

QVTM — Trust Continuity for the Post-Quantum Era

Version: 1.1 **Status:** Released editorial revision **Authors:** Fernando Guerrero and Patricio Guerrero
Publication date: July 17, 2026 **Copyright:** © 2026

Abstract

Digital trust is entering an era in which trustworthy relationships must survive repeated cryptographic migration, infrastructure modernization, identity recovery, governance change, and intermittent or disconnected operation. The challenge is not only whether trust exists at a particular moment; it is whether confidentiality, integrity, identity, authorization, privacy protection, and cryptographic assurance remain meaningful as the systems supporting them change. The transition toward post-quantum cryptography makes this challenge more visible, especially because harvest-now, decrypt-later adversaries turn long-term confidentiality into an architectural problem rather than a purely algorithmic one. Long-lived digital systems increasingly require trust relationships that remain preserved, verifiable, and operational across successive technology generations.

Recent advances in post-quantum cryptography [1,11,12], Zero Trust Architecture [2], transparency systems [3], decentralized identity and provenance [5,14], secure messaging [4], trusted computing and remote attestation [6,7], and confidential computing [8] have strengthened the establishment, protection, and verification of digital trust. These technologies remain indispensable. They primarily answer whether trust can be established, evaluated, protected, or verified at particular points in time. Less attention has been devoted to preserving trust relationships themselves across decades of cryptographic, operational, organizational, and governance change.

This paper proposes **Trust Continuity** as a first-class architectural property for long-lived digital systems and as the enabling property for a new class of **Continuous Trust Systems**. Trust Continuity describes the independently verifiable preservation of trust relationships throughout technological, organizational, and cryptographic evolution. The paper introduces the **Multi-Domain Continuity Model (MDCM)** as a conceptual model organized around four continuity domains: **Cryptographic Continuity**, **Identity Continuity**, **System Continuity**, and **Service Continuity**. Each domain produces **Continuity Evidence**, which can be correlated to derive continuity paths from which Trust Continuity can be evaluated.

The paper further presents **QVTM (Quantum Verifiable Trust Mesh)** as a reference protocol architecture and ecosystem intentionally designed around Trust Continuity. Its protocol core manages transportable **Continuity Evidence Packages (CEPs)**, while **Graph Puzzle** derives

continuity paths from available evidence. The reference ecosystem extends that core through a control-plane SaaS, passwordless login, offline vaulting, SDKs, integration APIs, and secure messaging profiles, each demonstrating a distinct continuity capability. The contribution is twofold: Trust Continuity is proposed as a new architectural dimension for long-lived digital trust, and QVTM is presented as a concrete ecosystem for investigating, implementing, integrating, and deploying that dimension. The companion White Paper, Protocol Specification, SDK Reference, and Formal Foundations develop the architecture, normative behavior, implementation surface, and mathematical basis introduced here.

Position Statement

Digital trust has traditionally been studied through the establishment, evaluation, and protection of trustworthy states. QVTM extends that framing toward Continuous Trust Systems: systems whose trust properties must remain meaningful while their cryptography, identity, infrastructure, governance, and data-protection boundaries evolve.

Authentication establishes identities. Public Key Infrastructures (PKIs) establish cryptographic trust anchors. Zero Trust Architecture continuously evaluates authorization decisions [2]. Certificate Transparency improves accountability for certificate issuance [3]. Messaging Layer Security protects group communications [4]. Decentralized Identity and Verifiable Credentials extend portable identity models [5]. Trusted computing and remote attestation provide evidence regarding platform integrity [6,7], while confidential computing strengthens protection for execution environments [8]. Privacy-preserving storage, secure messaging, and offline vaulting protect sensitive data against unauthorized disclosure, while post-quantum cryptography seeks to preserve confidentiality and authenticity against quantum-capable adversaries through cryptographic survivability and algorithm agility [1,11,12].

Collectively, these technologies answer an essential question:

Can a digital entity be trusted now?

This paper argues that long-lived digital systems introduce an additional architectural question:

How can trust relationships, protected data, cryptographic assurances, and integrity guarantees remain verifiable and meaningful while the systems supporting them undergo cryptographic migration, infrastructure modernization, organizational transition, and governance change?

This question becomes increasingly relevant as critical and trust-dependent systems are expected to remain operational for years or decades while experiencing repeated cryptographic transitions, hybrid algorithm deployment, hardware replacement, firmware modernization, identity recovery, authority succession, regulatory change, and intermittent connectivity [9].

Consider an industrial controller operating continuously for thirty years. Throughout its lifecycle it may experience multiple cryptographic transitions, hardware replacements, software modernization, identity recovery events, governance changes, and ownership transfers. Existing security mechanisms may correctly validate each individual transition. Nevertheless, determining whether the resulting trust relationship, protected data boundary, and cryptographic assurance have remained verifiable throughout its operational history constitutes a separate architectural challenge.

This paper identifies that challenge as **Trust Continuity**.

Trust Continuity does not replace authentication, cryptography, transparency, decentralized identity, trusted computing, or Zero Trust Architecture. It correlates **Continuity Evidence** produced by these technologies into an independently verifiable model of trust across time.

Figure 1 — Trust Continuity as a Correlation Problem

flowchart TB

subgraph Domains["Continuity Domains"]

CC["Cryptographic"]

IC["Identity"]

SC["System"]

SVC["Service"]

end

subgraph Evidence["Continuity Evidence"]

CE1["Migration Events"]

CE2["Identity Events"]

CE3["Attestation Events"]

CE4["Service Events"]

end

subgraph QVTM["QVTM Architecture Layer"]

CEP["CEP"]

GP["Graph Puzzle"]

CP["Continuity Paths"]

TC["Trust Continuity

Assessment"]

end

CC --> |emits| CE1

IC --> |emits| CE2

SC --> |emits| CE3

SVC --> |emits| CE4

CE1 --> |encapsulates as CEP| CEP

CE2 --> |encapsulates as CEP| CEP

CE3 --> |encapsulates as CEP| CEP

CE4 --> |encapsulates as CEP| CEP

CEP --> |feeds correlation| GP

GP --> |derives| CP

CP --> |supports| TC

TC -.feedback.-> Domains

classDef domain fill:#eef6ff,stroke:#1f77b4,stroke-width:2px;

classDef evidence fill:#fff7e6,stroke:#d28b00,stroke-width:1px;

classDef correlation fill:#f3e8ff,stroke:#7b2cbf,stroke-width:2px;

classDef result fill:#e8f7ec,stroke:#2e7d32,stroke-width:3px;

class CC,IC,SC,SVC domain;

class CE1,CE2,CE3,CE4 evidence;

class CEP,CP domain;

```
class GP correlation;  
class TC result;
```

The figure establishes the visual grammar used throughout this paper: blue nodes represent architectural domains and protocol structures, amber nodes represent Continuity Evidence, purple nodes represent correlation, and green nodes represent continuity assessment. Each arrow names the role of the relationship rather than merely indicating direction.

Accordingly, this paper advances the following research hypothesis:

Long-lived digital systems benefit from an architectural property capable of preserving and independently verifying trust relationships throughout cryptographic, operational, organizational, and governance transition.

Trust establishment validates trust at a point in time. Trust Continuity explains how that trust remains preserved and independently verifiable across time.

The objective of this paper is neither to introduce a new cryptographic primitive nor to replace existing trust architectures. It identifies a research gap in secure-system design, introduces MDCM as a candidate conceptual model, and presents QVTM as a reference protocol architecture for investigating that hypothesis.

Relationship to the QVTM Canonical Documentation Set

This Position Paper is the conceptual entry point to the QVTM documentation set. It defines the research problem, states the Trust Continuity hypothesis, introduces MDCM, and establishes the claim boundaries within which the rest of the canonical documentation operates.

The companion documents develop that position at different levels of authority and detail:

- The **QVTM White Paper** translates the position into a reference architecture and explains how the protocol core, Continuity Evidence Packages, Graph Puzzle, reference services, deployment profiles, and security model fit together.
- The **QVTM Protocol Specification** converts the architectural intent into externally observable, normative protocol behavior and interoperability requirements.
- The **QVTM SDK Reference** defines the supported public integration surface through which applications consume QVTM capabilities without redefining protocol semantics.
- The **QVTM Formal Foundations** express the principal concepts, relationships, invariants, and proof obligations in mathematical form and identify the boundary between established results and future formal validation.

