Trusted Heterogeneous Disaggregated Architectures

Atsushi Koshiba, Felix Gust, Julian Pritzi, Anjo Vahldiek-Oberwagner, Nuno Santos, Pramod Bhatotia
Disaggregation in Data Centers

Paradigm shift from monolithic servers to **disaggregated architectures**

- **Monolithic servers**
  - Interconnect
  - CPU
  - GPU
  - SSD

- **Resource disaggregation**
  - CPU pool
  - GPU pool
  - FPGA pool
  - DRAM pool
  - SSD pool

Disaggregation improves utilization, scalability, & flexibility in heterogeneous data centers

*image: Flaticon.com*
Security Demands in the Cloud

AI-based intelligent services with confidential data

Trusted Execution Environment (TEE)

Trusted computing is indispensable for emerging cloud workloads

image: Flaticon.com
Research Gap

- Unfortunately, security for the disaggregated architectures is not well studied
  - Most existing TEE technologies are device/host-specific (e.g., Intel SGX)

- Existing TEE technologies are incompatible with disaggregated systems
  - User code/data across a distributed set of heterogeneous devices

Challenging to establish secure isolation on disaggregated architectures
Problem Statement

How do we build trusted disaggregated heterogeneous architectures without losing flexibility and elasticity?
Proposal: A Trustworthy Disaggregated Architecture

- HW/SW co-design that offers a **virtual TEE (vTEE) abstraction**
  - A secure and customizable isolated domain over disaggregated resources
Threat Model

- **Adversaries**
  - Administrators (full access to HW & network)
  - Multi-tenancy (other vTEE users)

- **Untrusted domain**
  - Network
  - Software running in vTEEs
  - Memory/storage pools of WEs/DEs

To ensure confidentiality and integrity of data and code in trusted domains
Overview

vTEE A

App

CPU process

FPGA process

μ-kernel

Secure HW

vTEE B

App

FPGA

GPU process

μ-kernel

Secure HW

Memory space

 MEMORY

μ-kernel

Secure HW

secure p2p network

secure p2p network

image: Flaticon.com
Design Challenges

1. Heterogeneity of disaggregated devices
   ○ Harmonizing device-specific TEEs is complex

2. Data distribution through the untrusted network
   ○ Compromise data confidentiality & integrity

3. Secure domain isolation across disaggregated components
   ○ TEE configuration changes according to users’ requirements
Key Ideas

1. Unified trusted hardware modules
   ○ All devices have the same security properties

2. Distributed computing by a distributed microkernel-based OS
   ○ Securely bridging authorized WEs and DEs

3. vTEE initialization & mutual attestation
   ○ Establishing trust among all the WEs involved by a vTEE
Secure Hardware Modules (1/2)

Worker Isolation Unit (WIU)

- Secure Controller
  - Hardware root-of-trust
  - Runs the microkernel OS
- Communication Module
  - Secure P2P connections
- Local MMU
  - Cache data of remote DEs into the local memory
Secure Hardware Modules (2/2)

Data Isolation Unit (DIU)

- Secure Controller
  - Hardware root-of-trust
  - Runs the microkernel OS
- Communication Module
  - Secure P2P connections
- Remote MMU
  - Memory management for remote Processing Elements
Trustworthy Disaggregated OS (TDOS)

- Microkernel-based OS
  - **Capabilities** for accessibility control to disaggregated components

- Unified vTEE management
  - Trusted computing  – **attestation**
  - Cryptography  – encryption, signatures
  - Communication  – secure P2P network connection
vTEE Initialization & Attestation

- Mutual attestation proposed by MAGE [USENIX SEC’22]
  - Establish *mutual trust* among multiple enclaves (WIUs)

![Diagram of vTEE Initialization & Attestation](image: Flaticon.com)
Open Discussion Points

- Can we dynamically resize vTEEs, i.e., change the number of WEs?
- How do we verify the attestation protocol among disaggregated devices?
- How do we ensure application compatibility with a Linux system?
Conclusion

**Problem:**
Challenging to build secure isolation environments on disaggregated architectures

**Proposal:**
vTEE: A secure and customizable isolated domain over disaggregated resources

**Solution:**
HW/SW co-design: Secure hardware extension (WIU&DIU) + Microkernel (TDOS)

Questions?