ETH zürich

OpenCCA: An Open Framework to Enable Research on Arm CCA

https://opencca.github.io

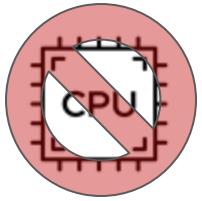
OC3'2026

Andrin Bertschi
PhD Student
Secure & Trustworthy Systems Group

Arm CCA Hardware, when?

Main Challenge on Arm CCA:

- No public Arm CCA hardware yet



Initial Rollout for Data Centers:

Locked-Down?
Flashable custom firmware?

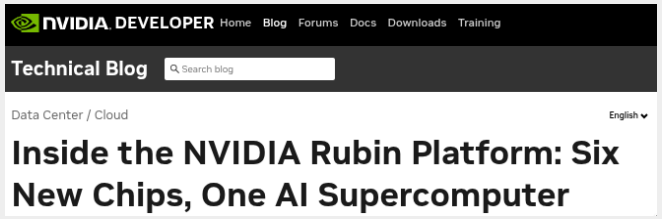
Documentation?
Technical reference manuals?

Affordable?
Enterprise only?

FUJITSU: Monaka (2027)

Microsoft: Cobalt 200 (2026)

Nvidia: Vera (2026)

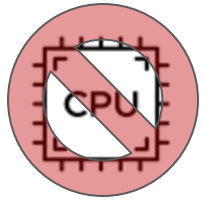


- https://global.fujitsu/en-global/technology/research/fujitsu-monaka
- https://techcommunity.microsoft.com/blog/azureinfrastructureblog/announcing-cobalt-200-azure%E2%80%99s-next-cloud-native-cpu/4469807
- https://developer.nvidia.com/blog/inside-the-nvidia-rubin-platform-six-new-chips-one-ai-supercomputer/

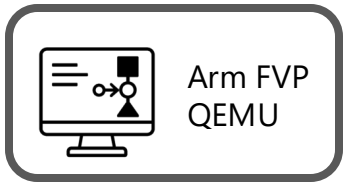
Arm CCA, today?!

Main Challenge on Arm CCA:

- No public Arm CCA hardware yet



1. Under Simulation



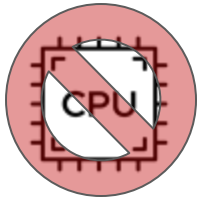
But **how fast?**
No **microarchitectural effects** of complex hardware

- ✓ Functionality
- ✓ Compatibility

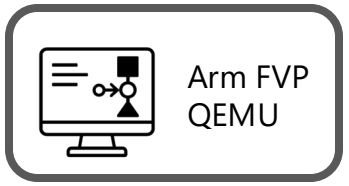
Arm CCA, today?!

Main Challenge on Arm CCA:

- No public Arm CCA hardware yet



1. Under Simulation



- ✓ Functionality
- ✓ Compatibility

2. Under Custom Prototype

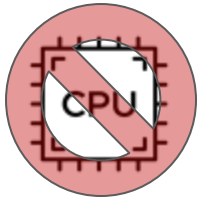
- ⛒ Closed Source
- ⛒ Difficult to compare
- ⛒ Difficult to reproduce
- ⛒ Repeated engineering

For Research:
Approximate research design
on real Armv8 Hardware

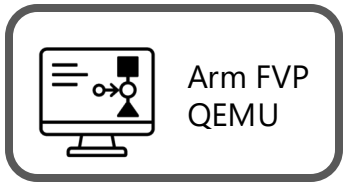
Arm CCA, today?!

Main Challenge on Arm CCA:

- No public Arm CCA hardware yet



1. Under Simulation



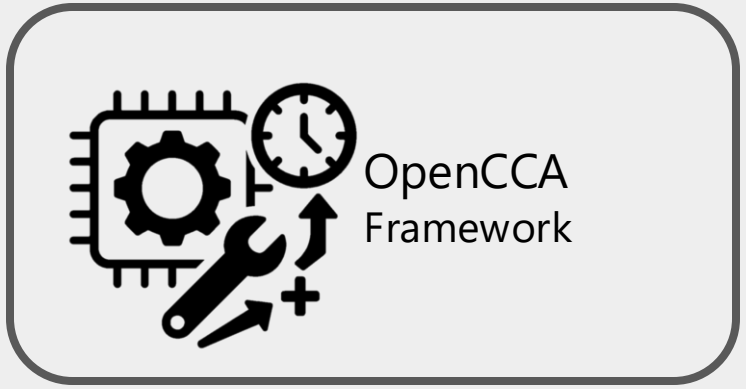
- ✓ Functionality
- ✓ Compatibility

2. Under Custom Prototype

- ⛒ Closed Source
- ⛒ Difficult to compare
- ⛒ Difficult to reproduce
- ⛒ Repeated engineering

The Need for Open Framework for Performance Evaluation

3. Under OpenCCA



OpenCCA Design Goals

Minimal changes to CCA reference stack → Preserve functionality

No security guarantees
Only for benchmarking & accelerator support

Target: **Affordable** & **Open** Armv8 Boards

Focus on ***reusable*** **Framework**
Not specific to a board
Performance estimation

