Cert Profile Modes: Operator-Facing PQ-off / PQ-fast / PQ-strict / PQ-heavy Ladder

Zach Kelling

Lux Industries

2026-06-03

Abstract

LP-017 v1 named cert profiles after astronomical phenomena (Pulsar, Aurora, Polaris). Those names describe internals, not posture: an operator cannot read "Aurora" and know whether their cert is PQ-protected or how much latency it costs. This LP replaces the operator-facing surface with a 4-mode additive ladder: **PQ-off** (BLS only), **PQ-fast** (BLS+Pulsar), **PQ-strict** (BLS+Pulsar+Corona), **PQ-heavy** (BLS+Pulsar+Corona+Magnetar), with three pure-PQ variants. Seven valid configurations total. The QuasarCert wire format (LP-182) is unchanged. The Corona/Pulsar parity audit (`lux/threshold/protocols/parity/parity_test.go`, 2026-06-03) measured Corona at 141× Pulsar cost at N=64 on apples-to-apples Apple M1 Max + DGX Spark GB10 – algorithmic, not optimization. PQ-fast = `BLS + Pulsar` is therefore locked; PQ-strict adds Corona on top for the never-trust-aggregator security property. This paper expands on LP-217 (`lps/LP-217-cert-profile-modes.md`) with the parity audit verdict, the per-mode use cases, and the LP-202 monotonic tier-degradation contract that lets a single cert serve all relying-party policies.

Contents

1 Introduction

This paper expands on LP-217 (`lps/LP-217-cert-profile-modes.md`) with the Corona/Pulsar parity audit verdict (2026-06-03), the per-mode use cases, and the LP-202 monotonic tier-degradation contract that lets a single cert struct serve all relying-party policies.

1.1 What was wrong before

LP-017 v1 named cert profiles after astronomical phenomena: Pulsar, Aurora, Polaris. Three concrete failures of the codename surface:

1. **Posture is opaque.** "Aurora" does not announce that BLS is the fast-path leg or that lattice diversity is present. Operators ask "is Aurora PQ?" – the answer requires reading LP-017's specification text.
2. **Ordering is non-monotone.** Pulsar → Aurora → Polaris is the v1 security ordering, but the names give no hint that Polaris is a superset of Aurora. A new operator reading the codenames in alphabetical order builds the wrong mental model.
3. **No room for pure-PQ variant.** A chain that does not need a classical fast-path (post-PQ-transition L1, sovereign network not anchored to a BLS bridge) has no v1 name; the v1 spec assumes BLS is always present.

1.2 What this LP solves

The 4-mode ladder fixes all three: names declare posture, ordering is monotone (each step is a superset), and pure-PQ variants are first-class (prefix `strict-`).

$$\text{PQ-off} \subset \text{PQ-fast} \subset \text{PQ-strict} \subset \text{PQ-heavy} \tag{1}$$

Each step adds exactly one independent security property. The pure-PQ variants drop BLS:

- `strict-PQ-fast` = Pulsar
- `strict-PQ-strict` = Pulsar + Corona
- `strict-PQ-heavy` = Pulsar + Corona + Magnetar

Seven valid configurations total. The QuasarCert wire format (LP-182) is unchanged; the cert struct shape is unchanged; the LP-017 v1 composition predicates (`IsPolaris`, `IsDoubleLattice`, `HasHashBased`) are unchanged. The new mode name is a single header byte on the operator config knob – `quasar.cert_mode`.

2 The Four Modes

Each mode defines a REQUIRED leg set. A cert is at mode M if and only if every leg in M's required set is present and verified. Modes are totally ordered by subset inclusion within their family (classical or pure-PQ).

2.1 Magnetar parameter set choice – locked 2026-06-04

SLH-DSA at the 192-bit security level has two parameter sets: *192s* (small, 16 KB sigs, 795 ms/sign on Spark) and *192f* (fast, 35 KB sigs, 36 ms/sign on Spark). The matrix bench measured 192s, which is what `~/work/lux/threshold/protocols/magnetar/` ships today. At 795 ms per per-validator sign, *no Magnetar- bearing profile is sub-second at any measured N*: PQ-heavy on Spark at N=64 blows past 1.33 s. Switching to 192f drops the per-validator sign 22× to 36 ms and restores the sub-1 s budget. Production deployments configuring PQ-heavy MUST use 192f.

