

## # NOORCHAIN 2.1 — M21 RC Verification Note (Internal)

**\*\*Document type:\*\*** Internal verification note (RC evidence)

**\*\*Scope:\*\*** Protocol operations and client-compatibility gates (HTTP + WS), leader/follower parity

**\*\*Not a third-party security audit.\*\***

**\*\*Native token:\*\*** NUR (JSON-RPC keeps `eth\_` methods for compatibility)

---

### ## 1. Executive Summary

NOORCHAIN 2.1 has reached a frozen **\*\*Mainnet-like Complete — Release Candidate (RC)\*\*** baseline.

This note documents:

- the **\*\*RC baseline reference\*\*** (tag/commit),
- the **\*\*verification gates\*\*** re-run on a clean environment,
- the **\*\*evidence artifacts\*\*** produced during verification,
- the current **\*\*non-minimal gap list\*\*** (limited and explicit),
- the locked roadmap beyond M21 (M22–M27).

This verification is an internal, reproducible, operational proof of compatibility and liveness properties.

It does **\*\*not\*\*** replace an independent security audit.

---

### ## 2. Baseline Reference (Frozen)

- **\*\*RC tag:\*\*** `M21-MAINNETLIKE-COMPLETE-RC`

- **\*\*Runbook reference:\*\*** committed baseline (commit `8ecb6f2`)
- **\*\*Branch:\*\*** `main`

---

### ## 3. Verification Scope

The verification covered:

#### ### 3.1 Network topology (mainnet-like local pack)

- 2-node setup: **\*\*leader\*\*** and **\*\*follower\*\***
- follower reads routed via `FollowRPC` where applicable
- ports and processes validated before and after restart

#### ### 3.2 HTTP JSON-RPC (compatibility gates)

- `eth\_chainId` parity across leader/follower
- `eth\_blockNumber` advances and follower alignment
- `eth\_call` for PoSS registry read-path
- `eth\_getLogs` range-based log retrieval with leader/follower parity

#### ### 3.3 WebSocket JSON-RPC (subscription gates)

- `eth\_subscribe("newHeads")` on leader
- `eth\_subscribe("newHeads")` on follower (proxy semantics)
- `eth\_subscribe("logs")` on leader for PoSS registry address
- Live event capture validated using an on-chain submission

#### ### 3.4 Restart invariants

- clean stop
- clean restart of leader/follower
- post-restart read-path continuity for PoSS state

---

## ## 4. Operational Invariants (Hard Requirements)

The following invariants were enforced during verification:

- **Native token is NUR** (no ambiguity with Ethereum native asset)
- JSON-RPC uses ``eth_*` methods strictly for compatibility
- **Private Key Gate (mandatory)**:
  - ``NOOR_PRIVATE_KEY`` must be set at runtime (0x-prefixed, length 66)
  - private keys must never be committed or written to disk
- **Registry Gate (mandatory)**:
  - ``NOOR_POSS_REGISTRY=0xC9F398646E19778F2C3D9fF32bb75E5a99FD4E56``

---

## ## 5. Evidence Artifacts

The verification produced and archived the following evidence:

- **P0 audit evidence (runtime capture)**:
  - ``/tmp/P0_AUDIT_EVIDENCE_20260129T181118Z.txt``
- **Non-minimal gaps (1-page list)**:
  - ``/tmp/P0_NON_MINIMAL_GAPS_20260129.md``

These artifacts are intended to be reproducible from the baseline runbook and can be re-generated at any time.

---

## ## 6. Results (Pass/Fail Summary)

### ### 6.1 PASS — Core Compatibility

- `eth\_chainId` parity: PASS (`0x849`)
- `eth\_blockNumber` advances: PASS
- follower alignment (read path): PASS
- WS `newHeads` leader: PASS
- WS `newHeads` follower (proxy): PASS
- WS `logs` subscribe: PASS
- **Live log capture** (submitSnapshot → WS log observed): PASS
- `eth\_getLogs` leader/follower parity (single-block range): PASS
- restart invariants + PoSS read continuity: PASS

### ### 6.2 Recorded Non-Minimal Gap(s)

See Section 7.

---

## ## 7. Non-Minimal Gap List (Explicit)

This RC baseline is “mainnet-like complete” for compatibility and operational gates.

It is **not yet** “non-minimal complete”. The following gap was observed during verification:

### 1) **WS newHeads** may sometimes emit zero roots

- Observation: at least one `newHeads` payload included:
  - `stateRoot = 0x000...000`
  - `receiptsRoot = 0x000...000`
- Example recorded during live session: blockNumber `0x11c0c` (UTC session 2026-01-29).

- Impact: non-minimal clients may treat zero roots as placeholder/invalid.
- Status: requires confirmation of determinism and scope; planned to be addressed in M22/M23.

The full gap note is stored in:

`/tmp/P0\_NON\_MINIMAL\_GAPS\_20260129.md`

---

## ## 8. Locked Roadmap Beyond M21 (Post-RC)

The post-M21 roadmap is locked as follows:

- **P0 Internal audit** (this note): re-run gates + archive evidence + 1-page gaps
- **M22:** Full EVM execution (remove any “proof writes” and rely on real execution path)
- **M23:** Real world-state persistence + real `stateRoot` commitment per block
- **M24:** Expanded RPC parity (client-grade semantics)
- **M25:** txpool / pending realism
- **M26:** fee model (documented choice and behavior)
- **M27:** hardening (limits, reliability, failure modes)
- Final freeze only after non-minimal gates PASS

---

## ## 9. What This Note Is (and Is Not)

**This note is:**

- a reproducible internal verification of M21 RC compatibility gates,
- a clear evidence anchor for partners and internal tracking.

**This note is not:**

- an independent security audit,
- a guarantee of vulnerability absence,
- a statement about financial outcomes.

---

## ## 10. Contact / Change Control

Any changes to the RC baseline must be made via explicit versioned milestones (M22+).

The M21 RC baseline remains frozen for reference and reproducible evidence generation.