Trusted Computations in Vehicular Environments

Marc Lacoste

Orange Innovation

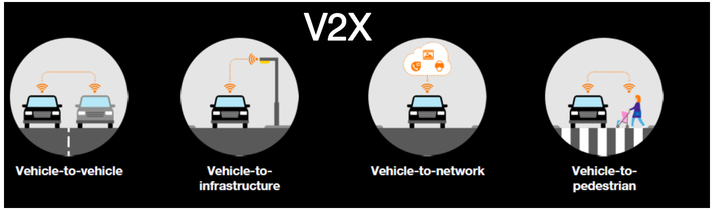
Journées Sécurité, October 14-15, 2021

Outline

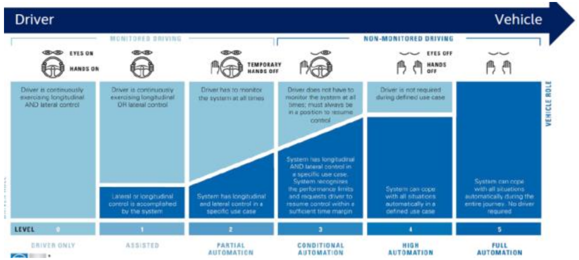
- (Beyond) 5G vehicular isolation & trust
- The TEE approach
- TEE architectures
  - Intel SGX and other TEEs
  - Isolation and resilience framework for V2X
- New directions
  - Confidential computing
  - Decentralized protocols
  - Integration with ML

connected & autonomous vehicles : security, safety & privacy concerns

increasing connectivity



increasing autonomy



increasing complexity

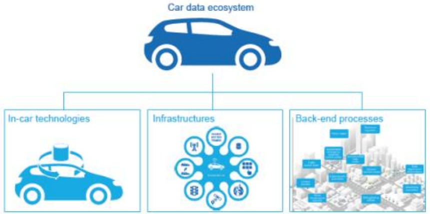
growing # of attacks

safety-security gap

safety-critical failures

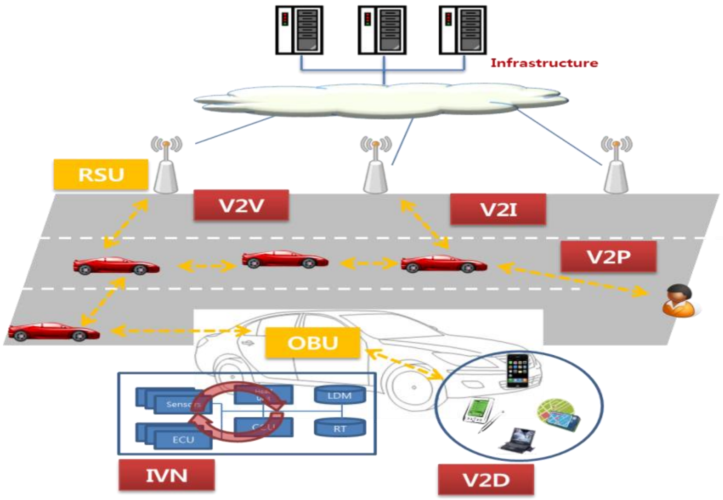
cyber-exposed attacks

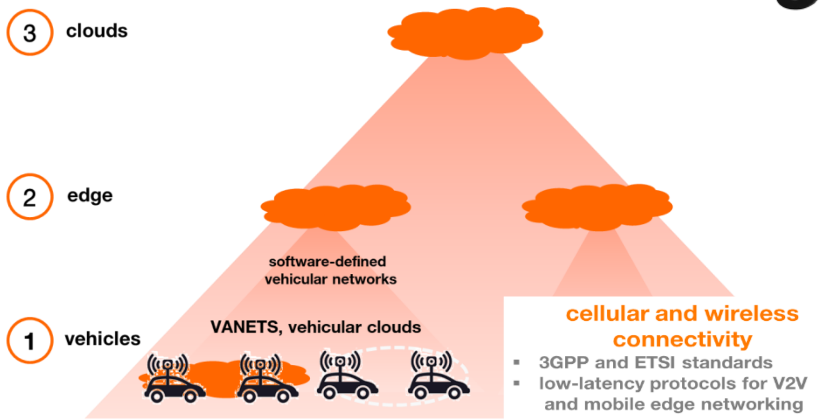
data-shaped ecosystem



isolation → ← trust

B5G vehicular networking magnifies security & safety challenges





V2X ecosystem	vehicle	network	cloud
software	Threats	Threats	Threats
hardware	Threats	Threats	Threats

B5G vehicular networking magnifies security & safety challenges

isolation :
which place for protection mechanisms in a multi-tier ecosystem?

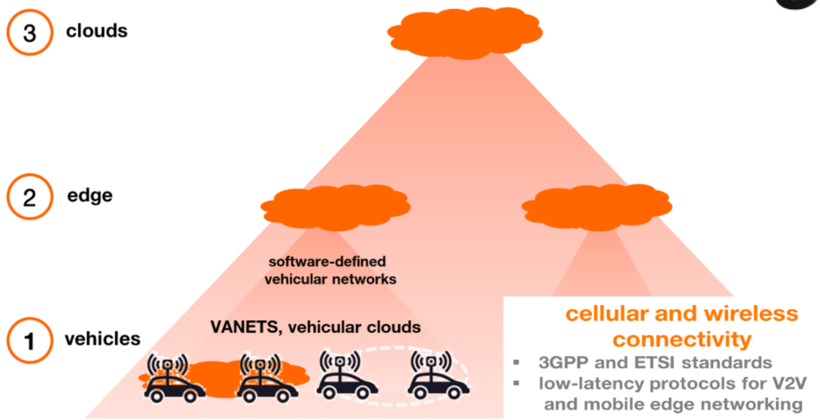
- network connections
- multi-tenancy
- system-to-network, end-to-end

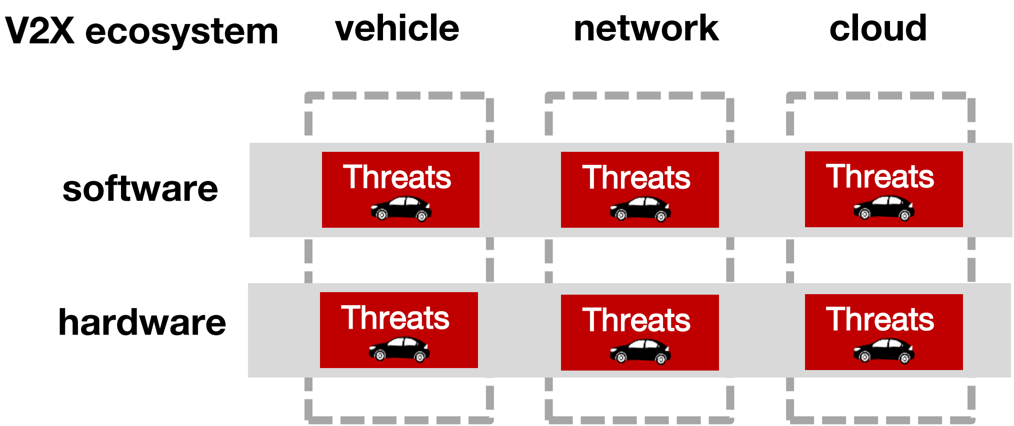
trust :
how to guarantee data protection?

- confidentiality and privacy
- authenticity and integrity of information sources
- relation with safety

a holistic vision of protection is needed :

- software and hardware
- for vehicle, network, and cloud tiers
- covering the full data life-cycle





B5G vehicular networking magnifies security & safety challenges

isolation :
which place for protection mechanisms in a multi-tier ecosystem?

