TECHNICAL PROPOSAL

Bitcoin Post-Quantum Infrastructure

An Open, Deployable Post-Quantum Security Architecture for Bitcoin

APPLICANT	Synergy Engineering Labs
CORE TECHNOLOGY	Aegis Post-Quantum Cryptography
PROJECT CLASSIFICATION	Bitcoin Protocol Implementation
PROPOSED DELIVERABLE	Open-source quantum-resistant Bitcoin infrastructure powered by Aegis
TECHNOLOGY STATUS	Aegis is an existing, functional, patent-pending post-quantum cryptography platform
LICENSING COMMITMENT	Bitcoin-specific use of Aegis will be authorized without licensing fees or royalties

Contents

1. Executive Summary

Bitcoin's principal quantum-computing exposure is well understood: transaction authorization depends on elliptic-curve cryptography, specifically ECDSA and BIP 340 Schnorr signatures over secp256k1. A sufficiently capable cryptographically relevant quantum computer using Shor's algorithm could recover private keys from exposed public keys and forge valid Bitcoin spends. Galaxy Research has characterized the problem in two complementary layers: a **protection layer**, through which Bitcoin gains post-quantum authorization capability, and a **mitigation layer**, through which existing Bitcoin is migrated away from quantum-vulnerable authority.

Bitcoin Post-Quantum Infrastructure, or **BPQI**, provides both layers as a coherent Bitcoin-focused implementation.

The project does not require the invention of a new cryptographic primitive. NIST has already standardized ML-DSA under FIPS 204 and SLH-DSA under FIPS 205 as post-quantum digital-signature standards. NIST states that ML-DSA and SLH-DSA are intended to provide the foundation for post-quantum digital-signature deployments and can be put into use now.

The project also does not require Bitcoin to invent an entirely new extensibility architecture. Bitcoin already provides versioned witness programs and Tapscript upgrade mechanisms. BIP 342 explicitly reserves OP_SUCCESSx opcodes for future soft-fork functionality and expressly permits a future OP_SUCCESSx definition to lift the existing 520-byte stack-element restriction when larger data objects are required.

BIP 360 further provides a complementary architectural direction through Pay-to-Merkle-Root, or P2MR, a draft output type that removes Taproot's quantum-vulnerable key-path spend while retaining a script-tree architecture intended to support future post-quantum signature mechanisms. BIP 360 therefore supplies a natural structural rail into which an Aegis-backed post-quantum signature path can be integrated.

BPQI combines these existing pieces into a deployable architecture:

1. a deterministic **Aegis-BTC cryptographic verification profile**;
2. a Bitcoin-native post-quantum signature verification rule;
3. an opt-in quantum-resistant Bitcoin output and spend path;
4. ML-DSA-based transaction authorization as the initial standardized baseline;
5. optional diversified recovery using SLH-DSA;
6. forward compatibility with FN-DSA when FIPS 206 is finalized and independently validated;
7. wallet, PSBT, descriptor, custody, and hardware-signing integration;
8. migration tooling for existing Bitcoin holdings;
9. consensus-safe resource accounting and denial-of-service protection;
10. independent cryptographic auditing and formal validation of the Aegis capabilities used by Bitcoin; and
11. a completely open-source Bitcoin-facing implementation.

Aegis itself is a pre-existing independent product and remains separately maintained. The grant-funded Bitcoin implementation will use Aegis through a tightly defined cryptographic dependency interface.

As part of the project, Synergy Engineering Labs will authorize Aegis for Bitcoin-specific use without licensing fees or royalties. This authorization will apply to defined Bitcoin-related uses while preserving Synergy Engineering Labs' ownership and licensing rights for non-Bitcoin applications.

The result is not a research paper describing how Bitcoin might someday become quantum resistant. It is an implementation program designed to deliver the actual software, consensus specification, migration tooling, validation evidence, benchmarks, interoperability infrastructure, and security assurance required to give Bitcoin a practical post-quantum path.

2. Project Objective

The objective of BPQI is to deliver a complete technical foundation through which Bitcoin can transition transaction authorization from quantum-vulnerable elliptic-curve signatures to quantum-resistant digital signatures without redesigning Bitcoin's monetary system, proof-of-work mechanism, UTXO model, or transaction model.

The project focuses on the portion of Bitcoin that requires replacement under a Shor-capable quantum threat: **public-key transaction authorization**.

Bitcoin's SHA-256-based hashing infrastructure is not exposed to Shor's algorithm in the same manner as ECDSA and Schnorr signatures. Quantum search algorithms such as Grover's algorithm change the security margin of hash functions rather than producing the categorical private-key recovery attack applicable to elliptic-curve cryptography. Galaxy Research similarly identifies Bitcoin's elliptic-curve signature infrastructure as the principal quantum threat surface.

BPQI therefore takes a targeted approach:

> **Preserve what remains secure and replace what becomes vulnerable.**

The project will add quantum-resistant transaction authorization while retaining Bitcoin's existing:

- UTXO accounting model;
- proof-of-work consensus;
- block structure;
- SHA-256 hashing foundations;
- SegWit transaction architecture;
- fee market;
- script-based spending model;
- peer-to-peer consensus model; and
- monetary policy.

This materially reduces implementation complexity and minimizes unnecessary consensus change.

3. Why the Implementation Path Is Already Defined

BPQI is an engineering and assurance project, not speculative cryptographic research.

Four foundational requirements are already satisfied.

3.1 Standardized post-quantum signatures already exist

NIST finalized FIPS 204 in August 2024, defining ML-DSA as a standardized module-lattice-based post-quantum digital-signature algorithm. NIST describes ML-DSA as intended to remain secure against adversaries possessing a large-scale quantum computer.

NIST simultaneously finalized FIPS 205, defining SLH-DSA, a stateless hash-based post-quantum digital-signature algorithm derived from SPHINCS+.

The initial BPQI implementation therefore does not depend upon an experimental signature construction.

