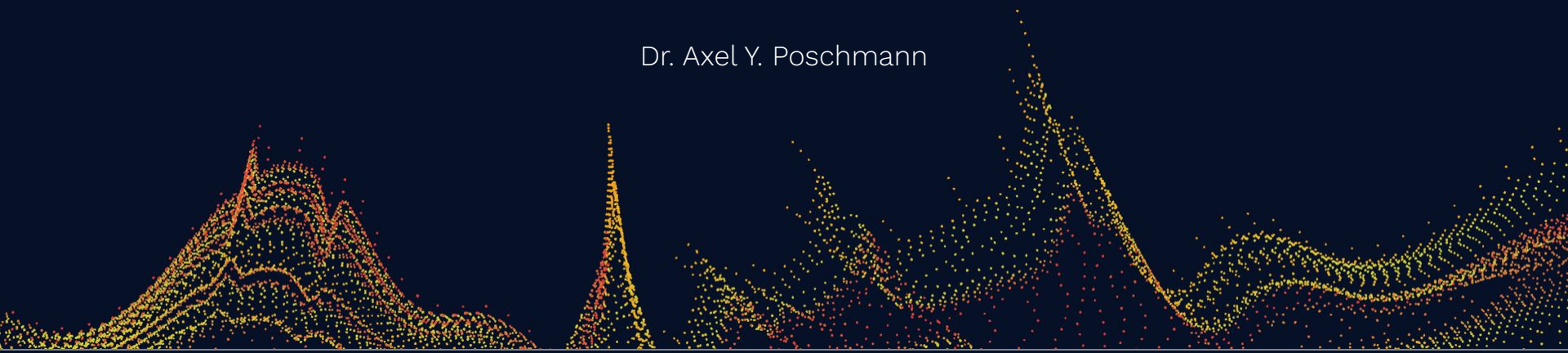
Designing a Post-Quantum Root of Trust

The Hardest Migration Problem You're Not Aware of

Dr. Axel Y. Poschmann



Speaker: about me

Dr. Axel Y. Poschmann

axel.poschmann@gmail.com

Linkedin

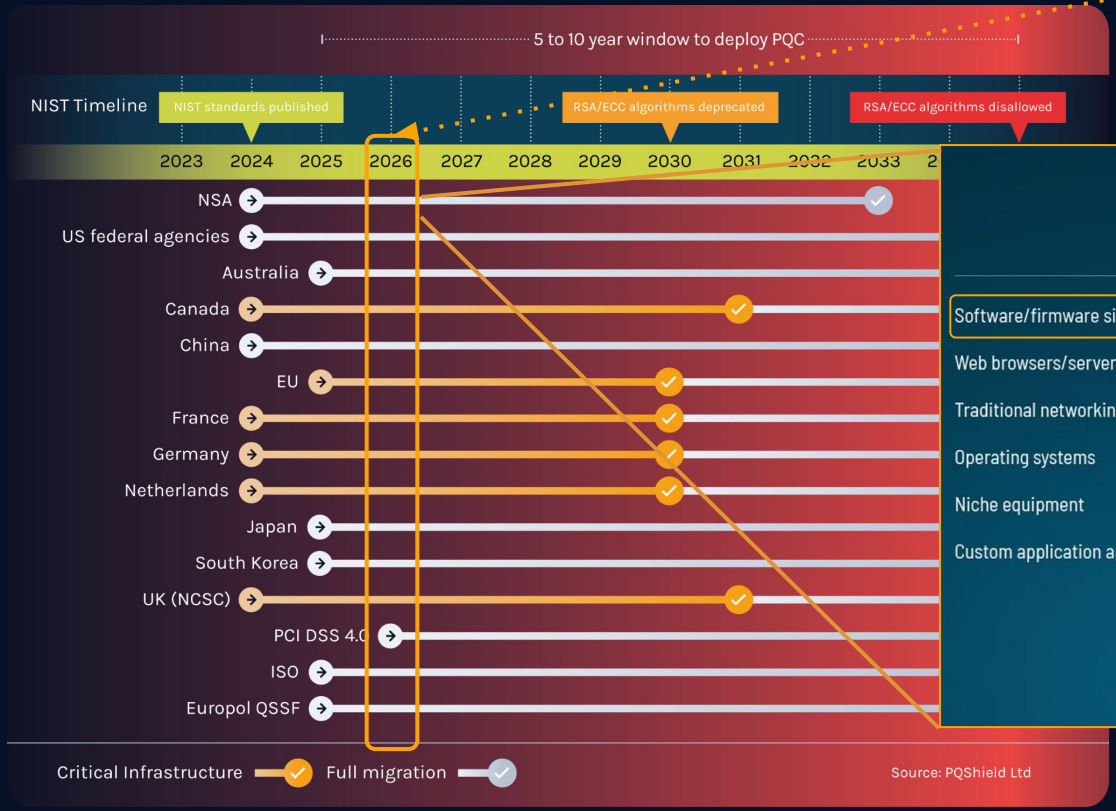
- **2006-2009 U Bochum:**
  PhD in Lightweight Cryptography
- **2009-2014 NTU:**
  Asst Prof in Cryptographic Engineering
- **2014-2017 NXP Semiconductors:**
  Leading internal hardware hacking team
- **2017-2023 xen1thLabs:**
  General Manager of the NTVL of the UAE
- **2022 INSEAD:**
  Executive MBA
- **2023 PQShield:**
  VP of Product Security & Innovation

