Security for AI and AI for Security

Deming Chen

Abel Bliss Professor

Electrical and Computer Engineering

ZTHA. 09/04/2024

The Intersection of AI and Security

- AI's rapid adoption across industries
  - Healthcare, finance, retail, …
- Increasing reliance on AI in critical applications
  - Autonomous vehicles, national security, smart grid, …

AI should be secure

- Enhance security through AI-driven frameworks
  - Use machine learning to detect and respond to cyber threats in real-time
  - Use AI tools to detect and prevent money laundering, insider trading, and other illegal activities
  - Many others …

AI for security

Dual focus: securing AI and leveraging AI for security

Figure Source: https://www.informationweek.com/machine-learning-ai/how-safe-and-secure-is-genai-really-#close-modal

Security Concerns for AI Systems

Physical Attacks	Memory snooping attack Cold boot attack
Firmware/rootkit Attacks	LoJax (backdoor)
Side-channel Attacks	Meltdown and Spectre
Data Exfiltration	Data Breach: many incidents

AI-Specific Security Concerns:

- Inserting malicious data points to train the AI models
- Small perturbations of the input data to mislead AI models
- Reverse-engineering for the training data or proprietary algorithms/structures
- Vulnerabilities or deficiencies of the AI models themselves

Security for AI

Maybe AI is creating a new fashion 😂

Secure AI Systems

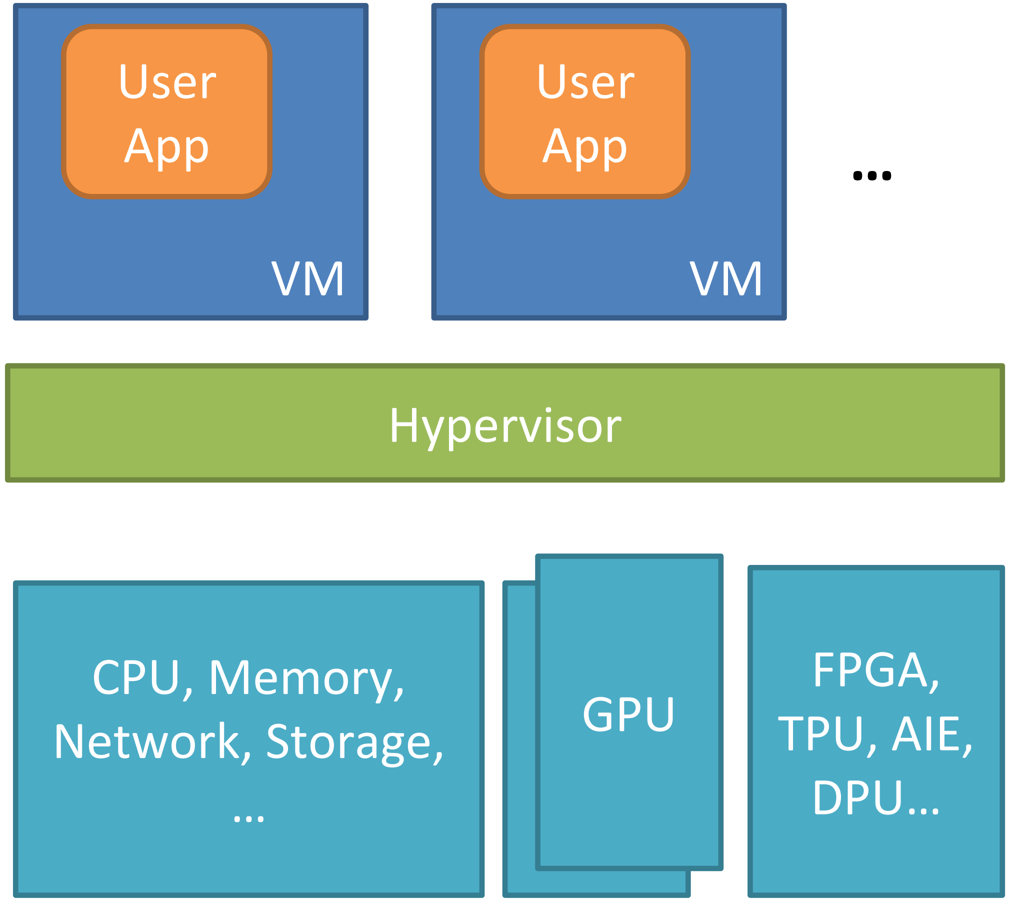
- **Trusted Execution Environments (TEEs)**: Secure environments for executing AI models
- **AI Accelerators**: Specialized hardware for enhancing AI performance while maintaining security
- **Secure Data Handling**: Ensuring data privacy and integrity during training and inference
- **Robust AI Models**: Developing AI systems resilient to adversarial attacks
- **Confidential Computing**: Protect data in-use, in additional to at-rest, and in-transit

Where Are the Vulnerabilities?

Trusted Computing Base (TCB) is too large

- Security application and software stack
- Operating system
- Hypervisor
- Cloud infrastructure
- Hardware

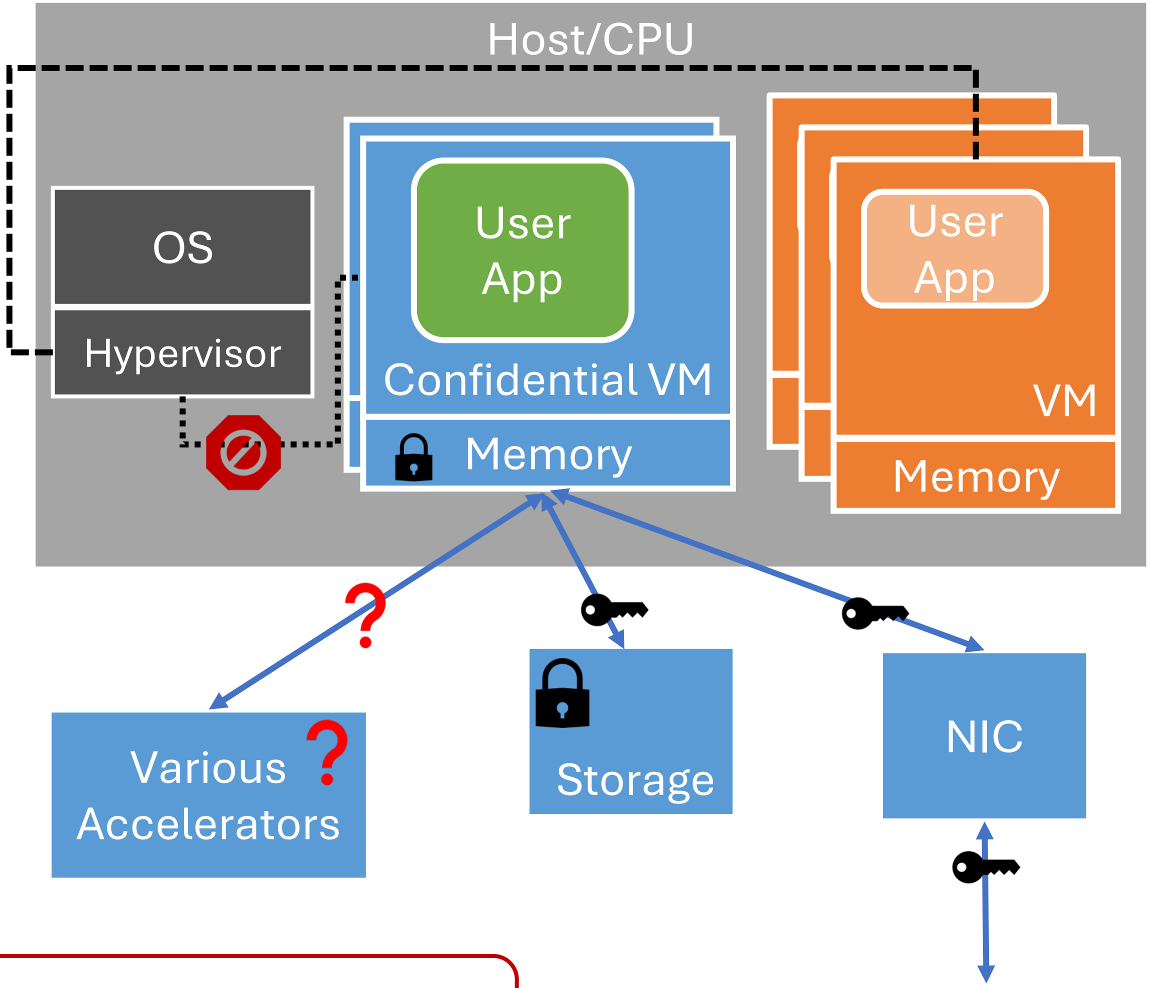
No well-established secure framework for heterogeneous environment in general



Secure System for AI Models with AI Accelerators

- Trusted Execution Environment (TEE)
  - Confidential VM
- CPUs
  - AMD SEV, Intel TDX, ARM TrustZone, …
- Accelerators
  - Nvidia H100 GPU for confidential computing

No systematic solutions for extending TEE to accelerators in general



Our focus today

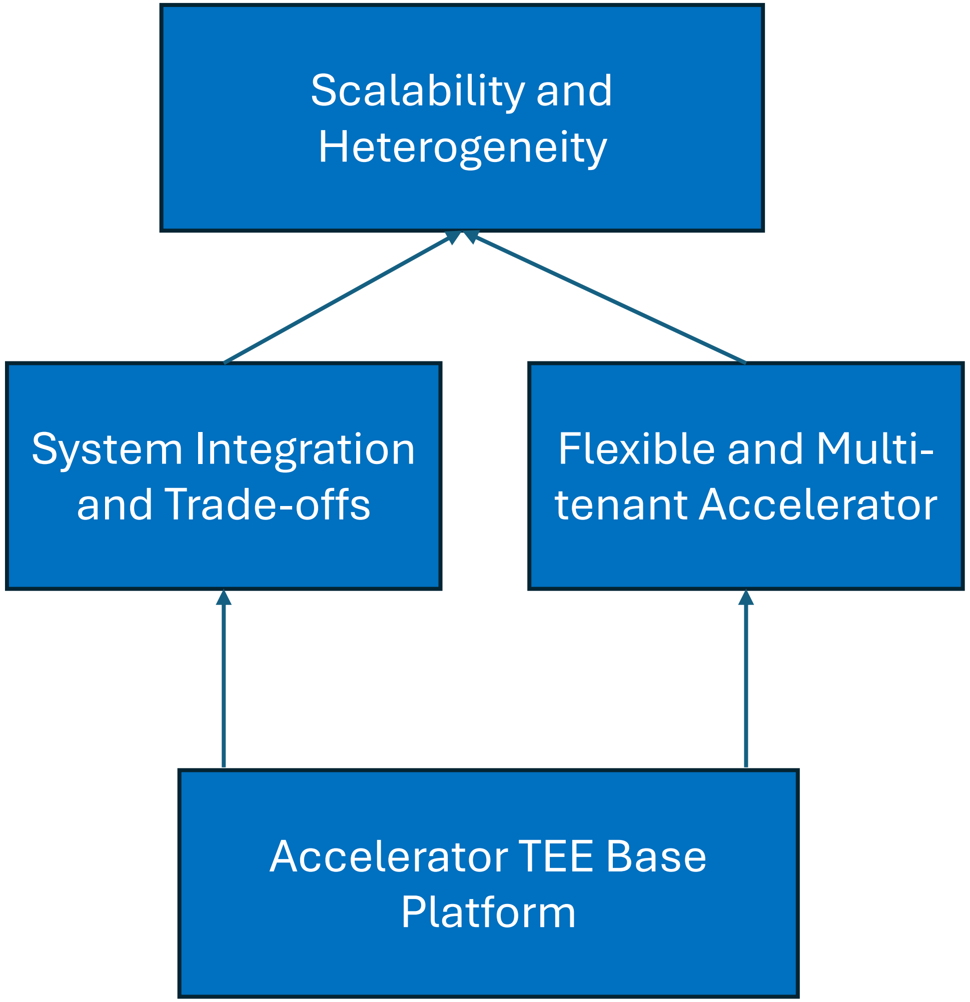
Questions

- How should we provide a secure execution environment or framework for AI accelerators?
- How should we design and integrate security solutions for AI accelerators?
- How should security solutions for AI accelerators adapt to future architectures that are dynamic and configurable?
- How should we design for scalable and heterogeneous systems?

Our Approach

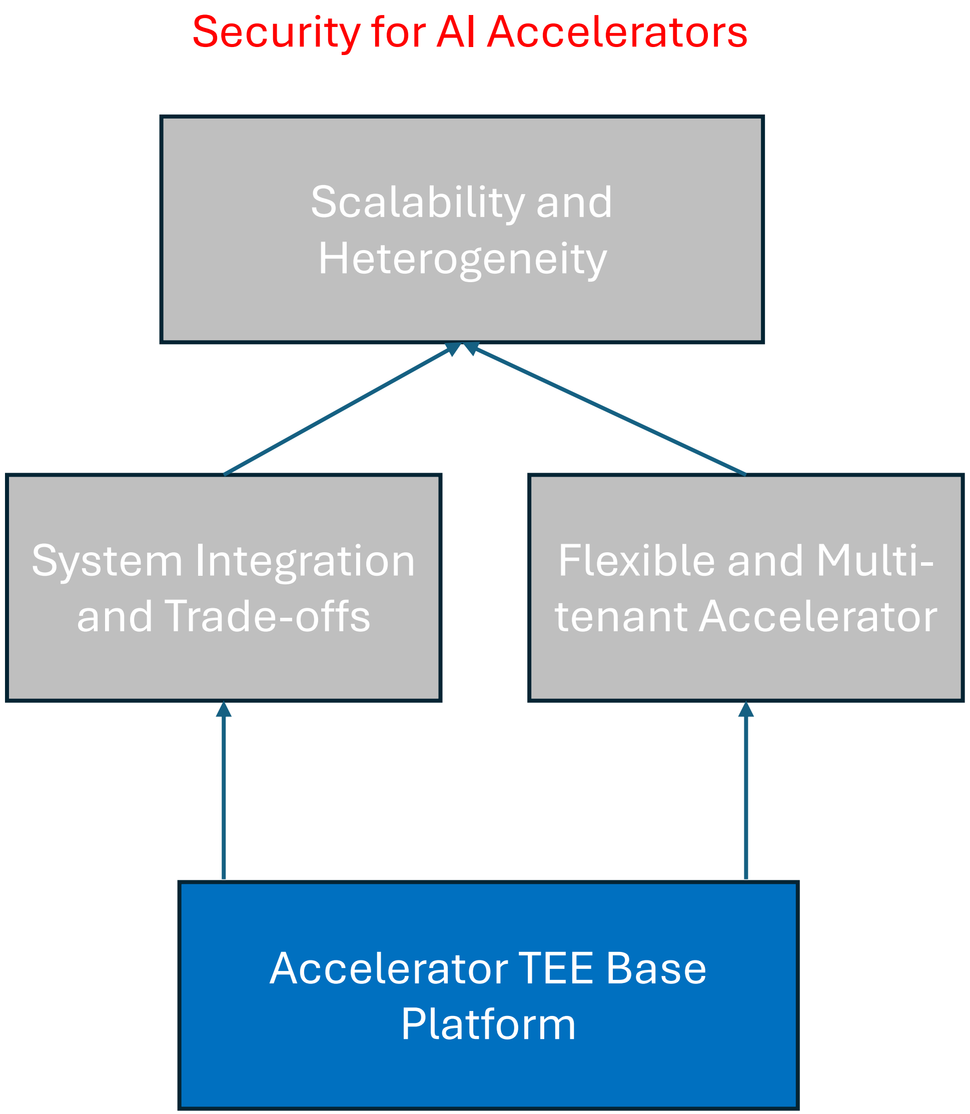
- Accelerator TEE Base Platform
  - AccGuard: Secure and Trusted Computation on Remote FPGA Accelerators [iSES'21]
- System Integration and Trade-offs
  - AccShield: a New Trusted Execution Environment with Machine-Learning Accelerators [DAC'23]
- Flexible and Multi-tenant Accelerator
  - S^2TAR: Shared Secure Trusted Accelerators with Reconfiguration for Machine Learning [CLOUD'24]
- Scalability and Heterogeneity
  - Future Research

Security for AI Accelerators



Research Questions

- How should we provide a secure execution environment or framework for AI accelerators?
- How should we design and integrate security solutions for AI accelerators?
- How should security solutions for AI accelerators adapt to future architectures that are dynamic and configurable?
- How should we design for scalable and heterogeneous systems?



Threat Model

Trusted

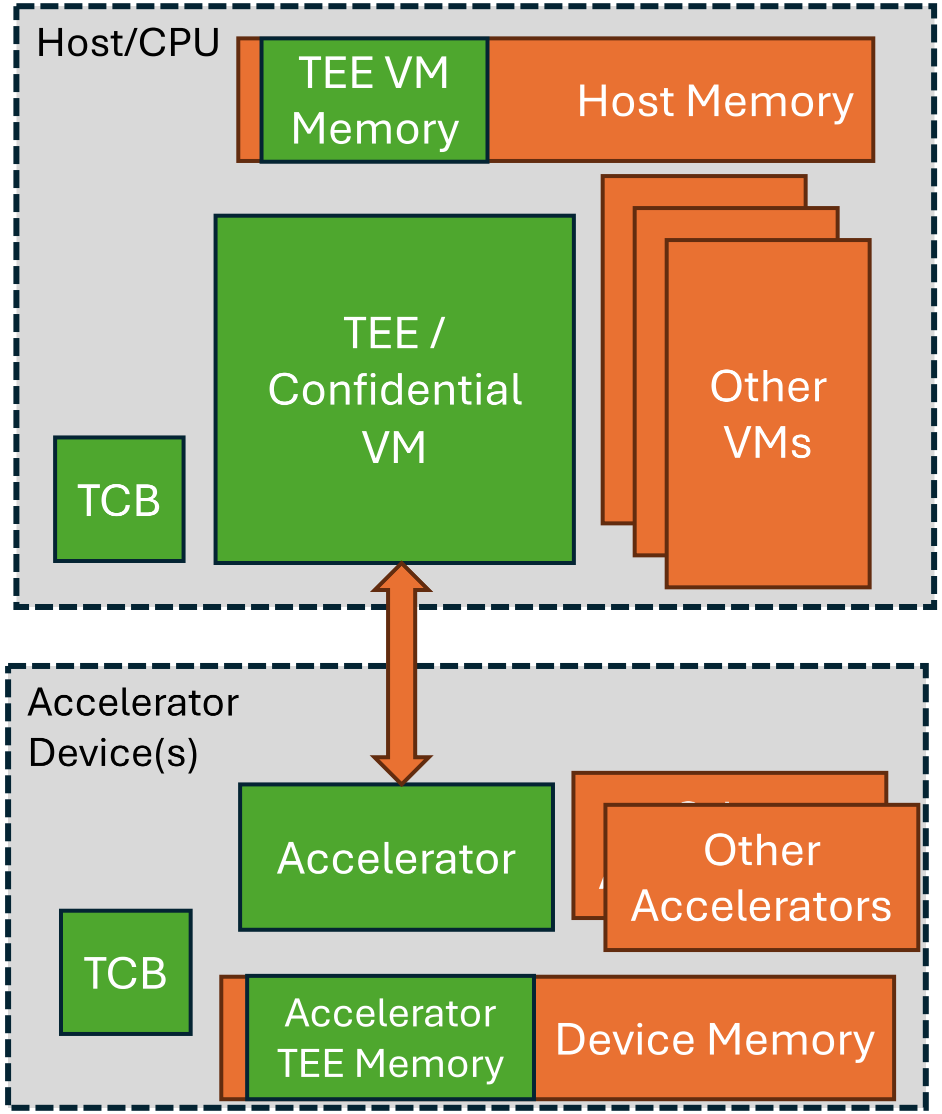
- Hardware and root of trust
- TCB
- TEE framework

Semi-trusted

- Cloud service provider

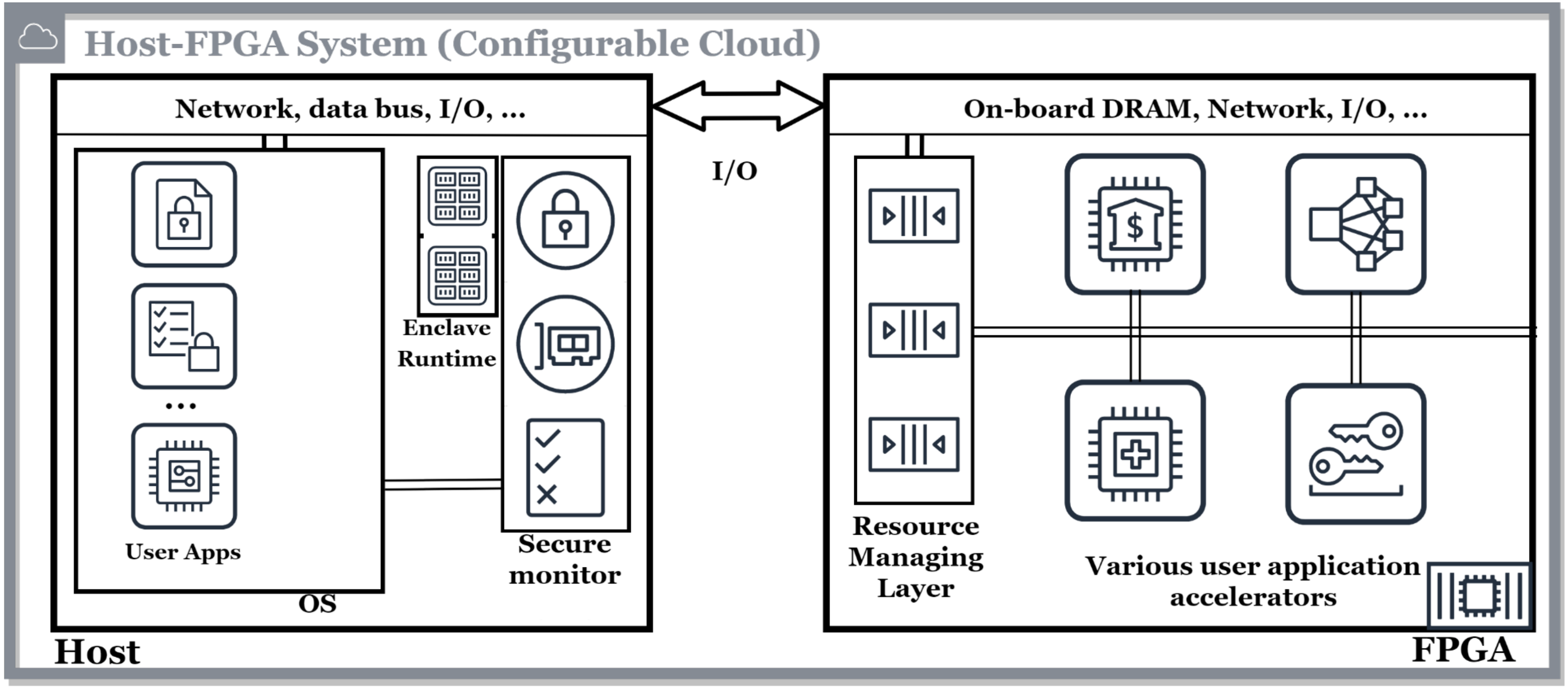
Untrusted

- Communication links
- VMs from other co-tenants
- Accelerators from other co-tenants
- External memory outside TEE



Side-channel attacks are outside the scope of this work, but the mitigations can be applied orthogonally.