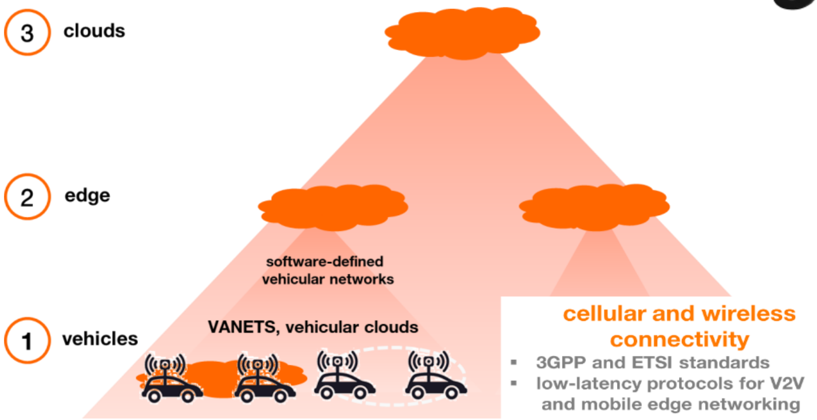
- network connections
- multi-tenancy
- system-to-network, end-to-end

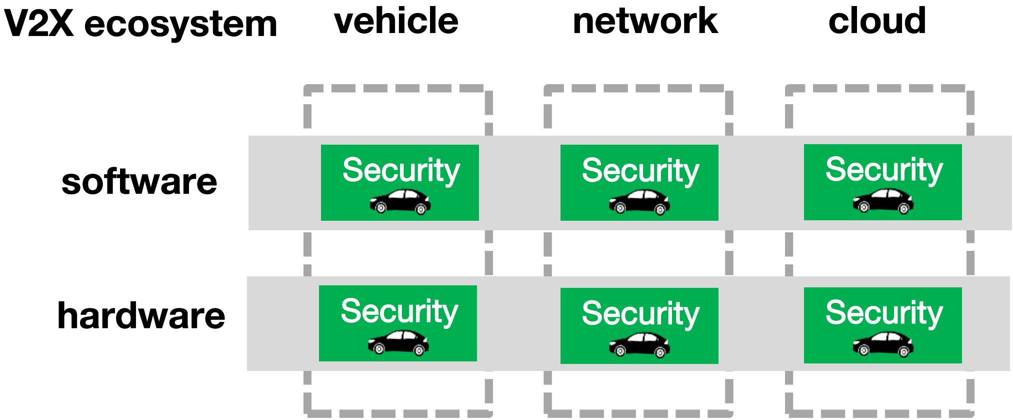
trust :
how to guarantee data protection?

- confidentiality and privacy
- authenticity and integrity of information sources
- relation with safety

a holistic vision of protection is needed :

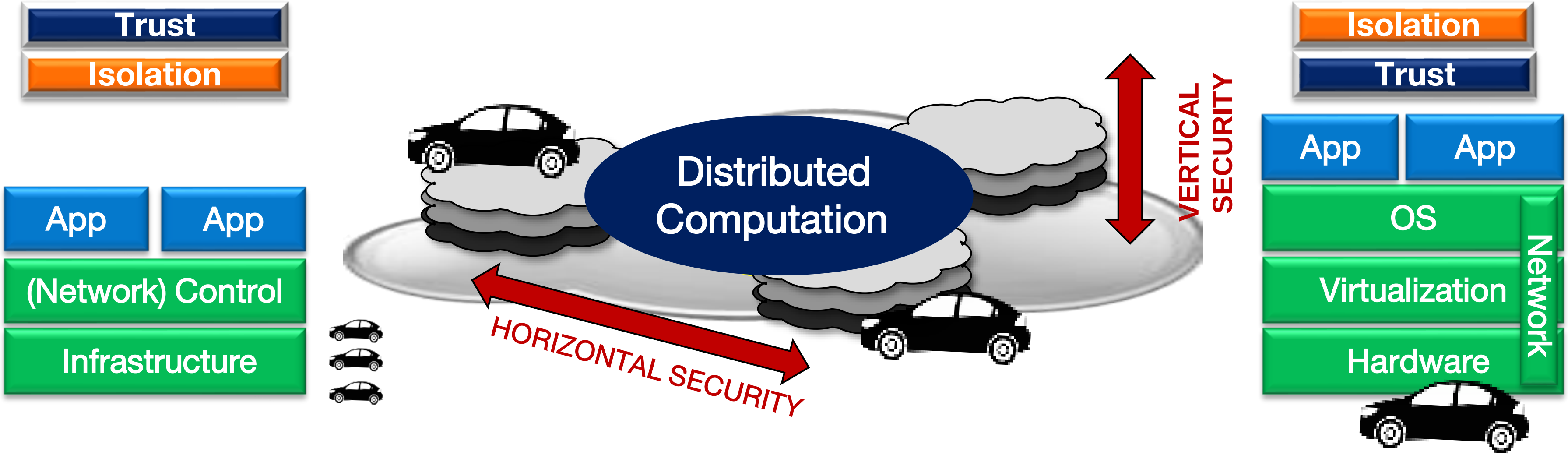
- software and hardware
- for vehicle, network, and cloud tiers
- covering the full data life-cycle





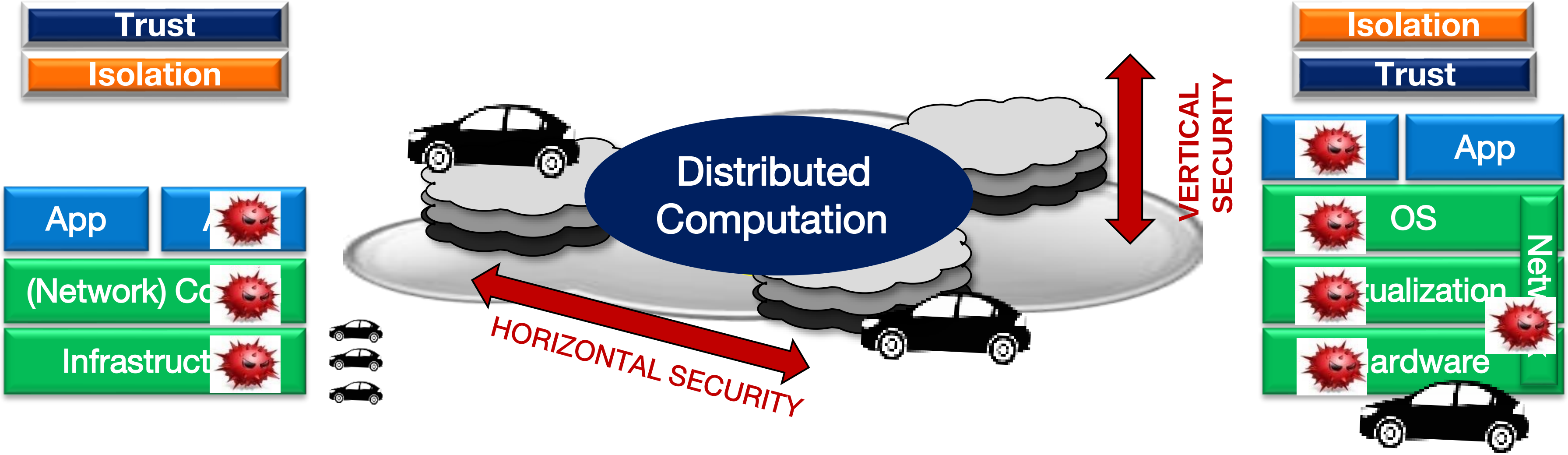
problem statement revisited

(Beyond) 5G infrastructures are virtualized, multi-domain and multi-layered, with many threats



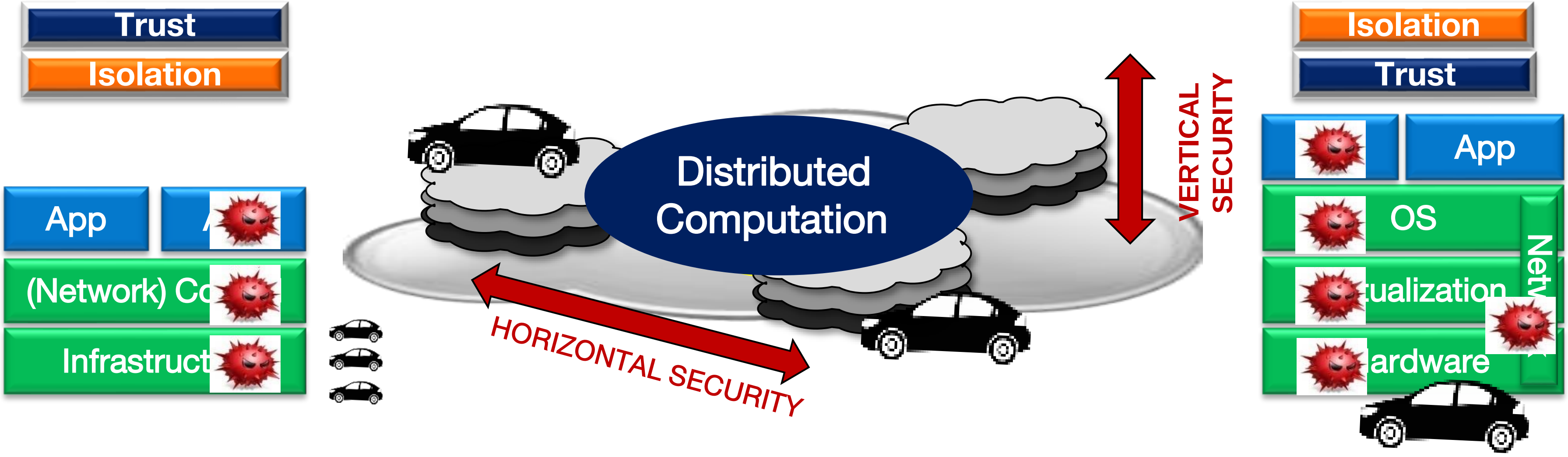
problem statement revisited

(Beyond) 5G infrastructures are virtualized, multi-domain and multi-layered, with many threats



problem statement revisited

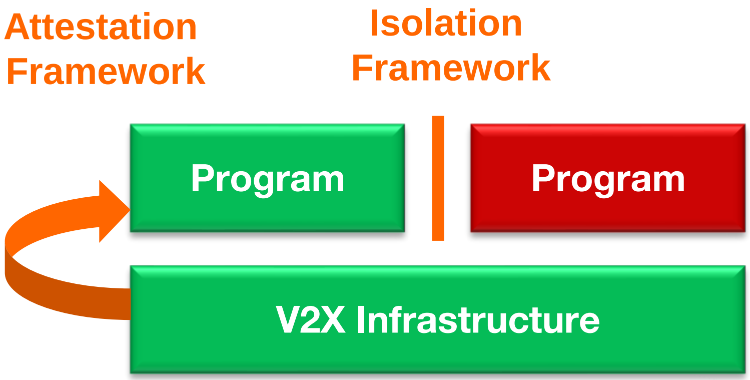
(Beyond) 5G infrastructures are virtualized, multi-domain and multi-layered, with many threats



how to perform (distributed) computations securely over untrusted B5G vehicular infrastructures?

