ARINC 653

Introduction

- Credits:
  - An Avionics Standard for Safe, Partitioned Systems – Wind River 2008 – IEEE CS Seminar
  - Masmano et al. - ARINC-653 APEX based on XtratuM
  - Ananda et al. - ARINC 653 API and its application – An insight into Avionics System Case Study
  - Samolej - ARINC Specification 653 Based Real-Time Software Engineering

Introduction

- More functionalities, more connectivity,…, in less space, weight, and power (SWaP)

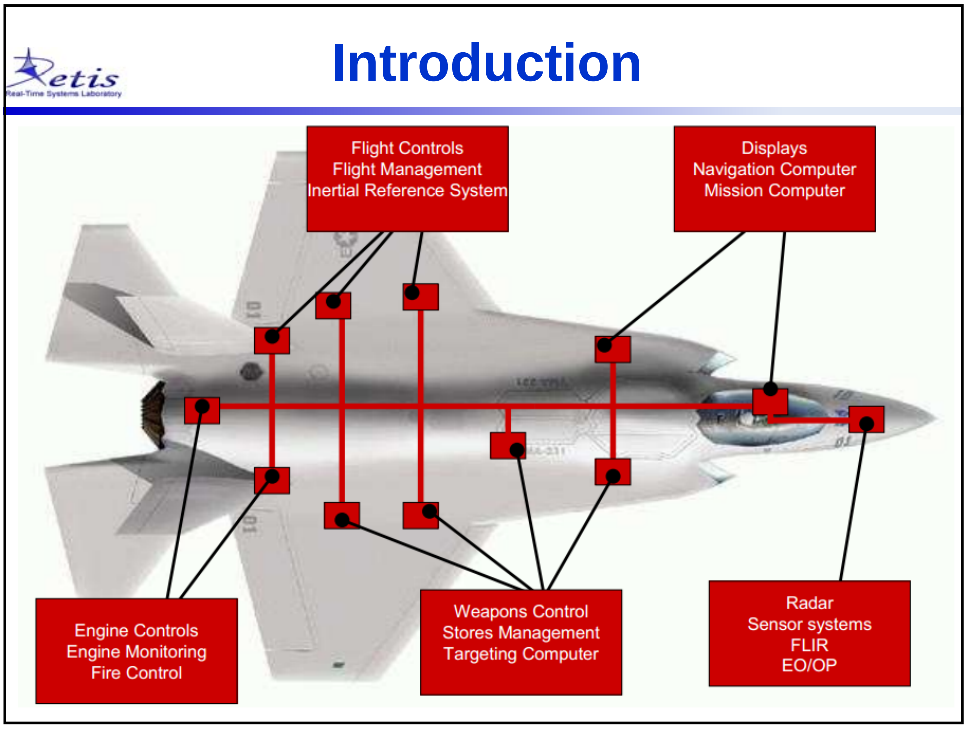
Hardware consolidation (multiple applications on fewer processors)

Software "pressure": larger volume of software comingled on fewer processors

New challenges to safe and secure

Introduction

- Federated vs IMA (Integrated Modular Avionics)
- Similarly to the automotive industry, the avionic industry is moving from a federated approach to an integration of multiple software systems on the same processing unit.



Retis Real-Time Systems Laboratory

Introduction

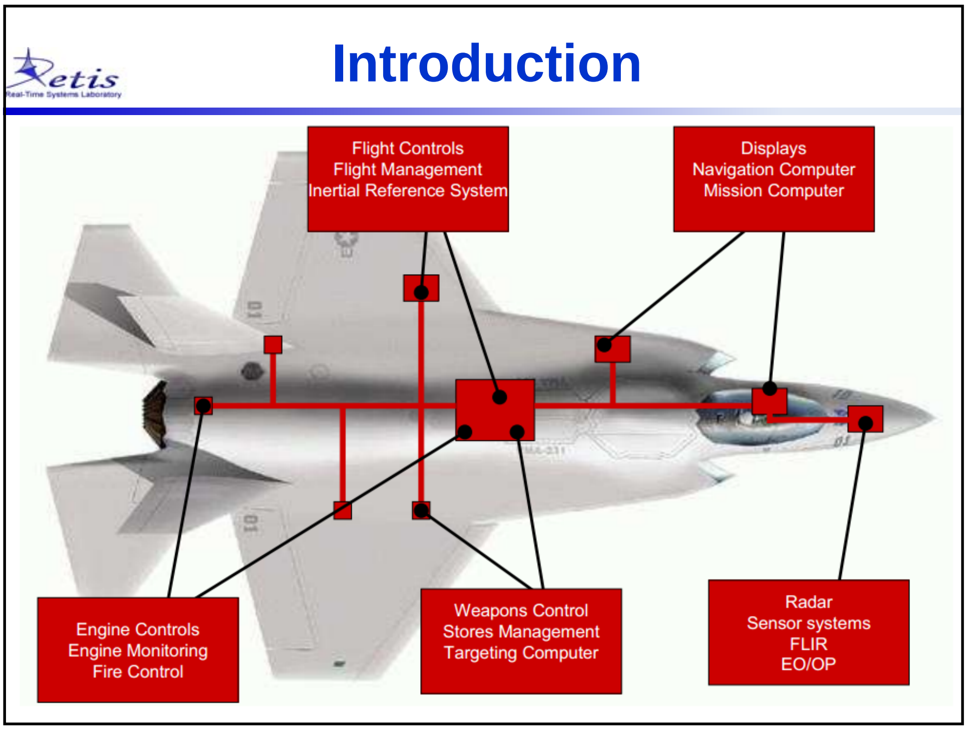
Flight Controls
Flight Management
Inertial Reference System

Displays
Navigation Computer
Mission Computer

Engine Controls
Engine Monitoring
Fire Control

Weapons Control
Stores Management
Targeting Computer

Radar
Sensor systems
FLIR
EO/OP



Retis Real-Time Systems Laboratory

Introduction

Flight Controls
Flight Management
Inertial Reference System

Displays
Navigation Computer
Mission Computer

Engine Controls
Engine Monitoring
Fire Control

Weapons Control
Stores Management
Targeting Computer

Radar
Sensor systems
FLIR
EO/OP

Introduction

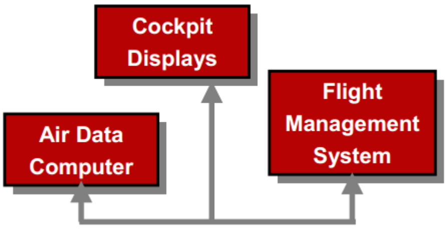
Federated

PROs

- Traditional methodology
- Relative “easy” design and certification
- Existing supply chain

