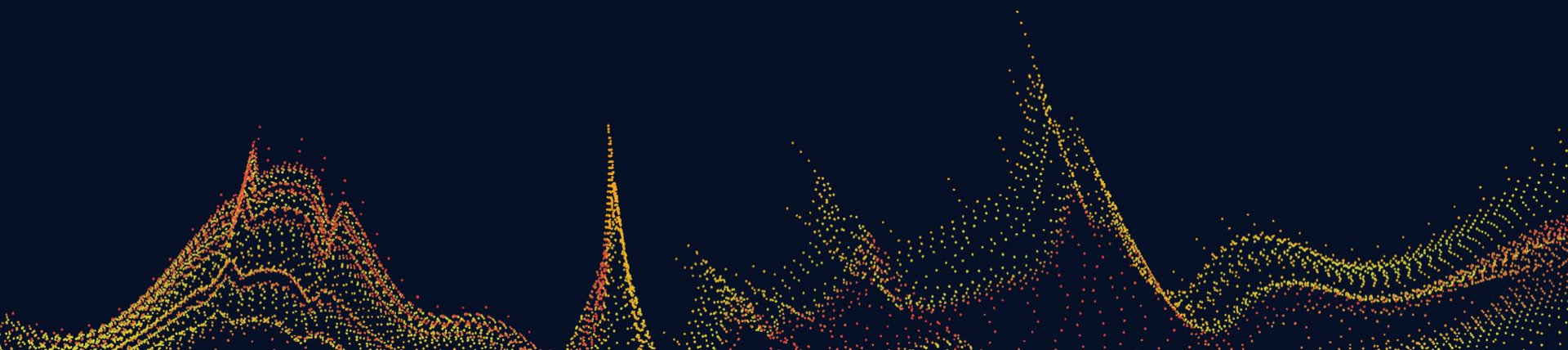


Making Post-Quantum Authentication Fit in Embedded Stacks

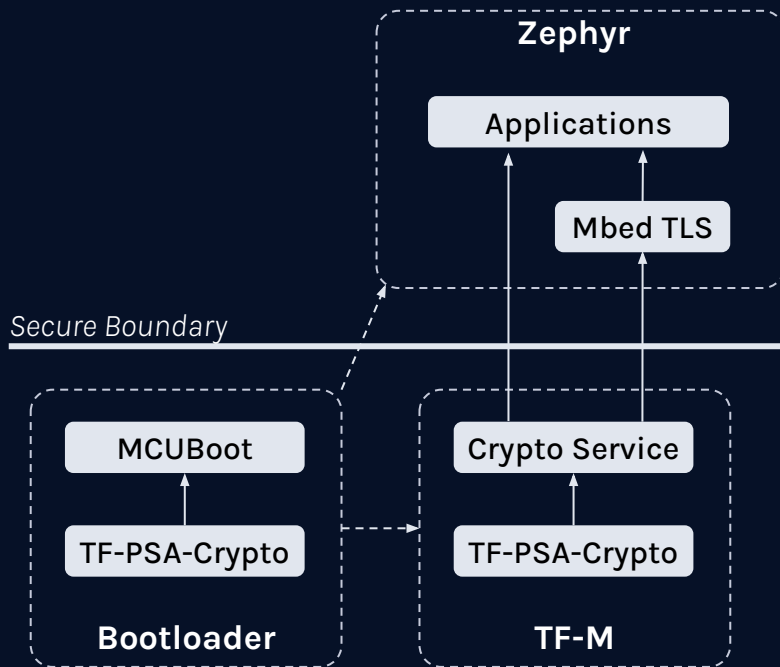
Luis Chan, PQShield



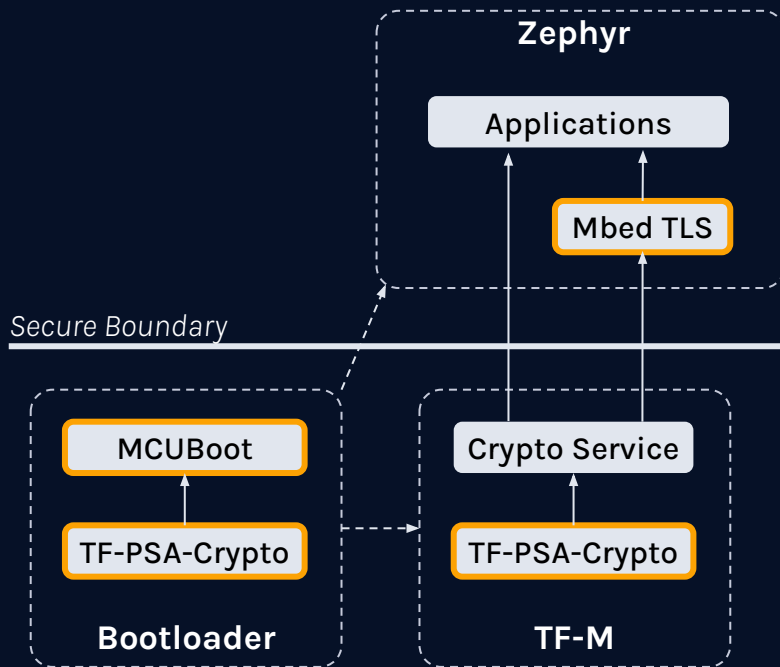
How we got here

- Heavy focus on developing PQC solutions that fit constrained environments
 - Across software and hardware, covering both products and research projects
- Early customers largely silicon vendors
 - Own platform and SDK
