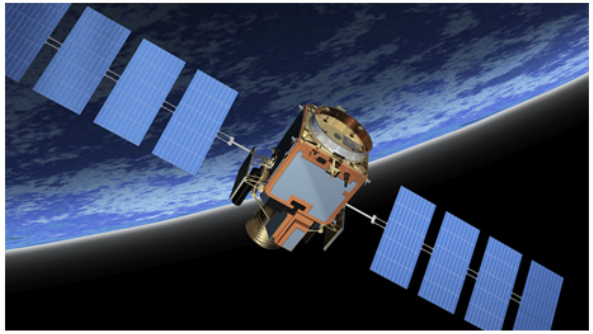
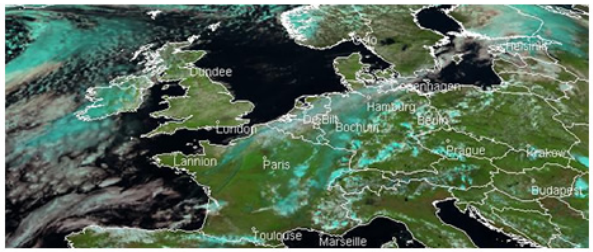
IMA Kernel Qualification preparation

Mark Hann, Fabrice Cros, Regis De Ferluc – Final Presenation

SCISYS, AIRBUS, TAS, Trinity College Dublin

9th June 2016





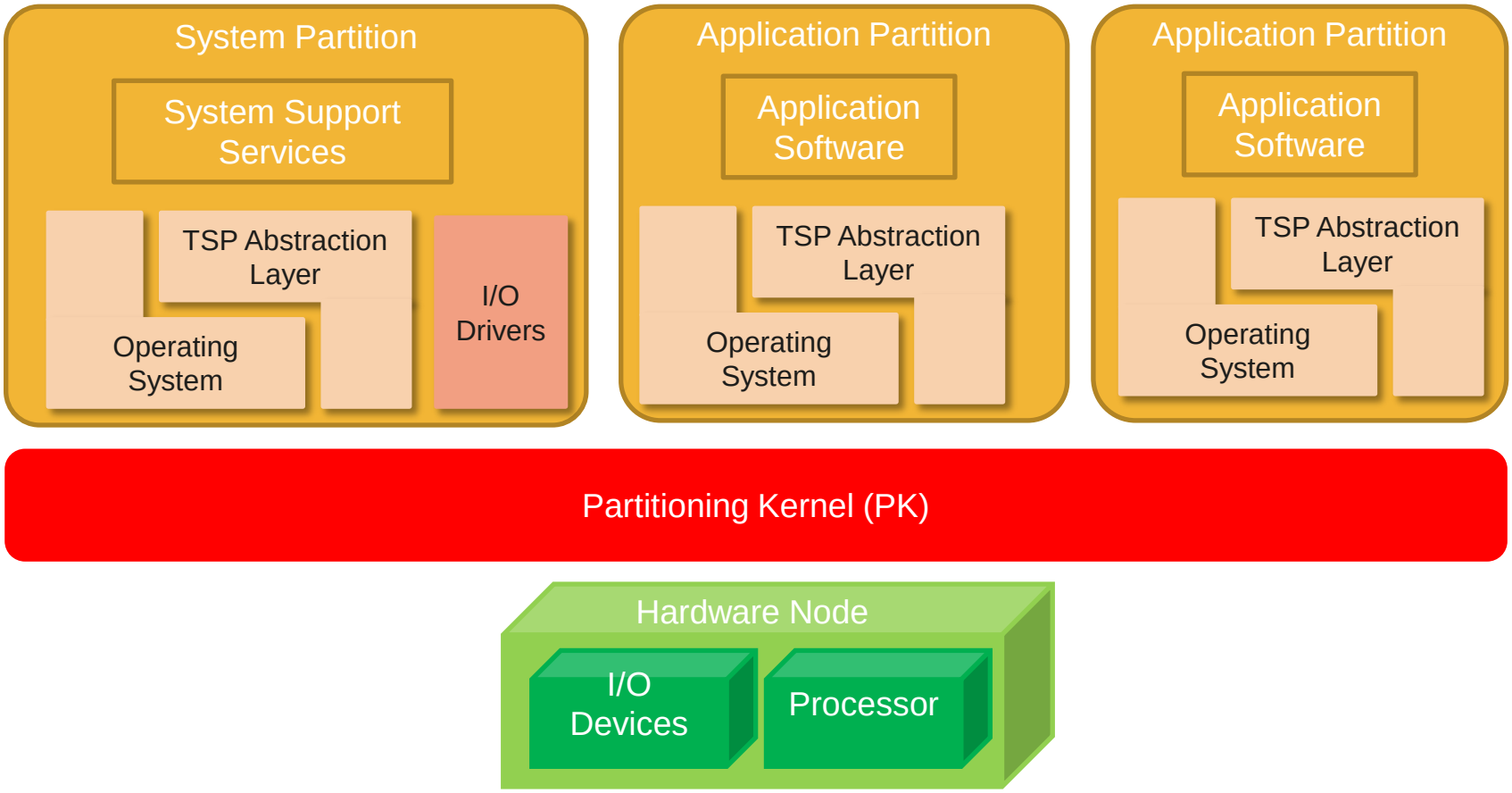
Kernel Qualification Preparation Objectives

- Define Partitioning Kernel Requirements for the European Space Domain.
- Define how the Partitioning Kernel will be qualified in European Space Domain
  - » ESA Study
- Consortium:

Project Lead:	SCISYS
Phase 1:	AIRBUS DEFENCE & SPACE
Phase 2:	ThalesAlenia Space
Phase 3:	SCISYS

- Supported by: cnes Trinity College Dublin

Scope of Qualification



SCISYS

Kernel Qualification Preparation

- **Phase 1: Consolidation of requirements baseline, for IMA Partitioning Kernels**
  - » Conformance assessment of shortlisted kernels, against requirements baseline.
- **Phase 2: Overall qualification strategy and detailed test plan for qualification of IMA Separation Kernels**
  - » Identification of suitable methods and techniques for verification and validation.
    - › Formal Methods Activities Described in Separate Presentation
- **Phase 3: Experimentation and evaluation of the defined qualification strategy**
  - » Evaluation of adequacy and appropriateness of the selected methods and techniques.

Kernel Qualification Preparation

- **Phase 1: Consolidation of requirements baseline, for IMA Partitioning Kernels**
  - » Conformance assessment of shortlisted kernels, against requirements baseline.