CONs

- SWaP
- Poor SW reuse
- Poor portability
- Poor modularity



Introduction

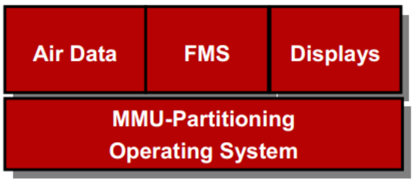
IMA

PROs

- SWap
- Excellent SW reuse
- Excellent portability
- Excellent modularity

CONs

- Modern methodology
- Complexity of design and certification
- Supply chain not setup for IMA projects



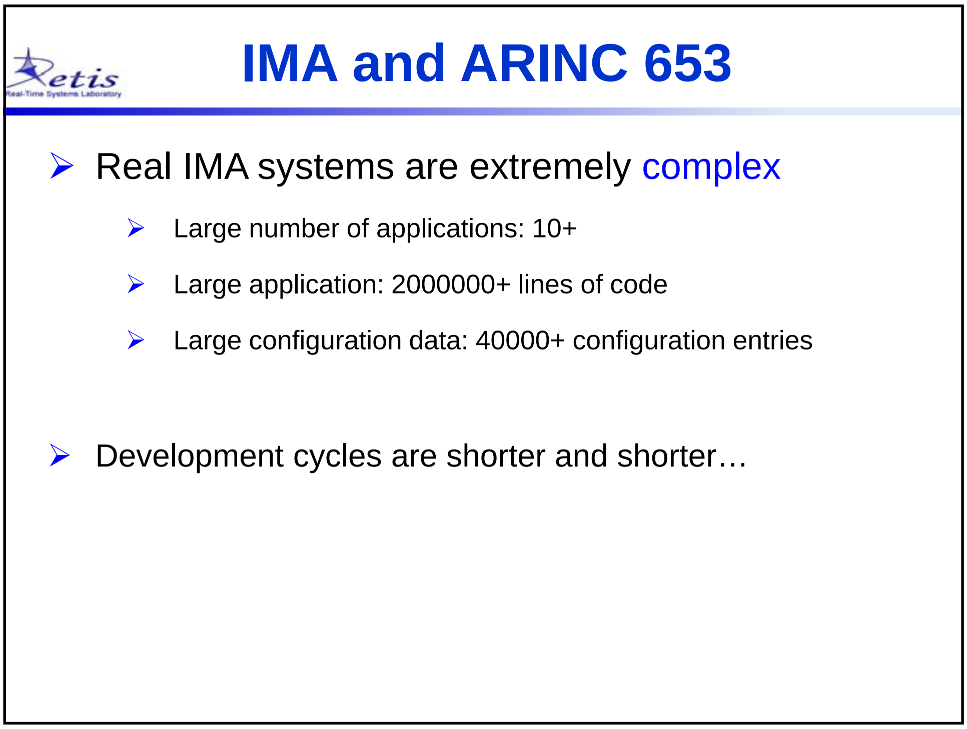
Introduction

Federated vs IMA – The reality of today

- They will co-exist for some time;
- E.g., flight controls (highly critical) are still preferred to be served by a dedicated execution unit

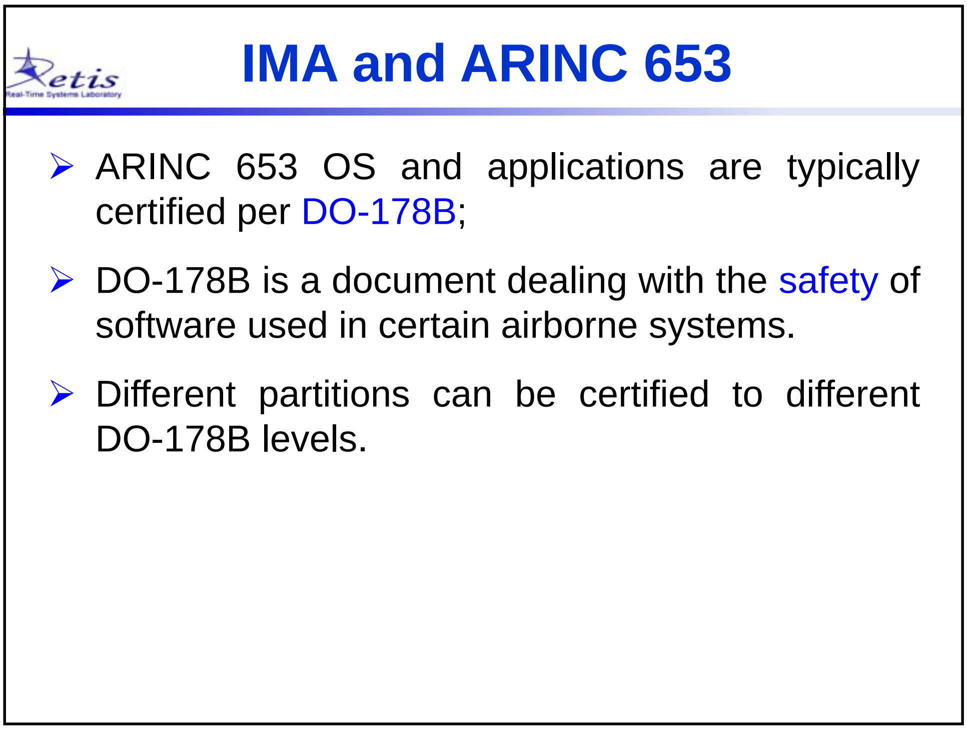
IMA and ARINC 653

- Integrating different systems into one CPU environment
  - Multiple vendors using the same processor;
  - Safety-critical control systems (potentially with different criticality levels);
  - Integrated platform with multiple OSes



