DEFENCE AND SPACE

Applying ECSS-E-TM-10-23 – lessons learnt

Harald Eisenmann, Claude Cazenave

ESA Ontology Workshop

AIRBUS

Applying ECSS-E-TM-10-23 – lessons learnt: Overview

Applying ECSS E-TM-10-23

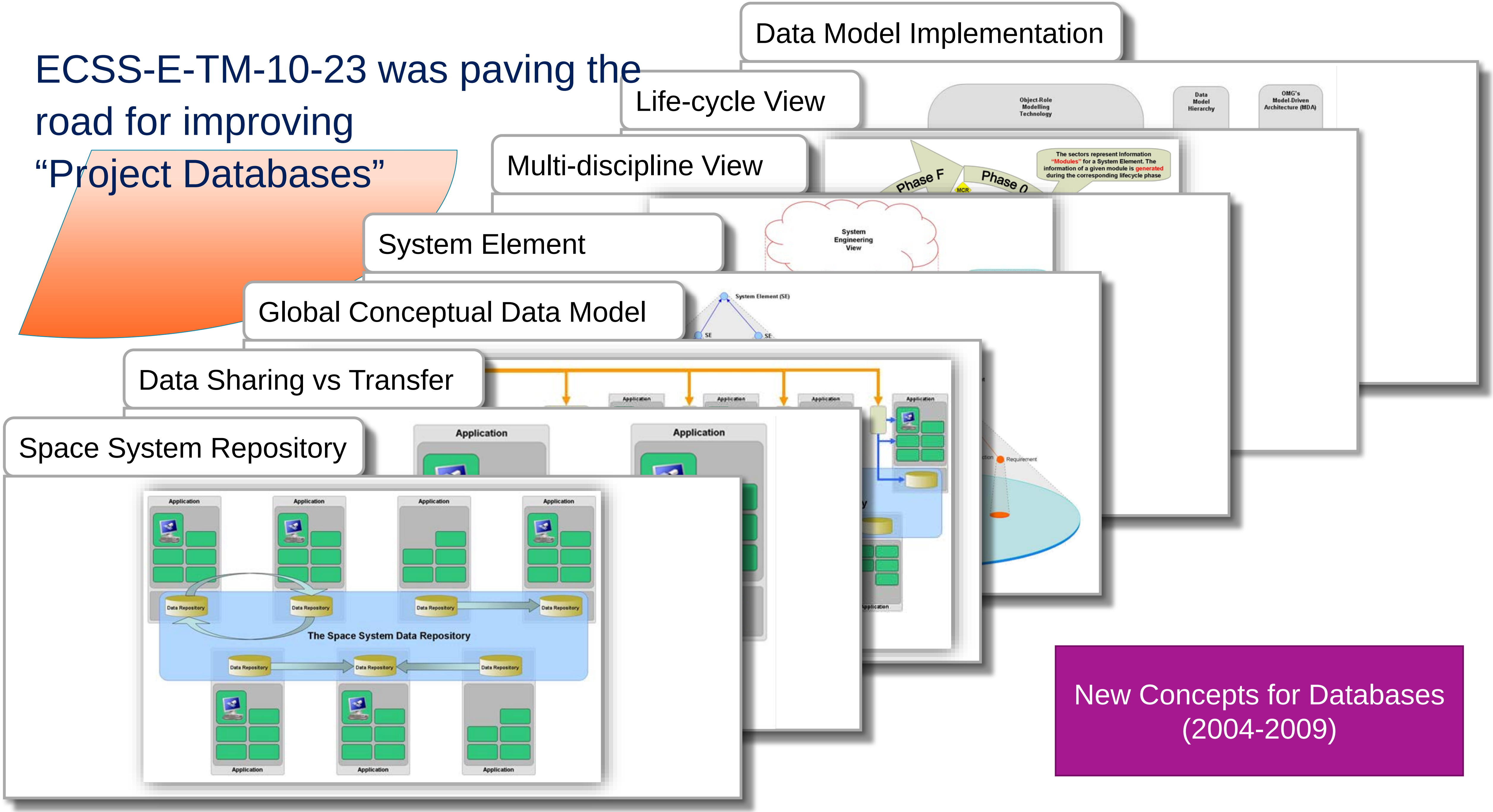
Chapter 1:
The Journey from „New Concepts“ to operational realization

Chapter 2:
Operational status of „Digital Continuity“ reached

Chapter 3:
Lessons learnt from applying ECSS-E-TM-10-23 concepts

AIRBUS

ECSS-E-TM-10-23 was paving the road for improving "Project Databases"



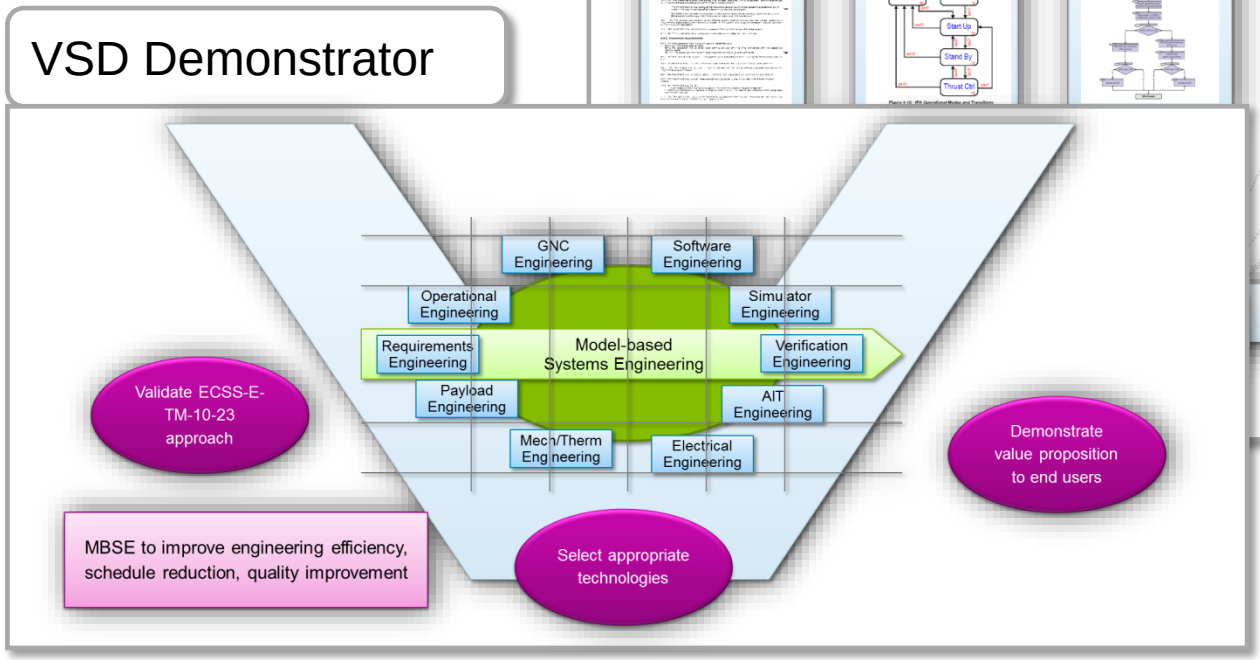
New Concepts for Databases (2004-2009)

