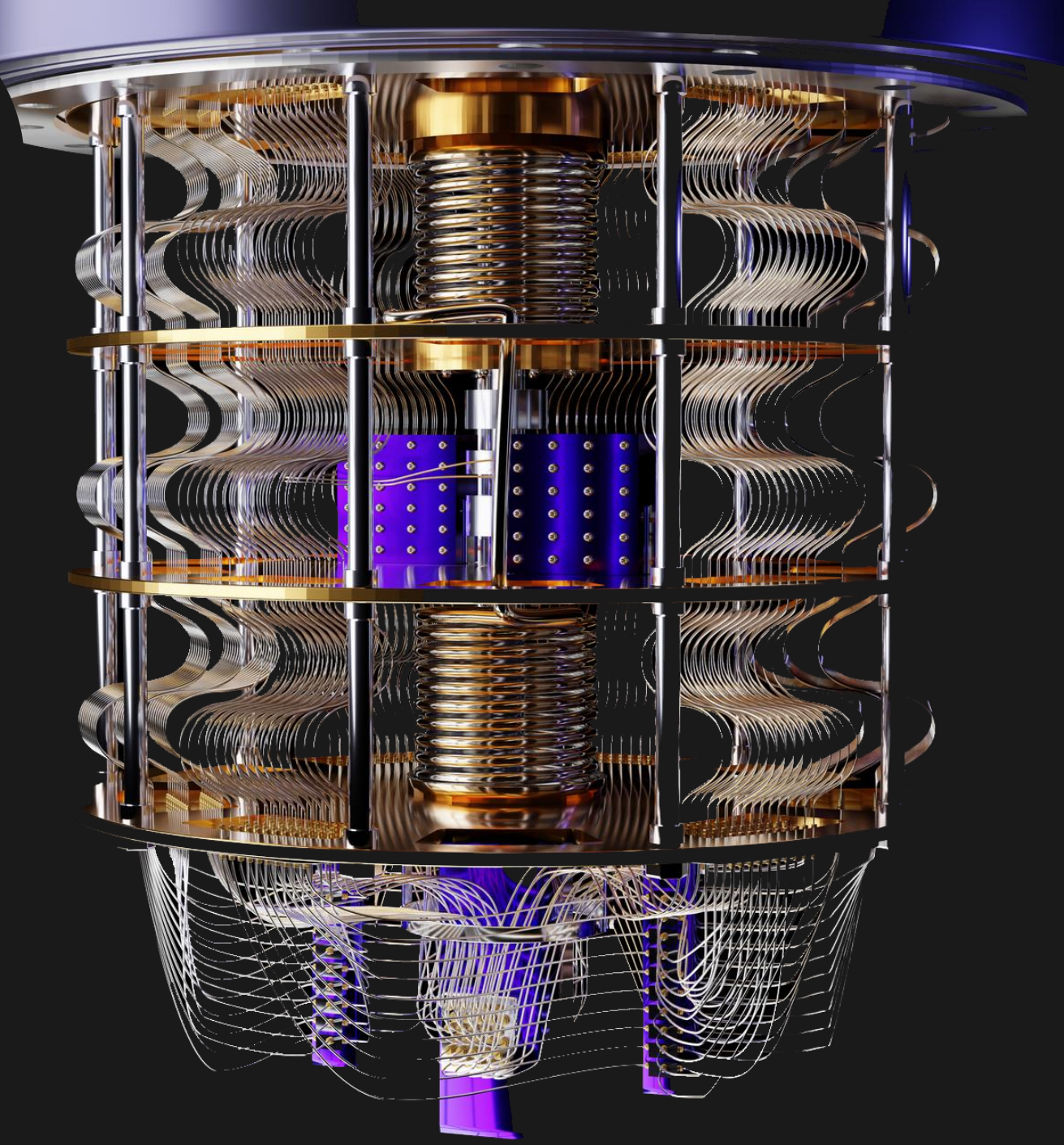
Interoperability in Action

Post-Quantum Cryptography

Tomas Gustavsson
Chief PKI Officer



KEYFACTOR

~~Dilithium~~

ML-DSA / FIPS 204

~~Kyber~~

ML-KEM / FIPS 203

~~SPHINCS+~~

SLH-DSA / FIPS 205

KEYFACTOR

Standards

KEYFACTOR

Certificate Standards?

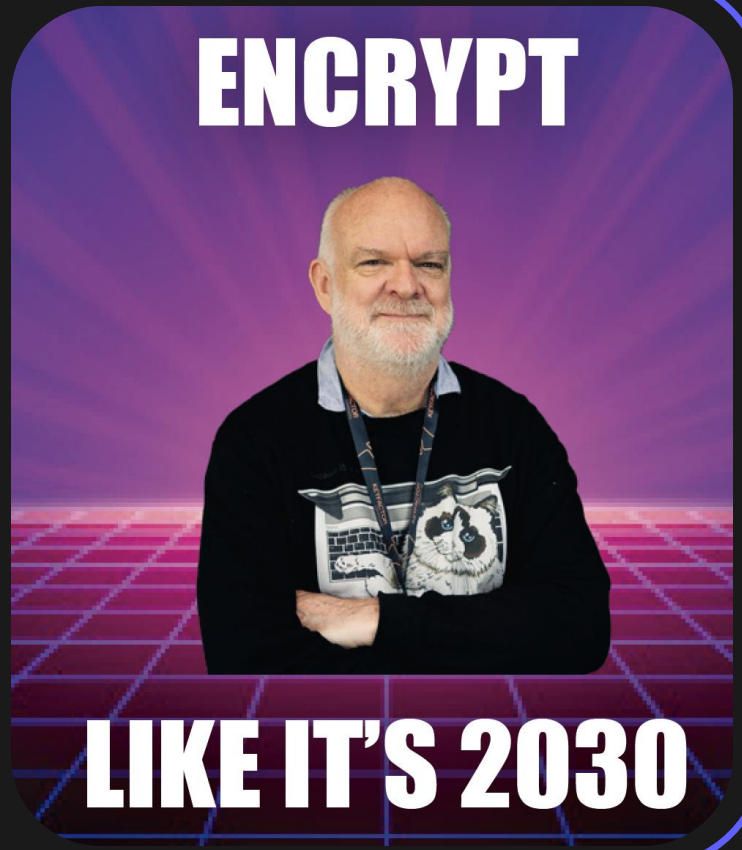
ML-DSA certificates →
Private key format changed in last months – last call

SLH-DSA certificates →
Added Hash-SLH-DSA in last months – last call

ML-KEM certificates →

Hybrid (backwards compatible) certificates →
ITU-T standard

Composite certificates →
Changed format last in last months - last call (Sept 21)



Usage Standards?

