SSAP Technical Overview

Decision architecture for controlled AI execution

This document explains **how SSAP makes decisions** without exposing internal code. It describes decision paths, telemetry, governance, deployment, and how pilots are measured.

Web vs PDF: The website answers *what decision is being made*. This PDF explains *how the decision is made*.

1. What SSAP is

SSAP is a **decision system** for modern AI. For every request, it chooses an explicit outcome:

• **NO_ACTION** — do not execute AI; return a deterministic response shape.

• **LIGHT** — attempt a cheap path first with bounded risk.

• **FULL** — run full inference only when justified by policy and confidence.

InferenceGate and SupportGate are **ways to use SSAP** (decision surfaces). They are not separate products: they share the same decision architecture and telemetry, but operate on different inputs and constraints.

2. SS → AP (high level)

Spectral Selection (SS) narrows the action space: it applies policy, risk checks, and context filters before any expensive work happens.

Asymmetric Probing (AP) performs a cheap attempt when allowed. If guardrails fail or confidence is insufficient, it escalates to FULL.

The key outcome is not prediction; it is **controlled execution**. NO_ACTION is a valid, auditable result.

3. Decision paths

Every request ends with an explicit decision path. The path is part of the API contract and can be logged for governance. Paths are stable, human-readable, and designed for audit.

Path	Meaning	Typical use
NO_ACTION	Do not execute AI. Return a deterministic output shape.	Policy blocks, low-value queries, unsafe contexts, missing inputs, hard constraints.
LIGHT	Try a cheap attempt first with bounded risk. Escalate only if needed.	Intent detection, classification, routing, quick summarization, structured extraction.
FULL	Run full inference when justified by policy and confidence.	High-value tasks, complex reasoning, critical customer workflows, long-form generation.

Recommended response behavior

NO_ACTION: Return a deterministic response shape (for example: safe fallback, cached result, or tool-only answer).

LIGHT: Return the cheap attempt result. If the cheap attempt fails its guardrails, escalate to **FULL**.

FULL: Return the full inference result. Optionally run post-decision validation when required by policy.

4. Telemetry and governance

SSAP produces structured telemetry by default so outcomes can be audited. Telemetry is designed to work without storing raw prompts or completions.

Typical telemetry fields:

• request_id, tenant_id (anonymized), endpoint/path

• decision_path (NO_ACTION / LIGHT / FULL) and decision reasons

• latency_ms (per stage), tokens/cost estimates per path

• policy blocks and risk flags (no sensitive content)

Governance teams can use telemetry for: audit trails, policy enforcement evidence, drift monitoring, and change reviews.

5. Deployment model

SSAP is deployed as a drop-in decision layer in front of execution. It preserves the client-facing API contract while adding decision outcomes and audit telemetry.

Common patterns:

• API gateway / proxy layer

• service wrapper (library/SDK) inside an existing backend

• staged rollout per endpoint (start with low-risk surfaces)

6. Pilot measurement (no promises)

Pilots are scoped to validate decision quality and operational impact. The goal is to measure outcomes using telemetry, not to promise specific savings. Your baseline, traffic, and policy choices determine results.

Metric	What it shows	Typical interpretation
Decision distribution	Share of NO_ACTION / LIGHT / FULL.	Whether execution is being controlled and where value thresholds are landing.
Cost and tokens	Total and per-path spend.	Whether savings occur and which policies drive them.
Latency	Median and tail latencies by path.	Whether the decision layer stabilizes tail risk.
Quality evidence (optional)	Sampled shadow FULL judged vs accepted outputs.	Whether cheap attempts remain within an acceptable quality envelope.
Safety and governance	Policy blocks, risk flags, audit completeness.	Whether non-execution is correctly enforced and explainable.

Pilot framing: A pilot is successful when the decision system produces measurable governance and operational benefits (cost stability, latency control, risk reduction) without breaking UX. Results vary by surface and policy.

7. Next steps

If the technical overview matches your surface, the next step is a short fit check to confirm endpoints, traffic, constraints, and success metrics.

Contact: marko@ssap.io