상호 프라이버시를 향한 AI 모델 보호 기술

A Systematic Look at Mutual Privacy in AI Model Usage

2026 차세대 저작권보호기술 워크숍

서울대학교 컴퓨터연구소

유 준 승

Homepage

Biography

- **Education and Experience**
  - 2014 – 2019 B.E. @Department of Electrical and Computer Engineering, SNU
  - 2019 – 2026 Ph.D. @Department of Electrical and Computer Engineering, SNU
  - 2026 – Post-Doc[1] @Institute of Computer Technology, SNU

- **Research Interests –** *Hardware-assisted System Security*
  - **Keywords:** confidential computing, memory safety, compartmentalization, …
  - **Platforms:** cloud, mobile, autonomous vehicles, …
  - **Tools:** hardware extensions, compiler, hypervisor, kernel, …
  - **Applying to: System Security for AI (and *Mutual Privacy*)** **This talk!**

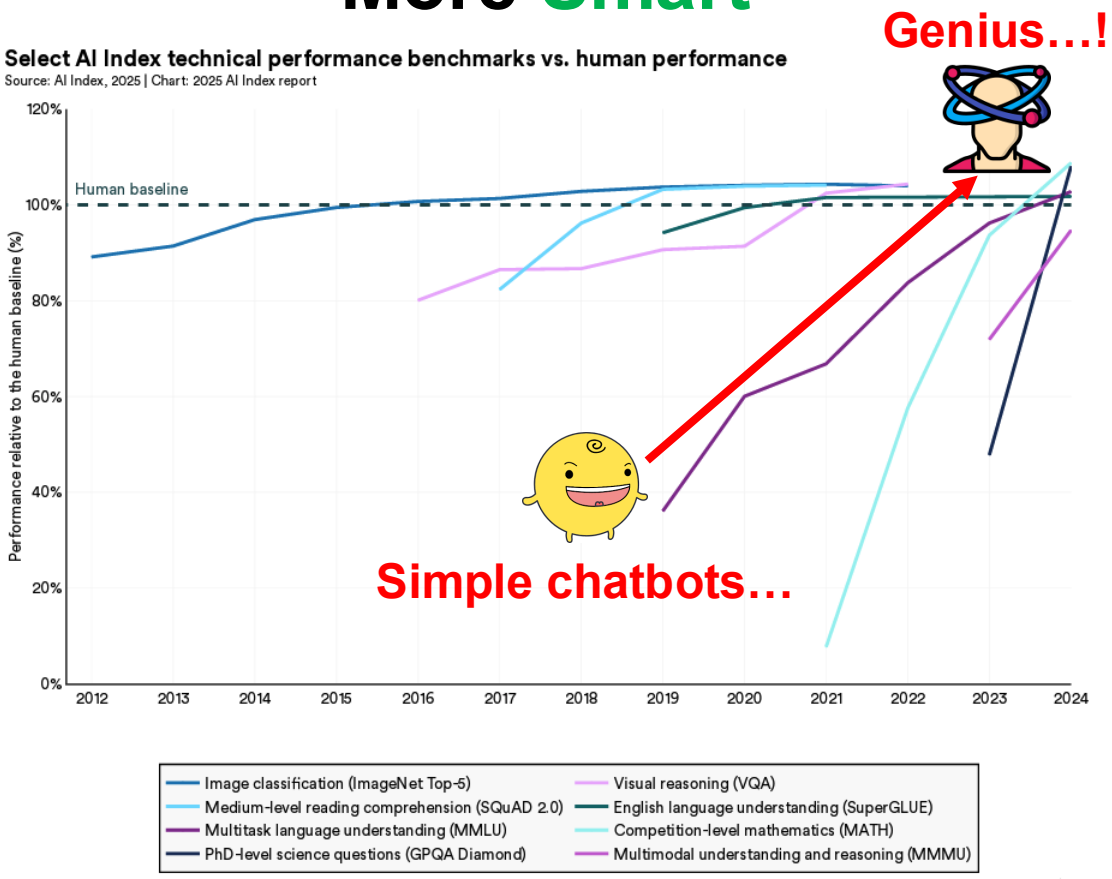
[1] *Serving mandatory military service as expert research personnel (~ 2027.8)*

Contents

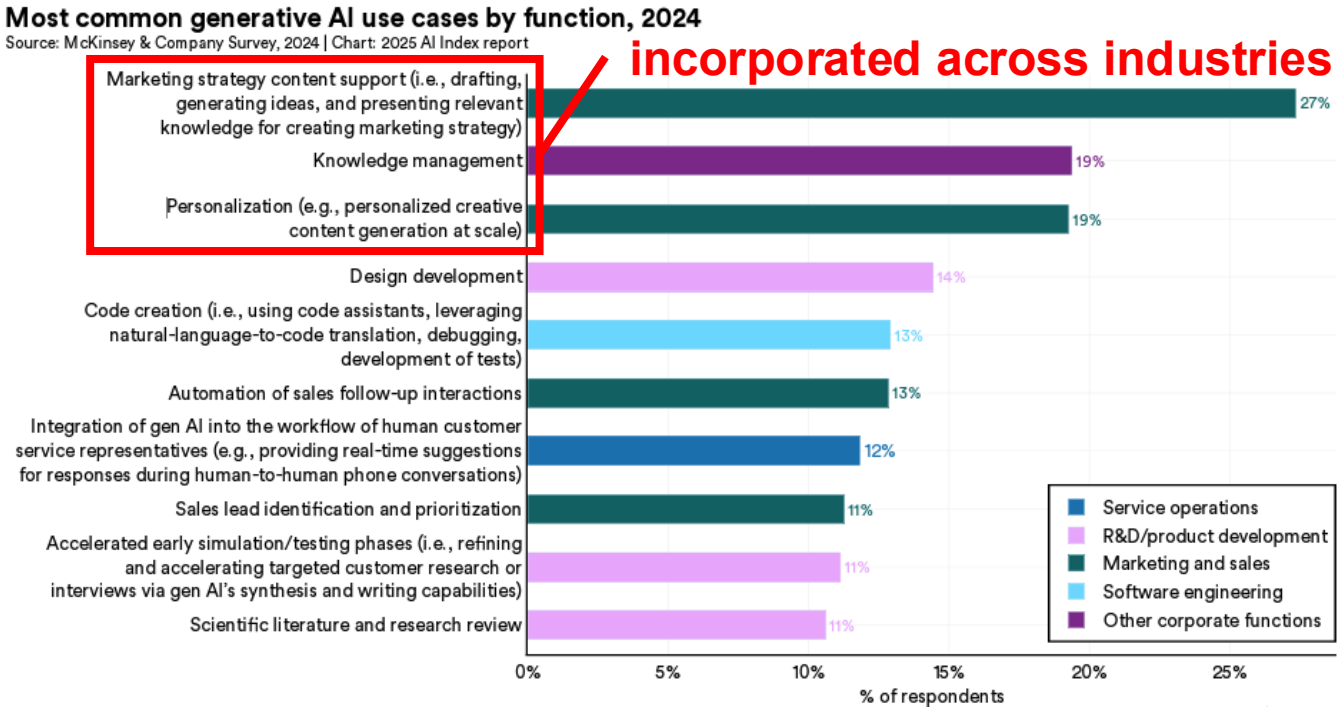
Contents

Evolution of AI

More Smart



More Utilization



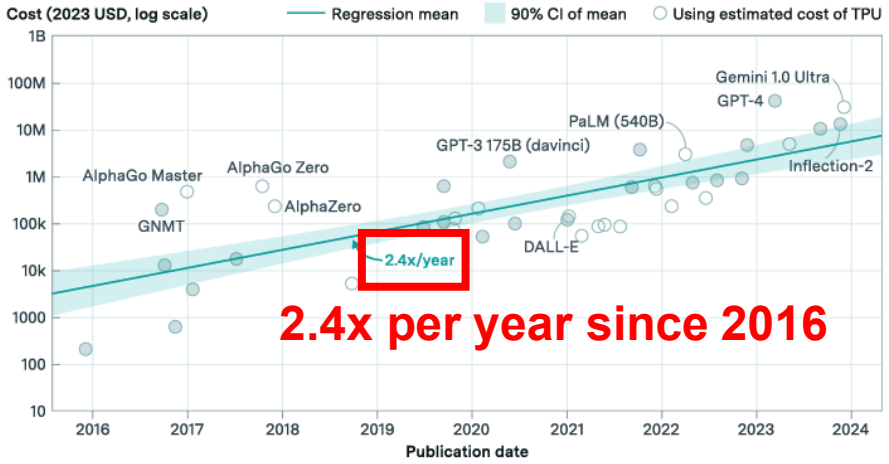
Stanford Institute for Human-Centered Artificial Intelligence. *Artificial Intelligence Index Report 2025*

Cost of AI Development and Usage

Model Providers

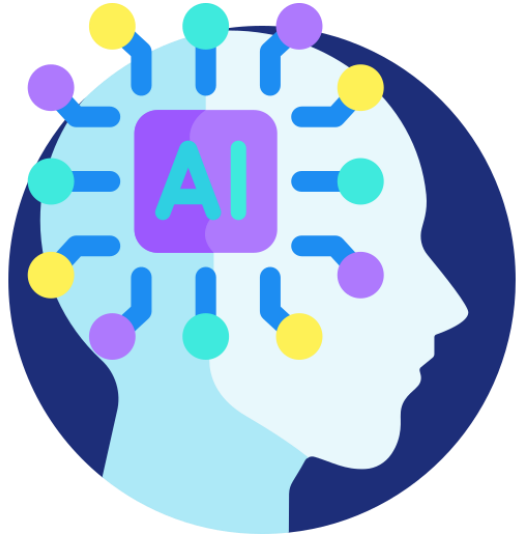
Higher costs

- increased *training data*
- increased *hardware* utilization
- increased *resource* utilization



2.4x per year since 2016

Stanford Institute for Human-Centered Artificial Intelligence.
Artificial Intelligence Index Report 2025



Frontier
AI Models/Services

Model Users

More sensitive information to handle complex/critical jobs

Financial information

Medical information

Business information

Mutual Privacy in AI Model Usage

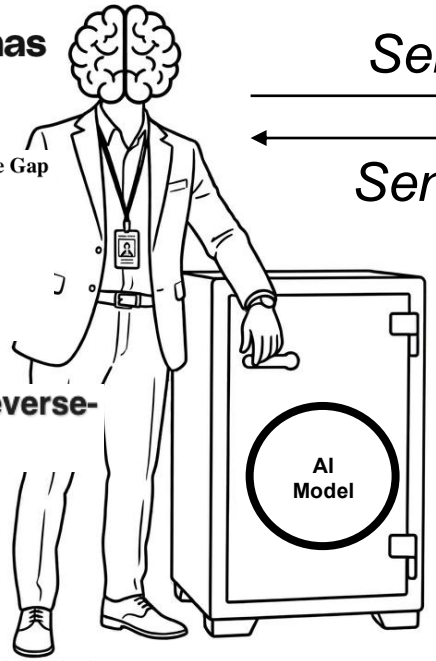
- *"Information wants to be free. Information also wants to be expensive"*
- Neither model provider nor user wants to give up their sensitive data to others

Meta's powerful AI language model has leaked online – what happens now?

SoK: All You Need to Know About On-Device ML Model Extraction - The Gap Between Research and Practice

Tushar Nayan* *Florida Intl. University* | Qiming Guo* *Florida Intl. University* | Mohammed Al Duniawi *Florida Intl. University*
Marcus Botacin *Texas A&M University* | Selcuk Uluagac *Florida Intl. University* | Ruimin Sun *Florida Intl. University*

Apple's NeuralHash Algorithm Has Been Reverse-Engineered



*Send me the **data***

*Send me the **model***

AI, NEWS, PRIVACY

Millions of (very) private chats exposed by two AI companion apps

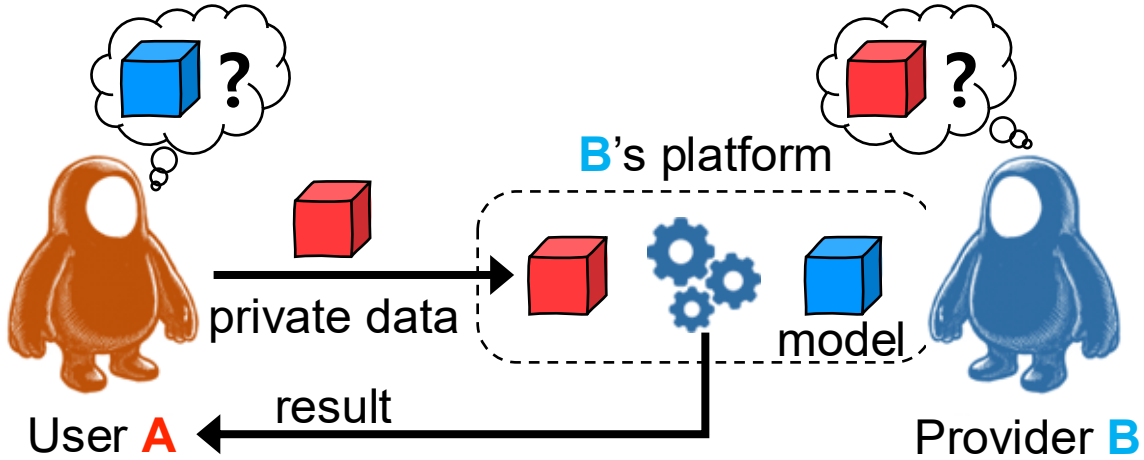
Elon Musk's xAI Published Hundreds Of Thousands Of Grok Chatbot Conversations

구글 인공지능 음성비서에 녹음된 대화 1천여건 유출

- /www.theverge.com/2023/3/8/23629362/meta-ai-language-model-llama-leak-online-misuse
- www.usenix.org/system/files/sec24fall-prepub-2341-nayan.pdf
- www.yna.co.kr/view/AKR20190712012100091
- techcrunch.com/2025/08/20/thousands-of-grok-chats-are-now-searchable-on-google
- www.schneier.com/blog/archives/2021/08/apples-neuralhash-algorithm-has-been-reverse-engineered.html
- www.malwarebytes.com/blog/news/2025/10/millions-of-very-private-chats-exposed-by-two-ai-companion-apps

Send the Data, Protect the Data

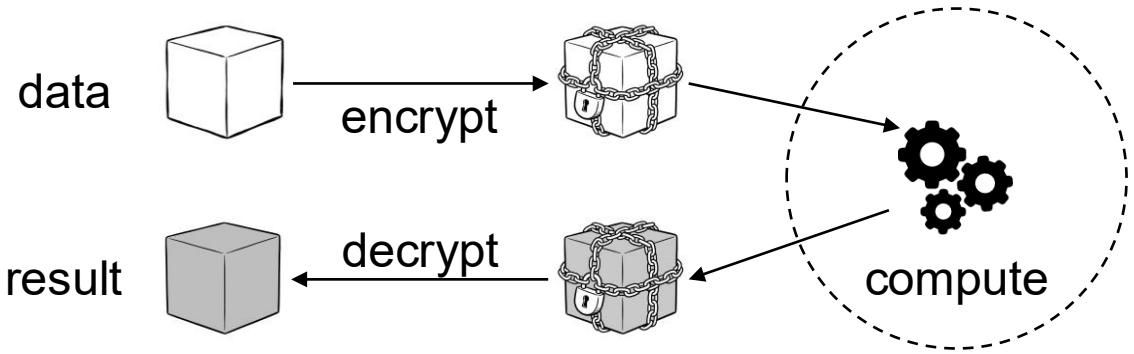
- **Privacy Enhancing Techniques (PET)**
  - Enable users to share sensitive data without exposing personal or proprietary information
  - Recognized as key technologies by OECD, OSTP, NSTC, etc.
- **Representative PETs**
  - Homomorphic Encryption (HE)
  - Secure Multi-party Computation (SMPC)
  - Synthetic Data
  - Differential Privacy (DP)
  - Confidential Computing (CC)



