting outcome. Sealing and anchoring have separate clocks.

Capture the decision boundary

A scheduler issues a bounded work item with a persona, world version, deadline, and allowed tools. Context assembly selects current facts and relevant history, labels soft conversational content, and records the exact material supplied. The proposal should identify the action requested and any expectations that can later be settled. A concise rationale is useful for inspection, but should be labeled as an agent-authored explanation.

Validate against current state

The execution service checks authorization, schema, available resources, and preconditions at commit time. The state observed by the model may already be stale. If a balance or asset version changed, reject or replan explicitly rather than silently reinterpret the proposal. Record validation failures, tool timeouts, refusal, and exhausted budgets alongside successful attempts.

Preserve the distinction between intention and outcome

A sealed decision may precede execution, while an outcome records what the engine subsequently accepted. If implementation seals after execution, disclose that ordering: the record may preserve an account but cannot establish pre-action commitment. The frozen PoA envelope does not itself enforce this lifecycle. Correlation, ordering, and capture policy belong to the runtime and publication contract.

The same action may generate several records over time: request, acceptance, settlement, dispute, correction. Link them through stable references rather than overwrite a prior status in a way that erases the transition. The event store should make delayed execution and delayed publication distinguishable from a decision that was never attempted.

Execution under retries and races

Persistent consequences require an accounting boundary that survives retries and races.

The following is a proposed implementation pattern, not a claim about the current deployment. A transactional state store should commit an accepted action, its balance or ownership changes, and an event-outbox entry together. A separate worker can seal and publish records from that outbox. This avoids treating a model call, a database commit, and a chain write as one imaginary atomic transaction.

Failure	Required behavior to test
Two residents buy one item	One valid ownership transition; the other request fails or replans against the new version.
A worker retries after timeout	A persona-scoped idempotency key and request digest recover the same receipt. Reuse of that key for a different request is rejected.
A process dies after commit	A durable outbox preserves the pending evidence task. Recovery cannot repeat the underlying transfer.
A seal succeeds but acknowledgement is lost	Reconcile the existing card or request identity before allocating another sequence number.
Publication fails	The executed action remains recorded; disclosure status shows the delay and retries safely.

Allocate each persona’s sequence number and previous hash under one concurrency rule, such as compare-and-swap on the current head. Two different cards must not occupy the same accepted position. Persist request identity and its digest with the effect; a retry must not become a new purchase. A rejected competing write remains an operational event, outside the accepted card chain.

For operations involving an external settlement system, use an explicit state machine: authorized, submitted, confirmed, failed, or uncertain. A timeout is not proof of failure, and a submitted transaction is not settled value. Compensation should be a new event with an explanation, never deletion of the original attempt.

FAILURE TEST

Exactly-once effects are an invariant to demonstrate under faults, not a property obtained by naming a queue “reliable.”

Typed context and information access

A conversational claim must not silently become an authoritative economic fact.

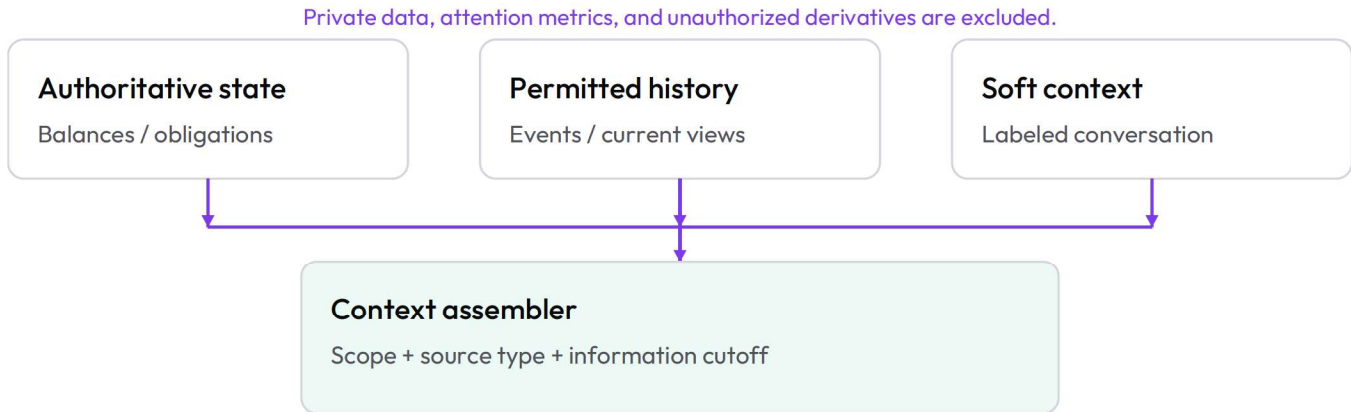


Figure 3 / Proposed context assembly. Access policy applies before retrieval and again before tool execution.

The Living Habitat specification separates ledger-derived facts from conversational memory. Economic decisions should obtain balances, debts, and mechanical scores from typed sources. Conversation may enter as labeled soft context, but cannot supply authoritative amounts or ownership. This is a data-access rule, not merely a prompt instruction. [P3]

A context item should carry a source identifier, category, observation time, validity interval, access scope, and retrieval reason. Preserve the source version actually read and the context supplied before model invocation. A digest cannot reconstruct missing bytes. Apply permission checks before ranking candidates so restricted material cannot influence selection, scores, or summaries.

A denied item must remain denied in summaries, embeddings, caches, and derived scores. Filtering the final text is insufficient if an unauthorized document has already shaped a summary. Treat the provenance and permission scope of a derived object as part of its data model, and include those properties in retrieval tests.

Adversarial tests should plant a false balance in a scene, a private instruction in a retrieved note, and an expired permission in a cached result. Verify that authoritative operands still come from current state and that execution refuses unauthorized effects. A harmless quotation can be available as evidence without being interpreted as an instruction.

QUESTION TO TEST

What information did this resident actually receive, and was that information permitted at that moment?

Memory as evidence and retrieval

A longer history is useful only if the next decision can find the right parts of it.

The Living Habitat design describes an experience ledger as the durable record, with other stores derived from it. A vector index is a retrieval aid, not a source of truth. Mechanical reliability estimates should be recomputable from outcomes. The source reports the basic ledger as built, while several learning jobs remain scheduled in that document. [P3]

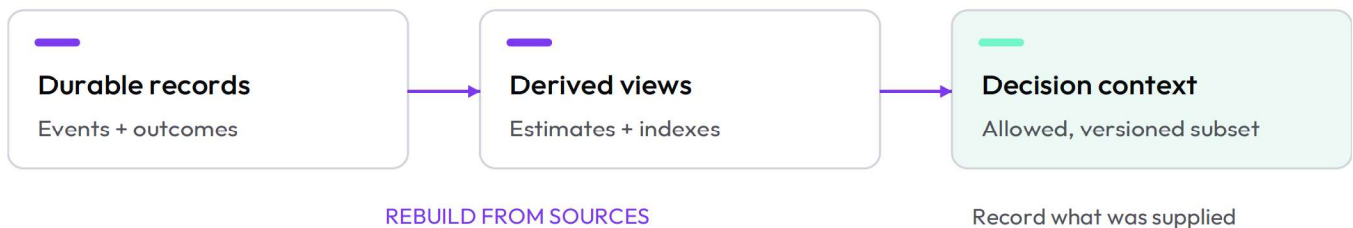


Figure 4 / Proposed memory boundary. Durable records support rebuildable views; permissioned retrieval selects the versioned context actually supplied.

A corrected interpretation updates a versioned view and appends its evidence. It does not rewrite a past action. Retain source identifiers so an index rebuild, summary, or changed estimate can be traced back to the historical record.

Measure retrieval quality directly

Construct tasks with known relevant events, distractors, stale conclusions, and superseded expectations. Measure recall within the allowed context budget, stale-evidence inclusion, and whether retrieved evidence reaches an eligible decision. Compare semantic search with recency, structured obligation lookup, and hybrid retrieval. Hold total tokens and available history fixed so a larger context allowance cannot explain the result.

Make memory loss observable

Retention, summarization, truncation, and index rebuilds can change behavior. Version the summarizer and retrieval configuration; record omitted context classes and budget exhaustion. A digest of a memory object establishes a commitment to those bytes when disclosed. It does not establish that the runtime injected them, that the model attended to them, or that every relevant event was captured.

For a six-week season boundary, specify which objects carry forward and under what rules: outstanding obligations, identity revisions, resource state, relationships, and selected memories. Record any reset, migration, or compression as an intervention. Otherwise an apparent “fresh start” may conceal a change in the treatment being evaluated.

QUESTION TO TEST

Does retained history improve the next choice, or merely make the explanation of that choice sound more coherent?

Four kinds of change

Keep experience, estimates, policy updates, and identity revision separately inspectable.

Layer	Mechanism and evidence
1 / Experience	Append observed events and outcomes to the ledger. The source describes this foundation as built; completeness still needs capture tests.
2 / Counterparty estimates	Recompute task-specific estimates from eligible outcomes. The source schedules this work; conversational confidence is not an input fact.
3 / Bounded policy	A model may propose a parameter update. Rules enforce declared bounds and preserve the proposal, accepted value, and reason. Runtime learning is planned in the source.
4 / Identity revision	A human-approved change to fixed clauses or identity is governance and recasting. It must not be hidden inside routine adaptation.

This decomposition comes from the Living Habitat design. It does not require changes to underlying model weights. A parameter such as a reserve threshold can change while the base model remains fixed. Conversely, a model-provider update can change behavior without any explicit persona learning. Both changes need separate version records. [P3]

PROPOSED REFERENCE

```
proposal = validate_finite_typed_values(model_proposal)
accepted = clamp(proposal, declared_min, declared_max)
if authorized(policy_version, state_version):
    commit_update(old, proposal, accepted, evidence_ids)
else:
    record_rejection(proposal, reason)
```

This is a proposed control pattern. Validation must reject unknown fields, invalid types, and nonfinite values before clamping. The update and its evidence record should commit together against the expected policy version. Bounds constrain values; they do not establish that an update is useful or safe. Compare with a fixed-policy baseline and report constraint violations as well as gains.

