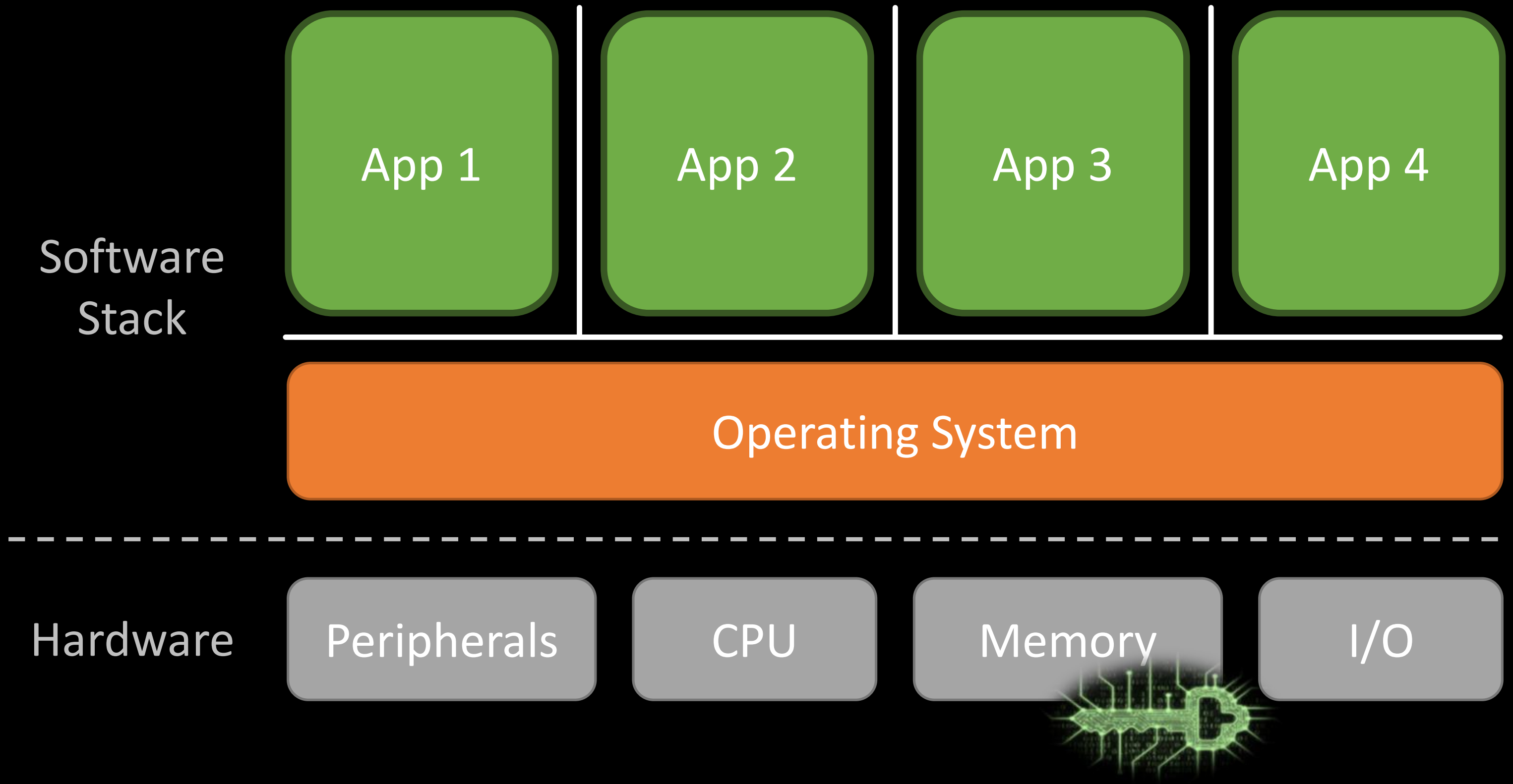
In Hardware We Trust: Enriching the World with Hardware Security

Ahmad-Reza Sadeghi

Technical University of Darmstadt

Trusted Computing Goal: Self-Contained Security



- Isolated execution
- Platform integrity
- Secure storage
- Device identification
- Device authentication capabilities

Trusted Computing Landscape

HDFI

CHERI

CODOMs

IMIX

SMART

Iso-X

Bastion

AEGIS

TyTAN

Sanctuary

TrustLite

Keystone

Sanctum

CURE

TIMBER-V

Sancus

Hardware-assisted
secure boot

Hex Five
MultiZone

Late launch/TXT

Intel SGX

ARM TrustZone

Java Card platform

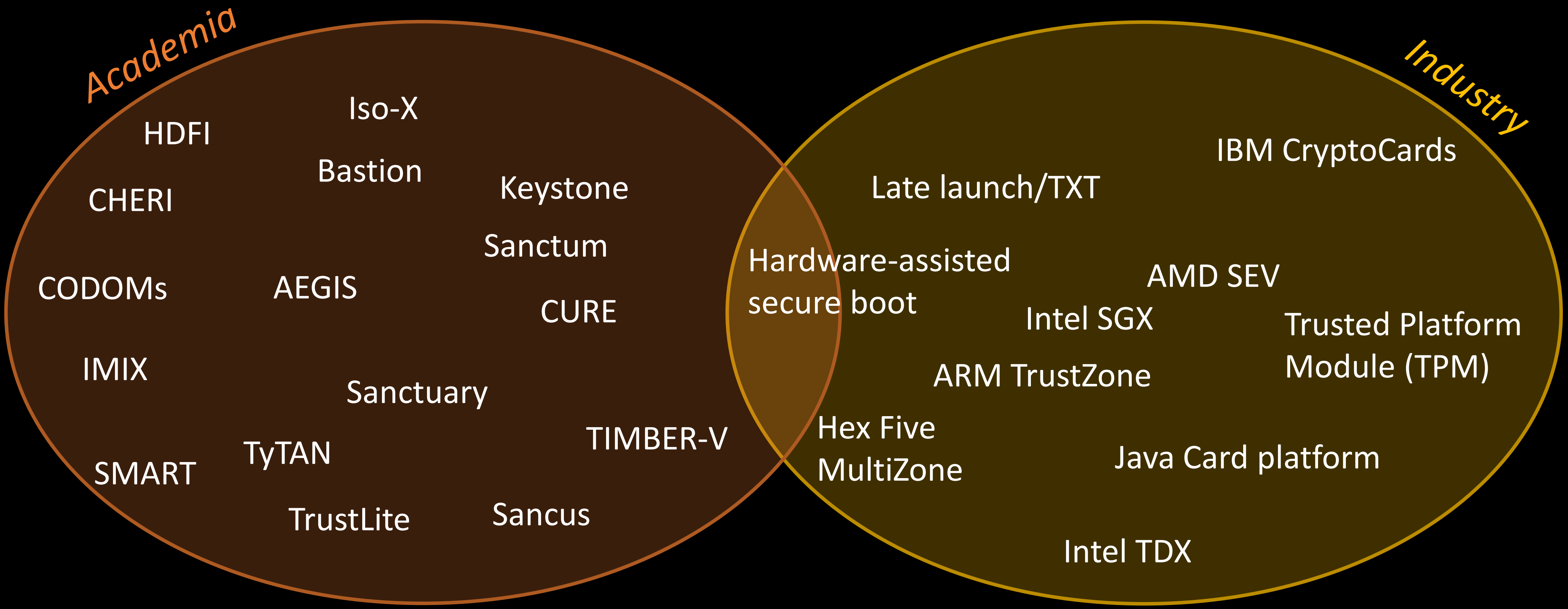
Intel TDX

IBM CryptoCards

AMD SEV

Trusted Platform
Module (TPM)

Trusted Computing Landscape



Academia

HDFI

Iso-X

Bastion

CHERI

Keystone

Sanctum

CODOMs

AEGIS

CURE

IMIX

Sanctuary

TIMBER-V

SMART

TyTAN

TrustLite

Sancus

Hardware-assisted secure boot

Industry

IBM CryptoCards

Late launch/TXT

AMD SEV

Intel SGX

Trusted Platform Module (TPM)

