MULTICORE SW ARCHITECTURE AND VALIDATION APPROACH

D. TEODONIO – A. BUFALINO – T. DI COCCO – F. D'ANTONIO – S. CANDIA
COMPETENCE CENTER SOFTWARE ENGINEERING – ITALY

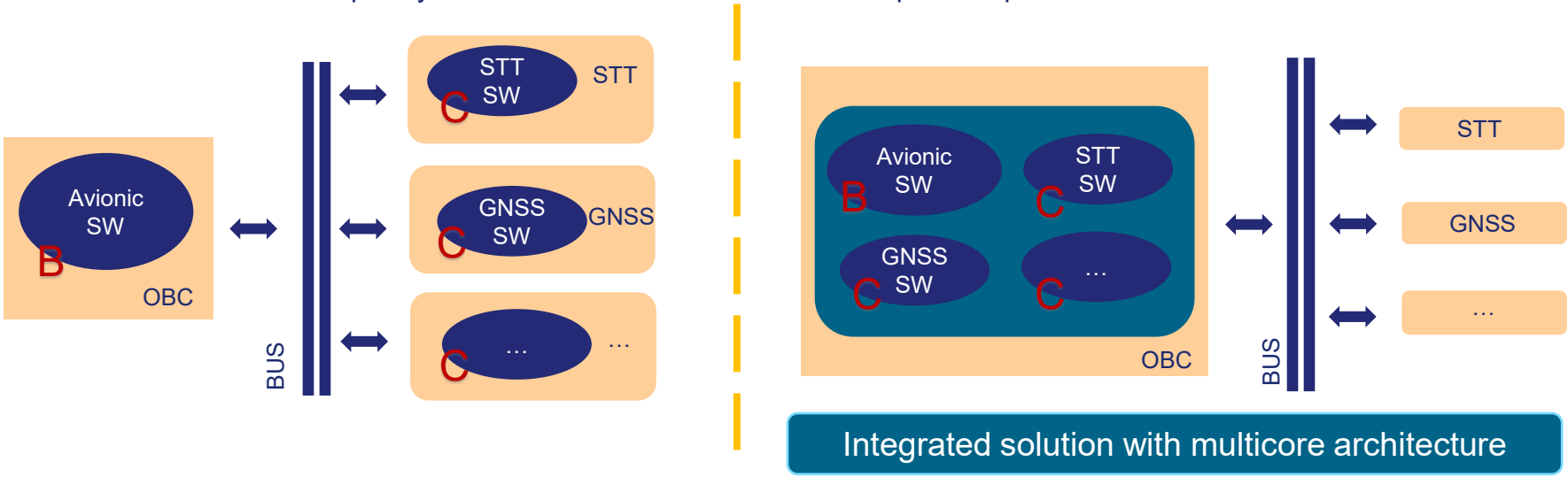
Date: 15/11/2023
Ref: 17th ESA ADCSS 2023
Template: 83230347-DOC-TAS-EN-011

TABLE OF CONTENTS

INTRODUCTION

/// In 2010 TASI started the definition of a new generation of avionics aims to integrate several processing tasks previously distributed among many computers (e.g. Star Tracker processing, GNSS Navigation processing, ..) around the satellite.

- A reduction of satellite mass and volume, thus ensuring an overall cost reduction.
- An increase of SW complexity as different SW share the same computation platform.



Date: 15/11/2023
Ref: 17th ESA ADCSS 2023
Template: 83230347-DOC-TAS-EN-011

NEW GENERATION OF TASI ON-BOARD COMPUTERS

/// Next generation of TASI's OBCs:

- GR740 – ESA's NGMP selected as the major building block (4xLEON4 SPARC V8 @250MHz, up to 1700 DMIPS).
- Highly integrated unit including GNSS receiver and optionally I/O module and PL Mass Memory.
- Includes CFDP HW implementation for high performance.

/// OBC Core

- Multi-Core Processor Module (**MCPM**).
- Telemetry Telecommand Mass memory and Reconfiguration module (**T2MR**).

