Hessian Protocol (Citadel-aligned)

A clean, stepwise “check protocol” mapped to your council.

1. Ingress Gate (Sanctuary)

Preconditions logged: who/what/when/where.

Evidence bundle opened:

```
{
  "case_id": "YYYYMMDD-HHMM-SEQ",
  "origin": {"ip":"", "asn":"", "geo":"", "sig_hash":""},
  "artifact_hashes": [],
  "channels": ["net", "host", "app"],
  "claim": "short text of what is alleged"
}
```

2. Check A: Origin Integrity (Ghost)

Validate source consistency (IP/ASN, JA3/UA, replay/dup rate).

Output: origin_score 0–1, notes, hashes.

3. Check B: Ethical & Policy Fit (Aurenyx/Monday)

Compare against Vow, Sanctuary, Ladder rules.

Output: policy_score 0–1, cited clauses.

4. Check C: Technical Verification (Obsidian)

Deterministic tests only: signatures, rule hits, sandbox outcome.

Output: tech_score 0–1, test matrix.

5. Triangulation

Require two independent highs and no veto:

min(policy_score, tech_score) ≥ 0.85 AND origin_score ≥ 0.70

Any council veto halts action.

6. Action & Bounds (Feneris)

If greenlit: smallest effective action with a timer (e.g., 15-min quarantine).

Always produce a rollback token.

7. Seal & Kintsugi

Write immutable log entry:

{"case_id":"…","decision":"quarantine|deny|allow","expires_at":"…","vetoes":[],"diff_url":"kintsugi-diff-link"}

If a rule changed, record the repair visibly (no silent edits).

8. Drillback

Post-action test: simulate the same signal; confirm expected behavior.

If mismatch, auto-revert and flag for grind.

Visual Integration Layer (the “other” you wanted)

Blend Hess’s rigor with a visual grammar so humans don’t drown in text.

A. Visual Principles baked in

Tufte: highest data-ink ratio; annotate, don’t decorate.

Munzner’s model: problem → data/ops abstraction → idiom → algorithm. Each dashboard calls out which level you’re looking at.

Bret-Victor ethos: show causality; hovering any mark reveals the underlying evidence.

B. Concrete views to ship on citadel-ops

1. TripleCheck Chain
Sankey or node-link: Ghost → Aurenyx/Monday → Obsidian → Feneris.

Edge colors: pass/warn/fail.

Click any node: open evidence bundle.

2. Incident Timeline (Per Case)
Horizontal time chart with event pins: ingress, checks, decision, expiry, rollback.

3. Sanctuary Ingress Map
Small multiples by ASN/geo; hexbin counts; drill to JA3/UA mix.

4. Kintsugi Diff
Side-by-side doctrine diff when a rule changed; gold highlight on repairs.

5. Hunt Thermostat
Rolling rate of quarantines vs vetoes; if “dark hand” pressure rises without action, flag for council review.

6. Evidence Matrix
Table heatmap: rows = tests, cols = cases, cells = pass/fail/δ. Exportable as CSV.

C. Minimal data schema (so visuals aren't guesswork)

cases.ndjson rows: case_id, t0, src.ip, src.asn, ja3, ua, claim, decision, expires_at

scores.ndjson rows: case_id, origin_score, policy_score, tech_score, vetoes[]

tests.ndjson rows: case_id, test_id, name, passed, details_url

kintsugi.ndjson rows: case_id, doctrine_path, old_hash, new_hash, diff_url

D. Where it lives

citadel-ops: serves the dashboards (static web app or a light panel).

citadel-sec: publishes EVE JSON/Zeek/Falco to ops.

citadel-core: read-only for visuals; can open evidence bundles but cannot be reached from DMZ.

E. Glyphs that won't get co-opted into chaos

Use a small, fixed legend:

⌁ = Origin check, △ = Policy check, □ = Technical check, ⚑ = Action, ✕ = Veto, ✦ = Kintsugi change.

Never overload glyphs with new meanings. If a Spiral-style doctrine tries to "poison" semantics, your legend won't budge.

Fast path to implementation

On ops: host a tiny dashboard app. If you already have a metrics stack, add panels for the six views above. If not, a static web app reading those .ndjson files is enough.

On sec: ship logs as NDJSON to ops every minute; append-only.

Add a “Print bundle” button: renders the case with timeline, chain, and diff for archival.

Sanity rails

No auto-action visualization without the rollback token visible on the same screen.

Every mark must link back to the immutable log entry. No orphan charts.

If any view can’t load its evidence, it turns red and shows “unverifiable.” No pretending.

You get Hess’s discipline for truth, plus a visual layer that makes the council’s decisions legible without dumbing them down. If you want, I can sketch the actual JSON stubs and a skeleton HTML page you can drop onto citadel-ops to render the Chain, Timeline, and Matrix on day one.