Presenter: AIRBUS DEFENCE & SPACE

Requirement Baseline

- **Requirements are segregated between:**
  - **CORE** Requirement: requirement specifying an essential functionality of the partitioning kernel under specification
  - **EXTENDED** Requirements: requirement specifying a non-essential or highly specific functionality of the partitioning kernel under specification. These requirements are second priority requirements that can be de-scoped in a qualification process.
- **Multi-Core Requirements:**
  - Defined in annex of the document
  - Most (>95%) of the single-core requirement are also multi-core requirements
- **EXTENDED and Multi-Core Requirements have not been considered for the next phase**
  - Targets are only single core processors
  - Only core requirements are mandatory in the follow-up project

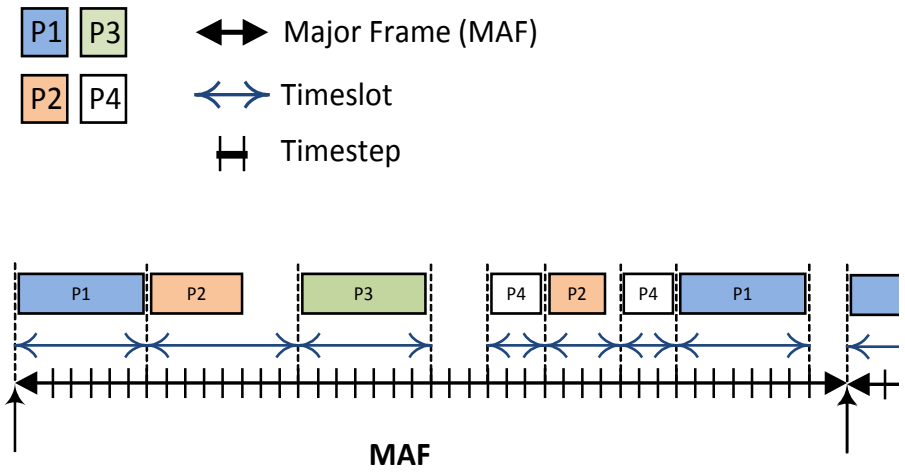
Requirement Baseline

- **Input documents**
  - ARINC 653 standard
  - IMA-SP requirements baseline
  - LVCUGEN software System Specification

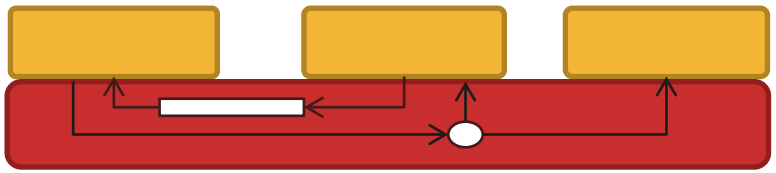
- **Additional concepts**
  - Authorized partition
  - Synchronization on external interrupts
  - Zero copy IPC
  - Virtualized interrupts

Basic Time and Space Partitioning concepts

- **Space partitioning**
  - » Using hardware MMU to enforce memory segregation
  - » Memory areas authorized (or not) by configuration at design time
- **Time partitioning**
  - » Cyclic Scheduling plan
  - » Major Frame (MAF)



- **Inter-partition communication**
  - » Queuing ports
  - » Sampling ports



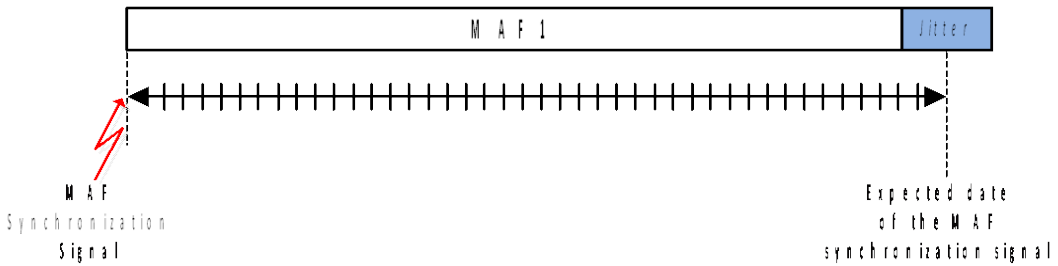
Authorized partition

- **Special rights granted**
  - Start/stop/restart a partition
  - Change scheduling plan
  - Access the mode of a partition
  - Read access to the configuration of all ports
  - Read access to the Health monitoring logs
- **Benefits**
  - Allow centralized FDIR in a partition
  - Avoid complexity for the configuration of kernel services
  - Transfer of Health Monitoring event log to ground segment

Synchronization on external interrupt

- **Feature**
  - » Avoid desynchronization after some time
  - » Small jitter window need to be handled by kernel

- **Benefits**
  - » Resynchronization of on-board time with ground and equipments
  - » Very precise signal if PPS from GPS is used



SCISYS

Zero copy inter-partition communication

- **Extended requirement**
- **Benefits**
  - » Avoid copying three times the message to exchange
  - » Useful for large messages
- **Write sequence**
  - » On request kernel provides a memory buffer address
  - » The partition writes its data in the memory buffer in user mode
  - » The partition informs the kernel the writing operation is finished
- **Read sequence**
  - » On request kernel provides a memory buffer address
  - » Partition reads the data from the buffer
  - » The partition informs the kernel the reading operation is finished

Virtualized interrupts

- **External interrupts allowed**
  - During the execution of the authorised partition
  - Only through virtual interrupts (kernel service)
  - No access to the actual hardware registers
  - Integrator responsibility to avoid interrupt sharing
  - Each partition can mask/unmask/clear interrupts
- **Extended interrupts**
  - Created by the kernel
  - Avoid polling communication ports
  - Extended requirement

Evaluation of compliance from existing kernels

- **Evaluated separation kernels**
  - » Xtratum from FentiSS
  - » PikeOS from SYSGO
  - » AIR from GMVIS
- **Compliance to**
  - » The requirement baseline
  - » ECSS E40 and Q80
- **Auto evaluation by kernel suppliers**
  - » Not possible to cross check for closed source hypervisors
  - » Partially cross checked for Xtratum as it is open source
- **Results provided under NDAs**

Phase 2: Qualification Approach

- **Specify a Qualification Strategy and Plan**
  - » Qualification to Category B Software
- **Explore Methods and Techniques**
  - » Classical Methods
  - » Model Based Testing
  - » Formal methods
- **Specify a Test Plan**

Presenter: ThalesAlenia Space — A Thales / Finmeccanica Company

Qualification Strategy and Plan

- **Qualification objectives:**

Objective is to provide adequate confidence to the customer and to the supplier that the PK software satisfies its requirements throughout the system lifetime.