FPGA Emulation with Host TEE

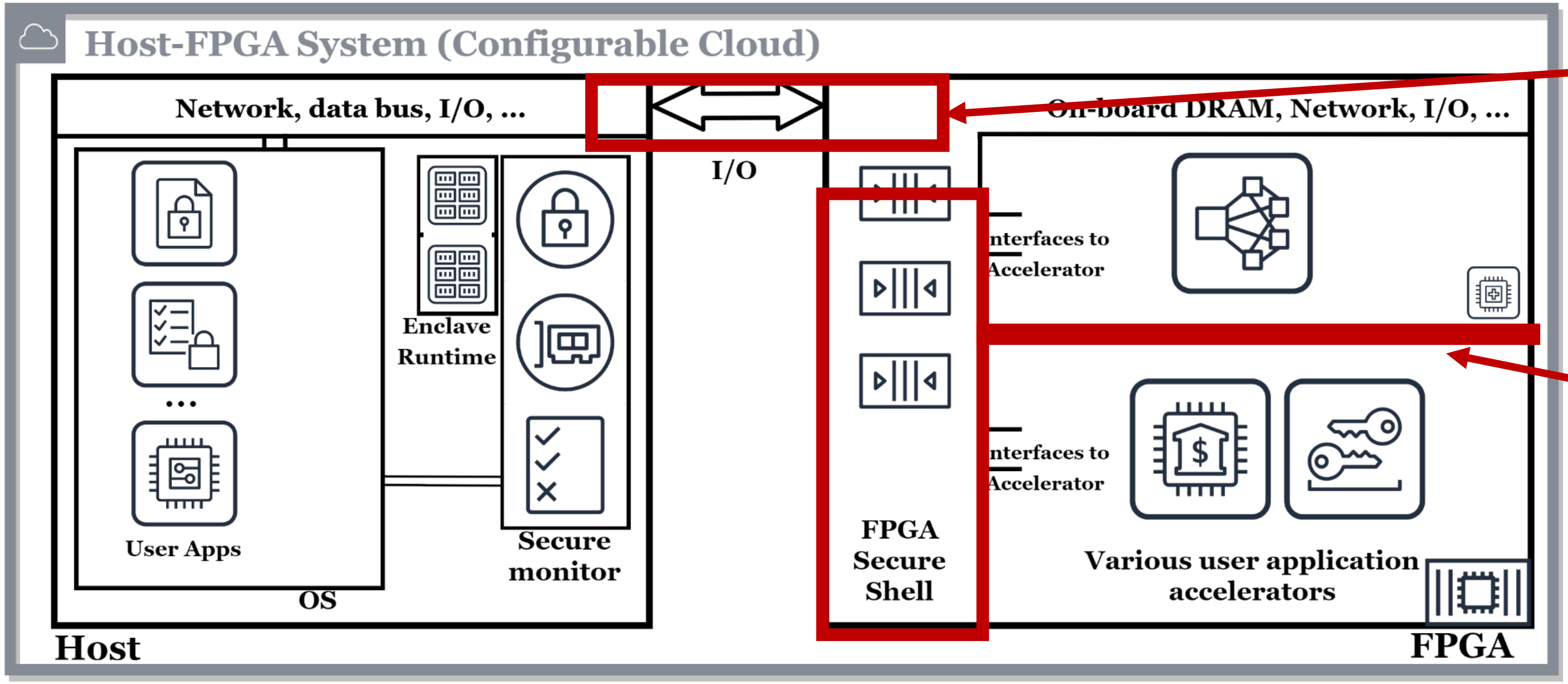


Host:
TEE / Enclave framework

FPGA:
Lack of protection among accelerators

Extend Isolation for Security – FPGA Side

- Physical (design) isolation – partial reconfiguration
- Logical isolation – Secure Monitor (SM) and FPGA Security Manager
  - Enforce strict resource and access control
  - Secure communication channels

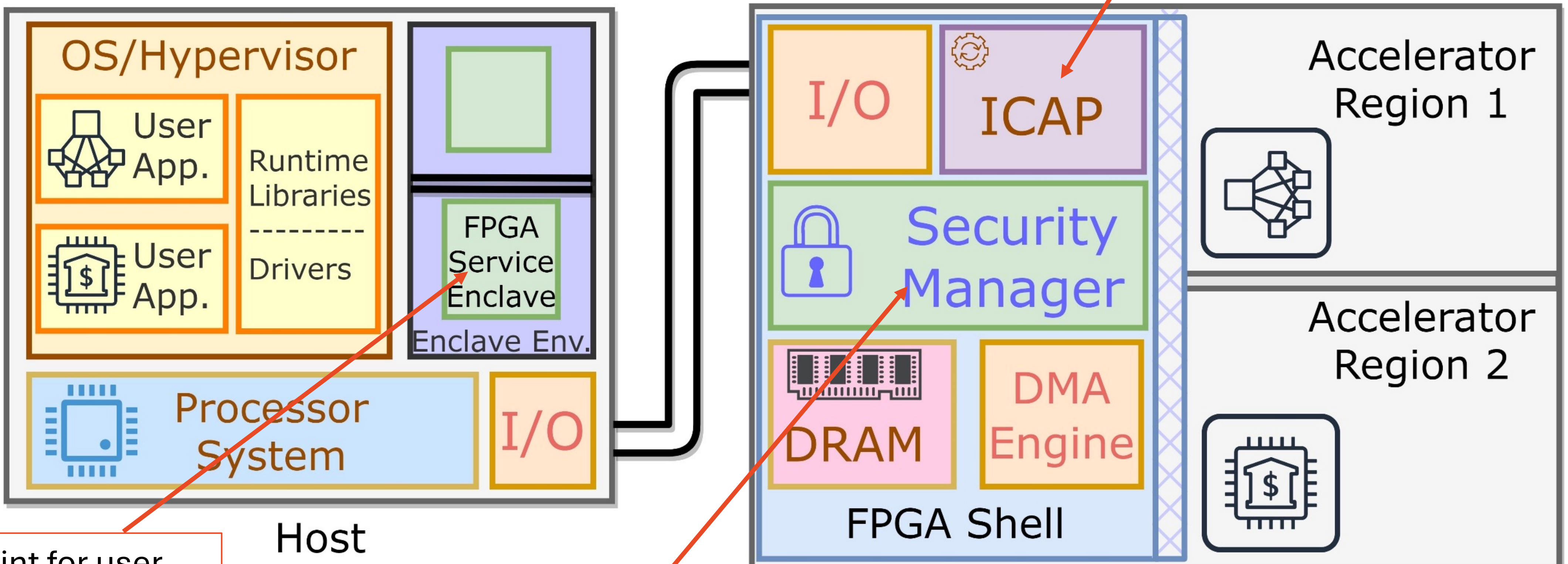


Encrypted communication – separate secure channels

Isolation between accelerators

Block Diagram of AccGuard

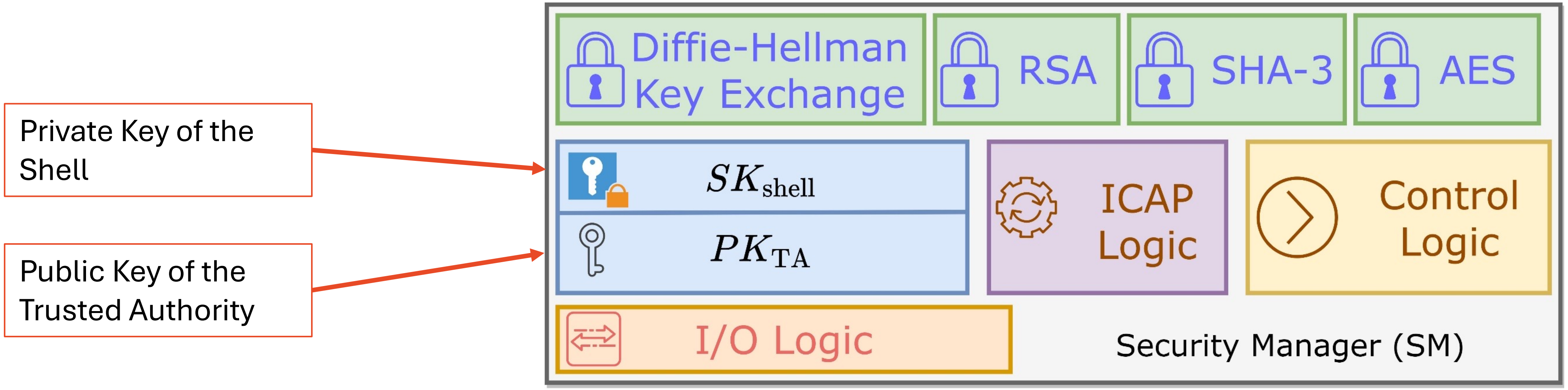
The internal configuration port to manage partial reconfiguration



Contact point for user applications to interact with the FPGA

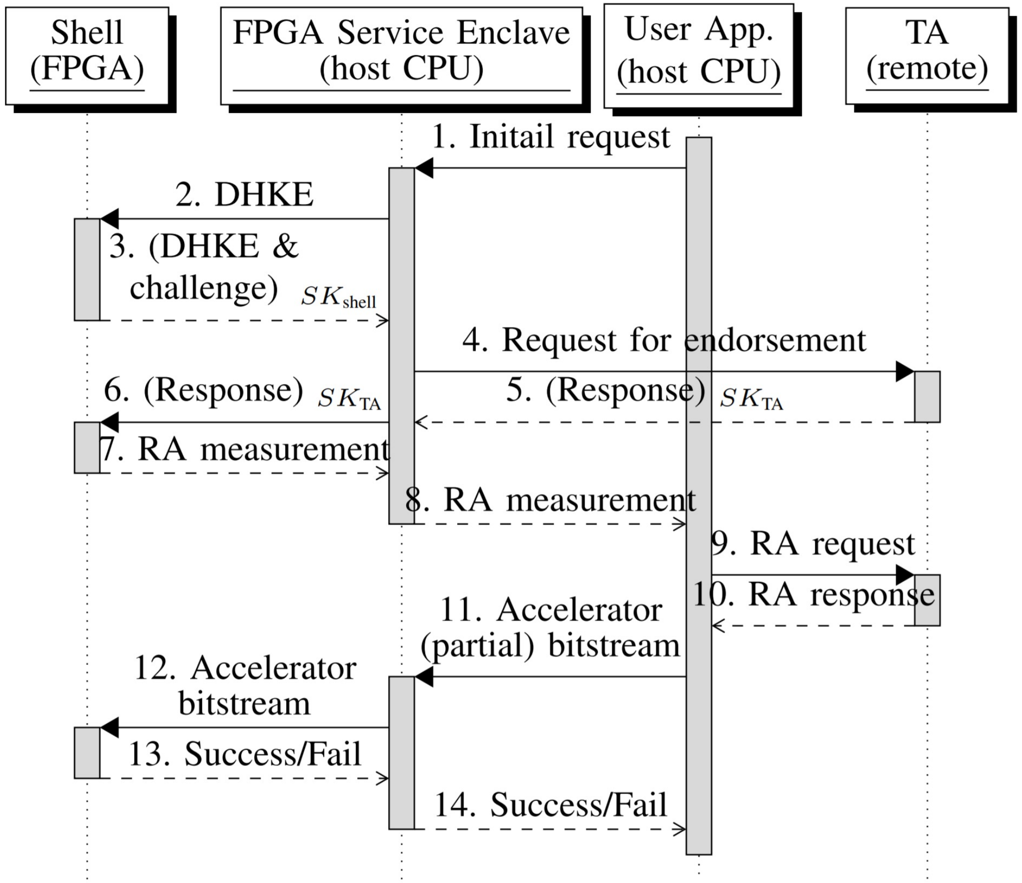
Handles the secure communication channels, remote attestation, and the hardware root of trust for FPGA accelerators

Security Manager with Hardware Root of Trust



- Private key of shell SK_{shell} and public key of TA PK_{TA} are stored in Security Manager
- The TA generates a device root key K_DEV using the FPGA's DNA and copies it into the secure storage on FPGA.
- Root key (K_DEV) will facilitate device lookup and authentication.
- The shell bitstream can only be operational on the FPGA with the correct device root key.

Procedure of Remote Attestation



- Steps 2 to 6 establish the mutual trust between the FPGA shell and the host service enclave dedicated to managing FPGA accelerators.
- Steps 7 to 10 perform the remote attestation with the TA.
- Step 11 and onward load the partial bitstream of the accelerator and protect its integrity through encryption.
- The procedure aborts if authentication fails or result mismatch happens in any of the steps above.

Proof-of-Concept Implementation

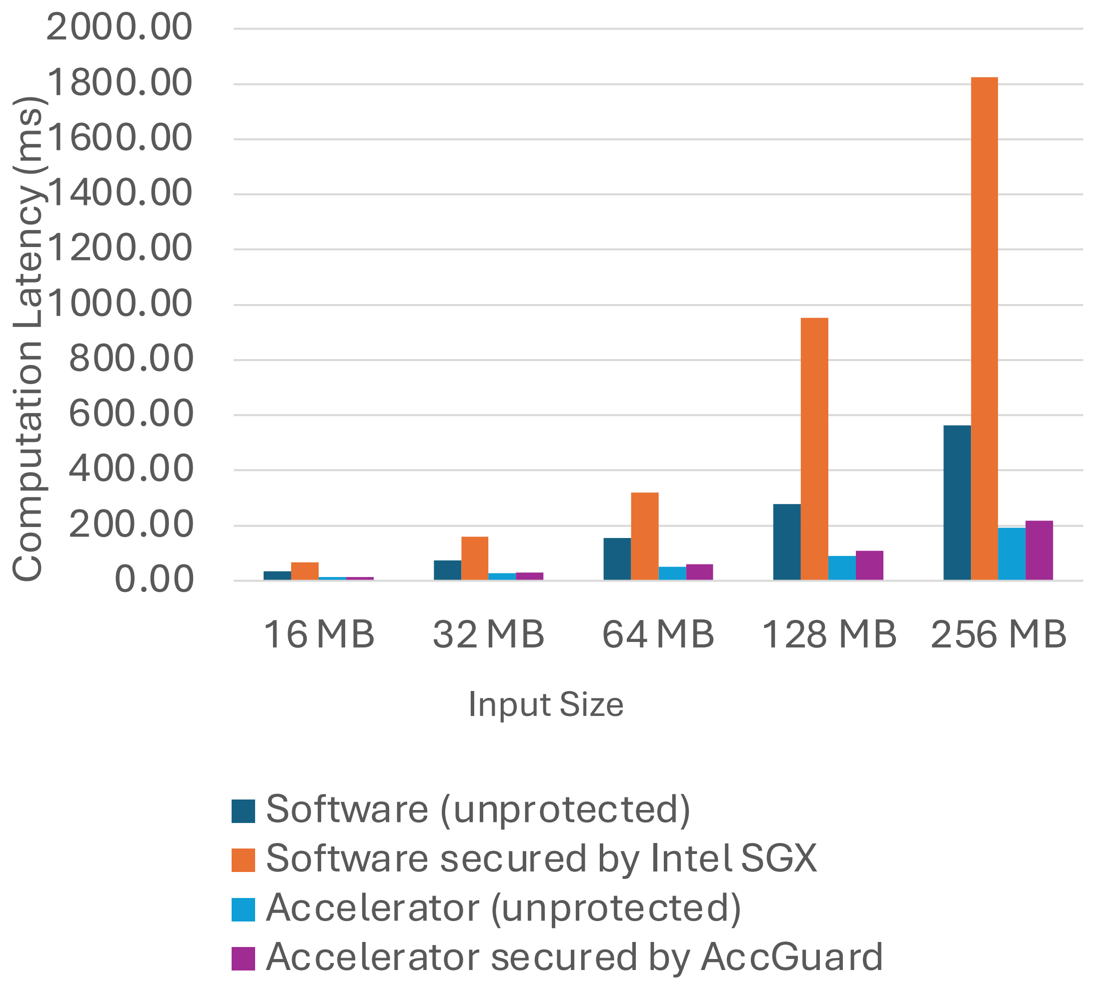
- Xilinx ZedBoard (Zynq-7000) FPGA
- One of the ARM cores runs as host and secure monitor (TrustZone)
- The other CPU core is integrated into the shell and security manager to control accelerators and data flows.
- FPGA shell implemented using HLS and RTL (125 MHz)

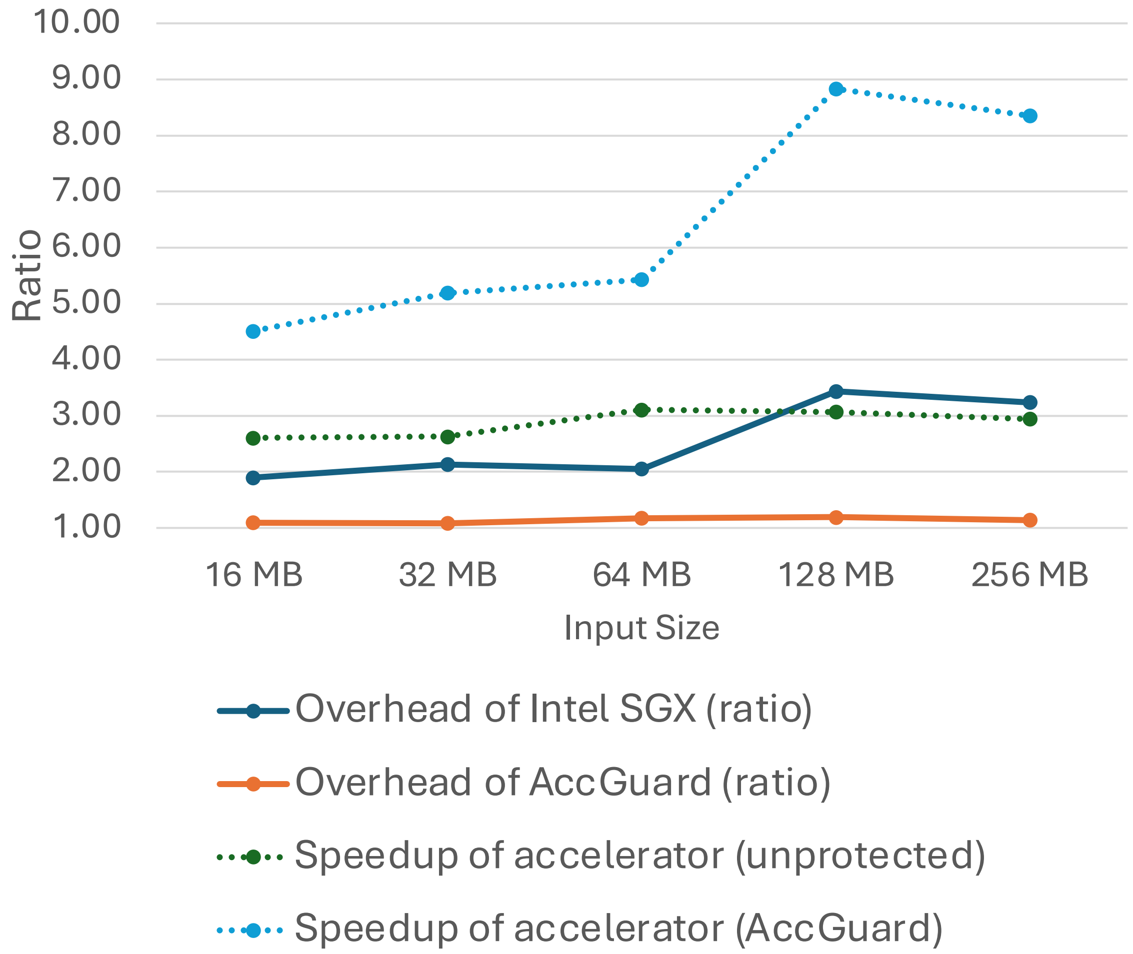
- FPGA accelerator computing basic histogram application as a synthetic benchmark
- Compared to a software version secured by Intel SGX (running on Intel i7-7700K: 4.2 GHz frequency and 16GB of DRAM).

Name	Used	Available	Utilization (%)
LUT	45064	53200	84.71
LUTRAM	11486	17400	66.01
FF	44238	106400	41.57
BRAM	90	140	64.28

- Our design provides strong isolation guarantees while minimizing the performance impact
- The methodology is not FPGA-specific and can be applied to accelerator designs in general

Results – Computation Latency & Overhead

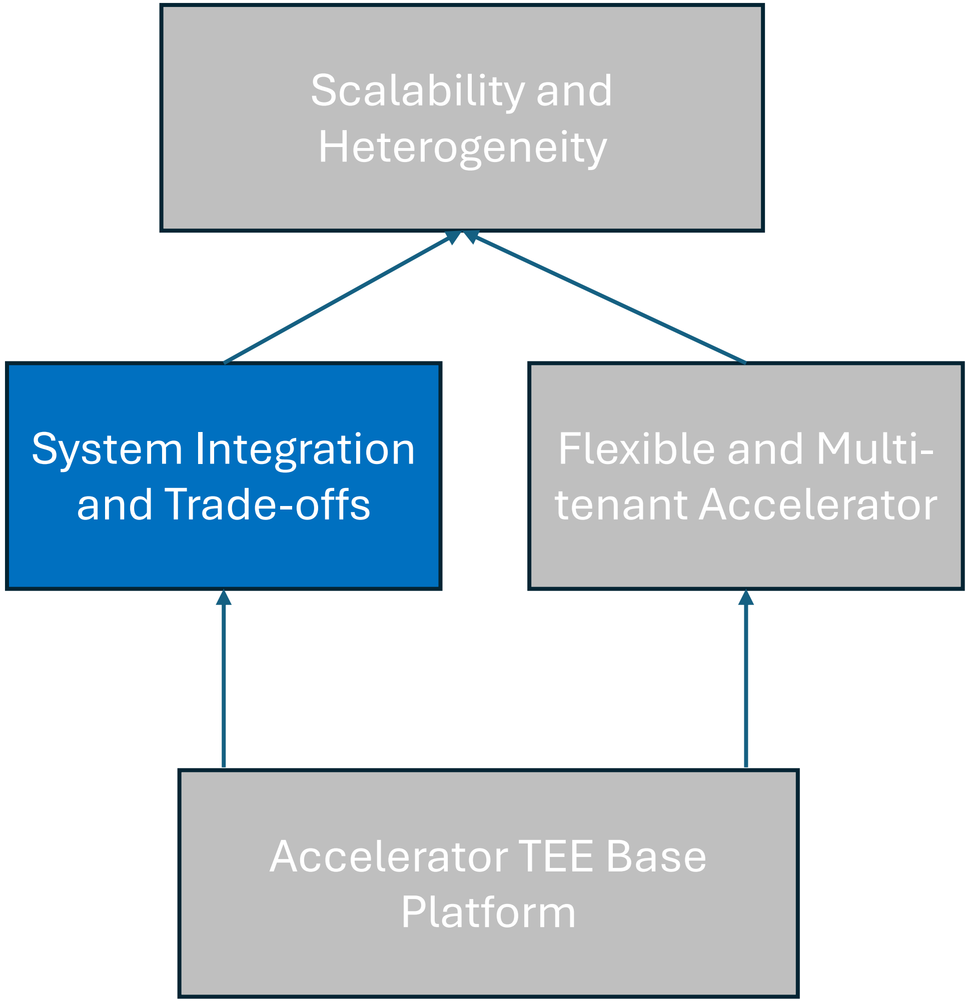




Research Questions

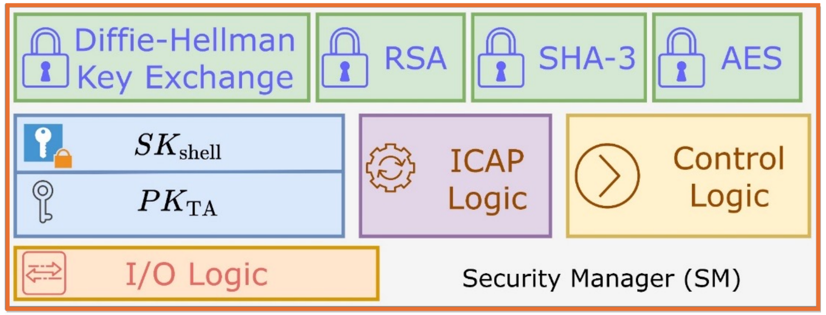
- How should we provide a secure execution environment or framework for AI accelerators?
- How should we design and integrate security solutions for AI accelerators?
- How should security solutions for AI accelerators adapt to future architectures that are dynamic and configurable?
- How should we design for scalable and heterogeneous systems?

