

Functional Verification Environment: Current Status and Breakthrough

H. Sondermann - Astrium GmbH / A. Darbin, B. Dillenschneider, Y. Laprade - Astrium SAS

All the space you need



© EADS Astrium 2008

Overview

- **Functional Verification Supported by Simulation**
- **Numerical Verification Environment**
SVF and Satellite Simulator
- **Classical Hardware in the Loop Test Benches**
Electrical Functional Model - EFM
- **Hybrid / Unit in the Loop Verification Environment**
EFM Platform Simulator

1

Satellite Functional Verification

Satellite Functional Verification

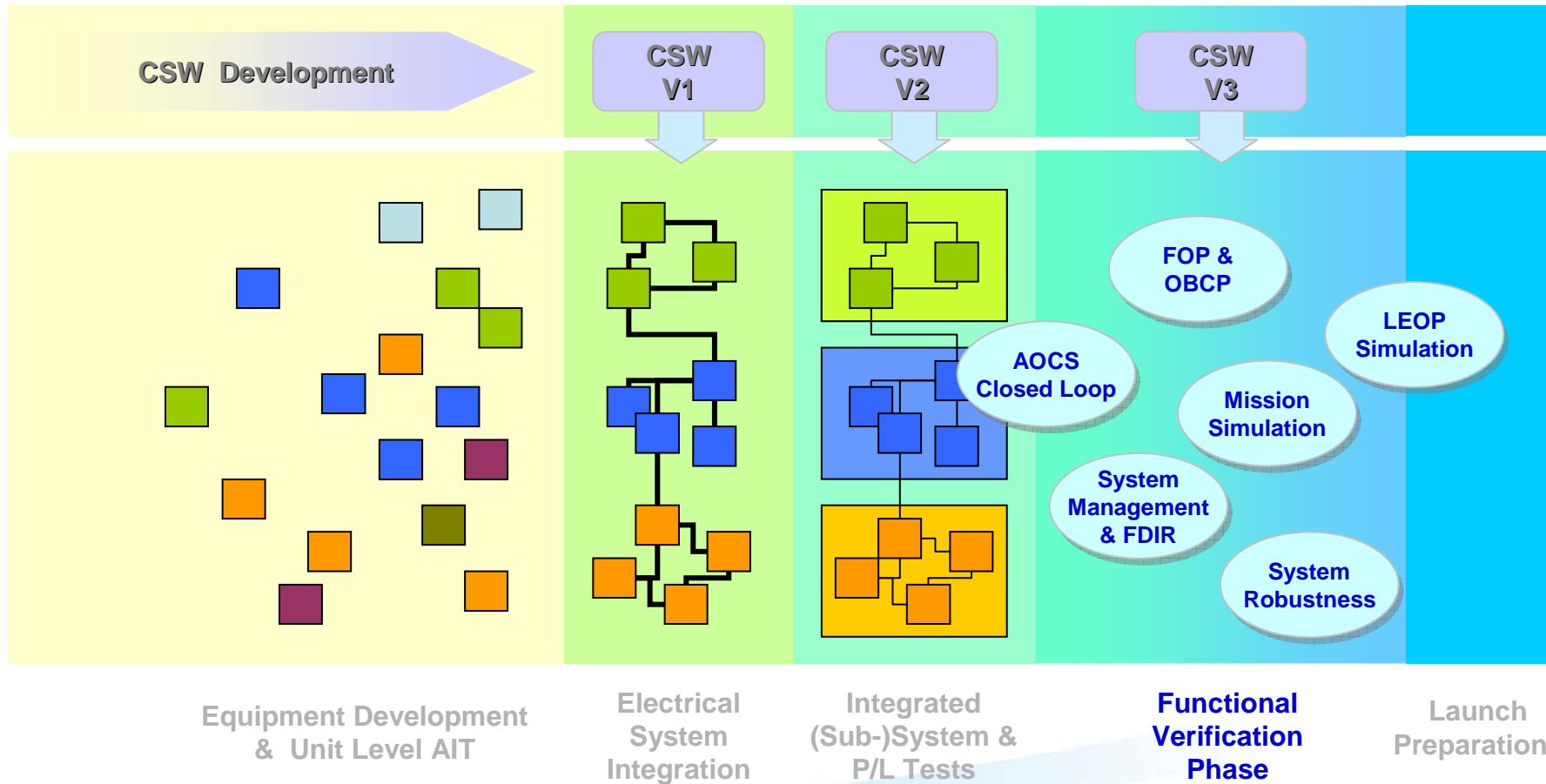
An engineering activity to verify/validate a system from a functional view

- Functional Verification covers both tasks of verification and validation, i.e.
 - verifying that the system design conforms to its specification, i.e. 'confirmation through the provision of objective evidence that specified requirements have been fulfilled', ECSS-E-10-02B.
 - validating that the specified design does what it is intended to do.
- Perimeter of Functional Verification
 - the verification of all functional requirements implemented by S/W on the target H/W environment
 - the verification of requirements related mission and operation scenarios
 - the validation of the design solution for this scope of responsibilities
- Functional Verification is embedded in the overall Avionics and System Level Verification Process

