**OSPERT – Operating Systems Platforms for Embedded Real-Time Applications**
**July 5th - 2016, Toulouse, France**

LaSIGE
Ciências ULisboa

# Timeliness Runtime Verification and Adaptation in Avionic Systems

**José Rufino and Inês Gouveia**

**LaSIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal**

Navigators

# Earlier Projects

## The roots of this work

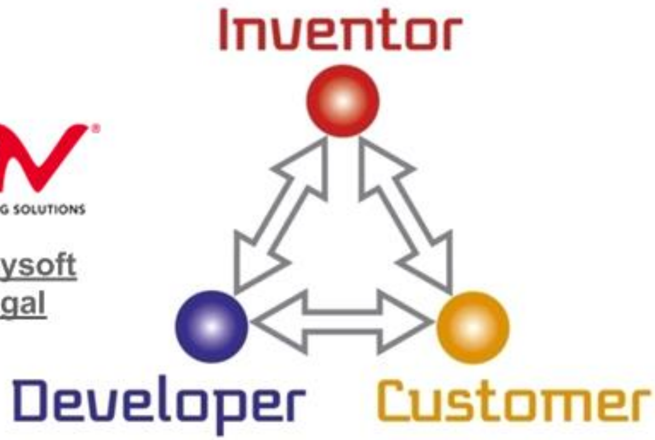
esa Innovation Triangle Initiative Program

Faculdade de Ciências UNIVERSIDADE DE LISBOA

LaSIGE - Navigators Group
Portugal

gmv INNOVATING SOLUTIONS

GMV-Skysoft
Portugal



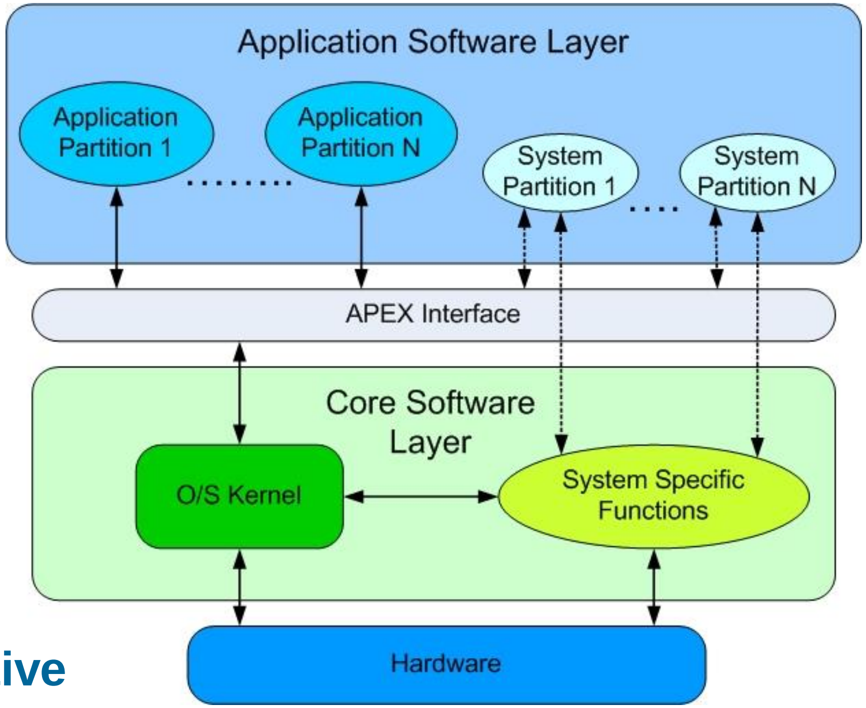

ThalesAlenia Space

Thales Alenia Space, France

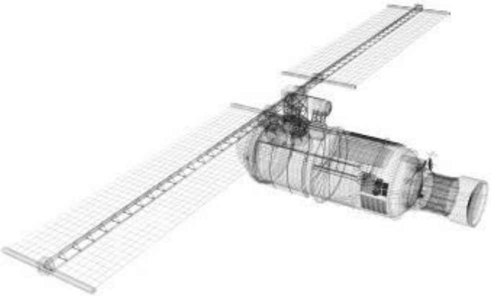
- Project AIR (Type A - Proof of Concept)
- Project AIR-II (Type B - Demonstration of Feasibility and Use)

# AIR – ARINC 653 Interface in Real-Time Operating Systems

- Civil aviation industry
  - A380 and B787 avionics
- Native ARINC 653 OS
- **Challenge**:
  - Use of COTS RTOS
    - **RTEMS - Real-Time Executive for Multiprocessor Systems**



# AIR – ARINC 653 Interface in Real-Time Operating Systems



- AIR Proof of Concept Main Objectives:
  - Study the adoption of ARINC 653 to space.
  - Evaluate how RTEMS can be adapted to fulfill the requirements of the ARINC 653 specification.

