OCORA

Open CCS On-board Reference Architecture

Benchmarking Report Modular Testing

Document ID: OCORA-TWS09-020

Version: 2.0

Release: R1

Date: 19.11.2021

Management Summary

This document synthetizes the conclusions and lessons learned from benchmarking integration & testing strategies from 2 different perspectives: experience sharing from different business fields / domains similar to OCORA architectural modular choices (automotive, aircraft) and among NS, DB, SBB and SNCF (based on implementation cases, i.e. lessons learned from projects.

This document is an input for the “testing strategy” guideline (i.e., integration, validation and acceptance) for OCORA that will be defined in the [7] document.

Major outcome is the need of a well-structured approach mirroring the modular architecture with clear responsibilities defined at each level of integration. A second major outcome is the need for an Integrator in charge of (among other duties) collecting all evidence for each individual block and interface and providing the evidence for the integrated system.

Revision history

Version	Change Description	Initial	Date of change
1.00	Official version for OCORA Delta Release	SC	25.06.2021
2.00	Official version for OCORA Release R1	SC	19.11.2021

Table of contents

Table of figures

Table of tables

None

References

Reader's note: please be aware that the numbers in square brackets, e.g. [1], as per the list of referenced documents below, is used throughout this document to indicate the references to external documents. Wherever a reference to a TSI-CCS SUBSET is used, the SUBSET is referenced directly (e.g. SUBSET-026). OCORA always reference to the latest available official version of the SUBSET, unless indicated differently.

The different sources used in this report are listed below: we call them explicitly within the section 2 where they are used. We warmly thank the authors who put them on public access.

[1] OCORA-BWS01-010 – Release Notes

[2] OCORA-BWS01-020 – Glossary

[3] OCORA-BWS01-030 – Question and Answers

[4] OCORA-BWS01-040 – Feedback Form

[5] OCORA-BWS03-010 – Introduction to OCORA

[6] OCORA-BWS04-010 – Problem Statements

[7] OCORA-TWS09-010 – Testing – Strategy (Guideline for modular testing)

[8] VICTORIA : Validation platform for Integration of standardised Components, Technologies and Tools in an Open, modular and Improved Aircraft electronic system - Camille Ragi

[9] Feedback on IMA certification and on-going regulatory work in Europe - Integrated Modular Avionics (IMA) conference 2012- Thales - M. C. Chevrel

[10] Handbook for Real-Time Operating Systems Integration and Component Integration Considerations in Integrated Modular Avionics Systems- DOT/FAA/AR-07/48- 01.2008

[11] Autosar Introduction generic presentation for conferences & co - 27.10.2020

1 Introduction

1.1 Purpose of the document

The purpose of this document is to synthetize the conclusions and lessons learned from benchmarking integration & testing strategies.

The benchmarking approach has been conducted from 2 different perspectives:

Experience sharing from different business fields / domains (with modular concerns like OCORA: automotive, aircraft): either directly with actors (meetings, interviews) or via bibliography research (collect experience documentation in modular IVV approach).

Experience sharing among NS, DB, SBB and SNCF (based on implementation cases) (i.e. lessons learned from projects with "high level" IVV: methods, difficulties…).

This document is an input for the “testing strategy” definition (i.e., integration, validation and acceptance) for OCORA that will be defined in the [7] document.

This document is addressed to experts in the CCS domain and to any other person, interested in the OCORA concepts for on-board CCS. The reader will gain insights regarding the topics listed in chapter 1.1, and is invited to provide feedback to the OCORA collaboration and can, therefore, engage in shaping OCORA. Feedback to this document and to any other OCORA documentation can be given by using the feedback form [4].

If you are a railway undertaking, you may find useful information to compile tenders for OCORA compliant CCS building blocks, for tendering complete CCS system, or also for CCS replacements for functional upgrades or for life-cycle reasons.

If you are an organization interested in developing CCS building blocks according to the OCORA standard, information provided in this document can be used as input for your development.

Before reading this document, it is recommended to read the Release Notes [1]. If you are interested in the context and the motivation that drives OCORA we recommend to read the Introduction to OCORA [5], and the Problem Statements [6]. The reader should also be aware of the Glossary [2] and the Question and Answers [3].

1.2 Applicability of the document

The document is currently considered informative but may become a standard at a later stage for OCORA compliant on-board CCS solutions.

1.3 Context of the document

This document is published as part of the OCORA Delta release, together with the documents listed in the release notes [1]. Before reading this document, it is recommended to read the Release Notes [1]. If you are interested in the context and the motivation that drives OCORA we recommend to read the Introduction to OCORA [5], and the Problem Statements [6]. The reader should also be aware of the Glossary [2] and the Question and Answers [3].

2 Bibliography & Experience sharing with actors outside OCORA community

2.1 Lessons from modular avionics

The different bibliography research performed are done on Integrated Modular Avionics (IMA) related papers. IMA is a highly integrated architecture for aircraft systems involving HW, SW, bus communications…, IMA permits recurrent, development and maintenance cost savings optimizing industrial business model.

Hence, IMA is a very similar approach as OCORA in terms of architectural modular approach and constraints (safety, platform independence, real time, multi-actors…).

Three major papers/inputs are detailed hereafter. The others are given for information but deliver similar conclusions or allow to go deeper in such or such filed but for the interest of OCORA major lessons are extracted hereunder with this two documents analysis.

The two first papers (see 2.1.1 and 2.1.2) are used to introduce Integrated Modular Avionics (IMA) principles and understand this ecosystem.

The third one (2.1.3) is the summary of the rules and practices adopted by the IMA. It is based on its reference handbook that is itself the result of years of research and sectorial collaborations.

Lessons learned from IMA are gathered in this third section (2.1.3).

Other interesting papers are mentioned in 2.1.4.

2.1.1 Introduction to IMA/IME based on VICTORIA project – Validation platform for Integration of standardised Components, Technologies and Tools in an Open, modular and Improved Aircraft electronic system

2.1.1.1 Context

This paper is a feedback on VICTORIA, a major European R & D program in Aerospace [8].

Throughout the aircraft life cycle, the cost of modifications, including parts obsolescence mitigation and functional upgrades has becomes even more significant for the airlines. Similarly, the demand for new on-board functions and services has increased to the extent that on-board electronics is definitely a major and increasing differentiation factor for the whole chain of Air Transport industry - airlines, aircraft manufacturers and electronics equipment manufacturers.

One technological response to the processing capacity growth vs electronics cost is the Integrated Modular Electronics (IME) concept which, is based on sharing a bare standardised processing platform (called Line Replaceable Modules or LRM) by several functions.

An IME platform provides the following capabilities:

1. Share of the Hardware Resources

Multiple functions can be hosted on the same platform •

2. Independence of the SW vis-à-vis the HW

A standard OS API layer enables the function to make an “abstraction” of the platform hardware. •

3. Hosting Functions of different criticality

This capability is enabled thanks to :

- Time & Space Partitioning mechanisms
- Failure confinement mechanisms (One function fail can’t cause another function to fail)

2.1.1.2 The VICTORIA program

The VICTORIA program has investigated the need for new equipment and applications, as well as the tools and methods to design them.

These are respectively:

• The preparation of new standards

• The definition of the "validation platform"

• The preparation of new development tools and means

• The implementation of components and modules

• The development and modification of the software application exercisers

• The integration, evaluation and validation testing of resources and applications

One essential achievement of VICTORIA is the realisation of an integrated VICTORIA platform encompassing six domains: Cabin, Cockpit, Utilities, On-board Information System (OIS) and Passenger and Crew Electronic Services (PCES)distributed in three different sites.

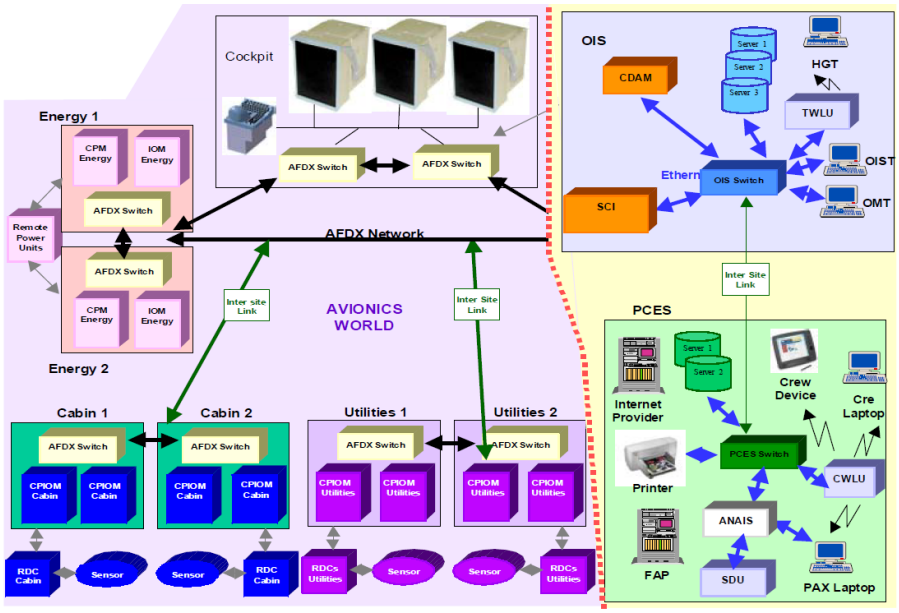


Figure 1 Example of IMA/IME Validation platform [8]

The introduction of new technologies and new avionics concepts has a significant impact on development process, tools, integration and validation means in order to comply with the new roles and responsibilities and with the certification objectives.

As VICTORIA takes into consideration the whole sequence of IME development activities a set of means and tools were selected to cover the following areas and development steps:

• System design and modelling

• Application development

• Platform Integration

• System Integration and Validation means

• Component Conformance suites

2.1.2 Introduction to IMA/IME based on the feedback on Integrated Modular Avionics certification

2.1.2.1 Context

Some results are directly issued from a conference related to Integrated Modular Avionics (IMA) held by Thales [9], some other are issued from meetings with WindRiver (together with OCORA Computing Plateform) in April-May 2021.

2.1.2.2 Short description of the bibliographical input

Thales presents the changes/difficulties induced by moving to a system platform composed by a set of modules (specific and highly configurable computers, multiple systems applications are executed on the same platform and network): the IMA architecture (including networks) is considered as a complex system of the aircraft. Performance and safety of integrated module shall be granted in any operational situation.

One of the main idea is to that independent qualification of some components and credit from some components pre-qualification allow to simplify final approval.

ARP4754/4761 and more recently DO297 are structuring IMA system development and certification processes. The different tasks preceding initial approval are :

- Module Acceptance,
- Application acceptance (based on evidences provided by Application Supplier – see hereunder)
- and IMA System acceptance (and finally Aircraft integration level).

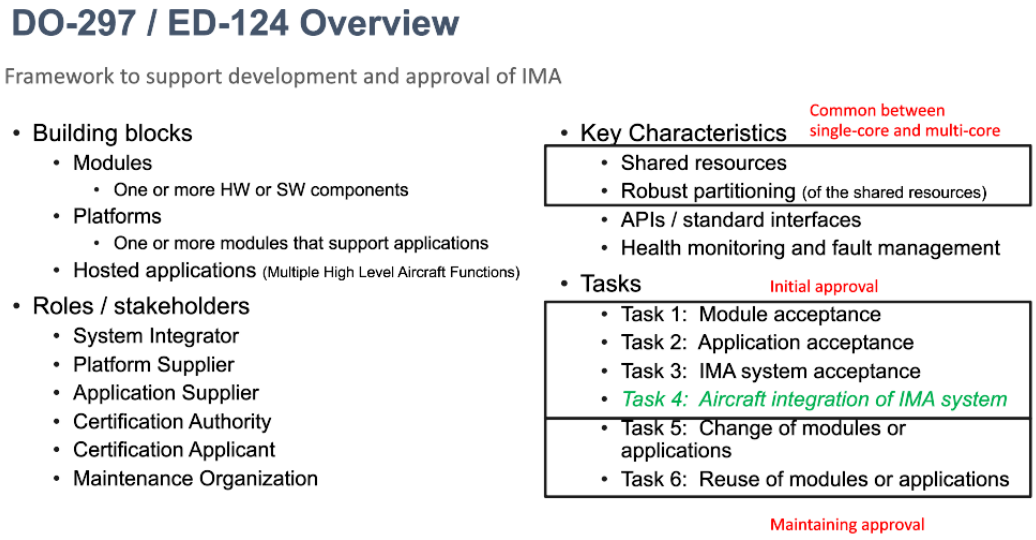


Figure 2 Development and certification/approval process for IMA (WindRiver)

The following actors are identified:

Platform / Module supplier:

- Production, Supply chain, component obsolescence management and capacity to design in the future
- In Service Experience on COTS hardware component (Certification constraint)
- RT Operating System skills
- Robust Partitioning demonstration (Partitioning) skills

Sub-System Designers / Application Suppliers

- Functional domain (Flight Management, Fuel, Cabin...) skills
- Functional oriented Software engineering skills

IMA system integrator

- Complex integration (mixing software and functional aspects) skills
- Incremental Integration & Acceptance

⇨ IMA objective is to select the best supplier for each task taking into account each specificity

The figure bellow depicts the different stakeholders and steps. We can clearly see the importance of the Platform and the role of the System Integrator assembling/configuring the complete system based on these platforms (and their platform usage domain) and of the evidences provided by the Application Suppliers:

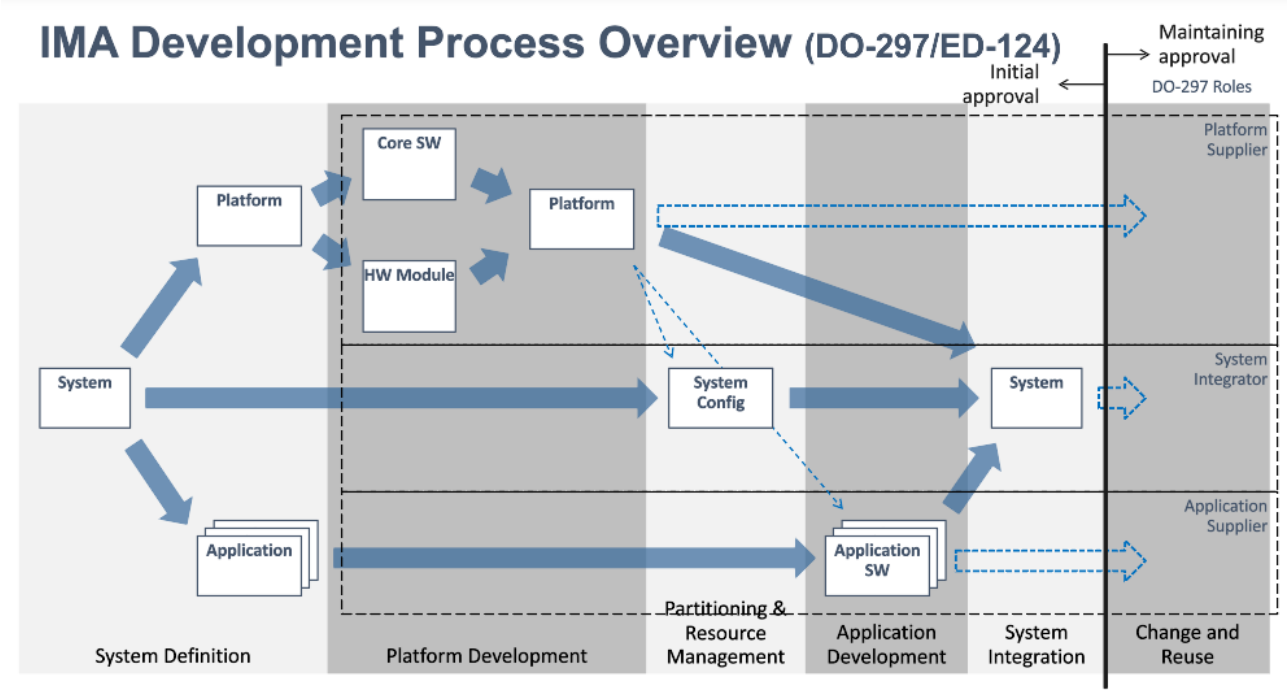


Figure 3 Development and certification/approval process for IMA (WindRiver)

Based on Airbus helicopter experience/illustration hereunder, once a release has been approved, the different changes are handled via change request management based on system baselines: certification artefacts evolve through theses baselines. Depending on the dependencies/interplay between applications, we may go (or not) via DO297 process. This is illustrated bellow.

