coolPACS

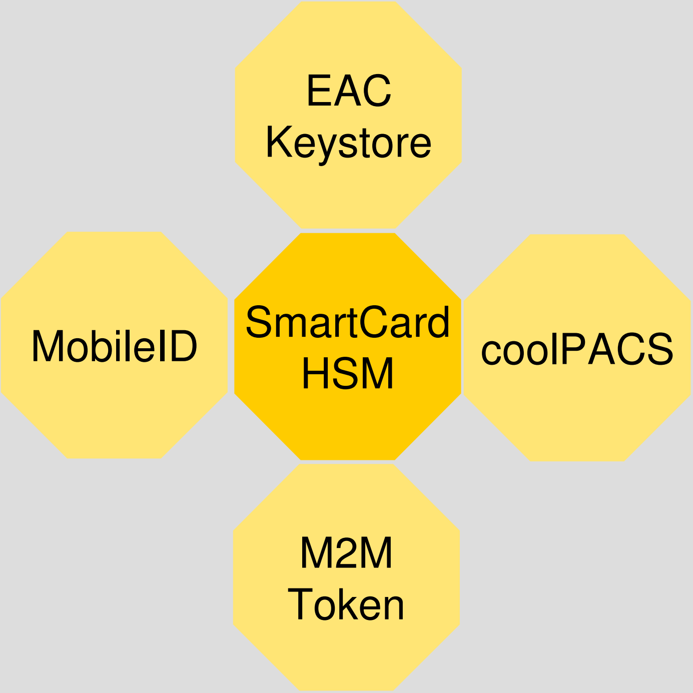
Convergence of Logical and Physical
Access Control

Motivation

ePassports and national eID cards have established strong and open technical standards for contactless identification technology

How can these developments be utilized to create the next generation employee badge that fosters the convergence of physical and logical access control ?