3.2 Aegis already provides the cryptographic substrate

Aegis supplies the reusable cryptographic infrastructure required around the raw algorithms, including:

- canonical algorithm identifiers;

- parameter-set management;
- deterministic signature verification;
- provider dispatch;
- typed failure behavior;
- lifecycle controls;
- secure key handling;
- cross-platform integration;
- migration semantics;
- algorithm deprecation;
- test-vector handling;
- conformance evidence;
- fail-closed operation; and
- provider-version management.

Bitcoin therefore does not need to independently recreate a complete PQC lifecycle framework around each signature implementation.

3.3 Bitcoin already contains upgrade mechanisms

Tapscript's OP_SUCCESSx mechanism was expressly designed to permit future soft-fork extensions. BIP 342 states that a future OP_SUCCESSx-derived operation can alter existing resource rules, including permitting stack elements larger than the present 520-byte limit.

This is directly relevant because standardized post-quantum public keys and signatures are larger than current Bitcoin signature objects.

3.4 Bitcoin's post-quantum output architecture is already advancing

BIP 360 proposes P2MR as a script-tree output without Taproot's elliptic-curve key-path spend. Its purpose is specifically to eliminate long-exposure vulnerability at the output level while leaving script trees available for future post-quantum authorization mechanisms. BIP 360 remains a Draft consensus proposal, but its design gives BPQI a well-aligned integration target.

BPQI will be architected so that its post-quantum verifier can operate with P2MR if BIP 360 is activated, while keeping the verifier specification separable enough to be attached to another future witness or script upgrade path if Bitcoin's consensus process selects a different structural mechanism.

This prevents BPQI from depending upon the adoption of any single unrelated BIP.

4. Threat Model

BPQI addresses the transaction-authorization consequences of a cryptographically relevant quantum computer.

The system assumes an adversary may eventually possess sufficient fault-tolerant quantum capability to derive secp256k1 private keys from corresponding public keys quickly enough to steal Bitcoin.

Two primary attack classes are addressed.

4.1 Long-exposure attacks

A public key may remain visible for a long period before an owner attempts to spend the associated output.

Relevant examples include:

- legacy P2PK outputs;

- Taproot P2TR output keys;
- reused P2PKH or P2WPKH addresses whose public keys were previously revealed;
- exposed multisig keys;
- published extended public keys;
- leaked descriptors; and
- other long-lived public-key material.

An attacker does not need to race the legitimate owner in these cases. Once sufficient quantum capability exists, the attacker can derive the classical private key from the exposed public key and construct a valid classical signature.

P2MR addresses an important portion of this problem by eliminating Taproot's exposed key-path public key.

BPQI completes the protection layer by introducing an actual post-quantum authorization mechanism.

4.2 Short-exposure attacks

Many Bitcoin output types conceal a public key behind a hash until the owner spends the coin.

Under a sufficiently capable quantum attack, the owner can still become vulnerable during the interval between:

1. broadcasting the transaction;
2. revealing the public key; and
3. receiving sufficient confirmation.

An attacker able to recover the private key during that window could construct a competing spend.

A post-quantum signature eliminates this attack for migrated outputs because disclosure of an ML-DSA or SLH-DSA public key does not expose a Shor-vulnerable elliptic-curve key.

4.3 Scope

The first BPQI release protects:

- newly created quantum-resistant outputs;
- funds migrated into those outputs;
- Bitcoin signing workflows using the new output type;
- wallet and custody key management for those outputs; and
- Bitcoin applications using the Aegis-BTC verifier.

Historical coins remain protected once their legitimate owners migrate them into quantum-resistant outputs.

Treatment of permanently abandoned, lost, inaccessible, or deliberately unmigrated classical UTXOs is a Bitcoin governance matter rather than a missing cryptographic primitive. BIP 361 already demonstrates that this policy question is being treated separately within the Bitcoin proposal process.

BPQI provides the technical infrastructure required regardless of which eventual policy Bitcoin adopts for unmigrated legacy outputs.

5. Architecture Overview

BPQI consists of six primary layers.

Layer 1 — Bitcoin Quantum-Resistant Output

A Bitcoin-native output commits to a script tree or witness program containing a post-quantum authorization condition.

The preferred integration path is compatible with P2MR.

Layer 2 — Bitcoin Post-Quantum Signature Rule

A new consensus-defined signature operation validates a supported post-quantum signature.

For discussion purposes this specification refers to the operation as:

OP_CHECKPQSIG

The final opcode number and activation mechanism would be assigned through the normal Bitcoin development process.

Layer 3 — Aegis-BTC Deterministic Verifier

Bitcoin invokes a deliberately narrow verification interface rather than the full general-purpose Aegis API.

The consensus interface provides only deterministic public verification functionality.

Layer 4 — Bitcoin Wallet and Custody Integration

Wallets receive:

- post-quantum key creation;
- descriptors;
- PSBT support;
- transaction signing;
- watch-only representations;
- backup and recovery semantics;
- fee estimation;
- hardware-wallet integration; and
- migration workflows.

Layer 5 — Migration Infrastructure

BPQI identifies vulnerable holdings and automates migration into quantum-resistant outputs.

Layer 6 — Assurance and Validation

The Aegis components relied upon by Bitcoin undergo:

- independent cryptographic review;
- implementation audit;
- malformed-input testing;
- fuzzing;
- differential validation;
- side-channel review;
- conformance testing;
- remediation;
- independent retesting; and
- applicable formal validation processes.

6. The Aegis-BTC Consensus Profile

General-purpose cryptographic agility is valuable for applications, but Bitcoin consensus requires exact deterministic behavior.

BPQI therefore defines a dedicated **Aegis-BTC Consensus Profile**.

The profile freezes all consensus-relevant behavior.

It specifies:

- accepted algorithm identifiers;
- exact algorithm revisions;
- exact parameter sets;
- public-key sizes;
- signature sizes;
- canonical encodings;
- accepted input lengths;
- sighash semantics;
- domain separation;
- verifier return semantics;
- malformed-input behavior;
- allocation limits;
- resource costs;
- provider-version requirements; and
- test vectors.

A Bitcoin node does **not** dynamically select an algorithm based on its local Aegis configuration.

