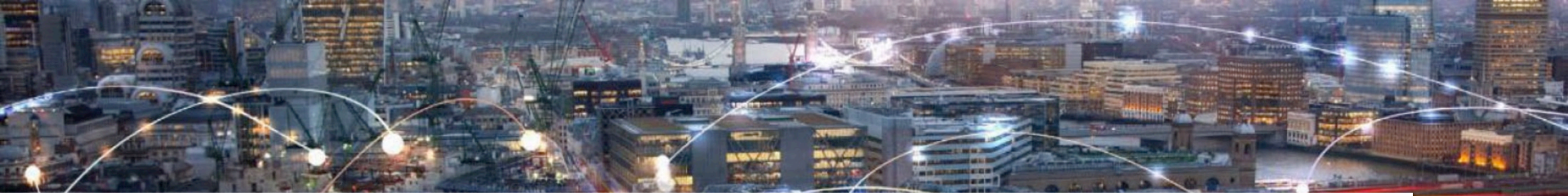
Hardening your embedded system with Secure Elements

Dr. Pierre Girard

Sindelfingen, December 6, 2019

THALES

gemalto
a Thales company



We enable trust in two interlocking ways…

…by developing secure, innovative software.

Our solutions are at the heart of modern life

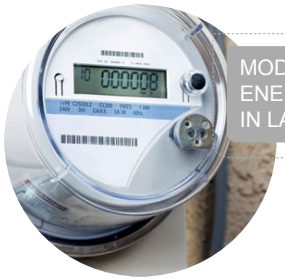
SECURING ECOMMERCE IN MEXICO

POWERING SAMSUNG'S CONNECTED WATCH

LICENSING AND PROTECTING SOFTWARE IN CHINA

SECURING DATA FOR AMAZON WEB SERVICES

ENHANCING ePASSPORT SECURITY FOR TURKEY



MODERNIZING ENERGY SUPPLY IN LATIN AMERICA

We also secure cars (CES 2015)

We also secure cars

- Car access granted for the next 48 h
- Young driver: speed limited to 90 km/h
- Insurance limitation: geo-fencing within EU

Which security solution shall be used?

AMD PSP

Google Titan

Secure Element

TPM

Trust Zone

Integrated Secure Element

Intel SGX

Virtualization

(e)UICC

SHE

HSM

Apple Secure Enclave

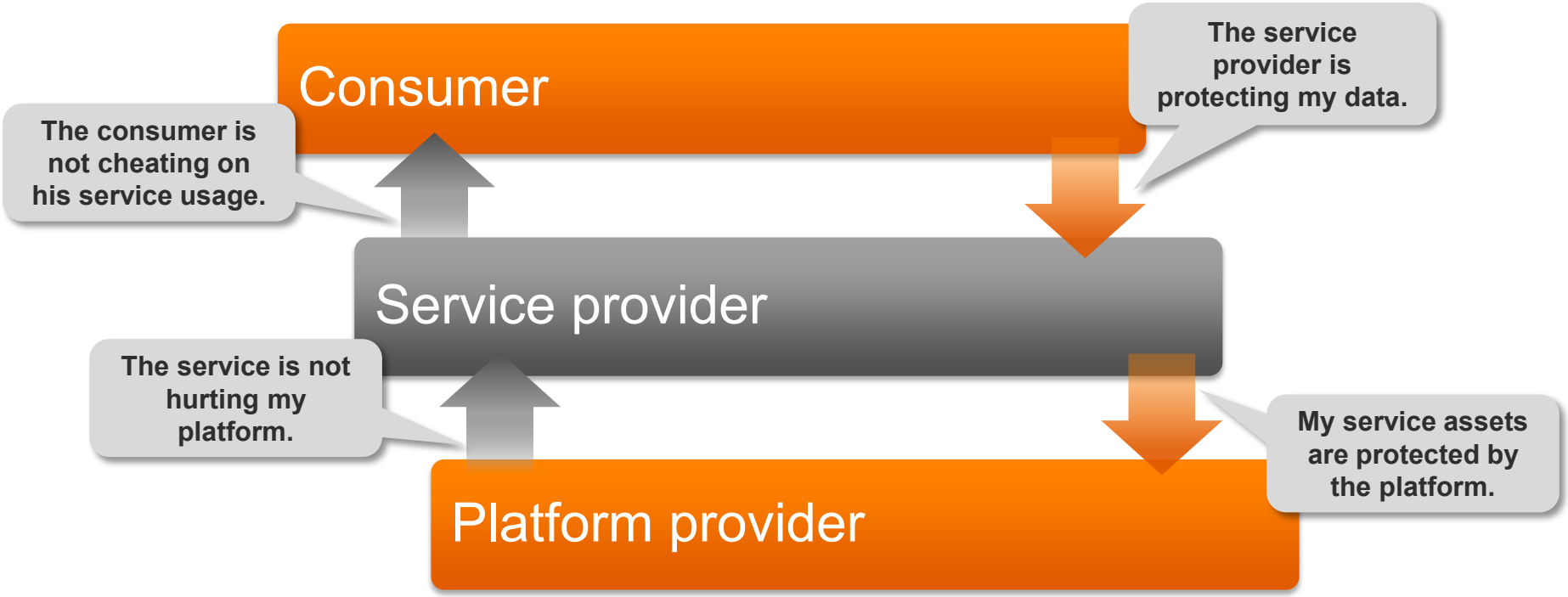
Agenda

- Motivating example
- Security needs and classical solutions for IT world
- Sorting-out the available solutions for embedded world
- Introducing Multiple Levels of Security

Why **trust?**

- Management of sensitive **devices**
  - Car engine, batteries, doors, …
- Management of sensitive **transactions**
  - Car sharing, car renting, mobility as a service
  - Energy: (not) consuming, storing …
  - Peer-to-peer transactions
- Management of sensitive **data**
  - Location / presence, behavior / driving patterns, voice streaming, …