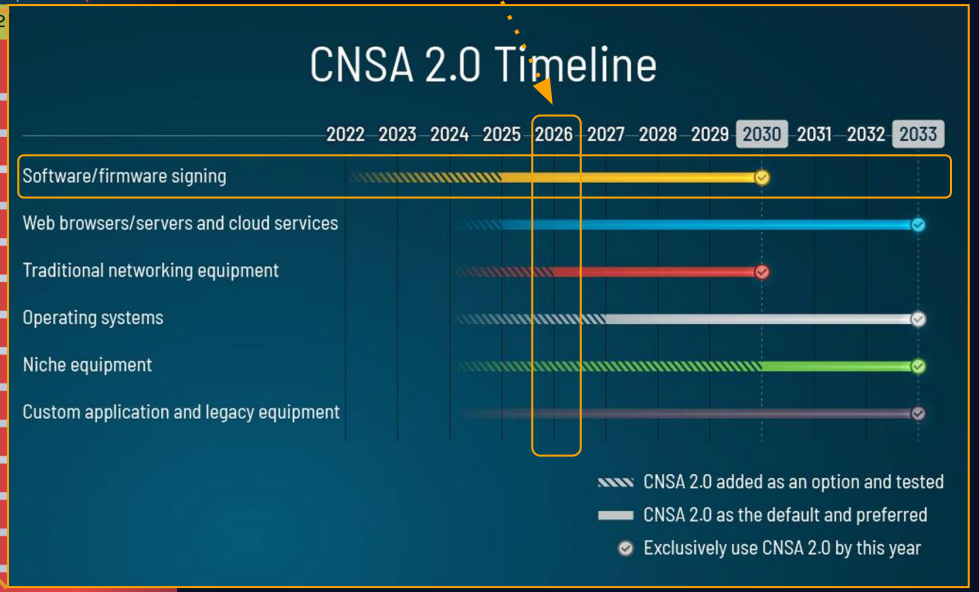
outline: overview

- Why migrate now?
- Where to migrate to?
- Why do we need Crypto Agility?
- What is the Problem with a Root of Trust?
- What is the FORTRESS Project?
- Conclusion

Migration timelines

We are here!





Migration timelines accelerate

विज्ञान एवं प्रौद्योगिकी विभाग
DEPARTMENT OF
SCIENCE & TECHNOLOGY

Implementation of Quantum Safe Ecosystem in India

Cloudflare targets 2029 for full post-quantum security

2026-04-07

Bas Westerbaan

Quantum frontiers may be closer than they appear

Mar 25, 2026
2 min read

We're setting a timeline for post-quantum cryptography migration to 2029.

Heather Adkins
VP, Security Engineering

S

Sophie Schmieg
Senior Staff
Cryptography Engineer

Sha

Ripple is introducing a multi-phase roadmap to prepare the XRP Ledger (XRPL) for a post-quantum future, with a target for full readiness by 2028.

Newer migration timelines are more aggressive, while existing timelines are being accelerated

outline: overview

- Why migrate now?
- Where to migrate to?
- Why do we need Crypto Agility?
- What is the Problem with a Root of Trust?
- What is the FORTRESS Project?
- Conclusion

Migration: full PQC or PQ/T Hybrid?

NCSC: Transition to full PQC
PQ/T hybrid should be carefully evaluated

NSA: Transition to full PQC
PQ/T hybrid not recommended (CNSA1.0 for IKEv2 indefinitely)

ASD: Transition to full PQC
PQ/T hybrid not recommended, but not prohibited

BSI: Transition to PQ/T hybrid
optional full PQC at a later stage

ANSSI: Transition to PQ/T hybrid
full PQC after 2030

Painpoints: PQ/T Hybrid = 2x complexity

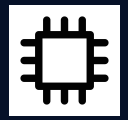
PQ/T Hybrid doubles number of algorithms to support

Increases cost due to increased

- Memory
- Area

Greater computational overhead

- Latency
- Energy consumption



Greater bandwidth overhead

- Latency
- Throughput

Increased complexity results in more difficult:

Maintenance

Analysis

Secure implementation

PQC Protectability: Fault Attacks

Algorithm	Grade (5 = best)	Argument
ML-DSA	3	Protection against loop-abort attacks required. Full signature recomputation has **relatively low performance cost** compared to rejection sampling overhead.
FN-DSA	2	Recomputation countermeasure against FA using faulty valid signatures, is **more costly** than for ML-DSA.
SLH-DSA	2	FA forcing multiple uses of one time signature (WOTS) scheme, **full redundancy effective but costly**.
LMS	1	Similar to SLH-DSA but with a **limited number of faults and additional difficulty to protect** due to state.
XMSS	1	Similar to SLH-DSA but with a **limited number of faults and additional difficulty to protect** due to state.

PQC Protectability: Side-Channel

Algorithm	Grade (5 = best)	Argument
ML-DSA	3	Operations computing directly on the **long term secrets are easy** to protect, while ephemeral secrets involve **more complicated** operations.
FN-DSA	1	Contains **floating point** operations vulnerable to SCA.
SLH-DSA	5	Due to the **structure of HBS**, very few SCA attack paths exist.
LMS	5	Similar to SLH-DSA, but with a **limited number of traces** available.
XMSS	5	Similar to SLH-DSA, but with a **limited number of traces** available.

