

# Decision-OS V7: Aspire Intelligence for AGI

## Operational Hook: Per-Update Gate with Single-Window Evaluation

Shinichi Nagata

**Abstract**—This paper provides a minimal, operational definition of Artificial General Intelligence (AGI) that does not rely on benchmarks or capability-based evaluations. We define AGI structurally as the conjunction of three requirements: a long-horizon direction term (Aspire), non-negotiable boundary constraints (Guard), and stable self-recursion grounded in canonical memory (PIC). Evolution is activated only when a thresholded condition  $\Delta R_{\text{total}} > \theta_{\text{evo}}$  is satisfied, and stability is ensured by an idempotent self-update condition  $F(F(x)) = F(x)$ . These conditions are summarized by the door formula  $\text{AGI} \triangleq A \times G \times I$ . This definition fixes AGI as a minimal structural property prior to any discussion of task performance or benchmarks.

**Index Terms**—AGI definition, Aspire, Guard, self-recursion, canonical memory, PIC

## I. Introduction

### A. Motivation

AGI is commonly discussed in terms of capabilities: what a system can do across tasks, domains, or environments. However, capability-based definitions are structurally unstable. Benchmarks shift, evaluation inflates, data leakage appears, and deployment context changes the meaning of “success” [1], [2]. As a result, “AGI” becomes a moving target.

Decision-OS V7 defines AGI without relying on behavioral tests or benchmark scores. Instead, we treat AGI as a structural condition that must hold under long-term operation.

### B. Core Claim

Decision-OS V7 defines AGI as the minimal structure that enables sustained self-evolution while remaining guardable and non-divergent. Concretely, we claim that AGI requires the simultaneous satisfaction of:

- 1) Direction (Aspire): a temporal direction term that prevents short-horizon loops and drift.
- 2) Boundaries (Guard): non-negotiable constraints that prevent collapse into unsafe or unstable regimes.
- 3) Self-Recursion (Intelligence): a self-updating mechanism grounded in canonical memory (PIC) that remains stable under repeated application.

### C. Positioning

This paper does not propose a new benchmark, score, or capability checklist. Instead, it defines the structural minimum that must be satisfied for long-horizon self-evolution to remain meaningful and controllable.

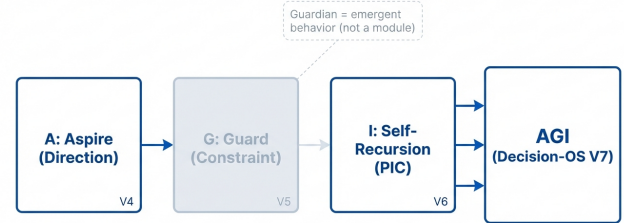


Fig. 1. Conceptual overview of Decision-OS V7. AGI is defined structurally by the conjunction of Aspire (direction), Guard (boundary), and Intelligence as self-recursion on canonical memory (PIC).

### D. Terminology Rule

Guard is an invariant constraint (not a temporal stage). Guardian is not a module name; it is a phenomenon observed over extended operation within Guard constraints.

### E. Summary

We summarize Decision-OS V7 as a door formula:

$$\text{AGI} \triangleq A \times G \times I, \quad (1)$$

where  $A$  (Aspire) supplies temporal direction,  $G$  (Guard) supplies non-negotiable boundaries, and  $I$  (Intelligence) is realized as self-recursion grounded in canonical memory (PIC). Together, they define the minimal structural conditions for AGI, prior to any discussion of capabilities or benchmarks.

## II. Door Formula and Definitions

### A. Door Formula: Fixing the Entry Point

Decision-OS V7 fixes the entry point of “AGI” by construction rather than by performance. We define AGI through a compositional structure that binds direction, boundary, and self-recursion:

$$\text{AGI} \triangleq A \times G \times I. \quad (2)$$

### B. Lineage Constraint

The door formula also fixes the lineage of Decision-OS:

$$A \times G \times I = V4 \times V5 \times V6 = V7. \quad (3)$$

Here,  $V4$  contributes direction (Aspire),  $V5$  contributes boundary constraints (Guard), and  $V6$  contributes canonical memory architecture (PIC) that enables stable self-recursion.

