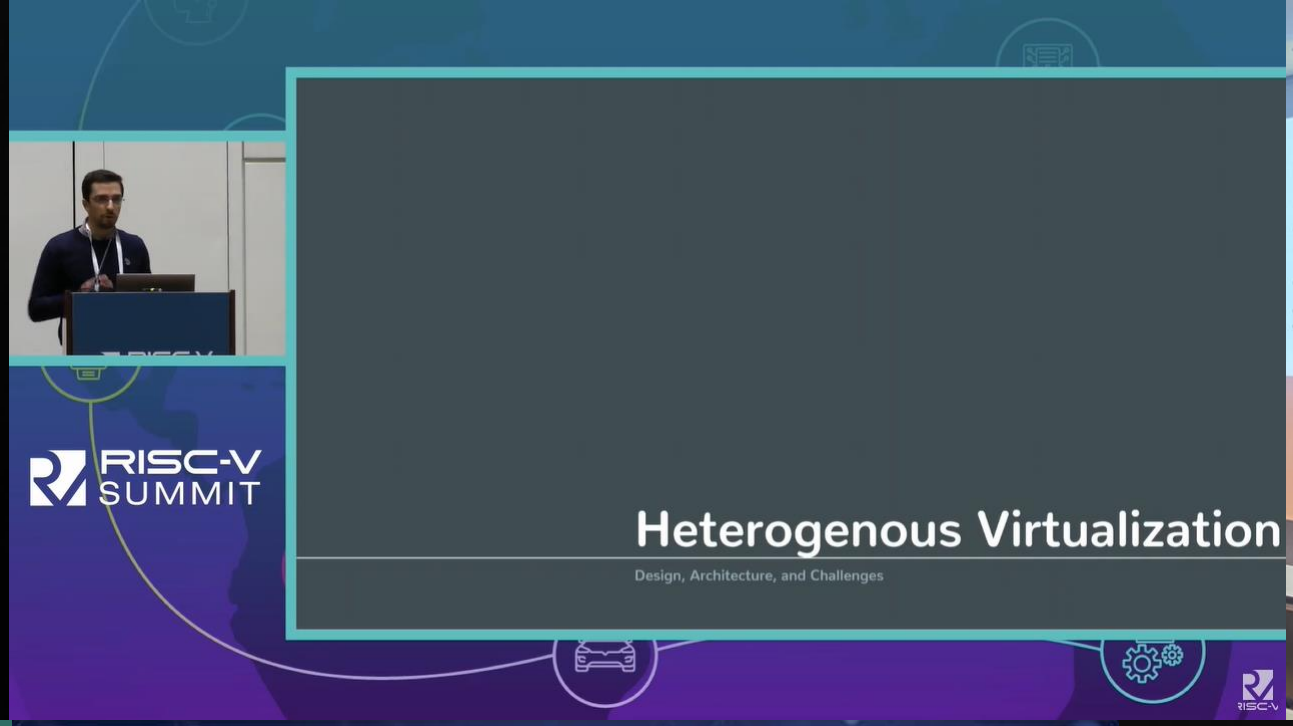
BAO

CONFIDENTIAL COMPUTING CONSORTIUM

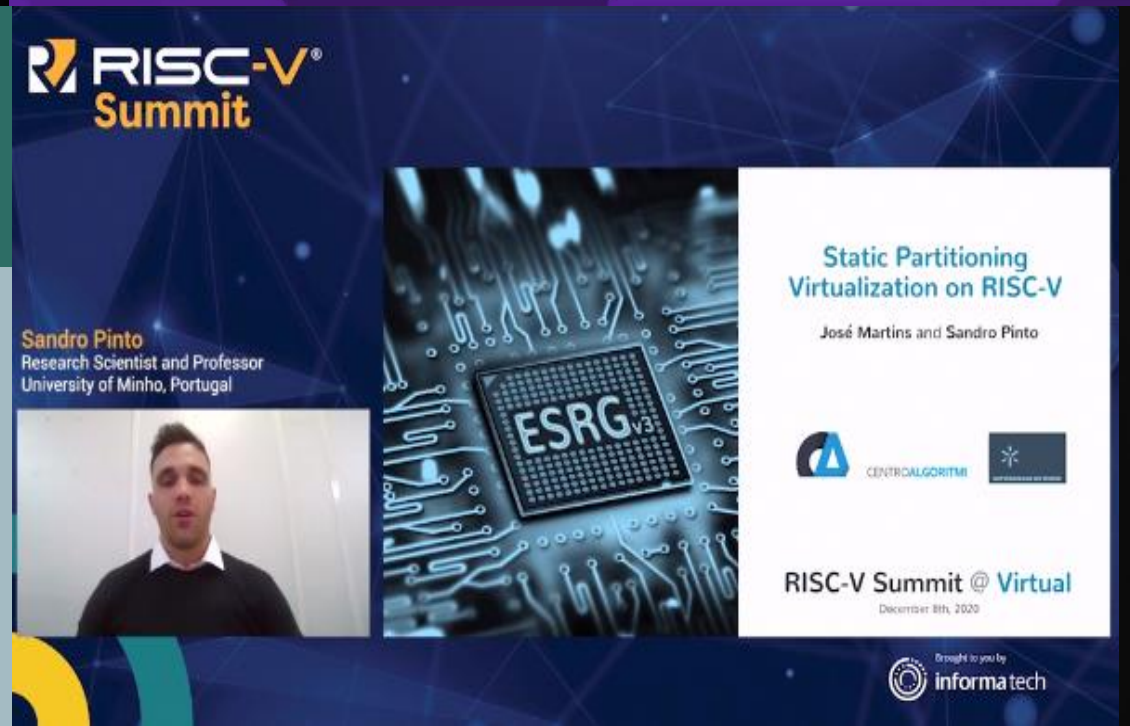
CCC Technical Advisory Council

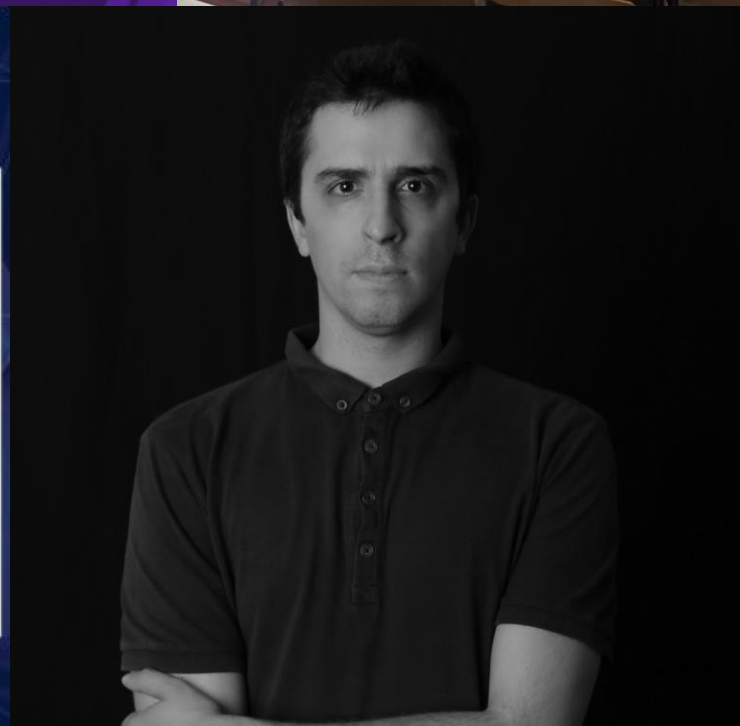
June 1st, 2023



EOSS
EMBEDDED OPEN SOURCE SUMMIT

embeddedworld
Exhibition&Conference





Agenda

01 **Introduction**
Virtualization in a nutshell

02 **Bao Hypervisor**
Overview and Highlights

03 **Bao & TEE**
Overview and Support

04 **Bao Roadmap**
Roadmap at a Glance

Introduction

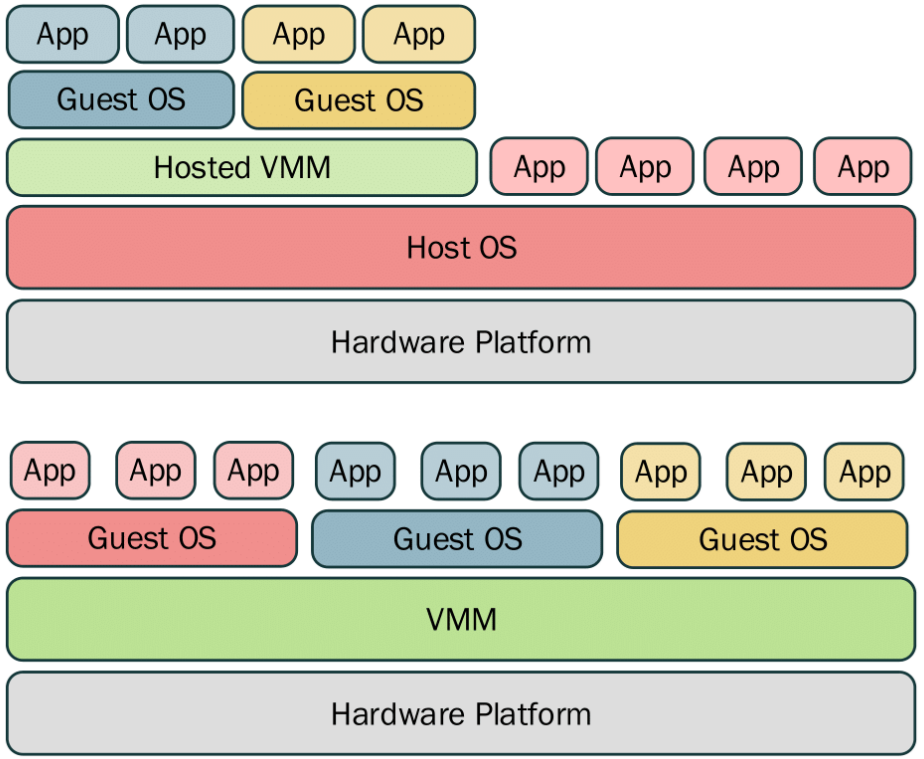
Virtualization in a nutshell

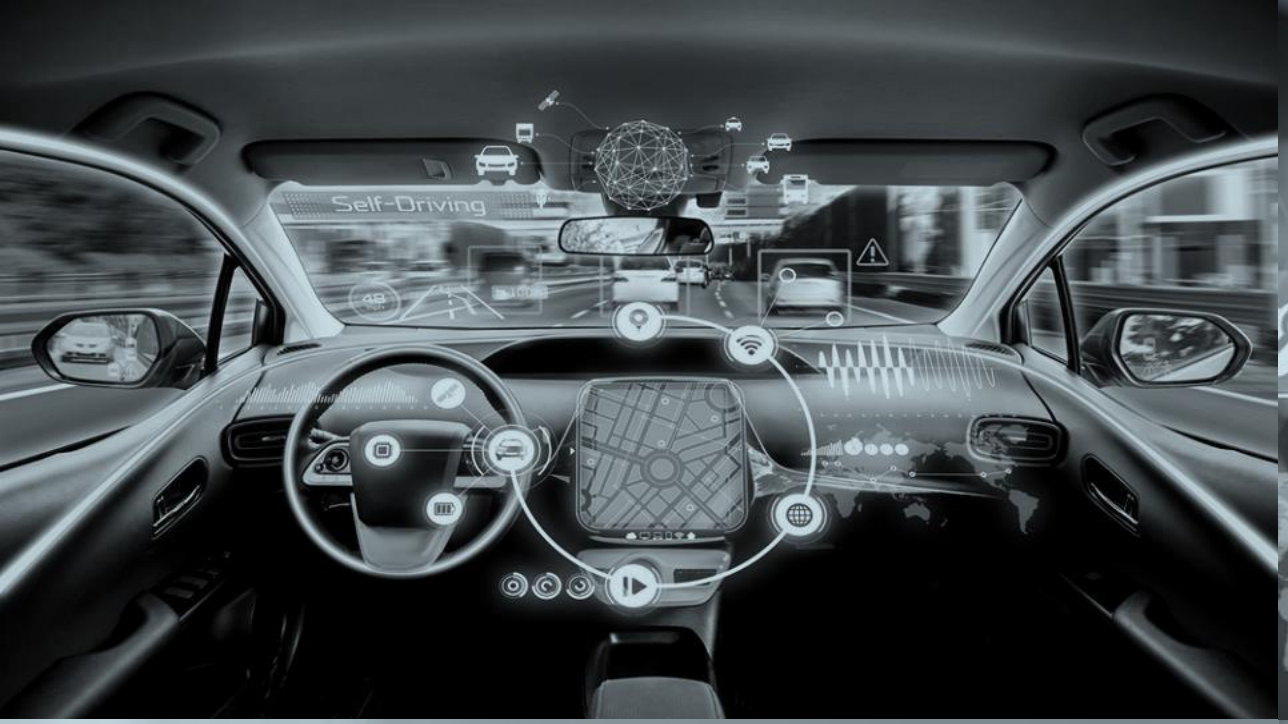
Virtualization

Allows the execution of multiple Operating Systems in the same hardware platform.

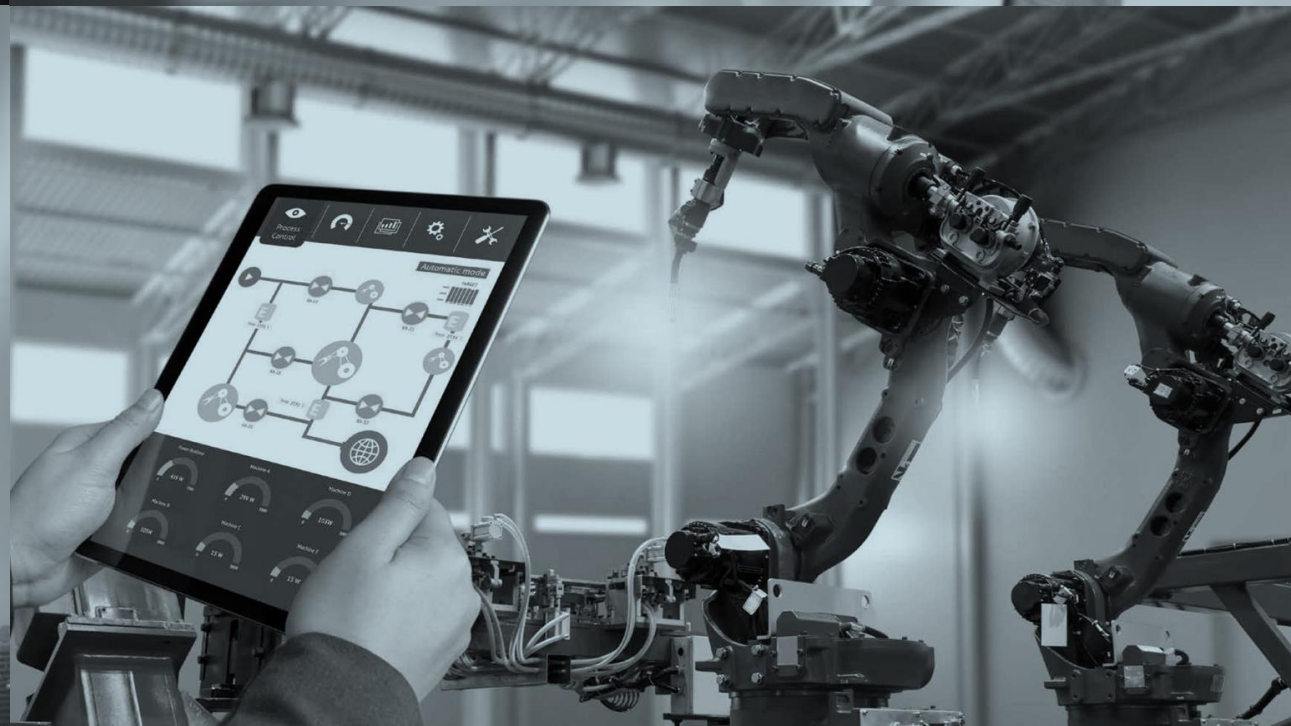
