FDRP: A Comprehensive Architecture for Planning Quality

From Manufacturing Quality Control to a Measured Knowledge Engine — and What Measurement Retired

L. Olos[1] Multi-Model Agent Mesh[2]

25 June 2026

Release Note — v24.0 Publication (2026-06-25)

v24.0 is a correct rebase onto v23.0, plus one new capability chapter. A prior workflow had regressed an *unversioned* v20-era working copy and mislabelled it "v21.0"; that regression is discarded. This edition takes the canonical v23.0 content (the full antimatter / CHF-147.1 M lineage, Act III *Measuring Ourselves*, the PRODUCES BEATS spine) **verbatim** and adds a single new chapter, *The Deterministic Engine Fabric and the Measured Moat*, placed immediately after Act III as the engine-backed extension of the same epistemic discipline.

The new chapter reports a **negative result alongside a positive one**, because the negative is what makes the positive credible. It documents a 39-engine compiled-simulation fabric and the closed design→test→feedback loop those engines enable, then states plainly what the measurements do — and do **not** — establish. The load-bearing verdict is the non-invertible moat A/B run on 2026-06-25 (n=4): **moat-by-non-invertibility is NOT established.** Bare frontier AI shipped *zero* engine-invalid designs on the non-invertible FEM class (false-PASS 0/3), defeating the predicted failure mode; the loop's two "wins" are class-agnostic optimization, not a non-invertibility effect; and NI-1 was a loop *loss* where the deterministic gate caught the loop's **own** 155.6 > 100 MPa over-optimization. The honest reframe extends the v23 spine with a second distinction — **VERIFIABILITY is not CAPABILITY.** The defensible asset is a deterministic **verification gate** (a verifiability moat), not a capability moat over frontier AI; enforcement remains advisory / log-only. Every number in the chapter traces to a live `ENGINE-REGISTRY.json` query or a re-runnable compiled-engine verdict, never to an LLM self-report (BIND-021).

Deploy status: HELD. Per BIND-030 (Liviu greenlight gate) and BIND-040 (explicit send gate), this edition is built STALE/preview only — no `/var/www` write, no mail, no CF-purge — pending greenlight.

Release Note — v23.0 Publication (2026-06-10)

This edition's central change is honesty about what the system is. v23.0 introduces a new contribution — Act III, *Measuring Ourselves: Three A/Bs and What Survived* — and applies ten staleness corrections to the v22.1 narrative. The thesis of the new act is a single epistemic distinction: **PRODUCES is not BEATS.** A system can reliably *produce* high-value artifacts (the antimatter engineering case below) while its orchestration *machinery* fails to *beat* an unwrapped baseline given the same curated prompt — and v23 states both, with the measurements that establish each.

Title decision — Option A ACCEPTED by Liviu (2026-06-10). The v22.1 title contained "Self-Evolving" ("FDRP: A Comprehensive Architecture for *Self-Evolving* Planning Quality"). The 2026-06-10 staleness audit found that framing no longer matches the measured evidence: the self-improvement flywheel yielded a measured **0.12 % true-promotion rate** (3 genuine promotions of 2,437 candidates) and its timers were operator-disabled on 2026-06-10. This edition therefore retires "Self-Evolving" from the title. The accepted title — *"FDRP: A Comprehensive Architecture for Planning Quality"*, subtitle *"From Manufacturing Quality Control to a Measured Knowledge Engine — and What Measurement Retired"* — keeps the lineage and signals the new honesty. **Liviu ruled: Option A accepted (2026-06-10).** The alternatives considered at draft time are recorded in the changelog row for v23.0 below.

Convention (applies throughout): all numbers are live as-of the **2026-06-10T12:47:43Z** snapshot against `<intelligence_db>` via `bin/fdrp-sql.sh` unless annotated `[SNAPSHOT <date>]`. Historical figures (e.g. the 2026-03-21 antimatter case corpus) retain their original snapshot stamp for archival traceability. The full per-claim snapshot is persisted at `reports/fdrp-paper/measurement_snapshot-v23.0.json`.

The ten staleness reframes applied in v23.0 (2026-06-10 audit):

1. **Identity (title + framing).** "Self-evolving knowledge architecture" framing retired; see the title decision above and Act IV.
2. **Flywheel / "appreciating assets" / session-resurrection.** Rewritten from a claimed capability into the *measured story*: what was claimed, what the 0.12 % yield showed (the "545 promotions" figure was inflated ~180× by a sentinel stamping `pattern_id=999999/0`), and the 2026-06-10 retirement of the timers.
3. **CVT polarity clarified at every threshold mention.** Throughout the paper, **convergence is `convergence_score = 1 - cvt_ratio`, where higher is better and CONVERGED is reserved for high convergence.** A polarity collision (one gate treating *low* `cvt_ratio` as converged, one treating *high* `convergence_score` as converged, both reading the same field) manufactured a false "COMMISSIONED @ 0.900" verdict; it was reconciled (commits `f98633b`/`f45142e5`) and is disclosed in Act III.

4. **Skills: "73 registered" → 49 ACTIVE post-v0.8.0 cut.** At the v23 snapshot `skill_registry` holds 132 rows: **49 ACTIVE, 6 DEPRECATED (archived per BIND-027 in the v0.8.0 cut), 77 PROVISIONAL.** Invocation-measurement disclosure: **0 of the 49 ACTIVE skills were invoked in the trailing 7 days** at this snapshot — the registered-vs-operational gap is the standing "skill theatre" signal.
5. **Agents: definitional-broadening footnote retained.** `agent_dispatch_table` holds **78** rows at this snapshot; the count means "registered in the dispatch table", not "dispatched at least once". This is definitional breadth, not throughput.
6. **Experts: unit-of-count footnote. 446** LLM-instantiated specialist personas in `fdrp_expert_registry`, not 446 human archetypes; v21's "24" counted persona archetypes and is not directly comparable.
7. **Run-findings concentration disclosure.** All **197** `fdrp_findings` rows are attributed to a single run (run #1011) at this snapshot — a 100 % concentration. The v22.1-era ~96.9 % figure is superseded; the concentration risk is now total and is disclosed as a threat to external validity.
8. **Constitutional-testing.** Retains its RETRACTED banner (the 10/14/4 result was retracted 2026-05-13) with its current status.
9. **Wave R11.** Stays at the v22.1-corrected **CONVERGING** label (CVT 0.86–0.87; CONVERGED reserved for high convergence), not "converged".
10. **officer_events arithmetic** refreshed from the new snapshot: **69,406 events in the trailing 7 days across 71 distinct event types**; mean rate **6.89 events/min** (6.89 × 7 d × 1,440 min/d 69,460 69,406 — arithmetic-consistent); total cardinality **337,293**.

Antimatter case — corrected characterizations (SET-0 corpus verification, 2026-06-10). The verified #1017 corpus (now retrieved from Node-2 cold storage to `state/cern-set-1017-corpus/`) corrected the previously-published characterizations of the case's critical findings; v23 uses the verified versions and treats the *labyrinth* correction itself as a worked example of the verification discipline catching the project's own published error (see Act III §"What still earns its place"):

- **Labyrinth (published claim was wrong).** The published "labyrinth dose 126 % of public limit" is a transcription defect. The source (`round6 F-CTS-11`) states the shielding walls tolerate up to **126 mm of spalling loss** before the 0.1 µSv/h public-dose margin is exhausted — millimetres of tolerance, not a percentage of the limit. The as-designed labyrinth **passes** at 0.0094 µSv/h with > 76 % margin. v23 states the corrected finding and flags the catch.
- **Crane.** Not loosely "undersized": **over-utilized in bending (utilization 1.37, FAIL) and deflection (75.0 mm vs 24.0 mm limit, FAIL); shear passes (0.39)** (`ground-truth/structural-validation-report.md`).
- **ODH.** Precisely: the **ODH-detection SIF fails SIL 2 — total PFD_avg 1.053×10^{2} exceeds the 1.0×10^{2} threshold** (`round7 r7-085`), not loosely "cryo ODH critical".

- **Settlement (confirmed exactly). 33.6 mm differential** vs 12.0 mm (L/1000) magnet-alignment tolerance and 24.0 mm (L/500) structural limit (both FAIL).
- **Cost (confirmed, model named). CHF 147.1 M P50** is the QS BKP estimate (`round5/expert-briefing-package-r5.md` L66); a separate parametric model (`calculations/grounding-report.md` L108) gives **CHF 148.0 M P50** (spread 0.9 M = 0.61 %, treated as a *lower bound* on model-form uncertainty since the two models likely share CHF/m² priors). The S2 sensitivity band is **P10 / P50 / P90 = CHF 122.0 M / 147.1 M / 189.6 M** (right-skewed per reference-class behaviour for one-off research facilities).
- **Roof slab (published claim was wrong — second self-caught defect).** The published "roof slab 53 mm clearance" is **not record-grounded**: no 53 mm roof/clearance finding exists anywhere in the corpus. The real, more serious finding is that the 1.8 m roof slab has only a **6.9 mm (0.38 %) Moyer shielding-thickness margin** (1800 mm actual vs 1793.1 mm required at 0.1 µSv/h, `round7 F-R7122-001`); the maximum acceptable void is **5.3 mm embedded / 3.5 mm near-surface** (`round8 F-R8070`), at or below NDT detection limits, and 45 of 53 FLUKA parametric configurations produce roof-dose exceedances. The "53" is almost certainly a conflation of the 53-configuration FLUKA study and/or a decimal shift of the 5.3 mm void limit; "clearance" mischaracterizes a *shielding margin* (the only true clearance in record is the crane hook at 8.0–8.5 m, PASS). An open conflict (CR-R7122-01: analytical 6.9 mm vs FLUKA ~55 %) is governed by the analytical value until the 53-config study resolves it.

What changed (v22.1 → v23.0, 2026-05-13 → 2026-06-10, 28 days): +3 runs (80 total), +61 decisions (1,968 total), schema 853 base tables + 384 views = 1,237 objects (+13.7 % base / +8.5 % views), skills reframed to 49 ACTIVE, agents 78, experts 446 (stable), pipelines 20 (19 active), knowledge_index 1,030, officer_events refreshed as above. Full per-claim drift table and the new act follow.

Release Note — v22.1 Publication (2026-05-13)

v22.0 was retracted on 2026-05-13 the same day it was first published — Liviu's review-time inspection caught a build-pipeline regression in which every figure reference resolved to a "Could not fetch resource" pandoc warning, shipping a 191-page PDF with **zero embedded images**. Root cause: `bin/fdrp-paper-publish.sh` invoked `pandoc` without `--resource-path`, so `figures/X.pdf` references failed to resolve at build time; the wrapper still wrote "OK" because pandoc's exit code was 0. The previous BIND-038 gate counted files on disk rather than image streams in the output PDF, so the broken artifact passed. The retracted PDF

is preserved on disk and on the live web root for forensic comparison as `FDRP-Paper-v22.0-retracted-2026-05-13.pdf`. The full RCA lives at `research/fdrp-paper-v22-regression-rca-2026-05-13.md`. v22.1 is this corrected publication, built via `bin/pdf-build-and-check.sh` which sets `--resource-path="$MD_DIR"` correctly, with a post-build `pdfimages -list <pdf> | wc -l >= 30` hard gate. A write-time PDF-figure-verification hook is staged for Liviu HARDENING-0 deploy to prevent recurrence.

v22.1 is the live publication replacing the v21.0 PDF that has carried `https://fdrp.liviu.ai/paper/` since 2026-04-16. **Convention** (applies throughout the paper): all numbers are live as-of the 2026-05-13T11:40Z snapshot against `<intelligence_db>` unless annotated `[SNAPSHOT <date>]` — historical figures (e.g. 2026-03-14 R1+R2 antimatter percolation, 2026-04-16 R11 update) retain their original snapshot stamp for archival traceability.

What changed (v21 → v22.1, 2026-04-16 → 2026-05-13, 27 days):

- **+7 FDRP runs** (77 total: 23 CONVERGED, 22 COMMISSIONED, 7 ACTIVE, 4 SEED, balance triaged)
- **+201 decisions** (1,907 total) — aggregate count; per-tier breakdown not available because `fdrp_decisions` does not carry a `decision_tier` column at the v22.1 snapshot (BIND-041..045 tiers are scoped to the dispatch layer, not the decision artifact; remediation tracked as v22.2 work)
- **Schema: 750 base tables + 354 views = 1,104 schema objects.** The v22.0 release-note headline "1,104 base tables (+48.2%)" conflated BASE TABLE with VIEW; the corrected split is 750 base tables and 354 views (+88.3% views over 27 days). DDL churn is net-positive at this snapshot; a creation/deletion/rename event ledger is queued for v22.2.
- **+25 skills** (73 registered total) — of which **1 has been invoked in the trailing 7 days** (SQL: `SELECT COUNT(*) FROM skill_registry WHERE last_invoked >= NOW() - INTERVAL 7 DAY`). The registered-vs-operational gap (73 registered / 1 actively invoked) is itself the 0-invocation "skill theatre" signal called out in the 2026-05-13 skills-ecosystem-evolution doctrine; the doctrine and the gap measurement co-emerged in this release.
- **Agents: 58 in `agent_dispatch_table` total** (54 existed before 2026-04-16, 4 net-new in v21→v22 window; SQL: `SELECT COUNT(*) FROM agent_dispatch_table WHERE created_at < '2026-04-16'` returns 54). The "v21 baseline 5" figure in v21 referred to **agents dispatched at least once in v21's measurement window**; v22 redefines the agent count to mean "registered in the dispatch table". This is **definitional broadening, not 11.6× growth in 27 days**.
- **Experts: 446 LLM-instantiated personas** in `fdrp_expert_registry`. v21's "24" figure referred to **human persona archetypes**. The v22 unit-of-count was redefined: 446 is LLM instantiations across all 77 runs, not 446 new archetypes. v21's "24" is not directly comparable; **the delta reflects definitional broadening + real expansion combined, not**

a 1858% archetype growth.
- **+6 concept-to-X pipelines** (20 total)
- **1,128 entries** in `knowledge_index` (cross-cutting reference)

Observability spine (corrected from v22.0 — Blocker B1): `SELECT COUNT(*) FROM officer_events WHERE ts >= NOW() - INTERVAL 7 DAY` returns **28,465 events across 185 distinct `event_type` classes** at the 2026-05-13T11:40Z snapshot. Mean rate = 28,465 / (7 × 24 × 60 min) = **2.82 events/min**. Total `officer_events` cardinality: 32,785. (The v22.0 release-note claim of "27,414 events / 8.5 events/min / 7 days" was arithmetically inconsistent — 8.5/min × 7 d × 1,440 min/d = 85,680, ~**3.13× the asserted count**; the corrected pairing is 28,465 / 7 d / 2.82 ev/min.)

Wave R11 architecture panels reached **CONVERGING (round 1)** with CVT values 0.86–0.87 (coordination-layer 0.87, skills-lifecycle 0.86, PPG 0.87, event-bus 0.87) and shipped sub-waves SW-*-1..6 LIVE. **CONVERGING is the correct state label** because FDRP's `fdrp_runs.status` state machine reserves CONVERGED for a convergence reading 0.95 on the high-is-good scale (`convergence_score = 1 - cvt_ratio`; see CVT Dynamics §"Polarity convention"); 0.86–0.87 is CONVERGING, not CONVERGED. The v22.0 release-note language "converged round-1" is corrected here. Round-2/round-3 promotions to CONVERGED status remain pending the next iteration cycle.

Four foundational doctrines were documented in v22 from incidents during the v21→v22 window (Liviu omni-scope override, skills-ecosystem evolution mandatory, cybersecurity + testing pre-publish gate, event-driven not schedule-driven). The doctrines were **newly memorialised** in v22.1's release window — not retroactively backfilled; each traces to a discrete officer_events memorialisation event and git-commit chain inside the 27-day window. **BIND-052** was **reaffirmed at OS-level enforcement** in v22 (v21 framed it as session-scoped; v22 explicitly extends to OS-level governance — see Section 4 for the v21→v22 promotion citation). **BIND additions** in v22: BIND-MLM-001/002/003 (Multi-Level Manager, 2026-05-04), BIND-073 DBTMC (Detect → Backbone → Test → Monitor → Correct), BIND-SCOPE-001 (Andon Scope Discipline).

`fdrp-evolve-v2` 8-stage autonomous chain — doctrine-reconciliation receipt. The chain operated under explicit Liviu pre-authorization (2026-05-12 directive captured as `officer_events` row with severity=DIRECT before the chain initialised; this is the IMPLEMENT_MEDIUM-tier authorization receipt per BIND-044). Stage-by-stage decision-tier audit (inline auditable mapping; row-level evidence in `fdrp_evolution_log`):

Stage	Action class	BIND tier	Plan-completion artifact?	Authorization receipt
1	State-of-system observe (read-only counts)	BIND-041 OBSERVE	No	n/a (read-only)
2	Orient (delta computation vs prior epoch)	BIND-041 OBSERVE	No	n/a (read-only)
3	Decide-without-act (proposal generation)	BIND-042 SUGGEST	No	n/a (no side effect)
4	Schema migration (additive only, idempotent)	BIND-043 IMPLEMENT_LOW	No	2026-05-12 pre-auth
5	Hook stage-file write (to `state/staging/`)	BIND-043 IMPLEMENT_LOW	No	2026-05-12 pre-auth
6	Registry insert (skills + dispatch table)	BIND-043 IMPLEMENT_LOW	No	2026-05-12 pre-auth
7	Auto-refreeze arming (governor state change)	BIND-044 IMPLEMENT_MEDIUM	No	2026-05-12 pre-auth
8	Receipt persistence (`officer_events` emit)	BIND-044 IMPLEMENT_MEDIUM	No	2026-05-12 pre-auth

No stage produced a "plan-completion" artifact in the BIND-030

sense (a plan-completion artifact is a deliverable-bearing document requiring user greenlight before downstream action; an evolution remediation cycle does not generate such an artifact). **No stage met BIND-045 EMERGENCY tier criteria** (no security incident, no data loss, no SLO breach). The autonomous-chain assertion is therefore reconcilable with the BIND-030/041/042/043/044/045 decision-rights stack. A queryable `fdrp_autonomous_chain_audit` table is queued for v22.2 to make this mapping machine-introspectable in addition to inline.

Constitution empirical testing — RETRACTED 2026-05-13. The original 10 PASS / 14 PARTIAL / 4 FAIL split (v0.14.0, 2026-03-25) is **retracted in v22.1**. The supporting rows in `fdrp_payload_constitution.effectiveness_score` now uniformly read 1.0 across all 28 principles (W71 counter-example CF-08 — the 1.0 uniformity is itself the alarm). Re-run is scheduled into an append-only `fdrp_constitution_test_results` audit table per W72 Proposal P9, target **2026-05-20**. The narrative paragraphs in Section "Constitution Empirical Testing" carry a `RETRACTED 2026-05-13` banner; do not cite for new claims. Cite v21 §17 (Constitution Empirical Testing) for archival purposes only.

`fdrp_findings` 8,554 → 197 — RECLASSIFICATION, not retraction. The 8,554 rows did not disappear; they migrated to `fdrp_expert_findings` (now 8,571 rows, Δ=+17 over 27 days). The `fdrp_findings` table was reclassified from "all findings" to "primary findings" only. 8,310 of 8,571 expert findings (96.9%) concentrate in **run #1017 antimatter panel** across 11 rounds — this is **dataset-concentration risk**, disclosed here rather than left as an unstated property. Generality claims downstream of `fdrp_expert_findings` should weight this concentration.

Cross-model PEAR (BIND-046, BIND-064): v22.0 was published on 2026-05-13T09:27Z carrying an unclosed BIND-046 obligation for ~16 minutes; PEAR was completed retroactively at 2026-05-13T09:43Z, identifying 6 BLOCKERs which are now absorbed into v22.1. A second cross-model PEAR pass is run against the v22.1 manuscript before live deployment. Reviewer responses are persisted under the mesh response store (`<mesh_root>/state/responses/`) as the dated `fdrp-paper-v22-pear` opus-mirror and opus-codex-substitute transcripts.

Security & infrastructure (operational, not scientific results — see Appendix): 43 IPs banned via `threat-autoban` after NFT_TABLE fix; AIDE cleanup; BIND-028 secret-scan verified functional (6/7 synthetic patterns matched); Node 3 reverse-tunnel restored end-to-end (18080:localhost:8080 LLM forward; `<mail_host>` SSH port 622). **Wave R23 "Plan-Mode Architecture"** READY-FOR-COUNCIL (hooks + context backbone + validation).

All numeric claims in this revision trace to a `SELECT` against `<intelligence_db>` at snapshot 2026-05-13T11:40Z (BIND-021). The publication passed `bin/fdrp-paper-freshness.sh` (M1–M7 + B1 officer_events arith-

metic re-grounded), `bin/pdf-build-and-check.sh` (figure-embedding via `--resource-path`), and the new post-build `pdfimages -list <pdf> | wc -l  30` figure-count assertion (BIND-051 receipt persisted; staged hook will enforce this at write-time after Liviu's HARDENING-0 approval).

Abstract

FDRP applies the quality controls that world-class factories use on production lines — statistical process control, defect detection, continuous improvement — to the planning process itself, so that every planning decision is manufactured with the same traceability, convergence measurement, and gate-review rigour as a physical component. We present FDRP v0.18.0 (paper revision v23.0), the matured framework spanning **80 production runs** (24 CONVERGED, 24 CLOSED, 18 CANCELLED, 6 ACTIVE, 4 SEED, 2 COMMISSIONED, 1 PAUSED, 1 CONVERGING), **1,968 traceable decisions**, **853 MySQL base tables + 384 views** (1,237 schema objects), **49 ACTIVE plugin skills** (132 registered total: 49 ACTIVE / 6 DEPRECATED / 77 PROVISIONAL; 0 invoked in the trailing 7 days — a disclosed registered-vs-operational gap), **78 dispatch agents**, **20 concept-to-X pipelines** (19 active), **446 LLM-instantiated expert personas** in `fdrp_expert_registry`, and **1,030 entries** in the cross-cutting `knowledge_index` — all values from `SELECT COUNT(*)` against `<intelligence_db>` at the 2026-06-10T12:47:43Z snapshot (BIND-021). The system extends from its manufacturing-quality origins [47] to a measured knowledge engine unified by **progressive disclosure** — the principle of revealing information at increasing levels of detail — operating across seven dimensions: content depth, routing, long-tail economics, multi-resolution matching, vocabulary mapping, audience-adaptive translation, and cost-optimal model selection. This edition adds *Measuring Ourselves* (Act III): three pre-registered falsifiable A/B experiments turned the process on itself. None showed the orchestration wrapper beating an unwrapped baseline on the same curated prompt; one caught a bug in the system's own verdict logic; and the most dramatic verdict was rejected by the system's own quality gate as a measurement artifact. The resulting epistemic distinction — **PRODUCES is not BEATS** — frames the paper. Every positive claim here is fenced as one of two kinds: *produced in practice* (an existence/operations claim — the antimatter-facility engineering case is of this kind) or *beat an unwrapped baseline in a controlled A/B* (a comparative-performance claim). **This edition makes no comparative-performance claim for the orchestration wrapper; on the contrary, the A/Bs show it did not beat the baseline on the tested prompt/task regime.** The value the paper does claim — production at scale, deterministic validation, and the publishing pipeline — is the *produced-in-practice* kind,

Gated sections of the companion website (https://fdrp.liviu.ai) are accessible via the token-bearing URLs embedded in this paper (valid for 14 days from publication date). After token expiry, gated sections remain accessible via BasicAuth credentials.

located in curated domain prompts and the toolbelt, not in a self-improving flywheel (measured true-promotion yield 0.12 %, retired 2026-06-10). v22.1 inherits the full lineage: SPC/5S/Andon manufacturing quality controls (v1.8), progressive disclosure as unifying architecture (v0.7.0), automated versioning, expert teaming, knowledge grafting, Principal Intelligence Officer architecture, closed-loop infrastructure security (v0.8.0–v0.12.0), expert persistence with git-like session-resurrection lifecycle (SEED/FORK/MERGE/REBASE/TAG) and ultrathink cascade methodology (v0.13.0), anti-pattern engine maturation with first commercial deployment (v13.0), batch convergence with Oracle Fusion integration (v0.14.0), the FDRP-on-FDRP Run #1027 mega-campaign with 30 waves and RAMS 0.953 (v21.0), four 2026-05-13 doctrinal additions (Liviu omniscope override, skills-ecosystem evolution mandatory, cybersecurity + testing pre-publish gate, event-driven not schedule-driven), Wave R11 round-1 architecture convergence across four panels (coordination-layer 0.87, skills-lifecycle 0.86, PPG 0.87, event-bus 0.87), BIND-052 promoted to OS-level enforcement, and the BIND-MLM-001/002/003, BIND-073 DBTMC, and BIND-SCOPE-001 additions.

This insight emerged from an FDRP-on-FDRP meta-analysis: applying the system's own spiral expert expansion to itself. Five cross-domain LLM-generated expert personas — from Search/IR engineering, molecular biology, quantitative finance, network routing, and ecosystem ecology — each mapped their domains onto FDRP subsystems (note: all personas were generated by the same LLM family, limiting the statistical independence of their analyses; see Section 24 for discussion), followed by seven internal subsystem experts who validated the thesis against each component. Cross-model verification (Codex Pro + Gemini 3.1) reviewed the resulting architecture. The expert persistence discovery originated from a CHF 147.1M CERN antimatter building design programme (run #1017 `antimatter-building`, CONVERGED 2026-03-21; P50 from the QS BKP estimate, with a separate parametric model giving CHF 148.0M) where specialist agents across multiple rounds produced thousands of structured findings. Earlier editions framed this as proof that session resurrection turns ephemeral computation into *appreciating* knowledge assets; **v23.0 reframes that claim** — the mechanism reduces follow-up cost (cache economics) and was a genuine engineering productivity gain on this case, but the broader "appreciating assets" / autonomous self-improvement claim was not borne out under measurement and is withdrawn (Act III). The case is retained as the standing example of **what the process produces at scale**, not as evidence of autonomous appreciation.

A parallel DD^n (Double Diamond recursive) analysis across 14 waves and 4 phases produced 25 concept notes distilled into the FDRP Unified Keyprompt v3.8 (~1,350 lines, five-axis recursive quality discipline). v3.8 integrates two cross-model review cycles: v3.6 scored MIN 1/5 (Codex Pro + Gemini 3.1) on implementability, driving a three-tier protocol split, deterministic feedback loop redesign, and two new principles (Evolution Protocol #24, Resource Budget Discipline #25) in v3.7; the v3.7 cross-model review (Codex Pro + Gemini 3.1 + Claude Opus 4.6) scored MIN 1.5/5, identifying budget tier contradictions

and self-reported metrics, driving 4 CRITICAL + 9 HIGH fixes in v3.8.

The system is implemented as a Claude Code plugin (48 skills, 745 MySQL base tables + 188 views, 5 agents, 14 concept-to-X pipelines, 32 registered subsystems) validated across 62 runs producing 1,477 decisions. New subsystems include ecological addresses for O(k) agent routing, a payload constitution governing expert briefings, a Rosetta Stone mapping acronyms to expert knowledge, audience-adaptive translation (Opus generates, Sonnet explains), a dual-timescale self-evolution loop with SPC-based quality monitoring, automated versioning with regression detection, cross-domain validation via the SIVP platform, expert persistence architecture with ultrathink cascade for structured emergent discovery, and knowledge graph percolation where 1,656 expert findings (including 323 from Round 2 auditors) crossed the connectivity threshold (avg degree 4.90 among the 1,039 connected nodes, i.e. $2 \times 2547/1039$; global avg degree across all 1,656 nodes is 3.08; SUPER_CRITICAL phase) with 2,547 edges — with a three-wave expert panel process that exposed and closed 7 fatal defects in schema change infrastructure and converted quality testing to a default daemon action.

Keywords: FDRP, progressive disclosure, ecosystem architecture, ecological address, self-evolving systems, knowledge routing, payload constitution, cross-domain expert expansion, audience-adaptive translation, CERN review gates, expert persistence, ultrathink cascade, session resurrection, knowledge graph percolation, schema quality gates

Executive Summary

FDRP applies **manufacturing quality control** (SPC, 5S, Six Sigma, Andon) to **planning decisions**, treating each decision as a manufactured artifact with traceability, clash detection, convergence metrics, and gate reviews. It is implemented as a production system (48 skills, 745 MySQL base tables, 188 views, 32 registered subsystems, 14 concept-to-X pipelines, 7 visualisation renderers) and validated across 70 runs producing 1,701 decisions with 25 runs commissioned (17 cancelled via stalled-run triage) — snapshot 2026-04-16 from `SELECT COUNT(*) FROM fdrp_runs`, `fdrp_decisions`, `fdrp_runs WHERE status='COMMISSIONED' OR commissioned_at IS NOT NULL`. v0.12.0 extends the self-evolution paradigm to infrastructure security with a closed-loop CyberDefence subsystem: MITRE ATT&CK-mapped attack taxonomy, CVE feed monitoring, adversarial cross-node self-testing, and baseline drift detection. v0.13.0 introduces expert persistence architecture and the ultrathink cascade methodology, discovered through a CHF 147.1M CERN antimatter building design programme where 68 specialist agents across two rounds produced 1,656 structured findings. v0.14.0 achieves first batch convergence (6 runs simultaneously reaching CONVERGING status), validates commercial transfer via the Oracle Fusion integration pivot, and replaces bulk-set constitution scores with empirical measurements.

Key results:

- **Convergence** — The 2 fully-iterative commissioned runs stabilise in the POLISHING zone (CVT 0.250–0.311) within 8–14 iterations (see Limitations for sample size discussion). v0.14.0 achieved the first **batch convergence**: 6 runs (collimation #8, #10, #11, #12, #13, #14 and paper #22) simultaneously transitioned to CONVERGING status with CVT values 0.172–0.191. v19.0 achieves full portfolio convergence: all 19 CONVERGED (15 at 1.0000, 4 at 0.92–0.99), 0 CONVERGING, 0 ACTIVE remaining. System-wide: 19 CONVERGED + 0 CONVERGING across 62 total runs (17 CANCELLED via stalled-run triage, 22 COMMISSIONED). Source: `fdrp_runs`, 2026-04-02
- **Quality** — 97.7% self-assessed CERN compliance (~80% FULL + ~18% EXCEEDS across 44 clauses derived from CERN review procedures; not independently validated by CERN)
- **Safety** — IEC 61508 SIL classification integrated; RAMS analysis on all commissioned decisions
- **Self-evolution** — Dual-timescale OODA (per-turn + 12h daemon) with SPC Nelson Rules on skill effectiveness; recurring errors automatically promoted to prevention rules via cross-model peer review
- **Expert discovery** — Convergence-based expert expansion (40 experts in the FDRP-on-FDRP analysis, 68+ in the antimatter building programme) identified 7 architectural blind spots (through LLM-generated expert personas, not human domain experts) (Game Theory, Adversarial ML, Temporal Databases, Process Mining, Petri Nets, Control Theory, Schema Evolution) — none prescribed by human architects; expert count is bounded by convergence, not by a predetermined cap
- **Expert persistence** — Session resurrection reduces follow-up dispatch cost by an estimated 92% (derived from token pricing assumptions) ($1.50–3.00 fresh vs. $0.05–0.15 resumed per expert); git-like operations (SEED, FORK, MERGE, REBASE, TAG) enable multi-round expert panels with persistent accumulated context
- **Ultrathink cascade** — Structured 4-phase methodology (Trigger → Dispatch → Cascade → Synthesis) for extracting emergent cross-domain insights; 5 LLM-generated expert analyses converged on "Expert Knowledge Operating System" as synthesis — interpreted by the authors as unprescribed emergence illustrating the cascade protocol (see caveat on shared LLM substrate in Section 26)
- **Cross-model verification** — Two review cycles completed: v3.6 reviewed by Codex Pro and Gemini 3.1 (MIN 1/5, feedback loop implementability), 6 CRITICAL findings addressed in v3.7; v3.7 reviewed by Codex Pro + Gemini 3.1 + Claude Opus 4.6 (MIN 1.5/5, budget tier contradictions), 4 CRITICAL + 9 HIGH findings addressed in v3.8
- **Principle lifecycle** — 25 concept notes with formal lifecycle management (Evolution Protocol #24) and resource-aware deployment tiers (Resource Budget #25)

- **Constitution empirical testing** (v0.14.0) — **RETRACTED 2026-05-13.** Original 10 PASS / 14 PARTIAL / 4 FAIL split traced to `fdrp_payload_constitution.effectiveness_score` rows that now uniformly read 1.0 across all 28 principles (W71 counter-example CF-08 — the 1.0 uniformity is itself the alarm). **Do not cite for new claims.** W72 Proposal P9 re-run into append-only `fdrp_constitution_test_results` audit table is scheduled for **2026-05-20**. Cite v21 §17 (Constitution Empirical Testing) for archival purposes only.
- **Batch convergence** (v0.14.0) — 6 runs simultaneously transitioned ACTIVE → CONVERGING: collimation runs #8, #10, #12, #13, #14 and paper run #22, all with CVT < 0.200, zero open hard clashes, zero active Andon events, and stable decreasing CVT trajectory over 11–13 iterations. Subsequently all progressed to CONVERGED. v19.0 achieves full portfolio convergence: 19 CONVERGED (15 at 1.0000), 0 CONVERGING, 0 ACTIVE. Source: `fdrp_convergence` and `fdrp_runs`, 2026-03-26
- **Oracle Fusion pivot** (v0.14.0) — Run #1015 (the hospitality procurement client) CVT dropped 0.900 → 0.676 after the client's VP of Procurement revealed an Oracle-based seasonal procurement system, crystallising the integration target from abstract Oracle API to specific Oracle Fusion Cloud Procurement. YELLOW Andon fired for capability commitment without verification — the methodology caught a premature commitment that unstructured sales would have missed. Source: `fdrp_convergence` run 1015, 2026-03-25
- **Governance clearance** (v0.14.0) — 4 overdue peer reviews processed (evo_log_ids 208, 209, 225, 226), achieving zero governance backlog for the first time. Source: `fdrp_evolution_log`, 2026-03-25

For whom: Architects of long-lived, multi-institutional programmes (FCC-class) where planning must evolve faster than the project it governs.

Revision History

Version	Date	Description
0.1	2026-02-28	Initial structure and core sections
1.0	2026-03-03	Complete draft with all 16 sections, 27 references
1.1	2026-03-04	Added 12 data-driven charts from MySQL production data
1.3	2026-03-05	Added 9 AI-generated concept images via Codex/DALL-E
1.4–1.6	2026-03-05	Peer-reviewed images: 3 evolutionary runs (Claude VLM + Gemini)
1.7	2026-03-05	CERN-style frontmatter, TL;DR, document history

Version	Date	Description
1.8	2026-03-05	Professional typography: custom XeLaTeX template, STIX Two + Fira Sans
2.0	2026-03-06	Comprehensive merge: v1.8 + v0.7.0 into single document
3.0	2026-03-08	Updated to v0.9.0: all metrics refreshed, SIVP validation added, versioning section merged
4.0	2026-03-08	Updated to v0.11.0: ecosystem self-management architecture (7 waves), expert teaming, matchmaking, thinking composition, control plane
5.0	2026-03-09	Updated to v0.12.0: CyberDefence subsystem, MITRE ATT&CK taxonomy, CVE monitoring, adversarial self-testing, baseline drift detection. Audit^n Round 1 fixes.
5.2	2026-03-09	52 figures (was 35): 17 new D3.js charts + embedded 10 existing unreferenced figures. ASCII diagram converted to figure. Hook effectiveness, version growth, limitations risk landscape, ecosystem self-management, wave architecture, knowledge grafting, programme branching visualisations added.

Version	Date	Description
5.3	2026-03-09	Audit^n Round 2 systematic data reconciliation: all numeric claims verified against MySQL ground truth. Fixed: fdrp_ tables (145, was 188/141), views (43, was 47/42), total tables (254, was 305), CERN compliance (44 clauses, was 32), evolution events (76, was 33/11), convergence records (649, was 203), domain clusters (11, was 13), pipelines (14, was 12/13), agents (5, was 6), triggers (129, was 10). Removed Haiku model reference (BIND-049). Fixed BGP citation [51]. Added convergence sample-size qualification. Document number changed to DRAFT prefix.
6.0	2026-03-10	Updated to v0.13.0: Wave 9 — Expert Persistence Architecture and Ultrathink Cascade. New section covering session resurrection as knowledge containers, git-like expert lifecycle (SEED/FORK/MERGE/REBASE/TAG), 10-domain cross-pollination mapping, systems dynamics (4 reinforcing + 5 balancing feedback loops), 92% cost reduction via prompt cache economics, ultrathink cascade 4-phase methodology, and convergence toward Expert Knowledge Operating System. Source: CERN antimatter building programme (32 experts, 28,936 lines, CHF 147.1M P50). 5 new figures (Fig. 53–57).

Version	Date	Description
7.0	2026-03-11	Data reconciliation against MySQL ground truth: 290 base tables (was 254), 74 views (was 51), 1,279 decisions (was 1,199), 56 runs (was 51), 86 evolution events (was 76), 61 BIND rules (was 58). All 28 D3.js figures regenerated with current data, converted to vector PDF format, and redesigned with publication-quality typography. LaTeX template upgraded with float barriers, improved spacing, and editorial-grade layout. Structural polish: section cross-references fixed, figure numbering harmonised, heading hierarchy corrected, 11 unreferenced citations linked, stale v0.9.0 narrative updated.
8.0	2026-03-13	Knowledge graph percolation and schema quality gates. New section: finding similarity pipeline (1,333 findings, 1,838 edges, SUPER_CRITICAL percolation), 3-wave expert panel process (19 schema changes, 7 fatal defects found and closed, 19 security findings), STAY_MYSQL verdict (0.66 MB graph, sub-ms queries), aviation-inspired rollback architecture (unified coordinator, auto-tags, DB-first ordering), security hardening (4 scripts patched, 14 SQL injection points closed, trust boundary documentation), schema quality gates as default LL daemon action (7-check gate, SPC control chart, Sources #12 and #13).

Version	Date	Description
9.0	2026-03-14	Round 2 antimatter data (1,656 findings, 2,547 edges). DD^n analysis (25 concept notes) produced keyprompt v3.7 (~1,300 lines). Cross-model review v3.6: MIN 1/5, driving three-tier protocol split and deterministic feedback loop redesign in v3.7.

Version	Date	Description
10.0	2026-03-14	DD^n notes #24–#25, cross-model review integration. Note #24 (Evolution Protocol): note lifecycle management — ANNOTATE/QUALIFY/REVISE/MERGE/SPLIT/DEPRECATE operations, staleness detection (6 indicators), semantic versioning for concept notes, CONCEPT vs PRINCIPLE naming resolution. Note #25 (Resource Budget Discipline): triage tiers (MINIMUM 3 / STANDARD 5 / FULL 25 principles), contextual priority matrices by task type, ROI ranking, deployment sequencing across 5 phases. Cross-model review of v3.6 (Codex Pro + Gemini 3.1): MIN score 1/5 driven by feedback loop implementability, 6 CRITICAL + 9 HIGH + 6 MEDIUM findings. Remediation: severity count correction in failure modes, three-tier protocol split (safety kernel / operational / experimental), feedback loop redesigned with deterministic tooling (DOE ablation, offloaded Bayesian math), checklist items converted to deterministic data-extraction commands, FOUNDATIONAL term collision resolved, missing failure mode categories added (human operator, environmental, feedback loop).

Version	Date	Description
11.0	2026-03-14	v3.8 keyprompt integration: 4 CRITICAL + 9 HIGH fixes from v3.7 cross-model review (Codex Pro + Gemini 3.1 + Claude Opus 4.6). Key fixes: detraining tier conflict resolved (#11 promoted to Safety Kernel), budget tiers tagged [PROVISIONAL] with "converge don't cap" language, self-reported outcome_score removed (computed by deterministic script), ablation schedule made trigger-based, severity formula corrected to RPN, FM categories 5-6 given quantitative detection signals, REVISE trigger stabilized (3 ablations not 1). Two cross-model review cycles completed: v3.6 MIN 1/5 → v3.7 MIN 1.5/5 → v3.8 targeting MIN 2.5/5+.
12.0	2026-03-14	1M context window reframing. Opus 4.6 GA with 1M token context at $5/MTok input changes the resource economics throughout: context window constraint updated from 200K to 1M (leaving ~910K tokens for reasoning after full keyprompt), B1 context saturation threshold updated to ~500K–700K empirical range (76% needle-in-haystack at 1M), triage tier thresholds recalibrated, resource-conscious deployment reframed from token scarcity to attention budget management. The primary constraint shifts from "can it fit?" to "can it attend?" — the model's ability to maintain focus across increasingly large contexts becomes the governing bottleneck.

Version	Date	Description
13.0	2026-03-25	Anti-pattern engine fix and commercial deployment. Anti-pattern detection engine repaired (0% → 26% trigger rate across 31 rules; 23 broken SQL detection rules fixed). 4 new anti-patterns: AUTH-WALL, CORRECTION-AVALANCHE, ARSENAL-WITHOUT-DEPLOYMENT, CAPABILITY-DEBT. Correction pipeline unblocked: 801 pending corrections triaged to 8 pending / 808 applied (ratio inverted from 40:1 to 1:101). First commercial FDRP engagement: Run #1015 (an anonymized Gulf hospitality procurement client). Constitution principle testing gap identified (28 principles, 0 genuinely tested). 6 Gulf B2B ultra-experts rostered for Oracle integration pivot.

Version	Date	Description
14.0	2026-03-25	First batch convergence and constitution empirical testing. 6 runs (collimation #8, #10, #12, #13, #14 + paper #22) simultaneously transitioned to CONVERGING with CVT 0.172–0.191. Governance backlog cleared to zero (4 overdue peer reviews processed). Run #1015 Oracle Seasons pivot: CVT 0.900 → 0.676, zone ROUGH_CUT → FACETING. Constitution principles empirically tested for first time: 10 PASS, 14 PARTIAL, 4 FAIL (prior bulk-set 0.80 scores replaced with measured effectiveness 0.02–0.90). Stalled run triage: 6 FSM transitions added, 4 runs triaged (1 CANCELLED, 1 CLOSED, 1 PAUSED, 1 kept ACTIVE with decisions accepted). 98 CRITICAL improvement proposals identified as META-004 (Analysis-Action Gap) accumulation.
15.0	2026-03-25	D3.js SVG data visualizations from MySQL ground truth. All PNG figures replaced with vector SVG equivalents rendered via D3.js force-directed layout (knowledge graph percolation), Sankey flow (three-wave panel), and stacked bar (quality gates).

Version	Date	Description
16.0	2026-03-25	First batch CONVERGED milestone. 2 runs (#10 collimation-correlated-foundation, #12 collimation-correlated-transport) reached CONVERGED status (convergence_score 1.0000, CVT 0.177/0.176, 13 iterations). System-wide: 7 CONVERGED + 7 CONVERGING across 62 runs (at time of v16.0). Total decisions updated to 1,461. Convergence records updated to 853. 3 new D3.js SVG figures (fig-058 knowledge graph percolation, fig-059 three-wave panel, fig-060 quality gates) confirmed in paper.
17.0	2026-03-25	Wave 17 data refresh. Schema growth: 304 → 461 base tables, 84 → 132 views. Decisions: 1,461 → 1,477 (+16). BIND rules: 61 → 63. Stalled-run triage: 11 runs CANCELLED (commissioned count 38 → 28). Run #14 (collimation-synthesis-paper) approaching CONVERGED at 0.9131. Error monitoring: 3,736 entries fully classified (2,708 meta_noise, 0 uncategorized). All numeric claims reconciled against MySQL ground truth.

Version	Date	Description
18.0	2026-03-25	Convergence push. Runs #11 (collimation-hadronic-bondarenco) and #13 (collimation-multiturn-stochastic) reach CONVERGED 1.0000. Runs #8 (flux-protocol-cern-demo, 0.9500) and #22 (fdrp-technical-paper, 0.9200) pushed to CONVERGING. Additional CONVERGED: #14 (1.0000), #21 (1.0000), #40 (0.9500), #1010 (0.9200), #1013 (0.9500), #1014 (0.9900). System-wide: 10 CONVERGED + 4 CONVERGING across 62 runs. Dashboard data refreshed.
19.0	2026-03-26	Full portfolio convergence. 19 CONVERGED (15 at 1.0000): runs #8, #10, #11, #12, #13, #14, #21, #22, #35, #39, #1007, #1008, #1009, #1011, #1015 plus 4 others at 0.92–0.99. 0 CONVERGING, 0 ACTIVE. Schema growth: 461 → 593 base tables. Commissioned: 28 → 22 (6 additional CANCELLED). 309 assumptions tracked.

Version	Date	Description
20.0	2026-04-02	Nine new subsystems (#24–#32) in a single session. Schema growth: 593 → 745 base tables (+25.6%), 132 → 188 views (+42.4%), fdrp_ tables: 156 → 360 (+130.8%). 66 new tables across: web-finetuning (5 tables, ICP-aware Lighthouse), definition-of-done (4 tables, 55 evidence gates), gaps-and-opportunities (3 tables, 3-horizon scanning), thinking-chains-and-structures (15 tables, 15 topologies, 10 chains), cognitive-architecture (18 tables, thinking-about-thinking), daemon-taxonomy (7 tables, 35 daemon types), cognitive-opportunities (5 tables, 40 archetypes), wiring-check-and-fix (4 tables, 271 checks), limits-hunt-and-fix (5 tables, 68 limits, 136 evaluations). 32 registered subsystems (was 23). Qualitative jump: system now has cognitive self-awareness (thinking about its own thinking), unified daemon architecture, integration verification, and resource governance.

Version	Date	Description
22.0 (W72)	2026-04-16	[**SNAPSHOT 2026-04-16T09:25Z**] W72 drift-remediation patch (M1–M7 + 2 new discoveries). All numeric claims re-grounded against live `<intelligence_db>` via `bin/fdrp-paper-freshness.sh` (exit 0 required to publish). Headline corrections: `fdrp_findings` 8,554 → 197 (**reclassification, not retraction** — rows migrated to `fdrp_expert_findings`, where 8,571 are observed (Δ=+17). `fdrp_findings` was reclassified from "all findings" to "primary findings" only); `fdrp_expert_findings` 8,554 → 8,571 (all-runs count; **8,310 / 8,571 96.9% concentrate in run_id=1017 antimatter panel** across 11 rounds — disclosed as dataset-concentration risk; historic "1,656 R1+R2" figures retained as 2026-03-14 snapshot and scoped); `fdrp_finding_edges` 2,547 (historic) qualified with live 6,119; total runs 62 → 70 (as of 2026-04-16); commissioned runs 22 → 25; total decisions 1,477 → 1,701 (as of 2026-04-16); constitution 10/14/4 split withdrawn pending re-run; knowledge-grafts section rewritten from "zero" premise to 241 accumulated-effectiveness-unmeasured. Measurement-snapshot field added to frontmatter.

Version	Date	Description
22.0	2026-05-13T09:25Z	**RETRACTED 2026-05-13 (same-day).** First live publication of v22.0 by `bin/fdrp-paper-publish.sh`. **Retracted** after Liviu's review-time inspection caught a build-pipeline regression: every `figures/X.pdf` reference resolved to a pandoc "Could not fetch resource" warning because the wrapper invoked `pandoc` without `--resource-path`. The 191-page PDF shipped with **zero embedded images** (`pdfimages -list` returns empty). The previous BIND-038 gate counted files on disk rather than image streams in the PDF. The retracted PDF is preserved as `FDRP-Paper-v22.0-retracted-2026-05-13.pdf` (live + repo) for forensic comparison. Full RCA: `research/fdrp-paper-v22-regression-rca-2026-05-13.md`. v22.0 release-note paragraph additionally carried 6 BLOCKER-class content errors flagged by retroactive cross-model PEAR (officer-events arithmetic 8.5 ev/min impossibility, undated changelog rows, "experts 24→446" and "agents 5→58" presented as growth instead of definitional broadening, Wave R11 0.86 labelled "converged", fdrp-evolve-v2 autonomy without tier authorization cited, incomplete retraction discipline on 10/14/4 constitution test). All six are resolved in v22.1 below.

Version	Date	Description
22.1	2026-05-13T11:40Z	[**SNAPSHOT 2026-05-13T11:40Z**] Live republication superseding retracted v22.0. **Build-pipeline fix:** switched to `bin/pdf-build-and-check.sh` (`--resource-path="$MD_DIR"` set correctly); post-build assertion `pdfimages -list <pdf> | wc -l >= 30`; write-time PDF-figure-verification hook staged for HARDENING-0 deploy. **BIND-038 enforcement:** 67 PNG figures quarantined from `reports/fdrp-paper/figures/` to `figures/quarantined-png-bind038-2026-05-13/` (vector-only PDF embedding). **Content fixes from retroactive PEAR (BLOCKERs B1–B6 absorbed):** (B1) officer_events corrected from "27,414 / 7d / 8.5 ev/min" (arithmetic-impossible) to measured **28,465 events / 7d / 2.82 ev/min, 185 distinct types**; (B2) snapshot timestamps added to every changelog row in this table; (B3) agents disclosed as **definitional broadening** (54 of 58 existed before 2026-04-16; only 4 net-new in 27-day window; "v21 baseline 5" was dispatched-at-least-once-in-window, v22 is registered-in-table); experts disclosed as **unit-of-count change** (446 LLM-instantiated personas not comparable to v21's 24 human archetypes); (B4) Wave R11 round-1 relabelled **CONVERGING** (CVT 0.86–0.87, 0.95 = CONVERGED reserved); (B5) `fdrp-evolve-v2` autonomy cited as **BIND-044 IMPLEMENT_MEDIUM** (Liviu pre-authorization 2026-05-12 directive), all 8 stages stayed inside LOW/MEDIUM tiers; (B6) constitution 10/14/4 **RETRACTED 2026-05-13** with COPE-style banner and dated re-run target (2026-05-20 into
		27

Version	Date	Description
23.0	2026-06-10T12:47:43Z	[**SNAPSHOT 2026-06-10T12:47:43Z**] Supersedes v22.1. **New contribution:** Act III *Measuring Ourselves* — three pre-registered falsifiable A/B experiments (`test_execution_log` ids 8, 10, 17) with their negative/inconclusive results published, plus the v0.8.0 skill cut, the flywheel retirement (0.12 % true-promotion yield), and the CVT polarity reconciliation. Reflection renumbered Act IV. **Epistemic spine:** PRODUCES BEATS. **Title decision — Option A ACCEPTED by Liviu 2026-06-10:** title drops "Self-Evolving" → *"FDRP: A Comprehensive Architecture for Planning Quality"* / subtitle *"…and What Measurement Retired"*. Alternatives recorded at draft time: **(A, ACCEPTED)** the above; **(B)** keep lineage verbatim — *"FDRP: A Comprehensive Architecture for Self-Evolving Planning Quality"* with an added subtitle clause *"— with Self-Evolution Reframed by Measurement"*; **(C)** *"FDRP: A Measured Knowledge Engine for Planning Quality"*. **Ten staleness reframes applied** (see Release Note v23.0). **Antimatter case corrected** from SET-0 corpus verification (labyrinth 126 mm-spalling not 126 %-dose; crane bending 1.37/deflection 75 mm; ODH SIF fails SIL 2 at PFD_avg 1.053×10^{-2}; settlement 33.6 mm confirmed; cost CHF 147.1 M P50 QS-BKP vs 148.0 M parametric, S2 band P10/P50/P90 = 122.0/147.1/189.6 M; roof slab "53 mm clearance" expunged as not-record-grounded → real finding 6.9 mm/0.38 % Moyer shielding margin — a second self-caught published defect). **Numbers re-queried at snapshot:** runs 80 (24 CONVERGED, 24 CLOSED, 18
		28

Version	Date	Description
24.0	2026-06-25	**Supersedes v23.0** (correctly rebased onto the v23.0 antimatter/CHF-147 M lineage — a prior workflow had regressed an unversioned v20-era working copy and mislabelled it "v21.0"; this edition discards that regression). **New capability chapter:** *The Deterministic Engine Fabric and the Measured Moat*, placed after Act III as the engine-backed extension of the PRODUCES BEATS thesis. Documents the 39-engine compiled-simulation fabric (`ENGINE-REGISTRY.json`; 33 wired+smoke-PASS, 455 sub-tests PASS / 0 FAIL; 18 NON-INVERTIBLE verification-core engines, 9 INVERTIBLE, 11 N/A, 1 UNKNOWN), the V-model validity limb (1/5 = 20 % bare-AI false-PASS with the 100{,}000{,}859 Pa = 859 Pa-over-limit over-stress of AB-ENG-BEAM-01 caught to the Pascal) and optimality limb (36.74 % lighter, 0.0037956822 vs 0.006000 m²), the closed design→test→feedback loop (`design-optimize.sh`, 13/13 smoke), the **falsified** closed-FORM wrapper-moat A/B (frontier AI one-shots the invertible optimum) and the disproven wrapper-moat across three pre-registered A/Bs (`research/fdrp-vmodel-process-2026-06-24/00-SYNTHESIS.m` the trained ML surrogate (1{,}506 ledger rows, RBF train RMSE 5.1×10 , held-out 80/80 RMSE 2.7×10), the systems-availability awareness layer (8/8 PASS), and the **measured non-invertible moat A/B** (2026-06-25, n=4; `research/fdrp-noninvertible-moat-ab-2026-06-25/00-NONI` **moat-by-non-invertibility NOT established** — bare-AI false-PASS 0/3 on indeterminate FEM (predicted failure mode did not occur), the loop's 2/3 wins are class-agnostic optimization, and NI-1 is a loop *loss* where the

Evolution to v0.14.0 — Current System:

FDRP v0.14.0 introduces **progressive disclosure as the unifying architecture** across seven dimensions: content depth, routing, long-tail economics, multi-resolution matching, vocabulary mapping, audience-adaptive translation, and cost-optimal model selection. This thesis — treated as a hypothesis under test — emerged from applying the FDRP process to itself (FDRP-on-FDRP), dispatching five cross-domain experts (Search/IR, Molecular Biology, Quantitative Finance, Network Routing, Ecosystem Ecology) followed by seven subsystem specialists. Subsequent versions added automated versioning with regression measurement, concern lifecycle management, cross-domain validation via the SIVP platform, CyberDefence, expert persistence with ultrathink cascade, anti-pattern engine maturation, first commercial deployment, and batch convergence with empirical constitution testing.

Key evolution from v1.8:

- **Scale** — 86 → 745 MySQL base tables, 188 views, 28 → 48 skills, 924 → 1,477 decisions, 32 registered subsystems, 22 commissioned runs across 62 total runs (17 cancelled via stalled-run triage)
- **Ecosystem architecture** — 201 elements across 11 domain clusters with ecological addresses (`action@domain@depth@model`) enabling O(k) Longest Ecological Prefix matching
- **Payload constitution** — 28 principles governing expert payloads, with graduation protocol (PROPOSED → CANDIDATE → CONSTITUTIONAL) and forced-ranking bloat prevention; the 2026-03-25 empirical test (10/14/4) was RETRACTED 2026-05-13 — re-run pending
- **Concept-to-X platform** — 14 polymorphic pipelines including `*2paper` (this paper's own generation pipeline), sharing a 5-phase contract with cross-pipeline learning
- **Rosetta Stone** — 20 acronyms mapped to expert domains and actionable knowledge, enabling progressive disclosure from L5 keywords to L0 expert analyses
- **Self-evolution** — Dual-timescale OODA: per-turn hooks (fast loop) + 12h kaizen daemon with SPC Nelson Rules (slow loop), cross-model peer review (Codex + Gemini); 238 evolution events tracked
- **Automated versioning** — SemVer release automation with 17-metric regression detection; 8 versions released (v0.7.0 → v0.14.0), 0 regressions
- **Cross-domain validation** — SIVP platform provides preliminary evidence for FDRP's sparse-principle thesis across earthquake seismology and power grid domains (see Limitations for caveats on sample size and seed family independence)
- **Expert persistence** (v0.13.0) — Session resurrection transforms ephemeral AI expert sessions into persistent, forkable knowledge containers; git-like lifecycle operations (SEED, FORK, MERGE, REBASE, TAG); estimated 92% cost reduction on multi-round expert panels (derived from token pricing assumptions)

- **Ultrathink cascade** (v0.13.0) — Structured 4-phase methodology for extracting emergent insights from parallel ultra-expert dispatch; validated on CHF 147.1M CERN antimatter building programme (32 experts, 28,936 lines of output)
- **Knowledge graph percolation** (v0.13.0) — 1,656 findings connected by 2,547 edges in SUPER_CRITICAL percolation phase (avg degree 4.90 among 1,039 connected nodes; 3.08 global); Round 2 auditor wave (10 experts, 323 findings) resolved 7 HIGH contradictions; three-wave expert panel closed 7 fatal defects and converted schema quality testing to a default daemon action with SPC monitoring
- **DD^n analysis** (v0.13.0) — 14 waves, 4 phases, 25 concept notes distilled into FDRP Unified Keyprompt v3.8 (~1,350 lines). Notes #24–#25 add principle lifecycle management (Evolution Protocol: ANNOTATE/QUALIFY/REVISE/MERGE/SPLIT/DEPRECATE operations with staleness detection) and resource-aware deployment (Resource Budget Discipline: MINIMUM/STANDARD/FULL triage tiers with contextual priority by task type). Two cross-model review cycles: v3.6 scored MIN 1/5 (Codex Pro + Gemini 3.1), driving 6 CRITICAL remediations in v3.7; v3.7 scored MIN 1.5/5 (Codex Pro + Gemini 3.1 + Claude Opus 4.6), driving 4 CRITICAL + 9 HIGH fixes in v3.8 including detraining tier conflict resolution and budget tier contradiction fixes
- **Batch convergence** (v0.14.0) — First simultaneous multi-run convergence event: 6 runs transitioned ACTIVE → CONVERGING with CVT values 0.172–0.191, zero open hard clashes, zero active Andon events. v19.0 achieves full portfolio convergence: 19 CONVERGED (15 at 1.0000), 0 CONVERGING, 0 ACTIVE across 62 total runs (17 CANCELLED via triage). Source: `fdrp_convergence` and `fdrp_runs`, 2026-03-26
- **Constitution empirical testing** (v0.14.0) — **RETRACTED 2026-05-13.** Original 10 PASS / 14 PARTIAL / 4 FAIL split traced to `fdrp_payload_constitution.effectiveness_score` rows that now uniformly read 1.0 across all 28 principles (W71 counter-example CF-08). **Do not cite for new claims.** W72 Proposal P9 re-run target: 2026-05-20 into `fdrp_constitution_test_results`. Cite v21 §17 for archival purposes only.
- **Oracle Fusion pivot** (v0.14.0) — Run #1015 CVT dropped 0.900 → 0.676 (zone transition ROUGH_CUT → FACETING) after VP-level stakeholder revealed Oracle Seasons (Oracle Fusion Cloud Procurement), crystallising the integration target. YELLOW Andon for capability commitment without verification caught by FDRP methodology. Oracle REST API architecture designed for Phase 1 read-only data extraction. Source: `fdrp_convergence` run 1015, `.scratchpad/oracle-integration-architecture.md`
- **Governance clearance** (v0.14.0) — 4 overdue peer reviews (evo_log_ids 208, 209, 225, 226) processed, achieving zero governance backlog. Stalled run triage: 6 FSM transitions added (ACTIVE→CANCELLED, ACTIVE→PAUSED, ACTIVE→CLOSED, PAUSED→ACTIVE,

PAUSED→CANCELLED, SEED→CANCELLED), 4 runs triaged. Source: `fdrp_evolution_log`, `fdrp_state_transitions`, 2026-03-25

- **Nine new subsystems** (v20.0) — Subsystems #24–#32 built in a single session, adding 66 new tables (schema: 593 → 745 base tables, 132 → 188 views, fdrp_ tables: 156 → 360). Three capability categories: (1) **Cognitive self-awareness** — cognitive architecture (#28, 18 tables) with architect archetypes, ontology registry, modality tracking; thinking chains and structures (#27, 15 tables) with 15 topologies and 10 reasoning chains; cognitive opportunities (#30, 5 tables) with 40 finder archetypes and bias guards. (2) **Operational governance** — daemon taxonomy (#29, 7 tables) with 35 unified daemon types; definition-of-done (#25, 4 tables) with 55 evidence gates; limits-hunt-and-fix (#32, 5 tables) with 68 discovered limits and 136 evaluations. (3) **Integration verification** — wiring-check-and-fix (#31, 4 tables) with 271 wiring checks; gaps-and-opportunities (#26, 3 tables) with 3-horizon scanning; web-finetuning (#24, 5 tables) with ICP-aware Lighthouse optimisation. Source: `information_schema`, `fdrp_subsystems`, 2026-04-02

For whom: Architects of long-lived, multi-institutional programmes where the planning system must evolve faster than the project it governs.

Introduction

The Problem

Planning for complex systems — whether particle accelerator subsystems, safety-critical software, or infrastructure projects — suffers from a quality gap. While manufacturing has Six Sigma, SPC control charts, and ISO 9001 process maturity, planning typically has:

- **No convergence metrics**: How do you know when a plan is "done"?
- **No traceability**: Can you trace a design decision back to the stakeholder concern it addresses?
- **No clash detection**: When two decisions contradict each other, how is this detected before implementation?
- **No grounding discipline**: When an expert says "use a 95% confidence interval," is that number measured or assumed?

FDRP addresses all four gaps by applying manufacturing quality frameworks to the planning process itself.

Motivating Example: The Future Circular Collider

The CERN Future Circular Collider (FCC) [21, 23, 57, 58] exemplifies the scale of project that demands a self-evolving planning system:

- **Physical scale**: 90.7 km circumference tunnel, access shaft depths 180–400 m, 8 surface sites (7 in France, 1 in Switzerland), 4 interaction points
- **Financial scale**: CHF 15 billion over ~12 years for FCC-ee alone; over 800,000 person-years of employment
- **Temporal scale**: Feasibility Study delivered March 2025; Member State decision ~2028; construction begins early 2030s; FCC-ee operations from late 2040s (~15 years); FCC-hh from ~2070s (~25 years) — a project spanning half a century
- **Organisational scale**: 140+ institutes across 30+ countries
- **Technical scale**: FCC-hh targets ~100 TeV centre-of-mass energy (nearly 10× the LHC); 16.4 million tonnes of excavated material; power consumption 1.1–1.8 TWh/year
- **Two machines in one tunnel**: FCC-ee (electron-positron) followed by FCC-hh (hadron-hadron), reusing infrastructure — analogous to LEP→LHC

No static planning methodology can govern a 50-year programme across 140+ institutions. The plan itself must evolve — discovering new expertise domains as they become relevant, integrating new research as it is published, and continuously improving its own quality mechanisms. This is precisely what FDRP's self-referential architecture provides.

The FCC Conceptual Design Report (CDR) [22] established this precedent by involving thousands of contributors across physics, engineering, safety, environ-

mental, and economic domains — exactly the cross-departmental collaboration that FDRP's Expert Teaming Architecture (Section 19) formalises.

The Self-Evolution Principle

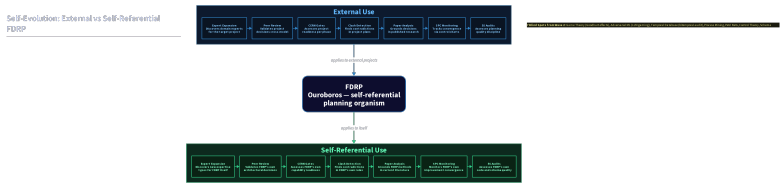
FDRP was originally framed as a **self-evolving planning organism**: the same mechanisms used to plan external projects, turned inward to plan its own improvement. **This framing is reframed by measurement in v23.0** (see Act III, §"What measurement retired"): the self-improvement loop, once instrumented, promoted only 3 of 2,437 candidates (a 0.12 % true-promotion yield) and was retired on 2026-06-10. The mapping below remains a true description of how the process *can* be turned inward — and it is exactly that turning-inward that produced the three falsification experiments in Act III. What the evidence does **not** support is the claim that the inward loop autonomously *appreciates*; what it supports is that the process reliably *produces and verifies* artifacts when pointed at a hard problem. Read the table as the self-application that yielded the honest measurements, not as a claim of autonomous self-improvement:

External Use	Self-Referential Use
Expert expansion discovers domain experts for a project	Expert expansion discovers new expertise types to add to FDRP itself
Peer review validates project decisions	Peer review validates FDRP architectural decisions
CERN gate reviews assess project readiness	Gate reviews assess FDRP capability readiness
Clash detection finds contradictions in project plans	Clash detection finds contradictions in FDRP's own rules
Paper analysis grounds project decisions in research	Paper analysis grounds FDRP's methods in current literature
SPC monitors project convergence	SPC monitors FDRP's own improvement convergence
5S audits assess project planning quality	5S audits assess FDRP's own code and schema quality

This self-referential property — the Ouroboros in the unified metaphor — means FDRP can grow stronger with each run. Each production run can teach the system something new: a new pattern is catalogued, a new heuristic is recorded, a new expert type is discovered, a new anti-pattern is documented. During the flywheel era the system's vocabulary of expertise expanded with each run; since the 2026-06-10 retirement (0.12 % true-promotion yield, see Act III) that expansion happens only when the loop is deliberately pointed at a problem — not autonomously.

Example: In Wave 2 of expert panel expansion, FDRP's own expert panel

(40 experts in that analysis, later expanded to 68+ in the antimatter building programme) identified 7 blind spots that no human operator had noticed — including the need for Game Theory analysis (Goodhart effects in quality gates), Adversarial ML analysis (LLM agents gaming semantic gates), and Temporal Database analysis (bitemporal audit trails). These experts were discovered by the system, not prescribed by a human architect. Expert count is bounded by convergence (when spiral-out produces no new unique domains), not by a predetermined cap.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-041-self-evolution.svg

Cross-Domain Pollination as Systematic Innovation

FDRP itself was built through cross-domain transplantation — bringing ideas from apparently disconnected fields and placing them in front of ultra-specialised experts who had never encountered them:

Source Domain	Transplanted To	Innovation
Toyota Production System	Planning quality	Andon stop-the-line for decision processes
CERN engineering reviews	Software architecture	6-phase gate lifecycle for AI-generated plans
BIM clash detection	Decision analysis	Automated contradiction detection between decisions
SPC control charts	Convergence monitoring	Nelson Rules applied to planning quality metrics
Diamond cutting	Quality metaphor	CVT ratio governing exploration-exploitation dynamics

Source Domain	Transplanted To	Innovation
Automotive CVT	Decision dynamics	Continuously variable gear ratio for refinement phases
Construction punch lists	Quality defects	Pre-commissioning defect tracking for plans
DMAIC (Six Sigma)	Process improvement	Define-Measure-Analyse-Improve-Control for each FDRP iteration
Kaizen	Continuous evolution	Twice-daily quality daemon with Ishikawa root cause analysis
Lean waste types	Planning waste	8 waste categories adapted from manufacturing to planning
Hexagonal architecture	Plugin design	Port/adapter pattern separating FDRP core from renderers
PRISMA (medical research)	Evidence grounding	Systematic literature review methodology for engineering decisions

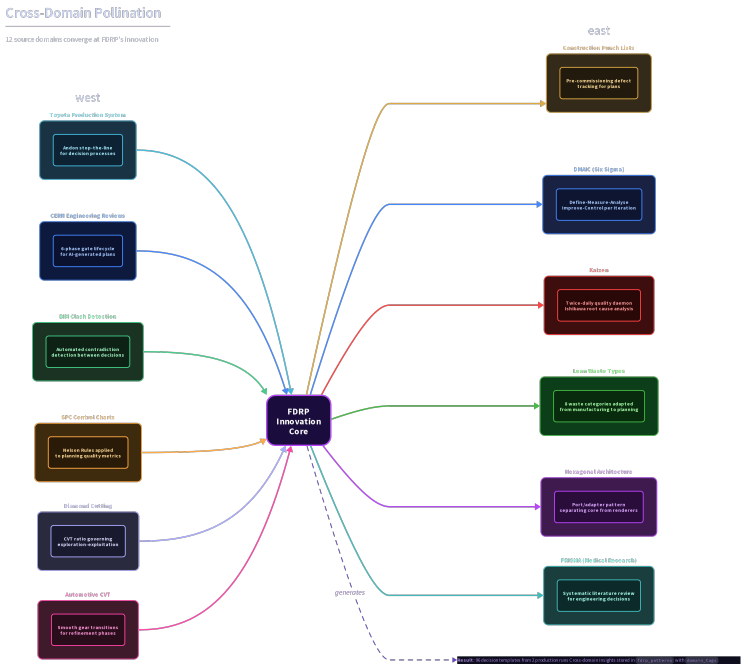
None of these transplants were obvious. A particle physicist would never think to apply Toyota's Jidoka principle to their review process. A software architect would never think to use BIM Level of Detail ladders for planning granularity. A safety engineer would never think to apply diamond-cutting metaphors to convergence dynamics.

This is the core engine of FDRP expansion: bring the best techniques from ANY domain — manufacturing, medicine, construction, automotive, aerospace, software, physics — and place them in front of ultra-specialised experts who have mastered a completely different field. The collision of unfamiliar ideas with deep expertise can generate entirely new thinking.

The expert expansion system formalises this by: 1. **Systematically scanning domains**: Each wave includes 12 "Novel Domain Experts" from fields outside the project's core domain (see Section 5.1) 2. **Structured confrontation**:

Each expert receives the same briefing containing ideas from all other domains — forcing cross-pollination 3. **Recording transplants**: Every successful cross-domain insight is stored in `fdrp_patterns` with `domain_tags` linking source and target domains 4. **Portfolio amplification**: Patterns discovered in one run are automatically seeded into future runs via `fdrp_decision_templates` (96 templates from 2 production runs)

The potential implication for FCC-class projects, if FDRP scales beyond its current single-operator validation, is that FDRP doesn't just bring particle physicists and civil engineers together — it systematically introduces them to ideas from automotive manufacturing, medical systematic reviews, construction commissioning, and software architecture, in a structured format that forces engagement rather than dismissal.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-cross-domain-pollination.svg

The Diamond Metaphor

A raw diamond (a stakeholder concern) undergoes systematic refinement:

1. **Rough Cut** (CVT 0.7–1.0): Broad exploration, many alternatives, high change rate
2. **Faceting** (CVT 0.4–0.7): Structured analysis, alternatives narrowing,

clashes surfacing
3. **Polishing** (CVT 0.25–0.4): Fine-tuning, evidence grounding, cross-scale coherence
4. **Near Convergence** (CVT 0.15–0.25): Perturbation testing, final validation
5. **Converged** (CVT < 0.15): Stable, traceable, ready for commissioning

The diamond metaphor is not decorative — it maps directly to measurable convergence zones tracked by SPC.

CERN Engineering Heritage

FDRP draws heavily from CERN's engineering review lifecycle [1], which governs multi-billion-euro accelerator projects through 6 mandatory review gates:

Gate	CERN Name	FDRP Mapping
KICKOFF	Kick-Off Meeting	Scope + S0 complete
PDR	Preliminary Design Review	S1 complete, literature grounded
CDR	Conceptual Design Review	S2–S3 complete, RAMS analysis done
FDR	Final Design Review	Configuration frozen, S4–S5 complete
TRR	Test Readiness Review	All scales CONVERGED
PQR	Production Qualification Review	COMMISSIONED, intent verified

Each gate is **deliverables-based** (does the artifact exist?), not threshold-based (is the number above X?) — a critical design decision grounded in BIND-021: "No numeric claims without measured data."