Consensus decides which algorithms are valid.

Aegis performs the cryptographic operation.

This separation is essential.

> **Consensus determines:**
>
> - what algorithm is permitted;
> - what key format is valid;
> - what signature format is valid;
> - what message is verified;
> - what resource cost applies; and
> - whether an invalid result invalidates the spend.

> **Aegis determines:**
>
> - how the selected standardized cryptographic operation is executed internally.

This means Aegis does not become a source of Bitcoin consensus policy.

It becomes a cryptographic implementation provider behind a Bitcoin-defined deterministic contract.

7. Proposed Verification Interface

The consensus-facing API is intentionally minimal.

Conceptually:

```
aegis_btc_verify(
  algorithm_id,
  public_key,
  signature,
  bitcoin_signature_message
) -> VALID | INVALID
```

For consensus verification, the interface has the following properties:

- no entropy input;
- no private-key input;
- no network access;
- no dynamic algorithm negotiation;
- no external configuration affecting consensus;
- bounded memory;
- bounded parsing;
- deterministic execution;
- exact input-length requirements;
- fixed error mapping;
- fail-closed behavior.

Any unexpected provider failure is treated as verification failure rather than acceptance.

Bitcoin Core releases will pin the Aegis-BTC API revision and supported Aegis runtime release.

There will be no consensus-critical automatic library update mechanism.

8. Bitcoin-Native Encoding

BPQI will not place Aegis's general-purpose serialization format directly into Bitcoin witness data.

Bitcoin needs a compact, consensus-native representation.

The on-chain format therefore contains only information Bitcoin actually requires.

A proposed quantum-resistant leaf can conceptually commit to:

```
version
algorithm_id
public_key_commitment
OP_CHECKPQSIG
```

The witness supplies:

```
post_quantum_public_key
post_quantum_signature
script
control_block
```

The verifier:

1. validates the algorithm identifier;
2. validates exact key and signature lengths;
3. hashes the supplied public key;
4. compares it with the committed key value;
5. constructs the Bitcoin consensus signature message;
6. invokes the pinned Aegis-BTC verifier;
7. maps the deterministic verification result to Bitcoin Script success or failure.

The public-key commitment allows the large post-quantum public key to remain outside the output itself.

The key is revealed when spending, but unlike a secp256k1 public key, revealing a post-quantum public key does not create the Shor-based private-key recovery vulnerability that BPQI is designed to remove.

9. Integration with P2MR

P2MR is an especially strong structural fit for BPQI.

BIP 360 removes the Taproot internal key and commits directly to a script-tree Merkle root. The result preserves the flexibility of script-path spending without leaving a quantum-vulnerable elliptic-curve key path exposed for the lifetime of the UTXO.

A conceptual BPQI P2MR output would therefore work as follows.

Creation

1. The wallet creates an Aegis post-quantum key pair.
2. The public key is encoded canonically.
3. Bitcoin computes a commitment to the public key.
4. A post-quantum spending leaf is constructed.
5. The leaf is committed into a P2MR script tree.
6. The P2MR Merkle root becomes the output witness program.

No quantum-vulnerable internal Taproot key is required.

Spending

1. The wallet constructs the transaction.
2. Bitcoin creates the consensus signature message.
3. The wallet requests the Aegis signing provider to sign that message.
4. The witness supplies the post-quantum signature and required public-key material.
5. Bitcoin validates the script-tree proof.
6. OP_CHECKPQSIG invokes the deterministic Aegis-BTC verifier.
7. The spend succeeds only if the post-quantum signature is valid.

The result is a Bitcoin output whose authorization security no longer depends upon the hardness of the secp256k1 discrete-logarithm problem.

10. Tapscript and Large Post-Quantum Objects

Current Tapscript retains Bitcoin's 520-byte stack-element limit.

That is smaller than ML-DSA public keys and signatures.

This does not require abandoning Tapscript.

BIP 342 expressly anticipates this category of upgrade. OP_SUCCESSx processing occurs before the existing stack-element size checks, and BIP 342 specifically notes that an OP_SUCCESSx-derived operation can be defined by soft fork to support stack elements larger than 520 bytes.

BPQI therefore proposes a narrowly scoped size-rule extension.

The larger-element allowance applies only to data consumed by the post-quantum verification operation and only within explicitly bounded sizes defined by the active algorithm profile.

For example, if ML-DSA-44 is active:

- public key must be exactly 1,312 bytes;
- signature must be exactly 2,420 bytes;
- no arbitrary multi-megabyte element is accepted;
- parser allocation is bounded before verification begins.

The system therefore lifts the historical 520-byte limitation only where post-quantum cryptography requires it, without creating a general unlimited-stack mechanism.

11. Initial Cryptographic Profile

11.1 Primary baseline: ML-DSA-44

BPQI proposes **ML-DSA-44** as the initial standardized interoperability baseline.

ML-DSA is finalized under NIST FIPS 204.

ML-DSA-44 provides a practical balance between:

- standardized post-quantum security;
- implementation maturity;
- public-key size;
- signature size;
- verification performance; and
- Bitcoin block-space requirements.

Aegis's current technical assessment uses the following object sizes:

- **Public key:** 1,312 bytes
- **Signature:** 2,420 bytes

The implementation will initially pin one exact FIPS revision and parameter set.

Consensus will not permit nodes to substitute ML-DSA-65, ML-DSA-87, FN-DSA, or another algorithm under the same algorithm identifier.

11.2 Diversified recovery: SLH-DSA

SLH-DSA is finalized under NIST FIPS 205 and uses a fundamentally different hash-based security construction.

BPQI will support SLH-DSA as a conservative diversified recovery or low-frequency fallback option.

This is useful because it avoids relying exclusively on one mathematical family.

The tradeoff is signature size.

The Aegis assessment identifies SLH-DSA-128s signatures at approximately 7,856 bytes, making the scheme less attractive for routine high-volume transactions but appropriate for:

- recovery;
- deep cold storage;
- emergency spending;
- high-value vaults; and
- cryptographic diversification.