- Proposed Methodology to AIR – specification of:
  - Modules to be included in RTEMS;
  - Modules needing to be modified/extended;
  - Modules needing to be removed.

  - <u>not strictly followed when project started.</u>

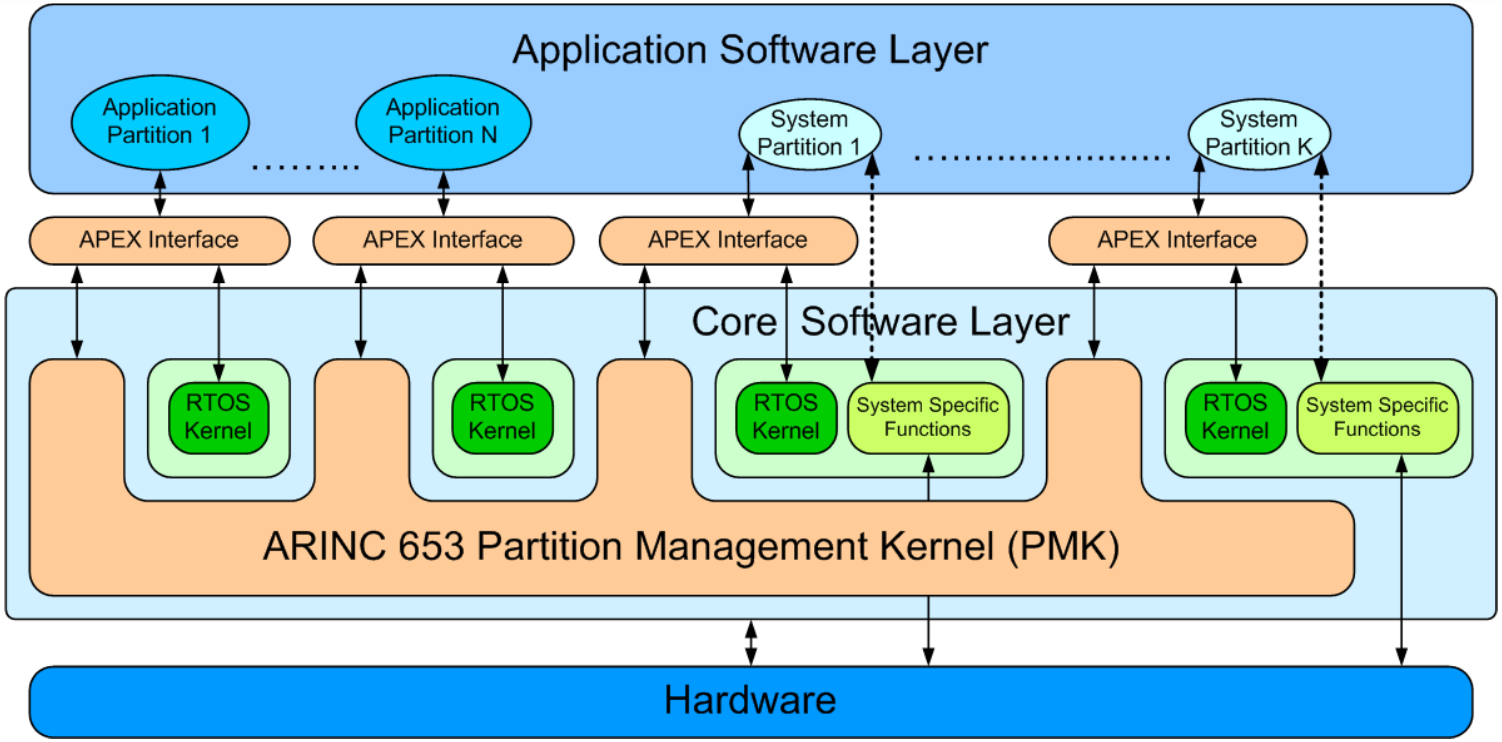
José Rufino, LaSIGE/FCUL, Portugal <jmrufino@ciencias.ulisboa.pt>

