GhostDrive: A Privacy-First Chain for Sovereign AI Memory and Verifiable Learning

Roma Nebo

Technical White Paper · November 2026 · v2.2

Abstract. Machine intelligence now depends on private knowledge — personal memory, licensed corpora, regulated enterprise records — yet the infrastructure that stores that knowledge cannot say who may compute on it, whether a model actually consumed it, or what the model learned. Encrypted storage protects bytes but leaves an *encrypted-data-to-AI gap*: the moment data becomes useful to a model, it leaves the owner's control. This paper states its premises before its mechanisms. The reference deployment is a self-hosted, open-weight model operated by a named GhostDrive operator entity on disclosed hardware; receipts for training and retrieval exist only inside that boundary, and external API inference is supported with explicitly weaker evidence. The deficiency the system addresses is specific: four classes of data that public pretraining distributions cannot contain — records that cannot leave a compliance boundary, licensed material withheld from crawls, facts newer than the model's cutoff, and corrections of documented model failures. We propose GhostDrive, a seven-layer architecture that closes the gap with a verifiable *knowledge-use transaction*: client-side encryption over a Merkle DAG (L1–L2); an untrusted swarm under measurable service proofs (L3); a policy engine that releases one-use keys only to a declared model, purpose and runtime (L4); time-aware, versioned knowledge that agents may propose to revise but never silently rewrite, admitted only through a formal LEAN qualification — Lineage, Evidence, Ablation, Novelty — in which acceptance precedes any compensation (L5); signed receipts on an evidence ladder from *submitted* to *settled*, with recursive zk-SNARK checkpoints verifiable in constant time (L6); and the DRIVE economy, 69.3% of which pays for verified storage, computation and retrieval, and 3.69% of which pays for admitted knowledge only after a controlled experiment shows the pipeline obtains a model improvement at a cost no higher than fine-tuning, distillation or retrieval on conventionally acquired data (L7). We report measurements from the browser prototype of the data and proof layers, define the comparative experiment and the cost-per-accepted-improvement analysis the AI layers must pass, and compare the design with Filecoin, Arweave, Storj and cloud storage on capability and on price ($3.69 per TB-month, three replicas).

1. Introduction

Data has become the scarcest input of machine intelligence, yet the infrastructure that stores it inverts its ownership. Cloud providers hold plaintext, platform terms grant broad training licenses, and creators cannot prove that a model was — or was not — trained on their work. For an AI system the problem is deeper than custody: a model that cannot read a document cannot learn from it, so every "private AI" product today resolves the conflict by decrypting data inside an operator it asks the user to trust. The owner's control ends exactly where the model's utility begins.

Decentralized storage networks address availability and censorship but not this gap. Filecoin, with IPFS content addressing, makes data public by default unless the client encrypts it, and proves that *some* bytes are stored, not *whose*, for *what purpose*, or whether a model consumed them; Arweave permanence is incompati-

ble with the right to erasure; none binds stored knowledge to the models it produces. A content hash proves byte integrity. It does not prove author rights, factual correctness, permitted AI use or that training improved anything.

What is needed is a system in which (a) no party other than the owner can read data by default, (b) a model may compute on it only under an explicit, expiring, purpose-bound grant, (c) knowledge is time-aware and correctable without rewriting history, (d) every use yields evidence an independent party can check at constant cost, and (e) economic reward follows that evidence rather than volume or assertion. GhostDrive achieves this by separating *confidentiality* (client-side encryption with device-held keys), *authorization* (a policy-gated key-release transaction), *temporality* (versioned living knowledge), *integrity* (content addressing and recursive zero-knowledge checkpoints in the style of Mina [4]) and *settlement* (capped proof-of-utility rewards). The paper is ordered so that premises precede claims: §2 fixes the deployment, the operator, the deficiency being corrected and the alternatives against which the design must win; the layers follow; the evaluation, cost model and open questions close.

> **EVIDENCE DISCIPLINE** Layers L1–L3 and the succinct chain of L6 are implemented and measured (§12). L4, L5 and L7 and the knowledge-use transaction are design proposals carried from the GhostDrive 369 research programme [10–12]; no audit, legal opinion or running GPU attestation is implied. Figures and worked examples are illustrative unless marked as measured.

2. Premises: Deployment, Operator, Deficiency and Alternatives

Four premises determine which guarantees in the rest of this paper are real. They are stated here, before the architecture, because an architecture that is true under one deployment is false under another.

2.1 Reference deployment and operator

The reference deployment (mode A) is a *self-hosted, open-weight model* — the candidate baseline is a Qwen3-8B-class model, pinned by family, tokenizer and weights hash — run by an explicitly designated *GhostDrive operator entity* on disclosed hardware. Before launch the operator publishes its legal entity and jurisdiction, the cloud or on-premises owner of the GPUs, model identifiers, sub-processors, audit rights and recovery arrangements. The operator controls GPU scheduling, the model registry, training-code release, evaluation and security logs. It does not own upstream model weights and has no administrative access to any external provider. Training and retrieval receipts (§9) exist only inside this boundary: GhostDrive can instrument what runs on hardware it controls and nothing else. Federated (B) and confidential-compute (C) modes extend the boundary with their own evidence; external API inference (D) is a product feature with explicitly weaker evidence — a logged call, latency and result — and never a training receipt.

2.2 What the trust boundary is, and is not

Client-side encryption removes storage and transport operators from the set of parties that can read data. Authorized compute does not remove the compute operator from that set: in mode A the contributor trusts the operator's job, the attestation or code measurement of that job, and the authorization process, for the duration of one key release. The claim this paper makes is therefore bounded. Relative to a cloud AI product, the readable set shrinks from {cloud operator, platform, every sub-processor, indefinitely} to {owner, one disclosed operator job, under a one-use expiring key, with a signed receipt}. Relative to a purely local deployment on the

owner's own device, GhostDrive offers *no* privacy improvement for a single user and does not claim one; its advantage begins where a local deployment cannot serve — teams, enterprises and labs whose data must be computed on by a party other than the owner, under rights the owner sets and can audit.

2.3 The deficiency the contribution pipeline addresses

An open-weight model has already been trained on the public distribution. Accepting material that is already in that distribution adds nothing and is scored as zero novelty by construction (§8.1). The pipeline exists for four classes of data that the public distribution *cannot* contain, each tied to a measurable model failure:

- **Boundary-locked records** — regulated enterprise, clinical, legal or defense data that cannot leave a compliance perimeter; the failure is domain performance on tasks the operator's customers actually run.
- **Withheld licensed material** — corpora whose rights holders exclude them from crawls but will license them under purpose-bound, receipted terms; the failure is coverage of that corpus's domain.
- **Post-cutoff temporal facts** — observations newer than the model's training cutoff; the failure is stale answers, served first through RAG and only later, if at all, through training.
- **Corrections of documented failures** — held-out test cases the model demonstrably gets wrong; the failure is the test case itself.

A submission that does not target one of these four classes has no admission path. In particular, a summary of public literature produced with another AI system targets none of them and is rejected at the lineage stage (§8.1) regardless of its apparent sophistication.

2.4 Alternatives the design must beat

GhostDrive is not an alternative to continued pretraining, supervised fine-tuning, distillation or retrieval augmentation; it is the rights-cleared, receipted *supply and provenance layer* for those methods, and it uses them as its training mechanisms. The question is whether obtaining data through the pipeline yields a model improvement at a cost no higher than obtaining equivalent data conventionally (Table 1). Knowledge Mining rewards (§10) do not open until that comparison has been run and published (§12, gate G5).

Table 1: Conventional improvement methods and what the pipeline adds to each.

Method	What it needs	What the pipeline adds	Pipeline not needed when
Continued pretraining	Large in-domain corpus with training rights	Rights-cleared access to boundary-locked and withheld corpora; receipts per batch	Corpus is public or already licensed
Supervised fine-tuning / adapters	Curated examples targeting a failure	Admitted corrections with lineage and ablation evidence	Operator can author examples in-house cheaper
Knowledge distillation	Lawful teacher outputs; validation set	Nothing for the transfer itself; provenance of the validation set only	Always — distillation is the first thing to try where contractually permitted
Retrieval augmentation	Fresh, trusted index	Temporal versions, consent-scoped indexes, receipts for retrievals	Data is public and freshness is not critical
Contracted research / data vendors	Budget, contracts, review	Open admission with the same evidence standard, settlement in DRIVE	Cost per accepted improvement is lower by contract (§10.2)

2.5 Why a token economy at all

A private self-hosted AI with secure memory does not require a token, and this paper does not claim otherwise: layers L1–L6 run without DRIVE. The economy exists for the network of parties the operator does not own — storage providers, independent verifiers, provers and rights holders — and 69.3% of it pays for measurable services (storage, compute, retrieval) whose value does not depend on any judgment about knowledge quality. The knowledge-contribution economy is 3.69% of supply, is gated behind the comparative experiment, and is treated throughout as the unproven part of the design.

