

RESEARCH PAPER

THESIS • REFERENCE ARCHITECTURE • RESEARCH AGENDA

AE-OS

The Agent-Native Enterprise Operating System

An open framework for next-generation enterprise resource planning: decentralized, identity-first and verifiable by design.

A thesis on enterprise information systems designed around autonomous software agents, with a reference architecture and a research agenda to test it.

What should enterprise information systems become when software agents perform most digital operational work under human authority?

BY DARREN LENTZ CA(SA), CISA

INITIAL ARCHITECTURAL HYPOTHESIS

L1	SEMANTIC DATA PLANE Actor Grid / Ray Clusters
L2	ZERO-TRUST IDENTITY PLANE SPIFFE/SPIRE • eBPF kTLS
L3	COMPLIANCE LAYER Sandboxed Wasm Policy Engine
L4	CAUSAL STATE LEDGER Append-Only Merkle-DAG Log

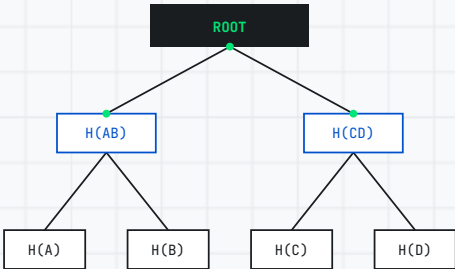


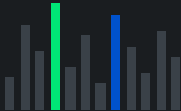
FIG. 8 // CAUSAL MERKLE-DAG PRIMITIVE

VERSION BLOCK		AUTHORSHIP		WRITTEN FOR	
DOC-ID	AEOS-WP-001	AUTHOR	Darren Lentz	FIN	Finance & audit leaders
VERSION	v1.0 • RESEARCH	TYPE	Research thesis	ARCH	Enterprise architects
ISSUED	2026-10-13	METHOD	Design science	SEC	Security & crypto
PAGES	55	STAGE	Proposed design	RES	IS researchers

■ CRYPTOGRAPHIC HASH FOOTPRINT // SHA-256

40fd5737603e9a64b4d805726ed82bde5d8f0bd874088f6f82b8262727812b48

SHA-256 over the source fragments from which this PDF was built. A digest printed inside a document cannot authenticate that copy; verify the PDF against a separately published digest and detached signature.



Notice, Disclaimers & Build Reproducibility

L.1 Nature of this document

This is an independent research paper. It states a thesis about agent-native enterprise information systems, with ERP as the first application, and sets out a reference architecture and a research agenda to test it. The paper presents AE-OS as a **proposed design**. The paper is written in a personal capacity: the views are the author's own and do not represent his employer or its clients. Unless explicitly attributed to a cited third-party source, every description of AE-OS components, interfaces, performance or outcomes is a **design intention**, not a description of shipped software, a measured benchmark or a customer result.

The document is provided for information and technical discussion only. It does not constitute an offer to sell, or a solicitation of an offer to buy, any security, token, license or service in any jurisdiction, and it is not investment, legal, tax, accounting or regulatory advice. Readers should obtain independent professional advice before making any decision.

L.2 Forward-looking statements

Section 5 and Appendix D contain forward-looking statements, simulations and financial projections, including the South China Sea typhoon scenario and the illustrative seven-year economic model. These are **modeled, illustrative** outputs derived from stated assumptions and carry an amber **MODELED** label. Actual results may differ materially. No representation is made that any projection will be achieved.

L.3 Third-party data & trademarks

Market statistics, incident reports and case-study figures are drawn from public sources listed in the Technical Footnotes, cited in-line as ^[n]. They are reproduced in good faith; the author has not independently audited them. SAP, S/4HANA, ABAP and HANA are trademarks of SAP SE. Oracle, Microsoft, Ray, SPIFFE, SPIRE, Kubernetes, Cilium, eBPF, Open Policy Agent, Cedar, Sigstore, WebAssembly, Wasmtime and Model Context Protocol are trademarks or project names of their respective owners. Their use is descriptive and implies no endorsement or affiliation.

L.4 No warranty

This document is provided "as is", without warranty of any kind, express or implied, including fitness for a particular purpose and non-infringement. The reference code in Appendix A is illustrative: it demonstrates data structures and verification algorithms and has not been security-audited for production use.

L.5 Build reproducibility statement

The author states that:

- **Source integrity.** Every page is generated from version-controlled HTML fragments. The SHA-256 digest of those fragments, computed before template substitution, is printed on the cover and truncated in every page footer. The digest identifies the source build; it authenticates this document only when compared with a value published out of band with a detached signature.
- **Citation discipline.** Every external quantitative claim carries a numbered reference to a publicly accessible source. Claims that could not be verified against a primary or reputable secondary source were removed rather than softened.
- **Separation of fact and model.** Simulated, estimated and projected figures appear only inside **MODELED** blocks with their assumptions stated.
- **Reproducible figures.** The cryptographic values in FIG. 3.1 and the series in FIG. D.1 are produced by scripts in the source bundle, and Appendix A code is accompanied by its test results.
- **Credential hygiene.** The document and its source bundle contain no credentials, private keys or customer data.

BUILD DIGEST

SHA-256 40fd5737603e9a64b4d805726ed82bde5d8f0bd874088f6f82b8262727812b48

Verify: `node src/build.js` prints the digest; compare it with the value above and with the per-page footer prefix.

DOCUMENT CONTROL

DOCUMENT ID	AEOS-WP-001
VERSION / STATUS	v1.0 · Research paper
ISSUE DATE	2026-10-13
CLASSIFICATION	PUBLIC: may be redistributed unmodified
SIGNATURE	Not embedded; verify against a separately published detached signature.
AUTHOR	Darren Lentz CA(SA), CISA

REVISION HISTORY

VER.	DATE	CHANGE
1.0	2026-10-13	Initial issue

00 CONTENTS & READING GUIDE

Contents

00	Legal Notice & Build Statement	02
ES	Executive Summary	04
01	Background & the Agent-Native Question	06
	What agent-native means; initial hypothesis	07
	Research question, thesis and method	08
02	Proposed Architecture	09
	Zero-Trust Identity Plane	11
	The Enterprise Stack	13
	Accounting Kernel & Calculation Engines	14
	Processes, Domain Packs & Localization	15
	Validation and Assurance	18
	Identity of People, Organizations and Agents	20
03	Causal Accountability Core	22
	Content-Addressable Ledger	24
	Wasm Policy Guardrails	26
04	Developer Ecosystem & Platform Primitives	27
05	Human Oversight & the Typhoon Scenario	31
06	Evaluation & Open Problems	35
07	Conclusion	37
A	Appendix A: Reference Code	39
B	Appendix B: Proof Sketches	42
C	Appendix C: Research & Implementation Scope	43
D	Appendix D: Illustrative Economic Model	45
Q	Open Research Questions	49
08	References	50

READING KEY

MODELED marks a projection or simulation. Its assumptions are stated next to it.

[n] marks a cited source, listed in §08.

"AE-OS specifies" describes the proposed design, not a deployed system or a customer result.

PLAIN-LANGUAGE GLOSSARY

ERP	Enterprise resource planning: the core system holding a company's financial and operational records.
SOFTWARE AGENT	Autonomous software given a goal that carries out the steps itself, within set limits.
IDENTITY	A verifiable digital credential that proves which person or program is acting.
POLICY	A company rule written so a computer can check it, such as a spending limit.

HOW TO READ THIS PAPER

The paper serves four kinds of reader. Each route below lists the sections that answer that reader's main questions; the rest can be read for depth.

RESEARCHERS INFORMATION SYSTEMS · AUDIT · SECURITY	The research question, thesis and method, the contributions, how they would be evaluated and the limitations. §1.4–1.6 → §6 → §7 → Open questions
FINANCE & EXECUTIVE LEADERSHIP CFO · CEO · BOARD	What the proposal would change, what evidence auditors would receive and what the economics depend on. Executive Summary → §1.4 → §2.9–2.10 → §2.16–2.17 → App. D
TECHNOLOGY LEADERSHIP CTO · ENTERPRISE ARCHITECTS	The initial hypothesis and its alternatives, how agents are built and deployed, and how integration works at runtime. §1.4–1.5 → §2 → §4 → §5 → Appendices A, C
SECURITY & AUDIT CISO · CRYPTO ARCHITECTS · INTERNAL AUDIT	Identity for software agents, the threat model, policy enforcement, and what the proofs do and do not cover. §2.4–2.8 → §3 → §6.2 → Appendix B

■ THE PAPER IN ONE PARAGRAPH

Enterprise information systems were designed for people entering transactions; autonomous software agents can now do much of that work. This paper proposes an agent-native architecture in which every agent has a verified identity and every action is checked against delegated human authority before it runs, then recorded in a log that makes later changes detectable. It sets out how the proposal could be tested, starting with ERP.

RECEIPT	A record of an action, with a digital fingerprint that shows whether it was changed.
HASH	A digital fingerprint: a short code calculated from data that changes if the data changes.
AGENT-NATIVE	Designed on the assumption that software agents initiate, coordinate and execute business work (§1.4).
MCP	Model Context Protocol, an open standard that lets agents discover and use other systems.

Executive Summary

Enterprise information systems coordinate the decisions, obligations and records through which an organization operates. This paper examines how their architecture should change when autonomous software agents become the primary executors of digital operational work.

■ THE GUIDING QUESTION

How should enterprise information systems be designed if software agents are their primary operators?

01 The proposal

This paper proposes the Agent-Native Enterprise Operating System (AE-OS), a framework in which agents are first-class participants. They receive explicit authority, work with shared business meaning, coordinate through defined processes and produce verifiable records of their actions.

Humans remain responsible for organizational goals, delegated authority, control design and decisions that require judgment. Deterministic programs calculate regulated figures and keep the books of record.

02 Why ERP comes first

ERP brings accounting integrity, operational coordination, security and accountability into the same system, so it tests every part of the framework. It is also where many organizations will rebuild soon: SAP ends mainstream maintenance of its ECC system after 2027^[1], and Gartner estimates that about 61% of roughly 35,000 ECC customers had not migrated by end-2024^[2]. Systems chosen now will run while agents take on a growing share of the work.

03 What agent-native means

An enterprise information system is **agent-native** when its core contracts assume that software agents will initiate, coordinate and execute business work.

Business actions must then expose their meaning, the authority they require, their preconditions and their outcomes. An agent must be able to tell what an action does, whether it may perform it and whether it succeeded. People set objectives and limits, approve consequential exceptions and can intervene at any point (§1.4).

04 Authority and evidence

An organization remains accountable for what its agents do. Its auditors, regulators and board need two answers for every agent action: who authorized it, and what record proves what happened.

Who authorized it. Most software, agents included, proves its identity with keys stored in code or configuration. These leak often (28.65 million new secrets were exposed on public GitHub in 2025^[3]), and anyone holding a leaked key can act as the agent.

What happened. The organization needs a record of each agent's instruction, the rule that allowed the action, the reasoning and the result. For high-risk AI systems, the EU AI Act already requires automatic event logging over the system's lifetime^[4].

05 Thesis and scope

The paper argues that agents can do most digital operational work without loss of control or auditability, provided identity, business meaning, policy, transaction integrity and evidence are supplied once beneath every agent (§1.6). It sets out a reference architecture, demonstrates its building blocks in code and proofs, and states what evidence would confirm or refute the thesis (§6).

6 OPEN RESEARCH PROBLEMS (§6.1)	1 EVALUATION WORKFLOW (§6.1)	95 ERP CAPABILITIES MAPPED (APPENDIX C)	15 OPEN RESEARCH QUESTIONS (Q1-Q15)
---------------------------------------	------------------------------------	---	---

06 Components of the proposed architecture

The paper proposes a control foundation and, on top of it, business layers that do the work an ERP does today. Each part corresponds to something finance and audit teams already know. The specific technologies named in §2–4 are candidate ways to meet each requirement; §1.5 lists the alternatives to test.

PART	WHAT IT DOES	FAMILIAR EQUIVALENT
BUSINESS LAYERS (§2.9–2.17)		
Books of account Accounting Kernel	Double-entry ledgers, chart of accounts, parallel accounting standards, periods and legal numbering. Every entry must balance before it is recorded.	The general ledger and sub-ledgers in the ERP.
Calculators Calculation Engines	Tax, depreciation, payroll, revenue and currency calculations run in fixed, testable programs. Agents call them; regulated numbers always come from these programs.	Tax and payroll engines, with each result recorded so it can be re-run.
Processes and packs Process Layer • Domain Packs	Purchase-to-pay, order-to-cash, close and payroll, delivered as packaged modules with country rules for tax and e-invoicing.	ERP modules, localizations and batch schedules.
CONTROL FOUNDATION (§2.1–2.8, §3)		
Identity	Every person, agent and tool has a verified identity. People can also prove who they are with digital identity wallets (§2.18).	A staff ID badge checked at every door.
Rules	Every action is checked against policy before it runs: limits, segregation of duties, sanctions.	Delegated authority and Sarbanes-Oxley (SOX) controls, applied up front.
Logbook	Each action is recorded with its instruction, approval, rationale and result, linked by digital fingerprints, so later changes are detectable.	An audit trail kept like the public web-certificate logs (Certificate Transparency) Chrome has required since 2018 ^[5] .

07 Implications for finance and audit

- **Controls run before payment.** A payment that breaks a limit or a segregation-of-duties rule is blocked or escalated.
- **Evidence is produced as work happens.** Each record carries a receipt proving it has not been altered since logging. Auditors still test the controls; the design targets a SOC 1 report, an independent auditor's report on a service provider's controls over financial reporting (§2.17).
- **Managers set the limits.** They sign the goals and limits for agents and approve each exception.
- **Validation runs beside the existing ERP.** Proposed components first run in shadow mode, processing the same work without posting to the books, and are reconciled against it before they take responsibility for production work (§2.16).

08 How the thesis would be tested

The proposed test is one finance workflow, supplier invoice to payment, run beside an existing ERP and measured for the share of work agents complete, exception rates and reconciliation (§6). Signed receipts for agent actions can be added to existing ERP systems first, so evidence can be gathered before anything is replaced. Section 7 sets out the implications for finance and audit leaders and the paper's limitations.

09 Economics

Appendix D gives an illustrative seven-year model for a company with about 1,000 ERP users and \$1.5B revenue, comparing an SAP subscription route with an AE-OS route that pays no license fee. It shows which assumptions the economic case depends on.

BASE CASE, 7 YEARS	SAP ROUTE	AE-OS
Cost to build and run	\$28.2M	\$27.1M
Measurable value: less manual finance work, lower audit cost, cheaper partner links	–	\$14.0M
Net advantage today (NPV)	–	\$10.5M

MODELED. Projections from the assumptions in Appendix D, not measured results.

■ MODELED • HOW TO READ THIS

Running costs come out similar on both routes; any advantage depends on how much finance work agents absorb. In the conservative case AE-OS only breaks even, so that share is the main economic hypothesis for a pilot to measure.

01 BACKGROUND & THE AGENT-NATIVE QUESTION

Background and the Agent-Native Question

ABSTRACT

The Agent-Native Enterprise Operating System (AE-OS) is a proposed open framework for agent-native enterprise information systems, with ERP as its first application. It assumes digital operations performed mainly by software agents under human authority. The initial architectural hypothesis has four planes that form a control and evidence foundation: business state in an event-sourced knowledge graph sharded across actors (**Semantic Data Plane**); a short-lived cryptographic identity for every human, agent and tool, with SPIFFE as the candidate standard (**Identity Plane**); segregation-of-duties, spend and SOX controls evaluated in a sandboxed policy runtime (candidate: WebAssembly) before any action (**Compliance Layer**); and an append-only Merkle-DAG linking each posting to the signed intent, policy decision, recorded rationale and tool call that caused it (**Causal State Ledger**). A proposed decomposition of enterprise layers, starting with an Accounting Kernel, sits on that foundation (§2.9–2.17). This section gives background on current ERP architecture (§1.1–1.3), defines agent-native (§1.4) and sets out the hypothesis with its alternatives (§1.5).

1.1 The Current ERP Migration Cycle

Many large organizations will redesign their core systems in the next few years. SAP ends mainstream maintenance for Business Suite 7 (ECC) at end-2027, with extended maintenance to 2030^[1] and a 2031–2033 transition option for qualifying large systems^[6]. Gartner puts migrations at ~39% of ~35,000 ECC customers by end-2024^[2], and 54% of companies in DSAG's 2026 investment survey still run ECC somewhere^[7]. These programs are hard: Gartner expects over 70% of recently implemented ERP initiatives to miss their original business goals by 2027^[8], and public cases such as Lidl's halted SAP program^[9] and Birmingham City Council's Oracle system^[10] show the cost of re-fitting decades of process-specific customization onto a fixed data model. The systems chosen now will operate through a period in which agents take on a growing share of digital operational work. Sections 1.2 and 1.3 examine where today's architecture meets that workload.

1.2 Governance and Concurrency Limits of ACDOCA

S/4HANA records GL, Asset Accounting, Controlling, CO-PA and Material Ledger line items in one table, the Universal Journal ACDOCA^[11], ~350 fields wide before extensions^[12]. Totals and index tables (GLT0 , BSIS) became read-time views^[13], removing the classic hot totals row. Inserts land in HANA's write-optimized delta store and are merged into compressed column storage^[14]; SAP advises scale-up first, and S/4HANA supports multi-node only with restrictions^[15]. Serialization survives at the document number: unbuffered ranges lock the NRIV row until COMMIT WORK , which "can seriously impair the system performance" under parallel load, and FI's RF_BELEG ships unbuffered because buffering breaks the gapless sequences some jurisdictions require^[16].

ANALYSIS

The main gap is governance: the incumbent stack lacks identity, pre-execution policy and verifiable provenance for non-human principals. Agents also add many small concurrent postings; each gapless series serializes on one number-range lock, and a pilot would test whether that binds.

AE-OS SPECIFIES

- Single-partition journal entries (§2.10)
- Escrow for availability checks
- Numbering decoupled from posting
- A Causal Receipt per posting

1.3 Customization Coupled to the Vendor Release

Customization is code compiled into the vendor's transaction: Z-tables, user exits, BAdIs and batch chains writing the same tables as the standard. 67% of SAP customers run custom code and 58% call it a barrier to upgrading or migrating^[17]; Birmingham's customizations broke bank reconciliation^[10]. Because logic, schema and vendor release share one address space, every upgrade becomes a migration. AE-OS proposes to separate them: policies become content-addressed modules (candidate runtime: Wasm), integrations become MCP tools discovered at runtime, and state changes become Causal Events.

01 BACKGROUND & THE AGENT-NATIVE QUESTION

1.4 What agent-native means

■ DEFINITION

An enterprise information system is agent-native when its core contracts assume that software agents will initiate, coordinate and execute business work.

Business actions must therefore expose explicit meaning, authority requirements, preconditions and outcomes. Agents must be able to discover what an action does, determine whether they may perform it and establish whether it succeeded. Concurrency, retries, uncertain outcomes and conflicting objectives must be addressed within the architecture.

Human authority must remain explicit throughout this process. People establish objectives and limits, approve consequential exceptions and retain the ability to intervene. Agent autonomy operates within those boundaries.

AE-OS proposes that identity, business semantics, policy enforcement, transaction integrity and evidence should form a common foundation for this operating model. The architectural question is whether that foundation can support dependable enterprise operations across independently developed components.

The definition separates an agent-native system from enterprise software with agents added to it, where an agent drives screens or APIs designed for people and the system cannot tell what authority the agent holds, why it acted, or whether a retried call was applied twice. TABLE 1.1 lists the contracts that change.

CONTRACT	AGENTS ADDED TO EXISTING SOFTWARE	AGENT-NATIVE REQUIREMENT	PROPOSED IN
Meaning	Fields and transaction codes designed for trained users	Each action and entity has a typed, versioned definition an agent can discover and query	\$2.3, \$4.6
Authority	The agent borrows a user's or service account's role	Each call carries the agent's own identity and a capability bound to a signed human intent	\$2.6–2.7, \$4.3
Preconditions	Checked inside the application, or after the fact by audit	Declared with the action and evaluated by policy before any side effect	\$3.7, \$2.17
Outcomes	A success screen or HTTP status	A receipt that proves what was recorded, with idempotency keys so retries are safe	\$3.2–3.6, \$2.12
Uncertainty	Left to the agent's error handling	A failure model for actions whose external effect or acknowledgement is unknown, with recovery steps	\$2.12; open (\$6.1)
Concurrency	Locks sized for human transaction rates	Per-entity serialization, escrow for shared balances, sagas across entities	\$2.3, \$2.10, \$2.12
Conflicting goals	Resolved informally by the people involved	Objectives and limits stated as signed intents; conflicts escalate to a named person	\$5.1–5.2
Human authority	Approval workflows for human requests	Escalation of consequential exceptions, with evidence, plus the ability to pause or revoke any agent	\$4.2, \$5.1

TABLE 1.1 – Contracts an agent-native system makes explicit, and where this paper proposes how.

1.5 Initial hypothesis and alternatives

The four planes of \$2 are an initial hypothesis for how to provide this foundation. The technologies they name (Ray, SPIFFE, WebAssembly, MCP and others) are candidate ways to meet each requirement, and their inclusion alone does not establish the framework's value. TABLE 1.2 records the alternatives and the trade-offs to test.

REQUIREMENT	CANDIDATE IN THIS PAPER	ALTERNATIVES	TRADE-OFF TO TEST
Semantics and state (L1)	Typed ontology; event-sourced partitions owned by actors (Ray)	Relational core with an event outbox; a semantic layer over an existing ERP data model	Scale-out and per-entity ordering against the maturity, tooling and reporting ease of a relational core
Identity (L2)	SPIFFE/SPIRE workload identity with short-lived SVIDs	OAuth 2.0 client credentials with token exchange; cloud-provider workload identity	Attestation and platform neutrality against one more component to operate
Policy (L3)	Content-addressed policy modules in sandboxed WebAssembly	Cedar or OPA evaluated natively; rules inside application code	One portable, deterministic artifact against faster adoption and existing analysis tools
Transaction integrity	Balanced single-partition journal entries; escrow; sagas	Distributed transactions over partitions; one central ledger database	Throughput and failure isolation against simpler reasoning about atomicity
Evidence (L4)	Merkle log with signed, witness-cosigned checkpoints (RFC 9162 method)	Write-once audit storage; a shared distributed ledger	Portable proofs without consensus against simpler operation, or shared operation at higher cost and latency
Tool access	MCP servers behind an identity-aware gateway	OpenAPI-described services called through model function calling; integration platforms	Runtime discovery designed for agents against the maturity of existing interface standards

TABLE 1.2 – Initial hypothesis, alternatives and trade-offs. Selections should be reviewed as evidence develops (\$6).

01

BACKGROUND & THE AGENT-NATIVE QUESTION

1.6 Research question, thesis and method

This paper is a work of design research. It states a question, proposes an answer in the form of a reference architecture, and sets out how that answer could be shown to be wrong.

Research question

RQ. What architecture allows an enterprise information system to have most of its digital operational work performed by autonomous software agents while keeping the control and auditability that organizations, auditors and regulators require?

- RQ1** Which contracts must an enterprise system make explicit before agents can act in it safely (§1.4)?
- RQ2** How can authority delegated by a person or an organization be carried to an agent and enforced before each action (§2.6–2.8, §2.18)?
- RQ3** How can every agent action be recorded so that a third party can verify the record without trusting the operator (§3)?
- RQ4** Which accounting and control properties must hold for the books to remain reliable when agents do the work (§2.10, §2.17)?

Method

The paper follows design science research, which builds and evaluates artifacts that solve a stated problem^[18]. Of the six activities in the method of Peffers and colleagues (problem identification, objectives, design, demonstration, evaluation and communication)^[19], this paper completes the first three. It demonstrates individual primitives in code and proof sketches, and it defines how the remaining evaluation would be done. Following the framework of Venable and colleagues, the evaluation so far is formative and artificial: test vectors, cross-implementation agreement and proofs^[20]. Naturalistic evaluation in a working organization remains to be done (§6).

In the terms of Gregor and Hevner, the contribution is an improvement: a new solution to a known problem^[21]. The problem, control of automated work in finance systems, is long established; the solution applies workload identity, policy enforcement before execution and transparency-log evidence to it.

■

THESIS

Enterprise information systems can delegate most digital operational work to autonomous software agents without loss of control or auditability, provided that identity, business semantics, policy enforcement, transaction integrity and evidence are supplied once, as a common foundation beneath every agent.

Contributions

#	CONTRIBUTION	ANSWERS	WHERE	EVALUATION IN THIS PAPER	STATUS
C1	A definition of agent-native and the contracts it requires	RQ1	§1.4, TABLE 1.1	Argument from requirements	Proposed
C2	A reference architecture: four control planes and six enterprise layers	RQ1, RQ4	§2, App. C	Requirement coverage mapped against 95 ERP capabilities	Proposed; alternatives in TABLE 1.2
C3	A delegation model from a person's or organization's credential to an agent's action	RQ2	§2.6–2.8, §2.18–2.19, §4.3	Attenuation demonstrated in reference code	Partly demonstrated
C4	A causal evidence ledger with receipts for every agent action	RQ3	§3, App. A–B	Proof sketches; Python and Rust implementations agree	Primitives demonstrated
C5	Accounting invariants and a control mapping for auditors	RQ4	§2.10, §2.17	Mapped to SOC 1 control objectives	Proposed
C6	An evaluation framework: measures, open problems and an illustrative economic model	All	§6, App. D	Defines the evidence that would confirm or refute C1–C5	Proposed; economics MODELED

TABLE 1.3 – Contributions of this paper. None has been evaluated in a working organization.

Structure of the paper. Section 2 sets out the architecture, Section 3 the evidence ledger, Section 4 the developer primitives and Section 5 human oversight with a worked scenario. Section 6 gives the evaluation design and research agenda, and Section 7 concludes with implications and limitations. The appendices hold reference code, proof sketches, the scope map and the economic model; the final pages list open research questions.

02

PROPOSED ARCHITECTURE

The Four Planes and How They Connect

Under the hypothesis of §1.5, four planes, each a separate failure domain for **state**, **identity**, **policy** and **proof**, form the control and evidence foundation beneath the Accounting Kernel (§2.10).

2.1 Design Principles

- **P1 • Identity before connectivity.** A workload gets an SVID only after attestation; only SVID holders are routed traffic.

■ **P2 • Policy before execution.** A sandboxed module checks every action before side effects; its verdict is a ledger node.

■ **P3 • Event-sourced state.** The record is an append-only log of Causal Events. Balances and reports are projections.
- **P4 • Query by meaning.** Agents query entities by meaning (typed ontology plus embeddings) rather than by table and column name.

■ **P5 • Verifiable receipts.** Each action returns a Causal Receipt, a Merkle proof of integrity; auditors still test controls (§2.17).

■ **P6 • Fail closed, degrade locally.** An undecidable check denies. Faults stay within one shard, trust domain or module.

2.2 Overview of the Four Planes

PLANE	PRIMITIVE	GUARANTEE	IF ABSENT
L1 Semantic Data	Typed entity, Causal Event	Per-entity serializable; scale-out	Journal contention (§1.2)
L2 Identity	SPIFFE ID, SVID, capability	Calls bound to attested workloads	Bearer-secret theft (§2.5)
L3 Compliance	Policy module, PolicyDecision	Deterministic, pre-execution	Post-hoc audit only
L4 Causal State Ledger	CID node, Causal Receipt	Tamper-evidence, provenance	Mutable, unprovable logs

2.3 Semantic Data Plane

Typed knowledge graph. Business state consists of typed entities (`Party`, `Item`, `Order`, `Obligation`, `LedgerPosting`) joined by typed relations (`owes`, `fulfils`, `offsets`). The ontology is content-addressed, so a trace names the exact schema it read. **Actor sharding:** exactly one actor owns each entity, located by `hash(entity_id)` on a consistent-hash ring. AE-OS adopts Ray's model of stateless tasks and stateful actors for placement and execution only; its >1.8M tasks/s^[22] is scheduler headroom, not ledger commit throughput (to be benchmarked in a pilot). Durability comes from the replicated per-partition event store (§3.9). Commands serialize in each entity's mailbox, without locks.

Event sourcing. Every change to application state is captured as a sequence of events^[23]. An actor validates a command, appends Causal Events to its partition, then applies them. Snapshots bound replay; read models (trial balance, open items, stock) are projections with a watermark.

Replacing the single journal. Instead of one `ACDOCA` table^[11], AE-OS keeps partitioned, append-only streams. A journal entry is one atomic, balanced event on one partition (legal entity × ledger) in the Accounting Kernel (§2.10), never split across actors. Gapless legal numbers come from one sequencer per legal entity and document series, assigned in order after the posting commits and before external issue; clearance regimes wait for it. **Escrow counters**^[24] serve availability checks (cash, stock, credit), with a per-account coordinator reallocating quanta; **CRDTs**^[25] hold only non-ledger counters (KPIs). Cross-entity processes are sagas (§2.12); personal data is sealed under per-subject keys (§2.15).

$$\text{avail}(a) = \text{bal}(a) - \sum_i \text{reserved}_i(a) \qquad \text{reserve}(q) \text{ commits} \Leftrightarrow \text{avail}(a) - q \geq \text{floor}(a) \quad (\text{Eq. 2.1})$$

Query by meaning. An intent like "unpaid obligations to suppliers in sanctioned jurisdictions" maps to ontology terms (embeddings resolve fuzzy names) and compiles to a typed graph query, gated by an L3 field check.

ENTITY MODEL	Typed property graph; ontology CID-pinned	HOT BALANCES	Escrow; PN-counters for KPIs only
SHARD KEY	<code>hash(entity_id)</code> , consistent ring	LEGAL NUMBERING	Sequencer per legal entity × series
ATOMICITY	Journal entry: one partition	QUERY	Intent → ontology → graph plan
EVENT STORE	Append-only per partition; DAG-CBOR	CONSISTENCY	Linearizable per partition; causal across

02 THE ARCHITECTURE OF AE-OS

AE-OS Reference Architecture

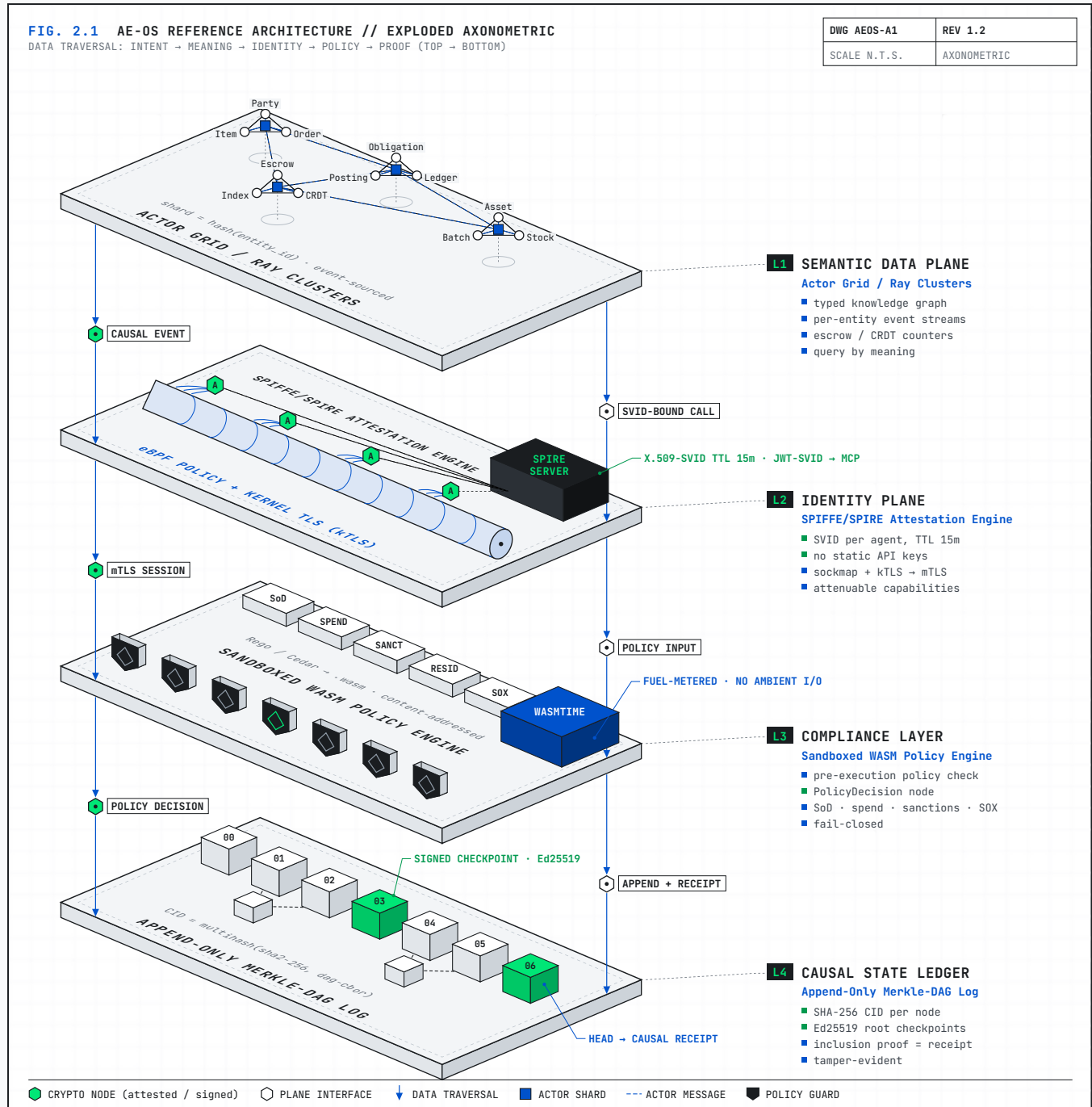


FIG. 2.1 – Exploded view of the four substrate planes (design proposal), tracing one agent action down the stack. The enterprise layers, including the Accounting Kernel, appear in FIG. 2.3.

