Samsung Research

From Pixels To Agents: Optimizing On-Device Performance of Confidential Computing in AI Evolution

Savas Ozkan – Engineering Manager & AI Tech Lead
Bokdeuk Jeong – Principle Engineer

Contributor: Sinan Mutlu, Mete Ozay

23/06/2026

Contents

Key Takeaways!...

- **On-device AI is booming …**
- **Agents are already here …**
- **Privacy is never free …**

01. Introduction & Motivation

AI Privacy

- **AI is booming**, specifically, with many on-device applications:
  - However, privacy guarantees have not kept pace.
- Modern AI systems process the most sensitive data — health, finance, identity, conversations —:
  - Yet, most deployments offer no hardware-level privacy guarantee for the inference step.

Cloud Inference

- User data must be transmitted to servers for inference.
- Model provider can inspect user inputs at any point.
- Encrypted in transit does not protect at inference time.

On-Device Without CC

- Privileged OS can read model weights and activations.
- Physical memory attacks remain possible.
- Malicious software can intercept inference results.

Differential Privacy

- Adds calibrated noise to model to prevent memorization.
- Accuracy degrades by up to 30% at strong privacy budgets.
- Privacy-utility trade-off is task and data specific.

** **Key insight:** a hardware-enforced isolation is needed that protects data **AND** the model — without sacrificing inference accuracy or usability, **but not without cost**.

Industry Moving Towards On-Device AI, Agents and Confidential AI

- AI and hardware companies are exploring **a device optimized for on-device AI tasks** [1].
  - Faster local-inference, better efficiency, agent-like capabilities.

- **Hybrid solution** with local models [3].
  - Combining local intelligence with cloud capabilities without full cloud dependency.

- **EU AI Act Regulatory [4]:** Tightening obligations around high-risk AI systems and sensitive data processing, pushing organizations towards hardware-attested systems.

- The shift **from cloud to on-device AI** is accelerating [2].
  - AI is moving off remove servers and onto personal hardware.
- **Raw AI power for on-devices** is here.
  - 1 Petaflop of AI performance.
  - 128 GB of unified memory
- **Privacy and security** are driving the adaptation.
  - Intelligently route queries to local models.
  - Disguise personal information sent to cloud.

- On-device Agentic [5].
  - Agentic model that **run fully on-device.**

[1] https://www.reuters.com/world/china/qualcomm-surges-report-openai-tie-up-ai-smartphone-processors-2026-04-27/
[2] NVIDIA and Microsoft Reinvent Windows PCs for the Age of Personal AI | NVIDIA Newsroom.
[3] https://medium.com/@denisov.shureg/hybrid-ai-in-flutter-routing-between-on-device-and-cloud-models-954da8f25373
[4] https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai
[5] https://developers.googleblog.com/bring-state-of-the-art-agentic-skills-to-the-edge-with-gemma-4/

On-device Confidential AI

- CC guarantees for AI is that:
  - **Memory Encryption:** All data inside the confidential VM — inputs, model weights, activations — is encrypted in DRAM.
  - **Execution Isolation:** Neither hypervisor, OS, nor RMM process can observe intermediate computation states.
  - **Model Integrity:** Model weights loaded into confidential VM cannot be tampered with by the host after attestation.
  - **Input Privacy:** User prompts and data never leave the confidential VM in plaintext.

