How to choose the OS in a multi-core Safety Certification context?

Olivier Charrier – Functional Safety Specialist

Agenda

1 – **Why Multi-core?**

2 – **Regulation**

3 – **Multi-core Interferences**

4 – **Which OS Type?**

WNDRVR

Why Multi-Core?

Consolidation of legacy single-core systems

- Expected Benefits
  - Solve Hardware Obsolescence
  - Reduces Space, Weight, Power and Cabling (SWaP-C),
  - Reduces BOM costs
  - Move to an IMA (Integrated Modular Avionics) approach

    To aggregate applications with different lifecycle, for example

- Possible Constraints to be Managed for Consolidation
  - Legacy OS and associated applications coming from different equipment
  - Can be heterogenous (OS, OS version, Different Standard Conformance, etc.)
  - Mixed-Criticality may exist (Partitioning may then need to be demonstrated)

Necessity for Computing Power

- Expected Benefits
  - More computing power than a single core
  - Provision for growth

- Use Cases
  - Image Processing
  - AI/ML (Artificial Intelligence / Machine Learning) – mainly Linux based
  - Predictable maintenance capabilities
  - …

Regulation Framework

FAA and EASA Funded Research

- FAA-funded project "Microprocessor Evaluations for Safety-Critical, Real-Time Applications: Authority for Expenditure No. 43“ (AR-06/34, AR-08/14, AR-08/55, AR-10/21, AR-11/05).
  - http://www.faa.gov/aircraft/air_cert/design_approvals/air_software/media/06-34_MicroprocessorEval.pdf
  - http://www.faa.gov/aircraft/air_cert/design_approvals/air_software/media/11-5.pdf
  - http://www.faa.gov/aircraft/air_cert/design_approvals/air_software/media/AR-10-21.pdf
- EASA-funded project 2011.C31 “MULCORS” Report.
  - http://easa.europa.eu/safety-and-research/research-projects/docs/large-aeroplanes/MULCORS_Final_Study_Report_EASA.6-2011.pdf

MULTI-CORE CERTIFICATION GUIDANCE

Multi-core certification guidance

- EASA CRI and FAA CAST-32 paper (2014)
- EASA CRI and FAA CAST-32A paper (2016)
- FAA AC (upcoming) / EASA AMC 20-193 (2022)

AC/AMC 20-193 aspects covered are

- Software Architecture Planning
- Planning and Setting of MCP Resources (configuration)
- Interference Channels and Resource Usage
- Software Verification
- Error Detection and Handling, and Safety Nets
- Reporting of Compliance with the Objectives

WNDRVR

AMC 20-193 Objectives

Objectives		Goals
MCP Planning 1	MCP Planning 2	Planning PSAC & PHAC
MCP Resource Usage 1	~~MCP Resource Usage 2~~ (covered by AMC 20-152A)	Configuration Settings
MCP Resource Usage 3	MCP Resource Usage 4	Interference Channels Resource Usage
MCP Software 1	MCP Software 2	Software Verification
	MCP Error Handling 1	Safety Net
	MCP Accomplishment Summary	Showing Compliance

DAL C and above (blue)

DAL B and above (grey)

Produced in conjunction with ConsuNova Inc

CAST-32A Objectives: Software Verification

RAPITA SYSTEMS A DANLAW COMPANY

WIND

MCP Software 1

MCP Software 2

Software Verification

WCET Analysis of all hosted SW Components

+ SW components function correctly.
+ SW components have sufficient time to complete their execution when all the hosted SW is executing in the intended final configuration.

A) MCP Platforms With Robust Partitioning

+ MCP Platform provides both Robust Resource and Robust Time Partitioning.
+ May verify applications separately on the MCP and determine their WCETs separately.

B) All Other MCP Platforms

+ Interference Mitigated/Precluded by design
  - WCET of a SW component may be determined separately.
+ Interference NOT Mitigated/Avoided
  - WCET should be confirmed by test on the target MCP with ALL SW components executing in the intended final configuration.