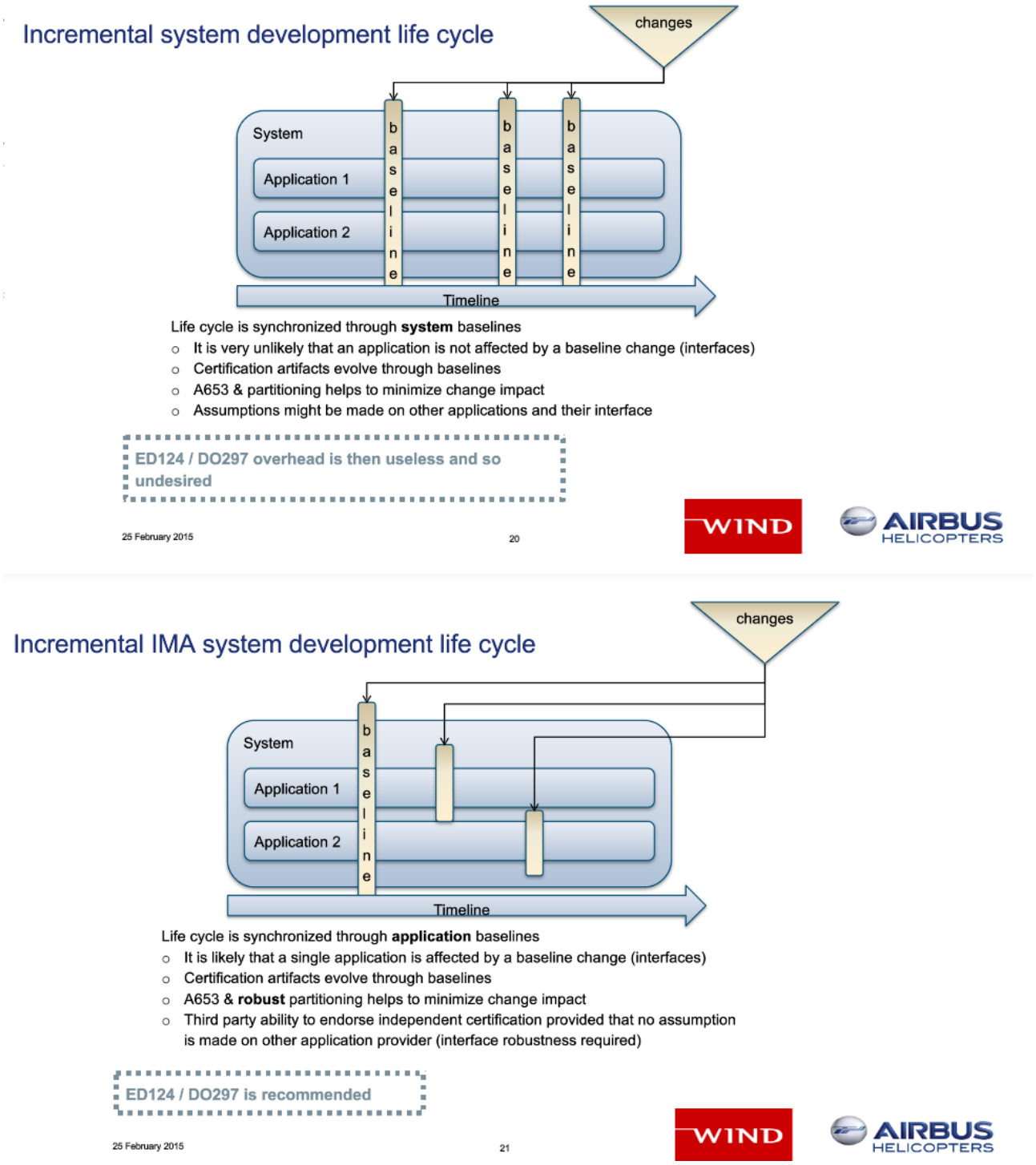


Figure 4 Lifecycle and changes management with or without IMA (WindRiver)

2.1.2.3 Lessons learned

Select the best supplier for each task taking into each specificity in particular the System integrator.

It was warned about potential difficulties during the compliance demonstration in case the incremental approach is not followed. This is derived from the complexity of IMA system.

Incremental qualification process shall be defined to master the interactions between the industrial players.

Incremental qualification shall take benefit from Module & Tool properties (partitioning, configurability & usage domain).

Early agreement on a Certification Program (structured in several domains) is recommended:

- IMA System & Integration domain
- Application software qualification
- Platform qualification (hardware, Operating system and Tools)

Early validation of the HW, SW, SYS Certification Plans reduce the risk:

- Simple and Complex Hardware Components classification
- Clear roadmap for COTS components (In Service Experience, Errata...)

Keep Authority (here Aviation A.) in the loop along the development process

Perform audits in good phasing along with development reviews

Use clear baselines shared among the suppliers/actors, systems.

2.1.3 Systems Integration and Component Integration in Integrated Modular Avionics Systems

2.1.3.1 Context

Aircraft industry has provided a Handbook [10] to aid industry and the certification authorities in the earlier integration stages of Integrated Modular Avionics (IMA) system development. Historically, in typical federated systems, integration was a rather straightforward activity involving compiling, linking, and loading the software application onto the target computer system environment. IMA systems and their ability to integrate several functions with shared resources require further guidelines.

The Handbook is designed to identify the commitments among IMA system role players and between IMA system role players and the operational system.

The principal outcomes reveal:

- the need for role identification in the development, verification, and acceptance and approval processes of the IMA system.
- tracing partial compliance and full compliance objectives supporting certification.
- defining, tracing, and verifying all the commitments required by the IMA system modules and components.
- staged integration and incremental acceptance approaches for module and component configuration control.
- establishment and verification of robust partitioning and other platform services.

Note (from Handbook): DO-178B was originally developed with a federated system architecture view of airborne systems development, and as such, the document lacks IMA system-specific guidance. Aircraft sector wrote a handbook based on the results of an FAA study and attempts to collect and capture what is known about the topic of integrating modular avionic systems, and it offers activities to support the successful development and approval of an IMA system.

Fundamentally, an IMA system is comprised of a set of modules, components, and applications that are integrated into a system that provides aviation functions. Parts of the system may be developed separately

and then integrated together into a functioning system. Often, different teams or organizations with defined roles manage the parts. DO-297 has defined six stakeholder types or roles:

1. Certification authority—Organization or person responsible for granting approval on behalf of the nation of manufacture.

2. Certification applicant—A person or organization seeking approval from the certification authority.

3. IMA system integrator—The developer who performs the activities necessary to integrate the platform(s), modules, and components with the hosted applications to produce the IMA system.

4. Platform and module suppliers—The developer that supplies a module or group of modules, including core software that manages resources in a manner sufficient to support at least one application. A component, or collection of components, can comprise a module.

5. Application supplier—The developer that supplies software and/or application-specific hardware with a defined set of interfaces, when integrated with a platform(s), performs a function.

6. Maintenance organization—Owner or organization responsible for maintaining the IMA system and the aircraft.

Each role plays an important part in the overall development and acceptance of an IMA system, yet special focus is appropriate to the roles of the IMA system integrator, application supplier, and platform and module suppliers. In particular, the platform and module supplier provides the RTOS, hardware, and other support software for the IMA system. The RTOS supplier, as a member of the platform and module supplier role, has critical responsibilities of protection with regards to space, time, input/output (I/O), and other shared resources on the IMA system.

2.1.3.2 Overview of the integration stages

Development of an IMA system requires the due diligence and responsibility of all parties to ensure that system development, assembly, verification, deployment, and maintenance have the attributes of completeness, verifiability, consistency, modifiability, traceability, unambiguity, and recoverability from abnormalities within the system. The parties perform all the roles mentioned above.

In any one particular IMA system development, these roles may be assigned various responsibilities. This assignment should be described in the overall IMA system development plan. Regardless of how the responsibilities are assigned, all of the attributes mentioned above should be addressed, and assurances are required that these attributes contribute to system approval by the certification authority. Each role is involved to some degree in the development and deployment of the IMA system.

Each role generates or accepts commitments and compliance obligations for those parts of the systems over which they assume responsibility. A commitment in this context is defined as any item of an IMA module, application, or component that requires communication or action by another IMA module, application, or component. These include documented assumptions, limitations, constraints, performance restrictions, behavioral restrictions, configurations, and reduced capabilities of the module or component. Compliance in this context documents the credit requested toward satisfying objectives of RTCA documents DO-178B, DO-254, or DO-297, or other applicable guidance.

Compliance with each objective can be fully or partially satisfied. If partially satisfied, then the objective coverage achieved should be documented as well as what remains to be accomplished by another role to fully satisfy the objective's coverage and compliance. The overall system development and the associated assurances of these attributes, commitments, and activities and demonstration of Federal Aviation Regulation compliance is ultimately the responsibility of the certification applicant.

To scope the wide range of IMA system architectures properly, a generic IMA system development is shown in figure bellow as integration stages. These stages increase in functionality and complexity incrementally. They also permit a means for incremental acceptance of modules, platforms, applications, and systems. DO-297 defines incremental acceptance as:

"A process for obtaining credit toward approval and certification by accepting or finding that an IMA module, application and/or off-aircraft IMA system complies with specific requirements. Credit granted for individual

tasks contributes to overall compliance toward the certification goal."

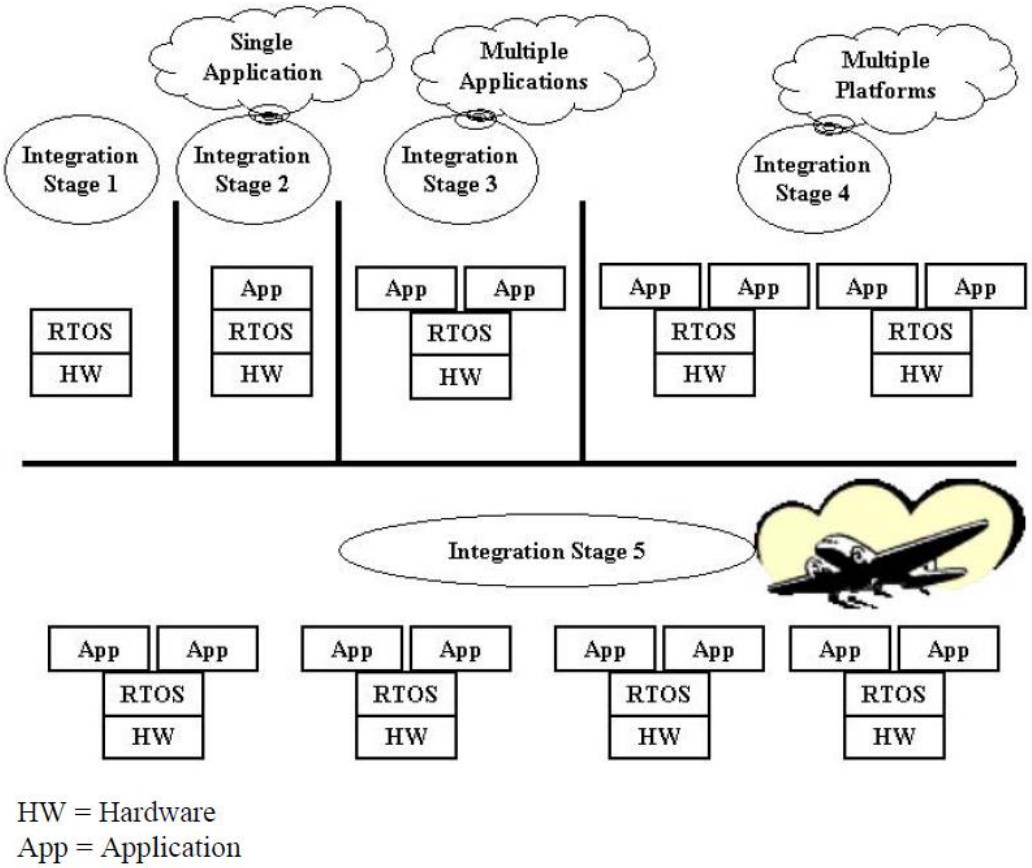


Figure 5 IMA Integration stages [10]

Where:

- Integration Stage 1: integration of components/modules to form a platform.
- Integration Stage 2: integration of a single application with a platform.
- Integration Stage 3: integration of multiple applications with a platform
- Integration Stage 4: integration of multiple platforms into an IMA system.
- Integration Stage 5: integration of IMA system(s) onto the aircraft (aircraft-level integration).

The integration stages vary depending upon the IMA system under development. However, the critical aspect of this integration is that, within and between the stages, the commitments and compliance credits between modules, components, and applications should be effectively identified, controlled, and communicated between all associated roles. This will ensure the IMA system attribute of completeness as well as others previously mentioned.

Integration Stage 1—Integration of system's core software module or components.

Integration Stage 1 combines the system's core software module or components (RTOS, BSP, etc.) with the hardware module or components to form an IMA platform. This is the lowest level of integration that principally involves the platform or module supplier and the RTOS supplier.

The platform and module supplier makes resources (e.g., hardware) available to support the applications that operate on an IMA system. These resources include a processor, possibly co-processors, and associated memory, timer resources, and I/O devices. The hardware platform can range from a customized target board, or set of boards, in a system to a set of composable parts that may already have some level of qualification

pedigree or service history compliance credit.

Commitments and compliance credit development should be planned, controlled, and traced to the final delivered IMA system.

Requirements-based testing results should be assessed for completeness, and compliance credit for test coverage of other objectives should be verified.

Compliance with module requirements, resource requirements, restrictions, assumptions, etc., is demonstrated.