Trust **relationships**

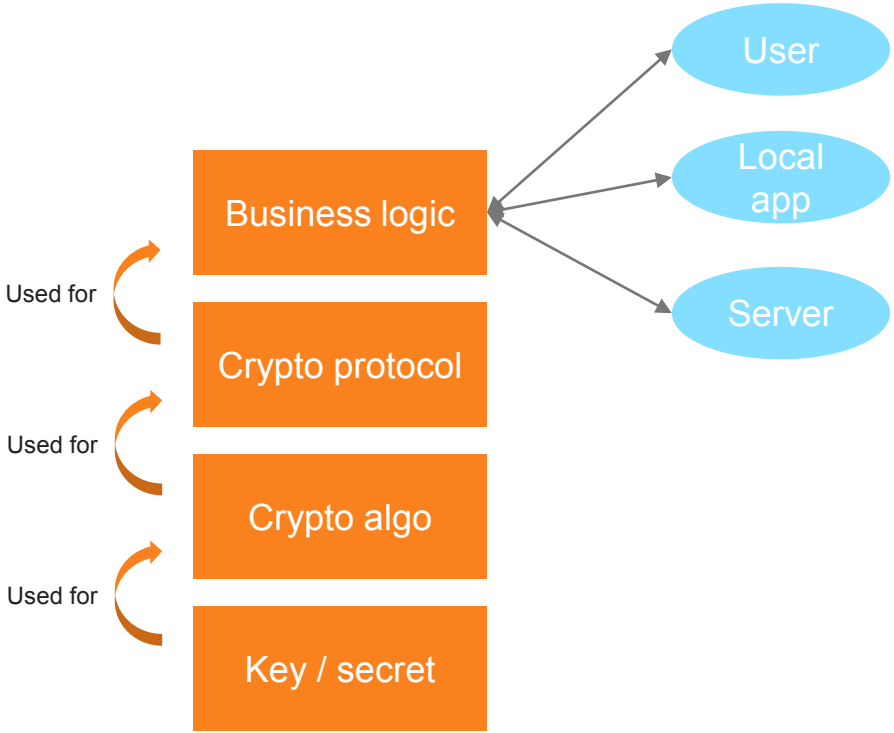


gemalto

How to **enable trust?**

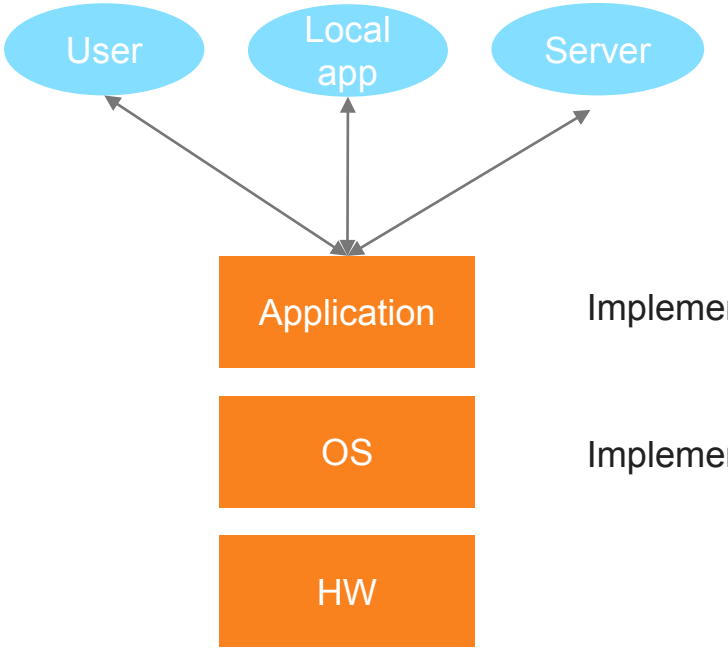
gemalto

What do we need as (security) developers ?



