CSE/CEN 598
Hardware Security & Trust

Hardware Root-of-Trust Design

Prof. Michel A. Kinsy

Trust Challenges in Computing Systems

- Digital/Cyber Identification
  - What is the identity of the requesting computer?
  - What proof is there that the claim of identity is genuine?
- Digital/Cyber Identification
  - How can these two computers establish whether either of them is operating as designed or that a compromise has occurred?
  - Identity alone is not enough for trust – what if one computer has been compromised with malware?
  - What is the basis for a genuine claim of hardware or software integrity?

Security vs. Trust

- Security issues arise from the computing system's vulnerability to attacks
- Trust issues arise from involvement of untrusted entities and components in the life cycle of the hardware
  - Untrusted IP and Compute-Aid Design (CAD) tool vendors, untrusted design, fabrication, test, and distribution facilities, and lack of traceable provenance
  - These parties are capable of violating the trustworthiness of the hardware component and system
- Trust issues often lead to security concerns

Trusted vs. Trustworthy

- When a component of a system is trusted
  - Security of the system depends on it
  - Failure of component will compromise the security policy
  - Determined by its role in the system
- When a component is trustworthy
  - Component is deemed to be trusted
    - e.g., It is implemented correctly
  - Determined by intrinsic properties of the component

Computing System Security Patterns

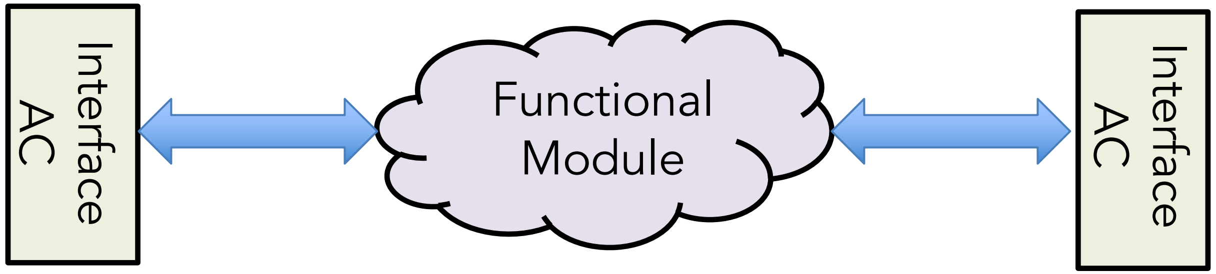
- Patterns
  - Procedural
  - Algorithmic
  - Structural
  - Horizontal and vertical views
    - Software and hardware
    - Methodologies
      - Secure by design approach
        - Methods to check the integrity of computing processes
        - Monitored and controlled access to system resources
        - Predictable computing services and behaviors

Secure by Design

- Some of the well understood concepts are
  - Least Privilege
    - Provide to each component (hardware module or software routine) only the privileges it needs
  - Fail-safe defaults
    - Only allow explicit permission
  - Efficient security mechanism
    - Implement simple security mechanisms to encourage usage
  - Formally Secure
    - Avoid relying on security by secrecy or obscurity
- They implementation is another issue altogether

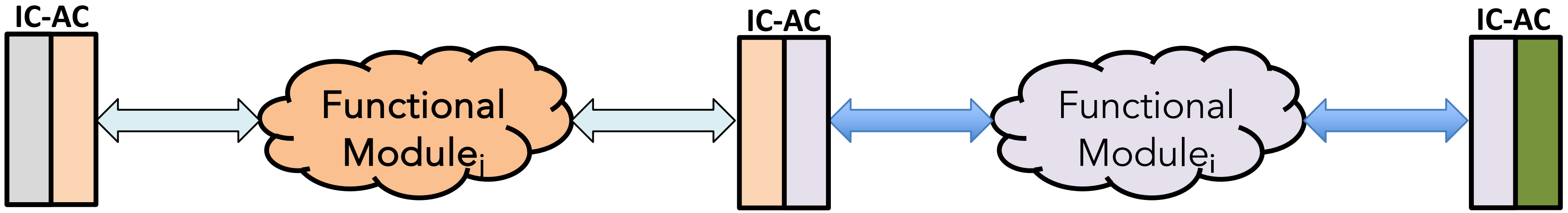
Interface-Centric Access Control

- Decide how to partition the design
  - Both physically and logically to implement IC-AC
- Regulate whether components have sufficient privileges to communicate through certain interfaces
- Provide secure interaction between secure/non-secure components
- Formally verify the composition of IC-AC policies, i.e., proper access privilege propagations
- Route messages according to policies implemented in the IC-ACs



Interface-Centric Access Control

- The module's security needs to encompass its interfaces
- It's each module's responsibility to guarantee its interface's security
- Proper hand-offs




```
Object_xi{
        Element a;
        Element b;
        Attribute p;
        Attribute q;
        Rule k;
}
```

