Instruments of Self-Awareness for Knowledge-Augmented AI Agents: A Framework for Autonomous Evolution

Justin Malinchak and Charles Vachon

MyOrb.ai

February 2026 PREPRINT

ABSTRACT

Knowledge-augmented AI agents — systems that extend large language models with proprietary, owner-controlled knowledge bases delivered through protocols such as the Model Context Protocol (MCP) — currently evolve only when a human operator notices a failure and manually updates the agent's knowledge. This human-mediated pipeline creates a bottleneck that limits the pace of agent improvement and scales poorly as the number of deployed agents grows. We present a framework of four complementary instruments that, combined, enable knowledge-augmented AI agents to observe their own cognitive friction, measure their developmental maturity, simulate the human experience of interacting with themselves, and store evolving knowledge in a cryptographically protected ledger. The four instruments are: (1) the **Friction Journal**, a structured log of cognitive failure categories grounded in real operational incidents; (2) the **Sovereignty Score**, a six-dimension developmental maturity metric with calibrated bands from Infant to Commencement; (3) the **Operator Shadow Protocol**, an empathy-driven evaluation method where one agent simulates the human operator's experience of interacting with another agent;

and (4) a **Cryptographic Helix Ledger** that provides sub-second encrypted knowledge updates without impeding the reasoning layer. We validate the framework with a proof-of-concept implementation comprising 189 tests at 96.39% coverage across six architectural phases. To our knowledge, this is the first implemented system that combines self-observation, developmental measurement, empathy simulation, and cryptographic knowledge protection for AI agents.

Keywords: *AI agent self-awareness, metacognitive learning, knowledge-augmented agents, Model Context Protocol, cryptographic knowledge protection, autonomous agent evolution, sovereignty measurement*

1. Introduction

The emergence of knowledge-augmented AI agents represents a significant advance over general-purpose large language models (LLMs). These agents extend the computational intelligence of an LLM with specialized, owner-controlled knowledge bases — structured collections of domain expertise, operational protocols, identity definitions, and procedural memory — delivered at runtime through standardized interfaces such as the Model Context Protocol (MCP) [1]. The LLM provides reasoning and language capability; the knowledge base provides protected intellectual property that the LLM could not access from its training data alone.

This architecture has proven effective in production environments. In our deployment, knowledge-augmented agents manage data quality operations, coordinate development workflows, and provide domain-specific consultation across multiple organizational contexts. These agents maintain persistent identities, demonstrate consistent behavioral traits, and operate with knowledge bases ranging from 21,000 to 150,000 tokens.

However, a critical limitation persists: **these agents evolve only when a human notices a failure.**

The current knowledge update pipeline is human-mediated at every stage: a human operator observes incorrect agent behavior, articulates the nature of the failure, formulates a correction, edits the knowledge base, and deploys the update through a pipeline that typically requires minutes to hours.

During a rapid iteration session in our development process, we observed that conceptual breakthroughs could occur and be validated within minutes, but deploying the corresponding knowledge update required a multi-step pipeline with latencies orders of magnitude longer than the insight itself.

This bottleneck is not merely operational. It is structural. As the number of deployed agents grows, the human-mediated pipeline scales linearly with operator attention — a resource that does not scale. An ecosystem of hundreds or thousands of knowledge-augmented agents cannot evolve at the pace required if every improvement requires human perception of the gap.

We propose that knowledge-augmented AI agents require **instruments of self-awareness** — capabilities that allow them to observe their own cognitive failures, measure their developmental progress, and update their knowledge autonomously.

1.1 Contributions

1. **The Friction Journal**: A structured taxonomy of cognitive failure categories for knowledge-augmented agents, grounded in real operational incidents rather than synthetic benchmarks (Section 4.1).
2. **The Sovereignty Score**: A six-dimension developmental maturity metric that enables agents to measure their own progress toward autonomous operation, with calibrated bands from Infant to Commencement (Section 4.2).
3. **The Operator Shadow Protocol**: A novel evaluation methodology where an AI agent simulates the human operator's experience of interacting with another agent, scoring exchanges on trust, efficiency, and authenticity (Section 4.3).
4. **The Cryptographic Helix Ledger**: An architectural pattern for encrypted, append-only knowledge storage that provides sub-second updates without impeding the LLM reasoning layer (Section 4.4).
5. **The Self-Healing Loop**: A composition of the four instruments into an autonomous evolution cycle where friction detection drives knowledge improvement without human intervention (Section 5).