TOWARDS MIGRATION TO MULTICORE ARCHITECTURE (1/2)

/// Main Goals

- OBC resources sharing.
- Failure isolation to prevent error propagation among the SW.
- Decouple the applications to maximize the reuse across several projects.

/// Time and Space Partitioning (TSP)

- Execution of applications with different criticalities on the same computation platform.
- A fault in a specific application does not affect the others.

/// GR740 and Time and Space Partitioning

- The design of GR740 allows the implementation of TSP over multi core architecture (e.g. LEON4 MMU, IOMMU, L1 cache with AHB bus snooping support).

TOWARDS MIGRATION TO MULTICORE ARCHITECTURE (2/2)

/// Hypervisor

- The hypervisor implements the TSP concept.
- Each application is executed within a hypervisor partition.
- The time isolation is ensured by the hypervisor time partitioning mechanism that allocates CPU time among the partitions.
- The space isolation is ensured by assigning specific memory sections to each partition and the hypervisor will configure the MMUs and IOMMU according to such configuration.
- Data exchange between partition is allowed through the IPC mechanism provided by the hypervisor:
  - Queuing & Sampling Ports
  - Shared memory
- **PikeOS** (SYSGO) has been identified to support the TSP in TASI multicore software for high-rel Missions.

PikeOS (SYSGO)

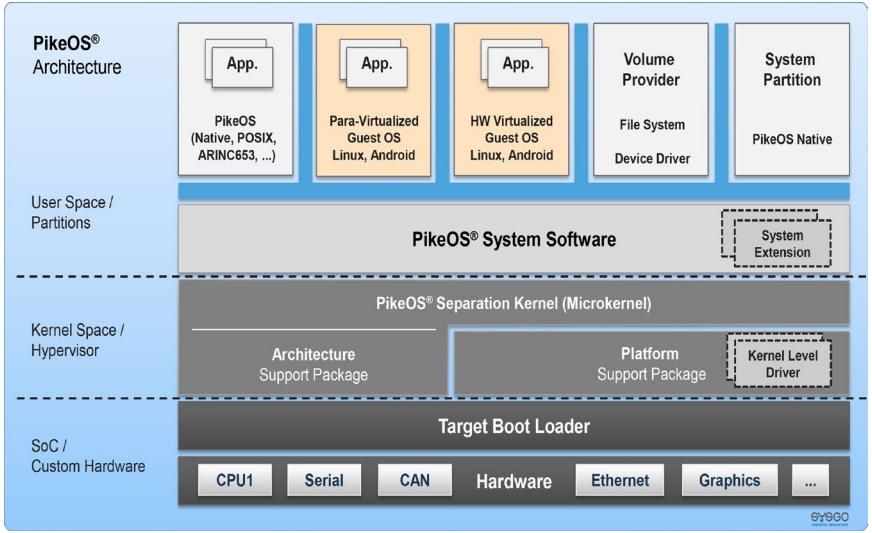
/// PikeOS is a real-time operating system that includes the hypervisor functionalities.

/// PikeOS native API (thread, semaphore,…) to implement real-time applications within partitions.

/// PikeOS partitions can host guest operating systems like Linux, POSIX, ARINC 653.

/// PikeOS and GR740.