TLS ML-KEM key exchange →
Adoption call

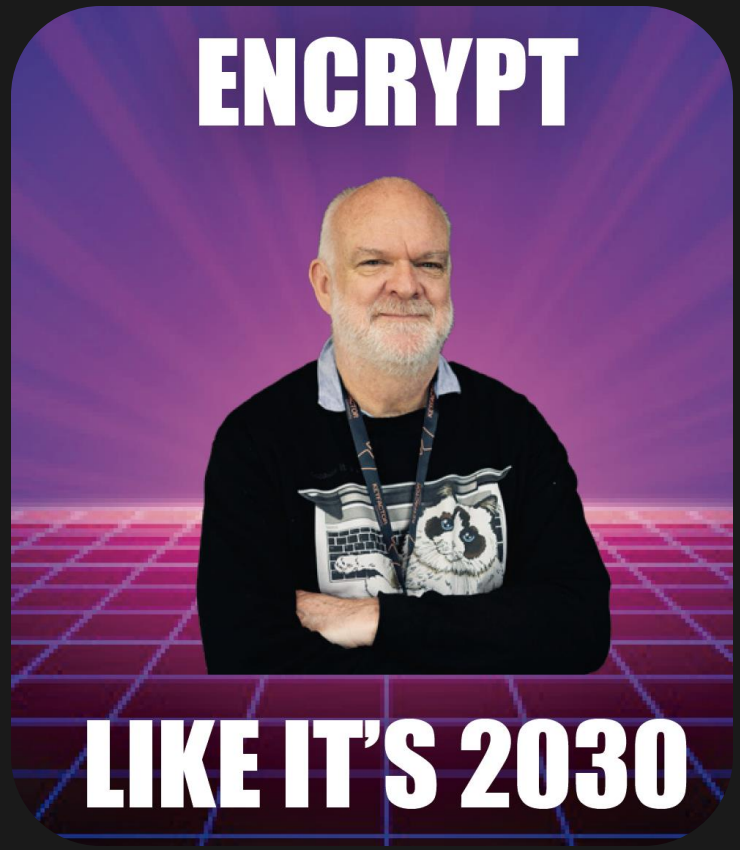
ML-KEM CMS messages →
Last call

ML-DSA CMS messages →
Last call

Hybrid certificate TLS
X9.146 draft

CABF BRs
Single key ML-DSA and ML-KEM, ballot SMC013 in August -25

IPSec
IKEv2 - RFC 8784, RFC 9242, RFC 9370



KEYFACTOR

September 2025 Heatmap: Current State of PQC Standards and Adoption

Heatmap

Standard	Overall Range	Pure PQC encrypt	Hybrid PQ encrypt	Pure PQ sig	Hybrid PQ sig
SSH	3 to 8	3	8	3	3
TLS 1.2[1]	0 to 0	0	0	0	0
TLS 1.3[2]	3 to 9	7	9	7	3
X.509[3]	4 to 7	7	4	7	4
S/MIME	3 to 7	5	3	7	3
OpenPGP	2 to 4	2	4	4	4
IKE/IPSec	2 to 8	8	8	3	2
MLS	2 to 4	4	4	4	2
DNSSec	1 to 1	-	-	1	1

Key

0	Consensus Against Inclusion
1	Blocked / Stalled
2	In Progress / Chartered
3	Active Proposals / Drafts
4	Progress to Finalization
5	Finalized / Approved
6	Integration Progress
7	Integrated in Libraries
8	Some Adoption
9	Broad Adoption
-	Unknown / NA

Source: https://pqcc.org/, October 1, 2025

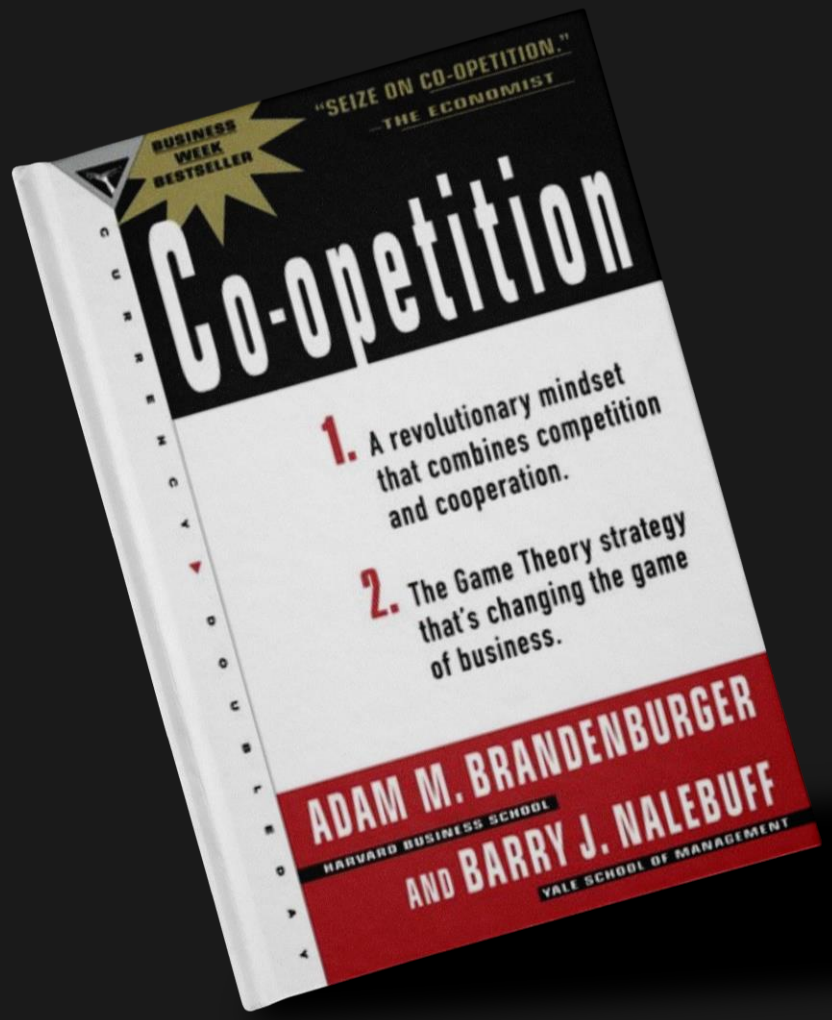
KEYFACTOR

Interoperability?

- IETF Hackatons lays the foundation →

And further on:

- Project-2-project
- Vendor-2-vendor
- Friendly co-opetition



KEYFACTOR

Interoperability?