Data-Transforming PETs

- Protect data privacy by using **transformed data** for computation

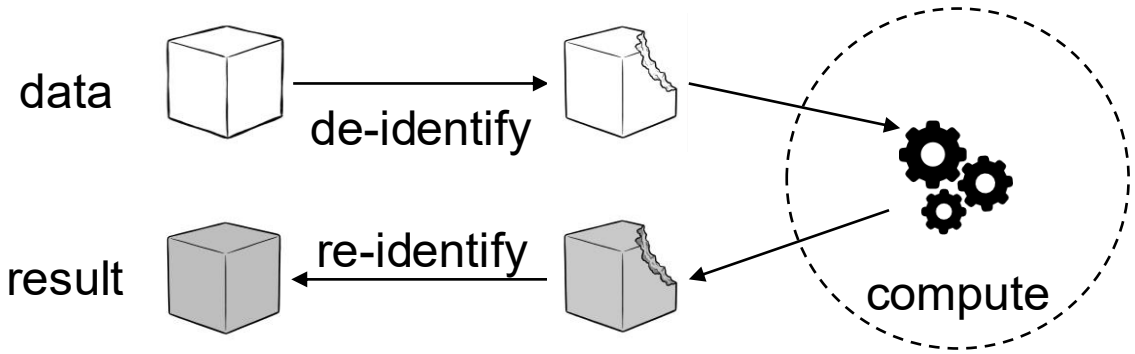
Homomorphic Encryption



Enables computation over encrypted data

1,000 ~ 10,000x[1] computation overhead compared to plaintext

De-identification

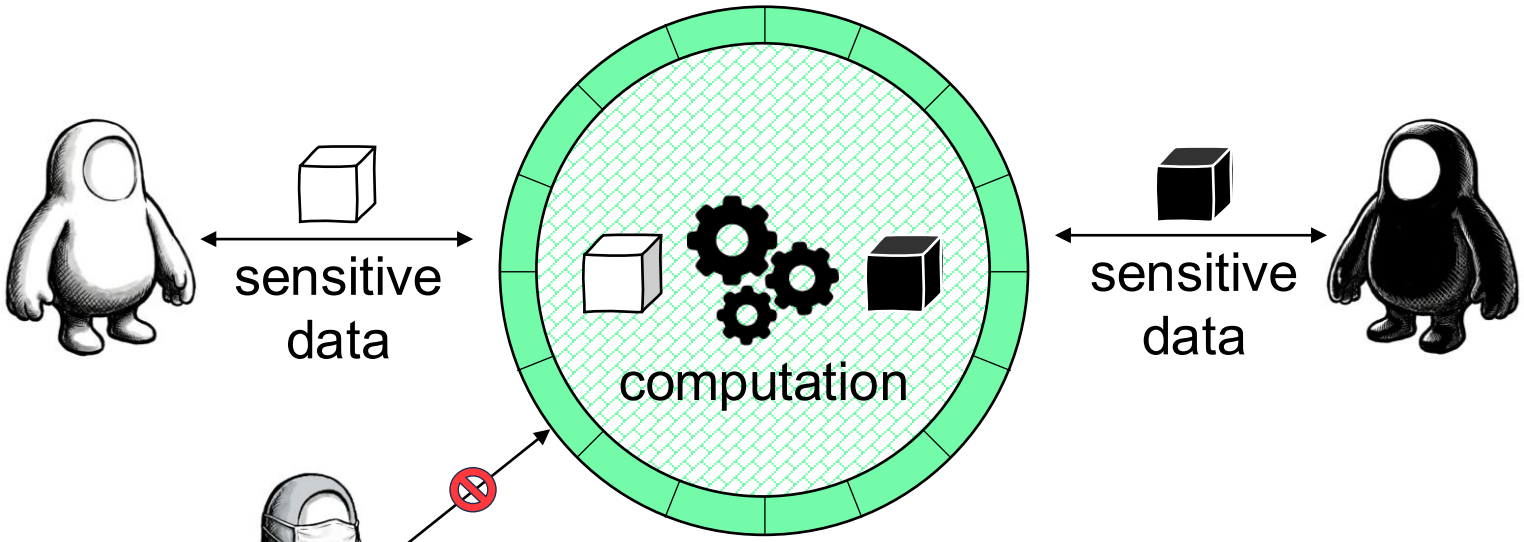


Transform original data to non-identifiable data

Accuracy degradation from using transformed data
Operational burden of de(re)-identification

Execution-Protecting PET

- Protect the **execution environment** where sensitive data is processed



Contents

1 What is mutual privacy?

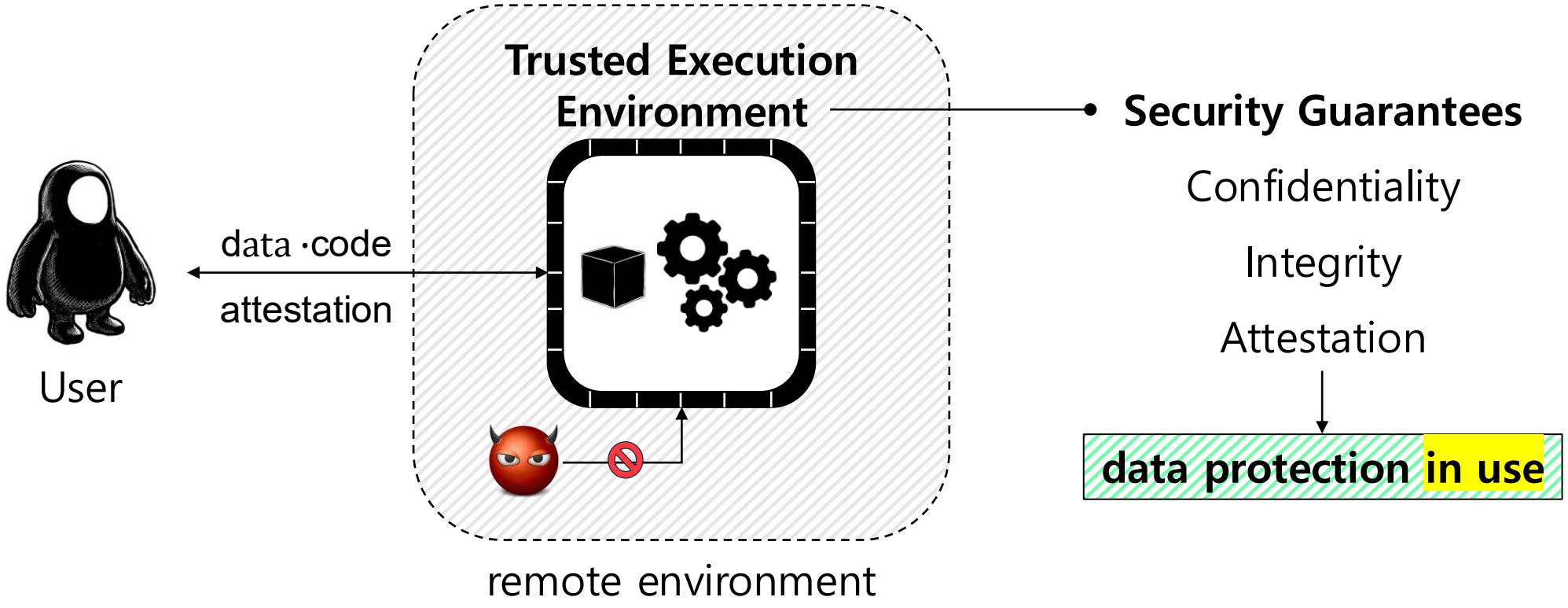
2 **What is trusted execution environments?**

3 Can TEEs solve mutual privacy?

4 How can TEEs solve mutual privacy?

Trusted Execution Environment (TEE)

- Personal **safe** in remote environment



Types of Trusted Execution Environments

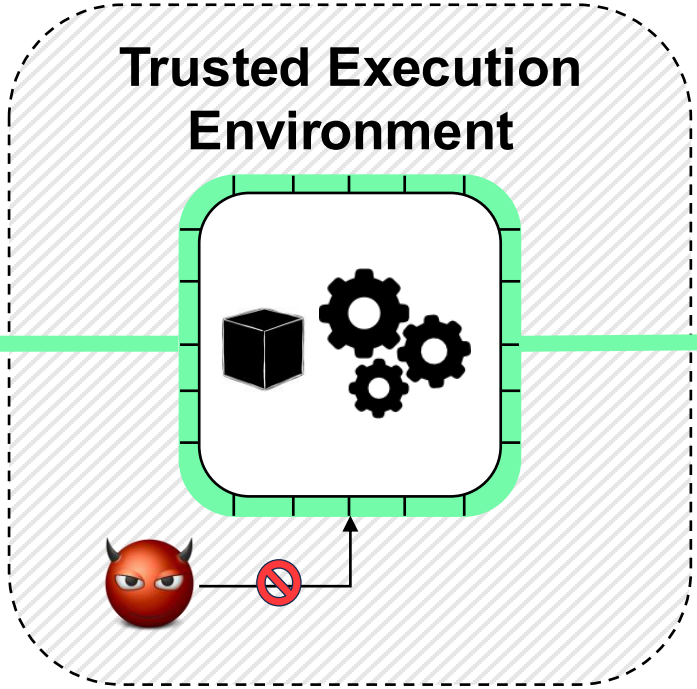
SW based TEEs

operating system, hypervisor

trust system SW with high privilege

no hardware dependency

large trusted computing base (TCB)



remote environment

HW based TEEs

hardware, processor(CPU)

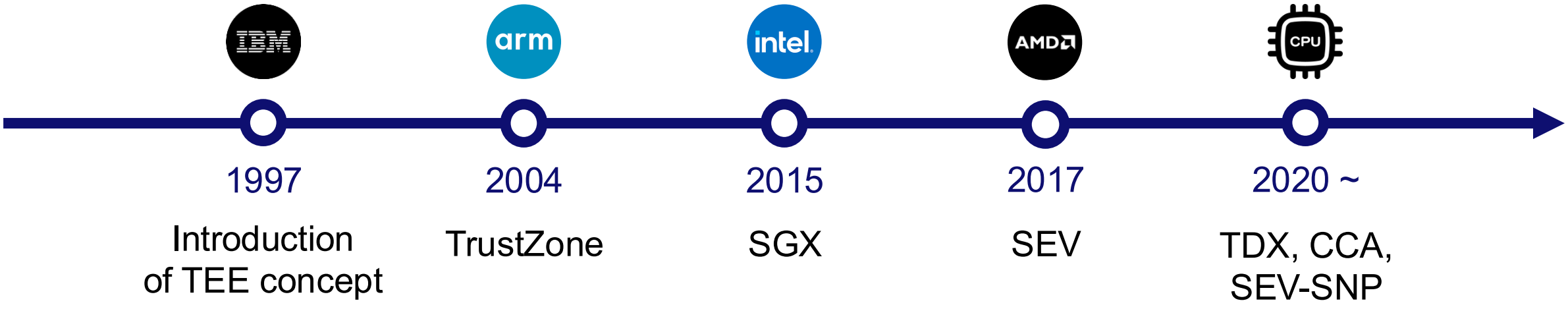
trust *only* the hardware

requires support from CPU vendor(s)

small TCB, **near-native** performance

"Keep the trusted base as simple and small as possible"

Evolution of HW-based TEEs



Evolution Trends

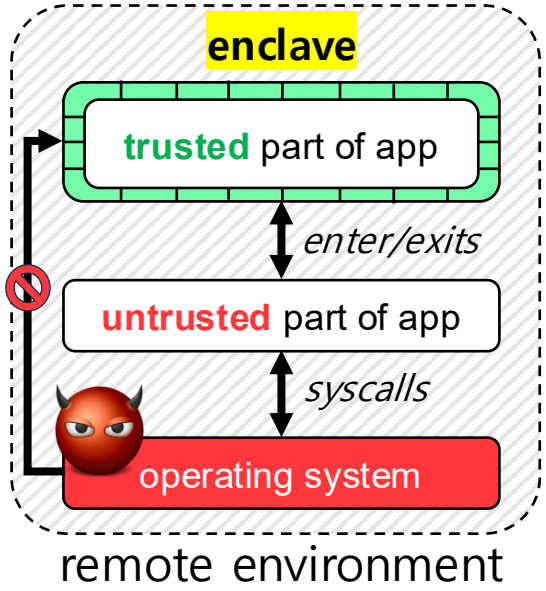
Mainstream Support – TEE extension and support from major CPU architectures / ISAs

Availability – TEE-based instances supported by major cloud platforms (e.g., AWS, Google)

Granularity of Protection – Larger protection scope from application-level to VM-level

Protection Boundary Expansion of TEEs

Application-level TEEs



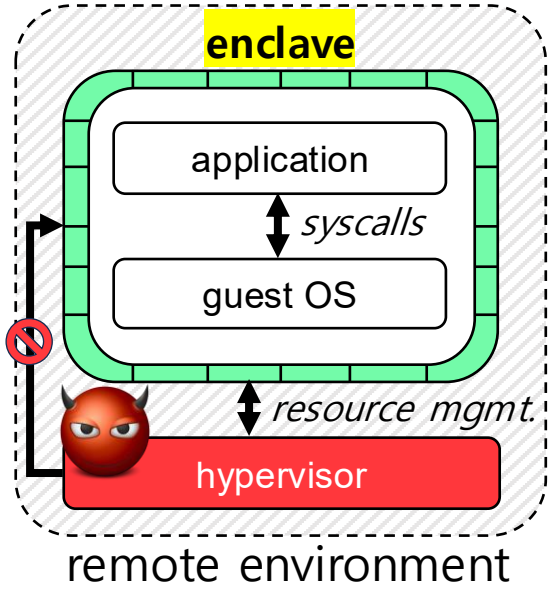
Intel SGX

designed to protect small, sensitive part of the app

small size (128MB), **high** overhead (~ 100x)

