

# OpenCCA: An Open Framework to Enable Research on Arm CCA

---

Andrin Bertschi, Shweta Shinde

ETH Zurich

[andrin.bertschi@inf.ethz.ch](mailto:andrin.bertschi@inf.ethz.ch)

4. July 2025

# Confidential Computing is Exciting!

## Industry Momentum

- Widely available in Hardware

**arm**  
CCA

**AMD**  
SEV-SNP

**intel**  
TDX

**RISC-V**  
Keystone

- Deployed in the cloud

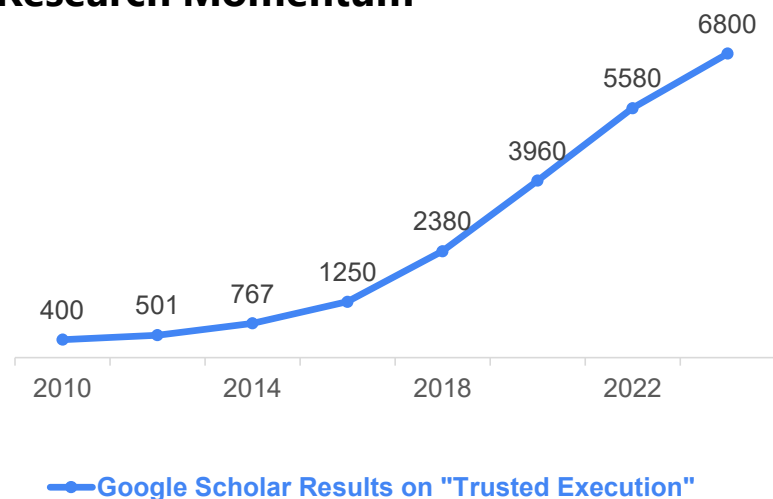
**aws**

**Azure**

**Google Cloud**

**Alibaba Cloud**

## Research Momentum



### Arm CCA

19 academic papers in last 4 years

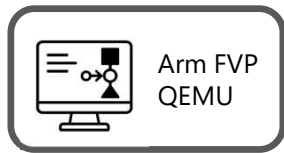
# How to research on Arm CCA

## Main Challenge on Arm CCA:

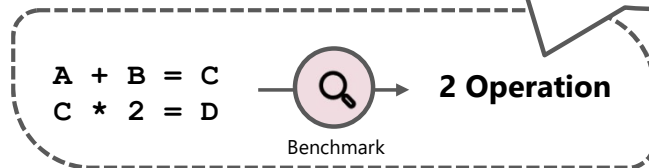
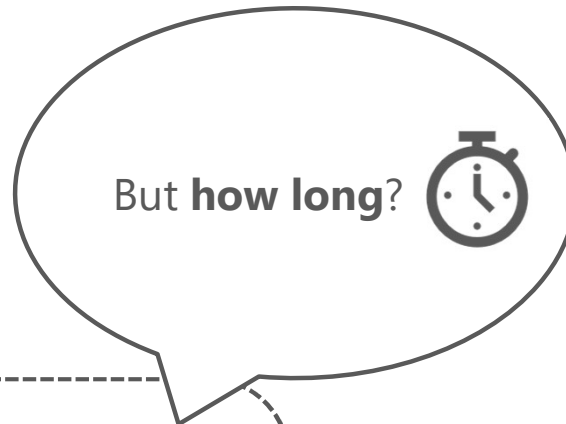
- No Hardware yet



## 1. Simulation



- Validate design + compatibility
- 17/19 papers



Academic Work	Simulation	Board Prototype	Board Arch.
Cage	FVP	Juno R2	8.0
Shelter	FVP	Juno R2	8.0
Scrutinizer	FVP	Juno R2	8.0
SoK on TZ & CCA	FVP	Juno R2	8.0
HitchHiker's Guide	FVP	Juno R2	8.0
FortifyPatch	FVP	Raspberry Pi 3B	8.0
RContainer	FVP	RK3399 Firefly	8.0
CubeVisor	FVP	RK3399 Rock 4B	8.0
ACAI	FVP	Zynq UltraScale+	8.0
TwinVisor	FVP	HiSilicon Kirin 990	8.2
Portal	FVP	OrangePi 5 Plus	8.2
Design & Verification	FVP	Neoverse N1	8.2
virtCCA	–	Undisclosed	8.4
Sharing is Leaking	–	AmpereOne	8.6
CPC Maintenance	FVP	SEV SNP (x86)	–
GuaranTEE	FVP	–	–
Devlore	FVP	–	–
BarriCCAd	QEMU	–	–
Aster	QEMU	–	–



## Arm CCA

19 academic papers in last 4 years

# How to research on Arm CCA

## Main Challenge on Arm CCA:

- No Hardware yet



### 1. Simulation



- Validate design + compatibility
- 17/19 papers

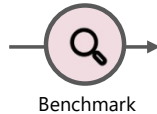
### 2. Board Prototype



- Validate performance
- 15/19 papers



$A + B = C$   
 $C * 2 = D$



**5 cycles**  
*metric of time*

Includes **microarchitectural effects of complex hardware**: out-of-order exec, pipeline stalls, cache misses, ...

Academic Work	Simulation	Board Prototype	Board Arch.
Cage	FVP	Juno R2	8.0
Shelter	FVP	Juno R2	8.0
Scrutinizer	FVP	Juno R2	8.0
SoK on TZ & CCA	FVP	Juno R2	8.0
HitchHiker's Guide	FVP	Juno R2	8.0
FortifyPatch	FVP	Raspberry Pi 3B	8.0
RContainer	FVP	RK3399 Firefly	8.0
CubeVisor	FVP	RK3399 Rock 4B	8.0
ACAI	FVP	Zynq UltraScale+	8.0
TwinVisor	FVP	HiSilicon Kirin 990	8.2
Portal	FVP	OrangePi 5 Plus	8.2
Design & Verification	FVP	Neoverse N1	8.2
virtCCA	–	Undisclosed	8.4
Sharing is Leaking	–	AmpereOne	8.6
CPC Maintenance	FVP	SEV SNP (x86)	–
GuaranTEE	FVP	–	–
Devlore	FVP	–	–
BarriCCAd	QEMU	–	–
Aster	QEMU	–	–



