KU LEUVEN

Trusted Execution with Real-Time and Availability Guarantees for Mixed-Criticality Embedded Systems

QA&TEST Safety and Security

Fritz Alder, Jo Van Bulck, Frank Piessens, Jan Tobias Mühlberg
imec-DistriNet, KU Leuven, Belgium

March 18, 2021

What?

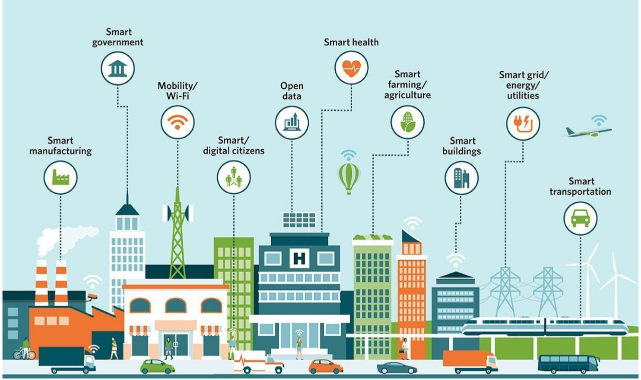
Trusted Computing / Trusted Execution. . .

- Strong integrity protection and isolation for software components
- Software attestation: cryptographically bind a software to the executing hardware
- Sealed storage: bind data to attested software

. . . for mixed-criticality systems

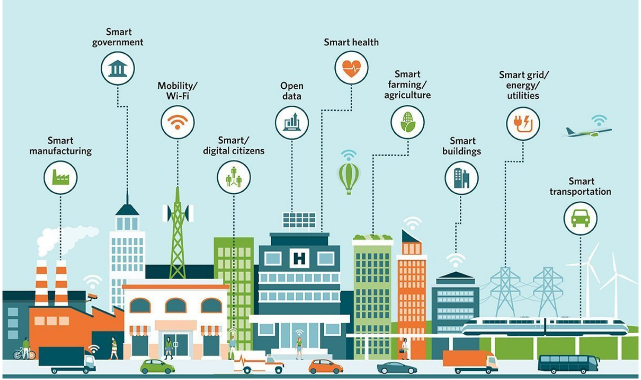
- Effective isolation of different criticalities?
- Real-time and progress guarantees?
- What are interesting use cases?

Security in Smart Environments



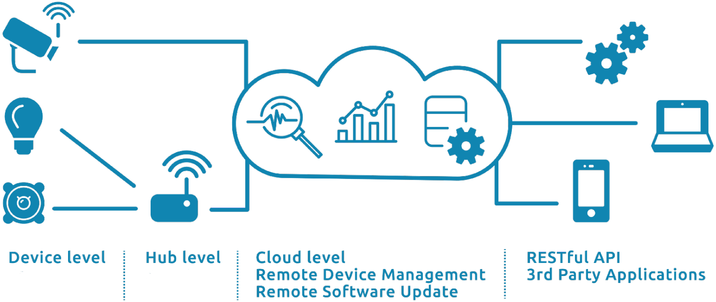
Infrastructure needs to be developed with safety and security in mind! What is critical infrastructure? What is critical code? What's the impact of failure?

Security in Smart Environments



Vulnerabilities can hide anywhere: There are 150M lines of code in a modern car. Compartmentalisation can help with managing complexity.