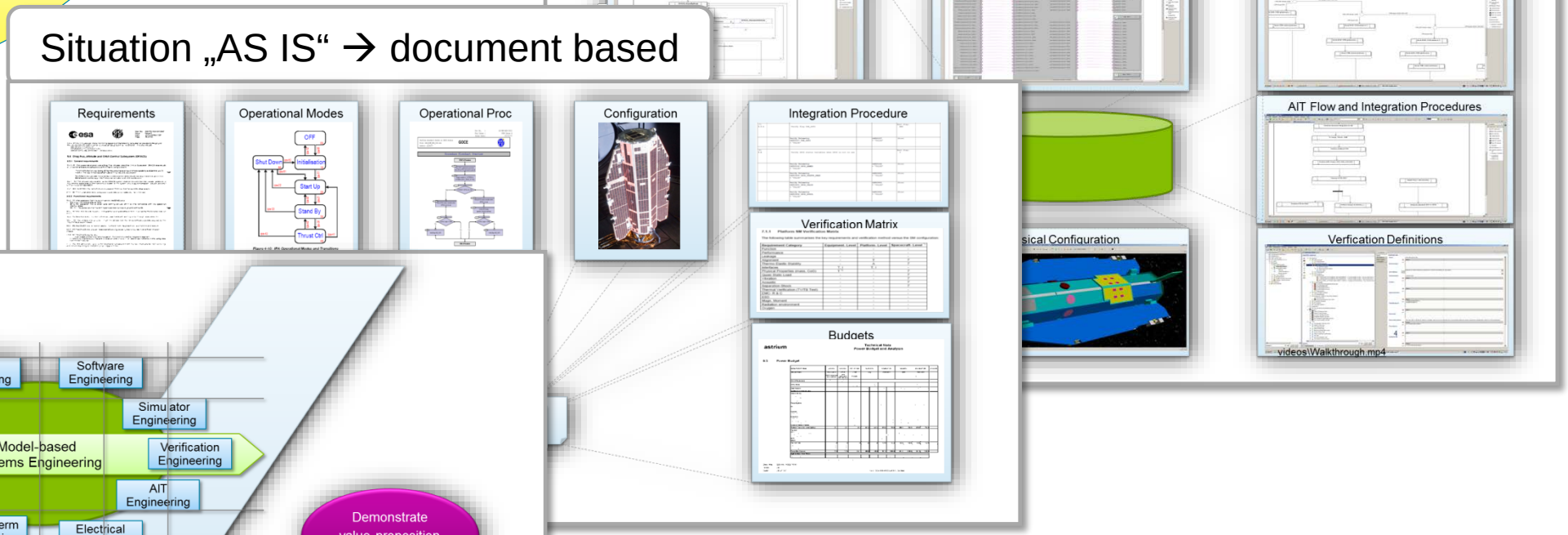
ESA “Virtual Spacecraft Design” (VSD) was do demonstrate / validate the beneficial use of ECSS-E-TM-10-23 for Systems Engineering

Demonstrated „TO BE“ → model-based

Situation „AS IS“ → document based

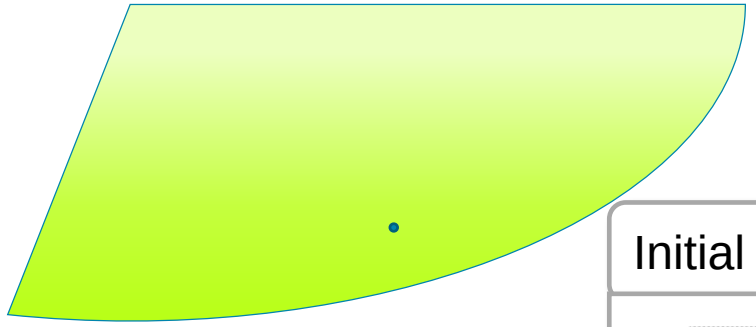
VSD Demonstrator



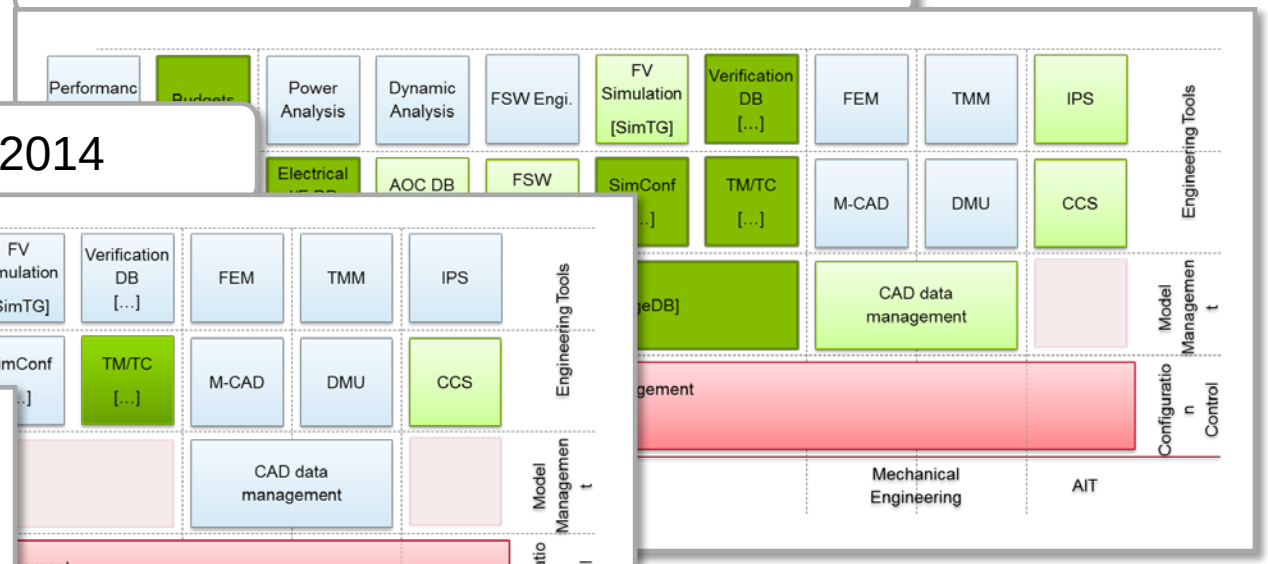


Demonstrating Feasibility (2008-2011)

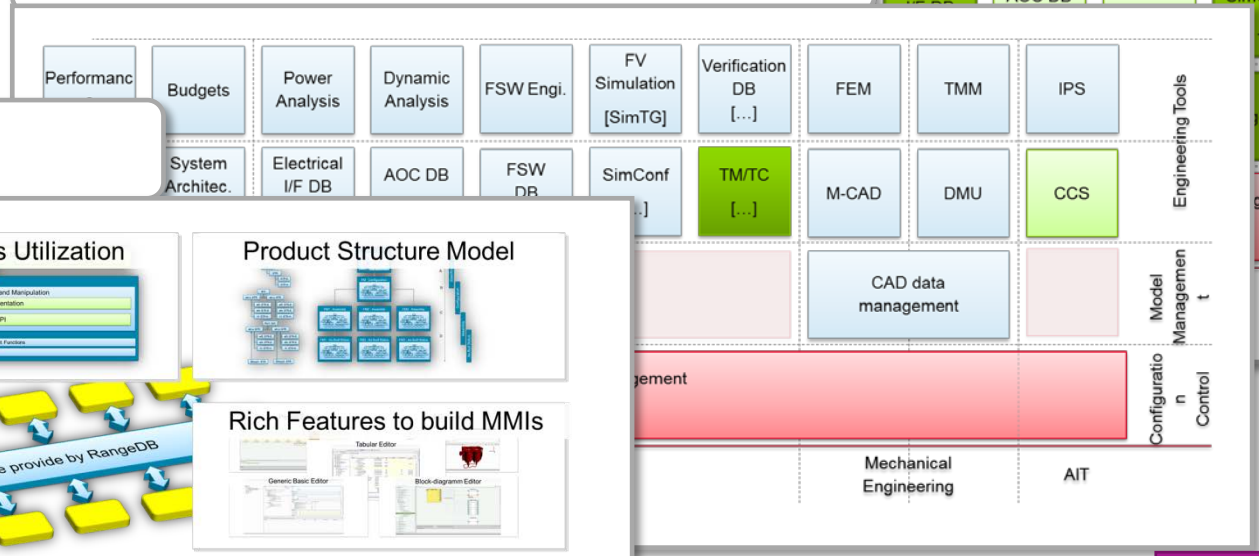
Adopting the VSD principles for Functional Avionics (→ SRDB): RangeDB



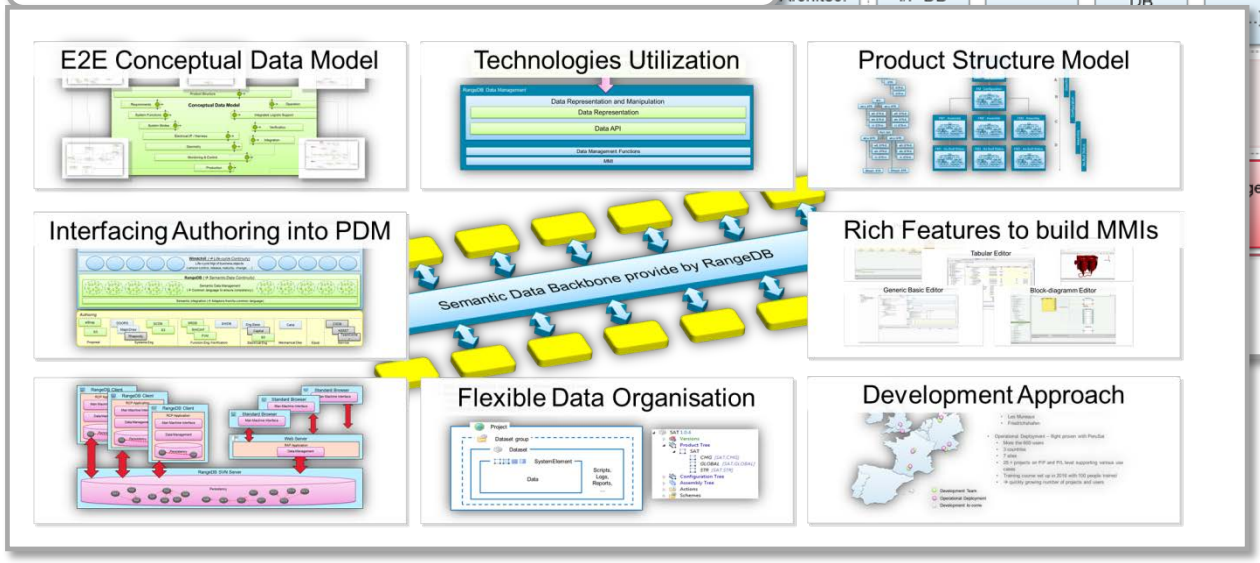
Evolution up to 2016



Initial Development Focus 2014



Framework principles



Devevelopment for operational use 2011-2014, operational use since.