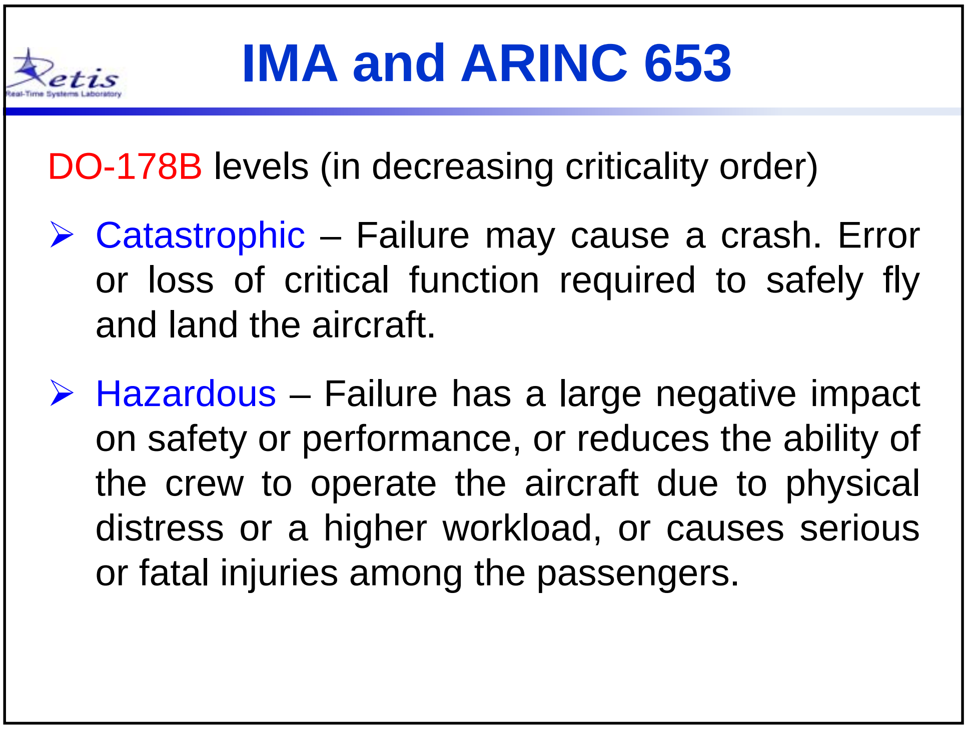
IMA and ARINC 653

- Real IMA systems are extremely complex
  - Large number of applications: 10+
  - Large application: 2000000+ lines of code
  - Large configuration data: 40000+ configuration entries
- Development cycles are shorter and shorter…



IMA and ARINC 653

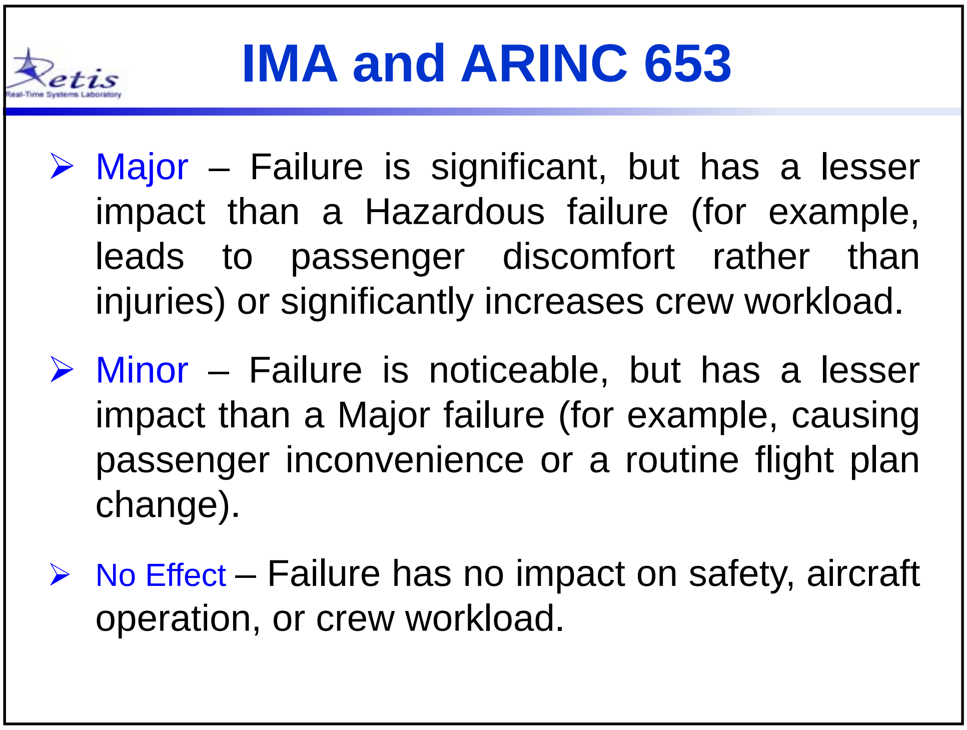
- ARINC 653 OS and applications are typically certified per DO-178B;
- DO-178B is a document dealing with the safety of software used in certain airborne systems.
- Different partitions can be certified to different DO-178B levels.



IMA and ARINC 653

DO-178B levels (in decreasing criticality order)

- Catastrophic – Failure may cause a crash. Error or loss of critical function required to safely fly and land the aircraft.
- Hazardous – Failure has a large negative impact on safety or performance, or reduces the ability of the crew to operate the aircraft due to physical distress or a higher workload, or causes serious or fatal injuries among the passengers.



IMA and ARINC 653

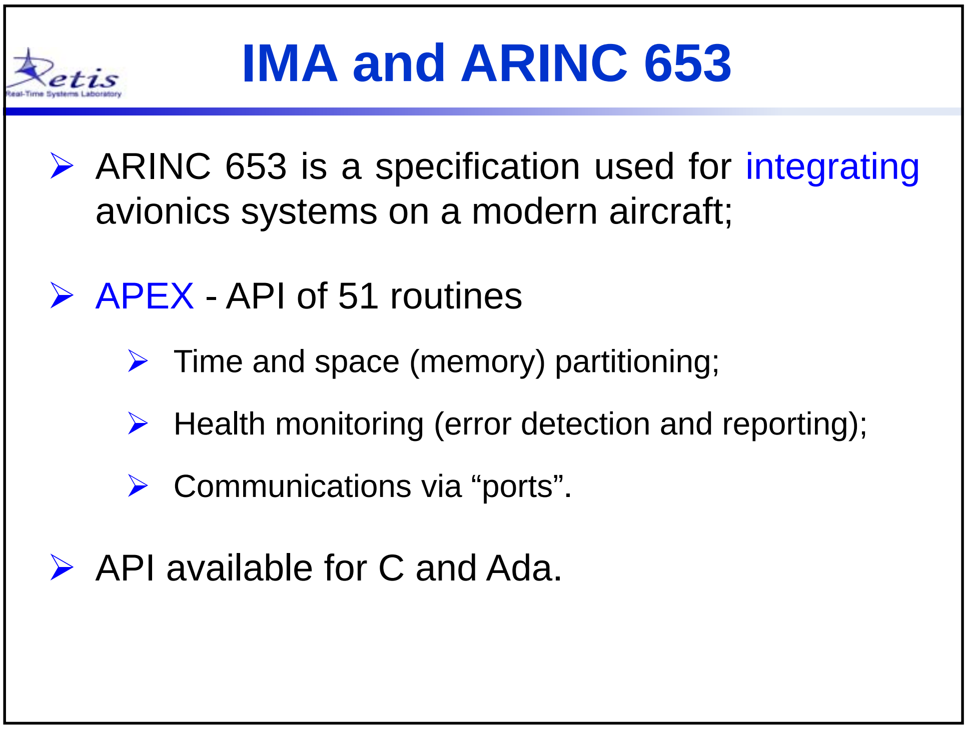
- Major – Failure is significant, but has a lesser impact than a Hazardous failure (for example, leads to passenger discomfort rather than injuries) or significantly increases crew workload.
- Minor – Failure is noticeable, but has a lesser impact than a Major failure (for example, causing passenger inconvenience or a routine flight plan change).
- No Effect – Failure has no impact on safety, aircraft operation, or crew workload.