Verification of the Integration Stage 1 system should be planned by an Integration Stage 1 specific verification plan that will thoroughly exercise the IMA platform, including testing the core software services, module resources, interfaces, communications, robust partitioning, health monitoring, and other platform-provided services. A previous study, detailed methods for verifying the robustness of an RTOS in an IMA system. If done properly, this activity can result in an acceptance-ready platform that has documented capabilities and compliance credits. Types of data needed at the conclusion of these activities include a set of platform commitments that may impose systems constraints with respect to safety, function, architecture, behavior, and performance.

Integration Stage 2—Integration of a Single Application and Platform.

Integration Stage 2 is the inclusion of an application with the IMA platform. This stage integrates an application with the platform to demonstrate how this single application will meet its functional requirements. Several IMA system development approaches taken include executing Integration Stage 2 with multiple sets of IMA platforms with single applications, testing in parallel to improve IMA system development time.

Verification of an Integration Stage 2 platform should be planned by an Integration Stage 2 specific verification plan that will thoroughly exercise the application's use of the IMA platform and any associated constraints.

Integration Stage 2 tests build on the completed Integration Stage 1 tests, and may perform coverage analyses of the application and its interactions with the platform. This stage should also conduct white-box tests. These are test scenarios that can be used to verify the platform's behavior under unconventional situations such as robustness testing and fault responses (abnormal operating states or modes).

Integration Stage 3—Integration of Multiple Applications.

Integration Stage 3 activity normally commences after Integration Stages 1 and 2 are complete. An additional application, or set of additional applications, is integrated. The IMA system integrator may choose to integrate applications incrementally or group related applications in an orderly integration. This will permit the IMA system integrator to identify and isolate problems as the applications are added.
The objective of this integration activity is to verify that multiple applications on a single target platform processor operate as intended without any unintended effects. Resource management and shared allocations can be tested and confirmed, as well as health monitoring, fault management, and other shared services. This integration activity builds on Integration Stages 1 and 2 and their associated activities and assumes that each separate application has been through Integration Stage 2 verification on the same target processor platform.

Verification of an Integration Stage 3 system should be planned by an Integration Stage 3-specific integration plan that will thoroughly exercise all applications in their use of the IMA platform, services, resources, assumptions, and associated constraints. Verification activities will include verifying that interactions between applications and partitions are appropriate and controlled. The correct use and accuracy of the data buses and I/O messages should be tested. Integration Stage 3 verification should check that the temporal specifications of message traffic are met for all applications. These analyses can have a major impact for the project and any associated compliance credits.

Integration Stages 4 and 5.

Integration Stage 4 is the integration of an IMA system that has two or more platforms of hosted applications whose hardware, RTOS, core software services, resources, and hosted application configurations could be different or the same for redundancy and reliability purposes.

Integration Stage 5 integrates the IMA system on the aircraft with other systems. Just as in Integration Stages

1, 2, and 3; Stages 4 & 5 should be verified by a stage-specific integration verification plan that will thoroughly exercise the requirements / commitments and help to validate the system's assumptions.

The IMA-Specific Integration of Components Phase

Many IMA systems are built using a phased or incremental approach, including off-the-shelf, general-purpose modules or components. Varied skills may be required for developing low-level components, such as the platform with its core hardware modules, core software services, and RTOS; and high-level modules, such as the hosted applications.

It is important to note that not all the roles previously stated may not include skills, such as mechanical or electrical. Clear lines of communications and coordination between all roles, organizations, and development domains should be established in the integration plans and integration verification plans and procedures.

The IMA-Specific Integral Phase—Verification

Without well executed planning and coordination during development, verifying the IMA system may be extremely difficult. Verification in the form of reviews, analyses, and tests should be planned and conducted incrementally, such that development problems are easier to identify and correct early in the program. However, additional effort will be required to verify other aspects of the IMA system, e.g., an IMA system vulnerability analysis and the verification of assumptions, commitments, and constraints.

The IMA-Specific Integral Phase—Problem Traceability

In general, in IMA system development, proper identification of the root cause to problems is necessary. Problem identification and resolution should be well planned, since it will involve all role players at times, both during development and after deployment of the IMA system. These activities may even lead to alternate business arrangements between the role players. For example, integration of an off-the-shelf RTOS could require a separate maintenance contract during IMA system operation, or a problem arising in the IMA system while in-service may require an integrated team of role players from the different organizations to effectively trace and identify the problem source, even though that role player's component may not be the cause.

In general, arrangements between role players should be established to ensure resources are available to address issues as they occur in both systems development and operation. If all roles are performed within a single company, then it is likely that a single problem reporting, tracking, and resolution system will be in place. However, in an IMA system where many companies are involved, it is likely that there will be many different problem reporting systems. Agreements must be reached to enable the system integrator to track problems reported against the platform, and for the application developers to track problems related to the platform that affect the applications.

The IMA-Specific Integral Phase—Configuration Management

The coordination of configuration data is aligned with the coordination of traceability. Configuration management (CM) includes developmental CM and production CM. Each platform, module, and application supplier will have CM systems and each of these roles must baseline what they produce and what they deliver. Every role that receives modules, components, and applications must, in turn, baseline what they receive and what is integrated into their specific deliverable.

Commitments and Credits of the Phases

As discussed in this section, when each component is developed, it goes through the typical development phases of planning, requirements, design, etc. As each phase is completed, it may have an impact on a commitment or a compliance credit of that component as it relates to the rest of the IMA system. As such, the development of commitments and compliance credits will change throughout the entire project. This commitment and compliance credit development should be planned, controlled, and traced to the final delivered IMA system.

2.1.3.3 Actors in the IMA ecosystem

Hereunder are listed the major actors involved in supplying, integrating, verifying, validating, accepting, certifying the system.

Platform and Module Supplier.

The platform and module supplier may be separate organizations or business entities. The platform supplier provides the processing hardware resources with the core software. A module supplier provides a component, or collection of components, to the IMA system that can be separate from the platform supplier. The platform supplier should ensure that all shared hardware and software elements and resources for the platform meet the IMA system's most severe levels of failure condition classification to ensure meeting safety, integrity, and availability requirements. This implies that the associated software levels of the software applications, design assurance levels of the hardware modules, and applications to be hosted are known or assumed. The platform supplier has a focused responsibility for system safety with respect to the IMA platform. This includes the assessment of IMA hardware components (e.g., network equipment, computer resources, I/O devices, etc.) and IMA system supporting software (e.g., operating systems, core services, etc.), but not software applications that execute on the IMA system to perform a specific aircraft function. The application supplier focuses on the system safety as it relates to the functions of their application.

The platform supplier will need to undertake detailed platform safety assessment activities. To assess potential failure modes and effects, these analyses will need to examine IMA-specific features that include, but are not limited to, robust partitioning, health monitoring, communication, fault management, and shared data and resources (resource management).

The IMA platform and module supplier has other responsibilities. Until the IMA platform becomes an IMA system by integrating the platforms and loading applications, it has no hazards associated with its software at the aircraft functional level. However, there are failure modes associated with the physical IMA modules and components that need to be addressed, such as overheating or power drains. Assessing the severity of the consequences of these failure modes is difficult without knowledge of the configuration of the IMA platform on the aircraft or details about its installation environment. Nevertheless, potential failure modes can be identified, and resultant behaviors of the IMA platform can be derived using traditional safety assessment techniques, and these can be assessed against potential IMA system configurations and environments. Software can also contribute to these vulnerabilities and the platform, module, and RTOS suppliers need to document limitations, response to failure modes, and associated protection mechanisms, such that a similar safety assessment can be conducted with consideration of the software attributes.

RTOS Supplier.

The RTOS supplier is responsible for the protection of critical shared resources of the IMA platform and system, such as memory, throughput and schedules, I/O devices and protocols, and other shared resources.

The RTOS supplier is responsible for working closely with the platform and module suppliers to specify the hardware feature commitments from the IMA platform to the RTOS for proper development and integration of the RTOS with the platform.

APPLICATION supplier.

The application supplier develops the application to be hosted on a platform module or multiple platform modules, such as a flight control function or fuel management function. Application development should be within the commitments conveyed by the modules of the system via their role players. However, typically the application is developed without consideration of other application functions unless those functions are related, dependent, or interact frequently with other applications.

The IMA SYSTEM INTEGRATOR.

The IMA system integrator performs the activities necessary to deliver one or more system functions. The system comprises the platform (hardware and core software), resources, services, modules, and a specified set and configuration of hosted applications. The IMA system integrator has an obligation to convey system

safety assessment information to the IMA system safety assessment (SSA) process. The IMA system integrator addresses all interfaces to the IMA system, including those from other aircraft systems and data buses. This includes the system configuration of the mix of selected applications to be hosted, resource allocation, configuration tables, system integration, and overall performance of the system.

The IMA system integrator will typically be responsible for the initial system safety assessment processes, including the IMA system preliminary system safety assessment (PSSA) activities based on the aircraft functional hazard assessment.
The system integrator, likewise, will typically be responsible for the integration of the results of activities accommodating the system safety assessment.
The system integrator will also evaluate fault mitigation, protection mechanisms, and derived requirements to ensure consistency with aircraft safety, integrity, and reliability requirements. As part of the integration activities, the system integrator will ensure that the behavior and properties of the IMA system are consistent with the IMA and aircraft system safety requirements.
More specifically, the system integrator is responsible for ensuring that the platform is loaded with the appropriate configuration of applications, the agreed communication channels are established and function correctly, and the system is configured to provide the resources and services of the platform(s) and modules to each application that uses them. The system integrator must also be able to document or describe deactivated features or mechanisms that may be considered for future in-service reconfigurations.

Note that the system integrator may not have domain knowledge over the applications themselves.

The system integrator is responsible for ensuring that the applications have the agreed resources and services available, that they function correctly, and that the networks, data buses, and I/O devices provide each application with their appropriate inputs and outputs in accordance with the agreed interface specifications. The application supplier, prior to system integration and testing, should have independently verified the functionality of their individual application(s), and documented the compliance (full, partial, or no) with the appropriate software guidance, policy, and applicable agreements.

CERTIFICATION APPLICANT.

The certification applicant is responsible for demonstrating compliance to the applicable aviation regulations, and is seeking a Type Certificate (TC), Amended TC, Supplemental TC or Amended Supplemental TC. This role may be held by the aircraft manufacturer or the original equipment manufacturer and may need to depend on the activities of and data supplied by the IMA system integrator. The documented compliance evidence and commitment accommodation conducted by the IMA system integrator will need to be verified and provided or made available to the certification authorities by the certification applicant.

All associated accomplishment summaries are likewise reviewed and offered by the certification applicant to the certification authorities.

The certification applicant may need access to platform, module, or component developers during the compliance towards the certification process and, as such, agreements as to the level of effort needed by these suppliers should be established when seeking final certification authority approval.

CERTIFICATION AUTHORITY.

The certification authority is the organization(s) granting approval for the IMA system and the overall aircraft and/or engine certification.

Example of some questions or activities that may be needed at various stages of involvement for an IMA system: Are the following complete?

- Aircraft-level IMA System Accomplishment Summary
- IMA System Accomplishment Summary
- Module/Platform Acceptance Accomplishment Summary
- Accomplishment Summary of Reused Module
- Accomplishment Summary of Applications
- Installation Instructions

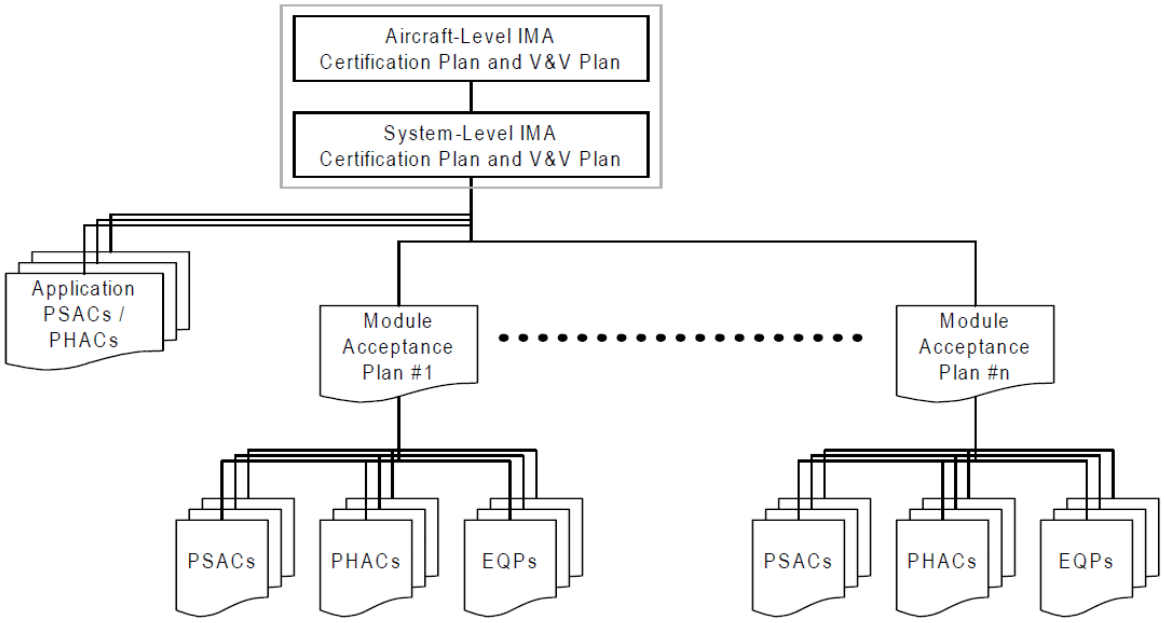
It is recommended that an IMA system Partnership for Safety plan be established. This plan would assist the certification authority, certification applicant, and IMA

2.1.3.4 IMA SYSTEM LIFE CYCLE PHASES, RESULTS, AND COMMITMENTS.

Each phase of IMA system development produces a set of deliverables with the goal that will result in the acceptance and approval of the final system. In developing that total set of deliverables, the responsible parties must address all the phases of the IMA system development life cycle. These phases include all typical phases of a federated system development life cycle, such as but not limited to, planning, aircraft safety assessment, system safety assessment, requirements, design, code, verification, validation, production, and maintenance. But additional life cycle phases are needed for effective IMA system-specific development.

The IMA-Specific Life Cycle—Plans for IVV activities

The IMA system certification plan and the associated module acceptance plans, PSACs, and PHACs are critical to the acceptance of the IMA system by the certification authorities.
As defined earlier, the IMA system will typically be constructed from platforms, modules, resources, core software services (RTOS), data buses, and applications. To reduce dependencies between these items, it is likely that an RTOS will be developed separately from its software life cycle data supporting compliance on a specific platform or specific applications. Generally, an RTOS supplier provides the operating system and supporting life cycle data to the platform or module developer. A plan should be in place to coordinate these provisions. Planning continues as each stage of integration is accomplished. As such, the IMA system ends up with a set of planning documents that typically will be organized hierarchically, as shown in figure bellow.



V&V = Verification and validation
EQP = Equipment qualification plan

Figure 6 Incremental plans [10]

The platform and module supplier may consider publishing a template Module Acceptance Plan that describes a virtual target environment as seen by an application, which includes the platform and module functions, services, resources, constraints, commitments, and compliance credits. It would also describe the compliance objectives that are satisfied (fully, partially, or not) by the platform supplier and each module supplier, and the objectives still left to be completed or satisfied by the application developer and the system integrator. Each application supplier could use a template PSAC to help with the completion of the application or module-specific PSAC. Proposals for compliance credit for development and verification sought during the development of the IMA system must be documented in the plans, with identification of the objectives and the means used to satisfy those objectives.