Multi-core Interferences

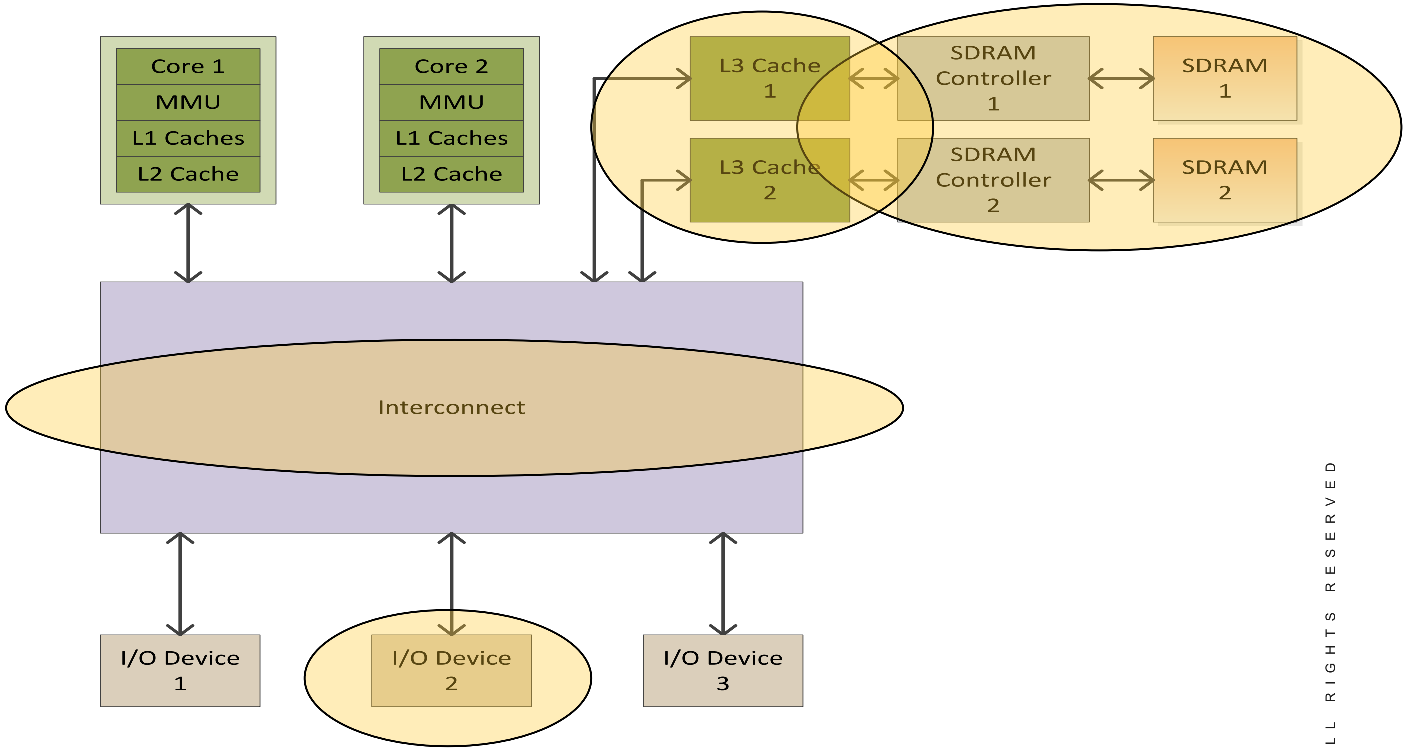
Multicore Hardware Safety Concerns

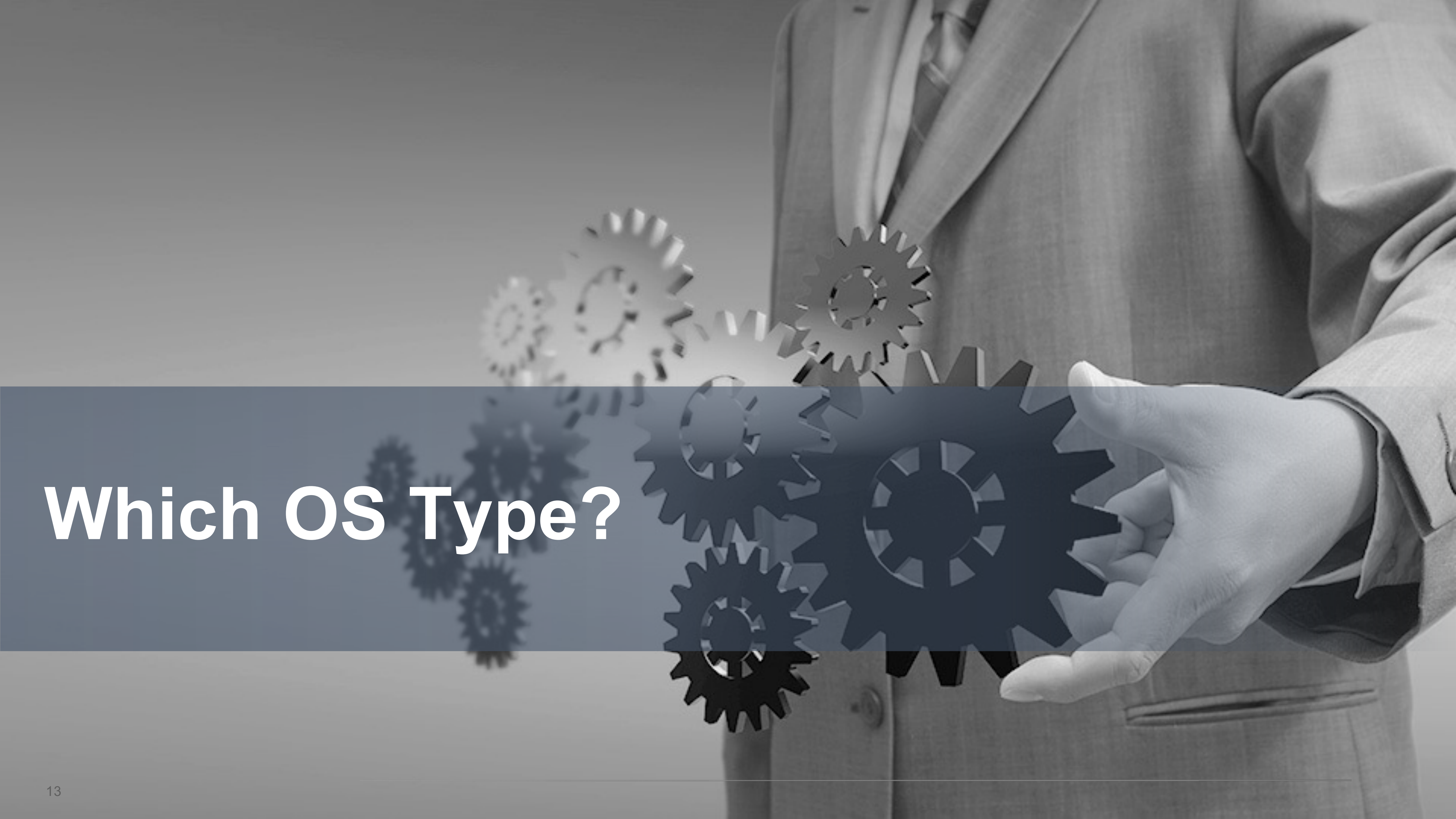
Hardware Shared Resources

- Caches
- Memory controllers
- Interconnect (coherency) module
