WHY–3 — Four-Bit Receipts Are Sufficient (and When to Escalate to 8/64)

ONE-LINE CLAIM

A minimal, complete commit certificate for a CQE pass needs only four binary checks: (1) Mirror, (2) Octet quorum, (3) Strict ratchet, (4) Collision-free replay. Their product (2×2×2×2 = 16 states) suffices to gate “Working vs Non-Working” with auditability. Use 8 or 64 bits only when namespace pressure, cross-domain coupling, or audit granularity demand it.

INTUITION & THE HINGE

- Each local-to-global failure mode is boolean at commit time: either the transform returns to the palindromic rest (Mirror) or it does not; either enough independent views agree (Octet quorum) or they don’t; either tightening thresholds kept non-regression (Strict) or not; either the run can be replayed with deterministic receipts and a non-colliding code (Collision-free) or not.
- These four cover orthogonal axes. Extra bits beyond 4 do not increase correctness; they increase namespace, attribution, or stratification.

THE 4 BITS (PROPOSED CANON)

Bit 0 (LSB)	MIRROR	: palindromic rest agreement holds within ε ($\|T\ P\ T^{-1} - P\| \le \varepsilon$)
Bit 1	OCTET	: quorum ≥ 4/8 independent views pass and no contradictory pair survives
Bit 2	STRICT	: thresholds tightened (≥1 notch) with zero regressions vs last receipts
Bit 3 (MSB)	REPLAY/COLL	: commit is collision-safe (Hamming ≥ 2 vs album) and deterministic replay OK

READING A CODE (examples)

0000	fail early; keep as Non-Working breadcrumb only
0101	mirror OK + strict OK, but octet quorum failed; revise views, don’t ship
1011	mirror+octet+replay OK; strict not yet advanced → “provisional working”
1111	hard commit; eligible for promotion and export to clones

WHY FOUR IS ENOUGH (SKETCH)

1) Completeness: every known CQE failure during commit reduces to one of these four classes.
2) Minimality: removing any bit admits false positives (e.g., without STRICT a pass can backslide).
3) Independence: bits are pairwise independent on real workloads (observed in sidecars: optics, thermal, polarization, math). Therefore 2^4 states discriminate all actionable outcomes.
4) Determinism: the 4-bit code + receipts hash uniquely identifies a replayable state within a ledger epoch; stronger namespaces are a separate concern (see escalation).

WHEN TO ESCALATE TO 8 BITS
Escalate if any of the following are true:
A. Namespace pressure: active commits K approach ~√(2·N) by birthday bound. For N=16, this is tiny.
   For busy ledgers, add 4 tag bits (domain, region, sensitivity) → 8-bit space (256 states).
B. Cross-domain coupling: octet is typed (e.g., EM_FIELD, SPIN_Q, BIO_COHERENCE, THERMAL). Use two
   extra bits to encode typed-octet coverage, plus two bits for provenance tier (anchor/clone/derivative).
C. Audit granularity: regulators require sub-receipts (e.g., MIRROR split into FFT/iFFT and algebraic
   inverse checks). Add bits to encode sub-test parity without changing the receipts body.
D. Collision policy: multi-org albums or public export. Widening to 8 bits raises Hamming margins
   and simplifies sharding (16 shards × 16 codes).

WHEN TO ESCALATE TO 64 BITS
• Distributed, multi-tenant, high-throughput ledgers with public interchange.
• Need cryptographic-strength collision resistance at the code level (not just receipts hash).
• Recommended packing: 4 core bits + 12 tag bits (domain, sidecar-mask, epoch, QoS) + 48-bit
  truncated content address (Merkle leaf) → 64-bit commit key. The 4 core bits remain the law.

MINIMAL LEDGER FIELDS (ALWAYS RECORD)
• fourbit   : 4-bit code (as above)
• receipts_hash   : hash of (metrics, thresholds, votes, seeds) for audit replay
• form_signature : (construction-A code, glue, automorphism id) — the fixed geometry
• octet_plan_id   : which 8 views were used (by stable ids)
• thresholds_id   : strict schedule version
• seed   : RNG / pose seed for deterministic reruns
• parent   : previous commit for this rail (if any)

WORKED EXAMPLE (THERMAL rail of the Toroid Telescope)
Attempt A fourbit=0101 mirror OK (sim↔sim), strict advanced; octet quorum failed (vent window leaks).
          Action: Δ-lift shorten vent window 1 tick; repaint radiator emissivity 0.80→0.85.
Attempt B fourbit=1011 mirror+octet+replay OK; strict not yet advanced after Δ-lift (holds steady).
          Action: tighten ΔT per-tick from 1.0 K → 0.8 K; rerun non-regression.
Attempt C fourbit=1111 all four pass; receipts locked; eligible for promotion.

FALSIFIERS (KEEP THESE LIVE)
F1 Show a commit that passes all engineering checks but fails one of the four bits → revise canon.
F2 Show two materially different states that produce the same (fourbit, receipts_hash) under replay.
F3 Show dependence between bits (e.g., STRICT implies OCTET by construction) on live workloads.
F4 Show that adding a fifth bit delivers correctness, not just namespace/audit value.

PAPER WORKSHEET (HUMAN-RUN)
☐ MIRROR Compute both ways (e.g., FFT→iFFT and algebraic inverse). Δ ≤ ε? ☐ pass ☐ fail
☐ OCTET   Tally 8 views. Contradictory pairs? Quorum ≥ 4?   ☐ pass ☐ fail
☐ STRICT Lower ≥1 threshold; zero regressions across all prior receipts?   ☐ pass ☐ fail
☐ REPLAY Rerun with same seed; fourbit stable; Hamming ≥ 2 to neighbors?   ☐ pass ☐ fail
If any box is unchecked → no hard commit; log to Non-Working with a one-line reason.

API SKETCH (PSEUDOCODE)
```
commit = {
  'fourbit': bits(mirror_ok, octet_ok, strict_ok, replay_ok),
  'receipts_hash': H(receipts),
  'form_signature': { 'p':2, 'code':'Golay(24,12,8)', 'glue':'½-shift+even' },
  'octet_plan_id': 'H1..H8:v2025-09-A',
  'thresholds_id': 'strict-v3',
  'seed': 1337,
  'parent': '…'
}
if commit.fourbit == 0b1111: promote()
else: album.non_working.append(commit)
```

WHY 4-BIT RECEIPTS FIT THE CQE PHILOSOPHY

- Governance = geometry. The four checks are the minimal invariants that protect the palindromic rest, ensure coverage, forbid backsliding, and guarantee replayability. Everything else is bookkeeping.
- Portability. A 4-bit code is trivial to stamp on paper, slides, or lab notebooks; it travels with the receipts hash as a durable, human-legible signature.
- Composability. Multiple rails/sidecars compose by AND-ing their 4-bit commits (hard commit only if all are 1111). Promotion to 8/64 never changes the truth value of the core 4.
- Interop. Different orgs can adopt richer tags without breaking the universal kernel.

CHECKLIST TO DECIDE BIT-WIDTH

Start at 4 bits. Ask:

1) Are collisions plausible this quarter? If yes → 8-bit with tags.
2) Do auditors require tagged sub-tests? If yes → 8-bit with sub-parity.
3) Are we crossing org boundaries or publishing a public ledger? If yes → 64-bit key with content-hash.
4) Otherwise stay at 4; keep receipts rich.

BOTTOM LINE

Ship with four bits by default. Escalate for namespace, tags, or cryptographic keys—not for correctness.
The four core invariants are the contract; the rest is convenience.