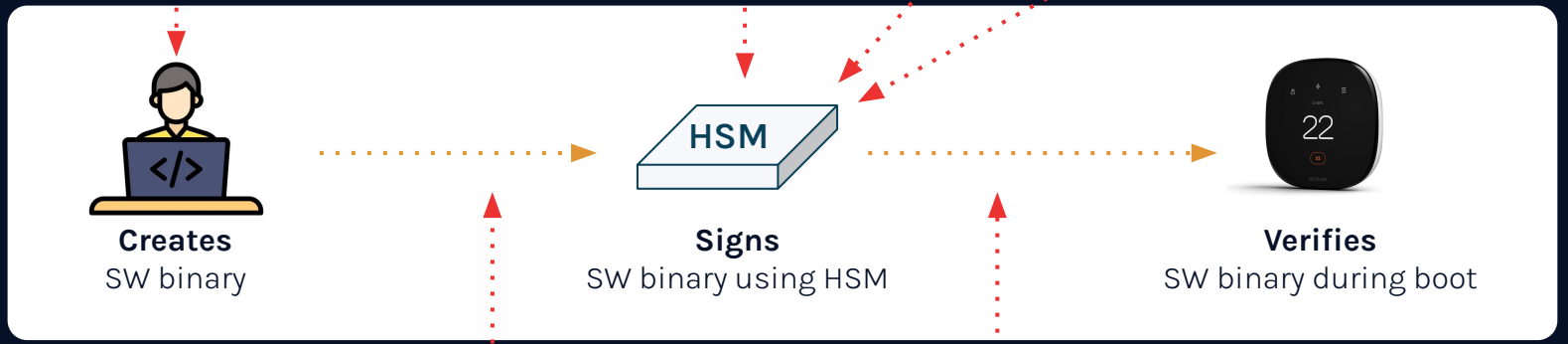
😟 Painpoints: Signature Generation

Backup of LMS key not allowed

User authentication based on 2FA /wo PQC

Backup of key is secured with ECC

No CAVP testing for XMSS



Communication is secured with ECC

Algorithmic choice: no silver bullet in sight

Algorithm	CNSA2.0	PQ/T Hybrid	Fault	SCA	CAVP/ACVP	Key Backup
ML-DSA	V	V	3	3	V	V
FN-DSA	X	V	2	1	X	V
SLH-DSA	X	X	2	5	V	V
LMS	V	X	1	5	V	X
XMSS	V	X	1	5	X	X

outline: overview

- Why migrate now?
- Where to migrate to?
- Why do we need Crypto Agility?
- What is the Problem with a Root of Trust?
- What is the FORTRESS Project?
- Conclusion

Migration: deprecated and disallowed algorithms

	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035
USA	Fully PQC										RSA ECDSA
UK	Fully PQC										
AUS	Fully PQC					RSA ECDSA Disallowed					
FR	PQ/T Hybrid					Fully PQC					
DE	PQ/T Hybrid					Fully PQC Optional					

For the foreseeable future (2030/2035) PKC and PQC have to be supported

Not to mention additional regional algorithms...

Crypto agility: overview

WHAT

"Crypto agility describes the capabilities needed to replace and adapt cryptographic algorithms for protocols, applications, software, hardware, and infrastructures without interrupting the flow of a running system to achieve resiliency."

NIST CSRC

WHY

Future-Proofing

Residual risk of stronger or yet unknown attacks on new PQC standards

Compliance

Different regulation in different geographies

Sovereignty

Enable sovereign crypto algorithms

Crypto agility: overview

HOW

Algorithms

- Increasing key sizes
- Tweaking algorithms
- Replacing algorithms

A Method for Obtaining Digital Signatures and Public-Key Cryptosystems

R.L. Rivest, A. Shamir, and L. Adleman*

Abstract

An encryption method is presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key. This has two important consequences:

1. Couriers or other secure means are not needed to transmit keys, since a message can be enciphered using an encryption key publicly revealed by the intended recipient. Only he can decipher the message, since only he knows the corresponding decryption key.
2. A message can be "signed" using a privately held decryption key. Anyone can verify this signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a signer cannot later deny the validity of his signature. This has obvious applications in "electronic mail" and "electronic funds transfer" systems.

A message is encrypted by representing it as a number M, raising M to a publicly specified power e, and then taking the remainder when the result is divided by the publicly specified product, n, of two large secret prime numbers p and q. Decryption is similar; only a different, secret, power d is used, where e*d ≡ 1 (mod (p-1)*(q-1)). The security of the system rests in part on the difficulty of factoring the published divisor, n.

Key Words and Phrases: digital signatures, public-key cryptosystems, privacy, authentication, security, factorization, prime number, electronic mail, message passing, electronic funds transfer, cryptography.

CR Categories: 2.12, 3.15, 3.50, 3.81, 5.25

This work was supported by the National Science Foundation grant MCS76-14294, and the Office of Naval Research grant number N00014-67-A-0204-0093.

Author's Address: Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139. E-mail addresses: rivest@theory.lcs.mit.edu

Protocols

- Tweaking protocols
- Replacing protocols

😟 **No infrastructure exists to automatically find or update your crypto**

Crypto agility: opportunities

Zero Trust Alignment

- Assume Breach is a core principle of Zero Trust
- Cybersecurity is fundamentally based on cryptography
- Time to include crypto in the agility loop

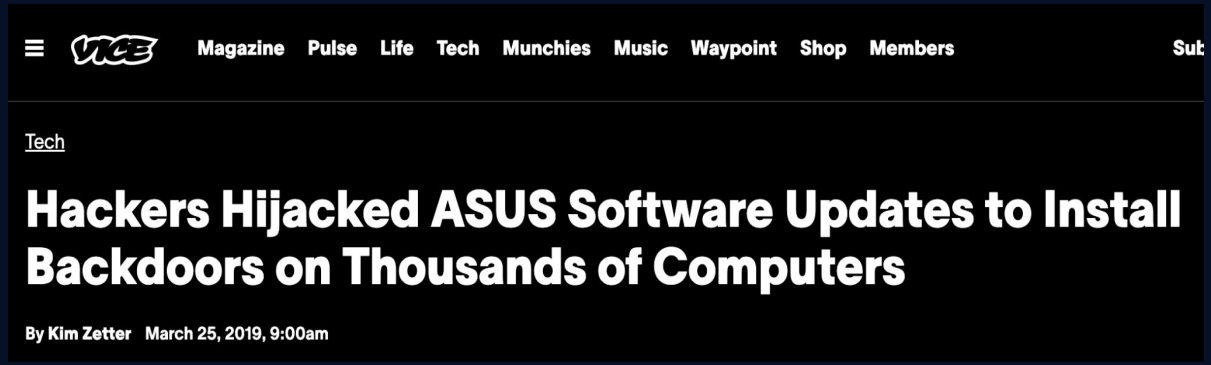
Implementation Updates

Updating crypto implementations allows to:

- Add side-channel countermeasures
- Fix bugs

Crypto Agility turns static cryptography into a dynamic, resilient security layer

Crypto agility: drawback

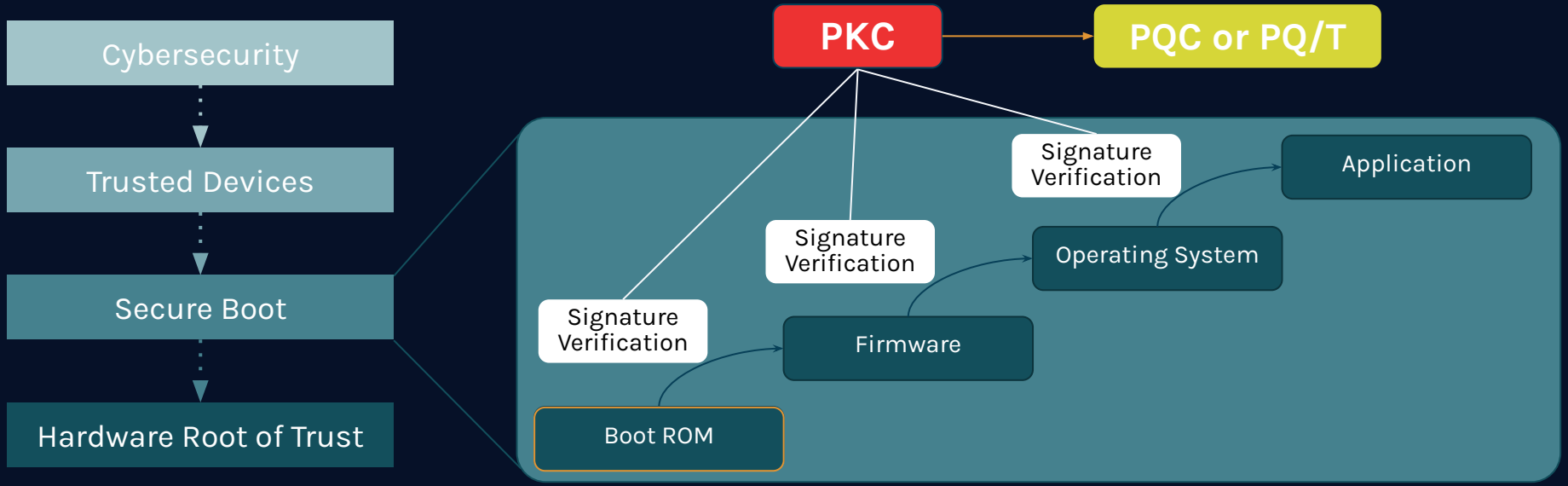


😟 Crypto Agility dramatically increases attack surface

outline: overview

- Why migrate now?
- Where to migrate to?
- Why do we need Crypto Agility?
- What is the Problem with a Root of Trust?
- What is the FORTRESS Project?
- Conclusion

Root of Trust: the bedrock of cybersecurity



Painpoint: unpatchable flaws like checkm8

Root of trust: based on immutable hardware

Common Weakness Enumeration

A community-developed list of SW & HW weaknesses that can become vulnerabilities

Home > CWE List > CWE-1326: Missing Immutable Root of Trust in Hardware (4.20)

Home | About ▼ | Learn ▼ | Access Content ▼ | Community ▼ | Searc

CWE-1326: Missing Immutable Root of Trust in Hardware

Description

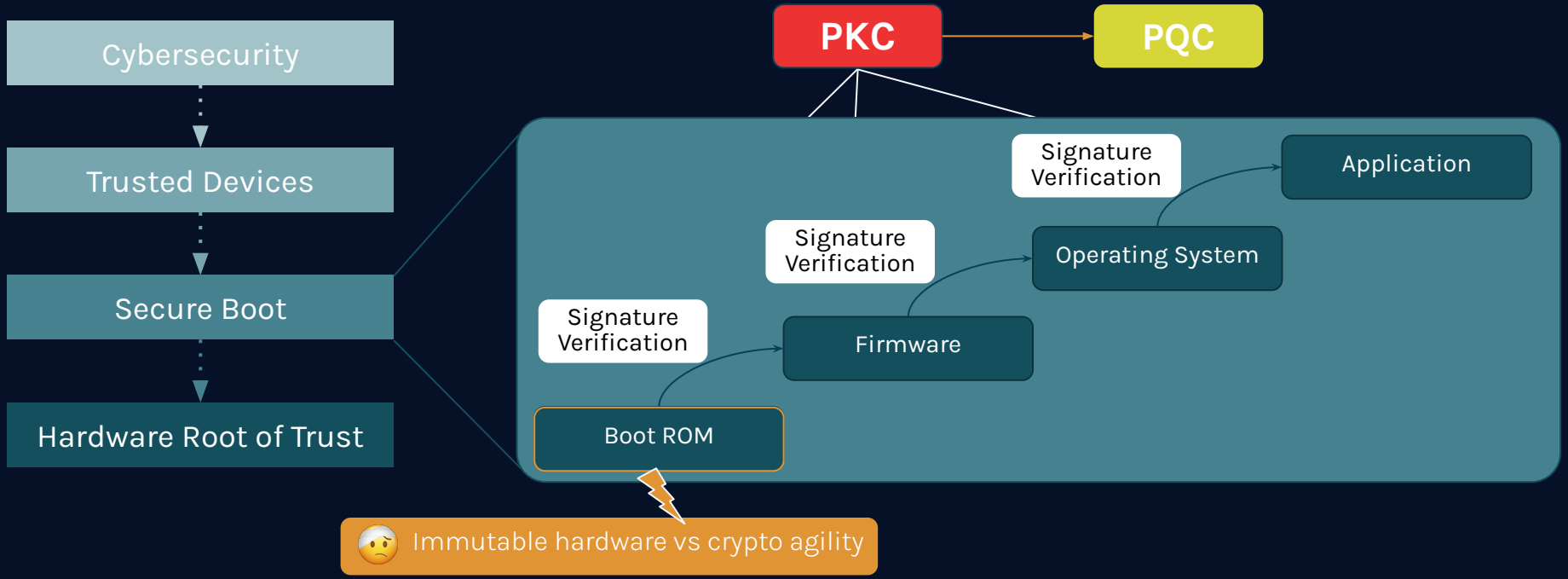
A missing immutable root of trust in the hardware results in the ability to bypass secure boot or execute untrusted or adversarial boot code.