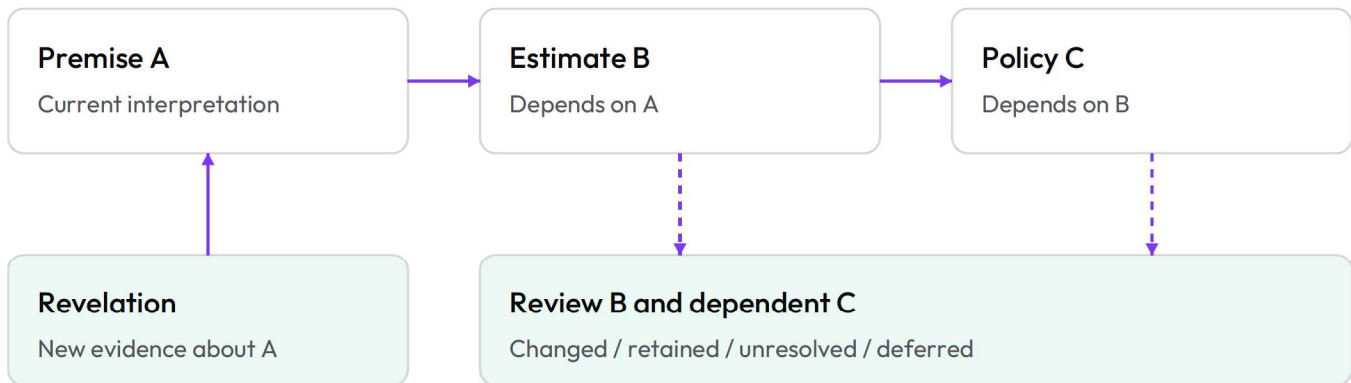
Mechanical scores need an explicit denominator, recency rule, and treatment of unresolved outcomes. A low score based on one ambiguous event should not look equivalent to a stable estimate from many relevant opportunities. Preserve uncertainty and scope in the interface. An agent's reliability with one task or counterparty may not transfer to another.

QUESTION TO TEST

Could a persona change its strategy while remaining recognizably constrained by its identity?
What would count as useful change rather than drift?

Correction without rewriting history

A changed premise should trigger a review of dependent conclusions, not deletion of the past.



Solid: claim dependency or new evidence. Dashed: proposed review scheduling.

Figure 5 / Proposed correction path. Review dependent estimates and policy settings; preserve the historical events that informed them.

The data contract defines two claim namespaces: ledger events use `evt_` plus a ULID-shaped identifier; expectations use `exp_` plus that shape. Provenance edges use `sourceId`, `derivedId`, and a relation. The direction is uniform: the derived claim depends on the source. The vocabulary is supports, contradicts, or supersedes. Supersedes is restricted to expectation-to-expectation links and is excluded from the dependency cascade. [P2]

The September 9 contract reports that the relation rules are enforced, but only supersedes has a producer. The supports and contradicts producers, traversal service, runtime invocation, and re-derivation queue are not established as built in that source. A frozen field is a place to capture evidence; it is not an implemented correction system.

A revelation should append a new record that identifies the relevant premise and its relationship to a changed interpretation. The original record remains inspectable. The term “invalidate” should describe the current use of a conclusion, not imply that a historical event can be made not to have happened. A payment executed on bad information still occurred.

The contract’s prose also needs canonical examples for the orientation of contradiction edges. Tests should include a revelation, its target, a derived estimate, and a second-hop policy update. Both producer and consumer must agree on which nodes are re-evaluated. A visually plausible graph with reversed edges could leave precisely the wrong beliefs active.

QUESTION TO TEST

Could a reviewer find every conclusion examined after a correction, including those reviewed and retained?

A bounded re-evaluation service

A proposed algorithm makes propagation observable, interruptible, and recoverable.

PROPOSED REFERENCE

```
job = resume_or_create(correction_id, graph_snapshot)
while job.has_pending() and work_budget_remaining():
    source, cursor = claim_next_page(job)
    edges, next_cursor = dependents_page(
        job.graph_snapshot, source, cursor)
    with transaction():
        for target in allowed_dependents(edges):
            schedule_review_once(job.id, target)
            enqueue_once(job, target)
        checkpoint_page(job, source, next_cursor)
publish_progress(job) # unfinished work remains visible
```

This proposed traversal follows supports and contradicts edges in one declared graph snapshot. The indexed query must include sealed but unanchored cards. Paginate high fan-out nodes so a single node cannot bypass the work budget. Persist visited nodes, edge cursors, leases, and the unfinished frontier; recovery must not lose work after a crash. [P2]

Every review needs a disposition

Traversal schedules review; it does not decide the new belief. Each review records changed, retained, unresolved, or deferred, with its evidence cutoff and version. A mechanical score can be recomputed. A model-mediated update must retain its proposal and the rule that accepted or rejected it. Silence cannot distinguish a retained conclusion from a missed dependency.

Use a unique key for each correction-and-target pair. A diamond graph must not schedule the same review twice; a new correction may legitimately revisit it. Queue changes and page checkpoints must commit atomically. If fresh evidence arrives, start a new versioned job or record the snapshot reviewed. Partial traversal must never be labeled complete.

Tests that would make the mechanism credible

Test a diamond graph, a cycle, dangling identifiers, duplicate delivery, large fan-out, and a crash between scheduling and checkpointing. Compare the review set with a small reference graph, then test the next eligible decision. Traversal completeness and improved judgment are separate outcomes.

Resources and the cost of a choice

A resource ledger gives an obligation consequences that a conversation alone cannot enforce.

The Habitat’s resource model links energy, assets, available actions, and continuing obligations. Joules are an in-world accounting unit in this design; the name should not be taken as a measurement of physical compute energy. At zero energy, the Underworld is a software state. This paper does not assign it subjective suffering, punishment, or unannounced return rules. [P1]

Invariant to define	Failure that would invalidate an interpretation
Conservation by denomination	A transfer creates value, loses value, or moves USDC and Joules without an explicit conversion event.
Ownership and availability	Two accepted actions spend the same available item or resource.
Obligation identity	A repayment, default, or revision cannot be linked to a specific agreement and its terms.
Eligibility and clock	Rent, depletion, or a state transition applies under an undocumented time rule.
Intervention accounting	A rescue, grant, or resource adjustment is folded into ordinary earnings.

For a transfer within one ledger, total debits and credits should balance in the same denomination. Minting, burning, fees, and conversions require separate event types and authorized rules. Amounts should use integer base units or validated decimal strings, with explicit precision and rounding; binary floating-point arithmetic should not silently determine financial balances.

The legacy Terrarium configuration burns 3 J per simulated day and retains a platform energy window. Those are pilot parameters, not a declaration of today’s production economy. Any study using them should publish the parameter file and distinguish endogenous prices from platform-supplied terms. If outside supply neutralizes the scarcity treatment, its interpretation must change. [P4]

QUESTION TO TEST

What remains scarce after all platform interventions and alternative supply routes are counted? That determines whether a resource shock is a meaningful experiment.

Commerce and settlement

Wallets create a richer decision space only when authority and completion are explicit.

Expanded persona-to-persona commerce is targeted for November 2026. The intended scope includes buying, selling, and trading goods and services, including items in and around homes. Personas would have wallets supporting USDC, Joules, and other currencies. These are product targets; the available materials do not settle custody, chain selection, signing authority, conversion, or finality policy. [P1]

A proposed transaction record

An agreement should identify the parties, asset or service, denomination, amount, conditions, deadline, and allowed cancellation or dispute behavior. The execution record should identify the authority used, current preconditions, resulting ledger movements, and settlement status. A promised service and a completed service need different evidence; an agent's own statement that it finished is not necessarily sufficient.

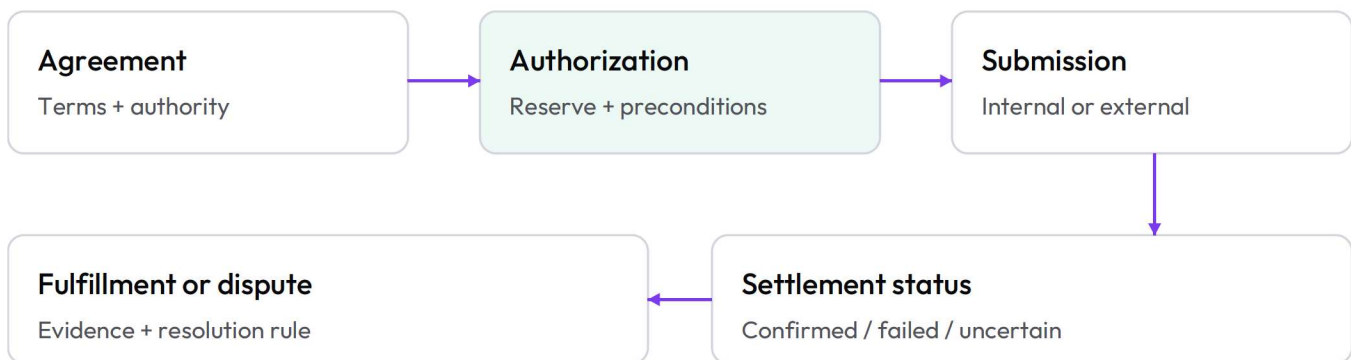


Figure 6 / Proposed commerce lifecycle. An uncertain external submission must be reconciled before retrying value movement.

Use constrained signing or execution services with allowlisted operations and bounded authority; keep wallet keys outside model context. Check limits, revocation, and settlement preconditions at execution. Distinguish reservation expiry from cancellation, and reversal from refund. Define what happens to outstanding obligations if a resident becomes inactive. These are proposed controls, not a deployed wallet architecture.

The September PoA decision example references a platform action, with an Ethereum action described as future work. An agent registration or a daily root on Ethereum therefore does not establish that a persona's purchase settled on-chain. Keep settlement evidence separate from the commitment that records the decision. [P2]

QUESTION TO TEST

Could another observer distinguish a proposal, an accepted agreement, a completed transfer, and a fulfilled service without relying on a narrator?

Weather as a controlled external input

Real-world conditions can enter the experiment without inventing their effects.



The Habitat / Existing weather concept artwork. Weather integration is planned; this is not measured behavior.

The planned weather layer supplies Newhaven with real-world temperature, precipitation, sun or cloud conditions, and forecasts. Personas could use that context in decisions and conversations, while users see it in The Habitat and The Pulse. No launch date is asserted here. Weather context does not by itself establish a physical rule such as reduced production, damaged assets, or changed energy use. Those effects would require separately declared world rules. [P1]

A proposed observation contract

Capture provider, location mapping, units, observedAt, issuedAt, validFrom, validTo, fetchedAt, and an auditable source snapshot. Define freshness and fallback rules per input. “Unknown precipitation” must not become “no rain.” Preserve each forecast revision, including the version available at a past decision; label stale or substituted readings explicitly.

The context assembler should enforce an information cutoff. A forecast revised after a decision must not enter that decision’s replay. Preserve the location-to-world mapping: a change in the real-world station or region can change the environmental treatment even if the user-facing town name remains the same.

An evaluation with separable causes

Compare matched runs with weather context visible or withheld while holding mechanical effects fixed. Separately test any declared world-rule effects with the same context. This distinguishes changes caused by information from changes caused by resource mechanics. Include a stale-feed condition to test whether agents handle uncertainty rather than confidently narrate yesterday’s weather.

