Entering to a new era of crypto engineering:

Cryptographic Visibility and Agility

Jihoon CHO (VP in Technology)
Hyojin YOON, Changhoon LEE, Eunkyung KIM, Janghyuk AHN, Hunhee YU
Samsung SDS

Questions regarding **PQC Migration**

Threat Timeline

When CRQC is available?

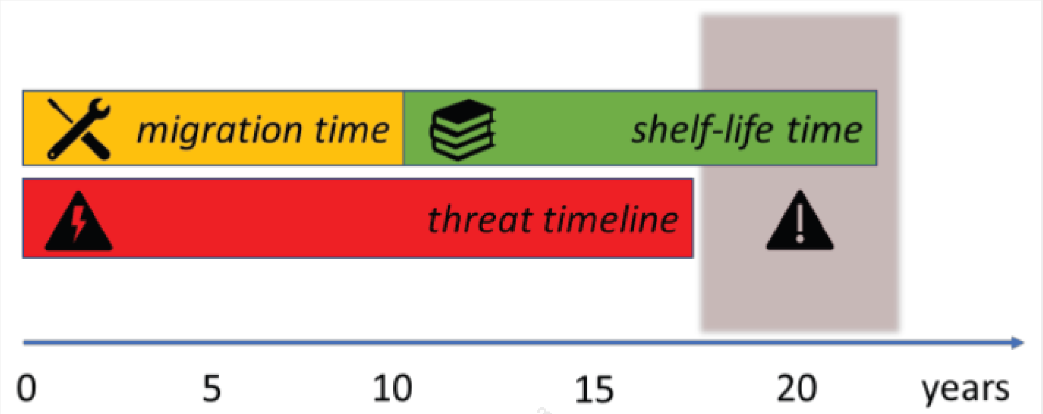
Shelf-life Time

How long data should remain protected?

Migration Time

How long does it take to migrate to PQC?

If x + y > z, you are in trouble.

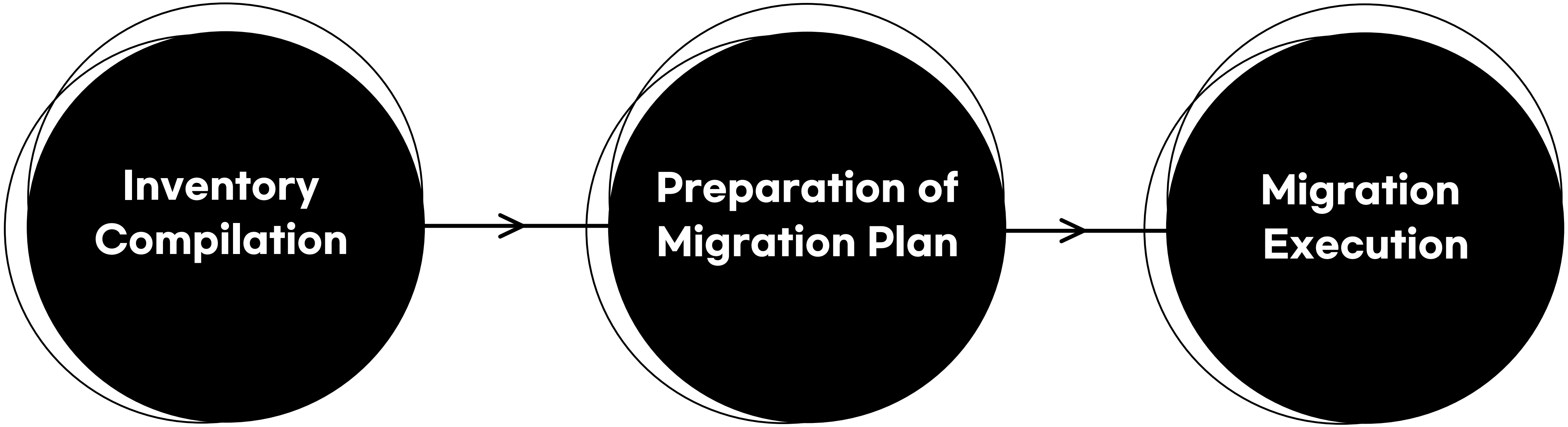


+ source: Dr. Mosca (Global Risk Institute)

- Widespread Cryptography in Enterprise IT
- Explosive Expansion of Enterprise IT
- Migration to PQC is **NOT Drop-in Replacement**

Migration Strategies and Recommendations (ETSI)

+source: ETSI TR 103 619 V1.1.1 (2020-07)