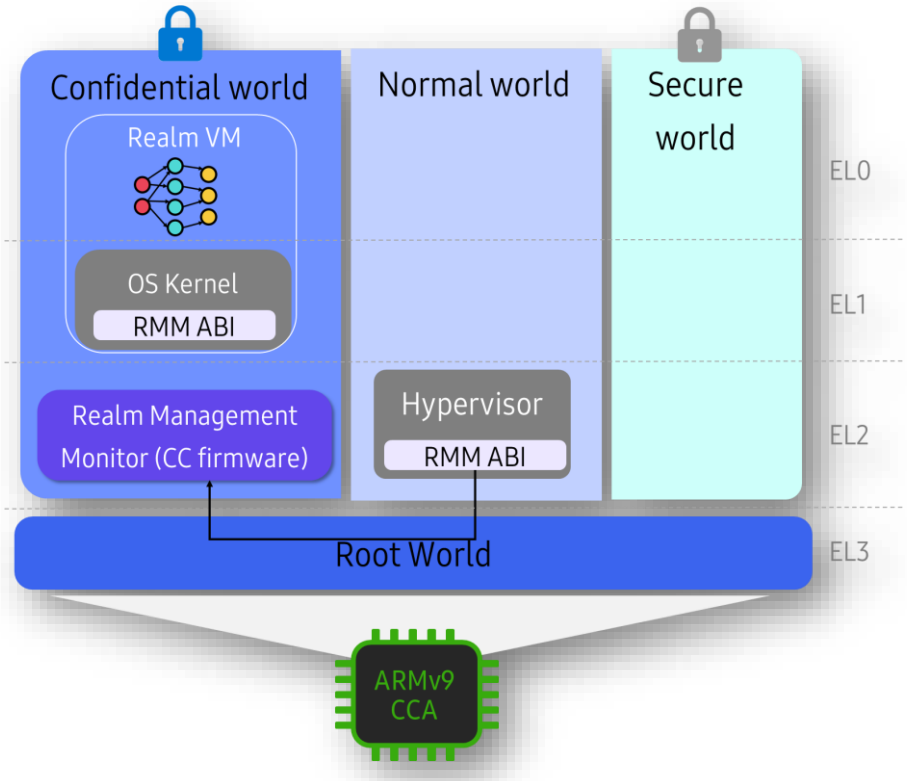


Illustration of Arm's CCA

- On-device CC provides additional advantages:
  - **Privacy by Architecture:** User data never leaves the device.
  - **Data Sovereignty:** User retains full ownership of data.
  - **Reduced Attack Surface:** No centralized server processing everyone's data.

01. Introduction & Motivation

On-device Confidential AI

- In summary, CC encrypts memory and isolates execution, but **privacy is never free**.
- Obviously, deploying AI inside a hardware-backed trusted execution environment (TEE) for edge-devices introduces multi-layered technical obstacles:

Performance*

- **Inference & Loading Overhead:** Latency increases significantly for model loading.
- **Resource-Constrained Hardware:** Model section constrained to small quantized variants.

Communication*

- **Message Parsing:** Communication overhead leads to actual computation cost.

Ecosystem+

- **Software Compatibility:** High engineering effort to port and validate frameworks.
- **Privacy-Accuracy Trade-off & Attention:** Security guarantees are important.

*: Topics for today's talk.
+: Samsung's Islet product (QEMU) is used for the tests/works in this talk: https://github.com/islet-project/islet

02. On-device Confidential AI

Observed Limitations from AI perspective

- Four critical categories must be recognized in the performance optimization of on-device confidential AI.

1) Inference/Execution
- **Observation:** Up to 10% overhead.
- **Solution:** Unfortunately, unavoidable 

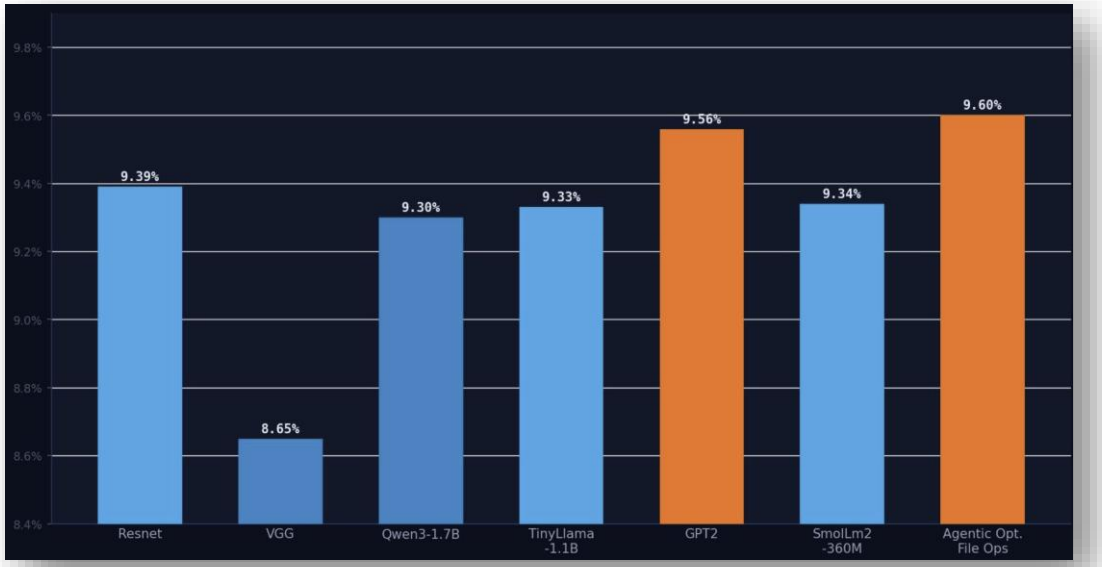
2) Model Loading Time
- **Observation:** Up to 90% overhead.
- **Solution:** Pre-mapping AI models to confidential VM during the system setup phase.

