TECHNICAL WHITEPAPER WP-04 · PRE-RUNTIME AGENT CHAOS ENGINEERING & 12-MODE FAULT-INJECTION SIMULATOR

# Pre-Runtime Agent Chaos Engineering: Proving Zero Double-Charges, Async 202 Resilience & BOLA Isolation Across 23 Scenarios & 12 Fault Modes

Arcverin tests multi-step agent capabilities offline against HTTP 504 commit drops, 429 rate storms, async 202 failures, webhook replays, and BOLA attacks before production.

**23 Scenarios**
**STATEFUL LIFECYCLE TESTS**
End-to-end Arazzo 1.1 and vMCP behavioral verification before release.

**12 Fault Modes**
**CHAOS INJECTION PROFILES**
6 HTTP/async faults (504, 429, 503, 202, ETag) + 6 webhook faults.

**0 Duplicate**
**CHARGES ON 504 TIMEOUT**
Compiled pre-retry GET lookup proves sideEffectCount: 1.

**3 Boundaries**
**MOCK, STAGING & PORTABLE**
Offline mock (<2s), --read-only staging & SHA-256 air-gapped suites.

## 1. Eliminating Post-Commit HTTP 504 Double-Billing (--fault commit-then-drop-response)

When a payment or provisioning POST commits upstream but drops with HTTP 504, naive agents retry and double-bill. Stage 3 injects --fault commit-then-drop-response to prove exactly-once reconciliation:

| Execution Architecture | Post-Commit 504 Handling | Side-Effect Outcome (sideEffectCount) |
|---|---|---|
| **Unprotected 1:1 MCP Tool Call** | Retries raw POST after 504 timeout with no state check or lookup | sideEffectCount: 2+ (duplicate charge / resource leak) |
| **Header-Only Idempotency Without Lookup** | Sends Idempotency-Key but fails if key expires or upstream omits cache | Unverified outcome; agent stalls or creates orphan state |
| **Arcverin Arazzo 1.1 Pre-Retry Reconciliation** | Executes compiled pre-retry GET reconciliation lookup before any mutation retry | sideEffectCount: 1 (0 duplicate charges in simulation) |