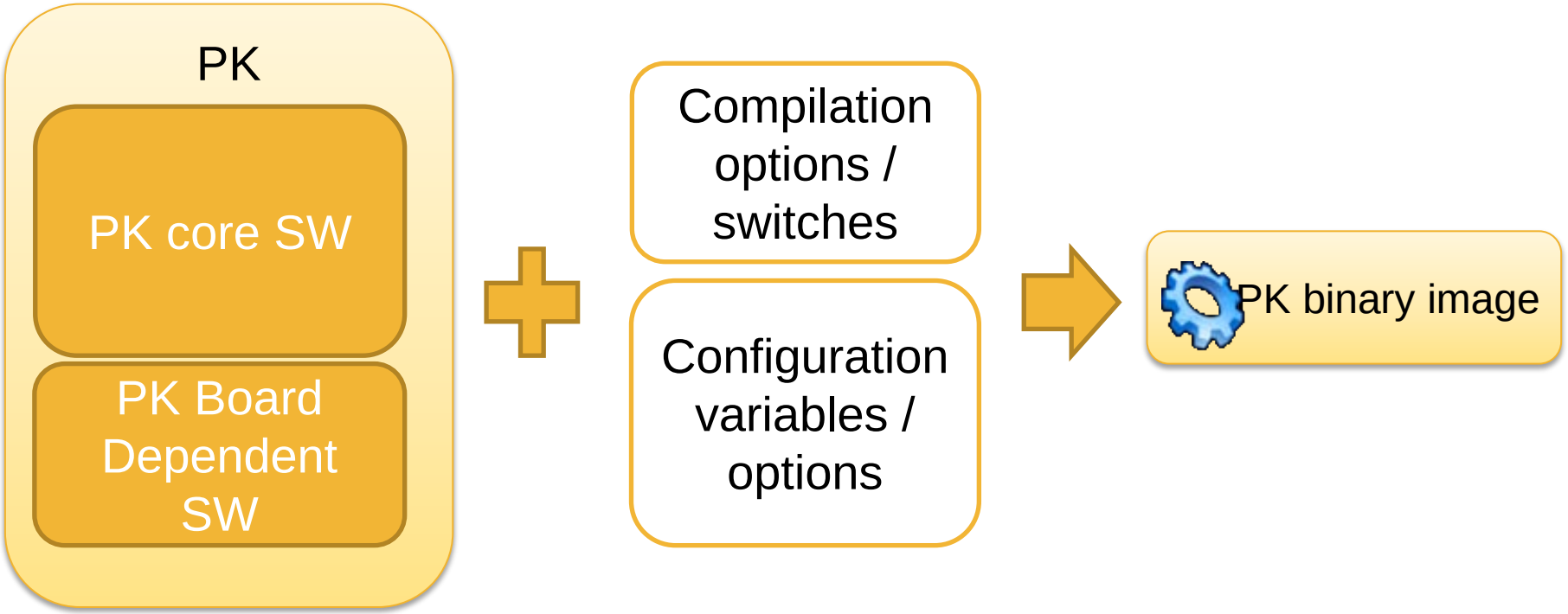
Qualification Strategy and Plan

- **Qualification strategy:**
  - Analysis of the ECSS standards
    - ECSS-E-ST-40C
    - ECSS-Q-ST-80C
  - Analysis of qualification/certification process in other domains
  - Definition of the qualification perimeter
  - Definition of a quality model
  - Definition of a Test Plan
  - Definition of the Qualification Plan

  - ITT requirements :
    - Qualification criticality level B (ECSS)

Qualification Strategy and Plan

- **Partitioning Kernel Qualification perimeter:**

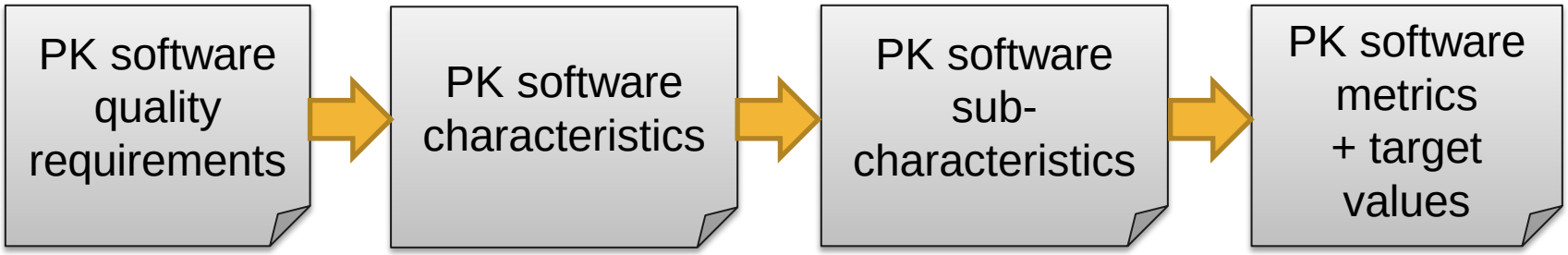


Qualification Strategy and Plan

- **Partitioning Kernel Qualification credits**

Qualification Strategy and Plan

- Quality Model



- Derived and tailored from ECSS-Q-HB-80-04A, March 2011, Space Product Assurance Handbook

- Example of applicable metrics: V&V coverage, Adherence to coding standards, Suitability of development documentation, code complexity metrics, Process Assessment, …

- Example of non-applicable metrics : efficiency metrics, reliability metrics, Project Management effectiveness, …

Qualification Strategy and Plan

- **Qualification of existing PK software in the frame of ECSS standards**

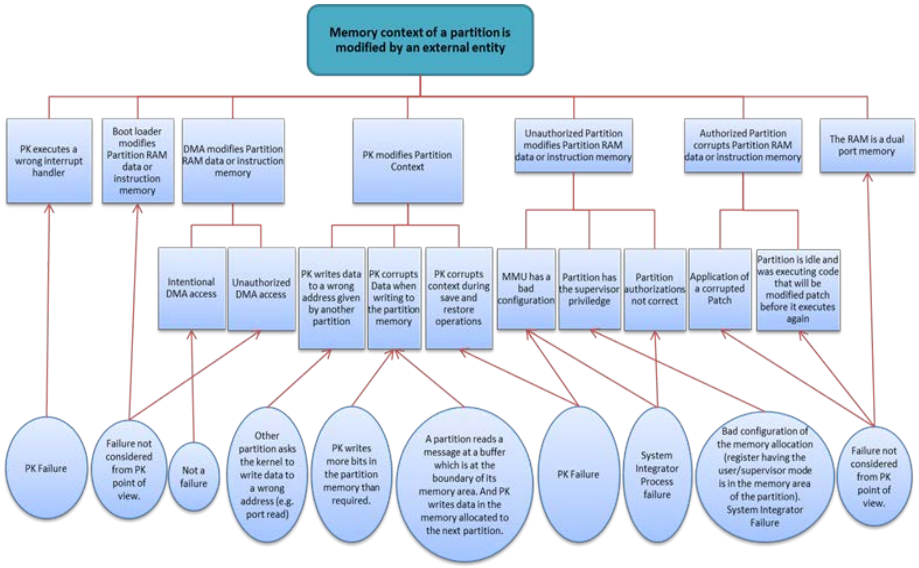
  - *For software products whose life cycle data from previous development are not available and reverse engineering techniques are not fully applicable, the following methods are applied:*
    - *1. generation of validation and verification documents based on the available user documentation (e.g. user manual) and execution of tests in order to achieve the required level of test coverage;*
    - *2. use of the product service history to provide evidence of the product's suitability for the current application.*

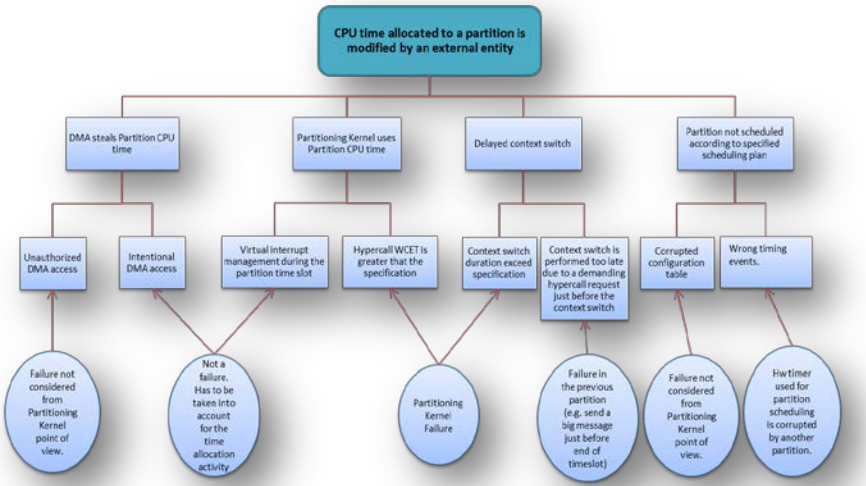
    SPEC Method improvement

  - If additional development are required to upgrade the PK to reach compliance to [D02] prior the qualification, it should be done according to ECSS process unless modification impact less than 20% of the code. In this case, the Software Reuse File approach is followed.

