



## Small satellites mission design enhancement through MBSE and DDSE toolchain

*Paolo Minacapilli, Flor Criado, Saúl Campo, Alberto Rodríguez, David Escudero*

All from Elecnor Deimos



Model Based Space Systems and Software Engineering (MBSE2022)

Airbus Leadership Academy at Toulouse (France)

22-24 November 2022

# Contents

- MBSE at Deimos
- Introduction
- Valispace (DDSE) and Capella (MBSE) capabilities
- V-Model
- MBSE toolchain definition
- MBSE toolchain consistency
- MBSE toolchain throughout the project lifecycle
- Conclusions and SWOT analysis

# MBSE at Deimos

SMALL SATELLITES  
DEVELOPMENT



## Comet Interceptor Probe B2

*Phase B*

- Valispace for requirements management and budgets
- Capella for detailed ConOps and Modes

## DRACO

*Phase A/B1*

- Valispace for requirements management, SS sizing and budgets
- Capella for full functional/physical architecture model



Others

# Introduction



Model Based Systems Engineering (MBSE) tool

Based on ARCADIA Methodology

Powerful for architecture modelling



Data Driven Systems Engineering (DDSE) tool

Simplified engineering process

Complete MBSE toolchain  
Shared models for internal team and external reviews

How to guarantee the single source of truth?