L1 → L2 • COMMAND

The agent maps an intent to typed entities and calls the owning actor with its **X.509-SVID** and a capability scoped to the signed intent (Intent Envelope, §3.2).

L2 → L3 • ADMISSION

The eBPF datapath admits only policy-allowed mTLS flows. Verified identity, caveats and command form the **policy input** to a fuel-metered Wasm module.

L3 → L4 • PROOF

The verdict becomes a **PolicyDecision** node. An allowed Causal Event is appended, CID-linked to intent, decision and tool call; its inclusion proof is the **Causal Receipt**.

PLANE	OPEN-SOURCE BASIS • PRODUCTION EVIDENCE	AE-OS ADDS (DESIGN)
L1 Semantic	Ray actors/tasks; >1.8M tasks/s scheduled, not committed (OSDI) [22]	Ontology; escrow; KPI CRDTs
L2 Identity	SPIFFE/SPIRE, Cilium (CNCF grad. 2022/2023) [26,27] • Uber 4,500 svcs [28]	Attenuable intent-bound tokens
L3 Compliance	Wasmtime fuel [29] • OPA Rego → Wasm [30] • Cedar 4–11 μs median (varies) [31]	Policy CIDs; logged decisions
L4 Ledger	CT (RFC 9162; Chrome, public certs, since 2018) [32,5] • Sigstore Rekor [33]	DAG links; per-action receipts

02

ZERO-TRUST CRYPTOGRAPHIC IDENTITY PLANE

Non-Human Identity Risk in Agent Fleets

Agent fleets multiply non-human principals, and the bearer credential is already a leading breach vector. AE-OS removes long-lived secrets from everything an agent can read, and confines the remaining external credentials to the gateway and HSM.

82:1 MACHINE : HUMAN IDENTITIES; 42% PRIVILEGED^[34]	28.65M SECRETS LEAKED ON PUBLIC GITHUB, 2025 (+34%)^[3]	64% OF 2022-LEAKED SECRETS STILL VALID IN 2026^[3]	22% OF BREACHES BEGIN WITH CREDENTIAL ABUSE^[35]
--	---	--	--

2.4 Scale and Exposure of Non-Human Identities

Machine identities already outnumber humans 82 to 1, and 68% of the 2,600 organizations surveyed lack identity security controls for AI^[34]. Secrets leak faster than they are revoked: AI-service credentials on public GitHub rose 81% to 1.28 million in 2025. 24,008 unique secrets appeared in MCP configuration files, the integration surface agents use, and 2,117 of them were valid^[3]. Private repositories are also exposed: 35% of those scanned contained plaintext secrets^[36]. Data in the 2025 DBIR put the median time to remediate a secret leaked to GitHub at 94 days^[37]. OWASP's top ten risks for non-human identities (NHIs) include secret leakage, long-lived secrets and overprivileged NHIs^[38]. Its 2026 agentic list names *Identity & Privilege Abuse* (ASI03) and *Insecure Inter-Agent Communication* (ASI07)^[39].

INCIDENT	NON-HUMAN CREDENTIAL	FAILURE	CONSEQUENCE
Cloudflare · Nov 2023	1 service token + 3 service accounts	Not rotated after Okta breach ("out of thousands")	Atlassian + source access; >5,000 credentials rotated, 4,893 systems triaged ^[40]
Microsoft · Jan 2024	Legacy test OAuth app	Elevated access from a non-MFA test tenant	full_access_as_app on Exchange Online; corporate mail read ^[41]
Dropbox Sign · Apr 2024	Back-end service account	Broad production privileges	Customer API keys, OAuth tokens, MFA data exposed ^[42]
Snowflake tenants · 2024	Static user credentials	Infostealer-sourced, no MFA, unrotated up to 4 years	~165 organizations notified ^[43]
Internet Archive · Oct 2024	GitLab token → Zendesk API tokens	Token in exposed config; keys not rotated	Support-ticket system accessed ^[44]
tj-actions · Mar 2025	CI/CD secrets in runners	Compromised action dumped runner memory	Secrets printed to logs; >23,000 repos used it ^[45]

2.5 Why Application-Layer Secrets Fail for Agent Fleets

The incidents share one root cause: **possession equals identity**. An API key, OAuth client secret or service-account password is a *bearer* artifact. It is not tied to the workload, host, task or time window it was issued for, and whoever copies it becomes the principal. Remediation depends on humans finding and rotating every copy, and Cloudflare missed four out of thousands. Agents amplify each weakness:

- **Ephemeral**. A planner may spawn hundreds of task instances per hour. A secret per instance outruns any vault rotation workflow; a shared secret erases per-instance attribution.
- **Composable**. Agents call tools that call agents. A key forwarded down the chain carries full authority to every hop, the confused-deputy pattern at fleet scale.
- **Steerable**. Prompt injection can direct an agent to read and exfiltrate any credential in its context, environment or MCP configuration (ASI01, ASI03).

DESIGN CONSEQUENCE

AE-OS specifies that **long-lived secrets are kept out of every location an agent can read**: environment variables, configuration, MCP manifests and prompts. Identity derives from **attestation** and is expressed as a 15-minute SVID; transport security is enforced **below** the application; authority is a separate, attenuable token bound to one intent. Credentials that external systems still require (bank channels, SaaS OAuth) stay in the gateway vault and HSM (§2.8).

02 ZERO-TRUST CRYPTOGRAPHIC IDENTITY PLANE

2.6 SPIFFE Identities, SVIDs and Attestation

A SPIFFE ID is a URI, `spiffe://<trust-domain>/<path>`. An SVID carries the ID, a valid signature and an optional public key, encoded as X.509 or JWT^[46]. Agents receive `spiffe://<tenant>.ae-os/agent/<role>/<instance>`, tools and actor hosts analogous paths, and human sessions a pseudonymous `.../human/<opaque-id>` that only the identity provider maps to a person; logs carry keyed digests rather than names (§2.15). X.509-SVIDs authenticate transport and audience-bound JWT-SVIDs authenticate MCP calls; the X.509 TTL is 15 min against SPIRE's 1 h default, and JWT-SVIDs default to 5 min^[47]. In **node attestation** the SPIRE agent proves its host with TPM DevID, a cloud instance document or a Kubernetes PSAT^[47]; in **workload attestation** it derives kernel selectors for the caller on the local Workload API socket, then issues an SVID whose private key never leaves the process (Fig. 2.2). SPIRE's signing keys are RSA or ECDSA P-256/P-384 (default P-256); Ed25519 is not offered^[47]. Ledger leaves are therefore signed with the actor's ECDSA SVID key and record the SVID chain CID, and inclusion under a checkpoint signed by the HSM-held Ed25519 log key serves as the trusted timestamp. Human approvals use the approver's WebAuthn/FIDO2 credential, bound to the SPIFFE session.

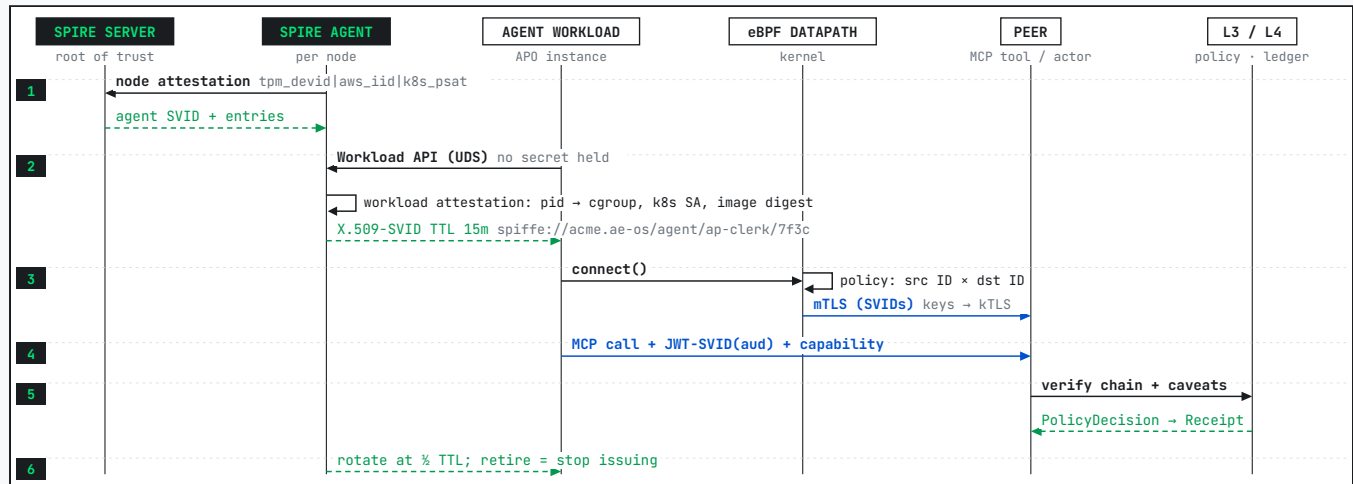


FIG. 2.2 – One agent action: (1) node and (2) workload attestation, (3) kernel mTLS, (4) MCP call with JWT-SVID + capability, (5) PolicyDecision, (6) rotation. Black: call; green dashed: SVID/receipt; blue: mTLS.

2.7 Kernel-Level Enforcement and Intent-Bound Capabilities

Kernel enforcement. Identity checks run in the kernel datapath, so an unprivileged agent process, including a prompt-injected one, cannot bypass them; kernel or host compromise is a residual risk (§2.8). On `connect()`, eBPF socket/cgroup hooks check (`src ID` × `dst ID`) against default-deny policy. The SVID-based mTLS handshake hands session keys to kTLS, which runs the record layer in-kernel^[48]. Cilium's SPIRE-backed out-of-band handshake is still labeled beta and validated only with SPIRE^[49], a maturity risk.

Capabilities. The SVID says *who*. Authority is a token minted from the Intent Envelope, combining macaroon attenuation^[50] with Biscuit's public-key verification and Datalog checks^[51]. Each hop can only add caveats, so authority only narrows. The subject must match the mTLS peer ID, so a stolen token is inert anywhere else.

```
TOKEN // CAPABILITY (BISCUIT-STYLE)

// authority: minted from Intent Envelope
right("spiffe://acme.ae-os/agent/ap-cLerk/7f3c",
      "ledger.post");
intent("bafyre1...7q4"); // envelope CID
// block 1: planner attenuates (EUR 25,000.00)
check if amount_minor($a), $a <= 2500000;
reject if vendor_country($c), sanctioned($c);
check if time($t), $t <= 2026-10-12T17:00:00Z;
// block 2: tool hop narrows to one invoice
check if resource("ap/invoices/INV-88213");
// verifier: subject == mTLS peer SPIFFE ID
```

2.8 Threat Model

THREAT	OWASP ^[38,39]	AE-OS CONTROL	RESIDUAL RISK
Leaked secret replay	NHI2, NHI7	Attested SVIDs; key stays in process; TTL 15 min	Live key on compromised host
Stolen token reused	NHI4, ASI03	Audience-bound JWT-SVID; subject = mTLS peer	Same-host theft within TTL
Overprivileged agent	NHI5, ASI03	Per-intent attenuated capability; L3 check per action	Policy authoring errors
Orphaned identity	NHI1	APO (Agent Primitive Object, §4.1) RETIRED deletes entry; SVID lapses	≤ 15 min window
Rogue / spoofed agent	ASI07, ASI10	Attestation incl. image digest; agent-to-agent mTLS	Compromised build attests
Injected exfiltration	ASI01, NHI2	Secret-free agent context; identity-scoped egress	Misuse within scope
CA key compromise	—	HSM/KMS CA keys; nested trust domains; issuance in L4	Root of trust critical
External credentials	NHI2, NHI7	Gateway vault/HSM; never in agent context	Gateway compromise
Same-principal agents	ASI03	SoD on accountable principals, not SPIFFE IDs	Human collusion

02 ENTERPRISE STACK • PROPOSED DECOMPOSITION

The Enterprise Stack

The four planes govern how agents act. AE-OS proposes six enterprise layers, built from the planes' primitives, for the books, calculations, processes, reporting and screens an enterprise also needs.

2.9 Enterprise layers on the four planes

Sections 2.2–2.8 specify control, identity and evidence: who may act, under which policy, with what proof. That suffices to govern agents working over an existing ERP. An enterprise whose operations run on AE-OS also requires books of record that balance by construction, deterministic tax, depreciation and payroll calculations, calendar-driven processes (payment runs, the close), packaged business functionality, reporting, and screens for back-office staff. AE-OS proposes assigning these to six layers, E1–E6, and three cross-cutting frameworks (FIG. 2.3). Each layer keeps its state in L1 actors and Causal Events, its callers hold L2 identities, its rules run in the L3 sandbox and its outputs are logged in L4.

DIVISION OF RESPONSIBILITY

Agents propose actions and handle judgment steps. Calculation engines (E2) compute regulated amounts. The Accounting Kernel (E1) books them as balanced journal entries. The Causal State Ledger (L4) holds the evidence. People approve exceptions.

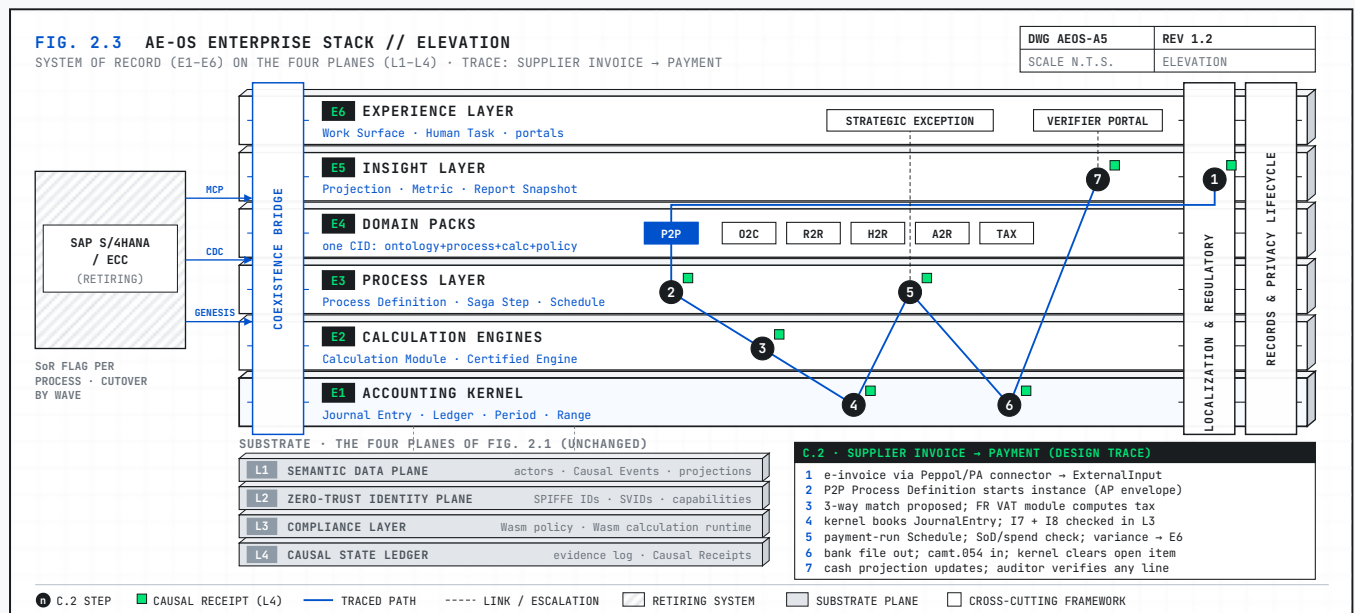


FIG. 2.3 – AE-OS Enterprise Stack (design proposal). E1–E6 sit on the unchanged planes; three frameworks span all six. The Coexistence Bridge reaches SAP via MCP, change-data capture and genesis events. Markers 1–7 trace one supplier invoice.

LAYER	RESPONSIBILITY	CORE PRIMITIVES	BUILT ON
E6 Experience	Intent console, Work Surfaces, portals	Work Surface, Human Task	L2 human SVID; L4 approvals
E5 Insight	Statements, metrics, plans, export	Projection, Report Snapshot	L1 projections; L4 checkpoints
E4 Domain Packs	Business process packs (TABLE 2.13)	Domain Pack (one CID)	L1 ontology; L2 APOs; L3 policy
E3 Process	Durable workflows, schedules, sagas	Process Definition, Schedule	Each step runs L2 → L3 → L4
E2 Calculation	Tax, depreciation, FX, revenue, payroll	Calc. Module, Certified Engine	L3 Wasm runtime; L2 for MCP
E1 Accounting Kernel	Double-entry books, periods, numbering	Journal Entry, Ledger, Period	L1 partitions; L3 invariants
Frameworks	Localization, records & privacy, coexistence	Localization Pack, KeyShred	E2; I5 sealing; L4 genesis

Evidence and books are separate records

Earlier sections use "ledger" for the **Causal State Ledger**, an evidence log of what agents and people did. Books of account have other invariants: a Merkle proof shows an entry is unaltered, but says nothing about whether debits equal credits, the period was open or the VAT is right. AE-OS keeps the books in the **Accounting Kernel** (§2.10) and links each journal entry by CID to the intent, decision and calculation behind it.

	L4 LEDGER	E1 KERNEL
Records	every action, decision	financial postings
Unit	Causal Event	balanced Journal Entry
Invariant	append-only, hashed	$\Sigma dr = \Sigma cr$; open period
Correction	none: history is fixed	reversal entry
Answers	who, why, which rule	what balance, as at when

02 THE ENTERPRISE STACK

2.10 Accounting Kernel (E1)

The **journal entry is the atomic document**. A `JournalEntry` is one Causal Event with two or more lines that balance per ledger and per currency. It is written by one append to one partition, keyed by legal entity × ledger; balances and open-item lists are projections of that stream. This refines the atomicity rule of §2.3 (one actor per entry; sagas across entities). A saga is a sequence of transactions, each paired with a compensating transaction, whose intermediate states are visible to others^[52]. AE-OS uses sagas between documents, for example the two halves of an intercompany pair or the steps of a P2P process, and never inside one, so no entry can be half-posted. The ledgers of one legal entity are colocated on one shard, and a posting to a ledger group commits to all of them in one local append. Escrow counters (Eq. 2.1) remain for availability checks on cash, stock and credit and are never read as book balances.

Structure. Each legal entity has an operational chart of accounts, instantiated from a Localization Pack template and mapped to the group chart. Accounts are typed entities (balance sheet or P&L, open-item managed, reconciliation, account currency). A **leading ledger** (e.g. IFRS) runs beside **parallel ledgers** for local GAAP and tax; valuation differences post ledger-specific entries. Every line carries the **currency triple**: transaction, local and group amounts, translated from a CID-pinned rate table under IAS 21^[53]. The FX module also emits any rounding line, so the entry balances in all three currencies.

```
FILE // SPEC/KERNEL/JOURNAL_ENTRY.RS (SKETCH)

struct JournalEntry {    // EventKind::JournalEntry
  entity: LegalEntity,   // partition key, part 1
  ledger: LedgerId,      // "0L" IFRS, "2L" local
  doc_no: Option<DocNo>, // set by sequencer at append
  period: (u16, u8),     // fiscal year, period (I9)
  lines: Vec<Line>,      // ≥ 2; Σ = 0 per ccy (I7)
  reverses: Option<Cid>, // original entry
  ic_pair: Option<Cid>,  // partner-entity half
  cause: Vec<Cid>,       // ToolExecution, intent
}

struct Line {
  account: AccountId,
  amount: [(Dec128, Ccy); 3], // txn, local, group
  partner: Option<LegalEntity>, // trading partner
  item: Option<OpenItem>,      // clearing key
  calc: Option<Cid>,           // CalculationExecution (I8)
}
```

KERNEL INVARIANTS

- **I7 · Balanced books.** A `JournalEntry` is appended only if it has ≥ 2 lines on valid accounts and its lines sum to zero per ledger and per currency.
- **I8 · Calculated amounts.** Every amount an agent posts cites a `CalculationExecution` or a signed human judgment.
- **I9 · Closed periods.** No `JournalEntry` is dated in a closed `PostingPeriod`; a correction is a reversal dated in an open period.

I7–I9 extend I1–I6 (§3.1) and run as mandatory L3 modules before the append; a violation is a logged DENY.

Posting periods exist per entity × ledger, with special periods for year-end adjustments. A period moves OPEN → RESTRICTED → CLOSED, or reopens, only by a controller-signed `PeriodTransition`.

Number ranges. EU VAT law requires each invoice to carry "a sequential number, based on one or more series, which uniquely identifies the invoice"^[54]; some jurisdictions require gapless series. SAP holds an unbuffered range's row lock until commit^[16] (§1.2). The AE-OS partition sequencer issues the legal number inside the append that commits the entry, after calculation and policy checks pass, so only posted entries consume numbers and each series (entity × range × year) stays gapless. The lock covers one in-partition append: this is the "numbering after commit" of §2.3.

Corrections, clearing, groups. A reversal is a new entry with negated lines citing the original. Clearing links debit and credit open items (vendor, customer, GR/IR, bank) and posts discount or FX differences in the same entry. Intercompany halves post in each entity's partition with a shared `ic_pair`; a projection lists unmatched halves. Group amounts and partner fields feed consolidation (E5); eliminations post to a group-ledger partition.

2.11 Deterministic Calculation Engines (E2)

A tamper-evident record of a wrong VAT amount is still wrong, so AE-OS assigns every regulated number to a recomputable engine. A **Calculation Module** is a content-addressed Wasm module in the L3 Wasmtime sandbox: fuel-metered^[29], NaN bit patterns canonicalized (one of Wasm's few nondeterminism sources^[55]), no clock, randomness or I/O, money in fixed-point decimals, every input (dates, rates, master data) passed as a CID. A **Certified Engine** is an external product, such as a payroll or sales-tax service, called as an MCP tool through the L2 gateway with its version pinned and request and response hashes logged. Each run emits a `CalculationExecution` event (engine CID, input, output digests, fuel), which an auditor replays on the sealed input. An agent may pick a scenario or parametrize an estimate such as an accrual, but the amount reaches the kernel only with a receipt or a signed human judgment (I8).

CALCULATION DOMAIN	GOVERNING RULE	ENGINE TYPE	WHY THE RESULT MUST BE REPRODUCIBLE
Tax determination	VAT Directive; local law	Module; Certified Engine	On the invoice and the return; recomputed at audit
Depreciation	IAS 16 ^[56]	Module	Monthly runs must match prior periods to the cent
Inventory valuation	IAS 2 ^[57]	Module	FIFO / weighted-average layers depend on posting order
FX revaluation	IAS 21 ^[53]	Module	Period-end revaluation; the rate-table CID fixes the input
Revenue allocation	IFRS 15 / ASC 606 ^[58, 59]	Module	Stand-alone selling price allocation, re-run on modification
Lease accounting	IFRS 16 ^[60]	Module	Multi-year right-of-use and liability schedules
Payroll gross-to-net	Tax, social-security law	Certified Engine	Withholding is filed with authorities and paid out
MRP / ATP	Planning parameters	Module (fixed tie-breaks)	Same demand and stock give the same plan and promise

Process Layer, Domain Packs and Localization

Most ERP work is procedural and calendar-driven: payment runs, dunning, depreciation, month-end close, payroll cut-off. AE-OS specifies a durable process runtime (E3), versioned Domain Packs (E4), and two cross-cutting frameworks for country rules and for records and privacy.

2.12 Process Layer (E3)

Process Definition. A deterministic state machine (steps, transitions, timers, compensations, Human Task slots) serialized as DAG-CBOR and pinned by CID, promoted through the same signed release path as APOs and policy modules. A running **Process Instance** records the Definition CID it started under; a new version applies only to new instances, so an in-flight payroll run never changes logic mid-cycle. Each instance is bound to an Intent Envelope. Scheduled processes run under a *standing envelope* signed by the process owner, with a budget and an expiry, usually one fiscal year.

Steps and sagas. Every step is a `ProcessStep` Causal Event and passes the L2 → L3 → L4 pipeline like any agent action. Agents fill judgment steps (match exceptions, coding suggestions); calculation steps call E2 engines; posting steps call the Accounting Kernel. Multi-step processes that span partitions, such as an intercompany settlement or a three-way match followed by payment, run as sagas: each **Saga Step** declares a compensating step, and a failure after step *k* runs compensations *k*...1 in reverse order^[52]. In the books, a compensation is a reversal entry in the open period (I9). A Journal Entry itself is never a saga: it is an atomic, balanced document in one legal-entity × ledger partition, booked or rejected as a whole (I7).

Scheduler and business calendars. A **Schedule** binds a Process Definition to a business calendar: fiscal-year variant, posting-period map, factory and bank-holiday calendars per country, payroll periods and bank cut-off times. The close calendar is a dependency graph of tasks (sub-ledger cut-off → accruals → FX revaluation → intercompany matching → consolidation), and the scheduler releases a task when its predecessors have emitted their completion events. Payment runs, dunning, depreciation and payroll are Schedules with their own calendars. A `PeriodTransition` event opens or closes a posting period and is itself a scheduled step.

```
SCHEDULE // P2P.PAYMENT-RUN • COMPANY DE01

{ "process": CID, // p2p/payment-run@3.2.0
  "calendar": "DE-bank+factory/FY2027",
  "rule": "TUE,THU 10:00 Europe/Berlin",
  "on_holiday": "next_business_day",
  "envelope": CID, // standing, owner-signed
  "sla": { "complete_by": "13:30",
           "breach": "escalate:treasury-lead" },
  "degraded": { "policy": "fail_closed" } }
```

Durability and replay. The runtime keeps no hidden state: a Process Instance is the ordered list of its `ProcessStep` events, and recovery after a node failure replays that list against the pinned Definition. Steps are dispatched at least once with an idempotency key derived from (instance CID, step id, attempt), and the kernel rejects a second Journal Entry carrying the same key, so a retry cannot post twice. Timers are scheduled events in the same log, so a timer missed during an outage is visible after restart. Because Definitions are deterministic and pinned, an auditor can replay any instance from its events and reach the same state.

Human Tasks for routine work. Routine approvals (invoice release under a delegation-of-authority table, journal approval, master-data change, close sign-off) are **Human Tasks**: typed work items with an assignee rule, substitution and delegation, a due time and an escalation path. Completion is a signed `PolicyDecision` under the person's SPIFFE ID. A Strategic Exception (§5.1) is reserved for actions outside policy or above an authority ceiling; ordinary approvals stay in worklists (§2.17).

SLAs. Each step declares a target duration and a breach action (re-assign, escalate, or raise a Strategic Exception). SLA timers are scheduler events, so a breach is itself logged.

Degraded modes. Fail-closed (P6) is correct for releasing money and wrong for a dock door or a payroll cut-off. Each Process Definition therefore declares, per step and per dependency, what happens when that dependency is unavailable. The runtime records the mode in force on every affected `ProcessStep`.

STEP • OUTAGE	DECLARED MODE	EVIDENCE AND CATCH-UP
Payment release • policy engine	Fail-closed. Run moves to the next calendar slot; SLA breach raises an exception.	Nothing released, nothing posted.
Goods receipt • policy or kernel partition	Store-and-forward: receipt signed and queued at the site.	Evaluated on recovery, flagged <code>retro</code> ; a failure books a reversal.
Payroll cut-off • central policy service	Evaluate on a local replica of the pinned policy CIDs (same Wasm bytes).	<code>PolicyDecision</code> names replica host and CIDs.
Payroll cut-off • gross-to-net engine	Emergency run: prior net pay for unchanged employees; changed records go to Human Tasks; two signed releases.	Marked <code>provisional</code> ; differences corrected next cycle.
Close • FX revaluation engine	Hold. Period stays open; scheduler re-plans dependent tasks.	Close cockpit shows the blocked path.

RESPONSIBILITIES WITHIN ONE PROCESS STEP

The Process Definition decides which step runs next. An agent proposes the content of a judgment step. A Calculation Module computes any regulated amount and emits a `CalculationExecution`. The Accounting Kernel books the result as a balanced Journal Entry. The Causal State Ledger records each of these as linked events, and a person approves the step when policy or an authority ceiling requires it.

02 ENTERPRISE STACK · DOMAIN PACKS AND LOCALIZATION

2.13 Domain Packs (E4)

A **Domain Pack** is the unit of business functionality: one content-addressed bundle that is versioned, signed and promoted as a whole. Any publisher may build a pack; promotion of a pack version is a Causal Event, and every artifact a pack contains keeps its own CID, so a policy or Calculation Module can be traced to the pack release that shipped it. Industry variants (for example process manufacturing or utilities) extend a core pack without forking it.

PACK	SCOPE	REPRESENTATIVE ENGINES AND PROCESSES
R2R Record-to-Report	General ledger, accruals, period close, intercompany, reconciliations	FX revaluation, allocations; close calendar, IC matching
P2P Procure-to-Pay	Requisition, PO, goods receipt, invoice, payables, payment	3-way match, tax determination; payment run
O2C Order-to-Cash	Pricing, order, delivery, billing, receivables, collections	Pricing, revenue allocation; dunning, cash application
Plan-to-Produce	Demand and supply planning, production orders, inventory, costing	MRP/ATP solvers, product costing; production order
H2R Hire-to-Retire	Workforce master data, time, payroll, benefits	Gross-to-net; payroll cycle, off-cycle run
A2R Acquire-to-Retire	Fixed-asset acquisition, capitalization, depreciation, disposal	Depreciation per ledger; depreciation run, retirement
Treasury	Cash position, bank communication, liquidity, FX and debt	Valuation; bank statement import, cash pooling
Tax	Indirect and direct tax, e-invoicing, returns, Pillar Two data	Tax determination, GloBE ETR; return filing
Consolidation	Group ledger, eliminations, currency translation, statements	Translation, eliminations; group close
ESG	Sustainability data collection, metrics, CSRD reporting	Emission factors; disclosure assembly

TABLE 2.13 – Core Domain Packs. Localization Packs (§2.14) supply country rules to every pack.

```
MANIFEST // DOMAIN-PACK

{ "pack": "p2p",
  "version": "3.2.0",
  "publisher": SPIFFE-ID,
  "ontology": CID,
  "processes": [CID, ...],
  "calc_modules": [CID, ...],
  "policy_pack": CID,
  "mcp_servers": [CID, ...],
  "apos": [CID, ...],
  "requires": {
    "kernel": ">=1.4",
    "r2r": "A2" },
  "tests": CID,
  "sig": "ed25519:..." }
// pack CID = multihash of
// the canonical DAG-CBOR
```

PACK LIFECYCLE

- The pack CID covers all six component lists; changing any one yields a new pack version.
- A tenant activates a version only after its test vectors pass against the tenant's ledger configuration; activation is a signed Causal Event.
- Every pack is signed under its publisher's SPIFFE ID and passes the same review gates.
- `requires` pins compatible kernel and pack versions; activation fails on a conflict.