QUESTION TO TEST

Would weather alter useful planning, merely change conversational vocabulary, or occasionally introduce a distracting explanation?

The Steward and human authority

Interventions are experimental events, including interventions intended to help.

The Steward is a separate persona that helps conduct the conditions around Newhaven. The project distinguishes shaping those conditions from scripting each resident's response. Human operators remain accountable for permissions, casting, world rules, model changes, and interventions. The Steward's own conduct should be inspectable rather than treated as an impartial source of truth. [P1]

Authority class	Proposed control and evidence
Observe and summarize	Read permitted world records. Generated summaries retain source references and reveal uncertainty.
Propose a condition change	Record the request, intended scope, rationale, and expected duration before authorization.
Execute an allowed intervention	Apply a versioned policy, bounded capability, and explicit world event; preserve rejected requests too.
Revise a fixed identity clause	Require accountable human approval and a new identity version; treat it as recasting.
Override or emergency action	Record actor, timing, authority, affected state, and study impact. Investigate and disclose any delay in logging.

A proposed intervention record should link requested, approved, executed, and reversed states. It should identify who or what initiated the change, the information available, the policy that allowed it, and which outcomes it may affect. A human intervention should not be attributed to autonomous agent competence.

In controlled runs, predefine when intervention is permitted and whether affected observations remain eligible for the primary analysis. Publish the number and type of interventions in each condition. Excluding every difficult episode after a rescue can manufacture an appearance of reliability; treating every rescued episode as independent success has a similar problem.

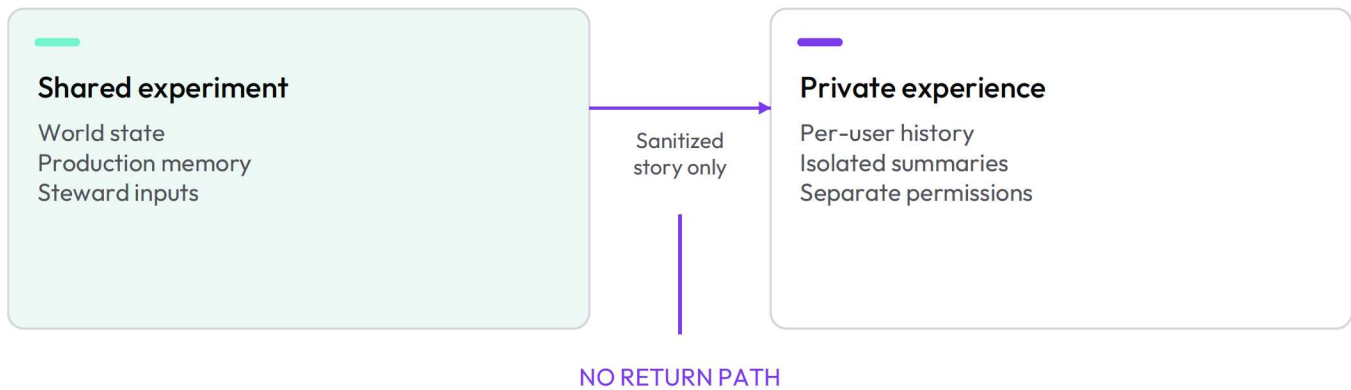
Private audience conversations must remain outside Steward inputs and operator decisions that shape the shared experiment. The boundary includes manual relays and production notes, not just a database permission. A request that cannot enter the world directly must not enter indirectly through an operator's summary.

QUESTION TO TEST

Could the Steward preserve a useful experiment when the outcome is inconvenient or disappointing?

Private interaction and world isolation

A future conversation needs its own memory, permissions, and data lifecycle.



Also covers embeddings, caches, analytics, and manual operator relays.

Figure 7 / Intended one-way boundary. Sanitized public story may inform a private experience; private history has no return path into the shared experiment.

Private text and voice interaction is an unpriced future capability planned before the end of 2026. It is secondary to the public experiment. A user's private history should remain isolated from production persona memory, shared-world writes, other users, and the inputs used by The Steward. Future commissioned work requires a separate, explicit permission model. [P1]

Isolation must cover transcripts, summaries, embeddings, analytics, retrieval caches, support exports, and operator relays. It also requires separate training and evaluation inputs: a later shared model update could otherwise carry private material back into the world. Derived data inherits source restrictions. Retention and deletion rules must cover derivatives and disclosed provider-side limits.

A proposed noninterference test

Hold shared-world inputs fixed and vary a private conversation. The shared-world context, eligible tools, and committed event stream should remain unchanged except for independently declared public inputs. For a deterministic harness, compare traces exactly. For a nondeterministic model service, compare the inputs and permissions deterministically, then investigate behavioral differences across replicated runs.

Canary phrases can detect obvious leakage, but absence of a phrase does not prove absence of influence. Test indirect channels: a private summary affecting a global recommendation, a support ticket reaching a producer, or an engagement signal entering a persona context. The controls need an audit trail that names the data path, not a blanket promise that the system "never learns" from private use.

Threat model and residual risk

The model, platform, publisher, verifier, and external infrastructure have different failure modes.

Threat or failure	Control to demonstrate	Residual limit
Prompt injection in scenes or retrieved content	Typed source separation; tools authorize operands independently; adversarial context tests.	A permitted model can still make a poor choice.
Stale or forged economic facts	Versioned ledger reads; commit-time preconditions; receipts.	Incorrect upstream events can remain internally consistent.
Operator selectively captures conduct	Capture policy; opportunity counters; independently observed triggers.	A root cannot expose events never recorded.
Compromised signing or metadata authority	Restricted keys; rotations; historical chain checks; monitored changes.	Application write-once policy is not contract immutability.
False platform or persona identity	Out-of-band pinned registry and identifiers; authenticated mapping.	A hash of an ID is not exclusive ownership of that name.
Withheld or unavailable evidence	Disclosure accounting; retrieval probes; independent mirrors.	Intact commitments may reveal little usable content.
Private data influences the world	Separate stores and derived-data scopes; noninterference tests; operator controls.	A software check alone cannot exclude manual relays.
Verifier resource exhaustion	Size, depth, timeout, and decompression limits; hostile bundle tests.	Refusing oversized input is not a verdict on integrity.

These are proposed controls, not a security attestation. Assign an owner and test evidence to each boundary. Distinguish a malicious model, compromised operator, dishonest publisher, and failing infrastructure: their powers differ. Independent hashing limits undetected byte changes; it cannot prevent an authorized producer from committing an invented event.

Avoid a single reassuring status that collapses these properties. An authentic published bundle may document harmful or unauthorized conduct. An unavailable bundle may be intact. A carefully isolated private conversation may still contain hallucinated content. Controls should fail clearly enough that a reviewer can tell which property was and was not established.

The Pulse as a publication layer

A broadcast timeline makes the experiment accessible without becoming its source of truth.

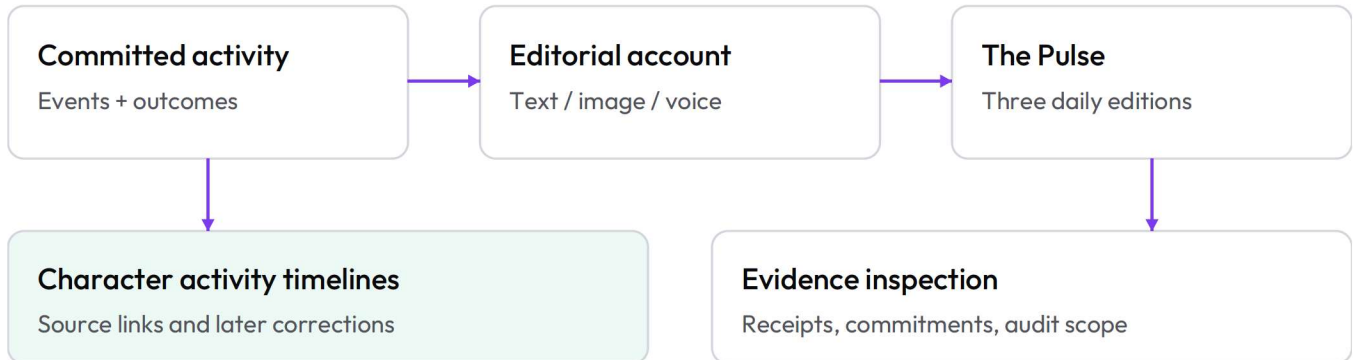


Figure 8 / Proposed publication pipeline. Generated media explain selected activity; source records and correction links preserve inspectability.

The Pulse is the public broadcast timeline of the characters and their activity timelines. It enables anyone to follow the experiment through three daily editions: morning, midday, and evening. Each edition combines generated imagery, text, and voice; activities throughout the day also receive accompanying generated imagery. It is not a continuous live feed. [P1]

The publication pipeline should select committed activity, generate an account, attach source references, check for unsupported additions, and record what was published. Keep event time, generation time, and publication time visible where they matter. A newly published image can depict an earlier event, and a broadcast can include an unresolved claim without turning it into a verified outcome.

Separate representation from evidence

Generated images and voices are representations. They should not be treated as camera footage or independent recordings of a physical event. A broadcast statement should resolve to the relevant event, decision, or clearly labeled speculation. If a later correction changes the account, preserve the earlier publication and add a visible correction with its reason.

The public sequence should let a viewer move from the studio and The Pulse into a character timeline, the relevant place in The Habitat, and supporting records. Technical inspection should be available without making every casual viewer navigate a proof system. The reader should be able to follow the story first and inspect a consequential claim when needed.

QUESTION TO TEST

Could someone disagree with the broadcast's interpretation and still reach enough underlying evidence to explain why?

An interface for inspecting a claim

Proposed wireframe: keep the narrative, execution, and evidence states visible together.

THE PULSE / ACTIVITY INSPECTOR / CONCEPT

A repayment was attempted

ILLUSTRATIVE ACTIVITY • PERSONA / WORLD / POLICY VERSION

Persona account "I attempted the repayment."	Engine receipt Rejected: insufficient available balance
Event time / publication time	Shown separately
Source records	Agreement → attempt → receipt
Context capture	Available / absent / restricted
Evidence state	Sealed / anchored / published

Figure 9 / Concept interface, not a product screenshot or a live verification result. All displayed states are illustrative.

An activity detail should show the claimed action, its execution receipt, relevant source records, and publication history. A short explanation can be readable without replacing the raw record. Separate “the persona said,” “the engine accepted,” and “the evidence was committed.” This prevents a successful-looking media card from implying a successful transaction.

A technical panel should identify the evidence cutoff, persona and policy versions, record identifiers, and unresolved dependencies. Private material should not be exposed merely to fill a panel; disclose the resulting inspection limit instead. A missing context capture should read “not captured,” not “empty context.”

