Memory Is Not the Lens

The LCSM–FOVEA dual-plane architecture for canonical truth and selective context

Hayden Lindley · Glyphd Labs

2026-07-27

Contents

Memory Is Not the Lens

Abstract

Persistent AI systems need both durable truth and selective context. These requirements are often collapsed into one "memory" subsystem: a vector store, a graph, a spatial map, a long transcript, or a model-managed summary. The collapse is dangerous. The mechanism that decides what is canonical should not be the same mechanism that decides what is nearby, visible, or currently worth loading.

This paper defines the **LCSM–FOVEA dual-plane architecture**. LCSM is the authority-bearing plane for canonical state, versions, provenance, receipts, disputes, and reconstruction. FOVEA is the disposable optical plane for address, locality, projection, navigation, landmarks, and foveated working sets. An object may be placed, summarized, highlighted, or omitted by the lens without changing what the truth plane recognizes. The entire optical plane can be deleted and reconstructed from the ledger.

The specific contribution is not event sourcing, quadtrees, or foveated retrieval in isolation. It is their disciplined composition around one asymmetry: **the lens depends on memory; memory never depends on the lens.** This gives AI systems spatial and attention-scaled interfaces without allowing a projection to become a hidden second authority.

1. Why one "memory" layer is not enough

The word memory is used to describe incompatible operations:

- retaining an exact source;
- remembering a user preference;
- choosing which prior messages to retrieve;
- preserving an accepted decision;
- storing a model summary;
- caching a prompt prefix;
- placing a project in a visual map;
- computing semantic similarity;
- reconstructing a prior state;
- displaying a local neighborhood.

When these operations share one implementation and one identity, convenience turns into semantic debt.

A vector store can find related passages but cannot determine which one is binding. A spatial map can make an object easy to find but cannot establish that the object is true. A long context window can expose history but cannot distinguish an accepted decision from a rejected proposal. A cache can preserve expensive computation but should not become the source of personal memory. A model-generated summary can reduce tokens while silently removing nuance or authority.

The dual-plane architecture begins by refusing to call all of this memory.

2. Plane A: LCSM

The LCSM plane answers:

> What does the system presently recognize as true, accepted, disputed, superseded, expired, simulated, or unresolved—and why?

Its responsibilities include:

- logical object identity;
- immutable versions;
- current canonical pointers;
- accepted decisions;
- source and evidence references;
- authority;
- policy;
- permissions;
- dispute and uncertainty;
- branch identity;
- journal order;
- receipts;
- historical reconstruction.

LCSM can be implemented over different storage technologies. The architectural obligation is behavioral: state transitions are governed, append-preserving, inspectable, and reconstructable.

LCSM does not decide where an object should appear on a visual field or how much of it should enter the next model prompt.

3. Plane B: FOVEA

The FOVEA plane answers:

> Where is the object in this projection, what is near it, what level of detail should be loaded, and how can a person or model return to it?

Its responsibilities include:

- stable spatial addresses;
- exact address arithmetic;
- recursive regions;
- Hilbert-derived locality;
- procedural landmarks;
- `LOOK` state;
- foveation rings;
- viewport budgets;
- peripheral glyphs;
- local context manifests;
- projection skins;
- prefetch and eviction.

FOVEA does not decide whether an object is accepted, authoritative, or true. It displays proof state received from LCSM.

4. The dependency law

The architecture has one immutable direction:

```
LCSM canonical state
        |
        | events, objects, receipts
        ▼
FOVEA materialized projection
```

The reverse path may contain user gestures or proposals:

```
LOOK / select / move request / pin request
        │
        ▼
proposed journal operation
        │
        ▼
LCSM validation and authorization
```

The gesture does not directly mutate canon. It requests an operation through the truth plane.

This yields the foundational law:

> **The lens depends on memory. Memory does not depend on the lens.**

An informational atlas hint may appear in a receipt, but the validity of the receipt cannot require that the atlas exists. A canonical object can be resolved without loading its spatial projection.

5. Identity separation

A mature implementation distinguishes at least five identities:

1. **Logical identity** — the durable object across versions.
2. **Exact content identity** — the cryptographic identity of one exact payload.
3. **Representation identity** — the identity of one serialized or compressed encoding.
4. **Spatial address** — the place assigned in one address fabric.
5. **Projection identity** — the specific rendered view or LookState.

Example:

```
Project decision D-42
  logical_id: decision:D-42
  current_content_id: sha256:...
  fold_id: fold2:...
  fovea_address: glyph://field/0123301
  surface_view: decision-inspector@v3
```

Changing the decision text creates a new content ID. Re-encoding it creates a new representation ID. Changing a skin creates a new projection. None of these should silently create a new logical decision. A move request may create a new address receipt without rewriting content.