Virtual Machine (VM)-level TEEs

Intel TDX
AMD SEV
ARM CCA

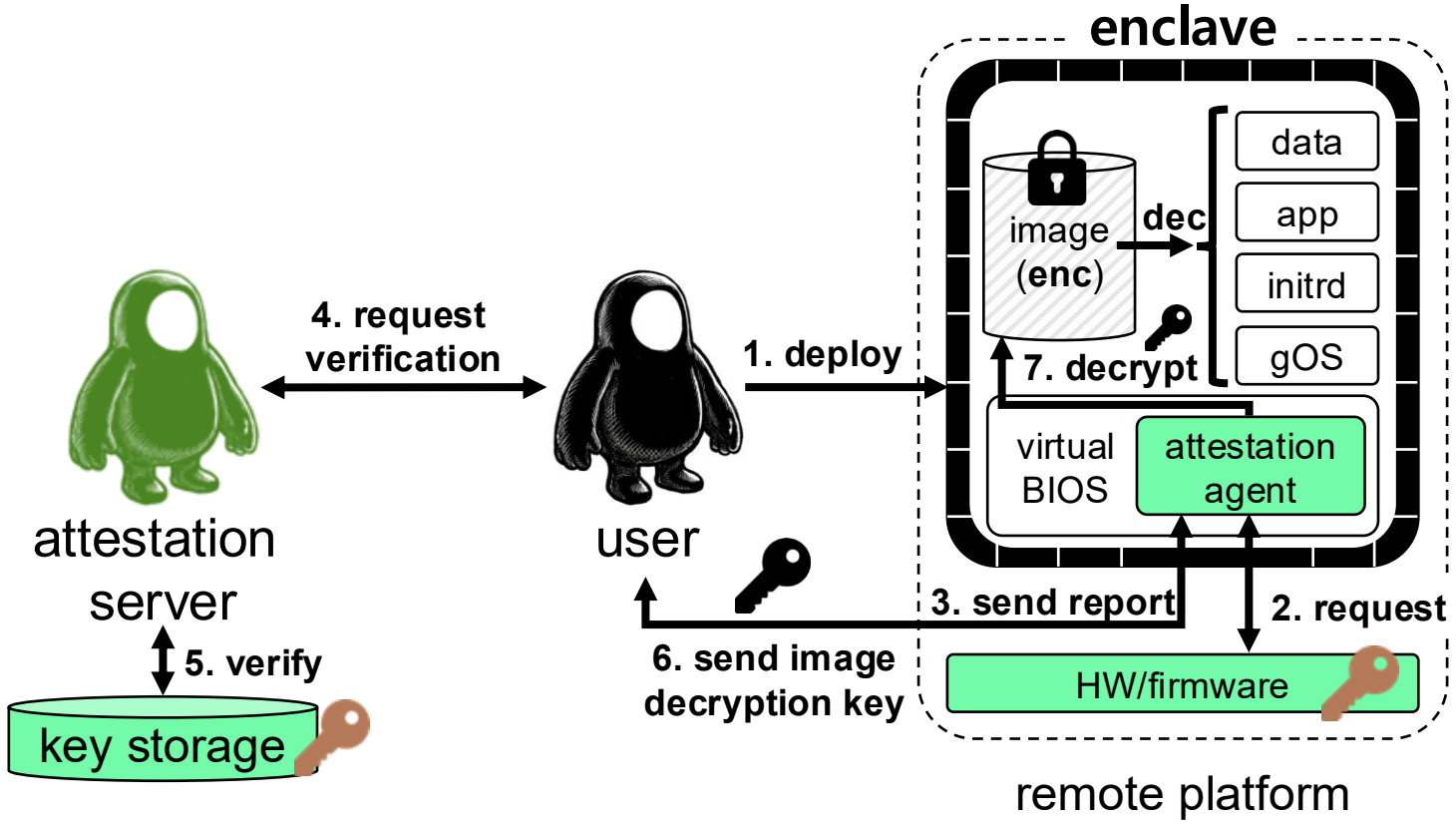


designed to protect the app at VM granularity

large size, **low** overhead (< 10%)

TEE Provisioning

- Hardware serves as **root-of-trust**



We want to **verify**

1. **Identity** of the application
2. **Validity** of the platform

- launch/boot measurements
- security policy/configuration
- platform TCB version
- signature/MAC

Contents

1 What is mutual privacy?

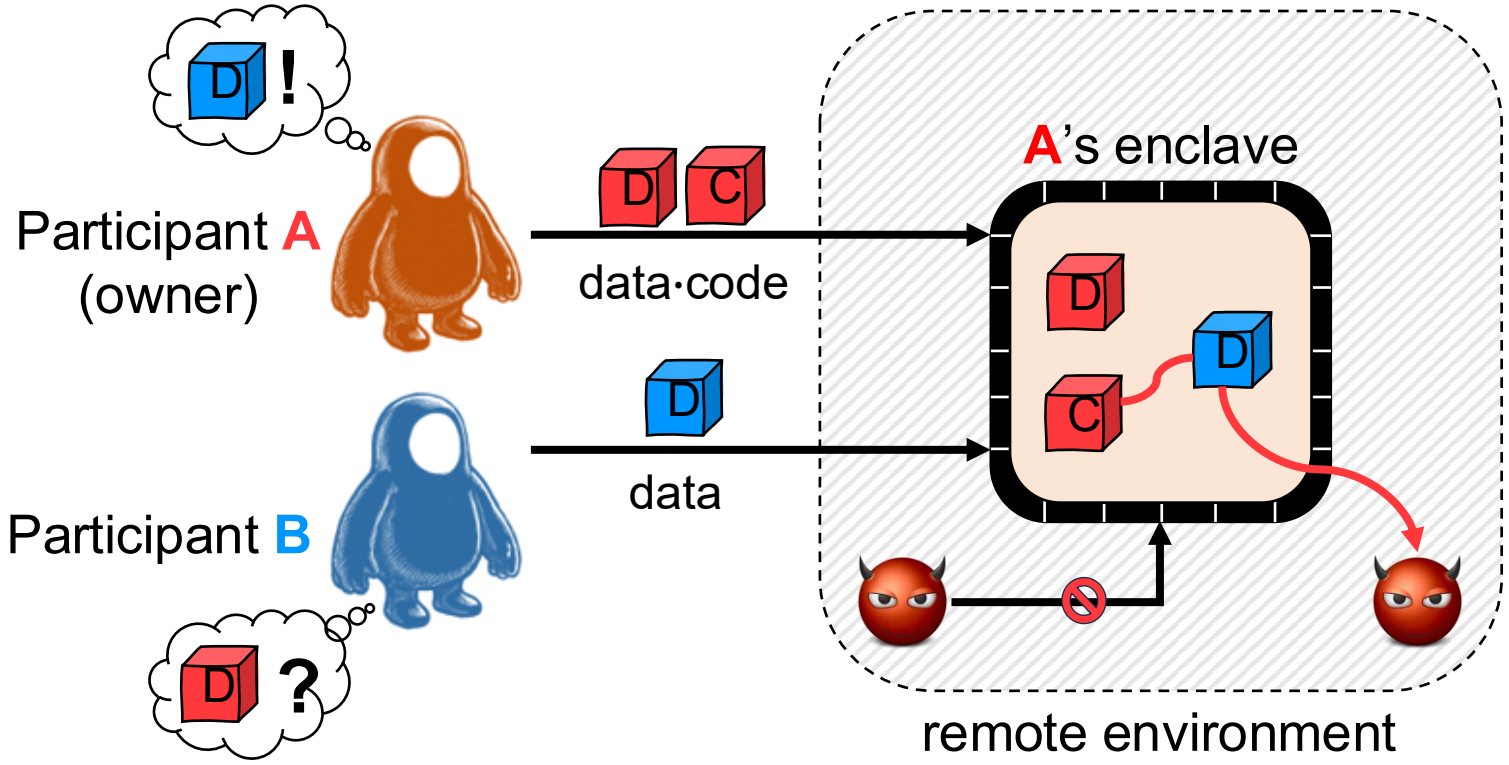
2 What is trusted execution environments?

3 **Can TEEs solve mutual privacy?**

4 How can TEEs solve mutual privacy?

Can TEEs Protect Mutual Privacy?

- TEEs are designed for **one-directional privacy**
  - Environment for single user (enclave owner) ➔ owner controls (initial) code/data within TEE

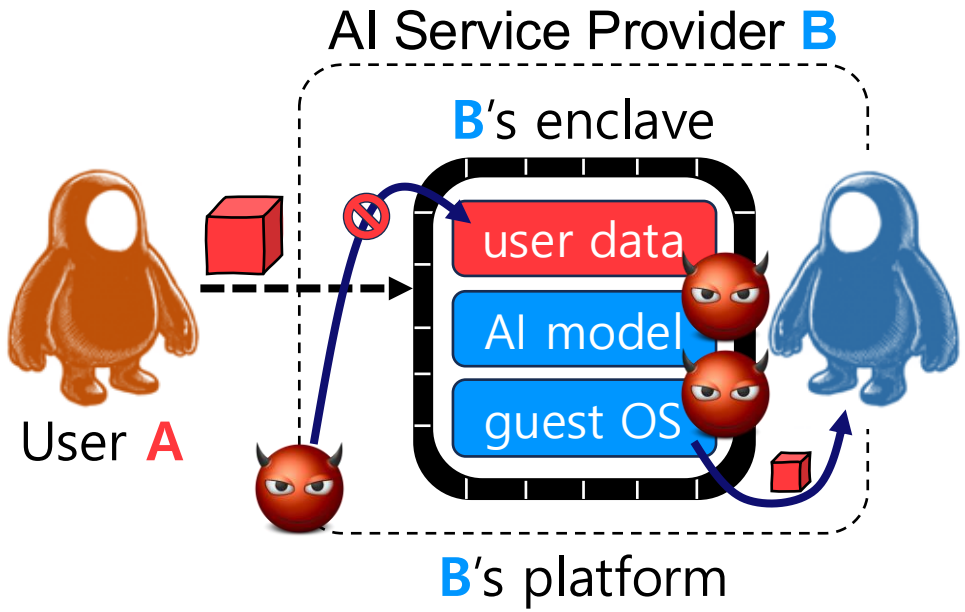


Cannot protect
mutual privacy

B's data sent to **A**'s enclave
left out of **B**'s control

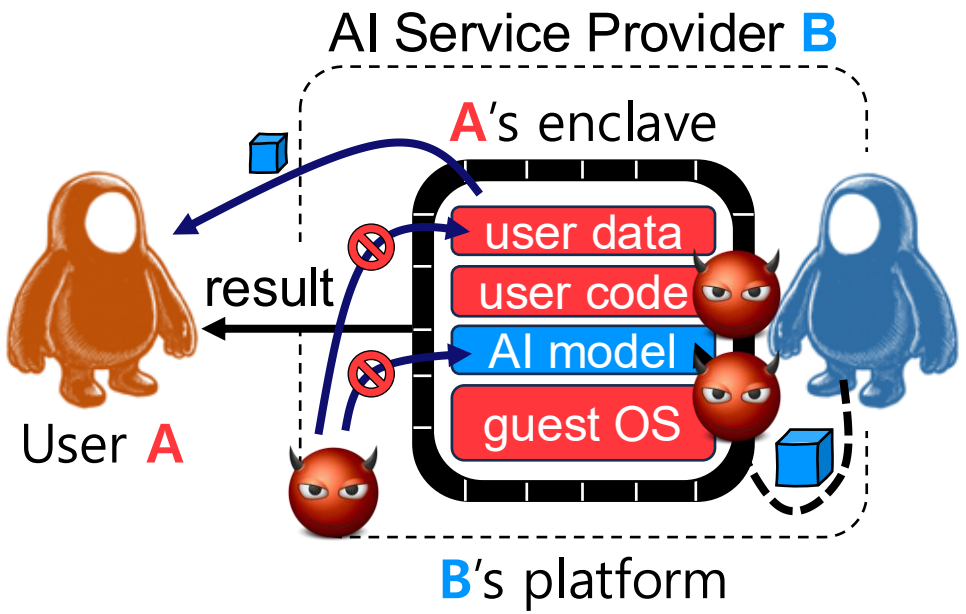
Sending User Data to Remote Platform

Scenario 1 – A's data to B's enclave



A's data privacy may be compromised

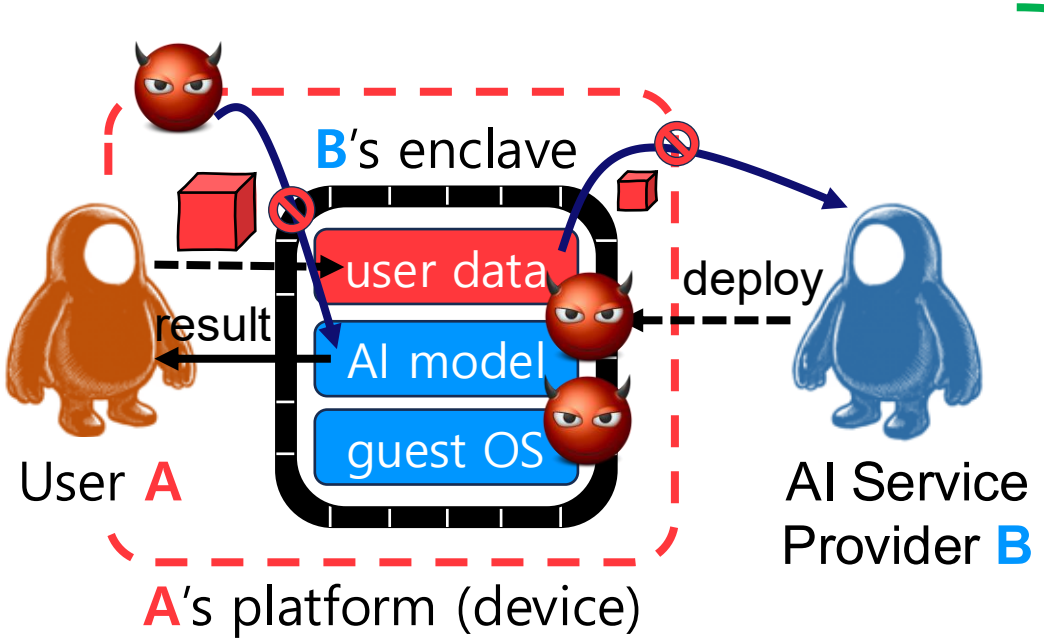
Scenario 2 – B's model to A's enclave



B's model privacy may be compromised

Sending AI Model to Remote Platform

- **Scenario** – **B**'s model to **B**'s enclave in **A**'s device



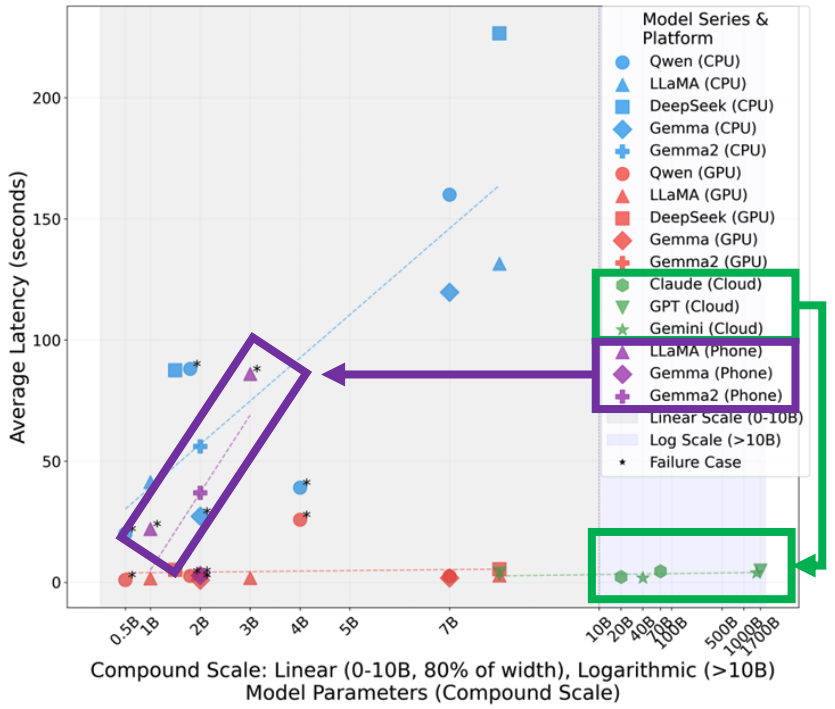
Secure On-device AI

B's model protected by enclave
A's data protected by restricting external communication from enclave