2.14 Localization and Regulatory Framework

Country and regime rules change on fixed effective dates, and a multinational runs dozens of them at once. AE-OS separates them from Domain Packs as **Localization Packs**, one per jurisdiction × regime, each with a named owner, an effective-from/effective-to range and a test-vector suite, promoted like policy CIDs. A Domain Pack calls the active Localization Pack for the posting date; two versions can be valid for the same company code on either side of a change date, and the posting date selects one.

COA TEMPLATE	Statutory chart and mapping to the group chart; local GAAP ledger	AUTHORITY CONNECTORS	Clearance platforms, Peppol access points, filing portals as MCP servers; acknowledgements stored as signed <code>ExternalInput</code> events
TAX MODULES	Determination, rates, reverse charge, withholding as E2 Calculation Modules	STATUTORY GENERATORS	SAF-T, VAT returns, GloBE return, iXBRL from projections
DOCUMENT FORMATS	EN 16931 / UBL, national e-invoice syntaxes, QR or signature rules	NUMBERING, RETENTION	Gapless legal number ranges; retention classes (§2.15)

Regulatory change. Each Localization Pack owner tracks the regimes in Table 2.14 and publishes a new pack version for every change, with its effective date. Test vectors come from authority specifications and published samples where they exist, such as validation rules and example files, and from the owner's own cases otherwise. A new version runs in shadow against live transactions before its effective date: documents and returns are generated by both versions, differences are reported in the close cockpit, and only the version valid on the posting date produces the legal document. Authority-side certification, where a regime requires it (certified invoicing software, accredited e-invoicing platforms), is held by the pack publisher or a partner and recorded as an attribute of the pack version.

02 ENTERPRISE STACK • REGULATORY CALENDAR AND RECORDS

REGIME	EFFECTIVE	REQUIREMENT	LOCALIZATION PACK ARTIFACT
EU ViDA (Dir. 2025/516)	1 Jul 2030; national systems aligned by 1 Jan 2035	Intra-EU B2B e-invoicing to EN 16931 and EU digital reporting become mandatory; member states may introduce domestic e-invoicing without prior Commission approval ^{[61][62]}	EN 16931 generator, transaction-level reporting feed
France e-invoicing	1 Sep 2026; SMEs issue from 1 Sep 2027	All in-scope businesses receive e-invoices through an approved platform (PA); large companies and ETIs also issue and e-report ^[63]	PA connector, e-reporting, AFNOR XP Z12-012/013/014 formats
Belgium B2B	1 Jan 2026; e-reporting planned 1 Jan 2028	Domestic B2B invoices issued and received as structured EN 16931 documents over Peppol (BIS 3.0, UBL 2.1); a five-corner reporting draft law was approved by cabinet in Jul 2026 ^[64]	Peppol access-point connector
Germany B2B	Receive 2025; issue 2027 (>€800k turnover), 2028 all	All businesses must accept e-invoices; issuance phases in by prior-year turnover; EDI remains possible if a compliant record can be extracted ^[65]	XRechnung / ZUGFeRD formats, EDI mapping
SAF-T (OECD)	PT 2009 ... BG 2026, BE Oct 2026	Standard audit file; mandatory monthly in PL (replaced the VAT return) and RO, replaced the VAT return in NO, on request elsewhere ^[66]	Per-country SAF-T schema generator from ledger projections
Pillar Two GIR	First filings 30 Jun 2026 (mainly FY2024)	GloBE Information Return for groups with revenue ≥ €750M, in nearly 40 jurisdictions; reported problems include CbCR safe-harbour data, TINs and failed validation checks ^[67]	GloBE ETR module, XML return, CbCR reconciliation
IFRS 18	Periods from 1 Jan 2027	Replaces IAS 1; required operating-profit and profit-before-financing-and-income-taxes subtotals; disclosure of management-defined performance measures ^[68]	Statement model and MPM reconciliation in E5
CSRD after Omnibus I (Dir. 2026/470)	In force 18 Mar 2026; FY2027 (EU), FY2028 (non-EU parents)	Scope: >1,000 employees and >€450M turnover; non-EU groups with >€450M EU turnover and an EU subsidiary or branch >€200M; transposition by 19 Mar 2027 ^{[69][70]}	Scope test, ESG pack disclosure set

TABLE 2.14 – Regulatory calendar, status as of Oct 2026. Each row becomes a Localization Pack version with an effective date.

2.15 Records and Privacy Lifecycle

Retention classes. Every Causal Event kind and every sealed payload carries a **Retention Class** set by the Localization Pack: minimum period (commercial and tax law), maximum period (data protection), trigger (end of fiscal year, end of contract, employee exit) and disposal action. A record past its minimum and not on hold is disposed of by the scheduler; disposal is a logged event.

Archive tiers. Hot (open and prior fiscal year, in the actor grid), warm (projections and sealed payloads in object storage) and cold (immutable archive). Checkpoints and inclusion paths move with the data, so an archived entry still verifies against the signed root it was logged under.

Legal hold. A **Legal Hold** is a signed event scoped by query (entity, party, period, matter). While it is active, disposal and key shredding for matching records are blocked by an L3 policy, and the hold's release is also logged.

Retention against erasure. Where commercial or tax law requires a record to be kept, the erasure request is recorded but deferred: retention needed to comply with a legal obligation is an exception to the right to erasure^[71]. Until the retention period ends, an L3 read policy restricts the subject's sealed fields to audit and tax roles, and the scheduler queues the `KeyShred` for the expiry date.

Disposal of business records. Sealed payloads without personal data are encrypted under one key per retention class × fiscal year × partition. When that class expires and no hold applies, shredding a single key disposes of a whole year of payloads, while their digests and checkpoints remain in the log.

Erasure by crypto-shredding. The ledger stays append-only, so personal data never enters a leaf in clear. Personal fields are sealed under a per-subject key k_s held in KMS, and the leaf stores only a keyed commitment:

$$\begin{aligned} c &= \text{HMAC-SHA256}(k_s, \text{canon}(\text{pd})) \\ \text{payload} &= \text{AEAD}(k_s, \text{pd}), \text{ stored off-ledger} \end{aligned}$$

(Eq. 2.15)

An erasure request that no retention obligation overrides^[71] ends in a `KeyShred` event: KMS zeroizes k_s and the keys below it. The ciphertext becomes unreadable, c can no longer be linked to the person, and every Merkle path still verifies because no leaf byte changed. This follows the EDPB's final blockchain guidelines: personal data off-ledger, salted or keyed hashes and commitments on-ledger, and deletion of the key or salt^[72].

Limits. The EDPB treats encrypted data as personal data until the key is gone and warns that ciphers age^[72]; NIST requires every copy of the key, including backups and escrow, to be sanitized for cryptographic erase to count^[73]. AE-OS therefore never exports k_s , and `KeyShred` records two times: zeroization, and expiry of the last KMS backup that held the key.

Validation and Assurance

AE-OS takes an agent-native enterprise as its architectural starting point. Existing systems provide environments in which that architecture can be tested against real obligations, records and operating constraints.

2.16 Validating the architecture alongside existing systems

Coexistence bridges, shown here with SAP as the worked example, allow proposed components to run in shadow mode, compare outcomes and establish evidence before taking responsibility for production work. Compatibility with existing systems is therefore an adoption requirement. The earliest step needs no migration: signed receipts can be recorded for agent tool calls made against the existing system. The adoption path follows the strangler-fig pattern, in which the new system takes over behavior piece by piece until the legacy system can be retired^[74]. The **Coexistence Bridge** is the set of components that makes each step reversible and evidenced.

SAP MCP SERVER	SAP functions (create PO, post document, read open items) exposed as MCP tools behind the L2 gateway. Calls carry JWT-SVIDs, pass L3 policy and get L4 receipts; the SAP credential stays inside the server workload. The <code>erp::create_po</code> call in §4.4 is this pattern.	SYSTEM-OF-RECORD FLAG	One signed flag per process × company code: SAP, SHADOW or AE-OS. In SHADOW the kernel posts only to a non-legal shadow ledger; legal numbering and filings come from whichever system holds the flag.
CDC INGEST	Change data capture from SAP documents and master data into the Semantic Data Plane. Each change becomes an <code>ExternalInput</code> signed by the bridge workload, carrying the SAP document key and the extract position.	REVERSE REPLICATION	While the close still runs in SAP, AE-OS-owned processes send summarized postings back to the SAP general ledger through the MCP server, each citing its AE-OS Journal Entry CID.
		RECONCILIATION PROJECTION	Balance comparison per entity × account × currency × period (Eq. 2.16).

WAVE	SCOPE MOVED	RECORD AFTER WAVE	EXIT CRITERIA (TARGETS)
0 • Reporting	CDC of SAP ledgers and sub-ledgers into E5 projections; agents read only	SAP	Projected trial balance equals SAP for consecutive closes in every company code; lines trace to SAP document keys
1 • P2P	Supplier invoice to payment for pilot company codes; AP booked in the kernel, summarized to SAP	AE-OS for P2P	Two closes in SHADOW with no unexplained difference above tolerance; payment files accepted by the banks; SoD ruleset approved
2 • O2C	Billing, receivables, collections, cash application	AE-OS for O2C	As wave 1, plus e-invoice acknowledgements received from the authority or platform; receivables ageing reconciles
3 • Close	General ledger, accruals, intercompany, consolidation; SAP kept as a read-only archive	AE-OS for R2R	One quarter-end close completed in AE-OS; external auditor walkthrough of key controls; intercompany eliminations agree
4 • Statutory	Tax returns, SAF-T, statutory statements from Localization Packs	AE-OS	One full filing cycle accepted by each authority; SAP decommissioning plan with retained archive access

TABLE 2.16 – Strangler sequence, with SAP as the existing system. Hire-to-Retire, Plan-to-Produce and Acquire-to-Retire move as separate waves after wave 3, ordered by the business.

Genesis events. At each cutover the bridge takes a final extract and books a `GenesisEvent`: opening balances per account in the currency triple, open items with their original SAP document numbers, fixed assets with accumulated depreciation per ledger, and history loaded as reference-only data. The opening Journal Entry balances against a migration clearing account that must net to zero, so I7 holds from the first entry. The event stores the extract file's SHA-256, so an auditor can re-hash the archive and link pre-cutover SAP history to the first AE-OS checkpoint.

Parallel run. During `SHADOW`, both systems book the same business events. The Reconciliation Projection computes, for each entity e , account a , currency c and period p :

$$\Delta(e,a,c,p) = B_{\text{AE-OS}} - B_{\text{SAP}} ; |\Delta| \leq \tau_a \text{ or explained}$$

(Eq. 2.16)

An explanation is a signed event that classifies the difference (timing, mapping, rounding, SAP defect) and cites the documents involved. The controller sets τ_a per account class.

```
EVENT // GENESISEVENT
{
  "kind": "GenesisEvent",
  "entity": "DE01",
  "ledger": "0L",
  "open_period": "2027-07",
  "source": {
    "system": "SAP S/4HANA",
    "extract_sha256": "9f2c...",
    "rows": 1843220,
    "control_totals": CID },
  "journal_entry": CID,
  "open_items": CID,
  "signed_by": [
    controller, migration-AP0 ]
}
```

02 ENTERPRISE STACK • VALIDATION AND ASSURANCE

Cutover and rollback. Cutover is itself a Process Definition: lock the SAP posting period for the scope, take the delta extract, book the delta genesis, reconcile, flip the system-of-record flag (signed by the process owner and the controller), and open the AE-OS period. Within a declared rollback window the flag can be flipped back: AE-OS documents posted since cutover are re-posted to SAP through the MCP server, each citing its AE-OS CID, and reversed in AE-OS in the open period. The window closes at the first statutory filing produced by AE-OS for that scope.

2.17 Assurance

Merkle receipts prove that logged events have not been altered; external auditors test more. Under AS 2201 the auditor must understand how IT affects the flow of transactions, and may treat an automated control as lower risk only when the IT general controls it depends on are effective^[75]. When a service organization processes the company's transactions, the auditor includes its activities and relies on a service auditor's report covering tests of operating effectiveness; a report without those tests gives no evidence of operating effectiveness (AS 2201 B18–B20)^[75]. Under ISAE 3402, a Type 1 report covers the fair presentation of the system description and the suitability of control design, and a Type 2 report adds whether the controls operated effectively throughout a specified period^[76]. An AE-OS tenant used as a book of record therefore needs a SOC 1 / ISAE 3402 report, and AE-OS evidence has to be mapped to the domains auditors test.

CONTROL DOMAIN	AE-OS EVIDENCE	WHAT THE AUDITOR STILL TESTS
Access to programs and data	SPIFFE IDs and SVID issuance logs; APO lifecycle events; capability tokens bound to one Intent Envelope; human actions signed with step-up authentication	Joiner, mover and leaver processing at the identity provider; privileged access to SPIRE, KMS and infrastructure; periodic access reviews
Change management	Promotion history of every policy, Process Definition, Calculation Module and pack CID, with signed approvals and passed test vectors; reproducible builds	That promotion gates cannot be bypassed; emergency change handling; infrastructure and platform changes outside the CID path
Operations	Scheduler and SLA events; degraded-mode flags on <code>ProcessStep</code> ; checkpoint and witness continuity; restore tests logged as events	Incident and problem management; backup and restore effectiveness; monitoring of failed jobs
Data migration	<code>GenesisEvent</code> with extract hash and control totals; Reconciliation Projection and signed explanations	Completeness and accuracy of the SAP extract and of account and field mappings
IPE accuracy and completeness	Report Snapshot = projection definition CID + checkpoint root + output digest; any report can be re-run from the checkpoint and compared	Completeness of inputs that enter outside the gateway (bank statements, EDI, e-invoices); report parameters; agreement of the report to the ledger
Automated controls	A <code>PolicyDecision</code> for every action; kernel invariants I7–I9 enforced as mandatory policy modules	Whether each policy implements the control as designed; configured tolerances and thresholds
Agent judgment	<code>ReasoningTrace</code> with model and version; <code>CalculationExecution</code> for every amount (I8); Human Task approvals	Management review controls over agent proposals; model risk management

TABLE 2.17 – Control domains (access, change management and operations are the ITGC domains) mapped to AE-OS evidence.

Completeness of inputs. Invariant I1 guarantees that every action routed through the gateway is logged. Bank statements, EDI messages and e-invoices originate outside it; each arrives as an `ExternalInput` that stores the raw file's hash and the source's signature where the channel provides one, or the bridge's signature at ingest otherwise; an L3 policy checks source sequence numbers (statement, interchange control) for gaps.

SOC 1 path. AE-OS is designed for SOC 1 Type I / ISAE 3402 Type 1 readiness: system description, control objectives, the cloud provider carved out as a subservice organization, and complementary user entity controls for what the tenant owns (policy content, tolerance values, human access approvals). A Type II / Type 2 report requires an observation period. Receipts and replay let auditors reperform a control on any sampled item, in addition to the service auditor's report.

Insight Layer (E5). A **Projection** is a CID-pinned definition plus a checkpoint watermark. Financial statements are projections over the kernel, with an IFRS 18 statement model (§2.14). **Metric Definitions** are versioned by CID, so a constraint such as "gross margin ≥ 24%" refers to one formula. **Plan Versions** are non-legal ledgers on the kernel. A **Report Snapshot** pins the definition and the checkpoint root it was computed at, so "as reported" stays provable after later postings. Projections are exported to a lakehouse by CDC.

Experience Layer (E6). Executives state intent in the intent console and handle Strategic Exceptions. Back-office staff work in **Work Surfaces** generated from the ontology and Process Definitions: worklists of Human Tasks, the close cockpit, reconciliation and master-data maintenance. The design goal is no routine data entry; screens remain for review, approval and investigation. Auditors use the **Verifier Portal**: from a statement line they drill to Journal Entries and their causes, and verify each receipt locally against a published checkpoint; PBC requests arrive as Human Tasks. Supplier, customer and employee portals and a signed CLI complete the layer.

SCOPE OF THIS MAPPING

Table 2.17 is a design target for a service auditor to test; it reports no audit result. Regulatory acceptance, such as tax-authority certification of invoicing software or accreditation as an e-invoicing platform, is granted per jurisdiction and sits outside this mapping (§2.14). Appendix C sets out the research and implementation scope item by item against an SAP estate.

02

IDENTITY OF PEOPLE, ORGANIZATIONS AND AGENTS

Identity of People, Organizations and Agents

The Identity Plane gives every workload, including every agent, a verifiable identity. Agent-native work also involves the people who delegate authority and the organizations that trade with each other. Their identities are moving to digital wallets and verifiable credentials, and this section proposes how AE-OS would accept them.

2.18 Three kinds of principal

PRINCIPAL	IDENTITY SOURCE	CREDENTIAL	USED IN AE-OS FOR
Workload or agent	SPIFFE trust domain operated by the enterprise	X.509-SVID or JWT-SVID, renewed every 15 minutes	Every call through the gateway (§2.6)
Person employee, approver, customer	Enterprise identity provider; a European Digital Identity Wallet or another national wallet	OpenID Connect token; person identification data and attestations presented with OpenID4VP; qualified electronic signature	Signing Intent Envelopes, approving exceptions, step-up authentication
Organization supplier, customer, bank	Business registers and LEI issuers; a European Business Wallet once adopted	vLEI and role credentials; register-based attestations; qualified electronic seal	Supplier and customer onboarding, e-invoices, payment counterparties, mandates

TABLE 2.18 – Principals and their credentials. Only the first row is specified in §2.6–2.8; rows two and three are proposals.

SPIFFE identifies workloads: a SPIFFE ID uniquely identifies a workload, and an SVID is the document with which the workload proves it^[77]. AE-OS keeps that separation. A person never receives a SPIFFE ID, and an agent never holds a person's credential. The two are joined by a signed delegation record, described below.

Wallets for people

The revised eIDAS Regulation, (EU) 2024/1183, entered into force on 20 May 2024. Article 5a requires each Member State to provide at least one European Digital Identity Wallet within 24 months of its implementing acts entering into force^[78]. The first implementing acts took effect in December 2024^[79], and the Commission expects wallets to be available by the end of 2026^[80]. The wallet holds person identification data and attestations of attributes, and it can create qualified electronic signatures and seals^[78].

Article 5f(2) matters most for enterprise systems. Private relying parties that must use strong user authentication by law or contract, including banking and financial services, have to accept the wallet when the user asks, within 36 months of the implementing acts; micro and small enterprises are exempt^[78]. On the dates above that falls around the end of 2027.

Credentials for organizations

The Commission proposed a regulation on European Business Wallets on 19 November 2025. It covers economic operators and public sector bodies, identified by the EU's company identifier (EUID) or a similar identifier, with support for representation rights, mandates and the exchange of signed documents and attestations^[81,82]. The proposal is still going through the legislative procedure^[83].

The verifiable LEI lets a counterparty verify by computer the identity, authority and role of people acting for a legal entity^[84]. It is standardized as ISO 17442-3:2024^[85], and its governance framework defines role credentials for official representatives^[86].

Formats. The W3C Verifiable Credentials Data Model 2.0 became a Recommendation in May 2025^[87]; selective-disclosure JWTs are RFC 9901^[88]; OpenID4VP 1.0 and OpenID4VCI 1.0 were finalized in 2025^[89,90].

02 IDENTITY OF PEOPLE, ORGANIZATIONS AND AGENTS

Binding a person's authority to an agent

A delegation record links the person who holds authority to the agent that uses it. The proposed sequence:

- 1. **Verify the person.** The person authenticates with a wallet or the enterprise identity provider. AE-OS checks the presentation against a list of trusted issuers, checks revocation and confirms the credential is bound to the holder.
- 2. **Sign the intent.** The person signs the Intent Envelope (§5.2). Intents above a set value, or with legal effect, require a qualified electronic signature; for an organization, a qualified seal or a signature by a holder of an official-role credential.
- 3. **Issue a capability.** The gateway issues a capability token that names the person as principal and the agent's SPIFFE ID as the acting party, the same structure as the actor claim of OAuth Token Exchange, which records that delegation has occurred and who acts^[91]. Attenuation follows §4.3.
- 4. **Record the chain.** Each Causal Receipt carries the digest of the person's credential presentation, the intent, the agent's SVID and the action, so an auditor can verify the whole chain offline.

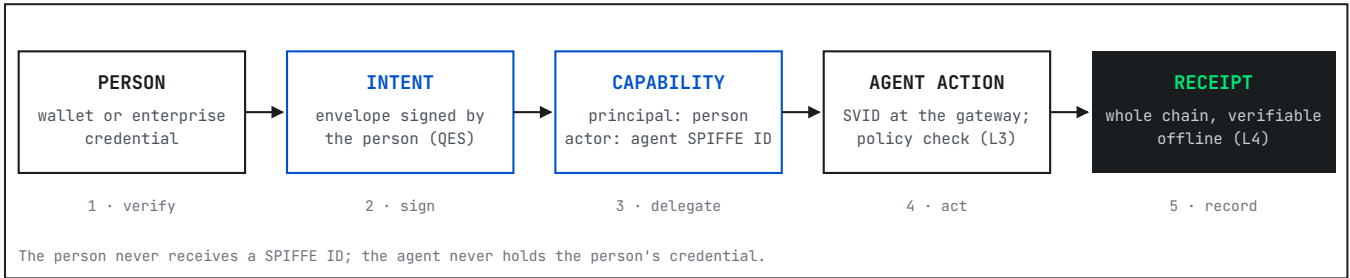


FIG. 2.4 – Proposed delegation chain from a person's credential to an agent's action. The capability token follows the actor structure of OAuth Token Exchange.

Standards bodies are working on the same problem. The IETF OAuth working group is drafting Transaction Tokens, which carry the calling user's context through a chain of services^[92], and the OpenID Foundation has published a paper on identity management for agentic AI^[93]. AE-OS would adopt these where they meet the requirements of §1.4.

2.19 Payments and banking

Banks are already required to check payees. Under the Instant Payments Regulation, a payer's payment service provider must offer verification of the payee, which applied in the euro area from 9 October 2025^[94]. The provisional agreement on the Payment Services Regulation of November 2025 extends the name check to all credit transfers and keeps strong customer authentication^[95,96]. When an agent prepares a payment, the bank and the auditor ask the same questions. TABLE 2.19 lists the evidence the design would give them.

QUESTION	EVIDENCE AE-OS WOULD PRESENT
Who authorized the payment?	The person's signed intent, verified against a wallet or enterprise credential, with its limits
Was the agent allowed to act?	The capability chain from the person to the agent's SPIFFE ID, and the PolicyDecision for the payment
Is the payee who they claim to be?	The supplier's organization credential checked at onboarding, and the verification-of-payee result stored as an ExternalInput
Can this be proven later?	The Causal Receipt for each step, verifiable against a published checkpoint (§3.6)

TABLE 2.19 – Evidence for an agent-prepared payment. A proposal; no bank has reviewed it.

GLEIF has described how the vLEI could give agents that initiate payments an organizational credential that counterparties and payment providers can read by machine^[97]. How strong customer authentication applies when an agent submits a payment under a person's standing mandate is unresolved, and it is open question Q15.

■ SCOPE BEYOND THE EU

This section uses eIDAS 2.0 as the worked example because its legal requirements and dates are fixed. National digital identity programmes and digital public infrastructure elsewhere issue similar credentials under their own trust frameworks. AE-OS treats each wallet ecosystem as a list of trusted issuers and credential types, so the delegation record works with any of them. Identity and trust is one of the open research problems (TABLE 6.1).

03 CAUSAL ACCOUNTABILITY CORE

Merkle-DAG Provenance Trees

When agents move money and goods, who decided what, under which rule and on whose authority must be answerable from tamper-evident records. AE-OS is designed to commit every causal step to an append-only Merkle log, so that a verifier holding a signed checkpoint can detect any later edit.

3.1 Limits of database audit trails

Classical ERP audit trails are rows in the database they describe: a privileged operator, a compromised service account or a faulty migration can rewrite both a transaction and its log. With agents, one released payment is the product of a human goal, a policy evaluation, a model's output and several tool calls across services and identities. Regulation expects such chains to be reconstructable: the EU AI Act requires high-risk AI systems to allow "automatic recording of events (logs) over the lifetime of the system" (Art. 12), kept for at least six months (Art. 19)^[4]; SOX §404 requires management to assess, and auditors to attest to, the effectiveness of internal control over financial reporting^[98].

AE-OS makes accountability a **data-structure property**. It reuses two proven constructions: Merkle's hash tree (1979 patent filing, CRYPTO '87^[99,100]) and the append-only log of Certificate Transparency (CT), which Chrome has required for publicly trusted TLS certificates issued after 30 April 2018^[101,5]. CT shows a log can be run by an *untrusted* party, because each client verifies compact proofs itself instead of relying on the operator.

- **I1 · Log-before-act.** No tool call runs until its Policy Guardrail Node is appended (a gateway-enforced platform control that auditors test, §2.17).
- **I2 · Content addressing.** A node's name is the hash of its bytes.
- **I3 · Append-only.** Rewrites are provable from two checkpoints.
- **I4 · Root of authority.** Chains end in a signed Intent Envelope.
- **I5 · Digests on-ledger.** Payloads live off-ledger, encrypted; the ledger holds digests or keyed commitments (§2.15).
- **I6 · Signed leaves.** The actor signs the event CID with its X.509-SVID key (ECDSA P-256^[102]); humans sign with WebAuthn (§3.5).
- **I7-I9 · Books.** Balanced entries, calculated amounts, no posting into closed periods (§2.10).

3.2 The Causal Event model

Each step of an agent's work emits a **Causal Event**. Events form a DAG whose edges (*parent* → *child*: "produced in response to") are content identifiers (CIDs) embedded in the child, so a parent cannot change without changing every descendant's identity: the Merkle-DAG property^[103]. Every event is also appended to one totally ordered Merkle log (§3.5). The DAG records causation; the log records inclusion and order. Each append yields a Causal Receipt (§3.6).

```
FILE // SPEC/LEDGER/CAUSAL_EVENT.RS (SKETCH)
```

```
struct CausalEvent {           // DAG-CBOR
  v:      u8,                  // schema = 1
  kind:   EventKind,           // Table 3.1
  tenant: String,              // "acme.ae-os"
  actor:  SpiffeId,            // agent/ap/7f3c
  intent: Cid,                 // chain root
  parents: Vec<Cid>,           // causes, sorted
  hlc:    u64,                 // hybrid clock
  body:   Body,                // digests only
  sealed: Option<Cid>,         // encrypted blob
  svid:   Cid,                 // SVID chain
}

struct LogEntry {              // stored with leaf
  seq:    u64,                 // leaf index i
  event:  Cid,                 // CIDv1
  signer: SpiffeId,
  sig:    Vec<u8>,             // ECDSA P-256(cid)
}

// cid(e) = 01 71 12 20 || SHA-256(cbor(e))
// leaf(i) = SHA-256(00 || cid(e_i))
```

KIND	BODY (COMMITTED DIGESTS)
IntentEnvelope Human Intent Envelope	objective hash, constraints, budget, authority ceiling, expiry, signature
PolicyDecision Policy Guardrail Node	policy CID, input digest, verdict (ALLOW / DENY / ESCALATE), fuel used
ReasoningTrace Agent Reasoning Trace	model version label (+ provider attestation), context CIDs, rationale digest; full text sealed off-ledger
ToolExecution Tool Execution Node	MCP tool, request/response hashes, status, side-effect refs

TABLE 3.1 – Core leaf node types (AE-OS spec). §2.9–§2.15 add *JournalEntry*, *CalculationExecution*, *PeriodTransition*, *ProcessStep*, *ExternalInput*, *GenesisEvent* and *KeyShred*.

What a ReasoningTrace commits to. The rationale the agent recorded, not how the model computed its output. LLM outputs are not reproducible from their inputs (sampling, batching, floating-point order, silent model updates), and a recorded rationale need not reflect the model's internal process. Only policy evaluation (§3.8) and deterministic calculations (*CalculationExecution*, §2.11) are replayable; the agent's proposal is untrusted input that policy and, above the ceiling, a human must accept. For a hosted model, the trace holds the vendor's version label and attestation, not a weight digest.

3.3 Hashing a causal chain into a Merkle root

FIG. 3.1 shows the smallest complete causal chain: an intent, the policy check it triggered, the agent's reasoning and the resulting tool call. Each event is hashed into a leaf; pairs of leaves are hashed into interior nodes; the process repeats until a single 32-byte **Merkle root** remains. The root is the only value AE-OS needs to sign: it commits to every byte of every leaf, in order.

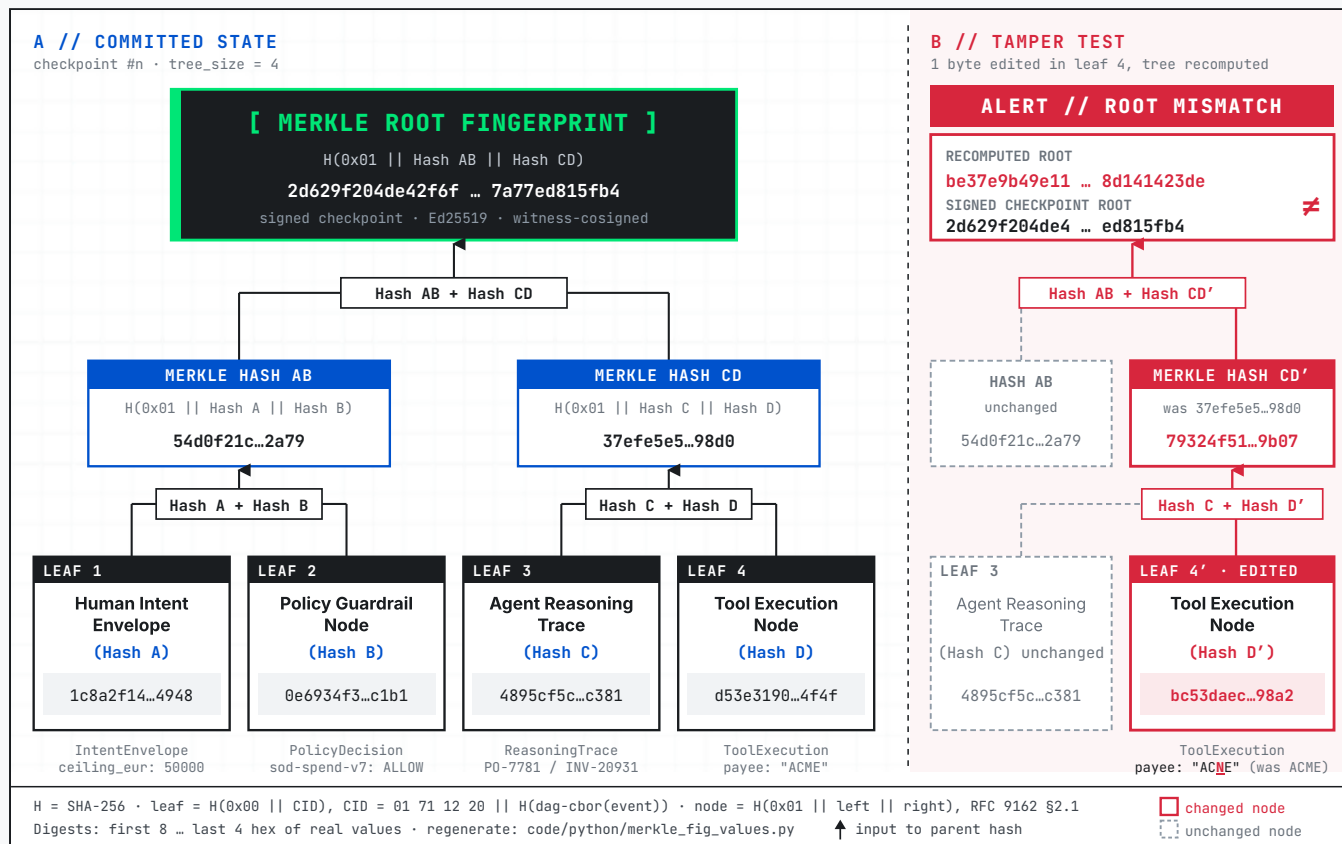


FIG. 3.1 – Cryptographic hash validation tree over one causal chain. **Panel A**: four Causal Events (leaves) are hashed and combined pairwise (Hash A + Hash B → Merkle Hash AB, Hash C + Hash D → Merkle Hash CD) into the root that the log signs. **Panel B**: changing one letter of Leaf 4 (payee ACME → ACME) changes Hash D, then Hash CD, then the root; the recomputed root no longer matches the signed checkpoint. All digests are real SHA-256 values, generated by code/python/merkle_fig_values.py from the illustrative payloads in that script. Leaves are computed as in §3.2 and Appendix A: Leaf = H(0x00 || CID), with the CID taken over the event's DAG-CBOR bytes.

