MODEL BASED AVIONICS: SAVOIR

THALES ALENIA SPACE (FRANCE, ITALY, BELGIUM)

ADCSS 2022: B.ATTANASIO

TABLE OF CONTENTS

1 Goal of the Study

2 Requirements Import

3 Logical Modelling & Req allocation

4 Physical Modelling

5 Way Forward

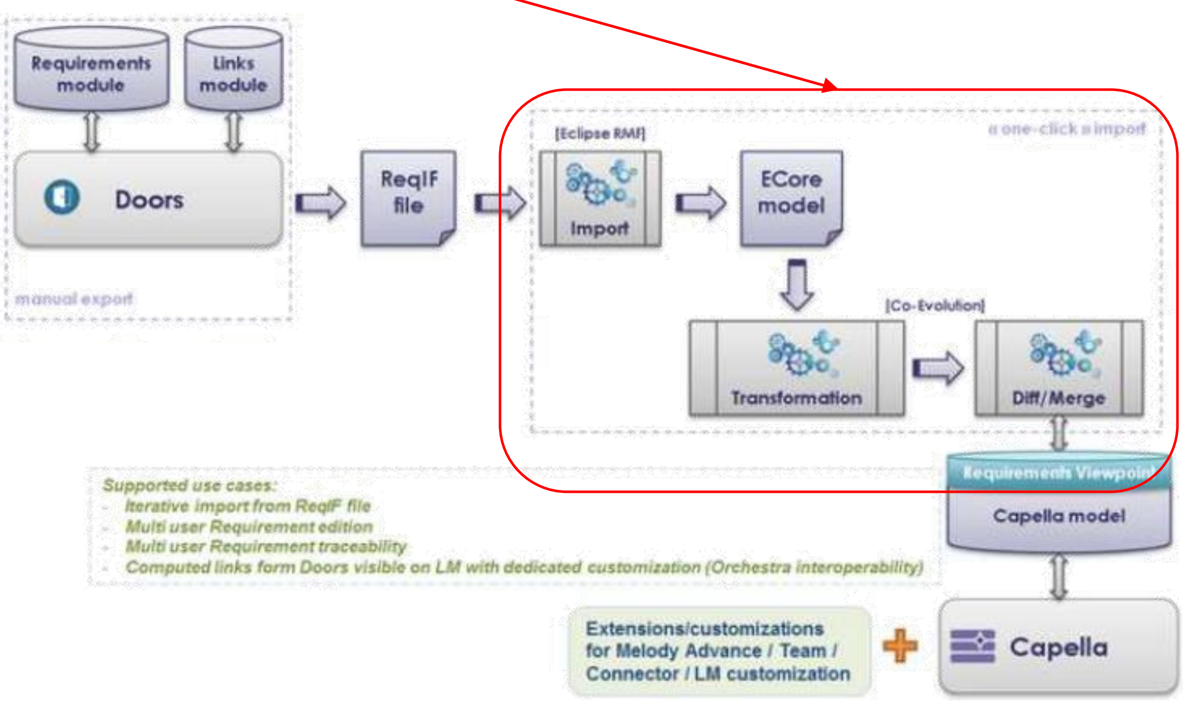
GOAL OF THE STUDY

DEVELOP MODELS OF AVIONICS BY A FUNCTIONAL CHAIN VIEWPOINT:

- **Main goal:** Rethink the avionics with a different viewpoint: by functional chain and not by block as it was historically done.
- Take SAVOIR as a basis. Integrate SAVOIR requirements & Review SAVOIR documentation
- Take the example of several products to note the divergence and propose a final ESA generic model after convergence.
- Map requirement on the different elements of the model: functional exchanges or logical functions.

REQUIREMENT IMPORT

/// Import the requirements



Supported use cases:

- *Iterative import from ReqIF file*
- *Multi user Requirement edition*
- *Multi user Requirement traceability*
- *Computed links form Doors visible on LM with dedicated customization (Orchestra interoperability)*

/// Remark on the req import: can be mapped at all level (logical or physical)

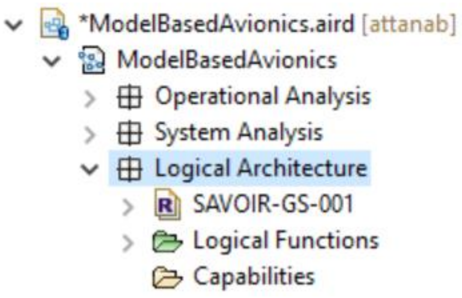
/// Following the SAVOIR documentations and the study philosophy ➔ start by the logical level.

➔ requirement import only for the Generic OBC specification & RTU

REQIF IMPORT ON CAPELLA MODEL – CONFIGURATIONS

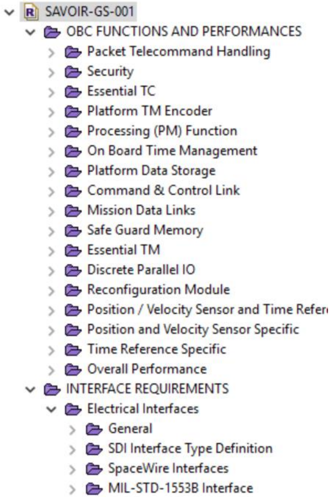
/// Capella Project Explorer: SAVOIR-GS-001 imported

SAVOIR-GS-001 imported



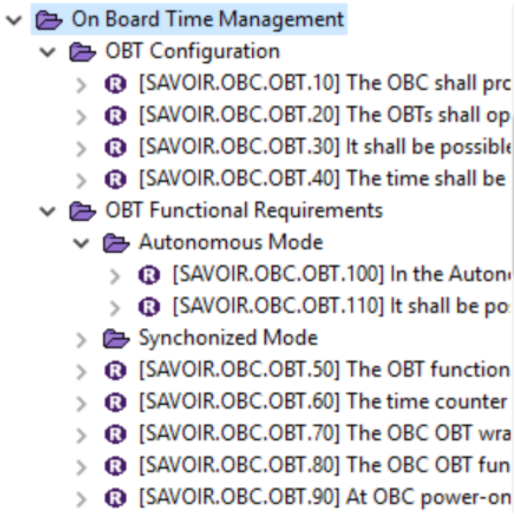
- *ModelBasedAvionics.aird [attanab]
  - ModelBasedAvionics
    - Operational Analysis
    - System Analysis
    - Logical Architecture
      - SAVOIR-GS-001
      - Logical Functions
      - Capabilities

Section by section: Avionics functions

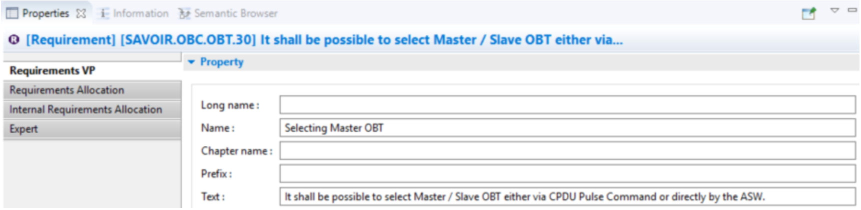


- SAVOIR-GS-001
  - OBC FUNCTIONS AND PERFORMANCES
    - Packet Telecommand Handling
    - Security
    - Essential TC
    - Platform TM Encoder
    - Processing (PM) Function
    - On Board Time Management
    - Platform Data Storage
    - Command & Control Link
    - Mission Data Links
    - Safe Guard Memory
    - Essential TM
    - Discrete Parallel IO
    - Reconfiguration Module
    - Position / Velocity Sensor and Time Refer
    - Position and Velocity Sensor Specific
    - Time Reference Specific
    - Overall Performance
  - INTERFACE REQUIREMENTS
    - Electrical Interfaces
      - General
      - SDI Interface Type Definition
      - SpaceWire Interfaces
      - MIL-STD-1553B Interface

Example of one Avionics function



- On Board Time Management
  - OBT Configuration
    - [SAVOIR.OBC.OBT.10] The OBC shall pro
    - [SAVOIR.OBC.OBT.20] The OBTs shall op
    - [SAVOIR.OBC.OBT.30] It shall be possible
    - [SAVOIR.OBC.OBT.40] The time shall be
  - OBT Functional Requirements
    - Autonomous Mode
      - [SAVOIR.OBC.OBT.100] In the Auton
      - [SAVOIR.OBC.OBT.110] It shall be po
    - Synchonized Mode
    - [SAVOIR.OBC.OBT.50] The OBT function
    - [SAVOIR.OBC.OBT.60] The time counter
    - [SAVOIR.OBC.OBT.70] The OBC OBT wra
    - [SAVOIR.OBC.OBT.80] The OBC OBT fun
    - [SAVOIR.OBC.OBT.90] At OBC power-on



[Requirement] [SAVOIR.OBC.OBT.30] It shall be possible to select Master / Slave OBT either via...

Requirements VP
Requirements Allocation
Internal Requirements Allocation
Expert

Property

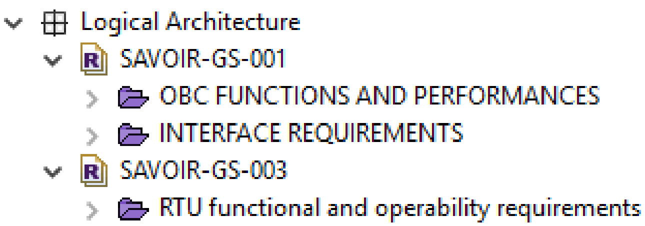
Long name :	
Name :	Selecting Master OBT
Chapter name :	
Prefix :	
Text :	It shall be possible to select Master / Slave OBT either via CPDU Pulse Command or directly by the ASW.

