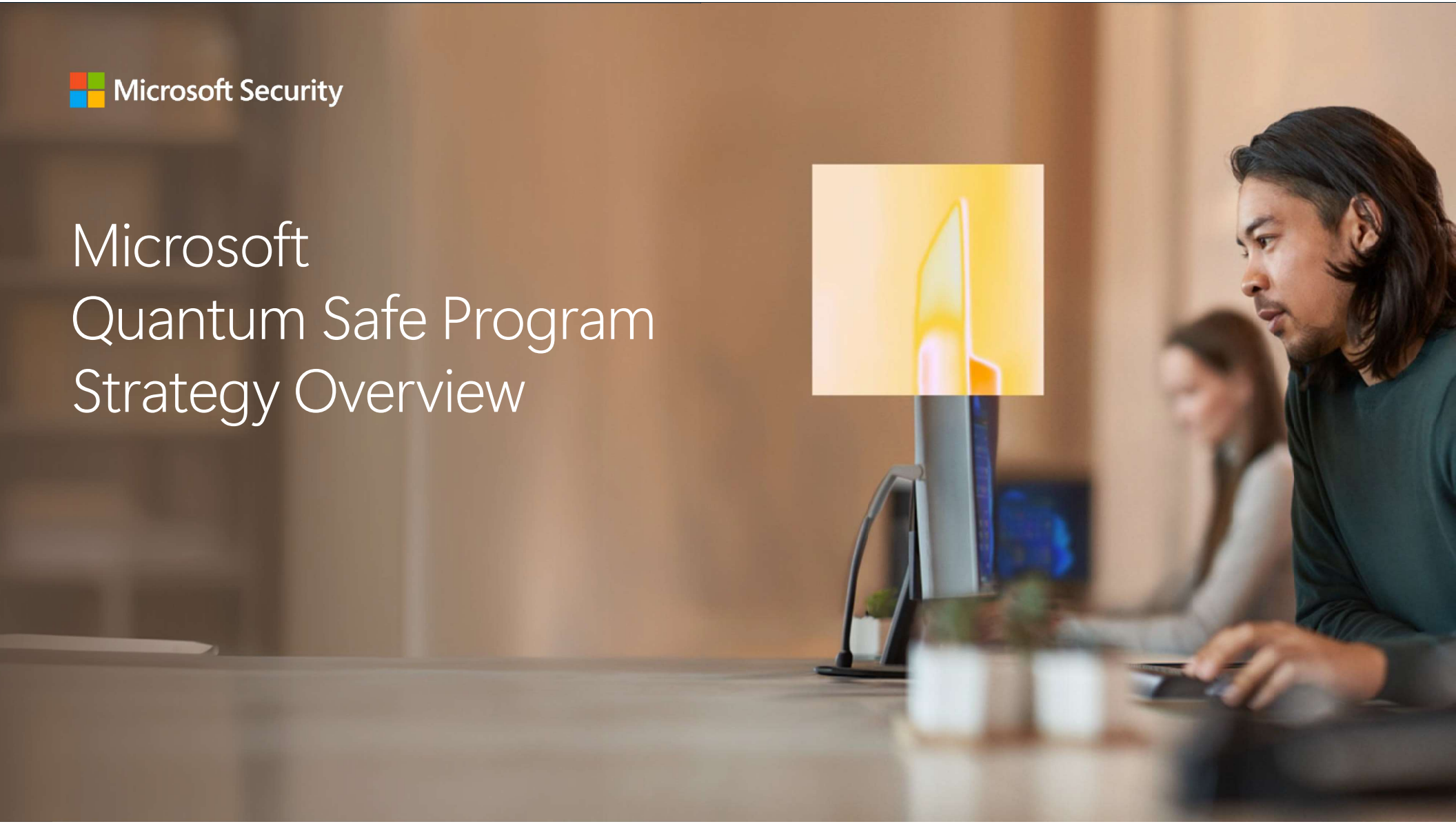

Microsoft Security
Microsoft
Quantum Safe Program
Strategy Overview

Michael Williams

Senior Software Engineer

Michael Williams is a Senior Software Engineer on the Microsoft Azure Storage team with 6 years of experience supporting and securing cloud-scale storage services. He currently leads post-quantum cryptography (PQC) efforts for Azure Storage to help prepare the platform for next-generation cryptographic threats.

Bob Peters

Senior Service Engineer

Bob Peters is a Senior Service Engineer in Azure Storage focused on security strategy and risk reduction at cloud scale. He helps teams translate evolving security standards into actionable plans, with an emphasis on encryption governance and next-generation cryptographic risk.

Sections

1 Quantum computing

2 Challenges of becoming quantum safe

3 Microsoft QSP strategy update

4 Building your Quantum Safe Program

Terminology

Quantum ready

Term used for discussion of *developing and implementing* quantum computing capabilities

Quantum computer

Using quantum mechanics, computational units are known as Qubits

Quantum safe

Term used for the discussion of *security topics* specific to *quantum safe cryptography*

