Designing for a Post-Quantum World

Srinath Perera,
Head of Research,
WSO2

WSO2

That guy who walks in and tells you where the bug is in the program?

Quantum Computing maps the problem to a physical process (e.g., to a low energy state) and finds the answer from a simulation of the process!

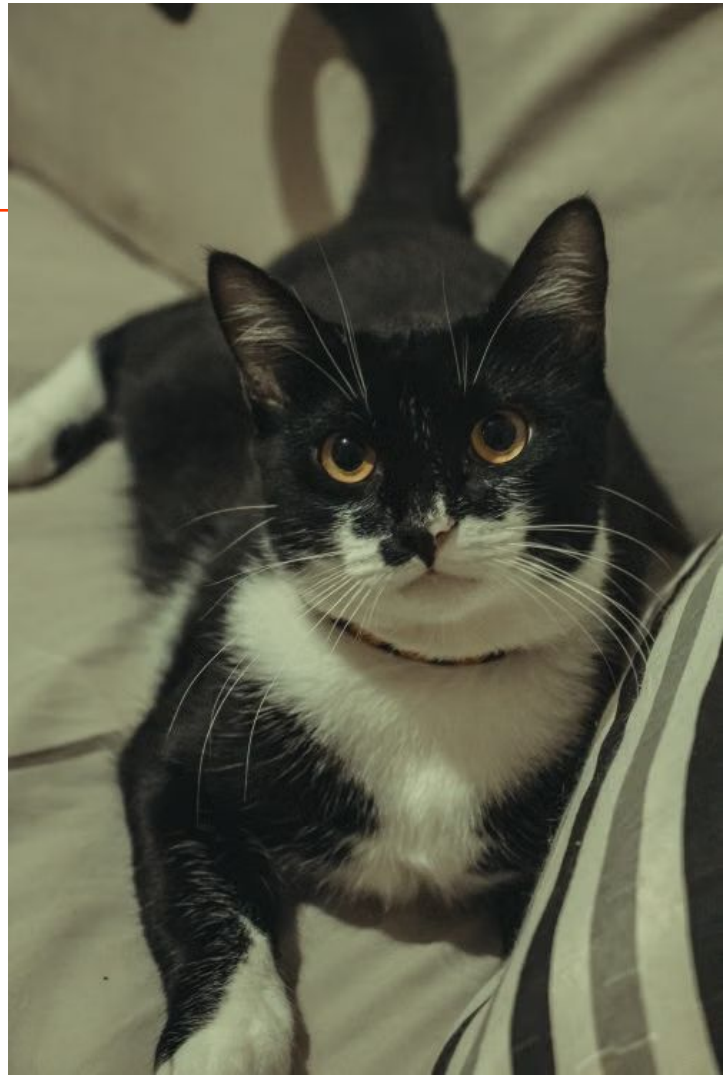
WSO2

Work has been happening for a long time!

- Shor’s (1994) and Grover’s (1996) algorithms, first working 2-qubit(IBM, 1998)
- A lot of parallel work from IBM, Google, Universities and companies (Oxford, UNSW, D-Wave ..)
- We are at 100s- 1000s of qubits (depends on how they are counted)

Qubit numbers are complicated.

- Physical vs. logical
- Each qubit is error-prone and collapses; many physical qubits back a logical qubit.
- Google Willow showed adding more physical qubits per logical qubit actually reduced the logical error rate, which is the long-sought proof that the error-correction approach scales.



Cryptographically Relevant Quantum Computer (CRQC)

- Factors on Q-Day
  - Qubits
  - Error correction rates and stability time
  - How good is the quantum algorithm?
- The current forecast is 10+ years.

So what?

Symmetric
Key Cryptography

Grover's algorithm weakens (but does not break)

halves the effective security strength

Answer: Double the key strength
- e.g., (AES-256, SHA-384/512)
- relatively cheap

Asymmetric
Key Cryptography

Prime factorization as an example

Shor's algorithm breaks today's asymmetric cryptography (both RSA and elliptic curve)

Why does it matter now?

- Harvest-now-and-decrypt-later
  - Exposes any data with a long confidentiality horizon
    - health records
    - government and defense communications
    - financial records
    - engineering archives/ source code
- Switching takes time
- Being late might mean it is expensive
- Timelines are unpredictable
- Quantum + classical combinations might break things faster

Why does it matter now? Mosca's inequality

Only thing organisation can control

often measured in years to decades.

is outside any single organization's control and estimates for it keep compressing

If T(migration) + T(protect) > T(collapse) − the time until Q-Day

an organization is already too late for data it needs to protect.

