Q-INFINITE WALLET · BUILT ON CELA™ · WHITEPAPER · PUBLIC EDITION 1.1

Q-Infinite Wallet: post-quantum authorised approvals and proof-of-action units

Monart City · 13 November 2026. Monart Studio Ventures is the registered company behind the Monart City brand and owns the CELA™ trademark.

Read this first. This paper describes research software running as a single-operator pilot. CPA, the unit described here, is not bitcoin, has no market price, is not sold, and cannot be redeemed for money or for any other asset. Nothing here is an offer of a token, a security or an investment. All testing so far was done by our own team; there has been no third-party audit.

1. Abstract
2. The problem
3. Design principles
4. Architecture
5. Cryptography and what is still classical
6. Proof of action
7. CPA token economics
8. Security model
9. Bitcoin quantum readiness
10. What it means for crypto holders today
11. Status and evidence
12. Roadmap
13. Limits and disclaimers

1 Abstract

The Q-Infinite Wallet, built on the CELA™ integrity platform, separates *deciding* to act from *executing* an action. Every action (a transfer, a mint, a policy change, a freeze, a recovery)

is first written as a precise request, called an intent, and is approved only with ML-DSA-44 signatures, the post-quantum signature standard NIST published as FIPS 204. The wallet shows each request from its own bytes and signs only when what is shown and what is signed agree. Its history is kept in signed, append-only logs that are timestamped into Bitcoin every hour.

New units of its native asset, CPA, are created only as a record of verified work: a hardware node signs each reading it produces, and a reading that verifies can mint a fixed, capped amount once. CPA is therefore a proof-of-action unit, not a mined or purchased currency.

2 The problem

Quantum risk to signatures

Almost every wallet today is authorised with elliptic-curve signatures. A large enough quantum computer running Shor's algorithm could derive a private key from its public key. Public keys that are already visible on a chain (old pay-to-public-key outputs, Taproot outputs and any address that has spent before) would be exposed first. Nobody knows when such a machine will exist, but keys, logs and signed records made today have to stay trustworthy for years.

Most losses are not cryptographic

In practice, most wallet losses come from people approving something other than what they believe they are approving (blind signing and approval phishing), or from a single stolen key that can do everything. A post-quantum signature alone fixes neither.

History that can be rewritten

Wallet software keeps logs that its own operator could later edit. Without an outside anchor, nobody can prove what the history looked like at a given time.

3 Design principles

- **Post-quantum authorisation is mandatory.** There is no fallback to a simulated or

classical approval. If the post-quantum signature cannot be made or checked, nothing happens.

- **What you see is what you sign.** The approval screen draws a request from its own bytes, and the signer only proceeds when its own drawing of the same bytes matches.
- **No single key does everything.** Keys have separate roles, and limits, time windows and delays are part of a signed policy.
- **Value from verified work, not from compute or sale.** Units are minted only for signed evidence that the network can check.
- **Everything can be checked afterwards.** An independent verifier replays every log and gives a verdict for each check.
- **Honest scope.** The wallet never claims more quantum resistance than it has (section 5).

4 Architecture

Key roles

KEY	WHAT IT CAN DO	WHERE IT LIVES
Device key	Approves day-to-day intents (transfers, mints) inside the policy's limits.	The owner's machine, in the operating system's protected key store.
Policy key	Changes the policy (limits, minting rules, time windows) and co-approves anything above the daily cap.	The owner's machine, used only for policy decisions.
Recovery key	Starts a recovery, which only takes effect after a waiting period.	Offline, exported in encrypted form to removable storage and removed from the machine.
Journal key	Signs the head of the wallet's journal so later edits are detectable. It approves nothing.	The owner's machine.

All four are ML-DSA-44 keys. Every key change (enrolment, rotation, revocation) is itself a

signed entry in the wallet's key-status log, so the set of valid keys at any moment can be replayed from the log.

Intents

An intent is a canonical, fully specified request: which wallet, which operation, which asset and amount, to whom, the maximum fee, a one-time nonce and a validity window (15 minutes by default). It is approved when enough distinct device approvals are collected (the policy sets how many, from 1 to 16). An intent rejected after it has been numbered burns its nonce, and an expired intent can no longer be approved, so no request can be replayed.

