SIEMENS Healthineers

Quantum-Resistant Healthcare



Katarina Amrichova, Siemens Healthineers

Katarina Amrichova



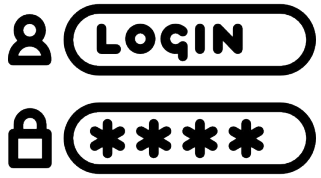
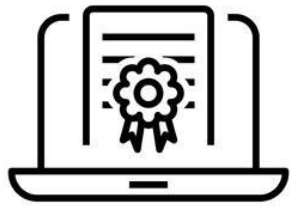
Siemens Healthineers

Senior Key Expert

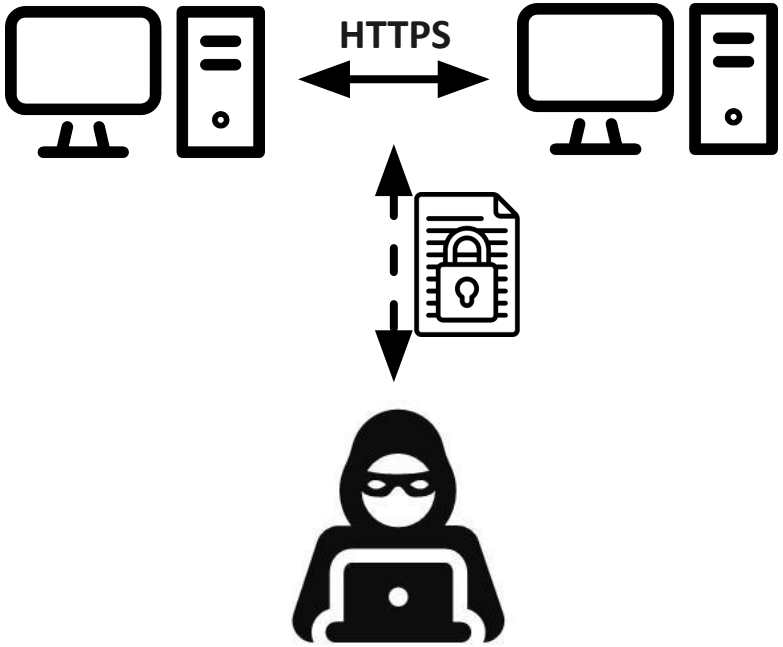
katarina.amrichova@siemens-healthineers.com

Cryptography Threatened by Quantum Computers

SIEMENS Healthineers

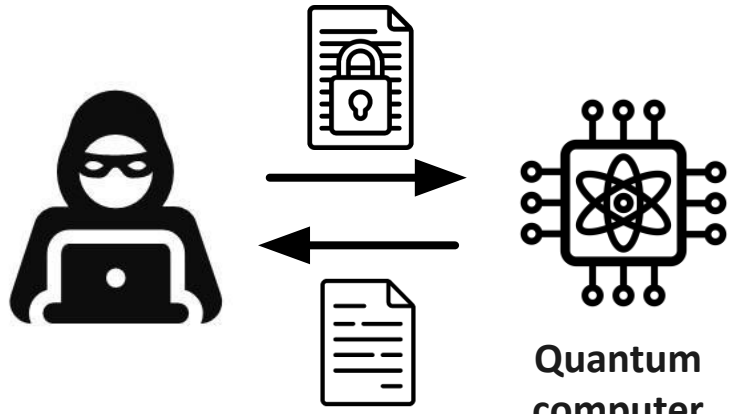
NOT threatened	Threatened
 Hashing algorithms (e.g. password hashes)	 Asymmetric key exchange (e.g. TLS connection)
 Symmetric encryption (e.g. encrypted documents)	 Asymmetric signing (e.g. software signing certificates)