Applying ECSS-E-TM-10-23 – lessons learnt

Applying ECSS E-TM-10-23

Chapter 1:

The Journey from „New Concepts“ to operational realization

Chapter 2:

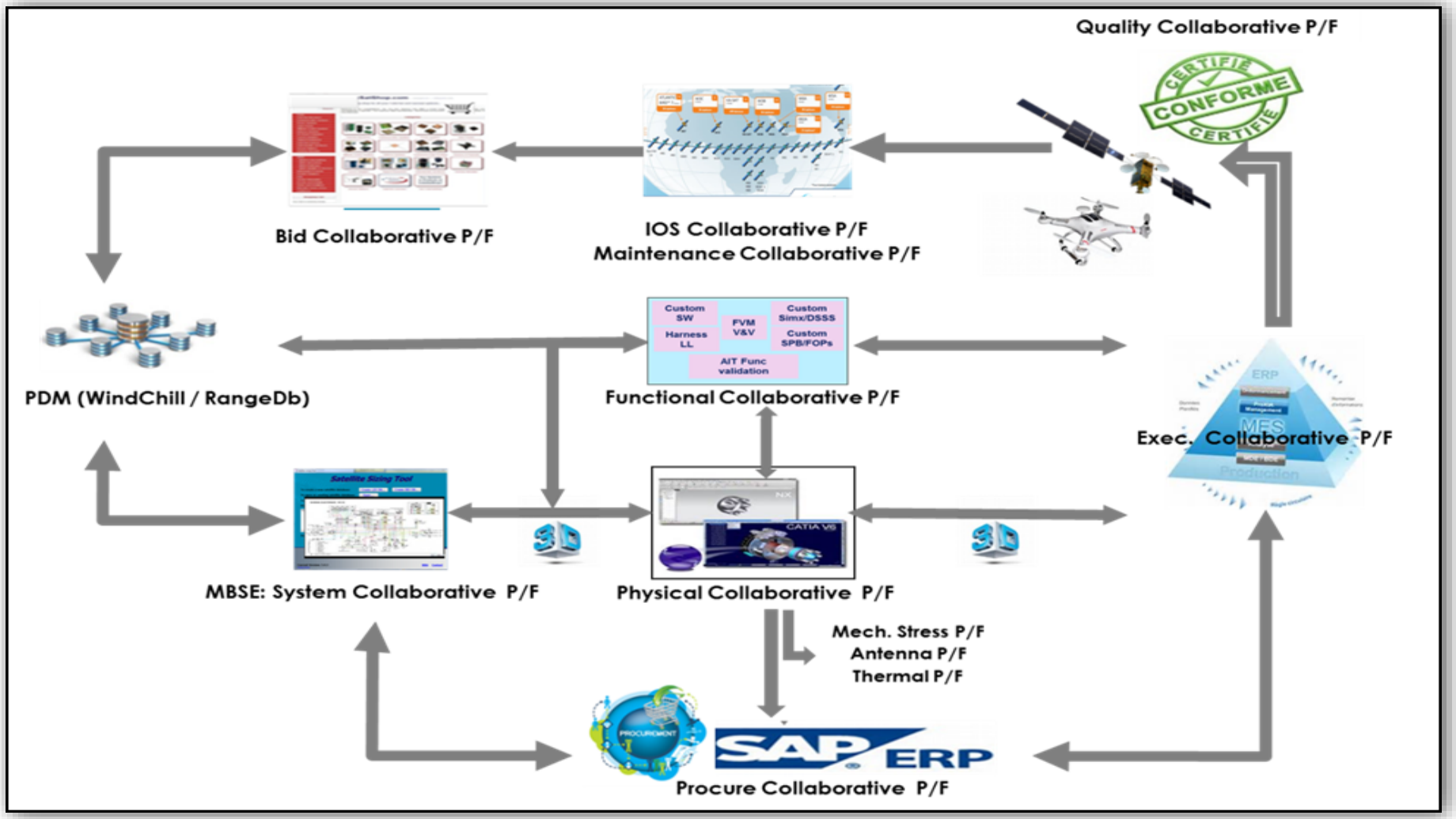
Operational status of „Digital Continuity“ reached

Chapter 3:

Lessons learnt from applying ECSS-E-TM-10-23 concepts

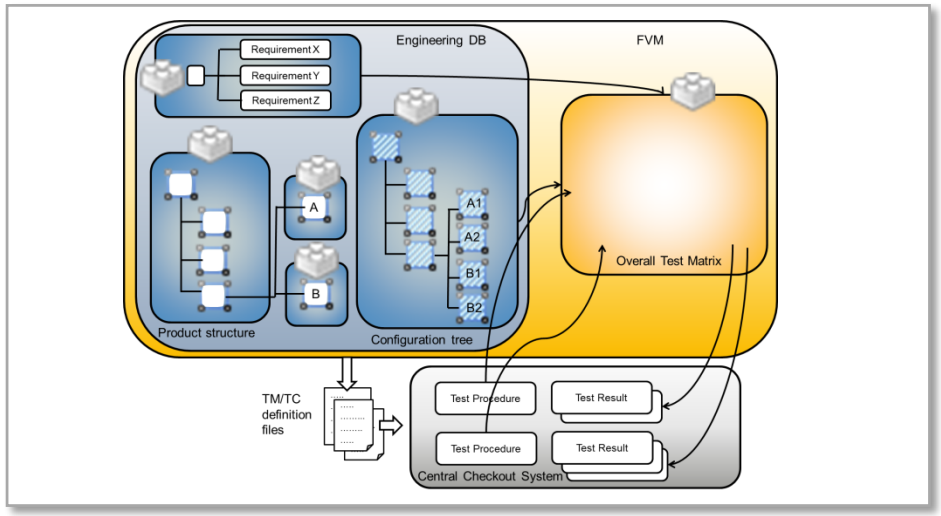
AIRBUS

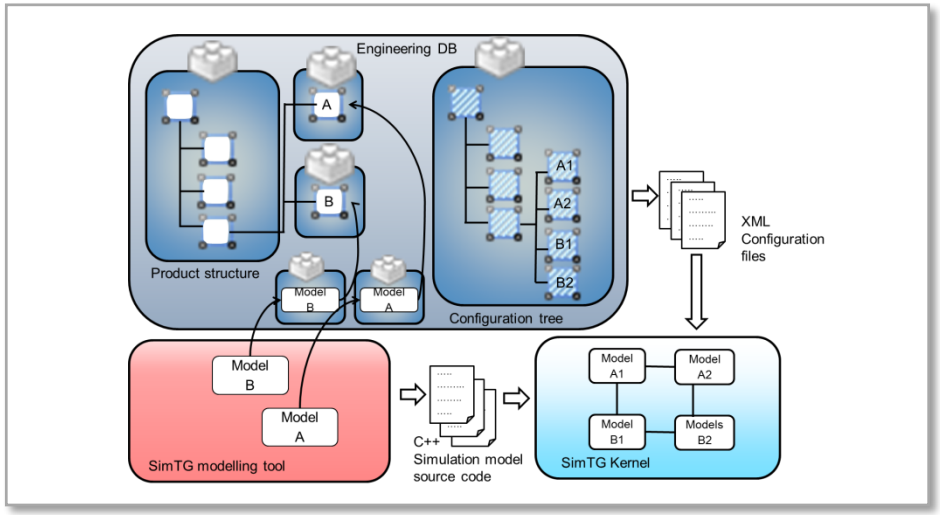
Digitalization Space System - DDMS Space System EuroStar NEO - Overview

