TECHNICAL WHITEPAPER WP-01 · DETERMINISTIC API-TO-AGENT QUALIFICATION, OVERLAYS & GOVERNED VMCP RUNTIME

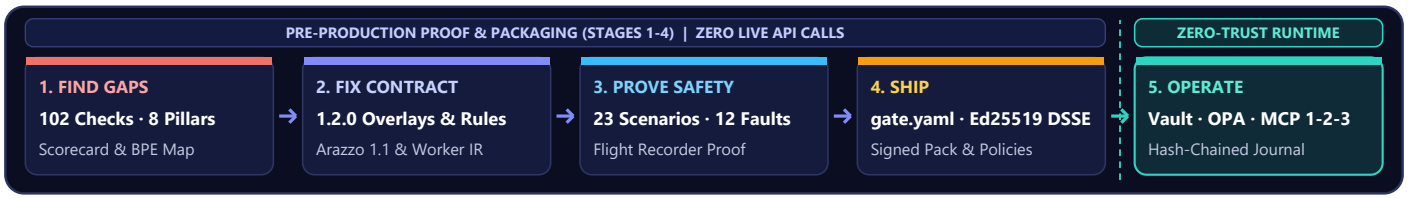
Deterministic API-to-Agent Capability Architecture: The 8-Pillar Readiness Model, Autonomy Tiers & OpenAPI 1.2.0 Overlays

Make your APIs production-ready for AI agents without rewriting backend code. Before an AI agent acts on a customer system, verify it acts on the right customer, executes once, and stops within your limits-with 8-pillar qualification, OpenAPI 1.2.0 Overlays, 23-scenario fault simulation, and DSSE Evidence Packs.

0 Rewrites	8 Pillars	137k -> 477	6 Frameworks
NON-DESTRUCTIVE OVERLAYS	**102 DETERMINISTIC RULES**	**EXACT o200k_base BPE**	**DSSE SIGNED EVIDENCE**
Layer OpenAPI 1.2.0 guardrails without changing backend service code.	Fail-closed AR0-AR4 Autonomy Tiers with zero LLM security guesswork.	Single-tool (-99.7%) or ~380+477 progressive (-99.4%) on 520-op enterprise catalog.	SOC 2, ISO 27001/42001, NIST AI RMF, EU AI Act, OWASP, PCI-DSS 4.0.

1. The Production Crisis: Why Raw REST APIs Fail Under Autonomous Agents

REST APIs expect human engineers to write retry backoff, status polling, and tenant checks. When AI agents call raw endpoints through 1:1 MCP wrappers, three production failures occur:



- Non-Idempotent Retries (Double-Billing): When a POST/PATCH commits on the server but drops with HTTP 504, naive agents retry blindly and charge twice. Compiled idempotency keys and pre-retry lookups prevent duplicates.
- Async 202 State Divergence: Background endpoints returning HTTP 202 without polling state machines cause agents to hallucinate completion or trigger overlapping cluster restores.
- Cross-Tenant Parameter Injection (BOLA / IDOR): Flat paths (e.g., GET /devices/{deviceId}) without bound tenant parameters let prompt confusion leak or delete records across accounts.