Inventory Compilation

Preparation of Migration Plan

Migration Execution

NIST's Migration to PQC Project (2022-2026)

+source. https://www.nccoe.nist.gov/crypto-agility-considerations-migrating-post-quantum-cryptographic-algorithms

Goal

"Initiating the **development of practices to ease migration** from the current set of public-key cryptographic algorithms **to replacement algorithms (PQC)** that are resistant to quantum computer-based attacks"

Consortium participants: AWS, Cisco, CISA, Cloudflare, Crypto4A Technologies, CryptoNext Security, Data-Warehouse GbmH, Dell Technologies, DigiCert, Entrust, HP, IBM, Information Security Corporation, InfoSec Global, ISARA Corporation, JPMorgan Chase Bank, N.A., Keyfactor, Kudelski IoT, Microsoft, NSA, Palo Alto Networks, PQShield, QuantumXchange, SafeLogic, **Samsung SDS**, SandboxAQ, Santander, SSH Communications Security Corp, Thales DIS CPL USA, Thales Trusted Cyber Technologies, Utimaco, Verizon, VMware, wolfSSL

① Inventory Compilation

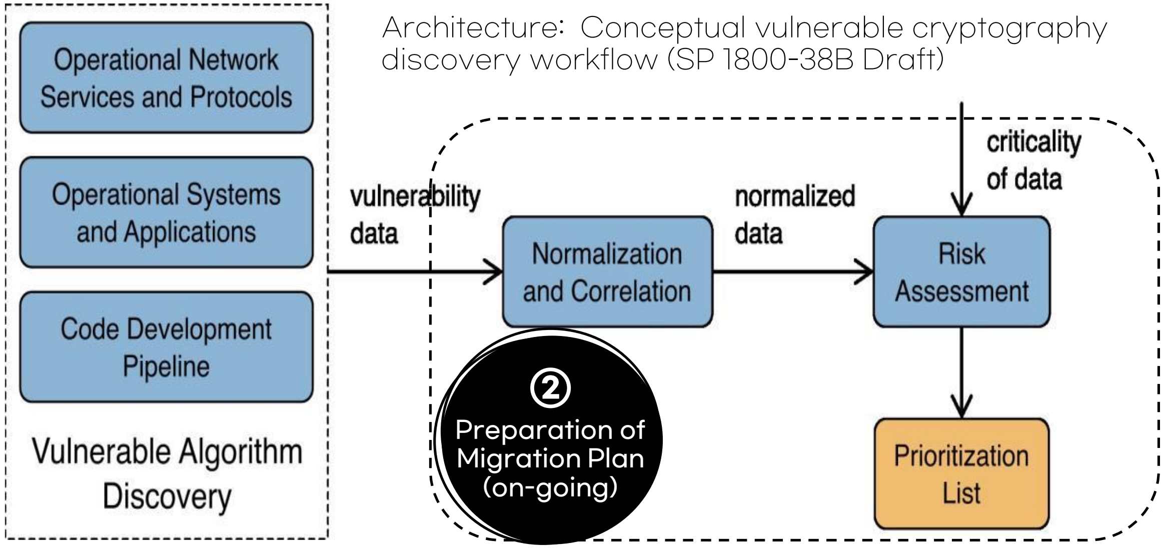
SP 1800-38B (Preliminary Draft):

Approach, architecture, and security characteristics of public key application **discovery** tools

③ Migration Execution

SP 1800-38C (Preliminary Draft):

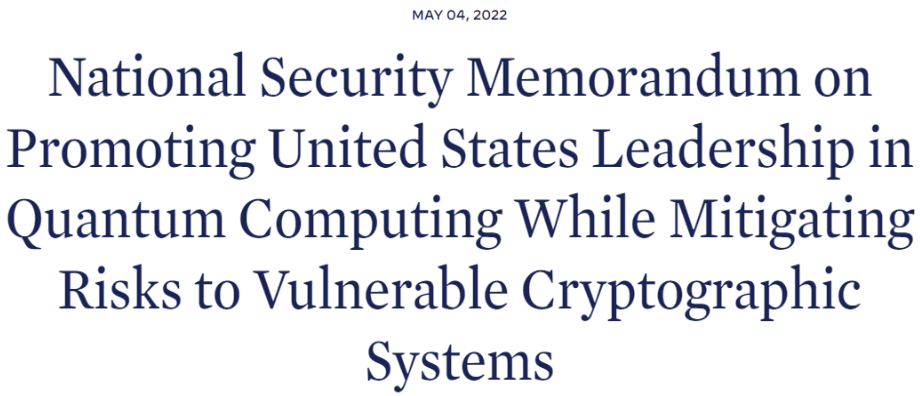
Quantum-resistant cryptography technology **interoperability** and **performance** report


