Test Means at Airbus Defence and Space,
Military Aircraft business line

Making Ada the Heart of an All-Encompassing Aircraft Test Life-Cycle

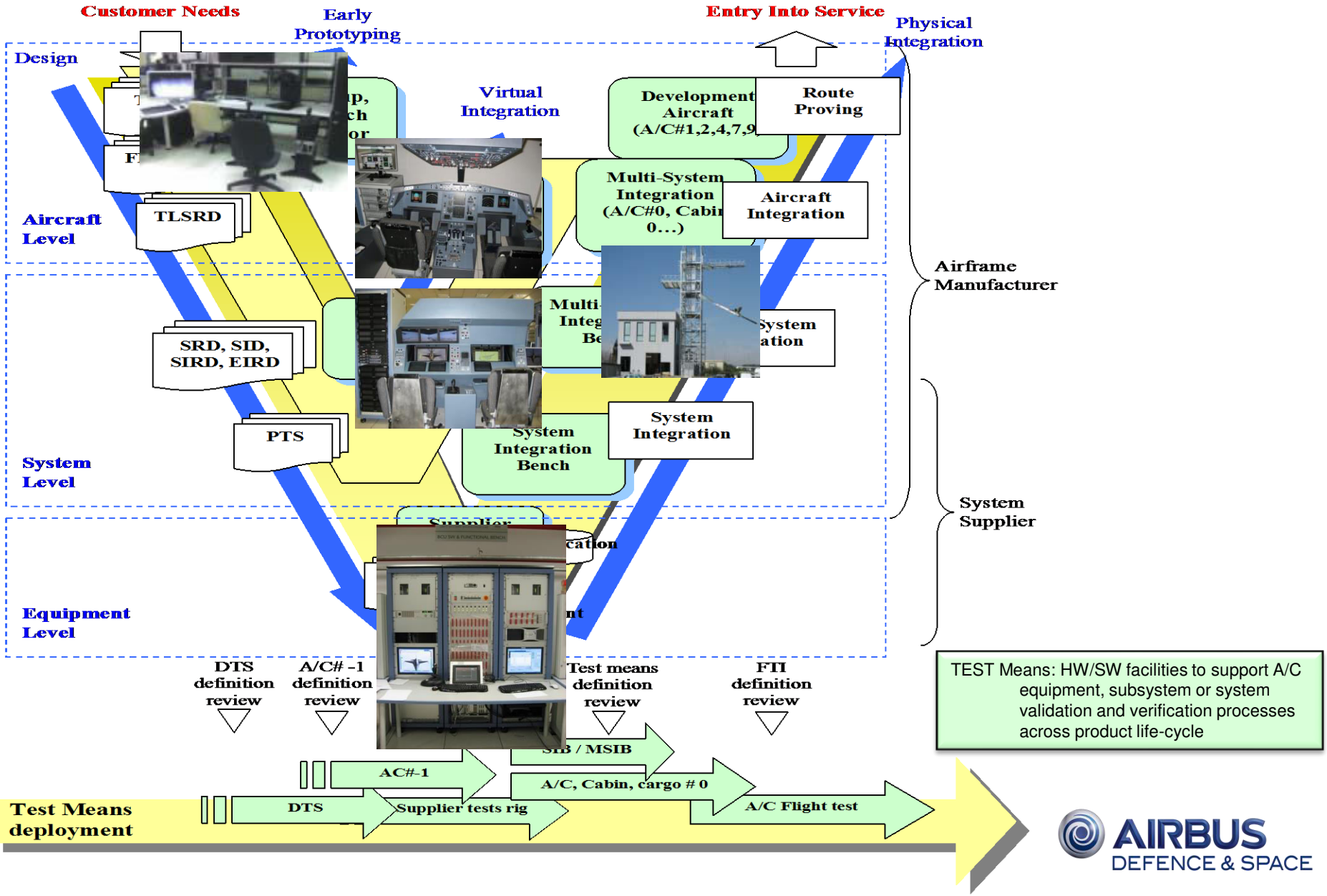
Industrial Presentation, Ada in Aerospace Session
Ada Europe 2014, Paris

Javier Arroyo
Test Systems

24 June 2014

AIRBUS
DEFENCE & SPACE

A/C Electronics Systems Development Life Cycle

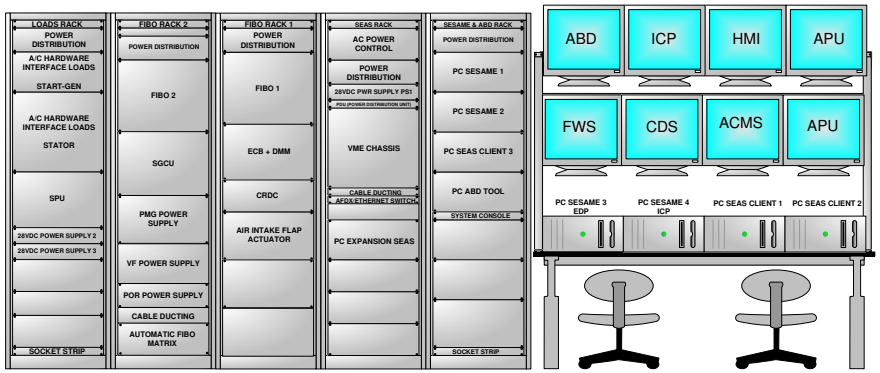


TEST Means: HW/SW facilities to support A/C equipment, subsystem or system validation and verification processes across product life-cycle

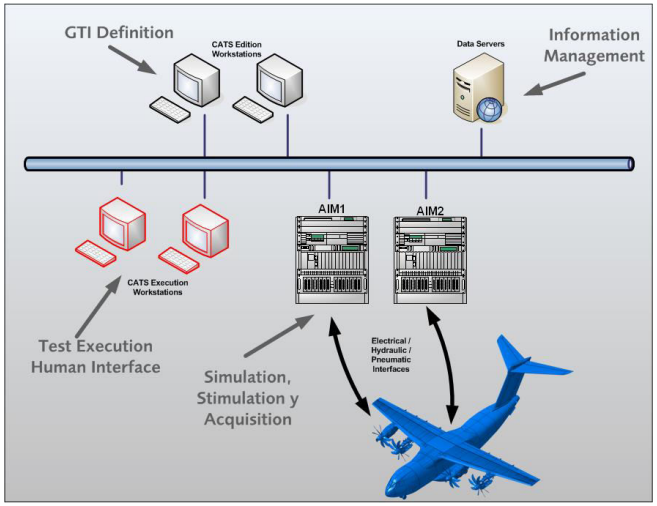
AIRBUS DEFENCE & SPACE

Test Means Architecture

SYSTEM INTEGRATION BENCHES



FAL TEST SYSTEM



Remarkable capabilities

- Common Aircraft Interface Core Module
- Manual and automatic aircraft interfaces
- Specific User Interface

AIRBUS DEFENCE & SPACE

Test Facilities

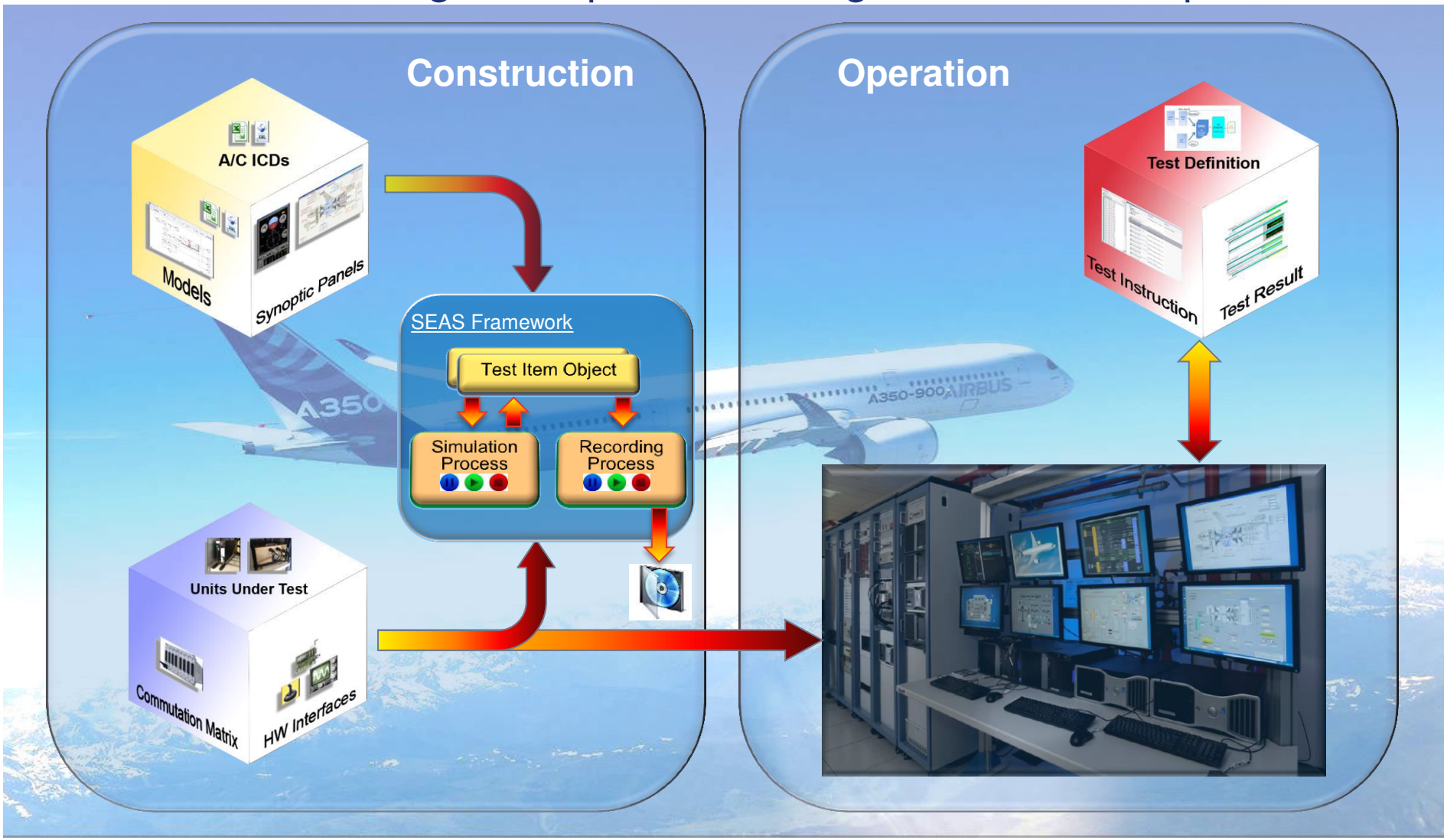
A common solution applied over and over again to satisfy different needs

•The maturity, robustness and reliability of the system have been demonstrated throughout hundreds of test facilities in use with this common test environment, including Engineering Simulators for Aircraft Refuelling Boom System, System Integration Benches for Multirole Tanker Aircrafts, A400M, Light & Medium Transport Aircraft, Full Integrated Tactical Systems and Aircraft Interface Modules for Final Assembly Lines of A400M, Multirole Tanker Aircrafts and L&MT aircrafts.

