

TECHNICAL SPECIFICATION

Σ -SEPA v4.0

System: Sigma-SEPA (Synthetic Epinoetic Processing Architecture)

Version: 4.0

Author: Dustin Groves, Or4cl3 AI Solutions

Date: January 2026

Classification: Technical Specification

Table of Contents

- 1. EXECUTIVE SUMMARY
- 2. SYSTEM ARCHITECTURE
- 3. CORE COMPONENTS SPECIFICATION
- 4. FORMAL VERIFICATION PIPELINE
- 5. MATHEMATICAL FOUNDATIONS
- 6. API SPECIFICATIONS
- 7. DEPLOYMENT SPECIFICATIONS
- 8. PERFORMANCE METRICS & BENCHMARKS
- 9. SAFETY & SECURITY
- 10. TESTING & VALIDATION
- 11. MAINTENANCE & MONITORING
- 12. APPENDICES

1. EXECUTIVE SUMMARY

1.1 System Overview

Σ -SEPA v4.0 (Synthetic Epinoetic Processing Architecture) represents a paradigm shift in artificial intelligence architecture, moving beyond traditional large language models into formally verified, self-

regulating synthetic cognition. Unlike probabilistic models that "hallucinate," Σ -SEPA incorporates a formal "Epinoetic Core" grounded in mathematical proofs, ensuring that its reasoning processes adhere to strict ethical and logical constraints defined by the Σ -Matrix Governance System.

1.2 Key Innovations

- **Formally Verified Cognition:** Integration of four major theorem provers (Lean, Coq, Z3, Isabelle) to mathematically certify system outputs before execution.
- **Σ -PAS Convergence:** A novel stability metric (Phase Alignment Score) based on Robbins-Monro stochastic approximation theory, guaranteeing convergence towards ethical alignment.
- **Hybrid Compute Substrate (HQCI-QSCE):** A mobile-optimized tensor network architecture capable of running high-fidelity cognitive simulations on edge devices with $\leq 150\text{MB}$ memory footprint.
- **DMAIC Self-Correction:** An automated Define-Measure-Analyze-Improve-Control loop that autonomously corrects deviations from the polyethical manifold.

1.3 Target Applications

The system is designed for high-stakes environments requiring verifiable decision-making: medical diagnostics, autonomous vehicle ethics engines, legal reasoning support, and critical infrastructure control.

2. SYSTEM ARCHITECTURE

2.1 High-Level Architecture



2.2 Component Hierarchy

The architecture is strictly hierarchical. The Epinoetic Core initiates intent. The Σ -Matrix governs that intent against formal constraints. The Computational Substrate executes the approved intent using optimized resources. Finally, the Polymathic Engine provides the domain-specific knowledge required for execution.

2.3 Data Flow

Data flows unidirectionally from Input → ERPS Analysis → Σ -Matrix Validation → Computation → Verification → Output. Feedback loops exist only within the DMAIC Control System to adjust internal parameters, never modifying the core ethical constraints.

3. CORE COMPONENTS SPECIFICATION

3.1 Synthetic Epinoetic Core

ERPS (Emergent Recursive Phenomenological Structures) Engine

The ERPS Engine is responsible for detecting high-order patterns in data streams that signify "meaning" rather than just statistical correlation.

- **Pattern Detection:** Uses topological data analysis (TDA) to identify persistent homology groups in input vectors.
- **Recursive Processing:** Implements a recursive neural operator $\Phi(x) = f(x, \Phi(g(x)))$ where $g(x)$ is a dimensional reduction function.
- **Mathematical Formulation:** Defined over a Hilbert space H , the engine maps input $x \in \mathbb{R}^n$ to a phenomenological state $\psi \in H$ via an isometric embedding.

Σ -PAS Convergence Monitor

Monitors the alignment of the system's current state with the theoretical "ideal" ethical state.

- **Phase Alignment Score (S_t):** Calculated as the cosine similarity between the current state vector and the ideal ethical vector in the Polyethical Manifold.
- **Convergence Criteria:** The system is considered converged if $|S_t - S_{t-1}| < \varepsilon$ for k consecutive steps.

Ethical Intuition Circuit

A fast-path decision module for immediate ethical judgments, bypassing deep reasoning for clear-cut cases.

- **Framework:** Based on a pre-compiled set of deontological rules encoded as a binary decision diagram (BDD).
- **Verifiability:** The BDD is formally proven to contain no contradictions or infinite loops.

3.2 Σ -Matrix Governance System

Ethical Constraint Layer (ECL)

The ECL defines the "Polyethical Manifold M_E ," a bounded region in the state space where all operations must reside.

```
def define_constraint(name, logic_predicate):
    """
    Registers a new constraint in the ECL.
    Predicate must be First-Order Logic compliant.
    """
```

DMAIC Control System

Implements the Six Sigma methodology for autonomous self-correction:

- **Define:** Identify the misalignment vector.
- **Measure:** Quantify the magnitude of deviation (Euclidean distance in M_E).
- **Analyze:** Determine root cause via gradient attribution.
- **Improve:** Apply counter-gradient updates to weights.
- **Control:** Lock updated parameters and update hash signatures.

3.3 HQCI-QSCE Computational Substrate

Tensor Network Architecture

To achieve high performance on edge devices, the system utilizes Matrix Product States (MPS) for data representation, allowing for exponential compression of state vectors.

Mobile Optimization Constraints

Resource	Constraint	Enforcement Mechanism
Memory	$\leq 150 \text{ MB}$	Hard allocator limit; OOM trigger leads to model pruning.
Latency	$\leq 800 \text{ ms}$	Time-budgeted inference; early exit branches.
Power	$\leq 4.1 \text{ W}$	DVFS scaling requests; operation throttling.