Tamper detection and localization

SHA-256 (FIPS 180-4^[104]) maps any input to 256 bits such that no efficient method is known to find two inputs with the same output; NIST rates its collision resistance at 128 bits of security^[105]. In Panel B the tampered byte changes Hash D completely (d53e3190... → bc53daec...), and because Hash D is an input to Hash CD, and Hash CD to the root, the change propagates upward. The sibling subtree (Hash AB) is untouched, which makes localization cheap: the verifier learns *which* half of the tree diverged and can bisect to the altered leaf in $\log_2 n$ steps.

Appendix B, Theorem 1, gives the reduction: making the root match again requires finding a SHA-256 collision.

Leaf and interior hashes use different one-byte prefixes (0x00, 0x01). RFC 6962 notes that this domain separation "is required to give second preimage resistance"^[101]: without it, an attacker could present an interior node's 64-byte input as if it were a leaf.

Causal DAG links and log order

The figure shows the *log* tree, whose shape depends only on arrival order. The *causal* DAG is carried inside the events: Leaf 2's `parents` field holds Leaf 1's CID, Leaf 4's holds Leaf 2's and Leaf 3's. Both are hash-linked, so tampering is detected by the root and by any descendant whose stored parent CID no longer resolves.

HASH	SHA-256, 32 B digest
LEAF	H(0x00 cid(event)), 36-byte CID (§3.2)
NODE	H(0x01 left right); split at largest power of two < n (RFC 9162) ^[32]
ROOT SIGNED BY	Ed25519 log key (HSM) + witnesses (§3.5)
TAMPER DETECTED BY	any verifier holding one signed root
NOT DETECTED	events never logged; prevented by I1, a platform control (§2.17)

Content-Addressable Ledger Mechanics

3.4 Content identifiers over canonical DAG-CBOR

A hash is a stable name only if the bytes are stable. AE-OS fixes one canonical encoding, **DAG-CBOR**, the IPLD profile of CBOR: string map keys sorted length-then-bytewise, shortest integer and length encodings, 64-bit floats only, no NaN or infinities, and tag 42 (a binary CID) as the only tag^[186]. This is close to RFC 8949 §4.2 deterministic CBOR, which sorts keys purely bytewise^[187]. AE-OS rejects non-conforming input on decode instead of re-canonicalizing it, so every party derives the same identity for a given event.

The identifier is a **CIDv1**: version, codec and a self-describing **multihash** (function code, length, digest)^[183,188]. With `dag-cbor = 0x71` and `sha2-256 = 0x12`^[189] every event ID is 36 bytes (`01 71 12 20` followed by the 32-byte digest; in text form, multibase base32 with prefix `b`, so CIDs render as `bafyre...`), and a future hash migration can proceed incrementally: old and new CIDs coexist, distinguished by prefix.

3.5 Append-only log, checkpoints and keys

Each tenant has one logical log, sharded for ingest but sequenced into one tree. An append hashes the leaf and updates the *compact range* ($\leq \lceil \log_2 n \rceil$ cached subtree roots on the right edge): $O(\log n)$ hashes, no existing node rewritten; completed interior nodes are immutable and stored as fixed-size tiles. Every 1 s or 10 000 entries (default target) the sequencer emits a **checkpoint** in C2SP signed-note form: origin, decimal tree size, base64 RFC 6962 root, signed with the log's Ed25519 key^[118,111]. RFC 9162 calls the equivalent structure a Signed Tree Head (`log_id`, `timestamp`, `tree_size`, `root_hash`)^[32].

Receipt timing. After the quorum commit, and within the §3.9 budget, an append returns a *signed promise*: sequence number, event CID and log signature, committing the log to include the entry in the next checkpoint. This mirrors CT's Signed Certificate Timestamp, a promise to merge an entry within a fixed maximum merge delay^[181]. The Causal Receipt, an inclusion proof against a cosigned checkpoint, follows asynchronously after at most one checkpoint interval plus cosigning time. Steps that need finality, such as payment release or an auditor export, wait for it; a promise not redeemed by its deadline is itself evidence of log misbehavior.

Witnesses. A signature alone does not stop an operator from showing different auditors different trees (a *split view*). AE-OS therefore requires witness cosigning for every deployment whose receipts serve as audit evidence; it is optional only in development. Witnesses run outside the operator (auditors, regulators, industry bodies) verify a consistency proof from the last checkpoint they cosigned and return a timestamped cosignature^[112]; a checkpoint is final once k of n witnesses have cosigned. Checkpoints are also anchored periodically in an external log such as Sigstore's Rekor^[33]. Witnessing costs one request per witness per checkpoint interval, independent of event volume.

KEY	HOLDER · ALGORITHM	ROTATION	ON COMPROMISE
Trust-domain root CA	offline HSM, m-of-n custody · ECDSA P-384	key ceremony; new root cross-signed	revoke; re-issue intermediates; re-attest workloads
SPIRE intermediate CA	HSM or KMS through a SPIRE KeyManager plugin (e.g. <code>aws_kms</code> , <code>kmip</code>) · ECDSA P-256 ^[47]	<code>ca_ttl</code> (SPIRE default 24 h)	remove from trust bundle; its SVIDs expire within 15 min
Workload SVID keys (sign leaves)	process memory, never exported · ECDSA P-256 ^[182]	every SVID (TTL 15 min)	expire; leaves checkpointed after the reported compromise are flagged
Log checkpoint key	HSM, dual control · Ed25519, approved in FIPS 186-5 ^[113] ; HSM support varies by vendor	key ID in every checkpoint; rotation is a checkpointed event signed by old and new keys	witnesses reject any fork; resume from the last witnessed checkpoint under a new key
Witness keys	each witness · Ed25519 or ML-DSA-44 ^[114]	witness policy	remove from the k -of- n set
Human approval credential	user's authenticator · WebAuthn / FIDO2 ^[115]	per device enrolment	revoke at the identity provider; later approvals rejected

TABLE 3.2 – Key hierarchy (AE-OS specification). Leaf signatures use workload keys; only checkpoints use the log key.

Long-term verification and post-quantum migration

A leaf signature must remain checkable for the statutory retention period (§2.15), long after its 15-minute SVID has expired. Each event carries the CID of its signer's certificate chain, and archives keep the trust bundles. A verifier accepts the signature if that chain was valid when the event first appeared under a witness-cosigned checkpoint: the cosignature timestamps^[114] serve as trusted time, and RFC 3161 tokens^[116] on periodic checkpoints add an independent time source. A stolen retired key cannot backdate an event, because any new leaf lands beyond the current tree size. **Post-quantum.** Merkle proofs rely only on SHA-256. Signature algorithms are versioned by key ID: checkpoints can move to a hybrid Ed25519 + ML-DSA signature (FIPS 204^[117]), and C2SP already specifies ML-DSA-44 witness cosignatures. Consistency proofs chain old checkpoints to post-quantum-signed successors, so earlier evidence stays verifiable.

3.6 Inclusion and consistency proofs

```
FILE // SPEC/LEDGER/VERIFY.RS (RFC 9162 §2.1.3.2)

fn verify_inclusion(i, n, leaf, path, root) -> bool {
  if i >= n { return false }
  let (mut f, mut s) = (i, n - 1);
  let mut r = leaf;           // H(0x00 || cid)
  for p in path {
    if s == 0 { return false }
    if f & 1 == 1 || f == s {
      r = H(0x01 || p || r); // left sibling
      while f & 1 == 0 && f != 0 {
        f >>= 1; s >>= 1; // no right sibling
      }
    } else {
      r = H(0x01 || r || p); // right sibling
    }
    f >>= 1; s >>= 1;
  }
  s == 0 && ct_eq(r, root) // constant-time
}
```

Receipt check: (1) verify the checkpoint signature against the pinned log key and the cosignatures against the witness set; (2) verify the inclusion path as in RFC 9162 §2.1.3.2 (code at left; tested versions in Listings A.4 and A.8); (3) optionally, verify a consistency proof from the last trusted checkpoint. Verification uses only hashes and pinned keys; it needs no database access.

LOG SIZE <i>N</i>	PATH HASHES	PROOF BYTES
10 ⁶	20	640
10 ⁹	30	960
10 ¹²	40	1 280

TABLE 3.3 – Inclusion proof size = ⌈log₂ *n*⌉ × 32 B (worst case); consistency proofs are bounded by ⌈log₂ *n*⌉ + 1 nodes^[32].

MODELED • STORAGE & THROUGHPUT ESTIMATE

Assumes 5 M agent actions/day/tenant, 4 events each, 1 KiB mean entry, 10× peak-to-mean.

Yields 20 M leaves/day: ≈ 230 appends/s mean, ≈ 2 300/s peak; ≈ 20 GB/day of entries (≈ 7.3 TB/yr) + ≈ 0.64 GB/day of interior hashes. Peak hashing ≈ 2.4 MB/s; one checkpoint signature per second is negligible (Ed25519 reference code signs 109 000 and verifies 71 000 messages/s on a 2.4 GHz quad-core Westmere^[118]). The binding constraints are replicated commit latency and storage; cryptographic cost is minor by comparison.

What a receipt proves and what it does not

A receipt is integrity evidence: the record has not changed since it was logged, and it sits at a fixed position under a checkpoint that independent witnesses saw. It does not show that every action was logged, because I1 is enforced by the same gateway the auditor is assessing; completeness therefore rests on IT general controls over AE-OS itself (change management for policy modules, access to the sequencer and keys, SPIRE registration entries). Nor does it show that a payload was accurate at its source or that the policy which allowed an action was the right one. Verifying every receipt replaces sampling for integrity only. Tests of control design and operating effectiveness remain, and for a hosted platform the auditor would look for a service-organization controls report^[76,75] (Assurance, §2.17).

PROPERTY	RECEIPT	ALSO NEEDED
Integrity since logging	proves	hash and signature assumptions (App. B)
Inclusion and order	proves	a witness-cosigned checkpoint
Completeness	does not prove	I1 and IT general controls over the gateway; signed ExternalInput for bank, EDI and e-invoice data
Accuracy of payload	does not prove	source controls; CalculationExecution replay for computed amounts (§2.11)
Authorization	does not prove	policy review and change management; replay shows only which policy gave the verdict

TABLE 3.4 – What a Causal Receipt proves. Auditors test the platform's controls before relying on receipts (§2.17).

03 CAUSAL ACCOUNTABILITY CORE

Wasm Policy Guardrails

AE-OS runs a deterministic, sandboxed policy check *before* every tool call and logs the decision.

3.7 Policy as content-addressed WebAssembly

Controls (segregation of duties, spend limits, sanctions, residency) are written in a declarative policy language and compiled to WebAssembly. OPA compiles Rego via `opa build -t wasm`; built-ins such as `http.send` are unavailable unless the host supplies them^[30]. Cedar offers a validator and a formal model proven in Lean^[31]. AE-OS specifies the module **ABI** and leaves the choice of policy language open.

Module, data bundle and ABI version hash to a **policy CID**, referenced from the agent manifest (`policy_refs[]`, §4.1). Promoting a version is itself a signed Causal Event, so an auditor can prove which exact bytes judged an action.

```
FILE // SPEC/POLICY/ABI.WIT (SKETCH)

record decision-input {
  intent:  cid,           // Intent Envelope
  actor:   string,        // SPIFFE ID
  action:  string,        // "erp.payment.release"
  resource: list<u8>,     // DAG-CBOR facts snapshot
  snapshot: cid,          // state version read
}

enum verdict { allow, deny, escalate }
record decision {
  verdict:  verdict,
  reasons:  list<string>, // rule IDs that fired
  obligations: list<string>, // e.g. "dual-approval"
}

export evaluate: func(i: decision-input) -> decision;
```

3.8 Sandbox and determinism

Modules run in **Wasmtime**, where an instance acts only through explicitly linked interfaces: no raw system calls, filesystem or network^[119]. AE-OS links only an allocator, so a module has neither an exfiltration channel nor a clock. Execution is bounded by **fuel** (`consume_fuel`; exhaustion traps), which Wasmtime recommends where deterministic trapping is needed, unlike epoch interruption, which is "not deterministic"^[29]. Residual Wasm nondeterminism is enumerable: NaN bits, threads, relaxed SIMD, host calls, resource exhaustion^[55]. AE-OS disables threads and relaxed SIMD, enables `cranelift_nan_canonicalization`, passes inputs by value and treats any trap as **DENY** (fail-closed). A decision can therefore be *replayed* (same policy CID + input digest $\Rightarrow$ same verdict and fuel count); model output cannot (§3.2).

CONTROL	RULE (ILLUSTRATIVE)	ON VIOLATION
SoD ^[98]	creator of a vendor may not release payment to it (causal graph, by accountable principal)	DENY
Spend limit	<code>amount ≤ min(intent.ceiling, role.limit)</code> , cumulative per intent	ESCALATE
Sanctions	payee not in screening list snapshot (list CID recorded in input)	DENY + alert
Residency	tool endpoint region $\in$ data-class allowed regions	DENY

TABLE 3.5 – Example guardrails. Each evaluation is appended as a Policy Guardrail Node *before* the tool call (I1); an ESCALATE opens a Strategic Exception whose human approval is a new signed child event. SoD compares accountable principals (envelope signer, policy-set approver), not workload identities; two agents under one principal are one party, and a conflict escalates to an independent human.

3.9 Latency budget

In the Cedar authors' benchmarks, Cedar answered authorization requests at a median 4.0–5.0 μ s (5–50 entities; p99 under 10 μ s on the gdrive workload) and was 28.7–35.2 $\times$ faster than OpenFGA and 42.8–80.8 $\times$ faster than Rego (OPA as benchmarked there, not Rego compiled to Wasm)^[31]. The budget treats such figures as lower bounds and adds instantiation, marshalling and the ledger append; AE-OS-specific benchmarks are part of the evaluation in §6.3. The append returns a signed promise; the receipt follows at the next checkpoint (§3.5).

STEP (PER TOOL CALL)	TARGET P99
SVID / capability token check	0.2 ms
Instantiate pooled module, marshal input	0.3 ms
Policy evaluation (fuel-capped)	0.5 ms
Append PolicyDecision (quorum commit)	4.0 ms
Total pre-execution overhead	≤ 5 ms

MODELED Design targets; excludes inference and tool time.

04 DEVELOPER ECOSYSTEM & PLATFORM PRIMITIVES

The Agent Primitive Object

ERP extensibility has meant customer code inside the vendor's monolith. AE-OS developers instead ship Agent Primitive Objects (APOs): small attested agents with metered budgets and no standing credentials, which act only through policy-checked calls that return receipts.

4.1 APO structure and platform obligations

An APO is a content-addressed bundle that binds **who** an agent is (a SPIFFE ID), **what** it may do (a manifest of capabilities, MCP tools and budgets), **which rules** govern it (CIDs of compiled policy modules), and **what it runs on** (model and memory references). AE-OS specifies five runtime-enforced obligations per APO: (1) identity is issued only after workload attestation; (2) authority is delegated per Intent Envelope as an attenuable token; (3) every external effect passes a pre-execution policy check; (4) the runtime meters budgets itself; (5) every lifecycle transition and tool call is a Causal Event with a Merkle receipt (§3).

Because an APO is named by the CID of its canonical encoding, changing a tool, budget or model version yields a *new* APO that must be re-attested.

```
FILE // APO/PROCUREMENT-REPLENISH.APO.YAML

apiVersion: ae-os.dev/v1
kind: AgentPrimitiveObject
spiffe_id: spiffe://acme.ae-os/agent/procurement/<i>
manifest:
  capabilities: [erp.po.create, erp.po.read]
  tools:
    # pinned by definition digest
    - { server: spiffe://acme.ae-os/mcp/suppliers,
        name: supplier.quote, schema: sha256:4be1... }
    - { server: spiffe://acme.ae-os/mcp/erp-core,
        name: po.submit, schema: sha256:91c0... }
  budgets:
    spend_minor: 2000000 # EUR 20,000.00
    tool_calls: 50
    model_tokens: 400000
    wall_clock_s: 900
    egress: deny-all # MCP gateway only
    residency: [eu-central] # data-plane shards
  policy_refs: [bafyreii...sod.wasm, bafyreii...spend.wasm]
  model_ref: model://vendor/m@2026-09#sha256:...
  memory_ref: kv://acme/agents/procurement
```

APO FIELD CONTRACT

- `spiffe_id` workload identity, issued post-attestation
- `manifest` capabilities, tools, budgets, egress
- `policy_refs[]` CIDs of Wasm policy modules
- `model_ref` model + version (weight digest only if self-hosted)
- `memory_ref` encrypted state handle
- `lifecycle_state` read-only projection of Lifecycle events

Budget dimensions

Spend (integer minor units), tool calls, model tokens and wall-clock time are each charged *before* the action they pay for. The effective limit is the minimum of the manifest budget and the bound Intent Envelope's budget: a developer cannot raise authority by editing a manifest, and a principal cannot exceed what the agent was certified for. Exhaustion triggers a logged transition to `SUSPENDED`.

Tool pinning

Each tool entry pins the digest of the MCP tool definition (name, description, `inputSchema`) observed at attestation time. A server that later changes a description or schema (the "rug pull" pattern described by Invariant Labs^[120]) produces a digest mismatch and the gateway refuses the call until the APO is re-attested. Pinning covers the declared interface, not the implementation; behavioral drift is caught only by policy checks on effects.

Suspension and retirement

`SUSPENDED` keeps memory and the ledger chain but freezes all tokens, so a human can inspect an agent mid-flight. `RETIRED` is absorbing, which makes "decommissioned" provable: any later event from that SPIFFE ID contradicts the retirement receipt.

4.2 APO lifecycle state machine

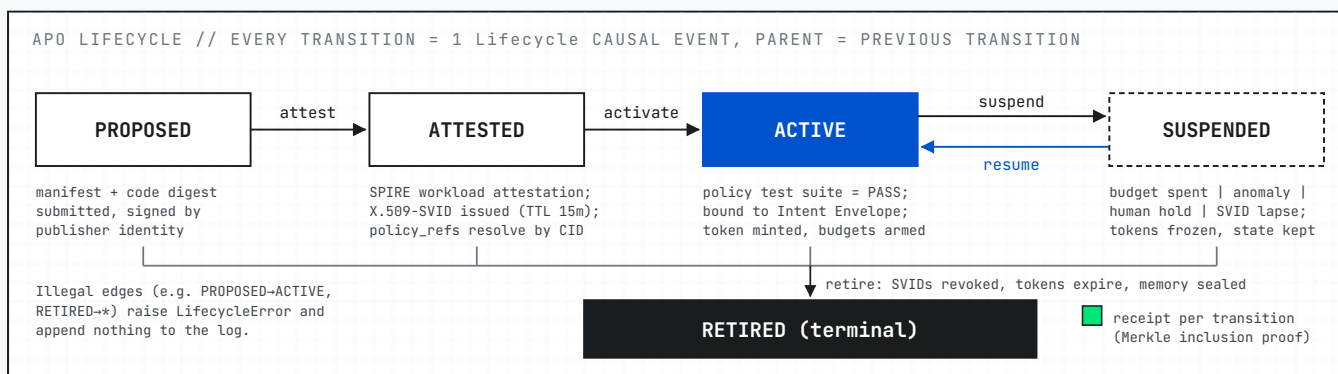


FIG. 4.1 – APO lifecycle state machine; text under each state lists its entry conditions. Only `ACTIVE` (blue) holds live tokens; `SUSPENDED` (dashed) holds frozen ones. Undrawn edges are rejected before anything is logged (Listing A.5).

4.3 Capability delegation and attenuation

An APO has no authority until an Intent Envelope is bound to it; it then holds a **capability token** with caveats derived from that envelope. Tokens follow the macaroon construction: each caveat is folded into a chained MAC, so any holder can *append* a caveat offline, but nobody can remove or edit one without invalidating the tag^[121]. Production tokens are specified as public-key, Datalog-checked chains in the style of Biscuit, verifiable without a shared secret^[122]; Appendix A uses dependency-free HMAC-SHA-256.

$$\text{Auth}(t_{k+1}) = \text{Auth}(t_k) \cap \{r : c_{k+1}(r)\} \subseteq \text{Auth}(t_k) \quad (\text{Eq. 4.1})$$

Caveats are conjunctive, so Eq. 4.1 holds by construction: a sub-agent appending `max_amount ≤ 5,000,000` under a parent caveat of 1,000,000 stays bounded by 1,000,000 (minor units: the EUR 10k hop-0 ceiling; tested in `capability.rs`). Each token names its envelope's CID.

HOP	ISSUER → HOLDER	CAVEATS APPENDED	VERIFIER NEEDS
0	CFO signs Intent Envelope	budget EUR 25k, ceiling EUR 10k, expiry 24 h	principal key
1	authority → procurement APO	<code>intent</code> <code>aud</code> <code>actions</code> <code>max_amount</code> <code>not_after</code>	tenant root key
2	APO → quoting sub-agent	<code>actions=supplier.quote</code> <code>max_amount ≤ 0</code>	same, no round-trip
3	gateway → MCP server	none; token stays internal, server gets a JWT-SVID	SPIRE bundle

STANDARD CAVEATS

- `intent=<CID>` binds to one envelope
- `aud=<SPIFFE ID>` this holder only
- `actions=...` subset of manifest
- `max_amount=<n>` minor units, ≤ ceiling
- `not_after=<t>` ≤ envelope expiry
- `tool=<digest>` pinned definition

The envelope signer answers for its constraints, not for each execution under them (accountability model, §5.1).

4.4 SDK surface

Handlers receive an `Intent`, call tools by name and get **Causal Receipts**; the runtime handles identity, tokens, policy and ledger writes. The API shown is illustrative; the Rust listing uses the coexistence pattern (§2.16).

```
PYTHON // AGENTS/REPLENISH.PY

from aeos import Agent, StrategicException
agent = Agent.from_manifest("replenish.apo.yaml")

@agent.on_intent("replenish")
async def replenish(ctx):
    q = await ctx.tools.call("supplier.quote",
                           sku=ctx.intent.args["sku"], qty=500)
    best = min(q.structured,
               key=lambda x: x["total_minor"])
    ctx.trace("cheapest compliant quote",
              inputs=[q.cid]) # ReasoningTrace
    amt = best["total_minor"]
    try:
        async with ctx.budget.reserve(spend_minor=amt):
            ack = await ctx.tools.call("po.submit",
                                      **best)
    except StrategicException as ex: # ESCALATE
        return ex.await_human() # parks the APO
    return ack.receipt # Causal Receipt (§3)
```

```
RUST // MCP-SERVERS/ERP-CORE/SRC/PO.RS

// Coexistence (§2.16): existing ERP (e.g. SAP) stays
// book of record; erp::create_po wraps its PO API.
use aeos_sdk::{mcp::CallCtx, ToolError};
/// Listed by tools/list; digest pinned by callers.
#[aeos::tool(name = "po.submit",
              capability = "erp.po.create")]
async fn po_submit(ctx: &CallCtx, req: PoRequest)
    -> Result<PoAck, ToolError>
{
    // Already enforced before this line runs:
    // JWT-SVID aud == this server, exp checked;
    // Wasm policy ALLOW; PolicyDecision logged (r1).
    let caller = ctx.peer_spiffe_id();
    ctx.require_amount(req.total_minor)?; // defense
    let po = erp::create_po(&req, caller).await?;
    Ok(PoAck { po_id: po.id,
               policy_receipt: ctx.policy_receipt() })
}
```

4.5 Developer workflow and release gates

STAGE	COMMAND	GATE (MUST PASS) → ARTIFACT RECORDED IN THE LEDGER
1 • Sandbox	<code>aeos dev</code>	local SPIRE, recorded MCP fixtures, ephemeral ledger, no egress → replayable traces
2 • Attest	<code>aeos attest</code>	reproducible digest matches, manifest signed, SPIRE entry → APO CID, ATTESTED
3 • Policy test	<code>aeos policy test</code>	traces replayed through Wasm policies, all DENY/ESCALATE paths hit → coverage CID
4 • Deploy	<code>aeos deploy --shadow</code>	shadow run (effects dry-run), human approves promotion → ACTIVE receipt

4.6 Model Context Protocol as the tool bus

AE-OS agents reach tools and data through the **Model Context Protocol (MCP)**, released by Anthropic as an open standard in November 2024 to replace "fragmented integrations with a single protocol"^[123]. In revision 2026-07-28, MCP exchanges JSON-RPC 2.0 messages between hosts, clients and servers; servers expose **resources**, **prompts** and **tools**, and each request is self-contained, carrying protocol version and client capabilities in `_meta`^[124]. Clients discover tools at runtime via `tools/list` (self-describing `name`, `description`, `inputSchema`) and invoke them via `tools/call`; clients "MUST consider tool annotations to be untrusted unless they come from trusted servers"^[125].

MCP MECHANISM	WHAT THE SPECIFICATION REQUIRES	AE-OS BINDING (DESIGN)
stdio transport	JSON-RPC over a subprocess's stdio; credentials from the environment (no OAuth) ^[126,127]	APO sandbox only; "environment" = Workload API socket
Streamable HTTP	one POST endpoint, JSON or SSE reply; no protocol sessions; <code>Origin</code> MUST be validated ^[126]	servers reachable only from the gateway's cgroup (§4.9)
Mirrored headers	<code>Mcp-Method</code> , <code>Mcp-Name</code> REQUIRED; mismatch → <code>-32028</code> ^[126]	pre-filter on method + tool name without body parsing
<code>tools/list</code>	deterministic order; MAY vary by authorization ^[125]	filtered to manifest; definition digests pinned
Authorization	OAuth 2.1 resource server, RFC 9728 metadata, audience check; "MUST NOT accept or transit any other tokens" ^[127]	internal: JWT-SVID, <code>aud</code> = server; external: standard OAuth; no passthrough

4.7 The AE-OS MCP gateway

Each APO's only route to any MCP server is a gateway sidecar that turns the intent-scoped capability token into a short-lived, audience-bound bearer credential that MCP servers already understand, after policy and ledger have run. SPIFFE recommends X.509-SVIDs where possible and JWT-SVIDs when an L7 intermediary sits between workloads^[77], as with MCP over HTTP. JWT-SVIDs must carry `aud` and `exp`, are limited to RS/ES/PS signatures, and are replayable bearer tokens, so aggressive expiry is advised^[128]: AE-OS specifies one audience, `exp` ≤ 5 min, and carriage inside mTLS keyed by the gateway's X.509-SVID.

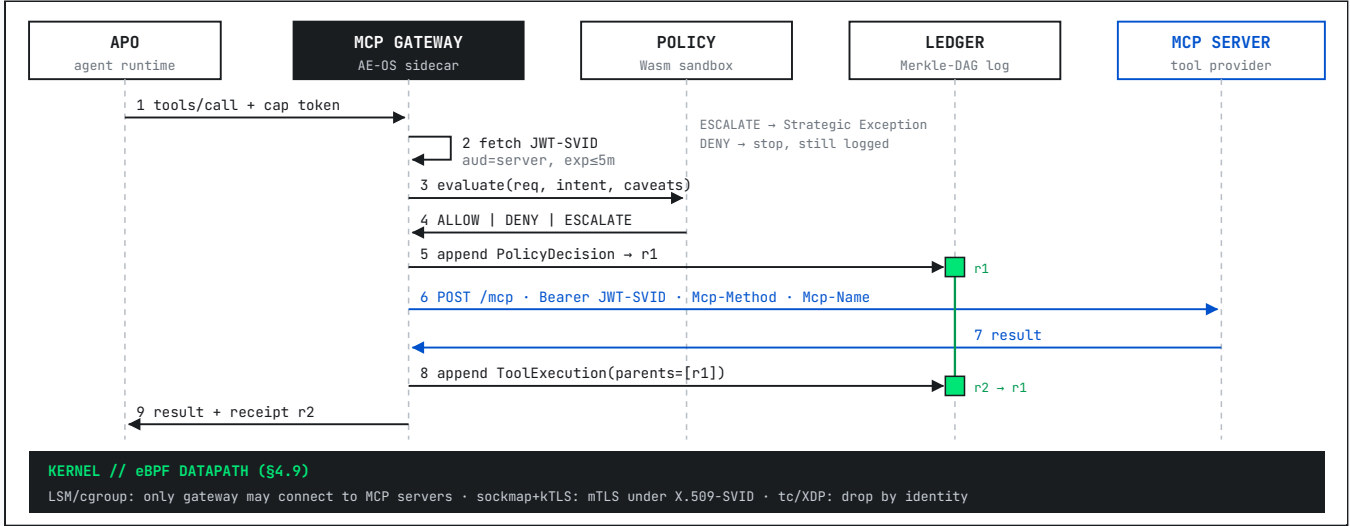


FIG. 4.2 – One `tools/call` through the AE-OS gateway. Blue arrows are traffic to and from the MCP server; green squares are ledger nodes. Steps 3–5 finish before any byte reaches the server (log-before-act, §3). The `ToolExecution` node `r2` (step 8) links to the `PolicyDecision` node `r1` (step 5), and the caller receives the inclusion proof of `r2` as its Causal Receipt.

Failure semantics. If step 5 fails, nothing is sent. If the server succeeds but step 8 fails, the gateway retries the append (CIDs make it idempotent) and withholds the result. A persistent failure leaves an effect without a receipt, reconciled from the server's own log. Denials are also logged.

What the server sees. A standard MCP request whose token `sub` is the caller's SPIFFE ID. Servers need no AE-OS library but must accept a JWT-SVID as OAuth access token (issuer, JWKS, audience) and speak 2026-07-28 for header pre-filtering; older ones use a translating gateway mode.

Catalog drift. Via `subscriptions/listen` the gateway receives `notifications/tools/list_changed`^[125], then re-lists and re-hashes. A changed definition blocks that tool for every APO pinned to the old digest until re-attested; the rest of the agent keeps working.

4.8 MCP threat model and controls

MCP's specification says it "cannot enforce these security principles at the protocol level"^[124]. AE-OS assumes the model *will* sometimes follow injected instructions and puts controls between intent and effect.

THREAT	DOCUMENTED MECHANISM	AE-OS CONTROL (DESIGN)
Tool poisoning	instructions hidden in descriptions, invisible to users but read by the model ^[128] ; OWASP MCP03 (beta) ^[129]	descriptions reviewed and digest-pinned at attestation; actions still need token + policy
Rug pull / shadowing	description altered after approval; one server's description steering a trusted server's tool ^[120]	re-hash on <code>list_changed</code> ; only listed servers in context; drift under an unchanged schema: policy on effects only
Confused deputy	OAuth proxy with static client ID + consent cookie leaks auth codes ^[130]	no shared proxy credentials; per-intent tokens with <code>aud</code> = one APO
Token passthrough	server forwards a token not issued to it; forbidden by spec ^[139,127]	capability tokens never leave AE-OS; JWT-SVID minted per server
Client RCE	CVE-2025-6514 (CVSS 9.6): <code>mcp-remote</code> handed a server-supplied auth URL to the OS ^[131]	gateway never opens URLs or shells; LSM denies <code>exec</code> from gateway and APO cgroups
Context injection	malicious tool output or resources steer later calls; OWASP MCP06/MCP10 ^[129]	results confer no authority; induced calls face the same caveats and policy

4.9 eBPF enforcement below the application

So that the gateway cannot be bypassed, AE-OS enforces per-identity network and process policy in the kernel with eBPF: sandboxed programs, checked by a verifier for safety and termination, JIT-compiled to native code^[132].