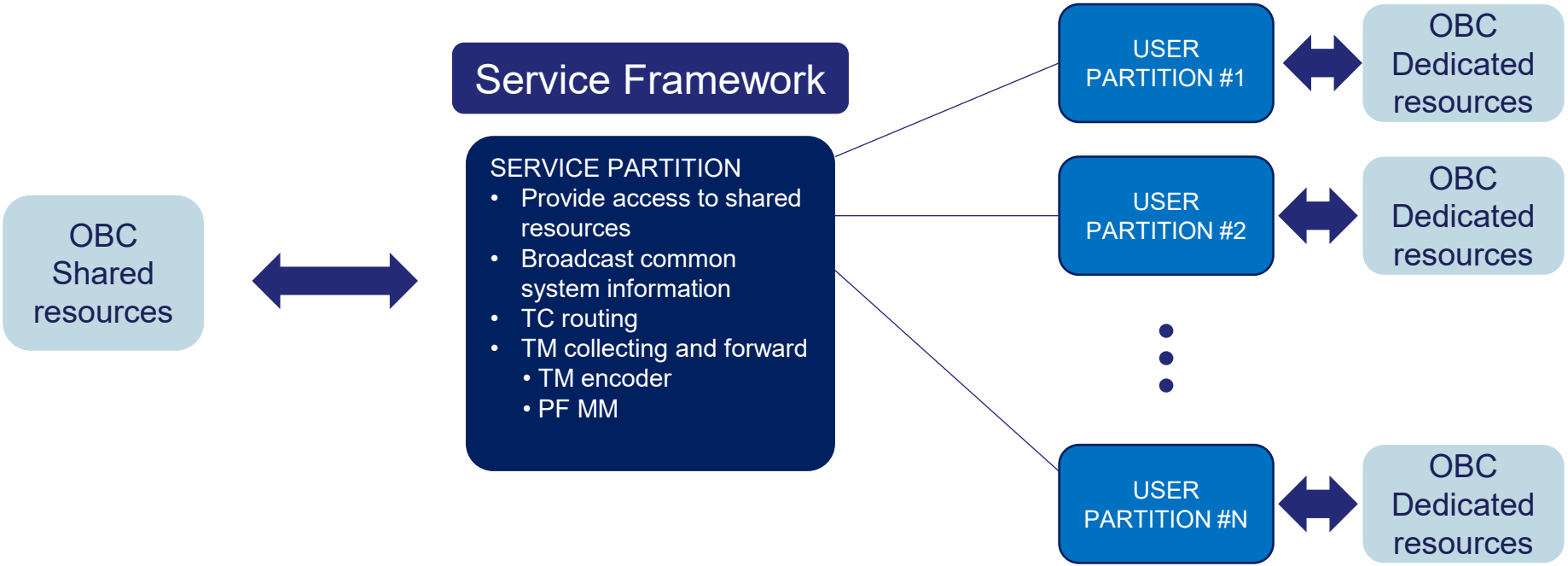
- Version 5.1.3 fully compatible with GR740
- ECSS Level A qualification process is on-going.



(reprinted from www.sysgo.com)

Date: 15/11/2023
Ref: 17th ESA ADCSS 2023
Template: 83230347-DOC-TAS-EN-011

SERVICE FRAMEWORK CONCEPT



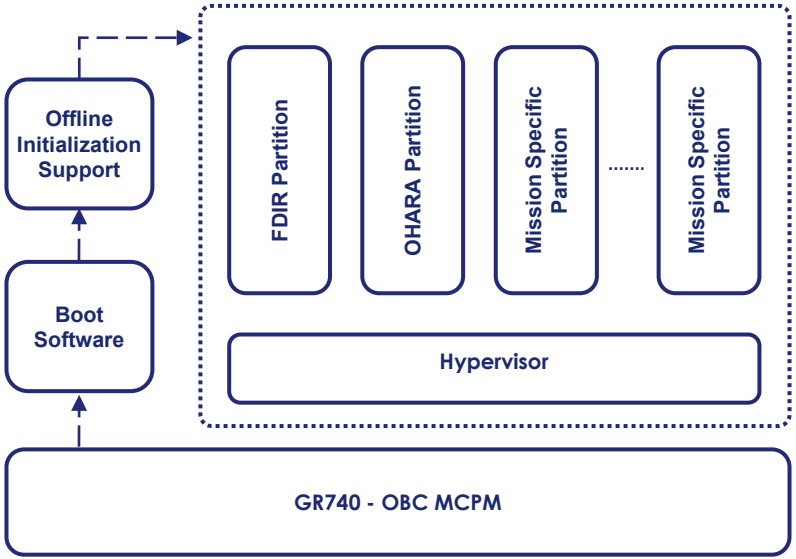
Date: 15/11/2023
Ref: 17th ESA ADCSS 2023
Template: 83230347-DOC-TAS-EN-011

MULTICORE SOFTWARE ARCHITECTURE (1/2)

/// The MCSW is based on MultiCore framework concept.