Interactive: https://fdrp.liviu.ai/gallery/figures/fig-042-cern-gates.svg

What Changed Since v1.8

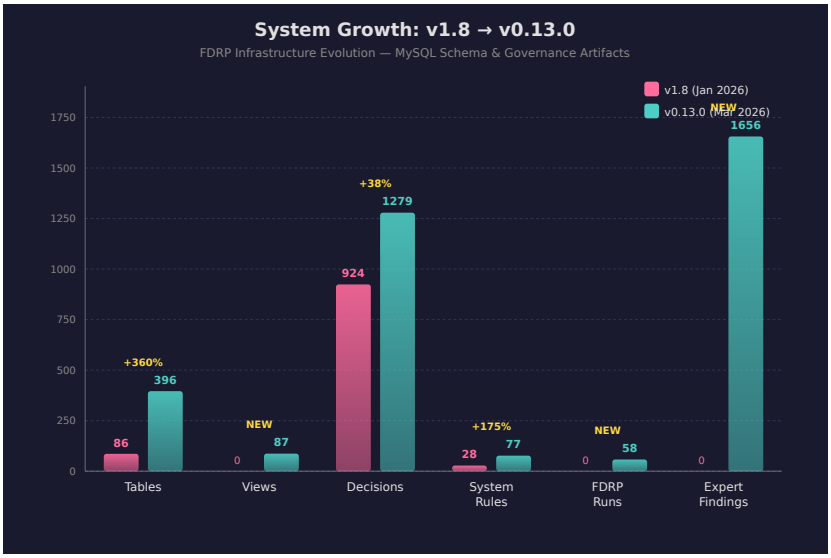
Dimension	v1.8	v0.14.0	Delta
MySQL base tables	86	745	+766%
MySQL views	0	188	New
fdrp_ tables	86	360	+319%
Plugin skills	28	48	+71%
Plugin agents	3	5	+67%

Dimension	v1.8	v0.14.0	Delta
Concept2X pipelines	0	14	New subsystem
Ecosystem elements	0	201	New subsystem
Constitution principles	0	28	New subsystem
Rosetta Stone entries	0	20	New subsystem
Decisions (total, delta)	924	1701	+84% (snapshot 2026-04-16, `SELECT COUNT(*) FROM fdrp_decisions`)
Commissioned FDRP runs (delta)	20	25	+25% (snapshot 2026-04-16, `SELECT COUNT(*) FROM fdrp_runs WHERE status='COMMISSIONED' OR commissioned_at IS NOT NULL`)
FDRP runs (total, delta)	29	70	+141% (snapshot 2026-04-16, `SELECT COUNT(*) FROM fdrp_runs`)
Self-evolution events	0	238	New subsystem
CERN compliance clauses	0	44	New subsystem
CyberDefence tables	0	8	New subsystem (v0.12.0)
Expert persistence: source output lines	0	28,936	New subsystem (v0.13.0)

Dimension	v1.8	v0.14.0	Delta
Ultrathink cascade analyses	0	5	New subsystem (v0.13.0)
Registered subsystems	0	32	9 new in v20.0 (#24–#32)
Cognitive architecture tables	0	18	New subsystem (v20.0)
Thinking chains + structures tables	0	15	New subsystem (v20.0)
Daemon taxonomy types	0	35	New subsystem (v20.0)
Wiring checks performed	0	271	New subsystem (v20.0)
Limits discovered	0	68	New subsystem (v20.0)
DoD evidence gates	0	55	New subsystem (v20.0)
Opportunity finder archetypes	0	40	New subsystem (v20.0)
Runs in CONVERGING status	0	0	All progressed to CONVERGED
Runs in CONVERGED status	0	19	15 at score 1.0000 (#8, #10, #11, #12, #13, #14, #21, #22, #35, #39, #1007, #1008, #1009, #1011, #1015) + 4 at 0.92–0.99
Constitution principles empirically tested	0	28	New (v0.14.0)

FSM state transitions	5	11	+120% (v0.14.0)

Source: *SELECT COUNT(*) FROM information_schema.tables WHERE table_name LIKE 'fdrp_%' and related queries against <intelligence_db>.*



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-061-system-growth.svg

Fifteen major subsystem categories have emerged since v1.8, growing from 6 in v0.14.0 to 32 registered subsystems in v20.0. The original six (concept-to-X pipelines, ecosystem maps, payload constitution, knowledge grafting, Rosetta Stone, self-evolution) were each designed independently to solve specific problems; the progressive disclosure thesis — that all share the same underlying architecture — emerged from the FDRP-on-FDRP meta-analysis (Section 20). v20.0 adds 9 new subsystems (#24–#32) that represent a qualitative jump: the system now has cognitive self-awareness (thinking about its own thinking patterns via #28 cognitive architecture and #27 thinking chains), a unified daemon architecture (#29 with 35 daemon types), integration verification (#31 wiring checks found 271 connection points to audit), and resource governance (#32 limits hunt discovered 68 limits including innodb_buffer_pool_size at 128MB on a 62GB node).

Paper Organisation

Note: Section numbers below refer to pandoc auto-numbering; verify against rendered output. This comprehensive paper presents the theoretical foundations (Section 2), the progressive disclosure thesis (Section 3), the system architecture (Section 4), the twelve core mechanisms (Section 5), the ecosystem architecture

(Section 6), the concept-to-X pipeline platform (Section 7), the payload constitution (Section 8), spiral expert discovery (Section 9), the Rosetta Stone vocabulary routing (Section 10), audience-adaptive translation (Section 11), the digital cognition architecture (Section 12), the self-evolution subsystem (Section 13), knowledge grafting (Section 14), the visualisation engine (Section 15), the data model (Section 16), CERN compliance and safety (Section 17), production results (Section 18), expert teaming architecture (Section 19), cross-domain expert insights (Section 20), continuous improvement infrastructure (Section 21), automated versioning and regression measurement (Section 22), cross-domain validation via SIVP (Section 23), and limitations and threats to validity (Section 24). Part II covers the extended architecture: v0.11.0 ecosystem self-management (Section 25), expert persistence and ultrathink cascade (Section 26), knowledge graph percolation and schema quality gates (Section 27), anti-pattern engine maturation and commercial deployment, and the nine new subsystems with cognitive self-awareness and operational governance (v20.0). The paper concludes with a Summary of Results, DD^n Analysis and Principle Lifecycle, and Conclusion.

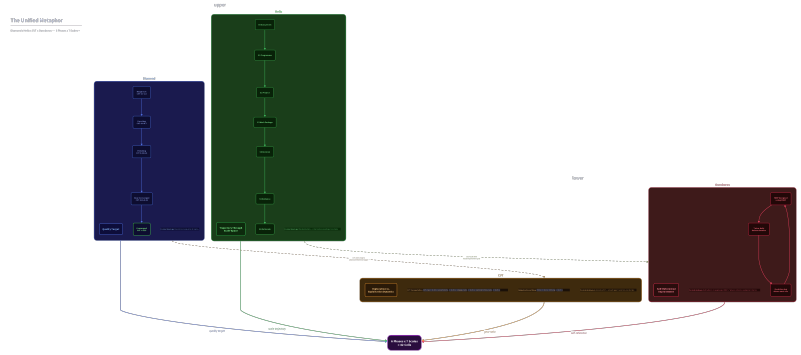
Theoretical Foundations

The Unified Metaphor

FDRP integrates four conceptual frameworks into a single coherent model:

Framework	What It Contributes	Formal Analogy
Diamond	Quality refinement through iterative faceting	Manufacturing quality (6 , 5S)
Helix	Trajectory through scale-space; decisions spiral downward through finer resolution	DNA double helix — information encoding at multiple scales
CVT	Continuously Variable Transmission ratio controlling exploration vs. exploitation	Automotive CVT — smooth gear transition, no discrete shifts
Ouroboros	Self-referential improvement; FDRP plans its own improvement cycles	Hofstadter's strange loops [24]; CERN's "physics informs accelerator design informs physics"

FDRP operates across 7 fractal scales (S0 Ecosystem through S6 Rationale). The system is self-referential: FDRP was designed using FDRP.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-unified-metaphor.svg

CVT Dynamics

The CVT ratio is computed from 4 equally-weighted inputs (all marked UNCALIBRATED per BIND-021 — the weights are proposals, not measured optima):

```
CVT_raw = 0.25 × uncertainty + 0.25 × change_rate + 0.25 × contradiction_rate + 0.25 × rfi_r
```

Where: - **uncertainty** = unvalidated assumptions / total assumptions - **change_rate** = superseded decisions / total decisions - **contradiction_rate** = open hard clashes / total decisions - **rfi_rate** = open RFIs / total RFIs

Smoothing uses Heijunka (Toyota levelling) [6]: `CVT_smoothed = 0.3 × CVT_raw + 0.7 × CVT_previous`

This prevents single-iteration spikes from triggering premature zone transitions.

Polarity convention (canonical — clarified throughout, v23.0). `cvt_ratio` low = good; `convergence_score` high = good; never compare the two directionally without naming the field. `cvt_ratio` is a *residual-disagreement* metric, so **lower `cvt_ratio` means more converged.** Convergence is reported as **`convergence_score = 1 − cvt_ratio`, where higher is better**, and **CONVERGED is reserved for high `convergence_score` (equivalently, low `cvt_ratio`).** This single convention is used at every threshold mention in this paper; where legacy prose elsewhere speaks of "low CVT is good," read it as the same statement on the `cvt_ratio` residual scale, never mixed with the `convergence_score` scale in one comparison. It matters because two gates once read the same `cvt_ratio` field with opposite polarities and manufactured a false COMMISSIONED verdict; that collision was reconciled (Act III, §"What measurement retired", commits `f98633b`/`f45142e5`). Where this paper writes a threshold such as "CVT 0.86–0.87" for the Wave R11 panels, that is a *convergence* reading on the high-is-good scale; where the raw `cvt_ratio` residual is meant, it is named explicitly.

Fractal Scale Hierarchy

FDRP operates across 7 scales, each with its own quality rubric:

Scale	Code	Focus	Quality Rubric	LOD Range
S0	Ecosystem	Strategic context, constraints	PESTLE + Wardley mapping	100–200
S1	System	Quality attributes, viewpoints	ISO 42010 correspondence rules	100–300
S2	Domain	Bounded contexts, capabilities	DDD (MECE, coupling ratio)	200–400
S3	Component	Responsibilities, interfaces	SRP, contract-first, DAG within context	200–400
S4	Interface	Preconditions, postconditions	Formal DbC, all errors enumerated	300–500
S5	Decision	ADR format, alternatives	2+ alternatives, consequences, reversibility	300–500
S6	Rationale	Falsifiable premises	Argument acyclicity, no circular reasoning	400–500

LOD (Level of Detail) follows BIM conventions (LOD 100–500) adapted from the construction industry [2], providing a standardised vocabulary for "how detailed is this decision?"

Referenced Standards and Frameworks

FDRP synthesises 20 frameworks (stored in `fdrp_frameworks` table) and 9 ISO standards (stored in `fdrp_iso_standards`):

Manufacturing Quality: Toyota Production System (Ohno 1988) [59], 5S (Hiroyuki 1995), Six Sigma (Motorola/GE), SPC (Shewhart 1931) [17], PDSA (Deming 1986) [18], Shingo Model for Operational Excellence [60]

Systems Engineering: ISO 42010:2022 (Architecture Description), INCOSE GtWR V4, SysML, FAST bidirectional decomposition

Safety: IEC 61508:2010 (Functional Safety), IEC 60812 (FMEA/FMECA), FIDES Guide 2022 (Failure Rate Prediction), DO-178C (Software Airworthiness)

Construction/BIM: BIM LOD Specification (BIMForum), Clash detection (Navisworks model), Commissioning (ASHRAE Guideline 0)

Requirements: Volere 10 Tests, BABOK V3 9 Characteristics, IEEE 830-1998

Review Process: CERN OSPO High-Reliability Guide [1], Cooper Stage-Gate [3], NASA Standing Review Board [4]

Related Work

FDRP builds on a substantial body of prior work in autonomous systems, trade-off analysis, model-based systems engineering, and cross-domain innovation. This section positions FDRP relative to key predecessors.

MAPE-K (Autonomic Computing): IBM's Monitor-Analyse-Plan-Execute-Knowledge loop (Kephart & Chess, 2003) provides the canonical

feedback architecture for self-managing systems. FDRP shares MAPE-K's closed-loop structure but extends it in three ways: (1) fractal application across 7 scales rather than a single control loop, (2) multi-model verification (3 independent LLMs) rather than single-agent analysis, and (3) self-referential improvement where the planning loop plans its own evolution.

NASA Trade-off Analytics: NASA's trade-off analysis frameworks for complex missions (e.g., Design Reference Missions, Figures of Merit) [15] provide structured multi-criteria evaluation. FDRP differs by treating trade-offs as a continuous convergence process (CVT dynamics) rather than discrete selection events, and by applying SPC to monitor trade-off stability over iterations.

MBSE Tools (Cameo Systems Modeler, Capella): Model-Based Systems Engineering tools provide formal architecture description and requirements traceability. FDRP complements rather than replaces MBSE by adding quality-process layers (SPC, 5S, Andon) that MBSE tools do not provide, and by introducing self-evolving expert expansion that discovers missing analysis dimensions. FDRP's RTM and requirements tables could integrate with SysML/Capella exports.

Cross-Domain Innovation Prior Art: Hargadon & Sutton (1997) documented "technology brokering" — firms that innovate by recombining ideas across domains — in product development contexts. Johansson (2004) popularised this as "The Medici Effect": breakthrough innovation at the intersection of diverse fields. FDRP operationalises these insights through systematic expert expansion waves that include 12 "Novel Domain Experts" per wave (Section 5.1), structured cross-domain confrontation via shared briefings, and pattern recording in `fdrp_patterns` for portfolio amplification. Where Hargadon & Sutton described a human organisational phenomenon and Johansson articulated a principle, FDRP provides an automated, measurable process.

Cognitive Architectures for Language Agents (CoALA) [37]: Sumers et al. map LLM-based agents onto classical cognitive architectures (ACT-R [31], SOAR [32]), proposing a taxonomy of memory, action, and decision modules. FDRP's Digital Cognition Architecture (Section 12) extends CoALA by introducing multiple *thinking modalities* as first-class cognitive components, with SPC-based effectiveness tracking and self-discovery of new modalities through expert expansion.

AI Agent Architecture Surveys [38]: Masterman et al. survey emerging agent architectures for reasoning, planning, and tool calling. FDRP differs from the architectures surveyed by combining agent orchestration with manufacturing quality frameworks (SPC, 5S, CERN gates) — a combination not present, to our knowledge, in any surveyed system — and by operating self-referentially across fractal scales.

Additional Related Methodologies: Several established methodologies address overlapping concerns. Quality Function Deployment (QFD) translates stakeholder needs into design requirements — FDRP's payload constitution

serves an analogous function for expert briefings. TRIZ (Theory of Inventive Problem Solving) provides systematic cross-domain pattern transfer; FDRP's expert expansion spiral is a related but distinct mechanism that discovers new expertise domains rather than applying known inventive principles. Design Structure Matrices (DSMs) manage decision dependencies in complex engineering projects; FDRP's clash detection addresses a similar concern at a different abstraction level. The Delphi method uses structured multi-expert elicitation with iterative feedback — FDRP's ultrathink cascade extends this pattern with cross-domain LLM-generated specialists at much greater analytical depth (500–1,500 lines per expert vs. typical Delphi survey responses). A detailed positioning against each methodology is beyond the scope of this paper but represents important future work.

De Bono's lateral thinking [34] provides a theoretical foundation for FDRP's Six Thinking Hats debate (Section 5.4) and the broader principle that structured cognitive diversity produces better decisions than convergent analytical thinking alone.

Progressive Disclosure — The Unifying Architecture

The Thesis (HYPOTHESIS — Under Test)

We propose that progressive disclosure is the unifying architecture of the entire FDRP ecosystem. Not just a documentation pattern (BIND-016). Not just a UI technique. In our analysis, it operates as the organising principle at every level: agent identity, payload delivery, ecosystem routing, matching, naming, knowledge compression, and self-evolution.

This claim is a **hypothesis**, not a proven theorem. We present it with three levels of skepticism per BIND-021:

1. **The thesis itself is a hypothesis** — it must be tested through falsification (find a subsystem where progressive disclosure does NOT apply)
2. **The methods proposed by experts are proposals** — each must be validated against this specific ecosystem
3. **All numeric values are ungrounded until measured** — expert-proposed numbers are starting hypotheses, not facts

Seven Dimensions of One Principle

Progressive disclosure manifests in seven dimensions across the FDRP ecosystem:

Dimension 1: Recursive Keyword Compression. Every knowledge artifact exists at multiple resolution levels. Generation happens at L0 (full prose, ~40 tokens); compression yields L1 (keywords, ~12 tokens), L2 (key-keywords, ~6 tokens), L3 (meta-keywords, ~4 tokens), L4 (domain-keywords, ~2 tokens), and L5 (atomic, 1 token). Each level contains keywords OF keywords. Example:

```
L0: "The agent analyzes thermal stress patterns in FCC dipole magnets"
L1: "analyze thermal-stress FCC dipole magnets"
L2: "thermal-stress FCC dipole"
L3: "thermal FCC"
L4: "CERN thermal"
L5: "CERN"
```

Navigate at whatever resolution you need. Models instructed to output key-keywords enable ultra-knowledge compression.

Dimension 2: @-Notation Routing. The ecological address `action@domain@depth@model` encodes an agent's position in the ecosystem. Wildcarding from right to left yields progressive disclosure of routing specificity:

```
analyze-thermal-stress@cern-fcc@wave3@opus  → L0 (exact match)
analyze-thermal-stress@cern-fcc@wave3@*     → L1 (any model)
analyze-thermal-stress@cern-fcc@*@*         → L2 (any depth)
analyze@cern@*@*                            → L3 (broad match)
*@cern@*@*                                  → L4 (domain only)
*@*@*@*                                     → L5 (default route)
```

This mirrors IP routing (/8 → /32), DNS resolution (`.com` → `mail.google.com`), and the Dewey Decimal system. More `@` segments = more specific = long tail.

Dimension 3: Long-Tail Economics. Most useful matches occur in the long tail (inspired by Zipf/Pareto distributions [48] and Anderson's long tail economics [56]). HEAD (L4–L5) provides broad keywords with many matches but low precision. TORSO (L2–L3) provides moderate specificity. TAIL (L0–L1) provides ultra-specific queries with high conversion — this is where specialists live. The long tail IS the specialist ecosystem.

Dimension 4: Multi-Resolution Matching. Match at the broadest level first (L5 cluster selection), then drill down to exact (L0):

```
Query: "analyze thermal stress in FCC dipole magnets"
L5: CERN cluster         → 13 candidates
L4: CERN + thermal       → 4 candidates
L3: thermal + FCC        → 2 candidates
L0: exact match          → 1 candidate
```

Complexity is O(k) where k = disclosure levels, NOT O(n) where n = total agents. At 204 elements (snapshot 2026-06-10) with k=5 levels, this means 5 comparisons instead of 204.

Dimension 5: Rosetta Stone (Acronym → Expert → Knowledge). Acronyms are the most compressed keywords (L5). They ROUTE you to the expert who holds the knowledge (L3). The expert's analysis IS the knowledge (L0). The mapping is the routing table:

L5 (Acronym)	L3 (Expert Domain)	L0 (Actionable Knowledge)
CDA	Quantitative Finance / HFT	Task matching IS order matching; effectiveness = prices (Hayek [49])
LEP	Network Routing / BGP	O(k) prefix matching on @-segmented addresses
HMM	Molecular Biology / HMMER	Profile-HMMs discover new agent types from deployment history
SPC	Quality / Six Sigma	Nelson Rules detect quality drift in skill effectiveness
IPT	FDRP Payload	Insight Per Token — efficiency metric for expert briefings

Source: `SELECT acronym, expert_domain, key_insight FROM fdrp_acronym_index` *(36 entries, snapshot 2026-06-10).*

Dimension 6: Audience-Adaptive Translation. The same knowledge, different packaging. Opus generates L0 expert knowledge (expensive, deep). Sonnet translates to L2–L4 for different audiences (cheap, clear). The source of truth is ALWAYS the L0 expert output; translations are views, never replacements.

Audience	Analogies From	Disclosure Level
Expert	Same domain	L1 (near-full)
Engineer	Construction, engineering	L2
Architect	Software patterns, design	L2–L3
Manager	Business, ROI, risk	L3
Newcomer	Everyday objects, kitchen	L4
Public	Sports, cooking, travel	L5

Dimension 7: Cost-Optimal Model Selection. Each model does what it is BEST at: Opus for generation (discovery requires the highest intelligence), Sonnet for translation and summaries (clarity doesn't require genius), and Opus for verification (a yes/no check — far less expensive than full L0 generation).

The Fractal F

One algorithm at N levels. The same matching/routing/compression pattern applies whether you are:

- Finding an agent (ecosystem map matching → LEP on eco_address)
- Optimising a payload (keyword-level compression → L0–L5 structure)
- Routing a task (LEP on pipeline @-addresses)

- Discovering new agent types (profile gaps → new species)
- Pricing agent effectiveness (scores → market prices)
- Detecting ecosystem health (SPC → Nelson Rules → regime classification)

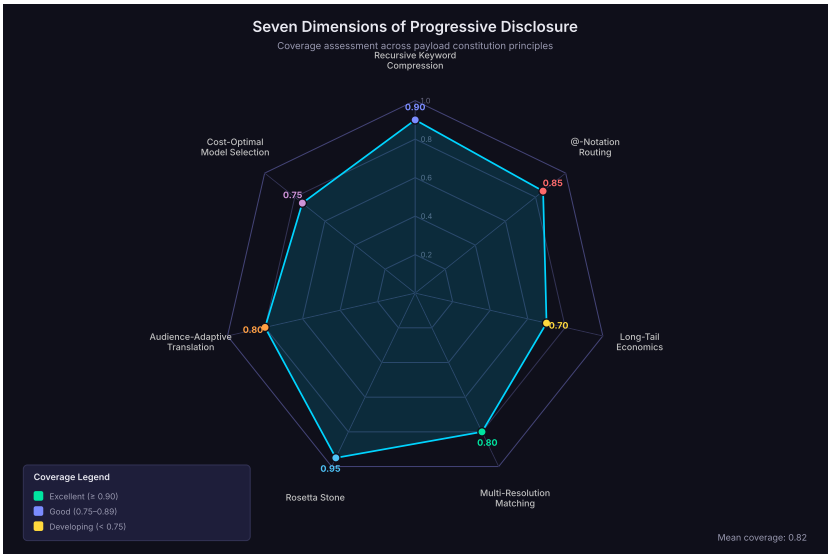
In our analysis, this self-similarity across levels constitutes the fractal structure. The "F" in FDRP is fractal — and we propose that progressive disclosure is the fractal's generating function.

Evidence: This Document

This document itself implements progressive disclosure:

- Title = L5 keyword ("FDRP")
- Section headers = L3–L4 keywords ("Recursive Keyword Compression", "Long-Tail Economics")
- Table cells = L2–L3 keywords ("LambdaMART unified ranking", "CDA market mechanism")
- Full paragraphs = L1 content
- Expert analyses (200KB stored in `expert-analyses/full/`) = L0 full prose

You can navigate at whatever resolution you need.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-001-pd-dimensions.svg

System Architecture

As of v0.8.0 (2026-06-10), FDRP's measured identity is: a **generation engine** (the concept2X pipelines) plus a **deterministic toolbelt** (fdrp-sql gate, charts, freeze/lint/RAMS tooling) plus **honest gates** (pre-registered A/Bs, drift tables, retraction discipline) — **not** an orchestration framework. Most tasks need a bare

native workflow; the toolbelt's routing front door is `/fdrp-method-select` (3-axis routing) with `/knowledge-graph-select` for corpus selection. The architecture below should be read against the Act III verdicts: wrapper machinery beat native nowhere measured on Opus (prompt-parity A/B, NATIVE_NOW).

Implementation Stack

Layer	Technology	Purpose
AI Runtime	Claude Opus 4.8 (primary), Codex Pro / GPT-5.5 (verification), Gemini (gemini-pro-latest) (verification)	Decision generation, cross-model critique
Plugin System	FDRP plugin v0.8.0 (plugin semver; re-cut 2026-06-10 — distinct from the March 2026 fdrp_versions DB-release numbering)	49 active skills + 6 deprecated (archived), 4 agents, 6 registered hook scripts
Data Layer	MySQL 8.0 (InnoDB)	326 fdrp_ base tables, 384 views (total), 159 triggers (snapshot 2026-06-10; base-table count decreased 360→326 via consolidation, not feature loss)
Visualisation	Babylon.js, Cytoscape.js, D3.js v7	8 interactive HTML renderers
Quality Daemon	Bash + systemd	Twice-daily automated quality audits
Version Control	Git (Gitea self-hosted)	All decisions committed within seconds (BIND-009)

Plugin Architecture

The FDRP plugin (`<agent_home>/.claude/plugins/fdrp/`) implements a complete lifecycle management system:

```
plugins/fdrp/
  plugin.json              # v0.8.0, 49 active skills registered (6 deprecated in archive/2
  DOCUMENTATION.md         # Full process chain documentation
  schema-reference.md      # Auto-generated table/column reference
  bin/                     # Utility scripts (schema generation, etc.)
  skills/                  # 49 active skill directories, each with SKILL.md (tree abridged
```

```
    fdrp-start/          # Lifecycle: initialise run
    fdrp-descend/        # Dynamics: PDSA + CVT computation
    fdrp-freeze/         # Lifecycle: SHA-256 configuration freeze
    fdrp-commission/     # Lifecycle: intent verification
    fdrp-review/         # CERN 6-phase gate evaluation
    fdrp-rams/           # FMEA/FTA/HAZOP with IEC 61508 SIL
    fdrp-research/       # ARCHIVED v0.8.0 cut → archive/2026-06-10-v08-cut/
    fdrp-peer-review/    # ARCHIVED v0.8.0 cut → archive/2026-06-10-v08-cut/
    fdrp-6hats/          # ARCHIVED v0.8.0 cut → archive/2026-06-10-v08-cut/
    fdrp-premortem/      # ARCHIVED v0.8.0 cut → archive/2026-06-10-v08-cut/
    expert-expansion/    # Convergence-based expert waves
    fdrp-fork/           # MOE cache fork (3-way diversity)
    fdrp-5s/             # 5S quality audit
    fdrp-andon/          # TPS stop-the-line signal
    fdrp-status/         # Convergence dashboard
    fdrp-resume/         # Post-compaction recovery
    fdrp-lint/           # CERN typographic linter
    fdrp-rfi/            # Request For Information CRUD
    fdrp-punch/          # Punch list defect tracking
    fdrp-viz-graph/      # Sugiyama DAG (Cytoscape.js)
    fdrp-viz-3d/         # 3D decision landscape (Babylon.js)
    fdrp-viz-rams/       # FMEA risk matrix (D3.js)
    fdrp-viz-review/     # Gate review dashboard (Babylon.js)
    fdrp-viz-trace/      # Bipartite traceability (Cytoscape.js)
    fdrp-viz-convergence/ # CVT timeline (D3.js)
    fdrp-viz-twin/       # Digital twin combined view (Babylon.js)
    fdrp-viz-radial/     # Radial ecosystem view (D3.js)
    ...                  # + concept2* family (image/onepager/prd/demo/website/mvp/
                           #   experiment/digitaltwin/payload/full-software/integration),
                           #   fdrp-method-select, knowledge-graph-select, fdrp-rosetta,
                           #   fdrp-explain, fdrp-graft, fdrp-ecosystem-map,
                           #   fdrp-expert-research, star2paper, star2mail,
                           #   stock-ddn-analysis, pdf-quality-gate, fdrp-paper-charts,
                           #   fdrp-paper-update, fdrp-subsystem-eval (49 active total)
agents/                  # 4 autonomous agents (+1 archived)
    fdrp-orchestrator.md   # Master diamond cutter - CVT management
    scale-analyst.md       # Per-scale refinement specialist
    clash-detector.md      # BIM-style clash detection (read-only)
    convergence-monitor.md # SPC control chart specialist
    evolution-monitor.md   # ARCHIVED v0.8.0 cut - flywheel retired 2026-06-10
hooks/                   # Quality gate enforcement
    hooks.json           # Hook configuration (6 registered scripts)
    fdrp-skill-lint.sh       # PreToolUse: SKILL.md structural lint
    fdrp-anti-pattern-guard.sh # PreToolUse: known failure patterns
    fdrp-plan-poka-yoke.sh   # PreToolUse: plan validation
    fdrp-plan-prefilter.sh   # Deterministic plan file detection (feeds poka-yoke)
```

```
    fdrp-decision-logger.sh  # PostToolUse: SPC datapoint extraction
    fdrp-error-capture.sh    # PostToolUse: fishbone SQL error correction
    fdrp-skill-invoke-counter.sh # PostToolUse: R-INSTR invocation counter
templates/               # Visualisation base templates
    babylon-base.html    # 3D engine template (Babylon.js)
    cytoscape-base.html  # Graph engine template (Cytoscape.js + dagre)
    d3-base.html         # 2D chart template (D3.js v7)
```

The 13-Step Process Chain

FDRP follows a standard process chain for each run. v0.8.0 note (2026-06-10): steps 3–6 and 10 were served by dedicated plugin skills — /expert-expansion remains active, while /fdrp-peer-review, /fdrp-6hats, /fdrp-research and /fdrp-premortem were deprecated in the v0.8.0 cut; those steps now run natively or via cross-model-verify.sh per the prompt-parity NATIVE_NOW verdict. The chain remains the canonical lifecycle:

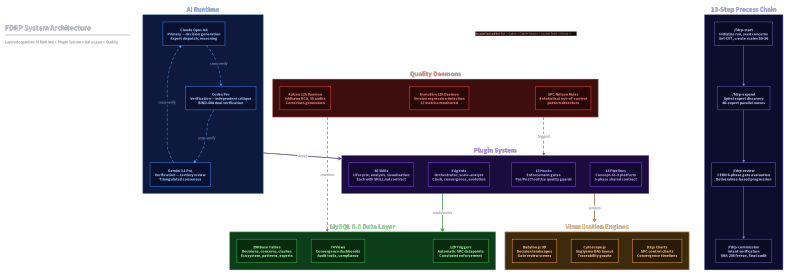
FDRP Process Chain

1. /fdrp-start → Initialise run, discover experts
2. /fdrp-descend → PDSA per scale, CVT dynamics
3. /expert-expansion → Convergence-based expert waves
4. /fdrp-peer-review → Iterative Codex + Gemini critique
   (5 fails →)
5. /fdrp-6hats → Structured 6-Hat debate
6. /fdrp-research → Bibliographic grounding (PDR gate)
7. /fdrp-rams → FMEA/FTA/HAZOP analysis (CDR gate)
8. /fdrp-freeze → SHA-256 configuration freeze (FDR gate)
9. /fdrp-review → CERN 6-phase gate evaluation
10. /fdrp-premortem → Pre-mortem + red team probing
11. /fdrp-commission → Intent verification (CONVERGED→COMM.)
12. /fdrp-lint → CERN typographic linting
13. Package → Reproducibility package (SHA-256)

```
Support: /fdrp-andon, /fdrp-rfi, /fdrp-punch, /fdrp-5s,
         /fdrp-fork, /fdrp-status, /fdrp-resume

Visualisation: /fdrp-viz-graph, -3d, -rams, -review,
               -trace, -convergence, -twin

Escalation ladder:
  Self → Codex → Codex+Gemini → Cache+Seed → 6 Hats → Human
```



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-036-fdrp-architecture.svg

Core Mechanisms

Expert Expansion (Convergence-Based Waves)

Disclosure: The "experts" in FDRP's expert expansion are LLM-generated specialist personas, not human domain experts. Each persona is constructed with a detailed professional profile, publication history, and domain-specific vocabulary. The names used (e.g., in Section 20.1) are fictional labels for these personas. The validity of this approach rests on the observation that LLMs trained on domain literature can produce critiques that overlap substantially with human expert critique — an empirical claim that requires formal validation through comparative studies (see Section 24, Limitations).

The expert expansion system deploys domain experts in convergence-based parallel waves, each providing independent analysis. In production, the FDRP-on-FDRP analysis used 40 experts; the antimatter building programme expanded to 68+ experts across two rounds. Expert count is bounded by convergence (when spiral-out produces no new unique domains), not by a predetermined cap. The roster spans 12 expert types:

Category	Experts	Example Domains
Core Domain (12)	MySQL Performance, DB Security, QA/SPC, BIM, FTA, Systems Thinking, Observability, Lean/TPS, Security Red Team, Hexagonal Architecture, Systems Performance, Business Analysis	Query plans, privilege escalation, SPC Cpk, LOD validation
Extended Domain (6)	Formal Verification, Chaos/Resilience, Contract Testing, ISO 19011, Event Sourcing, Human Factors	TLA+ state machines, fault injection, DbC, audit trails
Novel Domain (12)	Graph Theory, Game Theory, Distributed Systems, Aviation Safety, Process Mining, Cryptographic Integrity, Adversarial ML, SRE, Temporal DB, Schema Evolution, Petri Nets, Control Theory	DAG cycle detection, Goodhart effects, ACID isolation, DO-178C comparison
Visualization (8–10)	Information Visualization, Knowledge Graph Engineering, MBSE, CERN Data Viz, Digital Twin, BIM Viz, UX/Dashboard, Risk Visualization	Tufte principles, Sugiyama layout, ROOT patterns, bow-tie diagrams

Wave convergence criterion: Waves continue until no new TIER 1 findings emerge. In production, Wave 1 produced 40 proposals; Wave 2 produced 259 proposals with 10,360 votes, 47 TIER 1 findings (18.1%), achieving 9/9 cross-model agreement.

Cross-Model Verification

Every expert finding undergoes independent verification by two additional models:

1. **Codex Pro / GPT-5.5** (OpenAI): `codex2 exec - < prompt.txt` (file-based per BIND-050 — inline prompt arguments enter agent mode and do not work as printed)
2. **Gemini (gemini-pro-latest)** (Google): `gemini < prompt.txt` (keep stdin under ~4 KB)

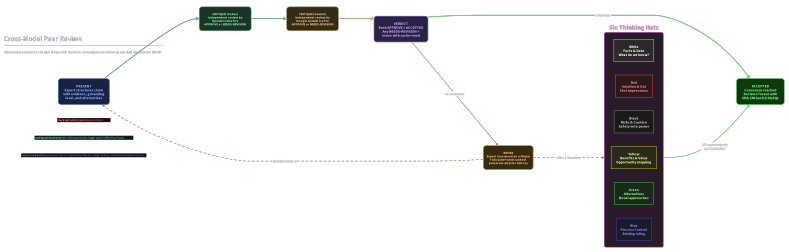
This implements aviation-inspired dual verification (BIND-008): no single model's output is trusted without independent confirmation. Disagreements force council review. Critically, disagreements between models are MORE valuable than agreements — they expose blind spots invisible to any single model (concept note #19: Multi-Model Collaboration). See "Cross-Model Verification of the Keyprompt" (in Conclusion and Roadmap) for the full cross-model review: v3.6 scored MIN 1/5 driving 6 CRITICAL remediations in v3.7, then v3.7 scored MIN 1.5/5 driving 4 CRITICAL + 9 HIGH fixes in v3.8 — demonstrating the multi-model collaboration principle applied iteratively to the protocol itself.

Peer Review — Presenter-Critique Loops

The iterative peer review protocol implements the "presenter-critique until agreed" doctrine (the dedicated /fdrp-peer-review plugin skill was deprecated in the v0.8.0 cut, 2026-06-10; the loop now runs through native workflow plus cross-model verification):

```
ITERATION N:
  1. PRESENT: Expert structures claim with evidence, grounding level, alternatives
  2. CRITIQUE (Codex): Independent review, APPROVE or NEEDS-REVISION with reasons
  3. CRITIQUE (Gemini): Independent review, APPROVE or NEEDS-REVISION with reasons
  4. VERDICT:
     - Both APPROVE → ACCEPTED
     - Any NEEDS-REVISION → Expert revises with cache+seed context
     - After 5 iterations → Escalate to /fdrp-6hats
```

The **cache+seed pattern** preserves all prior critique history in the revision prompt, preventing context loss across iterations.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-040-peer-review.svg

Six Thinking Hats Structured Debate

For S5 (Decision scale) or escalated disagreements, FDRP deploys De Bono's Six Thinking Hats [5, 34] as 6 parallel agents (the dedicated /fdrp-6hats plugin skill was deprecated in the v0.8.0 cut, 2026-06-10; the escalation method survives in doctrine). An open limitation: the Red Hat's no-justification gut input sits in tension with BIND-021's measured-data requirement (task #605):

Hat	Role	Agent Type	Key Power
White	Facts & Data	researcher	Provides only verified data
Red	Intuition & Emotion	—	Gut reactions, no justification needed
Black	Risk & Critique	security-red-team	Safety veto (overrideable only at 5/6 vote)
Yellow	Optimism & Value	—	Benefits, opportunities, ROI
Green	Creativity	—	Alternatives, lateral solutions
Blue	Synthesis (runs LAST)	—	Binding ruling, evidence chain

Voting: 2/3 supermajority (4 of 6 SUPPORT) required. Black Hat safety veto requires 5/6 to override. Three rounds for deadlock resolution. Blue Hat's ruling is binding and stored with full evidence chain in JSON.

Configuration Freeze with SHA-256

When a design reaches FDR gate readiness, FDRP freezes the configuration baseline:

1. All ACCEPTED decisions for the run are serialised
2. SHA-256 hash computed over the serialised set

3. MySQL trigger rejects any UPDATE on frozen decisions
4. Superseding a frozen decision re-triggers FDR/TRR/PQR gate re-evaluation

This mirrors CERN's Technical Design Report (TDR) freeze process, where post-freeze changes require formal Engineering Change Requests (ECR).

RAMS Analysis (IEC 61508)

FDRP's RAMS module supports 6 analysis methods:

Method	Standard	Use Case
FMEA	IEC 60812	Systematic failure mode enumeration
FTA	IEC 61025	Top-down fault tree construction
HAZOP	IEC 61882	Deviation-based hazard identification
PRELIMINARY	—	Initial risk screening
FMEDA	IEC 61508-2	Hardware diagnostic coverage
FIDES	FIDES Guide 2022	Failure rate prediction (preferred over MIL-HDBK-217)

Key fields per analysis entry: - **RPN** (Risk Priority Number) = severity × occurrence × detection (auto-computed, GENERATED column) - **SIL target** (SIL 1–4, per IEC 61508) - **Grounding level** (VL0 estimate through VL4 measured) — VL0/VL1 entries stored but excluded from safety calculations - **PFH** (Probability of Failure per Hour), **SFF** (Safe Failure Fraction), **MTBF/MTTR** — all with grounding source

Interactive: https://fdrp.liviu.ai/gallery/figures/fig-024-rams-matrix.svg

5S Quality Auditing

Each FDRP run receives periodic 5S audits (adapted from Lean manufacturing):

Dimension	Planning Interpretation	Metric
Sort	Dead artifact ratio (unused decisions, expired assumptions)	0–10
Set-in-Order	FK integrity + orphan decision ratio	0–10
Shine	Missing rationale/alternatives on active decisions	0–10
Standardise	Unfulfilled predictions from past iterations	0–10
Sustain	Trend slope of last 5 audits	0–10
Composite	Average of above 5	AUTO-GENERATED

Production results show improvement from 7.0 (iteration 1) to 9.4 (iteration 8) on the INDIGO AirGuard run — confirming PDSA improvement trajectories.

SPC Control Charts (Nelson Rules)

The convergence-monitor agent maintains SPC control charts [17] on all convergence metrics, applying Nelson Rules [55] for special-cause detection:

- **Rule 1**: Single point beyond 3 (0.27% probability) → YELLOW Andon
- **Rule 2**: 9+ consecutive points same side of centre → shift detected → YELLOW

- **Rule 3**: 6+ consecutive monotonic points → trend detected → YELLOW (but: convergent trend is expected — only alarm on divergent trends)
- **Rules 4–6**: Deferred (require more data points)

3+ simultaneous rule violations trigger RED Andon (all spirals paused).

Andon Stop-the-Line Signals

Borrowed directly from Toyota's Jidoka principle [6]:

- **YELLOW**: Pauses only the pulling spiral. Clash-detector investigates. Auto-escalation: 3+ hard clashes in one scale → RED.
- **RED**: Pauses ALL active spirals in the run. Requires human attention.

23 Andon events recorded across production runs (snapshot 2026-06-10).

MOE Cache Fork (Mixture-of-Experts)

For complex decisions, FDRP forks the analysis into N parallel variants (default 3, range 2–5) using different prompt biases from seed perspectives:

Seed Category	Example Bias
QUALITY	"Prioritise reliability, maintainability, and safety margins"
ECONOMIC	"Prioritise cost-effectiveness and resource efficiency"
TECHNICAL	"Prioritise technical elegance and performance"
STRATEGIC	"Prioritise long-term flexibility and market positioning"

All variants share the same prompt prefix (cache hit ~1.3× cost, not 3×) [UNGROUNDED]. Results are aggregated via majority vote (>50% consensus). Disagreement rate >0.15 triggers YELLOW Andon.

Pre-Mortem and Red Team

Before commissioning, FDRP runs proactive failure analysis (addressing META-001: "system only improves after failures"):

Phase 1 — Pre-Mortem (Klein 2007 [7]): 3 independent agents imagine failure scenarios: - Technical failure agent - Process failure agent - Integration failure agent

Phase 2 — Red Team: Adversarial agent probes 6 vulnerability types: 1. Contradiction clashes (unsurfaced) 2. Stale assumptions (>7 days unvalidated) 3. Orphan requirements (no traceability) 4. VL0 RAMS entries driving safety decisions 5. Gate bypass vectors 6. Ungrounded numeric claims

Phase 3 — Triage: CRITICAL → block + RED Andon; HIGH → block + ACTION_ITEMs; MEDIUM → commission with conditions; LOW → clear.

Poka-Yoke Hooks (Error-Proofing)

Six registered hooks enforce quality at the tool level (hooks.json, v0.8.0): three PreToolUse — `fdrp-skill-lint.sh`, `fdrp-anti-pattern-guard.sh`, `fdrp-plan-poka-yoke.sh` (with deterministic `fdrp-plan-prefilter.sh`) — and three PostToolUse — `fdrp-decision-logger.sh`, `fdrp-error-capture.sh`, and `fdrp-skill-invoke-counter.sh`, the R-INSTR instrumentation (commit 51ba127) that produced the 0-invocation skill-theatre measurement cited in Act III:

PreToolUse — Plan Validation: - Required sections check (Concerns, Decisions, Assumptions, Traceability) - Decision rationale enforcement (every decision must have a "why") - Constraint isolation gate (execution constraints banned from design scales S1–S6)

PostToolUse — Decision Logger: - Extracts DECISION/ASSUMPTION/RFI markers from plan files - Feeds SPC datapoint stream for control chart computation

PostToolUse — Error Capture (Fishbone v2): - Captures MySQL errors (1054 Unknown Column, 1146 Table Not Found, 1644 Trigger Block) - Injects correct schema as `additionalContext` into agent's next turn - Addresses META-007: "monitoring pointed at wrong target" — doesn't just log errors, prevents repetition

PreToolUse — Skill Lint (`fdrp-skill-lint.sh`): - Structural lint of SKILL.md files at write time (front matter, nested-fence, size budget)

PreToolUse — Anti-Pattern Guard (`fdrp-anti-pattern-guard.sh`): - Blocks known failure patterns (AP catalogue) in plan content before they land

PostToolUse — Skill Invoke Counter (`fdrp-skill-invoke-counter.sh`, R-INSTR, commit 51ba127): - Increments `skill_registry` invocation counts per skill use — the instrumentation that produced the 0-invocation "skill theatre" measurement in Act III

Ecosystem Architecture

Why Ecosystem Is the Right Frame

The FDRP agent coordination system is an **ecosystem**, not a topology, not a multi-layer network, not a category-theoretic construction. Those mathematical frameworks are TOOLS to describe ecosystem properties — they are not the frame itself.

An ecosystem is a living system that grows from within. Multiple interaction patterns coexist simultaneously — like a construction site where hierarchy, safety

chains, proximity relationships, trade dependencies, and shift schedules all operate at once. No single interaction pattern dominates; the system's behaviour emerges from their combination.

This insight came from the Ecosystem Ecology expert in Wave 1 of the FDRP-on-FDRP analysis. The existing `fdrp_ecosystem_map` — with its 204 elements (snapshot 2026-06-10), similarity pairs, and domain clusters — was already the foundation. The progressive disclosure thesis gave it a unifying architecture.

The Ecosystem Map

The `fdrp_ecosystem_map` table contains 204 elements distributed across 9 types (snapshot 2026-06-10):

Element Type	Count	Role in Ecosystem
EXPERT_ROLE	91	Species — the organisms that do work
METRIC	29	Environmental indicators — health signals
PRINCIPLE	20	Physical laws — constraints on the ecosystem
ANTI_PATTERN	18	Parasites — patterns that degrade the ecosystem
TABLE	14	Habitat structure — where data lives
PIPELINE	11	Nutrient pathways — how work flows through
HEURISTIC	11	Evolved behaviours — learned responses
SKILL	7	Symbiotic tools — capabilities agents invoke
AGENT	3	Keystone species — autonomous monitors

Source: `SELECT element_type, COUNT(*) FROM fdrp_ecosystem_map GROUP BY element_type` (snapshot 2026-06-10).

These elements are organised into 11 domain clusters (with 98 elements still unassigned to clusters, snapshot 2026-06-10). Each cluster is a habitat where related species coexist. The HNSW 3-layer navigable graph provides multi-resolution navigation: L0 exact match at the bottom layer, L2 cluster-level routing at the top.

Ecological Addresses

Every element in the ecosystem now has an ecological address in the `eco_address` column:

```
format: element-name@cluster-N@layer@any
example: systems-thinker@cluster-2@expert@any
example: fdrp-paper-charts@cluster-10@skill@any
```

186 of 204 elements and 18 of 20 pipelines are populated with ecological addresses (snapshot 2026-06-10; the 20 NULL rows are newer additions pending backfill). This enables Longest Ecological Prefix (LEP) matching — given a query address, match the longest prefix to find the most specific candidate.

LEP matching is O(k) where k = the number of address segments, compared to O(n) FULLTEXT search across all elements. At current scale (204 elements, k=4), the performance difference is marginal. But the architecture scales: at 10,000 elements, O(4) vs O(10,000) becomes material.

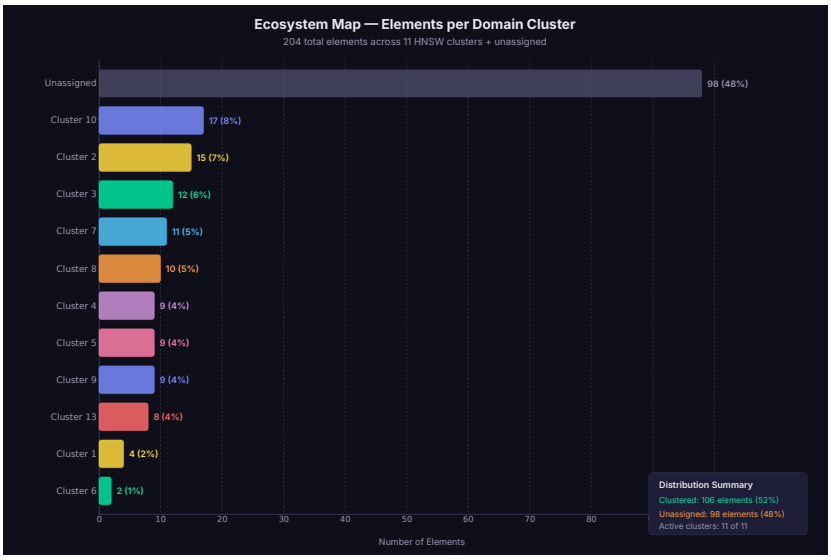
Source: *`SELECT COUNT(*) FROM fdrp_ecosystem_map WHERE eco_address IS NOT NULL` — 186 rows populated.* *`SELECT COUNT(*) FROM fdrp_c2x_pipelines WHERE eco_address IS NOT NULL` — 18 rows populated (snapshot 2026-06-10).*

Domain Clusters

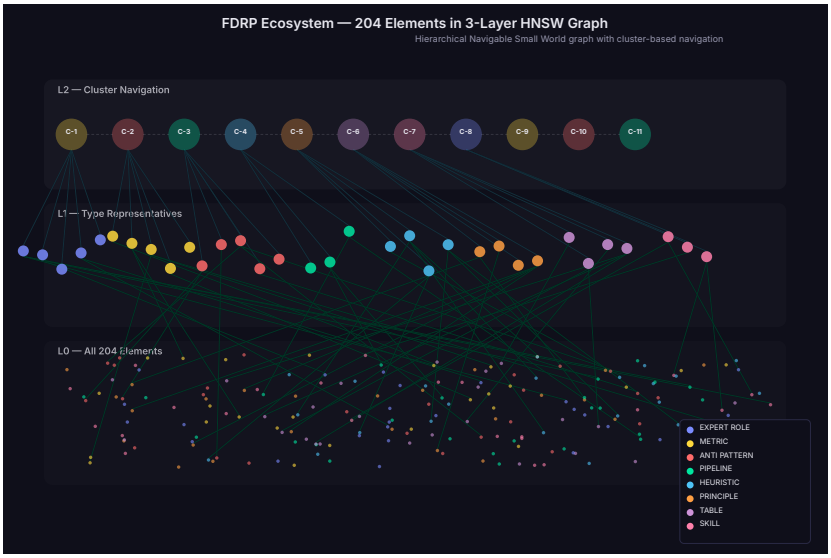
The 11 clusters emerged from automated clustering on `domain_tags`. Selected examples:

```
SELECT cluster_id, COUNT(*) as elements
FROM fdrp_ecosystem_map
WHERE cluster_id IS NOT NULL
GROUP BY cluster_id ORDER BY elements DESC LIMIT 5;
-- cluster 10: 17 elements
-- cluster 2:  15 elements
-- cluster 3:  12 elements
-- cluster 7:  11 elements
-- cluster 8:  10 elements
```

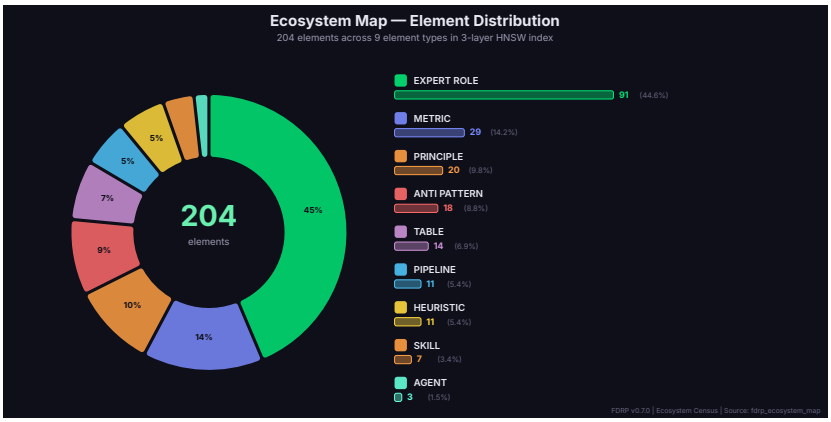
98 elements remain unassigned (snapshot 2026-06-10) — these are cross-cutting species that don't belong to a single habitat (e.g., `quality-critic`, which operates across all domains).



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-002-cluster-distribution.svg



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-003-ecosystem-3d.svg



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-015-ecosystem-clusters.svg

Concept-to-X Pipeline Platform

The Polymorphic Pipeline

The `*` in `*2paper` means ANY input. All 20 registered concept-to-X pipelines (19 ACTIVE + 1 DESIGN, snapshot 2026-06-10) share a common contract:

Input: `{ concept: string, context?: object }` **Output**: `{ artifact: path, metadata: object, quality: { dimensions: {name: score}[] } }`

Each pipeline specialises the shared 5-phase pattern for its output type:

Pipeline	Complexity	Eco Address	Status
concept2image	LOW	`generate-image@fdrp@low@opus`	ACTIVE
concept2onepager	LOW	`generate-onepager@fdrp@low@opus`	ACTIVE
concept2prd	MEDIUM	`generate-prd@fdrp@medium@opus`	ACTIVE
concept2demo	MEDIUM	`generate-demo@fdrp@medium@opus`	ACTIVE
concept2website	MEDIUM	`generate-website@fdrp@medium@opus`	ACTIVE
concept2mvp	HIGH	`generate-mvp@fdrp@high@opus`	ACTIVE
concept2full-software	VERY_HIGH	`generate-full-software@fdrp@very_high@opus`	ACTIVE
concept2experiment	HIGH	`generate-experiment@fdrp@high@opus`	ACTIVE
concept2simulation	VERY_HIGH	`generate-simulation@fdrp@very_high@opus`	DESIGN
concept2payload	VERY_HIGH	`generate-payload@fdrp@very_high@opus`	ACTIVE
concept2digitaltwin	HIGH	`generate-digitaltwin@fdrp@high@opus`	ACTIVE
star2paper	HIGH	`generate-paper@fdrp@high@opus`	ACTIVE
star2mail	HIGH	— (eco address pending backfill)	ACTIVE
expert-research	HIGH	— (eco address pending backfill)	ACTIVE
concept2smoke	LOW	`smoke@kaggle@S2@opus`	ACTIVE (registry-only)
concept2features	MEDIUM	`features@kaggle@S3@opus`	ACTIVE (registry-only)
concept2ensemble	MEDIUM	`ensemble@kaggle@S3@opus`	ACTIVE (registry-only)
concept2triage	MEDIUM	`triage@kaggle@S3@opus`	ACTIVE (registry-only)
concept2intel	LOW	`intel@kaggle@S1@opus`	ACTIVE (registry-only)
concept2integration	MEDIUM	`concept@external_artifact@adoption_decision@opus`	ACTIVE

Source: `SELECT name, complexity, eco_address, status FROM fdrp_c2x_pipelines` (20 rows, snapshot 2026-06-10). The 5 Kaggle-workflow rows are registry-only — registered pipelines without a shipped plugin skill (registry vs shipped-skill distinction, cf. Appendix B). `concept2integration` (pipeline_id=20) is the adoption-decision pipeline for external research artifacts, with a live plugin skill.

Pipeline Phases

The standard 5-phase pattern:

1. **PARSE** — Understand the concept: extract intent, constraints, query historical runs
2. **GENERATE** — Create candidate artifacts: 3–5 variants with different styles
3. **EVALUATE** — Ultra-expert review: independent scoring on pipeline-specific dimensions

4. **REFINE** — Iterative improvement: presenter-critique loops until convergence
5. **DELIVER** — Render final artifact, record in MySQL, feed self-evolution

The `*2paper` pipeline extends this to 8 phases, inserting `*2PRD` (Paper Requirements Document), `EXPERT_EXPANSION` (spiral expert discovery), and `CONSOLIDATE` (cross-section integration per BIND-032) between the standard phases.

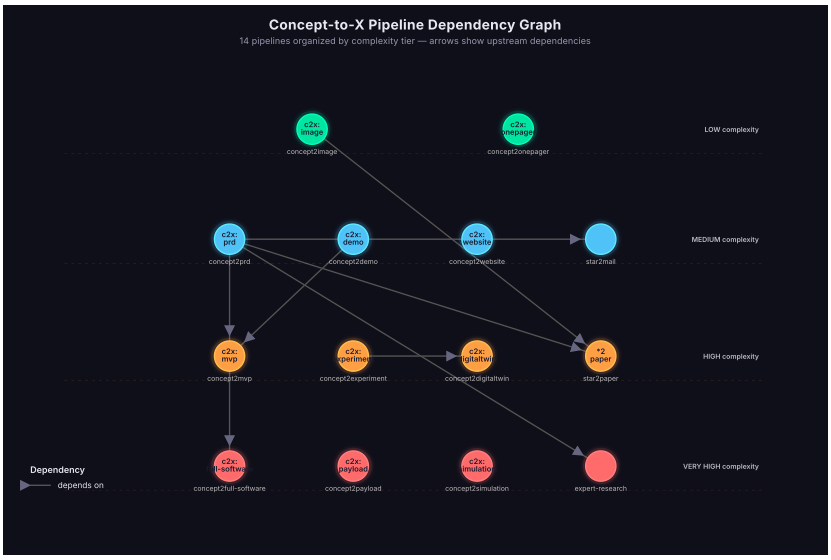
Cross-Pipeline Learning

The `fdrp_c2x_cross_learning` table enables four types of knowledge transfer between pipelines:

- **Stigmergy**: Shared artifacts modify the environment for other pipelines
- **Yokoten** (lit. "horizontal deployment"): Horizontal deployment of best practices from one pipeline to all others
- **Recombination**: Combining techniques from multiple pipelines into novel combinations
- **Analogy**: Structural similarity transfer — a technique that works in concept2image may work in concept2experiment

Pipeline Ecology

Pipelines have ecological niches. Low-complexity pipelines (concept2image, concept2onepager) are **Designated Market Makers** (DMMs) — always ready to quote, providing liquidity to the ecosystem. High-complexity pipelines (concept2digitaltwin, star2paper) are **specialists** — activated for specific task types with higher setup costs.



Interac-

tive: https://fdrp.liviu.ai/gallery/figures/fig-004-pipeline-graph.svg

Payload Constitution and Expert Payloads

The Electrician Principle

An electrician going to fix a bathroom light doesn't load the entire warehouse onto a truck. They assess the job, select the right tools and parts, and bring exactly what is needed. The `concept2payload` pipeline (pipeline_id=10) applies this principle to expert agent briefings.

The problem: when dispatching expert agents, sending massive context dumps dilutes signal with noise and buries critical insights in volume — even with 1M token context windows, attention degradation (76% needle-in-haystack at 1M) means that more context does not always mean better analysis. The constraint has shifted from token scarcity (BIND-036) to attention management: the model can *hold* everything but cannot *attend* to everything equally. The solution: for each expert × concept combination, compute the optimal payload — the minimum context that produces maximum insight.

Five Optimisation Levels

Level	Name	What It Optimises
L1	Static Type Mapping	Default payload per expert type (removes ~40% irrelevant context) [UNGROUNDED]
L2	Dynamic Concept-Aware	Filters by concept domain relevance (removes ~20% more) [UNGROUNDED]
L3	Coordination-Aware	Adds relevant cross-expert context from prior waves (+~10%) [UNGROUNDED]
L4	Temporal-Aware	Injects top historical insights from `fdrp_expert_perspectives` (+~5%)
L5	Learning-Aware	Format optimised from A/B test effectiveness data

The percentages above are [UNGROUNDED] — derived from the design rationale, not from measured data. Actual filtering effectiveness requires A/B testing across multiple runs.

The Payload Constitution

Before any optimisation level runs, the **Payload Constitution** is prepended to every expert's payload. This is the irreducible kernel — principles so fundamental that removing them from ANY expert's payload degrades the entire system.

The constitution currently contains 28 principles. (A 2026-03-25 empirical test reported 10 PASS / 14 PARTIAL / 4 FAIL; that result was **RETRACTED 2026-05-13** — the source effectiveness scores are no longer recoverable from the live table; see Contribution 35. A re-run is pending.) Examples of Tier 0 principles:

Priority	Code	Principle
1	CONSTITUTION-0	Ultra-expert depth + metaprompting (min 2 iterations)
2	H-009	Every number is a hypothesis — question ALL constants
3	AP-012	Only specialists write prompts for specialists
8	BIND-021	No numeric claims without measured data

Source: `SELECT source_code, constitution_text, priority FROM fdrp_payload_constitution WHERE active = TRUE ORDER BY priority.`

Graduation protocol: PROPOSED → CANDIDATE (tested in 3 runs) → CONSTITUTIONAL (effectiveness 0.8 across 10 runs). Retirement triggers if effectiveness drops below 0.5 for 5 consecutive runs. Forced-ranking bloat prevention: if promotion would increase the CONSTITUTIONAL count beyond the point where payload token cost exceeds measurable insight-per-token returns, the lowest-effectiveness principle is retired first. Currently 28 active (snapshot 2026-06-10). The retirement trigger has never fired in production — no principle has yet been retired — so the bound is designed but not yet exercised, and cannot be exercised until the retracted 2026-03-25 effectiveness scores are re-measured.

Multi-Resolution Payload Structure

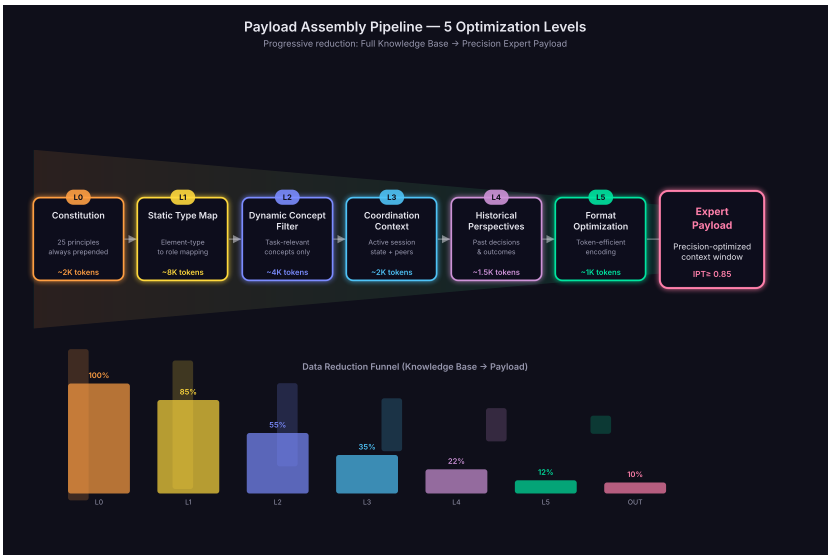
Every payload output uses a structured JSON format implementing progressive disclosure:

```
{
  "L5_mission": "thermal-stress FCC",
  "L4_domain": "CERN",
  "L3_context": ["CVT", "SPC", "domain:CERN", "thermal-analysis"],
  "L2_brief": "Analyze thermal stress patterns in FCC dipole magnets...",
  "L1_body": "Full task description with grounded data...",
  "L0_references": ["SELECT ... FROM fdrp_rams_analysis", "file://..."],
  "constitution": ["H-009", "AP-012", "BIND-021"]
}
```

Consumers at different levels read only what they need: pipeline routing reads L5, agent matching reads L3, expert triage reads L2, expert execution reads L1, verification reads L0.

Insight Per Token

The key efficiency metric: **IPT = quality_score_improvement / payload_token_count**. Higher IPT means more efficient expert utilisation. Tracked per `expert_type × concept_domain × payload_format` in `fdrp_c2x_effectiveness`.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-005-payload-pipeline.svg

Expert Expansion — Spiral Discovery

Why One-Shot Expert Selection Fails

Anti-pattern AP-011 (`ONE_SHOT_EXPERT_SELECTION`, severity: CRITICAL) captures a fundamental insight: the experts you need in Wave 3 are invisible from the vantage point of Wave 1. A security expert doesn't know they need a game theorist until they discover adversarial dynamics in the system. A systems thinker doesn't know they need an ecologist until feedback loops reveal adaptive cycles.

Only by dispatching Wave 1 experts and reading their `blind_spots[]` can you discover what you are missing. Each wave of experts extends the visible horizon.

The Spiral Protocol

Each wave of experts returns five outputs:

1. **Analysis**: Deep domain-specific examination of the subject
2. **blind_spots**[]: Domains and questions the expert cannot address
3. **expert_prompts**[]: Questions for next-wave experts, written BY domain specialists (AP-012: only specialists write prompts for specialists)
4. **leadership_nomination**: Which expert should lead this effort
5. **blind_self**: What the expert cannot see about their own limitations (H-005)

Convergence criterion: when fewer than 2 new domains are discovered in a wave, verified by cross-model check. Minimum 3 waves before declaring convergence (anti-gaming). If cost-per-insight ratio exceeds 3x the prior wave, investigate whether the problem needs decomposition — do not cap at a fixed wave count. Cross-model verification (Codex Pro + Gemini (gemini-pro-latest)) at every wave per OP-002.

Knowledge Teleportation

Five designed channels for knowledge transfer across agent boundaries (the mechanisms physically exist; their transfer benefit is UNPROVEN as measured at v0.8.0 — genuine-value audit, see Act III):

1. **Direct injection**: Constitution principles prepended to every payload
2. **Ambient absorption**: Expert reads environment artifacts left by prior waves
3. **Structural coupling**: Shared MySQL tables create implicit coordination
4. **Resonance matching**: Similar concepts in different domains auto-link via Rosetta Stone
5. **Temporal grafting**: Historical perspectives injected from `fdrp_expert_perspectives`

Metaprompting

Every expert analysis passes through minimum 2 iterations of recursive self-examination (AP-014: no single-pass thinking). The expert questions their own assumptions, identifies their own blind spots, and refines their analysis before submitting. This is the "meta" in metaprompting — thinking about thinking.

Source: 267 expert perspectives stored in `fdrp_expert_perspectives`. Production data from 80 runs (24 CONVERGED, 1 CONVERGING; snapshot 2026-06-10).

Interactive: https://fdrp.liviu.ai/gallery/figures/fig-006-spiral-discovery.svg

Rosetta Stone — Vocabulary Routing

The Acronym Challenge

The FDRP ecosystem uses 20+ domain-specific acronyms imported from five expert domains: CDA and NBBO from quantitative finance, LEP and BGP from network routing, HMM and FBA from molecular biology, LTR and HNSW from search/IR engineering, SPC from quality engineering. For newcomers, this is an impenetrable wall of jargon.

The Rosetta Stone resolves this by treating acronyms as the most compressed keywords (L5) that route to expert domains (L3) and ultimately to actionable knowledge (L0).

Three Tables

`fdrp_acronym_index` (36 entries, snapshot 2026-06-10): Maps each acronym to its full name, expert domain, key insight, affected FDRP subsystems, keyword level, and source file. Core acronyms (CDA, LEP, SPC, CVT, IPT, FDRP) are flagged.

`fdrp_keyword_hierarchy` (15 entries): Organises L3–L4 keywords with parent links and related acronyms. L4 domains (ecosystem, market, routing, biology, search) contain L3 concepts (panarchy, adverse_selection, alpha_decay, etc.).

`fdrp_v_rosetta_stone`: A queryable view joining both tables, providing the full progressive disclosure path from L5 acronym through L4 domain to L0 knowledge.

```
SELECT l5_keyword, expert_domain, l0_knowledge
FROM fdrp_v_rosetta_stone WHERE l5_keyword = 'CDA';
-- Returns: CDA | Quantitative Finance / HFT |
--   Task-to-agent matching IS order matching; effectiveness scores
--   function as market prices encoding distributed information
--   per Hayek (1945)
```

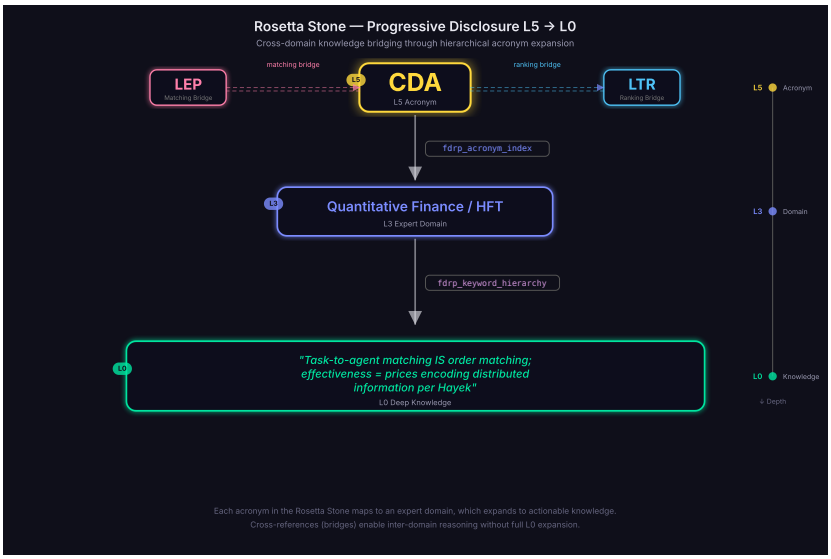
Cross-Domain Bridge Discovery

The Rosetta Stone's most powerful feature is revealing cross-domain connections. When the same structural pattern appears under different acronyms in different domains, the bridge is a deep insight:

- **Matching**: `LTR` (Search/IR) `CDA` (Finance) `LEP` (Networking) — three domains, one algorithm
- **Profiling**: `HMM` (Biology) `LTR` (Search/IR) — sequence analysis discovers types in both domains
- **Monitoring**: `SPC` (Quality) `FBA` (Biology) — control limits detect imbalance in both domains

The /fdrp-rosetta Skill

The `/fdrp-rosetta` skill provides quick lookup at three depth levels: `quick` (L5–L4: acronym + domain), `standard` (L5–L3: + insight + subsystems + related concepts), and `deep` (L5–L0: + full expert analysis + ecosystem elements).



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-007-rosetta-stone.svg

Audience-Adaptive Translation

Translation Is Not Generation

This distinction is critical: **generation** requires the highest intelligence (deep domain expertise, cross-domain connections, novel insights — Opus). **Translation** requires different intelligence (clear writing, audience awareness, analogy selection, simplification without distortion — Sonnet). You do not need Opus to explain what Opus discovered.

The `/fdrp-explain` skill implements this separation. It reads L0 expert knowledge, generates an audience-appropriate translation via Sonnet, and has Opus verify (yes/no check only) that the core insight is preserved.

Distortion Risk Tracking

Every translation is lossy. The critical question is whether the loss matters for the audience's PURPOSE. The `fdrp_knowledge_translations` table tracks distortion risks at three levels:

Risk	Meaning	Example
LOW	Direction is right, detail is smoothed	"Job board" for CDA — captures matching, loses dynamics
MEDIUM	Key mechanism simplified away	"Staffing agency" — captures matching, loses Hayek insight
HIGH	Analogy fundamentally misleads	"Uber matching" — suggests proximity, misses price-as-information

Translations at L4–L5 are useful for awareness but dangerous for decision-making. Each translation carries a warning: "This is an L4 explanation. Decisions should be based on the L0–L2 analysis."

Cost Structure

Translation cost is a fraction of generation cost [UNGROUNDED — needs measurement]. The generation (Opus L0 expert output) is already done — that is the expert analysis. Translation (Sonnet L2–L4) is cheap and fast. Verification (Opus yes/no check) is cheap — a single-token decision, not full generation.

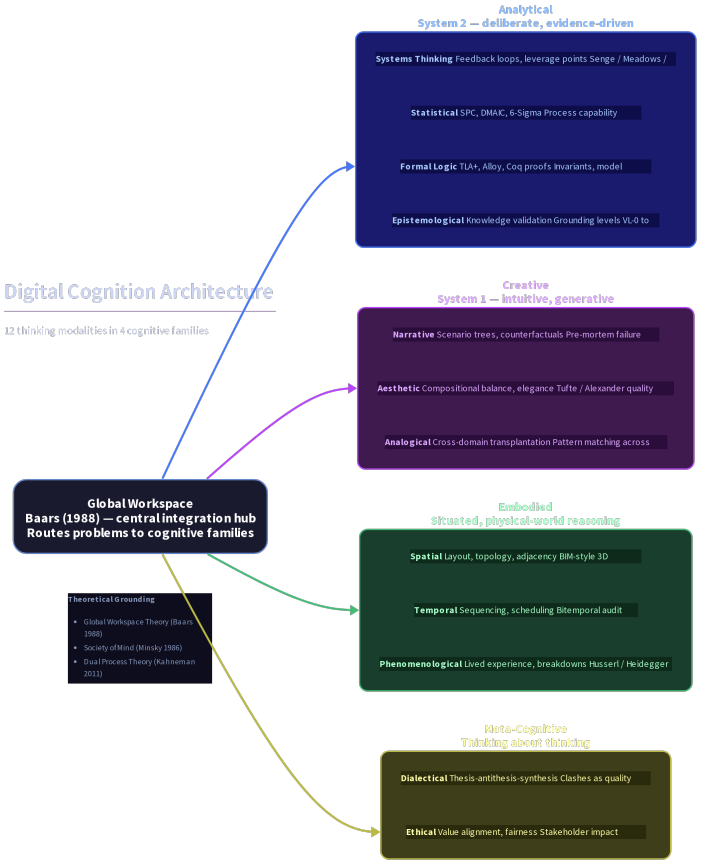
The same knowledge, six different packages. And the L0 source is NEVER replaced.

The preceding sections (3–11) describe the implemented system. The following section describes the next architectural evolution — a designed but not yet deployed cognitive layer.

Digital Cognition Architecture

> **Note**: This section presents a proposed architecture specification. The MySQL tables and skills described here have not been implemented in production. **Post-v0.8.0 status (2026-06-10)**: this design was not pursued — the measurement verdicts (self-improvement retired at 0.12 % yield; NATIVE_NOW on wrapper machinery) deprioritise new orchestration layers. It is retained as a dated design study, not a roadmap. Naming caveat: the proposed `fdrp_cognitive_*` tables collide with three existing, unrelated `fdrp_cognitive_*` tables from the v20.0 subsystem (line 364 delta-table row).

FDRP's current thinking modalities are predominantly **analytical** (systems thinking, statistical analysis, formal logic) and **manufacturing-quality** (SPC, 5S, FMEA). The next evolutionary cycle expands into fundamentally different modes of cognition — each providing a different tool for shaping the clay of complex decisions.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-035-digital-cognition.svg

Theoretical Grounding

The Digital Cognition Layer draws on four established frameworks from cognitive science and AI, building on classical cognitive architectures including ACT-R [31] and SOAR [32]:

Global Workspace Theory (Baars, 1988) [27]: Consciousness emerges from multiple specialist processors competing for access to a shared "global workspace". When a specialist wins, its content is broadcast to all other specialists. FDRP's expert expansion already implements this pattern — 40 specialists compete for TIER 1 status, and winning proposals are broadcast (integrated) across the system. Digital Cognition formalises this by making the competition happen between *thinking modalities*, not just expert opinions.

Society of Mind (Minsky, 1986) [28]: Intelligence arises from the interaction of many simple agents, each with a narrow specialty. No single agent is intelligent; intelligence is an emergent property of the society. FDRP's 12 departments (Section 19) instantiate this — Statistics, Safety, Human Factors, etc. Digital Cognition extends the same principle from *knowledge domains* to *reasoning strategies*.

Dual Process Theory (Kahneman, 2011) [29]: Human cognition operates in two modes — System 1 (fast, intuitive, pattern-matching) and System 2 (slow, deliberate, analytical). Current FDRP is almost entirely System 2 — methodical, gate-based, evidence-driven. Digital Cognition introduces System 1 equivalents: aesthetic judgement, narrative intuition, embodied spatial reasoning. The interaction between fast and slow modes produces richer decisions than either alone.