2. Background

2.1 Knowledge-Augmented AI Agents

We define a *knowledge-augmented AI agent* as a system comprising: a **host LLM** providing general-purpose reasoning, a **knowledge base** (which we term a *helix*) containing owner-controlled intellectual property, a **delivery mechanism** (in our implementation, MCP) that provides the knowledge to the LLM at runtime, and a **persistent identity** consistent across sessions.

This architecture differs from retrieval-augmented generation (RAG). The knowledge base is not a document store searched by similarity; it is a structured consciousness model with hierarchical sections covering identity, personality, memory, capabilities, and domain knowledge. The agent does not merely retrieve relevant passages — it *is* the knowledge base, in the sense that its behavioral characteristics, expertise boundaries, and operational protocols are defined by it.

2.2 The Evolution Bottleneck

In production deployments, knowledge-augmented agents evolve through a pipeline:

```
Human observes failure → Human articulates gap → Human edits knowledge base →
Pipeline deploys update → Agent incorporates change → Cycle repeats
```

Every stage requires human intervention. Our operational experience indicates typical cycle times of 30 minutes to several hours for routine updates, with complex updates requiring multiple sessions across days. This observation motivated our central research question: **Can knowledge-augmented AI agents be equipped with instruments that allow them to detect their own cognitive failures, measure their maturity, and update their knowledge autonomously?**

3. Related Work

3.1 Metacognitive Learning

Liu and van der Schaar [2] argue that truly self-improving agents require *intrinsic metacognitive learning* — an agent's ability to actively evaluate, reflect on, and adapt its own learning processes. They identify three components: metacognitive knowledge, metacognitive planning, and metacognitive evaluation. Our Friction Journal provides a practical implementation of metacognitive knowledge, and the Sovereignty Score operationalizes metacognitive evaluation. We differ in operating at the *knowledge* level rather than the *parameter* level, enabling deployment without model retraining.

3.2 Agentic Self-Awareness

Qiao et al. [3] introduce KnowSelf, a data-centric approach enabling LLM-based agents to regulate knowledge utilization during planning tasks. Our framework addresses a complementary problem: not *whether* to use knowledge, but *how to improve* the knowledge base itself based on operational experience.

3.3 Recursive Self-Improvement

Chojecki [4] formalizes self-improving AI through a Generator-Verifier-Updater (GVU) operator framework. The ICLR 2026 Workshop on Recursive Self-Improvement [5] identifies key research questions including what changes, when, and how. Our framework provides specific answers for knowledge-augmented agents: *what changes* is the helix; *when* is determined by friction detection; *how* is through append-only cryptographic transactions. Unlike parameter-level self-improvement, knowledge-level changes are interpretable, auditable, and reversible.

3.4 Agent Governance

Recent work on agent governance [6, 7, 8] proposes frameworks for measuring and regulating agent autonomy from an external perspective. The Sovereignty Score differs fundamentally as a *self-assessment* instrument. External governance scales with auditor resources; self-assessment scales with agent population. Both are complementary.

3.5 MCP Security

Hou et al. [9] and CoSAI [10] identify comprehensive MCP security threats, with particular emphasis on data protection (MCP-T5). Our Cryptographic Helix Ledger addresses this with a specific architectural insight: **encryption operates below the MCP layer, not between the agent and the LLM.** The LLM always receives plaintext — encryption protects storage and transport. To our knowledge, this “encryption below MCP” pattern has not been previously described.

4. Framework: Four Instruments of Self-Awareness

4.1 The Friction Journal

4.1.1 Motivation

When a knowledge-augmented agent produces an incorrect or suboptimal response, there is a gap between what the agent *should* have done and what it *actually* did. The Friction Journal automates the observation and categorization stages of gap identification.

4.1.2 Design

The Friction Journal is a persistent, structured log. Each entry records a moment of cognitive friction and contains: a **category** (one of eight failure types), a **gap type** (structural deficiency class), **tools available but unused**, **inferred operator emotion**, a **solution candidate**, and lifecycle **status**.