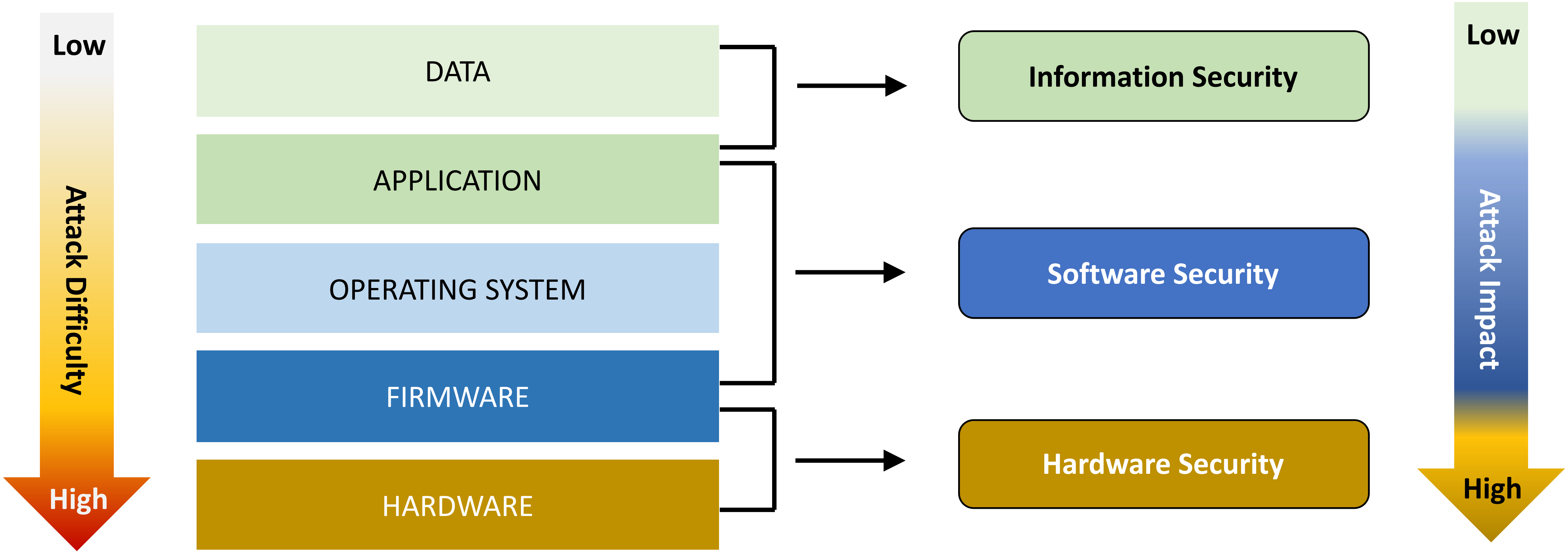
```
Object_i{
        Element b;
        Attribute p;
        Rule f(k);
}
```

```
Object_ij{
        Element e;
        Element f;
        Element g;
        Attribute u;
        Rule r_1, r_1;
}
```

```
Object_j{
        Element e;
        Element f;
        Attribute v;
        Rule f(r_2);
}
```

```
Object_jx{
        …
}
```

Computing System Layers

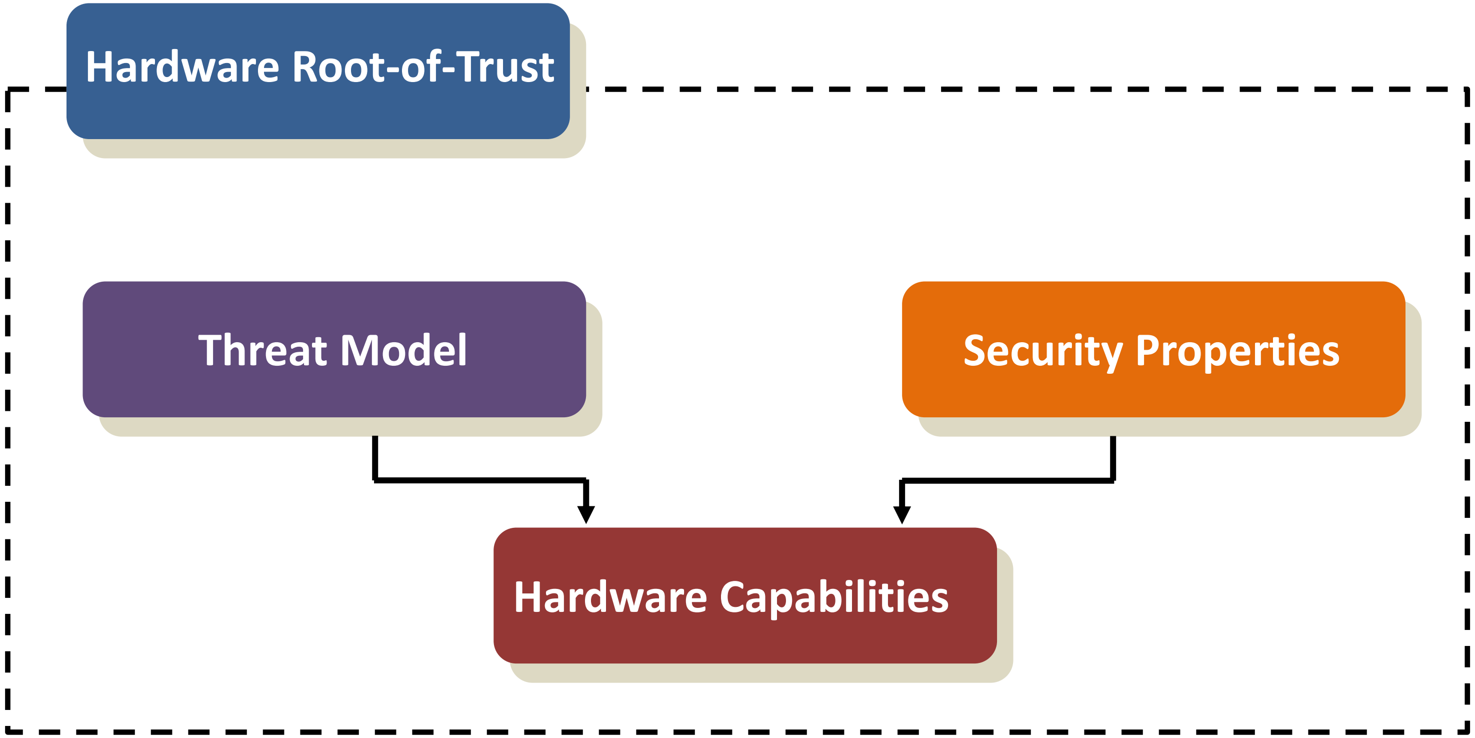


Security &Trust Anchor Candidate

- Since attack difficulty is at the highest with the hardware, it presents an excellent anchor for the compute security features
- Hardware as the Root-of-Trust (RoT) Design Methodologies
  - NIST (NIST SP 1800-19B) defines Hardware Root-of-Trust as "An inherently trusted combination of hardware and firmware that maintains the integrity of information."
  - Practically, Hardware Root-of-Trust (HRoT) is defined as the foundational building block(s) of different security schemes, protocols, products, or services within a secure computing system
  - Formally, Hardware Root-of-Trust (HRoT) is an immutable hardware component or a set of hardware components (e.g., an encryption engine and/or a dedicated secure processor) considered unconditionally trusted against a well-defined threat model

