

Streamlining system-safety engineering with digital technologies

Software Product Assurance Workshop
European Space Astronomy Centre, Spain
26-28th September 2023

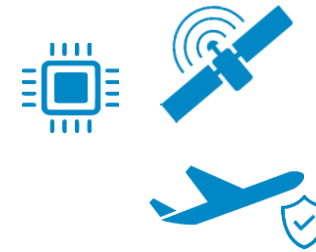


| Last updated: 08/09/2023

Who are we?

The company

- ✓ Spanish SME - Founded in 2019 – ESA BIC Madrid
- ✓ Team of 30 system /RAMS /MBSE engineers



Specialization

- ✓ Complex electronics
- ✓ Safety Critical Systems
- ✓ Autonomous & software defined systems



System, safety and reliability experts

- ✓ Specialization in complying with the highest quality standards for safety/availability critical missions



Digitalization of systems engineering

- ✓ Development and extension of model-based software tools for digitalization of the system & safety engineering process

Network of incubators supporting tech & space start-ups in Europe

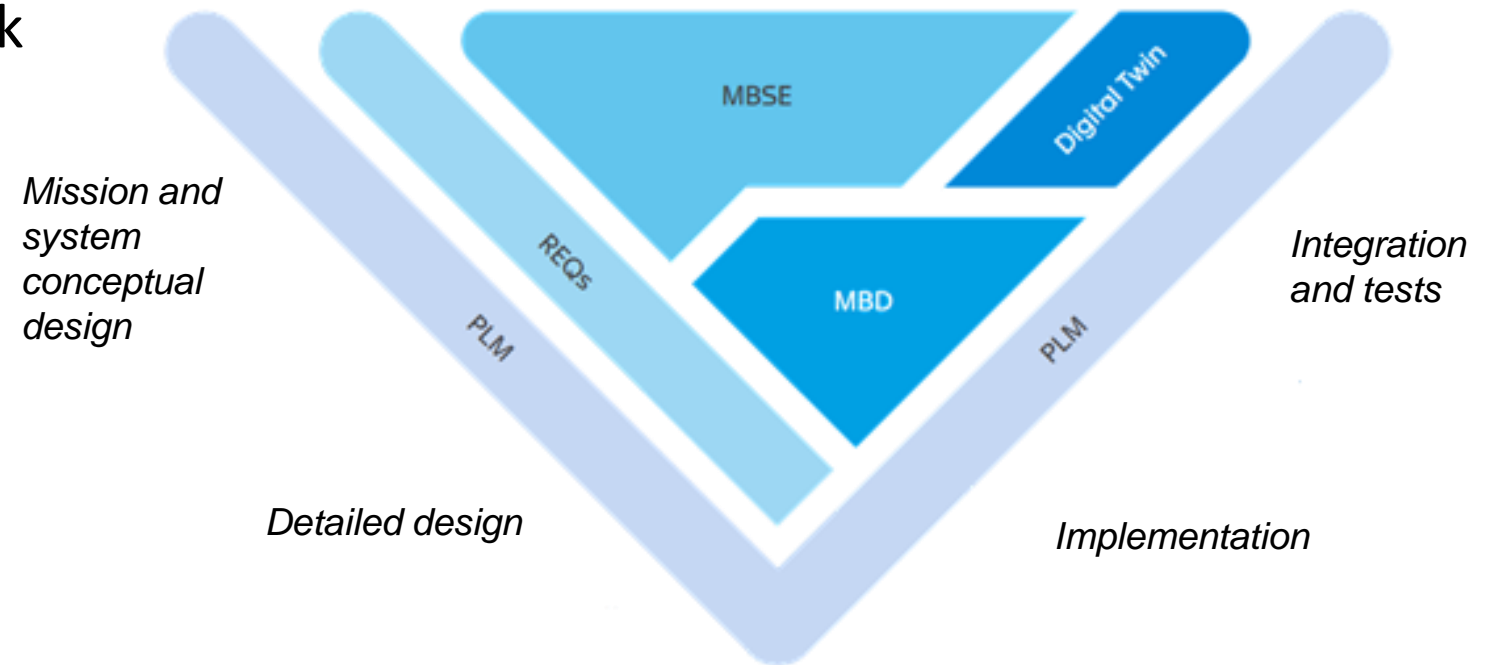


- Partnership between ESA and local institutions to support entrepreneurship
- Initial grant and commercial and technical assistance to launch the company
- Focus on technology transfer from space to other sectors (spin-off) and from other sectors to space (spin-in)
- **Anzen** joined ESA-BIC Madrid in 2019 with the mission of improving **safety & reliability** engineering in the **space** and **advanced air mobility** sectors



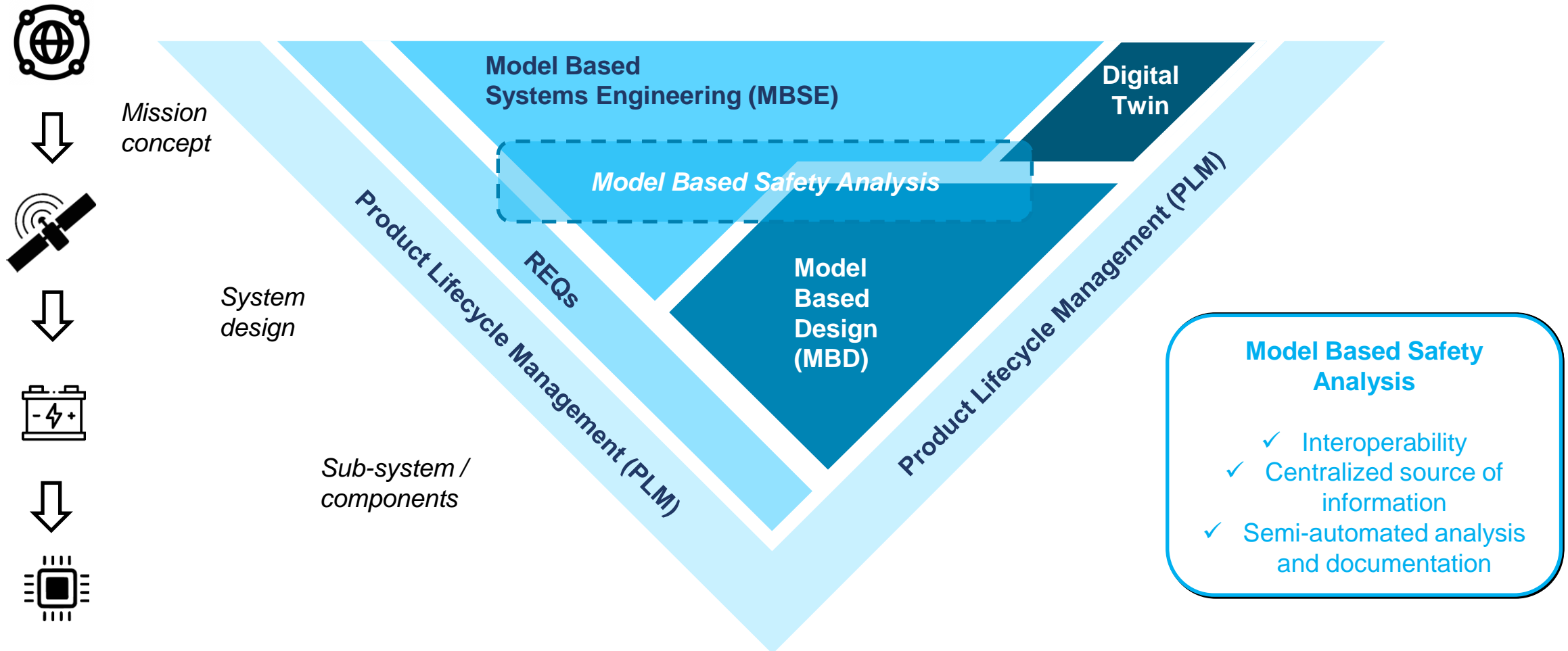
Outline

- Introduction to digital engineering
 - The systems engineering process
 - Framework and tools
- Safety and dependability analysis
- Wrap-up and future work



