

## **Hypervisor**

A Separation Concept to Achieve Adequate  
Independence between Safety Functions  
of Different SILs using the Virtualization Layer  
of Modern COTS CPUs

## ■ Agenda

- Definitions
- Motivation
- Use Cases
- Virtualization Challenges
- Virtualization Techniques
- Hypervisor Technology
- Available Technologies
- Outlook
- Summary

## ■ Definitions

### ■ **Virtualization**

Abstraction of computer resources, hiding the physical characteristics

### ■ **Hypervisor**

Virtualization platform that allows multiple operating systems to run on a host computer at the same time (Wikipedia)

### ■ **Virtual Board**

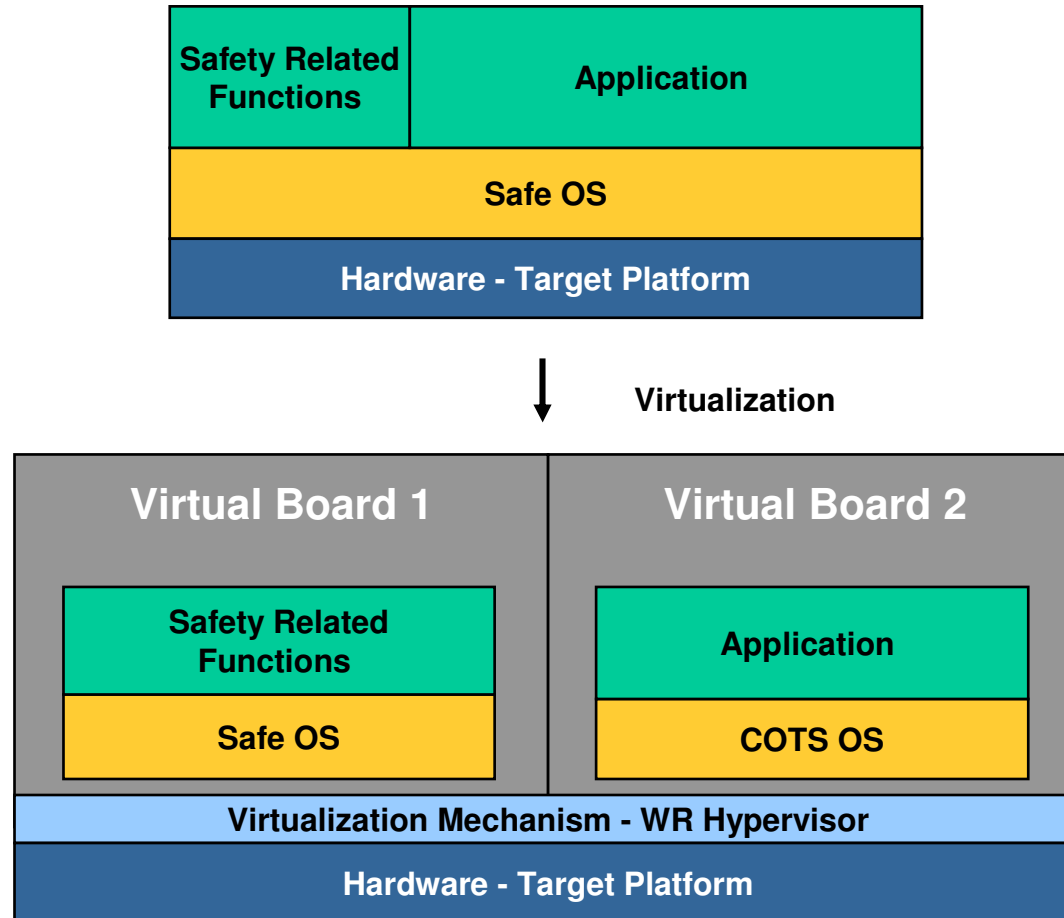
Environment for one operating system or bare application; has physical and/or virtual hardware controlled by the Hypervisor