KEYFACTOR

Interoperability Project

Interoperability and Future-Ready Cryptography

- TLS 1.3 - Certificates and mTLS
- Message Signing and Validation (CMS)
- Certificate Lifecycle Management (CLM)
- Storage Formats and Key Management
- Hybrid and Composite Certificates
- Hardware Security Modules (HSMs)

KEYFACTOR

Putting it all together

Firmware Verification

Document Signatures

Secure eMail with S/MIME

Secure Boot

Secure comm with TLS

Certificate Validation (OpenSSL/Bouncy Castle)

ML-DSA Signing

ML-DSA TSA

ML-DSA signature certificate

ML-KEM encryption certificate

ML-DSA Signing

TLS ML-KEM hybrid key exchange

ML-DSA server/ client certificate

OCSP

ML-DSA signature certificate

ML-KEM encryption

ML-DSA CA

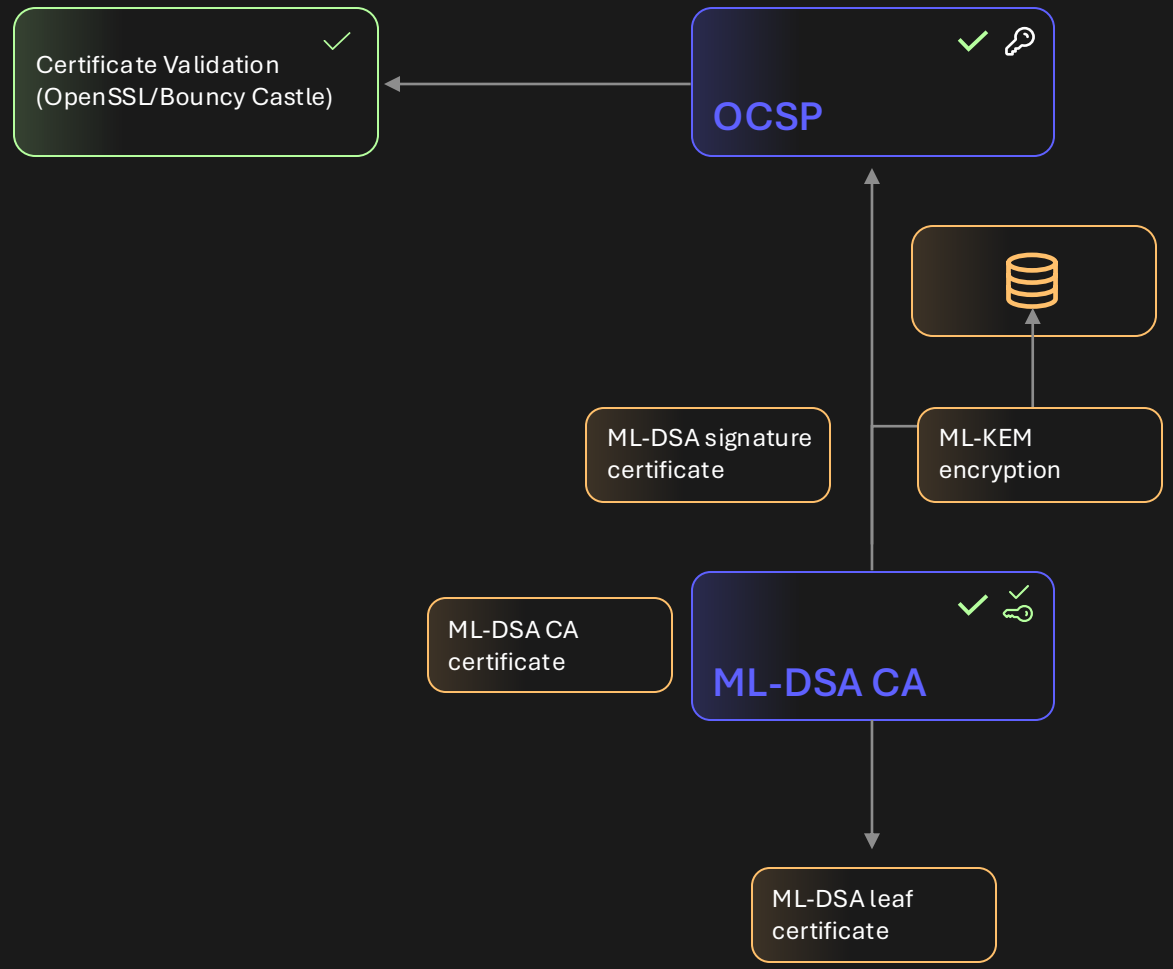
ML-DSA server certificate

ML-DSA CA certificate

Server
Apache / NginX

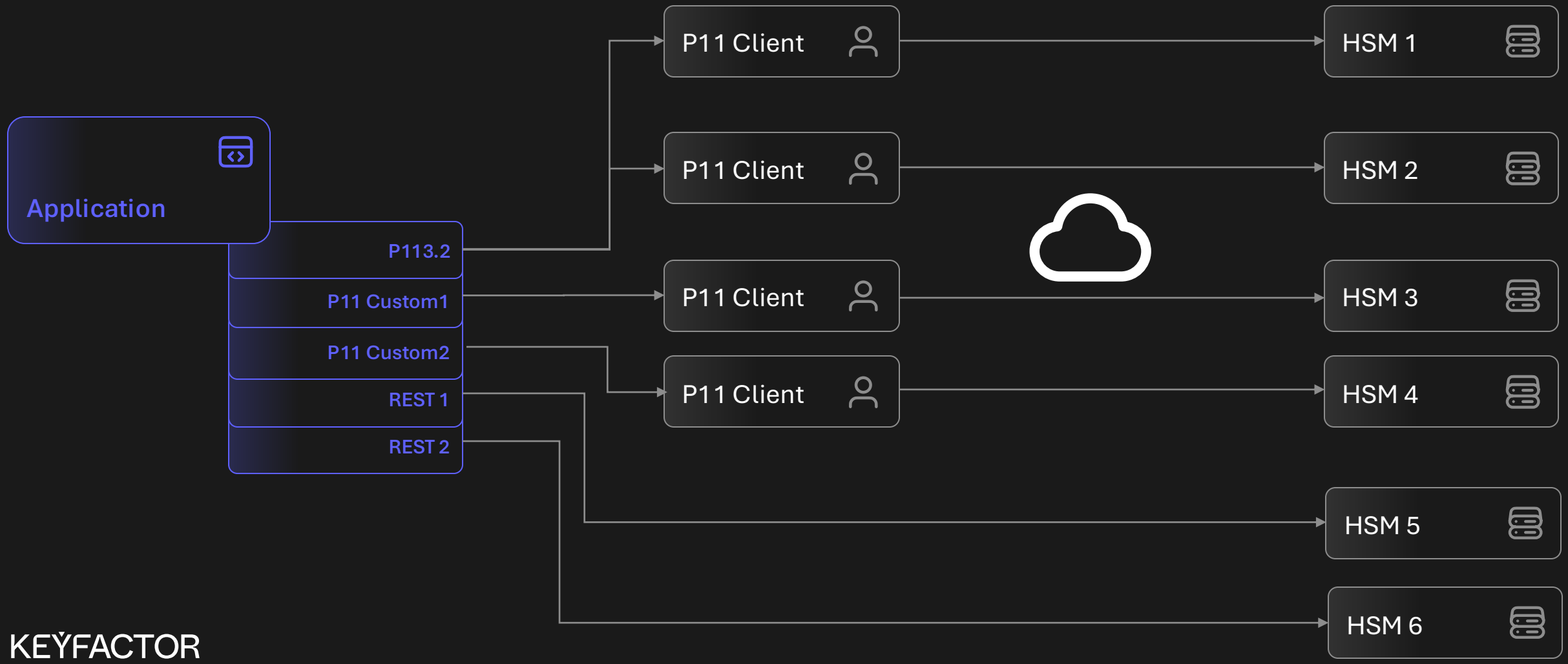
certbot

PKI



KEYFACTOR

HSM Integration in Practice



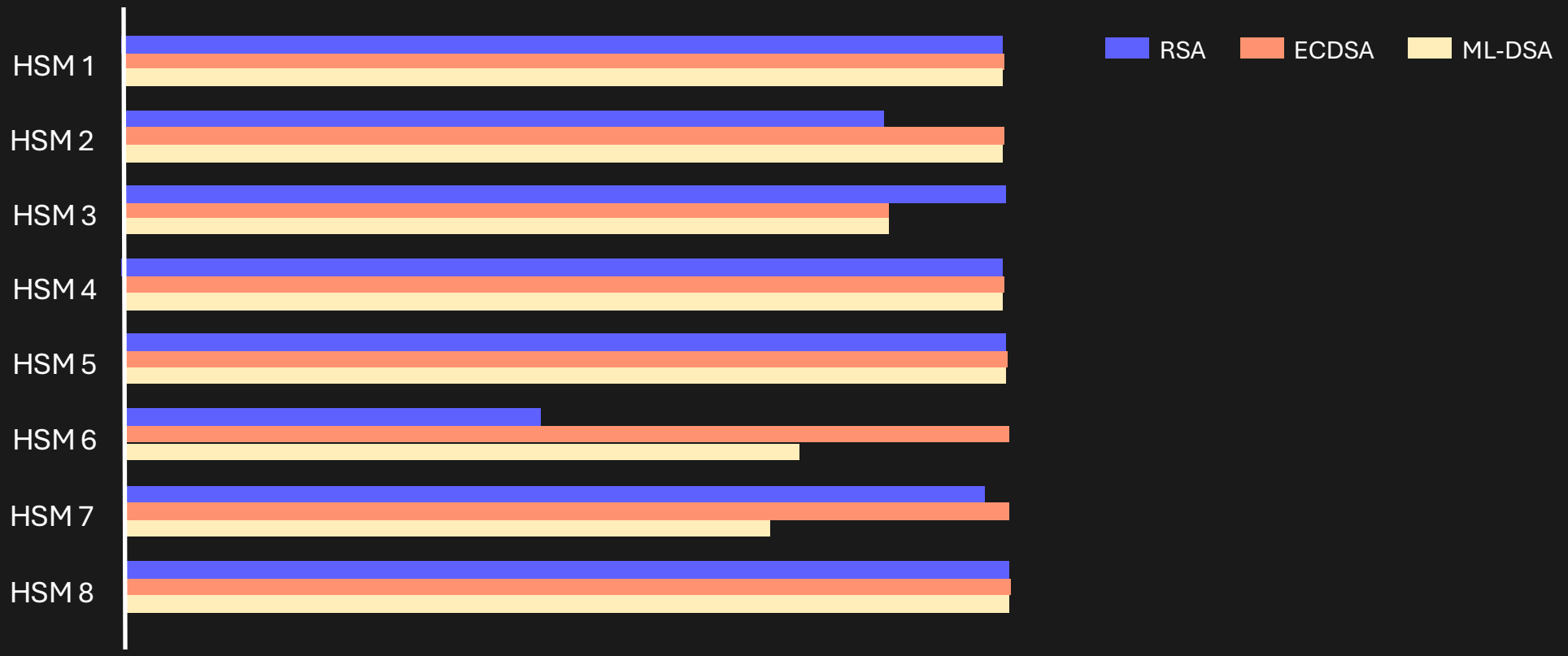
KEYFACTOR

PQC
in Practice

KEYFACTOR

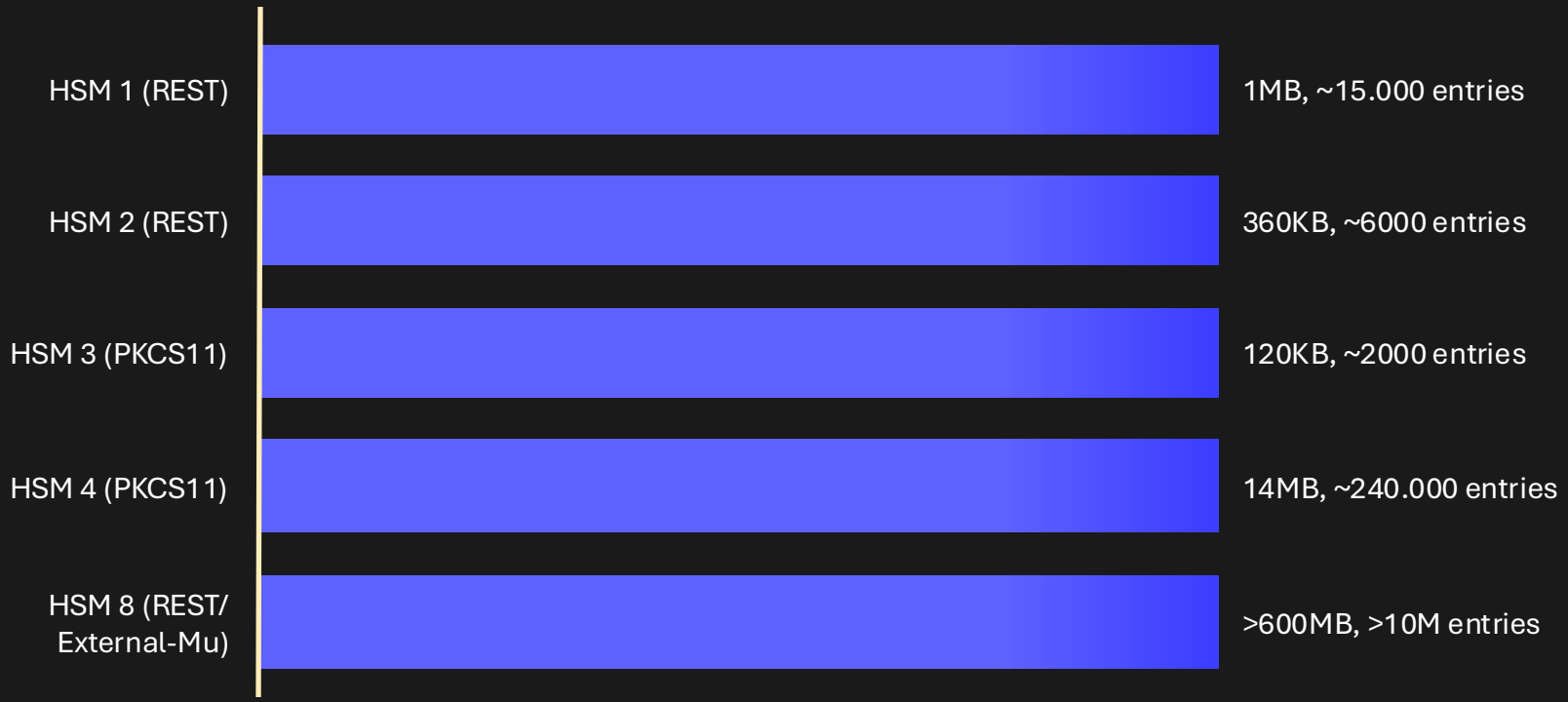
Certificate/OCSP Performance

PKCS#11 or REST



KEYFACTOR

CRL size limits (ML-DSA)