Mode	Required legs	Recommended use	M1 wall	Spark wall
`PQ-off`	BLS	HFT trading; non-financial L2s	6.9 ms	6.3 ms
`PQ-fast`	BLS + Pulsar	Payment rails; gaming; DEX matching	391 ms	585 ms
`PQ-strict`	BLS + Pulsar + Corona	**Default** – custody, treasury, regulated securities	360 ms	595 ms
`PQ-heavy`	BLS + Corona + Pulsar + Magnetar	Cross-chain bridges, sovereign anchors	2.22 s	1.33 s

Table 1: Four classical modes with measured wall-clock per signing round at N=64 (Quasar 4-scheme matrix bench, 2026-06-04). M1 Max (10-core darwin/arm64); Spark = NVIDIA GB10 Grace Blackwell (20-core linux/aarch64). Cert size at N=64: PQ-off 96 B, PQ-fast 33 154 B, PQ-strict 245 634 B, PQ-heavy 1.28 MB. Magnetar measured with SLH-DSA-192s – see §2.1 for the 192f production lock.

2.2 GPU dispatch is unmeasured today

The matrix bench captured 4595 samples of `nvidia-smi dmon` during the 2-hour run on Spark: mean `SM%` = 11.9 % (idle / background only). **The GPU was not exercised by any path in this bench.** Two boundary gaps prevent dispatch:

- `lux-lattice.pc` is not installed on Spark (`find_package(MLX REQUIRED)` fails on the CUDA target's dependency chain), so `lattice/v7/gpu` doesn't build under `-tags "cgo gpu"`.
- Magnetar SLH-DSA's CUDA kernel exists and passes KAT tests on Spark, but `lux_slhdsa_*` entry points are not yet exposed in `lux/accel/c_api.h`.

GPU numbers cited in earlier LP-203 drafts (PQ-strict ≈15 ms–50 ms) are projections, not measurements. Until both boundary gaps close, all GPU latency claims in downstream LPs are tagged "projected, not measured."

2.3 Pure-PQ variants

Mode	Required legs	Recommended use
`strict-PQ-fast`	Pulsar	Post-PQ-transition L1 with no BLS bridge dependency
`strict-PQ-strict`	Pulsar + Corona	Sovereign PQ-only treasury chain
`strict-PQ-heavy`	Corona + Pulsar + Magnetar	PQ-only long-archival anchor

Table 2: Pure-PQ variants drop BLS.

2.4 Mode-to-internal mapping

The v1 codenames remain in the codebase as INTERNAL identifiers (struct names, package paths, EasyCrypt artifact filenames). The mode-to-internal map for operator config translation:

2.5 Operator config

One struct – `CertPolicy` – declares the full cert posture for a chain at genesis. Three previously-scattered knobs (`cert_mode`, `cert_timeout_ms`, and LP-204's per-L1 mode picker) collapse to four fields of one record. LP-202 references `CertPolicy.TimeoutMs`; LP-204 attaches a

v2 mode	Required legs	Closest v1 codename	Notes
`PQ-off`	BLS	(no v1 name)	v1 implicitly required at least one PQ
`PQ-fast`	BLS + Pulsar	v1 "Pulsar profile"	Locked by parity audit
`PQ-strict`	BLS + Pulsar + Corona	Aurora	Same lattice diversity; ZK profile-ortl
`PQ-heavy`	BLS + Corona + Pulsar + Magnetar	Polaris	Cross-family maximum
`strict-PQ-*`	PQ legs only	(no v1 name)	New in v2

Table 3: v2 mode names mapped to v1 internal codenames.

