Known Limitations

Honest Assessment

Lux Network

This document is a deliberate counter-balance to marketing material. It states what the Lux MPTC stack does *not* do, what is not yet production-ready, what is research, and what is open work.

1 Cryptographic limitations

1.1 The Z-Chain Groth16 wrapper is classical, not post-quantum

The optional Groth16 wrapper around the ML-DSA cert set (LP-307) is a pairing-based SNARK over BLS12-381. Pairing-based SNARKs break under Shor's algorithm. The Groth16 wrapper compresses ML-DSA cert-set verification for EVM verifier compatibility; it is **not** a PQ contribution. PQ finality lives in ML-DSA + Pulsar.

This is acknowledged across the spec corpus (`proofs/quasar/horizon-soundness.tex` Remark `rem:groth16-wrapper`; `proofs/definitions/finality-definitions.tex` Remark `rem:groth16-not-pq`; LP-105 § Proof-lane classification). A future PQ-friendly proof system (P3Q, `luxfi/p3q`, or lattice-based) replaces Groth16 in the wrapper role; the production roadmap is clear, but the freeze tag ships Groth16-classical and labels it accordingly.

1.2 There is no production threshold ML-DSA

ML-DSA's per-validator signatures are independent; we run one keypair per validator. There is no FIPS standard for threshold ML-DSA. Research constructions are tracked but not in production; the rejection-sampling circular dependency (`quasar-cert-soundness.tex` App. A) blocks straightforward composition.

If a production-ready threshold ML-DSA appears, it could replace the per-validator ML-DSA cert set with a single threshold signature. Until then, per-validator ML-DSA + optional Groth16 compression is the production path.

1.3 Pulsar Bootstrap requires either a foundation MPC ceremony or DKG2

The default Pulsar Bootstrap path is a one-time foundation MPC ceremony. A toxic-waste assumption is confined to genesis of one key era; subsequent eras run on Reanchor, which itself requires a fresh ceremony or a Pedersen DKG over R_q (`pulsar/dkg2/`).

The DKG2 path eliminates the toxic-waste assumption but is not yet production-default; it ships in parallel as a research production option (LP-073 § Pedersen DKG over R_q). For chains with hard "no trusted dealer ever" requirements, DKG2 is the answer; for chains willing to accept a one-time genesis ceremony per era, the foundation MPC path is the lower-complexity production default.

1.4 Group assignment for grouped Pulsar is open research

Section 7 of the Quasar Horizon paper enumerates five open systems-research questions for grouped Pulsar deployments. The most pressing: a closed-form trade-off between G (group count), k (group size), f (corruption fraction), and per-group Pulse latency under WAN churn. The freeze tag ships $G = 1$ for the production $n = 21$ Lux mainnet; multi-group code paths are tested at $G \in \{2, 4, 8\}$ in `protocol/quasar/grouped_threshold_test.go` but not yet production-deployed at $n > 100$.

1.5 Pulsar upstream Feldman DKG is pseudoinverse-recoverable

The upstream Pulsar academic reference uses a Feldman-style commit DKG that has been shown pseudoinverse-recoverable in our internal red review (`luxcpp/crypto/corona/RED-DKG-REVIEW.tex`). Pulsar replaces this with either a foundation MPC ceremony or a Pedersen-style DKG over R_q (`pulsar/dkg2/`); the upstream DKG is not used in the Lux production stack.

This is documented; a chain operator who imports the upstream Pulsar reference verbatim without the Lux DKG replacement would inherit the upstream weakness.

2 Implementation limitations

2.1 SDK status (honest assessment)

Language	Status
Go	Production-ready
Python (`pkg/python/`)	Only complete non-Go SDK with real consensus logic
C (`pkg/c/`)	Data structures only, not real consensus
Rust (`pkg/rust/`)	FFI wrapper around C, not native
C++ (`pkg/cpp/`)	Stub

Production deployments must use the Go canonical. Other-language bindings exist for tooling and read-only paths but should not be relied upon for production consensus.

2.2 GPU production path is not enabled

The GPU code paths under `corona/gpu/{metal,cuda,wgsl}/` (LP-137) are research, not production. Reasons:

- Constant-time review of GPU code paths is not complete.
- Cross-platform KAT byte-equality (Metal + CUDA + WGSL) is not yet validated.
- The CPU path runs comfortably within the consensus block budget; GPU is a 2–6× improvement, useful but not blocking.

The freeze tag ships CPU-only for production. GPU is on the roadmap (LP-137) and projected numbers are reported in the benchmark report.