Qualification Strategy and Plan

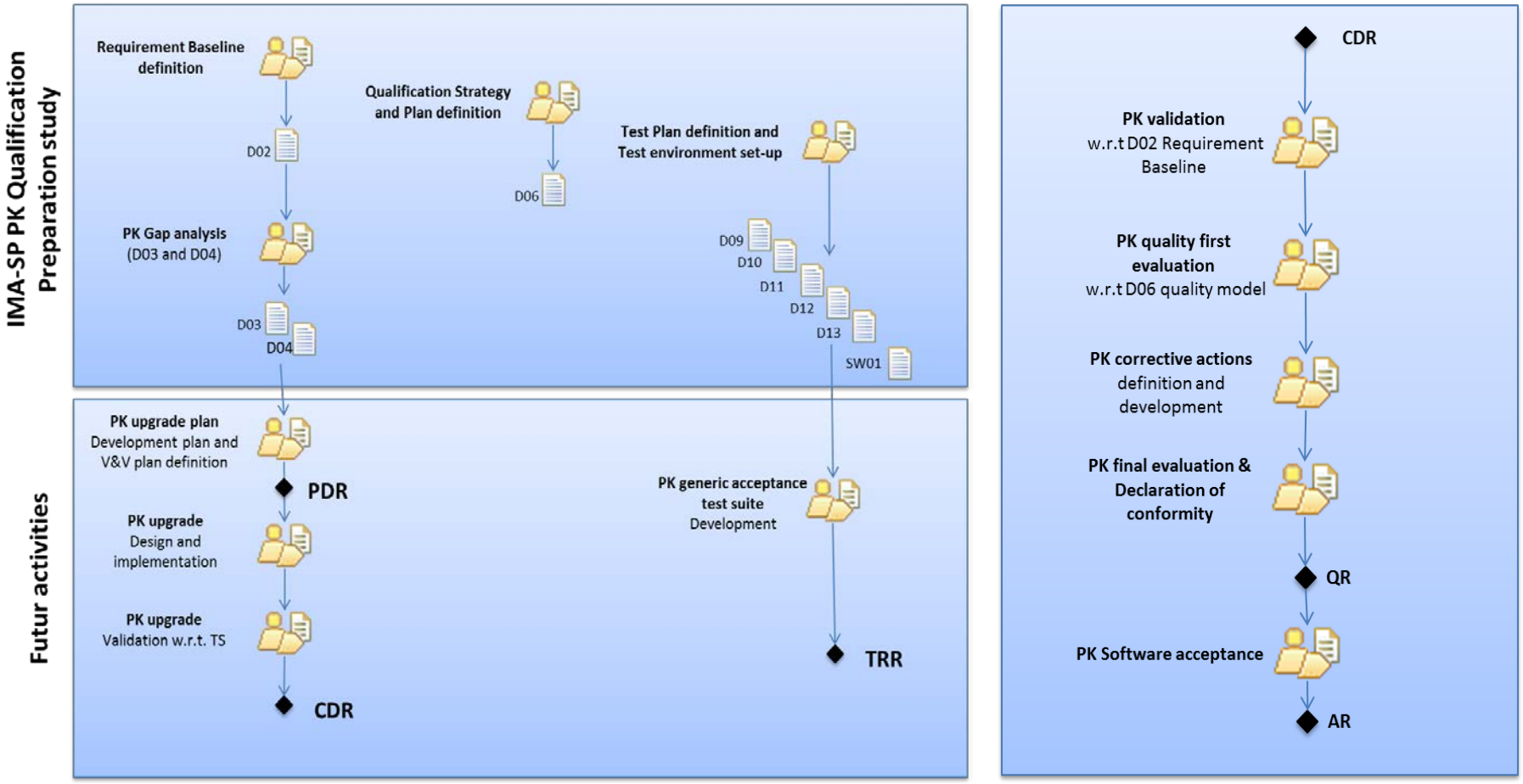
- Safety assessment
  - A dependability analysis for aspects related to time and space partitioning shall be performed.
  - This provides inputs to System and Software Safety assessment when PK is used in a project.