In properties, the text of the requirement is present

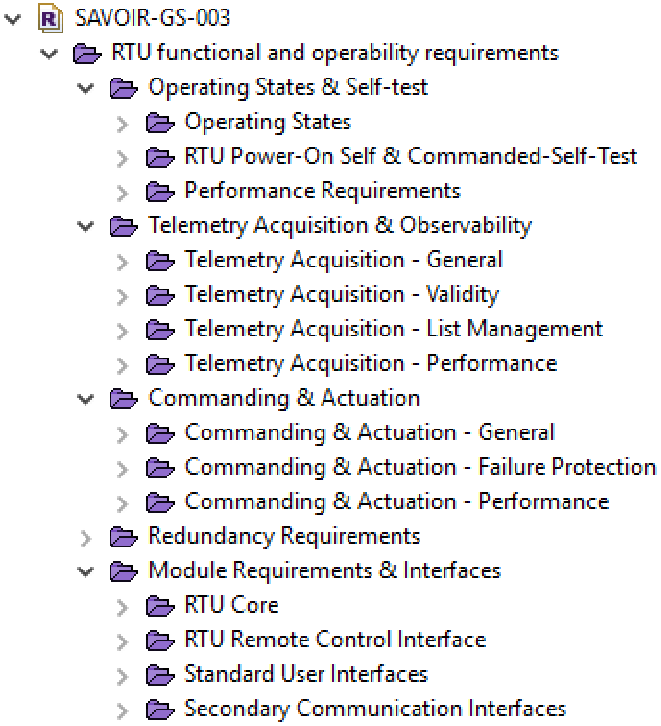
Requirement allocation enable to map it on a logical function or a functional exchange or any other element.

RTU REQUIREMENTS

/// RTU Requirements imported in the model

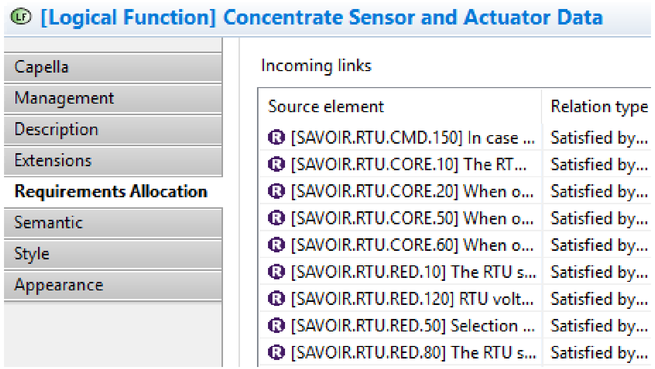


- Logical Architecture
  - SAVOIR-GS-001
    - OBC FUNCTIONS AND PERFORMANCES
    - INTERFACE REQUIREMENTS
  - SAVOIR-GS-003
    - RTU functional and operability requirements



- SAVOIR-GS-003
  - RTU functional and operability requirements
    - Operating States & Self-test
      - Operating States
      - RTU Power-On Self & Commanded-Self-Test
      - Performance Requirements
    - Telemetry Acquisition & Observability
      - Telemetry Acquisition - General
      - Telemetry Acquisition - Validity
      - Telemetry Acquisition - List Management
      - Telemetry Acquisition - Performance
    - Commanding & Actuation
      - Commanding & Actuation - General
      - Commanding & Actuation - Failure Protection
      - Commanding & Actuation - Performance
    - Redundancy Requirements
    - Module Requirements & Interfaces
      - RTU Core
      - RTU Remote Control Interface
      - Standard User Interfaces
      - Secondary Communication Interfaces

/// RTU Requirements allocated at the logical layer



[Logical Function] Concentrate Sensor and Actuator Data

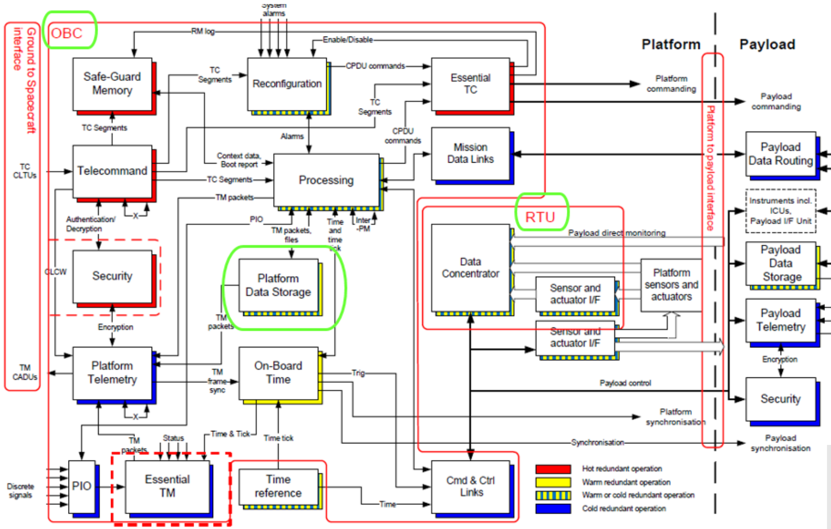
- Capella
- Management
- Description
- Extensions
- **Requirements Allocation**
- Semantic
- Style
- Appearance

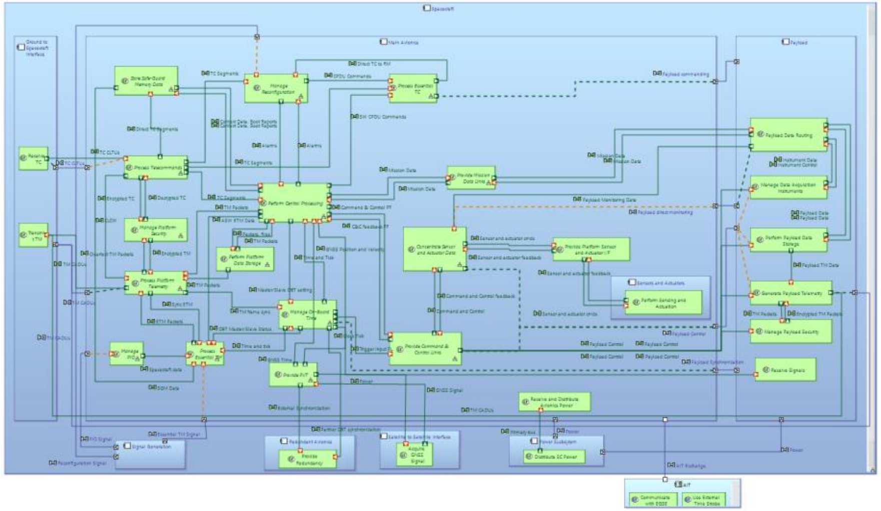
Incoming links

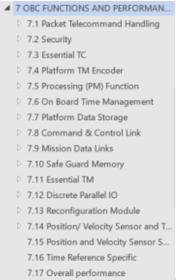
Source element	Relation type
[SAVOIR.RTU.CMD.150] In case ...	Satisfied by...
[SAVOIR.RTU.CORE.10] The RT...	Satisfied by...
[SAVOIR.RTU.CORE.20] When o...	Satisfied by...
[SAVOIR.RTU.CORE.50] When o...	Satisfied by...
[SAVOIR.RTU.CORE.60] When o...	Satisfied by...
[SAVOIR.RTU.RED.10] The RTU s...	Satisfied by...
[SAVOIR.RTU.RED.120] RTU volt...	Satisfied by...
[SAVOIR.RTU.RED.50] Selection ...	Satisfied by...
[SAVOIR.RTU.RED.80] The RTU s...	Satisfied by...

