33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

Introducing Variability in the R-MOFLT Architecture Framework

# MBPLE for MOFLT

AIRBUS

## Raphael Henrique Madeira

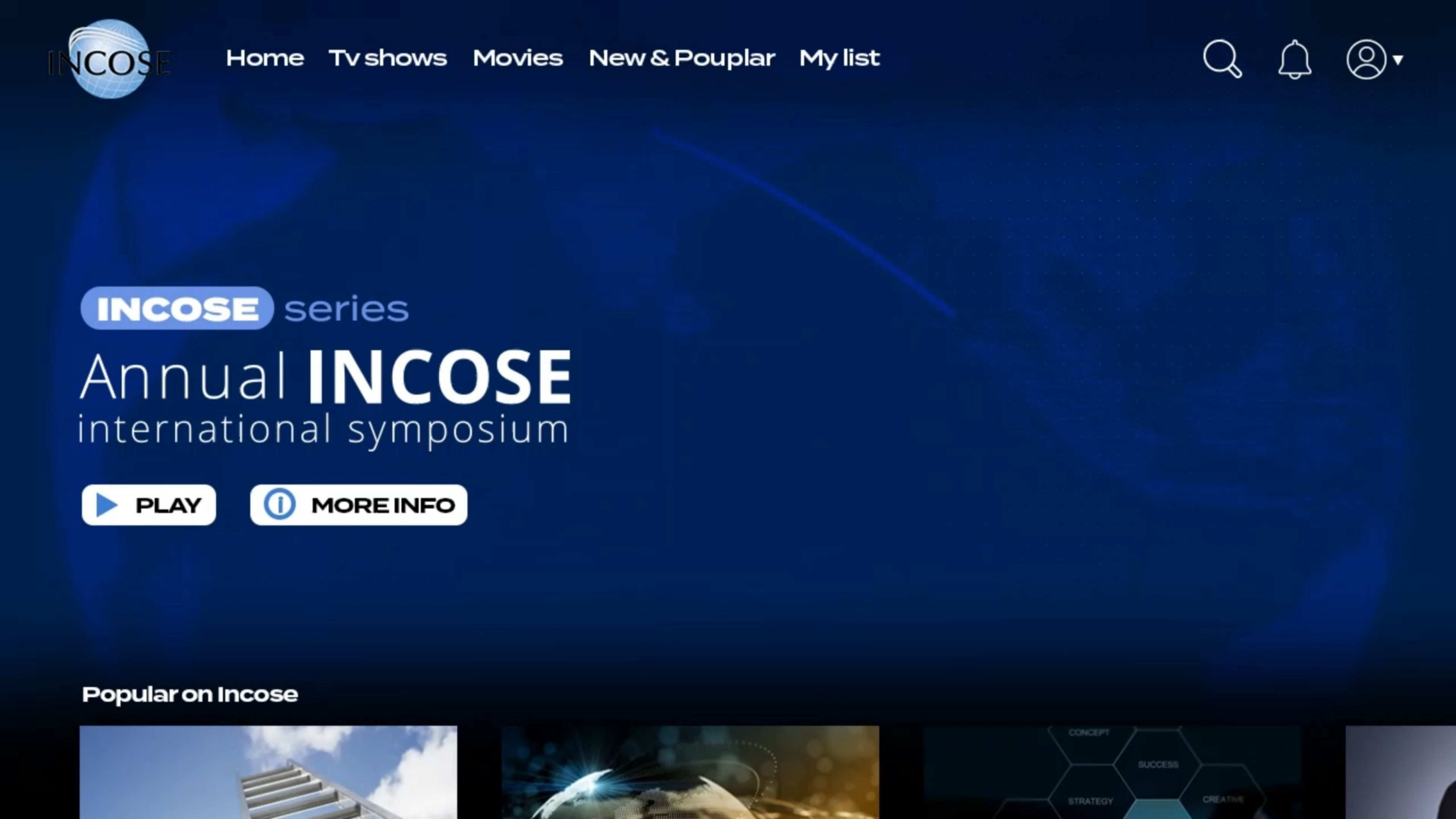
is a passionate Aeronautical Engineer with background in Flight Dynamics and Sizing, Loads and Aeroelastics, Systems Engineering, Model Based Systems Engineering and Product Line with more than 5 years of experience in Aerospace Business. At Airbus he leads the deployment of SE and MBSE at Flight Physics organisation and also acts as Product Owner for the MBPLE Framework, supporting the definition of key methodologies to allow better Variability Management for Product Line developments, Customization and Trade-offs.

