From Confidential Computing to Zero Trust Come Along for the (Bumpy?) Ride

Mengmei Ye (mye@ibm.com)
IBM Research

Acknowledgements: Sandhya Koteshwara, Derren Dunn, Hubertus Franke, Chris Porter, Tobin Feldman-Fitzthum, Angelo Ruocco, Daniele Buono, Claudio Carvalho, Apoorve Mohan, Mudhakar Srivatsa, Zhuoran Liu, Nelson Gonzalez

IBM

Disclaimer

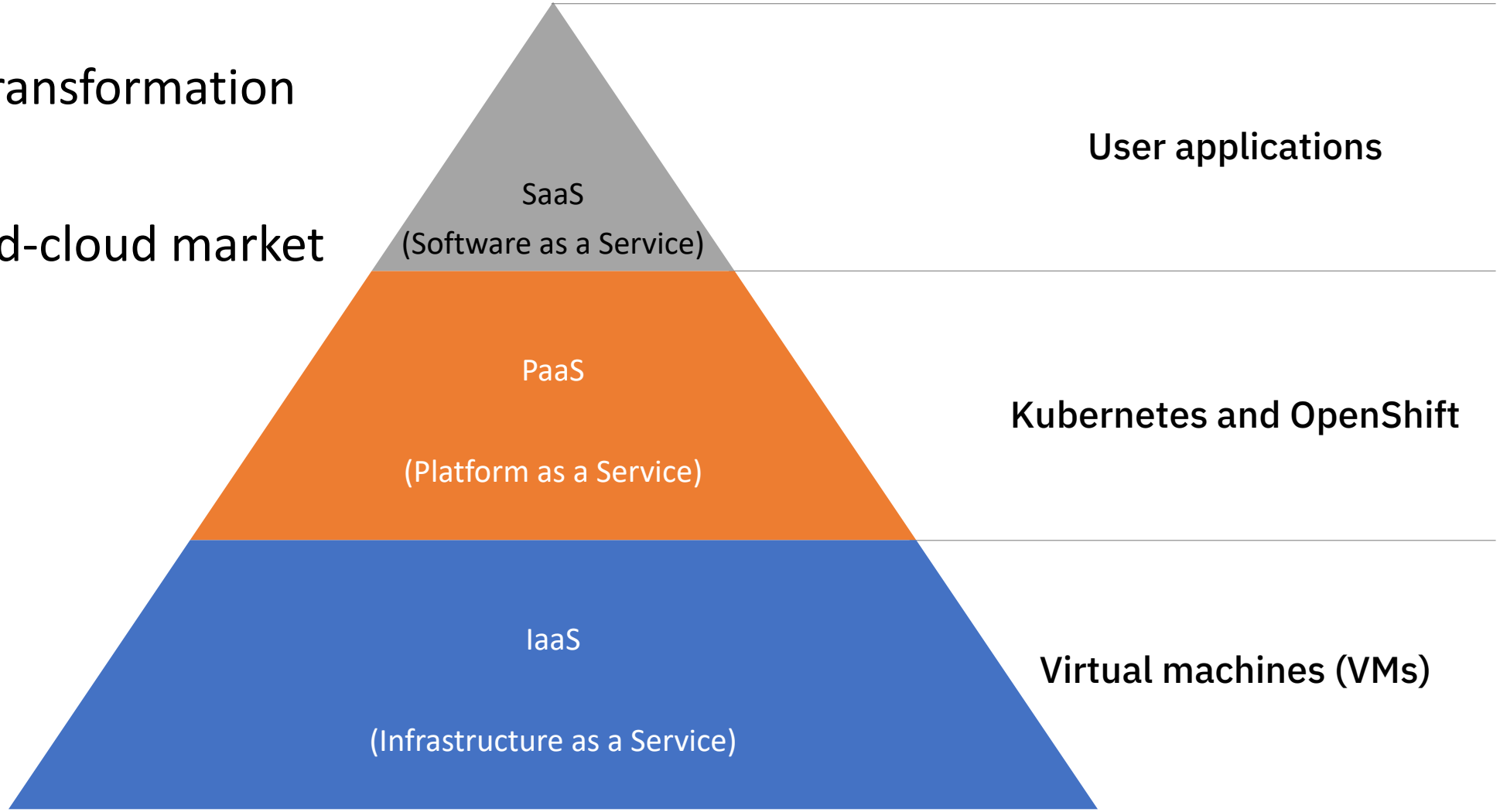
- This work represents the view of the authors and does not necessarily represent the view of IBM.
- The features described may not ultimately exist or take the described form in a product.
- IBM is a registered trademark of International Business Machines Corporation in the United States and/or other countries. Other company, product, and service names may be trademarks or service marks of others.

Outline

- What is Confidential Computing
- **Secure and High-Performance Implementation/Evaluation Using Confidential Computing Technology**:
  - **CPU**: Electronic Design Automation (EDA) workload
  - **CPU-GPU**: Microbenchmarks and Large Language Models (LLMs)
- From Confidential Computing to Zero Trust

Today’s Cloud Infrastructure

- Rapid digital transformation
- Growing hybrid-cloud market



Confidential Computing (CC)

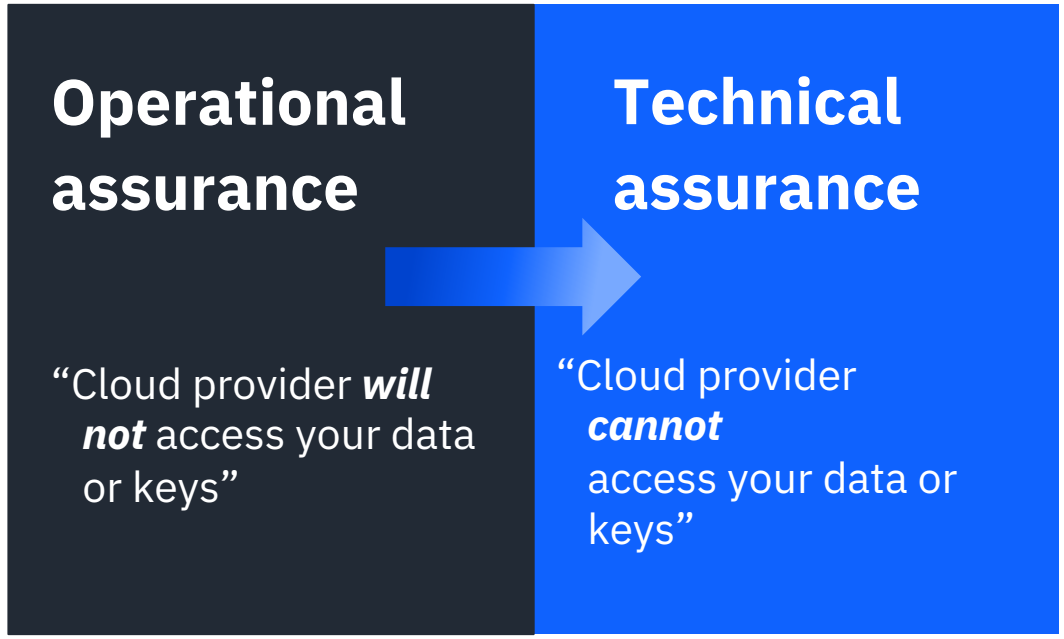
"Data is King" [1]:

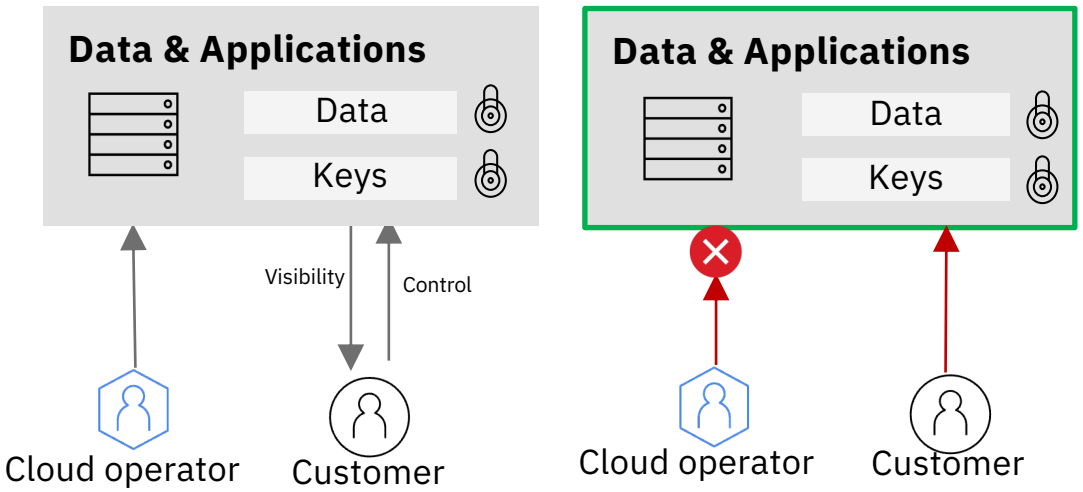
- High Value
- Sensitive Data must be protected
- Regulated Industry (e.g. Finance, Health)

- Lack of trust impedes move to public shared infrastructure (cloud)
- Confidentiality and Integrity → Confidential Computing

CC can be successful only if workload deployment is transparent and overhead is low

- CC can protect Intellectual Property and Sensitive Data
  - AI Models and Weights
  - Finance and Healthcare
  - EDA Chip Design
  - Data Sovereignty and Security Compliance
- Attacks on infrastructure have successfully stolen entire DNN models [2]





[1] https://www.solutions.kompass.com/blog/the-business-worlds-digital-transformation-why-data-is-king/
[2] https://www.usenix.org/system/files/sec21-zhu.pdf

Confidential Computing (CC) – Technical Assurance

- **Data at rest**
  - Secure storage with file- or block- level encryption
- **Data in motion**
  - Virtual Private Networking (VPN): secure communications through encrypted VPN tunnels
- **Data in use/compute** (*confidential computing*)
  - Memory is encrypted
  - Process based trusted execution environment (TEE): Intel SGX
    - ✗ Need to modify applications deployed in the systems
  - VM based TEE:
    AMD SEV, Intel TDX, IBM Z Secure Execution, etc.
    - ✓ Larger TCB but no need to changes to applications
- **Security requirements**
  - Data should *only* be decrypted inside of enclaves – i.e. encrypted VMs
  - CSPs, hypervisors should have no code running inside of enclaves

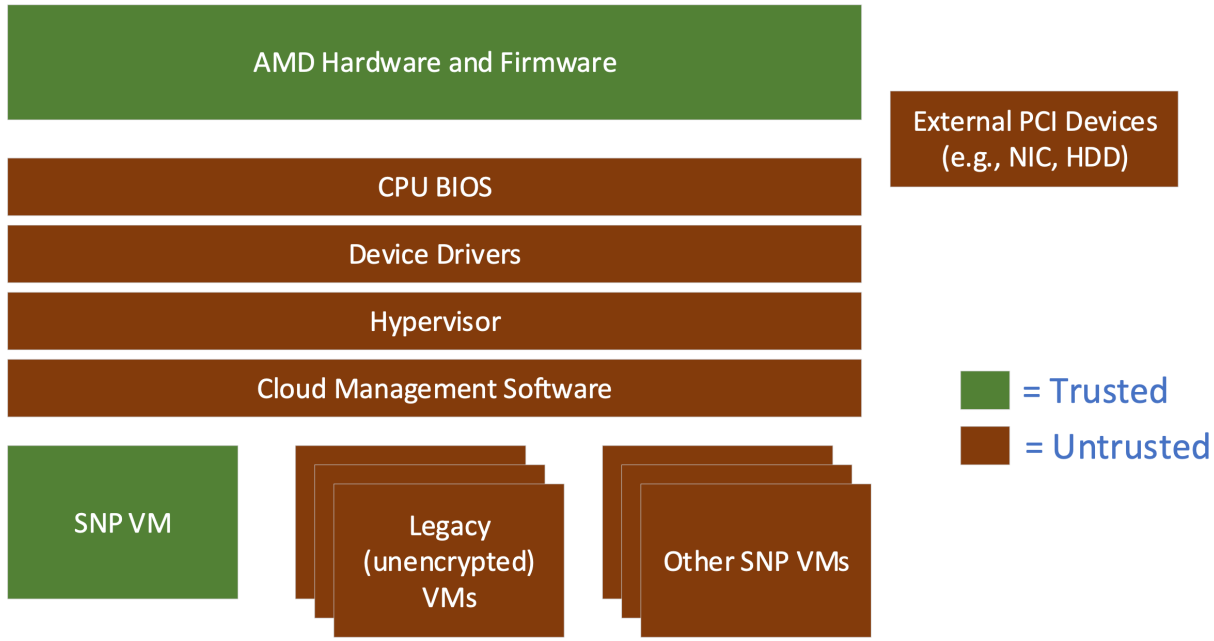
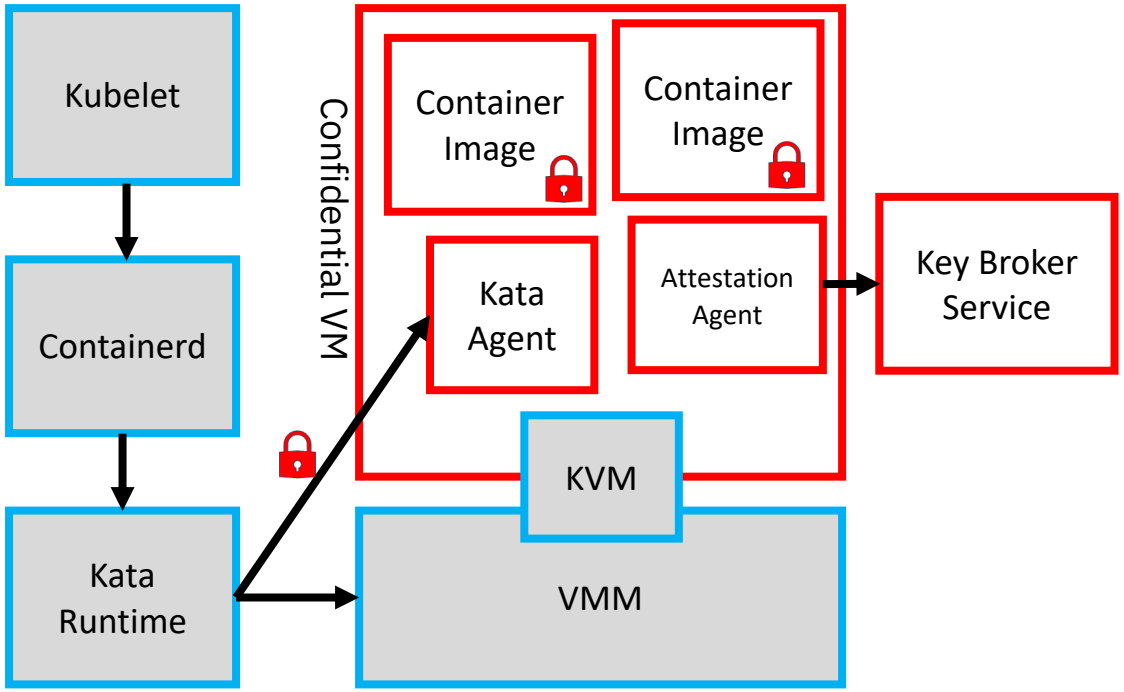


Image source – AMD SEV-SNP: Strengthening VM Isolation with Integrity Protection and More, January 2020, White Paper.