- I/O Devices

Possible Mitigations

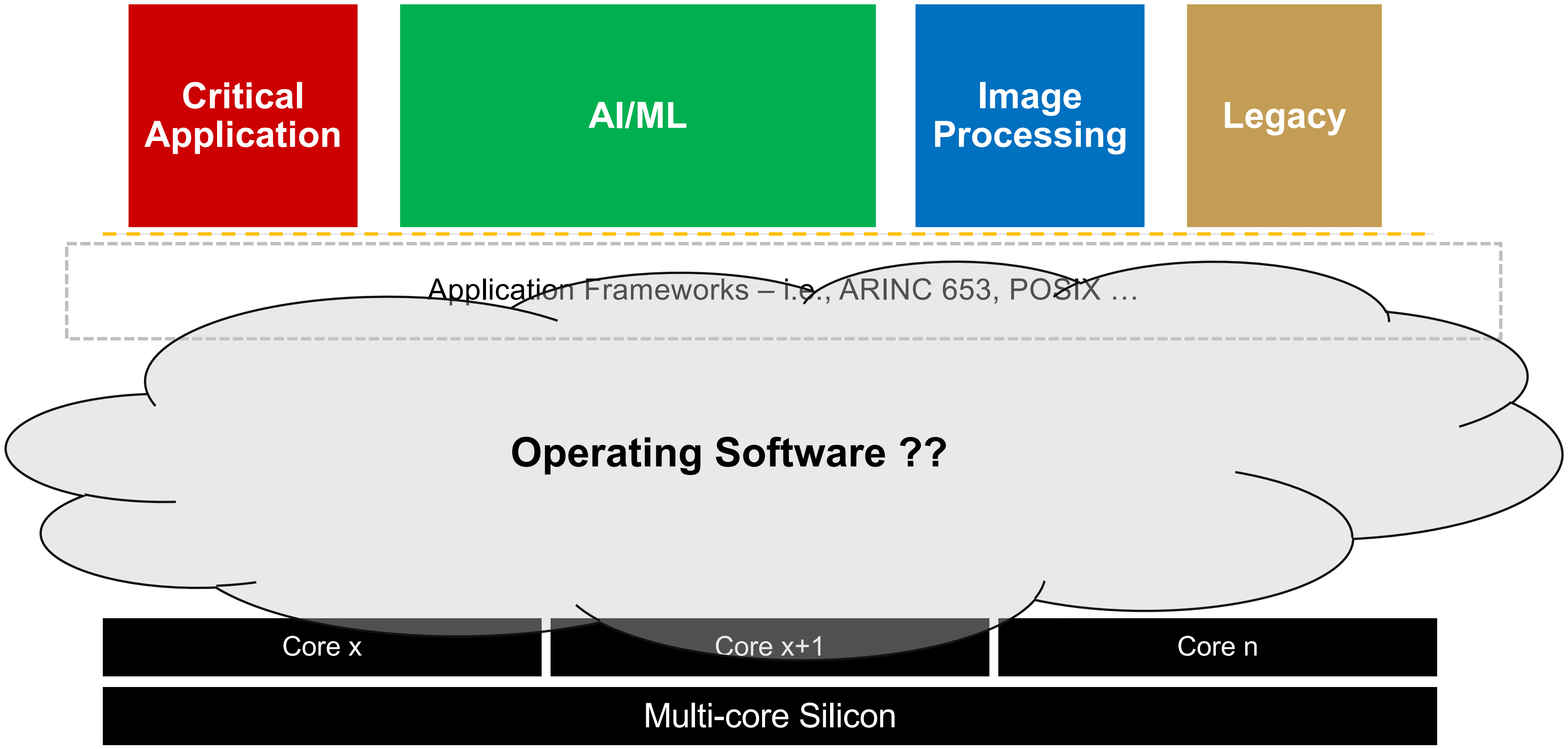
- Avoid sharing, completely
- Avoid sharing at the same time
- Share at the same time
  (Assuming WCET computation is achievable)





