QP2

Quantum Proof Protocol

WHITEPAPER

Version 1.0 · June 2026

qp2.org

Abstract

Shor's algorithm — published in 1994 and executable on a sufficiently large quantum computer — can derive the private key of any ECDSA wallet from its public key in polynomial time. On March 31, 2026, Google Quantum AI published research co-authored with the Ethereum Foundation showing that breaking 256-bit elliptic curve cryptography could require fewer than 500,000 physical qubits — a 20-fold reduction from prior estimates — with individual keys crackable in minutes. Every Ethereum account that has ever sent a transaction has permanently exposed its public key on-chain. There is no existing EVM protocol that provides a live, deployable solution allowing users to protect their on-chain identity without migrating to a new address.

QP2 — Quantum Proof Protocol — fills this gap. QP2 introduces a proxy contract architecture that decouples on-chain identity from cryptographic algorithm, enabling users to retain their permanent address while freely upgrading their authentication scheme as the threat landscape evolves. The protocol functions as a persistent cryptographic security layer: users begin with the OTA Verifier — operational today, requiring no new cryptographic primitives — and seamlessly graduate to hardness-assumption-based post-quantum schemes (ML-DSA, FALCON, SLH-DSA) as they become economically viable. A modular Verifier Registry governed by $QP2 token holders ensures the protocol adopts every future NIST standard with a single governance vote, while users migrate with a single on-chain transaction. This paper describes the threat model, protocol architecture, cryptographic construction, security guarantees, token economics, governance design, and long-term relevance of QP2.

1. The Quantum Threat to Blockchain Identity

1.1 A Solved Mathematical Problem Waiting for Hardware

In 1994, mathematician Peter Shor published an algorithm that solves the discrete logarithm problem — the mathematical foundation of elliptic curve cryptography — in polynomial time on a quantum computer [1]. ECDSA, the signature scheme protecting every Ethereum wallet and the vast majority of blockchain accounts, is broken at the algorithmic level by Shor's method. This is not a theoretical weakness requiring further research. The algorithm is fully specified. What remains is the engineering challenge of building quantum hardware at sufficient scale to execute it. For nearly three decades, this threat was treated as distant. That changed on March 31, 2026.

1.2 The March 2026 Inflection Point

> *"On March 31, Google published a whitepaper showing that breaking the 256-bit elliptic curve cryptography underlying Bitcoin and Ethereum would require fewer than 500,000 physical qubits — an approximately 20-fold reduction from previous best estimates — with individual keys crackable in minutes." — Google Quantum AI, co-authored with the Ethereum Foundation and Stanford University, March 31, 2026 [2]*

Google Quantum AI, in a paper co-authored with Justin Drake of the Ethereum Foundation and Dan Boneh of Stanford University, published the most significant revision to quantum threat assessment in the history of cryptocurrency. The paper presented two optimized quantum circuits for solving the 256-bit Elliptic Curve Discrete Logarithm Problem (ECDLP-256):

- Circuit A: fewer than 1,200 logical qubits and 90 million Toffoli gates
- Circuit B: fewer than 1,450 logical qubits and 70 million Toffoli gates
- Physical qubit requirement on superconducting hardware: under 500,000 — a 20× reduction from the Litinski 2023 estimate of ~9 million [3]
- Estimated runtime: minutes on a sufficiently advanced machine of this architecture
- On-spend attack success probability against Ethereum's 12-second slot time: approximately 41% for the largest accounts [2]

This paper is not an isolated result. It is the third in a sequence of papers in twelve months that have each reduced the estimated quantum resources required to break modern cryptography by an order of magnitude. These are sequential signals from an organization that builds quantum hardware, develops quantum algorithms, and has set an internal deadline of 2029 to migrate its own infrastructure to post-quantum cryptography.

1.3 The Scale of On-Chain Exposure

The quantum threat to blockchain is not uniform. It is specifically acute for accounts whose public keys have been exposed on-chain — which occurs the moment any account sends a transaction.

Metric	Value	Source
ETH with exposed public keys	~20.5 million ETH	Google / QuSecure, Apr 2026 [5]
Top 1,000 accounts — crack time	Under 15 hours	Google Quantum AI [2]
ETH in quantum-vulnerable formats	~20.5M ETH (~17% of supply)	Google Quantum AI [2]
Estimated dollar value	~$35B (ETH = $1,750)	Google / QuSecure [5]
Physical qubits to crack ECDSA-256	<500,000 (revised)	Google Quantum AI, Mar 2026 [2]
Prior estimate (Litinski 2023)	~9 million physical qubits	Litinski, Quantum 2023 [3]
Reduction factor	~20×	Google Quantum AI [2]

Table 1.1: Current quantified exposure of cryptocurrency to quantum attack (June 2026)

1.4 The Harvest-Now-Decrypt-Later Attack

The most immediate and actionable threat is not a real-time attack — it is retroactive. Nation-state actors and well-resourced adversaries can today collect public keys from the blockchain and store them for future decryption when a cryptographically-relevant quantum computer (CRQC) becomes available.

> *"The Federal Reserve's September 2025 analysis highlights that blockchain technologies create a particularly acute variant of this exposure, as permanent public records mean exposed public keys remain vulnerable regardless of future algorithmic migrations." — Federal Reserve / arXiv, 2025 [6]*

This attack — harvest now, decrypt later (HNDL) — creates present exposure even though exploitation is deferred. Every transaction you have sent since the genesis of Ethereum has exposed your public key permanently. That exposure does not expire. It cannot be revoked. It cannot be patched by a future Ethereum hard fork. The only mitigation is to retire the exposed key — which in the traditional EOA model means retiring the address and losing everything associated with it.

1.5 The Regulatory Dimension

Quantum-safety is no longer only a technical question. It is becoming a regulatory requirement:

- NIST IR 8547 (2024): Calls for quantum-vulnerable algorithms — explicitly including ECDSA — to be deprecated after 2030 and disallowed after 2035 [7].
- NSA CNSA 2.0: Mandates all new US national security systems be quantum-safe by January 2027 [8].
- Quantum Computing Cybersecurity Preparedness Act: Requires US federal agencies to inventory vulnerable systems and report annual migration progress [8].
- Google internal deadline: Full PQC migration by 2029, announced early 2026 [2].
- Coinbase (January 2026): Formed a quantum advisory board. Panel of six cryptographers concluded a CRQC "will eventually be built" and migration must begin now [9].
- Ethereum Foundation (January 2026): Formed a dedicated Post-Quantum Security team [10].

