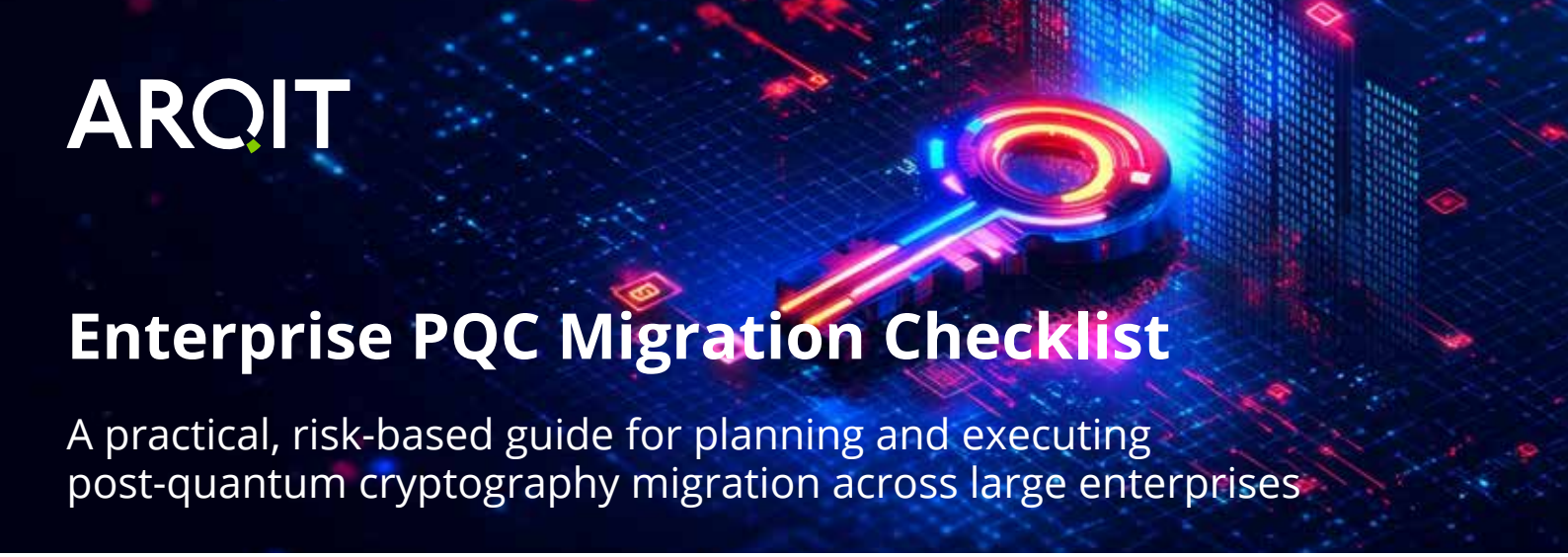

ARQIT

Enterprise PQC Migration Checklist

A practical, risk-based guide for planning and executing post-quantum cryptography migration across large enterprises

PQC migration is not a single cryptography upgrade. It is an enterprise discovery, prioritization, governance, vendor-alignment, implementation and lifecycle-management program. Hybrid deployment and cryptographic agility should be treated as design principles, not afterthoughts.

Regulatory direction and planning horizon

	Near-term marker	Mid-program marker	Completion target
United States	Annual prioritized inventories of CRQC-vulnerable systems required from 2023 onward	NIST PQC standards published in 2024; agencies and industry are expected to begin planning and applying them now	OMB cites NSM-10 goal to mitigate as much quantum risk as feasible by 2035
United Kingdom	**By 2028:** discovery complete and initial migration plan defined	**By 2031:** highest-priority upgrades completed and roadmap refined	**By 2035:** complete migration of systems, services and products
European Union	**By end-2026:** transition begins under coordinated roadmap	**By end-2030:** high-risk use cases migrated	**By end-2035:** migration completed across risk levels

How to use this checklist

Work through the five stages below in sequence, but treat them as overlapping workstreams. The objective is to reduce exposure to store-now-decrypt-later risk, migrate critical services without disruption, and avoid locking the enterprise into a brittle first-generation PQC architecture.

01. Discover

Build a cryptography inventory that is operationally useful, not merely theoretical.

- Identify where public-key cryptography is used - PKI, certificates, TLS termination, VPNs, email, code signing, identity systems, HSM-backed services, embedded devices, OT/IoT, applications and cloud platforms.
- Map trust chains and dependencies - Include internal platforms, managed services, SaaS, appliances, SDKs, protocol libraries and outsourced processing environments.
- Record upgrade constraints - Flag assets with hard-coded crypto, long hardware refresh cycles, proprietary interfaces, certification requirements or limited maintenance windows.

Consider using an automated discovery, detection and inventory tool, such as Encryption Intelligence.

02. Assess

Translate inventory into risk, business impact and migration priority.

- Prioritize by exposure, not asset count - Focus first on long-lived sensitive data, externally exposed trust anchors, critical infrastructure, regulated services and systems with high blast radius.
- Account for store-now-decrypt-later risk - Treat confidentiality lifetime as a planning variable; data needing protection for many years may require earlier action even if the system is not today business-critical.
- Separate quick wins from hard cases - Commodity IT may move with vendor updates; bespoke systems, legacy OT, constrained devices and interoperability-heavy environments need earlier design attention.

03. Plan

Create a migration program with accountable owners and an agile cryptography strategy.

- Stand up cross-functional governance - Include security, enterprise architecture, infrastructure, application owners, operations, procurement, legal/regulatory and GRC.
- Adopt hybrid and crypto-agile design principles - Plan for staged deployment using classical + PQC approaches where appropriate, and make future algorithm rotation feasible without major redesign.
- Engage vendors early - Require product roadmaps, standards alignment, interoperability evidence, certification plans and contractual clarity on support windows and remediation obligations.

Consider a scalable, software based PQC solution, such as Arqit's SKA-Platform.

04. Migrate

Execute in controlled waves while preserving resilience and interoperability.

- Pilot before broad rollout - Test performance, handshake size, latency, certificate impacts, device constraints, failover behavior and operational tooling before retiring legacy methods.
- Sequence by risk and feasibility - Migrate higher-risk and easier-to-modernize domains first, while using the lessons to prepare harder environments.
- Define exit criteria for legacy crypto - Do not declare success at pilot stage; specify when RSA/ECC dependencies can be decommissioned, where exceptions are allowed and how residual risk is governed.

05. Monitor

Treat PQC as an ongoing capability, not a one-time project.

- Maintain cryptographic visibility - Keep the inventory live and link it to CMDB, vulnerability management and architecture governance so new quantum-vulnerable uses are detected early.
- Track standards and implementation maturity - Monitor NIST, NSA/CNSA, NCSC, EU guidance, protocol updates and vendor implementation quality rather than assuming the first release is the final answer.
- Manage lifecycle and incidents - Be prepared to rotate algorithms, replace weak implementations, respond to interoperability failures and continuously validate protection for data at rest, in transit and in process.

Reference basis
US timing corroborated against OMB M-23-02, NIST PQC standards and transition guidance, and UK/EU milestone checks against NCSC and European Commission / ENISA publications.