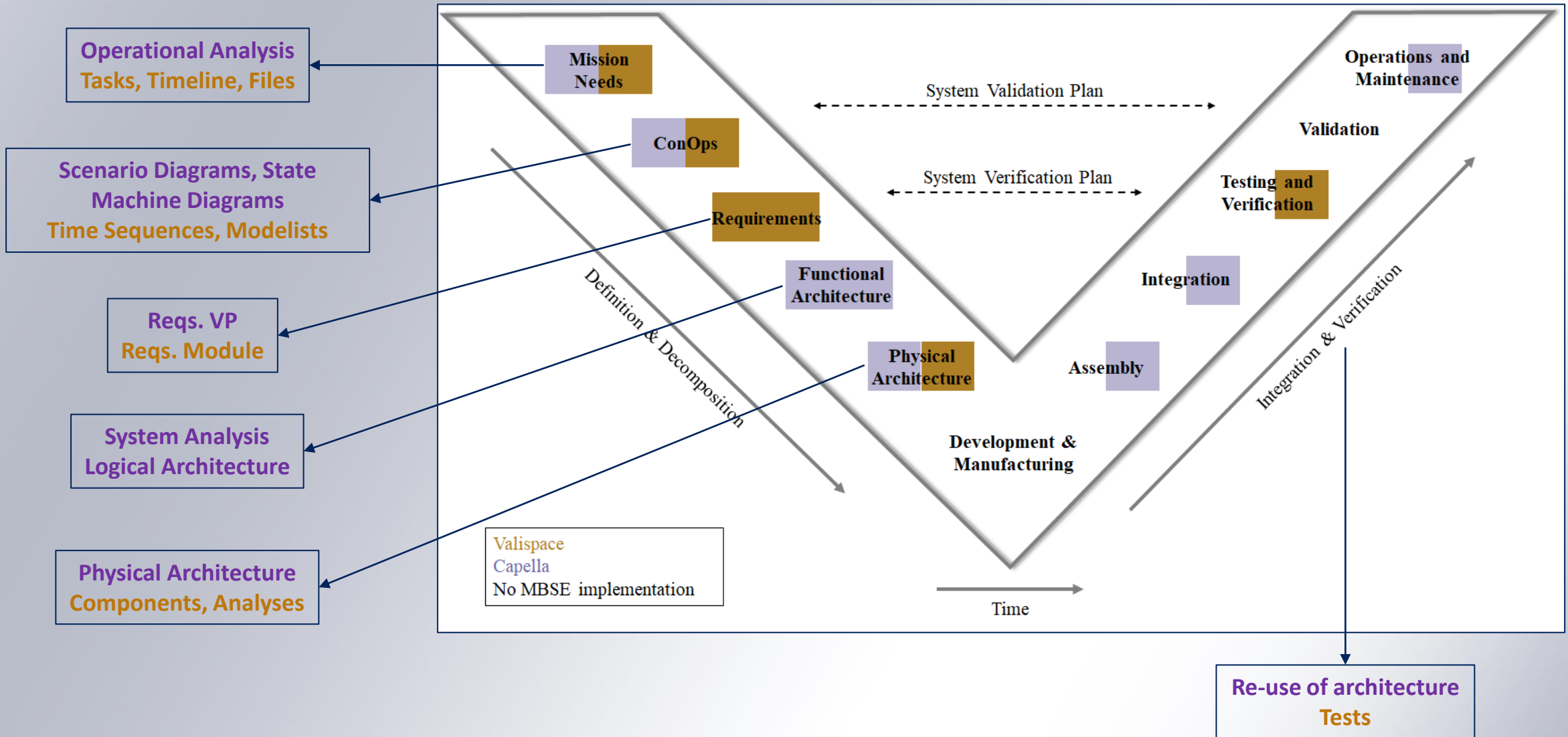


- Develop a clear methodology
- Interface the different workspaces
- Minimize information repetition

# Valispace (DDSE) and Capella (MBSE) capabilities

	Capella	Valispace
Requirements management	Flow-down, <b>traceability with any model element</b> , difficult to export	Digitization, flow-down, traceability, gaps analysis, <b>user-friendly</b> , easy to export
Concept of Operations	<b>Sequence diagrams</b>	<b>Limited to high level</b> , no branching
Modes and States	UML-inspired <b>state machines</b> , transitions modelled with detail and can be related to other model elements	Components and their values can be allocated to modes and simulated
Functional Analysis	Detailed <b>functionalities and functional interfaces</b> modelling at different levels, function trees	Definition and trace but poor model
Architecture management	Detailed logical and physical architecture, allocation of functions to components, functional chains, trees	Only physical without modelling interfaces, trade-offs
Budgets and simulation	Class diagrams can be used to model <b>static data structures</b> and relate them to model elements	Components can be modelled with their non-static sizing values, <b>end-to-end traceability</b> and <b>data consistency</b>
Interfaces management	Detailed functional and physical interfaces	Limited to definition
RAMS management	Need add-on	Limited to high level
Teamwork	Need add-on	Real-time, internal reviews, comments, notifications
Project management	Operational analysis only	<b>Tasks</b> , timeline, configuration control
AIV/AIT management	Architecture model can be re-used for AIV/AIT plan	Test procedures, requirements and verifications in the same tool
Change tracking and history	Weak	Full repercussion on the design