IMA and ARINC 653

- The aviation industry developed ARINC 653 as a standardized RTOS interface definition between the RTOS of an avionics computer resource and the application software.
- This benefits both the software developers as well as the hardware platform suppliers.

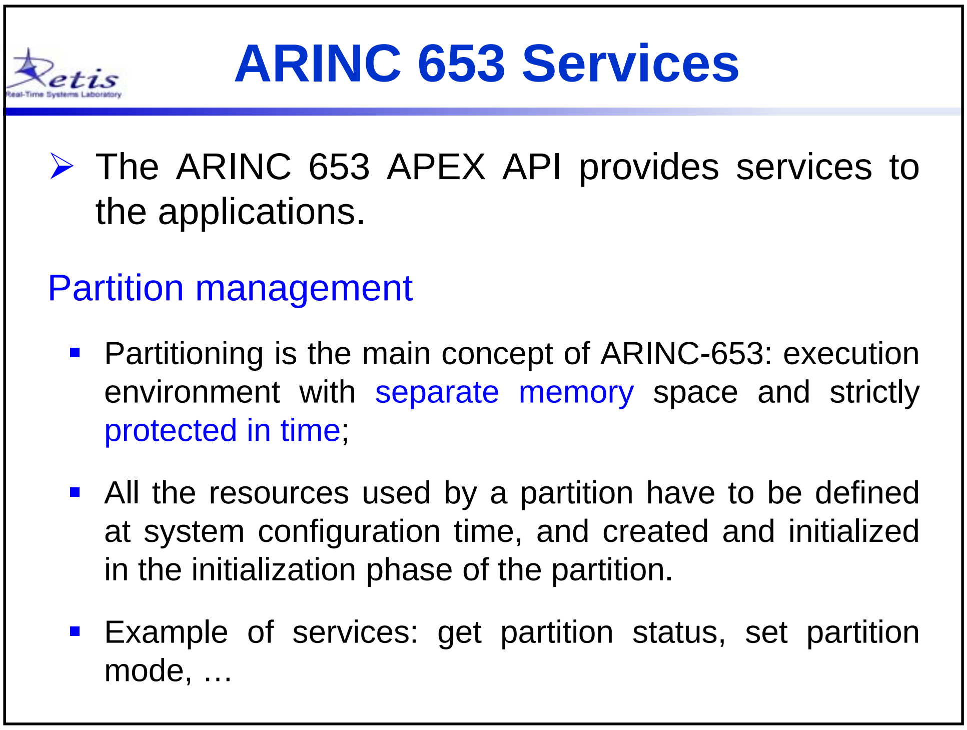
IMA and ARINC 653

- To meet software certification requirement of DO-178B, 3 main needs have been identified
  - Safety-critical – according to a law
  - Real-Time – response times must be within a predetermined time period
  - Deterministic – results of the execution must be predictable and repeatable
- ARINC 653's RTOS guarantee an interface boundary for avionics software development, thus allowing independence of the avionics software applications.



IMA and ARINC 653

- ARINC 653 is a specification used for integrating avionics systems on a modern aircraft;
- APEX - API of 51 routines
  - Time and space (memory) partitioning;
  - Health monitoring (error detection and reporting);
  - Communications via “ports”.
- API available for C and Ada.



ARINC 653 Services

- The ARINC 653 APEX API provides services to the applications.

Partition management

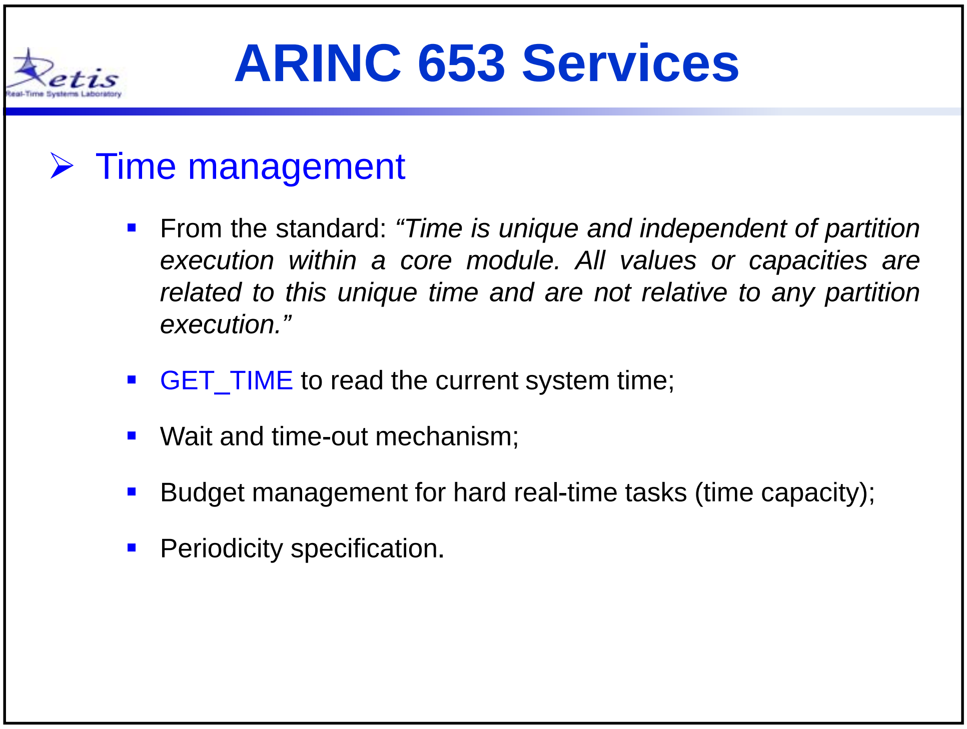
- Partitioning is the main concept of ARINC-653: execution environment with separate memory space and strictly protected in time;
- All the resources used by a partition have to be defined at system configuration time, and created and initialized in the initialization phase of the partition.
- Example of services: get partition status, set partition mode, …

ARINC 653 Services

- Process management
  - A partition comprises one or more processes;
  - Tipically the processes are scheduled according to Fixed-Priority preemptive (or limited preemptive) policy;
  - An ARINC 653 process can be in one of 4 available states
    - Dormant – ineligible for scheduling;
    - Waiting – not able to execute;
    - Ready – able to be executed;
    - Running – currently executing.

