

QVTM cryptographic conformance implementation note

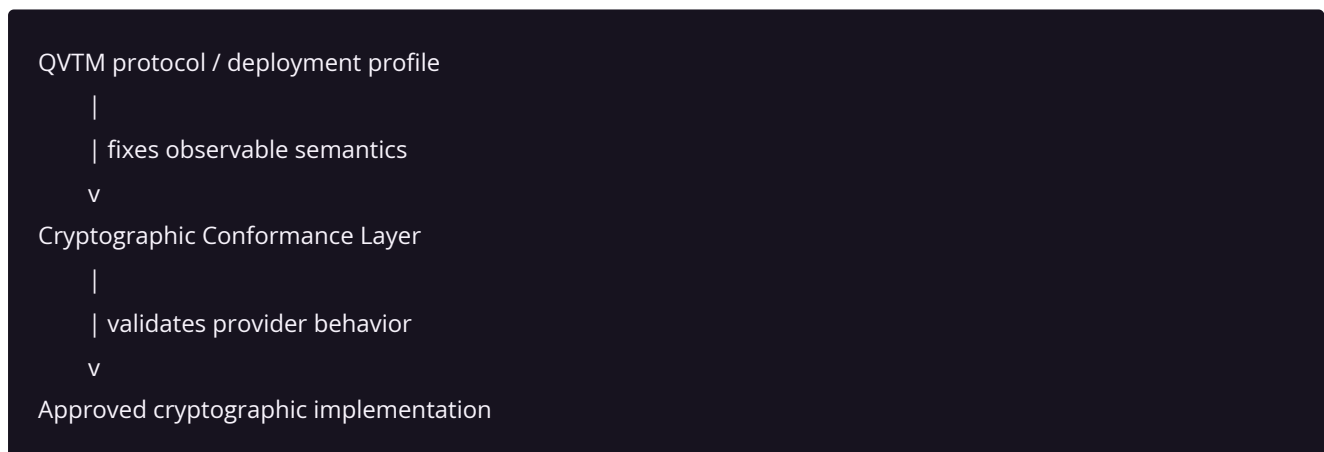
Status: Review **Version:** 1.0 **Authority:** Informative; non-normative **Scope:** Public explanation of the implementation boundary used to keep QVTM protocol semantics independent from a concrete cryptographic provider.

Purpose

QVTM defines protocol-visible semantics. Cryptographic libraries implement standardized primitives. Those responsibilities must remain separate.

The QVTM **Cryptographic Conformance Layer (CCL)** is an implementation pattern used by the Infinitrust Vault V3 reference implementation to preserve that separation. It is not a new cryptographic primitive and does not modify the QVTM Protocol Specification v1.0.

Conceptually:



The protocol or applicable deployment profile owns requirements such as algorithm selection, parameter sets, canonical encodings, context and domain-separation rules, transcript construction, accepted input/output forms, rejection behavior, and interoperability expectations. A concrete library, operating-system service, hardware module, or other provider is an implementation slot beneath that boundary.

Provider independence

Replacing a cryptographic provider is not, by itself, a protocol change when all protocol-visible semantics remain unchanged and the replacement passes the same applicable conformance evidence.

A provider replacement must not silently:

- change wire or stored encodings;
- downgrade algorithms or parameter sets;
- invent provider-specific context strings or transcript rules;
- change failure semantics in a way that weakens fail-closed behavior;
- expose secret material that the QVTM boundary treats as opaque; or
- invalidate interoperability with conforming implementations of the same profile.

Provider reputation, certification status, maintenance activity, or implementation language is not a substitute for conformance evidence. Those factors may be relevant to product assurance, but the implementation must still satisfy the exact QVTM-facing contract.

Conformance evidence

A product using this pattern should bind its provider choice to reviewable evidence appropriate to the profile, including:

1. normative algorithm evidence or applicable standardized test vectors;
2. deterministic adapter vectors where deterministic comparison is meaningful;
3. protocol-composition vectors covering canonical bytes, context binding, KDF/AAD construction, commitments, and signature inputs where applicable;
4. negative and adversarial cases that fail closed;
5. cross-implementation checks where feasible; and
6. release evidence identifying the exact provider source, version, configuration, features, and build target.

Conformance is established during development, CI, audit, and release. Normal offline protocol operation must not depend on fetching remote provider metadata or conformance material at runtime.

Relationship to QVTM v1

QVTM v1 already requires cryptographic agility, observable protocol determinism, interoperability within claimed profiles, and continuity across algorithm transitions. The CCL is one implementation

technique for enforcing those requirements in software.

It therefore has a deliberately narrow claim boundary:

- **Normative QVTM semantics** remain owned by the applicable Protocol Specification and Deployment Profile.
- **CCL contracts** constrain an implementation so a provider cannot redefine those semantics.
- **Provider assurance** remains a separate product and release responsibility.

The current Infinitrust Vault V3 CCL uses a fixed product profile including ML-KEM-768, ML-DSA-65, SHA-256, HKDF-SHA-256, and ChaCha20-Poly1305. That product profile must not be interpreted as making those exact algorithms permanent universal requirements of every future QVTM Deployment Profile.

Publication boundary

This note supplements the frozen QVTM v1 public documents without editing or superseding them. If the CCL concept is incorporated normatively into a future Protocol Specification, that change must be issued through normal public versioning and review rather than by silently modifying the frozen v1 publication.