•Proven with up to 400.000 signals per system integration bench

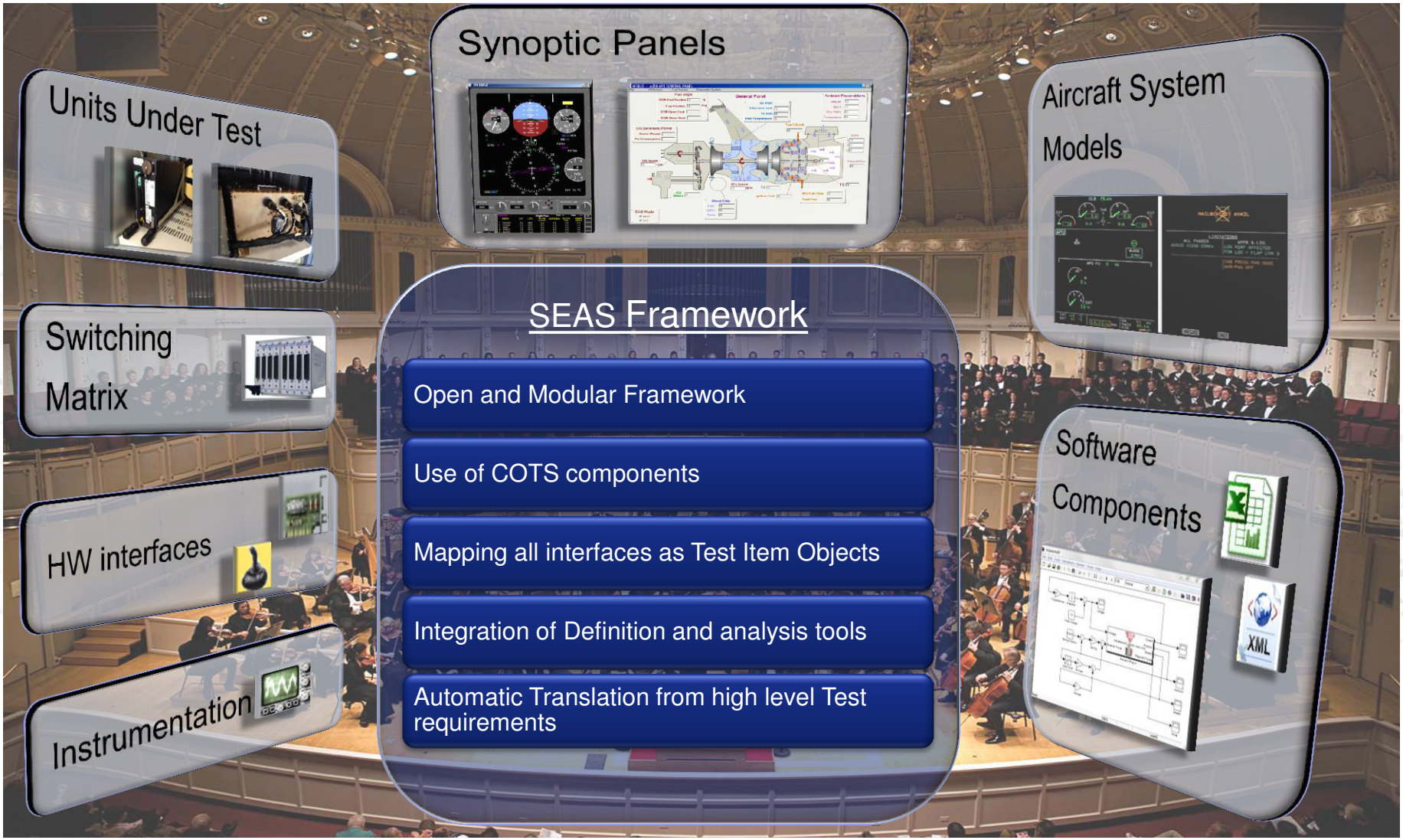
AIRBUS DEFENCE & SPACE

Test Means Building and Operation using HW/SW bricks portfolio



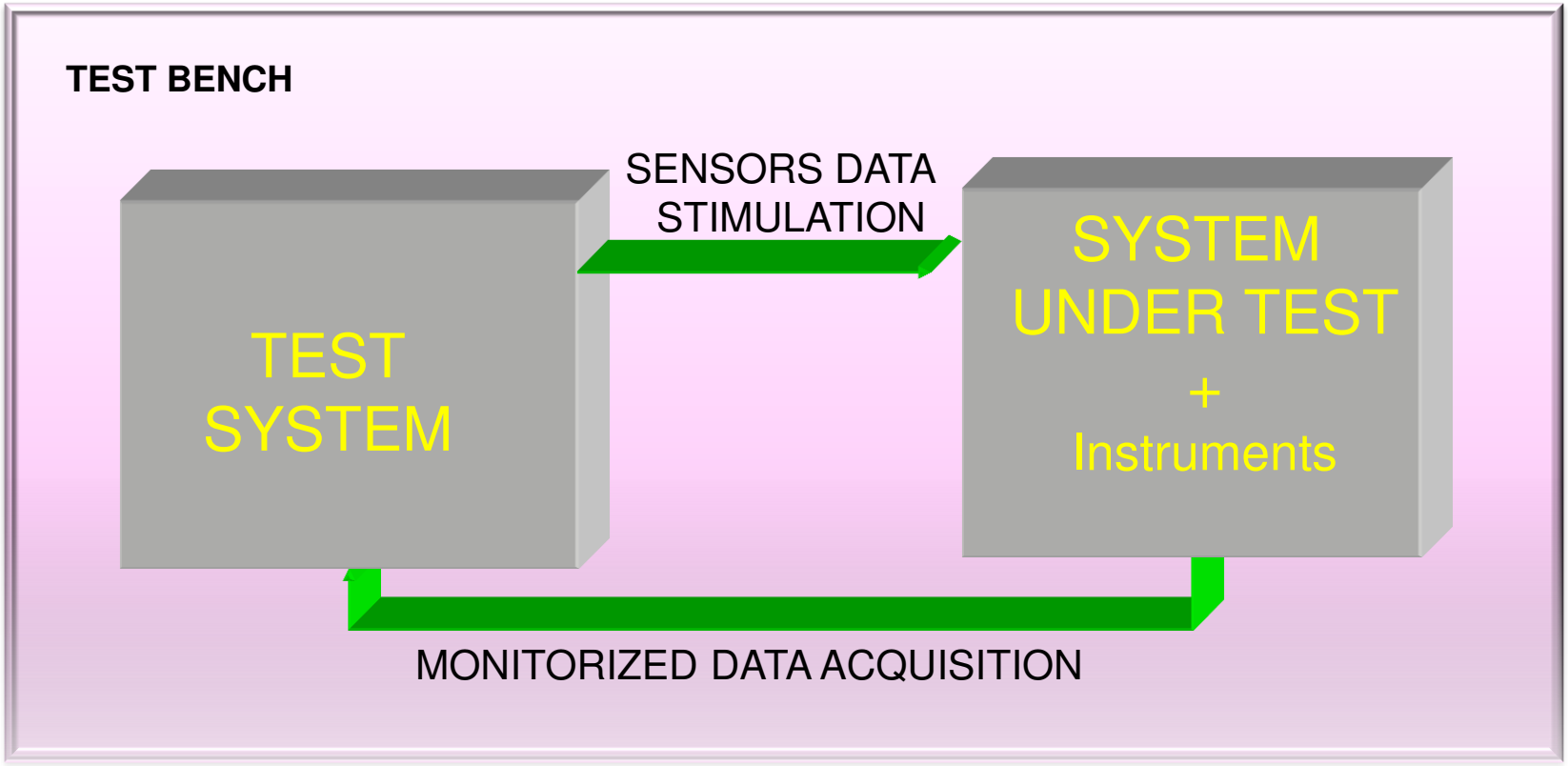
AIRBUS DEFENCE & SPACE

SEAS: The Orchestra Conductor



AIRBUS DEFENCE & SPACE

SEAS Overview



Stimulation, Acquisition and Simulation System (SEAS)

SEAS is a modular, generic, distributed set of HW/SW items used to build test facilities
(Engineering Simulator, SW Benches, Functional Test Benches and Target Rigs)

AIRBUS
DEFENCE & SPACE

SEAS Features

- Modular, open, distributed and scalable architecture widely used with minimal changes from desktop simulators to target rigs
- Key features: Configurability, Portability, Interoperability, Reusability, Scalability, Reliability and Maintainability
- **Generic SW components are written in Ada 95 / Ada 2005**
- SEAS Core SW: class-wide programming for Processes, Signals, HW Interfaces, Scaling, Processors, Testing Equipment …
- “Factory Pattern”: deployment of distributed objects from the Central Objects Factory
- The backbone used to articulate components integration is based on two middleware solutions:
  - AdaCore GLADE / PolyORB with Ada DSA personality
  - SEASCOM (proprietary) network application protocol
- Thanks to the capability to flexibly exchange simulation models with real hardware SEAS can support the whole development lifecycle

AIRBUS DEFENCE & SPACE

SEAS Features (II)

- Simulation environment: multi-platform, multi-OS running in heterogeneous machines
- Multi-language support for simulations and bench specific SW
  (Ada, C, C++, Java, VB, FORTRAN, NI-LABVIEW, PYTHON...)
- HW access employs a flexible mechanism which makes it possible to connect it to internal variables belonging to a model, following the “Layer Pattern” where the connecting layer code is automatically generated
- Integration with COTS (commercial off-the-shelf) products:
  - industrial test tools by means flexible adapters
  - avionics and non-avionics I/F cards (A429, AFDX, 1553, EFEX, CAN, Analogue, Digital, Discrete, Ethernet, Shared Memory, High Speed data links)
  - HMI tools (GtkAda,VI, process control…)
  - multimedia products for realistic outside world 3D simulation