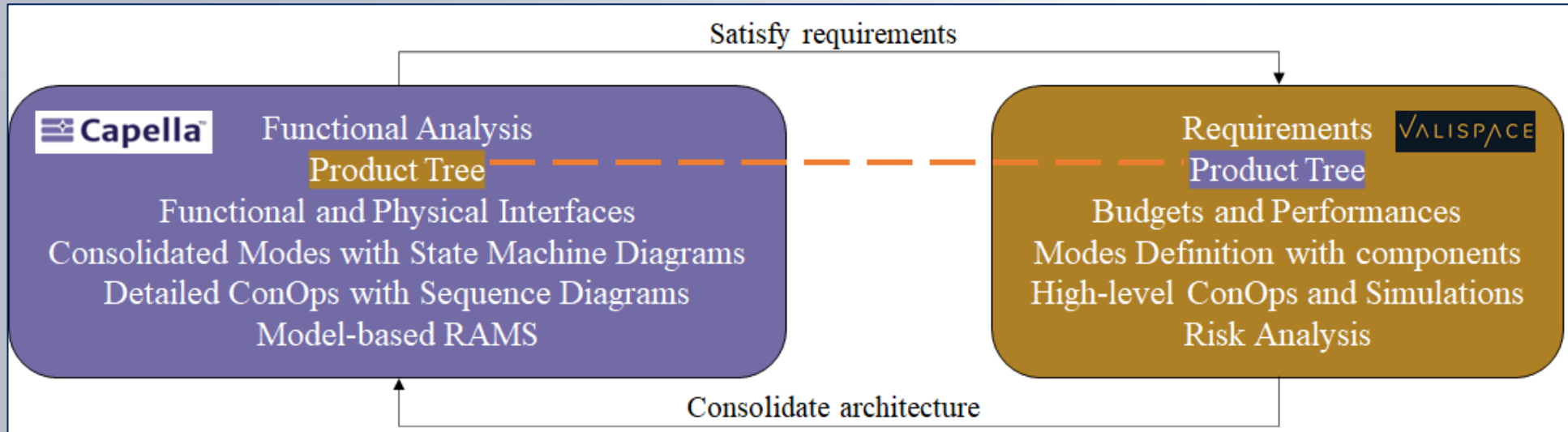
# V-Model



# MBSE toolchain definition

**Capella** is the main reference for the architecture definition, following the ARCADIA method.

**Valispace** is used to work with engineering data and requirements.



Requirements are managed in **Valispace** and traced with the logical/physical architecture in **Capella**.

Analyses conducted in **Valispace** are used to consolidate architectural design decision and serving as input for **Capella**.

Consistency supported by the shared **Product Tree**.

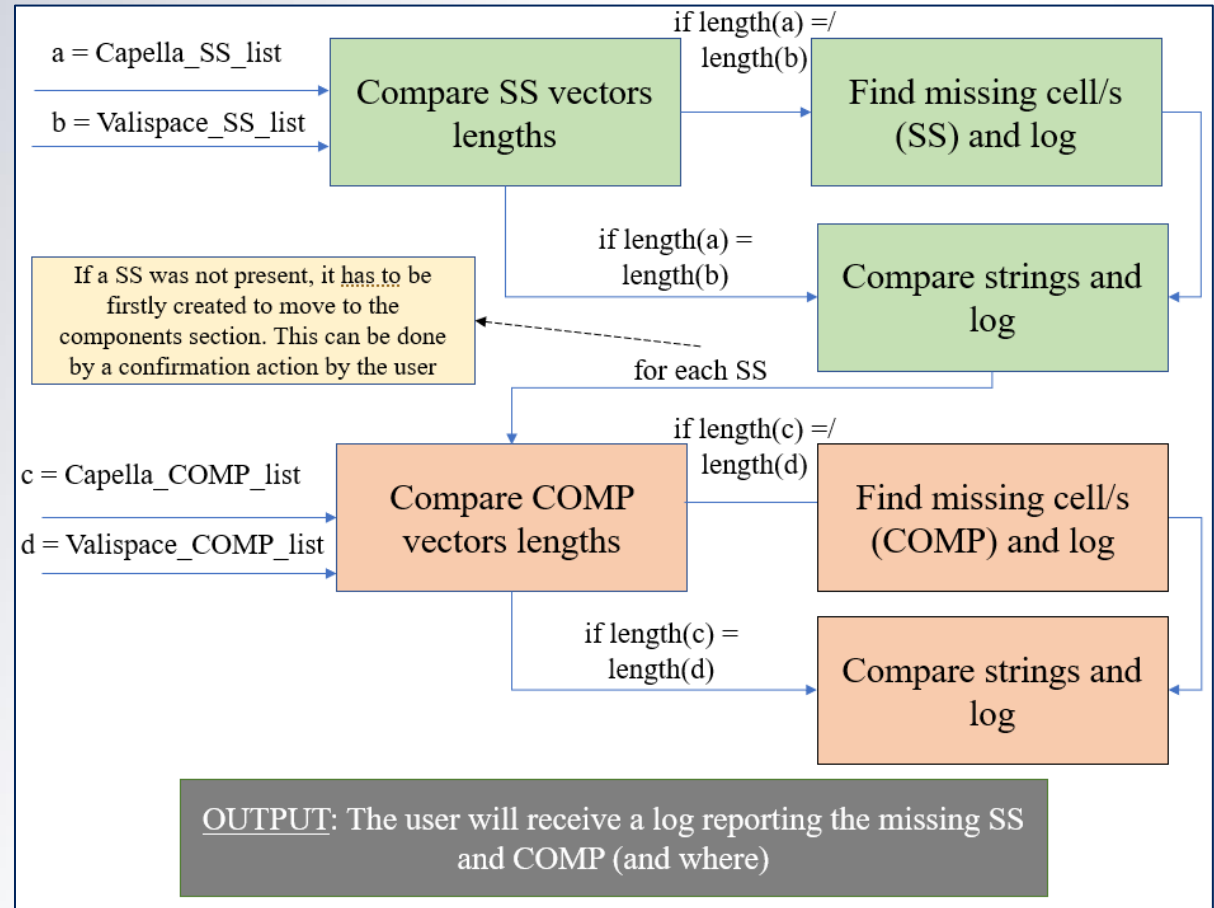
# MBSE toolchain consistency

Compare the vectors reporting the subsystems and components from Capella and Valispace, exploiting respectively the **Python4Capella** add-on and the **Valispace Python API** to manage the information in Python.