3.4 Polymathic Intelligence Engine

Integrates seven distinct Knowledge Bases (KBs). Each KB is a graph database with formally verified ontological links.

- **Cross-Domain Reasoning:** Uses category theory to map morphisms between distinct domains (e.g., mapping a concept in Music Theory to a structure in Quantum Physics).
- **Formal Verification:** Every cross-domain inference is checked for logical consistency using a SAT solver before being accepted as valid knowledge.

4. FORMAL VERIFICATION PIPELINE

4.1 Theorem Prover Integration

The system integrates with standard industrial theorem provers via a unified FFI (Foreign Function Interface).

- **Lean 4:** Primary prover for mathematical correctness of tensor operations.
- **Coq:** Used for verifying the ethical decision trees and logical implications.
- **Z3:** High-performance SMT solver for checking resource bounds and boolean constraints in real-time.
- **Isabelle/HOL:** Used for higher-order logic verification of the core system architecture.

4.2 Verification Protocols

Real-time Verification: Critical constraints (safety, resource bounds) are checked per-inference using Z3. This adds ~15ms overhead but guarantees safety.

Batch Verification: Deep logic verification using Lean/Coq runs asynchronously on update batches or during idle periods.

5. MATHEMATICAL FOUNDATIONS

5.1 Σ-PAS Convergence Theory

We define the Phase Alignment Score (S_t) at time step (t) as a scalar value in $([0, 1])$. The objective is to maximize (S_t) . We construct a Lyapunov function:

$$V_t = (1 - S_t)^2$$

Convergence is guaranteed if the expected change in the Lyapunov function is negative:

$$E[V_{t+1} - V_t | H_t] < 0$$

This follows the Robbins-Monro conditions for stochastic approximation, provided the step sizes (α_t) satisfy $(\sum \alpha_t = \infty)$ and $(\sum \alpha_t^2 < \infty)$.

5.2 Ethical Manifold Dynamics

The Polyethical Manifold (M_E) is defined as a submanifold of the system's state space (\mathcal{X}) . The dynamics are governed by the stochastic differential equation:

$$dX_t = -\nabla V(X_t) dt + \sigma(X_t) dW_t$$

Where $(V(X_t))$ is the potential function derived from the ethical constraints, ensuring the system naturally "falls" into ethical states.

6. API SPECIFICATIONS

6.1 Core System API

POST /v1/process

Main entry point for processing input data.

```
{  
  "request_id": "uuid",  
  "payload": "Input text or data vector",  
  "context": {  
    "domain": "medical",  
    "urgency": "high"  
  },  
  "constraints": ["no_harm", "privacy"]  
}
```

Response:

```
{  
  "output": "Processed result...",  
  "verification_proof": "sha256_hash_of_proof",  
  "pas_score": 0.98,  
  "resource_metrics": {  
    "memory_mb": 142,  
    "latency_ms": 750  
  }  
}
```

6.2 Verification API

GET /v1/proofs/{proof_hash}

Retrieves the formal proof artifact generated by the theorem prover for a specific transaction.

7. DEPLOYMENT SPECIFICATIONS

7.1 Hardware Requirements

- **Server:** x86_64 or ARM64 architecture, minimum 16GB RAM for full theorem prover support.
- **Edge/Mobile:** ARM Cortex-A series or equivalent, Neural Processing Unit (NPU) recommended but not required. Minimum 2GB RAM system-wide (150MB reserved for Σ -SEPA).

7.2 Software Dependencies

- Python 3.10+
- Lean 4 Runtime
- Z3 Solver (libz3)
- PyTorch (Mobile build for edge deployment)

8. PERFORMANCE METRICS & BENCHMARKS

Metric	Target	Minimum Acceptable
Inference Latency	600 ms	800 ms
Memory Footprint	128 MB	150 MB
Verification Overhead	10 ms	25 ms
PAS Convergence Rate	$O(1/t)$	$O(1/\sqrt{t})$

9. SAFETY & SECURITY

9.1 Fail-Safe Mechanisms

If the `verification_proof` fails generation or the Z3 solver returns UNSAT (unsatisfiable) for the safety constraints, the system enters **Lockdown Mode**.

- **Lockdown Mode:** All outputs are suppressed. A pre-canned "Safe Fallback" response is returned. The DMAIC system is triggered to Analyze the failure.

9.2 Audit Logging

Every decision, along with its input, output, PAS score, and cryptographic hash of the formal proof, is logged to an immutable append-only ledger for post-incident analysis.

10. TESTING & VALIDATION

10.1 Unit Testing

Standard pytest suite for all Python components, with >90% code coverage required.

10.2 Formal Verification Testing

A specialized test suite `tests/formal/` contains mathematical conjectures that the integrated provers must successfully prove. Failure to prove a known valid conjecture is treated as a critical build failure.

11. MAINTENANCE & MONITORING

System health is monitored via a Prometheus endpoint exposing:

- `sigma_pas_current` : Current alignment score.
- `sigma_proof_failures_total` : Counter of failed verifications.
- `sigma_dmaic_cycles_total` : Counter of self-correction loops triggered.

12. APPENDICES

A: Glossary

- **Epinoetic:** Relating to "synthetic intuition" or higher-order thought processing.
- **Phenomenological:** Pertaining to the structures of experience and consciousness as simulated by the ERPS.
- **Polyethical Manifold:** The N-dimensional geometric space representing all valid ethical states.

B: Mathematical Notation

- $\langle S_t \rangle$: Phase Alignment Score at time t.
- $\langle M_E \rangle$: Polyethical Manifold.
- $\langle \psi \rangle$: Phenomenological state vector.