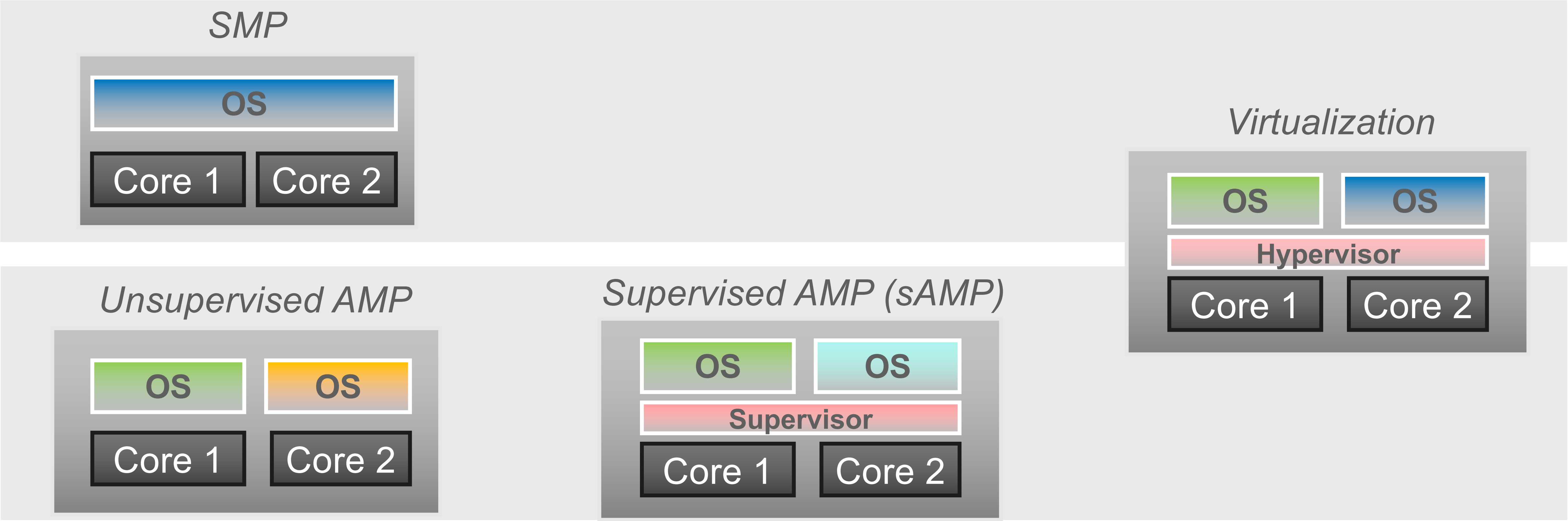
Which OS Type?

