Using OpenSSL and Bouncy Castle for Operational Technology PKI solutions

OpenSSL Conference 2025 in Prague, Czech Republic, October 8th

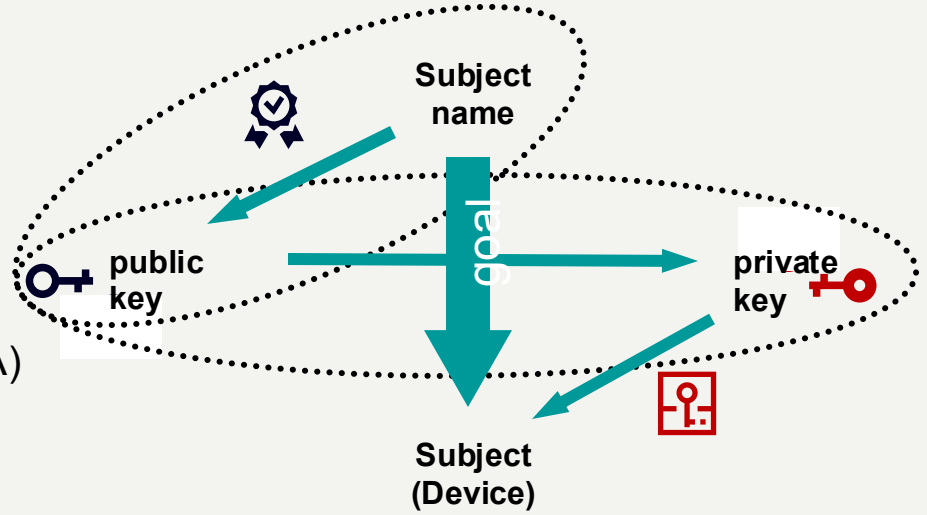
David von Oheimb, Siemens AG, Foundational Technologies, Cybersecurity

SIEMENS

Using Public-Key Infrastructure (PKI) for managing crypto keys

Need:

Secure communication etc. needed for devices/services in the field
Use asymmetric cryptography to simplify key distribution and management
Challenge: secure attribution of public keys to their owning party



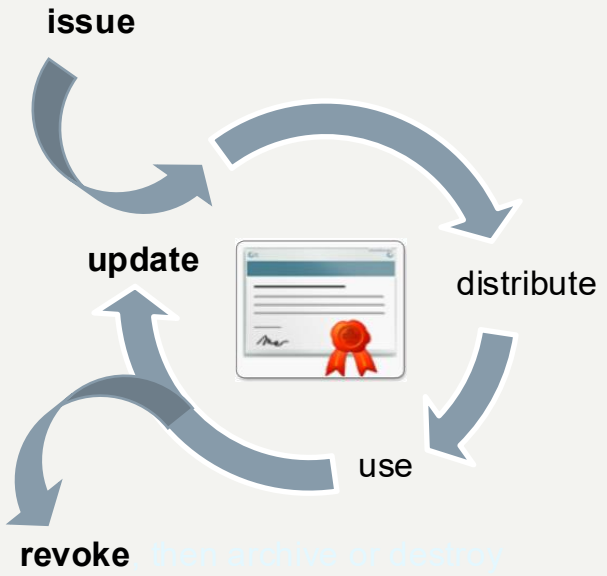
Approach:

Use public-key certificates as “digital passports”, signed by trusted third party (CA)

- Certificate binds public key to the owner’s identity
- Cryptography binds public key to private key
- Secure storage binds private key to owner’s device

Derived challenges:

Securely enroll and manage certificates **in an automated way**
Authorize certificate enrollment at Registration Authority (RA)
Need to trust in Certification Authority (CA) issuing certificates



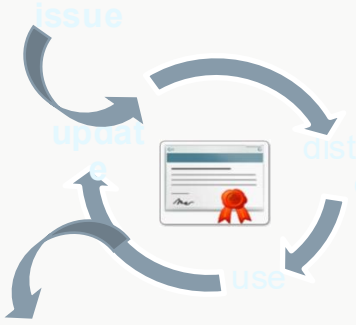
Prerequisites:

Operate PKI components (RA and CA) **securely**, in a trust center
Use a certificate management protocol between PKI components ensuring **proof of origin** and proof of possession for certificate requests