MODEL OF THE SAVOIR FUNCTIONAL REFERENCE ARCHITECTURE

/// SAVOIR-TN-001: Functional Reference Architecture





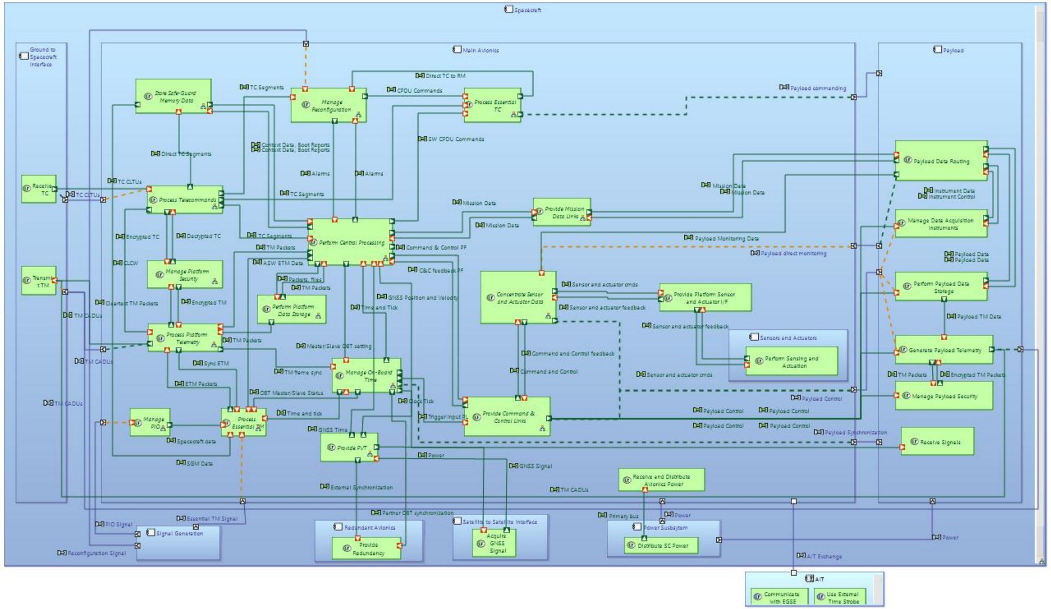


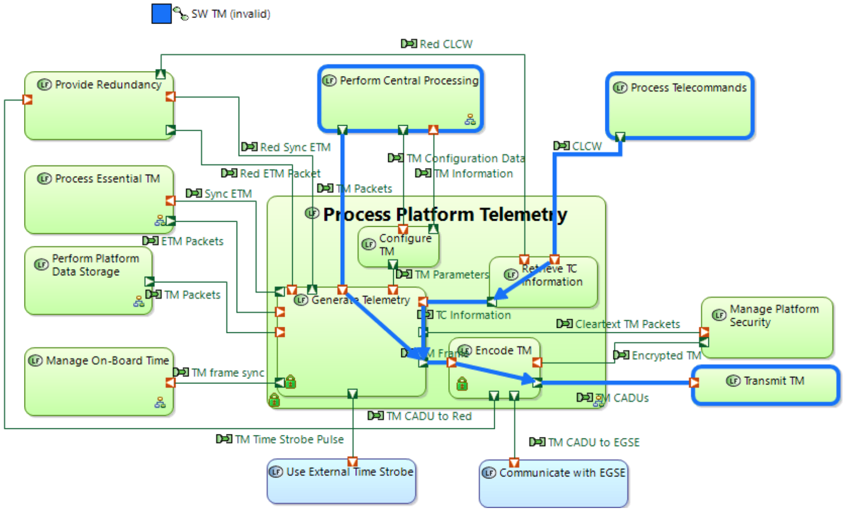
- 7 OBC FUNCTIONS AND PERFORMAN...
  - 7.1 Packet Telecommand Handling
  - 7.2 Security
  - 7.3 Essential TC
  - 7.4 Platform TM Encoder
  - 7.5 Processing (PM) Function
  - 7.6 On Board Time Management
  - 7.7 Platform Data Storage
  - 7.8 Command & Control Link
  - 7.9 Mission Data Links
  - 7.10 Safe Guard Memory
  - 7.11 Essential TM
  - 7.12 Discrete Parallel IO
  - 7.13 Reconfiguration Module
  - 7.14 Position/ Velocity Sensor and T...
  - 7.15 Position and Velocity Sensor S...
  - 7.16 Time Reference Specific
  - 7.17 Overall performance

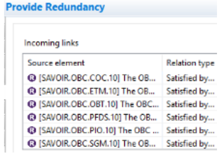
MODELLING AT LOGICAL LAYER

Overall Avionics Diagram

Specific Functional Chain for one avionics function



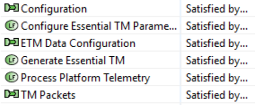




Provide Redundancy

Incoming links

Source element	Relation type
[SAVOIR.OBC.COC.10] The OB...	Satisfied by...
[SAVOIR.OBC.ETM.10] The OB...	Satisfied by...
[SAVOIR.OBC.OBT.10] The OBC...	Satisfied by...
[SAVOIR.OBC.PFDS.10] The OB...	Satisfied by...
[SAVOIR.OBC.PIO.10] The OBC ...	Satisfied by...
[SAVOIR.OBC.SGM.10] The OB...	Satisfied by...



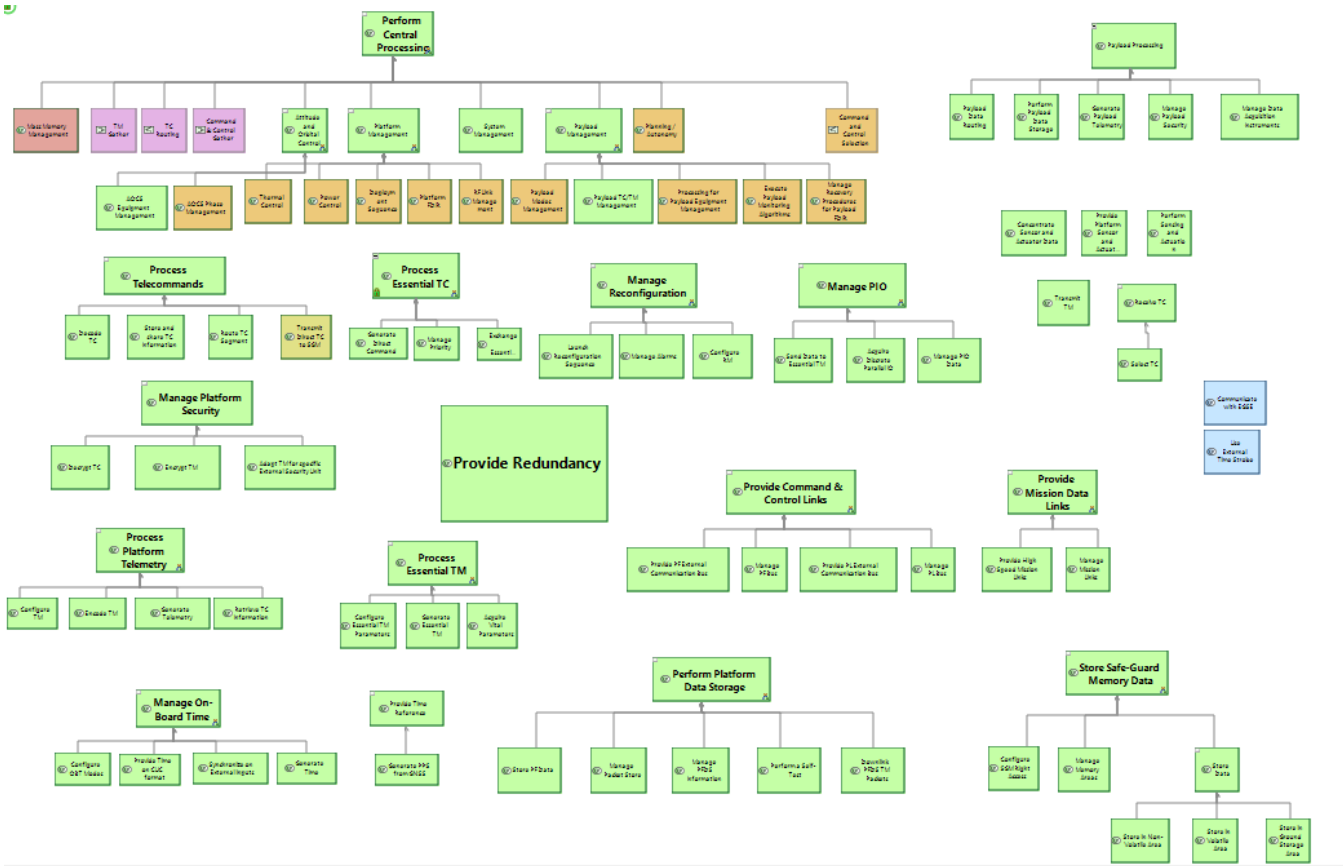
Configuration	Satisfied by...
Configure Essential TM Parame...	Satisfied by...
ETM Data Configuration	Satisfied by...
Generate Essential TM	Satisfied by...
Process Platform Telemetry	Satisfied by...
TM Packets	Satisfied by...

FUNCTIONS ANALYSIS

/// Detail of the functions

/// TAS to worked on the method:

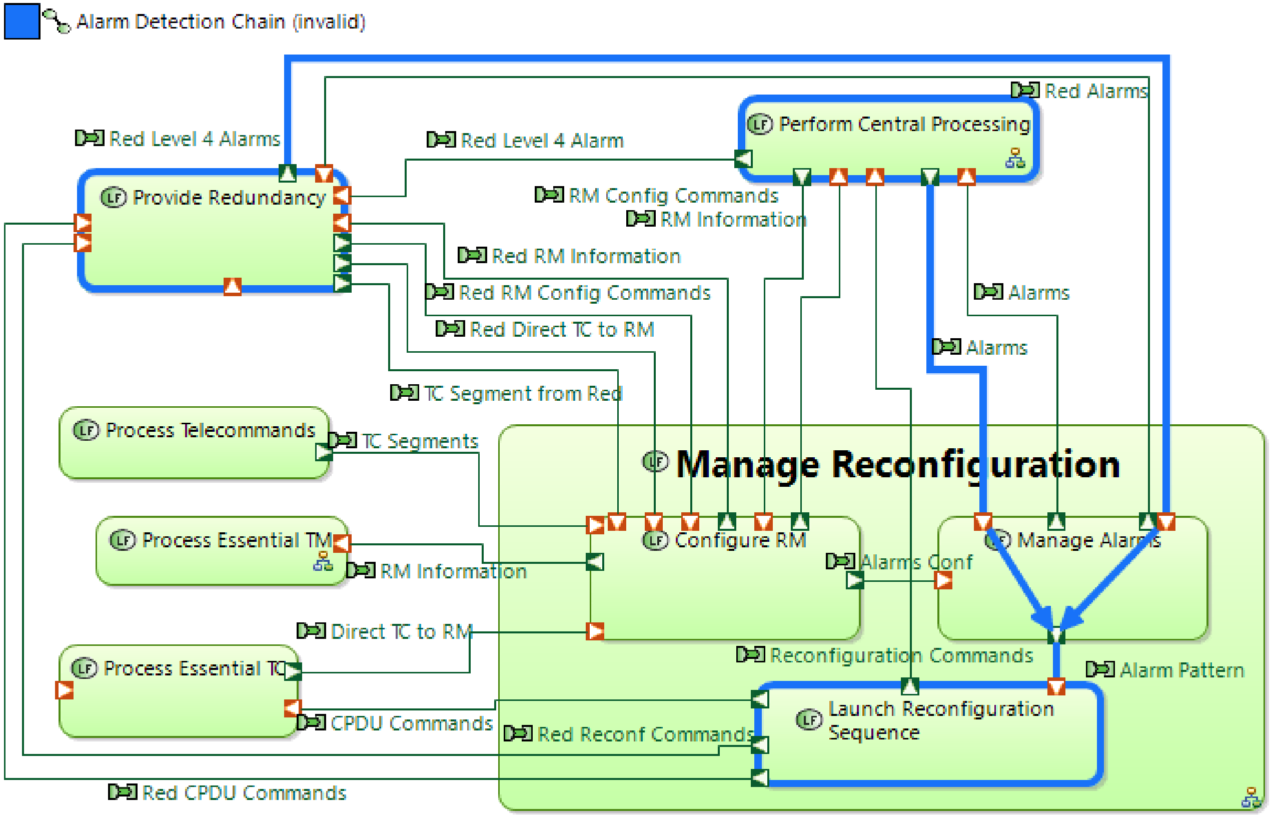
- Start from one section of Generic OBC Spec
- Take the “mother” functions related to that section
- Go through the requirements one by one and check if the requirement is fulfilled by a logical function or a functional exchange.