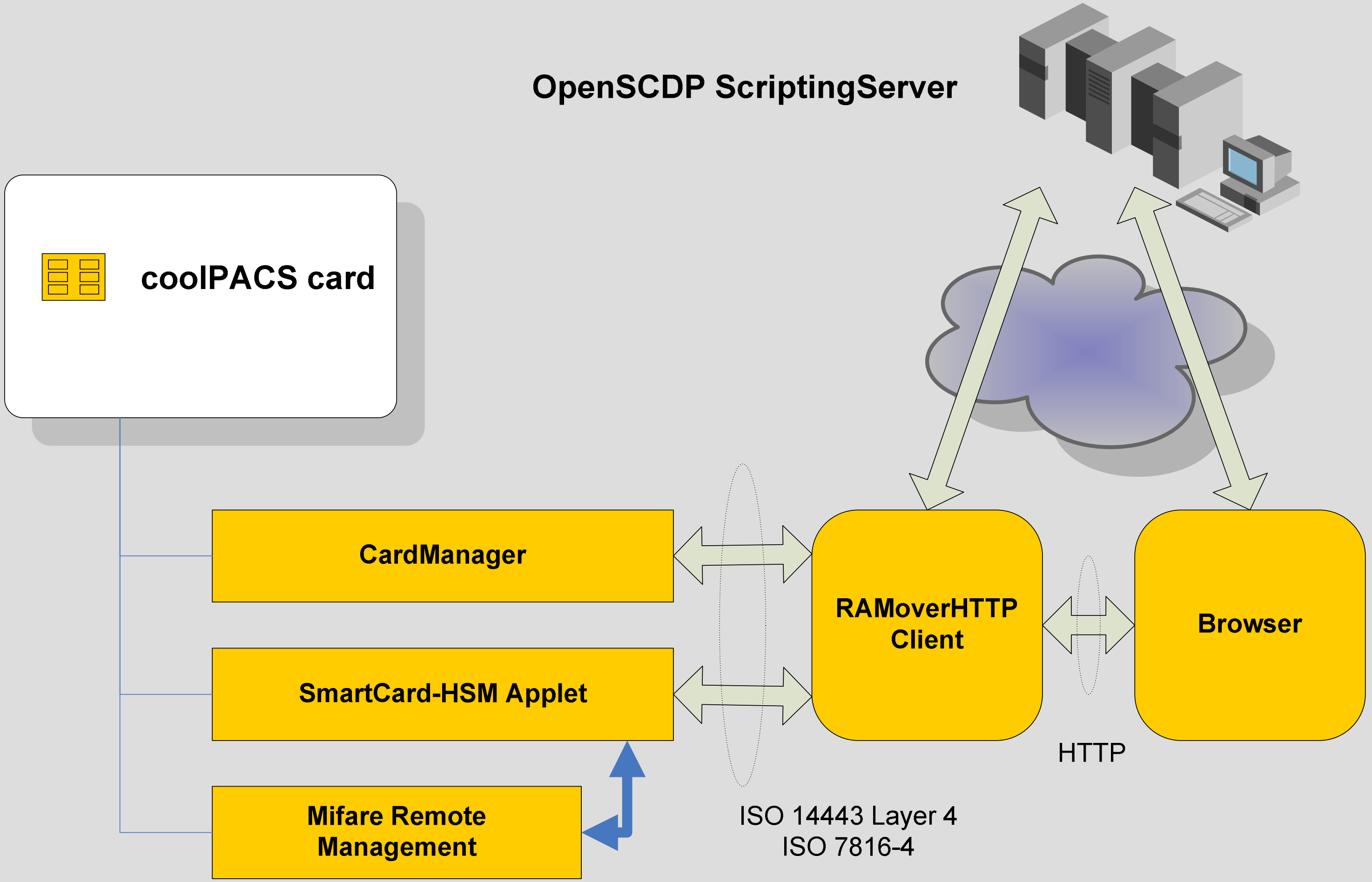
coolPACS context



coolPACS is an application specific development based on the SmartCard-HSM core product

The SmartCard-HSM is a user-centric, remote-manageable secure key store for RSA and ECC keys

coolPACS system elements



coolPACS card

- Form factors
  - Contact chip ID-1 card
  - Contactless or Dual-Interface ID-1 card
  - ID-000 Plug-In
  - USB-Stick
  - MicroSD card
- Technology
  - NXP JCOP 2.4.1r3/2.4.2r2 JavaCard

coolPACS card applets

- SmartCard-HSM Applet
  - Generic PKI key store with PIN protection
  - Asymmetric secure channel for remote management
- Mifare Remote Management Applet
  - Local / remote access to Mifare memory via SmartCard-HSM secure channel

RAMoverHTTP client

- Remote Application Management over HTTP is a protocol defined by Global Platform for the remote management of secure elements in mobile devices
- RAMoverHTTP allows a plain APDU interface between the SE (card/device) and the management server
- The coolPACS RAMoverHTTP client is integrated with the OpenCard Framework and distributed as small jar file
- A C-based RAMoverHTTP client will be integrated with the sc-hsm-pkcs11 module that is part of the SmartCard-HSM crypto middleware

OpenSCDP ScriptingServer

- is a web-application that can perform remote card update actions based on the Global Platform Profiles and Scripting specification
- Can be run as card management backend or with dedicated user interface
- Server side JavaScript for UI components
- Supports RAMoverHTTP protocol, SCP02 and EAC secure channel
- Strong cryptographic functionality with build-in support for PKCS#11 interface

PKI-as-a-Service

- Workflow application running on the OpenSCDP ScriptingServer
- User login with SmartCard-HSM applet via RAMoverHTTP client
- PIN management via web application
- Generate certificate requests to registered certification authorities
- Obtain certificates and store locally on SmartCard-HSM
- Register certification authorities created and operating offline
- Manage certificates (Issue / Revoke)