The industry consensus has shifted. The question is no longer whether quantum computers will threaten ECDSA. It is whether the infrastructure to protect users will be ready when they do.

2. Why Existing Solutions Are Insufficient

2.1 Ethereum's Native Roadmap

The Ethereum Foundation's post-quantum roadmap centers on EIP-8141, a native account abstraction proposal targeting the Hegota hard fork planned for the second half of 2026 [11]. EIP-8141 would give individual accounts signature agility — the ability to use post-quantum signature schemes without a protocol-wide forced migration.

The limitations are significant. First, EIP-8141 requires a successful hard fork — a multi-year coordination effort across client teams, validators, exchanges, and the entire ecosystem. The Foundation's own estimates place core post-quantum infrastructure completion at approximately 2029, with full ecosystem migration extending well beyond. Second, the proposal addresses Ethereum mainnet only. The hundreds of millions of accounts on Base, Arbitrum, Polygon, BSC, and other EVM chains are not covered. Third, users migrating under EIP-8141 would receive new addresses — losing accumulated DeFi positions, ENS names, protocol allowances, and on-chain reputation.

The deeper structural limitation is the on-chain signature size problem. Post-quantum algorithms produce signatures that are 500× to 7,000× larger than ECDSA (ML-DSA-65: 3,309 bytes; FALCON-512: 666 bytes; SLH-DSA: 7,856 bytes — versus ECDSA's 65 bytes). On Ethereum mainnet today, verifying an ML-DSA-65 signature in pure Solidity costs approximately $1.68 per transaction. This is not a software problem to be optimized away. It is a fundamental consequence of post-quantum cryptography's mathematical structure. Native precompiles — expected with Hegota (H2 2026+) — reduce this by 10–50×. Until precompiles land, full PQ signatures are economical only for high-value institutional accounts. QP2's layered architecture addresses this directly: the OTA Verifier delivers quantum resistance at $0.00046 per transaction today, and the VerifierRegistry transparently upgrades to precompile-accelerated ML-DSA/FALCON when Hegota ships, with no change to the user's address or DeFi positions.

2.2 Purpose-Built Quantum-Safe L1s

QRL 2.0, QANplatform, and QDay represent purpose-built quantum-resistant blockchain networks. These projects offer genuine cryptographic safety but require users to abandon the EVM ecosystem entirely — migrating assets, rebuilding positions, and accepting new addresses. For users with years of on-chain history, DeFi collateral, and protocol integrations, this is functionally equivalent to starting over.

2.3 Existing Account Abstraction Wallets

ERC-4337 smart account wallets (Safe, Biconomy, Alchemy, ZeroDev) enable programmable validation logic but are not designed for quantum safety. The underlying authentication remains ECDSA. EIP-7702, activated in Ethereum's Pectra hard fork in May 2025 and supported by wallets including MetaMask, delegates EOA execution to a smart contract, but the delegation authorization itself is signed with the user's existing secp256k1 private key — preserving ECDSA at the root of the security model [12].

2.4 Lamport / Hash-Based Signature Wallets

Anchor Wallet (Pauli Group, 2023) implements Lamport one-time signatures on Ethereum using ERC-4337. Lamport signatures are provably quantum-resistant under the sole assumption that the underlying hash function is secure. However, the signature size problem is extreme: Lamport signatures are 10–50 kilobytes per signature. This is not a minor inconvenience — it makes Anchor viable only as a cold storage solution. At current calldata prices, a single Lamport-signed transaction on Ethereum mainnet costs $5–15 in gas. On L2s, costs are lower but still orders of magnitude above ECDSA. No modular verifier system, no cross-chain identity, and no upgrade path to more efficient algorithms exist. QP2's OTA Verifier executes at ~52,000 gas — $0.00046 on Base — making it viable for everyday DeFi interaction today, with a clear upgrade path to FALCON-512 (666-byte signatures, the most compact NIST PQ standard) as precompiles arrive.

3. The QP2 Protocol — Core Architecture

3.1 The Foundational Insight

> *Traditional EVM: identity = keypair. Quantum breaks the key → identity destroyed.QP2: identity = contract address (permanent). Authentication = swappable storage slot.Quantum breaks the current algorithm → call switchVerifier() → done.*

Every existing EVM wallet makes the same assumption that has been implicit since Ethereum's genesis in 2015: the account address is derived from the private key, making the two cryptographically inseparable. Changing the key requires changing the address, losing all accumulated history. QP2 breaks this assumption. The proxy contract address is derived via CREATE2 — a deterministic function of the factory address, a user-chosen salt, and the contract bytecode. It has no cryptographic relationship to any private key. The private key is merely a plugin in a storage slot inside the contract. Swap the plugin, keep the address.

QP2 is therefore not a single cryptographic scheme — it is a cryptographic identity infrastructure layer. The OTA Verifier is the entry primitive: deployable today, no new cryptographic assumptions, quantum-resistant by timing argument. Users graduate to SHA-256 Vault for higher security, then to ML-DSA or FALCON when Hegota precompiles make them economical for everyday use. The VerifierRegistry is the protocol's long-term value: it ensures QP2 adopts every future NIST standard without requiring users to change addresses or move funds.