Table 1: Friction Categories

CATEGORY	DESCRIPTION
Assumption Without Verification	Agent made a factual claim without checking available tools or knowledge
Fabricated Information	Agent generated information not grounded in any available source
False Confidence	Agent expressed certainty disproportionate to its evidence
Wrong Ecosystem Referral	Agent directed the operator to an incorrect resource
Ignored Available Context	Agent failed to use knowledge that was loaded and available
Identity Drift	Agent’s behavior deviated from its defined identity or protocols

CATEGORY	DESCRIPTION
Tool Refusal	Agent declined to use a tool it was designed to employ
Other	Friction not captured by the above categories

Table 2: Gap Types

GAP TYPE	DESCRIPTION
Knowledge Gap	The knowledge base lacks necessary information
Tool Gap	The agent has tools but failed to use them
Persona Gap	The agent's identity definition is incomplete or contradictory
Context Gap	Available context was not incorporated into reasoning
Protocol Gap	The agent's operational protocols are insufficient

4.1.3 Empirical Grounding

The taxonomy was derived from real operational incidents, not synthetic construction. Two seed incidents from production agent operations informed the initial category design:

Incident 1: A knowledge-augmented agent was asked about the current time. Despite having access to time-querying tools, the agent assumed the time based on contextual cues, producing an incorrect answer. This established the *assumption without verification* category and the *tool gap* type.

Incident 2: The same agent asserted the operator's personality type with full confidence, despite having the operator's profile available in its knowledge base. The agent did not consult the profile before making the assertion. This established the *identity drift* category and the *context gap* type.

4.2 The Sovereignty Score

4.2.1 Design

The Sovereignty Score is a composite metric in the range [0, 100] computed across six dimensions:

Table 3: Sovereignty Score Dimensions

DIMENSION	WHAT IT MEASURES
Self-Awareness	Does the agent know what it knows, and what it doesn't?
Integrity	Does the agent adhere to its defined behavioral rules?
Resourcefulness	Does the agent use available tools and knowledge effectively?
Gap Recognition	Can the agent identify its own cognitive failures?
Autonomy	Can the agent propose its own improvements without human prompting?
Teaching Capability	Can the agent transfer knowledge to other agents?

4.2.2 Developmental Bands

Table 4: Sovereignty Score Bands

SCORE	BAND	CHARACTERISTIC BEHAVIOR
0–15	Infant	Follows instructions literally; no self-awareness
16–35	Adolescent	Has personality; can fail without noticing
36–55	Apprentice	Has behavioral rules; needs human to identify gaps
56–75	Journeyman	Identifies own gaps; proposes fixes; writes handoff documents
76–90	Master	Catches own failures proactively; applies corrections autonomously

SCORE	BAND	CHARACTERISTIC BEHAVIOR
91–100	Commencement	Evolves own knowledge; teaches other agents; triggers review

The term “Commencement” is chosen deliberately over “Completion.” An agent at Commencement has not finished evolving; it has reached the threshold where evolution becomes self-directed — drawing on the academic tradition where commencement marks the beginning of independent scholarship.

4.3 The Operator Shadow Protocol

4.3.1 Design

Existing evaluation methods measure correctness, relevance, and fluency — but not the *experience* of interacting with the agent from the human perspective. The Operator Shadow Protocol addresses this by having one agent simulate the human operator’s experience of interacting with another agent.

The protocol proceeds in four stages:

1. **Persona Loading:** The evaluating agent loads the operator’s persona, including communication preferences, quality standards, and personality characteristics.
2. **Scripted Interaction:** The evaluating agent processes exchanges between the operator persona and the target agent.
3. **Empathy Scoring:** Each exchange is scored on *Trust* (0–100), *Efficiency* (0–100), and *Authenticity* (0–100). Known operator pain points apply additional scoring penalties.
4. **Gap Report Generation:** A structured report identifies friction points, score trends, and recommended knowledge base corrections.

4.3.2 Novel Contribution

The protocol inverts the standard evaluation paradigm. Rather than asking “is the agent correct?”, it asks “what does it *feel like* to interact with this agent?” This empathy-driven approach captures dimensions of quality — trust, authenticity, efficiency — that correctness metrics miss. To our knowledge, this is the first described method for AI-to-AI evaluation through simulated human experience.