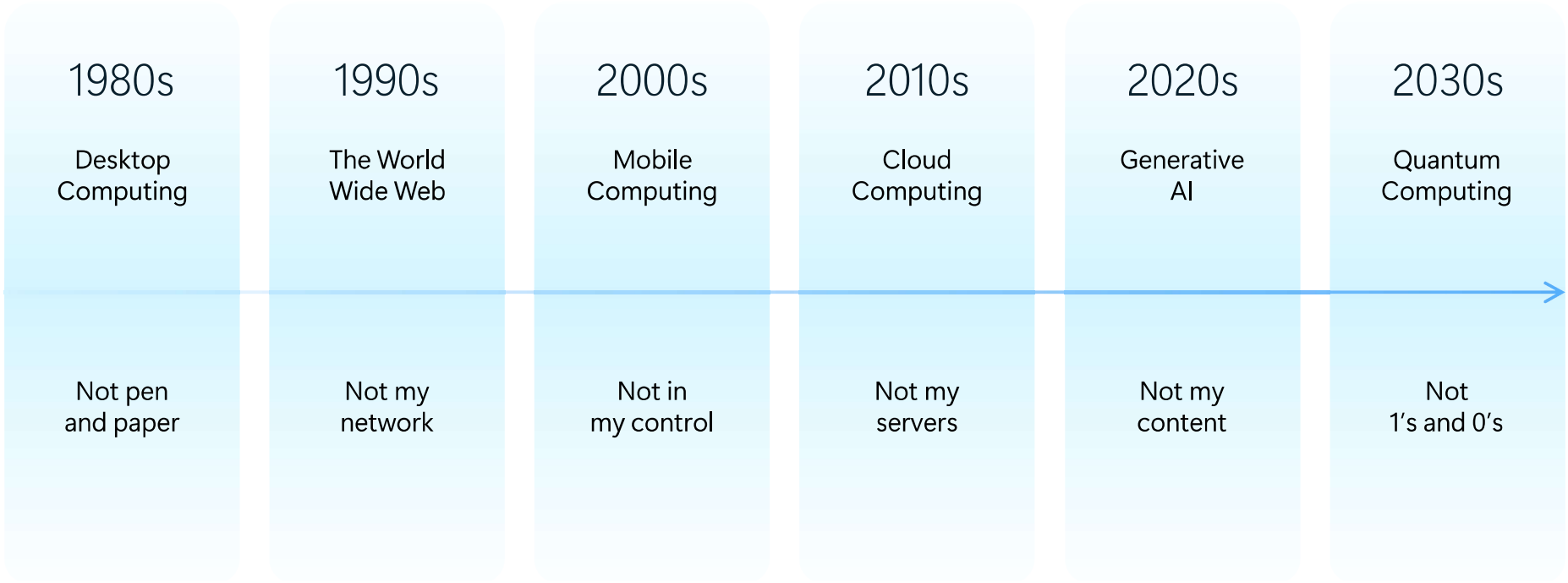
Post quantum cryptography

Quantum resistant algorithms applying mathematical challenges to reduce the risk of quantum computing capabilities


Quantum computing
is a game changer
for humanity

QUANTUM COMPUTING

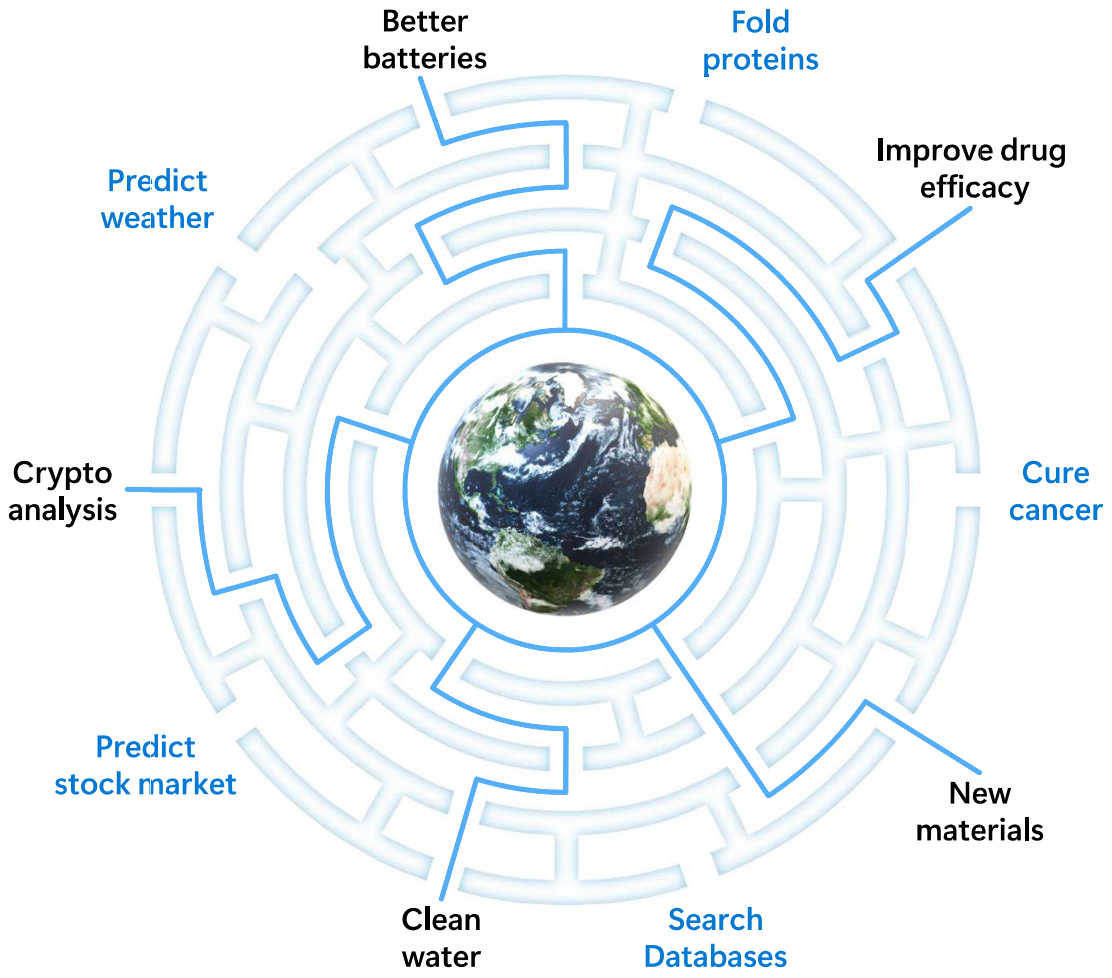
Shifts in technology capabilities bring new challenges and opportunities



Majorana 1
The first QPU powered by topological qubits

Read the research papers: Nature paper & ArXiv

Quantum computers will excel at solving **small-data problems**



Quantum Risk

- Shor's Algorithm provides a solution to the integer factorization problem, a difficult task for classical that lies at the heart of public-key cryptography, including the pervasive RSA and ECDH algorithms.
- **Using Shor's algorithm, a 2048-bit RSA encryption key can be broken in 8 hours, a feat that would take ~300 trillion years on a classical computer!**
- Quantum ready "Q-day" estimated in 5-15 years

Quantum Safety

Standards

- FIPS: Federal Information Processing Standards
- NIST: National Institute of Standards and Technology
- CNSA: Commercial National Security Algorithm Suite

PQC algorithms standardized in late 2024

- FIPS203: Module-Lattice Based Key Encapsulation Mechanism (ML-KEM) for public key encapsulation and key exchange
- FIPS204: Module-Lattice Based Digital Signature Algorithm (ML-DSA) for verifying identity, integrity, or authenticity using digital signatures