SIEMENS

History: Adding CMP to OpenSSL for use with Siemens Product PKI

- OpenSSL has been **supporting X.509 public-key certificate** generation and use in validation, but **no management protocol** for requesting, renewing, and revoking them.
- Certificate management/enrollment protocols: CMP, CMC, SCEP, EST, ACME



- In 2007, Martin Peylo at NSN/**Nokia developed** [CMPforOpenSSL]
  for use in LTE networks, where CMP is required by the standard
  Tried incorporating the addition with OpenSSL during 2013 .. 2015
  One of the issues: using libcURL due to poor HTTP support in OpenSSL

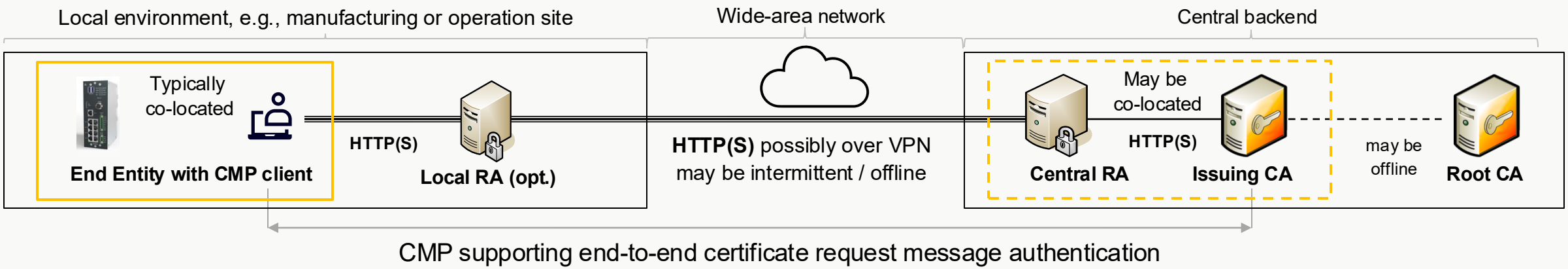
NOKIA
Connecting People

- In 2014/2015, Siemens tried using EST and contributing to Cisco's libEST, did not work out
- In 2015, Siemens Product PKI switched to **CMP** as it is **most flexible and secure**:
  transport-independent, supports end-to-end authentication, crypto agile → post-quantum crypto (PQC)
- I started **contributing** to CMPforOpenSSL, then **to OpenSSL**, based on business unit requirements
- Had to include many OpenSSL improvements; this way **became OpenSSL Committer**
- CMP (over HTTP) **standardized at IETF** in RFC 2510, then RFCs 4210 and 6712, now **RFCs 9810 and 9811**

SIEMENS

SIEMENS

Overview: OT PKI certificate management system architecture



CMP client

- performs CMP requests for End Entity
- implemented using **OpenSSL library**

Registration Authority (RA), typically a local and central one

- checks and forwards CMP requests and responses
- implemented using **Bouncy Castle**

Certification Authority (CA) with possibly separate CMP gateway

- issue certificates and certificate status information
- implemented using **Bouncy Castle**

SIEMENS

Why use OpenSSL for CMP clients at End Entities

Lots of software for Siemens field devices developed in C or C++
Many of them use OpenSSL as crypto/TLS library

Pros:

- ⊕ widely used, most experience available, fits with C-based device software
- ⊕ free OSS, easily available, reliably maintained, FIPS certified, active community
- ⊕ feature-rich: all sorts of crypto, RNG, X.509, OCSP, CMS, (D)TLS, HTTP, QUIC
- ⊕ future-proof by flexible API, e.g., supporting PQC transition
- ⊕ open platform for external, also HW-based crypto implementations (via provider mechanism)

Cons:

- ⊖ Memory-safety hard to achieve
- ⊖ API hard to use
- ⊖ Rather bulky

SIEMENS

How to use OpenSSL for CMP clients

OpenSSL features used by CMP client:

- certificate handling: signatures, ASN.1, X.509 (public-key certs and CRLs), optionally OCSP
- message handling: signatures or MAC, ASN.1, CRMF, CMP, optionally CMS with encryption for key transfer
- message transfer: HTTP, optionally TLS
- file handling etc.: OSSL_STORE and BIOs

CMP implementation inherits via EVP layer:

- post-quantum crypto (ML-DSA etc.)
- OpenSSL crypto provider support

Since OpenSSL 3.0, CMP is part of crypto library and CLI application,
ready to use out-of-the box for all OpenSSL users

Since OpenSSL 3.5, includes new CMPv3 features
defined in RFC 9483 (Lightweight CMP Profile) and RFC 9810 (CMP)

SIEMENS

Generic CMP Client based on OpenSSL library

External GitHub repo “genCMPClient” offers

- **convenient high-level API** focusing on industrial use cases described in RFC 9483 (Lw CMP Profile)
- intermediate CMP library providing all CMP features also with older OpenSSL versions
- demo using Insta Demo CA and EJBCA Community Edition as backends
- experimental implementation of (ML-)KEM and remote attestation, supporting IETF standardization

genCMPClient is **part of various Siemens developments**, in particular:

- **CoreShield Secure OS**, a hardened Linux by Siemens Mobility
- Mainly used in safety-critical (up to SIL 4) railway infrastructure components, such as the **Data Capture Unit (DCU)**.
- fulfill EU Rail System Pillar Cybersecurity requirements
- planned: provide upstream to Civil Infrastructure Platform (CIP) and Debian

SIEMENS

Using Bouncy Castle for RA and CA developments

Most Siemens PKI component developments are in Java or C#

Using **Bouncy Castle** as the crypto and CMP library is natural choice

- good support of crypto (JCE), ASN.1, X.509 (incl. CRLs), and OCSP
- ahead of other libraries w.r.t. PQC support
- low-level support for CMP messages including CRMF and CMS
- good support for TLS (JSSE provider)

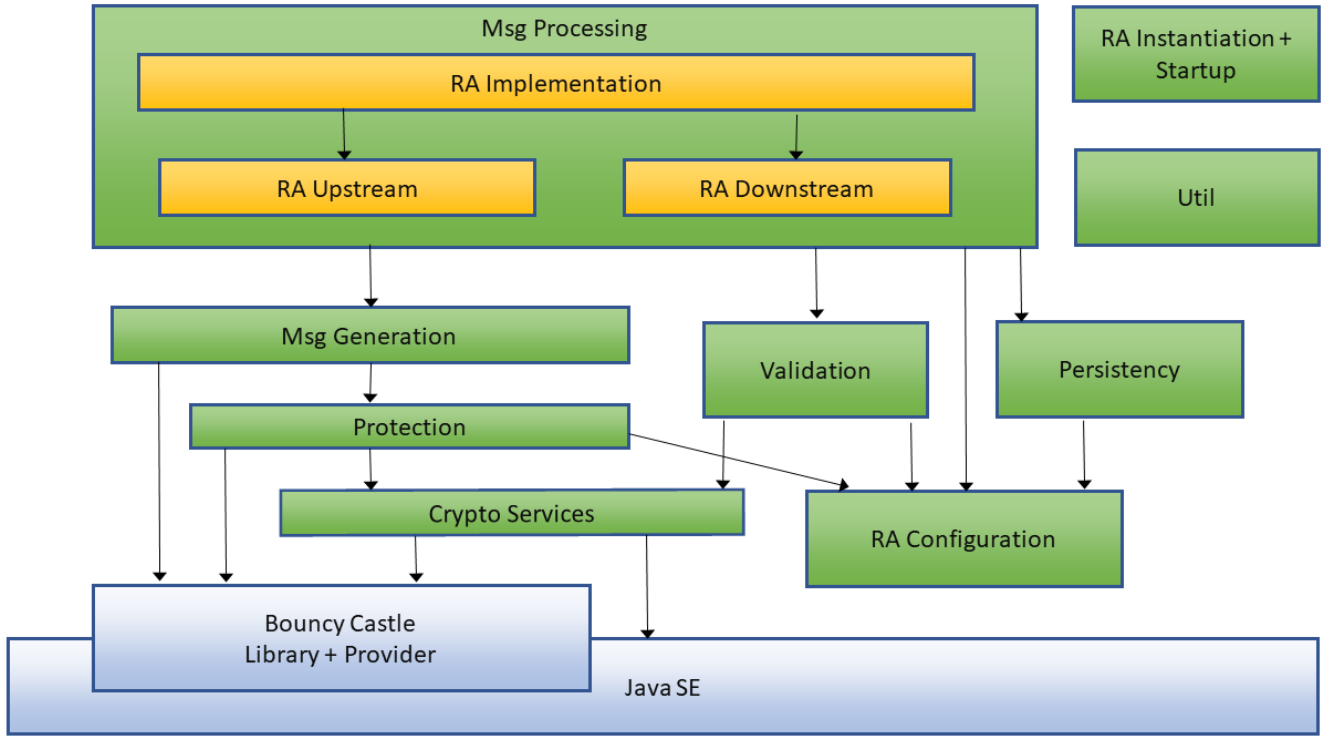
Keyfactor offers EJBCA implemented in Java **based on Bouncy Castle**, providing limited CMP support (part of the features of RFC 9810)
Siemens Product PKI uses EJBCA and thus Bouncy Castle