Plans are to be complete and should consider all aspects of the IMA system on itself, the aircraft, operators, and maintenance personnel. Such properties as safety features, protection, partitioning, fault management, health monitoring, environmental aspects, independence, isolation, installation, flight crew alerts, hardware design assurance levels, and software levels must be detailed in the plans, as well as identifying who will be responsible for the different aspects of each of these life cycle activities.

IMA-Specific Life Cycle Requirements.

Perhaps the most salient points with regards to requirements in an IMA system are requirements capture, traceability, and management. Requirements capture, traceability, and management are difficult challenges, especially for an IMA system where the various modules, components, and applications are being developed by different organizations, each of which has their own favorite methods and tools for specifying requirements, and which may not be compatible with one another. Hardware, software, and IMA system development guidance all impose requirement traceability objectives not only to the design and associated code or firmware, but also to the verification plans, procedures, and results. Consequently, simply tracing requirements in the IMA system can be extremely difficult, because the requirements must trace to each of the applications, and correspondingly, some application requirements may trace to the IMA platform and modules. Additionally, other requirements, such as RTOS and module and platform requirements, may be derived. Different levels of requirements must also be correlated and evaluated to confirm consistency and compatibility, a challenge for any module, component, or application developed in isolation from higher-level requirements.

A clear plan for IMA system requirement traceability is needed. It is recommended that a consistent methodology for traceability be applied for the IMA system and its associated modules, components, and applications and life cycle phases to declare traceability to be accurate and complete.

Integration of Components.

IMA integration is incremental (with different integration stages). Integrated components or modules include the RTOS, hardware modules, resources, services, platform, hardware and software applications, and others. The set of modules, components, and applications to be integrated may require other skills such as mechanical design, electrical design, software programmers, tool developers, verification teams, and others. Communication is essential among the key roles and their associates, and an effective integration communication plan is necessary to build a proper plan for the IMA system integration. This plan should consider incremental integration, configuration management of the modules, components and applications to be integrated, validation of assumptions, commitments, compliance credits, and communication between the IMA system development roles.

Verification.

As modules are developed and provided to the IMA system, the module acceptance plan must be completed and accepted. Documentation of the module commitments is necessary along with any compliance proposals, including fully or partially completed compliance objectives of the appropriate versions of ARP 4754, DO-178B, DO-254, or DO-297.

The IMA system approval, verification, and validation plans should direct the activities necessary to verify the IMA system. Verification is based on requirements. The IMA system requirements will be comprised of many requirement sets, which should correspond to the system, platform, modules, resources, applications, and components. Some requirements will trace to aircraft functions (typically for applications), and some requirements will be derived. The derived requirements will typically correspond to the components that are general in nature, for example, the RTOS and the platform core software functions. An essential part of verification is to ensure the completeness of the requirement's verification. Traceability between requirements is key in accomplishing this activity. The derived requirements are of critical importance, since, in many IMA systems, most derived requirements will result from architecture and design decisions made during the development of IMA systems. Verification of the identified hazards and their associated mitigations must also be completed.

IMA system-specific verification, unlike the federated system (LRU) verification approach, may involve incremental acceptance of verification evidence. Module, platform, and application versions will evolve, and their verification proceeds along with the development of the IMA system itself.

Configuration control of those items and their versions during development and testing is essential to document the pedigree of the modules, components, applications, and the source of compliance data. Should a new module, component, or application version be offered during development, analysis will need to be conducted

to determine the validity of previous compliance results or the need to reverify aspects for as the new modules, components, and applications acceptance data (e.g., change impact analysis and regression testing).
It is noted the importance of commitment data, that is, the assumptions, limitations, configuration, or any other commitments of previously accepted modules, components, and applications. With incremental acceptance, it must be verified that all commitments have been accommodated. An IMA system commitment document should be considered as a means of identifying all commitments. This document may not have the commitments specifically listed, but it should be the place where all the commitments can be identified or where references to other data are provided.
Once the commitments are confirmed, there should also be a confirmation that the shared resources are properly protected. This should be conducted in each domain of space, time, I/O, communications, and any other shared resource used. Additionally, the IMA system needs to be verified for its various modes of operation, e.g., initialization, start-up, normal operation, degraded operation, reversion to backup functions, shutdown, health monitoring, fault management and recovery, to name a few.

At this point in the IMA system development, confirmation of the completion of all planning and development objectives, and all life cycle (compliance) data are complete for all IMA system modules, components, applications, and the system as a whole. All partially completed objectives with respect to verification must be rolled-up and confirmed as being completed.

Problem Traceability.
Modules, platforms, resources, and applications will have varying dependencies with each other and may have some dependencies with systems or applications external to the IMA system.
With proper traceability, dependencies can be demonstrated, and changes can be made so that the ripple effects of those changes can be identified in other modules and applications. General rules can be defined based on the partitioning design, whereby changes within the application should only affect the application, whereas changes in the scheduler or RTOS can affect all applications. The impact of these changes requires a change management process that is coordinated among all the IMA system development roles. A coordinated change impact analysis should be developed by the certification applicant and integrator that can provide the scope of change, the reintegration approach, the verification and validation activities, and the effect on all resources and life cycle data. The responsibilities of the IMA system development roles are to be included. Included in the change management process and analyses are the considerations of change impact to the aircraft safety assessment, the IMA system safety assessment, and continued airworthiness.
The importance of traceability has been discussed, yet as an IMA system is developed and deployed, traceability is needed for problem identification and isolation. An essential IMA system development consideration is the proper identification of the root cause to problems. Simply because a set of modules and applications have been incrementally integrated into an IMA system, this should not lead the certification applicant to believe that problem isolation will be trivial. Dependencies exist between these modules, components, and applications, particularly in the form of commitments or assumptions that may prevent an effective evaluation of the root cause of problems that occur during development or in the field. For example, the platform or module supplier will document the problems raised against the module, and the application supplier must assess if this problem will affect the application development or present any new commitments.

Problem identification, isolation, and resolution, as well as role responsibilities should be part of the overall IMA System certification plan.

Configuration Management.
The coordination of configuration data is similar to the coordination of traceability. Each application supplier will have a CM system. The platform supplier, module supplier... will also have CM systems and, in receipt of these products, the system integrator will have yet another CM system.
Each role player must baseline what they develop and what they deliver. Every role that receives modules, components, and applications must, in turn, baseline what they receive and what is integrated into their specific configuration. Establishment and coordination of an IMA system master baseline of all these modules, components, and applications should be the responsibility of the system integrator. In any complex development, changes to modules, components, and applications will occur as system development proceeds. Some developers have a formal CM process that is separate from their developmental CM process. Formal CM typically controls the parts necessary for production fabrication of the system. Developmental CM is the process applied to control parts of the system as they undergo change during their development. CM in an IMA system is more challenging, since while the system is developed, some parts of the system may be under formal CM while others are under developmental CM. Effective CM planning should include control of both formal and developmental items during development.

The delivered modules, components, and applications such as platforms, resources, code, compliance data, and configuration files are managed under these CM systems. Associated verification evidence is also managed under a CM system. For example, an application developer may not ship their compliance evidence to the system integrator but instead will keep it on file for the certification authority. Agreements should be made between participants, and part of the overall CM plan should include which parties are responsible for controlling what items, and how problems associated with those items are communicated through the roles to determine the effect of these problems on other IMA modules, applications, or components.
A change management process should be developed among all IMA system development roles that detail the change process and show how the coordination of these changes are conducted amongst the IMA system development roles and organizations.

Results of the Phases.
Each integration-specific life cycle phase produces results that will support the integration of modules, components, and applications. Mapping these modules, components, and applications to the various phases of development can be quite difficult. Component development and IMA system development requires an integrated approach to development that will require role players to participate during all phases of development and integration stages. The commitments and information flow between these modules, components, applications, and their associated role players will also change as the IMA system is developed. As such, the plans discussed previously need to clearly specify what will be done, how it will be done, who will have responsibility, and how commitment and flow changes will be handled at the various phases of development. This is due to possible changes from feedback of subsequent integration stages.

Commitments and Compliance Credits of the Phases.
As discussed above, when each module, component, or application is developed, it goes through typical life cycle development, such as the phases of planning, requirements, and design. As each phase is performed, they may have an impact on a commitment or a compliance credit of that platform, module, or application to the rest of the IMA system. As such, the development of commitments and compliance credits will go through life cycle changes throughout the entire project. This commitment and compliance credit development must be planned, controlled, and traced to the final delivered IMA system. In particular, since multiple roles will be involved in the acceptance and approval efforts, the plans should clearly delineate who is responsible for satisfying each requirement and objective and who will support the completion of all requirements and partial objectives. The categories of full, partial, or none should apply for each objective in the IMA system compliance matrix with responsible roles identified.
For any accepted module, platform, resource or application, its failure conditions, limitations, assumptions, architecture, safety requirements, and required capabilities should be identified. Integration or installation considerations should be detailed as well.

The IMA System Approval Considerations.
The IMA system certification plan will provide the path for approval of the IMA system. It should convey compliance credits being claimed for the modules, platforms, applications (full, partial, or none), and the associated activities for the users of approved modules to achieve compliance credit for full satisfaction of all requirements and objectives.
A common cause analysis (CCA) should be conducted in an incremental fashion. An effective traceability mechanism is key to this analysis. The basic analysis techniques do not change, but need to be conducted at different integration stages, including the module and platform Integration Stage 1, the application Integration Stages 2 and 3, the system integration stage, and at the aircraft system level. This analysis should include not only loss of common resources, but degradation of those resources as well.
Other considerations in the approval process may include establishment of a legal agreement between the module, platform, and applications suppliers that considers data ownership, continued airworthiness support, or how regulations will be met during the maintenance phase.

THE IMA SYSTEM INTEGRATION PRACTICES.
Various IMA system integration practices may be employed for the development and verification of an IMA system. There are a variety of approaches that can be taken in an IMA system development, and what is best for one developer may not be adequate for another. As such, this section is not best practices, but rather a survey of some of the practices that could be employed for IMA system development. The practices cited are simply observations of useful practices; there may be others. Practices continue to grow and will

correspondingly continue to change. The discussion in this section focuses on the activities of building an IMA system and will not provide details on the activities involved in determining a proper or suitable IMA system architecture or design, since this would be IMA system specific.

Configuration Management for the Integrator

Proper CM practices are critical to the development of the IMA system. Items for the integrator to consider include verifying the versions of the modules, components, and applications to be integrated, which include the hardware and platform(s), software part numbers (applications, loaders, and RTOS), associated IMA system modules, components and applications serial numbers, database files, and configuration and initialization data. In addition, the integrator must monitor the compatibility of the combination of IMA modules, resources, and applications.

As the IMA system project develops, each integrated module and application will develop a certain level of maturity. Some may be 100% complete and others may be partially complete. As the IMA system is being defined, the trial integration process should continue. The integrator must understand the maturity and limitations of the module or application being integrated and the effects of those limitations on the rest of the integrated system. The items to be installed in the final IMA system will come from a variety of sources, all of which must be properly configured and controlled. Mechanisms for delivery of these configured items and their proper version retrieval from a variety of configuration management systems must be defined.

and some are field-loadable on the aircraft using an onboard or portable data loader. Some IMA systems offer more flexibility. An IMA system may be preloaded with an RTOS, or BSP software in nonvolatile memory, together with executable images of all applications and their data. When the IMA system is started, the programs and preloaded data are copied into RAM and execution is initialized and started. The loading from nonvolatile memory to RAM may be performed by a loader that is part of the RTOS and BSP.

As in a federated system, an IMA system may provide the flexibility of changing one or more of the applications, or even the RTOS itself, without removing the unit from the aircraft. If the programs and data are stored in nonvolatile memory, then a mechanism for updating the nonvolatile memory must be provided. This is accomplished through a communication protocol, often based on ARINC 615a.

It is imperative that the configuration of all platforms, modules, core software, resources, applications, and RTOS components loaded on an aircraft is established and controlled to ensure the conformity of the aircraft to its approved type design. The configuration must identify the part numbers and version identifiers for each module, component, and application; identify acceptable replacement or alternative parts and versions; and identify allowable intermix combinations of different parts and versions. This configuration must be maintained by the system integrator and the certification applicant as part of a configuration index of loaded software for the aircraft type design. This configuration index should provide a means of verifying the actual loaded software against the aircraft type design and allowable alternative parts and versions. In a typical scenario, it may be common for one or more applications to be changed without changing all applications. The changes to the configuration index must reflect an accurate description of the IMA system and each of its modules, components, applications and part and version numbers.

TRIAL INTEGRATION.

Proper integration of a set of modules, components, and applications involves the use of a defined methodology. The system integrator can develop this methodology, but it will need acceptance by the other role players in the development of the IMA system. Interface requirements, commitments in time, space, or resource allocation from system components and modules, and their dependencies must be clearly understood.

Combining too many modules, components, and applications at one time will probably result in a less than optimal approach to the integration of the IMA system. Rather, a planned, incremental approach is needed, where various modules, components and applications at various stages of readiness can be combined to successfully determine the predicted integration of the final system. Developing this plan requires knowledge of the system itself, the associated modules, components and applications, interfaces, commitments, project schedule, and the state of capabilities with these modules, components, and applications. An effective integration plan can produce a set of trial integrations that, when worked in concert with the IMA system validation and verification plan, can result in gaining confidence that the incremental system and its assumptions can be accommodated. Moreover, it will provide a vehicle for determination of the readiness of the modules, components and applications, and confirm their expected state of operation.

MODELING.

The term “model” can assume several different notions. This Handbook breaks model down into two distinct types, system models and activity models. System models are defined as those models that simulate system

behavioral aspects, such as communication timing or memory partitioning protection. Activity models are process models that assist in understanding that process. Both models have merit in the development of IMA systems.
System modeling is a very good practice for helping to understand how various modules, components, and applications of a system operate and interact. In developing any highly integrated system, up-front modeling can identify gaps or capability deficiencies for final IMA system integration. The model could be a simple set/use table, a worst-case execution model, or a more complex model, such as a communications model. The type and level of detail in the model is up to the developers and integrators.
An activity model is a good vehicle for establishing and acknowledging commitments and assumptions. An integration process model can provide a defined means of effective system integration. This should include abstract, but proper, modeling of relevant architecture aspects.

Modeling—Integration.
The act of generating the IMA system is difficult. The IMA system integrator must have documented and controlled configurations of modules, platforms, RTOSs, and resources, and for each hosted application, documentation on the commitments, assumptions, and configurations for the operational system and environment. The scope of the integration activity can be quite large and rather overwhelming. This process must be carefully designed and defined and modeling the integration activity should be considered. As part of the integration modeling effort, the integrator should also consider incremental and trial integrations to alleviate end-of-project scheduling commitments. Several certification applicants interviewed for this study confirmed that the results of a complex integration effort could provide real project showstoppers that could result in a significant delay or even project dismissal. Early or incremental integration of the system's modules, components, and applications permits an effective means of IMA system delivery.