Functional Verification Domains

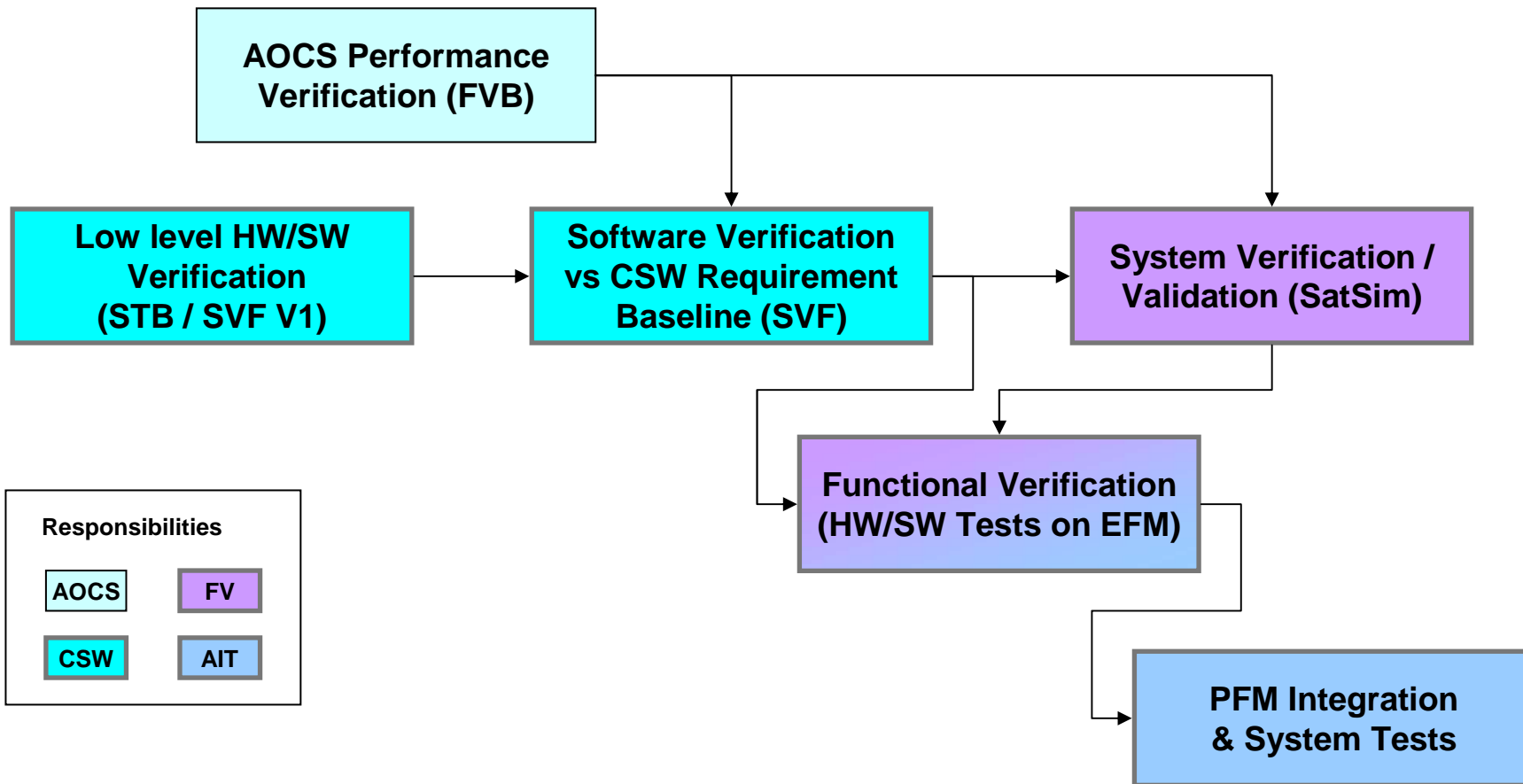
- On-board Software (CSW) system level acceptance & validation
- Verification of unit and sub-system functions controlled by CSW
- Verification of system level FDIR, based on lower level FDIR elements
- Verification & validation of system robustness (robustness and tolerance to unexpected problems)
- Verification & validation of Flight Operation Procedures (FOP) and On-board Control Procedures (OBCP)
- Mission Simulation involving AOCS Closed Loop Operations
- Support to SVTs.

Functional Verification Domains in Satellite AIT



This document is the property of Astrium. It shall not be communicated to third parties without prior written agreement. Its content shall not be disclosed.