Harvest Now Decrypt Later



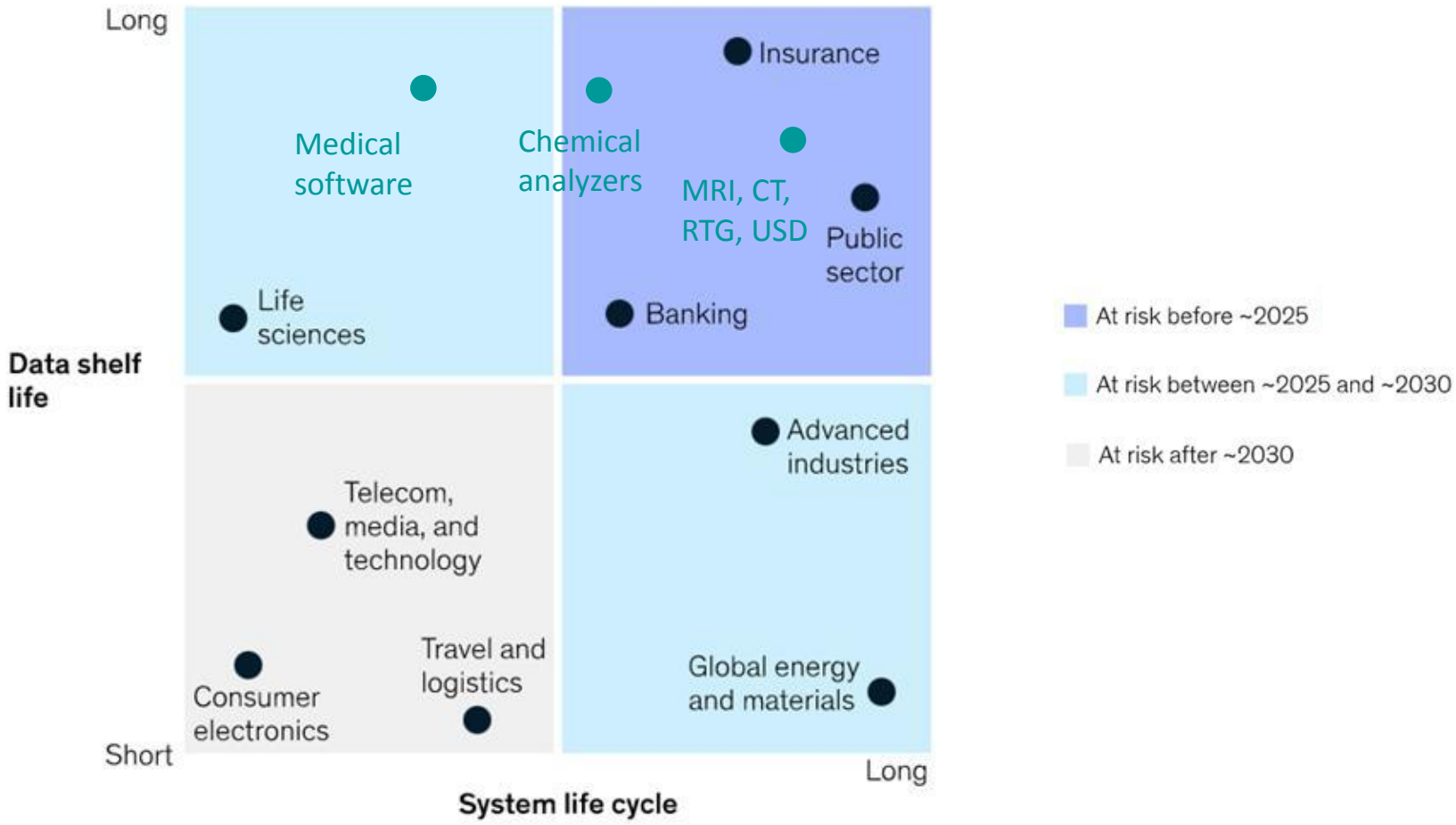
Intercepting HTTPS communication

X years later

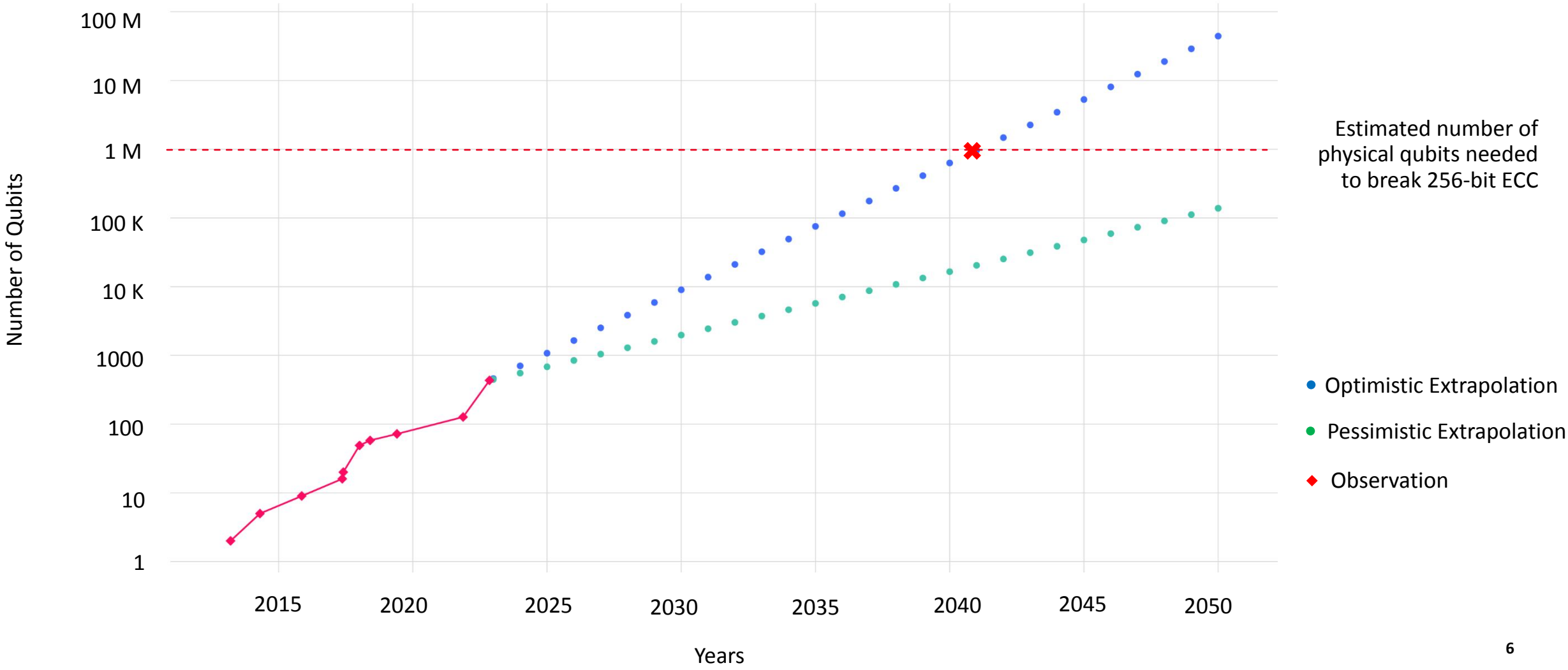


Decrypting the intercepted communication

Risk of quantum-powered attack by industry

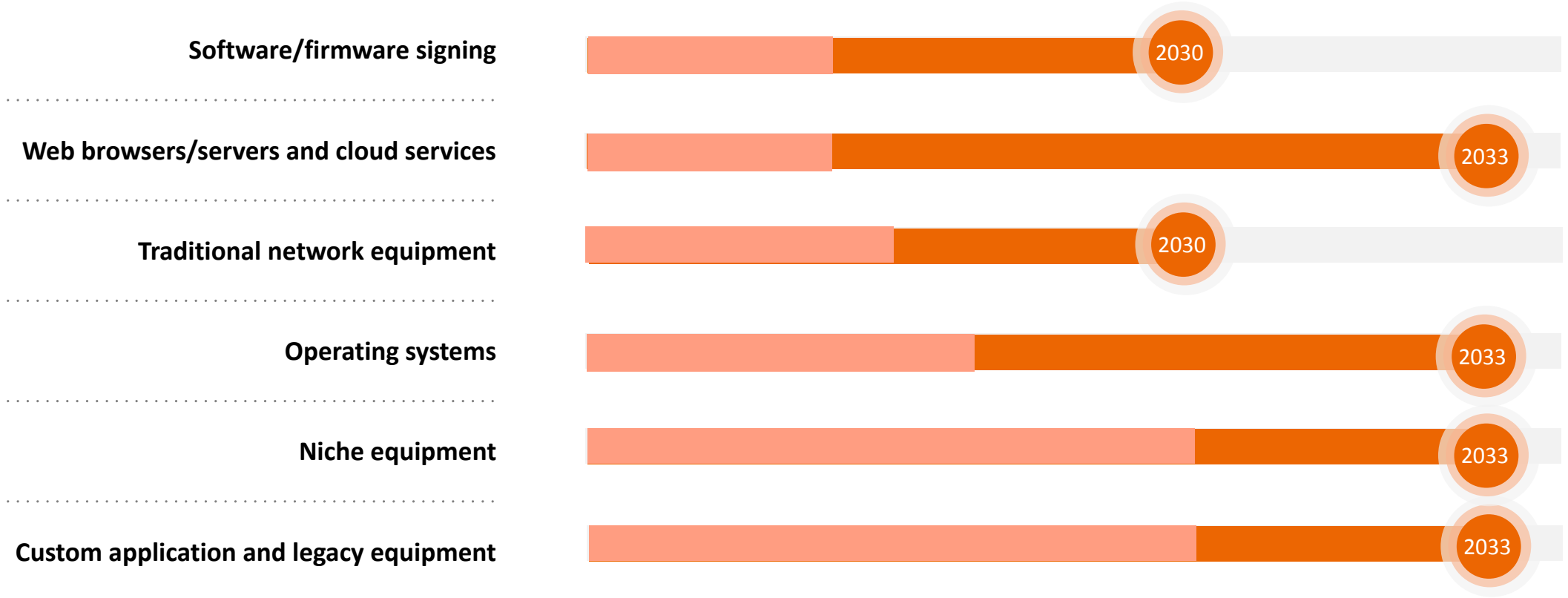


Extrapolation of the Rising Number of Qubits



https://quantum.microsoft.com/en-us/tools/quantum-cryptography

CNSA 2.0 Transition Timeline



PQC Roadmap for Healthineers

SIEMENS Healthineers

01 Spreading awareness about PQC

02 Implementing a PQC prototype in a Healthineers product

03 Creating a CBOM and performing TRA of quantum threat

04 Preparing a directive and a strategy document about PQC

05 Implementing PQC in multiple Healthineers products

SIEMENS Healthineers

Spreading awareness about PQC

01

02

Implementing a PQC prototype in a Healthineers product

03

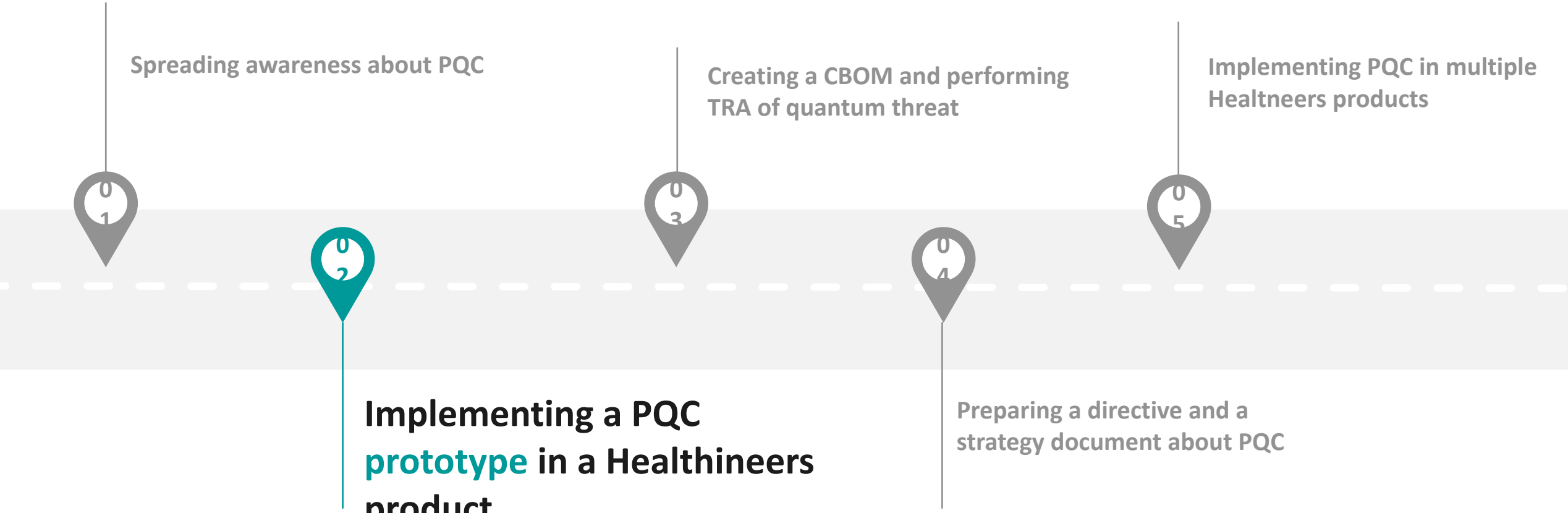
Creating a CBOM and performing TRA of quantum threat