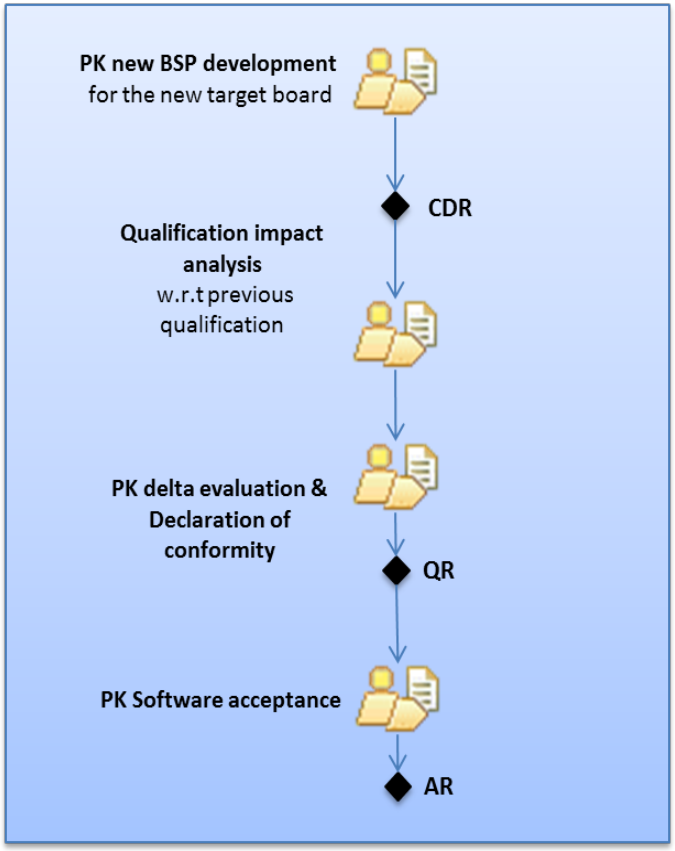
Qualification Strategy and Plan

- **Qualification Process overview**



Qualification Strategy and Plan

- **Qualification Process overview: new target board**



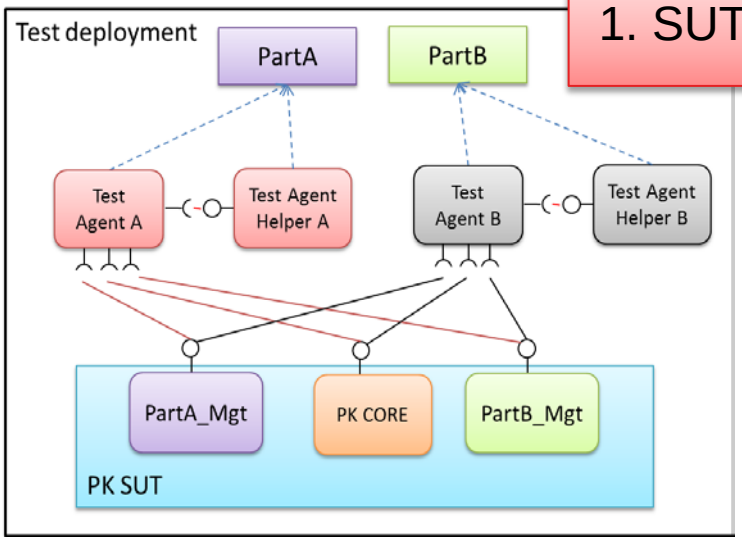
Model Based Testing techniques and Investigations

- **State of the art:**
  - » Test generation from behavioural model of the system
  - » Test script (code) generation from models of the tests (abstract test specification).

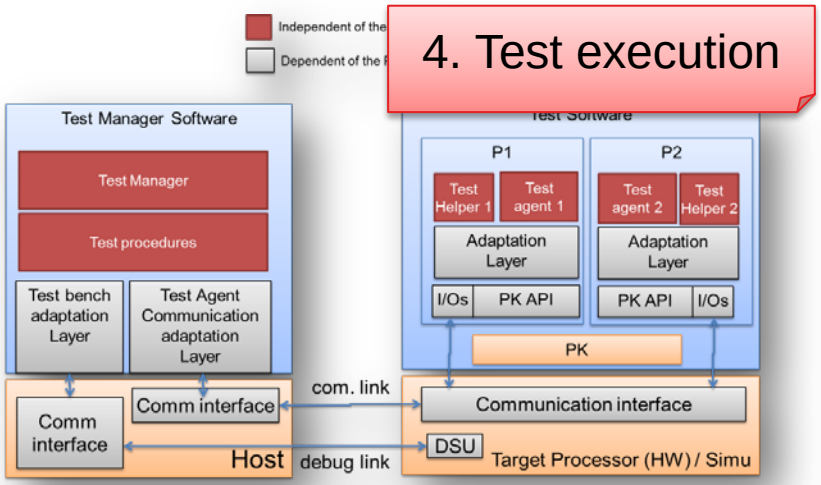
- **Proposed approach**
  - » Capture the System Under Test interfaces as models
    - › PK API
    - › HW platform interfaces
  - » Propose a formal Abstract Test Specification language
    - › Textual syntax
    - › Reference to the models by names
  - » Elaborate a test script production process
  - » Identification of expected benefits

Model Based Testing techniques and Investigations

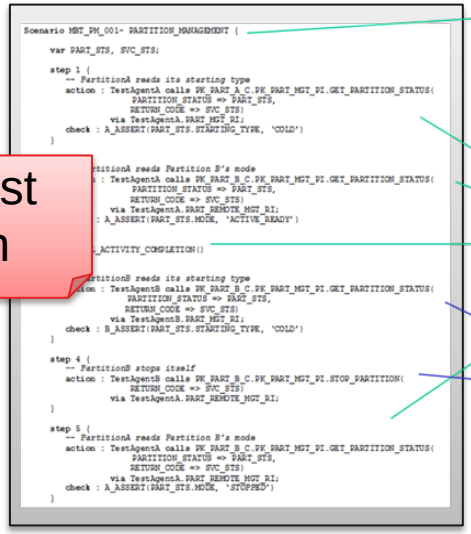
1. SUT modelling



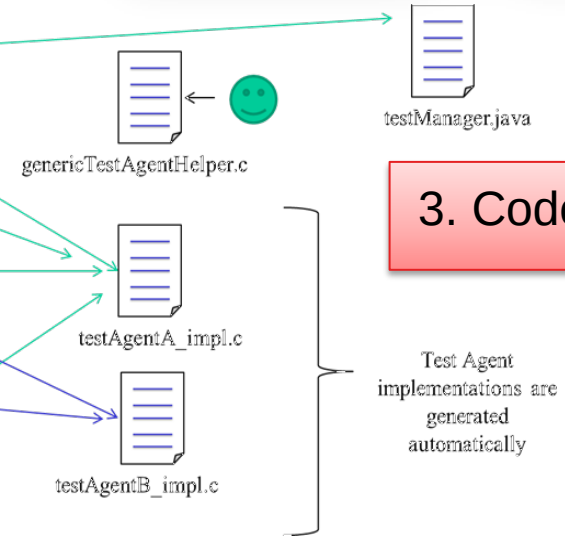
4. Test execution



2. Abstract test specifciation



3. Code generation



Model Based Testing techniques and Investigations

- **Expected Benefits**
  - » Test specification is aligned with system high level specification
  - » Test scripts development effort saving
  - » Test scripts aligned with test specification
  - » 3 PKs to test from a unique test plan
  - » Easier to adapt to a new target platform / CPU architecture
- **Pre-requisite:**
  - » Tooling !

Test Plan

- **Generic Test plan (not PK specific)**
- **Single Core requirements only**
  - » Core and Extended
- **117/132 Requirements are validated by test**
  - » 15 by Review & Inspection
- **SW Under test is PK and its Configuration, therefore 2 types of test:**
  - » Execution of PK test
  - » Configuration tests (allowed/not allowed)
- **37 Tests Specified**