Confidential Containers (CoCo): Cloud-Native Confidential Computing Technology

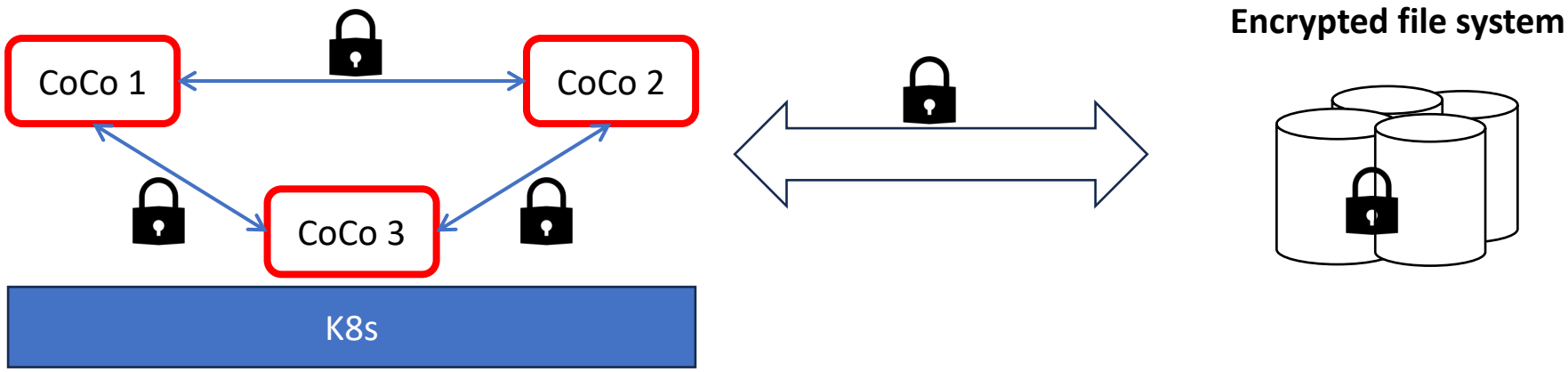
- Cloud Native Computing Foundation Sandbox project based on Kata Containers: *github.com/confidential-containers*
- *IBM community leadership*, IBM has contributed over 100 PR's and over 1000 Github contributions
- *Steering committee*: IBM (Z and Research), Red Hat, Microsoft, Alibaba, AMD, Intel, ARM, NVIDIA, Rivos
- Each pod is encapsulated in a confidential/encrypted VM
- Each container image must be pulled and decrypted within the enclave since the host must not get inside of the image
- Small interface (Kata Agent) sets up the pod inside the VM



CONFIDENTIAL CONTAINERS

Challenges of Running Real-World Applications in Cloud with Confidential Computing Technology

- **Security challenge**
  - Standard security solutions for storage and networks require host involvement, but **the host is not trusted in the CoCo framework**
  - Any sensitive data can *only* be decrypted inside of CoCo
- **Performance challenge**
  - What is the scalability of these security solutions? What is the performance overhead with end-to-end CoCo framework (i.e., CoCo + Secure Storage + Secure Network)?
- **Automation challenge**
  - Can we deploy real-world applications into end-to-end CoCo framework without modifications to the applications?



EDA Typical Design Flow

- Large storage requirement
  - PDKs and cell libraries (>1TB): read only
  - Tool installation (~500GB or more): read only
  - User workspace (>1TB per project): read and write
    - Different nodes read and write in the same file system
- Single established design flow is applied to multiple macros and units
- Characteristics of the workloads:
  - HPC-like requirements
    - 1,000's to 10,000's CPUs
    - High speed network and shared storage
  - **PDKs and IPs are high-value assets**

The cost of a leaked proprietary design is measured in millions of dollars, loss of competitiveness and brand damage

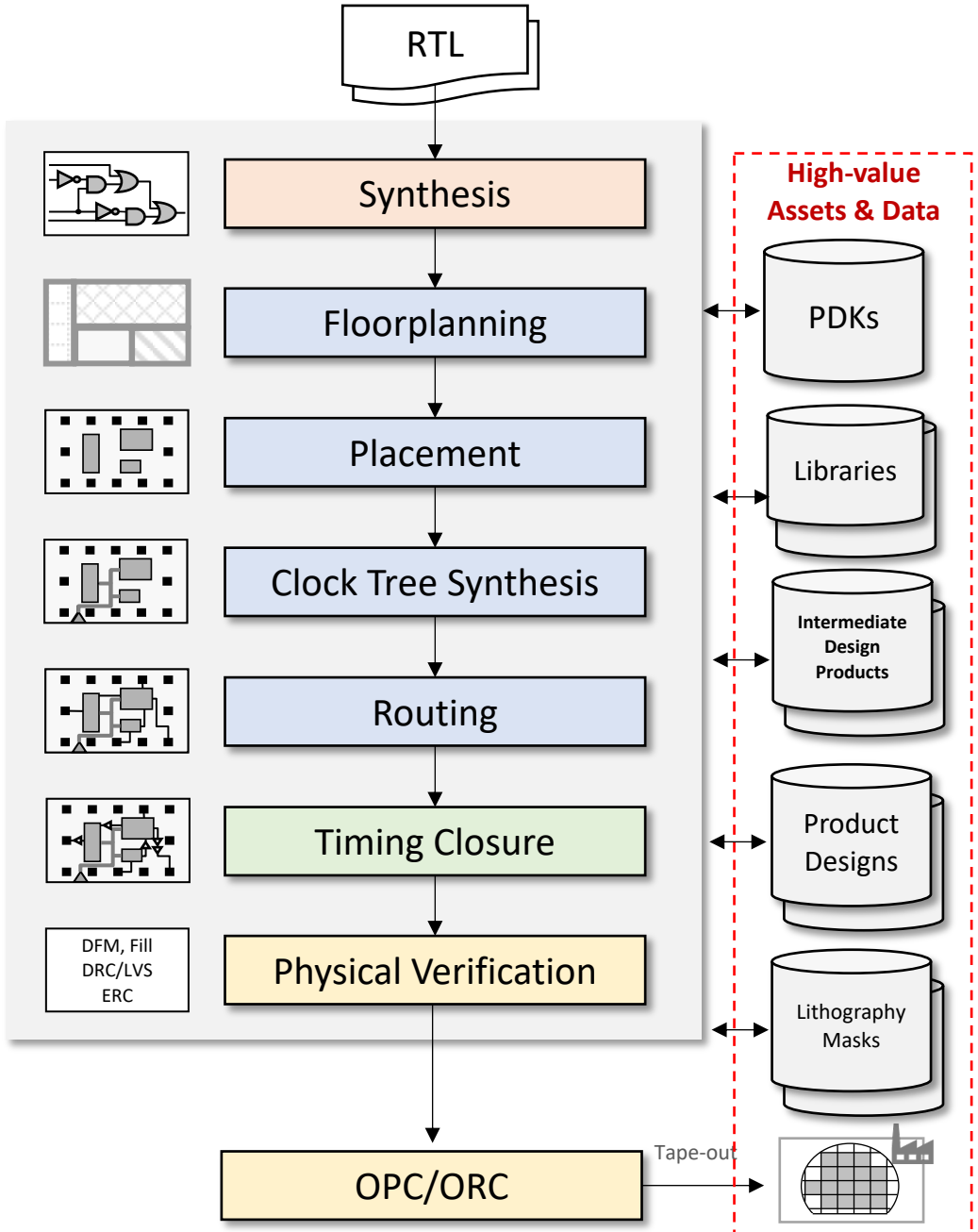
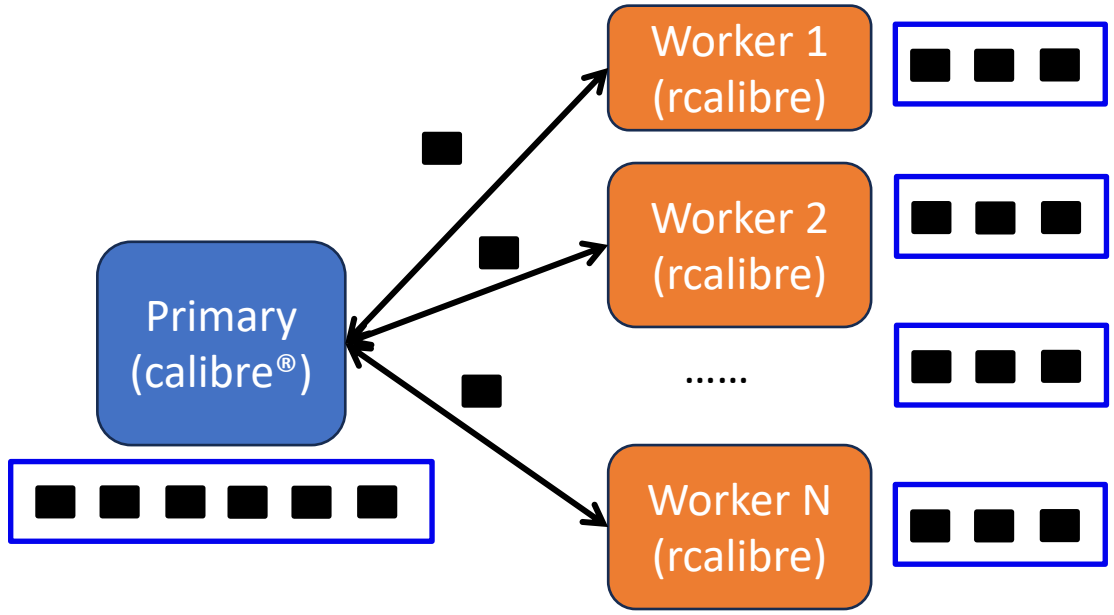