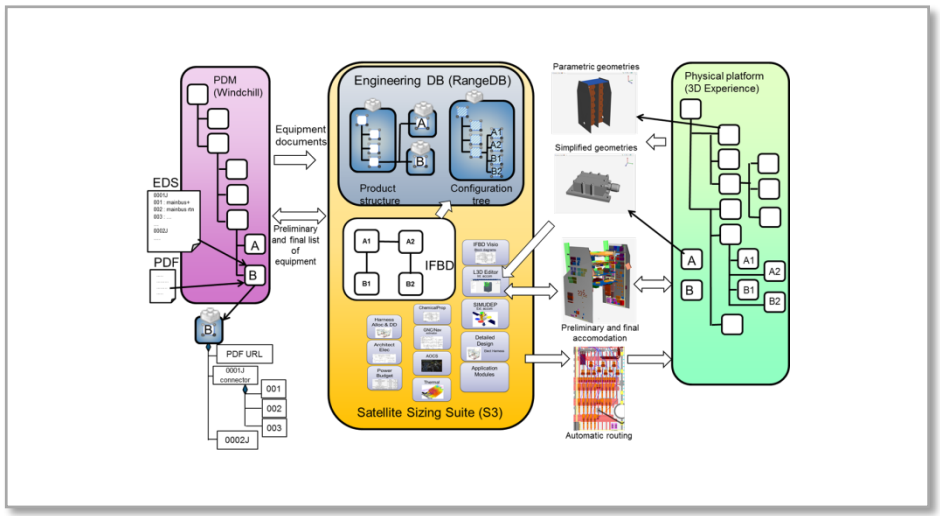


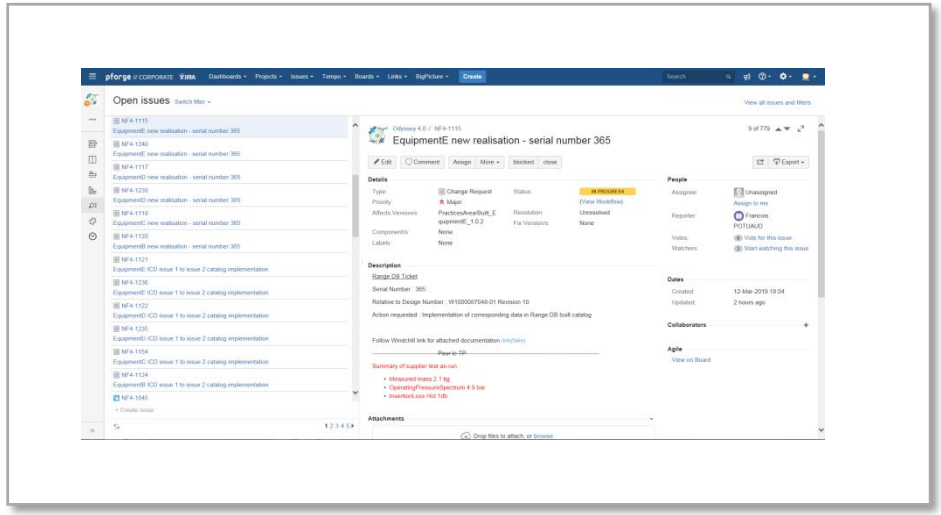
AIRBUS

Key objective is to improve the E2E Digital Flow of Information – some examples





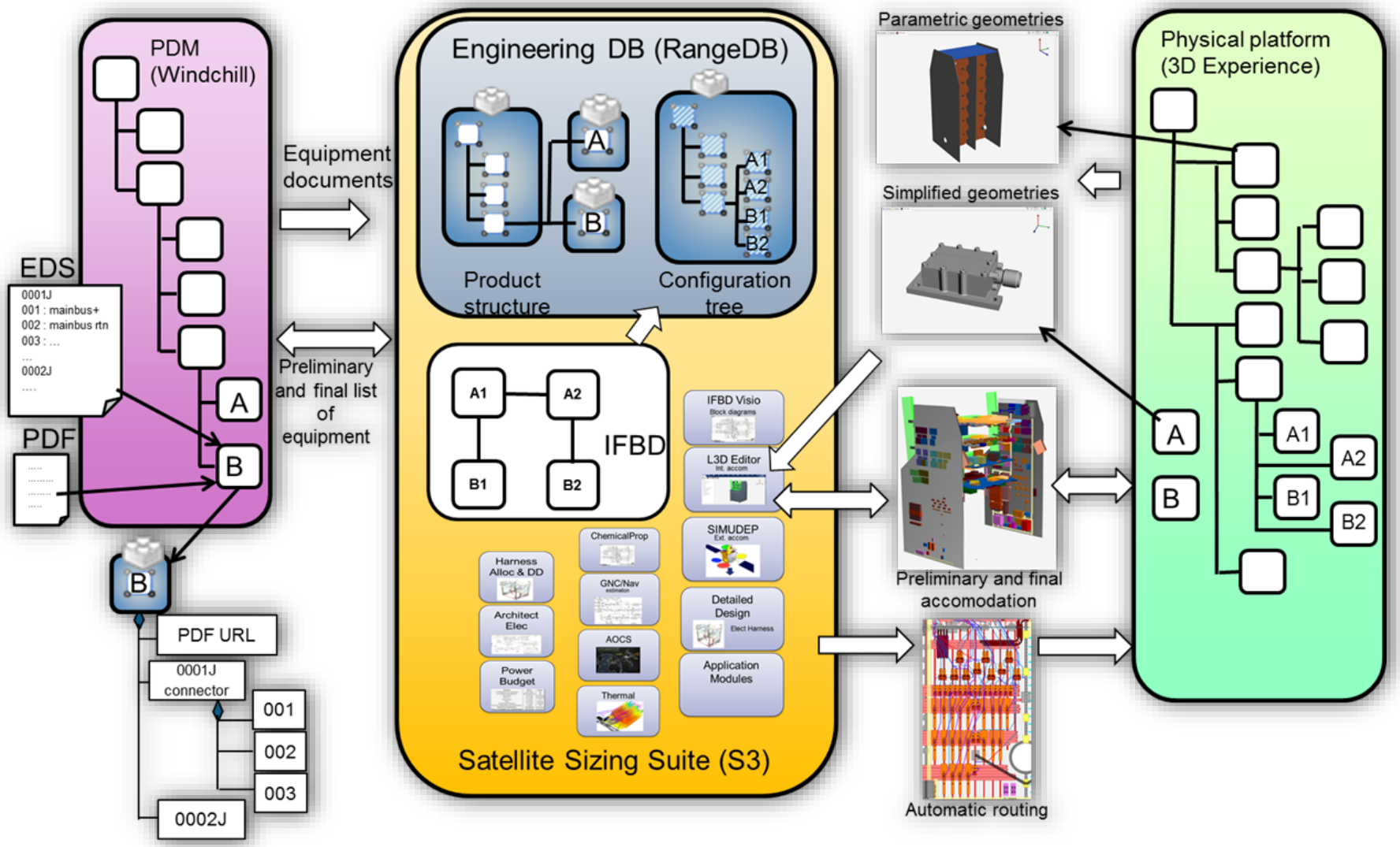




But there is more
- From product to project
- Multi-disciplinary engineering in early phases
- From SE into disciplines
- From disciplines into manufacturing
- Effective Co-engineering
- Overall agile working
- …