Hardware Root-of-Trust Properties

- Proof of authenticity and/or provenance
  - E.g., Uniquely identifiable and verifiable features
    - Physically Unclonable functions (PUFs)
- Immutable hardware component(s)
  - E.g., Anti-tamper features
- Anchor trust against a specific threat model



Roots of Trust in Trusted Computing

Root of Trust for Measurement (RTM)

- Initiates process of recording what software is running
- e.g., implement in an immutable or securely updateable component

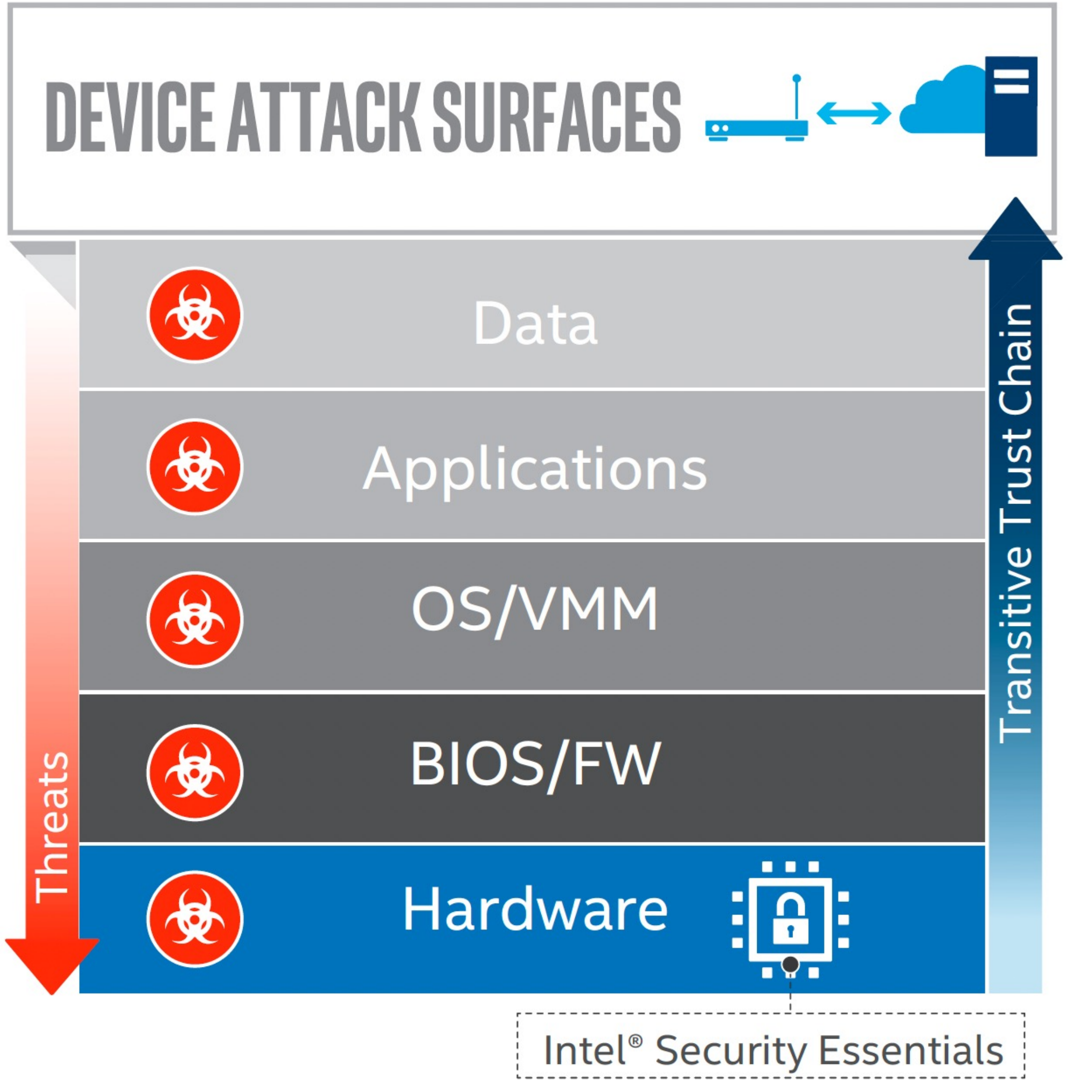
Root of Trust for Storage (RTS)