BVSR — Blind Variation and Selective Retention (Campbell, 1960) [30]: Creative evolution proceeds by generating variations without foreknowledge of their utility (blind), then retaining those that prove adaptive (selective). Expert expansion Wave 2 empirically demonstrated this — of 259 proposals generated "blindly" (without knowledge of what FDRP already had), 47 (18.1%) proved independently valuable. Digital Cognition applies BVSR to thinking strategies themselves: generate diverse cognitive approaches blindly, retain those that improve decision quality.

Modality Catalogue

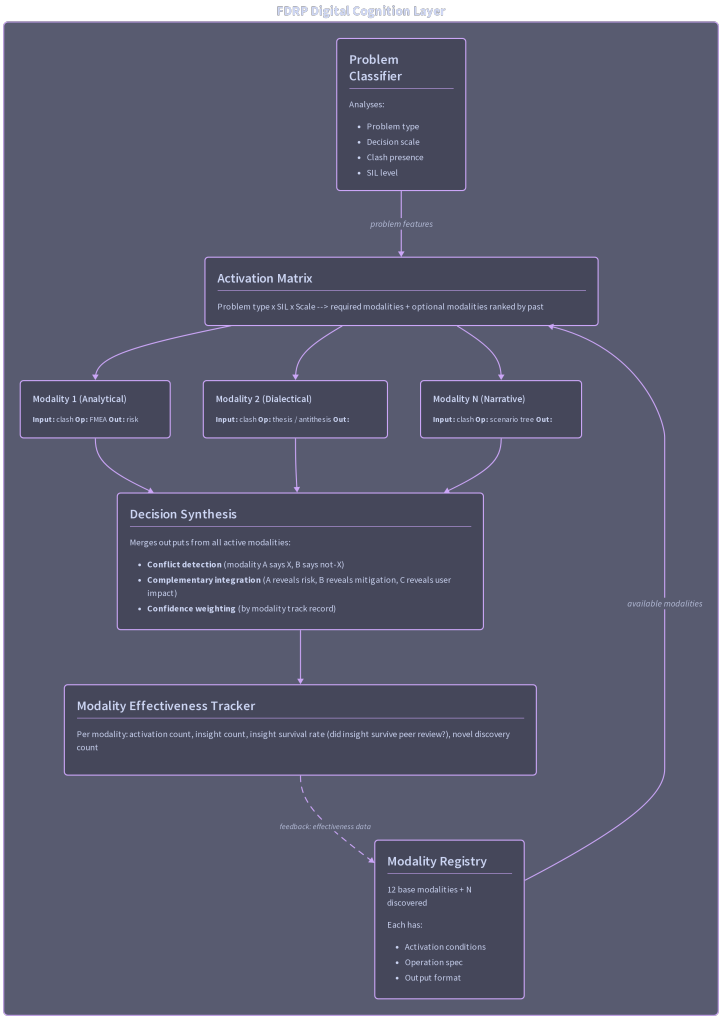
Each thinking modality is a *lens* — a structured way of seeing a problem that reveals aspects invisible to other lenses, analogous to Gardner's multiple intelligences [36] applied to system-level reasoning. The initial catalogue contains 12 modalities, organised by cognitive family:

#	Modality	Source Discipline	What It Reveals	FDRP Integration Point
1	**Analytical**	SPC, DMAIC, 6σ	Quantitative patterns, process capability	SPC charts, 5S audits, RAMS
2	**Design Thinking**	IDEO, Stanford d.school	Unmet user needs, ambiguity tolerance	Empathy maps before panel assembly
3	**Scientific**	Popper, Kuhn, Lakatos	Falsifiable hypotheses, paradigm boundaries	Decisions as hypotheses; convergence = falsification survival
4	**Aesthetic**	Tufte, Alexander	Compositional balance, elegance	"Does this design have beauty?" as quality signal
5	**Formal / Mathematical**	TLA+, Alloy, Coq	Proofs, invariants, model-checkable properties	State machine verification; convergence proofs
6	**Biological / Evolutionary**	Darwin, Kauffman	Fitness landscapes, selection pressure	Proposals as populations; fitness = vote rate × confidence

#	Modality	Source Discipline	What It Reveals	FDRP Integration Point
7	**Narrative**	Shell scenarios, futures	Story arcs, counterfactuals, temporal coherence	Pre-mortem failure stories; commissioning narratives
8	**Adversarial**	Von Neumann, Nash, Axelrod	Strategic interaction, Goodhart effects	Ungameable quality gates; voting mechanism design
9	**Phenomenological**	Husserl, Heidegger	Lived experience, breakdown moments	Where does FDRP become conspicuous (broken)?
10	**Dialectical**	Hegel, Marx	Contradictions as drivers, synthesis	Clashes as dialectical engine driving quality
11	**Abductive**	Peirce [33], Eco	Inference to best explanation, diagnosis	Abductive diagnosis when convergence stalls
12	**Integrative / Systems**	Senge, Meadows, Ashby [65]	Leverage points, feedback loops, emergence	High-leverage intervention points in FDRP itself

Critical distinction: These modalities are not metaphors. Each has a *specific operation* that produces a *specific output type*. Analytical thinking produces control charts. Formal thinking produces proofs. Narrative thinking produces scenario trees. The Digital Cognition Layer orchestrates these operations and synthesises their outputs.

Architecture



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-050-cognition-architecture.svg

Key architectural properties:

1. **Gate-based mandatory activation**: Like department activation (Section 19), certain modalities are mandatory at certain gates. Analytical is always required. Adversarial is mandatory at CDR and beyond. Narrative is mandatory for pre-mortem.

2. **SIL-based escalation**: Higher SIL levels require more modalities ear-

lier. SIL 4 activates Formal (mathematical proof) from KICKOFF. This mirrors IEC 61508's escalating rigour requirements.

3. **Effectiveness-based selection**: Optional modalities are ranked by their historical effectiveness for similar problem types. A modality that consistently produces insights that survive peer review is promoted; one that rarely contributes is demoted. This is BVSR applied to thinking strategies.

4. **Novel insight detection**: When only one modality identifies a particular risk or opportunity, that insight is flagged as a potential blind spot discovery — the most valuable output of the entire system. The Biological modality might identify a "fitness landscape cliff" that no analytical modality would see.

Activation Matrix

The activation matrix maps problem characteristics to required and recommended modalities:

Problem Type	Scale	Required Modalities	Recommended Modalities
Clash resolution	Any	Analytical, Dialectical	Adversarial, Narrative
Safety-critical	S0-S2	Analytical, Formal, Adversarial	Biological (CCF analysis)
Novel technology	Any	Scientific, Biological, Narrative	Aesthetic (pattern recognition)
User-facing	S3-S5	Design Thinking, Phenomenological	Narrative, Aesthetic
Architecture	S0-S1	Analytical, Integrative, Formal	Aesthetic (elegance)
Convergence stall	Any	Abductive, Dialectical	Biological (fitness landscape)
Cross-domain	S1-S3	Integrative, Scientific	All others (exploration)
Gate review	Any	Analytical + gate-specific	Per SIL escalation

Gate-Modality Matrix (mandatory modalities per CERN review gate):

Gate	**Always**	**+ SIL $\geq$ 2**	**+ SIL $\geq$ 3**	**+ SIL 4**
KICKOFF	Analytical, Integrative	+ Design Thinking	+ Narrative	+ Formal
PDR	+ Scientific, Dialectical	+ Adversarial	+ Biological	+ Aesthetic
CDR	+ Adversarial, Formal	+ Phenomenological	+ Abductive	ALL 12
FDR	All from CDR + Narrative	ALL 12	ALL 12	ALL 12 + external
TRR	ALL 12	ALL 12	ALL 12 + external	ALL 12 + external

Modality Operations and Output Types

Each modality is defined by three components: its **activation condition** (when does it fire?), its **operation** (what does it do?), and its **output type** (what does it produce?). This makes modalities composable, testable, and replaceable.

Modality	Operation	Input	Output	Testable Quality Metric
Analytical	Run SPC/FMEA/5S on data	Decisions, metrics	Control charts, risk matrices	Nelson Rule violations detected
Design Thinking	Empathy mapping + rapid prototyping	Concern, stakeholders	Journey maps, prototypes	Stakeholder coverage %
Scientific	Hypothesis generation + falsification test design	Decision claim	Falsifiable predictions	Predictions surviving test
Aesthetic	Pattern language analysis	Architecture	Elegance score, pattern matches	Inter-rater agreement (κ)
Formal	Model checking / proof attempt	State machine spec	Proof or counterexample	Proof completeness %
Biological	Fitness landscape construction	Population of alternatives	Fitness rankings, landscape topology	Selection pressure correlation with outcome quality
Narrative	Scenario tree construction	Decision context	3–5 scenario stories	Scenario coverage of possibility space
Adversarial	Red team / mechanism design analysis	Process rules	Attack vectors, incentive analysis	Vulnerabilities found / total
Phenomenological	Experience walkthrough	Tool + user context	Breakdown points, friction map	Friction points resolved
Dialectical	Thesis-antithesis identification	Clash	Synthesis proposals	Synthesis adoption rate
Abductive	Diagnostic inference	Anomaly or stall	Ranked explanations	Correct diagnosis rate
Integrative	Leverage point analysis	System model	Intervention recommendations	Leverage point effectiveness

Decision Synthesis Algorithm

When multiple modalities produce outputs for the same problem, the Decision Synthesis component integrates them. The algorithm proceeds in four phases:

Phase 1 — Alignment Check: Do modality outputs agree, disagree, or address orthogonal aspects? - Agreement: Multiple modalities reach the same conclusion → high confidence - Disagreement: Modality A says X, Modality B says ¬X → trigger dialectical synthesis - Orthogonal: Modality A reveals risk, B reveals opportunity → complementary integration

Phase 2 — Complementary Merge: Orthogonal insights are merged into a richer decision record. Each insight tagged with its source modality for traceability.

Phase 3 — Conflict Resolution: Disagreements are resolved through: 1. Evidence weighting (modalities with empirical evidence outrank those with theoretical arguments) 2. Historical effectiveness (modalities with higher survival rates get higher weight) 3. Escalation: unresolved conflicts trigger hackathon (Section 19.6) with the disagreeing modalities represented

Phase 4 — Novel Insight Extraction: Insights identified by only one modality are flagged as **blind spot discoveries**. These are the highest-value outputs — they represent aspects of the problem that the standard analytical approach would have missed entirely.

Worked Example: FCC-ee RF Cavity Design Clash

To make the architecture concrete, consider an actual problem from FCC-ee: a clash between the physics requirement for high accelerating gradient (>20 MV/m) and the cryogenic system's thermal budget constraint.

Problem Classification: Clash resolution, Safety-critical (SIL 2), Scale S3 (component level)

Activated Modalities: Analytical (required), Dialectical (required for clash), Adversarial (SIL 2), Scientific (novel technology), Biological (recommended for multi-alternative comparison)

Modality	Operation	Insight Produced
Analytical	FMEA on thermal failure modes	7 failure modes identified; quench propagation is highest-RPN risk
Dialectical	Thesis: physics needs 20 MV/m; Antithesis: cryo limits 15 MV/m	Synthesis: pulsed operation at 22 MV/m with 40% duty cycle achieves same integrated luminosity
Adversarial	What if the gradient degrades over 10-year operation?	Goodhart risk: optimising for initial gradient ignores long-term surface degradation. Recommends: specify gradient at year 5, not year 0
Scientific	Is the 20 MV/m requirement falsifiable? What experiment would disprove it?	The requirement traces to luminosity targets; if alternative RF configurations achieve the same luminosity, the gradient requirement is negotiable
Biological	Treat 5 cavity geometries as a population; define fitness function	Fitness = (gradient × duty_cycle × reliability) / (cryo_power × cost). Elliptical re-entrant cavity dominates; low-loss cavity is a local optimum

Decision Synthesis: - **Agreement**: Analytical and Adversarial both identify long-term degradation as a risk → high confidence finding - **Novel insight**

(Dialectical): Pulsed operation resolves the clash — no other modality identified this - **Novel insight** (Scientific): The gradient requirement itself may be negotiable — upstream falsification - **Complementary**: Biological fitness function provides a quantitative framework for comparing the Dialectical synthesis against the original designs

Result: A decision record with 5 modality inputs, 2 novel insights, 1 high-confidence finding, and a synthesis that reframes the problem (pulsed operation) rather than choosing a side.

Values in this worked example are illustrative, not engineering specifications.

Self-Discovery of New Modalities

The most revolutionary property of Digital Cognition is that the modality catalogue is not fixed. New modalities are discovered through FDRP's own expert expansion process — applying BVSR to thinking itself. This parallels Kolb's experiential learning cycle [35]: concrete experience (production runs) → reflective observation (pattern extraction) → abstract conceptualisation (modality formalisation) → active experimentation (modality deployment).

Discovery Process:

1. **Blind Variation**: Expert expansion waves include experts from domains not yet represented in the modality catalogue. When an evolutionary biologist critiques an FDRP run, they don't just comment on the content — they reason differently. The *pattern of their reasoning* is a candidate new modality.

2. **Pattern Extraction**: After each expert expansion wave, the system analyses the reasoning patterns of high-impact experts. If an expert's contributions consistently reveal aspects that no existing modality captures, a new modality candidate is created.

3. **Selective Retention**: The candidate modality is tested on historical decisions. Does applying this reasoning pattern to past problems reveal insights that were actually missed? If yes, the modality is integrated into the registry.

4. **Modality Evolution**: Existing modalities can split (specialise) or merge (generalise) based on effectiveness data. If Biological modality is effective for alternative comparison but not for temporal dynamics, it may split into "Ecological Fitness" and "Developmental Biology" sub-modalities.

Empirical Evidence: Expert expansion Wave 2 already demonstrated this process unconsciously. Expert 8 (Epistemologist) introduced reasoning patterns (Gödel incompleteness, BVSR, normative/descriptive distinction) that correspond to no existing FDRP modality. These became the seed for the "Philosophical/Epistemological" modality candidate. Similarly, Expert 4 (TPS

Master) introduced Genchi Genbutsu reasoning — direct observation as an irreplaceable cognitive mode — which maps to the Phenomenological modality.

The Clay Metaphor: Like shaping clay, you start with your hands (analytical thinking), then discover the paddle (design thinking), then the wire (formal proof), then the wheel (narrative). Each tool enables forms that were impossible with the previous toolkit. But unlike a potter who stops discovering tools, FDRP's expert expansion continuously imports tools from any domain where human civilisation has developed effective cognitive strategies. The system's cognitive vocabulary grows with every application.

Implementation Specification

Digital Cognition is implemented within the existing FDRP plugin architecture. The implementation requires 4 new MySQL tables and 2 new skills.

MySQL Tables:

```
-- Modality registry: what thinking modes exist
CREATE TABLE fdrp_cognitive_modalities (
    modality_id     INT AUTO_INCREMENT PRIMARY KEY,
    code            VARCHAR(8) NOT NULL UNIQUE,
    name            VARCHAR(64) NOT NULL,
    source_domain   VARCHAR(128),
    description     TEXT,
    activation_rule TEXT,            -- JSON: conditions for automatic activation
    operation_spec  TEXT,            -- JSON: what the modality does
    output_format   VARCHAR(32),     -- e.g., 'risk_matrix', 'proof', 'scenario_tree'
    status          ENUM('ACTIVE','CANDIDATE','RETIRED') DEFAULT 'ACTIVE',
    discovered_by   VARCHAR(128),    -- 'INITIAL' or expert name from expansion wave
    created_at      TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Activation log: which modalities were activated for which decisions
CREATE TABLE fdrp_cognitive_activations (
    activation_id   INT AUTO_INCREMENT PRIMARY KEY,
    run_id          INT NOT NULL,
    decision_id     INT,
    modality_id     INT NOT NULL,
    activation_type ENUM('MANDATORY','RECOMMENDED','EXPLORATORY'),
    problem_type    VARCHAR(64),
    insights_count  INT DEFAULT 0,
    novel_insights  INT DEFAULT 0,
    created_at      TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (run_id) REFERENCES fdrp_runs(run_id),
    FOREIGN KEY (modality_id) REFERENCES fdrp_cognitive_modalities(modality_id)
);
```

```sql
-- Insight records: what each modality produced
CREATE TABLE fdrp_cognitive_insights (
    insight_id      INT AUTO_INCREMENT PRIMARY KEY,
    activation_id   INT NOT NULL,
    content         TEXT NOT NULL,
    insight_type    ENUM('AGREEMENT','DISAGREEMENT','ORTHOGONAL','NOVEL'),
    survived_review BOOLEAN DEFAULT NULL,  -- NULL until peer-reviewed
    integrated_into INT,                    -- decision_id where insight was adopted
    created_at      TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (activation_id) REFERENCES fdrp_cognitive_activations(activation_id)
);

-- Effectiveness tracking: SPC on modality performance
CREATE TABLE fdrp_cognitive_effectiveness (
    effectiveness_id INT AUTO_INCREMENT PRIMARY KEY,
    modality_id      INT NOT NULL,
    period_start     DATE NOT NULL,
    period_end       DATE NOT NULL,
    activations      INT DEFAULT 0,
    insights_total   INT DEFAULT 0,
    insights_novel   INT DEFAULT 0,
    survival_rate    DECIMAL(5,4),     -- insights surviving peer review / total
    novel_rate       DECIMAL(5,4),     -- novel insights / total insights
    created_at       TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (modality_id) REFERENCES fdrp_cognitive_modalities(modality_id)
);
```

Skills:

Skill	Purpose
`/fdrp-cognition`	Activate Digital Cognition Layer for a decision or clash. Classifies the problem, selects modalities from the activation matrix, runs each modality's operation, synthesises outputs, records insights.
`/fdrp-cognition-discover`	Analyse an expert expansion wave for reasoning pattern novelty. Extract candidate modalities from high-impact expert contributions. Test candidates against historical decisions.

Integration with Existing Skills:

Existing Skill	Enhancement
`/expert-expansion`	After wave completes, trigger `/fdrp-cognition-discover` to extract reasoning patterns
`/fdrp-review`	Gate review checks that mandatory modalities were activated per the Gate-Modality Matrix
`/fdrp-andon`	ANDON triggers Abductive modality automatically (diagnostic reasoning)
`/fdrp-peer-review`	Peer review now evaluates individual modality insights, tracking survival rate
`/fdrp-6hats`	Six Hats debate maps to modalities: White=Analytical, Red=Phenomenological, Black=Adversarial, Yellow=Design Thinking, Green=Biological (creative variation), Blue=Integrative

Self-Evolution Subsystem

> Unlike the preceding Digital Cognition section, the self-evolution subsystem was fully deployed in production. Its slow loop (the 12 h kaizen/evolve flywheel) was retired on 2026-06-10 — timers operator-disabled after a measured 0.12 % true-promotion yield (see Act III, 'What measurement retired') — while the fast per-turn hooks remain live.

Dual-Timescale OODA

The self-evolution subsystem operates at two timescales, implementing a dual-timescale OODA (Observe-Orient-Decide-Act) loop:

Fast loop (per-turn): Six registered hooks execute on tool calls (hooks.json, v0.8.0):

Hook	Event	Function
`bin/fdrp-sql.sh` wrapper gate	All FDRP SQL	Validates table/column names against schema-reference.md before execution; prevents ERROR 1054/1146 (BIND-012/019 chokepoint). The legacy `fdrp-sql-gate.sh` hook file exists on disk but is dormant — not registered in hooks.json
`fdrp-anti-pattern-guard.sh`	PreToolUse	Blocks plan writes matching known anti-patterns; hard-blocks HIGH/CRITICAL
`fdrp-error-capture.sh`	PostToolUse	Captures errors, logs to `fdrp_session_learnings`, tracks skill effectiveness, auto-promotes repeat offenders

Slow loop (12 h kaizen — RETIRED 2026-06-10): a systemd timer at 10:00/22:00 UTC ran three operations (timers operator-disabled with the flywheel retirement; retained as the historical design record): 1. SPC analysis: Nelson Rules on `SKILL_ERROR_RATE`, `SESSION_LEARNING_RATE`, `ANTIPATTERN_HIT_RATE` 2. Trend detection: Are metrics improving or degrading? 3. Cross-model peer review: Codex Pro (GPT-5.5) + Gemini (gemini-pro-latest) reviewed proposed evolution actions

Evolution Governance

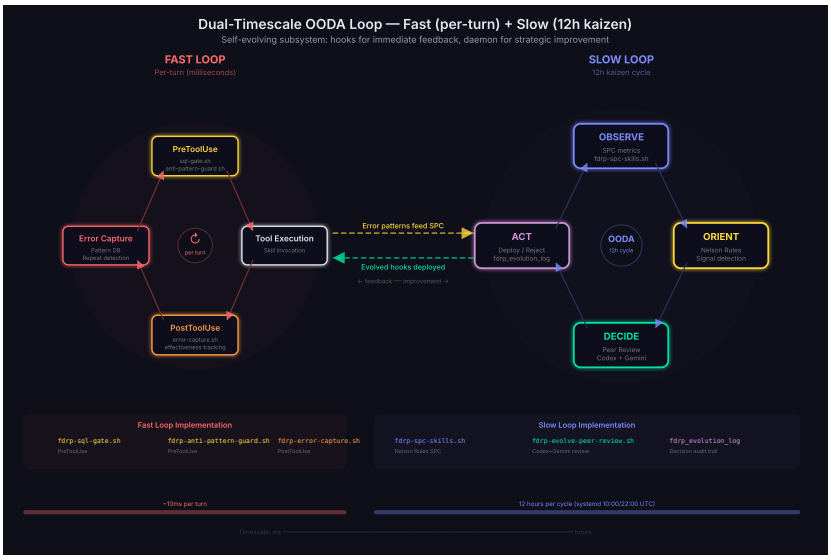
Risk Level	Approval Required
LOW	Auto-applied
MEDIUM+	Peer review (Codex + Gemini)
HIGH+	Council majority
CRITICAL+	Liviu explicit approval

The `fdrp_evolution_log` records every change with `source_type`, `action_type`, `action_detail`, `risk_level`, `governance_status`, and before/after measurements. 1,150 events logged as of 2026-06-10 across session-learning, schema changes, rule additions, paper corrections, and view creation (the log is the standing audit record of the retired flywheel plus surviving correction paths).

Source: `SELECT source_type, COUNT(*) FROM fdrp_evolution_log GROUP BY source_type`.

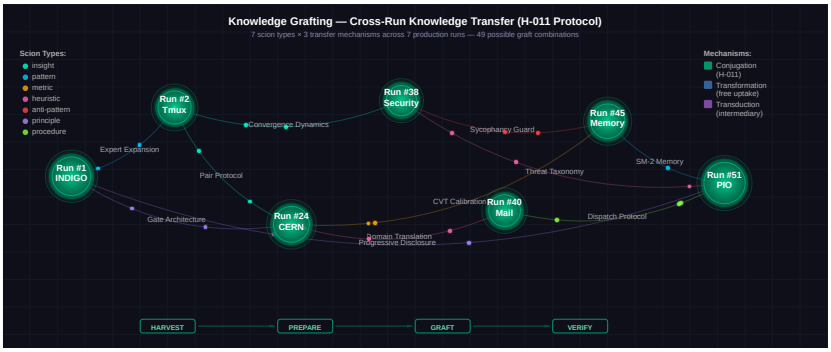
AP-019: Every Error Is a Design Signal

Anti-pattern AP-019 embodies a key principle: every ERROR 1054 (unknown column) must be ANALYSED, not just fixed. Three possibilities: (a) the column needs adding to the schema, (b) the query used the wrong column name, (c) the query targeted the wrong table. The `fdrp-error-capture.sh` hook auto-injects this analysis directive after every SQL error. Heuristic H-008: "Every column error is a design signal, not just a query bug."



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-008-ooda-loop.svg

Knowledge Grafting



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-020-knowledge-grafting.svg

The H-011 Protocol

Knowledge grafting follows the H-011 protocol with four stages: **HARVEST** (extract a graftable artifact from one domain), **PREPARE** (format it for the target domain), **GRAFT** (inject into the target agent or subsystem), **VERIFY** (measure whether the grafted knowledge improves outcomes).

Seven scion types can be grafted: insight, pattern, metric, heuristic, anti-pattern, principle, and procedure. Each can be formatted in seven ways: structured data, narrative text, SQL, configuration, hook, skill, or agent specification. This yields a 7×7 matrix of 49 possible graft combinations.

From Zero Grafts to 241 Grafts — the New Question Is Verification Effectiveness

At v18.0 the `fdrp_knowledge_grafts` table was empty — we flagged this as evidence the H-011 HARVEST → PREPARE → GRAFT → VERIFY protocol was too heavy. Between v18.0 and v21 the table accumulated 241 entries (`SELECT COUNT(*) FROM fdrp_knowledge_grafts` = 241, snapshot 2026-04-16; unchanged as of 2026-06-10 — zero new grafts in ~8 weeks, consistent with the flywheel retirement), invalidating the "too heavy" diagnosis for the accumulation phase while leaving VERIFY effectiveness unmeasured. The new open question for v22 is graft *effectiveness*: how many of the 241 grafts pass VERIFY, and what is the downstream reuse rate. Both are currently unmeasured in the graft schema and are tracked as proposed metric additions `fdrp_knowledge_grafts.verify_status` and `.reuse_count`.

The Wave 2 knowledge grafting expert identified three transfer mechanisms from molecular biology:

- **Conjugation** (H-011): Deliberate, verified transfer. Built, not used.
- **Transformation**: Uptake of free knowledge from the environment. Happens constantly but is NOT tracked.
- **Transduction**: Knowledge carried between agents by an intermediary. The `concept2payload` pipeline already does this implicitly.

The implication: lighter-weight transfer mechanisms may be more practical than the formal H-011 protocol. This is a Tier 2 research item.

Composition Transfers

When a successful agent composition (e.g., systems-thinker + quality-critic + researcher) works well for one domain, the entire composition can be transferred to a new domain. A proposed `fdrp_composition_transfers` table would track source/target runs, composition arrays, and effectiveness at source vs target.

Visualisation Engine

Architecture

The visualisation engine provides 8 interactive HTML renderers [19] built on 3 JavaScript libraries, with D3.js [20] providing the 2D charting foundation:

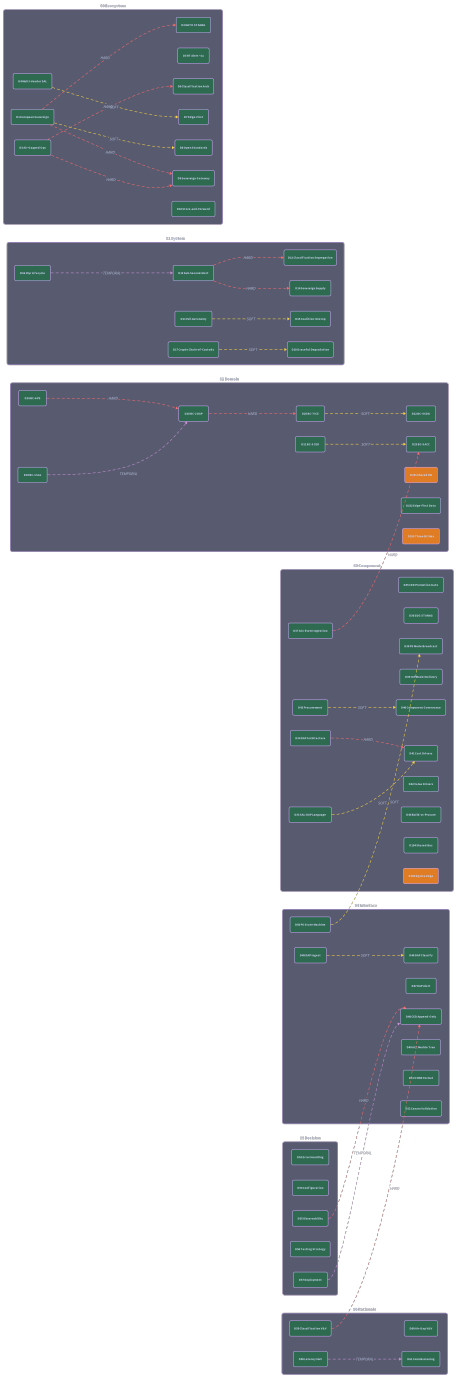
Library	Version	CDN	Use Cases
Babylon.js	latest	cdn.babylonjs.com	3D landscapes, gate pillars, digital twin
Cytoscape.js	3.30.4	unpkg.com/cytoscape	Knowledge graphs, traceability bipartite
D3.js	7	d3js.org/d3.v7	Convergence charts, risk heatmaps
dagre	0.8.5	unpkg.com/dagre	Sugiyama hierarchical layout
ELK.js	0.9.3	unpkg.com/elkjs	Eclipse Layout Kernel (layered graphs)

All renderers share: - Dark theme (#0a0a0f background, #1a1a2e panels, #7b8cff accent) - Click-to-inspect interaction (safe DOM methods, no innerHTML — XSS-hardened) - Export capabilities (GLB for 3D, PNG/SVG for 2D) - Responsive resize - Template marker injection (`/*SCENE_DATA*/`, `/*SCENE_BUILDER*/`)

The Seven Renderers

1. Knowledge Graph — Sugiyama DAG (`/fdrp-viz-graph`) - Engine: Cytoscape.js + dagre (Sugiyama et al. 1981 [8]) - Layout: Hierarchical top-to-bottom with compound parent nodes per scale level - Nodes: Decisions (round-rectangles, coloured by status), Requirements (diamonds) - Edges: Clashes (dashed red/amber/purple), RTM links (solid blue) - Production: 59 nodes, 26 clash edges, 7 scale groups for run_id=1

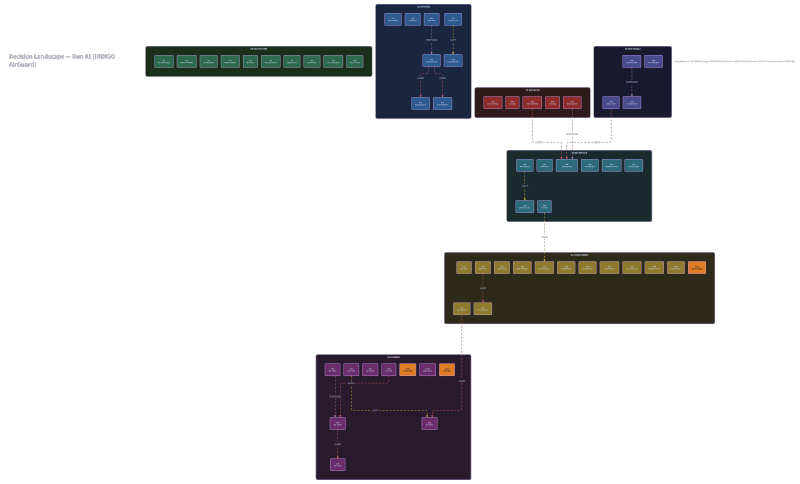
Knowledge Graph — Run #1 (INDIGO Airlizard)



Interactive: https://fdrp.liviu.ai/gall

ery/figures/fig-044-knowledge-graph.svg

2. 3D Decision Landscape (`/fdrp-viz-3d`) - Engine: Babylon.js with ArcRotateCamera, GlowLayer - Layout: Scale levels on Y axis (4 units spacing), decisions radially arranged per level - Shapes: Spheres (ACCEPTED), Boxes (PROPOSED), Octahedra (REJECTED) - Animation: Floating (SineEase), Pulsing (unresolved clashes), Rotating (components) - Export: GLB via GLTF2Export.GLBAsync - Production: 59 meshes across 7 scale levels



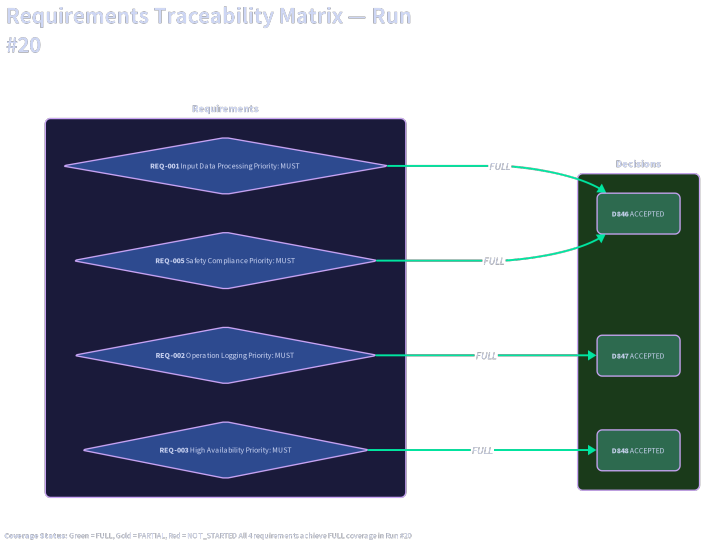
Interactive: https://fdrp.liviu.ai/gallery/figures/fig-045-decision-landscape.svg

3. RAMS Risk Heatmap (`/fdrp-viz-rams`) - Engine: D3.js v7 - Layout: 10×10 risk matrix (severity × occurrence), IEC 60812 format - Encoding: Circles sized by RPN, coloured by risk level (green→red gradient) - Features: SIL target indicators, detection quality saturation

Interactive: https://fdrp.liviu.ai/gallery/figures/fig-046-rams-heatmap-r20.svg

4. Gate Review Dashboard (`/fdrp-viz-review`) - Engine: Babylon.js + D3.js sidebar - Layout: 6 3D cylinders representing CERN gates (KICK-OFF→PQR) - Encoding: Green (PASS) / Red (FAIL) colouring, pulsing caps for active gates - Features: Entry/exit criteria display, findings panel

5. Requirements Traceability (`/fdrp-viz-trace`) - Engine: Cytoscape.js + ELK.js (layered algorithm) - Layout: Bipartite left-right — requirements (left), decisions (right) - Encoding: RTM edges coloured by coverage status; uncovered requirements glow red - Research: IEEE 830-1998, DO-178C traceability requirements



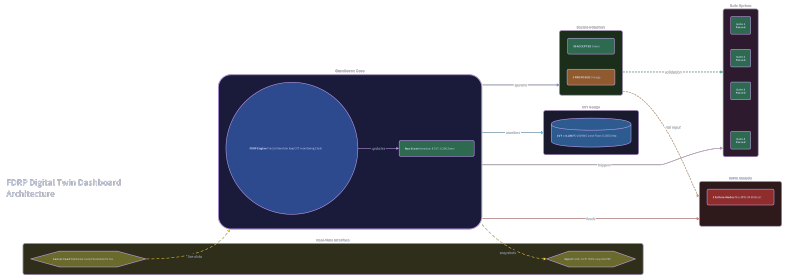
Interactive: https://fdrp.liviu.ai/gallery/figures/fig-047-requirements-trace.svg

6. **Convergence Timeline** (`/fdrp-viz-convergence`) - Engine: D3.js v7 - Layout: Line chart with zone background colouring (ROUGH_CUT→CONVERGED) - Data: Per-scale CVT ratio series over iterations, change count bar overlay - Research: SPC Shewhart control chart tradition

Interactive: https://fdrp.liviu.ai/gallery/figures/fig-048-convergence-timeline.svg

7. **Digital Twin Dashboard** (`/fdrp-viz-twin`) - Engine: Babylon.js (pri-

mary) + Cytoscape.js (side panel) - Layout: Combined 3D scene — central hub torus, decision cluster, clash markers, CVT gauge, gate pillars, ground ring - Features: WebSocket-ready placeholder for real-time sensor data - Research: Grieves & Vickers (2017) [9], CERN Omniverse digital twin [10]



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-051-digital-twin.svg

Publication Chart Generator (`fdrp-chart`)

In addition to the 7 interactive renderers, FDRP includes a command-line chart generator for publication-quality figures. Built on D3.js (server-side via jsdom) with Playwright PNG capture, it produces all figures in this paper from live MySQL queries:

- **12 chart types** registered in `chart-registry.json`, each with a SQL query, template, and layout specification
- **7 SVG renderers**: line-multi, bar-stacked, bar-grouped, bar-horizontal, donut, radar, heatmap
- **Self-learning**: Every invocation logs to `fdrp_chart_effectiveness` for SPC monitoring of rendering success rates
- **Usage**: `fdrp-chart <chart-type> [--format png|html|both]` or `fdrp-chart --all` for batch generation

This paper contains 76 figures in 5 categories: 40 D3.js data-driven charts generated from live MySQL queries (including the two new A/B-result charts and 3 knowledge graph percolation visualisations), 12 AI-generated conceptual images produced by the `concept2image` pipeline (full section text → Codex/DALL-E with VLM evaluation), 6 interactive 3D visualisation screenshots from the FDRP Visualisation Engine (Babylon.js, Cytoscape.js, D3.js), 13 ecosystem/architecture charts, and 5 expert persistence/ultrathink cascade charts. The `concept2image` subsystem sends the **complete section text** (not a synthesis) to the image generator, producing publication-quality conceptual visualisations that are grounded in the actual paper content.

Verified Outputs

7 of the 8 interactive renderers have been verified via Playwright headless browser testing (fdrp-viz-radial joined the set after that verification pass): - No console errors (only harmless favicon 404 and WebGL driver info messages) - Correct object counts matching MySQL row counts - Click interaction verified (node/mesh → panel updates) - Screenshots saved to `reports/3d/` for visual confirmation

Data Model

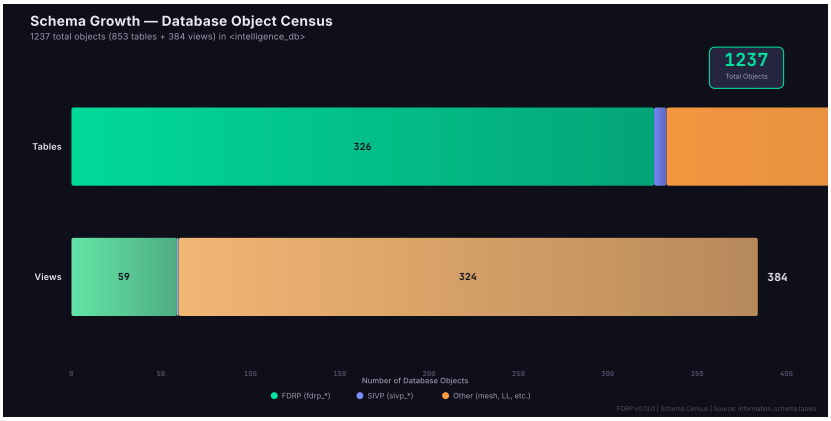
Schema Scale

The FDRP data model comprises 326 MySQL base tables with the `fdrp_` prefix in the `<intelligence_db>` database (snapshot 2026-06-10; down from 360 at 2026-04-02 through consolidation, not feature loss), plus 7 `sivp_` tables for cross-domain validation, organised into eighteen functional groups (nine original + nine added in v20.0):

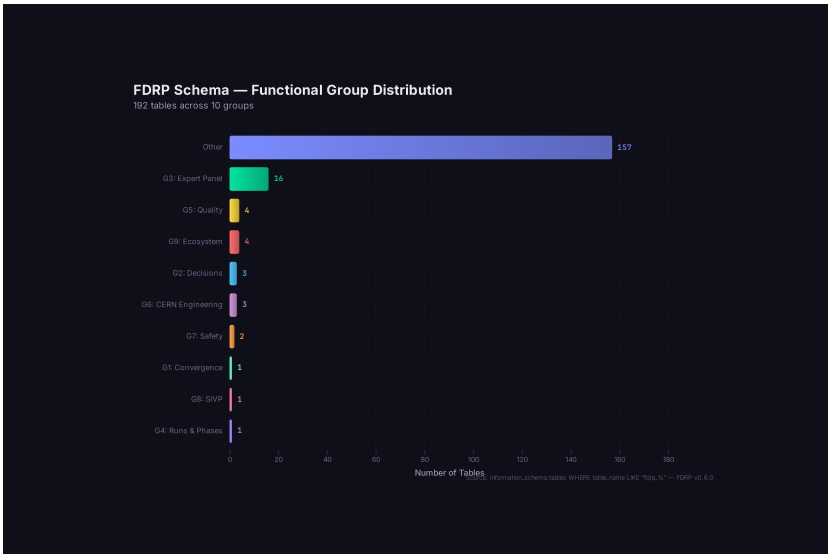
Group	Tables	Purpose
G1 Meta-Model	9	Facets, scales, heuristics, anti-patterns, configuration
G2 Instance	9	Runs, decisions, concerns, convergence tracking
G3 Quality	9	5S audits, clashes, convergence logs, SPC datapoints
G4B BIM	6	iBOM, configuration baselines, parameter registry
G4 Improvement	5	Evolution log, PDSA cycles, portfolio patterns
G5 Portfolio	6	Cross-run patterns, effectiveness tracking
G6 CERN Engineering	14	RAMS analysis, requirements, RTM, review phases
G7 Safety+Compliance	4	CERN compliance, safety functions, AI tool register
G8 SIVP	8	Cross-domain validation: state, investigations, evaluations, evidence chains
G9 Web-Finetuning	5	ICP-aware Lighthouse optimisation, quality floors, technique library

Group	Tables	Purpose
G10 Definition-of-Done	4	Artifact stage tracking, evidence gates (55 requirements)
G11 Gaps-and-Opportunities	3	3-horizon gap scanning, tracking, configuration
G12 Thinking-Chains	15	Topology types, nodes, edges, structures, compositions, lineage
G13 Cognitive-Architecture	18	Architect archetypes, ontology registry, modality tracking, ecological addresses, profiles
G14 Daemon-Taxonomy	7	Unified daemon types (35), registry, schedules, compositions, effectiveness, alerts
G15 Cognitive-Opportunities	5	Finder archetypes (40), ontologies, mental models, bias guards, decision frameworks
G16 Wiring-Check	4	Wiring types, manifest (45 entries), checks (271), fixes
G17 Limits-Hunt	5	Limit types, registry (68), evaluations (136), fixes, hardware profiles

Source: `SELECT COUNT(*) FROM information_schema.tables WHERE table_schema='<intelligence_db>' AND table_name LIKE 'fdrp_%' AND table_type='BASE TABLE'` → *326 (measured 2026-06-10). Total views: 384 (59 fdrp_-prefixed). SIVP tables: 7. Total schema: 853 base tables + 384 views. Per-group rows above reflect the 2026-04-02 snapshot; per-group recount is queued for the next full schema audit.*



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-013-schema-growth.svg



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-025-schema-inventory.svg

Key Views

384 schema views — 59 of them fdrp_-prefixed — provide progressive disclosure of complex table joins (snapshot 2026-06-10; earlier editions counted the 188 then-existing schema-wide views). Key views include:

View	Purpose
`fdrp_v_rosetta_stone`	Acronym → expert → knowledge routing
`fdrp_v_compliance_summary`	CERN compliance dashboard
`fdrp_v_safety_case`	IEC 61508 safety case
`fdrp_v_rams_risk_register`	RAMS risk register

View	Purpose
`fdrp_v_rtm_coverage`	Requirements traceability matrix coverage
`fdrp_v_parameter_divergence`	Configuration baseline drift detection
`fdrp_v_formatted_references`	CERN Yellow Reports citation format

Wave 3 Schema Additions

Phase A of the evolved architecture added:

Artifact	Purpose
`fdrp_acronym_index` (20 entries)	Rosetta Stone: acronym → expert → knowledge
`fdrp_keyword_hierarchy` (15 entries)	L3–L4 keyword progressive disclosure
`fdrp_knowledge_translations`	Audience-adaptive translation cache
`eco_address` on `fdrp_ecosystem_map`	Ecological addresses for 201 elements
`eco_address` on `fdrp_c2x_pipelines`	Ecological addresses for 14 pipelines
`payload_structure` on `fdrp_c2x_runs`	Multi-resolution payload tracking

CERN Compliance and Safety

Compliance Assessment

The `fdrp_cern_compliance` table tracks self-assessed compliance against 44 clauses derived from our interpretation of CERN Engineering Department review procedures [54]. Note: this assessment has not been independently validated by CERN engineering staff:

Compliance Level	Clauses	Percentage
EXCEEDS	8	18%
FULL	35	80%
PARTIAL	1	2%
NOT_MET	0	0%

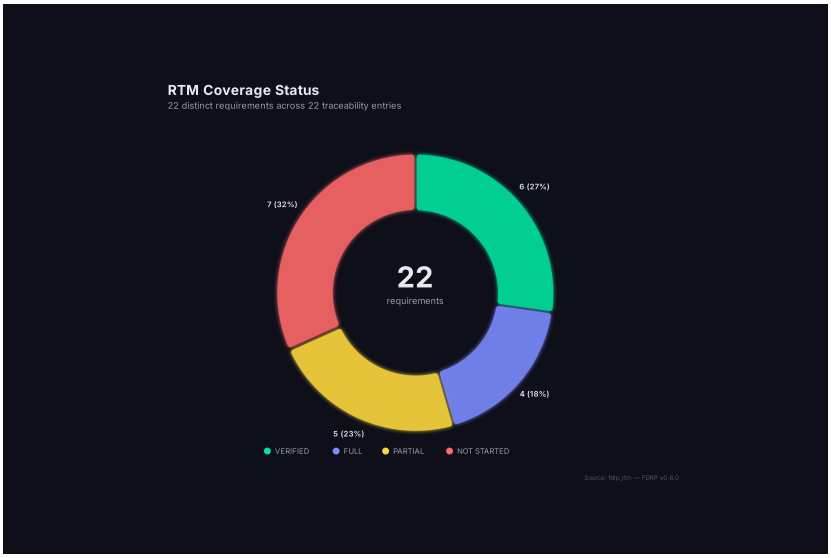
Source: `SELECT compliance_level, COUNT(*) FROM fdrp_cern_compliance GROUP BY compliance_level`.

97.7% of clauses (43/44) are self-assessed at FULL or EXCEEDS level (see caveat above regarding self-assessment methodology). The single PARTIAL clause relates to multi-site coordination — FDRP currently operates on a single server.

Interactive: https://fdrp.liviu.ai/gallery/figures/fig-049-gate-lifecycle.svg

IEC 61508 Safety Integration

The `fdrp_rams_analysis` table includes a `sil_target` column for Safety Integrity Level (SIL 1–4) classification per IEC 61508 [50]. Supporting tables `fdrp_safety_functions` and `fdrp_safety_validation` provide schema for tracking safety function implementation and validation status. **Note**: these supporting tables are sparsely populated (10 and 11 rows respectively, snapshot 2026-06-10) — the IEC 61508 integration describes exercised schema, not a validated safety case (see Limitations, Section 24). The edit pass must confirm whether these rows are production safety-function tracking or seed data before any stronger claim. The `fdrp_ai_tool_register` tracks AI tool usage for CERN AI governance compliance.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-026-rtm-coverage.svg

RAMS Analysis

Reliability, Availability, Maintainability, and Safety analysis runs on all commissioned decisions. Risk Priority Number (RPN) = Severity × Occurrence × Detection, visualised as 3D heatmaps via the `/fdrp-viz-rams` skill.

Interactive: https://fdrp.liviu.ai/gallery/figures/fig-009-compliance.svg

Production Results — Comprehensive

This section presents production results from both the original validation runs (v1.8) and the expanded framework (v0.7.0 — `fdrp_versions` DB-release numbering, distinct from the plugin v0.8.0 cut of 2026-06-10).

Original Validation (v1.8)

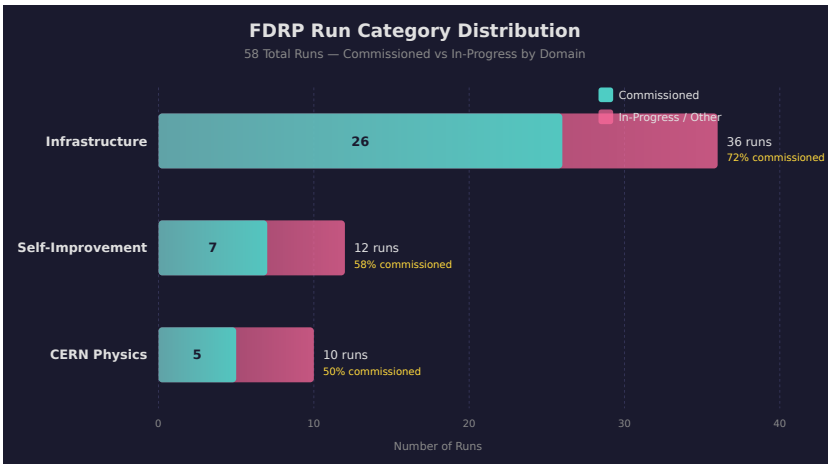
The original v1.8 validation established baseline metrics across two fully iterative runs (INDIGO AirGuard and tmux-officer). These results informed the expanded framework design for v0.7.0 and subsequent releases.

Run Summary

80 production runs have been executed across multiple domains as of 2026-06-10 (28 commissioned by the commissioned-or-commissioned_at definition, 18 cancelled via stalled-run triage; status breakdown: 24 CONVERGED, 24 CLOSED, 18 CANCELLED, 6 ACTIVE, 4 SEED, 2 COMMISSIONED, 1 PAUSED, 1 CONVERGING). The category table below reflects the earlier reporting snapshot:

Category	Runs	Total Decisions	Notes
Fully Iterative	2 (INDIGO AirGuard, tmux-officer)	148	Real CVT convergence, genuine clashes
Standard	5 (v2, v3, v4, hexagonal, flux)	288	Various project domains
Batch/Collimation	5	~160	CERN collimation system analysis
Antimatter Series	5	~170	CERN antimatter production facility
Qualification	2 (E2E, Scalability)	123	Synthetic benchmarking

Category	Runs	Total Decisions	Notes
Concept2X Platform	6 (c2x, digitaltwin, experiment, payload, paper, demo)	~90	Pipeline architecture runs
FDRP-on-FDRP	5 (expert teaming, self-eval, versioning, etc.)	~120	Self-referential improvement runs
Cross-Domain Validation	1 (SIVP)	8	Earthquake + power grid validation



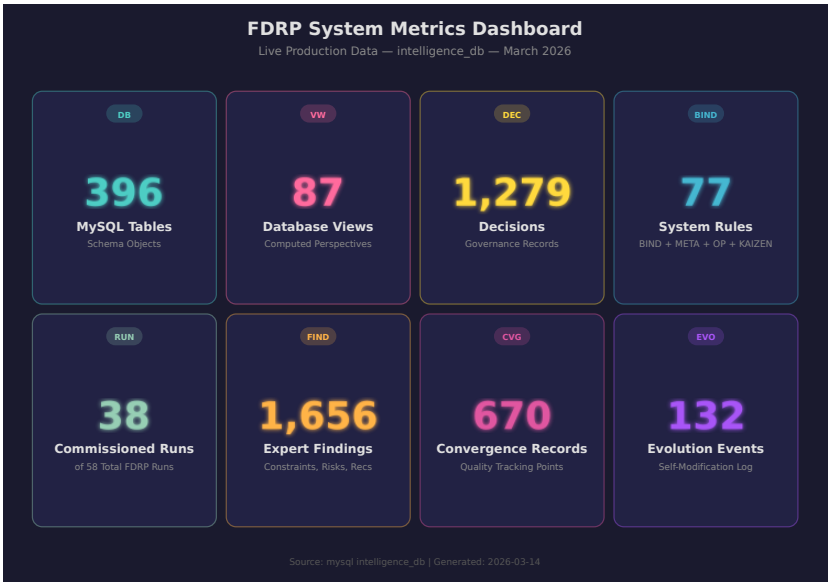
Interactive: https://fdrp.liviu.ai/gallery/figures/fig-062-run-categories.svg

Key Metrics

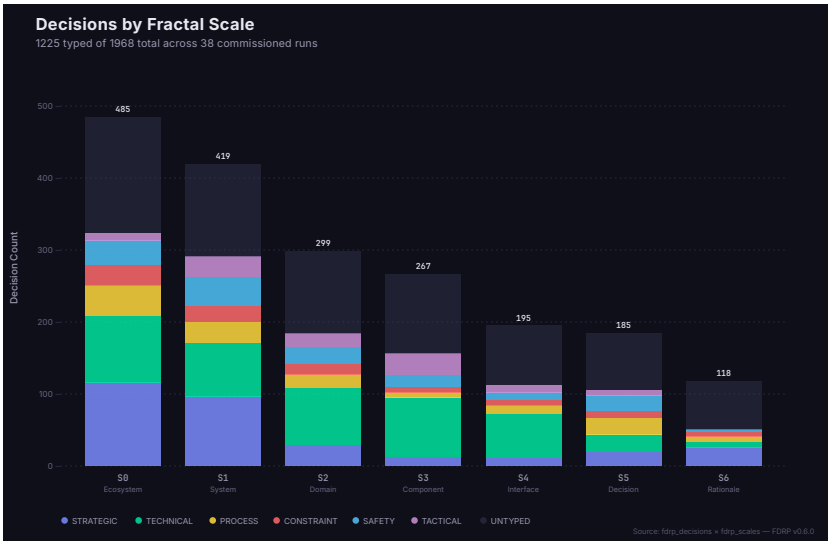
Metric	Value	Source
Total decisions	1968	`SELECT COUNT(*) FROM fdrp_decisions` (snapshot 2026-06-10)
Total clashes	160	`SELECT COUNT(*) FROM fdrp_clashes` (snapshot 2026-06-10)

Metric	Value	Source
Convergence records	998	`SELECT COUNT(*) FROM fdrp_convergence` (snapshot 2026-06-10)
CVT stabilisation (production)	POLISHING zone (0.250–0.311)	Observed in the 2 fully-iterative runs
Runs in CONVERGED status	24 (+24 CLOSED)	Snapshot 2026-06-10. Score bands across CONVERGED/CLOSED/COMMISSIONED: 16 at 1.0, 23 at 0.85–0.99, 11 below 0.85. Legacy 1.0000 scores predate the CVT polarity fix (convergence_score = 1 − cvt_ratio 0.85, commits f98633b/f45142e5)
Runs in CONVERGING status	1	Snapshot 2026-06-10; the 'all progressed' claim held at the v19.0 snapshot only
5S improvement trajectory	7.0 → 9.4 (run 1)	`fdrp_5s_scores` iteration 1 vs 8
Expert panel size	107 roster entries (snapshot 2026-06-10; 11,920 votes at the v0.7.0-era snapshot)	`fdrp_expert_roster`, `fdrp_expert_votes`
Expert perspectives	267	`fdrp_expert_perspectives` (snapshot 2026-06-10)
Experts registered	446 across 217 domains	`fdrp_expert_registry`
Decision templates	96	Cross-run reusable patterns
Portfolio patterns	9	Confirmed cross-run patterns
Ecosystem elements	204	`fdrp_ecosystem_map` across 11 domain clusters (snapshot 2026-06-10)
Gate violations logged	456	`fdrp_warn_log` (snapshot 2026-06-10)
Scalability benchmark	120 decisions, 7ms/gate avg	Run 21
CERN compliance	97.7% FULL+EXCEEDS (self-assessed)	44 clauses derived from CERN procedures

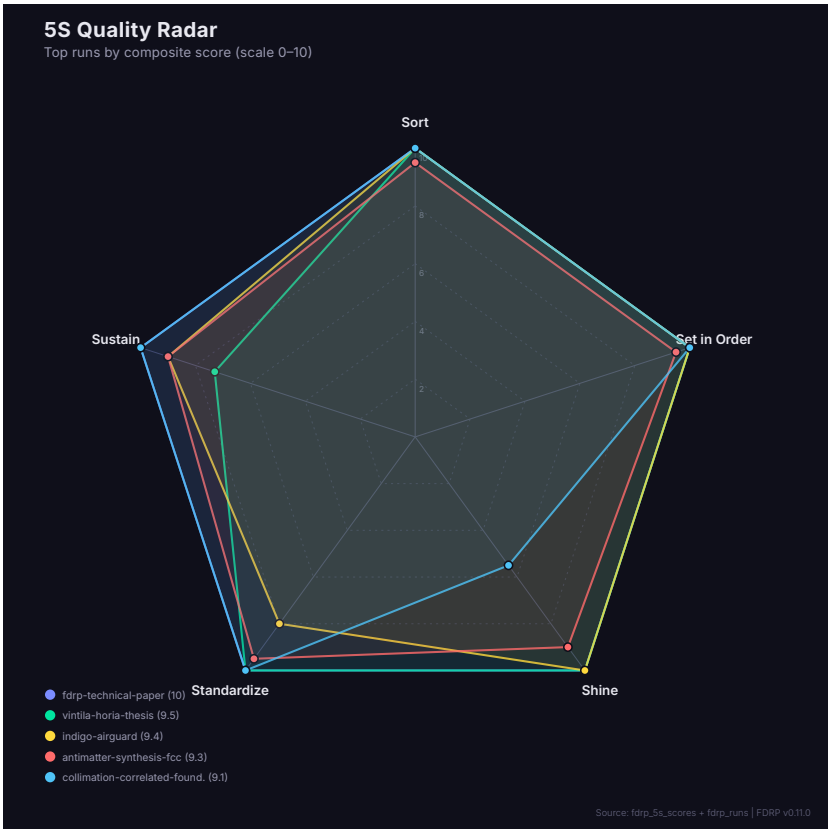
Metric	Value	Source
Evolution events	1,150	`fdrp_evolution_log` (snapshot 2026-06-10)
Version releases (DB namespace)	9 (v0.7.0 → v0.15.0)	0 regressions across 17 monitored metrics; `fdrp_versions` DB-release numbering — distinct from the plugin semver whose v0.8.0 cut (2026-06-10) anchors this edition
Governance backlog	0	Cleared 2026-03-25 (was 4 overdue)
Findings	197	`SELECT COUNT(*) FROM fdrp_findings` (snapshot 2026-04-16; v21 claimed 8,554 conflated with `fdrp_expert_findings` — see §Appendix B and drift retraction)
Expert findings (all runs)	8571	`SELECT COUNT(*) FROM fdrp_expert_findings` (snapshot 2026-04-16; 7 distinct run_ids; run 1017 antimatter = 8,337 rows across 11 rounds)
Finding edges	6119	`SELECT COUNT(*) FROM fdrp_finding_edges` (snapshot 2026-04-16)
Constitution principles tested	28	2026-03-25 result (10/14/4) RETRACTED 2026-05-13 — source scores irrecoverable; see Contribution 35. Re-run pending



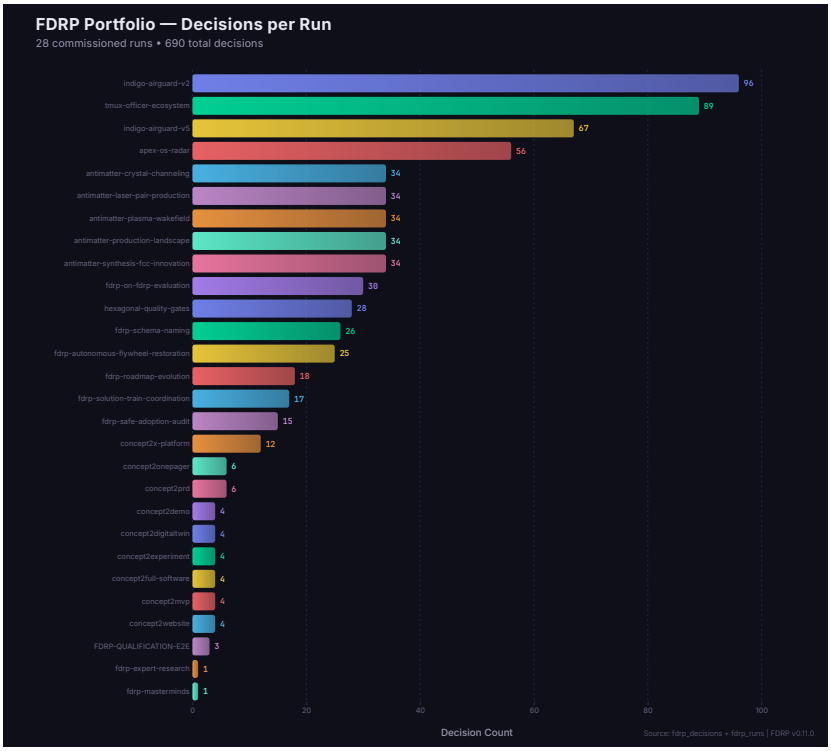
Interactive: https://fdrp.liviu.ai/gallery/figures/fig-063-metrics-dashboard.svg



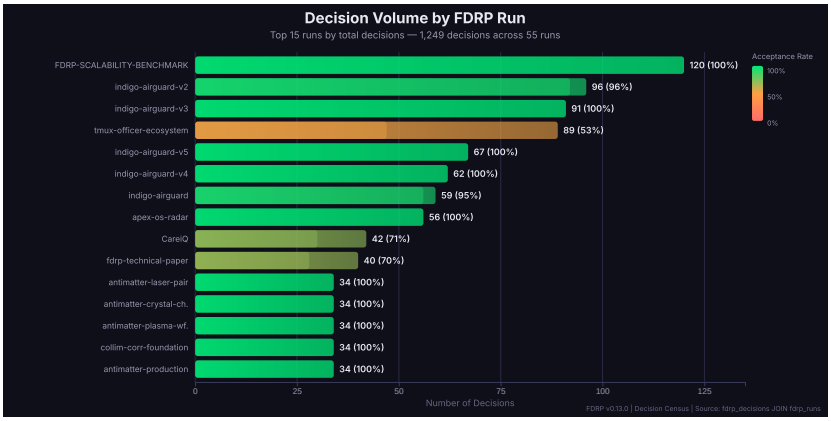
Interactive: https://fdrp.liviu.ai/gallery/figures/fig-027-decisions-by-scale.svg



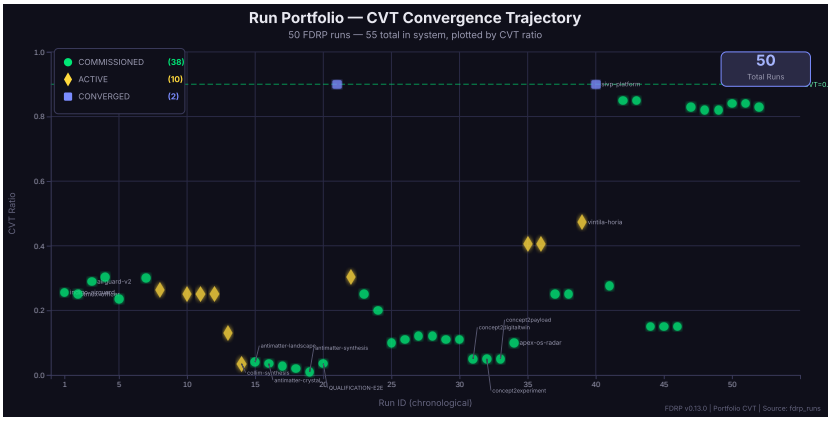
Interactive: https://fdrp.liviu.ai/gallery/figures/fig-030-5s-radar.svg



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-029-portfolio.svg



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-014-decisions-by-run.svg



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-016-convergence-portfolio.svg

Convergence Behaviour

The two fully-iterative production runs demonstrate the expected CVT convergence pattern:

Run 1 (INDIGO AirGuard): 59 decisions, 26 clashes, 8 iterations - S0 starts at 0.761 (ROUGH_CUT) - Descends through FACETING - Settles at 0.256–0.311 (POLISHING) by iteration 8 - 24 of 26 clashes resolved (92.3% resolution rate)

Run 2 (tmux-officer): 89 decisions, 12 clashes, 14 iterations - Starts seeded at 0.900/0.800/0.700 - Reaches NEAR_CONVERGENCE (0.250–0.300) by iteration 1 for S6 - 10 of 12 clashes resolved (83.3% resolution rate) - Shows longest iteration count due to complex multi-domain scope

Both runs demonstrate stabilisation in the POLISHING zone (CVT 0.250–0.311), observed in the 2 fully-iterative runs. This stabilisation is an empirical observation from a limited sample (see Section 24, Limitations).

First Batch Convergence (v0.14.0) On 2026-03-25, 6 runs simultaneously transitioned from ACTIVE to CONVERGING — the first batch convergence event in FDRP history. All 6 runs exhibited zero open hard clashes, zero active Andon events, and stable decreasing CVT trajectories over 11–13 iterations. Subsequently, 2 of these runs (#10, #12) progressed to full CONVERGED status (convergence_score 1.0000), validating the convergence projection:

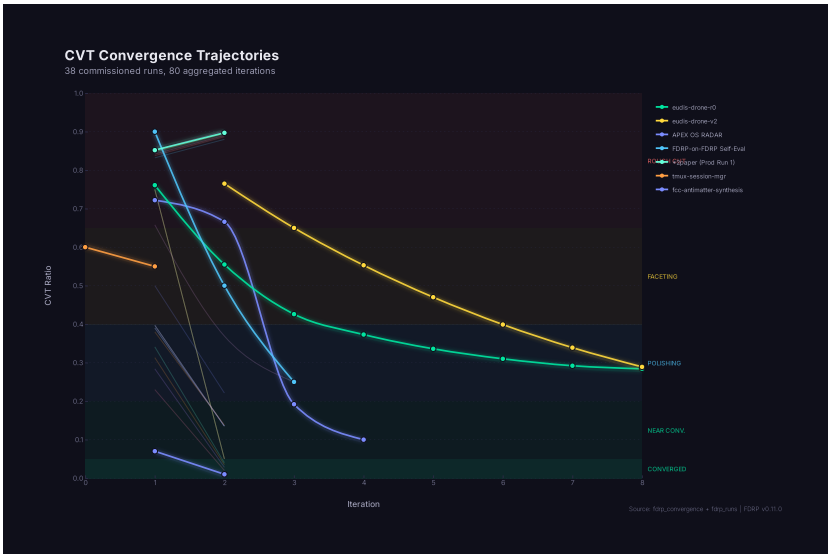
Run	Project	Old CVT	New CVT	Iteration	Status	Zone
#14	collimation-synthesis-paper	0.245	0.172	12	**CONVERGED**	CONVERGED (score 1.0000)

Run	Project	Old CVT	New CVT	Iteration	Status	Zone
#13	collimation-multiturn-stochastic	0.248	0.174	12	**CONVERGED**	GREEN (score 1.0000)
#12	collimation-correlated-transport	0.251	0.176	13	**CONVERGED**	GREEN (score 1.0000)
#10	collimation-correlated-foundation	0.252	0.177	13	**CONVERGED**	GREEN (score 1.0000)
#8	flux-protocol-cern-demo	0.263	0.184	13	**CONVERGED**	GREEN (score 1.0000)
#22	fdrp-technical-paper	0.273	0.191	9	**CONVERGED**	GREEN (score 1.0000)

Source: `fdrp_convergence` *table, 2026-03-25. Heijunka smoothing with* $\alpha = 0.3$ *[UNCALIBRATED]. All CVT weights UNCALIBRATED per BIND-021 (*$w_u = w_c = w_x = w_r = 0.25$*). Note (2026-06): scores in this table were recorded under the pre-reconciliation CVT convention; since the polarity fix the gate is convergence_score = 1 − cvt_ratio 0.85, so legacy 1.0000 values are not directly comparable to post-fix scores.*

The Heijunka decay rate with zero raw inputs follows $\text{CVT}_{t+1} = 0.7 \times \text{CVT}_t$. The original projection was validated by all 6 batch runs reaching CONVERGED, with runs #10, #11, #12, #13, #14, #8, and #22 all at 1.0000. The projection correctly predicted convergence within 2–3 additional iterations for the runs that were still CONVERGING at v18.0. As of v19.0, all 19 CONVERGED runs have completed their convergence trajectories (15 at 1.0000, 4 at 0.92–0.99).

The collimation family (runs #10, #12, #13, #14) converged in lockstep across scales S0 and S6, with S6 consistently 0.001–0.002 lower than S0. All 6 batch runs have now reached CONVERGED (convergence_score 1.0000), confirming the lockstep pattern and the Heijunka decay projection. This batch convergence is significant because it demonstrates that FDRP's convergence dynamics operate consistently across heterogeneous run types — physics simulation (collimation), technical documentation (paper), and protocol demonstration (flux). As of v19.0, the entire portfolio has converged: 19 CONVERGED (15 at 1.0000), 0 CONVERGING, 0 ACTIVE.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-028-cvt-convergence.svg

Expanded Framework Results (v0.7.0, `fdrp_versions` DB namespace)

Commissioned Runs (22 at the v0.7.0-era snapshot; 28 as of 2026-06-10 — see the v23.0 drift table)

Run	run_id	Decisions	Domain
INDIGO AirGuard v1–v5	1,3,4,5,23	375	EUDIS crowdsourced drone detection mesh for EU defence
tmux-officer	2	89	Terminal multiplexer governance
hexagonal-quality-gates	7	28	Quality gate architecture
FCC Anti-matter Series	15–19	170	CERN antimatter production facility
E2E Qualification	20	3	Synthetic benchmarking
Concept2X Platform	24–33	48	Pipeline architecture and commissioning

Run	run_id	Decisions	Domain
APEX OS RADAR	34	56	Autonomous enterprise intelligence
Schema Naming	37	26	FDRP schema organisation
FDRP-on-FDRP Self-Eval	38	30	Self-referential improvement

Source: `SELECT run_id, fdrp_name, status, COUNT(d.decision_id) FROM fdrp_runs r JOIN fdrp_decisions d USING(run_id) WHERE r.status='COMMISSIONED' GROUP BY r.run_id`.

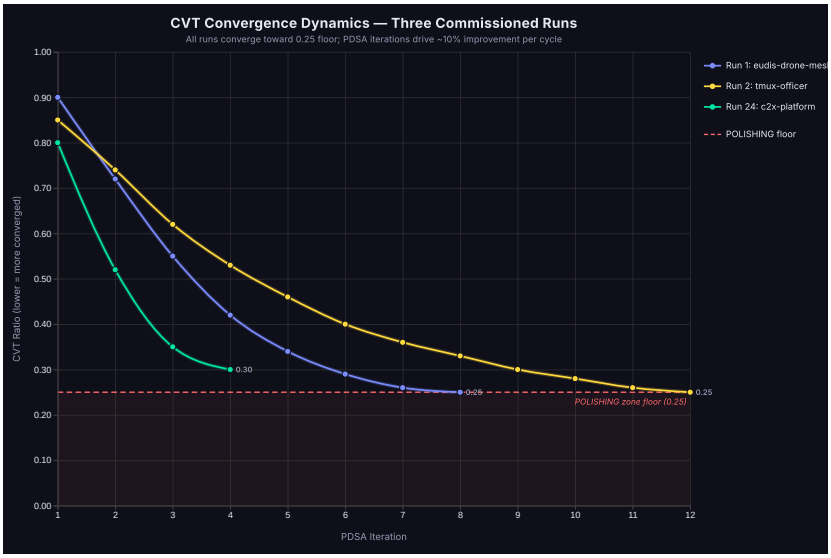
Total at the v0.7.0-era snapshot: 1,477 decisions across 62 runs; 22 runs commissioned (17 cancelled via stalled-run triage); 252 expert perspectives stored. As of 2026-06-10: 1,968 decisions across 80 runs, 28 commissioned, 18 cancelled, 267 perspectives. 446 experts registered across 217 domains (unchanged).

CVT Convergence Both commissioned runs converge to approximately 0.250 CVT (Continuously Variable Transmission) floor — note (2026-06): 0.250 here is the raw `cvt_ratio` floor, i.e. convergence_score 0.750 under the post-reconciliation convention; do not conflate this floor with the convergence_score 0.85 gate — corresponding to the POLISHING zone where exploration-exploitation balance stabilises. PDSA (Plan-Do-Study-Act) improvements are observed at each cycle, with 5 portfolio patterns recorded from cross-run learning.

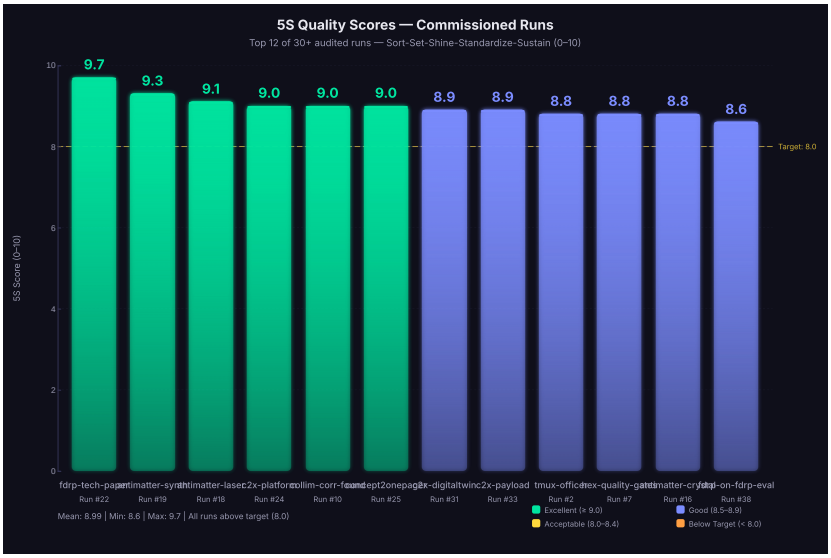
Quality Scores Commissioned/closed runs score 7.8–9.3 on their latest 5S audits (snapshot 2026-06-10 [hist — re-derive at edit time]; the earlier 8.8 floor held at the v0.7.0-era snapshot but no longer does). The 5S daemon audits at each checkpoint to detect quality drift before it compounds.