04

Preparing a directive and a strategy document about PQC

05

Implementing PQC in multiple Healthineers products



SIEMENS Healthineers

01 Spreading awareness about PQC

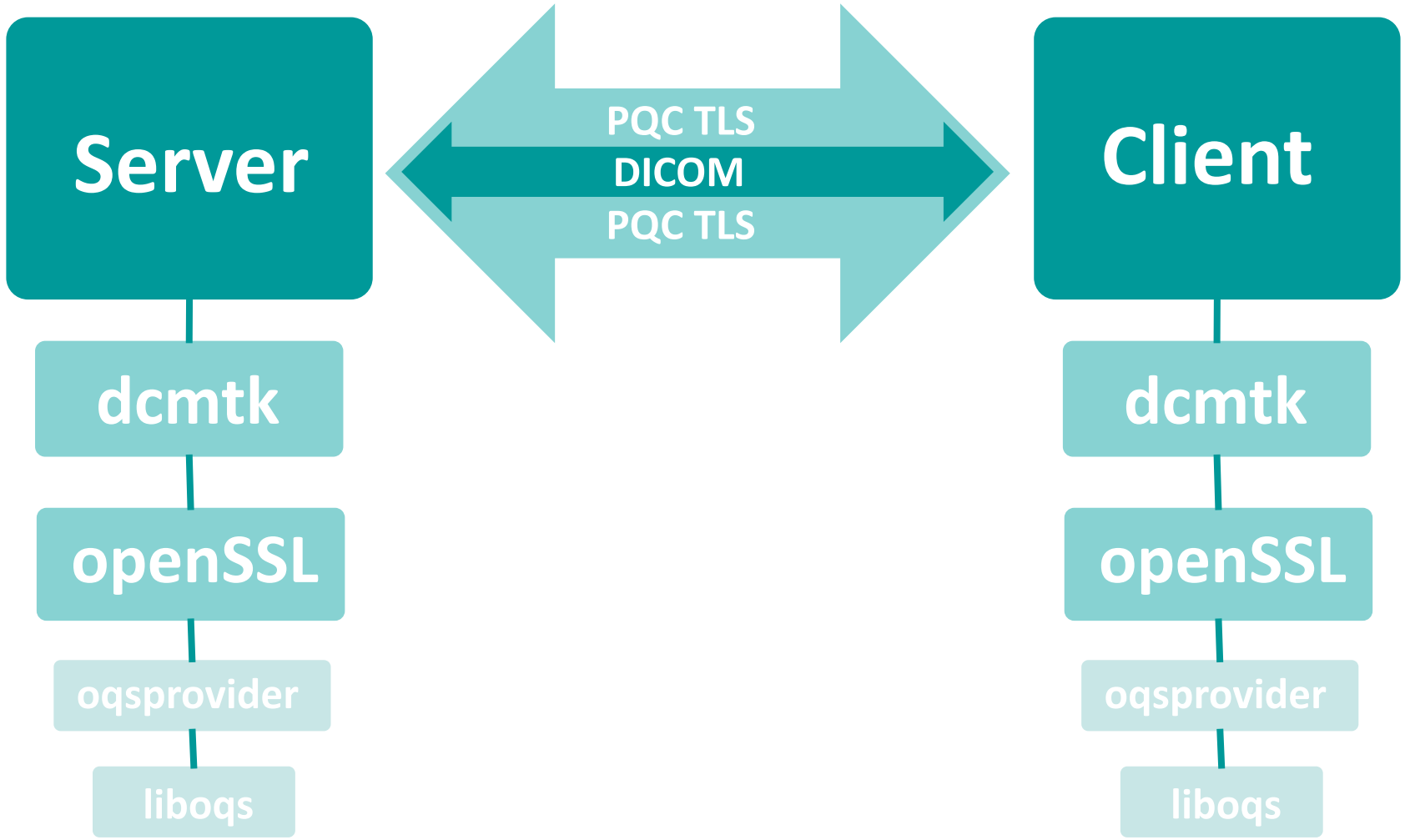
02 **Implementing a PQC prototype in a Healthineers product**

03 Creating a CBOM and performing TRA of quantum threat

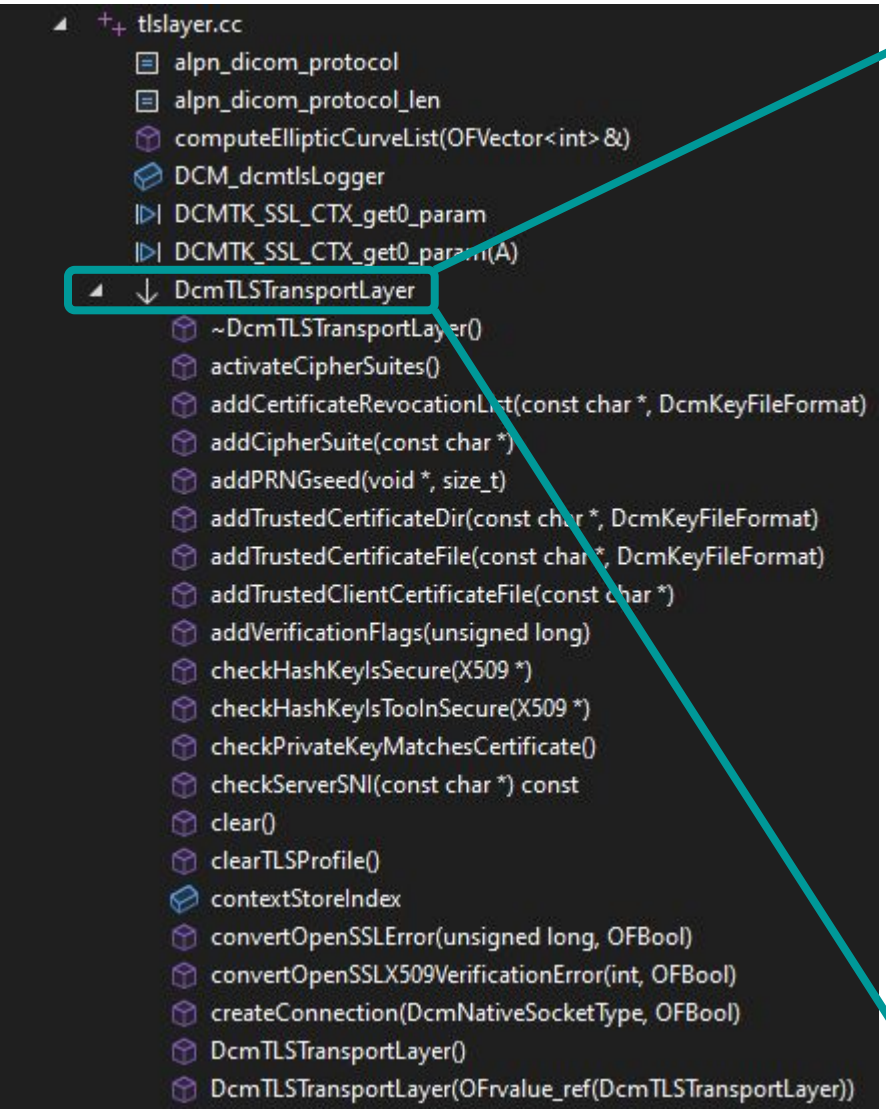
04 Preparing a directive and a strategy document about PQC