`CertPolicy` per chain-VM (default: inherit from parent L1); LP-218 reuses `CertPolicy.Mode` as the rollup's inherited cert mode.

```
type CertPolicy struct {
    Mode      CertMode     // PQ-off|PQ-fast|PQ-strict|PQ-heavy
    Variant   CertVariant  // Hybrid (with BLS) | Strict (pure PQ)
    TimeoutMs uint32       // max wait for full-mode cert; past TimeoutMs
                           //  LP-202 tier degradation kicks in
    Fallback  CertMode     // tier to settle at if Mode's legs don't
                           //  arrive in TimeoutMs; MUST be <= Mode
}
```

YAML form:

```
quasar:
  cert_policy:
    mode:       PQ-strict
    variant:    hybrid       # hybrid | strict
    timeout_ms: 1000
    fallback:   PQ-fast      # MUST be <= mode
```

2.6 Validation rules

`CertPolicy` is validated at chain genesis. A chain MUST refuse to launch with an invalid `CertPolicy`:

1. `Fallback <= Mode`. The fallback tier cannot be stronger than the target mode; the four modes are subset-ordered.
2. `Variant == Strict` requires `Mode >= PQ-fast`. Pure-PQ at `PQ-off` is the empty leg set; refused.
3. `TimeoutMs >= 2 * expected_floor_latency(Mode)`. A chain configured at a high `Mode` with a `TimeoutMs` shorter than the floor latency for that mode will routinely publish certs at a lower tier; the validator MUST reject such a configuration at genesis (no operationally-useful chain declares PQ-heavy and then never satisfies it). Floor latencies at Blackwell N=64: PQ-off ≈1 ms (min TimeoutMs 2 ms); PQ-fast ≈5 ms (min 10 ms); PQ-strict ≈50 ms (min 100 ms); PQ-heavy ≈80 ms (min 160 ms).
4. `Fallback` as a `(Mode, Variant)` pair under the chain's `Variant` must itself satisfy rule 2. A chain at `Variant=Strict` cannot fall back to `PQ-off` (the eighth-slot configuration).

3 Corona / Pulsar Parity Audit (2026-06-03)

Verdict: Pulsar-first is locked.

The Corona/Pulsar parity audit (`lux/threshold/protocols/parity/parity_test.go`, completed 2026-06-03) measured the two schemes apples-to-apples on Apple M1 Max + DGX Spark

GB10 and found Corona is 141× *slower than Pulsar at* N=64 *validators* (CPU baseline; GPU acceleration pending for both). This is not an optimization gap – it is algorithmic.

3.1 Cost-ratio table at N=64

Phase	Pulsar	Corona	Ratio
Keygen (one-time per epoch)	∼15 s	∼60 s	4×
Round 1 (per-validator)	—	∼30 ms	—
Round 2 / sign (per-validator)	∼0.4 ms	∼30 ms	∼ 75×
Aggregate (collector)	∼5 ms	∼800 ms	∼ 160×
Total signing round-trip	23 ms–28 ms	3.2 s–4.0 s	**141**×
Aggregate GPU (projected H100)	∼1 ms	∼15 ms	∼ 15×

Table 4: Pulsar vs Corona cost at N=64. CPU rows measured; GPU row projected from LP-203 datasheet extrapolation. The 141× CPU ratio is the load-bearing number.

3.2 Why the gap is algorithmic

Pulsar is a "reconstruct-on-aggregator" Shamir scheme. The aggregator briefly materializes the 32-byte master sk via Shamir reconstruction over GF(257), runs unmodified FIPS 204 ML-DSA sign once on that host, zeroizes. Output is byte-equal to vanilla ML-DSA – verifiable by any FIPS 204 verifier (circl, BoringSSL, RustCrypto). [1]

Corona is a genuine 2-round threshold lattice scheme (Corona family). Each party performs 392 NTT polynomial multiplies in Round 1 plus 256 independent Gaussian eliminations in Round 2 plus 2t PRF expansions. No party ever materializes the master sk. Different security property; different cost class.

At N=64 on the same hardware Pulsar lands at 28.5 ms total; Corona lands at 4031 ms total. The 141× gap is the price of "never trust the aggregator." Worth it for high-value chains; not worth it for the minimum-PQ tier.

3.3 Vector availability

Pulsar has committed KAT vectors (`lux/pulsar/vectors/*.json`) and cross-impl validation against circl. Corona's KAT vectors are pending; the audit harness produced the seed for the first set.