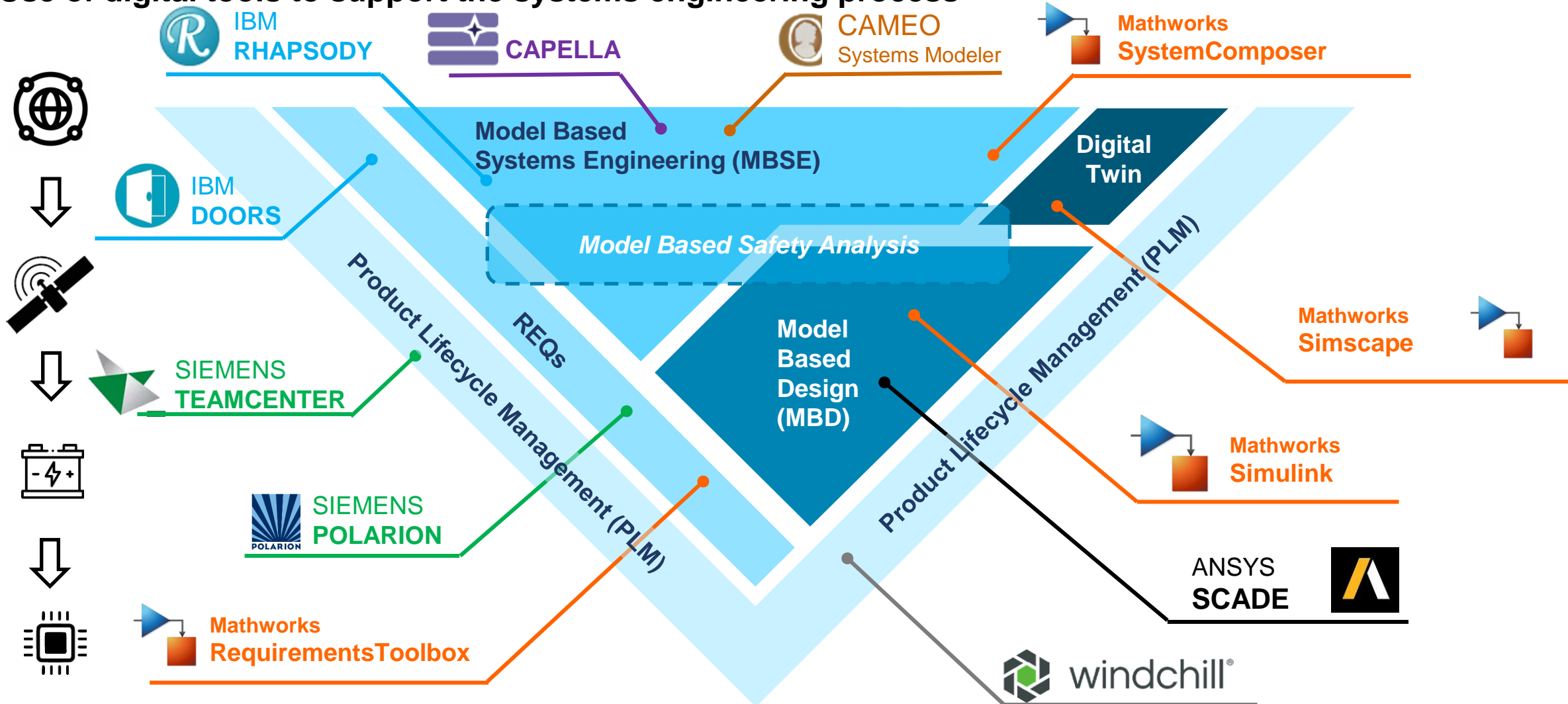
The systems engineering process

Use of digital tools to support the systems engineering process



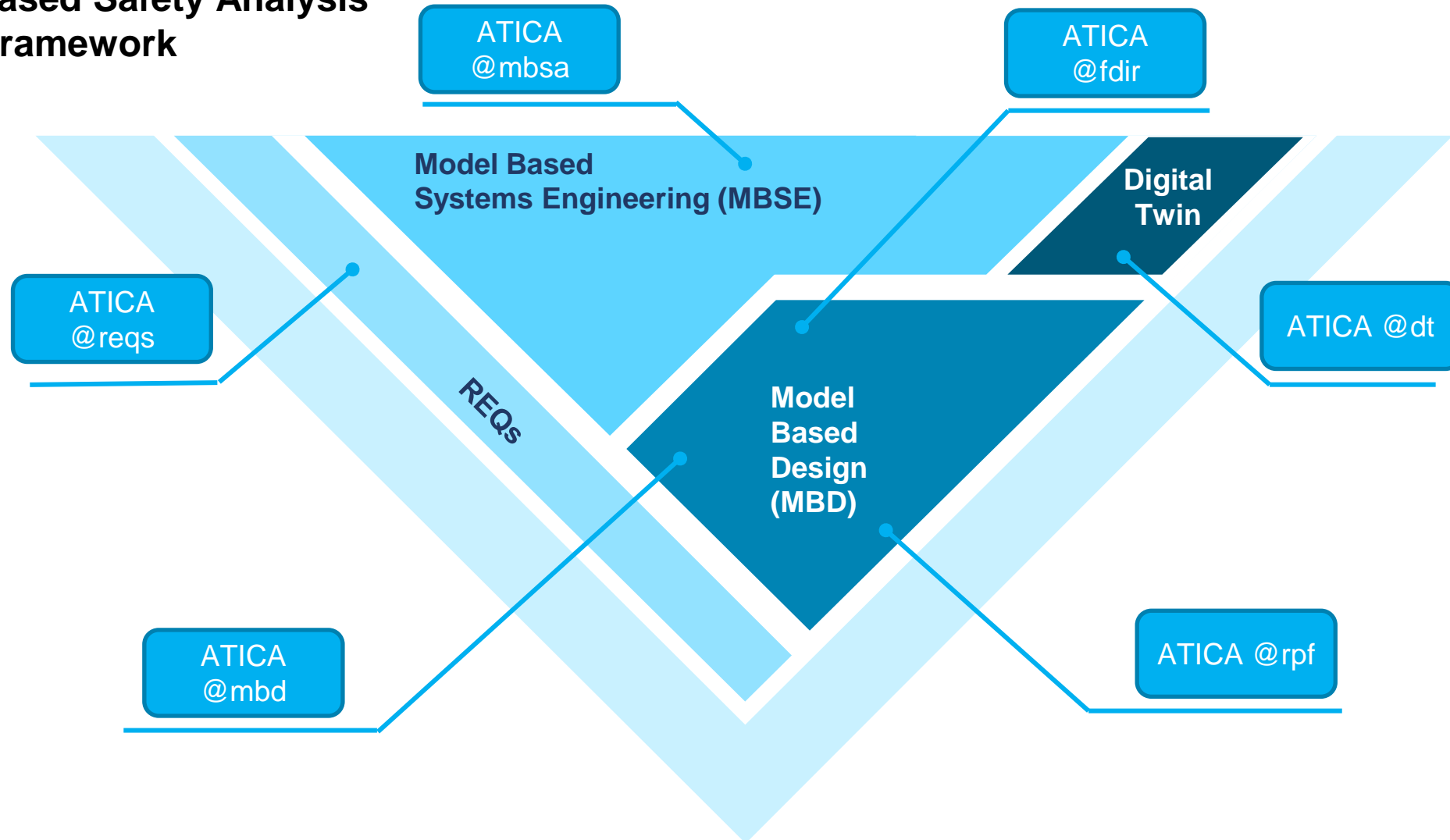
Framework and tools

Use of digital tools to support the systems engineering process

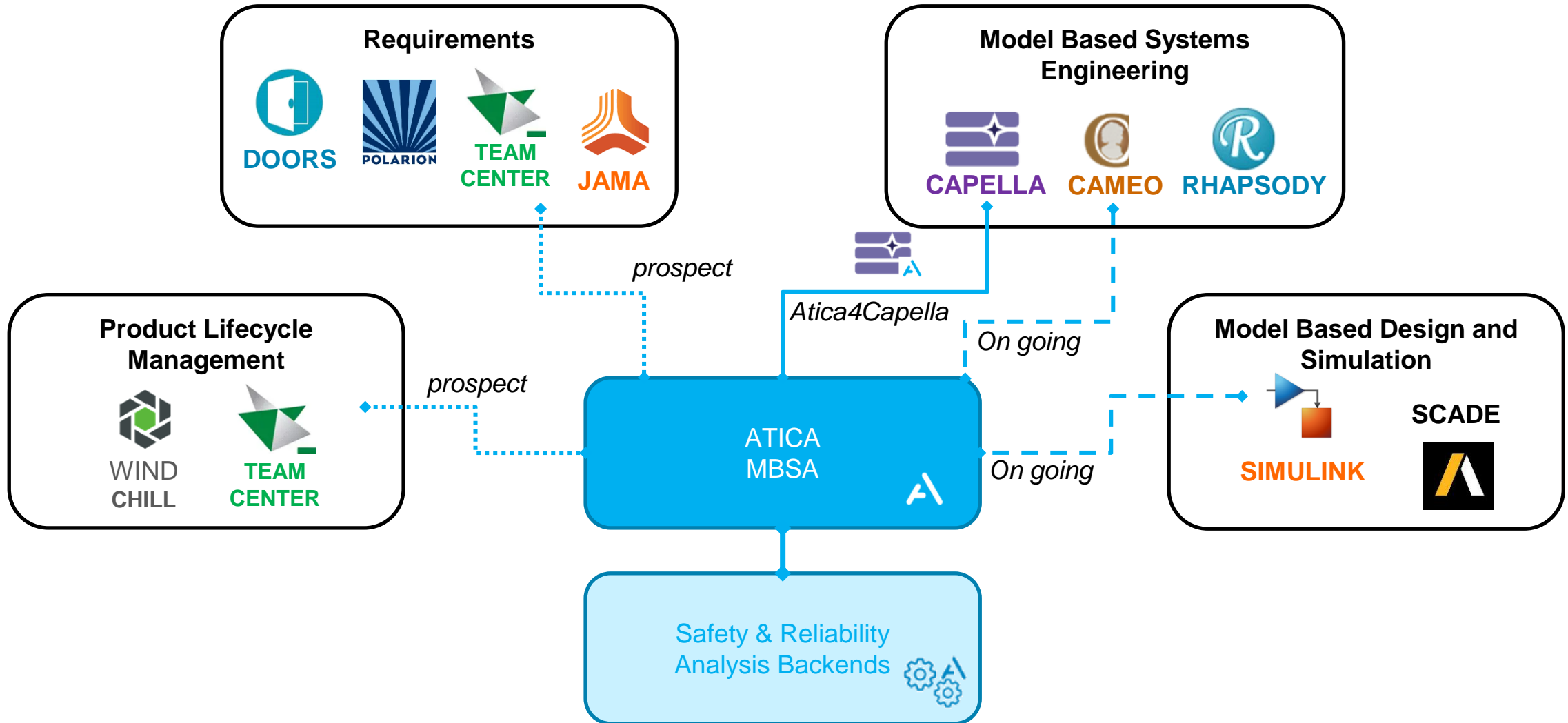


Framework and tools

Model Based Safety Analysis ATICA Framework



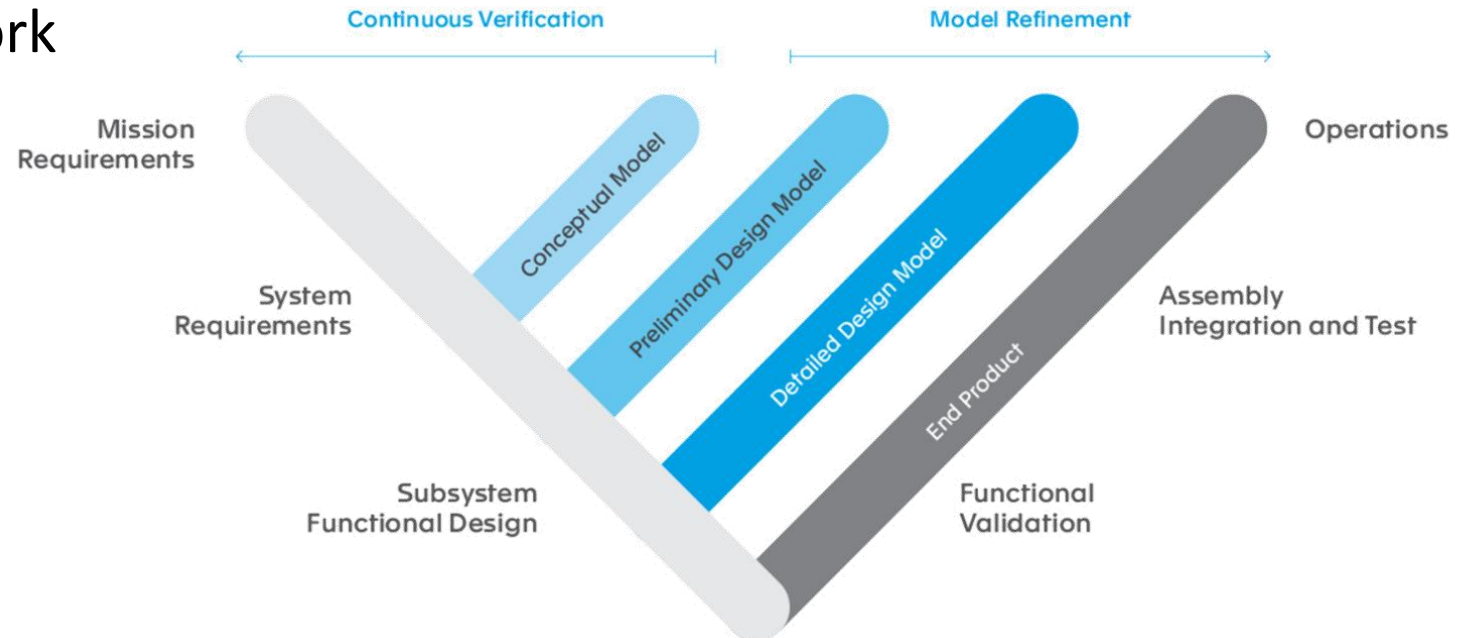
ATICA Model Based Safety Analysis



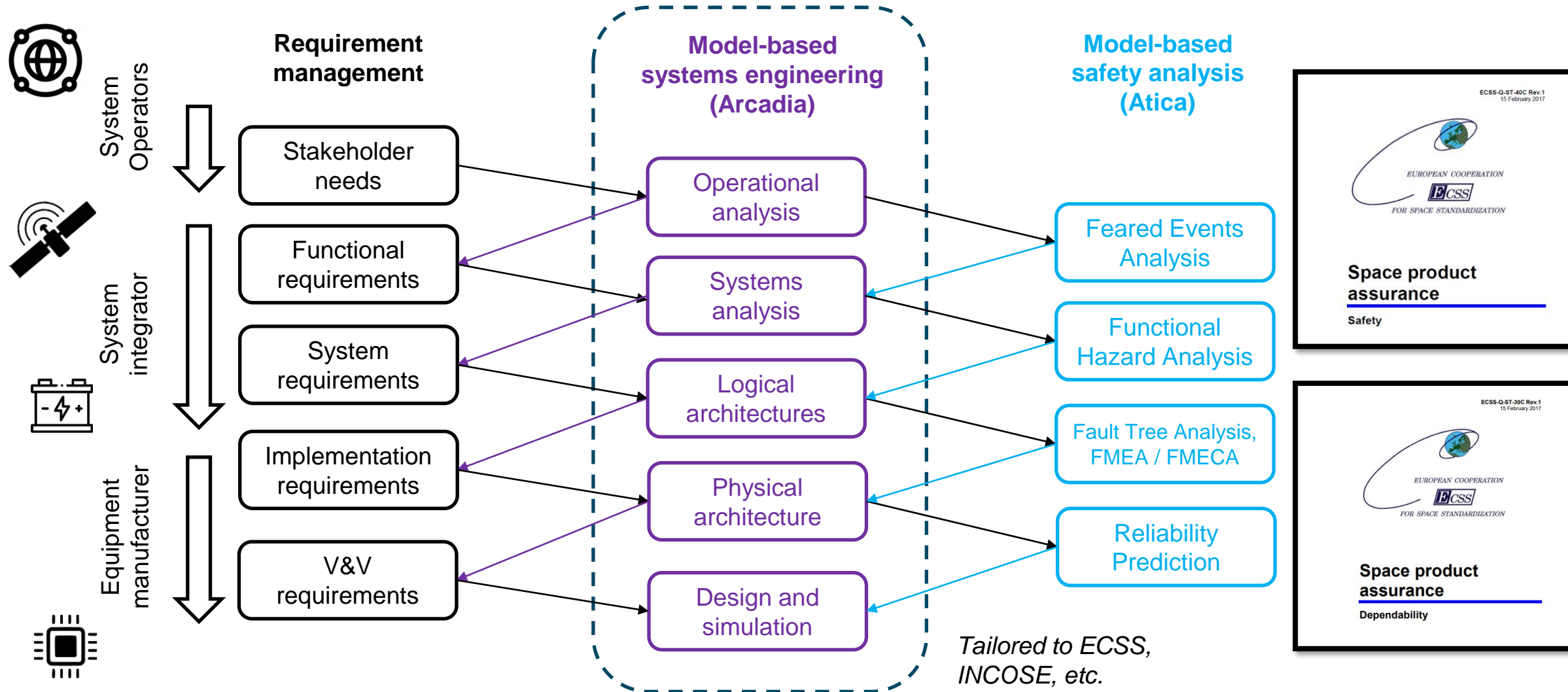


Outline

- Introduction to digital engineering
- Safety and dependability analysis
 - Model-based systems engineering and RAMS
 - Systems analysis
 - Logical and physical architectures