 - Products deployed for decades
 - **Main focus was making the primitives fit their HW constraints**
- As time has passed and deadlines draw nearer, interest in embedded space has started to move higher up the stack...
- **Which has led us to start looking into middleware integrations ourselves**

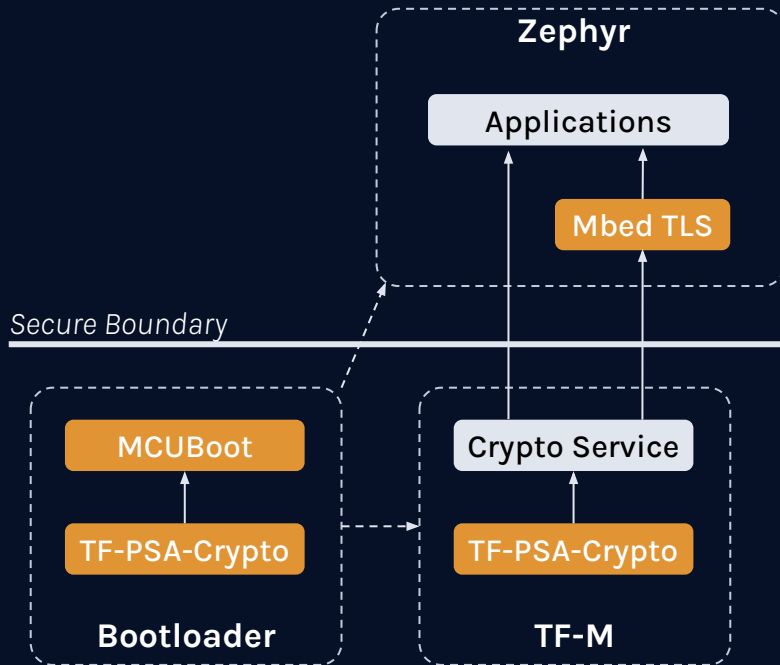
Simplified Zephyr Stack on TrustZone



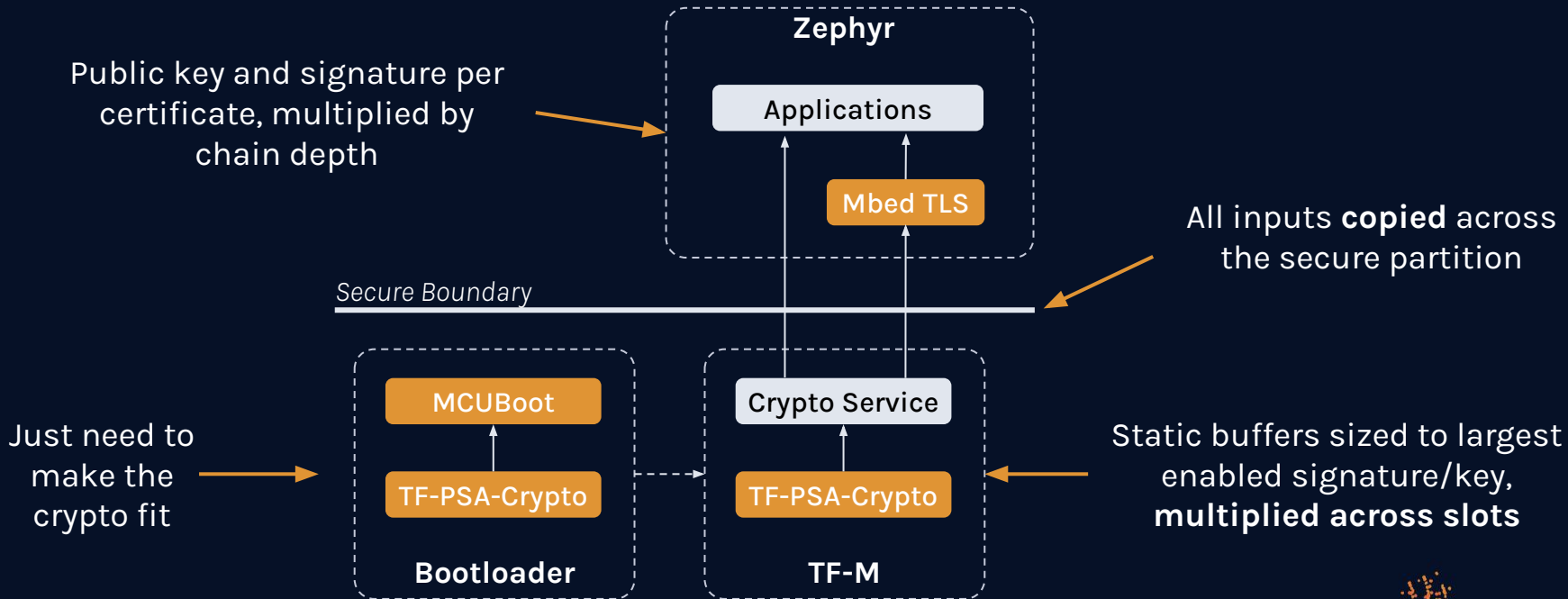
PQC is missing today...