# The motivation: nowadays…

- Application to autonomous unnamed vehicles
  - next generation spacecrafts
  - aerial drones
  - aquatic drones
  - terrestrial vehicles

- Some require low-end cost solutions
- Safety is sometimes a disregarded requirement

# AIR Architecture




- Combination of time and event-triggered approaches
- Multi-executive core layer with two-level hierarchical scheduling
- Portable APEX design concept

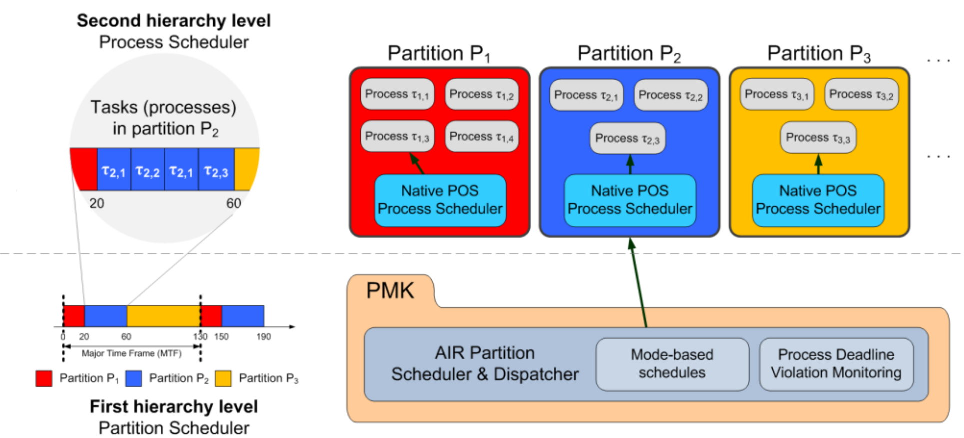
# Time and Space Partitioning

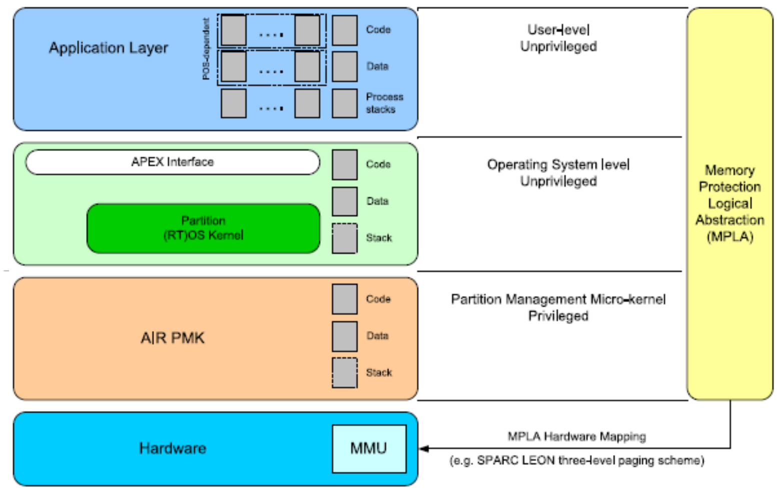
## Time partitioning

- Two-level hierarchical scheduling
- Fixed cyclic partition scheduling, RTOS process scheduling

## Space partitioning

- High-level processor-independent abstraction
- Mapping of high-level partition description to low-level mechanisms