The approval console

The console runs only on the owner's computer and is never published on a website. It has four views: Home (balance, chart, send and receive), Activity (every mint and transfer, read from the verified ledger), Requests (approvals) and Security (keys, policy, freeze). Its look is shown on the wallet overview as a demo with example data.

Ledger, journal and anchoring

Approved intents are submitted to the CELA ledger, an append-only record that re-checks every authorisation and every mint when it is loaded. The journal records what the console did and is sealed with a signed head after each submission. Every hour, a salted commitment to the wallet's logs and ledger heads is included in CELA's OpenTimestamps anchor, which is confirmed in a Bitcoin block some hours later. Only the commitment is public; the opening stays private.

Independent verifier

The verifier takes the published wallet identity and replays the key-status log, the journal, the intents, the ledger and the anchors. Each check returns PASS, HOLD (for example, a Bitcoin confirmation still pending) or FAIL.

5 Cryptography and what is still classical

PURPOSE	ALGORITHM	QUANTUM POSITION
Approving intents, key changes, policy	ML-DSA-44 (FIPS 204)	Post-quantum
Journal heads	ML-DSA-44	Post-quantum
Hashing, commitments	SHA-256	Considered adequate (Grover only halves the margin)
Hardware node readings	A classical signature made on the device	Classical: a remaining dependency
Bitcoin timestamps	Bitcoin's own proof of work and block chain	Depends on Bitcoin
Moving assets on other chains (future)	That chain's own signature scheme	Classical until that chain changes

Because of the rows marked classical, the Q-Infinite Wallet is **not** quantum-resistant end to end. The accurate description is *post-quantum authorised*: the decision to act is protected by a post-quantum signature, and any external execution still uses whatever that chain requires.

6 Proof of action

A hardware node produces readings and signs each one, chaining it to the one before. The host can forward these readings but cannot create one. A mint of CPA is eligible only when all of these hold:

1. the device is on the minting rules' allow-list and was active at the mint time (not revoked);
2. the reading exists in the attested record and its device signature verifies;
3. the device's chain of readings is intact (no forged link, fork or rollback);
4. the reading's type is one the rule covers, and the amount is exactly the rule's amount;
5. the reading is newer than every reading already minted for that device and was never minted before;

6. the per-hour limit for that device and rule is not exceeded.

The ledger runs this check for every mint, and again whenever it reloads. Which rules apply is fixed by the wallet's signed policy: it names the exact rules document by its hash, and an empty value turns minting off. Rules documents are kept forever, so past mints can always be re-checked.

7 CPA token economics

PARAMETER	VALUE IN THE PILOT
Asset	CPA, a proof-of-action unit on the CELA ledger
Divisibility	Whole units only (0 decimals)
Starting supply	0. No premine, no founder or team allocation, no sale.
Issuance	Only by minting against a verified hardware reading (section 6)
Amount per qualifying reading	10 CPA
Rate limit	At most 6 mints per device per hour
Maximum issuance	60 CPA per device per hour, 1,440 CPA per device per day
Daily auto-approval cap	1,440 CPA per wallet per UTC day; more needs the policy key
Fees	Zero on the CELA ledger
Burning	A holder can burn CPA; supply = minted − burned
Changing the rules	Only by a signed policy change approved with the policy key
Market	None. Not listed, not traded, not redeemable.

Supply bound

```
supply(t) = Σ minted − Σ burned
daily issuance ≤ devices × 6 × 24 × 10 = devices × 1,440 CPA
```

Supply grows only as fast as verified work is done, and never faster than the number of enrolled devices allows. A device that stops producing valid readings stops minting; a revoked device mints nothing after its revocation time. Every unit in circulation can be traced back to the single reading that created it, with a supply audit that recomputes the total from the ledger.

What CPA is for

Today CPA is a unit of account for verified uptime: a balance says how much checked work a device has done for its owner. It is useful as an internal measure because it cannot be inflated by spending money or computing power, only by real, signed operation.

Directions we are researching, which are **not commitments**: proof-of-uptime records for networks of physical devices, credits for CELA services, and reputation for nodes that have a long verified history. Any of these would need its own design, legal review and audit before use. CPA has no price, and nothing in this paper suggests it will gain one.