*Figure adapted from: Kahng et al., VLSI Physical Design, Springer, 2011

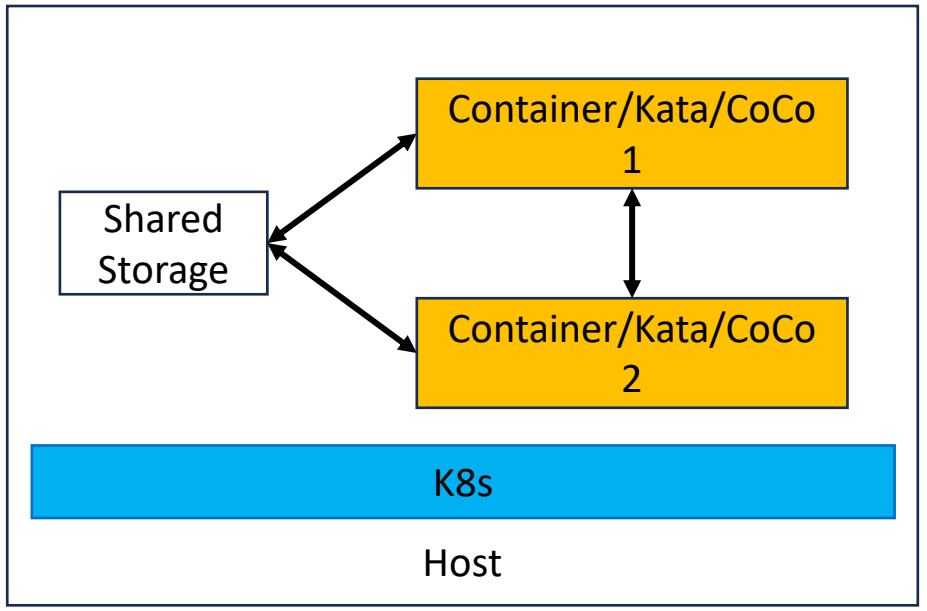
EDA Use Case: Siemens Calibre®

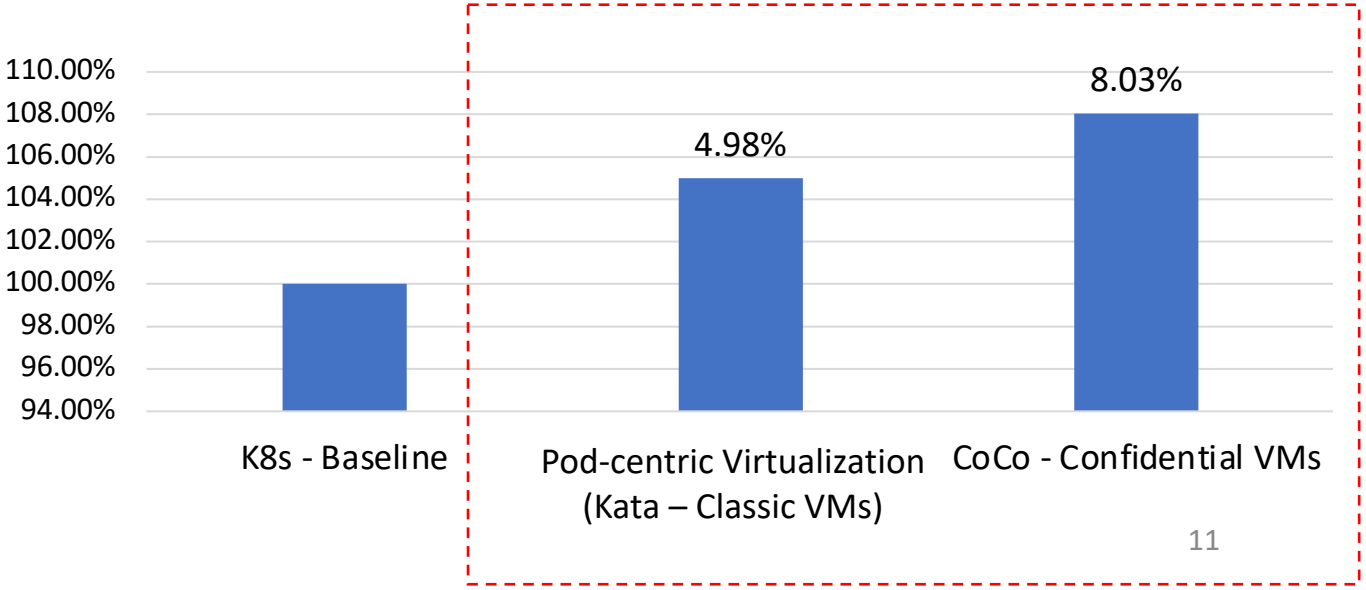
- “Classic” Primary/Worker HPC pattern
- Parallel distributed memory application
- OPC distributed runs:
  - Primary pod/process divides layout into tiles
  - Parses tiles out to processes on remote (worker) pods/processes
  - Completed tiles written to common filesystem
  - Tiles distributed until layout complete
  - Primary pod/process reads from common file system and assembles final output



EDA with CoCo

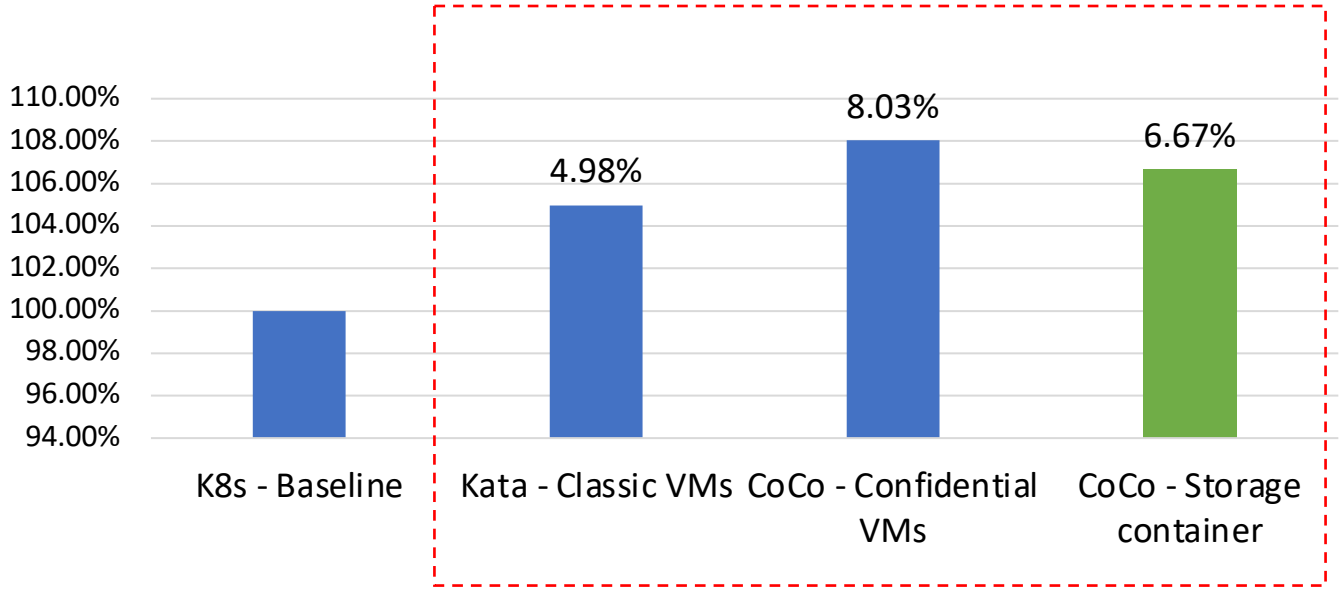
- K8s setup
  - 1 control plane + 28 worker nodes (1736 cores in total)
  - 2 pods per node (56 pods in total)
  - Each worker node supports AMD SEV-based CoCo
- Siemens Calibre® workloads are running in CoCo
  - Memory is encrypted
  - Unencrypted data is only inside of confidential VMs
- Kubernetes Control Plane runs on the host and is not trusted
- Experimental baseline: a k8s cluster running on bare-metal machines

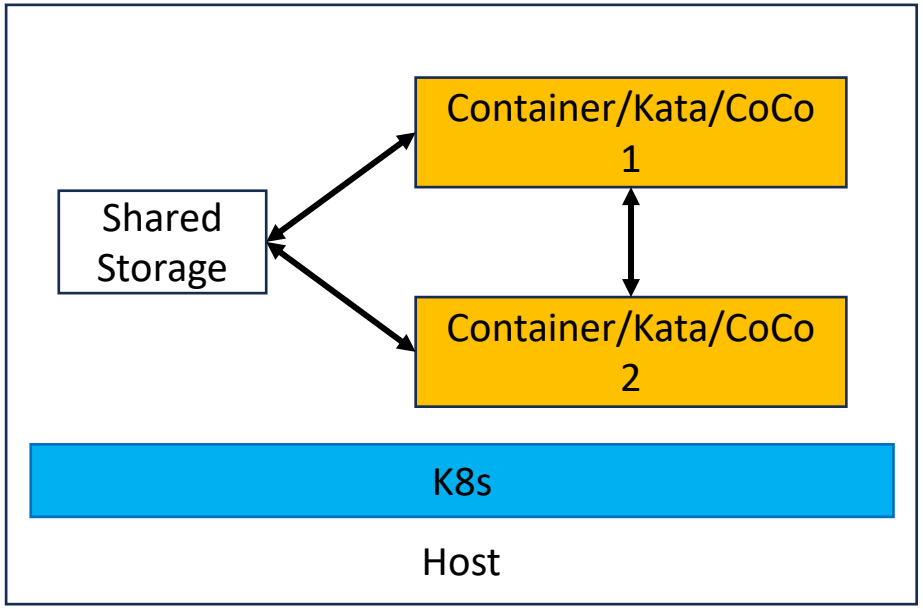


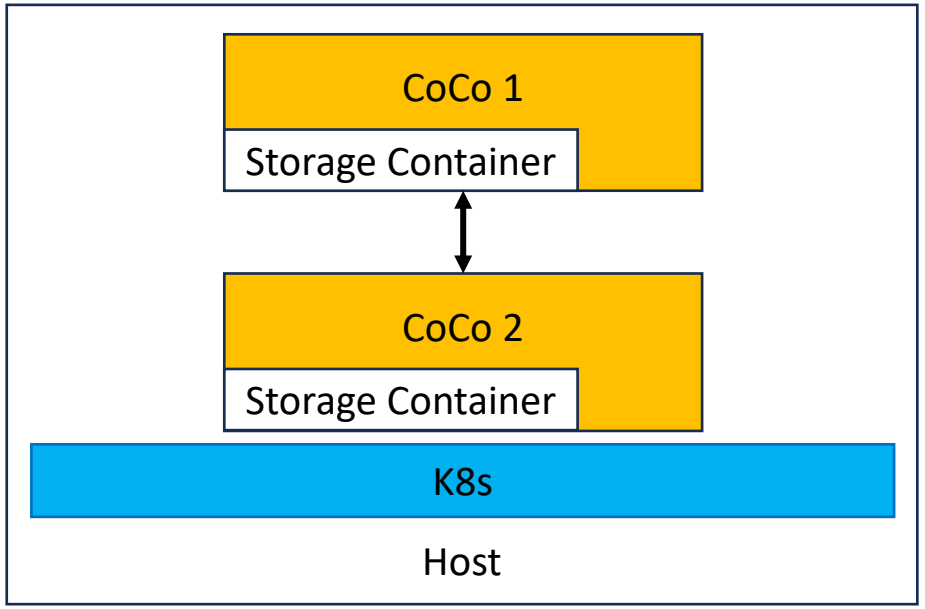


EDA with CoCo

- Performance improvement by mounting shared storage into CoCo as a sidecar
  - From 8.03% to ***6.67%***
  - *Eliminate Host<->VM disk overhead*







EDA with CoCo + Secure Storage

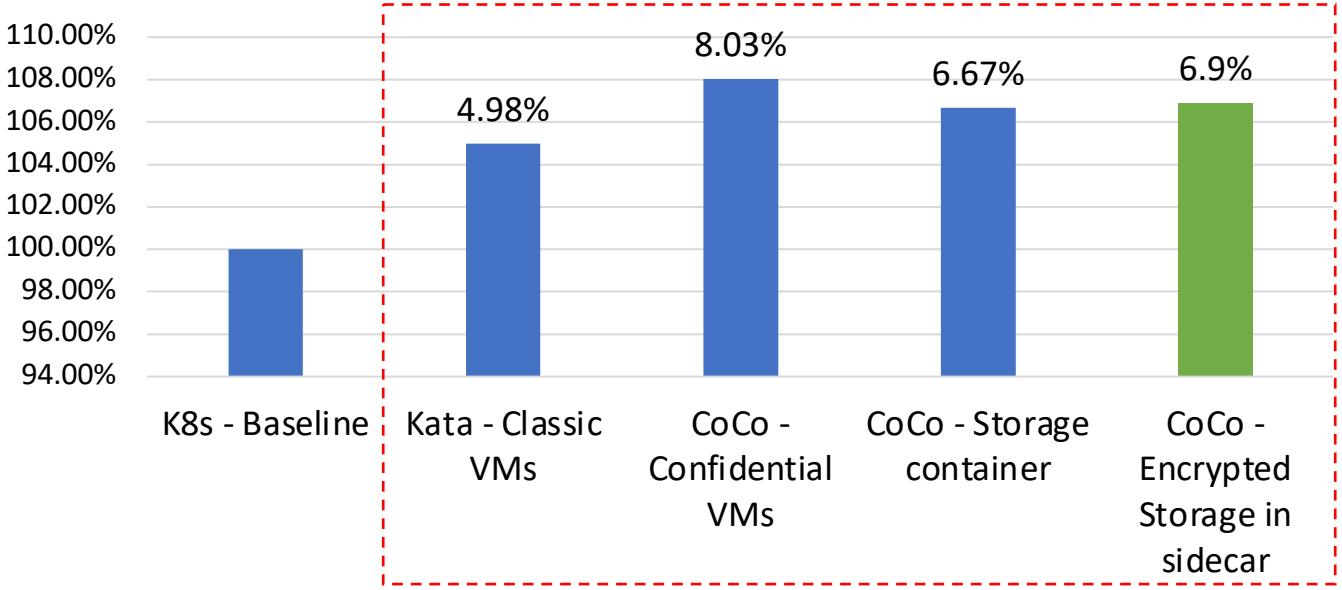
- Secure storage: encrypt PDK data by persona, derivative design data, etc.
- Based on GPFS, transparent encryption layer for NFS and other FS, etc.
  - Generic Sidecar for Encrypted File System layer (gocryptfs, fuse-based)
  - Bring-Your-Own-Sidecar for customized, non open-source solutions (e.g., GPFS)
- CoCo + NFS + Gocryptfs (nuetzlich.net/gocryptfs)
  - ***6.9%*** performance overhead in total