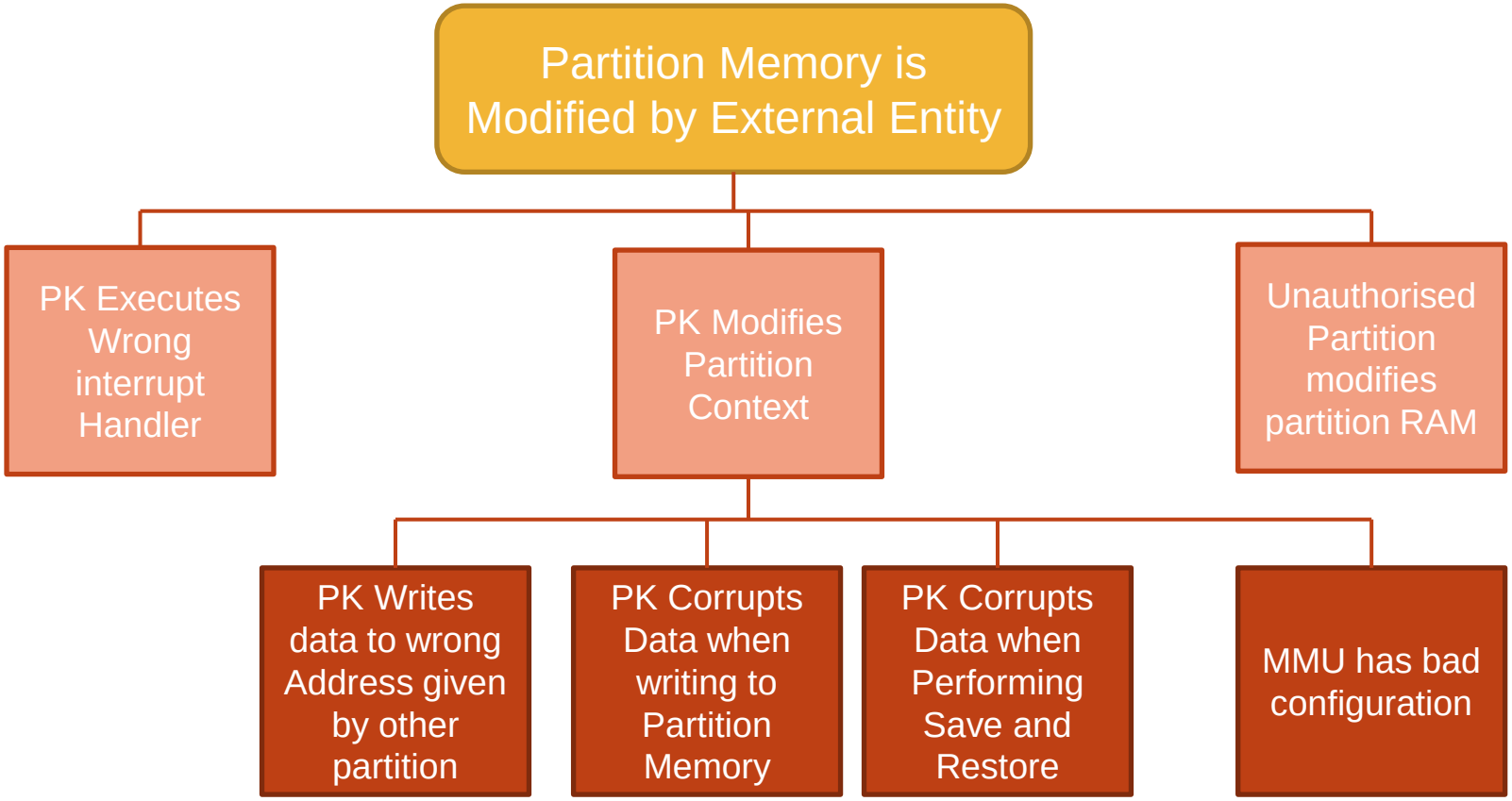
Presenter: SCISYS

Types of Tests

- **Validation of Requirements**

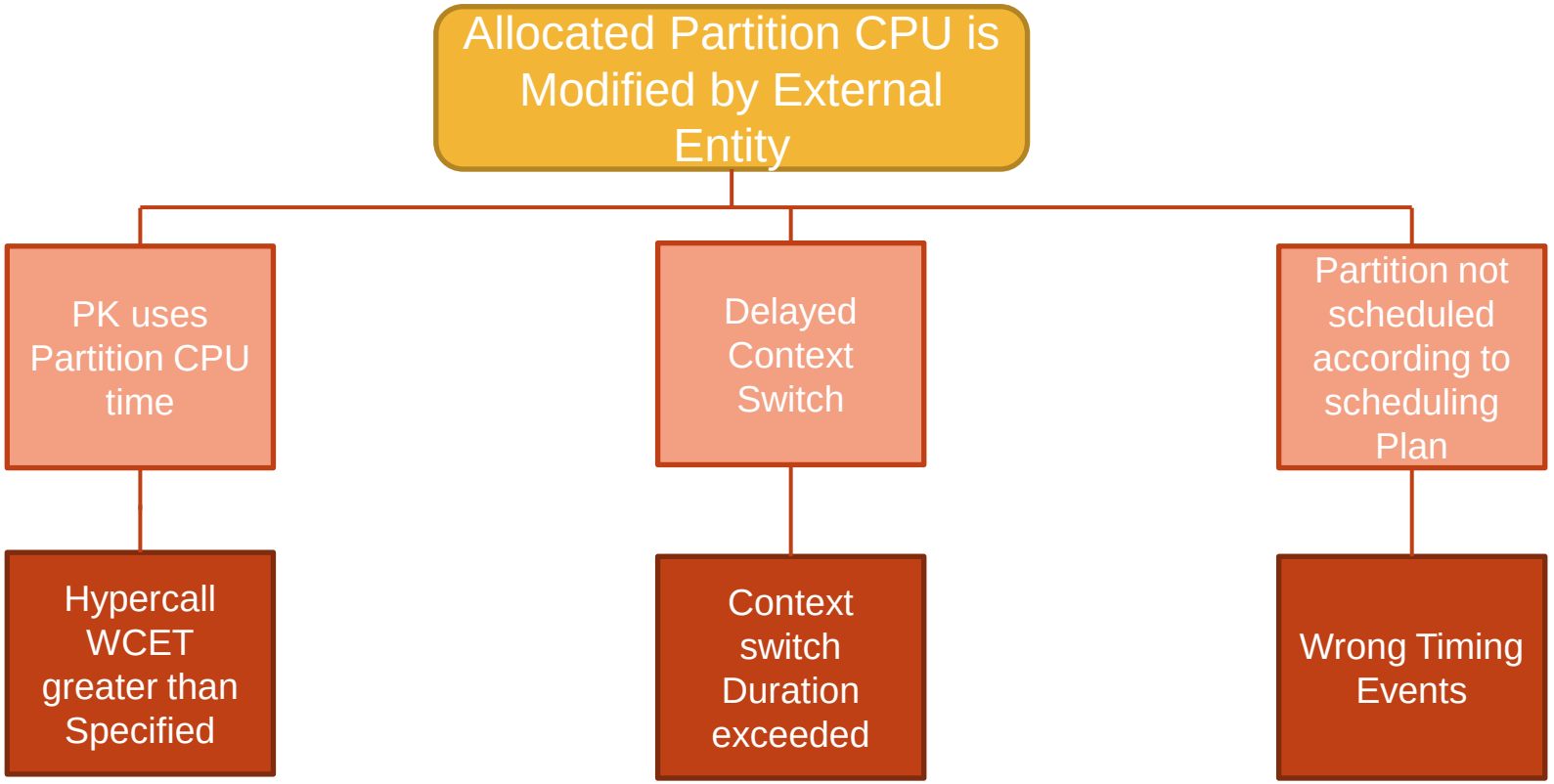
- **Fault Injection Tests**
  - » Invalid API errors
  - » Configuration Errors
- **Load/Stress Tests**
  - » Worse case configuration
  - » Long duration
- **Robustness Tests**
  - » Handling of exception traps
  - » Partition Overrun
- **Time and Space Partitioning Requirements**
  - » Fault Tree Analysis performed
  - » Failures Tested or by Inspection/Review performed

