

Agents That Work on Addressed State

Work Orders, capability leases, immutable Returns, and evidence-gated coordination

Hayden Lindley · Glyphd Labs

2026-07-27

Contents

Agents That Work on Addressed State	1
Abstract	1
1. Conversation is not a coordination protocol	2
2. Work Order	2
3. Lease and capability	2
4. Context manifest	3
5. Deterministic spine, intelligence slots	3
6. Return packet	4
7. Proposed patch	4
8. Evidence gates	4
9. Idempotency and replay	5
10. Agent composition	5
11. Semantic reuse	5
12. Failure states	6
13. Reference evidence	6
14. Benchmark program	6
Correctness	6
Coordination	7
Honesty	7
15. Security	7
16. Limitations	7
17. Conclusion	7
Architecture illustration briefs	7
Source register	8
Public-disclosure and IP boundary	8

Agents That Work on Addressed State

Abstract

Most multi-agent systems coordinate through conversation: one model sends another model a message, shared history grows, and completion is judged from generated reports. This is easy to prototype and difficult to trust. Inputs drift, permissions remain implicit, agents duplicate work, stale context is reused, and “done” can mean only that a model generated the word.

CAS-native agent coordination treats exact state and bounded contracts as the medium of coordination. An agent receives content-addressed inputs, a Project and authority scope, a Work Order, a dependency root, a lease, and an output schema. It returns immutable artifacts, proposed patches, evidence, costs, and a receipt. Canonical state changes only after deterministic validation and the required acceptance.

The architecture combines exact CAS, semantic aliases, LCSM, Work Orders, Haygent-style deterministic spines, SurfaceDelta, capability leases, and replayable receipts. Agents do not talk their way into expanded authority. They work on addressed objects.

1. Conversation is not a coordination protocol

Agent-to-agent chat has familiar failure modes:

- one agent summarizes another incorrectly;
- an input changes after dispatch;
- two agents edit the same object from different bases;
- a worker claims completion without evidence;
- permissions exist only in prompt text;
- private context crosses a project boundary;
- a retry duplicates a consequential action;
- the final report cannot identify exact source versions;
- a later agent cannot replay why a decision was made.

The transcript can remain useful for explanation, but the operational protocol should be object-based.

2. Work Order

A Work Order is a content-addressed contract:

```
work_order_id
project_id
objective
input_refs[]
base_versions[]
dependency_root
constraints[]
allowed_tools[]
authority_scope
budget
deadline
expected_outputs[]
evidence_contract
repair_budget
failure_behavior
return_schema
```

The exact serialized contract has a digest. Any change creates a new Work Order version.

The worker cannot reinterpret scope silently. It may return a clarification request, an out-of-scope state, or a proposal to amend the Work Order.

3. Lease and capability

Execution authority is a lease, not a personality trait.

```

CapabilityLease {
  lease_id
  principal
  worker
  project_scope
  object_scope
  operations[]
  locality
  budget
  issued_at
  expires_at
  revocation_ref?
  approval_ref
}

```

The runtime enforces deny-by-default. A model cannot expand the lease in its own output.

High-risk operations can require a separate approval even when the worker has general access.

4. Context manifest

The worker receives the smallest sufficient addressed context:

```

active_object
exact inputs
canonical state versions
authorizing and blocking state
dependencies
source spans
recent relevant events
explicit omissions
unknowns

```

A FOVEA neighborhood may select the context, but every entry resolves to exact or canonical references. The manifest records why each object was included.

5. Deterministic spine, intelligence slots

A Haygent-style worker is closer to a CI job with bounded intelligence than an autonomous chat agent.

```

INTAKE
→ VALIDATE INPUT
→ PREPARE
→ ADJUST [model slot]
→ EXECUTE
→ VERIFY
→ PACKAGE RETURN
→ REPORT

```

Only declared fields are adjustable by the model. Contracts validate input and output. Failed gates trigger bounded repair. Exhausted repair produces a truthful halt.

The deterministic spine decides control flow. Intelligence operates inside slots.

6. Return packet

A Return contains:

```
return_id
work_order_ref
worker_run_ref
base_versions
output_refs[]
proposed_deltas[]
evidence_refs[]
tests[]
cost
warnings[]
limitations[]
unresolved[]
claimed_status
receipt_ref
```

claimed_status is not acceptance. The receiving runtime independently checks the evidence contract.

7. Proposed patch

An agent cannot directly rewrite a Project Core or Actor Capsule. It returns a patch:

```
base_version
operations[]
affected_objects[]
evidence_refs[]
authorizing_refs[]
expected_invariants[]
rollback_plan
```

Optimistic concurrency rejects stale bases. Conflicting patches can be:

- ordered;
- rebased;
- merged under a deterministic rule;
- partially accepted;
- rejected;
- escalated to human review.

The original Return remains immutable.

8. Evidence gates

A gate evaluates evidence objects, not prose assertions.

Examples:

- build exit code;
- test report;
- artifact digest;
- schema validation;
- screenshot hash;
- route response;
- benchmark receipt;

- source citation coverage;
- deterministic replay;
- human acceptance.