Opening car example

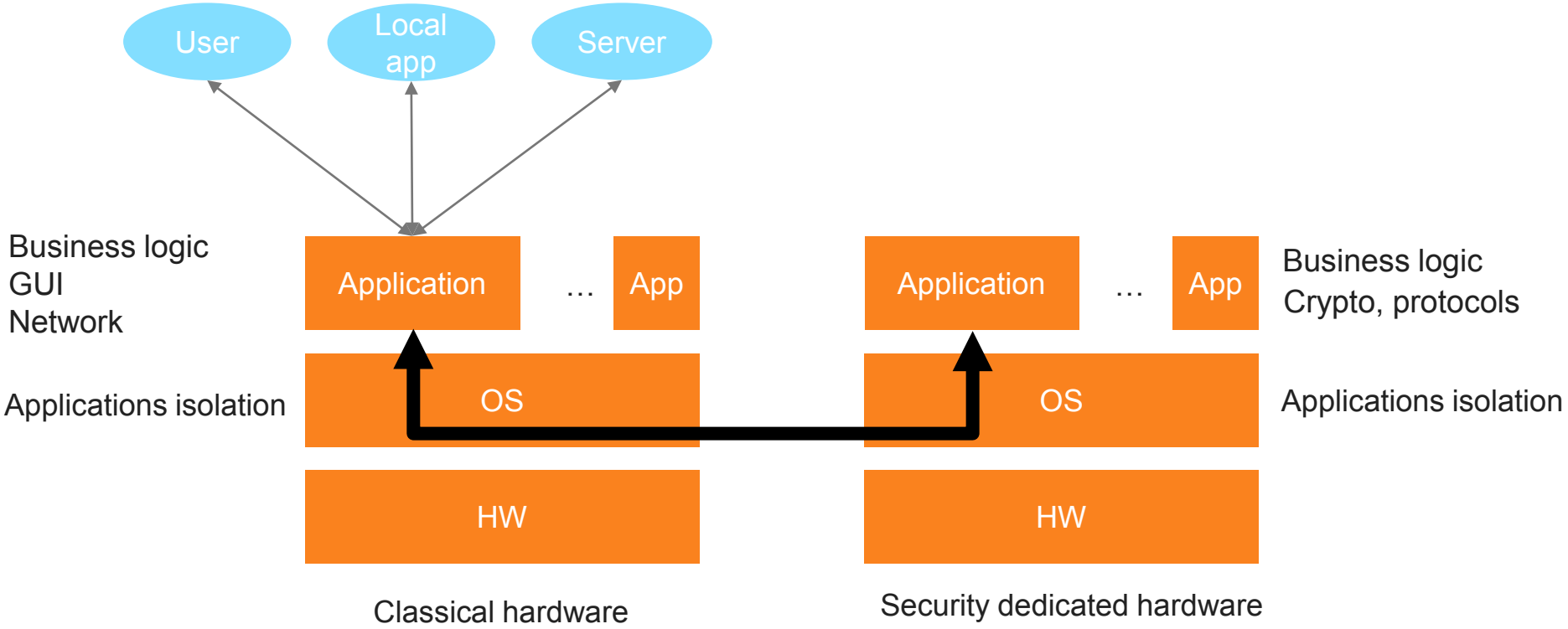
Simple implementation



Implements crypto, protocols, business logic, …

Implements applications isolation

Distributed implementation



Classical IT solutions

Soldered

Removable

Closed execution environment

TPM

Titan-M
Secure Enclave

SE

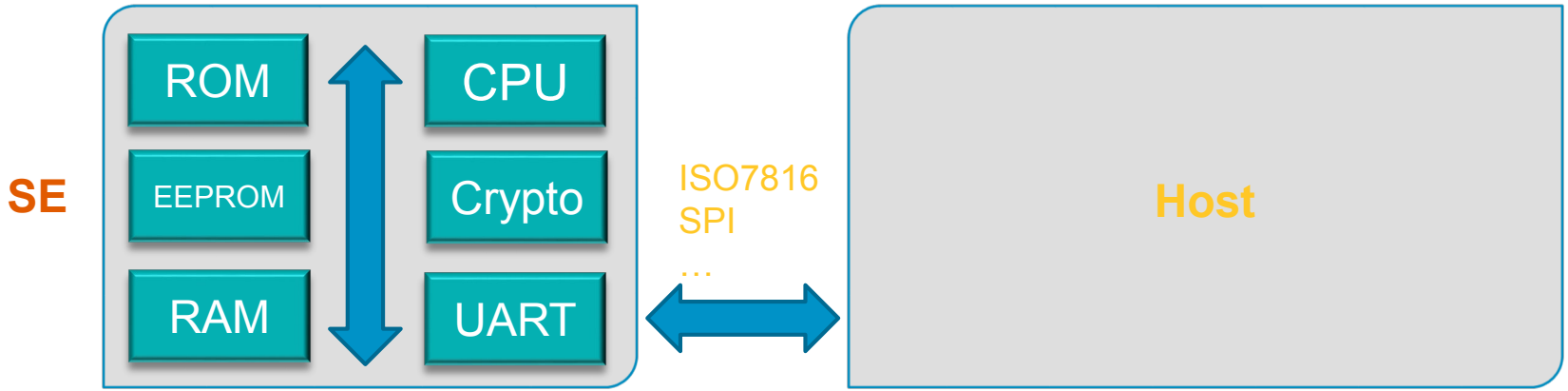
eUICC

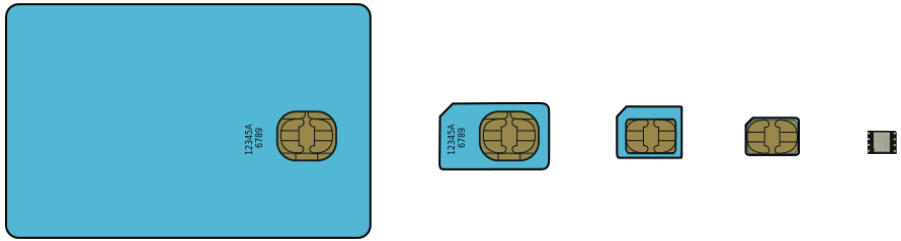
Classical hardware

Security dedicated HW

- Tamper resistant
- Managed
- Highly tested
- Certified

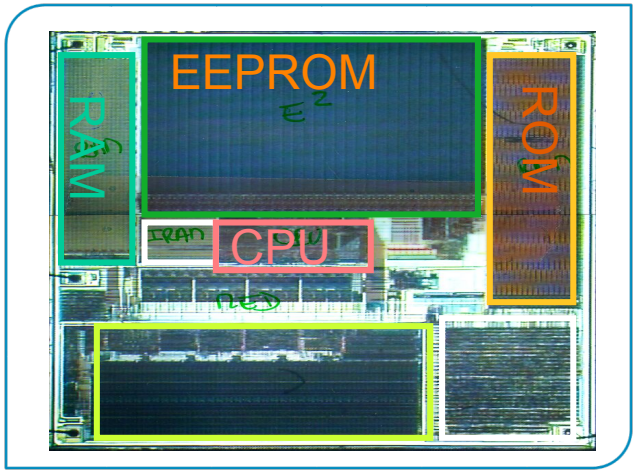
HW architecture of a **Secure Element**





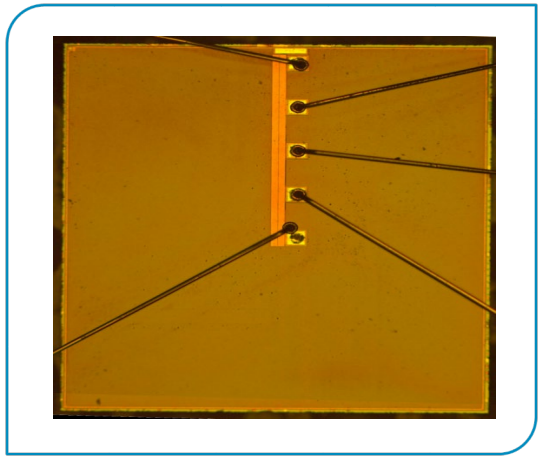
Tamper resistance at chip level

Classical hardware



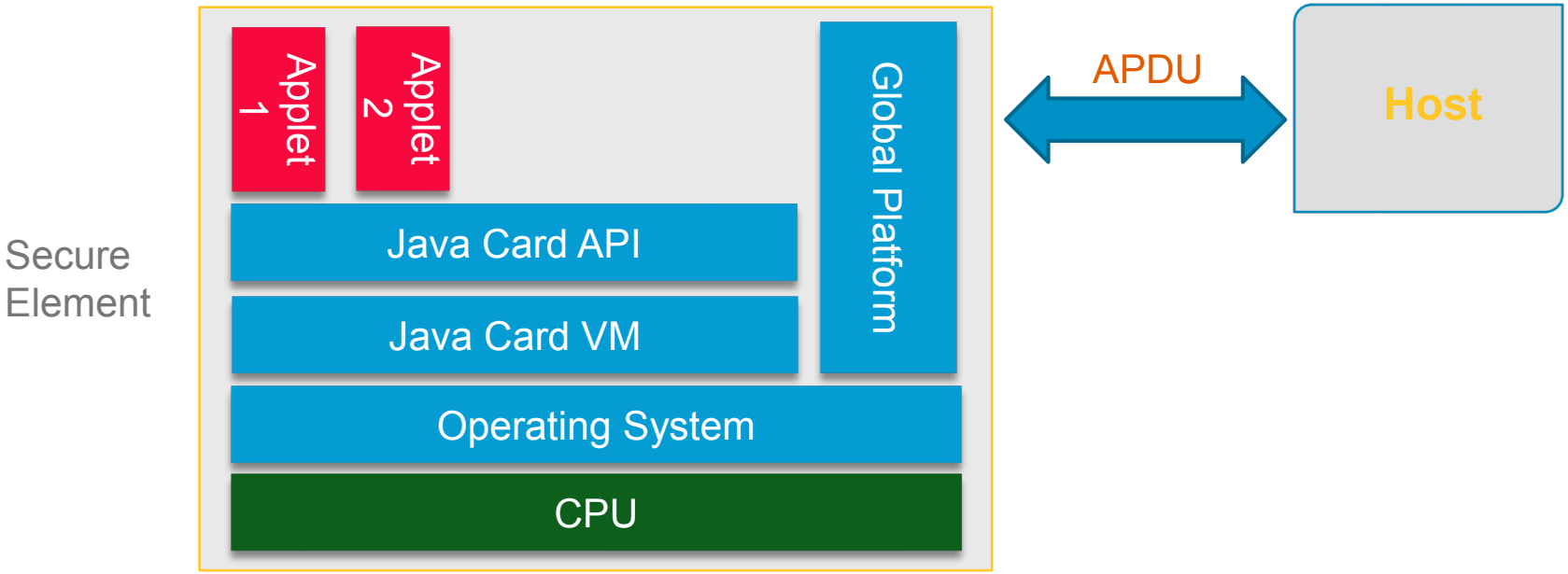
- Blocks can be easily identified
- No shield
- No glue logic
- Buses clearly visible

SE



- Shield
- Glue logic
- No Buses visible
- Memories and buses encryption
- Sensors

SW architecture of a **Secure Element**



Agenda

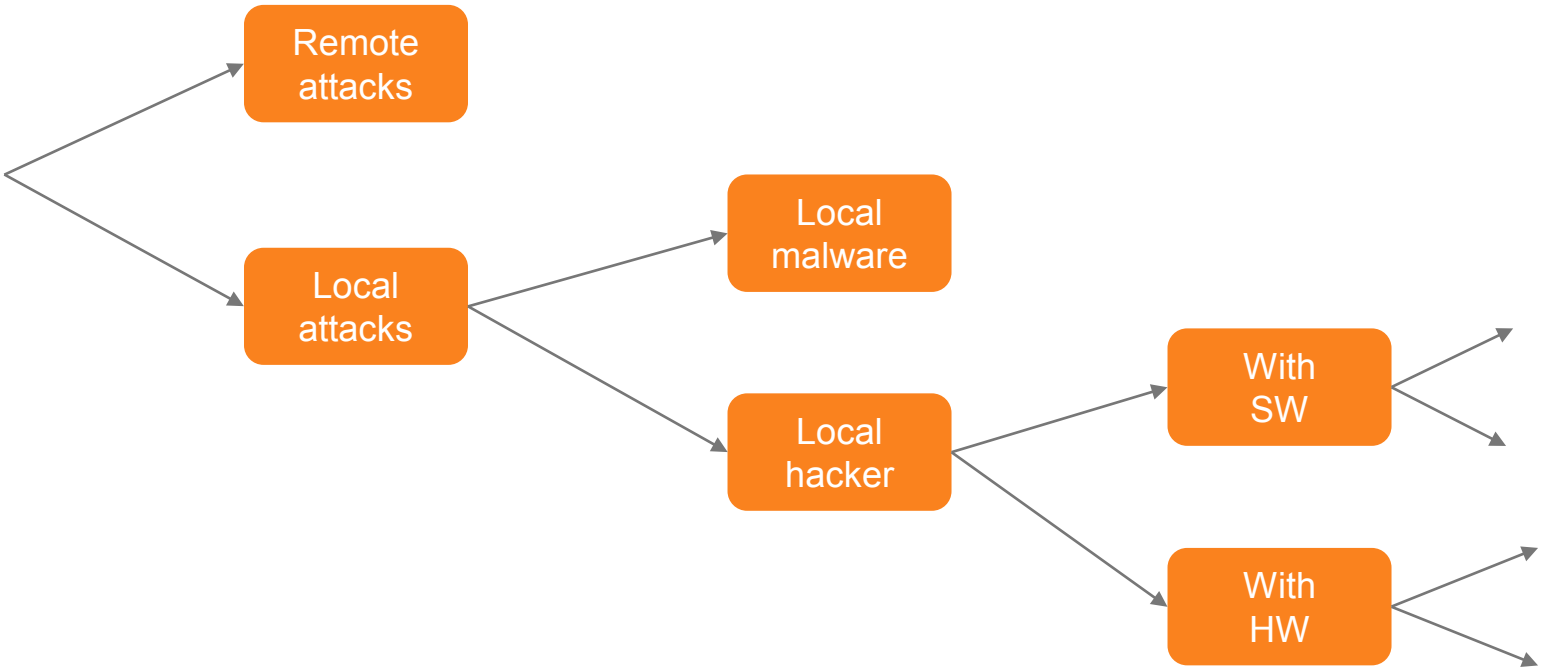
- Motivating example
- Security needs and classical solutions for IT world
- Sorting-out the available solutions for embedded world
- Introducing Multiple Levels of Security

Which security are we looking for ?