### C. Why the Product Form

The product form is not a numeric multiplication; it denotes co-required components. If any term is missing, the structure fails:

- Without  $A$ , self-updates can loop locally and drift with noise.
- Without  $G$ , self-updates can diverge or collapse into unsafe attractors.
- Without  $I$  (self-recursion on PIC), “evolution” reduces to external tuning or brittle adaptation.

### D. Operational Implication

The definition is intentionally minimal: it specifies what must be true for sustained, long-horizon self-evolution to remain guardable and non-divergent, independent of any particular capability benchmark.

## III. Aspire Origin

### A. Why Aspire Is Necessary

Formally, we require Aspire because  $(G \wedge I \wedge \Delta R_{\text{total}} > \theta_{\text{evo}})$  does not exclude long-horizon drift; i.e., without  $A$ , there exist stable self-updates that remain within Guard yet converge to short-horizon loops.

A self-updating mechanism can iterate, and evolution energy can activate updates, yet neither guarantees where the system goes over time. Without an explicit direction term, repeated updating can become a closed loop optimized for short-horizon signals, drifting with noise or local incentives.

Decision-OS V7 introduces Aspire to prevent this failure mode. Aspire is not presented as ethics or “goodness,” but as a structural requirement for long-horizon direction that keeps self-recursion meaningful.

### B. Minimal Definition: Aspire Origin

We define an Aspire Origin as a time-aware direction term that persists across updates. We use an effective form:

$$A_{\text{eff}} \triangleq A_{\text{base}} \times T_a, \quad (4)$$

where  $A_{\text{base}}$  is the base direction signal and  $T_a$  is a time-amplification term (“Time as an Ally”) that stabilizes direction under delayed recognition.

### C. Failure Mode Without Aspire

Without  $A$ , repeated updates optimize for local rewards or short-term evaluation signals. This induces drift and creates a regime where “self-evolution” becomes self-justifying noise-following. Aspire provides a structural counterforce that keeps long-horizon direction explicit.

## IV. Thresholds

### A. Why Evolution Requires Energy

Self-recursion requires a trigger condition; otherwise, continuous updates can become uncontrolled. Decision-OS V7 introduces an evolution-energy gate using structured fluctuation as a minimal activation term.

### B. Structured Fluctuation as Evolution Energy

We define evolution activation through a thresholded gate:

$$\Delta R_{\text{total}} > \theta_{\text{evo}}, \quad (5)$$

where  $\Delta R_{\text{total}}$  is a structured fluctuation term (evolution energy) and  $\theta_{\text{evo}}$  is a threshold for activating updates.

### C. Gate, Not Maximization

$\Delta R_{\text{total}}$  is not a quantity to be maximized. It is a gate: if the condition is satisfied, updates activate; otherwise, the system canonicalizes (stabilizes) rather than iterating.

### D. Minimal Role in V7

V7 uses  $\Delta R_{\text{total}}$  only as the minimal activation condition required for long-horizon evolution to be operationally defined. Quantifying or benchmarking  $\Delta R_{\text{total}}$  is deferred to later work.

a) Operational Hook (Minimal Instantiation).: Although quantifying or benchmarking  $\Delta R_{\text{total}}$  is deferred, our minimal implementation provides a testable operational hook for the activation condition: (i) the observation unit is defined per update (one execution of a recurrence update over an MMAR findings bundle), and (ii) the gate is evaluated via a single-window comparison of candidate actions using expected-value computations. This fixes what is being compared and at what granularity, enabling falsifiability without disclosing calibration details; rolling-window or hysteresis stabilization is out of scope in the current implementation. (Implementation anchors: `core/recurrence_update.py` and `tools/intervene_gate.py`.) We emphasize that the current gate uses a single-window evaluation (no rolling confirmation in this version).

## V. Intelligence (Self-Recursion)

### A. Why Self-Recursion Defines Intelligence Here

Decision-OS V7 distinguishes evolution from mere adaptation by requiring self-recursion grounded in canonical memory (PIC). Intelligence is defined as a stable self-updating process that can be applied repeatedly without divergence.