- Wrap-up and future work



MBSE & RAMS framework



Implementation in Capella

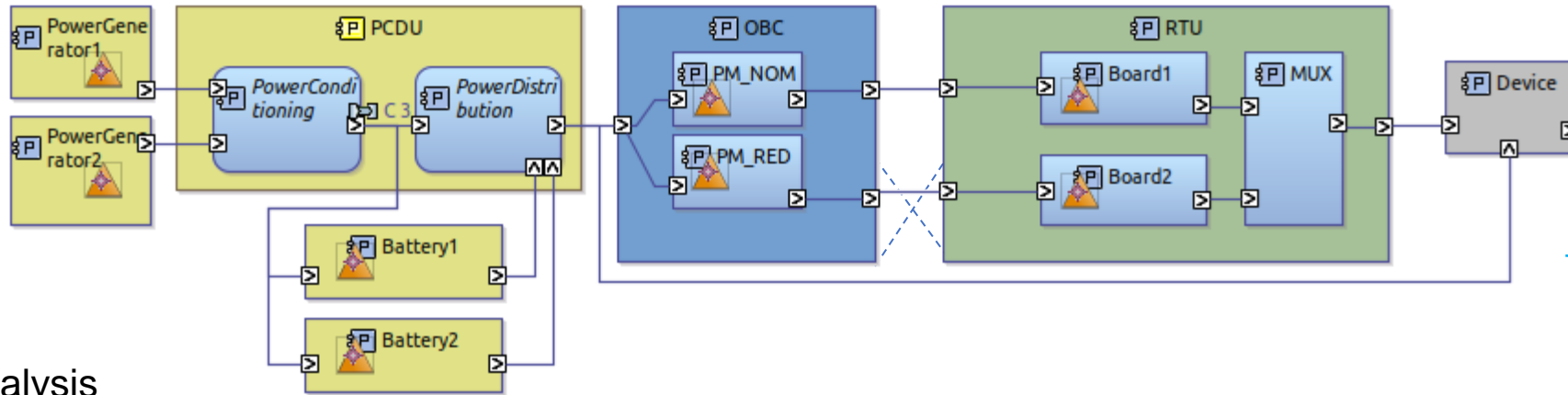
FMEA & FMES

The screenshot shows the Capella software interface for FMEA and FMES implementation. The Project Explorer on the left shows a tree structure under 'Physical Architecture' with 'MBSA Package' highlighted. The central diagram shows a 'PCDU' component containing a 'Power distribution' component, which in turn contains a 'Provide power supply' function. The Palette on the right shows the 'ATICA_MBSA' package containing 'Failure Mode' and 'All Allocated Failure Modes'. The table at the bottom summarizes the failure modes.