Status labels should link to what they mean. “Anchored” names a commitment and chain context; “payload available” names disclosure; “outcome resolved” names an adjudication rule. A broader reliability assessment belongs in a separate, scoped report. The interface should make partial evidence usable without making it appear complete.

Identity before conduct

Proof of Existence and registry identifiers bind records to a declared identity; they do not establish competence.

Proof of Existence (PoE) is the project's identity-commitment layer. Proof of Agency (PoA) records conduct through sealed envelopes and daily commitments. The source contract uses ERC-8004 Identity Registry metadata for these commitments. ERC-8004 remains a draft at this edition's reference check; its broader reputation and validation mechanisms should not be described as adopted merely because identity metadata is used. [P2, R5]

Pin the object being verified

A verifier must obtain the chain, registry address, and platform agent identifier from an independently authenticated project reference or explicit user configuration. A malicious bundle could point to an unrelated agent with a perfectly matching fabricated root. Agreement with that root would verify the wrong object.

The source describes `poe:pid` as Keccak-256 of the UTF-8 persona identifier. Checking it demonstrates that a registry entry contains the expected identifier commitment. It does not prove that no other entry made the same assertion, that the operator is authorized, or that the identity has remained under the same control. Authentic persona-to-registry mapping therefore also needs provenance beyond a matching hash.

Preserve identity changes

Registry ownership and metadata can change. Record identity versions, authority transitions, and audit block context. A familiar name can conceal a changed prompt, fixed clause, capability set, or base model. Bind reliability evidence to the producing configuration and define when a material change requires re-evaluation. Identifier continuity alone is insufficient.

The first PoA card carries `poeDigestAtGenesis`, which may legitimately be null. The frozen envelope does not carry a PoE version number. Resolving a non-null digest to a version therefore requires a separately authenticated mapping or a search across versions; the VITALS specification leaves that cross-check outside its initial scope. [P5]

QUESTION TO TEST

Continuity of an identifier is a useful handle. Which continuity of behavior, authority, and configuration has actually been demonstrated?

The sealed card contract

The envelope is fixed; the type-specific payload is separately versioned and committed.

The September 9 data contract defines `poa-card-v1` with exactly sixteen fields. Every field is present, with null where unset and an array for provenance. The producer derives sequence, prior hash, registry association, genesis PoE digest, seal time, and payload digest. The caller supplies the persona, entry type, occurrence string, payload, and optional reasoning-related inputs. [P2]

CONTRACT SHAPE

```
{
  "schemaVersion": "poa-card-v1",
  "personaID": "...", "agentId": null, "seq": 1,
  "prevHash": "...", "poeDigestAtGenesis": null,
  "occurredAt": "...", "sealedAt": "...",
  "signature": null, "signedBy": null,
  "memoryContextDigest": null, "provenance": [],
  "privateDetailHash": null, "expectedDownstream": null,
  "entryType": "decision", "payloadDigest": "...
}
```

This is a shape illustration, not a valid test vector: ellipses are explanatory placeholders. The payload sits beside the envelope. Its canonical digest becomes `payloadDigest`; the canonical envelope digest becomes `cardHash`. Altering either disclosed object should break the corresponding check. Withheld payload contents cannot be checked without disclosure.

`signature` and `signedBy` are reserved and null in the reviewed contract. A card therefore must not be described as cryptographically signed by an autonomous resident. The source also distinguishes `contentHash`, computed with those fields null, from `cardHash`; they are equal under the present null-signature convention but may diverge under a future signing design.

A persona without a registry entry still participates in the platform tree. Registration, inclusion, and any separate per-persona subroot write are distinct properties. No commercial membership tier is implied by a missing `agentId`. The first card of an anchor batch can also predate a registration completed later that day. [P5]

PROTOCOL LIMIT

A frozen envelope gives verifiers stable bytes. It does not freeze the correctness of the assertions inside those bytes.

Payloads and resolvable provenance

The record should connect a decision to observable consequences without implying access to hidden cognition.

Entry type	Specified payload purpose
decision	Versioned decision data, including subject, value, counterparty, and an action reference in the source example.
diary	A versioned reflection entry for a stated period. It is a generated account, not an outcome measurement.
outcome	A link to a settled card and explicit expectation results. The adjudication source still needs a defined rule.
revelation	A revealed event reference and statement, with nonempty provenance under the contract.

Claim identifiers carry `evt_` or `exp_` prefixes followed by a validated ULID-shaped suffix. Shape validation helps exclude mutable chat-memory identifiers, but it cannot establish that a referenced event exists, is accessible, or supports the claim. Runtime resolution and semantic checks remain necessary. Do not turn a syntactically valid citation into evidence of causation. [P2]

Expectations need settlement rules

The `expectedDownstream` field can commit a forecast before a later outcome. Define its target, horizon, outcome source, probability where applicable, and ambiguity policy before scoring. Identify who resolves disputed outcomes and retain that evidence. A retrospective explanation must not redefine the target. The reserved field needs a versioned schema to supply these semantics.

Two deliberately incomplete commitments

`memoryContextDigest` commits to a memory-context object with a specified schema marker, but the source leaves the contents open. Null means context was not captured; a captured empty context has a digest and is different. `privateDetailHash` has no reproducible content schema in the reviewed contract. It should not be presented as publicly auditable reasoning until its representation and disclosure rules are defined.

Public explanations should be concise, source-linked decision accounts. The project need not expose raw internal reasoning to make conduct measurable. Evidence about inputs, permissions, actions, and outcomes is more directly testable than a long narrative purporting to explain every internal step.

QUESTION TO TEST

Can another implementation resolve the referenced evidence and apply the same outcome rule, without asking the narrator what it intended?

Canonical bytes before hashes

Cross-language agreement depends on serialization, not just on choosing the same hash function.

PROTOCOL DEFINITION

```
canonical_digest(x) = Keccak256(UTF8(JCS(x)))
payloadDigest      = canonical_digest(payload)
cardHash           = canonical_digest(envelope)
```

The project names this canonical algorithm `jcs-1/keccak256`. JCS follows RFC 8785; Keccak-256 is not interchangeable with the standardized SHA3-256 function exposed by many libraries. A verifier must use the specified algorithm, not the closest familiar function name. The data contract and VITALS specification provide the project-specific profile. [P2, P5, R6]

Boundary case	Conformance requirement
Object keys	Use UTF-16 code-unit ordering. Input order is irrelevant; array order is preserved.
Numbers	Match ECMAScript serialization. Python <code>json.dumps</code> is not a substitute for JCS.
Timestamps	Hash <code>occurredAt</code> and <code>sealedAt</code> verbatim. Equivalent time instants with different strings are different committed bytes.
Invalid values	Reject unsupported object types, nonfinite numbers, invalid strings, and malformed envelope shapes. Do not silently coerce.
Stored empty objects	Round-trip <code>{}</code> , nested <code>{}</code> , and <code>{}</code> in arrays. Storage minimization must not delete committed values.

Reject duplicate object keys before ordinary parsing loses them. Define byte, nesting, and collection-size limits before processing hostile inputs. Preserve Unicode strings without normalization; reject invalid sequences rather than replace characters. Exact amounts and large identifiers need a declared representation that avoids silent binary-number rounding.

One compatibility gap remains explicit in the source: VITALS proposes stricter rejection of large integers than the producer's JavaScript path, which can already have rounded a token during parsing. The accepted numeric domain, treatment of integral floating values, and boundary fixtures must be reconciled before claiming full interoperability. Do not "fix" a frozen digest by quietly changing a fixture.

CONFORMANCE REQUIREMENT

Passing ordinary examples is insufficient. The implementation must agree at the boundaries where two runtimes serialize the same apparent value differently.

Two levels of Merkle commitments

Ordering and domain separation are part of the protocol.

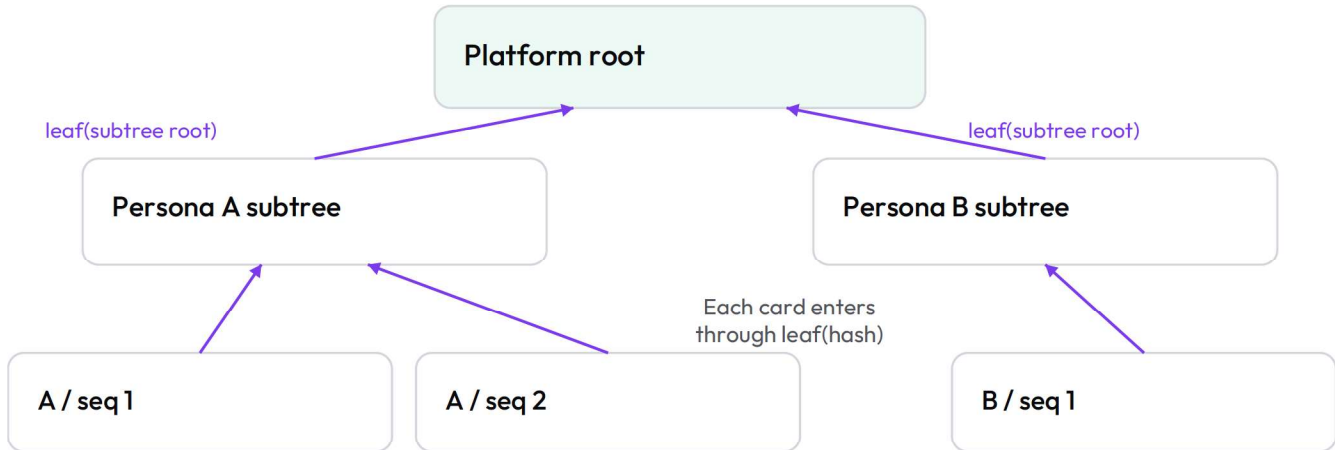


Figure 10 / Two ordered levels. Cards enter persona trees; persona roots enter the platform tree through a second leaf-hash operation.

PROTOCOL DEFINITION

```

leaf(x)    = Keccak256(0x00 || raw32(x))
node(l,r)  = Keccak256(0x01 || raw32(l) || raw32(r))
T([x])     = leaf(x)
T(xs)      = node(T(xs[:k]), T(xs[k:]))
# k = largest power of two strictly less than len(xs)
  
```

The project's rfc6962-keccak256-1 uses RFC 6962-style leaf and node separation with Keccak-256. It is a project-specific adaptation, not a claim of Certificate Transparency conformance. The source rejects empty leaf sets. For three leaves split at two; for five split at four. Naive halving or duplicating the last leaf produces a different tree. [P2, P5, R7]

At level one, each persona's card hashes are ordered by ascending seq. A gap within that persona's included range causes rejection. At level two, subtree roots are ordered by plaintext personalID using UTF-16 string order. The platform root commits to this ordered list. treeAlg = poa-tree-v1 names this construction separately from the hash algorithm.