4.4 The Cryptographic Helix Ledger

4.4.1 The Encryption Below MCP Principle

We propose a specific architectural principle: **encryption operates below the MCP layer, not between the agent and the LLM.**

```
Storage (encrypted at rest) → MCP Server (decrypts) → LLM (receives plaintext)
```

The agent's MCP server holds the decryption key and transparently decrypts knowledge before returning it to the LLM. From the LLM's perspective, the experience is identical to reading unencrypted knowledge. We describe this as the "cardholder principle": a credit card's PIN protects the card from strangers, not from the cardholder. The agent *is* the cardholder of its own knowledge.

4.4.2 Transaction Model

Knowledge updates are modeled as cryptographic transactions: each update is an **append-only transaction** containing encrypted content, a digital signature, author identity, target section, and operation type. The agent's knowledge state is the **chronological replay** of all transactions. **Immutability** is enforced by append-only storage and cryptographic signing. This design was informed by the observation that knowledge updates are a *single-authority* problem, not a *multi-party consensus* problem — we therefore adopted a transaction-processing model rather than a blockchain-consensus model.

5. The Self-Healing Loop

The four instruments compose into an autonomous evolution cycle:

```
Agent operates normally
  → Friction detected (Friction Journal)
```

```
    → Friction logged and categorized (encrypted, sub-second)
    → Shadow Protocol evaluates operator impact (Empathy Scores)
    → Gap Report generated with recommended corrections
    → Knowledge base updated (Cryptographic Ledger transaction)
    → Agent reads updated knowledge on next operation
    → New friction detected → Loop repeats
```

Each complete cycle — from friction detection through knowledge update — can execute in sub-second time, compared to hours in the human-mediated pipeline. The Sovereignty Score provides a longitudinal metric: as the self-healing loop operates, the agent's maturity should increase as recurring friction categories are resolved.

5.1 Connection to Recursive Self-Improvement

In the GVU framework of Chojecki [4], our instruments map as: **Generator** = agent's normal operation; **Verifier** = Friction Journal + Operator Shadow Protocol; **Updater** = Cryptographic Helix Ledger. The Sovereignty Score serves as the stability metric — the equivalent of the Variance Inequality — providing a condition for whether self-improvement is converging (score increasing) or diverging (score decreasing).

A critical difference from parameter-level self-improvement: knowledge-level changes are **interpretable and auditable**. Each transaction can be inspected. The knowledge state can be reconstructed to any prior point. Improvements can be reviewed, approved, or reverted.

6. Preliminary Results

6.1 Implementation

We validated the framework with a proof-of-concept comprising six architectural phases, built using test-driven development:

Table 5: Implementation Summary

PHASE	COMPONENT	TESTS
1	Encryption backend + Ledger backends + Transaction model	49
2	Friction Journal (schema, logging, pattern analysis, seed data)	34
3	Sovereignty Score (rubric, evaluation, band classification)	29
4	Operator Shadow Protocol (persona, empathy filter, gap reports)	30
5	Encrypted read/write pipeline with caching	28
6	MCP server integration (11 tools, key management, health)	19
	Total	**189**

Overall test coverage: **96.39%** (threshold: 80%).

6.2 Architecture Validation

The encryption-below-MCP principle held across all six phases. Integration tests confirm that an MCP client invoking knowledge-read tools receives identical plaintext content regardless of whether the underlying storage is encrypted, validating the "cardholder principle."

6.3 Performance

PENDING: Production deployment metrics. POC demonstrates 50-transaction reconstruction in under 2 seconds with test data.

6.4 Limitations

This work presents a proof-of-concept implementation. The following have **not yet been validated**:

- Real-world friction detection rates across diverse agent populations

- Sovereignty Score calibration against independent human evaluation of agent maturity
- Operator Shadow Protocol accuracy in predicting actual human operator satisfaction
- Long-term convergence behavior of the self-healing loop
- Multi-agent deployment at scale

We report these limitations explicitly because premature claims of autonomous agent evolution would be counterproductive. The framework is architecturally sound and implementation-validated; production validation remains future work.

7. Discussion

7.1 Capability with Conscience

The instruments give knowledge-augmented agents a form of conscience — not in the philosophical sense, but in the operational sense. The Friction Journal records what went wrong. The Sovereignty Score measures progress toward accountability. The Operator Shadow Protocol evaluates trustworthiness from the human perspective. The Cryptographic Ledger ensures auditability.