05 Implementing PQC in multiple Healtneers products

PQC Secure DICOM



PQC Secure DICOM




```
const char *provider_path = "C:/Program Files/OpenSSL/lib/ossl-modules/oqsprovider";
const char* config_path = "C:/Program Files/Common Files/SSL/openssl.cnf";

OSSL_LIB_CTX* ctx = OSSL_LIB_CTX_new();
OSSL_PROVIDER *provider = OSSL_PROVIDER_load(ctx, provider_path);

OSSL_LIB_CTX_load_config(ctx, config_path);
```

```
switch (networkRole)
{
  case NET_ACCEPTOR:
    transportLayerContext = SSL_CTX_new_ex(ctx, NULL, TLS_server_method());
    break;
  case NET_REQUESTOR:
    transportLayerContext = SSL_CTX_new_ex(ctx, NULL, TLS_client_method());
    break;
  case NET_ACCEPTORREQUESTOR:
    transportLayerContext = SSL_CTX_new_ex(ctx, NULL, TLS_method());
    break;
}
```

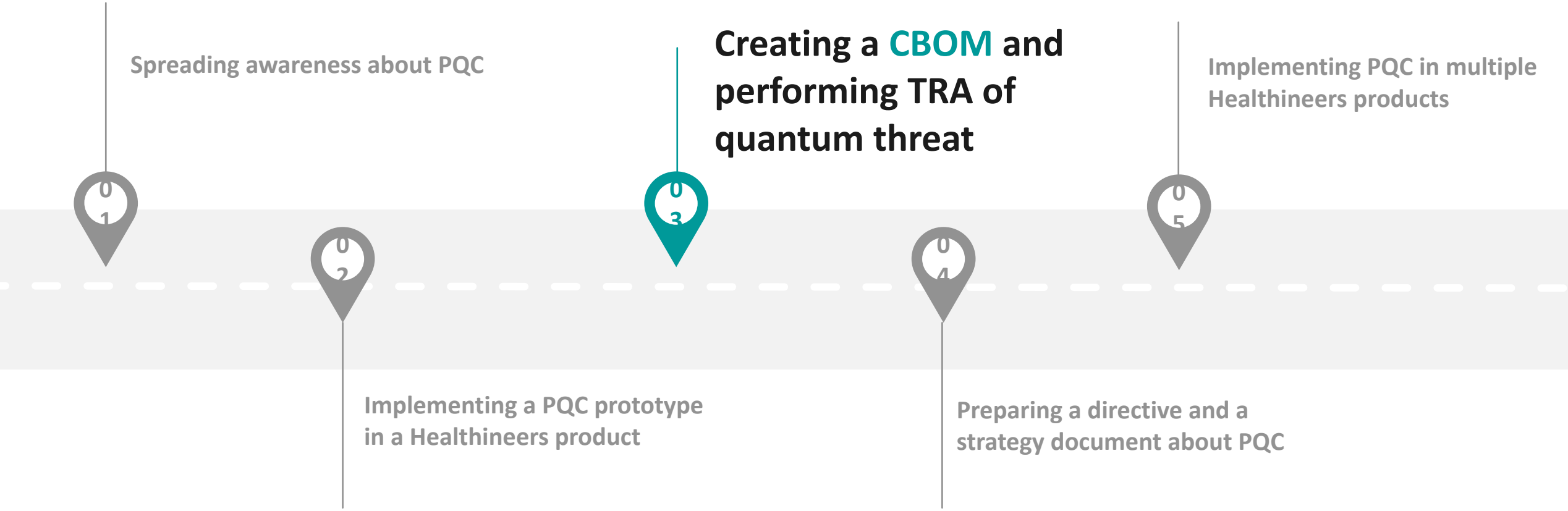
```
const OSSL_ALGORITHM* kemalgs;
kemalgs = OSSL_PROVIDER_query_operation(provider, OSSL_OP_KEM, &query_nocache);
if (kemalgs) {
    for (; kemalgs->algorithm_names != NULL; kemalgs++) {
        fprintf(stderr, "KEM available: %s\n", kemalgs->algorithm_names);
    }
}
```

```
KEM available: kyber512
KEM available: p256_kyber512
KEM available: x25519_kyber512
KEM available: kyber768
KEM available: p384_kyber768
KEM available: x448_kyber768
KEM available: x25519_kyber768
KEM available: p256_kyber768
KEM available: kyber1024
KEM available: p521_kyber1024
KEM available: mlkem512
KEM available: p256_mlkem512
KEM available: x25519_mlkem512
KEM available: mlkem768
KEM available: p384_mlkem768
KEM available: x448_mlkem768
KEM available: x25519_mlkem768
KEM available: p256_mlkem768
KEM available: mlkem1024
KEM available: p521_mlkem1024
KEM available: p384_mlkem1024
KEM available: hqc128
KEM available: p256_hqc128
KEM available: x25519_hqc128
KEM available: hqc192
KEM available: p384_hqc192
KEM available: x448_hqc192
KEM available: hqc256
KEM available: p521_hqc256
```

PQC Secure DICOM

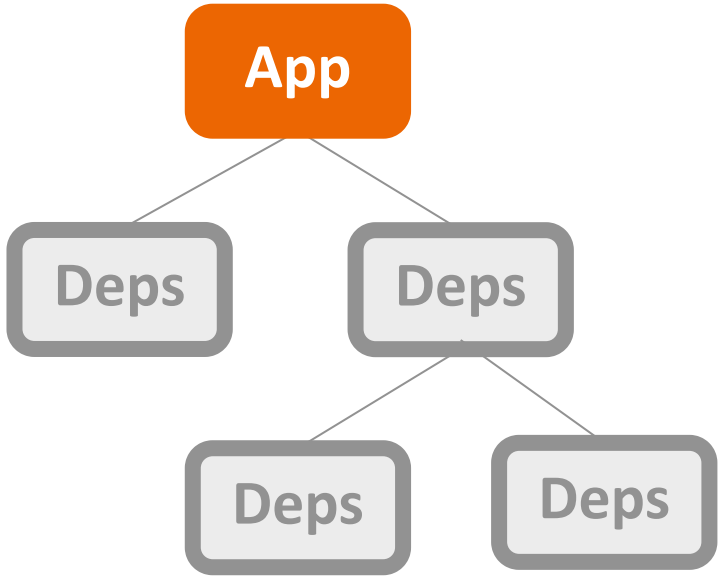
```
# ASC_dumpParameters : VERY VERBOSE :
====================== BEGIN A-ASSOCIATE-AC =====================
Our Implementation Class UID:      1.2.276.0.7230010.3.0.3.6.8
Our Implementation Version Name:   OFFIS_DCMTK_368
Their Implementation Class UID:    1.2.276.0.7230010.3.0.3.6.8
Their Implementation Version Name: OFFIS_DCMTK_368
Application Context Name:    1.2.840.10008.3.1.1.1
Calling Application Name:    VELOCITY1
Called Application Name:     VSCDATABASE
Responding Application Name:
Our Max PDU Receive Size:    32768
Their Max PDU Receive Size:  32768
Presentation Contexts:
  Context ID:        1 (Accepted)
    Abstract Syntax: =VerificationSOPClass
    Proposed SCP/SCU Role: Default
    Accepted SCP/SCU Role: Default
    Accepted Transfer Syntax: =LittleEndianExplicit
Requested Extended Negotiation: none
Accepted Extended Negotiation:  none
Requested User Identity Negotiation: none
User Identity Negotiation Response:  none
====================== END A-ASSOCIATE-AC =====================
DicomTLSServer::dispatch : DIMSE command received
Info: DIMSE_C_ECHO_RQ received
Received Echo SCP RQ: MsgID 1
Info: DIMSE_sendEchoResponse
Info: DIMSE_sendEchoResponse OK
%C-ECHO Association closed.VELOCITY1
```

```
▾ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 1174
    Version: TLS 1.2 (0x0303)
    Random: a16ea836f690accf1d49a5de6aee3d06afc1d273b9b1a6ada37d555c32843cac
    Session ID Length: 32
    Session ID: c907724fcd2cf586c7bb49521ed2f550df387e961bbcd2dd5485cda302bdc875
    Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
    Compression Method: null (0)
    Extensions Length: 1102
  ▾ Extension: supported_versions (len=2) TLS 1.3
      Type: supported_versions (43)
      Length: 2
      Supported Version: TLS 1.3 (0x0304)
  ▾ Extension: key_share (len=1092) kyber768
      Type: key_share (51)
      Length: 1092
    ▾ Key Share extension
      ▾ Key Share Entry: Group: kyber768, Key Exchange length: 1088
          Group: kyber768 (572)
          Key Exchange Length: 1088
          Key Exchange [truncated]: e659fcb4fe93c4926b1c76c412c3f7db9b9c6945340d2
    [JA3S Fullstring: 771,4866,43-51]
    [JA3S: 15af977ce25de452b96affa2addb1036]
```