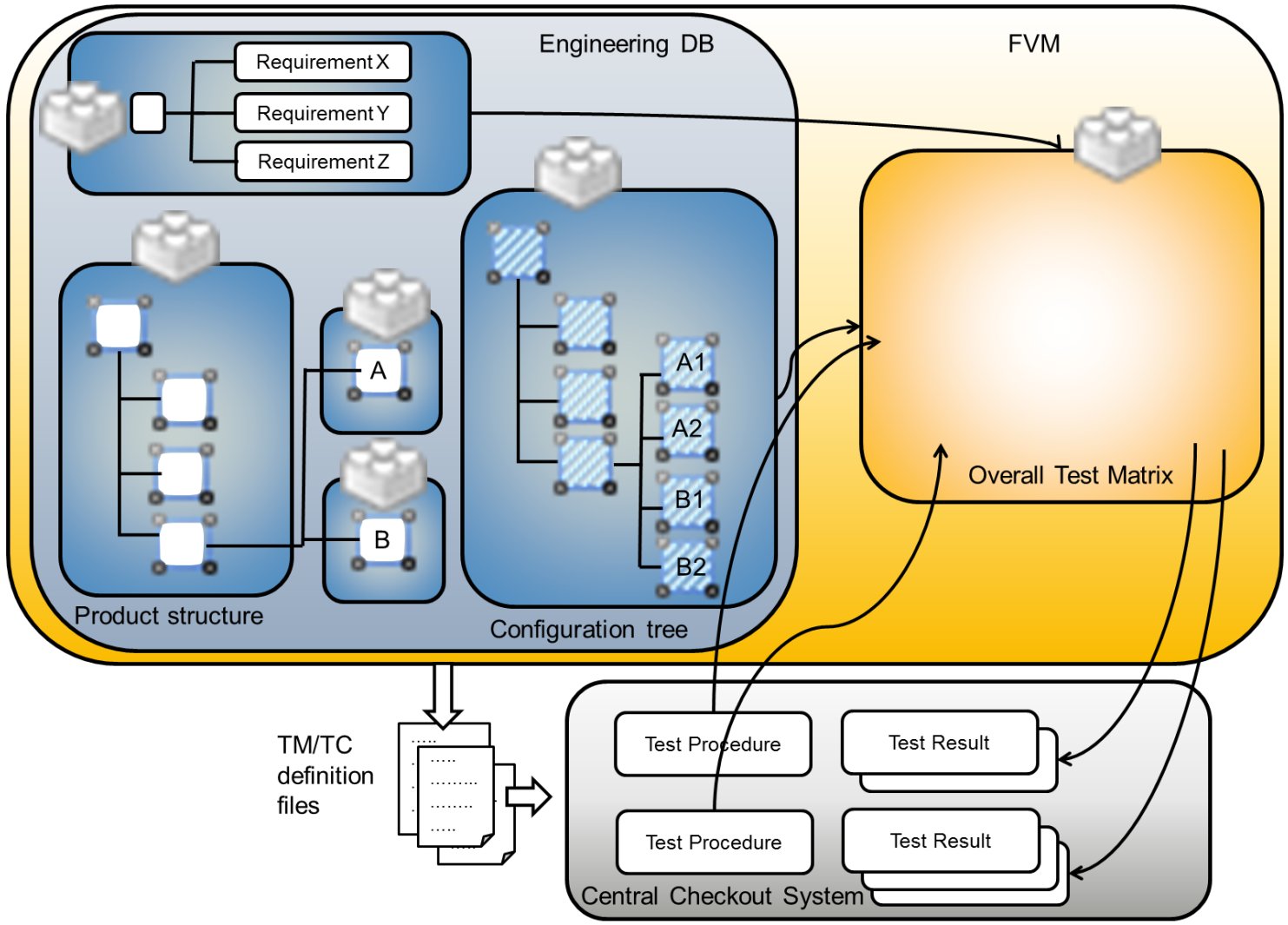
AIRBUS

Digital integration of Key Backbone Platforms



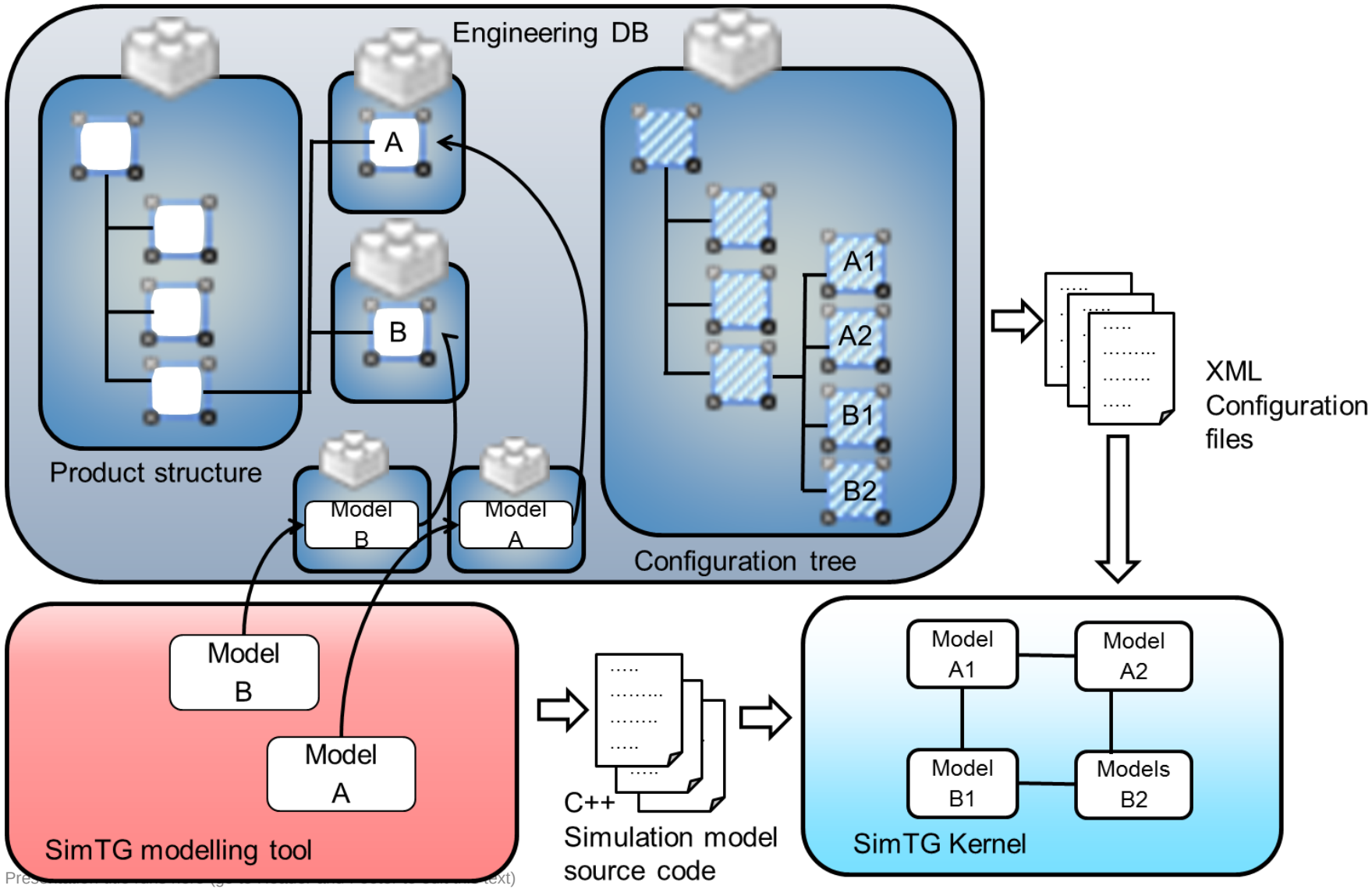
AIRBUS

Integration of Engineering, Test Engineering and Testing



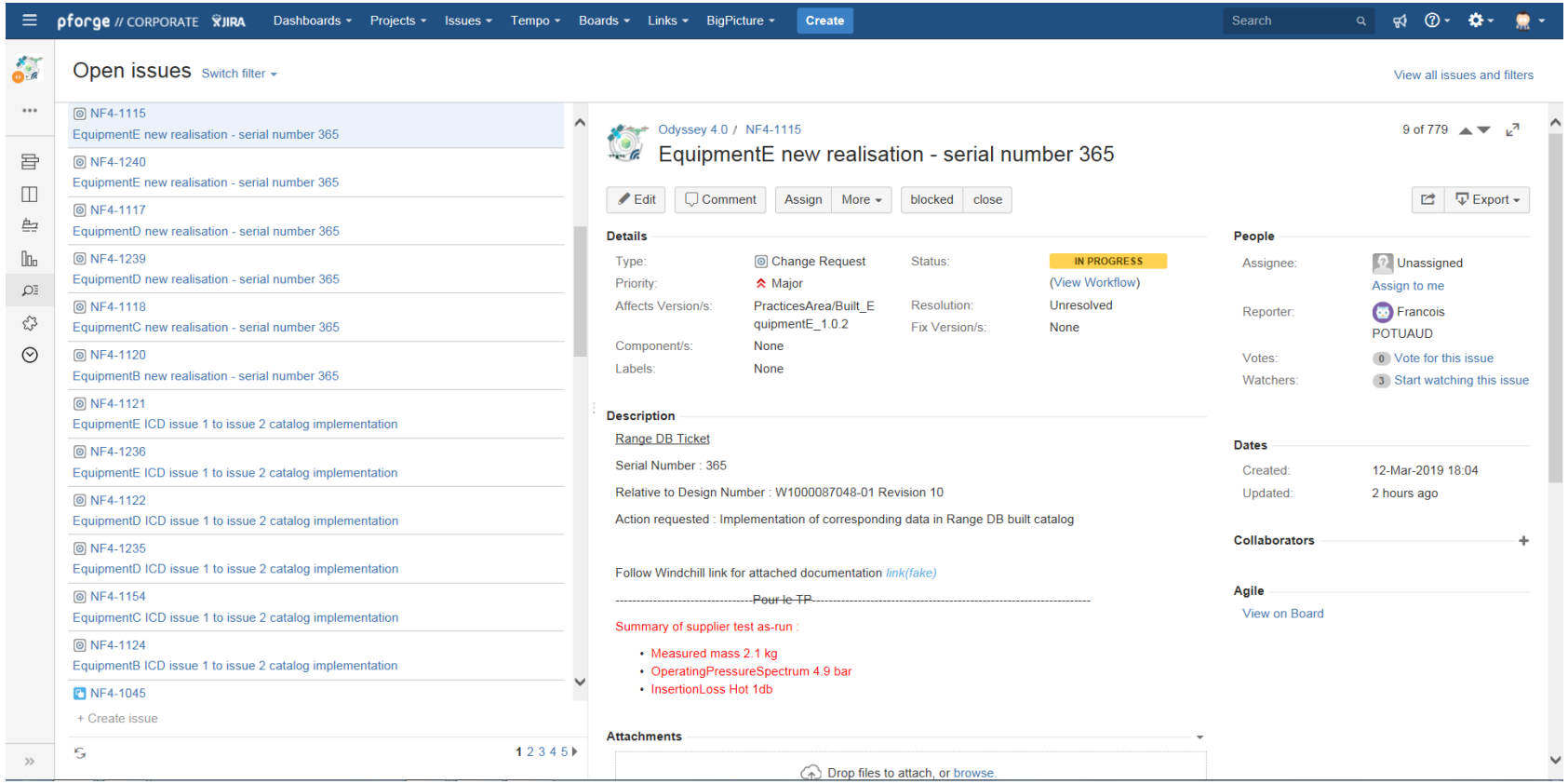
AIRBUS