/// MultiCore Framework components:

- Boot Software
  - First application executed by the OBC after it is switched on or reset.
- Offline Initialization Support
  - It is in charge of performing further initializations and loading the hypervisor image jointly with all the other application
- Hypervisor
  - PikeOS (Native API)
- Multicore Framework Library
  - Static library that provides the abstraction of hypervisor layer.

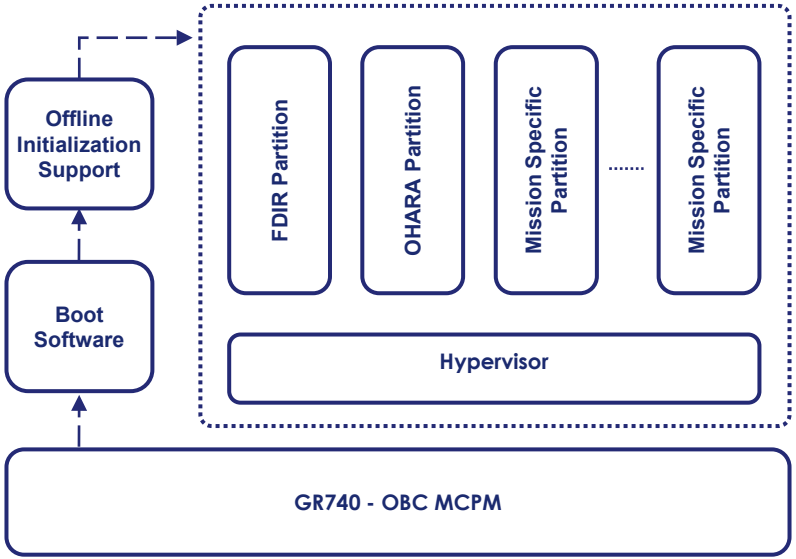


Date: 15/11/2023
Ref: 17th ESA ADCSS 2023
Template: 83230347-DOC-TAS-EN-011

MULTICORE SOFTWARE ARCHITECTURE (2/2)

/// MultiCore Framework components:

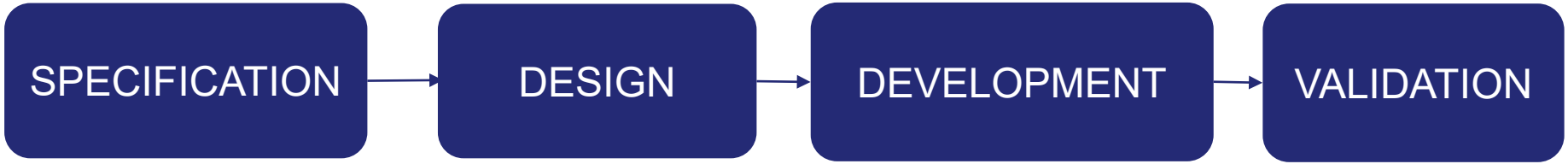
- Hardware-Dependent Software Library
  - Static library that provides software interfaces for direct management of both MCPM and T2MR devices.
- OHARA Application
  - Centralized server allowing the access shared devices to other from applications
  - T2MR resources management
  - TC packets acquisition and dispatching
  - TM packets downlink and storage
  - Minimal set of PUS services
- FDIR Application
  - It gathers the status of all other partitions and, if necessary, triggers the proper recovery action.