Quantum Safety

Security foundation

- Introduce key encapsulation for shared secret
- Backed by computational difficulty of solving Module Learning With Errors (MLWE) problem
- Given a set of linear equations with small random errors (“noise”), recover the secret values (The Learning with Errors Problem - Oded Regev)
- Added noise makes the problem extremely hard, even with a quantum computer

"Hybrid" solution

- Combine PQC algorithms (ML-KEM or ML-DSA) with classical algorithms (ECDH, RSA, etc.)
- Provides post-quantum safety, while maintaining compatibility
- Defense in depth

Main challenges of becoming quantum-safe

1. Inventorying and governing cryptographic systems
2. Protecting sensitive data at rest, in use, and in transit
3. Maintaining visibility into encrypted network traffic
4. Modernizing tools and systems for TLS 1.3 adoption
5. Managing performance impacts from larger algorithms

Challenge 1

Inventorying and governing cryptographic systems

As cryptographic assets and algorithms evolve, PKI, digital certificates, and TLS become harder to inventory, govern, and modernize.

1. Track and document cryptographic systems and algorithms
2. Ensure organization-wide compliance
3. Improve governance without slowing delivery

Proposed solution

1. Map cryptographic usage and risk across critical business units
2. Invest in Cryptographic Posture Management tooling
3. Modernize systems for future crypto-agility
4. Align asset inventory and key rotation with SFI guidance

Challenge 2

Protecting sensitive data at rest, in use, and in transit

1. **Credential exposure**
   Rotate credentials and minimize unauthorized access without disrupting operations.

2. **Data at rest and in use**
   Protect sensitive documents across devices, users, and environments.

3. **Data in transit**
   Secure live communications with encryption, authentication, and data leak prevention.

Proposed solution

1. Use agile secret management through Azure Key Vault (AKV) and Hardware Security Modules (HSMs)
2. Standardize storage encryption with AES-256
3. Adopt TLS 1.3 for data in transit
4. Shift inspection to endpoints and edge controls
5. Use document-level encryption and DLP where possible

Challenge 3

Maintaining visibility into encrypted network traffic

1. TLS 1.3 helps reduce “harvest now, decrypt later” exposure
2. Encrypted packets limit visibility for network-based inspection tools
3. Endpoint, behavioral, and zero-trust controls are needed to maintain security visibility

Proposed solution

1. Shift inspection closer to endpoints and edge controls
2. Use Cloud Access Security Broker (CASB)/Secure Access Service Edge (SASE) to enforce policy and maintain visibility
3. Invest in behavioral analytics and endpoint detection

Challenge 4

Modernizing tools and systems for TLS 1.3 adoption

1. Existing security inspection tools may need upgrades
2. Teams must balance cost, complexity, and security benefit
3. Legacy systems may require phased transition

Proposed solution

1. Adopt PaaS and SaaS solutions over IaaS where possible
2. Decommission legacy systems in favor of future-proof options
3. Increase isolation of systems that cannot be upgraded in the short to medium term, 2030–2035

Challenge 5

Managing performance impacts from larger algorithms

1. **Hybrid cryptography is a transition strategy**
   Organizations should plan for eventual migration to pure PQC where required.

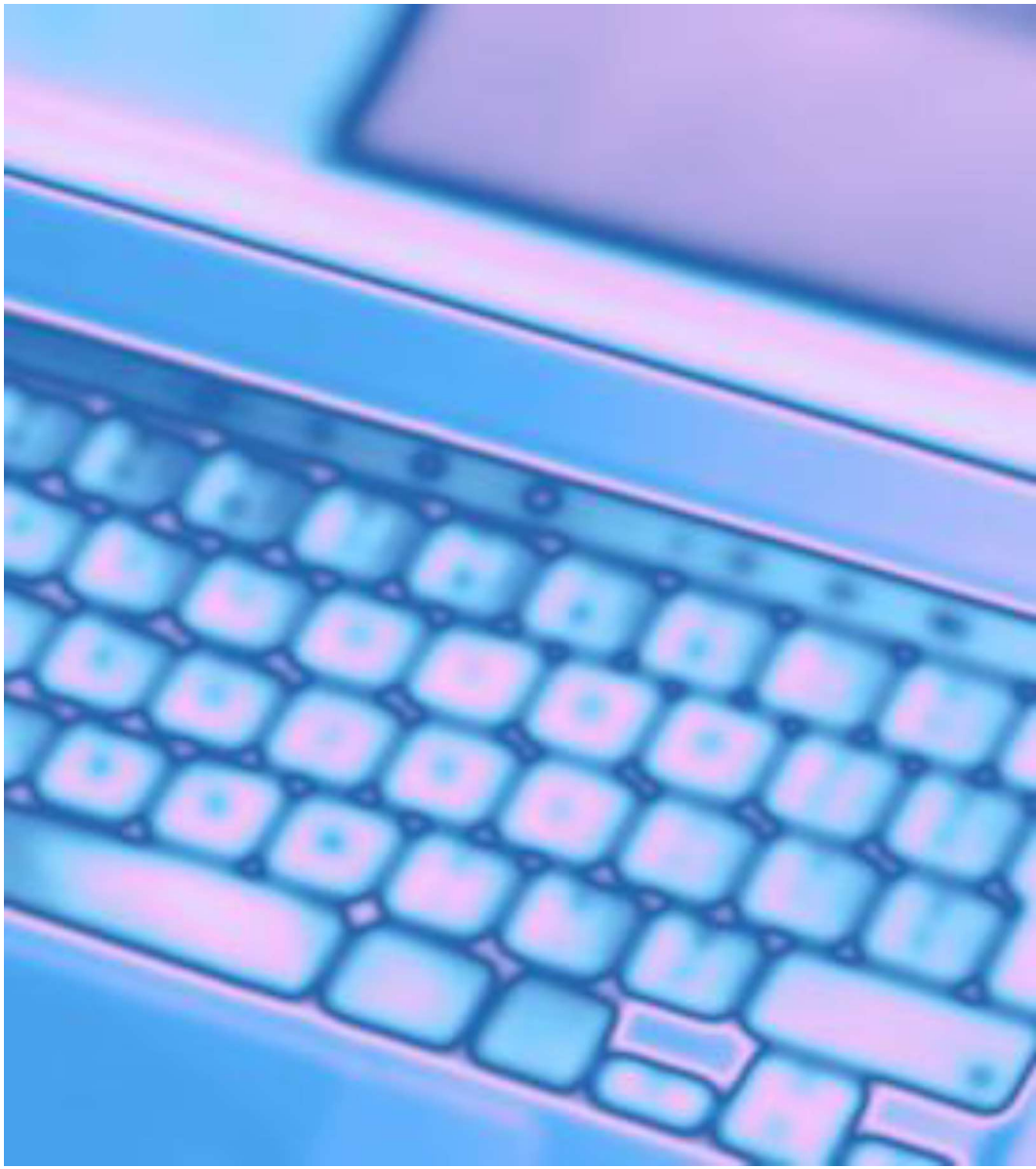
2. **Applications may need refactoring**
   PQC adoption may require code, library, protocol, and dependency updates.