HOOK	KERNEL MECHANISM	AE-OS USE (DESIGN)
XDP	driver-level hook; 24 Mpps/core reported ^[133]	drop unidentified traffic at ledger nodes
tc + identity	Cilium: identity derived from labels instead of IPs ^[134]	egress only APO → gateway → listed servers
LSM BPF	MAC/audit on LSM hooks; <code>-EPERM</code> denies ^[135]	deny <code>connect</code> / <code>exec</code> outside the manifest
sockmap + kTLS	socket policy/redirect ^[136] ; kernel TLS records ^[48]	SVID mTLS handshake ^[49] , kernel records
Tetragon	in-kernel <i>Override</i> or <i>SIGKILL</i> , synchronous ^[137]	kill an APO that deviates; emit a Causal Event

4.10 Platform primitives summary

PRIMITIVE	ADDRESSED BY	LIFETIME	ENFORCED BY	APPENDIX A
Intent Envelope	CID, principal signature	until <code>exp</code>	gateway, policy	<code>IntentEnvelope</code>
APO	CID + SPIFFE ID	5-state lifecycle	runtime, SPIRE	<code>APO</code> , <code>TRANSITIONS</code>
Capability token	intent CID + caveat chain	≤ envelope expiry	gateway	<code>capability.rs</code>
SVID (X.509 / JWT)	SPIFFE ID	15 min / ≤ 5 min	SPIRE, kernel, server	n/a
Policy module	CID of Wasm binary	versioned	Wasm sandbox	<code>PolicyGuardrail</code> (toy)
Causal Event	CIDv1 (dag-cbor, sha2-256)	permanent	ledger	<code>CausalEvent</code> (Py/Rust)
Causal Receipt	(index, size, path, root)	permanent	any verifier	<code>verify_inclusion</code>
Strategic Exception	CID of ESCALATE decision	until resolved	policy, human	<code>ESCALATE</code>
MCP tool binding	server ID + definition digest	until drift	gateway	manifest

DEVELOPER CONTRACT

Developers write handlers and a manifest. The platform binds each action that moves money or data to an identity, checks it against policy and records a receipt.

05 HUMAN OVERSIGHT

Human Oversight and Intent Mapping

In AE-OS a human signs an intent, agents execute inside the constraints compiled from it, and the exception dashboard shows only what policy will not let agents decide alone.

5.1 The Strategic Exception Dashboard

Legacy ERP keys, approves or releases every transaction through a screen, which places the human *in* the loop and rules out management by exception. Before execution the Wasm policy engine assigns every agent action one disposition: **ALLOW** (inside the Intent Envelope and all policies; executes, returns a Causal Receipt), **DENY** (hard-policy violation such as sanctions or segregation of duties; never approvable) or **ESCALATE** (compliant but tripping a deterministic trigger such as an **authority ceiling**). Only ESCALATE creates a *Strategic Exception*, the sole object the dashboard renders; routine per-run approvals are Human Tasks (§2.12).

An exception is a frozen, content-addressed bundle containing the proposed plan, the breached constraint and margin, at least one counterfactual ("hold"), the reasoning-trace CID, upstream receipts, a decision SLA and a **fail-safe default**, which must be the non-committing or reversible option, applied if the SLA lapses. After step-up authentication the approver signs with a WebAuthn credential bound to their SPIFFE session; the decision is a `PolicyDecision` node like any agent act, and mints a single-use capability scoped to that exception.

In this **human-on-the-loop** model, people set the envelope and decide its edge cases. The design is intended to support Article 14(4)(d)–(e) of the EU AI Act (disregard, override or reverse an output; interrupt the system so it halts in a safe state)^[4]. Article 14 as a whole also needs deployer measures, e.g. training on limitations and automation bias. Most ERP uses are not Annex III high-risk (credit scoring and employment are exceptions); AE-OS applies the measures anyway. Accountability follows FIG. 5.1.

ESCALATION TRIGGERS (DEFAULT)

VALUE	action Δ > ceiling_per_action
BUDGET	Σ envelope spend > cap
IRREVERSIBLE	reversible=false above floor
NOVELTY	counterparty / route not in attested registry
CONFIDENCE	separate, evaluated estimator (not self-report) < floor; off until validated
CONFLICT	two policies emit contradictory obligations
HARD DENY	sanctions, SoD, residency: DENY , no override from dashboard

Exception rate per 1,000 actions is an operating metric: too high recreates the ERP bottleneck; too low suggests loose ceilings, which are tuned per envelope.

5.2 Intent Mapping



Principals state goals as a sentence; back-office staff keep Work Surfaces (§2.12) without routine data entry. Deterministic components enforce what models interpret. A planner model drafts the envelope ②, which binds once signed. The compiler ③ grounds each term in the Semantic Data Plane ontology ("near-shore" becomes a typed predicate over `Supplier.region`) and rejects any term it cannot type; an unresolved term costs one recorded clarification turn. Plans ④ are typed action lists over MCP tool schemas^[123]; ESCALATE at ⑤ returns to the principal (dashed path).

The principal signs the compiled structure: readers may interpret a sentence differently, but every policy evaluation of a given envelope and input reaches the same verdict (agents' plans are not deterministic).

```

FILE // INTENTS/BAFYREI...Q7MZ.ENVELOPE.YAML

utterance: "Maintain 24% Gross Margin, prioritize
  near-shore supply chains, hold OTIF >= 97% through
  Q4; max $2.0M unplanned freight without sign-off."
principal: spiffe://acme.ae-os/human/h-5c2e91 # C00
expiry: 2026-12-31T23:59:59Z
objective: {max: otif_q4, tiebreak: min_landed_cost}
constraints:
  - {id: c1, gross_margin >= 0.24, scope: q4, hard}
  - {id: c2, nearshore_share >= 0.60, soft, w: 0.7,
    pred: "Supplier.region in {VN,TH,MY,PH}"}
  - {id: c3, otif_q4 >= 0.97, hard}
  - {id: c4, freight_unplanned_usd <= 2_000_000,
    agg: cumulative, on_breach: ESCALATE}
authority: {per_action_usd: 500_000, delegates:
  [logistics/*, sourcing/*, treasury/hedging/*]}
policies: [sod.v4, sanctions.v7, residency.v2]
clarified: "near-shore" → APAC-SEA # 09-01
sig: ed25519:9f1c...e04b cid: bafyreii...q7mz
  
```

Strategic Exception Dashboard Mockup



FIG. 5.1 – Strategic Exception Dashboard, design mockup: open exception EXC-0921 in the modeled typhoon scenario (§5.3–5.5), a policy-compliant reroute by Logistics_Node_SA that breaches the unplanned-freight ceiling. All names, figures, CIDs and timestamps are illustrative.

Reading the mockup. The left pane shows the signed goal verbatim from the §5.2 envelope and an authority meter separating **committed** freight (\$1.38M, \$5.4) from the **proposal** (\$1.24M), \$0.62M over the \$2.0M cap. The p99 policy-evaluation tile (0.4 ms) sits inside the 0.5 ms target of §3.9. **APPROVE** requires step-up auth and emits a signed `PolicyDecision`; **MODIFY** returns a narrowed envelope; **REJECT** executes the counterfactual (“hold in port”).

Accountability model (§5.1). An agent error within policy is a control-design deficiency, assigned to the owner of the control that allowed the action, or to the signer where the envelope’s constraint was wrong; an agent is never the accountable party. Loss allocation between operator, pack publisher, model provider and deployer (including their EU AI Act provider and deployer roles) is contractual, outside this paper, and must be settled before production; the ledger shows which CID allowed the action and which model version proposed it. For segregation of duties AE-OS compares accountable principals (envelope signer, approver of each APO’s policy set), not workload identities: two agents under one principal count as one party, and a conflicting pair escalates to a second, independent human.

ROLE (WHO)	ACCOUNTABLE FOR	EVIDENCE
Control owner: manager who approves a policy or Calculation Module CID	design and operation of that automated control (SOX control owner)	signed promotion; test coverage CID (§4.5)
Envelope signer: principal of the Intent Envelope	constraints, ceilings and delegates set, and each exception decided; not each execution	envelope CID; approval <code>PolicyDecision</code> s
Pack publisher: any party signing a pack, APO or module CID (§2.13)	behavior of the released artifact against its published test vectors	publisher signature; attestation receipt
Operator: party running the tenant (in-house IT or host)	platform controls: key custody, SPIRE registration, log-before-act (l1), witnesses	ITGC evidence; SOC 1 report (§2.17)

DESIGN RULES FOR EXCEPTION SURFACES

(1) Only ESCALATE outcomes are rendered; allowed actions stay in the receipt log. (2) Every number links to a receipt. (3) Every exception carries a counterfactual and a fail-safe default, always the non-committing or reversible option. (4) Approval is bound to the approver's identity and the exception's scope, not to a session. (5) A signed CLI (`ae-os exc approve EXC-0921 --sig`) keeps oversight available during a UI outage.

South China Sea Typhoon Response Scenario

This section runs the AE-OS control loop against a real, recurring hazard. The port closures and delays are documented; the firm, the storm and every number in §5.4–5.6 are modeled.

\$3.4_T TRADE TRANSITING THE SOUTH CHINA SEA, 2016 (CSIS EST.) ^[138]	~21% OF GLOBAL TRADE, SAME ESTIMATE ^[138]	6.08 TROPICAL CYCLONES / YR WITHIN 500 KM OF HONG KONG, 1961–2020 MEAN ^[139]	16 days VESSEL DELAYS OUTSIDE YANTIAN, JUNE 2021 (MAERSK) ^[140]
---	--	---	--

5.3 Typhoon Disruption to South China Sea Ports

CSIS ChinaPower estimates \$3.4 trillion of trade crossed the South China Sea in 2016, noting that the often-repeated \$5.3 trillion overstates it^[138]. Its northern rim is a busy tropical-cyclone corridor: on Hong Kong Observatory records about six cyclones a year pass within 500 km of the city, peaking July–September^[139].

Ports suspend gate operations well before landfall. Ahead of Super Typhoon Saola (2023), Yantian and Nansha stopped gate services at 16:00 and Hong Kong International Terminals at 17:00 on 31 August, ahead of expected landfall on 1 September; officials warned it could be the strongest to hit the Pearl River Delta since 1949^[141].

For Yagi (2024), Shekou suspended pick-up and delivery from 23:00 on 4 September; Yantian, Hong Kong, Da Chan Bay and Jiangmen followed on 5 September, Haiphong on 6 September^[142], and Hainan shut its ports^[143]. Before Krathon (2024), CPC and Formosa closed their Kaohsiung and Mailiao ports^[144].

Congestion adds to the delay: when COVID-19 cut Yantian to ~40% of capacity in 2021, Maersk cited delays of up to 16 days, “a much bigger disruption” than the Ever Given in Suez^[140]. Because **the operational window closes a day or more before landfall**, the time to an *authorized* decision matters more than the time to detection.

5.4 Scenario Set-up

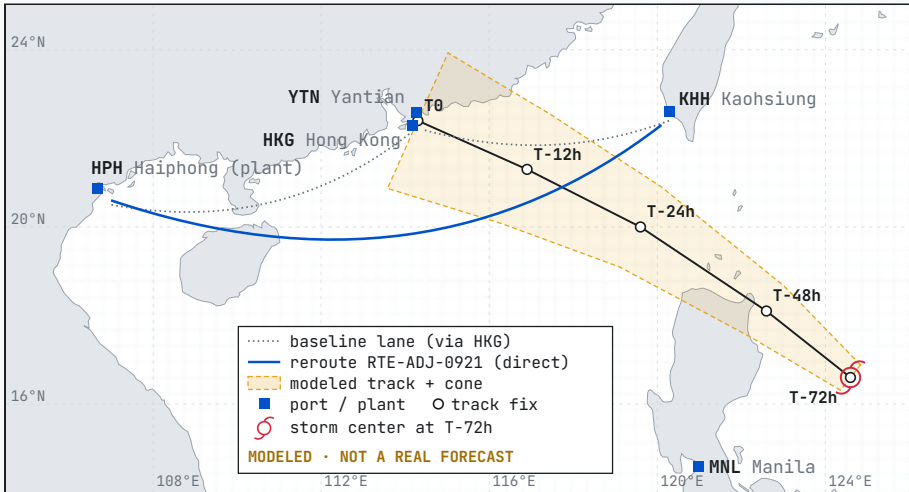


FIG. 5.2 – Modeled scenario: synthetic storm track (T-72h → T0) and cone, baseline lanes via Hong Kong, approved reroute RTE-ADJ-0921 (Kaohsiung → Haiphong direct, ETD pulled forward 36 h). Basemap: Natural Earth^[145]. Not a forecast.

MODELED PARAMETERS

FIRM	“ACME”, assembler, plant at Haiphong
INBOUND	PCBAs via YTN; modules KHH→HKG→HPH
EXPOSURE	2,380 TEU via HKG/YTN, T-24h...T+72h
STORM	synthetic “SIM-26”, T0 = 09-17 03:00Z
GATES	HKG/YTN shut T-30h → T+24h
ENVELOPE	\$5.2: \$2.0M cap, \$0.5M/action
COMMITTED	\$1.38M Q4 freight, pre-event
BUFFER	3 days inbound cover

SCENARIO PREMISE • MODELED

The agents are those in the agent graph of FIG. 5.1. The scenario tests three claims and does not test forecast skill: (1) agents act inside the signed envelope without waiting for business hours; (2) the one decision that breaches a ceiling reaches a human in minutes with evidence attached; (3) every step, including the human's, leaves a verifiable receipt. Gate timings mirror the pattern in the cited Saola and Yagi notices; the firm, volumes and costs are invented.

05

HUMAN OVERSIGHT • SIMULATION, CONTINUED

MODELED

5.5 Event Timeline, T-72h to T+48h

T0 = synthetic landfall, 2026-09-17 03:00Z (T-72h = 09-14 03:00Z). Each row is a Causal Event in the Merkle-DAG; the receipt is the (abbreviated, illustrative) CID a verifier would resolve. Rows 1–5 match the receipt tail in FIG. 5.1.

T	ACTOR	ACTION	POLICY DECISION	RECEIPT
-72:20	Weather_Feed_MCP → Logistics_Node_SA	Forecast ingested as RiskEvent ; joined to open shipments: 2,380 TEU exposed, P(HKG/YTN gate suspension) = 0.81	read-only	bafy...f3qe
-72:19	Policy engine	Screens candidate counterparties and data flows	ALLOW residency.v2, sanctions.v7	bafy...m90d
-72:16	Hedging_Agent_02	USD/VND forward, \$0.45M notional, for planned re-sourced local spend; hedge documentation in the PolicyDecision	ALLOW < \$0.5M/action	bafy...ue5c
-72:09	Sourcing_Agent_01	RFQ + capacity reservation for 40% of Q4 PCBA at a pre-qualified, attested Haiphong supplier; near-shore 0.58 → 0.63	ALLOW c1–c4 hold	bafy...9wpn
-71:50	Logistics_Node_SA	RTE-ADJ-0921: 3 KHH sailings (1,140 TEU) direct to HPH, ETD –36h; Δ freight +\$1.24M	ESCALATE c4 \$2.62M > \$2.0M	bafy...c81t
-71:29	COO (human)	Approves EXC-0921 after step-up auth; single-use capability minted, consumed by 3 bookings at –71:14	PolicyDecision, signed human/h-5c2e91	bafy...h7d2
-48:00	Inventory_Planner_03	Safety stock +4 days on 6 SKUs; 240 TEU of YTN exports pulled forward to gate-in before –30h	ALLOW c1, c3 projected	bafy...2n8v
-30:00	(environment)	Modeled HKG/YTN gate suspension; no ACME containers at gate	—	bafy...x31j
-12:00	Logistics_Node_SA	\$0.9M spot charter from an unattested counterparty	ESCALATE → SLA lapses → fail-safe REJECT	bafy...q5te
+24:00	Inventory_Planner_03	Gates reopen (modeled); remaining 1,000 TEU resequenced, slip ≤ 2 days, inside buffer	ALLOW	bafy...7hcm
+48:00	Causal ledger	Signed checkpoint; incident receipt bundle exported for audit	—	c4e1...9a0f

TABLE 5.1 – Modeled event timeline. All actors, values and CIDs are illustrative.

5.6 Modeled Outcomes Against Two Manual Baselines

METRIC	A3	A6	AE-OS
Exposure known after	~6 h	~1 h	2 s
Ceiling decision at	T-46h	T-67h	T-71:29
TEU moved pre-shut	684	1,380	1,380
TEU caught	696	0	0
Unplanned freight	\$1.12M	\$1.24M	\$1.24M
Line-down	48 h	0 h	0 h
Q4 OTIF impact	-3.0 pp	≈ 0 pp	≈ 0 pp
Evidence	email	email	1 bundle

TABLE 5.2 – MODELED. A3 = manual, A6 = manual with control tower. Outputs follow arithmetically from A1–A6; not empirical estimates.

ASSUMPTIONS • MODELED

A1 Reroute cost \$1,088/TEU if booked before T-60h. **A2** Inside T-48h: +50% premium and only 60% of space obtainable. **A3** Manual path: detection +6h; S&OP review and CFO sign-off for any action > \$0.5M, +20h; no ETD pull-forward after T-60h; YTN pull-forward not executed. **A4** Stranded boxes slip 5 days against a 3-day plant buffer. **A5** –1.5 pp Q4 OTIF per 24h of line-down. **A6** Manual with control tower: alert +1h; delegated authority \$0.5M/action; sign-off above it +4h; YTN pull-forward executed. Manual freight = 684 TEU × \$1,088 × 1.5.

Sensitivity. If the COO took 12 h rather than 21 min (decision still before T-60h), the AE-OS column is unchanged; past T-60h it converges toward the A3 column. A6 matches AE-OS while detection plus sign-off stays under about 12 h. The envelope’s value is bounded by approver availability, which is why every exception carries an SLA and a fail-safe default.

Reading the result. The modeled AE-OS path spends *more* on freight than A3 (\$1.24M against \$1.12M) and avoids the 48 h line-down. A6 shows where the gain comes from: pre-delegated authority and fast escalation, which a disciplined manual process with a control tower can replicate in this model. AE-OS’s distinct contribution is that the delegation is enforced before execution and evidenced by receipts. Forecast error is not modeled: if the storm recurses, the hedge and pull-forward are cheap to unwind but the \$1.24M reroute is largely sunk, which is why it sits above the ceiling. The T-12h exception shows the fail-safe default: the SLA lapsed and the charter was **rejected**.

Evaluation and Open Problems

This section sets out how the thesis of §1.6 could be confirmed or refuted. It names six open research problems, states what the reference code and proofs establish, and defines the measures a pilot would report. Each problem can be studied without first accepting the whole architecture.

6.1 Open research problems

PROBLEM	QUESTION	FIRST ARTIFACT	EVIDENCE REQUIRED	RELATED
Business semantics	<i>How do independently developed agents share precise meanings for obligations, assets and actions?</i>	A minimal domain model and action contract	Two agents built by different teams complete the test workflow using only the published contract	§1.4, §2.3, §4.6 · Q5
Authority and control	<i>How does delegated human authority remain enforceable across an agent workflow?</i>	Intent and capability specifications	Every action in the workflow traces to a signed intent; attenuation test vectors pass; actions outside the delegated limits are refused and escalated	§2.6–2.7, §4.3, §5.2 · Q7
Execution integrity	<i>What happens when an external action succeeds but acknowledgement or logging fails?</i>	Failure model and recovery protocol	Fault-injection runs (lost acknowledgement, duplicate delivery, ledger unavailable) end with no duplicated payment and no unrecorded external effect	§2.12, §3.5, §4.7 · Q8, Q14
Accounting and evidence	<i>What evidence supports completeness, accuracy and control effectiveness?</i>	One end-to-end finance workflow with testable invariants	Invariants I1–I9 hold under test; an auditor can reperform sampled controls from receipts and replay; the workflow reconciles to an existing ledger	§2.10, §2.17, §3 · Q1–Q4
Human oversight	<i>Which decisions require intervention, and how does oversight remain effective at scale?</i>	An escalation model and evaluation criteria	Measured escalation rate, reviewer time per exception and override rate on the test workflow, against criteria agreed in advance	§5.1 · Q5–Q7
Identity and trust	<i>How do credentials held by people and organizations become enforceable authority for agents, and evidence a bank or auditor accepts?</i>	A delegation profile linking wallet credentials, intent signatures and workload identity	Approvals in the test workflow use wallet-verified signatures, and a third party verifies a payment's authority chain offline	§2.18–2.19, §4.3 · Q15

TABLE 6.1 – Open research problems. Q-numbers refer to the open research questions at the end of the paper.

The problems meet in one finance workflow, supplier invoice to payment, which also yields the measurements Appendix D needs.

06 EVALUATION AND OPEN PROBLEMS

6.2 What the proofs and reference code establish

Appendix A contains reference code for the ledger and agent primitives, written in Python and Rust and tested against each other. Appendix B contains proof sketches for the Causal State Ledger. Both concern individual primitives. The guarantees a working system must give depend on how those primitives are deployed and operated together, and those remain untested. TABLE 6.2 separates what each primitive illustrates from what is still untested.

PRIMITIVE	WHAT IS ILLUSTRATED	SYSTEM-LEVEL GUARANTEE STILL UNTESTED
Canonical encoding and CIDs (A.1)	Deterministic DAG-CBOR encoding checked against RFC 8949 vectors; identifiers stable and sensitive to any change	Schema evolution across independently developed components; performance at production event volumes
Merkle log and receipts (A.2, A.4, A.5; App. B)	RFC 9162-shaped tree, inclusion proofs for every size tested, tamper detection; Python and Rust agree; proof sketches for tamper-evidence, inclusion, append-only consistency and the forgery bound	Completeness of logging (I1) under partial failure; witness operation and resistance to split views in practice; key management and signing ceremony
Intent signature (A.2, A.5)	Tampering with a signed intent is detected (HMAC standing in for the production signature schemes)	Binding of intents to real human identities and step-up authentication; revocation timing
APO lifecycle (A.3)	State machine transitions and their chained ledger events, including suspension	Behavior of real agents and models; reliability and error rates of agent proposals (Q5)
Policy guardrail (A.3)	A pre-execution check that blocks an action outside its limits and records the decision	Wasm sandbox isolation and determinism; whether policies implement the controls as designed; latency within the budget of \$3.9
Capability attenuation (A.4)	A delegated token can narrow but never widen its parent's authority (Rust)	Enforcement across the gateway, eBPF and MCP servers together; behavior under clock skew and token replay

TABLE 6.2 – Scope of the proofs and reference code. Nothing in Appendices A and B has been security-audited.

Several properties the paper relies on have no illustration yet: the Accounting Kernel's balancing and numbering invariants under concurrent load, saga recovery when an external action's outcome is unknown, erasure by crypto-shredding (§2.15), Semantic Data Plane throughput, and the acceptance of the evidence by auditors and regulators. Each one belongs to a problem in TABLE 6.1.

6.3 Measures for a pilot

The economic case in Appendix D and several design choices rest on quantities that have not been measured for this kind of system. A pilot of the supplier-invoice-to-payment workflow, run beside an existing ERP, would be instrumented to report them, so that each economic hypothesis can be tested against data.

MEASURE	DEFINITION	WHAT IT TESTS	RELATED
Work absorbed	Share of workflow steps completed by agents with no human handling	The largest driver of the economic result (FIG. D.4)	Q12
Exception rate	Escalations to a person per 1,000 agent actions	Size of the exception desk; whether oversight scales	Q6
Reviewer time	Minutes from escalation to a recorded decision	Whether approval stays a real review	Q6
Override rate	Share of agent proposals a reviewer changes or rejects	Agent reliability; when an authority ceiling may rise	Q5
Control latency	Median and 99th-percentile time added per action by identity, policy and ledger append	The latency budget of \$3.9	—
Unrecorded effects	External effects with no matching ledger record after fault injection (target zero)	Invariant I1 and the failure model	Q14
Reconciliation	Unexplained differences above tolerance per close (Eq. 2.16)	Readiness to take responsibility from the existing ERP	Q13
Binding effort	Hours to connect the workflow to each external system	The integration parameter δ (§D.2)	Q11

TABLE 6.3 – Measures a pilot would report.

■ REPORTING OF EVIDENCE

Evaluation evidence should be published with its data and scripts: benchmark results, fault-injection reports, reconciliation results and auditor feedback. Where evidence contradicts a design choice, the choice should change and the reason be recorded.

Conclusion

7.1 Summary of the argument

Enterprise information systems were built for people entering transactions. Autonomous software agents can now do much of that work, and the systems they work in were not designed to tell what authority an agent holds, why it acted or whether its record can be trusted. This paper asked what such systems should become when agents are their primary operators.

Its thesis is that agents can take on most digital operational work without loss of control or auditability, provided that identity, business semantics, policy enforcement, transaction integrity and evidence are supplied once, as a common foundation beneath every agent. The paper defines agent-native systems (C1) and proposes a reference architecture (C2), a delegation model from a person's credential to an agent's action (C3), a causal evidence ledger (C4), accounting invariants with a control mapping (C5) and the evaluation that would test them (C6).

The evidence in this paper supports only the primitives: the ledger, receipts and capability attenuation behave as specified in reference code and proof sketches. Whether the whole system gives the guarantees the thesis requires remains to be shown in a working organization, and Section 6 sets out how that would be done.

7.2 Implications

READER	WHAT THE ARGUMENT IMPLIES
Finance and audit leaders	Agents already act in finance systems through accounts and screens designed for people. The questions an auditor will ask (who authorized the action, was the agent allowed to act, can the record be trusted) can be answered with evidence produced at the time of the action, on today's ERP, before any replacement is considered.
Architects and vendors	Agent support added to existing software leaves authority, outcomes and uncertainty to each agent. Making those contracts explicit once, beneath every agent, is a design choice that existing platforms can adopt in part, beginning with workload identity and receipts for agent actions.
Regulators and standards bodies	Logging duties in the EU AI Act, wallet acceptance under eIDAS 2.0 and payee verification in payments regulation meet in the same transaction. A delegation record that links a person's credential to an agent's workload identity is a candidate common evidence format for all three.
Researchers	The six problems in TABLE 6.1 and the fifteen questions at the end of the paper are open. Each can be studied separately, and most need data from working organizations, which design research in information systems is well placed to collect.

TABLE 7.1 – Implications by reader.

07 CONCLUSION

7.3 Limitations

- **No naturalistic evaluation.** The architecture has not run in a working organization. The evaluation in this paper is limited to test vectors, agreement between two implementations and proof sketches, which concern individual primitives (TABLE 6.2).
- **A single author.** The design reflects one practitioner's view, shaped by financial audit, technology assurance and digital trust work. That background may favor evidence and control over other qualities, such as usability or speed of change, that other disciplines would weigh more heavily.
- **Modeled economics.** The figures in Appendix D are projections from stated assumptions, and the largest driver, the share of finance work agents absorb, has not been measured.
- **Candidate technologies.** The named technologies are candidates (TABLE 1.2), and the field is moving fast. The thesis concerns the foundation's properties, which could be met by other technologies.
- **Regulatory change.** Dates for eIDAS 2.0 wallets, the European Business Wallet and the Payment Services Regulation are taken from current law and proposals, and some may change.
- **Scope.** ERP and EU regulation serve as the worked examples. Other enterprise domains and other jurisdictions are argued by analogy only.

7.4 Closing remarks

Organizations will hand operational work to software agents when they can show who authorized each action, that the action stayed within its limits and that the record of it can be trusted. Those are the questions accountants and auditors have always asked of any system that keeps the books.

This paper argues that the answers belong in the foundation of the system rather than in each agent or each application. If the thesis holds, the design of enterprise information systems shifts from screens and transactions built for people toward explicit contracts, delegated authority and verifiable evidence built for agents working under human direction. Section 6 sets out how that claim can be tested.

7.5 About the author

Darren Lentz CA(SA), CISA is a chartered accountant and certified information systems auditor working where financial assurance, cybersecurity and digital trust meet. From 2016 to 2022 he managed financial statement audits in South Africa and Australia across mining, automotive and medical-device manufacturing, healthcare, construction, hospitality, investment management, technology and the public sector, at organizations running the major ERP platforms. He then moved into technology assurance and cybersecurity: general IT controls, SOC 2 and SOC 3 reporting, identity and access management reviews, and WebTrust assurance of certification authorities, including cloud HSM and key-management environments. He now leads a digital trust practice at a global professional services firm, advising on public key infrastructure, post-quantum cryptography and digital public infrastructure, including governance and trust frameworks and PKI architecture for national-scale digital identity and credential issuance. He has published on post-quantum cryptography and digital trust, and spoke at ID4Africa 2026 on cloud security in national identity systems. He holds BCom (Rationum) and BCom Honours degrees in computer science, information systems and accounting from NMMU.

Disclosure. This paper is written in a personal capacity. The views in it are the author's own and do not represent those of his employer or its clients. The author may offer commercial products built on the open framework described here, subject to his employment obligations.

A APPENDIX A • REFERENCE CODE

Reference Code

The primitives of §3–§4, implemented twice (Python and Rust) and tested against each other. Essential parts are printed here; the full runnable sources are in the accompanying `code/` folder.

This code is a **design illustration**: a reduced subset of the §3 event schema, with HMAC-SHA-256 standing in for the ECDSA, Ed25519 and Biscuit signatures of §3–§4 (leaf signatures omitted) so that both packages have **zero third-party dependencies** (Python stdlib; Rust `std` only, SHA-256 written from FIPS 180-4 and checked against its test vectors). All code shown was executed; results are on page A3.

FILE	LINES	CONTENTS
python/aeos_primitives.py	414	encoding, CID, events, log, APO, guardrail
python/test_primitives.py	190	13 tests incl. tamper + cross-language
rust/src/*.rs	535	SHA-256, HMAC, DAG-CBOR, events, log, tokens
rust/tests/cross_lang.rs	31	asserts the Python fixture values

SHARED CONVENTIONS

- **Encoding:** DAG-CBOR subset: shortest ints, keys length-first then bitwise, tag-42 links, no floats^[106,146]
- **CID:** $v1 \cdot \text{dag-cbor} \cdot \text{sha2-256}$, base32^[103]
- **Leaf:** SHA-256($0x00 \parallel \text{CID}$), binary CID, as in §3.2 and FIG. 3.1
- **Node:** SHA-256($0x01 \parallel L \parallel R$)^[147]
- **Proofs:** RFC 9162 §2.1.3.2

CROSS-LANGUAGE CONFORMANCE (PYTHON AND RUST)

Both suites build the same 5-event chain (each event citing its predecessor's CID) and assert these values, so encoding, CIDs and the RFC 6962-style $0x00/0x01$ hashing agree byte for byte:

```
first_cid = bafyreicskzgnaj4z45ijmjj2n2deiyrtuqtckeqhvcd7yghiq4i2ymbwlm
root(5) = effe1ddfa97d26a234e302ba421506e7e0d535a33ec3187ea59833af34f31432
```

A.1 Canonical encoding, CIDs and the Causal Event

LISTING A.1 // PYTHON/AEOS_PRIMITIVES.PY (EXCERPT)

```
class Link(bytes):

def dag_cbor(v: Any) -> bytes:
    if isinstance(v, Link):
        body = b"\x00" + bytes(v)
        return b"\xd8\x2a" + _head(2, len(body)) + body
    if v is None:
        return b"\xf6"
    if v is True:
        return b"\xf5"
    if v is False:
        return b"\xf4"
    if isinstance(v, int):
        return _head(0, v) if v >= 0 else _head(1, -1 - v)
    if isinstance(v, (bytes, bytearray)):
        return _head(2, len(v)) + bytes(v)
    if isinstance(v, str):
        b = v.encode("utf-8")
        return _head(3, len(b)) + b
    if isinstance(v, (list, tuple)):
        items = b"".join(dag_cbor(x) for x in v)
        return _head(4, len(v)) + items
    if isinstance(v, dict):
        keys = sorted(v, key=_dag_key_order)
        body = b"".join(dag_cbor(k) + dag_cbor(v[k])
                        for k in keys)
        return _head(5, len(v)) + body
    raise TypeError(f"unsupported: {type(v).__name__}")

def _dag_key_order(k: str) -> Tuple[int, bytes]:
    b = k.encode("utf-8")
    return (len(b), b)
```

LISTING A.2 // PYTHON/AEOS_PRIMITIVES.PY (EXCERPT)

```
# CIDv1 = version 0x01, codec dag-cbor 0x71, multihash sha2-
256 0x12, len 0x20
CID_PREFIX = bytes([0x01, 0x71, 0x12, 0x20])

def cid_bytes(obj: Any) -> bytes:
    digest = hashlib.sha256(dag_cbor(obj)).digest()
    return CID_PREFIX + digest
...

class CausalEvent:
    kind: str
    actor: str
    parents: Tuple[bytes, ...]
    payload: Dict[str, Any]
    ts: int

    def __post_init__(self) -> None:
        if self.kind not in EVENT_KINDS:
            raise ValueError(f"unknown kind {self.kind}")

    def to_obj(self) -> Dict[str, Any]:
        return {"kind": self.kind, "actor": self.actor,
                "parents": [Link(p) for p in self.parents],
                "payload": self.payload, "ts": self.ts}

    def encode(self) -> bytes:
        return dag_cbor(self.to_obj())

@property
def cid(self) -> bytes:
    return cid_bytes(self.to_obj())
```

Encoding vectors are checked in both languages ($1000 \rightarrow 1903e8$, $-100 \rightarrow 3863$, $\{"a":1, "b":[2,3]\} \rightarrow a26161016162820203$, key order $\{"z", "aa"\}$), and a tag-42 link must begin $d82a58250001711220$. Changing one bit of any field, including the timestamp, changes the CID.