3.2 System Components

Contract	Role
QP2Proxy	The user's permanent on-chain identity. Stores the active verifier address and opaque authState bytes. Executes all calls only after successful proof verification. Nonce-protected against replay.
IQP2Verifier	Interface implemented by every authentication module. A single verify() call returns (valid, nextAuthState). The proxy is completely agnostic to which algorithm is inside.
OTAVerifier	One-Time-Address verifier. Uses Ethereum's native ecrecover with rotating disposable keys. Deployable today. Gas cost comparable to standard ECDSA. Quantum-safe by timing argument — designed as the entry-point primitive; users graduate to hardness-assumption verifiers as they mature.
SHA256VaultVerifier	Two-transaction commit-reveal verifier. Adds SHA-256 hash commitment on top of OTA. Double quantum protection for high-value accounts. Deployable today.
MLDSAVerifier	ML-DSA-44 (NIST FIPS 204) post-quantum verifier. Deployed when gas economics permit or Ethereum PQ precompiles land. Provides unconditional PQ safety from NIST-vetted hardness assumptions.
VerifierRegistry	Protocol-level algorithm registry. Maps algorithm IDs to deployed, audited verifier contracts. Governed by $QP2 holders. Enables seamless adoption of future standards.
ProxyFactory	CREATE2 factory. Deploys QP2Proxy with identical address across all EVM chains for the same user inputs.
QP2Paymaster	ERC-4337 paymaster. Sponsors gas for QP2 accounts, fully decoupling gas from any ECDSA key.

Table 3.1: QP2 system components

3.3 Transaction Flow

A standard QP2 transaction proceeds as follows:

- User's wallet SDK reads the current nonce from the proxy contract.
- The SDK derives the current one-time address (addr_n) and the next one-time address (addr_{n+1}) from the user's master seed using a deterministic key derivation function.
- The SDK constructs the transaction payload and signs it with addr_n's private key using the EIP-191 signed data standard.
- The signed payload (calldata + ECDSA signature + addr_{n+1}) is submitted to the QP2 bundler.
- The QP2Proxy contract calls verifier.verify(), which runs ecrecover to confirm the signer equals addr_n, then returns nextAuthState = abi.encode(addr_{n+1}).
- If valid: nonce increments, authState updates to addr_{n+1}, calldata executes against the target contract.
- addr_n is permanently retired. Its public key was exposed for approximately 2 seconds on Base. A quantum computer requires hours to days minimum — the attack window has already closed.

3.4 Algorithm Migration

Migrating to a new algorithm is a single on-chain transaction, authorized only by the user's current authentication key:

- User generates a new keypair for the target algorithm (e.g., ML-DSA-65).
- User signs a switchVerifier() message with their current key (OTA address or existing PQ key).
- The proxy verifies the signature with the current verifier, confirms the new verifier is registered and active, then atomically updates both the verifier module and the authState.
- All subsequent transactions use the new algorithm. The proxy address is unchanged. No funds move.

The protocol operator (QP2 multisig, governed by $QP2 token holders) can register new verifiers but cannot execute this switch on behalf of any user. User sovereignty over algorithm migration is a hard protocol guarantee.

4. Cryptographic Primitives

4.1 One-Time-Address (OTA) Verifier — The Entry Security Layer

Security Basis

The OTA Verifier is the protocol's entry primitive: it delivers quantum resistance deployable today, without requiring new cryptographic hardware, precompiles, or large on-chain data. Its quantum resistance is based on a timing asymmetry between two observable quantities: the window during which a signing key's public key is visible on-chain, and the minimum time required for a quantum computer to run Shor's algorithm against that key.

Parameter	Value	Basis
Key exposure window (Base L2)	~2 seconds	Base block time, 500ms slot
Key exposure window (Ethereum mainnet)	~12 seconds	12-second slot time
Minimum Shor's runtime (2033 optimistic QC)	~9 minutes	Google Quantum AI, March 2026 [2]
Minimum Shor's runtime (conservative)	~3 hours	Webber et al., 2022 [13]
Security ratio (Base / 9-min attack)	1 : 270	Derived from above
Security ratio (Base / 3-hour attack)	1 : 5,400	Derived from above

Table 4.1: OTA timing security parameters. Exposure window measured from transaction broadcast to finality (~2 seconds on Base).

The attack the OTA scheme defeats: a quantum adversary observes addr_n's public key in the mempool and attempts to crack its private key to submit a competing transaction. By the time the crack completes (minimum 9 minutes on the most optimistic 2026 estimates), addr_n's transaction is already confirmed, addr_n is retired, and the cracked key controls nothing. The attacker would need to crack the next key (addr_{n+1}) before it signs — but addr_{n+1}'s public key is not yet exposed.

Role in the protocol lifecycle. The OTA Verifier is designed as the entry point of a layered security progression, not a permanent end state. For the 2026–2032 timeframe — where minimum quantum crack times substantially exceed any realistic mempool exposure window — OTA provides deployable protection for the entire user base at negligible gas cost. As quantum hardware matures and crack times decrease, the upgrade path is explicit and designed in: users with high-value accounts migrate to the SHA-256 Vault Verifier, then to ML-DSA or FALCON when Hegota precompiles make them economical for everyday use. This migration requires one on-chain transaction and zero funds movement. The VerifierRegistry exists precisely to make this progression seamless — the OTA Verifier is not a ceiling, it is a floor that secures users from day one while the stronger primitives mature.

Key Derivation

All one-time addresses are derived deterministically: addr_n = address(keccak256(masterSeed || n || chainId || proxy)). The user stores only the master seed. The wallet SDK derives any addr_n on demand. No per-transaction key management overhead.

4.2 SHA-256 Vault Verifier

Security Basis

The SHA-256 Vault Verifier adds a second independent quantum-resistant layer on top of the OTA scheme. It is a two-transaction commit-reveal protocol:

- Phase 1 (Commit): The user submits sha256(txnDataHash || addr_n || nonce) without revealing the transaction data. Signed by the current OTA key.
- Phase 2 (Reveal): After a minimum delay of 3 blocks (~6 seconds on Ethereum mainnet; configurable up to 64 blocks for maximum security), the user reveals the transaction data and executes. The contract verifies the commitment preimage matches.

Breaking the vault requires simultaneously: (1) cracking the OTA ECDSA key via Shor's algorithm (hours to days minimum), AND (2) reversing the SHA-256 preimage to learn the committed transaction data (2^128 operations under Grover's algorithm — computationally equivalent to the age of the universe). Both layers must be broken simultaneously. This is mathematically impossible for any foreseeable adversary.