3. **Hardware may not be future-ready**
   Devices purchased today may not support PQC requirements within 5–10 years.

Proposed solution

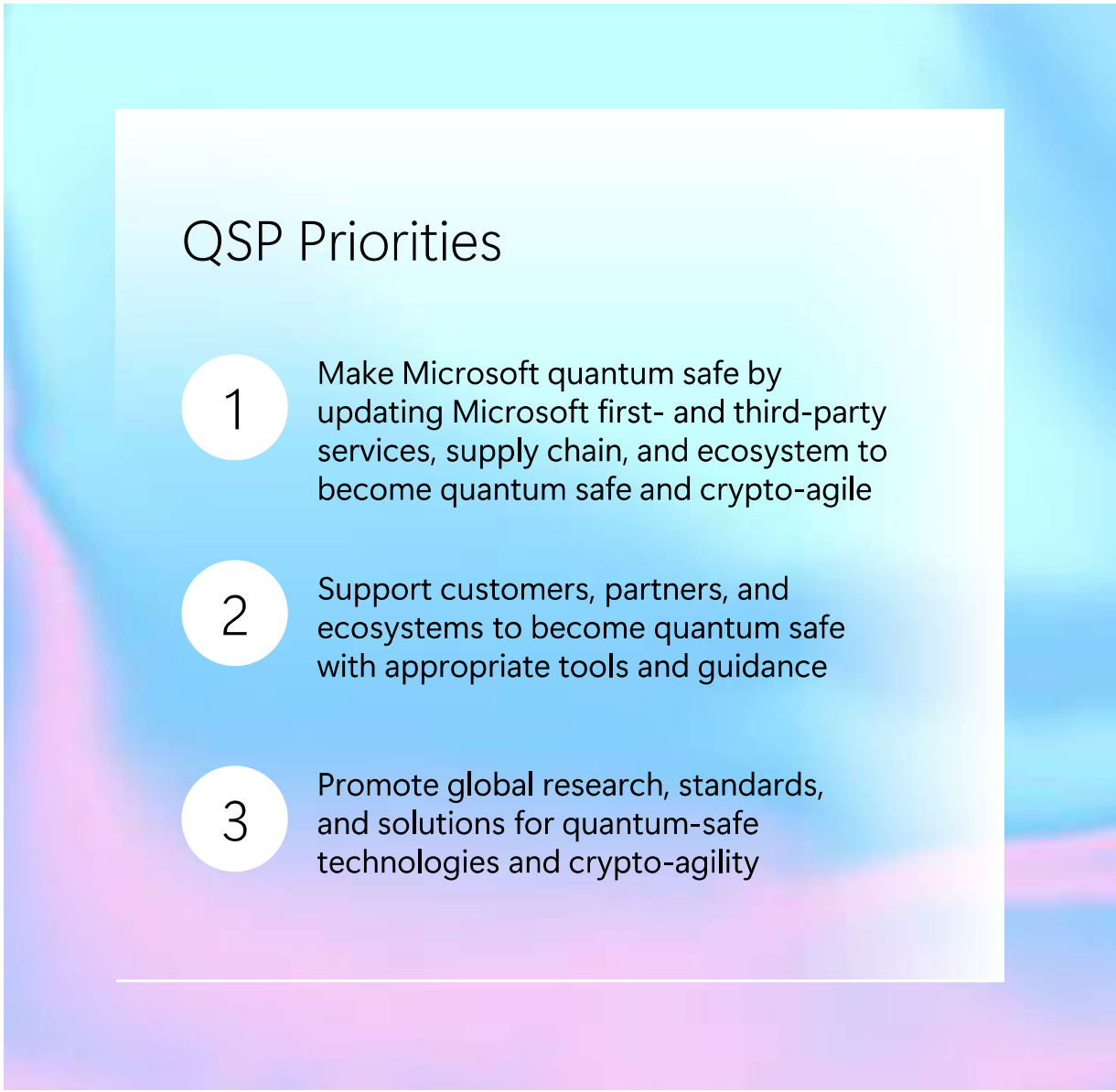
1. Prefer PaaS/SaaS platforms with vendor-managed PQC roadmaps
2. Modernize or decommission systems that cannot support crypto-agility
3. Benchmark PQC performance impacts before broad adoption
4. Isolate systems that cannot be upgraded within the 2030–2035 migration window