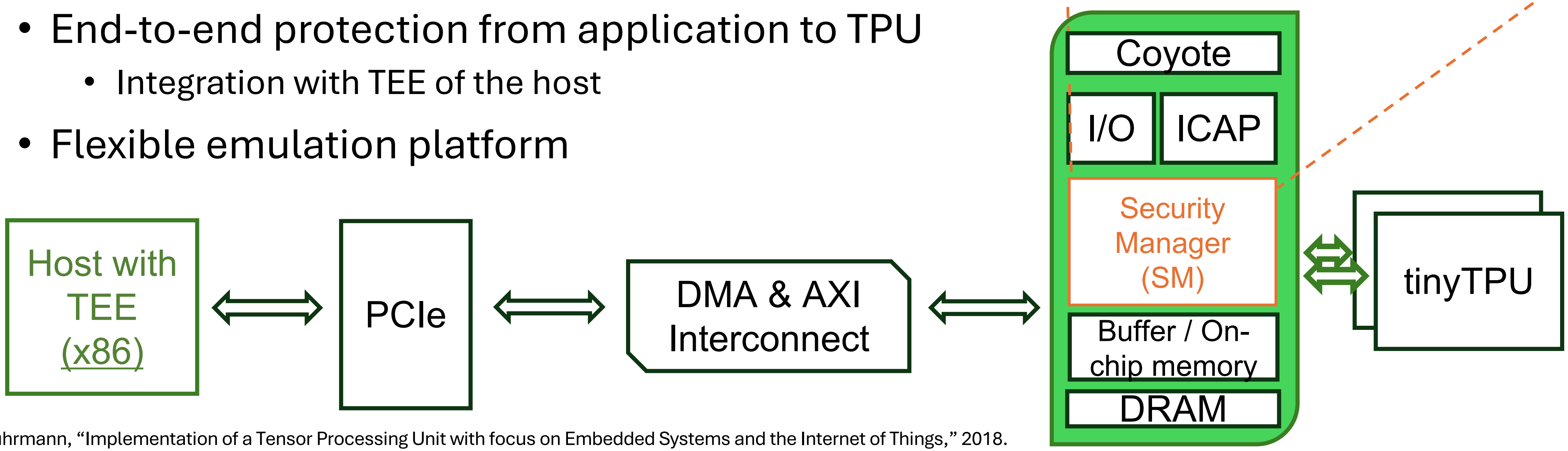
Security for AI Accelerators



AccShield – High-level Overview

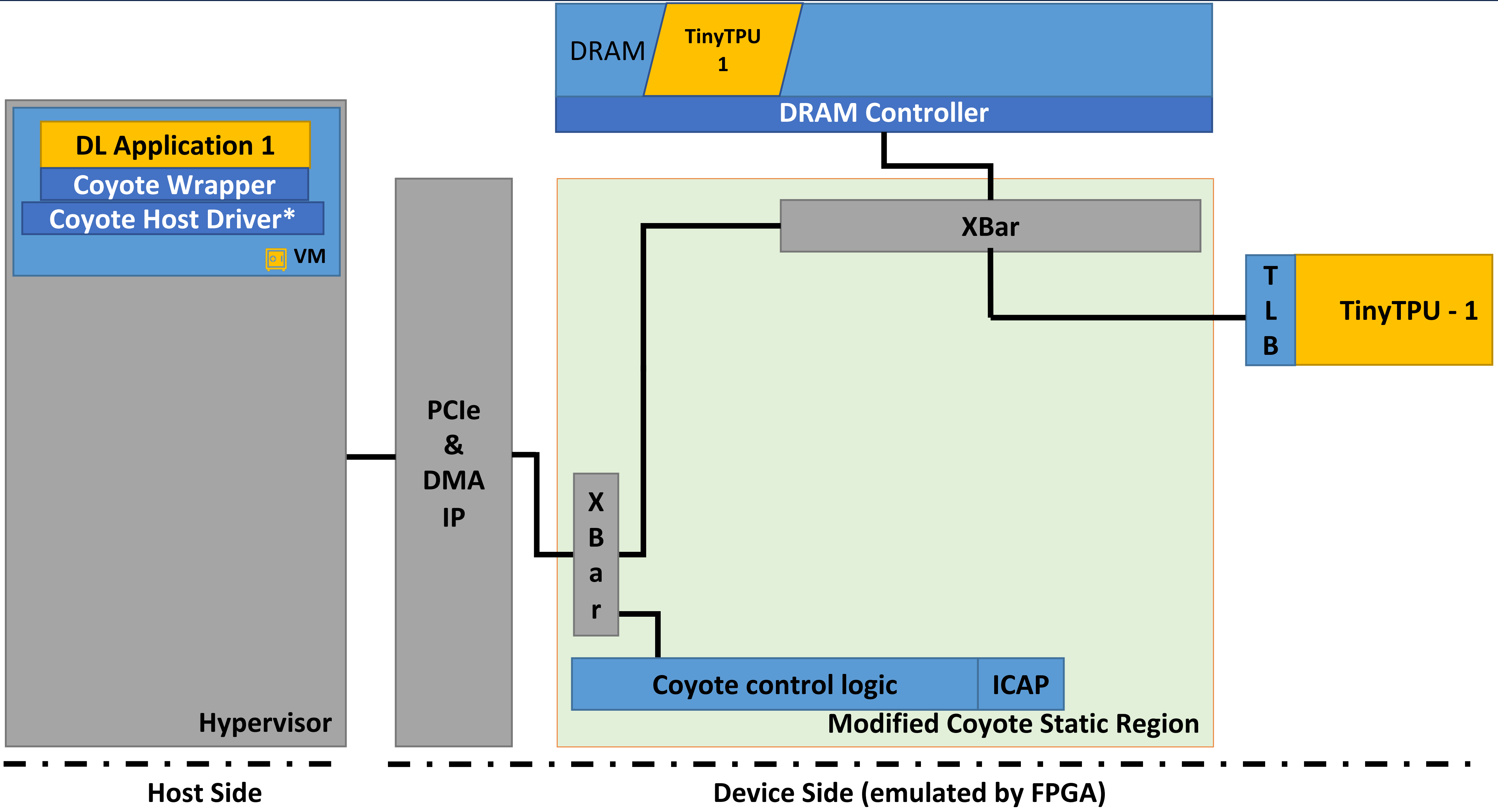
- tinyTPU[1] to simulate cloud TPUs
- Coyote[2] to support virtual memory for accelerators
- Improve system security for the cloud
  - Security features improved upon AccGuard[3]
- End-to-end protection from application to TPU
  - Integration with TEE of the host
- Flexible emulation platform



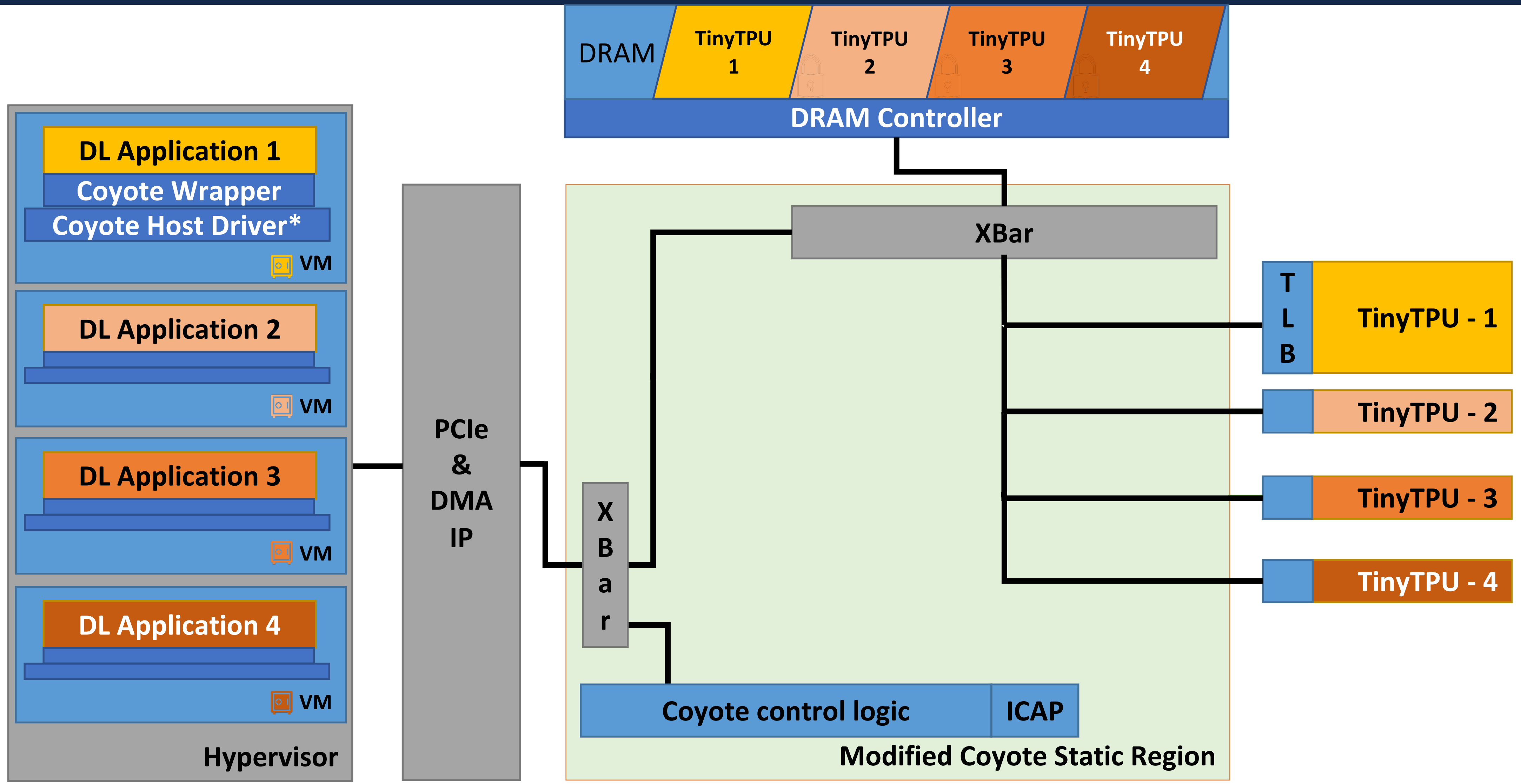


1. Fuhrmann, "Implementation of a Tensor Processing Unit with focus on Embedded Systems and the Internet of Things," 2018.
2. Korolija, T. Roscoe, and G. Alonso, "Do OS abstractions make sense on FPGAs?" in OSDI '20. USENIX, 2020, pp. 991–1010.
3. W. Ren, J. Pan and D. Chen, "AccGuard: Secure and Trusted Computation on Remote FPGA Accelerators," 2021 IEEE International Symposium on Smart Electronic Systems (iSES),

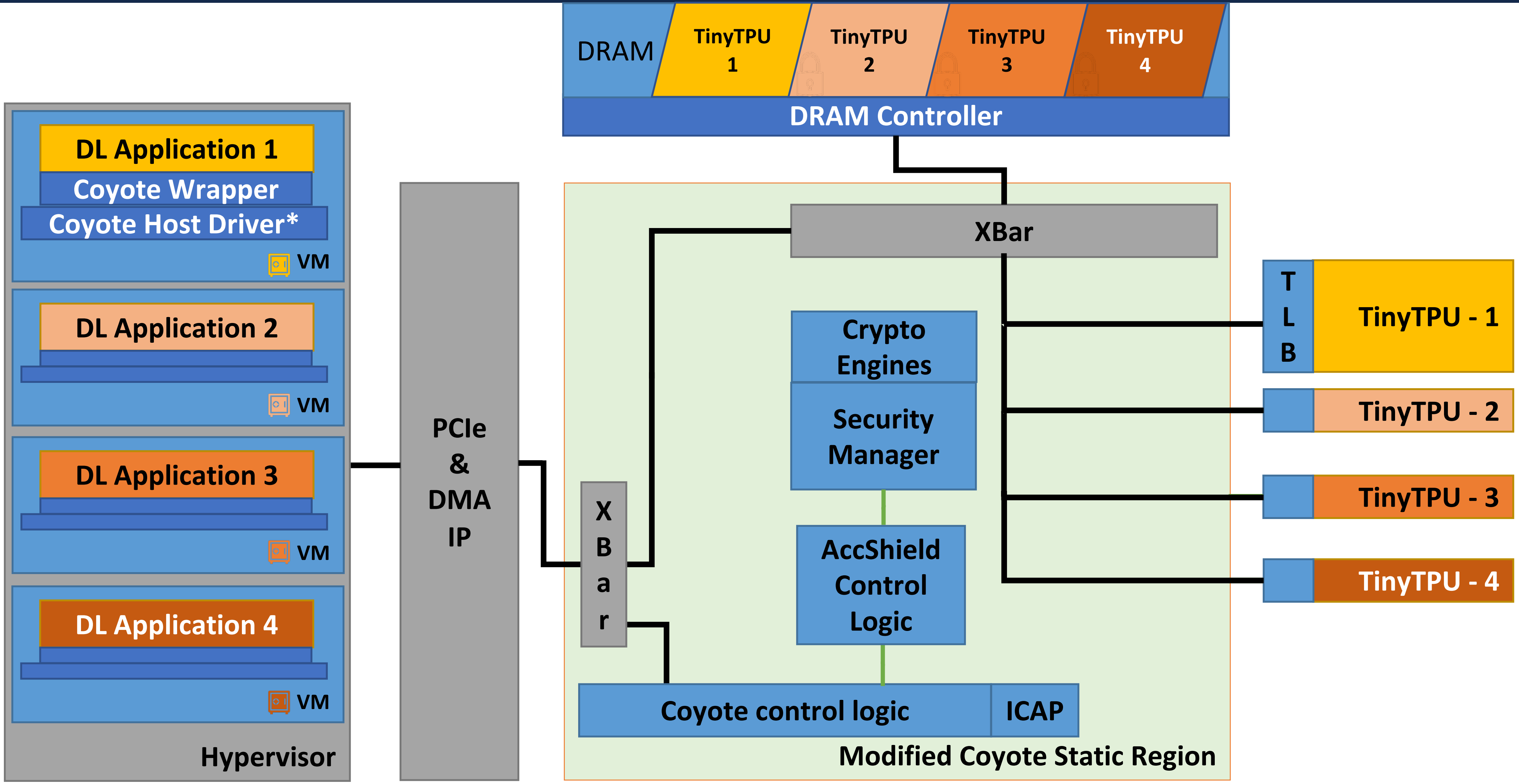
Architecture of AccShield



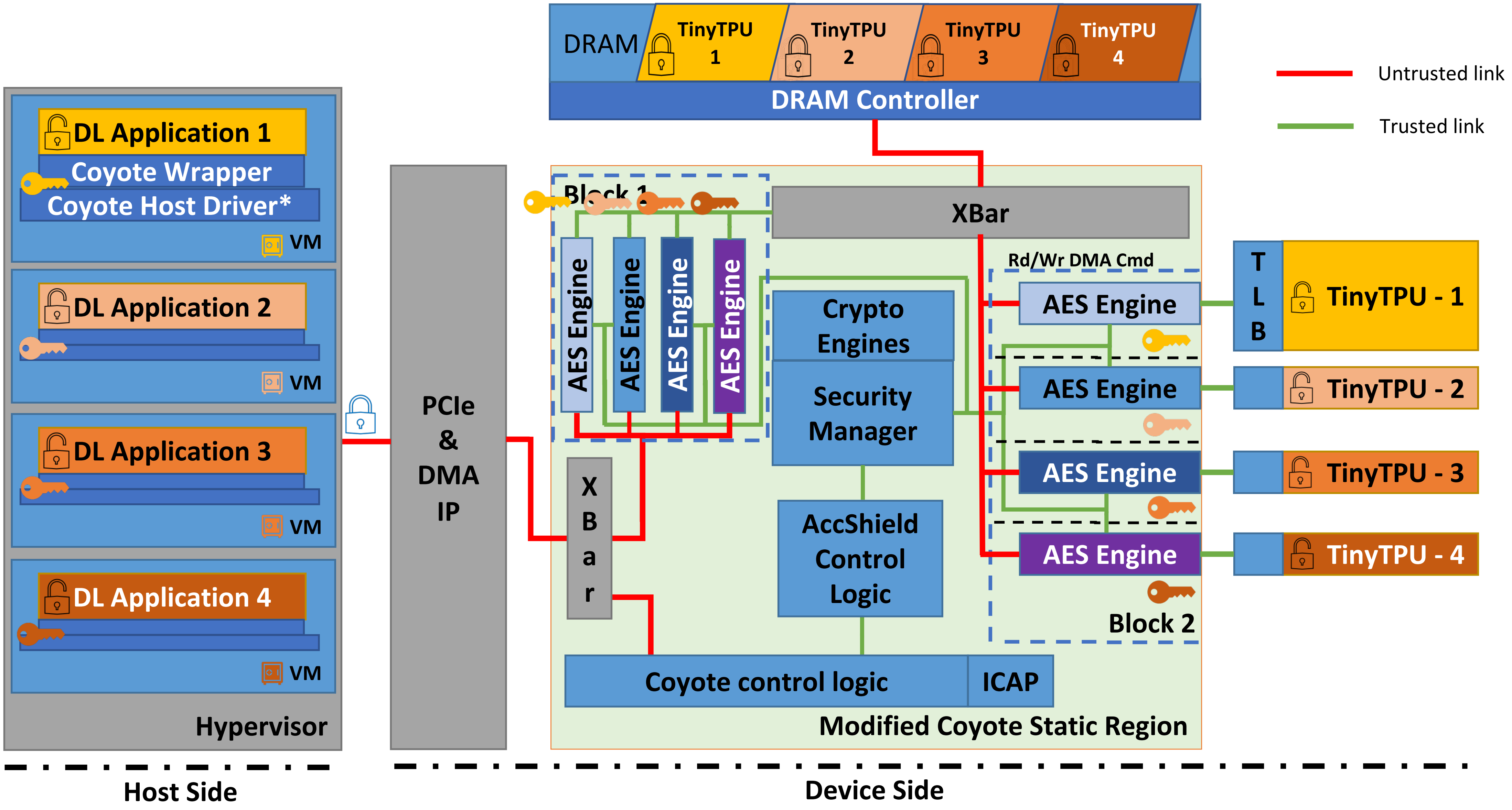
Architecture of AccShield



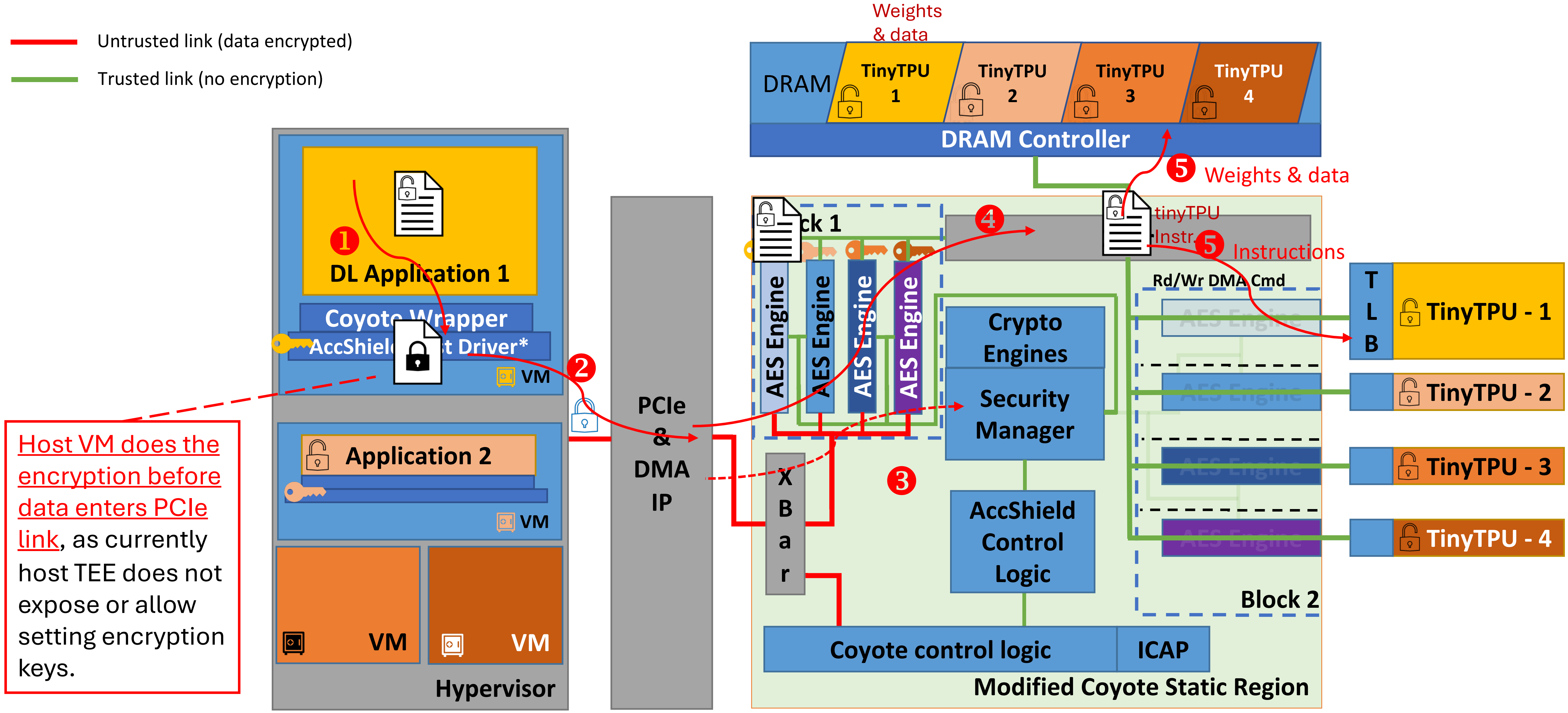
Architecture of AccShield



Architecture of AccShield

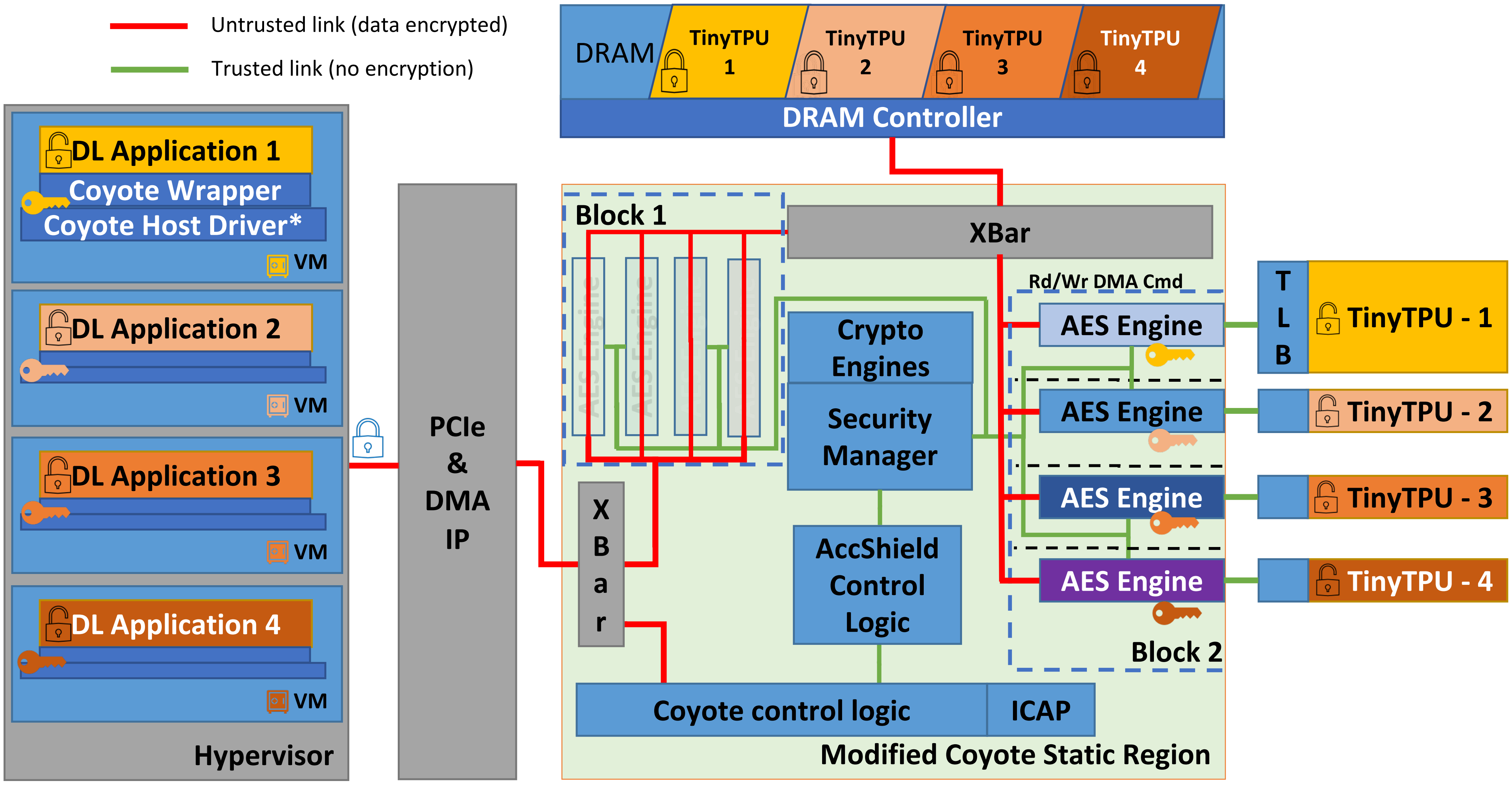


Partition-Based (Config. ②)