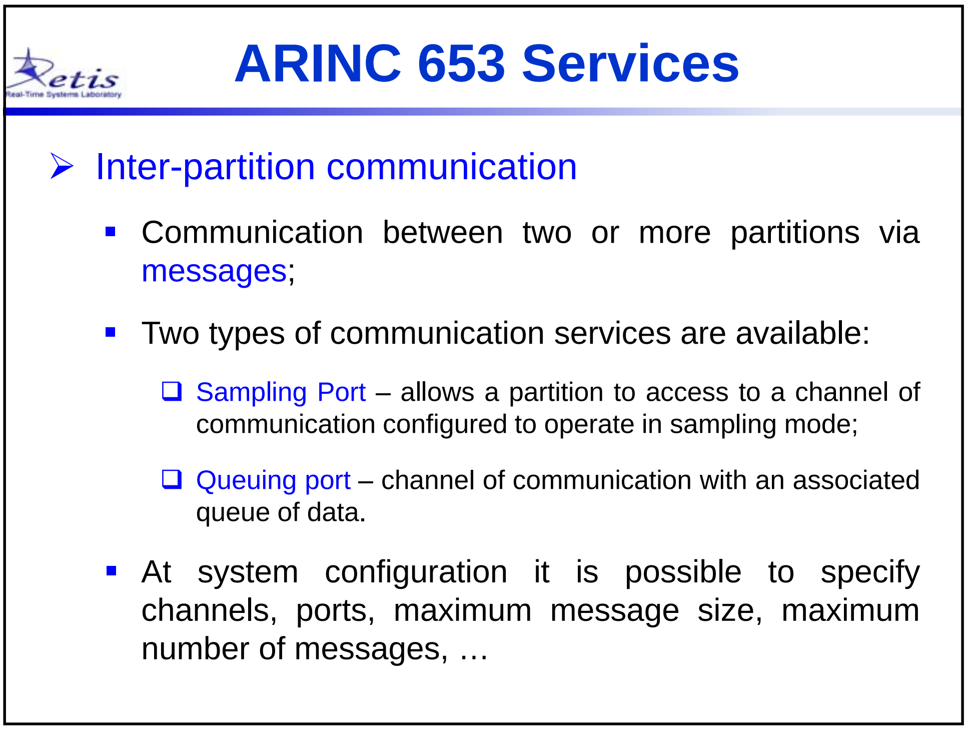
ARINC 653 Services

- Process management – typical operations
  - create process and collect process status or ID;
  - start, stop, suspend or resume the process;
  - prevent process pre-emption;
  - change the process priority.



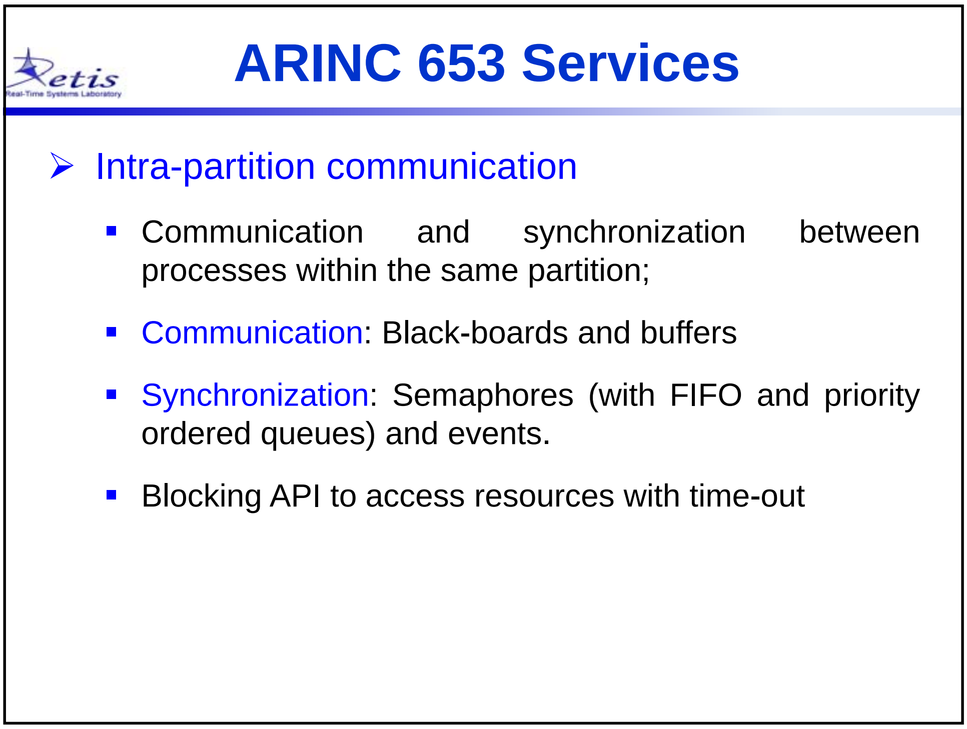
ARINC 653 Services

- Time management
  - From the standard: *"Time is unique and independent of partition execution within a core module. All values or capacities are related to this unique time and are not relative to any partition execution."*
  - GET_TIME to read the current system time;
  - Wait and time-out mechanism;
  - Budget management for hard real-time tasks (time capacity);
  - Periodicity specification.



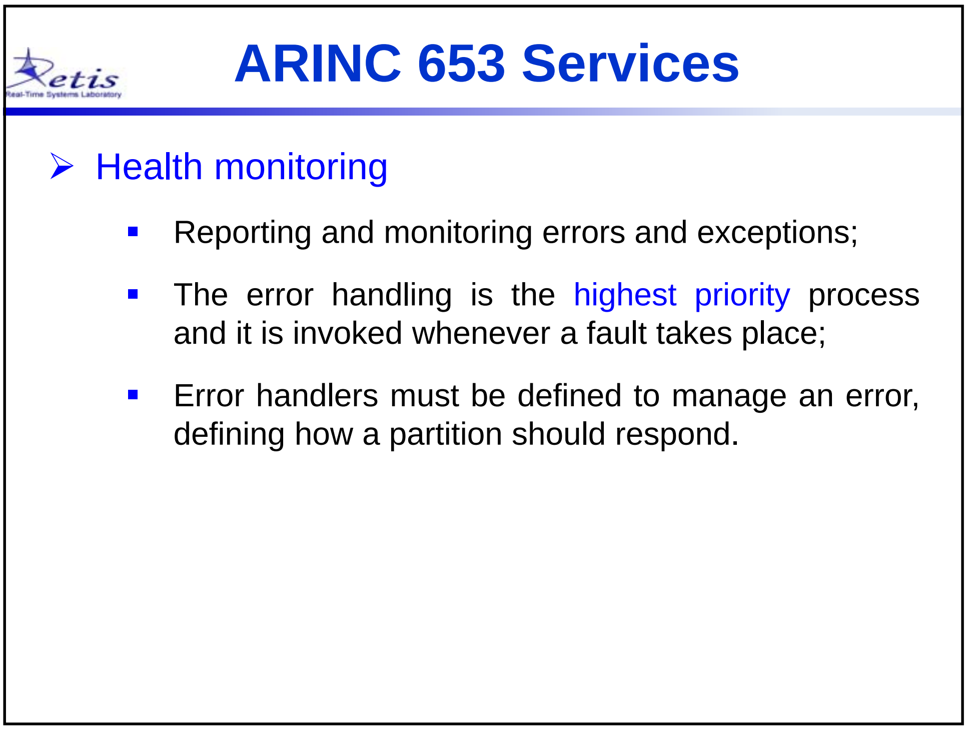
ARINC 653 Services

- Inter-partition communication
  - Communication between two or more partitions via messages;
  - Two types of communication services are available:
    - ❑ Sampling Port – allows a partition to access to a channel of communication configured to operate in sampling mode;
    - ❑ Queuing port – channel of communication with an associated queue of data.
  - At system configuration it is possible to specify channels, ports, maximum message size, maximum number of messages, …



