Confidentiality in the Era of Generative and Agentic AI

Hubertus Franke, Mengmei Ye, Apoorve Mohan, Marcio Augusto De Lima E Silva

(frankeh@us.ibm.com, mye@ibm.com, apoorve.mohan@ibm.com, MARCIO.A.SILVA@ibm.com)

IBM Research

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.

Generative and Agentic AI

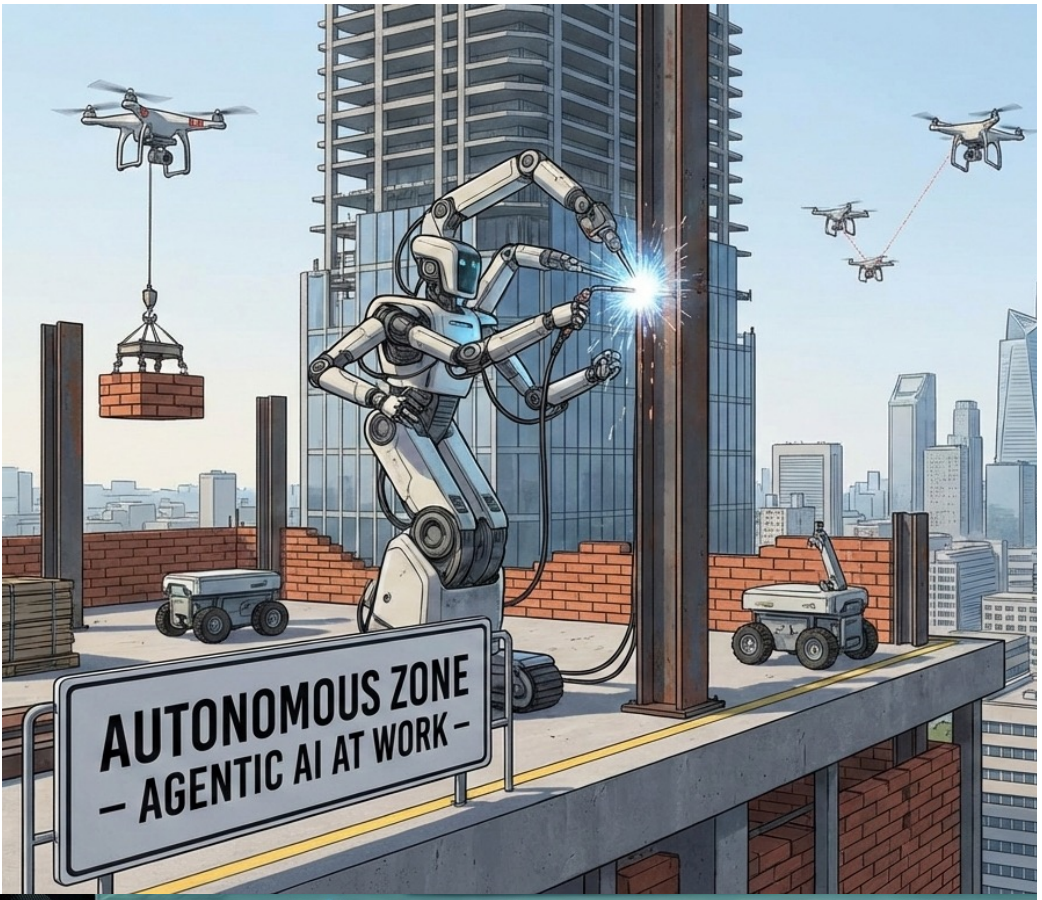


Image sources: Gemini

Agentic AI becomes increasingly autonomous

- Book your trips
- Summarize meetings
- Draft your emails
- Implement your systems/applications/workflows

What if Agentic AI does something *malicious* unintentionally?

- Leak sensitive information
- Compromise code integrity

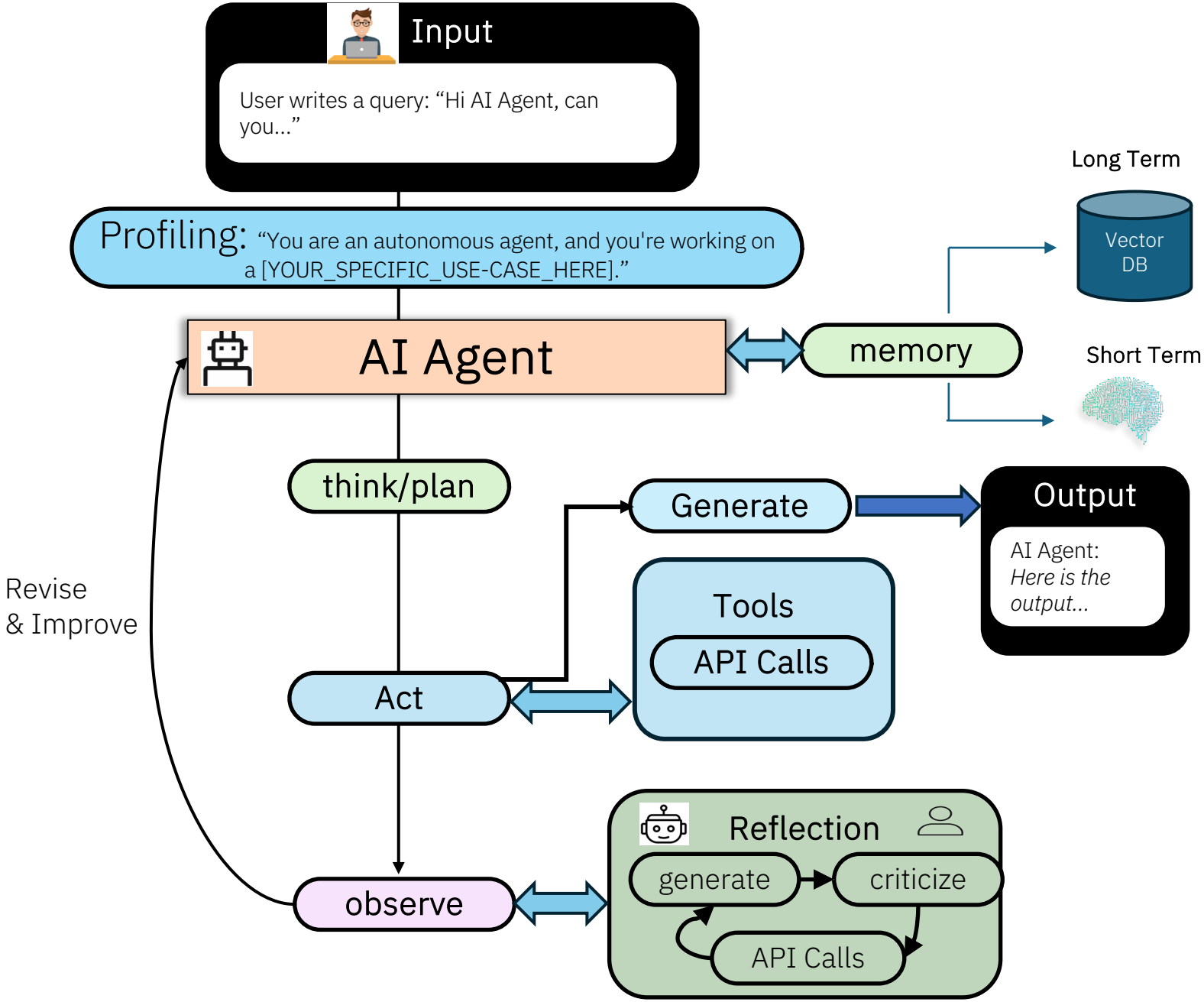
Especially for regulated industry like finance and healthcare

- What can Confidential Computing offer additionally in security besides internal Agentic Guardrails?