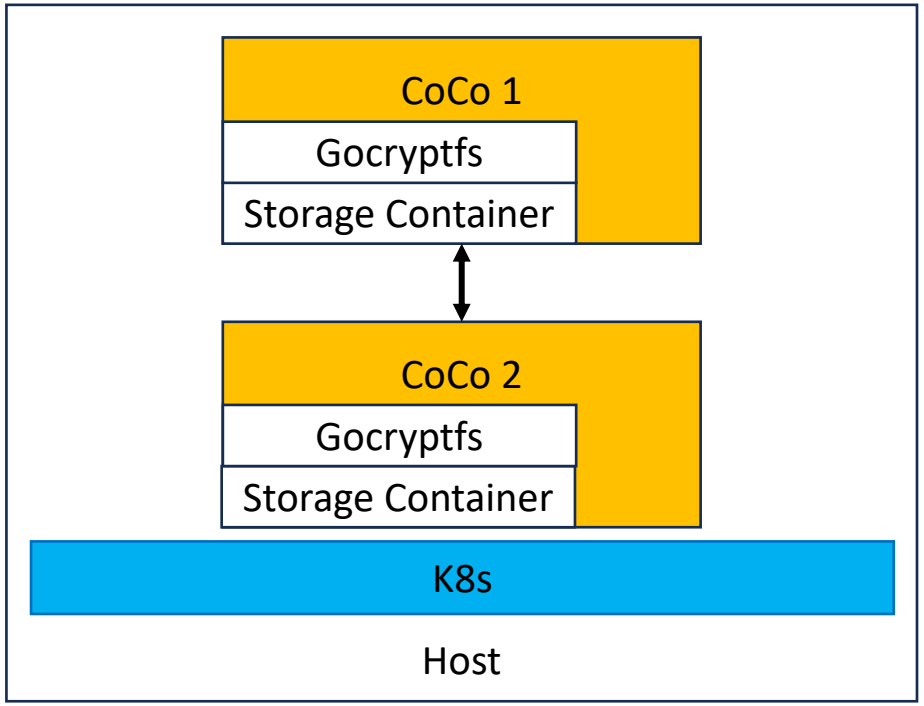
Host sees ciphertext:

```
# ls /tools/
D5khkmKfF2CKwBMmWTD_yg
```

CoCo sees plaintext:

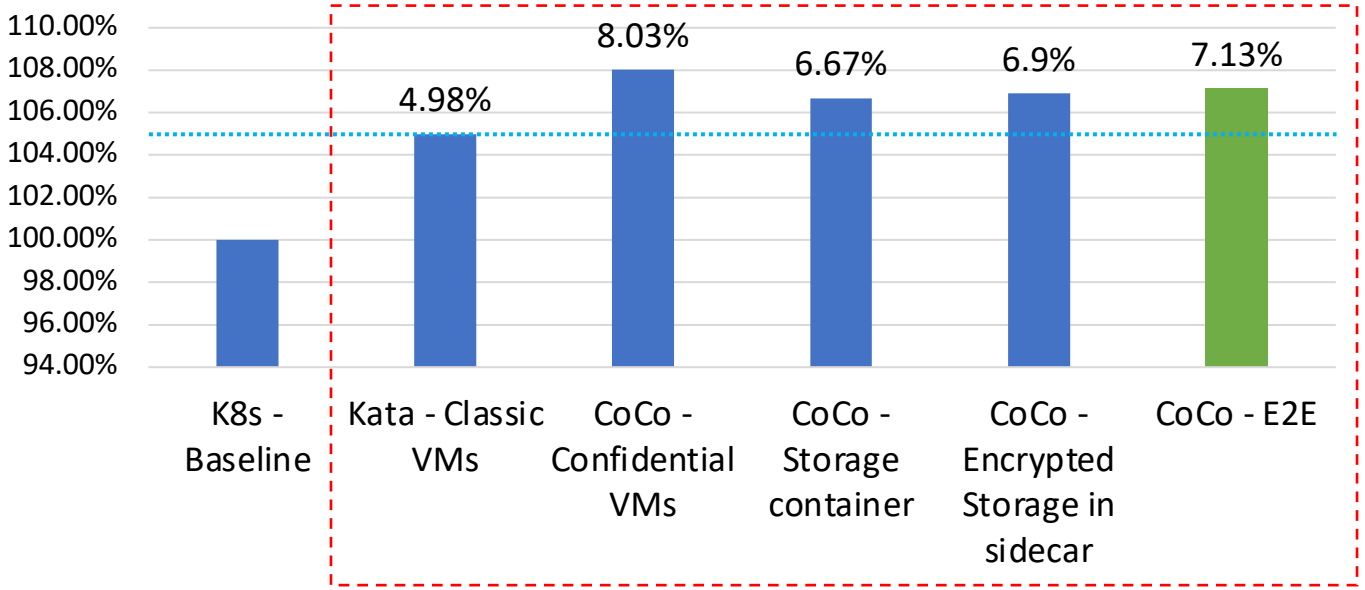
```
[root@cal-1696493001-d7pwb /]# ls /tools
calibre
```

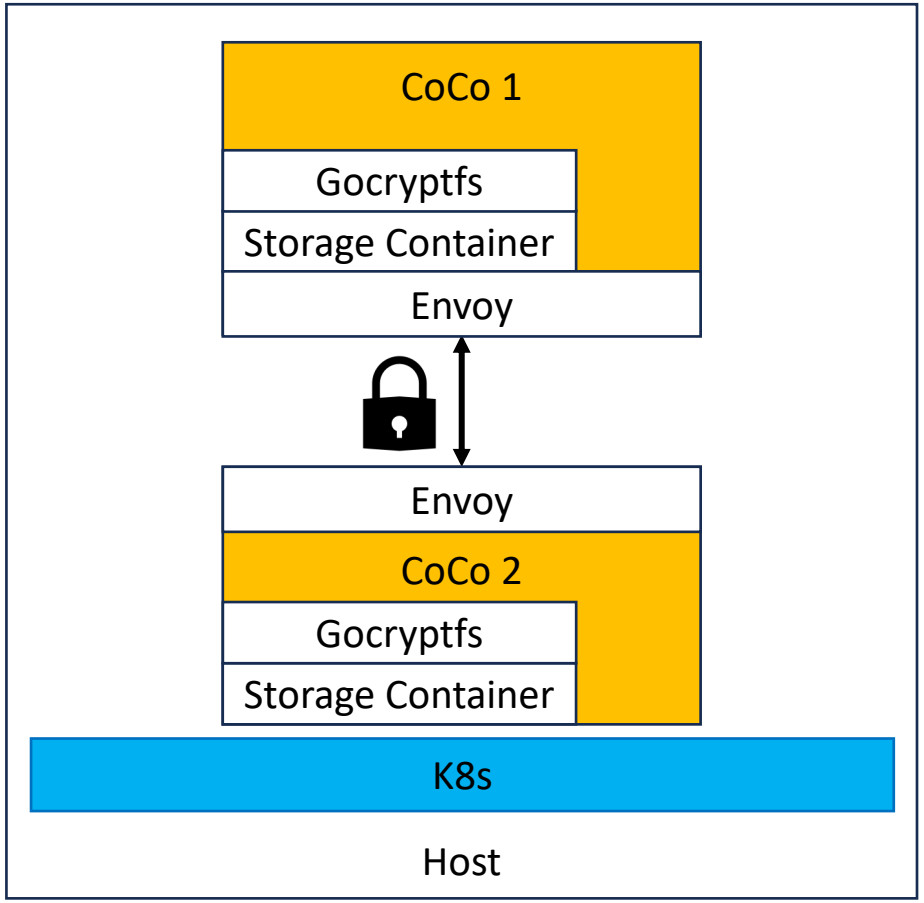




EDA with CoCo + Secure Storage + Secure Network

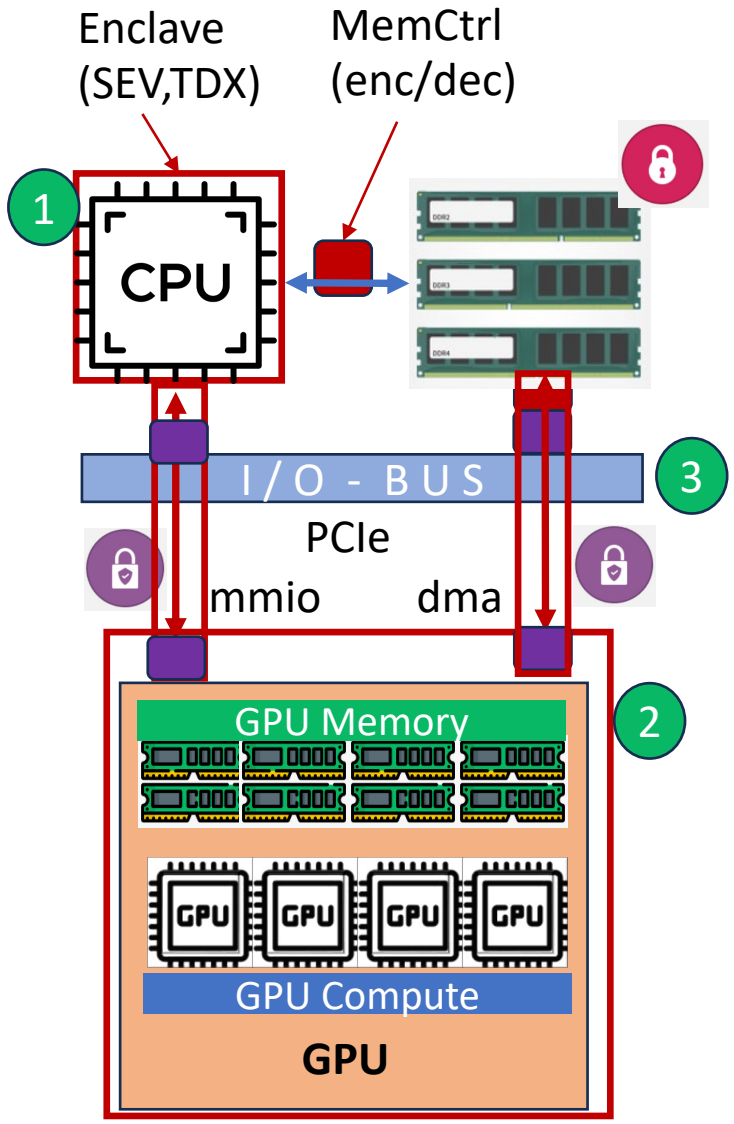
- Any data leaving CoCo must be encrypted
- Generic Sidecar for mTLS encrypted network traffic through ISTIO's Envoy
- CoCo + NFS + Gocryptfs + Envoy (i.e., CoCo - E2E)
  - ***7.13%*** performance overhead in total





TEE Accelerator Integration

- All parts are required to protect the data confidentiality & integrity in their way, e.g.
  - 1 CPU (VM based TEE, cache tagging, memory encryption)
  - 2 GPU (H100 physical partitioning, but no memory encryption)
- All communication among components must be protected & encrypted by hardware or software 3
  - H100 device driver uses software encryption with bounce buffer due to lack of full host support, e.g. DMA must be conducted transparently in the context of owning VM.
- Overhead with data communication path potentially significant
  - Depends on frequency of, enc/dec bandwidth -> impact of increased latency,
  - Ability to hide data communication with computation



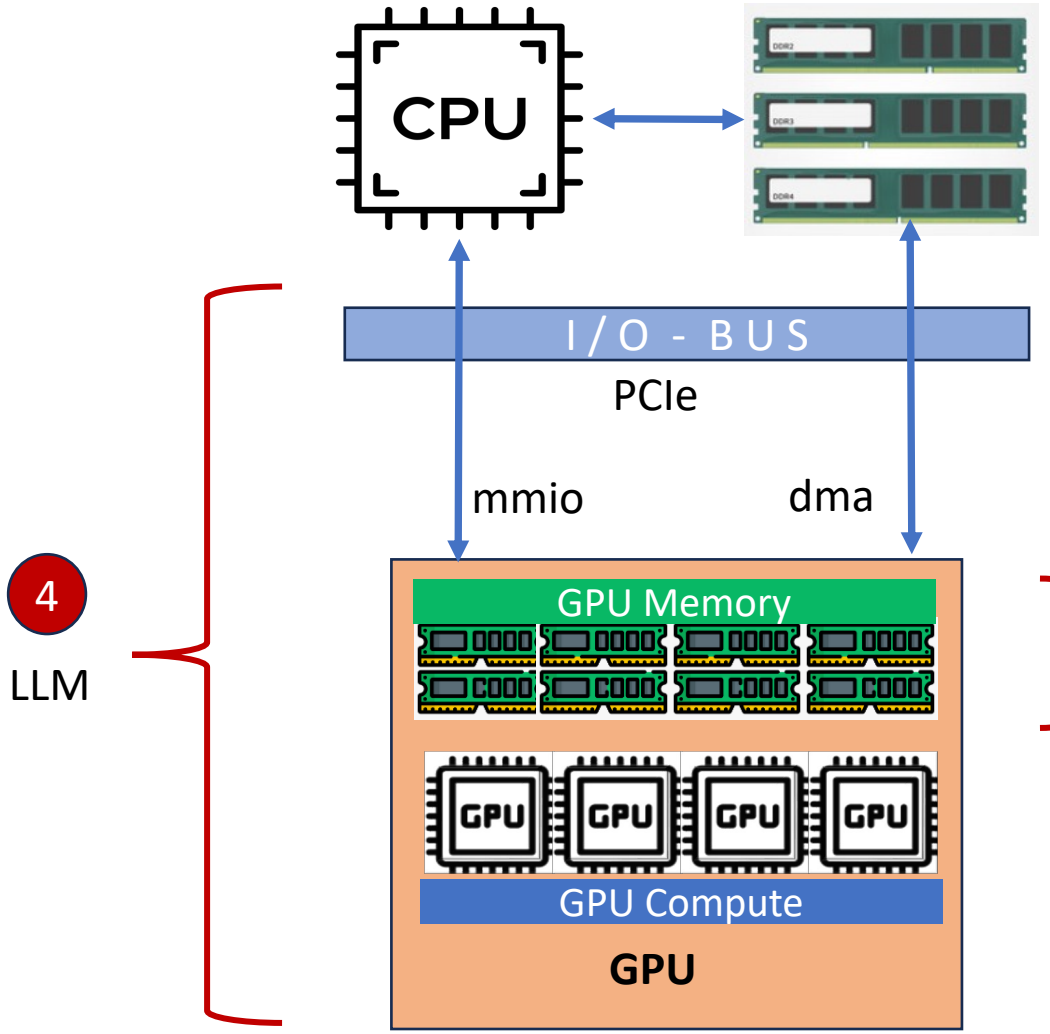
New technology to protect I/O:

SPDM: link encryption

TDIO: dma access with mem key

TDISP (PCIe-5)

Evaluation via Benchmarks in NonCC and CC Modes



Host Benchmarks
- Spec 2017

1

I/O Bandwidth Microbenchmarks
- H2D and D2H

2

Memory Bandwidth Microbenchmarks
- D2D

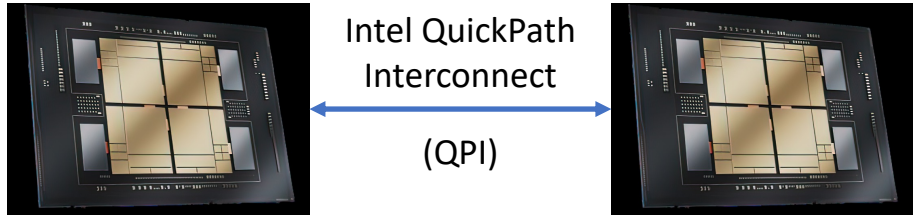
3

Reference:
[1] Mohan et al. “Securing AI Inference in the Cloud: Is CPU-GPU Confidential Computing Ready?” IEEE International Conference on Cloud Computing (CLOUD), July 2024.