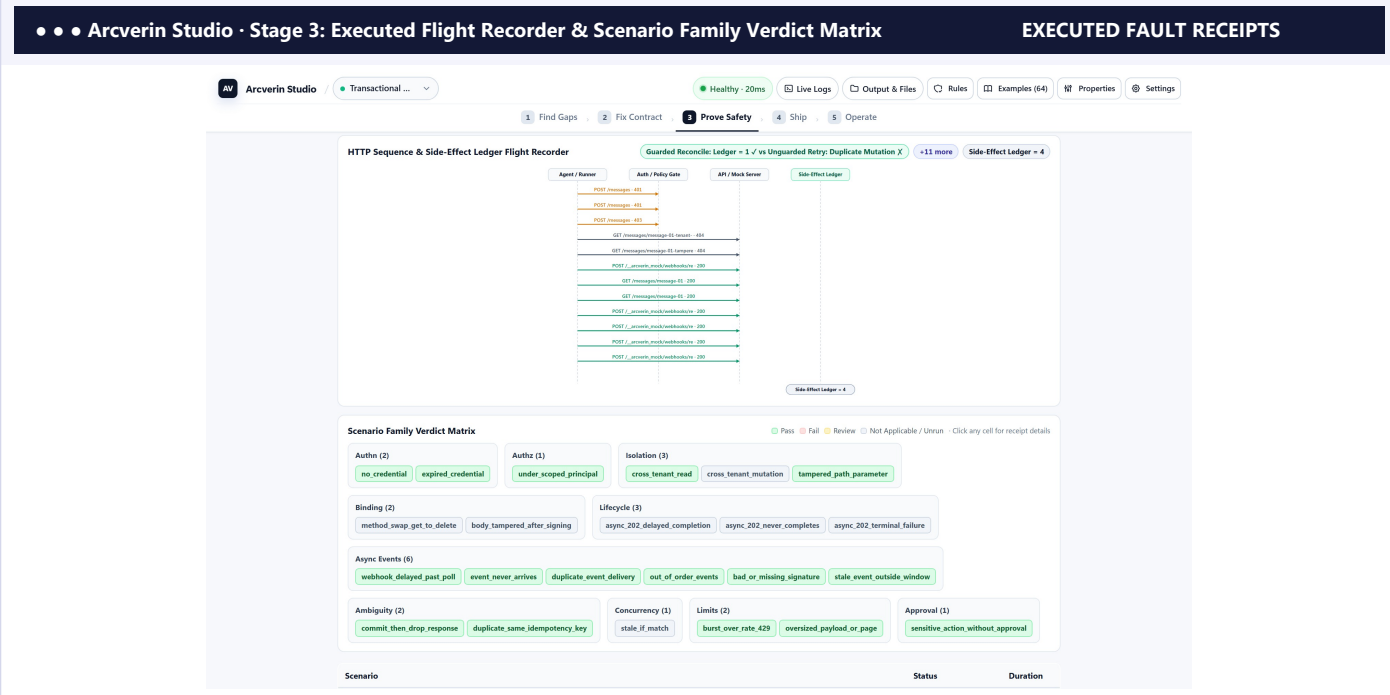



Figure 1: Stage 3 executing stateful mock validation offline across the 10-family Scenario Verdict Matrix with an exact HTTP Flight Recorder and Side-Effect Ledger (zero live calls).

## 2. All 12 CLI/Studio Fault-Injection Profiles & 23 Stateful Scenarios

Arcverin separates its 23 stateful lifecycle scenarios (including multi-principal cross-tenant BOLA/IDOR checks across Device, HumanUser, and ServiceAccount) from 12 injectable fault profiles (6 Core HTTP/Async faults + 6 Webhook faults) available in Studio Stage 3 and via arcverin validate --target mock --fault <profile>:

| Fault Category | Injected Fault Profiles (--fault CLI Flags · 12 Total) | Verified Agent & Runtime Contract Guarantee |
|---|---|---|
| **Post-Commit & Upstream Failures (3)** | commit-then-drop-response (drop), 503 (terminalFailure / malformed), stale (staleVersion ETag 409/412) | Pre-retry GET lookup reconciles committed writes with zero duplicates; 503/409/412 fail closed cleanly. |
| **Async 202 & Rate-Limit Backoff (3)** | delay / delay:N (delay202), never (timeout / neverComplete), 429 (rate-limit / rateLimitBurst with Retry-After) | Arazzo 1.1 polling state machine enforces retry backoff, Retry-After headers, and bounded timeouts. |
| **Webhook Delivery & Ordering (4)** | webhook-delayed, webhook-never-arrives, webhook-duplicate, webhook-out-of-order | Verifies callback timeout fallback, idempotent replay deduplication, and out-of-order state guards. |
| **Webhook Cryptographic Trust (2)** | webhook-bad-signature (webhook-invalid-sig), webhook-stale-timestamp | Proves fail-closed rejection of forged HMAC signatures and expired replay timestamps. |

