# Dyadic Rest Architecture (DRA): A Legality■First, Geometry■Native Framework for Deterministic AI

Version v1.0 — 2025-09-16 23:27 UTC


## Abstract

We present the Dyadic Rest Architecture (DRA), a legality-first, geometry-native framework that makes determinism, replay, and auditability the default in AI systems. DRA declares canonical "rest" states at dyadic scales (2/4/8/16/64) on a fixed 10■D toroidal carrier; everything else is motion: labeled, lawful deltas around the declared rest. Legality is enforced via Type■II Construction■A (Golay exception) leading to Leech-carrier evenness, a Monster order■13 screw (frame shaping), Moonshine envelopes, mirror parity, and 8■face gating with witness latches. Tuple arity selects CRT/ConA layers; odd lifts return to even before dynamics. Retrieval is "anchors, not recall": identity is bound to invariant keys (rest id, delta list, witness coverage), not cosine similarity. We provide algorithms, an example run (R4→R8), falsifiers, an evaluation plan, and a braided API surface suitable for service-mesh deployment.


## 1. Introduction

Modern AI systems optimize for similarity, not legality. This causes drift, opacity, and brittle behavior under distributional shift. Prior work in our sessions (CQE Stack and Scratchpad HQ) showed that if you move decision■making from "closest embedding" to "legally admissible moves on a fixed geometric carrier," you can gate, replay, and audit every step. The Dyadic Rest Architecture (DRA) packages that insight so it can run as a real service: multi■resolution rests, face■only cadence, overlay Return maps, and invariant■keyed retrieval ("anchors").

## 2. Core Thesis (one line)

Declare canonical rests at dyadic scales 2/4/8/16/64 on a 10■D toroidal carrier; require all computation to be lawful deltas around the chosen rest, advancing only when all eight witnesses latch. Identity and retrieval come from invariants (anchors), not recall or nearest■neighbor search.

## 3. Invariants & Carrier (the physics)

• Type■II ConA (Golay exception) ⇒ Leech: even, unimodular ambient; legality baseline. • Monster order■13 screw: fixes the pitch/phase frame ("13A/13B" class slice). • Moonshine envelope: optional, shapes radii/groove without changing legality. • Mirror parity in unison: right strand is time■reverse + π/13 phase shift of left. • 10■D torus T^10: four orthogonal 2■planes (8D) + groove (1D) + axis (1D). • Gating: eight facet functionals (E8■style); cadence advances only after all eight latches fire.

## 4. Dyadic Rest Hierarchy (R2/R4/R8/R16/R64)

R2: one 2■plane active (palindromic sub■view). R4: two 2■planes active (k■parity ordering). R8: full stage (four 2■planes + groove + axis) — primary even rest. R16: R8 + doubly■even tightening

(binary Type■II layer). R64: R16 + dyadic+CRT atlas (6■bit Δ■hypercube around the same center). Rule: pick the smallest rest consistent with active 2■plane count and CRT flags; snap to that anchor.

## 5. State Model (Rest ⊕ Motion)

Any live state decomposes as X = Rest ⊕ Δ, with Rest mirror■even and Δ mirror■odd. Legality demands Type■II evenness + Monster/Moonshine compliance + mirror pairing. Screw identity holds under cadence.

## 6. Face■Only Cadence (8■Face Gating)

OPEN(t) ⇔ Legal(Rest ⊕ Δ) AND all eight facet constraints are satisfied. Witness latches record that a face was legally touched by Rest or by a legal Δ. Cadence advances only when all eight latches = 1; then bounds pulse (tiny outward normal) and latches reset. This gives a geometric clock and deterministic replay.

## 7. Tuple→CRT Construction■A Selector

Tuple arity selects ConA layers: keep the Leech carrier fixed, add q■ary ConA per factor (include mod■13 for screw nativity; 2■powers for dyadic tiers), CRT■glue to a single modulus, then Return to even if any odd lift appears. Intersect the allowed set with T^10 and choose pitch; mirror pairing preserved.

## 8. Overlay Tree & Return

Overlay families label “outside” choices as deltas: P (parity façade), M (modulus/CRT), S (symmetry slice), H (Moonshine), G (gating policy), N (embedding classifier). Overlays are overlay■only until Return proves even legality; only then may dynamics proceed. Composition must be justified when generators don’t commute.