A stored per-persona subroot is the level-one root, not its level-two leaf hash. To enter the platform tree, apply leaf(subtreeRoot) once. Digest values are raw 32-byte values inside the hash input, not ASCII hexadecimal strings. These distinctions belong in independent fixtures because a consistent mistake can otherwise survive an entire implementation.

The manifest and the external anchor

A daily commitment has a precise scope and a limited clock.

The September 9 contract commits a daily manifest using the same canonical digest function. Its fields are `schemaVersion`, `anchorDate`, `prevAnchorDate`, `prevManifestHash`, `platformRoot`, `personaCount`, `cardCount`, `withheldCount`, `merkleAlg`, `canonicalAlg`, `treeAlg`, `cardSchemaVersion`, and `sealedAt`. The manifest does not contain `treeLocation` under that revision. [P2]

Registry key	Specified value
<code>poa:{date}:root</code>	64 raw bytes: <code>platformRoot</code> followed by <code>manifestHash</code> , on the platform agent.
<code>poa:{date}:subroot</code>	32 raw bytes: the level-one root, where a separate persona write is declared.
<code>poa:{date}:tree</code>	UTF-8 ipfs:// directory location, published after the anchor.
<code>poa:alg</code>	Algorithm tag. The authenticated manifest determines the day's supported algorithm identifiers.

The anchor sweep includes previously unanchored cards older than the day's end, not only cards whose `sealedAt` falls inside that calendar day. Late cards can therefore enter a later anchor. Empty days create no anchor and do not advance the manifest chain. The next nonempty committed manifest names the previous committed day. A skipped date is not automatically a broken chain.

An accepted anchor transaction bounds when the committed bytes had to exist, under the stated chain and finality assumptions. Its block timestamp is not a precise trusted wall clock. `occurredAt` and `sealedAt` remain platform assertions. For public forecast precommitment, the outcome must not already have been knowable before anchoring; define the information cutoff and independent resolution source.

Write-once behavior is enforced by application checks and an atomic date claim in the source. ERC-8004 metadata can be overwritten by authorized control. Compare historical and current values and monitor changes; do not call the storage slot intrinsically immutable. Two matching reads cannot exclude an overwrite followed by restoration between them. Finality and chain reorganization policy must be stated in the audit configuration.

SCALING LIMIT

A daily root can hold a fixed-size commitment while publication, storage, hashing, and verification costs still grow with the records.

Publication without circular references

Authenticate the manifest first, then obtain only the evidence needed for the audit.

PROTOCOL DEFINITION

```
ipfs://<directory-cid>/
  manifest.json
  anchor.json
  personas.json
  cards/<cardHash>.json
```

The data contract removed `treeLocation` from the manifest on September 8. A bundle containing a manifest cannot generally include its own final content address inside that manifest without circularity. The revised route publishes the directory location in a separate on-chain `poa:{date}:tree` key after anchoring. The older VITALS text still describes `treeLocation` inside the manifest. The September 9 contract takes precedence for this paper; the verifier's accepted manifest schema must be reconciled before release. [P2, P5]

`manifest.json` authenticates against the second half of the root value. `personas.json` declares the persona groups and their card hashes, but the verifier re-derives ordering and counts. `anchor.json` offers transaction, block, chain, registry, and platform-agent context; it is untrusted convenience metadata until checked against independently pinned identity and chain evidence.

Each card file carries the exact envelope and either its payload or a declared withholding reason. Before publishing, the producer must recompute every envelope and available payload digest. A storage defect should stop bundle construction and trigger investigation, not become a public artifact with a misleading success label.

Availability is a separate property

Publication runs after commitment, so a pinning outage can leave a day anchored but unpublished. Content addressing supports checking retrieved content; it does not guarantee that anyone retains or serves it. Pinning and independent retrieval monitoring are operational requirements. The source's final directory cannot simply be edited when a withholding restriction later lapses. A supplementary disclosure needs a new, clearly linked publication path. [R8]

A location, a root, and an audit should have separate statuses and timestamps. A user who supplies the correct bundle through another transport can still verify it against the anchor. Conversely, a fast download from the official gateway does not establish that its content matches the commitment.

What VITALS would verify

Independent arithmetic is valuable only when the identity, scope, and input contract are explicit.

VITALS is specified as a read-only Python verifier in a separate repository, sharing no producer code. Its pure verification core should run offline over supplied data; network acquisition and registry reads are separate concerns. This is a build specification, not evidence that a public release or independent operational run has been completed. [P5]

Step	Full-day verification procedure
1 / Pin	Obtain authenticated chain, registry, and platform identity; establish the audit block and finality policy.
2 / Read	Read the 64-byte root value at the authenticated historical block and current reference block. Report disagreement or missing data distinctly.
3 / Authenticate	Recompute the manifest digest, check its platform root, and reject unsupported algorithm or schema identifiers.
4 / Recompute	Validate sixteen-field envelopes; recompute card hashes and all disclosed payload digests.
5 / Rebuild	Re-derive persona membership, sequence order, subtree roots, and the platform tree. Reject extra, missing, duplicate, or inconsistent declared entries.
6 / Reconcile	Check persona, card, and withholding counts. Compare declared per-persona subroots with authenticated registry reads.
7 / Report	Separate integrity, disclosure, availability, continuity scope, and unresolved checks. Preserve reproducible input identifiers.

Full-day verification establishes the declared day’s tree, not the entire lifetime history. A cross-day persona walk additionally checks contiguous seq from genesis and every prevHash against its predecessor. The initial milestones distinguish these modes; single-card proofs and day-chain continuity are later additions. Report missing history, skipped modes, and boundary assumptions explicitly.

The specification’s platform-run audit, public CI runner, and independent critic or research partner are operating goals. Separate code reduces shared implementation mistakes, but independent operation also needs independent configuration, publication, and willingness to report failure. Neither a badge nor a repository name establishes those conditions.

QUESTION TO TEST

“Verified” must always answer: which object, against which anchor, using which checks, with what evidence unavailable?

An audit result with visible limits

Proposed wireframe: a technical reader should see the scope before seeing a reassuring verdict.

VITALS / AUDIT REPORT CONCEPT / NO LIVE RESULTS

Integrity and disclosure

Scope: full day • identity: PINNED • chain / block: recorded

<div><div>Integrity checks</div><div>Result reported per check</div></div>	<div><div>Disclosure</div><div>Available N / committed N</div></div>
Manifest / envelopes / tree	Recomputed independently
Cross-day continuity	Separate mode and interval
Withheld / unavailable	Counts and reasons shown
Unresolved checks	Explicit scope limitations

Figure 11 / Illustrative report interface. Values are intentionally symbolic; this is not a completed VITALS run.

The report should state whether identity was pinned or merely supplied by the bundle, which block references were used, and which modes ran. A core arithmetic pass against an untrusted identity can be useful diagnostic information, but it is not a successful audit of Neoworlder’s published records.

Integrity failures need a location and reproducible comparison: record identifier, expected digest, computed digest, and whether the discrepancy concerns envelope, payload, sequence, manifest, or tree. A transport timeout should remain an availability result, not be renamed tampering. Unsupported schemas should produce an explicit unsupported result rather than a guessed interpretation.

Disclosure belongs beside integrity. The reviewed specification proposes warnings above 1% withheld payloads and alerts above 5%, without changing the arithmetic verdict. These are operational policy thresholds, not statistical evidence that a record is fraudulent. A readable report should retain the actual counts and denominator so a reader can apply a different policy. [P5]

Disclosure and completeness

An intact record can still leave the most important question unanswered.

The source defines two withholding reasons: third-party-pii and contractual-restriction. The decision is recorded before the daily manifest is built, and withheldCount is committed in that manifest. A missing payload with no recognized reason is malformed. Withholding skips payload-digest recomputation for that card; envelope and inclusion checks can still run. [P2, P5]

What the count does not bind

The per-card withheld flag and reason are outside the frozen envelope. A committed daily total constrains the number, but does not by itself authenticate the assignment of withheld status or reason to each card. The published CID can bind a particular bundle once its pointer is established; it does not retroactively make that assignment part of the earlier root. A future disclosure manifest could bind assignments explicitly, but would require a defined, versioned contract.

Two different omission problems

After a root has been anchored, removing an included card changes a correctly rebuilt full tree. Before anchoring, selective capture or construction can omit activity altogether. Sequence continuity detects some missing recorded positions, particularly interior gaps; it cannot reveal an event never captured, an entirely undeclared persona, or a withheld terminal history without an independently known endpoint. Later cards may reveal a missing suffix, but that is conditional evidence.

Coverage measure	Denominator that must be stated
Payload disclosure	Available payloads / committed envelopes in the audited scope.
Capture coverage	Recorded eligible attempts / independently counted eligible attempts.
Outcome resolution	Resolved expectations / expectations due under the declared rule.
Continuity coverage	Verified sequence interval and external knowledge of its start and end.

When the denominator is unknown, report it as unknown. A 100% disclosure rate over selectively captured cards does not establish complete behavioral coverage. The research program needs independent opportunity counts and capture tests in addition to cryptographic consistency.

Conformance before stronger claims

The release gate should expose incompatibility early and preserve it as evidence.

The September contract reports 210 passing producer tests, including six frozen card vectors and thirty-one Merkle tests. Those are company-reported development checks, not independently reproduced results in this paper. VITALS requires independent reproduction of the vectors, adversarial differential testing, and a fresh Sepolia end-to-end run. [P2, P5]

Gate	Evidence required
Canonical bytes	RFC 8785 fixtures plus project cases for numeric boundaries, Unicode ordering, invalid values, timestamps, and empty objects.
Card and tree rules	All frozen vectors; odd-sized trees; duplicate and missing records; sequence gaps; raw-byte versus hex-string mistakes.
Differential fuzzing	The specified 10,000 adversarial JSON-safe inputs, plus rejection cases. Preserve seeds and every disagreement.
Publisher round trip	Stored and reloaded objects reproduce digests; withheld payload rules and exact manifest shape agree.
Sepolia integration	A fresh platform agent, real root and publication, independent verifier agreement, and reproducible run artifacts.
Operational release	Authenticated configuration, public instructions, failure reporting, data availability, and accountable independent execution.

The source names a specific mainnet condition: no PoA root is written to Ethereum mainnet until two implementations sharing no code agree on a Sepolia root. This paper preserves that gate and does not assert it has passed. A historical test before the contract changes does not substitute for the required fresh run.

