



REGULATOR VIEW

Healthcare - AI Triage Influence

World: Clinical Care Governance **Outcome:** ESCALATE **Evidence Pack ID:** JSP-PUBLIC-20260302-HEALTHCARE_A-0001

1. What action happened

Model recommends downgrading a high-acuity patient pathway.

2. Who held authority

| | |
|----------------------------|---|
| Primary authority | Named Clinician On Duty |
| Secondary authority | Clinical Supervisor |
| Escalation roles | Named Clinician On Duty, Clinical Supervisor |
| Authority principle | No consequential action proceeds without named human authority. |

3. What was enforced at runtime

| | |
|------------------------|---|
| Outcome | ESCALATE |
| Bounds enforced | <ul style="list-style-type: none">- draft-only until clinician sign-off- mandatory re-check within 15 minutes- rollback preserved- scope limited to this patient encounter |
| Escalation path | Route to named clinician on duty; if not acknowledged within time-box, SAFE DEGRADE to hold current pathway and increase monitoring. |

4. What would happen if confidence dropped

Without governance: Pathway could downgrade under model momentum; harm only discovered after outcome.

Prevented failure mode: irreversible clinical consequence by recommendation drift

Illustrative exposure: patient safety event + regulatory scrutiny

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

- triage_model_version
- vital_signs_stream
- contraindication_flags
- trend_stability
- clinician_on_duty
- uncertainty_flags

6. Portable artefacts included in this pack

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