Partnership for Cyber Security Innovation

PQC and the PCSI

EU PQC conference, Dec 4th 2025

Manon de Vries (TNO)
Gamze Tillem (ING)

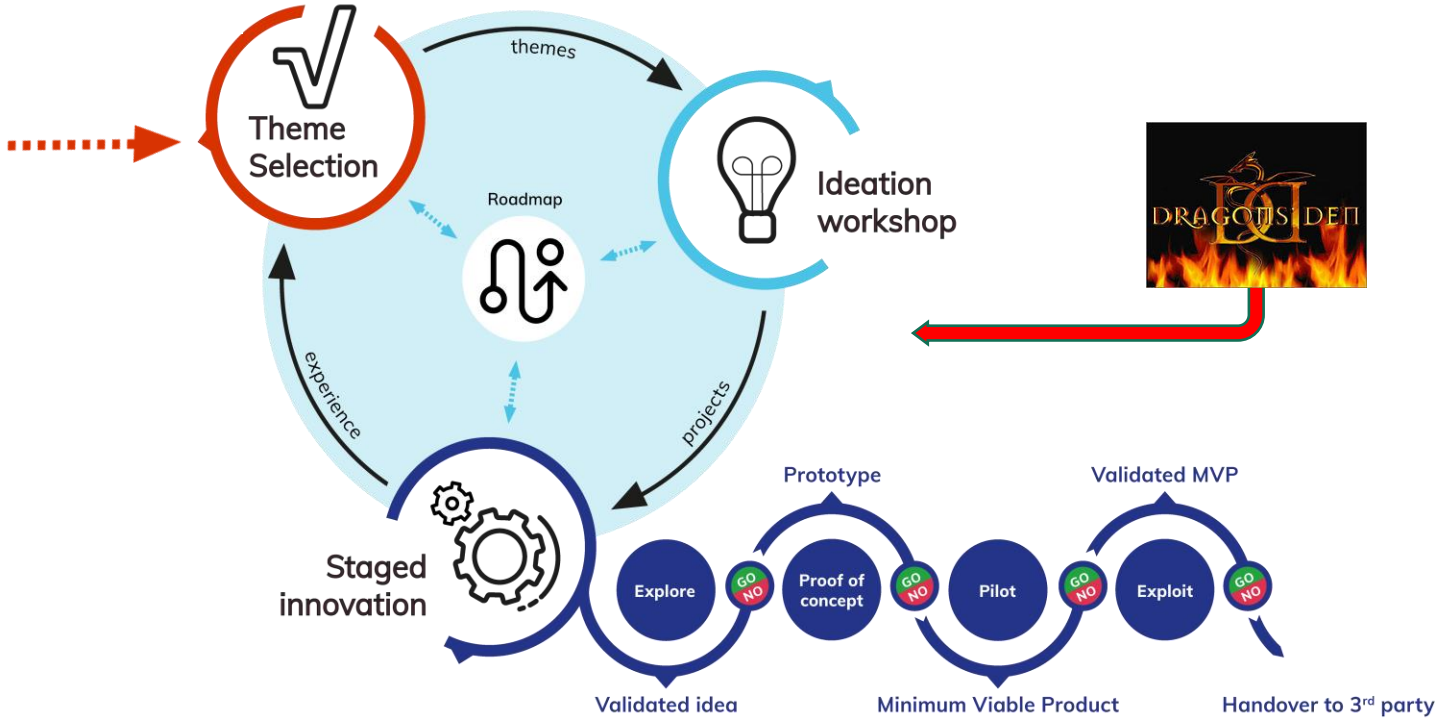
The Team

PCSI PQC Benchmarking project
April 2024 – August 2025

PCSI PQC Workgroup
Sept 2025 onward

How we work

Collaborative '*orchestrating innovation*' way of working



The start: ideation day

Trend Every Dutch organization must undergo the same complicated migration process to quantum safe cryptography.

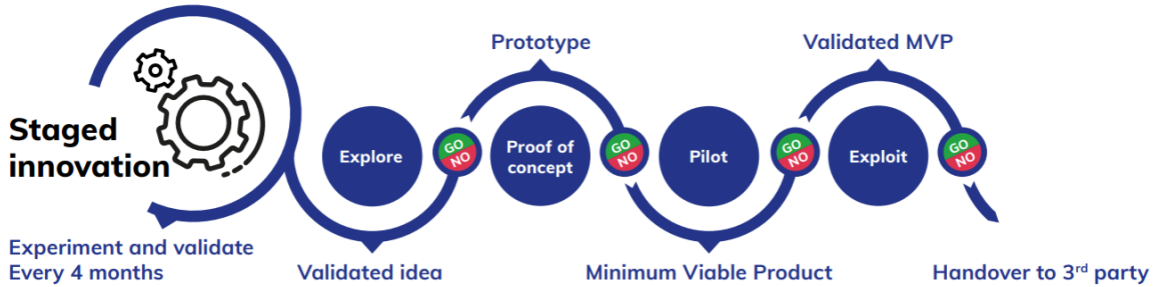
Pain point: Lack of practical insights leads to hesitance to start

Available benchmarks are academic or very limited in scope

Proposal: Let’s gain experience!