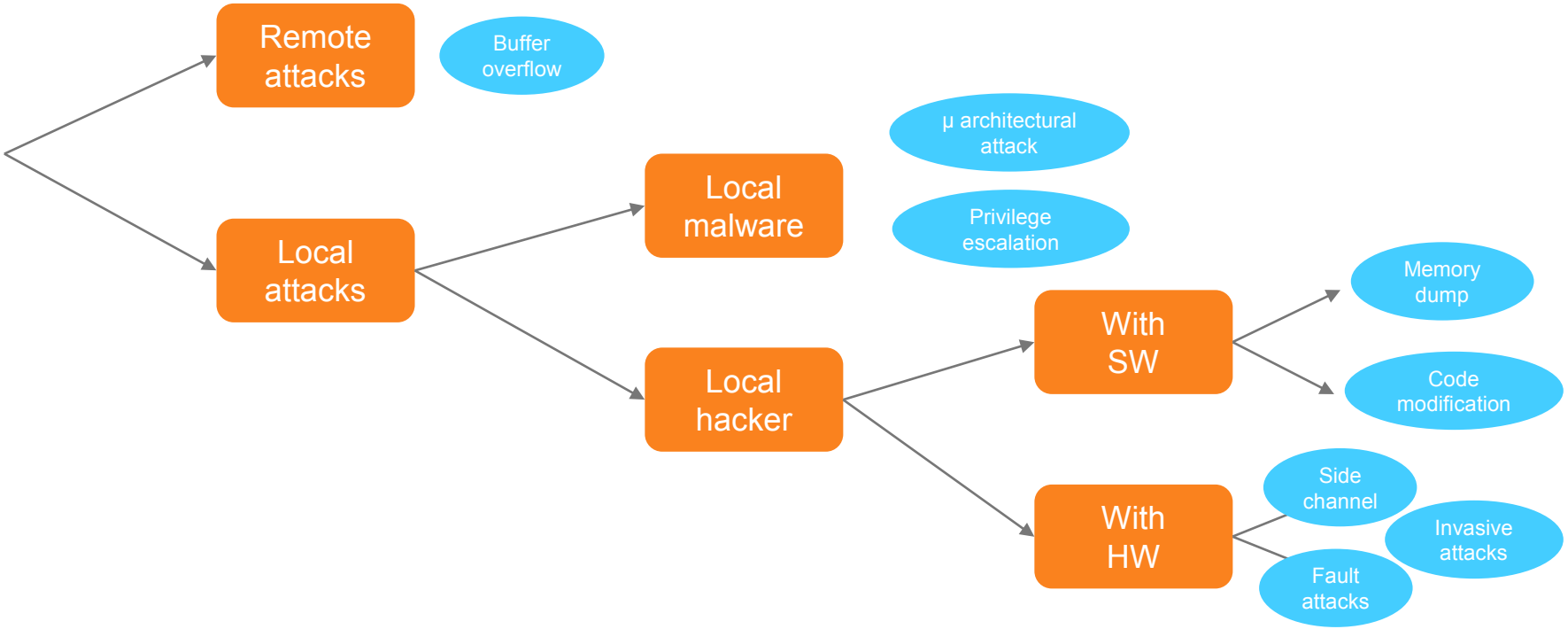
	Confidentiality	Integrity
Code	Proprietary algorithms	Business logic
Data	Keys	Certificates

The Root of Trust model for non monolithic hardware

Which attack model ?

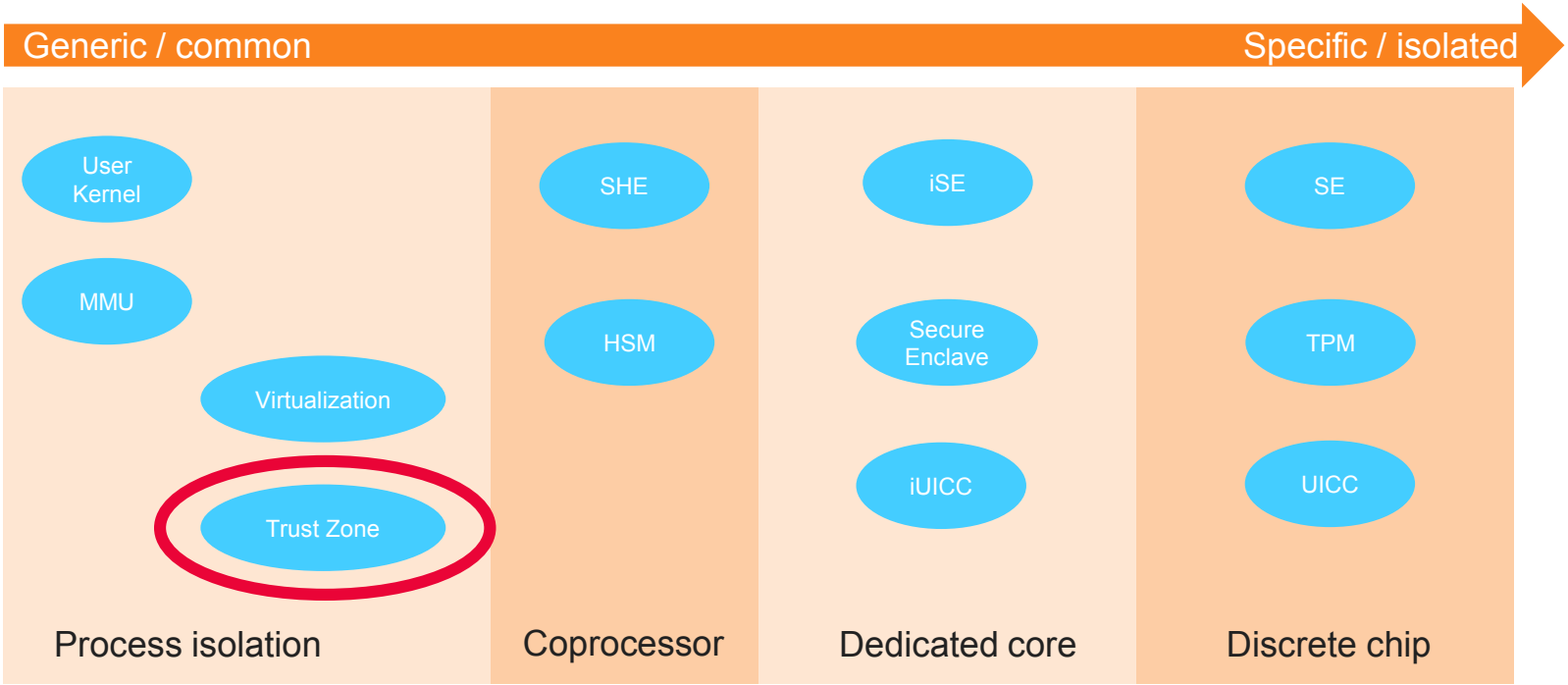


Mapping some attacks



Harder to protect against

Secure hardware classification



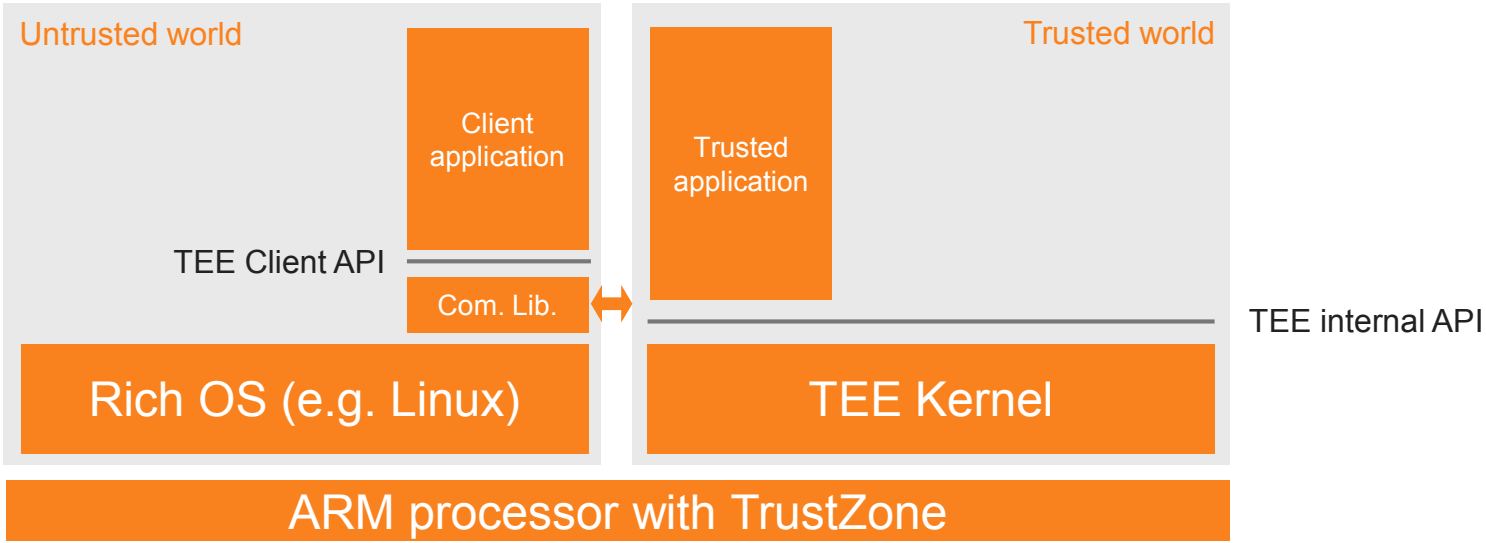
Dedicated memory ?