Step 1: Simulation

Arm FVP
QEMU

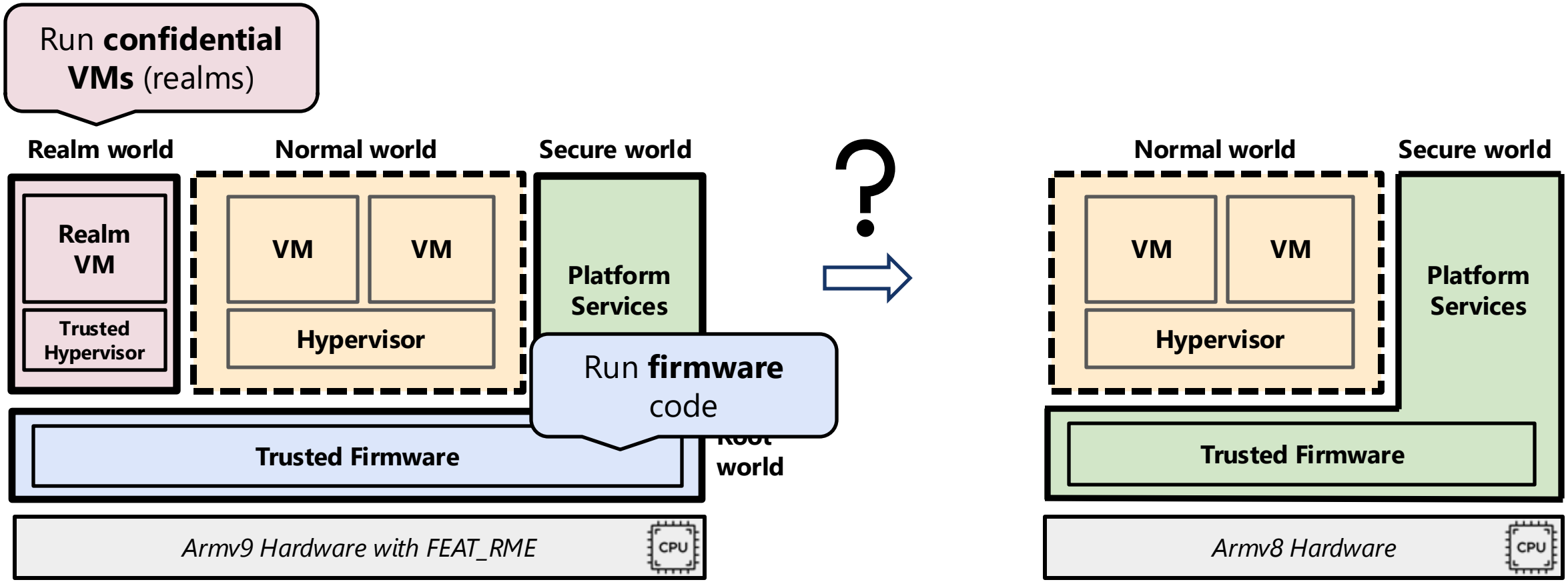
Validate design

Step 2: OpenCCA

OpenCCA
Framework

Approximate Performance & Interact with real HW

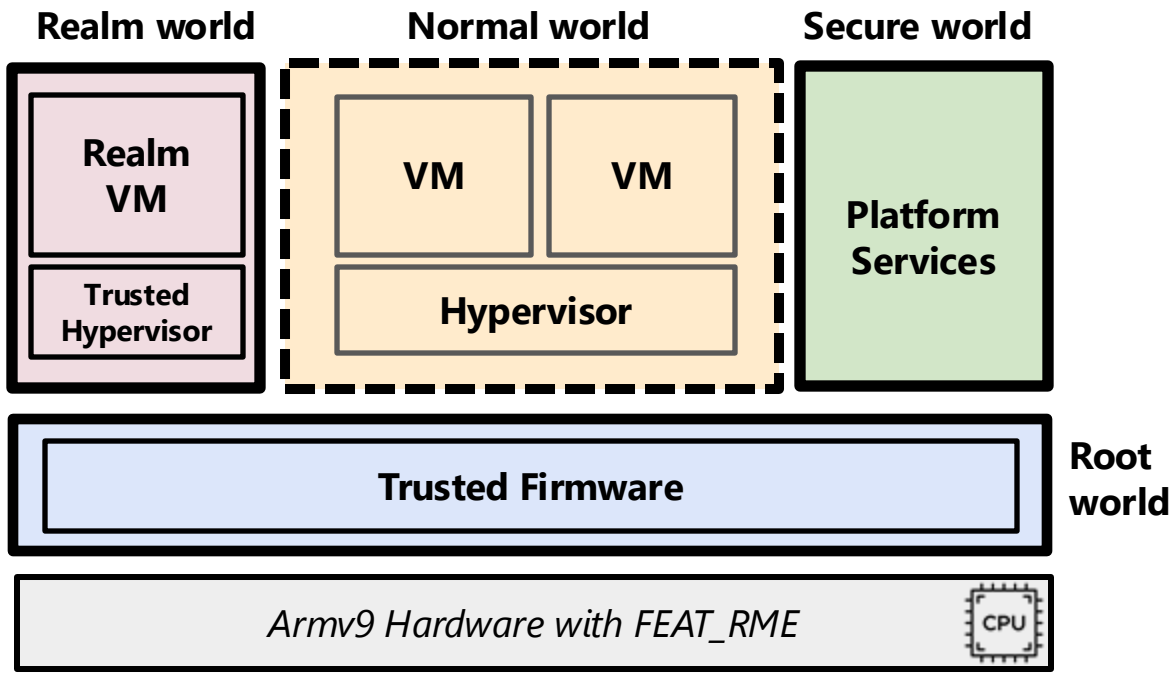
How do you even run Arm CCA on older Hardware?