11.3 Future bandwidth optimization: FN-DSA

Falcon was selected by NIST for standardization as FN-DSA and is planned for FIPS 206, but FIPS 206 remains under development.

Aegis already maintains an FN-DSA-oriented profile.

BPQI will reserve algorithm agility for FN-DSA without making the first production Bitcoin profile dependent upon an unfinished standard.

Once FIPS 206 is finalized and the implementation has completed appropriate independent review, FN-DSA can be considered for a future activation because its compact signatures are especially attractive for Bitcoin.

This gives BPQI immediate deployability through finalized standards while preserving a clear path to more bandwidth-efficient post-quantum signatures.

12. Algorithm Agility Without Consensus Ambiguity

BPQI supports cryptographic agility without allowing nodes to disagree about consensus.

The rule is simple:

> **An algorithm exists in Aegis before it exists in Bitcoin consensus.**

Bitcoin activates algorithm identifiers explicitly.

For example:

```
0x01 = ML-DSA-44
0x02 = SLH-DSA-128s
0x03 = reserved
0x04 = reserved
```

Future algorithms remain invalid until separately activated.

A node cannot locally enable algorithm 0x03.

A new algorithm therefore requires normal Bitcoin consensus deployment rather than an Aegis configuration change.

This produces both:

- long-term crypto agility; and
- deterministic consensus.

13. Signature Message and Domain Separation

BPQI will use Bitcoin-native signature-message semantics rather than introducing a generic external transaction format.

When operating through P2MR/Tapscript, the verifier can build upon Bitcoin's existing common signature-message construction.

The signature commits to all consensus-relevant transaction data required by the selected sighash mode.

Algorithm selection is additionally committed through the spending script and leaf commitment.

This ensures that an ML-DSA signature produced for the BPQI Bitcoin context cannot be silently reinterpreted as authorization for an unrelated Aegis application.

The Aegis-BTC profile will additionally assign a dedicated Bitcoin cryptographic context identifier internally so signing APIs cannot accidentally cross domain boundaries.

14. Hybrid Transition Mode

BPQI supports an optional transitional hybrid mode requiring:

classical signature AND post-quantum signature

rather than classical OR post-quantum authorization.

This provides defense in depth during early deployment.

Before a cryptographically relevant quantum computer exists:

- the classical signature protects against an undiscovered defect in the post-quantum implementation.

After a sufficiently capable quantum computer exists:

- breaking the classical signature does not enable theft because the attacker still lacks the post-quantum signature.

Hybrid authorization can therefore serve as an early-adoption safety mechanism.

It does not need to become permanent.

Once the post-quantum implementation has accumulated sufficient production evidence and external assurance, wallets can transition to PQ-only outputs.

15. Resource Accounting and Block-Space Economics

Post-quantum signatures are larger than Schnorr signatures.

This is a cost that must be measured and priced, not a technical blocker.

The existing Aegis Bitcoin assessment models a one-input ML-DSA-44 script-path spend at approximately 986 virtual bytes under its illustrative assumptions, compared with approximately 58 virtual bytes for a P2TR key-path input.

The Bitcoin fee market already prices transaction size.

The BPQI implementation therefore treats larger witness data as an explicit economic characteristic rather than attempting to conceal it.

The protocol will retain normal witness weight accounting while adding a post-quantum verification budget that captures computational cost.

Each accepted post-quantum algorithm receives:

- a fixed verification-cost unit;
- a bounded maximum key size;
- a bounded maximum signature size;
- an invalid-input cost benchmark;
- block-level aggregate limits where required; and
- consensus test vectors for all boundary conditions.

This prevents an attacker from constructing cheap transactions that impose disproportionately expensive verification work.

16. Denial-of-Service Resistance

Consensus cryptography must remain safe under hostile inputs, not merely valid inputs.

BPQI's security testing therefore includes:

- malformed signatures;
- truncated signatures;
- overlong signatures;
- noncanonical public keys;
- incorrect algorithm identifiers;
- invalid key commitments;
- extreme script structures;
- repeated verification;
- invalid-signature worst cases;
- memory-allocation pressure;
- parser boundary conditions;
- cache behavior;
- batch-verification fallback;
- fuzzing; and
- differential-provider testing.

Verification remains individually deterministic even when optimized batching is available.

Batch verification is an optimization only.

A node running without batching must reach exactly the same consensus result.

17. Wallet Architecture

Making a quantum-resistant output available at consensus is only part of a usable Bitcoin solution.

BPQI therefore includes wallet infrastructure as a first-class deliverable.

17.1 Key generation

Wallets will generate post-quantum keys through the Aegis key-management interface.

Classical secp256k1 private keys will never be reinterpreted directly as post-quantum private keys.

The Aegis-BTC profile defines domain-separated deterministic derivation or generation semantics appropriate to the selected PQ scheme.

17.2 Backups

The project will specify:

- deterministic recovery information;
- provider metadata;
- algorithm identifiers;
- version metadata;
- key-origin data; and
- recovery validation.

A wallet backup must remain recoverable even if the wallet software or Aegis provider has evolved.

17.3 Descriptors

BPQI will define descriptor extensions capable of representing:

- PQ-only outputs;
- hybrid outputs;
- recovery paths;
- algorithm identifiers;
- key commitments;
- derivation information; and
- migration state.

17.4 PSBT

PSBT extensions will carry the information required for offline and multisystem signing, including:

- PQ public-key data;
- key origin;
- algorithm identifier;
- post-quantum signature;
- hybrid-policy information; and
- signer metadata.

The project will maintain a clear path from experimental/proprietary PSBT fields to standardized fields through the Bitcoin proposal process.

17.5 Watch-only wallets

Watch-only infrastructure will understand PQ descriptors and output commitments without requiring access to private Aegis keys.

17.6 Fee estimation

Wallets will calculate fees using the actual expected post-quantum witness size before the transaction is signed.

This prevents user-interface surprises and allows accurate coin selection.

18. Hardware Wallets and Custody

Institutional Bitcoin infrastructure cannot migrate by replacing a single signing library.