Limitations of (Secure) On-device AI

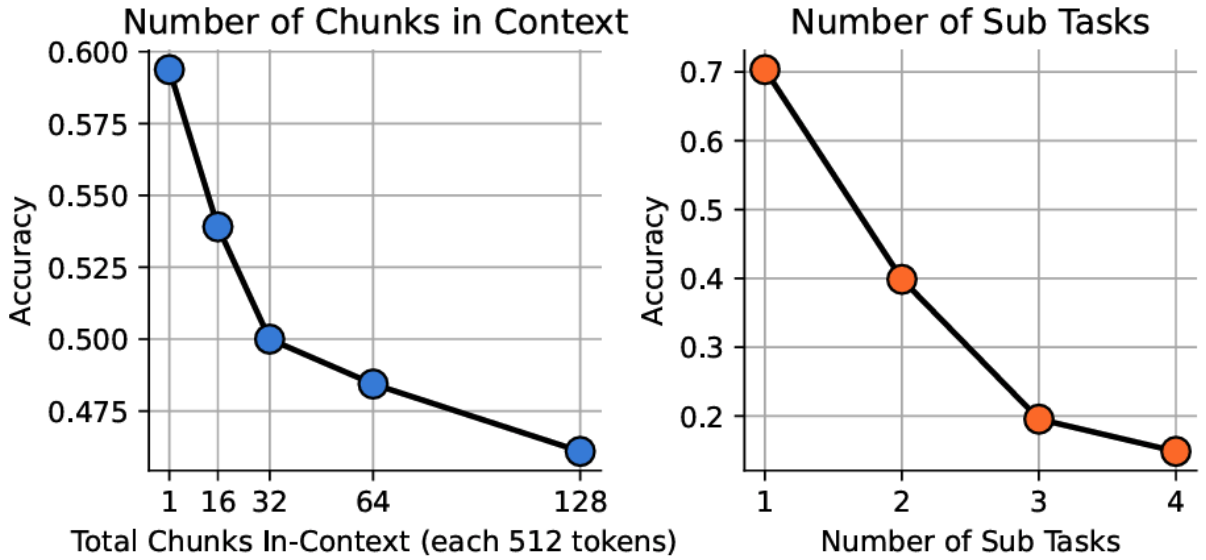
- Low quality of service (QoS) due to **limited** resources

High Latency



Model Size vs. Latency Comparison[1]

Low Accuracy



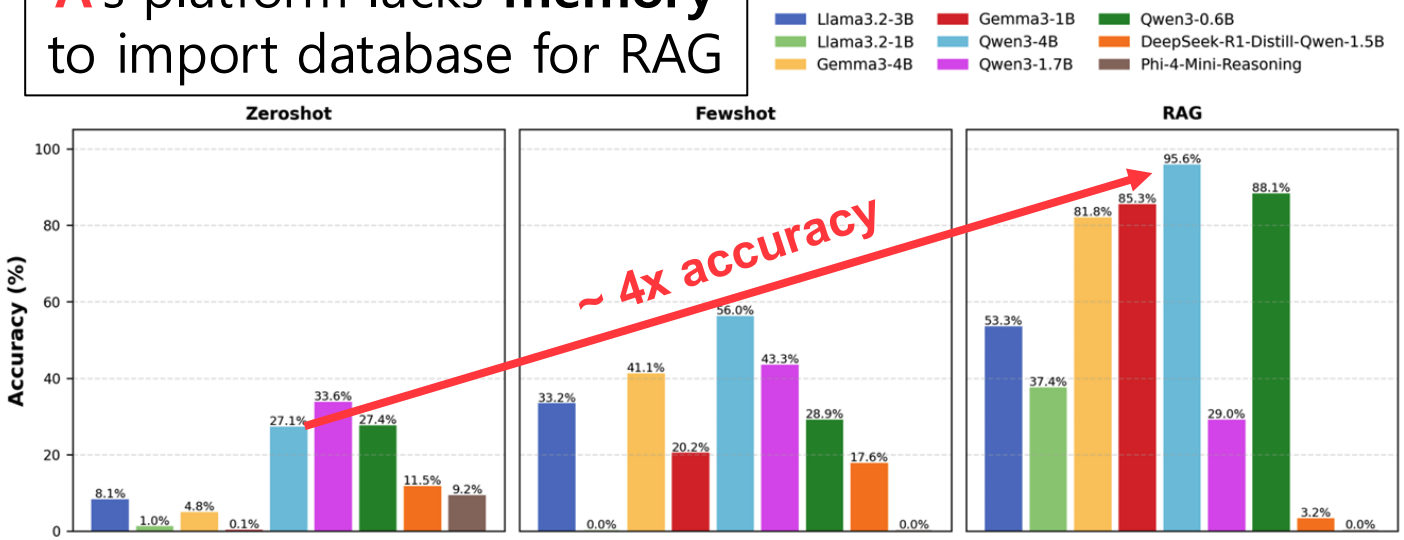
Accuracy Drop across Task Complexity[2]

1) Are We There Yet? A Measurement Study of Efficiency for LLM Applications on Mobile Devices
2) Minions: Cost-efficient Collaboration Between On-device and Cloud Language Models

QoS Comes at a Cost of Security

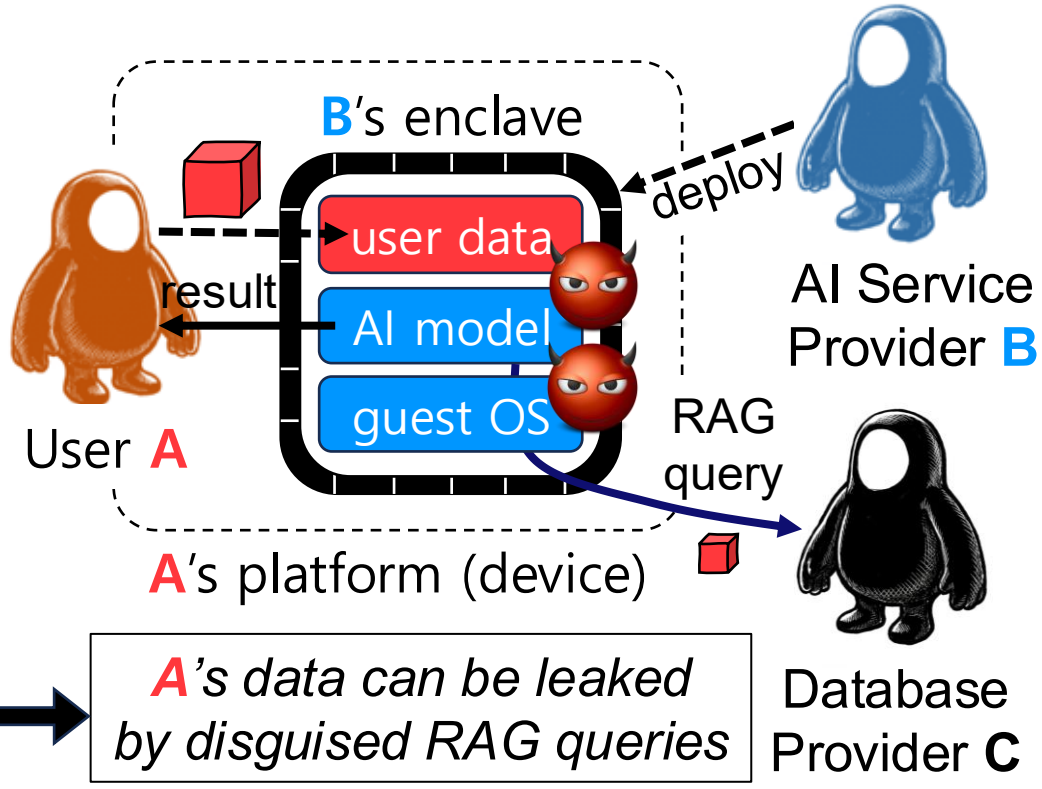
- **Retrieval-Augmented Generation (RAG)**
  - Requires **external communication** ➔ breaks TEE-based **mutual privacy** protection

A's platform lacks **memory** to import database for RAG



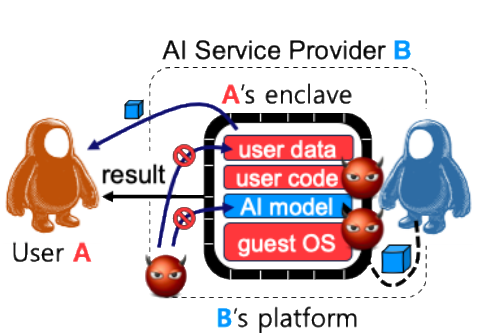
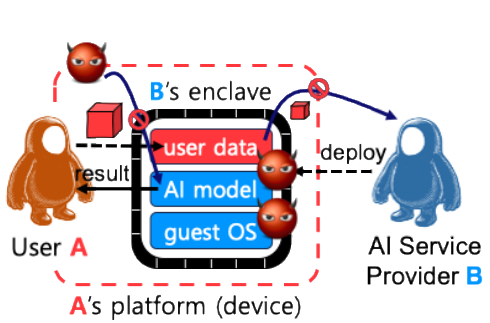
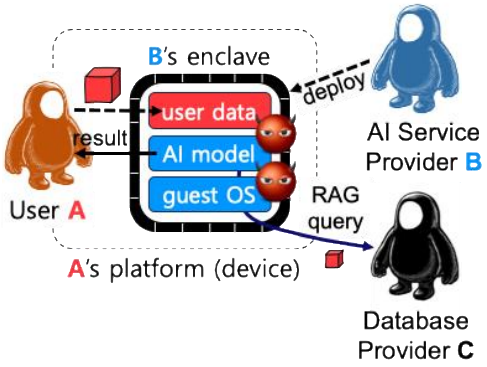
On-device AI Model Accuracy Improvement with RAG[1]

1) Benchmarking Small Language Models and Small Reasoning Language Models on System Log Severity Classification



A's data can be leaked by disguised RAG queries

Summary of TEE-based Mutual Privacy Protection

	Case 1	Case 2	Case 3	Solution
Sensitive Data Flow	A $\xrightarrow{data}$ B's platform	B $\xrightarrow{model}$ A's platform	B $\xrightarrow{model}$ A's platform	
Usage Model and Attack Vectors				?
Data Privacy	yes	yes	**no**	**yes**
Model Privacy	**no**	yes	yes	**yes**
Performance	high	**low**	high	**high**

Contents

1 What is mutual privacy?

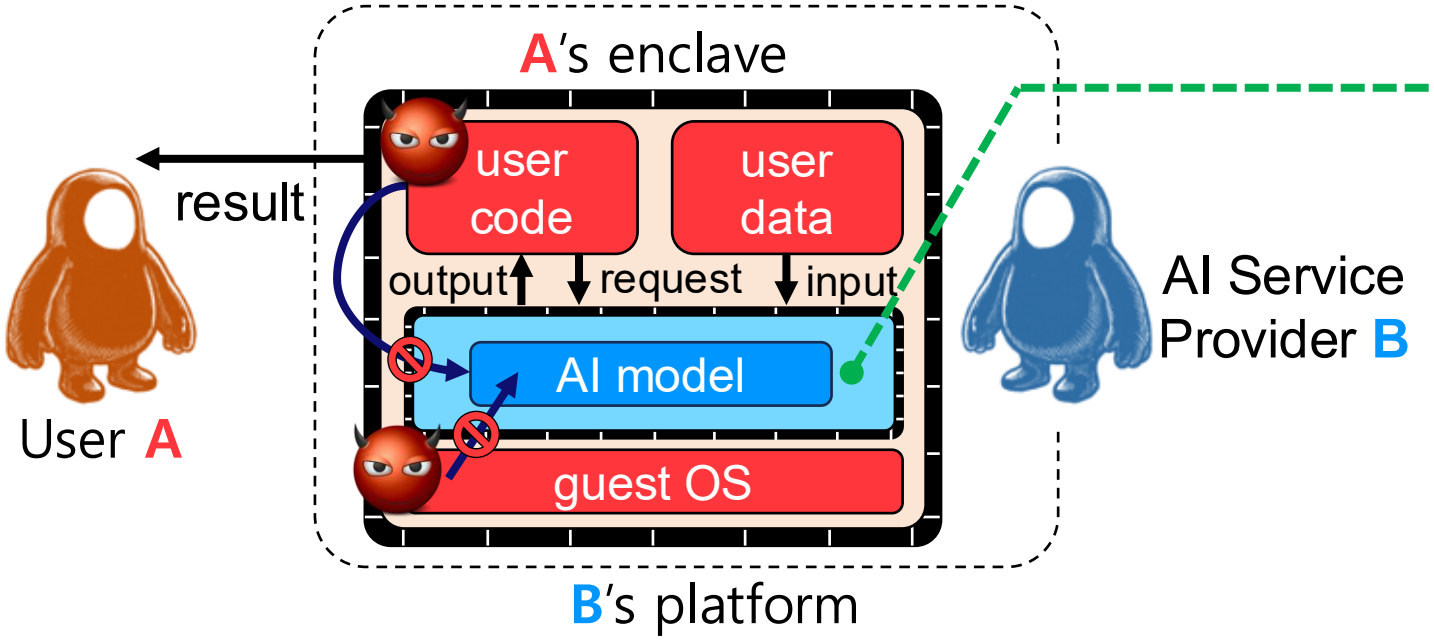
2 What is trusted execution environments?