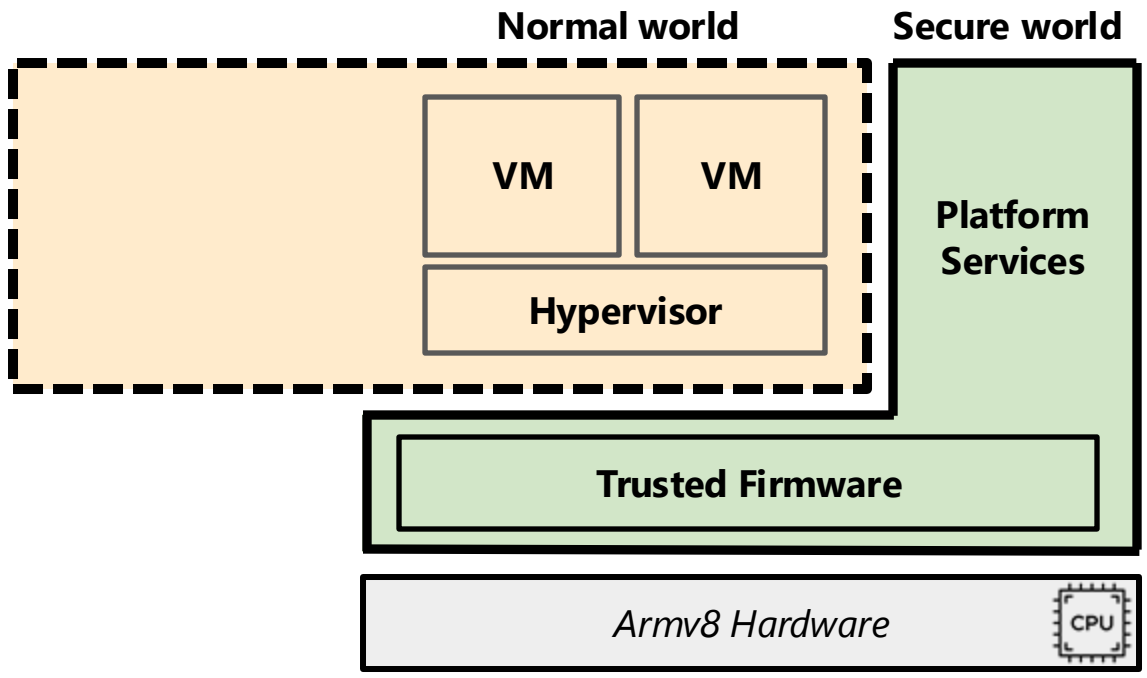
Armv9
(Arm CCA)

Armv8
(current
Hardware)

How do you even run Arm CCA on older Hardware?

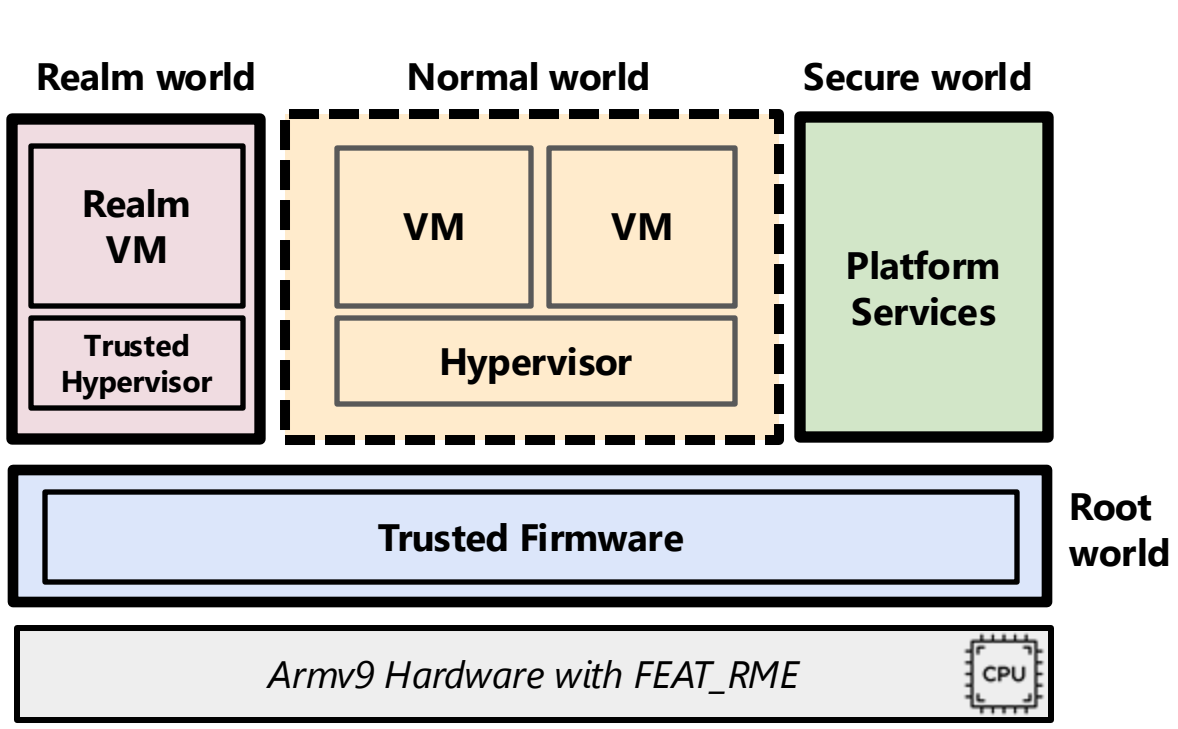


Armv9
(Arm CCA)