Operational Network Services and Protocols
Operational Systems and Applications
Code Development Pipeline
Vulnerable Algorithm Discovery
vulnerability data
Normalization and Correlation
normalized data
criticality of data
Risk Assessment
Prioritization List
② Preparation of Migration Plan (on-going)

Architecture: Conceptual vulnerable cryptography discovery workflow (SP 1800-38B Draft)

Cryptographic Agility: Key Element to Migration Effort

- **Cryptographic agility** reduces the time to transition and allows for seamless updates for future crypto standards,
- and it is a **Design Feature**.



MAY 04, 2022

National Security Memorandum on Promoting United States Leadership in Quantum Computing While Mitigating Risks to Vulnerable Cryptographic Systems

"**Central** to this **migration effort** will be an emphasis on **cryptographic agility**, both to reduce the time required to transition and to allow for seamless updates for future cryptographic standards.

...

the term **cryptographic agility** means a **design feature** that enables future updates to cryptographic algorithms and standards **without** the need to **modify or replace** the surrounding infrastructure .. "

+source: https://www.whitehouse.gov/briefing-room/statements-releases/

● After all, it is a software update in Enterprise IT for most cases.

Observation 01.

Design Feature for **DevSecOps**

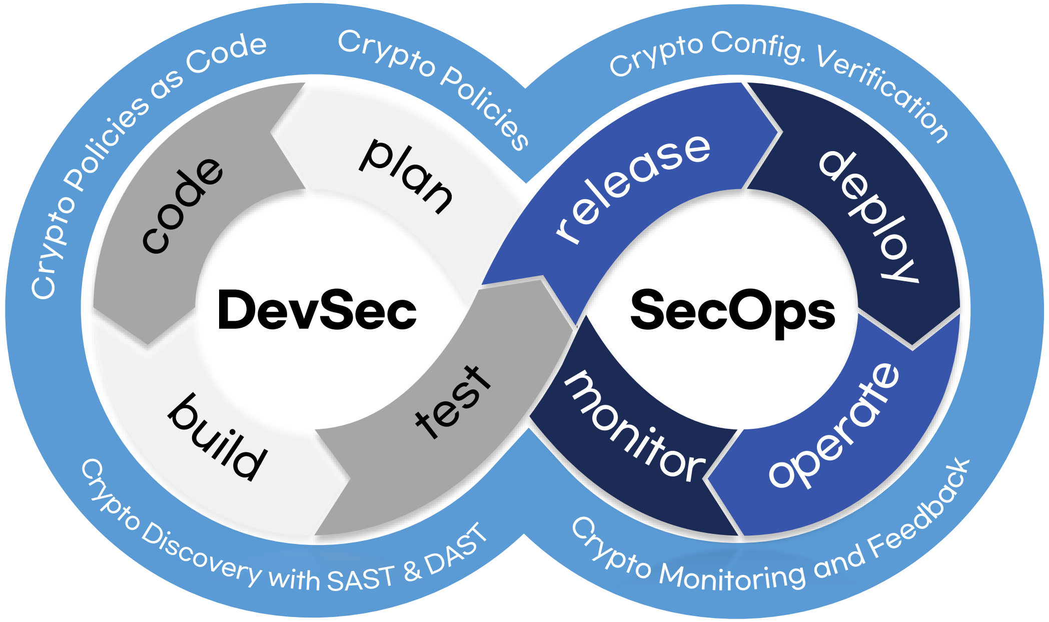
Cryptographic Agility X DevSecOps

Plan and perform PQC Migration in close alignment with the **DevSecOps**

- **DevOps** is an SDLC management methodology, shifting one large release to agile and continuous integration, and continuous delivery & deployment (CI/CD)