## Arm CCA

19 academic papers in last 4 years

# How to research on Arm CCA



Academic Work	Simulation	Board Prototype	Board Arch.
Cage	FVP	Juno R2	8.0
Shelter	FVP	Juno R2	8.0
Scrutinizer	FVP	Juno R2	8.0
SoK on TZ & CCA	FVP	Juno R2	8.0
HitchHiker's Guide	FVP	Juno R2	8.0
FortifyPatch	FVP	Raspberry Pi 3B	8.0
RContainer	FVP	RK3399 Firefly	8.0
CubeVisor	FVP	RK3399 Rock 4B	8.0
ACAI	FVP	Zynq UltraScale+	8.0
TwinVisor	FVP	HiSilicon Kirin 990	8.2
Portal	FVP	OrangePi 5 Plus	8.2
Design & Verification	FVP	Neoverse N1	8.2
virtCCA	–	Undisclosed	8.4
Sharing is Leaking	–	AmpereOne	8.6
CPC Maintenance	FVP	SEV SNP (x86)	–
GuaranTEE	FVP	–	–
Devlore	FVP	–	–
BarriCCAd	QEMU	–	–
Aster	QEMU	–	–



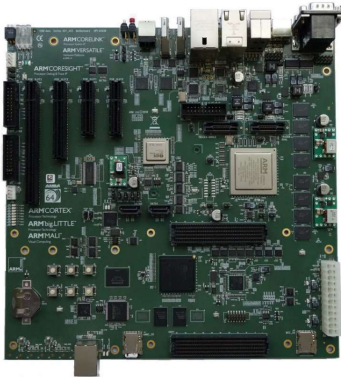
## Arm CCA

19 academic papers in last 4 years

# How to research on Arm CCA

## Arm Juno R2

- 10 years old
- No longer manufactured
- Initial Price: 10k USD



<https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/Juno%20r2%20datasheet.pdf>

Academic Work	Simulation	Board Prototype	Board Arch.
Cage	FVP	Juno R2	8.0
Shelter	FVP	Juno R2	8.0
Scrutinizer	FVP	Juno R2	8.0
SoK on TZ & CCA	FVP	Juno R2	8.0
HitchHiker's Guide	FVP	Juno R2	8.0
FortifyPatch	FVP	Raspberry Pi 3B	8.0
RContainer	FVP	RK3399 Firefly	8.0
CubeVisor	FVP	RK3399 Rock 4B	8.0
ACAI	FVP	Zynq UltraScale+	8.0
TwinVisor	FVP	HiSilicon Kirin 990	8.2
Portal	FVP	OrangePi 5 Plus	8.2
Design & Verification	FVP	Neoverse N1	8.2
virtCCA	–	Undisclosed	8.4
Sharing is Leaking	–	AmpereOne	8.6
CPC Maintenance	FVP	SEV SNP (x86)	–
GuaranTEE	FVP	–	–
Devlore	FVP	–	–
BarriCCAd	QEMU	–	–
Aster	QEMU	–	–



## Arm CCA

19 academic papers in last 4 years

# How to research on Arm CCA

Undisclosed



Academic Work	Simulation	Board Prototype	Board Arch.
Cage	FVP	Juno R2	8.0
Shelter	FVP	Juno R2	8.0
Scrutinizer	FVP	Juno R2	8.0
SoK on TZ & CCA	FVP	Juno R2	8.0
HitchHiker's Guide	FVP	Juno R2	8.0
FortifyPatch	FVP	Raspberry Pi 3B	8.0
RContainer	FVP	RK3399 Firefly	8.0
CubeVisor	FVP	RK3399 Rock 4B	8.0
ACAI	FVP	Zynq UltraScale+	8.0
TwinVisor	FVP	HiSilicon Kirin 990	8.2
Portal	FVP	OrangePi 5 Plus	8.2
Design & Verification	FVP	Neoverse N1	8.2
virtCCA	–	Undisclosed	8.4
Sharing is Leaking	–	AmpereOne	8.6
CPC Maintenance	FVP	SEV SNP (x86)	–
GuaranTEE	FVP	–	–
Devlore	FVP	–	–
BarriCCAd	QEMU	–	–
Aster	QEMU	–	–



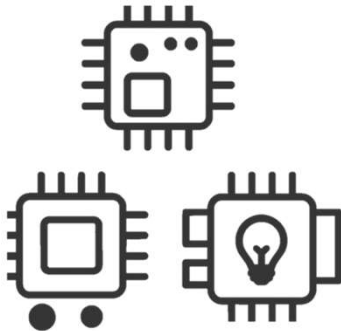
Source

## Arm CCA

19 academic papers in last 4 years

# How to research on Arm CCA

Variety of different boards,  
all with different hardware  
features



Academic Work	Simulation	Board Prototype	Board Arch.
Cage	FVP	Juno R2	8.0
Shelter	FVP	Juno R2	8.0
Scrutinizer	FVP	Juno R2	8.0
SoK on TZ & CCA	FVP	Juno R2	8.0
HitchHiker's Guide	FVP	Juno R2	8.0
FortifyPatch	FVP	Raspberry Pi 3B	8.0
RContainer	FVP	RK3399 Firefly	8.0
CubeVisor	FVP	RK3399 Rock 4B	8.0
ACAI	FVP	Zynq UltraScale+	8.0
TwinVisor	FVP	HiSilicon Kirin 990	8.2
Portal	FVP	OrangePi 5 Plus	8.2
Design & Verification	FVP	Neoverse N1	8.2
virtCCA	–	Undisclosed	8.4
Sharing is Leaking	–	AmpereOne	8.6
CPC Maintenance	FVP	SEV SNP (x86)	–
GuaranTEE	FVP	–	–
Devlore	FVP	–	–
BarriCCAd	QEMU	–	–
Aster	QEMU	–	–



**Arm CCA**

19 academic papers in last 4 years



# The Need for Open Framework for Performance Evaluation



## Software Simulation:

- ✓ Functionality
- ✓ Compatibility

Academic Work	Simulation	Board Prototype	Board Arch.
Cage	FVP	Juno R2	8.0
Shelter	FVP	Juno R2	8.0
Scrutinizer	FVP	Juno R2	8.0
SoK on TZ & CCA	FVP	Juno R2	8.0
HitchHiker's Guide	FVP	Juno R2	8.0
FortifyPatch	FVP	Raspberry Pi 3B	8.0
RContainer	FVP	RK3399 Firefly	8.0
CubeVisor	FVP	RK3399 Rock 4B	8.0
ACAI	FVP	Zynq UltraScale+	8.0
TwinVisor	FVP	HiSilicon Kirin 990	8.2
Portal	FVP	OrangePi 5 Plus	8.2
Design & Verification	FVP	Neoverse N1	8.2
virtCCA	–	Undisclosed	8.4
Sharing is Leaking	–	AmpereOne	8.6
CPC Maintenance	FVP	SEV SNP (x86)	–
GuaranTEE	FVP	–	–
Devlore	FVP	–	–
BarriCCade	QEMU	–	–
Aster	QEMU	–	–