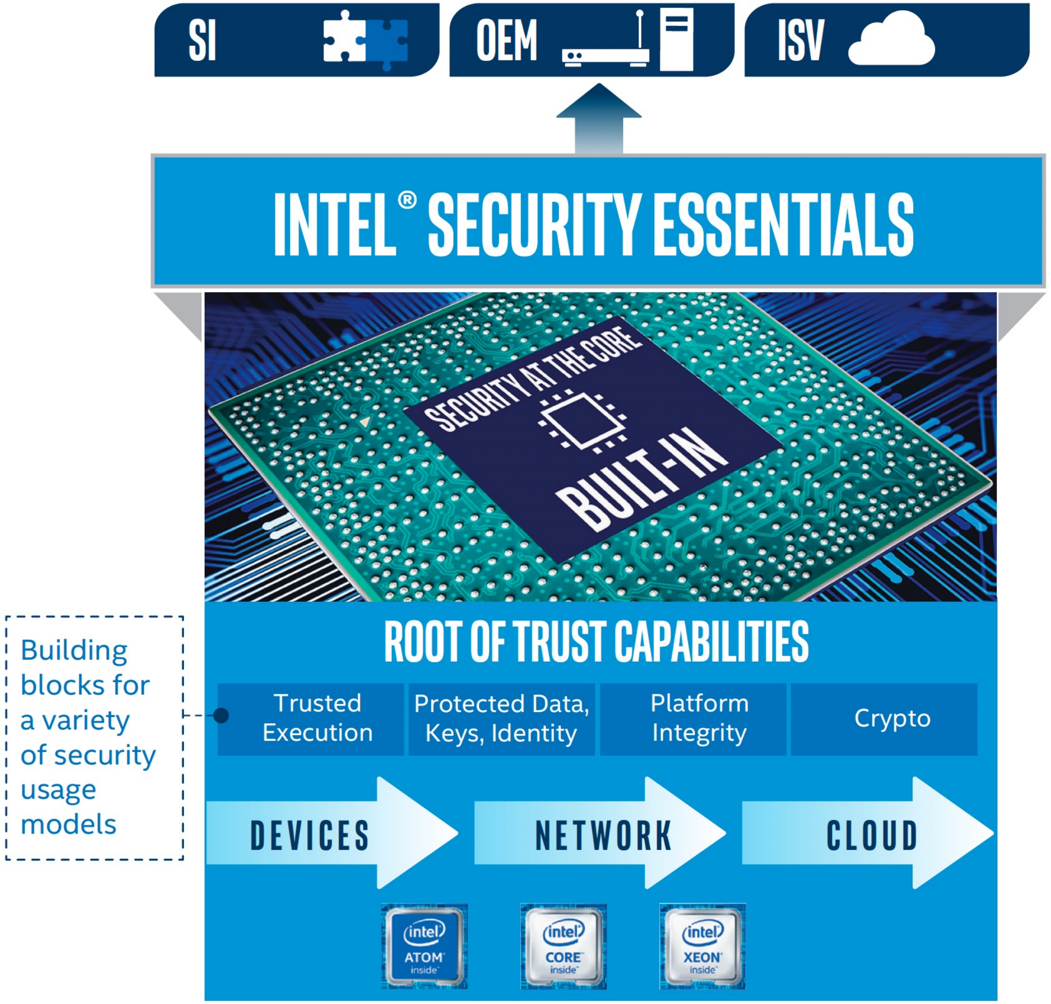
- Implements shielded locations like registers with special integrity or confidentiality
- Implemented in a Trusted Platform Module (TPM)

Root of Trust for Reporting (RTR)

- Uses cryptography to give assurances to third parties
- Built from keys burned into TPM at manufacture time

Intel Hardware Root-of-Trust Solutions



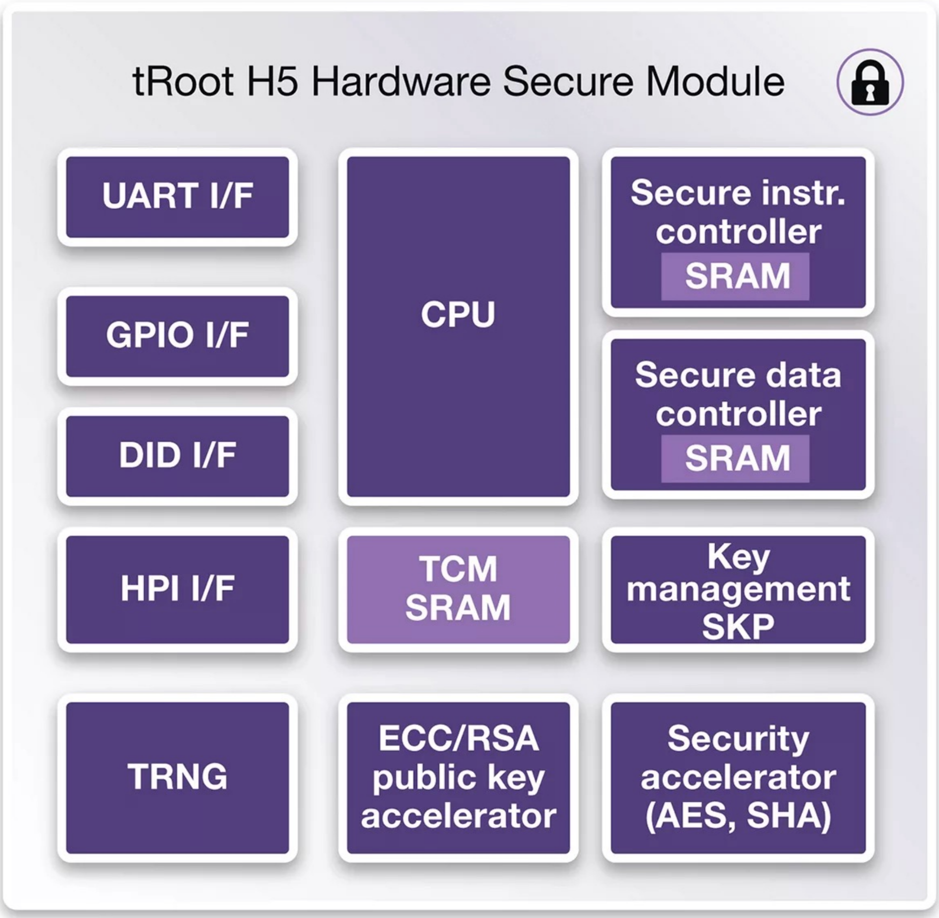


Intel Hardware Root-of-Trust Solutions

CORE CAPABILITY	TECHNOLOGY	PROTECTION FOR	DESCRIPTION
Platform Integrity	Intel® Platform Protection Technology with Boot Guard	BIOS/FW	Verifies OEM pre-OS boot loader code executing out of reset
	Intel® Platform Protection Technology with BIOS Guard	BIOS/FW	Enables a HW based static Root of Trust for measurement and verification for boot integrity
	Intel® Runtime BIOS Resilience	OS/VMM	Helps protect SSM from malicious code injection
	Intel® Platform Protection Technology with OS Guard	OS/VMM	Helps prevent malicious code from executing out of application memory space
	Intel® Platform Firmware Resiliance	BIOS/FW	Helps protect firmware from corruption; assists with system restoration in case of malware
Trusted Execution	Intel® Software Guard Extensions	Apps	Enables creation and use of isolated app enclaves to protect against attacks on executing code or data stored in memory
	Intel® Virtualization Technology	OS/VMM	Creates firewall between main OS and secure workloads running inside a secure VM
Protected Data, Keys, Identity	Intel® Platform Trust Technology	Data	Integrated HW TPM enables secure storage of keys/credentials, registry values, & boot block measurements for remote attestation
	Intel® Enhanced Privacy ID	Data	Cryptographic scheme provides direct anonymous attestation of hardware for privacy
Crypto Accelerators	Intel® Data Protection Technology with Secure Key	Data	High entropy source of random numbers to generate keys
	Intel® Advanced Encryption Standard New Instructions	Data	Accelerates math calculations for AES-NI encryption