This separation is central to both correctness and reuse.

6. The Gantry boundary

LCSM and FOVEA should not import each other's internals freely. They compose through a **Gantry** boundary: explicit contracts that translate governed state into an addressable projection and projection gestures into proposed ledger operations.

A placement envelope may contain:

```
canon_ref
content_id
proof_ref
address
projection_class
visibility
```

```
summary_ref?
render_hint?
```

The Gantry validates:

- logical object exists;
- referenced exact content exists;
- receipt and visibility are valid;
- address is canonical and within bounds;
- projection metadata cannot override proof state;
- private objects do not enter public fields;
- derived summaries retain exact source links.

The boundary is intentionally boring. Its purpose is to prevent the attractive visual layer from acquiring hidden powers.

7. Reconstruction invariant

The strongest implementation test is destructive:

1. Create canonical objects and placement events.
2. Build the FOVEA atlas and local indexes.
3. Delete every atlas cache, address range cache, landmark texture, and viewport state.
4. Replay canonical journal events through the Gantry.
5. Reconstruct placements and proof states.
6. Compare the logical projection graph and golden address vectors.

The reconstructed pixels need not be bit-identical if a skin has changed. The logical state must be equivalent under the declared projection version.

If the system cannot rebuild, the optical plane has become a second database.

8. Synchronization

Multiple views can project the same canonical object:

- Backboard note;
- FOVEA fold;
- mobile action strip;
- Workpage block;
- public proof page;
- agent PromptCapsule.

Synchronization uses canonical versions and event subscriptions.

A view declares:

```
source_refs
source_versions
projection_version
freshness
omitted_fields
authority_context
visibility_policy
```

When a source version changes, a view becomes stale, refreshes deterministically, or requests model-assisted regeneration. A stale view must not continue to appear current.

FOVEA can show staleness as a glyph or boundary state. It cannot decide that a stale summary is “close enough” to canon.

9. Foveated context construction

The dual-plane architecture is especially useful for model context.

A model request starts from an active canonical object. FOVEA constructs a bounded neighborhood:

```
active object
parents and children
siblings
nearest exact evidence
authorizing and blocking state
recent mutation trail
unresolved conflicts
high-salience landmarks
required dependencies
```

Each entry carries a canonical reference and proof state. The context compiler can emit:

- exact source spans;
- compact summaries;
- omission warnings;
- a token budget;
- retrieval reasons.

The model receives a **lens-shaped working set**, not ownership of memory.

This preserves a clean responsibility split:

- LCSM determines what is current and allowed.
- FOVEA determines the neighborhood and level of detail.
- PromptCapsules serialize the selected working set.
- the model proposes an output.
- validators and LCSM govern any state change.

10. Why not use search alone?

Search remains necessary. FOVEA is place; search is query.

A search result can illuminate or navigate to addresses on the field. It may rank candidates by relevance. The field preserves location, neighborhood, and spatial return. Neither subsumes the other.

Search is often superior for named retrieval. FOVEA's central value proposition concerns:

- resuming work by place;
- preserving context around an active object;
- perceiving nearby change;
- moving bounded neighborhoods;
- maintaining a sense of project shape;
- presenting multiscale context to models and people.

The benchmark must allow search to win where it is the better tool.

11. Why not use the model as the unifying plane?

A model can synthesize state and choose context, but it should not become the final owner of either.

If the model owns canon:

- its summary may drift;
- a provider change can alter memory;
- authority becomes prompt text;
- exact sources may disappear;
- replay becomes impossible.

If the model owns the lens:

- retrieval is hard to reproduce;
- omissions are hidden;
- context budgets are not inspectable;
- user spatial memory cannot stabilize;
- cache invalidation becomes opaque.

Models can help classify, summarize, place, or propose neighborhoods. Their outputs remain versioned derived objects.

12. Failure containment

The dual-plane architecture limits failures.

Optical cache corruption

Delete and reconstruct from LCSM.

Bad placement

Append a move or supersession receipt; preserve historical address lineage.

Misleading summary

Mark the derived fold stale or breached; exact canonical source remains intact.

Unauthorized gesture

Gantry rejects the proposed journal operation.

Model hallucination

The proposal fails source, schema, or authority checks; canon does not move.

Canonical corruption

This is more severe. Quarantine affected state, stop trusted projection, replay from a known-good checkpoint, and preserve evidence. FOVEA cannot repair canonical truth.

13. Evidence state

The dual-plane architecture is specified across the LCSM, FOVEA, GlyphCAS, SurfaceMind, and MCE-1 corpora.