3) Model Size
- **Observation:** max. 12-16 GB DRAM capacity.
- **Solution:**
  - Extensively quantized AI models, or
  - Adapting large AI models' task capabilities to small AI models.

4) Message Parsing
- **Observation:** Up to 10x overhead.
- **Solution:** A protocol-level solution can reduce this overhead.

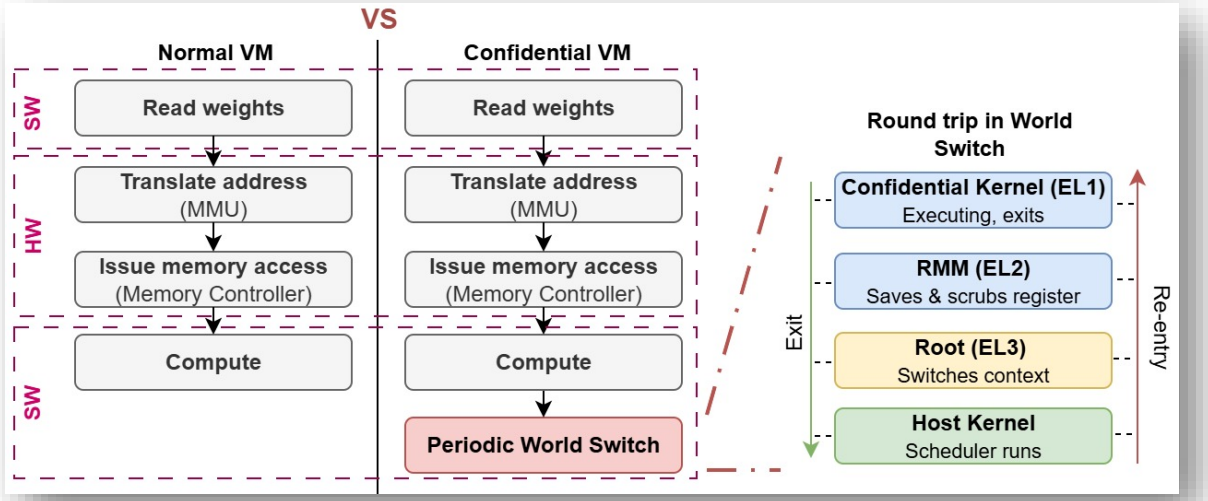
Inference/Execution

- As we said, privacy is never free – a trade-off between utilization and accuracy.
  - **Op1:** Differential privacy does not yield a slower response, but the accuracy drops appx. 30% [1, 2].
  - **Op2:** CC does not cause an accuracy drop, but the inference/execution time is slower by 10%.



AI overheads inside a confidential VM. *Models are quantized to either Q4 or Q8.

- For various models/opts, similar overheads are expected.



Visualization of why 10% overhead is introduced in confidential VM.