TEE 101

- TEE = Trusted Execution Environment
- Relies on ARM Trust Zone hardware feature
  - Trusted / Untrusted world partition (extended to peripherals)
- Rich OS runs in Untrusted world, TEE runs in Trusted world
- TEE and Rich OS run on the same processor, no tamper resistance

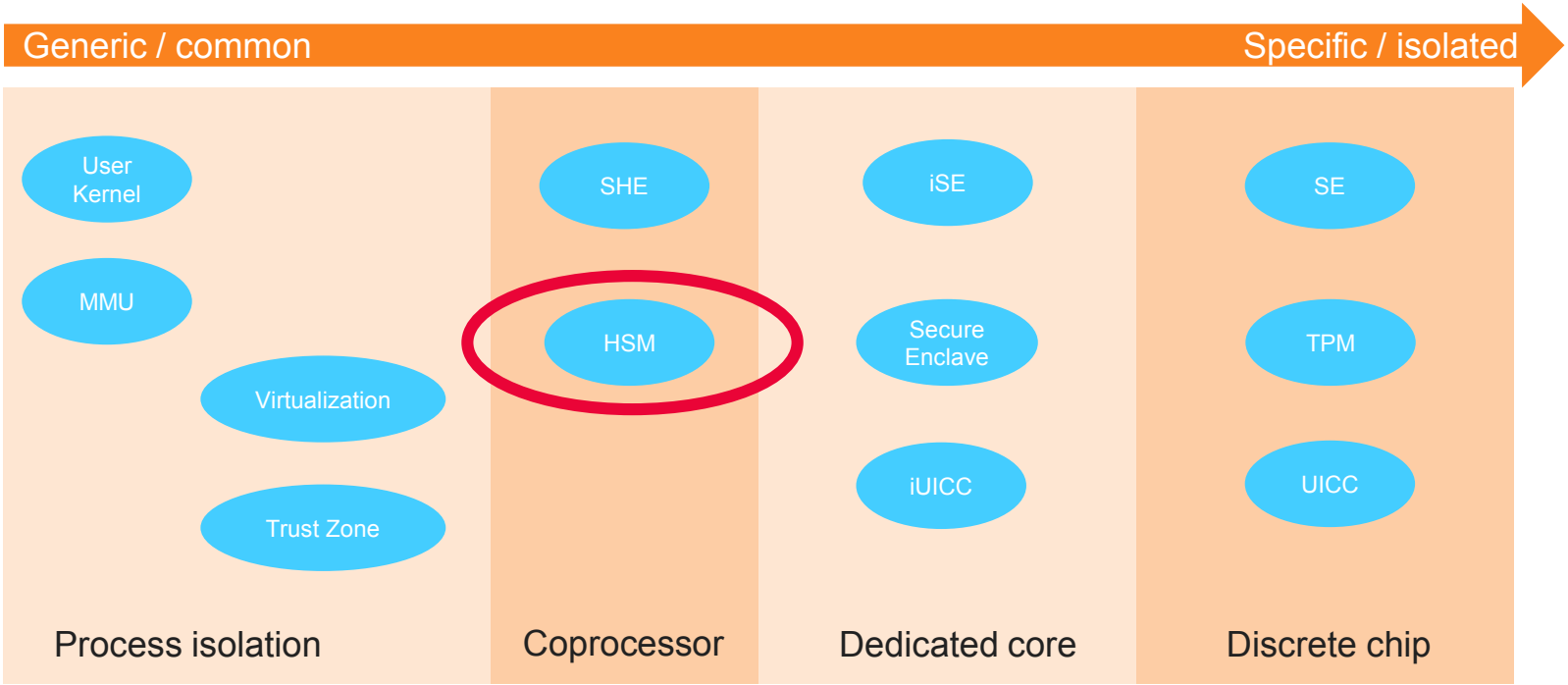
	Rich OS	TEE
Attack surface	Very large	Very limited
New features	Frequent	Very limited
Vulnerabilities discovery	Frequent	Very limited
Focus	Features, speed	Security