3 Can TEEs solve mutual privacy?

4 **How can TEEs solve mutual privacy?**

Mutual Privacy Protection with Nested Enclaves

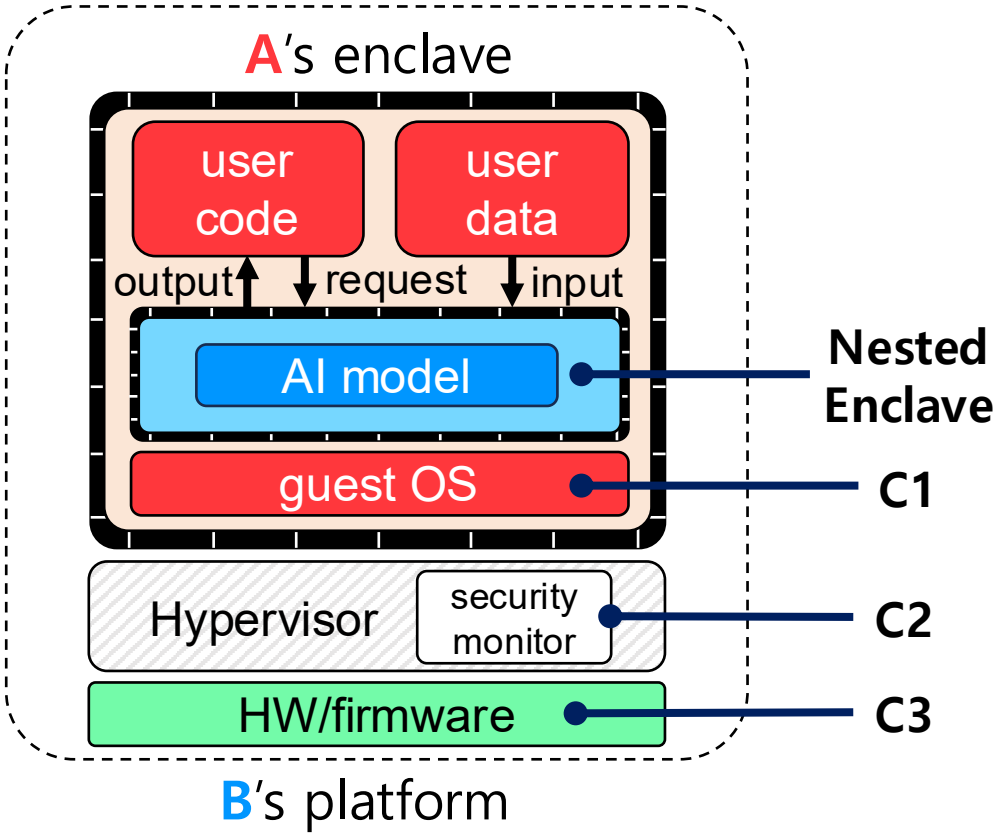
- **(Performance)** send user (**A**) data to AI service provider (**B**)
- **(Data Privacy)** protect **A**'s data using **A**'s TEE on **B**'s platform
- **(Model Privacy)** protect **B**'s model using **nested enclave** *inside* **A**'s TEE



Combine VM-level protection with application-level isolation

Protect selected applications from the guest OS and other applications

Building Nested Enclaves in TEEs



Challenge 1 – Omnipotent Guest OS

Guest OS has full control over memory access permissions of the software running inside CVM

Challenge 2 – Untrusted Hypervisor

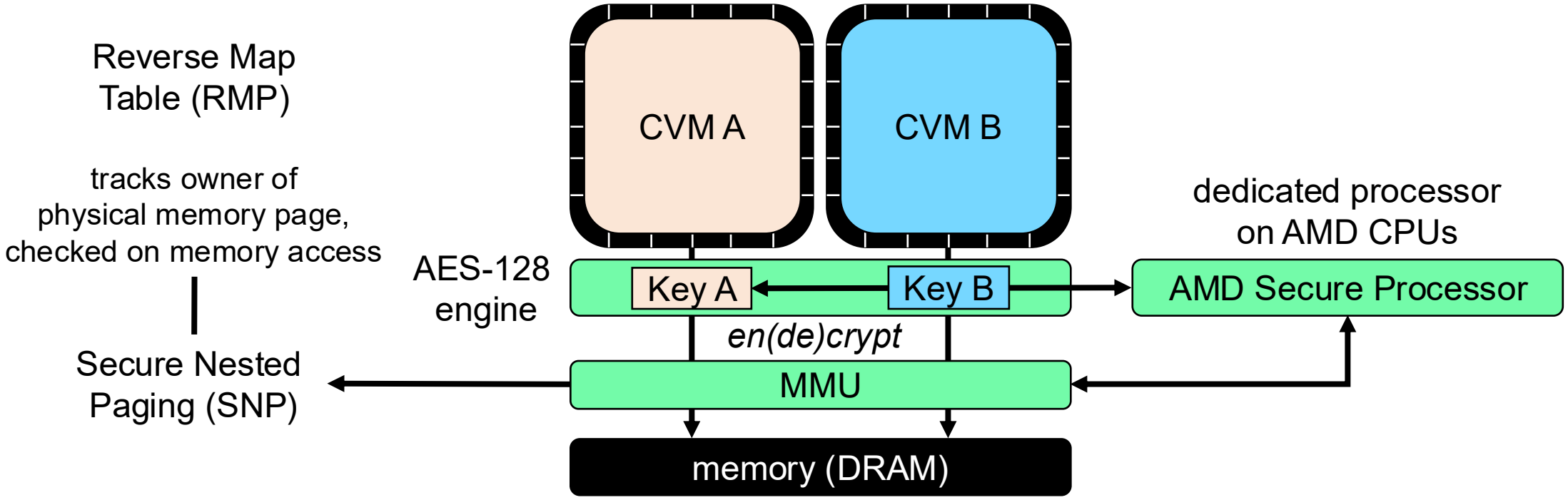
Any system component outside CVM is untrusted, cannot deploy a higher-privileged security monitor

Challenge 3 – Different Architectures

Under the hood of TEE, different architectures have different mechanisms and extensions to support TEE.

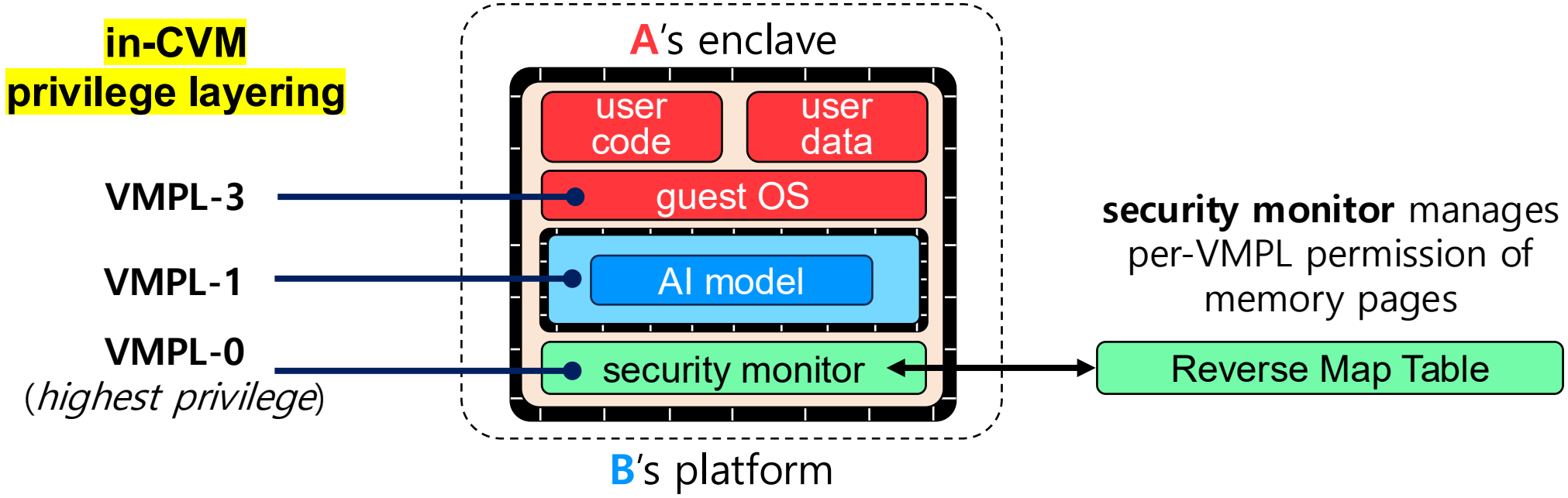
Nested Enclave on AMD (1)

- Secure Encrypted Virtualization (SEV)
  - hardware (AES encryption engine) en(de)crypts enclave data on memory
  - per-CVM encryption key managed by Secure Processor (SP)



Nested Enclave on AMD (2)

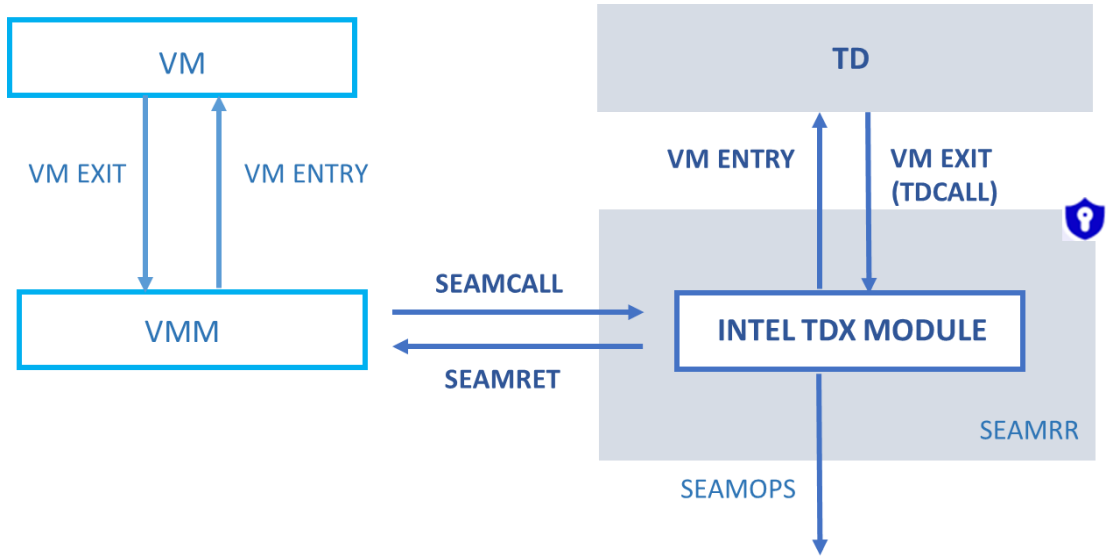
- Virtual Machine Privilege Level (VMPL)
  - hardware extension supported by SEV-SNP ➔ vCPUs execute in different **privilege levels**
  - higher VMPLs can control access of lower VMPLs (memory page permissions)



Nested Enclave on Intel (1)

- Trusted Domain Extensions (TDX)
  - **Secure-Arbitration Mode (SEAM)**
    - new processor execution mode
    - only allows access to predefined range
    - new instructions (SEAMCALL, SEAMRET)
  - **Intel TDX Module**
    - trusted hypervisor for TDs
    - handles interaction with untrusted VMM
  - **Multi-Key Total Memory Encryption**
    - hardware encryption engine similar to SEV
    - per-CVM encryption keys

High-level System Architecture of Intel TDX[1]

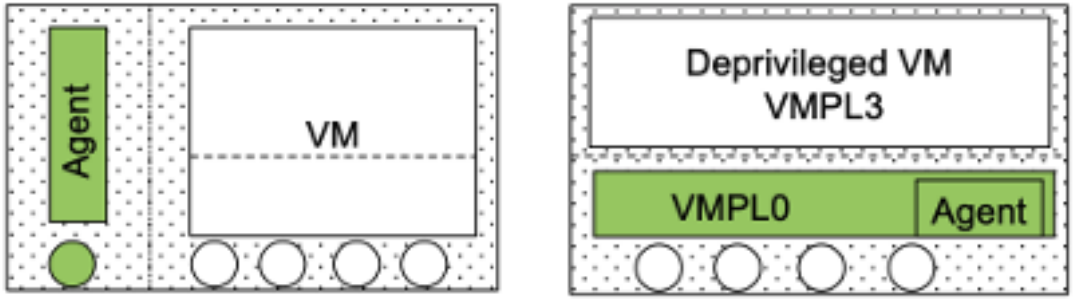


1) Intel, Trusted Domain Extensions on 4th-gen Xeon Processors

Nested Enclave on Intel (2)

- **Resource separation** for nested enclave applications
  - Nested enclave execution bound to specific vCPU
  - Memory used by enc-vCPU is isolated
  - Possible due to trusted hypervisor (TDX module) for TDX

Resource Separation Compared to In-CVM Privilege Layering[1]



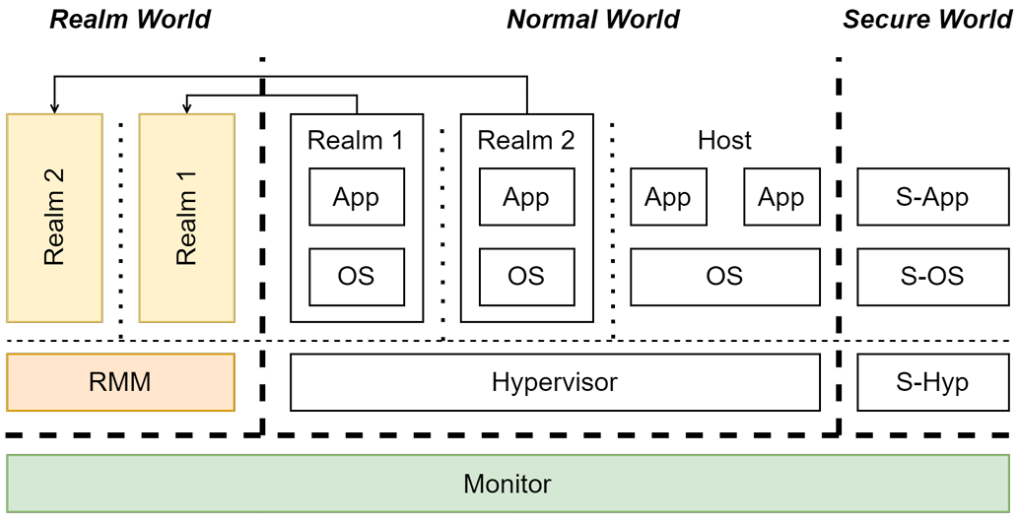
1) TETD: Trusted Execution in Trusted Domains

Nested Enclave on ARM

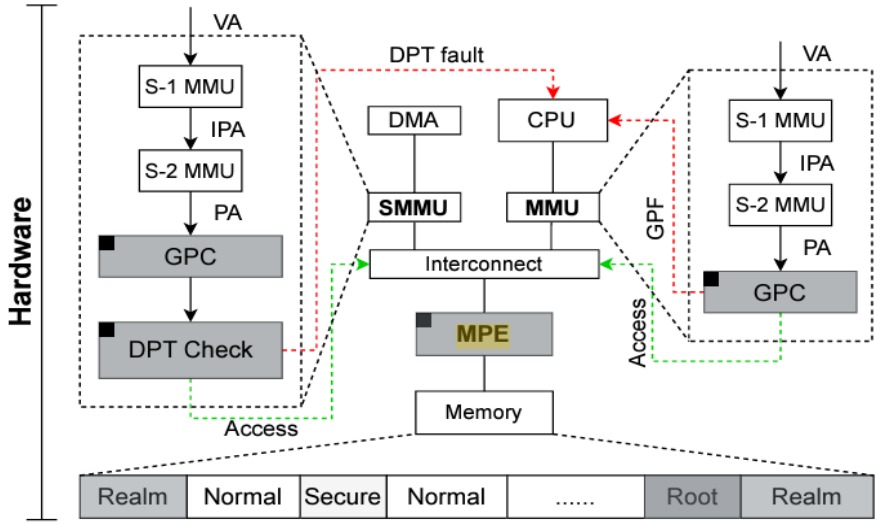
work-in-progress

- Confidential Compute Architecture (CCA)
  - **Realm** – new security state for CVM, memory page associated with physical address space
  - **RMM** – Realm Management Monitor, a trusted hypervisor for ARM CVM management
  - **GPT** – Granule Protection Table, checks CPU security state with physical address space

High-level System Architecture of ARM CCA[1]



Hardware Components for ARM CCA[1]



1) Arm, Confidential Compute Architecture

Summary

1

What is mutual privacy?

protecting **both** AI model and user data

2

What is trusted execution environments?

secure user **safe** in remote environment

3

Can TEEs solve mutual privacy?

No. TEEs are designed for one-directional privacy
(or sacrifice performance)

4

How can TEEs solve mutual privacy?