AIRBUS DEFENCE & SPACE

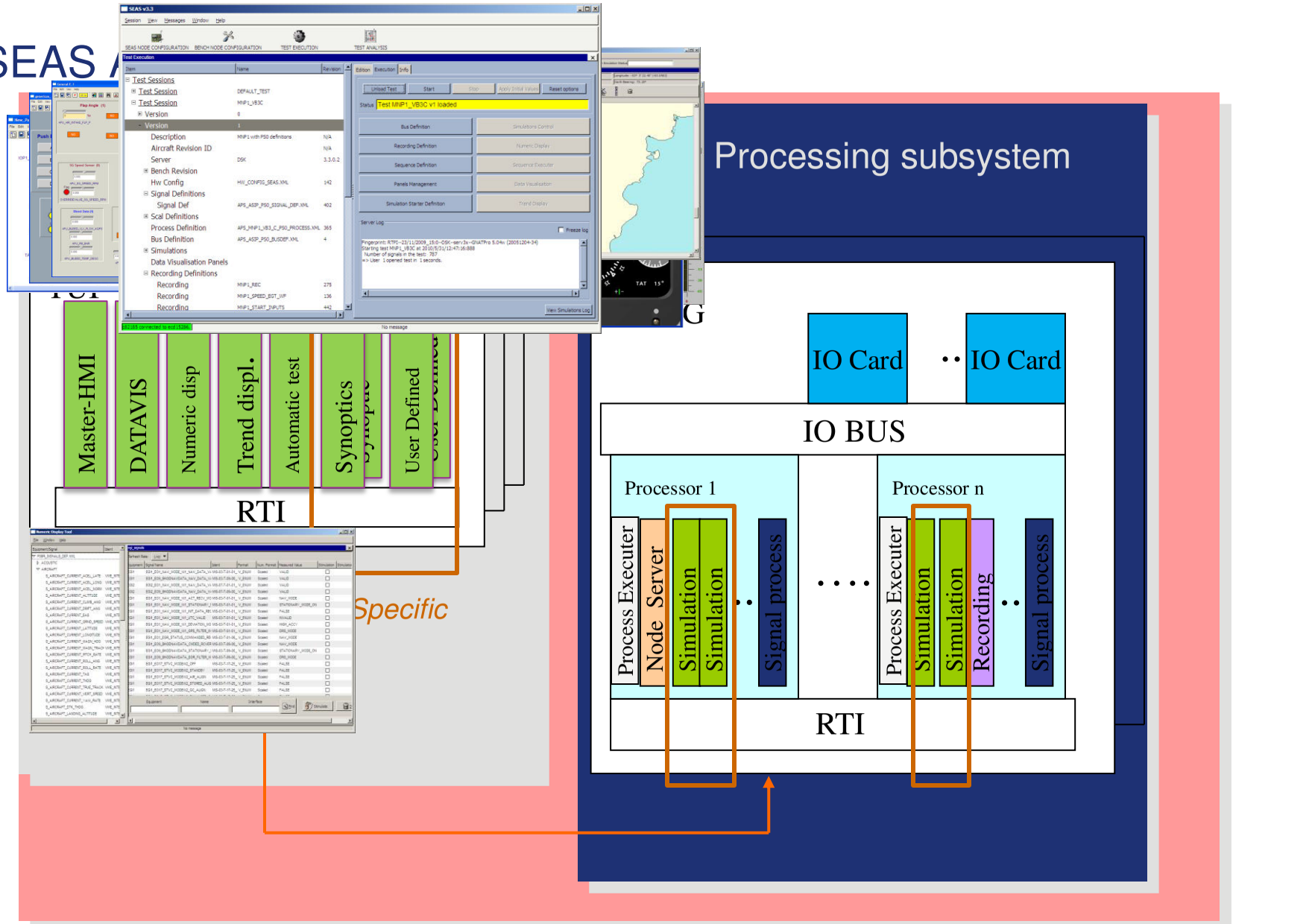
A Test System built around AdaCore GNAT Pro Toolsuite

Our software development process is based on the AdaCore GNAT Pro Toolsuite. Its comprehensive set of tools include all the elements necessary to produce our test solution, including key features like support for distributed applications and multi-language integration:

- **GNAT Pro's Integrated Development Environment** (GPS)
- **GNAT Pro Ada Compiler** (Ada 2012/2005/95/83)
- **GPRbuild**
- **GLADE, PolyORB/DSA**: Ada Distributed Systems Annex for shared memories, RPC, data, types and objects distribution
- **GPS Visual Debugger**
- **GNAT Libraries and Bindings** (POSIX API, Win32 API, …)
- **GtkAda** for HMI and for special synoptic like mission scenarios, radar display simulations
- **XML/Ada Pro XML processing library** (supporting XML, SAX, DOM) for test system definitions, bench HW/SW definition