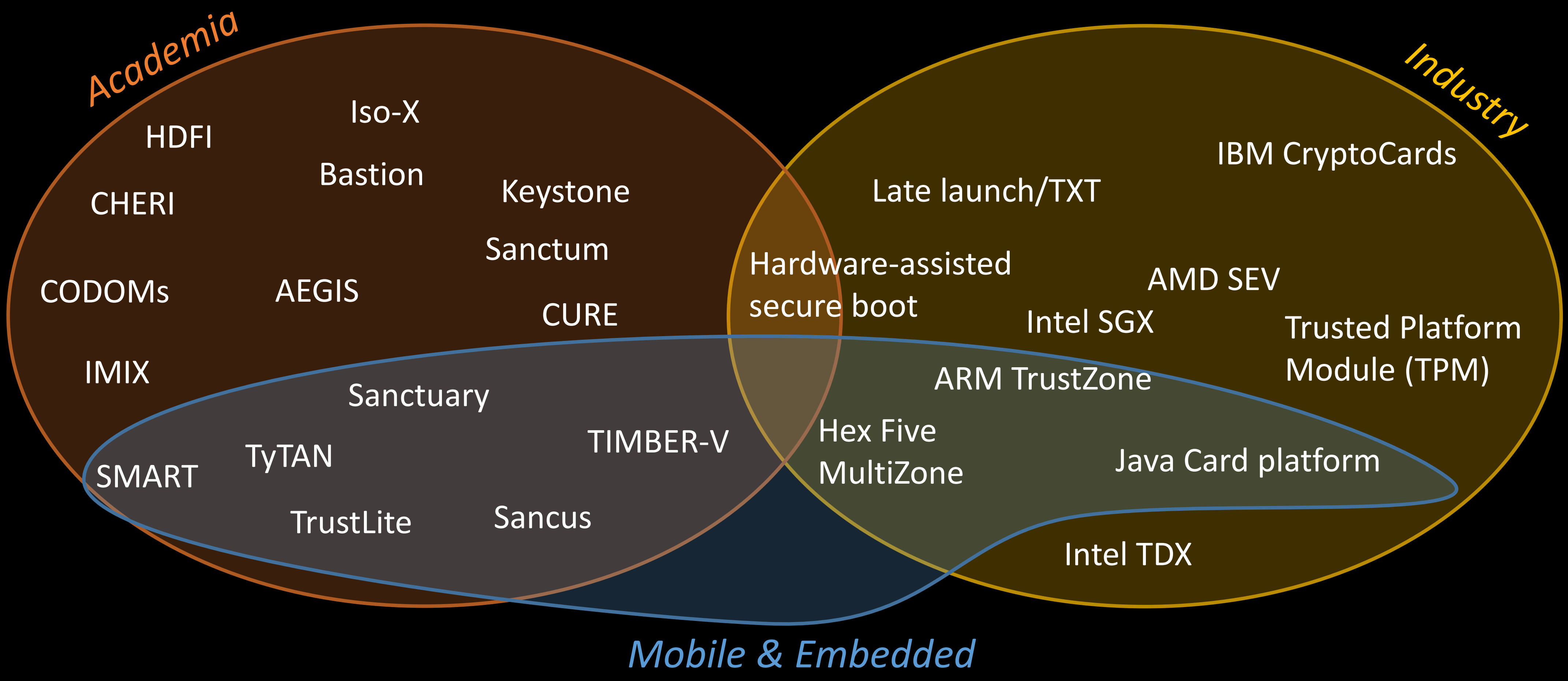
ARM TrustZone

Hex Five MultiZone

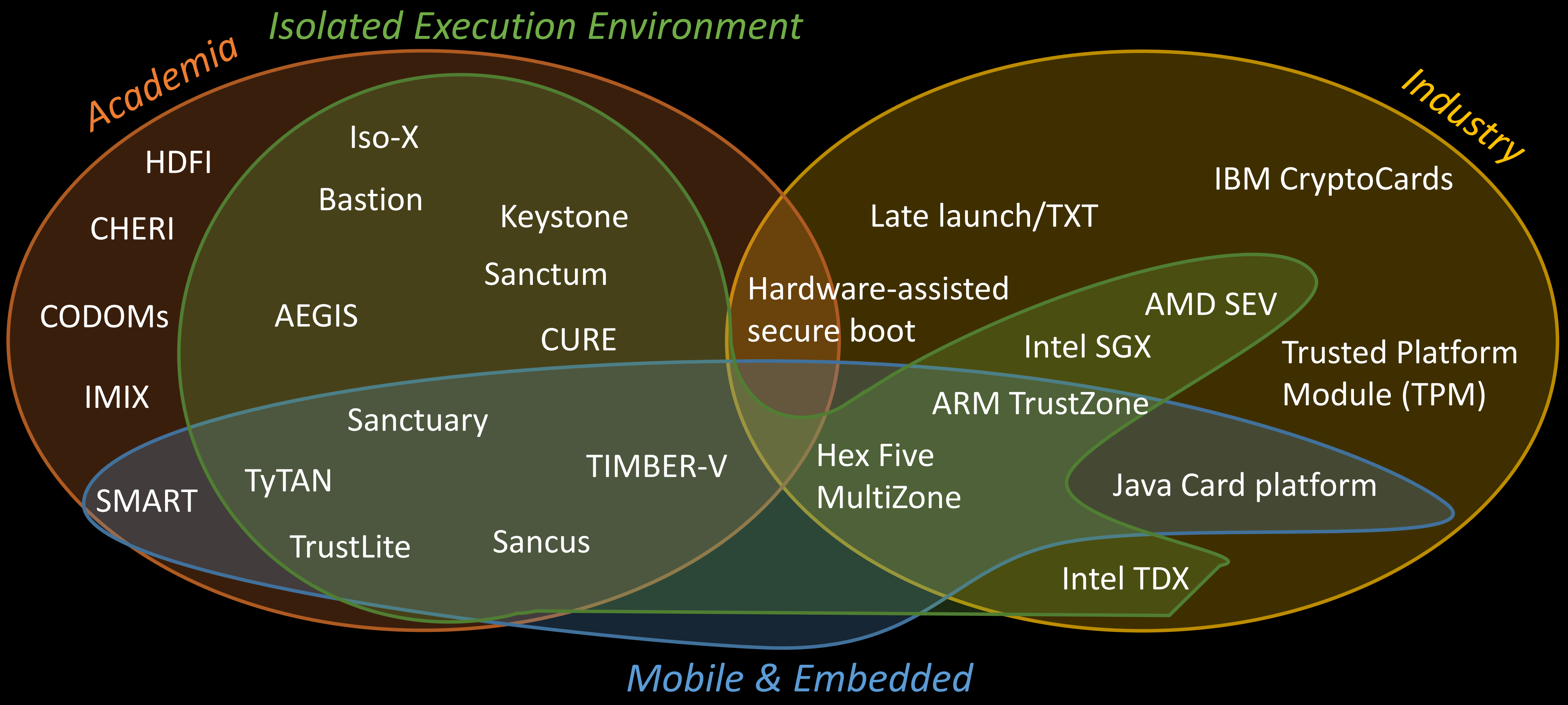
Java Card platform

Intel TDX

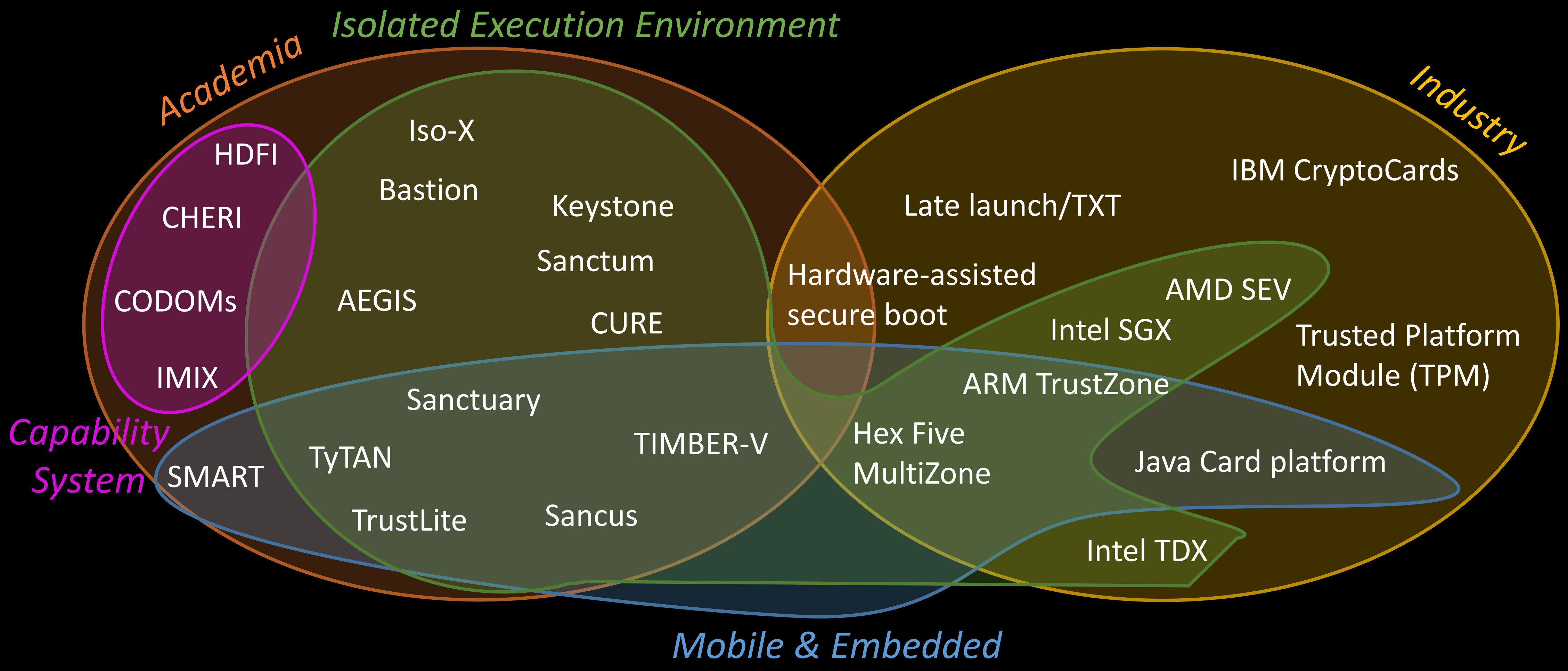
Trusted Computing Landscape



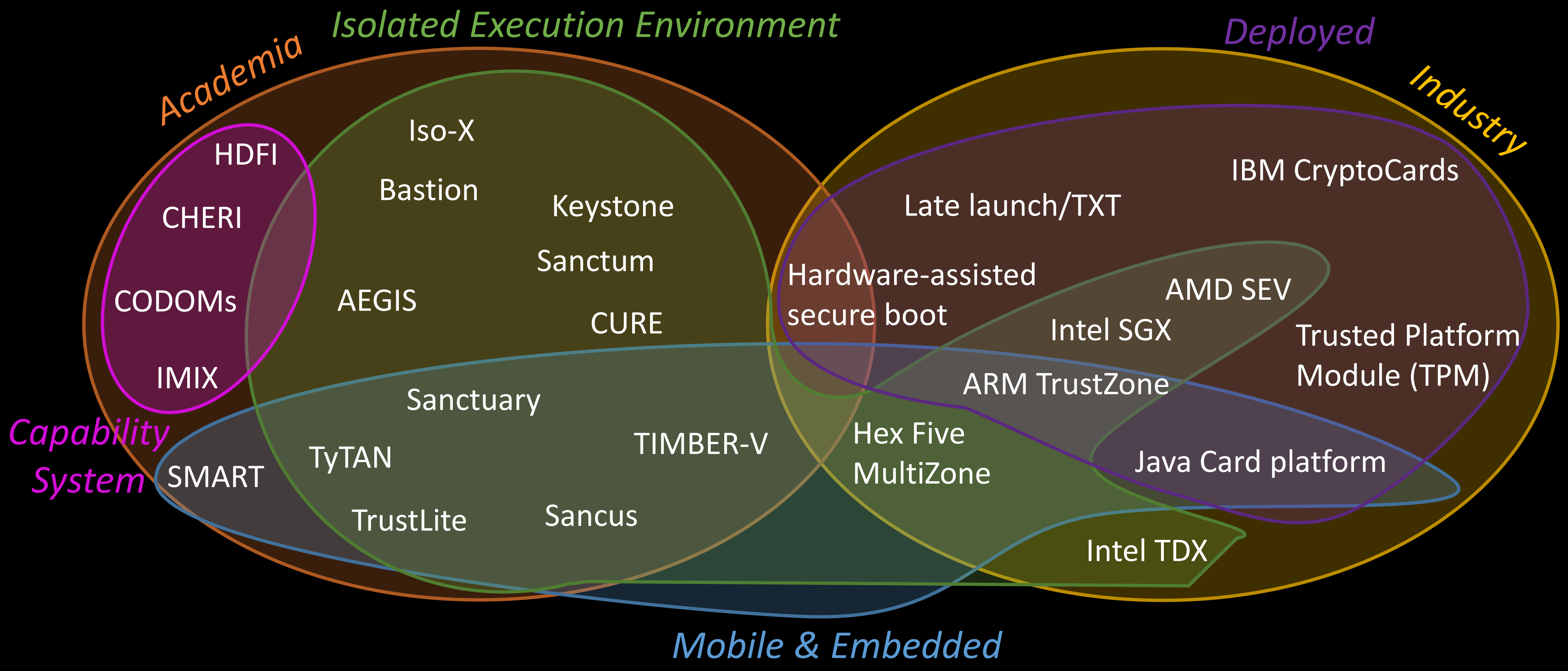
Trusted Computing Landscape



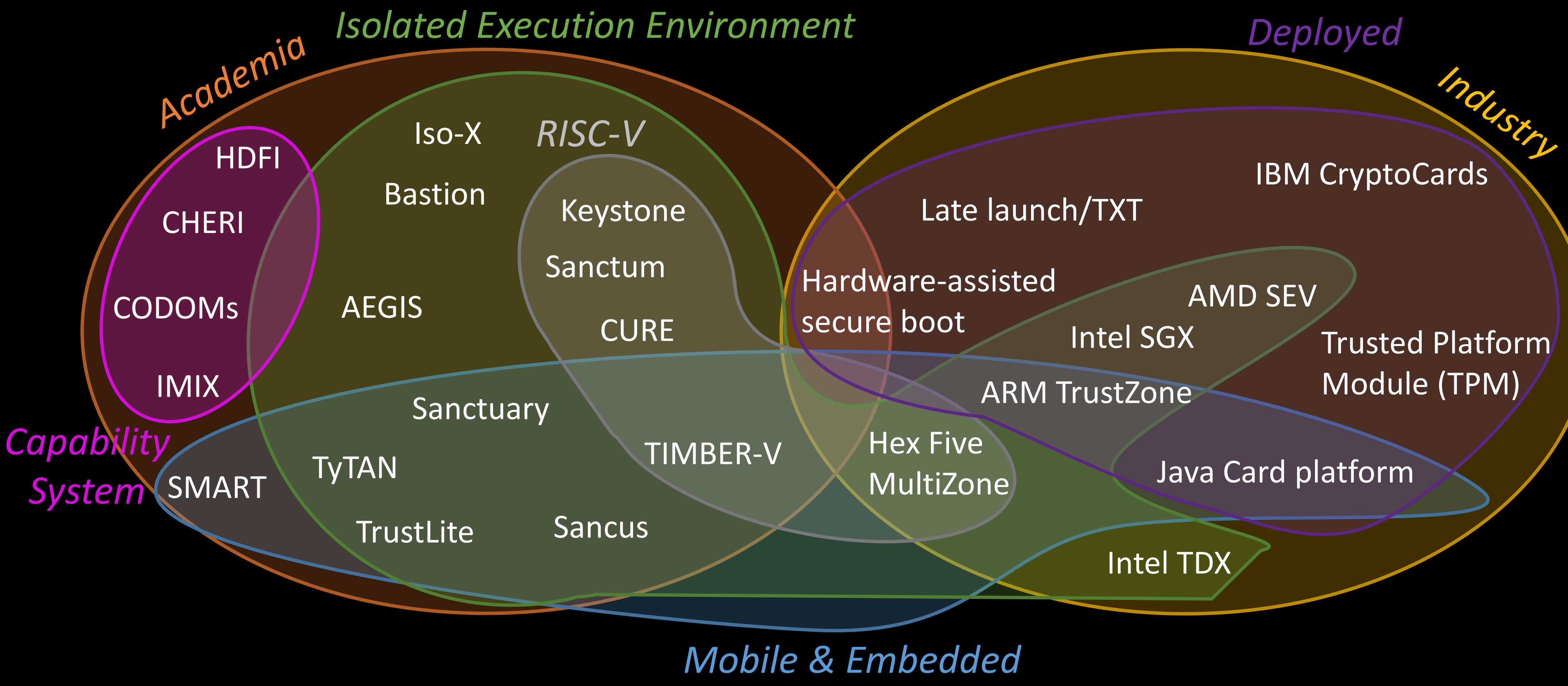
Trusted Computing Landscape



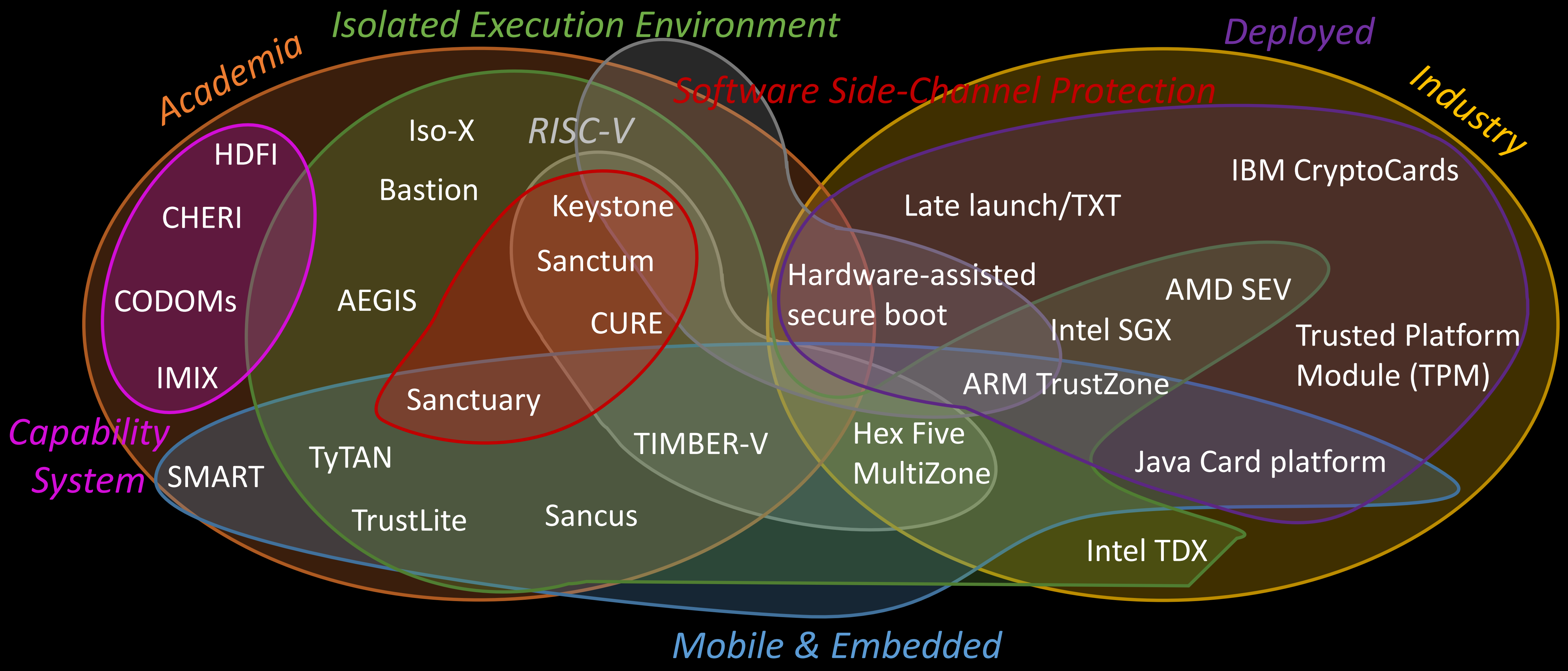
Trusted Computing Landscape



Trusted Computing Landscape



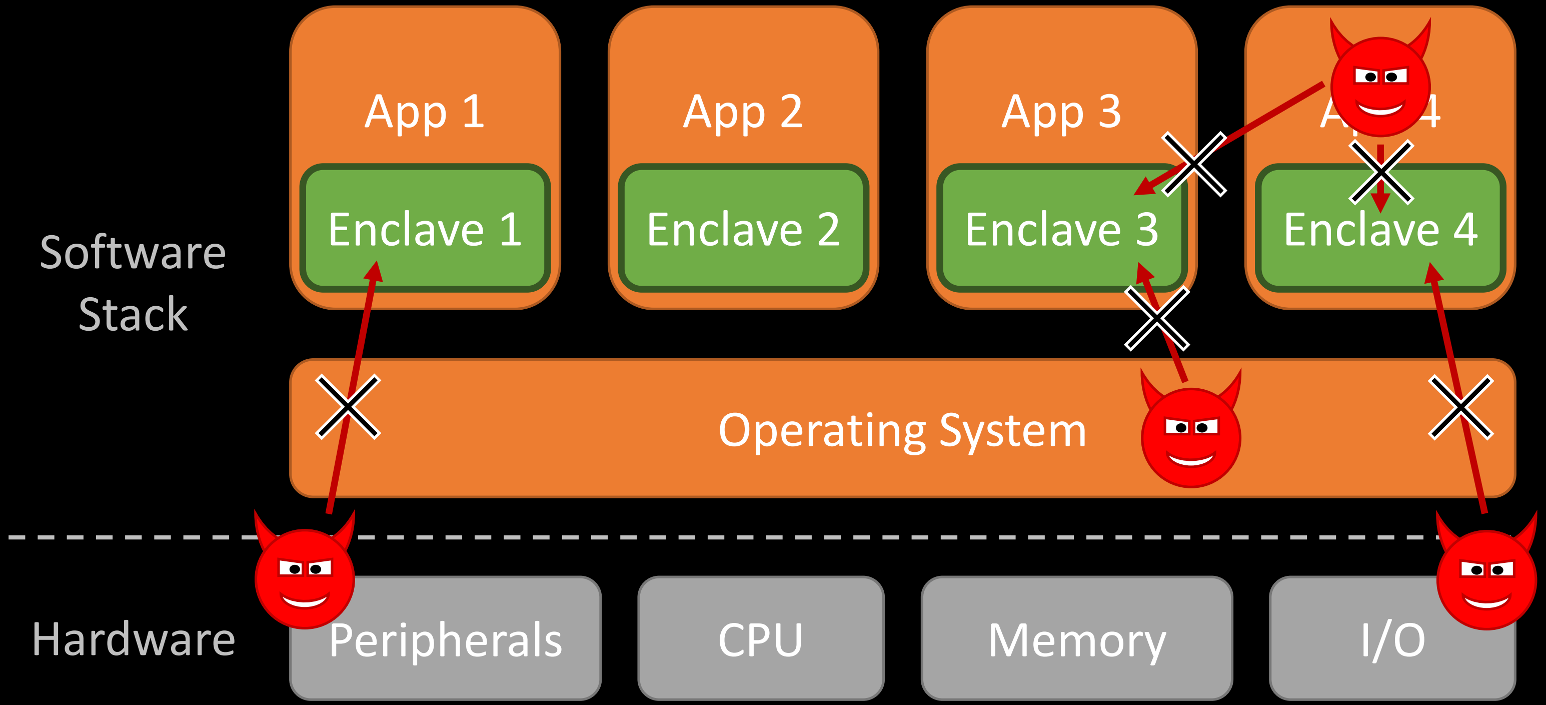
Trusted Computing Landscape



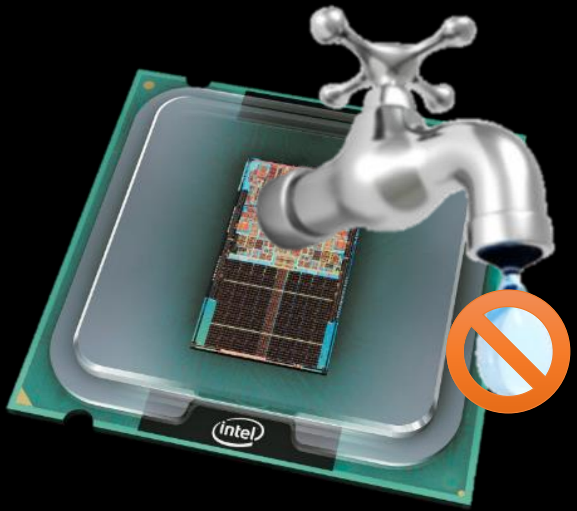
Example: Intel SGX

Intel Software Guard Extensions (SGX)

Assumptions: All software, and some hardware components, can be untrusted

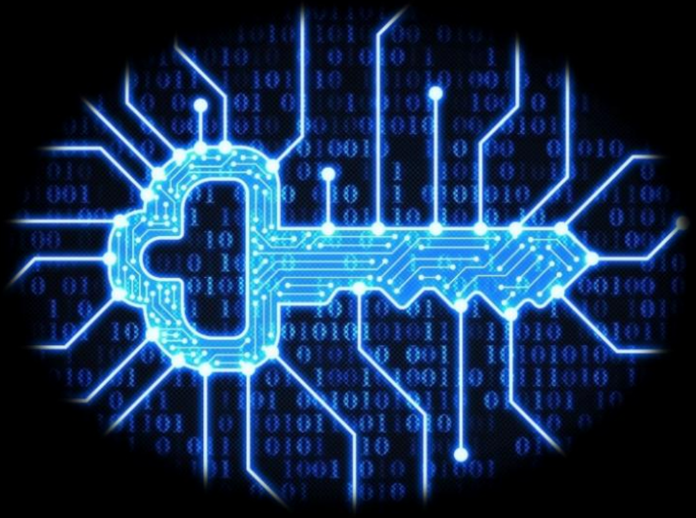


Information Leakage



Real-World Consequences

Extracting 2048-bit RSA decryption key from the enclave



Extracting genome sequences from the enclave