Security in Smart Environments



Device level

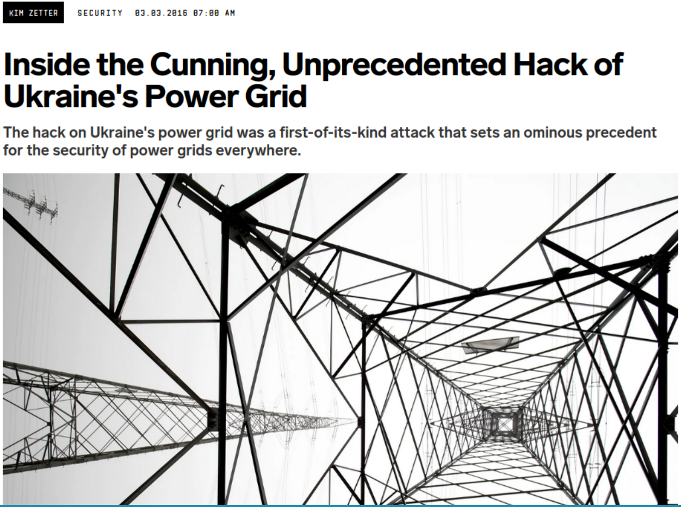
Hub level

Cloud level
Remote Device Management
Remote Software Update

RESTful API
3rd Party Applications

Understanding can be really difficult: What stake holders are involved? What are their objectives and abilities? What hardware and software is involved? Software quality? Data flows? Security requirements and guarantees?

Security in Smart Environments

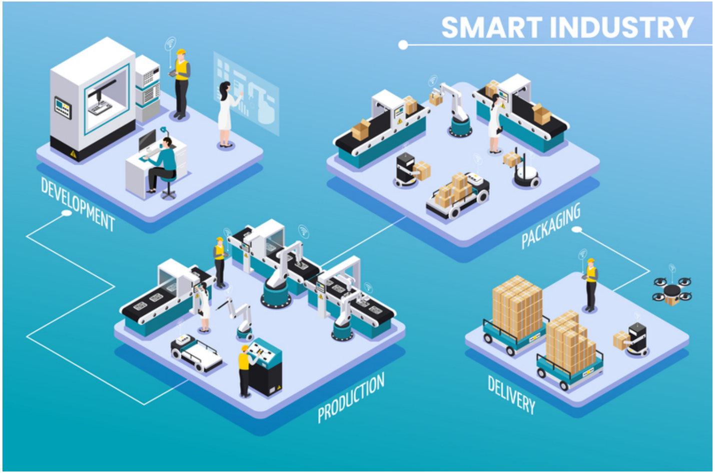


KIM ZETTER SECURITY 03.03.2016 07:00 AM

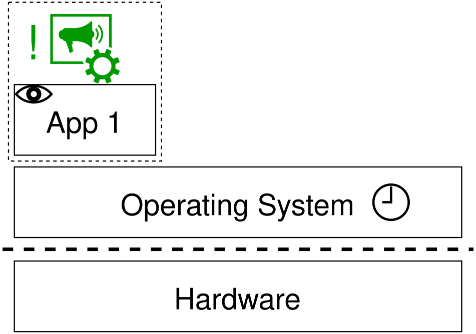
Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid

The hack on Ukraine's power grid was a first-of-its-kind attack that sets an ominous precedent for the security of power grids everywhere.

Most devices are not new. Their connectivity is new!

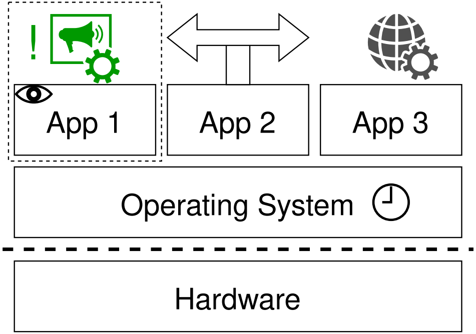


Safety-Critical Systems Overview



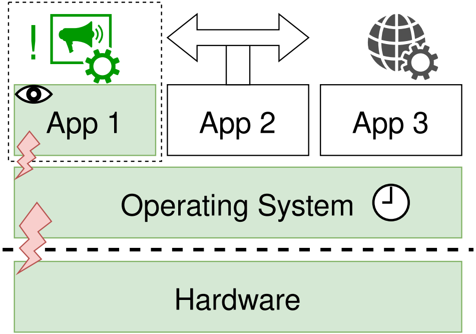
Safety-Critical System

Mixed-criticality Systems Overview



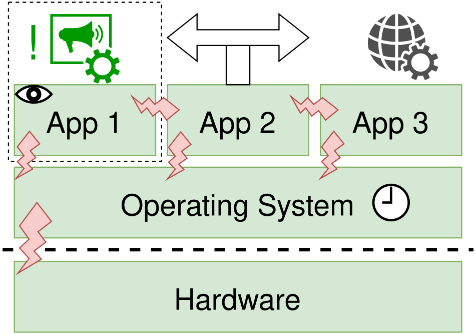
Mixed-Criticality System

Mixed-criticality Systems – Who do we want to trust?