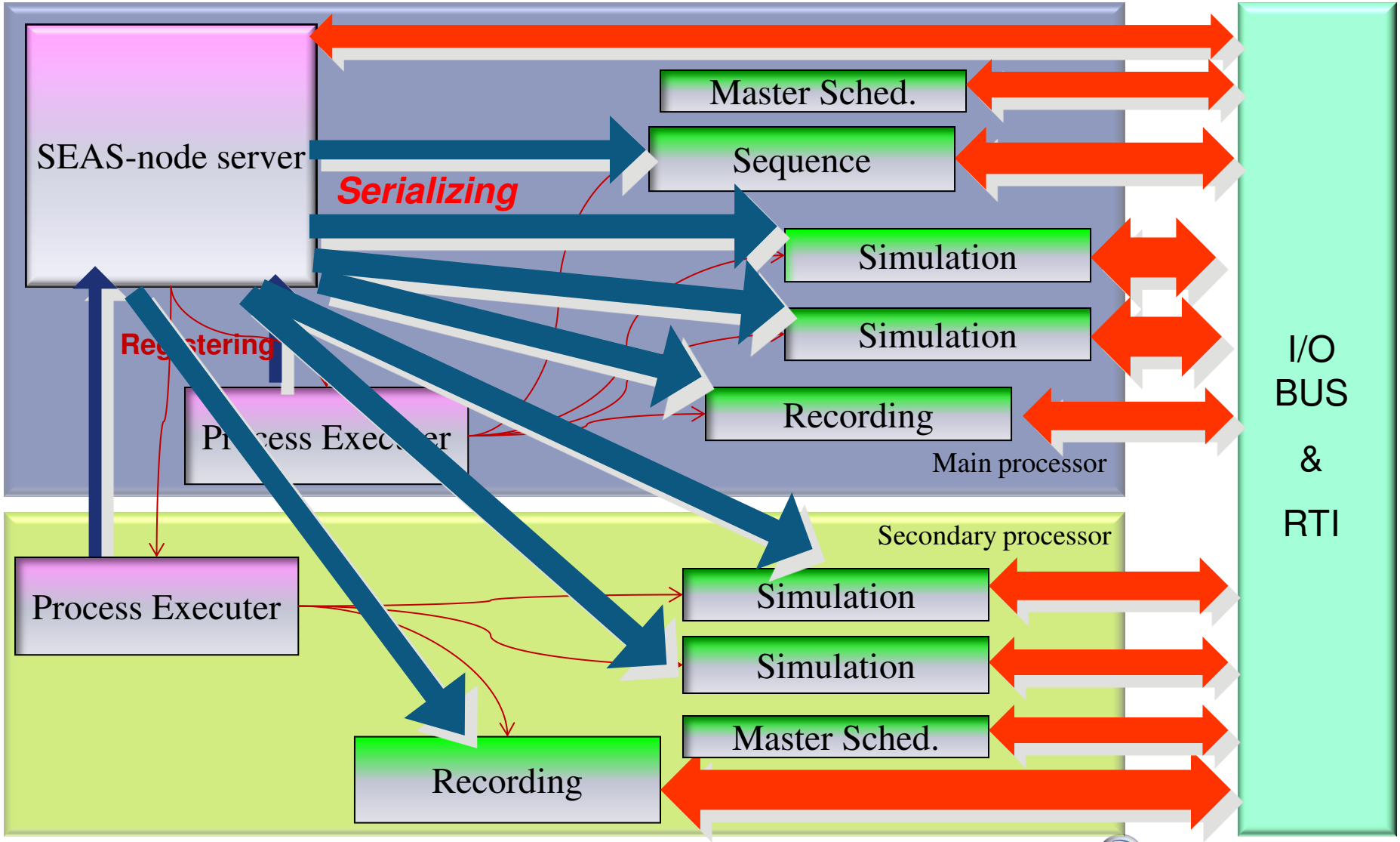
AIRBUS DEFENCE & SPACE

SEAS A



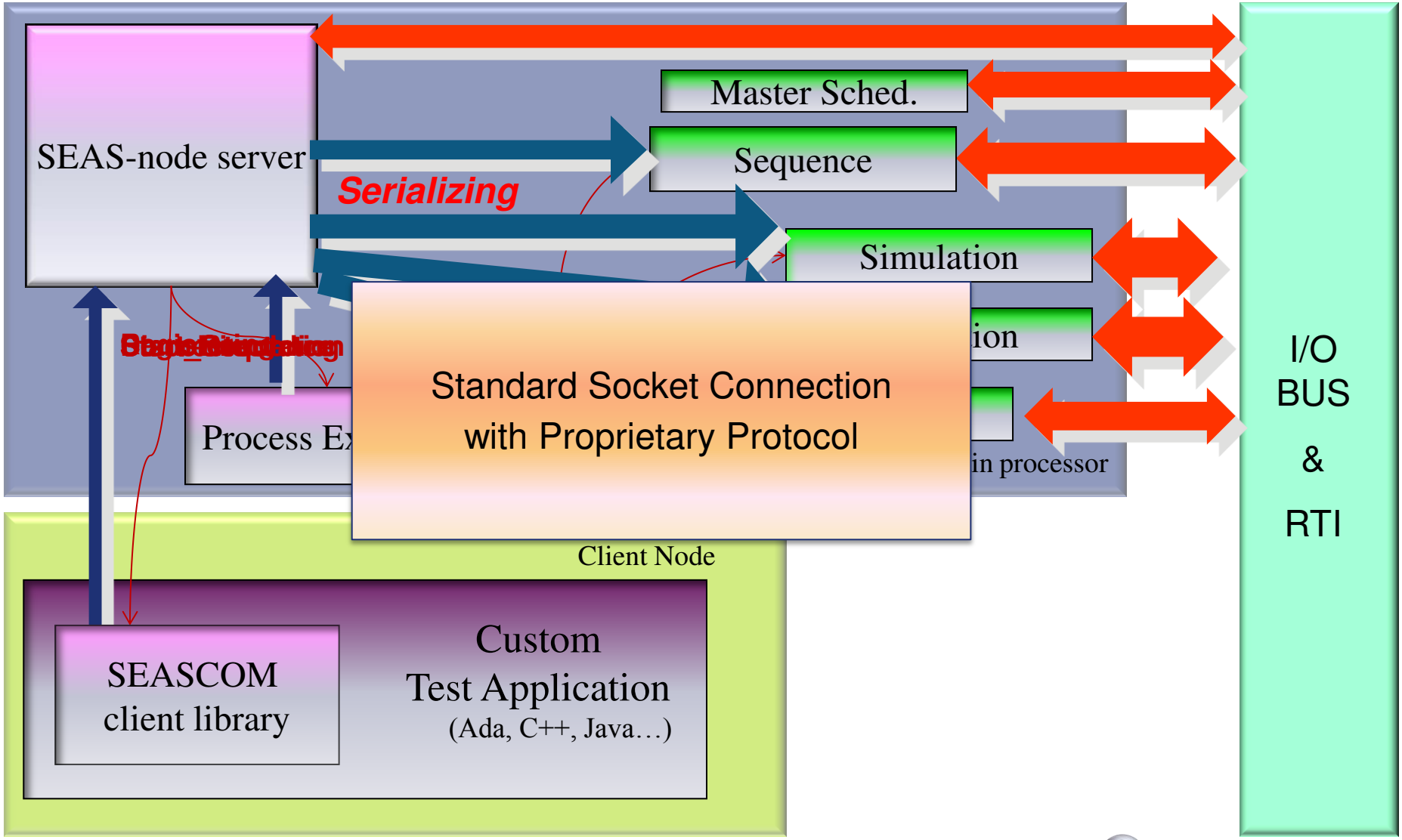
AIRBUS DEFENCE & SPACE

Signal Process Management



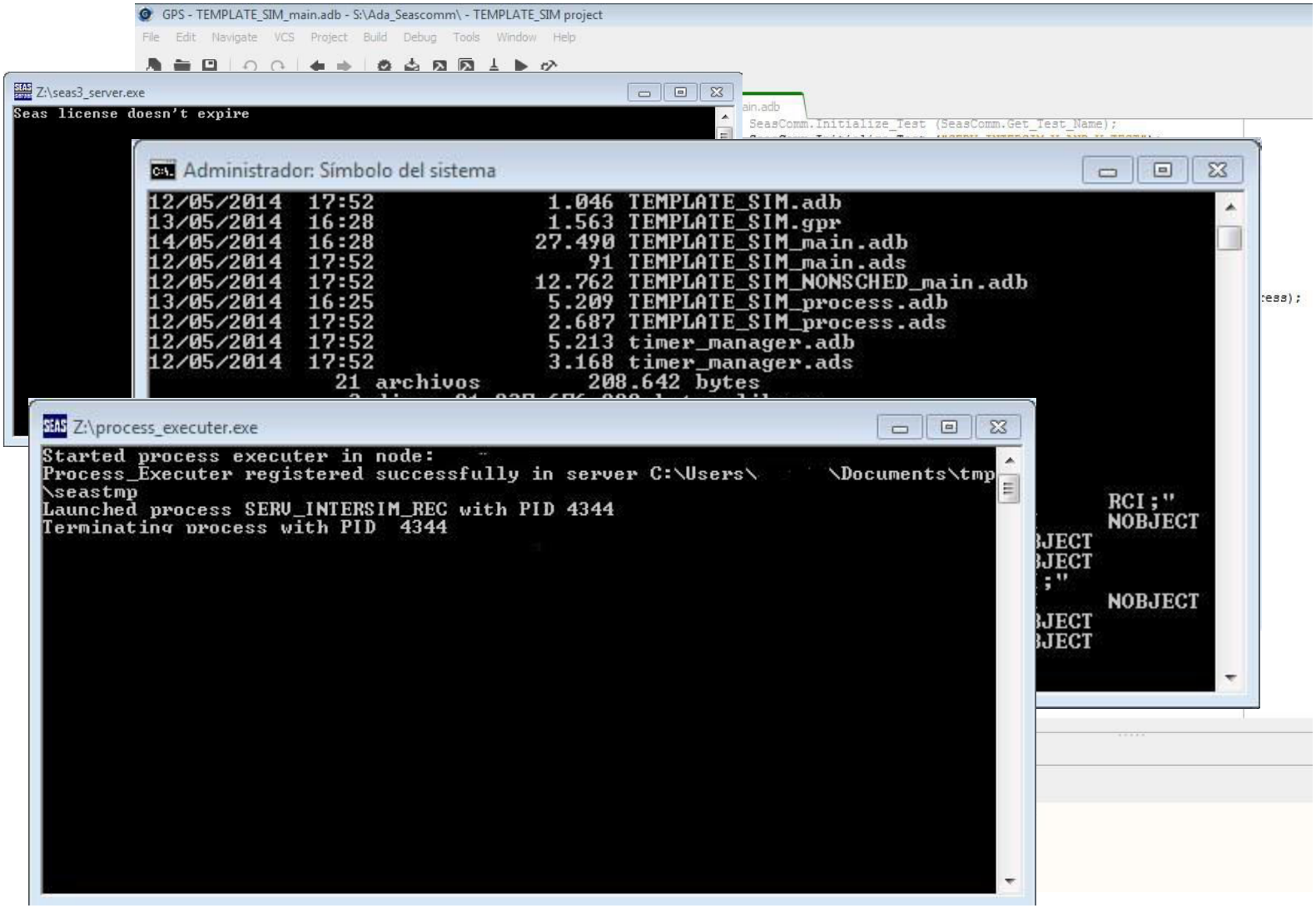
AIRBUS
DEFENCE & SPACE

SEASCOM Interface



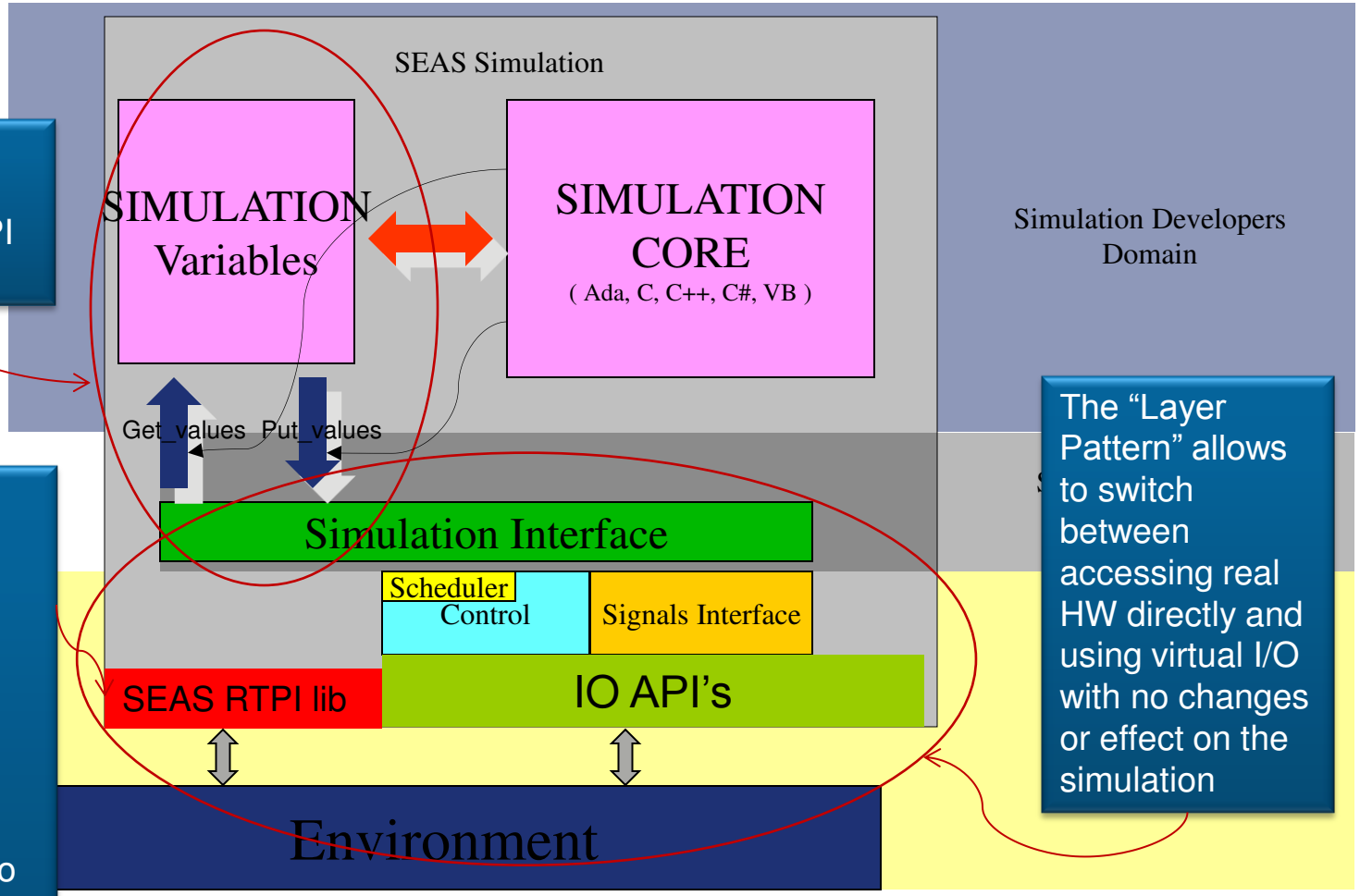
AIRBUS
DEFENCE & SPACE

SEASCOM Interface (II)



