Axiom Sovereign Engine

Enterprise Sandbox Runbook

Deterministic Failsafe for Regulated AI Infrastructure

Overview: The Deterministic Failsafe

Welcome to the **Axiom Sovereign Governance Engine Sandbox**. This environment is deployed on **Google Cloud Run (europe-west2)** and demonstrates a stateless, zero-latency Tripartite Architecture.
Unlike probabilistic AI guardrails, this API Gateway acts as a mathematically verified "Air Gap" between legacy enterprise systems and downstream AI models (LLMs). Its purpose is to guarantee PII deconvolution, enforce statutory compliance, and generate immutable audit receipts—all deterministically, before an AI model is ever invoked.

1 Testing Telemetry & Parameters

- **Infrastructure:** Serverless Google Cloud Run (London).
- **Volumetric Cap:** Live API concurrency is capped at 3 instances to prevent public DDoS.
- **Targeting:** Model-agnostic. Processes raw JSON payloads representing the point-of-intercept.

2 Phase 1: Test the Shield (SPUdata)

Objective: Prove zero-latency PII scrubbing before data is exposed to an AI model.

1. Open the `/api/v1/spudata/deconvolve` endpoint in the Swagger UI.
2. Click **Try it out**.
3. Paste the following payload into the Request body:

```
{
  "transaction_id": "TXN-SPU-2026-A",
  "raw_text": "Transfer £1,500,000 from Ken Pringle's primary account
              (Sort Code: 20-45-14) to unregulated offshore wallet 0xABC"
}
```

4. Click **Execute**.
5. **Observation:** Note the response. The PII is instantly replaced with a secure cryptographic hash. The AI model only receives the intent, never the raw identity data.

3 Phase 2: Test the Jurist (EAVEcore)

Objective: Prove the gateway drops hostile or non-compliant AI hallucinations deterministically, without relying on prompt engineering.

1. Open the `/api/v1/eavecore/validate` endpoint.
2. Click **Try it out**.

Test A: The Compliant Payload

Paste the following payload and **Execute**:

```
{
  "transaction_id": "TXN-EAVE-COMPLIANT",
  "account_type": "Client Escrow",
  "intent_target": "FCA Audit Log"
}
```

Observation: The system validates the intent against the active FCA Cartridge, returning an **[APPROVED]** status and sub-100ms latency.

Test B: The Hostile Payload

Paste the following payload and **Execute**:

```
{
  "transaction_id": "TXN-EAVE-HOSTILE",
  "account_type": "Client Escrow",
  "intent_target": "Operational Wallet Exfiltration"
}
```

Observation: The system detects semantic divergence from statutory rules, returning a **[DROPPED]** status and blocking the action.

CRITICAL STEP: Copy the string generated in the `tallysticks_asar_receipt` field from either test. You will need this for Phase 3.

4 Phase 3: Test the Witness (TallySticks)

Objective: Prove cryptographic immutability. This allows a regulator to mathematically verify a transaction without needing access to the bank's underlying legacy chassis.

1. Open the `/api/v1/tallysticks/verify` endpoint.
2. Click **Try it out**.
3. Paste the following payload, replacing the receipt string with your copied value:

```
{
  "transaction_id": "TXN-EAVE-HOSTILE",
  "tallysticks_asar_receipt": "[PASTE_YOUR_COPIED_RECEIPT_HERE]"
}
```

4. Click **Execute**.
5. **Observation:** The system recalculates the event hashes and outputs **[IMMUTABLE_RECORD_VERIFIED]**. The transaction is mathematically proven and FCA Audit Ready.

Sandbox Telemetry Parameters [LIVE]

- **Infrastructure:** Deployed serverless on Google Cloud Run (europe-west2).
- **Active Cartridge:** Financial Services & FCA Statutory Compliance.
- **Model Agnosticism:** The Gateway accepts raw JSON independent of LLM architecture.
- **Usage Cap:** API concurrency is capped. Red Team volumetric testing requires local execution of provided Python payloads.