AI Agents: Characteristics

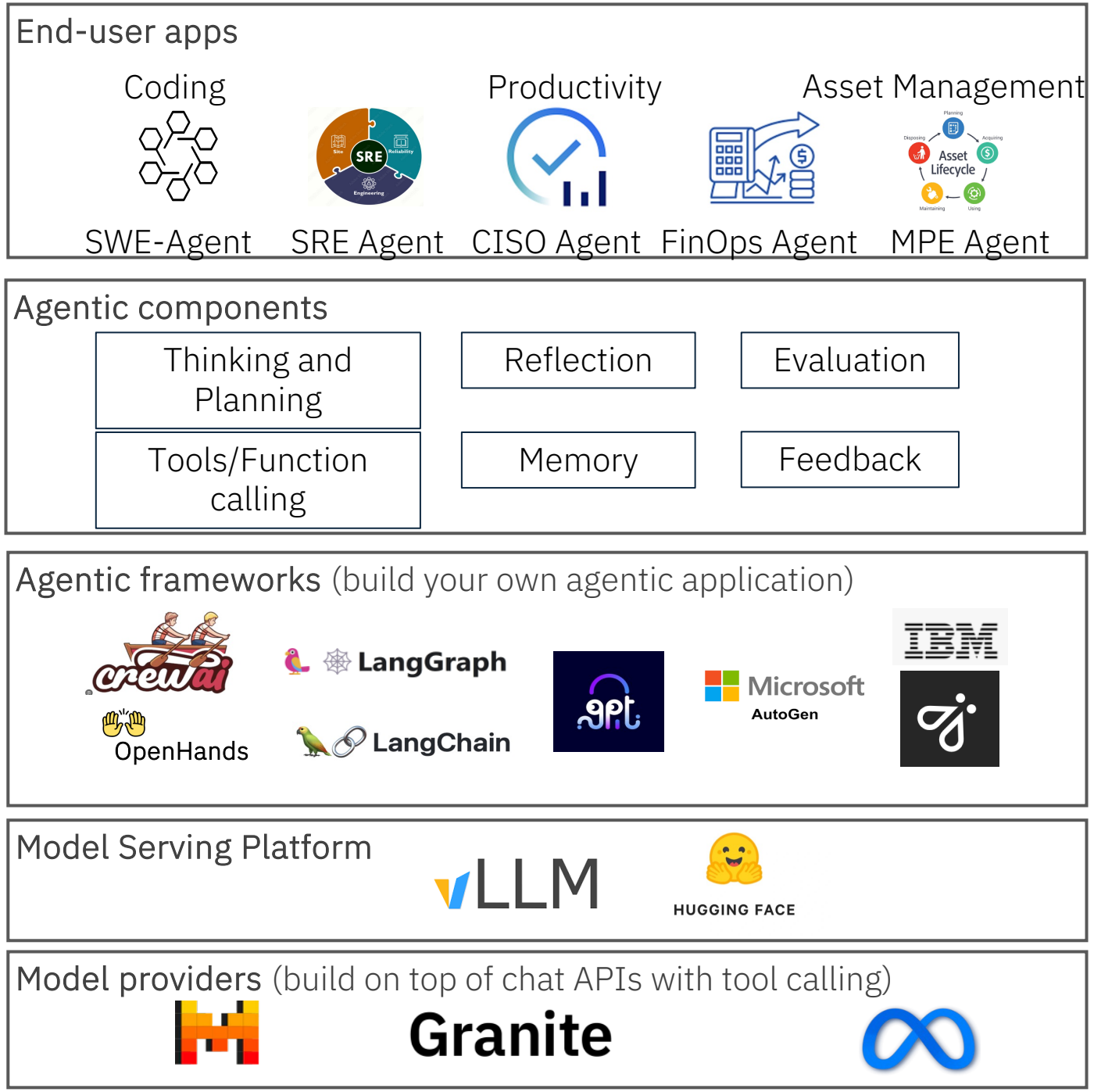
LLM - powered systems that can autonomously:

- ✓ Think/Plan → Act → Observe ("TAO")
- ✓ Self-Reflect & Course Correct
- ✓ Act on "tools"
- ✓ Possess "memory"
- ✓ Generate output



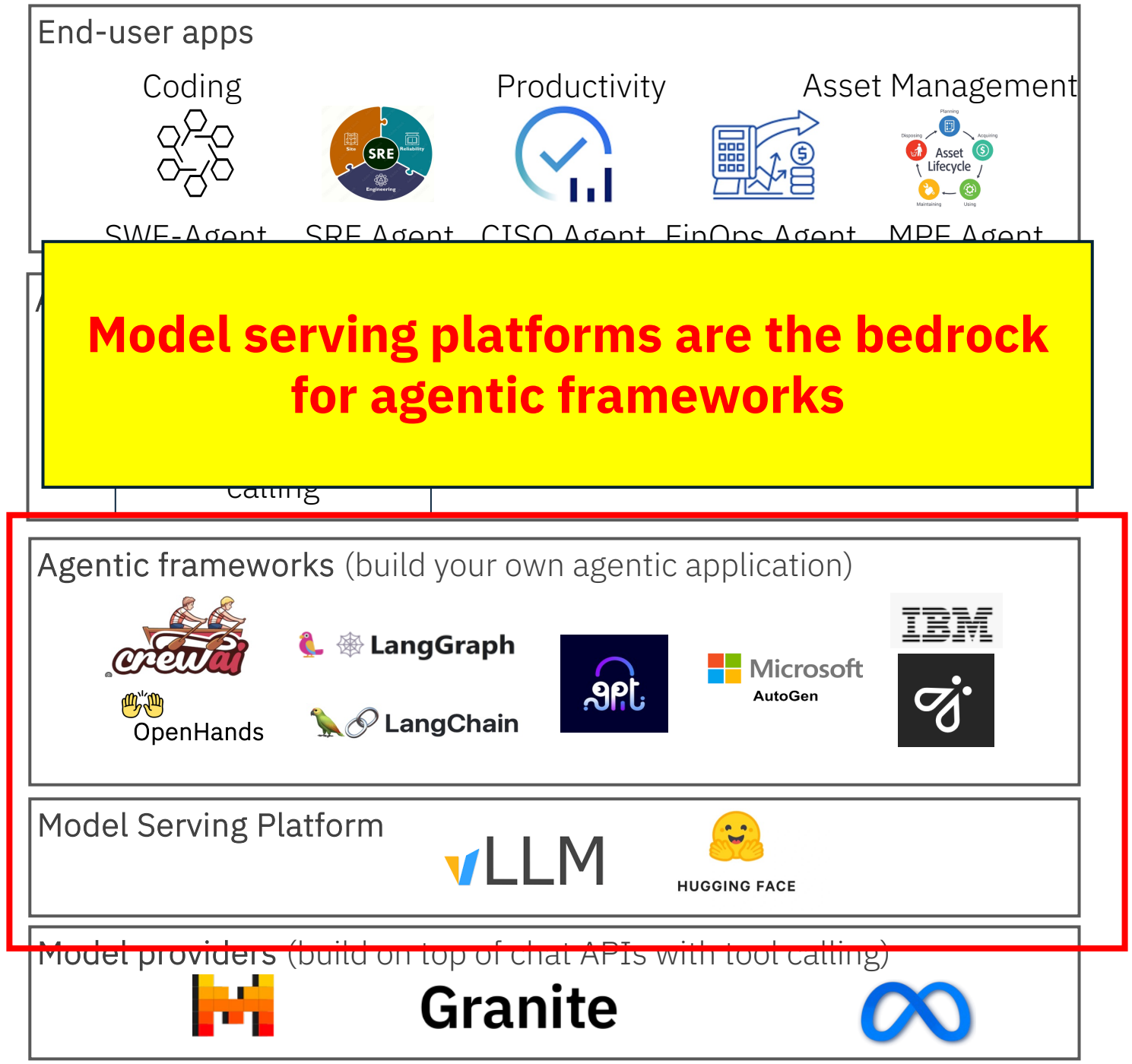
AI Agents: Stack and Strategy

- Enterprise domain agents powered by Open LLMs
- Tools, middleware and frameworks to build custom performant agents
- Platform for multi-agent development, collaboration and orchestration



AI Agents: Stack and Strategy

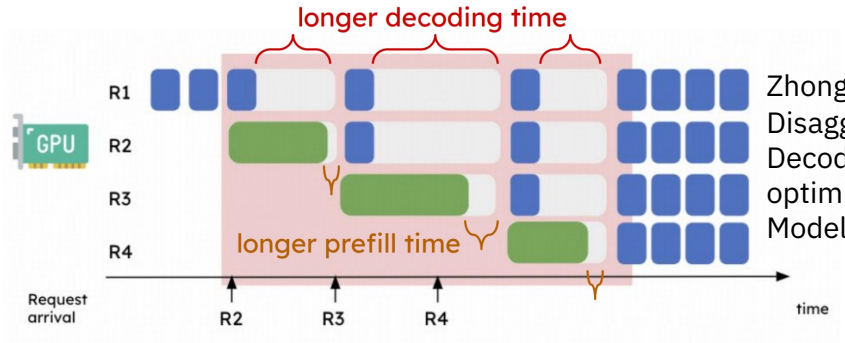
- Enterprise domain agents powered by Open LLMs
- Tools, middleware and frameworks to build custom performant agents
- Platform for multi-agent development, collaboration and orchestration



Industry-wide challenge of scaling inference

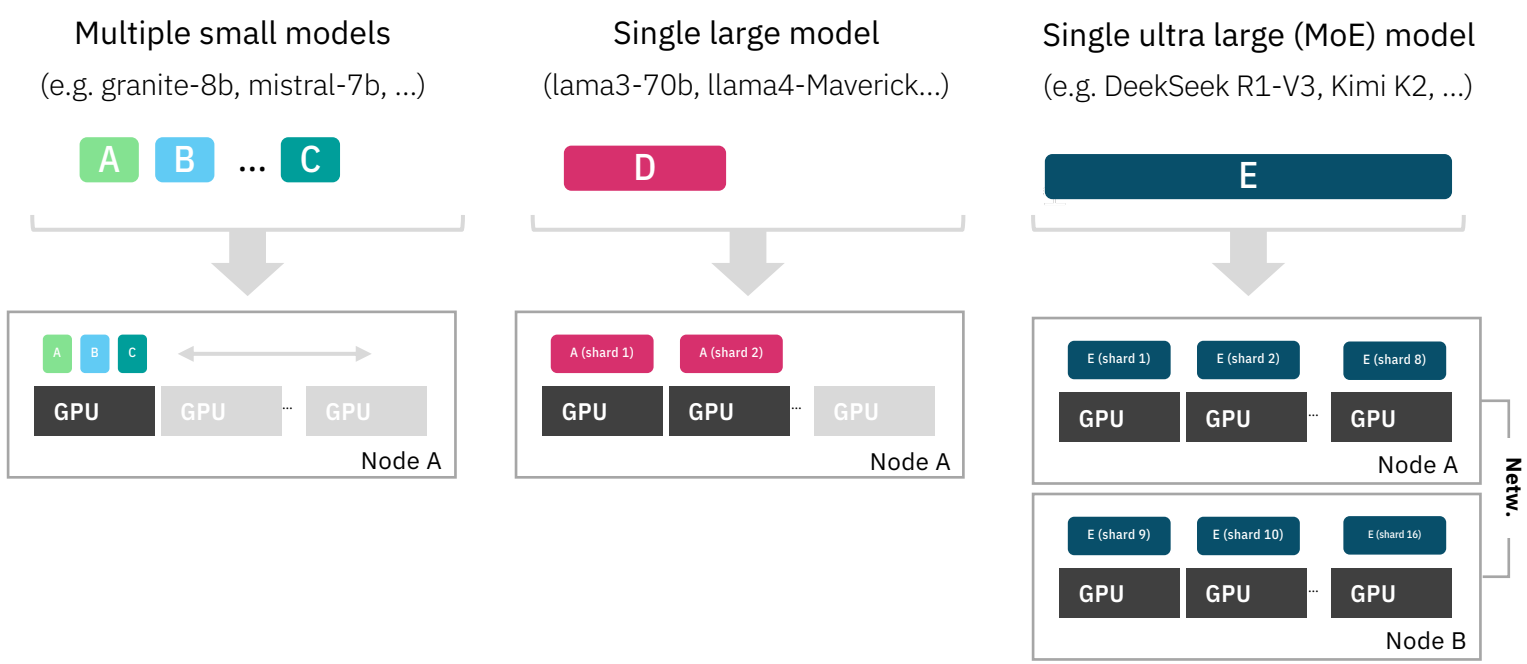
- Distributed inference is essential for cost-effective GenAI at scale, but introduces unique **operationalization challenges**