ThalesAlenia
a Thales / Leonardo company Space

RECONFIGURATION MODULE

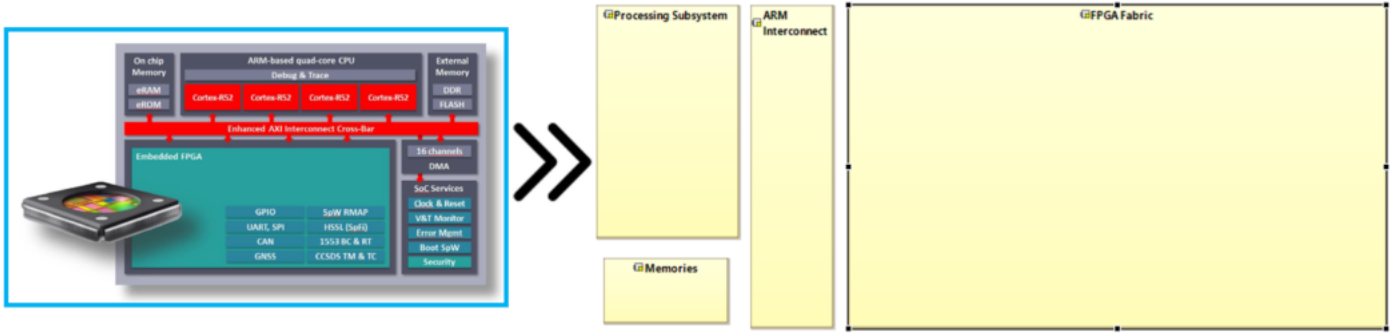
/// Alarm Detection



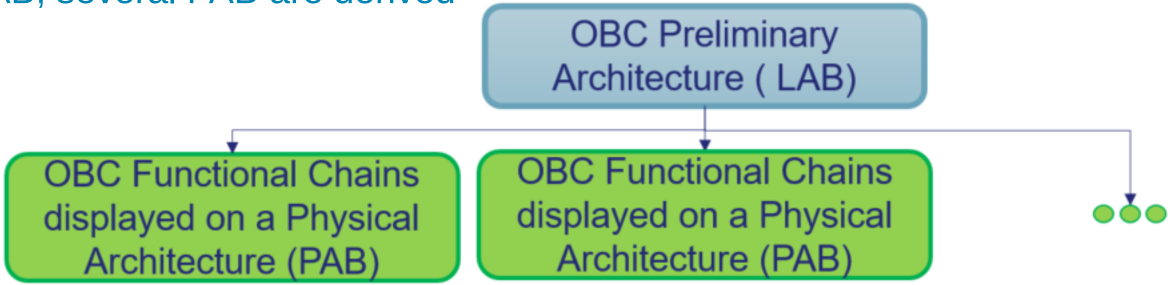
PHYSICAL MODELLING BASED ON SAVOIR

/// OBJECTIVE: **Create a Main Avionics [REC] library for a general purpose OBC** containing all its main Functional Blocks (as Behaviour Components)

/// Definition of physical nodes in the avionics and allocation of physical functions chain by chain



/// From the LAB, several PAB are derived

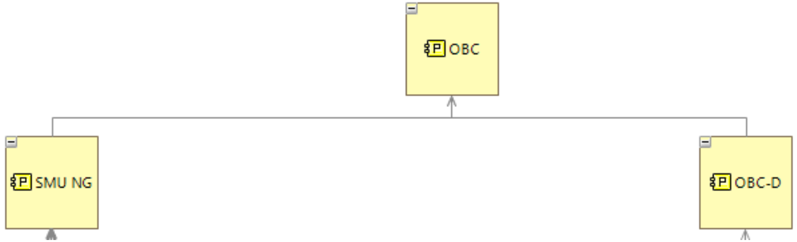

OBC Preliminary Architecture (LAB)

OBC Functional Chains displayed on a Physical Architecture (PAB)

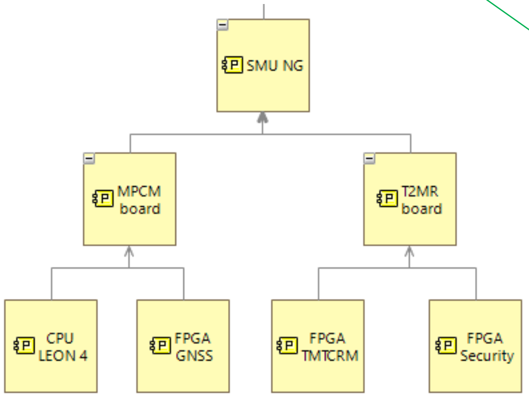
OBC Functional Chains displayed on a Physical Architecture (PAB)

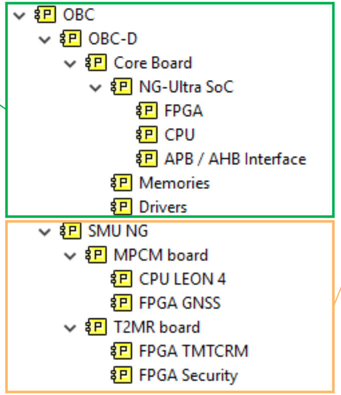
PHYSICAL NODES

/// Two different products



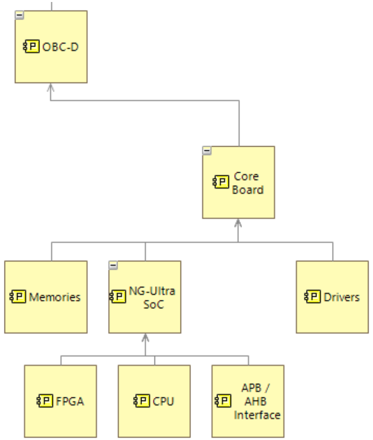
SMU NG / IPAC
(Processor + FPGA)





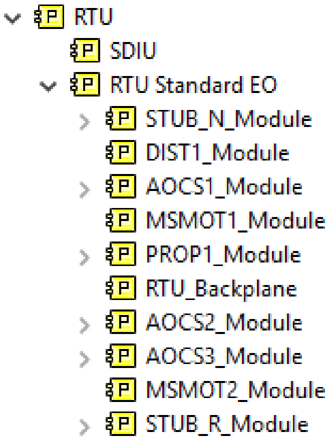
- OBC
  - OBC-D
    - Core Board
      - NG-Ultra SoC
        - FPGA
        - CPU
        - APB / AHB Interface
      - Memories
      - Drivers
  - SMU NG
    - MPCM board
      - CPU LEON 4
      - FPGA GNSS
    - T2MR board
      - FPGA TMTCRM
      - FPGA Security