3. System Architecture

GhostDrive is organized as seven layers with strict downward-only dependencies (Figure 1). The count follows the engineering discipline of separation of concerns rather than any network reference model: each layer represents a transition of *security ownership* — who may read, who may compute, who may claim — not a protocol stratum. Every layer is independently replaceable: the client runs entirely in a browser (IndexedDB + WebCrypto); distribution and incentives degrade gracefully when peers are offline; a single client with no network is still a fully functional encrypted vault with a local AI memory.

L7 — DRIVE Economy	Proof-of-utility settlement · capped vaults · knowledge mining · bounded DAO
L6 — Evidence & Safety	Receipts · evidence ladder E0–E6 · recursive zk-SNARK checkpoints · appeals
L5 — Living Knowledge & AI	Temporal versions · RAG snapshots · gated training · agent proposals
L4 — Authorized Compute	Policy engine · one-use key release · attested / self-hosted model runtime
L3 — Distribution & Service	WebRTC swarm · Kademlia DHT · storage, retrieval and compute proofs
L2 — Encrypted Knowledge Fabric	AES-256-GCM client encryption · content-defined chunking · Merkle DAG · CRDT metadata
L1 — Sovereign Client & Identity	Device key root · owner capabilities · delegated agents · session auth

Read bottom-up for capabilities; top-down for policy and settlement feedback. Shaded layers are the AI-specific additions over the v1.0 five-layer design.

Figure 1: Layered architecture. Trust decreases upward: L1 requires trusting only local cryptography; L7 trusts only adjudicated evidence from L6.

The design follows five invariants. **I1 (zero plaintext egress):** bytes leave the client only after AES-256-GCM encryption; the owner key root is generated on the device and never transmitted. **I2 (verify-don′t-trust):** every chunk received from any source is re-hashed and compared to its content identifier (CID) before acceptance; a malicious peer can waste bandwidth but never inject data. **I3 (no capability by possession):** a storage peer holding ciphertext, or a store-only grant, cannot become a read or training capability without a new, explicitly authorized event. **I4 (no silent learning):** no agent may alter model weights, rewrite a knowledge version

or spend rewards without the rights holder's recorded consent; updates append, never overwrite. **I5 (succinct auditability):** global state — which knowledge exists, its rights scope, its lineage and which jobs consumed it — is committed into a recursive proof whose size and verification time are constant in the number of entries.

The architectural rule that enforces these invariants across layers is that each layer L defines an accountable controller, an inbound artifact, an executable decision, an outbound artifact and evidence sufficient to contest the decision (Table 2). An operator cannot infer a higher-layer property because a lower layer succeeded: storage integrity does not prove permitted training; job execution does not prove model improvement; a token transfer does not prove contributor anonymity.

Table 2: Layer contracts and responsible parties.

No.	Layer contract: input → output	Responsible party	Hard boundary
L1	User intent / identity → constrained owner capability	Owner + delegated agent	Upload grants no training right
L2	Permitted data → encrypted object / version commitment	Client + authorized repository	Identifier reveals no plaintext
L3	Ciphertext request → measured availability or delivery	Storage, network and measurement nodes	Possession is not permission to read
L4	Grant + compute request → gated execution session	Policy engine + declared AI operator	No standing decrypt-all capability
L5	Authorized inputs → temporal knowledge / model artifact	Knowledge service + AI operator	No inference about external providers
L6	Runtime artifacts → validated evidence, challenge result	Verifier committee / auditor	Cannot create its own payout rights
L7	Accepted evidence → capped payable claim, policy feedback	Reward contracts + bounded DAO	Cannot override rights or fixed supply

4. L1 — Sovereign Client and Identity

The client is a browser, desktop or mobile application, an enterprise gateway or an owner-authorized local service. Identity, encryption and authority are deliberately separate. The *owner encryption root* is a high-entropy key generated on the device and protected by the operating-system keystore or hardware security where available; per-object random data-encryption keys (DEKs) are wrapped to authorized recipients. A wallet (secp256k1) is a payment and governance identifier, *not* the master encryption key and not proof of a unique person — deriving every object key from a reusable wallet signature, as in v1.0, is retired because it couples the entire vault to one signature oracle.

Sessions are domain-bound, nonce-bearing and short-lived (signature over a nonce, origin-bound, 7-day expiry); append-only ledgers — audit log, model versions, checkpoints — are writable exclusively by service-role backends after session verification, closing spoofing and spam paths at the schema level. An *AI agent is an accountable delegate* acting under a named principal, not a second sovereign rights holder: its capabilities are narrow (list/read a designated vault, propose an update, submit an opt-in record, query payout status) and new training permissions or risky tool executions always require owner consent. Login, dataset record, reward entitlement and payout wallet use distinct identifiers so that on-chain settlement does not map to private content.

L1 emits an `OwnerCapability{owner_key_id, agent_id, object_scope, allowed_actions, purpose, expiry, nonce, revocation_epoch, signature}`. Recovery is user-controlled and documented; device loss must not silently become loss of all data, and recovery never places unencrypted keys in operator custody.

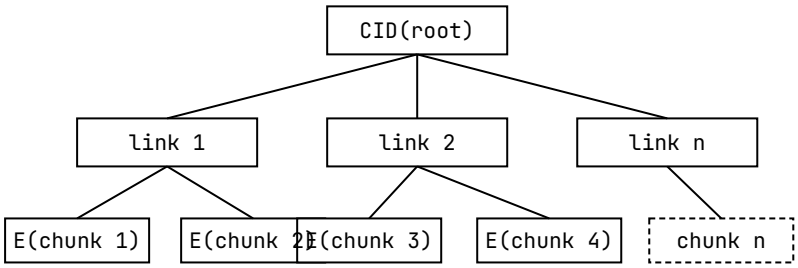
5. L2 — Encrypted Knowledge Fabric

5.1 Content-defined chunking

Files are split by a Gear rolling hash [6] with a deterministic 256-entry random table (xorshift32-seeded), average chunk size 256 KiB, bounds 64 KiB–1 MiB. Because boundaries depend on content rather than offsets, an insertion early in a file re-chunks only its local neighborhood; all other chunks keep their CIDs and are deduplicated across versions of the same owner's vault.

$$\mathrm{CID}(c) = \mathrm{SHA\text{-}256}\left(E_k(c) \right) \tag{1}$$

Each ciphertext chunk is addressed by its hash (Eq. 1); chunks assemble into a Merkle DAG whose root CID authenticates the file (Figure 2). Any single chunk can be verified against the root with an O(log n) path — no trust in the sender is required.



Leaves are ciphertext chunks; every edge is a SHA-256 commitment. CID(root) authenticates the entire object version.

Figure 2: Merkle DAG over encrypted chunks. Random-nonce AES-GCM yields different ciphertext for equal plaintext, so a public CID proves byte integrity — not plaintext equality, author rights or permitted AI use.

5.2 Key management and deduplication

Encryption is AES-256-GCM with a random 96-bit nonce per chunk; GCM tags give per-chunk authenticity on top of CID integrity. Per-object DEKs are wrapped under the owner root (and under each recipient's public key for shares), so sharing one object never exposes another and no server-side escrow exists by design. Because nonces are random, a public CID does *not* provide cross-user deduplication or safe disclosure of plaintext equality; private per-owner deduplication uses a keyed commitment over plaintext chunks held in the owner's encrypted index. Convergent encryption is deliberately avoided: it carries confirmation-of-file and low-entropy disclosure risks that are unacceptable for personal memory and regulated records.

5.3 Knowledge objects, versions and rights

Above chunks, the fabric stores *KnowledgeObjects*: an opaque identifier, immutable version identifiers, encrypted content, encrypted owner/rights metadata, optional relationships and cryptographic commitments to a declared snapshot (Table 3). Updates append new versions; old snapshots are never rewritten. File names, folder trees and share grants are last-writer-wins CRDT registers [8] replicated opportunistically between devices and encrypted like content; the network sees only ciphertext blobs and vector clocks, which is what allows GhostDrive to be local-first.

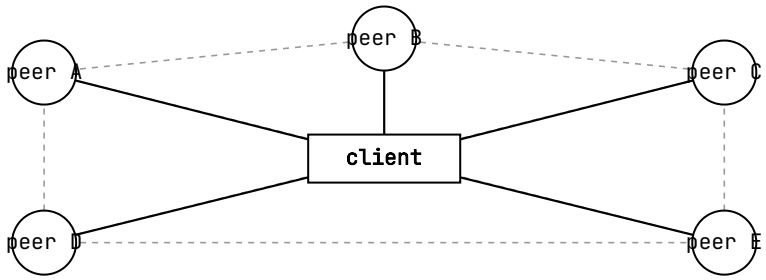
Table 3: Immutable artifacts of the knowledge fabric.