*An **Hypervisor or VMM** is to an OS, as an OS is to a process.*

- **Main functions**
  - Resource Management
  - Abstraction
  - Protection / Isolation
- **The hypervisor provides a Virtual Machine (VM) abstraction for guest OSes**
- **Well-established**
  - Servers (load balancing, power management)
  - Desktops (cross-platform, systems development)

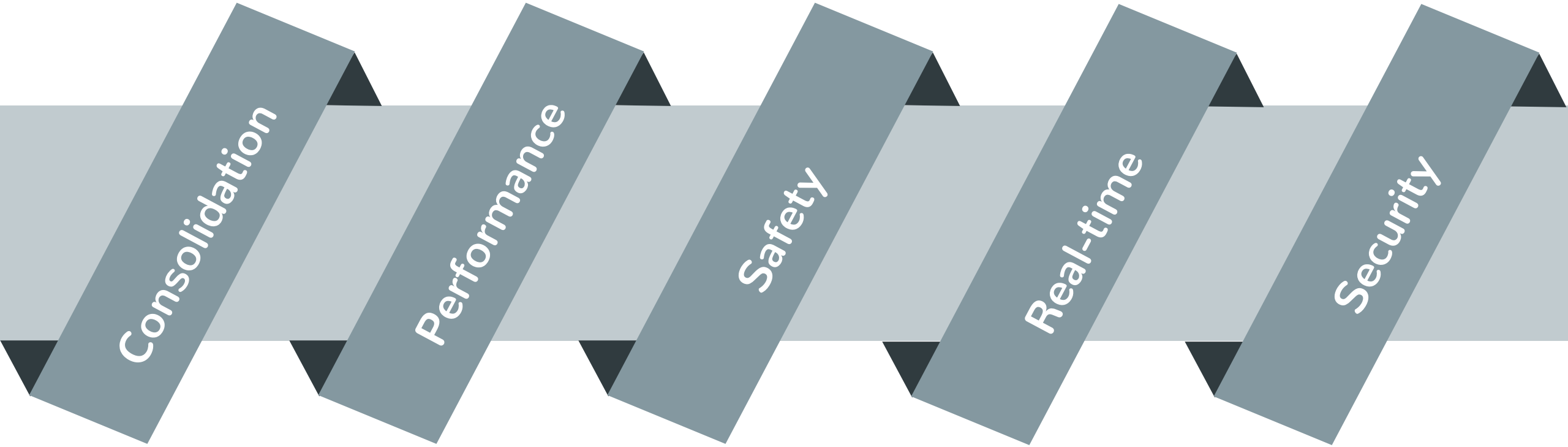



Self-Driving


MEDICINE
HEALTH
TREATMENT
DOCTOR
SURVEY
RECIPE
MEDICINE

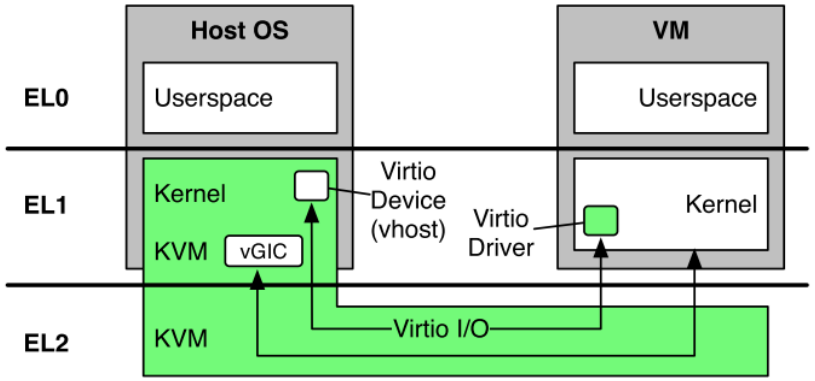

Automatic mode

Embedded Virtualization



Hypervisors Spectrum

01



Source: C. Dall, "The Design, Implementation, and Evaluation of Software and Architectural Support for ARM Virtualization"