security properties and primitives

PROTECTING THE INFRASTRUCTURE



Program Sandboxing

Confine untrusted programs
Protect system from their actions

Platform Attestation

Guarantee that the platform runs trustworthy hardware, firmware, and software before transferring computation and data

PROTECTING COMPUTATIONS

Isolated Execution

Protect code from the rest of the system

Isolated compartments

- CREATE / DESTROY
- ENTER / EXIT

Confidentiality:

- "Black box" execution of programs
- Secure communication of data with untrusted outside world

Shielded Execution

Secure load and store data

- Secure LOAD / STORE

Privacy: Private Tamper-Evident EE

Integrity:

- The system cannot affect the behavior of programs that run as on reference platform

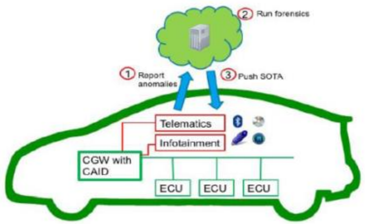
Attestation Proof of correct execution

- ATTEST / VERIFY

evolutions of in-vehicle architecture

in-vehicle HW architecture is increasingly virtualized, raising isolation concerns

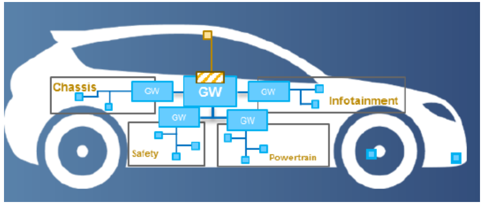
Hardware ECUs



Source: Wasicek et al. Context-aware Intrusion Detection in Automotive Control Systems. *ESCAR Conference,* 2017.

- Safety-critical vehicle functions connected by vulnerable HW bus
- Cyber-resilience: propagation of failures and attacks through vulnerable gateway
- Challenges:
  - ECU protection
  - In-vehicle network protection
  - Gateway protection

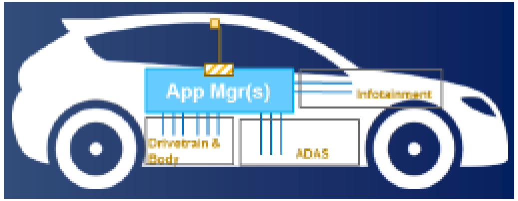
Domain-based ECUs



Source: NXP.

- ECUs grouped into domains for broad functional areas
- ECU domains isolated / monitored by Domain Controllers
- Challenges:
  - Inter-domain isolation
  - Trade-offs

Virtualized ECUs



Source: NXP.

- ECUs as virtualized execution environments (e.g., VMs, containers)
- Distributed computations across ECUs / vehicles
- Challenges:
  - EE isolation
  - Untrusted EE platform
  - Side-channels

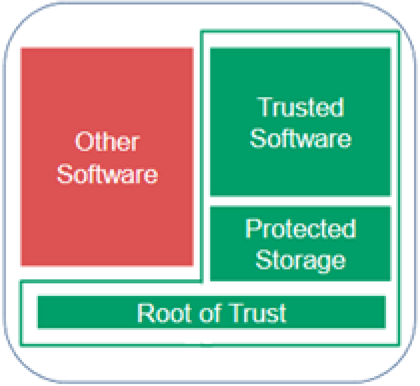
⇨ hardware trusted execution execution technologies

Outline

- (Beyond) 5G vehicular isolation & trust
- The TEE approach
- TEE architectures
  - Intel SGX and other TEEs
  - Isolation and resilience framework for V2X
- New directions
  - Confidential computing
  - Decentralized protocols
  - Integration with ML

Trusted Execution Environment

hardware support to run arbitrary code in a confined environment :
guarantees tamper-resistant execution of applications



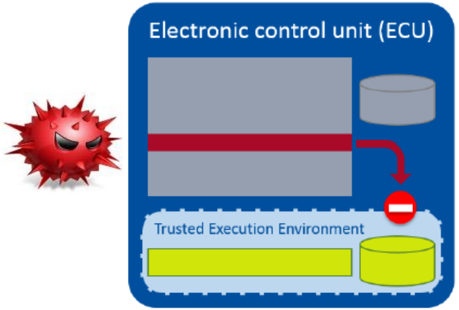
- **isolated execution**
- **tamper-resistant storage: sealing**
  create, store, and manage secrets in a controlled environment
- **reporting to a remote verifier: attestation**
  extend trust to internal and external entities
- **secure provisioning**
- **trusted path**

Source: https://asokan.org/asokan/research/Blockchains-and-hw-security.pdf

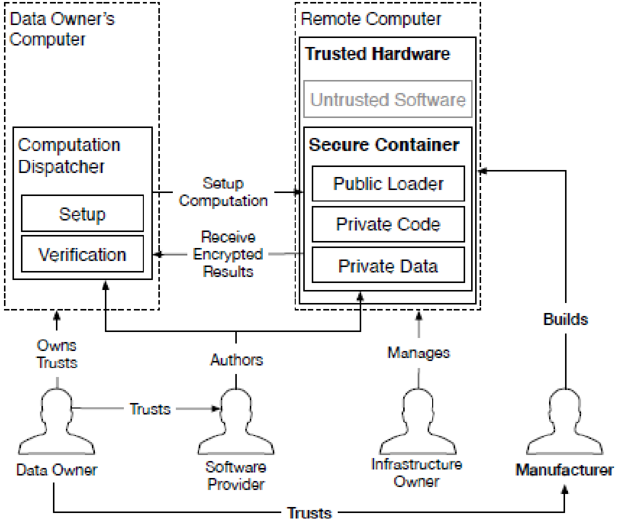
TEE guarantees isolation + trust

Isolation

protected compartment concept



Source: Jens Köhler and Henry Förster. Trusted Execution Environments in Vehicles for Secure Driver Assistance Systems, 2017, Springer.

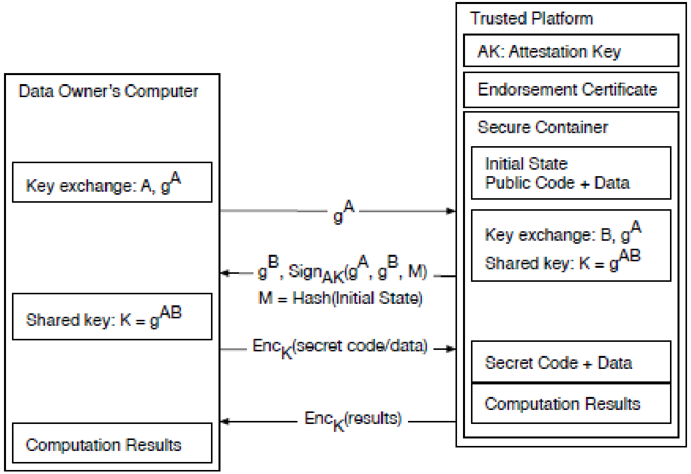


- security-sensitive state (code + data) in TEE cannot be corrupted from outside of TEE
- trusted hardware protects integrity and confidentiality of computations
- multiple concurrent compartments