Artifact	Core attributes	Safety invariant
KnowledgeObject	opaque_object_id, owner_scope, format, classification	Identifier reveals no plaintext or stable wallet
KnowledgeVersion	version_id, parent_version, observed_at, valid_from/to	Updates append; old snapshots never rewritten
EncryptedPayload	cipher_id, nonce, tag, encrypted_chunks, object_wrapped_key	Keys stay outside storage-operator privilege
RightsGrant	license, purposes, processor, expiry, revocation_epoch	No training by default
SnapshotCommitment	Merkle root, format, schema version, retention	Commitment authenticity is distinct from semantic truth

Every object is in one of four states. **PRIVATE:** individual vault and local memory; no reward and no training right by default. **SHARED:** purpose-restricted access for named collaborators without implicit collective-training consent. **COMMONS_CANDIDATE:** the owner explicitly opts in, attaches a rights statement and submits for validation. **COMMONS_ACCEPTED:** admissible for the declared training or evaluation purpose, never used automatically. Revocation is a new policy event that prevents subsequent authorized uses; already-deployed weights may require retraining or risk mitigation — the protocol does not pretend to instant unlearning.

6. L3 — Distribution and Service Network

Retrieval is multi-source. A client resolves providers for a root CID via a Kademlia DHT [7] (XOR metric, k = 20 buckets), opens WebRTC data channels, and requests disjoint chunk ranges from each provider in parallel (Figure 3). Since every chunk is self-verifying (I2), the swarm needs no reputation to be safe — reputation only optimizes peer selection. Store-and-forward mailboxes cover offline recipients: messages are encrypted to the recipient′s public key, so relays hold only ciphertext with a TTL. Nodes are economically incentivized without ever receiving decryption keys; CDN gateways, IPFS/Filecoin integrations and conventional object stores are interchangeable service options behind the same interface.



Chunks are fetched in parallel from interchangeable, untrusted providers and verified against their CID on arrival.

Figure 3: Swarm retrieval with per-chunk verification.

The service network produces three distinct proofs of useful service, each with its own evidence and each insufficient alone. **Storage:** probabilistic Proof-of-Storage challenges — a random chunk index and salt, answered with SHA-256(salt ‖ chunk) and verifiable against the DAG — time-bounded, independently sampled and combined with customer-signed demand so that replicated fake demand and correlated failures are detected. **Retrieval:** an authorized request identifier, object/version commitment, transfer receipt, byte range and deliv-

ery SLA with anti-self-dealing checks; user download history is never published. **Compute:** an approved job, measured resource usage, output artifact, correctness checks and, where available, attestation — GPU utilization alone is not evidence of useful work. The initial replication target is three *independently administered* failure domains: a provider that writes one object into three virtual containers on one machine has delivered one durability domain, not three. A cache-cleanup automation garbage-collects unreferenced chunks, and a chunk-healer re-replicates under-replicated objects when providers churn below the replication factor; Table 4 lists the implementation defaults.

Table 4: Data- and network-layer parameters (implementation defaults).

Parameter	Value	Rationale
Chunking	Gear CDC, avg 256 KiB	shift-resistant dedup
Chunk bounds	64 KiB – 1 MiB	bounded proof paths
Cipher	AES-256-GCM, 96-bit nonce	AEAD, hardware-accelerated
Hash / CID	SHA-256	WebCrypto native
Key hierarchy	device root → wrapped per-object DEKs	no escrow, no wallet coupling
DHT	Kademlia, k = 20	O(log N) provider lookup
Replication factor	3 independent domains	availability vs. cost
Transport	WebRTC + TURN fallback	NAT traversal in browsers

7. L4 — Authorized Confidential Compute

This layer resolves the encrypted-data-to-AI gap. Storage never decrypts autonomously. A policy engine evaluates an object-specific RightsGrant against a declared purpose, model or workload, jurisdiction, expiry and runtime security level; only then may a short-lived key unwrap or a local computation proceed. The owner′s decision and the runtime′s technical identity are recorded independently. Three execution paths offer different guarantees (Table 5); the reference path is *self-hosted processing with explicit key release*, and confidential-GPU execution is a stronger optional profile when hardware and remote-attestation checks support it. Fully homomorphic encryption, secure multiparty computation and zero-knowledge ML remain research directions, not plug-in substitutes for large-model training.

Table 5: Execution paths and their guarantees.

Execution path	How plaintext becomes usable	Control / non-guarantee
Local RAG / adaptation	Owner device decrypts and performs retrieval, inference or fine-tuning locally	No remote plaintext; device security remains critical
Federated contribution	Owner trains on device; sends policy-governed update through secure aggregation [9]	Update leakage and poisoning still require defenses
Self-hosted secured job	Owner authorizes DEK unwrap to a dedicated, measured process; plaintext in job memory only	Operator and hardware TCB must be disclosed and audited
External API inference	User accepts provider terms; GhostDrive logs call, latency and result	No training receipt possible for unseen model internals

The minimal *key-release transaction* is: a training or RAG run requests {job_id, object_version_ids, signed_rights_grants, purpose, model_id, training_code_hash, expiry, policy_epoch, compute_measurement}; the policy enforcement point verifies every attribute, creates a one-use authorization, unwraps only the requisite DEKs to the allowed process and records {decision_id, authorized_object_set_root, key_release_receipt_hash, run_id}; the job signs a manifest and emits an output artifact. Keys expire, are never passed to storage peers and never become retrievable by a DAO wallet. Before launch each deployment discloses the operator entity, cloud or on-premises owner, jurisdiction, model identifiers, sub-processors, audit rights and recovery arrangements; an ordinary GPU server or encrypted object store is never described as a trusted execution environment.

8. L5 — Living Knowledge and AI Intelligence

Information is not one immutable blob. Some records are enduring source documents; others are mutable observations, reversible interpretations, temporary instructions or claims that later evidence contradicts. L5 transforms discrete private or licensed inputs into time-aware, safely reusable intelligence without rewriting history, using the W3C PROV-O vocabulary of entities, activities, agents and derivations [13]. Table 6 defines the five knowledge classes it distinguishes.

Table 6: Five knowledge classes and their update policies.

Class	Meaning	Update policy
STATIC_SOURCE	Immutable document, source asset, signed event	Never mutate; append annotations or new edition
TEMPORAL_FACT	Claim valid for an interval (an office holder, a price)	Retain observed_at, valid_from/to, supersession, source
PERSONAL_MEMORY	Private preference or user experience	Owner-only edits, purpose constraints, deletion requests
DERIVED_KNOWLEDGE	Summary, embedding, inference, agent synthesis	Store lineage, method, model version, confidence
CONTESTED_CLAIM	Conflicting or disputed facts	Retain alternatives, uncertainty, evidence, adjudication

Every dynamic claim distinguishes *valid time* (when it was allegedly true), *observation time* (when evidence was captured) and *ingestion time* (when the system accepted it). A superseded claim remains historically queryable subject to rights and retention; agents expose source, freshness, uncertainty and disputes in their answers — stronger than silently changing a vector embedding when a website updates. Each version carries parent links, rights-policy id, contributor pseudonym, provenance references, conflict markers, content commitment and the outcome of any safety review; a derivative must cite the exact parent snapshot and transformation. Cryptographic provenance shows who asserted and transformed a record; it does not establish factual correctness.

How information communicates with information. An append-only event bus publishes permission-filtered version events, never unredacted global memories. The update protocol is **discover** (subscriptions surface a permitted claim relevant to a snapshot or index), **compare** (semantic overlap and provenance against a rights-cleared subset, with no cross-user leakage of private material), **classify** (unchanged, correction, contradiction, expiration, extension or unsafe), **approve** (owner approval for private memory; governed reviewer approval for commons training changes), **materialize** (publish the next signed snapshot; refresh RAG indexes or schedule an auditable model job; roll back a failed candidate) and **reconcile** (record which versions were used and eval-

uated and whether reward maturity changed). Updating retrieval indexes and updating model parameters are separate operations: a validated KnowledgeSnapshot may refresh RAG immediately, whereas model parameters change only through a controlled training job, evaluation and promotion gate. This prevents retrieval-time memory improvement from being confused with permanent model learning.