Integration of Engineering and Simulation Configuration



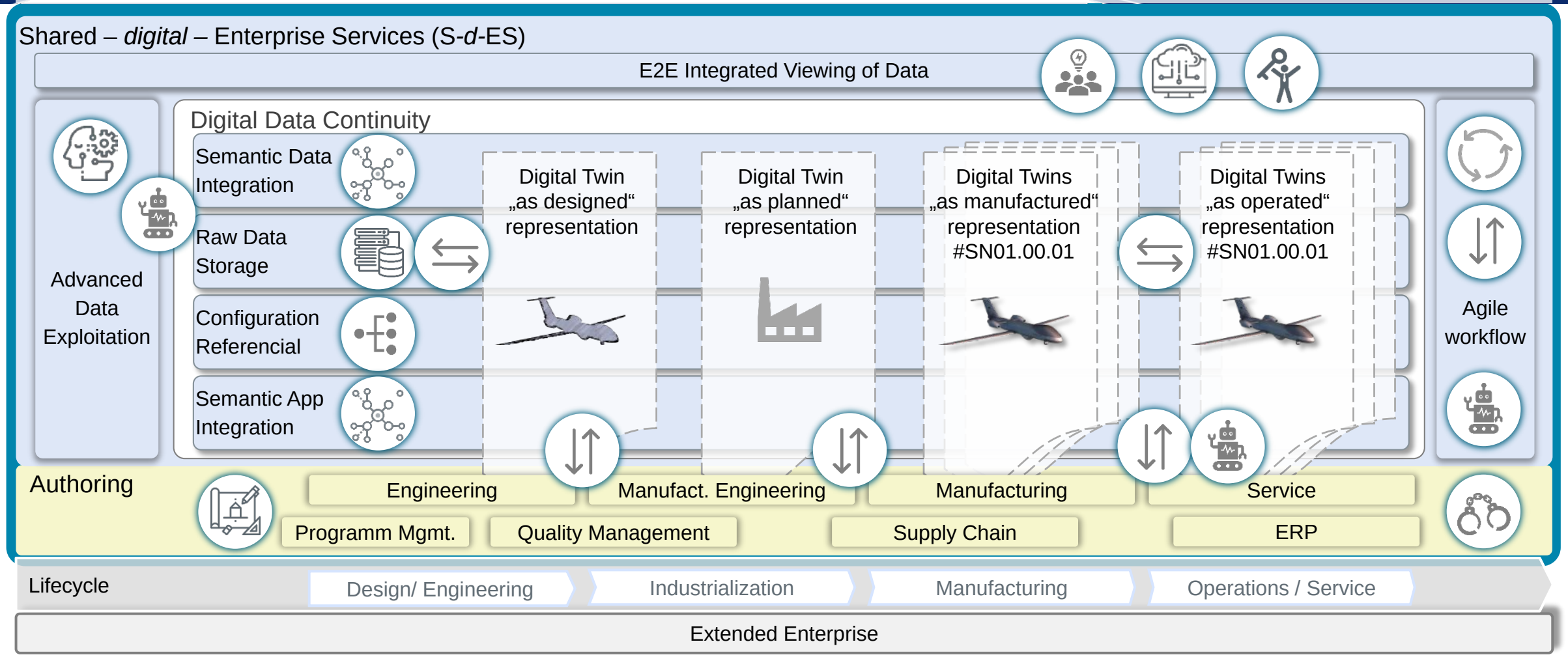
AIRBUS

Integration to obtain “overall agile workflow”



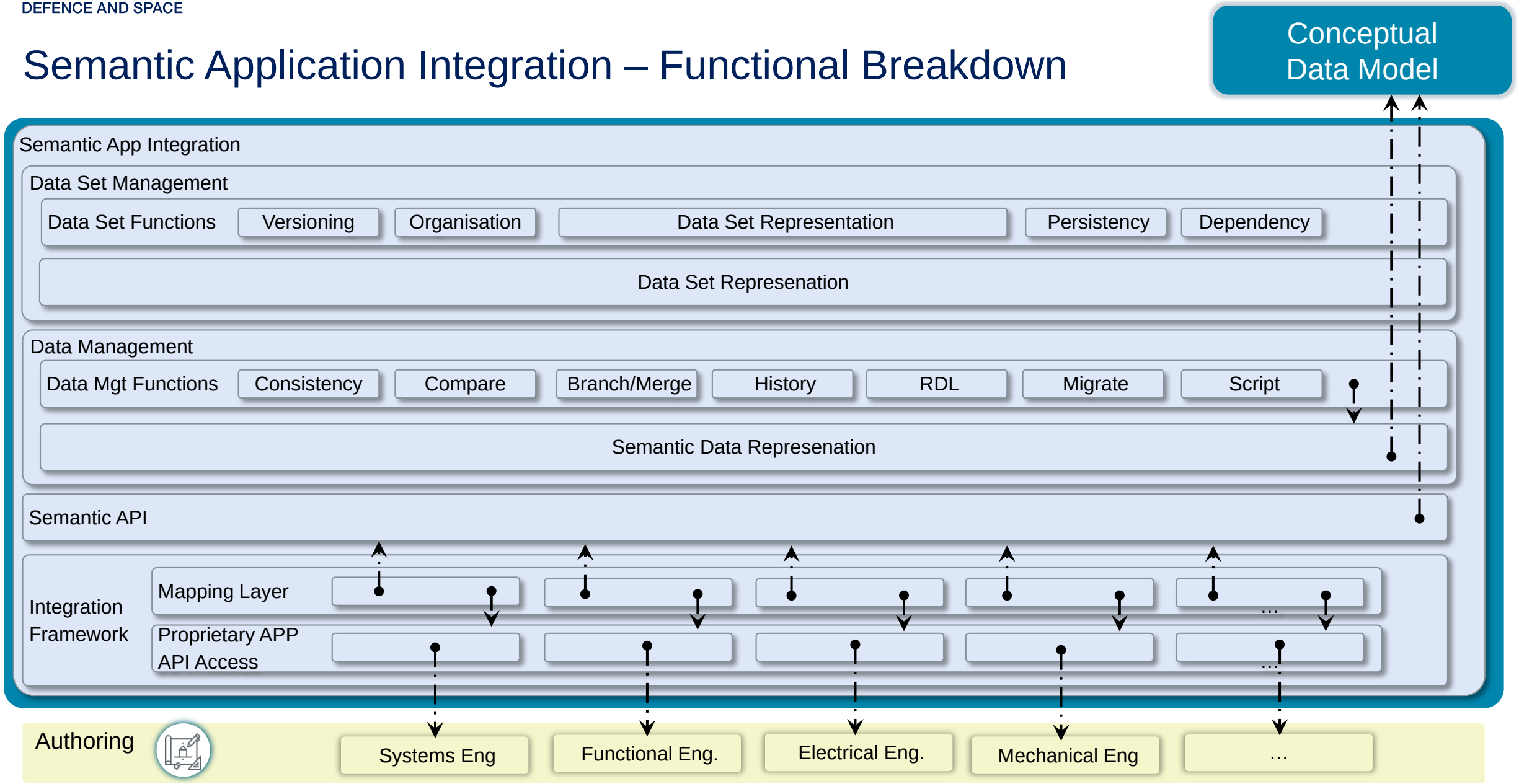
AIRBUS

Space System Data Repository is part of Shared -*digital*- Enterprise (S-*d*-ES) Services