Start by a Preliminary Software Design



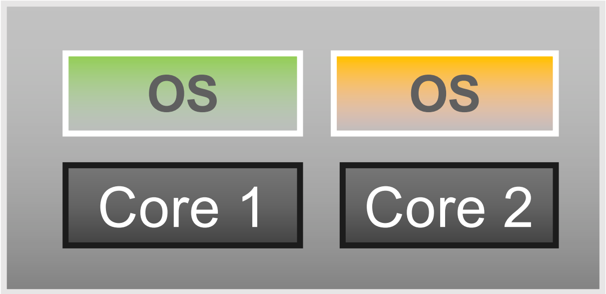
WNDRVR

Look for the Right Operating Software Model



Unsupervised AMP

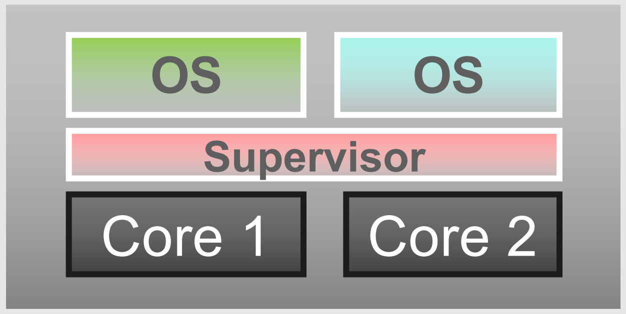
Unsupervised AMP



- Use case support
  - All use cases can be supported (because it can include different OSes)
  - But it all depend on hardware capabilities
- Interference Mitigation
  - Pushed on the hardware itself
  - ***Avoid resources sharing unless interferences impact can be assessed within WCET***
  - ***Easier with multiple memory controllers and/or “clusters of cores”***
  - ***Mitigation can be put in place by hardware configuration*** via the bootloader or by a “Safety Island” micro-controller

Supervised AMP

Supervised AMP (sAMP)



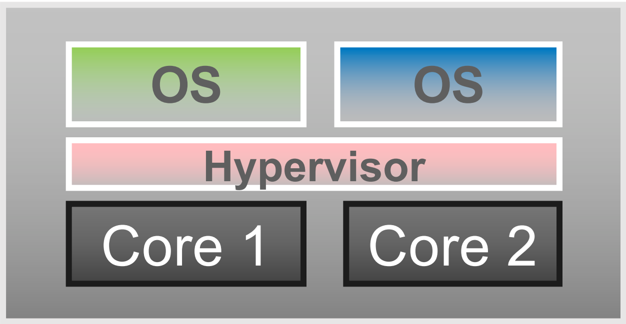
- Use case support
  - All use cases can be supported (because it can include different OSes)
  - But it all depend on hardware capabilities

- Interference Mitigation
  - Pushed on the hardware itself
  - ***Avoid resources sharing unless interferences impact can be assessed within WCET***
  - ***Easier with multiple memory controllers and/or "clusters of cores"***
  - ***Mitigation can be put in place by hardware configuration*** via the ***Supervisor***
  - ***The Supervisor can provide some error management on resource usage violation***

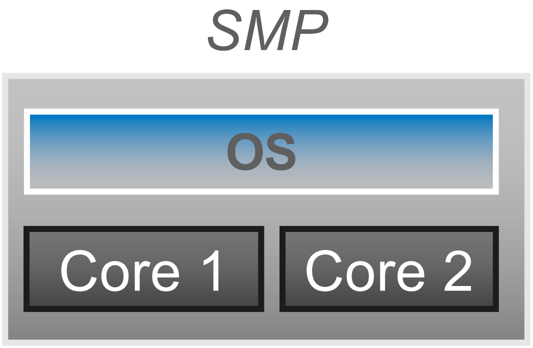
WNDRVR

Virtualization

Virtualization



- Use case support
  - All use cases can be supported (because it can include different OSes)
  - ***Hardware capabilities are key, but Software mitigations can also be applied***
- Interference Mitigation
  - Pushed on the hardware itself ***and on the Hypervisor***
  - ***Avoid resources sharing unless interferences impact can be assessed within WCET***
  - ***Easier with multiple memory controllers and/or "clusters of cores"***
  - ***Mitigation can be put in place by hardware configuration*** via the ***Hypervisor***
  - The ***Hypervisor*** can provide some error management on resource usage violation
  - ***The Hypervisor can perform some monitoring on resource usage***
  - ***The Hypervisor can put in place a time partitioning to control access to resources***



SMP OS

- Use case support
  - ***Limited to what the single SMP OS supports (because only one OS present)***
  - Hardware capabilities are key, but Software mitigations can also be applied