System Configuration

CPU

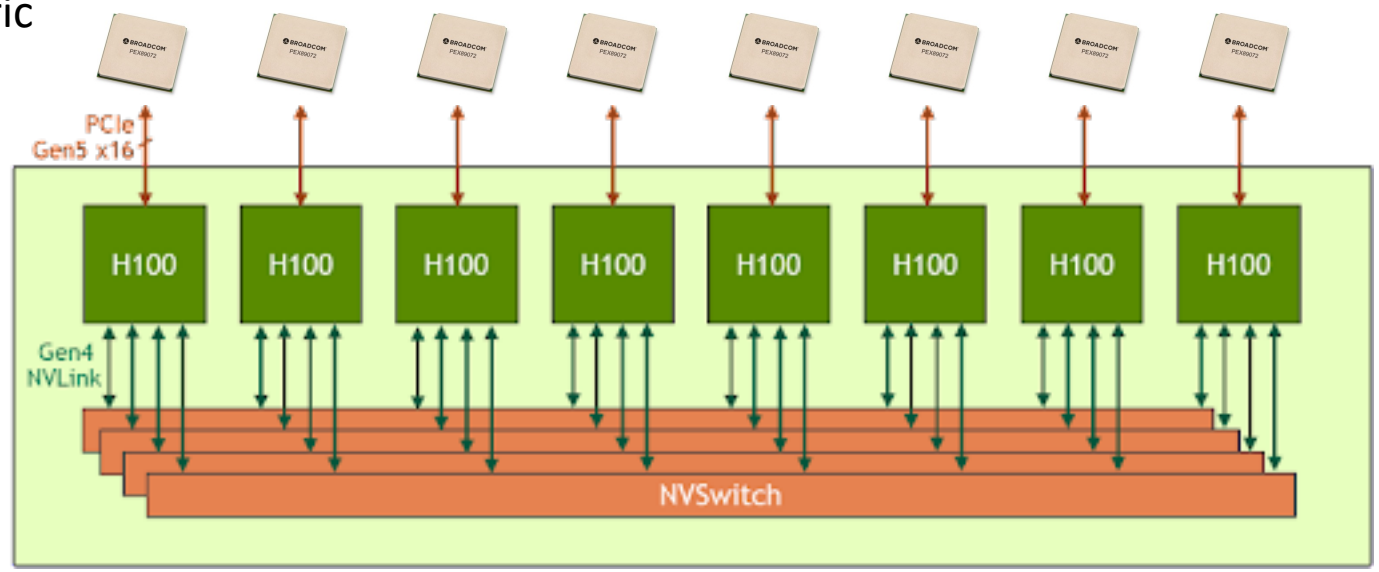
- Intel(R) Xeon(R) Platinum 8474C, 2*48 cores
- Memory 2TB
- Host OS: Ubuntu 22.04.2 LTS 6.2.0-mvp10v1+8-generic
- Guest OS: Ubuntu 22.04.3 LTS 6.2.0-mvp10v1+8-generic
- TDX 1.0



PCIe Gen5x16 128GB/s

NVIDIA HGX 8xH100 GPUs (HBM3)

- Memory 80GB
- L2 Cache 50MB
- NVIDIA Driver Version: 535.104.05
- CUDA Version: 12.2
- VBIOS 96.00.89.00.01



VMs (cc or nocc)

- SMP=8, Mem=160GB, 1 GPU @ vfio → all bound to individual numa node

Limitation: 1 GPU Per VM

Sources:
- https://developer.nvidia.com/blog/introducing-nvidia-hgx-h100-an-accelerated-server-platform-for-ai-and-high-performance-computing/
- https://www.nvidia.com/en-us/data-center/h100/
- https://www.techpowerup.com/gpu-specs/h100-sxm5-80-gb.c3900

CPU Performance Evaluation

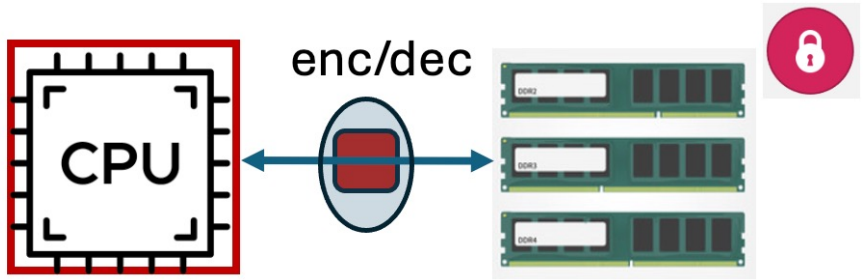
1

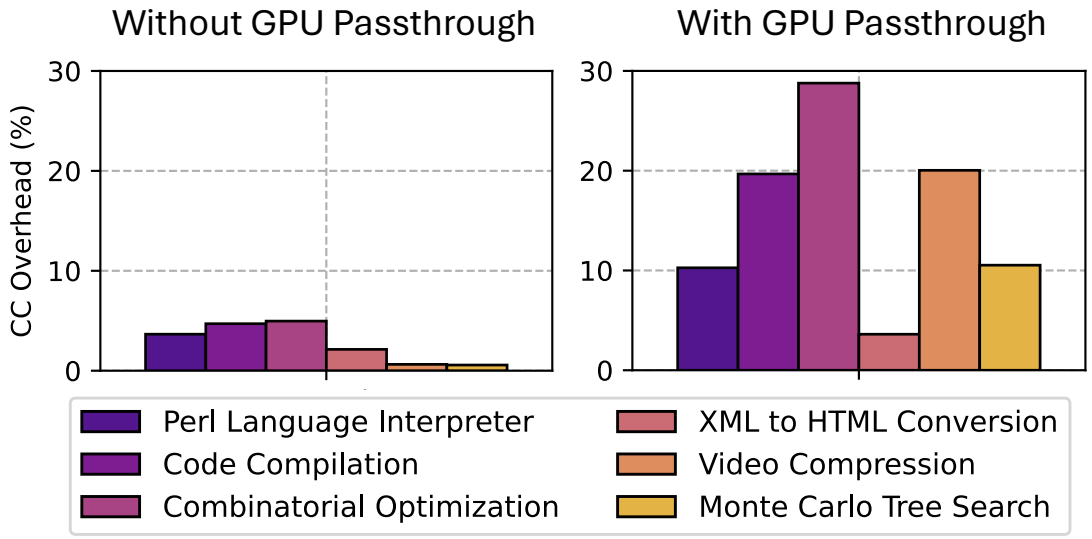
Intel Sapphire Rapid + TDX Enabled:

- CC-Mode Performance Overhead Directly Proportional to Memory Pressure.

- GPU passthrough causes higher performance overhead even when the application does not use the GPU.

- Informed Intel and NVIDIA.





Intel TDX TEE: Memory is Encrypted

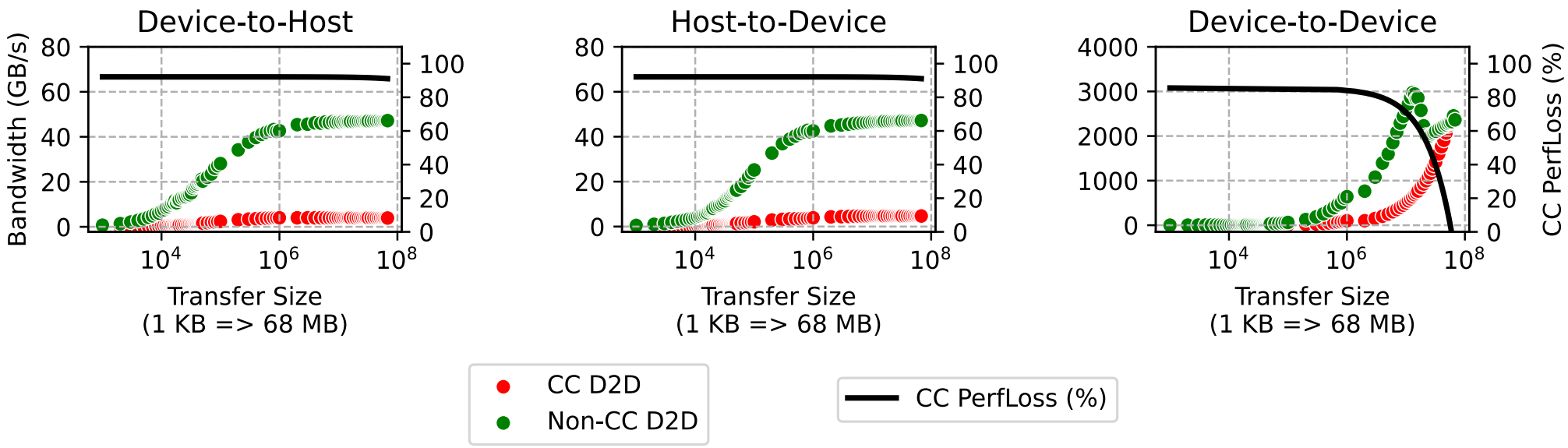
Device I/O and GPU Memory Performance

2

High Overhead on I/O Performance

3

High to Low Overhead on GPU Memory Performance

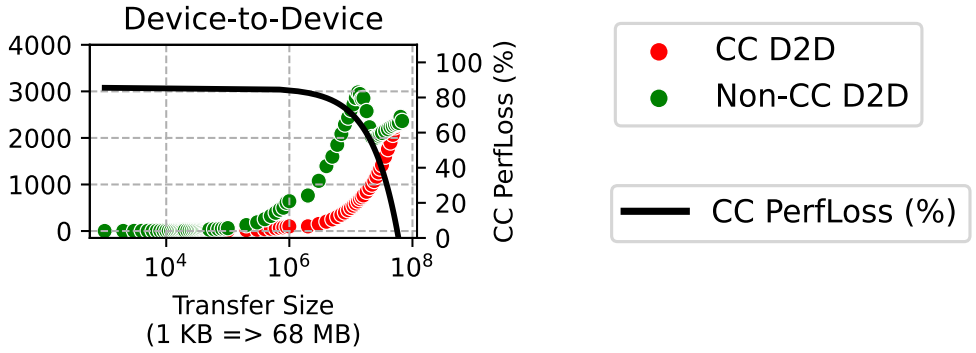


D2H & H2D : host to/from device communication

- Constant performance drop of 90% among different message sizes
- Related to bounce buffer enc/dec

D2D: Cuda memory copy mem buffer to mem buffer on same GPU

- Significant performance loss for small message sizes
- Surprising results: assumption was physical isolation on the card but it seems to be more in CC
- Drop for nonCC? (likely VRAM cache size)



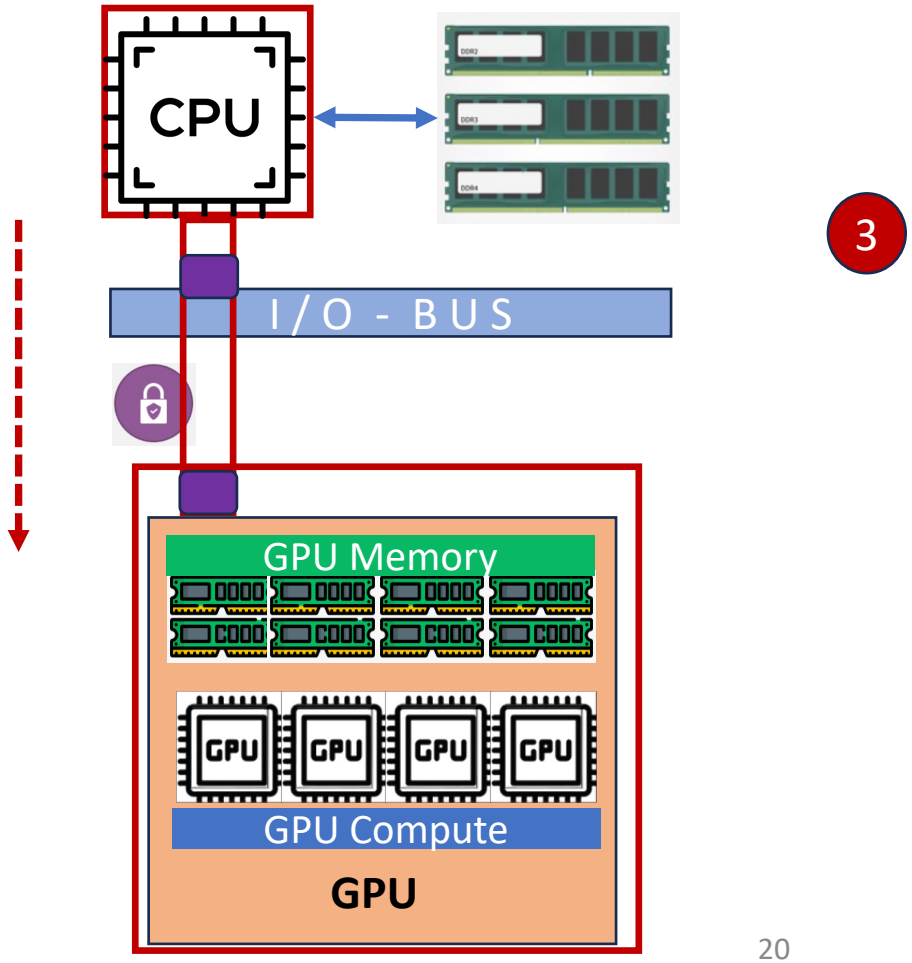
Why Are ***Intra Device*** Interactions Costly?