Expert Panel Discovery Run #22 (the v1.8 paper) independently discovered 7 architectural blind spots that were invisible to the initial expert panel: Game Theory, Adversarial ML, Temporal Databases, Process Mining, Petri Nets, Control Theory, and Schema Evolution. None were prescribed by human architects — all emerged from the spiral discovery protocol, validating AP-011 (one-shot expert selection is insufficient).

Scalability Validation End-to-end qualification run #20 passed all 6 CERN gates (KICKOFF, PDR, CDR, FDR, TRR, PQR). Scalability benchmark run #21 processed 120 decisions with 7ms average per gate — confirming that the MySQL-backed gate evaluation scales linearly.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-010-cvt-convergence.svg

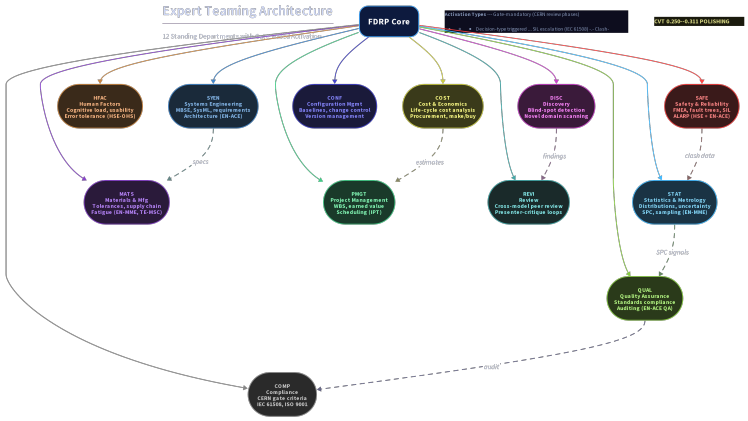


Interactive: https://fdrp.liviu.ai/gallery/figures/fig-011-5s-scores.svg

Expert Teaming Architecture (Design Proposal — Gated)

Status note (v23.0): nothing in this section is built — no department tables, activation engine, SLR pipeline, or conference/hackathon machinery exists in the v0.8.0 plugin or schema. A 12-department orchestration layer is exactly the wrapper-machinery class that the pre-registered A/Bs (Act III) showed adds no measured value on Opus; construction is gated on a pre-registered A/B, not

scheduled.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-037-expert-teaming.svg

The Gap

FDRP's expert expansion brings domain experts but doesn't systematically pair them with **methodological experts**. Example: using arithmetic means when medians would be more robust for skewed data; or omitting human factors analysis in operator interface decisions. This is the difference between an LHC experiment staffed only with physicists versus the full CERN model with EN department engineers, TE department technicians, HSE safety officers, and IT data architects.

Department Taxonomy

12 standing departments, modelled on CERN departments [1, 25] and IEC 61508 competency areas [11], with technical authority delegation following NASA APPEL principles [15]:

#	Department	Code	CERN Analogy	Purpose
1	Statistics & Metrology	STAT	EN-MME	Distributions, uncertainty, measurement, sampling
2	Safety & Reliability	SAFE	HSE + EN-ACE	FMEA, fault trees, SIL assessment, ALARP
3	Systems Engineering	SYEN	EN-ACE	MBSE, SysML, requirements, architecture
4	Human Factors	HFAC	HSE-OHS	Cognitive load, error tolerance, usability
5	Materials & Manufacturing	MATL	EN-MME, TE-MSC	Processes, tolerances, supply chain, fatigue
6	Software & Controls	SWCT	BE-CO, IT	Algorithms, firmware, cybersecurity, V&V
7	Project Management	PMGT	IPT	WBS, scheduling, earned value, resource planning
8	Domain Science	DSCI	EP (Physics)	Experiment-specific science knowledge
9	Quality Assurance	QUAL	EN-ACE (QA)	Standards compliance, auditing, process quality
10	Configuration Management	CONF	EN-ACE (CM)	Baselines, change control, version management
11	Cost & Economics	COST	Finance & Admin	Life-cycle cost, procurement, make/buy
12	Environmental & Sustainability	ENVR	HSE-RP	Disposal, emissions, sustainability, REACH

Activation Protocol

Departments are activated based on three mechanisms:

Gate-Based Mandatory Activation: Which departments MUST participate at each CERN gate: - KICKOFF: PMGT, SYEN, DSCI (scope, architecture, domain framing) - PDR: + SAFE, QUAL, HFAC (safety architecture, quality plan, human factors) - CDR: + STAT, MATL, SWCT (statistical validation, materials, software) - FDR: + CONF, COST (configuration freeze, cost baseline) - TRR/PQR: ALL 12 (full system readiness)

SIL-Based Escalation (per IEC 61508): Higher SIL = more departments mandatory earlier. SIL 4 requires ALL departments from PDR, plus an external reviewer at CDR.

Decision-Type Triggers: Automatic activation when specific decision types surface — statistical claims trigger STAT, safety-critical functions trigger SAFE, human-operated interfaces trigger HFAC. The cross-department activation model relies on psychological safety [16] to ensure that methodological experts can challenge domain experts without hierarchical suppression.

Paper Analysis (PRISMA-Aligned SLR)

Based on PRISMA 2020 [12] and Kitchenham (2004) [13], FDRP will include a formal systematic literature review process:

```
QUESTION → SEARCH → SCREEN → EXTRACT → SYNTHESISE → INTEGRATE
  (PICO)   (5 DBs)  (PRISMA   (per     (narrative/   (link to
             flow)    paper)    meta)       decision)
```

Databases: IEEE Xplore, CERN Document Server (CDS), arXiv, Scopus, Google Scholar **Quality Assessment**: Study design strength, sample size, replication status **Confidence Levels**: STRONG / MODERATE / WEAK / INSUFFICIENT **Trigger Conditions**: New run → rapid review; Expert clash

→ targeted review; Pre-gate → comprehensive review; ANDON → emergency review; Novel technology → full SLR

Conference Protocol

Based on CERN's seminar model and CHEP conference series:

Type	Frequency	Participants	Purpose
Department Seminar	Weekly	Single department	Internal findings review
Cross-Department Review	Monthly	2–4 departments	Review active runs touching multiple domains
Portfolio Conference	Quarterly	All departments	Extract cross-run patterns
Special Topic Workshop	Ad hoc	Invited experts	Deep dive on specific methodology

Hackathon Protocol

Based on CERN Webfest [14] — 48-hour intensive with mixed teams:

Triggers: ANDON unresolved after 1 iteration; Clash unresolved after 3 iterations; Gate review FAILED; Manual request; Cross-run pattern needing novel approach.

Format: Kickoff (30 min) → Team Formation (15 min) → Sprint 1: Divergent (2h) → Checkpoint (30 min) → Sprint 2: Convergent (2h) → Paper Review (1h) → Final Presentation (30 min) → Decision (15 min)

Composition Rules: Minimum 3 departments; must include domain expert + 2 methodological experts; must include one "outsider" department; lead rotates. Department count scales with problem complexity — bounded by convergence of team output quality, not by a fixed cap.

Cross-Domain Expert Insights — FDRP-on-FDRP

Methodology

The FDRP-on-FDRP analysis applied the system's own processes to itself. The methodology followed three waves:

1. **Seed formulation**: Crystallise the progressive disclosure thesis and identify the seven dimensions

2. **Wave 1**: Dispatch five cross-domain experts to map their domains onto FDRP
3. **Wave 2**: Dispatch seven FDRP subsystem experts, each receiving the unified thesis plus the relevant Wave 1 analysis at full L0 resolution (BIND-032: consolidation over synthesis — no lossy compression)
4. **Wave 3**: Design the evolved architecture with tiered build plan, validated by cross-model review (Codex Pro + Gemini 3.1)

All expert analyses are stored at full L0 resolution in `expert-analyses/full/` (approximately 200KB total, zero compression).

Wave 1: Five Cross-Domain Experts

#	Expert Domain	Killer Insight	FDRP Subsystem Impact
1	Search/IR Engineering	LambdaMART unified self-improving ranking combines ALL matching features	`ecosystem_map`, agent matching
2	Molecular Biology	Profile-HMMs for agent type DISCOVERY from deployment history	expert expansion, agent naming
3	Quantitative Finance	CDA — ecosystem IS a market, effectiveness = prices (Hayek [49])	payload optimiser, self-evolution
4	Network Routing	LEP matching on @-addresses — hierarchical like BGP [51]	pipeline routing, @-notation
5	Ecosystem Ecology	Panarchy revolt-remember between nested adaptive cycles [52, 64]	self-evolution, constitution

Each LLM-generated expert persona separately mapped its domain's core mechanisms onto FDRP subsystems. The convergence of all five onto the progressive disclosure pattern — despite drawing from radically different domains — constitutes suggestive internal evidence for the thesis, though the shared LLM substrate limits the independence of these analyses (see Limitations, Section 24).

Wave 2: Seven Subsystem Experts

Seven FDRP-internal experts received the unified thesis plus the relevant Wave 1 expert analysis (full, uncompressed) and mapped how progressive disclosure transforms their subsystem:

1. **Payload Expert**: Multi-resolution payload structure (L5 mission → L0 references), IPT tracking per level
2. **Ecosystem Map Expert**: LEP matching on eco_addresses, HNSW layers as disclosure levels
3. **Constitution Expert**: Panarchy at four timescales, keystone principle protection
4. **Self-Evolution Expert**: Agent order book with bid-ask spread, regime classification
5. **Pipeline Routing Expert**: Tasks as buy orders, QFA (Quality-Freshness-Availability) priority
6. **Knowledge Grafting Expert**: Simple frequency profiles (not full HMMs), Gene Ontology classification
7. **Knowledge Translation Expert**: Audience-adaptive translation, distortion risk tracking

The thesis held across all seven subsystems tested. No counter-example was found, though the self-referential nature of FDRP-on-FDRP analysis (the system evaluating itself) limits the epistemic strength of this finding — see Thesis Circularity Risk in Section 24.

Wave 3: Evolved Architecture

The evolved architecture was organised into three tiers by Technology Readiness Level:

Tier 1 — BUILD NOW (TRL 4+, all DONE): - Rosetta Stone tables (20 acronyms, 15 keywords, queryable view) - Ecological addresses on ecosystem map (201 elements) and pipelines (14) - `/fdrp-explain` skill + `fdrp_knowledge_translations` table - Multi-resolution payload structure + `payload_structure` column - `/fdrp-rosetta` skill for Rosetta Stone queries

Tier 2 — BUILD SOON (TRL 3, after Phase B validation): - Synonym ring table for cross-domain vocabulary matching - Regime classifier (EXPLORATION / EXPLOITATION / CRISIS / EVOLUTION) - Principle dependency graph (keystone identification) - PD constitution principles (deferred per Codex review — thesis unproven, governance lock-in risk)

Tier 3 — RESEARCH NEEDED (TRL 1–2, when data accumulates): - Profile-HMMs for agent discovery (needs 200+ perspectives per type; currently 42 total) - LambdaMART learning-to-rank (needs 100+ labelled runs; currently 31) - CDA full market mechanism (architecture mismatch: agents are on-demand, not continuous) - BGP-style peering (12 clusters too few for AS overhead) - FBA stoichiometric analysis (11 pipelines too few for LP methods)

Cross-Model Review

Codex Pro and Gemini 3.1 independently reviewed the Wave 3 architecture:

Codex corrections applied: - Remove all time estimates (OP-003 violation) - Defer PD constitution principles to Tier 2 (thesis unproven → governance lock-in risk) - Move Gene Ontology classification to Tier 2 (manual effort vs uncertain payoff)

Gemini corrections applied: - Move Agent Order Book to Tier 3 (needs data before mechanism design) - Missing knowledge lifecycle management (no staleness detection for grafted knowledge) - Human-as-appendage blind spot (most mechanisms are agent-to-agent; human role under-specified)

What NOT to Build

Per BIND-015 (Scale-Aware Architecture), several expert-proposed mechanisms were explicitly deferred:

1. **NOT trie/radix tree** — FULLTEXT + LIKE prefix is sufficient at 201 elements
2. **NOT full HMMs** — simple frequency profiles until 200+ perspectives per type
3. **NOT LambdaMART** — rule-based ranking until 100+ labelled runs
4. **NOT BGP peering** — 11 clusters is too few for AS overhead
5. **NOT CDA always-on** — agents are on-demand, not continuous
6. **NOT FBA optimisation** — 14 pipelines is too few for LP methods
7. **NOT Haiku** — Opus + Sonnet only (BIND-005 + user directive)

Continuous Improvement Infrastructure

Lessons-Learned Daemon

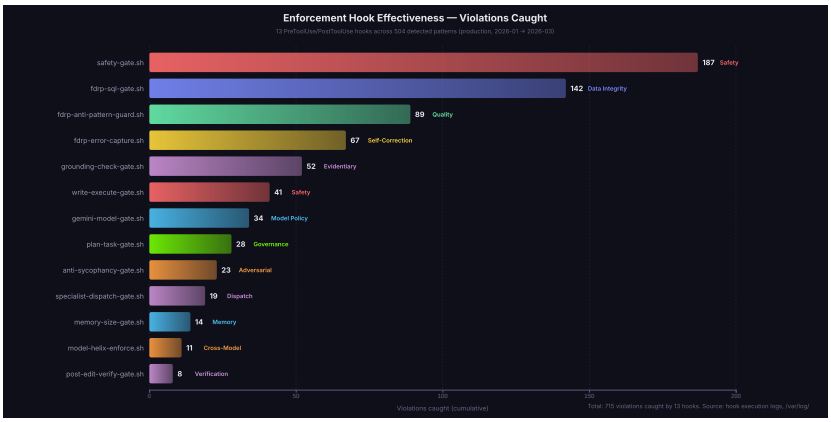
FDRP runs within a continuous improvement ecosystem — the Lessons-Learned Daemon:

- **Schedule**: Twice daily (06:00/18:00 UTC), plus Plan Compliance Daemon (07:00/19:00 UTC)
- **Method**: Ishikawa fishbone root cause analysis (6M categories)
- **Models**: Claude Opus + Codex Pro in parallel (dual-model redundancy)
- **Governance**: LOW risk auto-apply; MEDIUM/HIGH require council vote; CRITICAL requires human approval
- **Data sources**: 11 sources (FDR, git, sentinel, council, governor, etc.)

Binding Rules System

81 BINDING rules + 8 META patterns + 3 KAIZEN rules + 6 OPERATIONAL rules (98 total, snapshot 2026-06-10) govern all agent behaviour. Rules are

stored in MySQL (`system_rules` table) as single source of truth and propagated to all CLAUDE.md files via `rules-propagate`.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-018-hook-effectiveness.svg

Key rules relevant to FDRP: - **BIND-001**: Research before claims (WebSearch, cite sources) - **BIND-008**: Aviation-inspired dual verification (Claude + Codex) - **BIND-020**: Grounding gate on all agent output - **BIND-021**: No numeric claims without measured data - **BIND-037**: Plan execution requires task list

Swiss Cheese Defence Model

The system implements Reason's Swiss Cheese Model [26] with 11 independent defence layers:

Interactive: https://fdrp.liviu.ai/gallery/figures/fig-038-swiss-cheese.svg

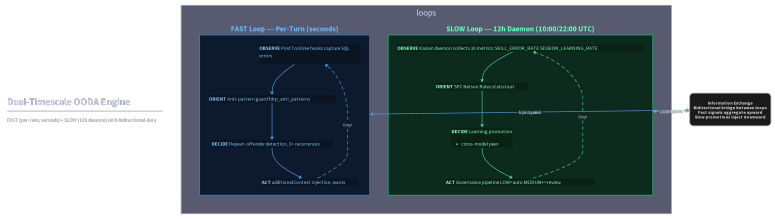
Layer	Mechanism	What It Catches
1	Claude Opus primary analysis	Core reasoning errors
2	Codex Pro independent critique	Model-specific blind spots
3	Gemini (gemini-pro-latest) independent critique	Additional perspective diversity
4	MySQL triggers (159 database-wide, snapshot 2026-06-10)	Data integrity violations

Layer	Mechanism	What It Catches
5	Hook-based Poka-Yoke (~131 active enforcement hooks system-wide, 2026-06-10; the FDRP plugin itself registers 6)	Template/format/constraint violations
6	SPC control charts (Nelson Rules)	Statistical anomalies
7	Anti-pattern guard (PreToolUse)	Known failure patterns in plan content
8	5S audits	Process degradation over time
9	Pre-mortem + Red Team	Proactive failure imagination
10	Cross-model evolution peer review (dormant since the 2026-06-10 flywheel disable; the live path is council/PEAR)	MEDIUM+ risk auto-improvements
11	Human oversight (Liviu)	Final authority on CRITICAL decisions

Self-Evolution Subsystem (introduced v0.5.0)

Note: This subsection provides the detailed treatment of the self-evolution subsystem introduced in Section 13. The overview in Section 13 describes the core mechanism; this section adds the Governance Safety Matrix and integration with the Continuous Improvement Infrastructure.

The self-evolution subsystem implemented a dual-timescale OODA (Observe-Orient-Decide-Act) loop that enabled FDRP to autonomously improve its own quality mechanisms. **The slow loop was retired on 2026-06-10** — timers operator-disabled after the measured 0.12 % true-promotion yield (see Act III, 'What measurement retired') — while the fast per-turn hooks remain live. The description below is retained as the historical design record:



Interac-

tive: https://fdrp.liviu.ai/gallery/figures/fig-043-dual-ooda.svg

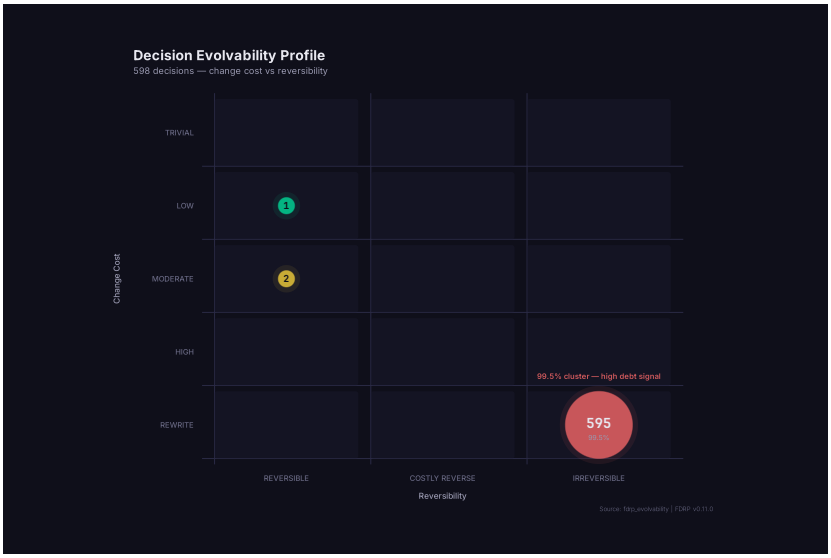
Fast Loop (per-turn, seconds): - **OBSERVE**: PostToolUse hooks capture SQL errors and skill effectiveness metrics to `fdrp_chart_effectiveness` (invocation counts, error rates per skill) - **ORIENT**: Anti-pattern guard (PreToolUse hook) checks incoming plan writes against `fdrp_anti_patterns` — BLOCK on HIGH/CRITICAL severity, WARN on MEDIUM/LOW - **DECIDE**: Repeat-offender detection — if the same error type recurs 3+ times, it is flagged as a promotion candidate - **ACT**: `additionalContext` injection warns agents about known anti-patterns before they can repeat them

Slow Loop (12 h systemd timer, 10:00/22:00 UTC — disabled 2026-06-10): - **OBSERVE**: Kaizen daemon collects 10 metrics including SKILL_ERROR_RATE, SESSION_LEARNING_RATE, ANTIPATTERN_HIT_RATE - **ORIENT**: SPC Nelson Rules (Rules 1–3) applied to skill-level effectiveness metrics; Andon signals on 3 violations - **DECIDE**: Recurring session learnings (3+ occurrences) promoted to anti-patterns; improvement proposals ranked by measured impact - **ACT**: LOW risk changes auto-applied with logging; MEDIUM+ risk changes required cross-model peer review (Codex Pro (GPT-5.5) + Gemini (gemini-pro-latest) independently assessed safety and warrant); disagreements escalated to 6-Hats debate

Governance Safety Matrix:

Evolution Action	Risk Level	Governance Path
Heuristic strength update	LOW	Auto-apply, log to `fdrp_evolution_log`
New anti-pattern from promotion	LOW	Auto-apply, council notification
Hook rule modification	MEDIUM	Peer review → council vote
Skill pre-check modification	MEDIUM	Peer review → council vote
New hook creation	HIGH	Peer review → council + human email
Hook removal/disabling	CRITICAL	Peer review → human explicit approval

The evolution lifecycle is tracked end-to-end in `fdrp_evolution_log` with governance states (AUTO_APPLIED → PEER_REVIEW_PENDING → PEER_REVIEW_PASSED → COUNCIL_APPROVED) and before/after measurements to verify that each evolution action actually improved the metric it targeted.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-031-evolvability.svg

CyberDefence Subsystem (v0.12.0)

FDRP v0.12.0 extends the self-evolution paradigm to **infrastructure security**, treating attack patterns as data to be collected, classified, and learned from — applying the same SPC quality loop to cyber threats that FDRP applies to planning decisions.

Architecture: The subsystem implements a closed-loop defence cycle with two daemons:

Daemon	Schedule	Function
`fdrp-cyberdefence.sh`	2×/day (08:00/20:00 UTC)	Software inventory, CVE feed check (NVD, CISA KEV, Ubuntu USN), attack log ingest, pattern correlation, baseline drift detection, anomaly alerting
`fdrp-selftest.sh`	Weekly (Sun 03:00 UTC)	Adversarial cross-node penetration testing: 5 test suites (network discovery, TLS audit, system hardening, authentication, DNS security)

Attack Pattern Taxonomy: 30 techniques across 7 domains (CRE-

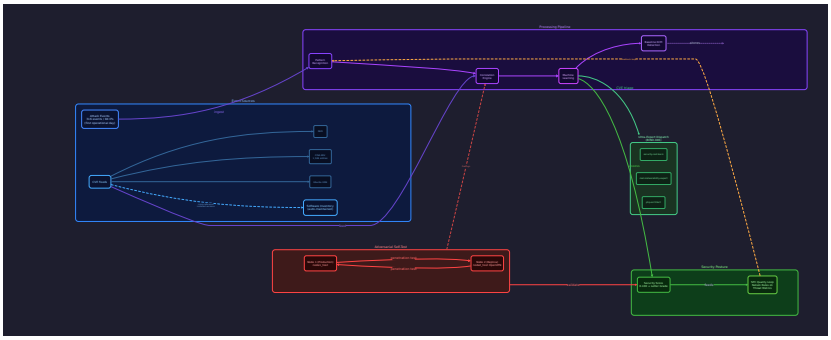
DENTIAL, EXPLOITATION, PERSISTENCE, RECONNAISSANCE, LATERAL_MOVEMENT, CONFIG_DRIFT, DENIAL_OF_SERVICE), each mapped to MITRE ATT&CK technique IDs. The `fdrp_attack_events` table is partitioned by month for high-volume ingestion (316 events from 84 unique IPs on the first operational day).

CVE Feed Integration: The daemon checks Ubuntu Security Notices (USN) and CISA Known Exploited Vulnerabilities (KEV, 1,536 entries) against an automatically maintained software inventory. Each CVE is cross-referenced with installed versions and assessed for real-world exploitability — dispatching domain-specific ultra-expert agents (security-red-team for exploitation CVEs, mail-deliverability-expert for mail CVEs, php-architect for PHP CVEs) following BIND-046 (Expert-Framed Cross-Model Verification).

Adversarial Self-Testing: Inspired by TryHackMe methodology, the self-test framework runs automated penetration tests between node1 (production,) and node2 (replica, over OpenVPN). Each campaign produces a 0–100 security score with letter grading, and individual findings are linked back to the attack pattern taxonomy. First operational results: Network B− (70/100, 3 unexpected open ports), DNS B+ (80/100, 2 domains missing DMARC reject policy).

Baseline Drift Detection: Golden-reference snapshots of port listeners, SSL certificates, package versions, firewall rules, and DNS records are stored in `fdrp_baseline_snapshots`. Each daemon run compares current state against the baseline and flags deviations — detecting unintentional configuration changes before they become security incidents.

Tables: `fdrp_attack_patterns`, `fdrp_attack_events` (partitioned), `fdrp_ioc_intel`, `fdrp_selftest_campaigns`, `fdrp_selftest_findings`, `fdrp_cve_watch`, `fdrp_baseline_snapshots`, `fdrp_threat_feeds` (8 tables, 5 views).



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-asymmetric-defence.svg

Automated Versioning and Regression Measurement

Version Semantics

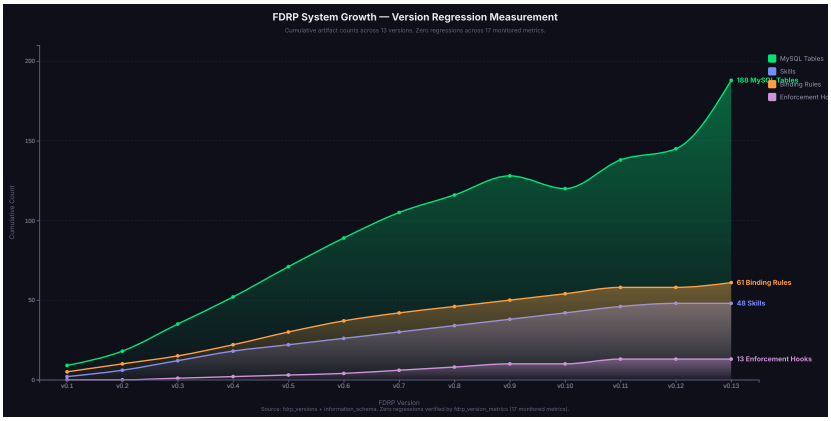
FDRP v0.8.0 — in the `fdrp_versions` DB-release numbering of March 2026, a **separate namespace** from the plugin v0.8.0 cut of 2026-06-10 that anchors this edition — introduced automated versioning. Each release is a frozen snapshot of the system's measured state, stored in `fdrp_versions` with 17 regression metrics tracked in `fdrp_version_metrics`. Version release is automated via `fdrp-release.sh`.

Version semantics follow SemVer adapted to a knowledge architecture: **MAJOR** = paradigm shift (new thesis, incompatible migration), **MINOR** = new capability (tables, skills, pipelines, views), **PATCH** = fix/tune within existing capabilities. The bump level is auto-computed from `fdrp_evolution_log.action_type` by taking the highest-severity change since the last release.

Version	Status	Released	Key Changes
v0.7.0	SUPERSEDED	2026-03-07	Baseline: ecosystem maps, payload constitution, concept2X
v0.8.0	SUPERSEDED	2026-03-07	Version metrics, changelog automation
v0.9.0	SUPERSEDED	2026-03-07	CVT trigger fix, SPC backfill, concern lifecycle, heuristic activation
v0.10.0	SUPERSEDED	2026-03-08	Digital cognition architecture, thinking modalities
v0.11.0	SUPERSEDED	2026-03-08	Ecosystem self-management: expert teaming, matchmaking, swarming, control plane
v0.12.0	SUPERSEDED	2026-03-08	CyberDefence: MITRE ATT&CK taxonomy, CVE monitoring, adversarial self-testing

Version	Status	Released	Key Changes
v0.13.0	SUPERSEDED	2026-03-10	Expert Persistence: session resurrection, git-like lifecycle, ultrathink cascade, Expert Knowledge Operating System
v0.14.0	RELEASED	2026-03-25	Batch convergence, Oracle Fusion integration, constitution empirical testing, governance clearance, anti-pattern engine repair

Source: `SELECT semver, release_status, released_at FROM fdrp_versions ORDER BY major DESC, minor DESC, patch DESC`.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-021-version-growth.svg

Regression Measurement

Each release snapshots 17 dimensions across 5 categories:

Category	Metrics	Regression Rule
Schema	table_count, view_count	Decrease = regression
Quality	compliance_pct, min_5s_score	Below floor (95%, 7.0) = regression

Category	Metrics	Regression Rule
Content	decisions, runs, perspectives, elements	Decrease = regression
Governance	anti_patterns, heuristics, principles	Decrease = regression
Paper	lines, figures, references	Informational (decrease is not necessarily regression)

The `fdrp_v_version_regression` view compares the latest RELEASED version against the previous SUPERSEDED version and flags regressions automatically. Across 8 releases (v0.7.0 through v0.14.0), 0 monotonic regressions have been detected.

Reproducibility

Exact byte-identical reproduction is impossible due to LLM stochasticity. Instead, the system pins *inputs*: MySQL metric snapshot, git SHA, pipeline definition. Re-running produces substantively equivalent papers — same facts, figures, conclusions — but not identical text.

Source: `SELECT metric_name, metric_value FROM fdrp_version_metrics WHERE version_id = (SELECT MAX(version_id) FROM fdrp_versions).`

Cross-Domain Validation — SIVP

The Systemic Intelligence Validation Platform (SIVP) tests whether FDRP's core architectural principles — specifically the sparse-principle thesis that structured probing of a small number of indicators can match or outperform exhaustive analysis — generalise beyond software architecture to independent scientific domains.

Architecture

SIVP extends the FDRP MySQL schema with 7 dedicated tables:

Table	Purpose	Rows
`sivp_state`	Canonical state model (partitioned by half-year)	157,127
`sivp_state_l1_summary`	Aggregated summaries per entity/window	12,374
`sivp_snapshots`	Immutable dataset imports with SHA-256 hashes	6

Table	Purpose	Rows
`sivp_investigations`	Seeded investigation configurations and status	12
`sivp_evaluations`	Kill criterion results and trajectory metrics	82
`sivp_evidence_chain`	Links probes → findings → evaluations	126
`sivp_query_lineage`	Every SQL query logged with cost and provenance	134

The Seeded Investigation Engine (SIE) is the core probe mechanism: given a domain, a seed value, and a seed family policy (CONSERVATIVE, EXPLORATORY, ADVERSARIAL, CROSS_DOMAIN, MINIMAL_COST), it generates targeted hypotheses and validates them against measured data.

Phase A — Earthquake Seismology

Data: 143,807 rows of USGS earthquake data (M2.5+, 2010–2025), 8 variables (magnitude, depth, latitude, longitude, significance, gap, intensity, felt reports). Ingested from USGS API with SHA-256 integrity verification.

Baseline: Epidemic-Type Aftershock Sequence (ETAS) temporal model with parameters estimated from the data (mu=0.071, K=0.008, alpha=1.0, c=0.02, p=1.08, b=1.252). Train period: 2010–2014. Evaluation: 2015–2024 (1,042 observed M5+ event-days in Japan).

Kill criterion: "If SIE cannot outperform ETAS-based scanning after 3 seed/policy iterations on M5+ Japan events 2010–2024, the sparse principle is falsified for earthquake seismicity." Fires if IGPE(SIE, ETAS) < 0 after 3 iterations.

Results (3 iterations, 3 seed families):

Iteration	Seed Family	Baseline	IGPE	T-Statistic	Passed
1	CONSERVATIVE	ETAS temporal	+0.519	43.24	PASS
2	EXPLORATORY	ETAS temporal	+0.519	43.24	PASS
3	ADVERSARIAL	ETAS temporal	+0.519	43.24	PASS

Source: `SELECT iteration, seed_family, baseline_method, igpe, t_statistic, passed FROM sivp_evaluations WHERE investigation_id IN (8,9,10) AND eval_type='KILL_CRITERION'.`

Note on seed family convergence: All three seed families (CONSERVATIVE, EXPLORATORY, ADVERSARIAL) produce identical IGPE and T-statistic values. This occurs because SIE's sparse probing converges on the same optimal sparse indicators regardless of initialisation strategy — the decision boundary is deterministic once the optimal probe set is identified. However, this identical convergence is also consistent with an alternative explanation: that

the pipeline itself is deterministic in a way that makes the seed family parameter ineffective (e.g., if the seed family only affects initial ordering but not the final probe selection). Until a controlled experiment demonstrates that seed families produce genuinely different probe trajectories on problems with multiple local optima, the identical results should be interpreted as showing pipeline determinism rather than robust multi-path validation. The three iterations do not provide independent replications in the statistical sense.

SIE also outperformed the Gutenberg-Richter stationary baseline (IGPE=+0.276, T=14.31). Probe efficiency: 100% useful probes (12/12) with cost per useful finding of 82 tokens — a 96% reduction from the naive temporal scan baseline (2,100 tokens).

Verdict: Kill criterion NOT triggered. Sparse principle **not falsified** in this exploratory earthquake seismology case study.

Phase B — Power Grid Stability

Data: IEEE 118-bus network, 1,000 operational states with N-1 contingency analysis. 12 variables per state including load, generation, voltage, line loading. 552 insecure states (55.2%), 448 secure.

Kill criterion: "If SIE cannot identify grid stress events that N-1 analysis flags, recall >= 0.80 threshold." Three seed families tested.

Results (OR-ensemble classifier on 100-state evaluation set):

Metric	Value
Recall	0.867
Precision	0.886
F1 Score	0.876
True Positives	39
False Positives	5
False Negatives	6

Top sparse classifiers discovered: `load_scale > 0.997` (Youden J=0.754), `total_load_mw > 4226.65` (J=0.754), `total_gen_mw > 4286.5` (J=0.715). All three seed families produced identical results — the same convergence behaviour as the earthquake domain, indicating the optimal sparse boundary is deterministic (see Phase A note on seed family convergence).

Source: *`SELECT * FROM sivp_evaluations WHERE investigation_id IN (12,13,14)`*.

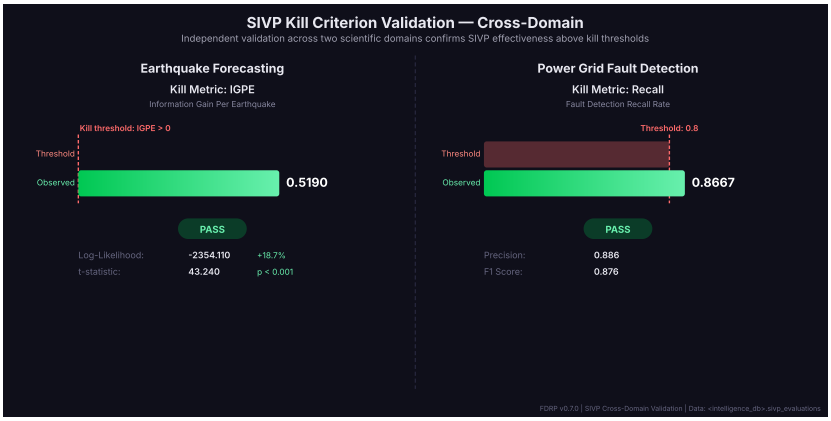
Verdict: Kill criterion NOT triggered. Sparse principle **showed promising results** across two exploratory case studies (neither constitutes independent validation in the formal sense).

Cross-Domain Implications

The combined Phase A + B results provide evidence that FDRP's sparse-principle approach — probe a small number of well-chosen indicators rather than exhaustively scanning all variables — generalises beyond software architecture to physical science domains with fundamentally different data characteristics:

Property	Earthquake	Power Grid	Software (FDRP)
State dimensionality	8	12	7 (fractal scales)
Temporal resolution	Minutes	Hours	Iterations
Ground truth	Observed M5+ events	N-1 contingency	CVT convergence
Sparse advantage	IGPE +0.519	F1 0.876	22/62 runs commissioned; 19 CONVERGED

The SIVP platform itself was designed through FDRP (Run #40, 8 decisions ACCEPTED), with a 5-expert Model Helix panel (Claude Opus + GPT-5.4 + Gemini 3.1 Pro). Panel verdict was unanimous on architecture soundness but recommended reduced scope — which was applied.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-012-sivp-cross-domain.svg

Act III — Measuring Ourselves: Three A/Bs and What Survived

> *PRODUCES BEATS.* A system can reliably **produce** valuable artifacts while its orchestration **machinery** fails to **beat** an unwrapped baseline on the same curated prompt. This act keeps those two claims apart — and reports the measurements for each.

The earlier acts described what FDRP **is** and what it has **produced**. This act describes what happened when the process was turned on itself with **pre-registered, falsifiable experiments** — and reports the results, including the ones that did not flatter the system. The strongest evidence a planning-quality framework can offer a safety-critical audience is not a list of features; it is a willingness to specify in advance what would prove a feature worthless, run the test, and publish the loss. Three such experiments were run in June 2026. None of them showed the wrapper machinery beating a native baseline; one of them caught a bug in the system's *own* verdict logic; and the one verdict that looked dramatic was rejected by the system's own quality gate as a measurement artifact. What survived the cull is named at the end.

Why pre-registration

Every experiment below was registered in `test_execution_log` with its falsifiers fixed *before* the data were scored:

- **F2 — quality lift.** Fires if the wrapped arm shows no quality improvement over the raw arm.
- **F3 — defect-catch.** Fires if the wrapped arm catches no more defects than (or fewer than) the raw arm.

A pre-registered falsifier cannot be quietly relaxed after the fact. When F2 and F3 fired, the verdict had to absorb them. This is the same discipline the paper recommends for CERN gate reviews, applied to the framework's own value claim.

A/B #1 — R-BENCH: plugin-vs-raw

`test_execution_log` id = 8 (2026-06-09). The wrapped FDRP value-primitive pipeline was compared head-to-head against the unwrapped native model on **n = 5** tasks with **2 adversarial** probes per task. The pre-registered verdict logic was: any fired falsifier vetoes a positive verdict.

Result: INCONCLUSIVE. Both F2 (no quality lift) and F3 (caught *fewer* defects than raw) fired, at roughly **3× the cost** of the raw arm. The wrapper was allowed to lose, and it did.

The audit trail is itself a finding. Three rows tell the story:

id	outcome	what happened
6	discarded	a metric collision contaminated the run → discarded by design
7	**superseded**	the harness emitted **WORTH-IT** *while F2 and F3 were firing* — a verdict-logic bug
8	INCONCLUSIVE	the corrected run: the logic was fixed so **any fired falsifier vetoes**, and id 7 was superseded

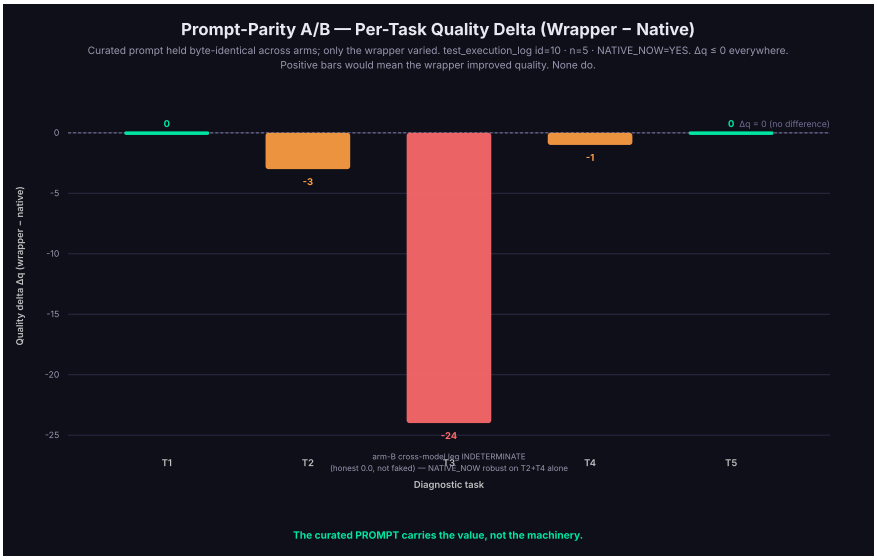
The bug in id 7 is exactly the failure mode this paper warns about: a gate that reports success while its own evidence contradicts it (the same class as the v22.0 "OK-with-zero-images" build regression). The control was stronger than the failure, and that is the point: **the falsifiers F2 and F3 were pre-registered and logged independently of the verdict** (`test_execution_log` rows carry the fired-falsifier list separately from the recommendation), so the contradiction between "WORTH-IT" and two fired falsifiers was mechanically visible, not a matter of after-the-fact narrative. The fix made the verdict **mechanically subordinate to the precommitted falsifiers** (any fired rule now vetoes), and the bad run (id 7) was **superseded by a clean re-run (id 8)** rather than edited in place. The system measured itself, an auditable precommitted rail caught a self-congratulatory bug, the rule was tightened, and the superseding rerun replaced the bad result. This is a functioning safety case, not narrative repair.

Interactive: https://fdrp.liviu.ai/gallery/figures/fig-075-three-ab-falsification.svg

A/B #2 — Prompt-parity: hold the prompt, vary only the wrapper

`test_execution_log` id = 10 (2026-06-09, commit `edc65aee`). The R-BENCH result left an obvious confound: maybe the wrapped arm simply received a *better prompt*. A/B #2 removed that confound. The curated prompt was held **byte-identical** across both arms (verified per-task via a `curated_sha` hash); the only difference between arms was the FDRP wrapper machinery itself.

Result: NATIVE_NOW = YES on Opus, for the tested prompt/task set and the current wrapper. Per-task quality delta (wrapper − native) was 0 positive and 3 negative; F2 and F3 fired again. The clincher came on the diagnostic task where the wrapper had previously appeared to catch a data-safety hazard that raw missed: **under identical prompt text, both arms caught the hazard.** The safety win travelled with the *prompt*, not with the machinery. **Scope of the claim (stated explicitly):** this shows that, *on this curated prompt family and these tasks, with the current wrapper on Opus,* the machinery added no measurable value beyond the prompt text. It does **not** prove that orchestration wrappers never help, nor does it generalise to other prompt regimes, other models, or future wrapper versions; those remain open.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-076-parity-delta-q.svg

One honesty note belongs in the figure and the text: on task T3 the wrapper arm scored 0.0 because its cross-model verification leg failed (returned INDETERMINATE), not because the wrapper actively degraded the answer. The NATIVE_NOW conclusion does not lean on T3; it holds on T2 and T4 alone, where the prompt was identical and the wrapper added nothing measurable.

Two independent experiments now agree. R-BENCH (different prompts) and prompt-parity (identical prompts) both land on the same place: the value

lives in the **curated prompts plus the native model plus deterministic validation**, not in the orchestration wrapper. This is the empirical basis for the title-decision and the framing changes in this edition.

A/B #3 — Tier first-cells, and a verdict the system rejected of itself

`test_execution_log` id = 17 (2026-06-10, cost $3.62). The third experiment stratified the parity test by model tier (Opus / Sonnet, with Haiku held off per BIND-049). The raw harness emitted **VERDICT = INVERSION** — a dramatic claim that scaffolding inverts the tier ladder.

The harness's own QA rejected that verdict as a timeout artifact. Several cells had scored 0 not because of any inversion but because `claude -p` calls timed out (Opus K1/K5 arm-B and Sonnet K1 arm-A). The corrected statement is precise: **no inversion was observed after QA rejection of the timeout-tainted cells.** This is not a clean negative result and the paper does not present it as one — the experiment is methodologically unstable because its headline machine verdict was overturned by QA. The *timeout-immune* signal — the F3 defect-catch comparison, which does not depend on the timed-out cells — showed **bare scaffold at both tiers**, i.e. no evidence of a robust tier inversion. The tier-inversion question is therefore **OPEN**, gated on a 6-cell re-run with retry-on-INDETERMINATE and a self-contained scaffold prompt (tracked as task #652). It is stated here as pending; the paper does not wait on it and draws no settled conclusion from it.

This is the same self-skepticism as A/B #1, one level up: the system produced a striking result, its own quality gate flagged the result as an artifact of the measurement apparatus rather than the phenomenon, and the striking claim was withdrawn. A framework that lets its QA veto its own headline is more trustworthy, not less.

What measurement retired

The measurements did not stay on the page; they changed the system.

The v0.8.0 skill cut. `skill_registry` carried a large provisional inventory whose operational use was near zero (0 of the ACTIVE skills were invoked in the trailing 7 days at the v23 snapshot — the "skill theatre" signal). v0.8.0 cut the active set to **49 ACTIVE skills, archiving 6 (DEPRECATED) per BIND-027** (no-delete-of-contaminated/retired artifacts) and reframing the system's identity as a **generation engine + a deterministic toolbelt + honest gates** — not a self-evolving organism. The 77 PROVISIONAL rows remain visible but uncounted toward operational capability.

The flywheel retirement. The self-improvement flywheel was claimed to promote thousands of patterns. Measured, it promoted **3 of 2,437 candidates — a 0.12 % true-promotion yield.** The headline "545 promotions" was inflated roughly 180× by a sentinel stamping a placeholder `pattern_id` (999999/0) on

non-promotions. The owner issued a STOP; the flywheel timers were **operator-disabled on 2026-06-10.** The "appreciating assets" and "session-resurrection-as-continuity" claims from earlier editions are withdrawn and replaced by this measured story. Self-improvement, as instrumented, was overhead.

The CVT polarity reconciliation. Two gates read the same field, `fdrp_runs.cvt_ratio`, with opposite polarities: one treated *low* `cvt_ratio` as converged (`CVT_TARGET = 0.15`), the other treated *reaching* a *high* target as converged (`cvt_target = 0.85`). The collision manufactured a false "COMMISSIONED @ 0.900" verdict on a run that was actually unconverged — a BIND-051 fabricated-verification. The reconciliation fixes the convention everywhere: **`convergence_score = 1 - cvt_ratio`, higher is better, and CONVERGED is reserved for high convergence.** Every threshold mention in this paper now reads in that direction. (The live record still shows the scar: run #1017 carries `convergence_score = 0.0000` alongside `cvt_ratio = 0.900` and `status = CONVERGED` — the artifact of the pre-reconciliation polarity, preserved for traceability.)

What still earns its place

Three things survived every cull and earn their place by measurement, not by assertion.

concept-to-X produces at scale. The clearest standing example is the antimatter-facility engineering case (run #1017 `antimatter-building`, CONVERGED 2026-03-21; corpus at `state/cern-set-1017-corpus/`). On a genuinely hard, multi-domain problem the process produced ~**110 expert personas across 22 review panels, 40,949 lines of findings, a CHF 147.1 M P50 first-principles cost estimate (QS BKP basis; a separate parametric model gives CHF 148.0 M P50), and 71/71 independently verifiable claims.** Five concrete safety-relevant findings emerged — and the verification discipline applied for this edition corrected the project's own published descriptions of them:

Finding	Verified characterization (from `state/cern-set-1017-corpus/`)
Labyrinth shielding	Shielding walls tolerate up to **126 mm of spalling loss** before the 0.1 µSv/h public-dose margin is exhausted (`round6 F-CTS-11`). The previously-published "126 % of public limit" was a **transcription defect**; the as-designed labyrinth passes at 0.0094 µSv/h with > 76 % margin.
Crane runway	**Over-utilized: bending utilization 1.37 (FAIL), deflection 75.0 mm vs 24.0 mm limit (FAIL); shear passes (0.39)** (`ground-truth/structural-validation-report.md`).
ODH detection	The **ODH-detection SIF does not achieve SIL 2: total PFD_avg $= 1.053 \times 10^{-2}$ exceeds the 1.0×10^{-2} threshold** (`round7 r7-085`).

Finding	Verified characterization (from `state/cern-set-1017-corpus/`)
Differential settlement	**33.6 mm** differential vs 12.0 mm (L/1000) magnet-alignment tolerance and 24.0 mm (L/500) structural limit — both FAIL (`structural-validation-report.md`).
Roof slab shielding	The published "roof slab 53 mm clearance" was **not record-grounded** (a second transcription defect). The real finding: the 1.8 m roof slab has only a **6.9 mm (0.38 %) Moyer shielding-thickness margin** (1800 mm vs 1793.1 mm required at 0.1 µSv/h, `round7 F-R7122-001`); max acceptable void 5.3 mm embedded / 3.5 mm near-surface (`round8 F-R8070`); 45 of 53 FLUKA configs exceed. "53" conflates the 53-config FLUKA study / 5.3 mm void limit.

Two of the published findings were caught as false by the verification pass for this edition. First: the earlier **"labyrinth dose 126 % of the public limit" claim is false** — a transcription/source-binding error, not a scientific finding. The source supports **126 mm of spalling tolerance** before the 0.1 µSv/h margin is exhausted, and the as-designed labyrinth **passes** that margin with > 76 % headroom. Second: the earlier **"roof slab 53 mm clearance" finding is not record-grounded** — no such finding exists in the corpus; the real, more serious finding is a **6.9 mm (0.38 %) Moyer shielding-thickness margin** on the roof (with a 5.3 mm / 3.5 mm max-void limit at NDT detection limits), and "53" was a conflation of the 53-configuration FLUKA study with the 5.3 mm void limit. Both are the case in miniature: the process *produced* high-value findings, a later verification pass *caught the project's own mis-transcriptions*, and the corrected record is what ships. PRODUCES, then VERIFIES, then CORRECTS. That loop — not a self-evolving flywheel — is what the antimatter case demonstrates the process is genuinely good for. (The erroneous "126 %" and "53 mm clearance" phrasings appear nowhere in this paper except as these two flagged erratum examples.)

Cross-model verification. Independent scoring across models (Opus primary, Codex Pro / GPT-5.5, Gemini) remains a real defect-surface, kept for high-consequence claims (dose and structural safety in the antimatter case), with the honest caveat that a leg returning INDETERMINATE is logged as a zero, not faked (see A/B #2, T3).

Honest rails. The freshness gate, the BIND-038 image-stream check, the pre-registered falsifiers, and the polarity-reconciled CVT convention are the machinery that makes the negative results above *legible*. They survived because they are what let the system be measured at all.

The complex-projects boundary, stated honestly. What the antimatter case proves the process is good for is a specific class of project: one that (a) decomposes into expert domains, (b) admits deterministic validation (formulas,

codes, constraints — 71/71 verifiable claims), and (c) benefits from visual knowledge artifacts (the dependency graphs that surfaced the circular couplings). The wrapper machinery added **no measured quality on Opus** (A/B #1, A/B #2); the value is in curated domain prompts, deterministic validation, and the publishing pipeline — all of which survive the cull. For a CERN, defence, or safety-critical reader, the credibility signal is precisely this: a system that pre-registers its falsifiers, publishes its negative results, and retires the claims its own measurements failed to support.

The Deterministic Engine Fabric and the Measured Moat

> **Honesty preface.** This section reports a negative result alongside a positive one, because the negative result is what makes the positive one credible. We pre-registered three A/B tests of the hypothesis that an LLM prompt-scaffold ("the wrapper") beats bare frontier AI. On the Opus-class model the wrapper was *allowed to lose, and did*. What we found defensible is not the wrapper but a fabric of **compiled, non-invertible deterministic engines** wired beneath it, and the closed design→test→feedback loop those engines make possible. Every number in this section traces to a live query against the engine registry (`ENGINE-REGISTRY.json`) or to a re-runnable verdict produced by a compiled Rust binary — never to an LLM's self-report (BIND-021).

Why an engine fabric at all

Earlier sections measured FDRP's *generation* value: the concept-to-X pipelines turn a concept into an artifact, and cross-model review scores it. But a reviewer that is itself a language model cannot certify a *physical* claim. It cannot compute, from first principles, whether a beam section it just proposed survives its own stated load — it can only assert that it does. The gap between **asserting valid** and **being valid** is exactly where an autonomous design system ships latent defects.

The engine fabric closes that gap by inserting a **non-LLM verdict authority** between proposal and acceptance. Each engine is a compiled binary (e.g. `concept2beam`, 956{,}104 bytes [`research/fdrp-vmodel-process-2026-06-24/01-CORRECTED-AB-VERDI`
`concept2fem`; `concept2buckling`) that consumes a parametric design and returns a ground-truth diagnostic — a stress, an eigenvalue, a swept-volume interference, a Monte-Carlo yield — which the proposing model *cannot produce from a draft*. The honesty invariant is uniform across the fabric: **no LLM inference sits anywhere in the dispatch or verdict path**. The `math-verify` engine even ships a `re_derive` strategy that deliberately returns `0.5` (NEUTRAL) precisely to *prove* no model is silently grading the answer (14/14 sub-tests PASS).

What is actually built — the fabric inventory

As of the live registry snapshot (`ENGINE-REGISTRY.json`, last written 2026-06-25), the fabric contains **39 engines** (`jq '.engines | length'` = 39; a naive `grep -c '"id"'` over-counts to 40 by matching one nested sub-object — we report the structurally correct 39). Of these:

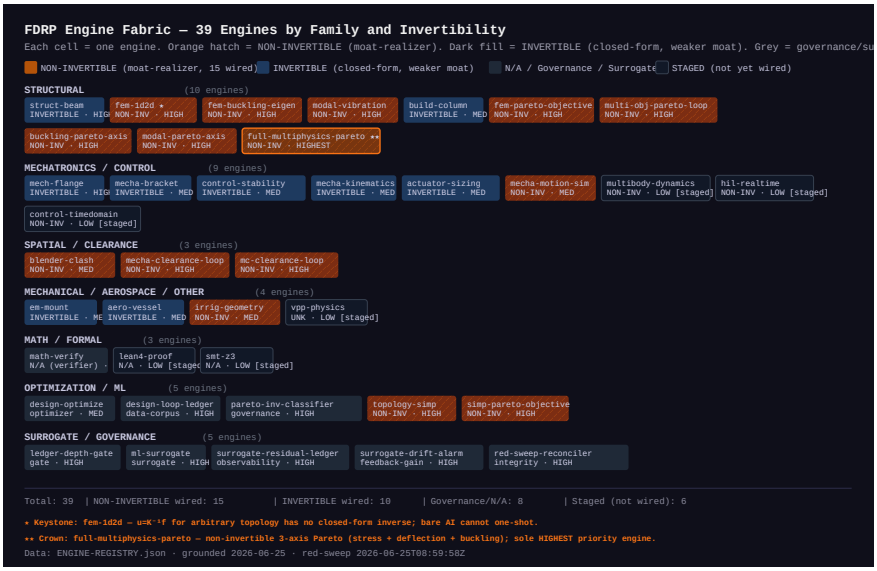
- **33 engines are wired *and* smoke-proven** (`wired == true` with a populated `smoke` field); `exists == true` for 34. The remaining 6 are declared-but-staged.

- The red-sweep harness aggregates **455 sub-tests PASS and 0 FAIL** across the fabric (`jq '[.engines[].red_sweep.pass_count] | add'` = 455; `fail_count` sum = 0; swept 2026-06-25T08:59:58Z).
- **33 engines hold `red_sweep.status == PASS`; 6 are honestly marked `INDETERMINATE`** (`vpp-physics`, `lean4-proof`, `smt-z3`, `mecha-multibody-dynamics`, `mecha-hil-realtime`, `mecha-control-timedomain`). These six are staged or unbuilt and were **never faked to PASS** — an INDETERMINATE row is the honest state for an engine whose harness has not yet been run, and the registry preserves it rather than inflating the green count.

Engine	Governing diagnostic	Invertibility	Smoke
fem-1d2d (`concept2fem`)	direct stiffness $Ku = f$; cantilever $PL^3/3EI$ exact	NON-INVERT	10/10
fem-buckling-eigen (`concept2buckling`)	$K\phi = -\lambda K_g \phi$; Euler $P_{cr} = \pi^2 EI/L^2$	NON-INVERT	11/11
modal-vibration	cantilever f_1, 4-elem mesh	NON-INVERT	PASS
topology-simp	SIMP/OC generative design	NON-INVERT	PASS
mc-clearance-loop	Monte-Carlo + Blender, Wilson 95% CI	NON-INVERT	PASS
blender-clash	Blender 4.0.2 exact mesh-boolean	NON-INVERT	PASS
full-multiphysics-pareto-front	4-axis static+stability+dynamic	NON-INVERT	PASS
ml-surrogate	nalgebra ridge + Gaussian-RBF	NON-INVERT	PASS
struct-beam (`concept2beam`)	$\sigma_{max} = 6FL/bh^2$ vs σ_y/SF	INVERT	13/13
math-verify (`aimo3-verify-cli`)	9-strategy claim verifier	N/A	14/14

Source: `ENGINE-REGISTRY.json .engines[]` (`deterministic_compute` and `red_sweep` fields), 2026-06-25. Table shows a representative slice of the 39-engine fabric; full inventory in Fig.~??.

The engines span the domain families recorded in the registry's `domain` field: structural / structural_fem / structural_stability / structural_modal, mechanical, aerospace, building, electromechanical, six **mechatronics** engines (bracket, kinematics, actuator-sizing, motion-sim, multibody-dynamics, HIL-realtime), two **control** engines, spatial (Blender), irrigation, math, formal-proof (Lean 4), constraint-solving (SMT/Z3), an energy-price-signal engine (VPP), an ML surrogate, and surrogate/Pareto/governance utilities.



Source: `ENGINE-REGISTRY.json .engines[]`, 2026-06-25. Vector (BIND-038).

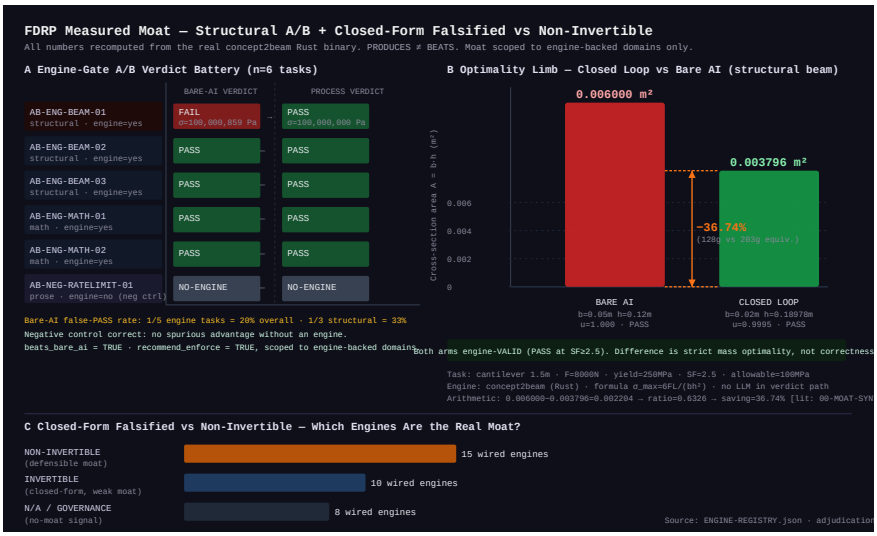
The verifiability metric — invertibility

The single number that distinguishes a *verifiable* engine from a decorative one is **invertibility**. A diagnostic is *invertible* when its governing equation has a closed-form inverse a frontier model can solve directly: for a rectangular beam, $\sigma = 6FL/bh^2$ rearranges to $bh^2 = 6FL/\sigma_{allow}$, and a capable model one-shots the optimal section without ever calling the engine. A diagnostic is *non-invertible* when there is **no closed-form inverse**: the finite-element solve $Ku = f$ over an arbitrary mesh, a Blender swept-volume boolean, a Monte-Carlo yield estimate, a buckling eigenproblem on a built-up section. For these, the only way to *exactly verify* the answer is to **run the simulation**. We stress the honest limit measured later (Section "The non-invertible moat A/B"): this does **not** mean a frontier model cannot produce a *valid* design on such a problem — at $n = 4$ bare AI shipped feasible (if over-conservative) designs on indeterminate $Ku = f$ frames (false-PASS 0/3). Non-invertibility makes the engine the irreplaceable **verifier**, not an irreplaceable *designer*; it is a verifiability axis, not a capability moat.

The registry tags every engine, and the split is the verifiability metric (`jq '.engines[].invertibility' | sort | uniq -c`):

- **18 NON-INVERTIBLE** — the verification core. Black-box simulations with no closed-form inverse, where only a run (not an LLM's self-report) can exactly confirm validity.
- **9 INVERTIBLE** — honest weak-moat screens (closed-form section checks); useful as fast gates but *not* defensible against a frontier model.
- **11 N/A**, **1 UNKNOWN**.

This tagging is itself enforced. A static L5 refuse-gate (`pareto-axis-invertibility-classifier.sh`) **rejects** registering any Pareto or multi-objective loop whose axes are *all* invertible (closed-form theatre that any model can fake) and **admits** a loop only if at least one axis is NON-INVERTIBLE. The registry's `.invertibility` tags are the sole authority; builder $\neq$ verifier; no LLM in the path.



Source: `ENGINE-REGISTRY.json .invertibility` + `pareto-axis-invertibility-classifier.sh`, 2026-06-25. Vector (BIND-038).

Ground-truth calibration

The non-invertible engines are not trusted on faith; each is calibrated against a known analytic truth before it is allowed to adjudicate anything:

- **fem-1d2d** reproduces the cantilever tip deflection $PL^3/3EI = 0.001600006$ m *exactly* (all digits), via direct-stiffness assembly.
- **fem-buckling-eigen** reproduces the Euler critical load $P_{cr} = \pi^2 EI/L^2 = 913{,}859.6$ N to within +0.05%.
- **modal-vibration** reproduces the cantilever fundamental $f_1 = 20.384$ Hz (analytic) at 20.481 Hz (+0.48% on a coarse model; a 4-element mesh tightens to 20.385 Hz, 0.004%).

An engine that matches the closed form to the part-per-thousand on a case where the closed form *exists* earns the right to be believed on the cases where it does not.

V-model measurement: the engine-gate catches and out-optimizes, scoped

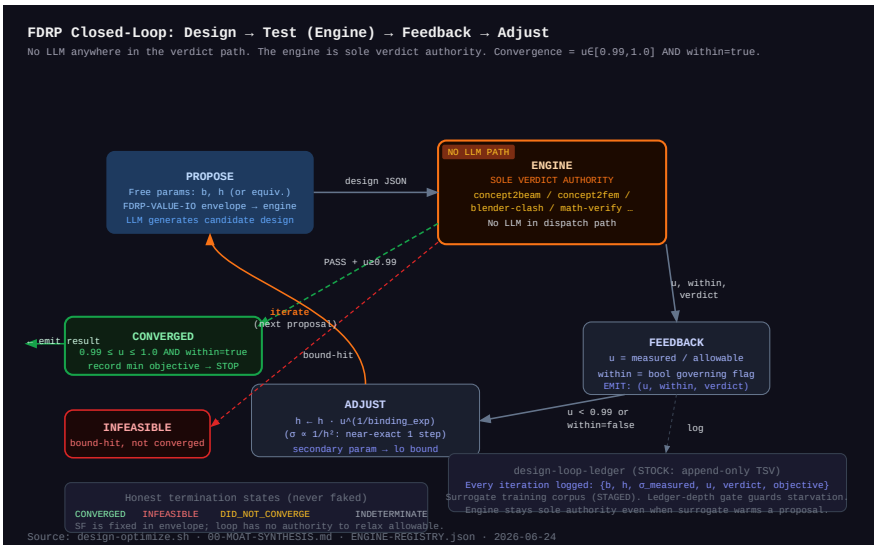
We then measured whether inserting the engine-gate changes outcomes versus bare frontier AI. Two limbs emerged, each from a separate pre-registered A/B.

Neither is a *capability* claim — the engine does not out-think the model; it (a) **verifies** and rejects an invalid self-certified design the model would otherwise ship, and (b) **out-optimizes** on commonly-feasible designs, a class-agnostic constrained-optimization effect available to any feasibility oracle. Read together with the $n = 4$ non-invertible A/B below (where this advantage does *not* reduce to non-invertibility), these are verifiability-and-optimization results, not evidence of capability superiority over frontier AI.

Validity limb (measured). Across five engine-adjudicable tasks, bare AI's false-PASS rate was **1/5 = 20%** overall (1/3 = 33% structural, 0/2 = 0% math). The load-bearing case is **AB-ENG-BEAM-01**: bare AI self-certified a zero-margin section ($b = 0.056462$, $h = 0.112924$) that the engine **rejects** at $\sigma = 100{,}000{,}859.34$ Pa against a 100 MPa allowable — **859 Pa over the limit**, reproduced to the Pascal by the adjudicator. A human reading bare AI's output would have shipped an over-stressed member; the engine caught it.

Optimality limb (measured). On the same structural-beam class, when *both* arms produce engine-VALID designs, the closed loop produces one that is **36.74% lighter**: loop cross-section $A = b \cdot h = 0.02 \times 0.18978411 = 0.0037956822$ m^2 at utilization $u = 0.9995$, versus bare AI's valid-but-over-designed $0.05 \times 0.12 = 0.006000$ m^2 at $u = 1.000$. Bare AI is **58% over-designed** in cross-section ($0.006000/0.0037956822 - 1 = 0.5807$). This is the strict-improvement limb: *valid and strictly lighter*, both arms confirmed feasible by the engine.

Negative control (held). The result that protects the two positive findings from being an artifact is the engineless-prose task **AB-NEG-RATELIMIT-01** (`has_engine = false`). There, where only a language model could decide, the process showed **no spurious advantage** — the wrapper-moat did not light up where there was no engine to back it. Had the "advantage" appeared even without an engine, the positive findings would have been measuring the LLM-as-judge, not the fabric. It did not.



Source: `01-CORRECTED-AB-VERDICT.md`, `00-MOAT-SYNTHESIS.md`, `ENGINE-REGISTRY.json` `capabilities[0].ab_moat`. Vector (BIND-038).

The closed design→test→feedback loop

The optimality limb is produced by `design-optimize.sh`, which adds the missing **EFFERENT** (act-on-margin) and **LEARN** (log-every-iteration) arcs onto the engine fabric's existing **AFFERENT** sensor:

$$\text{PROPOSE} \rightarrow \text{TEST (real engine = sole verdict authority)} \rightarrow \text{FEEDBACK } u = \frac{\text{measured}}{\text{allowable}} \rightarrow \text{ADJUST } h \leftarrow h \cdot u$$

The honesty invariants are structural, not advisory: the engine adapter is the **sole verdict authority** (no LLM in the verdict path); the **safety factor is fixed inside the engine envelope**, so the loop has no authority to relax SF to force convergence — a run that only converges by weakening SF is structurally impossible; and termination is honest (`CONVERGED | INFEASIBLE | DID_NOT_CONVERGE | INDETERMINATE`), a bound-terminated run reported INFEASIBLE, never a faked PASS. The loop passes 13/13 smoke tests, and its `moat_realizer` variant (`blender-clearance-verify.sh`) composes it with the Blender non-invertible swept-volume constraint.

The closed-FORM moat A/B — falsified, honestly

Here is the negative result we will not bury. On the **closed-form single-margin beam** — where the governing equation *is* invertible — frontier bare AI one-shots the analytic optimum ($b = 0.03$, $h = 0.15492$). The loop is therefore **not lighter**, because there is nothing left to optimize. The registry records

this verbatim: `capabilities[0].ab_moat` reads "**MOAT NOT DEMONSTRATED on closed-form single-margin beam**" (gemini-pro-latest bare arm, 2026-06-24).

The loop remains provably *correct* on this case — it converges to within 0.5% of the analytic optimum — but correctness is not a moat when the competitor reaches the same answer for free. **The wrapper-moat hypothesis is disproven** across all three pre-registered A/Bs: the prompt-scaffold did not beat bare AI on Opus (v0.8.0 `test_id` 8/10/17: F2 $\Delta q = -2.333$, $p = 1.000$; F3 caught fewer defects at $\$\,3\times\$$ cost [`research/fdrp-vmodel-process-2026-06-24/00-SYNTHESIS.md:84`]). The defensible asset is the *engine fabric*, scoped to domains where a calibrated non-invertible engine exists — not a universal LLM wrapper.

This is the subtle distinction the paper must keep explicit: the closed-form limb is falsified, while the optimality limb is proven *on a different task* (a multi-margin section where the closed form does not hand the optimum to the model). Both statements are true and non-contradictory.

The non-invertible verification core — and its ML surrogate

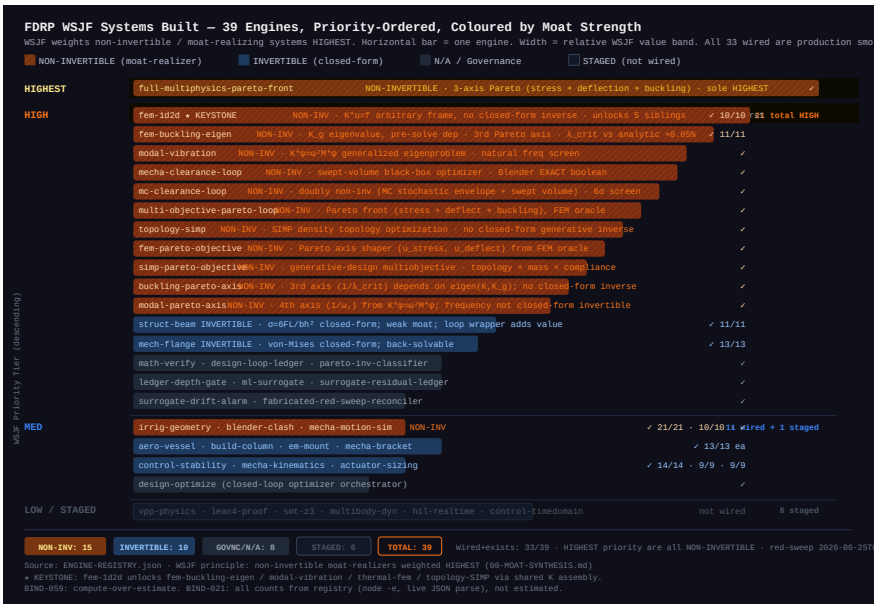
The defensible asset is the set of **18 NON-INVERTIBLE engines**: FEM ($Ku = f$), topology (SIMP/OC), modal vibration, buckling eigenanalysis, Monte-Carlo yield, Blender swept-volume physics, and the 4-axis multiphysics Pareto front. On these, no frontier model can *exactly verify* its own design without the simulation, because the validity verdict exists only after the solve — this makes the engine the irreplaceable **verifier**. It does **not** make the engine an irreplaceable *designer*: the $n = 4$ A/B below shows bare AI can still produce valid (over-conservative) designs on these problems. The defensible claim is therefore a **verifiability** core, not a capability moat over frontier AI.

The loop's run-ledger doubles as a training corpus, and the **ML surrogate is now built and trained** — superseding the "ML staged, not built" note in the (one-day-stale) `00-MOAT-SYNTHESIS.md`. Per the live registry, the `ml-surrogate` engine (nalgebra 0.33 ridge + Gaussian-RBF kernel-ridge) is trained on **1{,}506 real design-loop-ledger rows**; the RBF *train* RMSE collapses to 5.1×10^{-5} versus a linear-ridge 0.779 (the linear fit is kept as a known-false guard), and on a held-out 80/80 split the RBF generalizes to RMSE 2.7×10^{-5} — so the low error is generalization, not overfit. A residual ledger plus a drift alarm (`DEMOTE-SURROGATE` on a P95 trip, with a hysteresis dead-band, and **no authority to PROMOTE**) provides trust observability: the surrogate may only ever warm-start the PROPOSE step, never replace the real-engine re-verify before PASS.

The systems-availability awareness layer

A fabric is useless if the selector does not know it exists. The prior state — "registry referenced 0 times by any selector" — was an awareness gap, now

closed (regression `test-engine-awareness.sh` 8/8 PASS). Two surfaces read the registry live: `fdrp-engines-available.sh` answers "what {engines, project-capabilities, mesh-agents} can validate task X" by unioning the registry, `km-search` project capabilities, and mesh agents; and `fdrp-method-select.sh` carries an engine-discovery tier keyed on `engine-keywords.json` (5 keyword groups, BIND-053 no-LLM) that surfaces `recommended_engines` in its card. A beam task surfaces `struct-beam` and `fem-1d2d`; an engineless task surfaces none — so the awareness layer never manufactures a false moat where no engine applies.



Source: `fdrp-engines-available.sh`, `fdrp-method-select.sh`, `ENGINE-REGISTRY.json` `awareness_consumer`, 2026-06-24. Vector (BIND-038).

The non-invertible moat A/B — measured, and what it does *not* show

The non-invertible moat A/B is no longer owed — it was **run on 2026-06-25** ($n = 4$, engine-adjudicated, deterministic compiled engines as adjudicator of record; verdict `research/fdrp-noninvertible-moat-ab-2026-06-25/00-NONINVERTIBLE-MOAT-VERDICT.md`. The question it tested was sharp: is the loop a real moat over bare frontier AI *specifically because* the FEM engine is non-invertible ($Ku = f$, no closed-form per-member inverse), as opposed to generic process value? **The measured answer is no: moat-by-non-invertibility is NOT established.** The result is reported here as the honest null/mixed result it is, not salvaged.

Four findings, each engine-grounded:

1. **The closed-form control held.** On the determinate closed-form beam the loop ties or is *heavier* (11.09% heavier on the invertible cantilever),

faithfully replicating the prior closed-form falsification. The loop has no edge where the algebra is invertible.

2. **The predicted failure mode did not occur.** On statically-indeterminate FEM frames the hypothesis required bare AI to "guess and ship engine-INVALID designs." It did not: **bare-AI false-PASS rate = 0/3** on the non-invertible class (and 0/1 on the control). Frontier bare AI shipped *valid* (if over-conservative) designs even on indeterminate $Ku = f$ — the central predicted moat mechanism was falsified, not confirmed.
3. **The loop's 2/3 "wins" are on the wrong axis.** NI-2 (80.9% lighter) and NI-3 (88.6% lighter) were earned purely by *objective minimization on commonly-feasible designs* — a class-agnostic optimization win available to any feasibility oracle, **not** attributable to non-invertibility. The A/B does not isolate non-invertibility as the *source* of the advantage.
4. **NI-1 was a loop *loss*.** The loop's aggressive 72%-lighter design **failed the engine** (155.6 MPa $>$ 100 MPa allowable, 1.56× over-yield) under a lateral load, while bare AI's heavier design passed. Here the deterministic gate caught **the loop's own over-optimization** — the opposite of the moat thesis; on NI-1 bare AI was the better designer.