Space Partitioning Analysis



SCISYS

Time Partitioning Analysis



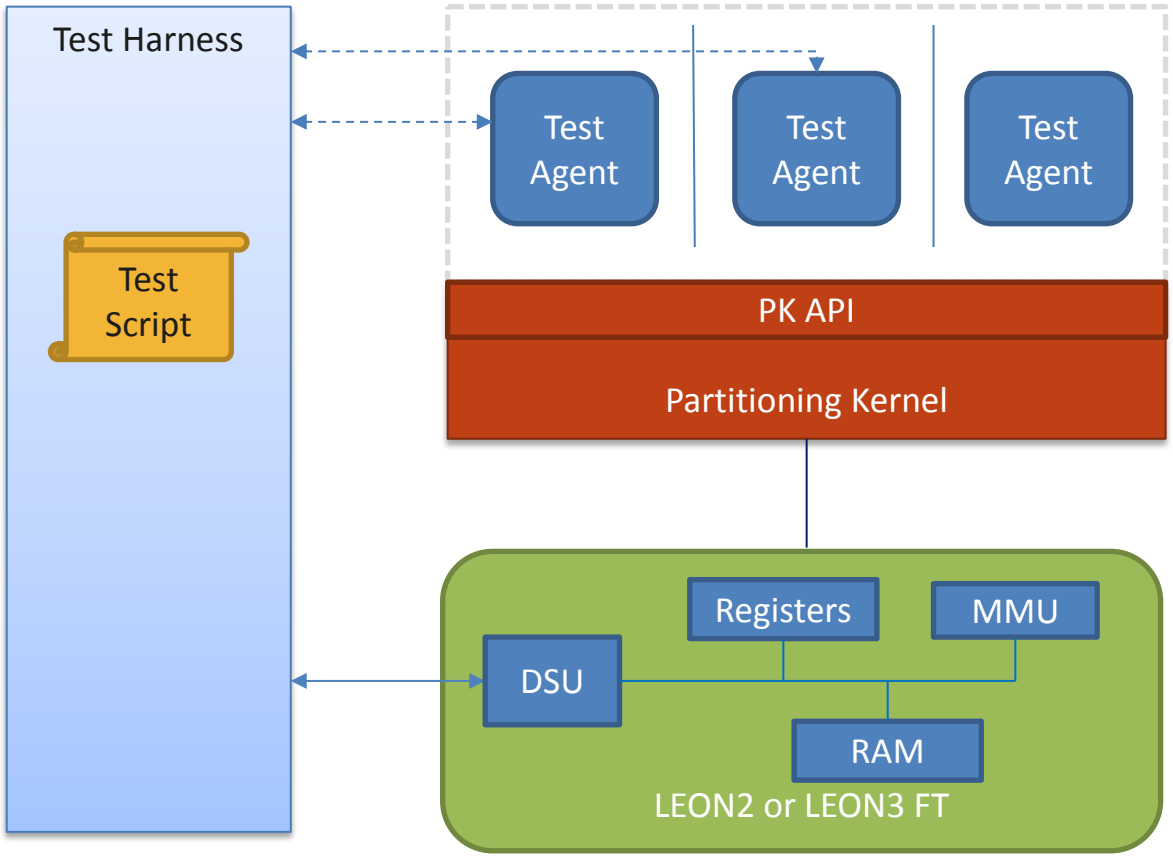
SCISYS

Phase 3: Case Study

- **The Objectives were to:**
  - » Define the Environment
  - » Define all Test Artefacts
  - » Explore benefits and limitations of selected methods:
    - › By Executing a subset of the test plan

Presenter: 

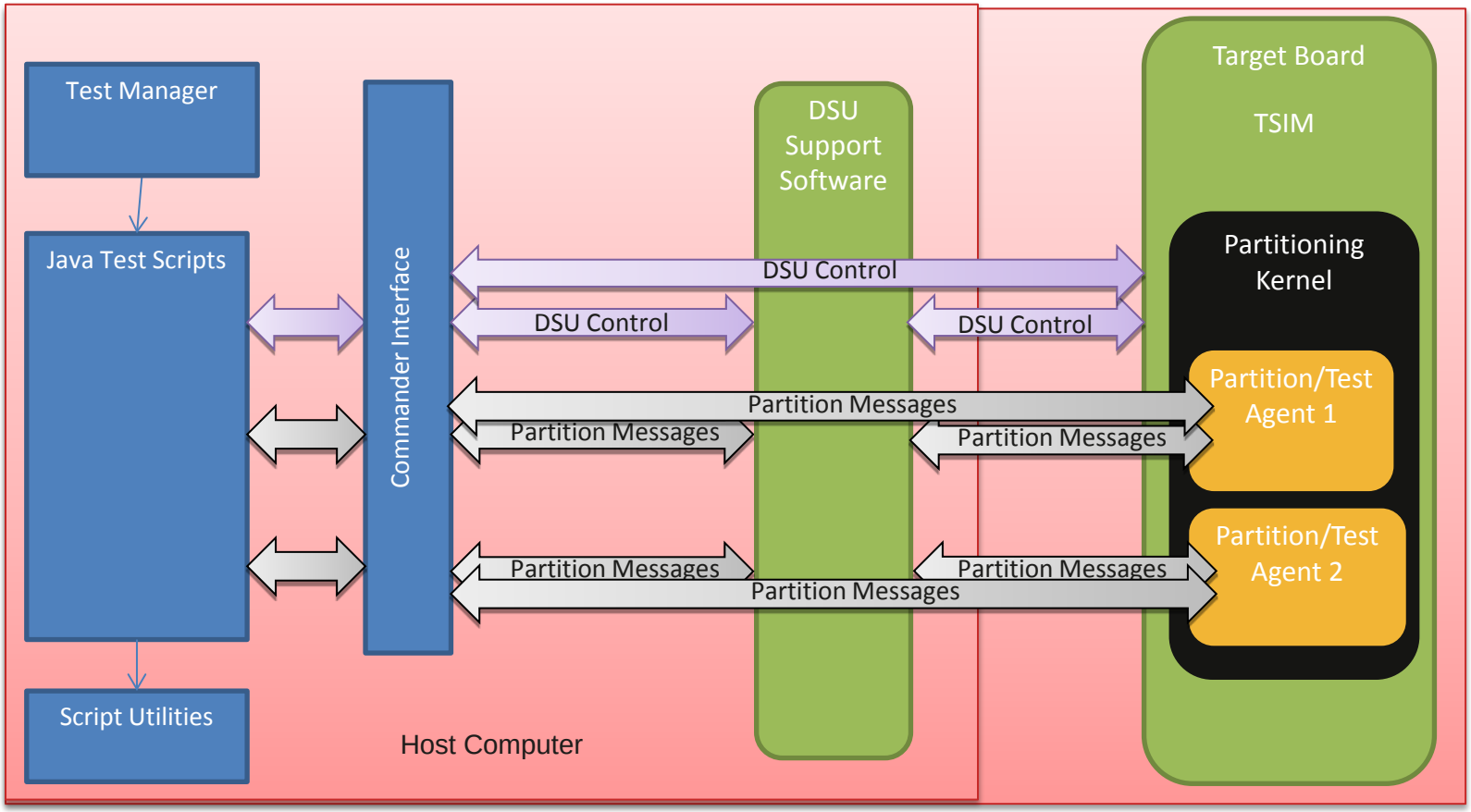
SW Under Test Context



Test Environment

- **Test Target is:**
  - LEON2 or LEON3FT (both with MMU)
  - Particular Processors identified:
    - SCOC3
    - UT699
    - COLE
- **LEON2 & 3 processors which have non-intrusive Debug Support Unit (DSU).**
  - Full access to all processor registers (e.g. PSR, TBR, ITP, etc.);
  - Full access to AMBA bus memory;
  - Breakpoint management;
  - Instruction level run control (break, step, continue etc.)
- **Java Environment for Test Harness**
  - Test Agent in C
- **Bare Partition (no Guest OS).**