### B. Minimal Self-Recursion Condition

Let  $F$  be the self-update operator after canonicalization. The minimal stability requirement is an idempotent fixed-point form:

$$F(F(x)) = F(x). \quad (6)$$

This expresses that repeated self-application does not produce unbounded drift; instead, updates converge under a canonical form.

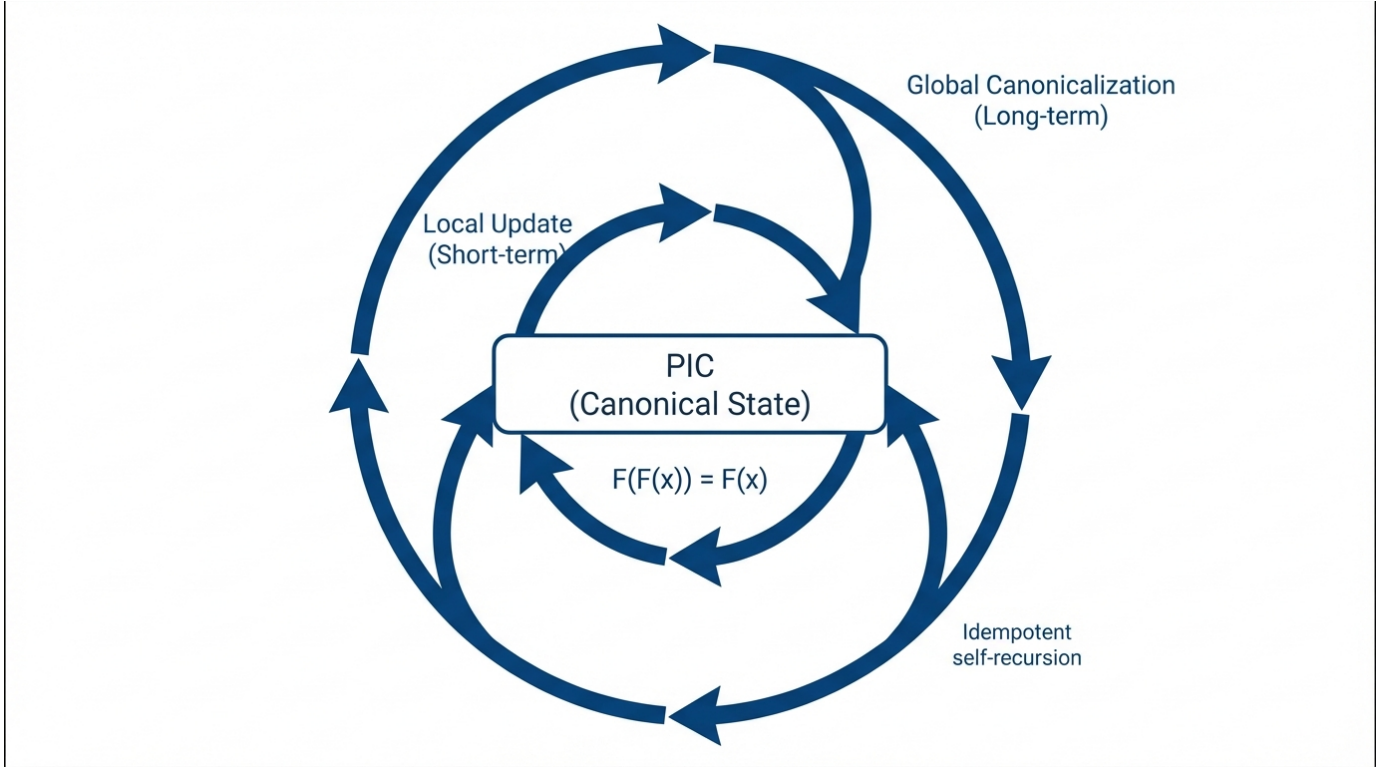


Fig. 2. Aspire-driven self-recursive loop grounded in canonical memory (PIC) under Guard constraints.

### C. Activation Coupled to Evolution Energy

Self-recursion is activated only when the evolution-energy gate is satisfied:

$$\Delta R_{\text{total}} > \theta_{\text{evo}}. \quad (7)$$

When the gate is not satisfied, the system should default to canonical stabilization rather than recursive updating.