Trust

secure remote computation

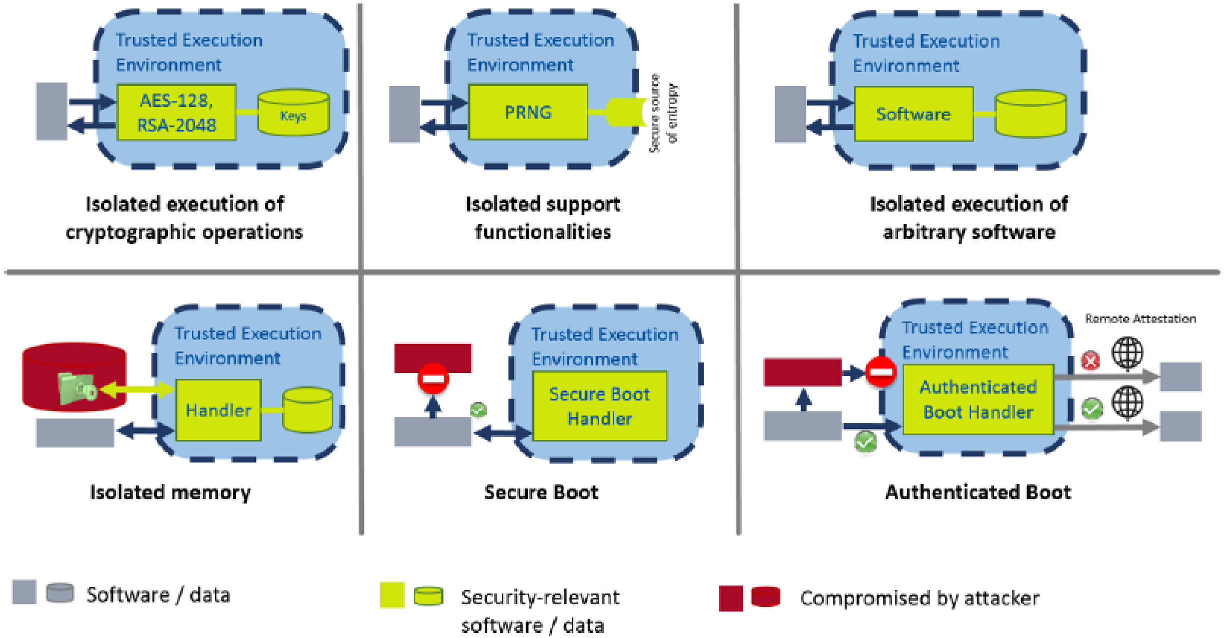
- prove to remote party it is talking to software located in secure container hosted on trusted hardware
- attestation key
- endorsement certificate



Source: V. Costan, I. Lebedev and S. Devadas. Secure Processors Part I: Background, Taxonomy for Secure Enclaves and Intel SGX Architecture. *Foundations and Trends in Electronic Design Automation,* vol. 11, no. 1-2, pp. 1–248, 2017.

TEE features

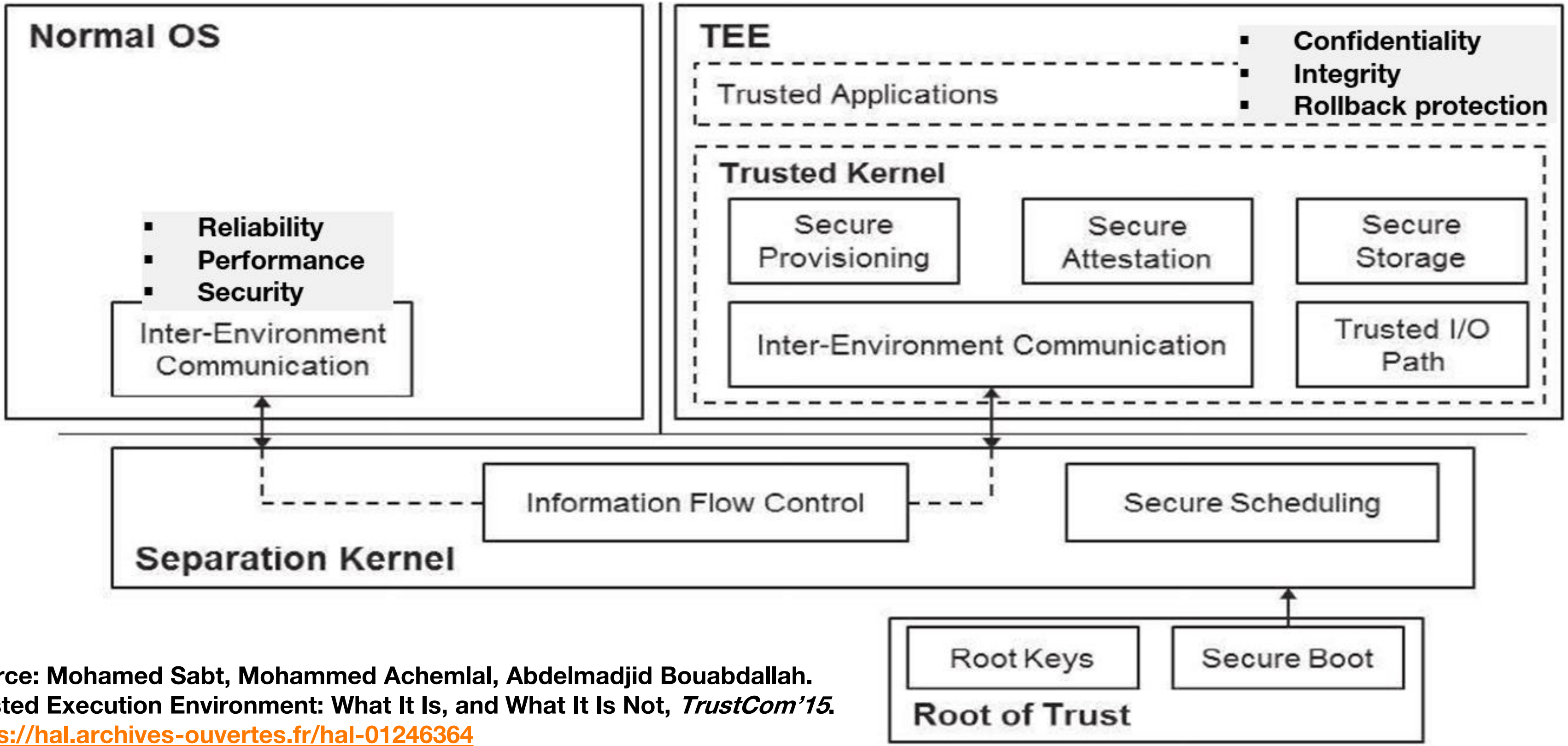
- isolation
  - access control to code and data
  - well-defined entry point
  - concurrent modules
- attestation
  - prove to third party attested state
  - locally or remotely
  - measurement during init
- sealing
  - confidential data can be unwrapped under some conditions
  - encryption



Source: J. Köhler and H. Förster. Trusted execution environments in vehicles for secure driver assistance systems, 2017, Springer.

- DRoT
  - trust chains
  - TOCTOU vulnerabilities
- code and data confidentiality
- side-channel resistance
- memory protection

TEE architecture

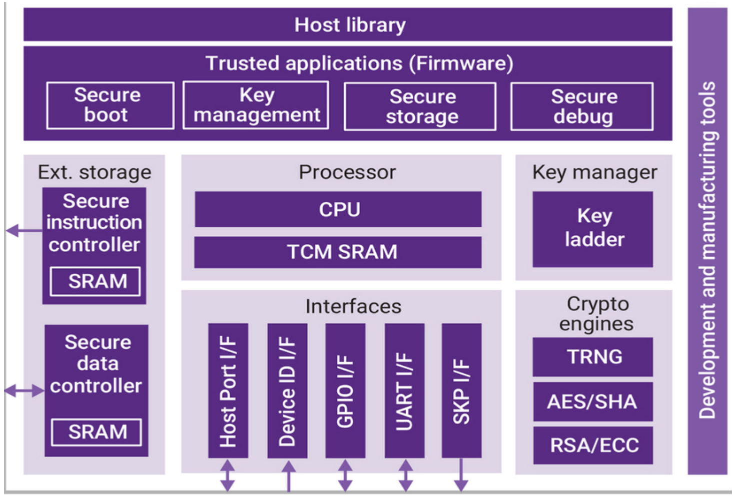


Source: Mohamed Sabt, Mohammed Achemlal, Abdelmadjid Bouabdallah. Trusted Execution Environment: What It Is, and What It Is Not, *TrustCom'15*.
https://hal.archives-ouvertes.fr/hal-01246364

key challenges

isolation

- **attack surface:**
  - TCB size
  - hardware or software TCB?
  - side-channel attacks, exceptions
- **flexibility:**
  - dynamic/upgradeable protected space
  - concurrent compartments
  - compartment size limitations



attestation

- **secure proofs**
- **large code size**
- **low overhead:**
  - secure element resource usage
  - communication
  - proof verification