## Custom Ad-Hoc:

- × Difficult to compare
- × Difficult to reproduce
- × Repeated engineering

## 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.

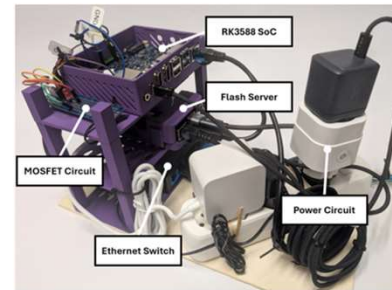


Figure 1. OPENCCA tooling. The RK3588 connects over ethernet to a flash server (Raspberry Pi). It controls a MOSFET and power circuit to flash new firmware and exposes UART access.

# OpenCCA: An Open Framework to Enable Research on Arm CCA

# OpenCCA Design Goals



Minimal changes to CCA reference stack → Preserve functionality



No security guarantees  
*Only for benchmarking & accelerator support*



Target: Affordable & Available Armv8 Boards



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

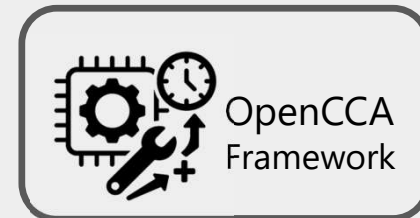