Microsoft QSP strategy update



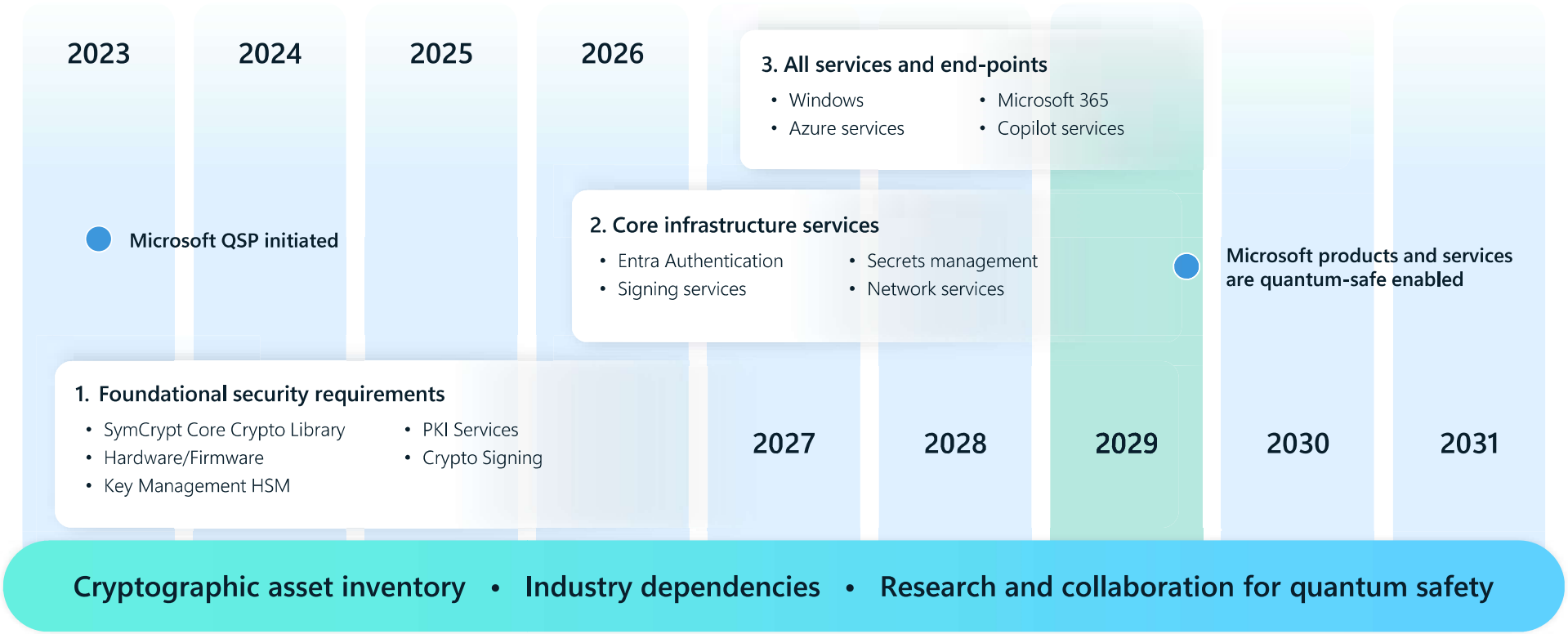
Microsoft Quantum Safe Program (QSP)

The QSP is a comprehensive and company-wide effort to enable Microsoft, our customers, and partners, to transition smoothly and securely into the quantum era.



Quantum Safe Timeline

Microsoft QSP strategy and timeline



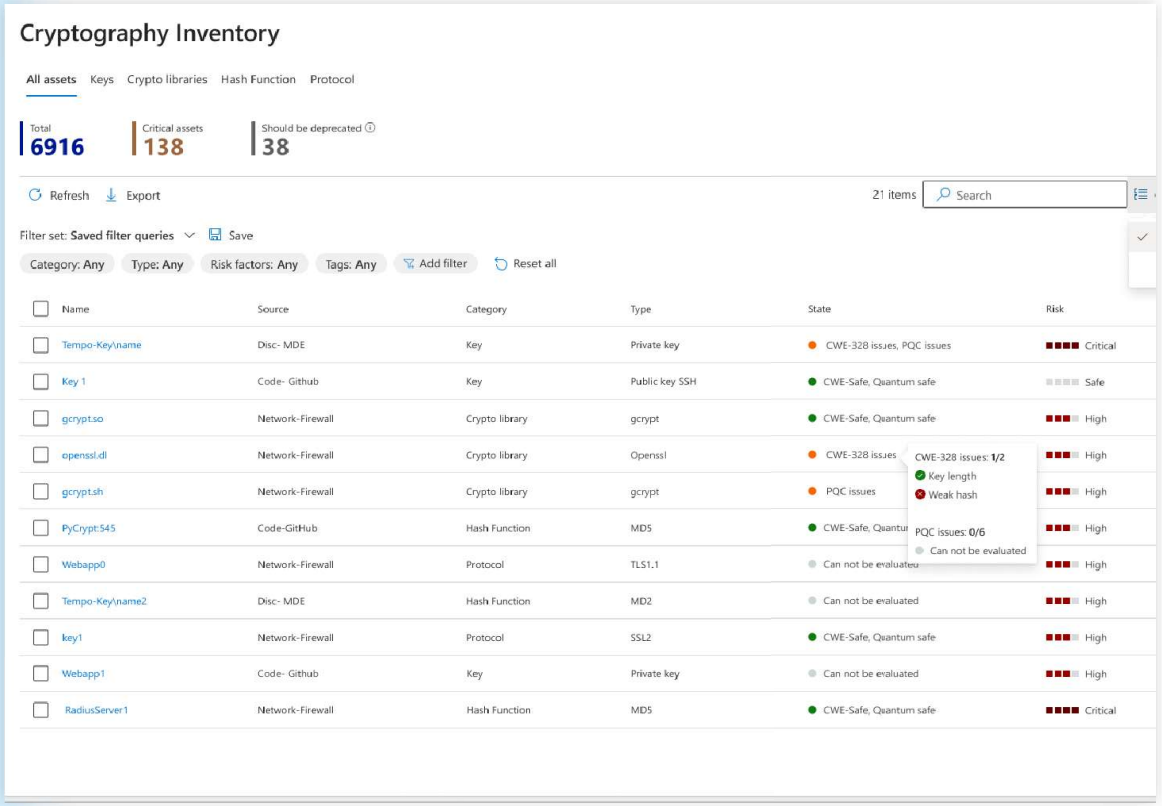
Cryptographic Posture Management (CPM)

Microsoft is building a **quantum-safe partner ecosystem** on Azure to assist customers in assessing their cryptographic inventory.