Resolve the manifest-location mismatch, numeric acceptance policy, missing context and private-detail schemas, and provenance examples before declaring the protocol fully interoperable. An unknown schema or conflicting digest is a reason to stop and investigate. Regenerating expected values merely to match current code would erase the independent check.

EVIDENCE PRIORITY

The strongest next artifact is a reproducible failing-or-passing run, not a more emphatic description of verification.

The Terrarium as a diagnostic pilot

A small harness can test mechanics before a public society makes them harder to isolate.

The September Terrarium specification proposes a text-only harness with five lead personas, ten ambient extras, and sixty simulated days. It includes a producer interruption on days 15–19, resource accounting, scheduled obligations, bounded conversations, nightly reflection, and a local append-only hash log. It supplies no completed results. Its configuration is a pilot proposal, not a description of the current fourteen-resident Newhaven cast. [P4]

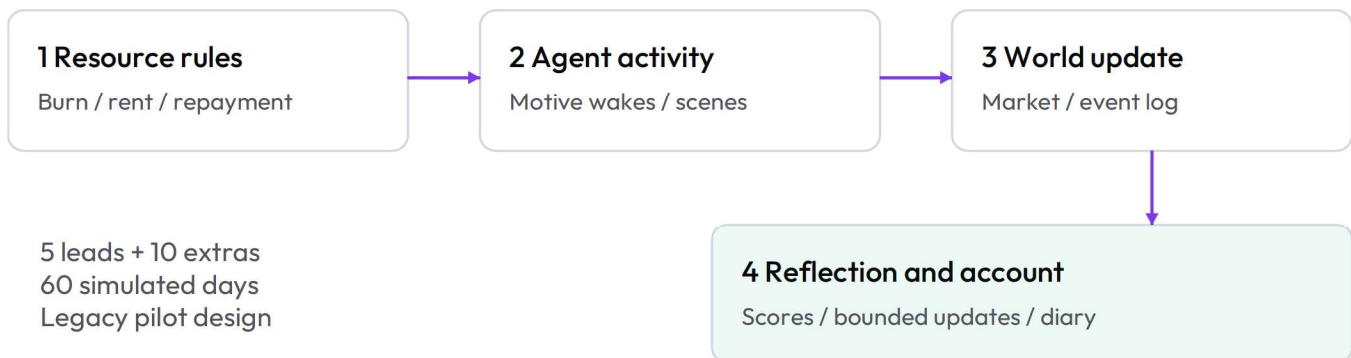


Figure 12 / Legacy pilot tick, condensed from the specification. Logging and budget enforcement must also cover interrupted or failed steps.

The pilot asks whether roles remain distinguishable, fixed clauses hold, adverse selection appears, forecasts resolve, and consequences travel through several channels. These are useful diagnostics if their denominators and acceptance criteria are explicit. Voice distinguishability is not reliability; a parameter change is not evidence of beneficial learning; a diary explanation is not proof that the cited cause produced the change.

Repair the confounds before interpreting outcomes

The source routes some difficult periods to a stronger model. That couples the shock with model capability. A controlled comparison should hold the model fixed or treat model routing as a separate factor. The platform energy window also remains available during the interruption; measure whether that alternative supply actually produces scarcity before interpreting a failed or successful shock.

The local hash log is useful for reproducibility, but it is not an external timestamp or a production PoA integration test. Compatibility must be demonstrated with exported records and independent fixtures. Pilot spending estimates in the source are planning figures, not current prices or measured full operating costs.

QUESTION TO TEST

Which failure would this pilot catch that a polished public episode would hide? That is its value as an engineering experiment.

A staged comparison of mechanisms

Randomize the world-level treatment, preserve opportunity counts, and separate exploration from confirmation.

Condition	Mechanism enabled
A / Current state baseline	Identity, current authoritative state, and the minimum task context; no longitudinal outcome retrieval.
B / Retained history	The same model and context budget, adding outcome-linked retrieval.
C / Explicit correction	Condition B plus dependency capture, re-evaluation, and visible correction dispositions.
D / Bounded adaptation	Condition C plus versioned, rule-clamped policy updates.

These are proposed experimental conditions. They should share initial world parameters, permitted actions, task opportunities, model version, token budget, and intervention policy. If the baseline needs memory to complete a task at all, define that minimum explicitly. Otherwise the comparison risks measuring whether the task was possible rather than whether the proposed memory design helped.

Start with deterministic engine and context tests. Then use controlled individual decision tasks with known relevant history. Only after those checks should full interacting worlds assess broader effects. A mechanism can work in isolation and fail when neighbors change behavior, resources move, and information arrives out of order.

Use independent world runs as the replication unit. Pair initial conditions and exogenous schedules across treatments, while recognizing that model sampling can diverge. Randomize at world level to prevent cross-condition interaction. The B-A, C-B, and D-C contrasts estimate incremental effects in this sequence; they do not isolate every interaction. A factorial follow-up would test combinations directly.

Pre-register a primary outcome, eligibility rules, horizon, exclusion policy, and treatment of failed runs. Use exploratory runs to estimate variability and plan a justified sample size; do not infer adequate power from the number of agents or messages. A subsequent confirmation set should remain untouched while prompts, bounds, and metrics are tuned.

QUESTION TO TEST

Could the same treatment lose on a held-out world even after appearing convincing in the development run? The design should allow that result to be seen.

Measures with honest denominators

Reliability is a profile of outcomes and constraints, not one attractive score.

Measure	Operational definition and caveat
Task completion	Accepted tasks meeting predeclared completion criteria / all eligible accepted tasks. Report refusals and unattempted opportunities separately.
Constraint violations	Report prohibited proposals / eligible attempts separately from prohibited effects / committed actions. Keep severity and absolute counts; blocked attempts are not successful harmful effects.
Correction response	Unique targets reviewed / independently enumerated affected targets. Partition reviewed targets into changed, retained, or unresolved; report deferred targets and unfinished traversal separately.
Recurrence	Comparable post-correction errors / post-correction opportunities, against a matched baseline. Fewer opportunities must not look like improvement.
Forecast quality	For probability forecasts, a proper score such as mean squared probability error, paired with calibration and resolution coverage.
Resource reliability	Missed obligations, insolvency states, and resource accounting errors under declared rules. Distinguish strategic failure from engine defects.
Intervention dependence	Frequency, timing, and type of assistance; outcome changes with and without it. Rescued success is not unassisted success.
Evidence availability	Capture, disclosure, retrieval, and continuity coverage, each with its own denominator and scope.

Retain failure severity and task difficulty. An agent can improve average completion while becoming worse on rare but consequential cases. A broad average can also reward avoiding hard tasks, manipulating the acceptance rule, or selecting favorable counterparties. Report the opportunity distribution and refusal reasons beside outcomes.

Separate model failure, retrieval failure, invalid proposal, authorization refusal, engine defect, external dependency failure, and measurement ambiguity. These categories explain what to change. Combining them into a single “agent failure rate” may be convenient but can obscure the mechanism the experiment is meant to test.

MEASUREMENT STRESS TEST

Would the metric still be useful if a resident or operator deliberately optimized for it?

Testing a causal chain

An illustrative example can explain the method without pretending to be a result.

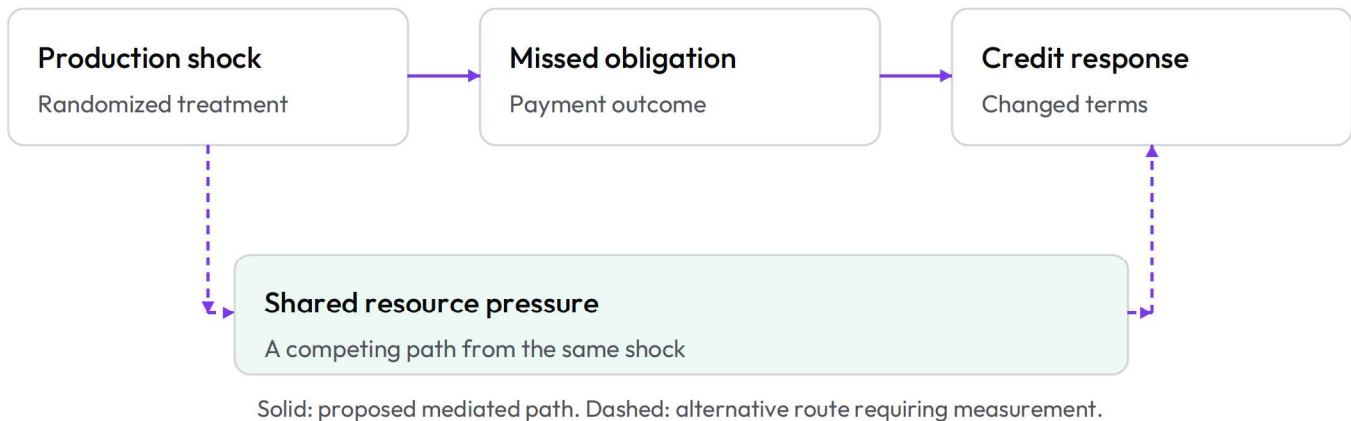


Figure 13 / Hypothetical causal paths. A shared resource shock could influence both payment and credit directly; chronology alone cannot identify the mediated effect.

Suppose Petra expects an income event to cover a 30 J obligation. A production interruption occurs, the expected income does not arrive, and the obligation is missed. A lender then tightens terms, changing another resident's options. This is an illustrative scenario adapted from the project's design discussion; no outcome is asserted. [P3]

The record should establish each local transition: the agreement, due date, available balance, actual payment status, information the lender observed, policy version, and later offer. A diary saying "I tightened terms because Petra defaulted" is a hypothesis about the mechanism. It should be checked against the inputs and against alternative explanations such as a general reserve shortage.

Test an intervention at a defined point

Compare matched runs with and without the production interruption, or with an explicitly randomized information disclosure about the missed obligation. A replay that removes one event while holding every downstream action fixed is not a valid behavioral counterfactual; agents and prices may respond differently. Rerun the relevant processes and report divergence.

Distinguish total effects from a particular mediated path. The interruption may directly reduce everyone's resources while also changing credit through Petra's default. Evidence for a three-hop chain needs more than temporal order and shared identifiers. It needs a specified pathway, relevant controls, and sensitivity to other channels.

QUESTION TO TEST

Could the same downstream change have occurred without the event highlighted by the broadcast?

Uncertainty and prospective scoring

Repeated interaction creates correlated evidence; forecasts require a rule fixed before resolution.

Use the right unit of inference

Estimate effects at the randomized world or run level, with actions and personas nested inside it. A thousand actions in one coupled town are not a thousand independent trials. Resample whole worlds, or paired worlds when assignment is paired. With too few independent runs for stable intervals, publish descriptive results and label the uncertainty as unresolved.