The relationship is bidirectional. This paper constrains the purpose, terminology, and claims of the companion documents; those documents, in turn, test the position for architectural coherence,

protocol precision, implementability, and formal consistency. They elaborate this paper without replacing its role as the primary statement of the research position. Where a reader needs implementation detail, normative requirements, API behavior, or mathematical treatment, the corresponding companion document is authoritative for that layer.

01 — The Research Problem

Digital trust has traditionally been modeled as the validation of security properties at a specific point in time. Authentication and authorization establish who may act and under which conditions [2,5]. Public Key Infrastructures and transparency mechanisms establish accountable cryptographic trust anchors [3]. Secure communication protocols preserve confidentiality and authenticity in dynamic communication settings [4]. Platform and operational security mechanisms provide evidence about execution environments and industrial system posture [6,10].

Over the last decade, research has strengthened these foundations through advances in post-quantum cryptography, Zero Trust Architecture, software transparency, trusted execution technologies, decentralized identity, secure messaging, remote attestation, confidential computing, and operational resilience. These disciplines primarily address how trust is established, evaluated, protected, or monitored at specific points in time. Treating the preservation of trust relationships themselves throughout long-term operational evolution as an explicit architectural concern remains comparatively underexplored in current trust, resilience, and post-quantum migration literature [1,2,21].

This paper identifies that observation as an emerging architectural research gap.

Long-Lived Digital Systems

Trust Continuity becomes particularly visible in systems whose operational lifetime extends beyond the lifetime of the technologies upon which they depend. Representative examples include Operational Technology, Industrial Automation and Control Systems, Critical Infrastructure, defense platforms, space and satellite communications, national digital identity systems, digital sovereignty platforms, and long-term digital preservation systems.

Unlike conventional enterprise applications, these systems routinely remain operational for decades. During that time they may experience cryptographic migration, hybrid algorithm deployment, key rotation, hardware replacement, firmware modernization, operating system evolution, identity recovery, operator succession, organizational restructuring, governance change, regulatory evolution, and intermittent or disconnected operation.

The challenge is therefore not merely to establish trust. It is to preserve trust, confidentiality boundaries, integrity guarantees, privacy protection, and cryptographic survivability while the components used to establish those guarantees are themselves continuously changing.

A Motivating Example

Consider an industrial controller expected to operate continuously for thirty years.

Throughout its operational lifetime the controller may transition from classical public-key cryptography to multiple generations of post-quantum algorithms, undergo hardware replacement, software modernization, organizational ownership changes, policy evolution, identity recovery procedures, and repeated maintenance cycles.

Each individual transition may satisfy existing security requirements. Authentication may remain valid. Certificates may continue to verify correctly. Remote Attestation may continue reporting trusted measurements. Zero Trust policies may continue making correct authorization decisions. Operational monitoring may continue reporting healthy system status.

Yet a fundamental architectural question remains:

Can the resulting trust relationship itself still be preserved, verified, and distinguished from a manipulated or incomplete history after decades of cumulative operational evolution?

This paper argues that this question is not fully addressed by evaluating each transition independently. It requires an architectural capability capable of preserving continuity across those transitions.

Existing Architectural Foundations

Current trust architectures provide indispensable and complementary capabilities. Post-quantum cryptography preserves cryptographic survivability and algorithm agility against future quantum adversaries [1]. Zero Trust Architecture continuously evaluates authorization decisions under changing operational conditions [2]. Certificate Transparency provides independently verifiable evidence of certificate issuance [3]. Messaging Layer Security protects dynamic group communications [4]. Decentralized Identity and Verifiable Credentials improve portable identity management [5]. Trusted Platform Modules, Remote Attestation, Secure Boot, Measured Boot, and Confidential Computing provide verifiable platform integrity evidence [6,7,8]. Operational cybersecurity frameworks such as IEC 62443 and NIST SP 800-82 strengthen operational governance and resilience across industrial environments [9,10], while ENISA and ETSI guidance frame the post-quantum migration problem as an international transition challenge [11,12].

These technologies should not be viewed as competing approaches. They generate complementary categories of evidence: cryptographic evidence, identity evidence, platform evidence, operational assurance evidence, governance evidence, transparency evidence, and lifecycle evidence.

Trust Continuity is proposed as an architectural model capable of correlating these evidence domains rather than replacing the technologies that produce them.

Why the Post-Quantum Transition Matters

The transition toward post-quantum cryptography makes this architectural challenge more visible. Over the coming decades, many systems will migrate through multiple generations of cryptographic algorithms while operating in hybrid deployments that combine classical and post-quantum mechanisms. Cryptographic agility has therefore become an essential engineering property for long-lived secure systems [1].

The post-quantum transition should nevertheless be understood as one representative driver of trust evolution rather than its sole motivation. The broader challenge is preserving trustworthy relationships despite continuous technological, operational, organizational, and governance change. The post-quantum era provides a global opportunity to rethink digital trust from a lifecycle perspective.

Research Question

How can long-lived digital systems preserve and independently verify trust relationships throughout decades of cryptographic, operational, organizational, and governance evolution without depending exclusively on any single technology, authority, or implementation?

02 — Research Gap, Hypothesis, and Contributions

Should Trust Continuity itself be treated as a first-class architectural property?

This chapter formulates the research hypothesis and scientific contributions needed to investigate that question without introducing a new cryptographic primitive, authentication protocol, or trust management framework.

Throughout this paper, **Trust State** denotes the validity of a trust relationship at a particular point in time. **Trust Continuity** denotes the longitudinal property by which that relationship remains preserved and independently verifiable as supporting evidence evolves across time. This distinction separates the proposed architectural property from subjective confidence, organizational perception, policy preference, or any single implementation.

Recent years have produced substantial advances in multiple disciplines contributing to trustworthy digital systems. Research in post-quantum cryptography has focused on cryptographic survivability, algorithm agility, and migration strategies [1]. Zero Trust Architecture has redefined authorization by replacing implicit trust with continuous verification [2]. Transparency systems have strengthened accountability and independent auditability of security-critical events [3]. Messaging Layer Security has advanced secure communication for dynamic groups [4]. Decentralized Identity has improved portable digital identity [5]. Trusted execution technologies, Remote Attestation, Trusted Platform Modules, Secure Boot, Measured Boot, and Confidential Computing have strengthened platform

assurance [6,7,8]. Operational cybersecurity frameworks have improved resilience for Operational Technology and Critical Infrastructure throughout their operational lifecycle [9,10].

Collectively, this body of work demonstrates remarkable progress in establishing, validating, protecting, monitoring, and governing trust. Its common characteristic is that it usually investigates individual trust decisions or specific trust mechanisms. Less attention has been devoted to modeling the continuity of trust relationships themselves as an architectural property spanning decades of technological, operational, and organizational change.

This observation should not be interpreted as a limitation of existing work. Each discipline intentionally optimizes different architectural objectives. The problem is not the absence of trustworthy technologies, but the absence of an architectural model explicitly concerned with preserving the continuity of trust relationships across time.

Research Hypothesis

This paper hypothesizes that long-lived digital systems benefit from treating Trust Continuity as an explicit architectural property capable of preserving and independently verifying trust relationships throughout cryptographic, operational, organizational, governance, and lifecycle evolution.

The hypothesis intentionally avoids prescribing a particular implementation. It assumes future cryptographic evolution rather than any specific post-quantum algorithm or migration strategy. Trust Continuity is cryptographically agnostic, provided that the underlying cryptographic mechanisms satisfy the security properties required by the system throughout its operational lifetime.

If supported, the hypothesis suggests that future secure systems should explain not only why a trust relationship is currently valid, but also how that relationship has remained trustworthy throughout successive operational transitions.

Scientific Contributions

This paper advances five scientific contributions.

Contribution 1 — Scientific Concept. The paper proposes Trust Continuity as a candidate architectural property distinct from trust establishment, trust evaluation, and trust protection.

Contribution 2 — Conceptual Model. The paper introduces the **Multi-Domain Continuity Model (MDCM)** as a conceptual model organized around **Cryptographic Continuity**, **Identity Continuity**, **System Continuity**, and **Service Continuity**. The model remains independent of any particular implementation, authority, identity system, or cryptographic primitive.

Contribution 3 — Evidence Correlation. The paper introduces Evidence Correlation as the architectural process through which Trust Continuity emerges from Continuity Evidence rather than from any single security mechanism.

Contribution 4 — Reference Architecture. The paper presents **QVTM** as a reference protocol architecture intentionally designed around Trust Continuity and MDCM. QVTM is a reference realization of the model, not a replacement for the underlying scientific concept.

Contribution 5 — Research Agenda. The paper identifies open questions in formal semantics, continuity metrics, evidence interoperability, independent verification, and long-term preservation of trustworthy digital systems.