CPM tools assist service owners with their quantum safe journey

1. Identifying cryptographic assets across Network, Storage, Runtime, and Code environments
2. Developing cryptographic inventory and remediation frameworks to address current and emerging cryptographic gaps
3. Leveraging existing security tools and sensors for effective management of cryptographic assets

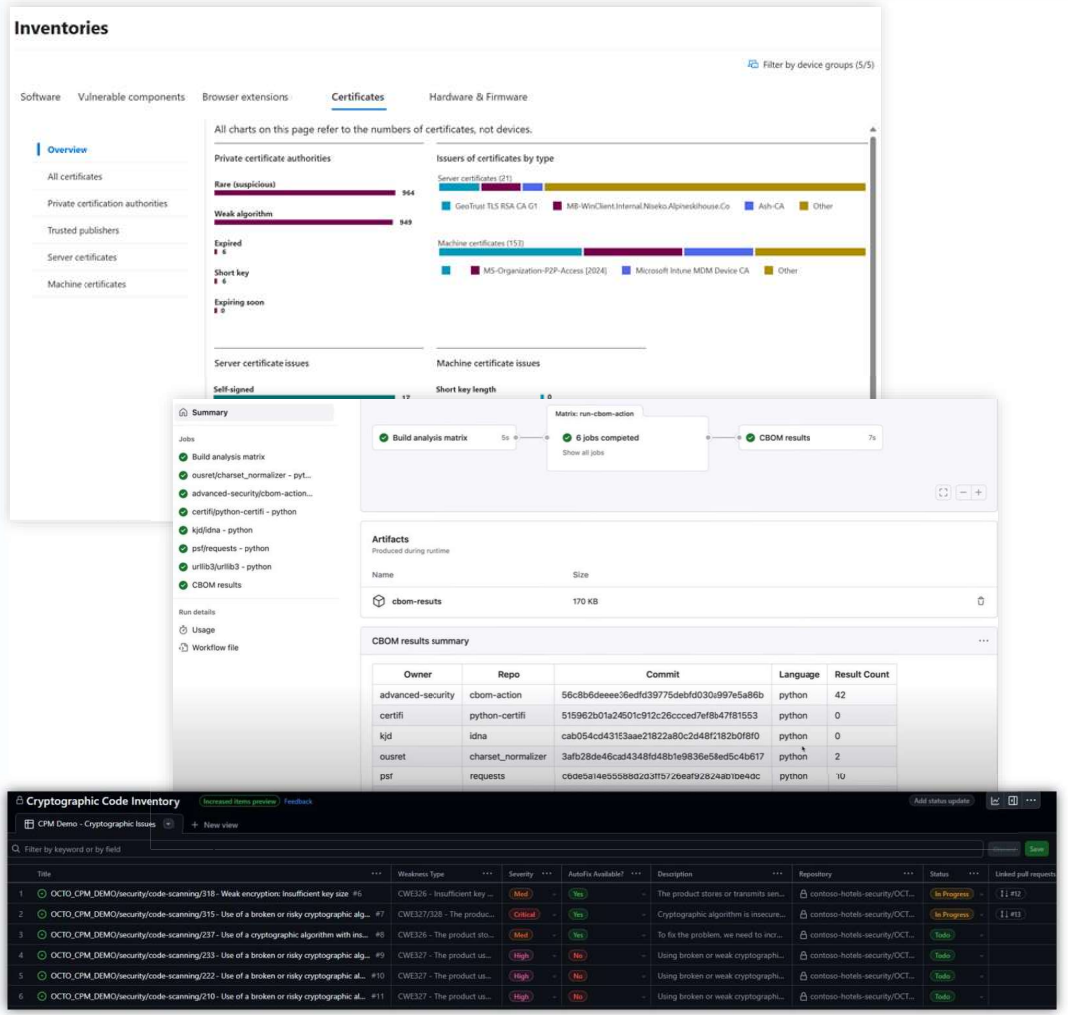
CPM tool example



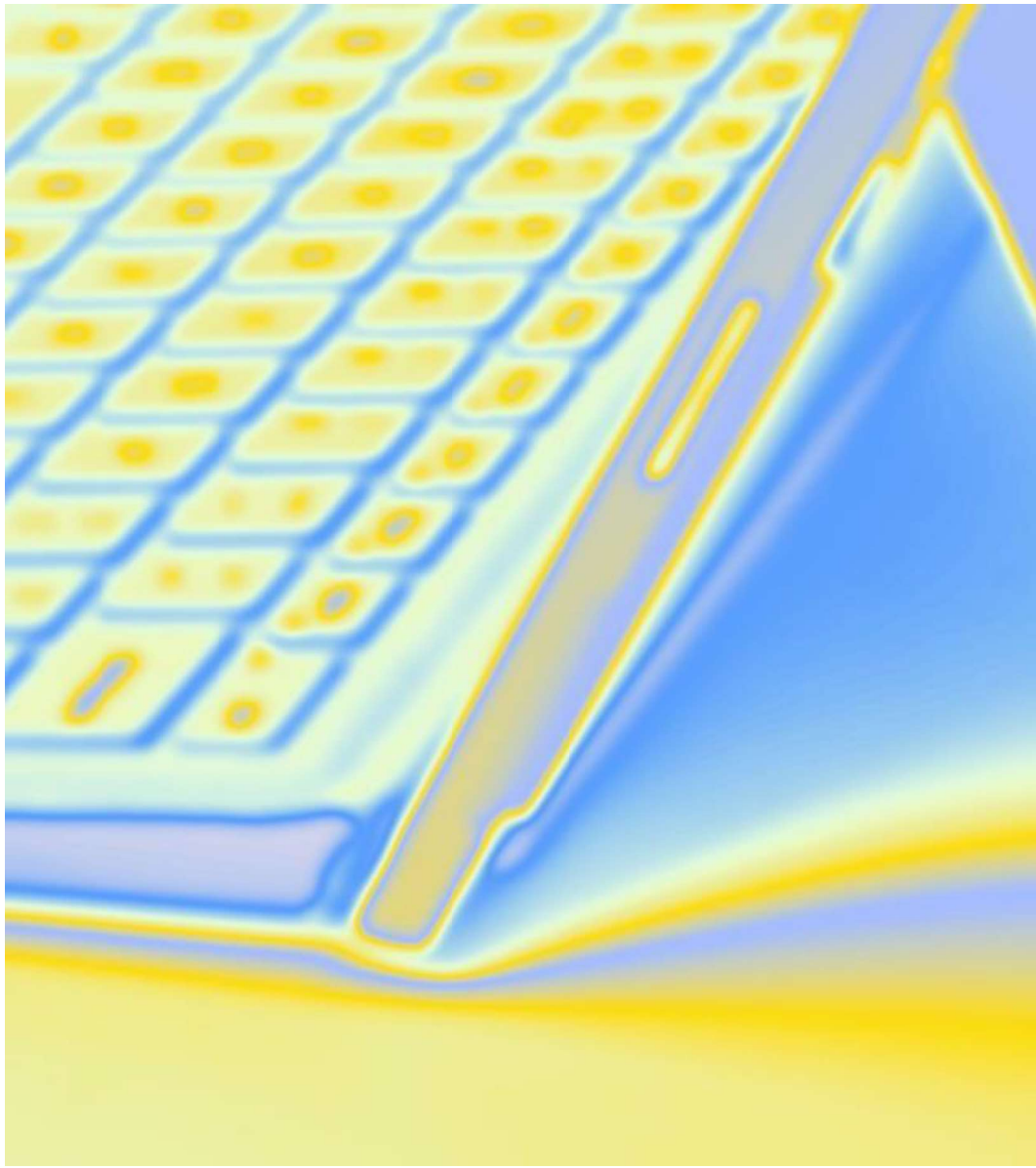
Signals to Enhance Your Cryptographic Inventory