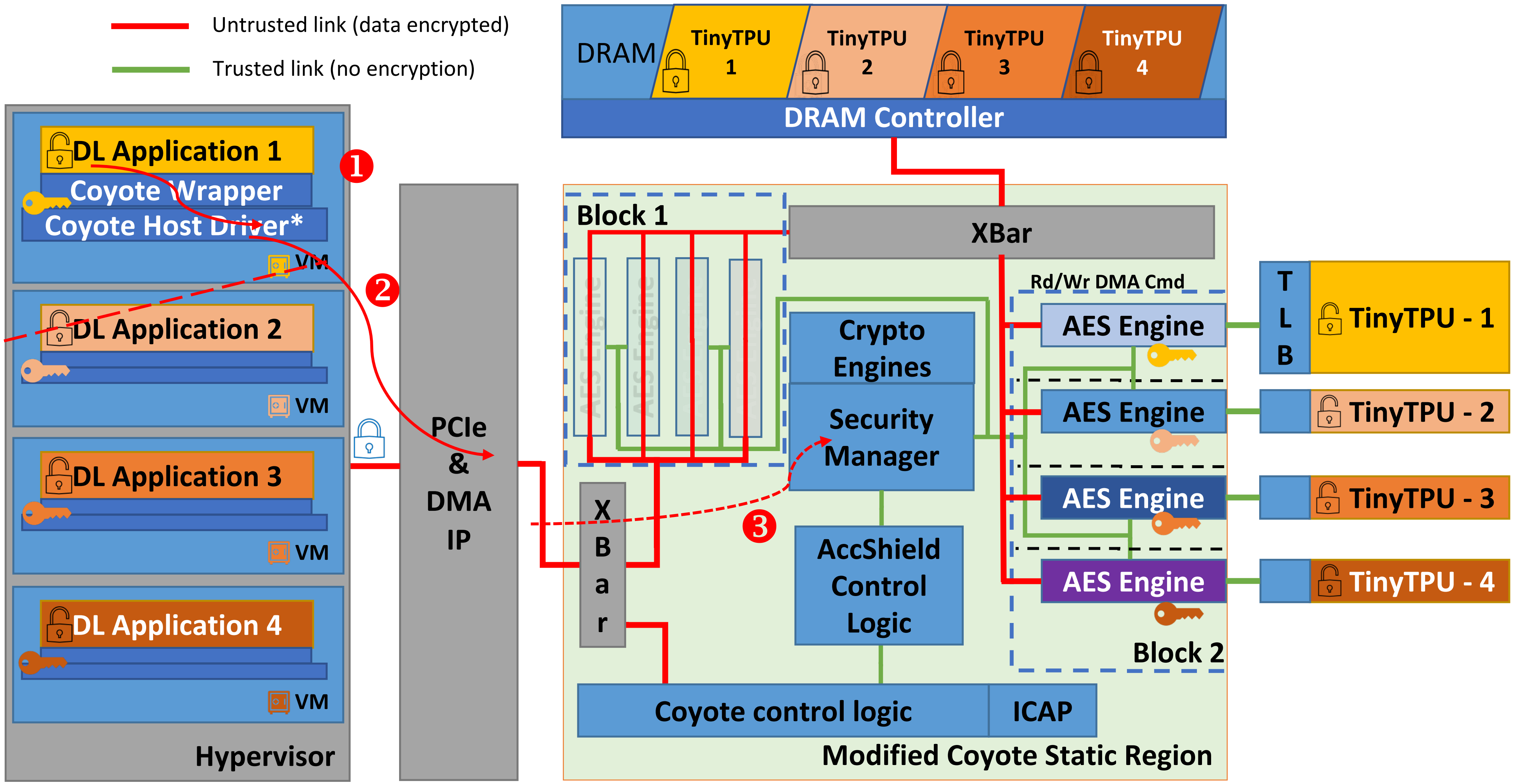


Host VM does the encryption before data enters PCIe link, as currently host TEE does not expose or allow setting encryption keys.

Encryption-Based (Config. ③)

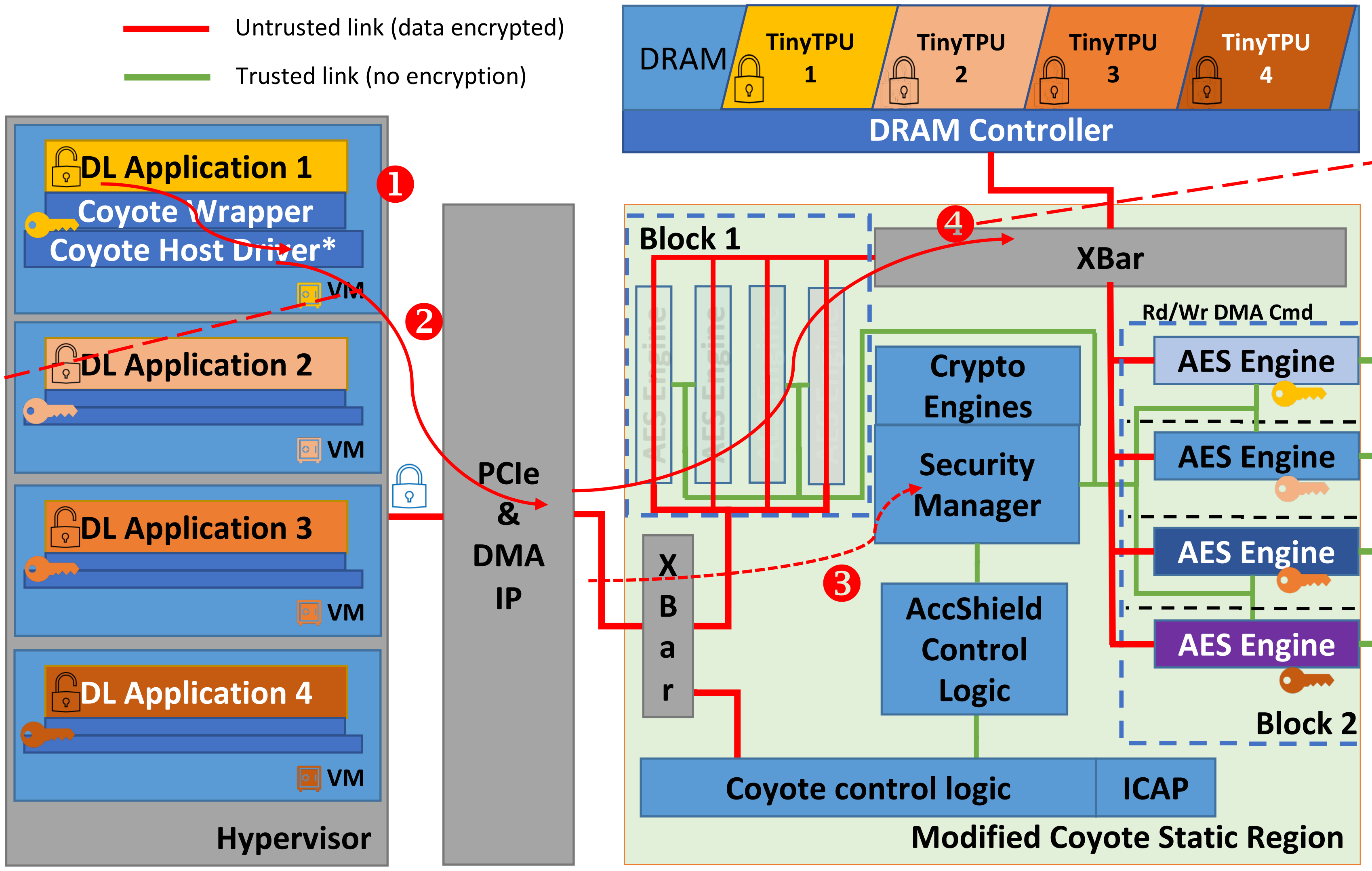


Encryption-Based (Config. ③)



Host does the encryption before data enters PCIe link, as currently host TEE does not expose or allow setting encryption keys.

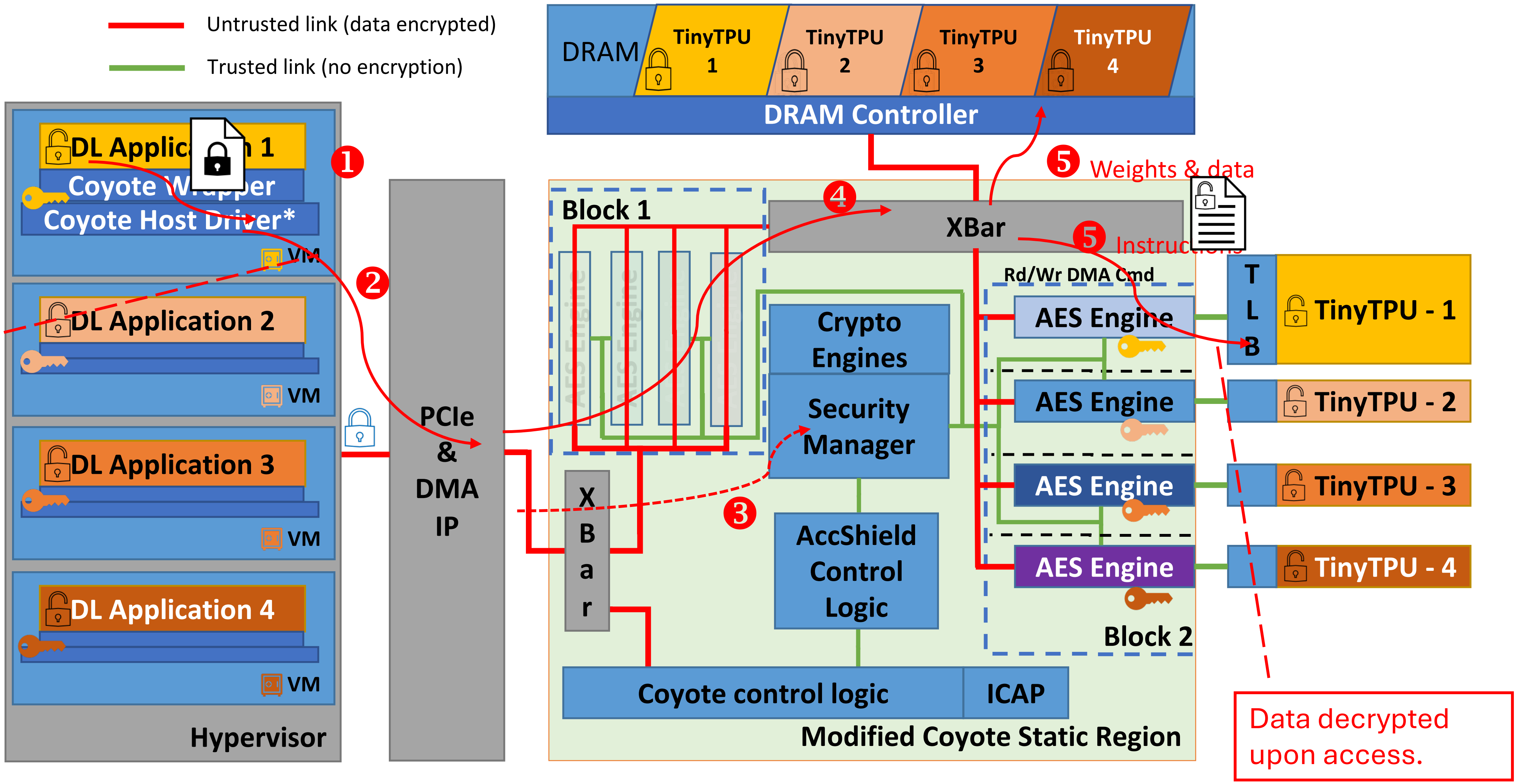
Encryption-Based (Config. ③)



Data remains Encrypted in Device memory

<u>Host does the encryption before data enters PCIe link</u>, as currently host TEE does not expose or allow setting encryption keys.

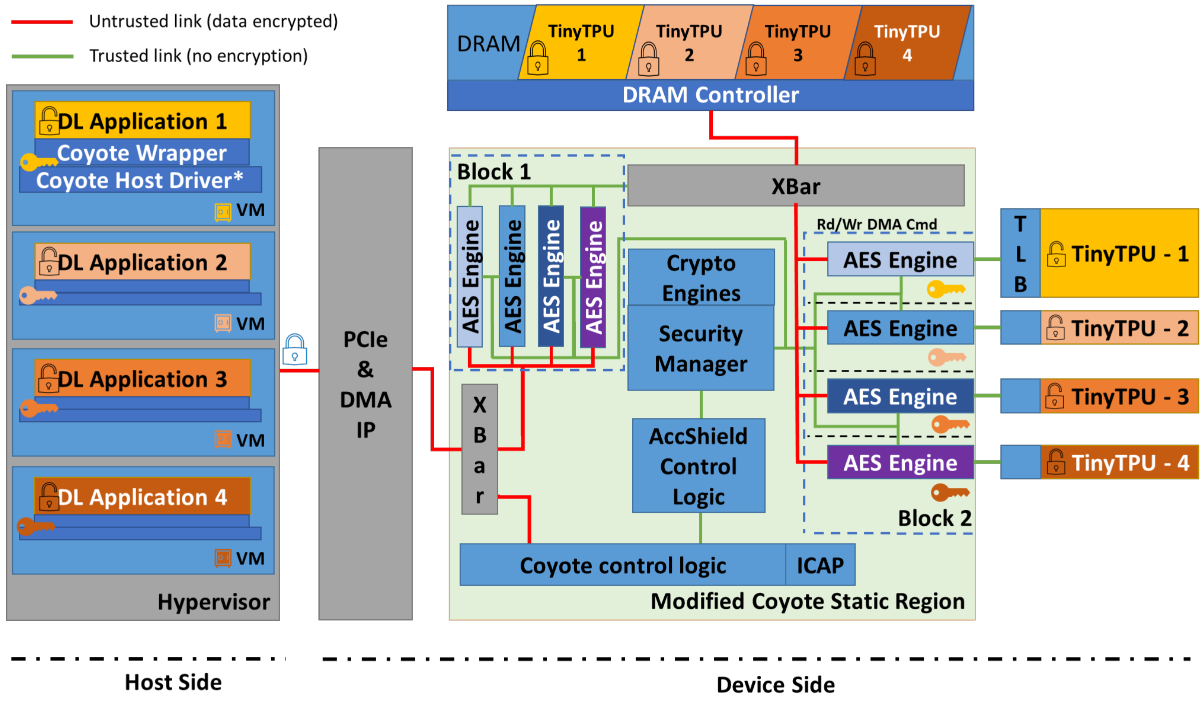
Encryption-Based (Config. ③)



Host does the encryption before data enters PCIe link, as currently host TEE does not expose or allow setting encryption keys.

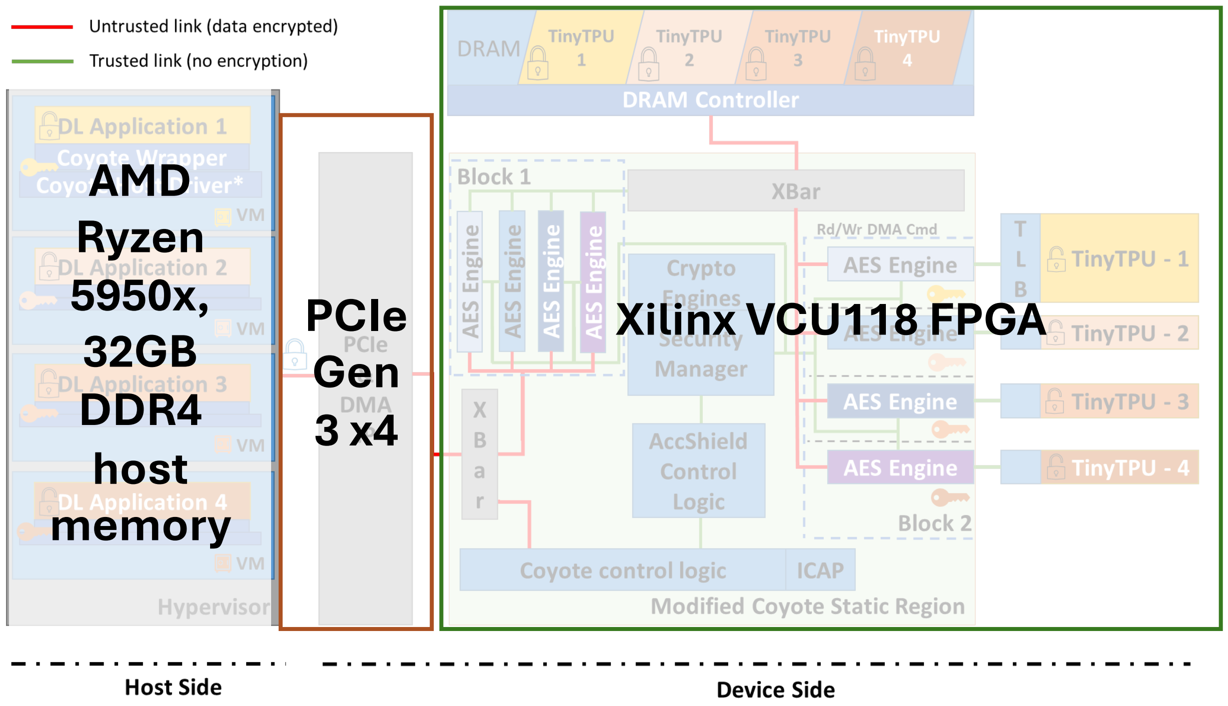
Experimental Setup

Config	Link Encryption?	Device Memory Encryption?	Block Engine 1	Block Engine 2
① Baseline	✗	✗	✗	✗
② Link Enc.	✓	✗	✓	✗
③ Link Enc. + Mem Enc.	✓	✓	✗	✓
④ Mem Enc.	✗	✓	✓	✓



Experimental Setup

Config	Link Encryption?	Device Memory Encryption?	Block Engine 1	Block Engine 2
① Baseline	✗	✗	✗	✗
② Link Enc.	✓	✗	✓	✗
③ Link Enc. + Mem Enc.	✓	✓	✗	✓
④ Mem Enc.	✗	✓	✓	✓



FPGA Resource Utilization of AccShield

	LUT	LUTRAM	FF	BRAM	DSP
AccShield	17.9%	2.41%	15.9%	15.7%	0.1%
Total (VCU118)	1182240	591840	2364480	2160	6840

Evaluation

- Partition-based design (②) incurs 4.11% overhead in FC and 0.9% in LeNet-5, much lower than device memory encryption-based design (③).
- On-chip memory/cache can significantly reduce overhead of device memory encryption.
- Device memory encryption will dominate the total overhead in future standards (e.g., TDISP).
- Demand paging still has large overhead in unified virtual memory.