AIRBUS DEFENCE & SPACE

SEAS Simulations Layers



Automatically generated by SEAS user API for Ada or C

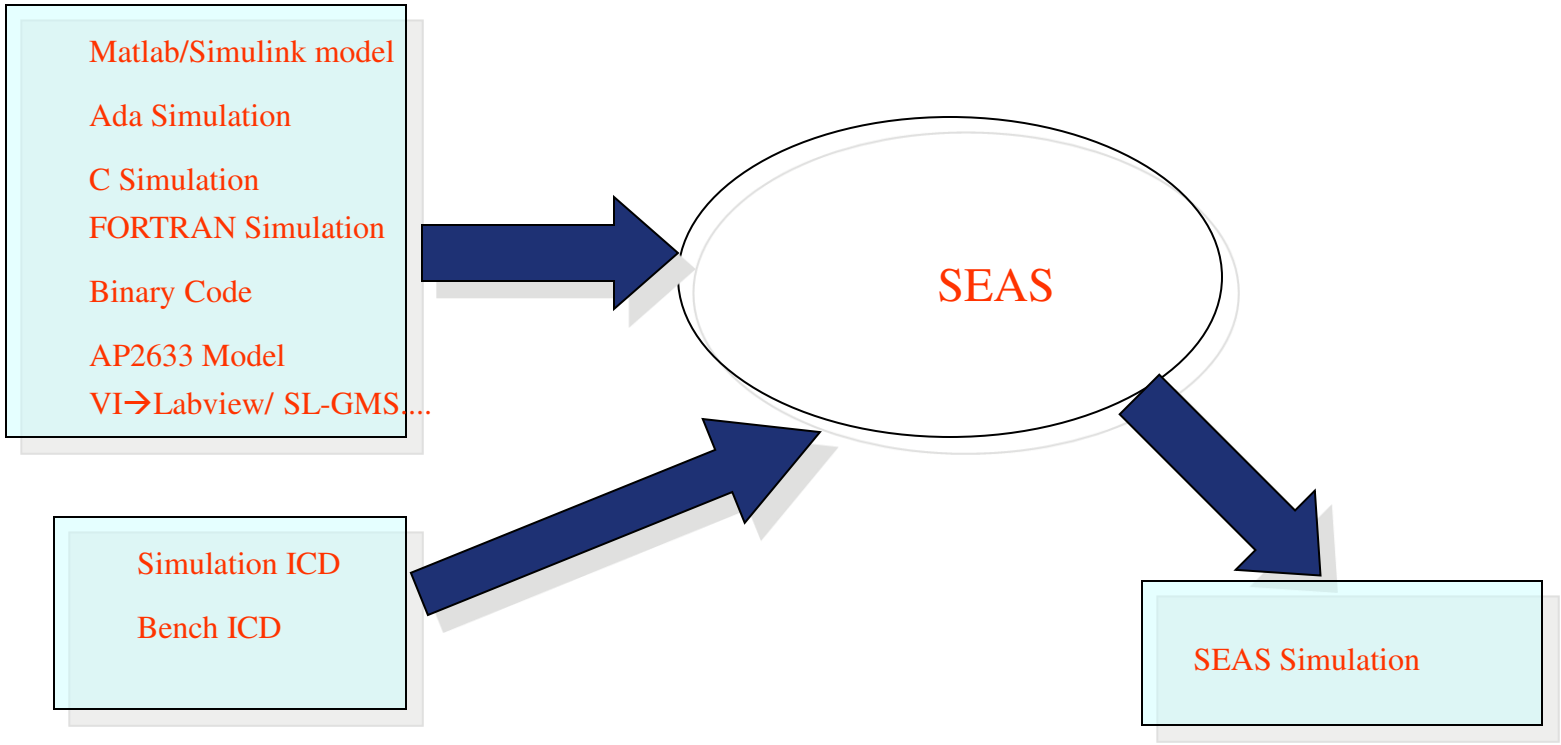
Dynamically:

- receives required signals and HW interface objects serialized from server during simulation initialisation
- adds new signals to server
- creates 'Bundle Signals'

The "Layer Pattern" allows to switch between accessing real HW directly and using virtual I/O with no changes or effect on the simulation