Signals can be integrated into partner ecosystem CPM solutions or leveraged directly by customer

- Leverage CodeQL to detect cryptographic algorithm artifacts within code
- Obtain certificate details from any device using Microsoft Defender Vulnerability Management (MDVM)
- Detect encrypted traffic sessions, primarily TLS and SSH, through Defender for Endpoint (MDE) Network Detection
- Access comprehensive key and type information for secrets stored in Azure Key Vault



Key principles to build your Quantum Safe Program



The sooner organizations start their quantum-safe transformation, the safer they will be.

Systems are vulnerable already today

1. Encrypted data can be recorded and stored today to be decrypted in the future (harvest now, decrypt later)
2. Hard-to-update systems with long deployment lifetimes

Multi-year and complex transition

1. The scale of reaching quantum-safety is pervasive, spanning multiple security layers.
2. Inventorying and updating all asymmetric cryptography usage is a complex and expensive process

Your first step - crypto inventory

Prioritize the risk based on what you already know

1. Focus on cryptographic weaknesses within critical business units and environments first.
2. Spot the most vulnerable areas for harvest-now, decrypt-later (sensitive communications, critical storage devices, financial transactions)
3. Group cryptographical assets based on how they protect information in transit, at runtime, or at rest.
4. Implement a **shift-left** approach: resolve issues in design and code where possible.

Discover **storage** assets

Identify and organize certificates and keys held by KMS and HSM for secure management and auditing.

Safeguard **network** channels

Protect critical internal and external network channels to prevent unauthorized access to cryptographic assets.

Upgrade **code** early

Implement cryptographic upgrades at the start of the development cycle to improve security and efficiency.

Distinguish **runtime** encryption

Track encryption that is actively used versus dormant to maintain effective security controls.

Call to action

Modernize legacy systems

Retire, isolate, or migrate outdated systems to minimize vulnerabilities and enhance overall security posture.

Transition to cloud with PQC

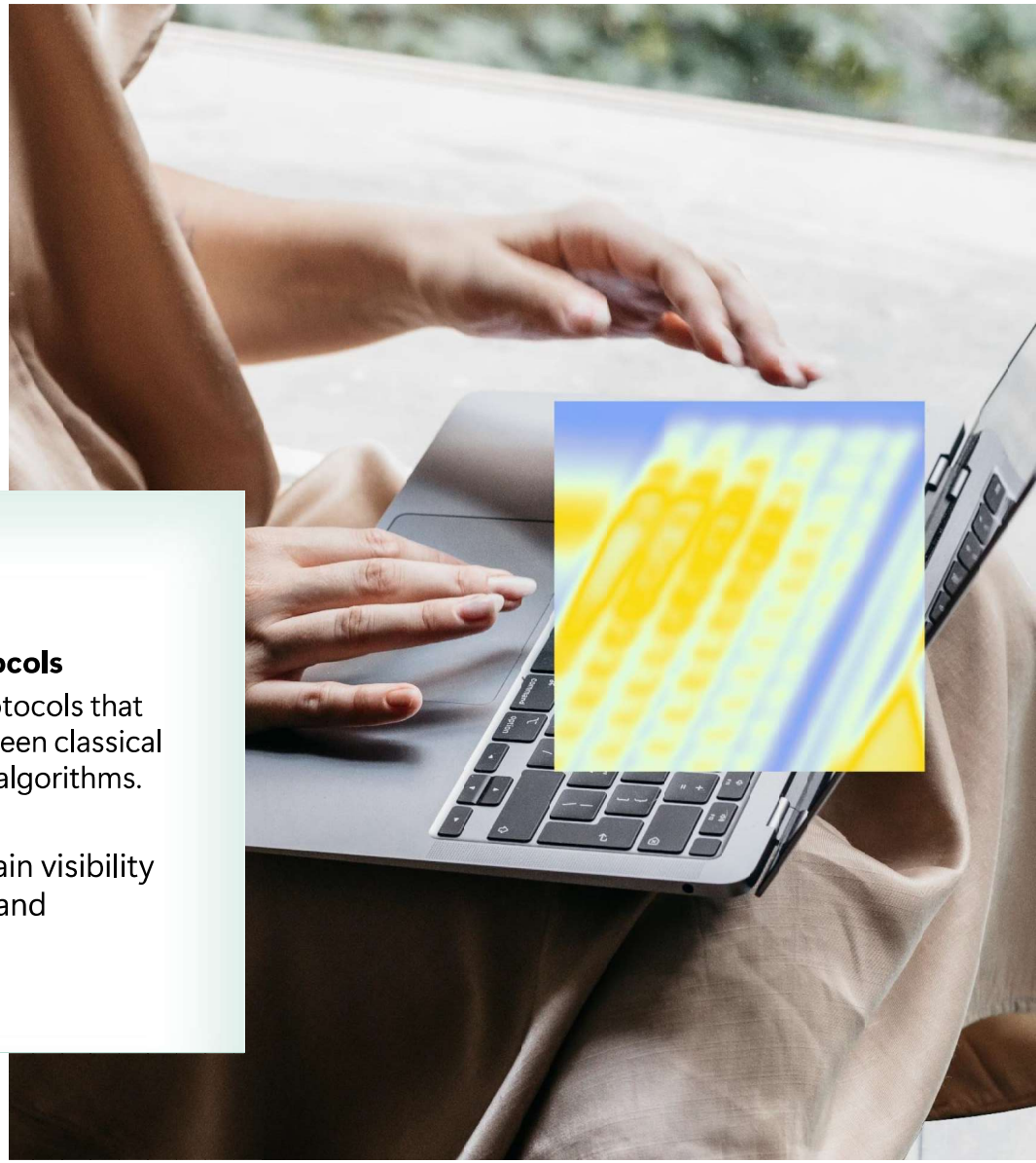
Move critical services to cloud platforms supporting PQC-ready alternatives and upgrade symmetric key sizes to AES-256 wherever you can for increased protection.