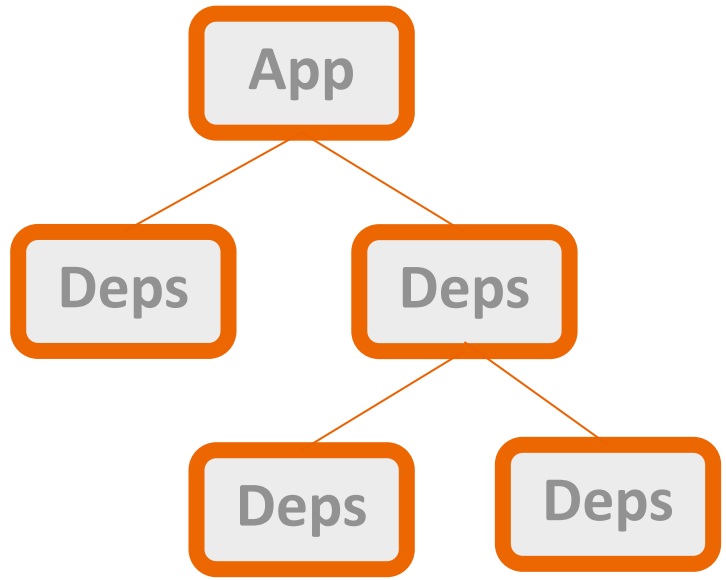

Spreading awareness about PQC
01
Implementing a PQC prototype in a Healthineers product
02
Creating a CBOM and performing TRA of quantum threat
03
Preparing a directive and a strategy document about PQC
04
Implementing PQC in multiple Healthineers products
05