System architecture with TEE



Remarks: not all TEE allow third party developers to write and load their own TA

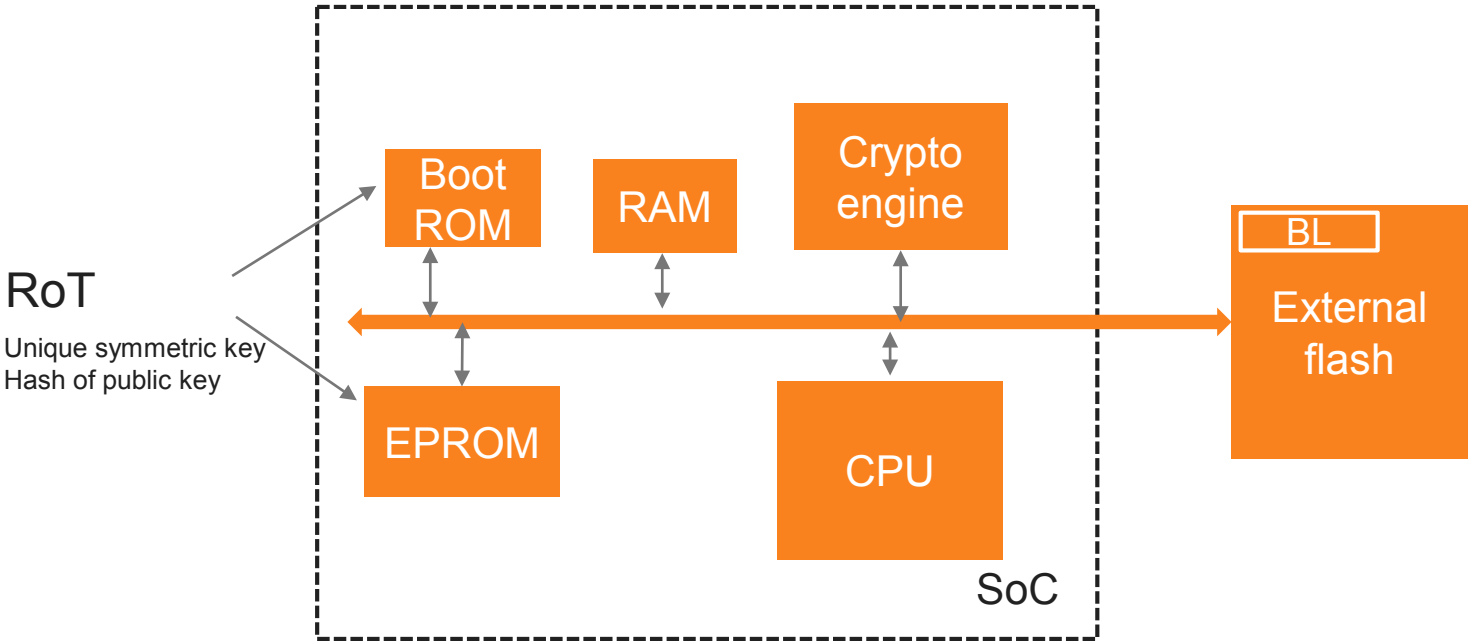
Secure hardware classification



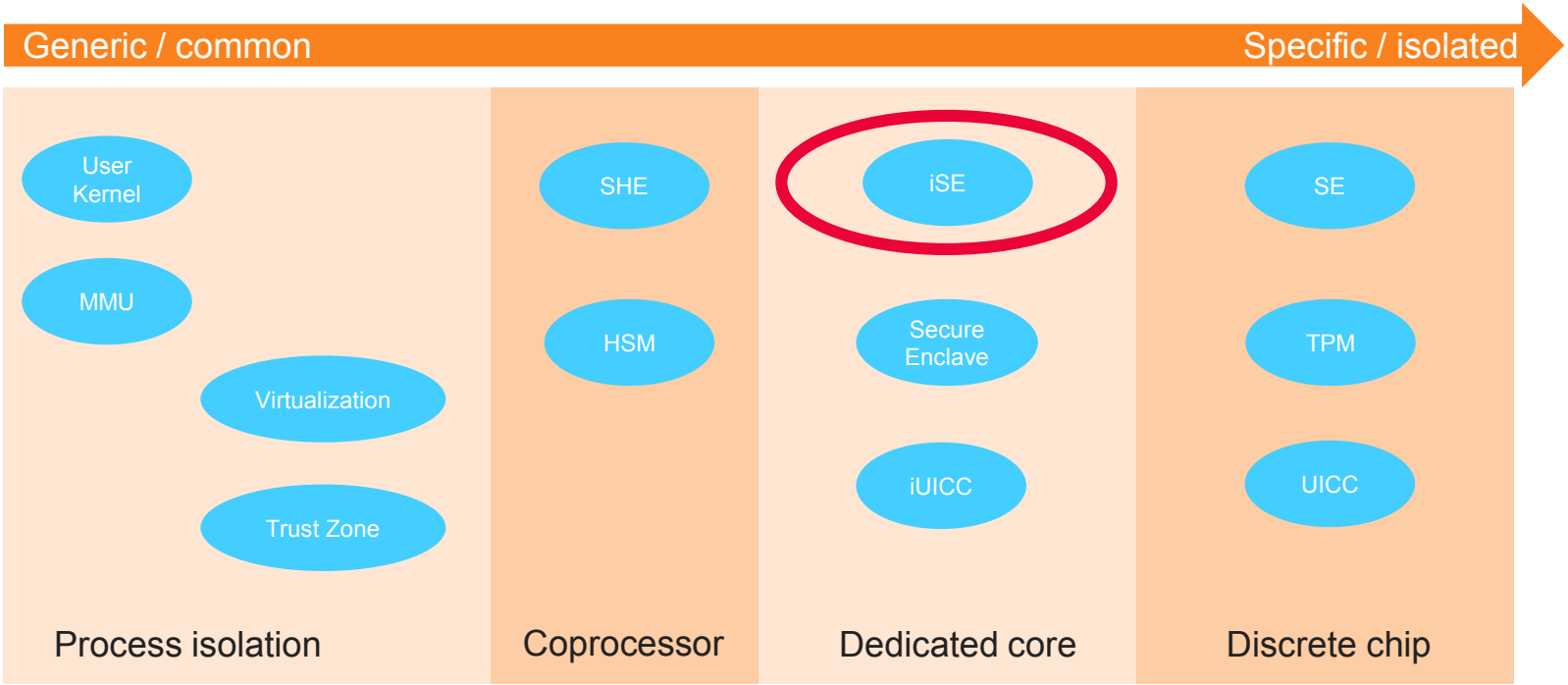
Dedicated memory ?

gemalto

System on Chip approach with crypto-coprocessor: NXP iMX.n



Secure hardware classification

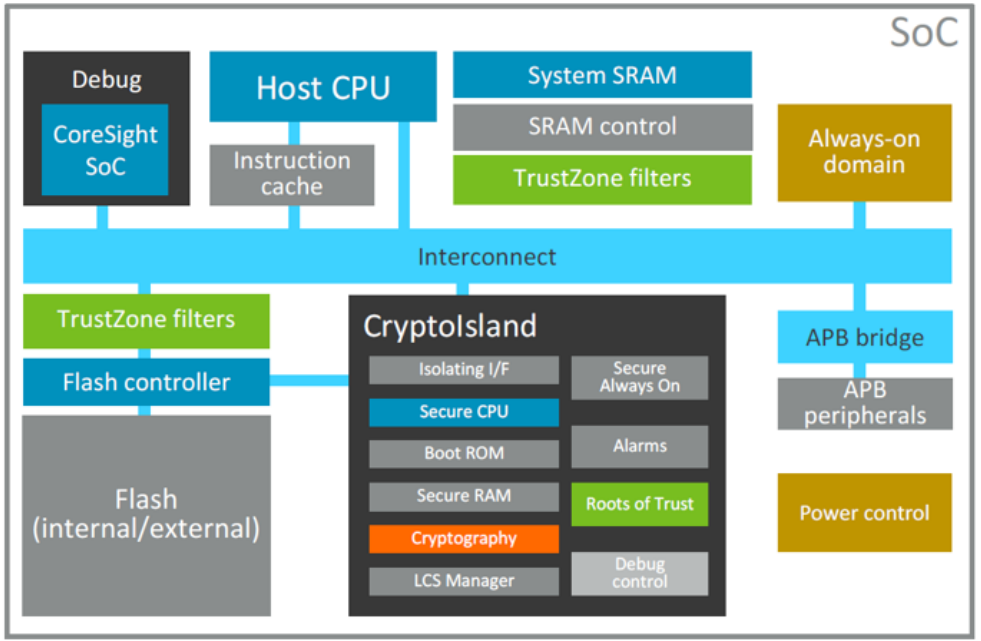


Dedicated memory ?

gemalto

System on Chip approach with dedicated core: ARM CryptoIsland

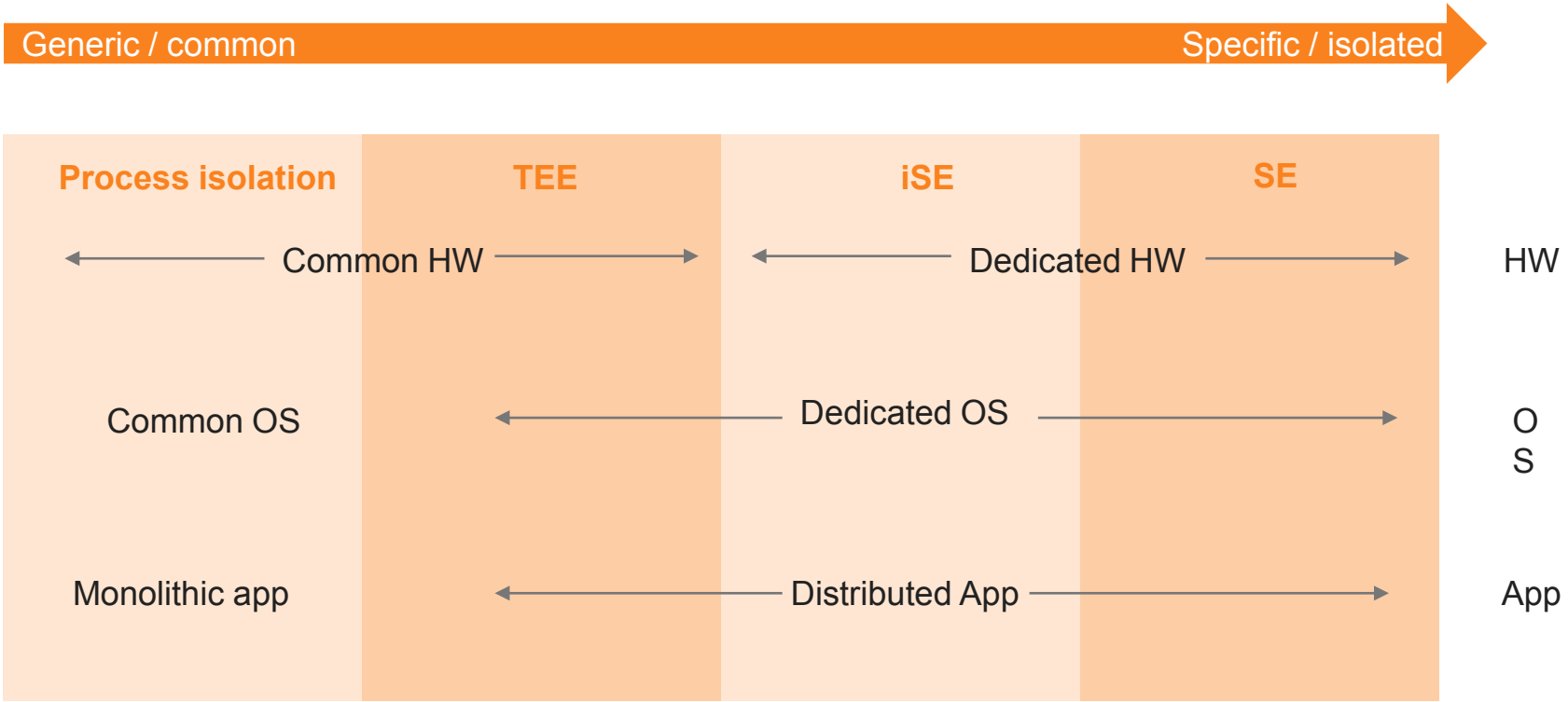
- A programmable security enclave to extend fixed function CryptoCell family
- **TrustZone CryptoIslands - an additional family of security solutions by Arm**
- Aimed at providing on-die security services, in a physically isolated manner (host CPU agnostic)
- Axiom: less sharing of resources leads to smaller attack surface and fewer vulnerabilities
- Certification, at a reasonable cost (i.e. reuse)