BPQI therefore provides integration specifications for:

- hardware wallets;
- HSM-backed signing;
- non-exportable key handles;
- offline signing;
- multisigner workflows;
- transaction approval systems;
- cold-storage ceremonies;
- policy engines;
- recovery systems; and
- custody audit controls.

The Aegis abstraction is particularly valuable here because a signing key can remain inside an approved provider while Bitcoin interacts with a stable Bitcoin-specific API.

An institution does not need to expose the underlying post-quantum private key to Bitcoin Core.

Bitcoin prepares the signature message.

The authorized Aegis provider signs it.

Bitcoin validates the result.

19. Quantum Migration Toolkit

BPQI will include an open-source **Bitcoin Quantum Migration Toolkit**.

The toolkit will identify the quantum-exposure status of wallet-controlled UTXOs and construct migration plans.

Outputs can be classified into categories such as:

- public key already exposed;
- Taproot public key exposed;
- previously reused address;
- public key currently hidden;

- multisig exposure;
- institutional/cold-storage designation;
- already migrated to quantum-resistant output.

Wallets can then prioritize migration.

For example:

Priority 1: already-exposed public keys
Priority 2: Taproot and reused-key holdings
Priority 3: hidden-key classical outputs
Priority 4: low-priority or operationally constrained holdings

The toolkit will support:

- exposure scanning;
- migration transaction generation;
- fee estimation;
- destination verification;
- batch migration;
- progress reporting;
- institutional inventory;
- migration evidence; and
- post-migration verification.

20. Historical UTXOs

BPQI separates two questions that are often incorrectly combined.

Question A

Can Bitcoin provide quantum-resistant authorization for coins whose owners migrate?

Yes.

That is the primary BPQI engineering deliverable.

Question B

What should Bitcoin ultimately do with coins that never migrate?

That is a Bitcoin governance and ownership-policy question.

BIP 361 demonstrates active work on the latter problem by proposing a migration and legacy-signature sunset framework.

BPQI does not need to settle that political and economic question in order to deliver the protection layer.

Instead, BPQI ensures that whichever migration policy Bitcoin ultimately chooses has:

- quantum-resistant destination outputs;
- working PQ signatures;
- wallet support;
- custody support;
- exposure scanning;
- migration tooling; and
- measurable migration state.

That makes the governance decision operationally executable.

21. Emergency Migration Support

The preferred strategy is migration before a cryptographically relevant quantum computer exists.

Nevertheless, BPQI will preserve compatibility with emergency migration mechanisms intended to protect hidden-key outputs if the threat accelerates unexpectedly.

This can include future Bitcoin commit/reveal or preauthorization mechanisms.

The BPQI architecture does not hard-code one emergency policy.

Instead, it ensures that any successfully authorized emergency spend can terminate in an Aegis-backed quantum-resistant output.

The post-quantum destination therefore remains the same regardless of the legacy escape mechanism used to reach it.

22. Lightning and Secondary Bitcoin Infrastructure

Base-layer quantum-resistant authorization is the first priority.

Lightning is a separate protocol system with additional secp256k1 dependencies in:

- funding;
- commitment transactions;
- HTLCs;
- revocation mechanisms;
- node identity;
- gossip authentication; and
- encrypted transport.

BPQI's architecture is intentionally reusable by this layer.

Aegis can provide:

- post-quantum digital signatures for Lightning identities and authorization;
- ML-KEM or hybrid key establishment for future transport profiles;
- secure key lifecycle;
- provider abstraction;
- hardware-backed signing; and
- algorithm-agility infrastructure.

The initial grant implementation will focus on Bitcoin base-layer protection and migration because that is the prerequisite for any subsequent Lightning migration.

A Lightning-specific Aegis profile can then reuse the audited cryptographic substrate rather than starting from zero.

23. Open-Source Architecture

All Bitcoin-specific software produced through the Galaxy-funded project will be open source.

This includes:

- Aegis-BTC adapter code;
- Bitcoin Core integration patches;
- script/witness implementation;
- reference implementation;
- test harness;
- test vectors;
- migration tooling;
- wallet integrations;
- descriptor support;
- PSBT support;
- benchmark tooling;
- interoperability tools;
- documentation;
- BIP drafts;
- security specifications; and
- supporting developer libraries.

The preferred licensing model for the Bitcoin-facing code will be a permissive license compatible with Bitcoin's existing open-source ecosystem.

Aegis itself is different.

Aegis is an existing, independent, patent-pending product and is not being created by the grant.

Its proprietary source code does not need to be incorporated into the open-source Bitcoin repository.

The Bitcoin project interacts with Aegis through the documented Aegis-BTC dependency interface.

24. Aegis Bitcoin Licensing Commitment

Synergy Engineering Labs is the corporate assignee of the applicable Aegis patent rights.

As part of BPQI, Synergy Engineering Labs will provide a dedicated **Bitcoin Authorized Use** license for Aegis.

The license will permit Aegis to be used for Bitcoin-related purposes without licensing fees or royalties.

The intended Bitcoin-authorized scope includes the technical activities necessary to make the infrastructure practically usable, including:

- Bitcoin node operation;
- Bitcoin transaction validation;
- Bitcoin transaction signing;
- Bitcoin wallet software;
- Bitcoin custody;
- Bitcoin hardware signing;

- Bitcoin mining infrastructure;
- Bitcoin protocol research;
- Bitcoin mainnet;
- Bitcoin testnet;
- Bitcoin signet;
- Bitcoin regtest;
- Bitcoin interoperability testing;
- Bitcoin security research; and
- directly supporting Bitcoin infrastructure covered by the final license instrument.

The authorization applies only to defined Bitcoin use.

It does not grant general-purpose rights to use Aegis for unrelated:

- blockchain networks;
- protocols;
- applications;
- products; or
- commercial deployments.

Synergy Engineering Labs retains ownership of Aegis and all non-Bitcoin licensing rights.

The effect is straightforward:

> **Bitcoin receives access to Aegis at no licensing cost for authorized Bitcoin use, while Aegis remains an independently owned product outside that scope.**

This license represents an in-kind technology contribution by Synergy Engineering Labs in addition to the engineering work funded through the grant.