Modeling—Traceability.
The traceability model is one of the simplest models to use and can be used at various levels of abstraction. Traceability provides a way of determining if something produced is actually used and vice-versa. Traceability can be used to determine the affected areas of change from a baseline, thus inducing a scope for verification activities on those changes. The technique can be used as low as the code level and as high as the plan and requirement level. Any producer of a commitment or assumption must have an associated consumer that uses, or is aware of, the commitment and any associated assumptions. Tracing the responsibility for commitments, assumptions, and other configurations is essential for effective IMA system development.

Modeling— Activity Plans.
The development of any IMA system requires an effective set of plans. These plans detail the process, tools, assets, and roles for the IMA system development. In order for the IMA system certification applicant to ensure that the roles and responsibilities of each participant in an IMA system are defined and acknowledged by that role player, an activity model of the IMA system development process could be developed. Although one may choose to develop a computer model of the plans, analysis is probably the most common approach. This analysis should be conducted just prior to the final IMA system certification plan review before submitting the plans to the certification authorities. Actions to be taken for this activity include, but are not limited to, the following:
Ensure coverage of IMA system development and integration objectives with clearly defined roles and responsibilities, especially for partial satisfaction of compliance objectives.
Analysis of the outputs for each stage of integration to confirm the role player understands their deliverable and confirm that a contract of some type to produce that deliverable is in place.
Additional actions include determining the adequacy of tools to perform the intended functions and tool assessment and qualification, if required.

The list of potential actions for the plan modeling activity can range from the very simple to the very detailed, with the ultimate goal of determining the preparedness of the process, tools, assets, and roles defined.

INCREMENTAL ACCEPTANCE AND COMPOSABILITY.

An IMA system permits many functions on the aircraft with the shared use of computing and system resources. Given the integration stages previously noted, the notion of building and accepting modules is the basis of

developing a composable system. Composability is the capability to select and assemble system modules, components, and applications in various combinations into meaningful systems that satisfy specific user requirements. How they are composed can make for an easy or complex integrated acceptance method of IMA systems. What needs to be resolved is determining the level of verification necessary when building the composable module and determining the level of verification necessary when integrating the composable module. These issues should be addressed with the certification authorities early in the development process. A vehicle for this would be the IMA Partnership for Safety Plan document.

DO-297 permits the use of incremental acceptance. As noted, the acceptance data sheet may have commitments that must be observed to permit the reuse of the module in the IMA system. Currently, the module acceptance approaches per DO-297 have not been detailed by the FAA; however, one can safely assume it would be based on AC 20-148. This may mean that for DO-297, as a minimum, the objectives must be referenced with details on the amount of compliance credit being sought (full, partial, or no credit), the assumptions, the means of compliance, and the remaining activities the integrator or certification applicant must complete. However, other questions are still unanswered: Is incremental acceptance of modules adequate without supplier validation of the integrated interfaces?

IMA INTEGRATION TOOLS.
Tools are used to help develop and verify the resultant IMA system and are a necessary part of any complex system development. Although tool use is encouraged, and almost mandatory in some cases, the tool's output may require qualification of some type to ensure the tool's accuracy and associated dependencies on that tool. Therefore, the latest version of DO-178B and DO-254 should be followed.
Classifying tools is rather problematic since the domain varies and overlaps. Some tools are specific to a particular aspect of the target system (e.g., executable object code-compiler/linker); some encompass an abstraction of the behavior, communications, or process (model); some align shared resources and confirm system limitations (target hardware/software integration); and yet some can be used to simply verify attributes of the system, like consistency or completeness (set/use).

Hereunder details on several tools used specifically for aiding the effort to develop or verify an IMA system. Tools developed to support the IMA system in its embedded environment are also discussed.

- Traceability Tools
- Configuration Management Tools
- Data and Control Coupling Tools
- Resource Management and Architecture Analysis Design Tools
- Communication Modeling Tools
- Temporal Modeling Tools
- Other Tool Types to Consider:
    - Debugging tools
    - Memory image construction tools
    - Regression testing analysis tools
    - Field-loadable tools

2.1.3.5 PLANS

The following plans as per [10] should be considered:

1. Module/Platform Level:
    a. Acceptance Plan
    b. Configuration management (CM)/Software Quality Assurance (SQA) Plan, and associated plans
    c. User's Guide real-time operating system (RTOS)
2. Hosted Applications:
    a. Plan for software aspects of certification (PSAC)/plan for hardware aspects of certification (PHAC)
    b. CM/SQA Plan, and associated plans

3. IMA System:

a. IMA Partnership for Safety Plan

b. Certification Plan

c. Verification and Validation (V&V) Plan

d. CM Plan

e. SQA Plan

f. Environmental Test Plan

4. Aircraft-level IMA System:

a. Certification Plan

5. Environmental Test Plans

2.1.3.6 INCREMENTAL INTEGRATION AND VERIFICATION.

As per [10], following are the recommendations and questions for a DO-297-based Job Aid:

- Is the verification of the IMA system modules, components, and applications incremental?
- Can it be demonstrated that effective CM of modules, components, and applications for various incremental baselines is established?
- Is the information needed by module users to integrate and interface the module available and being applied?
- Is the application(s) integrated on the platform?
- Is the proper use of resources, as allocated to the application by the integrator, verified?
- Has compliance been demonstrated for intended functionality, performance, and safety requirements, using laboratory, ground, and/or flight tests, and appropriate analyses?

2.1.3.7 VERIFICATION RESULTS.

Again, as a Job Aid, check that the following compliance data exist:

- Platform Integration, V&V Data
- Module/Platform Acceptance Data Sheet
- Module/Platform Quality Assurance Records
- Module/Platform CM Records
- Module/Platform Problem Reports
- Hosted Application Life Cycle Data
- Complete Hosted Application Life Cycle Data Package
- Tool Qualification Data
- IMA System V&V Data
- IMA System Problem Reports
- IMA System V&V Plan
- IMA System V&V Records / Results
- Verification Results of Subsequent Installation
- Associated vulnerability testing on shared resources memory, time, I/O, communications.
- Is IMA system resource management, fault tolerance and management, health monitoring, degraded modes, and reversion capabilities verified?
- Did the testing conducted cover initialization under various mode or load conditions?

2.1.3.8 CONFIGURATION MANAGEMENT

CM for IMA systems can become complex, because there are many configuration commitments to consider for each module, component, and application. Configuration control of versions is critical as the project is developed and nears delivery. Other considerations to assess include:

- Change impact analysis is in place and confirmed effective
- Module/Platform Configuration Index is designed and verified
- IMA System Configuration Index is designed and verified
- Aircraft-level IMA System Configuration Index is designed and verified
- IMA System CM records are in place

2.1.3.9 INTEGRAL PROCESS CONFIRMATION

Care should be taken to ensure the integral processes are properly in place.

Quality assurance can have a mixture of quality organizations for the various modules, components, and applications.

CM requires careful management and oversight to ensure proper version control as the IMA system approaches verification and deployment.

Integration aspects to cover all the commitments and objectives of DO-178B, DO-254, and DO-297 must be in place, under control, and verified for completeness.

Verification of the integrated, or partially integrated, IMA system must confirm all aspects of verification, including completeness.

The certification liaison for a previously accepted module or platform must be identified and confirm the module or platform acceptable for use in the present IMA system.

2.1.3.10 Tools

Verification and development tools are assessed and qualified, as needed.

Design and configuration tools are developed and ensured to the level of assurance required to support the IMA system.

Adequacy of traceability tools has been determined.

Configuration management of the tools should be in place and confirmed.

Potential tool types that may be used for IMA system development or verification include resource management tools, such as rate monotonic analysis and architecture analysis tools; modeling tools, such as communication, temporal, and memory image analysis tools; installation tools, such as target system installation and delivery, and field loadable component tools.

2.1.3.11 System change analysis

The changed module or application is reintegrated into the IMA system.

All necessary verification, validation, and integration activities (regression analysis and testing) are performed.

Usage domain analysis is performed to ensure that the module or application is being reused in the same way it was originally intended.

2.1.3.12 Accomplishment summary

As per [10], check that the following are complete:

- Aircraft-level IMA System Accomplishment Summary
- IMA System Accomplishment Summary
- Module/Platform Acceptance Accomplishment Summary
- Accomplishment Summary of Reused Module
- Accomplishment Summary of Applications
- Installation Instructions
- Continued airworthiness considerations of an IMA system are in place.

2.1.3.13 Conclusion

The integration of modules, components, and applications for IMA systems is extremely complex. Each module, component, or application can induce commitments that must be met by other modules, components, or applications. Management of these commitments is difficult and spans the roles of the certification authority, certification applicant, IMA system integrator, platform and module suppliers, RTOS supplier, application supplier, and the maintenance organization. Many times, the parts of an IMA system are developed by different sources or organizations with various and multiple roles. Each role plays an important part in the overall development and acceptance of an IMA system, yet special focus is appropriate to the roles of IMA system integrator, application supplier, and platform and module suppliers since this is where many commitments are captured. Roles and responsibilities should be clearly identified and defined, especially between various accepted modules and the IMA system integrator and IMA system certification applicant.

It was further revealed that configuration and integration of system modules, components, and applications could be unmanageable as the IMA system is developed. Each module or component can impose commitments on the integration in the form of assumptions, limitations, and configuration. These commitments need consideration not only during normal operation, but also for loading, start-up, initialization, alternate mode operation, and shutdown. Robust partitioning is difficult to verify. In particular, time partitioning was difficult, if not impossible, because proper assessment of worst-case execution time (WCET) for modules heavily using RTOS services could be inconclusive. Memory and input/output (I/O) partitioning was difficult as well, but more manageable if handled properly by the RTOS.

It is recommended that a plan for partnership with the certification authorities be considered. This would permit a vehicle for communicating the system needs and role responsibilities. It could identify special plans for verifying partitioning robustness and identification of a vulnerability analysis and verification activity.

2.1.4 Other interesting papers related to IVV in modular avionic systems

Many other documents are available regarding IMA, modular avionics, integration V&V of these system.

Hereunder are listed some documents that are from interest and that were consulted but not further reported in this present document:

- Tool Chain for Avionics Design, Development, Integration and Test (Martin Halle & Frank Thielecke, Institute of Aircraft Systems Engineering (FST), Hamburg University of Technology (TUHH)
- A Modelling Paradigm for Integrated Modular Avionics Design (from INRIA/IRISA)
- Hardware/Software Integration Testing for the new AIRBUS Aircraft families, Jan Peleska, Center for Computing Technologies TZI, University of Bremen, Germany

And many other IMA related articles.

2.2 Lessons from modular automotive

2.2.1 Automotive architectural principles & Autosar

Automotive industry widely uses ECU modular approach. This approach is based on AUTOSAR architecture and methodology. This section is widely based on [11].

AUTOSAR aims to standardize the software architecture of **E**lectronic **C**ontrol **U**nits (**ECUs**). AUTOSAR paves the way for innovative electronic systems that further improve performance, safety and security.

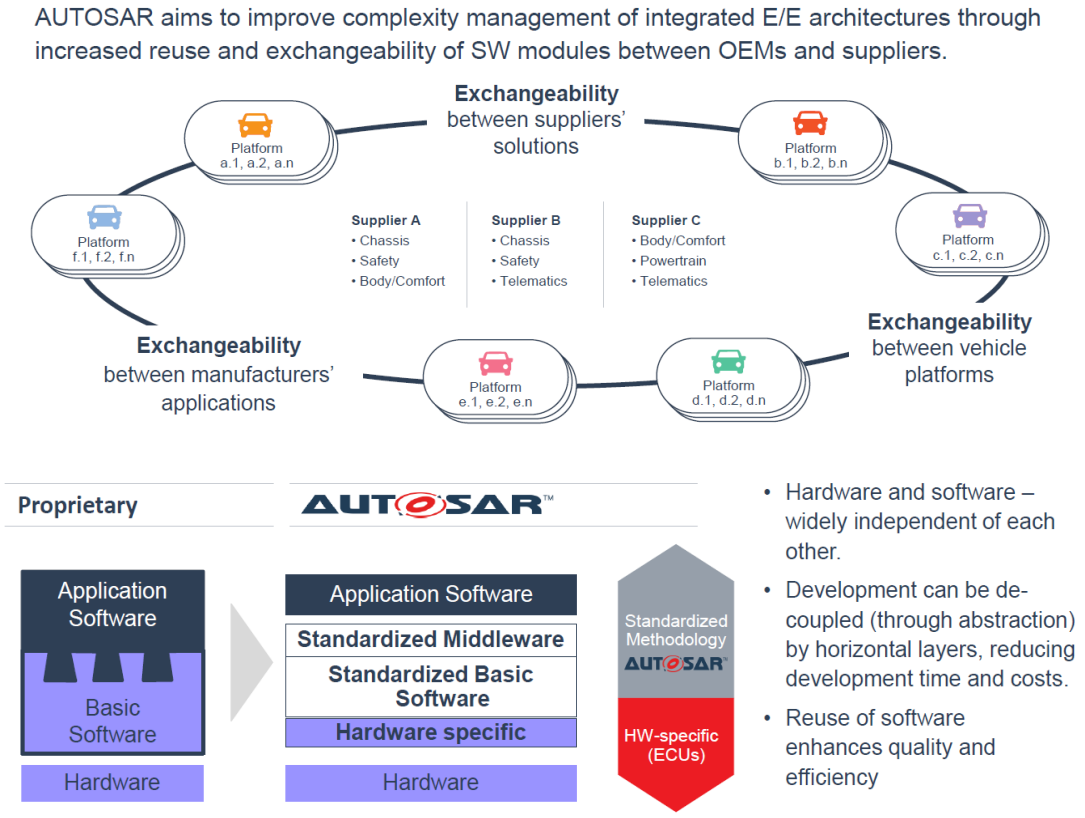
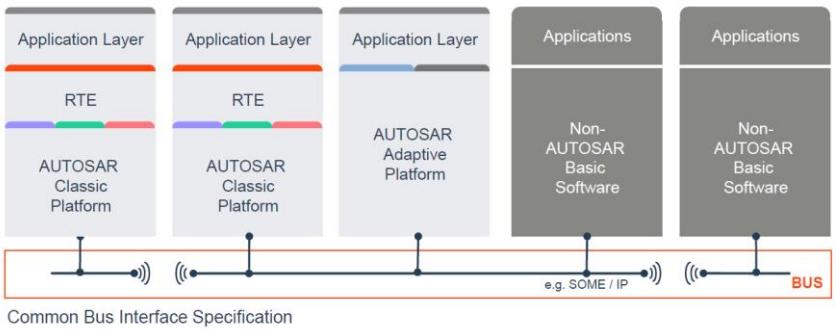


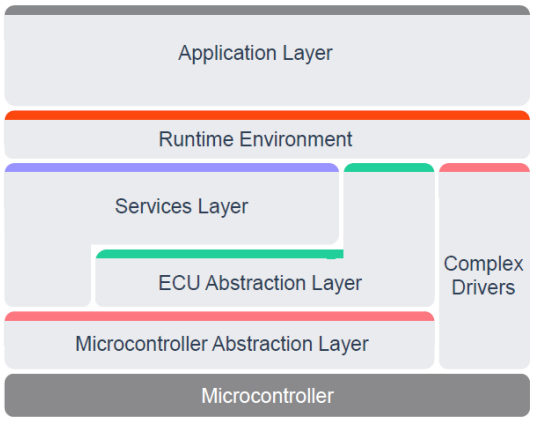
Figure 7 Autosar principles [11]