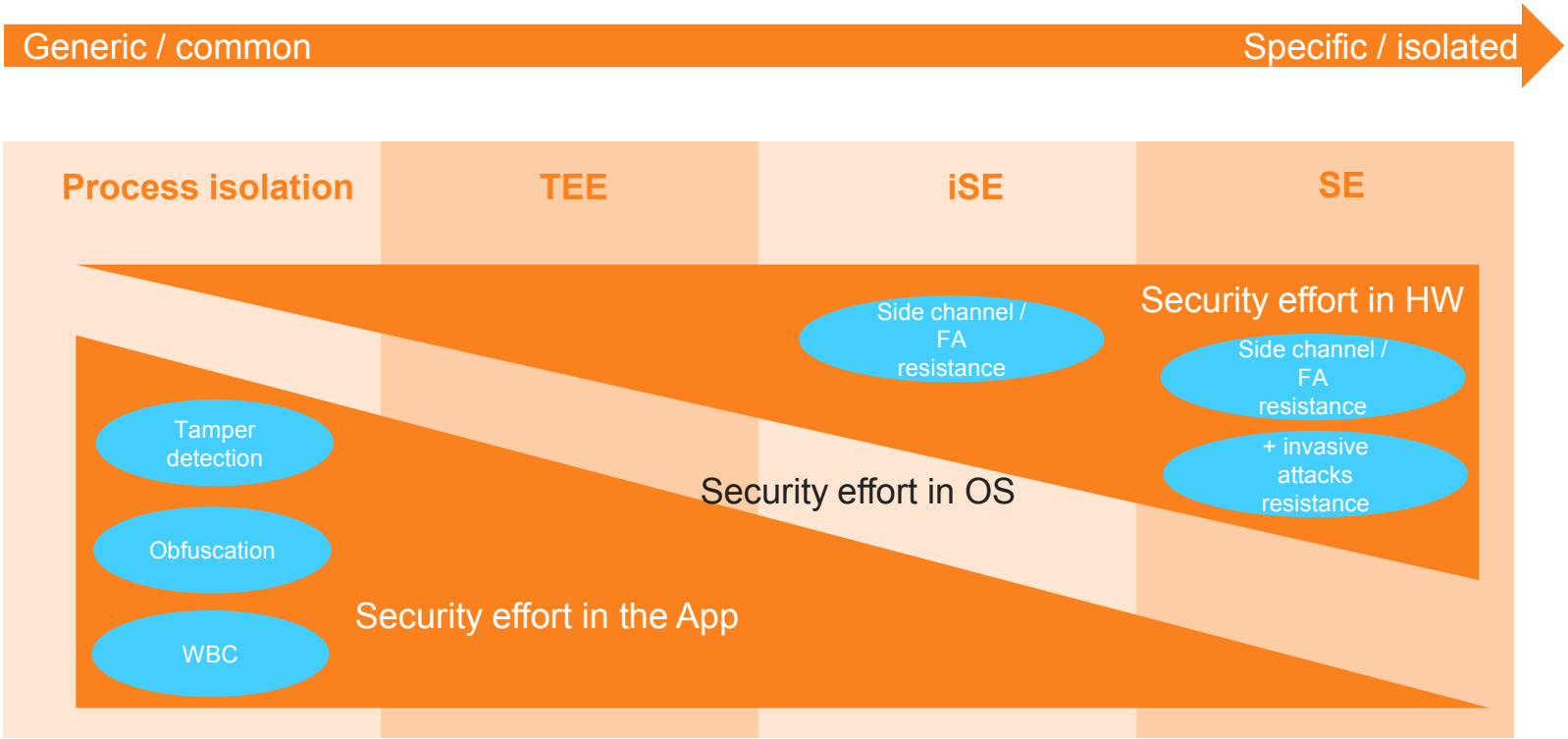
Support ARM v8-m & v7-m

arm TechCon 2017

Hardware security robustness on 4 solutions



Hardware security robustness on 4 solutions



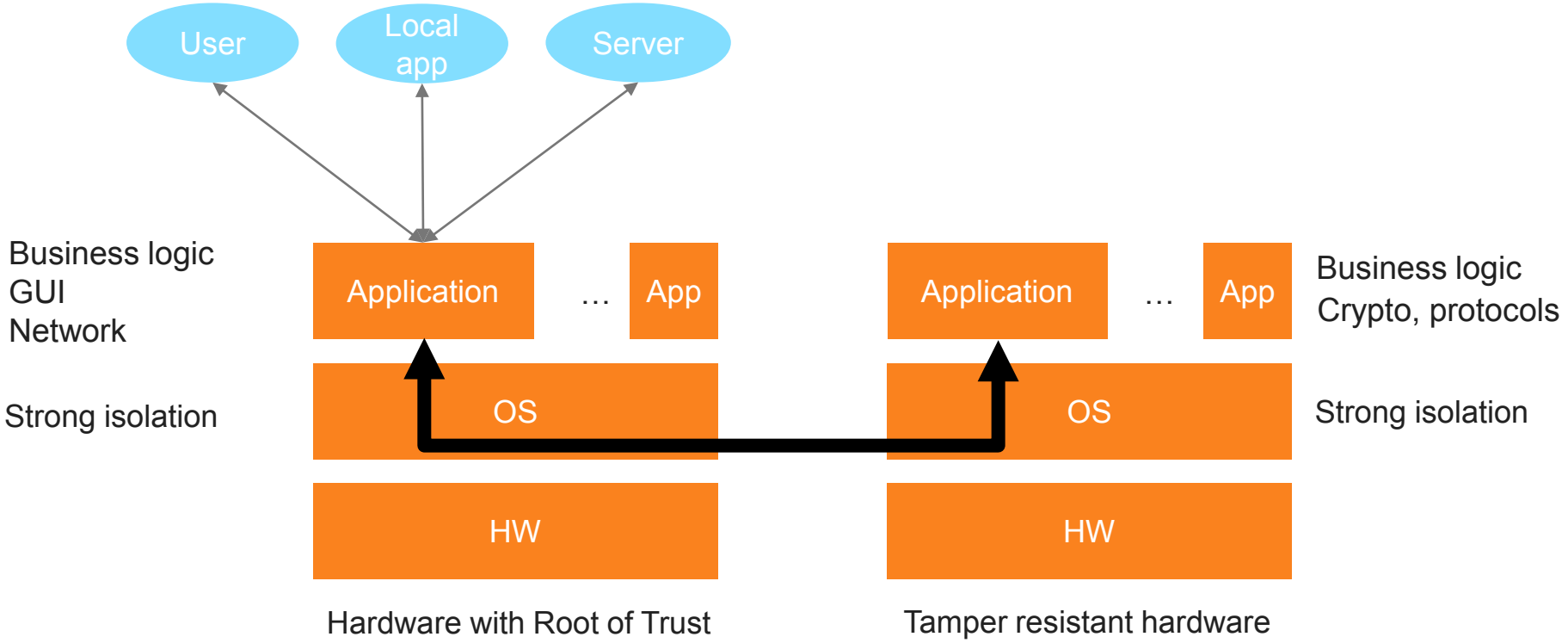
And what about certification ?