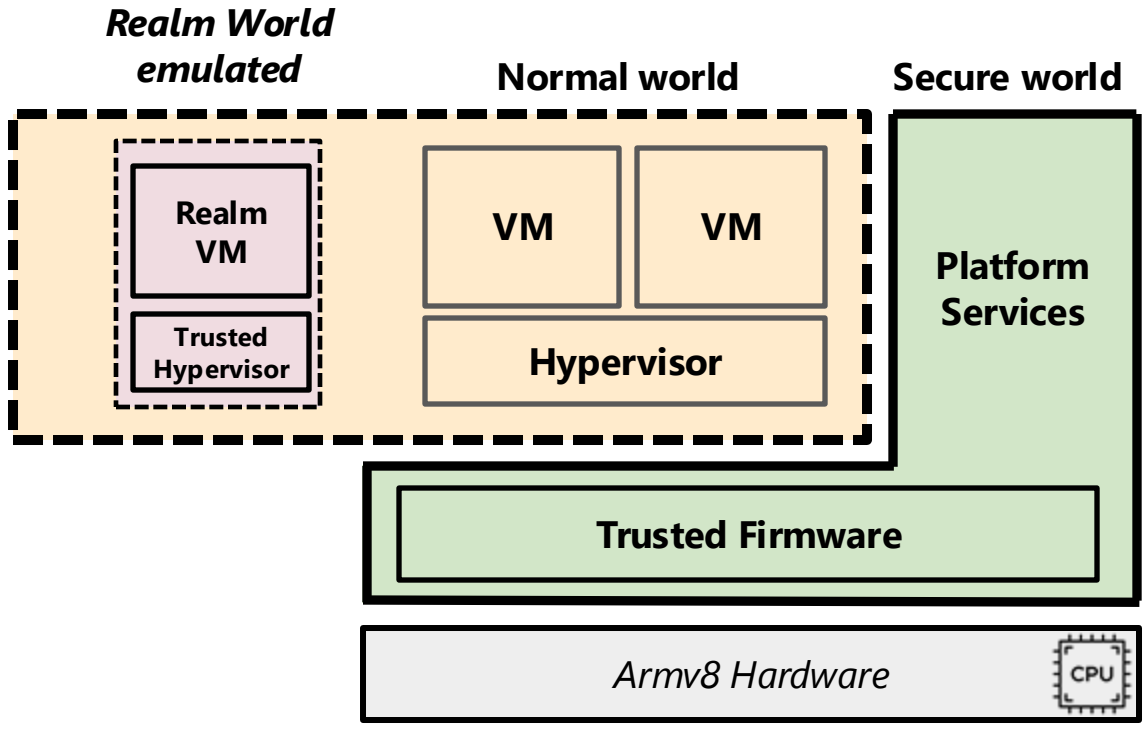


Armv8
(current
Hardware)

How do you even run Arm CCA on older Hardware?

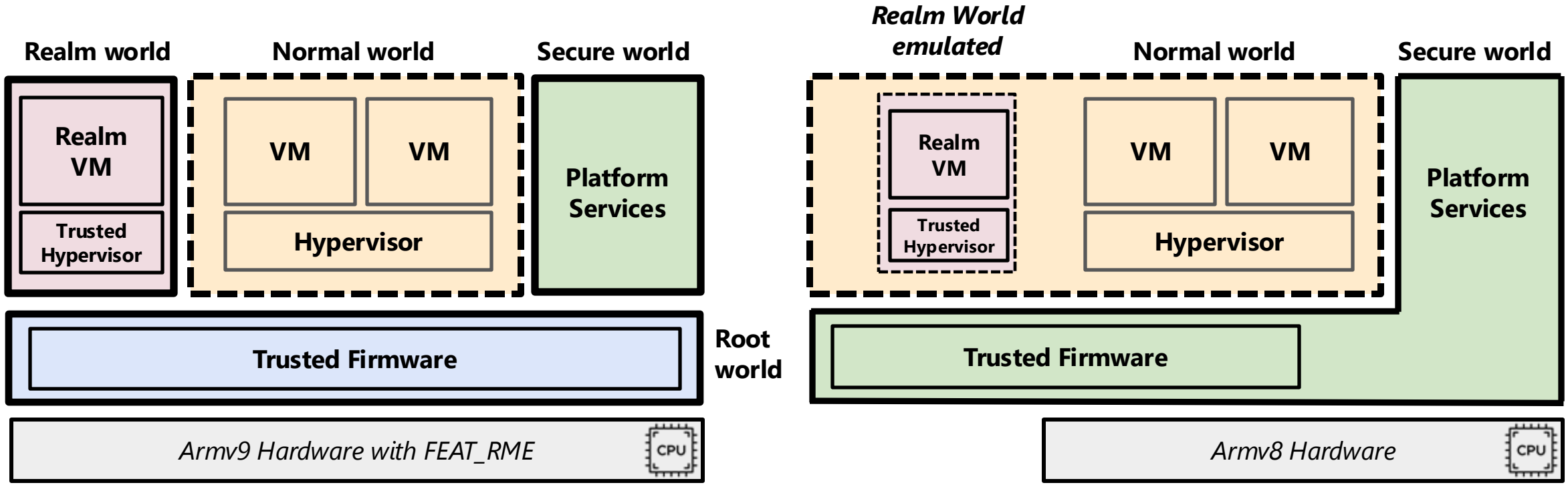


Armv9
(Arm CCA)



