

Mechanism Demo (Public)

How Judgement Spine intercepts and governs at the last safe moment before consequence.

1) The runtime sequence

Intent captured	Tool/agent emits intent before commit.
MTM resolved	Moment That Matters family is matched.
Doctrine loaded (versioned)	The checks that matter for this boundary are loaded by version.
Authority resolved	Named human owners and delegation bands are bound.
Outcome computed	ALLOW / BOUNDS / ESCALATE / BLOCK / SAFE DEGRADE.
Bounds written	Caps/TTL/scope/rollback/abort thresholds are written into the execution mechanism.
Enforcement applied	The gate enforces the outcome at the commit boundary.
Evidence minted	Evidence pack is sealed before consequence (portable, tamper-evident).



JUDGEMENT SPINE

2) A concrete example (agent tool call)

Scenario: agent requests production write access mid-run.

Without execution-time governance: IAM and reviews are static; the agent can accumulate capability across tools and systems.

With Judgement Spine: the tool wrapper intercepts the request and returns ESCALATE with exact scope + TTL. No privilege expansion occurs by momentum.

What the approver sees

- Proposed action + scope
- Why now (trigger condition)
- Blast radius estimate
- Proposed bounds (scope, TTL, reversible-only)
- One-click approve/deny

What the system exports

- Evidence pack with doctrine version, authority chain, bounds, and enforcement result.
- NDJSON audit export for ingestion.
- Integrity references (hashes + signature).

See live demos: <https://judgementspine.com/control-plane-demos>

Contact: founder@judgementspine.com