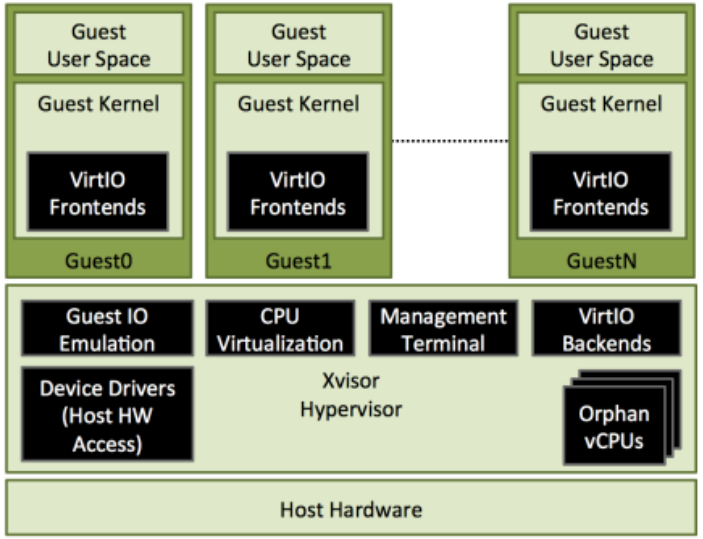
"Traditional" Hypervisors

- Large code base / TCB (Linux)
- Fully-featured (comm, networking, drivers)
- High-overhead I/O (emulation, paravirtualization)
- Cloud, server, and mobile
- **KVM, Xen**

02

- Resonable code base / TCB (100K-10K SLoC)
- Soft real-time
- Scheduling and N:1 virtual-to-physical CPU mapping
- Generic embedded systems
- **XVisor, ACRN**

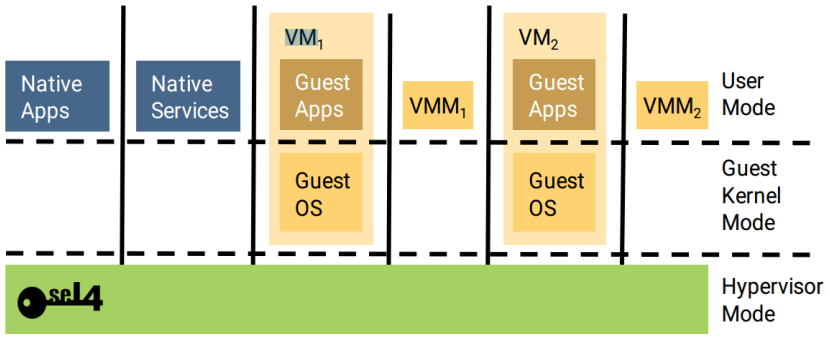
Embedded Hypervisors



Source: *A. Patel et al. "Embedded Hypervisor Xvisor: A comparative analysis"*

Hypervisors Spectrum

03



Source: *Gernot Heiser, "The seL4® Microkernel: An Introduction"*

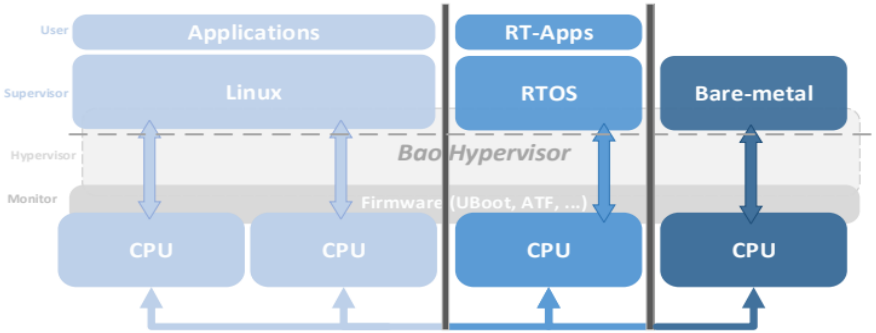
Microkernel Hypervisors

- Secure by design (formal verification)
- Capabilities and User-level VMMs
- Performance / Interrupt latency / Complexity
- Mobile, embedded, and mixed-criticality systems
- **seL4, NOVA**

- Small code base / TCB (5K-10K SLoC)
- Strong isolation & real-time
- Ineficiente resource usage
- Mixed-criticality systems
- **Bao, Jailhouse, Xen Dom0-less**

Static Partitioning Hypervisors

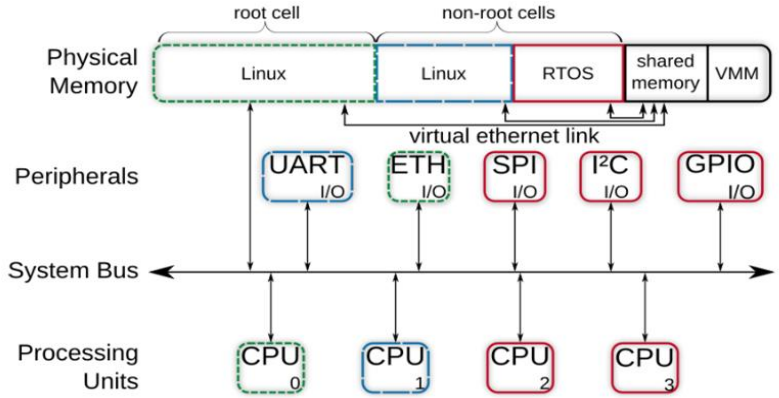
04



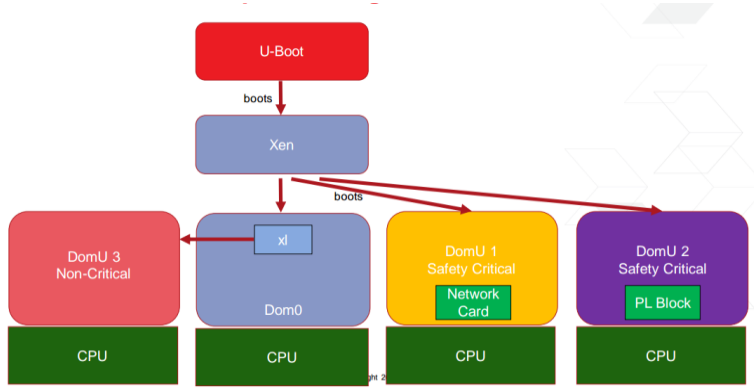
Source: *J. Martins et al. "Bao: A Lightweight Static Partitioning Hypervisor for Modern Multi-Core Embedded Systems"*

Static Partitioning Virtualization

- **Main Goals**
  - Consolidation
  - Strong isolation and real-time guarantees
  - Minimal code base
- **Static resource assignment**
  - 1:1 virtual-to-physical CPU mapping
  - No memory
  - Device passthrough
  - Hardware interrupts
- **Reliance on HW virtualization**



Source: *R. Ramsaur et al. "Look Mum, no VM Exits! (Almost)"*



Source: *S. Stabellini, "Static Partitioning with Xen"*

Bao Hypervisor

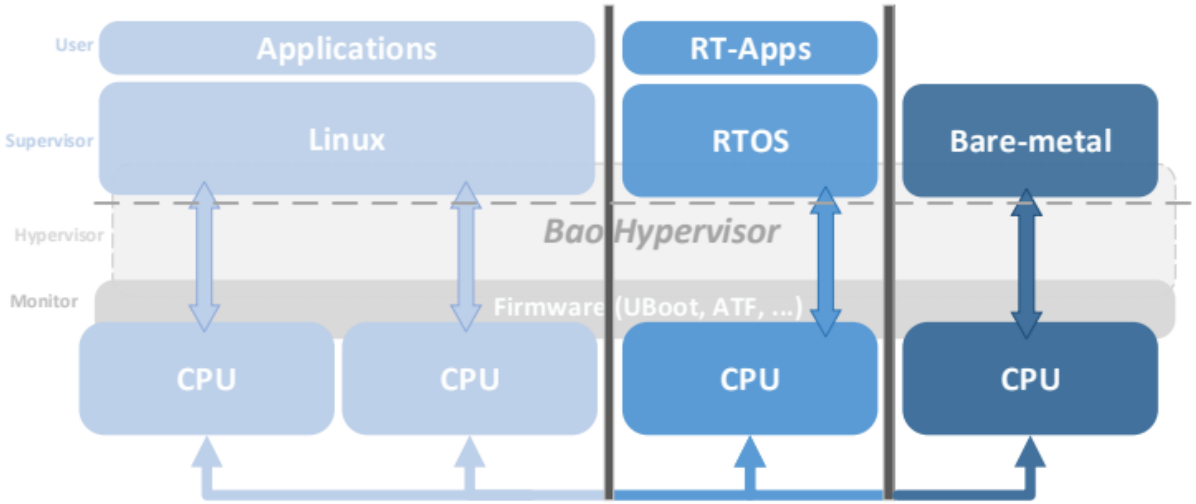
Overview and Highlights

Bao Hypervisor

- Type-1 hypervisor, static partitioning architecture
- Mixed-criticality systems with strong real-time and security requirements
- https://github.com/bao-project/bao-hypervisor