AIRBUS

Semantic Application Integration – Functional Breakdown



Applying ECSS-E-TM-10-23 – lessons learnt: Overview

Applying ECSS E-TM-10-23

Chapter 1:
The Journey from „New Concepts“ to operational realization

Chapter 2:
Operational status of „Digital Continuity“ reached

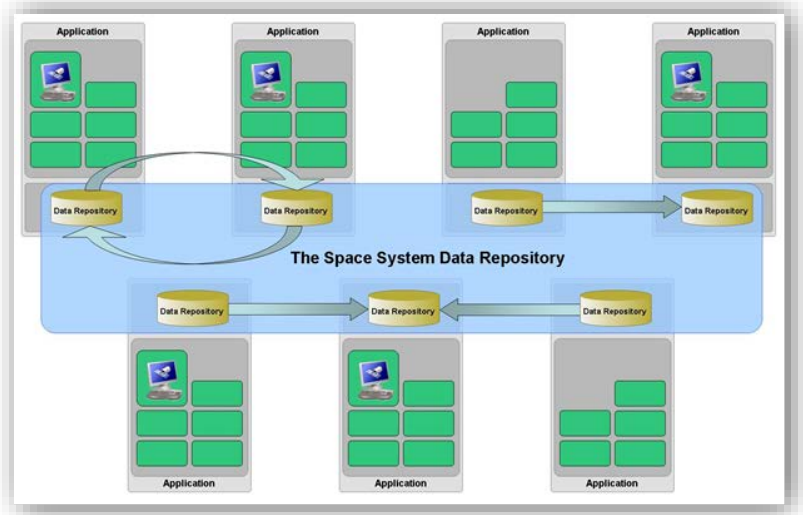
Chapter 3:
Lessons learnt from applying ECSS-E-TM-10-23 concepts

Lessons learnt – along the initial key concepts of ECSS-E-TM-10-23 - Space System Repository

[→ Space System Data Repository] was addressing the need to enable interoperability across different applications

- Applications are driven by the individual stakeholder, with the result – that the overall environment is still very diverse
- Commonly shared digital services are needed to enable sharing and exchange of data
- Shared digital services comprise of data management functions based semantic representation - derived from a [→ Conceptual Data Model]

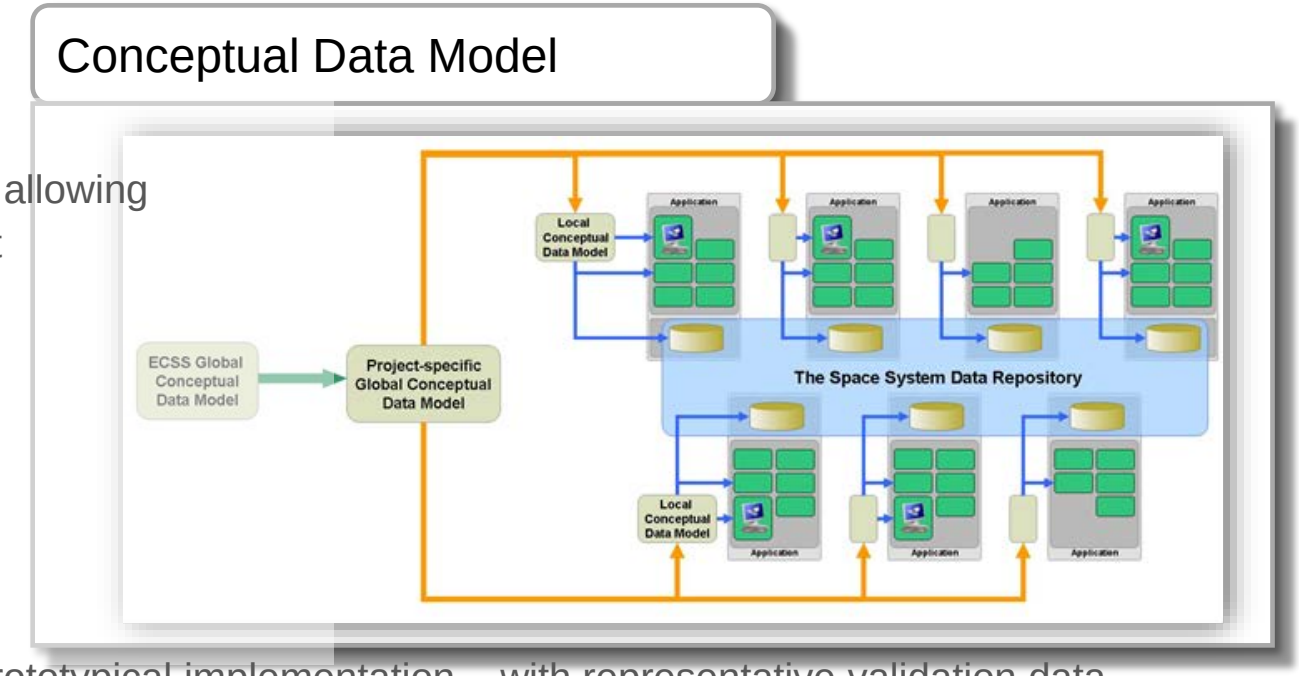
Space System Repository



AIRBUS

Lessons learnt – along the initial key concepts of ECSS-E-TM-10-23 - Conceptual Data Model

Conceptual Data Model is to specify the common semantic representation to provide a common E2E language



- [→ Conceptual Data Model] turned out to be the key element allowing to formally specify the data – outside the development context
- But the meaning is little if not backed with a concrete physical implementation – while closely preserving the semantics
- Setting up a common language across domains/phases is possible, enabled by close consideration of user terminology
- Data model engineering needs to be closely supported with prototypical implementation – with representative validation data
- The data model has 2 parts – one which is directly used for development – the other “categories” allow tailoring at runtime
- [→ Conceptual Data Model] will evolve since new integrated stakeholder will introduce their constraints (process chains)

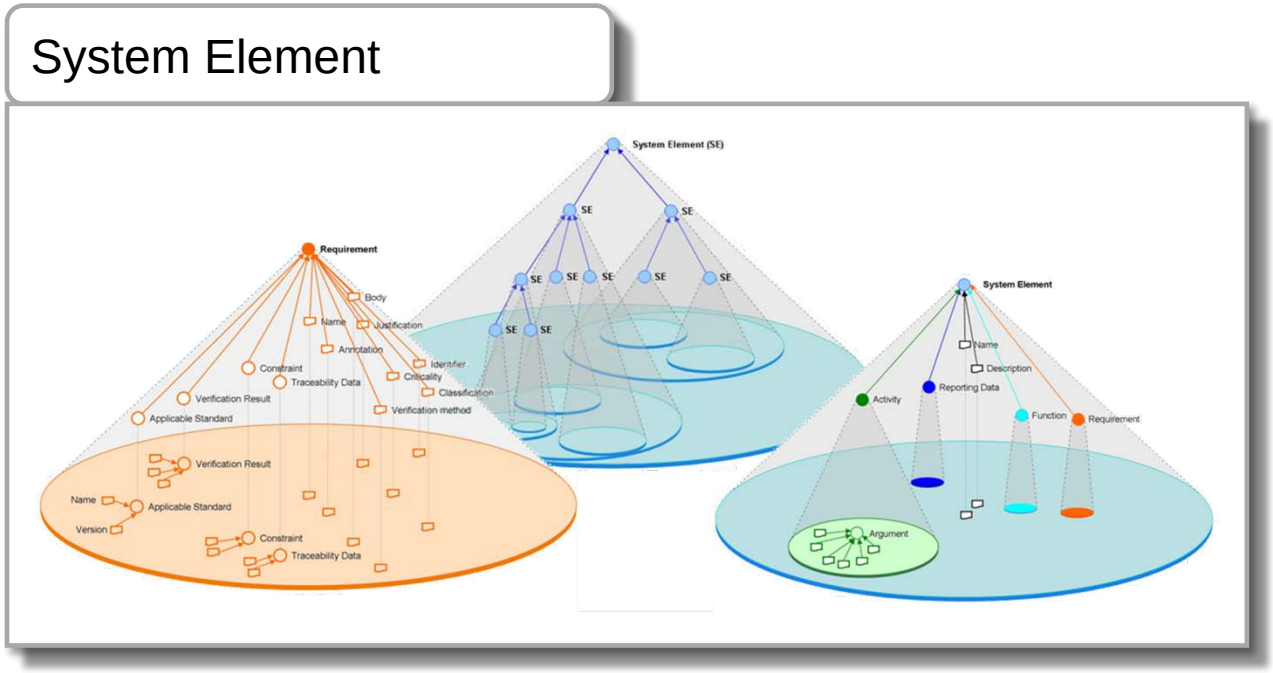
Lessons learnt – along the initial key concepts of ECSS-E-TM-10-23 - System Element