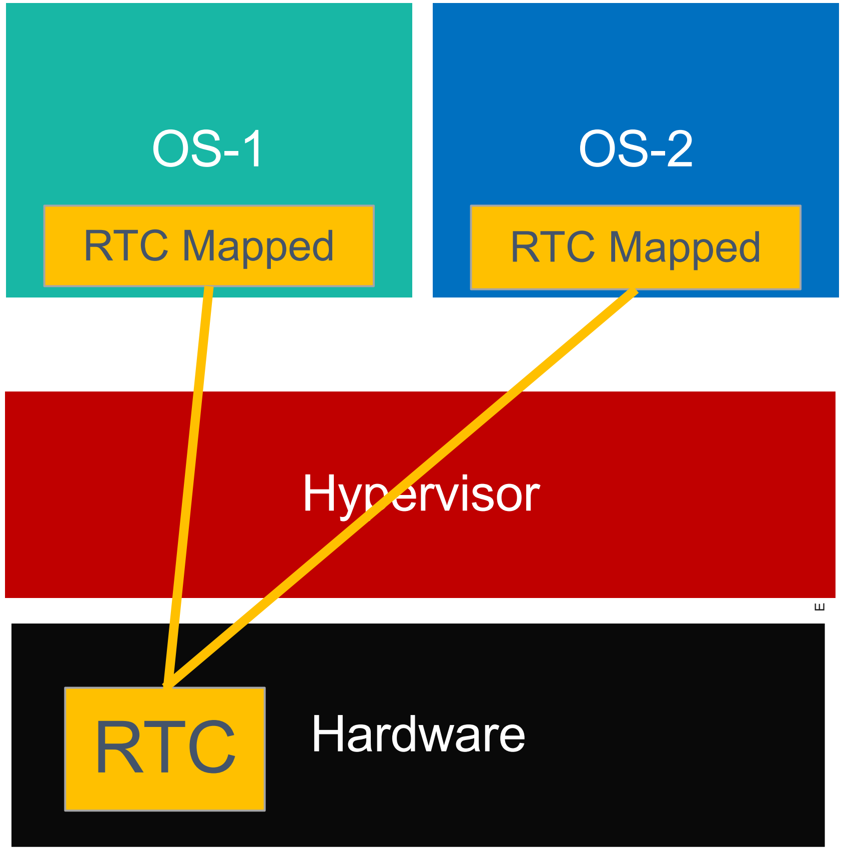
- Interference Mitigation
  - Pushed on the hardware itself ***and on the SMP OS***
  - ***Avoid resources sharing unless interferences impact can be assessed within WCET***
  - ***Easier with multiple memory controllers and/or "clusters of cores"***
  - ***Mitigation can be put in place by hardware configuration*** via the ***Supervisor***
  - The ***SMP OS*** can perform some monitoring on resource usage
  - The ***SMP OS*** can provide some error management on resource usage violation
  - The ***SMP OS*** can put in place a time partitioning to control access to resources

Resource Control – Simple Mapping

- Direct mapping to all Partitions
- Resource and interference management:
  - **Guest OS Level**
  - Write synchronized cooperatively among guests
  - Read-Only access

Caveat:

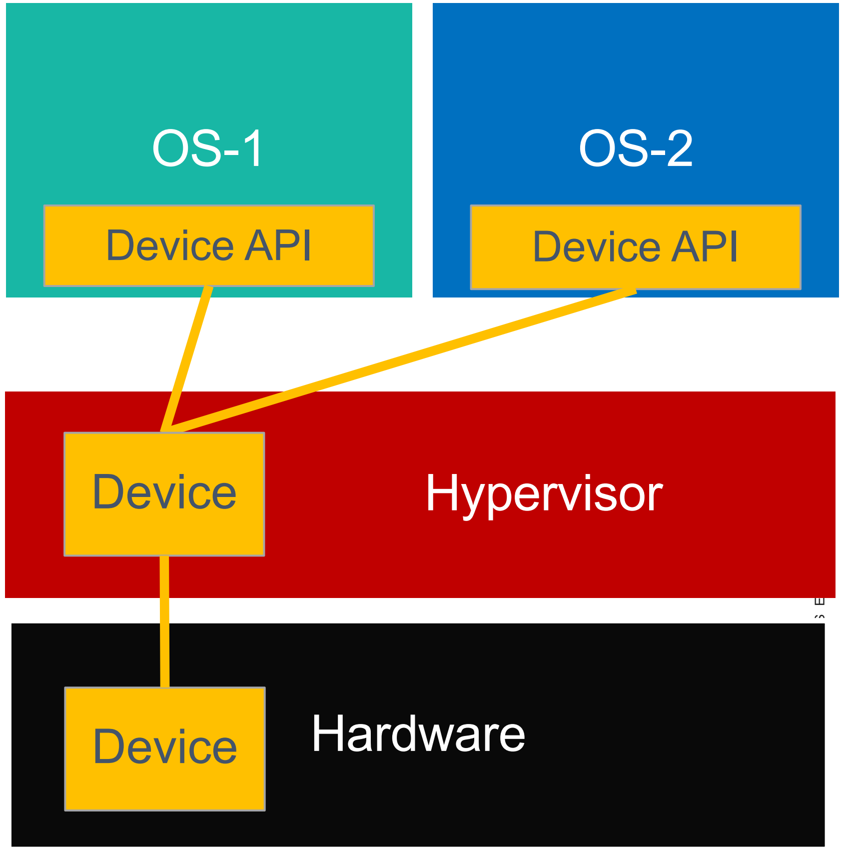
- Must have full hardware documentation to assess impact
- Not all resources can allow this