Desired trust

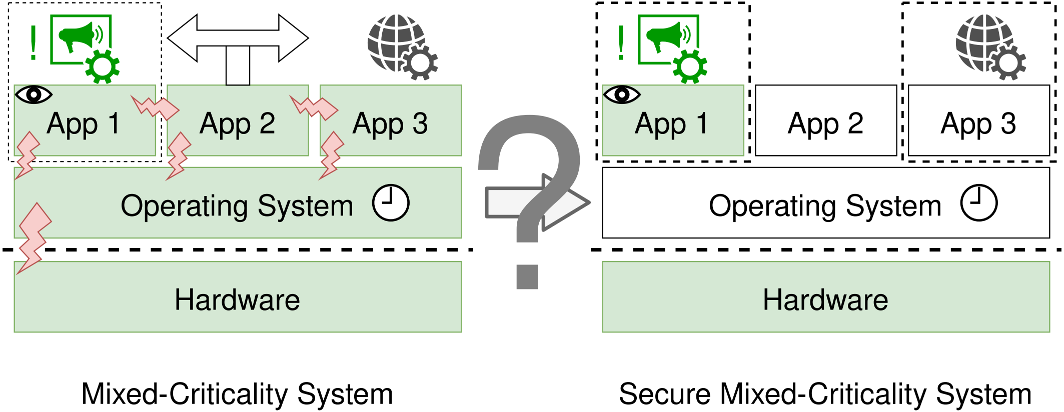
Mixed-criticality Systems – Who do we have to trust?



Actual trust for availability

- Monopolizing a system resource or stalling the CPU is often possible.
- **Hackers do not cooperate.**
- **Even postponing deadlines** can have harsh consequences.

Mixed-criticality Systems – What do we want?



Trusted Execution Environments: A castle inside the processor

Trusted Execution Environments: Only allow trusted access

Comparing Hardware-Based TEEs

	Isolation	Attestation	Sealing	Dynamic RoT	Code Confidentiality	Side-Channel Resistance	Memory Protection	Lightweight	Coprocessor	HW-Only TCB	Preemption	Dynamic Layout	Upgradeable TCB	Backwards Compatibility	Open-Source	Academic	Target ISA
AEGIS	●	●	●	●	●	○	●	○	○	●	●	●	○	●	○	●	–
TPM	○	●	●	○	●	–	◐	○	●	●	–	–	○	●	○	○	–
TXT	●	●	●	●	●	●	◐	○	●	●	○	●	○	●	○	○	x86_64
TrustZone	●	○	○	●	○	○	○	○	○	○	●	●	○	●	○	○	ARM
Bastion	●	○	●	●	●	○	●	○	○	○	●	●	●	●	○	●	UltraSPARC
SMART	○	●	○	●	○	–	○	●	○	○	–	–	○	●	○	●	AVR/MSP430
Sancus 1.0	●	●	○	●	○	●	○	●	○	●	○	○	○	●	●	●	MSP430
Soteria	●	●	○	●	●	●	○	●	○	●	○	○	○	●	●	●	MSP430
Sancus 2.0	●	●	○	●	●	●	○	●	○	●	◐	○	○	●	●	●	MSP430
SecureBlue++	●	○	●	●	●	○	●	○	○	●	●	●	○	●	○	○	POWER
SGX	●	●	●	●	●	○	●	○	○	○	●	●	●	●	○	○	x86_64
Iso-X	●	●	○	●	○	○	●	○	○	○	●	●	●	●	○	●	OpenRISC
TrustLite	●	●	○	○	○	●	○	●	○	○	●	●	●	●	○	●	Siskiyou Peak
TyTAN	●	●	●	●	○	●	○	●	○	○	●	●	●	●	○	●	Siskiyou Peak
Sanctum	●	●	●	●	●	●	○	○	○	○	●	●	●	●	◐	●	RISC-V