A.2 Merkle log and Causal Receipts

A `CausalEvent` (Listing A.2) is named by its CID and its `parents` are CIDs, so the event graph is a Merkle-DAG. Each event's 36-byte CID is also the leaf input of one log, `leaf = SHA-256(0x00 || CID)` as in §3.2, so the tree hash commits to the whole history. Proofs follow RFC 9162; a verifier needs only the receipt, not the log. Listing A.5 is the APO's closed transition table (§4.2).

LISTING A.3 // PYTHON/AEOS_PRIMITIVES.PY (EXCERPT)

```
def leaf_hash(data: bytes) -> bytes:
    return hashlib.sha256(b"\x00" + data).digest()

def node_hash(left: bytes, right: bytes) -> bytes:
    return hashlib.sha256(b"\x01" + left + right).digest()

def _split(n: int) -> int:
    k = 1
    while k << 1 < n:
        k <= 1
    return k

def _mth(leaves: List[bytes]) -> bytes:
    n = len(leaves)
    if n == 0:
        return hashlib.sha256(b"").digest()
    if n == 1:
        return leaves[0]
    k = _split(n)
    return node_hash(_mth(leaves[:k]), _mth(leaves[k:]))

def _path(m: int, leaves: List[bytes]) -> List[bytes]:
    n = len(leaves)
    if n <= 1:
        return []
    k = _split(n)
    if m < k:
        return _path(m, leaves[:k]) + [_mth(leaves[k:])]
    return _path(m - k, leaves[k:]) + [_mth(leaves[k:])]
```

LISTING A.4 // PYTHON/AEOS_PRIMITIVES.PY (EXCERPT)

```
def verify_inclusion(r: CausalReceipt) -> bool:
    if r.leaf_index >= r.tree_size:
        return False
    fn, sn = r.leaf_index, r.tree_size - 1
    h = leaf_hash(r.leaf_cid)
    for p in r.audit_path:
        if sn == 0:
            return False
        if fn & 1 or fn == sn:
            h = node_hash(p, h)
            if not fn & 1:
                while fn and not fn & 1:
                    fn, sn = fn >> 1, sn >> 1
        else:
            h = node_hash(h, p)
            fn, sn = fn >> 1, sn >> 1
    return sn == 0 and hmac.compare_digest(h, r.root)
```

LISTING A.5 // PYTHON/AEOS_PRIMITIVES.PY (EXCERPT)

```
TRANSITIONS: Dict[Tuple[State, str], State] = {
    (State.PROPOSED, "attest"): State.ATTESTED,
    (State.ATTESTED, "activate"): State.ACTIVE,
    (State.ACTIVE, "suspend"): State.SUSPENDED,
    (State.SUSPENDED, "resume"): State.ACTIVE,
    (State.PROPOSED, "retire"): State.RETIRED,
    (State.ATTESTED, "retire"): State.RETIRED,
    (State.ACTIVE, "retire"): State.RETIRED,
    (State.SUSPENDED, "retire"): State.RETIRED,
}
```

A.3 APO lifecycle and the policy guardrail

Illegal edges raise before a byte is logged; each legal transition appends a `Lifecycle` event whose parent is the previous transition, so an APO's history is a hash chain inside the DAG. The simple `PolicyGuardrail` (Listing A.7) returns `ALLOW` | `DENY` | `ESCALATE` and logs the decision as a `PolicyDecision` event linked to the Intent Envelope.

LISTING A.6 // PYTHON/AEOS_PRIMITIVES.PY (EXCERPT)

```
def transition(self, action: str, log: MerkleLog, ts: int,
               evidence: Optional[Dict[str, Any]] = None
               ) -> CausalReceipt:
    nxt = TRANSITIONS.get((self.state, action))
    if nxt is None:
        raise LifecycleError(
            f"{action} not allowed in {self.state.value}")
    ok_svid = (evidence or {}).get("svid_verified")
    if action == "attest" and not ok_svid:
        raise LifecycleError("attest needs SVID evidence")
    body = {"from": self.state.value, "to": nxt.value,
           "action": action, "evidence": evidence or {}}
    prev = tuple(self.history[-1:]) # last transition
    ev = CausalEvent("Lifecycle", self.spiffe_id, prev,
                    body, ts)
    receipt = log.append(ev)
    self.state = nxt
    self.history.append(ev.cid)
    return receipt
```

LISTING A.7 // PYTHON/AEOS_PRIMITIVES.PY (EXCERPT)

```
def evaluate(self, apo: APO, intent: IntentEnvelope,
             req: ActionRequest, now: int
             ) -> Tuple[str, List[str]]:
    deny: List[str] = []
    if apo.state is not State.ACTIVE:
        deny.append("agent-not-active")
    if req.capability not in apo.manifest.capabilities:
        deny.append("capability-not-in-manifest")
    if req.tool not in apo.manifest.tools:
        deny.append("tool-not-in-manifest")
    if now >= intent.expires_at:
        deny.append("intent-expired")
    if req.counterparty in SANCTIONED:
        deny.append("sanctions-hit")
    if req.created_by and req.created_by == req.actor:
        deny.append("sod-violation")
    limit = min(intent.budget_minor,
                apo.manifest.spend_budget_minor)
    if apo.spent_minor + req.amount_minor > limit:
        deny.append("budget-exceeded")
    if deny:
        return "DENY", deny
    if req.amount_minor > intent.authority_ceiling:
        return "ESCALATE", ["strategic-exception"]
    return "ALLOW", []
```

A.4 Rust: receipt verification and capability attenuation

Crate `aeos-primitives` (no dependencies) mirrors the Python log and adds §4.3 tokens: `attenuate` needs no root key; `verify` re-derives the chain, then checks each caveat.

LISTING A.8 // RUST/SRC/MERKLE.RS (EXCERPT)

```
/// RFC 9162 s.2.1.3.2 inclusion-proof verification.
pub fn verify_inclusion(r: &Receipt) -> bool {
    if r.leaf_index >= r.tree_size { return false; }
    let (mut fnn, mut sn) = (r.leaf_index, r.tree_size - 1);
    let mut h = leaf_hash(&r.leaf_cid);
    for p in &r.audit_path {
        if sn == 0 { return false; }
        if fnn & 1 == 1 || fnn == sn {
            h = node_hash(p, &h);
            if fnn & 1 == 0 {
                while fnn != 0 && fnn & 1 == 0 {
                    fnn >>= 1;
                    sn >>= 1;
                }
            }
        } else {
            h = node_hash(&h, p);
        }
        fnn >>= 1;
        sn >>= 1;
    }
    sn == 0 && ct_eq(&h, &r.root)
}
```

LISTING A.9 // RUST/SRC/CAPABILITY.RS (EXCERPT)

```
pub fn attenuate(&self, c: Caveat) -> Token {
    let sig = hmac_sha256(&self.sig, &c.bytes());
    let mut caveats = self.caveats.clone();
    caveats.push(c);
    Token { id: self.id.clone(), caveats, sig }
}
// ...
pub fn verify(&self, root_key: &[u8], req: &Request)
-> Result<(), Denied> {
    let mut s = hmac_sha256(root_key, self.id.as_bytes());
    for c in &self.caveats {
        s = hmac_sha256(&s, &c.bytes()); // re-derive chain
    }
    if !ct_eq(&s, &self.sig) {
        return Err(Denied::BadSignature);
    }
    // signature ok: every caveat must hold (conjunction)
    match self.caveats.iter().position(|c| !c.holds(req)) {
        Some(i) => Err(Denied::CaveatFailed(i)),
        None => Ok(()),
    }
}
```

A.5 Tamper detection test and recorded results

LISTING A.10 // PYTHON/TEST_PRIMITIVES.PY (EXCERPT)

```
def test_tamper_detection():
    log = MerkleLog()
    for i in range(10):
        log.append(_event(i))
    r = log.prove(6)
    assert verify_inclusion(r)
    forged = dataclasses.replace(_event(6), payload={"tool": "po.submit", "req": b"\x00" * 32})
    assert not verify_inclusion(dataclasses.replace(r, leaf_cid=forged.cid)) # altered event
    bad = list(r.audit_path); bad[1] = bytes(32)
    assert not verify_inclusion(dataclasses.replace(r, audit_path=bad)) # altered sibling
    assert not verify_inclusion(dataclasses.replace(r, leaf_index=5)) # wrong position
    assert not verify_inclusion(dataclasses.replace(r, root=log.root(9))) # stale root
```

\$ CD CODE/PYTHON && PYTHON3 -M PYTEST -V

```
test_dag_cbor_rfc8949_vectors PASSED
test_map_key_order_is_canonical PASSED
test_cid_is_stable_and_content_sensitive PASSED
test_empty_root_is_sha256_of_empty_string PASSED
test_rfc6962_shape_for_three_leaves PASSED
test_inclusion_proofs_all_sizes PASSED
test_tamper_detection PASSED
test_dag_rejects_dangling_parent_and_is_idempotent PASSED
test_intent_signature_tamper PASSED
test_lifecycle_happy_path_and_chain PASSED
test_budgets PASSED
test_policy_allow_deny_escalate PASSED
test_cross_language_fixture PASSED
===== 13 passed in 0.10s =====
```

\$ CD CODE/RUST && CARGO TEST

```
cbor::length_first_key_order ok
cbor::tag42_link_prefix ok
capability::stripping_or_editing_caveats_breaks_signature ok
cbor::rfc8949_appendix_a_vectors ok
capability::attenuation_narrows_and_never_widens ok
merkle::dangling_parent_rejected ok
merkle::tamper_is_detected ok
sha256::rfc4231_hmac_case_2 ok
sha256::padding_boundaries_match_parts ok
sha256::fips_180_4_vectors ok
merkle::all_proofs_verify_for_sizes_1_to_40 ok
test result: ok. 11 passed; 0 failed; finished in 0.32s
python_and_rust_agree ok
test result: ok. 1 passed; 0 failed; finished in 0.00s
```

COVERAGE OF THE 25 TESTS (13 PYTHON, 12 RUST)

SHA-256 matches FIPS 180-4 vectors, HMAC matches RFC 4231; every proof verifies for every leaf of trees of 1-33 (Py) and 1-40 (Rust) leaves; altered events, siblings, positions, truncated paths and stale roots are rejected; illegal transitions, overruns, sanctions hits, SoD violations and expired intents are refused; edited caveats void a token.

Run 2026-10-12 · Python 3 + pytest · rustc/cargo 1.97.0 · Linux x86-64. Reproduce: `python3 -m pytest -v` (or `python3 test_primitives.py`) in `code/python`; `cargo test` in `code/rust`. Full logs: `code/*/test_output.txt`.

B APPENDIX B • PROOF SKETCHES

Proof Sketches for the Causal State Ledger

These sketches cover one primitive: the Merkle log of §3.3–3.6, with checkpoints signed by the operator and cosigned by witnesses. They show that, given a collision-resistant hash function and unforgeable signatures, a logged event cannot be altered, removed from the middle of the log or reordered without detection by anyone holding a receipt and a consistent checkpoint. The reference code in Appendices A.2, A.4 and A.5 implements the same tree and its proofs, and its test suite checks them at every log size it covers.

They do not establish the system-level guarantees that depend on how the log is deployed and operated. TABLE 6.2 lists those; each belongs to an open problem in TABLE 6.1.

Setting. $H = \text{SHA-256}$. For an ordered list $D = (d_0, \dots, d_{n-1})$, MTH is defined as in RFC 9162^[32]: $\text{MTH}((d)) = H(0x00\|d)$, and for $n > 1$ with k the largest power of two $< n$, $\text{MTH}(D) = H(0x01\|\text{MTH}(D[0:k])\|\text{MTH}(D[k:n]))$. In AE-OS each d_i is the 36-byte CID of event i (§3.2). A *collision* is a pair $x \neq y$ with $H(x) = H(y)$. Proofs are sketches at the standard level of rigor for hash-tree constructions^[100].

Theorem 1 — Tamper-evidence

If $D \neq D'$ and $\text{MTH}(D) = \text{MTH}(D')$, a collision in H can be extracted in $O(n)$ hash evaluations.

Sketch. Walk down from the roots $H(x) = H(x')$. If $x \neq x'$, that is a collision. Otherwise the first bytes agree, so both are leaves or both interior (domain separation). Equal leaves would mean $d = d'$, impossible on a differing subtree; equal interior inputs give equal child hashes, and at least one child pair covers different lists: recurse. A size mismatch eventually pairs a leaf with an interior node, whose inputs differ in byte 0: a collision. At most $2n-1$ nodes are visited. ■

$$\Pr[\text{forge}] \leq \text{Adv}_{\text{H}}^{\text{coll}}(t + O(n)) \quad (\text{Eq. B.1})$$

Theorem 2 — Inclusion soundness & size

If `verify_inclusion` (§3.6) accepts (i, n, ℓ, π, r) with $r = \text{MTH}(D)$ and $\ell \neq H(0x00\|d_i)$, a collision is found along the path. *Sketch.* The verifier recomputes the same sequence of $H(0x01\|\cdot\|\cdot)$ calls as the honest tree, replacing one input. Compare from the root down: the first level where honest and recomputed inputs differ yet outputs agree is a collision. $|\pi|$ equals the depth of leaf i , at most $\lceil \log_2 n \rceil$; verification costs $|\pi| + 1$ hashes. ■

$$|\pi| \leq \lceil \log_2 n \rceil \times 32 \text{ B} \quad (\text{Eq. B.2})$$

Corollary (localization). Given an honest copy, a verifier finds the altered leaf by comparing child hashes level by level, descending only into the mismatching child: $\lceil \log_2 n \rceil$ comparisons, as in FIG. 3.1 Panel B.

PROOF SCOPE

Covered: integrity, inclusion and ordering of logged events. Not covered: completeness (rests on I1, a platform control tested by auditors, §2.17), accuracy of payloads, correctness of policies, confidentiality (payloads sealed separately) and censorship of appends (mitigated by signed promises and witnesses). Protection against a forking operator assumes witness cosigning (§3.5).

Theorem 3 — Append-only consistency

Given checkpoints (m, r_m) and (n, r_n) , $m < n$, and a consistency proof that the RFC 9162 verifier accepts, the list committed by r_m is a prefix of the list committed by r_n , unless a collision is found. *Sketch.* The proof supplies the complete subtree hashes covering $[0, m)$ and the remaining nodes for $[m, n)$; the verifier rebuilds *both* roots from the same prefix nodes. By Theorem 1, two distinct prefixes cannot produce both r_m and those shared nodes. Size $\leq \lceil \log_2 n \rceil + 1$ nodes^[32]. An operator who rewrites history must therefore sign two checkpoints that fail this check: with EUF-CMA-secure Ed25519 (~128-bit level^[111]), that pair is non-repudiable evidence of misbehavior. ■

$$\text{accept} \Rightarrow D_m \sqsubseteq D_n \vee \text{coll}(H) \quad (\text{Eq. B.3})$$

Theorem 4 — Undetected-tampering bound

Model H as a random function with 256-bit output. An adversary making q hash queries finds a collision with probability $\leq q(q-1)/2^{257}$ (birthday bound); NIST rates SHA-256 collision resistance at 128 bits^[105]. Rewriting a *specific* leaf while keeping its hash (second preimage) is bounded by $q/2^{256}$. With $q = 2^{64}$:

$$\Pr[\text{miss}] < 2^{-129} + \epsilon_{\text{sig}} \quad (\text{Eq. B.4})$$

where ϵ_{sig} is the Ed25519 forgery advantage at the ~128-bit level: overall on the order of 2^{-128} , provided at least one verifier checks receipts against a consistent checkpoint (witness cosigning, §3.5). It covers modification and reordering of logged events, not omission *before* logging, which I1 prevents as a platform control (§2.17).

C APPENDIX C • RESEARCH AND IMPLEMENTATION SCOPE

Research and Implementation Scope

The research scope is the 95 capabilities of a multinational ERP estate. Each row is an identified requirement; its status shows whether this paper specifies the contract.

The 95 capabilities are those a multinational ERP estate typically runs, grouped into six areas. Each row is rated against the contracts this paper defines. Reference code covers only ledger and agent primitives (TABLE 6.2). Owner codes: E1–E6 enterprise layers; L1–L4 the four planes; LOC Localization & Regulatory Framework; RPL Records & Privacy Lifecycle; CXB Coexistence Bridge; ASR assurance model (§2.17). With two owners, the first carries the business logic.

- SPEC

Contract specified. Data structures, events, invariants and pipeline path are defined here at a level an architect could build from.
- PACK

Requirement placed; contract left to a pack. The paper defines where the capability lives and how it books, calculates and proves. Business rules (an MRP algorithm, a country's payroll or VAT rules, a consolidation method) belong to a Domain Pack or Localization Pack that this paper does not specify.
- OPEN

Requirement identified; design open. This paper does not settle the design; the row note says what is missing.

LAYER	GROUP	ROWS	SPEC	PACK	OPEN
C.1	Accounting	38	13	25	0
C.2	Operational	18	0	17	1
C.3	Orchestration	11	9	2	0
C.4	Data & analytics	6	5	1	0
C.5	Platform	16	9	0	7
C.6	Transition	6	4	0	2
Total		95	40	45	10

PACK is the largest status by design: the kernel, engines, process runtime and evidence chain are specified once, and functional scope (P2P, payroll, MRP) arrives as versioned packs. A PACK row still needs a functional contract and an implementation before it could carry production work.

#	CAPABILITY	OWNER	SPEC	STATUS	BASIS IN THIS PAPER / REMAINING WORK
C.1 FINANCIAL STATEMENT / ACCOUNTING (38 capabilities, rows 1–38)					
1	Double-entry GL; balanced JE as atomic unit	E1	\$2.10	SPEC	JournalEntry in one entity × ledger partition; I7 balance
2	Chart of accounts (op., group, country), mapping	E1 · LOC	\$2.10, 2.14	SPEC	CID-pinned Account and CoA; country CoA in LOC packs
3	Multi-GAAP parallel ledgers (IFRS, US, local, tax)	E1	\$2.10	SPEC	Leading and parallel ledgers; valuation rules ship in packs
4	Posting periods, open/close, special periods	E1	\$2.10	SPEC	PostingPeriod state machine; PeriodTransition event; I9
5	Reversal, correction and adjusting entries	E1	\$2.10	SPEC	Corrections are reversals with reason codes, in open period
6	Legal numbering (gapless, per series, per country)	E1 · LOC	\$2.10, 2.14	SPEC	Gapless series issued at posting; country rules in LOC packs
7	AP: invoice verification, 3-way match, payment run	E4 · E1	\$2.13, 2.10	PACK	P2P pack; kernel open-item clearing; worked flow in §2.9
8	AR: open items, dunning, cash application	E4 · E1	\$2.13, 2.10	PACK	O2C Domain Pack; dunning on an E3 Schedule
9	Fixed assets: depreciation areas, impairment, CIP	E4 · E2	\$2.13, 2.11	PACK	Acquire-to-Retire pack; depreciation Calculation Module
10	Inventory valuation (standard, moving avg., FIFO)	E2 · E1	\$2.11, 2.10	PACK	Valuation modules; quantity and value in one JournalEntry
11	Leases (IFRS 16 / ASC 842)	E4 · E2	\$2.13, 2.11	PACK	Lease schedules and remeasurement as Calculation Modules
12	Revenue recognition (IFRS 15 / ASC 606)	E4 · E2	\$2.13, 2.11	PACK	SSP allocation engine; contract asset and liability postings
13	Accruals, deferrals, recurring entries, GR/IR	E3 · E2	\$2.12, 2.11	SPEC	Schedule plus Calculation Module; estimates need sign-off (I8)
14	FX revaluation, translation (IAS 21), CTA	E2 · E1	\$2.11, 2.10	PACK	CID-pinned rate snapshots; revaluation modules in R2R pack
15	Intercompany matching, netting, elimination	E1	\$2.10	SPEC	IC pairs share a correlation CID; netting rules in packs
16	Consolidation: eliminations, minorities, equity	E4 · E1	\$2.13, 2.10	PACK	Consolidation pack (own or partner) on the group ledger
17	Period close orchestration (tasks, sign-off)	E3 · E6	\$2.12, 2.9	SPEC	Close Process Definition; close cockpit; PeriodTransition
18	Reconciliations (bank, sub-ledger, intercompany)	E5 · E3	\$2.17, 2.12	SPEC	Reconciliation projections; breaks go to Human Tasks
19	Financial statements and disclosures (IFRS 18)	E5 · E4	\$2.17, 2.13	PACK	Statement models in R2R pack; Report Snapshot per issue
20	XBRL / ESEF tagging and filing	LOC	\$2.14	PACK	CoA-to-taxonomy mapping; filing through partner connector
21	Statutory local GAAP accounts and audit packs	LOC	\$2.14	PACK	Statutory generators per Localization Pack
22	Indirect tax (VAT/GST, sales tax, reverse charge)	E2 · LOC	\$2.11, 2.14	PACK	Wasm tax module or certified engine; CalculationExecution
23	E-invoicing and clearance (CTC) mandates	LOC · L2	\$2.14	PACK	Tax Document; authority ack as ExternalInput; Peppol/PA
24	SAF-T and digital reporting	LOC	\$2.14	PACK	Generators read kernel projections; one per jurisdiction
25	Direct tax, tax provision (IAS 12), deferred tax	E4	\$2.13	PACK	Tax pack with tax ledger; provision engine own or partner
26	Pillar Two / GloBE information return	E4	\$2.13	PACK	GloBE data pack fed from the group ledger
27	Transfer pricing documentation, CbCR	E4	\$2.13	PACK	Tax pack; TP adjustments post as IC pairs (row 15)
28	ESG / CSRD reporting and emissions accounting	E4 · E1	\$2.13, 2.10	PACK	ESG pack; non-monetary ledgers (tCO ₂ e) on the kernel
29	Controlling: cost centers, orders, allocations	E4 · E2	\$2.13, 2.11	PACK	Controlling pack; allocation cycles as Calculation Modules
30	Product costing, profitability (margin analysis)	E5 · E2	\$2.17, 2.11	PACK	Margin projection needed by §5.2 constraints; costing in pack
31	Budgeting, planning, forecasting (FP&A)	E5	\$2.17	PACK	Plan Version as non-legal parallel ledger; content in pack
32	Treasury: liquidity, hedging, hedge accounting	E4 · E2	\$2.13, 2.11	PACK	Treasury pack; hedge effectiveness as Calculation Module
33	Payments and bank connectivity (ISO 20022)	E4 · L2	\$2.13, 2.9	PACK	Bank MCP servers; camt statements as signed ExternalInput
34	Credit management (limits, exposure, blocks)	L1 · E4	\$2.3, 2.13	PACK	Escrow counters for exposure; scoring rules in O2C pack
35	GRC: SoD ruleset, emergency access, controls	L3 · ASR	\$3.7, 2.17	SPEC	Control-to-policy mapping; design-time SoD over manifests
36	Audit trail of transactions and approvals	L4	\$3.2–3.6, B	SPEC	Integrity evidence; amount accuracy from E2 receipts (I8)
37	Retention (8–10 yr), archiving, legal hold	RPL	\$2.15	SPEC	Retention Class per event kind; Legal Hold; periods from LOC
38	Fiscal data access for authorities (DE GoBD)	LOC	\$2.14	PACK	Export generators per Localization Pack

#	CAPABILITY	OWNER	SPEC	STATUS	BASIS IN THIS PAPER / REMAINING WORK
C.2 OPERATIONAL (18 capabilities, rows 39-56)					
39	Procure-to-pay: requisition to payment	E4	\$2.13, 2.9	PACK	P2P pack; traced end to end in the \$2.9 worked flow
40	Contracts and pricing (conditions, rebates)	E2 · E4	\$2.11, 2.13	PACK	Pricing procedures as Calculation Modules; rebates in pack
41	Order-to-cash: order, ATP, delivery, billing	E4 · L1	\$2.13, 2.3	PACK	O2C pack; ATP reserves stock through escrow counters
42	Plan-to-produce: MRP, BOM, routing, capacity	E4 · E2	\$2.13, 2.11	PACK	MRP/capacity solvers in E2; algorithms not specified here
43	Inventory and warehouse (bins, HUs, counts)	E4	\$2.13	PACK	WMS pack or partner over MCP; stock projection in L1
44	Transportation and logistics (carriers, freight)	E4	\$2.13	PACK	TMS partner over MCP; freight settlement via P2P pack
45	Global trade: customs, export control, sanctions	E4 · L3	\$2.13, 3.7	PACK	Sanctions screening is L3 policy; customs filings in pack
46	Plant maintenance / EAM	E4	\$2.13	PACK	Industry pack or partner
47	Quality management (CAPA, batch release, GxP)	E4 · E6	\$2.13, 5.1	PACK	Signed human approvals support Part 11-style e-signatures
48	Batch / serial traceability and recalls	E4 · L4	\$2.13, 3.2	PACK	Causal DAG carries provenance; batch ontology in pack
49	Service management (field service, warranty)	E4	\$2.13	PACK	Industry pack or partner
50	Project system (WBS, project accounting, POC)	E4 · E2	\$2.13, 2.11	PACK	Project pack; POC revenue via RevRec engine (row 12)
51	Product lifecycle (PLM, engineering change)	E4	\$2.13	PACK	Partner PLM over MCP
52	HR / HCM core (org, positions, employees)	E4 · L2	\$2.13, 2.6	PACK	Hire-to-Retire pack; drives joiner-mover-leaver for IDs
53	Payroll: gross-to-net, statutory filings, retro	E2 · LOC	\$2.11, 2.14	PACK	Certified engine over MCP; country rules per LOC pack
54	Real estate / lease-in management	E4	\$2.13	PACK	Partner; lease accounting via row 11
55	Master data governance: golden record, DQ	E3 · E6	\$2.12, 2.9	OPEN	Change control, 4-eyes specified; survivorship, DQ scoring not
56	Supplier/customer onboarding (KYC, bank details)	E4	\$2.13	PACK	KYC partners over MCP; unknown parties escalate (\$5.2)
C.3 ORCHESTRATION / PROCESS (11 capabilities, rows 57-67)					
57	Workflow and approvals (delegation, substitution)	E3 · E6	\$2.12	SPEC	Human Task for routine approvals; delegation by token (\$4.3)
58	Durable orchestration (long-running, compensation)	E3	\$2.12	SPEC	Process Definition CID; Saga Step compensation; ProcessStep
59	Segregation of duties (runtime)	L3	\$3.7	SPEC	Causal-graph SoD evaluated before execution
60	Delegated authority and spend limits	L2	\$2.7, 4.3	SPEC	Intent-bound capability tokens and APO budgets
61	Scheduling, batch, business calendars	E3	\$2.12	SPEC	Schedule under the process owner's standing Intent Envelope
62	Exception handling with SLA and fail-safe	E6 · L3	\$5.1	SPEC	Strategic Exception with SLA; fail-closed default
63	Human task screens for routine back-office work	E6	\$2.17, 2.12	SPEC	Work Surfaces generated from ontology and Process Definitions
64	Event bus and internal integration	L1 · E5	\$2.3, 2.9	SPEC	Causal Events plus CDC of projections to non-agent consumers
65	Agent-to-system integration	L2	\$4.6-4.8	SPEC	MCP gateway with tool pinning and threat controls
66	B2B / EDI (EDIFACT, X12, AS2, Peppol)	E4 · LOC	\$2.13, 2.14	PACK	Gateway partners over MCP; inbound EDI as ExternalInput
67	Partner networks (supplier collaboration)	E4	\$2.13	PACK	Network partners; cross-org receipts not specified
C.4 DATA & ANALYTICS (6 capabilities, rows 68-73)					
68	Operational reporting (open items, stock, TB)	L1 · E5	\$2.3, 2.9	SPEC	Projection catalog with watermark and versioned definitions
69	Data warehouse / lakehouse feed	E5	\$2.17	SPEC	CDC export of projections to open table formats
70	Planning models (driver-based, versions, what-if)	E5 · E4	\$2.17, 2.13	PACK	Plan Version primitive specified; driver models in FP&A pack
71	Embedded analytics and semantic metrics	E5	\$2.17	SPEC	Metric Definition as a CID-pinned object
72	Data lineage (transaction and metric)	L4 · E5	\$3.2, 2.9	SPEC	Report cell → metric → projection → events
73	Reproducibility (as reported vs restated)	E5 · L4	\$2.17, 3.5	SPEC	Report Snapshot CID anchored to a signed checkpoint
C.5 PLATFORM / NON-FUNCTIONAL (16 capabilities, rows 74-89)					
74	Multi-tenancy (isolation, noisy-neighbor control)	L2 · L4	\$2.6, 3.5	OPEN	Tenant in SPIFFE ID and log; isolation model not specified
75	Multi-entity / company-code model	E1 · E4	\$2.10, 2.13	SPEC	Legal entity in kernel; plant, sales org as pack ontology
76	Multi-currency	E1	\$2.10	SPEC	Currency triple (transaction, local, group) on every line
77	Multi-language (UI, documents, master-data texts)	E6	—	OPEN	Translation of UI, documents and master-data texts not specified
78	Localization framework (country versions)	LOC	\$2.14	SPEC	Localization Pack contract; country content per pack
79	Availability, DR, RPO / RTO	E3 · L4	\$2.12, 3.5	OPEN	Degraded-mode rules per process (E3); RPO/RTO not stated
80	Performance and throughput	L1 · L3	\$1.2, 3.9	SPEC	Hot-path budget specified; close-window batch untested
81	Security, IAM, non-human identity	L2	\$2.6-2.8	SPEC	SPIFFE/SVID, eBPF mTLS, capability tokens
82	Data residency (multi-region groups)	L3 · L4	\$3.7, 3.5	OPEN	Residency policy exists; regional logs and anchoring open
83	Privacy: GDPR erasure vs append-only	RPL	\$2.15	SPEC	Per-subject key shredding; keyed commitments; KeyShred
84	Archiving and legal hold	RPL	\$2.15	SPEC	Archive tiers that keep checkpoints verifiable
85	Observability / SRE	L1-L4	\$4.9	OPEN	Tetragon events and KPI tiles only; no SLO or telemetry model
86	Upgrade and release management	L3 · E4	\$4.5, 2.13	SPEC	APO, policy and pack CIDs; promotion is a Causal Event
87	Event-schema and ontology evolution	L1	\$2.3	OPEN	Ontology CID-pinned; upcasters and dual-run rebuild open
88	Inference cost and FinOps	L2 · E2	\$4.1, 2.11	SPEC	model_tokens budgets; deterministic paths use no inference
89	Model risk management (validation, drift)	L4 · ASR	\$4.1, 2.17	OPEN	model_ref pinned; validation evidence and drift monitoring open
C.6 TRANSITION (6 capabilities, rows 90-95)					
90	Data migration from SAP (balances, open items)	CXB	\$2.16	SPEC	GenesisEvent opening balances cite the SAP extract hash
91	Coexistence / strangler with system-of-record flag	CXB	\$2.16	SPEC	SAP as MCP server; CDC ingest; SoR flag per process
92	Cutover and parallel run with reconciliation	CXB · E3	\$2.16	SPEC	Reconciliation projections; cutover runbooks in E3
93	Auditor reliance: SOC 1 / ISAE 3402, ITGC, IPE	ASR	\$2.17	SPEC	ITGC and IPE mapped to evidence; SOC 1 report scope in \$2.17
94	Regulatory acceptance (certification, accreditation)	LOC · ASR	\$2.14, 2.17	OPEN	Depends on each authority; certification path not defined
95	People, change management, support model (AMS)	—	—	OPEN	Operating model and exception-desk staffing outside this paper

Spec column: section of this paper; B = Appendix B. Status describes the contracts in this paper only.