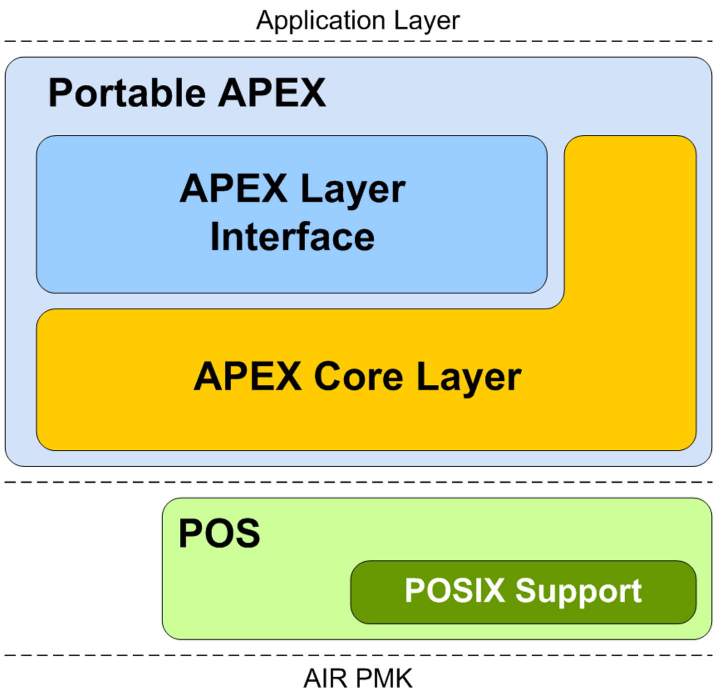




José Rufino, LaSIGE/FCUL, Portugal <jmrufino@ciencias.ulisboa.pt>

# AIR System Architecture

## APEX – Application Executive Interface

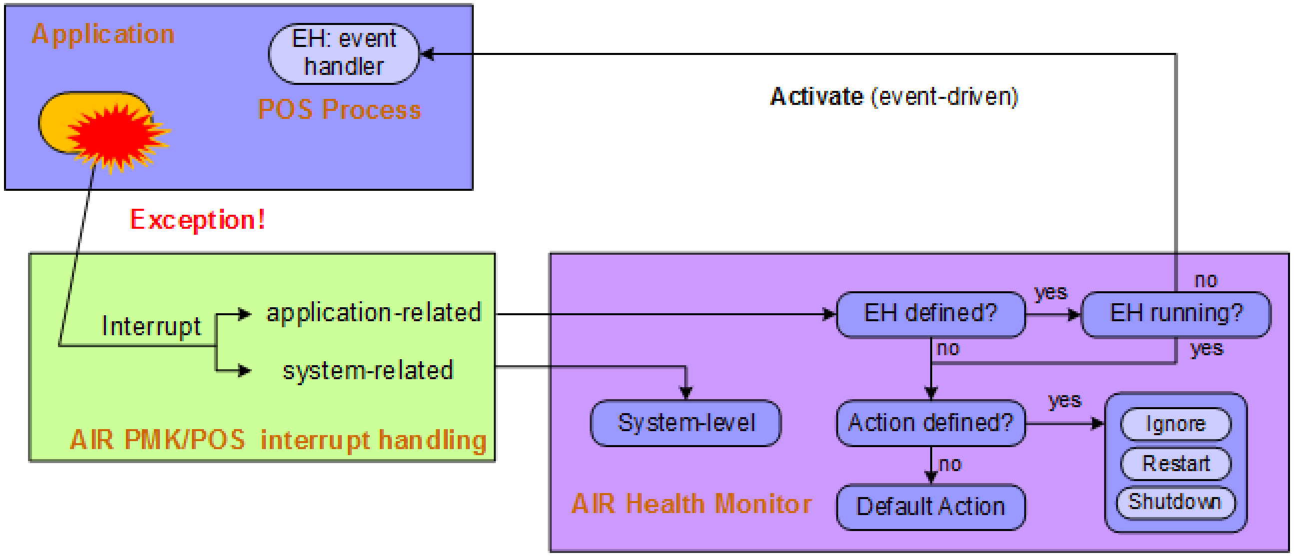
- Flexible Portable APEX
  - Services defined in the ARINC 653 specification
  - Generic OSs: only subset of the APEX services
    - Management/monitoring, interpartition communication



# AIR System Architecture

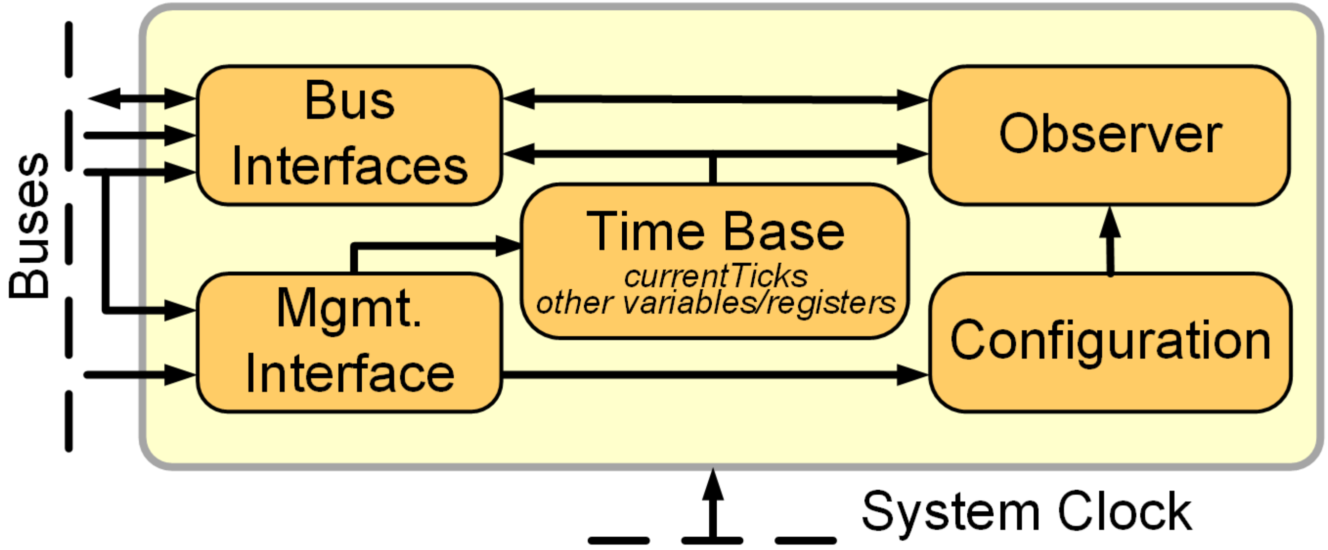
## AIR Health Monitoring (HM)