## Step 1: Simulation



Validate design



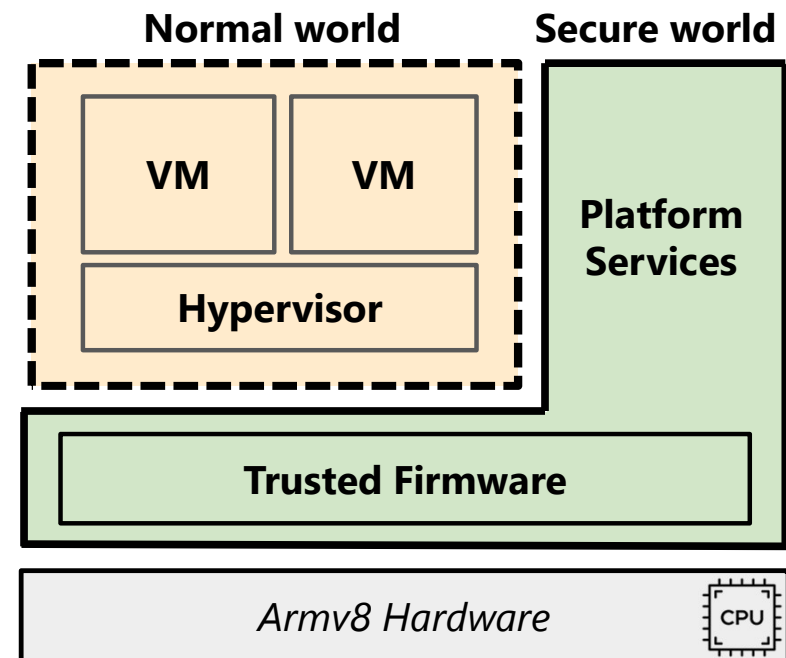
## Step 2: Hardware Prototype



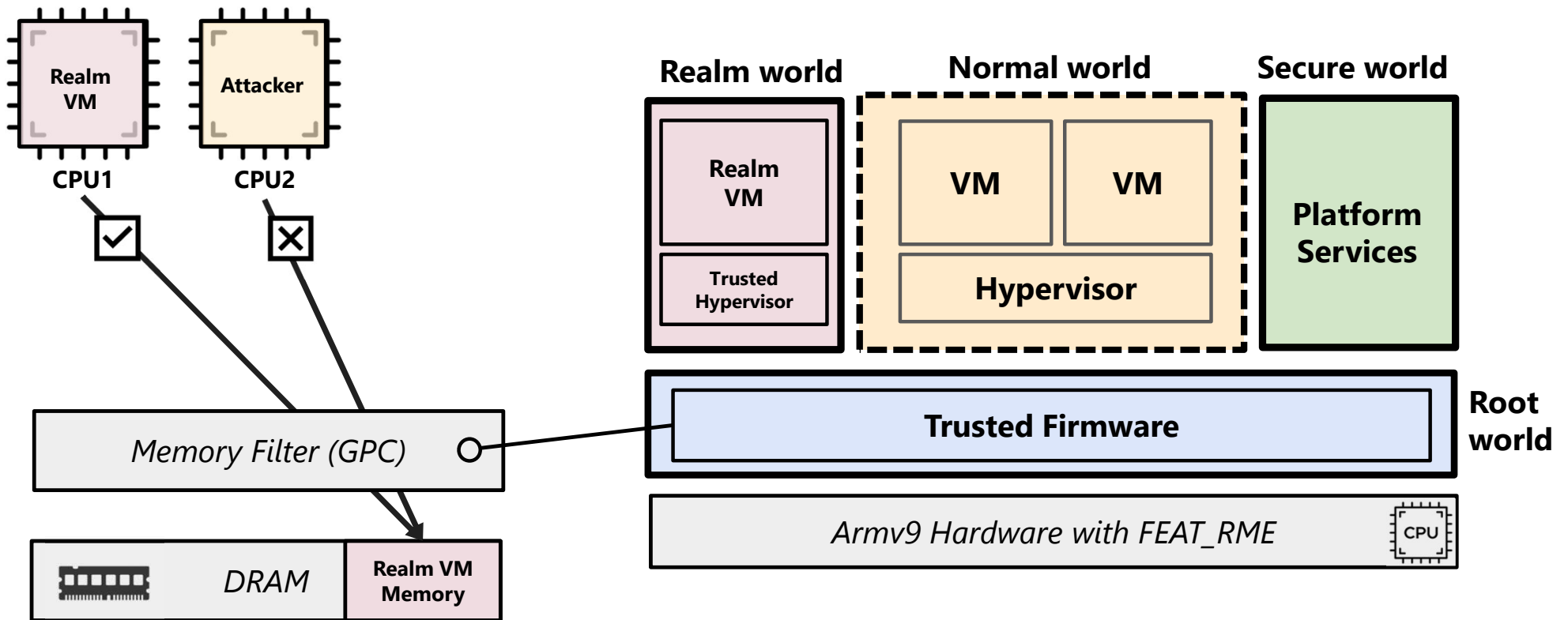
Evaluate & Compare  
Performance

# Background on Arm CCA

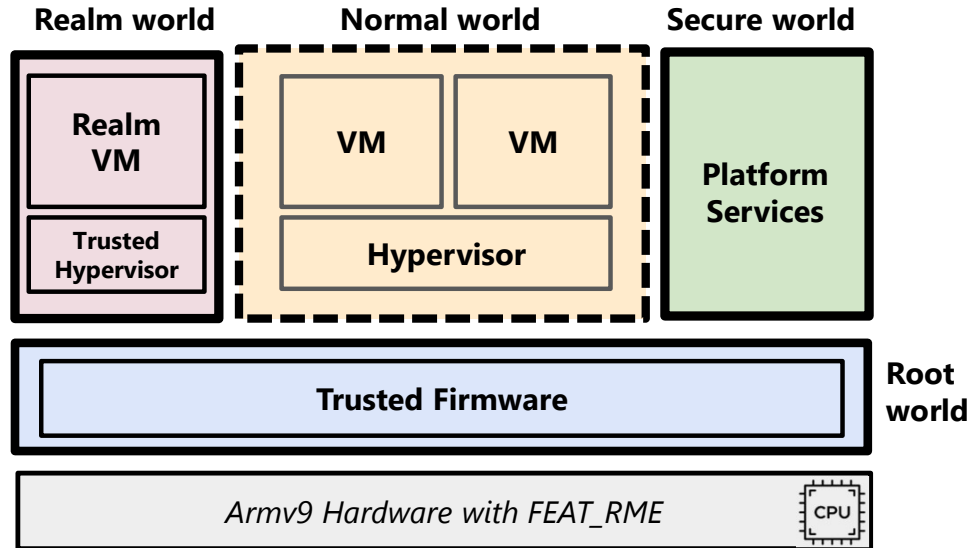
- Before Armv9: TrustZone



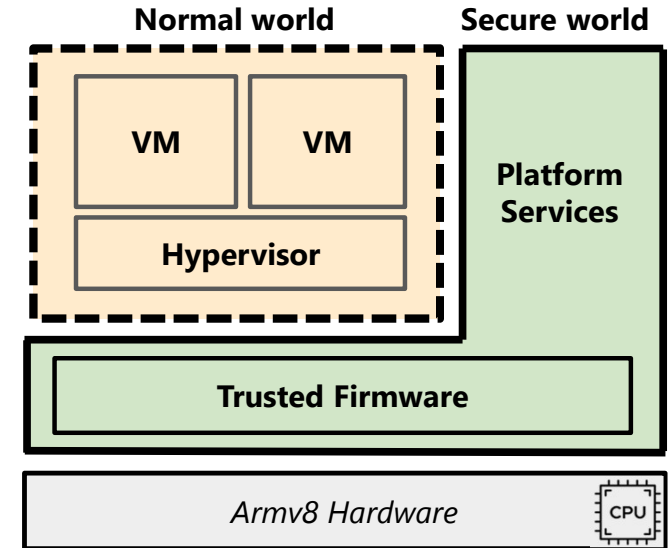
# Background on Arm CCA



# How do you even run Arm CCA on older Hardware?

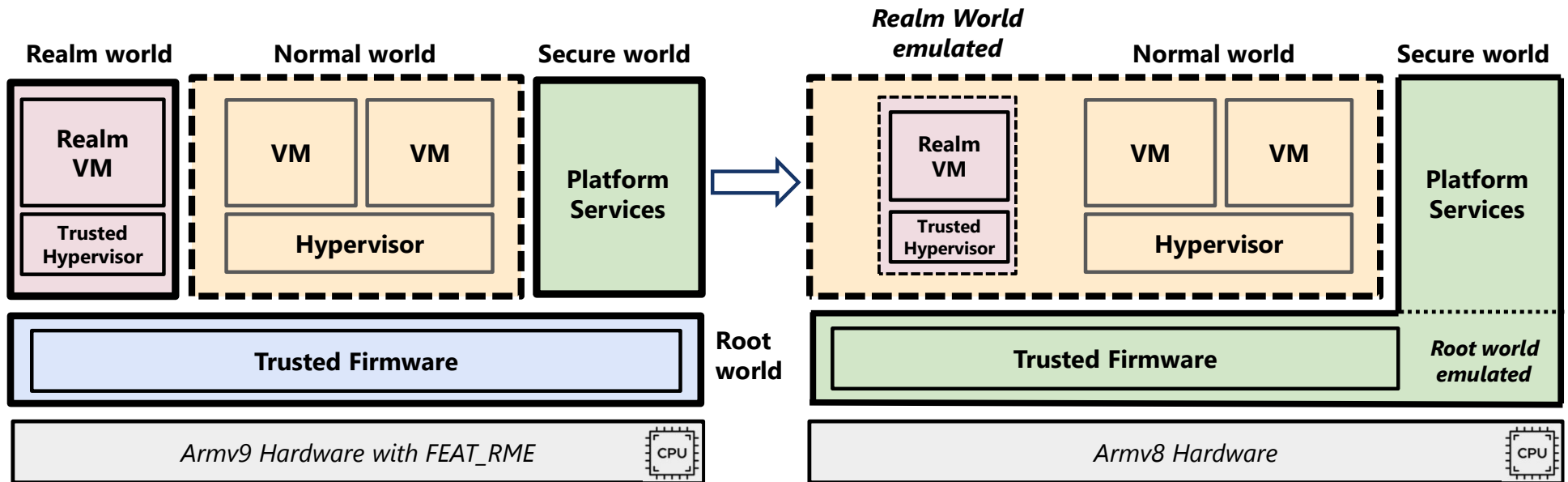


Armv9



Armv8

# How do you even run Arm CCA on older Hardware?



Emulate CCA in software within the constraints of Armv8 Hardware

# Example: Emulating Hardware in Software

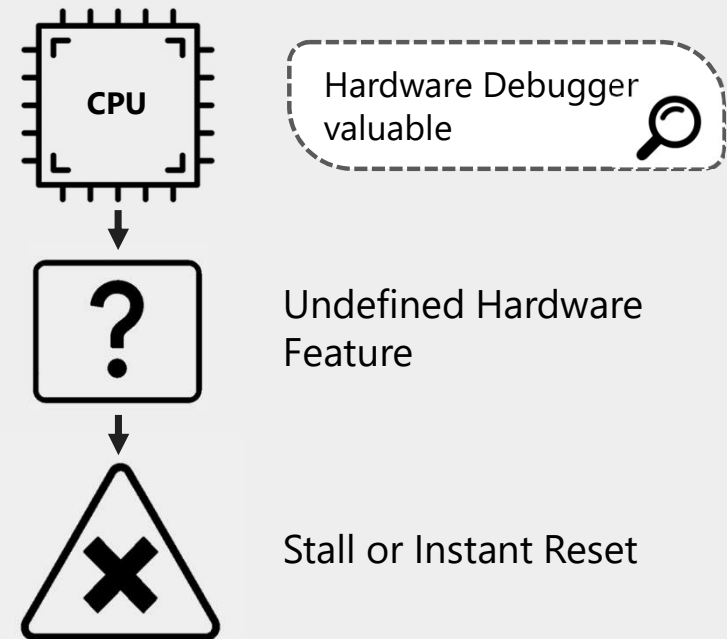
- **Tradeoff** between **Compatibility** & **Overhead**