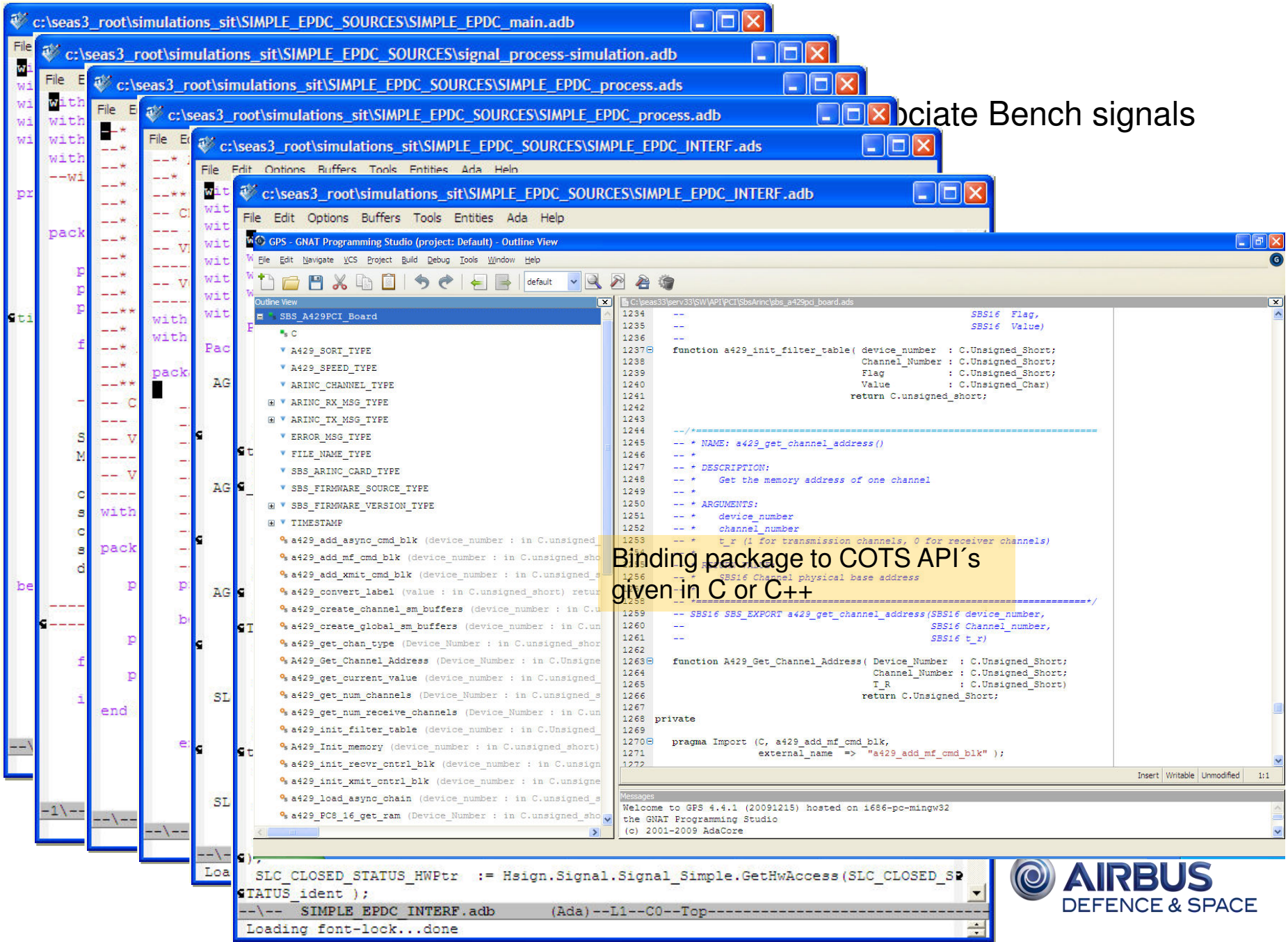
AIRBUS DEFENCE & SPACE

Simulations Build



AIRBUS DEFENCE & SPACE

ociate Bench signals



Binding package to COTS API´s given in C or C++

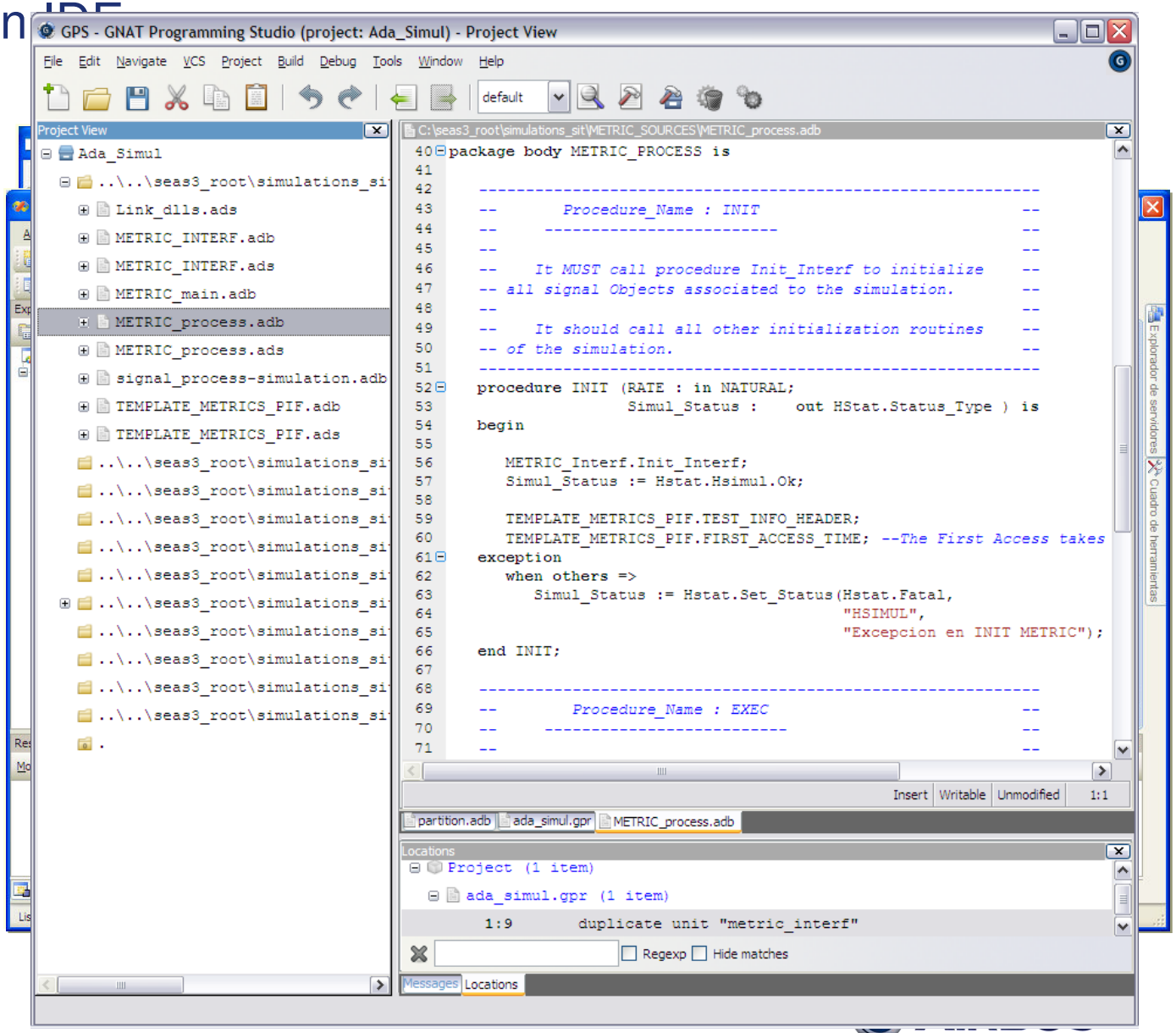
AIRBUS DEFENCE & SPACE

SEAS Simulation IDE

SEAS connected to native IDE for simulations and specific test tools

For Ada and C Code, generating a GPS project and launching GPS

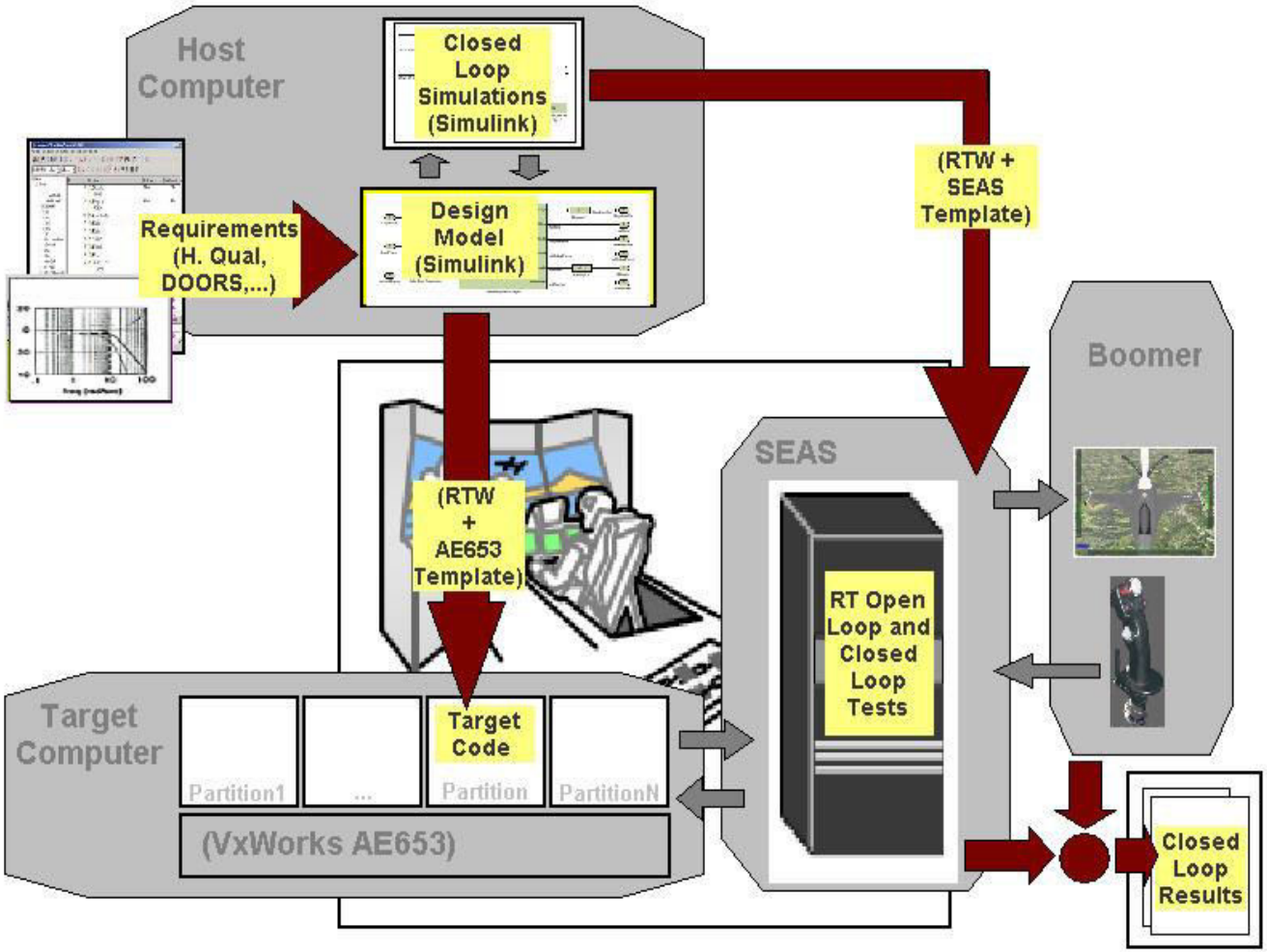
… Or for Visual C source code, generating a Visual C project and launching Visual Studio Application



DEFENCE & SPACE

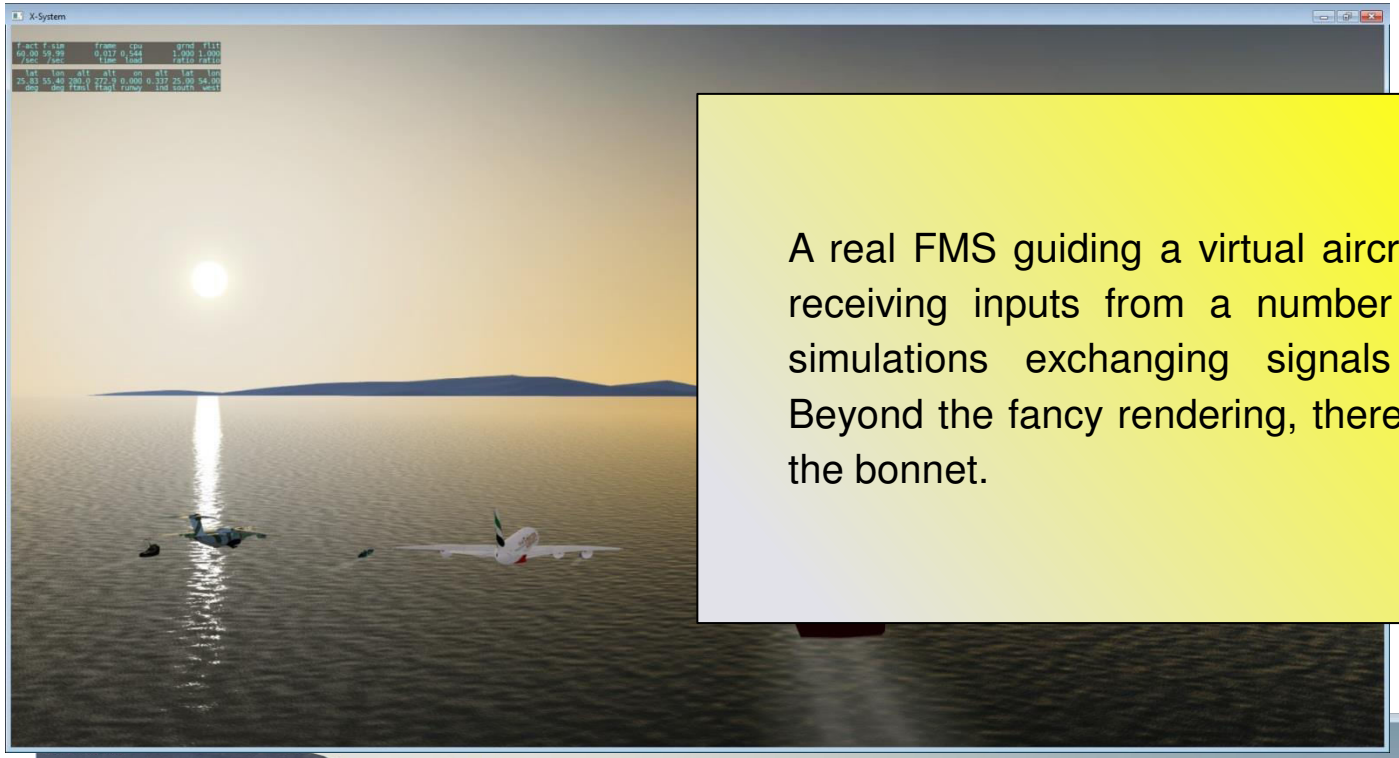
Tools Integration: Matlab Simulink models to SEAS Route

ARBS application

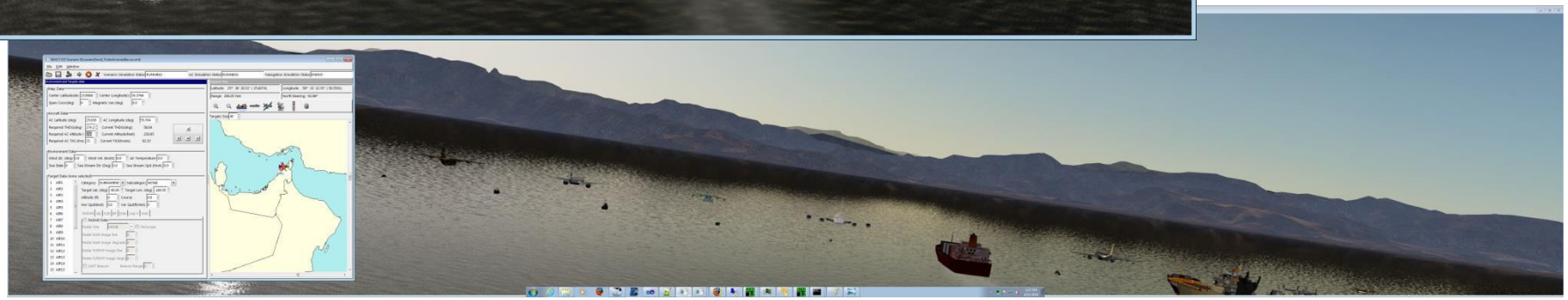


AIRBUS DEFENCE & SPACE

Tools Integration: Integration with realistic 3D modelling tools



A real FMS guiding a virtual aircraft on autopilot, receiving inputs from a number of coordinated simulations exchanging signals in real time. Beyond the fancy rendering, there is SEAS under the bonnet.