## ■ Motivation for Virtualization

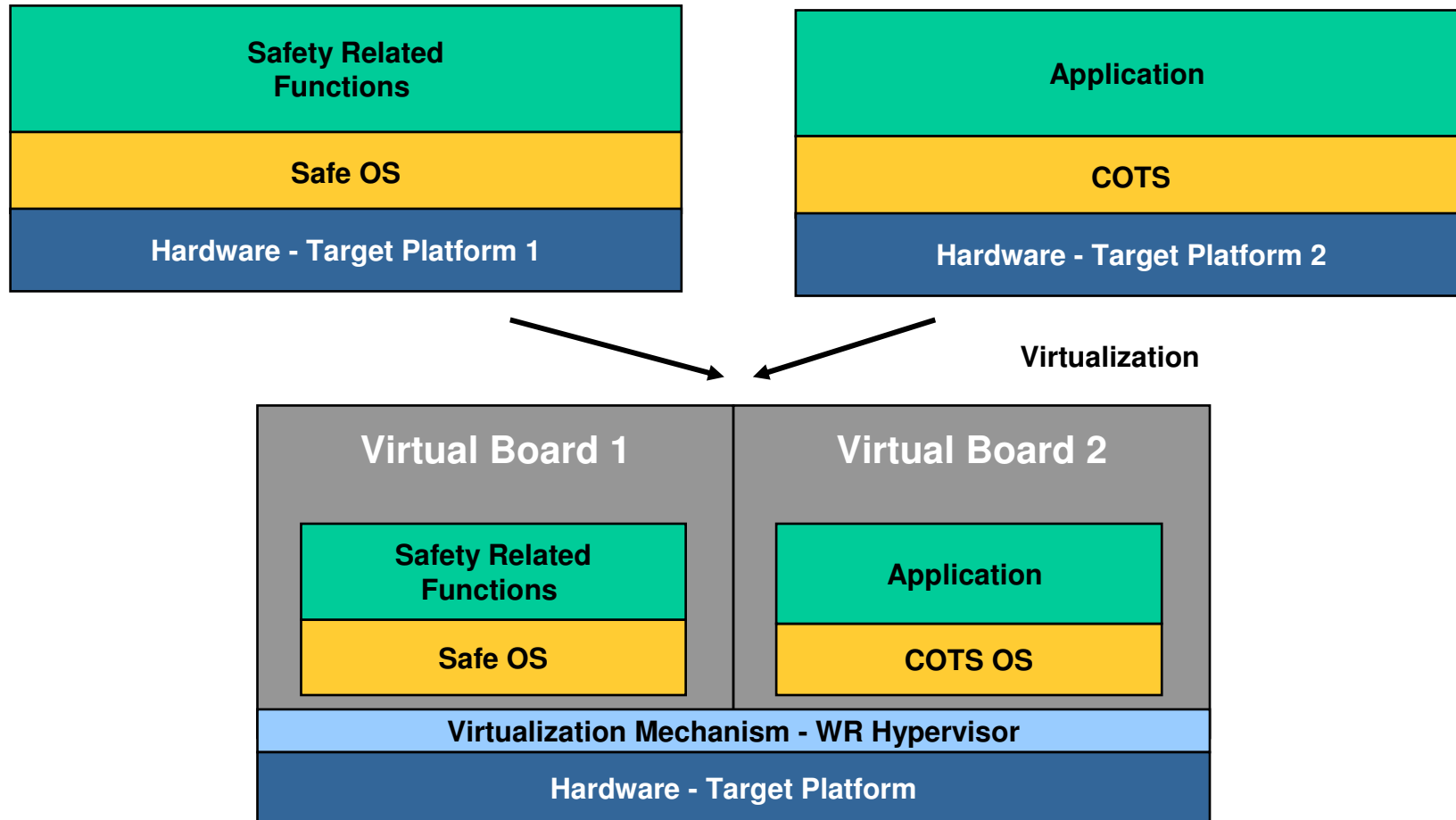
---

- **Limit Software Development Costs**  
Separate safety-related functions, Justify adequate independence.
- **Reliability**  
HW assisted separation, limit complexity, limit error propagation
- **Reduce Hardware**  
power consumption, thermal management, reduction of real estate.
- **Reusability**  
Legacy code runs with legacy OS, new applications on new OS
- **Hardware scalability and portability**  
SW systems unchanged between different HW, obsolescence