- Assurance versus resistance
- Target of Evaluation (IP versus product)
- Complexity limits ?
- Production ?
- Typical certification
  - TEE: EAL2+/3
  - SE: EAL5+/EAL6+

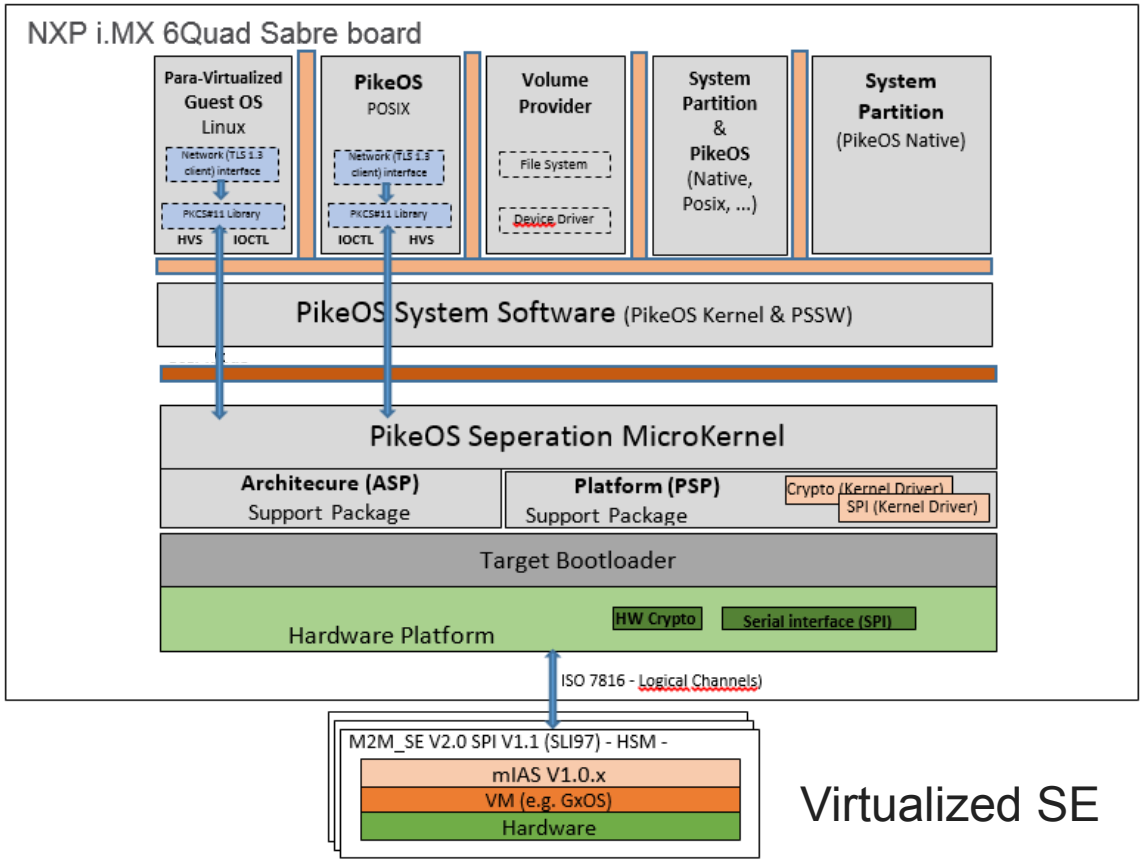
Agenda

- Motivating example
- Security needs and classical solutions for IT world
- Sorting-out the available solutions for embedded world
- Introducing Multiple Levels of Security

The ideal world: Multiple Levels of Security

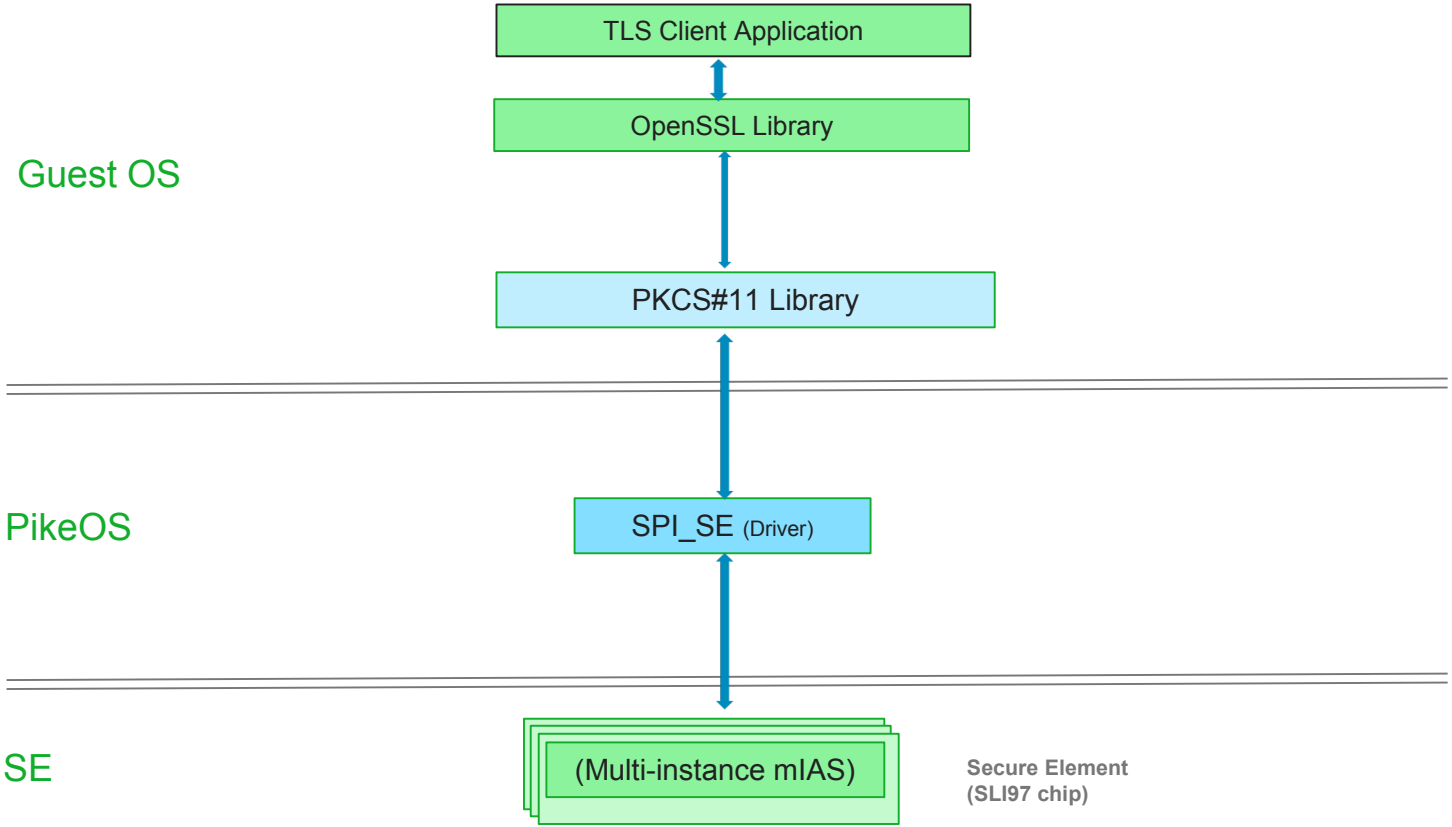


Case study: TLS on PikeOS / iMX.6 + SE



Virtualized SE

Layered Software Architecture



Take away

- Hardware security landscape still needs clarification
- Security model and requirements are key to pick a solution
- Security design patterns still need to be established
- Certification is mandatory to establish trust