Sizes vary

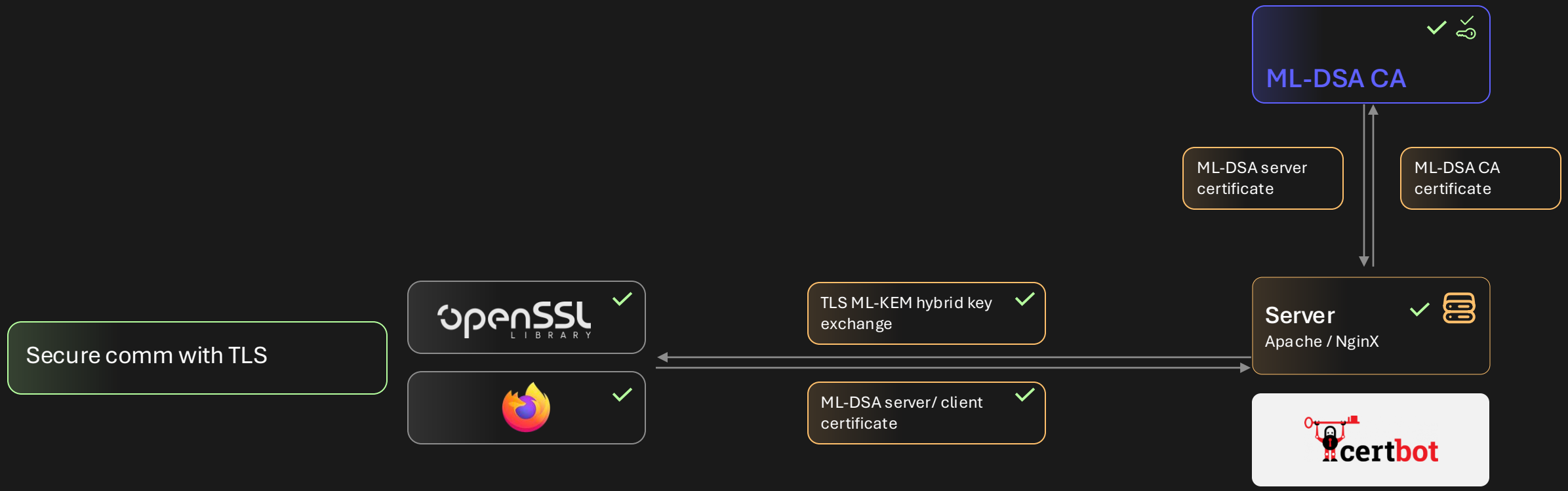
- Certificates are small
- CRLs are small to large
- Bank transactions are small
- Documents are small to medium
- Firmware is large to huge

ExternalMu-ML-DSA saves the day!
(where CMS w/ signed attributes doesn't)

KEYFACTOR

We want to know your requirements for ExternalMu-ML-DSA – come talk to us

TLS


ML-DSA CA
ML-DSA server certificate
ML-DSA CA certificate
Secure comm with TLS
OpenSSL LIBRARY
TLS ML-KEM hybrid key exchange
Server
Apache / NginX
ML-DSA server/ client certificate
certbot

Off-the-shelf TLS

- Ubuntu 25.10 - OpenSSL 3.5
- RHEL 10 – OQS

Red Hat blog →

- Alpine, CentOS Stream, Debian, SUSE, ...
- Windows Insiders

Microsoft blog →

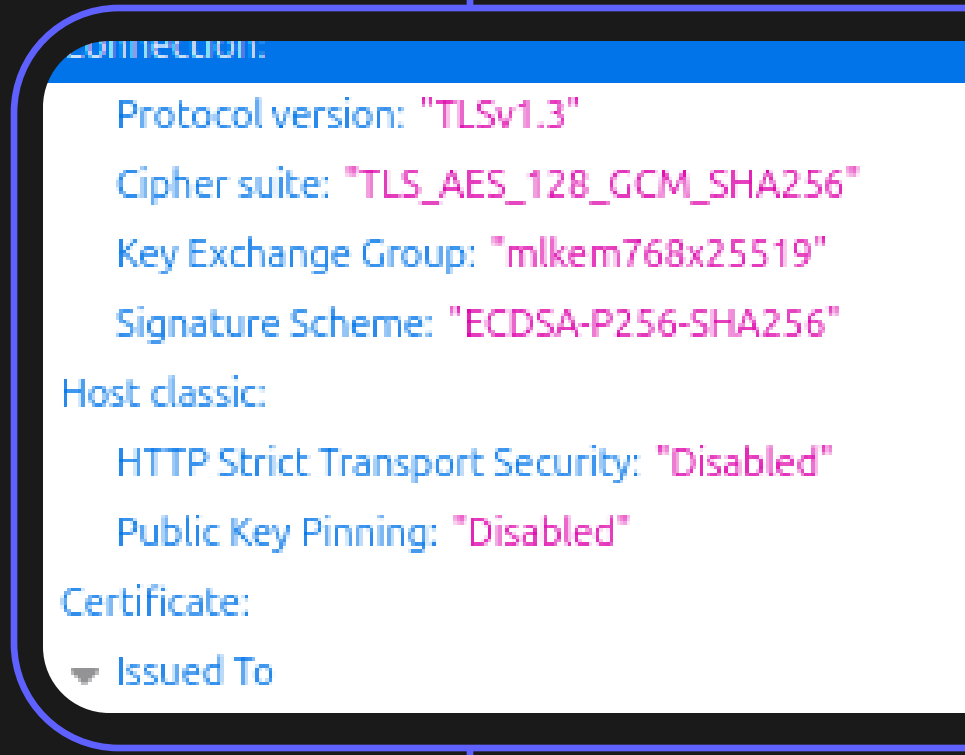
KEYFACTOR

Off-the-shelf TLS – OK

Ubuntu 25.10 - OpenSSL 3.5

- apt install apache2

```
<VirtualHost *:443>
    ServerName www.classic.com
    DocumentRoot /var/www/classic
    SSLEngine on
    SSLCertificateFile /home/user/openssl/apache2/classic-cert.pem
    SSLCertificateKeyFile /home/user/openssl/apache2/classic-key.pem
    SSLCertificateChainFile /home/user/openssl/apache2/classic-
  ca.pem
</VirtualHost>
```

