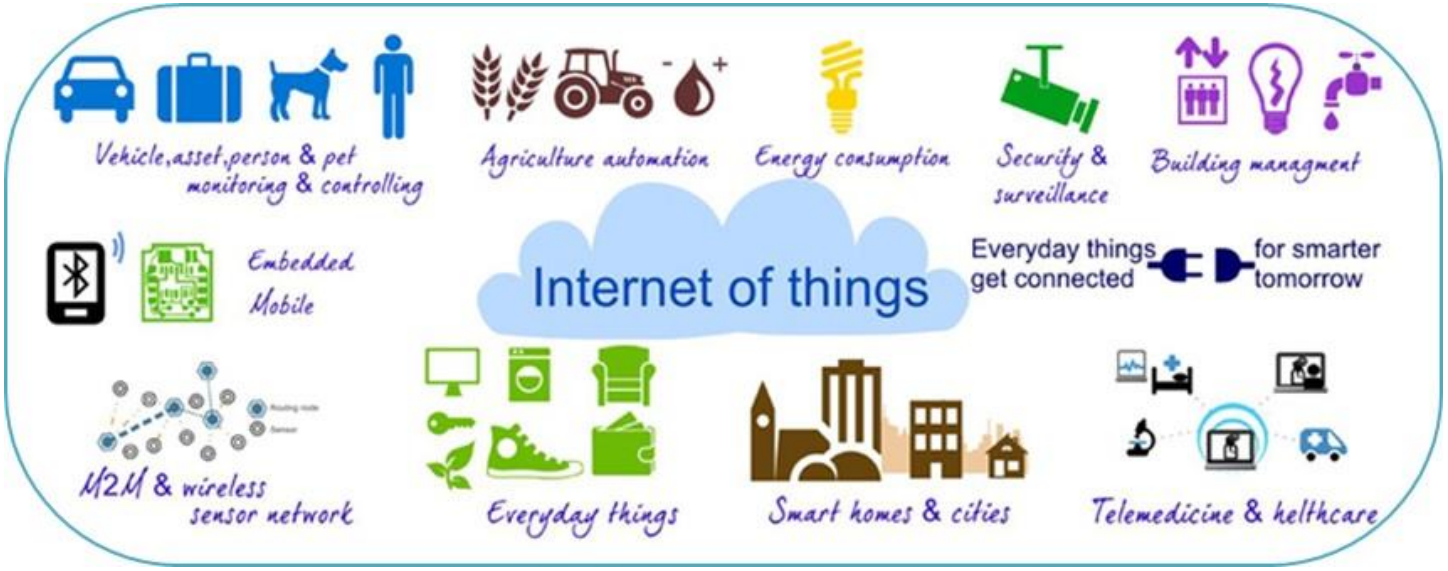
TEE Based Protection of Cryptographic Keys on Embedded IoT Devices

Máté Zombor, **Dorottya Papp**, Levente Buttyán

dpapp@crysys.hu

Introduction

- Embedded devices are increasingly connected to the Internet
  → Internet of Things (IoT)



https://iotworm.com/internet-of-things-applications-area/

- Remote management requires secure remote access
- Devices must store and use long-term cryptographic keys