- **Fake missing parts in software** while keeping changes small
- Return **predefined values** instead of querying the hardware

## Memory Filter (GPC) Registers:

GPCCR\_EL3  
GPTBR\_EL3

→ Fake →

AFSRx\_EL3  
Return fixed configuration





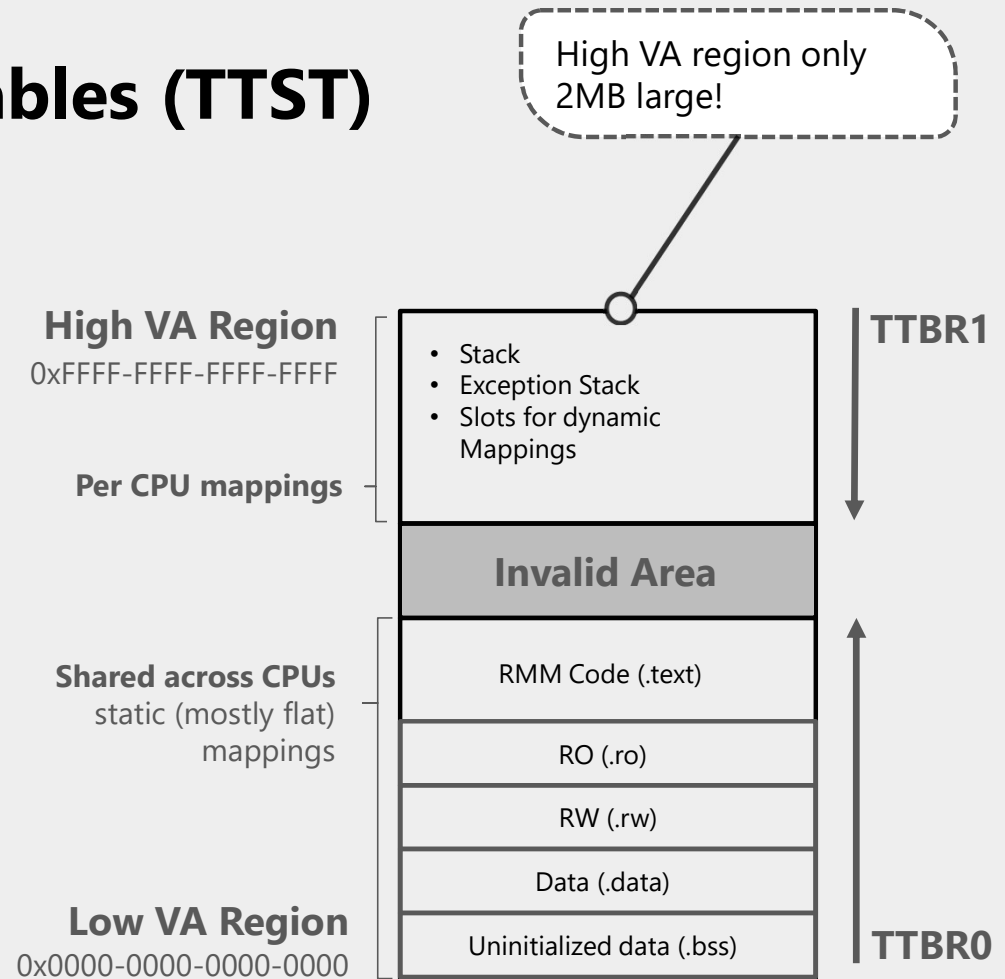
# Example: Short Translation Tables (TTST)

## Memory Layout in Trusted Hypervisor

- Paging implicitly uses FEAT\_TTST (Armv8.4)
- Decrease lower limit on size of VA region
  - Fewer page walks needed (L3 -> Page)
- Not available on our hardware!

## Workaround:

- Increase High VA region to 1GB
- Longer page table walk (L2 -> L3 -> Page)



## Details: Implementation

- Run general purpose realm VMs
- Enlighten 2 firmware components
- Keeping rest of reference stack unmodified