SIEMENS Healthineers

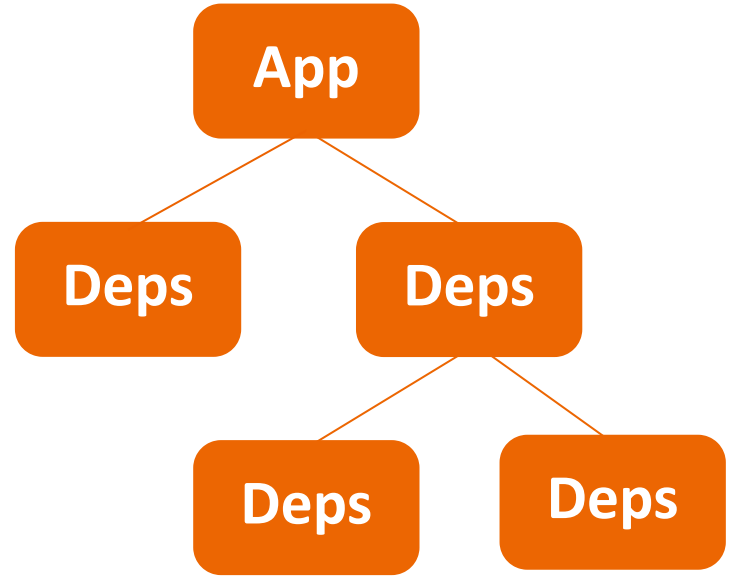
CBOM



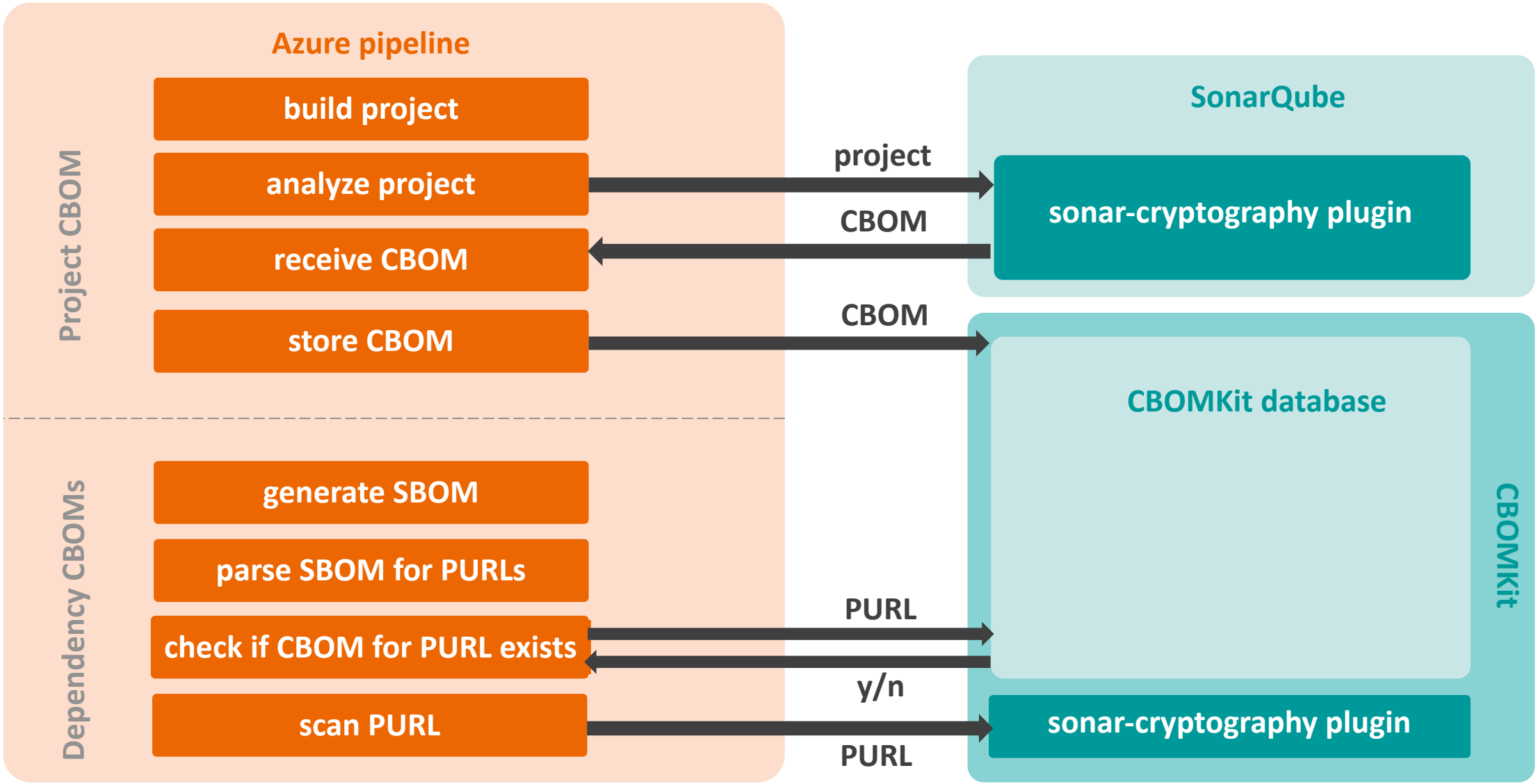
SAST capabilities

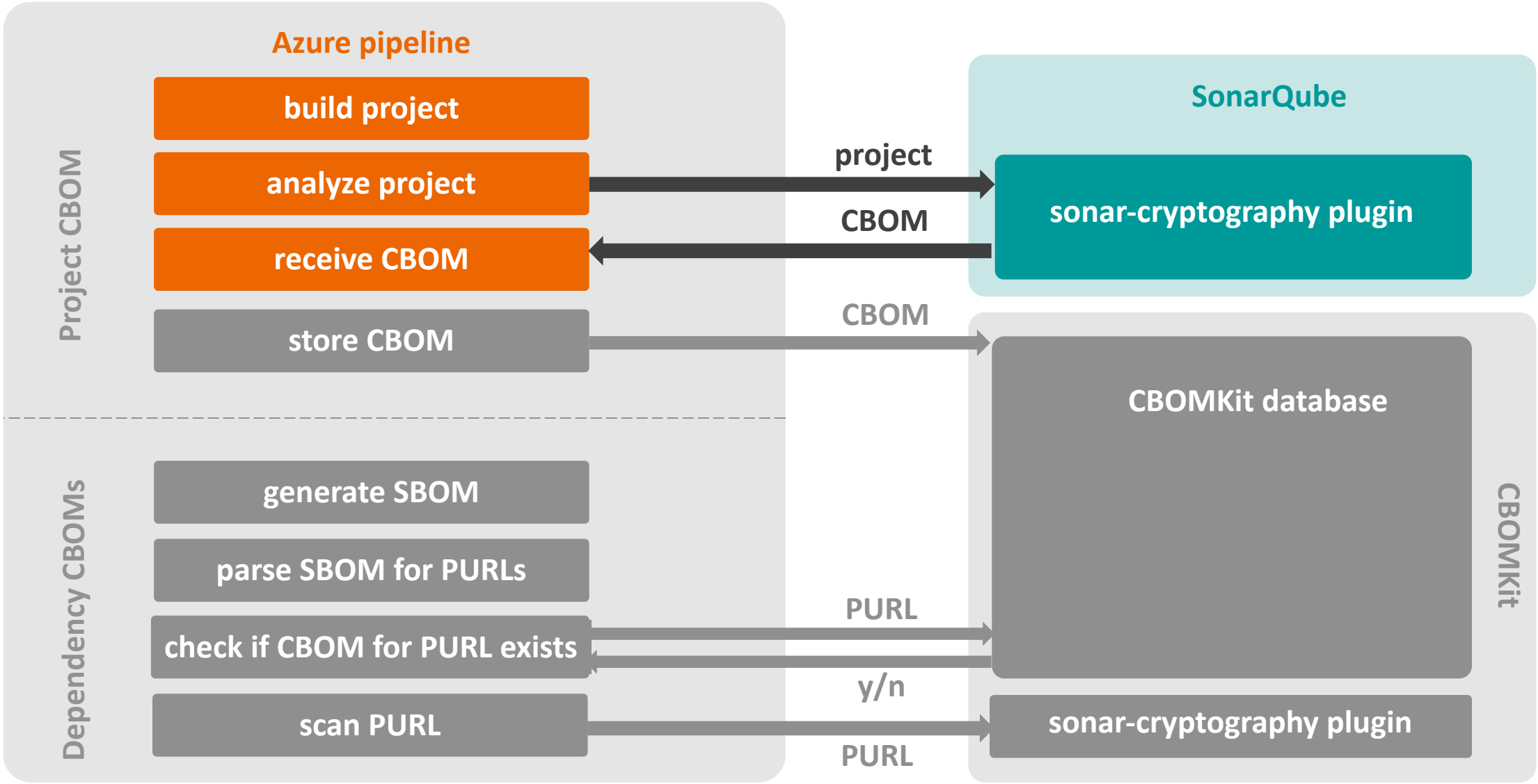


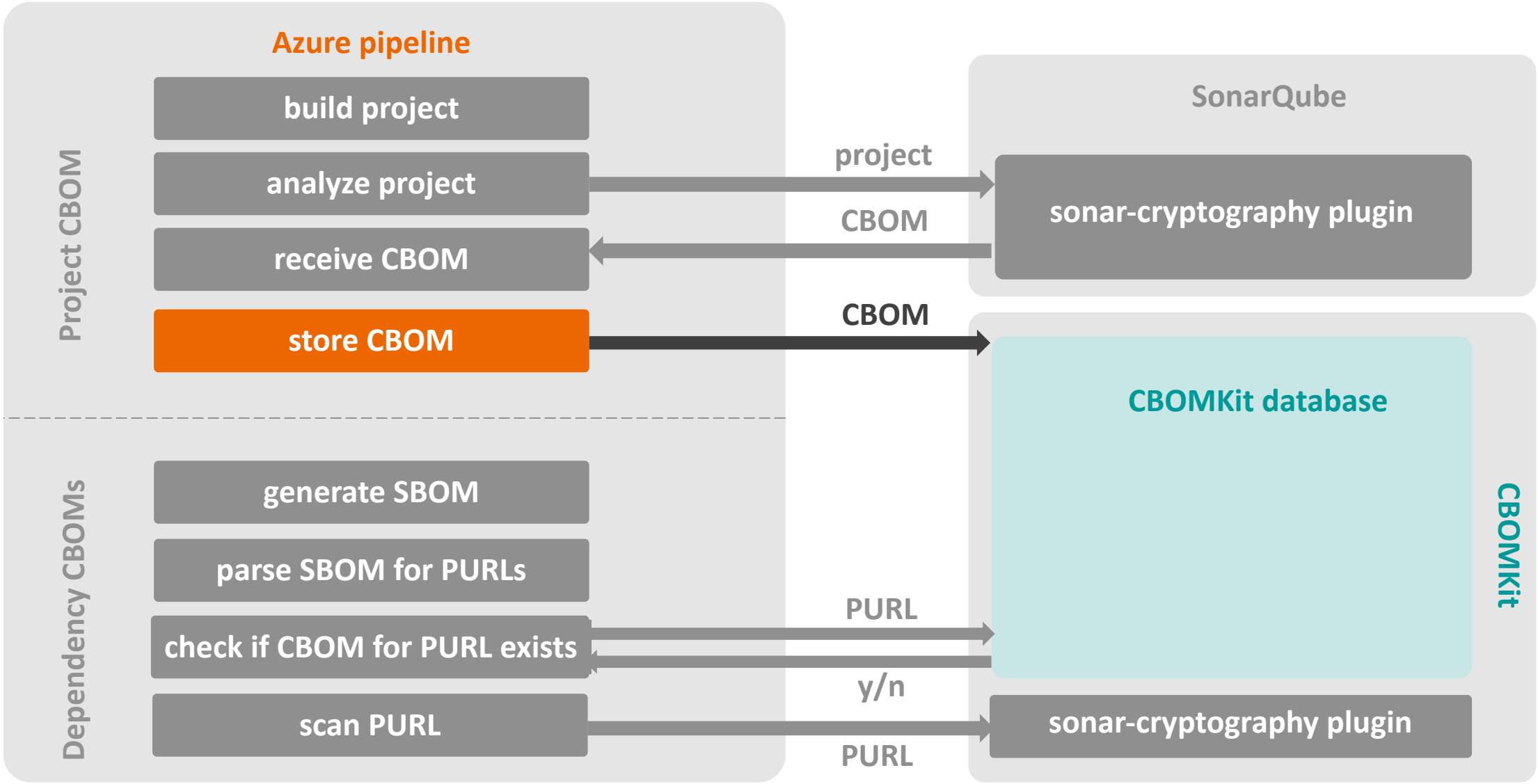
SCA capabilities

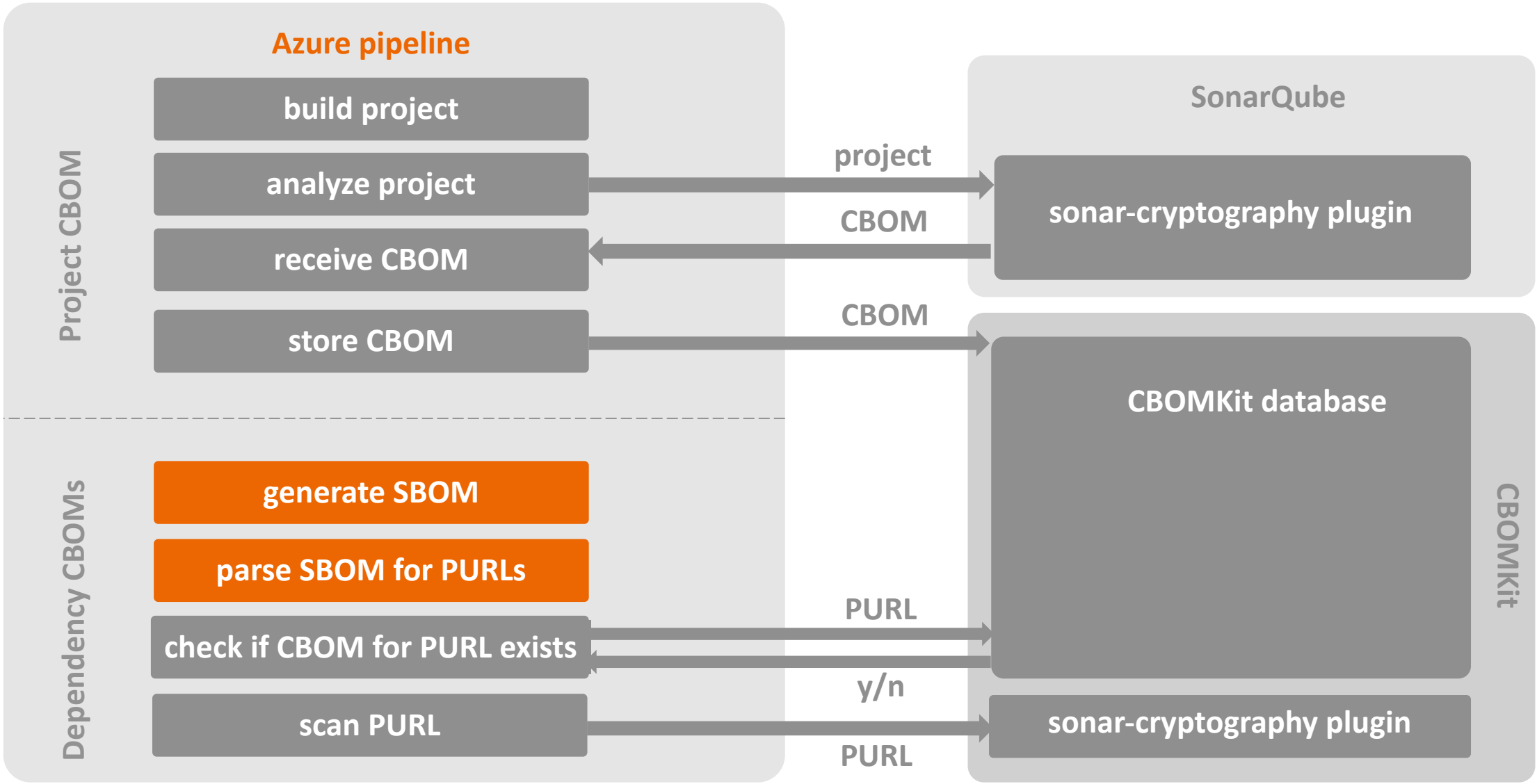


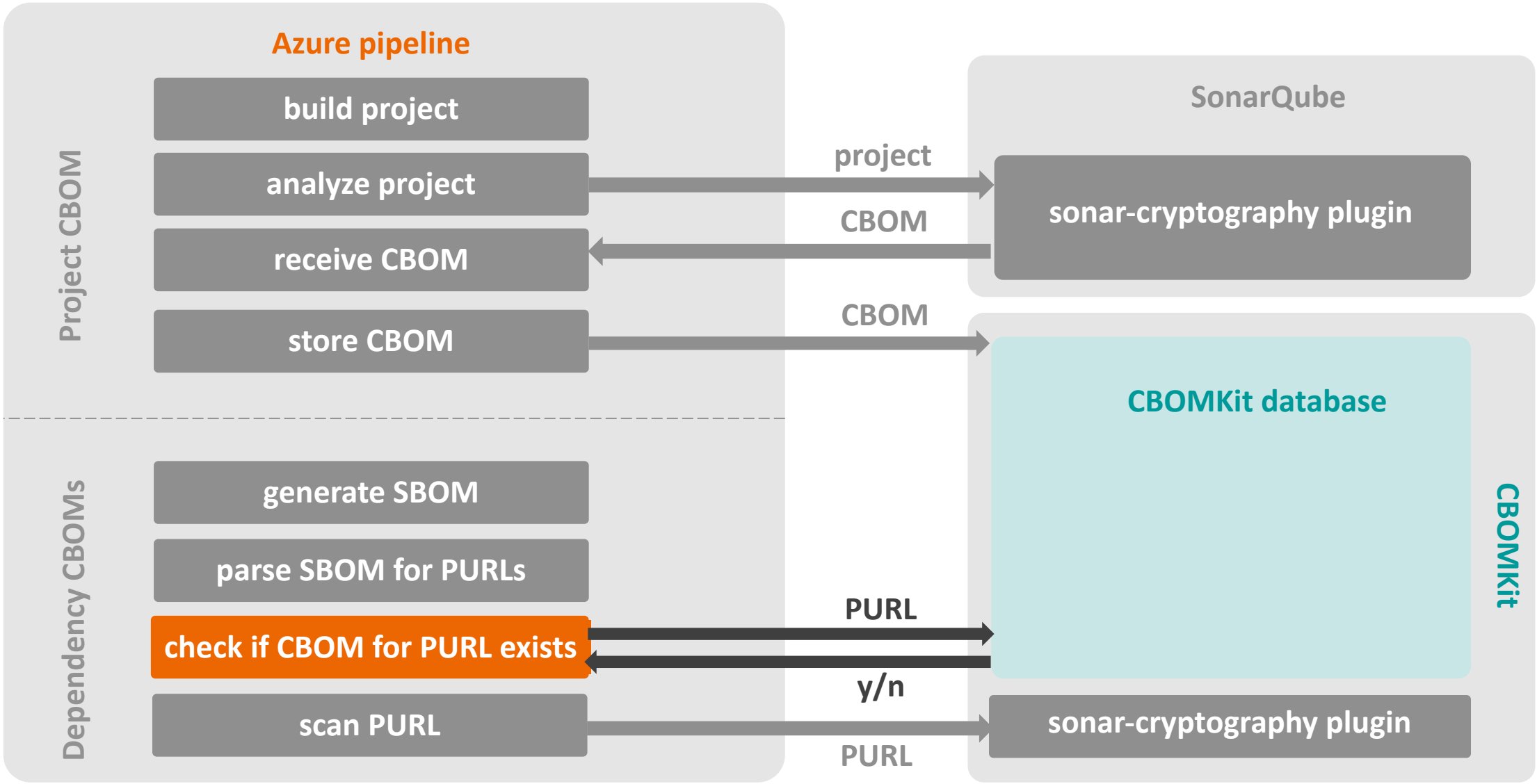
CBOM capabilities

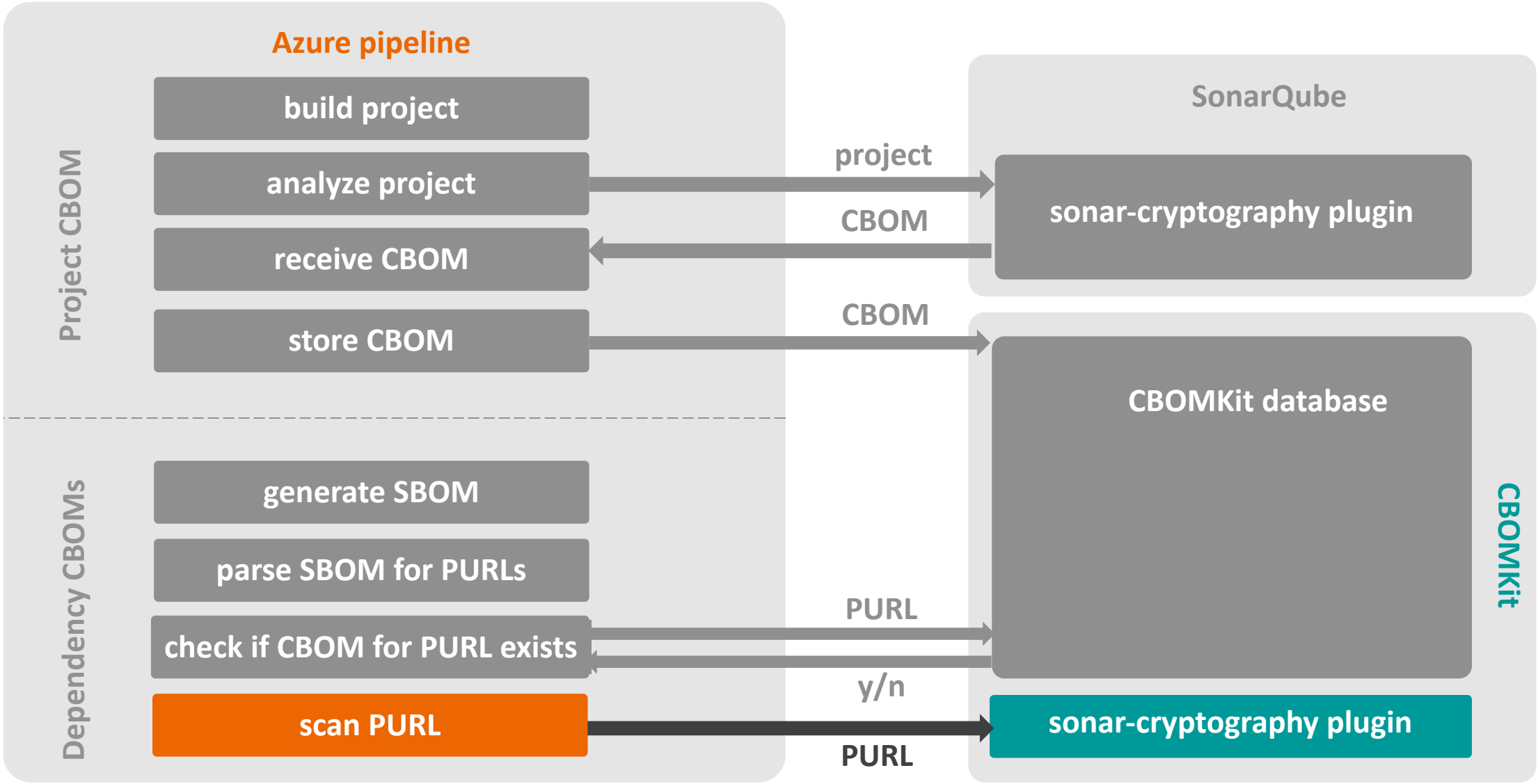

Azure pipeline
Project CBOM
build project
analyze project
receive CBOM
store CBOM
Dependency CBOMs
generate SBOM
parse SBOM for PURLs
check if CBOM for PURL exists
scan PURL
SonarQube
sonar-cryptography plugin
project
CBOM
CBOM
CBOMKit database
CBOMKit
PURL
y/n
sonar-cryptography plugin
PURL

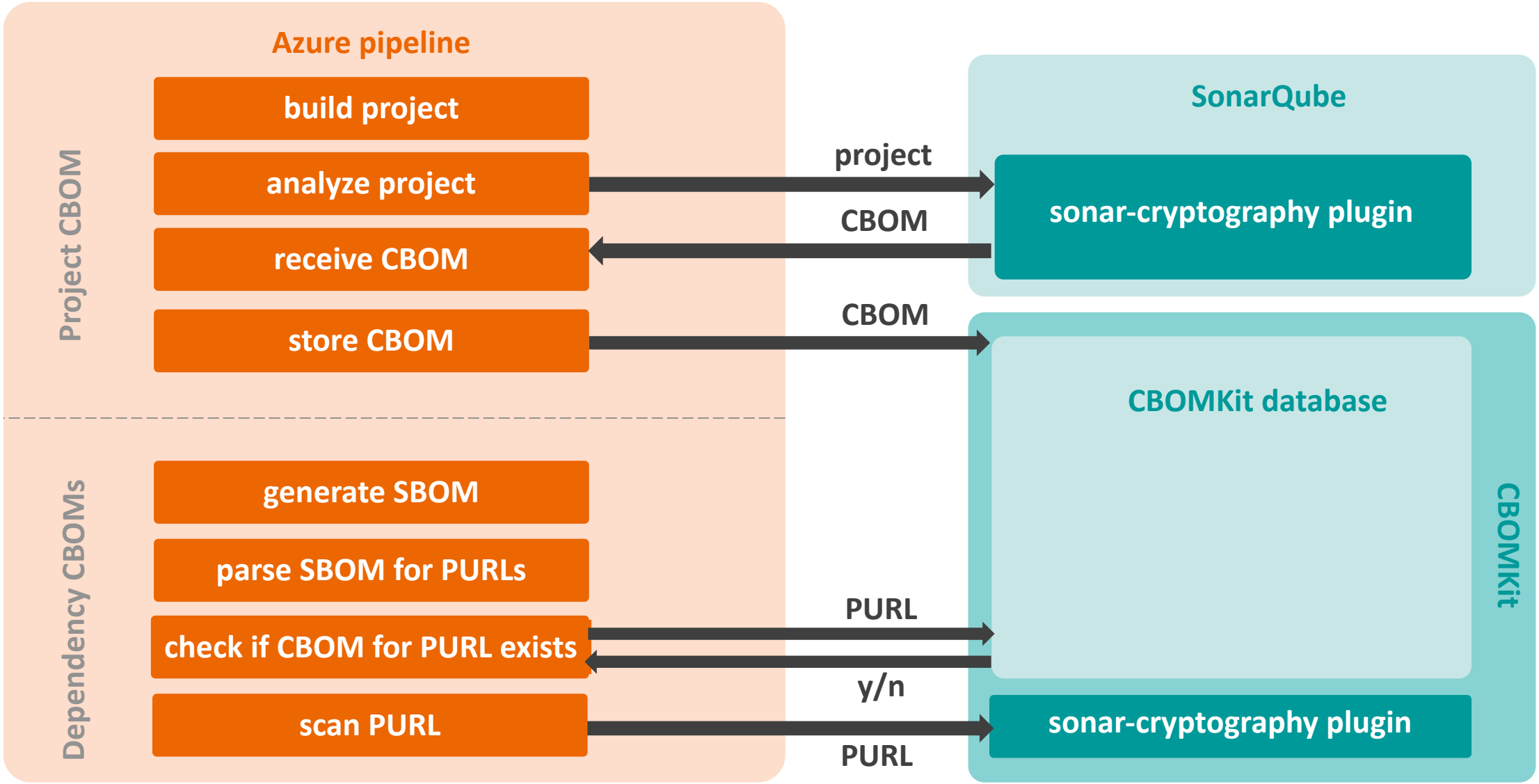

Azure pipeline
build project
analyze project
receive CBOM
store CBOM
Project CBOM
generate SBOM
parse SBOM for PURLs
check if CBOM for PURL exists
scan PURL
Dependency CBOMs
project
CBOM
SonarQube
sonar-cryptography plugin
CBOM