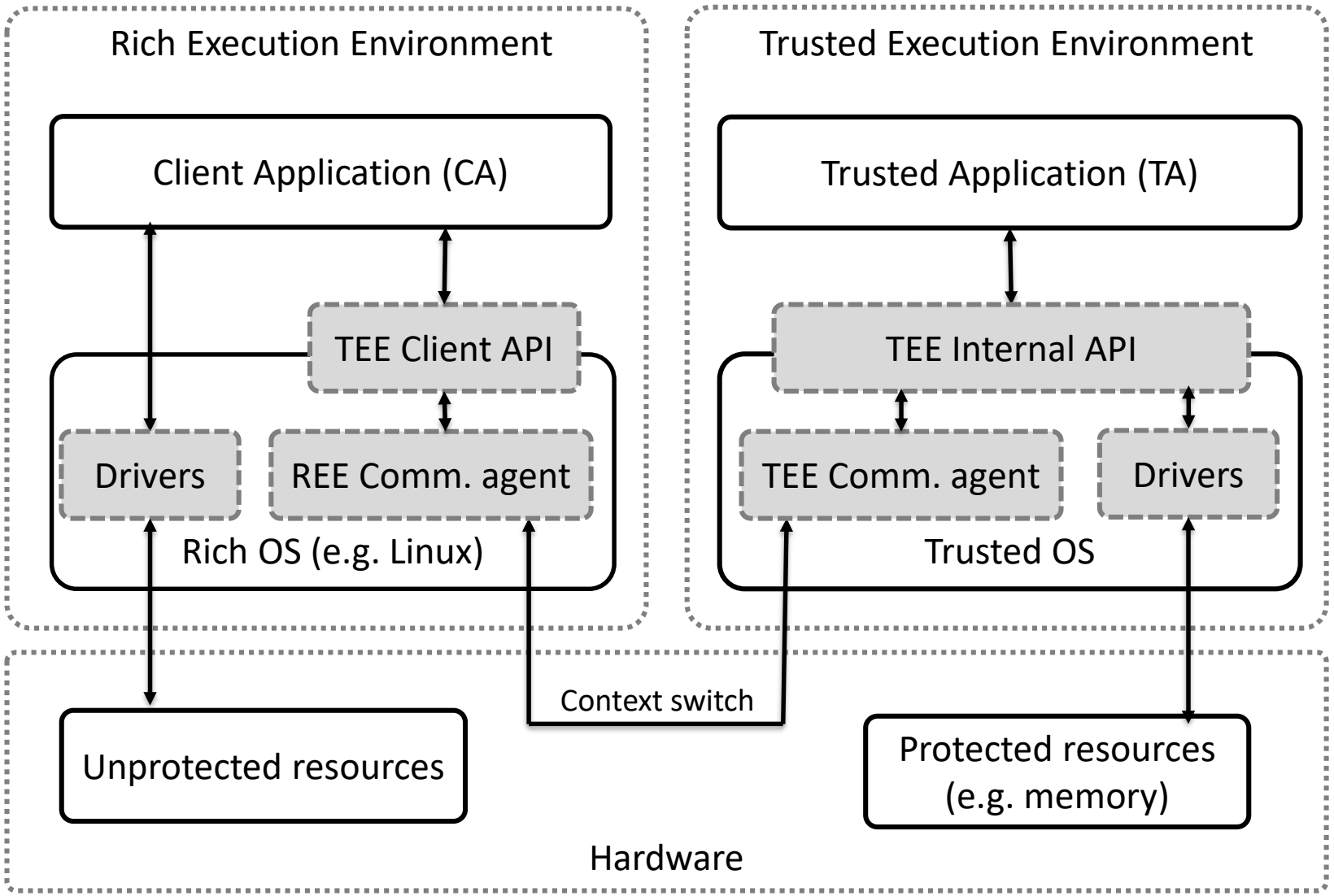
Protecting crypto keys

- Traditionally: additional physical component
  - Trusted Platform Module (TPM) and secure co-processors
  - Hardware security module (HSM)

- Pros: hardened, intrusion-resistant
- Cons: costly →← IoT devices must be cost effective