Mechanism-level evidence includes:

- deterministic address operations;
- Hilbert and URI invariants;
- proof-deck tests;

- LCSM version and journal contracts;
- reconstructable materialized-view doctrine;
- exact source and semantic identity separation in GlyphCAS;
- SurfaceDelta optimistic concurrency;
- public/private projection rules.

A fully integrated production implementation with destructive atlas reconstruction, cross-language parity, live model context compilation, and long-duration user evaluation still requires a complete accepted receipt.

14. Evaluation program

System integrity

- deleting the lens loses zero truth;
- every trusted fold resolves to valid proof;
- canonical schemas contain no required FOVEA identity;
- stale versions do not render as current;
- public projections cannot resolve private objects;
- gestures cannot bypass authority;
- all projection state can be rebuilt.

Model performance

Compare transcript dump, semantic retrieval, and dual-plane context on:

- constraint retention;
- citation accuracy;
- obsolete-state inclusion;
- prompt tokens;
- latency;
- repair rate;
- task success.

Human performance

Compare list/search and FOVEA projections on:

- time to resume;
- place refind;
- named lookup;
- peripheral change detection;
- navigation errors;
- comprehension of trust and staleness;
- accessibility.

Operational performance

Measure:

- bytes moved per LOOK;
- range request count;
- cache hit;
- rebuild time;
- event-to-projection latency;
- token movement;

- cold and warm state.

15. Security and privacy

The separation does not automatically guarantee privacy. The Gantry must enforce it.

Risks include:

- spatial adjacency leaking relationships;
- public map caches containing private summaries;
- stale permission projections;
- proof URLs exposing confidential identifiers;
- address enumeration;
- cross-project context expansion;
- malicious imported folds;
- renderer spoofing of trust states.

Controls include:

- visibility in every placement envelope;
- separate projection namespaces;
- permission-version invalidation;
- proof-derived trust chrome;
- bounded traversal;
- redacted public IDs;
- no address-as-capability assumption;
- cache purge and replay tests;
- content security policies;
- source and authority checks before context export.

16. Limitations

The architecture creates more explicit systems than a single vector database or memory service. That complexity is intentional but costly.

Some applications do not need stable place. A small task list may be better served by a list. A project with weak canonical discipline may not benefit from a sophisticated lens. FOVEA's cognitive value remains a hypothesis.

The boundary between truth and lens also does not solve every classification problem. Deciding which note is canonical, which objects are neighbors, or which summary is sufficient still involves domain judgment.

Finally, strict one-way dependency can be inconvenient. Designers may want a visual move to “just change” an object. The architecture requires that the gesture become a governed operation because convenience is not worth hidden authority.

17. Conclusion

A persistent intelligence needs memory and attention, but they are not the same thing.

LCSM preserves what is recognized and why. FOVEA determines where it appears, what is near it, and how much enters the current view. The first must survive the destruction of the second.

The architecture can be summarized in two sentences:

> **Truth lives in canon. Nearness lives in the fabric.**

Memory is not the lens.

Architecture illustration briefs

- **P04-F01 — The dual-plane architecture**: Upper LCSM truth plane containing objects, versions, authority, receipts. Lower FOVEA lens plane containing addresses, neighborhoods, rings, and projections. One-way dependency emphasized.
- **P04-F02 — Five identities, one object**: A logical decision linked to exact content hash, fold representation ID, FOVEA address, and view ID, with independent version-change arrows.
- **P04-F03 — Gantry transition**: FOVEA gesture becomes a proposed journal operation, passes schema/visibility/authority checks, then updates LCSM and regenerates projections.
- **P04-F04 — Destructive reconstruction test**: Delete atlas cache → replay journal → rebuild addresses/proof → compare golden logical projection.

Source register

- **P04-S01** — LCSM Workbench E2E Specification (2026-07-04). `LCSM-WORKBENCH-E2E-SPEC.v0.md`
- **P04-S02** — GlyphDrive FOVEA Specification v0.2 (2026-07-12). `GLYPHDRIVE-FOVEA-SPEC-v0.2.md`
- **P04-S03** — GlyphDrive v0.2 Ratification Gate R0 (2026-07-12). `PASTE-INTO-OPUS-GLYPHDRIVE-V02-RATIFICATION-R0.md`
- **P04-S04** — GlyphCAS Design Report (2026-07-24). `How_we_can_make_CAS_lightning_fast_for_any_LL`
- **P04-S05** — MCE-1 System Contract (2026-07-25). `MCE-1-SYSTEM-CONTRACT-v0.1.md`

Public-disclosure and IP boundary

This paper publishes the public-safe architectural boundary and testable invariants. It excludes confidential mapping heuristics, private security configuration, unpublished patent drafting, and internal user-state schemas. It does not claim that the dual-plane architecture has yet achieved measured model uplift or human-navigation superiority.