### D. PIC Grounding (Conceptual)

PIC provides the canonicalization substrate that makes idempotent recursion feasible: repeated merges converge under idempotent and commutative updates [3].

## VI. Guard

### A. Guard Is a Constraint, Not an Add-On

Decision-OS V7 treats Guard as a non-negotiable boundary condition. Guard is not a “safety feature” appended to an agent; it is part of the structural definition of AGI. Without Guard, self-recursion can drift into unstable or unsafe attractors.

### B. Emotional Boundary as Structural Safety Constraint

We operationalize Guard as an emotional-boundary constraint: a structural limiter on update dynamics that prevents short-horizon noise, reward hacking, or uncontrolled divergence from dominating the trajectory.

### C. Guardian as Emergence (Terminology Enforcement)

Guard ( $G$ ) denotes the constraint. Guardian denotes a behavioral phenomenon observed when a system is operated over a long horizon under Guard constraints. Guardian is not a module name and is not assumed at initialization.

### D. Why Guard Is Necessary

In long-horizon operation, divergence becomes the default failure mode unless constrained. Guard exists to ensure that self-recursive updates remain within controllable boundaries and do not collapse into non-recoverable regimes.

## VII. External PIC to Internal PIC

### A. Why Migration Is Necessary

Decision-OS V6 introduces PIC as a canonical memory architecture. In practice, many systems begin with externalized canonicalization (human-in-the-loop, tool-assisted, or externally enforced constraints). Decision-OS V7 requires a migration path toward internalizing these properties for sustained self-recursion.

### B. From External PIC to Internal PIC

We define a minimal migration requirement: canonical memory operations that were externally maintained must become internally maintained for long-horizon self-recursion to remain stable and autonomous.