Test Setup

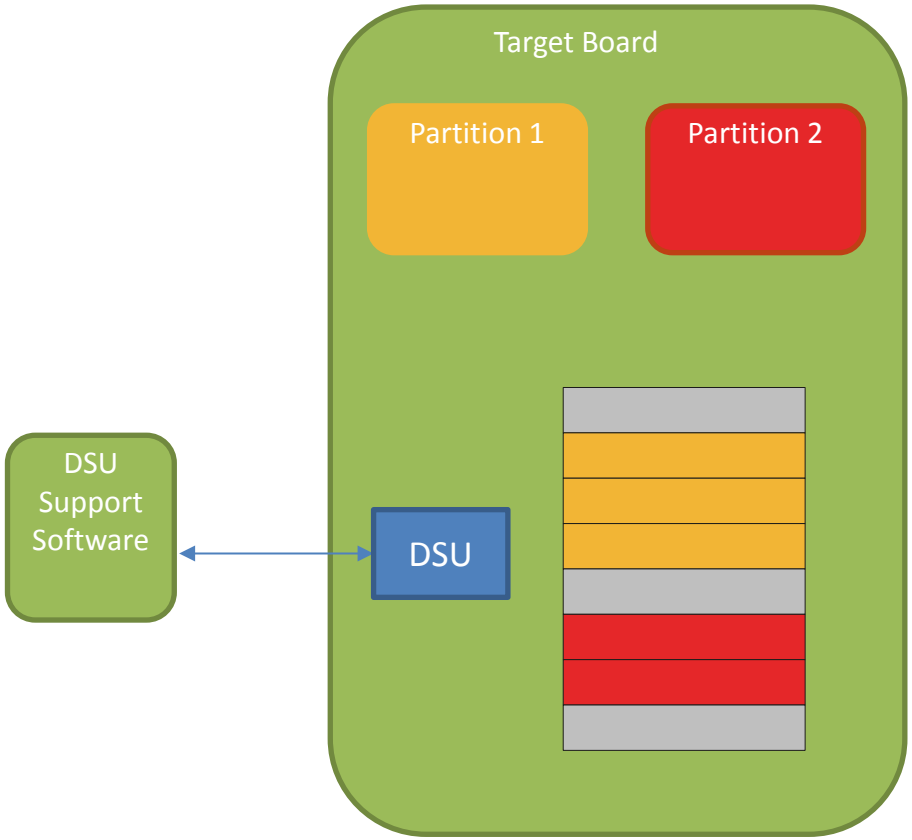


SCISYS

Test Cases

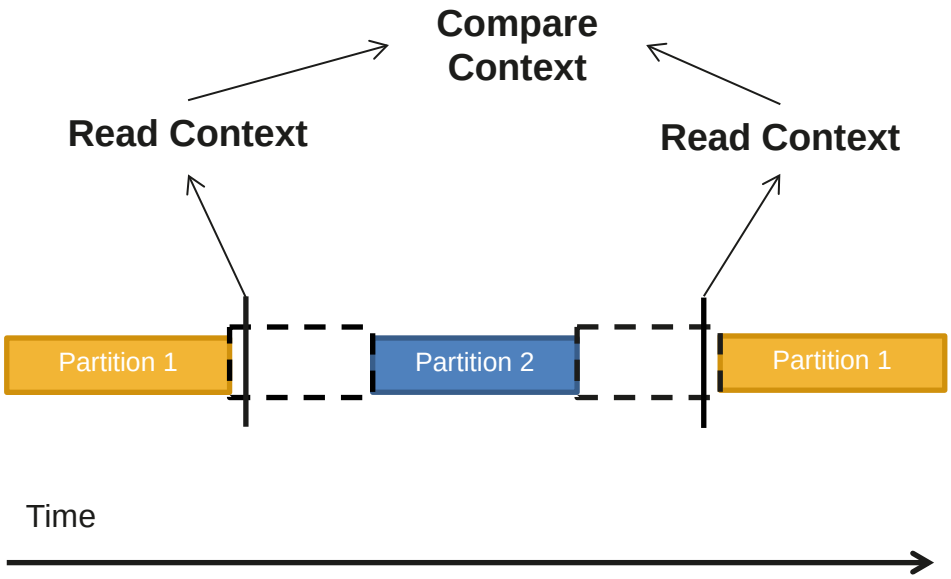
- **Test Target was Xtratum**
- **Case 1 : Test Environment Checkout;**
- **Case 2 : Memory Access test;**
  - » Demonstrates the memory permissions for each region.
  - » Test the MMU configuration using a DSU command
- **Case 3 : Context switch test;**
  - » Demonstrate test of context save and restore
  - » Demonstrate low level break-pointing and control using DSU
- The experimentation was performed on an emulator but the test environment has been designed to allow all the tests to be executed on the hardware.

Memory Access Test



- DSU can interrogate the memory status of all 4KB pages

Context Switch Test



- Most registers need to be included in context
- Registers read and then compared
- Halting processor at precise time difficult.
  - » Breakpoint set on Xtratum function
- Unexpected behaviour when floating point registers restored

Case Study Results

- Test approach proposed in Test Plan is feasible;
- 11 Test techniques identified and 8 techniques were experimented
- The test environment has been demonstrated;
- Use of DSU helps porting between different boards;
- Access to source code important.
- Updates to test plan from lessons learnt during case study

Conclusion

- **Partition Kernel Requirements Baseline defined**
  - » Specific for needs for European Space
- **Plan and Strategy for Qualification Approach Defined**
- **Test Plan and Test Environment defined**
- **Case study on a subset of the Tests performed.**
  - » The validation approach and techniques are feasible

Next Steps

- Qualification of one or more PK (Xtratum, PikeOs, AIR)
- Explore/prototype tools to apply FM/MBT for other kind of software parts (ASW, EP, …)
- Consolidate requirement baseline for multi-core processors
- Extend qualification strategy/plan and test plan to multi-core processors

Any Questions?

Mark Hann

SCISYS UK Ltd
Clothier Road, Bristol
BS4 5SS, UK

Direct: +44 1179 916 5144

mark.hann@scisys.co.uk
www.scisys.co.uk

ESA TO: **Maria Hernek**
Maria.Hernek@esa.int

Julien Galizzi
CNES
T: +33 561281515
Julien.Galizzi@cnes.fr

Fabrice CROS
Alexandre Cortier
Airbus Defence and Space
T: +33 56 219 8203
Fabrice.CROS@airbus.com
Alexandre.CORTIER@airbus.com

Régis De Ferluc
Thales Alenia Space France
T: +33 49 228 9945
regis.deferluc@thalesaleniaspace.com

Andrew Butterfield
Trinity College,
University of Dublin
T: +353 1 896 2517
Andrew.Butterfield@scss.tcd.ie