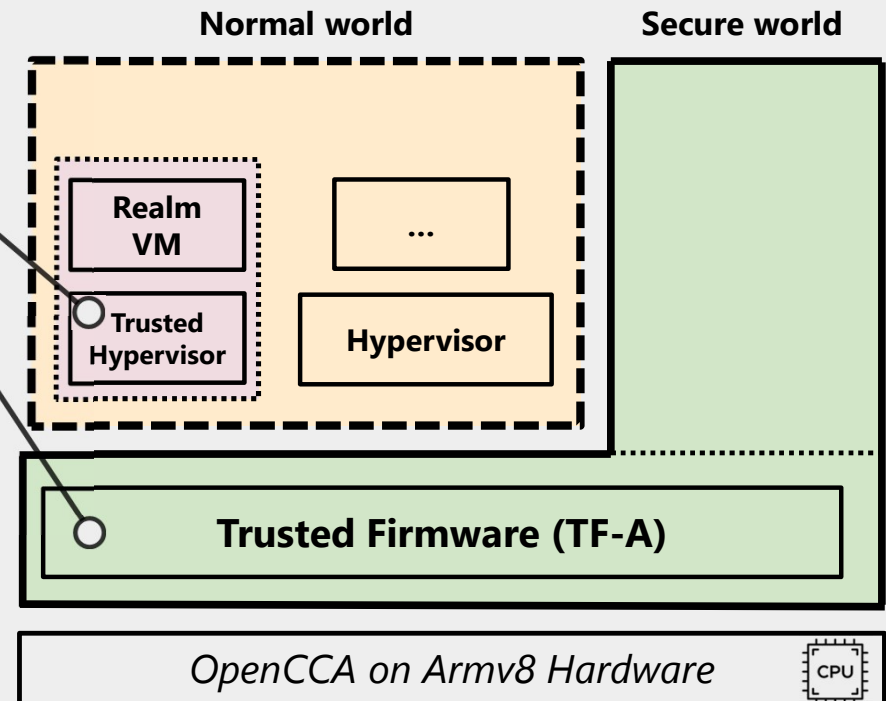
### Software changes:

- Trusted Hypervisor (RMM): +1440 LoC
- Trusted Firmware (TF-A): + 940 LoC

---

**Changes (C/C++/Asm)** < +1% LoC

Preserve functionality without security



# Choosing a Hardware Platform

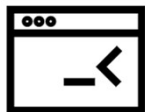
## Selection Criteria



No Vendor lock  
*Firmware flashable*



Documentation  
*Technical Reference Manual*

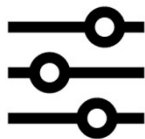


Support for Firmware  
*Arm Trusted Firmware*

## Also relevant



Price &  
Availability



Modern Hardware  
Features

Board	Released	SoC	GIC	Price (USD)	Cores	GPU	TF-A Code
Intel Stratix 10 SX DK	2013	Intel Stratix 10	GICv2	9,000	A53	N/A	✓
AmloGIC Meson S905 (GXBB)	2015	S905	GICv2	unknown	A53	Mali 450	✓
HiKey	2015	Kirin 620	GICv2	75-100	A53	Mali 450	✓
Arm Juno r2	ca. 2015	Juno r2 SoC	GICv2	10,000	A72, A53	Mali T624	✓
NXP i.MX7 WaRP7	2016	i.MX 7 Solo	GICv2	100	A7, M4	N/A	✓
A64-OLinuXino	2016	Allwinner A64	GICv2	100	A53	Mali 400	✓
AmloGIC Meson S905x (GXL)	2016	S905x	GICv2	unknown	A53	Mali 450 MP3	✓
NXP i.MX 8QM MEK	2016/17	i.MX 8QM	GICv3	1,200	A72, A53, M4F	GC7000XSVX	✓
NXP i.MX 8MQ EVK	2016/17	i.MX 8MQ	GICv3	500	A53, M4	GC7000Lite	✓
NXP i.MX 8ULP EVK	2016/17	i.MX 8 ULP	GICv3	550-650	A53, M33	GC520	✓
Xilinx Zynq ZCU102 EVK	ca. 2017	2FFVB1156E	GICv2	3,200	A53, R5F	Mali 400 RP2	✓
AmloGIC Meson A113D (AXG)	2017	S400	GICv2	unknown	A53	2D GFX Engine	✓
HiKey 960	2017	Kirin 960 SoC	GICv2	250	A73, A53	Mali G71 MP8	✓
AmloGIC Meson S905X2 (G12A)	2018	S905x2	GICv2	unknown	A53	Mali-G31 MP2	✓
HiKey 970	2018	Kirin 970	GICv2	300	A73, A53	Mali G72 MP12	✓
Raspberry Pi 3 (B+)	2018	BCM2837B0	custom	25	A53	VideoCore IV	✓
Intel Agilex 7M HBM2e DK	2019	Intel Agilex 7	GICv2	10,000	A53	N/A	✓
Marvell CE7 CN9132 EVB	2019	CN9132	GICv2	600-700	A72	N/A	✓
Ziver MTK8183 Dev. Board	2019	MT8183	GICv3	150	A73, A53	Mali G72 MP3	✓
Raspberry Pi 4	2019	BCM2711	GICv2	35	A72	VideoCore VI	✓
Huawei Mate 30 Pro	2019	Kirin 990	unknown	300	A76, A55	Mali-G76	✗
Arm Neoverse N1 SDP	2020	Dawn Ares	GICv4.1	10,000	N1	HDLCD	✓
Aspeed AST2700 EVB	2020	AST2700	GICv3	unknown	A35, M4	AST2700 2D VE	✓
RK3399 Rock4	2021	RK3399	GICv3	<200	A72, A53	Mali-T864	✓
NVIDIA Jetson TX2 NX DK	2021	Tegra X2	GICv2	350	Denver2, A57	GP10B	✓
MediaTek 8186	2021	Kompanio 520	GICv3	unknown	A76, A55	Mali-G52 MP2	✓
MediaTek 8192	2021	Kompanio 820	GICv3	unknown	A55, A76	Mali G57 MC5	✓
MediaTek 8188	2022	Kompanio 838	GICv3	unknown	A55, A78	Mali G57 MC3	✓
MediaTek 8195	2022	Kompanio 1380	GICv3	unknown	A55, A78	Mali G57 MC5	✓
Genio 700 (MT8390)	2023	MT8390	GICv3	700	A78, A55	Mali-G57	✓
Orange Pi 5 Plus	2023	RK3588	GICv3	<200	A76, A55	Mali-G610	✓
Supermicro MegaDC (Server)	2023	AmpereOne	unknown	unknown	custom built	N/A	✗
Raspberry Pi 5	2023	BCM2712	GICv2	120	A76	VideoCore VII	✓
<b>Radxa Rock 5b (OPENCCA)</b>	<b>2023</b>	<b>RK3588</b>	<b>GICv3</b>	<b>250</b>	<b>A76, A55</b>	<b>Mali-G610</b>	✓
NXP i.MX 93 QS EVK	ca. 2024	i.MX 93	GICv3/v4	300	A55, M33	N/A	✓
Arrow AXES-Eagle DK	2024	Intel Agilex 5	GICv3	900-1,000	A55, A76	N/A	✓
Radxa Orion O6	2024	Cix CD8180	GICv4	500-600	A720, A520	ImtIs. G720 MC6	✗