- Responsible for handling HW/SW errors
- Deviation from a system specification
- Isolate errors within domain of occurrence
  - Process
  - Partition
  - System



# Non-Intrusive Runtime Verification

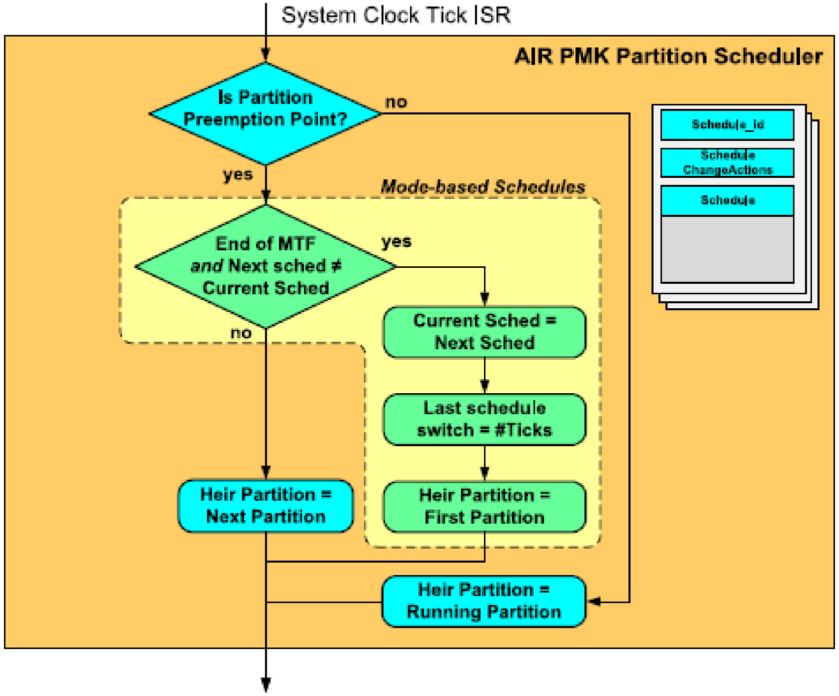
## AIR Observer




- Implemented as an hardware-based module
- System bus observation is non-intrusive
- Events are registered either statically or dynamically
- Activity developed under national Project READAPT, proceeded within COST Action IC1402 – Runtime Verification Beyond Monitoring (ARVI)

# Adaptability mechanisms: <u>Different PST schedules</u>

- Adaptation to different phases of the mission
  - takeoff;
  - approach flight;
  - exploration;
  - flight back;
  - landing.