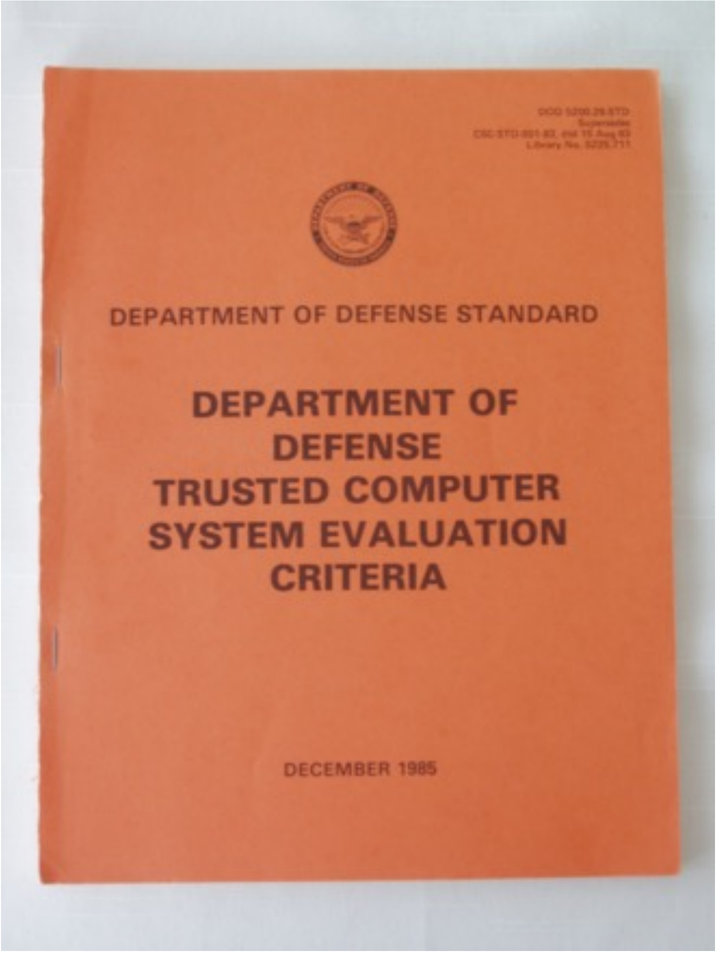
Synopsys Hardware Root-of-Trust Solutions

- Synopsys describes the tRoot™ H5 Hardware Secure Module (HSM) as their highly secure hardware root of trust
  - Enables connected devices to securely and uniquely identify and authenticate themselves to create secure channels for remote device management and service deployment



Hardware Root-of-Trust Usage Model

- DoD Orange Book
  - *"The ability of a trusted computing base to enforce correctly a unified security policy depends on the correctness of the mechanisms within the trusted computing base, the protection of those mechanisms to ensure their correctness, and the correct input of parameters related to the security policy."*



Hardware Root-of-Trust Usage Model

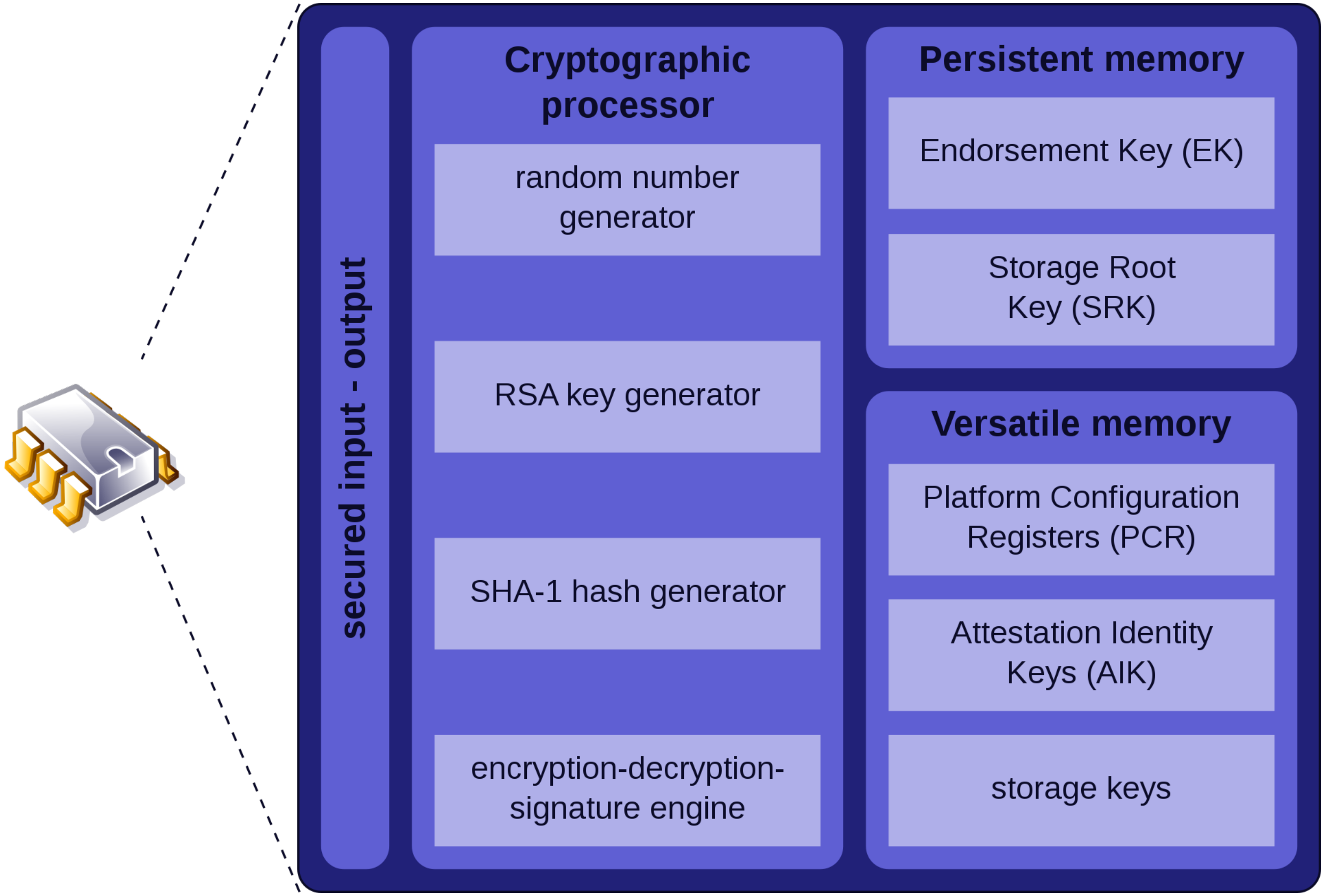
- Security kernel
  - Hardware, firmware, and software elements of a trusted computing base implementing the reference monitor concept
  - Security kernel must mediate all accesses, be protected from modification, and be verifiable as correct
- Trusted Computing Base (TCB)
  - Every secure computing system must have some TCB
  - Hardware and software necessary for enforcing all security rules
  - Vulnerabilities in the TCB can jeopardize the security of the entire system
  - Ideally
    - Rooted in hardware and small
    - Should be isolated from the rest of the computing components
    - Its correctness and runtime state should be easily and independently verifiable
- Hardware RoT to support the implementation the trusted computing base