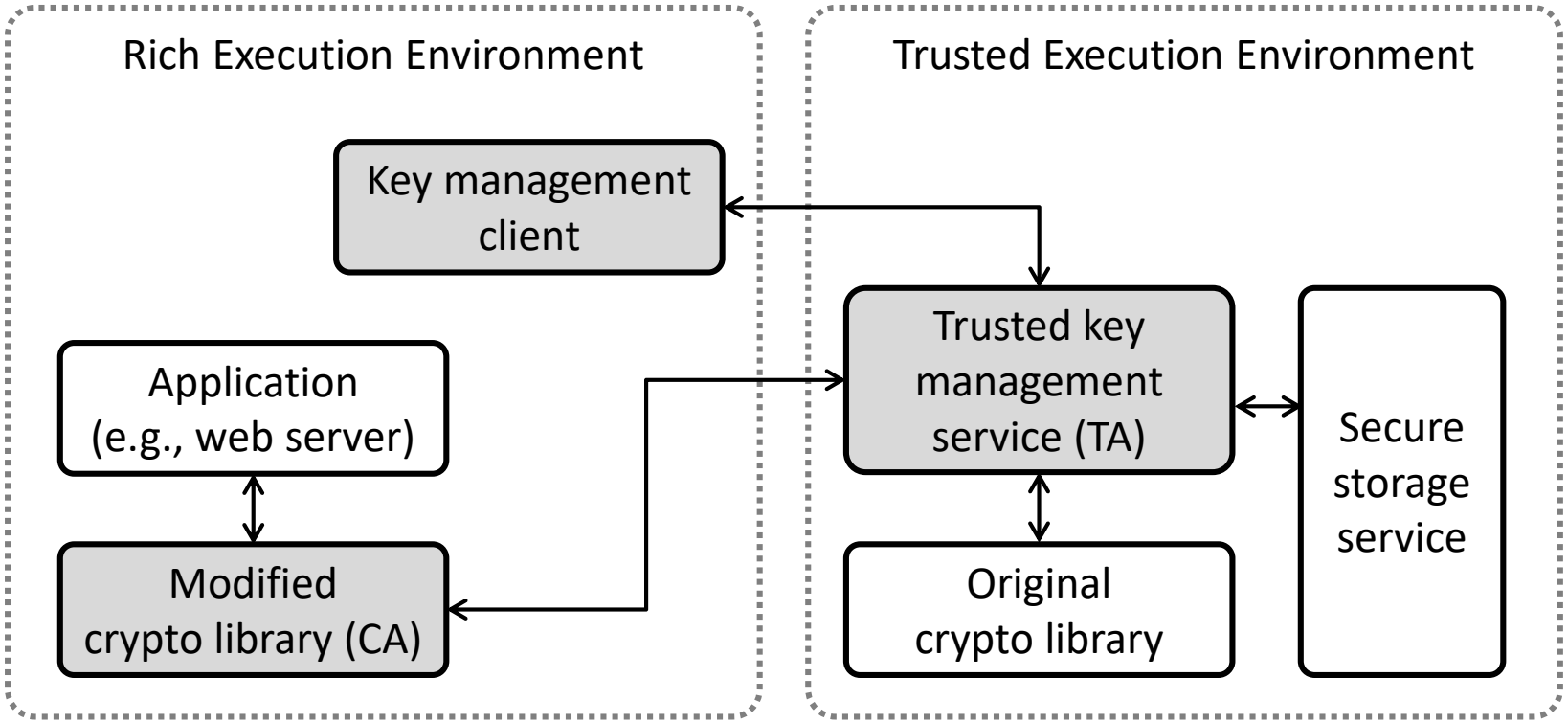
- Emerging technology: Trusted Execution Environments (TEEs)
  - Software with minimal hardware support
  - Available on many platforms, e.g. ARM TrustZone, Intel SGX, AMD Secure Technology