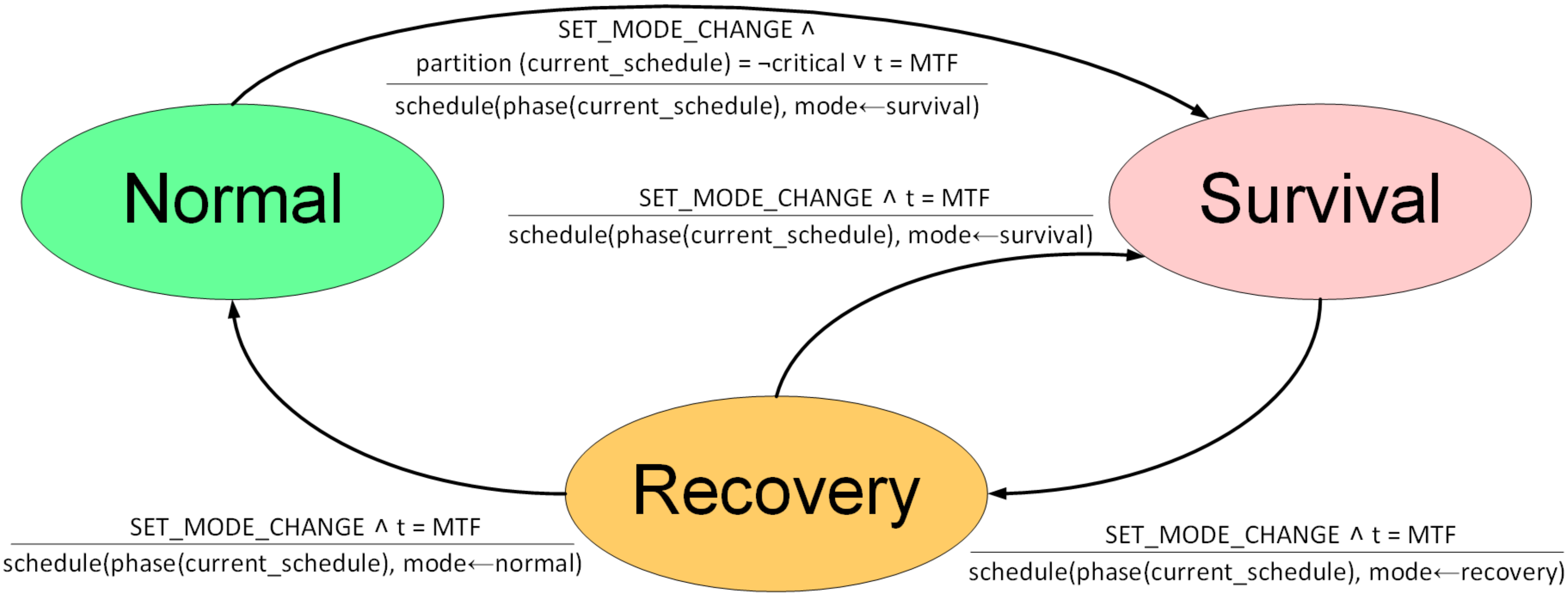

- Accommodation of component failure

José Rufino, LaSIGE/FCUL, Portugal <jmrufino@ciencias.ulisboa.pt>

# Adaptability mechanisms: Mode-changes

- No (mandatory) need to wait for the end of the MTF
- For each mission phase three schedules are defined (one for each mode)
- An approach similar to mission phase adaption is used (schedule a different PST)
- Only slight changes to AIR native scheduler are required

# Adaptability mechanisms
# Mode-changes



# Adaptability mechanisms

## Mode-changes

| Mode | Partition (and function) | | | |
|---|---|---|---|---|
| | AOCS | TTC | FDIR | Payload |
| Normal | Full | Full | Partial (detection) | Full |
| Survival | Full | Partial (e.g. abort) | Partial (detection) | only the required |
| Recovery | Full | Partial (e.g. normal) | Full | only the required |

AOCS – Attitude and Orbit Control Subsystem
TTC – Telemetry Tracking and Command
FDIR – Fault Detection, Isolation and Recovery

# Adaptability mechanisms

## Impact on the APEX interface

| Primitive | Short description |
|---|---|
| **Need to register/update critical execution period bounds in the AO** | |
| SET_MODE_SCHEDULE | Requests a mode change for a new schedule<br>Served if/when no critical activities |
| SET_PHASE_SCHEDULE | Requests a new mission phase schedule<br>Served in normal mode, at the end of a MTF |
| **No need to register/update critical execution period bounds in the AO** | |
| GET_MODE_SCHEDULE_ID | Obtains the current schedule identifier |
| GET_MODE_SCHEDULE_STATUS | Obtains the current schedule status |

# Integrating Adaptability and Non-Intrusive RV

| Functional system | Verification sub-system |
|---|---|
| Hardware-based | Software-based (minimum intrusiveness) |
| Software-based | Hardware-based (non-intrusive) |

- Partition scheduling and dispatching (trigger)
- Support  to phase-dependent and mode-based schedules
- Process deadline verification