Trusted Computing Base (TCB) Anchor

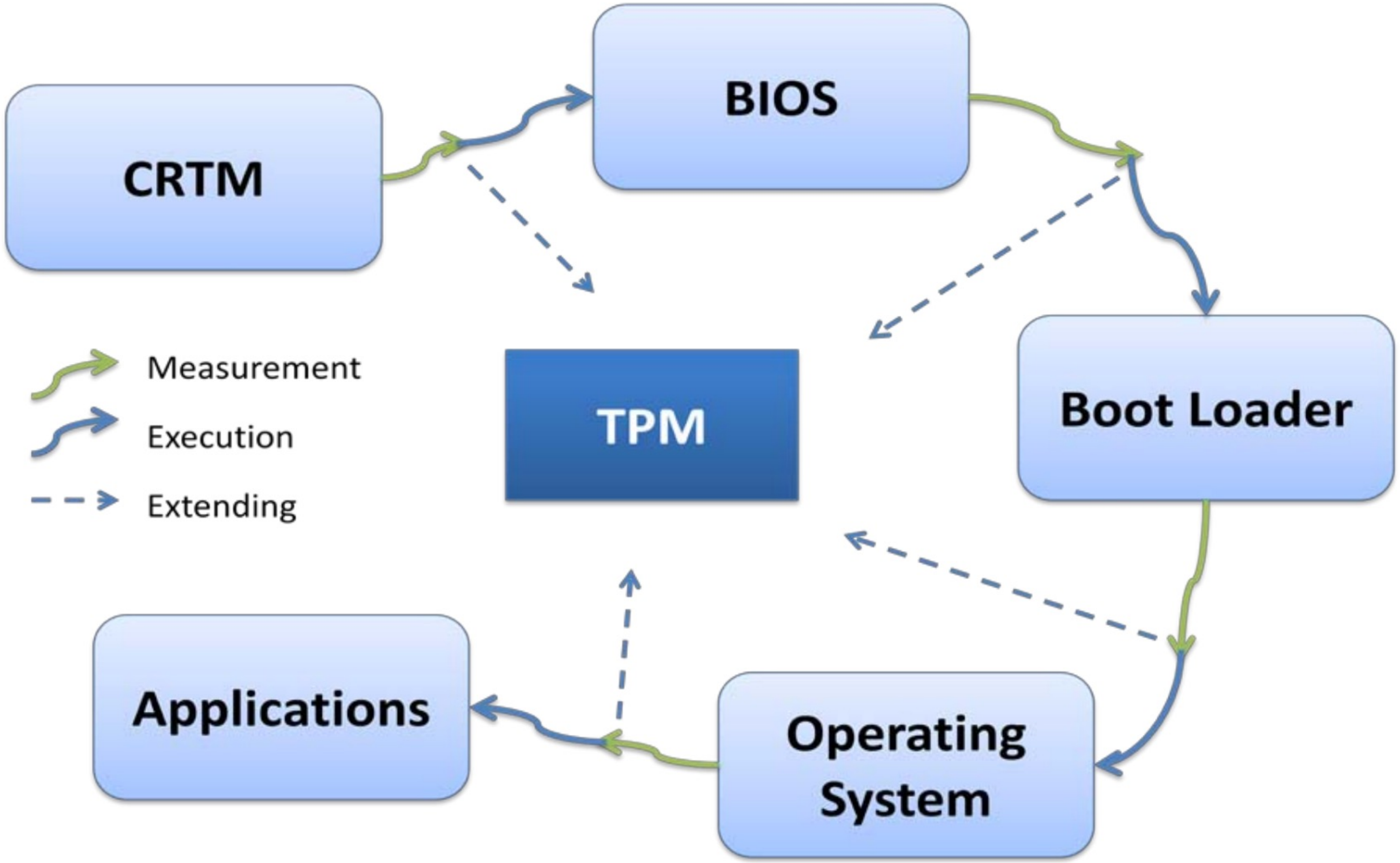
- At the heart of the Trusted Computing Base (TCB) is the Trusted Platform Module (TPM)
- The TPM provides hardware-based authentication, integrity, and attestation to the TCB
  - It is designed as a small tamper-resistant chip that provides the following functions
    - A root-of-trust for reporting and storage
    - Measurement and attestation of platform integrity
    - Platform identification and authentication
    - Core and highly constrained cryptographic functions