## ■ Virtualization – Use Case Separation



# ■ Virtualization – Use Case Integration



### ■ Adequate Independence

Adequate independence between the safety functions of the different safety integrity levels can be shown in the design. The justification for independence shall be documented.

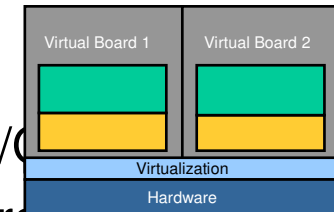
It shall be demonstrated either **(1)** that independence is achieved both in the **spatial** and **temporal** domain, or **(2)** that any violation of independence is controlled.

*[source: IEC 61508-3, First edition, Dated: 1998-12; IEC 61508-3, ED.2. Version 4:2007, Dated: 2007]*

## ■ Virtualization Techniques

### ■ Spatial separation

- MMU & I/O MMU to separate memory domains and I/O
- VMMU & VI/O MMU to set up a system of virtual boards
- Save Inter Process Communication (SIPC)



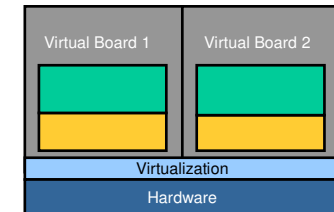
### ■ Temporal separation

- Deterministic scheduling
  - Scheduling policy (time slice, priority)
- Exception Handling
- Cache and DMA Management

## ■ Virtualization Techniques

### ■ Health Monitoring Framework

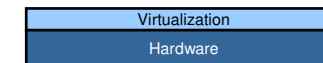
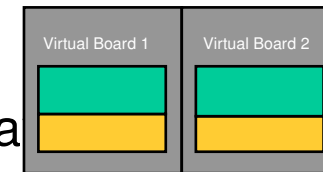
- Fault detection (BSP)
- Fault Injection
- Fault logging
- Fault routing (configuration)
  - Operating System Level
  - Virtual Board Level
  - System Level
- Fault handling (BSP)