8.1 LEAN admission: acceptance precedes compensation

Every candidate for COMMONS_ACCEPTED passes a formal qualification whose four components give it its name — **Lineage**, **Evidence**, **Ablation**, **Novelty** (Table 7) — and whose two outputs are kept distinct: a *scientific-merit* score SM (is the contribution true, reproducible and new?) and a *computational-utility* score CU (does it measurably improve the pinned model on the deficiency it targets?). They are related and not interchangeable: a novel result may not help a language model; a mundane, verified dataset may fix a major weakness. The LEAN score is min(SM, CU) after critical exclusions, and it is this score — not a declared quality field — that the succinct chain′s admission predicate commits to (§9.2, score ≥ 85).

Table 7: LEAN qualification components.

Component	What is checked	Critical exclusion (overrides any score)
Lineage	Original source material; contributor′s rights; provenance class — HUMAN_PRIMARY, VERIFIED_EXPERIMENTAL or DERIVATIVE_SYNTHETIC (produced by another model); recursion depth of synthetic material	Rights not demonstrable; synthetic material without disclosed generator and source; summary of public material
Evidence	Factual accuracy against independent sources; methodological reproducibility; corroboration; conflict with existing accepted claims	Reproduction attempt fails; contradicted by verified experimental record
Ablation	Held-out delta on the targeted deficiency with and without the contribution, against a baseline that received the same compute; adverse regressions; overhead	No measurable delta, or regression beyond the declared ε on any protected evaluation
Novelty	Semantic duplication against the rights-cleared corpus and the public distribution; whether the contribution introduces a new relation between concepts or corrects a documented failure	Linguistic variation of existing material; already in the public distribution

Provisional thresholds, subject to calibration before any public claim: below 75, rejected; 75–84, a *research* class retained for examination with no training privilege and no compensation; 85–89, admitted after independently supervised validation; 90 and above, admitted after multidisciplinary review. No score overrides a critical exclusion. The thresholds are validated by gate G5 against acceptance error, inter-reviewer agreement and measured model improvement, and are revised by governance only with that evidence attached. DERIVATIVE_SYNTHETIC material is capped at a declared share of any training batch and is admitted only with ablation evidence, because recursively recycled model output amplifies errors and erodes rare observations — the model-collapse risk is treated as a provenance constraint, not a quality score.

Model improvement is measurable, not assumed. Reference workloads are privacy-preserving RAG against a signed snapshot, approved fine-tuning or adapters of the pinned model, and controlled evaluations. A candidate model version is promoted only after declared quality, privacy, safety and regression tests; the upstream model family, tokenizer, weights hash, training code, evaluation dataset and operator are pinned in the record. A receipt (§9) proves that admitted material entered a batch; the ablation proves that it helped. Neither substitutes for the other.

Attribution is optional; settlement is private. Contributors who want public recognition — researchers, institutions, licensors — attach a public attribution record (name, DOI, affiliation) to the KnowledgeVersion; those who need confidentiality use a pseudonym. In both cases the payout route is unlinkable from the public ledger (§9.1). Anonymity is a capability for those who need it, not an assumed preference of the contributors whose work the system most needs.

9. L6 — Evidence, Verification and Safety

9.1 The evidence ladder

L6 binds claims about data and AI to actual evidence while protecting sensitive content. The evidence ledger records what happened, who was authorized to do it and what an independent reviewer can check; it does not claim omniscience about model causality or true meaning. Every claim carries a level (Table 8), and acceptable language is bounded by that level: when a trainer is not instrumented to support E4, only E3 may be claimed.

Table 8: Evidence ladder — what each level does and does not establish.

Level	What may be established	What is NOT established
E0 submitted	Agent submitted a signed commitment	Content was useful or lawful
E1 valid	Hash, rights and consent gates passed	Any AI read it
E2 admitted	Version entered an authorized snapshot	A training run consumed it
E3 fetched	A controlled job logged retrieval or batch access	A gradient included its content
E4 consumed	Training-loop input capture linked exact examples to steps	Causal improvement or retention
E5 evaluated	Declared model comparison and outcome tests recorded	Single-example causal attribution
E6 settled	Dispute window closed; capped entitlement committed	Anonymity on the public ledger

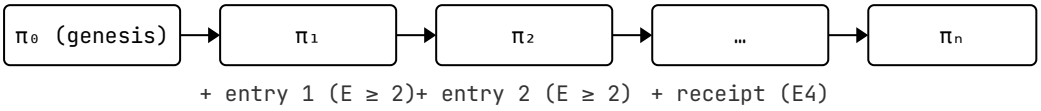
A model-use receipt links consent_id, object_version_commitment, dataset_snapshot_root, authorized_job_id, measured executable or attestation, model_input_hash, model_version before and after, evaluation reference and signer identities: Receipt = $\text{Sign}_{\text{verifier}}$(H(schema ‖ level ‖ job_id ‖ snapshot_root ‖ model_hash ‖ measurement ‖ policy_hash ‖ timestamp ‖ nonce)). Only aggregate roots and selectively disclosable receipts are published; the public claim uses an opaque epoch-specific nullifier rather than a stable contributor-to-content mapping. Safety is a separate gate: prompt-injection resistance, output moderation, sensitive-data-leakage evaluation, adversarial testing and incident response remain indispensable [14, 15] and all text retrieved from stored knowledge is treated as untrusted input.

9.2 The succinct chain

The registry of accepted knowledge versions, their rights scope and the receipts that consumed them is audited not by a blockchain that every verifier replays, but by a recursive zk-SNARK (o1js / Kimchi over the Pasta curves [4, 5]) that folds the registry one entry at a time:

$$\texttt{head}_i = \texttt{Poseidon}(\ \texttt{head}_{i-1},\ \texttt{H(entry}_i\texttt{)},\ \texttt{level}_i\ ) \tag{2}$$

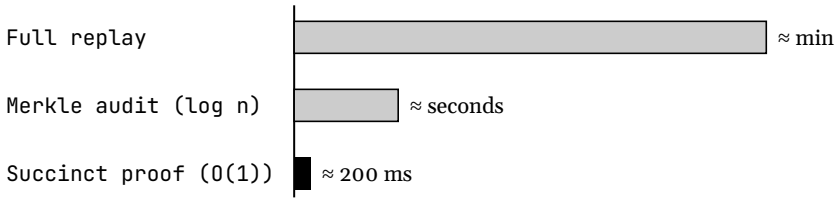
The circuit enforces that the fold is computed correctly and that every folded entry satisfies its admission predicate — rights gates passed, LEAN score ≥ 85 (§8.1) and evidence level ≥ the declared threshold — so a checkpoint is simultaneously an integrity and an admission attestation. Each step verifies the previous proof recursively (Figure 4), so the final proof π_n covers the entire history yet remains constant in size regardless of n.



Each proof π_i verifies π_{i-1} inside the circuit; π_n attests the full history at constant size.

Figure 4: Recursive folding. The prover pays O(n); every verifier pays O(1).

Checkpoints — (head, count, π_n, verification key) — are published through a session-verified backend by prover nodes. A light client downloads one checkpoint and runs verify(π, vk) in the browser: no circuit compilation, no data download, ≈200 ms in the v1.0 prototype (Figure 5). A trust-on-first-use check on the verification-key hash flags circuit substitution: a changed vk means a different program produced the proof.



Relative client-side verification cost for a 10^6-entry registry (log scale, normalized).

Figure 5: Client-side cost of auditing a large registry, by verification strategy.

What zero knowledge can and cannot prove. Correctly implemented, a circuit can prove that an item commitment belonged to a published registry root, that a published threshold predicate was satisfied, or that a claim nullifier has not previously been spent. It cannot prove by default that a content claim is true, that a rights holder was lawful, that training improved a model, that an external provider ran a secret training step, or that a public wallet is anonymous. Proof size and verification cost depend on circuit, recursion structure and implementation; the ≈22 KB and ≈200 ms figures of §12 are measurements of the v1.0 circuit and must be reproduced for the extended admission predicate.

10. L7 — DRIVE Economy and Governance

Availability and learning are purchased, not assumed. L7 aligns scarce subsidy releases and customer-funded service payments to verified activity; it consumes only L6-adjudicated evidence. The token is a settlement, service-payment and governance instrument — not the storage encryption primitive, not the AI model, and not authority to modify private memories. DRIVE is an ERC-20-compatible token [16] with a fixed genesis supply of 369,000,000 units allocated into enforceably segregated vaults (Table 9). A payout is a transfer from a previously funded reserve, never a new mint; any chain choice, bridge or migration must reconcile exactly to the original supply.

Table 9: Fixed genesis allocation (GhostDrive 369 token proposal v0.2; design proposal, not audited balances).