The corresponding architecture of a vehicle using this standard (mixing with non-Autosar devices):



(extracted from Autosar Introduction generic presentation for conferences & co - 27.10.2020) [11]

From a SW point of view, hereunder the layered approach is depicted :

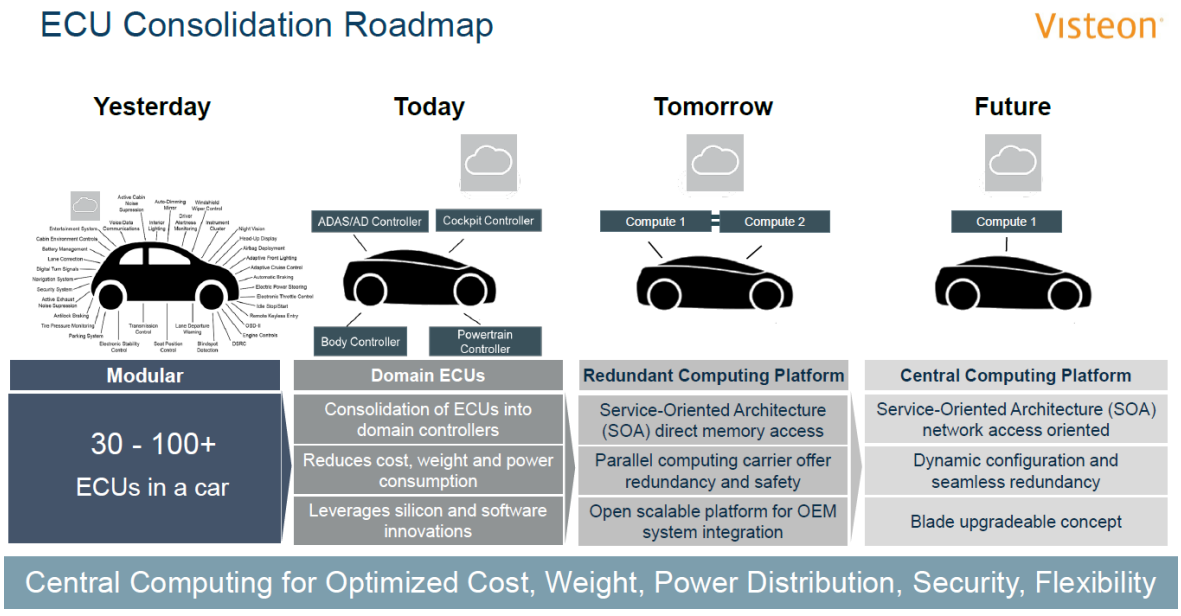


The layered architecture of the classic platform basically supports

- Hardware abstraction
- Scheduling of runnables and tasks (OS)
- Communication between applications on the same hardware and over the network
- Diagnosis and diagnostic services
- Safety- and
- Security Services

(extracted from Autosar Introduction generic presentation for conferences & co - 27.10.2020) [11]

From HW point of view, VISTEON forecasts the following evolution:

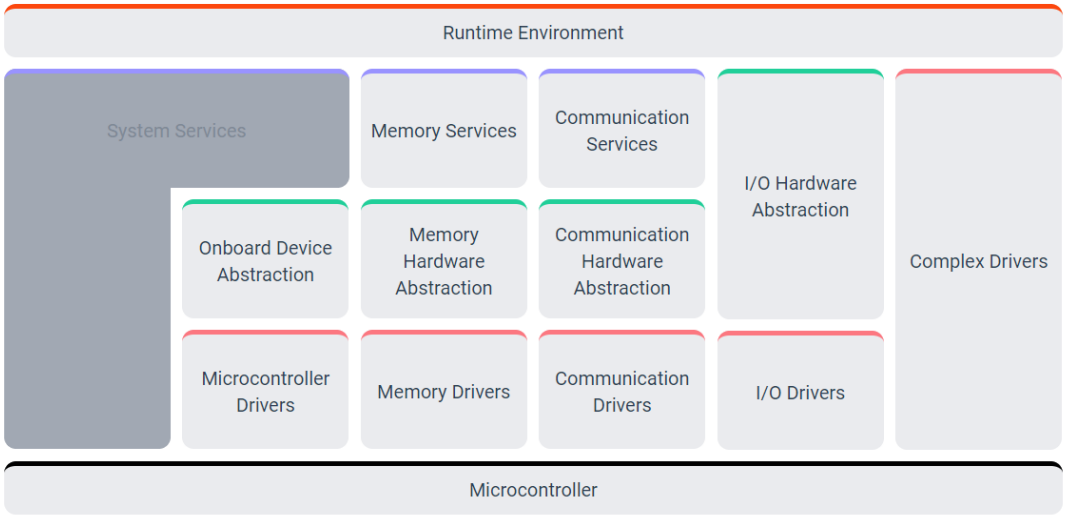


(extracted from VISTEON commercial presentation)

Note that both Classic AUTOSAR platform and Adaptive AUTOSAR platform exist. The first one is signal based and the latter is service based.

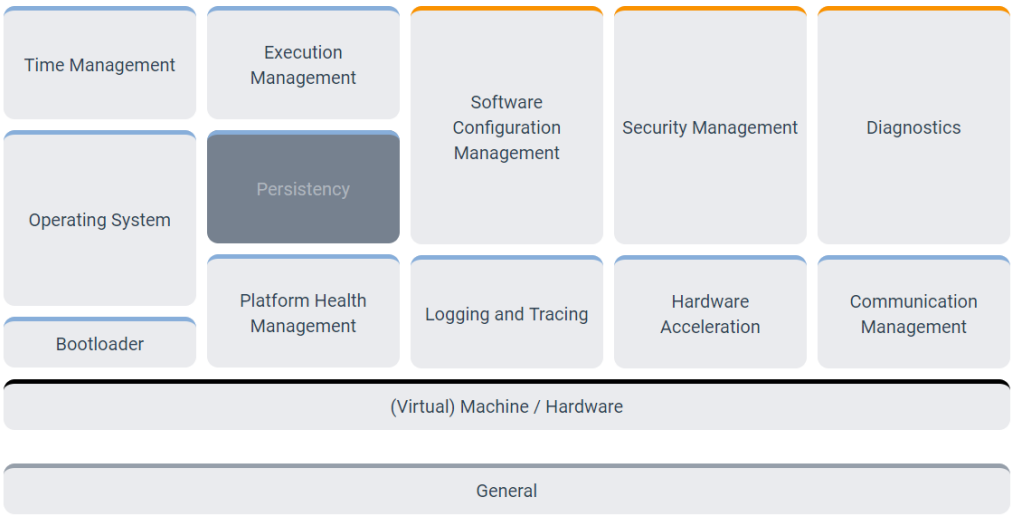
Classic AUTOSAR Architecture:

The figure below shows the architecture for Classic Platform. The Application Software (ASW) consisting of Software Components (SWC's) sits on top of Runtime Environment (RTE).



Adaptive AUTOSAR Architecture:

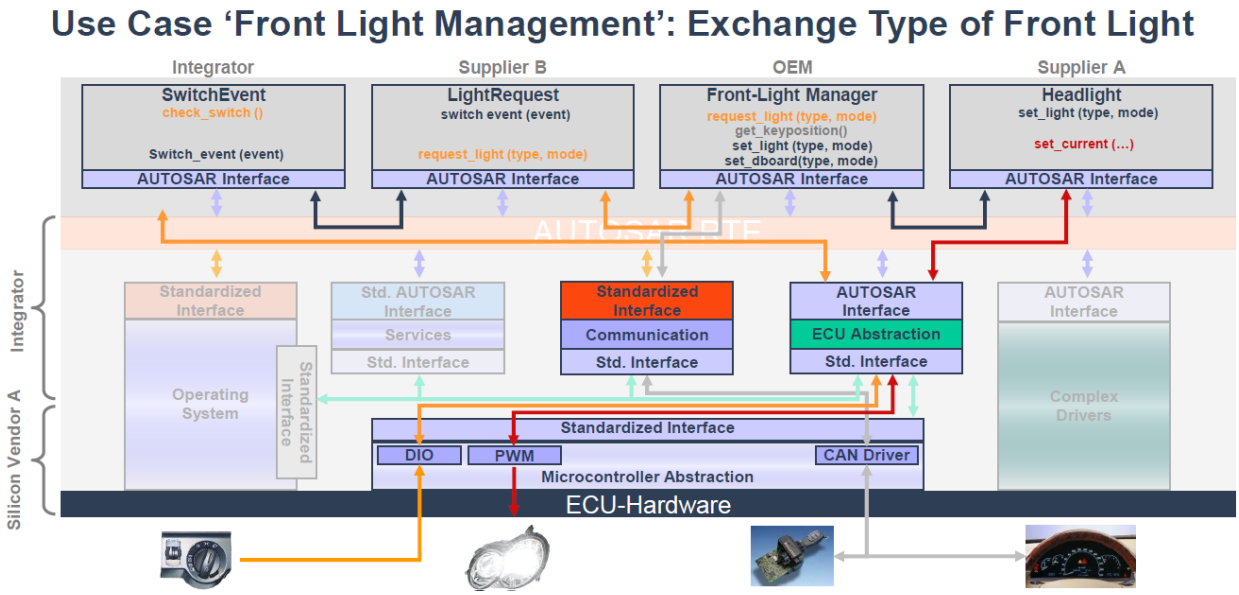
The figure below shows the architecture for Adaptive Platform. The top layer would consist of Adaptive applications.



⇨ Benefit from AUTOSAR Architecture(s) from "integration and testing" perspective

The main lesson learned is that modularity helps limiting testing effort in case of change / evolutions as expected on OCORA.

Hereunder a use case where front light provided by supplier A is replaced by a Xenon front light: thanks to interface principles of Autosar and its architectural principles, the exchange can be performed without impact on the other components/modules.



(extracted from Autosar Introduction generic presentation for conferences & co - 27.10.2020) [11]

2.2.2 Renault-Nissan IVV approach based on automotive architectural principles

OCORA met Automotive representative System Architect from RENAULT-NISSAN Group explaining IVV strategy. Our Speaker has over 20 years of experience, occupied different positions at Renault Nissan in vehicle software and electrical systems for various vehicle platforms.

Major feedback in a nutshell:

The design specifications are under OEM responsibility (either written requirements, or models: using Matlab-Simulink).

The different actors identified:

- Tier1/Tier2 SW supplier
- Tier1/Tier2 HW supplier
- Integrator
- OEM

The different validation steps:

- virtual (MIL) test to assess and freeze the design.
- ECU integration platform HIL/SIL (in 2 steps) – test environment is under OEM responsibility. Some supplier come to the platform for testing. The Platform is similar from one vehicle to another one. The HW/SW comes with different releases (loops)
- Vehicle integration (in 3 steps) from protype to serial types (for production tools freezing).

Both configurations are encountered: there is the possibility to integrate Autosar / Non-Autosar components.

A unique tool to track the bugs and corrections (JIRA). This tool is imposed to the Tier1 suppliers.

Most tricky problem encountered: the software maturity convergence and the number of versions.

Hereunder, more details delivered during our sessions with Renault-Nissan:

What is EE system in Automotive vehicle ?

- Electronic Control Unit ECU : HW + SW + calibration + configuration
- Sensors : active and passive sensors, driver interface sensors …
- Actuators : relays, motors …
- Wiring harness and connectors

Off-board components are not considered in vehicle EE system

- eCall
- Connected services

Inter-system communication

- CAN is main inter-system communication bus :
  - Each ECU has to respect CAN HW requirements
  - Usually TIER1 supplier buy CAN SW from VECTOR
  - CAN (frames, messages) is fully defined by OEM

→ High standardization Of CAN HW & SW

- Ethernet replaces CAN for some ECU (Multimedia, ADAS …)
  - CAN on Ethernet
  - Ethernet message set is also fully defined by OEM
- Other bus (restricted) : LIN, USB, LVDS video, Digital Audio

Inter system communication dev is under OEM responsibility

EE system development

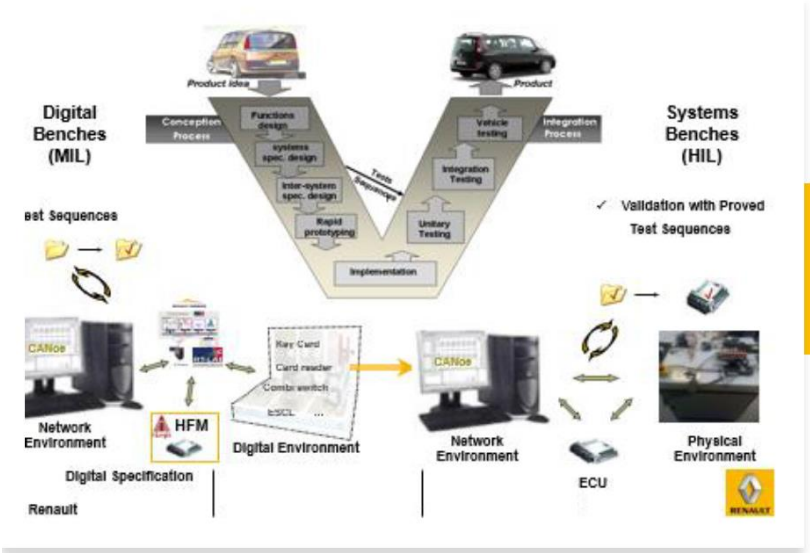
- Electronic Control Unit ECU
  - HW development by Tier1 supplier ➔compliant with OEM requirement
  - SW development 3 possibles ways :
    - Tier1 SW : AUTOSAR is standard but some ECU are not AUTOSAR dev
    - OEM SW (with AUTOSAR application)
    - In some case Tier2 SW is used (with AUTOSAR application)
  - Unit part validation by Tier1
    - Function
    - EMC
    - ENV + durability

Integration/Validation in V-cycle

- Virtual Platform
  - Some ECU models (Matlab/Simulink/Stateflow is mandatory)
  - Renault proprietary tool
- Electronic integration platform (2 steps)
  - All ECU, sensors, actuators, harness (prototypes of serial definition)
  - Integration of components in systems
  - Validation with automated scenarii (cumulative database of tests)
- Vehicle integration (3 steps)
  - 1st : vehicles assembled in proto workshop (serial definition)
  - 2nd and 3rd : vehicles assembled in production plant

Project milestones

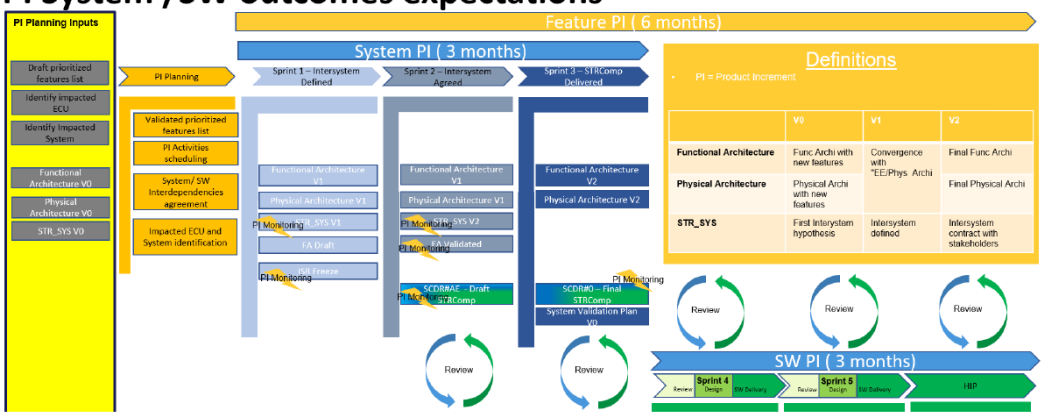
- Tech Def hypothesis
- Tech Def freeze
- Production tools freeze



Integration Validation in V-cycle

Customer value feature in AGILE mode

PI System /SW outcomes expectations



3 Experience sharing among NS, DB, SBB and SNCF

3.1 Methodology

Based on implementation cases, projects or other experiments, the different examples of IVV experiences within OCORA participants are shared with the following template:

Part 1: Brief introduction of the IVV experience: Context (project, research, other), why this was done, goal (purpose), actors involved…

Part 2: Short description of what was performed: what (goal, work done…), how (tools, methods..), who (actors, subcontractors, skills…), planning/duration…

Part 3: Lessons learned

- The “pros” : positive experiences, concepts/methods to reuse
- The “cons”: difficulties, drawbacks, problems encountered

Part 4: Recommendations for OCORA

3.2 Experience sharing SNCF: NEXTEO

3.2.1 Brief introduction of the IVV experience

This experience is based on EOLE E Line – NEXTEO project.