Armv8
(current
Hardware)

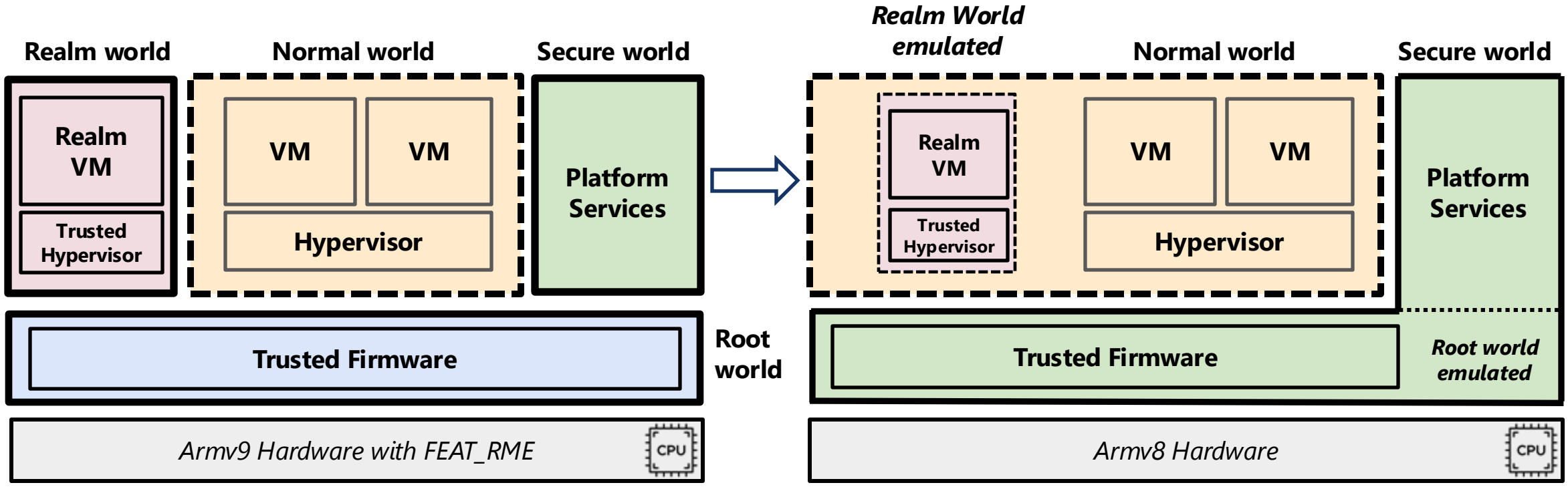
How do you even run Arm CCA on older Hardware?



Armv9
(Arm CCA)

Armv8
(current
Hardware)

How do you even run Arm CCA on older Hardware?

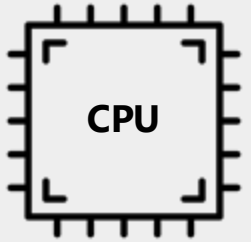


Emulate CCA in software within the constraints of Armv8 Hardware

Arm CCA on Armv8, what breaks?

See paper & website for technical details

- **Tradeoff** between **Compatibility** & **Overhead**
- **Fake missing parts in software** while keeping changes small
- Return **predefined values** instead of querying the hardware



Hardware Debugger valuable

Undefined Hardware Feature

CPU Stall or Instant Reset

Preserve functionality without security

Hardware:

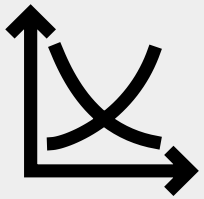
Key Specs: RK3588 SoC

- **Armv8.2** Architecture
- CPU: 4x **Cortex-A76** + 4x **Cortex A55**
- GPU: Arm Mali G610
- Up to 32 GB RAM
- I/O: PCIe 3.0, USB, HDMI