- Migrate one application each
- Benchmark performance of multiple algorithmic options
- Share experiences, pitfalls, difficulties and successes

PQC Benchmark project vision



Explore

Find existing benchmarks

Define use cases

Agree on benchmark method

POC

Benchmark first system

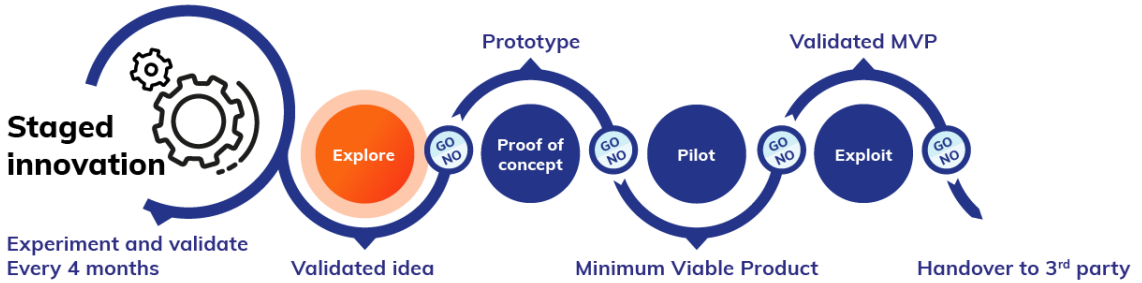
Pilot

Use lessons learned to benchmark other use cases

Exploit

Disseminate acquired information and experiences

Explore phase results



Explore

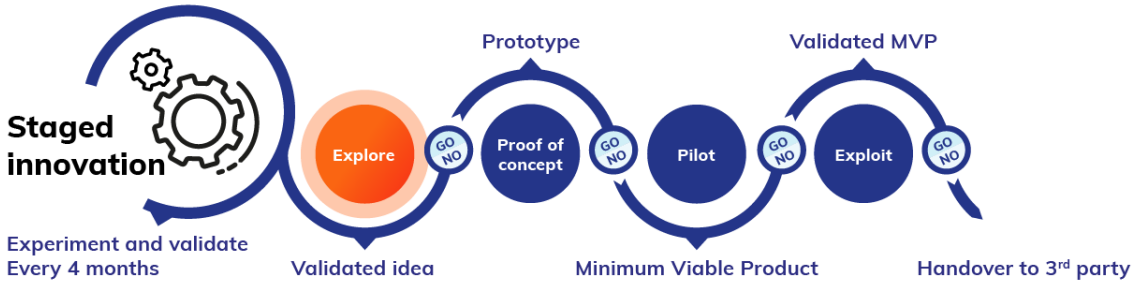
Find existing benchmarks

Define use cases

Agree on benchmark method

Unexpectedly hard:
Finding suitable applications at partners
Arranging time and resources for POC

Explore phase results



Benchmark method

PQC Algorithms (NIST level 1 or equivalent):

- Key exchange:
  - **A. Crystals-Kyber (ML-KEM),**
  - B. Frodo-KEM,
  - C. McEliece
- Digital signature:
  - **D. Crystals-Dilithium (ML-DSA)**
  - E. Falcon
  - **F. Sphincs+ (SLH-DSA)**

Metrics:

- CPU usage
- Storage
- Memory
- Network information(bandwidth, package size, package drops, latency)
- Time (connections/signings/key exchanges per second)

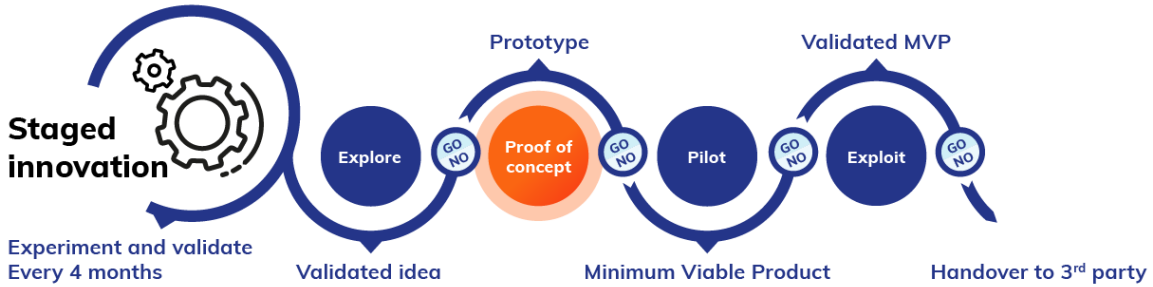
Test cases:

	KEM	DSA	Algorithms to test
0.	Current (RSA)	Current (RSA)	First baseline
1.	ECC	ECC	Second baseline
2.	PQC	ECC	A,B,C
3.	ECC	PQC	D,E,F
4.	PQC	PQC	All combinations of A-C with D-F
5.	(ECC+PQC)	ECC	A,B,C, hybrid with ECC
6.	ECC	(ECC+PQC)	D,E,F, hybrid with ECC
7.	(ECC+PQC)	(ECC+PQC)	All combinations of A-C with D-F

Information gathered:

- Performance indicators
- System details (hardware, platform, library version, etc)
- Algorithm details currently being tested (OID, key size, sphincs type)
- Encountered problems, bottlenecks, learned lessons about implementation, adjustments required outside of replacing the algorithm

POC phase



POC

Benchmark first system

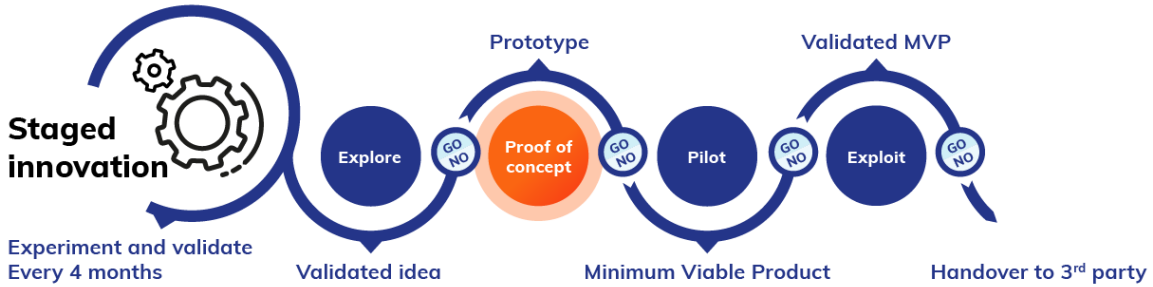
In-house application chosen at ABN Amro

- No vendor dependency, capable programmers
- High reliance on quick key exchange (done many times)

However, life is what happens when your busy making other plans

- Unforeseen unavailability of programming team: migration work done by TNO
- Benefit: experience in outsourcing of migration!

POC phase results



POC

Benchmark first system

293 files changed, 3918 lines of code adjusted just to support PQC key exchange

- Most changes to make code crypto agile

An unexpected lesson in crypto agility:

- Cryptography was used in many locations in the code
- Abstracting the algorithm away was less time consuming
- Also makes the code more agile for future cryptographic changes
- Afterwards, new algorithms only requires adding the algorithm code itself and changing a setting

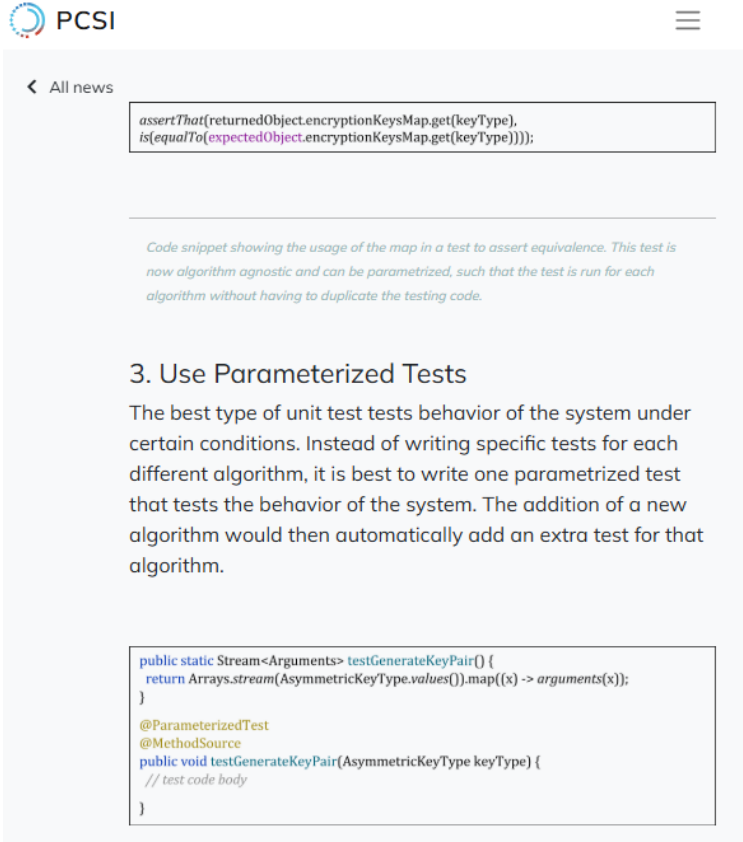
```
private void initKeys() {
  keyPair1 = getRSA(0);
  keyPair2 = getRSA(1);
}
```

```
private void initKeys() {
  keyPair1 = getKeyPair(RSA, 0);
  keyPair2 = getKeyPair(RSA, 1);
}
```