4.3 ML-DSA and FALCON Verifiers (Phase 2)

As PQ precompiles become available on Ethereum and L2s, QP2 will register verifiers for NIST-standardized post-quantum algorithms. These provide unconditional security from NIST-vetted hardness assumptions, independent of any timing argument. The signature size problem — the primary obstacle to everyday PQ verification today — is addressed directly by precompile acceleration: ML-DSA-65 drops from ~$1.68 to ~$0.08 per transaction post-Hegota; FALCON-512 drops from ~$0.56 to ~$0.03.

Algorithm	NIST Std	PQ Security	Pub Key	Signature	Gas Today	Gas Post-Precompile
ML-DSA-44	FIPS 204	128-bit	1,312 B	2,420 B	~$1.19 (ETH)	~$0.06
ML-DSA-65	FIPS 204	192-bit	1,952 B	3,309 B	~$1.68 (ETH)	~$0.08
ML-DSA-87	FIPS 204	256-bit	2,592 B	4,627 B	~$2.31 (ETH)	~$0.12
FALCON-512	FIPS 206	128-bit	897 B	666 B	~$0.56 (ETH)	~$0.03
SLH-DSA	FIPS 205	128-bit	32 B	7,856 B	~$2.80 (ETH)	~$0.14

Table 4.2: Post-quantum algorithm parameters. Gas at June 2026 Ethereum mainnet fees (0.4 gwei, ETH = $1,750). Post-precompile assumes ~20× reduction from Hegota native PQ precompiles.

4.4 Replay and Domain Separation

Every signed message in QP2 includes the proxy address, chain ID, and current nonce. This construction prevents: (1) replay attacks; (2) cross-chain replay; (3) cross-proxy replay; (4) cross-function replay. Each function uses a unique domain tag (QP2_EXEC, QP2_SWITCH, QP2_VAULT_COMMIT, etc.).

5. Security Analysis and Threat Model

5.1 Threat Model

Adversary	Capability	QP2 Defense	Residual Risk
Classical attacker	Brute force, best classical attack on secp256k1 (2^128)	OTA: key expires in 2 seconds. Infeasible regardless.	None
Quantum attacker (near-term 2028–2030)	Shor's algorithm, insufficient qubits for secp256k1	OTA keys expire before any attack is possible at this scale.	None
Quantum attacker (2030–2033 CRQC)	Shor's on secp256k1, T_crack ≥ 9 min (Google 2026)	OTA: T_expose ≈ 2s on Base (270× margin). Vault: sha256 preimage infeasible. ML-DSA: unconditional PQ hardness.	Minimal — upgrade path is one tx
Mempool front-runner	Copy signed transaction with higher gas	nextAuthority is bound in signature. Copying fails — attacker has no addr_{n+1} private key.	Standard mempool risk identical to normal wallets
Harvest-now-decrypt-later	Collect public keys today, crack when CRQC available	OTA keys are retired after one use. Cracking a retired key yields zero value.	None after OTA adoption
Registry / governance attacker	Propose malicious verifier via governance	$QP2 Security Council veto, 7-day timelock, mandatory bond, 48h public review window. See Section 5.4.	Low — Security Council provides pre-governance safety net
Protocol multisig	Full registry control	Cannot execute user transactions. Cannot switch user verifiers without user's current key proof. Transitions to full on-chain governance.	Trust in multisig signers — time-bounded
Relayer / bundler	Observes signed UserOperation	Cannot modify payload — signature covers entire operation hash.	None
Physical device compromise	Access to master seed	Guardian recovery, Vault commit-reveal delay, hardware enclave.	Standard hardware security risk

Table 5.1: QP2 full threat model and mitigation analysis

5.2 Formal Security Properties

- Authentication integrity: QP2Proxy executes calldata if and only if a valid proof over the exact tuple (proxy, target, value, data, nonce, chainId) is verified by the current active verifier.
- Nonce replay prevention: State-changing calls increment the nonce before any external interaction. A proof valid for nonce N is rejected at nonce N+1.
- Forward secrecy: OTA keys are single-use. Retroactively cracking a retired OTA key provides zero value — the key no longer authorizes any operation.
- Migration sovereignty: switchVerifier() requires a valid proof from the current active verifier. The protocol multisig, EntryPoint, bundlers, and guardians cannot unilaterally switch a user's authentication algorithm.
- Vault commitment binding: After Phase 1 of the vault scheme, the transaction data is committed via sha256. The contract will execute only the committed transaction — even if the OTA key is subsequently compromised.
- Cross-chain identity preservation: The proxy address is deterministic via CREATE2. The same user inputs produce the same address on every EVM chain.

5.3 Known Limitations and Mitigations

- Initial deployment requires gas: The first proxy deployment requires an EOA transaction. Mitigated by QP2 Paymaster sponsored deployment and EIP-7702 delegation from existing addresses requiring only one ECDSA signature — the user's final ECDSA operation.
- Verifier implementation risk: A bug in a verifier contract could permit unauthorized execution. Mitigated by Security Council mandatory audit gate, formal verification targets for core verifiers, and user ability to switch to a different verifier at any time.
- Device / seed compromise: Loss of master seed compromises all derived OTA keys. Mitigated by guardian recovery mechanism, hardware wallet Secure Enclave storage, and Shamir secret sharing for high-security users.

5.4 Governance Security — The $QP2 Security Council

The VerifierRegistry is the protocol's most security-critical surface. A malicious or buggy verifier registered through governance could expose user funds. QP2 addresses this with a layered governance security architecture that operates independently of token voting.

Mechanism	Description
Security Council composition	7 independent cryptographers and security researchers elected by $QP2 holders. Seats renewed annually. At most 2 seats may be held by protocol-affiliated individuals.
Veto power	Any 4-of-7 Council majority can veto a verifier registration within the 48-hour public review window, without requiring a full governance vote. A veto triggers mandatory third-party re-audit before re-submission.
Emergency deprecation	A 5-of-7 Council supermajority can trigger emergency verifier deprecation with a 2-hour timelock — bypassing the 7-day governance cycle when an active exploit is confirmed.