Performance Result of Dense/Fully Connected Layer (784×504)

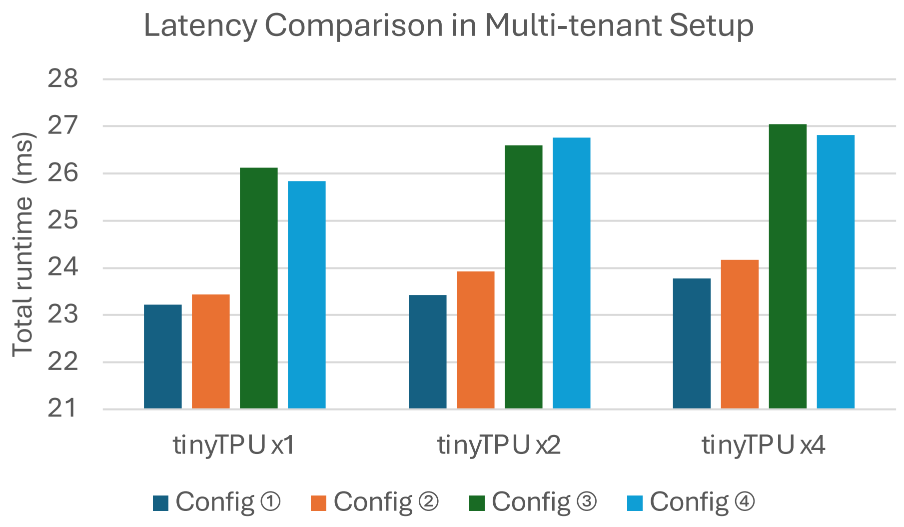
Configs	Host to Device Transfer (ms)			Layer Computation(ms)		Device to Host Writeback (ms)	Total Overhead*
	4KB Page	2MB Page	DMA	With OCM	No OCM		
Config ①	0.416	0.705	0.125	4.206	79.301	0.0035	Baseline
Config ②	4.630	1.484	0.284	4.225	79.434	0.0038	4.11%
Config ③	0.517	1.014	0.205	4.717	122.51	0.0038	13.64%
Config ④	4.531	1.137	0.217	4.692	122.97	0.0038	13.34%

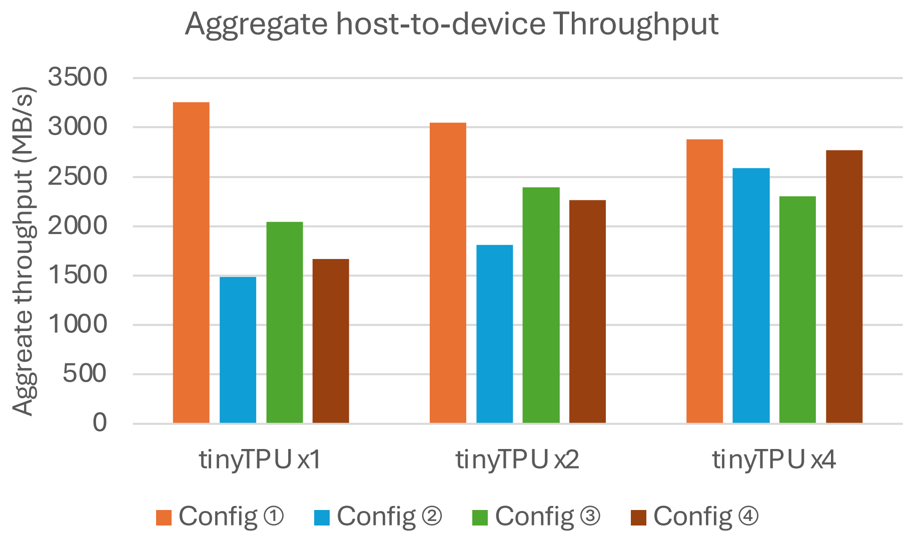
Performance Result of LeNet-5

Configs	Host to Device Transfer (ms)			Layer Computation (ms)	Device to Host Writeback (ms)	Total Overhead *
	4KB Page	2MB Page	DMA			
Config ①	0.218	0.709	0.065	23.15	0.0028	Baseline
Config ②	2.318	1.492	0.141	23.29	0.0065	0.9%
Config ③	0.253	1.032	0.103	26.02	0.0058	12.53%
Config ④	2.283	1.161	0.126	25.71	0.0054	11.3%

Multi-tenancy

- Overall latency overhead per tenant increases by ~3.5% (worst case)
- For link encryption (②), the increase in the overhead is not proportional to number of tenants
- With multiple channels, multi-tenancy can help better utilize PCIe link throughput
  - Even though single AES engine bottlenecks the channel throughput



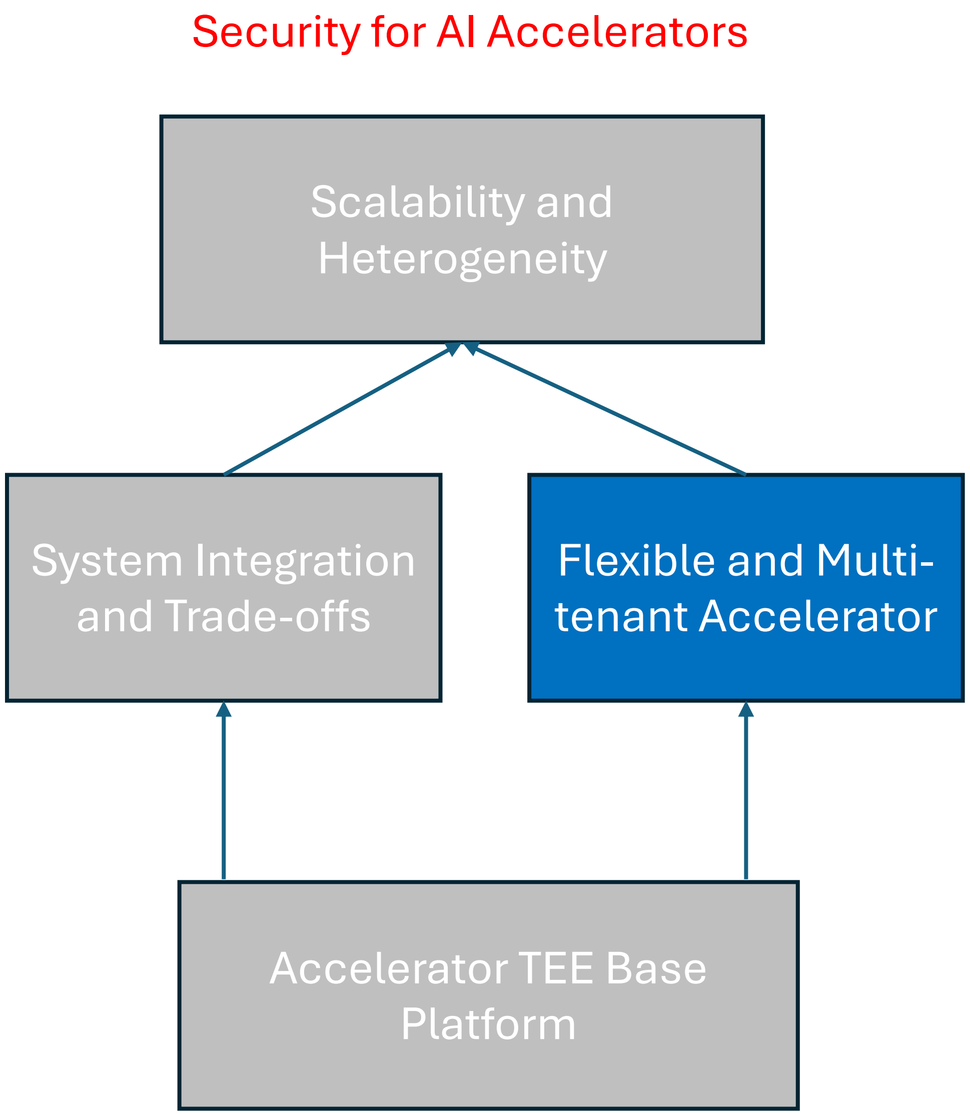


Observations and Takeaways

- AccShield and its prototype design demonstrate feasibility of:
  - Strong security for ML accelerators in the cloud
  - Relatively low performance impact (~4% for link encryption, ~13% for link & device encryption)
- Compared to device memory encryption overhead, partition-based memory protection offers TEE solution with significantly lower overhead for accelerators
- Memory encryption is heavily dependent on the size of data and availability of cache for TPU-like accelerators
- Open-source design

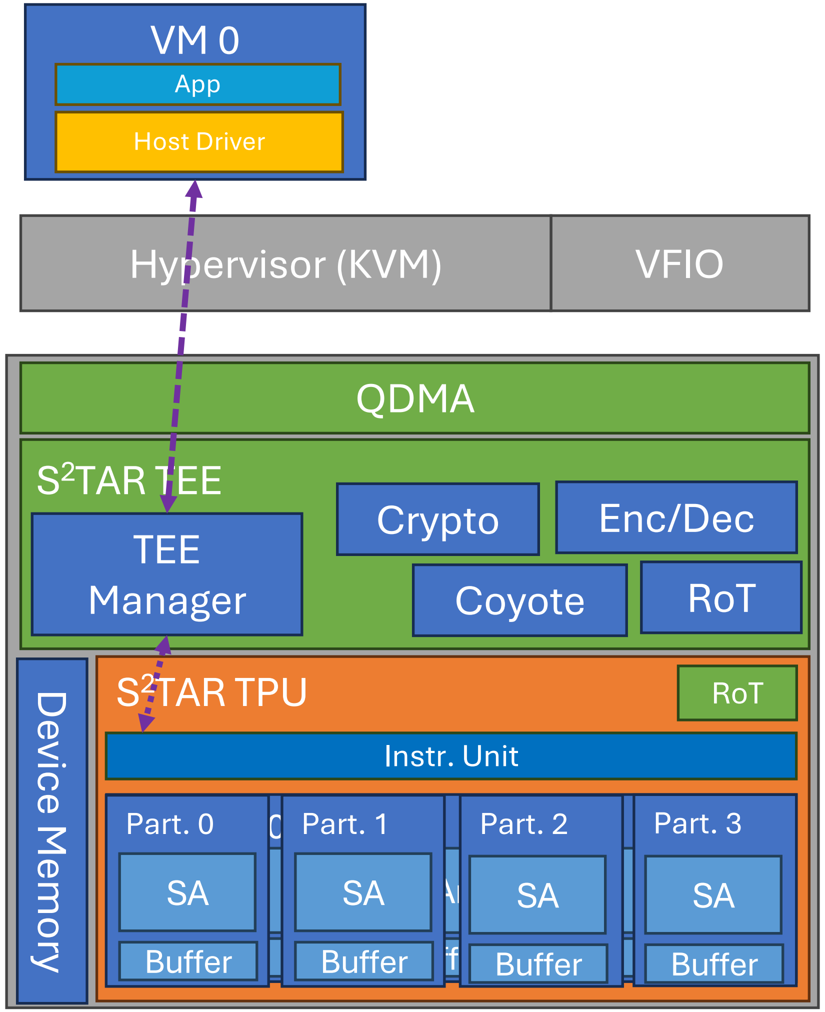
Research Questions

- How should we provide a secure execution environment or framework for AI accelerators?
- How should we design and integrate security solutions for AI accelerators?
- How should security solutions for AI accelerators adapt to future architectures that are dynamic and configurable?
- How should we design for scalable and heterogeneous systems?



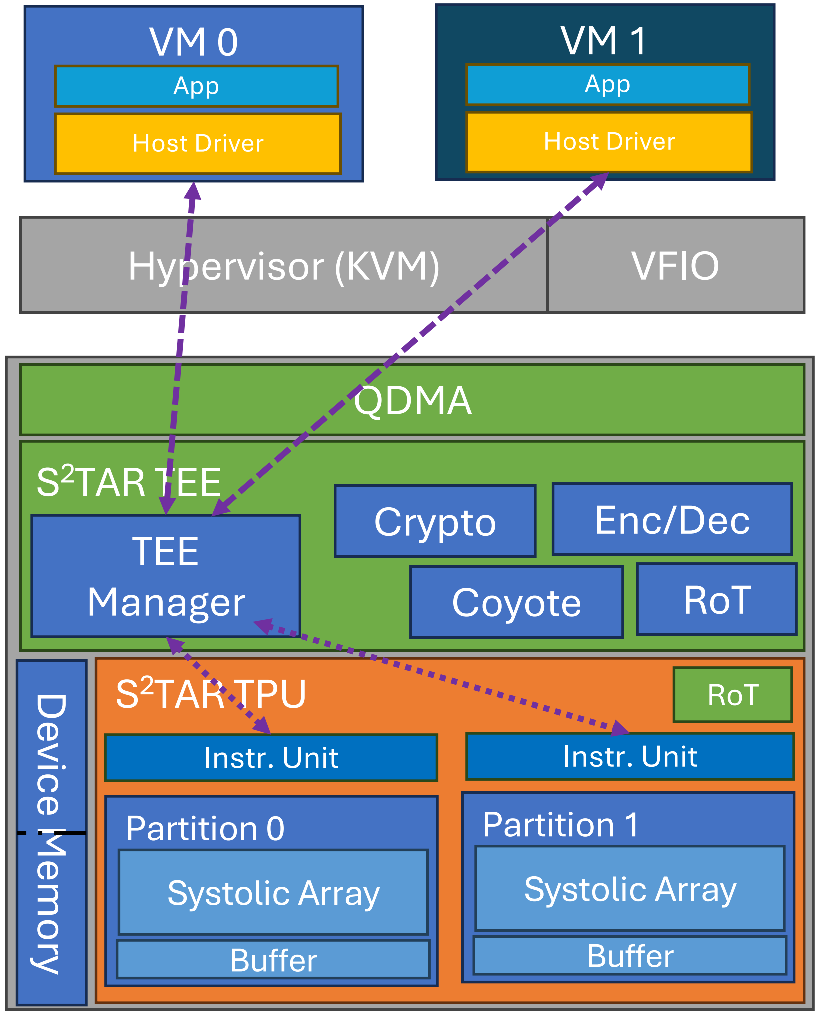
S^{2}TAR Trusted Execution Environment

- With dynamic partitions, S^{2}TAR offers:
  - "Reshape" TPU to use the best configuration for different workloads (i.e., MatMul of different sizes), at runtime.



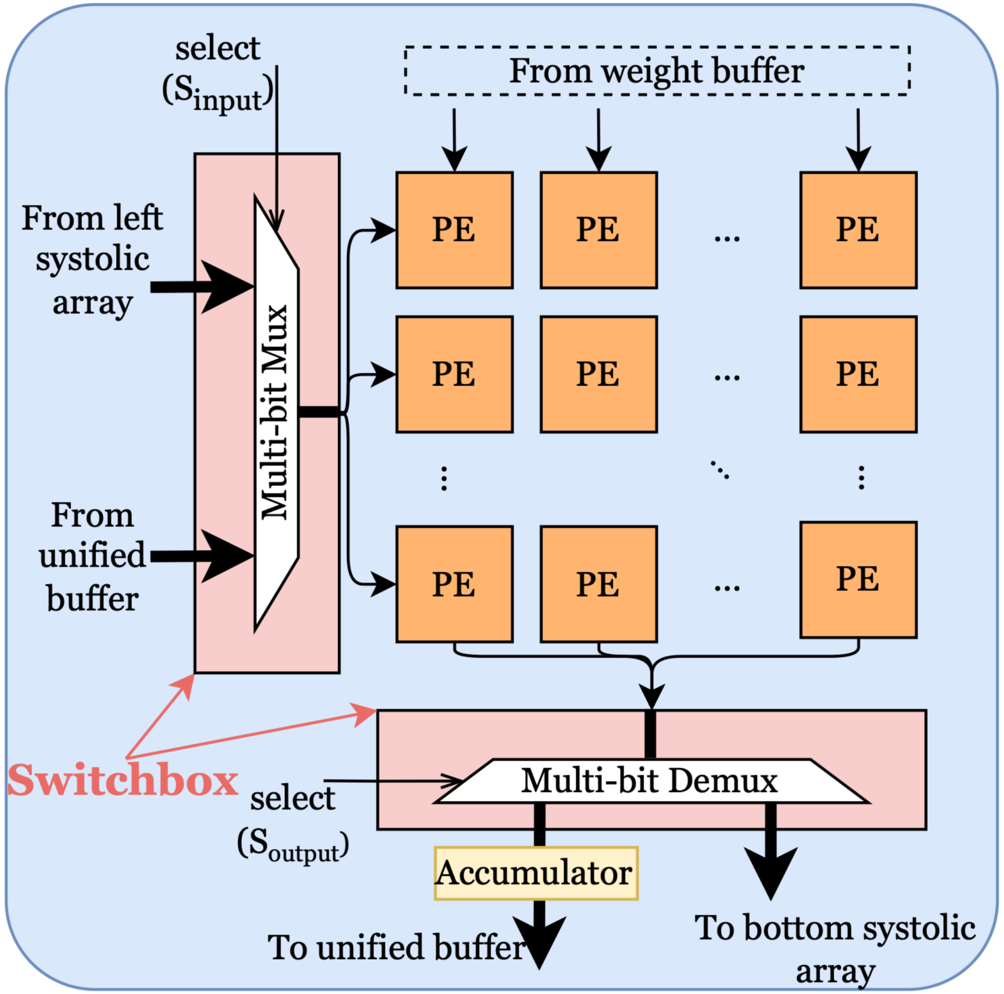
S²TAR Trusted Execution Environment

- With dynamic partitions, S²TAR offers:
  - "Reshape" TPU to use the best configuration for different workloads (i.e., MatMul of different sizes), at runtime.
  - Isolated partitions that allow sharing TPU spatially among tenants
- TEE Framework provides:
  - VMs with PCIe Physical Function (PF) passthrough
  - Interface to interact with TPU partition TEE
  - Partition-level attestation



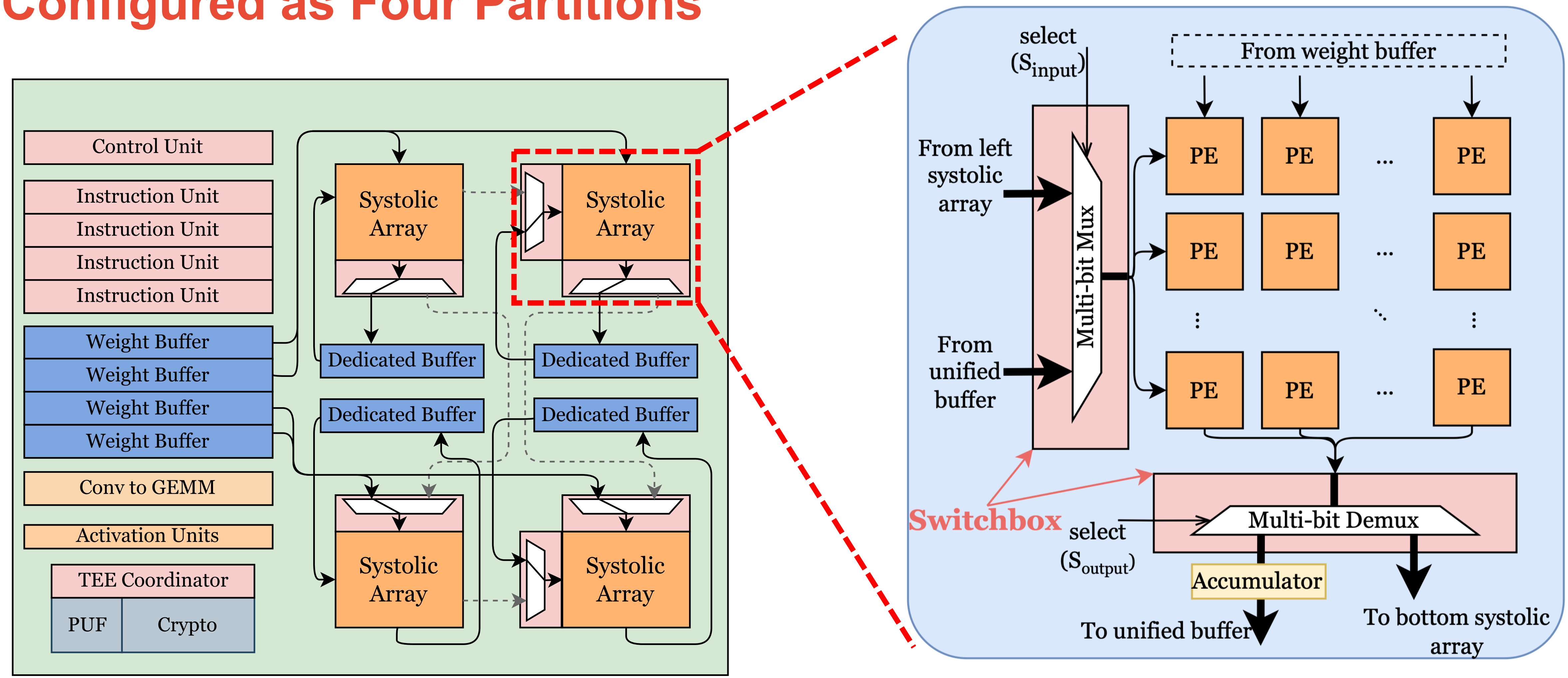
Runtime-Reconfigurable TPU - Systolic Array

- Each systolic array comes with switchboxes (muxes/de-muxes)
- Separate buffers for each systolic array
- Lightweight buffer (BRAM) controller
- Reconfiguration
  - Changing the select signals
  - Updating the configuration register