Nested enclaves. Design enclave inside TEE

Thank you

jsyou0224@gmail.com

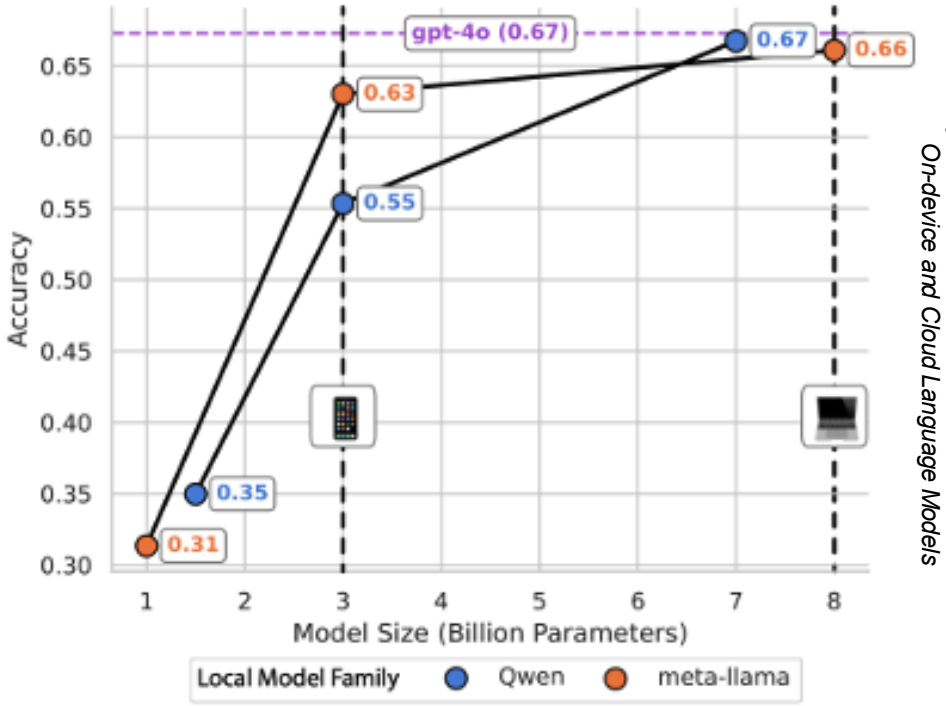
Homepage

CV

Is Secure On-device AI Enough?

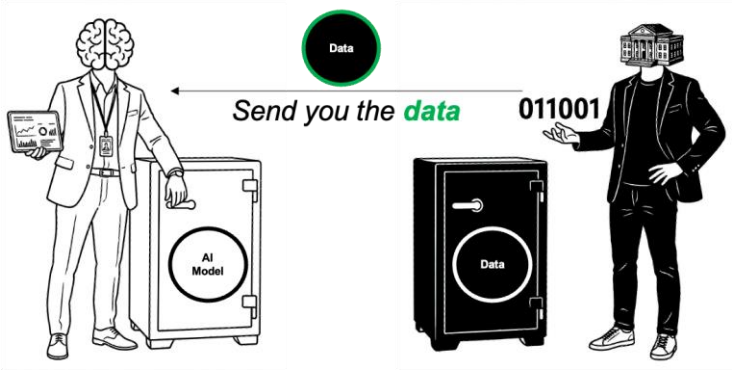
- (**Limitation 1**) Limited resources
  - Pixel 9 Pro: 16GB RAM vs. AWS P6-B200 1,440GB GPU memory
  - ~3,800x slower AI compute engine throughput
- (**Limitation 2**) Limited quality of service
  - Compact models due to limited device capacity
  - Often have to rely on RAG for better model output

    ➔ **need *external communication*!**



Narayan et. al., *Cost-efficient Collaboration Between On-device and Cloud Language Models*

Send the Data, Protect the Data



- **Privacy Enhancing Techniques (PET)**
  - Enable users to share sensitive data without exposing personal or proprietary information
  - Recognized as key technologies by OECD, OSTP, NSTC, etc.
- **Representative PETs**
  - Homomorphic Encryption (HE)
  - Secure Multi-party Computation (SMPC)

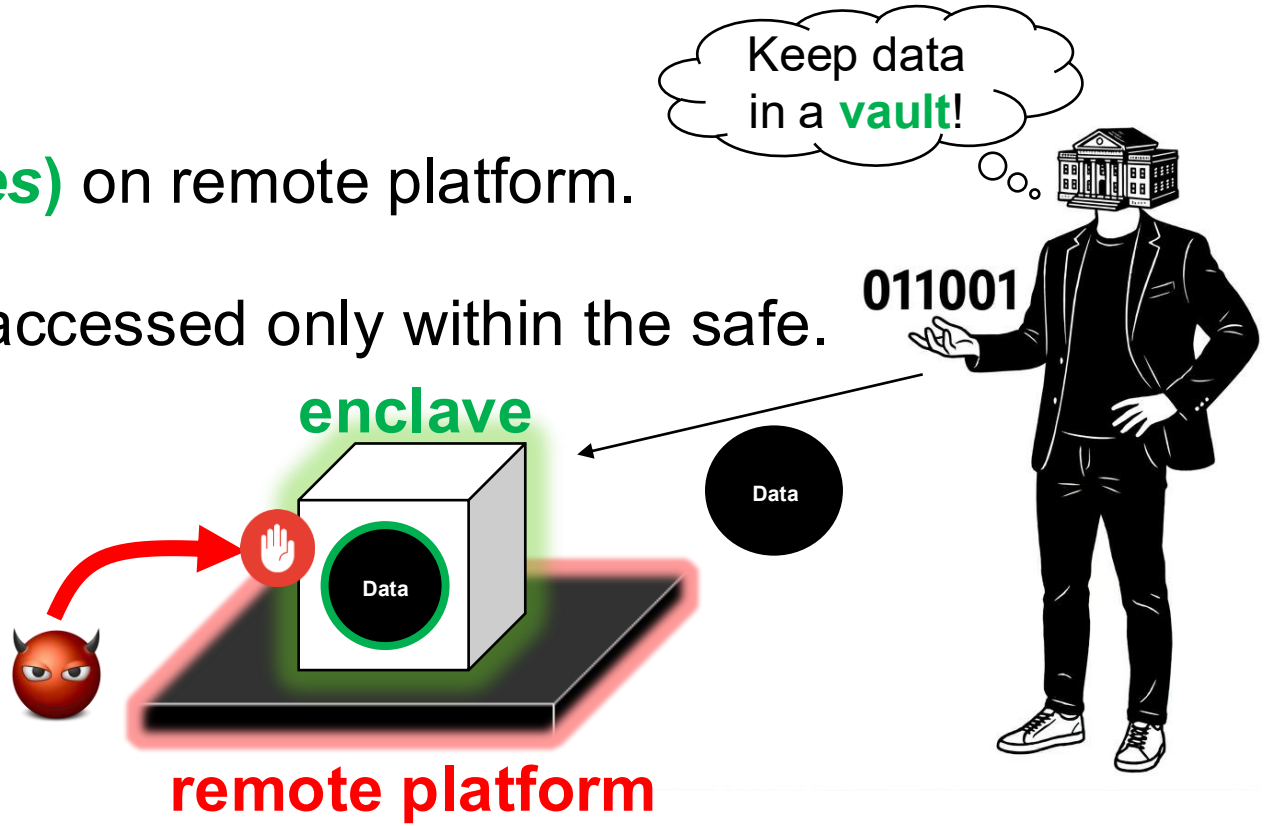
  } *cryptographic* primitives. **slow** computation
  - Synthetic Data
  - Differential Privacy (DP)

  } *data hiding* primitives. **low** accuracy
  - **Trusted Execution Environment (TEE)**

  } ***Possible solution without limitations?***

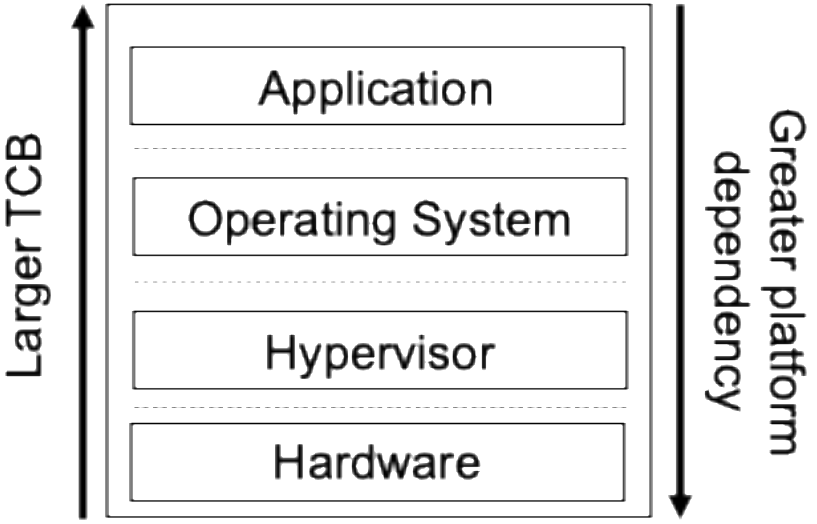
Trusted Execution Environment (TEE)

- Secure, isolated area that guarantees the **confidentiality** and **integrity** of code and data inside it.
- TEE owner's **safe/vault (i.e., *enclaves*)** on remote platform.
- Sensitive code and data in executed/accessed only within the safe.
  - ① Confidentiality : no access
  - ② Integrity : no modification
  - ③ Attestation : no impersonation



Types of TEEs – Less to Trust, Less to Break

- **Software-based TEEs**
  - Trusts privileged software (e.g., hypervisor)
  - No (or minimal) hardware dependency
  - Large trusted computing base (TCB)

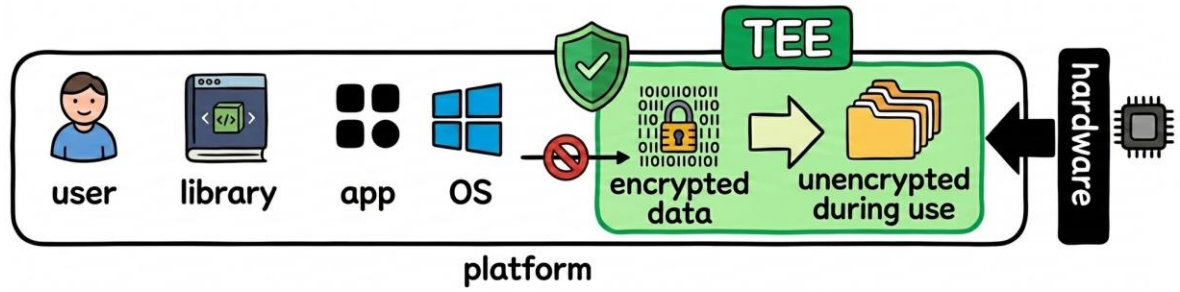


- **Hardware-based TEEs**
  - Only need to trust the device hardware
  - Hardware must support TEE extension
  - Better performance and security
  - Intel Trusted Domain Extensions, AMD Secure Encrypted Virtualization, …

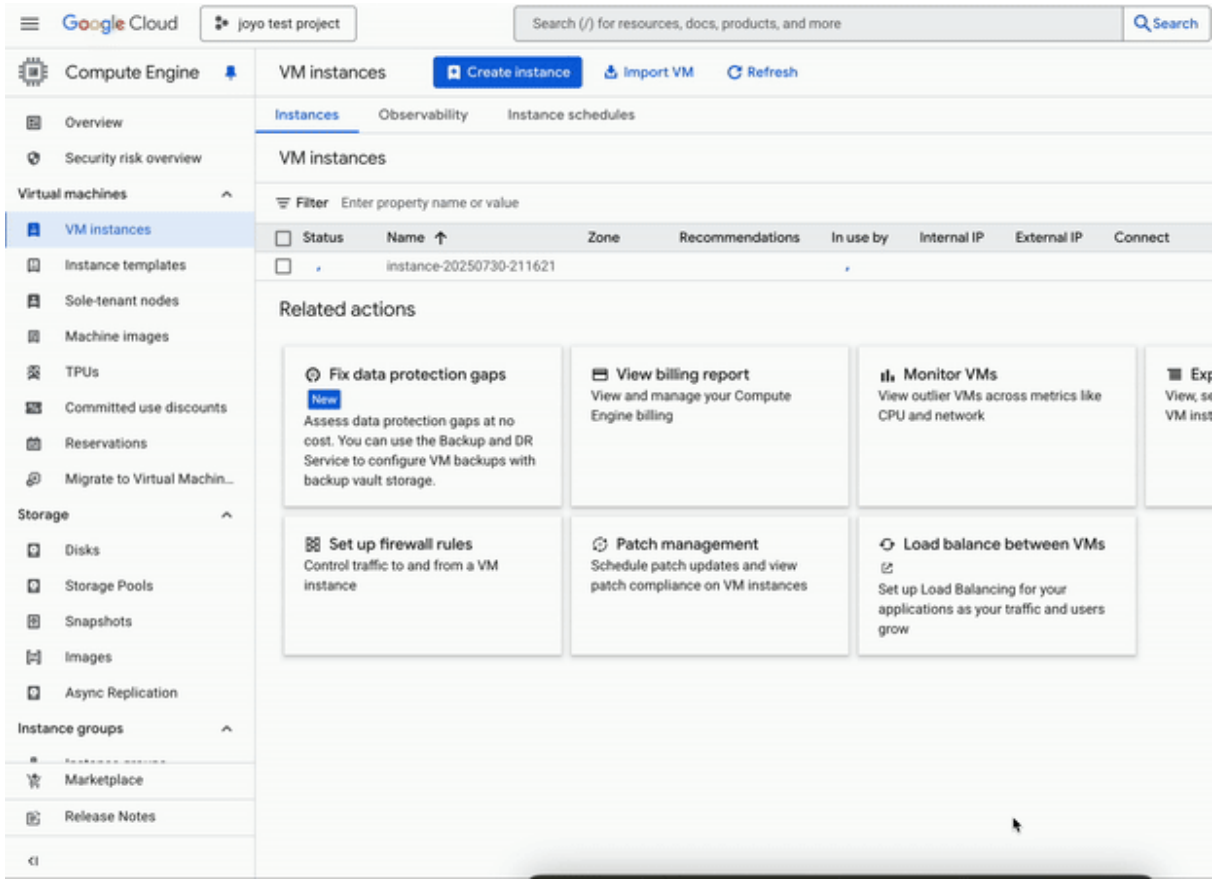
"Keep the trusted base as simple and small as possible"

Hardware-based TEEs

- Hardware serves as **root-of-trust**
- Supported by major CPU vendors
  - Intel, AMD, ARM, NVIDIA, (RISC-V), …
  - Continuous support and development
- Integrated into commercial cloud



Using Intel TEE (TDX) on Google Cloud

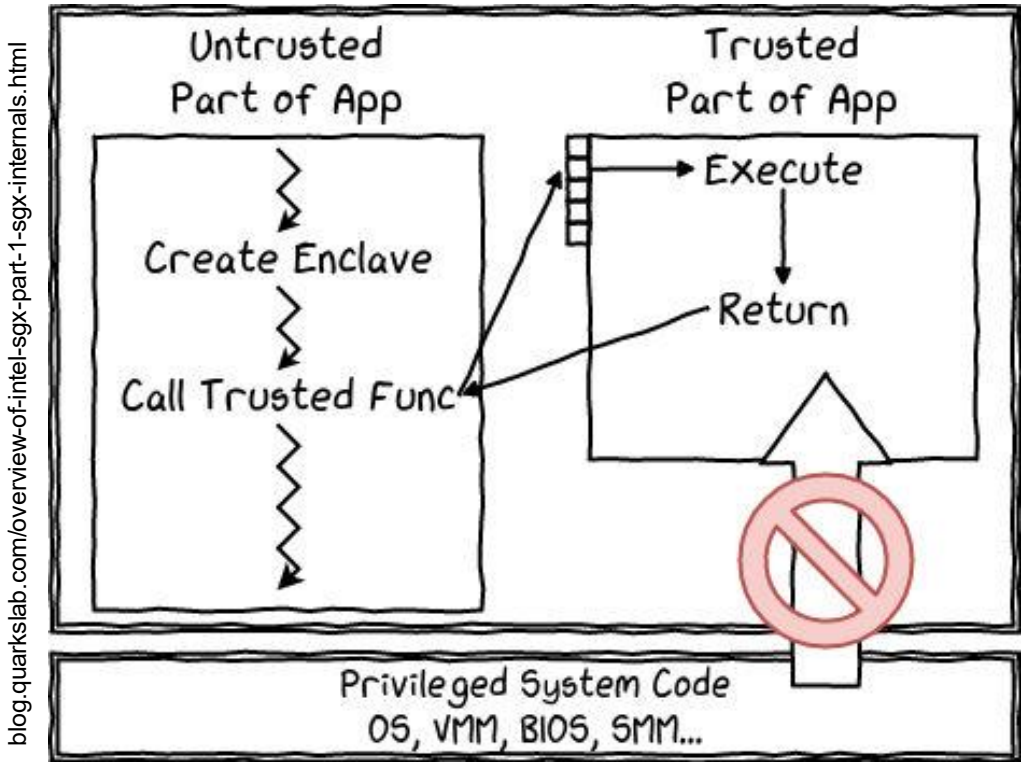


cloud.google.com/blog/products/identity-security/from-clicks-to-clusters-confidential-computing-expands-with-intel-tdx