Siemens has RA developments **based on Bouncy Castle**, in particular:

- CmpRaComponent and demo application in Java

SIEMENS

Bouncy Castle used in CmpRaComponent



CmpRaComponent and its demo application

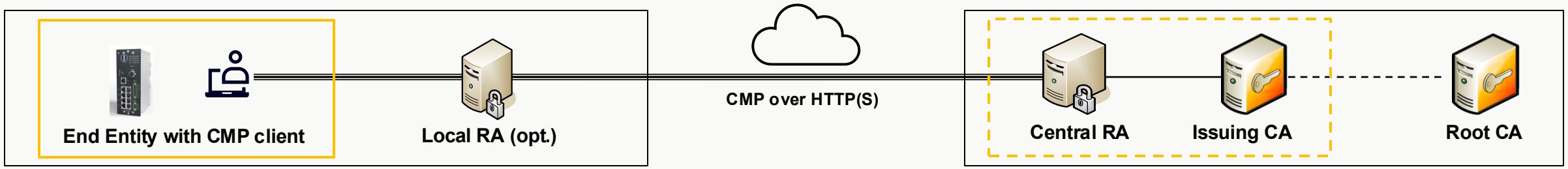
- are available on GitHub repos
- provides CMP Registration Authority (RA) and also CMP client functionality
- **build on Bouncy Castle** for CMP message en-/decoding and cryptography

CmpRaComponent is designed as a **library for Java-based application servers and clients**

- provides CMPv2 and CMPv3 features profiled in RFC 9483
- has very lean API, relieving users from details of CMP
- has flexible and dynamically adaptable configuration
- has general interface for application-specific request message checking and response tracking
- is used in developments of several Siemens business units

SIEMENS

Wrap-up: integrated use of OpenSSL library and Bouncy Castle for CMP-based PKIs



Using OpenSSL components for all parts of the OT PKI:

- **Clients** implemented using **OpenSSL library**, optionally using genCMPClient API
- **RAs** implemented using **Bouncy Castle**, typically using CmpRaComponent
- **CAs** implemented using **Bouncy Castle**, typically EJBCA
- good interoperability for standardized CMP features, and for:
  - PQC support, tested with FIPS 203 .. 205 (ML-DSA, SLH-DSA, ML-KEM)
  - remote attestation PoCs

SIEMENS

Conclusion

SIEMENS

Siemens CMP implementations integrate OpenSSL library and Bouncy Castle

- leverage their specific strengths
- contribute missing pieces and fixes
- provide useful components as OSS, with Apache 2.0 license

openSSL CORPORATION

Take-away:

OSS components for use with OT PKI readily available.
They are based on OpenSSL library and Bouncy Castle.

Questions?

 Unrestricted | © Siemens AG 2025 | Dr. David von Oheimb | OpenSSL conference in Prague | 2025-10-08

Abbreviations

ASN.1	Abstract Syntax Notation One
CA	Certificate Authority
CMS	Cryptographic Message Formats (RFC 5652)
CMP	Certificate Management Protocol (RFC 9810)
CRMF	Certificate Request Message Format (RFC 4211)
CRL	Certificate Revocation List (RFC 5280)
EST	Enrollment over Secure Transport (RFC 7030)
OCSP	Online Certificate Status Protocol
RA	Registration Authority
RFC	Request For Comment
PKI	Public-Key Infrastructure
PQC	Post-Quantum Cryptography
X.509	ITU-T standard for public-key certificates (~ RFC 5280)

SIEMENS