- Both Code and Data Need to be offloaded to device.
- Any CUDA function calls that interact with GPUs "MUST" go through encrypted software bounced buffer.

```
1   // copy host memory to device
2       cudaMemcpyAsync(d_A, h_A, mem_size_A, cudaMemcpyHostToDevice, stream);
3       cudaMemcpyAsync(d_B, h_B, mem_size_B, cudaMemcpyHostToDevice, stream);
4  ...
5  ...
6  ...
7  ...
8
9   // Setup execution parameters
10  dim3 threads(block_size, block_size);
11  dim3 grid(dimsB.x / threads.x, dimsA.y / threads.y);
12
13 ...
14 ...
15 ...
16 ...
17
18  // Execute the kernel
19  int nIter = 300;
20
21  for (int j = 0; j < nIter; j++) {
22    if (block_size == 16) {
23      MatrixMulCUDA<16>
24          <<<grid, threads, 0, stream>>>(d_C, d_A, d_B, dimsA.x, dimsB.x);
25    } else {
26      MatrixMulCUDA<32>
27          <<<grid, threads, 0, stream>>>(d_C, d_A, d_B, dimsA.x, dimsB.x);
28    }
29  }
30
31 ...
32 ...
33 ...
34 ...
35
```

Data Copy

Code Execution

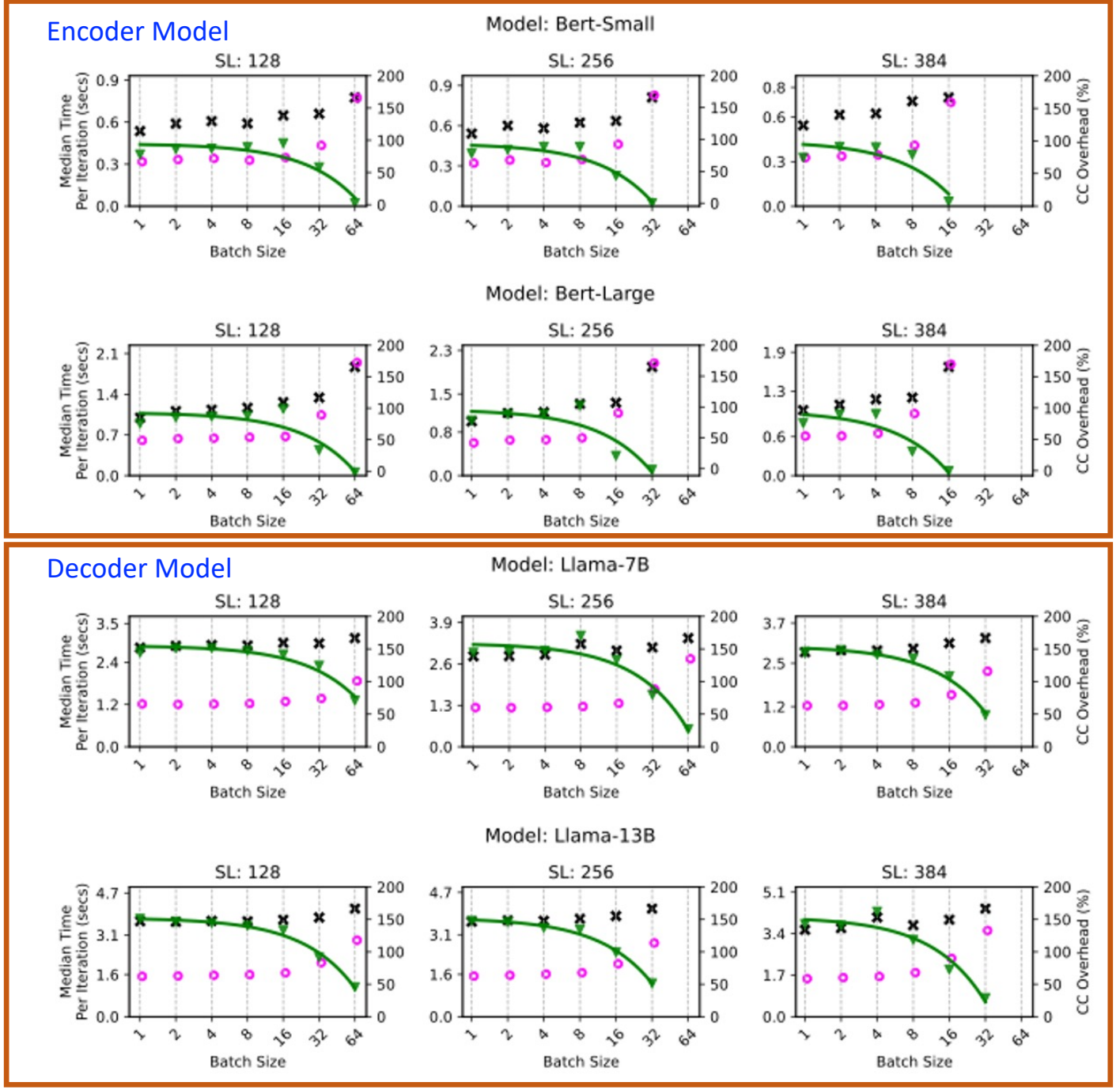


LLM Sensitivity Exploration

- What happen if users want to use LLMs out of the box
- Chose open-source Bert models and LlaMa
- Explore different sizes:
  - Bert-small (110M), Bert-Large (340M)
  - Llama-7B, Llama-13B.
- Different sequence lengths with different batch sizes inference (limited by GPU mem)

Observations:

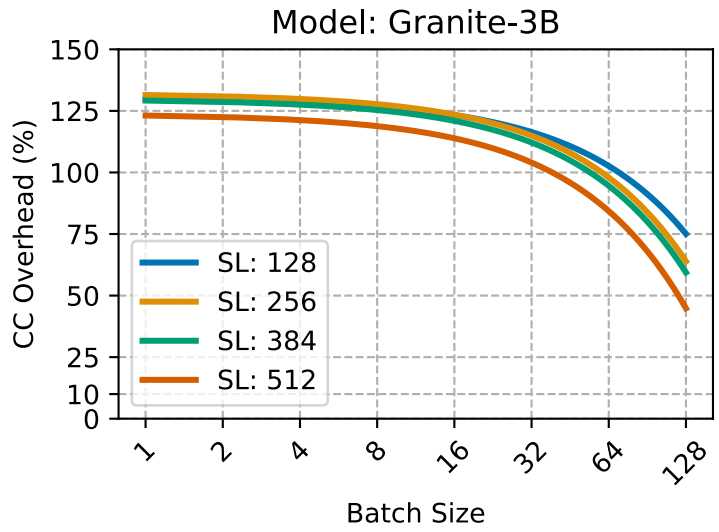
1. Overhead decreases as batch size increases across every LLM workload we tested and gap closes.
2. Overhead is not always negligible for large batches.
3. CC overhead is getting amortized as increasing batch size
4. **Compute-2-IO ratio increases with larger models and larger batch sizes**

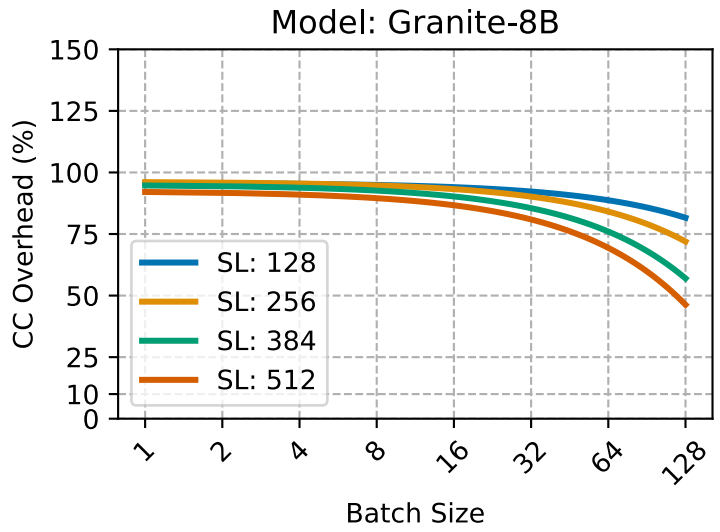


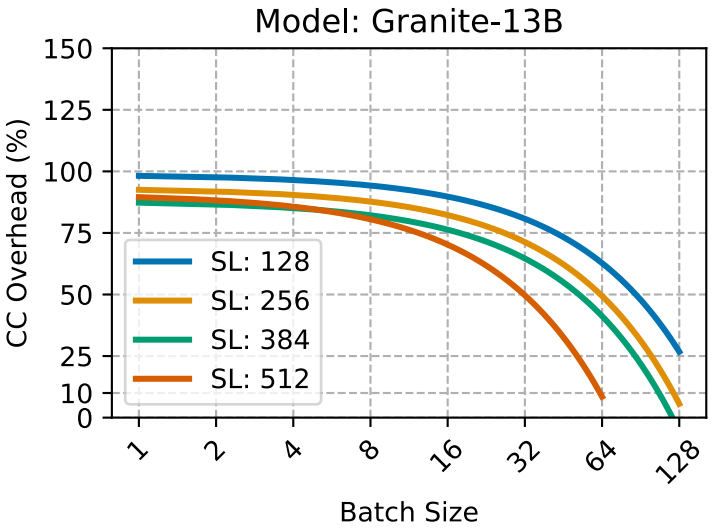
IBM Granite Models

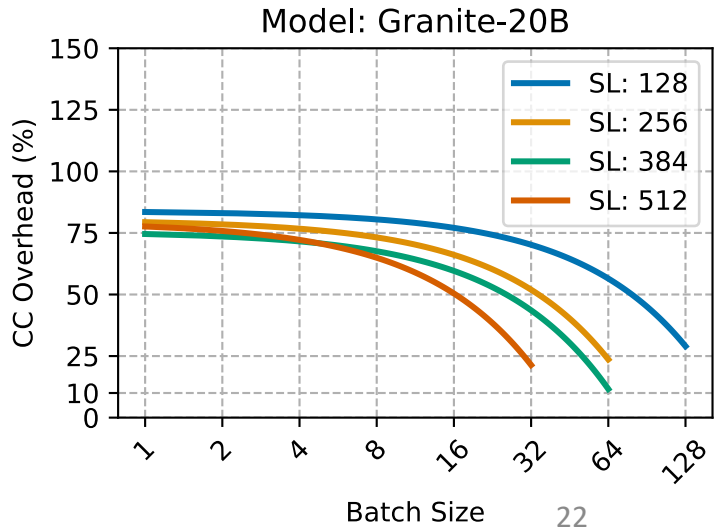
- Increasing presence of “private” models.
- “Granite” is a family of multi-size foundation generative AI models for both language and code.
- Observation:
  - Similar trends observations in the models that IBM developed
  - Higher CC overheads for smaller batch size with significant reduction for higher batch sizes.
  - Compute-2-IO ratio increases for larger models thus it reduce the CC overhead