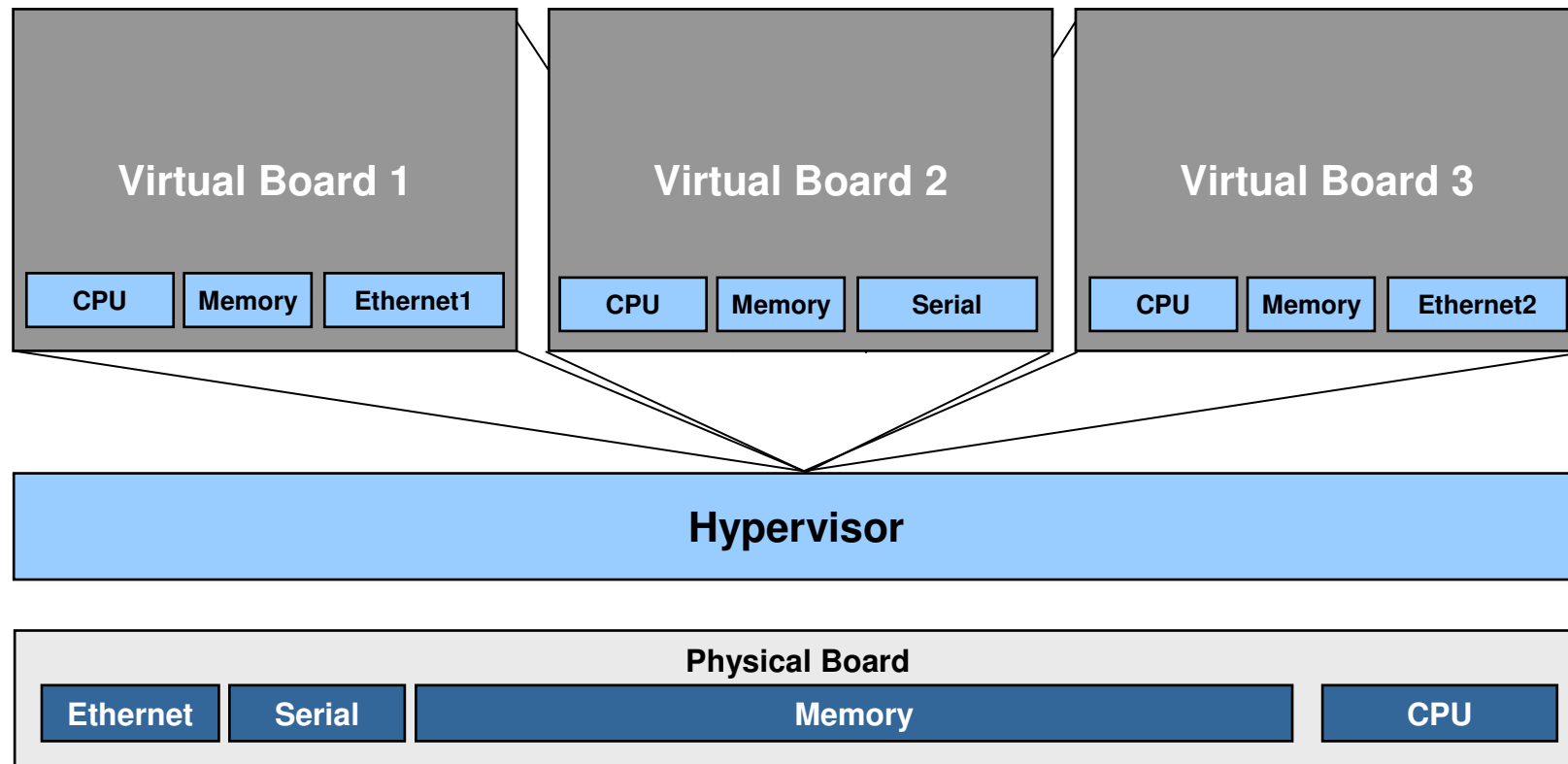
## ■ Virtualization Techniques

### ■ Development

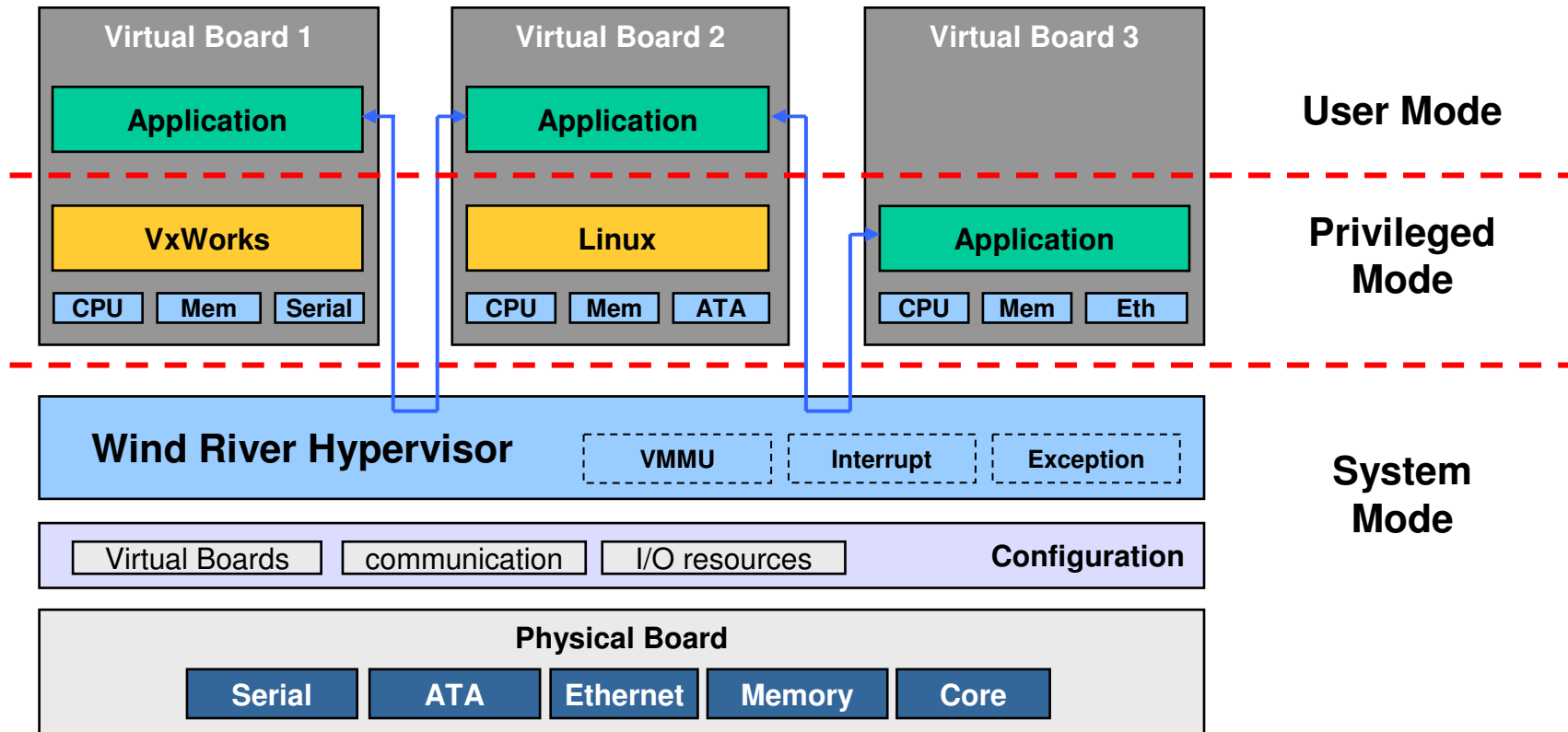
- Development organisation
  - Application Supplier
  - Safety Platform Supplier (virtualisation la
  - System Integrator