- **DevSecOps** automates security enforcement in CI/CD pipelines

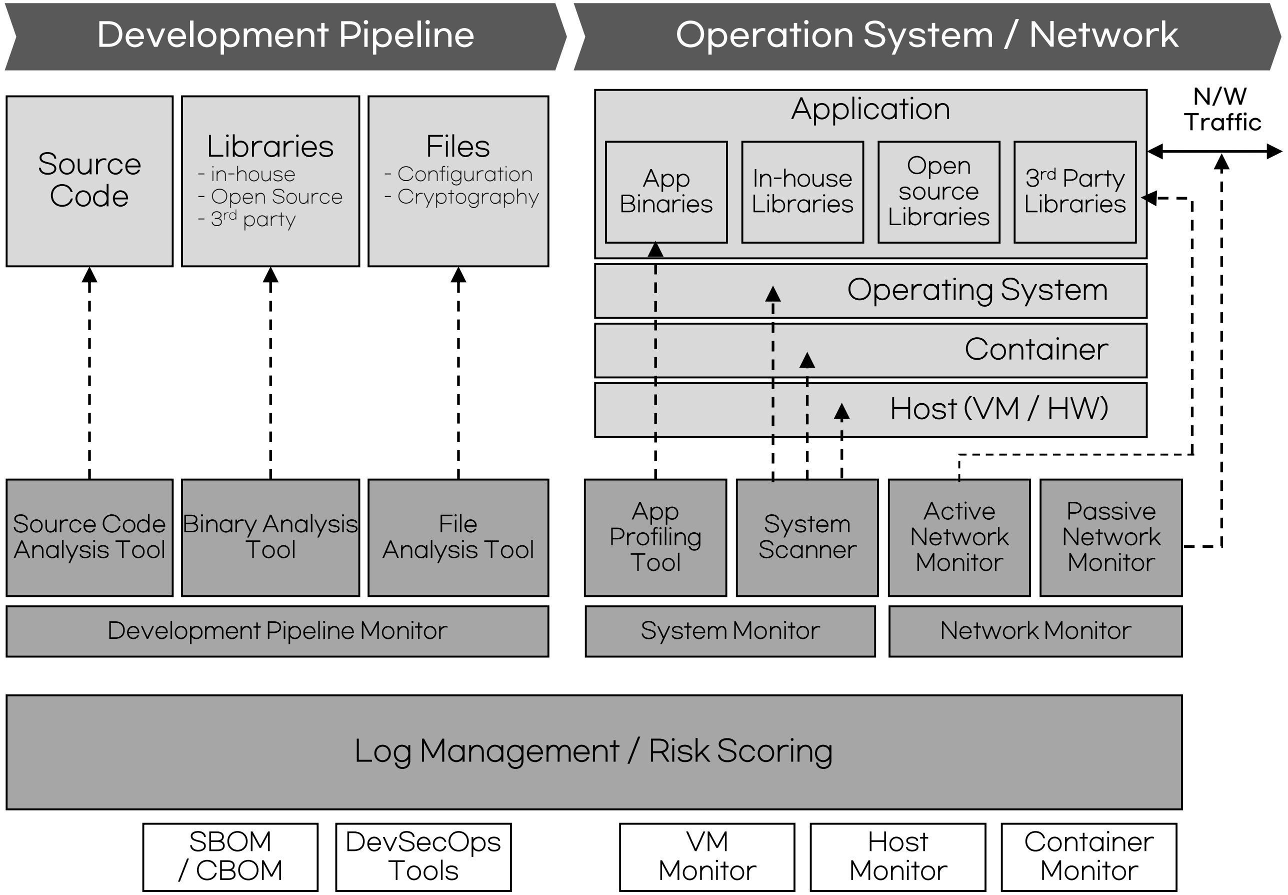
- **Cryptographic policies** can be **defined as declarative codes**, and enforced using DevSecOps platform



Cryptographic Agility into DevSecOps

Case Study 01. Cryptographic Discovery in CI/CD Pipelines

Quantum-vulnerable cryptography can be discovered using DevSecOps tools.



● Development Pipeline Analysis

- Implement plugin open-source SAST tool (e.g., spotbugs)
- Discover the use of quantum-vulnerable cryptography functions or parameters for Java Crypto API

● Operational System Analysis

- Analyze deployed modules or running process using Java built-in tools, and discover methods or classes

● Operational Network Analysis

- Perform active and/or passive monitoring by using open-source packet tools (e.g., tshark), or by running TLS clients, and discover quantum-vulnerable key exchanges

● Risk Estimation

- Estimate risk scoring based on discovered data

Case Study 02. Separating Cryptographic Configuration

JCA separates cryptographic configuration from application, and
enables to migrate to PQC by updating the configuration file without modifying application