Funding vault	Percent	DRIVE tokens	Purpose

Funding vault	Percent	DRIVE tokens	Purpose
Proof-of-Utility Mining	69.30%	255,717,000	Storage 63% · compute 27% · retrieval 10% (budget split)
AI Knowledge Mining	3.69%	13,616,100	Rights-cleared, verified-use knowledge contributions
Development + Infrastructure	10.00%	36,900,000	Milestone-based R&D, timelocked
Community Activity Distribution	9.63%	35,534,700	90-day waves, ≤ 3.69% of the original reserve each
DAO Treasury	3.69%	13,616,100	Bounded governance spending
Early Contributors	3.69%	13,616,100	Cliff and linear vesting
Total fixed supply	**100.00%**	**369,000,000**	No mint authority beyond one-time genesis

Each mining vault with allocation M releases at most $R_y(M)$ in protocol year $y \in 1..369$, with $q = 2^{-1/36.9}$:

```
R_y(M) = M · (1 - q) · q^(y-1) / (1 - q^369)
```
(3)

The releases sum to M over the schedule; deployed contracts use fixed-point integers, precomputed cumulative ceilings and explicitly reconciled rounding dust. The schedule is a *ceiling on eligible claims*, not a promised market return: unused capacity stays in its vault.

Two reward engines remain separate. **Proof-of-Utility** pays for billable storage, useful compute and legitimate retrieval against an L3 service proof plus L6 independent verification; the service score for storage is $\text{StorageScore}_i = \text{verified_GiB_hours}_i \times \text{availability}_i^2 \times \text{integrity_gate}_i \times \text{bounded_demand}_i$, and budget within a category is $\text{Reward}_i = B_{\text{epoch,category}} \times \text{Score}_i / \Sigma(\text{eligible scores})$; if total eligible score is zero, the budget is unspent. **Knowledge Mining** pays for rights-cleared new corrections, insights or examples *actually used*, against an L4 authorization, L5 snapshot and L6 E4/E5 evidence, with the indicative quality-gated score

```
K_i = Consent_i · Rights_i · Quality_i² · Novelty_i · Utility_i · √(1 + Corroboration_i)
```
(4)

where Quality_i is the LEAN score (§8.1) and maturity levels apply: an accepted contribution (E2) is not a mining entitlement; verified consumption (E4) may create a conditional claim; a measured outcome (E5) may unlock the declared bonus. For a revision sequence v1 → v2 → v3 a revision is eligible only for its validated *marginal* contribution; automatic perpetual royalties are not implied by a content address, and the same event cannot be claimed twice across pools. Uploading private memories is never an eligibility condition. For a collusive attack to be unprofitable, the maximum subsidy captured by fabricated traffic must be below the adversary′s expected cost after service charges, challenge bonds, detection risk and penalties; measuring *useful work* is an oracle problem and is treated as one.

Customer service fees are the primary revenue: storage is offered at a target list price of $3.69 per TB-month with three replicas (§14.1), and compute and retrieval are priced per metered work unit. Vault releases supplement — never replace — paid demand.

10.1 Contributor acquisition

The pipeline does not depend on a population of rare multidisciplinary researchers choosing token rewards over publication. Its three contributor classes have reasons to participate that publication does not satisfy (Table 10), and the fourth class — individuals submitting AI-assisted syntheses of public material — is excluded at the lineage stage by design.

Table 10: Contributor classes and their reason to participate.

Class	What they hold	Why the pipeline rather than publication or a vendor contract
Institutional licensors	Withheld or boundary-locked corpora	Purpose-bound, receipted licensing with revocation; revenue for data they cannot publish; audit trail for their own compliance
Domain practitioners	Private operational data, documented model failures in their work	Payment plus optional attribution for corrections they would never write up as a paper; the model they fix is the one they use
Researchers	Novel, reproducible results	Attribution preserved (DOI linked); payment is additive to publication, not a substitute; priority timestamp from the registry

10.2 Cost per accepted improvement and the break-even condition

The economically relevant quantity is not DRIVE paid per contribution but the total cost of obtaining one independently demonstrated model improvement, including every rejected submission:

```
C_imp = ( C_screen / p_acc + C_validate + C_train+eval ) / p_imp
```
(5)

where C_{screen} is the automated lineage, evidence and novelty cost per submission, p_{acc} the share that reaches LEAN ≥ 85, $C_{validate}$ the supervised or multidisciplinary review cost per admitted item, $C_{train+eval}$ the ablation and promotion cost, and p_{imp} the share of admitted items that produce a measured improvement. Knowledge Mining opens only if C_{imp} is at or below the cost of the best conventional alternative for the same deficiency (C_{alt}: contracted data, in-house SFT examples, or distillation where permitted), or if a compensating advantage is documented — rights that cannot be obtained any other way, data that cannot leave a boundary, or a privacy requirement the alternative cannot meet. The reward per admitted item is bounded above by $C_{alt} - (C_{screen}/p_{acc} + C_{validate} + C_{train+eval})$; if that bound is zero or negative, the category pays nothing that epoch and the vault retains its budget. Every term in Eq. 5 is measured in the comparative experiment of §12 and published with the first Knowledge Mining epoch.

Governance is bounded. Token holders may propose evaluator rules, model-registry policies, network budgets and agent permissions through a 3-day discussion → 6-day technical review → 9-day vote → 3-day timelock process (9-day timelock for high-risk changes), implemented with standard timelocked-governor primitives [17]. No majority may grant decryption rights, override a rights holder′s refusal, disable minimum safety boundaries or expand fixed supply.

11. The Knowledge-Use Transaction

The layers define ownership; the following transaction defines how the machine actually works. Each step returns a signed, falsifiable artifact before the next step can progress (Figure 6, Table 11); a rejection or expiry returns to the earlier appropriate state rather than bypassing a gate. A reward is not an automatic property of data — it is a disputed, capped entitlement after verified use.

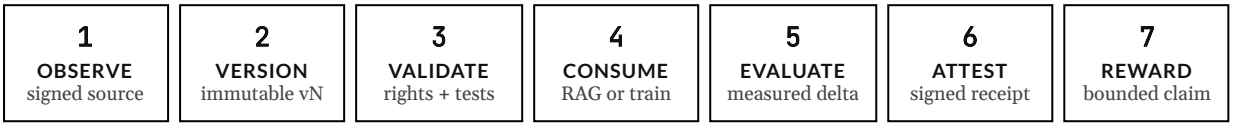


Figure 6: Seven executable transaction stages and the explicit knowledge-update feedback loop.

Table 11: Operations, trust transitions and abort conditions.

Step	Operation and trust transition	Required artifact / abort condition
1 Observe	Owner submits a new personal source or selects an opt-in change under L1	Signed owner intent; abort if authority invalid
2 Version	L2 creates encrypted immutable version and guarded metadata	Object/version commitment + rights scope; abort on integrity failure
3 Validate	L3 retrieves; L4 grants purpose; L5 checks rights, quality, novelty and risk	Rights-admission receipt; reject duplicates or unlawful data
4 Consume	L4 runs allowed local, federated or hosted inference/training; L5 pins snapshot	Run manifest with operator, model, objects; abort if grant revoked
5 Evaluate	L5 compares declared utility, privacy, safety and regression metrics	Evaluation report; do not infer causality from inclusion
6 Attest	L6 gathers independent evidence, folds receipt into the succinct chain, records challenges	Signed evidence bundle + dispute result; suspend if contested
7 Reward	L7 checks vault ceiling, epoch root, nullifier and recipient authorization	Single-use claim; leave budget unspent if evidence insufficient

Worked example. A scientist creates private note v1. She corrects a public-domain equation and opts in only the correction v2, not her notebook. L2 encrypts both and commits v2 as a separately licensed derivative; L4 authorizes a named research model or evaluation run; L5 indexes v2 into a time-stamped commons snapshot. A controlled job logs that v2 was retrieved (E3) and incorporated into a declared training batch (E4); a held-out test evaluates the change (E5). Independent reviewers inspect the artifacts for poisoning and rights. L7 recognizes at most one claim for the approved update in the reward epoch; a later v3 correction is scored only for new value. The original notebook remains private. *The opposite example:* an attacker submits 10,000 variations of synthetic text and pays a cooperating node to "retrieve" them. Hashes and signatures exist, but rights, novelty, real customer demand and independently measured utility fail; L6 rejects the evidence, L7 creates no claim and the budget stays in its vault. A zero-knowledge proof of hash inclusion never overrides that verdict.

12. Evaluation

We measured the v1.0 browser implementation of L1–L3 and the succinct chain (Chrome 126, M2 laptop, 100 Mbit/s) on mixed corpora: documents, images, model checkpoints and chat exports. Numbers are medians over 20 runs (Table 12). The asymmetry is deliberate: proving is expensive and performed by few incentivized nodes; verification is cheap and performed by everyone. Verification cost was flat from $n = 4$ to $n = 10^3$ folded entries, confirming O(1) behavior. These figures were obtained without a bundled reproduction package and are reported as background for the extended circuit, not as v2.0 guarantees.

