



Lux MPTC Domain- Separation Registry

Zach Kelling

Lux Industries

2026-03-03

Abstract

We present Lux, a hybrid post-quantum consensus architecture for DAG-native chains. Lux separates fast operational finality from durable post-quantum finality. Validators emit Photons over a Nebula DAG substrate; classical BLS aggregates form Beams for low-latency confirmation, while ML-DSA attestations and Pulsar lattice-threshold Pulses provide post-quantum finality over Nebula roots. Prism binds all certificate lanes to the same transcript, and Quasar reaches Horizon finality when the bound lanes verify.

This document. The complete domain-separation registry for the Lux MPTC stack. Every QUASAR-* / PULSAR-* / LENS-* / WARP-* personalization tag is enumerated, with the function it personalizes and the canonical reference. Cross-tag replay is structurally impossible by collision resistance of TupleHash256 / cSHAKE256 / KMAC256 with personalization (FIPS 202 / NIST SP 800-185).

1 Pulsar tags

1.1 Hash-suite personalizations (Pulsar-SHA3 default)

Tag	Function	Use
PULSAR-HC-v1	cSHAKE256 32-byte hash	Challenge hash $H_c(\tau)$ over Pulsar transcript
PULSAR-HU-v1	cSHAKE256 XOF	Discrete-Gaussian challenge derivation $H_u(\tau, n)$
PULSAR-TRANSCRIPT-v1	TupleHash256	Pulsar transcript hash $H_T(\tau_1, \dots, \tau_k)$
PULSAR-PRF-v1	KMAC256	Pseudo-random function over share state
PULSAR-MAC-v1	KMAC256	MAC over Pulsar wire frames
PULSAR-PAIRWISE-v1	KMAC256	Pairwise PRF/MAC material between two parties

Reference: Definition `def:pulsar-sha3` of `proofs/definitions/transcript-binding.tex`; canonical Go: `pulsar/hash/`.

1.2 Lifecycle message tags

Tag	Function	Use
QUASAR-PULSAR-ACTIVATE-v1	Pulsar threshold signature	Pulsar key-era activation message under unchanged GroupKey
QUASAR-PULSAR-BUNDLE-v1	Pulsar threshold signature	Pulse over Nebula bundle root in consensus path
QUASAR-PULSAR-RESHARE-v1	Pulsar reshare transcript	VSR transition from old to new validator set
QUASAR-PULSAR-REFRESH-v1	Pulsar refresh transcript	HJKY97 same-set zero-poly refresh
QUASAR-PULSAR-REANCHOR-v1	Pulsar reanchor transcript	Fresh ceremony for new key era

Reference: Definition `def:pulsar-activation-msg` of `proofs/definitions/transcript-binding.tex`; canonical Go: `pulsar/reshare/activation.go`, `pulsar/reshare/transcript.go`.

2 Lens tags

2.1 Hash-suite personalizations

Tag	Function	Use
LENS-CHALLENGE-v1	SHA-512 (Ed25519) / SHA-256 (secp256k1) / SHA-512 (Ristretto255)	FROST challenge hash
LENS-NONCE-v1	per-curve	FROST nonce derivation
LENS-COMMIT-v1	TupleHash256	Lens commit hash for VSR
LENS-TRANSCRIPT-v1	TupleHash256	Lens transcript hash (mirror of Pulsar's)

Reference: `lens/hash/`; algorithm signatures in Definition `def:lens-sign` of `proofs/definitions/algorithm`

2.2 Lifecycle message tags

Tag	Function	Use
QUASAR-LENS-ACTIVATE-v1	Lens threshold signature	Lens key-era activation under unchanged GroupKey
QUASAR-LENS-RESHARE-v1	Lens reshare transcript	Curve-side VSR
QUASAR-LENS-REFRESH-v1	Lens refresh transcript	Curve-side same-set refresh
QUASAR-LENS-REANCHOR-v1	Lens reanchor transcript	Curve-side fresh ceremony

3 Quasar Horizon tags (consensus composition)

Tag	Function	Use
QUASAR-TRANSCRIPT-v1	TupleHash256	The canonical <code>QuasarTranscript</code> over (NebulaRoot, chain_id, network_id, epoch, round, validator_set_hash, KeyEraID, Generation, hash_suite_id)
lux-quasar-horizon-v1	FIPS 204 ML-DSA context	ML-DSA cert set context string for Horizon transcript
QUASAR-PRISM-v1	Domain marker	Prism predicate output binding (informal; not on the wire)

Reference: Definition `def:horizon-transcript` of `proofs/definitions/transcript-binding.tex`; canonical Go: `consensus/protocol/quasar/horizon.go`.