Updating "java.security" file

Modifying the priority for crypto provider

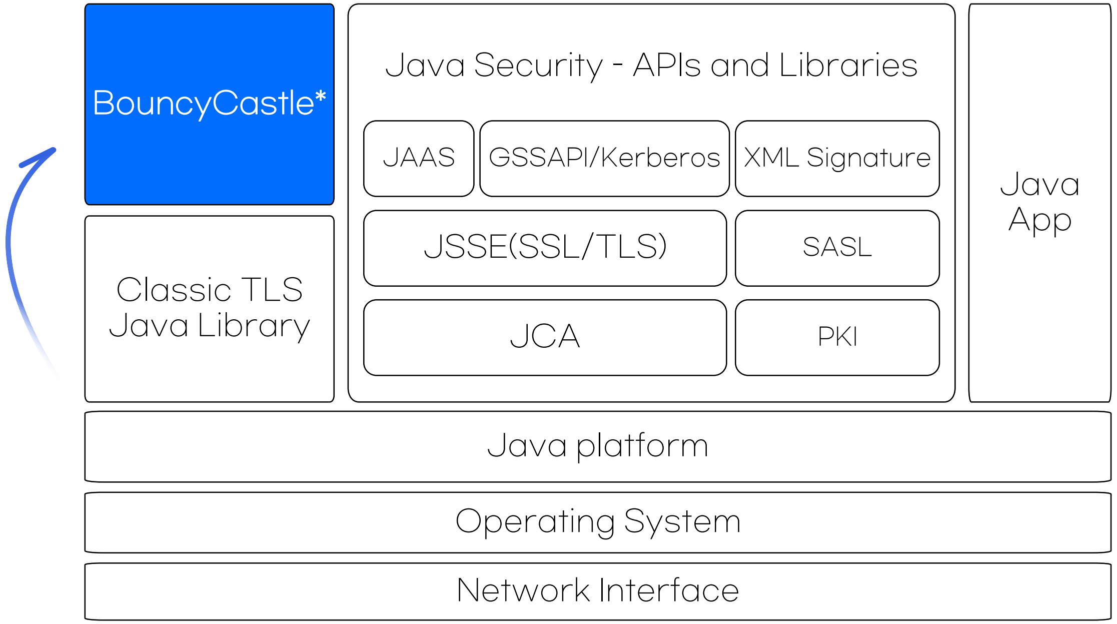
```
security.provider.1=SUN
security.provider.2=SunRsaSign
security.provider.3=SunEC
security.provider.4=SunJSSE
security.provider.5=SunJCE
```

↓

```
security.provider.1=org.bouncycastle.jce.provider.BouncyCastleProvider
security.provider.2=org.bouncycastle.jsse.provider.BouncyCastleJsseProvider
security.provider.3=SUN
security.provider.4=SunRsaSign
security.provider.5=SunEC
```

Enforcing to use hybrid key exchange

```
jdk.tls.namedGroups=secp521r1_kyber1024
```



* Samsung SDS updated BC with hybrid key exchange, and created a PR(Pull Request) to BC.

We have too many cryptographic modules in Enterprise IT
How to manage them more effective and efficient ways?

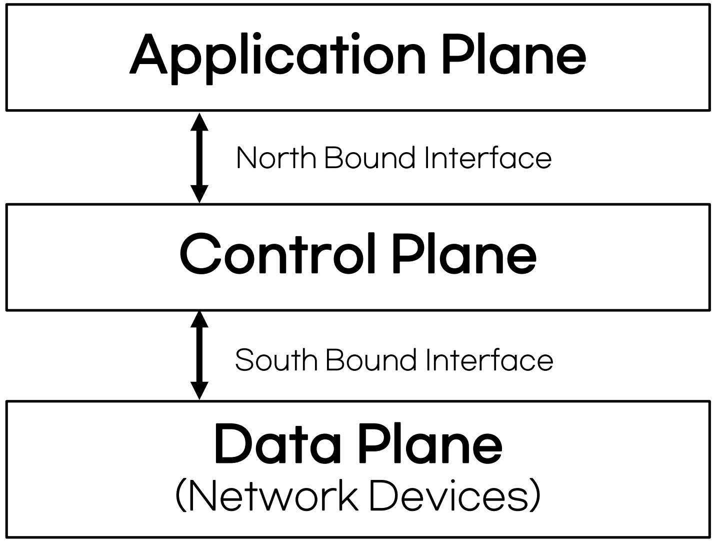
Observation 02.

Design Feature for **Software-Defined X**

Network Done IT Already!