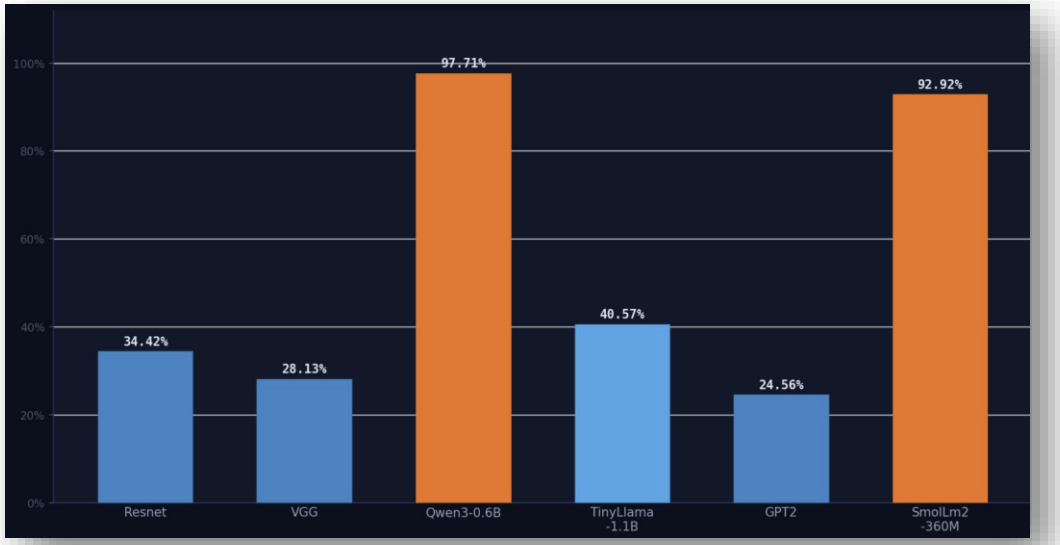
- Confidential VM does not have a separate CPU.
- Context switch is necessary, but expensive.

Unavoidable tax to pay…

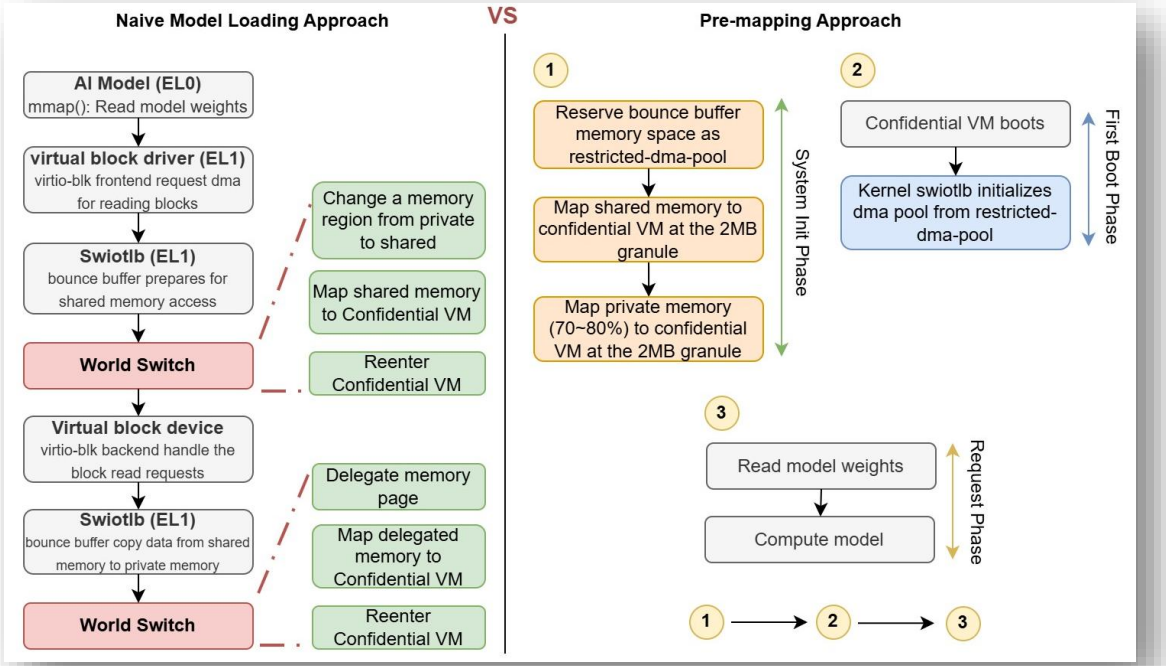
[1] A Study of Improving The Privacy-Utility Trade-off of Task-specific Models with Learnable Privacy, SRUK paper.
[2] Dp-dylora: Fine-tuning transformer-based models on-device under differentially private federated learning using dynamic low-rank adaptation , SRUK paper.