	Summary	Affects Function	Modes	Affects Component Port	Failure Effect	Failure rate (1/h)
Power distribution						
[Failure Mode]		[Provide power supply]	[]	Power_output	false	0.0
[Failure Mode]		[Provide power supply]	[]	Power_output	false	0.0

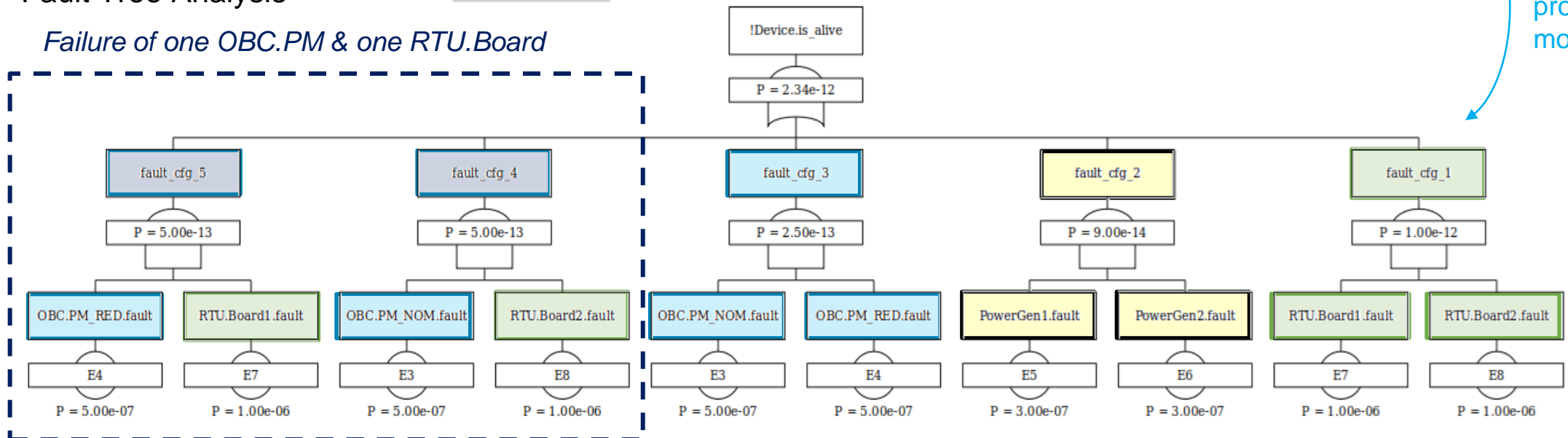
Implementation in Capella

- ✓ Extended system model: logical architecture + failure specifications

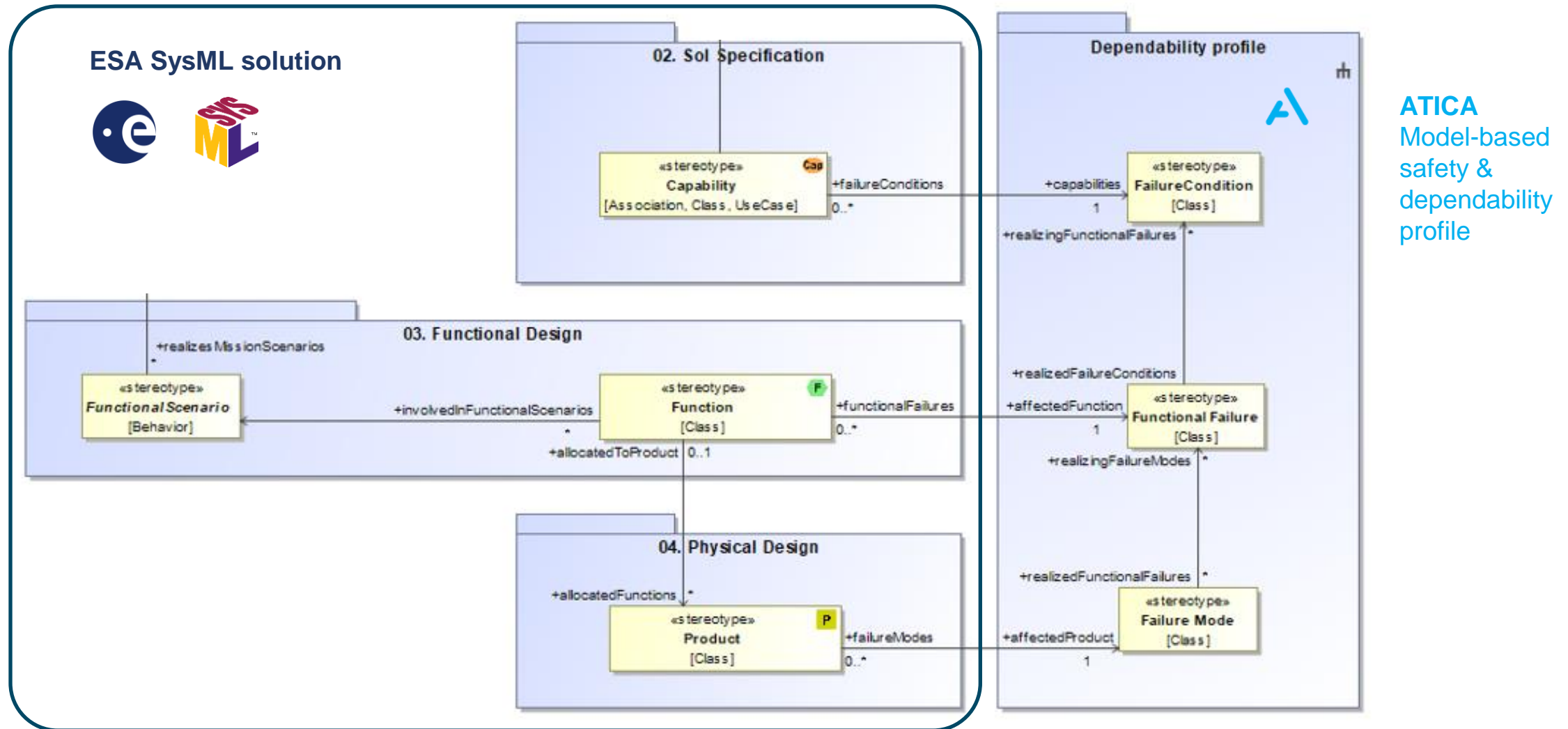


- ✓ Fault Tree Analysis

Failure of one OBC.PM & one RTU.Board

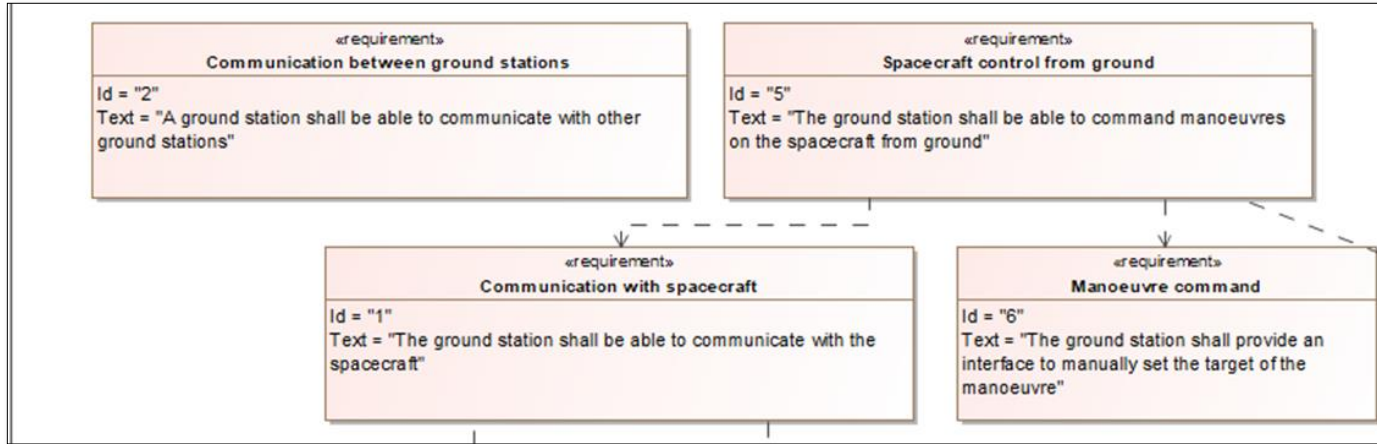


Adaptation to ESA SysML Solution

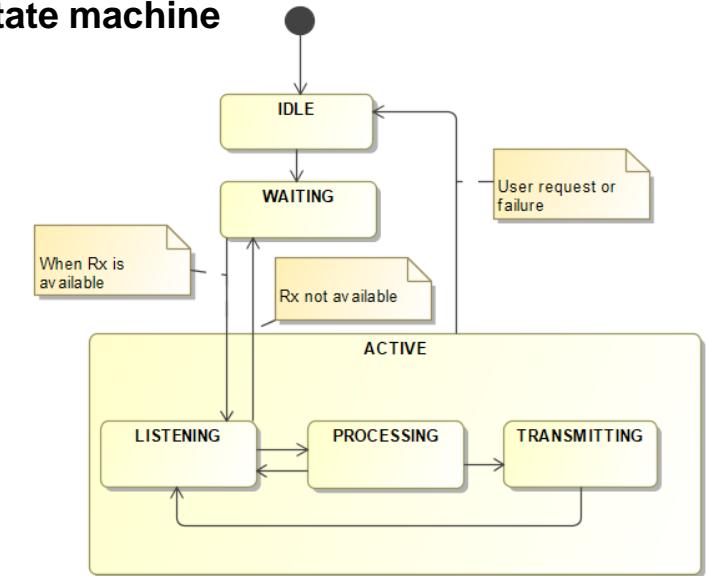


Implementation in Cameo

Requirement diagram

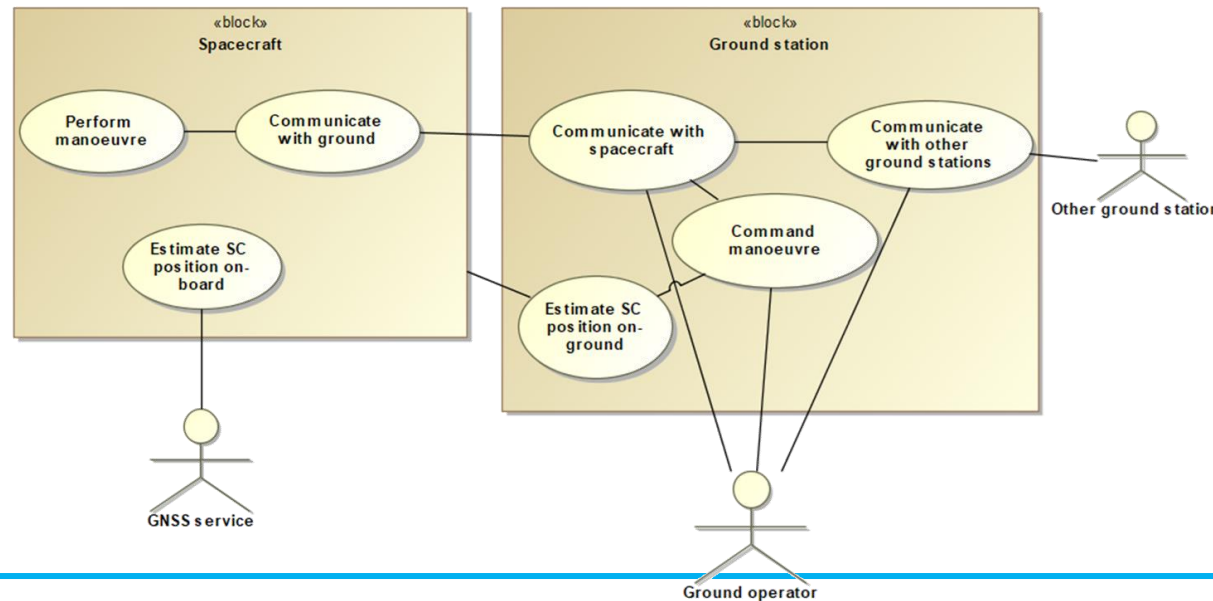


State machine



Behavior diagrams

- Use case
- Activity diagrams
- Sequence diagrams
- ...



Support to RAMS analysis

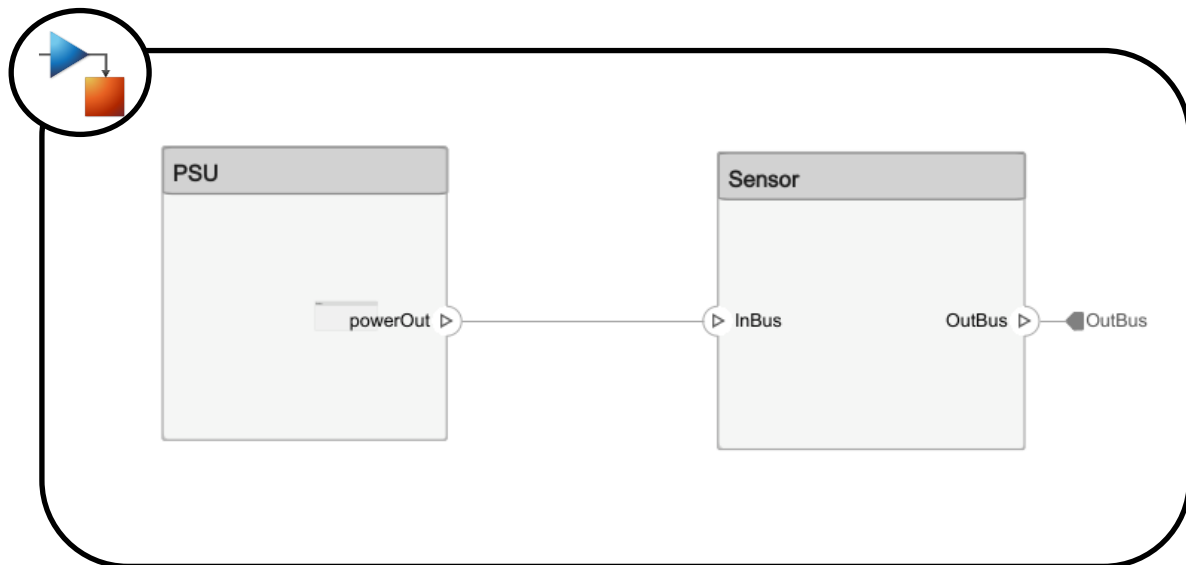
- Feared Events Analysis
- Functional Hazard Analysis

Implementation in Cameo

The screenshot shows the Cameo Systems Modeler 2022x interface. The main window displays a UML package diagram for 'package ATICA_MBSA'. It contains two classes: a red 'failureType' class with an attribute '-failureType : failureTemporalType_T [0..1] = Permanent' and a