## 9. Anchors, Not Recall

StateAnchor = { id=hash(rest ⊕ Δ ⊕ vers), E=canonical embedding, tags, bounds_sig (8 latch bits) }. Lookup is invariant: compute canonical representation → nearest anchors by exact tags + legality proof, not cosine. If no match, mint a new anchor with its proof. This removes drift, leakage, and non■reproducible hits.

## 10. Algorithms (pseudocode)

select_rest(context) → R ∈ {R2,R4,R8,R16,R64} Λ = [0]*8 # witness latches loop over pulse order: if Legal(R) and H_i(R) ≥ -ε: Λ[i]=1 else: S = {Δ | Legal(R⊕Δ) and H_i(R⊕Δ) ≥ -ε} if S: Λ[i]=1 log(step) if all(Λ): bounds_pulse(); advance_cadence(); Λ=[0]*8 Commit once at the normal form; identity = anchor hash. All steps are ledgered with proofs.

## 11. Worked Example (R4→R8)

Scenario: A user asks a factual question that touches two internal lenses (two 2■planes) and trips a mod■8 constraint (dyadic tag). The engine selects R4. During the face■only sweep, faces 1–6 latch on Rest alone; face 7 requires a legal Δ from overlay M (CRT tag), face 8 requires a minimal Δ from overlay G (policy). All eight latches fire → cadence advances; the normal form is committed with an anchor id. The same inputs later replay to the same anchor. If a second query adds two more active lenses, the engine promotes to R8, proves Return for overlays, and repeats the sweep. In both cases, identity is invariant and the ledger shows which Δ satisfied which faces.

## 12. API Surface (braided, key■scoped)

POST /rest/select → {rest_id, proof} POST /delta/apply → {ok, new_tags, return_proof?} POST /gate/scan → {latches[8], touched_faces, legal?} POST /advance → {ok, cadence_id, bounds_sig} POST /anchor/match → {anchor_id?, mint_if_absent} GET /proof/state → {rest_id, deltas[], latches[], legal_certs[]} mTLS + JWKS; per■family keyscopes (P/M/S/H/G/N); idempotency keys on every call.

## 13. Evaluation Plan & Falsifiers

Metrics: gate efficiency by rest; confluence hash across random pulse orders and rest promotions; energy per commit vs baseline; anchor hit■rate. Falsifiers: (F1) cross■rest non■confluence; (F2) odd overlay admitted to dynamics without Return; (F3) cadence advance without eight latches; (F4) anchor mismatch under identical inputs; (F5) legality breach (Type■II or CRT). Any Fk invalidates the run and emits a minimal counterexample.

## 14. Safety & Governance

Face■only progression + Return guards mean “views” never silently become “acts.” The ledger stores proofs, not private content. Cross■frame comparisons are illegal by construction; undecidable claims trigger explicit, priced extensions (new modulus/face), not speculation. This structure reduces hallucinations to lawful refusals.

## 15. Limitations & Open Questions

Cross■rest proof of confluence needs a formal certificate. R64 semantics should be nailed down (which 6 bits). Witness hysteresis (ε) must be calibrated to prevent floating■point latch inversions. Odd■prime overlays may warrant an explicit “R64★” label. These are tractable engineering/theory tasks.

## 16. Conclusion

DRA packages the CQE + Scratchpad insight into a deployable, deterministic system: dyadic rests for scale, face■only cadence for replayable timing, overlay Return for safety, and anchors for identity. It replaces search with context motion on a lawful carrier—and it’s operable as a modern, braided API.

## Appendix A — Acceptance Tests (sketch)

AT1: R4-only closure → identical anchor across random pulse orders. AT2: R4→R8→R4 round■trip → same anchor hash and witness map. AT3: Odd overlay proposed → rejected until Return cert arrives; then accepted. AT4: Cadence cannot advance with any latch=0 (prove refusal). AT5: Injected illegality (bad CRT glue) → hard REJECT with channel■local reason.

## Appendix B — Session Breadcrumb (recap)

We began with CQE (parity→CRT→ConA→Alena→ledger), built Scratchpad HQ (witness wall, replay), adopted pose■as■gauge, introduced E8/Weyl faces and the 8■vertex Δ atlas, and finally crystallized all of that into DRA: dyadic rests, face■only cadence, overlay Return, and anchors. This document is the unique idea set that ties the stack together for deployment.