The Rise & Hype of Cross-Layer Exploits

- Recent microarchitectural/hardware-based attacks are exploiting issues that originate in the underlying hardware/microarchitecture

Meltdown

CLKScrew

Spectre

TLBleed

RIDL

Rowhammer

ZombieLoad

Foreshadow

Spoiler

CacheBleed

Fallout

Platypus

The Rise & Hype of Cross-Layer Exploits

- Recent microarchitectural/hardware-based attacks are exploiting issues that originate in the underlying hardware/microarchitecture

Spectre

Meltdown

TLBleed

Zom

Foreshadow

cheBleed

Fallout

Platypus

Hardware cannot be patched in-silicon!

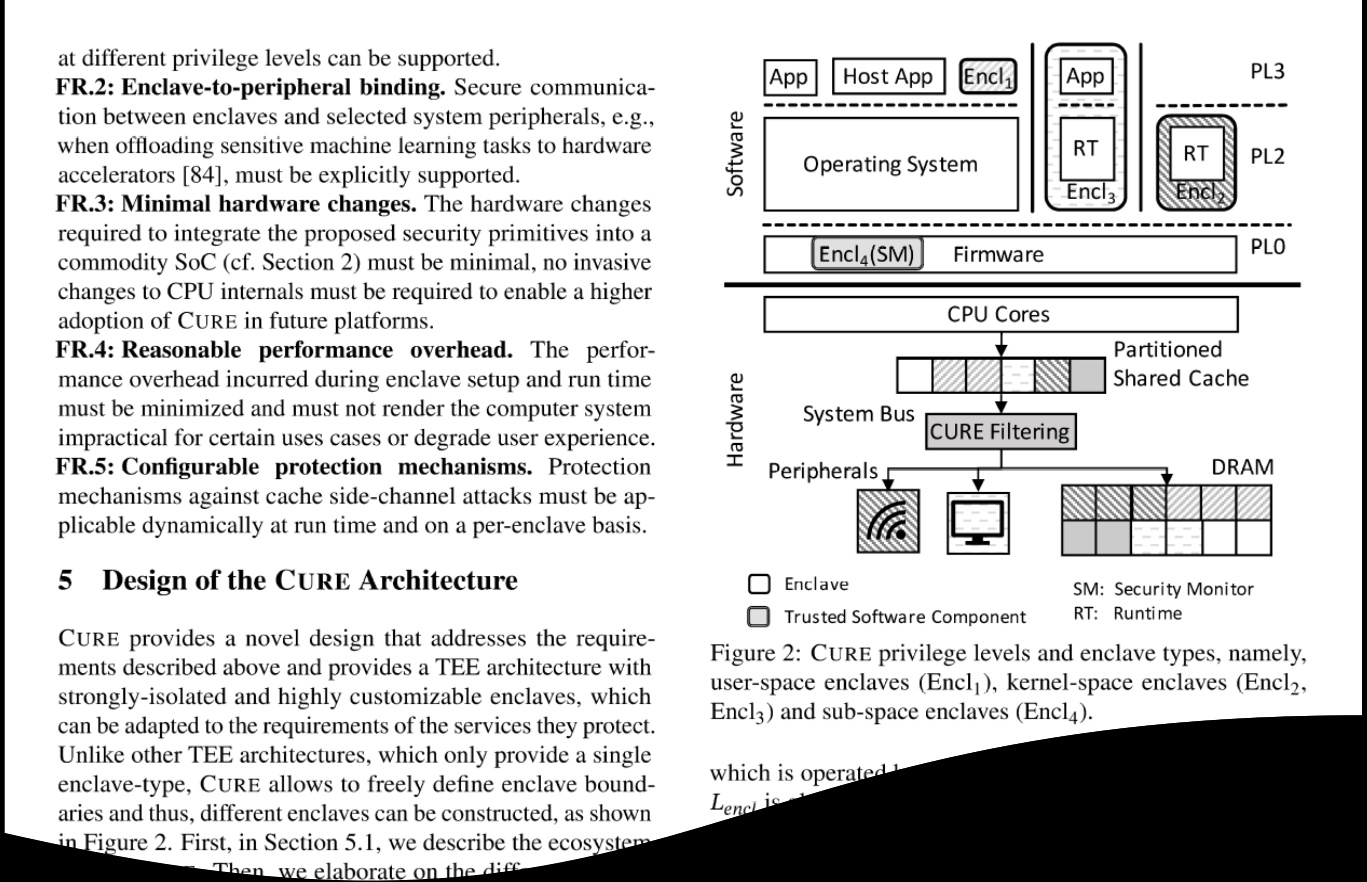
SPECTRE

MELTDOWN

Towards Resilient
Enclave-based Security Architectures

CURE: A Security Architecture with CUstomizable and Resilient Enclaves

Raad Bahmani, Ferdinand Brasser, Ghada Dessouky, Patrick Jauernig, Matthias Klimmek, Ahmad-Reza Sadeghi, Emmanuel Stapf at USENIX Security 2021



at different privilege levels can be supported.

FR.2: Enclave-to-peripheral binding. Secure communication between enclaves and selected system peripherals, e.g., when offloading sensitive machine learning tasks to hardware accelerators [84], must be explicitly supported.

FR.3: Minimal hardware changes. The hardware changes required to integrate the proposed security primitives into a commodity SoC (cf. Section 2) must be minimal, no invasive changes to CPU internals must be required to enable a higher adoption of CURE in future platforms.

FR.4: Reasonable performance overhead. The performance overhead incurred during enclave setup and run time must be minimized and must not render the computer system impractical for certain uses cases or degrade user experience.

FR.5: Configurable protection mechanisms. Protection mechanisms against cache side-channel attacks must be applicable dynamically at run time and on a per-enclave basis.