**Algorithm 1** AIR Partition Scheduler with runtime verification featuring adaptation through mode-based schedules

```
▷ Entered upon exception: partition preemption point signalled by the AO
▷ Runtime verification actions
if (mode(currentSchedule) = mode(nextSchedule) ∧
      schedules_currentSchedule.table_tableIterator.tick ≠
      (currentTicks − lastScheduleSwitch) mod
      schedules_currentSchedule.mtf) ∨
      (mode(currentSchedule) ≠ mode(nextSchedule) ∧
      schedules_currentSchedule.table_tableIterator.critical >
      (currentTicks − lastScheduleSwitch) mod
      schedules_currentSchedule.mtf) then
    HEALTHMONITOR(activePartition)
else    ▷ Partition Scheduling Table (PST) and partition switch actions
    if currentSchedule ≠ nextSchedule ∧
          ((mode(currentSchedule) ≠ mode(nextSchedule) ∧
          (currentTicks − lastScheduleSwitch) mod
          schedules_currentSchedule.mtf ≥
          schedules_currentSchedule.table_tableIterator.critical) ∨
          ((currentTicks − lastScheduleSwitch) mod
          schedules_currentSchedule.mtf = 0)) then
        ▷ PST switch actions
        currentSchedule ← nextSchedule
        lastScheduleSwitch ← currentTicks
        tableIterator ← 0
    end if
    ▷ Partition switch actions
    heirPartition ←
        schedules_currentSchedule.table_tableIterator.partition
    tableIterator ← (tableIterator + 1) mod
        schedules_currentSchedule.numberPartitionPreemptionPoints
end if
```