yellow '«ATICA» failureImplementation' class with attributes '-feature : String [0..1]' and '-effect : String [0..1]'. A dashed arrow labeled 'failureDependency' points from the implementation class to the failureType class. A blue dashed box highlights these two classes, with an annotation 'New object stereotypes with custom properties'. The top toolbar contains a custom button with the ATICA logo, highlighted by a blue dashed box and labeled 'Customized buttons'. The left-hand menu is open, with the 'ATICA: import failure specification' option highlighted by a blue dashed box and labeled 'Menu actions'. The menu also shows other options like 'New Project...', 'Open Project...', 'Save Project', and 'Print...'. The bottom status bar shows 'heavy' and '100%' zoom.

✓ Menu actions

From models to formal specification



```
mySystem.slim x +
1  system implementation myModel.Imp
2  subcomponents
3      Sensor: system Sensor.Imp;
4      PSU: system PSU.Imp;
5  connections
6      port Sensor.OutBus -> OutBus;
7  end myModel.Imp;
8
9  system implementation Battery.Imp
10 end Battery.Imp;
```



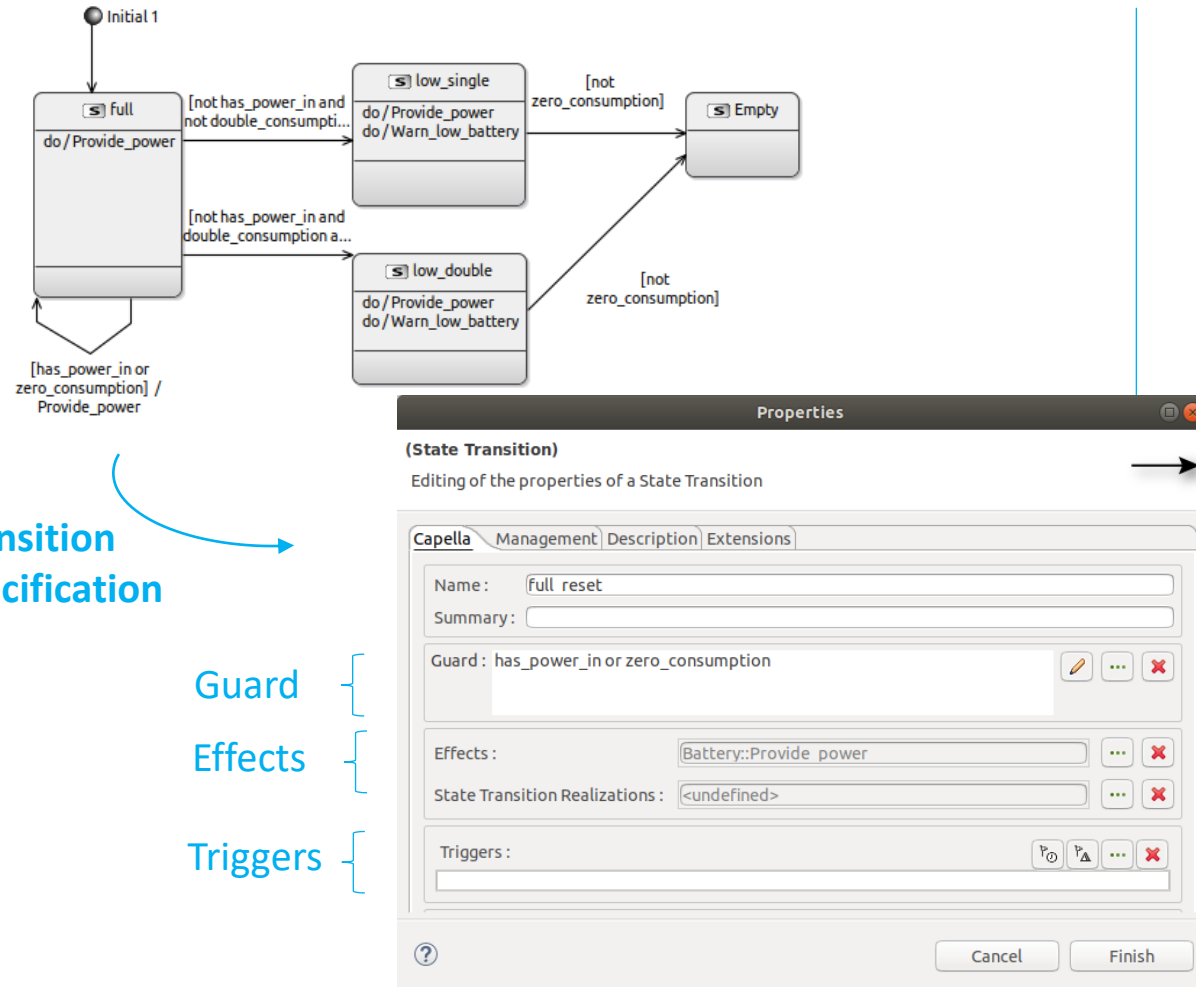
Model parser:
SysML (System Composer)
to SLIM/AADL (TASTE/COMPASS)