25. Independent Audit and External Assurance

A post-quantum algorithm is only as trustworthy as its implementation.

A major portion of Galaxy grant funding will therefore be used to move the Bitcoin-relevant Aegis implementation through comprehensive independent assurance.

Qualified auditors will receive the source access required to inspect the underlying Aegis implementation even though the Aegis repositories themselves are not public.

The assessment scope will include:

Algorithm implementation

- ML-DSA implementation correctness;
- selected fallback algorithms;
- parameter enforcement;
- key generation;
- signing;
- verification;
- error behavior.

Memory safety

- unsafe memory operations;
- bounds handling;
- allocation behavior;
- zeroization;
- secret lifetime;
- key exposure.

Side-channel resistance

- timing behavior;
- secret-dependent branching;
- memory-access patterns;
- signing leakage;
- platform-specific behavior.

Randomness

- entropy requirements;
- random-number generation;
- failure handling;
- deterministic modes where defined;
- fault scenarios.

Consensus interface

- canonical encoding;
- malformed-input behavior;
- deterministic verification;
- provider equivalence;
- return-value mapping;
- fail-closed behavior.

Supply chain

- dependency review;
- release provenance;
- binary signing;
- artifact hashes;
- SBOM generation;
- version pinning.

Adversarial testing

- fuzzing;
- negative vectors;
- malformed keys;
- malformed signatures;
- pathological verification inputs;
- denial-of-service profiling.

Any finding affecting the Bitcoin-relevant security boundary will be remediated and independently retested.

26. Standards Conformance and Formal Validation

The project will pursue applicable standards-conformance and formal validation for the Aegis cryptographic module used by BPQI.

The initial baseline deliberately relies upon algorithms that already have finalized NIST standards.

ML-DSA is standardized by FIPS 204.

SLH-DSA is standardized by FIPS 205.

FN-DSA will remain a future option until the corresponding FIPS 206 standard is finalized and the selected implementation has completed the required independent evaluation. NIST currently lists FIPS 206 as in development.

The validation program will include, where applicable:

- known-answer tests;
- algorithm conformance;
- negative testing;
- cross-provider comparison;
- formal cryptographic-module validation preparation;
- accredited laboratory testing;
- implementation documentation;
- security-policy documentation;
- release evidence; and
- remediation of validation findings.

The purpose is to give the Bitcoin ecosystem evidence beyond vendor self-attestation.

27. Independent Differential Verification

Consensus safety requires confidence that different implementations interpret every accepted and rejected input identically.

BPQI will therefore maintain at least two verification paths during development:

1. the Aegis-BTC provider; and
2. an independently sourced/reference implementation used for differential testing.

Both implementations receive identical:

- messages;
- public keys;
- signatures;
- malformed inputs; and
- boundary inputs.

The expected result must match.

Any discrepancy becomes a release-blocking defect.

The test corpus will be public even where the proprietary Aegis implementation source is not.

This allows outside researchers to independently test the consensus behavior.

28. Public Test Infrastructure

BPQI will release machine-readable test vectors covering:

- valid ML-DSA spends;
- invalid signatures;
- incorrect keys;
- incorrect commitments;
- incorrect sighash types;
- altered transaction fields;
- wrong algorithm identifiers;
- truncated data;
- oversized data;
- malformed witness structures;
- hybrid failures;
- script-boundary cases;
- block-resource limits; and
- cross-version incompatibility.

The project will also provide reproducible transaction fixtures for:

- regtest;
- signet;
- wallet integration;
- PSBT workflows; and
- migration scenarios.

A developer should be able to determine whether an independent BPQI implementation is consensus-compatible without access to any private Aegis source code.

29. Supply-Chain and Provider Controls

A consensus-critical dependency cannot silently change behavior.

BPQI therefore applies strict provider controls.

Each approved Aegis Bitcoin runtime release will have:

- an explicit semantic version;
- cryptographic artifact digest;
- signed release metadata;
- supported-platform declaration;
- API revision;
- algorithm manifest;
- test-vector digest;
- SBOM;
- audit status;
- validation status.

Bitcoin releases pin an accepted runtime profile.

A provider update does not automatically become consensus relevant merely because a newer Aegis release exists.

This prevents ordinary software-update mechanisms from becoming implicit consensus changes.

30. Implementation Deliverables

The Galaxy-funded project will produce the following concrete deliverables.

Deliverable 1 — Aegis-BTC Normative Specification

A complete specification defining:

- algorithm registry;
- consensus verifier contract;
- key encoding;
- signature encoding;
- signature message;
- Bitcoin context binding;
- error behavior;
- resource limits;
- provider requirements;
- versioning;
- hybrid mode;
- security assumptions.

Deliverable 2 — Open-Source Bitcoin Core Reference Implementation

A working Bitcoin Core integration implementing:

- the post-quantum signature rule;
- consensus verification;
- policy validation;

- regtest support;
- signet support;
- RPC interfaces;
- transaction decoding;
- error reporting.

Deliverable 3 — Quantum-Resistant Output Implementation

A working post-quantum output/spend path compatible with the selected Bitcoin upgrade architecture.

Where P2MR is available, BPQI will integrate naturally with it.

Deliverable 4 — Wallet Implementation

Working support for:

- PQ key generation;
- addresses/output construction;
- signing;
- descriptor import/export;
- watch-only operation;
- backups;
- migration.

Deliverable 5 — PSBT Extension

A usable offline-signing workflow for Aegis-backed Bitcoin transactions.

Deliverable 6 — Migration Toolkit

Open-source exposure scanning and migration planning software.

Deliverable 7 — Benchmark Suite

Cross-platform measurement of:

- key generation;
- signing;
- verification;
- invalid verification;
- memory;
- transaction weight;
- fee impact;
- block verification;
- batch behavior.

Deliverable 8 — Consensus Test Suite

Public positive, negative, malformed, fuzz-derived, and differential vectors.