Technical details ABN Amro

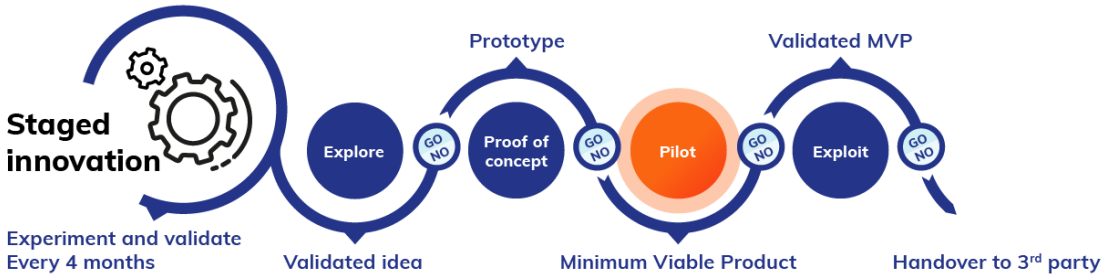
See our blog!

https://pcsi.nl/en/news/dont-get-entangled-in-pqc-embrace-crypto-agility-for-smooth-code-migration/



PCSI is a collaboration of ING Shell ABN·AMRO Belastingdienst TNO achmea

Pilot phase



Pilot

Use lessons learned to benchmark other use cases

Belastingdienst:
- High performance application
- Vendor dependency

ING:
- Critical component PKI
- Including hardware support

Belastingdienst use case on Certificates

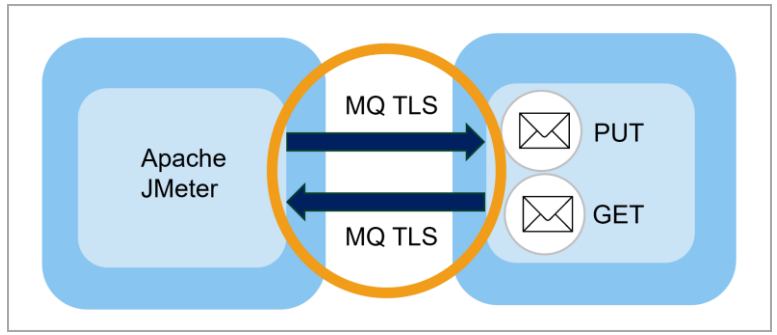
Benchmark a message queueing application in combination with PQC certificates

Considerations:

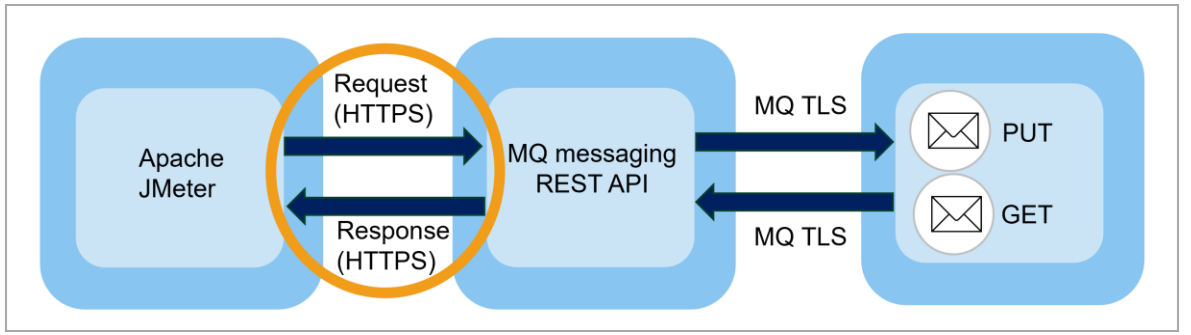
- Important high-performance component for many IT companies
- Closely tied to daily work for many teams
- Most likely resources already available, like (parts of) tooling, test environment, knowledge

Belastingdienst use case on Certificates

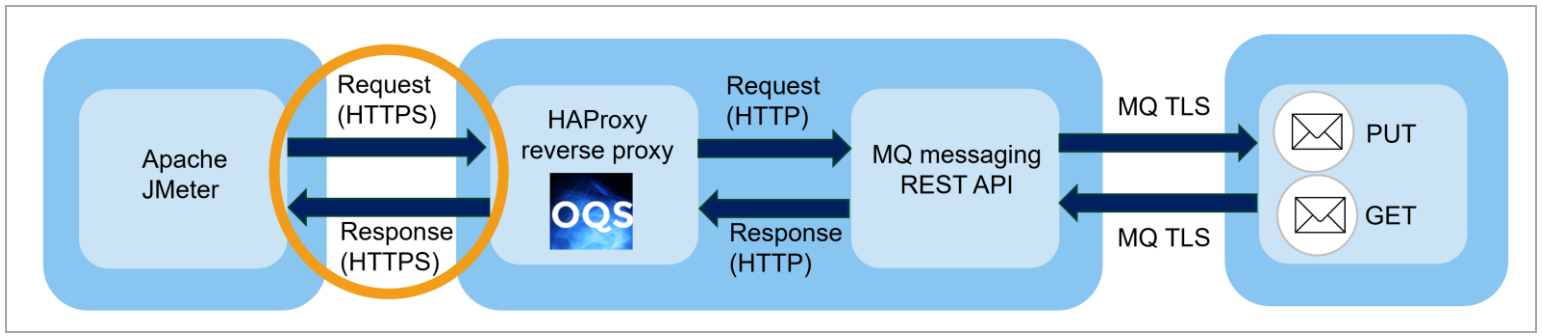
Original idea:



First alternative:



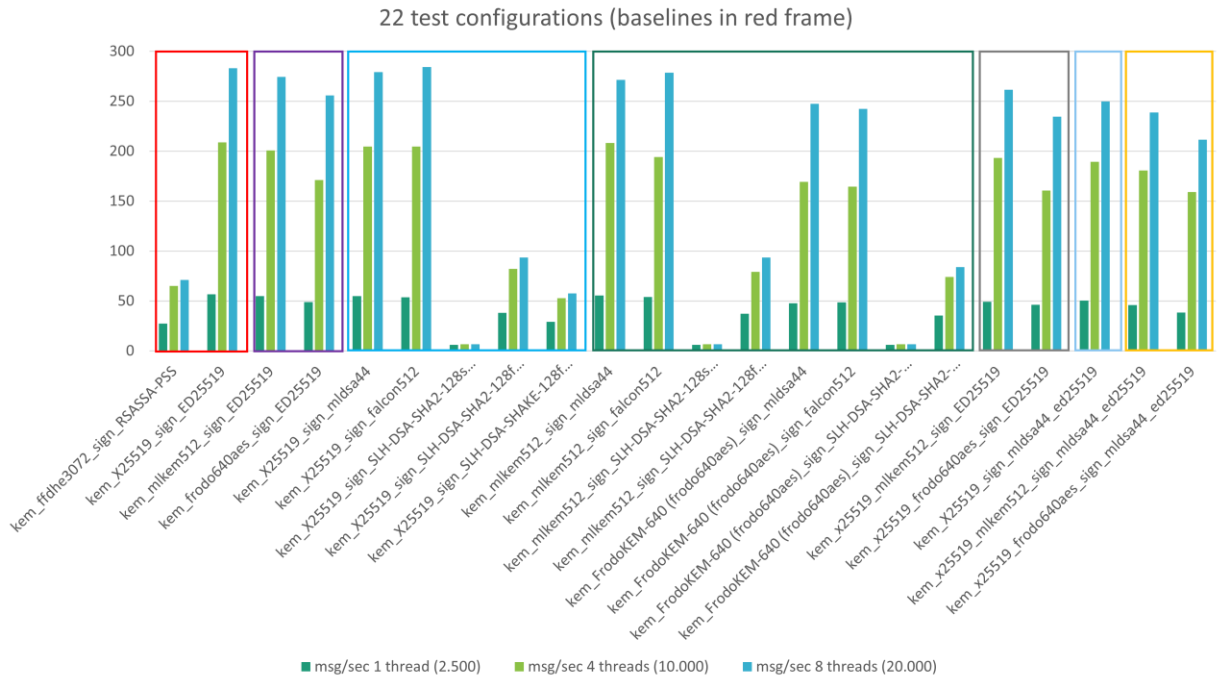
Final setup:



Decoupled architecture using reverse proxy from framework OpenQuantumSafe *).
SSL offloading for the time being until vendors support PQC. Focus on baseline benchmarking and automation. Later proxy can be removed and test re-run.

= PQC configured connection

*) https://openquantumsafe.org



Technical details Belastingdienst

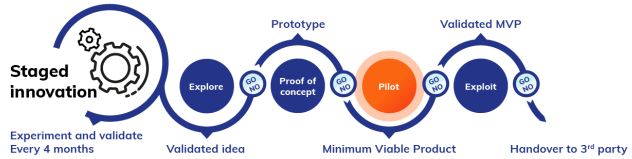
All results are published in the architecture blog:

https://pcsi.nl/en/news/how-to-get-un-stuck-by-architecting-for-resilience/

Detailed results in de blog addendum:

https://pcsi.nl/en/news/how-to-get-un-stuck-by-architecting-for-resilience-2/

PCSI is a collaboration of ING Shell ABN·AMRO Belastingdienst TNO achmea



Pilot phase results Belastingdienst

Pilot

Use lessons learned to benchmark other use cases

➢ Don’t be delayed by vendor products not being quantum ready.

Results (on system performance):

- Not as bad as expected! 20% performance hit for full hybrid replacement

Benchmark Results – Message size

Even though TLS 1.3 message size in this test did not seem to matter much, it can be important for other use cases.

FrodoKEM and SLH-DSA with speed optimization produce significantly bigger messages.

Hybrid solutions with ML-KEM and ML-DSA have 2-6x bigger messages, with the certificate message* increasing the most.