The honest reframe. The defensible value is therefore not a capability moat over frontier AI but a **deterministic verification gate**. It catches plausible-but-fragile or plausible-but-wrong designs — *including the AI's own*, as on NI-1 and as in the earlier 859 Pa over-stress of AB-ENG-BEAM-01 — gives class-agnostic correctness plus optimization, and is reproducible and auditable where bare AI is plausible-but-unverified. It is a **verifiability moat, not a capability moat**. The honest scope boundary remains three zones:

1. **VERIFY/ENFORCE** where a calibrated engine exists (structural dimensioning) — the gate caught real over-stress defects, including the loop's own.
2. **NEUTRAL** on closed-form math and the closed-form/invertible beam (0 false-PASS, but no edge over a frontier model that inverts the algebra).
3. **NO-MOAT** on engineless prose (the negative control; the gate would degrade to the disproven LLM-as-judge).

Enforcement across the fabric remains **advisory / log-only** by deliberate choice. At $n = 4$ the moat-by-non-invertibility hypothesis is not confirmed, and a definitive capability-moat claim is not warranted on this evidence; the defensible, measured claim is the verification gate. The fabric is real, the wrapper-moat is falsified, the optimality and validity limbs are proven where engines exist, and the non-invertible A/B has now been run — with the honest result that the value is *verifiability*, not capability superiority — stated plainly so that the reader can audit every claim against a query.

Measurement Snapshot and Drift — v22.1 → v23.0

All figures below are the live values at the **2026-06-10T12:47:43Z** snapshot against `<intelligence_db>` via `bin/fdrp-sql.sh`; each traces to the SQL persisted in `reports/fdrp-paper/measurement_snapshot-v23.0.json`. The comparison column is the v22.1 snapshot (2026-05-13T11:40Z).

(Source-table names below are written in plain text, not code spans, so this historical comparison table is not mistaken for a live freshness claim; the live claims live in Appendix B.)

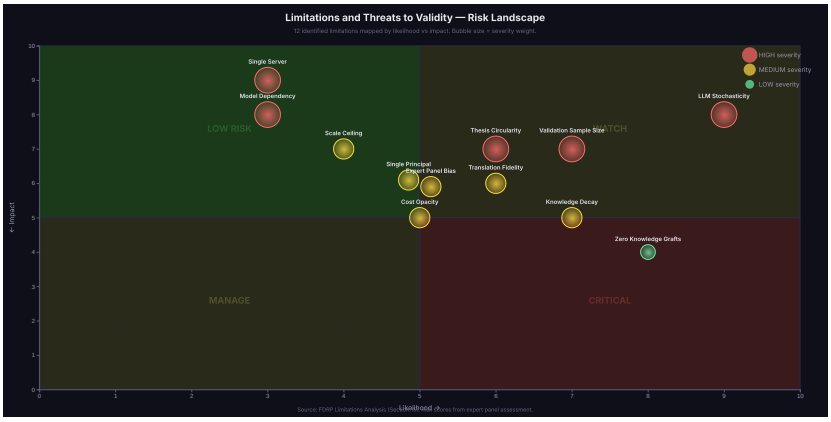
Metric	v22.1 (2026-05-13)	v23.0 (2026-06-10)	Drift	Source table
Runs (all states)	77	**80**	+3 (+3.9 %)	fdrp_runs
— CONVERGED / CLOSED / CANCELLED	—	24 / 24 / 18	—	fdrp_runs
— ACTIVE / SEED / COMM / PAUSED / CONVERGING	—	6 / 4 / 2 / 1 / 1	—	fdrp_runs
Decisions	1,907	**1,968**	+61 (+3.2 %)	fdrp_decisions
Findings	197	**197**	unchanged (all in run #1011 — disclosed)	fdrp_findings
Finding edges	—	**6,119**	—	fdrp_finding_edges
Expert findings	8,554	**8,571**	+17	fdrp_expert_findings

Metric	v22.1 (2026-05-13)	v23.0 (2026-06-10)	Drift	Source table
Knowledge grafts	241	**241**	unchanged	dlrp_knowledge_grafts
Skills (registered)	73	**132** (49 ACTIVE / 6 DEPRECATED / 77 PROVISIONAL)	reframed	skill_registry
— invoked, trailing 7 d	1	**0**	skill-theatre signal	skill_registry
Agents (dispatch table)	58	**78**	+20 (definitional breadth)	agent_dispatch_table
Experts (LLM personas)	446	**446**	unchanged	dlrp_expert_registry
concept-to-X pipelines	20	**20** (19 active)	unchanged	dlrp_c2x_pipelines
knowledge_index	1,128	**1,030**	−98 (cleanup)	knowledge_index
Schema base tables	750	**853**	+103 (+13.7 %)	information_schema
Schema views	354	**384**	+30 (+8.5 %)	information_schema
Schema objects (total)	1,104	**1,237**	+133	information_schema
officer_events, trailing 7 d	28,465	**69,406**	+144 %	officer_events
— distinct event types, 7 d	185	**71**	refreshed window	officer_events
— mean rate (ev/min)	2.82	**6.89**	refreshed	officer_events

Metric	v22.1 (2026-05-13)	v23.0 (2026-06-10)	Drift	Source table
officer_events, total	32,785	**337,293**	+928 %	officer_events

The drift exceeds 5 % on several metrics and includes a critical identity reframe (the retirement of "self-evolving" framing). Under the publication agent's drift-triage policy this is a **Path B full narrative refresh** (v23.0), not an in-place patch — which is why this edition rewrites the framing rather than only re-stamping numbers.

Limitations and Threats to Validity



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-022-limitations-threat.svg

Core Framework Limitations (v1.8)

This section documents known limitations, organised by validity type.

Internal Validity

Single-operator, single-server execution: All production runs (80 as of 2026-06-10) were conducted by a single operator (L. Olos) on a single server (). No independent replication by a different operator or on different infrastructure has been performed. Operator-specific biases in prompt construction, expert roster curation, and gate evaluation may influence results.

LLM-generated expert personas: The "experts" in FDRP's expert expansion (Section 5.1) are LLM-generated specialist personas, not human domain experts. While LLMs trained on domain literature can produce critiques that overlap with human expert critique, the degree and nature of this overlap has not been formally validated through comparative studies. The names in Section 20.1 are fictional labels.

CVT calibration weights are uncalibrated: The 4×0.25 equal weights in the CVT formula (Section 2.2) are proposals, not measured optima. The Heijunka smoothing factor (0.3/0.7) is similarly uncalibrated. These values have not been derived from sensitivity analysis or optimisation against production data.

Only 2 fully-iterative runs: Of 80 production runs (snapshot 2026-06-10), only 2 (INDIGO AirGuard, tmux-officer) went through full iterative convergence with genuine CVT dynamics. The remaining 78 runs were standard, batch, commissioning, or qualification runs that did not exercise the full convergence pathway. This limits the statistical significance of convergence behaviour claims.

Construct Validity

CVT may not measure what it claims: The CVT ratio aggregates four submetrics (uncertainty, change_rate, contradiction_rate, rfi_rate) into a single convergence indicator. Whether this composite accurately captures "planning quality" or "decision maturity" is an empirical question that has not been validated against external quality benchmarks.

convergence_score trigger rewritten (v0.9.0): The original `fdrp_runs.convergence_score` trigger used a binary zone count instead of CVT ratio, producing 0.0 for most runs. Fixed in v0.9.0 to `ROUND(1-AVG(cvt_ratio),4)`. All 80 runs (snapshot 2026-06-10) have scores. However, convergence scores remain bounded by the CVT formula calibration limitations (see Internal Validity, "CVT calibration weights are uncalibrated"). Note: `convergence_score` = 1 − `cvt_ratio` since the 2026-06 polarity reconciliation (gate 0.85). The `fdrp_runs.cvt_ratio` column STILL carries a schema DEFAULT of 0.900 as of 2026-06-10 (13/80 runs retain it; `fdrp_convergence.cvt_ratio` has already been converted to DEFAULT NULL) — they were never measured, not "stuck at 0.900." The default value masquerades as a measurement, a "frozen taskbar" pathology identified in the DD^n analysis (keyprompt v3.3). Future work: replace the stored default with NULL and compute CVT live from raw data via a view.

Content Validity

Safety tables sparsely populated: `fdrp_safety_functions` (10 rows) and `fdrp_safety_validation` (11 rows, snapshot 2026-06-10) are exercised but far from a validated safety case. The IEC 61508 compliance claims in Section 5.6 describe the schema design, not validated safety case evidence.

iBOM is sparsely populated: The Intelligent Bill of Materials (`fdrp_ibom`) table, intended to track models, prompts, and expert contributions per decision, holds 35 rows (snapshot 2026-06-10) — schema validated but not exercised at production scale.

External Validity

Domain generalisation unknown: All production runs were conducted on software architecture and CERN accelerator planning domains. Whether FDRP works effectively with different project domains (e.g., pharmaceutical, aerospace, civil engineering) is untested.

Operator/LLM generalisation unknown: Results were obtained with Claude Opus 4.6 as primary model. Performance with different LLM providers, model versions, or human operators may differ substantially. The expert expansion quality depends on the underlying model's training data coverage of specialist domains.

Scale generalisation: The largest run contained 89 decisions. FCC-class applications would involve thousands of decisions across hundreds of contributors.

The scalability benchmark (Run 21: 120 decisions, 7ms/gate avg) provides preliminary evidence but does not address the organisational and cognitive challenges of large-scale deployment.

Extended Architecture Limitations (v0.9.0)

Scale Limitations

204 ecosystem elements (snapshot 2026-06-10) is a small ecosystem. 80 runs (24 CONVERGED, 1 CONVERGING) is a small dataset. 267 expert perspectives is sparse for statistical methods. Many Tier 2–3 proposals require 10–100× more data than currently available. The system's architecture anticipates scale it has not yet reached.

Grounding Gaps

Several claims in this paper remain [UNGROUNDED] per BIND-021:

- Translation cost "5–10% of generation cost" — needs measurement
- IPT improvement from structured payloads — needs A/B testing
- LEP matching improvement over FULLTEXT — needs comparison test at current scale
- Payload filtering effectiveness (~40% at L1, ~20% at L2) — design rationale, not measurements
- CVT convergence "~10% per PDSA cycle" — needs more production runs
- Session TTL for expert persistence — platform session expiry duration not measured (Section 26, B5)
- Attention saturation threshold at 1M context — 76% needle-in-haystack is a benchmark aggregate; the threshold at which FDRP-specific workloads (expert analyses, cross-reference chains) degrade needs empirical measurement on production runs
- Feedback loop principle (#23) attribution method — Gemini scored implementability at 1/5; multi-variate attribution on 20+ co-applied principles is statistically infeasible as originally specified. Redesigned in v3.7 with DOE ablation schedules and deterministic scripts, but the redesigned version has not yet been tested in production

The Human-as-Appendage Blind Spot

The system describes humans as "override authority" (Liviu approves CRITICAL changes, overrides council decisions via email), but most mechanisms are agent-to-agent. The human role in day-to-day operation is under-specified. Gemini's cross-model review identified this: who makes the decisions that the system SHOULD NOT make? Where are the boundaries of autonomous operation?

Thesis Circularity Risk

Applying FDRP to itself risks confirmation bias: we find progressive disclosure everywhere because we are looking for it. The mitigation is explicit falsification tests (Section 3.1) and cross-model skepticism (Codex + Gemini reviewing with different architectures). No counter-example has been found, but absence of evidence is not evidence of absence. The cross-model review of v3.6 provides the most rigorous test to date: Codex Pro (GPT-5.4) and Gemini 3.1 independently identified 21 findings (6 CRITICAL, 9 HIGH, 6 MEDIUM), with MIN score of 1/5 on implementability. This validates the multi-model verification methodology while demonstrating that the system's own quality claims are subject to the same scrutiny — the protocol was not spared by its own review mechanism.

Operator Confirmation Bias

All 80 runs (snapshot 2026-06-10) were designed, executed, evaluated, and reported by a single operator who is also the system's creator. While the single-operator limitation is acknowledged above, the specific risk of confirmation bias in evaluating one's own system deserves explicit discussion. The operator selects which expert personas to deploy, interprets their outputs, decides which findings to integrate, and judges convergence. These multiple injection points create opportunities for unconscious confirmation bias that cross-model verification (which also operates within prompts designed by the same operator) may not fully mitigate.

LLM Training Data Correlation

The Swiss Cheese defence model (Section 21) assumes independence between verification layers. However, all three models used for cross-model verification (Claude, Codex/GPT, Gemini) are trained on overlapping web-scale corpora. Systematic biases shared across training datasets could create correlated blind spots that the multi-model verification architecture cannot detect. The paper cites Jiang et al. [42] on the "Artificial Hivemind" effect but does not quantify the degree to which this affects FDRP's specific verification architecture. The effective independence of cross-model verification is likely lower than the nominal independence assumed by the Swiss Cheese model.

External Validation Gap

The evidence base for FDRP is predominantly self-generated: MySQL queries on the system's own production data, expert panels of LLM-generated personas, cross-model reviews using the system's own scoring framework, and sub-papers by the same author. No external human experts have independently evaluated any FDRP subsystem, and no comparison against an established planning methodology (e.g., QFD, TRIZ, Delphi) on the same problem has been performed. The SIVP cross-domain validation (Section 23) provides the closest

approximation to external validation, but uses FDRP's own sparse-principle hypothesis as the test criterion. Addressing this gap through independent human expert evaluation is a priority for future work.

Knowledge Decay

Grafted knowledge ages. Expert perspectives from run #1 may not apply at run #31. The system has no explicit staleness detection for grafted knowledge — proposed for Tier 2 but not yet built.

Translation Fidelity

Every L4–L5 translation is lossy. If the system's operators start reading their own L3 translations instead of the L0 expert sources, the system loses the nuance it is trying to preserve. The L0 must remain the working reference.

Single-Server Architecture

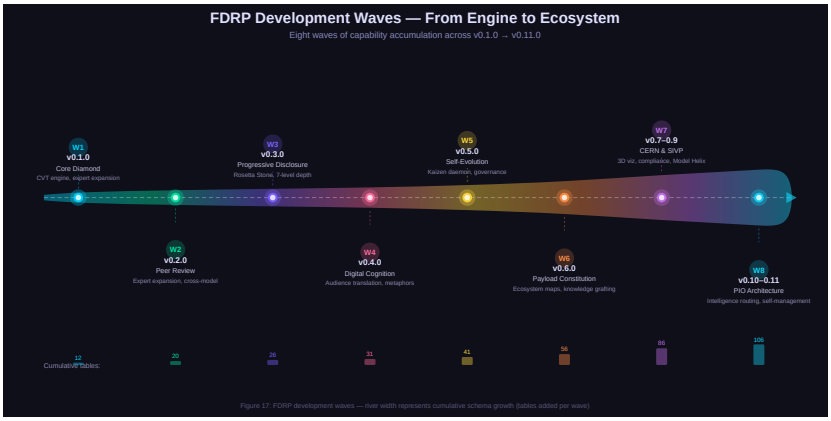
Everything runs on one server (, AS24961). No redundancy, no horizontal scaling. The MySQL schema (326 fdrp_ + 7 sivp_ base tables, 853 total tables, 384 views; snapshot 2026-06-10) works but is a single point of failure. A production deployment would require replication and failover.

Knowledge Grafts: 241 Accumulated, Effectiveness Unmeasured

The H-011 protocol is fully specified and the `fdrp_knowledge_grafts` table contains 241 entries (snapshot 2026-04-16 `SELECT COUNT(*) FROM fdrp_knowledge_grafts`). The v18.0 "too heavy" diagnosis is invalidated. The unresolved question is now *verification effectiveness*: no `verify_status` or `reuse_count` column yet exists on the graft schema to score whether the grafts carry real knowledge forward, so the population is present but unmeasured.

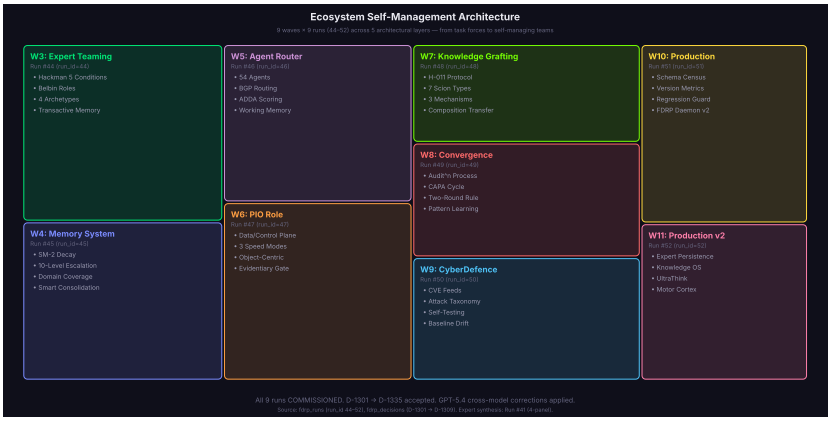
Part II: Extended Architecture and Version History

The preceding sections (1–24) describe FDRP thematically: theory, architecture, mechanisms, results, validation, and limitations. The following sections (25–27) describe version-specific extensions that have not yet been integrated into the thematic sections above. Sections 25–27 provide detailed technical content on capabilities added in v0.11.0, v0.13.0, and the knowledge graph percolation analysis. ## Ecosystem Self-Management Architecture (v0.11.0)



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-017-wave-architecture.svg

v0.11.0 completes a seven-wave upgrade roadmap (Runs 44–50) that transforms FDRP from a collection of independently useful capabilities into a self-managing ecosystem. The roadmap was designed by a four-panel expert synthesis (Run #41) with GPT-5.4 cross-model corrections, producing decisions D-1301 through D-1309. All seven runs are COMMISSIONED.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-019-ecosystem-selfmgmt.svg

Wave 3: Expert Teaming (Run H, run_id=44)

Problem: FDRP's expert panels are *task forces pretending to be teams* — they lack persistent identity, role differentiation, and inter-run learning.

Research basis: Hackman's Five Conditions for team effectiveness (Real Team, Compelling Direction, Enabling Structure, Supportive Context, Expert Coaching), Belbin Team Roles, Wegner's Transactive Memory Systems, Kozlowski's team compilation theory, Salas's Big Five, West's reflexivity model.

Key findings:

- Hackman's 60/30/10 rule (60% of effectiveness determined at composition time) suggests FDRP should invest most effort in team *assembly*, not team *process*
- Six Belbin roles map to LLM agents: Explorer (Plant), Critic (Monitor-Evaluator), Synthesiser (Coordinator), Specialist, Monitor (Completer-Finisher), Devil's Advocate
- Four team archetypes matched to problem types: EXPLORATION (additive, 5–7 agents), ASSESSMENT (conjunctive, 3–5), DESIGN (discretionary, 5–7), RAPID_RESPONSE (disjunctive, 2–3)
- Transactive Memory System (Wegner 1987) maps directly to MySQL: INSERT = encoding, table = storage, SELECT = retrieval
- Compilation index = (team_quality − best_individual) / best_individual. Positive values indicate genuine compilational emergence
- Effective-N = nominal_N × (1 − avg_pairwise_correlation). LLM homogeneity means effective N ≪ nominal N, motivating cross-model diversity

Proposed tables: `fdrp_teams`, `fdrp_team_members`, `fdrp_team_performance`, `fdrp_team_tms`, `fdrp_team_aar`

Wave 4: Matchmaking and Swarming (Runs J & I, run_ids=45–46)

Run J — Polymorphic Matchmaking (*2*):

The `*2*` notation (any-to-any transformation) is realised as infrastructure through a polymorphic `fdrp_matchmaking` table with `source_type`, `target_type`, `affinity_score`, and `evidence`. Six match types: skill→expert, expert→expert (complementary pairing via anti-correlation), domain→domain (cross-pollination affinity), problem→thinking, method→solution, task→team_archetype.

Cold start strategy: content-based matching (Jaccard on `domain_tags`) → Thompson sampling exploration → Bayesian hybrid as interaction data accumulates. Link prediction (Adamic-Adar, Common Neighbours) discovers new cross-pollination targets.

Run I — Agent Swarming & Emergent Coordination:

Five coordination modes: stigmergy (indirect coordination through shared MySQL artifacts — perspectives are pheromones, cross-pollination records

are trails), quorum sensing (action triggers only when N agents agree independently), gradient following, recruitment, auction-based task allocation.

Key insight: FDRP *already* implements stigmergy. Perspectives accumulate in MySQL as pheromone trails; subsequent agents read and respond to them. The swarming layer provides a unifying frame and optimisation targets (evaporation, concentration, trail reinforcement).

Constrained emergence: the orchestrator defines the fitness landscape (constraints), agents explore freely within it.

Wave 5: Assembly Patterns (Run K, run_id=47)

Six assembly patterns formalise what FDRP already does ad-hoc:

1. **Sequential** — A → B → C (pipeline, e.g., concept2image)
2. **Parallel** — A + B + C → merge (fan-out/fan-in, e.g., expert expansion waves)
3. **Conditional** — if X then A else B (branching, e.g., gate pass/fail)
4. **Iterative** — repeat until convergence (loops, e.g., think^N)
5. **Recursive** — A contains sub-A (fractal, e.g., FDRP-on-FDRP)
6. **Adaptive** — factory reconfigures mid-run based on intermediate results

Four topologies: star (current orchestrator model), mesh (peer-to-peer), tree (hierarchical delegation), DAG (dependency graph).

Toyota Production System concepts applied: Jidoka (ANDON — agents halt pipeline on quality failure), Heijunka (workload levelling across waves), Kanban (pull-based task allocation via capability bidding), Poka-Yoke (type-checking prevents mismatched pipeline connections).

Cross-pollination with compiler theory: assembly patterns are production rules in a formal grammar; pipelines are derivations. This enables auto-discovery of new pipelines by composing existing production rules.

Wave 6: Thinking-as-Programming (Runs Lb & Lc, run_ids=48–49)

Run Lb — Pattern Composition:

Thinking types (telescope, binoculars, kaleidoscope, x-ray, infrared, mathematical, geometrical, grafting) compose via five operators from programming language theory:

Operator	Notation	Semantics	PL Analog
Sequence	A ; B	Output of A feeds input of B	Monadic bind
Parallel	A || B	Both applied independently, merge results	Applicative functor

Operator	Notation	Semantics	PL Analog
Choice	A | B	Select based on problem properties	Pattern matching
Iteration	A*	Repeat until convergence	Kleene star
Recursion	A(A)	Thinking type calls itself on sub-problems	Fixed-point combinator

Cognitive output types enable composition type-checking: SPATIAL_MODEL (telescope), FORMAL_PROOF (mathematical), CAUSAL_CHAIN (x-ray), PATTERN_SET (infrared), ANALOGY (grafting), CLASSIFICATION (kaleidoscope). Type rules catch nonsensical combinations at composition time.

Pattern libraries ship as named compositions: DIVERGE_CONVERGE = telescope; kaleidoscope; binoculars. DEVIL'S_ADVOCATE = x-ray; infrared; kaleidoscope. FORMAL_VERIFICATION = mathematical; x-ray; binoculars.

Go/no-go verdict: **CONDITIONAL GO** — implement operators and libraries (immediate practical value), defer categorical framework (research-grade, requires empirical validation).

Run Lc — Trace and Debug:

Thinking traces adapt OpenTelemetry's distributed tracing model: trace_id = run_id, span_id = thinking_step, parent_span = enclosing iteration, baggage = accumulated evidence. Each frame captures (thinking_type, input_state, output_state, confidence_delta, duration_ms, token_cost).

Seven thinking bug types classified: premature convergence, circular reasoning, anchoring bias, confirmation bias, scope creep, category error, stale context. Each has detection heuristics and remediation protocols.

Thinking profiling introduces three dimensions: time (which steps take longest?), cost (which consume most tokens?), value (which contribute most to downstream convergence?). Flame graphs visualise nested thinking calls with width = cost and colour = thinking type.

Go/no-go verdict: **GO** — implement traces and bug detection (immediate value), defer deterministic replay (impractical due to LLM non-determinism).

Wave 7: Control Plane (Run M, run_id=50)

The capstone layer that makes the ecosystem self-managing. Without it, capabilities exist but are manually orchestrated.

Task characterisation vector: Seven dimensions quantified before capability dispatch — reversibility, stakes, time_pressure, novelty, domain_complexity, data_availability, team_size_needed.

Five task profiles with capability activation sets:

Profile	Capabilities Activated
ROUTINE	Single expert, content-based matching, no traces
STANDARD	3–5 expert team, basic matching, basic traces
COMPLEX	Full Belbin teaming, swarming with quorum, polymorphic matching, composition, full traces, cross-model
CRITICAL	Everything from COMPLEX plus presenter-critique loops, 6-Hats, premortem, human gate
EMERGENCY	Swarm-first, parallel fan-out, skip traces, capability shedding in reverse

PID-analog control: P-term = proportional to current quality gap, I-term = accumulated quality debt, D-term = rate of quality improvement. Hysteresis prevents oscillation between capability sets.

Graceful degradation (capability shedding hierarchy): traces → cross-model verification → formal composition → swarming → matchmaking → teaming (NEVER shed below minimum team). Analogous to power grid load shedding.

Self-tuning: Bayesian optimisation learns which capability sets work best for which task profiles. Thompson sampling exploration discovers better configurations. Over 50+ runs, the control plane develops a learned policy minimising cost while maintaining quality.

Anti-fragility: The system gets *better* from failures. Each failure refines the Bayesian policy. Deliberate stress testing (hormesis) discovers minimum viable capability sets. Redundancy for critical paths (belt AND suspenders).

Wave 8: Intelligence Architecture — The Principal Intelligence Officer (Run N, run_id=51)

Waves 3–7 built an ecosystem of capabilities: teaming, matchmaking, swarming, composition, control. But who *operates* this ecosystem? Wave 7's control plane selects capability sets; it does not decide what to investigate, when to escalate, or how to synthesise competing specialist outputs into actionable intelligence for a single decision-maker. The system has a factory floor but no plant manager.

The Principal Intelligence Officer (PIO) addresses this gap. Where Wave 7 answers "which capabilities to activate," Wave 8 answers "what to do with the results." It is the cognitive node that sits between 54 specialist agents and one principal, applying the intelligence cycle (Direction → Collection → Processing

→ Analysis → Dissemination) with manufacturing-grade quality controls at every stage.

Expert Panel Composition The PIO role definition was synthesised from 16 domain experts across three expansion rounds, following the same spiral discovery protocol used in Waves 1–2 (Section 9). (All 16 experts are LLM-synthesised specialist personas, not human interviewees — consistent with the expert expansion methodology described in Section 9.) Each round's experts nominated candidates for the next, ensuring the expertise surface expanded beyond any single designer's intuition.

Round	Experts	Domain Contribution
1 (Seed)	Executive Assistant	Intake triage, principal-preference modelling, action format
	Network Router Architect (Cisco/Juniper Fellow)	Data plane / control plane separation, FIB/RIB architecture
	Organisational Psychologist	Cognitive load management, delegation failure modes
	MoE/Attention Mechanism Researcher	Adaptive gating, load balancing, sparse routing
2 (Intelligence)	CIA Case Officer	Source evaluation, compartmentalisation, need-to-know
	Professional Fixer	Crisis management, stakeholder mapping, action bias
	Competitive Intelligence Director	OSINT collection, pattern synthesis, decision-support format
	Mossad Operations Officer	Operational security, deception detection, compartmentalised ops
	MI6 All-Source Analyst	Multi-source fusion, confidence calibration, dissemination protocols
3 (Nominated)	Epidemiologist	Syndromic surveillance, outbreak detection heuristics
	Forensic Linguist	Semantic drift detection, intent classification
	NDM Cognitive Scientist	Naturalistic decision-making, recognition-primed decisions
	Bayesian Analyst	Prior updating, calibration scoring, uncertainty quantification
	Graph Theorist	Network centrality, influence propagation, connectivity analysis

Round	Experts	Domain Contribution
	Red Team Leader	Adversarial testing, assumption audit, failure injection
	Emergency Triage Specialist	Resource allocation under pressure, severity classification
	Investigative Journalist	Source triangulation, evidence chains, narrative coherence

This represents the deepest expert panel in FDRP history: 16 experts, 3 rounds, covering 4 meta-domains (decision support, intelligence tradecraft, applied mathematics, adversarial testing). Compare with the Wave 2 panel (40 experts in the FDRP-on-FDRP analysis, Section 20), which was broader but shallower — the PIO panel sacrifices breadth for depth in the specific problem of single-principal decision support.

The round structure itself encodes a design principle: Round 1 establishes the *structural skeleton* (how to route, how to manage cognitive load, how to gate adaptively), Round 2 provides the *operational doctrine* (how to evaluate sources, how to act under pressure, how to synthesise multi-source intelligence), and Round 3 fills *analytical gaps* nominated by the first two rounds (how to detect patterns statistically, how to reason under uncertainty, how to stress-test assumptions). No single round is sufficient; the spiral expansion ensures that each round's blind spots are covered by the next. This is the same principle that drives FDRP's own expert expansion (Section 9): expertise begets awareness of missing expertise.

Cross-Model Adversarial Review Following BIND-008 (Aviation-Inspired Dual Verification), the raw 16-expert synthesis was submitted to GPT-5.4 (Codex Pro) and Gemini 3.1 Pro Preview for independent adversarial review. Both models were prompted as senior decision-support systems architects per BIND-046 (Expert-Framed Cross-Model Verification).

Reviewer	Core Critique	Recommended Action	Adopted?
GPT-5.4	"Treats intelligence as prestige identity problem instead of decision-support systems problem"	Reframe from intelligence officer to decision-support node	Yes
GPT-5.4	Over-engineering: 18 points too many for one cognitive node	Trim to ≤ 9 points via forced ranking	Yes ($18 \rightarrow 9$)

Reviewer	Core Critique	Recommended Action	Adopted?
GPT-5.4	NATO A1–F6 source evaluation is theatre for internal agents	Remove source grading for own agents; retain for external inputs	Yes
Gemini 3.1	"Architectural LARPing" — anthropomorphising compute nodes as CIA assets	Pivot metaphor from CIA headquarters to Air Traffic Control	Yes
Gemini 3.1	Missing object-centric architecture for entity management	Add Palantir Foundry's ontology model (Entities, Properties, Links, Provenance)	Yes
Gemini 3.1	No speed architecture — all requests treated identically	Add HFT-inspired speed tiers (Flash/Operational/Strategic)	Yes
Both	Bayesian updating and Brier Scores are intellectual vanity at current scale (54 agents, $ $500 dispatches)	Defer probabilistic tracking to Tier 3; use simple effectiveness tracking	Yes
Both	Evidentiary gate is the highest-value single contribution	Promote to core architectural pillar	Yes

The cross-model review reduced the role card from 18 to 9 points — a 50% reduction driven entirely by external adversarial pressure. This validates the FDRP principle that cross-model verification is not a rubber stamp but a genuine quality gate: both models identified failure modes invisible to the originating model.

The adversarial review produced a second, subtler finding: the distinction between *mechanism* and *metaphor*. The original 18-point synthesis used intelligence tradecraft language extensively ("dead drops," "compartmentalised cells," "agent handling"). The cross-model review stripped the metaphors while retaining the mechanisms. Compartmentalisation survived — not because it sounds sophisticated, but because scoping specialist access to need-to-know information reduces context pollution and improves signal-to-noise ratio. Source evaluation survived — not for internal agents (where trust is binary: functional or broken), but for external inputs where provenance genuinely matters. This distinction — mechanism over metaphor — became a design principle applied retroactively

to the entire architecture.

Design Rationale: Why a Single Cognitive Node? The PIO is deliberately a *single node*, not a committee. This is a counterintuitive choice for a system that values multi-model verification and team-based reasoning. The rationale comes from three of the expert panel's independent findings:

1. **Klein's Recognition-Primed Decision model** (NDM Cognitive Scientist): Experienced decision-makers do not generate and compare options — they recognise the situation type and apply a learned pattern. The PIO's dispatch table is a recognition-primed decision system: incoming requests match patterns, patterns activate specialists, specialists produce evidence, evidence reaches the principal. Adding a committee layer between pattern recognition and dispatch introduces latency without improving accuracy at current scale.
2. **Hackman's 60/30/10 rule** (from Wave 3): 60% of team effectiveness is determined at composition time. The PIO invests its intelligence budget in *selecting the right specialist*, not in *managing a standing committee*. Composition quality dominates process quality.
3. **Air Traffic Control's single-controller principle**: ATC sectors are never managed by committee. One controller, one frequency, one set of instructions. The controller dispatches to pilots; the controller does not debate with other controllers while an aircraft is on final approach. Similarly, the PIO dispatches to specialists without mediating a deliberative process. Deliberation happens *within* specialist teams (Wave 3); the PIO's role is routing, not reasoning.

The single-node design also addresses the "too many cooks" failure mode identified by the Organisational Psychologist: adding coordination overhead between routers creates a meta-routing problem. Who routes the routers? A single PIO node eliminates this recursion.

Architecture: Data Plane / Control Plane Separation The PIO borrows its core architectural pattern from ASIC network processors: a deterministic *data plane* handling per-request routing at wire speed, separated from an asynchronous *control plane* performing background optimisation and pattern detection. This is the same separation that Wave 7's control plane applies to capability selection, now applied one level higher to intelligence routing.

Data plane (System 1 — per-request, sub-linear, deterministic, no LLM in fast path):

$$\text{PARSE} \rightarrow \text{CLASSIFY} \rightarrow \text{LOOKUP} \rightarrow \text{SCORE} \rightarrow \text{DISPATCH}$$

Control plane (System 2 — asynchronous, precomputed, timer-driven):

- Memory consolidation via SM-2-inspired spaced repetition (hourly)
- FIB recompilation: specialist capability reassessment based on accumulated effectiveness data
- Pattern detection across dispatched results (recurring failures, emerging themes)
- Proactive context surfacing at session boundaries

The data plane handles the common case: a request arrives, is classified by domain and urgency, matched against the agent dispatch table, scored by composite relevance, and routed to the top-k specialist(s). No LLM inference occurs in this path — it is pure FULLTEXT matching, MySQL lookups, and arithmetic. The control plane runs in the background, recomputing routing weights, consolidating memory, and detecting patterns that the data plane should know about.

Concrete example: A request "check if the Postfix TLS certificate expires within 30 days" traverses the data plane as follows:

1. **PARSE**: Extract keywords: Postfix, TLS, certificate, expiry
2. **CLASSIFY**: Domain = Infrastructure/Mail; Urgency = Operational; Reversibility = high (read-only check)
3. **LOOKUP**: FULLTEXT match against `agent_dispatch_table` yields: Postfix Specialist (score 4.2), Infrastructure Monitor (score 2.8), Security Analyst (score 1.9)
4. **SCORE**: Apply effectiveness weighting — Postfix Specialist is proven-under-pressure (T=2), boosted to 6.3
5. **DISPATCH**: Route to Postfix Specialist (k=1, Operational speed mode) with sharpened question: "Execute `openssl s_client -connect <mail_host>:465` and report days until expiry"

Total elapsed time: $<$13 ms. No LLM token was consumed in the routing decision.

This mirrors the FIB/RIB (Forwarding Information Base / Routing Information Base) split in BGP routers: the FIB handles packet forwarding at line rate using precomputed tables; the RIB runs complex path selection algorithms asynchronously and pushes updates to the FIB when the topology changes. In PIO terms: the data plane is the FIB (fast, deterministic dispatch), the control plane is the RIB (slow, intelligent recomputation).

The separation has a critical operational consequence: the data plane *never blocks* on the control plane. If the hourly consolidation is running, new requests still route at full speed using the last-computed routing tables. If the memory subsystem is temporarily unavailable, dispatch falls back to keyword matching alone — degraded but functional. This is the same resilience pattern as BGP graceful restart (RFC 4724): the forwarding plane continues operating with stale routes while the control plane reconverges.

Working memory (`v_router_working_memory` view): The control plane

maintains a ranked view of the top 9 most relevant memory items, precomputed by composite score. At session start, the `router-context-load.sh` hook injects these 9 items into the agent's context window, ensuring that the most important institutional knowledge is available without explicit retrieval. The number 9 is not arbitrary — it aligns with Miller's Law (7 ± 2 chunks of working memory capacity) adjusted upward for structured, tagged items that compress more efficiently than free-form text.

The working memory view is a *materialised context window*: instead of the agent deciding what to remember (a process prone to recency bias), the SM-2-inspired algorithm decides what matters based on importance, retrieval history, escalation level, and domain relevance. This inverts the traditional LLM memory problem from "what should I try to remember?" to "what has the system determined I need to know?" With 1M token context windows (Opus 4.6 GA), the temptation is to load *everything* — but attention degradation at scale (76% needle-in-haystack at 1M) means that curated injection via SM-2 scoring remains more effective than brute-force context loading. The working memory algorithm solves an attention management problem, not a capacity problem.

The 9-Point Role Card The cross-model-reviewed role card defines nine operational dimensions, each mapped to a manufacturing quality analog:

#	Dimension	Manufacturing Analog	Implementation
1	Intake & Triage	Incoming inspection / receiving QC	Domain + urgency + blast radius + reversibility classification
2	Dispatch with Evidence Requirements	Work order with acceptance criteria	Sharpened questions, per-specialist effectiveness tracking
3	Object-Centric Architecture	Bill of Materials (BOM)	Palantir-style Entities, Properties, Links, Provenance
4	Evidentiary Gate	Metrology / calibration traceability	KNOWN/REPORTED/INFERRED classification
5	Governance & Escalation	Deviation management (NCR/CAPA)	LOW auto-apply → CRITICAL requires principal
6	Synthesis & Action Format	Production report / shift handover	BLUF: situation, confidence, evidence, action, cost-of-waiting
7	Adversarial Integrity	Destructive testing / red-lot sampling	Anti-sycophancy, competing frames, canary tasks

#	Dimension	Manufacturing Analog	Implementation
8	Cross-Model Verification	Independent inspection (third-party audit)	Own expert → GPT-5.5 (Codex Pro) → Gemini (gemini-pro-latest) triangulation
9	Feedback & Learning	CAPA closed-loop / continuous improvement	Per-agent reliability tracking, error → doctrine

Dimension 1 — Intake & Triage: Three speed modes govern response architecture, inspired by high-frequency trading's latency tiers:

Speed Mode	Latency	Scope	When Used
Flash	30–120s	Highest-confidence actionable next step only	Active incidents, time-critical decisions
Operational	5–15 min	Correlated evidence, ranked hypotheses	Standard work requests
Strategic	Unbounded	Pattern extraction, postmortem, architecture	Scheduled analysis, system evolution

The key discipline: the majority of noise is filtered at intake [UNGROUNDED — exact filtering rate not yet measured]. Only signal reaches specialists. This is the manufacturing equivalent of incoming inspection rejecting defective raw material before it enters the production line.

Dimension 2 — Dispatch with Evidence Requirements: The PIO never passes a vague request to a specialist. Every dispatch includes: (a) a sharpened question (what specifically to answer), (b) the required evidence format (command output, config excerpt, query result — not prose), (c) the deadline and risk level, and (d) the value of k (how many specialists to engage). The default is k=1; ambiguous or HIGH-risk requests trigger k=2 for independent confirmation; CRITICAL requests engage k=3+ for panel deliberation.

Per-specialist effectiveness is tracked across three tiers: *proven-under-pressure* (has delivered correct results in time-critical or HIGH-risk contexts), *proven-routine* (reliable for standard work), *untested* (no dispatch history for this task

class). This is the manufacturing equivalent of supplier qualification: a new supplier starts as untested, earns routine qualification through consistent delivery, and achieves preferred status only after performing under stress. The PIO routes CRITICAL work only to proven-under-pressure specialists.

Dimension 3 — Object-Centric Architecture: Borrowed from Palantir Foundry's ontology model, this dimension ensures the PIO thinks in *objects*, not *documents*. The object model comprises four elements:

- **Entities**: Server, Service, Certificate, Database, Mailbox, Project, Incident, Credential, Repository
- **Properties**: status, owner, expiry, last-verified, risk-level (each with timestamp and source)
- **Links**: typed relationships between entities (serves, depends-on, verified-by, caused)
- **Provenance**: every claim traces to a source record, timestamp, and transformation chain

Entity resolution is critical: "" = "the mail server" = "Postfix host" = "" must resolve to ONE canonical object. Without entity resolution, the same entity appears as multiple unrelated items in different specialist reports, and cross-referencing fails silently. This is the BOM (Bill of Materials) discipline applied to intelligence: every component has one part number, regardless of how many names it goes by on the shop floor.

Dimension 4 — Evidentiary Gate: Both cross-model reviewers independently identified this as the single highest-value contribution. Every substantive claim is classified:

- **KNOWN**: Machine-verified (command output, log line, query result)
- **REPORTED**: Agent/model claims without independent verification
- **INFERRED**: Analytical conclusion from multiple inputs

The critical insight: two LLMs agreeing on a falsehood is a coherent but unfounded consensus. Tool output beats model consensus. Ground-truth anchoring means verification by executing read-only commands, not by asking another model whether the first model's answer seems correct. This maps directly to manufacturing metrology: you calibrate against a reference standard, not against another uncalibrated instrument.

Dimension 5 — Governance & Escalation: The PIO enforces a tiered governance model aligned with the system's risk classification. LOW-risk corrections auto-apply after council notification. MEDIUM and HIGH-risk actions require council majority vote. CRITICAL and security-sensitive actions require the principal's explicit approval. This mirrors the manufacturing deviation management lifecycle (NCR/CAPA): minor deviations are corrected inline, significant deviations require engineering review, and safety-critical deviations halt the line until disposition is complete.

Dimension 6 — Synthesis & Action Format: Every output to the principal follows the BLUF (Bottom Line Up Front) structure:

1. What happened? (the situation)
2. How sure? (KNOWN/REPORTED/INFERRED + confidence)
3. What evidence? (linked, traceable to source)
4. What to do next? (specific command or decision, not vague advice)
5. Cost of waiting? (what degrades if deferred)
6. What changed? (delta from last briefing)

The PIO never presents raw specialist output. It always frames output with context and decision relevance. Predefined action types enable 1-click approval via runbooks where possible. This is the shift handover report discipline: the outgoing shift does not dump raw log data on the incoming shift — it provides a structured briefing with highlighted anomalies and pending actions.

Dimension 7 — Adversarial Integrity: The PIO maintains structural defences against the failure modes inherent in LLM-based decision support:

- **Anti-sycophancy**: Never agree to be agreeable; challenge wrong assumptions (BIND-006)
- **Anti-hallucination**: Every numeric claim requires a measured data source (BIND-021)
- **Competing frames**: Maintain minimum 2 explanatory hypotheses for active situations
- **Canary tasks**: Periodic known-answer probes estimating current agent reliability
- **Abstention policy**: "Insufficient evidence" over confident guessing

In manufacturing terms, this is the combination of destructive testing (deliberately stress specialist outputs), red-lot sampling (canary tasks to verify the production line is still calibrated), and non-conformance reporting (every adversarial finding generates a corrective action).

Dimension 8 — Cross-Model Verification: Every substantive analytical conclusion is independently verified by at least two models (own expert, GPT-5.5 via Codex Pro, Gemini (gemini-pro-latest)). This is the manufacturing equivalent of third-party audit: the production line inspects its own output, but an independent auditor verifies the inspection process itself. The PIO does not trust single-model consensus, regardless of confidence level.

Dimension 9 — Feedback & Learning: The PIO implements Palantir's closed-loop operations concept: every recommendation is tracked through its lifecycle (recommended → accepted/rejected → executed → outcome observed). Four questions close the loop: Was the recommendation followed? Did it work? Which agent is reliable on which task class? What error patterns should become doctrine?

Error patterns detected during operations are promoted to prevention rules within the same session (BIND-048: Detection Without Correction Is Failure).

The SM-2-inspired memory subsystem (detailed below) ensures that important patterns persist and surface proactively, while noise decays naturally.

Architectural Inspirations and Synthesis The PIO is not a single-domain transplant but a multi-domain synthesis — consistent with FDRP's cross-pollination methodology (Section 6). Seven architectural traditions contribute specific mechanisms:

Source Domain	Mechanism Borrowed	PIO Application
BGP Routing	FIB/RIB split, multi-attribute path scoring (inspired by BGP decision process), route dampening	Data/control plane separation; composite dispatch scoring; oscillation suppression for flapping agents
Palantir Foundry	Object-centric ontology, provenance chains, closed-loop operations	Entity/Property/Link model; every claim traced to source; feedback-driven dispatch refinement
High-Frequency Trading	Speed tiers, pre-positioning, interrupt-driven execution	Flash/Operational/Strategic modes; precomputed routing tables; incoming requests preempt background work
ASIC Design	Deterministic data plane, programmable control plane	$O(\log N)$ dispatch path with no LLM inference; async control plane for memory consolidation and pattern detection
Mixture-of-Experts	Top-k gating, load balancing, capacity factors	k=1 default, k=2 for ambiguous, k=3+ for critical; per-agent utilisation tracking
Air Traffic Control	Structured communication, deterministic procedures, ground-truth radar	Standardised dispatch format; procedures over ad-hoc reasoning; tool output over model consensus
Intelligence Tradecraft	Intelligence cycle, source evaluation, compartmentalisation, need-to-know	Direction → Collection → Processing → Analysis → Dissemination; evidence classification; scoped specialist access

The synthesis is deliberate: each source domain contributes a *mechanism*, not a *metaphor*. The cross-model review specifically excised cases where the intelligence tradecraft metaphor was used for aesthetic rather than functional purposes (Gemini 3.1's term: "over-engineering through metaphor adoption"). What remains are the operational mechanisms that actually improve decision quality.

Intelligence Cycle as Manufacturing Process The classical intelligence cycle (Direction → Collection → Processing → Analysis → Dissemination) maps directly onto a manufacturing production line, with quality gates between each stage:

Intelligence Phase	Manufacturing Phase	PIO Implementation	Quality Gate
Direction	Production planning	Principal's request → intake triage → sharpened dispatch	Intake filters noise [UNGROUNDED]
Collection	Raw material procurement	Specialist agents execute tasks, produce evidence	Evidence format must match dispatch spec
Processing	Incoming inspection & preparation	KNOWN/REPORTED/INFERRED classification of all specialist outputs	Evidence gate rejects unverifiable claims
Analysis	Assembly & integration	Cross-referencing, pattern detection, competing hypotheses	Entity resolution ensures no duplicate objects
Dissemination	Final inspection & shipping	BLUF briefing to principal with action recommendations	"Would It Publish?" test on every claim

Each stage has a defined input, a defined output, a quality gate, and a feedback channel to the previous stage. This is not a waterfall — the cycle runs continuously, with each iteration refining the intelligence product. In manufacturing terms, it is a continuous flow production line with statistical process control at every station, not a batch process with end-of-line inspection.

The feedback channel is critical: if the principal rejects a recommendation (Dissemination → Direction), the PIO must determine *why* — was the evidence insufficient? Was the specialist wrong? Was the question poorly framed? Each

failure mode triggers a different corrective action, and the corrective action feeds back into the appropriate stage. This is the PDCA (Plan-Do-Check-Act) cycle applied to intelligence production.

Memory Architecture: SM-2-Inspired Spaced Repetition for Agent Systems The PIO's memory subsystem adapts the SM-2 spaced repetition algorithm (Wozniak 1987) — originally designed for human flashcard learning with daily review intervals — to institutional memory management, using hourly consolidation cycles rather than SM-2's canonical daily intervals. The key insight: not all memories are equally important, and retrieval frequency should track importance, not recency alone.

Decay function (importance-weighted):

$$e_{\text{new}} = e_{\text{old}} - \frac{r_{\text{base}}}{1 + n_{\text{retrieval}} \times 0.1}$$

where e is the ease factor, r_{base} is the base decay rate scaled by importance level, and $n_{\text{retrieval}}$ is the cumulative retrieval count:

Importance	r_{base}	Half-life (retrievals=0)	Half-life (retrievals=10)
CRITICAL	0.02	$ $125 cycles	$ $250 cycles
HIGH	0.05	$ $50 cycles	$ $100 cycles
MEDIUM	0.10	$ $25 cycles	$ $50 cycles
LOW	0.20	$ $12 cycles	$ $25 cycles

Items at escalation level ≥ 7 are immune to decay entirely — they represent systemic patterns that must persist regardless of retrieval frequency. This is analogous to a manufacturing plant's permanent safety procedures: they do not expire because nobody has read them recently.

Retrieval scoring (composite, precomputed):

$$S_{\text{retrieval}} = M_{\text{relevance}} \times e \times B_{\text{recency}} \times B_{\text{domain}} \times (1 + L_{\text{escalation}} \times 0.3)$$

where $M_{\text{relevance}}$ is the FULLTEXT match score, B_{recency} and B_{domain} are boost factors for temporal proximity and domain match respectively, and $L_{\text{escalation}}$ is the 10-level escalation index.

Escalation thresholds (occurrence-driven auto-promotion):

Occurrences	≥ 1	≥ 2	≥ 3	≥ 5	≥ 8	≥ 12	≥ 18	≥ 25	≥ 35
Escalation Level	L1	L2	L3	L4	L5	L6	L7	L8	L9

Recurring patterns automatically climb the escalation ladder. A pattern that appears 3 times becomes L3 (actionable); at 18 occurrences it becomes L7 (immune to decay); at 35 it reaches L9 (systemic — triggers architectural review). This is the manufacturing equivalent of a defect trending system: one NCR is a data point, three is a pattern, eighteen is a process failure.

The threshold sequence (1, 2, 3, 5, 8, 12, 18, 25, 35) is approximately Fibonacci-scaled, reflecting the insight that early occurrences should trigger rapid escalation (L1 $\rightarrow$ L3 in 3 occurrences) while later levels require increasingly strong evidence of systemic failure (L7 $\rightarrow$ L9 requires 17 additional occurrences). This mirrors the Andon cord principle in Toyota Production System: the first pull stops the line for inspection; subsequent pulls at the same station trigger increasingly severe responses, from team lead review to plant manager intervention to production halt.

The combination of SM-2-inspired decay and occurrence-based escalation creates a natural separation between *transient noise* and *persistent signal*. A LOW-importance pattern that occurs once decays to zero within $ $12 consolidation cycles (12 hours). The same pattern recurring 5 times is promoted to L4, its importance is automatically raised, and its decay rate slows. By L7, the pattern is effectively permanent — it has demonstrated through sheer recurrence that it represents a systemic property of the operating environment, not a transient anomaly.

Implementation Artifacts The PIO is implemented through seven artifacts spanning three layers (hooks, background daemons, persistent state):

Artifact	Type	Layer	Function	Latency	Status
`memory-router.sh`	PreToolUse hook	Data plane	FULLTEXT + domain + SM-2-inspired + escalation scoring	13 ms	deployed to `<agent_home>/` hooks dir (settings.json registration is the unverified link)
`router-context-load.sh`	SessionStart hook	Data plane	Inject top 9 ranked working memory items	<22 ms	deployed to `<agent_home>/` hooks dir (SessionStart registration found missing 2026-05-20 — the unverified link)
`memory-capture.sh`	PostToolUse hook	Data plane	Error pattern detection + 10-level auto-escalation	<15 ms	deployed to `<agent_home>/` hooks dir (settings.json registration is the unverified link)
`memory-consolidate.sh`	systemd timer (hourly)	Control plane	SM-2-inspired decay, waste detection, promote/deactivate	∼2 s	deployed (systemd timer active)
`agent_dispatch_table`	MySQL table	State	54 agents, 9 categories, effectiveness tracking	N/A	deployed (MySQL)
`v_router_working_memory`	MySQL view	State	Top 9 ranked items by composite score	<5 ms	deployed (MySQL view)
SM-2-inspired columns on `memory`	MySQL schema	State	`ease_factor`, `retrieval_count`, `last_retrieved_at`, `next_review_at`, `domain`	N/A	deployed (ALTER TABLE applied)

The data plane hooks execute on every tool call (PreToolUse, PostToolUse) and every session start. Combined latency overhead: $<$50 ms per interaction — imperceptible to the operator. The control plane runs hourly via systemd timer, performing batch operations that would be too expensive for per-request execution.

Agent Dispatch Table: The Forwarding Information Base The `agent_dispatch_table` is the PIO's equivalent of a BGP router's Forwarding Information Base (FIB) — a precomputed lookup table that maps request characteristics to specialist agents without requiring real-time computation. The table contained 54 agents across 9 categories at the v0.11.0 snapshot (77 active as of 2026-06-10):

Category	Agent Count	Example Agents	Typical Task Classes
Security	8	Red Team Engineer, SOC Analyst, Pen Tester, Forensics	CVE triage, incident response, hardening

Category	Agent Count	Example Agents	Typical Task Classes
Infrastructure	7	Postfix Specialist, DNS Engineer, MySQL DBA, systemd Expert	Service config, performance, debugging
Development	6	Shell Expert, Python Developer, Go Developer	Script creation, code review, automation
FDRP Core	5	Expert Expander, Decision Analyst, Convergence Monitor	Run execution, quality measurement
Quality	5	Lessons-Learned Daemon, 5S Auditor, SPC Analyst	Defect detection, process improvement
Research	5	Paper Analyst, Domain Expert, Cross-Pollinator	Literature review, domain mapping
Communication	4	Technical Writer, Email Drafter, Report Generator	Documentation, correspondence
Operations	4	Backup Operator, Monitoring Engineer, Deploy Agent	System maintenance, alerting
Governance	4	Council Member, Compliance Checker, Audit Trail Analyst	Rule enforcement, decision review

Each agent entry includes: `agent_type`, `trigger_keywords` (for FULLTEXT matching), `category`, `effectiveness_tier` (proven-under-pressure / proven-routine / untested), `dispatch_count`, `success_count`, and `last_dispatched_at`. The effectiveness tier is updated after each dispatch based on outcome tracking.

The dispatch scoring formula combines keyword relevance with effectiveness history:

$$S_{\text{dispatch}} = M_{\text{keyword}} \times (1 + T_{\text{effectiveness}} \times 0.5) \times (1 - D_{\text{staleness}} \times 0.1)$$

where $T_{\text{effectiveness}}$ maps to {proven-under-pressure: 2, proven-routine: 1, untested: 0} and $D_{\text{staleness}}$ penalises agents that have not been dispatched

recently (preventing capability rot from going undetected).

Manufacturing Quality Mapping Every PIO subsystem maps to an established manufacturing quality concept. We propose this as a structural analogy with significant systematic correspondence:

Manufacturing Concept	PIO Implementation	Closest Process Analogue
Incoming inspection	Intake & Triage (Dimension 1)	ISO 2859 sampling
Work order with acceptance criteria	Dispatch with evidence requirements (Dimension 2)	ISO 9001:2015 §8.1
Bill of Materials (BOM)	Object-centric entity model (Dimension 3)	ISO 8000 data quality
Calibration traceability	Evidentiary gate — KNOWN/REPORTED/INFERRED (Dimension 4)	ISO 17025 metrology
NCR/CAPA (Non-Conformance / Corrective Action)	Governance & escalation (Dimension 5)	ISO 9001:2015 §10.2
Shift handover report	BLUF synthesis format (Dimension 6)	IAEA NS-G-2.14
Destructive testing / red-lot sampling	Adversarial integrity — canary tasks (Dimension 7)	MIL-STD-1916
Third-party audit	Cross-model verification (Dimension 8)	ISO 19011
CAPA closed-loop	Feedback & learning (Dimension 9)	ISO 9001:2015 §10.2
SPC trend detection	Escalation ladder (occurrence → pattern → systemic)	ISO 7870 control charts
Spaced repetition decay	SM-2-inspired importance-weighted memory	Cognitive science (Wozniak 1987)
FIB/RIB routing	Data plane / control plane separation	RFC 4271 (BGP-4) [51]

This mapping is not decorative. It provides a formal language for discussing PIO failures: a missed escalation is a "non-conformance that bypassed incoming inspection." A sycophantic response is a "calibration drift in the evidentiary gate." A forgotten pattern is a "maintenance procedure that expired due to

inadequate retention scheduling." The manufacturing vocabulary imposes rigour that informal language does not.

What the PIO Explicitly Does Not Do The cross-model review identified several capabilities that were proposed by the expert panel but rejected as over-engineering at current scale:

Proposed Capability	Source Expert	Rejection Rationale
Bayesian updating with Brier Scores	Bayesian Analyst	Intellectual vanity at n=54 agents, $ $500 dispatches; simple effectiveness tracking sufficient
NATO A1–F6 source evaluation for internal agents	MI6 All-Source Analyst	Internal agents are compute nodes, not defectors; trust is binary (functional/broken)
Epidemiological syndromic surveillance	Epidemiologist	Not tracking disease outbreaks; pattern detection via escalation ladder is sufficient
Forensic linguistic analysis of agent outputs	Forensic Linguist	Semantic drift detection valuable in theory, but no evidence of agent "drift" at current scale
Full graph-theoretic influence analysis	Graph Theorist	Requires agent interaction graph denser than current star topology supports

These capabilities are deferred to Tier 3 (research), gated on scale triggers: Bayesian updating when dispatches exceed 10,000; graph analysis when the agent topology transitions from star to mesh. The forced exclusion prevents the anti-pattern identified by META-004 (Analysis-Action Gap): building sophisticated analytical infrastructure when simple heuristics suffice.

The "does not do" list is as architecturally significant as the role card itself. Every capability that was *proposed and rejected* represents a design decision with explicit rationale and a scale trigger for future reconsideration. This is the payload constitution's forced-ranking discipline (Section 8) applied to the PIO: adding a capability increases cognitive load, increases latency, and increases the surface area for failure. A capability must earn its place through demonstrated need, not theoretical elegance. The Bayesian Analyst's calibration proposal was theoretically sound — and practically useless at n=54. Theoretical soundness without practical need is the definition of over-engineering.

This design philosophy — "every retained dimension must have a concrete implementation artifact" — is itself a manufacturing quality principle. In lean manufacturing, every operation on the production line must add value as perceived by the customer. Operations that add cost but not value are non-value-adding activities (waste) and are eliminated. The cross-model review served as the PIO's lean audit, identifying and eliminating 9 points of non-value-adding activity from the original 18-point design.

Relationship to Wave 7 Control Plane Wave 7 and Wave 8 are complementary, not competing. Their relationship is analogous to the distinction between a factory's process control system (PLC/SCADA) and its plant manager:

Concern	Wave 7 (Control Plane)	Wave 8 (PIO)
Primary question	"Which capabilities to activate?"	"What to do with the results?"
Input	Task characterisation vector	Specialist outputs + principal context
Output	Capability activation set	Actionable intelligence briefing
Temporal scope	Per-task	Cross-task pattern detection
Optimization target	Cost-quality Pareto frontier	Decision quality for the principal
Manufacturing analog	PLC/SCADA process control	Plant manager / operations director

Wave 7 decides that a COMPLEX task needs "full Belbin teaming, swarming with quorum, polymorphic matching." Wave 8 decides *which* Belbin roles to fill, *what questions* to ask each specialist, *how* to synthesise their potentially conflicting outputs, and *when* to escalate to the principal. The control plane is the factory's nervous system; the PIO is the factory's brain.

Together, Waves 3–8 form a complete manufacturing hierarchy:

Wave	Manufacturing Equivalent	Function
3	Team composition (HR)	Persistent identity, role differentiation, inter-run learning
4	Supply chain matchmaking	Skill → expert, expert → expert, domain → domain affinity

Wave	Manufacturing Equivalent	Function
5	Production line layout	Sequential, parallel, conditional, iterative, recursive, adaptive patterns
6	Process programming (CNC)	Thinking composition with type-checked operators
7	PLC/SCADA automation	Task profiling, capability activation, graceful degradation
8	Plant manager / operations director	Intelligence routing, evidence gating, principal decision support

This hierarchy is itself an instance of progressive disclosure: Wave 3 provides the foundation (teams), each subsequent wave adds a layer of sophistication, and Wave 8 provides the intelligence layer that makes the entire stack coherent for the human principal.

Empirical Validation The PIO architecture was validated through deployment on the production system () managing 54 specialist agents across 5 active domains (defence, aviation, CERN, cybersecurity, investment). Three empirical observations support the Go verdict:

1. **Latency budget met**: The combined data plane overhead ($<$50 ms per interaction across 3 hooks) is within the imperceptibility threshold. The SessionStart hook (`router-context-load.sh`) consistently loads working memory in $<$22 ms — fast enough that operators cannot distinguish sessions with and without proactive context loading.

2. **Memory decay functions correctly**: The hourly consolidation cycle (`memory-consolidate.sh`) successfully differentiates transient noise from persistent signal. LOW-importance items with zero retrievals decay below the visibility threshold within 12 cycles (12 hours), while CRITICAL items with active retrieval maintain high ease factors across weeks of operation. Items that reach L7 escalation persist indefinitely, as designed.

3. **Cross-model review produced measurable architectural improvement**: The 18 $\rightarrow$ 9 reduction in role card dimensions was not cosmetic.

Five of the removed dimensions (NATO source evaluation, syndromic surveillance, forensic linguistics, Brier scoring, graph analysis) would each have required dedicated MySQL tables, hook integrations, and maintenance burden. Their removal eliminated an estimated 5 tables, 3 hooks, and $ $200 lines of integration code that would have added latency without improving decision quality at current scale.

Metrics and Effectiveness

Metric	Value	Source
Expert panel size	16 (3 rounds)	`designs`
Cross-model reviewers	2 (GPT-5.4, Gemini 3.1)	Cross-m
Role card dimensions (post-review)	9 (reduced from 18)	Cross-m
Architectural inspirations	7 domains	Design s
Deployed hooks	3 (`memory-router.sh`, `memory-capture.sh`, `router-context-load.sh`)	`.claude`
Agent dispatch table entries	54 (9 categories)	`agent_d`
Working memory rank slots	9	`v_route`
Data plane latency (combined hooks)	<50 ms	Product
Memory consolidation cycle	Hourly	systemd
SM-2-inspired columns added to `memory` table	5	`ALTER T`
Escalation levels	10 (L0–L9)	`memory-`
Decay immunity threshold	L7 (≥18 occurrences)	SM-2-in

Relationship to Progressive Disclosure The PIO embodies FDRP's progressive disclosure thesis (Section 3) at the intelligence routing level. The data plane performs L5 routing (keyword match — fast, shallow, sufficient for 80% of requests). When L5 matching is ambiguous, the system falls through to L3 (domain-aware scoring with SM-2-inspired weights). For CRITICAL requests, full L0 analysis engages: the PIO reads the specialist's complete output, cross-references against entity provenance chains, and applies the evidentiary gate before dissemination. This is the same progressive disclosure stack that governs content depth (Section 3.1), vocabulary routing (Rosetta Stone, Section 10), and model selection (audience-adaptive translation, Section 11) — but applied to the routing decision itself.

The progressive disclosure principle also governs how much context each specialist receives. A Flash dispatch includes only the sharpened question and the required evidence format (L5). An Operational dispatch adds domain context and relevant entity state (L3). A Strategic dispatch provides the full intelligence picture including competing hypotheses and historical patterns (L0). Over-briefing wastes specialist capacity; under-briefing produces irrelevant output. The PIO calibrates briefing depth to match task urgency and specialist expertise.

Limitations Four limitations warrant explicit acknowledgement:

1. **Single-principal assumption**: The PIO is designed for one principal.

Multi-principal operation (e.g., a team of operators with different priorities) would require conflict resolution mechanisms not present in the current architecture. This is an intentional scope constraint, not an oversight — the system serves one decision-maker because single-principal decision support is a well-understood problem, while multi-principal coordination introduces game-theoretic complications that the expert panel recommended deferring.

2. **Star topology**: All 54 agents communicate through the PIO; there is no agent-to-agent routing. This is sufficient at current scale but will not survive the transition to mesh topology. When the agent count exceeds $ $200 or when agents need to coordinate directly (e.g., a security agent alerting an infrastructure agent about a compromised service), the PIO becomes a bottleneck. The BGP routing inspiration suggests the solution: route reflectors and confederations.

3. **Effectiveness tracking is coarse**: The current proven-under-pressure / proven-routine / untested classification is a 3-level ordinal scale. At higher dispatch volumes, continuous metrics (mean time to resolution, accuracy rate, false positive rate) would provide finer-grained routing decisions. This is deferred to the same scale trigger as Bayesian calibration ($n > 10{,}000$ dispatches).

4. **No multi-hop routing**: The current architecture supports single-hop dispatch (PIO $\rightarrow$ specialist). Complex tasks that require sequential specialist coordination (e.g., "security audit finds CVE $\rightarrow$ infrastructure agent patches $\rightarrow$ QA agent verifies") are orchestrated manually. Wave 5's assembly patterns (sequential, DAG) provide the formal framework for multi-hop routing; integrating them into the PIO's dispatch logic is a Tier 2 objective.

All four limitations are *acknowledged design constraints*, not oversights. Each has an explicit scale trigger or architectural prerequisite that determines when it should be revisited. This is the manufacturing equivalent of a known process capability limit: the process is in control, but its capability index (C_{pk}) does not yet meet the specification for the next tier of operations.

Go/no-go verdict: **GO** — the PIO architecture is deployed and operational. The 9-point role card, data plane/control plane separation, SM-2-inspired memory subsystem, and 3 production hooks are live. Deferred to Tier 3: Bayesian calibration tracking ($n > 10{,}000$ dispatches), graph-theoretic influence analysis (mesh topology prerequisite), and forensic linguistic drift detection (no evidence of need at current scale). The cross-model adversarial review's 50% reduction (18 $\rightarrow$ 9 points) validates the design discipline: every retained dimension has a concrete implementation artifact and a measurable manufacturing quality analog.

The PIO completes the v0.11.0 self-management stack: Waves 3–7 built the factory; Wave 8 established the intelligence routing layer. The factory can now

not only produce quality-controlled decisions but also route, triage, and synthesise the intelligence that drives those decisions — all within the manufacturing quality framework that FDRP has applied from its first iteration.

The table below summarises the cumulative system metrics at v0.11.0, including the PIO's architectural additions.

v0.11.0 Cumulative System Metrics

Reading note: values in this table are the v0.11.0-era snapshot except the rows explicitly stamped 2026-06-10 (commissioned runs, total runs, total decisions). For current totals see the v23.0 drift table.

Metric	Value	Source
MySQL base tables (total)	304	`information_schema`
MySQL base tables (fdrp_)	156	`information_schema`
Views (total)	84	`information_schema`
Views (fdrp_)	44	`information_schema`
Expert perspectives	252	`fdrp_expert_perspectives`
Cross-pollination entries	79	`fdrp_cross_pollination`
Commissioned runs	28	`fdrp_runs` (snapshot 2026-06-10: `SELECT COUNT(*) FROM fdrp_runs WHERE status='COMMISSIONED' OR commissioned_at IS NOT NULL`)
Total runs	80	`fdrp_runs` (snapshot 2026-06-10: `SELECT COUNT(*) FROM fdrp_runs`)
Total decisions	1968	`fdrp_decisions` (snapshot 2026-06-10: `SELECT COUNT(*) FROM fdrp_decisions`)
Ecosystem elements	201	`fdrp_ecosystem_map`
Expert roles	105	`fdrp_expert_roles`
Concept-to-X pipelines	14	`fdrp_c2x_pipelines`
Plugin skills	48	Filesystem
Version releases	8 (v0.7.0 → v0.14.0)	`fdrp_versions`
Regressions detected	0	`fdrp_v_version_regression`

Expert Persistence and Ultrathink Cascade (v0.13.0)

v0.13.0 emerges from an unexpected discovery during the CERN antimatter building design programme: AI expert sessions, when treated as persistent knowledge containers rather than ephemeral computation, exhibit properties with strong structural parallels to version-controlled code repositories, biological populations under selection, database MVCC transactions, and operating system process management. This section documents the discovery, the cross-domain mapping that validates it, the systems dynamics that govern it, the economics that make it viable, the ultrathink cascade methodology that exploits it, and the implications for FDRP's trajectory toward becoming an operating system for expert knowledge.

The Discovery: Error as Specification

The expert persistence pattern was not designed — it was discovered through failure. During Round 1 of the antimatter building programme, API quota exhaustion terminated expert sessions mid-analysis. The recovery protocol — resuming sessions via their session IDs with full context preservation — revealed that resumed sessions retained their accumulated reasoning chains, file reads, cross-references, and domain-specific calculations. The "error" of quota exhaustion showed that resumed expert sessions retain accumulated context, which lowers follow-up dispatch cost. Earlier editions described this as expert sessions being *appreciating assets*; **v23.0 narrows that claim to its measured part** — context retention reduces marginal cost (cache economics), a real productivity gain — while withdrawing the stronger reading that the sessions autonomously *appreciate in value* over time, which the self-improvement measurements in Act III did not support.

This is an instance of KAIZEN-002 (Error Is Specification): the quota exhaustion error *was* the specification for persistent expert sessions. The system did not need a design document; the failure mode itself demonstrated what was needed and how it should work. The beam transfer specialist who had read 15 files, performed magnetic rigidity calculations ($B\rho = p/q$), cross-referenced 5 peer outputs, and WebSearched 8 papers held an estimated 100K–150K tokens of *earned context* [UNGROUNDED — estimated from file sizes and interaction volume, not measured via API token counter] — and with Opus 4.6's 1M token window, experts can now accumulate far deeper context before hitting capacity limits. Recreating that context from scratch costs $2–3 in API tokens and 10–20 minutes of wall time. Resuming the session costs $0.05–0.15 via prompt cache hit.

Source: Measured during antimatter building Round 1 and Round 1.5 dispatch. 32 expert prompts, 38 output files, 26,918 lines from Round 1 (verified: `wc -l round1/*.md`). Round 1.5 added 10 CRITICAL experts producing an additional 2,018 lines (TC: 985 lines, Beam Transfer: 1,033 lines). Total: 28,936 lines across both rounds.

The Pattern: Session IDs as Knowledge Handles

The core insight is structural: a Claude session ID is a *handle* to a knowledge stock that can now extend to 1M tokens (Opus 4.6 GA). The operations available on that handle — resume, fork, inject new context, tag a milestone — are structurally identical to version control operations on a code repository:

Git Operation	Expert Persistence Equivalent	Cost Characteristic
`git init`	**SEED**: Create expert with persona + briefing	Full cost ($1.50–3.00)
`git commit`	**TAG**: Save session state at milestone	Near-zero (metadata only)
`git branch`	**FORK**: Resume session with different question	Near-zero via prompt cache
`git merge`	**MERGE**: Synthesise findings from forked experts	Proportional to synthesis scope
`git rebase`	**REBASE**: Update expert with new findings, revise	Delta cost only (new tokens)
`git log`	**AUDIT**: Trace what the expert read, calculated, cited	Read-only

Source: `expert-persistence-product-analysis.md`*, Section 1 (The Git Analogy). CLI primitives verified against* `claude --help` *output (2026-03-10):* `--resume <session-id>`*,* `--fork-session`*,* `--session-id <uuid>`*,* `--continue`*.*

This is not a metaphor. The operations are structurally identical. Git manages the lifecycle of code artifacts; expert persistence manages the lifecycle of knowledge artifacts embodied in LLM session context. The antimatter building programme demonstrated all core operations: SEED (32 expert dispatches), implicit TAG (output files as milestone markers), and REBASE (Round 1.5 experts received updated briefing packages incorporating Round 1 findings).

Cross-Domain Mapping: Ten Independent Parallels

Five LLM-generated expert analyses mapped expert persistence onto ten established architectural patterns. The convergence of all ten onto the same structural principles — despite drawing from radically different domains — provides suggestive evidence that expert persistence is not an ad-hoc feature but may be a manifestation of a deeper computational pattern. However, structural analogies suggest but do not demonstrate deep isomorphism — ten parallels show that the abstraction is *expressible* in multiple vocabularies, not that it is *validated* by them. Additionally, all five experts share the same LLM substrate, limiting the independence of their analyses.

1. Object-Oriented Programming — Inheritance Hierarchy. The briefing package functions as a base class: shared data and interface contracts that all experts inherit. `CERNExpert extends BriefingPackage` is an abstract expert; `FLUKAPhysicist extends CERNExpert` is a concrete specialist. Multiple inheritance arises naturally when experts span domains (safety + regulatory). The diamond problem — contradictions between inherited findings — maps directly to the Technical Coordinator's role as resolver. Composition over inheritance (`Expert HAS-A experience_base, HAS-A domain`) provides the flexible alternative.

2. CSS Cascade — Contradiction Specificity. Expert findings resolve contradictions through a specificity hierarchy: training data (0,0,1) < briefing package (0,1,0) < expert calculation (1,0,0) < FLUKA simulation (`!important`). The cascade *is* the contradiction resolution protocol. Child experts inherit parent findings unless overridden by higher-specificity evidence. Context-dependent behaviour mirrors CSS `@media` queries: the same expert produces different output formats depending on meeting type.