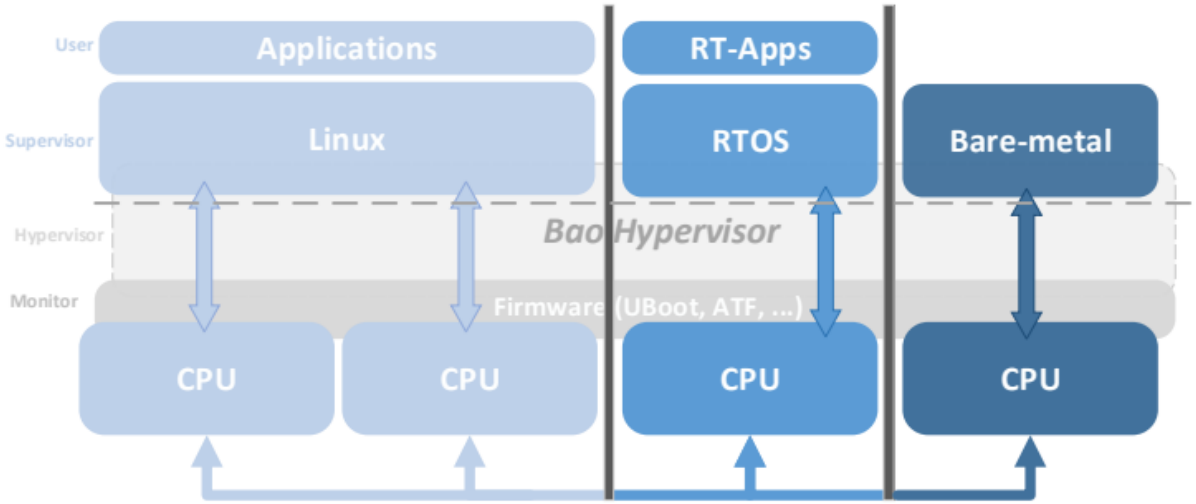
Bao Hypevisor

- **Type-1 / Bare-metal**
- **Static Partitioning:**
  - 1:1 vCPU-to-pCPU mapping
  - Static memory assignment
  - Device Pass-through
  - Hardware interrupts
- **Inter-VM communication:**
  - Shared memory
  - Notifications (Inter-VM interrupts w/ access-control)
- **Dependencies**:
  - No external libraries
  - No privileged VMs/Oss
  - Simple drivers for UART log



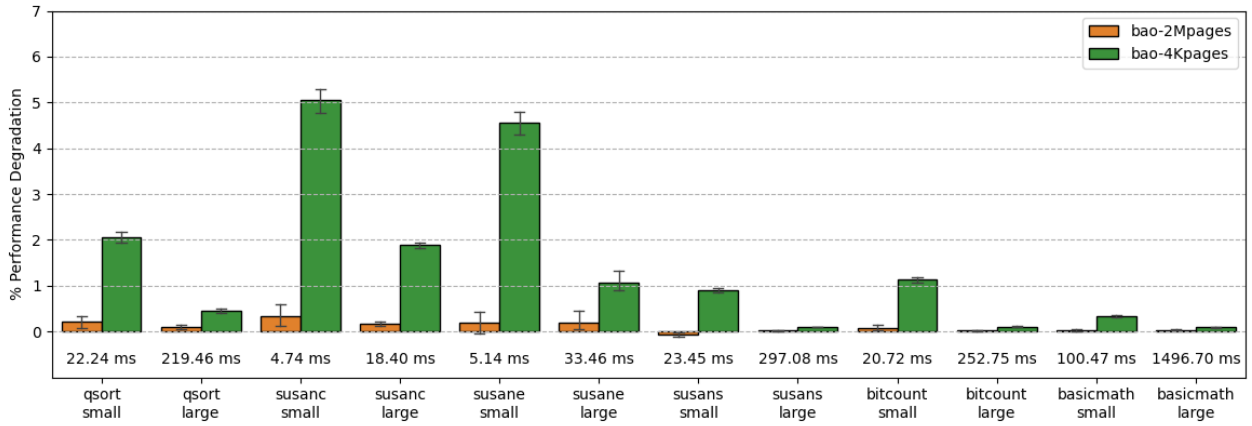
Bao Hypevisor

- **Hardware-assisted:**
  - 2nd-stage translation
  - Interrupt virtualization support
  - IOMMU
- **Internal Isolation:**
  - Private CPU mappings
  - No mapping of VM memory
- **Superpages:**
  - Reduced TLB pressure
  - Reduced page-table memory
- **Cache Coloring (Partitioning)**:
  - Avoid interference on LLC
  - Guests and hypervisor



Reduced

Performance Overhead



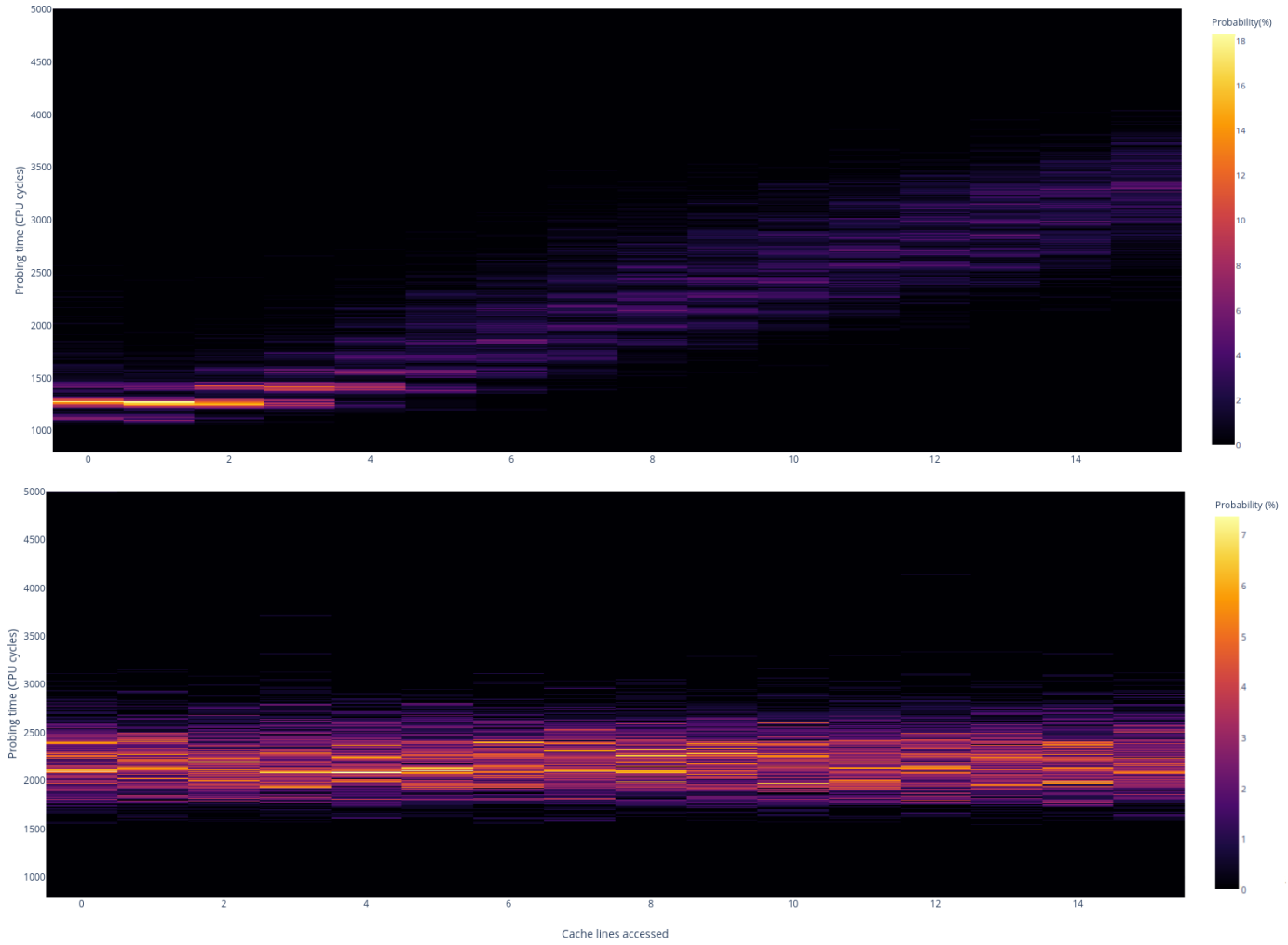
Minimal

Trusted Computing Base

~8K SLoC

Cache Side-Channel

Resistance via Cache Coloring



Bao Support

- **Architectures:**
  - Armv7-A, Armv8-A, Armv8-R
  - RISC-V
- **Platform Highlight:**
  - Zynq US+ (ZCUx and Ultra96)
  - NXP i.MX8
  - Nvidia Tegra TX2
  - QEMU
  - Rocket @ ZCU
  - CVA6 @ Genesys2/VCU118
  - +++
- **Firmware:**
  - Arm Trusted Firmware (PSCI) on Arm
  - Supervisor Binary Interface (SBI) on RISC-V
- **Guests:**
  - Bare-metal
  - Linux / Android
  - RTOSs (Zephyr, FreeRTOS, Erika)