Resource Control – Split Model

- Virtualization is applied to each Guest OS
- Resource and interference management:
    - **Hypervisor**

Caveat: introduces overhead

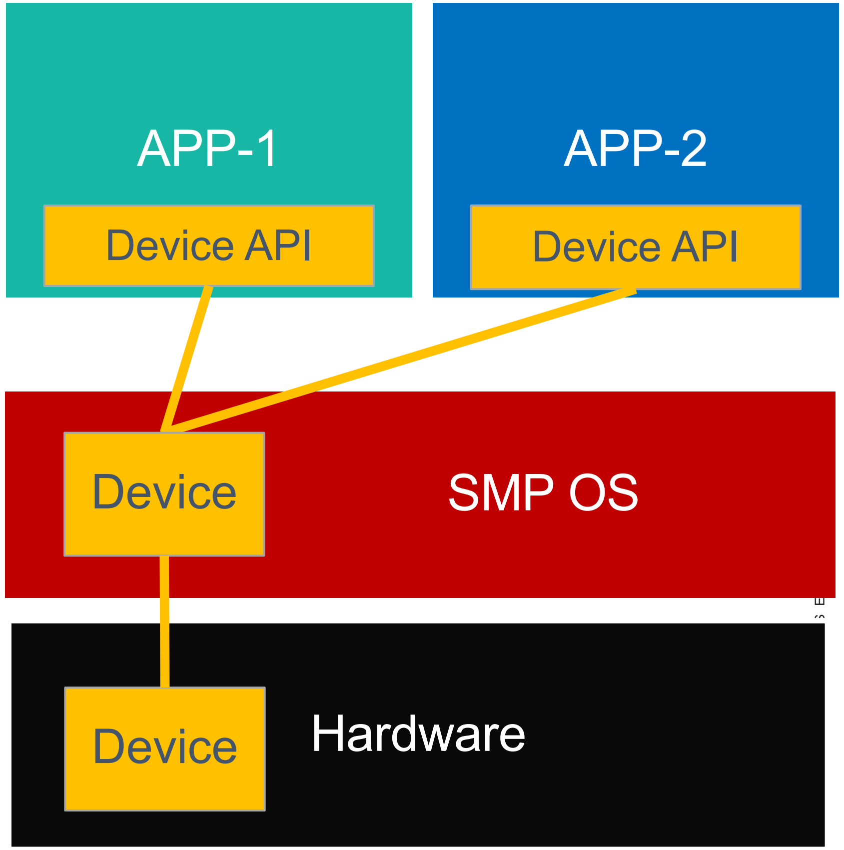


WNDRVR

Resource Control – Split Model

- Virtualization is applied to each Guest OS
- Resource and interference management:
  - **SMP OS Kernel Level**