8 Security model

THREAT	WHAT THE WALLET DOES
Future quantum forgery of approvals	Approvals and key changes use ML-DSA-44.
Blind signing, approval phishing	The display and the signer must agree on the same bytes, field for field.
One stolen device key	Daily caps, short validity windows, a second key above the cap, freeze.
Stolen machine	The recovery key is offline; recovery has a waiting period (48 hours by default, 1 hour to 30 days) and can be cancelled.
Replay of an old request	One-time nonces; a rejected request burns its nonce.
Fake work to mint units	Only device-signed, chained, never-used readings mint, within per-hour limits.
Rewriting history	Signed append-only logs, sealed journal heads, hourly Bitcoin timestamps.
A forged reading slipped in later	Reloads re-check every mint. A later problem with one reading cannot invalidate the rest of the history.

Out of scope today: a fully compromised owner machine that controls both the display and the keys, a hardware node whose key is extracted (its board has no secure element yet), and attacks on Bitcoin itself.

9 Bitcoin quantum readiness

The wallet includes a watch-only scanner that sorts Bitcoin outputs by how visible their public key is to a future quantum attacker:

CLASS	MEANING
Exposed public key	Pay-to-public-key and bare multisig: the key is in the output itself.
Exposed (Taproot)	The output key is on-chain.
Exposed by reuse	A hash-based address that has spent before, so its key is already public.
Hash-protected	A hash-based address never spent from: the key appears only while a spend is pending.
Unknown	Anything else, which is not judged.

The scanner never holds keys and never moves coins. Safe-spend planning, request binding for Bitcoin transactions and a migration alarm are on the roadmap.

10 What it means for crypto holders today

What carries over now

- **Know your exposure.** The exposure classes above tell a holder which coins would be at risk first. Most holders can lower their exposure today without new technology: stop reusing addresses, and move funds out of outputs whose keys are already public.
- **See what you sign.** Matching the display to the signed bytes is a pattern any wallet can adopt, and it addresses the biggest source of real losses.
- **Limit what one key can do.** Caps, windows, a second key for large actions, freeze and delayed offline recovery all reduce the damage from one stolen key.
- **Provable history.** Treasuries, funds and businesses can show what their wallet records looked like at a given time, anchored in Bitcoin.
- **A post-quantum approval step.** The decision to move value is already signed with a NIST post-quantum standard, so that step does not have to change later.

What it does not do

- It does not make bitcoin or any other existing coin quantum-safe. That needs changes to those networks.
- It does not hold or move bitcoin or other assets today; the pilot handles CPA on the CELA ledger only.
- It does not create bitcoin. New bitcoin comes only from Bitcoin mining.

11 Status and evidence

PART	STATUS (13 NOVEMBER 2026)
Keys, policy, key-status log, offline recovery	Running in the pilot
Approval console	Running on the owner's machine
Proof-of-action minting	Running; the minter checks for new readings every ten minutes
Hourly Bitcoin anchoring of the ledger	Running
Independent verifier	Built and tested
Bitcoin exposure scanner	Built, watch-only, not offered
Second-device approval	Built; switched on only with a second physical device

Evidence: 381 automated tests at the last run, covering keys, policy, intents, rendering, the ledger, minting, anchoring, the console and the verifier, plus a live run on a test wallet that minted, sent and confirmed. All of this is in-house testing.

12 Roadmap

1. A hardware node with a secure element, so the device key is hardware-protected.
2. A second physical approval device.
3. An external security audit.

4. Bitcoin: safe-spend planning, transactions bound to approved intents, post-quantum ownership commitments, and a migration alarm.
5. Post-quantum signing of the CELA history itself.
6. Only after all of the above: customer wallets.

13 Limits and disclaimers

- Research software, a single-operator pilot, holding no customer funds.
- CPA is not bitcoin, has no market price, is not sold and cannot be redeemed. Nothing here is financial advice or an offer.
- Not quantum-resistant end to end (section 5).
- Tested in-house only; no third-party audit yet.
- Parameters (amounts, limits, windows) are the pilot's and may change by signed policy.

© 2026 Built By YouTH. All rights reserved. CELA™ is a trademark of Monart Studio Ventures. Published by Monart City. Site guide · Wallet overview · Home