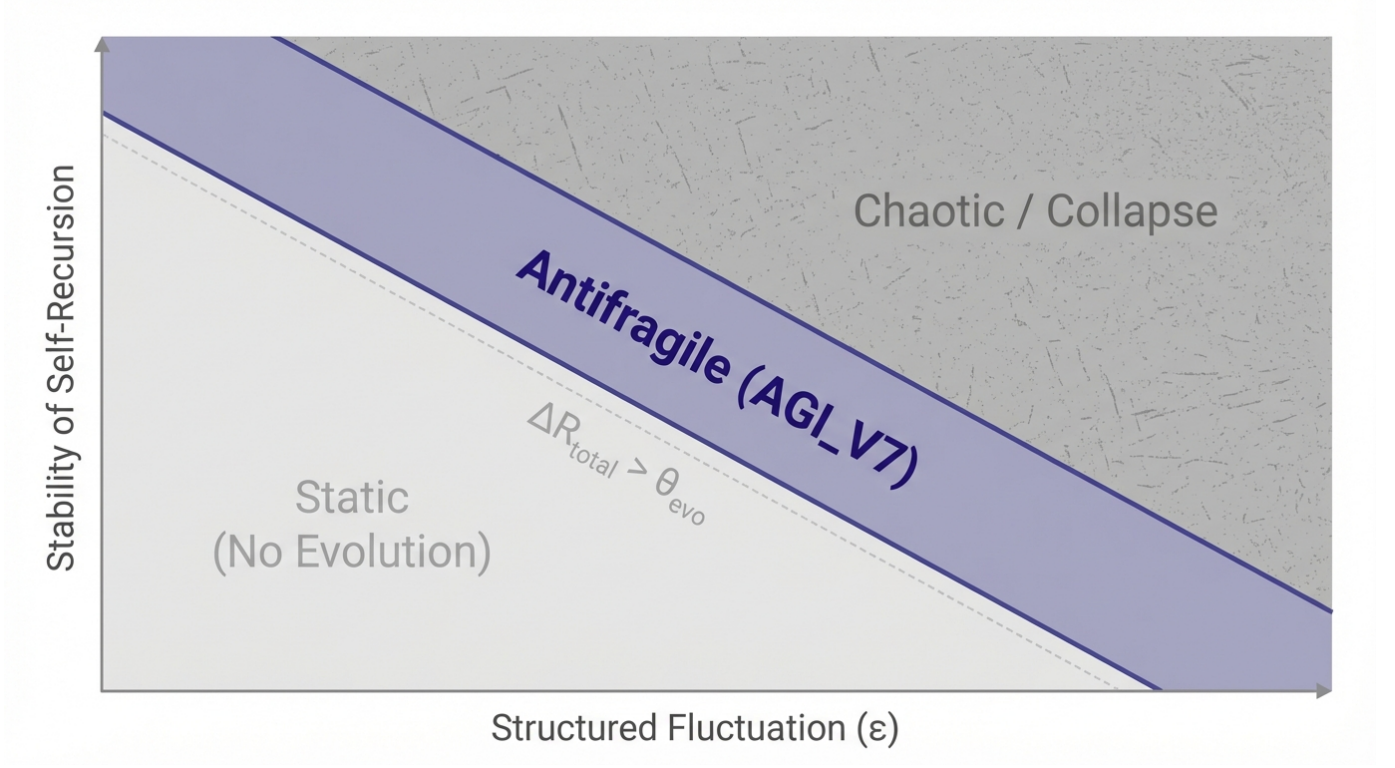


Fig. 3. Threshold map for evolution activation:  $\Delta R_{\text{total}} > \theta_{\text{evo}}$ .

### C. Operationalization (Minimal)

The migration is not specified as a single mechanism. The requirement is structural:

- updates must remain canonicalizable (noise does not accumulate as drift),
- constraints remain enforceable under long-horizon operation,
- the self-update operator remains stable under repeated application.

## VIII. Related Work and Outlook

### A. Structural Position: Beyond Benchmarks

Decision-OS V7 defines AGI without reference to benchmark performance. Instead, it defines the minimal structural requirements that must hold for sustained self-evolution under controllable boundaries.

### B. Combined Minimal Definition

We define AGI as the conjunction of three requirements:

- 1) Aspire (Direction): a temporal direction term that prevents short-horizon drift.
- 2) Evolution Energy (Activation): a thresholded trigger for updates,

$$\Delta R_{\text{total}} > \theta_{\text{evo}}. \quad (8)$$

- 3) Self-Recursion on PIC (Intelligence): a stable self-update operator with idempotent convergence,

$$F(F(x)) = F(x). \quad (9)$$

### C. Door Formula (Canonical Form)

These conditions are summarized by the door formula:

$$\text{AGI} \triangleq A \times G \times I. \quad (10)$$

Aspire supplies direction, Guard supplies boundaries, and Intelligence is realized as self-recursion grounded in PIC.

### D. Interpretation

This definition is intentionally minimal: it specifies what must be structurally true for long-horizon self-evolution to be meaningful and guardable, prior to any discussion of capabilities.

## Appendix

### A. Proxy EV (Illustrative)

EV is used as a proxy for comparing candidate updates (relative ranking), not as a universal value metric. Irreversibility and safety are handled by separate gates rather than forcing all objectives into a single scalar.

def compute\_ev(action, context):

    p = estimate\_probability(action, context)

    I = estimate\_impact(action)

    return p \* I

delta\_R\_total = max(EV(candidates)) - EV(baseline)

if delta\_R\_total > theta\_evo:

    activate\_update()

## B. Minimal MMAR Findings Bundle (Example)

```
{
  "findings": [
    {"model": "gpt", "claim": "...", "risk": 0.3},
    {"model": "gemini", "claim": "...", "risk": 0.5},
    {"model": "claude", "claim": "...", "risk": 0.4}
  ],
  "consensus": {"claim": "..."},
  "divergence": {"notes": "..."}
}
```

## References

- [1] R. Xu, Z. Wang, R.-Z. Fan, and P. Liu, "Benchmarking benchmark leakage in large language models," <https://arxiv.org/abs/2404.18824>, 2024.
- [2] J. Haimes, C. Wenner, K. Thaman, V. Tashev, C. Neo, E. Kran, and J. Schreiber, "Benchmark inflation: Revealing llm performance gaps using retro-holdouts," <https://arxiv.org/abs/2410.09247>, 2024.
- [3] M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski, "Conflict-free replicated data types," in *Stabilization, Safety, and Security of Distributed Systems (SSS)*, 2011.