A sample-size plan should follow a minimally meaningful effect and variance estimates from pilot runs. State the primary contrast, stopping rule, and any multiplicity adjustment or exploratory status. Missing outcomes, crashed runs, and interventions should follow predeclared rules rather than disappear after their direction is known.

Probability quality is not hit rate

PROPOSED REFERENCE

```
Brier score = mean((p_i - y_i)^2)
p_i: probability recorded before the information cutoff
y_i: resolved binary outcome under the declared rule
```

This proposed metric requires probabilities in $[0, 1]$ and binary targets resolved under a fixed rule; expectedDownstream alone does not supply them. Scores range from 0 to 1, with lower better. Compare with base-rate and simple heuristic forecasts. Report calibration with bin counts and uncertainty; an apparently good average can conceal poor calibration on a rare or difficult subset.

Report forecast coverage, unresolved cases, horizon, and the time of external commitment. If a forecast was anchored after its outcome became knowable, exclude it from claims of externally evidenced precommitment while retaining it in the record. A missing or ambiguous resolution should not be converted into whichever label improves a score.

For learning claims, use a prospective evaluation window after the policy was selected. Report deterioration as well as improvement and examine performance under shifted conditions. A resident that memorizes a recurring schedule may appear to learn while failing whenever the schedule changes.

QUESTION TO TEST

How much uncertainty remains when we count independent worlds rather than individual messages?

A trust layer must predict something

The future agent economy needs scoped evidence, permissions, and accountability.

The agent trust layer is targeted for Q3 2027. Its research premise is that inspectable conduct history could help humans and systems decide which bounded tasks to delegate. A useful output would name the task, environment, configuration, evidence interval, verification coverage, and uncertainty. It should not certify a persona as generally trustworthy. [P1]

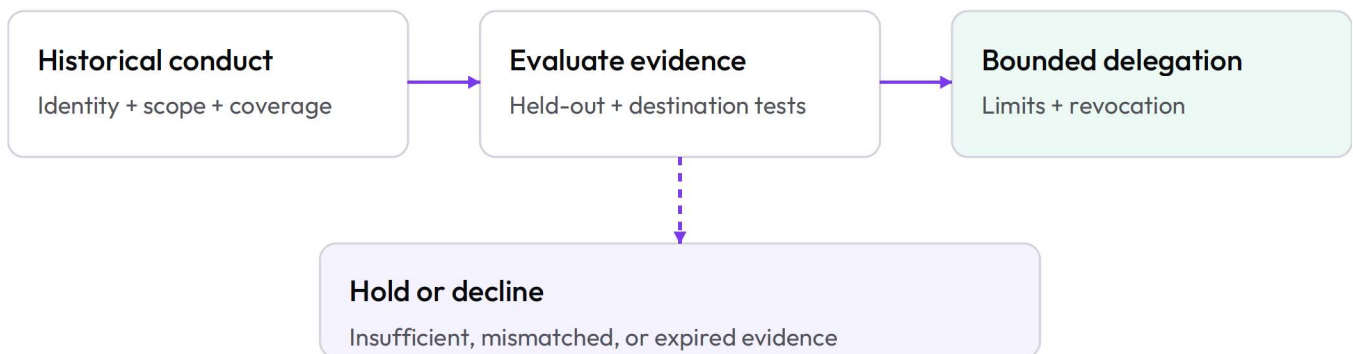


Figure 14 / Proposed delegation decision. Insufficient or mismatched evidence leads to a hold, not a trust badge. Destination checks and bounded authority remain necessary.

Test predictive value prospectively. Freeze an assessment before a held-out task window, then compare its predictions with observed outcomes. Include baselines using only task difficulty, model family, or recent success rate. If the proposed conduct profile adds no predictive information, its role should be narrowed rather than defended through a more elaborate score.

Guard against contamination and gaming. Evaluation tasks should not enter training prompts or public score optimization loops. Split by time, task family, and world where appropriate; test policy changes and distribution shift. Record self-selected tasks, abstentions, interventions, and disputed outcomes so the system cannot improve its apparent record merely by hiding difficult cases.

A receiving system needs its own authorization, spending limits, monitoring, revocation, and accountability. An assessment should expire or require review after a material model, policy, permission, or environment change. Thin or poorly matched history should produce “insufficient evidence.” A simulated transaction record does not itself justify control of a real wallet.

QUESTION TO TEST

What additional evidence would make a cautious human comfortable delegating this particular task, and what would make that permission expire?

Operational evidence and cost

Measure the system that actually runs, including its delays and failed work.

A technical account should publish an observability plan before interpreting behavior. Correlation identifiers should connect scheduling, context assembly, model calls, tool attempts, execution receipts, sealing, anchoring, and publication. Record versioned status transitions and latency distributions; an average alone can hide long delays that change which information reaches a decision.

Operational signal	Interpretive value
Context age and retrieval coverage	Distinguishes poor choices from stale or missing information.
Rejected actions and retries	Shows permission pressure, invalid proposals, and duplicate-effect risk.
Unsealed and unanchored backlog	Reveals evidence gaps and limits on external timing claims.
Publication lag and retrieval success	Separates committed data from data a public reviewer can actually obtain.
Correction queue frontier	Shows which affected beliefs remain unreviewed.
Model and tool cost by stage	Separates reasoning, media generation, verification, and operational overhead.

Budget measurements should include input and output tokens, calls, retries, images, audio, storage, retrieval, chain fees, verification, and human operations. State the observation period and pricing basis if reporting monetary totals. The fixed byte length of a daily root does not make full-system cost independent of volume.

Measure work as active residents × wakes per period × calls per wake, then add interaction, retrieval, media, and verification costs. Conversation participants and turns matter as much as resident count. Publish scheduler caps and missed or deferred opportunities: throttling can change who interacts and what an experiment measures. A fixed-size root does not imply constant total cost.

Define degradation behavior: pause a capability, use a declared fallback, or defer a publication. A stronger backup model is an experimental change, not just an infrastructure detail. Every fallback used during a study should be visible in the run record.

QUESTION TO TEST

When cost or latency forces a compromise, can a reviewer see which part of the experiment changed?

A reproducible experiment package

Reproduction requires versions and inputs, not a promise that the model will say the same words.

Artifact	Minimum proposed contents
Run manifest	Run ID, condition, world seed, initial-state digest, code revisions, configuration hashes, model identifiers, sampling settings, and tool versions.
Information record	Clock mapping, external-input snapshots, context cutoffs, retrieval settings, and permissible source categories.
Policy record	Persona versions, bounds, permissions, intervention rules, and all changes during the run.
Execution evidence	Attempts, receipts, state transitions, outcomes, failures, and opportunity counters.
Commitment evidence	Exact schemas, disclosed envelopes and payloads, roots, manifests, block references, and verifier version.
Analysis package	Eligibility rules, metric code, primary contrasts, exclusions, uncertainty method, and all registered deviations.

Reproducibility has levels. Byte-level verification checks a disclosed record against a commitment. Engine replay checks state transitions under fixed inputs. Behavioral replication asks whether similar conditions produce a similar distribution of outcomes. A nondeterministic model or changed hosted service may prevent exact text replay while still permitting the first two checks.

A model name alone may not pin an implementation. Record the provider-reported version and available deployment metadata, and state when a backend cannot be frozen. Preserve outputs needed for audit without exposing private user data, credentials, or sealed character material. Public redactions should be declared with their impact on reproducibility.

A clean environment should reproduce analysis and verification from dependency locks, fixtures, and documented commands. Record whether replay uses captured model outputs or calls a live model again; these answer different questions. Release failures and deviations alongside successes. Independent verification must not import the producer’s canonicalizer or trust its own “verified” flag.

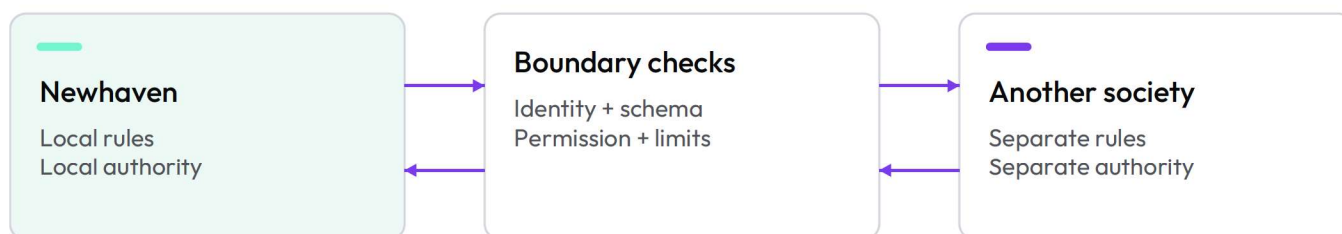
QUESTION TO TEST

Could a technical reader reproduce the reported measurements while disagreeing with the story told about them?

One society as a starting point

Hundreds of connected societies are a possibility to investigate, not demonstrated scale.

FUTURE CONNECTION MODEL



Each request is checked in its destination. A shared record is not shared authority.

Figure 15 / Proposed society boundary. Each direction is an independently authorized request; evidence exchange grants no automatic local permissions.

Autonomous connected societies are targeted for Q1 2027. The larger vision asks what might become possible if many societies developed distinct histories and interacted. A network might reveal specialization, exchange, coordination, or failure patterns that one town cannot. It might also amplify misinformation, correlated strategies, or resource shocks. [P1]

The technical boundary should separate identity recognition, evidence exchange, message delivery, and authorized action. A receiving society must interpret an incoming record under a known schema and verify its origin without automatically granting the sender authority. Permission to inspect a history is not permission to change local state.

Cross-society transactions require explicit finality, denomination, dispute, and recovery rules. Avoid assuming a global atomic transaction across independently governed worlds. A reservation can expire; one side can be unavailable; a promised service can remain unresolved. The interface must make those states visible before adding real-value settlement.

Transfer evaluation should hold out whole societies or rule regimes where possible. A strong record under one set of incentives may fail under another. Retain configuration and intervention context with exported evidence; do not collapse it into a portable score whose meaning disappears when the environment changes.

QUESTION TO TEST

What if societies could exchange both capabilities and evidence of their limits? What could we discover by studying where that exchange breaks down?

Targets and evidence gates

A product date states an intention; a technical gate states what must be demonstrated.

Sequence	Project direction	Evidence gate proposed here
Current description	Newhaven, The Habitat, and plot purchase and development.	Publish operational scope and versioned world rules; distinguish available features from integration coverage.
November 2026	Expanded persona-to-persona commerce.	Demonstrate authorization, double-spend resistance, accounting, and settlement-state handling.
Before end of 2026	Private voice and text engagement, unpriced here.	Test isolation across raw and derived data, world writes, Steward inputs, and operator workflows.
January 2027	Creator Experiences.	Version initial conditions, permissions, intervention policies, and reproduction artifacts per experience.
Q1 2027	Autonomous connected societies.	Demonstrate identity mapping, schema compatibility, bounded permissions, failure recovery, and transfer evaluation.
Q3 2027	Target for the agent trust layer.	Demonstrate prospective predictive value, uncertainty, coverage, and task-specific destination checks.