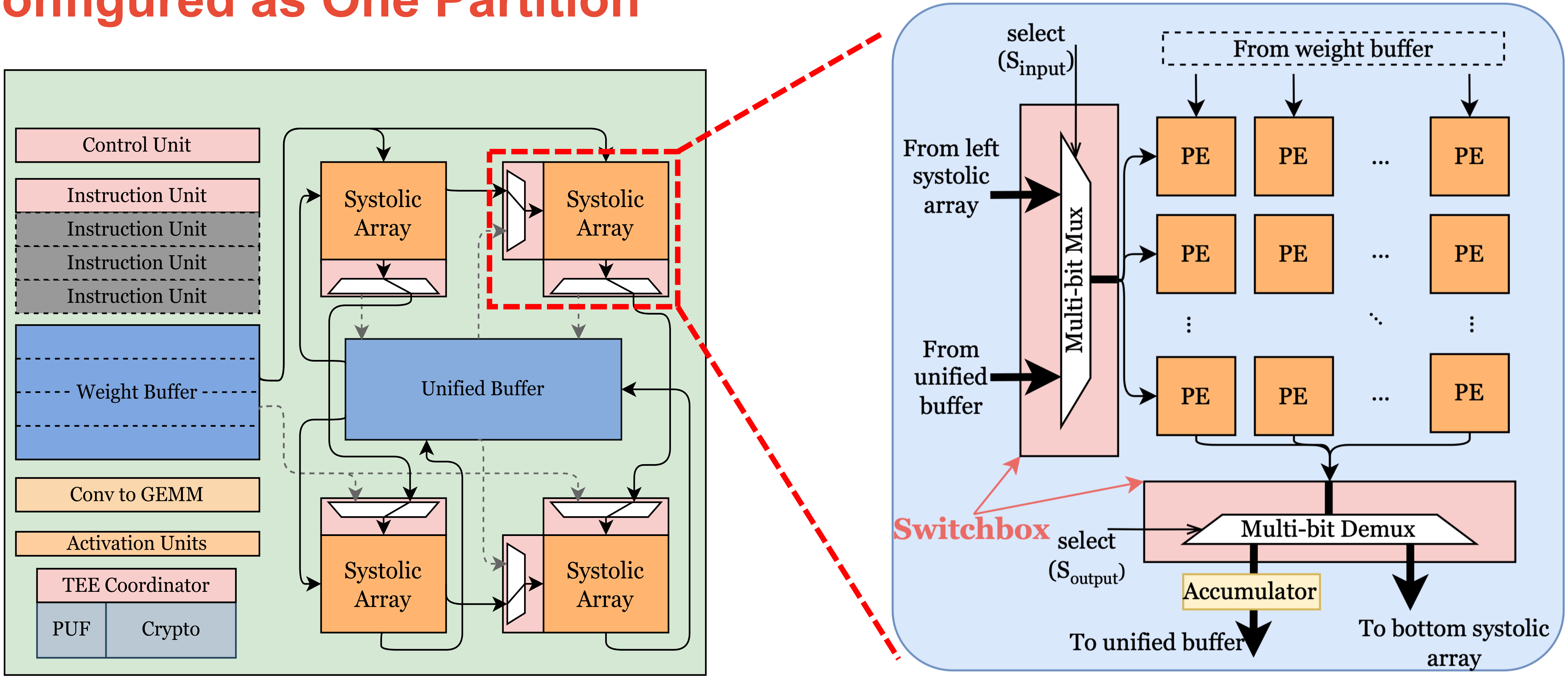
Runtime-Reconfigurable TPU

Configured as Four Partitions



Runtime-Reconfigurable TPU

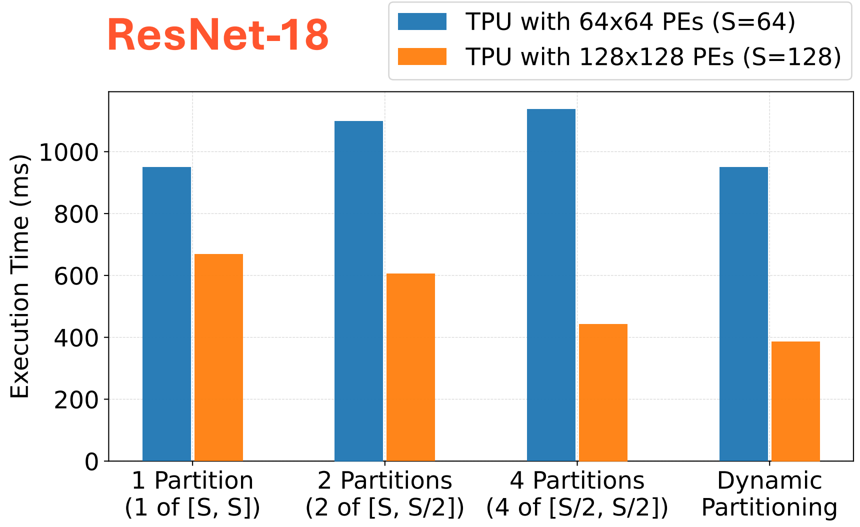
Configured as One Partition

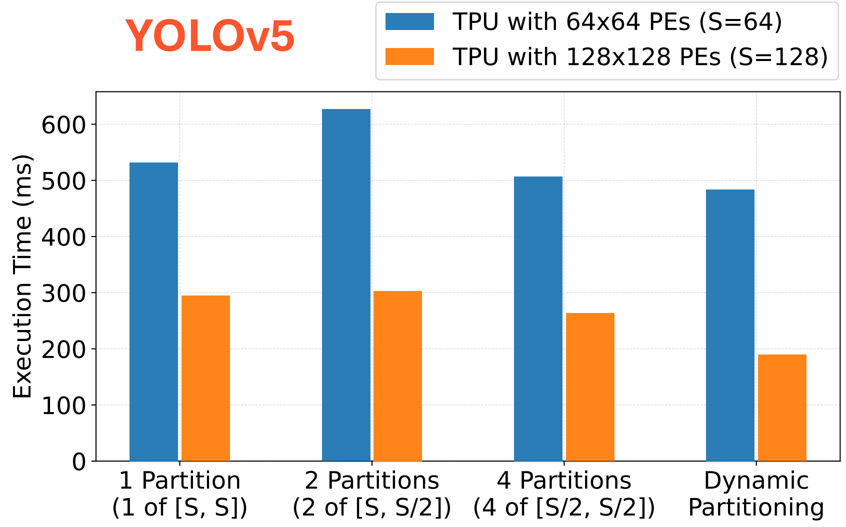


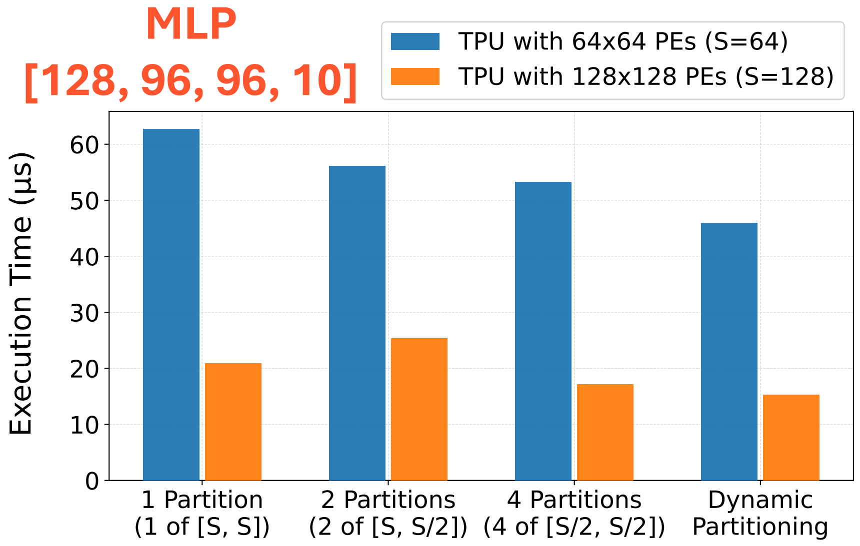
Partition-level Attestation

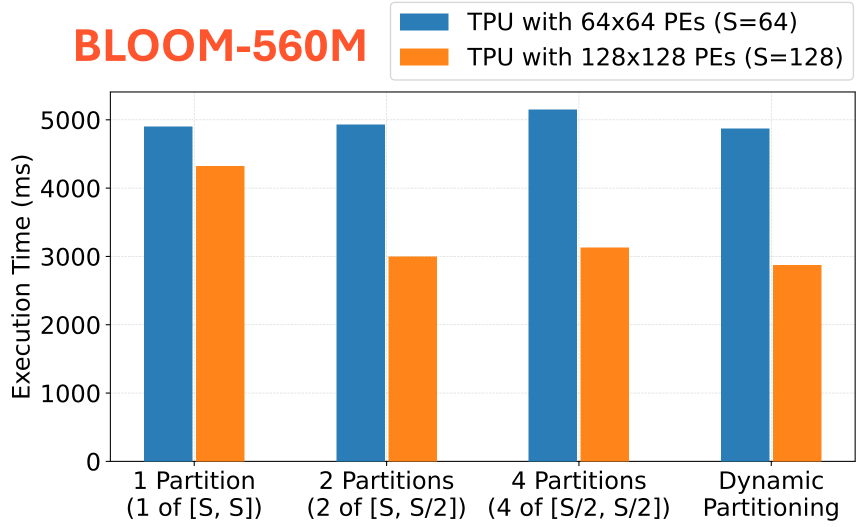
- **Finer granularity** - attestation of individual sub-device partitions within the TPU
- **Improved attestation efficiency** - decouples host and accelerator TEE attestation
- Attestation with dynamic partition update:
    1. Dynamic partitioning triggers internal configuration register update
    2. Attestation report includes configuration details for verification
    3. Signed with unique key derived from TPU's Root of Trust (RoT)
        - PUF-based RoT guarantees report authenticity and integrity

Model Benchmarks



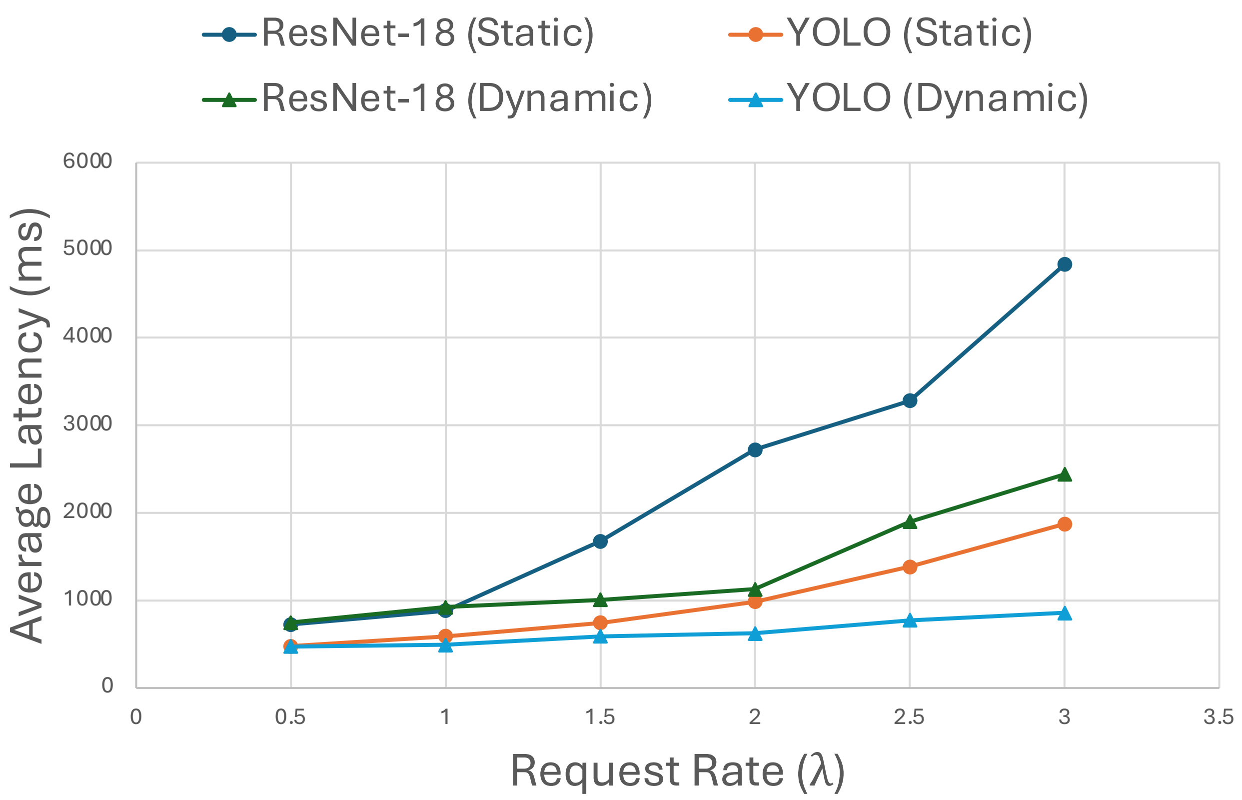






Multi-tenant Experiments

- Enhances overall task throughput and reduces tail latency by partitioning TPUs for concurrent requests.
  - Reduced average latency up to 17.5%
  - Reduced worst-case latency up to 33.1%



t: Time interval (e.g., 1 second)

λ: Requests per second

Requests follow a Poisson distribution

Measure how long each request takes (on average)

Observations and Takeaways

- **Dynamic Reconfiguration:**
  - Performance improvement with partitions adapted according to workload requirements
  - Effective utilization improvement by reducing idle PEs
- **Multi-tenant Security:** Enforces isolation and leverages TEEs for secure execution within partitions
- **Fine-grained Trust with Remote Attestation:** Extends trust guarantees to individual partitions, enabling fast re-attestation after dynamic reconfiguration
- **Open-source Design**

AI for Security

Leveraging AI for Enhanced Security

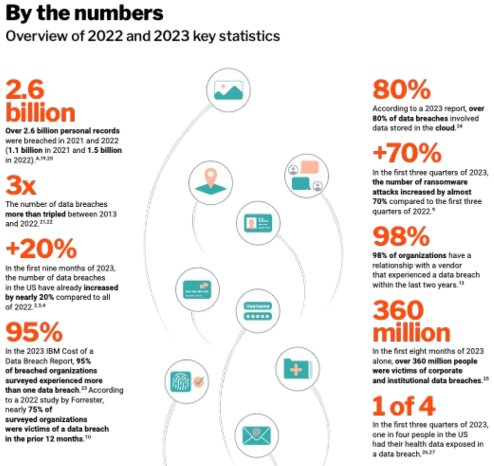
- **Real-Time Threat Detection**: Using AI to identify and mitigate cyber threats
- **Real-Time Monitoring and Response**: AI-driven security monitoring

Our focus today

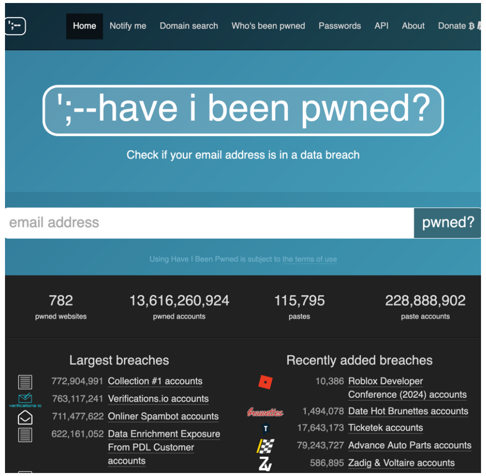
- **Predictive Analysis**: AI's ability to forecast potential security breaches
- **Automated Response Systems**: AI-driven frameworks for rapid incident response
- **Feedback Loop**: Continuous improvement and updates to models and systems

Threats in Critical Systems and Data Centers

23andMe Data Breach[1]



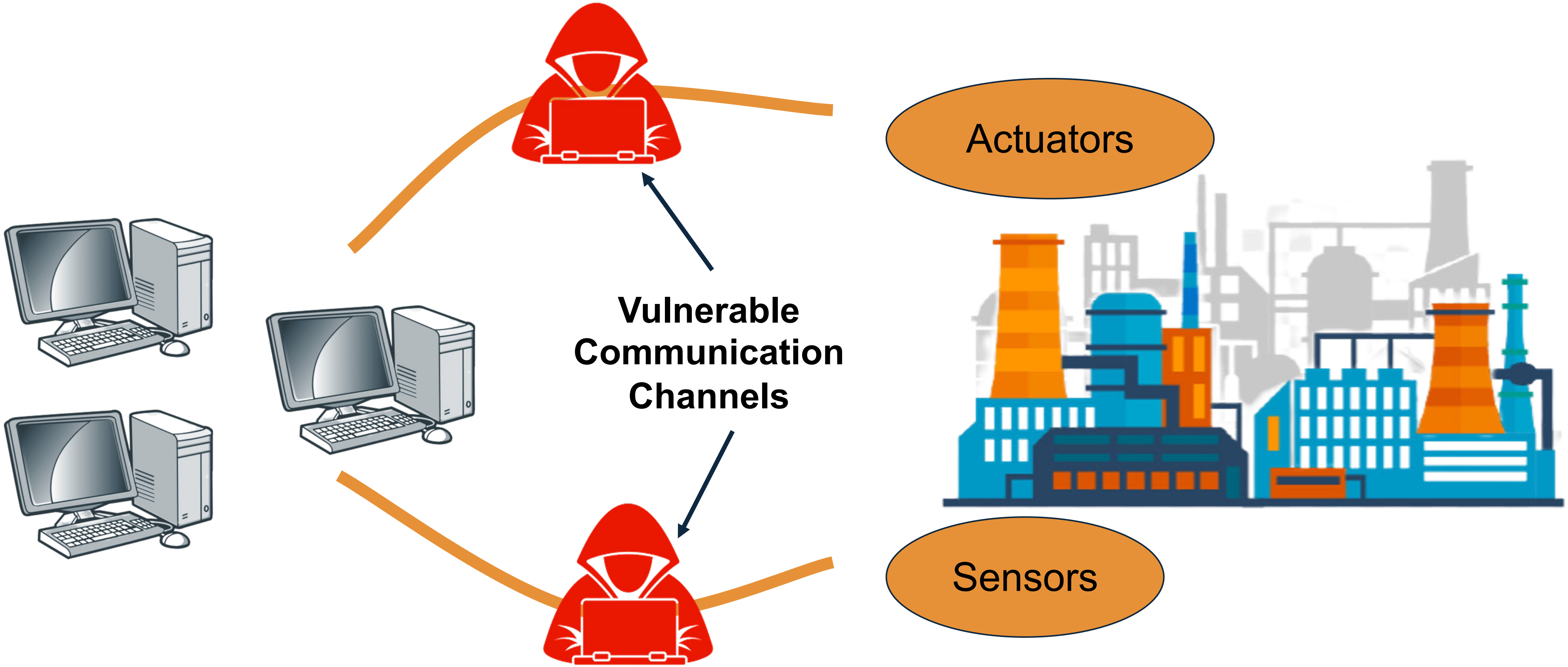
The Continued Threat to Personal Data[2]



Have I been pawned?[3]

1: https://techcrunch.com/2023/12/04/23andme-confirms-hackers-stole-ancestry-data-on-6-9-million-users/
2: Stuart E. Madnick, *The Continued Threat to Personal Data: Key Factors Behind the 2023 Increase*, Dec 2023
3: https://haveibeenpwned.com/

One Example on Cyber Physical Systems



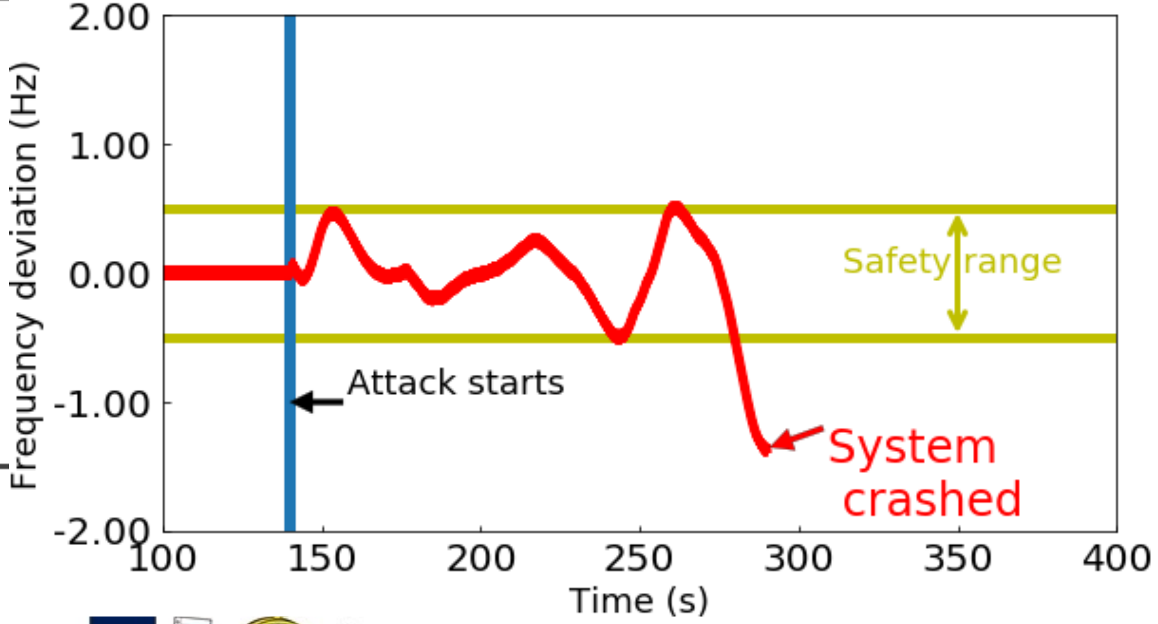
H. Farhangi, “The path of the smart grid”, IEEE power and energy magazine, vol. 8, no. 1, pp. 18–28, 2009.

One Attacking Scenario: the Delay Attack

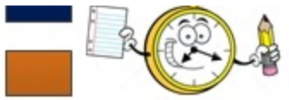
Tom

Alice

Delay attack in automatic generation control (AGC)



Tom

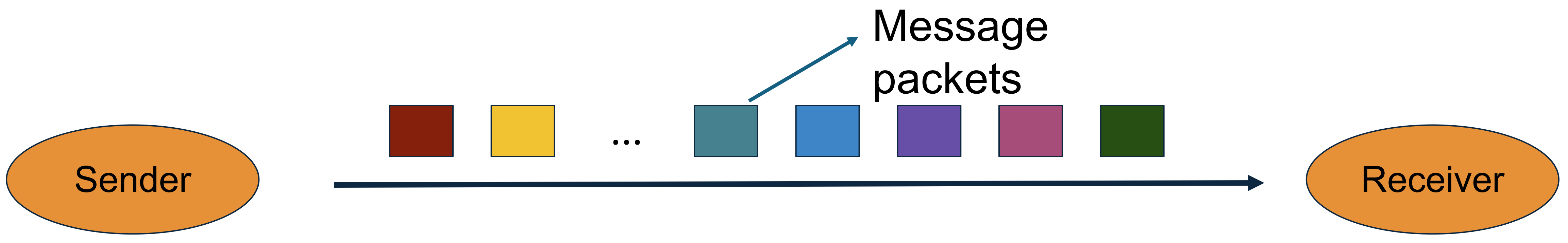


...