Timelock on all registry changes	Minimum 48-hour public review window between a governance proposal passing and on-chain execution. Users with high-value accounts may switch verifiers during this window if suspicious activity is detected.
Mandatory audit gate	No verifier may be submitted to governance without a completed audit report from a Council-approved firm, published on-chain. Governance cannot waive this requirement.
Proposer bond	Minimum bond of 50,000 $QP2 required to submit a verifier registration. Bond is fully slashed if the verifier is rejected or later found to contain an exploit. Slashed bonds are burned.
Token concentration safeguard	During the first 18 months post-TGE, no single address (or cluster of addresses controlled by one entity) may cast more than 15% of votes on verifier registration proposals. Anti-Sybil enforced by Council review.

Table 5.2: $QP2 Security Council — composition and powers

The Security Council provides a professional security backstop that operates before and during governance votes. The 48-hour timelock between a successful governance vote and on-chain execution creates an explicit window for users to react: any user who observes a suspicious verifier registration during the timelock period may switch their own verifier using their existing authentication key, without waiting for Council intervention. User sovereignty is the ultimate defense layer.

The Security Council retires at the completion of Phase 4 (2027–2028), when full on-chain DAO governance activates and the protocol's security track record is established. The transition to pure token governance occurs only after independent audit of the governance mechanism itself.

6. Gas Economics and Network Costs

6.1 OTA Verifier — Deployable Today

The OTA Verifier uses Ethereum's native ecrecover precompile (~3,000 gas) plus storage operations. Total per-transaction overhead at June 2026 fee levels:

Operation	Gas Units	Cost — Base (0.005 gwei)	Cost — Mainnet (0.4 gwei)
Deploy proxy (one-time)	~295,000	$0.0026	$0.21
execute() — OTA verify	~52,000	$0.00046	$0.04
executeBatch() — 3 calls	~96,000	$0.00084	$0.07
switchVerifier()	~62,000	$0.00054	$0.04
Phase 1 commit (Vault)	~56,000	$0.00049	$0.04
Phase 2 reveal + execute (Vault)	~74,000	$0.00065	$0.05

Table 6.1: Gas costs for QP2 OTA verifier operations. Base at 0.005 gwei, ETH = $1,750.

On Base L2, the OTA Verifier operates at near-zero cost — orders of magnitude cheaper than ML-DSA Solidity implementations and directly competitive with standard ECDSA transactions. The one-time proxy deployment cost of ~$0.21 on Ethereum mainnet can be sponsored by the QP2 Paymaster, eliminating the upfront cost barrier for new users.

6.2 Future Cost Trajectory

When Ethereum's Hegota hard fork (targeted H2 2026+) introduces native PQ precompiles, the economics of ML-DSA and FALCON verifiers improve by 10–50×. QP2 verifiers will be transparently upgraded to call precompiles — users' proxy addresses and authState remain unchanged. The protocol absorbs the upgrade invisibly. This is precisely the scenario the VerifierRegistry architecture is designed for: the same address, a better algorithm, zero user action required beyond a governance vote passing.

7. Cross-Chain Identity and Deployment

7.1 One Address, Every EVM Chain

QP2 uses CREATE2 with a canonical factory address deployed via Nick's method — a keyless deployment technique that produces the same contract address on every EVM chain without relying on a specific deployer key. The same user inputs — master seed, salt, initial verifier — produce the same proxy address on Ethereum mainnet, Base, Arbitrum, Polygon, BSC, and any future EVM-compatible network. A user's QP2 identity is genuinely chain-agnostic.

7.2 Supported Networks at Launch

Chain	Status	Chain ID	Notes
Base	Phase 2	8453	Primary L2. ~$0 gas. Lowest friction for high-frequency users.
Ethereum Mainnet	Launch	1	Primary chain. Largest ecosystem. Full EIP-7702 support.
Arbitrum One	Phase 2	42161	Low gas, large DeFi ecosystem.
Polygon PoS	Phase 2	137	High throughput, low cost.
BNB Chain	Phase 2	56	Large user base in Asia.
Optimism	Phase 2	10	OP Stack, same address as Base.

Table 7.1: QP2 network deployment plan

8. The QP2 Wallet — Extension and Mobile

8.1 A Full Non-Custodial Wallet

QP2 ships as a complete wallet stack: a browser extension, iOS application, and Android application. It does not depend on MetaMask or any third-party wallet. The QP2 extension injects at both window.ethereum (for full dApp compatibility with the EIP-1193 standard) and window.qp2 (for QP2-native features). From a dApp's perspective, QP2 accounts are indistinguishable from any other EVM account.

8.2 Key Management

Master seed generation uses cryptographically secure randomness via the Web Crypto API (browser) or the OS secure random source (mobile). The master seed is encrypted with the user's password using AES-256-GCM and stored in extension local storage. On mobile, the master seed is stored in iOS Secure Enclave or Android StrongBox Keymaster — hardware-backed, biometric-protected storage that prevents extraction even on a rooted device. The master seed never leaves the device. For hardware wallet users, the master seed can be stored on a Ledger or Trezor, with the QP2 extension communicating via standard hardware wallet protocols.

8.3 Existing Address Migration via EIP-7702

For users with existing Ethereum addresses who do not want to change their address, QP2 provides a migration path via EIP-7702 (shipped in Pectra, May 2025 [12]):

- User opens QP2 extension and selects 'Migrate existing address'.
- User connects their existing wallet (MetaMask, Ledger, etc.).
- QP2 prepares a Type 4 (SET_CODE) EIP-7702 authorization signed by the existing ECDSA key — this is the last time the existing key is ever used.
- The authorization sets the EOA's code to QP2's OTA delegator contract.
- QP2 initializes the first OTA authority from the user's new QP2 master seed.

After migration, all DeFi positions, NFTs, ENS names, and protocol integrations remain at the same address. Important security note: under the EIP-7702 specification, an EVM node will accept a new SET_CODE authorization signed by the original ECDSA key if one were ever broadcast. Users should treat the original seed phrase as permanently compromised. QP2's wallet UI guides users through secure destruction of the original seed at the conclusion of migration.

9. Token Economics — $QP2

The $QP2 token has five native utilities directly tied to protocol operation. One distinguishing feature sets QP2 apart from most crypto protocols: it has a real, on-chain revenue model generating ETH fees independent of token price. A meaningful portion of that revenue is used to market-buy and burn $QP2 — creating genuine, protocol-activity-correlated buy pressure.