Exploring Hardware Boards  
for OpenCCA

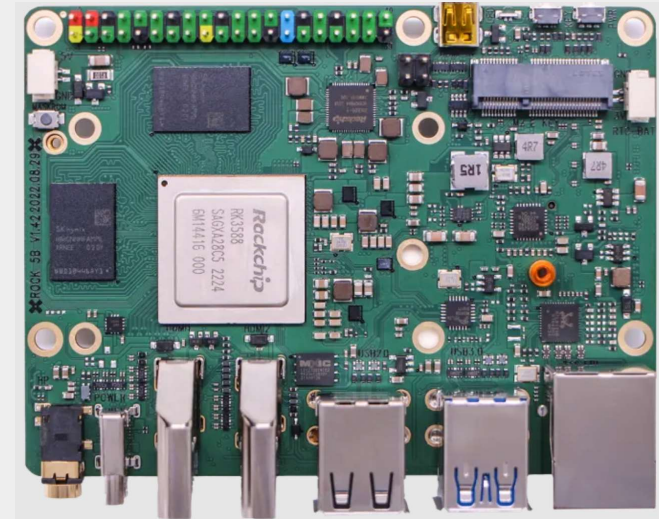
# RK3588 SoC as a Platform

## Key Specs:

- 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

### **RK3588 Rock5b Board:**

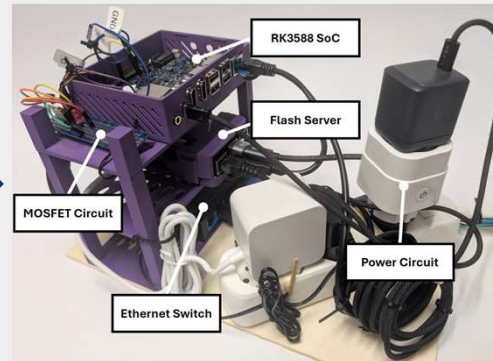
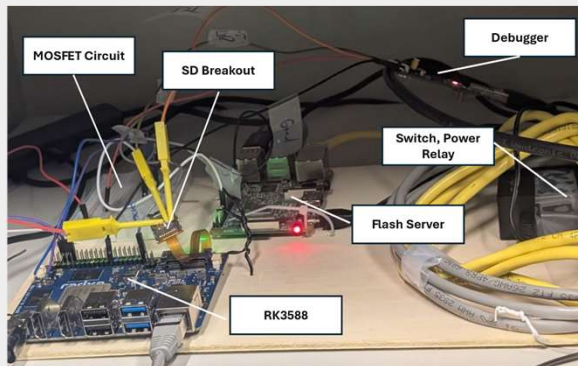
- ✓ Not vendor locked
- ✓ Well documented
- ✓ Affordable & Available
- ✓ Support System