Why does it matter now? (Contd.)

- US Executive Order 14412 ("Securing the Nation Against Advanced Cryptographic Attacks," June 22, 2026)
  - High-value assets deadlines: December 31, 2030, for key-establishment/encryption migration and December 31, 2031, for digital signatures
- EU Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography
  - End of 2026 — EU member states should have begun their PQC transition.
  - End of 2030 — critical infrastructure operators must have completed their transition.
- More from the UK, Germany, Australia, India, etc.

What parts will break?

- TLS - TLS/mTLS termination points
  - E.g., ingress, service mesh, service-to-service calls
- Signatures
  - E.g., digital signatures used for non-repudiation, audit trails, and code/firmware signing, token signing (JWT/OIDC/SAML, OAuth)
- Key Management
  - PKI hierarchies and certificate chains
- Blockchain/ledger consensus or wallet signatures
- VPNs and IPsec/IKEv2 tunnels, SSH
- Data-at-rest encryption, hash
- IoT - Embedded devices and industrial controllers

Post-Quantum Security (PQC)

- The good news is, work has been happening for some time
- Main work is from NIST - they have published algorithms and recommendations
- ISO and IETF are also doing some work
- Some governments (e.g., Germany, China) have suggested slightly stronger algorithms than NIST
- PQC algorithms are new and are not mathematically proven unbreakable.
    - Will discuss how to work around this

Fundamental Algorithms

- Key encapsulation
  - ML-KEM (formerly Kyber)
    - Default choice for key exchange; fast, moderate size
- Digital signature
  - ML-DSA (formerly Dilithium)
    - Default general-purpose signature
  - SLH-DSA (formerly SPHINCS+) Digital signature FIPS
    - Hash functions only, conservative, minimal assumptions; large, slow signatures

Roman bronze ring key
3rd–4th century CE

How do Algorithms solve Problems?

- TLS - TLS/mTLS termination points
  - ML-KEM (e.g., OpenSSL 3.5+, BoringSSL)
- Signatures
  - ML-DSA (FIPS 204) and SLH-DSA (FIPS 205)
  - Needs a defined encoding before a JWT library can use it
- Key Management - PKI hierarchies and certificate chains
  - ML-DSA/SLH-DSA sign certificates directly once a CA supports them.
  - E.g., HSMs (Te.g. hales Luna, Entrust nShield), CA/PKI platforms (EverTrust, DigiCert, cloud Private CA services)
  - OpenSSL 3.5+/Bouncy Castle (direct access) .
- Blockchain/ledger consensus or wallet signatures if applicable
  - being discussed
- VPNs and IPsec/IKEv2 tunnels, SSH
  - ML-KEM spec is a draft
- Data-at-rest encryption
  - AES 2X key
- IoT - Embedded/IoT/OT devices and industrial controllers
  - ML-DSA/SLH-DSA

How heavy are those implementations?

- Compute overhead is small to negligible.
- Size overhead is the dominant cost: public keys, ciphertexts, and signatures are 5–50x larger than classical equivalents (kilobytes vs. tens/hundreds of bytes), and Classic McEliece-style code-based public keys exceed a megabyte.
  - May be affected by slow networks
  - Hardware acceleration often covers the gap.

What about Cloud, Middleware … ?

- TLS library
  - OpenSSL 3.5+
- Browser
  - Most browsers support it and have it enabled by default
    - Without client support, connections fall back to classical
    - Cloudflare says ⅔ already negotiate hybrid PQC
- Edge/CDN
  - CloudFront
  - Google Cloud CDN
- Load balancing
  - AWS, Google Cloud, NGINX, HAProxy, F5
  - Can switch via TLS lib
- Key Management
  - Most cloud ones do; HashiCorp Vault
- IAM/ Identify
  - WSO2 IAM partially
  - Others at discussion level
- Databases - can get via TLS switch
  - MySQL yes

PQC Architecture Patterns

- The gateway / "bump-in-the-wire" pattern
  - A network security term for a device in the communication path to add security without modifying either endpoint.
  - Terminate the PQC, and handshake at a gateway, reverse proxy, load balancer, or service-mesh sidecar that you do control (Don't touch the legacy application or database at all)
    - E.g., NGINX and HAProxy (ML-KEM), F5 (ML-KEM), Istio/Envoy sidecars in a service mesh (ML-KEM)
  - The pattern ONLY buys time; you need to upgrade or retire the legacy component.
  - Often Used for Databases
- Crypto-agility (the foundational, enabling pattern)
  - Concretely, this pattern is usually implemented as an abstraction layer between application code and the underlying crypto library

PQC Architecture Patterns (Contd.)