Armv8-A

Beginning of a Journey

Arch/Platform Support

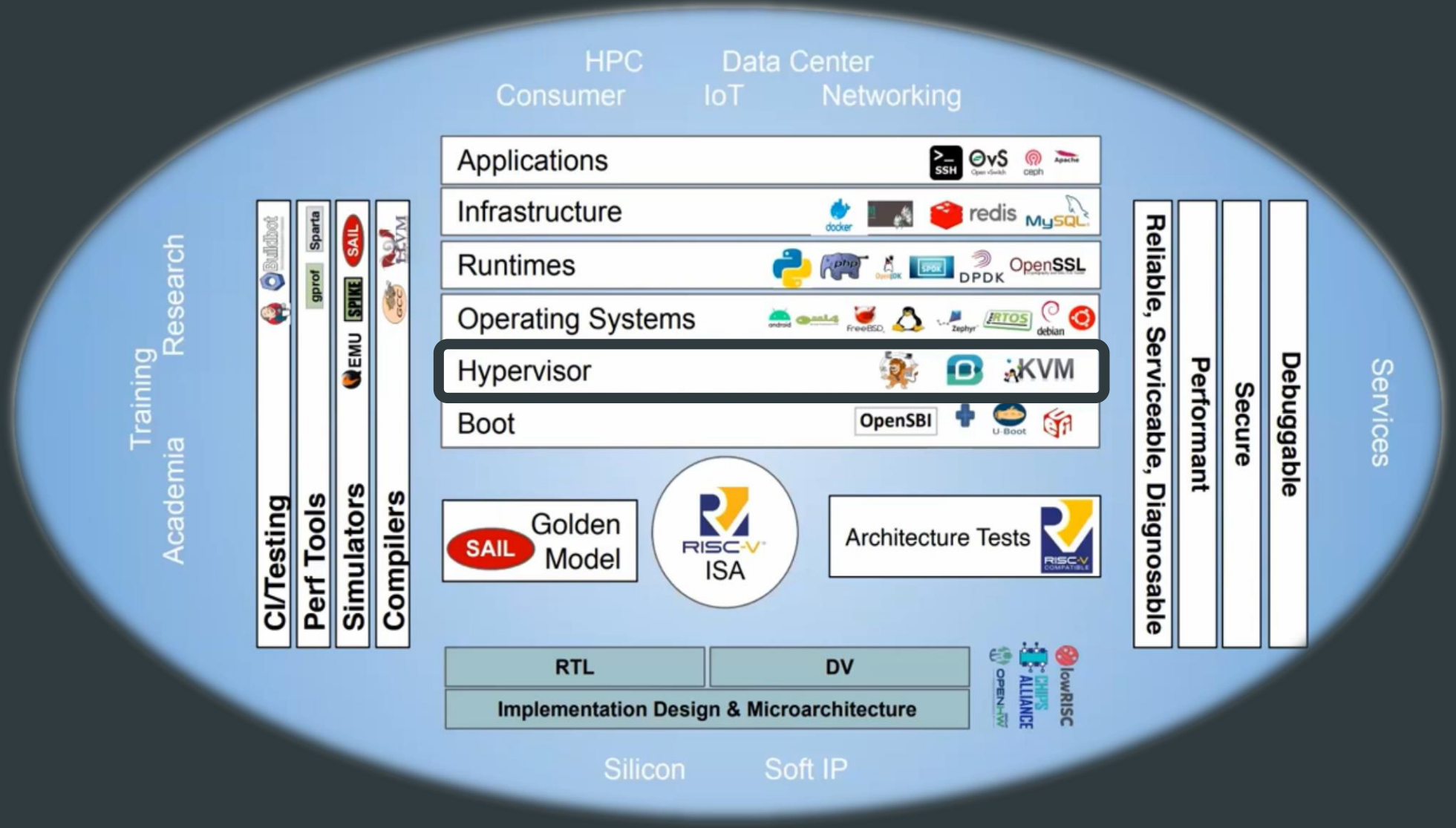
- 32- and 64-bit support
- QEMU support
- Arm Fast Model
- +8 HW platforms

Interrupt Support

- Generic Interrupt Controller (GIC) v2
- Generic Interrupt Controller (GIC) v3

Platform-level Isolation

- System Memory Management Unit (SMMU) v2
- System Memory Management Unit (SMMU) v3 - WiP
- Platform-specific System Level Controllers



Source: *Philipp Tomsich & Mark Himelstein, "Maturing the RISC-V Ecosystem: From Technology to Product", RISCV-Summit https://www.youtube.com/watch?v=LY98foD0SkY*

WHITE PAPER

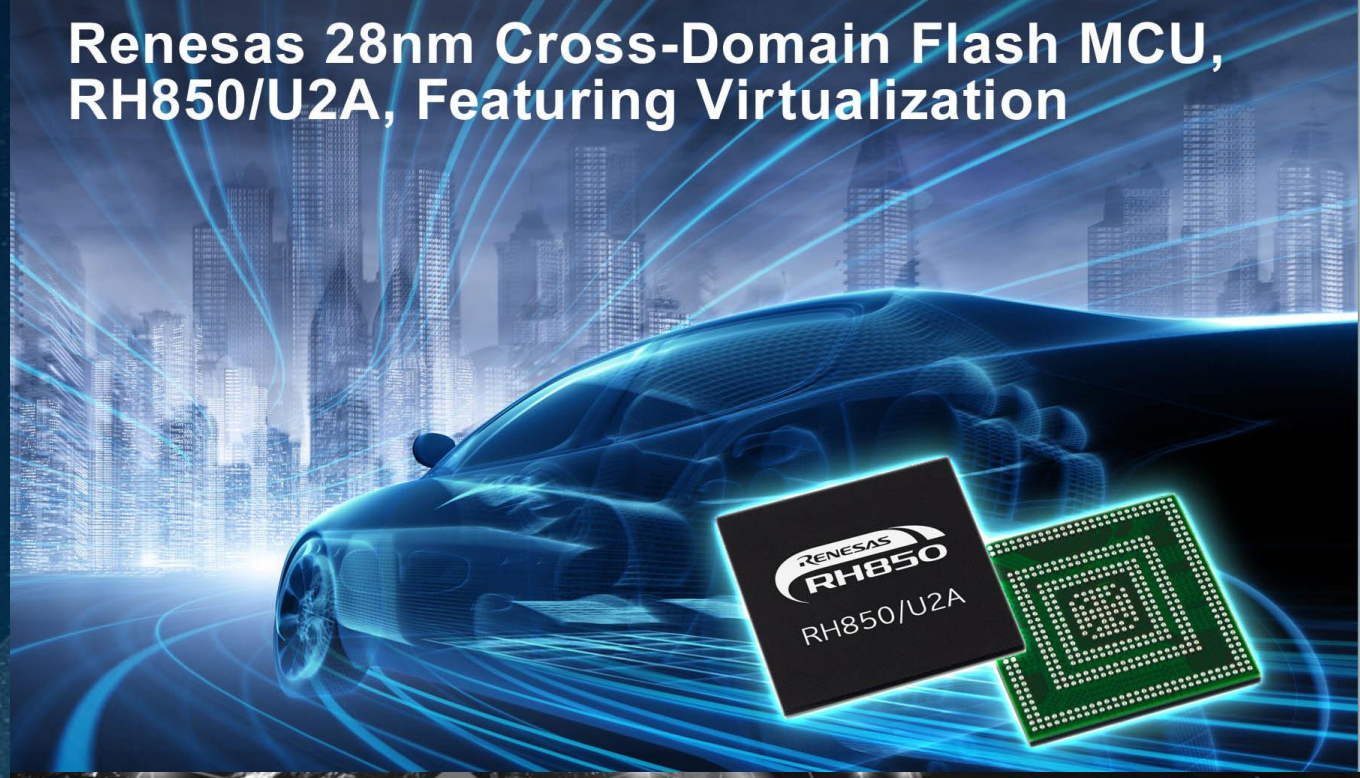
arm

ETAS

Best Practices for Armv8-R Cortex-R52+ Software Consolidation

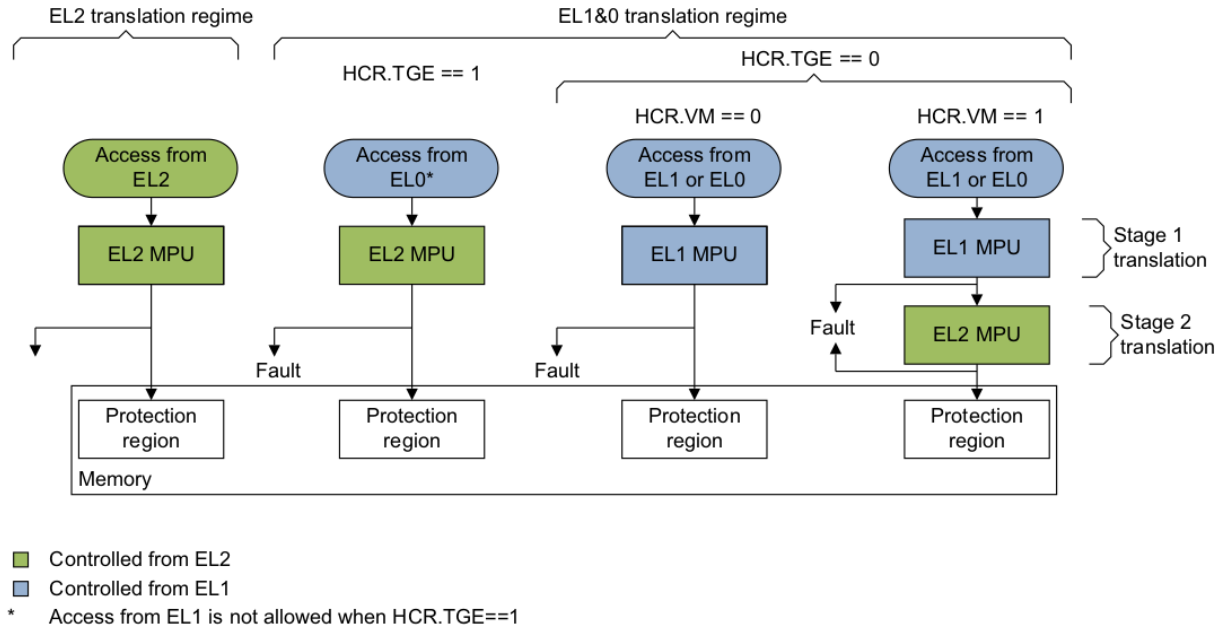
Dr Paul Austin, Principal Software Engineer, ETAS
Dr Andrew Coombes, Senior Product Manager, ETAS
Paul Hughes, Lead System Architect and Distinguished Engineer ATG, Arm
James Scobie, Director Automotive Product Management, Arm
Bernhard Rill, Director Automotive Partnerships EMEA, Arm