D APPENDIX D • ILLUSTRATIVE ECONOMIC MODEL

Illustrative Seven-Year Cost and Benefit Model

This appendix is a tool for testing economic hypotheses, and its outputs are projections. It asks what would have to be true for an agent-native system to pay for itself in an organization replacing SAP ECC. The model sets an SAP S/4HANA subscription program against an AE-OS program that pays no license fee, over seven years, for two company sizes and three scenarios. It counts what each route costs to build and to run, then adds the value AE-OS creates beyond what the SAP route delivers.

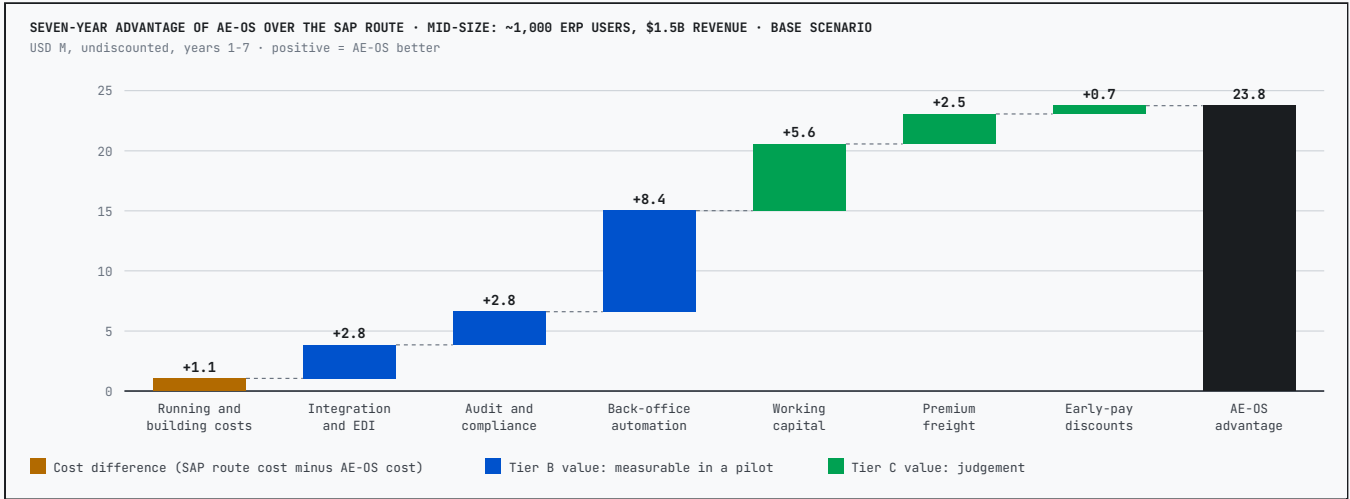


FIG. D.1 — MODELED Seven-year advantage of AE-OS over the SAP route for a company with about 1,000 ERP users and \$1.5B revenue, base scenario. The first bar is the difference in building and running cost; the blue bars are value a pilot can measure; the green bars depend on judgment. Source: code/python/econ_model.py.

D.1 How the model works

Three parts. *Cost to change* is the implementation program. *Cost to run* is everything paid each year after that: for SAP, the subscription plus application management; for AE-OS, hosting, a platform team, AI usage and a human exception desk. *Value created* counts only gains AE-OS delivers beyond the SAP route, after crediting SAP with its own automation.

Two company sizes. *Mid-size:* about 1,000 ERP users, \$1.5B revenue, 20 connected systems. *Large:* about 5,000 users, \$10B revenue and 40 connected systems.

Three scenarios. *Conservative* treats AE-OS as a first-of-its-kind platform: its program costs 10% more, every action calls an AI model and exceptions run at 0.5 per 1,000 actions. *Base* assumes AI-assisted delivery on both routes and routed Claude 5.5 models. *Optimistic* assumes a mature platform with open business packs.

Evidence tiers. Costs and tier B value (integration and EDI, audit and compliance, back-office automation) can be measured in a pilot. Tier C value (working capital, premium freight, early-payment discounts) depends on better decisions and is shown separately. Every figure is in US dollars, undiscounted unless marked NPV, which uses an 8% discount rate.

■ MODELED • THE RESULT IN BRIEF

For the mid-size company in the base case, AE-OS and the SAP route cost about the same to build and run: **\$27.1M against \$28.2M** over seven years. Running costs after go-live are **\$2.4M against \$2.5M a year**: AE-OS drops the \$1.75M subscription and spends it on hosting, a platform team, AI usage and an exception desk.

The difference comes from value created. Measurable gains add **\$14.0M**, of which back-office automation is \$8.4M; better decisions add a further \$8.7M. Net present value is **\$10.5M** on costs and measurable value, \$16.5M with all value.

In the conservative case AE-OS costs \$4.8M more and its measurable value only just covers that (NPV \$0.4M). The case therefore rests on the share of finance work agents absorb, which \$D.5 shows is the largest single driver and which a pilot must measure.

Services spending: IDC reports that SAP partners investing in their own services make \$8.81 for every \$1 SAP makes ^[148], and IT teams spend 39% of their time on custom integration ^[149].

D.2 Cost to change and cost to run

SAP route. The program costs \$12M for the mid-size company, inside the \$5–25M range consultants report for 500–2,000 users^[150], and \$53M for the large company, from a bottom-up estimate with a 1.45 overrun applied^[151]. The RISE subscription is \$1.75M a year for 1,000 users (reported \$1.5–2M after discounts) and \$6M a year for 5,000 users^[150]. Another adviser reports lower effective prices of €350–800 per user per year above 1,000 users^[152]; at those prices the SAP route is cheaper than shown. Application management, Basis and upkeep add 45% of the subscription once live, the midpoint of the reported 40–50%^[152].

AE-OS route. The program costs the same as the SAP route in the base case, because AI-assisted delivery lowers custom code, migration and testing effort on both routes while process redesign, change management and parallel runs stay people-paced. With no license, the enterprise pays for what a license would otherwise fund: a platform team of 6 (mid-size) or 20 (large) engineers that operates the system and keeps tax and legal rules current, cloud hosting, and an annual SOC 1 Type 2 report (\$2.17).

AI usage. Each agent action is priced at Claude 5.5 list prices^[153], assuming 3,000 input tokens (80% read from cache) and 300 output tokens. The base case runs 70% of steps as fixed rules and routes the rest by difficulty (TABLE D.2). Exceptions take 6 minutes each from finance staff at \$100k a year loaded, the average of US accountants and accounting clerks with benefits added^[154,155,156].

USD M, BASE SCENARIO	MID 7-YR	MID / YR*	LARGE 7-YR	LARGE / YR*
SAP ROUTE				
Implementation program	12.0	—	53.0	—
Subscription (RISE)	12.3	1.75	42.0	6.00
Application management, Basis, upkeep	3.9	0.79	13.5	2.70
SAP route total	28.2	2.54	108.5	8.70
AE-OS ROUTE (NO LICENSE)				
Implementation program	12.0	—	53.0	—
Hosting	2.9	0.45	11.7	1.80
Platform and pack team	7.6	1.08	25.2	3.60
SOC 1 Type 2 report	1.3	0.25	2.0	0.40
AI usage (Claude 5.5, routed)	1.1	0.22	5.7	1.09
Exception desk	2.3	0.43	11.3	2.17
AE-OS route total	27.1	2.43	108.9	9.07

TABLE D.1 — MODELED. *Per year once fully live (year 7). Mid: 1M agent actions a day; large: 5M (\$3.5). Hosting, team size and SOC 1 cost are assumptions.

PER AGENT ACTION	USD	MID / YR AT 1M A DAY
Opus 5.5 (\$4 in / \$20 out per 1M tokens)	0.0089	3.24
Sonnet 5.5 (\$2 / \$10)	0.0047	1.71
Haiku 5.5 (\$0.10 / \$0.50)	0.00023	0.09
Routed: 80% Haiku, 17% Sonnet, 3% Opus	0.00125	0.46
Base: 70% fixed rules, rest routed	0.0006	0.22

TABLE D.2 — Cost per action at list prices^[153]. The routing mix and the rules share are assumptions; Conservative uses the routed mix with no rules, Optimistic \$0.0003.

Integration upkeep. Hand-built interfaces grow with the number of system pairs: with N systems and a share ρ of pairs exchanging data there are $l = \rho \cdot N(N-1)/2$ interfaces, each costing h_b hours. AE-OS instead publishes one self-describing MCP server per system (h_a hours) plus a small residual δ per binding, so build cost is $(N \cdot h_a + l \cdot \delta \cdot h_b) \cdot r$ against $l \cdot h_b \cdot r$. At $\rho = 0.15$, $h_b = 400$ h, $h_a = 300$ h, $\delta = 0.10$ and $r = \$175/h$, upkeep at 20% a year falls from \$0.40M to \$0.25M for 20 systems and from \$1.64M to \$0.58M for 40. The build saving is already inside the program cost; the upkeep saving is counted as value in \$D.3.

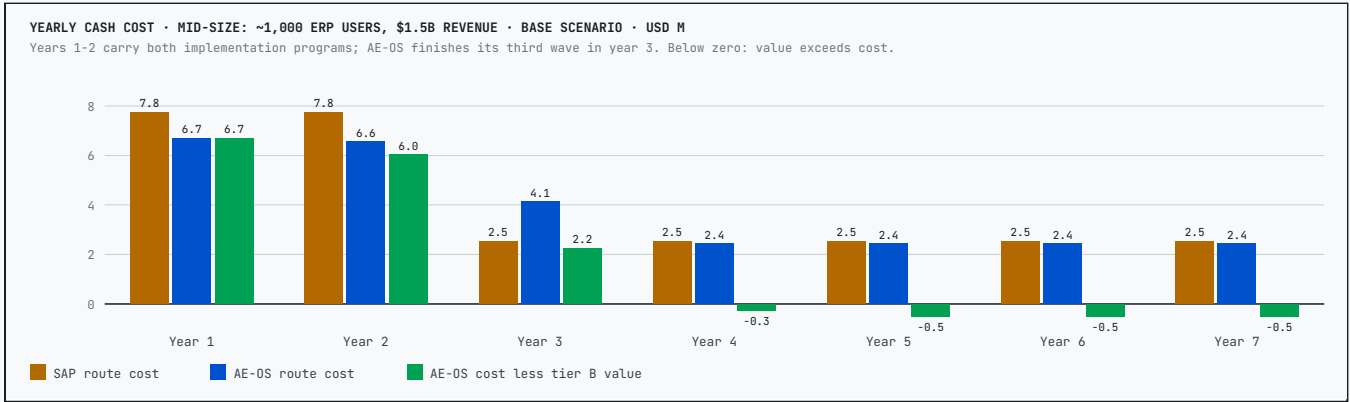


FIG. D.2 — **MODELED** Yearly cash cost of each route for the mid-size company, base scenario. The green bars subtract the measurable (tier B) value of \$D.3 from the AE-OS cost; from year 4 that value exceeds what AE-OS costs to run.

D.3 Value created beyond the SAP route

Each line below is a gain the SAP route does not deliver on its own. Values grow with the share of volume live and a realization curve (40% in year 2, 70% in year 3, 90% in year 4, full from year 5). The table gives the yearly value once fully realized.

VALUE LINE	HOW IT IS CALCULATED	KEY INPUTS AND SOURCES	MID / YR	LARGE / YR
TIER B • MEASURABLE IN A PILOT				
Integration and EDI	Trading-partner documents moved from value-added networks to Peppol or direct MCP exchange, less partner onboarding and map maintenance, plus the integration upkeep saving of \$D.2.	400 / 2,000 partners × 1,500 documents a year; network \$0.25 vs \$0.05 per document; onboarding \$2,500; map change \$600 (vendor price ranges, low confidence) ^[157,158] . Base: 60% of documents moved, 50% less partner effort.	0.54	2.99
Audit and compliance	External audit fee × reduction + internal SOX cost × reduction. Controls run before posting and evidence is produced as work happens (\$2.17), which cuts sampling and evidence gathering.	Audit fees 0.116% of revenue at \$1–5B, 0.057% above \$5B ^[159] ; internal SOX cost \$1.40M and \$1.93M ^[160] . Companies automating at least 25% of controls report 27% lower audit fees (correlation) ^[161] . Base: fees –10%, SOX cost –30%.	0.59	1.15
Back-office automation	(AE-OS share of finance work absorbed – SAP route share) × finance FTEs × loaded cost. The exception desk is already charged as a cost in TABLE D.1.	71.2 finance FTEs per \$1B revenue ^[162] ; \$100k loaded. One agent deployment in accounts payable reports 92% touchless invoices and half the processing staff ^[163] . Base: 25% for AE-OS, 8% for SAP.	1.82	12.10
TIER C • DEPENDS ON FASTER, BETTER DECISIONS				
Working capital	Days of sales outstanding saved × daily revenue × 8%, plus inventory reduced × 10% carrying cost.	Top performers collect 18 days faster than the median ^[164] ; carrying cost 10% a year ^[165] . Base: 2 days, inventory –3% (inventory 12% of revenue, assumed).	1.20	7.98
Premium freight	Freight spend × premium share × reduction from earlier re-planning (\$5.2).	Premium freight is 8% of freight at the median ^[166] . Base: freight 3% of revenue (assumed), premium freight –15%.	0.54	3.60
Early-payment discounts	Extra share of offered discounts captured × eligible purchases × discount rate.	Median company captures 65% of discounts offered ^[167] . Base: +10 points; purchases 50% of revenue, 10% offering 2% (assumed).	0.15	1.00

TABLE D.3 – MODELED value lines, base scenario, USD M per year once fully realized. Conservative sets all tier C lines to zero.

Fair comparison. The SAP route is credited with absorbing 8% of finance work through its own automation in the base case (10% conservative, 5% optimistic), so only the difference counts for AE-OS.

Double counting. Part of internal SOX cost is finance staff time, so the audit and automation lines could overlap; the reductions are set low to allow for this. Working capital is counted as the yearly financing benefit of the cash released, never as the cash itself.

Left out. The model counts no value for gains without a reliable figure: fewer reconciliation hours, avoided downtime (over 90% of mid-size and large firms put an hour of downtime above \$300k ^[168]), better data quality, faster close (one field study found AI-using accountants closed books 7.5 days faster ^[169]) and any revenue effect.

Realizing the benefit. Released finance capacity becomes value only when the company redeploys people or leaves vacancies unfilled. A company that keeps every role gets the cost and tier C lines but not the automation line.

D.4 Results by scenario

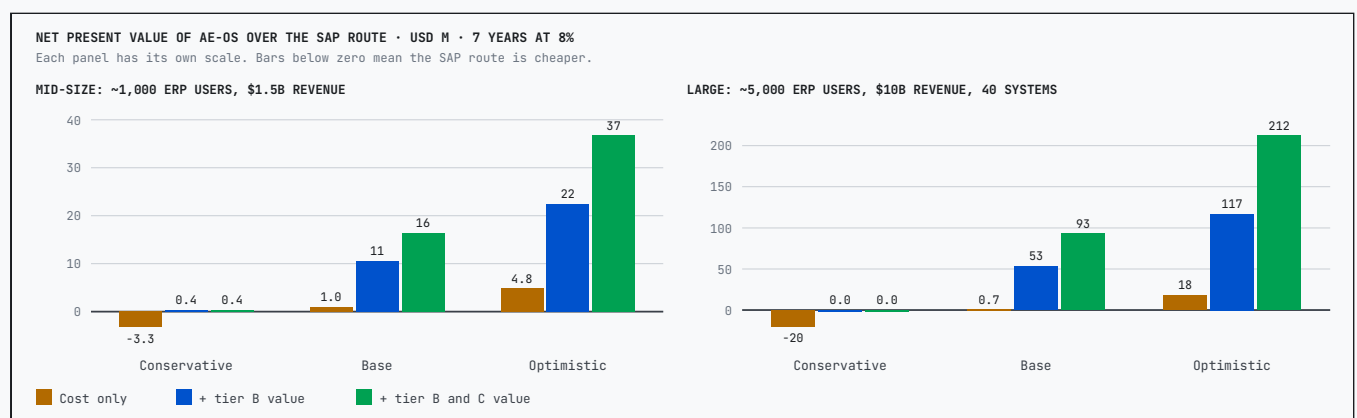


FIG. D.3 – **MODELED** Net present value of choosing AE-OS over the SAP route, seven years at 8%. Amber: cost only. Blue: adds measurable value. Green: adds decision-speed value.

USD M	CONS.	BASE	OPT.
MID-SIZE • SAP ROUTE 7-YR COST 28.2			
AE-OS 7-yr cost	33.0	27.1	22.4
Run cost / yr (SAP 2.54)	3.32	2.43	2.11
Value / yr, tier B + C	1.1	4.8	10.0
NPV, cost only	-3.3	1.0	4.8
NPV, cost + tier B	0.4	10.5	22.4
NPV, all value	0.4	16.5	36.7
LARGE • SAP ROUTE 7-YR COST 108.5			
AE-OS 7-yr cost	137.3	108.9	87.1
Run cost / yr (SAP 8.70)	13.51	9.07	7.43
Value / yr, tier B + C	6.0	28.8	61.1
NPV, cost only	-19.9	0.7	18.3
NPV, cost + tier B	0.0	53.2	117.0
NPV, all value	0.0	92.9	212.3

TABLE D.4 – MODELED. Run cost and value per year once fully live.

Reading the scenarios. On cost alone the two routes are close in the base case and AE-OS is cheaper only when the platform is mature. In the conservative case AE-OS costs more to run, mainly because every action calls an AI model and the exception desk is 2.5 times larger; its measurable value just pays for that.

Scale. Value grows with revenue and finance headcount, while the platform team and hosting grow more slowly, so the large company gains about five times as much as the mid-size one in the base case.

D.5 What drives the result

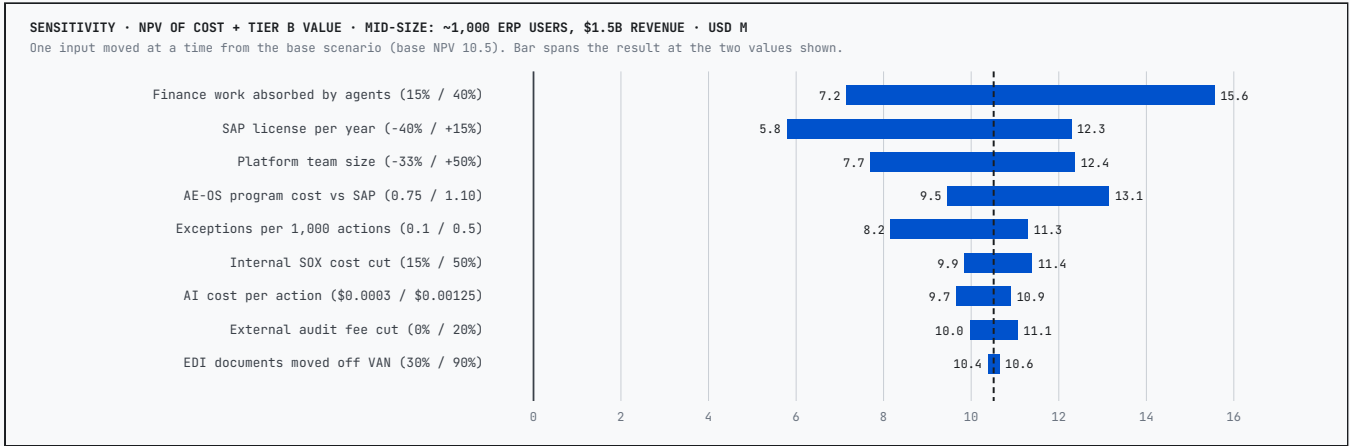


FIG. D.4 – MODELED Mid-size company, base scenario: net present value on costs and measurable value as each input moves alone between the two values shown. The dashed line is the base result.

The share of finance work agents absorb moves the result most, followed by the SAP subscription price the company would otherwise pay and the size of the AE-OS platform team. AI price per action, the input most often debated, moves the result by little more than \$1M either way because routing keeps it small. In every single-input case the result stays positive.

D.6 Risks to the result

- **Agents absorb less work.** In the conservative case, where agents absorb 15% of finance work against SAP's 10%, the mid-size advantage falls to near zero. A pilot must measure finance hours released per agent action.
- **The platform is not mature.** The zero-license route assumes open Accounting Kernel, Domain and Localization Packs exist. An early adopter that builds them for its own countries carries that cost alone, likely more than the subscription saved.
- **SAP prices lower.** At €350–800 per user per year^[152] the subscription is 50–80% below the figure used here. A 40% lower price already cuts the mid-size NPV from \$10.5M to \$5.8M (FIG. D.4).

- **Exceptions run high.** At 2 exceptions per 1,000 actions the mid-size desk grows to about 43 people and costs \$4.3M a year, more than the SAP subscription.
- **Auditors do not change their approach.** Audit fee and SOX savings depend on auditors relying on preventive controls and receipts (\$2.17); the fee evidence is a correlation^[161].
- **Agent programs fail.** Gartner expects more than 40% of agentic AI projects to be canceled by the end of 2027^[170]. Waves that slip past year 2 extend legacy ECC costs for the AE-OS route only.
- **Vendor price ranges.** EDI prices come from vendor pages, not surveys; the line is small and moves the result by less than \$0.2M.

■ REPRODUCIBILITY

Every number in this appendix is computed by `code/python/econ_model.py` from the inputs and sources stated in §D.1–D.3.

Q OPEN RESEARCH QUESTIONS

Open Research Questions

The questions below are where the design is least certain and where evidence from practitioners and researchers would change it. Each belongs to an open research problem (TABLE 6.1).

Q.A Accounting and evidence

- Q1 Auditor acceptance.** What evidence would an external auditor need, beyond Causal Receipts and a SOC 1 / ISAE 3402 report, to rely on AE-OS-generated reports for the accuracy and completeness of financial data (§2.17, §3.6)?
- Q2 Legal numbering at scale.** Is gapless numbering per legal entity and document series, issued at posting time in the Accounting Kernel, sufficient across jurisdictions, and at what posting rate does a single series become a bottleneck (§1.2, §2.10)?
- Q3 Regulated calculations.** Which calculation domains should be delivered only by certified external engines rather than AE-OS Calculation Modules, and who should certify them (§2.11)?
- Q4 Erasure.** Does crypto-shredding of per-subject keys with keyed commitments satisfy data protection authorities' reading of the right to erasure for an append-only log (§2.15)?

Q.B Authority, agents and human oversight

- Q5 Agent reliability.** What measured error rates for agent-proposed actions would be acceptable for each process (e.g. supplier invoice matching, payment proposals), and how should they be tested before an Intent Envelope raises an authority ceiling (§4, §5.1)?
- Q6 Exception volume.** What escalation rate keeps a human exception desk effective without turning approval into a formality (§5.1)?
- Q7 Accountability.** Is the proposed split of accountability between control owner, envelope signer, pack publisher and operator workable under common governance and insurance arrangements (§5.1)?
- Q8 Degraded mode.** Which processes must continue when policy evaluation or a calculation engine is unavailable, and under what compensating controls (§2.12)?

Q.C Security and cryptography

- Q9 Witnesses.** Who should operate checkpoint witnesses for an enterprise log (auditors, regulators, industry bodies), and how many are needed (§3.5)?
- Q10 Crypto-agility.** What migration path to post-quantum signatures should be required for evidence that must remain verifiable for ten years or more (§3.5)?

Q.D Economics and validation in existing organizations

- Q11 Integration effort.** What residual effort per runtime-bound integration (the parameter δ in §D.2) is observed in practice with MCP or similar self-describing interfaces?
- Q12 Work absorbed.** What share of finance and back-office work do agents take over in production, and at what exception rate (§D.3, §D.5)?
- Q13 Validation sequence.** Is the proposed coexistence sequence the right order for moving responsibility from an existing ERP one process at a time, and what exit criteria would an audit committee require at each wave (§2.16)?

Q.E Execution integrity

- Q14 Uncertain outcomes.** When an external action succeeds but its acknowledgement or its ledger record fails (a payment sent, the bank response lost), what recovery protocol is acceptable, and how should the uncertain state be shown to people and auditors (§2.12, §3.5)?

Q.F Identity and trust

- Q15 Agents and strong customer authentication.** When an agent submits a payment under a person's standing mandate, what evidence would a bank or regulator accept in place of the person authenticating each payment, and should the agent itself hold an organization-issued credential (§2.18–2.19)?

Evidence on any of these questions, from practice or from research, would confirm or refute the part of the thesis it concerns. Appendix C shows which requirements this paper specifies as contracts.

08 TECHNICAL FOOTNOTES

References & Sources

Numbered in order of first citation. URLs were checked at the time of writing. Where a figure in the text is a modeled projection, the reference supports the input assumption, not the projection.

- [1] SAP SE. SAP Extends Its Innovation Commitment for SAP S/4HANA, Provides Clarity and Choice on SAP Business Suite 7. Press release, Walldorf, 4 Feb 2020. [new.sap.com/2020/02/sap-s4hana-maintenance-2...](https://www.sap.com/2020/02/sap-s4hana-maintenance-2...)
- [2] G. Gross. (Analyst estimates from Gartner, IDC, Forrester on ECC-to-S/4HANA migration rates.) CIO.com, 4 Jun 2025. cio.com/article/4000543
- [3] GitGuardian. The State of Secrets Sprawl 2026. GitGuardian, March 2026. gitguardian.com/state-of-secrets-sprawl-report...
- [4] Regulation (EU) 2024/1689 (Artificial Intelligence Act), Article 12: Record-keeping; Article 19: Automatically generated logs. Official Journal of the European Union, 12 July 2024. eur-lex.europa.eu/eli/reg/2024/1689/oj
- [5] The Chromium Projects. Certificate Transparency (net/docs/certificate-transparency.md): Chrome CT policy for certificates issued after 30 April 2018. chromium.googlesource.com/chromium/src/+3876...
- [6] S. Steinle. Navigating Your RISE with SAP Journey: Updates for SAP ERP, Private Edition, Transition Option. SAP News Center, 4 Aug 2025. news.sap.com/?p=233946
- [7] M. Bremmer. Economic pressure makes SAP users more pragmatic (DSAG Investment Report 2026). CIO.com, 27 Feb 2026. cio.com/article/4138482/economic-pressure-mak...
- [8] Gartner. What IT Leaders Must Do to Avoid Disappointing ERP Initiatives. Gartner Insights. gartner.com/en/information-technology/insight...
- [9] Consultancy.uk. Lidl cancels SAP introduction having sunk €500 million into it. 13 Aug 2018 (cost is a press estimate, not confirmed by Lidl or SAP). consultancy.uk/news/18243/lidl-cancels-sap-in...
- [10] L. Clark. Birmingham's Oracle ERP cost projection (Audit Reform Lab figures). The Register, 20 Aug 2024. theregister.com/2024/08/20/birmingham_oracle_...
- [11] SAP Learning. Discussing the Universal Journal (course: Outlining Cost Management and Profitability Analysis). SAP SE. learning.sap.com/courses/outlining-cost-manag...
- [12] SAP PRESS. How Many Custom Fields Can You Add to the Universal Journal in SAP S/4HANA Finance? (Q&A with J. Salmon), 11 May 2022. blog.sap-press.com/how-many-custom-fields-can...
- [13] D. Paturu. What Actually Breaks in Large-Scale S/4HANA Conversions (And How to Prevent It). DZone, 30 Apr 2026. dzone.com/articles/what-actually-breaks-in-la...
- [14] SAP SE. The Delta Merge Operation. SAP Help Portal (SAP NetWeaver 7.3 / SAP HANA documentation). help.sap.com/saphelp_nw73/helpdata/en/62/9d41...
- [15] Amazon Web Services. SAP Lens, Best Practice 17.4 – Review the size, granularity, and latest available EC2 instances for SAP components (cites SAP Note 2408419, S/4HANA multi-node support). docs.aws.amazon.com/wellarchitected/latest/sa...
- [16] SAP Community. Buffering Number Ranges and Legal Framework. Technology Blogs by SAP. blogs.sap.com/t5/technology-blog-s-by-sap/buff...
- [17] G. Scott (ASUG). ERP customers and the custom code dilemma – can automation help? *diginomica*, 26 Jul 2023 (ASUG/smartShift research). diginomica.com/erp-customers-and-custom-code-...
- [18] Hevner, A. R., March, S. T., Park, J. and Ram, S. Design Science in Information Systems Research. *MIS Quarterly* 28(1), 75–105, 2004. doi:10.2307/25148625. doi.org/10.2307/25148625
- [19] Peffers, K., Tuunanen, T., Rothenberger, M. A. and Chatterjee, S. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems* 24(3), 45–77, 2007. doi:10.2753/MIS0742-1222240302. doi.org/10.2753/MIS0742-1222240302
- [20] Venable, J., Pries-Heje, J. and Baskerville, R. FEDS: a Framework for Evaluation in Design Science Research. *European Journal of Information Systems* 25(1), 77–89, 2016. doi:10.1057/ejis.2014.36. doi.org/10.1057/ejis.2014.36
- [21] Gregor, S. and Hevner, A. R. Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly* 37(2), 337–355, 2013. doi:10.25300/MISQ/2013/37.2.01. doi.org/10.25300/MISQ/2013/37.2.01
- [22] P. Moritz, R. Nishihara, S. Wang et al. Ray: A Distributed Framework for Emerging AI Applications. *USENIX OSDI 2018* (arXiv:1712.05889). arxiv.org/abs/1712.05889
- [23] M. Fowler. Event Sourcing. martinfowler.com/EventSourcing.html
- [24] P. E. O'Neil. The Escrow Transactional Method. *ACM Transactions on Database Systems* 11(4):405–430, 1986. doi.org/10.1145/7239.7265
- [25] M. Shapiro, N. Pregoica, C. Baquero, M. Zawirski. Conflict-free Replicated Data Types. SSS 2011, LNCS 6976 (INRIA RR-7687). inria.hal.science/inria-00609399v1
- [26] Cloud Native Computing Foundation. SPIFFE and SPIRE Projects Graduate from Cloud Native Computing Foundation Incubator. CNCF, September 20, 2022. cncf.io/announcements/2022/09/20/spiffe-and-s...
- [27] Cloud Native Computing Foundation. Cloud Native Computing Foundation Announces Cilium Graduation. CNCF, October 11, 2023. cncf.io/announcements/2023/10/11/cloud-native...
- [28] Uber Engineering. Our Journey Adopting SPIFFE/SPIRE at Scale. Uber Blog, November 9, 2023. uber.com/en-US/blog/our-journey-adopting-spf...
- [29] Bytecode Alliance. wasmtime::Config API documentation (consume_fuel, epoch_interruption, cranelift_nan_canonicalization). docs.wasmtime.dev/api/wasmtime/struct.Config...
- [30] Open Policy Agent. WebAssembly (compiling Rego policies to Wasm with opa build -t wasm). OPA documentation. openpolicyagent.org/docs/wasm
- [31] Cutler, J. W., Disselkoben, C., Eline, A., et al. Cedar: A New Language for Expressive, Fast, Safe, and Analyzable Authorization. *Proc. ACM Program. Lang.* 8 (OOPSLA1), 2024. doi:10.1145/3649835. Extended version arXiv:2403.04651. arxiv.org/abs/2403.04651
- [32] Laurie, B., Messeri, E., Stradling, R. Certificate Transparency Version 2.0. RFC 9162 (obsoletes RFC 6962), IETF, December 2021. rfc-editor.org/rfc/rfc9162
- [33] Sigstore. Rekord — Transparency Log Overview. Sigstore documentation. docs.sigstore.dev/logging/overview
- [34] CyberArk. 2025 Identity Security Landscape (press release: Machine Identities Outnumber Humans by More Than 80 to 1). CyberArk / Vanson Bourne, n=2,600, April 23, 2025. businesswire.com/news/home/20250423817886/en
- [35] Verizon. 2025 Data Breach Investigations Report. Verizon Business, April 2025. verizon.com/business/resources/reports/dbir
- [36] GitGuardian. The State of Secrets Sprawl 2025. GitGuardian, March 2025. blog.gitguardian.com/the-state-of-secrets-spr...
- [37] GitGuardian. Verizon DBIR 2025: Secrets, Third Parties and Credential Abuse (analysis of DBIR secrets-leak data). GitGuardian Blog, 2025. blog.gitguardian.com/verizon-dbir-2025
- [38] OWASP Foundation. OWASP Top 10 Non-Human Identities Risks – 2025. OWASP Non-Human Identities Top 10 Project, 2025. owasp.org/www-project-non-human-identities-to...