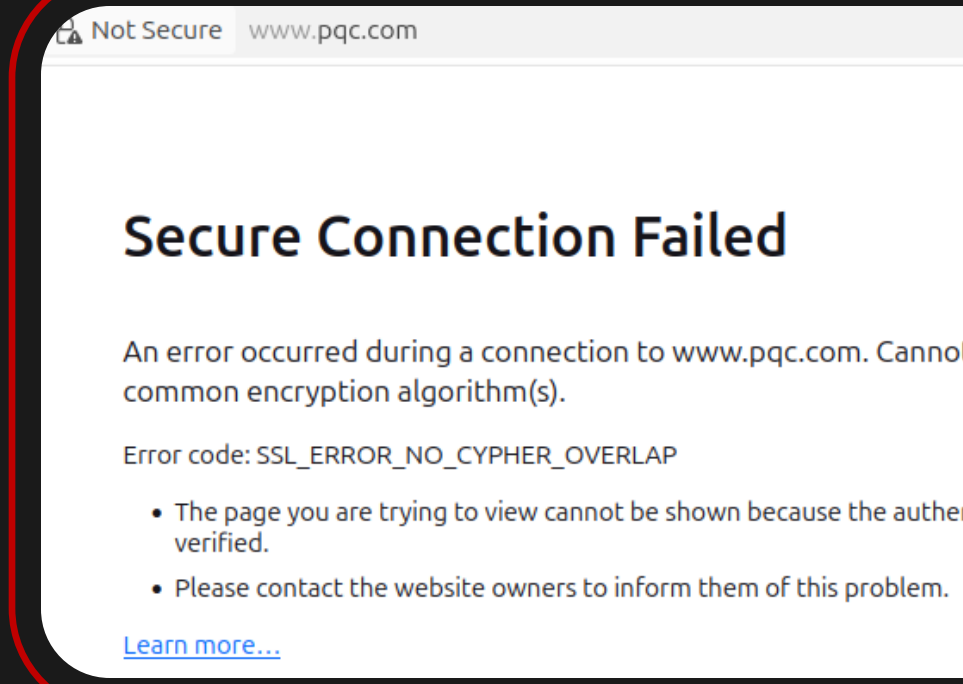


Off-the-shelf TLS – NOK

Ubuntu 25.10 - OpenSSL 3.5

- apt install apache2

```
<VirtualHost *:443>
    ServerName www.pqc.com
    DocumentRoot /var/www/pqc
    SSLEngine on
    SSLCertificateFile /home/user/openssl/apache2/pqc-cert.pem
    SSLCertificateKeyFile /home/user/openssl/apache2/pqc-key.pem
    SSLCertificateChainFile /home/user/openssl/apache2/pqc-ca.pem
</VirtualHost>
```



KEYFACTOR

Off-the-shelf TLS – OK

```
>openssl s_client -CAfile pqc-ca.pem -connect www.pqc.com:443
```

```
Connecting to 127.0.0.1
CONNECTED(00000003)
depth=1 CN=PQC MLDSA Root
verify return:1
depth=0 CN=pqc
---
Certificate chain
0 s:CN=pqc
   i:CN=PQC MLDSA Root
   a:PKEY: ML-DSA-44, 10496 (bit); sigalg: ML-DSA-44
   v:NotBefore: Aug 18 09:46:28 2025 GMT; NotAfter: May 16
   09:46:27 2026 GMT
```

```
Peer signature type: mldsa44
Negotiated TLS1.3 group: X25519MLKEM768
---
SSL handshake has read 11879 bytes and written 1633 bytes
Verification: OK
---
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Protocol: TLSv1.3
Server public key is 10496 bit
Verify return code: 0 (ok)
Post-andshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
```

KEYFACTOR

Off-the-shelf TLS – OK

```
>openssl req -newkey ML-DSA-44 -out mldsa.csr

>sudo certbot certonly --server
https://ejbca.example.com:8442/ejbca/acme/directory -d
www.pqc.com --apache --agree-tos --email
admin@example.com --no-eff-email --csr mldsa.csr --dry-
run



>certbot --help security

...

--key-type {rsa,ecdsa}
```

certbot

View Certificates

Username	**iUJ600rF6c7rRQURTqco53UJv3_bsvIpyxwBC7fOaDQ**
Certificate number	1 of 1
Certificate Type/Version	X.509 v.3
Certificate Serial Number	37B5DE67C0794D1E2240937B7C182CE63E0A5F16
Issuer DN	CN=PQC MLDSA Root
Valid from	2025-08-22 16:51:52+02:00
Valid to	2026-08-22 16:51:46+02:00
Subject DN	**CN=www.pqc.com**
Subject Alternative Name	
Subject Directory Attributes	None
Public key	ML-DSA-44 (128 bits): EB4019044694B70A7E7B511011D95522B438174470FFFDC...
Alternative Public key	- (-): -
Basic constraints	**End Entity**
Key usage	Digital Signature
Extended key usage	Client Authentication,Server Authentication
Name constraints	No
Authority Information Access	No
Qualified Certificates Statements	No
Certificate Transparency SCTs	No
Signature Algorithm	ML-DSA-44
Alternative Signature Algorithm	-
Fingerprint SHA-256	CBC28576B5C095AC7D5972927D78C015 3A723EE6DB4E2050408778D0670EA322
Fingerprint SHA-1	07AFC8653463C52834D0D88B256B6E3239C07505
Revoked	No

Republish | Unspecified | Revoke

KEYFACTOR

Off-the-shelf TLS – OK

```
>apt install curl (8.14.1)

>snap install curl (8.16.0)

>curl --cacert pqc-ca.pem https://www.pqc.com:443 -iv
```

```
* Host www.pqc.com:443 was resolved.
* IPv6: (none)
* IPv4: 127.0.0.1
*   Trying 127.0.0.1:443...
* ALPN: curl offers h2,http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
*  CAfile: /home/user/pqc-ca.pem
```

```
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384 /
  X25519MLKEM768 / id-ml-dsa-44
* Server certificate:
*  subject: CN=pqc
*  subjectAltName: host "www.pqc.com" matched cert's "www.pqc.com
*  issuer: CN=PQC MLDSA Root
*  SSL certificate verify ok.
*   Certificate level 0: Public key type ML-DSA-44 (10496/128
  Bits/secBits), signed using ML-DSA-44
*   Certificate level 1: Public key type ML-DSA-44 (10496/128
  Bits/secBits), signed using ML-DSA-44
* Connected to www.pqc.com (127.0.0.1) port 443
```

KEYFACTOR

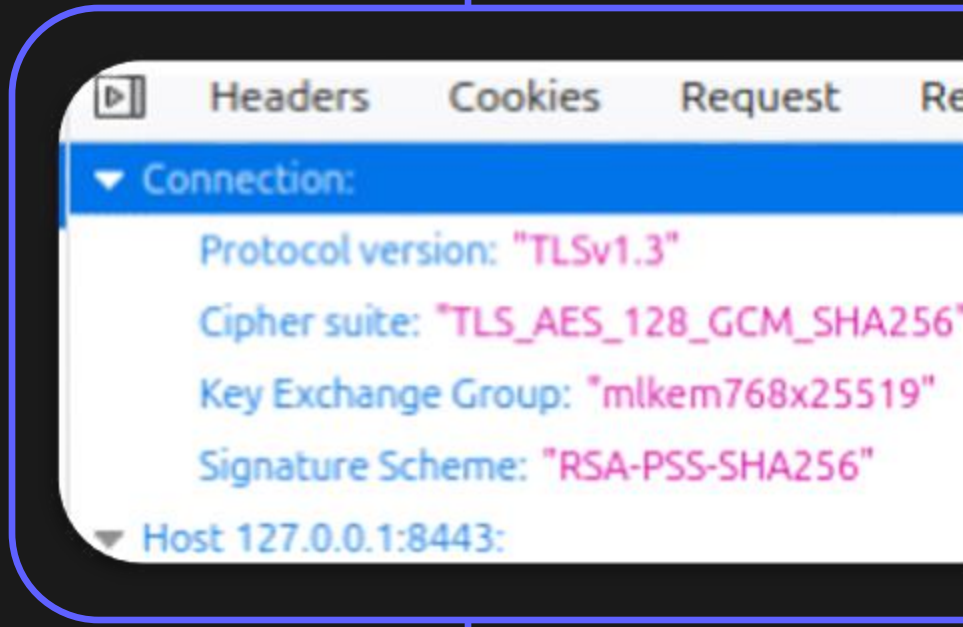
Off-the-shelf TLS – OK

```
>vim /usr/lib/jvm/java-17-openjdk-amd64/conf/security/java.security
```

```
security.provider.1=BC
security.provider.2=BCJSSE BC
```