Trusted Computing Base (TCB) Anchor

- Trusted Platform Module (TPM) architectures
  - Storage
  - Random number generation
  - Cryptographic function and processing
- Trusted Platform Module (TPM) types
  - Discrete TPMs
  - Integrated TPMs
  - Firmware TPMs (fTPMs)
  - Hypervisor TPMs (vTPMs)
  - Software TPMs

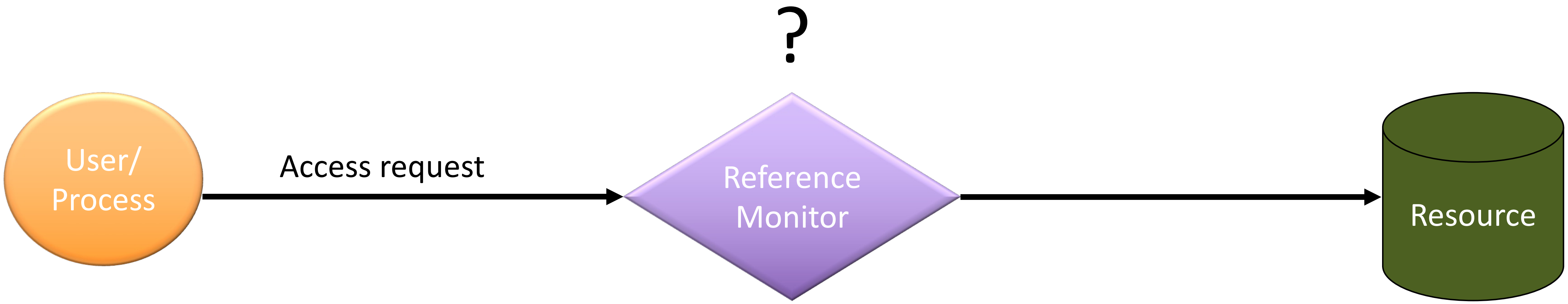


Trusted Computing Base (TCB) Anchor



Trusted Computing Base (TCB) Anchor

- Reference monitor is an abstraction that is used to validate access to objects by authorized subjects.
  - Complete mediation
  - Tamperproof
  - Verifiable



How is it implemented?

Trusted Computing Base (TCB) Anchor

- Access control list (ACL)
  - Store column of matrix with the resource.
- Capability
  - Allow user to hold a "ticket" for each resource.
  - Store row of matrix with the user.

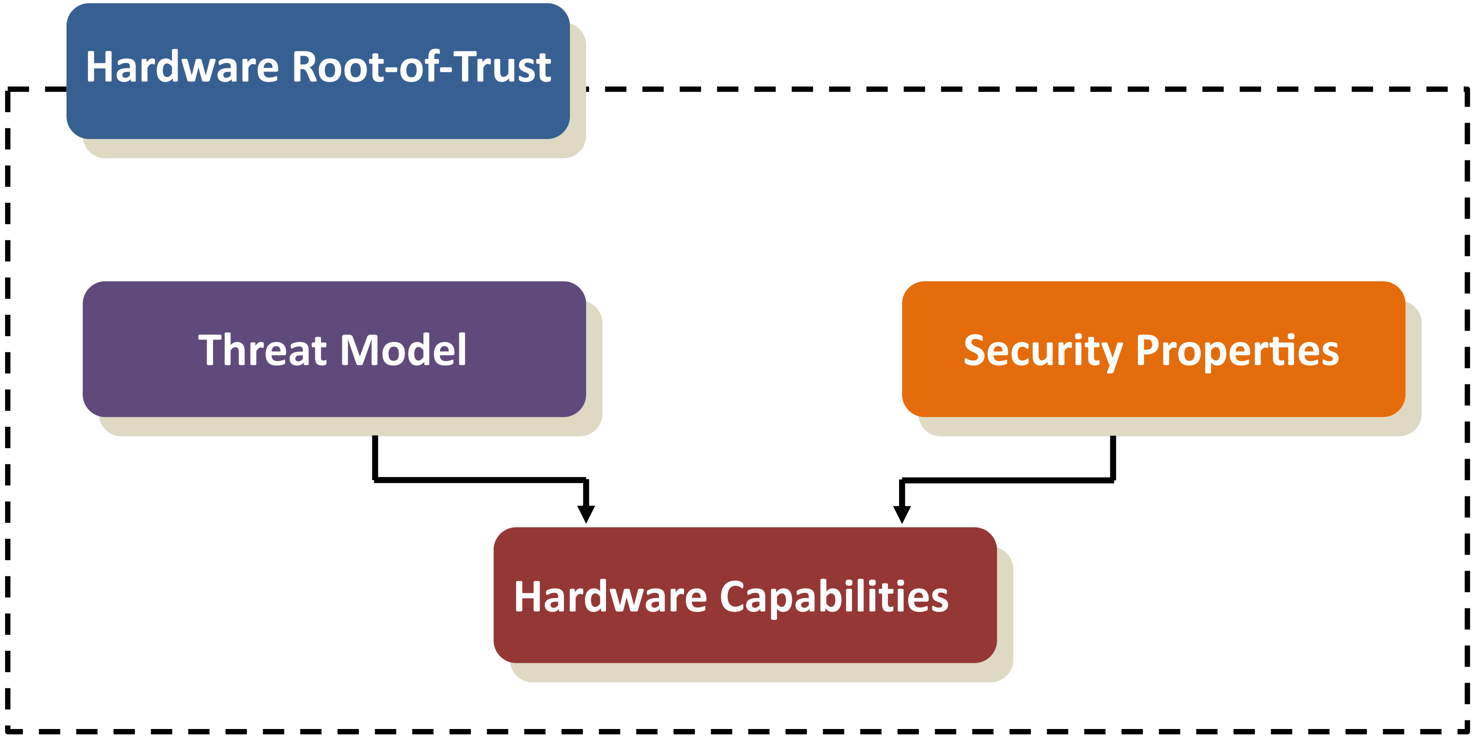
	File 1	**File 2**	**File 3**	**…**
User 1	read/write	write	-	-
User 2	write	read	read	-
User 3	-	-	write	read
…				
User n	write	read	write	-