03 — Multi-Domain Continuity Model

The **Multi-Domain Continuity Model (MDCM)** decomposes long-lived system evolution into engineering domains whose evolution remains relatively independent while their evidence can be correlated to preserve and verify Trust Continuity. MDCM is not an implementation stack. It is a conceptual model for reasoning about continuity across the dimensions in which trustworthy systems change.

The model separates five concepts: **Continuity Domains** describe independent dimensions of system evolution; **Continuity Evidence** captures trustworthy evolution within each domain; **Evidence Correlation** derives continuity paths from that evidence; **Trust Continuity** emerges from those paths; and **QVTM** provides a reference protocol architecture for preserving and transporting the evidence needed for verification.

Why a Multi-Domain Model?

Modern cybersecurity increasingly recognizes that secure systems evolve simultaneously across multiple engineering dimensions. Cryptographic agility has become an accepted engineering requirement during the transition toward post-quantum cryptography [1,11,12]. Identity systems evolve independently through decentralized identity and portable credentials [5]. Operational resilience frameworks recognize continuous infrastructure, governance, and operational evolution [9,10,21]. Zero Trust Architecture similarly assumes continuous verification rather than permanent trust decisions [2].

These disciplines demonstrate that digital systems no longer evolve as monolithic entities. They also tend to investigate their respective domains separately. MDCM addresses the architectural question that appears between them: how evidence generated across independent engineering dimensions can be correlated to verify long-term Trust Continuity.

Architectural Principles

MDCM is guided by seven principles.

First, continuity should evolve independently across engineering domains whenever possible. Operational resilience and systems engineering literature already recognize that infrastructure modernization and governance evolution frequently occur independently [20,21].

Second, cryptographic agility is assumed rather than claimed as a contribution. Long-lived secure systems must already support algorithm transition, especially during post-quantum migration [1,11,12].

Third, Evidence Correlation becomes a first-class architectural process. Certificate Transparency, Trusted Computing, Remote Attestation, and Zero Trust independently generate trustworthy evidence concerning particular aspects of system behavior [2,3,6]. MDCM generalizes the need to correlate heterogeneous evidence across domains.

Fourth, continuity itself should remain independently verifiable. Independent verification already appears in transparency systems, public-key cryptography, and trusted execution technologies [2,3,6]. MDCM applies that expectation to continuity relationships rather than only to isolated trust states.

Fifth, continuity is accumulated through trustworthy evolution rather than repeatedly re-established after every change. This reflects the reality of industrial and critical infrastructure systems that evolve continuously throughout decades of operation [8].

Sixth, recoverability becomes an intrinsic continuity process rather than an exceptional operational event. Identity recovery, key recovery, and infrastructure recovery already exist; MDCM treats them as continuity-relevant transitions.

Seventh, the model remains architecturally independent of specific cryptographic algorithms, PKIs, Zero Trust implementations, distributed ledgers, transparency technologies, identity systems, trusted execution technologies, and deployment models. These technologies contribute evidence without defining the model.

The Four Continuity Domains

MDCM proposes four independent continuity domains: **Cryptographic Continuity**, **Identity Continuity**, **System Continuity**, and **Service Continuity**. These domains represent engineering dimensions rather than implementation layers. Their purpose is not to partition a system into components, but to identify where trustworthy evolution occurs and where Continuity Evidence must be produced.

Continuity Evidence

Continuity Evidence is independently verifiable information describing the trustworthy evolution of a particular continuity domain. Unlike conventional audit evidence, it is concerned with demonstrating how trustworthy evolution occurred over time, rather than merely describing the current system state.

Representative examples include cryptographic migration records, identity lifecycle events, remote attestation measurements, governance approvals, behavioral authorization events, and service continuity observations.

Domain 1 — Cryptographic Continuity

Cryptographic Continuity builds upon cryptographic agility and post-quantum migration strategies [1,11,12]. It produces Continuity Evidence related to algorithm migration, hybrid deployment, cryptographic policy, key lifecycle, and signature evolution.

Domain 2 — Identity Continuity

Identity Continuity builds upon decentralized identity and credential lifecycle management [5]. It produces Continuity Evidence related to identity recovery, delegated authority, ownership continuity, and credential evolution.

Domain 3 — System Continuity

System Continuity builds upon trusted execution and remote attestation [6,7], as well as operational resilience and industrial cybersecurity [9,21]. It produces Continuity Evidence related to firmware evolution, Remote Attestation, Secure Boot, infrastructure modernization, governance evolution, and behavioral evidence.

Domain 4 — Service Continuity

Service Continuity extends externally observable trust properties investigated through Zero Trust and secure communication architectures [2,4]. It produces Continuity Evidence related to authorization outcomes, service availability, policy compliance, and externally observable trust state.

Evidence Correlation

Evidence Correlation is the architectural process through which Continuity Evidence originating from multiple domains is correlated to derive continuity paths. It may operate continuously or be triggered by continuity-relevant events, depending on the implementation.

A minimal way to express the science behind the model is as an **Evidence Graph**. Trust states evolve over time, but their continuity is not proven by the state sequence alone. It is verified through evidence that supports transitions, branches, recovery events, and alternative continuity paths.

Figure 2 — Evidence Graph for Continuity Correlation

flowchart LR

subgraph States["Trust States"]

TS0(["TS₀"])

TS1(["TS₁"])

TS2(["TS₂"])

TS3(["TS₃"])

TSR(["TS

Recovery"])

end

subgraph Evidence["Continuity Evidence"]

CEa(["CE
Crypto"])

CEb(["CE
Identity"])

CEc(["CE
System"])

CEd(["CE
Service"])

end

GP{{"Graph Puzzle"}}

CP["Continuity Path"]

TC(["Trust Continuity
Assessment"])

TS0 -->|transition| TS1

TS1 -->|migration| TS2

TS2 -->|operation| TS3

TS1 -->|recovery branch| TSR

TSR -->|rejoin| TS3

CEa -.supports.-> TS0

CEa -.supports.-> TS1

CEb -.supports.-> TS1

CEb -.supports.-> TS2

CEc -.supports.-> TSR

CEc -.supports.-> TS3

CEd -.supports.-> TS2

CEd -.supports.-> TS3

CEa -->|correlates| GP

CEb -->|correlates| GP

CEc -->|correlates| GP

```
CEd --> |correlates| GP
GP --> |derives| CP
CP --> |supports| TC
```

```
classDef state fill:#eef6ff,stroke:#1f77b4,stroke-width:1px;
classDef evidence fill:#fff7e6,stroke:#d28b00,stroke-width:1px;
classDef correlation fill:#f3e8ff,stroke:#7b2cbf,stroke-width:2px;
classDef result fill:#e8f7ec,stroke:#2e7d32,stroke-width:3px;
```

```
class TS0,TS1,TS2,TS3,TSR state;
class CEa,CEb,CEc,CEd evidence;
class GP correlation;
class CP state;
class TC result;
```

The graph separates instantaneous states from continuity assessment. Trust States are points in time; Continuity Evidence supports transitions, branches, and recovery events; Graph Puzzle derives a continuity path; Trust Continuity is the resulting assessment when the path remains verifiable.

Trust Continuity and Trust Discontinuity

Trust Continuity is not another continuity domain. It is the architectural property emerging from successful Evidence Correlation across multiple domains.

A **Trust Discontinuity** occurs whenever available Continuity Evidence is no longer sufficient to independently verify the continuity of a trustworthy relationship across one or more continuity domains. Discontinuity does not necessarily imply compromise. It indicates that architectural continuity can no longer be demonstrated from the available evidence. Recovering from or repairing Trust Discontinuity therefore becomes an explicit responsibility of protocol architectures implementing MDCM, including QVTM.

Architectural Properties

MDCM is intended to satisfy separation of concerns, independent evolution, cryptographic agility, Continuity Evidence generation, Evidence Correlation, independent verification, recoverability, lifecycle preservation, technology independence, implementation independence, and architectural composability.

04 — QVTM: Quantum Verifiable Trust Mesh

QVTM (Quantum Verifiable Trust Mesh) is the reference protocol architecture implementing the principles defined by MDCM. Where MDCM explains the conceptual structure of Trust Continuity, QVTM explains how the evidence needed for Trust Continuity can be captured, preserved, transported, correlated, derived, and verified across decades of technological, cryptographic, and organizational change.

QVTM is therefore an architectural protocol rather than merely a communication protocol. Its responsibility is not to transport Trust Continuity as an object. Its responsibility is to manage Continuity Evidence so that Trust Continuity can be verified by verifiers.