In general, if the boot code, key materials and data that enable "Secure Boot" are all mutable, the implementation is vulnerable.

Good architecture defines RoT as immutable in hardware. One of the best ways to achieve immutability is to store boot code, public key or hash of the public key and other relevant data in Read-Only Memory (ROM) or One-Time Programmable (OTP) memory that prevents further programming or writes.

https://cwe.mitre.org/data/definitions/1326.html

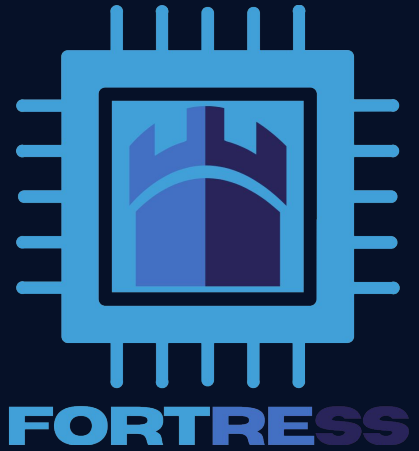
Root of Trust: the bedrock of cybersecurity



outline: overview

- Why migrate now?
- Where to migrate to?
- Why do we need Crypto Agility?
- What is the Problem with a Root of Trust?
- What is the FORTRESS Project?
- Conclusion

PQSHIELD



Fully Optimized Root of Trust for Robust Embedded Secure Systems

FORTRESS: project overview

HORIZON_CL3-2024-CS-01 project.
Grant agreement number: 101225722

36 Months
Sep. 2025 - Aug. 2028

Budget: 4.9 M

Project Consortium

EURESCOM Innovation through Collaboration
PQ SHIELD
eShard
cyberhive
Codasip
der Bundeswehr Universität München

FORTRESS: Project objectives

1 - KPI Exploration

Develop a benchmarking framework for PQ/T hybrid secure-boot systems to evaluate trade-offs in security, cost, and performance.

2 - PQ/T Hybrid RoT

Design a flexible, CHERI-enabled PQ/T Hybrid Root of Trust integrating hardware and software components.

3 - Secure Boot Demonstrator

Showcase real-world secure boot implementations for IoT and CNI environments.

4 - Security Tool Deployment

Provide security testing and validation tools to support large-scale PQC adoption.