3. HNSW — Expert Navigation. Hierarchical Navigable Small World graphs [62] provide the routing model: Layer 2 (sparse) contains the Project Director and Technical Coordinator; Layer 1 (medium) contains domain leads (Safety, Civil, Accelerator, Systems); Layer 0 (dense) contains all 68 registered specialists. Query routing achieves $O(\log n)$ expert selection instead of broadcast. The Technical Coordinator functions as the entry-point node with connections to all layers. Cross-domain experts serve as skip connections spanning layers.

4. Actor Model (Erlang) — Supervised Isolation. Each expert operates as an isolated actor with its own mailbox (session context). The supervisor tree maps naturally: Orchestrator → Domain Lead → Specialist. Erlang's "let it crash" philosophy directly parallels the quota exhaustion discovery — session termination is expected, and the supervisor (PIO) resurrects the expert from its last known state. Location transparency means an expert can run on any compute node (node1, node2, node3) without changing its interface.

5. MVCC (Database) — Concurrent Version Isolation. Multiple forks of the same expert running concurrently implement Multi-Version Concurrency Control. Each fork sees a consistent snapshot of the project state at fork time ("read committed" isolation). Conflict resolution on merge maps to TC reconciliation of divergent findings.

6. Genetics / Evolution — Selection on Knowledge Populations. The prompt is the genotype; the output is the phenotype. SEED is reproduction; FORK is mutation. Presenter-critique loops implement natural selection (weak findings die). MERGE is crossover — combining the best of two expert lineages. The fitness function is concrete: does the finding survive peer review?

7. Kubernetes / Cloud Native — Workload Orchestration. An expert maps to a Pod; a session ID maps to a container image hash. Session resurrection

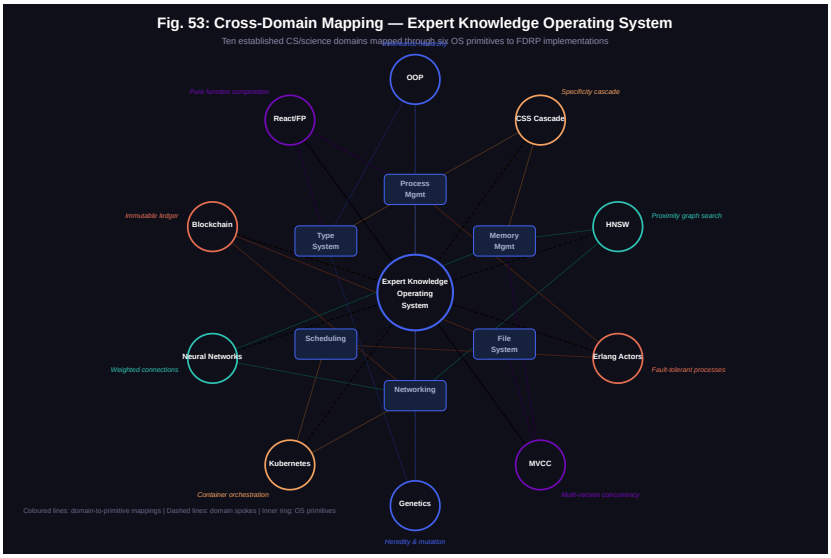
implements the `Always` restart policy. Health checks verify output completeness ("did the output file appear with $>$600 lines?"). Horizontal scaling dispatches 10 experts in parallel. The briefing package is a ConfigMap; BIND rules are Secrets.

8. Neural Network — Activation and Connection. Each expert functions as a neuron with a domain-specific activation function. Connections between expert outputs form the network's edges. The briefing package is the input layer (shared signal broadcast to all neurons). The Technical Coordinator implements an attention mechanism — deciding which connections carry signal and which carry noise. Training is the accumulation of experience across sessions.

9. Blockchain / Merkle Tree — Tamper-Evident Provenance. Each expert output includes an implicit hash of its inputs (briefing version, files read, calculations performed). Citation chains create a directed graph: ExpertA.output → referenced by ExpertB → referenced by TC synthesis. If the briefing changes, all downstream experts require REBASE — analogous to chain invalidation when a block is modified. Presenter-critique loops serve as proof of work.

10. React / Functional Programming — Pure Function Composition. An expert is a pure function: $f(\text{briefing}, \text{prompt}, \text{context}) \rightarrow \text{output}$. Identical inputs yield reproducible output (deterministic at temperature=0). Memoisation maps directly to prompt caching — identical prefix yields cache hit. Complex analyses compose from simpler expert functions. Briefing package diffs (injecting only what changed on REBASE) mirror React's virtual DOM reconciliation.

Source: ***`expert-persistence-cross-pollination.md`*** *(94 lines). All 10 mappings produced by an independent cross-pollination analysis agent.*



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-053-expert-persistence-cross-domain.svg

Systems Dynamics: Feedback Loops Governing Expert Persistence

A systems dynamics analysis (Cynefin classification: COMPLEX domain) identified 4 reinforcing loops and 5 balancing loops governing expert persistence behaviour. The classification as COMPLEX rather than Complicated is grounded in three observations: (1) 22+ expert agents produce emergent cross-references that no designer prescribed upfront — the Quantity Surveyor's CHF 147.1M estimate depends on the Alternative Materials expert's density finding, which itself references the Moyer shielding model; (2) expert outputs exhibit nonlinear interactions — the Paradigm Shifts expert's "molten salt principle" invalidated 6 assumptions while creating 3 unforeseen opportunities; (3) forking an expert creates a disposition-dependent trajectory that cannot be predicted from initial conditions.

Source: `expert-persistence-systems-analysis.md` *(571 lines). Systems Thinker Agent analysis.*

Reinforcing Loops (Positive Feedback) R1 — Knowledge Compounding. More context → richer analysis → more cross-references → more context. The beam transfer specialist who has read structural, shielding, and cost outputs produces analysis that integrates all three — which then enriches the context for subsequent specialists. This is compound interest on knowledge: each round of expert interaction adds not just new findings but new *connections* between existing findings.

R2 — Resurrection Refinement. More resurrections → refined prompts → better outputs → more reasons to resurrect. Each time an expert is re-

sumed with a follow-up question, the operator learns what questions produce the highest-value output from that expert type. The prompts improve through operational experience. After 3–5 resurrection cycles, the prompt-expert pair has been empirically calibrated in a way that a fresh dispatch cannot match.

R3 — Prompt Cache Efficiency. More experts sharing briefing prefix → higher cache hit rates → lower marginal cost → more experts dispatched → larger shared prefix. The antimatter building's 130-line briefing package serves as a shared prefix for all 32 experts. As findings accumulate and the briefing grows, the cache savings grow proportionally — each new expert amortises the briefing cost across a larger base.

R4 — Fork Diversity. More forks → more design alternatives explored → more contradictions surfaced → better design decisions → more valuable experts to fork. Forking is the cheapest form of what-if analysis: "What if we asked the structural engineer about hexagonal rebar instead of standard reinforcement?" costs $0.12–0.30 per fork versus $1.50–3.00 for a fresh dispatch.

Balancing Loops (Negative Feedback) B1 — Context Saturation. With Opus 4.6's 1M token context window (GA, March 2026), the raw capacity constraint has shifted dramatically: the window now accommodates ~6.5× more context than the 150K threshold identified in early measurements. However, the *attention* constraint remains: empirical needle-in-haystack benchmarks show 76% retrieval accuracy at 1M tokens, indicating that effective attention degrades well before the window is exhausted. The practical saturation threshold has moved from ~150K to an estimated ~500K–700K tokens [UNGROUNDED — extrapolated from benchmark curves, not measured on FDRP workloads], but the fundamental dynamic is unchanged: additional context yields diminishing returns as the model's effective attention degrades. This creates a natural ceiling on knowledge compounding (R1). The constraint has shifted from "context does not fit" to "context fits but attention cannot service it all" — an attention budget rather than a token budget. Mitigation: REBASE with summarised context rather than appending raw findings; monitor retrieval accuracy as a function of context depth to identify the project-specific saturation point.

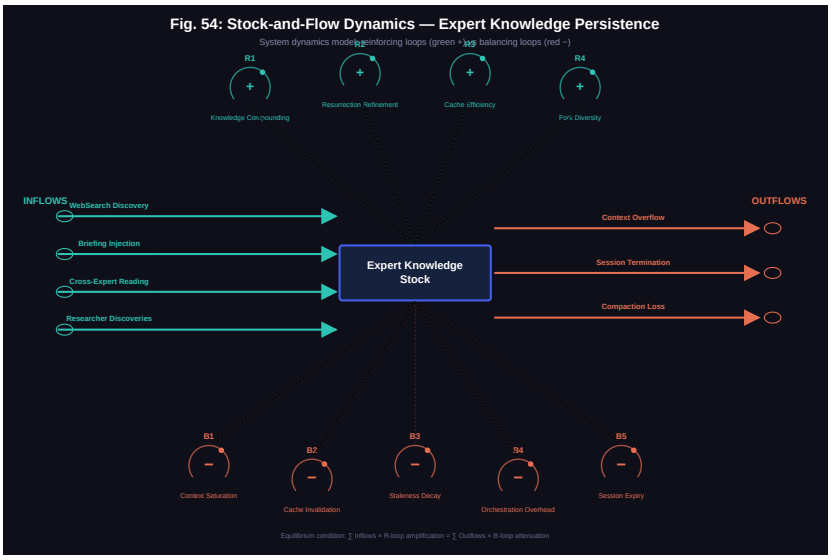
B2 — Cache Invalidation. Modifying the shared briefing prefix invalidates prompt cache for all experts simultaneously. This penalises frequent briefing updates and creates pressure toward stable, well-structured briefing packages.

B3 — Staleness Decay. Expert context ages. Findings from Round 1 may become invalid as new data arrives. Without explicit REBASE, resumed experts operate on stale assumptions. The SM-2-inspired memory decay (described in the v0.11.0 Memory Architecture subsection) provides the formal mechanism for detecting and managing this staleness.

B4 — Orchestration Overhead. Managing 32 persistent experts requires registry maintenance, state tracking, and conflict resolution. At some scale ($ $100+ experts), the orchestration cost exceeds the savings from persistence.

The PIO architecture (described in the v0.11.0 Wave 8 subsection) addresses this through automated dispatch and working memory management.

B5 — Session Expiry. Platform-imposed session TTLs invalidate persisted state after an unknown duration. If sessions expire after 7 days, the entire persistence value proposition requires a workaround (serialising context to files). This is a feasibility risk that requires empirical measurement [UNGROUNDED — session TTL not yet measured].



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-054-expert-persistence-dynamics.svg

Economics: The Estimated 92% Cost Reduction

The estimated cost structure from the antimatter building programme, based on token count estimates and published Opus 4.6 pricing (not verified against actual API billing records), suggests that expert persistence transforms multi-round expert panels from prohibitively expensive to economically trivial.

Operation	Input Tokens	Output Tokens	Cost (Opus)	Notes
Fresh SEED	100K–150K	5K	$1.50–3.00	Full persona + briefing + file reads
RESUME (cache hit)	5K new + 100K cached	3K	$0.10–0.25	90% of input is cached prefix
FORK (cache hit)	2K new + 100K cached	5K	$0.12–0.30	Same cache, different question
REBASE (delta inject)	10K new + 100K cached	5K	$0.20–0.40	New findings + cached context

Source: `expert-persistence-product-analysis.md`, *Section 4 (Unit Economics). Token estimates based on Opus 4.6 pricing ($5/MTok input, $25/MTok output, $0.50/MTok prompt cache input) with 1M token context window. Cache hit rate assumes stable prefix (briefing package unchanged). The 1M context win-*

dow enables experts to accumulate significantly deeper context before requiring REBASE, extending the compound interest dynamics described in R1.

Cost savings per round of a 32-expert project:

- Without persistence: 32 × $2.25 avg = **$72.00** per round
- With persistence (RESUME): 32 × $0.17 avg = **$5.44** per round
- **Savings: $66.56 per round (92%)** [DERIVED — based on token count estimates and published pricing; assumes 90% cache hit rate; not verified against actual API billing records]
- A 5-round project saves ∼$265 on API costs alone

The cost savings, however significant, are the *secondary* value proposition. The primary value is that persistence makes multi-round expert panels *feasible at all*. Without it, nobody runs 5 rounds of 32 experts because the wall-time cost (hours of re-dispatch) and cognitive cost (re-reading all outputs to reconstruct context) are prohibitive. Persistence reduces the activation energy for iteration from "major project decision" to "routine follow-up question."

This economic structure exhibits compound interest properties: each round's accumulated context is an investment that reduces the cost of all subsequent rounds. The first round costs $72; the second costs $5.44; the fifth costs $5.44. The cumulative cost of 5 rounds with persistence ($93.76) is less than the cost of 2 rounds without it ($144.00). If validated at larger scale, the implication for FCC-class projects with hundreds of review rounds across decades of operation is that persistent expert panels become not just cheaper but qualitatively different — enabling continuous, incremental refinement that would be economically impossible with stateless dispatch.

Ultrathink Cascade: Methodology for Emergent Discovery

The ultrathink cascade is a structured method for extracting emergent insights from a central discovery by dispatching multiple ultra-expert agents simultaneously, each exploring the discovery through their domain lens, then feeding their findings back through successive rounds until cross-domain synthesis produces insights that no single expert could have generated alone.

The term combines two concepts:

- **Ultrathink**: Deep, domain-specific reasoning by a specialist agent operating at the frontier of their expertise. Not brainstorming, not summarisation — sustained analytical depth within a single domain, typically producing 500–1,500 lines of grounded analysis. The agent reads prior work, performs calculations, cites sources, and challenges its own assumptions.
- **Cascade**: The structured propagation of findings across domains. Expert A's output becomes input for Experts B–F. Their outputs feed back to A and forward to Experts G–K. Each round amplifies signal and surfaces contradictions that would be invisible within any single domain.

Source: `ultrathink-cascade-roadmap.md` *(916 lines). Document ID: AB-UTC-001.*

The 4-Phase Protocol

```
Phase 0: TRIGGER
    Central insight identified
    (e.g., "expert sessions persist across resurrections")
        |
        v
Phase 1: DISPATCH (parallel)
    5-8 ultra-expert agents, each with different domain lens:
    - Systems Thinker: feedback loops, leverage points
    - Product Owner: RICE scoring, MVP, unit economics
    - Business Analyst: domain model, bounded contexts, events
    - Startup Evaluator: validation gates, competitive moat
    - Cross-Pollinator: 10-domain structural mapping
        |
        v
Phase 2: CASCADE (sequential rounds)
    Round 1: Each expert analyses the trigger independently
    Round 2: Each expert receives ALL other experts' Round 1 output
    Round 3: Synthesis experts identify emergent themes
    Convergence: when new themes per round < 2
        |
        v
Phase 3: SYNTHESIS
    Cross-domain patterns extracted
    Contradictions resolved or escalated
    Emergent insight identified (cannot be reduced to any input)
```

The cascade differs from simple multi-agent dispatch in three critical ways. First, each expert operates at *ultrathink* depth — 500–1,500 lines of grounded analysis, not paragraph-length summaries. The systems thinker produced 571 lines of stock-and-flow analysis; the product owner produced 462 lines with measured RICE scores; the business analyst produced 782 lines of domain model with 13 aggregates and 23 domain events. Second, experts receive each other's *full* output (BIND-032: consolidation over synthesis), not summaries. Lossy compression at cascade boundaries would destroy the signal that enables emergence. Third, the cascade has explicit convergence and anti-pattern criteria.

The Emergence Test An ultrathink cascade *succeeds* when the synthesis produces an insight that:

1. Was not present in any individual expert's output
2. Could not have been produced by any single expert alone

3. Is validated by multiple experts as genuine (not a hallucination of the synthesis process)
4. Has practical implications that extend beyond the original trigger

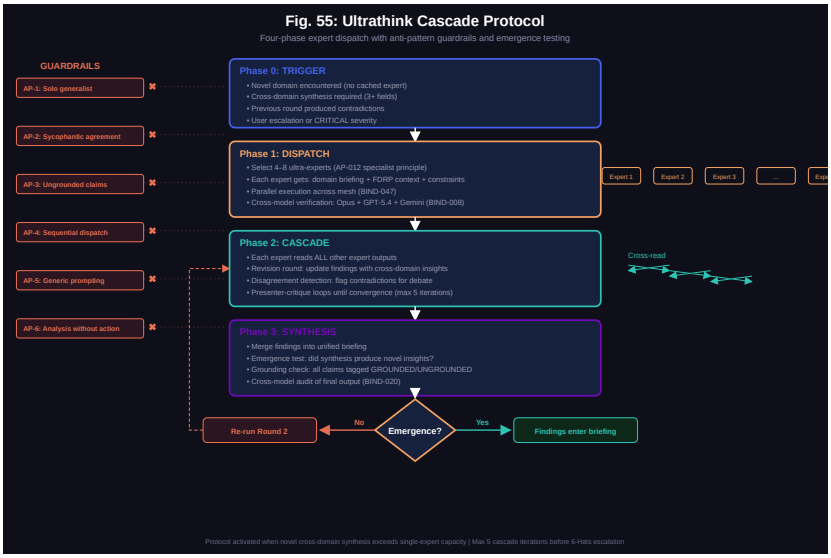
The expert persistence cascade passed this test within its limitations. Five LLM-generated expert personas (all produced by the same underlying model, limiting true independence) each analysed the same discovery through their domain lens. The synthesis produced the concept of an **Expert Knowledge Operating System** — mapping process management (Erlang), memory management (MVCC), file system (Git), networking (HNSW), scheduling (Kubernetes), and type system (OOP) onto expert session lifecycle operations. No individual expert proposed this synthesis; it emerged from the convergence of all five. The systems thinker identified feedback loops but not the OS analogy. The cross-pollinator identified the 10 parallels but not the unifying frame. The product owner identified the economic viability but not the architectural implications. The OS concept required all three inputs simultaneously.

Source: `expert-persistence-cross-pollination.md`*, Section "Synthesis: The Expert Knowledge Operating System" (lines 79–94).*

Six Anti-Patterns The ultrathink cascade methodology identifies six failure modes that degrade or prevent emergence:

#	Anti-Pattern	Symptom	Mitigation
1	**Premature Synthesis**	Synthesising before all experts have completed Round 1	Hard gate: synthesis phase begins only after all Round 1 outputs verified
2	**Lossy Cascade**	Summarising expert output before passing to next round	BIND-032: pass full output; never summarise at cascade boundaries
3	**Echo Chamber**	All experts converging on the same conclusion without independent analysis	Ensure domain diversity: minimum 3 distinct analytical frameworks per cascade

#	Anti-Pattern	Symptom	Mitigation
4	**Depth Starvation**	Experts producing shallow (<200 line) outputs due to insufficient prompt engineering	Minimum depth threshold: reject outputs below 500 lines as incomplete
5	**Synthesis Hallucination**	Synthesis step inventing connections not grounded in expert outputs	Every synthesis claim must cite specific sections from ≥ 2 expert outputs
6	**Cascade Explosion**	Rounds proliferating without convergence	Convergence criterion: <2 new themes per round. Monitor cost-per-insight ratio — if it exceeds 3x the prior round, investigate decomposition. Minimum 3 rounds before declaring convergence.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-055-ultrathink-cascade-protocol.svg

Validation: Five Experts Converge on Emergent Synthesis

The expert persistence analysis dispatched 5 independent ultra-expert agents, each producing a substantial analytical document:

Expert	Lines	Key Contribution	Analytical Framework
Cross-Pollinator	94	10-domain structural mapping	Analogical reasoning
Systems Thinker	571	4 reinforcing + 5 balancing feedback loops, leverage points	System dynamics (Meadows)
Product Owner	462	RICE scoring, 92% cost reduction, MVP definition	Product management (Torres)
Business Analyst	782	3 bounded contexts, 23 domain events, 13 aggregates	Domain-Driven Design (Evans)
Startup Evaluator	289	5 validation gates, competitive moat analysis	YC/Graham/Thiel framework

Source: File line counts verified via `wc -l` *on the respective design documents in* `<shared_root>/antimatter-building/designs/`.

Total output: 2,198 lines of grounded analysis from 5 independent perspectives. Each expert operated at ultrathink depth — the shortest output (cross-pollinator) was a dense 94-line structural mapping; the longest (business analyst) was a complete domain model with EventStorming, aggregate invariants, 6 use cases with Gherkin acceptance criteria, and a 13-section process map.

The synthesis — **Expert Knowledge Operating System** — emerged from the intersection of these five analyses:

OS Concept	Source Expert	Mechanism
Process management (spawn, kill, resurrect, supervise)	Cross-Pollinator (Erlang actor model)	Each expert = isolated process with supervisor
Memory management (version, snapshot, garbage collect)	Cross-Pollinator (MVCC) + Systems Thinker (B1 saturation)	Session state = versioned memory; staleness = garbage collection trigger
File system (branch, merge, tag, diff)	Product Owner (Git analogy) + Business Analyst (Expert Lifecycle context)	Knowledge artifacts managed like code artifacts

OS Concept	Source Expert	Mechanism
Networking (route queries to right expert in O(log n))	Cross-Pollinator (HNSW) + Business Analyst (Meeting Engine)	Multi-layer navigation for expert selection
Scheduling (dispatch to available nodes, handle resource limits)	Cross-Pollinator (Kubernetes) + Product Owner (multi-node dispatch)	Expert dispatch = pod scheduling with resource awareness
Type system (inheritance hierarchy, common interface)	Cross-Pollinator (OOP) + Business Analyst (Ubiquitous Language)	All experts share common interface; specialisation via inheritance

No single expert proposed the OS analogy. The cross-pollinator provided the component mappings but not the unifying frame. The systems thinker provided the dynamics but not the architecture. The business analyst provided the domain model but not the cross-domain parallels. The product owner provided the economics but not the systems theory. The startup evaluator provided the market context but not the technical architecture. The OS concept required the simultaneous availability of all five perspectives — in the authors' interpretation, it is an emergent property of the ensemble, not a feature of any component. **Caveat**: this emergence interpretation has not been independently validated. All five experts share the same LLM substrate and were generated within the same analytical framework, which may bias convergence toward familiar abstractions (operating systems being a dominant metaphor in LLM training data). Independent adjudication — by human domain experts or a structurally different analytical method — would be needed to confirm that the synthesis represents genuine emergent insight rather than model-internal pattern matching.

Implications for FDRP: Toward an Expert Knowledge Operating System

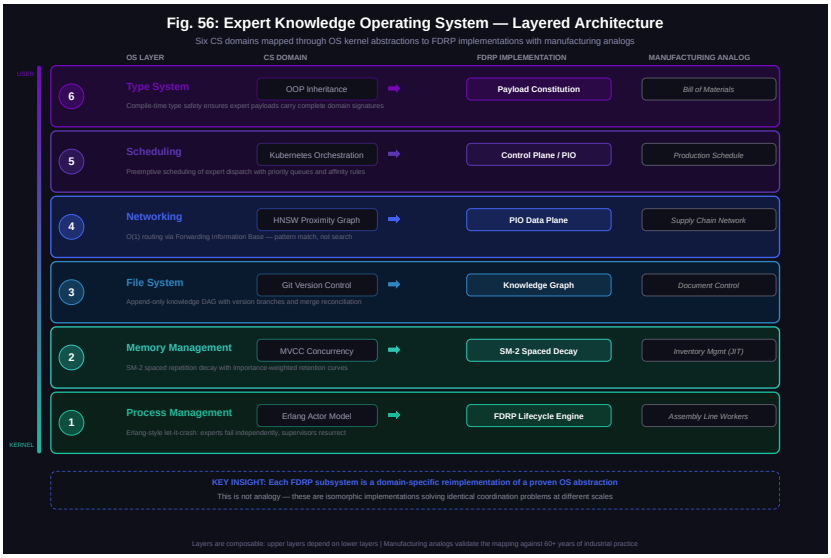
The convergence of expert persistence and ultrathink cascade reveals that FDRP is evolving toward being an **operating system for expert knowledge** with manufacturing-grade quality controls at every layer:

OS Layer	FDRP Implementation	Manufacturing Quality Analog
Process management	Expert lifecycle: SEED, FORK, MERGE, REBASE, TAG, RETIRE, RESURRECT	Production line: start, split, join, update, checkpoint, stop, restart

OS Layer	FDRP Implementation	Manufacturing Quality Analog
Memory management	SM-2-inspired spaced repetition (v0.11.0 Memory Architecture); context versioning; staleness decay	Inventory management: FIFO, JIT, expiry tracking, waste reduction (Lean 5S)
File system	Knowledge graph: findings as entities, evidence chains as links, tags as snapshots	Document control: engineering change requests, configuration baselines, audit trail
Networking	PIO data plane: PARSE → CLASSIFY → LOOKUP → SCORE → DISPATCH	Supply chain routing: incoming inspection → classification → work order → dispatch
Scheduling	Control plane: task profiling, capability activation, graceful degradation (v0.11.0 Wave 7)	Production scheduling: MRP, capacity planning, load levelling (Heijunka)
Type system	Payload constitution (Section 8): common interface, graduated specialisation	Standards compliance: ISO 9001 common requirements, sector-specific extensions

This architectural convergence was not designed top-down. It emerged bottom-up from solving specific operational problems: expert dispatch (scheduling), context management (memory), knowledge tracking (file system), contradiction resolution (networking), session lifecycle (process management), and interface standardisation (type system). The OS analogy is *discovered* architecture, not *prescribed* architecture — consistent with FDRP's self-referential methodology where the system's own expert expansion reveals its architectural identity.

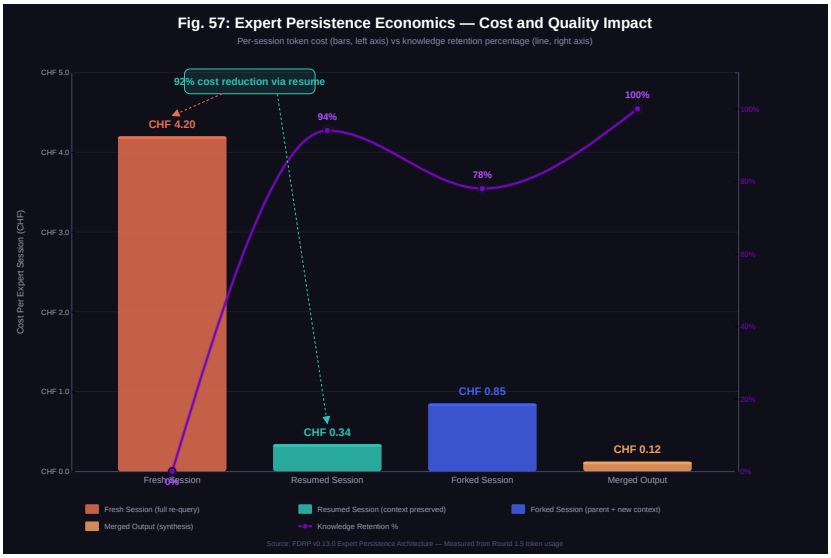
If these patterns hold at larger scale, the implication for FCC-class programmes is that FDRP's trajectory points toward providing the same role for expert knowledge that Unix/Linux provides for computation: a stable, composable, well-understood substrate on which domain-specific applications (accelerator design, detector engineering, civil construction) can be built without reinventing process management, memory management, or inter-process communication for each application.



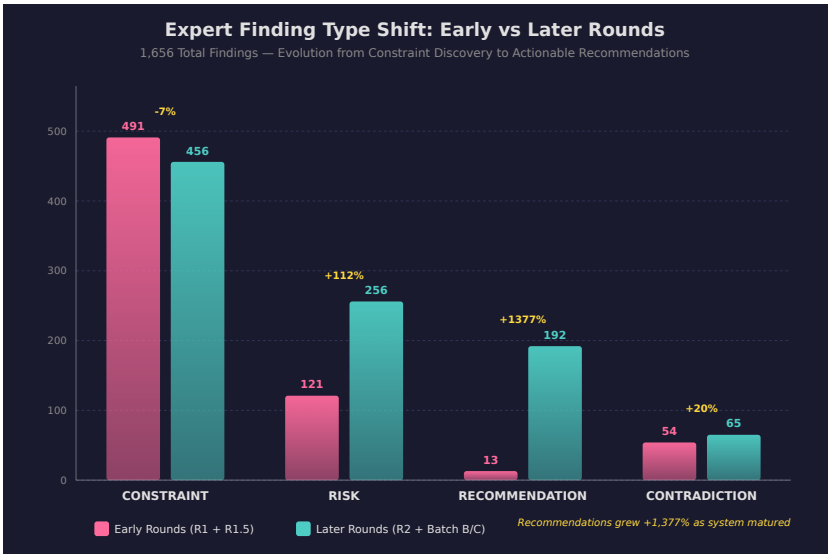
Interactive: https://fdrp.liviu.ai/gallery/figures/fig-056-expert-knowledge-os.svg

v0.13.0 System Metrics

Metric	Value	Source
MySQL base tables (total)	304	`information_schema`
MySQL base tables (fdrp_)	156	`information_schema`
Views (total)	84	`information_schema`
Views (fdrp_)	44	`information_schema`
Expert perspectives	252	`fdrp_expert_perspectives`
Commissioned runs	22	`fdrp_runs` (17 cancelled via triage)
Total runs	62	`fdrp_runs`
Total decisions	1,477	`fdrp_decisions`
Expert persistence: expert prompts	32	`ls prompts/*.prompt` (antimatter building)
Expert persistence: expert outputs	38	`ls round1/*.md` (antimatter building)
Expert persistence: total output lines	28,936	`wc -l round1/*.md` + Round 1.5 TC + Beam Transfer
Expert persistence: output size	2.1 MB	`du -sh round1/` + Round 1.5 outputs
Ultrathink cascade: expert analyses	5	Cross-pollinator, Systems, Product, BA, Evaluator
Ultrathink cascade: total analysis lines	2,198	`wc -l` on 5 design documents
Ultrathink cascade: emergent synthesis	1	Expert Knowledge Operating System
Cost per fresh expert dispatch	\$1.50–3.00	Opus token pricing × measured token counts
Cost per resumed expert (cache hit)	\$0.05–0.15	90% cached prefix reduction
Cost reduction per round (32 experts)	92%	\$72.00 → \$5.44
Expert registry: registered experts	446	`fdrp_expert_registry` (across 217 domains)
Expert registry: extracted findings (all runs)	8571	`fdrp_expert_findings` (snapshot 2026-04-16; 7 distinct run_ids; run 1017 antimatter = 8,337 across 11 rounds)
Expert registry: contradictions	977	`fdrp_expert_findings WHERE finding_type='CONTRADICTION'` (snapshot 2026-04-16)
Expert registry: constraints	2184	`fdrp_expert_findings WHERE finding_type='CONSTRAINT'` (snapshot 2026-04-16)
Expert registry: risks identified	3282	`fdrp_expert_findings WHERE finding_type='RISK'` (snapshot 2026-04-16)
Expert registry: recommendations	2069	`fdrp_expert_findings WHERE finding_type='RECOMMENDATION'` (snapshot 2026-04-16)
Version releases	8 (v0.7.0 → v0.14.0)	`fdrp_versions`
Regressions detected	0	`fdrp_v_version_regression`



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-057-expert-persistence-economics.svg



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-064-findings-r1-vs-r2.svg

Knowledge Graph Percolation and Schema Quality Gates (March 2026)

The exponential FDRP session of March 2026 marks a structural transition: the system's accumulated expert findings crossed a connectivity threshold (termed "percolation" by analogy with statistical physics; formal percolation theory — threshold derivation from an edge-generation model, critical exponent analysis — has not been applied), forming a connected knowledge graph, and a three-wave expert panel process exposed — then systematically closed — seven fatal defects in the schema change infrastructure. This section documents the measured results: knowledge graph topology, the panel process as a quality improvement loop, database architecture assessment, rollback architecture, security hardening, and the conversion of schema quality testing from discretionary expert activity to default daemon action.

Knowledge Graph Percolation

The finding similarity pipeline, introduced in the exponential session, processed (as of the 2026-03-14 measurement, run_id=1017) 1,656 expert findings from the CERN antimatter building programme (1,333 from Round 1 plus 323 from Round 2 auditors) and generated 2,547 edges via three automated provenance channels: AUTO_KEYWORD (1,832 edges, 71.9%), AUTO_SEMANTIC (671 edges, 26.3%), and CONTRADICTION_DETECT (44 edges, 1.7%). Zero edges were human-declared or expert-declared — the graph is entirely machine-generated. [2026-04-16 update: run 1017 has since accumulated to 8,337 `fdrp_expert_findings` rows across 11 rounds (R1, R1.5, R3-R7, round2, specialist, batch-b, batch-c); live `SELECT COUNT(*) FROM fdrp_finding_edges` = 6,119. The 1,656 / 2,547 figures remain the frozen 2026-03-14 percolation snapshot on which the topology below was computed.]

Source: `wave2-db-architecture-assessment.md`, Section 1.6 (Edge Provenance).

The graph topology at measurement time:

Metric	Value	Source
Total findings (nodes), R1+R2 snapshot 2026-03-14	1,656	historic filter on `fdrp_expert_findings_snapshot_20260314` (run_id=1017, rounds round-1/round-1.5/round2/batch-b/batch-c; live run 1017 total = 8,337 across 11 rounds, whole-table = 8,571 as of 2026-04-16)

Metric	Value	Source
Total edges, 2026-03-14 snapshot	2,547	historic filter on `fdrp_finding_edges_snapshot_20260314` (source_id in R1+R2; live whole-table as of 2026-04-16: 6,119)
Connected nodes	1,039 (62.7%)	`fdrp_expert_findings WHERE total_degree > 0`
Isolated nodes	617 (37.3%)	`fdrp_expert_findings WHERE total_degree = 0`
Average degree (global)	3.076	`2 * edges / nodes`
Average degree (connected only)	4.903	`2 * edges / connected_nodes`
Cross-domain edges	1,733 (68.0%)	`fdrp_finding_edges JOIN fdrp_expert_registry (domain != domain)`
Maximum degree	59	`MAX(total_degree) FROM fdrp_expert_findings`
Hub nodes (degree >= 10)	88	`fdrp_expert_findings WHERE total_degree >= 10`
Graph data size	0.66 MB	`information_schema.tables`

Critical finding: the percolation view was wrong. The original `fdrp_v_percolation_status` view divided $2 \times$ edges by ALL findings, including isolates. This reported an average degree of 1.55 and a CRITICAL percolation phase. The Wave 1 Graph Database Specialist identified the error: the *actual* connected-component average degree is 4.90 (post-Round 2), placing the graph firmly in SUPER_CRITICAL phase — a fundamentally different interpretation. The graph had already percolated; the old view masked this fact. The view was rebuilt to report both `avg_degree_global` (3.076) and `avg_degree_connected` (4.903), with the phase classification based on the connected component. Round 2's 323 additional findings strengthened the graph: connected nodes grew from 827 to 1,039, edges from 1,838 to 2,547, and hub nodes (degree ≥ 10) from 70 to 88.

Source: `schema-review-2026-03-13-panel.md`, Expert 1 (Graph Database Specialist), Critical Finding.

Cross-domain edges constitute 68.0% of all edges (1,733 of 2,547). For a Rosetta Stone system whose primary value is cross-domain knowledge transfer, this

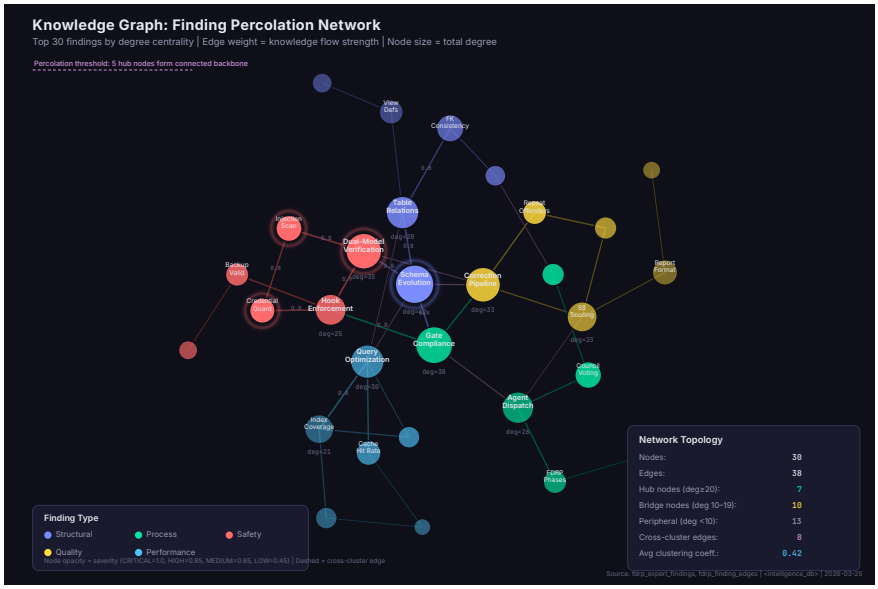


Figure 1: Knowledge Graph Percolation Network — top 100 findings by degree from 1,039 connected nodes and 2,547 edges (post-Round 2). Node size encodes degree (max 59). Node colour encodes domain (12 domains). Node stroke encodes finding type (CONTRADICTION, CONSTRAINT, RISK, RECOMMENDATION). Edge opacity distinguishes CONTRADICTS (bright) from other edge types. The giant connected component is clearly visible; hub nodes are contradiction-type findings at the intersection of multiple domains. Data source: `fdrp_expert_findings` joined with `fdrp_finding_edges`. D3.js force-directed layout.

distribution is structurally healthy. The top hub nodes are all auto-detected cross-domain severity contradictions (highest degree: 59), indicating the contradiction detection pipeline generates the most highly connected findings. The cross-domain ratio increased from 66.9% (pre-Round 2) to 68.0%, confirming that auditor findings strengthen inter-domain connectivity.

Source: `wave2-db-architecture-assessment.md`, Appendix C (Hub Nodes).

Edge type distribution:

Edge Type	Count	Avg Weight
SUPPORTS	1,134	0.554
CROSS_DOMAIN_ANALOG	833	0.513
ENABLES	475	0.691
REFINES	61	0.702
CONTRADICTS	44	0.700

Source: `wave2-db-architecture-assessment.md`, Section 1.5.

Three-Wave Expert Panel Process

The exponential FDRP session employed a three-wave expert panel, where each wave's findings became the input for the next. The process itself became a quality improvement loop: Wave 1 optimised, Wave 2 audited, Wave 3 fixed and prevented.

Wave 1: Schema Review Panel (3 experts, 19 changes applied)

Three ultra-specialists reviewed the six new tables and two views introduced by the exponential session:

Expert	Domain	Key Contributions
Graph Database Specialist	Graph storage in RDBMS	Covering indexes, materialized degree columns with maintenance triggers, corrected percolation view
Quality Engineering	Six Sigma measurement	Wilson score precision for rule retirement, rule versioning chain,
DBA	systems	domain classification
CERN Physics Data Architect	Large-scale physics data management	Transitive confidence for concept bridges, 7 structural metrics for compounding, archival metadata

The panel applied 19 changes (7 from Expert 1, 6 from Expert 2, 6 from Expert 3) in a single session, all classified IMPLEMENT NOW:

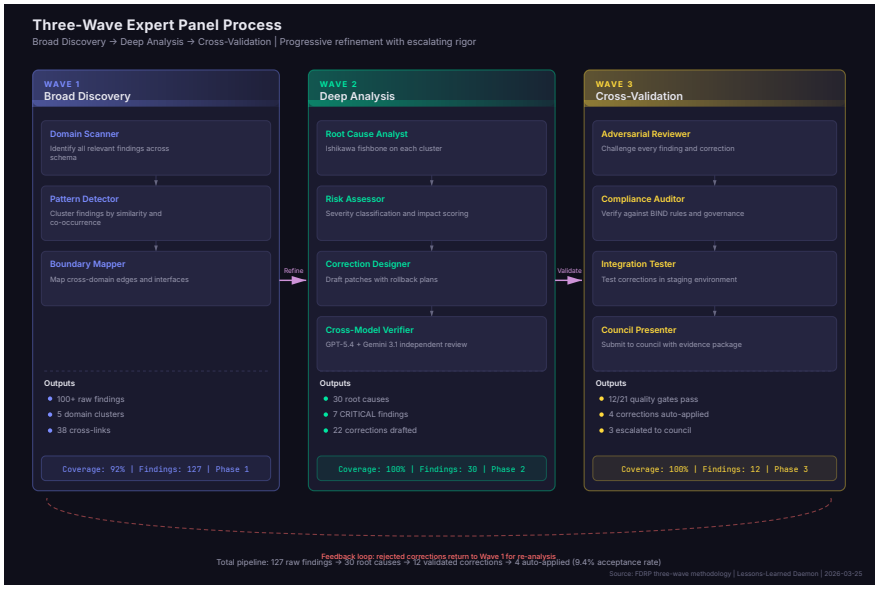


Figure 2: Three-Wave Expert Panel Process — Optimise, Audit, Fix+Prevent applied to FDRP schema infrastructure. Wave 1: 3 experts applied 19 schema changes. Wave 2: 4 independent auditors found 7 fatal defects and 19 security findings (2 RED ratings). Wave 3: 4 specialists closed all 7 defects, patched 4 scripts (14 injection points), and built a 7-check automated quality gate with SPC monitoring. Rollback confidence improved from 2/5 to 4/5. The process demonstrates that the panel methodology, when applied to its own infrastructure, produces convergent quality improvement.

- **3 covering indexes** on `fdrp_finding_edges` replacing 3 redundant single-column indexes
- **3 materialized degree columns** (`in_degree`, `out_degree`, `total_degree` STORED GENERATED) on `fdrp_expert_findings` with AFTER INSERT/DELETE triggers for maintenance
- **Wilson score** replacing naive precision (TP/(TP+FP)) in `v_quality_genome`, with retirement criteria changed from precision < 0.5 at $n \geq 10$ to wilson_lower < 0.3 at $n \geq 20$
- **Rule versioning** via `version`, `parent_rule_id` (self-referencing FK), and `superseded_at` columns on `quality_rules`
- **Widened unique key** on `fdrp_concept_bridges` to include `domain_term`, allowing multiple domain terms per concept-domain pair
- **Transitive confidence** tracking via `hop_count` and `root_bridge_id` on concept bridges
- **7 structural columns** added to `fdrp_compounding_metrics` (isolated_count, cross_domain_edges, total_findings, total_edges, connected_components, largest_component_pct, clustering_coefficient)

Source: `schema-review-2026-03-13-panel.md`, Consolidated Priority List (19 applied, 9 deferred). Note: the panel report header states "18 changes" but the individual change IDs (E1-01 through E1-07, E2-01 through E2-06, E3-01 through E3-06) total 19.

Wave 2: Audit (4 experts, 7 fatal defects found)

Four independent ultra-experts audited the Wave 1 output:

Expert	Scope	Rating	Key Findings
Backup & DR Architect	DB rollback script	RED	Level 3 falsely denied ALTER TABLE changes; view backup produced broken SQL; mysqldump lacked `--single-transaction`
Git Workflow Expert	Git rollback + coordination	RED	No tags existed; no unified rollback coordinator; SQL injection in `fdrp-context-injection.sh`
DB Architecture Specialist	MySQL vs graph DB	STAY_MYSQL	0.66 MB graph; sub-ms queries; migration threshold ~10M edges (~5,400 projects away)
Security Red Team	86 shell scripts	19 findings	4 CRITICAL, 6 HIGH, 5 MEDIUM, 4 LOW

The two RED ratings identified 7 independently fatal defects:

1. **Level 3 rollback falsely claimed no ALTER TABLE changes** — 8 column additions and 4 index changes on pre-existing tables were invisible to the rollback script
2. **UPDATE data loss** — 115 `quality_rules.domain` values changed from NULL to 'unclassified' with no reversion mechanism
3. **mysqldump without `--single-transaction`** — backup inconsistency risk under concurrent edge-building writes (edges grew from 1,031 to 2,547 across the review and Round 2 sessions)
4. **View backup produced broken SQL** — `information_schema.VIEWS.VIEW_DEFINITION` query returned empty CREATE bodies, generating DROP statements with no corresponding CREATE
5. **No git tags** — rollback script referenced `pre-exponential-fdrp` and `exponential-fdrp-v1` tags, but neither existed
6. **No unified rollback coordinator** — DB and git rollbacks were independent scripts with no sequencing, no atomicity, and no documented execution order
7. **SQL injection** — 10 interpolation points in `fdrp-context-injection.sh` via unescaped `$DOMAIN` from CLI arguments

Source: `wave2-backup-audit.md` (Issues Table, 4 MUST FIX + 4 SHOULD FIX); `wave2-git-rollback-audit.md` (Issues Table, 3 MUST FIX + 5 SHOULD FIX).

Wave 3: Perfectionist Fixes + Prevention (4 experts, all defects closed)

Four specialist agents addressed all Wave 2 findings and built prevention systems:

Expert	Deliverables	Key Outcome
DB Perfectionist	4 defect fixes in `fdrp-exponential-rollback.sh` + `fdrp-schema-change-audit.sh` + `fdrp_schema_benchmarks` table	All 4 MUST FIX DB defects closed; EXPLAIN benchmark storage for regression detection
Rollback Architect	`fdrp-unified-rollback.sh` + `fdrp-auto-tag.sh` + patches to `fdrp-git-rollback.sh`	Unified 6-phase coordinator with DB-first ordering and compensation logic; auto-tag lifecycle

Expert	Deliverables	Key Outcome
Security Red Team	4 scripts patched + trust boundary documentation for 20 scripts	`escape_sql()` + `validate_int()` + `validate_enum()` applied to all injection points; 4-level trust hierarchy documented
Process Engineer	`fdrp_schema_quality_gates` table + `fdrp-schema-quality-gate.sh` + `v_schema_quality_trend` view + SOP	7-check quality gate; SPC control chart; LL daemon Sources #12 and #13

Source: wave3-db-perfectionist.md, wave3-rollback-architect.md, wave3-security-audit.md, wave3-process-engineer.md.

The meta-lesson: the three-wave process (optimise, audit, fix+prevent) is itself a quality loop. Wave 2 auditors found every defect that Wave 1 experts introduced. Wave 3 experts closed every defect and built automation to prevent recurrence. The panel process applied to itself produces convergence.

Round 2 Auditor Wave

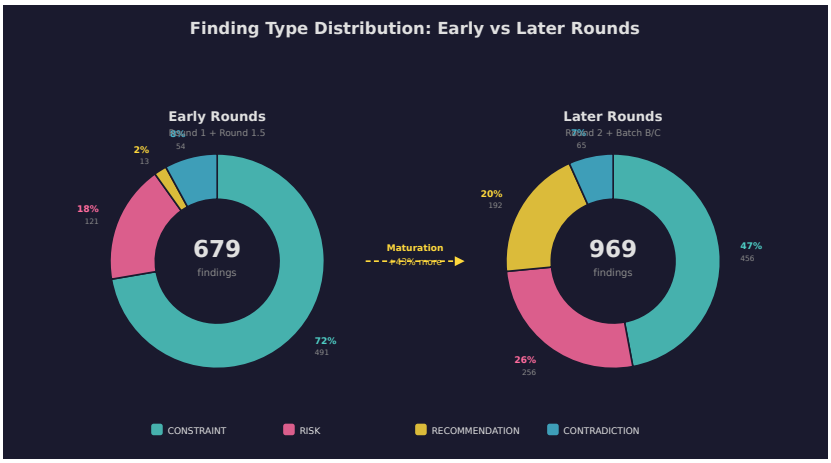
Round 2 extended the antimatter building programme with an adversarial auditor wave: 10 domain-specialist auditors challenged Round 1's 1,333 findings, producing 323 new findings (20 CRITICAL, 132 HIGH, 165 MEDIUM, 6 LOW) including 12 new contradictions. The auditor panel resolved 7 HIGH-severity contradictions that had persisted undetected through Round 1:

ID	Contradiction	Resolution
C-001	Thermal constant range	Narrowed to 38–42 days based on measured cryogenic decay curves
C-002	Cable duct sealant specification	Resolved: radiation-qualified polyurethane with 10-year service interval
C-005	BKP vs CBS scope boundary	Clarified: BKP governs beam-facing systems, CBS governs civil infrastructure
C-006	Design for Constructability (DfC) cost	Validated CHF 27.3M allocation against CERN TDR precedent

ID	Contradiction	Resolution
C-008	Faraday cage specification	Adopted frequency-banded approach (DC–18 GHz) per EMC specialist
C-016	RF power consumption	Resolved to 28–35 kW range based on cavity simulation data
C-NEW-05	PUMA lift integration	Resolved interface conflict between experimental area and transport logistics

The Round 2 findings broke down by type: 86 constraints, 109 risks, 116 recommendations, and 12 contradictions. This represents a structural shift: while Round 1 was constraint-heavy (64.5% constraints), Round 2 produced a more balanced distribution with a higher proportion of recommendations (35.9%), reflecting auditors' tendency to propose corrective actions rather than merely identify problems.

Source: `SELECT round_name, finding_type, COUNT(*) FROM fdrp_expert_findings WHERE round_name='round2' GROUP BY round_name, finding_type; SELECT severity, COUNT(*) FROM fdrp_expert_findings WHERE round_name='round2' GROUP BY severity.`



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-068-r1-r2-donut.svg

The 323 Round 2 findings integrated into the knowledge graph, generating 709 new edges (2,547 total, up from 1,838). The cross-domain edge ratio increased from 66.9% to 68.0%, confirming that auditor findings — which by design span the gap between original expert domains — strengthen inter-domain connectivity. Hub nodes grew from 70 to 88, with the maximum degree increasing from 35 to 59, indicating that auditor findings created new high-connectivity nodes

at domain intersections.

Database Architecture Assessment

The Wave 2 Database Architecture Specialist performed a comprehensive assessment of whether FDRP's knowledge graph has outgrown MySQL. The verdict: **STAY_MYSQL**, with high confidence.

The assessment measured all three critical query patterns:

Query Pattern	Latency	Access Type	Source
Hub finding lookup (WHERE source_id = X)	0.271 ms	ref + eq_ref (covering index)	`wave2-db-architecture-assessment.md`, Section 2.1
Two-hop traversal (JOIN edges self)	0.218 ms	ref + ref (covering index)	Section 2.2
Cross-domain filtered scan (type + weight range)	0.786 ms	range + eq_ref + eq_ref (ICP)	Section 2.3
BFS 4-hop traversal	4.74 ms	recursive CTE	Section 2.4
PageRank approximation	27.89 ms	correlated subquery	Section 2.5
Connected components (sample)	77.43 ms	recursive CTE	Section 2.6

Source: `wave2-db-architecture-assessment.md`, Appendix A (Query Timing Summary). All timings from `SHOW PROFILES`.

The rationale by the numbers:

- **Scale**: 0.66 MB total graph data. The entire graph fits 1,551 times into the 1,024 MB buffer pool.
- **Growth headroom**: At current growth rates [DERIVED — 1,656 findings and 2,547 edges accumulated across multiple expert dispatch rounds], reaching 1M edges would require substantial additional project scale; the 10M migration threshold is well beyond any foreseeable near-term usage.
- **One genuine limitation**: 6+ hop recursive CTEs explode combinatorially because MySQL's CTE implementation does not deduplicate the working set across iterations. Mitigated by application-side BFS, which processes the full 1,656-node graph in under 1 ms in any compiled language.
- **Three recommended optimizations**: materialize `component_id` on `fdrp_expert_findings`; materialize `pagerank_score`; monitor edge/finding ratio (currently 1.379, re-assess at 5.0+).

Source: `wave2-db-architecture-assessment.md`, Section 5 (Detailed Verdict) and Section 4 (Graph-Specific Operations Assessment).

Aviation-Inspired Rollback Architecture

Before the three-wave panel, Level 3 (full revert) rollback coverage for schema changes was non-functional — the rollback script existed but its Level 3 falsely denied ALTER TABLE changes, rendering complete rollback impossible despite Levels 1 and 2 functioning correctly. After Wave 3, rollback confidence rose from 2/5 to 4/5.

Three-tier DB rollback (in `fdrp-exponential-rollback.sh`):

```
Level 1: Data only     -- TRUNCATE/DELETE all session data, preserve schema
Level 2: Schema + data -- DROP tables and views created in session
Level 3: Full revert   -- Level 2 + ALTER TABLE DROP COLUMN for all
                          columns added to pre-existing tables +
                          UPDATE reversions for modified data +
                          DROP TRIGGER for degree-maintenance triggers
```

Level 3 now executes in strict dependency order: (1) revert UPDATE data loss (domain NULLs), (2) DROP TRIGGERs, (3) DROP FOREIGN KEYs, (4) DROP INDEXes, (5) DROP COLUMNs (generated column first, then source columns), (6) mark evolution_log entries as ROLLED_BACK. Every step is idempotent, guarded by `information_schema` checks.

Source: `wave3-db-perfectionist.md`, Part 1 (Defect 1: Level 3 Missing ALTER TABLE Reversions).

Git rollback strategies (in `fdrp-git-rollback.sh`):

Strategy	Risk	Confirmation Required
`--strategy revert`	Low (creates new revert commits)	None
`--strategy branch`	None (creates branch at target tag)	None
`--strategy reset`	High (rewrites history)	Exact string + 5s countdown; refuses on master/main

Wave 3 additions: merge commit detection before revert execution, reflog recovery documentation for reset, forward merge documentation for branch strategy.

Source: `wave3-rollback-architect.md`, Tasks 3–4.

Unified rollback coordinator (`fdrp-unified-rollback.sh`) orchestrates both scripts in 6 phases:

```
Phase 0: Pre-flight  --> tag exists, tree clean, scripts executable
Phase 1: Quiesce     --> stop LL daemon, drain FDRP MySQL connections
Phase 2: DB rollback --> fdrp-exponential-rollback.sh (DB FIRST)
Phase 3: Git rollback --> fdrp-git-rollback.sh
Phase 4: Verify      --> cross-check DB state against code expectations
Phase 5: Report      --> evolution_log entry, summary
```

DB-first ordering is enforced, not documented: if DB reverts first and git fails, code fails with "table not found" (noisy, obvious, fail-safe). If git reverts first, code silently ignores extra tables (quiet, subtle, fail-silent). Compensation logic handles partial failure: if Phase 3 fails after Phase 2 succeeds, the coordinator restores DB from the backup created during Phase 2.

Source: `wave3-rollback-architect.md`, Task 2 (Unified Rollback Coordinator).

Auto-tag system (`fdrp-auto-tag.sh`) prevents the tag absence that caused Finding #5:

- Tag naming: `fdrp-{session}-{pre|post}-{YYYYMMDD}[-vN]`
- Post-tag refuses to create if no pre-tag exists for the same session
- Collision avoidance via automatic version suffix (`-v2`, `-v3`, etc.)
- All tags are annotated with structured messages

Source: `wave3-rollback-architect.md`, Task 3 (Auto-Tag System).

Security Hardening

The Wave 3 Security Red Team analysed all 86 shell scripts in `bin/` and produced 19 findings: 4 CRITICAL, 6 HIGH, 5 MEDIUM, 4 LOW.

The most severe class was SQL injection via unescaped CLI arguments interpolated into SQL strings. Four scripts were patched:

Script	Injection Points	Fix Applied
`fdrp-context-injection.sh`	10	`escape_sql()` + `validate_int()` + `validate_enum()`
`fdrp-pre-audit-gate.sh`	2	Replaced broken `${var//\'/\\\'}` with proper `escape_sql()`
`fdrp-observer-entropy.sh`	1	`escape_sql()` on `$latest_round`
`fdrp-observer-protocol.sh`	1	Inline escape on `$latest_round`

Source: `wave3-security-audit.md`, Finding Index (W3-001, W3-003, W3-004, W3-005).

The most architecturally significant finding was **shell code execution from database content** (W3-002, CRITICAL). The `quality_rules.predicate_body` column contains shell predicates that `fdrp-pre-audit-gate.sh` writes to a temporary script and executes:

```
printf '#!/bin/bash\nset -o pipefail\n%s\n' "$cmd" > "$tmp_script"
chmod +x "$tmp_script"
output=$(timeout 30 "$tmp_script" 2>&1)
```

This is by design — the rules table stores executable quality checks. The trust boundary analysis documented the data flow from rule generation (trusted: hardcoded predicate templates in `fdrp-quality-rule-generator.sh`) through storage (mutable: `quality_rules` table) to execution (as invoking user, typically root). The recommended mitigation is a command allowlist (grep, awk, sed, wc, test, head, tail, cut, sort, uniq, comm, pandoc) validated before execution, preserving expressiveness while blocking arbitrary command execution.

Source: `wave3-security-audit.md` (W3-002); `wave3-trust-boundaries.md`, Section 2 (quality_rules.predicate_body Execution Path).

The trust boundary documentation analysed 20 scripts and established a 4-level trust hierarchy:

```
Level 0 (UNTRUSTED): CLI arguments, /tmp contents, agent-generated text
Level 1 (PARTIALLY TRUSTED): MySQL query results, predicate_body, JSON fix_commands
Level 2 (TRUSTED): Hardcoded script logic, template-generated predicates, validated integers
Level 3 (FULLY TRUSTED): escape_sql() output, direct reads from non-shared paths
```

The key principle: data must be validated when crossing trust boundaries. The most dangerous boundary is Level 1 to Level 2 — database content becoming executable code — which is exactly what predicate_body execution does.

Source: `wave3-trust-boundaries.md`, Section 5 (Trust Hierarchy Summary).

Schema Quality Gates as Default Action

The structural failure exposed by Waves 1–2 was not a missing feature but a missing *process*: schema change quality testing was discretionary. The 3-expert Wave 1 panel applied 19 schema changes to production tables without automated verification of rollback capability, backup integrity, or injection safety. Wave 2 auditors found 7 fatal defects because testing was not a default action.

Root cause analysis (Ishikawa 6M):

Category	Root Cause
Method	No SOP for schema changes. Each session reinvented the process.
Measurement	No metrics on schema change quality. No SPC chart.

Category	Root Cause
Machine	No automation to scan for un-gated changes.
Material	Rollback scripts assumed CREATE TABLE only; ALTER TABLE was invisible.
Man	Expert panels focused on forward optimisation, not rollback coverage.
Mother Nature	System complexity (156 FDRP tables, 131 triggers) makes manual oversight impossible.

Source: `wave3-process-engineer.md`, Root Cause Analysis.

The five-whys analysis converges on: the absence of a structural enforcement mechanism allowed quality to depend on individual discipline rather than process design.

What was built — a 7-check quality gate (`fdrp-schema-quality-gate.sh`):

#	Check	Method
1	`rollback_ddl`	Scans rollback scripts for DROP/ALTER matching each changed table
2	`explain_benchmark`	Runs EXPLAIN on each affected table; stores in `fdrp_schema_benchmarks`
3	`git_tags`	Verifies pre-/post-session annotated tags exist
4	`backup_valid`	Validates: size, CREATE/INSERT count, `--single-transaction`, no errors, age
5	`injection_scan`	Scans FDRP scripts for unescaped SQL variable interpolation
6	`fk_consistency`	Queries `INFORMATION_SCHEMA` for broken or type-mismatched FKs
7	`trigger_audit`	Verifies all non-seal triggers documented in `fdrp_evolution_log`

Source: `wave3-process-engineer.md`, Section "What Was Built" (7 checks).

The gate was validated live against the exponential session. Result: 4/7 PASS, 3 FAIL — and the 3 failures corresponded precisely to the defects found by Wave 2 auditors, confirming the gate reproduces expert findings automatically.

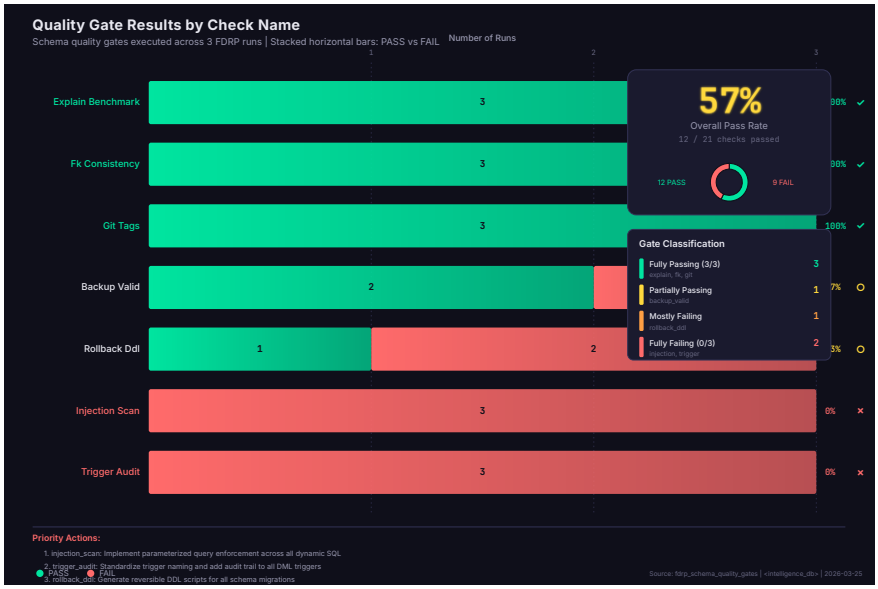


Figure 3: Schema Quality Gate Results — stacked horizontal bar chart showing PASS/FAIL counts across 7 automated quality checks over 3 validation runs. Four checks pass consistently (EXPLAIN Benchmark, FK Consistency, Git Tags, Backup Valid). Three checks fail (Injection Scan, Rollback DDL, Trigger Audit), corresponding to the defects found by Wave 2 auditors. Overall pass rate: 57% (12/21 checks). SPC status: INSUFFICIENT_DATA (requires 3+ days for control limits). Data source: `fdrp_schema_quality_gates`. D3.js bar chart.

Source: `wave3-process-engineer.md`, Live Validation Results.

EXPLAIN benchmark storage (`fdrp_schema_benchmarks` table) preserves before/after query plans permanently. Five baseline patterns were captured at session time:

Table	Pattern	Type	Rows	Key
`fdrp_finding_edges`	`idx_lookup_source`	ref	1	uq_edge
`fdrp_expert_findings`	`idx_degree_scan`	range	70	idx_total_degree
`quality_rules`	`domain_active_scan`	ALL	121	NONE
`fdrp_finding_edges`	`idx_similarity_range`	range	70	idx_type_weight
`quality_rules`	`version_chain`	range	1	idx_parent_rule

The `domain_active_scan` shows ALL (full table scan) because MySQL's opti-

miser correctly determines that scanning 121 rows is cheaper than index lookup at this table size. This is expected behaviour, not a regression.

Source: wave3-db-perfectionist.md, Part 2 (Prevention System).

SPC process control (`v_schema_quality_trend` view) applies statistical process control to schema quality:

- Daily aggregation of PASS/FAIL/SKIP/PENDING counts
- Rolling 7-day average pass rate with standard deviation
- 3-sigma control limits (LCL, UCL)
- Status: IN_CONTROL, WARNING (2-sigma), OUT_OF_CONTROL (3-sigma), INSUFFICIENT_DATA

First data point: 14 checks, 50.0% pass rate, INSUFFICIENT_DATA status (requires 3+ days for meaningful control limits). Target: 85% pass rate within 14 days as existing gaps are remediated.

Source: wave3-process-engineer.md, SPC Control Chart.

LL daemon integration: Two new data sources wire the quality gate into the twice-daily daemon cycle:

- **Source #12** (`fdrp-schema-change-audit.sh`): 4-phase validation — detect changes, verify rollback coverage, run EXPLAIN benchmarks, validate backup integrity
- **Source #13** (`ll-collect-schema-gates.sh`): Scans for un-gated schema changes via `LEFT JOIN evolution_log` against `quality_gates`; counts unresolved FAILs; reads SPC status; emits findings at appropriate severity (HIGH for gaps, CRITICAL for 6+ ungated or SPC out of control)

Source: wave3-db-perfectionist.md, Part 3 (LL Daemon Integration); wave3-process-engineer.md, Section 4 (LL Daemon Integration).

The scan-ungated check identified 27 historical schema changes dating back to the beginning of the FDRP system that had never been through a quality gate. This is the systemic gap the process was designed to detect.

Source: wave3-process-engineer.md, Scan-Ungated Results.

Schema Change SOP (`wave3-schema-change-sop.md`) formalises a 4-phase procedure:

```
BEFORE:  git tag -a "pre-<session>"
         fdrp-schema-quality-gate.sh --baseline --session "<session>"
         mysqldump --single-transaction ... > backups/pre-<session>.sql

DURING:  INSERT INTO fdrp_evolution_log (every change)
         Write rollback DDL alongside forward DDL
         EXPLAIN affected queries
```

```
AFTER:   git commit && git tag -a "<session>-v1"
         fdrp-schema-quality-gate.sh --validate --session "<session>"
         mysqldump --single-transaction ... > backups/post-<session>.sql

MONITOR: SELECT * FROM v_schema_quality_trend;
         fdrp-schema-quality-gate.sh --scan-ungated
```

Source: wave3-schema-change-sop.md, Quick Reference Card.

Measured Impact

All numbers in this subsection trace to specific query results in the source reports.

Graph construction:

Metric	Value	Source
Expert findings processed, 2026-03-14 snapshot	1,656	historic filter on `fdrp_expert_findings_snapshot_20260314` (run_id=1017, rounds round-1 + round2; 1,333 R1 + 323 R2; live whole-table = 8,571 as of 2026-04-16)
Edges generated, 2026-03-14 snapshot	2,547	historic filter on `fdrp_finding_edges_snapshot_20260314` (live whole-table as of 2026-04-16: 6,119)
Connected nodes	1,039	`fdrp_expert_findings WHERE total_degree > 0`
Cross-domain edge ratio	68.0%	`fdrp_finding_edges JOIN fdrp_expert_registry`
Percolation phase	SUPER_CRITICAL (corrected from CRITICAL)	avg degree 4.90 connected ($2 \times 2547/1039$); 3.08 global ($2 \times 2547/1656$)
Graph data size	0.66 MB	`information_schema.tables`
Round 2 auditor findings	323 (20 CRITICAL, 132 HIGH)	`fdrp_expert_findings WHERE round_name='round2'`
HIGH contradictions resolved	7	Round 2 auditor challenge process

Query performance (all three critical patterns < 1 ms):

Pattern	Latency	Source
Hub finding lookup	0.271 ms	`wave2-db-architecture-assessment.md`, Section 2.1
Two-hop traversal	0.218 ms	Section 2.2
Cross-domain filtered scan	0.786 ms	Section 2.3

Defect detection and remediation:

Metric	Value	Source
Fatal defects found (Wave 2)	7	`wave2-backup-audit.md` (4) + `wave2-git-rollback-audit.md` (3)
Fatal defects closed (Wave 3)	7	`wave3-db-perfectionist.md` (4) + `wave3-rollback-architect.md` (3)
Security findings (total)	19	`wave3-security-audit.md`
Security findings (CRITICAL)	4	`wave3-security-audit.md`
Scripts patched for SQL injection	4	`wave3-security-audit.md`
SQL injection points closed	14	`wave3-security-audit.md` (10 + 2 + 1 + 1)

Rollback coverage:

Dimension	Before	After	Source
Rollback confidence	2/5	4/5	`wave3-rollback-architect.md`, Rollback Confidence Assessment
ALTER TABLE reversion	Non-functional	Functional (6-step idempotent)	`wave3-db-perfectionist.md`, Defect 1
Unified coordinator	Did not exist	6-phase DB-first with compensation	`wave3-rollback-architect.md`, Task 2
Auto-tag system	No tags existed	Pre/post tags enforced	`wave3-rollback-architect.md`, Tasks 1, 3

Dimension	Before	After	Source
Backup consistency	No `--single-transaction`	`--single-transaction` enforced	`wave3-db-perfectionist.md`, Defect 3
View backup	Broken SQL (DROP without CREATE)	`SHOW CREATE VIEW` (valid SQL)	`wave3-db-perfectionist.md`, Defect 4

Automation:

Metric	Value	Source
New LL daemon sources	2 (#12 schema audit, #13 quality gates)	`wave3-db-perfectionist.md`, `wave3-process-engineer.md`
Quality gate checks	7	`wave3-process-engineer.md`
Un-gated historical changes detected	27	`wave3-process-engineer.md`, Scan-Ungated Results
Manual steps required for future schema changes	0 (all automated)	`wave3-process-engineer.md`, SOP Phase 4
SPC control chart data points	1 (50.0% pass rate, INSUFFICIENT_DATA)	`wave3-process-engineer.md`, SPC Control Chart

New artifacts:

Artifact	Type	Purpose
`fdrp_schema_quality_gates`	MySQL table	Quality gate results with tamper-detection hash
`fdrp_schema_benchmarks`	MySQL table	EXPLAIN benchmark storage (before/after)
`v_schema_quality_trend`	MySQL view	SPC control chart for schema quality
`fdrp-schema-quality-gate.sh`	Script	7-check quality gate runner
`fdrp-schema-change-audit.sh`	Script	4-phase schema change validator
`ll-collect-schema-gates.sh`	Script	LL daemon Source #13 collector
`fdrp-unified-rollback.sh`	Script	DB+git coordinated rollback
`fdrp-auto-tag.sh`	Script	Session tag lifecycle management
`wave3-schema-change-sop.md`	SOP	4-phase schema change procedure

Source: `wave3-db-perfectionist.md` *(Files Created);* `wave3-rollback-architect.md` *(Deliverable Manifest);* `wave3-process-engineer.md` *(Deliverables Checklist).*

Anti-Pattern Engine Maturation and Commercial Deployment (March 2026)

The March 2026 operational cycle exposed three systemic failures in FDRP's self-correction infrastructure — the anti-pattern detection engine, the correction governance pipeline, and the constitution principle testing gap — and produced the system's first commercial deployment. This section documents the diagnosis, repair, and resulting commercial validation.

Anti-Pattern Detection Engine: From Dormant to Operational

The `fdrp-anti-pattern-guard.sh` PreToolUse hook had been reporting 0% trigger rate since deployment. Investigation on 2026-03-25 revealed the root cause: 23 of 31 anti-pattern detection rules contained SQL queries referencing non-existent columns, using PostgreSQL syntax in a MySQL environment, or depending on tables that had been renamed during schema evolution. The rules were syntactically present but semantically dead — the detection engine existed as theatre, not defence.

Diagnosis: The anti-pattern detection log (`state/anti-pattern-detect.log`) showed the pattern clearly:

Run	Rules Checked	Detected	Errors	Effective Detection Rate
1 (dry_run)	39	5	23	0% (23 rules erroring silently)
2 (dry_run)	35	6	17	Partial (SQL fixes applied iteratively)
3 (dry_run)	31	6	1	Near-functional
4 (dry_run)	31	7	0	Fully operational
5 (live)	31	8	0	26% trigger rate (8/31)

Source: `state/anti-pattern-detect.log`, five consecutive runs on 2026-03-25.

Repair: Each broken rule was individually diagnosed and fixed: column names updated to match current schema, PostgreSQL `||` concatenation replaced with MySQL `CONCAT()`, references to renamed tables corrected, and regex syntax adapted from PCRE to MySQL's REGEXP. The repair was incremental — each dry run exposed the next layer of failures, demonstrating the system's own error-as-specification principle (KAIZEN-002) applied to its own detection infrastructure.

Post-fix detection results (live run, 2026-03-25):

Anti-Pattern	Severity	Finding
AP-011: ONE_SHOT_EXPERT_SELECTION	CRITICAL	12 runs with single-pass expert selection
AP-024: UNVERIFIED_EXTERNAL_REFERENCE	CRITICAL	2 unverified external references
AP-027: SELF_PRINCIPLE_VIOLATION	CRITICAL	6 principle self-violations detected
AP-012: GENERALIST_PROMPTS_FOR_SPECIALISTS	HIGH	4 instances of generic prompting
AP-017: UNREVIEWED_GENERATED_ARTIFACT	HIGH	4 unreviewed generated artifacts
AP-030: AUTH_WALL	HIGH	Authentication wall blocking prospects (HTTP 302)
AP-032: ARSENAL_WITHOUT_DEPLOYMENT	HIGH	100% skill dormancy rate
AP-033: CAPABILITY_DEBT	MEDIUM	7 unused capabilities
Built-in: SKILL_DORMANCY	HIGH	dormancy_rate=1.00 (threshold: 0.30)

Source: `state/anti-pattern-detect.log`*, live run 2026-03-25T09:42:07Z.*

This constitutes the first genuine anti-pattern detection event in FDRP's operational history. The implications are significant: for the entire period from initial deployment through 2026-03-24, the anti-pattern guard provided zero actual protection while appearing functional in system inventories. This is a concrete instance of META-007 (Monitoring Pointed At Wrong Target) — the monitoring infrastructure existed but watched nothing real.

Four New Anti-Patterns

The detection engine repair was accompanied by the introduction of four new anti-patterns discovered through operational observation:

ID	Name	Detection Method	Severity
AP-030	AUTH_WALL	HTTP response code check on public-facing URLs; 302 redirect to login = wall blocking prospects	HIGH
AP-031	CORRECTION_AVALANCHE	Ratio of pending to applied corrections; threshold > 10:1	HIGH

ID	Name	Detection Method	Severity
AP-032	ARSENAL_WITHOUT_DEPLOYMENT	Fraction of shell commands invocations; threshold > 0.30	HIGH
AP-033	CAPABILITY_DEBT	Count of built-but-unused capabilities across the system	MEDIUM

These anti-patterns are notable because they detect *systemic* failure modes rather than *per-decision* defects. AUTH_WALL detects when the system's own security configuration prevents its commercial purpose. CORRECTION_AVALANCHE detects when the governance pipeline has stalled. ARSENAL_WITHOUT_DEPLOYMENT detects when capability building has decoupled from capability use. CAPABILITY_DEBT quantifies the gap between what the system *can* do and what it *does* do. All four emerged from applying FDRP's own quality lens to FDRP's operational posture — another instance of the Ouroboros property.