9.1 Five Core Utilities

* Actual QP2 amount will be decided during TGE

Governance — Vote on Verifier Registration

The VerifierRegistry is QP2's most security-critical component. It determines which cryptographic algorithms users can trust. $QP2 token holders vote on: verifier registration (new algorithms); verifier deprecation (weakened algorithms); emergency fast-track deprecation; auditor commissioning; fee parameter updates. Tokens must be staked to vote and are locked for 7 days from vote submission. New verifier submissions require a bond of 50,000 $QP2, partially slashed if the submission is rejected or the verifier is later found to contain an exploit.

Gas Subsidies — Hold $QP2 for Gasless Transactions

The QP2Paymaster sponsors gas for proxy accounts. Holding $QP2 in the proxy address determines subsidy tier:

$QP2 Balance	Gas Subsidy	Effective cost per transaction
0 QP2	0%	User pays gas normally
100 QP2	25%	75% of normal gas cost
500 QP2	50%	Half of normal gas cost
2,000 QP2	100%	Fully gasless
10,000 QP2	100% + priority	Gasless + priority bundler inclusion

Table 9.1: Gas subsidy tiers, Actual QP2 amount will be decided during TGE

Proxy Setup Fee Discount

Deploying a QP2 proxy carries a one-time protocol fee of 0.001 ETH. Users who hold $QP2 in their new proxy address receive a 50% discount, paying in $QP2 at USD-equivalent value. $QP2 collected as fees: 40% burned, 40% to paymaster treasury, 20% to development fund.

Enterprise Access — Lock $QP2 for SDK and Fast-Track

Enterprise integrations — white-label wallet SDK, custom verifier fast-track Security Council review, dedicated bundler capacity — require locking a minimum of 100,000 $QP2. Tokens are locked, not spent. They are returned in full on exit. This ensures enterprise partners have permanent stake in protocol security and their locked tokens participate in governance.

Security Council Staking

Security Council members are required to stake a minimum of 200,000 $QP2 for the duration of their term. Stake is locked; slashable if a Council member is found to have acted in bad faith (e.g., approving a verifier later proven malicious). This aligns Council incentives with token holders and creates meaningful skin-in-the-game for those holding the protocol's most sensitive veto power.

9.2 Real Revenue Model

QP2 is one of the very few crypto protocols with genuine on-chain revenue — ETH fees that accrue from protocol usage independent of token price. This revenue creates a structural buyback mechanism: 40% of all ETH protocol fees are used to market-buy $QP2 and burn it, creating permanent supply reduction correlated with adoption.

Revenue Stream	Denominated In	Buyback / Burn Mechanic
Verifier registration fee	ETH + $QP2 bond	40% of ETH fees used to market-buy $QP2 and burn; 40% to paymaster treasury; 20% to dev fund
Proxy deployment fee (0.001 ETH)	ETH	Same 40/40/20 split. At 1M proxies deployed: ~$1.75M buyback pressure
Paymaster ETH from enterprise integrators	ETH	Direct ETH inflow funds buyback programme independent of token price
Enterprise SDK lock (100k $QP2 minimum)	$QP2 locked	Tokens locked, not spent. Permanent governance stake. Enterprise exit = supply re-circulates at market price.
Verifier bond slashing	$QP2	Slashed bonds are burned, creating supply reduction correlated with protocol activity

Table 9.2: QP2 revenue streams and $QP2 buyback mechanics

To illustrate the scale: if QP2 reaches 1 million proxy deployments (a realistic target given ~20.5M ETH worth of vulnerable accounts), the deployment fee alone generates ~$1.75M in ETH buyback pressure at current prices. Enterprise SDK integrations add direct ETH inflows from institutional clients. Verifier registration fees and bond slashing create additional buy-and-burn events that scale with protocol activity. The paymaster treasury is funded by ETH inflows — not $QP2 — ensuring the gas subsidy programme remains solvent across all token price scenarios.

9.3 Supply and Distribution

Allocation	% of Supply	Vesting
Community / ecosystem	30%	4 years, linear
Protocol treasury	20%	Community governed
Team	20%	4 years, 1-year cliff
Early investors	15%	2 years, 6-month cliff
Security staking rewards	10%	Emitted over 5 years
Launch liquidity	5%	Unlocked at TGE

Table 9.3: $QP2 token distribution. All team and investor vesting enforced by on-chain contracts.

9.4 Token Supply and Initial Parameters

Total supply: 1,000,000,000 $QP2 (one billion tokens, fixed). No inflation schedule after the initial 5-year security staking reward emission. The gas subsidy tiers (Table 9.1) are calibrated at launch token price: the 2,000 QP2 fully-gasless threshold is designed to represent approximately $20–$40 USD at initial listing, making full subsidy access achievable for active users without significant capital. Enterprise lock thresholds (100,000 QP2) represent meaningful protocol stake at institutional scale. All thresholds are adjustable via $QP2 governance with a 7-day timelock.

Token Generation Event (TGE) is targeted for Q4 2026, concurrent with Phase 2 multi-chain expansion and mobile wallet launch. Tokens allocated to team and early investors are subject to on-chain vesting contracts — not off-chain agreements. The Security Council token concentration safeguard (Section 5.4) applies from TGE to prevent coordinated governance attacks during the early distribution period.

10. Long-Term Relevance and Protocol Evolution

10.1 Why QP2 Remains Relevant at Every Stage

QP2 is not a product for a single moment in the quantum timeline. It is designed to be the persistent identity layer for EVM accounts across every phase of the quantum transition.

Phase	Timeline	Quantum Status	QP2 Role
Pre-threat	2026–2028	No CRQC for ECDSA	Quantum-safe accounts for security-conscious users. Early adopters accumulate on-chain history at the address they will keep forever.
Awareness	2028–2030	100K+ qubit machines plausible	Mass-market adoption begins. Enterprises and high-value holders migrate. $QP2 buyback pressure scales with user volume.