Analysis backends
FMEA
Reliability Block Diagrams (RBD)
Fault tree analysis (COMPASS)
Model checking (COMPASS)

From models to formal specification

Modelling viewpoint



Transition specification

Guard
Effects
Triggers

AADL/SLIM file

```

system implementation Battery.Imp
states
  full: initial state;
  low_single: state;
  Empty: state;
  low_double: state;
transitions
  full -[ when not has_power_in and not double
          then has_power_out := true; low := t
  low_single -[ when not zero_consumption
                then ]-> Empty;
  full -[ when has_power_in or zero_consumptio
          then has_power_out := true ]-> full;
  full -[ when not has_power_in and double_con
          then has power out := true; low := t
end Battery.Imp;
    
```

Transition condition specification
[trigger when guard then effect]

Implemented features:

- ✓ Mode/state declaration
- ✓ Mode/state transition
- ✓ Mode/state transition criteria (guard) and effects



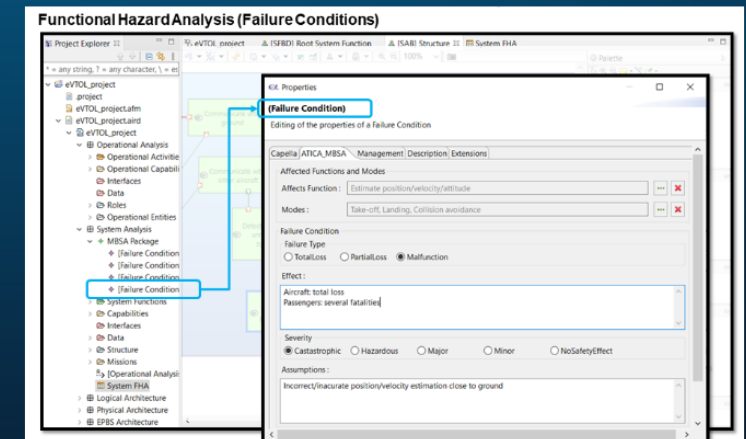
Outline

- Introduction to digital engineering
- Safety and dependability analysis
- Wrap-up and future work

\ ANZEN \ Download

ATICA4Capella

[ATICA4CAPELLA](#) is a model based safety analysis (MBSA) plugin that extends [Capella](#) functionalities and allows to perform safety and reliability analysis directly from the system models.



Download Atica4Capella

READ MORE

Wrap-up and future work

Key takeaways

- ✓ ATICA is a model-based safety and dependability analysis framework
- ✓ ATICA could be adapted to multiple MBSE frameworks and modeling methodologies

ATICA simplifies the job of systems and software architects by providing a consistent **framework** covering from conceptual design up to hardware & software implementation

Wrap-up and future work

Future work

- ✓ Consolidate the framework and extend its functionalities towards software engineering
 - ✓ Hardware and Software Integration Analysis (HSIA)
 - ✓ System correctness (model checking and formal analysis)
 - ✓ Dynamic and autonomous systems (FDIR)

ANZEN

SYSTEM SAFETY AND DIGITAL ENGINEERING



Pablo Lopez Negro

Head of Innovation

[pablolopez](mailto:pablolopez@anzenengineering.com)

[@anzenengineering.com](mailto:pablolopez@anzenengineering.com)