Astrium Generic Functional Verification Reference Process

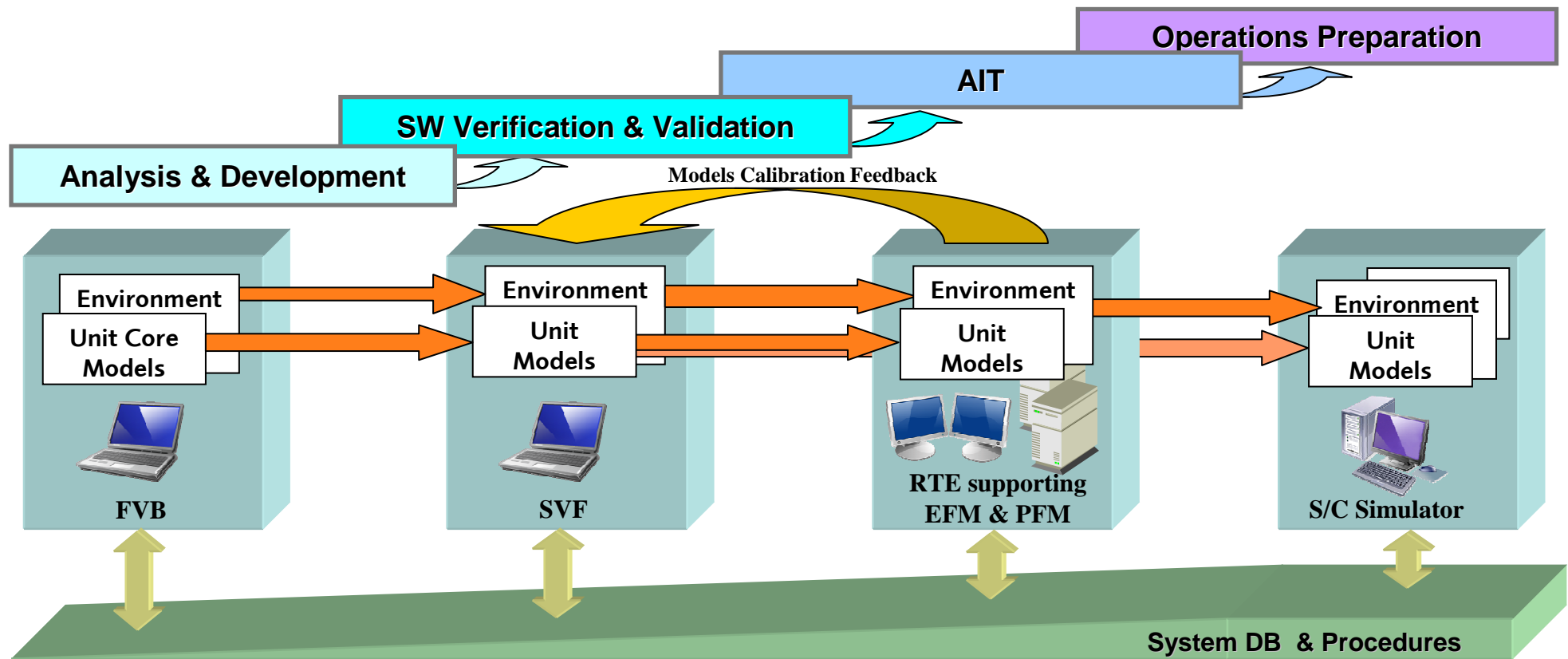


Responsibilities

AOCS	FV
CSW	AIT

This document is the property of Astrium. It shall not be communicated to third parties without prior written agreement. Its content shall not be disclosed.

Simulation Infrastructure in the Satellite Verification Flow



This document is the property of Astrium. It shall not be communicated to third parties without prior written agreement. Its content shall not be disclosed.

2

Numerical Verification Environment

Numerical Verification Environment

■ FVB - (AOCS) Functional Validation Bench

Usage:

- To support the development, verify and validate AOCS algorithms and performance.
- allowing closed loop simulations with either an image of the AOCS flight S/W application or modules of the AOCS flight S/W in the loop.

Implementation:

- Full simulation of mission environment and flight dynamics
- Need for speed -> optimised for run-time performance
- No emulation of on-board processor(s)
- Only unit core functions are simulated
- No TM/TC ground interface
- Single PC / Workstation

Numerical Verification Environment

■ SVF – Software Verification Facility

Usage:

- To support the development, verification and validation of the on-board SW.
- allowing to verify essential parts of the SW requirements (SW-SW integration tests & global tests) in an open and/or closed loop set-up,
- with debugging and failure injection capabilities

Implementation:

- Full simulation of mission environment and flight dynamics
- Full emulation of the on-board processor, 'simulated real-time'
- Unit simulation with core model and functional model interface layer.
- Full TM/TC ground interface
- Operated through a lightweight operational I/F (SIMOPS)
- Single PC / Workstation

Numerical Verification Environment

■ Satellite Simulator = Augmented SVF

Usage:

- the prototyping and verification of test / operations procedures including system level operations verification as a preparation of the EFM / PFM AIT phase.
- the preparation and debugging of AOCS open and closed loop test cases as inherited from the FVB (reference cases) and augmented by the final operational layer.
- the composition and debugging of complex mission scenarios, the generation and verification of associated flight operations procedures.
- the debugging and verification of on-board control procedures.
- the debugging and pre-validation of SVT and launch preparation procedures.
- the verification of FDIR scenarios.
- the verification of the satellite database and the CCS operational interface
- the regression testing of incremental CSW deliveries and bug fixes at system level
- the troubleshooting for problems experienced on EFM or PFM thus avoiding a blocking of the H/W benches with lengthy failure investigation activities.

Implementation:

- Derived from SVF....
- Operated through the Central Check-Out System (CCS) or Ground Operation System
- Can be upgraded with ground TTC network and visibility modelling.

3

Hardware in the Loop Test Benches

Hardware in the Loop Test Benches

■ Software Test Bench – STB

-- The first Hardware-in-the-Loop test bench of a project --

Usage:

- To support the development, verification and validation of the low level part of the CSW.
- To calibrate the OBC numerical simulator of SVF V1 against the real OBC (BB) unit.

Implementation:

- Build around a breadboard of the on-board computer (OBC)
- Interface boards and stub models available for all OBC / processor board interfaces, e.g. Mil-1553B, SpaceWire, UART, HPC, etc.
- Operation through TMTC FrontEnd and OBC Service Interface (SIF)
- IRIG based timing and synchronisation of stimuli and monitor stubs
- Toolbox for OBC characterisation
- Based on the same (VME) infrastructure as the later real-time test environment

Hardware in the Loop Test Benches

- **Electrical Functional Model - EFM** supported by the **Real-Time Test Environment – RTE (aka Avionics SCOE)**

Usage:

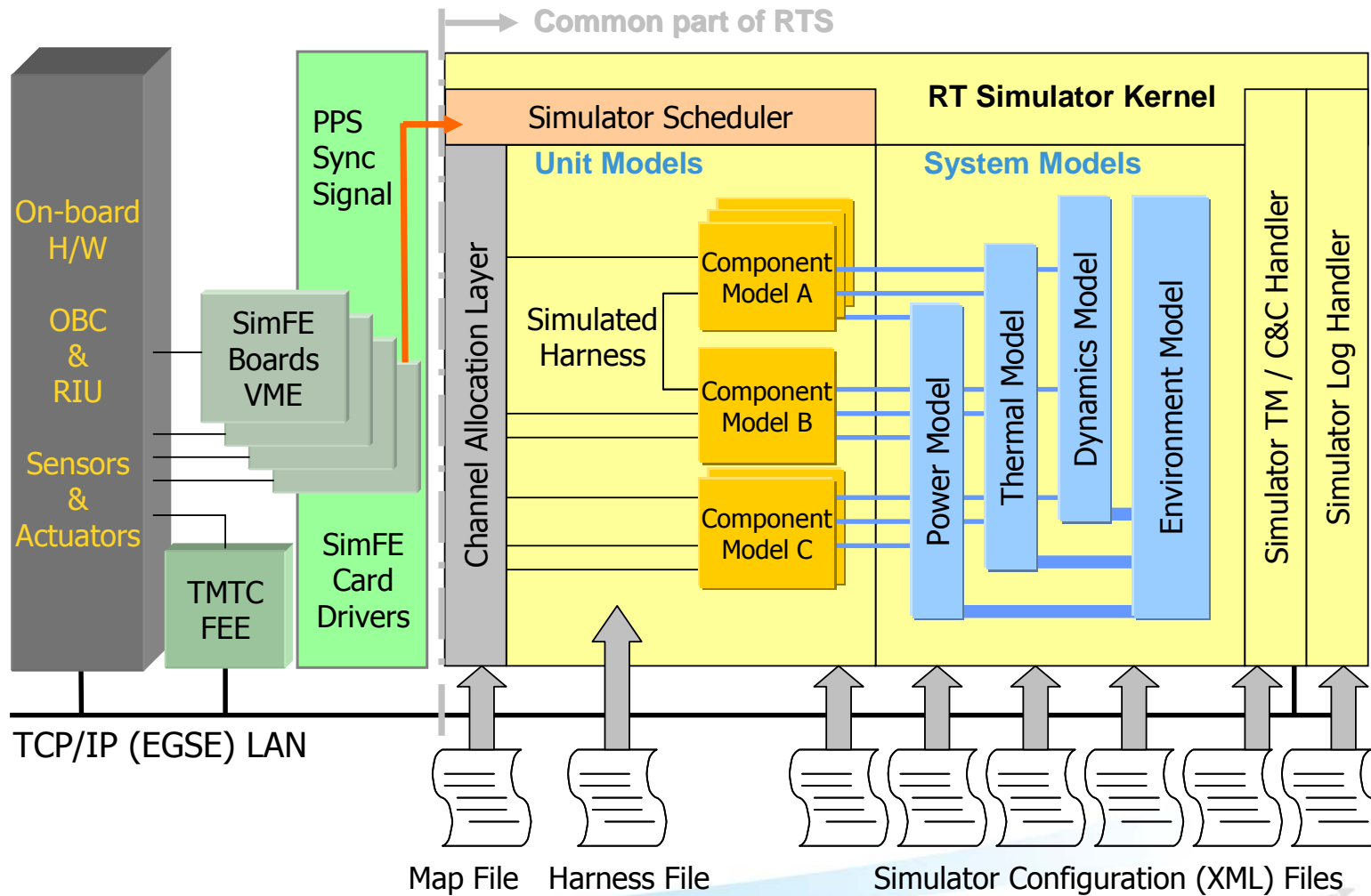
- Most of the use cases already listed under the Satellite Simulator topic plus those tests which cannot be performed on the SVF / SatSim

Implementation:

- Minimum hardware in the loop: OBC and I/O unit (RIU)
- Re-uses real-time simulator (RTS) of SVF / SatSim
- Unit simulation as core model with functional and electrical interface layer.
- Unit stimulation models interfacing with SCOEs or directly with on-board H/W
- Hard real-time requirement, synchronised to the on-board time reference
- Running under Vxworks or Real Time Linux

The RTE offers maximum flexibility to augment the S/C hardware configuration under test to a complete system as necessary for the intended verification task!

Real-Time Test Environment (RTE) for EFM/PFM



This document is the property of Astrium. It shall not be communicated to third parties without prior written agreement. Its content shall not be disclosed.

4

EFM Platform Simulator - SimEFM

EFM Platform Simulator - SimEFM

■ Unit in the Loop Verification Environment

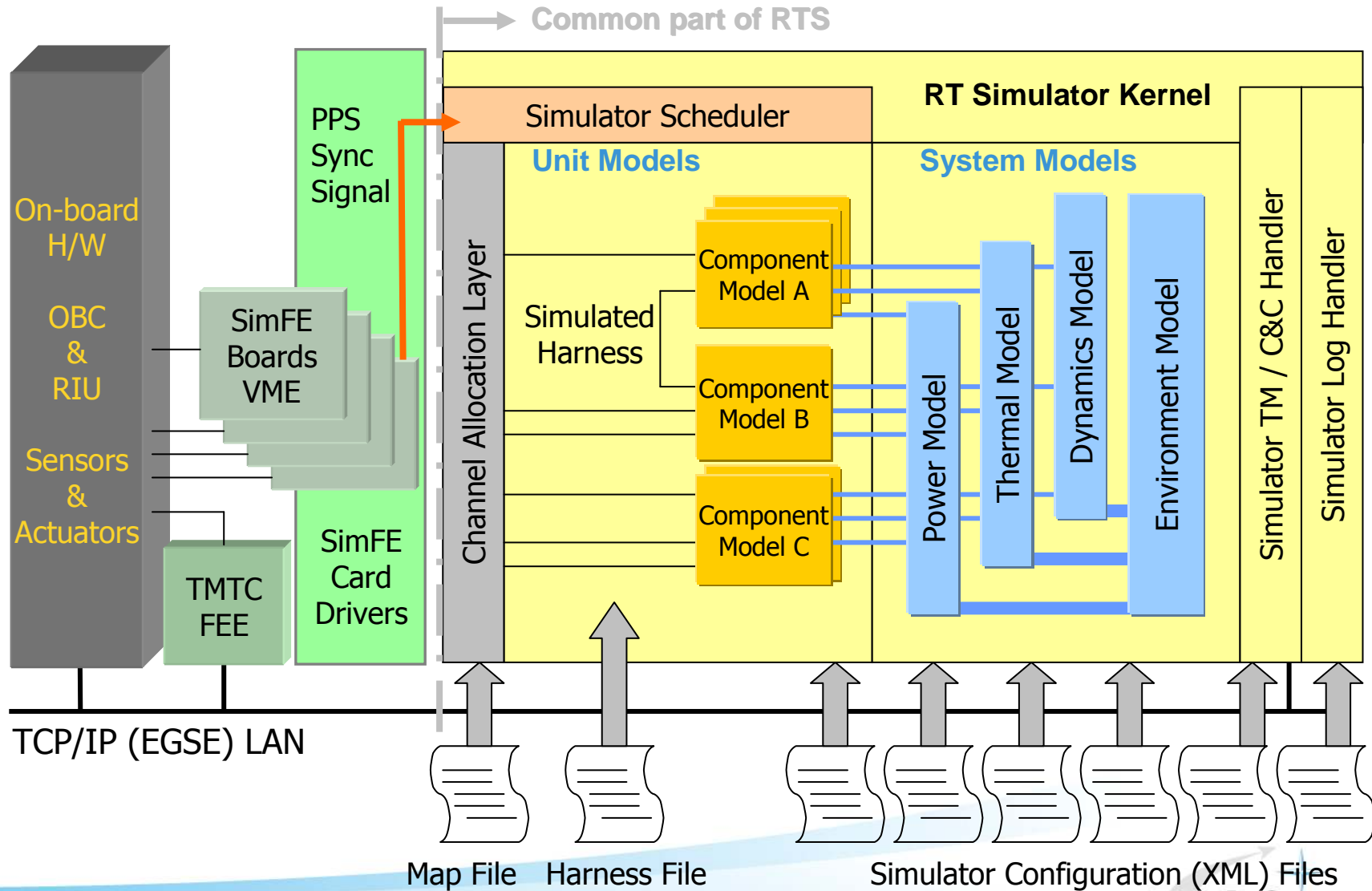
Usage:

- To have an alternative test bed to prepare the AIT phase on EFM and PFM, i.e. a numerical platform simulator which can be used for
 - SCOE / EGSE pre-integration and validation
 - sensors & actuators characterisation
 - payload operational verification prior to its mating with the platform.

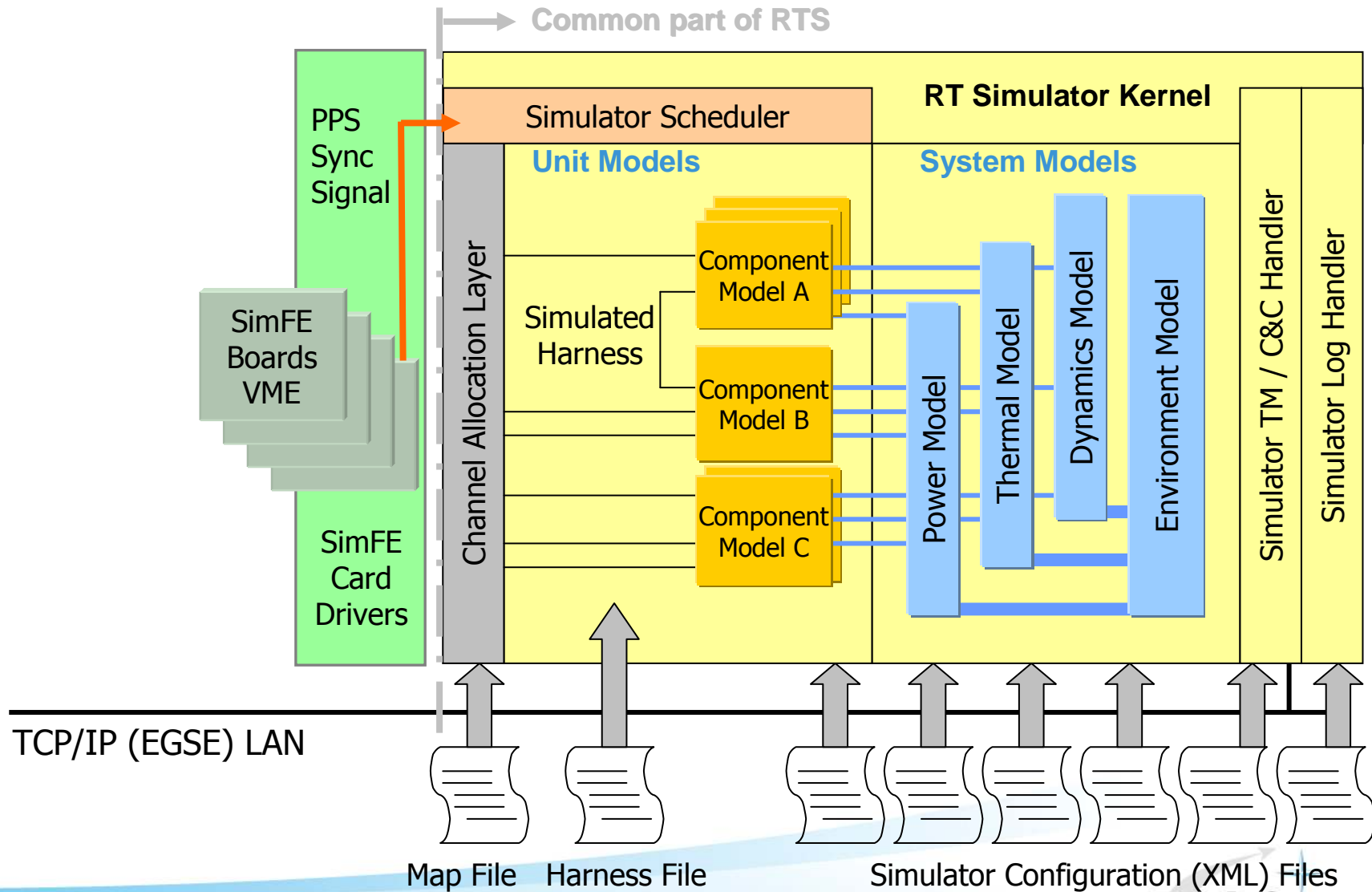
Implementation:

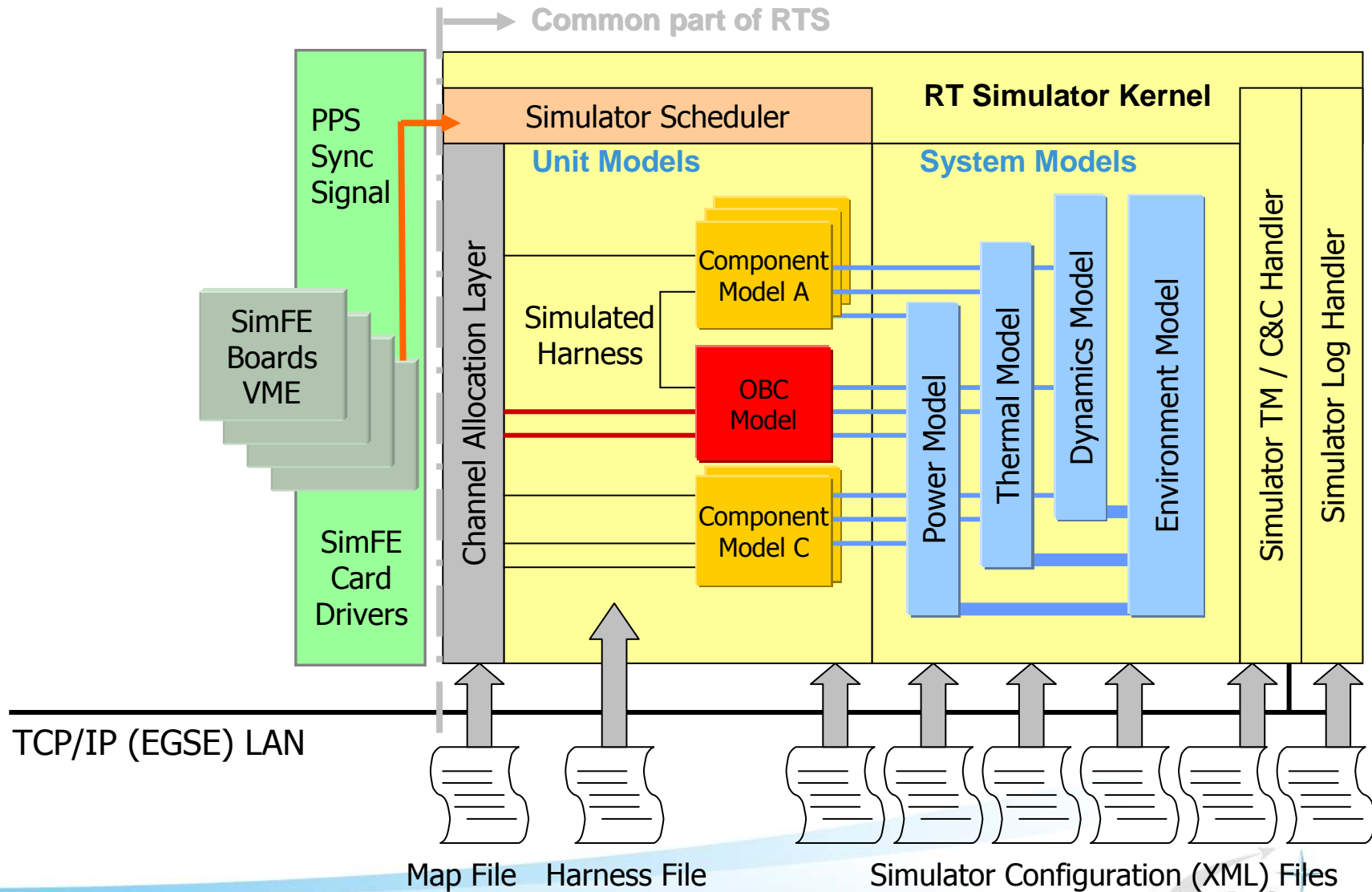
- Include a high fidelity model of the OBC in the real-time simulator of the EFM
- Find a suitable timing and synchronisation module
- Ensure maximum real-time performance of the OBC simulator.
- Attach an appropriate interface to the RTS workstation, e.g.: MIL 1553B, Spacewire or UART
- Run it under Real Time Linux