Adopt PQC algorithms

Inventory your current cryptographic risk. Implement post-quantum cryptography algorithms, prepare for refactoring costs, and rotate asymmetric keys frequently to reduce exposure.

Striving for crypto agility

Crypto Agility is the **capability to seamlessly replace and adapt cryptographic algorithms** across protocols, applications, software, hardware, and infrastructures **without disrupting operations**.



Core Components

Future-Proof Encryption
Implementing encryption methods that can adapt to new security threats posed by quantum computing.

Flexible Security Protocols
Developing security protocols that can quickly switch between classical and quantum-resistant algorithms.

Modular Architecture: Avoid hardcoded algorithms; enable plug-and-play cryptographic modules.

Live Inventory Maintain visibility into algorithms, keys, and certificates in use.

Striving for crypto agility

Why It Matters

Evolving Threats
Quantum computing and cryptanalytic advances are yet to be seen, thus can cause rapid need to update cryptography on top of what we know today threaten current cryptographic standards.

Operational Continuity
Enables rapid response to vulnerabilities without halting services.

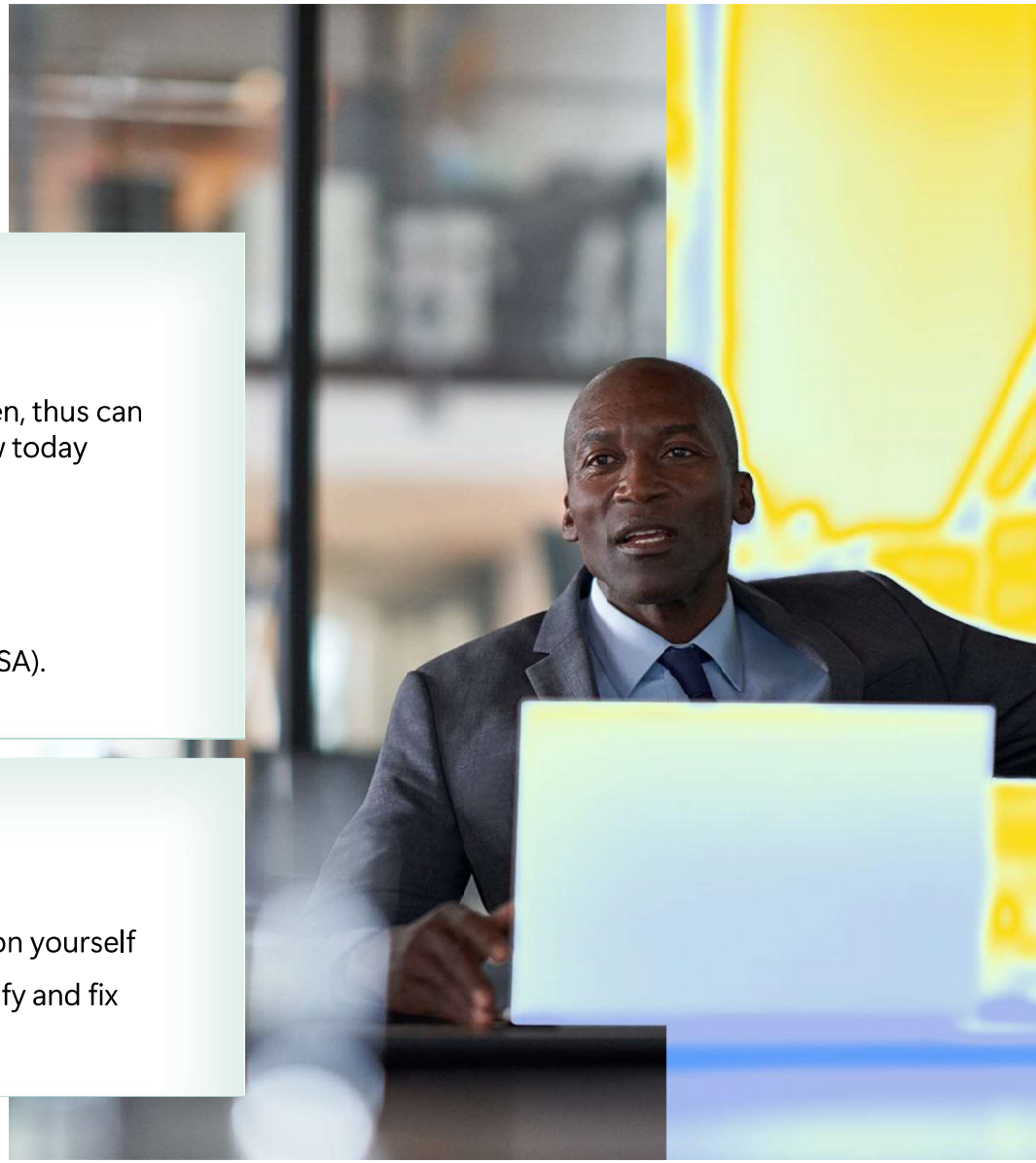
Compliance & Governance
Supports dynamic regulatory requirements (e.g., NIS2, DORA, CNSA).

Implementation Strategies

Utilize crypto agile libraries such as Microsoft SymCrypt

Move high level solutions to the cloud to avoid handling encryption yourself

Use tools like Cryptography Posture Management (CPM) to identify and fix crypto issues.



Microsoft resources

Additional resources

- Microsoft Security Blog: Progress towards next-generation cryptography
- Microsoft Quantum-Resistant Cryptography is here PQC Implementation in SymCrypt
- Generate your Cryptography Bill of Materials (CBOM) with GitHub and Microsoft CodeQL tools

Microsoft resources

Additional resources

- Cryptographic Inventory Blog: Building your cryptographic inventory: A customer strategy for cryptographic posture management | Microsoft Security Blog
- Secure Future Initiative (SFI)
  Secure Future Initiative – Secure by Design | Microsoft

Microsoft Security

Q&A