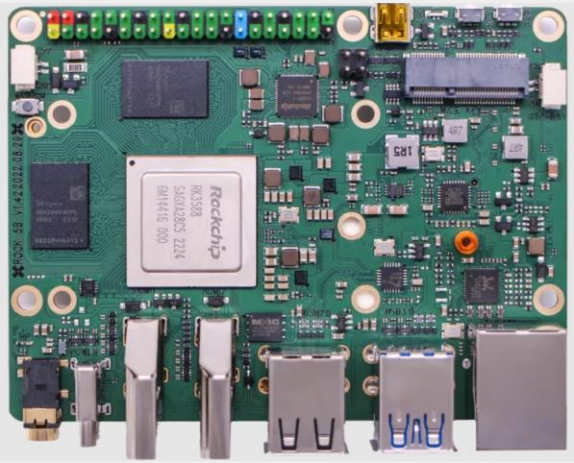
We looked into
~ 40 boards in 2025

No Vendor lock
Unlocked EL3

Documentation
Technical Reference Manual



Affordable +
Available + modern

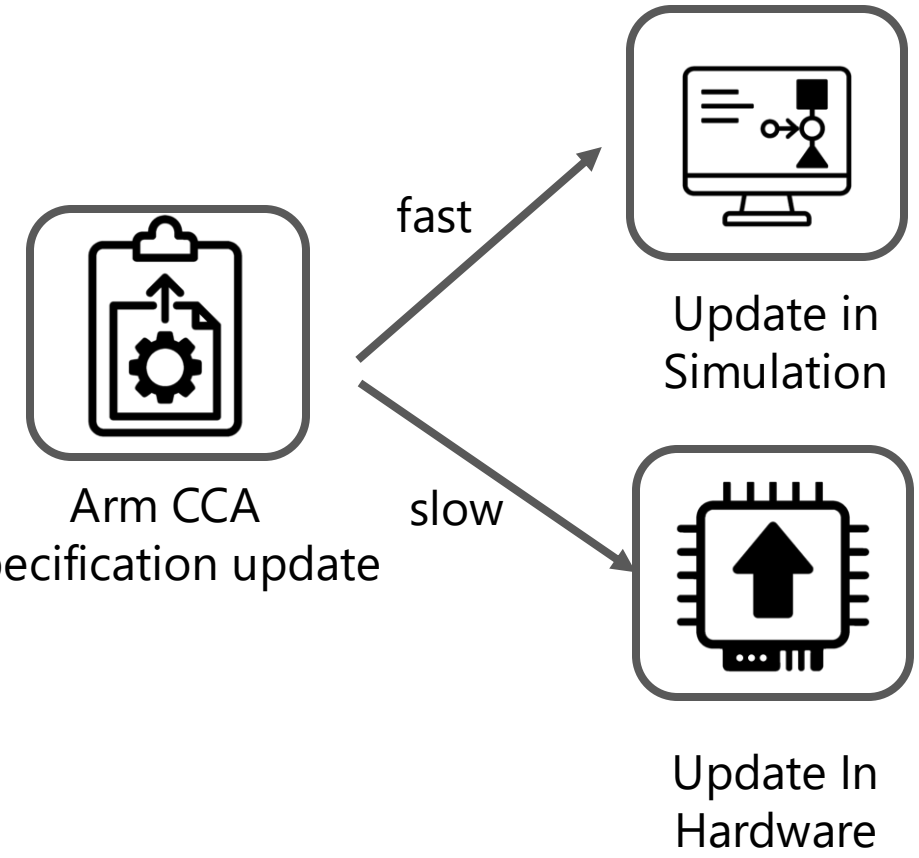


Radxa Rock5b RK3588 ~ 250 USD

https://radxa.com/products/rock5/5b/

What about OpenCCA once we have CCA Hardware?

OpenCCA bridges Gap between Specification and Hardware Catch-Up for Performance Estimation



We need:
- ✓ No locked-down firmware
- ✓ Documentation
- ✓ Affordable & Available

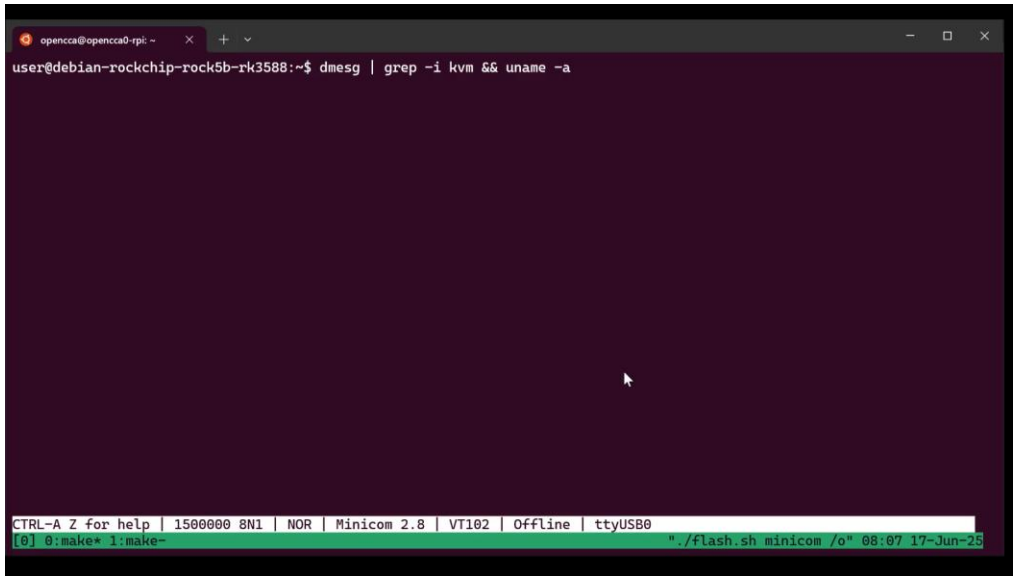
New FEAT_ features take time until implemented in HW

Estimate in Software with OpenCCA to experiment with firmware stack & estimate overheads

OpenCCA Demo
Realm VM: Compute heavy Benchmark

Identical binaries of: benchmark, guest + host kernel

Realm VM (2 vCPUs, 750 MB RAM)