**Radxa Rock5b RK3588**  
**~ 250 USD**

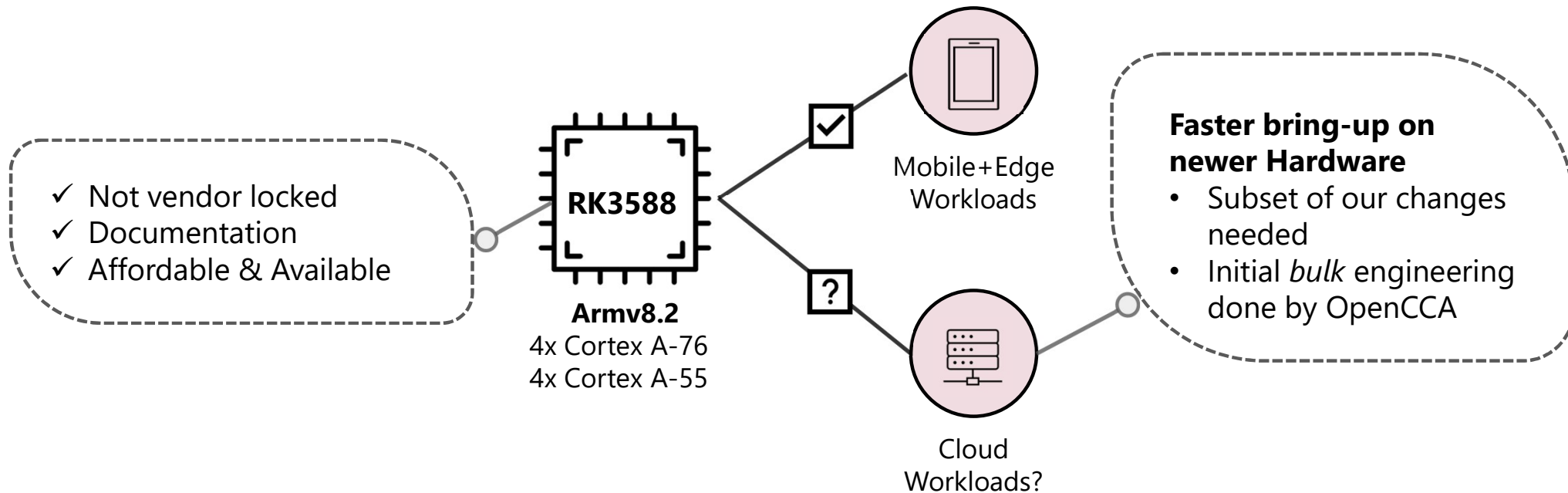
<https://radxa.com/products/rock5/5b/>

# OpenCCA on RK3588



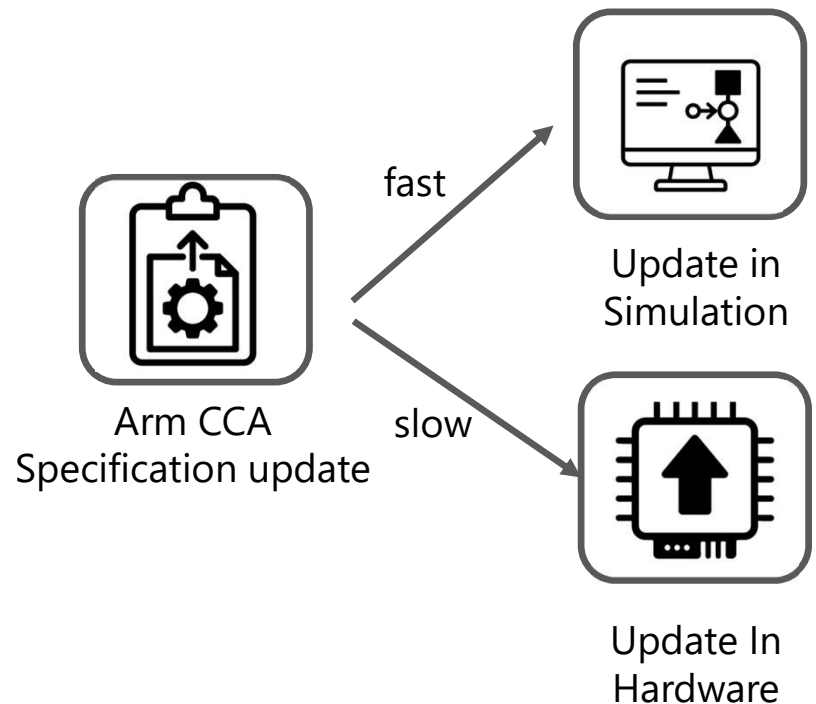
- OpenCCA "Box" with support for hardware debugging, firmware flashing and power management

# Suitable Workloads on RK3588 and Porting Work to other boards



# ***What about OpenCCA once we have CCA Hardware?***

OpenCCA bridges Gap between Specification and Hardware for Performance Estimation



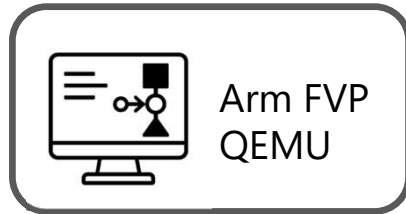
## **Example Case: Arm CCA Planes**

- Alpha spec released, likely not in first iteration of hardware

➡ **Estimate in Software with OpenCCA to experiment with Feature**

# How we evaluate Designs with OpenCCA

## Step 1: Simulation



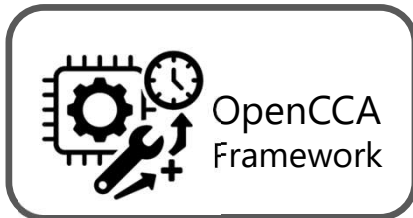
Validate design

## Step 2: Benchmark Baseline



Unmodified OpenCCA stack

## Step 3: Benchmark design



New research design

## Step 4: Analyze Results



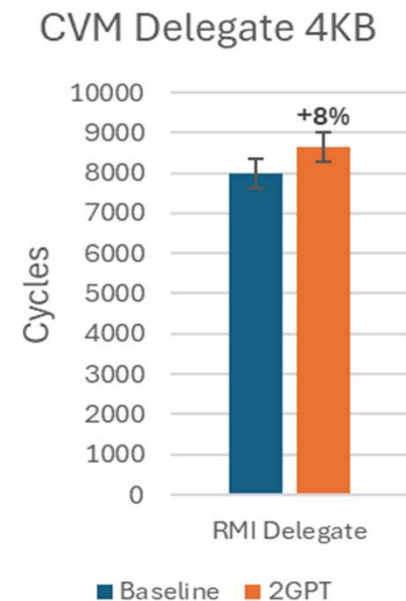


# Example Benchmark Campaign with Case Study

- Demonstrate **OpenCCA adaptability and ease of integration** with Case Studies
- 2 Memory Filter Case Study
  - Filter1: System Default
  - Filter2: Mark memory as Normal world
- **Changes:** ~2300 LoC added, 4 person hours



Checkout paper & code



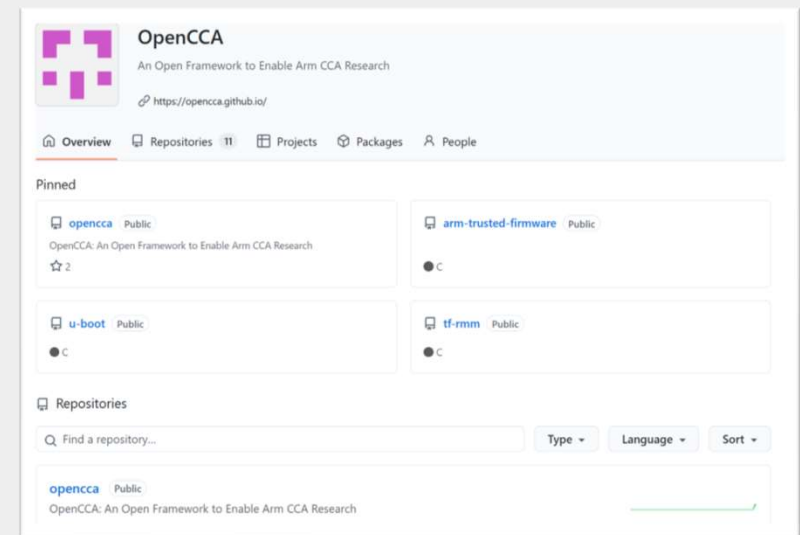
# Current Status & Next Steps

## Current Status

- Run confidential VMs with Arm reference stack
  - TF-A: v2.11
  - RMM: v0.5.0
  - Linux 6.12 (cca/full-v5+v7)
  - Kvmtool (cca/v3)
- Initial version Open Source

## Next Steps:

- Update to latest version of reference stack
- Software Bring Up for RK3588



<https://opencca.github.io>

# Live Demo

- Comparison of **live demo of OpenCCA** with pre-recorded **software simulation**
- Run **realm VM with compute heavy task** to visualize performance benefits



# 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.

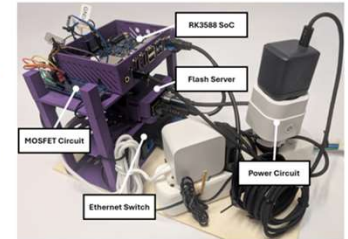


Figure 1. OPENCCA tooling. The RK3588 connects over ethernet to a flash server (Raspberry Pi). It controls a MOSFET and power circuit to flash new firmware and exposes UART access.



X: andrinbertschi

email: [andrin.bertschi@inf.ethz.ch](mailto:andrin.bertschi@inf.ethz.ch)

web: <https://opencca.github.io>