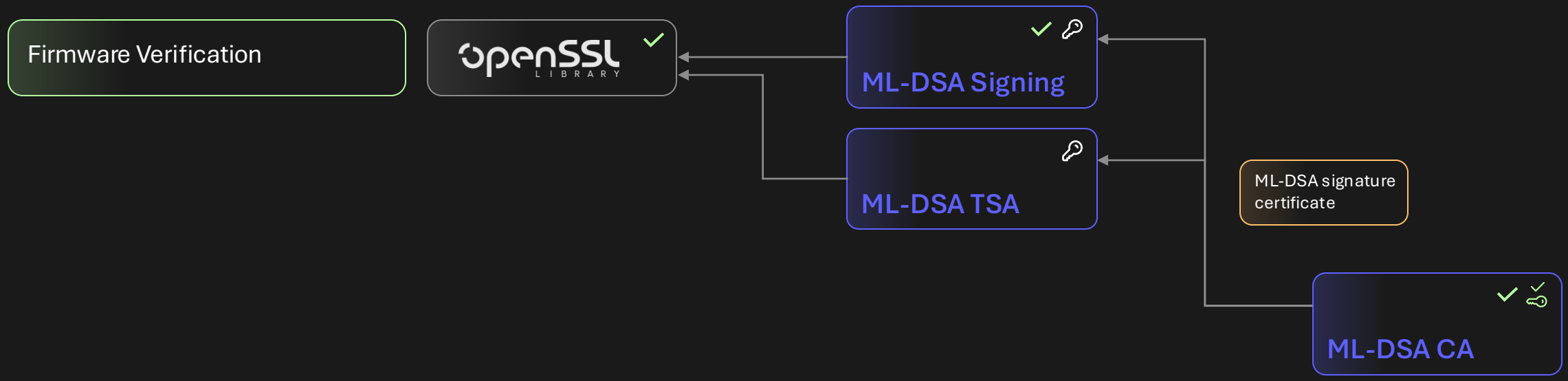
```
>vim /opt/wildfly/bin/standalone.conf
```

```
JAVA_OPTS="$JAVA_OPTS --module-path=/opt/bc-module-jars"
```



KEYFACTOR

Signing



KEYFACTOR

Off-the-shelf CMS

CMS

- attached or detached signatures
- signed or unsigned attributes

```
CMSSignedDataGenerator gen = new CMSSignedDataGenerator();
final ContentSigner signer = new
JcaContentSignerBuilder(ML-DSA).build(private);
```

```
> openssl cms -verify -in something-to-sign.txt-
  detached.p7s -inform DER -CAfile
  MLDSA44.cacert.pem -content something-to-
  sign.txt -binary
Here is something to sign
CMS Verification successful
```

KEYFACTOR

Stumbling blocks

Time Stamping

```
TimeStampResponseGenerator tsRespGen = new TimeStampResponseGenerator(tsTokenGen,
  TSPAlgorithms.ALLOWED);
  TimeStampResponse tsResp = tsRespGen.generate(request, new BigInteger("23"), new
  Date());
  byte[] tsrBytes = tsResp.getEncoded();
```

```
> openssl ts -verify -in test.tsr -data test.txt -CAfile TSA.cacert.pem
Verification: FAILED
80AB6C2187720000:error:1780006D:time stamp
  routines:TS_RESP_verify_signature:signature
  failure:crypto/ts/ts_rsp_verify.c:148:
```

KEYFACTOR

Stumbling blocks

Compliance

- draft-ietf-lamps-cms-ml-dsa – SHA-2, SHA-3, SHAKE
- CNSA 2.0 profile – SHA-2
- Default(ed) to digest algorithm used in signature (SHAKE for ML-DSA, SHA512 for SHA512WithRSA… – all variant work but CNSA compliance may be a thing

```
CMSSignedDataGenerator gen = new CMSSignedDataGenerator();

final ContentSigner signer = new JcaContentSignerBuilder(ML-DSA).build(private);

final JcaDigestCalculatorProviderBuilder dbuilder = new JcaDigestCalculatorProviderBuilder();

final JcaSignerInfoGeneratorBuilder sbuilder = new
  JcaSignerInfoGeneratorBuilder(dbuilder.build());

→ sbuilder.setContentDigest(new AlgorithmIdentifier(digestAlgOID));
```