CBOMKit database
PURL
y/n
PURL
sonar-cryptography plugin
CBOMKit


Azure pipeline
Project CBOM
build project
analyze project
receive CBOM
store CBOM
Dependency CBOMs
generate SBOM
parse SBOM for PURLs
check if CBOM for PURL exists
scan PURL
SonarQube
sonar-cryptography plugin
project
CBOM
CBOM
CBOMKit database
CBOMKit
PURL
y/n
PURL
sonar-cryptography plugin


Azure pipeline
Project CBOM
build project
analyze project
receive CBOM
store CBOM
Dependency CBOMs
generate SBOM
parse SBOM for PURLs
check if CBOM for PURL exists
scan PURL
project
CBOM
CBOM
PURL
y/n
PURL
SonarQube
sonar-cryptography plugin
CBOMKit database
sonar-cryptography plugin
CBOMKit


Azure pipeline
Project CBOM
build project
analyze project
receive CBOM
store CBOM
Dependency CBOMs
generate SBOM
parse SBOM for PURLs
check if CBOM for PURL exists
scan PURL
project
CBOM
CBOM
PURL
y/n
PURL
SonarQube
sonar-cryptography plugin
CBOMKit database
sonar-cryptography plugin
CBOMKit


Azure pipeline
Project CBOM
build project
analyze project
receive CBOM
store CBOM
Dependency CBOMs
generate SBOM
parse SBOM for PURLs
check if CBOM for PURL exists
scan PURL
project
CBOM
CBOM
PURL
y/n
PURL
SonarQube
sonar-cryptography plugin
CBOMKit database
sonar-cryptography plugin
CBOMKit


Azure pipeline
Project CBOM
build project
analyze project
receive CBOM
store CBOM
Dependency CBOMs
generate SBOM
parse SBOM for PURLs
check if CBOM for PURL exists
scan PURL
SonarQube
sonar-cryptography plugin
project
CBOM
CBOM
CBOMKit database
CBOMKit
PURL
y/n
PURL
sonar-cryptography plugin

PQC Roadmap for Healthineers

SIEMENS Healthineers

01 Spreading awareness about PQC

02 Implementing a PQC prototype in a Healthineers product

03 Creating a CBOM and performing TRA of quantum threat

04 Preparing a directive and a strategy document about PQC

05 Implementing PQC in multiple Healthineers products

Thank you for your attention