+ source: Software-Defined Networking (SDN):
Layers and Architecture Terminology (RFC 7426)

- **Problem:** Too many network devices to manage!
- **Software-Defined Network (SDN)** emphasizes the role of software:
  - Introducing an abstraction for the data plane and, by doing so, separating it from the control plane
  - Enabling network programmability to centrally manage the behaviors of network as a whole
- As a result, **SDN** obtains **visibility** of the entire network as well as **automation** for configuration of network policies


Application Plane

North Bound Interface

Control Plane

South Bound Interface

Data Plane
(Network Devices)

Simplified SDN Architecture

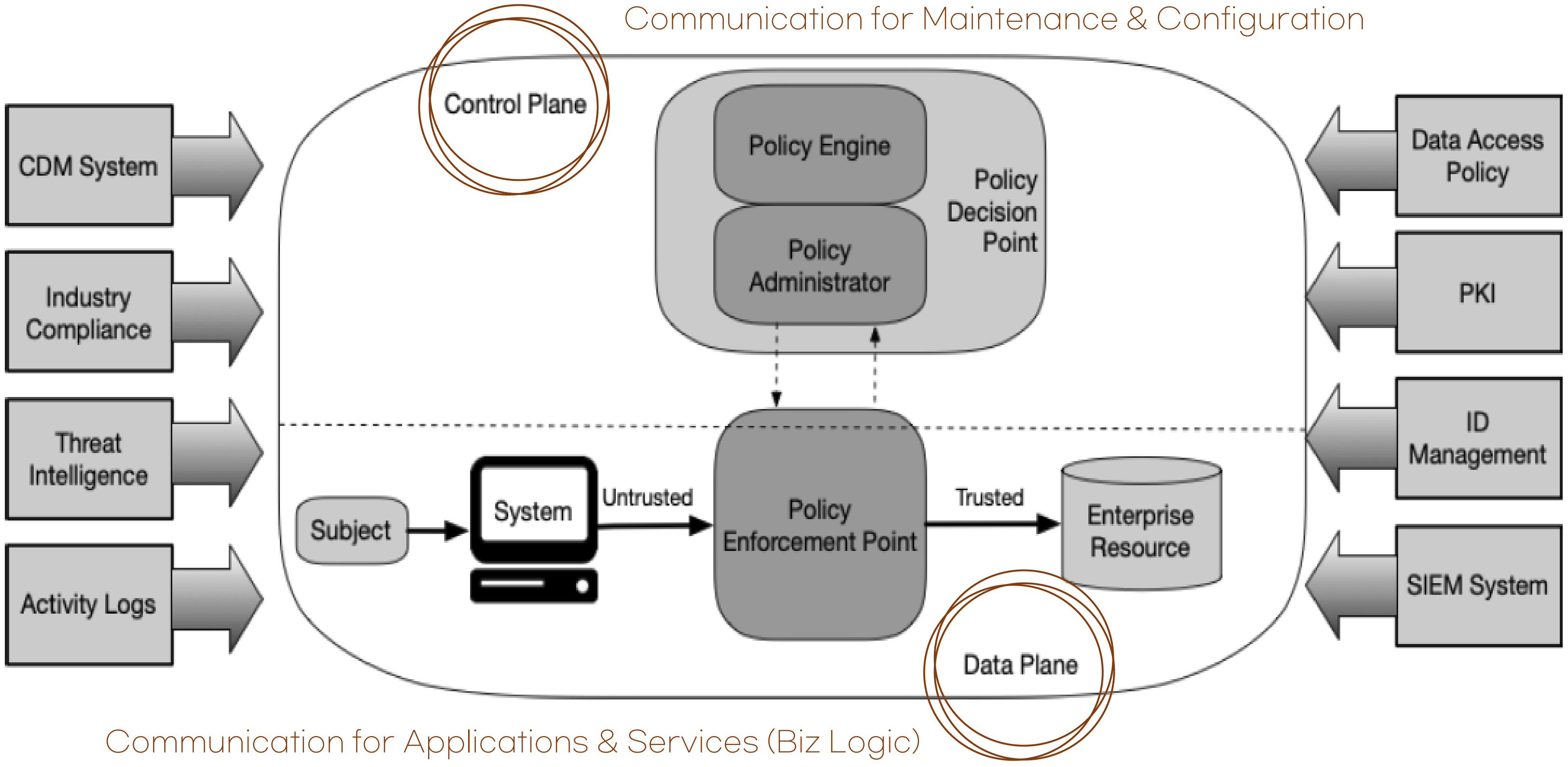
Software-Defined approach has been applied to
SW-Defined Storage, SW-Defined Data Center, SW-Defined Perimeter, and …

Zero Trust Architecture DID it.

+ source: NIST SP 800-207

"NEVER TRUST, ALWAYS VERIFY!"