SoC based Computer

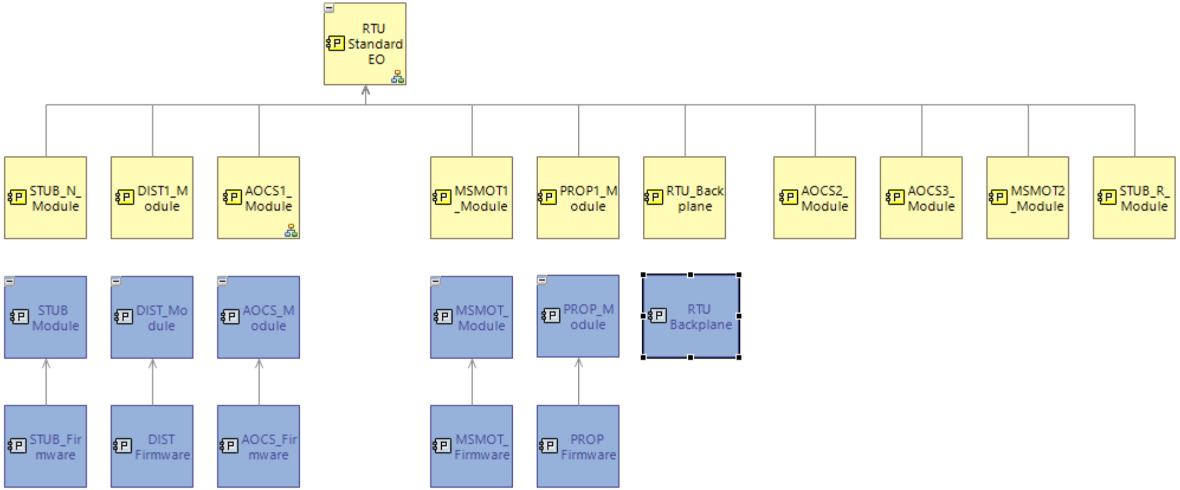


OTHER PHYSICAL NODES

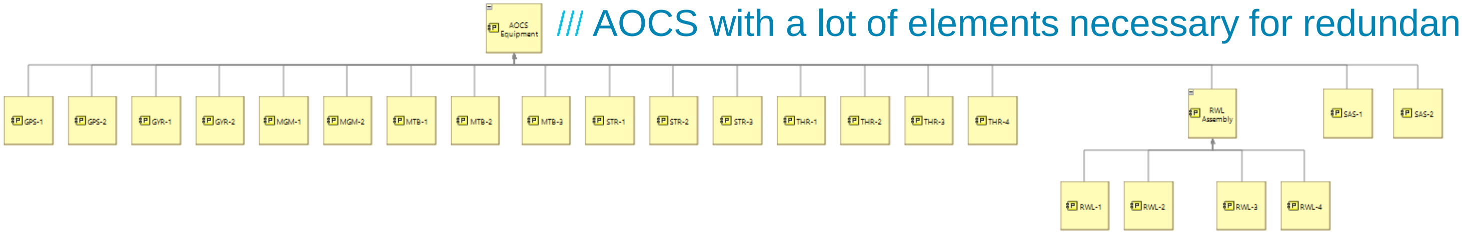
/// RTU detailed with 2 different products. Only one generic is presented



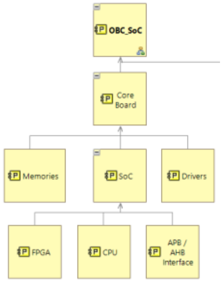
- RTU
  - SDIU
  - RTU Standard EO
    - STUB_N_Module
    - DIST1_Module
    - AOCS1_Module
    - MSMOT1_Module
    - PROP1_Module
    - RTU_Backplane
    - AOCS2_Module
    - AOCS3_Module
    - MSMOT2_Module
    - STUB_R_Module



/// AOCS with a lot of elements necessary for redundancy in RTU diagrams

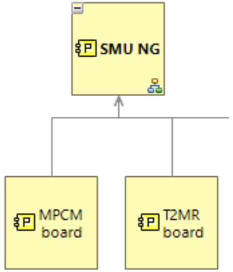


OBC MODELS



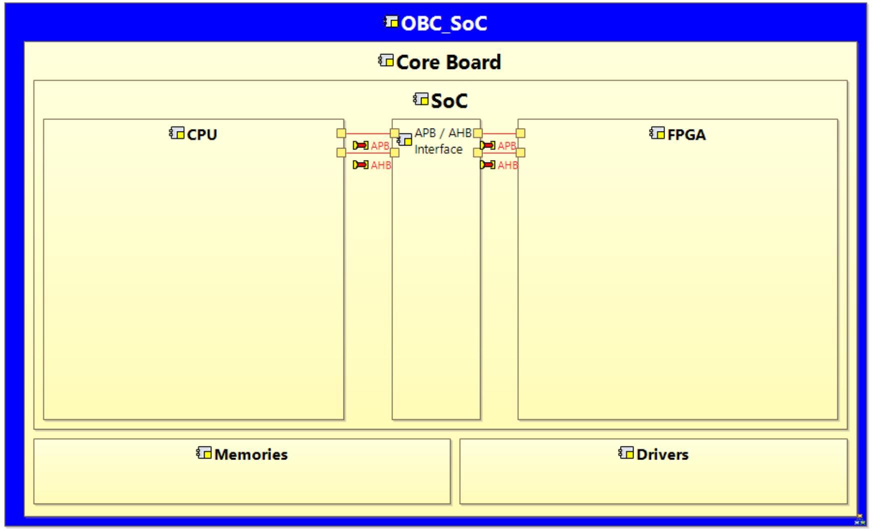
/// All the OBC are on the same model:

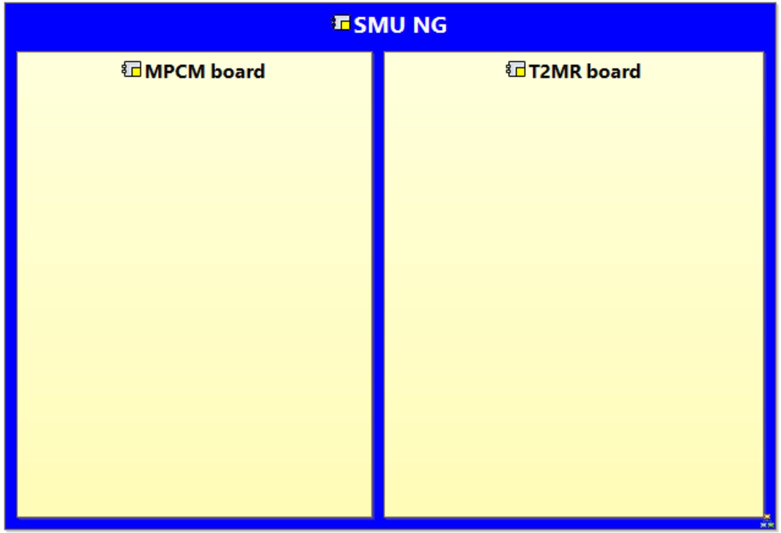
- OBC Soc
- OBC SMU NG



One single board
One single chip

Two boards
Two chips





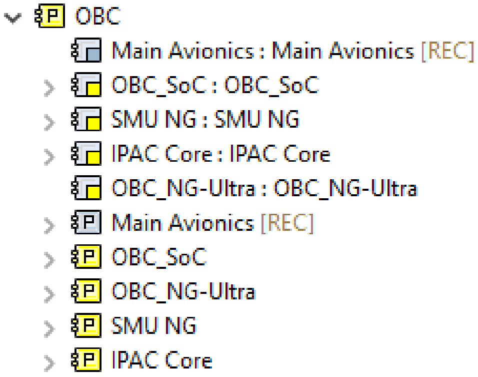
MODELS ORGANISATION

/// Model based on same logical components and on same functions

2 Kinds of Designs → 1 Model

/// 1 Instantiation by kind of design, the two kinds of OBC are instantiated on TAS computers:

- OBC Soc is instantiated in OBC including the NG-Ultra SoC
- SMU NG is instantiated in IPAC including the GR740 and the RTG4

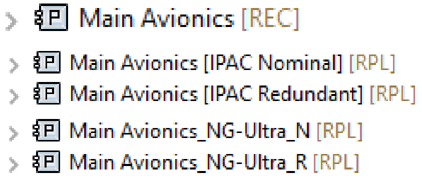


```
OBC
    Main Avionics : Main Avionics [REC]
  > OBC_SoC : OBC_SoC
  > SMU NG : SMU NG
  > IPAC Core : IPAC Core
    OBC_NG-Ultra : OBC_NG-Ultra
  > Main Avionics [REC]
  > OBC_SoC
  > OBC_NG-Ultra
  > SMU NG
  > IPAC Core
```

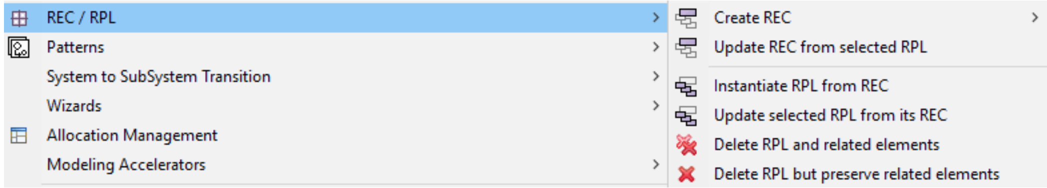
[REC/RPL] FOR BEHAVIOR COMPONENTS

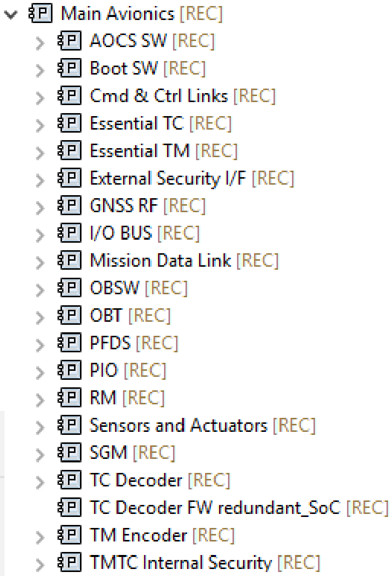
/// The two OBC are instantiated on TAS computers:

- OBC Soc is instantiated in OBC including the NG-Ultra SoC
- SMU NG is instantiated in IPAC including the GR740 and the RTG4
- Use of Rec/RPL in Capella
  - **REC** is like a general purpose library
  - **RPL** are the replica of the library
  - RPL are useful for **Redundancy** and for **Allocation** on different instantiation