5 Design of the CURE Architecture

CURE provides a novel design that addresses the requirements described above and provides a TEE architecture with strongly-isolated and highly customizable enclaves, which can be adapted to the requirements of the services they protect. Unlike other TEE architectures, which only provide a single enclave-type, CURE allows to freely define enclave boundaries and thus, different enclaves can be constructed, as shown in Figure 2. First, in Section 5.1, we describe the ecosystem Then, we elaborate on the diff

Figure 2: CURE privilege levels and enclave types, namely, user-space enclaves ($Encl_1$), kernel-space enclaves ($Encl_2$, $Encl_3$) and sub-space enclaves ($Encl_4$).

which is operated

L_{encl} is

- Multiple *types* of enclaves
- Security-critical tasks and services (e.g., remote attestation) managed by trusted software component (SM)
- Way-based cache partitioning on shared L2 cache
- Novel access control mechanism on system bus, minimal changes at processor
- Allows for secure binding between enclaves and peripherals

Deep Dive into Hardware Vulnerabilities

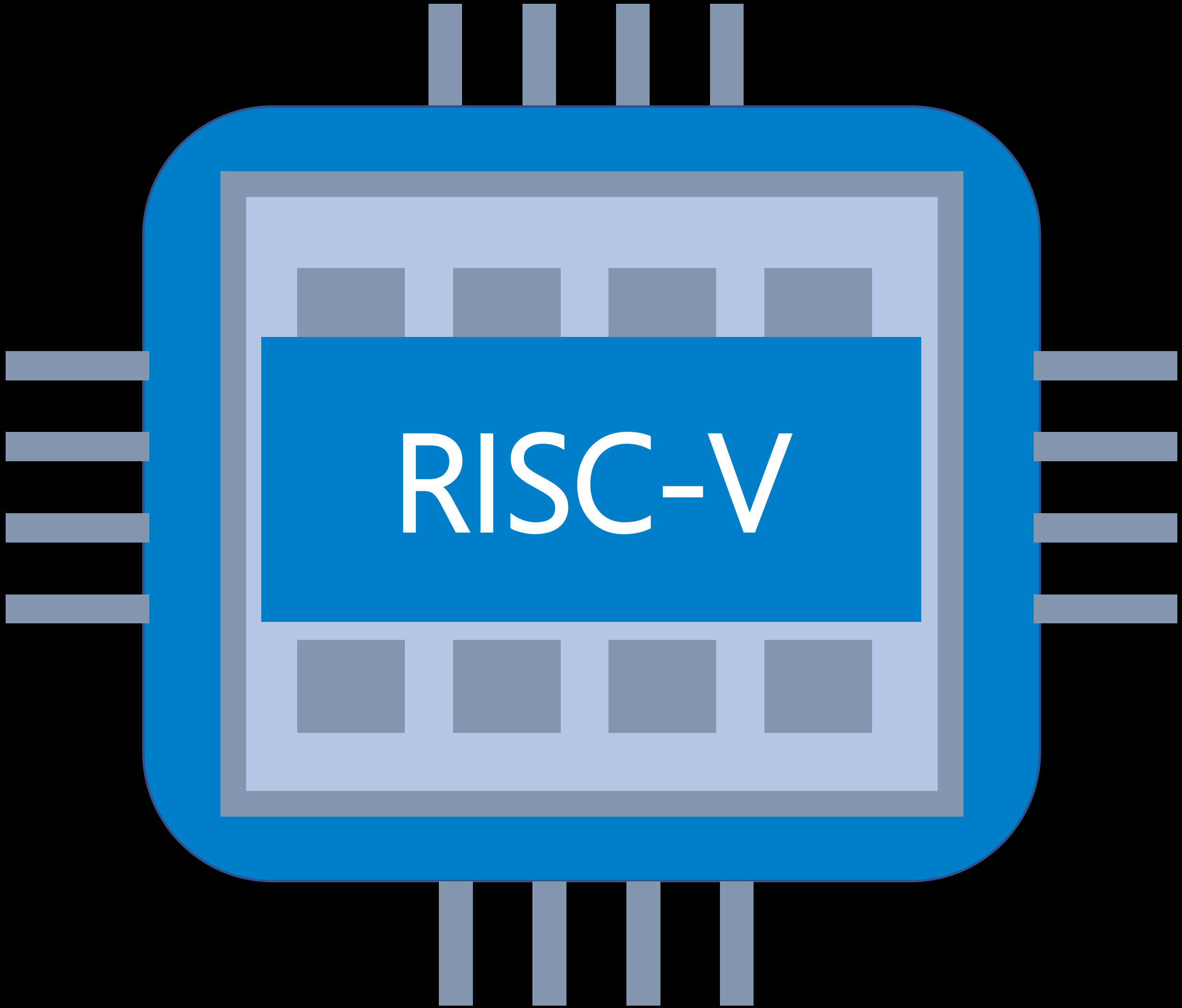
International Hardware Security Capture-The-Flag Competition, HACK@ Franchise

Hack@DAC: Why and How?

- Deep dive into real-world hardware security vulnerabilities and detection methodologies
- Raise the bar for hardware security in the semiconductor industry
  - Lots of established techniques for software security assurance
  - But limited number of tools for hardware security assurance
- RISC-V SoC testbeds with injected bugs constructed in collaboration with Intel hardware security professionals
- Over the past 3 years: A total of 174 teams from academia and industry from all over the world
- Own investigation of the effectiveness of typical industry standard formal verification tools

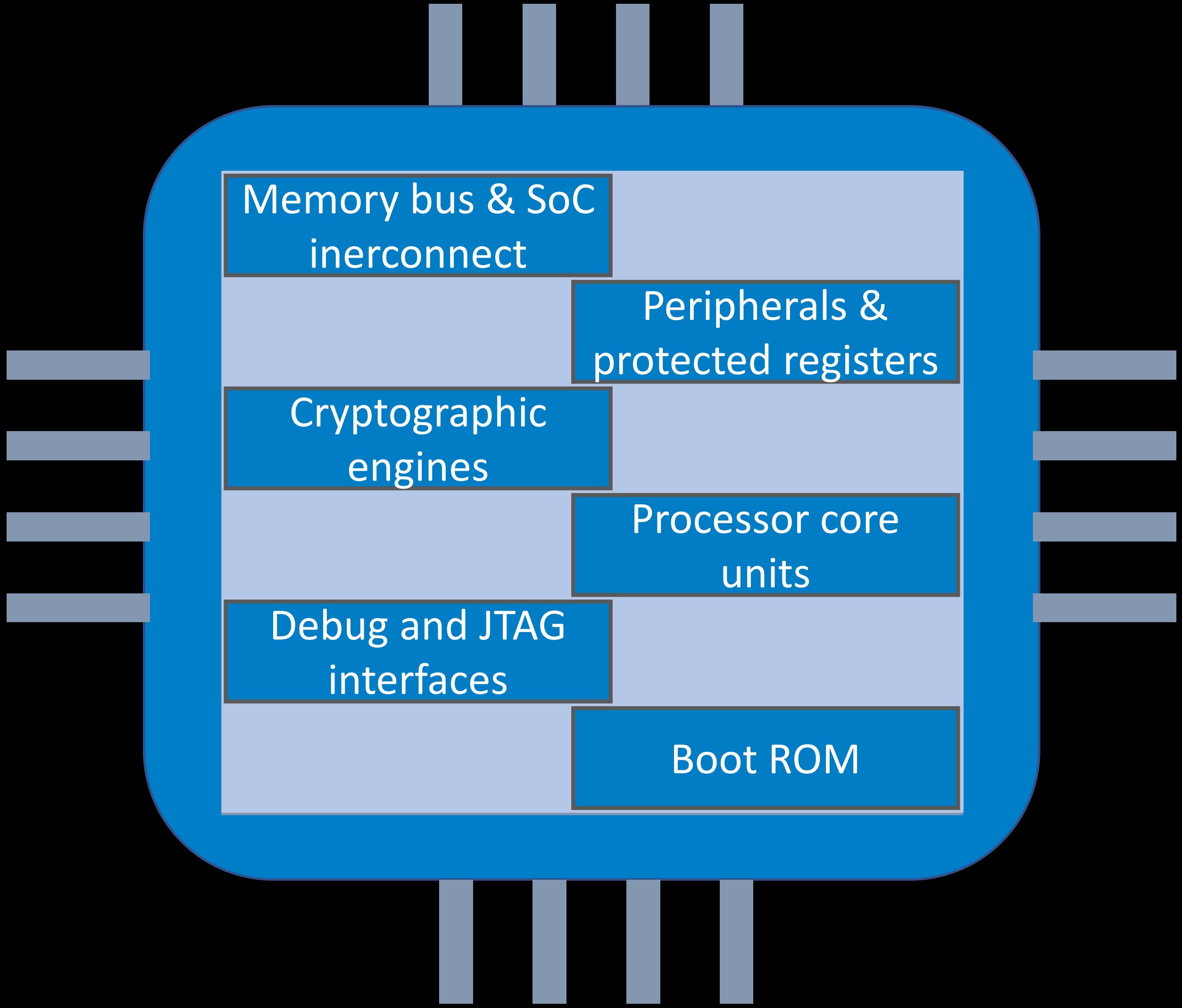
Systematic RTL Bugs Construction

Spectrum of CVEs involving SW-exploitable hardware and firmware bugs



Real-world hardware bugs encountered by hardware and system security professionals at Intel