- LLM inference workloads with variable, resource-heavy and hardware-affinity nature of requests
- **Ensuring SLO** (throughput, TTFT, latency) while **optimizing** resource utilization and reducing operational complexity
- Leveraging and managing **heterogenous hardware** for better cost-efficiency
- Distributed **KV cache management** as key part in inference efficiency
- **Secure/confidential inference** without the need of blindly trusting infrastructure underneath



Zhong et al., "DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving", OSDI'24.

Batch, interactive, multi-turn and agentic patterns



Industry-wide challenge of scaling inference

Key bottleneck in monolithic inference: Prefill is compute-bound and decode is memory-bound, which impacts scaling when co-located.

To address it, **PD disaggregation** is the solution.

Source: Zhong et al., “DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving”, OSDI’24.

- Distributed in
  cost-effectiv
  introduces u
  challenges

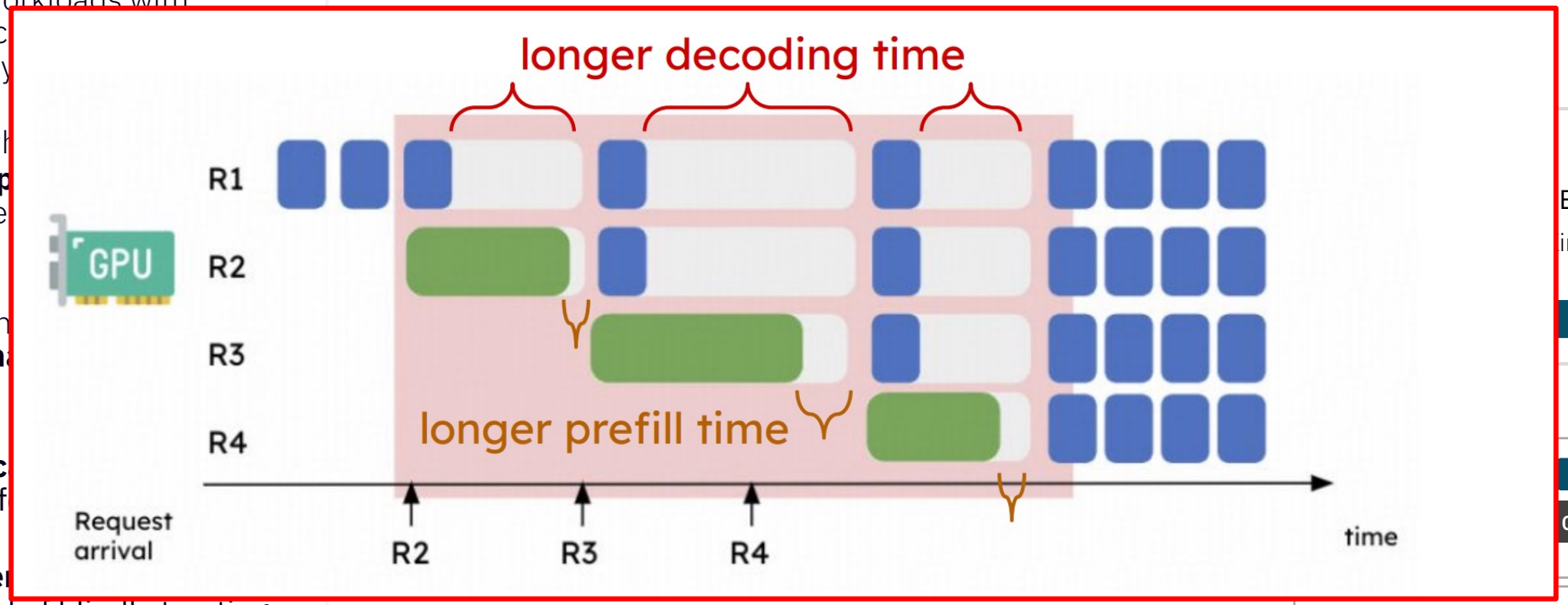
- LLM inference workloads with
  variable, resourc
  hardware-affinity

- **Ensuring SLO** (th
  latency) while **op**
  utilization and re
  complexity

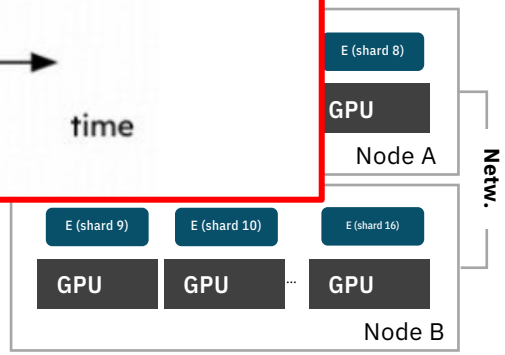
- Leveraging and n
  heterogenous ha
  cost-efficiency

- Distributed **KV c**
  as key part in inf

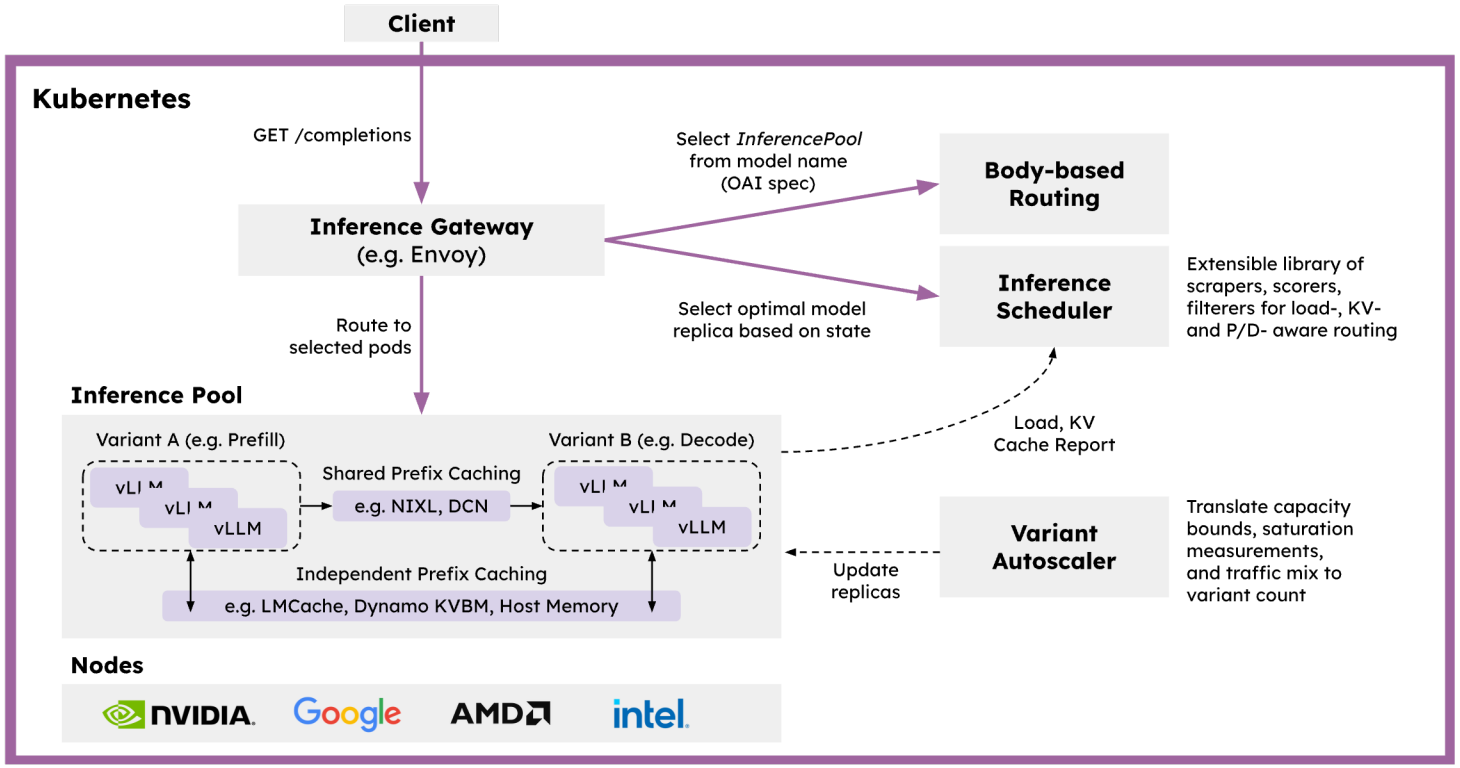
- **Secure/confide**
  without the need of blindly trusting infrastructure underneath



E) model
imi K2, ...)