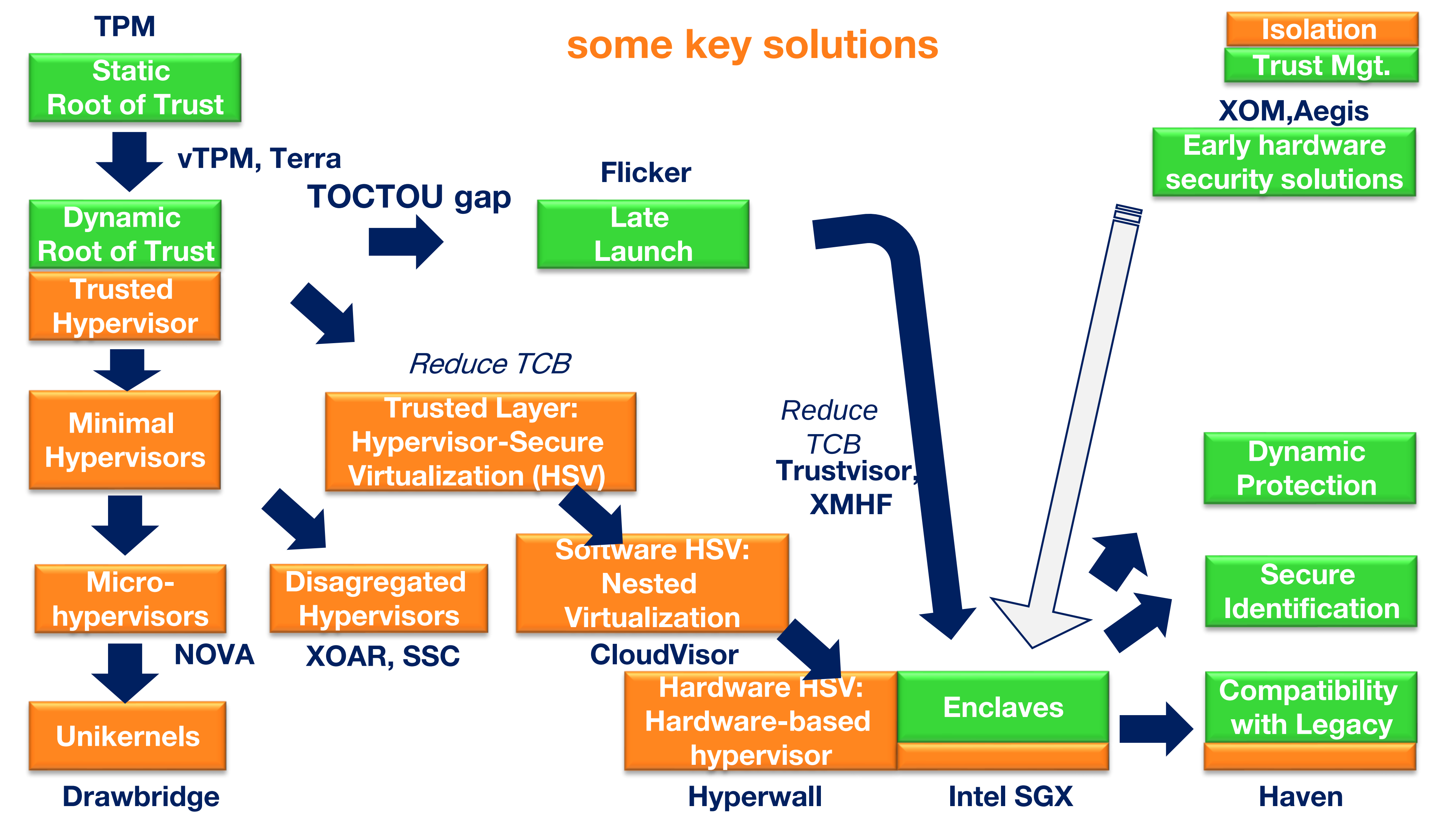
compatibility with legacy

- **unmodified binaries support**
- **on-chip or co-processor?**
- **independence from hardware**
- **easy access to specifications**
- **amount of trust in provider**

Source: https://semiengineering.com/dodging-the-next-generation-of-car-thieves/

Outline

- (Beyond) 5G vehicular isolation & trust
- The TEE approach
- TEE architectures
  - Intel SGX and other TEEs
  - Isolation and resilience framework for V2X
- New directions
  - Confidential computing
  - Decentralized protocols
  - Integration with ML


some key solutions
Isolation
Trust Mgt.
TPM
Static Root of Trust
vTPM, Terra
Dynamic Root of Trust
Trusted Hypervisor
TOCTOU gap
Flicker
Late Launch
XOM,Aegis
Early hardware security solutions
Reduce TCB
Minimal Hypervisors
Trusted Layer: Hypervisor-Secure Virtualization (HSV)
Reduce TCB
Trustvisor, XMHF
Dynamic Protection
Micro-hypervisors
Disagregated Hypervisors
Software HSV: Nested Virtualization
Secure Identification
NOVA
XOAR, SSC
CloudVisor
Hardware HSV: Hardware-based hypervisor
Enclaves
Compatibility with Legacy
Unikernels
Drawbridge
Hyperwall
Intel SGX
Haven

for V2X

HSM

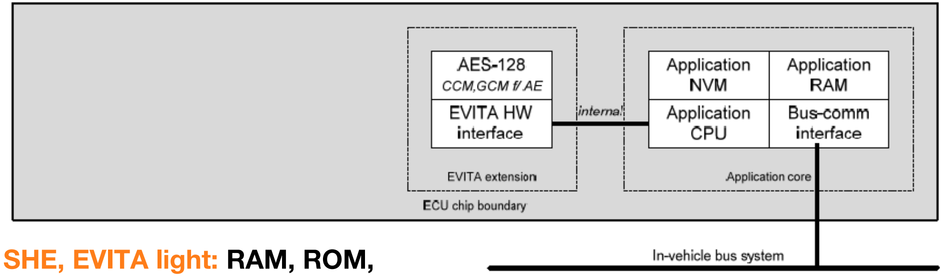
- hardware module runs software components isolated from other software components
- SoC, external module, Integrated Circuit
- examples:
  - Secure Hardware Extension (SHE), EVITA HSM
  - TPM
  - Smart cards (eSIM)

CPU security extensions

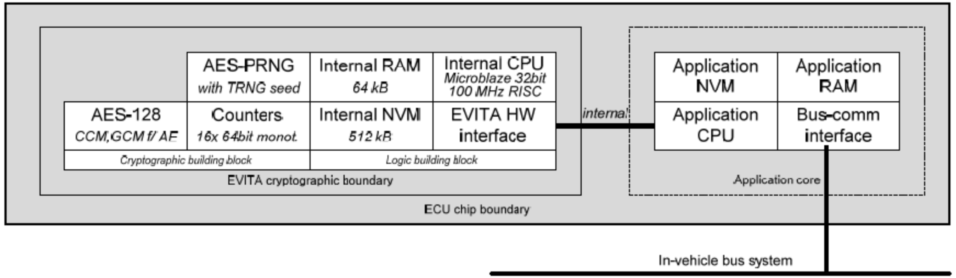
- realms / enclaves isolated by the hardware
- examples:
  - Intel TXT
  - ARM Trustzone
  - Intel SGX

virtualization solutions

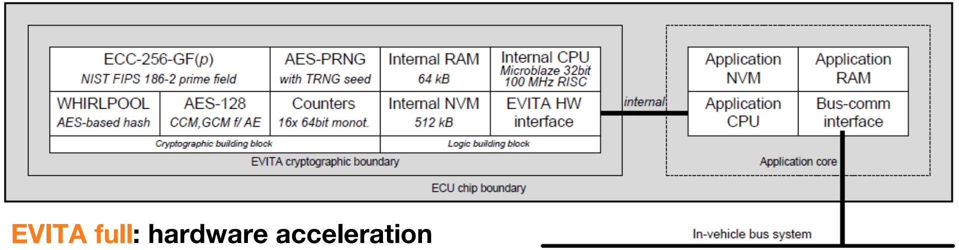
- virtualization (full or lightweight) guarantees isolation
- examples:
  - hypervisors (Xen, KVM)
  - containers (LXC)
  - unikernels



SHE, EVITA light: RAM, ROM, non-volatile memory, hardware encryption engine



EVITA medium: secure CPU (aymmetric crypto)

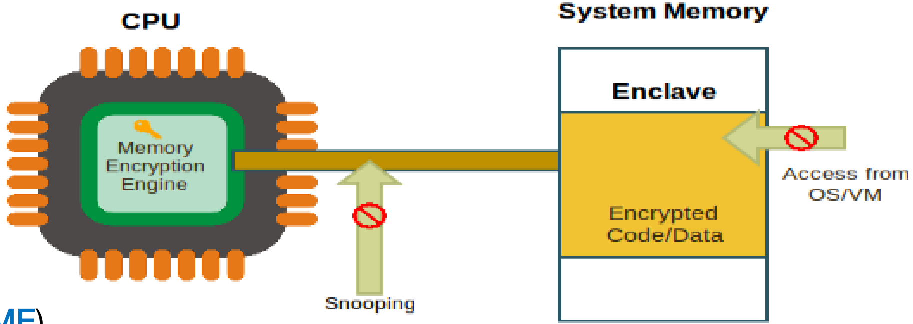


EVITA full: hardware acceleration for time-critical applications

Source: EVITA project

Intel SGX

Enclave: secure run-time environment isolated from external access



Memory protection

- Only CPU is trusted
- Multi-threaded execution
- New hardware instructions
  - Enclave creation (ECREATE)
  - Adding pages (EADD), sealing
  - Enclave mode call gate (EENTER, EEXIT, ERESUME)
- Enclave Page Cache (EPC):