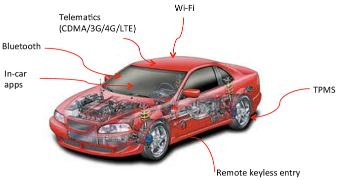
● = Yes; ◐ = Partial; ○ = No; – = Not Applicable

Adapted from “Hardware-Based Trusted Computing Architectures for Isolation and Attestation ”, Maene et al., IEEE Transactions on Computers, 2017.

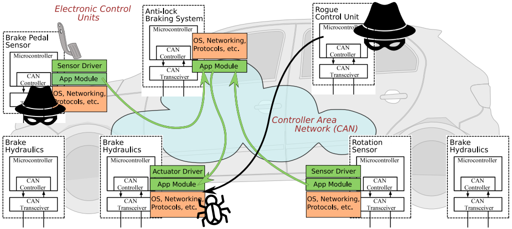
Secure Automotive Computing

Modern cars can be hacked!

- Network of more than 50 ECUs
- Multiple communication networks
- Remote entry points
- Limited built-in security mechanisms



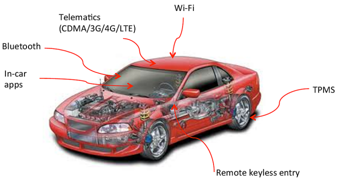
Miller & Valasek, “Remote exploitation of an unaltered passenger vehicle”, 2015



Secure Automotive Computing with Sancus

Modern cars can be hacked!

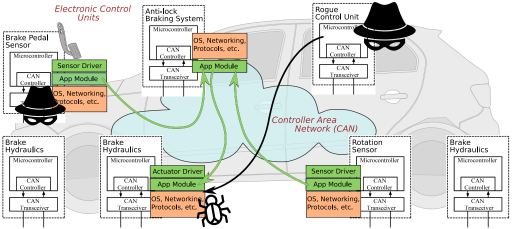
- Network of more than 50 ECUs
- Multiple communication networks
- Remote entry points
- Limited built-in security mechanisms



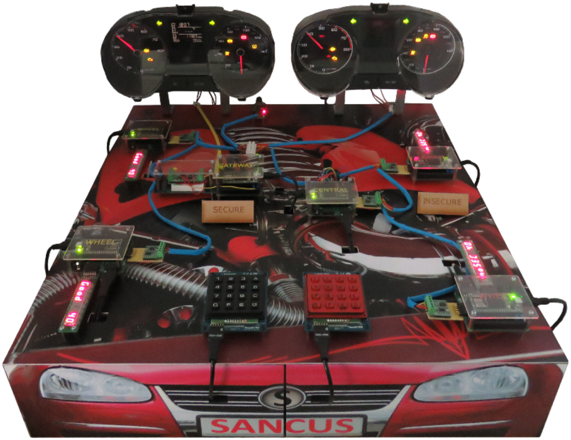
Miller & Valasek, "Remote exploitation of an unaltered passenger vehicle", 2015

Sancus brings strong security for embedded control systems:

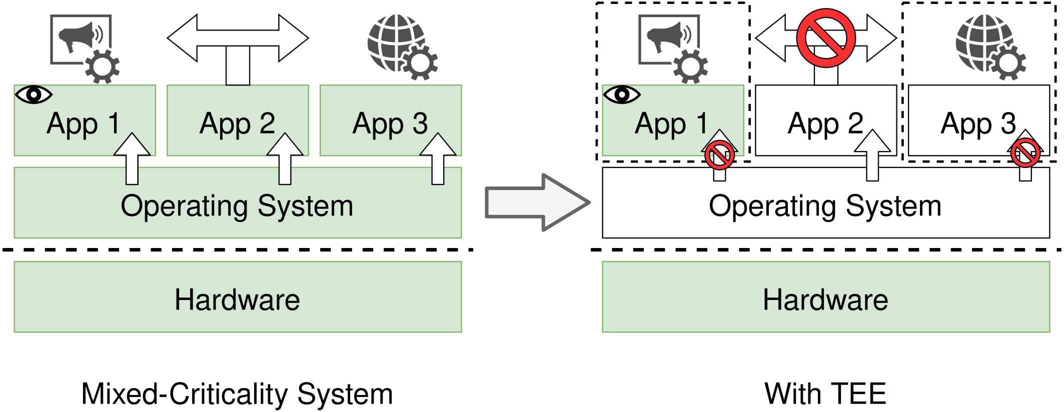
- Message authentication
- Trusted Computing: software component isolation and cryptography
- Strong software security
- Applicable in automotive, ICS, IoT. . .



Secure Automotive Computing with Sancus



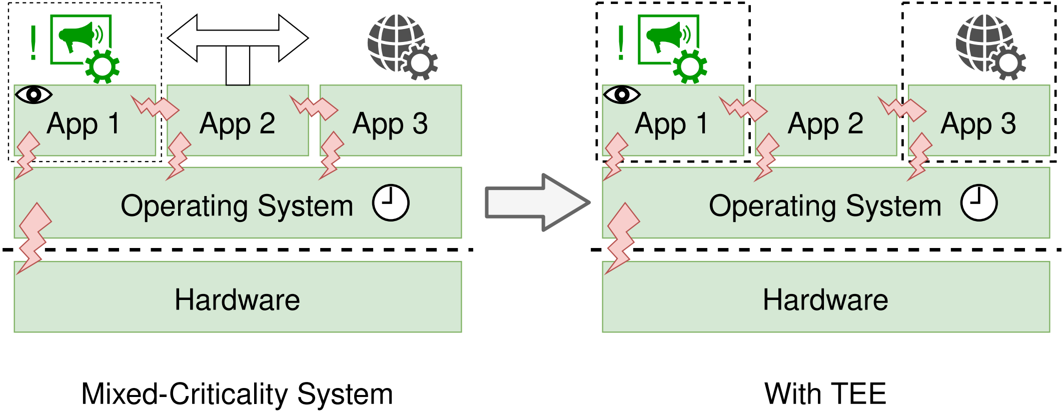
Trusted Execution: Reducing the Attack Surface



Trust for confidentiality and integrity

Authentic Execution of Event-Driven Applications

Trusted Execution: Reducing the Attack Surface



Trust for availability

Requirements for Dependable Mixed-Criticality with TEEs

Requirements for Dependable Mixed-Criticality with TEEs

We want security:

- Spacial isolation, memory curtaining, enclaves
- Enclave attestation
- Dynamic deployment

Requirements for Dependable Mixed-Criticality with TEEs

We want security:

- Spacial isolation, memory curtaining, enclaves
- Enclave attestation
- Dynamic deployment

We also want availability:

- Preemption
- Bounded atomicity
- Protected scheduler with dynamic policies

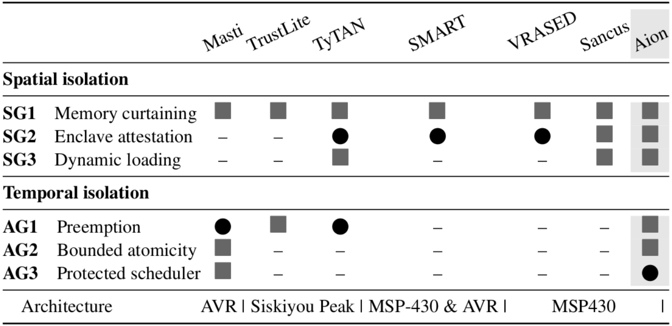
Requirements for Dependable Mixed-Criticality with TEEs

We want security:

- Spacial isolation, memory curtaining, enclaves
- Enclave attestation
- Dynamic deployment

We also want availability:

- Preemption
- Bounded atomicity
- Protected scheduler with dynamic policies



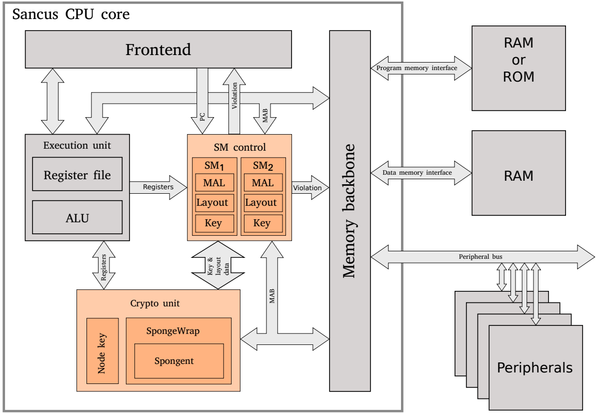
		Masti	TrustLite	TyTAN	SMART	VRASED	Sancus	Aion
Spatial isolation								
SG1	Memory curtaining	■	■	■	■	■	■	■
SG2	Enclave attestation	–	–	●	●	●	■	■
SG3	Dynamic loading	–	–	■	–	–	■	■
Temporal isolation								
AG1	Preemption	●	■	●	–	–	–	■
AG2	Bounded atomicity	■	–	–	–	–	–	■
AG3	Protected scheduler	■	–	–	–	–	–	●
Architecture		AVR	Siskiyou Peak		MSP-430 & AVR	MSP430		

We want it all on a (cheap) light-weight IoT processor.

Dependable Mixed-Criticality with TEEs

Sancus as a Starting Point

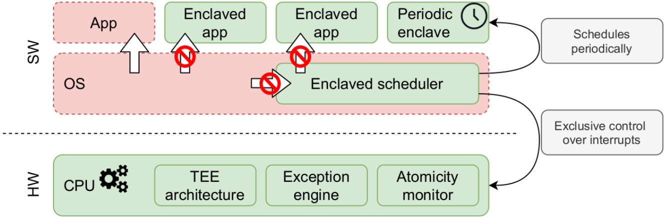
- Open-source hardware-only TEE
- Tiny footprint, low power, extends openMSP430



Dependable Mixed-Criticality with TEEs

Sancus as a Starting Point

- Open-source hardware-only TEE
- Tiny footprint, low power, extends openMSP430



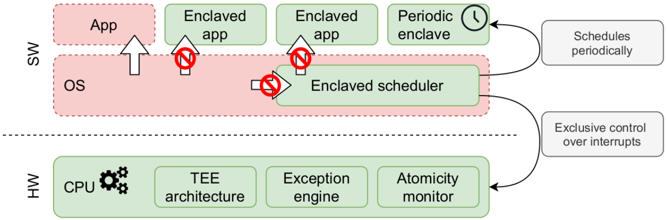
Hardware Extensions

- Exception Engine facilitates interruption of (protected) threads
- Atomicity Monitor provides control over interrupts to scheduler, guarantees bounded critical sections

Dependable Mixed-Criticality with TEEs

Sancus as a Starting Point

- Open-source hardware-only TEE
- Tiny footprint, low power, extends openMSP430



Hardware Extensions

- Exception Engine facilitates interruption of (protected) threads
- Atomicity Monitor provides control over interrupts to scheduler, guarantees bounded critical sections

Trusted Software

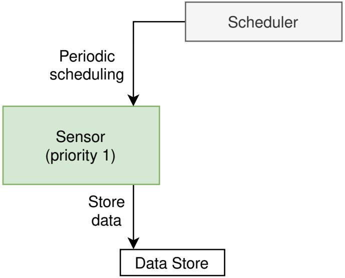
- **Protected Scheduler** controls interrupts and scheduling decisions

Results – Case Study

- We can guarantee an activation latency of 5228 cycles (291ns @ 20Mhz).
- What does this mean in practice?