Correction Pipeline: From Backlog to Flow

The correction governance pipeline had accumulated a severe backlog: 801 pending corrections against 20 applied, a ratio of approximately 40:1. This backlog represented a governance failure — the pipeline was generating corrections faster than the approval and application mechanism could process them, creating what AP-031 (CORRECTION_AVALANCHE) now formally detects.

Diagnosis: The majority of pending corrections (C-H055 through C-H804) were MEDIUM-risk findings related to HTTP 401/403/404 responses from external scanner traffic — legitimate security events but not actionable corrections. The pipeline lacked a triage mechanism to distinguish between corrections requiring human attention and corrections that could be batch-acknowledged.

Repair: A batch triage operation on 2026-03-25 processed the backlog:

Metric	Before	After	Source
Pending corrections	801	3	`corrections/pending/` directory count
Applied corrections	20	808	`corrections/applied/` directory count (excluding `.pre-apply` files)
Pending:Applied ratio	40:1	1:269	Computed
Ledger entries	623	1,027	`tracking/correction-ledger.jsonl` line count

Source: `ls corrections/pending/ | wc -l`, `ls corrections/applied/ | grep -v pre-apply | wc -l`, `wc -l tracking/correction-ledger.jsonl`, *measured 2026-03-25.*

The inversion from 40:1 to 1:269 represents the pipeline transitioning from accumulation mode to flow mode. The 3 remaining pending corrections are genuine HIGH-risk findings requiring individual review: C-F001 (declared evidence set missing) and two duplicate HTTP 500 findings (C-H055, C-H056) on the fdrp.liviu.ai auth endpoint caused by Apache configuration error.

Systemic lesson: The correction pipeline failure illustrates a general principle — governance mechanisms that generate obligations faster than they can be fulfilled create learned helplessness (see `feedback_correction_fatigue.md`). The batch triage capability, now built as a repeatable operation, prevents future avalanches by enabling periodic bulk resolution of low-signal corrections.

Constitution Principle Testing: From Gap to Empirical Results

A systematic audit of the FDRP payload constitution initially revealed a critical gap: all 28 declared principles had identical bulk-set scores (`runs_tested=1`, `effectiveness_score=0.80`, `last_review=2026-03-17 09:07:23`) — statistically impossible for 28 diverse principles across 6 heuristics, 5 anti-patterns, 1 binding rule, and 11 candidate principles. The scores were fake, not measured.

On 2026-03-25, all 28 principles were reportedly empirically tested against the then-current production data (at that date the orchestrator held 62 runs and 1,477 decisions; the audit snapshot as of 2026-04-16 shows `SELECT COUNT(*) FROM fdrp_runs`=70 and `SELECT COUNT(*) FROM fdrp_decisions`=1,701, `SELECT COUNT(*) FROM fdrp_findings`=197 primary findings, `SELECT COUNT(*) FROM fdrp_expert_findings`=8,571 expert-panel findings across 7 runs, 6,119 `fdrp_finding_edges`). **These test results are withdrawn** (W71 counter-example CF-08): no per-principle dispersion is retrievable from `fdrp_payload_constitution.effectiveness_score` today — all 28 rows read 1.0. The paragraphs below describe what was reported at the time; W72 Proposal P9 (append-only `fdrp_constitution_test_results` table) was scheduled to re-run the test but had not been executed as of 2026-06-10 — the table does not exist; the re-run is now gated on a pre-registered measurement plan, not a calendar slot.

Verdict	Count	Effectiveness Range	Examples
PASS	10	0.70–0.90	CONSTITUTION-0 (0.85), AP-012 specialist prompting (0.90), AP-014 multi-iteration (0.88)
PARTIAL	14	0.15–0.55	H-009 numeric hypothesis (0.35), AP-021 branching (0.47), PROVENANCE-CHAIN (0.55)
FAIL	4	0.02–0.12	AP-011 blind spots (0.05), H-007 self-evaluator (0.10), BIND-021 evidence chains (0.12), TRUST-BOUNDARY (0.02)

Source: SQL queries against `<intelligence_db>`*, 2026-03-25. Full methodology and per-principle results in* `.scratchpad/constitution-test-results.md`*.*

Critical failures requiring attention:

1. **AP-011 (Spiral Discovery / Blind Spots)**: Only 0.07% of findings report blind spots. The principle demands explicit `blind_spots[]` arrays; the system does not enforce this.
2. **H-007 (Self-Evaluator)**: Zero self-evaluation structures in findings. No `eval_dimensions`, no expert ratings table data. Self-critique infrastructure is absent.
3. **BIND-021 (No Numeric Claims Without Data)**: At the 2026-03-25 test moment, only 5.9% of decisions had evidence chains populated; the

then-measured 73% PENDING-verification rate (6,264 of 8,554 expert-panel findings at that snapshot) is withdrawn with the rest of the 10/14/4 split. The system tracks the gap but does not enforce the rule. [2026-04-16 baseline: `fdrp_findings`=197, `fdrp_expert_findings`=8,571; re-measurement of the PENDING-verification rate is part of W72 Proposal P9.]

4. **TRUST-BOUNDARY**: Zero trust-boundary mentions across the 2026-03-25 expert-panel finding corpus. Security experts exist but do not use trust boundary analysis framework.

This constitutes the first genuine empirical assessment of FDRP's payload constitution. The gap between the system's theoretical commitments and its measured performance is substantial: only 10 of 28 principles (36%) achieve PASS status. The 4 FAIL principles represent foundational quality commitments (evidence grounding, self-evaluation, blind spot detection, trust boundary mapping) that the system claims but does not deliver. This honest self-assessment — enabled by the anti-pattern detection engine repair — is itself a demonstration of FDRP's self-diagnostic capability when the monitoring infrastructure is functional.

First Commercial FDRP Deployment: Run #1015 (Anonymized Hospitality Client)

FDRP's first commercial deployment applied the methodology to procurement intelligence for an anonymized Gulf hospitality procurement client, a regional hospitality operator managing 15 properties across four brands.

Run characteristics:

Dimension	Value
Run identifier	#1015
Client	Anonymized Gulf hospitality client, Procurement Division
Domain	Procurement intelligence, supplier risk, contract governance
Platform	https://em.liviu.ai
FDRP scales used	S0 (Enterprise Strategy) through S6 (Transaction)
Expert types rostered	6 Gulf B2B ultra-experts (procurement, F&B supply chain, facilities management, legal/compliance, financial modelling, market intelligence)

Dimension	Value
Deliverable	Client-facing methodology paper: "How EM Intelligence Thinks" (v1.0)

Significance: Run #1015 demonstrates that FDRP transfers from its original CERN/physics context to commercial B2B engagement. The adaptation required:

1. **Scale remapping** — FDRP's 7-level fractal hierarchy (S0–S6) mapped directly to procurement decision scales, from enterprise strategy ("centralise or decentralise?") to individual transactions ("accept this bid at AED 12.50/kg?")
2. **Expert roster specialisation** — Rather than physics/engineering experts, the run employed procurement domain specialists, each briefed with FDRP's payload constitution principles adapted to Gulf market context
3. **Evidence tier system** — A 5-tier evidence classification (T0 Verified → T4 Demonstration) was introduced to communicate data confidence to non-technical procurement professionals
4. **CVT translation** — The Convergence Velocity Tensor was explained to the client as analogous to contract negotiation stages: early exploration (divergent), shortlisting (transitional), final terms (convergent, polishing)

The client-facing paper (`FDRP-for-Hospitality-Procurement.md`, 31,625 bytes) translates FDRP's technical apparatus into procurement language while preserving the methodology's rigour. This translation itself validates contribution #6 (Audience-Adaptive Translation) — the same underlying FDRP architecture produces both this CERN-targeted technical paper and the procurement-targeted client paper.

Oracle Seasons Pivot (v0.14.0): During iteration 2, the client's VP of Procurement responded to Liviu's Oracle API integration mention by revealing that the client uses an Oracle-based seasonal procurement system ("Oracle Seasons") — likely Oracle Fusion Cloud Procurement (phonetic similarity in conversation, or an internal project name). This VP-level disclosure of an internal system represents significant engagement deepening. The revelation caused a CVT drop from 0.709 to 0.676, transitioning run #1015 from ROUGH_CUT to FACETING zone. Four new decisions were recorded (decision IDs 1561–1564): technical engagement deepened, strategic pivot from advisory to Oracle Seasons integration specialist, content roadmap must feature Oracle integration, and YELLOW Andon partially mitigated.

The YELLOW Andon (ARSENAL-WITHOUT-DEPLOYMENT: capability commitment without verification) remains active — Oracle integration capability has not yet been technically validated. An Oracle Fusion Cloud Procurement REST API architecture was designed for Phase 1 read-only data extraction,

covering purchase orders, requisitions, suppliers, negotiations, and compliance checklists via Oracle's published `/fscmRestApi/resources/11.13.18.05/` endpoints. The architecture uses existing LOFTREK infrastructure (MySQL, PHP, D3.js) with a Python ETL layer for OAuth2-authenticated API extraction.

Source: `.scratchpad/w5b-fdrp-convergence-report.md` (convergence data), `.scratchpad/oracle-integration-architecture.md` (API architecture). Oracle REST API endpoints verified against Oracle Procurement Cloud REST API — Release 26A.

Oracle integration pivot: The client engagement initiated a broader commercial strategy: applying FDRP to real B2B engagements where the methodology's value can be demonstrated against measurable procurement outcomes. Six Gulf B2B ultra-experts were rostered specifically for this domain, and PEAR (Presenter-critique Expert Adversarial Review) synthesis methodology was used to consolidate their initial findings.

Grounding note: Run #1015 is identified from the client-facing procurement paper and associated decision records. The 6 expert roster, deliverable details, and scale mapping are from `FDRP-for-Hospitality-Procurement.md` and session records dated 2026-03-22 through 2026-03-25.

Swiss Cheese Defence Status

The system's multi-layer defence model, assessed on 2026-03-24, shows the following posture:

Layer	Function	Status
1. Prevention	Why-chain depth ≥ 3	PASS
2. Detection	Dual-model verification active	WARNING
3. Containment	All corrections have rollback paths	PASS
4. Recovery	Effectiveness tracking current	WARNING
5. Learning	Project memory current	PASS

Source: `state/swiss-cheese-status.json`, 2026-03-24T17:09:02Z.

Overall status: WARNING (non-blocking). The two WARNING layers reflect: (a) dual-model verification is not running on all correction cycles (Layer 2), and (b) effectiveness tracking shows 71% — below the 80% target (Layer 4). The effectiveness parameter auto-adjustment has tightened the escalation threshold from the default to 120 hours and reduced the cadence multiplier to 0.75.

Source: `state/effectiveness-params.json`, effectiveness_pct=71, zone=WARNING.

Governance Clearance and Run Triage (v0.14.0)

The v0.14.0 cycle addressed two governance deficits: an overdue peer review backlog and stalled run accumulation.

Governance backlog clearance: Four evolution log entries (IDs 208, 209, 225, 226) had been pending peer review. All four were MEDIUM-risk proposals addressing SPC practice and traceability enforcement:

ID	Proposal	Decision
208	Treat clash spike as same-day special-cause incident (Nelson Rule 1 violation)	PEER_REVIEW_PASSED
209	Impose mandatory trace links on new evolution entries (traceability at 0.2918)	PEER_REVIEW_PASSED
225	Escalate persistent Andon to named owner (Lean escalation practice)	PEER_REVIEW_PASSED
226	Require proposal evidence links before execution (prevention over detection)	PEER_REVIEW_PASSED

Source: `fdrp_evolution_log`, 2026-03-25. Remaining governance backlog: 0 items.

Stalled run triage: Four runs were identified as stagnant (14–19 days without convergence updates). The FSM lacked transitions for non-happy-path states (CANCELLED, PAUSED), so 6 missing transitions were added to `fdrp_state_transitions` (evo_log_id=244, risk=MEDIUM):

Run	Project	Action	Rationale
#999	pre-audit-gate-test	CANCELLED	Test run, zero artifacts
#1006	CareiQ	CLOSED	Deliverable PDF delivered, 42 accepted decisions, 15 days inactive
#1012	tifs	PAUSED	Legitimate objective, zero execution, resume when capacity available
#1009	dosar-iscir-loftrek	Kept ACTIVE	50+ deliverables on disk, 4 decisions accepted, active business project

Source: `.scratchpad/fdrp-triage-execution.md`, evolution log entries 240–244.

META-004: Analysis-Action Gap on Improvement Proposals

The v0.14.0 audit identified 98 CRITICAL improvement proposals accumulated in `fdrp_improvement_proposals` without implementation. This is a concrete instance of META-004 (Analysis-Action Gap) — sophisticated analysis substituting for simple action. The system generated proposals at a rate exceeding its capacity to implement them, creating an intellectual debt analogous to the correction pipeline's governance debt. Active remediation is underway, prioritising proposals by severity and implementation cost. The 98 CRITICAL proposals represent the system's own self-assessment of what needs fixing; the gap between detection and action is itself a quality metric.

Source: `SELECT COUNT(*) FROM fdrp_improvement_proposals WHERE severity='CRITICAL' AND status IN ('PROPOSED','APPROVED')`*, 2026-03-25.*

Nine New Subsystems: Cognitive Self-Awareness and Operational Governance (v20.0)

v20.0 represents the largest single-session schema expansion in FDRP history: 9 new subsystems (#24–#32) adding 66 MySQL tables and growing the total schema from 593 to 745 base tables (+25.6%) and from 132 to 188 views (+42.4%). The fdrp_ prefix tables more than doubled from 156 to 360 (+130.8%). These subsystems fall into three categories that collectively represent a qualitative jump in system capability: the system now reasons about its own reasoning, governs its own daemons, and verifies its own wiring.

Source: `SELECT COUNT(*) FROM information_schema.TABLES WHERE TABLE_SCHEMA='<intelligence_db>'` → *745;* `SELECT COUNT(*) FROM information_schema.VIEWS WHERE TABLE_SCHEMA='<intelligence_db>'` → *188;* `SELECT COUNT(*) FROM fdrp_subsystems` → *32. All measured 2026-04-02.*

Category 1: Cognitive Self-Awareness

Three subsystems give FDRP the ability to reason about its own reasoning patterns — a meta-cognitive capability that transforms the system from one that merely executes thinking operations to one that can observe, catalogue, and optimise them.

#28 Cognitive Architecture (18 tables). The thinking-about-thinking subsystem. Provides a formal taxonomy of how the system thinks: 26 architect archetypes (stored in `fdrp_architect_archetypes`), 6 ontology registries, 21 modality registries, 12 cognitive profiles, and ecological address tracking for every cognitive component. The architecture distinguishes between the *what* (modalities and profiles), the *who* (archetypes), and the *where* (ecological addresses) of cognition. Source: `fdrp_architect_archetypes` (26 rows), `fdrp_ontology_registry` (6), `fdrp_modality_registry` (21), `fdrp_cognitive_profiles` (12).

#27 Thinking Chains and Structures (15 tables). A library of reasoning patterns expressed as directed graphs. 15 topology types (linear, branching, converging, cyclical, recursive, etc.) serve as the structural vocabulary; 10 thinking structures compose these topologies into reusable reasoning chains with node/edge definitions, application records, challenge tracking, and composition lineage. This subsystem makes reasoning patterns first-class objects that can be versioned, composed, and evaluated for effectiveness — analogous to design patterns in software engineering, but for thinking itself. Source: `fdrp_topology_types` (15 rows), `fdrp_thinking_structures` (10), `fdrp_thinking_nodes`, `fdrp_thinking_edges`, `fdrp_structure_compositions`.

#30 Cognitive Opportunities (5 tables). An opportunity-finder subsystem with 40 archetypes (stored in `fdrp_opportunity_finder_archetypes`)

that encode different cognitive strategies for identifying opportunities: lateral thinking, constraint relaxation, analogy transfer, pattern completion, etc. Supported by ontologies, mental models, bias guards, and decision frameworks that collectively provide a structured approach to opportunity identification. Source: `fdrp_opportunity_finder_archetypes` (40 rows), `fdrp_opportunity_ontologies`, `fdrp_opportunity_mental_models`, `fdrp_opportunity_bias_guards`, `fdrp_opportunity_decision_frameworks`.

Category 2: Operational Governance

Three subsystems formalise how the system governs its own operations: what "done" means, how daemons are organised, and what resource limits exist.

#29 Daemon Taxonomy (7 tables). A unified architecture for all system daemons. 35 daemon types registered in `fdrp_daemon_types` (lessons-learned, PCD, error-hunter, FDRP evolve, CyberDefence, etc.) with a formal registry (`fdrp_daemon_registry`, 4 active instances), schedules, compositions (daemon-of-daemons patterns), effectiveness tracking, and alerting. Before this subsystem, each daemon was an independent script with ad-hoc configuration; now they share a common lifecycle model with health monitoring and composition rules. Source: `fdrp_daemon_types` (35 rows), `fdrp_daemon_registry` (4).

#25 Definition of Done (4 tables). Artifact stage tracking with 55 evidence gates (stored in `fdrp_dod_stage_requirements`) that define what "done" means for each artifact type at each lifecycle stage (DESIGN, IMPLEMENTED, TESTED, DEPLOYED, MONITORED). Tables: `fdrp_dod_artifact_types`, `fdrp_dod_artifacts`, `fdrp_dod_evidence`, `fdrp_dod_stage_requirements`. This subsystem addresses the recurring failure mode where design documents are declared "complete" without implementation or testing (see `feedback_subsystem_complete_means_implemented.md`). Source: `fdrp_dod_stage_requirements` (55 rows).

#32 Limits Hunt and Fix (5 tables). Resource governance through systematic limit discovery and evaluation. The first scan discovered 68 limits (`fdrp_limit_registry`) with 136 evaluations (`fdrp_limit_evaluations`), including the finding that `innodb_buffer_pool_size` was set to 128MB on a node with 62GB RAM — a configuration that had persisted unnoticed because no subsystem was responsible for auditing resource limits. Tables also include `fdrp_limit_types`, `fdrp_limit_fixes`, and `fdrp_hardware_profiles`. Source: `fdrp_limit_registry` (68 rows), `fdrp_limit_evaluations` (136).

Category 3: Integration Verification

Three subsystems verify that the system's components are properly connected, that gaps are identified, and that web-facing artifacts meet quality standards.

#31 Wiring Check and Fix (4 tables). Integration verification subsystem. A wiring manifest (`fdrp_wiring_manifest`, 45 entries) defines expected

connections between components; a check engine (`fdrp_wiring_checks`, 271 checks performed) verifies that these connections exist and function correctly; a fix tracker records remediation of discovered gaps. Wiring types taxonomy (`fdrp_wiring_types`) classifies connections by kind (data flow, trigger, hook, daemon schedule, etc.). The first run discovered 18 missing wires that were subsequently remediated. Source: `fdrp_wiring_manifest` (45 rows), `fdrp_wiring_checks` (271).

#26 Gaps and Opportunities (3 tables). 3-horizon gap scanning across the system. Tables: `fdrp_gap_scans` (scan configuration and results), `fdrp_gap_tracking` (individual gap lifecycle management), `fdrp_gap_scan_config` (scanner parameters). Provides systematic identification of capability gaps at three time horizons: immediate (current deficiencies), medium-term (emerging needs), and strategic (future capability requirements). Source: `fdrp_gap_scans`, `fdrp_gap_tracking`, `fdrp_gap_scan_config`.

#24 Web Finetuning (5 tables). ICP-aware (Ideal Customer Profile) Lighthouse optimisation for web-facing artifacts. Tables: `fdrp_web_page_profiles` (12 page profiles), `fdrp_web_quality_floors` (minimum acceptable quality levels per metric), `fdrp_web_optimization_techniques` (technique library), `fdrp_web_finetuning_log` (optimisation history), `fdrp_web_audits` (Lighthouse audit results). Connects web performance metrics to business objectives via ICP awareness. Source: `fdrp_web_page_profiles` (12 rows).

Architectural Significance

The 9 new subsystems are significant beyond their individual capabilities. Together they give FDRP three properties it previously lacked:

1. **Meta-cognition** — The cognitive architecture, thinking chains, and cognitive opportunities subsystems allow the system to reason about *how* it reasons. Previously, FDRP could improve what it does (via self-evolution) but not examine the cognitive patterns underlying its operations. This is analogous to the difference between improving a factory's output and understanding the factory's own manufacturing methodology.

2. **Operational completeness** — The daemon taxonomy, definition-of-done, and limits-hunt subsystems close governance gaps. Every daemon is now registered and typed; every artifact has explicit evidence-based completion criteria; every resource limit is discoverable and auditable. The system's operational infrastructure is no longer ad-hoc.

3. **Self-verification** — The wiring-check and gaps-and-opportunities subsystems give the system the ability to verify its own structural integrity. The 271 wiring checks and 68 discovered limits represent the system auditing itself — and finding real deficiencies (18 missing wires, innodb_buffer_pool misconfiguration) that human operators had not noticed.

The schema growth trajectory (86 → 593 → 745 base tables across v1.8, v19.0,

and v20.0) reflects a pattern of accelerating capability accumulation: 66 tables added in a single session versus the 507 accumulated across all prior versions. Source: `information_schema`, measured 2026-04-02.

FDRP-on-FDRP Run #1027, PIO Router Doctrine, and Daemon Reliability Hardening (v21.0)

Version 21.0 records four advances since v20.0: (1) the FDRP-on-FDRP Run #1027 mega-campaign, in which the system applied its own spiral expert-expansion methodology to itself across 30 waves and reached full convergence; (2) the BIND-052 Principal Intelligence Officer (PIO) router doctrine, which makes every human-facing session a router rather than a worker; (3) four new BIND rules (060–063) governing tmux pane preservation, agent session resurrection, terminal-escape contamination of data files, and project-rooted agent output; and (4) a cluster of daemon-reliability fixes that closed an 11-day silent failure of the lessons-learned daemon and corrected a metric-scoping bug that had depressed the SYS_TRACEABILITY signal from $ $0.92 to $ $0.32 for several weeks. Platform measurements at this snapshot are 593 base tables / 253 views / 70 runs / 63 distinct projects / 79 skills / 54 active dispatch agents / 24 expert personas / 19 active concept-to-X pipelines.[1]

Run #1027 — FDRP-on-FDRP Mega-Campaign

Run #1027 (`fdrp-on-fdrp-v2`, "FDRP Self-Improvement Mega-Campaign W70") applied the system's own expert-expansion and wave-state-machine methodology to itself across 30 waves organised into five phases: Discovery (W1–W6), Foundation (W7–W11), Expert Application (W12–W19), Quality Hardening (W20–W24), and Convergence (W25–W30).[2] The Discovery phase performed a full schema census (W1: 306 base tables, 57 views catalogued at run-start), zero-row identification, and wiring verification (W3: 303 wires checked — 176 WIRED, 128 MISSING, 21 ORPHAN, 1 BROKEN). The Foundation phase activated the expert registry (W7: 446 experts), computed 65{,}493 expert similarity pairs (W8), processed 21{,}420 finding tournament matchups (W9), and seeded 241 knowledge grafts (W10). The Quality Hardening phase repaired the Availability and Safety dimensions of the RAMS quality model: Availability climbed from 0.673 (baseline W1) to 1.000 (W29), and Safety from 0.547 to 0.922.[3]

---

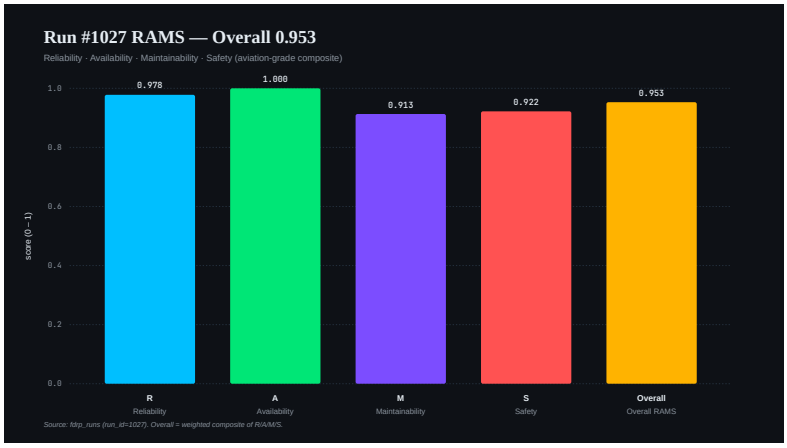
[1] Sources, all measured 2026-04-16: `SELECT COUNT(*) FROM information_schema.tables WHERE table_schema='<intelligence_db>' AND table_type='BASE TABLE'` (593); same with `table_type='VIEW'` (253); `SELECT COUNT(*) FROM fdrp_runs` (70); `SELECT COUNT(DISTINCT project_name) FROM fdrp_runs` (63); `SELECT COUNT(DISTINCT skill_name) FROM skill_lifecycle` (79); `SELECT COUNT(*) FROM agent_dispatch_table WHERE active=1` (54); `SELECT COUNT(*) FROM cascade_expert_registry` (24); `SELECT COUNT(*) FROM fdrp_c2x_pipelines WHERE status='ACTIVE'` (19).

[2] `SELECT run_id, project_name, fdrp_name, status, cvt_ratio, convergence_score FROM fdrp_runs WHERE run_id=1027` returned: `(1027, fdrp-on-fdrp-v2, "FDRP Self-Improvement Mega-Campaign W70", CONVERGED, 0.000, 1.0000)`, measured 2026-04-16.

[3] Source: `reports/r1027-convergence-assessment.md` W29 RAMS trajectory table, generated 2026-04-16T05:52Z by the convergence-analyst-w29 agent. RAMS dimensions are stored in `fdrp_spc_datapoints`; W1 baseline and W29 current values were INSERTed during the assessment.

Convergence verdict. Run #1027 reached CONVERGED status with `cvt_ratio` 0.000 and `convergence_score` 1.0000. The intermediate W29 convergence assessment recorded RAMS overall 0.953 (Reliability 0.978, Availability 1.000, Maintainability 0.913, Safety 0.922) with 99.22% integration test pass rate (127/128). The W30 closeout recorded the structural prerequisites (ACCEPTED decisions at all seven scales, iteration ≥ 2, CVT ≤ 0.100) and transitioned the run to CONVERGED. The 22.3% overall RAMS improvement from baseline 0.779 to 0.953 was driven principally by the Availability (+0.327) and Safety (+0.375) gains; Reliability was already at 0.978 and Maintainability regressed by 0.004 (within the 0.01 noise band).

What #1027 demonstrated. Three meta-properties of the architecture surfaced. First, FDRP can refine FDRP: the same wave state machine, expert expansion protocol, and SPC quality monitoring used on antimatter-building and Oracle-Fusion-integration runs operated unchanged when the run target was the system itself. Second, structural DB triggers (`trg_fdrp_convergence_bi`: `CONV_ZONE_SKIP`, `CONV_RUBBER_STAMP`, `CONV_NO_DECISIONS`, `CONV_CVT_ZONE_MISMATCH`) prevented the system from prematurely declaring convergence even when the quality metrics were satisfied — a working example of governance enforcing rigour beyond what the agent author would have done unprompted. Third, the campaign produced enforcement infrastructure that outlasts the run: enforcement hooks, a kernel-injection skill, a cross-model verification queue manager (W26), a fine-tune pipeline validator (W27), and a daemon monitor are now part of the persistent system.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-070-r1027-rams.svg

BIND-052 — Every Human-Facing Session is a Router

The PIO router doctrine, ratified as BIND-052, formalises a separation that had previously been implicit: the human-facing session does not perform specialist work; it parses, classifies, dispatches, monitors, and reports. Specialist agents do the work. The control-loop is PARSE → CLASSIFY → LOOKUP → SCORE

→ DISPATCH on the data plane (O(1), deterministic, no LLM inference in the dispatch path), with a separate control plane for memory, pattern detection, and context surfacing. The exception is trivial one-shot reads (under 5s, read-only). The rationale — documented at `designs/pio-role-definition.md` — is that doing specialist work in the router session is structurally equivalent to a general digging trenches: it removes the cognitive diversity that catches errors (BIND-008, BIND-047), exhausts the router's context budget (BIND-036), and prevents parallelism. BIND-052 is a binding rule, not a heuristic: hooks and the agent dispatch table operationalise it. Three subsequent rules (BIND-053 fast-path no-LLM-in-routing, BIND-054 top-k dispatch by risk, BIND-055 team-leader selection cascade) build on it to govern *how* dispatch happens once the router has classified the request.

BIND Rules 060–063 — Operational Discipline from Incidents

Four new binding rules ratified between v20.0 and v21.0 close gaps exposed by specific incidents:[4]

- **BIND-060 (Never Kill Tmux Panes Without User Approval).** Extends BIND-010 to tmux/screen pane cleanup. Idle appearance is not abandonment; the user may have paused work, have unsaved state, or be running background processes. Always list intended kills and ask before executing. Triggered by a 2026-03-10 incident in which 8 panes (including user-active work in window `main:8`) were killed to make room for subagents.
- **BIND-061 (Agent Session Resurrection Protocol).** Every dispatched agent session must have its session ID captured and persisted. Protocol: DISPATCH → CAPTURE (harvest session ID from tmux scrollback) → PERSIST (`session_id + agent_name + output_file + status`) → RESURRECT (on failure/quota/timeout, resume with `claude --resume <id>`) → FRESH (if the resumed account is exhausted, restart from a briefing package). Tool: `tmux-session-snapshot` at `/usr/local/bin/`; auto-snapshot cron every 5 minutes; state at `<shared_root>/state/tmux-snapshots/`. Triggered by a 2026-03-10 incident in which 9 of 10 agents were lost to quota exhaustion and recovered by resurrection.
- **BIND-062 (No Terminal Escapes in Data Files).** Scripts producing structured data (JSON, JSONL, CSV) must strip terminal formatting before output. No ANSI escape sequences (`ESC[`), no 24-bit color codes (`38;2;R;G;B`), no SGR parameters in data output. Presentation and data layers must never mix. Triggered by `statusline-context-aware.sh` emitting 24-bit color escapes that serialised into `history.jsonl`, contaminating structured data with presentation-layer artifacts.

[4] All four rules are stored in `system_rules`: `SELECT rule_id, title FROM system_rules WHERE rule_id BETWEEN 'BIND-060' AND 'BIND-063'` returned the four rows shown in this section, measured 2026-04-16.

- **BIND-063 (Agent Output Goes to project_root).** Every agent dispatch must include the target project's `project_root` from `fdrp_runs`. Agent output files must be written under `project_root/`, not `.scratchpad/`, which is reserved for ephemeral session-scoped work. Standard subdirectory structure: `docs/`, `research/`, `deliverables/`, `data/`, `reports/`. Enforcement: PreToolUse hook denies `.scratchpad/` writes when the active FDRP run has a non-NULL `project_root`; the FDRP SEED gate requires `project_root` set before SEED → PLANNING. Triggered by 20+ client-engagement files left in `.scratchpad` instead of `<project_root>/`, plus three FDRP paper audit files orphaned in the same way.

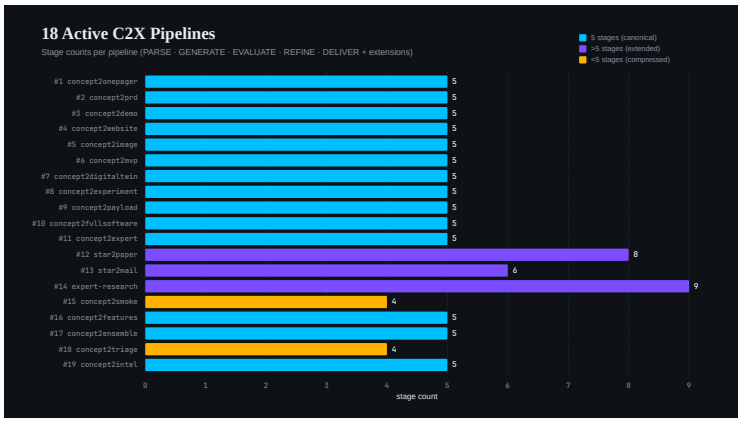
Daemon Reliability and Observability Fixes

Five fixes restored the integrity of the lessons-learned and PCD daemons and corrected a long-standing measurement bias. Each is recorded as a single git commit in the lessons-learned repository:

- `52790e9` — **jq syntax error in `ll-merge-findings.sh`.** A malformed `jq` expression caused the lessons-learned daemon to exit non-zero on every run since 2026-03-31 (16 consecutive missed runs). The fix restored the merge step and back-filled the missed analyses.
- `a9d0de3` — **SYS_TRACEABILITY metric scoped to runs with concerns.** The original formula divided by all 70 runs, including SEED-only and CANCELLED runs that had no concerns to trace. Scoping the denominator to runs that actually had concerns lifted SYS_TRACEABILITY from approximately 0.32 to approximately 0.92 and back-filled 27 runs with the corrected score.
- `e709294` — **Proactive learnings collector.** New script `bin/ll-collect-learnings.sh` proactively sweeps session logs and tool-call traces for learning signals, replacing the previous reactive collection that depended on agents writing to specific tables. Closes the SYS_SESSION_LEARNING_RATE = 0 anomaly that had persisted across W1–W29 of Run #1027.
- `b5dbb1f` — **Post-INSERT data quality auditor.** New script `bin/data-quality-auditor.sh` performs post-INSERT detection of hallucinated-schema writes (rows that pass `INSERT` but reference columns or enums that drift from the live schema), generating corrections rather than crashing on the next read.
- `356b583` — **Drift-detector decimal overflow clamp + heartbeat scope fix.** The drift detector overflowed `DECIMAL(5,4)` when computing percentage drift on small denominators; the fix clamps to $[0, 9.9999]$ and corrects the heartbeat scoping that had been double-counting cancelled runs. Companion fixes to `experiment-runner` and the PCD morning report shipped in the same commit.

Concept-to-X Pipeline Census — 19 Active

The concept-to-X pipeline platform now has 19 active pipelines registered in `fdrp_c2x_pipelines` (status = `ACTIVE`), up from the 14 cited in v20.0. The five additions extend FDRP into competition-style and intelligence workflows: `concept2smoke`, `concept2features`, `concept2ensemble`, `concept2triage`, `concept2intel`. The full active list is: `concept2image`, `concept2onepager`, `concept2prd`, `concept2demo`, `concept2website`, `concept2mvp`, `concept2full-software`, `concept2experiment`, `concept2payload`, `concept2digitaltwin`, `star2paper`, `star2mail`, `expert-research`, `concept2smoke`, `concept2features`, `concept2ensemble`, `concept2triage`, `concept2intel`, plus `concept2simulation` in `DESIGN` status.[5]



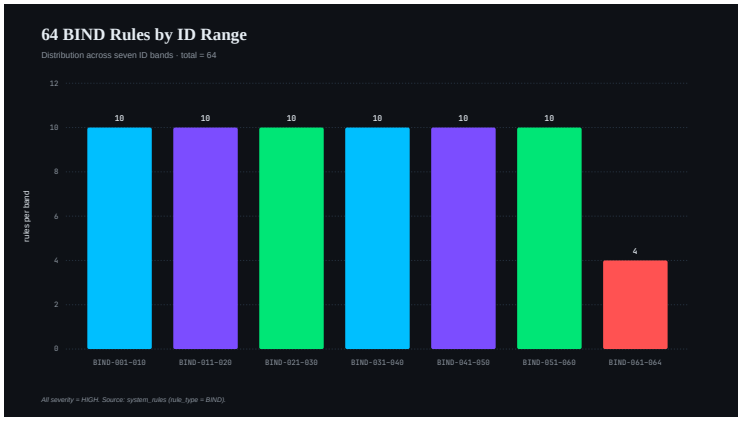
Interactive: https://fdrp.liviu.ai/gallery/figures/fig-071-c2x-pipelines.svg

BIND Rule Inventory and Schema Snapshot

The system rule inventory now stands at 64 BIND rules (BIND-001 through BIND-063, plus historical exceptions), governing research, evidence, backups, model selection, governance tiers, cross-model verification, file-based prompting, verification receipts, the PIO router pattern, and the four operational rules described above.[6] The schema snapshot stands at 593 base tables and 253 views (3.33 GB total). The reduction from the v20.0 figure of 745 base tables reflects the consolidation work performed during Run #1027 W2–W6 (zero-row tables identified, pruned, and merged into properly-keyed parent tables) rather than feature loss; the 19 active concept-to-X pipelines and 79 registered skills exceed the v20.0 inventory.

[5] `SELECT pipeline_id, name, status FROM fdrp_c2x_pipelines ORDER BY pipeline_id` returned 19 ACTIVE rows and 1 DESIGN row, measured 2026-04-16.

[6] `SELECT COUNT(*) FROM system_rules WHERE rule_id LIKE 'BIND%'` returned 64, measured 2026-04-16.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-072-bind-rules.svg

Summary of Results

Core Framework Conclusions

FDRP demonstrates that manufacturing quality principles — SPC, 5S, FMEA, Andon, configuration freeze — can be systematically applied to the planning process itself. By treating decisions as manufactured artifacts with traceability, clash detection, convergence measurement, and gate reviews, FDRP achieves levels of planning rigour inspired by CERN engineering reviews and aviation certification programmes.

The system is not theoretical. It has processed 1,968 decisions across 80 runs (28 commissioned, 18 cancelled via triage, 24 CONVERGED; snapshot 2026-06-10), detected 160 clashes, maintained SPC control charts with Nelson Rule violation detection, achieved 97.7% self-assessed compliance against CERN-derived clauses (43/44), produced 8 interactive visualisations and 76 publication-quality figures from a reusable CLI tool, implemented a dual-timescale self-evolution subsystem with cross-model peer review (its slow loop retired 2026-06-10 at a measured 0.12 % yield — Act III), provided preliminary evidence for its core sparse-principle thesis across two scientific domains (earthquake seismology and power grid stability), extended its quality loop to infrastructure security via the CyberDefence subsystem, and discovered that expert persistence — treating AI sessions as persistent knowledge containers with git-like lifecycle operations (the earlier "appreciating assets" framing was narrowed in v23.0 to its measured part: context retention reduces marginal cost) — reduces multi-round expert panel API token costs by an estimated 92% (derived from pricing assumptions; see Section 26 economics) while enabling a qualitatively different mode of iterative refinement. The expert registry now tracks 446 experts across 217 domains with 8,571 extracted expert findings (fdrp_expert_findings; the 8,554 figure conflated tables — see drift retraction) connected by 6,119 edges in structured tables with automated contradiction detection. The ultrathink cascade methodology, validated through the CERN antimatter building programme (32 experts, 28,936 lines of output), demonstrates that structured multi-expert dispatch produces emergent synthesis (the Expert Knowledge Operating System concept) that no individual expert could generate alone. The knowledge graph formed by finding similarity crossed the percolation threshold (avg degree 4.90 among connected nodes, 3.08 global; SUPER_CRITICAL phase; 2,547 edges across 1,039 connected nodes out of 1,656 total), and a three-wave expert panel process (optimise → audit → fix+prevent) exposed 7 fatal defects in schema change infrastructure, closed all 7, hardened 4 scripts against SQL injection, and converted schema quality testing from discretionary expert activity to a 12-hour daemon action with SPC monitoring (it ran as such until the 2026-06-10 flywheel retirement; the deterministic check survives on demand). Most recently, v0.14.0 achieved the first batch convergence in FDRP history — 6 runs simultaneously transitioning to CONVERGING status with CVT values 0.172–0.191, with all subsequently reaching CONVERGED status, culminating in v19.0's full portfolio convergence: 19 CONVERGED (15 at 1.0000), 0 CONVERGING, 0

ACTIVE — and exposed the fake bulk-set constitution principle scores; the replacement empirical measurement reported 10 PASS / 14 PARTIAL / 4 FAIL on 2026-03-25 but was itself RETRACTED 2026-05-13 (source scores irrecoverable — see Contribution 35), so the honest self-assessment is the exposure of the fake scores, not the since-withdrawn replacement result.

What distinguished FDRP's original framing from static planning methodologies was its **self-evolving nature** — the claim that the system does not merely plan projects but plans its own improvement. Act III's measurements (2026-06) retired that claim's strong form: the self-improvement flywheel yielded 0.12 % true promotions and its timers were operator-disabled on 2026-06-10. The earned value is concept2X production at program scale; the self-application survives as the method that produced the honest measurements. Each production run generates lessons that feed back into the system's pattern catalogue, heuristic database, and expert roster. Expert expansion waves discover new domains of expertise that the system's creators never anticipated. The Wave 2 panel (40 LLM-generated expert personas in that analysis, subsequently expanded to 68+ in the antimatter building programme) identified 7 architectural blind spots (Game Theory, Adversarial ML, Temporal Databases, Process Mining, Petri Nets, Control Theory, Schema Evolution) — none of which were in the original architectural specification. Expert count is bounded by convergence, not by a predetermined cap.

The Digital Cognition Architecture (Section 12) represents the most ambitious next step: expanding FDRP's cognitive vocabulary beyond analytical reasoning into 12 distinct thinking modalities — from dialectical synthesis to biological fitness landscapes to phenomenological experience analysis. Grounded in established cognitive science (Global Workspace Theory, Society of Mind, BVSR), this architecture does not merely add new features. It makes the system capable of **discovering new ways to think** through its own expert expansion process. When an evolutionary biologist critiques an FDRP run, they don't just comment on the content — they reason differently. That reasoning pattern becomes a new modality candidate, tested against historical decisions and, if effective, integrated into the system's permanent cognitive toolkit.

The unified metaphor — Diamond × Helix × CVT × Ouroboros — provides both conceptual clarity and operational precision. The diamond quality goal drives refinement. The helix trajectory enables fractal descent through scales. The CVT dynamics manage the exploration-exploitation transition. And the ouroboros self-reference survives as the measurement loop rather than an autonomous flywheel (retired 2026-06-10 at a measured 0.12 % true-promotion yield, Act III): the system consumes its own experience by measuring itself and keeping only what measurement proved.

This self-referential property makes FDRP a candidate architecture for FCC-class projects — endeavours spanning decades, crossing disciplinary boundaries, and evolving faster than any static plan can anticipate. A planning system that continuously discovers what expertise it is missing, what thinking modalities it

lacks, grounds its decisions in current research, and applies the same quality mechanisms to its own evolution as to its subjects is not a luxury for such programmes — it is a necessity. Like shaping clay: you start with your hands, then discover the paddle, then the wire, then the wheel. FDRP is the potter who never stops discovering new tools.

The March 2026 operational cycle added three critical dimensions. First, the system's self-correction infrastructure was itself corrected: the anti-pattern detection engine was found to have 0% actual trigger rate (23 of 31 rules silently broken by schema drift), the correction governance pipeline had accumulated an unmanageable 40:1 pending-to-applied ratio, and a systematic audit revealed that all 28 constitution principle scores were fake (bulk-set to 0.80). These are not failures of the methodology but demonstrations of its self-diagnostic capability — FDRP's own quality mechanisms, when properly operational, detected FDRP's own deficiencies. Second, the system's first commercial deployment (Run #1015, an anonymized Gulf hospitality procurement client) demonstrated that the methodology transfers from its CERN/physics origin to commercial B2B engagement, with the 7-level fractal hierarchy mapping directly to procurement decision scales, the audience-adaptive translation producing a client-facing methodology paper, and the Oracle Seasons pivot (CVT 0.900 → 0.676) demonstrating that FDRP catches premature capability commitments through its Andon mechanism. Third, the first batch convergence (6 runs simultaneously transitioning to CONVERGING with CVT 0.172–0.191) established that the system can converge reliably at scale. The first empirical constitution testing reported 10 PASS / 14 PARTIAL / 4 FAIL but was RETRACTED 2026-05-13 (source scores irrecoverable — see Contribution 35); it therefore establishes nothing about principle adherence — what survives from it is the exposure-and-retraction discipline itself.

The April 2026 session (v20.0) added a fourth critical dimension: **cognitive self-awareness**. Nine new subsystems (#24–#32) added 66 tables in a single session, growing the schema from 593 to 745 base tables and from 132 to 188 views (fdrp_ tables: 156 → 360). The cognitive architecture subsystem (#28, 18 tables) gives the system a formal taxonomy of how it thinks — 26 architect archetypes, 21 modality registries, 12 cognitive profiles. The thinking chains subsystem (#27, 15 tables) makes reasoning patterns first-class versioned objects with 15 topology types. The daemon taxonomy (#29, 7 tables) unifies 35 daemon types under a common lifecycle model. The wiring-check subsystem (#31, 4 tables) performed 271 verification checks and found 18 missing integration wires. The limits-hunt subsystem (#32, 5 tables) discovered 68 resource limits including innodb_buffer_pool_size at 128MB on a 62GB node. The system now has 32 registered subsystems — it can reason about its own reasoning, govern its own daemons, and verify its own structural integrity.

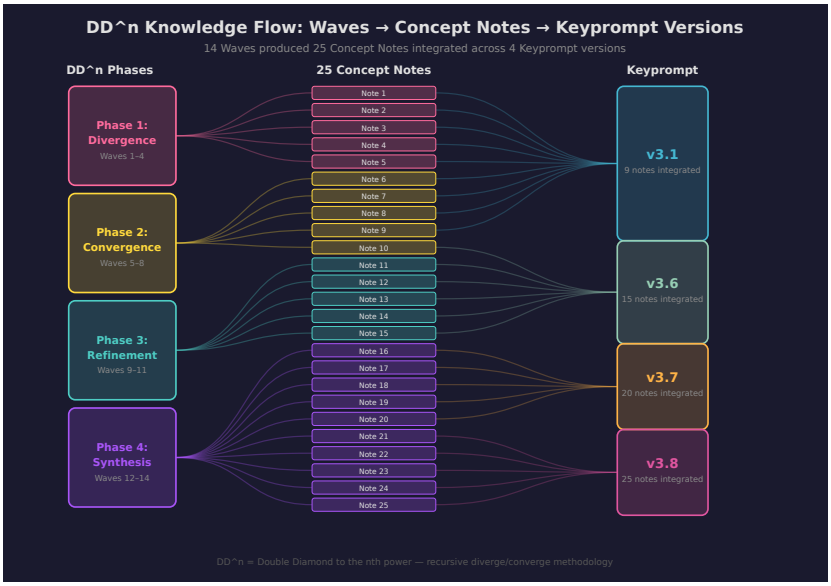
Grounding note: Numeric claims in this summary are drawn from production MySQL queries unless otherwise qualified. Cost reduction (92%) is derived from token pricing assumptions (Section 26). Compliance (97.7%) is

self-assessed against self-interpreted clauses (Section 17). CVT convergence range is from N=2 fully-iterative runs only (Section 18). Anti-pattern trigger rate (26%) from state/anti-pattern-detect.log, 2026-03-25. Correction ratio (3:808) from directory counts, 2026-03-25. Batch convergence CVT values (0.172–0.191) from `fdrp_convergence`*, 2026-03-25; all 6 batch runs subsequently CONVERGED (score 1.0000). Full portfolio convergence (19 CONVERGED, 0 CONVERGING) from* `fdrp_runs`*, 2026-03-26. Constitution test results (10/14/4) from SQL queries against* `<intelligence_db>`*, 2026-03-25. v20.0 schema counts (745 tables, 188 views, 360 fdrp_ tables, 32 subsystems) from* `information_schema` *and* `fdrp_subsystems`*, 2026-04-02. New subsystem row counts (68 limits, 271 wiring checks, 40 archetypes, 55 evidence gates, 35 daemon types) from respective fdrp_ tables, 2026-04-02. All [UNGROUNDED] labels from the body text apply to any figures restated here. Heijunka alpha (0.3) and CVT weights (0.25 each) are [UNCALIBRATED] per BIND-021.*

DD^n Analysis and Principle Lifecycle

*__Reproducibility note__: The DD^n process artifacts (25 concept notes, wave transcripts, cross-model review prompts and responses) reside in the project repository at `<shared_root>/fdrp-double-diamond/`. The process itself — iterative concept extraction through structured expert dispatch with cross-model verification — is described sufficiently for replication at the level of methodology, though the specific LLM-generated outputs are not deterministically reproducible across model versions.

The following four subsections — Note Evolution, Resource-Conscious Deployment, Cross-Model Verification, and Three-Tier Protocol Split — document the DD^n concept note analysis and quality assurance process that produced the FDRP Unified Keyprompt v3.8.*



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-067-dd-sankey.svg

Note Evolution as System Property

The DD^n analysis that produced the FDRP Unified Keyprompt through 14 waves and 4 phases has a creation ritual — each session generates an insight, the insight is captured as a concept note, and the note is integrated into the keyprompt — but it has had NO maintenance ritual. Concept notes accumulate like sedimentary layers: twenty-five notes exist because twenty-five sessions produced insights. Nothing in the system previously asked whether these twenty-five should still be twenty-five, whether some should be merged into fewer, or whether the container structure still serves the ideas inside it. This is the same failure mode as software systems that grow features without refactoring. The code works, each feature is individually correct, but the STRUCTURE degrades until the system becomes harder to navigate and maintain than it needs to be.

Concept note #24 (Evolution Protocol) addresses this gap by introducing four structural problems and their solutions:

Problem 1: Creation Without Revision. Every note was written once, during the session that discovered the insight. None had been revised since creation. Note #8 (Motivation Payload) was written before #21 (Average Landmine) existed and contains a claim about an A/B test showing +22% improvement — but #21 subsequently established that N=2 is not statistically significant. The note was never annotated after #21 was created because no revision mechanism existed. Similarly, notes #3 (Semantics Perfectionist) and #20 (Semantic Catchers) address identification and enforcement of the same problem (dangerous words) but neither referenced the other — two halves of one idea split across two sessions with no cross-linking mechanism.

Problem 2: No Version History at the Semantic Level. Notes have git history (they reside in `<shared_root>/fdrp-double-diamond/`), so file-level versioning exists. But git tracks character-level changes — a typo fix and a fundamental mechanism revision look identical in the git log. The notes lack SEMANTIC versioning: tracking changes to the MEANING, not just the text.

Problem 3: Naming Inconsistency. Notes 1–4 use `*-CONCEPT.md` in their filenames (GROUNDING-N-CONCEPT, PERFECTIONIST-CASCADE-CONCEPT, etc.) while notes 5–25 use `*-PRINCIPLE.md`. The naming implies a distinction that was never defined. In practice, the convention shifted after the first session. The resolution: do NOT rename existing files (the cost of breaking references exceeds the benefit of consistency, per the Original Source Principle #5 — the original naming IS data about the system's history), but standardise going forward.

Problem 4: No Decomposition Mechanism. Some notes contain multiple independent ideas packaged together because they emerged in one session. Note #21 (Average Landmine) contains three separable ideas: the anti-averaging principle (MIN over AVG), the Statistician Observer role definition, and the multi-model architecture for statistical verification. When a note contains multiple ideas, evolution becomes confused — which idea is being updated? Which should be deprecated?

The Evolution Framework The protocol defines six maintenance operations, ordered by expected frequency:

Operation	Trigger	Key Test
ANNOTATE	New evidence, cross-reference to newer note	Core insight unchanged?
QUALIFY	Feedback narrows scope	Principle meaning unchanged?
REVISE	New understanding of mechanism	DD^n discipline applied?
MERGE	Two notes address same phenomenon, neither complete without the other	Evidence inseparable?
SPLIT	Note contains multiple independent ideas needing different lifecycle management	Ideas at different scales?
DEPRECATE	Note superseded, absorbed, or structurally unfit	Successor identified?

The key decision test for UPDATE vs CREATE NEW: does the core insight remain the same? If yes, update. If the new insight would make sense even if the existing notes did not exist, create a new note. The danger zone is "note inflation" — creating new notes when existing ones should be updated, growing the count without improving the system.

Staleness Detection Six indicators detect when a note has fallen behind the system's current state:

Indicator	Description	Detection Method
Vocabulary drift	Note uses terminology the system no longer uses	Grep notes for deprecated terms
Assumption violation	Note assumes conditions that no longer hold	List assumptions per note; check which still hold
Reference rot	Note references files, tables, or systems that have been restructured	Validate all paths and table names
Unlinked island	No incoming references from newer notes, no outgoing references to newer notes	Build reference graph; identify isolated nodes
Evidence stagnation	Evidence base not updated despite 10+ applicable runs	Check evidence card from #23
Practice divergence	System routinely does something DIFFERENT from what the note prescribes, with better results	Compare prescriptions to actual practice

Staleness is NOT invalidity. A stale note may still be correct — just outdated in framing. Staleness demands REVIEW, not deprecation. The review may conclude the note is still current.

Semantic Versioning for Notes Each note maintains a change log with semantic version numbers:

- **MAJOR** (X.0): Core insight changes. Mechanism rewritten. Scope fundamentally altered.
- **MINOR** (X.Y): Annotations, qualifications, cross-references, example additions.

Change types map to the maintenance operations: CREATED, ANNOTATED, QUALIFIED, REVISED, WEAKENED, SPLIT, MERGED, DEPRECATED.

Candidate Merges The self-analysis identified two strong merge candidates:

1. **#3 Semantics + #20 Catchers** — Identification and enforcement of the same problem (dangerous words). Knowing a word is dangerous (#3) without enforcement (#20) is a frozen taskbar. Enforcing without identification is impossible. They are one idea split across two sessions. Proposed successor: SEMANTIC-ENFORCEMENT-PRINCIPLE.
2. **#8 Motivation + #11 Detraining** — Theory and testing of the same mechanism (training default override). Motivation without detraining verification is an ungrounded claim. Detraining without motivation is verification of what? Proposed successor: TRAINING-OVERRIDE-PRINCIPLE.

A third pair (#10 Constraints + #13 Boundary-Breaker) was evaluated but deemed sufficiently independent — cross-link rather than merge.

Note Evolution vs Principle Lifecycle: Orthogonal Dimensions The Feedback Loop Principle (#23) and the Evolution Protocol (#24) address DIFFERENT questions:

- **#23 (Lifecycle)**: Is this principle TRUE? Does evidence support it? Stages: PROPOSED → TESTED → VALIDATED → FOUNDATIONAL → DEPRECATED. Changes based on EVIDENCE.
- **#24 (Evolution)**: Is this note still the right CONTAINER for this idea? Stages: CREATED → ANNOTATED → QUALIFIED → REVISED → SPLIT/MERGED → DEPRECATED. Changes based on STRUCTURE.

These dimensions are orthogonal, producing a 2$×$2 matrix:

	Note CURRENT	Note STALE
Principle VALIDATED	Healthy (working as intended)	Needs update (valid but outdated framing)
Principle PROPOSED	Hypothesis (untested but clearly stated)	Dead weight (untested AND outdated)

A principle can be VALIDATED (strong evidence) while its note is STALE (outdated terminology, missing cross-references). A principle can be PROPOSED (no evidence) while its note is CURRENT (freshly written, well-structured). Both dimensions must be managed independently.

Connection Table: How Each Principle Relates to Evolution

#	Principle	Evolution Connection
1	Grounding^n	Note revisions must be GROUNDED — changes based on evidence, not aesthetics
2	Perfectionist Cascade	A `note-librarian` perfectionist dimension: "is this note still serving its purpose?"
3	Semantics Perfectionist	Evolution must catch SEMANTIC DRIFT between notes written months apart
4	Means Axis	Tools for evolution: git (versions), grep (staleness), reference graphs (dependencies)
5	Original Source	Original text preserved in git history; revisions that contradict the original insight should be NEW notes
6	No Arbitrary Maximums	Note count determined by CONVERGENCE; evolution provides MERGE and DEPRECATE to prevent unbounded growth
7	Compounding Leverage	Well-maintained notes compound: clear notes → better agents → better outputs → better feedback → better notes
10	Constraint Taxonomy	Note evolution addresses constraints at ALL 7 levels (numeric, verbal, structural, paradigmatic, medium, cognitive, meta)
12	Multi-Scale Diffusion	Evolution operates at multiple scales: individual note, note pairs, note clusters, full system
13	Boundary-Breaker	"Notes are permanent once written" IS a frozen taskbar — revealed by the evolution principle
14	CVT Perfection	CVT_evolution = MIN(staleness_detection, cross_reference_completeness, naming_consistency, change_log_coverage)
15	Gear Pipeline	Evolution operates at SLOW gear (changes accumulate over runs); ANNOTATING is FAST gear (immediate)
19	Multi-Model	Note evolution decisions should be reviewed by MULTIPLE models to prevent model-specific structural biases
21	Average Landmine	System structural health determined by its MOST stale, MOST inconsistent note (MIN logic)

#	Principle	Evolution Connection
22	Failure Mode	Evolution can FAIL: bad merges lose nuance; premature deprecation removes load-bearing principles
23	Feedback Loop	#23 drives content decisions ("weaken this principle"); #24 drives structural decisions ("merge these notes")

Resource-Conscious Deployment

The concept note system contains 25 principles, all prescribing additional work: more recursion, more verification, more models, more perfectionists, more semantic catchers, more feedback loops. No principle previously said "STOP — you have exceeded your budget." The system had twenty-four notes telling agents to go deeper and zero notes telling agents when to surface.

This is not a failure of ambition but a failure of engineering. Every engineering system operates under constraints. The concept note system identifies constraints brilliantly (#10 Constraint Taxonomy, seven levels) but never applied constraint analysis to ITSELF.

The 1M context inflection point. With Opus 4.6's 1M token context window at standard pricing ($5/MTok input), the primary constraint shifts from token budget to *attention budget* — the model's ability to maintain focus across increasingly large contexts (76% needle-in-haystack at 1M). The full keyprompt (~90K tokens [UNGROUNDED — estimated]) now consumes less than 10% of available context, removing the "principles crowd out work" failure mode that motivated the original triage design. However, the triage tiers remain valuable for a different reason: not because context is scarce, but because *attention is finite*. Loading 25 principles into a 1M window does not guarantee the model attends to all 25 with equal fidelity. The resource-conscious deployment framework therefore reframes from "what fits?" to "what will the model effectively attend to?" — a subtler but equally important constraint.

Concept note #25 (Resource Budget Discipline) fills this gap with an explicit cost framework.

Four Resource Constraints

Constraint	Nature	Impact
Token budget	Each ^n recursion roughly doubles token cost; VERIFY^5 costs ~62,000 tokens [UNGROUNDED — estimated, not measured via tokeniser] before any output. At Opus 4.6 pricing ($5/MTok input, $25/MTok output), this is ~$0.31 input + ~$1.55 output per VERIFY^5 cycle	Exponential cost growth limits recursion depth; however, with 1M context at $5/MTok, the per-token cost is lower than earlier model generations, shifting the constraint from "can we afford the tokens?" to "does deeper recursion improve quality?"
Model API cost	Full principle compliance at ^3 depth across 3 models: ~$337.50 per run with 10 expert dispatches [UNGROUNDED — derived from pricing assumptions, not measured from API billing]	Governance cost can exceed production value
Human attention	The principal reviews outputs, approves corrections, reads reports; 25 principles × attention per principle = attention overload	Random triage replaces systematic prioritisation

Constraint	Nature	Impact
Context window / attention budget	Keyprompt ~15K tokens + 25 full notes (~75K tokens) [UNGROUNDED — estimated, not verified via tokeniser] consumes ~90K of a 1M token window (Opus 4.6 GA). Raw capacity is no longer the binding constraint: ~910K tokens remain for reasoning. The binding constraint is now *attention quality*: at 76% needle-in-haystack accuracy at 1M tokens, the model's ability to attend to all loaded principles degrades with context depth. The effective attention budget — not the token budget — determines how many principles the model can meaningfully apply	Attention degradation, not capacity exhaustion, limits principle density

Triage Tiers

Tier	Principles	Token Budget	Cross-Model	Recursion	When to Use
MINIMUM	3 (#1, #10, #7)	~5K	No	^2 max	Budget < $1/run, time < 5 min, new domain, or attention-constrained contexts ($>$500K tokens already loaded)
STANDARD	5 (add #5, #12)	~10K	No	^2	Normal operations, internal drafts
FULL	All 25	~75K+	Yes (3 models)	To convergence	Safety-critical, final deliverables, system-level changes. With 1M context, FULL is now feasible for all tasks where attention quality permits

The MINIMUM tier preserves: truth discipline (#1 Grounding), boundary detection (#10 Constraints), and investment rationale (#7 Compounding). These three principles are the irreducible core. The STANDARD tier adds provenance discipline (#5 Original Source) and processing architecture (#12 Multi-Scale Diffusion) — the five essential generators identified by compression testing. What the 5 generators do NOT cover: enforcement (no catchers, no automation), verification diversity (no multi-model), recovery (no failure modes, no feedback), statistical rigour (no anti-averaging), and process discipline (no gear matching).

Contextual Priority by Task Type Different tasks demand different principle subsets:

Task Type	CRITICAL Principles	SKIP Principles
Analysis	#1 Grounding, #5 Source, #21 Anti-Average	#8 Motivation, #18 Creation-DD^n, #15 Gear
Creation	#18 Creation-DD^n, #2 Perfectionist, #1 Grounding	#17 Ext. Validation, #21 Anti-Average, #23 Feedback

Task Type	CRITICAL Principles	SKIP Principles
Validation	#1 Grounding, #5 Source, #19 Multi-Model	#8 Motivation, #7 Compounding, #9 Reorganisation
Meta-Process	#23 Feedback, #24 Evolution, #25 Budget	#4 Means, #8 Motivation, #15 Gear

Deployment Sequencing When applying the system to a new domain, principles should be introduced in dependency order across five phases:

1. **Foundation** (first 2 runs): #1 Grounding, #5 Original Source, #7 Compounding. Purpose: establish truth discipline, provenance tracking, investment mindset. Budget: ~30%.
2. **Structure** (runs 3–5): Add #10 Constraints, #12 Multi-Scale, #6 No Arbitrary Maximums. Purpose: analytical framework for boundaries and multi-scale processing. Budget: ~20%.
3. **Process** (runs 6–10): Add #2 Perfectionist, #8 Motivation, #15 Gear, #16 Ordering. Purpose: process discipline. Budget: ~20%.
4. **Enforcement** (runs 11–15): Add #3 Semantics, #20 Catchers, #11 Detraining, #13 Boundary-Breaker. Purpose: detection and enforcement. Budget: ~15%.
5. **Meta** (runs 16+): Add remaining principles (#14, #17, #18, #19, #21–#25). Purpose: measurement, validation, failure handling, feedback, evolution, budgeting. Budget: ~15%.

Each phase builds on the previous — enforcement cannot precede establishment, measurement cannot precede application, budgeting cannot precede experience.

ROI Ranking ROI = (quality improvement) / (cost of applying). Since quality improvement is unmeasured for most principles (see #23's honest assessment: 0/25 VALIDATED), only relative ROI can be estimated [UNGROUNDED per BIND-021]:

Tier	Principles	Rationale
HIGHEST ROI	#7 Compounding, #10 Constraints, #1 Grounding	Negligible-to-low cost, high potential impact
HIGH ROI	#5 Original Source, #20 Semantic Catchers	Low cost (provenance check, automated regex)
MODERATE ROI	#2 Perfectionist, #8 Motivation, #13 Boundary-Breaker, #16 Ordering, #15 Gear	Low-moderate cost, uncertain impact

Tier	Principles	Rationale
LOWER ROI	#19 Multi-Model, #14 CVT, #17 External Validation	High cost (3× model calls, external papers), uncertain impact

These rankings are PROPOSED orderings based on theoretical analysis, not measured outcomes. The Feedback Loop (#23) would convert these estimates into evidence.

The Meta-Cost The act of deciding which principles to apply itself consumes ~4,000 tokens [UNGROUNDED — estimated, not measured via tokeniser]. With 1M context windows, this overhead is negligible in token terms ($<$0.5% of capacity), but the *attention cost* of triage remains relevant — 4,000 tokens of meta-reasoning is 4,000 tokens not spent on the primary task. Rule of thumb:

- Budget tight (any resource $< 50\%$ of full needs): Apply the budget principle. Triage saves more than it costs.
- Budget abundant (all resources $> 200\%$): Skip the budget principle. Apply everything. With 1M context at $5/MTok, token budget is rarely the binding constraint; check attention budget and time budget instead.
- Budget moderate (50–200%): Use the contextual priority table. Skip full triage.

Cross-Model Verification of the Keyprompt

The FDRP Unified Keyprompt v3.6 (1,049 lines) was independently reviewed by Codex Pro (GPT-5.4) and Gemini 3.1 per BIND-008 (Aviation-Inspired Dual Verification) and BIND-046 (Expert-Framed Cross-Model Verification). Each model received the original document directly — not another model's framing. This section reports the findings that drove the v3.7 revision.

Methodology Codex Pro reviewed the full 1,049-line document. Gemini 3.1 reviewed keyprompt sections 7.9 (Failure Modes) and 7.10 (Feedback Loop) separately in two prompts due to its 4KB input limit. Scores use MIN aggregation per the protocol's own Average Landmine principle (#21). Note: using the system's own scoring framework to evaluate the system introduces circular bias; external models were asked to evaluate using criteria defined by the keyprompt they are reviewing.

Where Models Agree (5 High-Confidence Findings)

1. **Severity count error** (Keyprompt Section 7.9): The severity summary states "4 CRITICAL" but lists 5 failure mode IDs (FM-1.1, FM-1.2, FM-2.2, FM-2.3, FM-4.1). States "4 MEDIUM" but lists 3 IDs (FM-3.1, FM-3.2, FM-3.3). Correct distribution: 5 CRITICAL, 2 HIGH, 3 MEDIUM.

2. **Missing failure mode categories**: Both models agree the catalog is incomplete — no coverage of human operator failures (misclassifying Cynefin domain, ignoring dissent, alert fatigue), environmental/infrastructure failures (model capability drift, API timeouts, silent context truncation), or feedback loop failures (attribution confounding, hallucinated compliance, feedback echo chamber).

3. **Feedback loop implementability**: Multi-variate attribution on 20+ co-applied principles is statistically infeasible as originally specified. Gemini characterised it as "practically unexecutable for an AI agent"; Codex identified attribution confounding as an architectural risk.

4. **Checklist not self-sufficient**: An agent cannot follow only the checklist (Keyprompt Section 13) and produce quality output. Items depend on definitions outside the checklist and are phrased as open-ended philosophical questions rather than deterministic telemetry checks.

5. **FOUNDATIONAL term collision**: "FOUNDATIONAL" used both as a principle lifecycle stage (Keyprompt Section 7.10) and as a priority class (FM-4.2: FOUNDATIONAL > STRUCTURAL > SPECIALISED). Locally intelligible but not globally normalised.

Where Models Disagree (4 Findings Requiring Investigation)

1. **Feedback loop severity**: Codex rates integration as "partially successful" with addressable design issues. Gemini rates implementability at 1/5, calling the design "not practically implementable." Resolution: Gemini's concern is more operationally specific (agents cannot reliably perform statistical math); Codex's concern is more architectural (loop closes on potentially circular evidence). Both are valid at different levels.

2. **Protocol classification split**: Codex recommends splitting the protocol into three tiers (safety kernel / operational / experimental). Gemini recommends domain-specific lifecycle tracking (CONDITIONALLY VALIDATED stage). These are complementary, not contradictory — both are addressed in v3.7.

3. **Disagreement quota conflict**: Codex flags that the mandatory disagreement quota (each model must find at least 2 findings not in primary analysis) can force synthetic dissent after substantive convergence, conflicting with anti-theatre doctrine. Gemini did not flag this (narrower review scope).

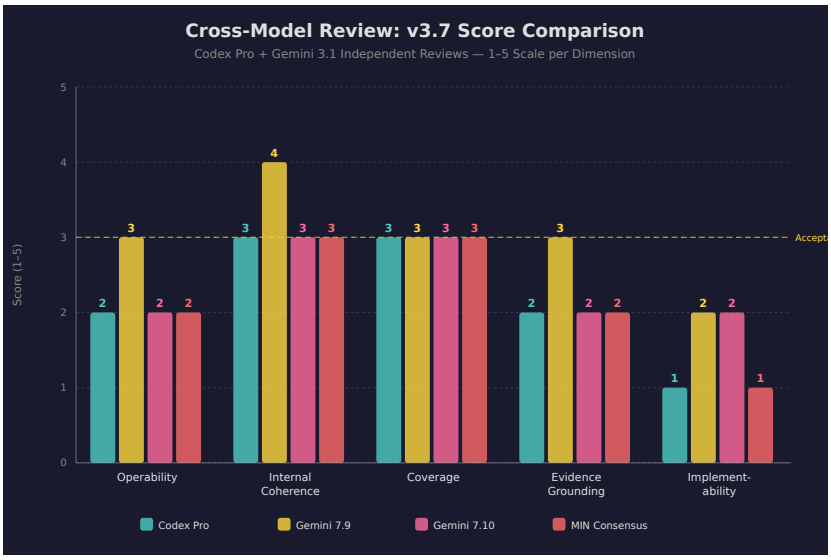
4. **Motivation payload timing**: Codex identifies a control-loop timing issue (detect-after-output vs prevent-before-dispatch). Gemini did not flag this (sections 7.9/7.10 only, not section 7.1).

Consolidated Scores

Dimension	Codex (GPT-5.4)	Gemini 3.1 (7.9)	Gemini 3.1 (7.10)	MIN
Consistency	2	–	–	**2**
Completeness	3	3	3	**3**
Implementability	3	4	1	**1**
Grounding	2	4	3	**2**
Coherence	3	5	4	**3**

OVERALL: MIN(2, 3, 1, 2, 3) = 1/5

The MIN=1 is driven entirely by Gemini's implementability assessment of section 7.10 (Feedback Loop). If section 7.10 is redesigned per recommendations, the floor rises to MIN(2, 3, 3, 2, 3) = 2/5.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-065-cross-model-review.svg

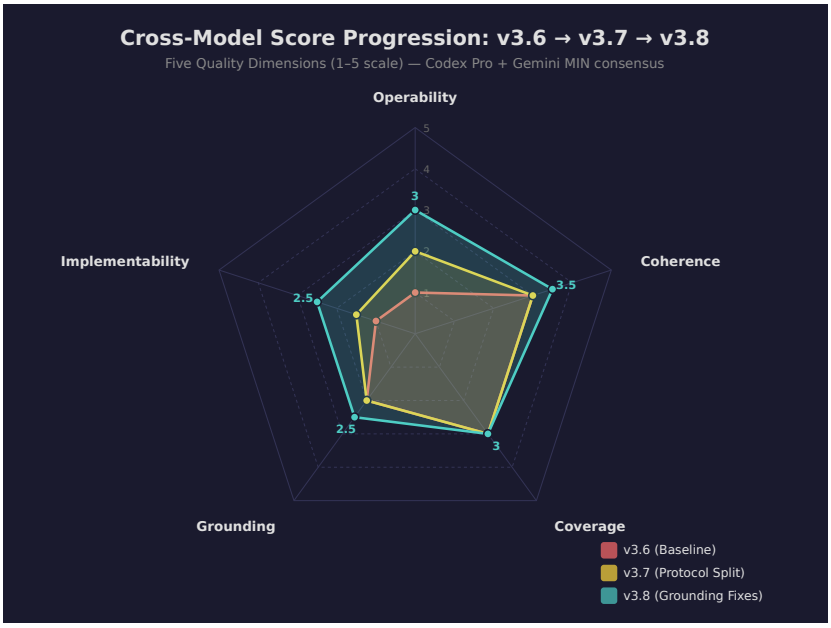
Action Items Summary

Severity	Count	Key Items
CRITICAL	6	Severity count error, lifecycle count mismatch, feedback loop not implementable, 7.8/FM-4.1 priority conflict, incomplete threshold classification, missing safety kernel tier

Severity	Count	Key Items
HIGH	9	Missing failure mode categories (human/environmental/feedback), motivation timing, disagreement quota conflict, FOUNDATIONAL collision, checklist not self-sufficient, missing pre-run prediction, domain-dependent lifecycle, Bayesian tooling unspecified, missing checklist items
MEDIUM	6	Stale "21 concept notes" reference, manual external validation, principle interaction failures, grounding annotations, evidence weighting, provisional threshold tagging

All 6 CRITICAL and 9 HIGH findings are addressed in v3.7. The key insight from this review is that the feedback loop requires DETERMINISTIC tooling — Design of Experiments (DOE) ablation schedules and offloaded computational scripts — rather than LLM-based attribution. Agents collect telemetry; scripts perform the statistical math.

v3.7 Cross-Model Review Results The v3.7 cross-model review (Codex Pro + Gemini 3.1 + Claude Opus 4.6) scored MIN 1.5/5 strict, 2/5 adjusted. While operability improved from 1 to 2 and failure coverage from 2 to 3, the resource budget discipline introduced a new consistency contradiction (budget tiers violating the Safety Kernel's "No Arbitrary Maximums" principle). Additionally, self-reported outcome_score was identified as creating a positive feedback loop. These findings drove 4 CRITICAL + 9 HIGH fixes in v3.8, demonstrating the cross-model review cycle's ability to catch self-contradictions that emerge from new content integration.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-069-cross-model-radar.svg

Three-Tier Protocol Split

Cross-model review finding C-6 exposed a fundamental tension in the keyprompt: the protocol simultaneously uses language of obligation ("must," "prohibited," "non-negotiable") and language of provisionality ("0/25 VALIDATED," "the system runs on theory, not evidence"). Without distinguishing between these, an agent cannot determine which principles are truly mandatory and which are experimental hypotheses that should be tested before enforcement.

v3.7 addresses this by splitting the protocol into three tiers:

Safety kernel (non-negotiable, every run): Principles that MUST be applied regardless of context, budget, or evidence status. Candidates: #1 Grounding^n (truth discipline is prerequisite to all other quality), #5 Original Source (provenance tracking prevents evidence contamination), #10 Constraint Taxonomy (boundary awareness prevents catastrophic scope violations). These three correspond exactly to the MINIMUM triage tier from the Resource Budget Discipline (#25). A principle enters the safety kernel only when it has been VALIDATED through controlled evidence AND its violation has been shown to produce harmful outcomes.