4 Warp 2.0 tags (cross-chain envelope)

Tag	Function	Use
WARP-PULSAR-ENVELOPE-v1	TupleHash256	The Warp 2.0 envelope's Pulse signs over (sourceChainID, payload, sourceNebulaRoot, sourceKeyEraID, sourceGeneration)
WARP-PULSAR-ENVELOPE-BATCHED-v1	TupleHash256	Reserved for future batched Warp 2.0 envelopes (multiple source Nebula-Roots in one Pulse)
WARP-MLDSA-ENVELOPE-v1	FIPS 204 ML-D SA context	ML-DSA cert set context string for Warp 2.0 envelope

Reference: Definition `def:warp-v2-envelope` of `proofs/definitions/transcript-binding.tex`;
canonical Go: `warp/pulsar/pulsar.go` `SigningPrefix` constant.

5 Cross-protocol context strings (FIPS 204 ML-DSA)

Every ML-DSA / SLH-DSA caller in the Lux stack binds signatures to a context string per FIPS 204/205. The full registry:

Context	Used by	File
lux-x-chain-utxo-v1	UTXO Fx plugins	<code>utxo/mldsafx</code> , <code>utxo/slhdafx</code>
lux-evm-precompile-mldsa-v1	EVM precompile	<code>precompile/mldsa/contract.go</code>
lux-evm-precompile-slhdsa-v1	EVM precompile	<code>precompile/slhdsa/contract.go</code>
lux-reshare-v1	Key resharing HKDF	<code>threshold/mpc</code>
lux-wave-v1	Wave voting	<code>consensus/protocol/wave</code>
lux-quasar-horizon-v1	ML-DSA cert set on Horizon	<code>consensus/protocol/quasar/horizon.g</code>
lux-warp-v2-mldsa-v1	transcript ML-DSA cert set on Warp 2.0 envelope	<code>warp/pulsar/pulsar.go</code>

6 Versioning discipline

Every personalization tag includes a version suffix (`-v1`). When a tag's semantics change, the version is bumped (`-v2, ...`), and the old tag remains valid for already-signed objects. There is no negotiation between versions; senders and receivers agree on the version out-of-band (typically via the chain's protocol version or the Warp envelope's wire version).

No tag is reused across protocols. A future `WARP-PULSAR-ENVELOPE-v2` would not collide with `QUASAR-PULSAR-ACTIVATE-v1` even if the underlying message bytes happened to coincide; the personalization tag prevents the cross-context replay.

7 Why this matters

7.1 Cross-protocol replay

Without tag separation, a Pulse over a Nebula bundle root could be lifted into a Pulse over a Warp envelope (or vice versa). The wire bytes would be syntactically distinguishable but semantically interchangeable. Tag separation makes the bytes semantically distinguishable: a `TupleHash256` with one personalization is not equal to a `TupleHash256` with another personalization, even on identical inputs.

This is the formal claim behind LP-073 [1] §"Domain-separated message prefixes": the QUASAR-PULSAR-BUNDLE-v1 prefix prevents a bundle Pulse from being replayed as a Warp envelope Pulse.

7.2 Cross-suite collision

Pulsar-SHA3 and Pulsar-BLAKE3 use distinct personalization tags within their respective hash families. A signature produced under one suite does not verify under the other (Theorem `thm:hash-suite-separation` of `proofs/pulsar/hash-suite-separation.tex`). The HashSuiteID is bound into every transcript; cross-suite forgery would require a collision on TupleHash256 with the suite's personalization.

7.3 Cross-version collision

A future v2 protocol bump uses a v2 tag suffix. Signatures under v1 do not verify as v2 signatures. This is forward-only versioning: the old protocol continues to exist for already-signed objects, but the new protocol's tags are distinct.

8 Test coverage

The domain-separation discipline is enforced in code:

- `warp/pulsar/hashsuite_mismatch_test.go`: tests that mismatched HashSuiteID returns `ErrSuiteMismatch`.
- `pulsar/hash/` unit tests: each hash function's personalization is checked against canonical KAT vectors.
- `pulsar/reshare/negative_transcript_test.go`: 17 transcript-mutation cases $\times$ 3 sites confirming a wrong tag fails verification.

9 Adding a new tag

The procedure for adding a new domain-separation tag:

1. Choose the tag string per the LP-105 [2] naming rules: cosmological / optical / cryptographic, Lux-owned, technically accurate.
2. Add a versioned suffix (-v1).
3. Document the tag in this registry (this file is normative).
4. Add a test in the appropriate `*_test.go` that verifies the tag is bound into the relevant transcript.
5. Update the canonical reference (e.g., `proofs/definitions/transcript-binding.tex`).

The registry is the single source of truth. A signature using a tag not in the registry is unverifiable by definition.

References

- [1] Zach Kelling. Pulsar: Dynamic lattice threshold signatures for permissionless validator sets. Lux Industries, Inc.; `papers/lp-073-pulsar/`, 2026.
- [2] Lux Industries. LP-105: The lux finality stack. Lux Improvement Proposal, 2026.