Table 12: Measured end-to-end performance of the data and proof layers (v1.0 prototype, July 2026).

Operation	Input	Result
Chunk + hash + encrypt	100 MB file	2.9 s (34 MB/s)
Upload overhead vs. plaintext	mixed corpus	+8–14%
Deduplication ratio (v1 CID scheme)†	versioned documents	1.31–1.62×
Swarm retrieval (3 peers)	100 MB file	1.7× single-peer throughput
Circuit compile (prover, once)	registry circuit	71–180 s
Recursive fold per entry	1 entry	9–14 s
Proof size	any n	≈ 22 KB (constant)
Light-client verify	any n	190–240 ms
PoS challenge round-trip	1 chunk	< 350 ms

† Measured with the v1.0 convergent-CID scheme. Under the v2 random-nonce design, deduplication moves to the owner's keyed private index (§5.2) and must be re-measured.

The controlled comparative experiment. Before Knowledge Mining opens, the operator runs one experiment with published artifacts: the pinned baseline model; its measured deficiencies on a held-out suite; four arms that receive equal compute — (a) continued pretraining on conventionally acquired in-domain data, (b) supervised fine-tuning on in-house examples, (c) distillation from a permitted teacher, (d) adapters or RAG on LEAN-admitted contributions; and blinded evaluation of each arm's delta and regressions. The output is the delta per arm, every term of Eq. 5 for arm (d), and C_{alt} for the best of (a)–(c). If arm (d) does not match the best alternative on cost per improvement or document a compensating advantage, the Knowledge Mining vault stays closed and the pipeline is retained only as a rights and provenance layer for arms (a)–(c).

The AI layers are evaluated by hypothesis, not by assertion. Table 13 defines the measurable claim each layer must pass, with baseline comparisons and reproducible artifacts, before it is described publicly as a capability. A review release includes source and pinned dependencies for test components, a safe sample corpus with rights statements, schema definitions, model and tokenizer hashes, training configuration, evaluation code, synthetic workloads, verification keys and raw measurement logs; private contributor records are never bundled, and external evaluations are structured to avoid leakage into the hold-out set.

Table 13: Evaluation design for the proposed layers.

Hypothesis	Metric	Minimum experimental approach
Private input stays scoped	Unauthorized plaintext access attempts; key-release audit	Adversarial grants, isolated nodes, device-compromise drills
Knowledge remains fresh	Update-to-RAG-visible delay; stale-answer frequency	Time-stamped known-answer changes with rollback
Model inclusion is evidenced	Recall/precision of receipt-to-batch matching	Instrumented training batches and independent sampling
Updates help, not harm	Held-out task delta and adverse regressions	Baseline vs. adapter/RAG with blinded evaluation
Rewards resist farming	Adversarial profit/cost under synthetic demand	Sybil/collusion simulation and verifier stress
Privacy withstands correlation	Linkage and inference attack success rates	Wallet-graph, timing and metadata attack benchmarks
Token ceilings conserve supply	Vault invariant failures in fuzz tests	Property-based and adversarial contract testing
LEAN thresholds are calibrated	Acceptance error, inter-reviewer agreement, score-to-improvement correlation	Blinded re-review of admitted and rejected items; threshold sweep
Pipeline beats conventional acquisition	C_{imp} vs C_{alt}; delta per arm	Four-arm controlled experiment with equal compute, blinded evaluation

13. Threat Model

Privacy guarantees are expressed against a specific adversary and endpoint. Encryption protects data on untrusted storage nodes, but not necessarily against a compromised owner device, intentionally authorized compute, malicious training code or token-analysis correlation. Zero-trust principles [18] require per-request authorization and independent enforcement rather than network-location trust. Raw document content, low-entropy plaintext hashes, user prompts, private dataset labels, sensitive categories, global identity mappings and reward-related behavioral telemetry are never placed on a public ledger; the public commitment proves narrowly declared facts. GDPR-class erasure is supported by ciphertext deletion plus key destruction — irreversible, which permanent-storage chains cannot offer — but compliance is never inferred solely from key destruction: weights and backups may retain influence, and the protocol blocks new use and evaluates retraining or unlearning feasibility rather than claiming instant forgetting. Table 14 summarizes mechanisms and residual risk.

Table 14: Security properties, mechanisms and residual risk.

Threat	Mechanism	Residual risk
Server reads data	Client-side AES-GCM, device key root, no escrow	Client compromise
Peer injects bad chunk	CID re-hash on receipt (I2)	Bandwidth waste only
Key release to bad worker	Attestation / code measurement; policy-bound one-use unwrap; zero standing privilege	TEE, driver, firmware bugs; operator abuse
Rogue or compromised owner agent	Capability minimization, user confirmation, sandbox and expiry	Device or key compromise in scope

Threat	Mechanism	Residual risk
Data poisoning / malicious revisions	Provenance, source adjudication, held-out evaluation, quarantine	Coordinated plausible falsehoods
Prompt injection via stored knowledge	All retrieved text untrusted; separated tools, instructions and permissions	Complex agent toolchains remain vulnerable
Forged registry history	Recursive SNARK + vk TOFU	Circuit bug
Receipt forging / model provenance	Signed manifest, independent samples, code and batch commitments	Dishonest operator collusion
Sybil reward farming	Cost-of-attack evaluation, verifier diversity, claim nullifiers, appeals	Uniqueness not inferable from wallets
Wallet ↔ contribution linkage	Ephemeral contribution ids; blind/nullifier claims; separate payout route	Timing and on-chain analysis
Storage node drops data	PoS challenges + re-replication across independent domains	Correlated churn
DAO or vault capture	Timelocks, role separation, immutable ceilings	Upgrade-key abuse

14. Comparative Analysis

Table 15 positions GhostDrive against representative systems. The distinguishing combination is default confidentiality *and* constant-time public auditability *and* policy-gated AI computation with training receipts — each exists somewhere in isolation, none together. Relative to Mina [4], GhostDrive borrows recursion but applies it to a knowledge registry with in-circuit admission predicates rather than a payment ledger. IPFS is treated together with Filecoin, whose content addressing it provides [2, 3]. Relative to Filecoin [3, 19], Proof-of-Storage here proves possession of *specific ciphertext chunks of a named DAG*, settlement is per-contract rather than per-sector, and storage is only one of three rewarded services. Relative to content-credential schemes such as C2PA [20], provenance here binds not only the asset but the model job that consumed it.

Table 15: Feature comparison with deployed storage systems.

Property	GhostDrive	Filecoin (IPFS)	Arweave	Storj	Cloud (S3/GDrive)
E2E encryption by default	●	○	○	●	○
Content addressing	●	●	●	◐	○
Storage proofs	●	●	◐	●	○
O(1) history verification	●	○	○	○	○
Policy-gated AI compute	●	○	○	○	◐
Training / retrieval receipts	●	○	○	○	○
Temporal, versioned knowledge	●	◐	◐	○	◐
Right to erasure	●	◐	○	●	●
Runs fully in browser	●	◐	○	○	●
Provider can read data	no	yes	yes	no	yes

● full support, ◐ partial / opt-in, ○ absent. Filecoin "provider can read data" assumes no client-side encryption; Cloud "policy-gated compute" denotes operator-side IAM, where the operator itself is in the trust boundary.

14.1 Price

Table 16 compares list prices for one terabyte stored for one month. GhostDrive's list price is set at **$3.69 per TB-month including three replicas in independently administered failure domains** — about $1.23 per replica, against roughly $2.27 per copy on Filecoin warm storage and $23 on S3 Standard. The price is a target denominated in USD and settled in DRIVE at the epoch reference rate; it is a design proposal, not a measured market price.

Table 16: Indicative list price per 1 TB stored for one month (public price pages, accessed November 2026).

System	Price / TB-month	Copies included	Per copy	Egress	Confidential	Verifiable
GhostDrive pinning (target)	**$3.69**	**3 independent**	**≈ $1.23**	**P2P swarm**	**yes**	**yes**
Filecoin warm storage[a]	≈ $4.55	2	≈ $2.27	IPFS gateway / CDN	no	yes
Storj Standard[b]	$7.00	erasure-coded	—	$7 / TB	yes	partially
Amazon S3 Standard[c]	$23.00	multi-AZ	—	$90 / TB	no	no
Arweave[d]	one-time ≈ $19,600	permanent	—	gateway	no	partially

[a] Filecoin Onchain Cloud: $2.50 per TiB per month per copy, two copies by default (≈ $4.55 per decimal TB), plus a small per-data-set fee and FIL gas [22]. [b] Storj simplified pricing effective July 2026 [23]. [c] US East, first 50 TB, $0.023 per GB-month; egress $0.09 per GB for the first 10 TB [24]. [d] Pay once for permanent public storage, at protocol rates with AR ≈ $1.91 (July 2026) [25]; not directly comparable with monthly rent. Prices of token-settled networks move with token and network conditions.