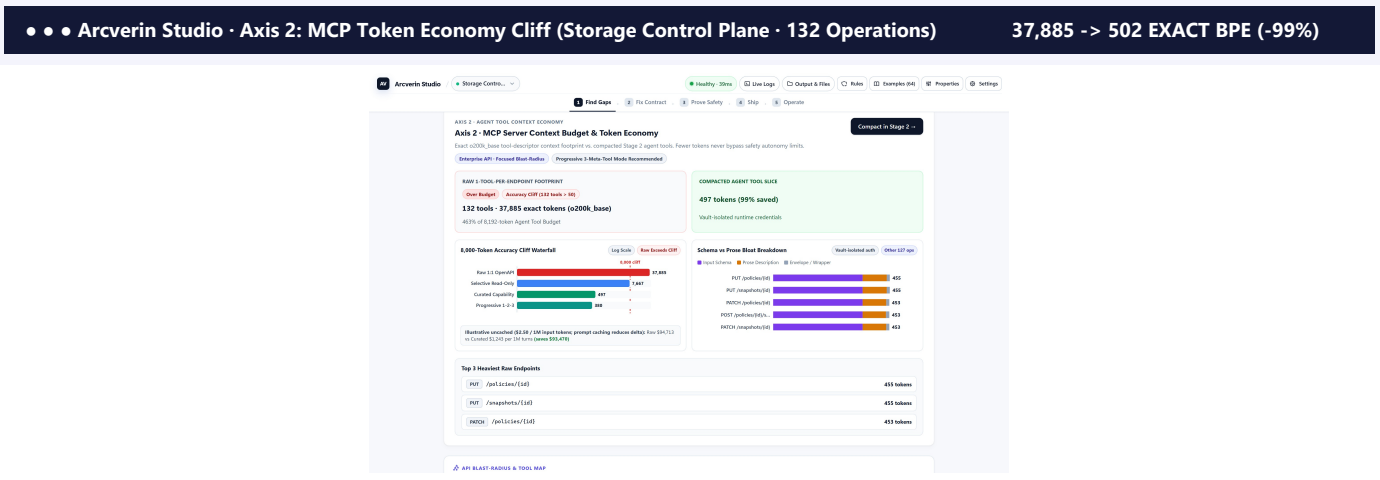


Figure 1: Arcverin Studio profiling 132 synthetic Storage Control Plane operations (reference-benchmarks.zip) on a log-scale accuracy cliff-cutting 37,885 raw o200k_base tool-descriptor tokens to 502 curated tokens (-98.7% exact / -99% displayed) or 380 upfront progressive tokens. The in-panel dollar figure is an illustrative uncached $2.50/1M baseline scaled down proportionally by prompt caching.

2. How Arcverin Complements Developer Portals, Enterprise Gateways, WAAP & Sandboxes

Enterprise IT API Management teams already govern APIs through Developer Portals (Layer7 Portal, MuleSoft Exchange, Apigee, APIM, Backstage), gateways (Broadcom Layer7, Apigee, MuleSoft, IBM API Connect, Azure APIM, Kong, Envoy, AWS Bedrock AgentCore), external WAAP/API-security scanners (such as Akamai API Security), and V8 Code Mode sandboxes. Arcverin complements all of them upstream in Stages 1-4-qualifying internal and external portal specs, synthesizing 1.2.0 Overlays, simulating 23 fault scenarios, and exporting open-standard Overlays/Arazzo/SARIF/OPA/Cedar plus native APIM/Kong/Envoy/AgentCore policy bundles:

Stack Layer / Requirement	OpenAPI CLIs & Workflow Brokers	Code Mode & V8 Sandboxes	Enterprise MCP Gateways	Arcverin Studio & vMCP Runtime
Primary Stack Role	Spec hygiene, live-traffic drift & open Arazzo hosts	Dynamic read loops & in-isolate payload filtering	Stage 5 edge routing, Entra/IAM auth & ACLs	Upstream Stages 1-4 proof layer + optional Stage 5 runtime
Contract Readiness & Autonomy Tiers	Static scoring (SARIF) & live drift; pairs with upstream Overlays	Generates typed TS APIs; benefits from pre-hardened specs	Enforces edge rules; leaves pre-runtime repair upstream	8 Pillars, 102 Rules, {value|unknown} gates, 1.2.0 Overlays & AR0-AR4 Runtime Tiers (AR0-AR6 Lattice)
Token Economy & Read/Write Lanes	Lint/run authored Arazzo; no BPE economy compiler	Excels at dynamic exploration over large read payloads	Proxies tools; pairs with compact upstream descriptors	~380+477 BPE Progressive (-99.4%) & deterministic Arazzo 1.1 write DAGs
Pre-Release Failure Proof	Live drift checks & happy-path workflow execution	V8 memory & egress isolation for generated scripts	Live edge traffic enforcement & rate-limit telemetry	23-scenario stateful mock & 12-mode fault injector + staging
Policy Exports & Signed Proof	Consumes standard Overlays, Arazzo 1.1 & SARIF 2.1.0	Two-lane handoff: Code Mode reads + compiled Arazzo writes	Consumes Bedrock+Cedar, Envoy, APIM, Kong & OPA bundles	Ed25519 DSSE Pack, 6-framework crosswalk & optional vMCP Vault

3. Deterministic 8-Pillar Qualification & Sandboxed Visual JDM Policy Studio

Each operation is scored offline against 102 deterministic rules across 8 pillars: Auth, Authorization, Tenant Isolation, Blast Radius, Idempotency, Async 202, Verification, and Human Approval. Unknown facts clamp to AR0 (Blocked) until resolved via an OpenAPI 1.2.0 Overlay or signed Owner Decision.