Model Loading Time

- Loading an AI model into a confidential VM requires executing multiple repetitive operations, including RM calls between the host hypervisor and RMM:
  - This is designed for privacy architecture, but not speed: Up to 90% overhead.
  - **Solution:** Pre-mapping AI model to confidential VM during the system setup phase.
    - Constraints: A fixed and single model whose parameter size is known.
    - The overhead can be reduced by 80%.



AI overheads inside a confidential VM.



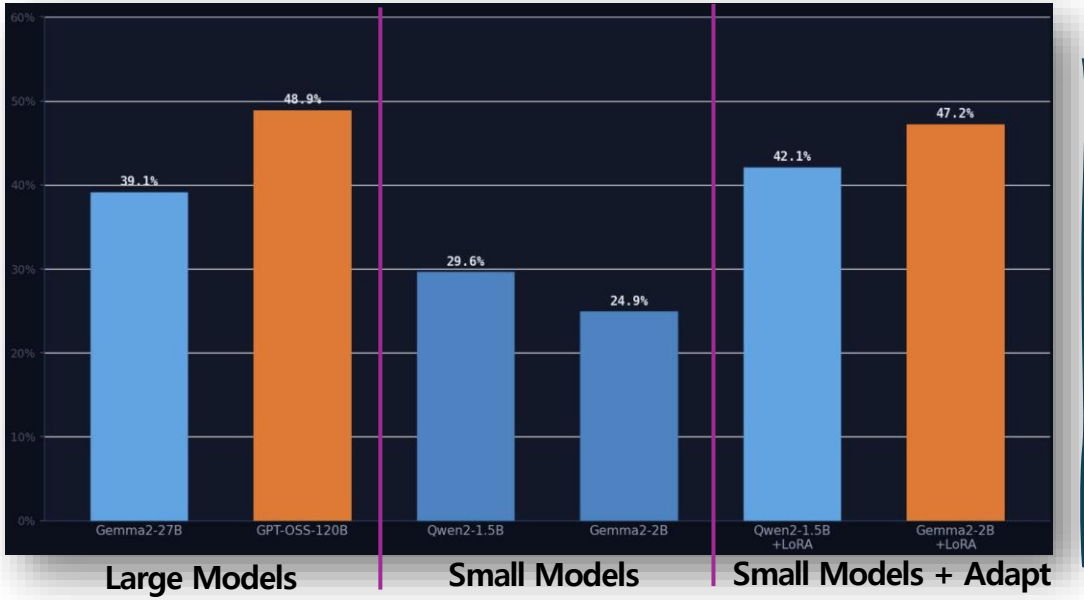
Visualization of pre-mapping approach

02. On-Device Confidential AI

Model Size

OS and essential services	~2-3GB
CCA related metadata and footprint	~200MB
Other applications	~2-3GB
Remaining for AI model in CC	~2-3GB

- Confidential VMs are limited for on-device applications.
- For a scenario where a device has 8GB DRAM capacity:
- This is why on-device confidential AI today is constrained to small models:
  - Their performance is not realistic with weaker task accuracy.
  - **Solution:** Transfer task capability from large models to small ones using adaptors [1, 2].



Task accuracy vs model size

Future direction for Adaptors in CC:

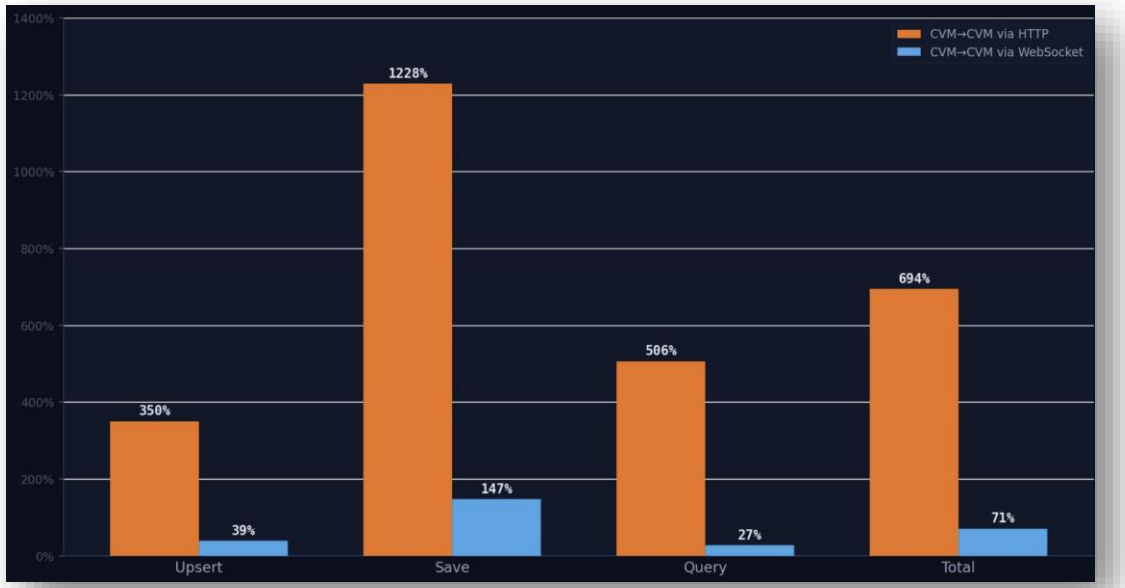
- Current workload is monolithic:
  One fixed model with one adapter.
- Future workload is adaptive*:
  One model with multiple capabilities (multi-Adaptors).
  - Adapters are sealed to VM's identity, treating as IPs.
  - Base model can be public, but adapters are confidential.

[1] MemLoRA: Distilling Expert Adapters for On-Device Memory Systems, SRUK paper.
[2] DuoMem: Towards Capable On-Device Memory Agents via Dual-Space Distillation, SRUK paper.

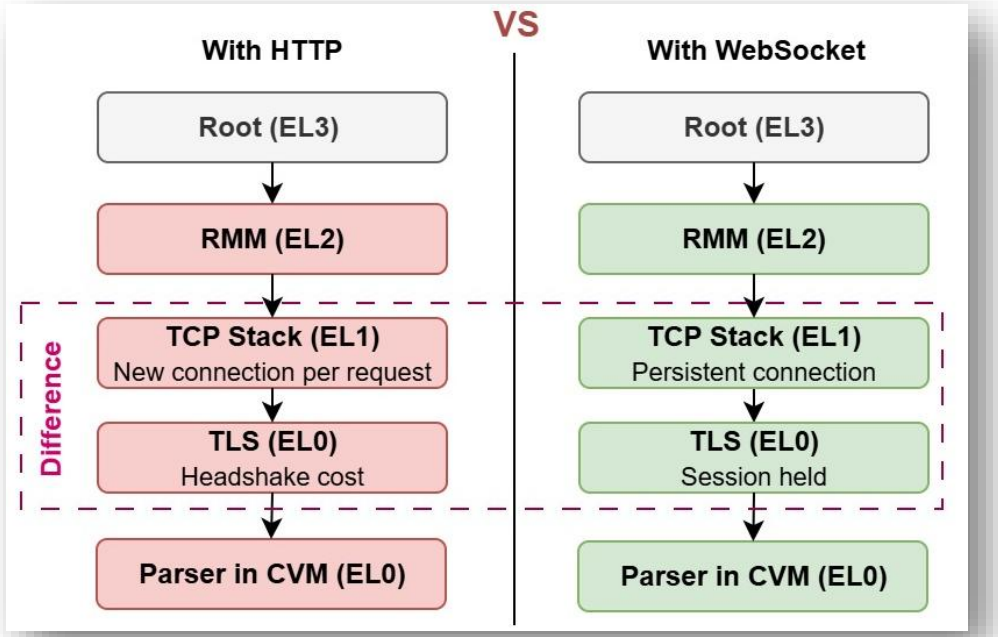
*: Needs contribution to Ecosystem.

Message Parsing

- Message parsing inside a confidential VM is slower than in a normal environment:
  - In naïve way, every message crossing must pass through mandatory steps in CC – world switch, checks, delegation –. Introduces Up to 10x overhead.
  - **Solution:** Establish connection once, keep communicate over the same channel:
    - Instead of multiple handshakes, the session is held opened.