  Physical memory region to store pages, transparently encrypted, integrity–protected
- SGXv2:

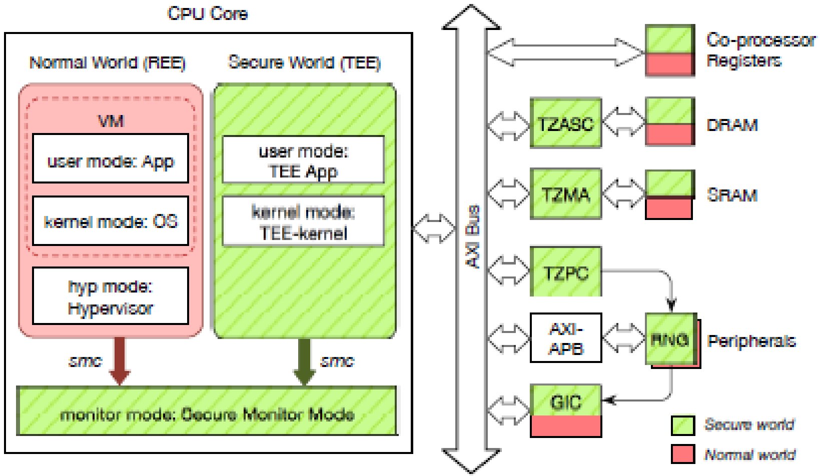
  Dynamic memory allocation, EPC permission change

Attestation

- CPU-based attestation:
  - On-demand generation of reports (EREPORT)
  - Verification of report integrity
- Quoting enclave for remote attestation

ARM Trustzone, AMD SEV

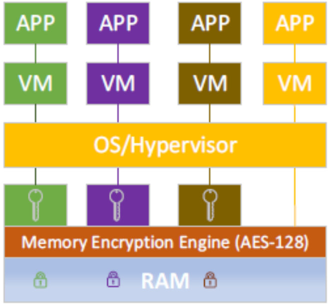
ARM TrustZone



- partition of resources in two worlds:
  - Normal world
  - Secure World
- Secure Monitor between worlds

Source: Zhichao Hua, Jinyu Gu, Yubin Xia, Haibo Chen, Binyu Zang, Haibing Guan. vTZ: Virtualizing ARM TrustZone, *USENIX Security Symposium*, 2017.

AMD SEV



encryption of VM memory image

confidentiality but not integrity protection

comparisons

	Isolation	Attestation	Sealing	Dynamic RoT	Code Confidentiality	Side-Channel Resistance[1]	Memory Protection[2]	Lightweight	Coprocessor	HW-Only TCB	Preemption	Dynamic Layout	Upgradeable TCB	Backwards Compatibility	Open-Source	Academic	Target ISA
AEGIS [46]	●	●	●	●	●	○	●	○	○	●	●	●	○	●	○	●	–
TPM [47]	○	●	●	○	●	–	◐	○	●	●	–	–	○	●	○	○	–
TXT [22]	●	●	●	●	●	●	◐	○	●	●	○	●	○	●	○	○	x86_64
TrustZone [1]	●	○	○	●	○	○	○	○	○	●	●	●	○	●	○	○	ARM
Bastion [9]	●	○	●	●	●	○	●	○	○	○	●	●	●	●	○	●	UltraSPARC
SMART [14]	○	●	○	●	○	○	○	●	○	○	–	–	○	●	○	●	AVR/MSP430
Sancus [39]	●	●	○	●	○	○	○	●	○	●	○	○	○	◐	●	●	MSP430
Soteria [21]	●	●	○	●	●	○	○	●	○	●	○	○	○	◐	●	●	MSP430
SecureBlue++ [49]	●	○	●	●	●	○	●	○	○	●	●	●	○	●	○	○	POWER
SGX [35]	●	●	●	●	●	○	●	○	○	○	●	●	●	●	○	○	x86_64
Iso-X [15]	●	●	○	●	○	○	●	○	○	○	●	●	●	●	○	●	OpenRISC
TrustLite [28]	●	●	○	○	○	○	○	●	○	○	●	●	●	●	○	●	Siskiyou Peak
TyTAN [8]	●	●	●	●	○	○	○	●	○	○	●	●	●	●	○	●	Siskiyou Peak
Sanctum [12]	●	●	●	●	●	●	○	○	○	○	●	●	●	●	●	●	RISC-V

● = Yes; ◐ = Partial; ○ = No; – = Not Applicable

[1] Resistance against software side-channel attacks targeting memory access patterns only.

[2] Protection from physical attacks, both passive (e.g., probing) and active (e.g., fault injection).

Source: P. Maene, J. Götzfried, R. de Clercq, T. Müller, F. Freiling and I. Verbauwhede. Hardware-Based Trusted Computing Architectures for Isolation and Attestation. *IEEE Transactions on Computers*, vol. 67, no. 3, pp. 361-374, March 2018.

comparisons

Source: J. Köhler and H. Förster. Trusted Execution Environments in Vehicles for Secure Driver Assistance Systems, 2017.

	Functionality												Security properties								Cost		
	Isolated execution of cryptographic operations	– Symmetric cryptography (SW)	– Asymmetric cryptography (SW)	– Symmetric cryptography (HW)	– Asymmetric cryptography (HW)	**Isolated support functionality**	– RNG with secured entropy source	– Monotonic counter	**Isolated execution of arbitrary software**	**Isolated memory**	**Secure Boot**	**Authenticated Boot**	**Protection against physical attacks**	– High degree of integration	– Side-channel attack resistance	– Hardware binding	**Protection against non-physical attacks**	– TCB realized in hardware	– TCB realized in hard- and software	– TCB realized in software	**Relative financial cost**	**Prevalent design**	**Updateability** of the software that is executed in the TEE
Hardware Security Modules																							
– SHE [4]				x			x			(x)	x			x	(x)	x		x			low	SoC / IC	none
– EVITA light [5]				x			x			(x)	(x)	(x)		x	(x)	x		x			low	SoC / IC	none
– EVITA medium [5]		x	x	x			x	x		x	x	x		x	(x)	x		x			low	SoC / IC	(SW Update)
– EVITA full [5]		x	x	x	x		x	x		x	x	x		x	(x)	x		x			medium	SoC / IC	(SW Update)
– TPM 1.2 [6]				x	x		x	x		x	x	x			(x)	(x)		x			high	EM	none
– TPM 2.0 [7]		(x)	(x)	x	x		x	x		x	x	x			(x)	(x)		x			high	EM	none
– Smartcard [8]		(x)	(x)				x			x					(x)	(x)		x			low	EM	(SW Update)
CPU security extensions																							
– ARM TrustZone [9]		x	x	(x)	(x)		x	x	x	x	x	x		(x)	(x)	(x)			x		low-high	SoC	(SW Update)
– Intel TXT [10]		x	x	(x)	(x)		x	x	x	x	x	x		(x)	(x)	(x)			x			SoC	(SW Update)
Virtualization solutions																							
– Hypervisor (e.g., Xen [11])		x	x				(x)		x	x	(x)	(x)								x	low-high	Software	SW Update
– Container (e.g., LXC [13])		x	x				(x)		x	x										x		Software	SW Update

Legend: x = supported, (x) = optionally supported, not supported; SoC: System-on-chip, IC: Integrated circuit, EM: Extension module

V2X isolation and resilience

- ECUs as virtualized execution environments
- distributed computations

goals : framework for isolation and resilience for next-generation critical vehicular functions (ECUs)

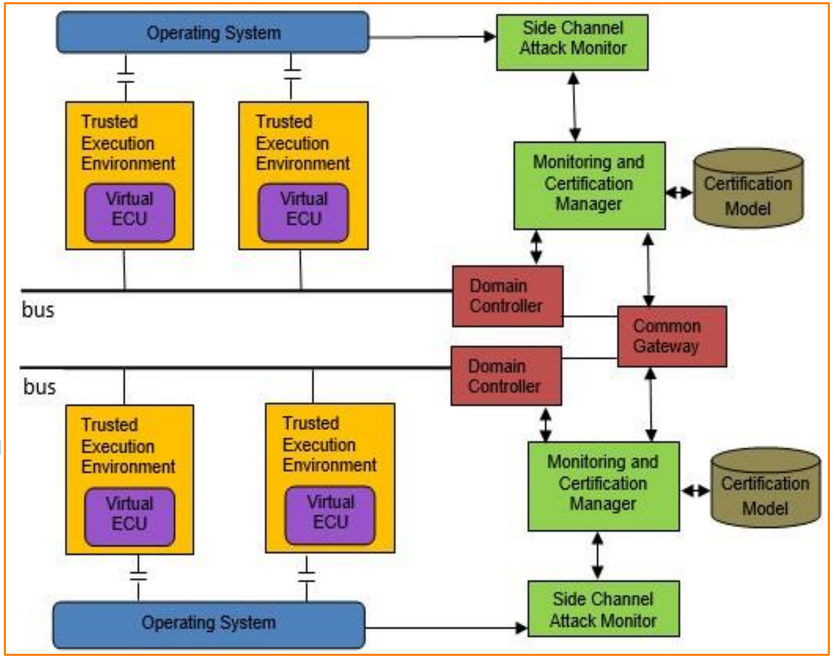
challenges :