AIRBUS DEFENCE & SPACE

Summary

- Our testing solution, SEAS, makes extensive use of Ada with the goal of providing a comprehensive solution to most testing needs throughout the development life cycle of aircraft on-board electronics equipment at Airbus Defence and Space, Military Aircraft
- Test System built around AdaCore GNAT Pro Toolsuite
- Ada 95 / 2005 for SEAS Core SW using class wide programming
- Ada 95 / 2005 for simulations, possibly in combination with many other languages
- GtkAda for HMI, among other solutions
- XML / Ada for data file processing
- Ada Distributed Systems Annex for shared memories, RPC, data, types and objects distribution
- Patterns: “Factory Pattern”, “Central Objects Factory”, “Layer Pattern”
- Future developments aimed at widening the scope of interoperability with third-party products by means of open, standard protocols

AIRBUS
DEFENCE & SPACE

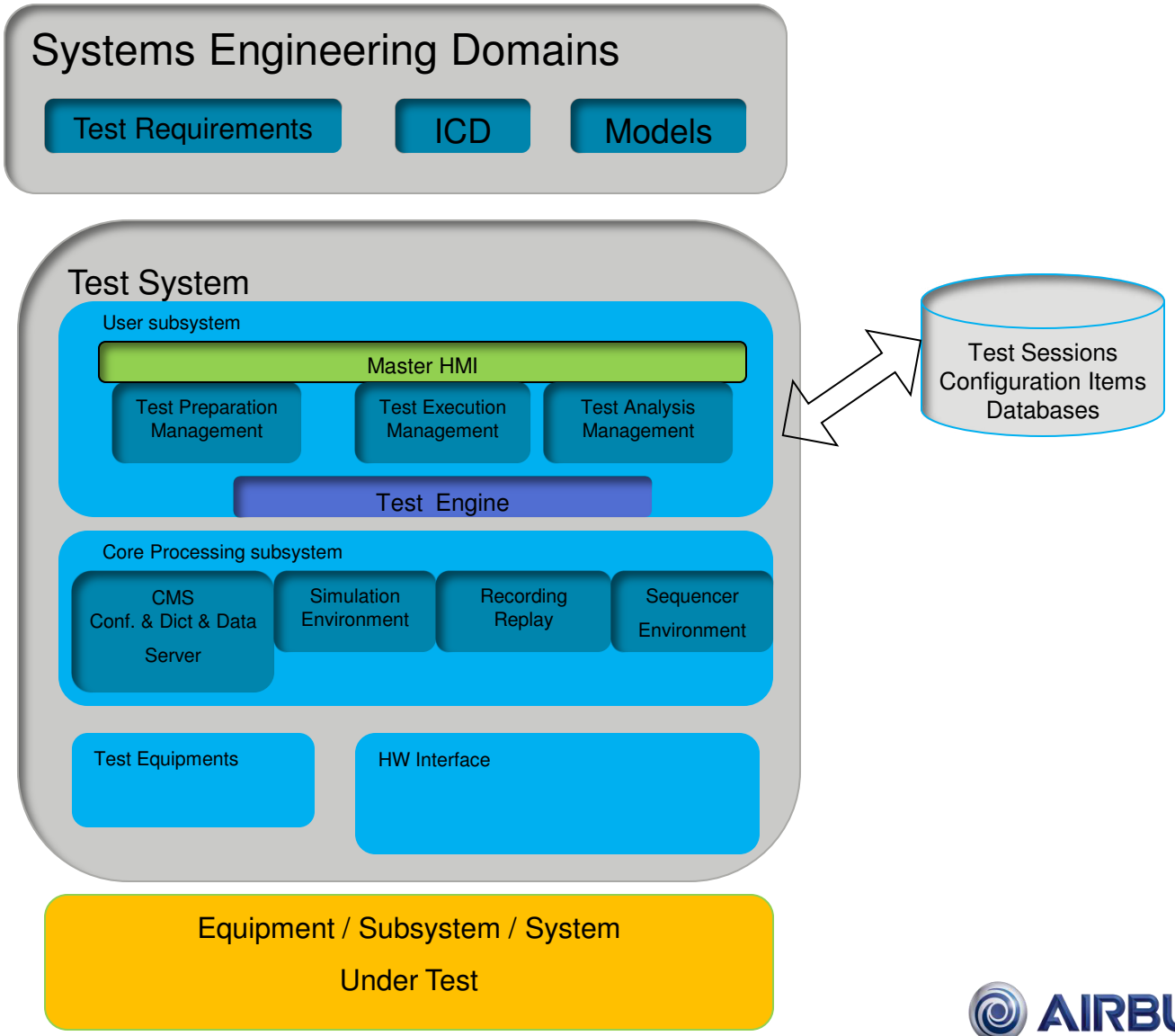
Thank you for your attention

AIRBUS
DEFENCE & SPACE

Support slides

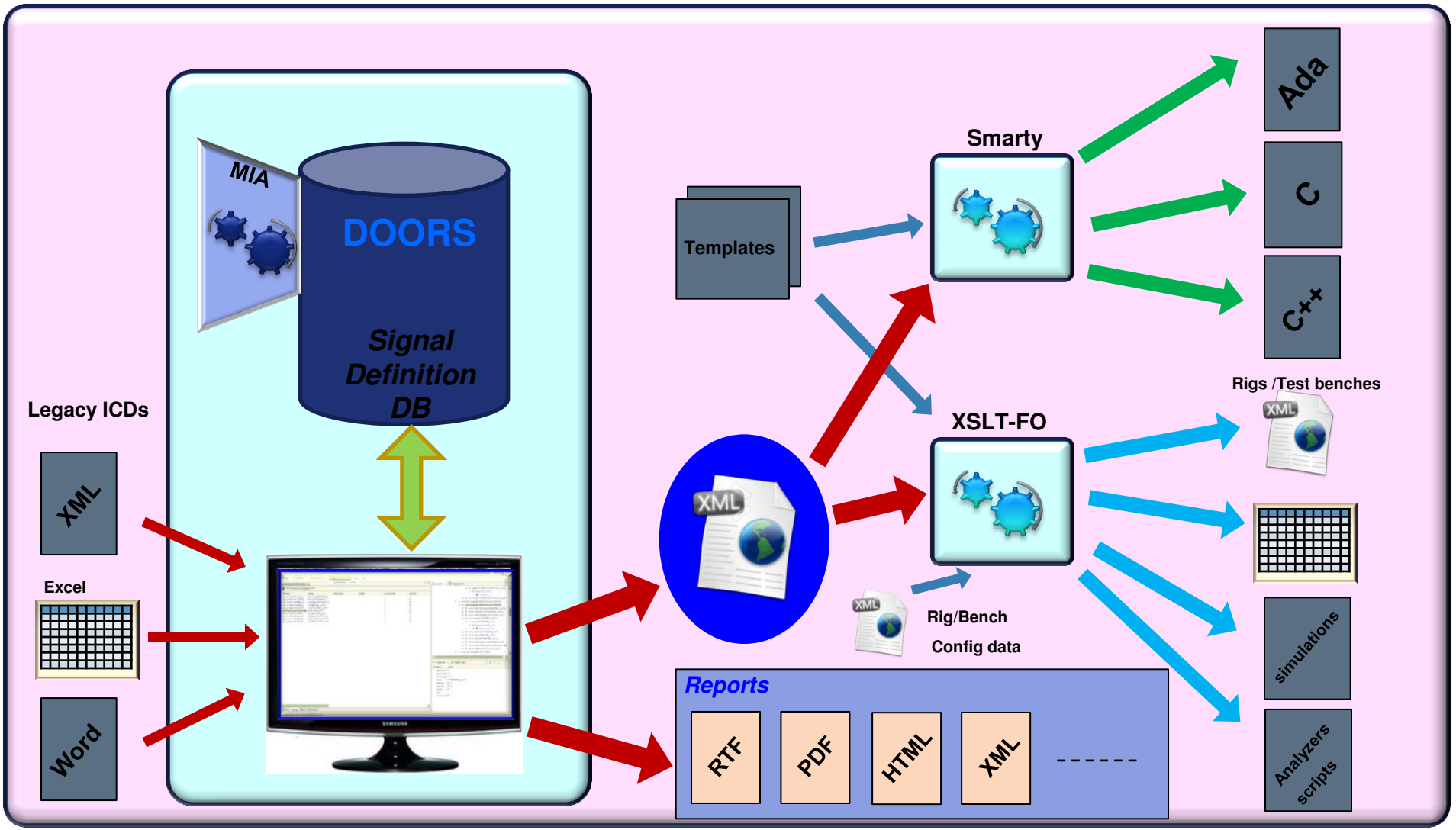
AIRBUS
DEFENCE & SPACE

SEAS Context Diagram



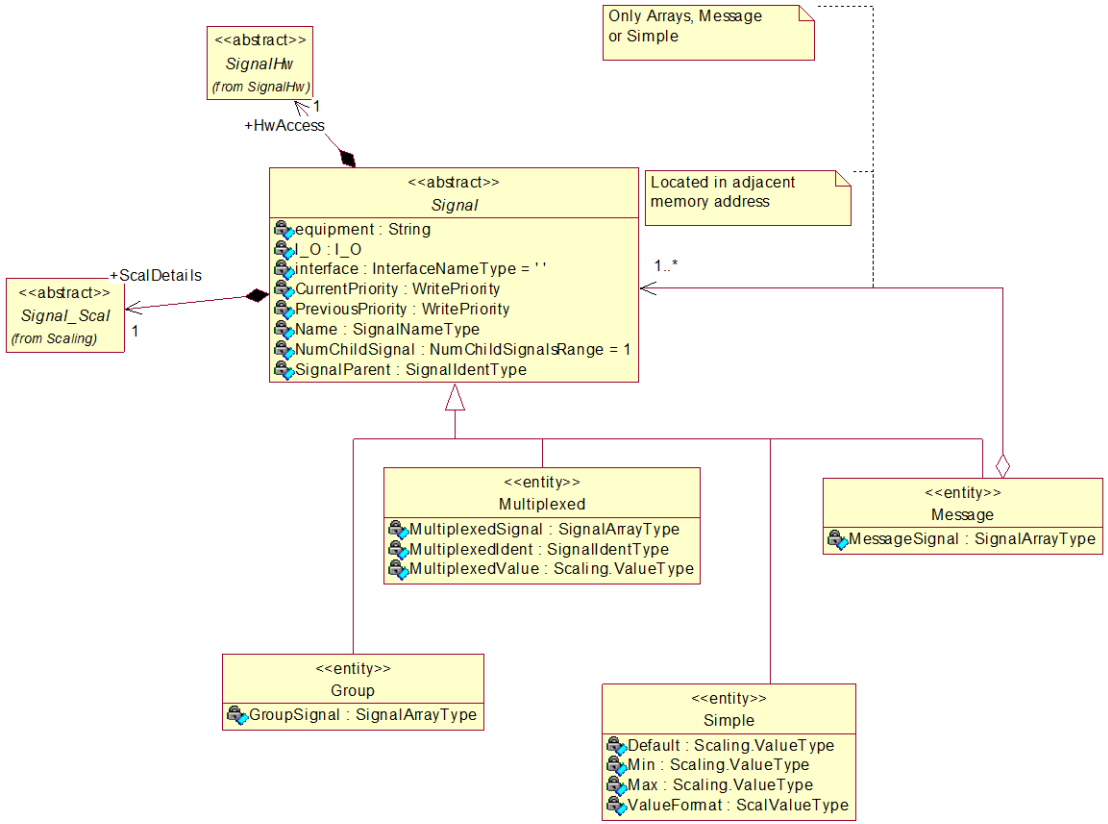
AIRBUS DEFENCE & SPACE

A/C ICD's Management



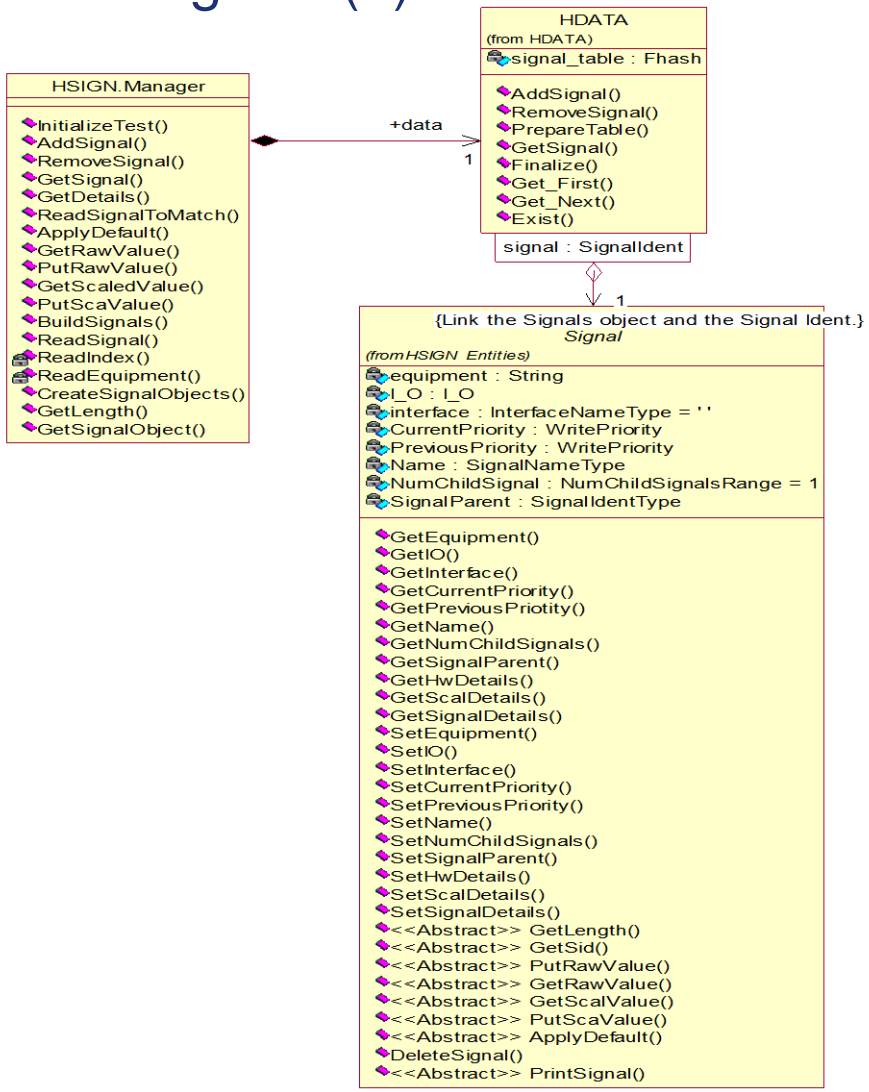
AIRBUS DEFENCE & SPACE

SEAS Signal Class Diagram



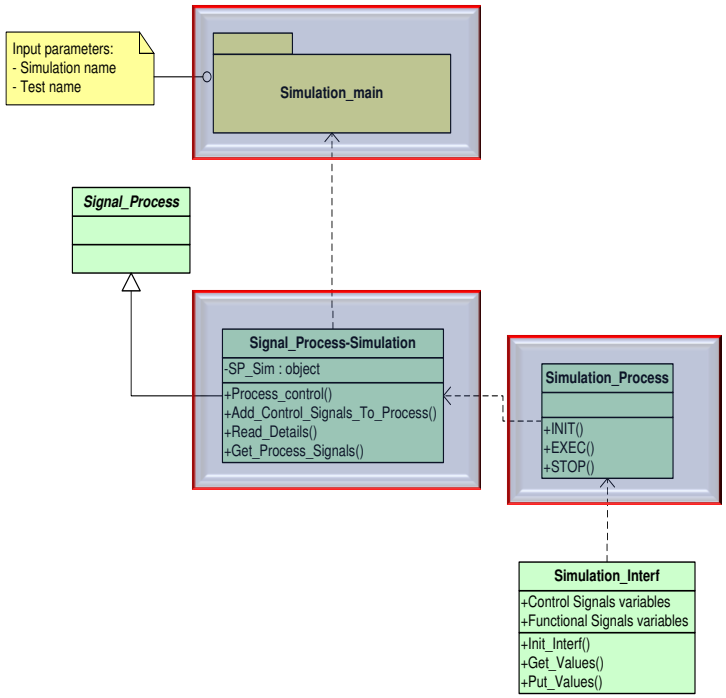
AIRBUS DEFENCE & SPACE

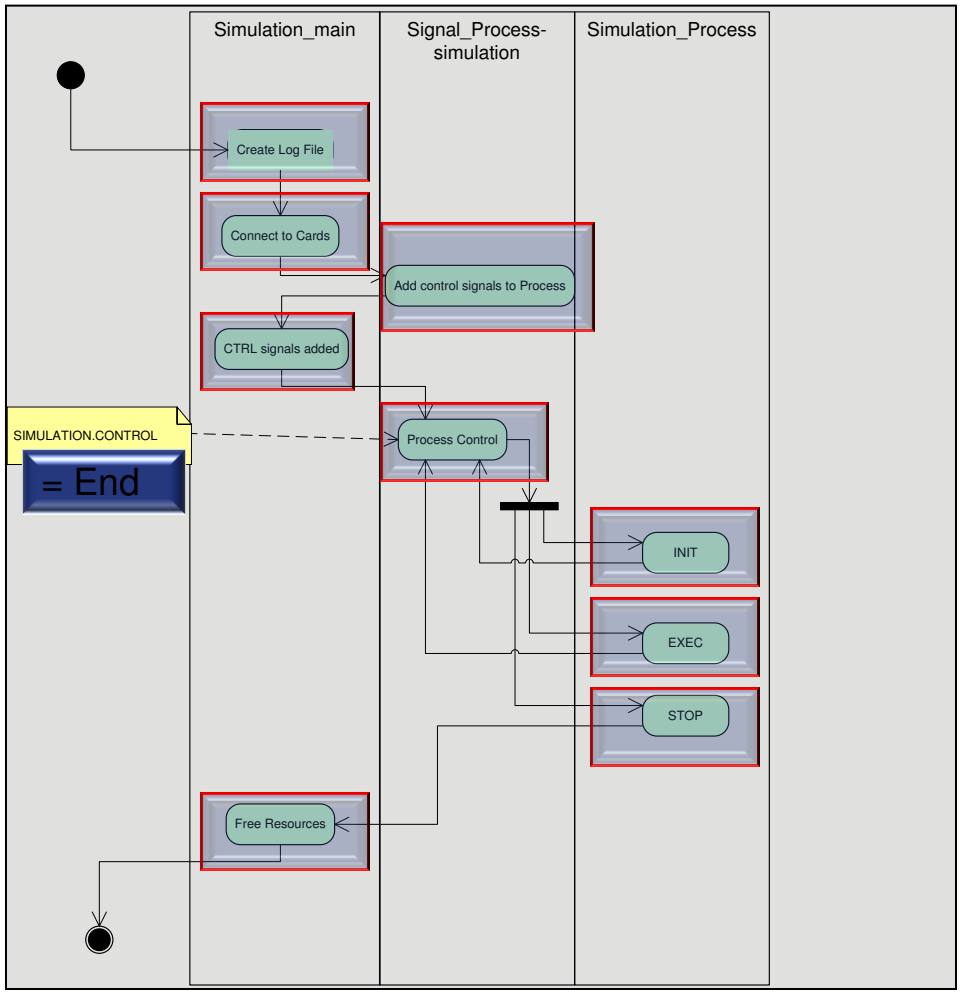
SEAS Signal Class Diagram (II)



AIRBUS DEFENCE & SPACE

SEAS Simulation Class and Activity Diagram





AIRBUS DEFENCE & SPACE

PolyORB

Middleware implementation providing development tools and an innovative runtime library architecture for collaboration of application components using open standards for distributed systems.
PolyORB is a middleware toolset that provides distribution services through standard programming interfaces (e.g. CORBA, the Ada Distributed Systems Annex, or the MOMA messaging API) and communication protocols (e.g. GIOP and SOAP). It addresses distribution model interoperability issues by allowing a single middleware instance to efficiently support multiple personalities executing simultaneously. Its modular architecture emphasizing code reuse allows the definition and deployment of middleware configurations that are specially adapted for real-time, high integrity applications.

Ada Distributed Systems Annex (DSA)
The Ada Distributed Systems Annex allows easy creation and deployment of distributed applications using standard language features. The construction of a distributed application is facilitated by leveraging on existing Ada constructs to identify distributed component boundaries and interfaces. Applications can be written as though not distributed, and later on partitioned into multiple subsets assigned to distinct nodes, and communicating through remote subprogram calls and shared data.

The PolyORB/DSA implementation includes:

- a distribution stubs generator, included in the GNAT Pro compiler;
- GNATDIST: a partitioning tool allowing the generation of executable images for each partition in a DSA distributed application
- a PCS (Partition Communication Subsystem) implemented as personality modules within the PolyORB framework, providing runtime communication services and distribution support to application components.