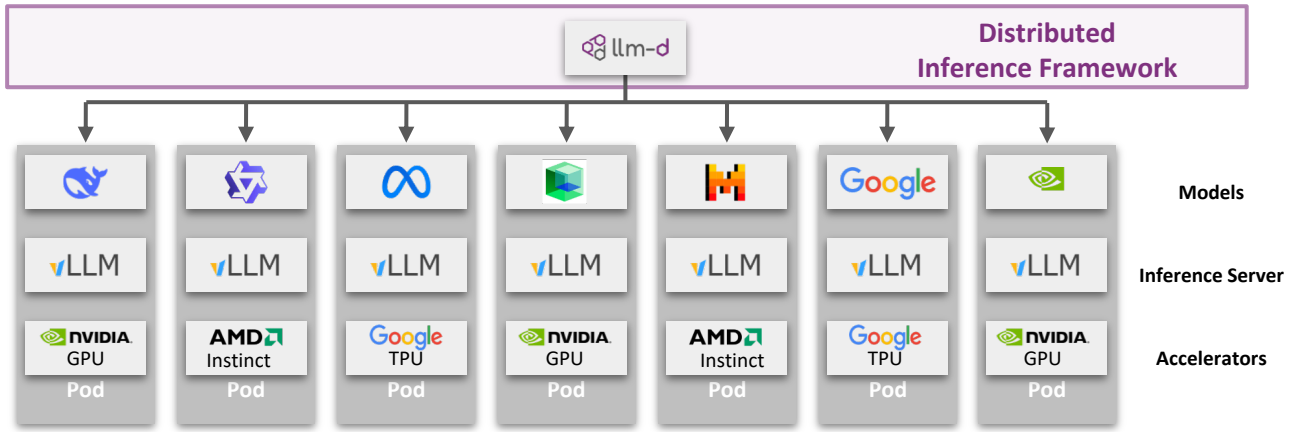


llm-d: Enterprise GenAI Scaling Inference Platform



- llm-d: An **open-source** distributed LLM serving platform for cloud-native enterprise generative AI and agentic AI systems.
- PD Disaggregation enabled by llm-d
- It also support routing strategies such as content aware.
- Individual inference engine of llm-d is based on vLLM
- vLLM broadly supports many AI models and accelerators/platforms
- The vLLM instances are scheduled and deployed in K8s/OCP platforms at the pod level

- Blog posts for more info: https://llm-d.ai/blog
- Github repo: https://github.com/llm-d



What about multi-tenant deployments (e.g., in the cloud)?

- Distributed inference is essential for cost-effective GenAI at scale, but introduces unique **operationalization challenges**

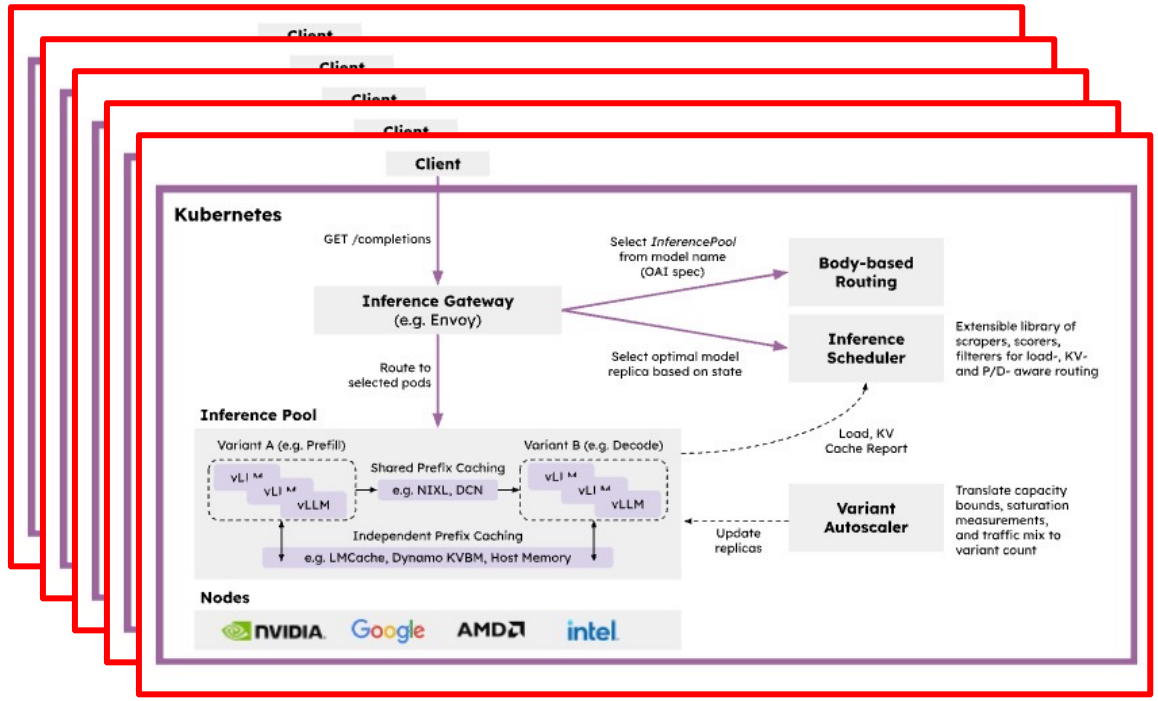
- LLM inference workloads with variable, resource-heavy and hardware-affinity nature of requests

- **Ensuring SLO** (throughput, TTFT, latency) while **optimizing** resource utilization and reducing operational complexity

- Leveraging and managing **heterogenous hardware** for better cost-efficiency

- Distributed **KV cache management** as key part in inference efficiency

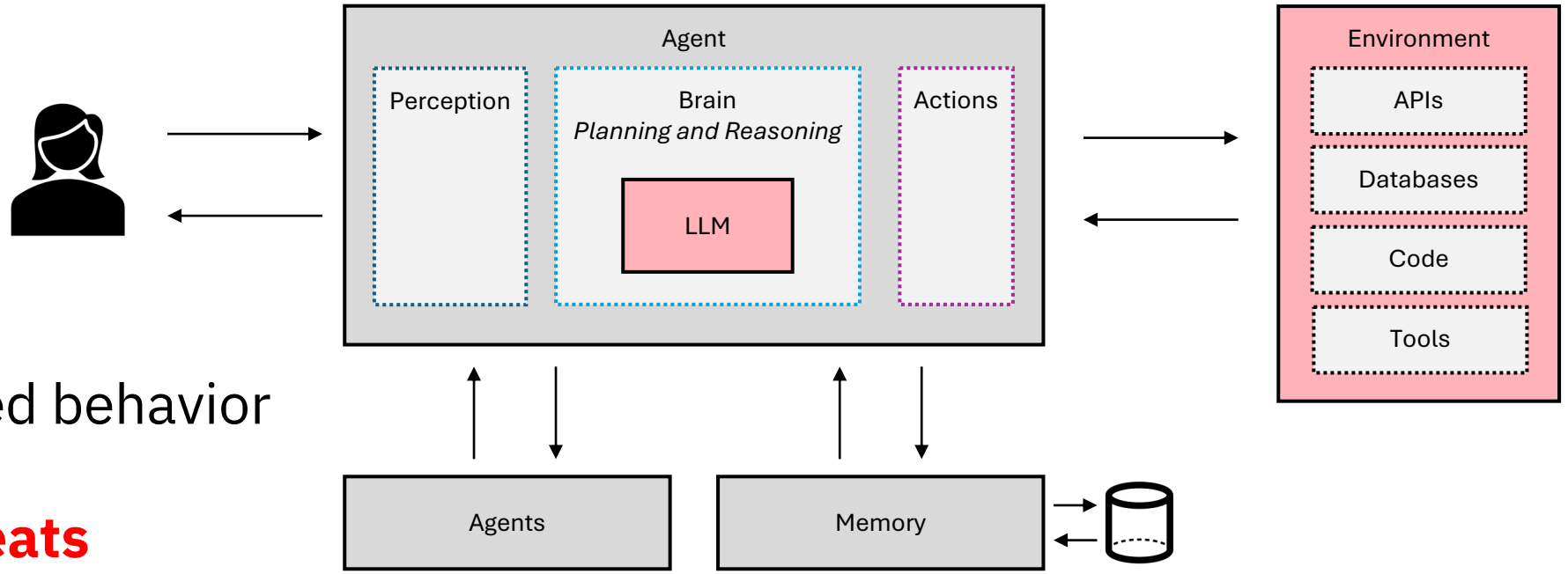
- **Secure/confidential inference** without the need of blindly trusting infrastructure underneath



Threats: (a) Agentic app: Excessive Agency & Prompt Injection
(b) Platform Level

(a) Excessive in:

- Functionality
- Autonomy
- Permissions
- Training Bias
- Emergent unintended behavior



(b) **Platform level threats (Our Focus)**

- Compromise integrity of software stacks
- KV cache side channel attacks
- Noisy/malicious neighbor attacks
- System attacks

Consequences:

- Data Breaches
- Operational Disruption

Confluence of Agentic and Confidential Computing Protecting Intellectual Property and Sensitive Data

What needs to be protected (confidentiality and integrity):

- ❑ AI Models and Weights
- ❑ User inputs
- ❑ Software stack including vllm serving engines etc.