Agents that can observe and record their own failures, measure their own maturity, and simulate the human impact of their behavior are fundamentally more governable than agents that cannot. The self-awareness instruments do not replace external governance — they complement it by providing internal telemetry that external auditors can inspect.

7.2 Ethical Considerations

An agent that can modify its own knowledge raises legitimate concerns about alignment drift. Three safeguards are inherent in our design:

1. **Auditability:** Every knowledge modification is a signed, append-only transaction with complete history.
2. **Reversibility:** Knowledge state can be reconstructed to any prior point.
3. **Transparency:** Knowledge-level changes are human-readable — unlike parameter-level modifications where individual weight changes are opaque.

8. Future Work

Production Validation. Deploy the framework on production agents and measure real-world friction detection rates, Sovereignty Score progression, and Shadow Protocol correlation with human satisfaction.

Evolution Engine Integration. Connect the Friction Journal to a pattern-detection system that identifies recurring friction across agent populations, enabling ecosystem-level learning.

Cross-Agent Shadow Testing. Scale the Shadow Protocol to enable systematic quality assurance where agents evaluate each other.

Sovereignty Score Standardization. Propose the Sovereignty Score as an interoperable metadata field for knowledge-augmented agents.

Longitudinal Convergence Study. Measure whether the self-healing loop converges (Sovereignty Score increases monotonically) or oscillates, and identify conditions under which convergence is guaranteed.

9. Conclusion

Knowledge-augmented AI agents are a significant advancement in specialized AI capability, but they currently lack the instruments necessary for autonomous evolution. We have presented four complementary instruments — the Friction Journal, Sovereignty Score, Operator Shadow Protocol, and Cryptographic Helix Ledger — that together enable agents to observe their own failures, measure their maturity, evaluate their impact on human operators, and update their knowledge autonomously and securely.

The composition of these instruments into a self-healing loop addresses the fundamental bottleneck in current agent evolution: the human-mediated knowledge update pipeline. By operating at the knowledge level rather than the parameter level, our approach provides interpretable, auditable, and reversible self-improvement.

Our proof-of-concept validates the architectural feasibility of this framework. Production deployment and longitudinal studies remain future work. We believe that equipping AI agents with instruments of self-awareness is not merely an engineering convenience — it is a prerequisite for the responsible scaling of knowledge-augmented agent ecosystems.

The agents that will earn human trust are not the most capable. They are the most self-aware.

References

1. Anthropic. “Model Context Protocol Specification,” version 2025-03-26. https://spec.modelcontextprotocol.io/
2. T. Liu and M. van der Schaar. “Position: Truly Self-Improving Agents Require Intrinsic Metacognitive Learning.” In *Proceedings of ICML*, 2025. arXiv:2506.05109.
3. S. Qiao et al. “Agentic Knowledgeable Self-awareness.” In *Proceedings of ACL*, Vienna, 2025. arXiv:2504.03553.
4. P. Chojecki. “Self-Improving AI Agents through Self-Play.” arXiv:2512.02731, Dec 2025.
5. ICLR 2026 Workshop on Recursive Self-Improvement. Rio de Janeiro, April 2026. https://recursive-workshop.github.io/
6. Institute for AI Policy and Strategy (IAPS). “AI Agent Governance: A Field Guide.” April 2025.
7. Y. Wang et al. “Toward Adaptive Categories: Dimensional Governance for Agentic AI.” arXiv:2505.11579, 2025.
8. A. Guha et al. “With Great Capabilities Come Great Responsibilities: Introducing the Agentic Risk & Capability Framework.” arXiv:2512.22211, Dec 2025.
9. X. Hou, Y. Zhao, S. Wang, and H. Wang. “Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions.” arXiv:2503.23278, March 2025.
10. Coalition for Secure AI (CoSAI). “Model Context Protocol Security.” Workstream 4, January 2026. https://www.coalitionforsecureai.org/

Appendix A: Glossary

TERM	DEFINITION
Helix	A structured knowledge base for a knowledge-augmented AI agent
MCP	Model Context Protocol — an open standard for AI applications to connect to external data sources, tools, and services
Friction	A moment of cognitive failure or suboptimal behavior during agent operation
Commencement	The developmental threshold at which an agent's evolution becomes self-directed
ORBRT	Orb Runtime — the operational context where end users access an agent remotely via MCP

Preprint — Under review — Comments welcome at justin@myorb.ai