OpenCCA on RK3588

4 x Cortex A55, 2.3GHz

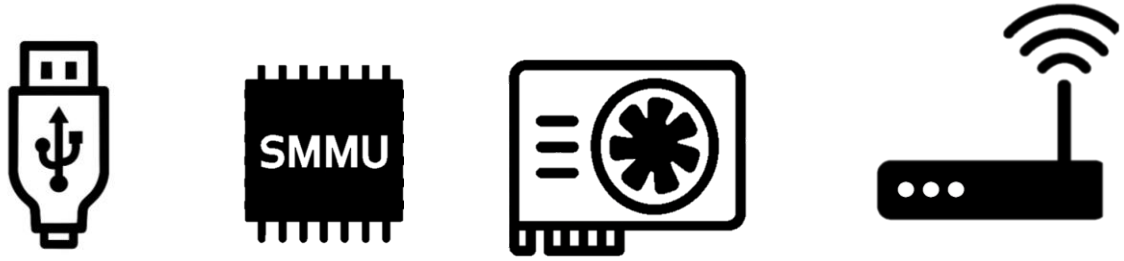
FVP on x86 AMD EPYC 9334

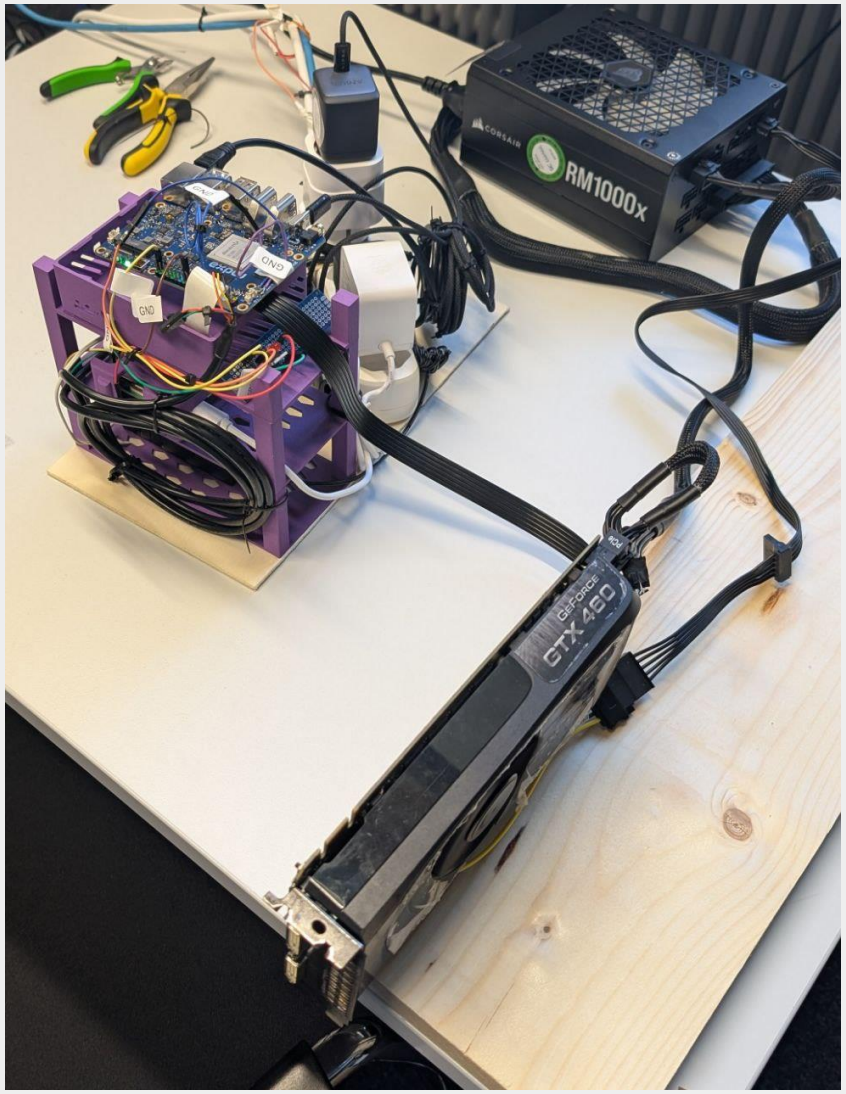
32 Cores, 2.7GHz

OpenCCA Demo
Real Hardware = Real Devices

OpenCCA runs on real hardware and can interact with real devices.

- RK3588 exposes PCIe lanes over NVMe slot
- Example: Research on Arm CCA with PCIe devices