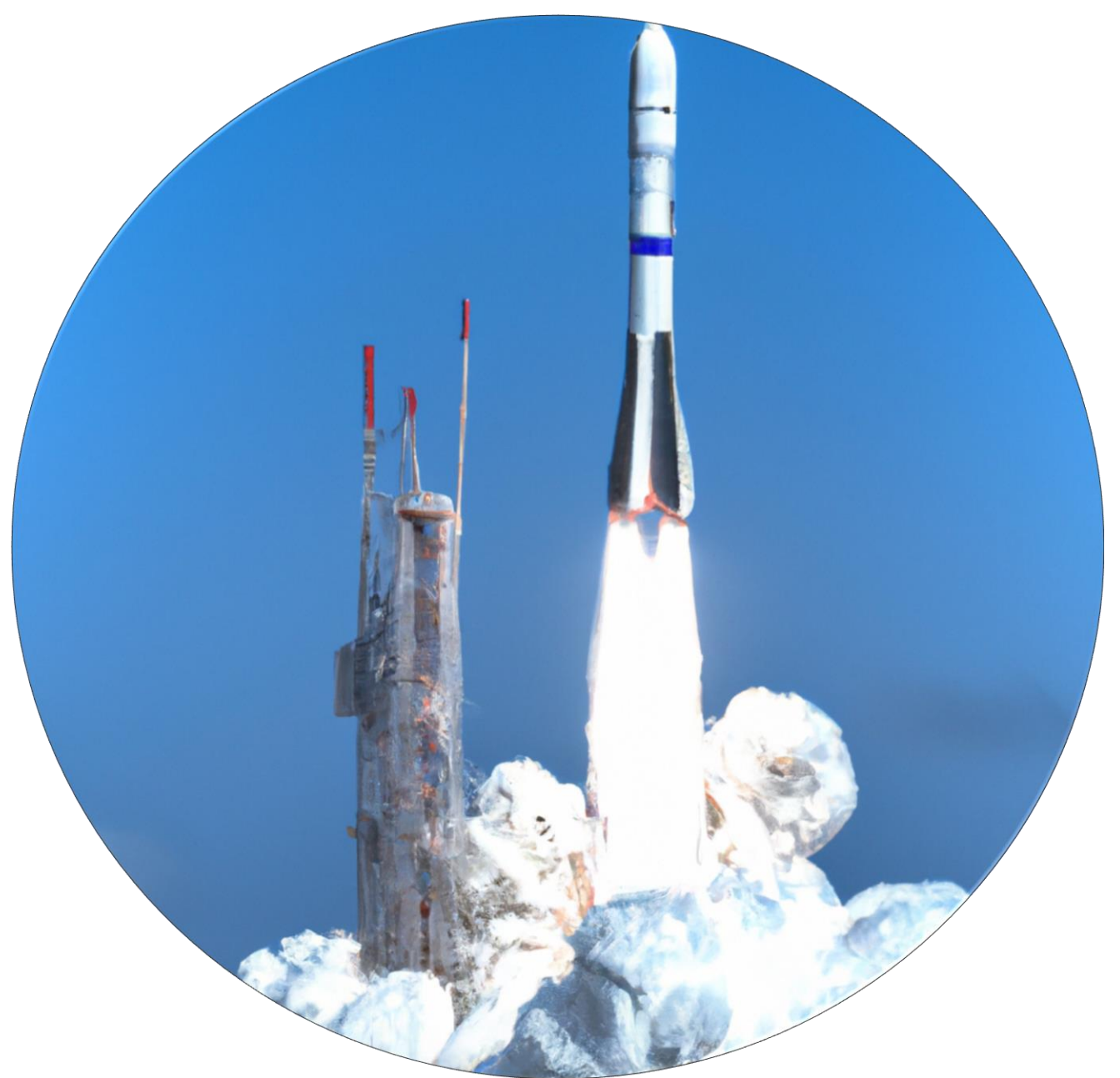


María Alonso López

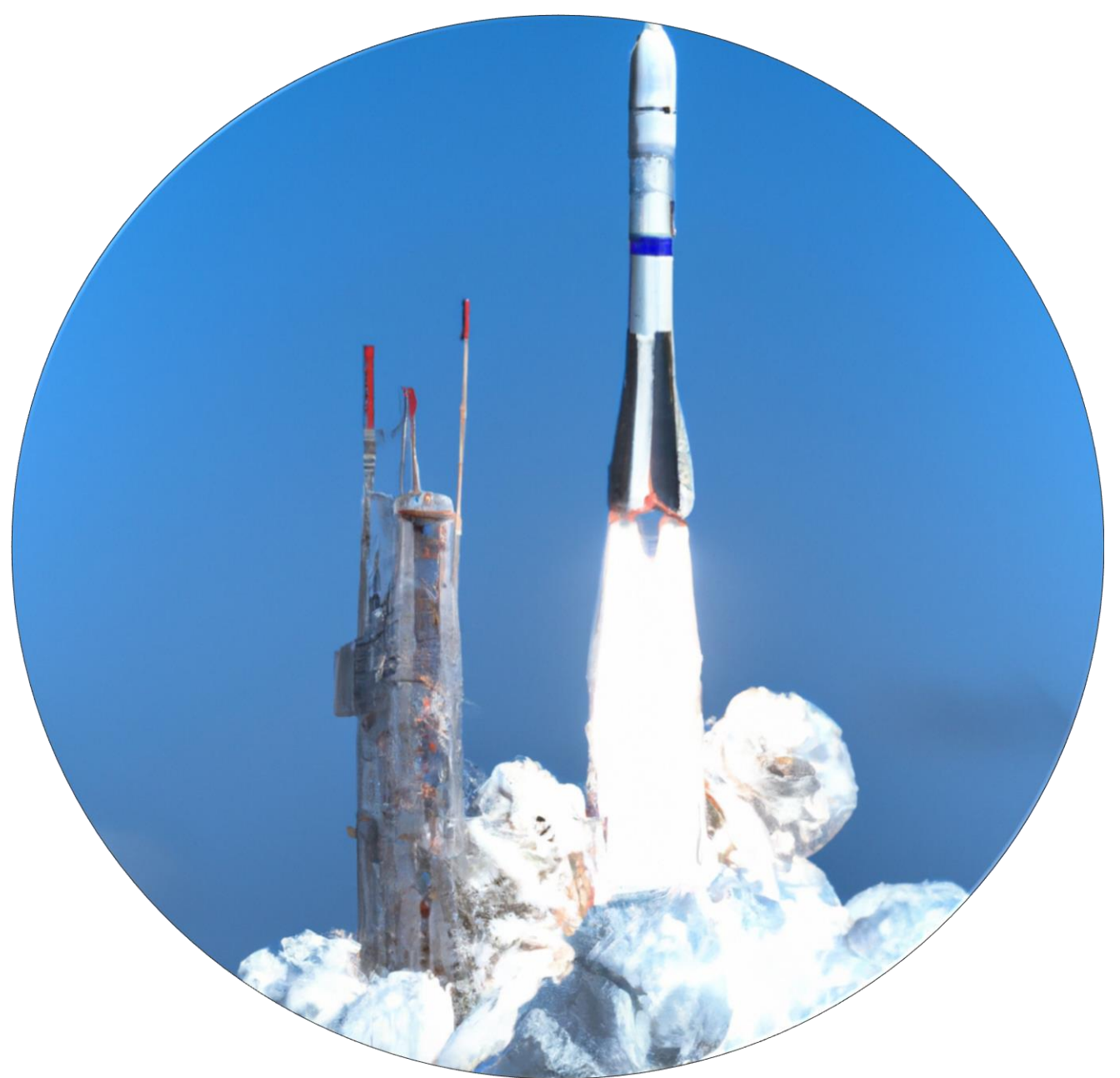
PA & RAMS Lead

[mariaalonso](mailto:mariaalonso@anzenengineering.com)

[@anzenengineering.com](mailto:mariaalonso@anzenengineering.com)



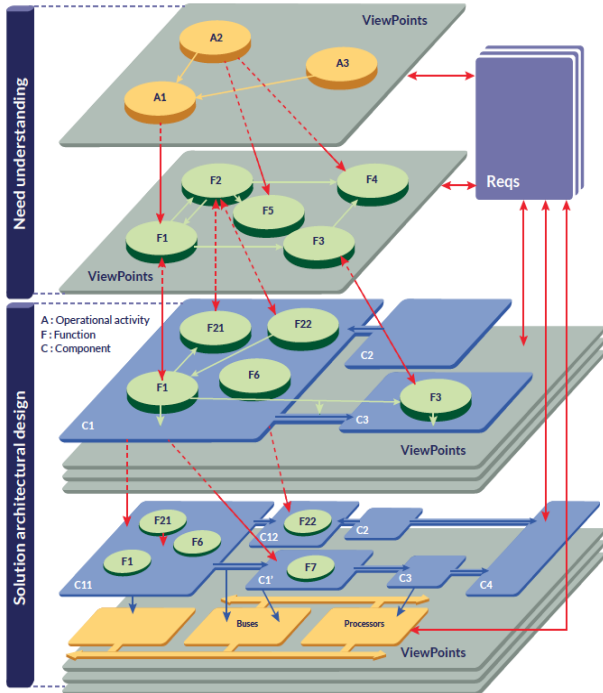
Backup slides



Model-Based Systems Engineering



CAPELLA / ARCADIA - MBSE Framework



Operational Analysis
What the users of the system need to accomplish

Functional & Non Functional Need
What the system has to accomplish for the users

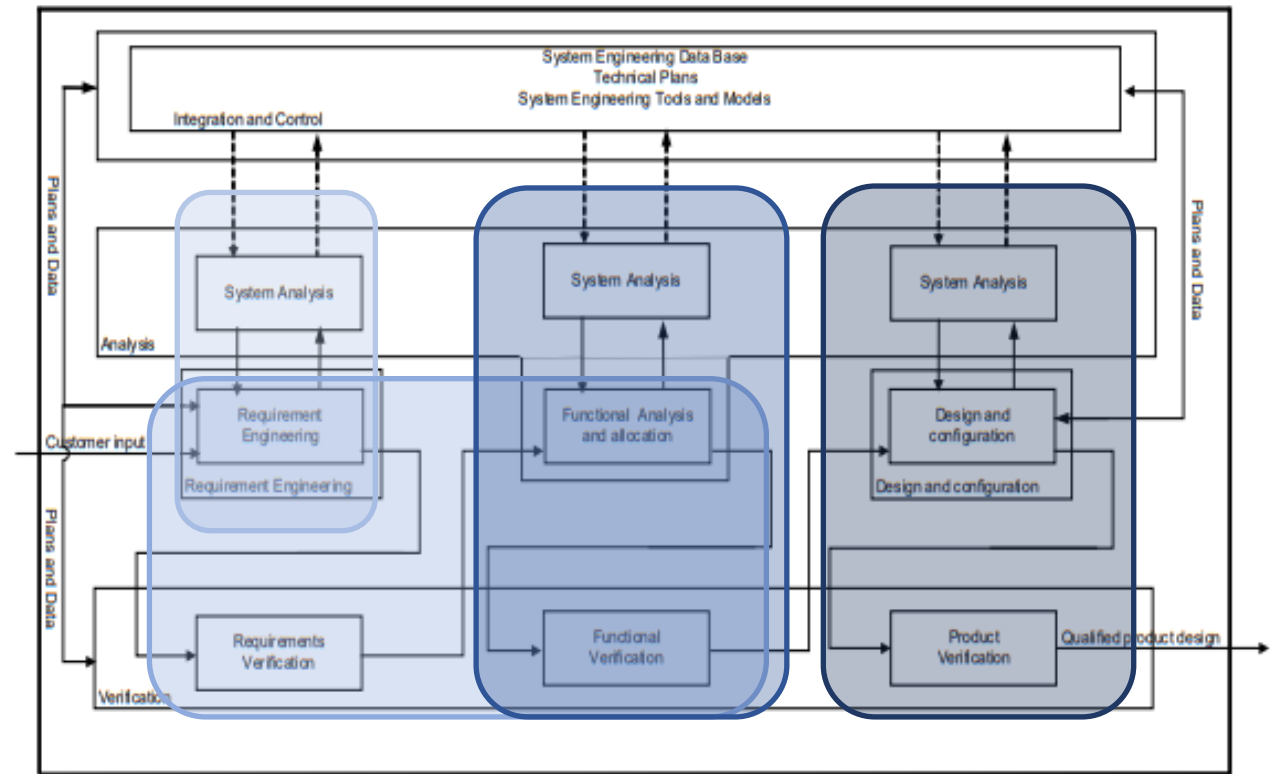
Logical Architecture
How the system will work to fulfill expectations

Physical Architecture
How the system will be developed and built



tailoring

ECSS-E-ST-10C Rev1 – System Engineering General Requirements



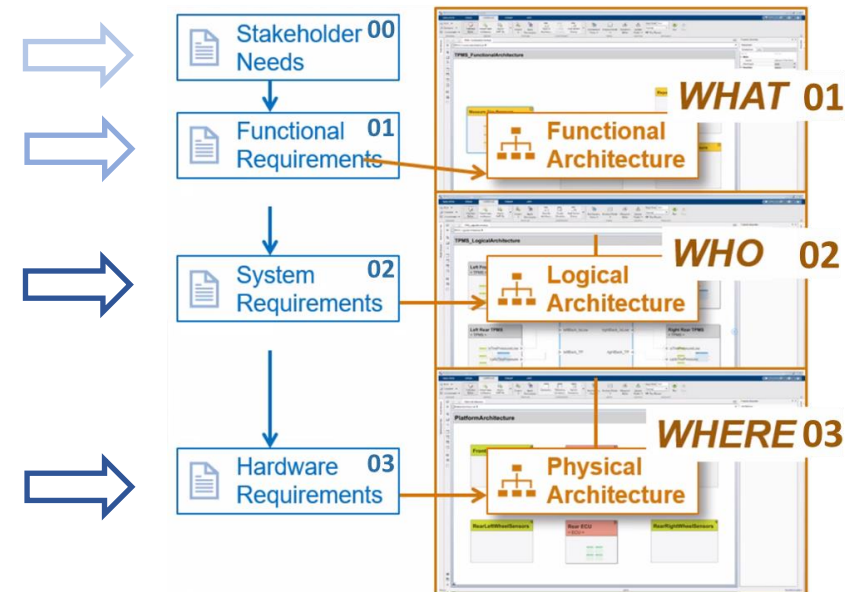
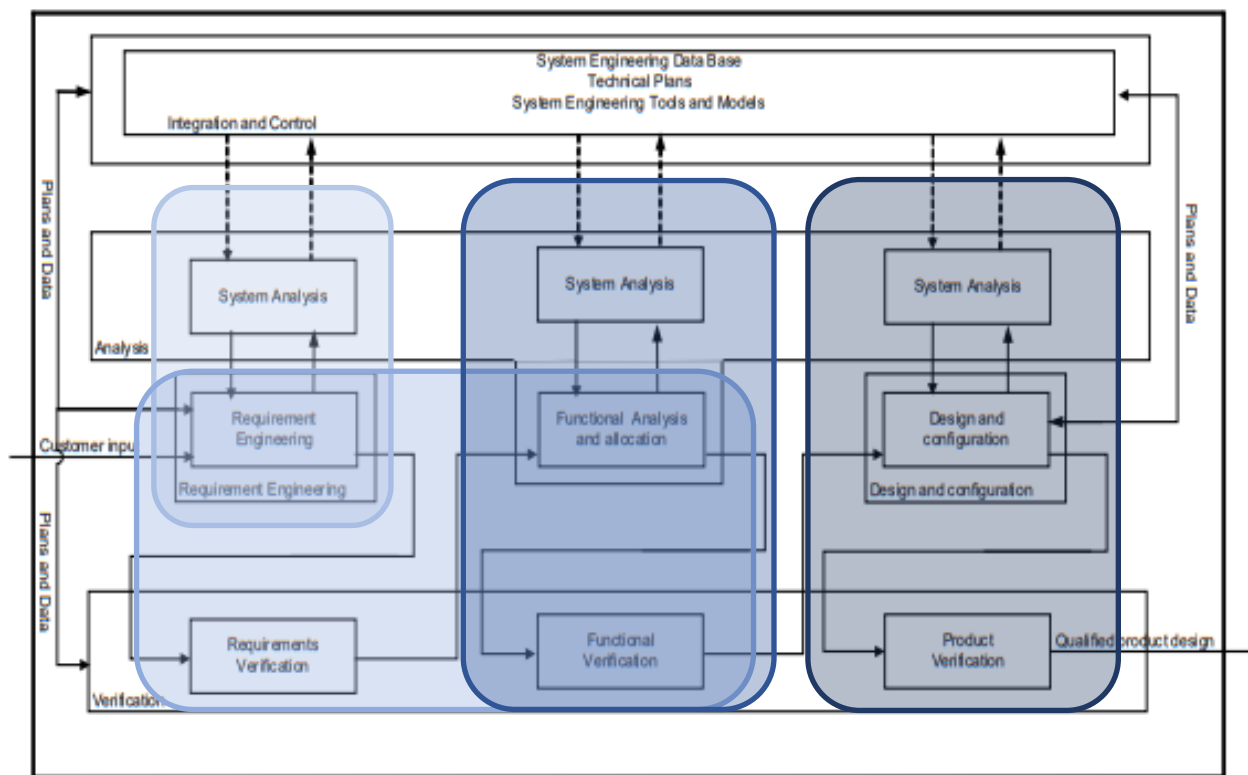
Model-Based Systems Engineering