Without **visibility & automation**, infeasible to enforce zero trust principles



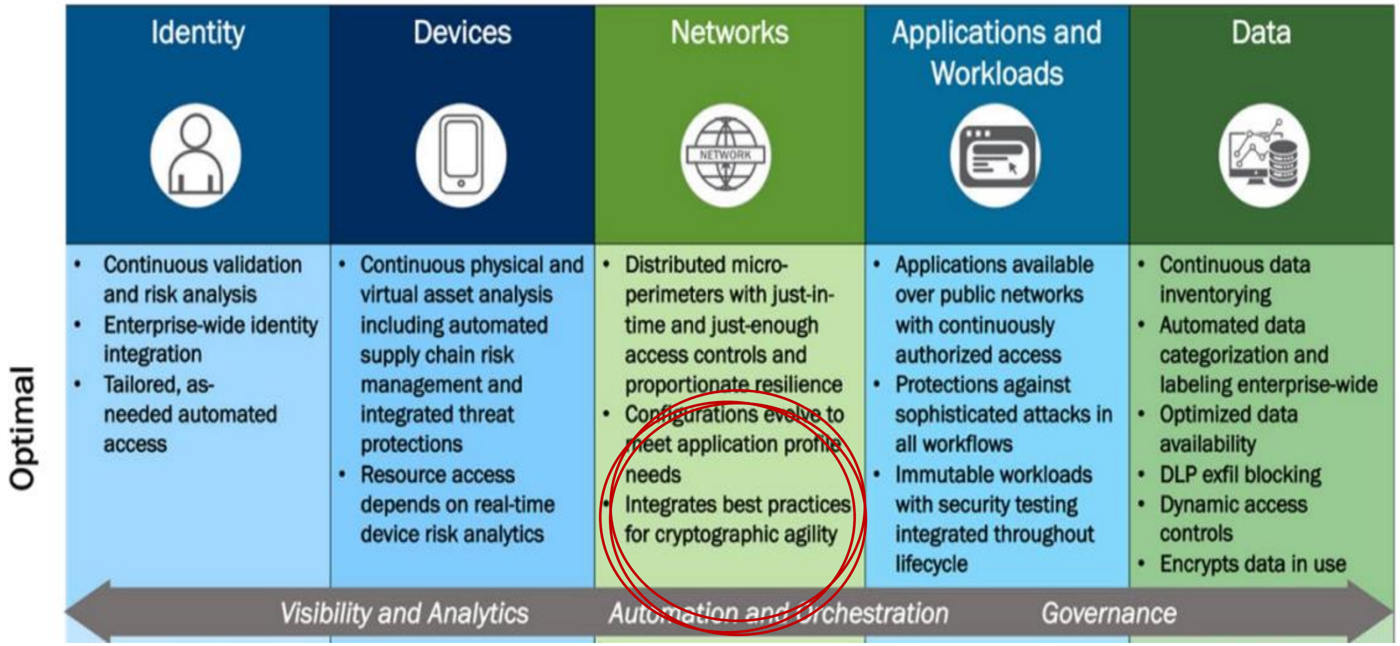
3.4.1 Network Requirement to Support ZTA (NIST SP 800-207)

"… The data plane and control plane are logically separate"

Cryptographic Agility & Zero Trust

+ Source: Zero Trust Maturity Model 2.0 (CISA)

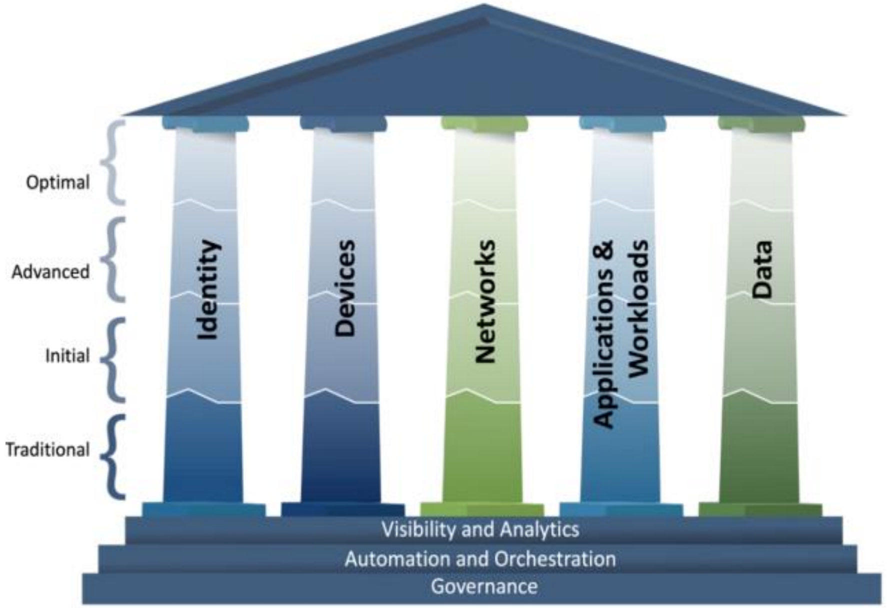
- Higher level of **automation & visibility** is required for cryptographic agility, and it could be obtained by **following SDx approaches**!