The script will return a log indicating if a subsystem/component is missing and in which model.

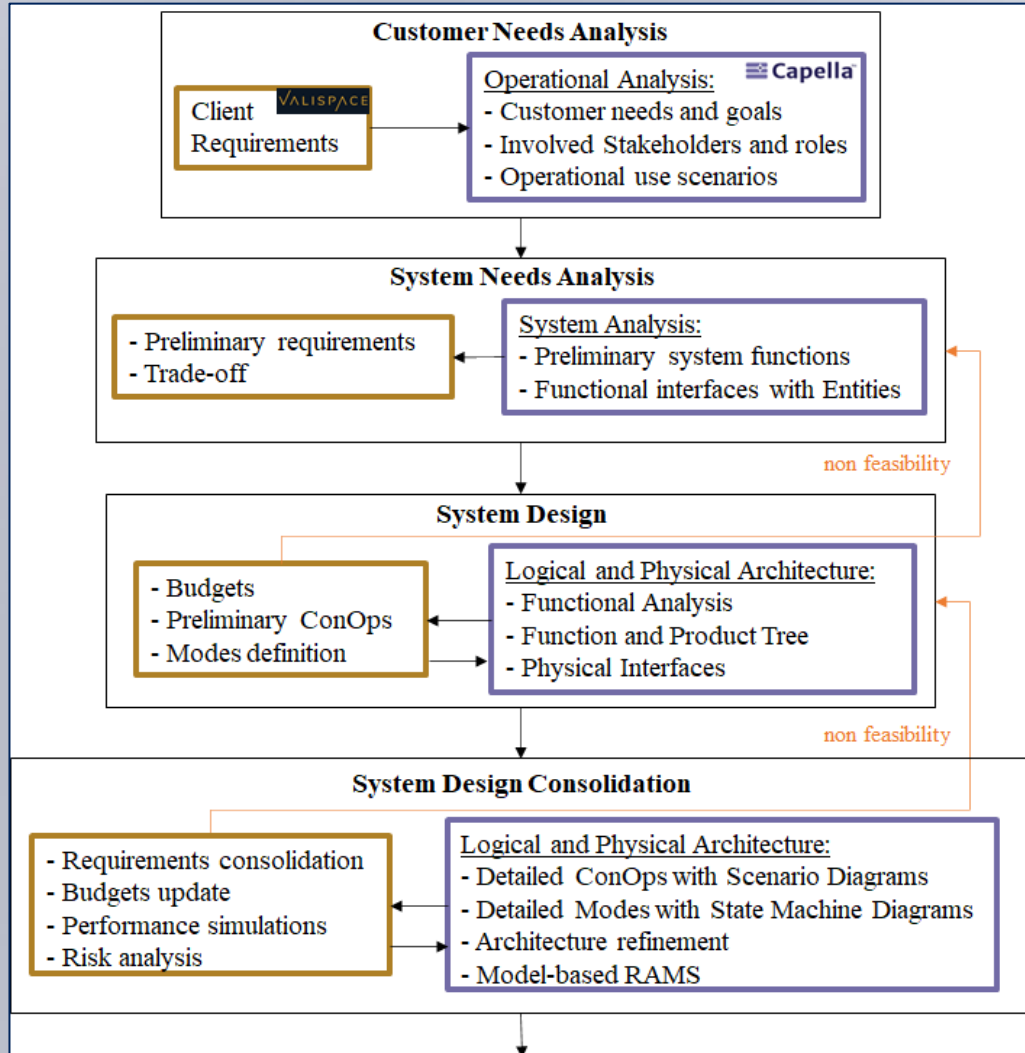
The possibility of directly creating the subsystem/component in either tool environment is not excluded (keeping also the log alert).

Applicable also to requirements.