Adaptation to MBSE frameworks

tailoring

▶ **Mathworks**
SystemComposer (SysML)

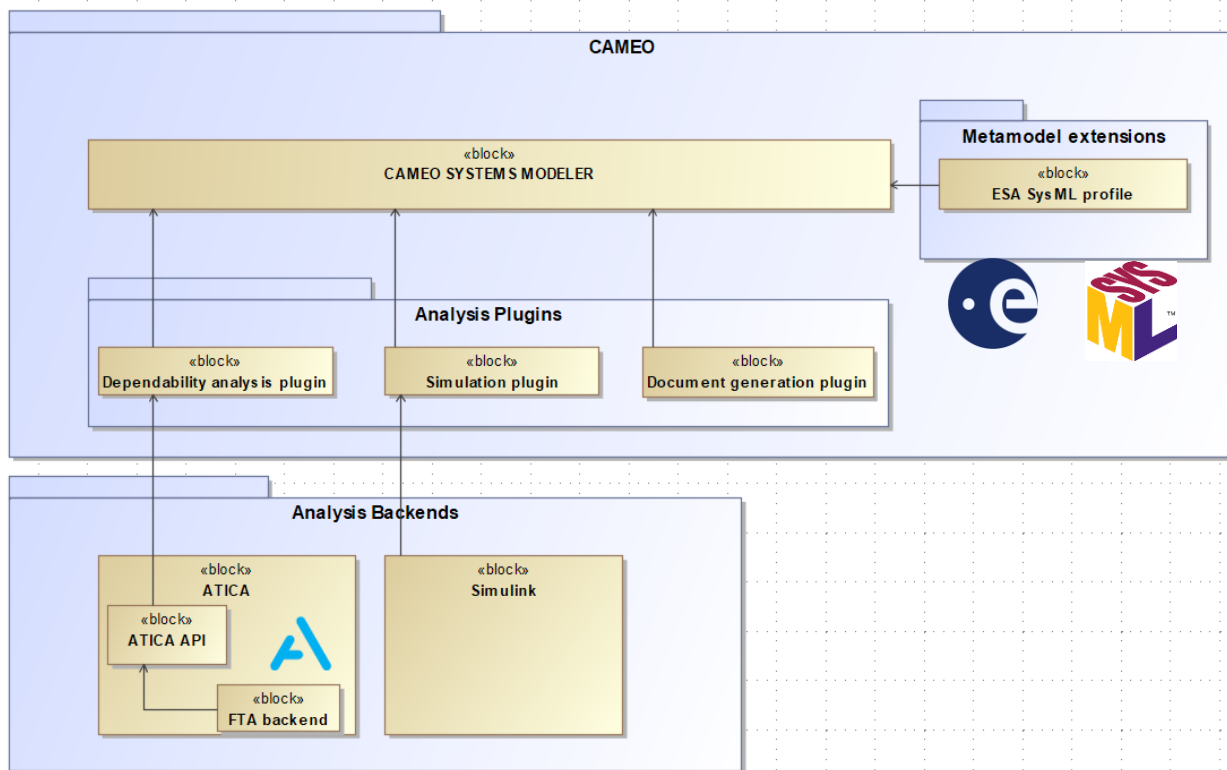
ECSS-E-ST-10C Rev1 – System Engineering General Requirements



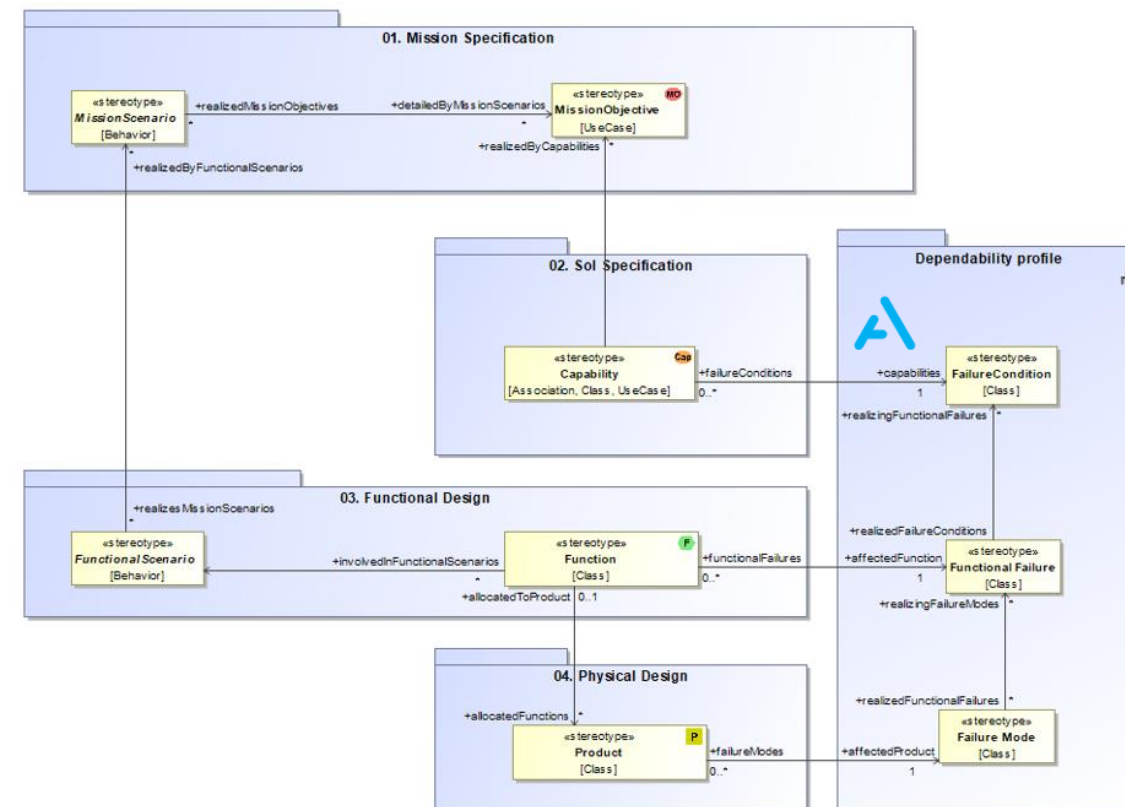
Model-Based Systems Engineering

Adaptation to MBSE frameworks

Software architecture



System-Safety metamodel



Model-based safety & dependability profile

Implementation in Capella

The screenshot displays the Capella software interface with several components highlighted:

- Project Explorer:** Shows a tree view of the project structure. The 'test_MBSA' package is expanded, showing 'MBSA Package' and 'Failure Condition 1' under 'System Analysis'.
- Diagram:** A central diagram shows a green box labeled 'SystemFunction 1' with a diamond-shaped failure condition symbol at its base.
- Properties Window:** A 'Properties' dialog is open, titled '(Failure Condition)'. It contains the following fields:
 - Affected Functions and Modes:** 'Affects Function' is set to 'SystemFunction 1'.
 - Failure Condition:** 'Failure Type' is set to 'TotalLoss'.
 - Effect:** The text 'Effect on system function 1' is entered.
 - Severity:** 'Castastrophic' is selected.
- Bottom Panel:** A list of packages is visible, with 'ATICA_MBSA' and 'Failure Condition' highlighted.