Model Size, Batch Size, Sequence Length:
Larger is Better for H100 CC



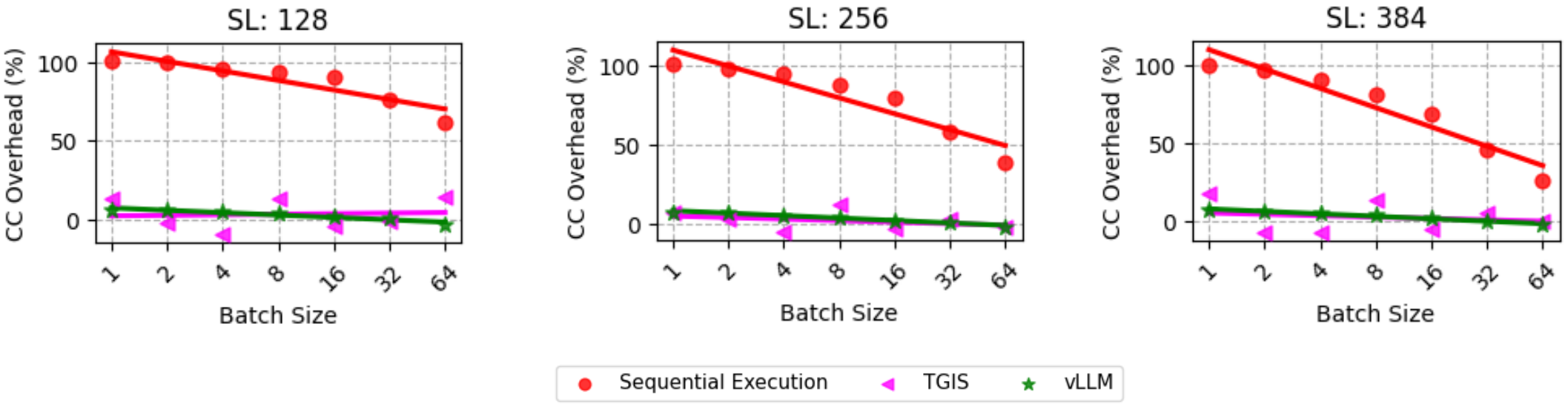






Compute-IO Overheads Can be Hidden Using Parallelization Strategies

Granite 13-B CC Mode Percentage Overhead

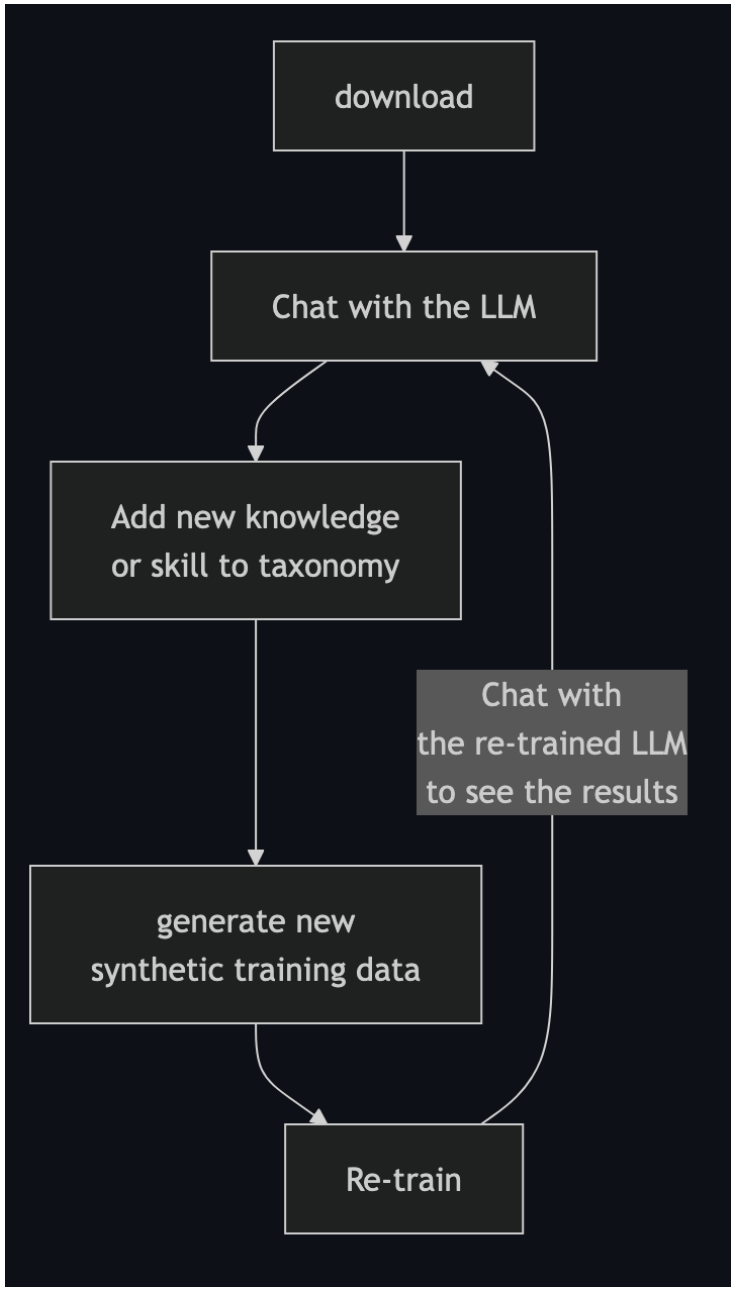


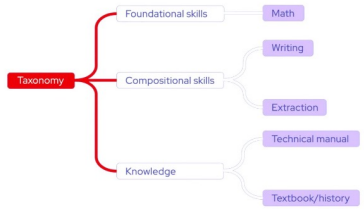
% Overhead over Non-CC Baseline (lower is better)

AI service backends, such as vLLM and TGIS, employ parallelization strategies (e.g., pipelined execution) which can significantly hide the overhead (due to encrypted PCIe link) as they enable “Compute-IO overlapping”.

IMPORTANT

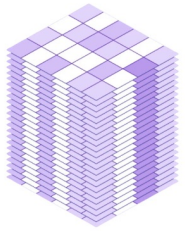
Instruct Lab – Fine tuning (beyond just inference)





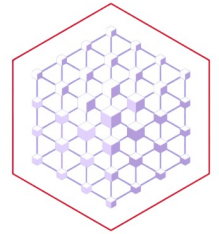
Taxonomy-driven Data Curation

Folder structure with Q&A pairs for topics to teach a model



Large-scale Synthetic Data Generation

Generate additional training data to expand dataset automatically



Model Training with New Data

Phased, large-scale alignment tuning (with knowledge and skills)

InstructLab

Instruct Lab

- Core element of RHEL AI
- Open-source framework to democratize the tuning and development of LLMs
- For user without data science background, they can use Instruct Lab to ensure the model is tailored to a specific domain with improved accuracy and relevance

✅ We have successfully demonstrated Instruct Lab in a Confidential VM with single Confidential GPU without modifying the framework/models.

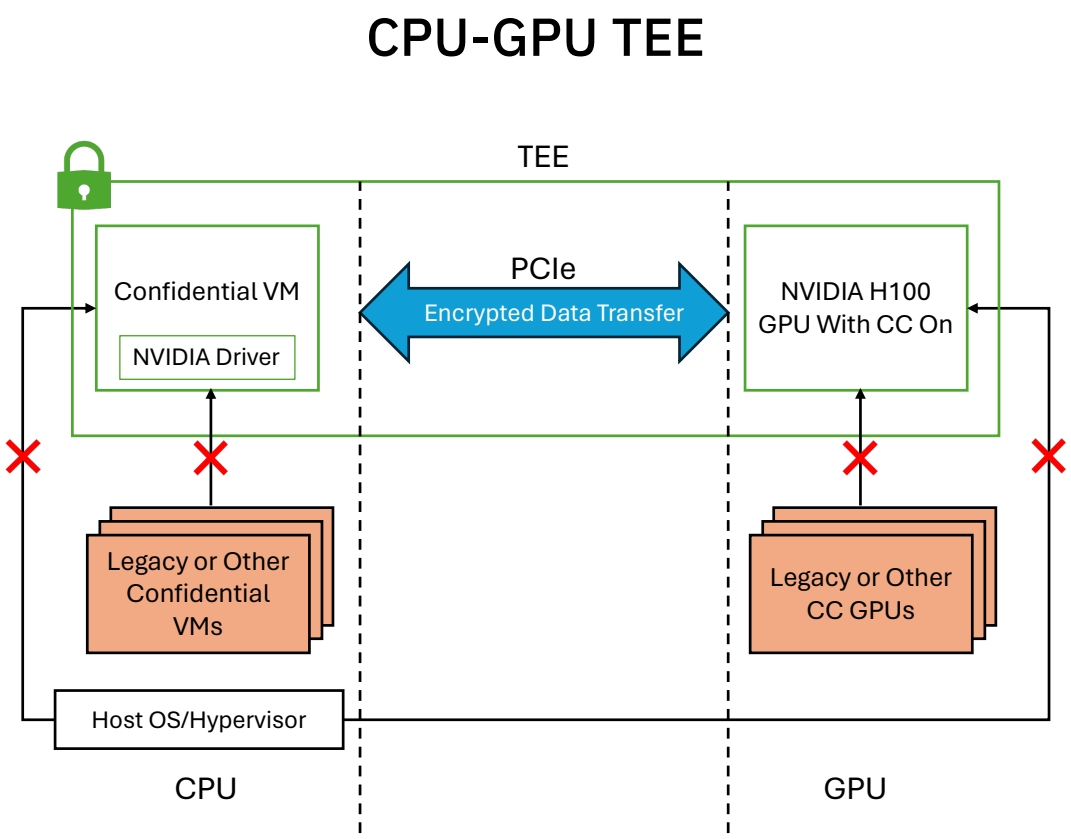
References:
[1] https://www.redhat.com/en/blog/what-is-instructlab
[2] https://developers.redhat.com/articles/2024/08/05/how-instructlab-enables-accessible-model-fine-tuning-gen-ai

Are CPU-GPU TEE's Ready?

Yes

1. Despite, high overheads due to bounce buffers-based encryption/decryption when interacting over the PCIe link, employing parallelization strategies can already enable real-world use despite high overheads.

2. Blackwell generation of GPUs will support TEE-IO CPUs which will reduce performance overheads significantly due to elimination of the bounce buffer.



From Confidential Computing to Zero Trust

Is the state-of-the-art Confidential Computing ready for Zero Trust computing where users do not have to blindly trust compute infrastructure in the cloud but can always verify every compute component?

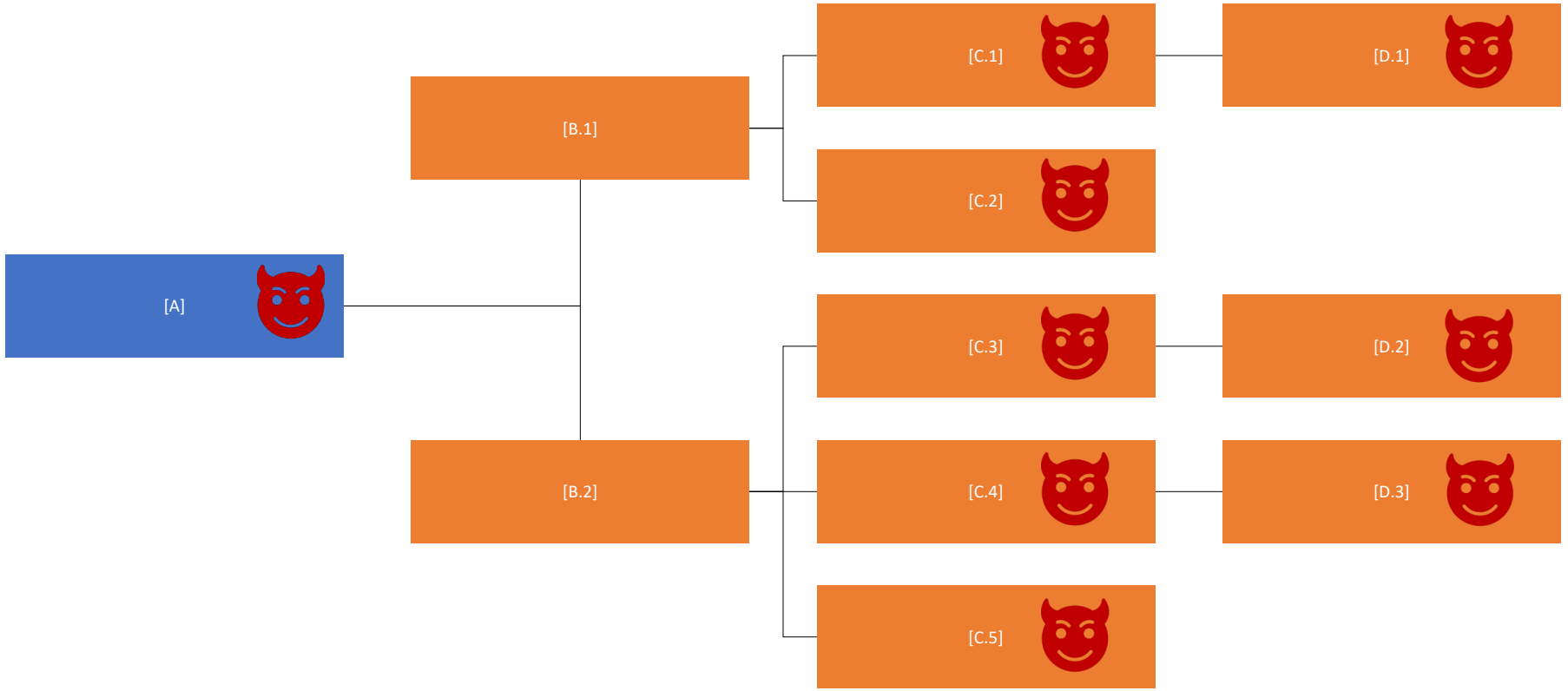
Why Zero Trust

Network Framework

“Never trust, always verify” each network tenant or user

SolarWinds Hack

SolarWinds systems were hacked in early 2020. The software updates with malicious code of SolarWinds were sent out to the clients’ systems.