ARINC 653 Services

- Intra-partition communication
  - Communication and synchronization between processes within the same partition;
  - Communication: Black-boards and buffers
  - Synchronization: Semaphores (with FIFO and priority ordered queues) and events.
  - Blocking API to access resources with time-out



ARINC 653 Services

- Health monitoring
  - Reporting and monitoring errors and exceptions;
  - The error handling is the highest priority process and it is invoked whenever a fault takes place;
  - Error handlers must be defined to manage an error, defining how a partition should respond.

ARINC 653 Services

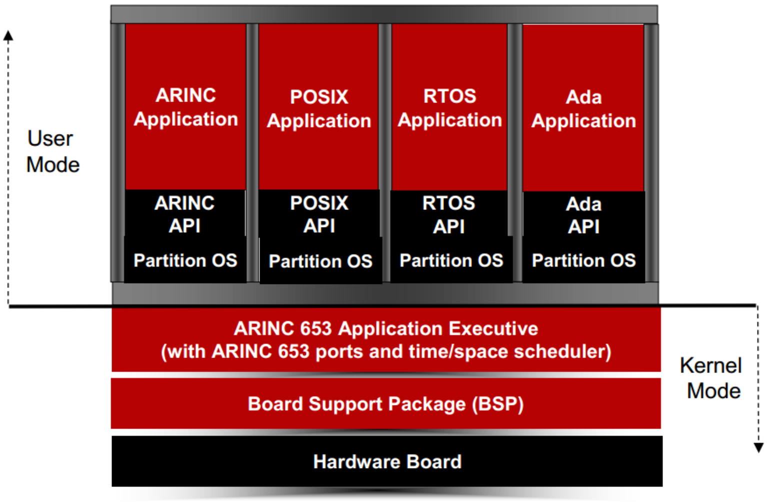
- Health monitoring – Example of error handling
  - I. Log the error;
  - II. Stop or restart the failed process;
  - III. Eventually stop or restart the entire partition;
  - IV. Invoke the registered handler for the specific error code

ARINC 653 Services

- All the OS configurations are specified through XML;
- XML specifications are also used for testing, verification and certification of the system;
- There exist tools (from Wind River) to keep track of software requirements in the system configuration.

ARINC 653 OS

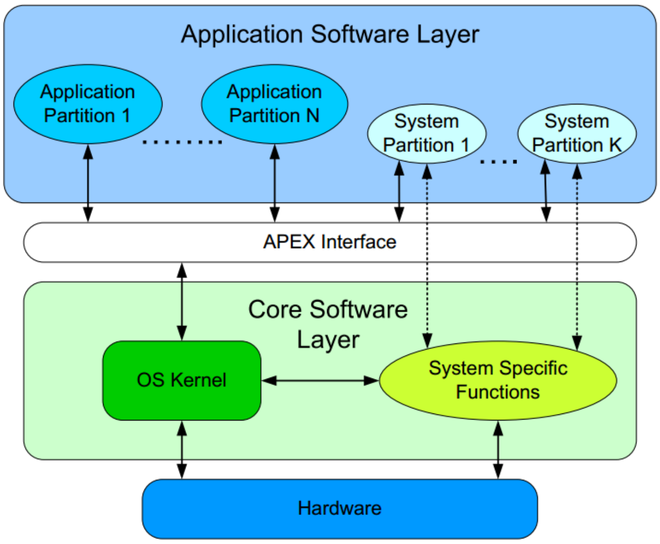
ARINC 653 OS



ARINC 653 OS

- Spatial partitioning – must ensure that software in one partition cannot change the software or private data of another partition, nor command the private devices or actuators of other partitions.
- Temporal partitioning – must ensure that the service received from shared resources by the software in one partition cannot be affected by the software in another partition in terms of rate, latency, jitter, and duration of scheduled access to it.

ARINC 653 OS



Application Software Layer

Application Partition 1

Application Partition N

System Partition 1

System Partition K

APEX Interface

Core Software Layer

OS Kernel

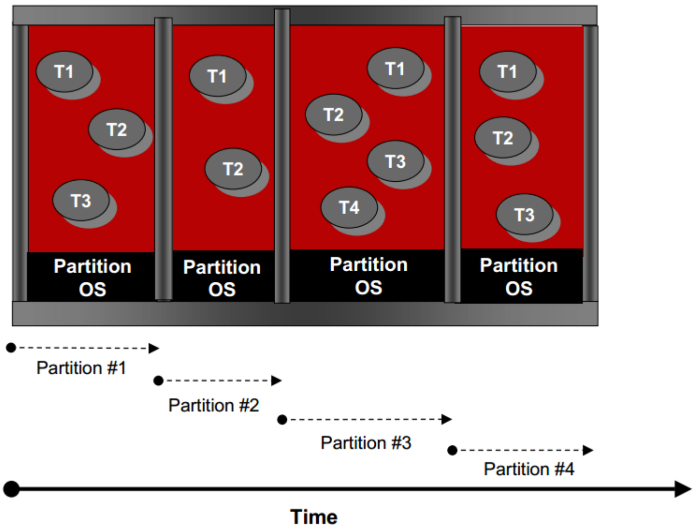
System Specific Functions

Hardware

ARINC 653 OS

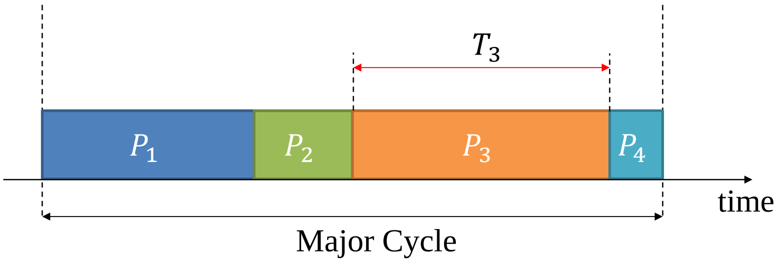
- The partitions are divided into two categories: application partition and system partition.
  - Application partitions execute avionic applications and interact with the environment by means of the APEX interface.
  - The system partitions are optional and their main role is to provide services not available in APEX, such as device drivers or fault management, actually bypassing the APEX interface.