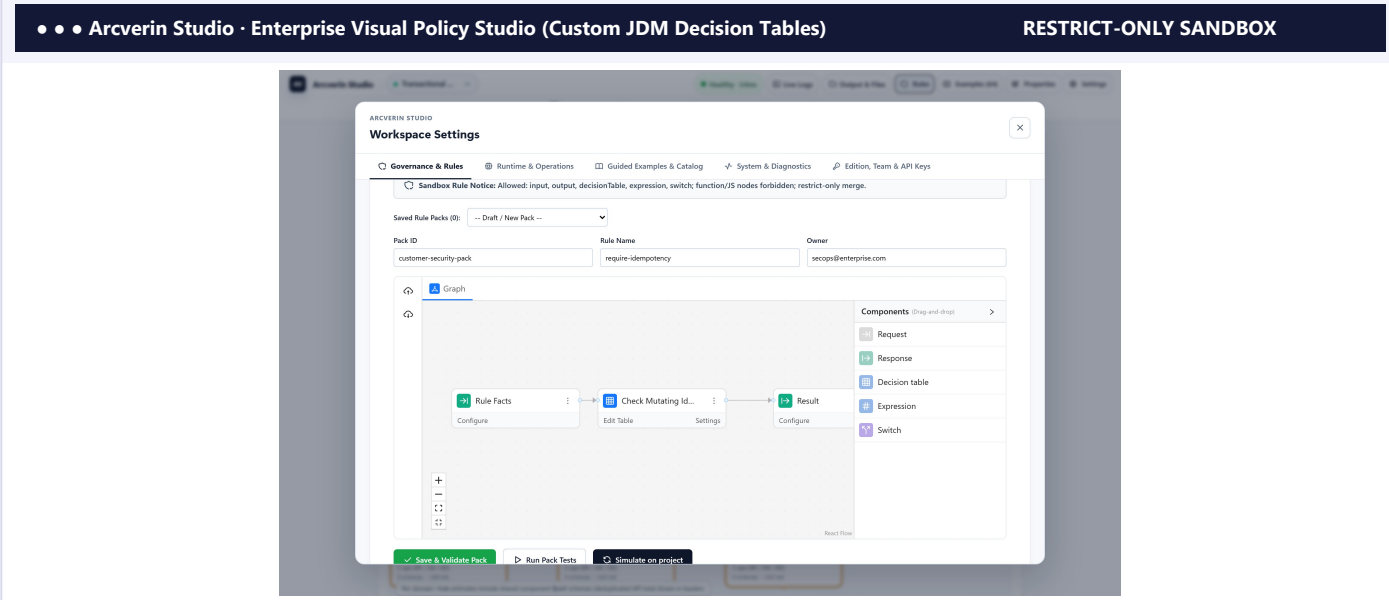


Figure 2: Enterprise Visual Policy Studio lets architects author Custom Visual JDM Decision Tables, run unit tests, and preview fleet impact. Rules run in a zero-script sandbox that can only tighten-never weaken-the 102 built-in rules.

4. Fail-Closed CI Release Gates, DSSE Evidence Packs & Gateway Exports

In Stage 4 (4. Ship), Quick Ship evaluates gate.yaml fail-closed and seals a SHA-256 DSSE Evidence Pack (Ed25519 or Customer-Managed Ed25519 Keys resolved as software seeds from your secret pipeline with key-URI provenance metadata, not remote HSM signing RPC). It also generates fail-closed 6-framework compliance crosswalks and exports portable Arazzo 1.1, OPA Rego, AWS Cedar, Temporal (starter skeletons, Team runnable TS/Python project bundles, and Enterprise Multi-System vMCP Temporal Sagas), Azure APIM, Kong, Envoy, and AWS Bedrock AgentCore bundles.