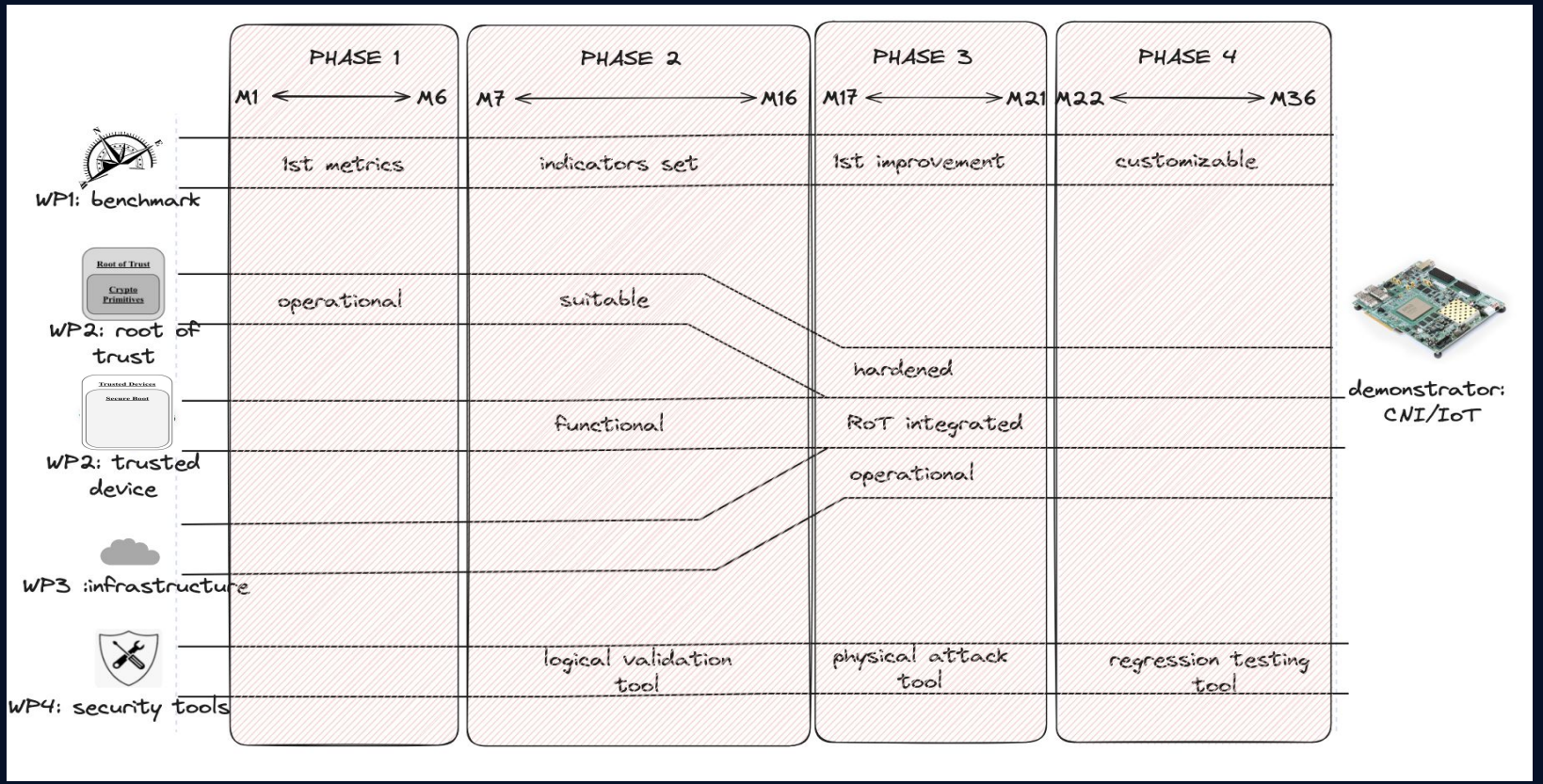
5 - Trusted Communication Platform

Build a Trusted Communication Platform using PQ/T hybrid encryption for secure, efficient IoT and CNI connectivity.

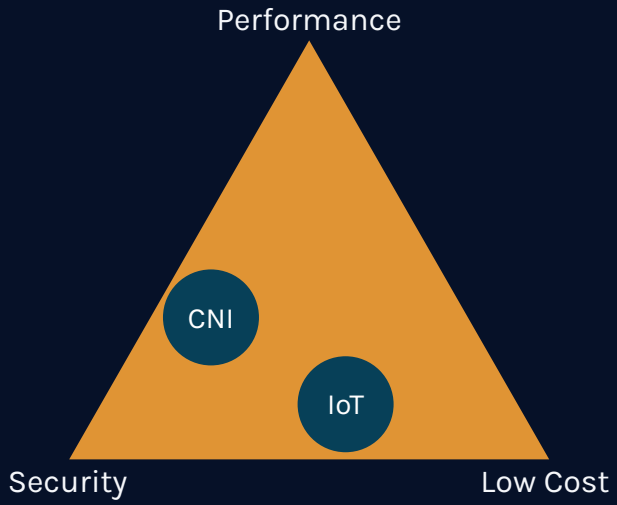
6 - Dissemination & Standardisation

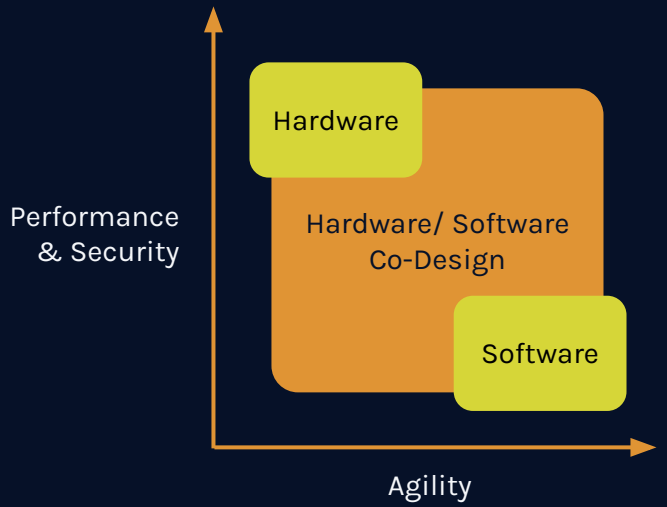
Maximise impact by promoting FORTRESS innovations across industry, academia, and standardisation bodies.