- Hybrid cryptography (classical + PQC combined)
  - Use both a classical algorithm (X25519, RSA, ECDSA) and a PQC algorithm (e.g., ML-KEM, ML-DSA) side by side
  - a hedge against an undiscovered flaw in the still-young PQC standards
  - at the cost of larger handshakes/certificates and two algorithms
- Cryptographic discovery and inventory (CBOM)
  - Bill of Materials covering certificates, TLS/SSH/IPsec configurations, embedded/hard-coded algorithm choices, HSM key material, code-signing and firmware-signing keys, and data-at-rest encryption
  - Discovering application-layer and embedded/hard-coded cryptography is hard.
- Confidentiality-first staged rollout (KEM before signatures)
  - Harvest-now-and-decrypt-later risk applies to data confidentiality but not (in the same way) to signatures, do key-establishment/encryption migration ahead of digital-signature migration

What do you have to do? (greenfield)

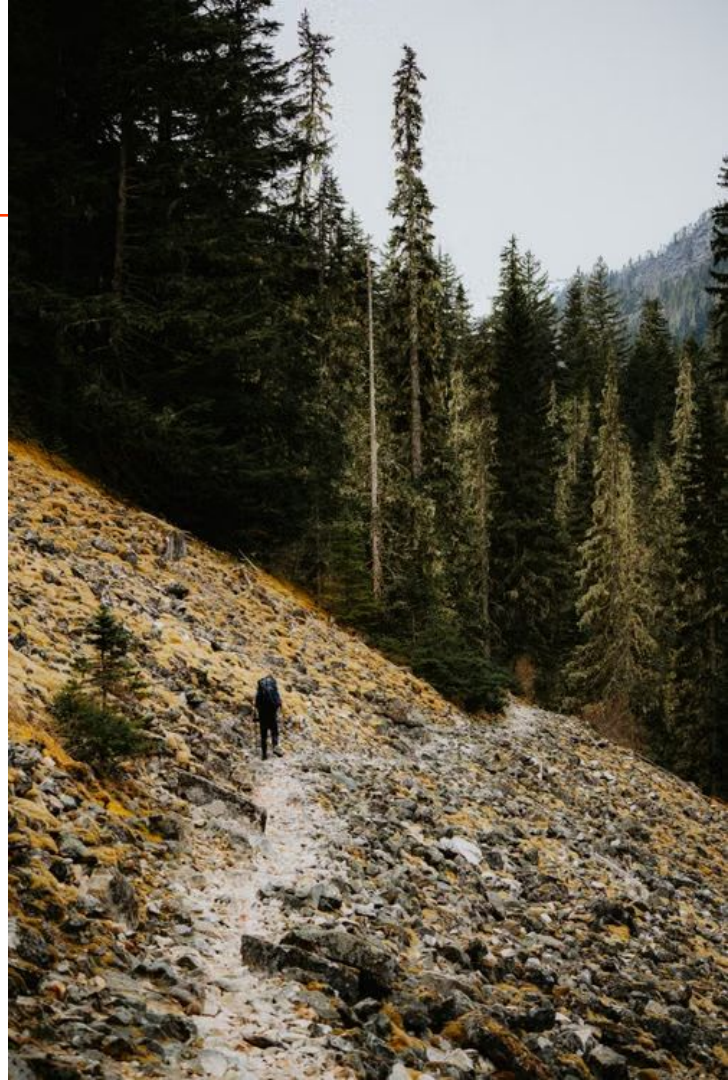
- Greenfield - You can change all parts of the system
- Steps
  - Build the crypto inventory first
  - Adopt crypto-agility pattern
  - Upgrade the TLS stack to a PQC-capable library at every termination point (incoming and outgoing)
  - Hybrid cryptography
  - Move PKI to hybrid certificates, re-encrypt long-lived data-at-rest, and migrate signing roots.

What do you have to do? (Contd.)

- Brownfield — some services and databases can't be changed
- Same as above, plus
  - Use the gateway/"bump-in-the-wire" pattern
  - Push PQC requirements into vendor management and procurement so the constraint is temporary rather than permanent, and maintain dual (old+new) support with a defined transition window rather than a hard cutover.