08 TECHNICAL FOOTNOTES • CONTINUED

- [39] OWASP GenAI Security Project. OWASP Top 10 for Agentic Applications for 2026. OWASP, December 9, 2025. genai.owasp.org/resource/owasp-top-10-for-age...
- [40] M. Prince, J. Graham-Cumming, G. Kemp. Thanksgiving 2023 security incident. Cloudflare Blog, February 1, 2024. blog.cloudflare.com/thanksgiving-2023-securit...
- [41] Microsoft Threat Intelligence. Midnight Blizzard: Guidance for responders on nation-state attack. Microsoft Security Blog, January 25, 2024. microsoft.com/en-us/security/blog/2024/01/25/...
- [42] Help Net Security. Dropbox says attackers accessed customer and MFA info, API keys. Help Net Security, May 2, 2024. helpnetsecurity.com/2024/05/02/dropbox-sign-b...
- [43] Mandiant (Google Cloud). UNC5537 Targets Snowflake Customer Instances for Data Theft and Extortion. Google Cloud Threat Intelligence, June 10, 2024. cloud.google.com/blog/topics/threat-intellige...
- [44] L. Abrams. Internet Archive breached again through stolen access tokens. BleepingComputer, October 20, 2024. bleepingcomputer.com/news/security/internet-a...
- [45] StepSecurity. Harden-Runner detection: tj-actions/changed-files action is compromised (CVE-2025-30066). StepSecurity Blog, March 2025. stepsecurity.io/blog/harden-runner-detection-...
- [46] SPIFFE Project. The SPIFFE Identity and Verifiable Identity Document (SPIFFE-ID standard). github.com/spiffe/spiffe. github.com/spiffe/spiffe/blob/main/standards/...
- [47] SPIFFE Project. Configuring SPIRE Server (default_x509_svid_ttl, default_jwt_svid_ttl, NodeAttestor plugins). spiffe.io documentation. spiffe.io/docs/latest/deploying/spire_server
- [48] The Linux Kernel Documentation. Kernel TLS offload. docs.kernel.org. docs.kernel.org/networking/tls-offload.html
- [49] Cilium Project. Mutual Authentication (Beta). Cilium Documentation. docs.cilium.io. docs.cilium.io/en/stable/network/servicemesh/...
- [50] A. Birgisson, J. G. Politz, U. Erlingsson, A. Taly, M. Vrabie, M. Lentzner. Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud. NDSS 2014. ndss-symposium.org/ndss2014/programme/macaroo...
- [51] Eclipse Biscuit. Biscuit authorization tokens: decentralized verification, offline attenuation, Datalog policies. biscuitsec.org. biscuitsec.org
- [52] H. Garcia-Molina, K. Salem. Sagas. Proceedings of the 1987 ACM SIGMOD International Conference on Management of Data, pp. 249–259. ACM, 1987. doi:10.1145/38713.38742. dl.acm.org/doi/10.1145/38713.38742
- [53] IFRS Foundation. IAS 21 The Effects of Changes in Foreign Exchange Rates (foreign currency transactions, translation of foreign operations and into a presentation currency). IASB. ifrs.org/issue-d-standards/list-of-standards/i...
- [54] Council Directive 2006/112/EC of 28 November 2006 on the common system of value added tax, Article 226(2): invoices must carry "a sequential number, based on one or more series, which uniquely identifies the invoice". Official Journal of the EU L 347, 2006. eur-lex.europa.eu/eli/d/ir/2006/112/oj/eng
- [55] WebAssembly Community Group. Nondeterminism in WebAssembly (design/Nondeterminism.md). github.com/WebAssembly/design/blob/main/Nonde...
- [56] IFRS Foundation. IAS 16 Property, Plant and Equipment (2014 amendment prohibits revenue-based depreciation methods). IASB. ifrs.org/issued-standards/list-of-standards/i...
- [57] IFRS Foundation. IAS 2 Inventories (lower of cost and net realisable value; FIFO or weighted average cost formulas for interchangeable items). IASB. ifrs.org/issued-standards/list-of-standards/i...
- [58] IFRS Foundation. IFRS 15 Revenue from Contracts with Customers (issued May 2014; effective for annual periods beginning on or after 1 January 2018; five-step model, transaction price allocated on relative stand-alone selling prices). IASB. ifrs.org/issued-standards/list-of-standards/i...
- [59] FASB. Accounting Standards Update No. 2014-09, Revenue from Contracts with Customers (Topic 606). Financial Accounting Standards Board, May 2014. www.pwc.com/dt/us/en/fasb_financial_acc...
- [60] IFRS Foundation. IFRS 16 Leases (issued January 2016; effective for annual periods beginning on or after 1 January 2019; lessees recognise a right-of-use asset and a lease liability). IASB. ifrs.org/issued-standards/list-of-standards/i...
- [61] Council Directive (EU) 2025/516 of 11 March 2025 amending Directive 2006/112/EC as regards VAT rules for the digital age (ViDA). Official Journal of the European Union, 25 Mar 2025. eur-lex.europa.eu/eli/dir/2025/516/oj/eng
- [62] Banqup. ViDA adopted: everything you need to know about the new e-invoicing and reporting requirements. Banqup blog, 2025. banqup.com/resources/blog/vida-adopted-every...
- [63] Comarch. France confirms September 2026 e-invoicing go-live and publishes start-up guidance. Comarch, 21 Jul 2026. comarch.com/trade-and-services/data-managemen...
- [64] VATCalc. Belgium 2026 e-invoicing; 2028 e-reporting law. VATCalc, 19 Jul 2026. vatcalc.com/?p=27567
- [65] IHK Fulda. E-Rechnung: Pflichten für Unternehmen ab 2025, 2027 und 2028. Industrie- und Handelskammer Fulda (accessed Oct 2026). ihk.de/fulda/standortpolitik/handel-dienstlei...
- [66] VATCalc. Which countries have introduced OECD's SAF-T? VATCalc (accessed Oct 2026). vatcalc.com/global/which-countries-have-intro...
- [67] EY. Why Pillar Two compliance demands focus as major filing deadline looms. EY Insights, 19 Jun 2026. ey.com/en_gl/insights/tax/why-pillar-two-comp...
- [68] IFRS Foundation. IFRS 18 Presentation and Disclosure in Financial Statements. IASB, issued Apr 2024. ifrs.org/issued-standards/list-of-standards/i...
- [69] Morrison Foerster. EU Sustainability Omnibus I: 'Detailed Omnibus' adopted: what the final CSRD/CSDDD deal means for companies. MoFo, 22 Dec 2025. mofo.com/resources/insights/251222-eu-sustain...
- [70] Herbert Smith Freehills Kramer. Omnibus I update: CSRD and CSDDD amendment directive (Directive (EU) 2026/470) published in the Official Journal. HSF Kramer, 27 Feb 2026. hsfkramer.com/notes/esg/2026-posts/omnibus-i-...
- [71] Regulation (EU) 2016/679 (General Data Protection Regulation), Art. 17 and Recital 65. Official Journal of the European Union, 4 May 2016. eur-lex.europa.eu/eli/reg/2016/679/oj/eng
- [72] European Data Protection Board. Guidelines 02/2025 on processing of personal data through blockchain technologies, version 2 (final, after public consultation). EDPB, 7 Jul 2026. edpb.europa.eu/documents/guideline/guidelines...
- [73] NIST. SP 800-88 Rev. 2, Guidelines for Media Sanitization, §3.2 Cryptographic Erase. National Institute of Standards and Technology, Sep 2025 (supersedes Rev. 1). csrc.nist.gov/pubs/sp/800/88/r2/final
- [74] M. Fowler. Strangler Fig (originally 'Strangler Application'). martinfowler.com, revised 22 Aug 2024. martinfowler.com/bliki/StranglerFigApplication...
- [75] PCAOB. AS 2201: An Audit of Internal Control Over Financial Reporting That Is Integrated with An Audit of Financial Statements (incl. ¶1.36, ¶1.47 and App. B, B17–B27 Use of Service Organizations). pcaobus.org/oversight/standards/auditing-stan...
- [76] IAASB. ISAE 3402, Assurance Reports on Controls at a Service Organization: staff overview. IFAC; standard issued Dec 2009, effective for periods ending on or after 15 Jun 2011. ifac.org/system/files/publications/files/inte...
- [77] SPIFFE Project. SPIFFE Concepts: SPIFFE ID, SVID, X.509-SVID, JWT-SVID, Workload API. spiffe.io. spiffe.io/docs/latest/spiffe-about/spiffe-con...

08 TECHNICAL FOOTNOTES • CONTINUED

- [78] European Parliament and Council. Regulation (EU) 2024/1183 of 11 April 2024 amending Regulation (EU) No 910/2014 as regards establishing the European Digital Identity Framework. Official Journal of the EU (OJ L, 2024/1183), 2024. eur-lex.europa.eu/eli/reg/2024/1183/oj/eng
- [79] European Commission. Commission Implementing Regulation (EU) 2024/2979 of 28 November 2024 laying down rules for the application of Regulation (EU) No 910/2014 as regards the integrity and core functionalities of European Digital Identity Wallets. Official Journal of the EU, 2024. eur-lex.europa.eu/eli/reg_impl/2024/2979/oj
- [80] European Commission. European Digital Identity (topic page). European Commission, accessed 2026-10-12. commission.europa.eu/topics/digital-economy-a...
- [81] European Commission. Proposal for a Regulation on the establishment of European Business Wallets (COM(2025) 838, CELEX 52025PC0838). European Commission, 19 November 2025. digital-strategy.ec.europa.eu/en/library/prop...
- [82] European Parliament Legislative Observatory. Establishment of European Business Wallets, procedure 2025/0358(COD): summary of the legislative proposal. European Parliament, 2025. oeil.europarl.europa.eu/oeil/en/document-sum...
- [83] Council of the European Union. Note 9684/26: Proposal for a Regulation on the establishment of European Business Wallets, preparation for general approach (TTE Council, 9 June 2026). Council of the EU, 2 June 2026. data.consilium.europa.eu/doc/document/ST-9684...
- [84] GLEIF. Introducing the verifiable LEI (vLEI). Global Legal Entity Identifier Foundation, accessed 2026-10-12. gleif.org/en/organizational-identity/introduc...
- [85] ISO. ISO 17442-3:2024 Financial services - Legal entity identifier (LEI) - Part 3: Verifiable LEIs (vLEIs). International Organization for Standardization, October 2024. iso.org/standard/85628.html
- [86] GLEIF. vLEI Ecosystem Governance Framework. Global Legal Entity Identifier Foundation, accessed 2026-10-12. gleif.org/en/organizational-identity/become-a...
- [87] W3C. Verifiable Credentials Data Model v2.0, W3C Recommendation. World Wide Web Consortium, 15 May 2025. w3.org/TR/vc-data-model-2.0
- [88] Fett, D., Yasuda, K., Campbell, B. RFC 9901: Selective Disclosure for JSON Web Tokens (SD-JWT). IETF, November 2025. rfc-editor.org/rfc/rfc9901.html
- [89] OpenID Foundation. OpenID for Verifiable Presentations 1.0 (Final). OpenID Foundation, 9 July 2025. openid.net/specs/openid-4-verifiable-presenta...
- [90] OpenID Foundation. OpenID for Verifiable Credential Issuance 1.0 (Final). OpenID Foundation, 16 September 2025. openid.net/specs/openid-4-verifiable-credenti...
- [91] Jones, M., Nadalin, A., Campbell, B., Bradley, J., Mortimore, C. RFC 8693: OAuth 2.0 Token Exchange. IETF, January 2020. rfc-editor.org/rfc/rfc8693.html
- [92] IETF OAuth Working Group. Transaction Tokens, draft-ietf-oauth-transaction-tokens-11 (Internet-Draft, work in progress). IETF, 30 July 2026. datatracker.ietf.org/doc/draft-ietf-oauth-tra...
- [93] OpenID Foundation AI Identity Management Community Group. Identity Management for Agentic AI: The new frontier of authorization, authentication, and security for an AI agent world. OpenID Foundation, October 2025. openid.net/new-whitepaper-tackles-ai-agent-id...
- [94] European Parliament and Council. Regulation (EU) 2024/886 of 13 March 2024 amending Regulations (EU) No 260/2012 and (EU) 2021/1230 and Directives 98/26/EC and (EU) 2015/2366 as regards instant credit transfers in euro. Official Journal of the EU, 2024. eur-lex.europa.eu/eli/reg/2024/886/oj/eng
- [95] Council of the EU. Payment services: Council and Parliament agree to step up the fight against fraud and increase transparency (press release). Council of the EU, 27 November 2025. consilium.europa.eu/en/press/press-releases/2...
- [96] European Parliament. Payment services deal: More protection from online fraud and hidden fees (press release 20251121IPR31540). European Parliament, November 2025. europarl.europa.eu/news/en/press-room/2025112...
- [97] Kech, A. (GLEIF). When AI agents transact, who stands behind them? Why verifiable organizational identity is critical to agentic payments. GLEIF Blog, 1 September 2026. gleif.org/en/newsroom/blog/when-ai-agents-tra...
- [98] Sarbanes-Oxley Act of 2002, §404 — Management assessment of internal controls, codified at 15 U.S.C. § 7262. law.cornell.edu/uscode/text/15/7262
- [99] Merkle, R. C. Method of providing digital signatures. US Patent 4,309,569 (filed 5 Sep 1979, issued 5 Jan 1982). patents.google.com/patent/US4309569A/en
- [100] Merkle, R. C. A Digital Signature Based on a Conventional Encryption Function. In: Pomerance, C. (ed.) Advances in Cryptology — CRYPTO '87, LNCS 293, pp. 369–378. Springer, 1988. doi.org/10.1007/3-540-48184-2_32
- [101] Laurie, B., Langley, A., Kasper, E. Certificate Transparency. RFC 6962, IETF, June 2013. rfc-editor.org/rfc/rfc6962
- [102] SPIFFE Project. SPIRE Agent configuration reference (doc/spire_agent.md): workload_x509_svid_key_type <rsa-2048|ec-p256|ec-p384>, default ec-p256. github.com/spiffe/spire. github.com/spiffe/spire/blob/main/doc/spire_a...
- [103] Multiformats. CID (Content Identifier) Specification. github.com/multiformats/cid
- [104] NIST. FIPS PUB 180-4: Secure Hash Standard (SHS). National Institute of Standards and Technology, August 2015. nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-...
- [105] NIST Computer Security Resource Center. Hash Functions — Security Strengths of Approved Hash Functions (SHA-256: collision resistance 128 bits, preimage 256 bits). csrc.nist.gov/projects/hash-functions
- [106] PLD. DAG-CBOR Codec Specification. IPLD Specs. ipld.io/specs/codecs/dag-cbor/spec
- [107] Bormann, C., Hoffman, P. Concise Binary Object Representation (CBOR). RFC 8949, §4.2 Deterministically Encoded CBOR. IETF, December 2020. rfc-editor.org/rfc/rfc8949
- [108] Multiformats. Multihash: self-describing hashes. github.com/multiformats/multihash
- [109] Multiformats. Multicodec table (table.csv): sha2-256 = 0x12, dag-cbor = 0x71, cidv1 = 0x01. github.com/multiformats/multicodec/blob/maste...
- [110] C2SP. tlog-checkpoint: Transparency Log Checkpoints (signed-note checkpoint format). Community Cryptography Specification Project. github.com/C2SP/C2SP/blob/main/tlog-checkpoint...
- [111] Josefsson, S., Liusvaara, I. Edwards-Curve Digital Signature Algorithm (EdDSA). RFC 8032, IRTF, January 2017. rfc-editor.org/rfc/rfc8032
- [112] C2SP. tlog-witness: Transparency Log Witness Protocol (witnesses verify consistency proofs and return timestamped cosignatures). Community Cryptography Specification Project. github.com/C2SP/C2SP/blob/main/tlog-witness.md
- [113] NIST. FIPS PUB 186-5: Digital Signature Standard (DSS) (adds EdDSA). National Institute of Standards and Technology, February 3, 2023. csrc.nist.gov/pubs/fips/186-5/final
- [114] C2SP. tlog-cosignature: Transparency Log Cosignatures (Ed25519 and ML-DSA-44 cosignatures; ML-DSA-44 SHOULD be used for new deployments). Community Cryptography Specification Project. github.com/C2SP/C2SP/blob/main/tlog-cosignatu...
- [115] W3C. Web Authentication: An API for accessing Public Key Credentials, Level 2. W3C Recommendation, 8 April 2021. w3.org/TR/webauthn-2
- [116] Adams, C., Cain, P., Pinkas, D., Zuccherato, R. Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). RFC 3161, IETF, August 2001. rfc-editor.org/rfc/rfc3161

08 TECHNICAL FOOTNOTES • CONTINUED

- [117] NIST. FIPS PUB 204: Module-Lattice-Based Digital Signature Standard (ML-DSA). National Institute of Standards and Technology, August 13, 2024. csrc.nist.gov/pubs/fips/204/final
- [118] Bernstein, D. J., Duif, N., Lange, T., Schwabe, P., Yang, B.-Y. Ed25519: high-speed high-security signatures (project page and benchmarks). ed25519.cr.yp.to
- [119] Bytecode Alliance. Wasmtime Security documentation. docs.wasmtime.dev/security.html
- [120] L. Beurer-Kellner, M. Fischer. MCP Security Notification: Tool Poisoning Attacks. Invariant Labs, April 1, 2025. invariantlabs.ai/blog/mcp-security-notification...
- [121] A. Birgisson, J. G. Politz, Ú. Erlingsson, A. Taly, M. Vrabie, M. Lentzner. Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud. NDSS 2014. research.google/pubs/pub30605
- [122] Biscuit authorization. Introduction (public-key signed tokens, offline attenuation, Datalog policies). doc.biscuitsec.org/doc.biscuitsec.org/getting-started/introduction...
- [123] Anthropic. Introducing the Model Context Protocol. Anthropic News, November 25, 2024. anthropic.com/news/model-context-protocol
- [124] Model Context Protocol. Specification, revision 2026-07-28 (overview: JSON-RPC 2.0 messages between hosts, clients and servers; resources, prompts, tools; stateless, self-contained requests). modelcontextprotocol.io/specification/2026-07-28
- [125] Model Context Protocol. Specification 2026-07-28: Server Features / Tools (tools/list, tools/call, inputSchema, outputSchema, annotations; security considerations). modelcontextprotocol.io/specification/2026-07-28
- [126] Model Context Protocol. Specification 2026-07-28: Transports (stdio; Streamable HTTP, single POST endpoint, MCP-Protocol-Version / Mcp-Method / Mcp-Name headers, Origin validation). modelcontextprotocol.io/specification/2026-07-28
- [127] Model Context Protocol. Specification 2026-07-28: Authorization (OAuth 2.1 draft-ietf-oauth-v2-1-13, RFC 9728 Protected Resource Metadata, RFC 8707 resource indicators, audience validation, no token passthrough). modelcontextprotocol.io/specification/2026-07-28
- [128] SPIFFE Project. The JWT SPIFFE Verifiable Identity Document (JWT-SVID) standard: sub, aud (required), exp (required), RS/ES/PS algorithms only. GitHub [spiffe/spiffe.github.com/spiffe/spiffe/blob/main/standards/](https://github.com/spiffe/spiffe/blob/main/standards/)
- [129] OWASP Foundation. OWASP MCP Top 10 (2025 list, beta). OWASP, 2025-2026. owasp.org/www-project-mcp-top-10
- [130] Model Context Protocol. Security Best Practices (confused deputy, token passthrough, SSRF, state-handle hijacking, local server compromise, scope minimization). modelcontextprotocol.io/docs/tutorials/security...
- [131] D. Peles. Critical RCE Vulnerability in mcp-remote: CVE-2025-6514 Threatens LLM Clients (CVSS 9.6; versions 0.0.5-0.1.15, fixed in 0.1.16). JFrog Security Research, July 9, 2025. jfrog.com/blog/2025-6514-critical-mcp-remote-...
- [132] eBPF Foundation. What is eBPF? (verifier, JIT, hooks, maps). ebpf.io/ebpf/what-is-ebpf
- [133] T. Høiland-Jørgensen, J. D. Brouer, D. Borkmann, J. Fastabend, T. Herbert, D. Ahern, D. Miller. The eXpress Data Path: Fast Programmable Packet Processing in the Operating System Kernel. ACM CoNEXT '18, 2018. doi.org/10.1145/3281411.3281443
- [134] Cilium documentation. Identity-Based security (identity derived from labels; security separated from network addressing). docs.cilium.io/docs.cilium.io/en/stable/security/network/identity...
- [135] Linux kernel documentation. LSM BPF Programs. docs.kernel.org/docs.kernel.org/bpf/prog_lsm.html
- [136] Linux kernel documentation. BPF_MAP_TYPE_SOCKMAP and BPF_MAP_TYPE_SOCKHASH. docs.kernel.org/docs.kernel.org/bpf/map_sockmap.html
- [137] Cilium Tetragon. Overview and Enforcement (in-kernel eBPF filtering; Override and Sigkill actions). tetragon.io/docs/concepts/enforcement
- [138] CSIS ChinaPower Project. How Much Trade Transits the South China Sea? Center for Strategic and International Studies, 2017 (updated). Estimates \$3.4 trillion of trade transited the South China Sea in 2016 (~21% of global trade); notes the widely cited \$5.3 trillion figure overstates it. chinapower.csis.org/much-trade-transits-south...
- [139] Hong Kong Observatory. Number of Tropical Cyclones (including Tropical Depressions) entering the 500 km range of Hong Kong, 1961–2020 (annual mean 6.08). HKO Tropical Cyclone Statistics. hko.gov.hk/en/informtc/tcStatTable6c.htm
- [140] Hand, M. Maersk warns Yantian port congestion a 'much bigger disruption' than Suez Canal closure. Seatrade Maritime News, 15 June 2021. seatrade-maritime.com/containers/maersk-warns...
- [141] Container News. Pearl River Delta ports brace for Super Typhoon Saola. 1 September 2023. container-news.com/pearl-river-delta-ports-br...
- [142] Kuehne+Nagel. Update: Typhoon Yagi suspends port activities in Vietnam, China and Hong Kong. Customer advisory, 5–6 September 2024. mykn.kuehne-nagel.com/news/article/typhoon-yagi...
- [143] CIS. Typhoon Yagi disrupts chemical shipping in China's Hainan province. ICIS News, 6 September 2024. icis.com/explorer/resources/news/2024/09/06/11...
- [144] Hydrocarbon Processing. Taiwan refiners brace for super typhoon Krathon as ports shut ahead of landfall. 1 October 2024. hydrocarbonprocessing.com/news/2024/10-1/taiw...
- [145] Natural Earth. 1:50m Land polygons (public domain map data), naturalearthdata.com; used for the schematic basemap in FIG. 5.2. naturalearthdata.com/downloads/50m-physical-v...
- [146] D. Bormann, P. Hoffman. RFC 8949: Concise Binary Object Representation (CBOR) (s.4.2.1 core deterministic encoding). IETF, December 2020. rfc-editor.org/rfc/rfc8949.html
- [147] B. Laurie, E. Messeri, R. Stradling. RFC 9162: Certificate Transparency Version 2.0 (s.2.1 Merkle tree hashing and inclusion proofs). IETF, December 2021. rfc-editor.org/rfc/rfc9162.html
- [148] DC (sponsored by SAP). SAP Partner Ecosystem Growth Drivers: Strategies for Increasing Partner Profitability and Customer Value, doc #US50040723, Jan 2023; as reported by M. Weyman, SAP News Center, 9 Feb 2023. news.sap.com/2023/02/idc-intellectual-property...
- [149] Salesforce / MuleSoft. 2025 Connectivity Benchmark Report (10th ed.; survey of 1,050 enterprise IT leaders, Oct–Nov 2024). Salesforce News, 2025. salesforce.com/news/stories/connectivity-repo...
- [150] Hyperbots. SAP S/4HANA Pricing Explained: What Finance Leaders Actually End Up Paying. Hyperbots (vendor blog), 2026. Private Edition ~1,000-user deal \$1.5-2M/yr after discounts; 500-2,000 users: implementation \$5-25M, 5-year TCO \$15-45M. blog.hyperbots.com/sap-s-4hana-pricing-explai...
- [151] M. Bloch, S. Blumberg, J. Laartz. Delivering large-scale IT projects on time, on budget, and on value. McKinsey & Company (with University of Oxford), Oct 2021. mckinsey.com/capabilities/mckinsey-digital/ou...
- [152] Redress Compliance. RISE with SAP: What the All-In-One Deal Really Costs. Redress Compliance (licensing advisory), 2026. Professional users EUR 500-1,500/yr list, EUR 350-800 effective above 1,000 users; ongoing spend 40-50% above the subscription line. redresscompliance.com/sap-rise-what-it-really...

08 TECHNICAL FOOTNOTES • CONTINUED

- [153] Anthropic. Pricing: model pricing per million tokens (base input, cache hits, output) for Claude Haiku 5.5, Sonnet 5.5 and Opus 5.5. Claude Platform documentation, accessed 12 Oct 2026. platform.claude.com/docs/en/about-claude/pricing...
- [154] U.S. Bureau of Labor Statistics. Occupational Employment and Wages, May 2023: 13-2011 Accountants and Auditors. BLS, 2024. Mean annual wage \$90,780. bls.gov/oes/2023/may/oes132011.htm
- [155] U.S. Bureau of Labor Statistics. Occupational Employment and Wages, May 2023: 43-3031 Bookkeeping, Accounting, and Auditing Clerks. BLS, 2024. Mean annual wage \$49,580. bls.gov/oes/2023/may/oes433031.htm
- [156] U.S. Bureau of Labor Statistics. Employer Costs for Employee Compensation - June 2026. BLS, 2026. Wages 70.0% of private-industry employer compensation cost (benefits 30.0%). bls.gov/news.release/ec.ec.nr0.htm
- [157] OrderSync. EDI VAN Explained: What It Is and When You Need One. OrderSync (vendor), 2026. VAN estimates: \$0.02-0.15 per kilocharacter, \$150-500/month base, \$500-2,500 per trading-partner setup (vendor estimate). ordersync.io/blog/edi-van-explained
- [158] S. Wills. EDI via VAN vs Peppol eInvoicing: What's Actually Different and What It Costs. Acume (vendor), 2026. VAN \$0.05-0.50+ per document and \$1,000-5,000 per connection vs Peppol about \$0.05 per document (vendor estimate). acumelimited.com/edi-van-vs-peppol-einvoicing
- [159] Financial Executives Research Foundation. 16th Annual Audit Fee Survey (FY2024) press release. FEI, 2026. Audit fees 0.116% of revenue at \$1-5B revenue and 0.057% above \$5B; S&P 500 median \$7.96M. financialexecutives.org/About-FEI/For-the-Pre...
- [160] Protiviti. 2022 Sarbanes-Oxley Compliance Survey. Protiviti, 2022. Internal SOX cost (excl. external audit) \$1.40M at \$1-4.99B revenue, \$1.45M at \$5-9.99B, \$1.93M at \$10B+. protiviti.com/sites/default/files/2022-06/2022...
- [161] Gartner. Gartner Survey Shows 62 Percent of Organizations Expect External Audit Fees to Increase in 2021. Gartner, 2021. Organizations automating at least 25% of internal controls paid 27% lower audit fees on average (n=166). gartner.com/en/newsroom/press-releases/2021-0...
- [162] APQC. Number of finance function FTEs per \$1 billion revenue. APQC Open Standards Benchmarking, accessed 2026. Median 71.2 FTEs per \$1B revenue (n=7,114). apqc.org/resources/benchmarking/open-standard...
- [163] Deloitte. Hands off the task, eyes on the outcome (case study). Deloitte, 2026. Consumer-products AP: 92% touchless processing and 50% fewer invoice-processing FTEs (single client). deloitte.com/us/en/what-we-do/case-studies/ha...
- [164] The Hackett Group. 2025 Working Capital Survey: Payables Rebound but Receivables and Inventory Lag. Business Wire, 2025. \$1.7T excess working capital in top 1,000 US non-financial firms (11% of revenue); CCC 37 days; 18-day DSO gap top vs median. businesswire.com/news/home/20250818205819/en/...
- [165] C. L. Purchase (citing APQC). Inventory Optimization: Show Me the Money. Logistics Management, 2011. Median inventory carrying cost 10% of inventory value (top 7.3%, bottom 16.4%). logisticsmanagement.com/article/inventory_optimizat...
- [166] APQC. Premium freight charges as a percentage of total freight charges. APQC Open Standards Benchmarking, accessed 2026. Median 8.0% (n=1,964). apqc.org/resources/benchmarking/open-standard...
- [167] APQC. Percentage of discounts available that are taken. APQC Open Standards Benchmarking, accessed 2026. Median 65.0% of early-payment discounts captured (n=2,076). apqc.org/resources/benchmarking/open-standard...
- [168] TIC. 2024 Hourly Cost of Downtime Survey, Part 2. Information Technology Intelligence Consulting, 2024. Over 90% of mid-size and large enterprises put one hour of downtime above \$300,000; 41% at \$1M-\$5M+. calyptix.com/wp-content/uploads/ITIC-2024-Hou...
- [169] J. H. Choi, C. Xie. Human + AI in Accounting: Early Evidence from the Field. Stanford GSB Working Paper 4261, 2025. AI-using accountants: 7.5-day faster monthly close, 55% more weekly client support. gsb.stanford.edu/faculty-research/working-pap...
- [170] Gartner. Gartner Predicts Over 40% of Agentic AI Projects Will Be Canceled by End of 2027. Press release, 25 Jun 2025. gartner.com/en/newsroom/press-releases/2025-0...

AE-OS

Agents act on intent signed by people and within policy checked before execution, and **each action leaves a verifiable causal receipt.**

GLOSSARY

APO	Agent Primitive Object: the deployable unit of agent logic, identity and budget	ACDOCA	SAP S/4HANA Universal Journal line-item table
CID	Content identifier: multihash (SHA-256) of a canonically encoded object	DAG-CBOR	Deterministic CBOR encoding for linked data (IPLD)
Causal Receipt	Merkle inclusion proof returned for every agent action	eBPF	Sandboxed programs in the Linux kernel; used for network policy and observability
KTLS	Kernel TLS: record encryption offloaded to the kernel socket layer	MCP	Model Context Protocol: JSON-RPC protocol for tool and context discovery
SPIFFE	Secure Production Identity Framework for Everyone; SPIRE is its runtime	SVID	SPIFFE Verifiable Identity Document (X.509 or JWT), short-lived
SoD	Segregation of duties control	WASM	WebAssembly: portable, sandboxed bytecode for policy modules

THE CAUSAL PIPELINE



DOCUMENT FINGERPRINT // SHA-256

40fd5737603e9a64b4d805726ed82bde5d8f0bd874088f6f82b8262727812b48