GlobalPlatform's TEE specification



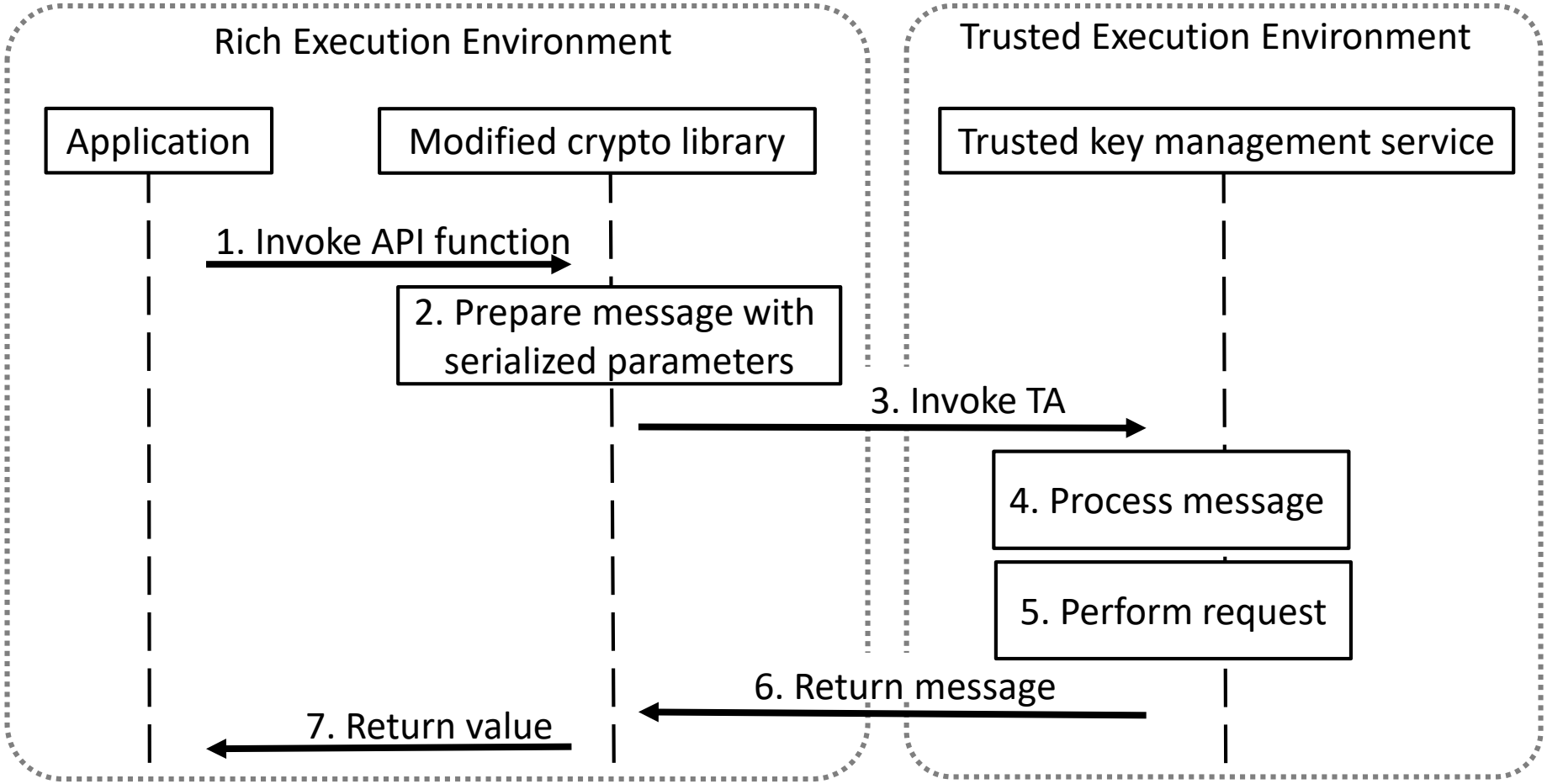
Architectural Overview

- Long-term crypto keys are stored and used in the TEE
- Crypto libraries and key management apps can request actions