Deliverable 9 — Independent Security Audit

Third-party cryptographic and implementation assessment.

Deliverable 10 — Remediation and Retest

All material audit findings affecting the project scope resolved and independently retested.

Deliverable 11 — Standards/Validation Evidence

Applicable conformance and formal validation artifacts for the selected Aegis cryptographic boundary.

Deliverable 12 — BIP-Quality Technical Specification

A publication-ready Bitcoin protocol proposal describing the consensus integration and its rationale.

Deliverable 13 — Signet Demonstration

A publicly reproducible Bitcoin environment in which users can:

- create PQ outputs;
- fund them;
- sign transactions;
- broadcast them;
- validate them;
- migrate classical test coins; and
- independently verify the results.
-

31. Implementation Plan

The project is divided into parallel workstreams rather than a single sequential research effort.

This allows cryptographic assurance, Bitcoin engineering, tooling, and documentation to progress simultaneously.

Phase 1 — Specification Freeze

Target: Months 1–2

Deliver:

- Aegis-BTC v1 profile;
- initial ML-DSA-44 profile;
- API contract;
- encoding rules;
- signature-message rules;
- resource model;
- open-source repository;
- test-vector format;
- Bitcoin Authorized Use licensing framework.

Exit condition: every consensus-relevant cryptographic behavior is machine-testable.

Phase 2 — Regtest Implementation

Target: Months 2–4

Deliver:

- Bitcoin Core integration;
- PQ verifier;
- output/spend path;
- RPC support;
- wallet prototype;
- descriptor support;
- PSBT prototype;

- benchmark harness.

Exit condition: complete quantum-resistant transactions execute deterministically on regtest.

Phase 3 — Independent Audit and Validation

Target: Months 2–7, in parallel

Deliver:

- auditor source access;
- cryptographic review;
- implementation review;
- side-channel analysis;
- conformance testing;
- remediation;
- independent retest;
- validation package preparation.

Exit condition: no unresolved critical or high-severity finding affecting the Bitcoin security boundary.

Phase 4 — Migration and Custody Tooling

Target: Months 4–7

Deliver:

- UTXO exposure classifier;
- migration planner;
- wallet migration flow;
- batch migration;
- institutional workflow specification;
- hardware-signing API;
- custody guidance.

Exit condition: classical regtest holdings can be discovered, classified, and migrated end to end into PQ outputs.

Phase 5 — Signet and Interoperability

Target: Months 6–9

Deliver:

- signet deployment;
- independent test implementation;
- public vectors;
- fuzzing corpus;
- long-running compatibility tests;
- performance results;
- reference client documentation.

Exit condition: independently implemented verifiers produce identical consensus results across the published test suite.

Phase 6 — BIP and Release Candidate

Target: Months 9–12

Deliver:

- BIP-quality specification;

- reference implementation;
- final audit evidence;
- validation evidence;
- public technical documentation;
- release candidate;
- deployment guidance;
- migration guidance.

Exit condition: the technical package is complete for Bitcoin developer review and potential consensus deployment.

Mainnet activation itself is deliberately not used as a grant success criterion because activation is controlled by Bitcoin's decentralized consensus process rather than by any individual developer, company, or grant recipient.

The project instead delivers the complete engineering package required for that decision.

32. Acceptance Criteria

BPQI will be considered technically complete when the following conditions are satisfied.

Cryptographic correctness

- ML-DSA verification matches the selected FIPS profile.
- All known-answer tests pass.
- Independent verification agrees with Aegis.
- Invalid signatures are reliably rejected.

Consensus determinism

- identical transactions produce identical results across supported implementations;
- no local provider option changes consensus;
- no entropy is used during verification;
- all error conditions fail closed.

Resource safety

- memory is bounded;
- key and signature lengths are bounded;
- invalid-input performance is measured;
- consensus verification costs are defined;
- fuzz testing reveals no unbounded parser behavior.

Bitcoin interoperability

- regtest transactions work end to end;
- signet transactions work end to end;
- wallet creation works;
- signing works;
- descriptor workflows work;
- PSBT workflows work;
- migration transactions work.

Assurance

- independent audit completed;
- material findings remediated;
- retest completed;
- conformance evidence produced;
- applicable validation work completed or formally progressing through the external validation process.

Public deliverables

- Bitcoin-specific code published;
- technical specification published;
- vectors published;
- benchmarks published;
- migration tooling published;
- documentation published.

33. What Galaxy Funding Enables

The core Aegis technology already exists.

Galaxy funding is therefore not required to invent the underlying post-quantum cryptography.

Funding primarily converts that existing capability into independently assured, Bitcoin-specific infrastructure.

The principal funded activities are:

1. independent cryptographic and security audit;
2. formal validation and standards-conformance work;
3. remediation and independent retesting;
4. Bitcoin Core integration;
5. consensus specification;
6. wallet and PSBT engineering;
7. migration tooling;
8. signet infrastructure;
9. performance and denial-of-service testing;
10. interoperability testing;
11. BIP preparation;
12. public documentation.

Synergy Engineering Labs additionally contributes the Bitcoin-specific use of Aegis without licensing fees or royalties.

Galaxy's capital is therefore leveraged against pre-existing technology and intellectual property rather than being spent recreating a PQC platform from the beginning.

34. Strategic Value to Bitcoin

BPQI gives Bitcoin several capabilities at once.

Immediate standardized post-quantum authorization

The initial implementation can use an already-finalized NIST digital-signature standard rather than waiting for a future algorithm.

Cryptographic agility

Bitcoin receives a controlled mechanism for future post-quantum algorithms without allowing runtime configuration to alter consensus.

Quantum-resistant destination outputs

Wallets gain somewhere secure to migrate value before a quantum emergency.

Institutional migration support

Custodians can begin engineering and rehearsing migration while classical cryptography remains secure.

Independent security evidence

Aegis's Bitcoin-relevant implementation is subjected to external cryptographic review rather than relying solely on internal claims.

Open Bitcoin implementation

The Bitcoin-facing infrastructure remains inspectable, testable, forkable, and independently implementable.