ARINC 653 OS



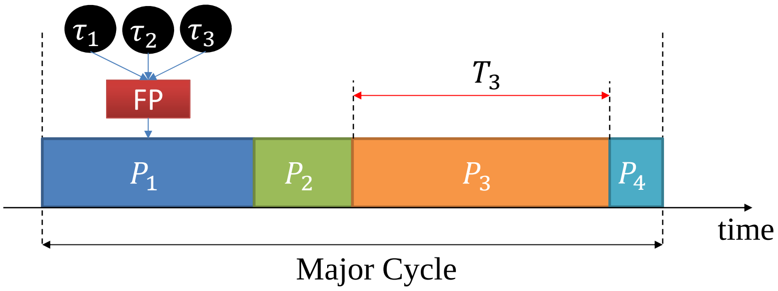
ARINC 653 OS

- Time-Division based scheduling of predetermined set of partitions $P_1, P_2, \ldots, P_n$

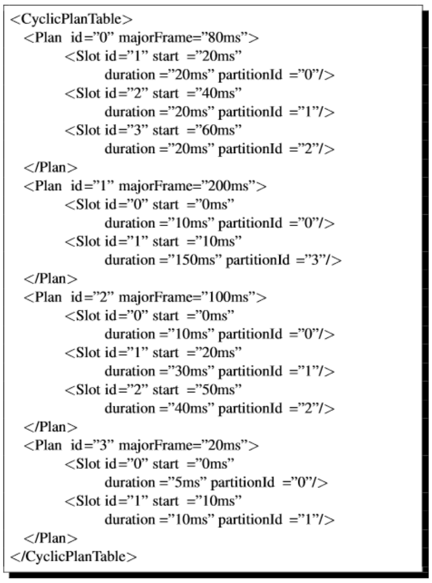


ARINC 653 OS

- Hierarchical scheduling – each partition hosts a specific OS scheduler (typically based on fixed priorities)

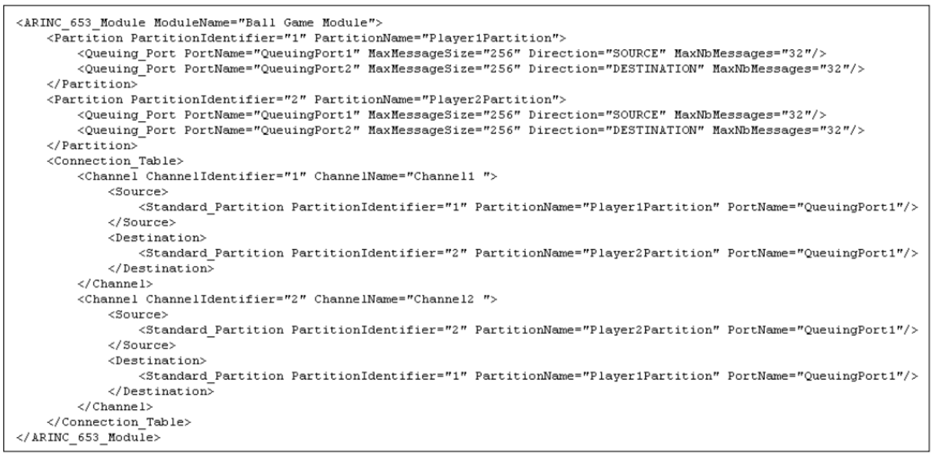


ARINC 653 OS

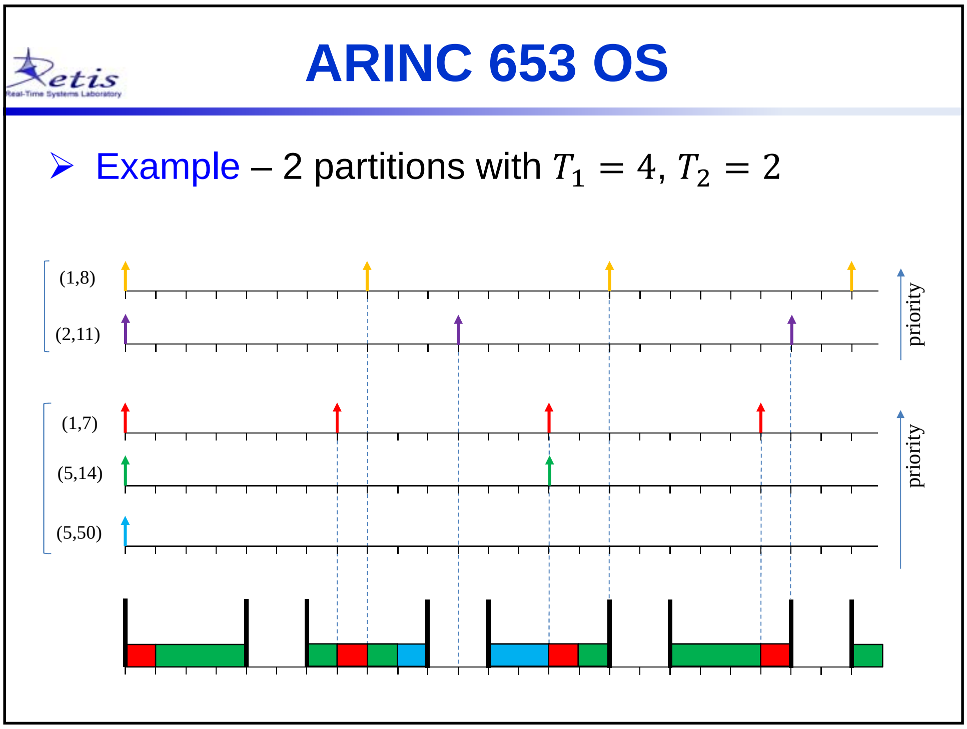


```
<CyclicPlanTable>
  <Plan  id=”0” majorFrame=”80ms”>
        <Slot id=”1” start  =”20ms”
              duration =”20ms” partitionId  =”0”/>
        <Slot id=”2” start  =”40ms”
              duration =”20ms” partitionId  =”1”/>
        <Slot id=”3” start  =”60ms”
              duration =”20ms” partitionId  =”2”/>
  </Plan>
  <Plan  id=”1” majorFrame=”200ms”>
        <Slot id=”0” start  =”0ms”
              duration =”10ms” partitionId  =”0”/>
        <Slot id=”1” start  =”10ms”
              duration =”150ms” partitionId  =”3”/>
  </Plan>
  <Plan  id=”2” majorFrame=”100ms”>
        <Slot id=”0” start  =”0ms”
              duration =”10ms” partitionId  =”0”/>
        <Slot id=”1” start  =”20ms”
              duration =”30ms” partitionId  =”1”/>
        <Slot id=”2” start  =”50ms”
              duration =”40ms” partitionId  =”2”/>
  </Plan>
  <Plan  id=”3” majorFrame=”20ms”>
        <Slot id=”0” start  =”0ms”
              duration =”5ms” partitionId  =”0”/>
        <Slot id=”1” start  =”10ms”
              duration =”10ms” partitionId  =”1”/>
  </Plan>
</CyclicPlanTable>
```