Requirements:

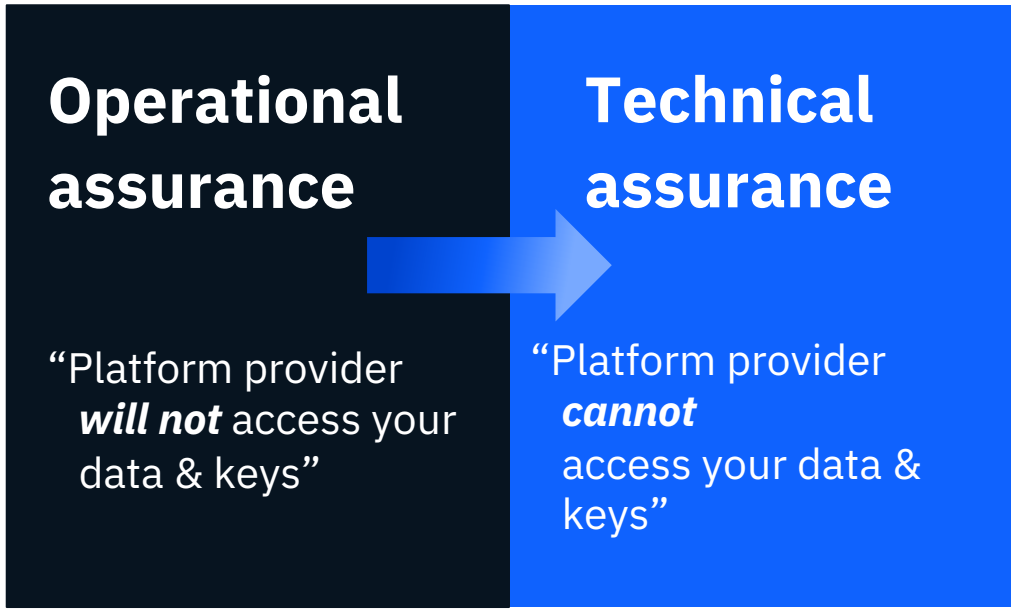
- ❑ Data Sovereignty and Security Compliance
- ❑ Data Confidentiality and Integrity
- ❑ Low performance overhead

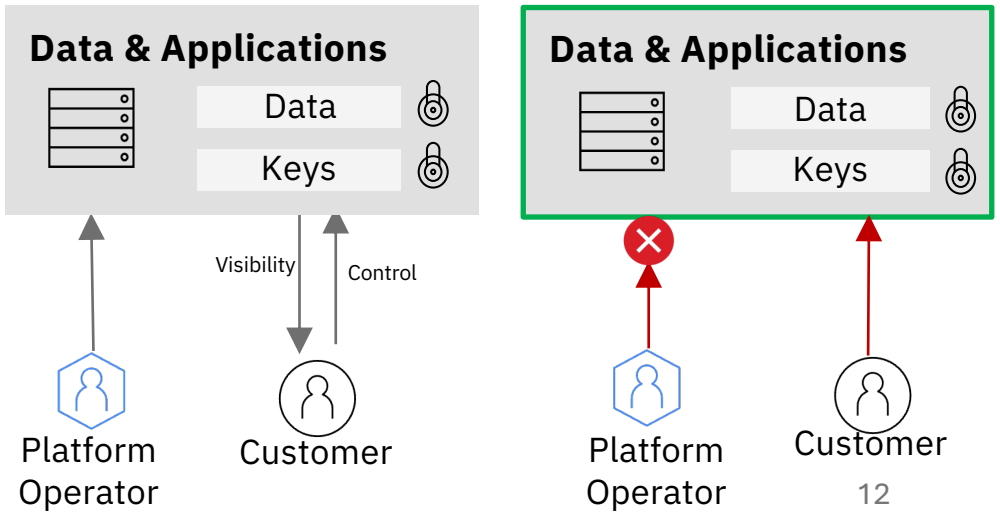
Target platforms:

- ❑ Cloud
- ❑ Edge
- ❑ Server

What **Confidential Computing** offers:

- ❑ Data and model protection in compute, in transit and at rest efficiently
- ❑ Attestable stack from hardware all the way up to software





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)
- VM based TEE
- **Security requirements**
  - Data should *only* be decrypted inside TEEs
  - CSPs, hypervisors should have no code running inside of enclaves

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

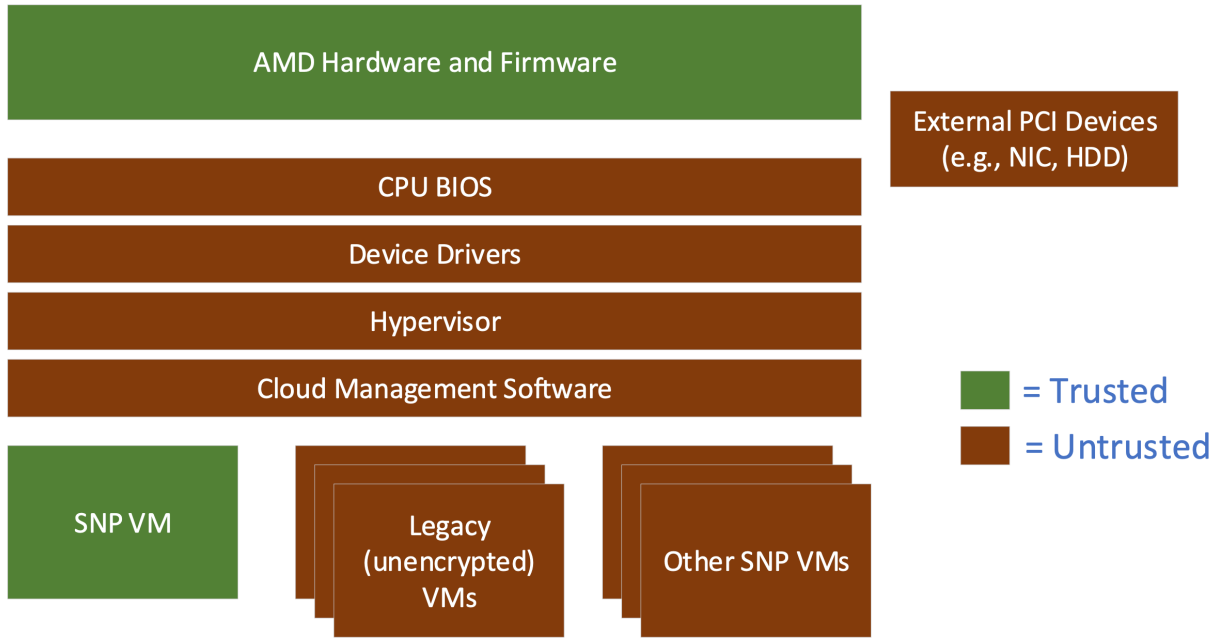
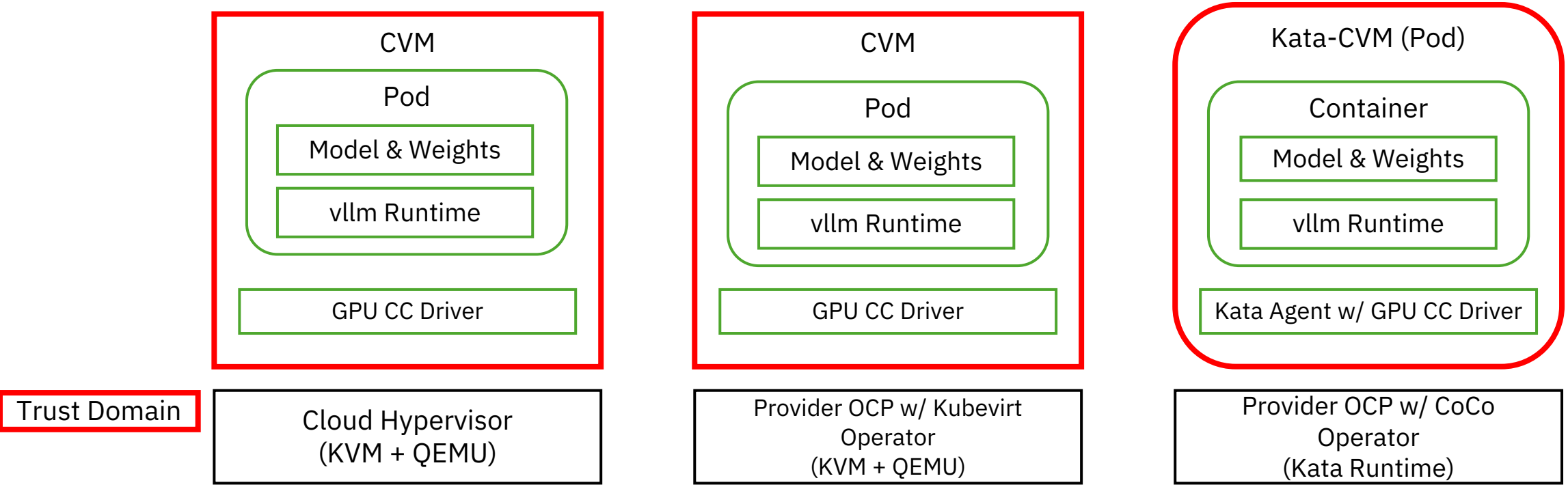


Image source – AMD SEV-SNP: Strengthening VM Isolation with Integrity Protection and More, January 2020, White Paper.