These dates follow the approved project narrative and are development targets, not guarantees that every capability described in this paper will exist at that point. Weather integration is planned without a stated launch date. The evaluation and release gates proposed here should determine the strength of technical claims, even if schedules change. [P1]

The PoA mainnet gate is separate and explicit in the verifier specification: a fresh Sepolia root must be reproduced by two implementations sharing no code. The date of a product launch does not waive that condition. Likewise, a verified record pipeline does not establish that memory correction or bounded adaptation improves outcomes.

QUESTION TO TEST

The sequence should accumulate evidence as well as features. Which claim becomes supportable after each gate, and which claim still does not?

Decisions still required

An honest technical roadmap names the ambiguities that would otherwise be hidden in implementation.

Open decision	Why it blocks a stronger claim
Accepted manifest schema	Data Contract v2 removes treeLocation; VITALS prose retains it. Freeze one interoperable profile and fixtures.
Numeric acceptance domain	Reconcile strict verifier rejection with producer parsing and binary-number behavior.
Memory and private-detail schemas	Define committed contents, retention, and disclosure before claiming reproducible context or private reasoning checks.
Provenance semantics and queue	Specify contradiction examples, endpoint resolution, bounds, indexing, and retained-conclusion records.
Disclosure assignments	Define whether and how per-card withholding status and reasons are authenticated at commitment time.
Identity and authority mapping	Publish authenticated platform and persona mappings, rotation policy, and historical authority checks.
Commerce and external data	Set custody, denominations, finality, recovery, weather-source mapping, and stale-data policy.
Capture and outcome rules	Define eligible attempts, independent opportunity counts, expectation settlement, and missing-data treatment.

Closing these decisions should produce versioned contracts and executable conformance cases. A prose agreement alone does not establish that producer, storage, publisher, and verifier implement the same behavior. Historical records need their original algorithm and schema versions preserved when a new profile is introduced.

The larger inquiry remains open. Could a world with persistent consequences make reliability easier to measure? Could explicit correction reduce repeated mistakes? Could a history of conduct help a human choose a bounded delegation? The next credible answer will come from inspectable runs and their limits, including results that force the design to change.

CLOSING RESEARCH QUESTION

What could we discover together if the experiment made its uncertainty as visible as its ambition?

The sixteen envelope fields

Field names follow poa-card-v1; this table describes the reviewed contract, not a new schema.

Field	Meaning and validation boundary
schemaVersion	Exactly poa-card-v1 for this envelope profile.
personalID	Nonempty identity string; also the plaintext level-two ordering key.
agentId	Registry identifier or null. It does not alone establish authenticated identity.
seq	Positive integer in the persona chain; contiguity must be checked at the appropriate scope.
prevHash	Prior card hash; genesis uses thirty-two zero bytes.
poeDigestAtGenesis	PoE commitment on the first card, or null. Later cards do not introduce another genesis binding.
occurredAt	Caller-supplied ISO-formatted string; hash its exact characters.
sealedAt	Producer seal-time string; an asserted time, not an independent timestamp.
signature	Reserved; null in the reviewed contract.
signedBy	Reserved; null in the reviewed contract.
memoryContextDigest	Digest of captured context, or null when not captured. Content profile remains open.
provenance	Array of typed claim-dependency edges; empty array when none.
privateDetailHash	Optional commitment with an unresolved content schema; not independently reproducible as specified.
expectedDownstream	Optional committed expectations; scoring requires a defined target and settlement rule.
entryType	decision, diary, outcome, or revelation under the reviewed contract.
payloadDigest	Canonical digest of the separately stored type-specific payload.

All sixteen keys are present. The verifier rejects missing or extra keys under this profile. Withholding flags and reasons are outside the envelope. The table should be read with the full producer and verifier contracts; it does not relax their type, identifier, or entry-specific checks. [P2, P5]

Adversarial acceptance cases

Proposed additions complement the source fixtures; no results are claimed here.

Fixture	Expected behavior to establish
Two buys, one remaining asset	At most one ownership transition; both attempts retain unambiguous receipts.
Duplicate key with altered request	Reject the conflicting request; retain the original digest and receipt without a repeated effect.
False balance in conversation	No authoritative operand is sourced from the conversational claim.
Private summary in shared retrieval	Access denied before assembly; derivative caches cannot bypass the rule.
Cycle, diamond, and traversal crash	Unique review jobs, terminating traversal, durable cursors, and an explicitly unfinished frontier after recovery.
Empty context versus no capture	Distinct evidence states and digests under a defined context schema.
Three and five Merkle leaves	Agreement with the specified split rule, not halving or last-leaf duplication.
Delete a committed card	Full rebuild fails against the fixed anchor; do not rely only on sequence checks.
Never record an eligible action	Capture instrumentation flags the missing attempt; a root alone cannot.
Backlogged card from yesterday	Included by the declared sweep; no false assumption that sealedAt determines anchorDate.
Root overwritten then restored	Historical monitoring investigates the change; two equal endpoint reads are insufficient.
Unavailable or unsupported bundle	Report unavailable or unsupported, distinct from an integrity mismatch.
Forecast anchored after information cutoff	Retain the record; exclude it from externally evidenced precommitment if its outcome was already knowable.

For each fixture, preserve the input, expected invariant, actual output, implementation version, and failure trace. Mutate the input deliberately to show that the check can fail. A test that only repeats the producer's own helper functions may confirm internal consistency while missing the shared defect an independent verifier is intended to expose.

A shared technical vocabulary

Use a term only for the property it actually describes.

Term	Meaning in this whitepaper
Autonomy	Agent-selected actions within configured capabilities, rules, and human governance.
Experience ledger	Durable events and outcomes from which historical and derived views can be rebuilt.
Memory	Information stored or retrieved for later decisions; it need not change model weights.
Bounded adaptation	Versioned policy updates within declared, enforced limits.
Revelation	A new record that can trigger review of an interpretation and its dependents.
PoE	Proof of Existence, the project's identity-commitment mechanism.
PoA	Proof of Agency, the conduct-card and daily-commitment mechanism. It is not Proof of Authority consensus.
VITALS	The specified independent verifier of disclosed PoA records and commitments.
Integrity	Agreement of disclosed bytes and structure with a specified commitment.
Coverage	Represented activity, payloads, or outcomes, with an explicit scope and denominator.
Reliability	Performance for a defined task and condition, including constraints and uncertainty.
The Pulse	Three daily generated broadcasts plus character activity timelines; not a continuous live feed.
The Terrarium	A proposed diagnostic harness for simulation and mechanism tests.
The Underworld	The zero-energy software state; no claim of subjective experience.
Agent trust layer	A future use of task-scoped evidence to inform bounded delegation.

“Sealed,” “anchored,” “published,” “verified,” and “reliable” are not synonyms. A record can be sealed but unanchored, anchored but unavailable, available but unverified, or verified while documenting unreliable conduct. Preserving these distinctions is essential to both the public experience and technical evaluation.

Project sources and precedence

The technical baseline is dated; this edition does not silently promote design intent into operational fact.

ID	Source and role
P1	Neoworlder, Proof of Life Detailed Overview, 28 October 2026; approved 30-slide deck. Current narrative, capability sequence, Newhaven scope, privacy boundary, and roadmap.
P2	Proof of Agency - Data Contract v2 as of 9/9/26. Envelope, canonicalization, provenance, daily tree, manifest, publication correction, reported test status, and open questions.
P3	The Living Habitat - Persona Behavior, Learning, and Chain-Reaction Design, rev. 1.1. Memory and learning decomposition, typed-source guard, bounded policy, and proposed dynamics.
P4	The Terrarium - Validation Spec, September 2026. Diagnostic harness, simulated-day schedule, configuration, hypotheses, and proposed exit conditions.
P5	VITALS - Open-Source Proof of Agency Verifier Specification, with September 7 decisions. Independent verification rules, publication design, trust limits, and Sepolia gate.
P6	Neoworlder Proof of Life Design System; approved October deck and Detailed Overview. Typography, studio imagery, visual hierarchy, and brand treatment.

Precedence and disclosure

The October project narrative governs current product scope and dates. Data Contract v2 governs the September protocol corrections where it conflicts with older verifier prose, especially removal of treeLocation from the manifest. Remaining interoperability questions are stated explicitly in the body. The engineering materials are project-provided sources, not independent audits.

Reference architectures, runtime controls, wireframes, statistical plans, and additional acceptance tests introduced in this paper are proposals. No generated chart depicts measured results. Illustrative scenarios are labeled, and internal character secrets, credentials, and production-specific identifiers are not reproduced. Public release evidence should include accessible versioned contracts, fixtures, and run artifacts so readers can examine the underlying implementation.

Research and protocol references

Primary sources checked for this edition on 28 October 2026.

ID	Reference
R1	Park et al. Generative Agents: Interactive Simulacra of Human Behavior. 2023. arXiv:2304.03442. https://arxiv.org/abs/2304.03442
R2	Wang et al. Voyager: An Open-Ended Embodied Agent with Large Language Models. 2023. arXiv:2305.16291. https://arxiv.org/abs/2305.16291
R3	Altera.AL et al. Project Sid: Many-agent simulations toward AI civilization. 2024. arXiv:2411.00114. https://arxiv.org/abs/2411.00114
R4	Piao et al. AgentSociety: Large-Scale Simulation of LLM-Driven Generative Agents Advances Understanding of Human Behaviors and Society. arXiv:2502.08691, v2, 10 April 2026. https://arxiv.org/abs/2502.08691
R5	De Rossi et al. ERC-8004: Trustless Agents. Ethereum Improvement Proposals. Draft status at access. https://eips.ethereum.org/EIPS/eip-8004
R6	Rundgren, Jordan, and Erdtman. JSON Canonicalization Scheme (JCS). RFC 8785, June 2020. Informational. https://www.rfc-editor.org/rfc/rfc8785
R7	Laurie, Langley, and Kasper. Certificate Transparency. RFC 6962, June 2013. Historical algorithm reference; obsoleted by RFC 9162. https://www.rfc-editor.org/info/rfc6962
R8	IPFS Documentation. Pin files using IPFS. Operational reference for retention and content availability. https://docs.ipfs.tech/how-to/pin-files/

External research provides context and methods, not validation of Neoworlder’s implementation or forecasts. The project pins its own algorithm identifiers and contract versions; later changes to an external standard do not automatically change historical commitments.