Renesas 28nm Cross-Domain Flash MCU, RH850/U2A, Featuring Virtualization



Bao @ Armv8-R

- **MPU-based:**
  - Double stage MPU
  - 2nd-stage MPU shared by guest and hypervisor
  - Abstraction layer on memory management
  - Modification of data layout
  - Challenge due to reduced set of MPU entries
  - For AArch64 allows for 1st-stage MMU
- **32-bit support:**
  - First targets available for Armv8-R are AArch32
- **Firmware functionality:**
  - No EL3 available
  - System timer initialization
  - Platform-specific initialization
    - System clocks
    - Core power-up



Bao & TEE

Overview and Support

1. Virtualizing the Secure World is costly

- Ring Depriviliging
- Shadow Page-table
- Trap-and-Emulate (High-frequency)

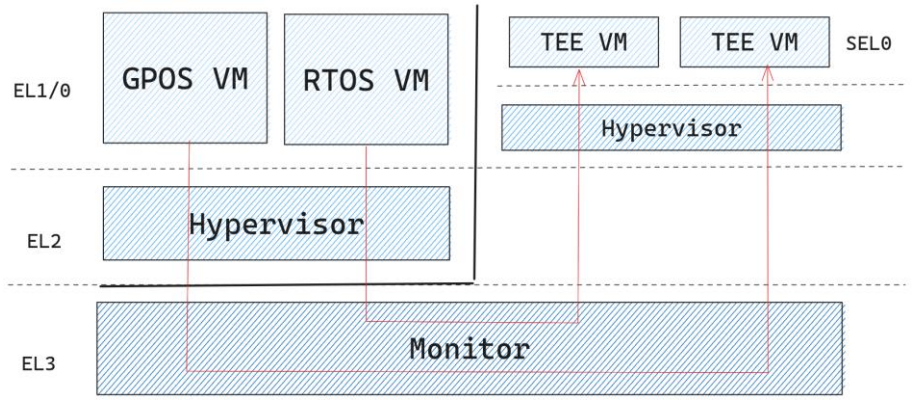
2. Shared Trusted OS / TEE for all VMs

- Sharing is not caring!
- OP-TEE @ Xen supports separate VM contexts

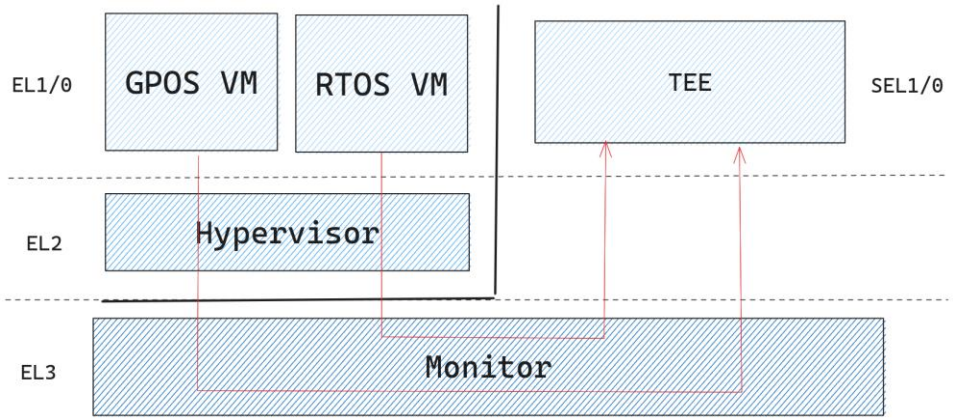
3. Secure “Hypervisor” in the Secure World

- Only on ArmvA-8.4

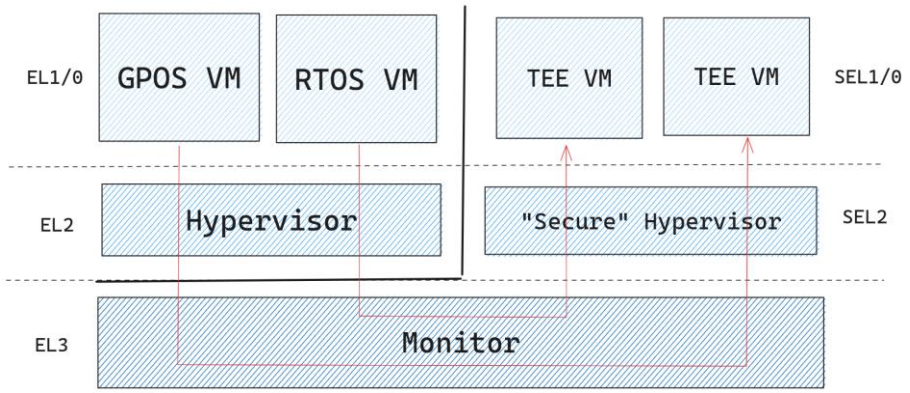
1



2

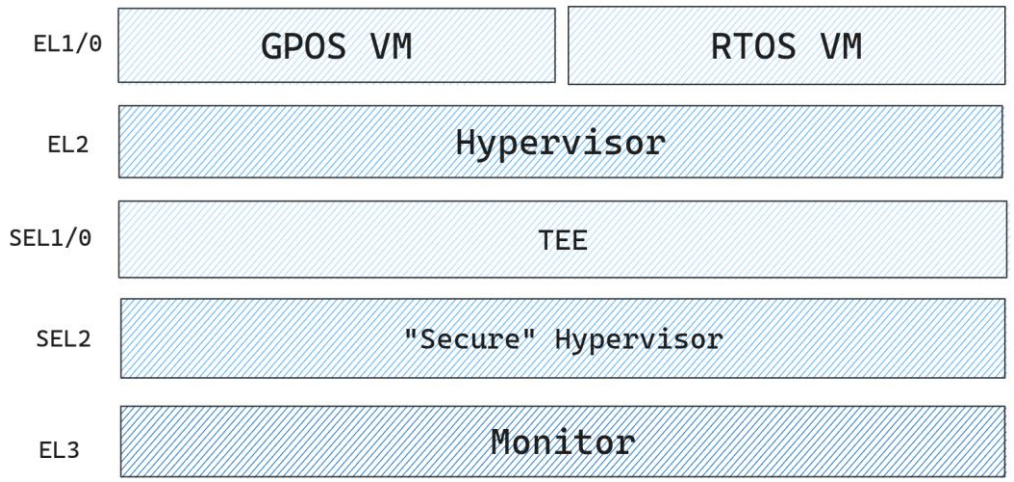


3



Bao TEE Motivation

- **Virtualization-based TEEs as a trend:**
  - pKVM, Arm Realms, RISC-V AP-TEE, AMD SEV, Intel TDX
- **Static partitioning is too rigid:**
  - sharing of resources (cores) across partitions
  - Increased layers complexity and code base hampers certification
- **Not flexible:**
  - Only TrustZone model supported
  - Per-process SGX-like enclaves not supported
- **Not deployable/portable:**
  - SEL2 not present in most used Arm platforms
  - Cortex-R does not have secure world
  - Dual-hypervisor not fully yet supported on RISC-V
- **Bao can take the place of security monitor:**
  - But... dedicating a core to a single TEE is wasteful!



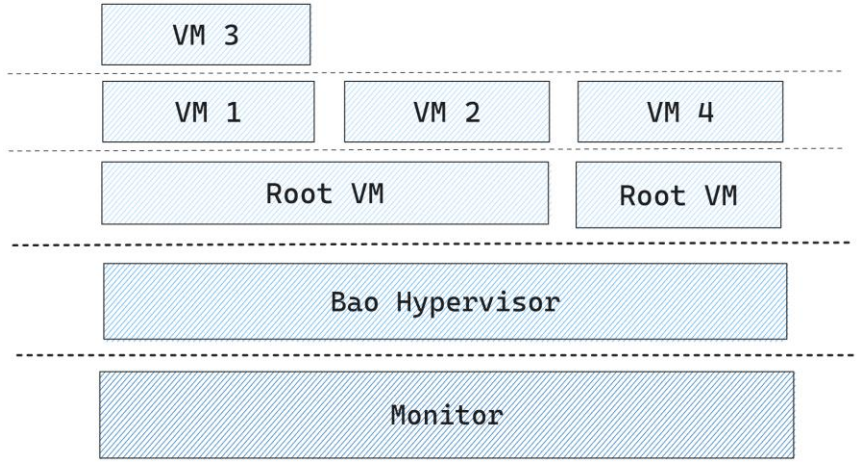
Ideally we would like separate TEE instances for each VM!

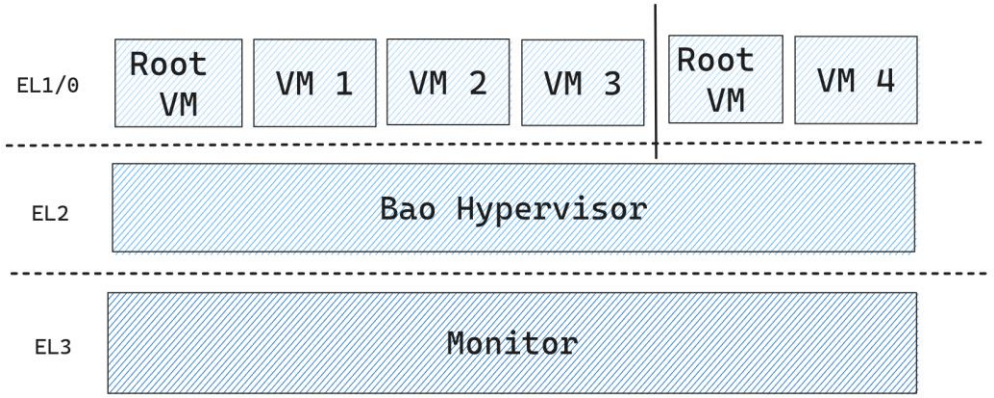
Design Goal

Extend static partitioning to allow multiple vCPUs per pCPU
with simple yet flexible mechanism