Message parsing overheads compared to CVM>CVM.



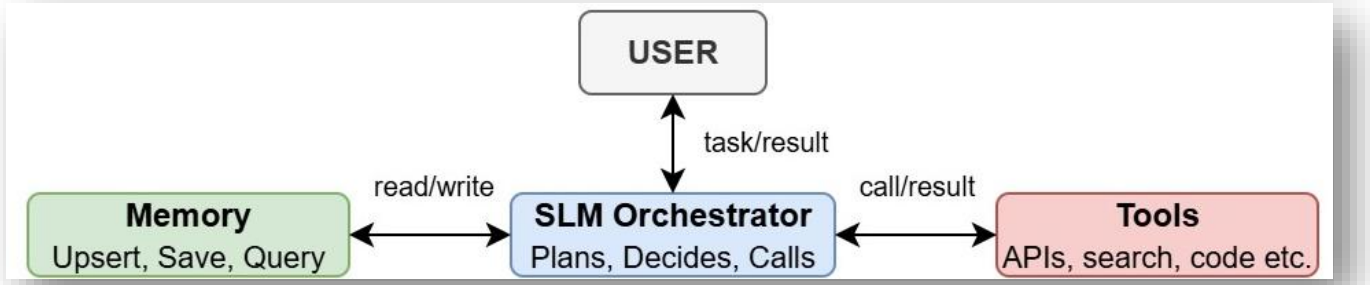
Visual comparison of message parsing using different protocols.

03. On-device Confidential Agentic AI

Agentic AI from privacy perspective

- **Agents are already here!...**
- In Agentic system, LMs sit at the centre and orchestrate everything:
  - This system mainly contains LMs, Agentic Tools and **Memory**.

⚠ Memory stores conversation and user history, learned facts and past outcomes.



A simple agentic system

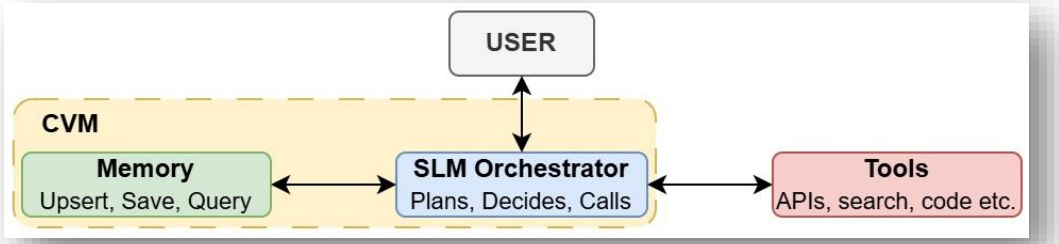
- **CC must be used to control the whole loop that runs on a device which you don't fully control**;
  - While keeping the data and model protected from the device itself.

- **For Memory:** CC keeps the memory usable but unreadable to everything outside the VM.
- **For Models:** CC lets the model run on data that it does not own while staying protected from the host.

Agentic AI from privacy perspective

- Of course, there can be many other designs for Agentic systems in CC.

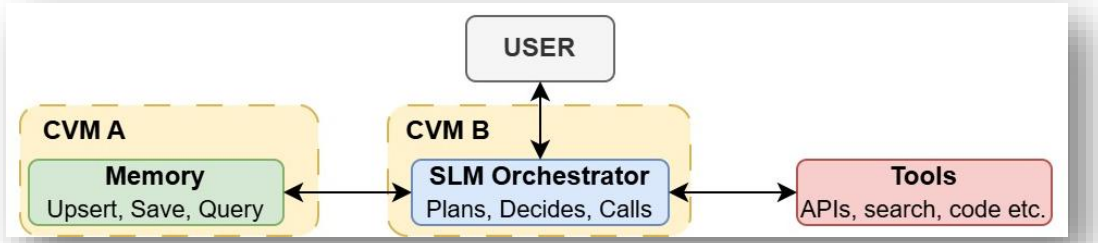
One-CVM Design:



One-CVM Agentic System

- **Pros:**
  - Host cannot observe either SLM or memory.
  - Simpler and lower-overheads.
- **Cons:**
  - No isolation between SLM and memory: sensitive to prompt injection, jailbreak and other types of attacks.