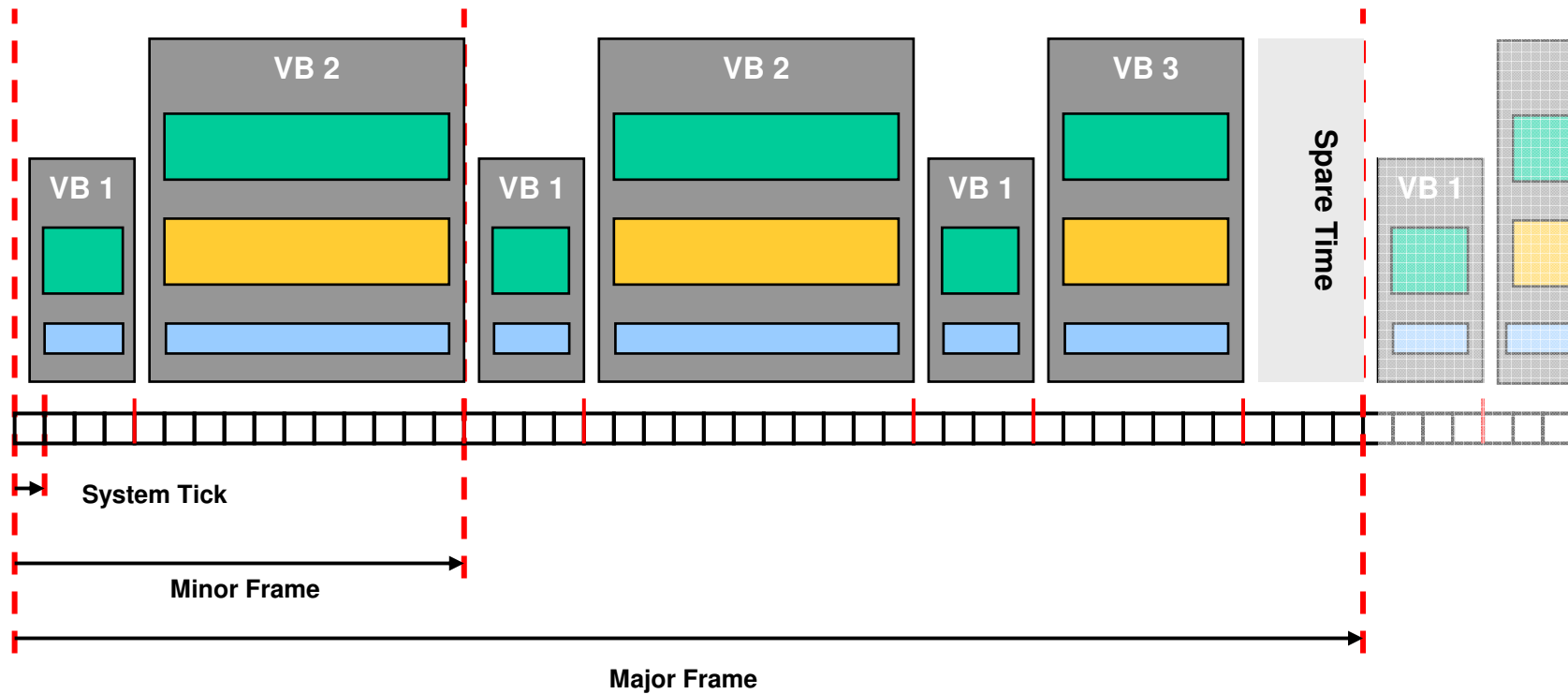
- Resource configuration (CPU time, Memory, I/O Devices)
- Proof and document time and space independence provided by Hypervisor (Robust Partitioning)