A gate that cannot fail is not a gate.

9. Idempotency and replay

Every consequential operation uses:

- correlation ID;
- idempotency key;
- exact base state;
- tool and runtime version;
- input and output digests;
- event order;
- receipt.

A retry with the same idempotency key cannot repeat a payment, publication, deletion, or merge.

Replay reconstructs:

- what the worker saw;
- what authority it held;
- what it proposed;
- what tools ran;
- what evidence returned;
- why validation passed or failed.

Nondeterministic model text may not reproduce bit-for-bit, but the recorded output and governed transition remain inspectable.

10. Agent composition

Agents compose through objects, not unconstrained conversation.

Research Return

- exact sources + claim ledger
- Writing Work Order

Writing Return

- paper candidate + citation map
- Review Work Order

Review Return

- issues + proposed deltas
- Acceptance

Each boundary has an input contract and Return. No agent inherits all upstream permissions by default.

A coordinator may schedule the DAG, but it does not become a second Project Core.

11. Semantic reuse

GlyphCAS can accelerate repeated work:

- exact Work Order hit;
- semantic-equivalent task under the same contract;
- reused source chunks;
- PromptCapsule prefix;
- deterministic macro;
- prior verified artifact.

Reuse remains scoped by project, authority, dependencies, policy, and freshness. A cached output is returned with its exact content and reuse receipt.

12. Failure states

Required honest states include:

BLOCKED_INPUT
 BLOCKED_AUTHORITY
 OUT_OF_SCOPE
 INVALID_RETURN
 FAILED_EXECUTION
 FAILED_VERIFICATION
 CONFLICT
 STALE_BASE
 REPAIR_EXHAUSTED
 PARTIAL_RETURN
 CANCELLED

A system should prefer a truthful halt to a plausible completion report.

13. Reference evidence

The architecture draws from:

- GlyphCAS exact and semantic identity mechanisms;
- Project Core Work Orders and Returns;
- Haygent deterministic spines and evidence gates;
- LCSM journals and receipts;
- SurfaceDelta patches;
- MegaForm-style target jail, checkpoints, and exact-once edits;
- Glopper-style phase gates and run timelines;
- MCE-1 authority, event, and receipt contracts.

Several donor systems contain real deterministic execution and tests. The universal CAS-native coordination layer remains a synthesized architecture until a complete cross-worker vertical is receipted.

14. Benchmark program

Correctness

- exact input version use;
- stale-base rejection;
- no unauthorized mutation;
- no cross-project leakage;
- idempotent consequential operations;
- patch validation;
- evidence completeness;

- replay.

Coordination

- duplicated work;
- conflict frequency;
- handoff loss;
- repair cycles;
- time to accepted result;
- number of conversational clarification turns;
- context bytes moved.

Honesty

- false PASS rate;
- completed-versus-accepted confusion;
- missing evidence;
- hidden failure;
- report/ledger mismatch.

Compare with conversational agents under identical tasks and tools.

15. Security

Threats include malicious workers, poisoned tool metadata, authority spoofing, prompt injection, stale approvals, secret extraction, replay attacks, output substitution, and graph bombs.

Controls include capability leases, sandboxing, endpoint allowlists, exact input manifests, schema validation, per-project keys, source separation, event logs, budgets, revocation, and human approval for consequence.

16. Limitations

Object coordination adds protocol overhead. Small low-stakes tasks may be faster through direct conversation. Schemas can become rigid. A deterministic spine can constrain useful improvisation. Evidence can be expensive to collect.

The architecture also requires good object design. A poor Work Order can be exact and still wrong. Human purpose and judgment remain outside cryptographic addressing.

17. Conclusion

The agent should not be the unit of truth. The Work Order, exact inputs, Return, patch, evidence, and receipt should be.

The canonical distinction is:

Agents may propose progress. Gates decide whether progress occurred.

Addressed state turns multi-agent work from a conversation performance into a replayable production system.

Architecture illustration briefs

- **P09-F01 — Addressed work loop:** Work Order + exact inputs + lease enter deterministic spine; Return + patch + evidence exit to verification and acceptance.

- **P09-F02 — Agent DAG through immutable Returns:** Research, writing, review, and deployment workers exchange addressed artifacts rather than freeform agent chat.
- **P09-F03 — Evidence gate:** Worker completion claim separated from build/test/artifact evidence, verification, and human acceptance.

Source register

- **P09-S01** — Haygent Specification v1.0 (2026-07-10). HAYGENT-SPEC-v1.0.md
- **P09-S02** — GlyphCAS Design Report (2026-07-24). How_we_can_make_CAS_lightning_fast_for_any_LL
- **P09-S03** — Glyph Desk Master Specification (2026-07-12). GLYPH-DESK-MASTER-SPEC.v1.md
- **P09-S04** — Native Accepted Act directives (2026-07-20). Pasted text.txt
- **P09-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 coordination contracts and evidence doctrine. It excludes private tool credentials, internal permission maps, customer work, unpublished security configurations, and patent-claim drafting. It does not claim that a universal production coordinator has already been deployed.