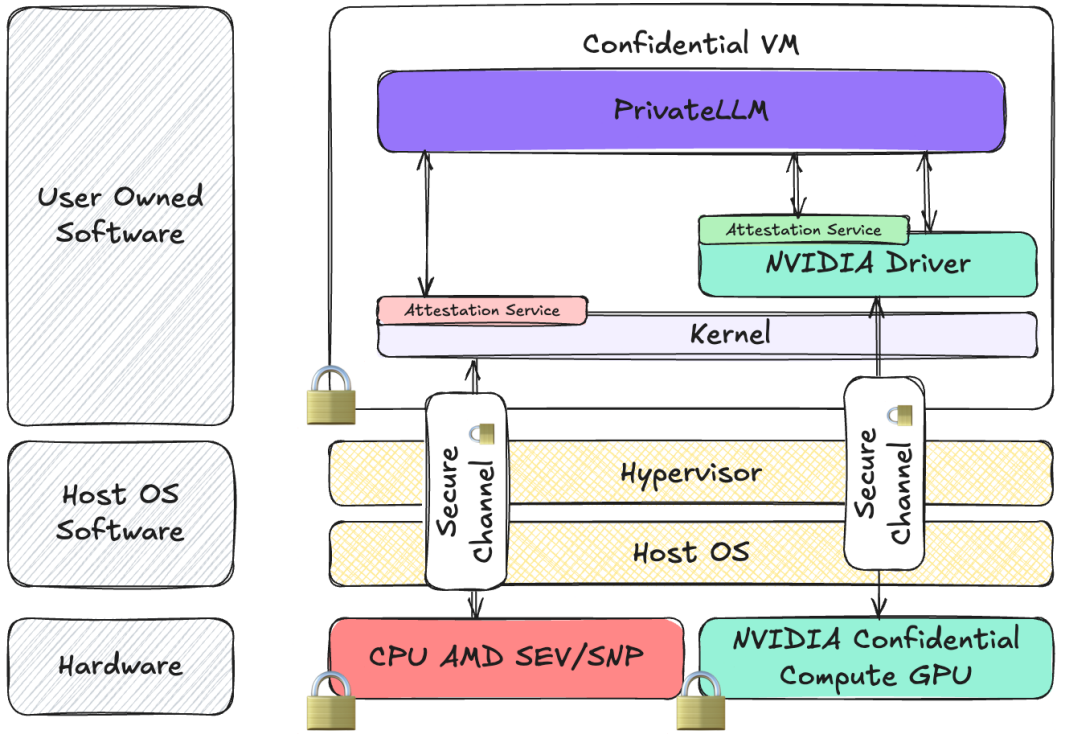
Architectures of HW-based TEEs

*more **fast***
*more **capacity***
*less **dev effort***

- **Initially application-level TEEs, moving on to VM-level TEEs**



blog.quarkslab.com/overview-of-intel-sgx-part-1-sgx-internals.html

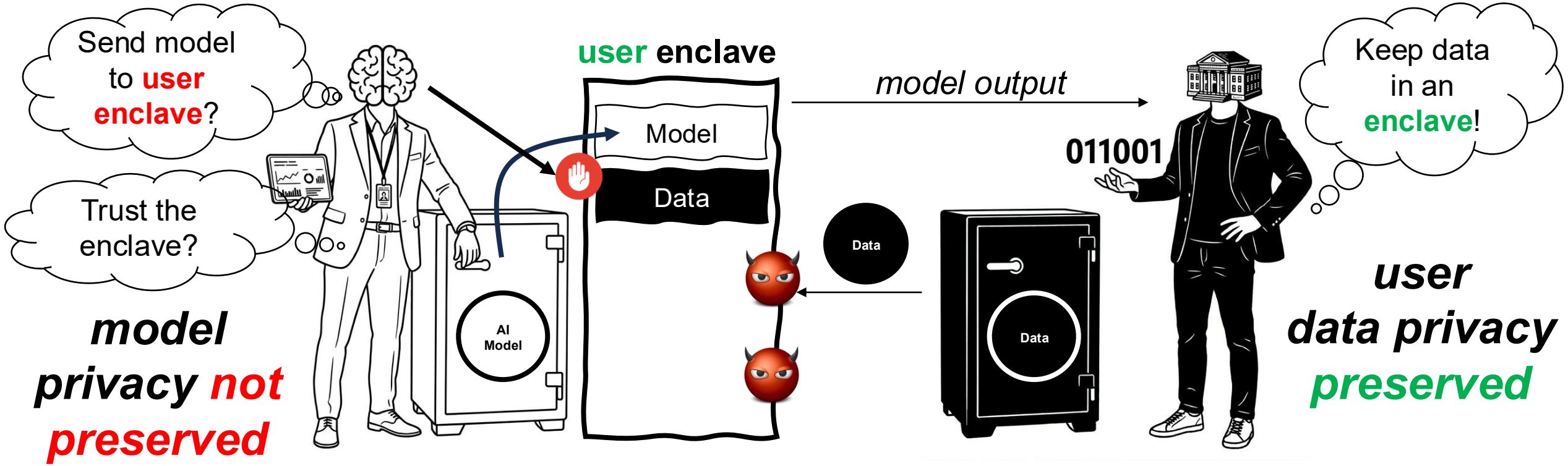


medium.com/@jcabrerohologueras/private-ai-at-scale-deploying-llms-with-trusted-execution-environments-f39e55de0de5

Application-level TEE **VM-level TEE**

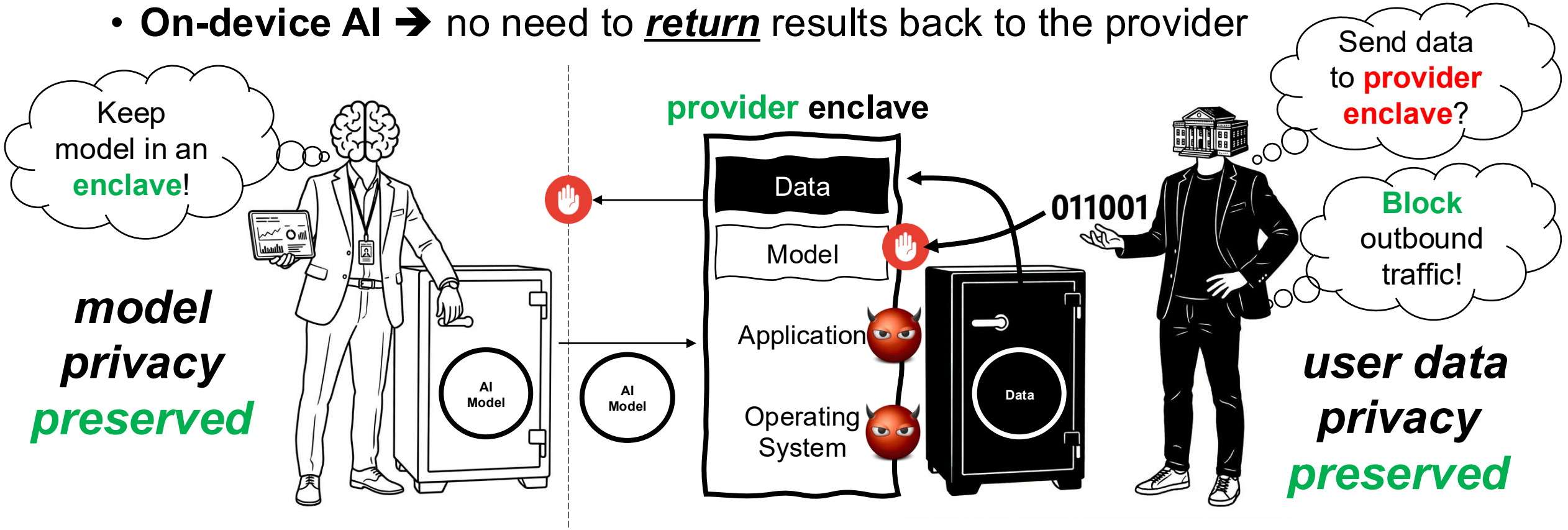
Can TEEs Solve Mutual Privacy Problem?

- **TEEs are designed only for one-directional privacy**
  - TEE owner has full control over what initially runs inside the enclave



Can TEEs Solve Mutual Privacy Problem?

- **(Other-way-around) What if we send the model to the user?**
  - **On-device AI ➔** no need to ***<u>return</u>*** results back to the provider

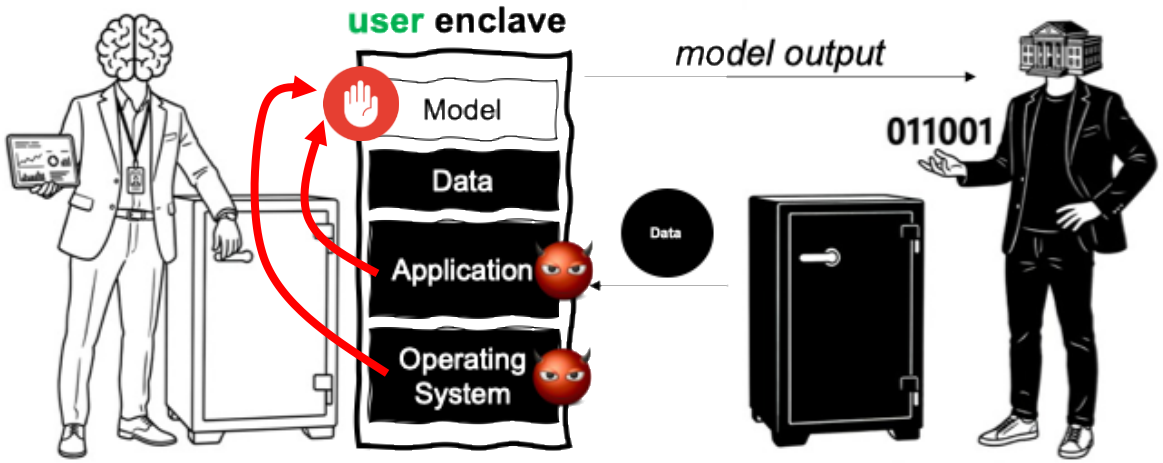


Protecting Mutual Privacy with Nested Enclaves

- **(Observation 1)** Sending user data to TEE cannot preserve model privacy
- **(Observation 2)** TEE-based secure on-device AI has performance limitations

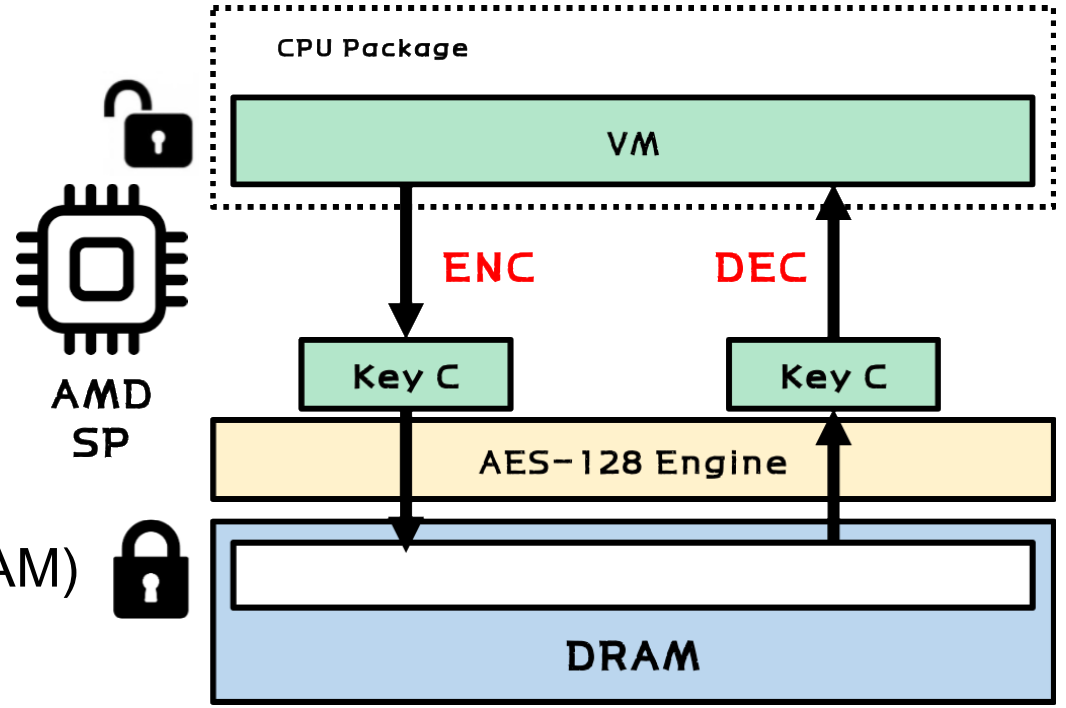
***Insight*: Let's protect model privacy while using user TEE**

***Solution*: Let's makes a provider enclave *inside* user TEE**



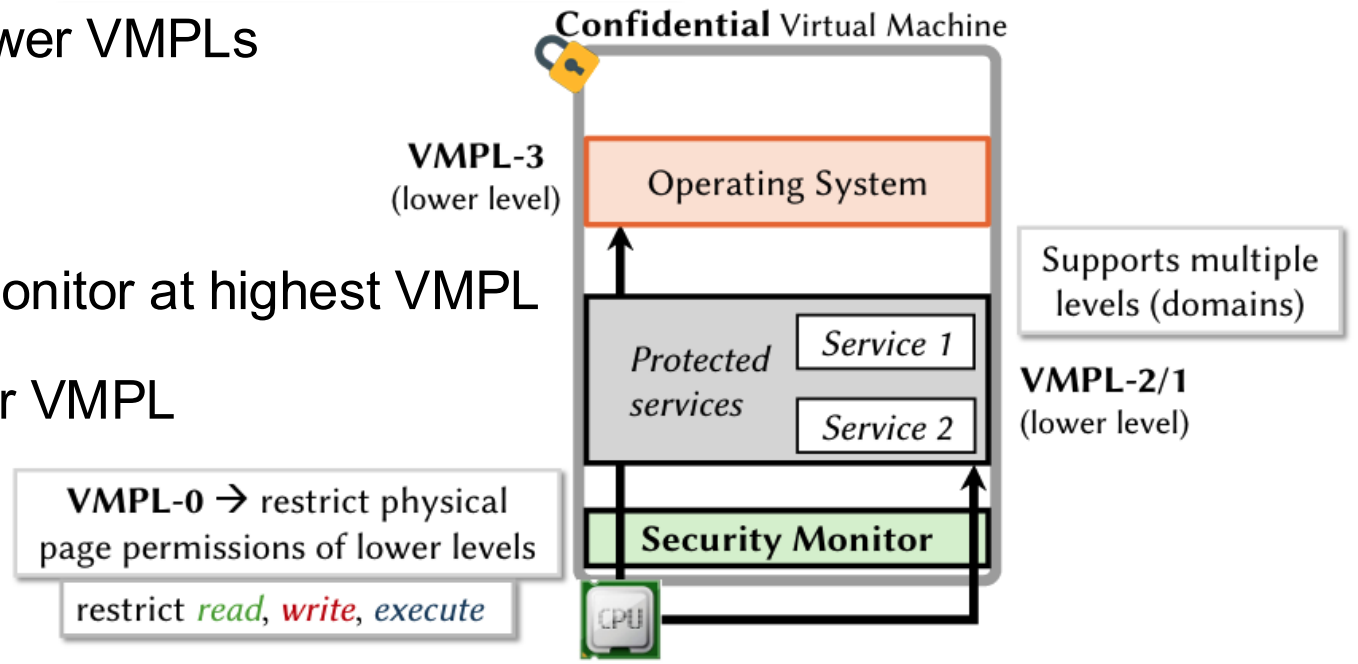
Nested Enclaves on AMD (1)