Bao VM Stacking

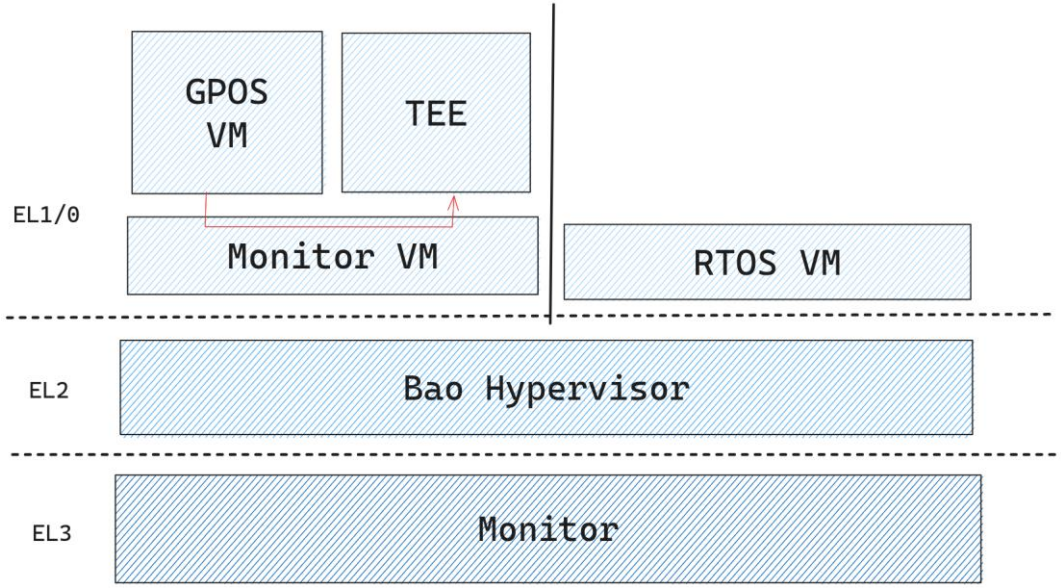
1. Per-partition VM hierarchy defined at configuration time;
2. VMs can schedule/invoke direct children via hypercalls;
3. VMs can (optionally) control state children register state;
4. Unknown VM exceptions are forward to parent;
5. VMs can preempt any descendent (e.g. through interrupts);
6. Effectively emulate unlimited privilege levels;
7. “Lightweight, para-virtualized nested virtualization”





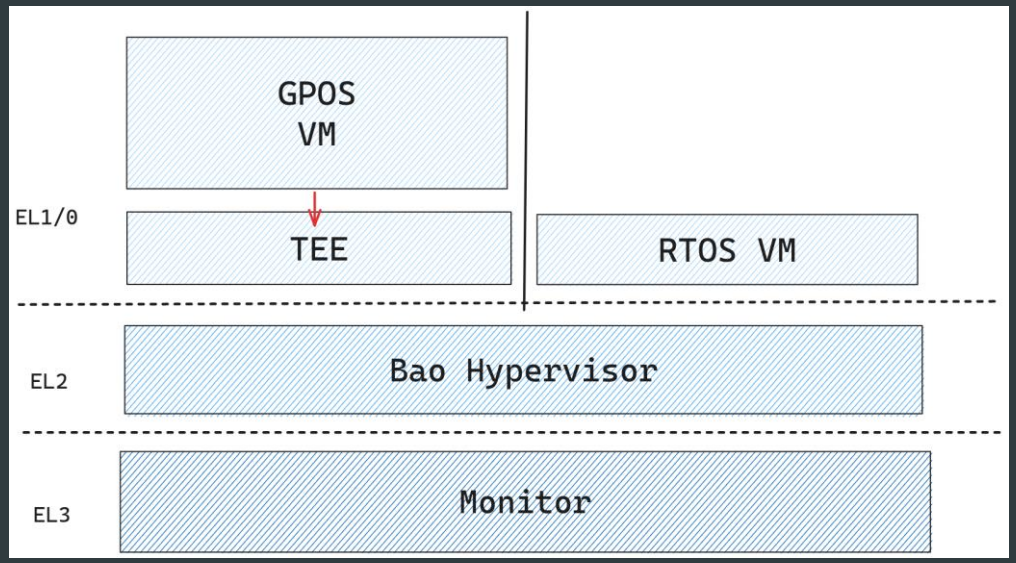
OS TEE Model 1

Dedicated Monitor VM



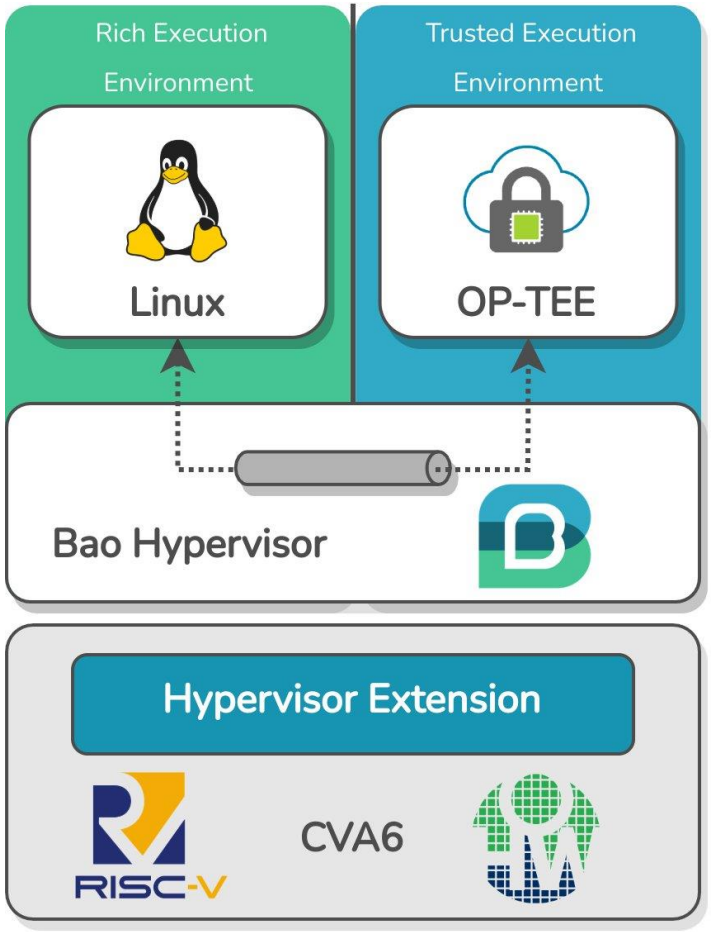
OS TEE Model 2

TEE VM as root



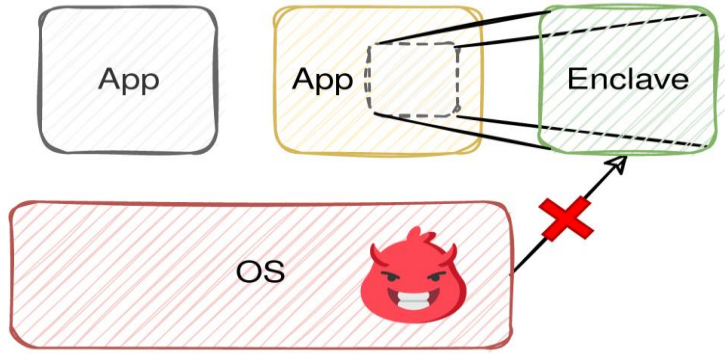
Bao & OP-TEE

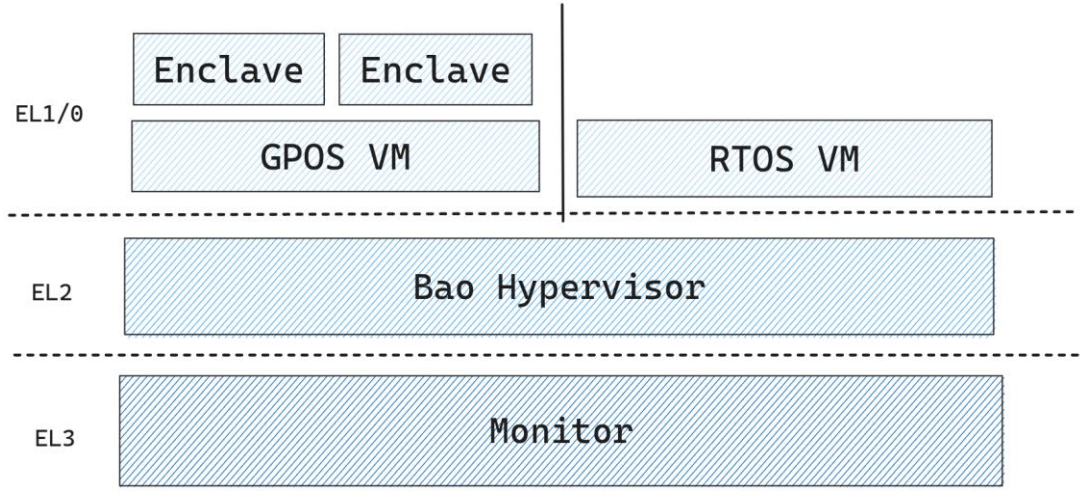
- **"Secure world" approach is flawed:**
  - Security Problems!
- **Bao implements OS TEE Model 2:**
  - TEE VM as root
- **OP-TEE restricted view of "normal world":**
  - Motivation: "ReZone" USENIX Security 22 Paper
- **Only statically defined shared memory buffers:**
  - *CFG_CORE_RESERVED_SHM*
- **Non-invasive implementation:**
  - Little to none modifications to OP-TEE source code
- **Scalable for Arm and RISC-V**
  - Same setup running in both target Architectures



Bao-Enclave

- **SGX-like abstraction:**
  - Process Enclave
- **VM Stacking:**
  - Enclave is a children VM
- **Dynamic VM Creation**
  - Resource Donation
- **Children VM Management:**
  - Permissions
  - Manage CPU state
- **Ad-hoc Runtime and ABI**
  - Stack state passed via shared memory
  - Unmodified SGX applications on Arm