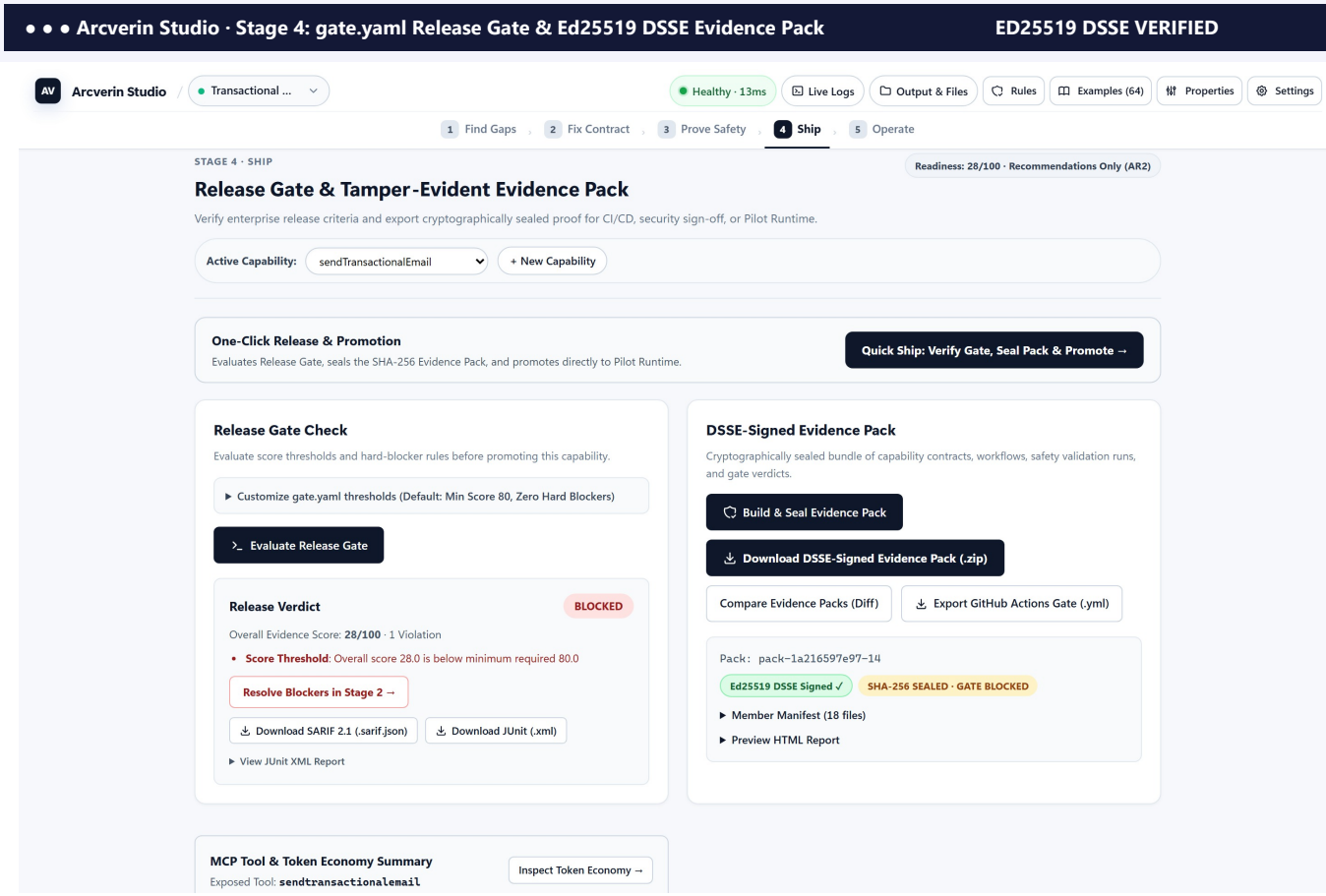


Figure 3: Stage 4 (4. Ship) enforcing fail-closed gate.yaml thresholds, exporting SARIF and JUnit CI reports, and sealing an offline-verifiable Ed25519 DSSE Evidence Pack.

Explore the Arcverin Technical Documentation & CLI Reference at https://www.arcverin.com/docs/ →

https://www.arcverin.com/docs/

Architectural scope note: Arcverin qualifies specifications, compiles contract guardrails, and executes bounded simulation/staging tests. Reference benchmarks use downloadable synthetic fixtures (not audited customer savings). Upstream backend services remain responsible for server-side authentication and resource authorization.