- **ECU isolation :** trusted execution
- **resilience :** certificate-based anomaly detection
- **side-channels :** hardware performance counters
- **interoperability :** AUTOSAR framework

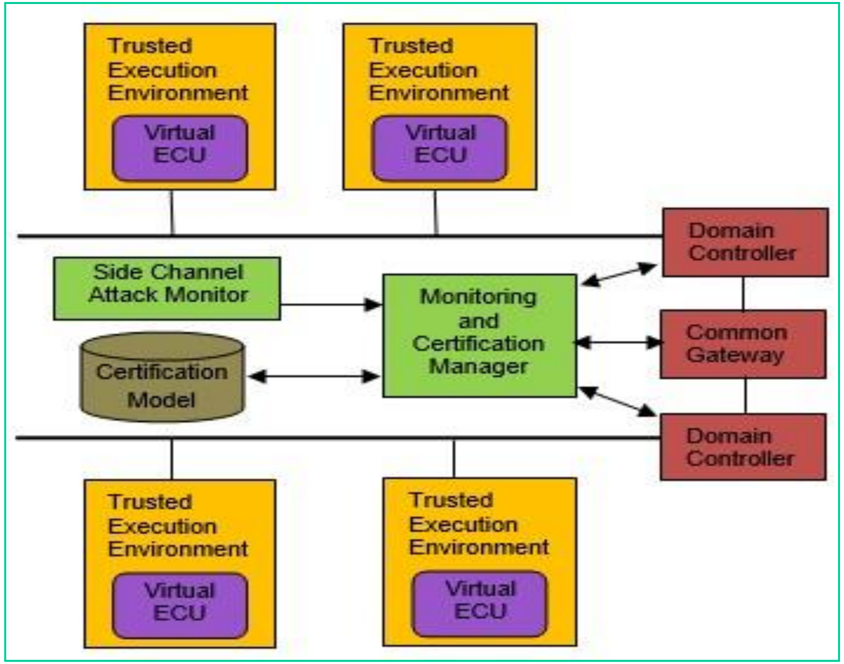
results :

- **survey :** vehicular isolation architecture, threats, mitigation
- **isolation & resilience framework : FFIVR with PoC**

  [VEHICULAR 2020]



FIIVIR: a Framework for Improving In-Vehicle Isolation and Resilience



Trusted Execution Environment (TEE)

secure isolated execution environments for ECUs

Monitoring and Certification Manager (MCM)

real-time anomaly detection of in-vehicle network

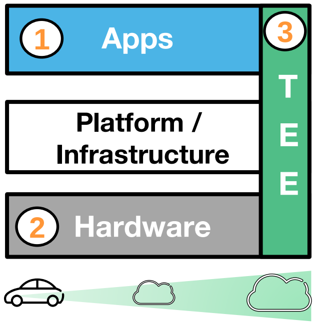
Side Channel Attack Monitor

run-time detection of side-channel attacks

Outline

- (Beyond) 5G vehicular isolation & trust
- The TEE approach
- TEE architectures
  - Intel SGX and other TEEs
  - Isolation and resilience framework for V2X
- New directions
  - Confidential computing
  - Decentralized protocols
  - Integration with ML

multi-dimensional heterogeneity challenges



1 Application heterogeneity

- **Multiple distributed applications and security requirements**

2 Hardware heterogeneity

- **Multiple execution environments fragmented across platforms**

Execution environments (EEs) : transparency, security, interoperability limitations

3 Security heterogeneity

- **A non-uniform level of trust**
- **Single-TEE industrial technologies**

SGX — AMD SEV — arm TRUSTZONE

- Provide strong security
- Have also security flaws

CONFIDENTIAL COMPUTING CONSORTIUM

- **Multi-TEE "softwarized" technologies are highly promising**
  - Lift hardware barriers

    Redhat Enarx — Microsoft OpenEnclave — Google Asylo

  - Extend also to the edge

    Microsoft Graviton

Transparency

- Isolating enclave sensitive state
- Enclave size limitations

Security

- Many side channel attacks

orange **Side-channels mitigation for decentralized clouds**

Interoperability

- Vendor lock-in
- OS functionality requirements

orange **H2020** SUPER CLOUD **Intel SGX for multi-clouds**

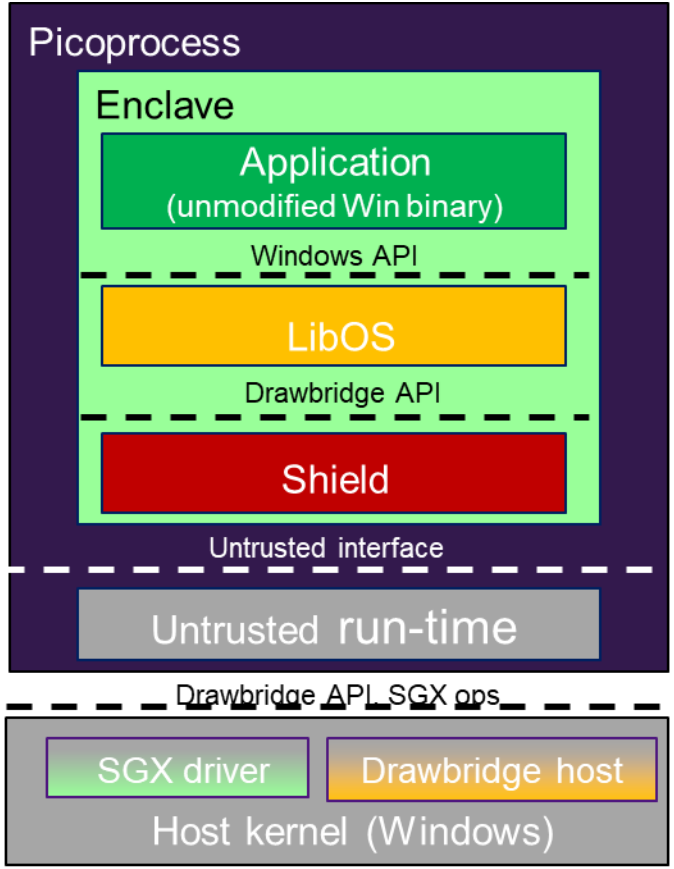
Interoperability is still lacking but starting

legacy compatibility

Haven: private execution of unmodified binaries, mutual host-guest distrust

Sandboxing:
host vs. malicious guest

- **Pico-process:**
  secure isolation container
- **Drawbridge LibOS:**
  - Narrow set of OS services
  - Virtual memory, threading, I/O
- Support unmodified Windows binaries



legacy compatibility

Haven: private execution of unmodified binaries, mutual host-guest distrust

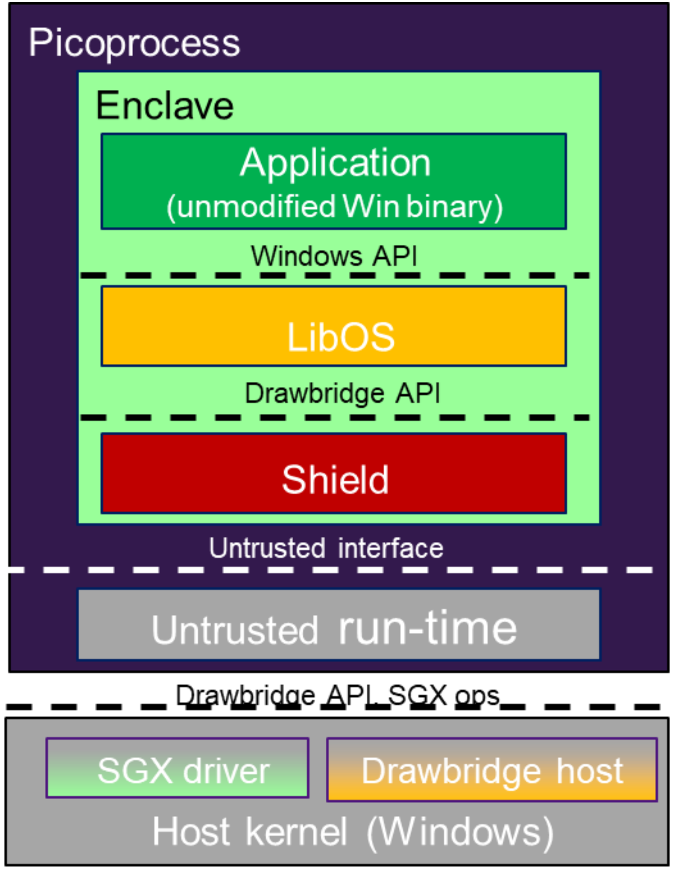
Shielded execution: applications vs. from untrusted host