Work in Progress

01

API/ABI & Abstractions

Extra hypercalls (register state different models)
Register sanitizing by hypervisor

02

Interrupt/exception model

Enclave Preemption Notification
DoS protection

03

Enclave Runtime

Unikernel
RPC protocols

04

Enclaves as Lightweight VMs

Reduce State
No interrupt Controller

05

Secure Hard-wired Resources

Secure world shim to perform accesses

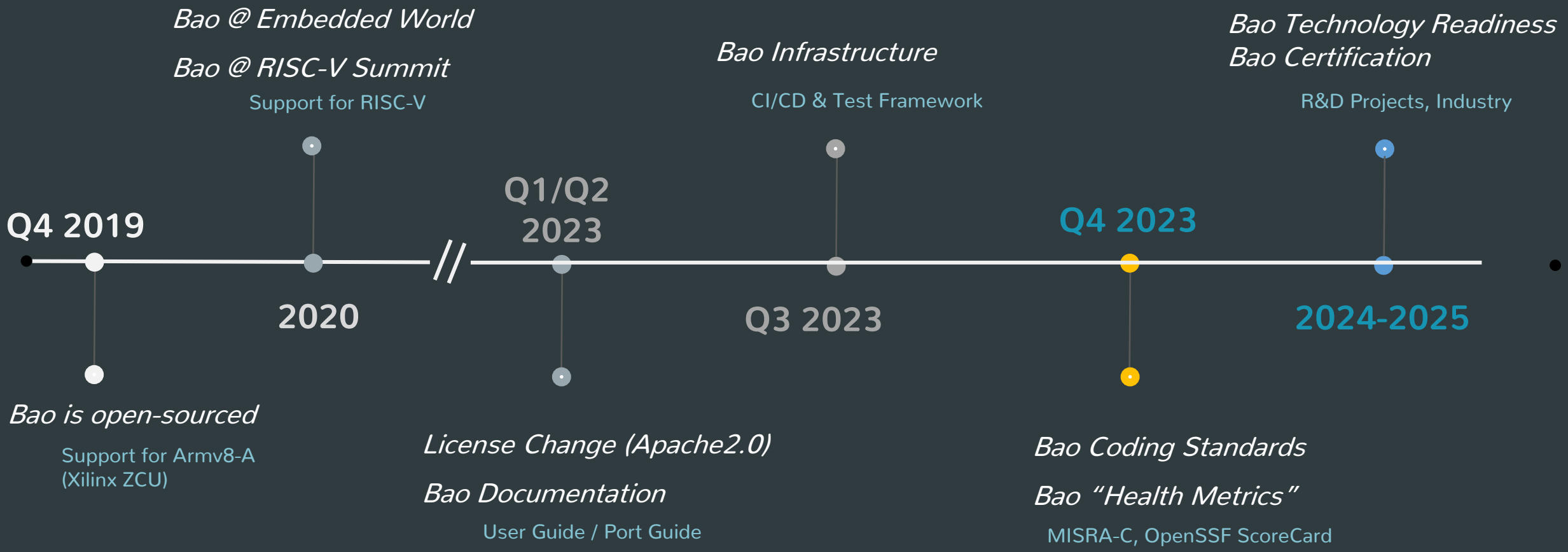
06

Device sharing

Different TEEs instances accessing the same device
Para-virtualization approach (e.g. VirtIO)

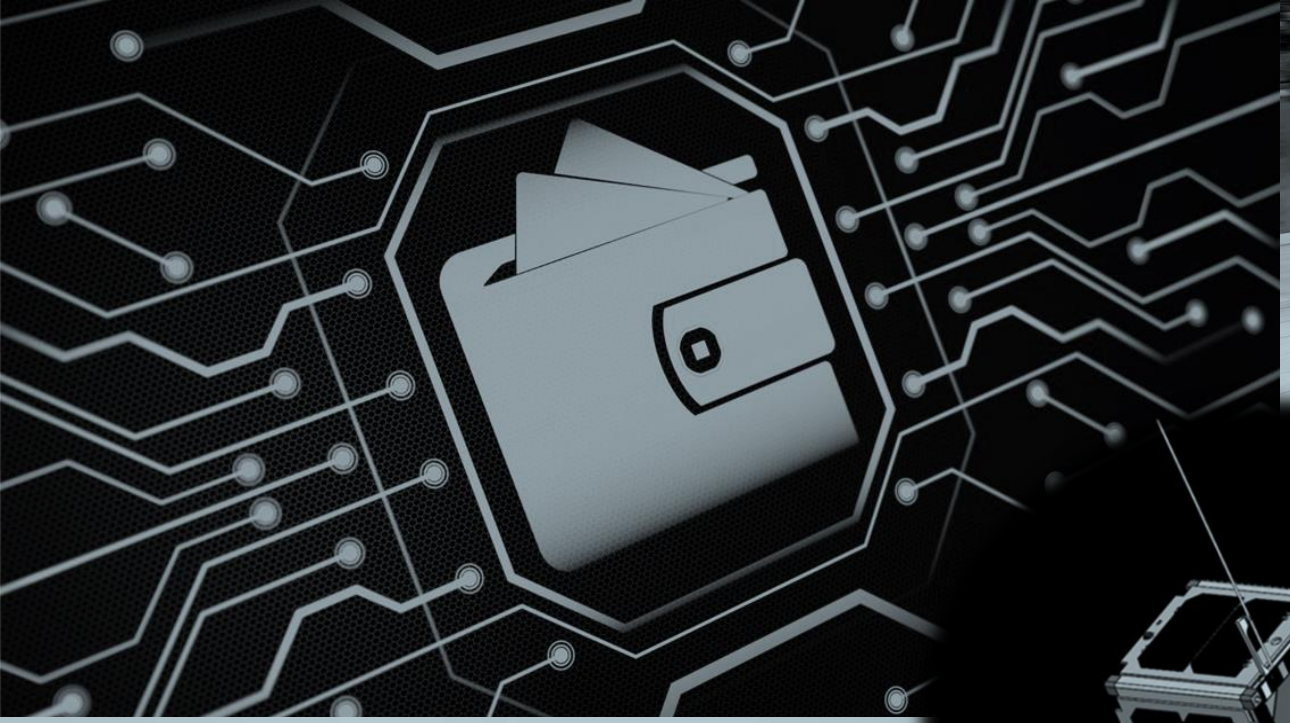
Bao Roadmap

Roadmap at a Glance



Bao Project

Short- to Mid-Term Roadmap




Continental


IoT

Follow Bao!!!

bao-project / bao-hypervisor

Public

Code Issues 2 Pull requests 5 Discussions Actions Security Insights

main 7 branches 2 tags Go to file Code

josecm update: add @JorgeMVP to contributors ... 37f1820 5 days ago 372 commits

.github	ci(codeowners): add CODEOWNERS file	2 months ago
configs/example	license: change from GPL-2.0 to Apache-2.0	4 months ago
scripts	feat: allow defining cpu master statically	last month
src	aarch32: boot use lda, stl to sync up between multiple PEs at init	5 days ago
.clang-format	Initial Commit	4 years ago
.gitignore	add builtin configuration build option	3 years ago
CONTRIBUTORS	update: add @JorgeMVP to contributors	5 days ago
LICENSE	license: change from GPL-2.0 to Apache-2.0	4 months ago
Makefile	build: treat linker warnings as errors	3 months ago
README.md	update (Readme): changed platform to S32Z/E	2 months ago

README.md

Bao - a lightweight static partitioning hypervisor

Introduction

Bao (from Mandarin Chinese "bǎohù", meaning "to protect") is a lightweight, open-source embedded hypervisor which aims at providing strong isolation and real-time guarantees. Bao provides a minimal, from-scratch implementation of the partitioning hypervisor architecture.

About

Bao, a Lightweight Static Partitioning Hypervisor

security arm embedded virtualization hypervisor static safety partitioning mpu mmu armv8 risc-v cortex-a cortex-r mixed-critcality

Readme
Apache-2.0 license
190 stars
28 watching
66 forks
Report repository

Releases

2 tags

Contributors 11

Languages

master 4 branches 0 tags Go to file Code

D3boker1 and josecm fix(platforms/x/README.md): fix typos 8cac41d last month 65 commits

demos	fix(demos/linux+zephyr): fix address in step-by-step guide	last month
guests	fix(guests/zephyr): fix multiple issues with step-by-step guide	2 months ago
platforms	fix(platforms/x/README.md): fix typos	last month
.gitignore	initial commit	2 years ago
COPYING	initial commit	2 years ago
Makefile	fix: add missing bin wrkdir	2 months ago
README.md	fix: add missing fvp-r-aarch32 README symbolic link	last month

README.md

Bao Hypervisor Demo Guide

This tutorial provides a step-by-step guide on how to run different demo configurations of the Bao hypervisor featuring multiple guest operating systems and targeting several supported platforms. The available demos are:

- Single-guest Baremetal
- Dual-guest Linux+FreeRTOS
- Dual-Guest Linux+Zephyr
- Dual-Guest Zephyr+Baremetal

NOTE

This tutorial assumes you are running a standard Linux distro (e.g. Debian) and using bash.

If you have any doubts, questions, feedback, or suggestions regarding this guide, please raise an issue in

About

A guide on how to build and use a set of Bao guest configurations for various platforms

Readme
View license
18 stars
5 watching
15 forks
Report repository

Releases

No releases published

Packages

No packages published

Contributors 6

Languages

Makefile 93.2% Assembly 6.8%

THANK YOU!

jose.martins@bao-project.org

LinkedIn - https://www.linkedin.com/in/josecmar/
Twitter - https://twitter.com/josecarmartins
Github - https://github.com/josecm

sandro@bao-project.org

LinkedIn - https://www.linkedin.com/in/sandro2pinto
Twitter - https://twitter.com/sandro2pinto
Github - https://github.com/sandro2pinto/

Q&A