Danger zone	2030–2033	CRQC plausible. HNDL attacks profitable	Critical infrastructure. Users upgrade OTA → ML-DSA via one transaction. Emergency migration demand.
Post-CRQC	2033+	CRQC operational. ECDSA broken.	QP2 is the standard EVM identity layer. ML-DSA via Hegota precompiles. Address unchanged.
Post-NIST Round 2	2035+	Next-gen PQ standards	New verifiers registered via governance. Users upgrade with one transaction. QP2 addresses now years old with full history intact.

Table 10.1: QP2 relevance across the quantum transition timeline

10.2 The Protocol Outlives Any Algorithm

NIST published ML-DSA and SLH-DSA in August 2024. It is already working on next-generation standards. The history of cryptography is a history of algorithms becoming obsolete and being replaced. DES was replaced by AES. SHA-1 was replaced by SHA-256. secp256k1 will eventually be superseded by something we have not yet standardized.

QP2 is the first EVM protocol designed from the ground up to survive this cycle. When the next NIST standard is published — whether in 2030 or 2040 — QP2 deploys a new verifier, the Security Council validates it, governance votes to register it, and users migrate with one transaction. Their address remains unchanged. Their DeFi positions remain intact. Their on-chain history continues accumulating.

> *A QP2 proxy address created in 2026 should still be the same user's primary on-chain identity in 2045 — regardless of how many cryptographic algorithms have been born, adopted, and retired in between.*

10.3 Ethereum Protocol Alignment

QP2 is not in competition with Ethereum's native post-quantum roadmap. It is an acceleration of it. When EIP-8141 ships with Hegota, QP2 proxies gain native AA support, improving bundler efficiency and reducing gas. When Hegota introduces PQ precompiles, QP2 verifiers are upgraded transparently. QP2 provides what Ethereum cannot: a deployable solution today, for every EVM chain, that preserves existing user identity through the entire transition.

11. Competitive Landscape

Approach	Works Today	Same Address	Quantum Proof	Gas Cost / txn
Ethereum EIP-8141	2029+	✗ New addr	2029+	~$0.20 ETH (PQ precompile)
QRL / QANplatform	✓	✗ New chain	✓	~$0.001 (own chain)
Anchor (Lamport)	Cold only	✓	✓	$0.85–$22 ETH mainnet
Coinbase Smart Wallet	✓	✓	✗ Not PQ	~$0.35 ETH (4337 userOp)
Safe / ERC-4337	✓	✓	✗ Not PQ	~$0.42 ETH (multi-sig)
EIP-7702	✓	✓	✗ BY DESIGN	~$0.15 ETH (ECDSA = unsafe)
QP2 Protocol	✓ Now	✓ Always	✓	~$0.18 ETH · $0.001 L2 (+8k gas, quantum-safe)

Table 11.1: Competitive landscape — approaches compared on deployability, address continuity, quantum resistance, and gas cost.

The differentiating combination no other protocol offers: quantum-safe today, on any EVM chain, without changing address, with the ability to upgrade algorithms as the field advances, and with a real revenue model funding a structural token buyback. Each competitor satisfies a subset of these requirements. QP2 satisfies all of them simultaneously.

A note on the post-quantum signature size problem and competitors: projects like QANplatform that support Dilithium (ML-DSA) signatures on an EVM-compatible chain still face the fundamental calldata cost constraint. The cost is not a software bug — it is a consequence of post-quantum cryptography's mathematical structure. On any EVM-compatible chain without native PQ precompiles, full PQ signature verification costs $0.56–$2.80 per transaction. QP2's OTA Verifier sidesteps this entirely for the current hardware era, and the VerifierRegistry ensures a seamless transition to precompile-accelerated PQ when the economics support everyday use.

12. Founder

QP2 is built by the person who has spent a decade constructing exactly the infrastructure it depends on — not adjacent to it, not inspired by it, but the actual protocols: account abstraction before ERC-4337 existed, non-custodial gas payment systems, isolated DeFi lending architecture, and open-source contributions to Uniswap V4. The quantum threat to EVM accounts is a protocol engineering problem. Suryansh Kumar has been solving EVM protocol engineering problems since 2017.

Suryansh Kumar • Founder & CEO, QP2 • 13+ yrs engineering • 9 yrs EVM • 2 DeFi protocols shipped

Chief Technology Officer — UniLend Finance 2020 – 2025

Architected the first isolated-pool DeFi lending protocol — every asset in its own risk-isolated market, eliminating systemic contagion across collateral types. Raised $3M+ and shipped to production. UniLend remains one of the few DeFi protocols to solve isolated lending at the contract level without relying on centralised risk parameters.

Co-Founder & CTO — MetaTransact Protocol 2019 – 2020

Built a non-custodial protocol that allowed users to pay Ethereum gas fees in any ERC-20 token — account abstraction before ERC-4337 was a standard, before the concept had a name. Required zero dApp-side changes: any existing Ethereum application worked immediately. This is the direct architectural ancestor of QP2's proxy execution model.

Open-Source Contributor — Uniswap V4 Open Source

Contributed to the core contracts of the largest decentralised exchange on Ethereum. Uniswap V4 represents the highest bar for production Solidity engineering in the ecosystem — hook architecture, singleton design, flash accounting, and sub-second settlement. Open-source contribution verifiable on GitHub.

QP2 is the natural convergence of this decade of work: a protocol that treats on-chain identity as infrastructure, not an afterthought — built by someone who has shipped the exact primitives it requires.

13. Roadmap

Security audits for Phase 1 core contracts (OTAVerifier, SHA256VaultVerifier, QP2Proxy, ProxyFactory, VerifierRegistry) are being scoped for engagement immediately following the fundraise, targeting completion prior to Q3 2026 mainnet launch. Audit firms will be announced publicly before any mainnet deployment. No contracts will be deployed to production without a completed and published audit report.