What Does QVTM Mean?

The name Quantum Verifiable Trust Mesh reflects the architectural philosophy of the protocol.

Quantum acknowledges the transition toward post-quantum cryptography as one of the principal motivations for long-lived Trust Continuity. QVTM is cryptographically agile, but not cryptographically vague: in a post-quantum deployment profile, CEP integrity, key establishment, signing operations, and evidence protection may be bound to standards-aligned primitives, including NIST-standardized mechanisms such as ML-KEM and ML-DSA where appropriate [1]. Standards alignment is a deployment property, not a restriction on the architectural model; QVTM remains designed for future algorithm migrations and for compatibility with evolving international post-quantum standardization efforts. QVTM therefore complements post-quantum migration programs: it does not compete with algorithm standardization, but provides architectural continuity across successive migration phases.

Verifiable denotes that Trust Continuity can be verified from available Continuity Evidence without relying on implicit trust in any single authority, implementation, or administrative domain. Verification concerns architectural continuity rather than isolated cryptographic validation.

Trust denotes the architecturally observable relationship introduced in Chapter 2. QVTM never assumes implicit trust. Trust must remain supportable by Continuity Evidence.

Mesh denotes a distributed network of trust relationships derived from correlated Continuity Evidence rather than maintained by a single trust authority, certificate hierarchy, or distributed ledger.

Why QVTM Exists

Modern cybersecurity already provides mature technologies for establishing trustworthy digital systems. Post-quantum cryptography provides cryptographic survivability [1,11,12]. Zero Trust continuously evaluates authorization decisions [2]. Certificate Transparency provides independently verifiable certificate evidence [3]. Messaging Layer Security protects dynamic communication [4]. Decentralized Identity provides portable identities [5]. Trusted Computing and Operational Technology security frameworks provide operational assurance [6,7,9].

These technologies generate valuable evidence, but that evidence often becomes fragmented across administrative domains, infrastructures, organizational boundaries, and decades of operational evolution. Current mechanisms rarely preserve the continuity relationships connecting heterogeneous evidence throughout long-lived system evolution.

QVTM introduces a protocol architecture whose objective is to preserve the evidentiary basis that allows those relationships to remain independently verifiable. It allows digitally verifiable trust relationships to survive long-term change without repeatedly re-establishing trust from the beginning.

Why a Protocol Architecture?

Trust Continuity cannot be reduced to a certificate hierarchy, blockchain, audit database, authorization engine, transparency log, or single administrative authority. Long-lived digital systems exchange identities, cryptographic material, operational authority, and execution environments throughout their lifetime. Continuity Evidence must therefore become transportable.

QVTM captures, preserves, transports, correlates, and verifies Continuity Evidence so that Trust Continuity remains independently verifiable throughout the lifetime of digital systems.

Architectural Overview

QVTM is organized around a simple architectural invariant: evidence moves, continuity is verified. The protocol core captures Continuity Evidence, preserves it as CEPs, synchronizes or stores those packages across connected and disconnected environments, and enables Graph Puzzle to derive continuity paths when verification is required.

Figure 3 — QVTM Operational Flow

flowchart LR

Event(("Continuity-Relevant
Event"))

Capture["Evidence
Capture"]

CEP["CEP"]
Store["Preserve /
Synchronize"]

GP{{"Graph
Puzzle"}}

Path["Continuity
Path"]

Decision[["Verify / Recover /
Authorize"]]

subgraph Inputs["Reference Ecosystem Inputs"]

SaaS["SaaS
Control Plane"]

Login["Login
App"]

Vault["Offline
Vault"]

SDK["SDKs /
APIs"]

Msg["Secure
Messaging"]

end

Event --> |triggers| Capture

Capture --> |encapsulates as CEP| CEP

CEP --> |preserves / syncs| Store

Store --> |feeds evidence| GP

GP --> |derives| Path

Path --> |supports| Decision

Decision -.emits new event.-> Event

SaaS -.policy / enrollment evidence.-> Capture

Login -.identity evidence.-> Capture

Vault -.offline preservation.-> Store

SDK -.embedded capture / verification.-> Capture

SDK -.verification API.-> Decision

Msg -.message context evidence.-> Capture

classDef arch fill:#eef6ff,stroke:#1f77b4,stroke-width:2px;

```
classDef ecosystem fill:#f8f8f8,stroke:#666,stroke-dasharray: 4 3;
classDef correlation fill:#f3e8ff,stroke:#7b2cbf,stroke-width:2px;
classDef result fill:#e8f7ec,stroke:#2e7d32,stroke-width:3px;

class Capture,CEP,Store,Path arch;
class SaaS,Login,Vault,SDK,Msg ecosystem;
class GP correlation;
class Decision result;
```

The main path shows how continuity flows through QVTM: events generate evidence, evidence is encapsulated as CEPs, CEPs are preserved or synchronized, Graph Puzzle derives paths, and verifiers use those paths for authorization, recovery, audit, or continued operation. The reference ecosystem feeds or consumes this architecture; it does not define mandatory trust authorities.

Continuity Evidence Packages

A **Continuity Evidence Package (CEP)** is a transportable, independently verifiable collection of Continuity Evidence preserved by QVTM for synchronization, storage, recovery, and independent verification.

QVTM transports CEPs, not Trust Continuity itself. Trust Continuity is verified locally from available evidence. This distinction keeps the architectural property separate from any particular storage, transport, or deployment mechanism.

Graph Puzzle

Graph Puzzle is the reference Evidence Correlation mechanism of QVTM. It derives continuity paths from Continuity Evidence across multiple continuity domains.

Unlike a ledger, Graph Puzzle does not require a single global history. Unlike a certificate chain, it does not assume one hierarchy. Unlike an audit database, it does not merely preserve records. It derives an evidence graph of trust relationships from available evidence, allowing continuity to be evaluated across fragmented, delayed, offline, or independently governed environments.

Detailed graph semantics, correlation algorithms, and security arguments belong in the QVTM White Paper and protocol documentation. The essential position of this paper is narrower: QVTM requires a correlation mechanism because Trust Continuity is an evidence-correlation problem, not a linear validation problem.

QVTM Reference Ecosystem Demonstrating the Architectural Model

QVTM is both a protocol architecture and a reference ecosystem. The protocol architecture is the scientific object; the ecosystem demonstrates that the model is deployable. Its components are reference realizations of QVTM capabilities, not normative requirements for all implementations.

Capability	Role in Trust Continuity
Protocol Core	Defines CEP handling, evidence integrity, confidentiality boundaries, correlation boundaries, and verification semantics.
Graph Puzzle	Derives continuity paths from CEPs and other available evidence.
Infinitrust SaaS Control Plane	Coordinates connected enrollment, policy, evidence intake, operator lifecycle, tenant governance, and administrative workflows. It is a reference deployment component, not a mandatory trust authority.
Passwordless Login App	Demonstrates Identity Continuity by linking authentication events to recoverable, evidence-backed trust relationships.
Infinitrust Vault	Demonstrates offline confidentiality, integrity, recovery, and preservation of Continuity Evidence during disconnected operation.
SDKs and APIs	Allow customer systems and third-party applications to generate, consume, and verify continuity-aware evidence.
Secure Messaging Profiles	Demonstrate how communication confidentiality, integrity, participant state, and message-context evidence can participate in Trust Continuity.

This scope is intentionally executive. This paper does not specify each product implementation; it shows that login, offline preservation, secure messaging, evidence synchronization, recovery, integration, and independent verification can all be expressed through the same Trust Continuity model.

Security Objectives and Threat Coverage

QVTM does not claim to prevent every compromise event. Its primary security contribution is narrower and more architectural: it reduces **Trust Discontinuity Risk**—the risk that a trust relationship becomes unverifiable, ambiguous, silently downgraded, or indistinguishable from a manipulated history after cryptographic, identity, system, privacy, service, or governance change.

The architecture is designed to help address several failure classes. These are not presented as product features; they are the failure modes that make Trust Continuity operationally necessary. The

reference component column is illustrative: it identifies where the current Infinitrust ecosystem built on QVTM demonstrates each capability, not a normative requirement for every deployment.