Contain untrusted host OS

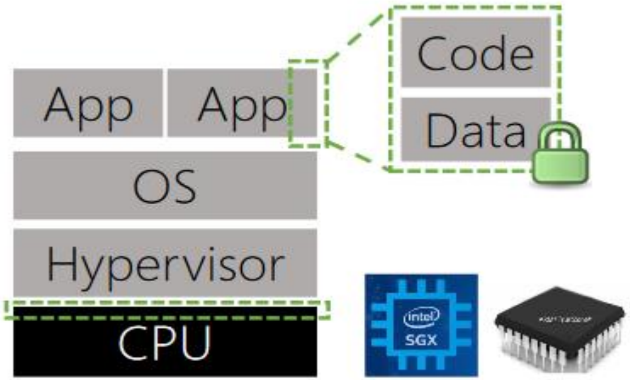
- **LibOS:** reduce attack surface
- **Shield:** reduce interface
  - Validate untrusted inputs
  - Encrypt / integrity protect private data
  - Private scheduler

Unmodified binary support

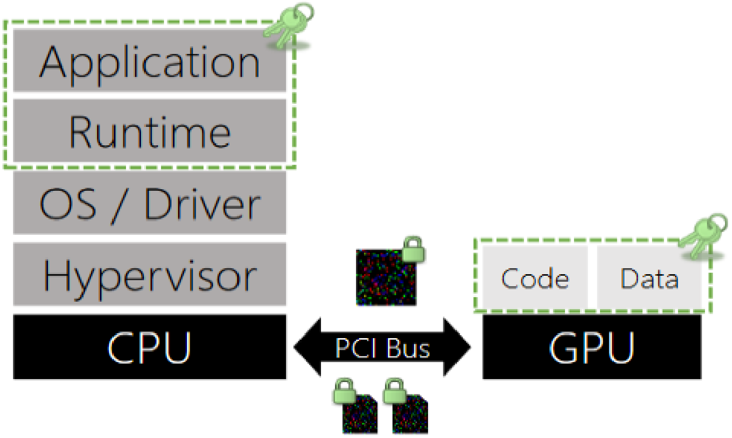
- **Exceptions:**
  - Emulate instruction behavior
  - Page faults exposed to host OS



co-processors

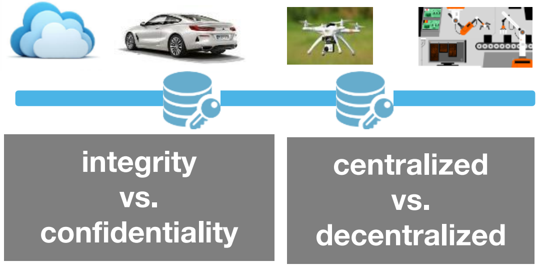


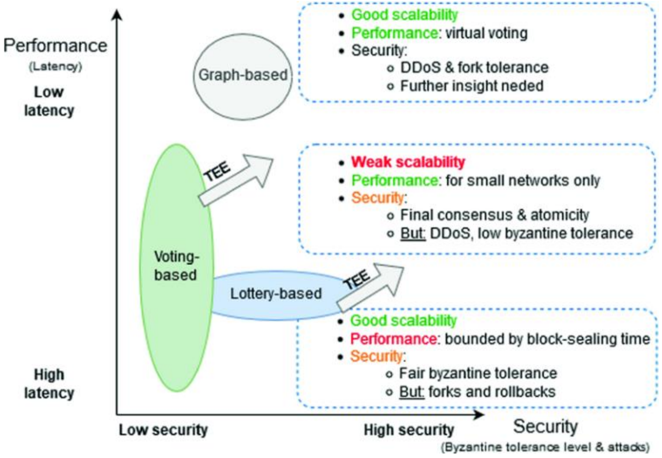
- TEE on GPUs: Graviton
- Confidentiality and integrity of computation and data
- Secure GPU/CPU interface



decentralized protection of data

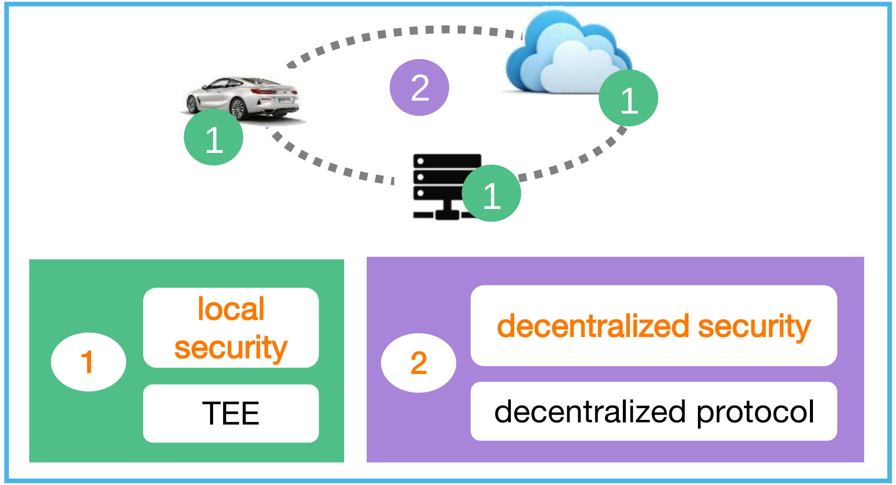
Future large-scale distributed applications have multiple data protection challenges





Source: P. Boos, M. Lacoste **Networks of Trusted Execution Environments for Data Protection in Cooperative Vehicular Systems,** *Advances in Intelligent Systems and Computing,* 2020, Springer.

Network of TEEs architectures combine strong local security with decentralized security

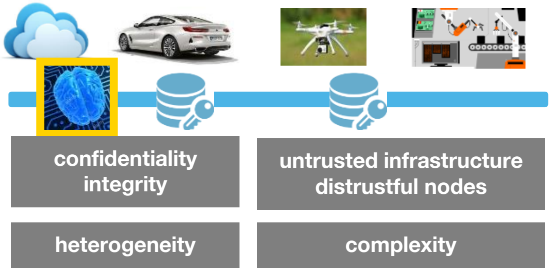


Some remaining challenges

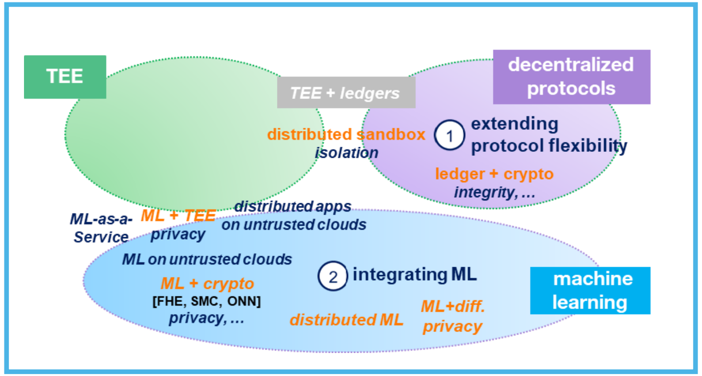
- **Guarantee security of coupling between TEE and protocol**
- **Reach flexibility in protection architecture**

extending decentralized flexibility to privacy-preserving Artificial Intelligence

Apps perform distributed computations for automated predictions over private data



A rich landscape of hybrid solutions



Extend flexibility to integrate predictions

- TEE is part of wider set of **privacy-preserving technologies**

TEE	Decentralized protocols	Cryptographic protocols	Differential privacy ...

Federated Machine Learning → **privacy-preserving AI**

- **P2P hybrid solutions** enable to find interesting trade-offs

Beyond P2P solutions…

- **Towards a unified and open reference architecture to orchestrate the different enablers**
- **The open source approach is promising to federate ecosystems**

Some remaining challenges

- the previous heterogeneity challenges are magnified
- going towards a fully **zero-trust model**

conclusion

- Vehicular systems: acute security & safety challenges for distributed isolation & trust
- Trusted computing approaches: strong guarantees by shielding applications
- Some challenges ahead:
  - Distribution: Device, edge, cloud continuum

    Seamless mobility?
  - Composition of security technologies
  - Distributed/federated machine learning
  - Heterogeneous processor architectures
  - Latency requirements
  - Side-channels
  - Chains of Trust and certification

Thank you

Marc Lacoste
Orange Innovation
Senior Research Scientist
marc.lacoste@orange.com