3.4 The architectural decision

The audit locks two architectural choices:

1. **PQ-fast = BLS + Pulsar.** Pulsar is cheap (28.5 ms) CPU, FIPS 204 interoperable, and has committed KAT vectors. It is the minimum-PQ tier.
2. **PQ-strict adds Corona on top.** For chains that need the never-trust-aggregator property (custody, treasury, regulated securities), the ∼ 100× latency cost buys a real security upgrade (threshold lattice without aggregator trust), not a redundant lane. Corona is therefore on top of Pulsar at PQ-strict, not in place of it.

GPU caveat. Once GPU-accelerated Corona lands (the NTT-batched aggregate kernel is in flight, per LP-203 follow-up), the absolute cost drops by an estimated 50× (4.0 s → ∼80 ms); the *ratio* to Pulsar drops by less because Pulsar's aggregate also benefits from GPU. The architectural ranking (Pulsar cheap, Corona expensive) is preserved.

This paper adds: the Corona/Pulsar parity audit verdict as a *first-class section* of the LP. The LP itself references the audit; this paper makes the verdict explicit, names the 141× ratio, and ties it to the LP-217 mode ladder locking decision.

3.5 Update from the Quasar matrix bench (2026-06-04)

The matrix bench (Section **??**) measured PQ-strict total wall-clock at N=64 on Spark = 595 ms, vs the parity audit's per-host Corona total of 4.0 s. The reconciliation: the parity-audit measurement is single-host CPU work; the matrix bench is the parallel deployment model where each validator runs its R1/R2/sign on its own host with the whole machine to itself, then the slowest validator's wall-clock dominates. With N=64 validators each doing ≈ 30 ms of Corona work in parallel, the max-validator wall-clock is bounded by the slowest single-host ceremony plus aggregation, not by $N\times$ ceremony cost.

The architectural ranking from the parity audit stands – Pulsar is intrinsically cheaper per-validator than Corona – but production deployments see the parallel wall-clock, not the per-host total. PQ-strict at N=64 inside 600 ms on Spark validates the "Corona + Pulsar at PQ-strict" choice: the cross-family lattice diversity arrives inside the round budget when measured at deployment scale.

Pulsar's per-validator sign on Spark is 175 µs, well under Corona's per-validator ceremony of ∼30 ms. Pulsar therefore composes "for free" on top of Corona in the wall-clock sense: each validator's Pulsar leg completes during the Corona R1 window. This is why PQ-strict's wall-clock is barely above PQ-fast (∼595 ms vs ∼585 ms at N=64 Spark) – the Pulsar leg overlaps the Corona R1 leg.

4 LP-202 Tier-Degradation Contract

4.1 Required-set semantics

A cert is at mode M iff every leg in M's required set has been aggregated and verified at the round digest. A cert with extra legs is still at mode M – extra legs are allowed but not required.

4.2 Tier degradation

If a required leg fails to aggregate by the round deadline, the cert is published with that leg empty. Relying parties whose policy requires mode M wait for the next height or fall back to the previous height; relying parties at the highest mode whose required set is satisfied finalize normally. Consensus liveness is preserved by the surviving legs.

4.3 Monotone observability

The cert is observable at progressively higher modes as legs arrive. A single cert struct serves all relying-party policies; the relying party applies its mode policy against the cert's observed leg set.

Worked example. Parent L1 configured PQ-heavy; the cert arrives in tiers:

4.4 Pure-PQ variants drop BLS

`strict-PQ-*` modes do not require BLS. A cert with BLS present and the required PQ legs satisfies both the classical mode (`PQ-*`) and the pure-PQ variant (`strict-PQ-*`). Operators pick the mode their relying-party policy enforces; the cert is the same.

t (ms)	Event	Cert observable at	Relying party at PQ-strict acts?
0	Round start; cert begins aggregating	n/a	no
1	BLS leg arrives	PQ-off	no
5	Pulsar leg arrives	PQ-fast	no
20	Corona leg arrives	PQ-strict	**yes**
70	Magnetar leg arrives	PQ-heavy	(already acted)

Table 5: Monotonic cert-tier observability. A single cert struct serves all relying-party policies.