	Identity	Devices	Networks	Applications and Workloads	Data
Optimal	• Continuous validation and risk analysis • Enterprise-wide identity integration • Tailored, as-needed automated access	• Continuous physical and virtual asset analysis including automated supply chain risk management and integrated threat protections • Resource access depends on real-time device risk analytics	• Distributed micro-perimeters with just-in-time and just-enough access controls and proportionate resilience • Configurations evolve to meet application profile needs • Integrates best practices for cryptographic agility	• Applications available over public networks with continuously authorized access • Protections against sophisticated attacks in all workflows • Immutable workloads with security testing integrated throughout lifecycle	• Continuous data inventorying • Automated data categorization and labeling enterprise-wide • Optimized data availability • DLP exfil blocking • Dynamic access controls • Encrypts data in use

Visibility and Analytics | Automation and Orchestration | Governance

High-Level Zero Trust Maturity Model Overview: Cryptographic agility is required to achieve the optimal level for zero trust maturity



Zero Trust Maturity Evolution

SW-Defined Cryptography

as Design Feature of Cryptographic Agility

Service Mesh

"SW-Defined Cryptography" enables
visibility of use of cryptography &
automation of cryptographic configuration

01.

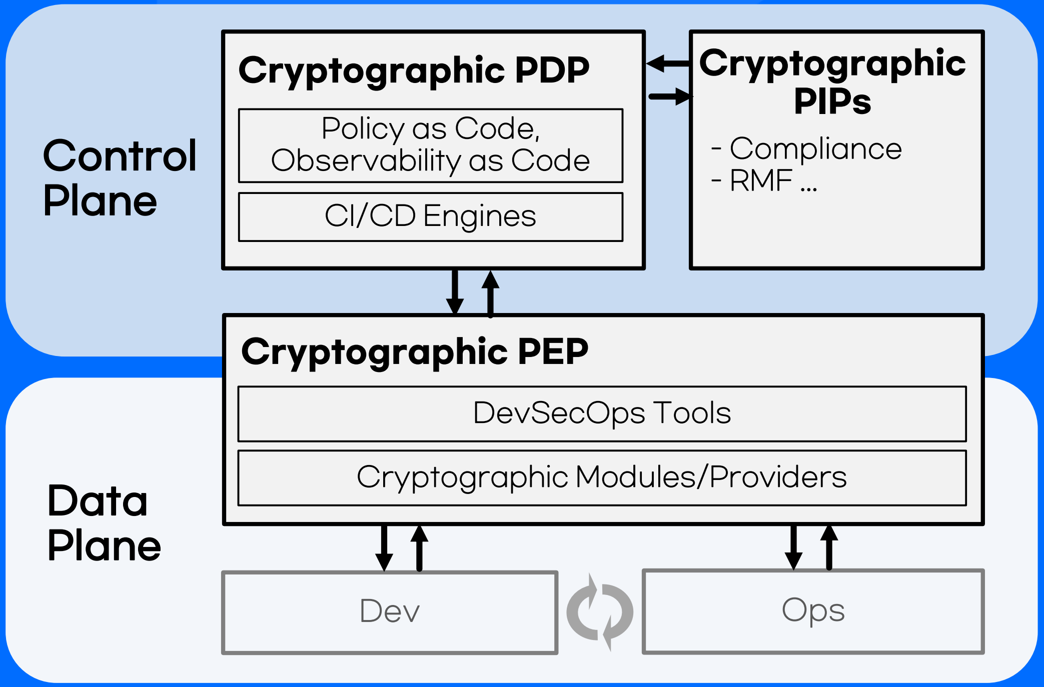
Cryptographic **policies can be defined as codes**, and enforced using **DevSecOps** platform.

Don't forget to **separate cryptographic configuration** from application.

02.

Adopt **software-defined approach** to **centrally** manage cryptography in Enterprise.

Maintain a **Cryptographic Center of Excellence (CCoE).**



Control Plane

Cryptographic PDP

Policy as Code, Observability as Code

CI/CD Engines

Cryptographic PIPs

- Compliance
- RMF ...

Data Plane

Cryptographic PEP

DevSecOps Tools

Cryptographic Modules/Providers

Dev

Ops

PIP: Policy Information Point, PDP: Policy Decision Point, PEP: Policy Enforcement Point