Access Control List vs. Capability

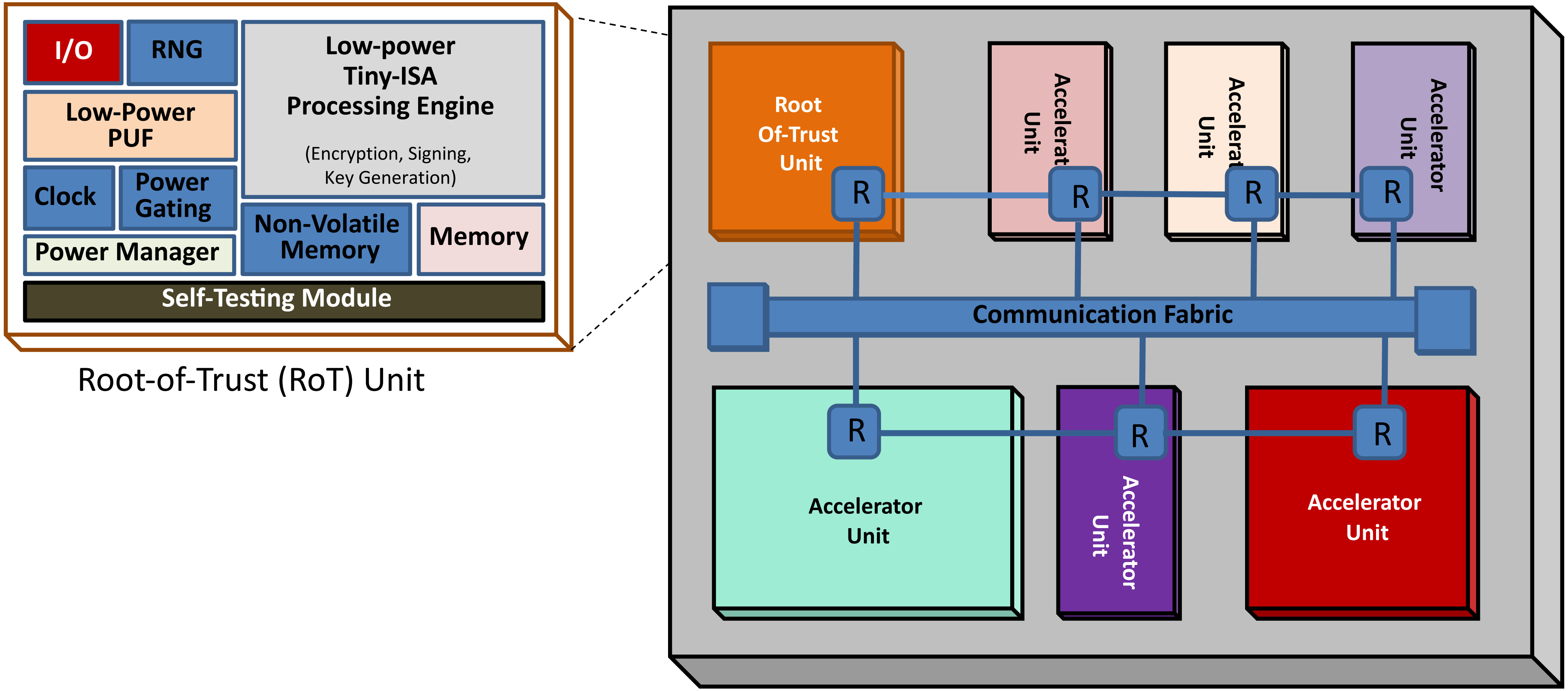
- Access control list
  - Associates list with each object
  - Checks user/group against list
  - Relies on authentication
- Capability
  - Is unforgeable "ticket"
  - Can be passed from one process to another
  - Checks ticket without requiring the identity of user/process

Hardware Root-of-Trust Properties

- Proof of authenticity and/or provenance
  - E.g., Uniquely identifiable and verifiable features
    - Physically Unclonable functions (PUFs)
- Immutable hardware component(s)
  - E.g., Anti-tamper features
- Anchor trust against a specific threat model



A RISC-V Hardware Root-of-Trust Design



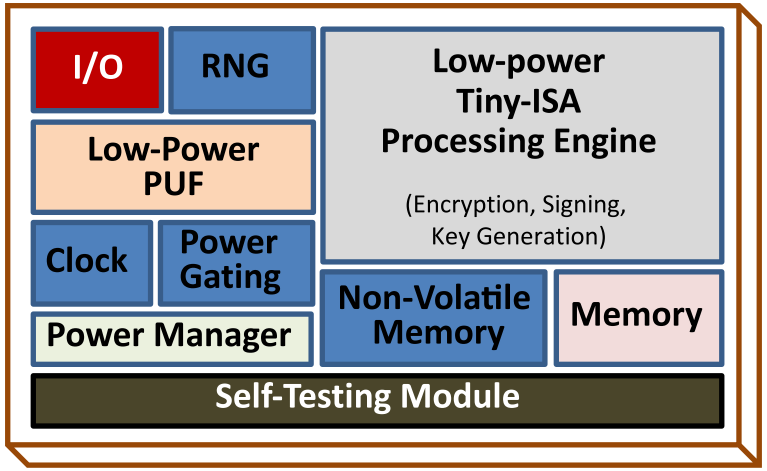
Root-of-Trust (RoT) Unit

SoC Chip

A. Ehret, K. M. Gettings, B. R. Jordan Jr. and M. A. Kinsy: "A Hardware Root-of-Trust Design for Low-Power SoC Edge Devices". In the 2020 IEEE High Performance Extreme Computing Conference (HPEC), 2020. Outstanding Student Paper Award

A RISC-V Hardware Root-of-Trust Design

- RoT unit is always on, although can go into a dormant state where its power is very minimal
  - The RoT unit does self-checking of its microcode
  - Non-volatile memory stores the secure boot sequence
  - The RoT also performs pre-boot and post-boot states checking
  - Runtime controlled access management
  - Auditing through security-related events logging and checking



Root-of-Trust (RoT) Unit

Upcoming Lectures

- Digital Design & Hardware Trojans
- Anti-tamper & Physically Unclonable Functions (PUFs)