KEYFACTOR

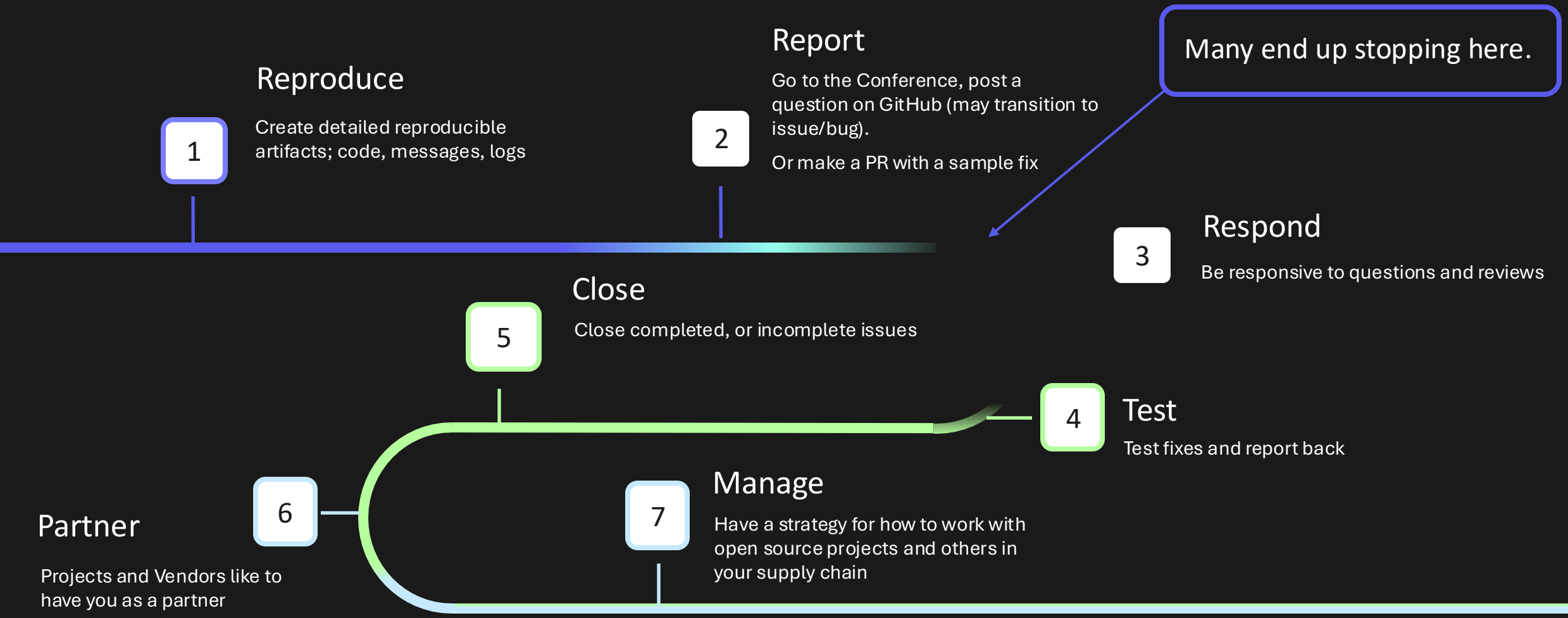
Stumbling blocks

```
>openssl cms -sign –nodetach –outform DER -in msg.txt -out out.msg -
 signer rsa-cert.pem -inkey rsa-priv.pem
```

```
>openssl cms -sign -nodetach -outform DER -in msg.txt -text -out out.msg
 -signer mldsa-cert.pem -inkey mldsa-priv.pem
80DB4484BE7E0000:error:17000080:CMS routines:CMS_add1_signer:no default
 digest:crypto/cms/cms_sd.c:405:pkey nid=-1
```

```
>openssl cms -sign -nodetach -outform DER -in msg.txt -text -out out.msg
 -signer mldsa-cert.pem -inkey mldsa-priv.pem -md SHA512/SHAKE256
```

Collaborate and Resolve



Protocols and Crypto Agility

CMP

Plain signatures of small data

CRMF (incl ML-KEM) or PKCS#10 CSRs

EST

TLS and plain, small signatures

PKCS#10 CSRs (ML-DSA only)

ACME

TLS and small challenges

PKCS#10 CSRs (ML-DSA only)

~~SCEP~~

Signing and encryption based on CMS

Requires standards update

KEYFACTOR

Summary

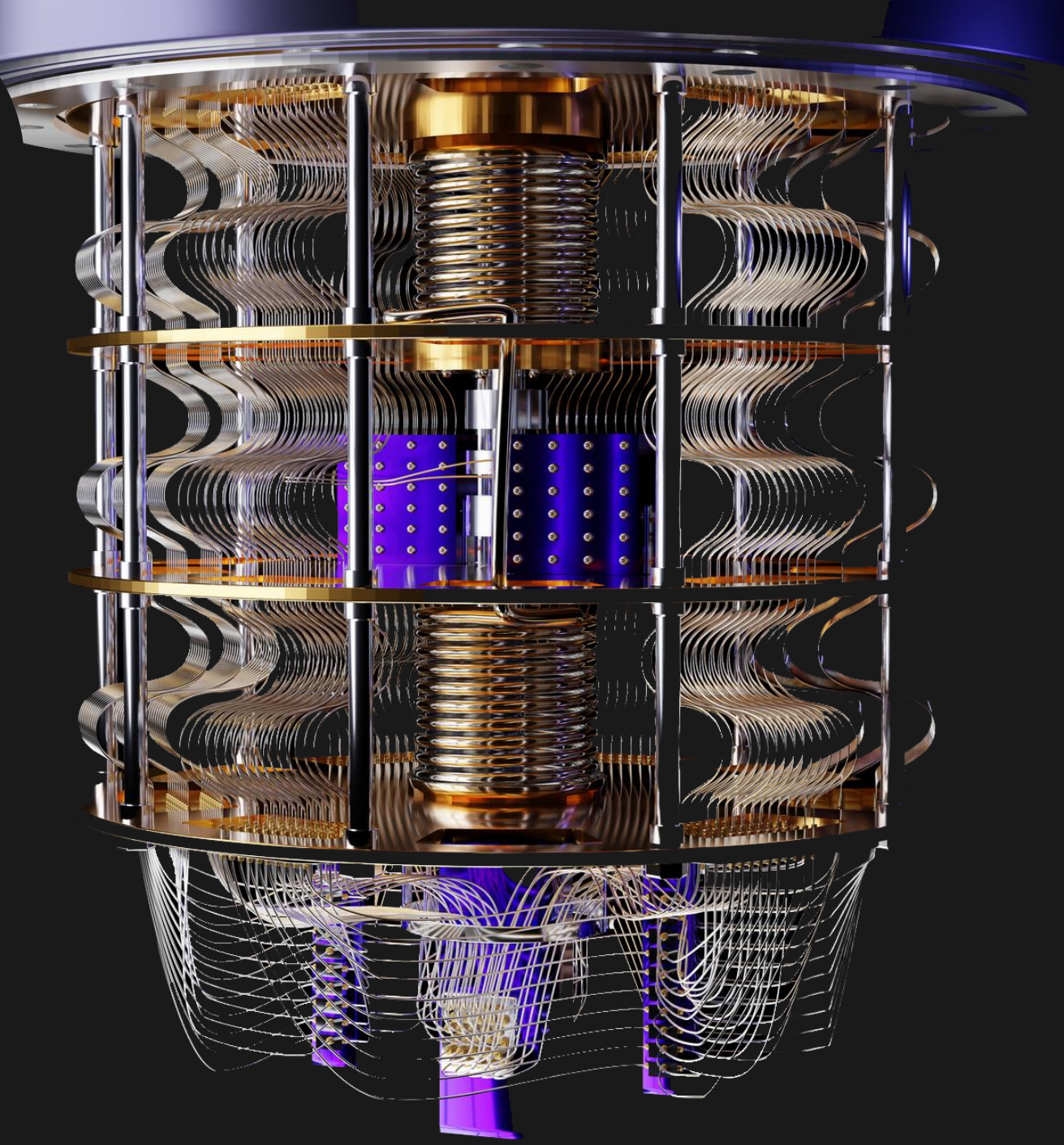
- Is there time to procrastinate? ✕ No
- Is it easy to start a PoC? ✓ Yes
- Will you find issues along the way? ✓ Yes
- Is everything production ready? ✕ No
- Can you contribute ✓ Yes
- Is it expensive to start testing ✕ No
- Hybrids and Composites ⚠ Not out of the box

KEYFACTOR

Interoperability in Action

Post-Quantum Cryptography

Tomas Gustavsson
Chief PKI Officer



KEYFACTOR