Real-Time Simulator in EFM Configuration

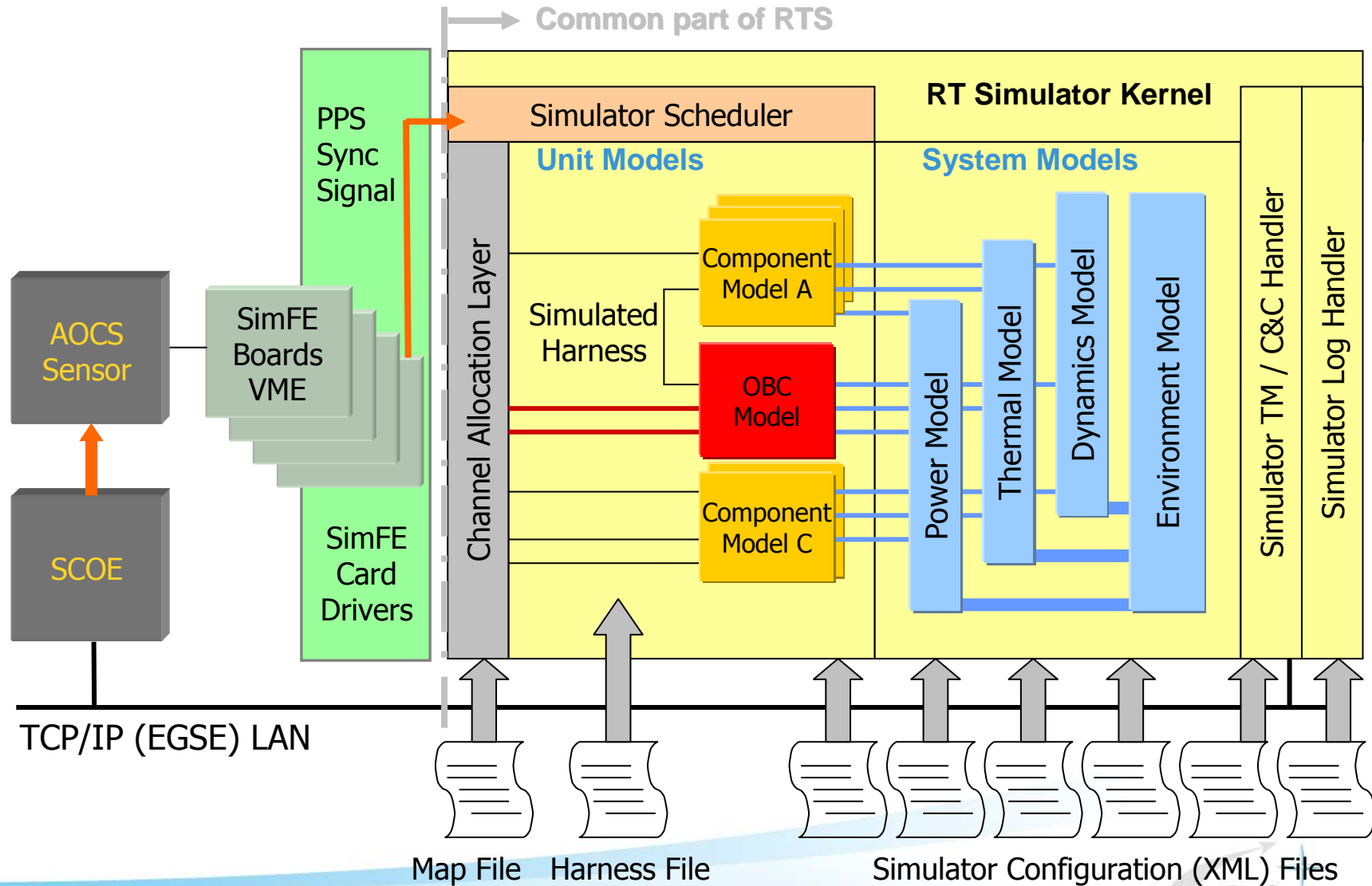


This document is the property of Astrium. It shall not be communicated to third parties without prior written agreement. Its content shall not be disclosed.





Real-Time Simulator in SimEFM Configuration

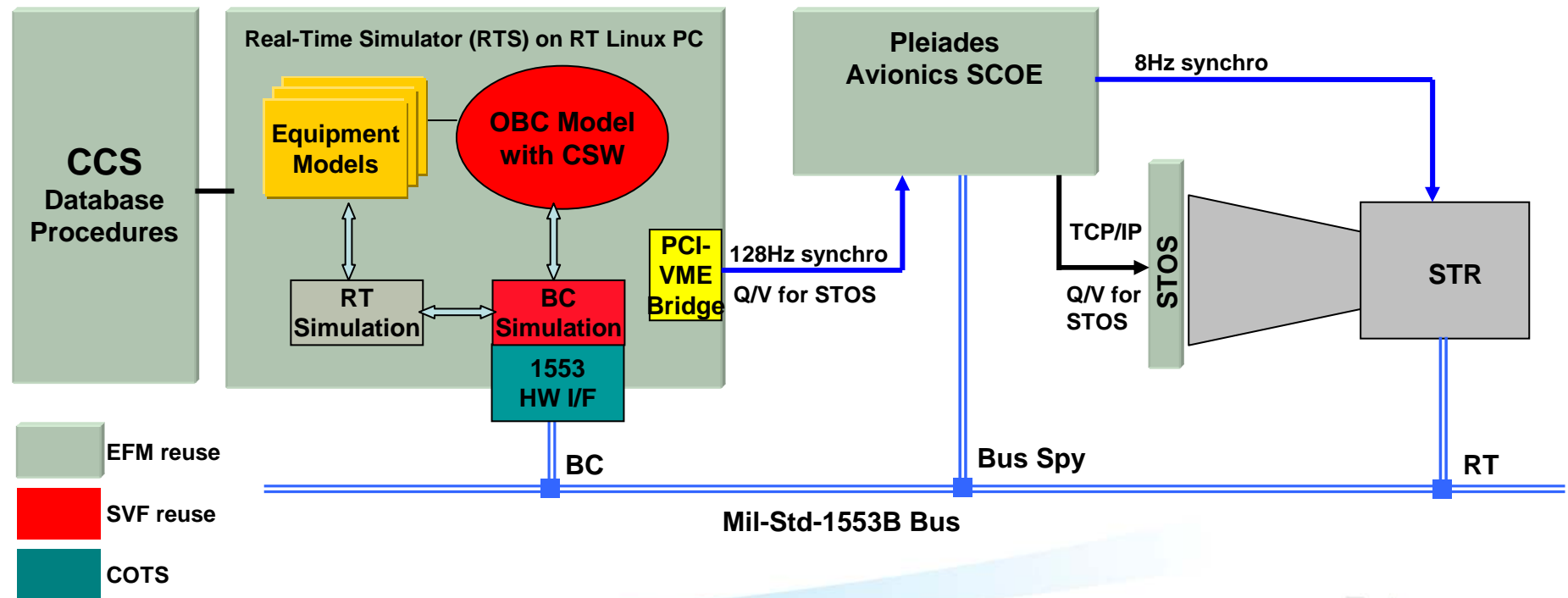


This document is the property of Astrium. It shall not be communicated to third parties without prior written agreement. Its content shall not be disclosed.

SimEFM Prototype Architecture

SimEFM is derived from EFM RTS with numerical SVF OBC model replacing the real OBC :

- Hard real-time constraint is applied to OBC model and CSW running in emulated ERC32
- OBC model becomes Bus Controller on real Mil-Std-1553B through a COTS I/F board
- Maximum communality with EFM & SVF platform (procedures, models, databases)



This document is the property of Astrium. It shall not be communicated to third parties without prior written agreement. Its content shall not be disclosed.