4.5 ZK leg is profile-orthogonal

The optional zero-knowledge leg (execution-side certification, LP-017 v1) is not part of the mode ladder. It is attached or absent independently per chain; modes describe consensus-cert posture only.

5 Per-Mode Use Cases

PQ-off. HFT trading chains where the 50 µs preliminary cert is in the critical path of an order-matching engine; non-financial L2s whose threat model does not include store-now-decrypt-later adversaries; legacy bridge endpoints that consume only classical BLS-verifiable certs.

PQ-fast. Payment rails (the 500 µs cost is below typical ACH/card-rail end-to-end latency); gaming chains where PQ insurance is desired but 5 ms is the upper budget; DEX matching layers settling to a PQ-strict L1.

PQ-strict. Custody and treasury chains; regulated securities L1s where the cert is the audit anchor; store-of-value chains. The intra-lattice diversity (two Module-LWE parameter sets — Pulsar + Corona) hedges against a parameter- or implementation-specific break.

PQ-heavy. Cross-chain bridges (the cert is the proof a remote chain consumes); sovereign-network anchors; long-archival commitments (cert verified 30+ years out, where the hash-based Magnetar leg provides defense against any future lattice break).

6 Composition With Sibling LPs

6.1 Wire format unchanged

The QuasarCert struct shape (LP-182) is the same across all modes. The operator's mode choice is a header byte on the runtime config, NOT a wire format change. Relying parties parse the cert and apply mode policy locally.

7 Activation Marker

```
activates:      2025-12-25T16:20:00-08:00
activates-unix: 1766708400
```

The 4-mode ladder is the operator-facing surface from activation forward. v1 codenames remain valid internal identifiers and are not ripped from the codebase.

LP	Composition
LP-017 v1	Codenames (Pulsar / Aurora / Polaris) deprecated as operator-facing names; remain as INTERNAL identifiers
LP-181	Magnetar threshold SLH-DSA; PQ-heavy / strict-PQ-heavy leg
LP-182	QuasarCert wire format; unchanged across all modes
LP-202	Cert tier degradation contract; monotonic observability
LP-203	GPU-native verify; per-leg dispatch lets modes share kernels
LP-204	Network of blockchains; each chain picks one mode at genesis
LP-205	Pipelined Quasar; throughput model parameterized by mode
LP-208	DAG mempool; ordering committed at the chosen mode
LP-209	Mysticeti ordering; cert per wave at the chosen mode
LP-210	Block-STM; staged-prepare at the chosen mode
LP-211	Cross-shard atomic commit; cert at the parent-L1 mode
LP-218	ZAP P3Q PQ rollup; `CertModeFloor` per rollup

Table 6: Composition of LP-217 with sibling LPs.

8 Backwards Compatibility

None. LP-017 v1 codenames remain as INTERNAL identifiers – operator config tooling, dashboards, and docs use the v2 mode names. Code that references `polaris.go`'s `PolarisLegs` struct or `IsPolaris` / `HasHashBased` predicates is unaffected: those are internal-API surfaces, not operator-facing.

A chain genesis published before activation cannot use the v2 mode names. Pre-activation chains are on the pre-Quasar Edition Lux network and are out of scope for this LP entirely.

9 Security Considerations

The mode ladder does not change the cryptographic guarantees of any individual leg. Mode security is the floor of the strongest leg in the required set – the same property LP-017 v1 stated for cert profiles.

The v2 mode names declare posture explicitly. An operator who picks `PQ-off` is making an informed declaration that their threat model excludes store-now-decrypt-later adversaries; the v1 codename "Pulsar" did not make that declaration nearly as legibly. This is a security property of the SPEC SURFACE, not the protocol – but operator-error reduction is a real security concern at the human layer.

The pure-PQ variants drop BLS. A chain at `strict-PQ-*` cannot interoperate with any relying party that requires BLS in the cert; the relying party will see an empty BLS leg and degrade per LP-202. This is intended – pure-PQ chains are explicit about not consuming BLS.

References

[1] NIST. FIPS 204: Module-lattice-based digital signature standard (ML-DSA), 2024. NIST FIPS 204; https://csrc.nist.gov/pubs/fips/204/final.