[→ System Element]is a key element aggregate knowledge on system hierarchy, life-cycle or domain

- System element turned out to be a “strong concept” in the [→ Conceptual Data Model] to describe different notions
  - System breakdown structure
  - Considering evolution along [→ Life-Cycle] : Definition, Configuration Occurrence, Realization
  - Across different domains to obtain [→Multi-disciplinary views]
  - Individual properties to capture equipment characteristics

- [→ System Element] is a key enabling element towards “Digital Twin”

- System Element aspect is also close to versioning of data



AIRBUS

Lessons learnt – along the initial key concepts of ECSS-E-TM-10-23 - Multi-discipline View

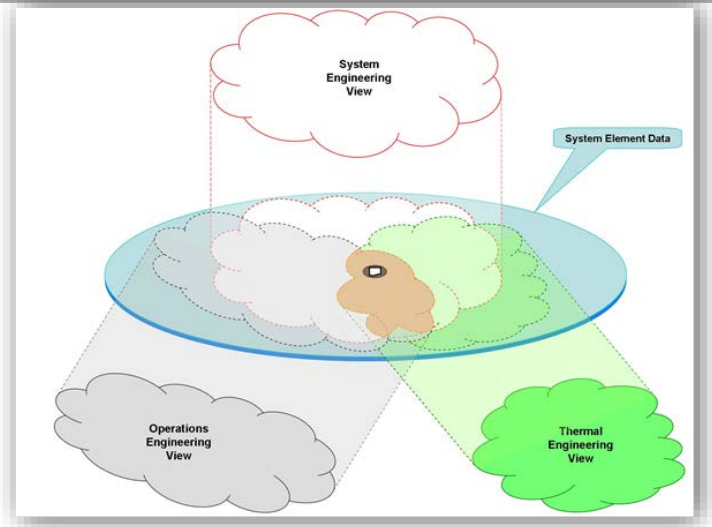
Multi-disciplinary views are needed to respect the different views on [→ System Elements] from different domains

- Notion of System Element has been enhanced with “aspects” to enable individual – but linked compartments

- The aspects comprise areas such as
  - Electrical: connector, pin, func channel / net, channel, …
  - Monitoring & Control: Packet, parameter, calibration…
  - Mechanical: position, orientation, …

- Aspects do represent different hierarchy level of data across the life-cycle

Multi-discipline View



Lessons learnt – along the initial key concepts of ECSS-E-TM-10-23 - Life-cycle View

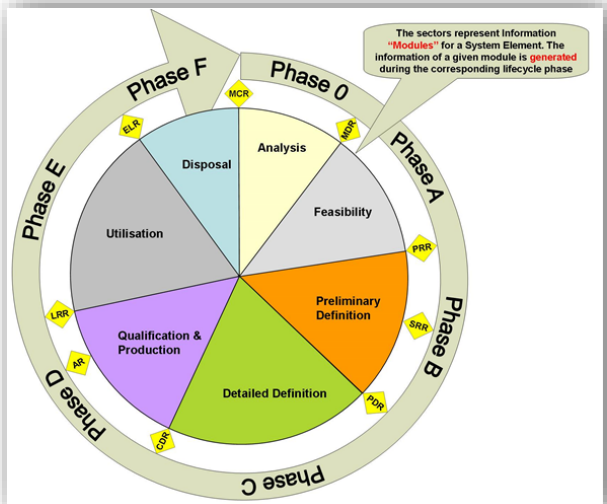
- *Life-cycle consideration to ensure that all data created along the life-cycle can be captured in the [→ Space System Data Repository]*

- [→ Conceptual Data Model] is defined to capture all data from requirements, design into FV and operation

- However for full life-cycle representation of data the following needs to be considered – and linked - as well
  - Data managed in the context of configuration / PLM
  - Data needed to manage Agility
  - Extended enterprise

- For extended enterprise good experience has been made with EDS for “simple data” only – for more complex data more appropriate ways need to be *agreed* !

Life-cycle View



AIRBUS

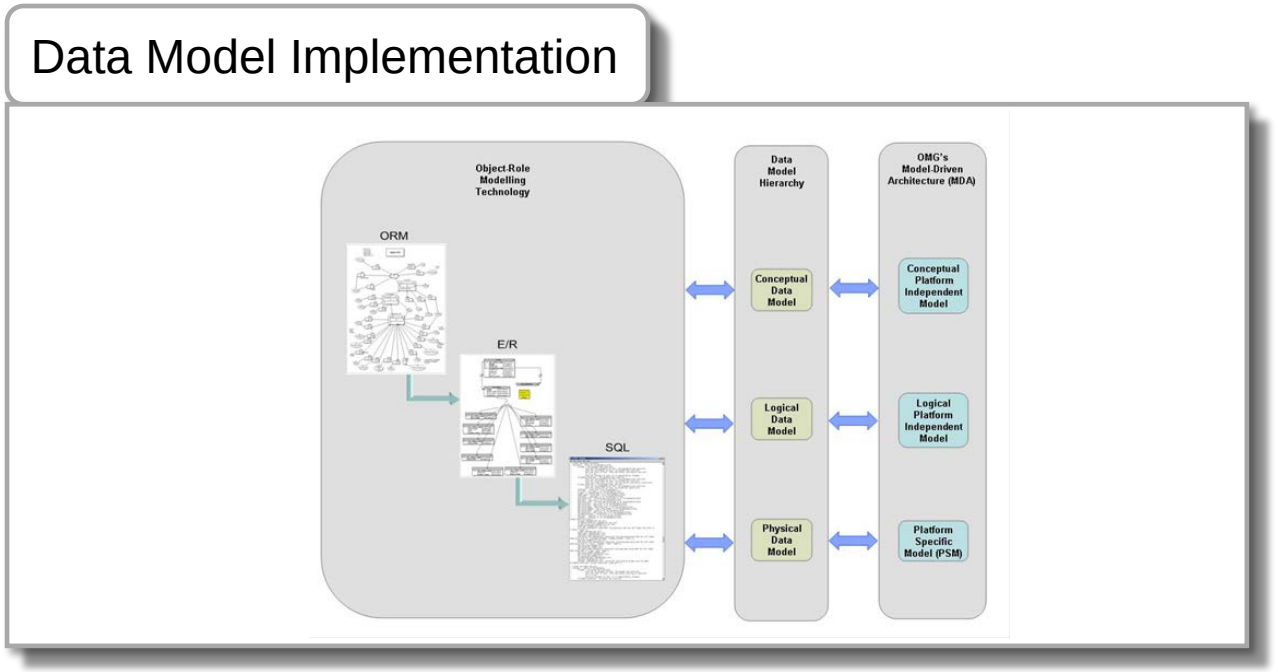
Lessons learnt – along the initial key concepts of ECSS-E-TM-10-23 - Data Model Implementation

Implementation view is important to ensure that [→ Conceptual Data Model] can be properly mapped into the implementation

- Overall data model structured with the aim to clearly separate the different concerns
  - [→ Conceptual Data Model] (domain concepts)
  - Technical data model (enriched with technology mapping, performance, …)
  - Physical data model (actual resulting source code)

- Support of “state or the art” is supporting the conceptual data model engineering – with early and continuous prototyping

- UML turned out to be a good choice, mainstream technology, many tools, knowledge widely spread



AIRBUS

Wrapping up

- Having a formal conceptual data model is an essential foundation – but the actual truth is in the physical model !
- Therefore Ontology / Conceptual Data Model is not a purpose on its own: it has to support the efficient provision of solutions, which effectively improve the envisaged processes
- For this early and continuous prototypes are enabling the validation of on data model – and consequences at run time !
- Use of main stream technologies is key to involve different parties, engineering domains, tool vendor, development, IM, domain
- Digital transformation is an ongoing effort – in general data model is converging – however continuous updates needed – even in operation
- Solutions within organizations are in place (and where not, emerging) – collaboration across enterprises are needed, to effectively enable collaboration across organizations
  - EDS is a start for simple structured data
  - There is much more to be tackled for operational exchange of data