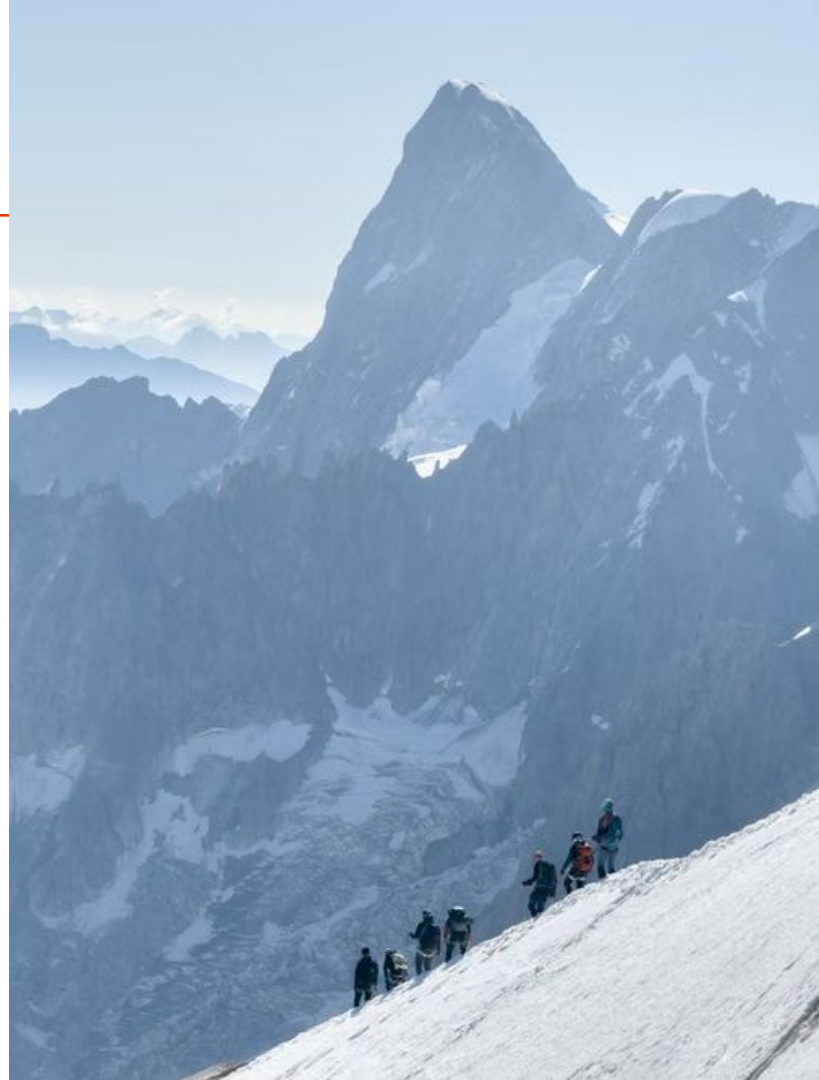
Challenges

- Mixed fleets during transition
  - Some peers support PQC, some don't
- Migration effort is reliably underestimated, even by teams staffed with genuine experts
- Cross-team/cross-organization coordination overhead
  - Changes often ripple into other teams' code
- Organizational inertia, cost, and skills gap
- Fragmented, evolving international regulation
- A false sense of "we're done" from partial coverage



WSO2's Own Migration Status

- TLS (inbound and outbound)
  - IS7, IS8 (upcoming), Ballerina
- Signatures
  - IS8 (upcoming)
- Data-at-rest encryption, hashing
  - IS8, APIM, IS7
- Key Management
  - IS7
- Rest is future work



Conclusion

- Due to harvest-now-and-decrypt-later attacks, we are entering a critical stage where PQC needs serious attention
  - Government recommendations strengthen this
- Algorithms and middleware are in place; we need to integrate
  - Delays can be costly.
- As a developer, architect, product manager
  - Create a PQC roadmap; communicate to vendors
  - Use tools that support PQC
  - Update your applications
- As a vendor
  - Make a roadmap and communicate
  - Work towards PQC

Questions?

WSO2

Who am I?

- Long term Open Source Contributor
  - Apache Member, one of the people who started Apache Axis2, working on open source projects for 20+ years
- Author of the book “Software Architecture and Decision-Making” (Pearson, 2024)
- Researcher working on AI + Systems
  - (e.g., we had agent reliability paper at ACM CAIS), has 1500+ citations, 40+ peer reviewed papers
- Other
  - Did my Ph.D. in US, came back and live in Sri Lanka
  - Enjoys reading and photography
  - Opinionated guy but believe in continual dialog