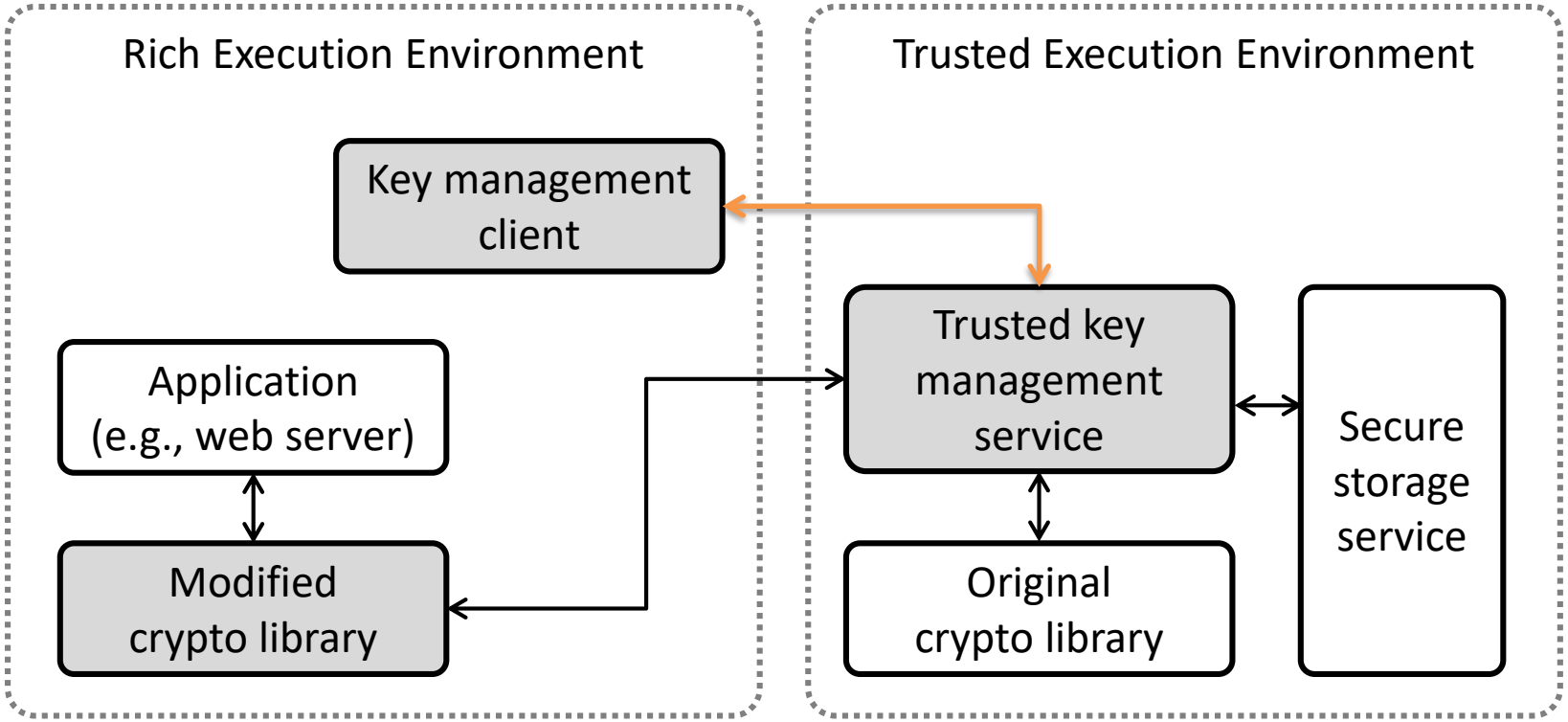
Communication with the TEE

- Programmer invokes API functions
- Under the hood: params are serialized and passed to TA (TEE)



