Reference Architecture

Runtime PEP Integration Patterns

Strategic goal: Eliminate integration objections

Architectural Positioning

RegulatedRuntime operates as a **lightweight enforcement substrate**, not a framework.

It is deployed as:

- a sidecar or middleware layer
- between the **Orchestrator (App Layer)** and the **Tool Gateway (Infra Layer)**

It does **not** replace:

- models
- governance tooling
- identity providers
- observability platforms

Execution Flow

1. Agent proposes an action
2. Runtime evaluates:
    - identity and delegation
    - capability scope (read vs write, create vs delete)
    - data classification and purpose
    - active policy version
    - current risk posture
3. Enforcement decision:
    - allow
    - require approval
    - downgrade to read-only
    - block
4. Execution occurs or safely fails
5. Evidence artifact is committed asynchronously

Decision-Time Controls

Authorization Semantics

- Agent identity is distinct from user delegation
- Capabilities are operation-scoped
- Privilege inflation is prevented by default

Retrieval Governance

- Only provenance-tagged chunks are eligible
- Staleness budgets enforced before inference
- Prevents "wrong truth" from entering reasoning

Containment Primitives

Safe Mode

Automatically degrades agent authority when:

- injection signals spike
- anomaly thresholds are crossed
- upstream systems destabilize

Kill Switch

A global mechanism to disable all agentic writes:

- no redeploy
- no application changes
- immediate containment

Performance Characteristics

- Enforcement overhead designed for sub-20ms
- Evidence capture is asynchronous
- No dependency on model availability
- Fail-closed semantics for regulated writes

Non-Goals

RegulatedRuntime is not:

- a policy authoring UI

- a model wrapper
- a governance dashboard
- an observability replacement

It is **runtime infrastructure**.