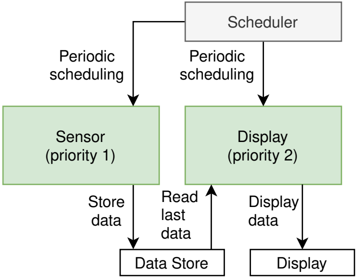
Results – Case Study

- We can guarantee an activation latency of 5228 cycles (291ns @ 20Mhz).
- What does this mean in practice?
- **High-Priority** jobs can rely on strict interrupt arrival times.



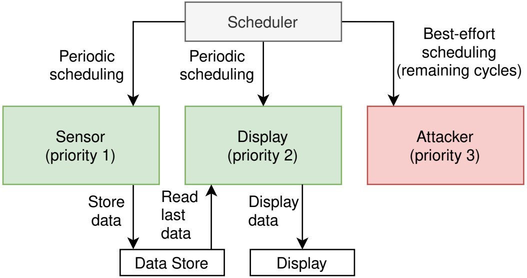
Results – Case Study

- We can guarantee an activation latency of 5228 cycles (291ns @ 20Mhz).
- What does this mean in practice?
- **High-Priority** jobs can rely on strict interrupt arrival times.
- **Low-Priority** jobs can take over secondary tasks.



Results – Case Study

- We can guarantee an activation latency of 5228 cycles (291ns @ 20Mhz).
- What does this mean in practice?
- **High-Priority** jobs can rely on strict interrupt arrival times.
- **Low-Priority** jobs can take over secondary tasks.
- **Attackers** can only obtain priority levels up to the priority of their compromised job.



Authentic Execution of Event-Driven Applications

Authentic Execution of Event-Driven Applications

"...if the application produces a physical output event (e.g., turns on an LED), then there must have been physical input events such that, when processed by the application, the output event is produced ...",

Authentic Execution of Event-Driven Applications

"...if the application produces a physical output event (e.g., turns on an LED), then there must have been physical input events such that, when processed by the application, the output event is produced ...",

Mitigates Attacks

- Network-level attacks including modification and replay
- Direct interference of a strong software-level attacker

Dependable Execution of Event-Driven Applications

Events (e.g., a button pressed) are guaranteed to be processed with deterministic deadlines and priorities, such that

"...if the application produces a physical output event (e.g., turns on an LED), then there must have been physical input events such that, when processed by the application, the output event is produced ...",

and real-time requirements are respected.

Mitigates Attacks

- Network-level attacks including modification and replay
- Direct interference of a strong software-level attacker

Dependable Execution of Event-Driven Applications

Events (e.g., a button pressed) are guaranteed to be processed with deterministic deadlines and priorities, such that

“. . . if the application produces a physical output event (e.g., turns on an LED), then there must have been physical input events such that, when processed by the application, the output event is produced . . . ”,

and real-time requirements are respected.

Mitigates Attacks

- Network-level attacks including modification and replay
- Direct interference of a strong software-level attacker
- Temporal resource monopolisation by a software-level attacker

What can we do with it?

- Secure critical sensing and control
- Share platform for components with different criticality
  - Visualisation and user feedback
  - Monitoring or intrusion detection
- Can be integrated with heterogeneous environments.

Summary

Trusted Execution Environments

- Strong application isolation and attestation: hardware-level security and taming complexity

Sancus (Try it out: **https://distrinet.cs.kuleuven.be/software/sancus/**)

- Light-weight, hardware-only, open-source TEE
- Built upon openMSP430 16-bit MCU, applications in IoT and embedded control systems
- Now with real-time and availability support

Exciting Use Cases

- Strong security and availability for control systems
- Mixed-criticality with safety functions on same platform

Image sources

- https://internetofthingsagenda.techtarget.com/definition/smart-city
- https://medium.com/connected-news/iot-foundation-what-is-an-iot-platform-c37c5e72d4a0
- https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/
- https://unsplash.com/photos/kEP-zO-w4nE
- https://www.freepik.com/macrovector
- https://unsplash.com/photos/OtbkhHNWjgc
- https://www.freepik.com/free-photo/interior-warehouse-logistic-center-have-agv-robot-arm_9316667.htm