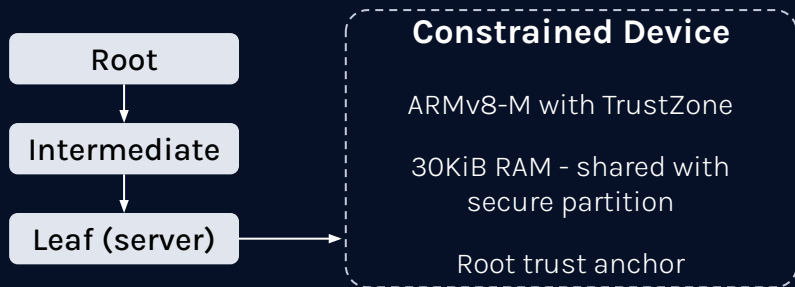
Where we're adding it



What we ran into



A customer-shaped constrained deployment



P-384, CNSA 1.0 Compliant Operator Profile

Public key: 97 B

Signature: 104 B

Certificate chain: ~1.4KiB

CertificateVerify signature: 104B

Secure working copies: ~200B

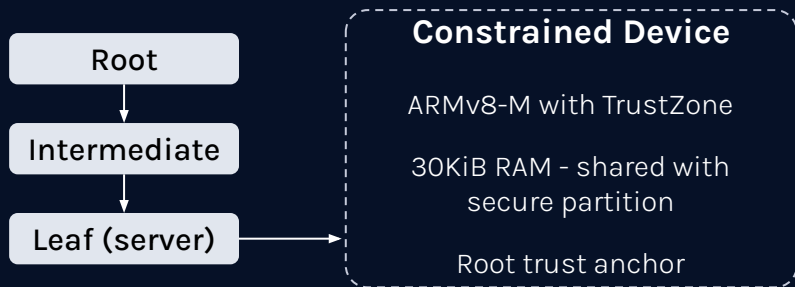
Total Usage* (out of 30KiB): ~1.7KiB

**Not accounting for:*

Handshake RAM, Crypto Primitive RAM, Application RAM

With traditional signature schemes, this workflow can be tuned to fit

A customer-shaped constrained deployment



ML-DSA-87, CNSA 2.0 Compliant Operator Profile

Public key: 2592 B Signature: 4627 B

Certificate chain: ~15KiB

CertificateVerify signature: 4.5KiB

Secure working copies: 7KiB

Total Usage* (out of 30KiB): ~26.5KiB

**Not accounting for:
Handshake RAM, Crypto Primitive RAM, Application RAM*

With PQC, this same workflow becomes unaffordable from authentication objects alone.

Resource constraints belong in authentication design

- Engineering can optimise the lower levels - buffers, copies, configuration
- But at the top of these stacks, formats and protocols determine how far those costs are amplified
- Customers see it too, and have started asking us about MTC on embedded
- Authentication designed with large PQ objects as a starting assumption will be increasingly important as adoption reaches more constrained deployments
- Which is why we're following developments like MTC and NIKEs closely, and how they can be used in embedded systems

What would we design differently if large PQC objects were a starting assumption?

- Protocol designers – where is this already shaping the work?
- Implementers and operators – where are these constraints starting to bite?
- Where would PQShield's implementation and research effort be most useful?