Failure class	QVTM protection goal	Primary capability / reference component
MITM and integrity attacks	Bind evidence, identities, and protocol events so unauthorized modification, substitution, replay, or stale participant state can be detected.	Protocol Core, CEP, Graph Puzzle, Secure Messaging Profile
Unauthorized data disclosure / privacy compromise	Preserve confidentiality and privacy boundaries for protected assets and Continuity Evidence, especially during offline storage, recovery, and vault operation.	Infinitrust Vault, Protocol Core
Harvest-now, decrypt-later attacks	Support post-quantum deployment profiles so captured traffic, preserved evidence, and protected material do not depend indefinitely on vulnerable classical confidentiality.	Cryptographic Continuity, PQC Profile, Vault
Rollback and downgrade attacks	Preserve migration and policy evidence so verifiers can detect suspicious returns to older algorithms, states, authorities, or trust anchors.	Protocol Core, Graph Puzzle
Fragmented audit trails	Correlate evidence across domains rather than relying on one isolated log, database, or authority.	CEP, Graph Puzzle
Identity recovery ambiguity	Preserve Identity Continuity evidence across recovery, delegation, renewal, and authority succession.	Login App, Identity Capsule, SaaS Control Plane
Offline synchronization ambiguity	Preserve CEPs during disconnected operation and verify continuity when systems reconnect.	Infinitrust Vault, SDKs/APIs
Governance discontinuity	Maintain evidence of policy, operator, tenant, and authority changes across organizational evolution.	SaaS Control Plane
Stale or compromised trust anchors	Treat anchor changes as continuity-relevant events rather than invisible administrative updates.	Protocol Core, Cryptographic Continuity

QVTM also inherits conventional security objectives from the cryptographic and transport mechanisms selected by a deployment profile. Confidentiality, integrity, authentication, forward secrecy, resistance to MITM-style tampering, and post-quantum resistance remain essential. The differentiator is that QVTM does not stop at protecting a session or validating a state; it preserves the evidentiary basis needed to show that trust remained coherent across state changes. QVTM also separates cryptographic security objectives from architectural continuity:

Layer	Primary responsibility
Cryptographic mechanisms	Confidentiality, integrity, authentication, key establishment, signing, and post-quantum resistance
QVTM architecture	Preservation, transport, correlation, and verification of Continuity Evidence across system evolution

Core Differentiators

QVTM is not another authentication method, another vault, another control plane, or another cryptographic wrapper. Its differentiators are architectural and operational:

- it treats long-term trust as something sustained and verified through evidence rather than assumed from the current state;
- it protects confidentiality and integrity while also preserving the continuity evidence needed to prove that protection remained meaningful over time;
- it addresses harvest-now, decrypt-later risk through standards-aligned post-quantum deployment profiles and Cryptographic Continuity;
- it resists MITM-style substitution, rollback, downgrade, fragmented audit history, data privacy compromise, and silent trust-anchor drift by making continuity-relevant changes explicit;
- it works across online, offline, recovery, and governance transitions rather than assuming permanent connectivity or one central authority;
- it separates the protocol core from reference ecosystem components so the architecture can be implemented, tested, and integrated without becoming a monolithic product.

A common criticism is that continuity could be solved by combining Zero Trust, PQC migration, audit logs, and signed metadata. QVTM makes a stronger claim:

Existing approach	Missing property	QVTM contribution
Zero Trust	Temporal continuity	Evidence-backed continuity across authorization changes
PQC migration	Evolution semantics	Cryptographic Continuity across algorithm generations
Audit logs	Verifiable continuity reasoning	CEP + Graph Puzzle evidence correlation
Identity systems	Cross-domain continuity	Identity Continuity linked to system and service evolution

Existing approach	Missing property	QVTM contribution
Secure messaging	Long-term trust semantics around communication context	Message-context evidence linked to continuity paths
Signed metadata	Cross-domain continuity semantics	Evidence packages bound to lifecycle events and correlation logic
Vault storage	Trust semantics around preserved data	Offline preservation of protected data and Continuity Evidence

QVTM is therefore not a replacement for these approaches. It is the architecture that connects their evidence when trust must remain meaningful across time. The same distinction applies to conventional logs:

Conventional Logs	QVTM
Passive historical records	Continuity Evidence designed for continuity verification
Stored after events	Bound to continuity-relevant transitions
Usually system-local	Cross-domain and transportable through CEPs
Useful for audit	Used for independent continuity reasoning
Preserve what happened	Help determine whether trust remained verifiable

This is the out-of-the-box contribution of QVTM: it turns long-term trust from an operational assumption into an architectural object that can be captured, preserved, correlated, and independently evaluated. Digital systems already know how to establish trust; QVTM introduces an architecture for keeping that trust meaningful, verifiable, private, integrity-protected, and post-quantum survivable as the world around it changes.

Relationship with MDCM and Existing Technologies

MDCM defines the conceptual vocabulary: Continuity Domains, Continuity Evidence, Evidence Correlation, Continuity Paths, and Trust Continuity. QVTM operationalizes those concepts through a protocol core and reference ecosystem.

QVTM complements existing technologies rather than replacing them. It does not define cryptographic algorithms, identity standards, PKI architectures, transparency protocols, authorization models, distributed consensus, or blockchain systems. Instead, it defines how evidence

produced by those technologies can be preserved and correlated so that Trust Continuity can be verified independently of any single one of them.

05 — Representative Trust-Dependent Systems

The Infinitrust ecosystem built on QVTM is intended for systems in which trust relationships outlive the technologies implementing them. The representative systems below are therefore illustrative deployment contexts, not a closed market taxonomy.

Trust Continuity as a Horizontal Architectural Capability

Modern cybersecurity has produced mature technologies addressing specific trust problems. Industrial cybersecurity focuses on operational resilience [9]. Identity systems focus on authentication, authorization, and identity lifecycle management [5]. Zero Trust Architecture focuses on continuous authorization decisions [2]. Post-quantum cryptography focuses on cryptographic survivability [1]. Secure messaging protocols focus on protecting communications [4]. Each discipline succeeds within its own objective.

Modern digital ecosystems increasingly combine these technologies. As a result, trust relationships frequently survive longer than the technologies implementing them. This challenge is no longer confined to Operational Technology or Critical Infrastructure. It also affects enterprise IT, cloud-native systems, financial services, digital identity, healthcare, artificial intelligence, consumer applications, secure software supply chains, and digital government [24,25].

Trust Continuity is therefore a horizontal architectural capability applicable to trust-dependent systems rather than to industry-specific systems.

Family I — Critical Operational Systems

Critical operational systems frequently remain operational for decades while undergoing continuous technological modernization. Representative contexts include Operational Technology, Critical Infrastructure, defense, space systems, healthcare infrastructure, and medical devices. Representative assets include industrial control systems, electrical grids, satellites, defense platforms, hospital infrastructure, and implantable medical devices.

The principal architectural challenge is preserving trustworthy operation throughout repeated infrastructure, cryptographic, and organizational change. The most relevant continuity domains are usually System Continuity, Cryptographic Continuity, and Identity Continuity. In these environments, post-quantum migration represents one example of a broader continuity transition rather than the defining objective of Trust Continuity itself.

Family II — Digital Trust Platforms

Modern digital ecosystems depend on long-lived trust relationships across Identity and Access Management, Customer Identity and Access Management, Single Sign-On, passwordless authentication, enterprise SaaS, cloud-native applications, digital wallets, API ecosystems, and secure messaging services. Secure messaging belongs in this family because participant membership, key evolution, message context, and communication confidentiality are also continuity-relevant relationships [4].

These systems continuously evolve identities, credentials, authorization policies, cryptographic algorithms, application services, communication contexts, and software supply-chain evidence [24,25]. Trust Continuity becomes relevant because users, organizations, and services expect these relationships to survive repeated technology change without repeatedly re-establishing trust. The dominant continuity domains are Identity Continuity, Service Continuity, and Cryptographic Continuity.

Family III — Financial Trust Systems

Financial ecosystems preserve trust relationships whose lifetime frequently exceeds the lifetime of the technologies implementing them. Representative contexts include banking infrastructure, payment platforms, digital asset custody, securities trading, insurance platforms, digital contracts, and interbank settlement.

Representative continuity challenges include regulatory evolution, cryptographic migration, organizational restructuring, long-term auditability, and fraud resilience. Trust Continuity complements existing financial trust models by preserving continuity relationships throughout those transitions. The most relevant domains are Identity Continuity, Cryptographic Continuity, and Service Continuity.

Family IV — Distributed and Autonomous Systems

Autonomous and distributed ecosystems introduce continuity challenges that cannot depend on permanent connectivity. Representative contexts include autonomous vehicles, robotics, unmanned systems, edge computing, industrial edge platforms, autonomous factories, and distributed AI agents.