2.3 Fuzz harnesses run for 60 s in CI

The fuzz harnesses (`warp/pulsar/fuzz_*.go`, `warp/fuzz_envelope_test.go`, `pulsar/sign/fuzz_*.go`) run for 60 s per harness in CI. This is a production-grade fuzz cadence but does not constitute a formal verification. A bug missed by the harness for 60 s is still possible; the production posture is to add new corpus entries on every find and to keep the harness running indefinitely on dedicated infra (the fuzz-discovered lattigo `ReadUint64Slice` vulnerability of Mar-3 was found within a 60-s fuzz run, which is encouraging but not exhaustive).

2.4 Constant-time review is layer-specific

The constant-time review (`pulsar/CONSTANT-TIME-REVIEW.tex`, `lens/CONSTANT-TIME-REVIEW.tex`) covers the lattice math and the curve math respectively. It does not cover:

- The Go runtime's garbage-collection-induced timing variance (fundamental to Go).
- Cache-timing on heterogeneous hardware (varies across deployment targets).
- Constant-time GPU code paths (GPU production not enabled).
- The Lattigo deserializer (the lattigo `ReadUint64Slice` finding of Mar-3 demonstrates that the deserializer is a separate hardening surface).

For deployments that need stricter constant-time guarantees, the recommendation is to deploy on bare-metal x86 with `GOGC=off` and custom thread-pinning; the Lux production stack does not currently enforce this.

2.5 Snapshot retention is finite

LSS snapshot manager retains 8 generations by default (`threshold/protocols/lss/rollback.go`). Rollback to a snapshot older than 8 generations requires a Reanchor (governance event). For chains that need indefinite snapshot retention (audit / compliance), the snapshot retention window is configurable via chain governance, but storage cost scales linearly.

3 Operational limitations

3.1 Cross-WAN p99 latency is sensitive to network conditions

The cross-WAN $K = 21$ Pulse latency at p99 is ~ 1.3 s on Lux mainnet's current 5-region deployment (Section 8 of the benchmark report). Transient WAN events (BGP routing changes, undersea cable issues, regional-DC outages) can spike p99 above the 3-second bundle interval; in such cases the bundle stays at Beam-final and the next bundle's transitive transcript inclusion absorbs the lag. This is graceful degradation, not failure; downstream consumers that depend on per-bundle Horizon-final must be prepared for occasional lag.

3.2 Validator-set rotation is a discrete event, not continuous

LSS resharing is triggered by epoch transitions, not continuously. A validator that joins or leaves at the start of an epoch is reflected in the next reshare; intra-epoch joins and leaves are not reflected. For chains that need finer-grained validator-set evolution, a custom epoch boundary policy can be configured, but the resharing cost caps the epoch frequency in practice.

3.3 Reanchor requires governance

A new η (KeyEraID) is created via a Reanchor governance event. Reanchor is intentionally heavyweight to prevent accidental key-era resets. For chains that need a faster Reanchor cadence (e.g., monthly key-era rotation for security hygiene), the governance configuration can be adjusted, but each Reanchor requires either a foundation MPC ceremony or a DKG2 run, neither of which is sub-second.

4 Security analysis limitations

4.1 UC treatment of the full composition is open work

The current proof bucket carries SUF-CMA reductions and BFT-style safety / liveness theorems. A universally composable (UC) treatment of the full Quasar Horizon composition — Beam + ML-DSA + Pulse + LSS lifecycle + DAG-BFT engine + Lumen transport — is open work. The Tamarin spec at `tamarin/QuasarConsensus.spthy` verifies the protocol-level handshake / vote / bundle flow under standard symbolic abstractions; a UC treatment with concrete security bounds remains future work.

4.2 Adaptive corruption is bounded but not zero

The proof bucket supports static-corruption reductions; adaptive-corruption reductions appeal to Fischlin / erasure hybrids (`quasar-cert-soundness.tex` App. C). The hybrid argument adds factors that are absorbed in the concrete-security parameter; the production parameters give classical $\geq 2^{142}$ and quantum $\geq 2^{130}$ even after the hybrid loss. For deployments with long-term archival requirements (decades), the parameter sets should be re-validated against future cryptanalytic advances on Module-LWE / Module-SIS.

5 In and out of scope

In scope for this submission

Pulsar (kernel + lifecycle), Lens (kernel + lifecycle), LSS (lifecycle framework), Quasar Horizon (consensus composition), Warp 2.0 (cross-chain envelope). All as documented in the spec/ directory.

Out of scope for this submission

Lumen (PQ encrypted transport — forthcoming Lumen LP), Warp Private (FHE-encrypted payload — production research), threshold ML-DSA (no FIPS standard), grouped Pulsar at $G > 1$ at $n > 100$ (open research), GPU production (research path under LP-137).

These items are tracked in the future-work sections of the companion papers (`luxquasarhorizon` §10, `luxwarpv2` §Future). Each will be the subject of a separate submission or revision to this submission as it reaches production-readiness.