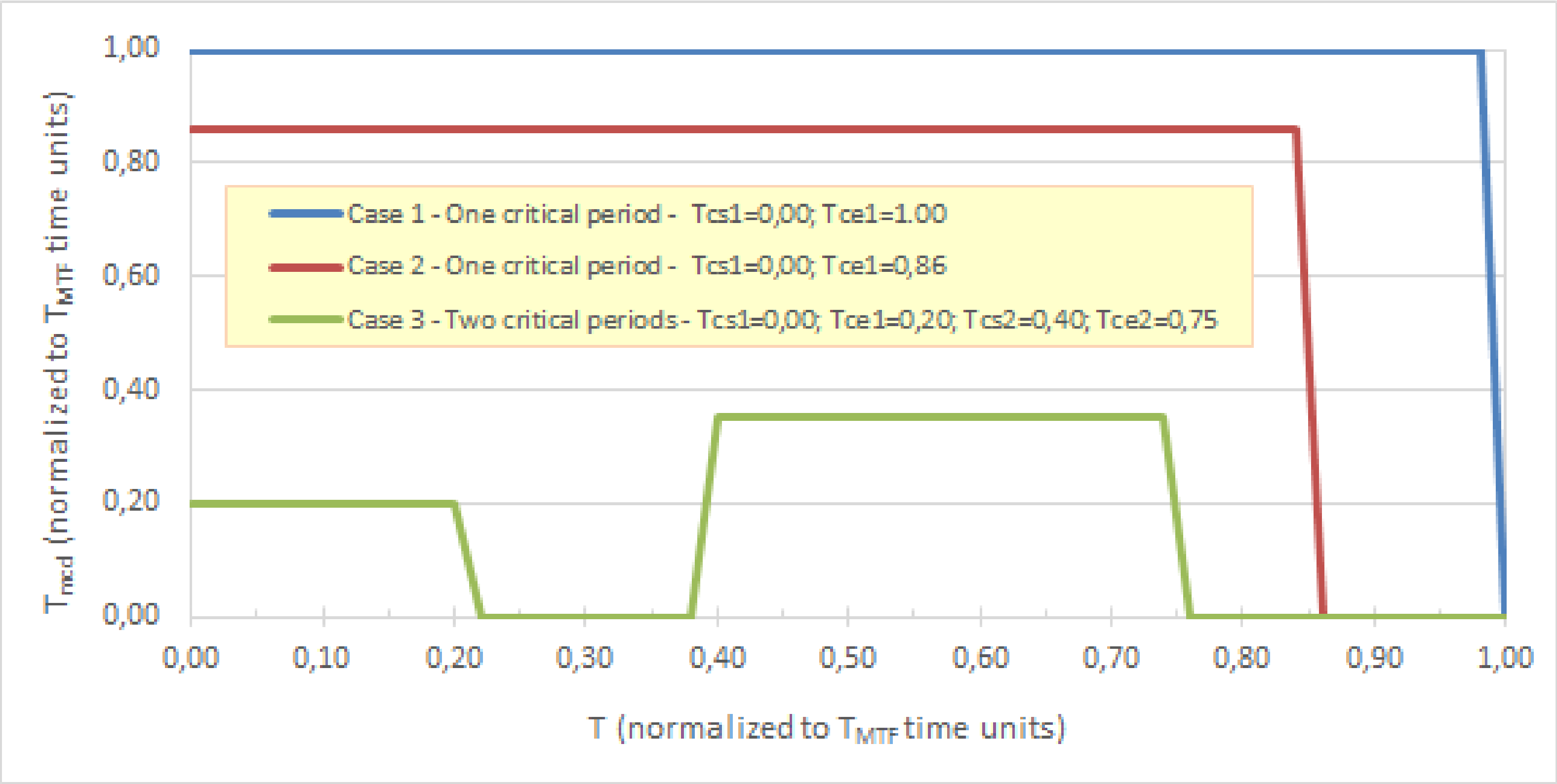
Triggered by hardware

Runtime verification
(software)

PST switch
(software)

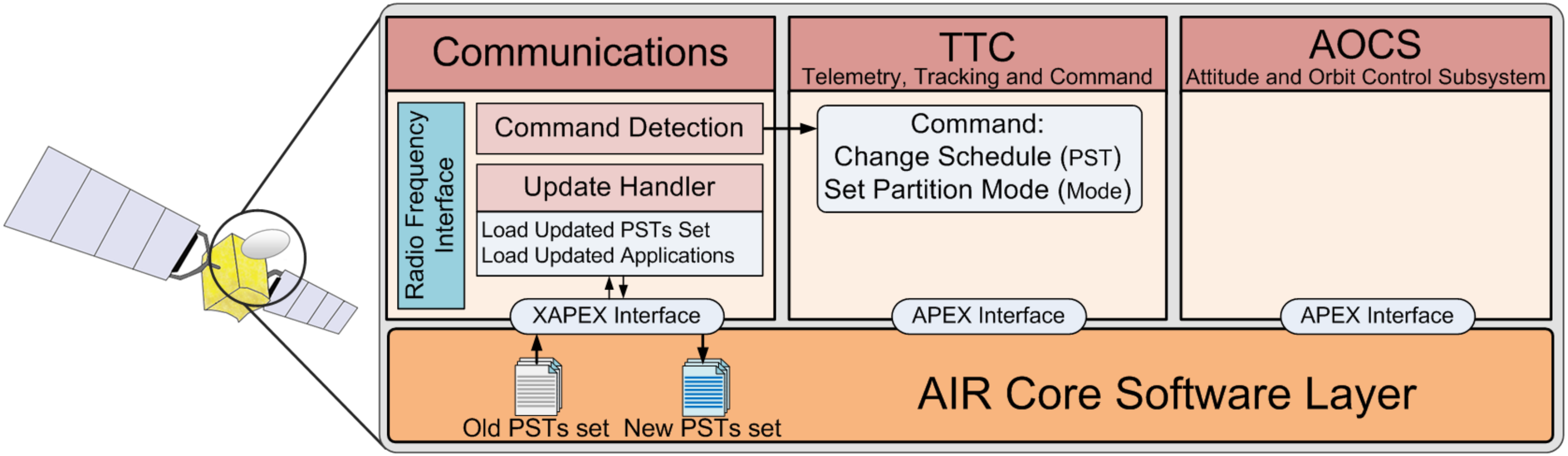
Partition switch
(software)

# Integrating Adaptability and Non-Intrusive RV



# AIR Prototyping Activities

## Mockup of Integration on Spacecraft Onboard Platform




José Rufino, LaSIGE/FCUL, Portugal <jmrufino@ciencias.ulisboa.pt>

# AIR – Prototype


José Rufino, LaSIGE/FCUL, Portugal <jmrufino@ciencias.ulisboa.pt>

# Conclusion

- Simple modifications to AIR native technology:
  - Improved (self-)adaptation features
  - Allows timely responses to unexpected events
  - Hardware-assisted approach
  - Complemented with software-based components

# Questions?

## Further information:

http://air.di.fc.ul.pt

http:www.navigators.di.fc.ul.pt/wiki/Project:READAPT

https:www.cost-arvi.eu

**José Rufino**

**LaSIGE/FCUL, Lisboa, Portugal**

**Homepage:** http://www.di.fc.ul.pt/~ruf/

**E-mail:** jmrufino@ciencias.ulisboa.pt