Date: 15/11/2023
Ref: 17th ESA ADCSS 2023
Template: 83230347-DOC-TAS-EN-011

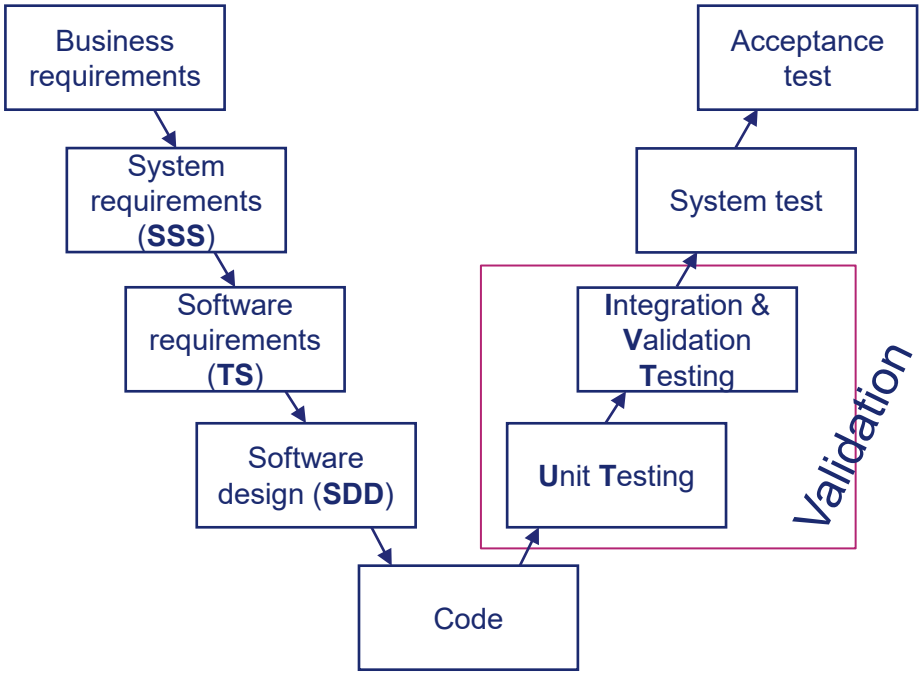
MULTICORE SOFTWARE LIFECYCLE: SUMMARY

- The MCSW lifecycle is based on the waterfall methodology.
- The MCSW input is the **Software System Specification (SSS).**
- From SSS, specific requirements are written to describe the MCSW behaviour. The output of this phase is the **Software Technical Specification (TS).**
- The design phase implements the software logical model from the TS. The output of this phase is the **Software Design Document (SDD).**
- The development phase follows the design phase: the output is the **MCSW executable code.**
- The validation phase defines the tests logic (**Software Validation Test Specification (SVTS)**) and produces their results (**Software Validation Test Result (SVTR)**).
- Typically multiple iterations of this model are applied.