Phase	Target	Deliverables
Phase 1 — Launch	Q3 2026	OTAVerifier + SHA256VaultVerifier (security audits in progress, funded post-raise). QP2Proxy + ProxyFactory + VerifierRegistry on Ethereum mainnet. QP2Paymaster (ERC-4337). Browser extension v1 (Chrome, Brave, Firefox). TypeScript SDK (@qp2/sdk). Developer testnet open. Documentation.
Phase 2 — Expansion	Q4 2026	Hardware wallet integration (Ledger, Trezor). Multi-chain: Base, Arbitrum, Polygon, BSC, Optimism. $QP2 token launch (TGE).
Phase 3 — PQ Precompiles	2027	Mobile wallet: iOS + Android with Secure Enclave key storage. FALCONVerifier + MLDSAVerifier44 (audit-gated). Hegota precompile integration (10–50× gas reduction on ML-DSA/FALCON). SLHDSAVerifier (hash-only, maximum conservatism). Institutional SDK (multi-sig, threshold keys). Guardian recovery network launch.
Phase 4 — Maturity	2027–2028	Threshold PQ keys (2-of-3 Shamir). Full on-chain DAO governance ($QP2 Security Council retires). Hardware security module (HSM) enterprise support. NIST Round 2 integration as published.
Phase 5 — Universal Standard	2028+	Universal EVM identity layer. DeFi protocol native QP2 integration. Cross-chain identity attestations. Full ecosystem compatibility. Ongoing algorithm governance as cryptographic standards evolve.

Table 13.1: QP2 protocol roadmap

14. References

[1] Shor, P.. *Algorithms for quantum computation: discrete logarithms and factoring.* Proceedings 35th Annual Symposium on Foundations of Computer Science. IEEE, 1994. https://doi.org/10.1109/SFCS.1994.365700

[2] Google Quantum AI, Drake, J. (Ethereum Foundation), Boneh, D. (Stanford) et al.. *Securing Elliptic Curve Cryptocurrencies Against Quantum Vulnerabilities: Resource Estimates and Mitigations.* Google Research Blog, March 31, 2026. https://research.google/blog/safeguarding-cryptocurrency-by-disclosing-quantum-vulnerabilities-responsibly/

[3] Litinski, D.. *Magic State Distillation: Not as Costly as You Think.* Quantum 7, 1077, 2023. Prior best estimate: ~9M physical qubits for ECDLP-256..

[4] The Quantum Insider. *Q-Day Just Got Closer: Three Papers in Three Months Are Rewriting the Quantum Threat Timeline.* March 31, 2026. https://thequantuminsider.com/2026/03/31/q-day-just-got-closer-three-papers-in-three-months-are-rewriting-the-quantum-threat-timeline/

[5] Krauthamer, R., Buss, G. (QuSecure / Citi Institute). *Quantum Risk Assessment for Cryptocurrency Infrastructure.* Citi Institute Report / QuSecure, April 2026. https://www.qusecure.com/google-quantum-threat-bitcoin/

[6] Jain, R. et al.. *Securing Cryptography in the Age of Quantum Computing and AI: Threats, Implementations, and Strategic Response.* arXiv:2603.06969, March 2026. https://arxiv.org/html/2603.06969v1

[7] NIST. *NIST IR 8547: Transition to Post-Quantum Cryptography Standards.* National Institute of Standards and Technology, 2024. https://csrc.nist.gov/pubs/ir/8547/ipd

[8] NSA. *CNSA 2.0: Commercial National Security Algorithm Suite 2.0.* National Security Agency, 2022 (updated 2025).

[9] Coinbase. *Quantum Advisory Board Formation.* Coinbase Institutional, January 2026. Referenced in Chainalysis, April 2026.. https://www.chainalysis.com/blog/quantum-computing-crypto-security/

[10] Ethereum Foundation. *Post-Quantum Security Team Formation.* EF Blog, January 2026. https://blog.ethereum.org/2026/01/post-quantum-security-team

[11] Ethereum Foundation. *Future-Proofing Ethereum: Post-Quantum Roadmap. EIP-8141 targeting Hegota H2 2026..* ethereum.org, May 2026. https://ethereum.org/roadmap/future-proofing/quantum-resistance/

[12] Buterin, V. et al.. *EIP-7702: Set EOA Account Code.* Ethereum Improvement Proposals. Activated in Pectra hard fork, May 7, 2025.. https://eips.ethereum.org/EIPS/eip-7702

[13] Webber, M., Elfving, V., Weidt, S., Hensinger, W.K.. *The impact of hardware specifications on reaching quantum advantage in the fault tolerant regime.* AVS Quantum Science 4, 013801, 2022. https://doi.org/10.1116/5.0073075

[14] NIST. *FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA).* National Institute of Standards and Technology, August 2024. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf

[15] NIST. *FIPS 205: Stateless Hash-Based Digital Signature Standard (SLH-DSA).* National Institute of Standards and Technology, August 2024. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf

[16] Kelecsényi, A. et al.. *poqeth: Efficient post-quantum signature verification on Ethereum.* Proceedings of ASIA CCS 2025. ACM..

[17] Buterin, V. et al.. *ERC-4337: Account Abstraction Using Alt Mempool.* Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-4337

[18] Deloitte Netherlands. *Quantum computers and the Bitcoin blockchain.* Research paper, 2023. ~4M BTC in quantum-exposed address formats..

[19] Khodaiemehr, H., Bagheri, K., Feng, C.. *Navigating the quantum computing threat landscape for blockchains: A comprehensive survey.* Computer Science Review, Vol. 59, February 2026. https://doi.org/10.1016/j.cosrev.2025.100846

[20] Drake, J. (Ethereum Foundation). *Commentary on Google Quantum AI whitepaper.* The Block, March 31, 2026. https://www.theblock.co/post/395944/no-longer-drill-googles-latest-quantum-breakthrough-debate-bitcoins-security

[21] **Mosca, M. (University of Waterloo).** *Risk assessment: 1-in-7 probability of quantum-relevant threat by 2026.* Referenced in Kavout, April 2026. https://www.kavout.com/market-lens/is-quantum-computing-an-existential-threat-to-bitcoin-and-ethereum

[22] **Ledger Donjon.** *Quantum Computing's Threat to Blockchain.* February 2026. https://www.ledger.com/blog-quantum-computing-threat-to-blockchain

[23] **altFINS.** *Google Quantum AI Achieves 10x Reduction in Resources to Break Bitcoin's Cryptography.* March 2026. https://altfins.com/knowledge-base/can-quantum-computers-break-bitcoin/

QP2 Protocol · Whitepaper v1.0 · June 2026

Your address stays. Your security evolves.

qp2.org