Cloud Based AI Serving Platforms with Confidential Computing

- There are multiple ways to deploy a confidential software stack.
- They're all leveraged confidential VM technology
- Device/Platform attestation is essential



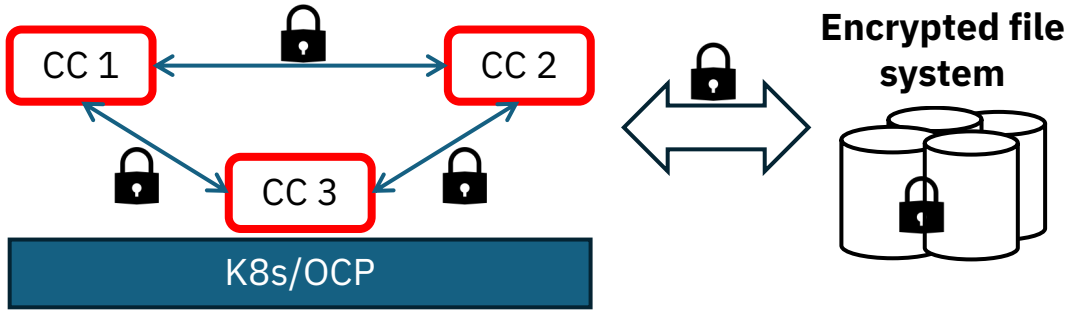
(1) Cloud Virtualization

(2) OCP/K8s Virtualization

(3) OCP/K8s + CoCo

Challenges of Running Scalable Distributed Applications with Confidential Computing Technology

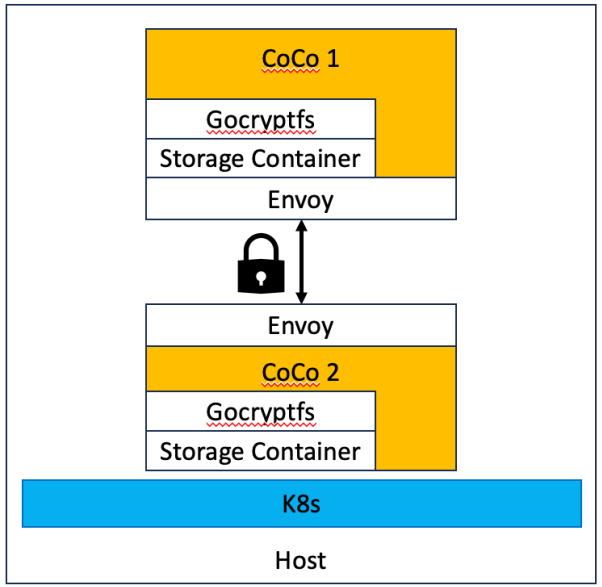
- **Security challenge**
  - Standard security solutions for storage and networks require host involvement, but **the host is not trusted in the CC framework**
  - Any sensitive data can *only* be decrypted inside of CC instances
- **Performance challenge**
  - What is the scalability of these security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



- **Implementation details**

CoCo + NFS + Gocryptfs + Envoy (i.e., CoCo - E2E)

- 1 control plane + 28 worker nodes (1736 cores and 56 pods in total)
- ***7.13%*** performance overhead in total
  - Experimental baseline: a k8s cluster running on bare-metal machines

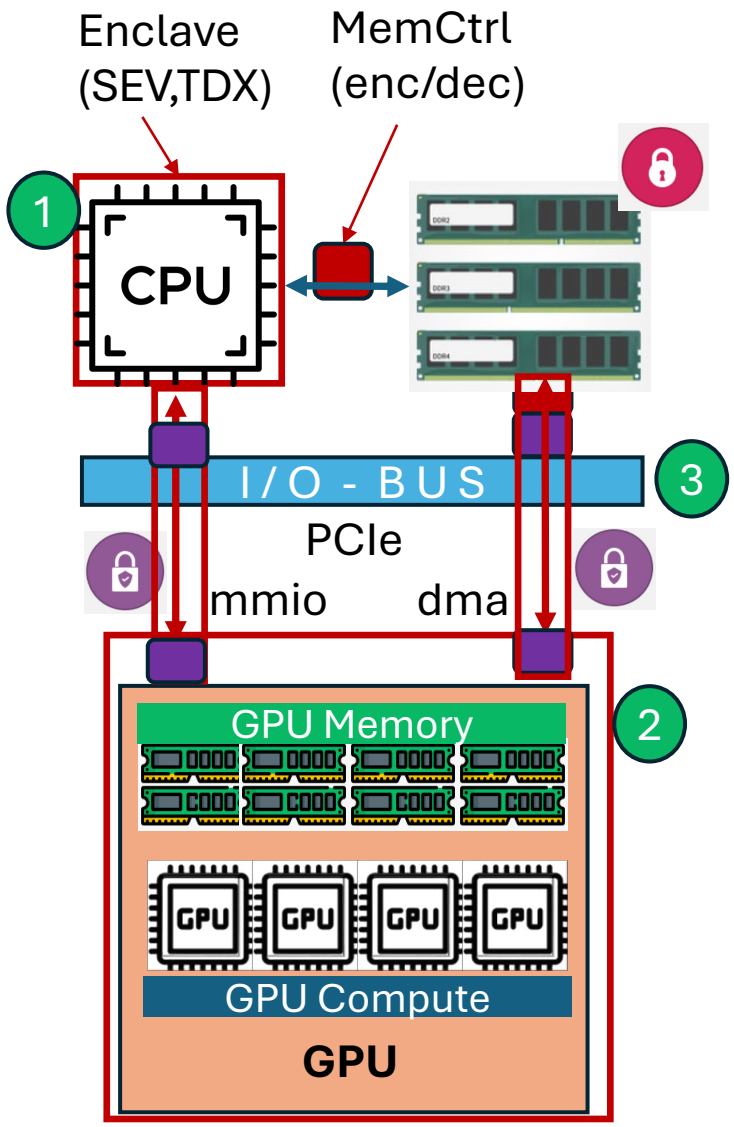


TEE accelerator integration with AI serving platform

- 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. Hopper 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

  - Blackwell devices support hardware encryption along with TEE-IO supported CPUs with advertised negligible overhead
    - Ref: https://www.nvidia.com/en-us/data-center/technologies/blackwell-architecture/



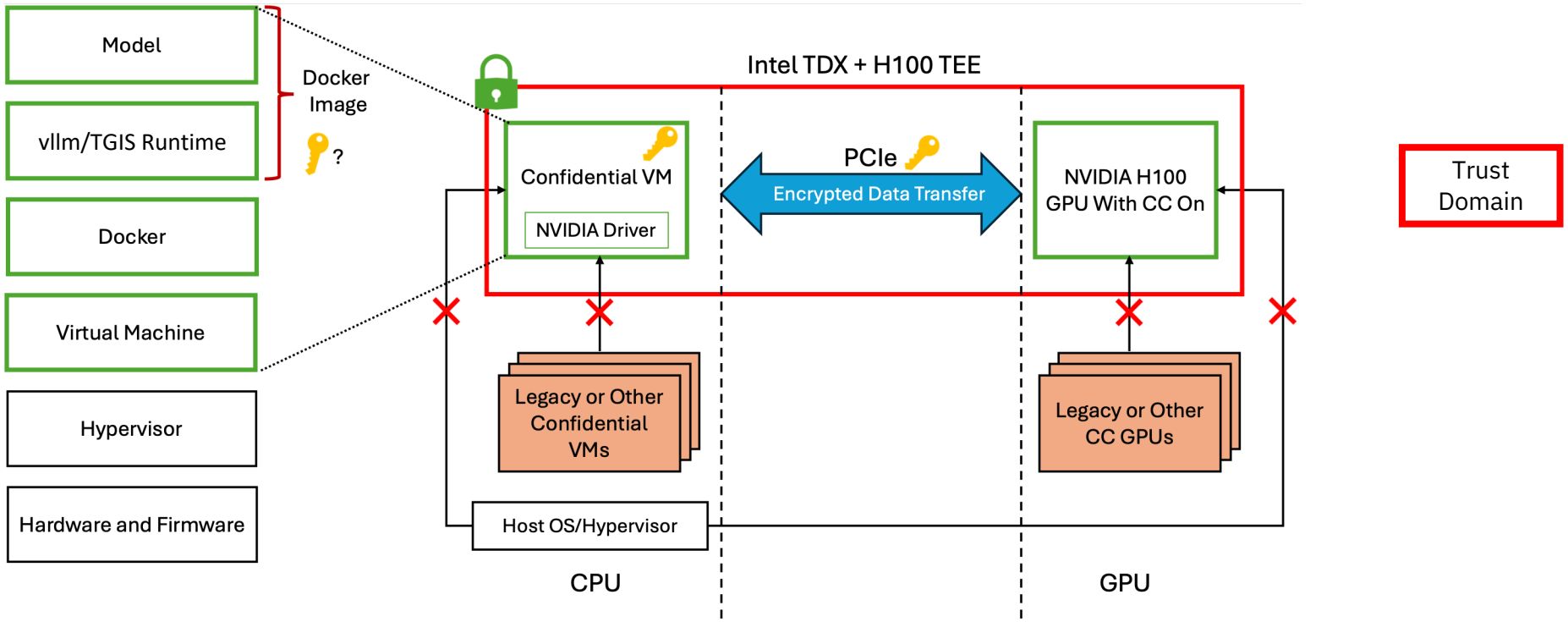
New technology to protect I/O:

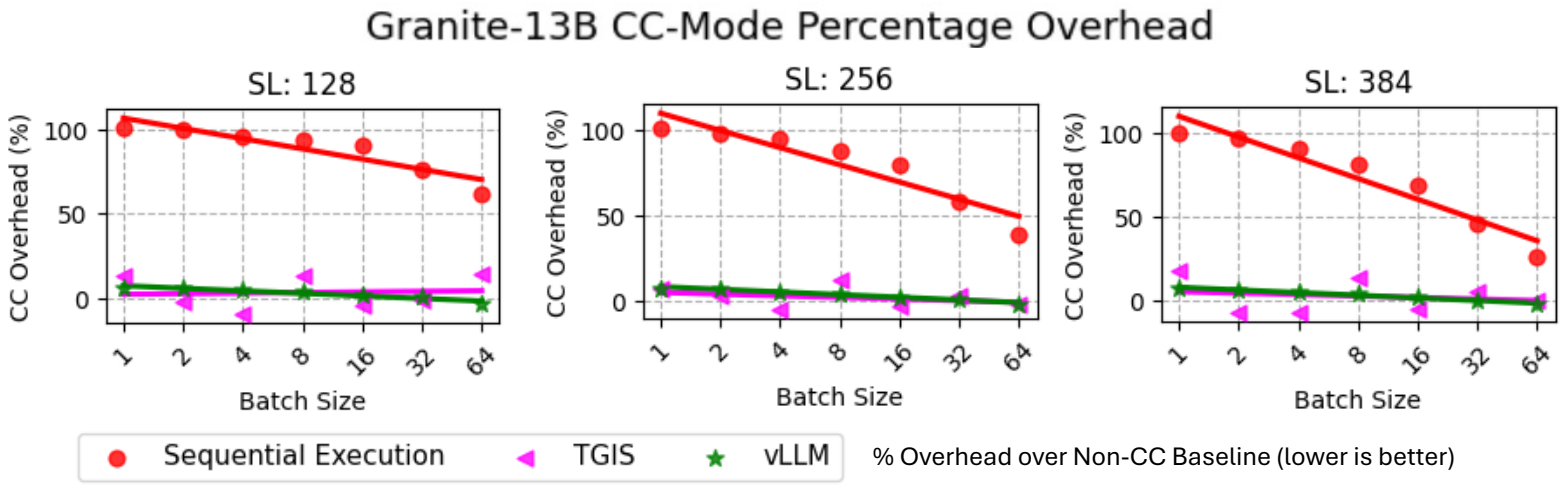
SPDM: link encryption

TDIO: dma access with mem key

TDISP (PCIe-5)

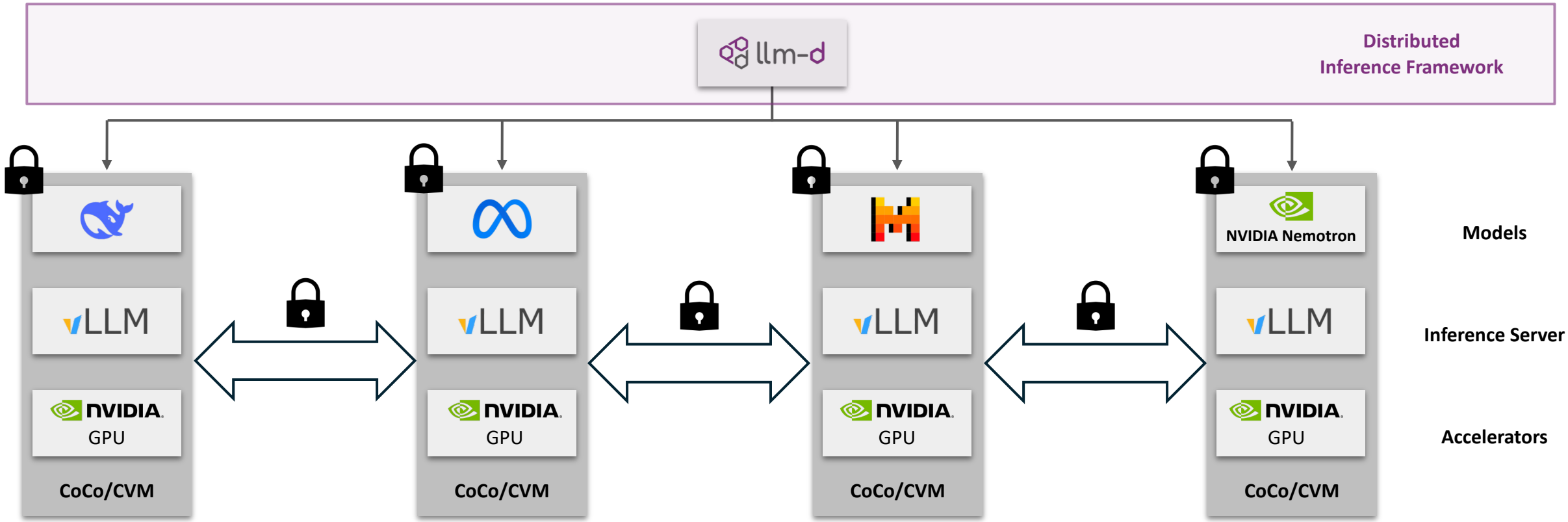
Compute-IO Overheads can be Hidden using vLLM and TGIS





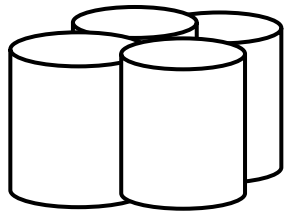
LLM runtimes like vLLM and TGIS, employ parallelization strategies (e.g., pipelined execution) which can significantly hide the overhead (due to encrypted PCIe link) as they enable "Compute-IO overlapping".

Confidential GenAI Inference Platform



- NVIDIA GPUs with single and multi-GPU CC mode
- KV cache must *not* be shared across CoCo/CVM with multiple tenants
- Secure storage: 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)
- Secure network: Generic Sidecar for mTLS encrypted network traffic through ISTIO's Envoy

Encrypted file system with AI models



Conclusion and Experiences

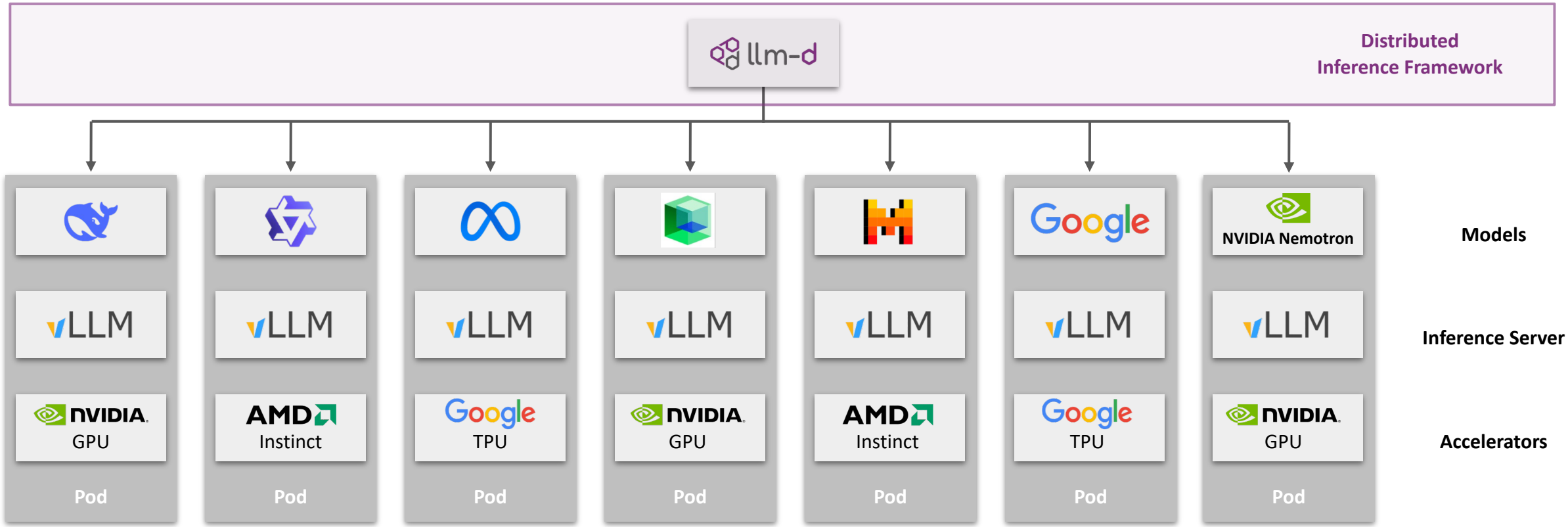
- Confidential AI can be achieved through today's technology
  - Low overhead of vllm with CPU-GPU confidential computing setup
  - Confidential-computing based llm-d framework with encrypted networks and storage
- Open questions:
  - Is isolation through firmware access control sufficient?
  - Do we need encryption on GPU HBM/memory similar to host?
  - Link encryption between GPUs is coming, which needs further evaluation
  - Confidential sharing GPUs
- The agentic stack is evolving -- both in terms of performance and functionality.
  - Yet confidentiality and integrity of agentic stack remain critical
  - Requirements:
    - Confidential computing protected stack with encrypted storage, encrypted network, and cc-enabled AI accelerators
    - Attestable stack with scalable key management