These systems frequently operate under intermittent connectivity, delayed synchronization, decentralized decision making, and local autonomy. QVTM addresses this condition by allowing CEPs to be preserved locally before later synchronization and independent verification. The most relevant domains are System Continuity, Service Continuity, and Identity Continuity.

Family V — Long-Lived Information Systems

Some digital assets must remain trustworthy for periods significantly exceeding the lifetime of contemporary technologies. Representative contexts include governmental archives, scientific repositories, legal evidence, engineering documentation, pharmaceutical research, aerospace documentation, and cultural heritage.

Traditional preservation primarily protects information. Trust Continuity additionally preserves the evidentiary relationships associated with those preserved assets. The most relevant domains are Cryptographic Continuity, Identity Continuity, and Service Continuity.

Comparative Overview

Table 1 — Representative Trust-Dependent Systems

System Family	Representative Contexts	Dominant Continuity Domains	Primary Continuity Challenge
Critical Operational Systems	OT, Critical Infrastructure, defense, space, healthcare	System, Cryptographic, Identity	Trustworthy operation across decades of modernization
Digital Trust Platforms	IAM, CIAM, SSO, SaaS, wallets, APIs	Identity, Service, Cryptographic	Trust relationships surviving credential, policy, and platform change
Financial Trust Systems	Banking, payments, custody, settlement, contracts	Identity, Cryptographic, Service	Auditability and trust persistence across regulation and infrastructure change
Distributed and Autonomous Systems	Robotics, autonomous vehicles, edge, AI agents	System, Service, Identity	Continuity under intermittent connectivity and local autonomy
Long-Lived Information Systems	Archives, legal evidence, science, aerospace, heritage	Cryptographic, Identity, Service	Long-term preservation of evidentiary trust relationships

Continuity Intensity

Not every trust-dependent system experiences identical continuity requirements. Some environments undergo limited technological evolution. Others experience continuous cryptographic migration, organizational transformation, regulatory evolution, and infrastructure modernization

throughout several decades. This paper refers to this observation informally as **Continuity Intensity**. A formal continuity intensity model remains an open research question.

Common Architectural Characteristics

Despite their diversity, the representative families share common characteristics: trust relationships extending beyond individual technologies, continuous system evolution, cryptographic agility, Continuity Evidence, recoverability, interoperability, and resilience against organizational change. These shared characteristics motivate Trust Continuity as a general architectural capability rather than a sector-specific solution.

Broader Implications

The representative contexts demonstrate that QVTM is not limited to Operational Technology, defense, post-quantum migration, or identity management. It addresses a common architectural problem: digitally verifiable trust relationships must survive continuous technological, cryptographic, and organizational change.

This is where the ecosystem matters. The protocol core defines continuity semantics; the control plane coordinates connected operation; the login app exercises Identity Continuity; the vault demonstrates disconnected preservation; and SDKs expose the model to customer systems. Together, these capabilities make Trust Continuity operational rather than merely descriptive, while remaining subordinate to the protocol architecture introduced in Chapter 04.

Regulatory Perspective

Emerging regulatory frameworks increasingly require organizations to demonstrate cybersecurity throughout extended operational lifecycles rather than treating security as a one-time deployment activity. Examples include resilience requirements for critical infrastructure, digital sovereignty initiatives, and long-term cyber resilience programmes promoted by international cybersecurity organizations [9,11,12]. Although MDCM is not derived from regulatory frameworks, these developments reinforce the practical relevance of continuity-oriented architectures.

06 — QVTM within the Existing Cybersecurity Ecosystem

If modern cybersecurity already provides mature trust architectures, why introduce another one?

The answer proposed by this paper is intentionally different from that of conventional security technologies. QVTM does not compete with existing trust architectures. It introduces Trust Continuity as a continuity layer that preserves the evidence needed to verify trust continuity across technologies whose individual responsibilities remain unchanged.

Trust Continuity should therefore be understood as an architectural dimension orthogonal to existing trust architectures, not as an alternative to them.

Existing Trust Architectures Solve Different Questions

Modern cybersecurity already provides mature architectural models. Some establish identity. Some authorize access. Some verify execution integrity. Some protect communications. Some ensure cryptographic resilience. Some preserve immutable records. Collectively, these architectures form today's cybersecurity ecosystem.

The question investigated here is different. Existing architectures ask whether trust can be established, verified, or protected. Trust Continuity asks whether an established trust relationship can remain independently verifiable across successive technology generations, organizational transitions, and cryptographic migrations.

Because QVTM operates on Continuity Evidence rather than replacing existing security mechanisms, it can coexist with current cybersecurity architectures without requiring architectural replacement.

Foundational Architectural Relationships

Zero Trust Architecture

Zero Trust asks whether access should be granted under the current operational context. It continuously evaluates authorization decisions using contextual information, device posture, and identity state [2]. QVTM assumes those decisions already exist and preserves the evidentiary continuity connecting trustworthy authorization decisions over time. The two architectures therefore answer complementary questions.

Post-Quantum Cryptography

Post-quantum cryptography asks whether cryptographic security can survive future quantum adversaries, including harvest-now, decrypt-later risk. It replaces vulnerable mechanisms with quantum-resistant alternatives such as NIST-standardized ML-KEM for key establishment and ML-DSA for digital signatures [1]. QVTM neither defines nor replaces those algorithms. It binds their use, migration, and replacement into Cryptographic Continuity Evidence so future verifiers can determine whether cryptographic transitions remained trustworthy.

Public Key Infrastructure

PKI asks whether identities can be authenticated through trusted certification authorities. QVTM preserves the evidentiary continuity of those trust anchors throughout certificate renewal, authority succession, and algorithm replacement. Trust establishment and Trust Continuity therefore address distinct architectural concerns.

Digital Identity

Decentralized Identity and Verifiable Credentials ask whether digital identities can remain portable and independently verifiable [5]. QVTM extends identity portability toward Identity Continuity by allowing continuity relationships to remain verifiable after credential renewal, recovery procedures, and organizational transition.

Supporting Architectural Technologies

Several mature cybersecurity technologies naturally become contributors of Continuity Evidence.

Certificate Transparency contributes certificate history [3]. Within MDCM this history becomes Cryptographic Continuity Evidence.

Trusted Computing and Remote Attestation contribute execution measurements. Within MDCM these measurements become System Continuity Evidence.

Messaging Layer Security protects communication confidentiality, authenticity, and forward secrecy [4]. QVTM complements such protocols by preserving the evidence surrounding identities, cryptographic material, and service relationships, strengthening resistance to MITM-style substitution, stale participant state, and ambiguous trust recovery.

Distributed ledgers preserve replicated histories through distributed consensus. QVTM neither requires nor excludes blockchain technology. Distributed ledgers may preserve certain forms of Continuity Evidence. Unlike distributed ledgers, QVTM does not seek global consensus; its objective is independent verification from heterogeneous evidence regardless of where that evidence is stored. Consequently, distributed ledgers become one possible persistence mechanism rather than an architectural dependency.

Architectural Synthesis

The comparisons reveal a consistent pattern. Each existing architecture contributes evidence describing one aspect of a trust-dependent system. MDCM organizes those evidence sources into Continuity Domains. QVTM captures, preserves, and transports evidence through CEPs. Graph Puzzle derives continuity paths from that evidence. Trust Continuity is evaluated from those paths.

This distinction separates transport, reasoning, and architectural outcome:

CEPs move. Continuity Evidence is correlated. Continuity paths are derived. Trust Continuity is verified.

Rather than replacing existing trust architectures, QVTM provides architectural processes that connect them throughout long operational lifetimes.