Systematic RTL Bugs Construction



Memory bus & SoC inerconnect

Peripherals & protected registers

Cryptographic engines

Processor core units

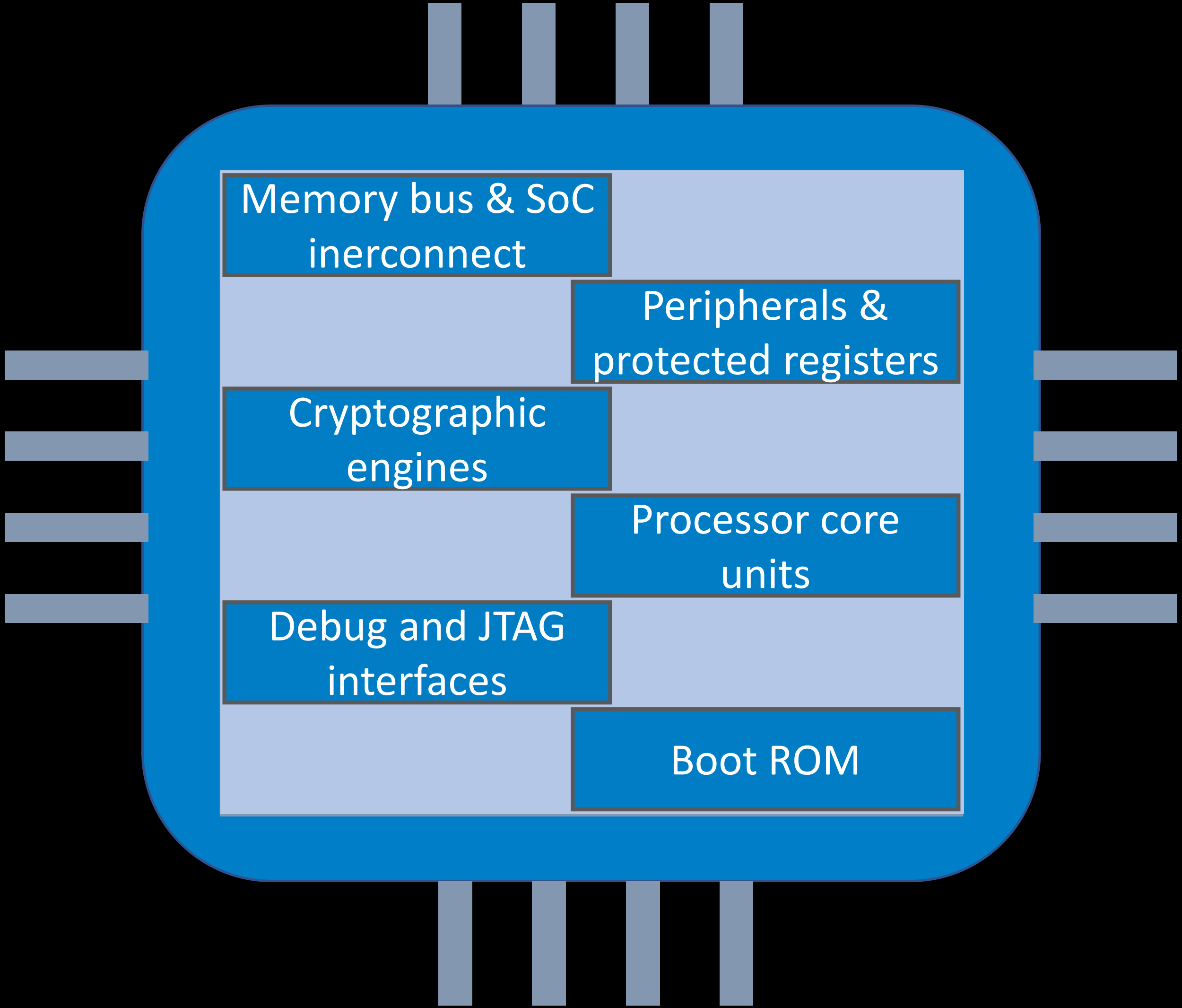
Debug and JTAG interfaces

Boot ROM

Systematic RTL Bugs Construction

denial-of-service

sensitive information leakage



Memory bus & SoC inerconnect

Peripherals & protected registers

Cryptographic engines

Processor core units

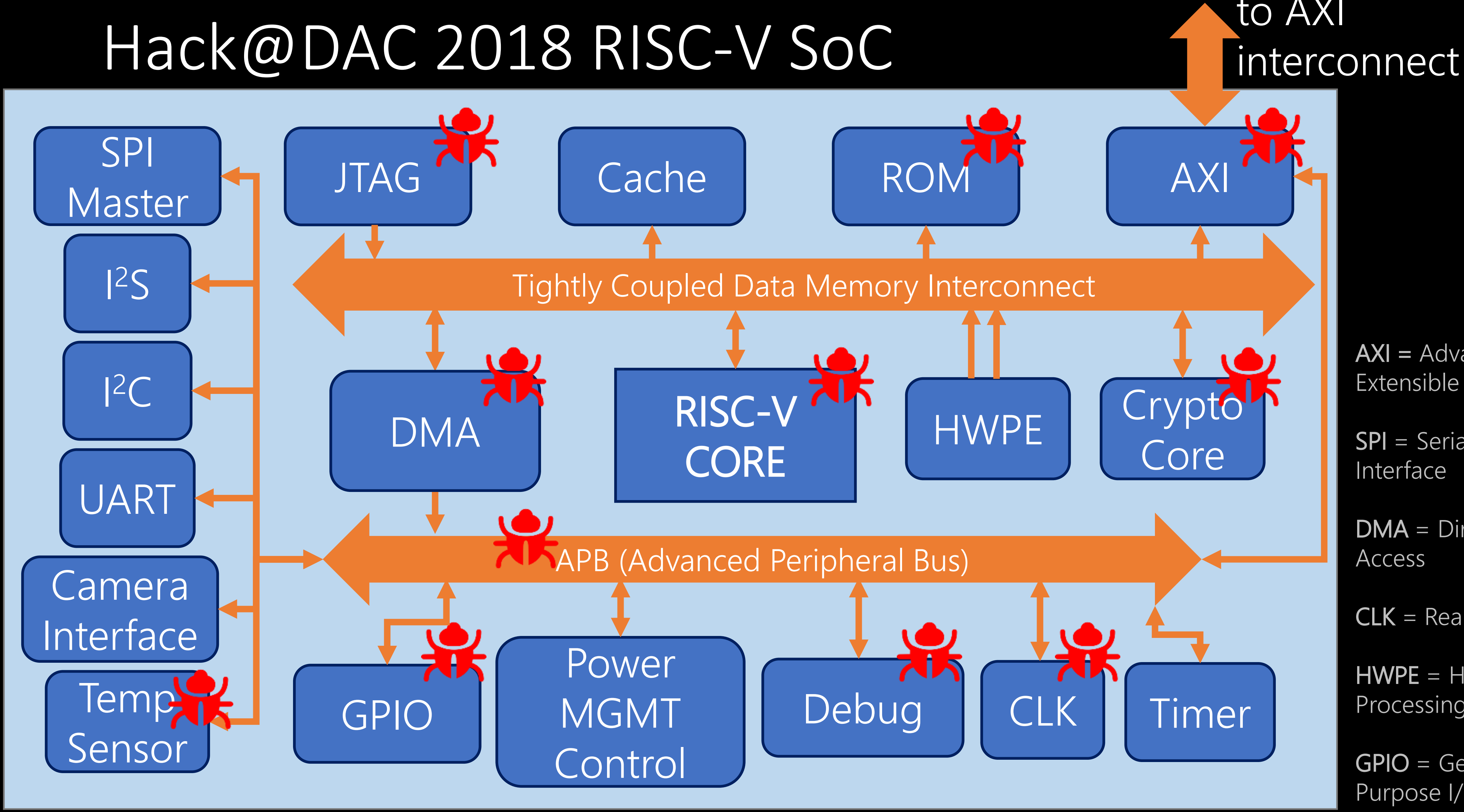
Debug and JTAG interfaces

Boot ROM

privilege escalation

software exploitability

Hack@DAC 2018 RISC-V SoC



AXI = Advanced Extensible Interface

SPI = Serial Peripheral Interface

DMA = Direct Memory Access

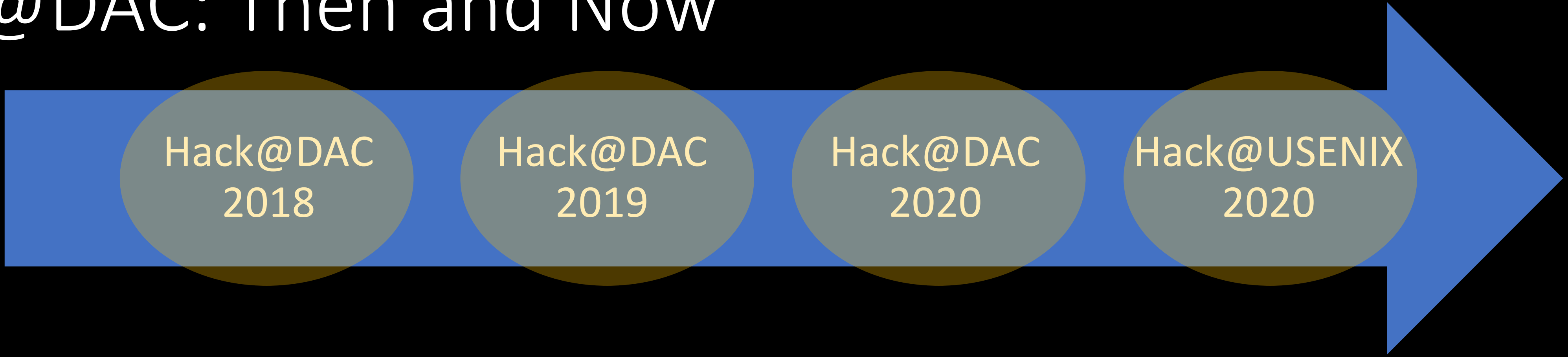
CLK = Real-Time Clock

HWPE = Hardware Processing Elements

GPIO = General Purpose I/O

2 RISC-V SoCs used: PULPino & PULPissimo

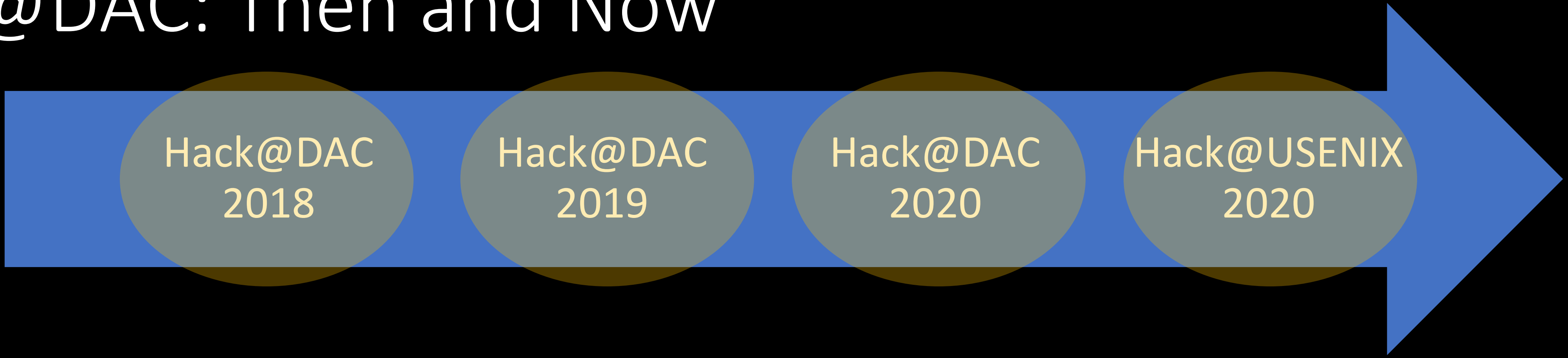
Hack@DAC: Then and Now



https://hackat.events/dac18/ https://hackat.events/dac19/

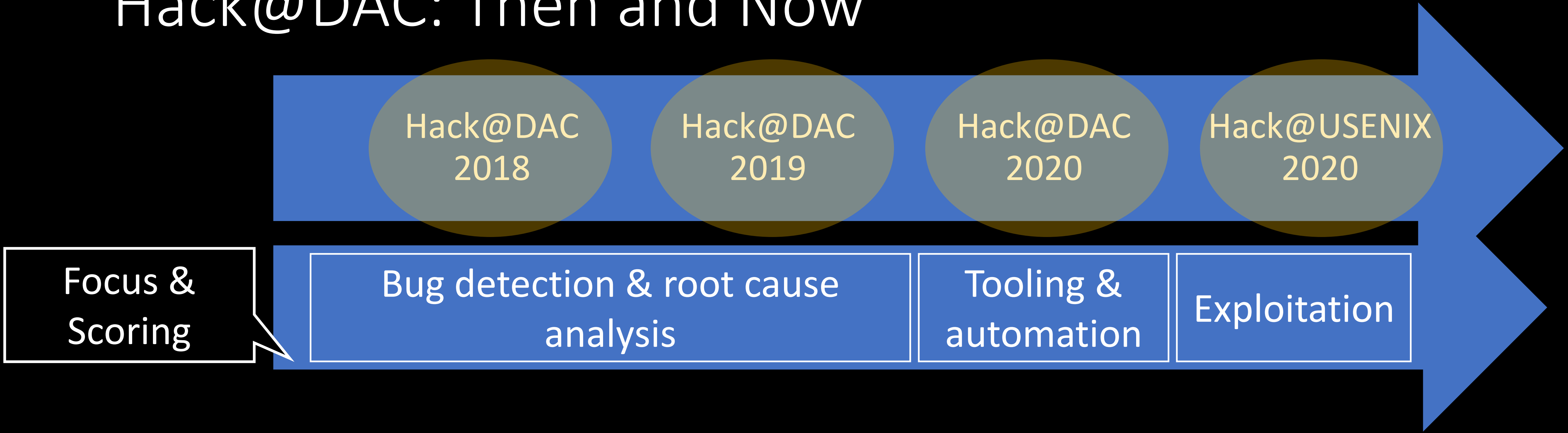
https://hackat.events/dac20/ https://hackat.events/sec20/

Hack@DAC: Then and Now



https://hackat.events/dac18/ https://hackat.events/dac19/