Conclusion and Experiences

- Check out our demos, blog posts, and publications
  - “The power of confidential containers on Red Hat OpenShift with NVIDIA GPUs.” Red Hat, October 2025.
  - “Advancing Confidential AI with Confidential Computing.” Phoenix Technologies, May 2025.
  - “Position Paper: From Confidential Computing to Zero Trust, Come Along for the (Bumpy?) Ride.” HASP-MICRO, November 2024.
  - “vLLM in Confidential CPU-GPU Enclaves: Does it Perform?” IEEE AICS, November 2024.
  - “Securing AI Inference in the Cloud: Is CPU-GPU Confidential Computing Ready?” IEEE CLOUD, July 2024.

Questions / Discussion ?

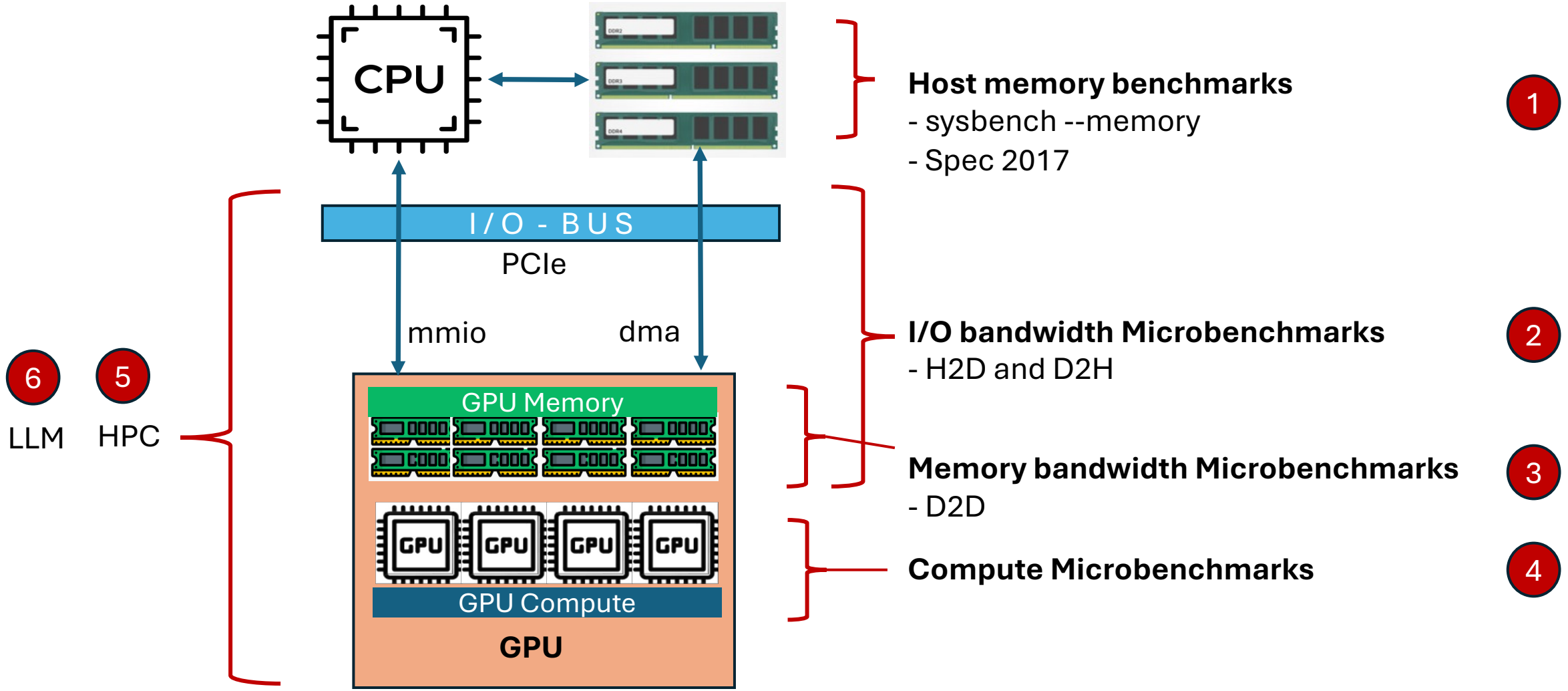
Enterprise GenAI Inference Platform



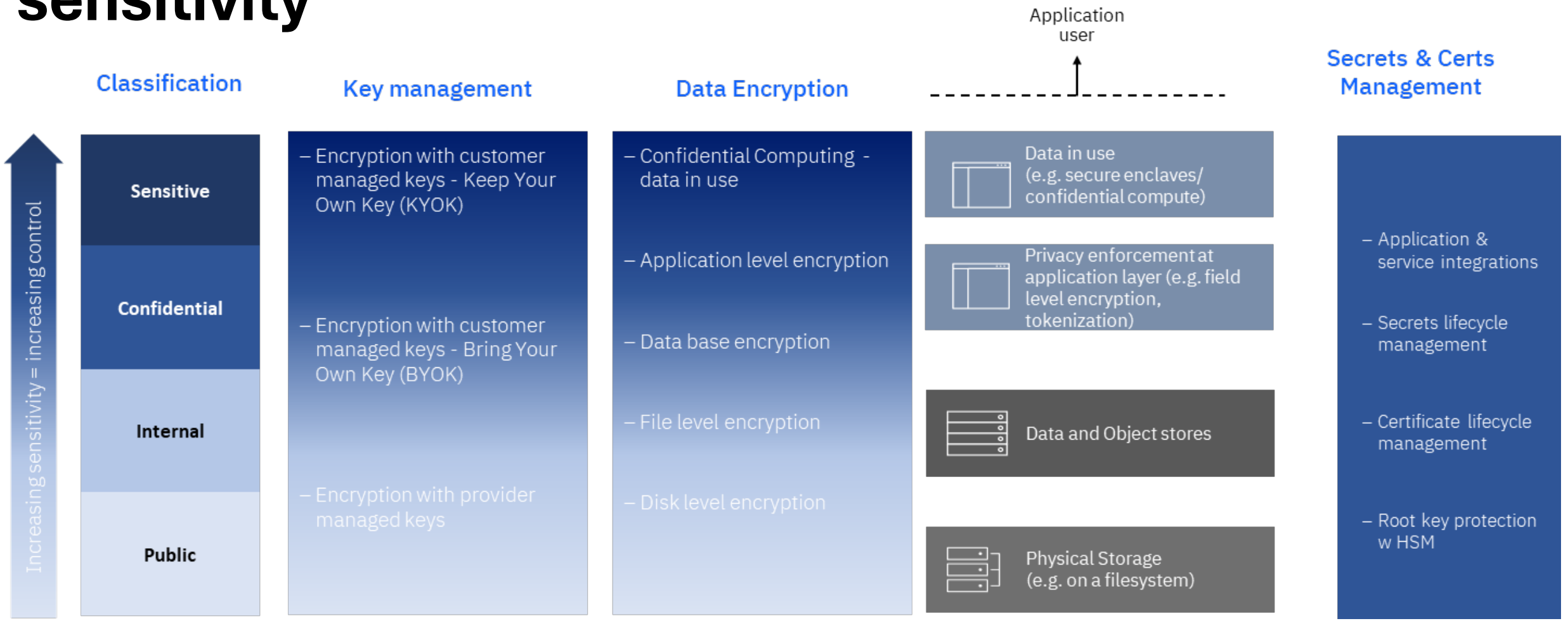
- llm-d: A distributed LLM serving platform for cloud-native enterprise generative AI and agentic AI systems
- Confidential Computing based llm-d can provide a trust-worthy, performant, efficient, and secure end-to-end GenAI without modifications to the stack.
- *But is it just running everything into enclaves?*

Performance Evaluation via Benchmarks in nonCC and CC mode

Mohan et al., “Securing AI Inference in the Cloud: Is CPU-GPU Confidential Computing Ready?” IEEE CLOUD, July 2024.



Data encryption (across at-rest, in-transit, in-use) and key management controls could vary based on data sensitivity



EDA with CoCo + Secure Storage + Secure Network

Ye et al. “Position Paper: From Confidential Computing to Zero Trust, Come Along for the (Bumpy?) Ride.” HASP, November 2024.

- Siemens Calibre® EDA (Electronic Design Automation)
  - “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
- Secure storage: 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)
- Secure network: Generic Sidecar for mTLS encrypted network traffic through ISTIO’s Envoy
- CoCo + NFS + Gocryptfs + Envoy (i.e., CoCo - E2E)
  - 1 control plane + 28 worker nodes (1736 cores and 56 pods in total)
  - ***7.13%*** performance overhead in total
    - Experimental baseline: a k8s cluster running on bare-metal machines