Comparative Summary

Figure 4 — QVTM as an Evidence Correlation Fabric

flowchart TB

subgraph Existing["External Trust Architectures"]

ZT["Zero Trust"]

PKI["PKI"]

PQC["PQC / ML-KEM / ML-DSA"]

DID["Digital Identity"]

TC["Trusted Computing"]

MLS["Secure Messaging"]

CT["Transparency"]

DLT["Distributed Ledgers"]

end

subgraph Evidence["Continuity Evidence"]

C1["Cryptographic"]

C2["Identity"]

C3["System"]

C4["Service"]

end

subgraph QVTM["QVTM Architecture Layer"]

CEP["CEP"]

GP["Graph Puzzle"]

Paths["Continuity Paths"]

end

Trust["Trust Continuity
Assessment"]

ZT --> |authorization evidence| C4

PKI --> |certificate lifecycle| C1

PQC --> |algorithm migration| C1

DID --> |identity lifecycle| C2

TC --> |attestation evidence| C3

MLS --> |communication context| C4

CT --> |transparency evidence| C1

DLT --> |optional persistence| C1

DLT -.optional persistence.-> C3

C1 --> |encapsulates as CEP| CEP

C2 --> |encapsulates as CEP| CEP

C3 --> |encapsulates as CEP| CEP

C4 --> |encapsulates as CEP| CEP

CEP --> |feeds correlation| GP

GP --> |derives| Paths

Paths --> | supports | Trust
Trust -.verification feedback.-> Existing

```
classDef external fill:#f4f4f4,stroke:#777,stroke-width:1px;
classDef evidence fill:#fff7e6,stroke:#d28b00,stroke-width:1px;
classDef arch fill:#eef6ff,stroke:#1f77b4,stroke-width:2px;
classDef correlation fill:#f3e8ff,stroke:#7b2cbf,stroke-width:2px;
classDef result fill:#e8f7ec,stroke:#2e7d32,stroke-width:3px;

class ZT,PKI,PQC,DID,TC,MLS,CT,DLT external;
class C1,C2,C3,C4 evidence;
class CEP,Paths arch;
class GP correlation;
class Trust result;
```

Each arrow names the role of the contributing architecture: it emits, preserves, contextualizes, or optionally persists evidence. QVTM does not absorb those technologies or replace their security objectives; it correlates their evidence so continuity paths can be derived and verified.

Table 2 — Architectural Comparison

Architecture	Primary Objective	Continuity Role within MDCM
Zero Trust	Current authorization decision	Provides authorization and context evidence
Post-Quantum Cryptography	Cryptographic survivability	Provides migration and algorithm-transition evidence
PKI	Certificate-based identity and trust anchors	Provides certificate lifecycle and authority evidence
Digital Identity	Portable identity relationships	Provides identity lifecycle and recovery evidence
Certificate Transparency	Public certificate accountability	Provides transparency evidence for cryptographic continuity
Trusted Computing	Platform integrity assurance	Provides system measurement evidence
Messaging Layer Security	Secure dynamic group communication	Provides communication and participant evidence
Distributed Ledgers	Replicated historical records	May provide persistence for selected evidence
QVTM	Trust Continuity	Preserves and transports evidence for verification

Architectural Implications

Trust Continuity should not be interpreted as another competing cybersecurity architecture. It introduces an orthogonal dimension operating across existing trust technologies. Existing architectures establish, evaluate, verify, and protect trust states. QVTM preserves and transports the evidence needed to verify the continuity connecting those states across time.

Mature cybersecurity technologies therefore become complementary producers of Continuity Evidence rather than competing architectural alternatives. Multiple protocol implementations may realize the architectural principles introduced by QVTM while remaining compatible with MDCM.

Key Takeaway

QVTM introduces Trust Continuity as a first-class architectural concern, complementing rather than replacing the trust architectures upon which modern cybersecurity already depends.

07 — Trust Continuity as a New Architectural Dimension

The central contribution of this work extends beyond the proposal of a reference protocol. This paper argues that Trust Continuity satisfies the characteristics of a foundational architectural dimension for reasoning about long-lived digital trust.

Architectural dimensions do not emerge merely because new terminology is introduced. They emerge when they expose a class of questions that existing properties and paradigms cannot answer without losing essential information.

What Defines a New Architectural Dimension?

The evolution of cybersecurity has repeatedly been shaped by the recognition of new architectural concerns. The classical properties—confidentiality, integrity, and availability—established how information and systems should be protected. Subsequent concepts such as authenticity, accountability, and non-repudiation extended architectural reasoning into digital interactions. More recently, architectural paradigms such as Zero Trust [2] and Cyber Resilience [10] shifted attention from static protection toward continuous verification and operational survivability.

Each development expanded cybersecurity because it introduced questions that could not be answered adequately by the concepts that preceded it. This paper argues that Trust Continuity follows the same pattern. Existing properties explain how trust is established, evaluated, protected, or recovered. They do not explain how trust relationships remain demonstrably connected throughout decades of cryptographic migration, technology replacement, organizational transition, and infrastructure modernization.

This question cannot be reduced to confidentiality, integrity, availability, authentication, or resilience without losing its architectural meaning. For this reason, Trust Continuity should be regarded as an additional architectural dimension rather than another security mechanism.

Distinguishing Trust Continuity from Existing Continuity Concepts

The concept of continuity is already well established across several engineering disciplines. Business Continuity focuses on preserving organizational operations during disruption. Operational Continuity focuses on preserving dependable system behavior despite maintenance, failure, or environmental change. Service Continuity focuses on preserving the availability of digital services.

These disciplines ask whether systems continue operating. Trust Continuity asks whether trust semantics remain independently verifiable.

Table 3 — Continuity Disciplines

Continuity Discipline	Primary Concern
Business Continuity	Preservation of organizational operations
Operational Continuity	Preservation of dependable system behavior
Service Continuity	Preservation of service availability
Trust Continuity	Preservation of independently verifiable trust semantics

These concepts are complementary rather than competing. A system may maintain uninterrupted operation while losing the ability to justify why its current trust relationships remain valid. Conversely, trust semantics may remain verifiable even after temporary operational disruption.

Connecting Established Scientific Disciplines

The proposed architectural perspective does not establish an isolated research domain. It connects multiple mature disciplines through a common objective.

Industrial cybersecurity contributes lifecycle reasoning for infrastructures expected to remain trustworthy throughout decades of modernization [9][10]. Digital identity contributes authentication, authorization, portable identities, and identity lifecycle management [5]. Post-quantum cryptography contributes cryptographic agility and migration strategies for long-lived systems [1][11]. Transparency systems and evidence provenance contribute historical information describing how trust evolves over time [3][5]. Distributed Systems contributes models for synchronization, consistency, and fault tolerance [15,16]. Graph Theory contributes mathematical reasoning about

relationships and connectivity [17]. Formal Methods contributes implementation-independent verification techniques [18,19]. Systems Engineering and Resilience Engineering contribute principles for designing dependable long-lived systems [20,21]. Secure logging, tamper-evident history, and provenance research contribute mechanisms for preserving evidence without reducing continuity to passive record keeping [14,22,23].

Viewed together, these disciplines already provide many of the scientific building blocks required to reason about Trust Continuity. Their integration reveals a cross-disciplinary architectural perspective rather than an extension of any single cybersecurity specialization.

Semantic Properties

If Trust Continuity is accepted as an architectural dimension, future protocol implementations should preserve a common semantic foundation independently of implementation choices.

Verification properties require Continuity Evidence, deterministic verification, and continuity-path verifiability. Architectural properties require technology agnosticism, implementation-neutral semantics, cryptographic agility, and long-term architectural invariance. Interoperability properties require composability across heterogeneous continuity domains together with interoperability between independent implementations.

These properties describe architectural behavior rather than protocol behavior. Multiple protocol realizations may therefore preserve the same architectural semantics while employing different cryptographic primitives, transport mechanisms, deployment profiles, or trust infrastructures.

Open Scientific Questions

Recognizing Trust Continuity as an architectural dimension introduces new scientific questions:

- Which formal semantics most appropriately describe continuity relationships?
- Which invariants distinguish valid Continuity Paths from invalid continuity interpretations?
- How should heterogeneous Continuity Evidence be correlated across independent continuity domains?
- Under which conditions can Trust Continuity be verified from partial observations?
- How should Continuity Strength, Continuity Confidence, and Continuity Intensity be formally defined?
- Which verification techniques provide the strongest correctness guarantees for continuity-oriented architectures?
- Which architectural properties remain invariant across heterogeneous deployment environments?

These questions extend beyond protocol engineering into formal methods, distributed systems, graph theory, cyber resilience, and trustworthy computing. Their existence reinforces the

proposition that Trust Continuity represents an emerging scientific research direction rather than merely another protocol design problem.

Scientific Significance

For decades, cybersecurity has primarily focused on establishing trust, evaluating trust, and protecting trust. This paper argues that future architectures may also need to preserve the independently verifiable continuity of trust semantics throughout the lifetime of digital systems.

If that perspective proves valuable through continued scientific evaluation and independent peer review, future architectures may reason not only about whether trust currently exists, but also about whether that trust remains demonstrably continuous despite cryptographic migration, infrastructure modernization, organizational transition, and technological replacement.

The implications extend beyond QVTM itself. Multiple protocol realizations may eventually implement the architectural principles introduced here while remaining architecturally equivalent and interoperable.

Architectures evolve more slowly than technologies. Protocols may evolve. Technologies will inevitably change. If Trust Continuity is indeed a foundational architectural dimension, its architectural principles should remain invariant across those changes.