Caveat: introduces overhead

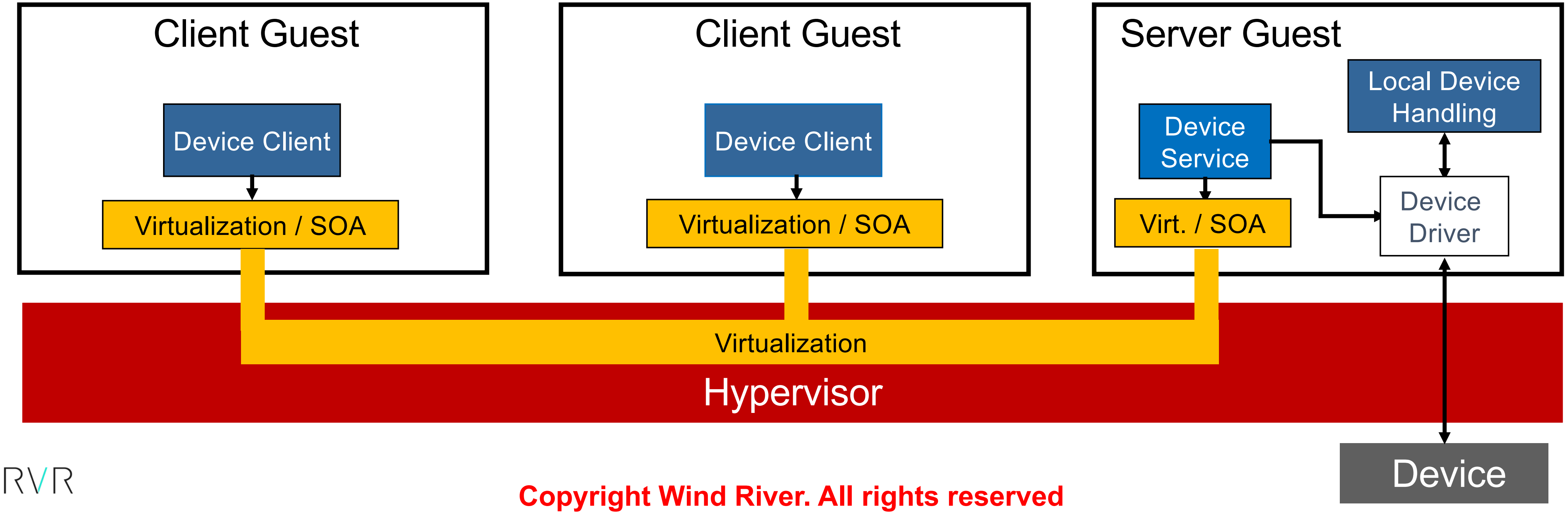


WNDRVR

Resource Control – Client / Server

- Direct mapping to server guest
- Sharing by client-server model using Virtualization

Caveat: Can typically introduce latencies



WNDRVR

Key Takes Away

- The choice of the OS has an impact on who will be in charge of the multi-core certification aspect
- Addressing AC/AMC 20-193 is typically an iterative process looping around:
  - The selection of the MCP configuration
  - The identification of the interferences induced by shared resources
  - The mitigation of such interferences
- In the course of such looping process, the software architecture selected can evolve to cope with/ease the multi-core certification
- Thinking in advance about the multi-core certification aspects can avoid making strong changes later in the project lifecycle

WNDRVR

Impact of the OS Type Selection

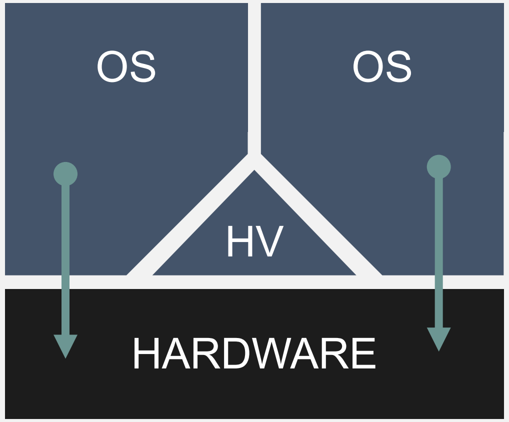
- Does it meet functional and specific project requirements?
- Who needs to take care of the addressing AC/AMC 20-193?
  - Bootloader supplier
  - BSP/Drive Supplier
  - OS Supplier
  - IMA Platform Supplier
  - System Integrator
  - Some of the above
  - All of the above
- Does Robust Partitioning Demonstration bring any value?
- Do you have enough information from the Hardware and the OS Supplier?

WNDRVR

Device Drivers Models

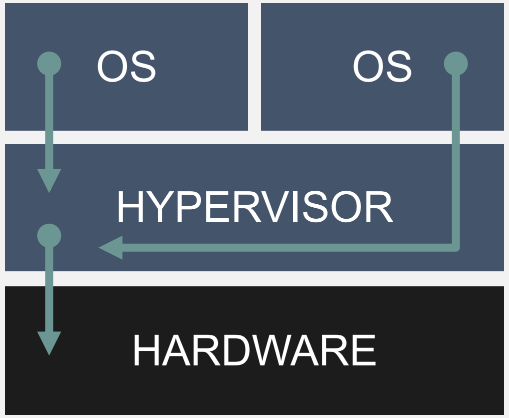
DIRECT ACCESS

- Hypervisor exposes hardware resources to Guest OS
- Native drivers in Guest OS
- Exclusive access
- Highest performance



EMULATED

- Hardware fully managed by Hypervisor, or virtual HW
- Native drivers in Guest OS
- Hypervisor intercepts HW access
- Sharing managed in Hypervisor



PROXY ACCESS

- Direct access to hardware for one Guest OS as I/O server
- SafeIPC used to access I/O server from other Partitions