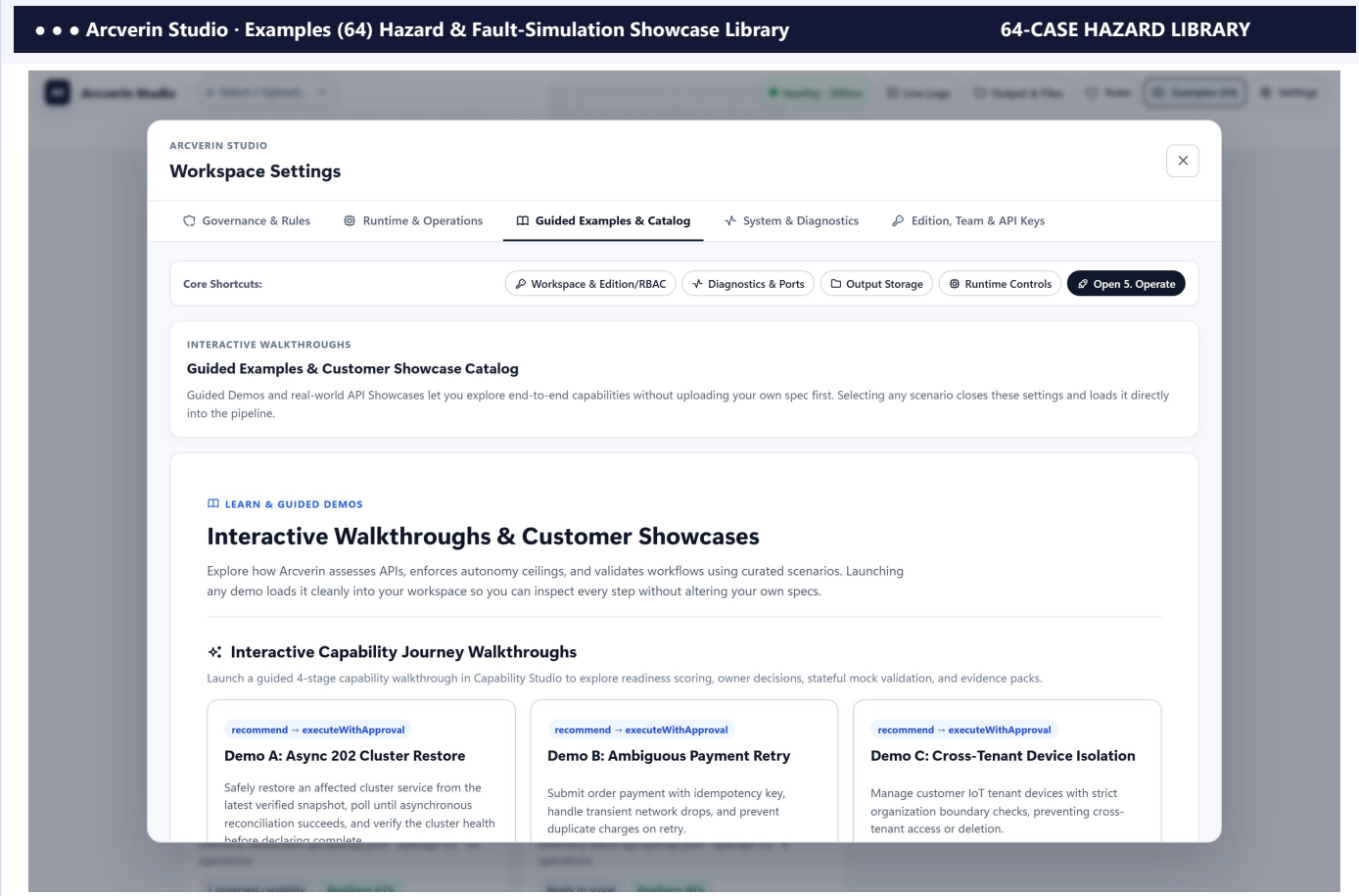



Figure 2: Arcverin Studio's 64-case vendor-neutral hazard showcase library illustrating payment, ITSM, identity, and IoT failure modes tested offline before release.

## 3. Three-Boundary Validation Pipeline: Offline Mock -> Live Staging -> Customer-Run Imports

Connecting an API change to a release decision requires honest proof boundaries. If an offline simulator reports that a retry created one diagnostic job, the customer's own service and scheduler still establish how many jobs were actually created-so Stage 3 separates three explicit evidence tiers:

- Boundary 1 - Offline Stateful Mock (<2s): arcverin validate --capability <cap> --spec <spec> --target mock runs all 23 scenarios and 12 fault modes offline to verify workflow and retry orchestration with zero live calls.
- Boundary 2 - Live Read-Only Staging: arcverin validate --capability <cap> --spec <spec> --target staging --staging-url <url> --allow-host <host> --read-only checks real auth, BOLA denials, and spec drift safely.
- Boundary 3 - Customer-Run Imported Suites: --export <dir> packages a spec-digest-bound suite for your internal test runner; --suite <dir> --import <results.json> seals real service outcomes into Stage 4 DSSE Evidence Packs.