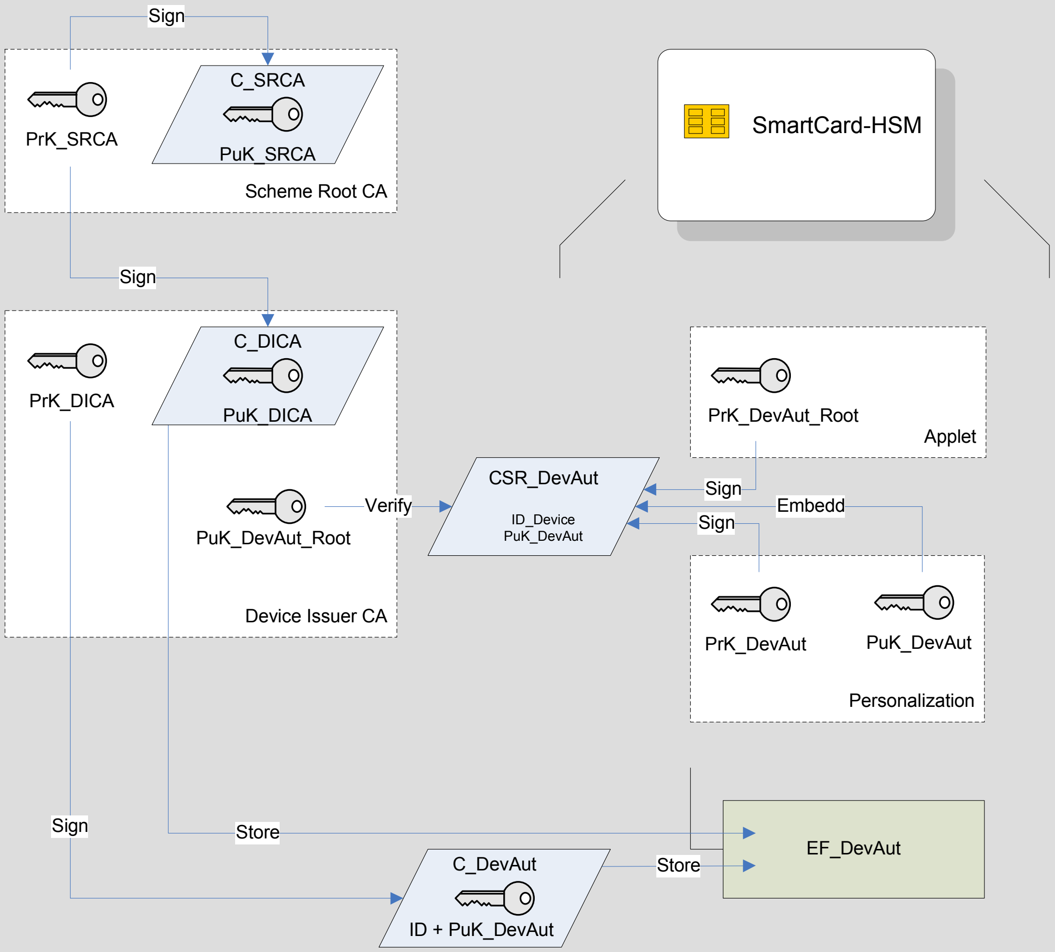
SmartCard-HSM Key Features

- On-board key generation
- Authenticated public key export for remote certification
- Asymmetric device authentication and secure channel
- Encrypted key backup and restore (Key escrow)
- Stateless operations for multi-threaded environments
- Device protection with key or PIN
- PIN management (status, change, unblock)
- Password protected token initialization
- Dynamic key and file allocation and deallocation
- One command tampering

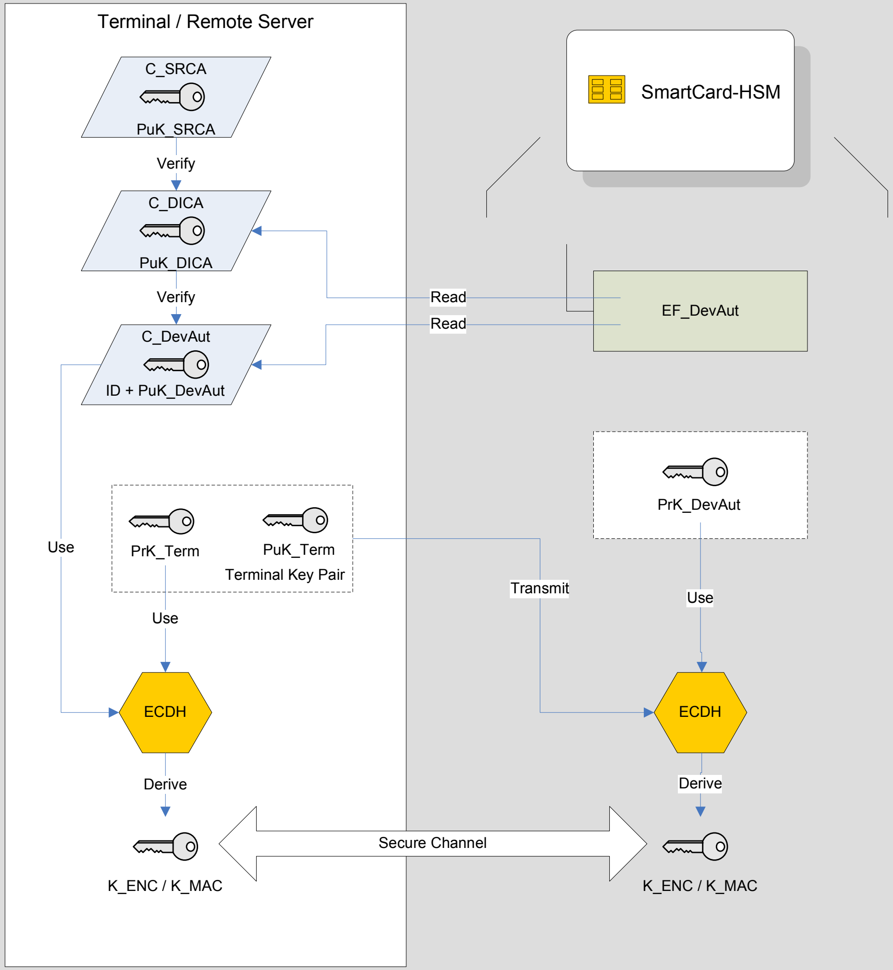
SmartCard-HSM Secure Channel

- Central security function of the SmartCard-HSM design
- Cryptographically strong device identification and authentication
    - Unique Device ID (SRCA/DICA/Device Path)
    - Device Certificate links device identity to authentication key
- Remote authentication and secure session establishment similar to SSL server authentication
- Secure channel encrypts and authenticates all communication between the applet and the server
- No keys required at the server