Two-CVMs Design:

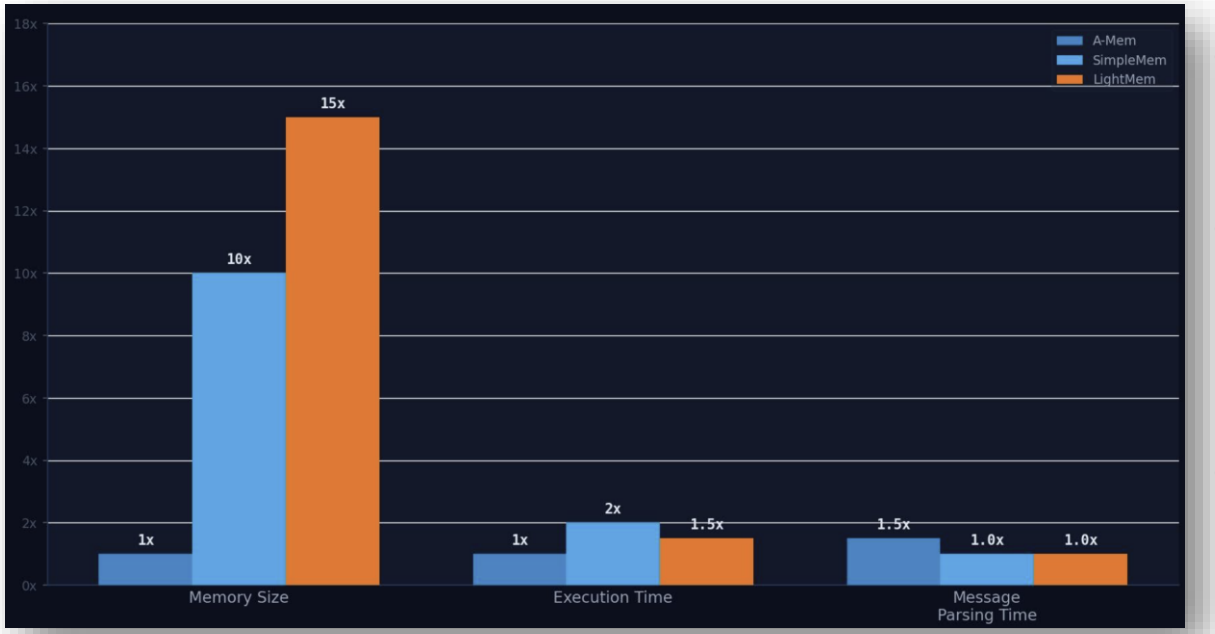


Two-CVM Agentic System

- **Pros:**
  - Host cannot observe either SLM or memory.
  - Isolation between SLM and memory.
- **Cons:**
  - Higher-overheads due to the necessary communications.

Agentic AI from privacy perspective

- Also, memory types significantly effect the performance depending on different use-cases.
  - **A-MEM:** High recall quality, but multiple LM calls for operations.
  - **SimpleMem:** Compact responses with less crossing, but slow offline compression.
  - **LightMem:** Low-online overhead, but expensive organizing cost.



Comparison of memory types for different items.

- For **one-CVM** or **two-CVMs** design, different memory might be more effective:
- For **one-CVM**: LM and memory are inside the same CVM, free read/write. A-Mem might be the best option.
- For **two-CVMs**: There will be crossing between CVM. LightMem might be the best option.

04. Final Remarks & Takeaways

04. Final Remarks & Takeaways

Key Takeaways

- **On-device AI is booming …**
  - Privacy cannot be an afterthought, we must build it in from the start.
- **Agents are already here …**
  - Powerful, yes, but without privacy protection, it can be dangerous.
- **Privacy is never free …**
  - There is always a cost, so choose deliberately, based on your needs.

Benchmark Framework

- Our work aims to advance the CC community:
  - To enable systematic evaluation, a software framework is developed using the open-source Islet CC project.
  - The framework supports configurable benchmarking of AI agent tools, and will be publicly released to enhance the reproducibility, collaboration, and the accelerate innovation within the CC community.
  - Project link (Will be active very soon):

https://github.com/savas-ozkan/armcc-for-agent

Samsung Research

Thank You!