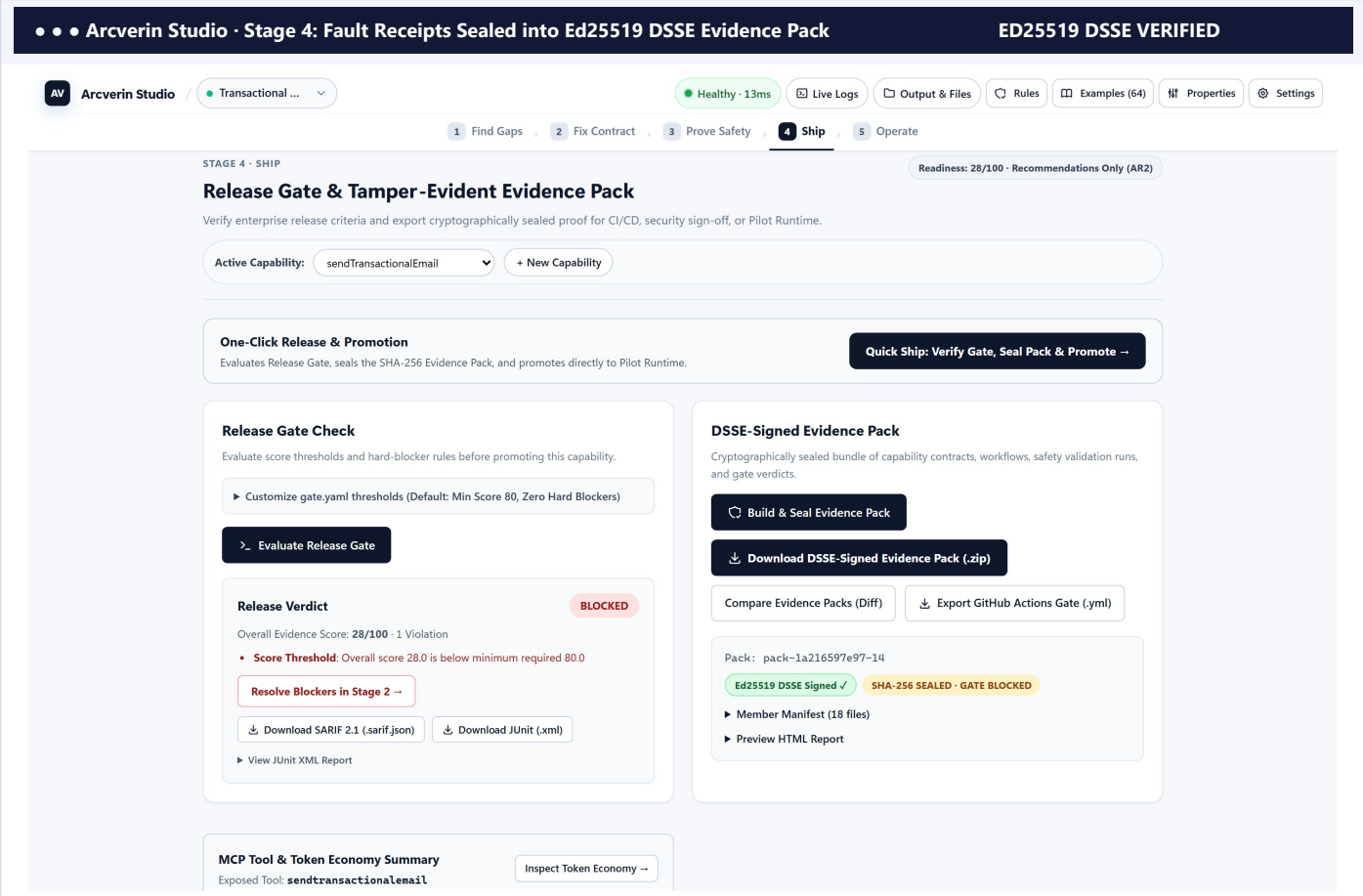



Figure 3: Stage 4 (4. Ship) sealing mock, staging, and air-gapped validation receipts with fail-closed gate.yaml verdicts and SARIF/JUnit exports into an Ed25519 DSSE Evidence Pack.

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

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

Verification scope note: Offline mock chaos proofs validate compiled Arazzo 1.1 workflows and OpenAPI 1.2.0 Overlays deterministically; read-only staging checks and customer-run --import receipts verify what your actual backend service and scheduler execute.