FORTRESS: Methodology



FORTRESS: trade-offs, hardware, and software





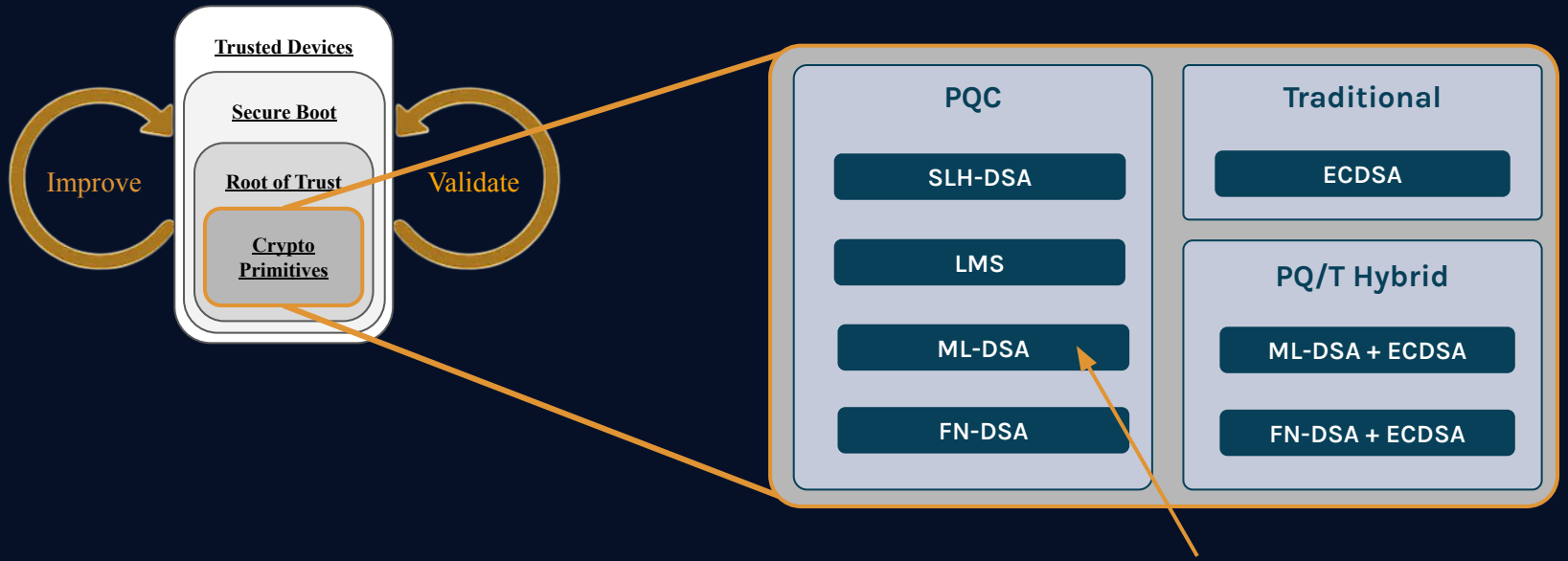
Use-case specific optimization goals

- Performance
- Security
- Cost

Optimized integrated solution

- Hardware for performance and security
- Software for agility

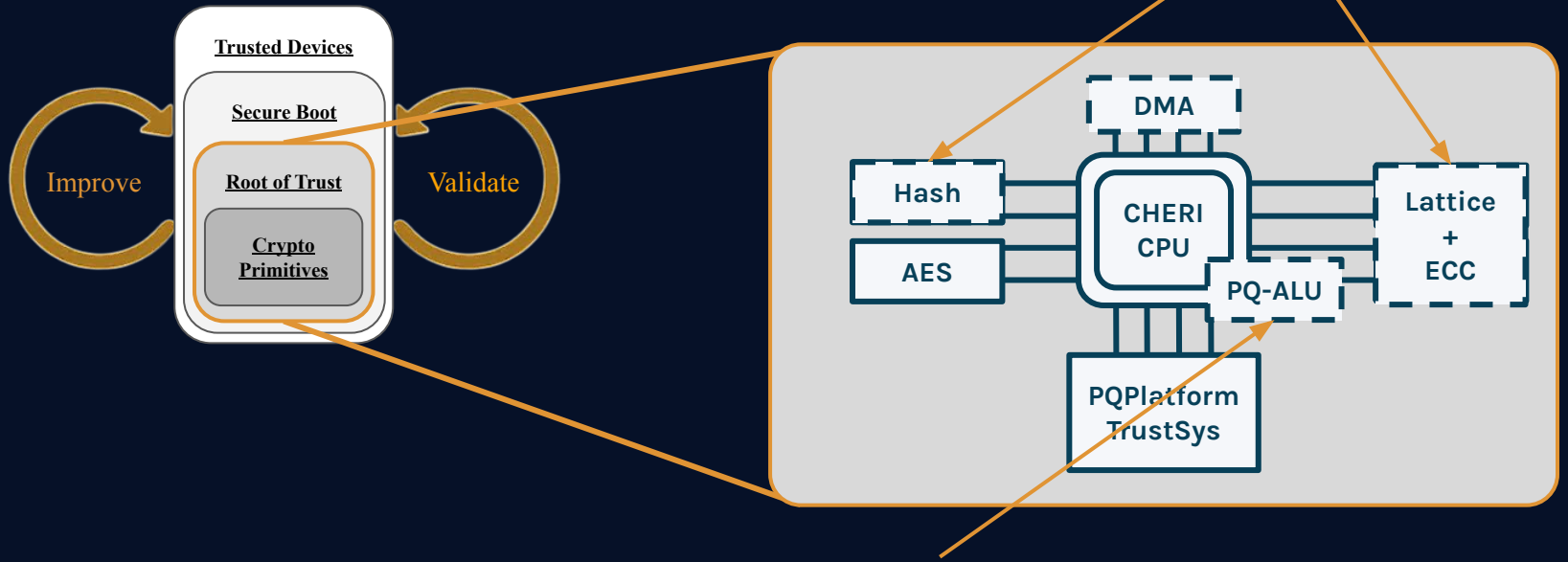
FORTRESS: approach and algorithms



[Apples, Oranges, and Signatures: Pitfalls and Methodology in ML-DSA Benchmarking, Riou et al., MAGICS 2026]

