WHITEPAPER

Crypto-Agility: Building Cryptographic Systems That Can Adapt

Post-quantum cryptography isn't the last cryptographic transition your organisation will face. Crypto-agility is what determines whether the next one takes months or years.

This Has Happened Before

The post-quantum transition is significant, but it isn't unprecedented. MD5 and SHA-1 were both once considered secure hashing algorithms and are now deprecated after practical collision attacks were demonstrated. 512-bit and then 1024-bit RSA keys were both once standard and are now considered too weak. Every cryptographic algorithm has a shelf life determined by advancing cryptanalysis and computing power — quantum computing is simply the largest such advance the industry has had to plan for at once, across every public-key algorithm simultaneously.

Organisations that handled previous transitions smoothly generally shared one trait: cryptography wasn't hard-wired throughout their systems in ways that made a change require touching every application individually.

What Crypto-Agility Actually Means

Crypto-agility is the ability to change cryptographic algorithms, key sizes, and protocols across an environment without re-architecting the applications and systems that depend on them. In practice, that means:

- Algorithms are configured, not hard-coded, in applications and infrastructure.
- Certificate and key lifecycle management is automated, not manual and ad hoc.
- There is a current, accurate inventory of what algorithms are in use and where.
- Systems can run classical and new algorithms side by side during a transition.

Why the CBOM Is the Foundation of Agility, Not Just Migration

It's easy to think of a Cryptography Bill of Materials as a one-time artefact needed specifically for the post-quantum transition. That undersells it. A living, continuously updated CBOM is what makes any future cryptographic change tractable — this one or the next one. Without it, every transition starts with the same expensive discovery phase, repeated from scratch, because nothing was kept current after the last migration finished.

Agility Is an Architecture Decision, Not a Migration-Week Task

Retrofitting agility into a system already built with hard-coded algorithm choices is far more expensive than designing for it from the outset. For new systems, that means abstracting cryptographic operations behind interfaces that can be reconfigured, favouring standard libraries with pluggable algorithm support over custom implementations, and automating certificate and key rotation from day one rather than treating it as a manual operational task to fix later.

For existing systems, the practical path is incremental: use the current migration as the opportunity to remove hard-coded cryptographic assumptions wherever they're touched, rather than making a like-for-like algorithm swap that leaves the same rigidity in place for the next transition.

Measuring Agility, Not Just Migration Progress

A migration percentage answers "how much have we moved." A crypto-agility metric answers a more durable question: "if we needed to change algorithms again next year, how much of our estate could adapt without a re-architecture project." The second question is the one that determines whether this transition is the last expensive one, or simply the first of several.