Zero Aegis licensing cost for Bitcoin

Bitcoin users and infrastructure providers within the authorized Bitcoin scope do not need to pay licensing fees or royalties to use Aegis.

Algorithm diversity

Bitcoin can begin with ML-DSA while retaining SLH-DSA as a conservative diversified option and FN-DSA as a potential future bandwidth optimization after standardization and assurance.

35. Relationship to Existing Bitcoin Quantum Work

BPQI is designed to complement rather than duplicate existing Bitcoin quantum-readiness efforts.

BIP 360

P2MR provides a strong structural output format by removing the quantum-vulnerable Taproot key path.

BPQI supplies the post-quantum signature implementation that such an output ultimately needs for short-exposure protection.

BIP 361

BIP 361 addresses migration policy and the eventual treatment of legacy quantum-vulnerable signature types.

BPQI supplies the quantum-resistant destination, implementation, and migration tooling necessary to make such a transition executable.

Emergency commit/reveal concepts

Emergency schemes protect selected classical outputs during an unexpected quantum event.

BPQI provides the post-quantum destination into which recovered or protected coins can be moved.

The project therefore fills a distinct gap:

> **Bitcoin needs an actual audited post-quantum authorization implementation, not only an output structure or migration policy.**

Galaxy itself identifies the development and integration of post-quantum signature schemes into Bitcoin, implementation and review of quantum-resistant transaction proposals, migration tooling, and formal security audits as priority areas for its Bitcoin Quantum Readiness Initiative. BPQI directly spans those priorities.

36. Why Aegis

The primary advantage of Aegis is not merely that it can calculate a post-quantum signature.

Open implementations of individual algorithms already exist.

Bitcoin needs substantially more than a signing function.

A production transition requires:

- stable algorithm identification;

- provider lifecycle;
- consensus-safe verification;
- key lifecycle;
- secure storage;
- multiple implementation targets;
- hardware support;
- migration controls;
- deprecation;
- error semantics;
- conformance;
- auditability;
- evidence;
- version control;
- fail-closed behavior.

Aegis was designed as a reusable post-quantum cryptography platform around these concerns.

That is what allows BPQI to begin at the Bitcoin integration layer rather than first building an entire cryptographic product stack.

Aegis also has practical deployment experience. It is already used as the post-quantum cryptography platform for Synergy Network, providing a real-world production example of Aegis operating as cryptographic infrastructure inside a blockchain environment.

That production experience reduces execution risk for the Bitcoin project while remaining independent of the BPQI architecture itself.

37. Engineering Position

BPQI does not depend on a speculative breakthrough.

It does not require:

- inventing a new post-quantum signature scheme;
- replacing Bitcoin proof of work;
- replacing SHA-256;
- replacing the UTXO model;
- creating an entirely new transaction system;
- creating a new blockchain;
- creating a trusted migration authority; or
- waiting for an unknown cryptographic standard before beginning.

The core requirements already exist:

standardized algorithms + existing Bitcoin upgrade mechanisms + an existing PQC platform + a defined migration architecture.

The project is therefore a bounded implementation and assurance program.

The remaining work is concrete:

- specify;
- integrate;
- test;

- audit;
- validate;
- benchmark;
- publish;
- deploy to Bitcoin test environments;
- prepare for consensus review.

That is precisely the class of work developer grant funding can accelerate effectively.

38. Expected Result

At completion, the Bitcoin ecosystem will have a working, independently reviewed post-quantum infrastructure stack capable of demonstrating the full lifecycle of a quantum-resistant Bitcoin transaction:

```
Generate PQ key
    ↓
Construct quantum-resistant Bitcoin output
    ↓
Receive BTC
    ↓
Construct spend
    ↓
Generate Bitcoin signature message
    ↓
Sign through Aegis
    ↓
Broadcast transaction
    ↓
Bitcoin consensus invokes Aegis-BTC verifier
    ↓
Post-quantum signature validates
    ↓
Transaction confirms
```

Existing Bitcoin can additionally be migrated through:

```
Discover classical UTXOs
    ↓
Classify quantum exposure
    ↓
Prioritize migration
    ↓
Generate BPQI destination
    ↓
Construct migration transaction
    ↓
Sign using current authority
    ↓
Move BTC into PQ-protected output
    ↓
Future spends use PQ authorization
```

This establishes the fundamental capability Bitcoin requires before a quantum emergency exists:

> **a functioning, testable, independently reviewed place to move Bitcoin where spending authority no longer depends upon quantum-vulnerable elliptic-curve signatures.**

39. Conclusion

Bitcoin's quantum-computing problem is solvable with technology that exists today.

The essential engineering requirement is to give Bitcoin a consensus-visible post-quantum authorization path and a practical way for users and institutions to migrate into it.

Bitcoin Post-Quantum Infrastructure provides that path.

Aegis contributes the post-quantum cryptographic platform.

Bitcoin contributes mature consensus, transaction, witness, and script upgrade mechanisms.

NIST provides finalized post-quantum signature standards.

Existing Bitcoin proposals provide complementary output and migration architecture.

BPQI integrates these components into one coherent implementation.

The project will deliver open-source Bitcoin infrastructure, a deterministic Aegis-BTC verifier profile, standardized post-quantum transaction authorization, wallet and custody support, migration tooling, independent cryptographic audit, formal validation work, performance evidence, public test vectors, interoperability infrastructure, and a BIP-quality consensus specification.

Synergy Engineering Labs will additionally authorize Aegis for defined Bitcoin use without licensing fees or royalties, giving the Bitcoin ecosystem access to the underlying technology while preserving Aegis's independent ownership and non-Bitcoin commercial rights.

The objective is not merely to demonstrate that a post-quantum signature can be verified.

The objective is to provide Bitcoin with a complete, implementation-ready quantum-resistance layer that developers, wallet providers, custodians, miners, security researchers, and the wider Bitcoin community can test today and deploy through Bitcoin's established consensus process.

Aegis provides the cryptography.
BPQI provides the Bitcoin infrastructure.
Together, they provide a practical path to quantum-resistant Bitcoin.