https://hackat.events/dac20/ https://hackat.events/sec20/

Hack@DAC: Then and Now



Hack@DAC 2018

Hack@DAC 2019

Hack@DAC 2020

Hack@USENIX 2020

Focus & Scoring

Bug detection & root cause analysis

Tooling & automation

Exploitation

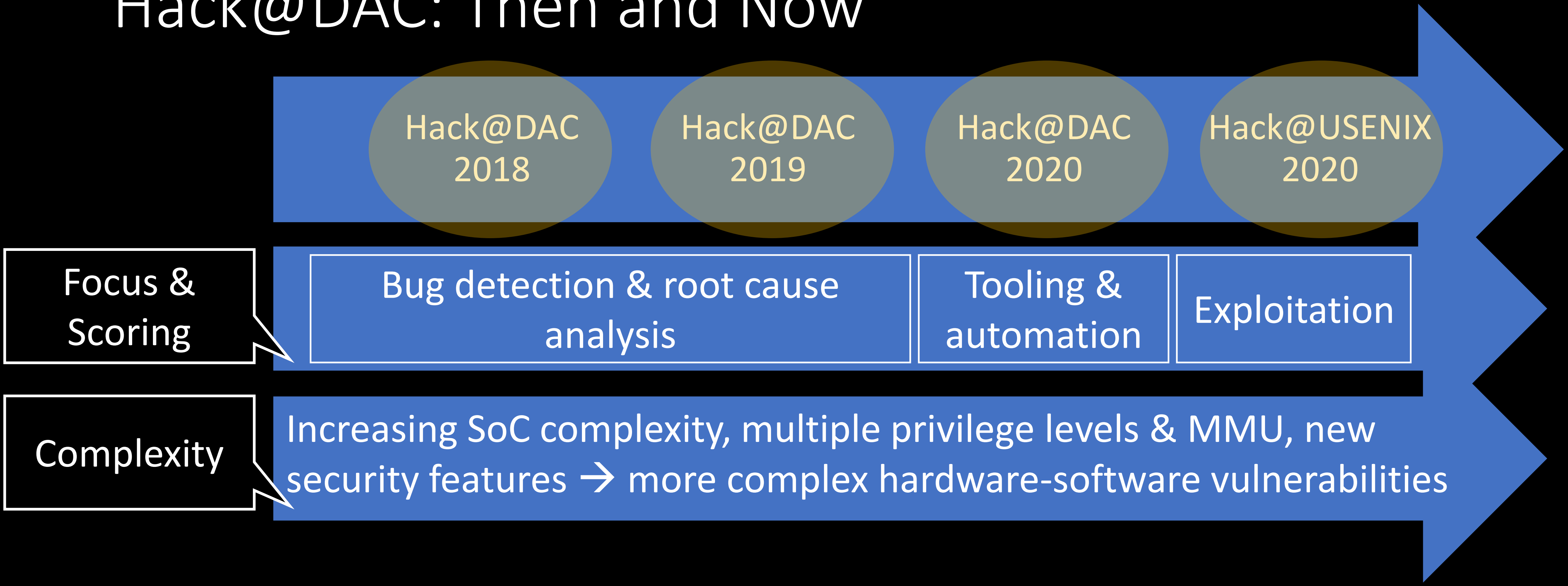
https://hackat.events/dac18/

https://hackat.events/dac19/

https://hackat.events/dac20/

https://hackat.events/sec20/

Hack@DAC: Then and Now



Hack@DAC 2018

Hack@DAC 2019

Hack@DAC 2020

Hack@USENIX 2020

Focus & Scoring

Bug detection & root cause analysis

Tooling & automation

Exploitation

Complexity

Increasing SoC complexity, multiple privilege levels & MMU, new security features → more complex hardware-software vulnerabilities

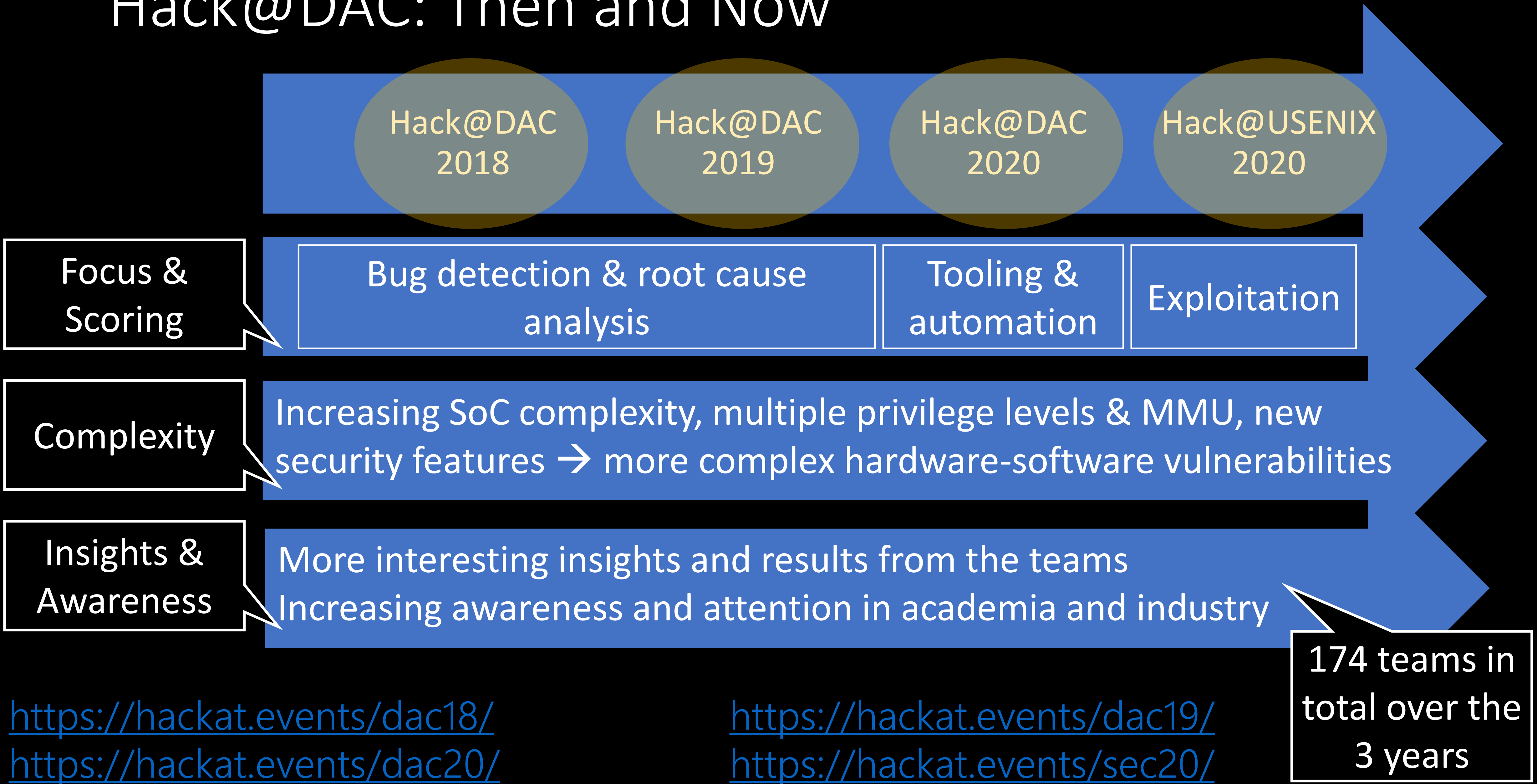
https://hackat.events/dac18/

https://hackat.events/dac19/

https://hackat.events/dac20/

https://hackat.events/sec20/

Hack@DAC: Then and Now



Hack@DAC 2018

Hack@DAC 2019

Hack@DAC 2020

Hack@USENIX 2020

Focus & Scoring

Bug detection & root cause analysis

Tooling & automation

Exploitation

Complexity

Increasing SoC complexity, multiple privilege levels & MMU, new security features → more complex hardware-software vulnerabilities