Key management

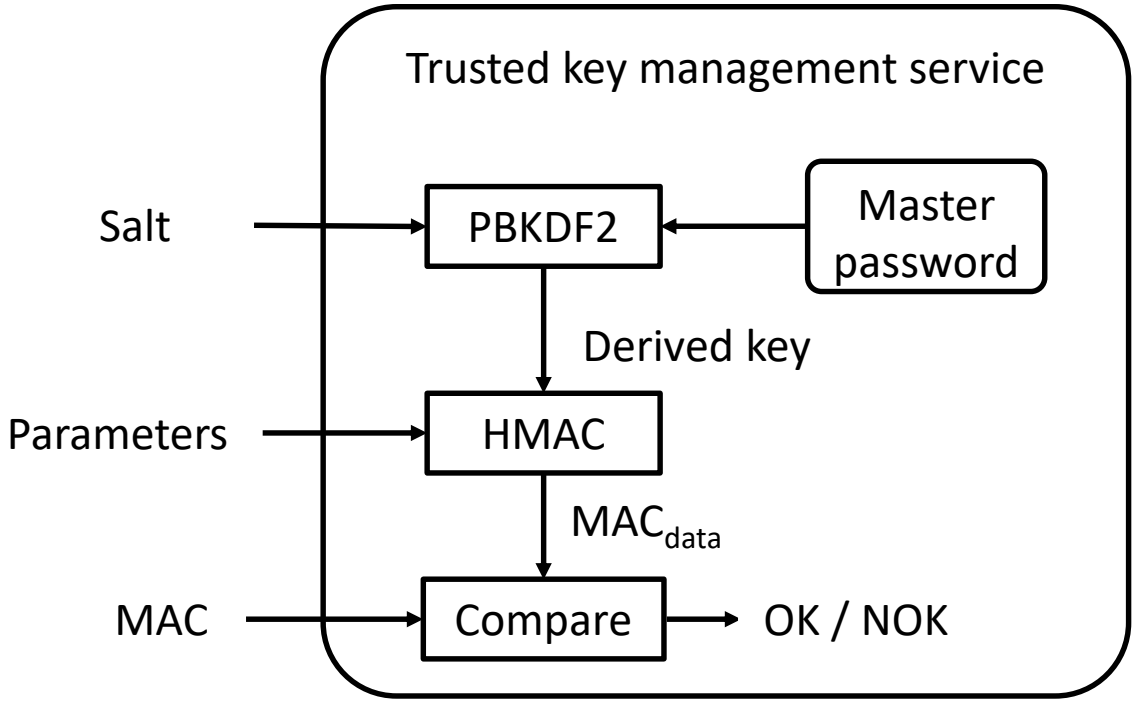
- Generate new keypair
- Load existing keypair
- Remove keypair



Authenticating requests

- Key management functions can only be invoked by the operator
  → Requests must be authenticated

- Requests contain: parameters of invoked operation, salt, MAC



Implementation: OP-TEE + mbedtls

- **`tee_set_master_password`** – install master password

- Key management functions:
  - **`tee_load_keypair`** – load a keypair into TEE
  - **`tee_generate_keypair`** – generate a new keypair in TEE
  - **`tee_remove_keypair`** – remove existing keypair from TEE

- Public functions for all applications
  - **`tee_pk_decrypt`** – decrypt data using a key handle
  - **`tee_pk_sign`** – sign data usign a key handle
  - **`tee_get_publickey`** – extract a public key
  - **`tee_get_keyinfo`** – get information about a key (size, type)

Summary and future work

- Remote access is important for embedded IoT devices
  → Long-term crypto keys are required

- New method for protecting crypto keys with TEE
  - Emerging technology supported by many chips
  - Keys are stored in the Secure Storage → Access to keys is restricted
  - Apps can only reference keys using handles → Keys are not leaked
- Crypto operations (even sign and decrypt) can be called by anyone (even attacker)
- BUT: Private keys cannot be extracted from the device
  → Rogue device can't be cloned

- Future work: remote attestation using crypto keys in TEE

Acknowledgements

The presented work was carried out within the SETIT Project, which has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the 2018-1.2.1-NKP funding scheme (project no. 2018-1.2.1-NKP-2018-00004).