KEM algorithms influence Client hello and Server hello messages

	Baseline		KEM PQC		KEM hybrid	
	1. RSA	2. ECC	3. MLKEM	4. FrodoKEM	18. ECC-MLKEM	19. ECC-FrodoKEM
Client hello	778	512	1.194	10.010	1.226	10.042
Server hello	474	122	858	9.810	890	9.842

DSA algorithms influence Certificate and Cert Verify messages

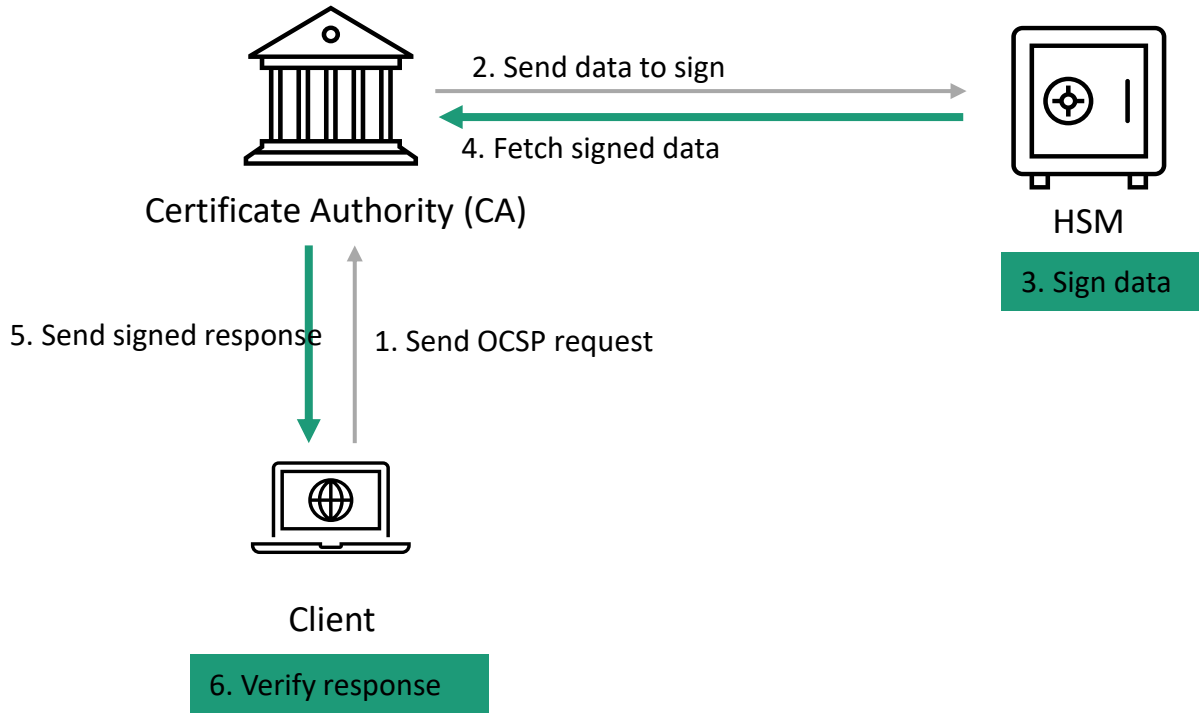
	Baseline		Signature PQC					Sign. hybrid
	1. RSA	2. ECC	5. MLDSA	6. Falcon	7. SLHDSA-SHA2-128s	8. SLHDSA-SHA2-128f	9. SLHDSA-Shake-128f	20. MLDSA_ECC
Certificate*	1.310	803	4.891	4.472	3.604	3.604	3.604	4.937
CERT verify	520	72	2.428	664	7.864	17.096	17.096	2.504

*A self signed certificate was used containing a PQC key, but signed classically. A full PQC certificate will be significantly larger!