FORTRESS: hw/sw architecture

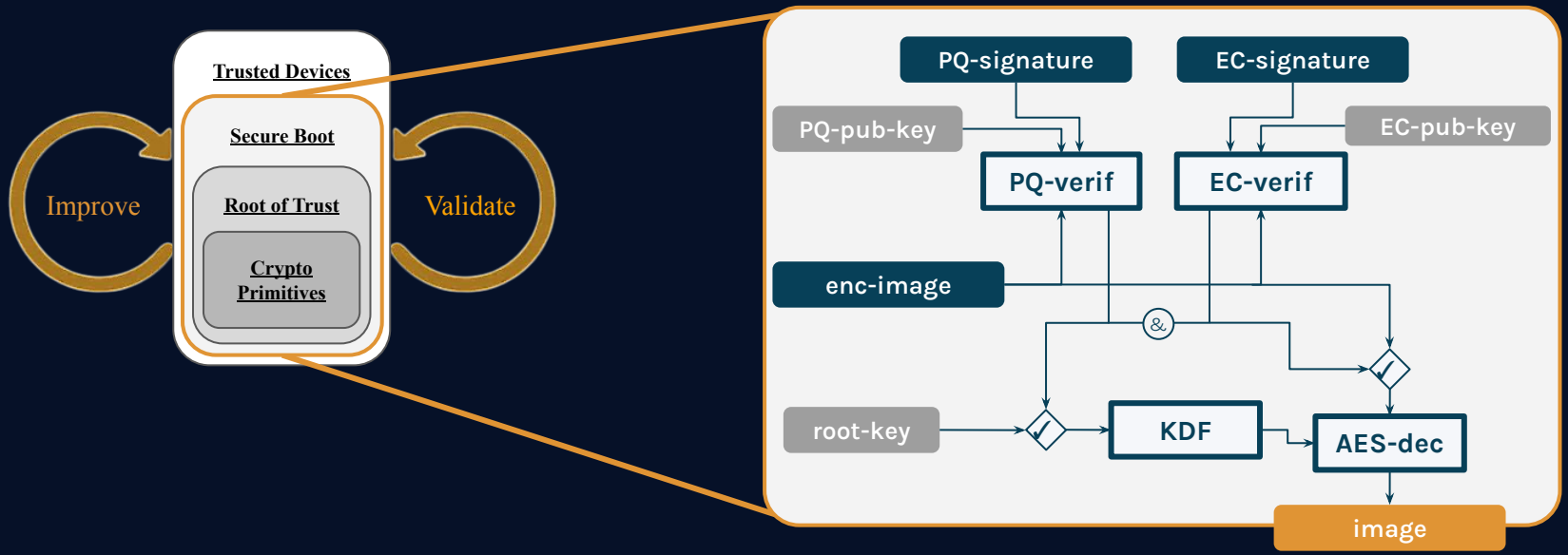
Publications coming up soon, stay tuned!



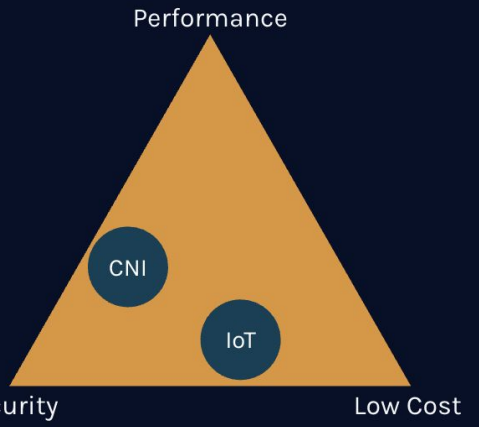
[Raise the Shields: A Modular RISC-V Extension for Post Quantum Cryptography, Piscopo et al., DAC 2026]

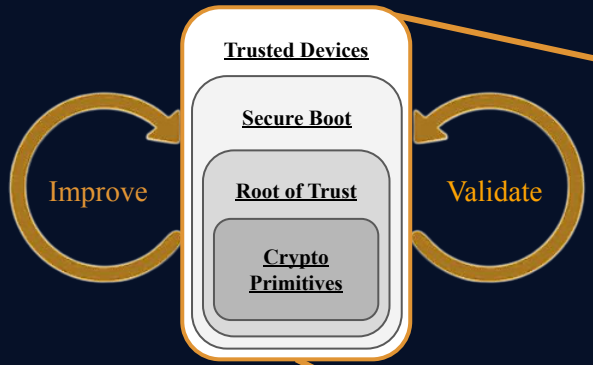
FORTRESS: secure boot flow

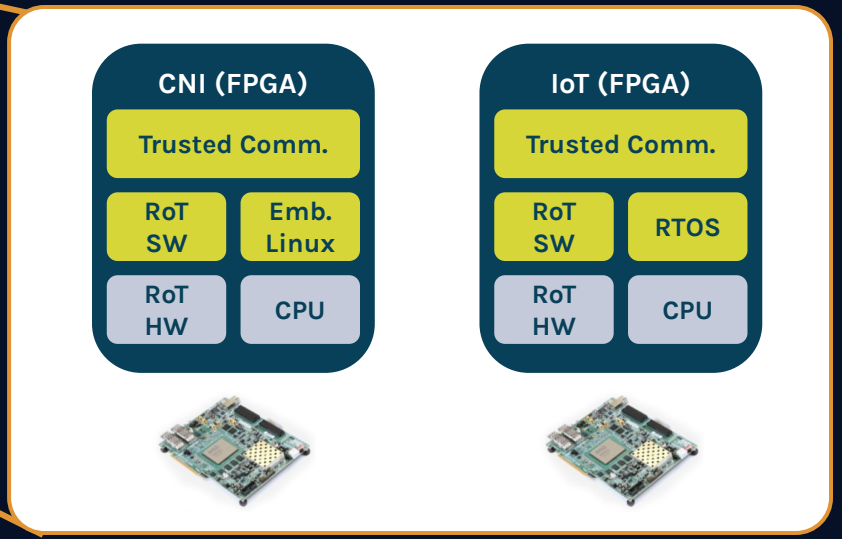
Checkpoint



FORTRESS: demonstrators







outline: overview

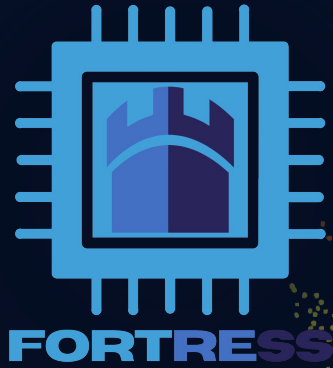
- Why migrate now?
- Where to migrate to?
- Why do we need Crypto Agility?
- What is the Problem with a Root of Trust?
- What is the FORTRESS Project?
- Conclusion

Take away: Root of trust and crypto agility

- CNSA2.0 sees sw/fw updates as highest priority
- Different regulatory timelines -transition and deprecation- require configurability
- Crypto agility is required to enable this transition
- Root of trust is the bedrock of cybersecurity
- Immutability is both bug and feature
- FORTRESS project investigates trade-offs between performance, security, and cost
- We invite everyone to collaborate with us!

PQ SHIELD

Questions?



www.pq-fortress.eu