Date: 15/11/2023
Ref: 17th ESA ADCSS 2023
Template: 83230347-DOC-TAS-EN-011

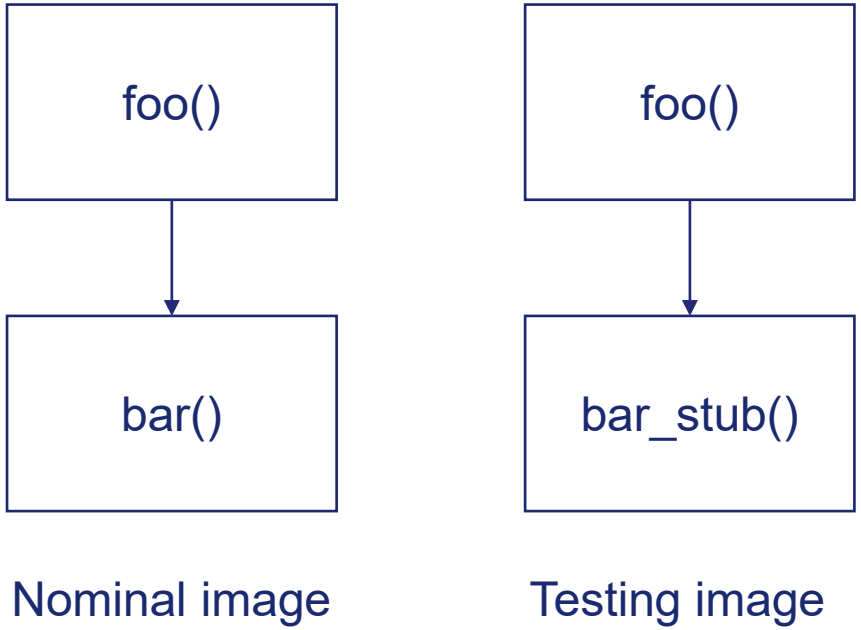
MULTICORE SOFTWARE LIFECYCLE: V-MODEL



Date: 15/11/2023
Ref: 17th ESA ADCSS 2023
Template: 83230347-DOC-TAS-EN-011

MULTICORE SOFTWARE: UNIT TEST

- **Fully automatic Unit Testing** performed by testing functions in isolation, with stubbed functions underneath.
- These tests are usually performed together with coverage testing, in order to gain full control of each execution path.
- Very useful to highlight bugs early in the validation process.
- Run by separate test driver.
- **VectorCAST** tool with **TSIM LEON4** as processor emulator is used.



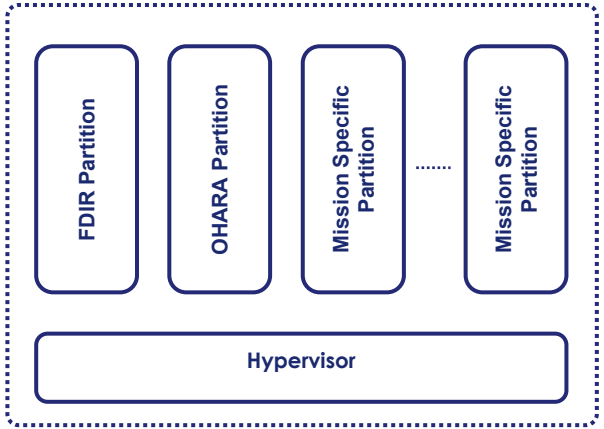
Date: 15/11/2023
Ref: 17th ESA ADCSS 2023
Template: 83230347-DOC-TAS-EN-011

MULTICORE SOFTWARE: INTEGRATION TEST

A TAS MultiCore Software image spans over several partitions, each of which is composed by several layers of functionality.

/// Several challenges:

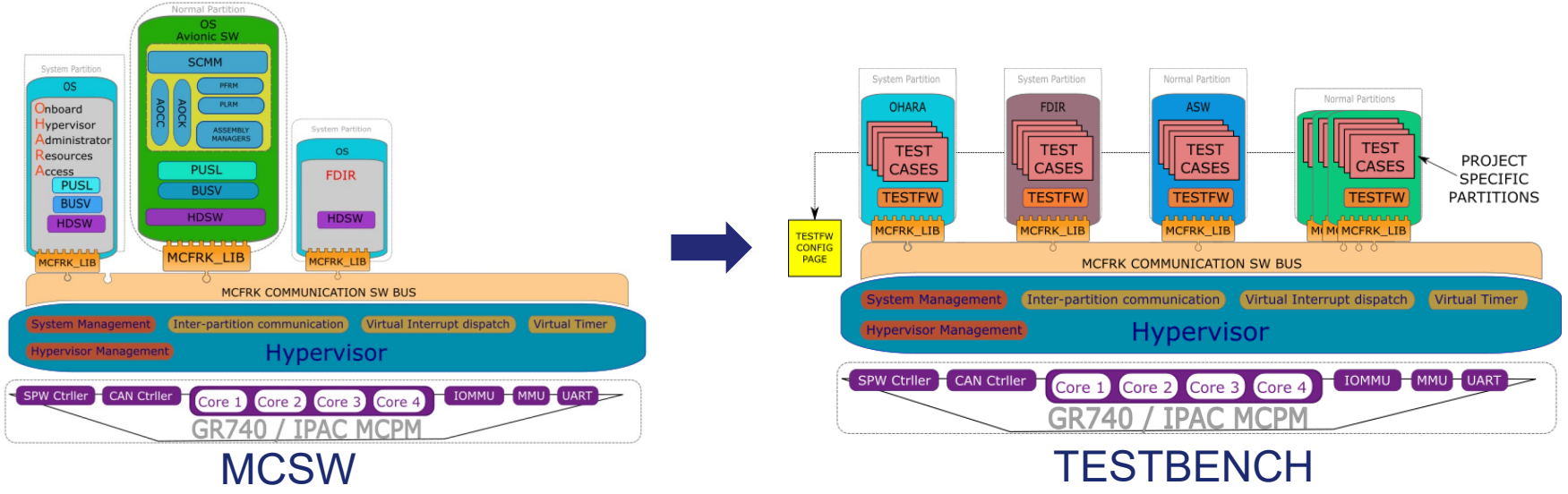
- Integration testing of each of the layers requires the underlying layers to be present as-is (no stubs shall be present) → vertical integration
- Each component under test in a partition may require other partitions to be present either as-is or with some instrumentation as well → horizontal integration
- Several degrees of external instrumentation may be present depending on the requirement to be verified
- Unification of the testing methodology over requirements, while at the same time avoid producing one image for each of the requirements to be verified (effort)
- Reproduction of the same test with different inputs