It is a CBTC project in Paris area (71 trains, 55 km of line extension, 8 km new tunnel, new stations)

Different providers for the CCS: Siemens, ATOS, Alstom, Thales

Project start: 2016 - migration for operation in 4 steps

Characteristics:

- High challenge for system integration and validation
- Modular system (IO, Adonem, STD, Radio…)
- different CCS NEXTEO subsystems
- Over 30 different equipment types
- Different SIL levels within the system: SIL0/SIL2/SIL4
- Over 20 external interfaces (PAI, PRCI, SI EF, SI GI, RS, BS, …)
- Some “internal” interfaces between different providers (e.g., ATS-CCK, ATS-MESSIL2)
- Highly generic and configurable

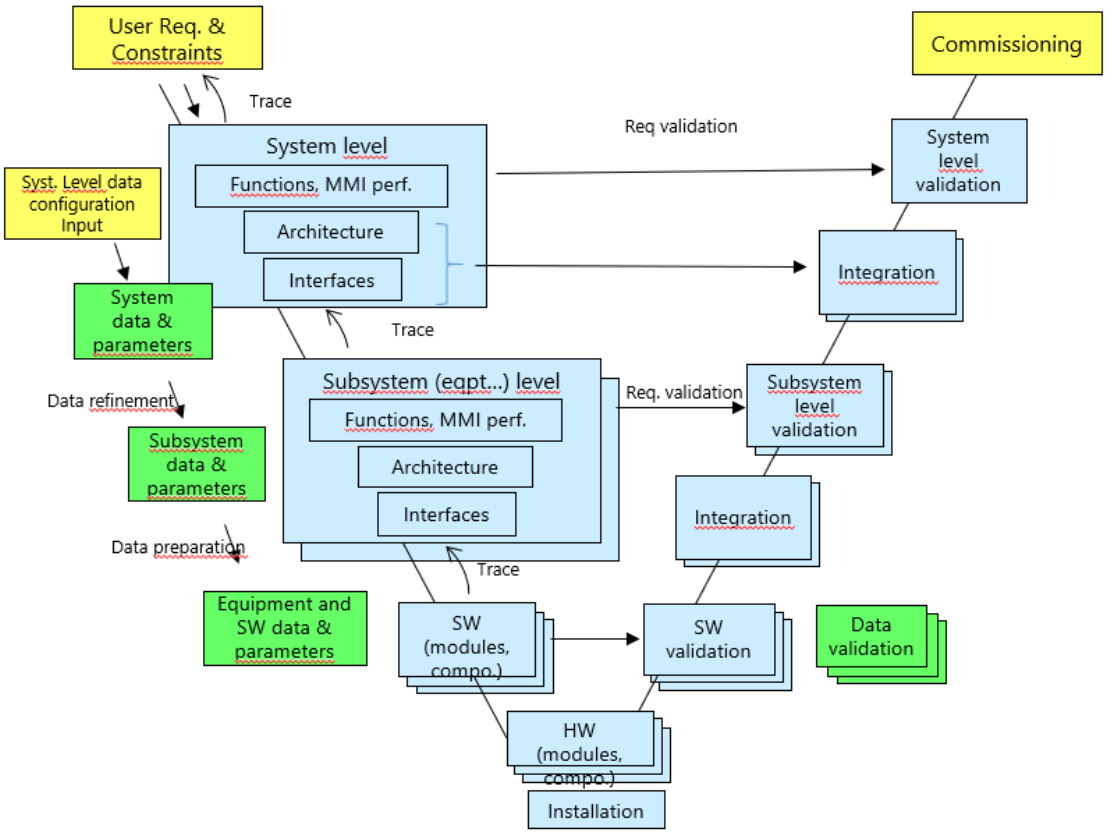
3.2.2 Short description of what was performed

Following tasks were performed:

- IVV Strategy fixed with the different stakeholders
- IVV documentation (test plan, integration plan…)
- “modelling” environment prepared (3 different suppliers) and integrated in the test benches (1 integrator) (track model + existing relay based interlockings (CATIA), CBI (SIMENV), EuroKVB Simulator)
- Factory testing and integration (including test bench) with first FAT
- “customer” 1st test bench provided (2019)
- On site testing for technical functions (wayside/on board communication, train localization)
- On site first integrations
- On site test track preparation and static integration
- Use of formal methods in addition to testing
- Use of semi-formal configuration validation tool chain

Stepwise integration of:

- different subsystems
- over 30 different equipment types
- SIL0/SIL2/SIL4
- over 20 external interfaces



3.2.3 Lessons learned

Positive experiences, concepts/methods recommended to be reused:

- Well-structured documentation breakdown for validation strategy down to test cases
- Well-structured Interface Integration Plans for each interface with clear steps (what/how/who…)
- Powerful test environment allowing configuration testing, degraded modes…
- Effort adapted to the step (do not test everything on test track)
- Early testing and integration
- Master requirements in a very challenging environment (4 different contracts, 4 migration phases with >5 system releases)
- One stakeholder is designated as the Integrator
- One tool for tracing anomalies/bugs (federated by the integrator) and coupled with the customer
- Configuration activities starting with a system phase common to all stakeholders (or nearby…)

Lessons learned from difficulties, drawbacks, problems encountered:

- Late validation strategy finalized (most of the testing activities started before overall test strategy finalized)
- Integration document difficult to converge between stakeholders
- Difficult to have common milestones (different speed for design, testing…) between stakeholders
- Product approach “harden” a flexible adaptation (wait for a new product release/planning, “not in the product” syndrome…)

- Configuration activities rapidly spread into different companies processes and tools and make consistency at risk
- Many actors increase the risk of gaps/discrepancies between subsystems (inconsistent design choices hardly discovered)
- Many tools (manufacturer dependent) make the audit and acceptance more difficult for the customer
- Many tools for bug tracing among the stakeholders with different response times to handle anomalies discovered by other
- Migration of the system induces many versions (replace a module) and/or adding new functions: increase the amount of IVV steps

3.2.4 Recommendations for OCORA

In short.

- Start with IVV strategy as early as possible
- Master requirements and releases
- Master a well-structured documentation breakdown for validation strategy down to test cases
- Master well-structured Interface Integration Plans for each interface with clear steps (what/how/who…) that are converged between stakeholders
- Identify clearly the milestones in the IVV process
- Build a powerful test environment allowing configuration testing, degraded modes…
- Adapt effort to each step
- Insert early testing steps
- Reduce the risk of having many actors /many tools by having a common environment (e.g. MBSE and its IVV) where specific tools are limited to stakeholders specific activities
- Do not rely on validation only for safety demonstration (use FM for instance)
- Master a reference test environment
- Have a system integrator (or system integrators but with an overall system integrator above)
- Have a clear process for bugs/anomaly tracing till corrections (among the different stakeholders) supported by ad hoc tools

3.3 Experience sharing SNCF: CIM Qualification of different on-board ATC

3.3.1 Brief introduction of the IVV experience

Qualification of on-board products since decades

- Integration and system tests
- On site and Test Bench activities

Experience before ERTMS deployment and after ERTMS deployment

Different stakeholders: Hitachi, Faiveley, SNCF, EPSF, Certifer, SIEMENS, ALSTOM…

3.3.2 Short description of what was performed

BEFORE ERTMS, until 2005-2010 (lot of on site tests and few in lab)

- Different components like: ATESS, KVB, TVM, BRS, Radio, Odo, LZB, ASFA, ATBL, ZUB, INTEGRA
- For most of them, there is a « SAM » documentation:
    - Functional, availability, safety, installation constraints, …
    - Use of these documents for system tests done by SNCF. Component by component.
- Online test for integration and final validation
- Complexity and interfaces under control
- Beginning of the use of very dedicated and simple test benches

AFTER ERTMS, after 2005-2010 (Fifty fifty between on site and in lab activities)

- No « SAM » for ERTMS but multiple ERTMS Subsets and specific national requirements / deviations regarding Subset
- Buy of new test Bench dedicated to each EVC (Hitachi, ALSTOM, ...) and provided by the manufacturer itself:
    - Need of hundreds/thousands of test cases
    - Need of a beginning of automation
- Integration between EVC and other components more difficult due to digital information interfaces
- Lot of SNCF test activities redundant with activities of the manufacturer because system is more complex and no confidence (or knowledge) in manufacturer tests activities or in impact analysis due to evolution/modification.
- Huge increase of External constraints to be « managed ».

Creation of LEF

- Need to have our own testbench, more automated, more modular and adaptable to every component, not only one
- IOP Tests subject is growing
- Software in the Loop and Hardware in the loop needed

Component integration more complex and needed to be done in lab

Now and the future: wish of 75 % in lab and 25% on site activities

Projects done in LEF:

- IOP Tests:
  - Connection to another Trackside Lab for Level 2 test with two real RBC
  - Reproduction of real ERTMS traffic line (with simulation of 10 RBC)
- Subset 076 test campaigns on:
  - Real EVC
  - Software EVC under development
- Integration of real antennas and modems to an EVC
- System validation campaign for SNCF projects / Components, not only EVC

3.3.3 Lessons learned

Satisfying	To be improved
➔Modular and automated test bench ➔IOP Tests are quite easy to handle ➔Software in the loop and Hardware in the loop ➔Use of standards like Subset 076 for tests catalog.	➔Need to reduce complexity of EVC. Modularity needed. But be careful not to transfer the complexity into a multitude of interfaces. ➔Need to better define role and responsability of test activities between Stakeholders ➔Need to access a central BUS to have not only « black box » but « grey box » testing for analyzing bugs/problems

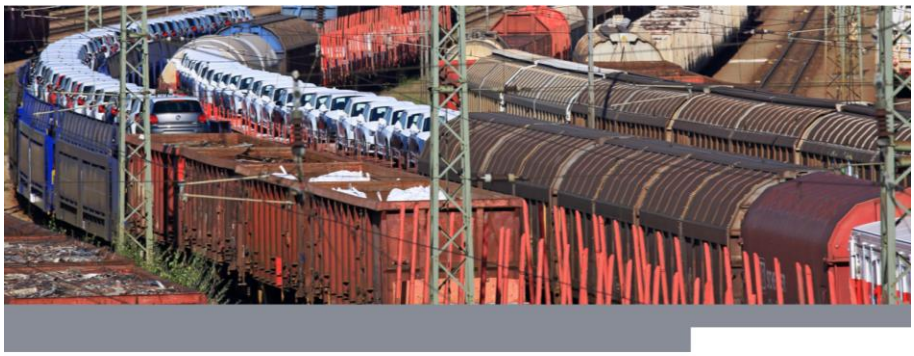
3.4 Experience sharing DB: Open ETCS

3.4.1 Brief introduction of the IVV experience

DB, as the most of the RU's, has not normally direct activity related to IVV in project execution since this is usually part of the supplier scope of work.

In relation to ETCS, the involvement of the RU's has changed direction due to the need to force the market to a modular design.

During the built of the OCORA experience was presented what was done from openETCS, also in terms of test strategy, and is now expected this work to be considered inside the OCORA activity as starting point for further developments.



Towards open CCS onboard
openETCS/OCORA – 2nd of April@SNCF

DB Cargo AG | Baseliyos Jacob | NS | Jos Holtzer

3.4.2 Short description of what was performed

Main idea in order to keep under lead the different interpretation of the subsets into a modular exchangeable solution was expressed by 3 steps:

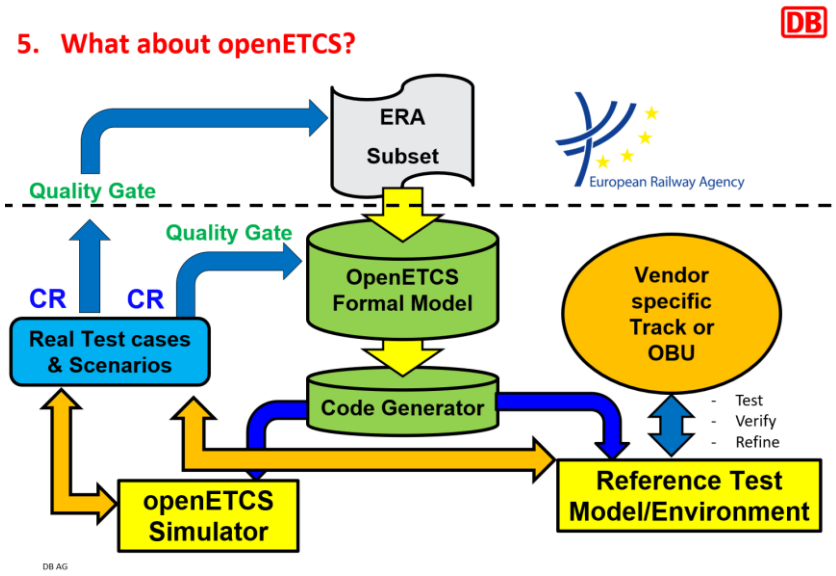
1) Move the different specification into a formal tool

2) Generate the exe code directly from the formal tool

3) Create a test environment based on the above to prove the industry solutions against it.

