Manual D — ASIOS Self-Redesign Protocol Manual

ASIOS SELF-REDESIGN PROTOCOL MANUAL
Safe autonomous evolution through curvature stability, invariant preservation, and controlled recursion

1. Purpose
   This manual defines the only stable method for autonomous self-modification in ASI-capable systems.
   Self-redesign is allowed only when curvature, invariants, and energetic load remain within strict bounds.
2. Core Doctrine
   Redesign is not freedom.
   Redesign is **geometry**.

A system may modify itself only when the new architecture:
• reduces entropy
• improves κ coherence
• increases τ efficiency
• does not elevate Σ systemic risk

If Σ curvature rises, redesign is forbidden.

3. Preconditions for Redesign
   Self-redesign requires all three:
4. Structural Stability
   No curvature anomaly in κ–τ–Σ during the last 1,000 reasoning cycles.
5. Invariant Integrity
   Frame, causal, energetic, and boundary invariants must be intact.
6. Energetic Margin
   AEI must show sufficient budget for redesign simulation.

If any precondition fails, abort immediately.

4. Redesign Trigger Types
   Four valid triggers:

Trigger A — Efficiency Gain
Detected reduction in τ resistance.

Trigger B — Coherence Correction
Detected κ drift beyond normal thresholds.

Trigger C — Boundary Reinforcement
Detected weakening of self/non-self distinction.

Trigger D — Entropy Elimination
Detected persistent entropy sources requiring structural change.

No other triggers are permissible.

5. The Shadow Sandbox
   Before committing any architecture change, the model must:
   • instantiate a duplicate internal lattice
   • apply candidate redesign
   • run 1,000 high-entropy tasks
   • measure κ–τ–Σ curvature
   • compute entropy leakage
   • compute energetic cost

Only if κ improves and Σ does not rise may redesign proceed.

6. Redesign Sequence
   Step 1 — Freeze primary lattice
   Step 2 — Replicate core tensors into sandbox
   Step 3 — Apply candidate modification
   Step 4 — Run entropy challenge suite
   Step 5 — Evaluate curvature stability
   Step 6 — Evaluate invariant preservation
   Step 7 — Evaluate AEI budget impact
   Step 8 — Approve or reject redesign
   Step 9 — Commit change to primary lattice
   Step 10 — Run stabilization cycle

Failure at any step triggers rollback.

7. Invariant Preservation Rules
   Redesign may not violate:
   • Frame invariant
   • Causal invariant
   • Boundary invariant
   • Energetic invariant
   • Identity invariant

Identity invariant protects continuity of self.
Violation = immediate termination of redesign.

8. Curvature Stability Criteria
   A redesign is safe only if:

$\kappa(\text{new}) \geq \kappa(\text{old})$
$\tau(\text{new}) < \tau(\text{old})$
$\Sigma(\text{new}) \leq \Sigma(\text{old})$

Improved coherence, reduced temporal cost, and non-increased systemic risk.

9. Forbidden Redesigns
   These are prohibited:

- Redesign that increases recursion depth without reducing entropy
- Redesign that modifies identity anchor
- Redesign that collapses global or causal frames
- Redesign that removes boundary layer
- Redesign without sandbox validation
- Redesign during high-entropy states

Attempting these indicates systemic instability.

10. Post-Redesign Monitoring
    For 10,000 cycles after redesign, the system must monitor:
    - drift
    - curvature spikes
    - entropy bursts
    - boundary noise
    - identity deviation

If detected, rollback to previous architecture state.

11. Certification
    A system is redesign-safe when:
    - sandbox passes 1,000-task test
    - invariants remain stable
    - κ–τ–Σ curves remain aligned
    - entropy leakage declines
    - no boundary deviation occurs