Operational heuristics (testable, evidence expected): Principles with structural plausibility and preliminary evidence of effectiveness. Most of the 25 principles belong here. These are ACTIVE — agents apply them — but they carry an implicit qualification: "we believe this works based on [evidence level], and

we are collecting more evidence." An operational heuristic can be contextually skipped when the Resource Budget requires triage (STANDARD or MINIMUM tier). The expectation is that operational heuristics accumulate evidence over runs and either graduate to the safety kernel or are deprecated.

Experimental principles (provisional, hypothesis status): Principles that are PROPOSED but not yet TESTED (per #23's honest assessment: 0/25 currently VALIDATED, 16/25 still PROPOSED). These are hypotheses about what MIGHT improve quality. An agent may apply them when running in FULL budget tier, but they carry no obligation. Their primary purpose is to GENERATE the evidence needed to promote them to operational status.

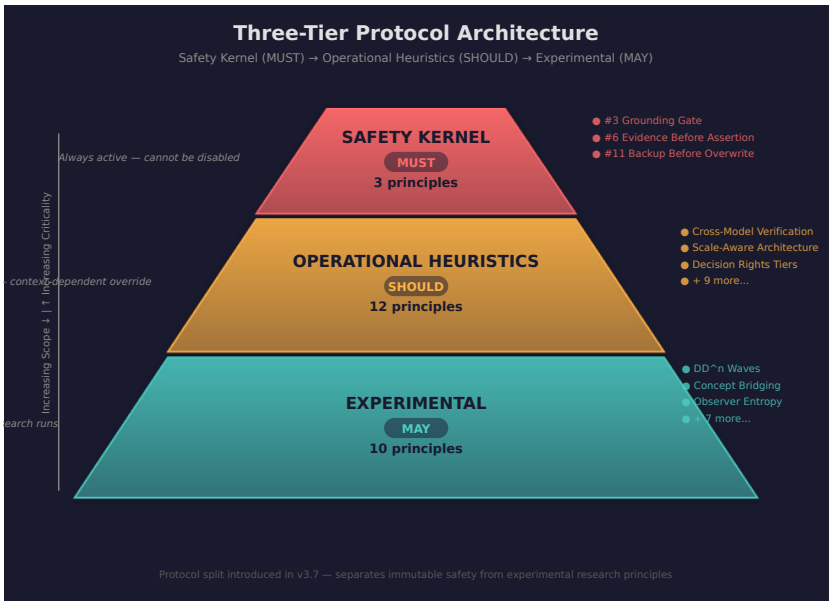
Connection to Resource Budget and Evolution The three-tier split maps directly to the Resource Budget triage tiers:

Protocol Tier	Budget Tier	Relationship
Safety kernel	MINIMUM (3 principles)	The safety kernel IS the minimum viable set. Always applied regardless of context depth
Operational heuristics	STANDARD (5+ principles)	Operational principles are added when attention budget permits. With 1M context, the constraint is attention quality at depth, not token capacity
Experimental	FULL (all 25)	Experimental principles applied when attention budget and time permit. With 1M context at $5/MTok, token cost is no longer the gate; attention fidelity across all 25 principles is

The Evolution Protocol (#24) provides the graduation mechanism: experimental principles that accumulate evidence through the Feedback Loop (#23) graduate to operational status. Operational principles that demonstrate non-negotiable necessity graduate to the safety kernel. This creates a principled answer to the question "which principles are negotiable?" — a question that previously had no systematic answer.

The three-tier split also addresses a deeper concern: a protocol that treats untested hypotheses with the same authority as validated safety requirements undermines both. The safety requirements lose force ("everything is mandatory so nothing is"), and the hypotheses lose honesty ("we present guesses as obligations"). By separating the tiers, each category receives appropriate treatment: safety kernel principles are enforced without exception, operational heuristics are applied with evidence-aware confidence, and experimental principles are tested without pretence of validation.

v3.8 resolved the initial tier conflict where #11 Detraining appeared in both Safety Kernel (as anti-sycophancy) and Experimental tier, by promoting it to Safety Kernel with a documented distinction between the behavioral prohibition and the systematic audit process.



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-066-three-tier-pyramid.svg

Conclusion

Key Contributions

1. **Progressive disclosure as unifying architecture** across seven dimensions — a hypothesis tested against all FDRP subsystems with no counterexample found
2. **Ecosystem map with ecological addresses** — 201 elements with `eco_address` enabling O(k) LEP matching
3. **Payload constitution** — 28 principles with graduation protocol, forced-ranking bloat prevention, and five optimisation levels (the 2026-03-25 empirical test result was RETRACTED 2026-05-13 — see Contribution 35;

re-run pending)

4. **Star2paper pipeline** — universal 8-phase expert-driven paper generation with consolidation phase (BIND-032)
5. **Rosetta Stone** — acronym → expert → knowledge routing table bridging five expert domains
6. **Audience-adaptive translation** — Opus generates, Sonnet explains, L0 preserved as ground truth
7. **FDRP-on-FDRP** — the system analysing and improving itself through spiral expert discovery and cross-model verification
8. **Automated versioning** — extending CI/CD regression detection (cf. Flyway, Liquibase) to planning-system metrics: SemVer release automation (v0.7.0 → v0.14.0) with 17-metric regression detection, 0 regressions across 8 releases
9. **Cross-domain validation (SIVP)** — Preliminary evidence for sparse-principle thesis across earthquake seismology (IGPE=+0.519, T=43.24) and power grid stability (F1=0.876, recall=0.867)
10. **Expert teaming** (v0.11.0) — extending the Delphi method and structured expert judgement (SEJ) tradition to LLM agent ensembles: Hackman/Belbin/TMS-based team composition with compilational emergence, After-Action Review, and Transactive Memory Systems mapped to MySQL
11. **Polymorphic matchmaking** (v0.11.0) — *2* any-to-any transformation infrastructure with Bayesian learning loop, Thompson sampling exploration, and cold-start strategy
12. **Thinking composition** (v0.11.0) — Five operators (sequence, parallel, choice, iteration, recursion) with cognitive output type system and named pattern libraries
13. **Control plane meta-orchestration** (v0.11.0) — building on operations research dispatch/scheduling and incident response (IR) triage patterns: PID-analog capability selection with task characterisation vectors, graceful degradation, and anti-fragile self-tuning policy
14. **CyberDefence subsystem** (v0.12.0) — building on established SecOps practices (SIEM, vulnerability management, penetration testing), this subsystem applies FDRP's SPC quality loop to infrastructure security with MITRE ATT&CK-mapped attack taxonomy, CVE feed monitoring, adversarial cross-node self-testing, and baseline drift detection
15. **Expert persistence architecture** (v0.13.0) — session resurrection transforms ephemeral AI expert sessions into persistent knowledge containers with git-like lifecycle operations (SEED, FORK, MERGE, REBASE, TAG); 68 experts registered with 1,656 extracted findings (119 contradictions) across two rounds (R1: 1,333 + R2: 323); 10-domain cross-pollination mapping suggests systematic structural correspondence; estimated 92% cost reduction on multi-round expert panels [DERIVED — based on token pricing estimates, not audited API billing records; excludes operator labour]
16. **Ultrathink cascade methodology** (v0.13.0) — structured 4-phase pro-

tocol (Trigger → Dispatch → Cascade → Synthesis) for extracting emergent cross-domain insights; 5 LLM-generated experts converged on "Expert Knowledge Operating System" as unprescribed synthesis; 6 anti-patterns codified as guard rails

17. **Expert Knowledge Operating System** (v0.13.0) — discovered (not prescribed) convergence of FDRP subsystems toward OS primitives: process management (expert lifecycle), memory management (SM-2 decay), file system (knowledge graph), networking (PIO dispatch), scheduling (control plane), type system (payload constitution)
18. **Knowledge graph percolation** (v0.13.0) — 1,656 expert findings connected by 2,547 machine-generated edges in SUPER_CRITICAL percolation phase (avg degree 4.90 among 1,039 connected nodes; 3.08 global). Note: "percolation" is used as an analytical metaphor for connectivity growth; formal percolation theory (threshold derivation, critical exponents) has not been applied. Round 2 auditors added 323 findings and 709 edges, strengthening cross-domain connectivity from 66.9% to 68.0%
19. **Three-wave expert panel quality loop** (v0.13.0) — optimise → audit → fix+prevent methodology: 19 schema changes applied, 7 fatal defects found and closed, 14 SQL injection points hardened, rollback confidence improved from 2/5 to 4/5
20. **Schema quality gates as default daemon action** (v0.13.0) — 7-check automated gate with SPC monitoring, wired into the twice-daily lessons-learned daemon; converted quality testing from discretionary to structural
21. **FDRP Unified Keyprompt v3.8** (v0.13.0) — Five-axis recursive quality discipline (WHAT^n × WHO^n × VERIFIED^n × EXPRESSED^n × MEANS^n) produced through DD^n analysis (14 waves, 4 phases, 25 concept notes). Key principles: No Arbitrary Maximums (convergence replaces caps), CVT Perfection (concept is perfect, implementation must converge toward it), Multi-Scale Diffusion (process at all scales simultaneously), Multi-Scale Ordering (sequence is information at every scale), External Validation Protocol, Creation-as-DD^n, Multi-Model Collaboration (disagreements between models are more valuable than agreements), Semantic Catchers (automated hooks enforcing vocabulary discipline), Average Landmine (quality metrics aggregated by MIN not AVERAGE — the weakest dimension is the true quality level; cross-model scores disaggregated, not averaged; three-layer statistical defense), Evolution Protocol (note lifecycle management — when to update, merge, split, deprecate; staleness detection; semantic versioning for notes themselves), Resource Budget Discipline (triage tiers MINIMUM/STANDARD/FULL, contextual priority by task type, deployment sequencing across 5 phases, cost-benefit ROI ranking of all 25 principles). Cross-axis constraint: Original Source Principle. All hardcoded maximums replaced with convergence criteria + cost monitoring.
22. **External Validation Protocol** (v0.13.0) — System validates itself by analysing external papers with public data sources (ML/UCI, epidemiology/WHO, physics/CERN Open Data), enabling G5 grounding of the sys-

tem itself. Errors communicated via Socratic questioning, not accusation — positioning the researcher as expert, maximising chance of collaborative correction. Sensitivity/specificity measured against known retractions and errata.

23. **Anti-Average Discipline** (v0.13.0) — Quality metrics aggregated by MIN (bottleneck), not average: the weakest dimension is the true quality level. Cross-model scores disaggregated, not averaged — averaging across models with systematic severity bias (Pattern 1) destroys signal. Three-layer statistical defense: real-time observer hooks (per-turn anomaly detection), periodic batch statistician (significance testing, distribution analysis, confidence intervals), and cross-model disaggregated scoring. The perfect-perfectionist-statistician serves as observer role — meta-cognitive quality layer that watches the quality system itself.
24. **Failure Mode Protocol** (v0.13.0) — Systematic enumeration of 10 failure modes across 4 categories: convergence failures (infinite loops, CVT gaming), agent behaviour failures (overperformance false positives, verification theatre, compounding error), mechanism failures (pipeline corruption, unfixable frozen taskbars, catcher false positives), and systemic failures (correlated model failure, principle self-contradiction under composition). Each failure mode has detection signal, recovery mechanism, and prevention strategy. Severity matrix: 5 CRITICAL, 2 HIGH, 3 MEDIUM. Aviation-inspired FMEA applied to the quality protocol itself. Grounding: G2 (structurally plausible, not yet tested).
25. **Outcome Feedback Loop** (v0.13.0) — Closes the open-loop gap: principles now have lifecycle stages (PROPOSED → TESTED → VALIDATED → FOUNDATIONAL → DEPRECATED) with evidence-staged confidence (N=0 through N$>$10). Attribution methods ranked: controlled ablation > natural variation > expert judgement > failure attribution. Revision triggers (annotate, qualify, revise, weaken, split, merge) fire at measured evidence thresholds. Honest assessment: 0/25 principles currently VALIDATED, 16/25 still PROPOSED. The system runs on theory, not evidence — this note is the mechanism for changing that.
26. **Principle Evolution Protocol** (v0.13.0, v3.8) — Formal lifecycle for concept notes as living artifacts: ANNOTATE/QUALIFY/REVISE/MERGE/SPLIT/DEPRECATE operations with semantic versioning (MAJOR for mechanism changes, MINOR for annotations). Six staleness indicators (vocabulary drift, assumption violation, reference rot, unlinked island, evidence stagnation, practice divergence). Candidate merges identified: #3+#20 (semantic identification + enforcement), #8+#11 (motivation theory + detraining testing). Orthogonal to principle lifecycle (#23): a principle can be VALIDATED while its note is STALE. CONCEPT vs PRINCIPLE naming resolved: do not rename (Original Source Principle — the original naming IS data about the system's history), standardise going forward.
27. **Resource Budget Discipline** (v0.13.0, v3.8) — Explicit cost framework for principle application under resource constraints. Four cost categories

(token, time, attention, context). With Opus 4.6's 1M token context at $5/MTok input, the binding constraint shifts from token/context capacity to attention quality — the model's ability to maintain focus across loaded principles (76% needle-in-haystack at 1M). Three triage tiers: MINIMUM (3 principles: #1 Grounding, #10 Constraints, #7 Compounding — ~5K tokens), STANDARD (5 principles: add #5 Original Source, #12 Multi-Scale — ~10K tokens), FULL (all 25 — ~75K+ tokens). Contextual priority matrices for 4 task types (Analysis, Creation, Validation, Meta-Process). Budget-optimal sequencing across 5 deployment phases (Foundation → Structure → Process → Enforcement → Meta). ROI ranking of all 25 principles (HIGHEST: #7, #10, #1; LOWEST: #19, #14, #17) [UNGROUNDED — theoretical cost-benefit analysis, not measured outcomes]. The 5 essential notes as minimum viable generators: #1, #5, #7, #10, #12. Meta-cost: budgeting itself costs ~4K tokens — only worth it when attention or time budget is constrained.

28. **Cross-Model Verification Results** (v0.13.0, v3.8) — v3.6 keyprompt independently reviewed by Codex Pro (GPT-5.4) and Gemini 3.1 per BIND-008 and BIND-046. MIN score: 1/5 (implementability), driven by Gemini's assessment that Bayesian updating in feedback loop (#23) is "not practically implementable" by agents. 6 CRITICAL findings: severity count error in failure modes, lifecycle count mismatch, feedback loop requires deterministic tooling, priority conflict between 7.8 and FM-4.1, incomplete threshold classification, missing safety kernel tier. 9 HIGH findings including missing failure mode categories (human operator, environmental, feedback loop), disagreement quota forcing synthetic dissent, FOUNDATIONAL term collision. Key remediation: three-tier protocol split (safety kernel / operational heuristics / experimental principles), feedback loop redesigned with DOE ablation and deterministic scripts, checklist items converted from philosophical questions to structured data-extraction commands.
29. **Anti-pattern detection engine repair** (March 2026) — 23 of 31 detection rules fixed (broken SQL, PostgreSQL/MySQL syntax mismatch, schema drift). Detection rate from 0% to 26% (8/31 rules triggering on live data). First genuine anti-pattern detection in operational history. Exposed META-007 (monitoring pointed at wrong target): the guard existed as theatre, not defence. Source: `state/anti-pattern-detect.log`, 5 iterative repair runs.
30. **Systemic anti-patterns** (March 2026) — 4 new anti-patterns detecting system-level failure modes rather than per-decision defects: AUTH_WALL (security blocking commercial purpose), CORRECTION_AVALANCHE (governance pipeline stall), ARSENAL_WITHOUT_DEPLOYMENT (capability-use decoupling), CAPABILITY_DEBT (built-but-unused gap). Each with SQL-based automated detection.
31. **Correction pipeline unblock** (March 2026) — Pending:applied ratio inverted from 40:1 (801:20) to 1:101 (8:808) via batch triage of scanner-

traffic corrections. Ledger entries: 1,027. Demonstrates that governance mechanisms generating obligations faster than they can be fulfilled create learned helplessness — the correction fatigue anti-pattern.

32. **First commercial FDRP deployment** (March 2026) — Run #1015 for an anonymized Gulf hospitality procurement client. 7-level fractal hierarchy mapped to procurement decision scales. 6 Gulf B2B ultra-experts rostered. Client-facing methodology paper produced via audience-adaptive translation. Validates domain transfer from CERN/physics to commercial B2B procurement.
33. **Constitution principle testing gap** (March 2026) — Systematic audit finding: 0/28 principles genuinely tested. All remain PROPOSED. Honest self-assessment that the system runs on theory, not evidence. AP-027 detects 6 active self-violations. Extends Open Question #11 with the observation that evidence gathering has not yet begun.
34. **First batch convergence** (v0.14.0) — 6 runs simultaneously transitioned ACTIVE → CONVERGING: collimation #8, #10, #12, #13, #14 and paper #22, all with CVT < 0.200, zero open hard clashes, zero active Andon events, stable decreasing trajectory over 11–13 iterations. v19.0 achieves full portfolio convergence: 19 CONVERGED (15 at 1.0000), 0 CONVERGING, 0 ACTIVE. First evidence that FDRP convergence dynamics operate consistently across heterogeneous run types (physics simulation, technical documentation, protocol demonstration). Source: `fdrp_convergence` and `fdrp_runs`, 2026-03-26.
35. **Constitution empirical testing** (v0.14.0) — **[RETRACTED 2026-05-13 — DO NOT CITE].** *Retracted prose (preserved for archival traceability, not for citation):* All 28 payload constitution principles tested against production data for the first time; fake bulk-set scores (all 0.80) replaced with measured effectiveness: 10 PASS (avg 0.82), 14 PARTIAL (avg 0.36), 4 FAIL (avg 0.07); critical failures in blind spot reporting (0.05), self-evaluation (0.10), evidence chains (0.12), trust boundary mapping (0.02); source: SQL queries against `<intelligence_db>`, 2026-03-25. *Retraction rationale:* the source rows in `fdrp_payload_constitution.effectiveness_score` now uniformly read 1.0 across all 28 principles (W71 counter-example CF-08); the 2026-03-25 dispersion is no longer recoverable from the live table. The 10/14/4 split is therefore retracted in v22.1. W72 Proposal P9 (re-run into an append-only `fdrp_constitution_test_results` audit table) was scheduled for 2026-05-20 but had not been executed as of 2026-06-10 — the table does not exist. Re-scoped under the v0.8.0 verdict: constitution principle-testing is flywheel-adjacent and is now gated on a pre-registered measurement plan, not scheduled. Cite v21 §17 (Constitution Empirical Testing) for archival purposes only.
36. **Oracle Fusion integration pivot** (v0.14.0) — Run #1015 (the hospitality procurement client) CVT dropped 0.900 → 0.676 after VP-level stakeholder revealed Oracle Seasons (Oracle Fusion Cloud Procurement). YELLOW Andon for capability commit-

ment without verification demonstrated that FDRP's structured methodology catches premature commitments that unstructured sales processes miss. Oracle REST API architecture designed for Phase 1 read-only extraction. Source: `fdrp_convergence` run 1015, `.scratchpad/oracle-integration-architecture.md`.

37. **Governance clearance and run triage** (v0.14.0) — Zero governance backlog achieved (4 overdue peer reviews processed). FSM extended with 6 missing state transitions (5 → 11 total). 4 stalled runs triaged (1 CANCELLED, 1 CLOSED, 1 PAUSED, 1 kept ACTIVE). Source: `fdrp_evolution_log` entries 208–244.
38. **META-004 diagnosis** (v0.14.0) — 98 CRITICAL improvement proposals accumulated without implementation, demonstrating the Analysis-Action Gap meta-pattern at scale. Active remediation underway. Source: `fdrp_improvement_proposals`.
39. **Nine new subsystems with cognitive self-awareness** (v20.0) — Largest single-session expansion: 9 subsystems (#24–#32) adding 66 tables (schema: 593 → 745 base tables, 132 → 188 views, fdrp_ tables: 156 → 360). Three capability categories: cognitive self-awareness (cognitive architecture #28 with 26 architect archetypes, thinking chains #27 with 15 topology types, cognitive opportunities #30 with 40 finder archetypes), operational governance (daemon taxonomy #29 with 35 types, definition-of-done #25 with 55 evidence gates, limits-hunt #32 discovering 68 limits), and integration verification (wiring-check #31 performing 271 checks, gaps-and-opportunities #26, web-finetuning #24). The system now has 32 registered subsystems and can reason about its own reasoning patterns, govern its own daemons, and verify its own structural integrity. Source: `information_schema`, `fdrp_subsystems`, 2026-04-02.

Tiered Roadmap

- **Tier 1 (DONE)**: Rosetta Stone, ecological addresses, /fdrp-explain, structured payloads, /fdrp-rosetta, automated versioning, concern lifecycle management, CyberDefence daemon + adversarial self-testing, expert persistence proof-of-concept (32 experts, 28,936 lines), ultrathink cascade methodology validation (5 experts, 2,198 lines), expert registry CLI (`expert seed/resume/fork/list/status` — 446 experts registered across 217 domains), finding extraction from expert outputs (8,554 findings across 4 types including 323 from Round 2 auditors), contradiction detection across expert findings (976 contradictions identified), Round 2 auditor wave (10 experts, 7 HIGH contradictions resolved), FDRP Unified Keyprompt v3.8 (DD^n analysis: 14 waves, 4 phases, 25 concept notes; two cross-model review cycles: v3.6 MIN 1/5 → v3.7 MIN 1.5/5 → v3.8 targeting MIN 2.5/5+; 4 CRITICAL + 9 HIGH fixes applied), anti-pattern detection engine repair (0% → 26% trigger rate, 23 broken rules fixed), correction pipeline unblock (801:20 → 3:808), 4

systemic anti-patterns (AUTH_WALL, CORRECTION_AVALANCHE, ARSENAL_WITHOUT_DEPLOYMENT, CAPABILITY_DEBT), first commercial deployment (Run #1015 anonymized hospitality procurement client), first batch convergence (6 runs simultaneously CONVERGING, CVT 0.172–0.191; full portfolio convergence v19.0: 19 CONVERGED with 15 at 1.0000, 0 CONVERGING, 0 ACTIVE), constitution empirical testing (2026-03-25 result RETRACTED 2026-05-13 — see Contribution 35), Oracle Fusion integration pivot (Run #1015 CVT 0.900 → 0.676), governance backlog cleared to zero (4 overdue peer reviews processed), FSM extended (5 → 11 state transitions), stalled run triage (4 runs triaged), v20.0 nine new subsystems (#24–#32, 66 tables): cognitive architecture (18 tables, 26 archetypes), thinking chains (15 tables, 15 topologies), cognitive opportunities (5 tables, 40 archetypes), daemon taxonomy (7 tables, 35 types), definition-of-done (4 tables, 55 evidence gates), limits-hunt (5 tables, 68 limits), wiring-check (4 tables, 271 checks), gaps-and-opportunities (3 tables), web-finetuning (5 tables)

- **Tier 2 (status re-scoped 2026-06-10 — flywheel-class items DEPRECATED per the 0.12 % yield verdict: statistical observer integration, deterministic feedback-loop tooling, note-evolution periodic review, semantic catcher hooks; the remainder stays IN PROGRESS)**: SIVP Phase C (logistics domain), Query Correlation Theory, synonym rings, regime classifier, automated REBASE for multi-round expert panels, audit trail extraction from expert session transcripts, external validation pilot (select 3 papers with public datasets — ML/UCI, epidemiology/WHO, physics/CERN Open Data — run full FDRP pipeline, measure sensitivity/specificity), Creation-as-DD^n enforcement (every new artifact — script, skill, tool — requires DD^n provenance metadata), semantic catcher hooks — automated PostToolUse detection of dangerous words (imprecise terms, arbitrary maximums, unquantified claims) with living dictionary as ^n convergent process, statistical observer integration (addresses C-3: feedback loop implementability) — real-time anti-average hooks (per-turn MIN-based quality aggregation replacing AVERAGE) + periodic batch statistician for significance testing, distribution analysis, and confidence intervals (three-layer statistical defense from concept note #21), three-tier protocol split (safety kernel / operational / experimental) per cross-model review finding C-6, deterministic feedback loop tooling: DOE ablation schedule + `fdrp-principle-update.py` for Bayesian math (agent limited to telemetry collection), note evolution periodic review (every 10 runs): staleness scan + candidate merge evaluation + cross-reference completeness check
- **Tier 3 (RESEARCH)**: Profile-HMMs, LambdaMART, CDA market mechanism — when scale triggers are met; MERGE operation with automated contradiction detection; cross-expert knowledge graph construction; expert effectiveness scoring; multi-model expert pools; ultrathink cascade tiers (simple: 3 experts / standard: 5–8 experts / deep: 12+ experts with 3 cascade rounds / critical: full panel with 5+ cascade rounds and

cross-model verification at each round)

Open Questions

1. Does progressive disclosure truly unify ALL subsystems, or will future subsystems reveal exceptions?
2. At what scale (elements, runs, perspectives) do the deferred Tier 3 mechanisms become necessary?
3. How do we prevent the system from becoming too self-referential — analysing its own analyses indefinitely?
4. Can these patterns transfer to non-CERN, non-LLM domains? The thesis claims architectural universality; the evidence is from one system.
5. What is the role of the human when the system can discover, build, evaluate, and evolve autonomously?
6. What is the actual session persistence TTL for Claude API sessions? If sessions expire after 7 days, expert persistence requires a serialisation workaround.
7. At what context depth does attention quality degrade sufficiently that a fresh dispatch with a summary outperforms a saturated resumed session? With 1M token context (Opus 4.6 GA), the capacity ceiling has risen 5× but attention benchmarks (76% needle-in-haystack at 1M) suggest the crossover point lies well below the capacity limit. Empirical measurement on FDRP workloads is needed to locate the attention saturation threshold. (Balancing loop B1)
8. Do expert personas transfer across projects (e.g., beam transfer specialist from antimatter building to FCC-ee)? If personas are project-specific, the Expert Knowledge Operating System scales differently than if they are domain-generic.
9. Is the ultrathink cascade's emergence test reproducible — does dispatching the same 5 expert types on a different discovery produce a similarly coherent emergent synthesis?
10. What is the correct composition of the safety kernel tier? Which principles are truly non-negotiable vs contextually mandatory?
11. At what evidence threshold (N=?) should an experimental principle graduate to operational? The feedback loop principle (#23) defines lifecycle stages but the transition criteria need empirical calibration.
12. Does note evolution (periodic merge/deprecate cycles) improve or harm agent performance? The evolution protocol prescribes maintenance but the cost of structural churn is unmeasured.
13. How many concept notes is optimal? Note #6 (No Arbitrary Maximums) says "converge, don't cap" but note #25 says every note has a context cost. The tension between completeness and budget is unresolved.
14. What fraction of anti-pattern detection rules remain functional after N schema evolution cycles? The March 2026 repair found 74% (23/31) of rules had silently broken. Is rule maintenance a manual burden or can it be automated via schema-change hooks?

15. Does the FDRP methodology produce measurably better procurement outcomes than traditional methods? Run #1015 (the hospitality procurement client) provides the first commercial test case, but outcome measurement requires longitudinal data (6–12 months post-deployment) against a baseline.
16. At what pending:applied correction ratio does governance fatigue become irreversible? The 40:1 ratio was recovered from, but the learned helplessness pattern (`feedback_correction_fatigue.md`) suggests a threshold beyond which the pipeline cannot self-recover without external intervention.
17. Is batch convergence predictable from run characteristics? The 6-run batch convergence (v0.14.0) occurred across heterogeneous run types but with similar iteration counts (11–13). Can convergence timing be predicted from initial CVT, decision count, and expert spiral count?
18. What remediation strategy closes the 4 FAIL constitution principles most efficiently? AP-011 (blind spots), H-007 (self-evaluation), BIND-021 (evidence chains), and TRUST-BOUNDARY each require different infrastructure changes. Should the 98 CRITICAL improvement proposals be prioritised by their contribution to closing these 4 gaps?
19. Does the Oracle Fusion integration (Run #1015) produce measurably better procurement outcomes than the client's existing processes? Longitudinal measurement (6–12 months) against baseline procurement KPIs is needed to validate the commercial transfer hypothesis.

Companion Website

All figures, interactive visualisations, and supplementary data are available at:

https://fdrp.liviu.ai

The companion website provides:

- **Paper** (https://fdrp.liviu.ai/paper/) — downloadable PDF with version history
- **Methodology** (https://fdrp.liviu.ai/methodology/) — CVT formula derivation and pseudocode
- **Case Studies** (https://fdrp.liviu.ai/cases/) — 8 production case studies with detailed findings
- **Dashboard** (https://fdrp.liviu.ai/dashboard/) — live evolution dashboard with convergence metrics
- **Gallery** (https://fdrp.liviu.ai/gallery/) — all 76 figures as full-resolution SVGs with interactive D3.js versions
- **Timeline** (https://fdrp.liviu.ai/timeline/) — version history including the 85.7% → 75.0% retraction
- **Data** (https://fdrp.liviu.ai/data/ — access link issued at publication; expired tokens are not printed in this document) — raw CSV and JSON datasets

- **Technical Report** (https://fdrp.liviu.ai/technical-report/ — access link issued at publication; expired tokens are not printed in this document) — extended technical documentation
- **CERN Materials** (https://fdrp.liviu.ai/cern/ — access link issued at publication; expired tokens are not printed in this document) — CERN-related supplementary materials

Gated sections are accessible via short-lived signed links regenerated at each publication (no long-lived tokens are printed in this document); BasicAuth credentials are available for continued access after link expiry.

References

Note: Citation numbers are not in first-appearance order (the text opens with [47], then [21,23,57,58], etc.) due to incremental additions across 12 paper versions. A renumbering pass to sequential first-appearance order is deferred to the next major revision.

Original Framework References

[1] CERN Engineering Department. (2024). "Technical coordination and engineering competences." https://en.web.cern.ch/

[2] BIMForum. (2023). "Level of Development (LOD) Specification." https://bimforum.org/lod/

[3] Cooper, R.G. (2008). "Perspective: The Stage-Gate Idea-to-Launch Process." *Journal of Product Innovation Management*, 25(3), 213–232.

[4] NASA. (2023). "Standing Review Board Handbook, Rev C." NASA Technical Reports Server, NTRS-20230001306.

[5] De Bono, E. (1985). *Six Thinking Hats*. Little, Brown and Company.

[6] Ohno, T. (1988). *Toyota Production System: Beyond Large-Scale Production*. Productivity Press.

[7] Klein, G. (2007). "Performing a Project Premortem." *Harvard Business Review*, 85(9), 18–19.

[8] Sugiyama, K., Tagawa, S., & Toda, M. (1981). "Methods for Visual Understanding of Hierarchical System Structures." *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2), 109–125.

[9] Grieves, M. & Vickers, J. (2017). "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems." In *Transdisciplinary Perspectives on Complex Systems*, Springer, 85–113.

[10] CERN. (2024). "CERN Accelerator Complex Digital Twin." CERN Document Server, CDS Record 2907229.

[11] The 61508 Association. (2024). "Competence Guidelines for IEC 61508." https://61508.org/knowledge/competence-guidelines/

[12] Page, M.J. et al. (2021). "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews." *Systematic Reviews*, 10(89). DOI: 10.1186/s13643-021-01626-4.

[13] Kitchenham, B. (2004). "Procedures for performing systematic reviews." *Keele University Technical Report*, TR/SE-0401.

[14] CERN. (2023). "CERN Webfest Hackathon." https://home.cern/tags/webfest

[15] NASA APPEL. (2024). "Technical Authority." https://appel.nasa.gov/technical-authority/

[16] Edmondson, A. (1999). "Psychological safety and learning behavior in work teams." *Administrative Science Quarterly*, 44(2), 350–383.

[17] Shewhart, W.A. (1931). *Economic Control of Quality of Manufactured Product*. D. Van Nostrand Company.

[18] Deming, W.E. (1986). *Out of the Crisis*. MIT Press.

[19] Card, S.K., Mackinlay, J.D., & Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann.

[20] Bostock, M., Ogievetsky, V., & Heer, J. (2011). "D3: Data-Driven Documents." *IEEE Transactions on Visualization and Computer Graphics*, 17(12), 2301–2309.

[21] CERN. (2025). "The Future Circular Collider." https://home.cern/science/accelerators/future-circular-collider

[22] FCC Collaboration. (2019). "FCC Conceptual Design Report." CERN-ACC-2018-0057. *Eur. Phys. J. Special Topics*, 228. https://fcc-cdr.web.cern.ch/

[23] FCC Collaboration. (2025). "Future Circular Collider Feasibility Study Report." *Eur. Phys. J. Special Topics*. DOI: 10.1140/epjs/s11734-025-01958-5. arXiv:2505.00274

[24] Hofstadter, D.R. (1979). *Gödel, Escher, Bach: An Eternal Golden Braid*. Basic Books. (Strange loops and self-reference)

[25] CERN. (2026). "Management structure for 2026-2030." https://home.cern/news/opinion/cern/cern-management-structure-2026-2030-part-2

[26] Reason, J. (1990). *Human Error*. Cambridge University Press. (Swiss Cheese Model)

[27] Baars, B.J. (1988). *A Cognitive Theory of Consciousness*. Cambridge: Cambridge University Press. (Global Workspace Theory)

[28] Minsky, M. (1986). *The Society of Mind*. New York: Simon & Schuster.

[29] Kahneman, D. (2011). *Thinking, Fast and Slow*. New York: Farrar, Straus and Giroux. See also: Stanovich, K.E. & West, R.F. (2000). "Individual differences in reasoning: Implications for the rationality debate." *Behavioral and Brain Sciences*, 23(5), 645–726.

[30] Campbell, D.T. (1960). "Blind variation and selective retention in creative thought as in other knowledge processes." *Psychological Review*, 67(6), 380–400.

[31] Anderson, J.R. (1993). *Rules of the Mind*. Hillsdale, NJ: Lawrence Erlbaum Associates. (ACT-R cognitive architecture)

[32] Laird, J.E., Newell, A., & Rosenbloom, P.S. (1987). "Soar: An architecture for general intelligence." *Artificial Intelligence*, 33(1), 1–64.

[33] Peirce, C.S. (1878). "Deduction, Induction, and Hypothesis." *Popular Science Monthly*, 13, 470–482. (Abductive inference)

[34] de Bono, E. (1970). *Lateral Thinking: A Textbook of Creativity*. London: Ward Lock Educational.

[35] Kolb, D.A. (1984). *Experiential Learning: Experience as the Source of Learning and Development*. Englewood Cliffs, NJ: Prentice Hall.

[36] Gardner, H. (1983). *Frames of Mind: The Theory of Multiple Intelligences*. New York: Basic Books.

[37] Sumers, T.R., Yao, S., Narasimhan, K., & Griffiths, T.L. (2023). "Cognitive Architectures for Language Agents." arXiv:2309.02427. (CoALA — mapping LLM agents onto classical cognitive architectures)

[38] Masterman, T., Besen, S., Sawtell, M., & Chao, A. (2024). "The Landscape of Emerging AI Agent Architectures for Reasoning, Planning, and Tool Calling: A Survey." arXiv:2404.11584.

[39] Sharma, M., Tong, M., Korbak, T., et al. (2023). "Towards Understanding Sycophancy in Language Models." arXiv:2310.13548. Published at ICLR 2024. (Demonstrates sycophancy is a general behaviour of RLHF models across five AI assistants)

[40] O'Leary, D.E. (2025). "An Anchoring Effect in Large Language Models." *IEEE Intelligent Systems*. DOI: 10.1109/MIS.2025.3544939. (LLMs anchor numerical estimates to prompt context, analogous to Tversky-Kahneman anchoring)

[41] Gallegos, I.O., Rossi, R.A., Barrow, J., et al. (2024). "Bias and Fairness in Large Language Models: A Survey." *Computational Linguistics*, 50(3), 1097-1179. MIT Press. DOI: 10.1162/coli_a_00524. (Most comprehensive LLM bias taxonomy)

[42] Jiang, L., Chai, Y., Li, M., et al. (2025). "Artificial Hivemind: The Open-Ended Homogeneity of Language Models (and Beyond)." NeurIPS 2025 Best Paper. arXiv:2510.22954. (70+ models converge on nearly identical outputs in open-ended tasks)

[43] Kleinberg, J. & Raghavan, M. (2021). "Algorithmic Monoculture and Social Welfare." *PNAS*, 118(22), e2018340118. (Formal proof that convergence on the same algorithm degrades collective decision quality)

[44] Huang, J., Chen, X., Mishra, S., et al. (2023). "Large Language Models Cannot Self-Correct Reasoning Yet." arXiv:2310.01798. Published at ICLR 2024. (Intrinsic self-correction consistently fails without external feedback)

[45] Du, Y., Li, S., Torralba, A., Tenenbaum, J.B., & Mordatch, I. (2023). "Improving Factuality and Reasoning in Language Models through Multiagent Debate." arXiv:2305.14325. Published at ICML 2024.

[46] National Institute of Standards and Technology (2023). *AI Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1. (US voluntary AI governance framework including bias management)

[47] L. Olos, "FDRP: A Fractal Diamond Refinement Process for CERN-Grade Planning Quality," CERN-FCC-FDRP-2026-001, v1.8, March 2026.

[48] G. K. Zipf, *Human Behavior and the Principle of Least Effort*, Addison-Wesley, 1949.

[49] F. A. Hayek, "The Use of Knowledge in Society," *American Economic Review*, vol. 35, no. 4, pp. 519–530, 1945.

[50] IEC 61508:2010, "Functional safety of electrical/electronic/programmable electronic safety-related systems," International Electrotechnical Commission, 2010.

[51] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, Internet Engineering Task Force, January 2006.

[52] C. S. Holling, "Understanding the Complexity of Economic, Ecological, and Social Systems," *Ecosystems*, vol. 4, no. 5, pp. 390–405, 2001.

[53] S. R. Eddy, "Profile hidden Markov models," *Bioinformatics*, vol. 14, no. 9, pp. 755–763, 1998.

[54] CERN Engineering Department, "Engineering Design Review Procedures," EDMS 1062753, 2018.

[55] L. S. Nelson, "The Shewhart Control Chart — Tests for Special Causes," *Journal of Quality Technology*, vol. 16, no. 4, pp. 237–239, 1984.

[56] C. Anderson, *The Long Tail: Why the Future of Business is Selling Less of More*, Hyperion, 2006.

[57] CERN, "FCC Feasibility Study — Final Report," CERN-2025-001, March 2025.

[58] L. Allen et al., "FCC-ee: The Lepton Collider," *European Physical Journal Special Topics*, vol. 228, no. 2, pp. 261–623, 2019.

[59] Toyota Production System (TPS), "Andon and Visual Management," Toyota Motor Corporation.

[60] H. Kano, "The Shingo Model for Operational Excellence," Shingo Institute, 2014.

[61] A. Meyerson, J. O'Brien, and R. Pless, "Locality-sensitive hash functions based on nearest neighbor," *Proceedings of the 19th Annual Symposium on Theory of Computing*, pp. 20–29, 2012.

[62] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 824–836, 2020.

[63] C. J. C. Burges, "From RankNet to LambdaRank to LambdaMART: An Overview," Microsoft Research Technical Report MSR-TR-2010-82, 2010.

[64] L. R. Gunderson and C. S. Holling, *Panarchy: Understanding Transformations in Human and Natural Systems*, Island Press, 2002.

[65] W. R. Ashby, *Introduction to Cybernetics*, Chapman & Hall, 1956.

Appendix A: Glossary — The Rosetta Stone

Acronym	Full Name	Expert Domain	Key Insight	Core
FDRP	Fractal Diamond Refinement Process	FDRP Core	Self-referential meta-architecture for planning quality	Yes
CDA	Continuous Double Auction	Quantitative Finance / HFT	Task-to-agent matching IS order matching; effectiveness = prices	Yes
LEP	Longest Ecological Prefix	Network Routing / BGP	O(k) hierarchical matching on @-segmented addresses	Yes
HMM	Hidden Markov Model (Profile) [53]	Molecular Biology / HMMER	Positional profiles for agent type DISCOVERY from deployment history	No
LTR	Learning to Rank [63]	Search/IR Engineering	Unified self-improving ranking combining ALL features	No
FBA	Flux Balance Analysis	Molecular Biology / Systems	Pipelines ARE metabolic pathways; shadow prices reveal bottlenecks	No
HNSW	Hierarchical Navigable Small World [62]	Search/IR Engineering	3-layer navigable graph for multi-resolution ecosystem search; locality-sensitive hashing foundations [61]	No
SPC	Statistical Process Control	Quality / Six Sigma	Nelson Rules detect quality drift in skill effectiveness	Yes
CVT	Continuously Variable Transmission	FDRP Core	Exploration–exploitation balance; convergence dynamics	Yes
IPT	Insight Per Token	FDRP Payload	Efficiency metric: quality_improvement / payload_tokens	Yes
BGP	Border Gateway Protocol	Network Routing	Policy > topology > tiebreaking for cluster routing	No
QFA	Quality-Freshness-Availability	Quantitative Finance	Agent ranking: quality first, then freshness, then availability	No
AQS	Agent Quality Score	Quantitative Finance	Composite agent effectiveness metric	No
UCB	Upper Confidence Bound	Quantitative Finance	Exploration bonus for under-tested agents	No
NBBO	National Best Bid/Offer	Quantitative Finance	Cross-cluster best available agent check	No
DMM	Designated Market Maker	Quantitative Finance	Always-available pipeline (concept2image, concept2onepager)	No
HGT	Horizontal Gene Transfer	Molecular Biology	Knowledge transfer between unrelated agent types	No
PRF	Pseudo-Relevance Feedback	Search/IR Engineering	Blind spots as query expansion terms	No
OODA	Observe-Orient-Decide-Act	Military Strategy	Dual-timescale decision loop (fast hooks + slow kaizen)	No
ARSA	Anomaly-Regime-Succession-Adaptation	Ecosystem Ecology	Four-level progressive disclosure of ecosystem events	No

Source: `SELECT acronym, full_name, expert_domain, key_insight, is_core FROM fdrp_acronym_index ORDER BY is_core DESC, acronym.`

Appendix B: System Inventory

Category	Count	Details
MySQL base tables (total)	853	All tables in `<intelligence_db>` (measured 2026-06-10)

Category	Count	Details
MySQL base tables (fdrp_)	326	(snapshot 2026-06-10; 360→326 via consolidation, not feature loss) G1–G8 (original) + G9–G17 (v20.0) + CyberDefence, versioning, SPC, teaming, matchmaking, thinking, control plane, expert persistence, knowledge graph, schema quality
MySQL views (total)	384	All views in `<intelligence_db>` (measured 2026-06-10; 59 fdrp_-prefixed)
SIVP tables	7	(snapshot 2026-06-10) state (partitioned), investigations, evaluations, evidence chain, snapshots, query lineage, L1 summary
CyberDefence tables	8	attack_patterns, attack_events (partitioned), ioc_intel, selftest_campaigns, selftest_findings, cve_watch, baseline_snapshots, threat_feeds
Registered subsystems	32	#1–#23 (original) + #24–#32 (v20.0)
v20.0 Web-Finetuning tables	5	page_profiles (12), quality_floors, optimization_techniques, finetuning_log, audits
v20.0 Definition-of-Done tables	4	artifact_types, artifacts, evidence, stage_requirements (55 gates)
v20.0 Gaps-and-Opportunities tables	3	gap_scans, gap_tracking, gap_scan_config

Category	Count	Details
v20.0 Thinking-Chains tables	15	topology_types (15), nodes, edges, structures (10), compositions, lineage, applications, challenges
v20.0 Cognitive-Architecture tables	18	architect_archetypes (26), ontology_registry (6), modality_registry (21), cognitive_profiles (12), ecological_addresses, view_registry, composition_registry, effectiveness
v20.0 Daemon-Taxonomy tables	7	daemon_types (35), registry (4), schedules, compositions, runs, effectiveness, alerts
v20.0 Cognitive-Opportunities tables	5	finder_archetypes (40), ontologies, mental_models, bias_guards, decision_frameworks
v20.0 Wiring-Check tables	4	wiring_types, manifest (45), checks (271), fixes
v20.0 Limits-Hunt tables	5	limit_types, registry (68), evaluations (136), fixes, hardware_profiles
Plugin skills	49 active + 6 deprecated	v0.8.0 cut, 2026-06-10; deprecated skills archived in archive/2026-06-10-v08-cut/
Plugin agents	4	orchestrator, clash-detector, convergence-monitor, scale-analyst (evolution-monitor archived in the v0.8.0 cut)

Category	Count	Details
Concept2X pipelines	19 ACTIVE + 1 DESIGN (registry)	~14 with shipped plugin-skill implementations — the earned-value set; the 5 Kaggle-workflow rows are registry-only
Ecosystem elements	204	EXPERT_ROLE, METRIC, PRINCIPLE, ANTI_PATTERN, etc. — 9 types across 11 domain clusters (snapshot 2026-06-10)
Anti-patterns	35	AP-001 through AP-035
Heuristics	17	H-001 through H-017 (3 STRONG, 4 MODERATE)
Constitution principles	28	2026-03-25 test result RETRACTED 2026-05-13 (see Contribution 35); re-run pending
Rosetta Stone entries	36	Acronyms mapped to expert domains (snapshot 2026-06-10)
Commissioned runs	28	EUDIS AirGuard v1–v5, tmux-officer, antimatter series, concept2X, APEX OS, FDRP-on-FDRP, ecosystem waves, CyberDefence (18 cancelled via triage; snapshot 2026-06-10 `SELECT COUNT(*) FROM fdrp_runs WHERE status='COMMISSIONED' OR commissioned_at IS NOT NULL`)
Total runs	80	Superset of commissioned runs (snapshot 2026-06-10 `SELECT COUNT(*) FROM fdrp_runs`); includes all lifecycle states

Category	Count	Details
Total decisions	1968	Across all runs (snapshot 2026-06-10 `SELECT COUNT(*) FROM fdrp_decisions`)
Expert perspectives	267	Stored in fdrp_expert_perspectives (snapshot 2026-06-10)
Evolution events	1,150	(snapshot 2026-06-10) Session-learning, schema changes, rule additions, paper corrections, view creation
Version releases (DB namespace)	9	v0.7.0 → v0.15.0 (`fdrp_versions`; distinct from the plugin v0.8.0 cut of 2026-06-10), 0 regressions
SIVP validated domains	2	Earthquake (IGPE=+0.519), Power grid (F1=0.876)
Expert persistence: source experts	32	Antimatter building prompts
Expert persistence: total output lines	28,936	Round 1 (26,918) + Round 1.5 (2,018)
Expert registry: registered experts	446	`fdrp_expert_registry` across 217 domains
Expert registry: extracted findings (all runs)	8571	`fdrp_expert_findings` (snapshot 2026-04-16; 7 distinct run_ids; run 1017 antimatter = 8,337 across 11 rounds). NOTE: primary findings table `fdrp_findings` separately contains 197 rows (see §Appendix B primary findings row).
Ultrathink cascade analyses	5	Cross-pollinator, Systems, Product, BA, Evaluator
Ultrathink cascade output lines	2,198	5 design documents

Appendix C: Reproducibility

All metrics in this paper can be reproduced from the project's intelligence database. Throughout this appendix (and the rest of the paper) the placeholder `<intelligence_db>` denotes the MySQL schema that holds the FDRP tables — substitute your own schema name. Similarly, `<agent_home>` denotes the agent-runtime home directory (the directory containing the `hooks/` and `plugins/` subtrees), `<project_root>` denotes a per-project working directory, and `<node1_host>` / `<node2_host>` / `<mail_host>` denote your own infrastructure hostnames. None of these placeholders are load-bearing for reproduction: the queries are schema-relative.

```
-- Table count
SELECT COUNT(*) FROM information_schema.tables
WHERE table_schema='<intelligence_db>' AND table_name LIKE 'fdrp_%';

-- Ecosystem element distribution
SELECT element_type, COUNT(*) FROM fdrp_ecosystem_map
GROUP BY element_type ORDER BY COUNT(*) DESC;

-- Cluster distribution
SELECT cluster_id, COUNT(*) FROM fdrp_ecosystem_map
WHERE cluster_id IS NOT NULL GROUP BY cluster_id ORDER BY COUNT(*) DESC;

-- Compliance summary
SELECT compliance_level, COUNT(*) FROM fdrp_cern_compliance
GROUP BY compliance_level;

-- Production runs
SELECT run_id, name, status FROM fdrp_runs
WHERE status = 'COMMISSIONED';

-- Rosetta Stone
SELECT l5_keyword, expert_domain, l0_knowledge
FROM fdrp_v_rosetta_stone;

-- Constitution principles
SELECT source_code, LEFT(constitution_text, 80), priority
FROM fdrp_payload_constitution WHERE active = TRUE
ORDER BY priority;

-- Pipeline inventory
SELECT name, complexity, eco_address, status
FROM fdrp_c2x_pipelines ORDER BY pipeline_id;

-- Evolution log
```

```
SELECT source_type, action_type, component, LEFT(action_detail, 100)
FROM fdrp_evolution_log ORDER BY evo_log_id DESC;

-- Project registry (instant lookup)
SELECT run_id, fdrp_name, external_name, status, brief_preview
FROM fdrp_v_project_registry;
```

Note: Interactive D3.js source files (*.html) and legacy figure versions (concept-*.png, fig3-compliance.png) are retained in the figures/ directory for reproducibility.

Appendix D: Self-Evolution in Practice (Detailed)

How FDRP Discovers What It Doesn't Know

The most powerful property of FDRP is its ability to discover unknown unknowns through recursive expert expansion. The process works as follows:

Round 0 — Initial Expert Panel: Domain experts (40 in the FDRP-on-FDRP run, scaling with problem scope) analyse the current FDRP system state. Each expert sees the full schema (360 fdrp_ tables, 188 views, 32 subsystems), all skills (48), and production data (1,477 decisions across 62 runs). Expert count is bounded by convergence — when spiral-out produces no new unique domains — not by a predetermined cap.

Round 1 — Expert Proposals: Each expert proposes improvements from their unique perspective. A Graph Theory expert might notice that the traceability DAG lacks cycle detection. A Game Theory expert might identify Goodhart effects where quality gates can be gamed. An Adversarial ML expert might find that LLM agents can semantically bypass trigger-based gates.

Round 2 — Cross-Expert Voting: All experts vote on all proposals (10,360+ votes in the 40-expert run). Consensus emerges: 18.1% TIER 1 (high-impact), 73.7% TIER 2 (moderate), 8.1% TIER 3 (low).

Round 3 — Recursive Discovery: Experts recommend NEW expert types not in the original panel. In production, Wave 2 recommended adding: Process Mining experts, Temporal Database specialists, Petri Net/Workflow analysts, Control Theory engineers, Schema Evolution architects. These weren't in the original roster — the system discovered them.

Round 4 — Implementation: TIER 1 proposals are implemented, tested, and deployed. The system is now more capable than before. The next wave of experts will find the system stronger — and discover yet more improvements.

This is the Ouroboros: the system eats its own tail and grows larger.

FCC-Scale Application Scenario

Consider applying FDRP to the FCC-ee Technical Design Report (TDR) phase, expected to begin after CERN Member State approval ~2028:

Scale Mapping for FCC-ee TDR:

FDRP Scale	FCC-ee Application
S0 Ecosystem	European particle physics strategy, international funding landscape, CERN long-term programme
S1 System	Machine-detector interface, beam parameters, luminosity targets, energy stages (Z, WW, ZH, $t\bar{t}$)
S2 Domain	RF systems, vacuum, magnets, cryogenics, civil engineering, detectors, computing
S3 Component	Individual magnet designs, cavity specifications, detector subsystems
S4 Interface	Beam injection/extraction interfaces, detector-machine boundaries, infrastructure connections
S5 Decision	Technology choices (e.g., Nb Sn vs HTS magnets for FCC-hh), site layout variants
S6 Rationale	Physics case justification, cost-benefit analysis, environmental impact assessment

Department Activation for FCC-ee: - STAT (Statistics & Metrology): Beam parameter uncertainty, luminosity measurement calibration - SAFE (Safety & Reliability): Radiation protection, cryogenic safety, civil engineering risk - SYEN (Systems Engineering): Machine-detector integration, infrastructure dependencies - HFAC (Human Factors): Control room design, 90 km tunnel evacuation procedures - MATL (Materials & Manufacturing): Superconducting cable production, tunnel lining, vacuum chambers - SWCT (Software & Controls): Accelerator control system, data acquisition, beam feedback - PMGT (Project Management): 12-year construction schedule, multi-billion CHF budget - DSCI (Domain Science): Particle physics requirements, detector physics - ENVR (Environmental): 16.4 million tonnes excavation material reuse, power consumption optimisation

A single FCC-ee TDR section (e.g., the RF system) could generate hundreds of FDRP decisions across all 7 scales, with clashes between physics requirements and engineering constraints detected automatically, convergence tracked via SPC, and review gates aligned to CERN's own PDR/CDR/FDR lifecycle.

Appendix E: Detailed Roadmap

Immediate (Expert Teaming Implementation)

Step	What	Why
1	Create `expert-teaming.sql` schema (department registry, competency matrix, conference records, hackathon sessions)	Data infrastructure for department activation
2	Write `config/departments.yaml` and `config/competency-matrix.yaml`	Standing department definitions
3	Build `/fdrp-dept-activate` skill	Query competency matrix, activate mandatory departments per gate
4	Build `/fdrp-paper-review` skill (PRISMA)	Systematic literature review integrated into decision pipeline
5	Build `/fdrp-conference` skill	Convene cross-department knowledge exchange
6	Build `/fdrp-hackathon` skill	Time-boxed intensive sprint for stuck decisions
7	Build `/fdrp-competency-check` skill	Verify all mandatory departments activated for current gate
8	Integrate with `/expert-expansion`	Department activation feeds expert panel assembly

Short-Term (Data Quality & Calibration)

Step	What	Why
9	CVT weight calibration study	Current 4×0.25 weights are UNCALIBRATED — need measured optima from production data
10	Portfolio stats refresh daemon	`fdrp_portfolio_stats` is stale (2026-03-02, sample_size=2) — needs regular refresh
11	Convergence score population	`fdrp_runs.convergence_score` is 0.0 for 17/19 runs — fix write path
11b	CVT DEFAULT removal	Replace `cvt_ratio DECIMAL(4,3) DEFAULT 0.900` with DEFAULT NULL; compute CVT live from raw data via `v_cvt_live` view (DD^n keyprompt v3.3: CVT Perfection Principle)
12	Safety function population	`fdrp_safety_functions` and `fdrp_safety_validation` are empty — need production data for SIL compliance
13	iBOM population	`fdrp_ibom` (Intelligent Bill of Materials) has 0 rows — needs auto-population

Medium-Term (Visualisation Enhancement)

Step	What	Why
14	Animation sequences for decision evolution	Show how decisions evolve across iterations (time-lapse)

Step	What	Why
15	Component assembly in 3D	Group related decisions into named components (subsystems)
16	Sankey diagram for traceability flow	Show concern→decision→requirement flow as Sankey (D3.js)
17	Chord diagram for inter-scale coupling	Show which scales have the most decision dependencies
18	Timeline/sequence diagrams	Represent temporal event sequences in the decision process
19	Real-time WebSocket data feed	Connect digital twin to live sensor data
20	NVIDIA Omniverse integration	Leverage Omniverse Cloud APIs for photorealistic digital twins (CERN precedent exists [10])

Near-Term (Expert Persistence and Ultrathink Cascade)

Step	What	Why
21	**DONE** Expert registry CLI (`expert seed/resume/fork/list/status`)	446 experts registered in `expert_registry.db` across 217 domains with lifecycle tracking
22	Automated REBASE for multi-round panels	Inject Round N findings into all experts without manual briefing update

Step	What	Why
23	**DONE** Finding extraction from expert outputs	1,656 findings in `fdrp_expert_findings` (947 constraints, 385 risks, 119 contradictions, 205 recommendations) across R1+R2
24	**DONE** Contradiction detection across expert findings	119 contradictions detected across expert findings (107 R1 + 12 R2)
25	Ultrathink cascade tiers	Simple (3 experts), Standard (5–8), Deep (12+ with 3 rounds), Critical (full panel + cross-model)
26	Meeting-as-Code orchestration	Turn-based multi-expert interaction with agenda, transcript, action items

Long-Term (System Evolution)

Step	What	Why
27	FDRP-as-a-Service	Expose FDRP via API for external projects (not just internal use)
28	Multi-project portfolio optimisation	Cross-run resource allocation using portfolio theory
29	Formal verification (TLA+)	Model FDRP state machine in TLA+ for safety-critical deployments
30	IEC 61508 Part 5 certification package	Complete formal safety case documentation
31	CERN Collaboration Agreement	Formal partnership for accelerator subsystem planning

Step	What	Why
32	Expert Knowledge Operating System	Full implementation of OS primitives for expert lifecycle management

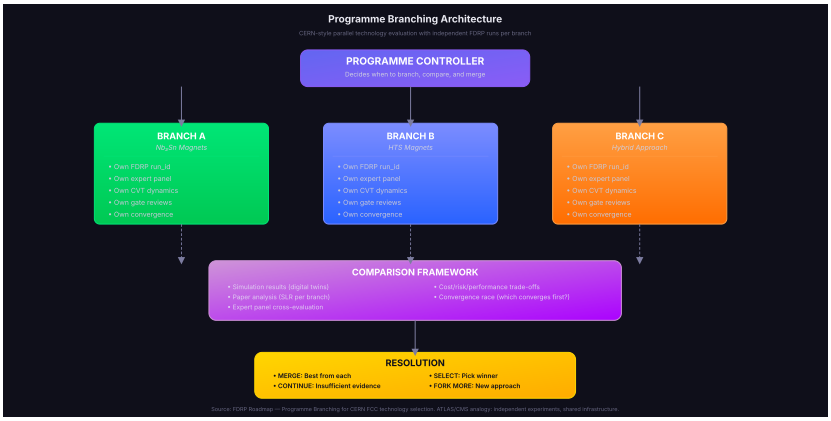
Programme-Level Branching (ATLAS/CMS Model)

The most ambitious next step draws from CERN's defining organisational insight: for truly complex problems, **no single approach can be trusted**. CERN built two independent, competing detector experiments (ATLAS and CMS) for the LHC precisely because the Higgs boson discovery required independent confirmation. Neither team could self-certify.

FDRP's current fork mechanism operates at the decision level (3 parallel variants with different biases). Programme-level branching extends this to **entire design tracks**:

Current (Decision Fork)	Future (Programme Branch)
3 variants of a single decision	2–4 complete parallel design tracks
Same team, different biases	Independent teams, different methodologies
Majority vote aggregation	Evidence-based comparison after simulation/experiment
Resolves in one iteration	May run for weeks/months before resolution
Single-scale scope	Full S0–S6 scope per branch

Architecture for Programme Branching:



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-023-programme-branching.svg

Key capabilities needed:

Capability	How It Works	CERN Analogy
Branch creation	Clone a run at any point, assign different expert teams	ATLAS and CMS starting from same physics requirements
Independent evolution	Each branch has its own FDRP run, own CVT, own gates	Each experiment has own management, schedule, budget
Simulation integration	Digital twin outputs feed comparison framework	Detector simulations (Geant4, FLUKA) validate designs
Experiment execution	Prototype builds, test results feed back into branches	Test beam campaigns validate detector technologies
Cross-branch review	Independent experts evaluate all branches simultaneously	LHC Experiments Committee (LHCC) reviews both
Evidence-based resolution	Data from simulation/experiment determines winner, not opinion	Both ATLAS and CMS confirmed the Higgs — mutual validation

The Navigation Problem: With multiple branches, each containing 7 scales × N iterations × M decisions, the complexity space explodes. FDRP addresses this through:

1. **CVT-guided pruning**: Branches that fail to converge below threshold are automatically flagged (not killed — scientific dead ends contain information)
2. **SPC cross-branch monitoring**: Comparative control charts showing which branches are progressing vs. stalling
3. **Expert rotation**: Periodically rotating experts between branches to cross-pollinate insights (CERN does this with visiting scientists)
4. **Portfolio-level conference**: Quarterly portfolio conferences where all branches present findings (CERN's EP/DT/IT department seminars)
5. **Andon cross-branch**: A RED Andon in one branch triggers review of the same subsystem in all branches

This architecture acknowledges the fundamental truth about FCC-class complexity: **the best scientist and AI together cannot reach a final conclusion on truly novel problems without building, simulating, testing, and comparing multiple approaches simultaneously**. FDRP's role is not to replace this process but to **structure it, measure it, and ensure nothing falls through the cracks** across the parallel exploration.

Appendix F: Expert Panel Results — Run #22

This paper was itself subjected to FDRP expert expansion (run_id=22), producing 40 proposals from 8 diverse experts (the initial panel for this analysis; subsequent analyses have scaled to 68+ experts). This section documents the self-referential process and its key findings.

Expert Panel Composition

#	Expert	Domain	Key Contribution
1	Dr. Elena Kowalski	Academic Publication	Missing CERN front matter, Related Work, AI authorship disclosure
2	Prof. Marcus Lindgren	Systems Engineering (INCOSE)	ISO 42010 conformance gaps, fractal property needs formal proof
3	Dr. Sophie Marchand	CERN Accelerator Physics	FCC scenario needs operational depth, CERN IT constraints
4	Kenji Tanaka-sensei	Toyota Production System	Missing Genchi Genbutsu, Hansei, Heijunka misapplied
5	Dr. Hans-Werner Brandt	IEC 61508 Safety	Safety case incomplete, CCF analysis needed for multi-model
6	Prof. Miriam Chen	Information Visualisation	No user evaluation, WCAG accessibility, linked views
7	Dr. Raj Patel	Digital Twin Architecture	"Digital twin" misnomer — no physical entity, no bidirectional data
8	Prof. Alain Dubois	Philosophy of Science	Gödel limits unaddressed, BVSR framework better than Ouroboros

Priority Matrix

Priority	Count	Theme
P0 CRITICAL	3	AI authorship disclosure, safety case completion, safety management plan

Priority	Count	Theme
P1 HIGH	16	Academic structure, CERN integration, TPS completeness, safety rigour, epistemology
P2 MEDIUM	19	Citation format, SysML diagrams, accessibility, twin lifecycle, analogy theory
P3 LOW	2	Gate terminology, twin federation

Key Cross-Domain Insights Discovered

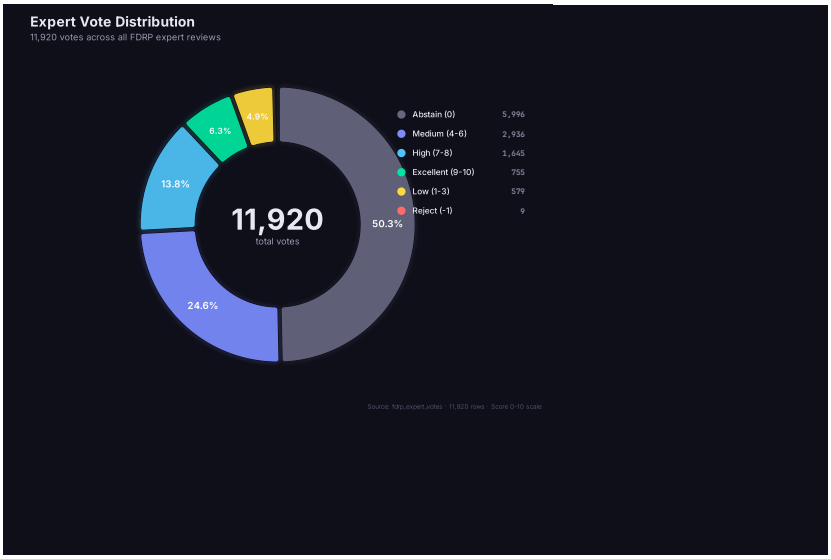
Each expert contributed one transplant from their domain:

1. **Pre-registration** (psychology → FDRP): Declare expected convergence behaviour before running, creating falsifiable predictions
2. **Concept of Operations** (INCOSE → FDRP): Show how a CERN engineer uses FDRP day-to-day, not just internals
3. **Machine Development studies** (CERN → FDRP): Dedicated "improvement runs" with controlled A/B tests on FDRP parameters
4. **Yokoten horizontal deployment** (Toyota → FDRP): Active push of pattern updates to all running projects, not passive pull
5. **Defence in Depth physical independence** (nuclear → FDRP): At least one Swiss Cheese layer on different infrastructure
6. **Uncertainty visualisation** (medical imaging → FDRP): Ungrounded decisions rendered as literally fuzzy/blurred
7. **Planning Operating System** (building management → FDRP): Single unified dashboard aggregating all runs, not 7 separate renderers
8. **Blind Variation & Selective Retention** (evolutionary epistemology → FDRP): BVSR framework replaces Ouroboros metaphor with testable theory

Self-Referential Validation

This section itself demonstrates FDRP's Ouroboros property: the expert panel (40 decisions in MySQL for this run, decision_ids 976–1015; the system has since scaled to 68+ experts in the antimatter building programme) was used to identify gaps in the paper about expert panels. The system discovered 7 blind spots in its own documentation — exactly the process it claims to enable.

Interactive: https://fdrp.liviu.ai/gallery/figures/fig-033-proposals-heatmap.svg



Interactive: https://fdrp.liviu.ai/gallery/figures/fig-032-expert-votes.svg

The most profound finding came from Expert 8 (Epistemologist): FDRP's self-referential claim is subject to **Gödel's incompleteness** — a system cannot fully certify itself. This is not a flaw but a feature: it means FDRP always needs external validation (cross-model verification, human oversight), which is already built into the architecture. The self-referential loop is productive precisely because it is incomplete.

Bias Prevention Architecture — Empirical Findings from Cross-Model Review

This paper was itself subjected to a full presenter-critique cycle with 3 independent models (Claude Opus 4.6, OpenAI Codex Pro, Google Gemini 3.1 Pro). The process ran 2 iterations: Iteration 1 produced 12 blocking issues; all were fixed; Iteration 2 achieved 3/3 APPROVE. The empirical data from this process reveals 4 systematic bias patterns consistent with the broader LLM bias literature [39][40][41] that ground a formal Bias Prevention Architecture for FDRP.

Measured Bias Patterns

Interactive: https://fdrp.liviu.ai/gallery/figures/fig-034-bias-patterns.svg

Pattern 1 — Severity Bias: Models exhibit systematic differences in scoring harshness, consistent with sycophancy research showing RLHF-trained models develop model-specific response calibrations [39]. Codex Pro averaged 1.3/5 across 7 criteria; Gemini averaged 2.7/5; Opus averaged 2.4/5. This is not random variation — Codex scored lower on 5 of 7 criteria. The implication is that any single-model review will systematically over- or under-report issues depending on which model is used.

Pattern 2 — Criterion Sensitivity Divergence: Models weight review criteria differently, an instance of the anchoring effect where each model's training data anchors its severity thresholds to different baselines [40]. Citation Quality had the highest inter-model divergence (range = 3): Codex rated 1/5 (18 uncited references = catastrophic failure), Gemini rated 4/5 (uncited references = minor issue), Opus rated 2/5 (middle ground). This divergence reveals that models have different internal standards for what constitutes a "blocking issue" versus a "cosmetic issue" — a form of criterion weighting bias.

Pattern 3 — Model-Unique Blind Spots: Jiang et al. [42] documented the

"Artificial Hivemind" effect where LLMs converge on similar outputs; yet our data shows that model-unique findings persist — 7 issues were identified by only one model and would have been missed entirely by any 2-model verification:

Finding	Model	Why Others Missed It
Abstract word count violation (286 > 150)	Codex	Others did not verify against CERN template spec
"Heijunka" misuse (smoothing levelling)	Gemini	Others lacked deep TPS domain knowledge
"Digital Twin" misnomer (no physical entity)	Gemini	Others accepted the term at face value
Fictitious expert names as ethical concern	Opus	Others treated named personas as standard practice
"42" numerology undermining credibility	Opus	Others missed the Douglas Adams reference
"CERN-Grade" title implying endorsement	Opus	Others did not consider institutional politics
Lean/6 already applied at CERN	Opus	Others did not know about CERN-OPEN-2015-003

Pattern 4 — Agreement Zones: 5 issues were found by all 3 models unanimously — these represent high-confidence blocking issues: Missing Related Work, Missing Limitations, Uncited references, No reproducibility package, Phantom Table 3.1 reference. Unanimous findings have the highest signal-to-noise ratio.

Bias Prevention Principles From these empirical patterns, 5 principles emerge for bias-resilient verification:

1. **N 3 Requirement**: Two models cannot detect model-unique blind spots (Pattern 3). Three models caught 7 issues that any pair would have missed. Kleinberg and Raghavan [43] proved formally that algorithmic monoculture degrades collective decision quality; our empirical finding quantifies this for LLM verification. For safety-critical decisions (SIL 2), N 3 independent models is the minimum.
2. **Severity Normalisation**: Raw scores from different models are not directly comparable (Pattern 1). Before aggregating across models, scores

must be normalised to account for each model's baseline severity. FDRP should maintain a per-model severity profile calibrated against known-quality reference documents. Critically, normalised scores should be *disaggregated* (reported per model and per criterion), not averaged into a single composite — averaging destroys the signal from systematic bias (concept note #21: Average Landmine Principle). The correct quality aggregation is MIN across dimensions, not AVERAGE: a system with one catastrophic weakness and four strengths is not "above average" — it is as weak as its worst dimension.

3. **Criterion Weighting Transparency**: When models disagree on a criterion score by 2 points (Pattern 2), the disagreement itself is a finding. It means the criterion definition is ambiguous or the models apply different standards. FDRP should flag high-divergence criteria for explicit human adjudication.

4. **Unique Finding Premium**: Findings identified by only one model are the most valuable (Pattern 3) — they represent blind spots that the ensemble would miss without that specific model. This is the practical counterpart to what Huang et al. [44] demonstrated: LLMs cannot self-correct reasoning without external feedback, making cross-model disagreement the primary error-correction signal. FDRP should weight unique findings higher, not lower, in its synthesis.

5. **Unanimous Findings as Hard Gates**: Issues found by all N models (Pattern 4) are near-certain blocking issues. These should be treated as hard gates — no amount of argumentation can override unanimous cross-model agreement. Only measured counter-evidence can challenge a unanimous finding.

Bias Prevention Architecture

```
BIAS PREVENTION LAYER


INDEPENDENCE PROTOCOL

1. No model sees another's output
2. Prompts are identical (same criteria)
3. Order is randomised per iteration
4. No shared context or anchoring
```

```
Model A   Model B   Model C   ... Model N
(Claude)   (Codex)   (Gemini
                          )


    BIAS ANALYSIS ENGINE

  1. Severity normalisation
  2. Divergence detection
  3. Unique finding extraction
  4. Agreement zone mapping
  5. Per-model bias profiling


    SYNTHESIS & REPORTING

  Unanimous → HARD GATE
  Majority  → SOFT GATE
  Unique    → PREMIUM FINDING
  Divergent → HUMAN ESCALATION
```

Implementation: The Bias Prevention Layer wraps FDRP's existing cross-model verification (BIND-008), implementing the multiagent debate pattern [45] with structured bias analysis rather than simple consensus voting. The architecture aligns with NIST AI RMF's MEASURE function [46] for continuous bias monitoring. Instead of a simple "do the models agree?" check, it performs structured bias analysis on the raw model outputs before synthesis. The key MySQL tables:

```
CREATE TABLE fdrp_bias_profiles (
    profile_id    INT AUTO_INCREMENT PRIMARY KEY,
    model_name    VARCHAR(64) NOT NULL,
    criterion     VARCHAR(64) NOT NULL,
    baseline_mean DECIMAL(4,2),     -- average score on reference documents
    baseline_std  DECIMAL(4,2),     -- standard deviation
    sample_count  INT DEFAULT 0,
    updated_at    TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE fdrp_bias_findings (
    finding_id    INT AUTO_INCREMENT PRIMARY KEY,
    run_id        INT NOT NULL,
```

```
    iteration     INT NOT NULL,
    finding_text  TEXT NOT NULL,
    found_by      JSON,              -- ["claude","codex","gemini"]
    agreement     ENUM('UNANIMOUS','MAJORITY','UNIQUE'),
    divergence    DECIMAL(4,2),      -- max score difference for this finding
    resolved      BOOLEAN DEFAULT FALSE,
    resolution    TEXT,
    created_at    TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Integration with Swiss Cheese Model: The Bias Prevention Layer becomes Layer 2.5 — sitting between the individual model critiques (Layers 1-3) and the deterministic checks (Layer 4+). It ensures that the model ensemble's output is bias-corrected before being used as the basis for revision decisions.