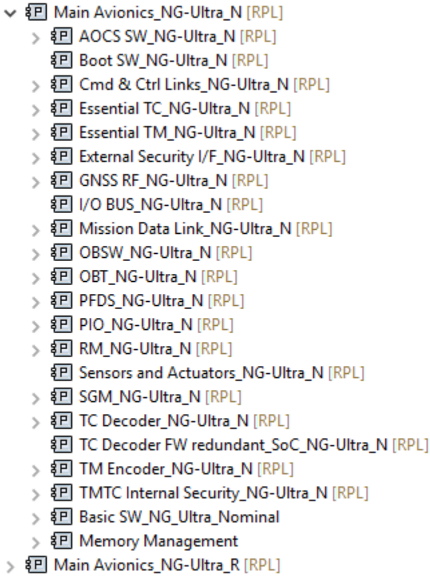


Main Avionics [REC]
Main Avionics [IPAC Nominal] [RPL]
Main Avionics [IPAC Redundant] [RPL]
Main Avionics_NG-Ultra_N [RPL]
Main Avionics_NG-Ultra_R [RPL]


REC / RPL
Patterns
System to SubSystem Transition
Wizards
Allocation Management
Modeling Accelerators
Create REC
Update REC from selected RPL
Instantiate RPL from REC
Update selected RPL from its REC
Delete RPL and related elements
Delete RPL but preserve related elements



Main Avionics [REC]
AOCS SW [REC]
Boot SW [REC]
Cmd & Ctrl Links [REC]
Essential TC [REC]
Essential TM [REC]
External Security I/F [REC]
GNSS RF [REC]
I/O BUS [REC]
Mission Data Link [REC]
OBSW [REC]
OBT [REC]
PFDS [REC]
PIO [REC]
RM [REC]
Sensors and Actuators [REC]
SGM [REC]
TC Decoder [REC]
TC Decoder FW redundant_SoC [REC]
TM Encoder [REC]
TMTC Internal Security [REC]

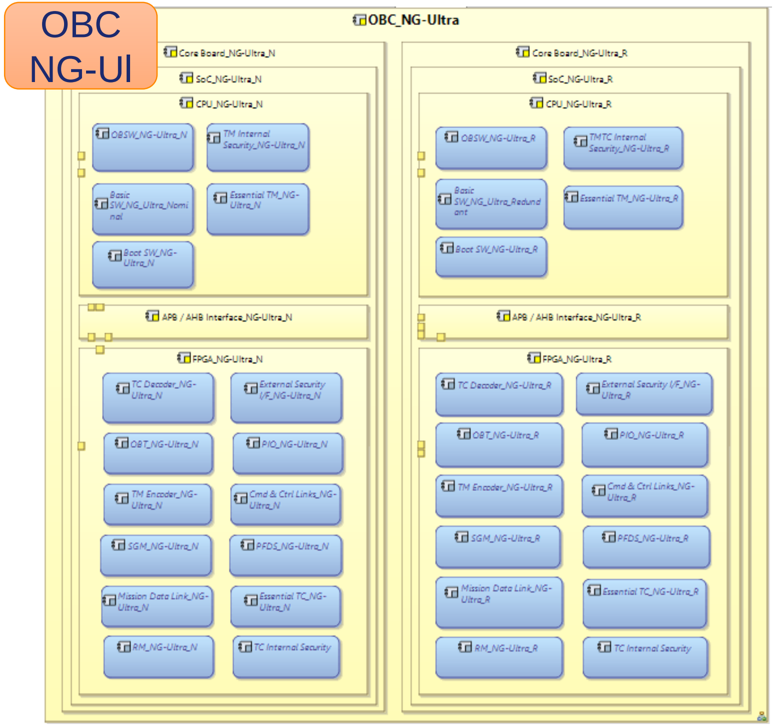


Main Avionics_NG-Ultra_N [RPL]
AOCS SW_NG-Ultra_N [RPL]
Boot SW_NG-Ultra_N [RPL]
Cmd & Ctrl Links_NG-Ultra_N [RPL]
Essential TC_NG-Ultra_N [RPL]
Essential TM_NG-Ultra_N [RPL]
External Security I/F_NG-Ultra_N [RPL]
GNSS RF_NG-Ultra_N [RPL]
I/O BUS_NG-Ultra_N [RPL]
Mission Data Link_NG-Ultra_N [RPL]
OBSW_NG-Ultra_N [RPL]
OBT_NG-Ultra_N [RPL]
PFDS_NG-Ultra_N [RPL]
PIO_NG-Ultra_N [RPL]
RM_NG-Ultra_N [RPL]
Sensors and Actuators_NG-Ultra_N [RPL]
SGM_NG-Ultra_N [RPL]
TC Decoder_NG-Ultra_N [RPL]
TC Decoder FW redundant_SoC_NG-Ultra_N [RPL]
TM Encoder_NG-Ultra_N [RPL]
TMTC Internal Security_NG-Ultra_N [RPL]
Basic SW_NG_Ultra_Nominal
Memory Management
Main Avionics_NG-Ultra_R [RPL]

BEHAVIOR COMPONENTS ALLOCATION

/// 2 different Physical Implementation:

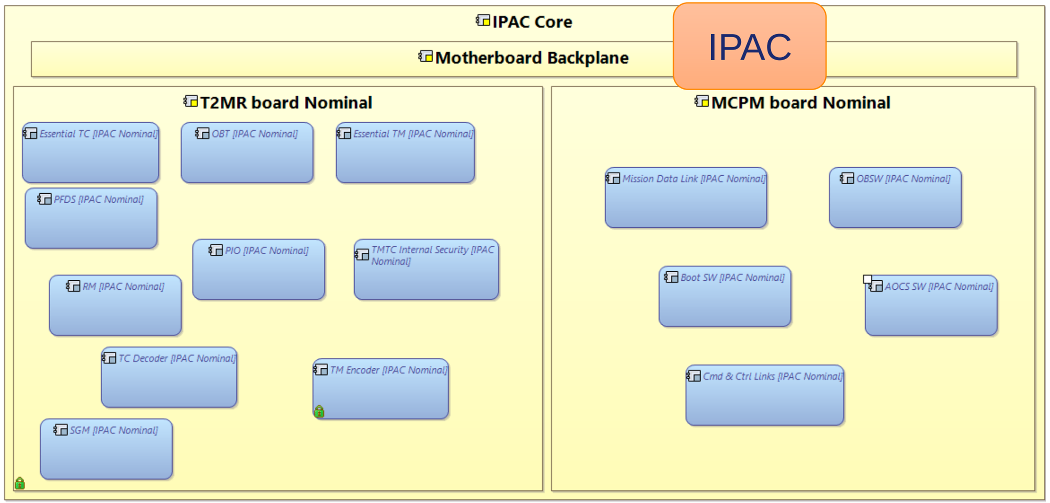
/// Same behavioral components



/// A lot of common use and similar physical allocations.

/// Modifications for some of the Behavior Components:

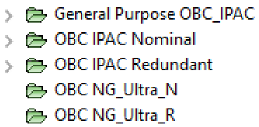
- CMD/CTRL LINKS AND MISSION DATA LINKS
- TMTC INTERNAL SECURITY NOT USED IDENTICALLY
- DIFFERENT ADDITIONAL SW.



PHYSICAL FUNCTIONS MANAGEMENT

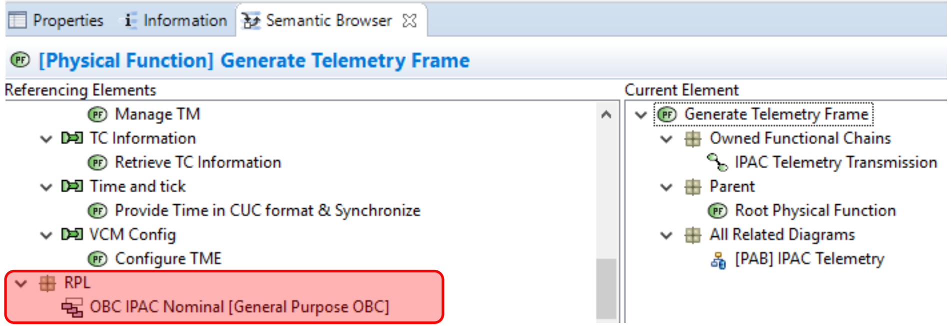
/// Functions are automatically duplicated as RPL (replica) from the REC (recurrent) physical functions

/// They can be stored in a folder reserved for the specific OBC →



- General Purpose OBC_IPAC
- OBC IPAC Nominal
- OBC IPAC Redundant
- OBC NG_Ultra_N
- OBC NG_Ultra_R

/// It is possible to check in the Semantic Browser to which RPL it is belonging to:



[Physical Function] Generate Telemetry Frame

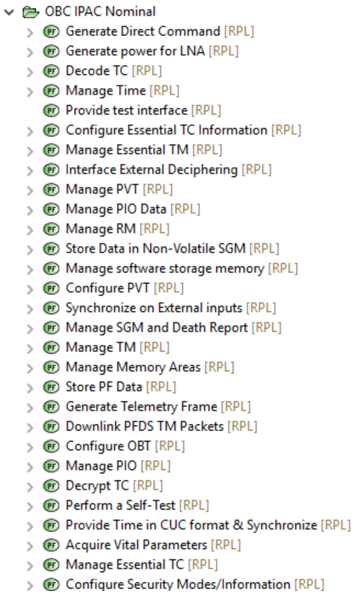
Referencing Elements

- Manage TM
- TC Information
  - Retrieve TC Information
- Time and tick
  - Provide Time in CUC format & Synchronize
- VCM Config
  - Configure TME
- RPL
  - OBC IPAC Nominal [General Purpose OBC]

Current Element

- Generate Telemetry Frame
  - Owned Functional Chains
    - IPAC Telemetry Transmission
  - Parent
    - Root Physical Function
  - All Related Diagrams
    - [PAB] IPAC Telemetry