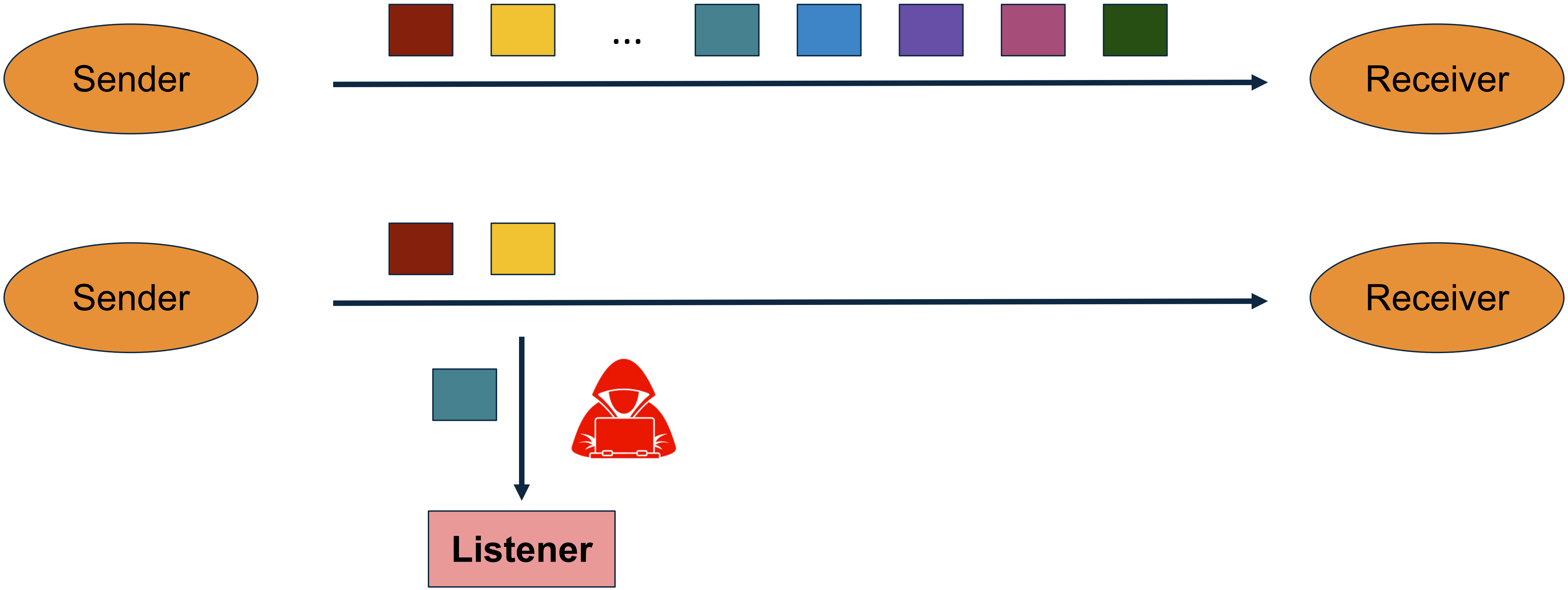
Sending packets/commands from time T

Packet interval is τ

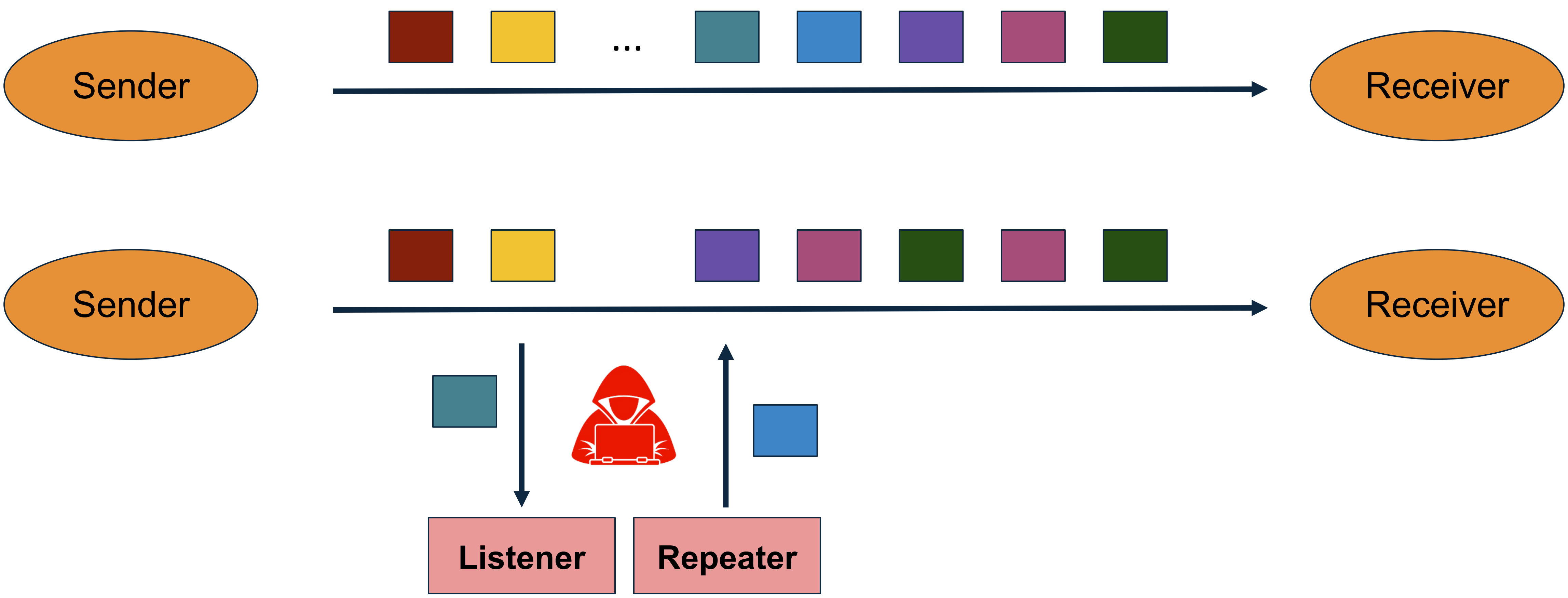
Time Delay Attack



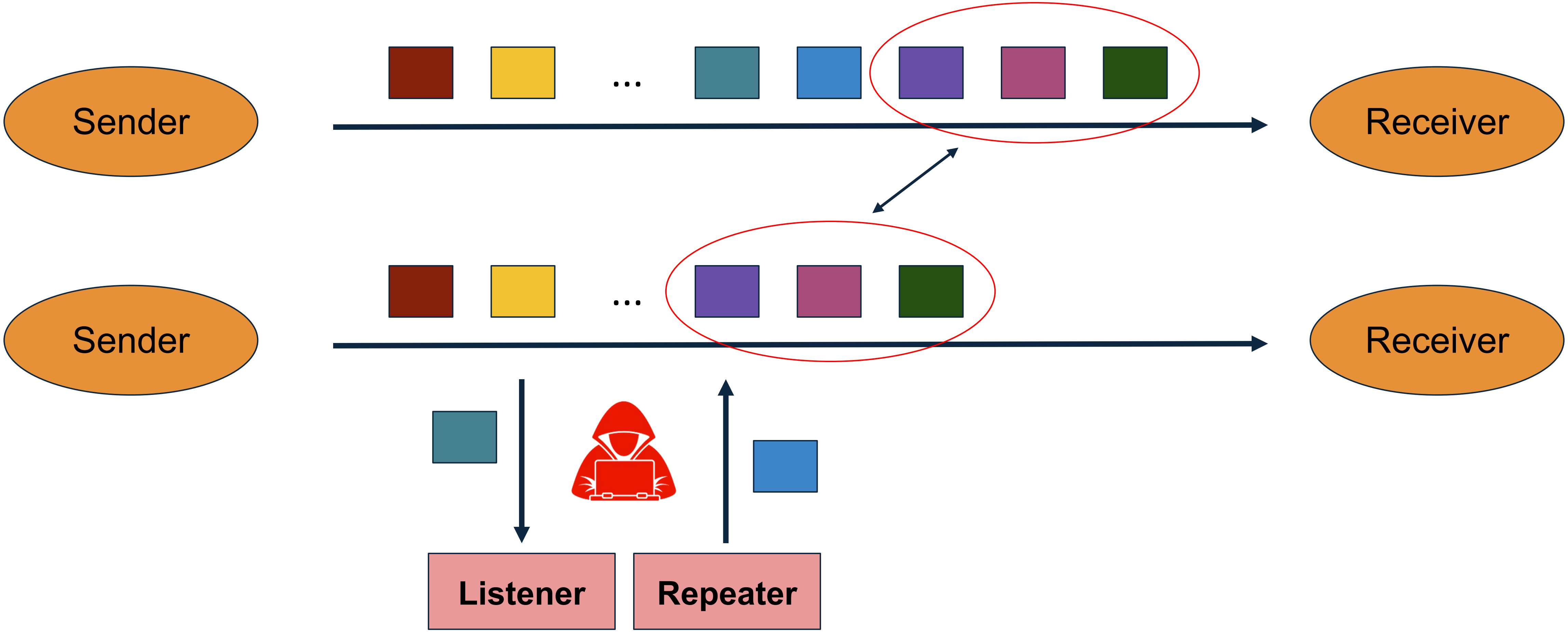
Time Delay Attack



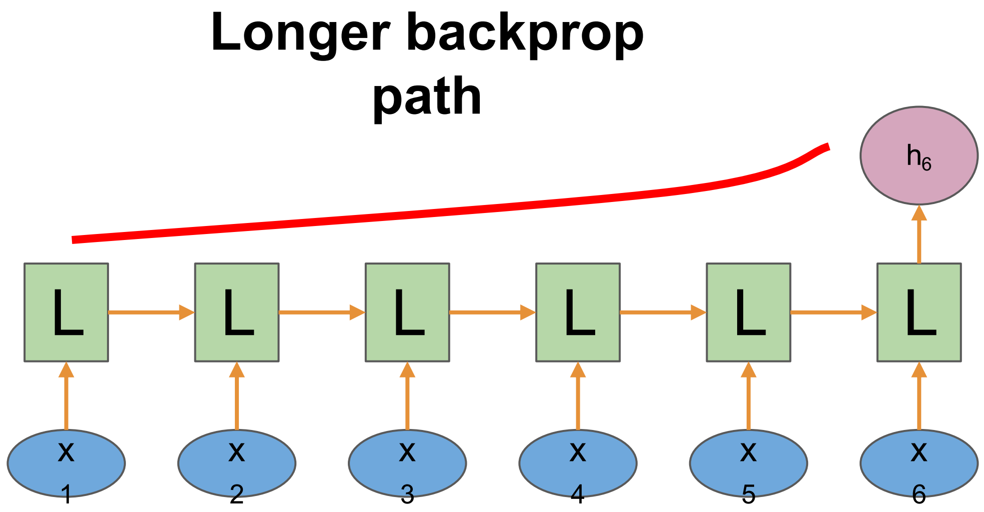
Time Delay Attack



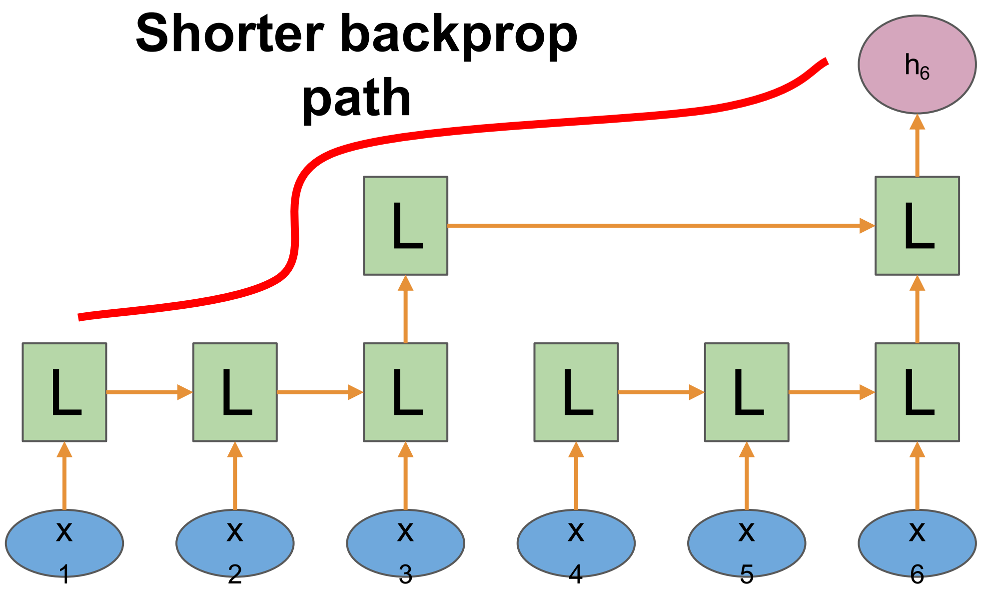
Time Delay Attack



Hierarchical LSTM for Real-Time TDA Detection



Traditional LSTM

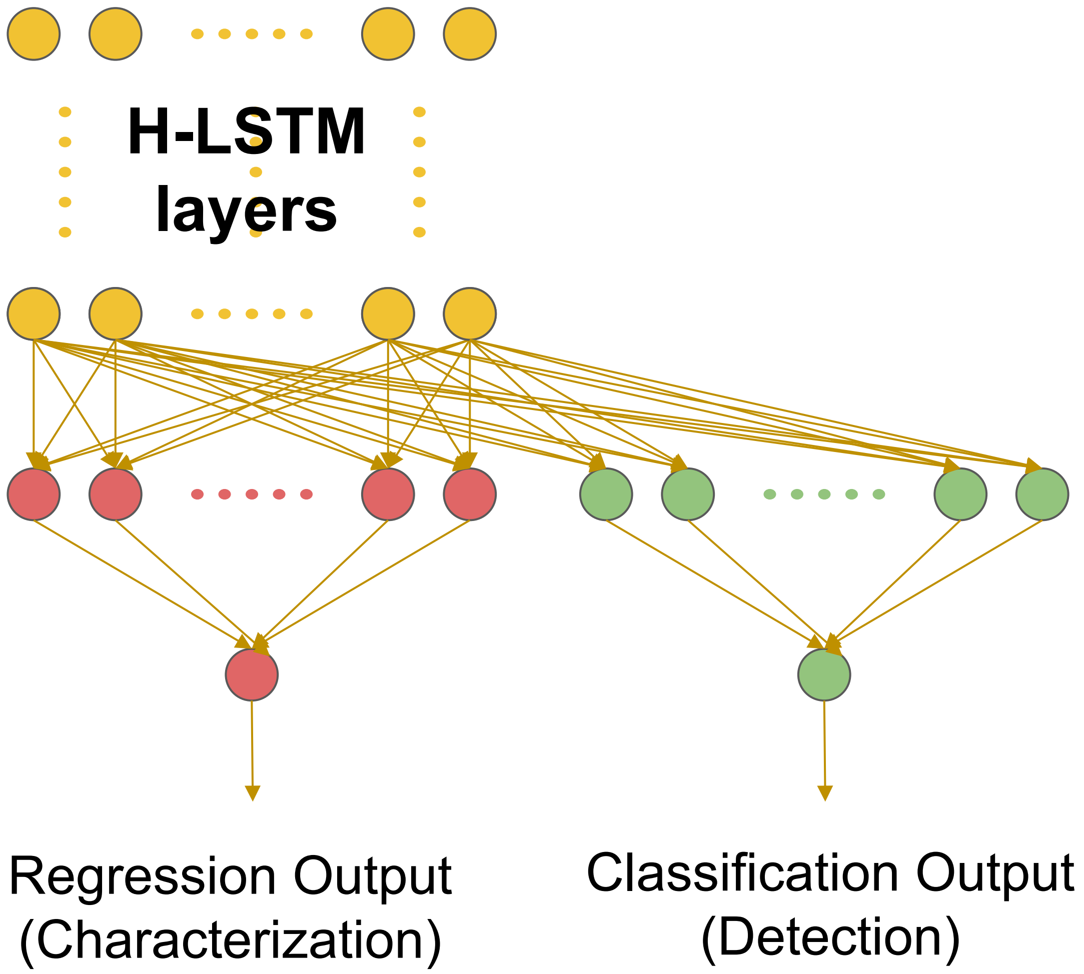


Hierarchical LSTM

- Cells are connected sequentially
- Long backpropagation through all cells loses effectiveness

- First level is cut into small groups
- Additional layers of cells are added to shorten the backprop path for each group

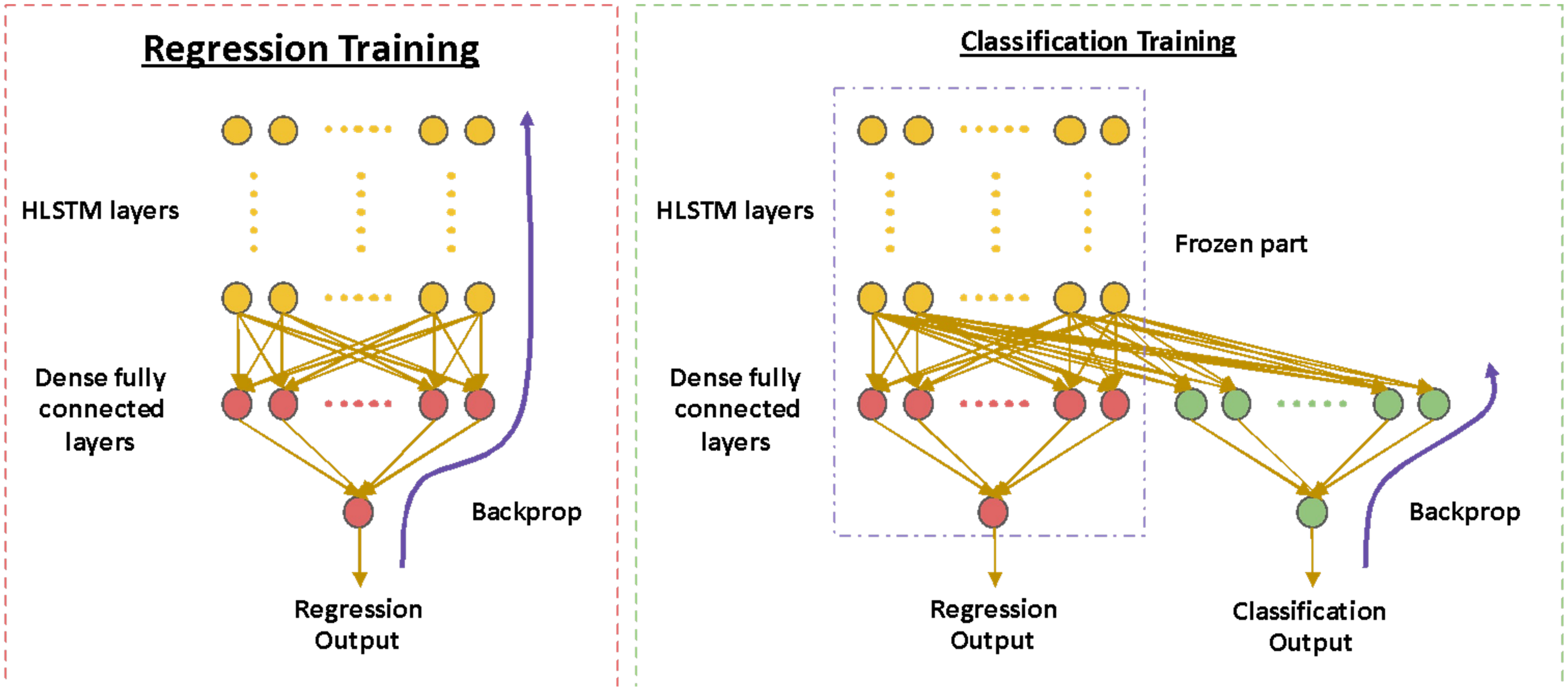
Specialized Detection and Characterization



Detecting and characterizing the TDA requires different feature processing:

- H-LSTM is used as the backbone to capture the features
- Different ‘heads’ for different functionalities
- Regression uses classification results to further improve the accuracy

Asynchronous Training



- Detection and Characterization heads are trained separately
- First train the regression head with the LSTM backbone
- Then freeze the trained part during the training of classification

Comparing Against Baseline Models

- Our model provides the minimal errors compared against traditional models
- All existing methods provide post-mortem analysis, but our method can provide real-time results
  - reduce the average reaction latency from 300s to 128s

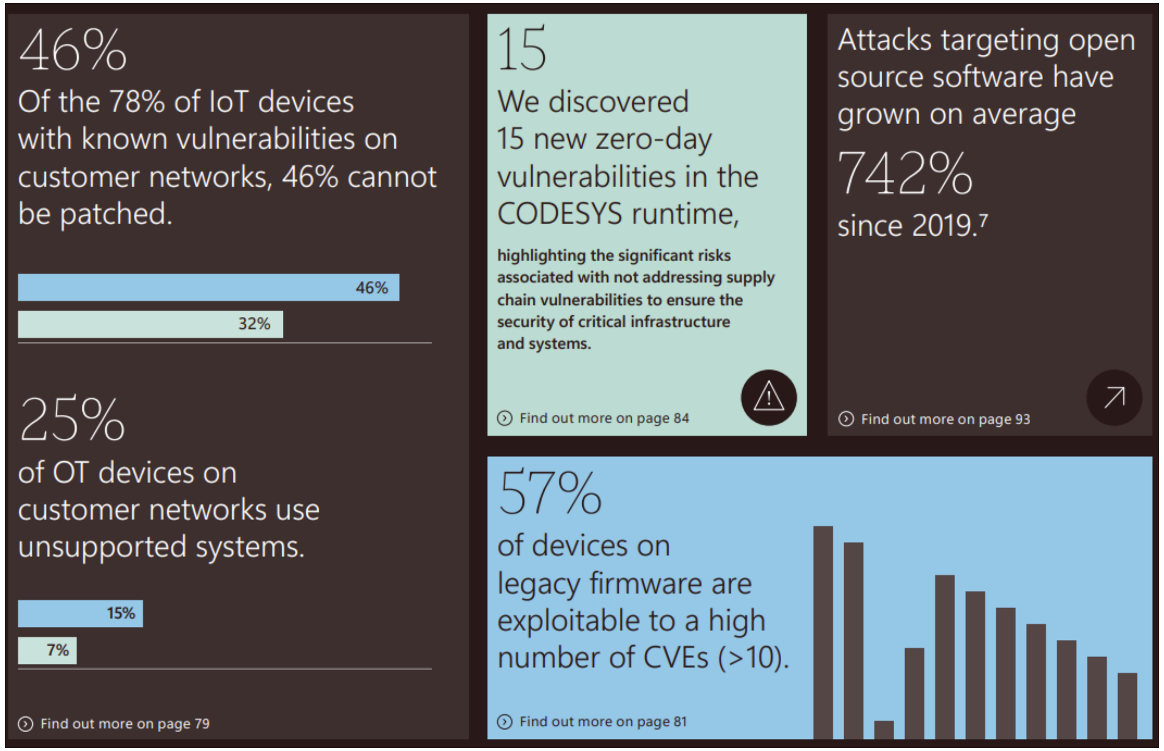
Approach	Classification (Detection)			Regression (Characterization)		
	Accuracy	FP	FN	MAE	RMSE	T_{avg}
kNN	72.6%	11.8 %	15.6 %	6.23	9.48	300
Random Forest	80.82%	5.2%	13.9 %	6.44	10.32	300
(Lou et al, 2019)	--	--	--	3.73	6.84	300
Our Model [TSG'21]	**92.39%**	**4.7%**	**2.9%**	**2.03**	**5.48**	**128**

MAE: mean absolute error
RMSE: root-mean-square error

[TSG'21] P. Ganesh, et.al., "Learning-based Simultaneous Detection and Characterization of Time Delay Attack in Cyber-Physical Systems", *IEEE Transactions on Smart Grid*, July 2021.

Threats in Cross-Domain Communication

- Cross-domain communication is important in multiple domains, e.g., Military and Defense, Healthcare, Internet of Things (IoT), etc.
- Devices in each domain can be compromised.

