PLANCKCHRON FLAGSHIP TECHNICAL SERIES

SECURITY ARCHITECTURE / VERSION 1.0 / 09 AUGUST 2026

Post-Quantum Transition

A Crypto-Agile Migration Architecture for Long-Lived Systems

ABSTRACT

The publication of NIST's first post-quantum cryptography standards shifts migration from algorithm selection to enterprise execution. This paper proposes a crypto-agile transition architecture built around discovery, dependency mapping, data-lifetime analysis, policy-based algorithm selection, provider abstraction, hybrid testing where appropriate, observability, and rollback. It distinguishes key establishment from digital signatures and treats migration as a portfolio of protocol, product, certificate, hardware, vendor, and data-retention changes. The framework uses NIST FIPS 203, 204, and 205 as primary standards references, together with public migration guidance from NIST, CISA, and NSA. It is educational guidance. No security assessment, implementation guarantee, legal opinion, certification, or representation that Planckchron operates a post-quantum security product is claimed.

PLANCKCHRON RESEARCH

Self-published technical paper / not peer-reviewed / open access

planckchron.com/publications/post-quantum-transition-crypto-agility

TECHNICAL NOTE

Executive brief

The publication of NIST's first post-quantum cryptography standards shifts migration from algorithm selection to enterprise execution. This paper proposes a crypto-agile transition architecture built around discovery, dependency mapping, data-lifetime analysis, policy-based algorithm selection, provider abstraction, hybrid testing where appropriate, observability, and rollback. It distinguishes key establishment from digital signatures and treats migration as a portfolio of protocol, product, certificate, hardware, vendor, and data-retention changes. The framework uses NIST FIPS 203, 204, and 205 as primary standards references, together with public migration guidance from NIST, CISA, and NSA. It is educational guidance. No security assessment, implementation guarantee, legal opinion, certification, or representation that Planckchron operates a post-quantum security product is claimed.

EVIDENCE BOUNDARY

Educational architecture and readiness framework only. No security assessment, implementation guarantee, certification, legal opinion, or commercially available Planckchron security product is claimed.

Four operating conclusions

- Cryptographic inventory is a dependency graph, not a spreadsheet of algorithms.
- Data lifetime and migration friction can make a system urgent before a cryptographically relevant quantum computer exists.
- Crypto-agility requires controlled change, interoperability testing, telemetry, and rollback - not only an abstraction interface.
- Algorithm and parameter choices must follow current standards, protocol profiles, implementation guidance, and system context.

STATUS	**Practitioner framework**
PUBLISHER	Planckchron Inc.
CITATION	*Planckchron Research (2026). Post-Quantum Transition: A Crypto-Agile Migration Architecture for Long-Lived Systems. Version 1.0.*

TECHNICAL NOTE

Reference architecture

The architecture below is a proposed control and evidence flow. It is a design artifact, not a representation of deployed capability.

01

DISCOVER

Find cryptographic use in code, protocols, certificates, devices, data stores, vendors, and operational procedures.

02

MAP

Connect each use to data lifetime, business process, system owner, dependency, replaceability, and failure impact.

03

PRIORITIZE

Rank by confidentiality horizon, exposure, criticality, migration friction, and external dependency.

04

ABSTRACT

Move policy and algorithm selection behind versioned provider and protocol interfaces where feasible.

05

TEST

Validate interoperability, size, performance, failure, downgrade, rollback, and operational observability.

06

MIGRATE

Deploy in evidence-gated waves, retire vulnerable paths, and retain a complete decision and exception record.

TECHNICAL NOTE

1. Why transition is an architecture problem

Post-quantum migration affects more than cryptographic libraries. Public-key algorithms appear in transport security, virtual private networks, code signing, software update, certificates, identity, email, document workflows, hardware roots of trust, embedded devices, cloud services, vendor products, and long-lived data protection. Each dependency changes on a different schedule.

The relevant risk is not only when a future computer may break an algorithm. It is also whether sensitive data can be collected now and whether the organization can complete migration before current protections are no longer acceptable. Systems with long confidentiality lifetimes or slow replacement cycles may require earlier action.

TECHNICAL NOTE

2. Standards baseline

NIST FIPS 203 specifies ML-KEM for key establishment. FIPS 204 specifies ML-DSA for digital signatures. FIPS 205 specifies SLH-DSA, a stateless hash-based signature method. These functions are not interchangeable. System design must distinguish establishing shared secrets from authenticating software, identities, documents, and protocol messages.

Implementers should follow current standards pages, errata, protocol profiles, validation requirements, and later guidance rather than copying parameters from a secondary summary. NIST SP 800-227 provides recommendations for key-encapsulation mechanisms. Transition timelines and permitted combinations differ by environment, especially for national-security systems.

TECHNICAL NOTE

3. Cryptographic dependency graph

An inventory record should include system and business owner, cryptographic function, algorithm and parameter, library or provider, protocol and version, certificate or key lifecycle, hardware dependency, data classification and confidentiality lifetime, external counterparty, update mechanism, test environment, rollback path, and retirement condition.