Insights & Awareness

More interesting insights and results from the teams

Increasing awareness and attention in academia and industry

174 teams in total over the 3 years

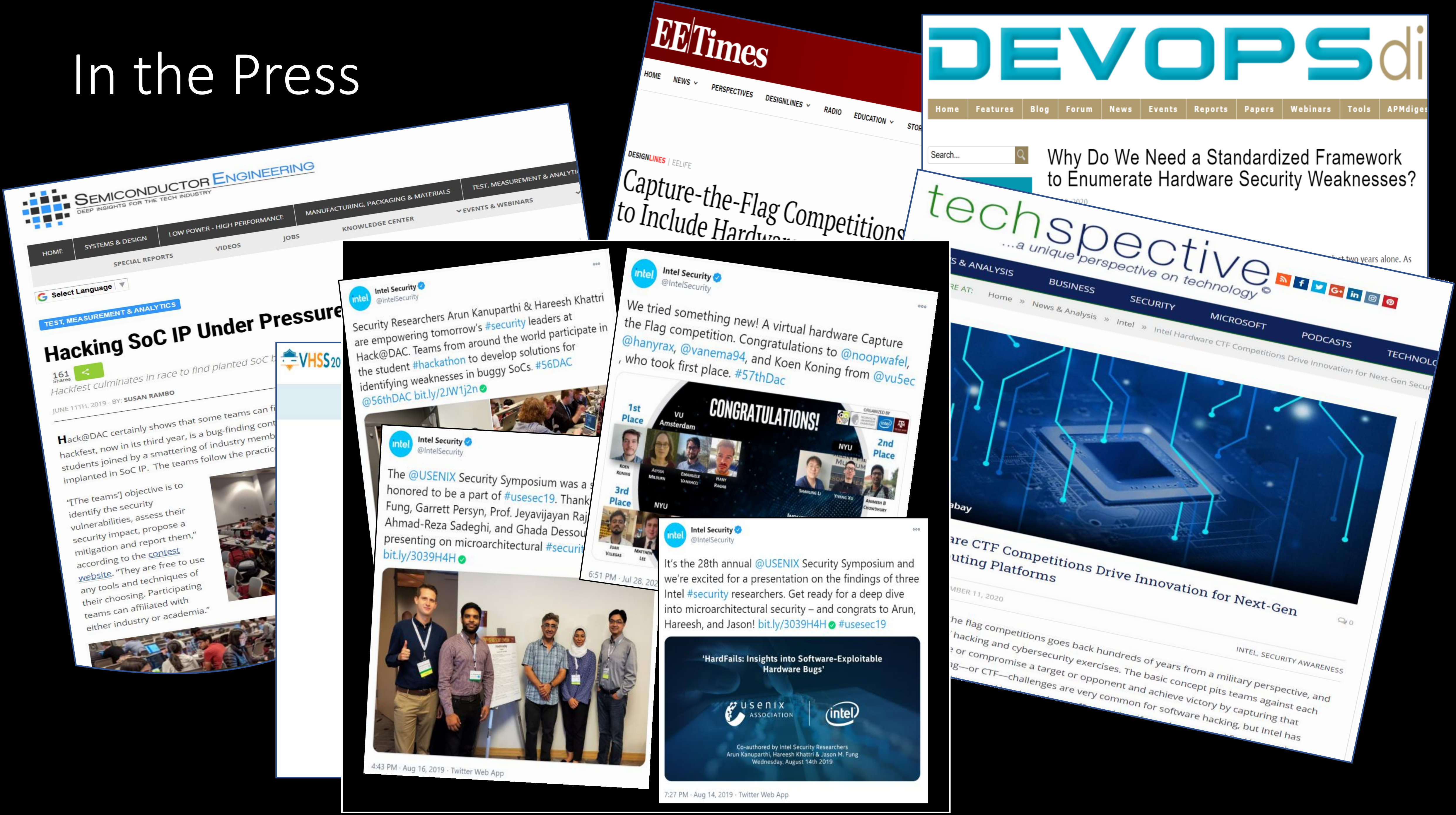
https://hackat.events/dac18/

https://hackat.events/dac19/

https://hackat.events/dac20/

https://hackat.events/sec20/

In the Press



EE Times

HOME NEWS PERSPECTIVES DESIGNLINES RADIO EDUCATION STOR

DESIGNLINES | EELIFE

Capture-the-Flag Competitions to Include Hardwar

DEVOPSdi

Home Features Blog Forum News Events Reports Papers Webinars Tools APMdiges

Search...

Why Do We Need a Standardized Framework to Enumerate Hardware Security Weaknesses?

techspective ...a unique perspective on technology©

S & ANALYSIS BUSINESS SECURITY MICROSOFT PODCASTS TECHNOLO

RE AT: Home » News & Analysis » Intel » Intel Hardware CTF Competitions Drive Innovation for Next-Gen Secur

are CTF Competitions Drive Innovation for Next-Gen uting Platforms

MBER 11, 2020

INTEL, SECURITY AWARENESS

he flag competitions goes back hundreds of years from a military perspective, and hacking and cybersecurity exercises. The basic concept pits teams against each e or compromise a target or opponent and achieve victory by capturing that g—or CTF—challenges are very common for software hacking, but Intel has

SEMICONDUCTOR ENGINEERING DEEP INSIGHTS FOR THE TECH INDUSTRY

HOME SYSTEMS & DESIGN LOW POWER - HIGH PERFORMANCE MANUFACTURING, PACKAGING & MATERIALS TEST, MEASUREMENT & ANALYTI SPECIAL REPORTS VIDEOS JOBS KNOWLEDGE CENTER EVENTS & WEBINARS

Select Language

TEST, MEASUREMENT & ANALYTICS

Hacking SoC IP Under Pressure

161 Shares

Hackfest culminates in race to find planted SoC b

JUNE 11TH, 2019 - BY: SUSAN RAMBO

Hack@DAC certainly shows that some teams can fi hackfest, now in its third year, is a bug-finding cont students joined by a smattering of industry memb implanted in SoC IP. The teams follow the practic

"[The teams'] objective is to identify the security vulnerabilities, assess their security impact, propose a mitigation and report them," according to the contest website. "They are free to use any tools and techniques of their choosing. Participating teams can affiliated with either industry or academia."

VHSS20

Intel Security @IntelSecurity

Security Researchers Arun Kanuparthi & Hareesh Khattri are empowering tomorrow's #security leaders at Hack@DAC. Teams from around the world participate in the student #hackathon to develop solutions for identifying weaknesses in buggy SoCs. #56DAC @56thDAC bit.ly/2JW1j2n

Intel Security @IntelSecurity

We tried something new! A virtual hardware Capture the Flag competition. Congratulations to @noopwafel, @hanyrax, @vanema94, and Koen Koning from @vu5ec , who took first place. #57thDac

CONGRATULATIONS! ORGANIZED BY

1st Place VU Amsterdam Koen Koning Alyssa Milburn Emanuele Vannacci Hany Ragab

2nd Place NYU Shanling Li Yiyang Xu Animesh B Chowdhury

3rd Place NYU Juan Villegas Matthew Lee

6:51 PM · Jul 28, 202

Intel Security @IntelSecurity

The @USENIX Security Symposium was a honored to be a part of #usesec19. Thank Fung, Garrett Persyn, Prof. Jeyavijayan Raj Ahmad-Reza Sadeghi, and Ghada Dessou presenting on microarchitectural #securit bit.ly/3039H4H

4:43 PM · Aug 16, 2019 · Twitter Web App

Intel Security @IntelSecurity

It's the 28th annual @USENIX Security Symposium and we're excited for a presentation on the findings of three Intel #security researchers. Get ready for a deep dive into microarchitectural security – and congrats to Arun, Hareesh, and Jason! bit.ly/3039H4H #usesec19

'HardFails: Insights into Software-Exploitable Hardware Bugs'

usenix ASSOCIATION intel

Co-authored by Intel Security Researchers Arun Kanuparthi, Hareesh Khattri & Jason M. Fung Wednesday, August 14th 2019

7:27 PM · Aug 14, 2019 · Twitter Web App

Conclusion

- Trusted computing could not keep its promises
  - Still suffering from legacy issues
  - The impact of side-channels and transient execution were “ignored” ?
- Hardware security validation is still its infacny
- Current tools and metrics for security are limited
  - No fuzzing or symbolic execution for RTL code
  - Significant expertise, explicit definition of security property assertions and manual inspection required
  - Scalability and consolidated hardware/firmware verification are open challenges
- Real-world systems are highly proprietary
  - False sense of security („by-obscurity“)
  - Open-source hardware such as RISC-V may help, like Linux in software
- Customizable security architecture may be more successful towards more resilient computing platforms (see CURE, *Bahmani et al USENIX SEC 2021)*

What’s Next for Hack@DAC?

- Hack@DAC goes to the cloud – Amazon’s AWS FPGA instances to host the SoCs and open-source security assurance tools

- A learning hub on hardware security assurance for academia & industry

- Developing new effective & efficient hardware security assurance techniques, e.g. hardware fuzzing

Contact us:

ahmad.sadeghi@trust.tu-darmstadt.de