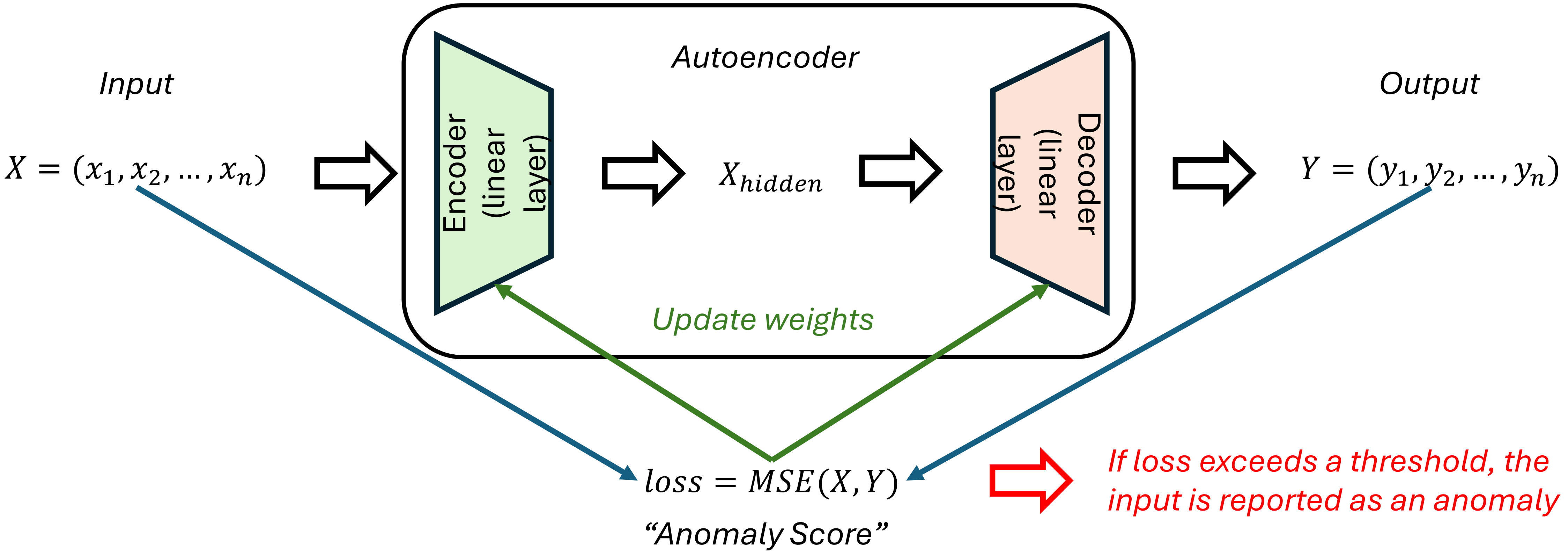


> **“Artificial Intelligence will be a critical component of successful defense. In the coming years, innovation in AI-powered cyber defense will help reverse the current rising tide of cyberattacks.”**
>
> Tom Burt, Corporate Vice President, Customer Security and Trust, Microsoft

https://www.microsoft.com/en-us/security/security-insider/microsoft-digital-defense-report-2023

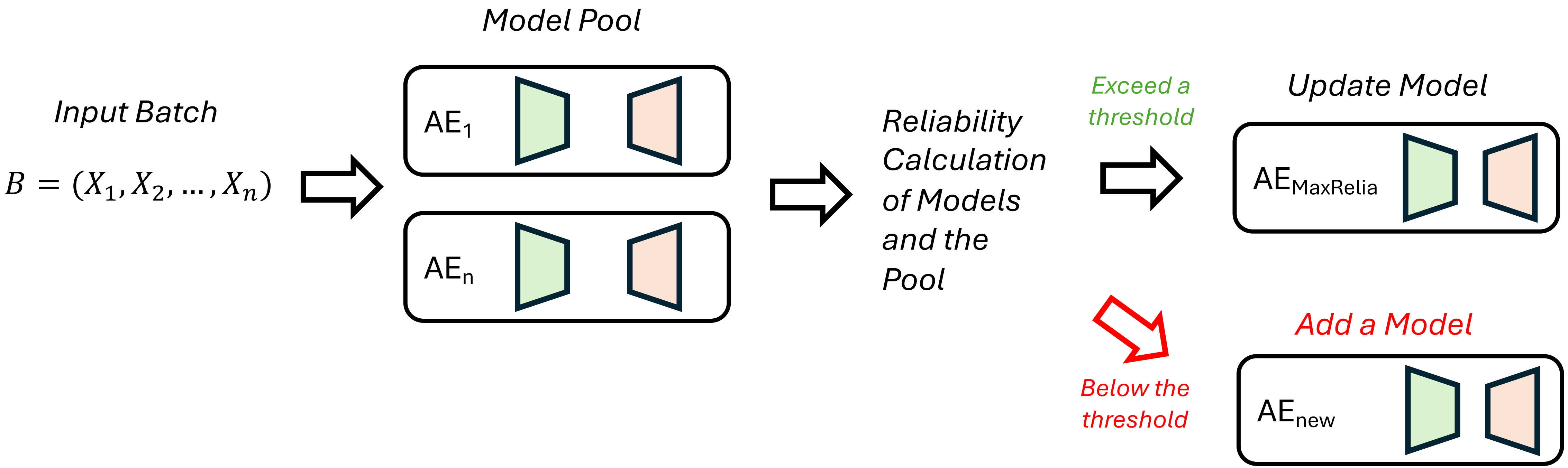
Another Scenario: Autoencoder against Cyberattacks

- Autoencoder-based AI models show strong potential for anomaly detection and work in an unsupervised fashion.



How to Deal with Evolving Data Streams?

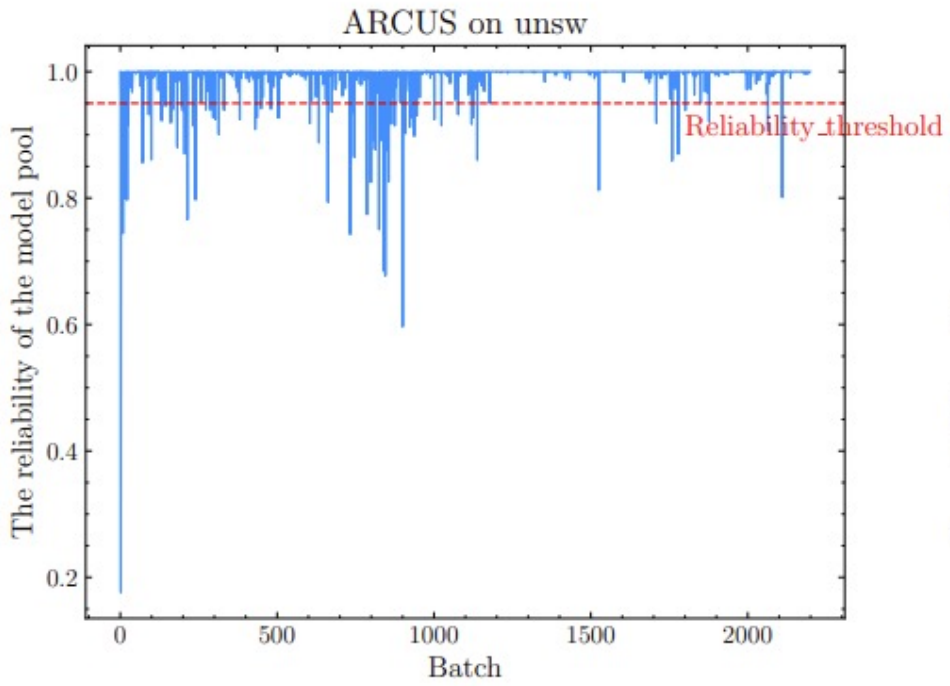
- One prior work, ARCUS[1], proposes to deploy multiple models to adapt to the evolving data



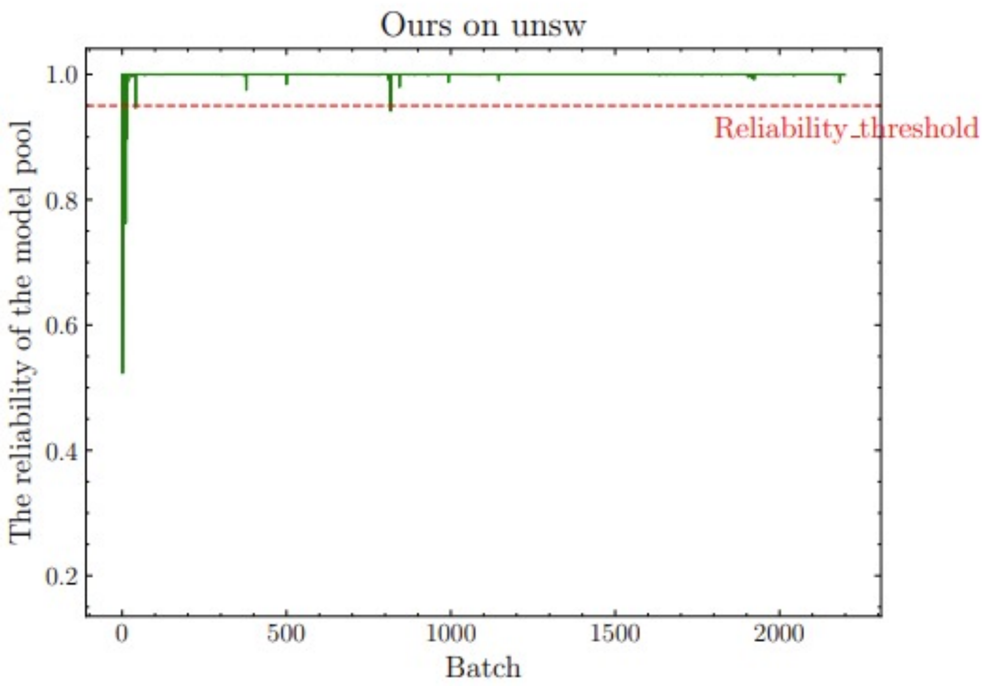
1. Yoon, Susik, et al. "Adaptive model pooling for online deep anomaly detection from a complex evolving data stream." *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022.

Raw Data Is not Stable in Evolving Data Streams

- ARCUS feeds unstable raw data into models, which leads to low reliability of the model pool.



(a) The reliability of models using ARCUS on unsw



(b) The reliability of models using ours on unsw

How to Preprocess the Data?

- Given the fact that the related data of normal objects should not change quickly at a certain period, we propose to feed the degree of change (DoC) instead of the raw data into the model.

$$DoC(x, target) = (x - target)/target$$

- Which "value" should be the target?
  - Historical data: historical data may form some clusters
  - Recent data: recent data reflects current trend

$$DoC_x = \alpha * DoC_c + (1 - \alpha) * DoC_r$$

α indicates the ratio of DoC contributed by historical data

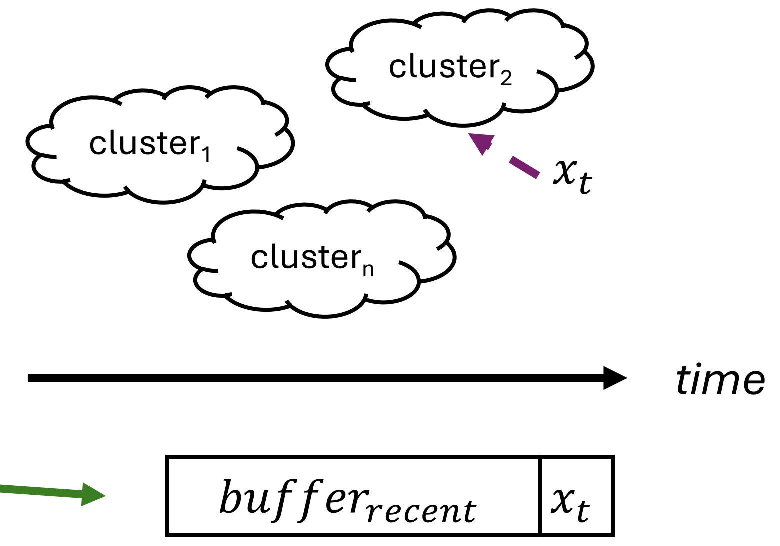
The Preprocessing Algorithm

- How to get the DoC?
- We use $\alpha, thres$ (the DoC threshold to determine whether one item can belong to a cluster), and $step$ (the number of buffered recent items).

Algorithm 1 The Proposed Data Preprocessing Algorithm

Input: The input data stream s, α, $thres$, $step$
Output: The degree of change DoC of each item x in s

```
Initialize clusters, DoC, and buffer_recent with step elements
for item x in s do
    Find the target cluster according to the degree of change threshold thres
    if the number of items in the target cluster is less than 5 then
        DoC_c equals the degree of change compared to the closest cluster
    else
        DoC_c equals the degree of change compared to the target cluster
    end if
    Update buffer_recent in a FIFO manner
    Calculate DoC_r targeting the mean of buffer_recent
    Calculate DoC_x using α, DoC_c, and DoC_r
    Append DoC_x to DoC
end for
Return DoC
```



The Accuracy of Anomaly Detection

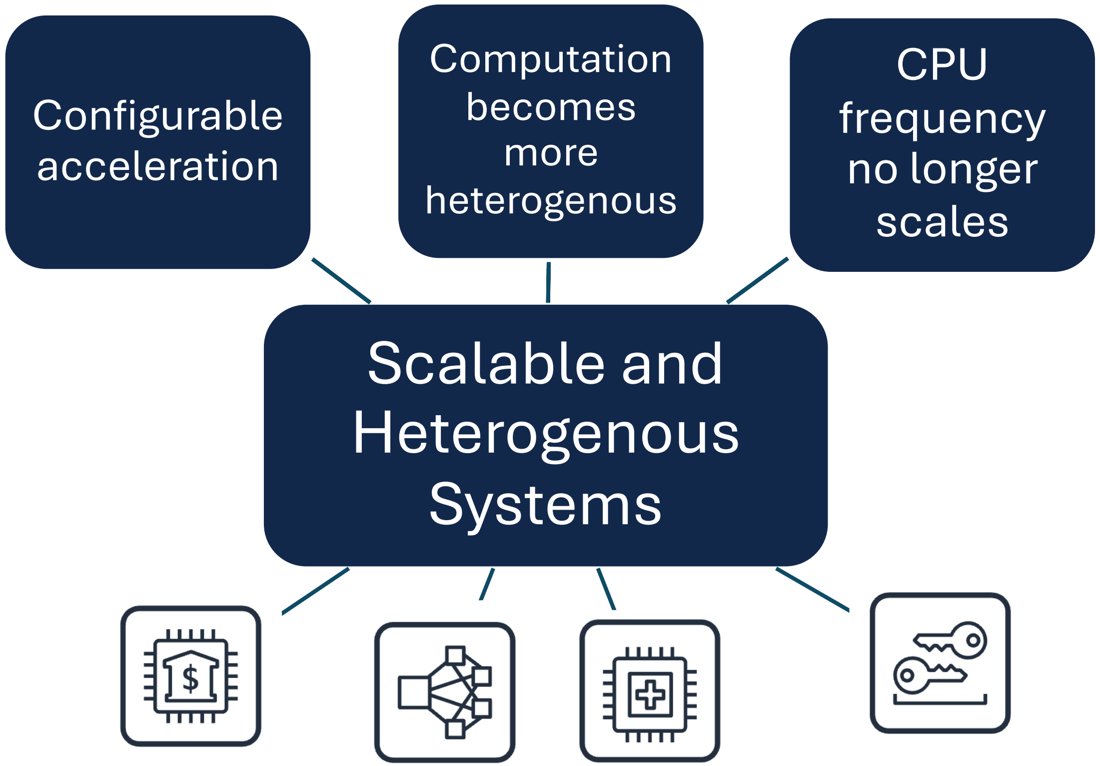
- We have increased around 0.11 AUC (i.e., the accuracy) in average compared to ARCUS. Our proposed method can achieve the top-2 accuracy on all the benchmarks compared to related works.

Dataset	Ours	ARCUS	sLSTM-ED	sREBM	STARE	RRCF	MiLOF	DILOF	MStream
INSECTS-Abr	**0.753**	0.631	**0.749**	0.471	0.555	0.695	0.393	0.730	0.709
INSECTS-Inc	**0.706**	0.600	0.696	0.383	0.559	0.669	0.415	**0.757**	0.593
INSECTS-IncGrd	**0.753**	0.641	**0.795**	0.575	0.594	0.719	0.395	0.746	0.628
INSECTS-IncRec	**0.743**	0.634	0.709	0.491	0.551	0.680	0.381	**0.743**	0.637
GAS	**0.88**	**0.878**	0.408	0.506	0.635	0.804	0.589	0.470	0.480
RIALTO	**0.875**	**0.784**	0.617	0.492	0.532	0.731	0.456	0.742	0.699
unsw	**0.579**	0.466	NA	NA	NA	NA	NA	NA	NA

Secured AI for Security

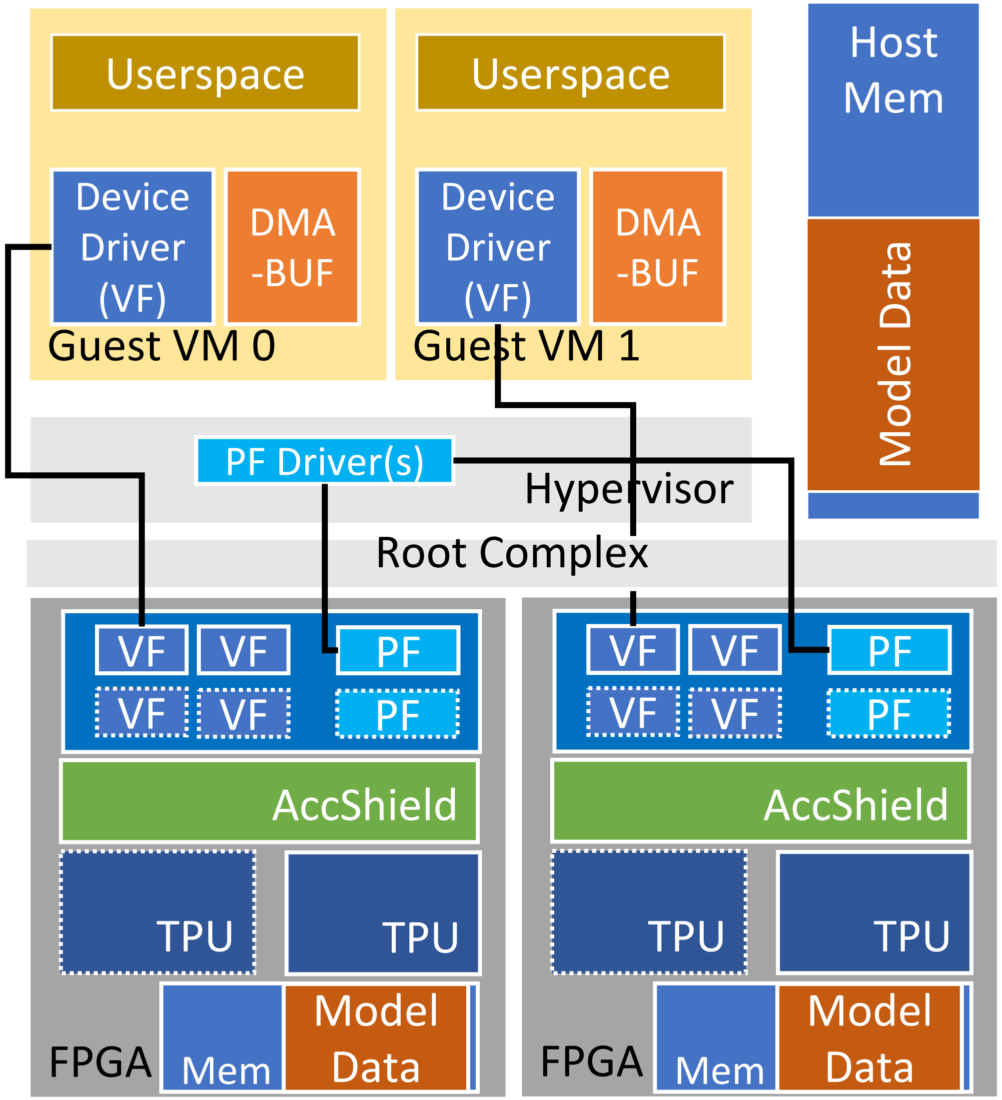
Multi-Dimensional Objectives

- Scalability
- Heterogeneity
- Efficiency
- **Security**



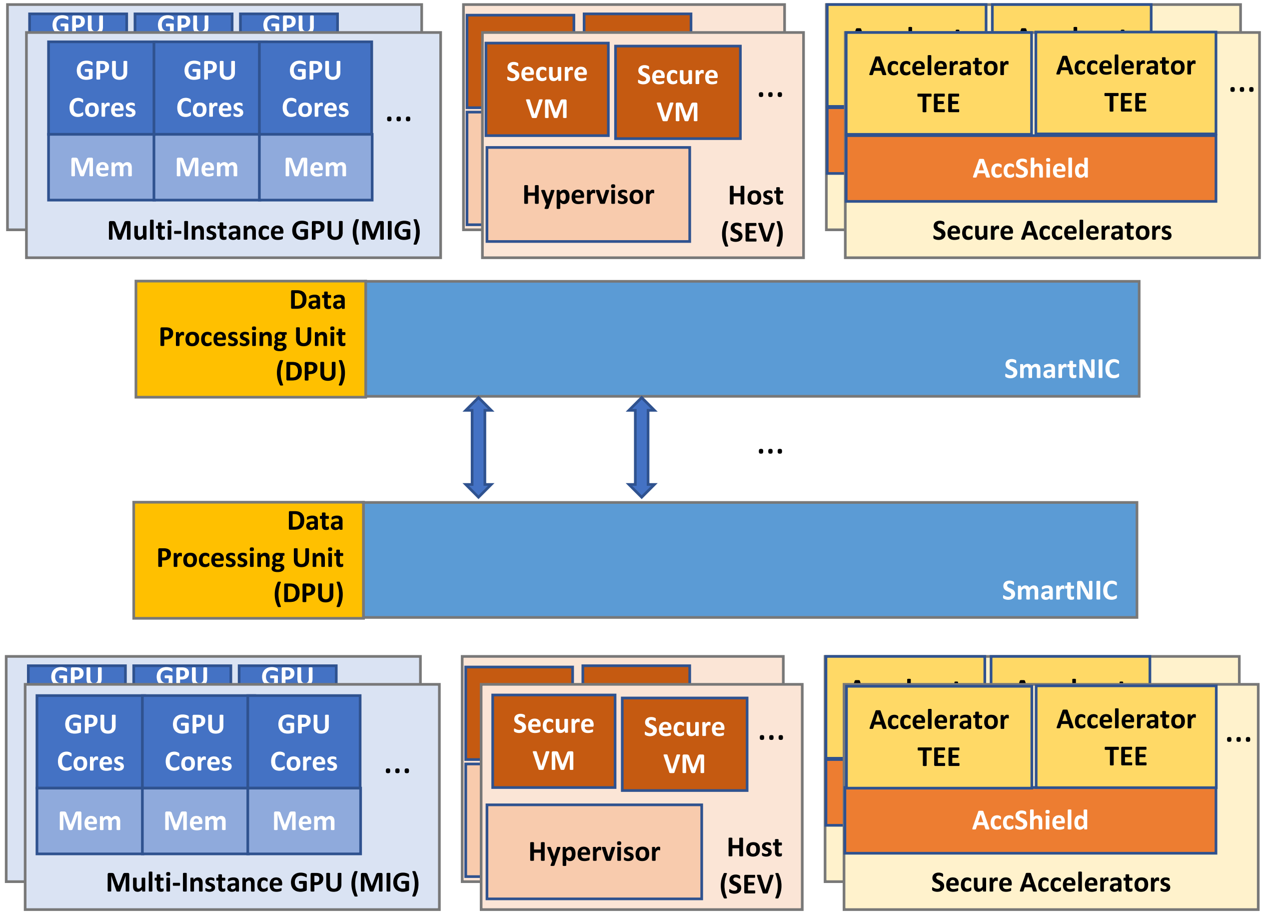
Efficient and Secure Data and Model Sharing

- Intra-user TEEs
  - Multiple TEEs belong to the same user
- Inter-user TEEs
  - TEEs of different users
- Energy monitoring (e.g., IBM Kepler)
- Applications:
  - Large Foundation Model (e.g., LLMs) shared by multiple downstream tasks
  - Workload traffic dynamically changing and evolving



Scalability, Heterogeneity, Efficiency, Security

- Distributed AI workloads with many accelerator TEEs collaborate across nodes or even network
- Sustainability throughout the system stacks as an important dimension of global optimization
- New efficient & secure programming models, libraries, and tools for AI applications
- AI-driven resource management and control and AI-assisted attack detection and containment
- End-to-end protection



Conclusions

- **Security for AI** is essential
  - Accelerator TEE is an important building block
  - System Integration and Trade-offs
  - Adaptability - Flexible and Multi-tenant Accelerator design
  - Challenges: to be consistent and scalable for various AI accelerators
- **AI for Security** is emerging
  - On-line real-time attack & anomaly detection and responses
  - Smart resource management and control methods
  - Challenges: accuracy, robustness, predictability, interpretability
- **The Two Working Together** is exciting
  - Scalability + Heterogeneity + Efficiency + Security
  - Distributed TEE environment and end-to-end protection for dynamically changing AI workloads

Acknowledgement

Collaborators:

IBM: Sandhya Koteshwara, Mengmei Ye, Hubertus Franke

UIUC: Wei Ren, Junhao Pan, William Kozlowski

ARCS: Prakhar Ganesh, Xin Lou, Yao Chen, David K.Y. Yau, Rui Tan, Marianne Winslett, Shiqing Li

Sponsors:

The Grainger College of Engineering
IBM-Illinois Discovery Accelerator Institute

C3SR center for cognitive computing systems research

ILLINOIS ARCS ADVANCED RESEARCH CENTER AT SINGAPORE

AMD XILINX

Questions