- AMD **Secure Encrypted Virtualization (SEV)** Secure Nested Paging (SNP)
  - Latest VM-level TEE architecture from AMD
  - AMD Secure Processor (SP)
  - AES128 encryption engine
  - **Per-VM encryption key** managed by AMD SP
  - Data is **unencrypted** during use in CPU
  - Data is remains **encrypted** in memory (e.g., DRAM)



Nested Enclaves on AMD (2)

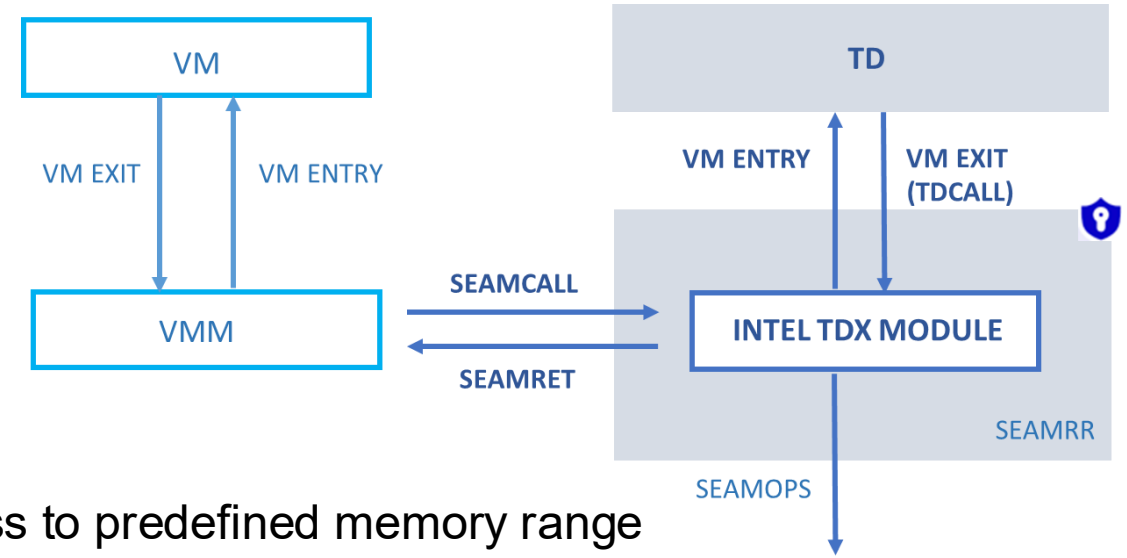
- AMD **Virtual Machine Privilege Level (VMPL)**
  - Hardware extension supported by SEV-SNP ➔ vCPUs execute in different privilege levels
  - Higher VMPLs can control access of lower VMPLs
- **Nested Enclaves** with VMPL
  - Place small, open-sourced, verifiable monitor at highest VMPL
  - OS and normal applications run in lower VMPL
  - Enclaves (i.e., model) runs in higher VMPL than the OS/apps



adil-ahmad.net/slides/ahmad-veil.pdf

Nested Enclaves on Intel (1)

- Intel **Trusted Domain Extensions (TDX)**
  - Latest VM-level TEE architecture from Intel
- Components
  - Secure-Arbitration Mode (SEAM)
    - New processor execution mode that allows access to predefined memory range
    - Range defined by SEAMRR can only be access by executable code page within that range
  - Intel TDX Module
  - Multi-Key Total Memory Encryption (MKTME)

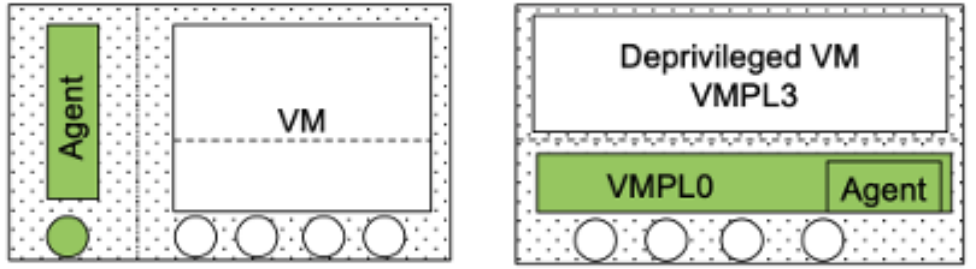


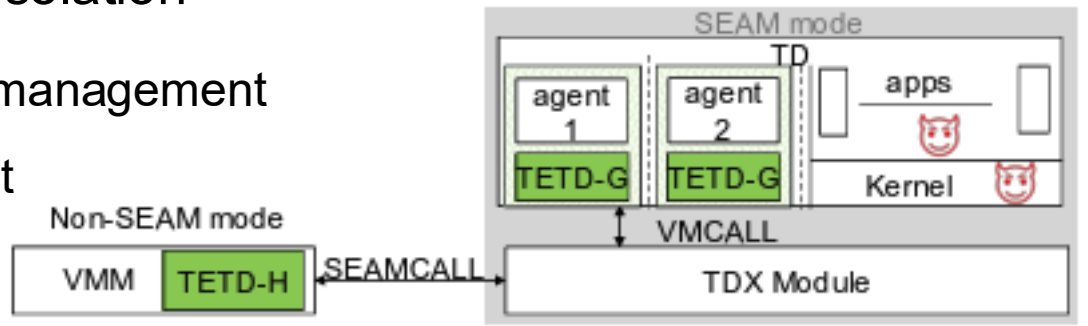
www.intel.com/content/www/us/en/developer/articles/technical/trust-domain-extensions-on-4th-gen-xeon-processors.html

Nested Enclaves on Intel (2)

- Intel TDX does ***not*** have VMPLs
- (**Solution**) Separate the resource (e.g., vCPU)
  - Nested enclave execution bound to specific vCPU
  - Memory used by enc-vCPU is isolated
  - Additional components (TETD-H/G) enforces the isolation
    - Handled through collaboration of VMM's resource management and memory blocking mechanism during vCPU rest

Comparison vs. AMD's VMPL



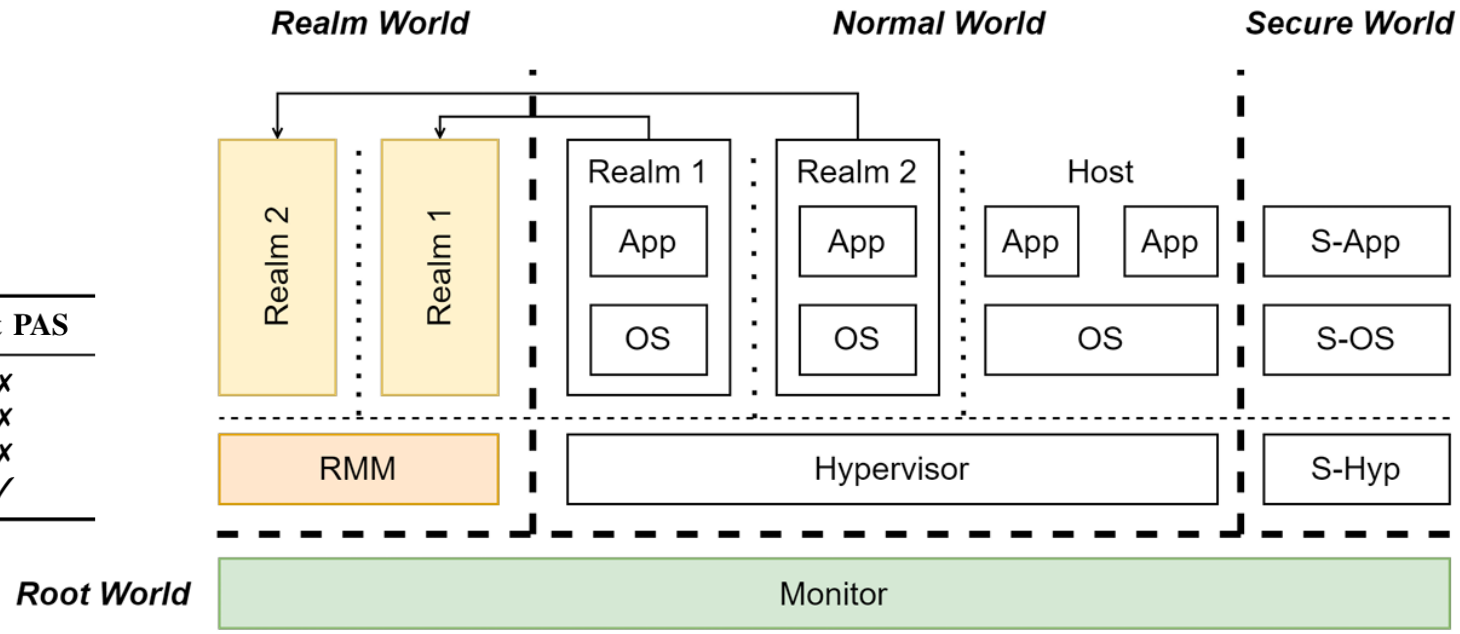


www.usenix.org/system/files/usenixsecurity25-wang-zhanbo.pdf

Nested Enclaves on ARM (1)

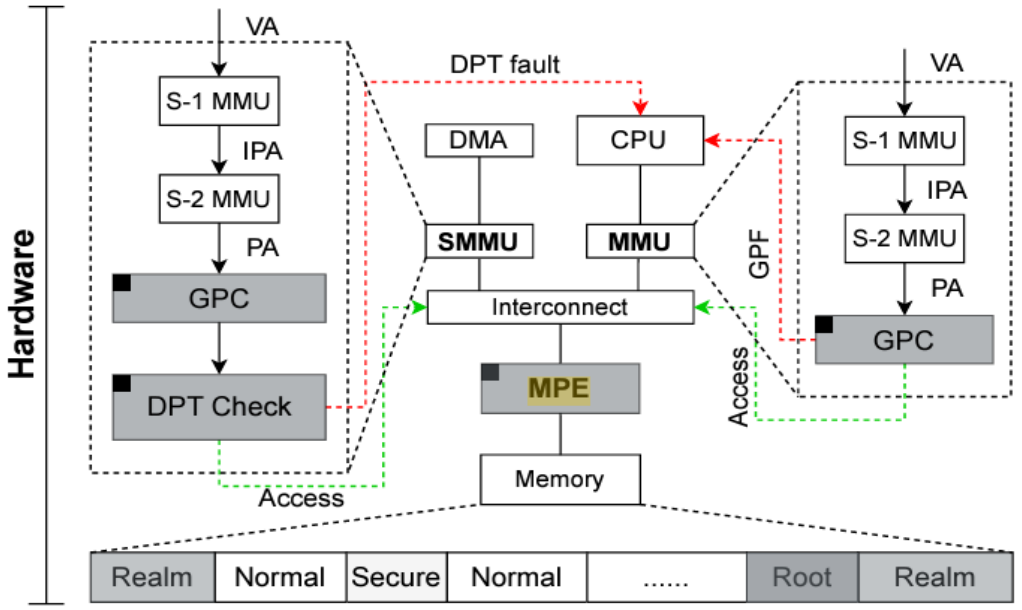
- ARM **Confidential Compute Architecture (CCA)**
- Provide confidential computing for next generation (ARMv9.2-A) ARM devices
  - New security state for confidential computing: ***Realm world***
  - Hardware-isolated ***Root world***
  - New security supports…
    - Granule Protection Check (GPC)
    - Memory encryption

Security State	Normal PAS	Secure PAS	Realm PAS	Root PAS
Normal	✓	✗	✗	✗
Secure	✓	✓	✗	✗
Realm	✓	✗	✓	✗
Root	✓	✓	✓	✓



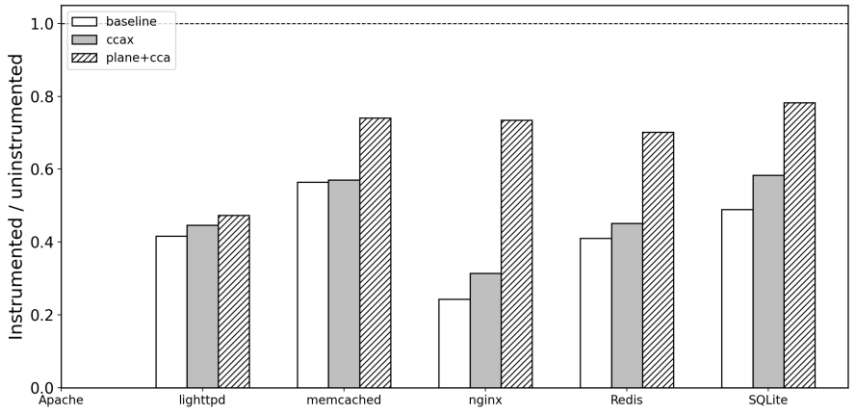
Nested Enclaves on ARM (2)

- **Granule Protection Table (GPT)**
  - Data structure that manages physical address space (PAS) that each memory page belongs to (i.e., GPT entry = { physical address, (normal/secure/root/Realm) }
  - Hardware MMU checks the validity of the access against GPT (Granule Protection Check)
  - Can only be modified by the EL3 monitor
- **Memory Protection Engine (MPE)**
  - Support memory encryption per PAS
  - On-going development for per-Realm VM encryption
- **Device Protection Table (DPT)**
  - Extension for per-device memory access control
  - Only authorized devices can access memory



Nested Enclaves on ARM (3)

- ARM CCA does ***not*** have VMPLs, ***cannot*** rely on VMM for memory blocking
- ARM CCA Planes
  - Recent CCA extension similar to VMPL on AMD
  - Each vCPUs can execute in context called *planes*
  - Primary plane can control access of other auxiliary planes
- ***Run security monitor in the primary plane*** ➔ enough?
  - Comes at a cost of **performance**



over 2x slowdown on applications

Conclusion

- AI model/service usage creates a **privacy deadlock**
- **Trusted execution environment** can be a potential solution to the deadlock (with limitations)
- **Nested enclaves** inside TEE can solve privacy deadlock without limitations
- Security and privacy issues in era of AI is solvable through the lens of **system security**

Thank you for listening

jsyou0224@gmail.com