/// Example of the Physical Functions Replicated →



- OBC IPAC Nominal
  - Generate Direct Command [RPL]
  - Generate power for LNA [RPL]
  - Decode TC [RPL]
  - Manage Time [RPL]
  - Provide test interface [RPL]
  - Configure Essential TC Information [RPL]
  - Manage Essential TM [RPL]
  - Interface External Deciphering [RPL]
  - Manage PVT [RPL]
  - Manage PIO Data [RPL]
  - Manage RM [RPL]
  - Store Data in Non-Volatile SGM [RPL]
  - Manage software storage memory [RPL]
  - Configure PVT [RPL]
  - Synchronize on External inputs [RPL]
  - Manage SGM and Death Report [RPL]
  - Manage TM [RPL]
  - Manage Memory Areas [RPL]
  - Store PF Data [RPL]
  - Generate Telemetry Frame [RPL]
  - Downlink PFDS TM Packets [RPL]
  - Configure OBT [RPL]
  - Manage PIO [RPL]
  - Decrypt TC [RPL]
  - Perform a Self-Test [RPL]
  - Provide Time in CUC format & Synchronize [RPL]
  - Acquire Vital Parameters [RPL]
  - Manage Essential TC [RPL]
  - Configure Security Modes/Information [RPL]

GENERAL DIAGRAM

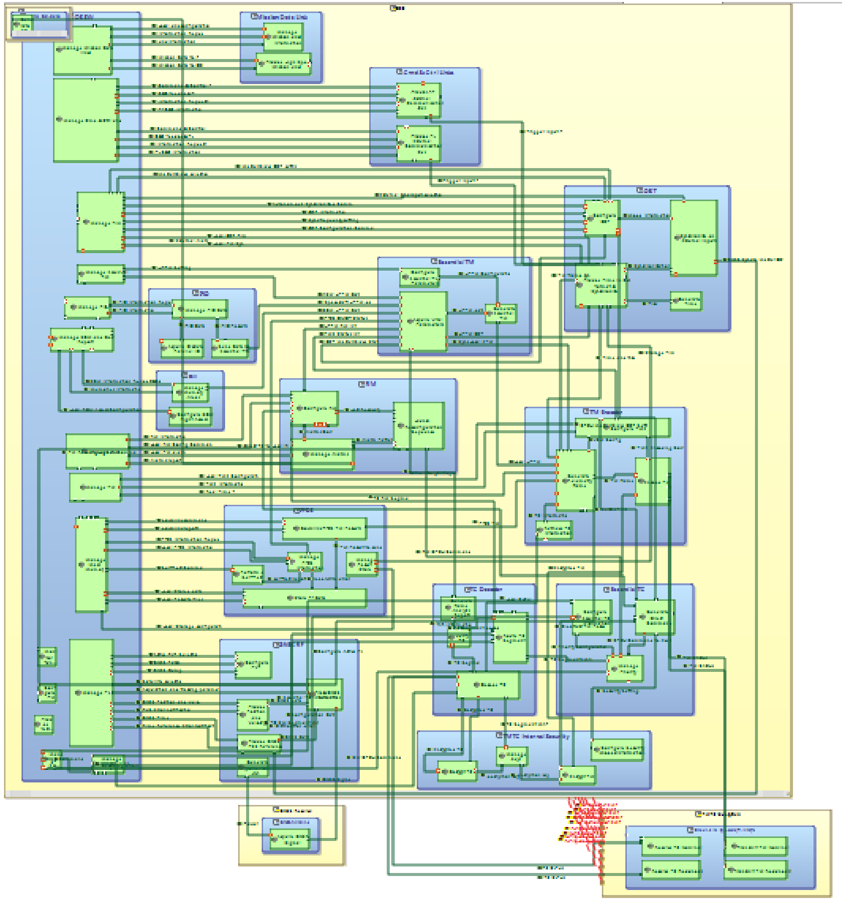
/// One General Design

/// Then design functional chain by functional chain

/// Basis for the REC with:

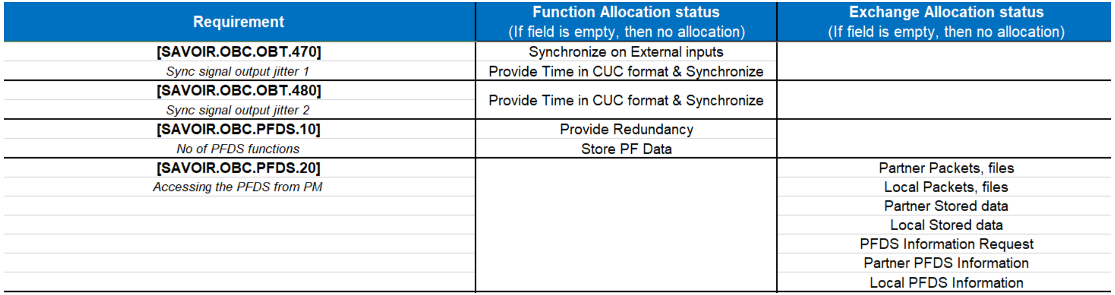
- Behavioral component
- Physical functions
- On the OBC REC

/// Detailed view of the functions with functional chains.



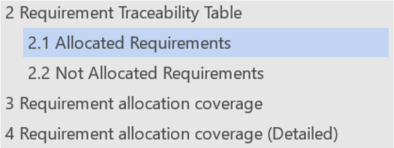
WAY FORWARD: DEVELOP REQUIREMENT TRACEABILITY REPORT

/// Excel File

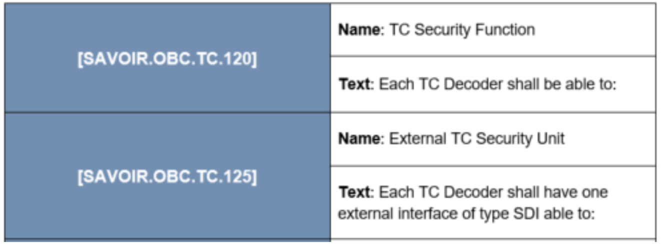


Requirement	Function Allocation status (If field is empty, then no allocation)	Exchange Allocation status (If field is empty, then no allocation)
[SAVOIR.OBC.OBT.470]	Synchronize on External inputs	
Sync signal output jitter 1	Provide Time in CUC format & Synchronize	
[SAVOIR.OBC.OBT.480]	Provide Time in CUC format & Synchronize	
Sync signal output jitter 2		
[SAVOIR.OBC.PFDS.10]	Provide Redundancy	
No of PFDS functions	Store PF Data	
[SAVOIR.OBC.PFDS.20]		Partner Packets, files
Accessing the PFDS from PM		Local Packets, files
		Partner Stored data
		Local Stored data
		PFDS Information Request
		Partner PFDS Information
		Local PFDS Information

/// Word File

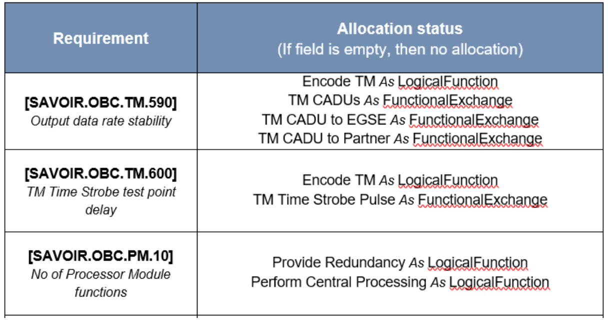


Description



[SAVOIR.OBC.TC.120]	**Name**: TC Security Function
	Text: Each TC Decoder shall be able to:
[SAVOIR.OBC.TC.125]	**Name**: External TC Security Unit
	Text: Each TC Decoder shall have one external interface of type SDI able to:

Matrix



Requirement	Allocation status (If field is empty, then no allocation)
[SAVOIR.OBC.TM.590] *Output data rate stability*	Encode TM *As* LogicalFunction TM CADUs *As* FunctionalExchange TM CADU to EGSE *As* FunctionalExchange TM CADU to Partner *As* FunctionalExchange
[SAVOIR.OBC.TM.600] *TM Time Strobe test point delay*	Encode TM *As* LogicalFunction TM Time Strobe Pulse *As* FunctionalExchange
[SAVOIR.OBC.PM.10] *No of Processor Module functions*	Provide Redundancy *As* LogicalFunction Perform Central Processing *As* LogicalFunction

EXTEND TO COMMAND CONTROL

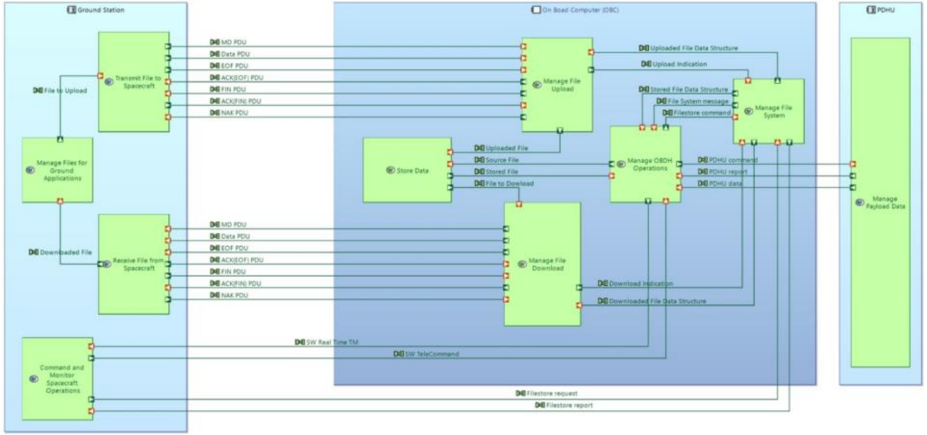
/// Improve the model with Command/Control (CFDP, DTN) including Ground Station.