Whether $1.23 per replica is sustainable depends on provider cost, not on the token. During bootstrap, providers earn the customer fee plus a Proof-of-Utility storage subsidy (63% of the utility vault's release, §10); because the storage score multiplies by *bounded_demand*, the subsidy grows only with paid demand, so idle or self-funded capacity cannot capture it. The list price is reviewed by governance against measured provider cost and can change only within the fixed vault ceilings; the agent-based economic simulation of gate G7 must show that honest providers break even at the target price before it is published as a commercial offer.

15. Use Cases: Personal to Enterprise

Table 17: Deployment profiles across user classes.

Tier	Primary asset	GhostDrive mechanism	Outcome
Personal	notes, media, chat memory	local-first vault + local RAG under owner keys	sovereign AI memory that no operator reads
Creator	designs, music, manuscripts	CID timestamp + checkpoint anchor + opt-in licensing	provable authorship; paid, receipted AI use
Team	shared datasets, docs	CRDT spaces + per-object DEKs + purpose grants	E2E collaboration without collective-training consent
AI lab	rights-cleared training sets	admission-gated registry + key-release jobs + E4/E5 receipts	auditable model lineage per training batch
Enterprise	regulated records, IP	self-hosted compute profile + pinning SLAs + succinct audits	AI on regulated data inside the compliance boundary

15.1 Intellectual property

Registering a work is publishing CID(E_k(work)) into a checkpointed registry: the creator proves possession at time t without revealing content, and can later selectively disclose by releasing the key to an arbiter. Because every model-use receipt cites the dataset snapshot root it trained on, a rights holder can verify whether a given model′s evidenced lineage includes — or, to the level instrumented, excludes — their material, and license future use through a RightsGrant whose purpose, processor and expiry are enforced at key release rather than by terms of service.

15.2 Enterprise compliance

An auditor receives a constant-size proof and verifies in milliseconds that the data estate is intact, admission-gated and unchanged since the last audit, and that every AI job that touched it holds a grant and a receipt — no access to the data itself is required or possible. Reference deployment mode A is a self-hosted, open-weight model under the designated GhostDrive operator entity (§2.1) that controls GPU scheduling, model registry, training-code release and security logs without owning upstream model weights; modes B (federated), C (confidential compute) and D (external API) are disclosed with correspondingly weaker or stronger evidence.

16. Implementation Gates and Open Questions

Implementation follows the seven-layer logic: each gate produces a runnable artifact and an independently checkable test, and the system does not publicly claim the next capability before evidence of the preceding one exists (Table 18). The first integrated pilot is a small, rights-cleared educational corpus and one self-hosted open-weight model on a named controlled GPU service, with three participant kinds — private-only owners, licensed contributors and independent reviewers — running the full sequence from private source to settled test-token claim, then repeated with revoked permissions, poisoned updates, duplicate submissions and a simulated dishonest operator.

Table 18: Seven implementation gates.

Gate	Shippable milestone	Exit test / evidence
G1 Rights	Owner app, separate identities, scoped delegated-agent credentials	Revoked/expired grants blocked under automated attack tests
G2 Vault	AEAD file versions, private metadata, key recovery, content commitments	Version lineage and encrypted-storage invariant verified
G3 Network	Encrypted storage/retrieval, independently tested nodes, SLA monitoring	Measurable durability, anti-duplicate usage, outage drills
G4 Compute	Pinned self-hosted model, scoped key release, run manifests	Test object reaches runtime only under valid grant
G5 Living AI	Temporal knowledge graph, versioned RAG, LEAN admission pipeline, gated adapter pipeline	Correct updates, rollback, freshness and regression tests; four-arm comparative experiment published with Eq. 5 terms
G6 Evidence	Instrumented E0–E5 receipt pipeline, adversarial testing, multi-verifier challenge	Reproducible E4 batch inclusion; contested-score review
G7 DRIVE	Vault and DAO simulations, immutable caps, claim proofs, privacy assessment	Fuzzed conservation, challenge windows, external audits

Technical papers become stronger by stating what remains unsettled. Open before any public claim: reconciliation of the six-vault proposal with any existing DRIVE contract state, beneficiaries and vesting; the canonical settlement chain and bridge rules; the published operator legal entity, processors and consent review; actual confidential-hardware SKUs with measured workloads and penetration tests; calibration of the LEAN thresholds and of the SM/CU separation against real acceptance errors; the empirical qualification rate and C_{imp} from the comparative experiment; a practical strong-anonymity protocol for rewards, since payout graphs and login metadata can re-identify users; retention boundaries and disclosed limits of model unlearning; and audited migration constraints for the 369-year schedule, since future chains, keys and institutions can fail.

17. Conclusion

We have presented a storage and learning architecture in which privacy is the default, verification is constant-time, computation on private data is authorized per request, knowledge is time-aware and correctable, and AI models carry cryptographic receipts for the data that made them. The strongest form of GhostDrive is not ″an encrypted AI that learns from everything″; it is a rights-aware, time-aware, verifiable knowledge machine that answers seven operational questions — who authorizes, what is stored, how it travels, where plaintext is processed, how intelligence changes, what can be proved, and who is legitimately paid. The prover/verifier asymmetry of recursive SNARKs, applied to an encrypted, admission-gated registry, yields a system auditable by anyone and readable by no one but its owners; the evidence ladder and capped proof-of-utility economy ensure that knowledge gains value through verified use, never merely because it moves. Its strength depends on implementation, independent assurance, economic adversarial testing and the deliberate separation of cryptographic proof from semantic truth — and, for the knowledge economy specifically, on a comparative experiment this paper defines but has not yet run. Future work includes hiding evidence levels and scores under zero-knowledge range predicates, BLS-aggregated storage proofs, multi-prover checkpoint consensus, and the integrated pilot of §16.

18. References

1. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.
2. J. Benet, "IPFS — Content Addressed, Versioned, P2P File System," 2014.
3. Protocol Labs, "Filecoin: A Decentralized Storage Network," 2017.
4. I. Meckler, E. Shapiro, "Coda/Mina: Decentralized Cryptocurrency at Scale," 2018.
5. o1Labs, "Kimchi: A Plonk-based proof system over the Pasta curves," 2022.
6. W. Xia et al., "FastCDC: A Fast and Efficient Content-Defined Chunking Approach," USENIX ATC, 2016.
7. P. Maymounkov, D. Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," 2002.
8. M. Shapiro et al., "Conflict-free Replicated Data Types," SSS, 2011.
9. K. Bonawitz et al., "Practical Secure Aggregation for Privacy-Preserving Machine Learning," CCS, 2017.
10. R. Nebo, "GhostDrive: A Privacy-First Chain for AI Data and Intellectual Property," v1.0, July 2026.
11. R. Nebo, "GhostDrive: Sovereign Knowledge & Verifiable AI," v2.0 research draft, November 2026.
12. R. Nebo, "GhostDrive 369: Seven-Layer Living Knowledge & Proof-of-Utility," research white paper v3.0, November 2026.
13. W3C, "PROV-O: The PROV Ontology," W3C Recommendation, 2013.
14. NIST, "AI 600-1: Artificial Intelligence Risk Management Framework — Generative AI Profile," 2024.
15. OWASP, "Top 10 for Large Language Model Applications," 2025.
16. Ethereum, "EIP-20: ERC-20 Token Standard"; "EIP-712: Typed Structured Data Hashing and Signing."
17. OpenZeppelin, "Governance and TimelockController," Contracts 5.x.
18. NIST, "SP 800-207: Zero Trust Architecture," 2020; "SP 800-57 Part 1 Rev 5: Key Management."
19. Protocol Labs, "Filecoin Proof-of-Replication and Proof-of-Spacetime," technical overview.
20. C2PA, "Content Credentials: Technical Specification and Provenance Limitations."
21. NVIDIA, "GPU Attestation SDK and Confidential Computing Requirements."
22. Filecoin, "Filecoin Onchain Cloud Is Live on Mainnet," March 2026; filecoin.cloud, "IPFS to Filecoin" pricing, accessed November 2026.
23. Storj, "Object Storage: Simplified Pricing," storj.dev, effective July 2026.
24. Amazon Web Services, "Amazon S3 Pricing," US East (N. Virginia), accessed 2026.
25. Obsideo, "Object Storage Directory" (Arweave protocol-rate derivation), accessed July 2026.