ARINC 653 OS

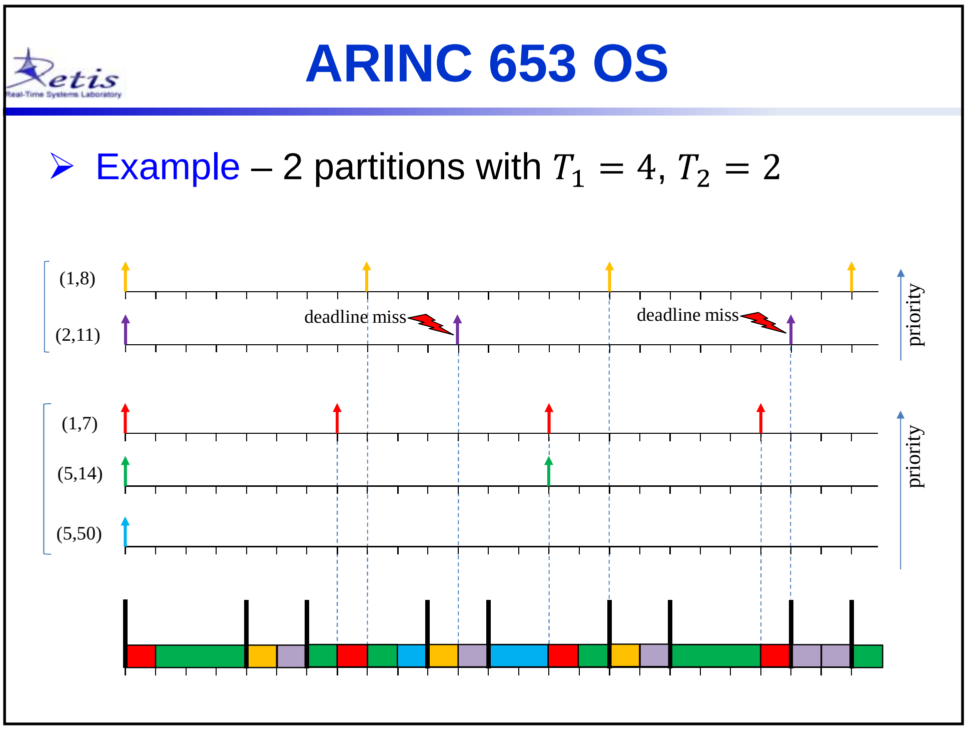


```
<ARINC_653_Module ModuleName="Ball Game Module">
    <Partition PartitionIdentifier="1" PartitionName="Player1Partition">
        <Queuing_Port PortName="QueuingPort1" MaxMessageSize="256" Direction="SOURCE" MaxNbMessages="32"/>
        <Queuing_Port PortName="QueuingPort2" MaxMessageSize="256" Direction="DESTINATION" MaxNbMessages="32"/>
    </Partition>
    <Partition PartitionIdentifier="2" PartitionName="Player2Partition">
        <Queuing_Port PortName="QueuingPort1" MaxMessageSize="256" Direction="SOURCE" MaxNbMessages="32"/>
        <Queuing_Port PortName="QueuingPort2" MaxMessageSize="256" Direction="DESTINATION" MaxNbMessages="32"/>
    </Partition>
    <Connection_Table>
        <Channel ChannelIdentifier="1" ChannelName="Channel1 ">
            <Source>
                <Standard_Partition PartitionIdentifier="1" PartitionName="Player1Partition" PortName="QueuingPort1"/>
            </Source>
            <Destination>
                <Standard_Partition PartitionIdentifier="2" PartitionName="Player2Partition" PortName="QueuingPort1"/>
            </Destination>
        </Channel>
        <Channel ChannelIdentifier="2" ChannelName="Channel2 ">
            <Source>
                <Standard_Partition PartitionIdentifier="2" PartitionName="Player2Partition" PortName="QueuingPort1"/>
            </Source>
            <Destination>
                <Standard_Partition PartitionIdentifier="1" PartitionName="Player1Partition" PortName="QueuingPort1"/>
            </Destination>
        </Channel>
    </Connection_Table>
</ARINC_653_Module>
```



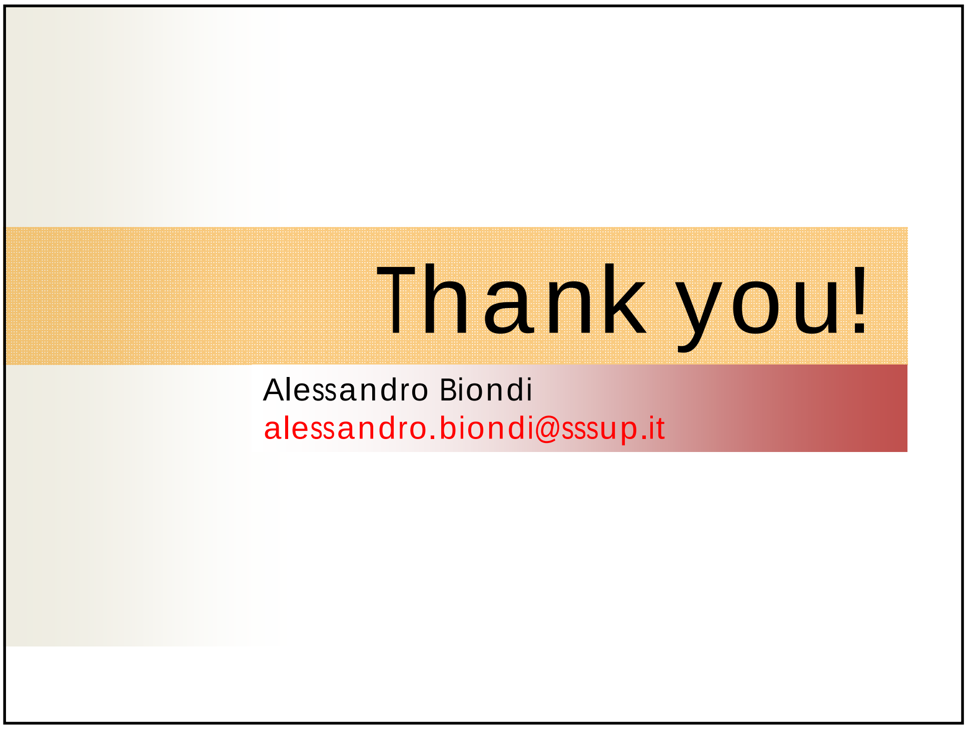
ARINC 653 OS

- Example – 2 partitions with $T_1 = 4$, $T_2 = 2$



ARINC 653 OS

- Example – 2 partitions with $T_1 = 4$, $T_2 = 2$



Thank you!

Alessandro Biondi
alessandro.biondi@sssup.it