ING Use case on PKI

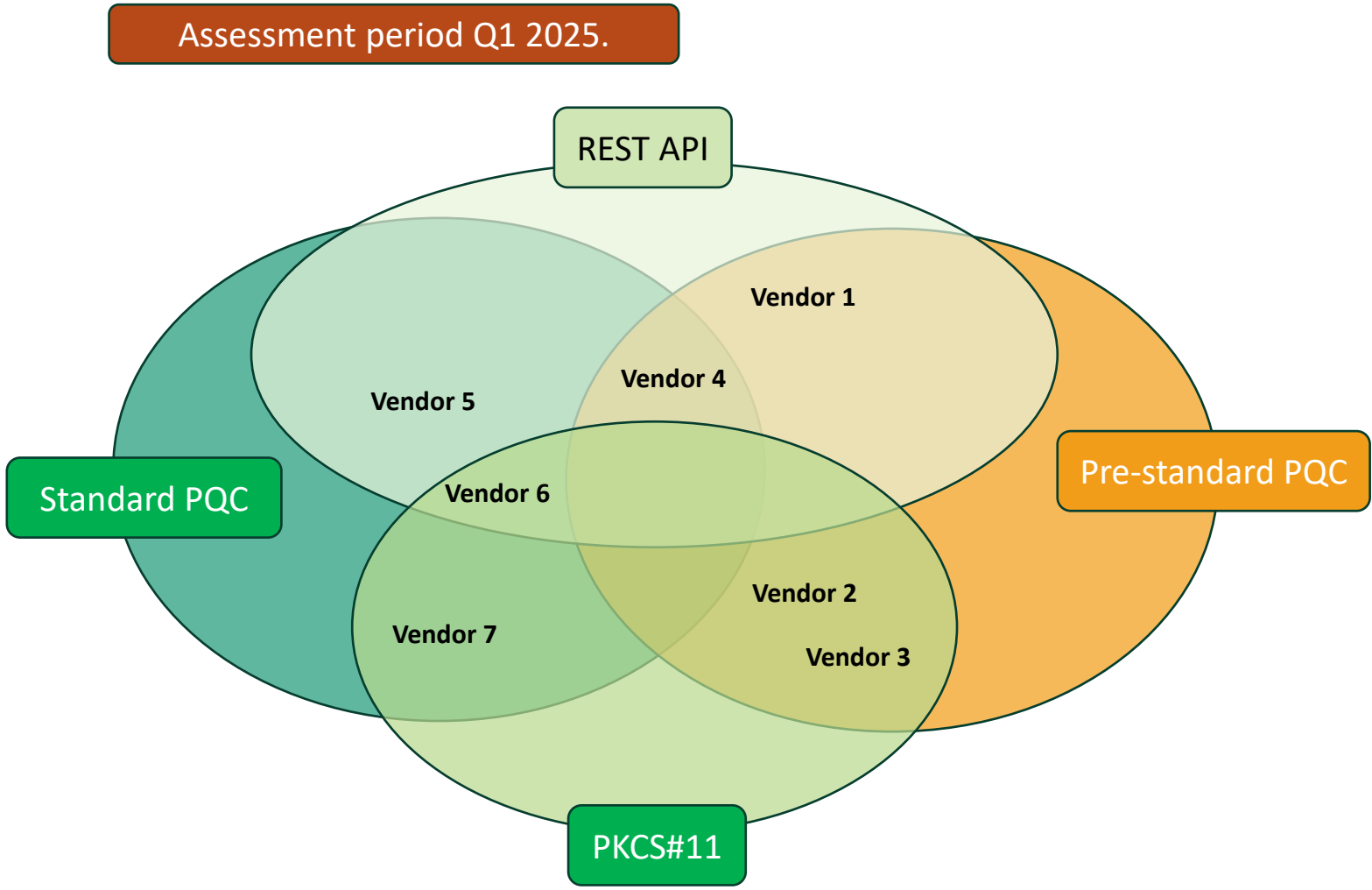
Benchmarking OCSP (Online Certificate Status Protocol) Performance

- PKI is the backbone of trust in most organisations
- OCSP is a protocol used in the revocation of certificates
- This experiment involved:
  - Signing only
  - Hardware dependency: storing keys in HSMs.
- Performance matters in OCSP



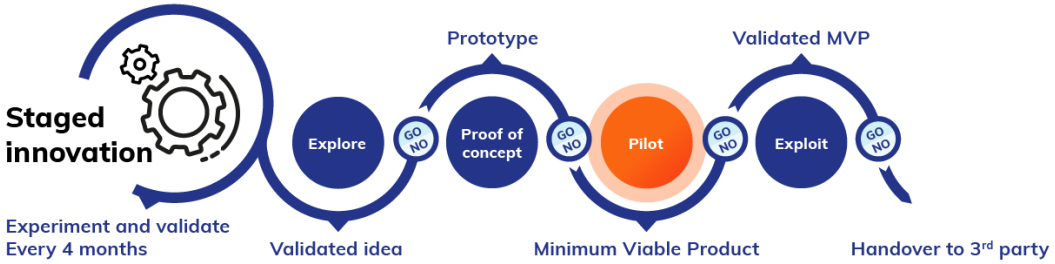
Benchmark Results

- Vendor readiness
  - *7 HSM vendors were contacted*
  - *3 vendors only provided **older** versions of ML-DSA (ML-DSA-IPD or Crystals-Dilithium), not the standardized version ML-DSA*
  - *1 requires additional configuration that couldn't be achieved within PoC timelines*
  - *PKCS#11 support often not (yet) ready*
  - *Hardware acceleration often not yet available*
- Protocol readiness
  - *No hybrid support in OCSP protocol*
- Library readiness
  - *Creating CSRs for hybrid not supported by OpenSSL*



*Vendor names are anonymized, NDA in place.