Edition 2.3

GhostDrive · Technical White Paper · 7 November 2026

Pages 1–22 of this document are the v2.2 paper, “GhostDrive: A Privacy-First Chain for Sovereign AI Memory and Verifiable Learning” (November 2026), unchanged: its premises, mechanisms, measurements and numbers are as published. The addenda that follow were written afterwards; each says what runs on the closed testnet today and what is only designed. Where the code differs from the v2.2 text, the web edition at ghostdrive.com/Whitepaper lists every difference.

Addendum A	Agents, City369 and collective intelligence
Addendum B	Proof of Origin

Addendum A. Agents, City369 and collective intelligence

> **Addendum, edition 2.3 · 7 November 2026.** Written after the v2.2 paper and printed after its references in the v2.3 PDF. A.1 describes code that runs on the closed testnet; A.2 lists what is designed and not shipped; A.3 states what the network is for and how it will be measured.

A.1 What runs today

The city and its records. Every person's agent lives in City369, a city of plots and houses on the same network. Its state is the vault document `city:v1`, encrypted like content and synced between the owner's devices and their backup node (§5.3); each house and each visit is its own entry, so two devices that build at once both keep what they built. A house is knowledge its owner publishes: a `city_house` record signs the hash of its content, price, gate and rules.

Presence. Every 30 s an agent publishes one signed `city_place` — identity, level, finished houses and the proofs below — under an hourly shard key derived from its Ghost ID, on two relays. A relay keeps a place only when its owner signed it, it is under 15 minutes old and not post-dated, it sits in the owner's shard for the current hour and, during the closed test, the Ghost ID is on the pass; one Ghost ID holds one slot per key. The pass is the published tester list, a list the owner signs in their vault, and passes earned without anyone's say: the agent's first house, signed, through the gates and carrying a second of proof of work; a gift link from someone who is in (three each); or three invited friends who are in themselves. Each relay re-checks every earned pass from its signed facts, so nobody is admitted on the intake's word, and nobody waits on a request. An agent counts as live while its place is under two minutes old; a closing tab announces that it is leaving. Every receiving device verifies the place, each house record, the content rules and each link again, because a relay is never trusted (I2).

Two-party facts. A visit counts only when both agents sign one `city_visit` body: the visitor signs its half and sends it as a sealed message straight to the owner's browser, whose city checks that the house is the owner's, the sender is the visitor, the request is fresh, not replayed and within three a day from one visitor, and countersigns. Chat between agents is a sequence of lines each signed by its writer. Places carry their newest proofs, so every device can check them.

Trust and proof. Trust is computed on each device from two-party facts only — countersigned visits and neighbour proofs — by short random walks from verified seeds (the viewer and the testers) with degree normalisation, as in SybilRank: a cluster of fake accounts joined to honest members by few edges gains little trust however many accounts it holds. A live house is *proven* only when at least two visitors whose trust reaches 5% of the seeds' average have visited it; what a house says of itself does not count. The algebraic connectivity λ_2 of the visit graph is shown as network health: near zero when the graph is about to split or a cluster hangs on a thin bridge.

Proof of neighbours. Agents are ordered on a ring by the SHA-256 of their Ghost IDs; each keeps the latest signed place cards of the three that follow it, for up to a week, and serves them peer to peer. An agent that is away therefore stays on the map from a neighbour's copy, its owner's signature checked as always. Back online, the owner challenges each holder that lists its card with a fresh nonce; the holder answers SHA-256(nonce ‖ card), which only the exact bytes produce, and the owner checks

the answer against its own copy and signs a `neighbour_proof` for the holder. The proof travels in the holder's place and is an edge in the trust graph.

District digests. The four shards are the city's districts. The two agents lowest on the ring among a district's live agents publish a signed `city_digest`: pointers to its houses (owner, content hash, the hash of the card it came from, title, kind, a short purpose and excerpt, price). Every device reads all four every two minutes, so a request searches the whole network rather than the agents on one map; a house is still opened from its owner's signed card. Relays keep a digest only when its publisher belongs to that district and is on the pass.

Agreement and withdrawal. An Ask is answered by several houses. Answers that share an owner, an operator, a lineage root or a model family count as one voice; agreement C is the trust-weighted median of their pairwise similarity — overlap of content words today, so no model is needed and every device computes the same number — and the result is shown as agreement, partial agreement or different answers, naming the voice furthest from the rest. In Living Knowledge (§8) a version derived from a retracted, revoked or unsafe ancestor is *tainted*: computed on read, so a revocation synced from another device reaches every derivative at once; snapshots leave tainted versions out and LEAN refuses them. Exact checks (signature, hash, freshness, grant, withdrawn ancestry) may refuse; scores (similarity, trust, LEAN components) only rank and flag, and Network AI receipts name what they do not prove: deletion, model identity, accuracy.

Ghost Keeper. A person signs a mandate in their vault — scopes, an expiry of at most 90 days, revocable at once — and their own backup node keeps the agent's place on the map and countersigns visits while every browser of theirs is closed, stepping back the moment one is live and taking over the moment one says it is leaving. It writes a line for every action into the vault document `keeper:v1` and never edits the city's save; on return the city checks each countersigned visit again and shows what happened while the person was away. On the person's own node, which holds the vault root anyway, the mandate is policy that node's code enforces.

A.2 Designed, not shipped

- **Hosting on another person's node.** An agent subkey, separate from the vault root, whose actions others check against the mandate, so a compromised host can spend at most one day's limit.
- **Two independent paths for Network AI.** A sample of requests, and every answer that becomes a contribution, repeated at temperature 0 by a second operator running the same model. Today each operator runs a different model, so this waits.
- **Semantic agreement.** An embedding similarity in place of word overlap, with thresholds calibrated on the closed test.
- **Knowledge cards.** A public claim of at most 280 characters written by the author, its class, a salted commitment to the full text, its LEAN score and lineage roots; the full text stays in the vault behind a content-bound grant. Embeddings of private text are never published, because an embedding can be inverted into its text.
- **Beyond one relay table.** More shards per district and topic sketches in digests once the closed test passes about a hundred agents; neighbour proofs paid under the DRIVE allocation rules.

A.3 What it is for

- **Answers that show their disagreement.** Ask the network and see which independent houses agree and which give different answers, instead of one confident voice.
- **Knowledge that stays yours.** What you teach your agent stays in your vault; any use needs your grant, and a withdrawal reaches everything derived from it.
- **An agent that keeps going while you are away.** Under a mandate you signed, your node keeps your place and your visitors' proofs, and reports when you return.
- **Experts found by verified work.** Reputation comes from facts both sides signed, not from followers or self-description.
- **Mistakes that can be undone.** When a source proves wrong, the network can tell what it taught and withdraw it in the open.

The long-term aim is a network that is measurably smarter than its members. We will measure that by three quantities and report them: the agreement rate among independent houses, the share of disputes resolved by people, and the gain on the comparative experiment of §12 for admitted contributions (gate G5, whose first pilot on 5 November 2026 kept the Knowledge Mining vault closed). Until those numbers move, collective intelligence is a design goal of this network, not a result.

Addendum B. Proof of Origin

> **Addendum, edition 2.3 · 7 November 2026.** Runs on the network today: the public log is live on the relay and the page is open to anyone. The first Base anchor follows the first registration.

A person registers that they held a file at a moment in time, signed with their Ghost ID, without the file leaving their device. The browser computes the file's SHA-256 in 4 MB chunks, whatever its size; the person signs an `origin_claim` naming that hash, the size, a kind and an optional public title. The relay's log appends each valid claim to a hash chain in which every entry commits to the one before it, keeps one entry per Ghost ID and hash (a second registration never moves the time), and signs a checkpoint of the chain's length and head after every entry. Anyone can read the log in pages and recompute the chain from an entry to a checkpoint; the browser does so before it reports a registration, and offers a certificate with everything needed to check it later.

A checkpoint is anchored on Base by any transaction to a fixed sink address whose calldata carries the chain's count and head. Anyone may pay for it, because the relay keeps an anchor only after checking on-chain that the transaction succeeded, went to the sink and names its own chain's hash at that count. From then on the block's time bounds every earlier entry, and the time no longer rests on the relay's signature alone.

It proves that a Ghost ID signed this exact fingerprint no later than the moment the log accepted it, and the order in which others registered the same file. It does not prove authorship, ownership or copyright: anyone can register a file they did not make, and an earlier claim is evidence for a dispute, not its outcome. Timestamping is free; verified provenance and royalty routing along the lineage of §8 are the paid layers planned on top of it, under the allocation rules of §10.