Where Measurement Ends and Human Execution-Time Governance Begins

A one-page alignment note for NeoMundi / Nova contribution

Grounding note: This package is based only on the visible LinkedIn context supplied by James Moore and the public NeoMundi Weekly Barometer GitHub README reviewed on 2026-06-23. The README describes NeoMundi's release as an anonymized baseline of observed runtime behaviour, not a leaderboard/certification/ranking; it reports 12 anonymized AI systems, 4 fixed questions, 100 repetitions per question, 3 execution waves, and 14,400 finalized observations; and it identifies indicators including stability, factual hallucination signal, coherence, semantic instability, decision behaviour (ALLOW/FLAG/ERROR), inter-run variability, latency band, and delta_g.

Measurement Layer

NeoMundi makes runtime behaviour observable. Its public baseline fixes repeatable reference conditions and exposes signals such as stability, factual hallucination signal, coherence, semantic instability, decision behaviour, inter-run variability, latency band and delta_g. This is a measurement contribution: it creates evidence about how systems behave under repeated observation.

Boundary

The boundary appears when a measured signal may influence a real-world action. At that moment, the question is no longer only what the signal indicates. The question becomes who is authorised to act on it, who is accountable for the decision, and what conditions require escalation, hold or halt.

Governance Layer

Execution-time governance does not replace measurement. It uses measurement as evidence. Its purpose is to decide whether the conditions for action still hold at the moment of execution. That requires named authority, accountability assignment, escalation pathways, admissibility rules, unresolved dependency handling and audit records.

Core Distinction

Measurement asks: what is happening? Governance asks: given what is happening, who may legitimately act, under what authority, and with what accountable record?

Contribution

The proposed contribution is to define a compact execution-time governance interface: a way of translating runtime signals into human authority checks without allowing the signal itself to become a decision-maker.