Governed Autonomous SAP S/4HANA & Compound Enterprise AI Systems: 4-Layer MCP, SAP-RPT-1, and Pre-Execution Policy Gates

Technical Architecture Whitepaper · Production Lessons Across 750,000+ Enterprise Workflows

Author: Nagarjuna Reddy Kamireddy (Nagarjuna Kamireddy) ? Lead AI Architect
LinkedIn: https://www.linkedin.com/in/nagarjuna-kamireddy/ | GitHub: https://github.com/nagarjunak-pixel
Canonical Hub: https://nagarjunak-pixel.github.io/ | Published: November 18, 2026

---

1. Abstract & Enterprise Problem Statement

Enterprise AI deployments across SAP S/4HANA, regulated banking, and global manufacturing frequently stall between proof-of-concept and production due to three structural failure modes: (1) semantic blindness over custom Z-tables and account-determination rules such as OBYC GBB/VBR versus GBB/VAX, (2) unbounded frontier-model token burn on routine lookups, and (3) probabilistic write-back risk against deterministic ledgers such as ACDOCA.

Drawing on 22+ years of enterprise SAP HANA/BW/BPC architecture (GSK UK, Deloitte/PETRONAS, DHL, Symantec, Lenovo) and leading 23+ production AI initiatives at John Deere (750,000+ automated requests and 42,000+ hours saved in FY26), this paper defines a governed 4-Layer Compound AI & Autonomous ERP reference architecture optimized for Cost-per-Completed-Workflow.

2. The 4-Layer SAP MCP Architecture & SAP-RPT-1 Integration

Layer 1 ? CAP Domain MCP (@cap-js/mcp-server): Exposes Clean Core SAP BTP CDS domain models and typed service contracts.

Layer 2 ? Read-Only ABAP ADT & Z-Code Reconnaissance: Inspects custom Z-tables, BAPIs, CDS views, and OBYC valuation/account determination rules in read-only mode prior to planning.

Layer 3 ? Stateless Streamable HTTP MCP Gateway with SAP IAS OAuth 2.1: Propagates human and ephemeral Non-Human Identity (NHI) permissions end-to-end without static admin credentials.

Layer 4 ? Relational + Graph Context Pushdown (SAP-RPT-1 + Hybrid BM25/Vector/Neo4j GraphRAG) + Deterministic Policy Sandboxes (aegis-runtime, prism-gate, tracegit): Separates Planner, Executor, and Independent Verifier swarms so no model ever grades its own high-stakes ledger mutations.

Reference Implementations:
- https://github.com/nagarjunak-pixel/aegis-runtime (Zero-Trust Policy DSL, Saga Rollbacks, Merkle/Ed25519 Audit Log)
- https://github.com/nagarjunak-pixel/prism-gate (TCP-Vegas Concurrency & FinOps Cascade LLM Gateway)
- https://github.com/nagarjunak-pixel/chronos-mesh (Simplex + Branch-and-Bound MILP & CVaR95 Supply-Chain Engine)