The graph matters because one visible service may depend on a load balancer, identity provider, hardware module, certificate authority, client library, managed database, device firmware, and third-party integration. A migration plan that does not include those edges can report progress while the vulnerable path remains.

TECHNICAL NOTE

4. Prioritization model

Priority should be based on data lifetime, collection exposure, business and safety criticality, migration friction, protocol and vendor readiness, replacement cycle, and consequence of interoperability failure. The result should be an explainable tier, not a fictional precision score.

High-priority candidates commonly include long-lived confidential archives, identity and signing roots, software and firmware update chains, systems with long procurement cycles, widely distributed devices, and dependencies controlled by vendors with uncertain roadmaps. Each organization needs its own threat, legal, contractual, and operational analysis.

TECHNICAL NOTE

5. Crypto-agile reference architecture

A crypto policy service defines approved functions, algorithms, parameters, transition states, and exceptions by system class. Applications call versioned provider interfaces rather than embedding algorithm decisions throughout business logic where feasible. Provider adapters connect to validated libraries, services, or hardware. A certificate and key lifecycle service handles issuance, rotation, inventory, revocation, and audit. Protocol adapters negotiate approved combinations while preventing unsafe downgrade.

Telemetry records which algorithms and parameters are actually used, not only configured. A migration controller supports pilot, dual-read or dual-verify patterns where appropriate, staged rollout, error monitoring, and rollback. An exception registry records systems that cannot yet migrate, the reason, compensating controls, owner, and expiration.

TECHNICAL NOTE

6. Hybrid transition and downgrade risk

Some migration paths combine classical and post-quantum mechanisms during transition. A hybrid design may reduce dependence on one new implementation, but it also adds complexity, message size, negotiation paths, and failure modes. It is not automatically safer.

The security property, combiner construction, protocol binding, downgrade resistance, certificate behavior, and interoperability must be explicit. Teams should use standards and protocol-specific guidance, not invent combiners. A failed hybrid handshake must not silently fall back to an unacceptable mode.

TECHNICAL NOTE

7. Test and evidence program

Testing should cover known-answer and conformance cases, interoperability across approved implementations, key and certificate lifecycle, message and certificate size, latency and throughput under representative load, memory and storage on constrained devices, failure injection, downgrade attempts, malformed inputs, side-channel review where relevant, update and rollback, backup and recovery, logging, and operational support procedures.

Every result should identify implementation, version, build, configuration, platform, test data, protocol path, and limitations. Passing a library test does not validate the complete system.

TECHNICAL NOTE

8. Phased migration plan

Phase zero establishes ownership and policy. Phase one creates the cryptographic dependency graph. Phase two remediates visibility gaps and introduces provider abstraction in systems already being changed. Phase three pilots selected low-blast-radius paths. Phase four migrates prioritized production classes with rollback and monitoring. Phase five retires vulnerable algorithms, removes fallback, closes exceptions, and verifies the observed environment.

The plan should be reviewed when standards, errata, vendor roadmaps, protocol profiles, cryptanalytic knowledge, or system context changes. Completion is not a one-time date; crypto-agility becomes an ongoing capability.

TECHNICAL NOTE

References and source notes

References are provided for the external standards, public guidance, and primary research that informed this paper. A citation does not imply endorsement, partnership, certification, or validation of Planckchron capability.

01 **NIST FIPS 203**

Module-Lattice-Based Key-Encapsulation Mechanism Standard

https://csrc.nist.gov/pubs/fips/203/final
Primary public standard; implementers should monitor the official errata and revision record.

02 **NIST FIPS 204**

Module-Lattice-Based Digital Signature Standard

https://csrc.nist.gov/pubs/fips/204/final
Primary public digital-signature standard.

03 **NIST FIPS 205**

Stateless Hash-Based Digital Signature Standard

https://csrc.nist.gov/pubs/fips/205/final
Primary public digital-signature standard.

04 **NIST SP 800-227**

Recommendations for Key-Encapsulation Mechanisms

https://csrc.nist.gov/pubs/sp/800/227/final
External public implementation guidance finalized in 2025.

05 **NIST NCCoE**

Migration to Post-Quantum Cryptography

https://www.nccoe.nist.gov/applied-cryptography/migration-to-pqc
External public migration project and implementation reference.

06 **CISA, NSA, and NIST**

Quantum-Readiness: Migration to Post-Quantum Cryptography

https://www.cisa.gov/resources-tools/resources/quantum-readiness-migration-post-quantum-cryptography
External public readiness roadmap; system-specific requirements still apply.

07 **NSA**

Post-Quantum Cybersecurity Resources

https://www.nsa.gov/Cybersecurity/Post-Quantum-Cybersecurity-Resources/
External public national-security guidance; no Planckchron relationship or approval is implied.

PUBLICATION DISCLOSURE

These self-published technical papers are not peer-reviewed, audited, or independent validation. They describe research methods, proposed architectures, and evaluation plans. Product capability exists only where a dated evidence record expressly reports a result. Educational architecture and readiness framework only. No security assessment, implementation guarantee, certification, legal opinion, or commercially available Planckchron security product is claimed.