References:
[1] *What is a Zero Trust Architecture*, Palo Alto Networks.
[2] *A 'Worst Nightmare' Cyberattack: The Untold Story Of The SolarWinds Hack*, NPR News, April 2021.
[3] *Why Implement Zero Trust*, YouTube – IBM Technology channel, July 2022. https://www.youtube.com/watch?v=lT11tGaEC3s

Applying the principles of zero trust

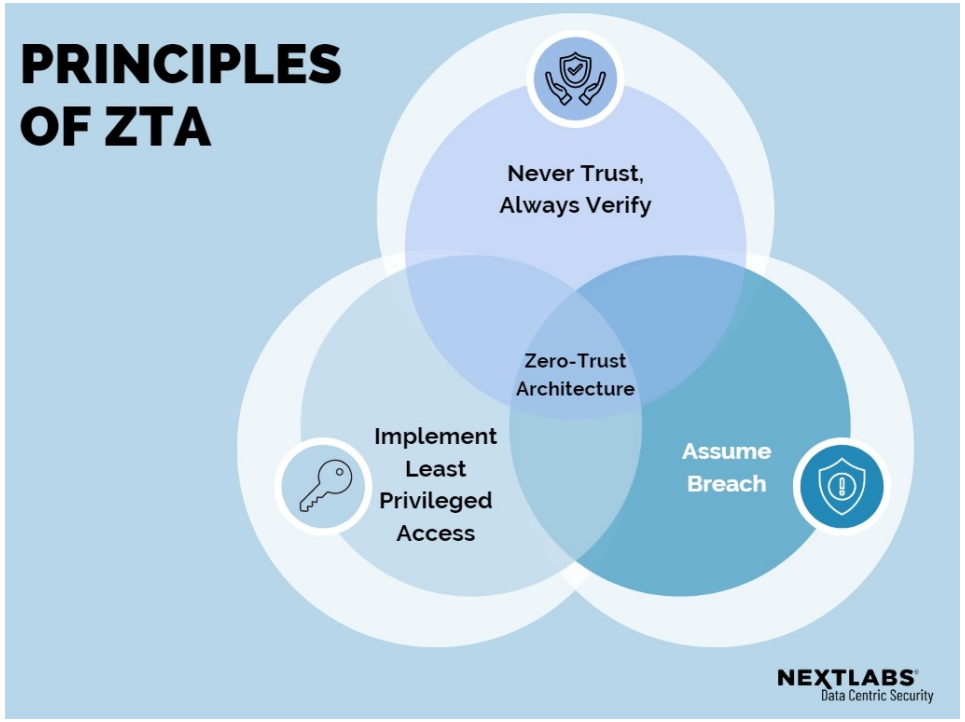
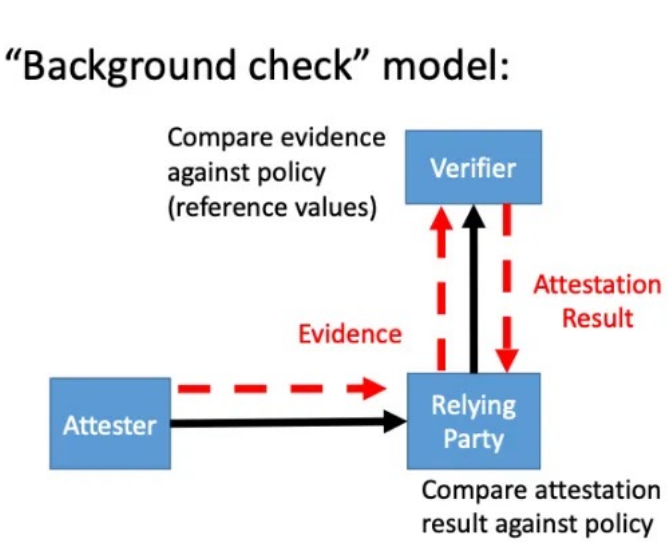


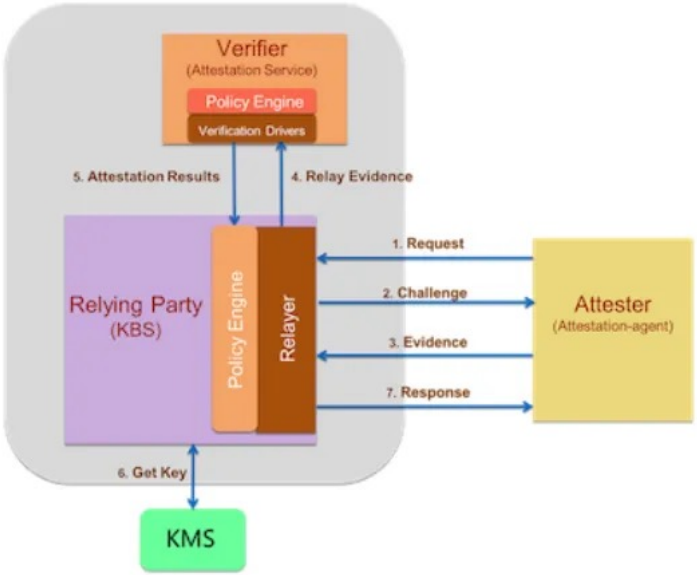
Image: https://www.nextlabs.com/solutions/zero-trust/

- Attestation
- Supply chain security
- Risk assessment
- Side channel prevention/mitigation
- …

Attestation

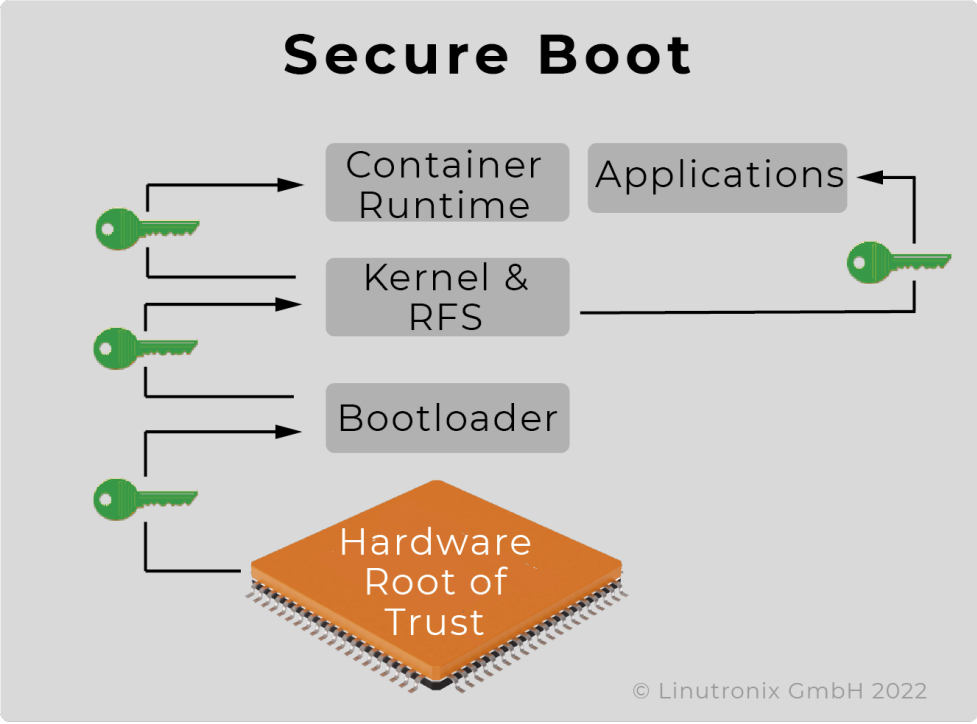
- The [Confidential Containers] (CoCo) project follows the [IETF Remote Attestation Procedures (RATS)] "background check" model
- Hardware-based attestation which relies on CC hardware to measure all the CoCo guest components and state when it loads them





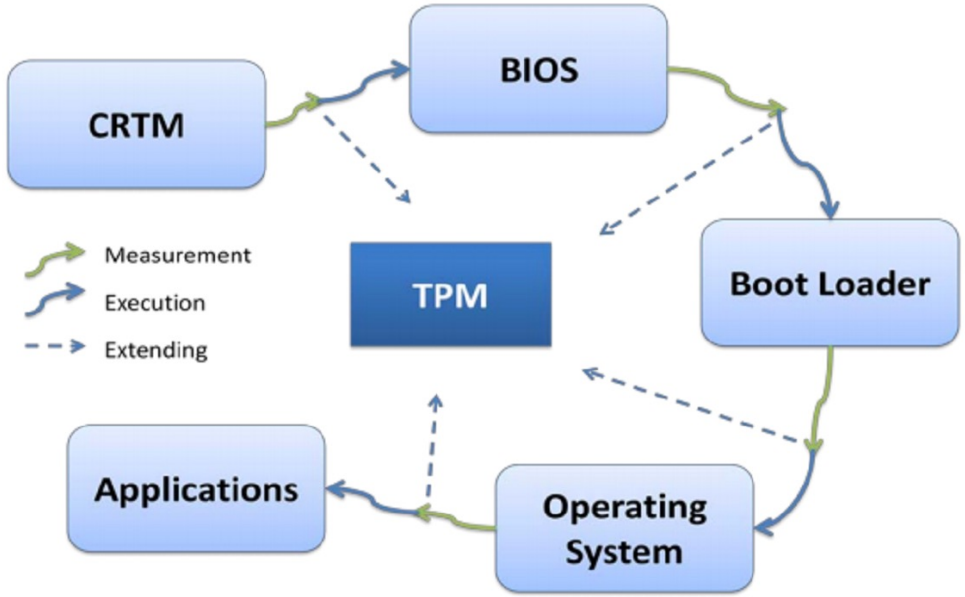
https://www.redhat.com/en/blog/understanding-confidential-containers-attestation-flow

Building a chain of trust



https://www.linutronix.de/industrial_linux/security_tpm_hsm_modul.php

Measured Boot



https://www.laseroffice.it/blog/2020/10/25/measured-boot-trusted-boot-e-tpm-unintroduzione/

- Hardware Roots-of-Trust are the basis of secure and measured boot required for attestation
- Can we trust the hardware that helps build the chain of trust?

Semiconductor supply chain risks

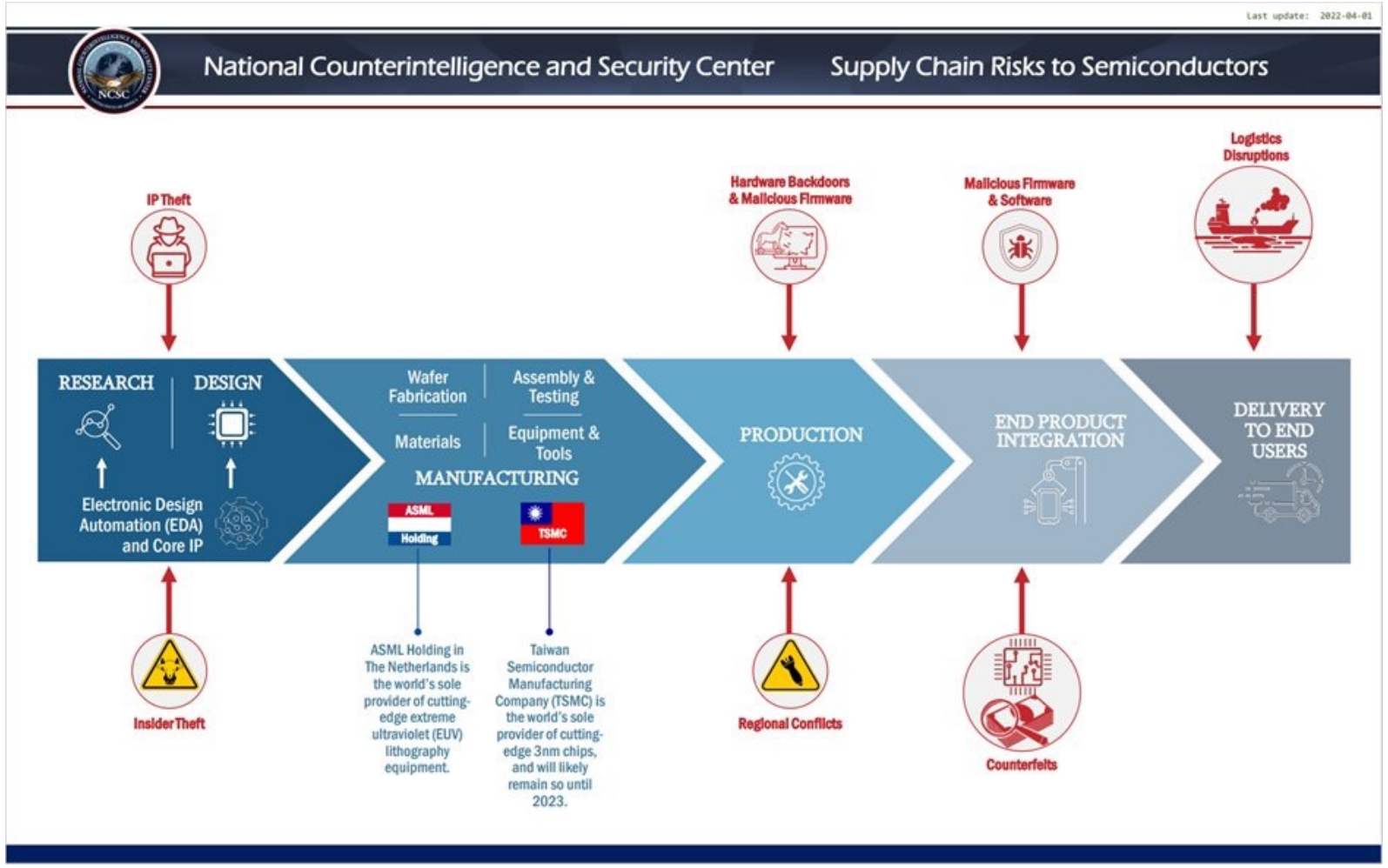


Image: https://www.dni.gov/files/NCSC/documents/supplychain/semiconductor-supply-chain-2022-39E2C6B0-.pdf

Is the state-of-the-art Confidential Computing ready for Zero Trust computing where users do not have to blindly trust compute infrastructure in the cloud but can always verify every compute component?

Many challenges remain in bringing zero trust all the way down to hardware:

- Secure and private AI and large-scale workloads in heterogeneous computing systems
- Dynamic or runtime verification/reverification
- Trusted computing and cryptographic implementation challenges of real-time hardware for IoT and autonomous vehicles
- Supply chain security of hardware and firmware
- Threat models for applications of zero-trust architecture
- Security and trust in Cloud/Edge computing and infrastructure
- Role of open-source designs and standards for security and trust
- Other merging topics in security and trust such as post-quantum cryptography, homomorphic encryption, secure multi-party computation etc.
- ……

Call for Action!

Workshop on Zero Trust Hardware Architectures (ZTHA)

- **Sandhya Koteshwara** (sandhya.koteshwara@ibm.com)
- **Mengmei Ye** (mye@ibm.com)
- **Hubertus Franke** (frankeh@us.ibm.com)

https://zerotrustworkshop.github.io