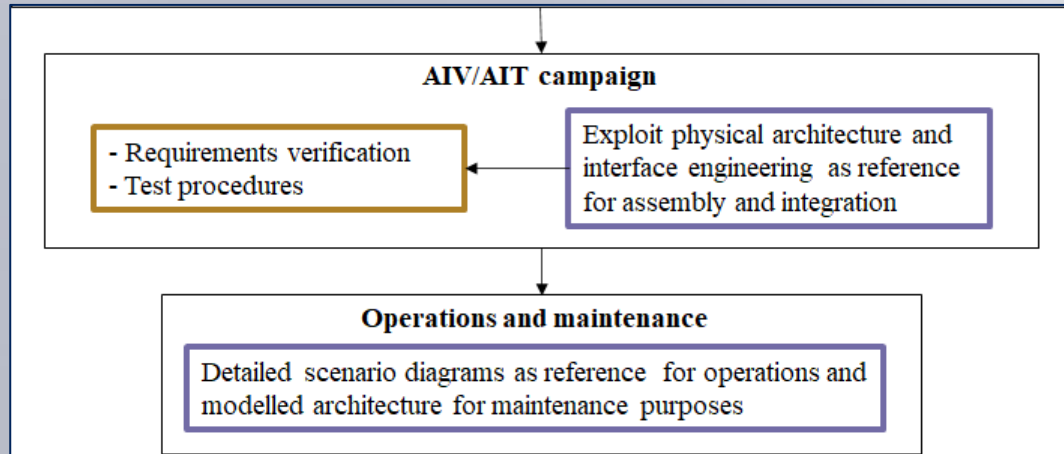


# MBSE toolchain throughout the project lifecycle



- Mission/client needs imported in Valispace as requirements
- Operational Analysis in Capella to define stakeholders and operational use scenarios
- Basic functional analysis in Capella
- Mission architecture trade-off analysis conducted in Valispace
- Iterative design using Logical/Physical Architecture in Capella and quantitative sizing in Valispace
- The architecture model can be re-used for RAMS analyses and software development

# MBSE toolchain throughout the project lifecycle



- Re-use architecture model for AIV/T campaign definition
- Produce detailed test procedures, update requirements verification, assign responsible
- Re-use architecture model for Manual of Operations and maintenance activities

# Conclusions and SWOT analysis

Most of the systems engineering activities are enhanced for small satellites design with the MBSE toolchain.

<p style="text-align: center;"><b><u>STRENGTHS</u></b></p> <ul style="list-style-type: none"> <li>• Requirements traceability</li> <li>• Enhanced ConOps modelling and simulation</li> <li>• Enhanced architecture modelling (functional, logical and physical) with respective budgets and trade-off</li> <li>• Enhanced interfaces definition and control</li> <li>• Enhanced definition and implementation of RAMS (FMEA, FTA, FDIR)</li> <li>• Leverage Valispace Files Management module as the single source of truth for both tools</li> <li>• Leverage Valispace project management feature for tasks allocation between both tools</li> </ul>	<p style="text-align: center;"><b><u>WEAKNESSES</u></b></p> <ul style="list-style-type: none"> <li>• Model information in different workspaces</li> <li>• Double information for certain activities</li> <li>• Capella user needs to review Valispace model for updates and task</li> </ul>
<p style="text-align: center;"><b><u>OPPORTUNITIES</u></b></p> <ul style="list-style-type: none"> <li>• Foster the development of an all-in-one-place tool which gathers the strong features of both considered tools</li> <li>• Foster the implementation of MBSE for the whole lifecycle of a space project</li> </ul>	<p style="text-align: center;"><b><u>THREATS</u></b></p> <ul style="list-style-type: none"> <li>• Loss of updates between both models</li> <li>• Bridge python script developed ad-hoc may not be compatible with future versions of the tools</li> </ul>

Further work is needed to ensure full consistency (requirements, components, properties, etc.).

*Thank you for the attention!*

## Small satellites mission design enhancement through MBSE and DDSE toolchain

*Paolo Minacapilli, Flor Criado, Saúl Campo, Alberto Rodríguez, David Escudero*

All from Elecnor Deimos

[paolo.minacapilli@deimos-space.com](mailto:paolo.minacapilli@deimos-space.com)



Model Based Space Systems and Software Engineering (MBSE2022)

Airbus Leadership Academy at Toulouse (France)

22-24 November 2022