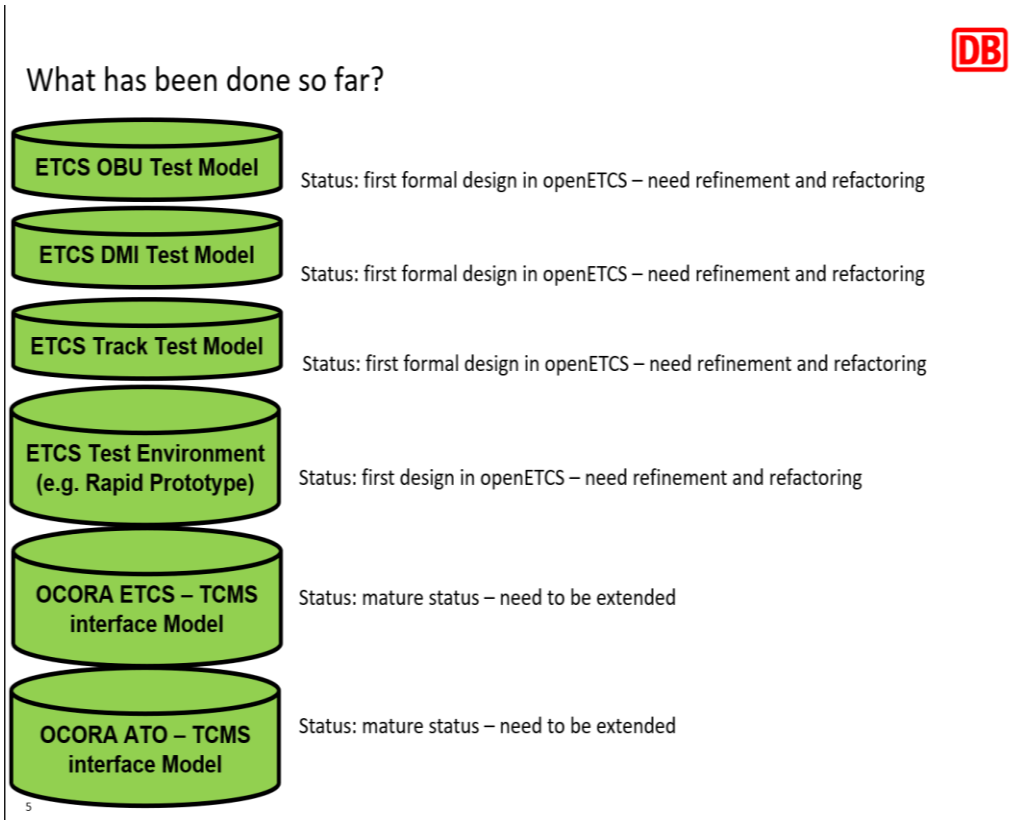
openETCS - to deliver a proof of concept for ...

- **Creating a formal specification for ETCS (onboard) kernel units acc. ERA TSI subset 026 to avoid ambiguities in the software**
  - to enhance the understanding of the subset
  - to be able to animate the model for testing and analysing purpose at system level
  - to provide information of the completneness and soundness of the SUBSET 026, Subset 035, …
  - to be used as a reference semi-formal specification for the implementation of an (on-board) kernel unit (by the OpenETCS project and by industrial actors)
- **Providing a tool chain and process/methodologies for developing an on-board software that can fullfil the CENELEC requirements for SIL 4 software**
  - The design process of the system and the associated tools of the tool chain shall be suitanble to provide a certifiable product.
  - The full safety process to make OpenETCS certifiable according to CENELEC shall be described in detail
  - The use of formal methods, supported by tools, is highly recommended in this safety process for specification, design, verification and validation of the certifiable product.
- **Provide an executable software package generated from the specification of on-board ETCS**
  - An executable software of this specification shall be provided, as well as a non vital implementation of the on-board unit for laboratory test, simulation and as reference. It will be a non-vital implemtenation, able to be executed in real- time and in interaction with other components.



Several test model (OBU, DMI, Track) including a test, environment to run them (PC based), can eventually be inherited and brought to final design.

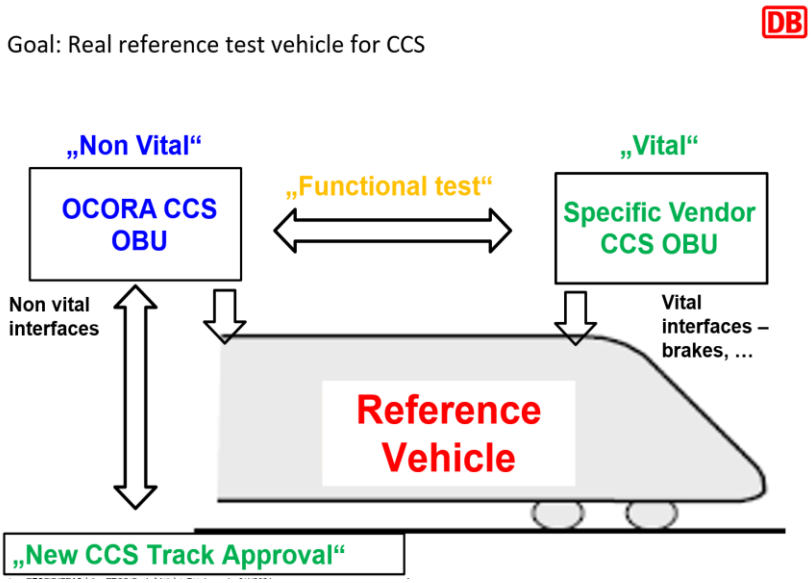
A formal work on the specification related to the interface ETCS-TCMS has been recently initiated and already introduced to suppliers in the recent bids.



3.4.3 Lessons learned / Recommendations for OCORA

Final target could be to have a reference vehicle to test, by contract, the different solution from the specific vendors.

A “moving test laboratory” POC on a vehicle is under contract execution from DB



Several test model (OBU, DMI, Track) including a test, environment to run them (PC based), can eventually be inherited and brought to final design.

A formal work on the specification related to the interface ETCS-TCMS has been recently initiated and already introduced to suppliers in the recent bids.

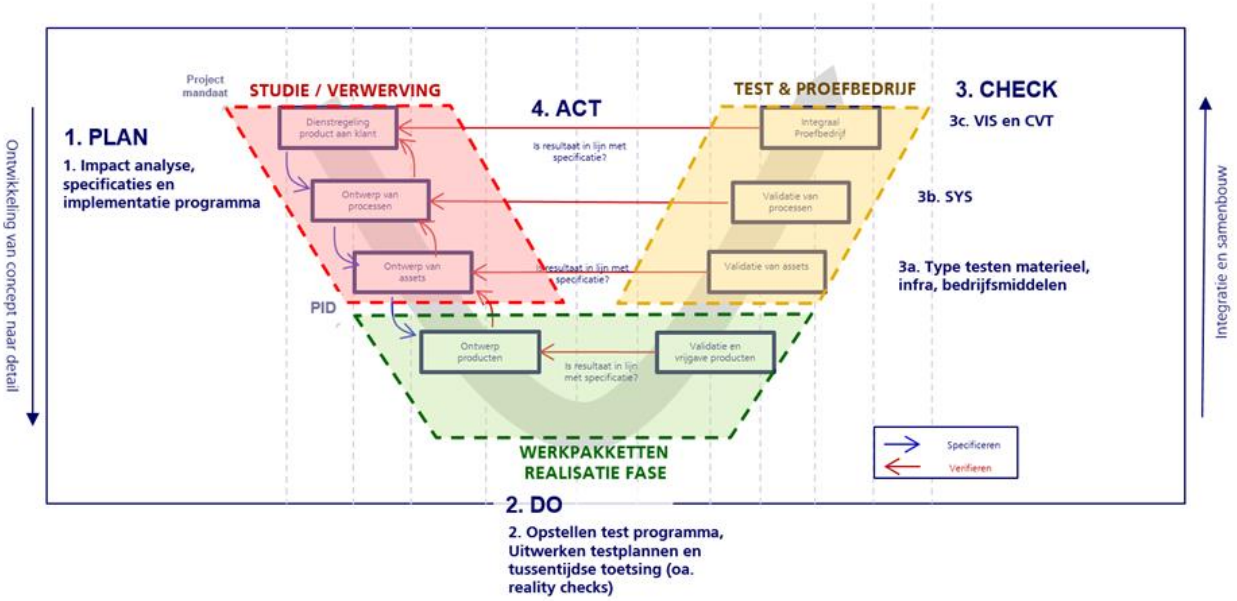
3.5 Experience sharing: Test strategy ERTMS at Dutch Railways (NS)

3.5.1 Brief introduction of the IVV experience

In the present section, NS present its test strategy and activities for ERTMS at Dutch Railways : different steps and goals, associated sequence/time frame.

3.5.2 Short description of what was performed

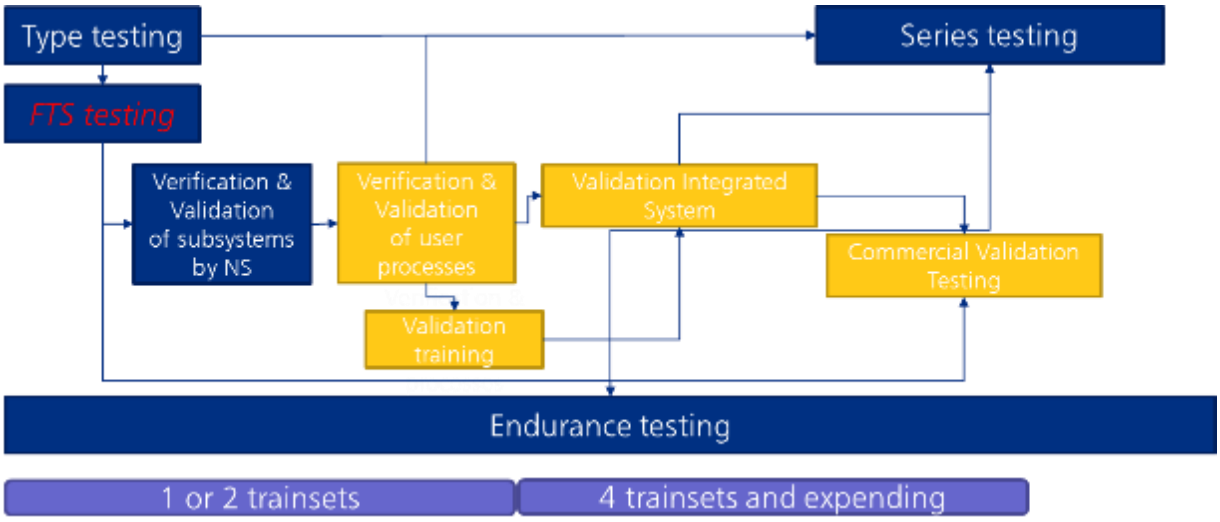
The following stepwise approach are typically performed:



The sequence is the following:

- Step 1: Type testing
  - Type Authorisation (TA)
  - *Testing for Technical Specs from NS*
- Step 2: Series testing;
  - For APoM (Authorisation for Placing on the Market) / for Dutch inspection: Declaration of Conformity
  - Acceptance by NS
- Step 3: Validation subsystems and processes
- Step 4: Validation endurance
- Step 5: Validation training (drivers and maintenance)
- Step 6: Validation Integrated System (VIS); running a simulated train service on actual ERTMS-tracks
- Step 7: Commercial Validation Testing (CVT); testing of the complete chain for operation. Running an actual trainservice, with travelers, not yet compliant with all Key Perfomance Indicators

The corresponding overall time frame is the following :



3.5.3 Lessons learned / Recommendations for OCORA

Performance testing ERTMS: how it is done:

Lessons Learned HSL: retrofit V&V is complex and has extensive interfaces with infra manager.

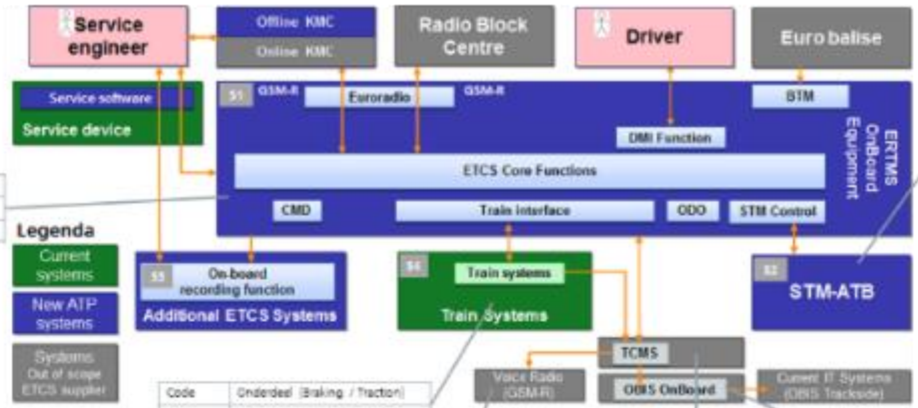
Business case approach: reliability growth is approached from operational perspective, incremental testcases

Special attention is paid to representativity of testcases, operational concept description and clearly defined system boundaries (operational and technical).

Architecture driven approach to retrofit of ETCS OBE:

Legend
Periphery component for STM-ATB / ETCS only
Periphery component for STM-ATB / ETCS and Train systems
Periphery component to be discussed
Periphery component not relevant for STM-ATB / ETCS

Code	Onderdeel (ATB)
PW482041	Opneemspoel met kabel L=1100+/-10
TBD	Aansluitdoos voor ATB opneemspoelen
FA202209	relais 4K10 (5km/h detectie)
TBD	2S2 Stuurstroomslot
FD271307	Remkraan
44601006	Relais 2K47 Dimmen signaallampen
44601006	Relais 2K12 hulprelais geen bediende cabine
44601006	Relais 2K13 Dienstvaardig
44601006	Relais 2K14 Dienstvaardig lokaal
55100203	9S2 ATB Buitenbedrijf schakelaar
KF106237	9Y1 MAGNEETKLEP WBEMV810-001 110V NW12
FD301282	9C1 Schakelaar magneetklep IN/UIT
55661500	automaat 9F2
TBD	Koppeldoos
TBD	Boordnet
FD296999	Aagevversamenstelling ATB fase4
FD296254	Aansluitdoos ATB aagever
57318504	ATB rembel
57431066	ATB GONG
55661500	automaat 9F1



Code	Onderdeel
TBD	Boordnet
TBD	Koppeling/koppeldoos

Code	Onderdeel (Braking / Traction)
TBD	Remkraan

Code	Onderdeel (Cab radio GSM-R)
TBD	Relais 2K14Dienstvaardig lokaal
FD270237	GSMR radio
TBD	Microfoon
PP430000	Antenne
TBD	Automaat 2F6

Code	Onderdeel (Diagnose)
Var.	Automaat 2F4
Var.	Relais 2K1
Var.	SSM1
Var.	SSM1
Var.	POM2

Code	Onderdeel (ORBIT en OBIS)
55992000	glasbus zekering
FD321714	OBU
FD322373	Antenne 9T1
FD205646	Excider 9P100
44601006	Relais 2K14Dienstvaardig lokaal
55661500	automaat 9F2
44602100	Relais 3K00 inschakelen NON preferente net
44601006	Relais 2K23 bediening sluimeren
44601006	Relais 2K21 Reinigen
44604544	Relais 3F25 Detectie 110VDC Accu
44940211	Relais 3F00 Minium spanning

Performance testing ERTMS: risk-based approach (CSM explicit risk assessment combined with operational risk and Human factors assessment):