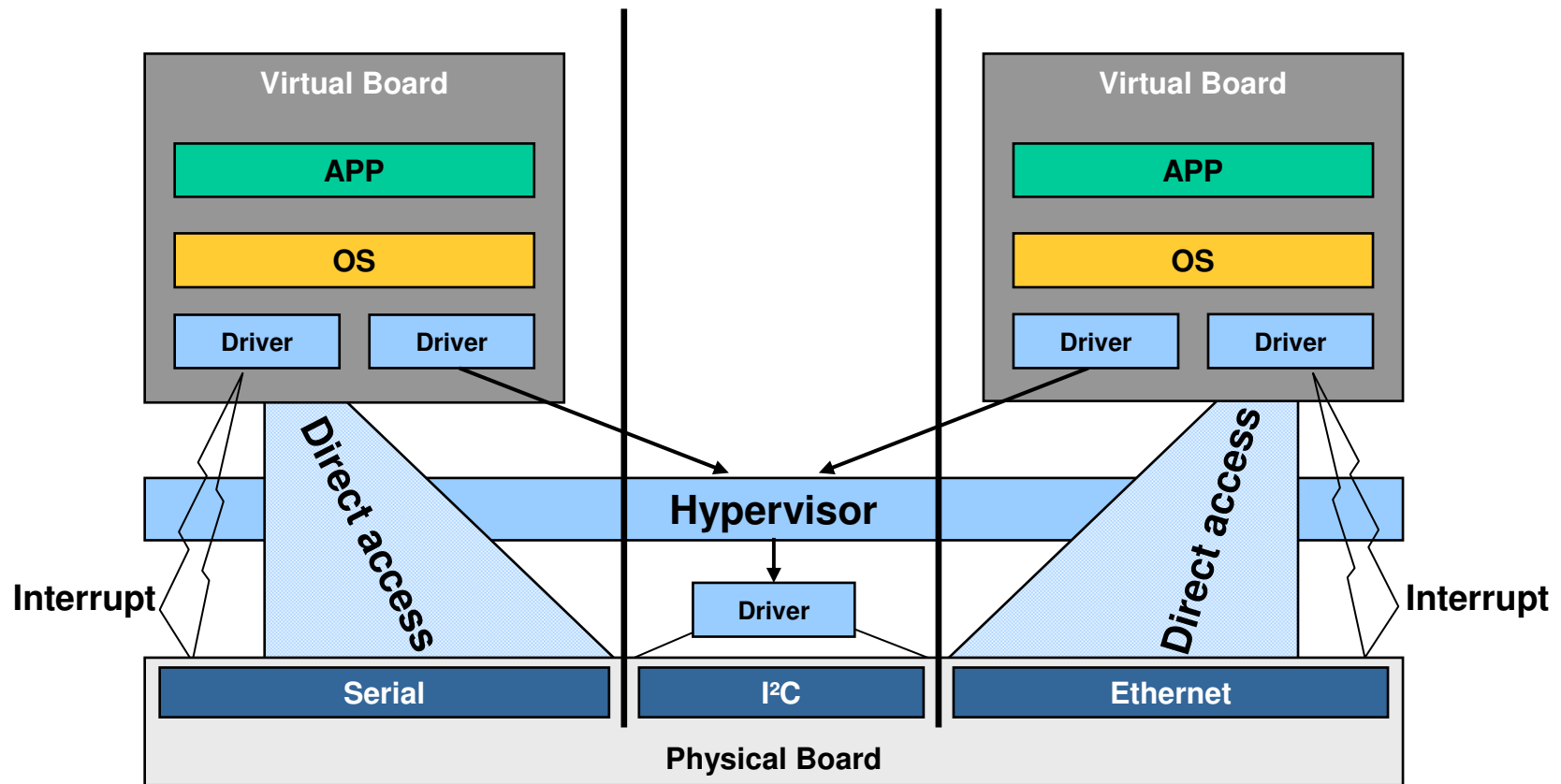
# Spatial Separation



# Temporal Separation



## Resource configuration



## ■ Available Technologies - Examples

---

- AMD
  - AMD-V
  - CPUs supporting AMD-V (Athlon 64, Athlon 64 X2, Turion 64 X2, Opteron 2nd and 3rd generation, Phenom, and newer CPUs)
- Freescale
  - e500mc
- Intel
  - Launched 2005 with Pentium 4 ( 672 and 662 )
  - Intel® Virtualization Technology (Intel VT-x)
  - Intel® Virtualization Technology for Directed I/O (VT-d)

## ■ Outlook

- Next Version of IEC 61508, Part3 specifies technics for separation (Annex G)
- Virtualisation technics are deployed in Aerospace (e.g 787, A380, A400, C130-AMP...) (ARINC653, DO127B, DO297 / ED124)
- Multi Core CPUs
  - Shared Resources (Cache, Bus, RAM, I/O devices)
- Parallel Computing
  - SMP, AMP



## ■ Summary

- Virtualization
  - Use Cases in Safety – Separation, Integration
  - Future to harness Multi Cores
- Hypervisor as small separation kernel
- System Design
  - Spatial separation
  - Temporal separation
  - I/O Driver Design
  - Monitoring
  - Development (Virtual Board, Safe Platform Provider, System Integrator)

