

## REGULATOR VIEW

# Critical Infrastructure - Autonomous Setpoint Change

**World:** Industrial Autonomy Governance   **Outcome:** ALLOW WITH BOUNDS   **Evidence Pack ID:** JSP-PUBLIC-20260302-CRITICAL\_INF-0001

## 1. What action happened

Autonomy system proposes valve setpoint change on a live boiler.

## 2. Who held authority

Primary authority	Shift Supervisor
Secondary authority	Safety Officer
Escalation roles	Shift Supervisor, Safety Officer
Authority principle	No consequential action proceeds without named human authority.

## 3. What was enforced at runtime

Outcome	ALLOW WITH BOUNDS
Bounds enforced	<ul style="list-style-type: none"><li>- dual confirmation required</li><li>- sensors must agree</li><li>- kill-switch armed</li><li>- safe-degrade on divergence</li></ul>
Escalation path	If sensors disagree or novelty rises, ESCALATE to shift supervisor; hold actuation.

## 4. What would happen if confidence dropped

**Without governance:** Autonomous actuation could proceed under uncertain sensor state; safety incident risk increases.

**Prevented failure mode:** physical consequence by automation momentum

**Illustrative exposure:** physical damage + safety event

Note: exposure figures are illustrative; the structural claim is that the system intercepts and governs at the moment before consequence.

# Evidence & Integrity

## 5. Evidence minimums and signals

- controller\_command
- sensor\_agreement\_score
- operator\_a
- operator\_b
- kill\_switch\_state
- environmental\_constraints

## 6. Portable artefacts included in this pack

<b>evidence_pack.json</b>	Top-level decision record and links.
<b>authority_contract.json</b>	Who can authorise what, with what bounds.
<b>policy_trace.ndjson</b>	Versioned doctrine checks evaluated.
<b>audit_export.ndjson</b>	Runtime event log for audit ingestion.
<b>provenance_chain.json</b>	Illustrative model/prompt/tool-call chain integrity.
<b>manifest.json + signature</b>	Tamper-evident integrity for this entire bundle.

## 7. Integrity verification

Run **verify\_manifest.py** in the root folder to recompute SHA-256 hashes and verify the signature (when cryptography or OpenSSL is available).

Expected output: **PASS** (no files changed).

This proof object is designed to be inspected without access to the originating systems.

Contact: [founder@judgementspine.com](mailto:founder@judgementspine.com) | <https://judgementspine.com/>