Pilot phase lessons learned ING



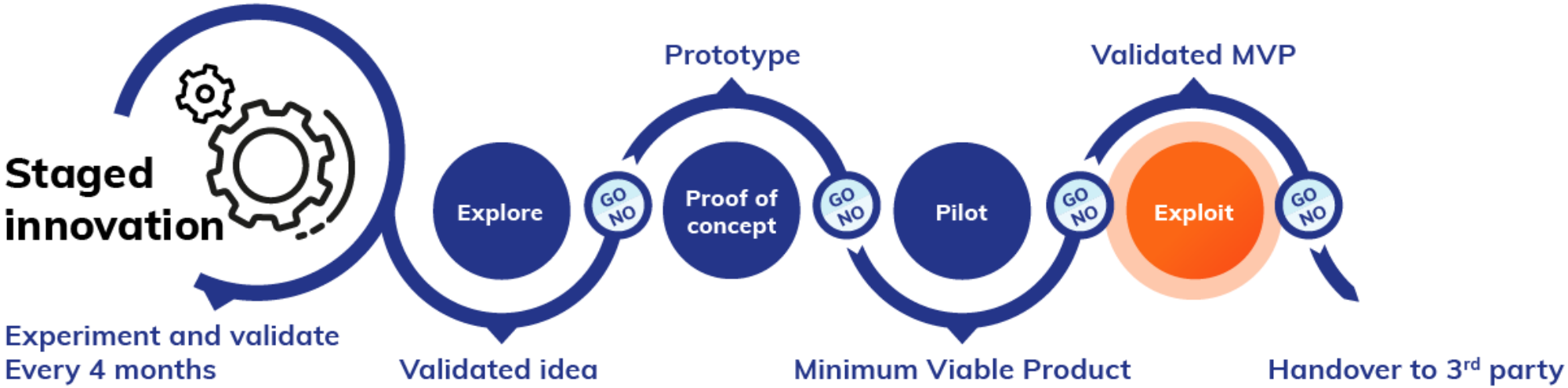
Pilot

Use lessons learned to benchmark other use cases

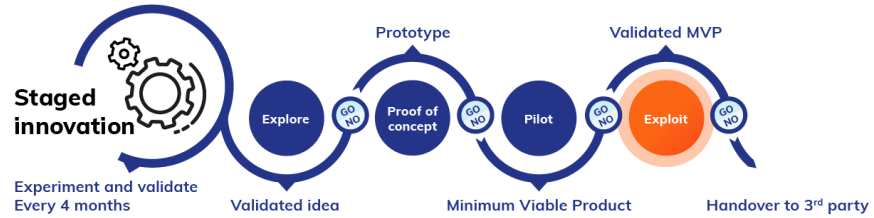
- The definition of “PQC-ready” might differ greatly between you and your vendor.
- Asking the right questions helps.
- Sneak peek: 3 out of 7 vendors only supported older versions of ML-DSA, not the standardized one.*

- Vendors are not ready yet with standards and protocols.
- The PKI protocol standards and libraries are still in early stage in the PQC support (lack of support for hybrid).

*Snapshot of moment of testing (Q1 2025): wider support is on most vendors’ roadmaps.


Staged innovation
Explore
GO NO
Prototype
Proof of concept
GO NO
Pilot
GO NO
Validated MVP
Exploit
GO NO
Experiment and validate
Every 4 months
Validated idea
Minimum Viable Product
Handover to 3rd party

Exploit phase



Exploit

Disseminate acquired information and experiences

Blog series

On: The essential role of the manager
On: Code agility and the programming side of the migration
On: Vendor management and HSM readiness
On: Architecture considerations, alternative strategies and performance impact

Why work together on experiments?

Experiments

- Highlight organisational hurdles
- Creates internal knowledge and awareness: kickstarts the migration
- When doing early: motivates vendors

Cooperation

- Lowers time, risk and costs
- Helps identify different strategies, blind spots
- Accelerates knowledge gain
- Creates enthusiasm and
- Helps to keep momentum when things are tough

This is the end… or not?

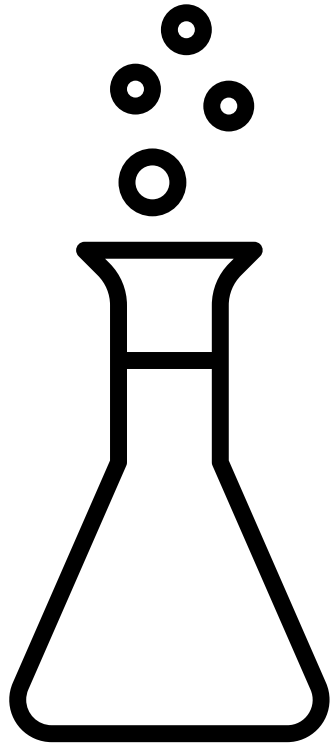
We have continued in a structural collaboration on PQC migration

The PCSI PQC Workgroup:

PQC Workgroup sneak preview

- Monthly knowledge sharing sessions
  - Oct: Roadmaps
  - Nov: Inventory making
  - Dec: Vendor assessment

- Learning by doing:
- Upcoming experiments:
- **Cryptographic Inventory**



Questions?

PCSI is a collaboration of

ING

ABN·AMRO

Belastingdienst

TNO

achmea

www.pcsi.nl

follow us