SmartCard-HSM Authentication PKI



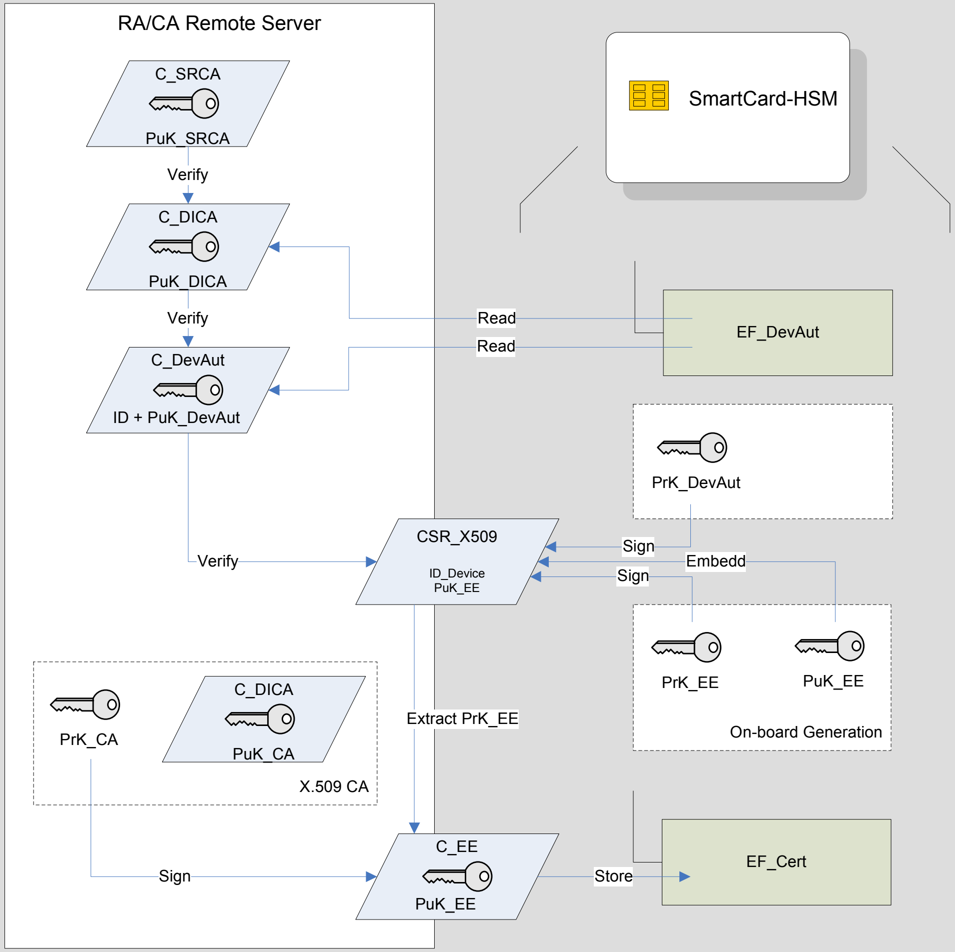
Device Authentication



1. C_SRCA at terminal or remote server
2. Read and verify C_DICA and C_DevAut
3. Perform ECDH at terminal and applet to create common session secrets K_ENC / K_MAC
4. Establish secure channel

- Terminal key is either ephemeral or static
- Identification + Authentication in less than 400 ms

Certificate Issuance Offline



SmartCard-HSM algorithms

- RSA with 1024, 1536 and 2048 bit
- PKCS#1 V1.5 and PSS padding format
- ECDSA with 192, 224, 256 and 320 bit standardized and proprietary domain parameter
- SHA-1, SHA-224 and SHA-256

Middleware and Software Support

- PKCS#11 Support for Windows, Linux and MacOS
- CSP Minidriver for Microsoft Applications
- OpenCard Framework for Java Applications
- Java JCE Crypto Provider for Java Applications
- Online CA and PKI-as-a-Service CA
- Integration and Test Tools (APDU, PKCS#11, JCE)

- All software components are available as open source via the CardContact Developer Network

Thank you very much

For questions please contact

andreas.schwier@cardcontact.de

+49 (0) 571 56149