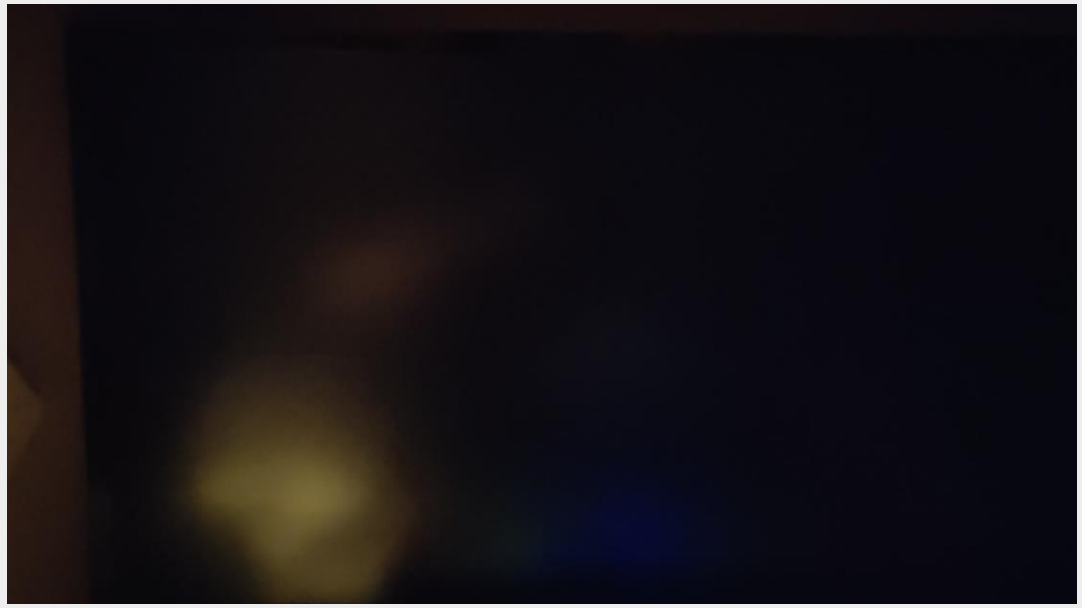


Connect Discrete GPU to OpenCCA

OpenCCA Demo
Android on RK3588 *(Work in Progress)*

- Porting Android to OpenCCA
- Platform boot:
  - Android on OpenCCA: **~ 1min 30 secs**
  - Android on FVP: **> 12 hours**

Android 14 (Linux 6.12) on OpenCCA/ RK3588

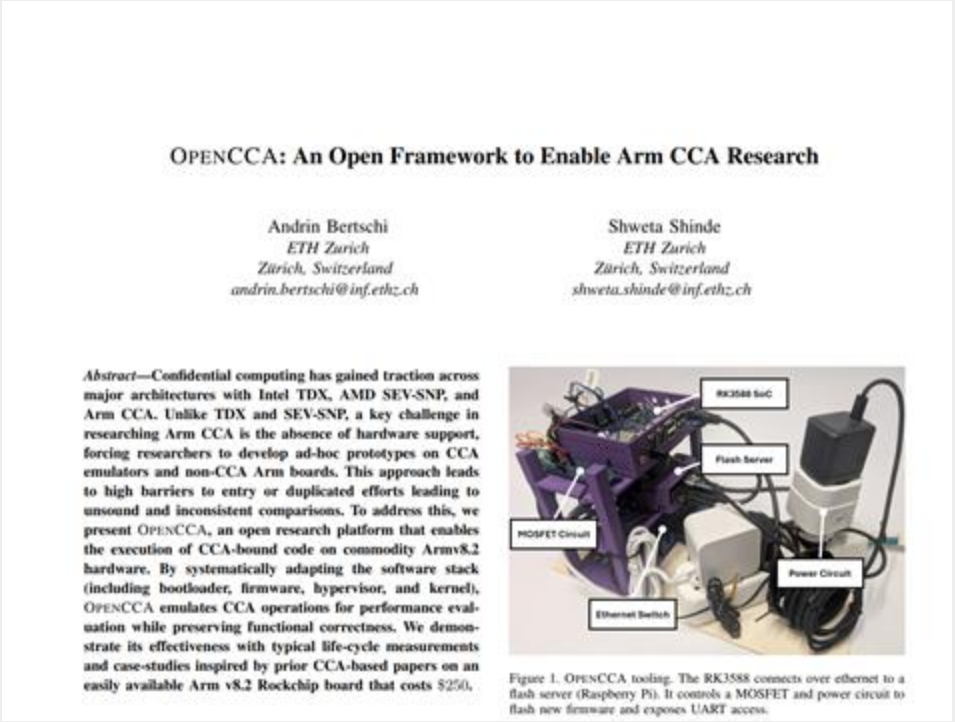


Thank You

- Paper and source code is online
- Get in touch!

OpenCCA:

- Open *Framework* for Performance Estimations
- Enable CCA on commodity Armv8 hardware for performance and accelerators support



OPENCCA: An Open Framework to Enable Arm CCA Research

Andrin Bertschi
ETH Zurich
Zürich, Switzerland
andrin.bertschi@inf.ethz.ch

Shweta Shinde
ETH Zurich
Zürich, Switzerland
shweta.shinde@inf.ethz.ch

***Abstract*—Confidential computing has gained traction across major architectures with Intel TDX, AMD SEV-SNP, and Arm CCA. Unlike TDX and SEV-SNP, a key challenge in researching Arm CCA is the absence of hardware support, forcing researchers to develop ad-hoc prototypes on CCA emulators and non-CCA Arm boards. This approach leads to high barriers to entry or duplicated efforts leading to unsound and inconsistent comparisons. To address this, we present OPENCCA, an open research platform that enables the execution of CCA-bound code on commodity Armv8.2 hardware. By systematically adapting the software stack (including bootloader, firmware, hypervisor, and kernel), OPENCCA emulates CCA operations for performance evaluation while preserving functional correctness. We demonstrate its effectiveness with typical life-cycle measurements and case-studies inspired by prior CCA-based papers on an easily available Arm v8.2 Rockchip board that costs $250.**

Figure 1. OPENCCA tooling. The RK3588 connects over ethernet to a flash server (Raspberry Pi). It controls a MOSFET and power circuit to flash new firmware and exposes UART access.

X: andrinbertschi

email: andrin.bertschi@inf.ethz.ch

web: https://opencca.github.io