Date: 15/11/2023
Ref: 17th ESA ADCSS 2023
Template: 83230347-DOC-TAS-EN-011

MULTICORE SOFTWARE: INTEGRATION TEST & TESTBENCH

- **Fully automatic IT testbench: it allows testing the Multicore Framework and possibly the high level apps.**
- A custom test image is produced integrating the unmodified unit/system under test
- For each test to run there is up to one function registered on each partition implementing the test (named “test case”)
- A test case loader is integrated in each partition, selecting the test case to run
- Shared memory used for configuration (externally modifiable)

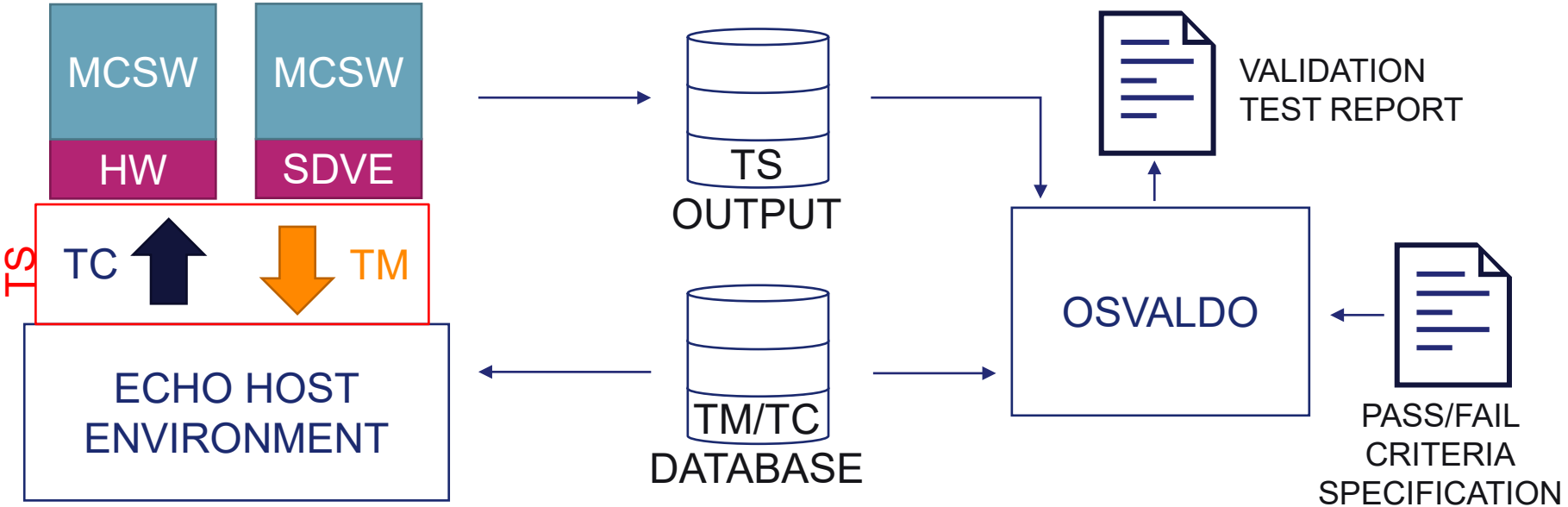


Date: 15/11/2023
Ref: 17th ESA ADCSS 2023
Template: 83230347-DOC-TAS-EN-011

MULTICORE SOFTWARE: VALIDATION TEST

/// Black-box **semi-automatic** validation performed by means of the **ECHO/OSVAldo** environment

- Unmodified MCSW image, comprised of boot software, hypervisor, multicore framework and applications.
- **TESLA** test sequences used to send telecommands, receive and parse telemetries and perform runtime checks.
- OSVAldo retrieves test sequence data, post-processes it and validates it by producing a validation report.



Date: 15/11/2023
Ref: 17th ESA ADCSS 2023
Template: 83230347-DOC-TAS-EN-011

TEST ENVIRONMENTS AND TOOLS SUMMARY

/// Various environments are used to perform the MCSW validation:

- GR740 emulator – integrated in the **SDVE** and used by **VectorCAST** UT
- **SDVE** with **ECHO/OSVAldo** – for TS validation
- **EBB** with **ECHO/OSVAldo** and real/simulated payloads – for HW dependent test and performance test

/// The IT testbench can be configured to account for differences on any of the environments above:

- Custom stubs for each environment e.g. to avoid invalid access to hardware when not present
- TCL scripts are used to switch within various tests from the same testbench and configure the test with data
- Custom test runner runs several tests automatically from a single test plan file
- CI pipelines are setup to automatically run tests on emulator and check for regressions at every pushed commit (git)

MULTICORE SOFTWARE: NEW VALIDATION ASPECTS (1/2)