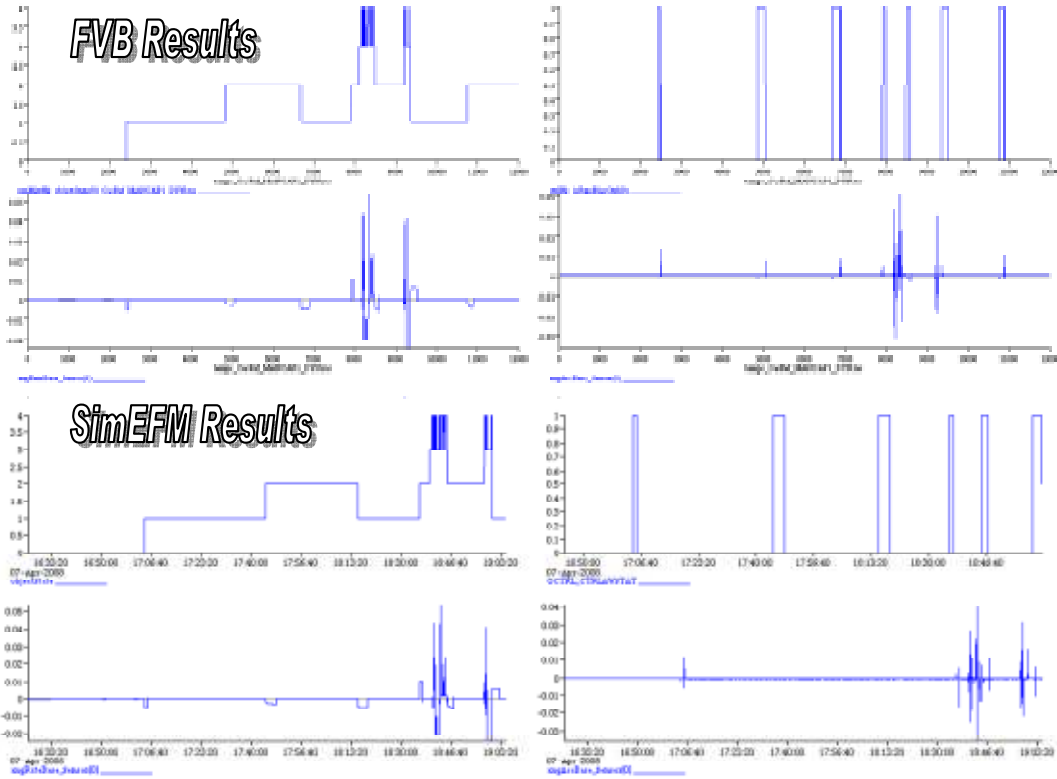
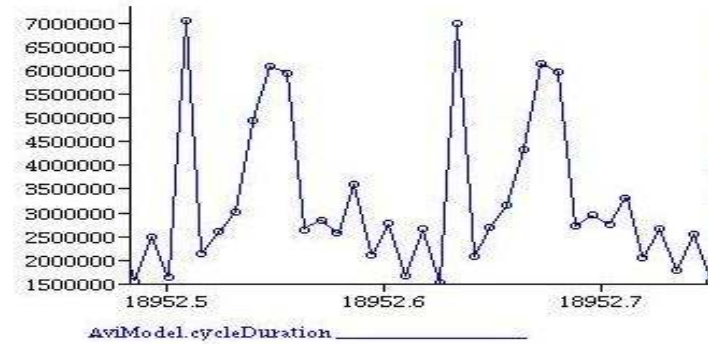
SimEFM Prototype

- Star Tracker campaign
 - Stimulated with STOS
 - Open-loop integration tests
 - Equipment Characterization
 - STOS alignment
 - Closed-Loop AOCS scenario
- Encouraging & positive results
 - Low-cost development
 - Performance achieved
 - 1553 timings
 - OBC Model coped with hard real-time
 - Fidelity achieved :
among 4 NCs detected on EFM, 3 of them could have been detected already on SimEFM



SimEFM - Achieved Performance

- Hard real-time with OBC model and ERC32 emulation
- Successful closed-loop normal mode/capture scenario with real STR
- Good 1553 timing fidelity



	Average value	Expected theory
DeltaT multiple 125ms_SA01	11829,88	12499 μ s
DeltaT SA01_SA28	1526,25	1526 μ s
DeltaT SA28_SA13	2574,29	2584 μ s
DeltaT SA13_SA14	2165,94	2166 μ s
DeltaT SA14_SA15	2501,66	2502 μ s
DeltaT SA15_SA16	2166,51	2166 μ s
DeltaT SA16_SA17	2464,27	2463 μ s
DeltaT SA17_SA18	2422,14	2422 μ s
DeltaT SA18_SA19	2166,20	2166 μ s

This document is the property of Astrium. It shall not be communicated to third parties without prior written agreement. Its content shall not be disclosed.

SimEFM - Improvements & Perspectives

- Improvement of OBC model coupling to 1553 BC thread
 - Current version programs one message after the other, introducing a 10 μ sec overhead per message
 - Improvement of message buffering in the OBC model to optimise handling chained messages

- SimEFM is based on ERC32 emulator
 - a performance improvement of 20% of emulator is required for LEON-based systems

- Configuration under assessment for Astrobus 250 production line
 - Recurring platform
 - Performance aspect with LEON emulator to be addressed
 - SimEFM Use Cases to be refined

Conclusions

- **A standardised Functional Verification Process harmonised across all Astrium sites and based on a common FV infrastructure**
- **Numerical and Hybrid Simulators are configured from a minimum number of recurring elements**
- **Continuous improvements in simulator fidelity and architectures**
⇒ **Presentation on SimTG**
- **Expanding the use of simulators to ‘non-standard’ applications**
- **Full coverage of Functional Verification Use Cases achieved**