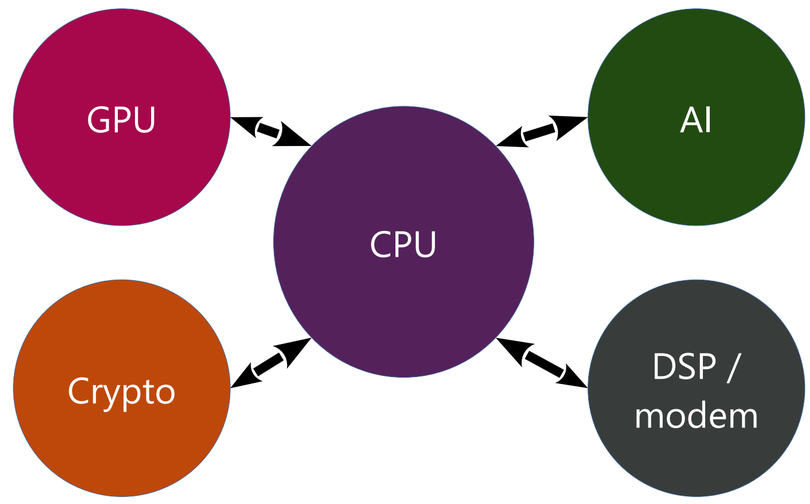
CAPcelerate: Capabilities for Heterogeneous Accelerators

Theo Markettos, Paul Metzger, Matthew Naylor, Timothy Jones
Department of Computer Science and Technology, University of Cambridge
theo.markettos@cl.cam.ac.uk

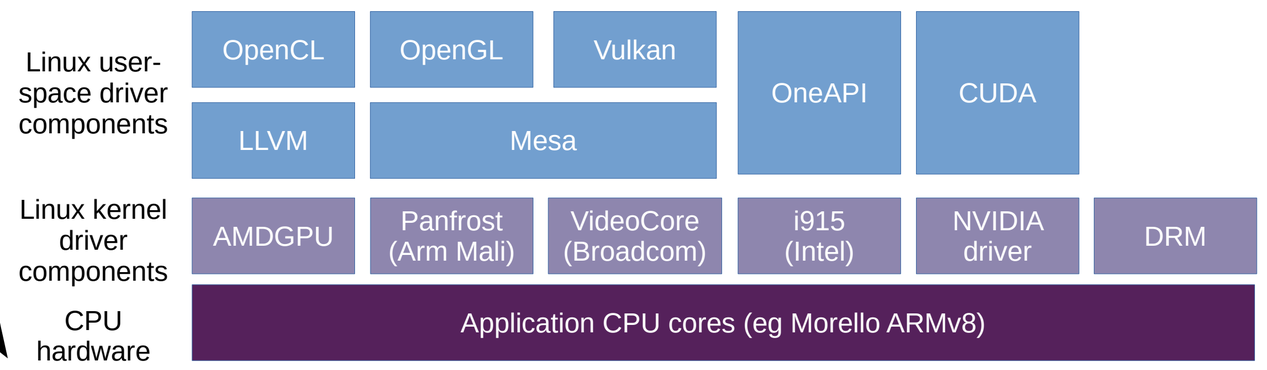
Modern systems have multiple *heterogenous accelerators*. Accelerators:

- **run code independently** of the CPU
- have **complex software stacks** spanning application, compiler, host OS kernel, firmware and accelerator compute



Accelerator software stacks are **poorly defended** from compromise and **a historic source of vulnerabilities**

Key question: **Can we use capabilities in accelerator hardware and software to improve security?**



Research direction:
Characterisation and tracing

- **GPUs** are popular accelerators with the **highly performance-critical hardware** and **rich software stacks** – good case study
- **Understand** how all **the GPU hardware/software pieces** fit together
- Many are **complex / proprietary / undocumented**
- **Understand** what **commercial accelerator hardware** actually **does**
- WIP: **Reason about its security models**
- WIP: **Reason about how capabilities can be integrated into the hardware/software models**, eg memory management

Research direction:
Capability-enabled 'GPU' hardware

- Q: Is it feasible to implement **capabilities in an accelerator** with **minimal impact on performance**?
- Q: Can we **efficiently enable multiple distrusting users** of the GPU hardware?
- DONE: **Built a GPU-like SIMT processor** on **FPGA** (2048 concurrent threads)
- WIP: **Added capability support** to the hardware
- WIP: **Exploring microarchitectural challenges** to make design suitable for capabilities with acceptable performance
- TODO: **Explore software stack** for such an accelerator

Research direction:
Accelerator software security

- **Key source of vulnerabilities is hardware drivers**
- **GPU drivers** are the **largest driver codebases** in existence
- Q: Can we **use capabilities** and **CHERI compartments** to **apply principle of least privilege to drivers**?
- Q: Can we **improve security** by moving **privileged code** from kernel driver into a **user-space CHERI compartment**?
- Q: What can we do to **improve security in the absence of a capability-enabled GPU**?
- WIP: **Compartmentalisation of the Panfrost kernel driver** for Arm Mali GPUs
- TODO: **Port to Morello hardware and its physical Mali GPU**
- TODO: **Explore implications** for **other parts** of the stack
- TODO: **Explore interaction when the hardware does understand capabilities**

UNIVERSITY OF CAMBRIDGE