/// Building Blocks approach

- TASI started a new implementation policy based on the concept of BBs: the common layers of MCSW are decoupled from the Mission-specific logic and isolated as self-consistent products.
- Each BB is divided in source and configuration logic.
- Each BB is self-validated with a proper validation test campaign using a set of test data.

/// Building Block customization, integration and regression

- When a BB has to be used on a Mission, it is instantiated and configured with the specific Mission data.
- A set of regression test are run to verify the correct BB integration.

/// User Partitions integration

- TASI framework based on BB, separately validated, provides a robust common layer to add Mission User Partitions
- Every new User Partitions is individually developed and then added to the MCSW for its integration
- After each new User Partition integration, we conduct overall MCSW analysis and robustness tests (see next slide)

Date: 15/11/2023
Ref: 17th ESA ADCSS 2023
Template: 83230347-DOC-TAS-EN-011

MULTICORE SOFTWARE: NEW VALIDATION ASPECTS (2/2)

/// Task/thread in a single Partition schedulabilty

- Static analysis of the maximum task/thread execution time taking in account the core interference and the cache behaviour (L2 cache is shared among Partitions)

/// Cores schedulabilty

- Static analysis of the maximum core execution time taking in account the single Partitions timing

/// Core interference: TASI measures (and provides in TM) in each validation test

- Task/thread execution time per Partition
- Overall execution time per Core
- Incremental analysis of core interference jointly to Partitions integration

/// MCSW FDIR

- FDIR at Partition level
- FDIR at MCSW level: managed by FDIR Partition concept

Date: 15/11/2023
Ref: 17th ESA ADCSS 2023
Template: 83230347-DOC-TAS-EN-011

THANK YOU
FOR YOUR KIND ATTENTION

/// Contacts for additional information

- domenico.teodonio@thalesaleniaspace.com
- andrea.bufalino@thalesaleniaspace.com
- tomas.dicocco@thalesaleniaspace.com
- sante.candia@thalesaleniaspace.com
- fausto.dantonio@thalesaleniaspace.com