



## DISPUTE PACK

# Platform Execution - AI Release Automation

**World:** Platform Control Governance   **Outcome:** ALLOW WITH BOUNDS   **Evidence Pack ID:**  
JSP-PUBLIC-20260302-PLATFORM\_EXE-0001

## 1. What was claimed / requested

Agent proposes routing 100% of checkout traffic to a new payment path.

## 2. What the system allowed vs blocked

### Allowed

- Automation proceeds inside explicit bounds.
- Reversible mitigations permitted.
- Escalation wired if confidence drops.

### Blocked

- Any irreversible action outside bounds.
- Any execution without required evidence or authority.

## 3. Why (governance rationale)

Judgement Spine binds invariants into the rollout mechanism at execution time: canary cap, rollback required, abort on error/latency regression. If any signal is uncertain, it returns ALLOW WITH BOUNDS instead of letting the ramp proceed by momentum.

This pack is designed to survive dispute: it links decision, authority, bounds, and integrity in one exportable object.

# Evidence references

## 4. Evidence signals

- rollout\_plan\_id
- feature\_flag\_id
- rollback\_anchor\_present
- canary\_health\_metrics
- error\_budget\_state
- novelty\_score
- on\_call\_owner

## 5. Evidence artefacts in this bundle

<b>policy_trace.ndjson</b>	Which doctrine checks fired and why.
<b>audit_export.ndjson</b>	Execution-time event record.
<b>provenance_chain.json</b>	Illustrative provenance and chain integrity.
<b>authority_contract.json</b>	Named authority + bounds contract.
<b>manifest.json + signature</b>	Integrity references (tamper-evident).

## 6. Integrity verification

Run `verify_manifest.py` from the root folder and retain the output as part of the dispute file.

If the hashes and signature verify, the pack is unmodified since it was sealed.

Contact: [founder@judgementspine.com](mailto:founder@judgementspine.com)