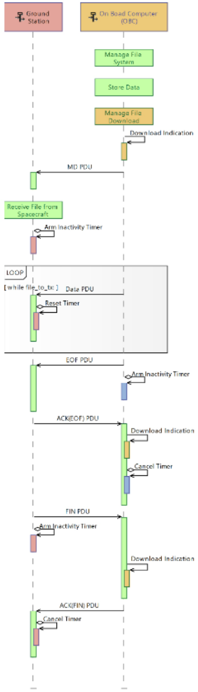
/// Example with CFDP

/// New functions and functional exchange: Ground to Board

/// ESA Operations with ESOC

/// Detailed sequence



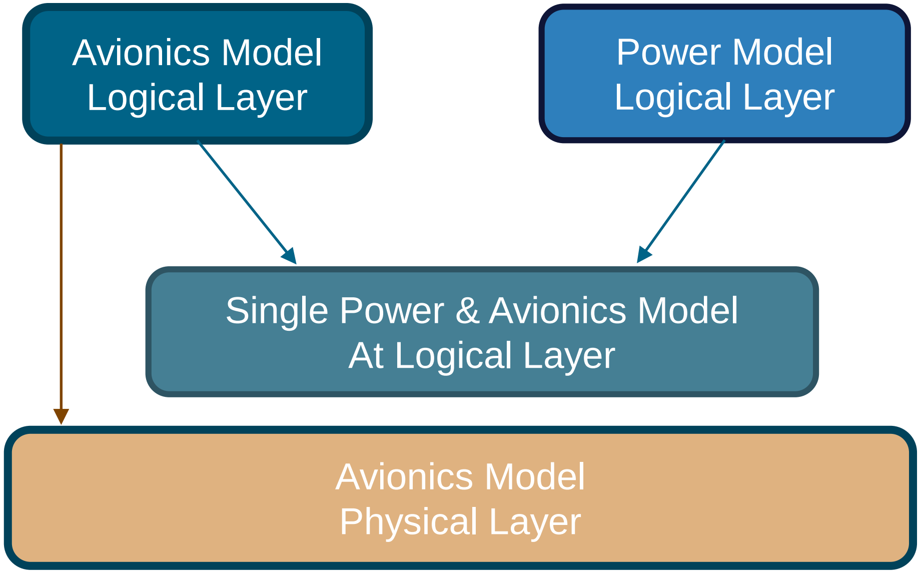


EXTEND TO OTHER SUBSYSTEMS

/// Merging is fast in the model, coherency takes longer.

/// Only at Logical Layer because first, subsystem models shall be done at Logical Layer

/// Demonstrator for other avionics subsystems (future work): AOCS, TCS,…

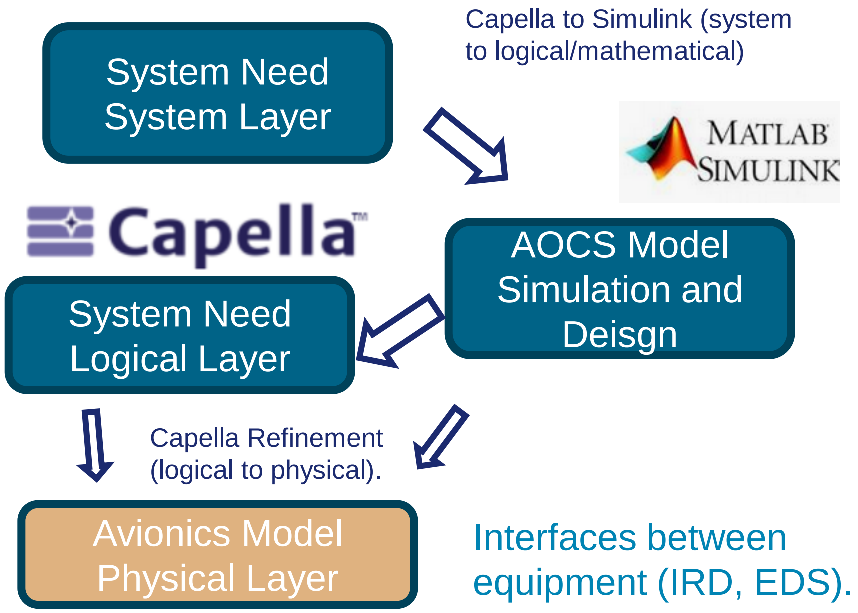


Avionics Model Logical Layer

Power Model Logical Layer

Single Power & Avionics Model At Logical Layer

Avionics Model Physical Layer



Capella to Simulink (system to logical/mathematical)

System Need System Layer

AOCS Model Simulation and Deisgn

System Need Logical Layer

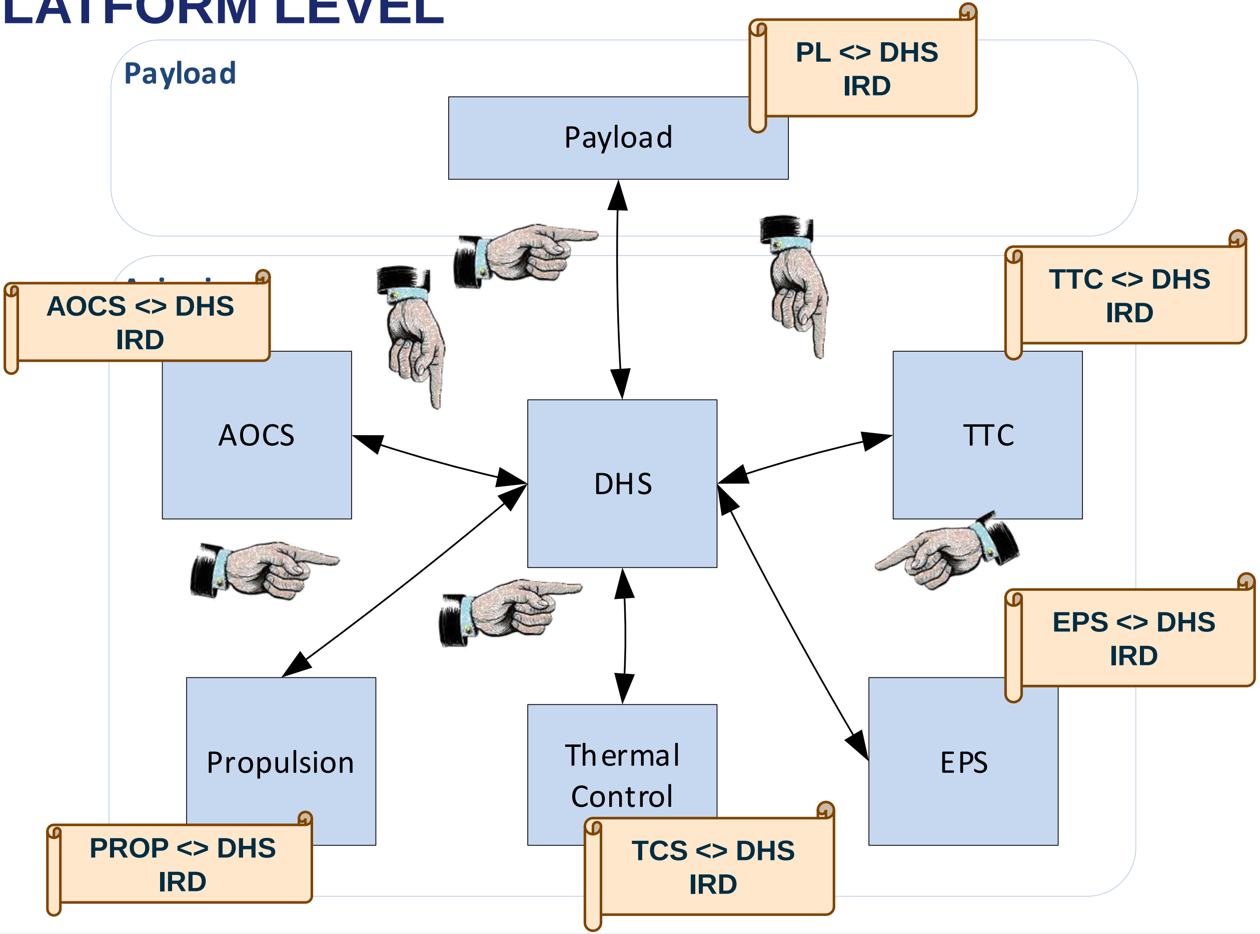
Capella Refinement (logical to physical).

Avionics Model Physical Layer

Interfaces between equipment (IRD, EDS).

WAY FORWARD AT PLATFORM LEVEL

- /// Target for Standardization
- /// Interconnecting subsystems:
  - Used tools for interfacing subsystems
  - Interface Requirement Documents based on models
  - Details of the Electronics Data Sheet only with models.



CONCLUSION

SUMMARY & MAIN MESSAGES

/// Modelling : Logical Layer to Physical layer

- Savoir Functional reference Architecture
- Functional Chains Highlight in the main Avionics functions
- Logical Layer derived in several Physical functions mapped on Physical components

/// Traceability of requirements.

- Savoir Specifications Imported in the Capella Model.
- Functional Requirements Mapped to the Logical layer of the model.
- Automatic generation of Traceability Matrix

/// Physical Layer REC/RPL approach with 2 implementations

- Example for 2 OBC
- Example for RTU in different configurations

/// Tool well suited for Update of the Model / Maintenance.

QUESTIONS?