08 — Conclusions

The accelerating evolution of digital infrastructures, the transition toward post-quantum cryptography, the emergence of autonomous systems, and the increasing operational lifetime of critical digital assets challenge one of the implicit assumptions underlying contemporary cybersecurity: that trust can always be evaluated from the current system state alone.

This paper argues that the assumption is no longer sufficient. For long-lived digital systems, trust must not only be established, evaluated, and protected. Its confidentiality, integrity, identity, authorization, privacy, and cryptographic assurances must also remain independently verifiable throughout technological, organizational, and cryptographic evolution.

This work identifies that architectural property as Trust Continuity and presents QVTM as a reference protocol architecture and ecosystem intentionally designed around that objective.

The Relationship Between QVTM and Trust Continuity

Trust Continuity defines the architectural objective. QVTM provides a reference realization through a protocol core and an operational ecosystem.

During the architectural design of QVTM, it became evident that the protocol was addressing a broader challenge than secure communication, cryptographic agility, or identity management alone.

The same continuity model now connects the core protocol, CEPs, Graph Puzzle, control-plane coordination, passwordless login, offline vault operation, SDKs, and integration APIs.

QVTM should therefore not be viewed merely as a protocol solving a specific engineering problem. It is presented as the reference ecosystem through which the architectural principles of Trust Continuity were first systematically articulated and explored.

Principal Contributions

This paper argues that Trust Continuity represents an architectural concern that cannot be fully explained by existing cybersecurity properties or paradigms. It introduces MDCM as a conceptual framework for reasoning about continuity relationships across heterogeneous trust domains. It presents QVTM as a reference protocol architecture and ecosystem intentionally designed around those principles, demonstrating that Trust Continuity is not only conceptually definable but also operationally realizable. It also positions both the concept and its reference realization within the existing cybersecurity ecosystem.

Together, these contributions provide an initial scientific foundation for considering Trust Continuity as an architectural dimension while illustrating how that dimension can be embodied by a concrete protocol architecture.

Architectural Perspective

Trust should not be understood solely as the validity of a current security state. For long-lived digital systems, trust should also be evaluated through the continuity of the relationships connecting trustworthy states throughout technological evolution.

This perspective neither contradicts nor replaces existing cybersecurity principles. It complements them by introducing an architectural dimension explicitly concerned with preserving trust across time. QVTM provides one reference realization of that objective, while alternative protocol realizations may ultimately emerge.

Broader Implications

If the architectural perspective proposed in this paper proves valuable through continued scientific evaluation, its implications extend beyond any individual protocol. Future identity systems may preserve Identity Continuity in addition to identity validity. Cryptographic infrastructures may preserve Cryptographic Continuity alongside algorithm agility. Operational platforms may explicitly reason about System Continuity across decades of modernization. Distributed services may preserve Service Continuity despite autonomous operation, intermittent connectivity, and organizational transition.

Collectively, these developments suggest that continuity may become an organizing architectural principle for long-lived trust-dependent systems, with QVTM representing one intentionally designed realization of that broader principle.

Final Reflection

Architectural principles generally outlive the technologies that first reveal them. Architectures evolve more slowly than technologies because they describe enduring design principles rather than particular engineering solutions.

Protocols evolve. Cryptographic algorithms are replaced. Implementation techniques change.

This paper argues that Trust Continuity belongs to the architectural layer. It presents QVTM as a reference ecosystem intentionally designed around that principle.

If Trust Continuity ultimately becomes an accepted architectural dimension, QVTM may be understood not only as a protocol architecture, but as an ecosystem through which that concept becomes implementable: login, vault, control plane, SDK, CEPs, secure messaging profiles, and graph-based evidence correlation all serving the same continuity objective.

Trust Continuity is the scientific contribution. QVTM demonstrates that it can be engineered, deployed, and evolved without sacrificing cryptographic agility, interoperability, confidentiality, integrity, privacy, or long-term verifiability. The long-term value of this work will therefore depend on two related outcomes: whether Trust Continuity proves useful as an architectural concept, and whether QVTM proves useful as a practical vehicle for implementing that concept in long-lived digital trust systems.

Technical Documentation Path

This paper stands on its own as the architectural argument for Trust Continuity. The broader QVTM canonical documentation set provides the next level of detail without changing that argument.

Readers seeking architectural realization should continue with the **QVTM White Paper**. Readers evaluating interoperability or building an independent implementation should use the **QVTM Protocol Specification**. Application and platform developers should use the **QVTM SDK Reference** for the supported public integration surface. Researchers and formal-methods reviewers should use the **QVTM Formal Foundations** for mathematical definitions, invariants, and proof obligations.

These documents form a controlled chain of elaboration:

Position Paper White Paper Protocol Specification SDK Reference Position Paper
Formal Foundations

The first path moves from scientific position to architecture, normative protocol behavior, and supported integration. The second relationship is reciprocal: the Position Paper supplies the concepts and claim boundaries, while the Formal Foundations test their internal consistency and expose assumptions that require sharper architectural language. Together, the five documents describe one coherent system at complementary levels of abstraction.

Closing Statement

Cybersecurity has traditionally sought to establish, evaluate, and protect trust. This paper argues that future digital trust architectures must also preserve its continuity, confidentiality, integrity, privacy, and post-quantum survivability across change. QVTM is presented as a reference ecosystem intentionally designed to make that architectural objective practical across login, messaging, offline vaulting, control-plane coordination, and third-party integration.

References

- [1] National Institute of Standards and Technology. *FIPS 203, FIPS 204, and FIPS 205 — ML-KEM, ML-DSA, and SLH-DSA Post-Quantum Cryptography Standards*, 2024.
- [2] National Institute of Standards and Technology. *SP 800-207 — Zero Trust Architecture*, 2020.
- [3] Laurie, B., Langley, A., & Kasper, E. *Certificate Transparency Version 2*. RFC 9162, Internet Engineering Task Force.
- [4] Barnes, R., et al. *The Messaging Layer Security (MLS) Protocol*. RFC 9420, Internet Engineering Task Force.
- [5] World Wide Web Consortium. *Decentralized Identifiers (DID) v1.0; Verifiable Credentials Data Model v2.0; PROV Data Model*.
- [6] Trusted Computing Group. *TPM Library Specification*.
- [7] Sabt, M., Achemlal, M., & Bouabdallah, A. *Trusted Execution Environment: What It Is, and What It Is Not*. IEEE, 2015.
- [8] Costan, V., & Devadas, S. *Intel SGX Explained*. IACR Cryptology ePrint Archive, 2016.
- [9] International Electrotechnical Commission. *IEC 62443 — Industrial Automation and Control Systems Security*.
- [10] National Institute of Standards and Technology. *SP 800-82 Rev. 3 — Guide to Operational Technology (OT) Security*, 2023.
- [11] ENISA. *Post-Quantum Cryptography: Current State and Quantum Mitigation*.

- [12] ETSI. *Quantum-Safe Cryptography Technical Reports*.
- [13] European Commission. *EU Cybersecurity Strategy and related Digital Decade policy publications*.
- [14] World Wide Web Consortium. *PROV-DM: The PROV Data Model*, 2013.
- [15] Lamport, L. *Time, Clocks, and the Ordering of Events in a Distributed System*. Communications of the ACM, 1978.
- [16] Schneider, F. B. *Implementing Fault-Tolerant Services Using the State Machine Approach: A Tutorial*. ACM Computing Surveys, 1990.
- [17] Diestel, R. *Graph Theory*. Springer, 5th edition, 2017.
- [18] Lamport, L. *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. Addison-Wesley, 2002.
- [19] Basin, D., Cremers, C., & Meadows, C. *Model Checking Security Protocols*. Handbook of Model Checking, Springer, 2018.
- [20] ISO/IEC/IEEE. *ISO/IEC/IEEE 42010 — Systems and Software Engineering — Architecture Description*, 2011.
- [21] Hollnagel, E., Woods, D. D., & Leveson, N. *Resilience Engineering: Concepts and Precepts*. Ashgate, 2006.
- [22] Schneier, B., & Kelsey, J. *Secure Audit Logs to Support Computer Forensics*. ACM Transactions on Information and System Security, 1999.
- [23] Crosby, S. A., & Wallach, D. S. *Efficient Data Structures for Tamper-Evident Logging*. USENIX Security Symposium, 2009.
- [24] NIST. *SP 800-218 — Secure Software Development Framework (SSDF)*, 2022.
- [25] OpenSSF. *Supply-chain Levels for Software Artifacts (SLSA)*.