## Davi Henrique de Sousa Pinto

has over 5 years of experience in the Aerospace Business. His background ranges from systems engineering, model-based systems engineering, model-based product line engineering, flight operations to software development. At Airbus, he is acting as Epic Owner of MBPLE, supporting the development and deployment of a holistic MBPLE framework to support the different functions and programmes.

## Marco Forlingieri

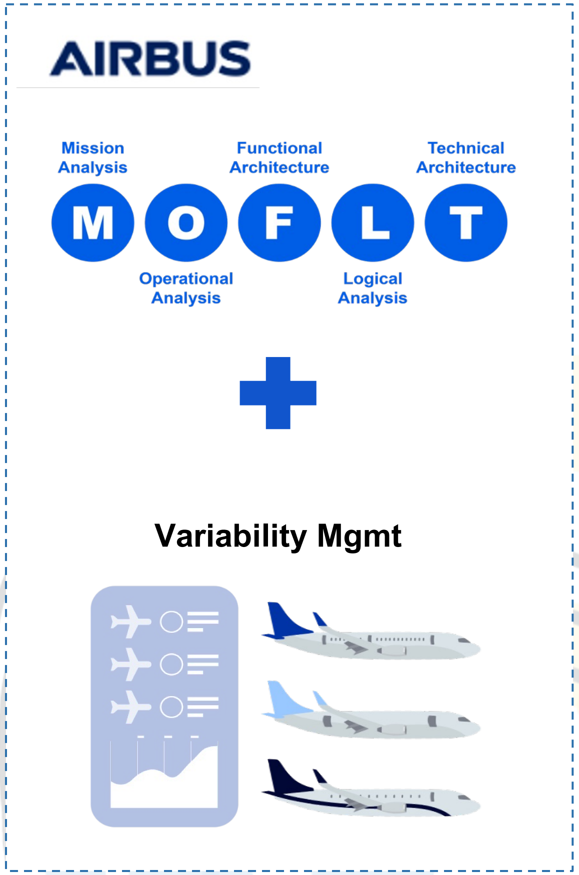
Italian based in Singapore, Marco leads at IBM the Engineering Activities in Southeast Asia. He has over 10 years of experience in the field of PLE combined with MBSE mainly within aerospace, defence, automotive and railway industries in Europe, China and North America. Before re-joining IBM, his role at Airbus consisted of developing and implementing process, methods, and tools capabilities for the model-based development of product lines across different functions and programs. Marco is co-chair of the INCOSE PLE Working Group and Assistant Director for the Asia & Oceania Sector.



Annual INCOSE international symposium

# MBPLE for MOFLT: Context and Assumptions

## Deployment of Variability at MBSE and Usage of R-MOFLT



AIRBUS

Mission Analysis

Operational Analysis

Functional Architecture

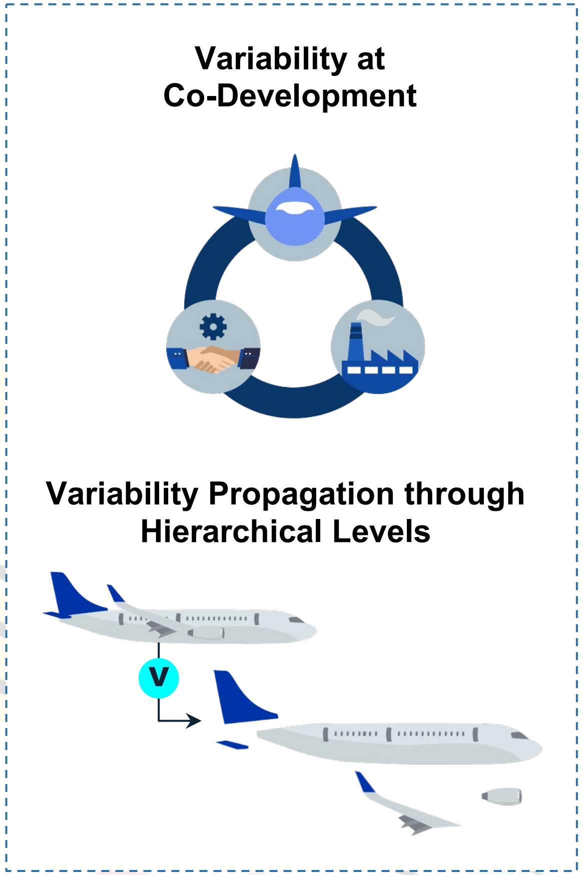
Logical Analysis

Technical Architecture

M O F L T

Variability Mgmt

## Co-Development and Multi Hierarchical Level Variability Management

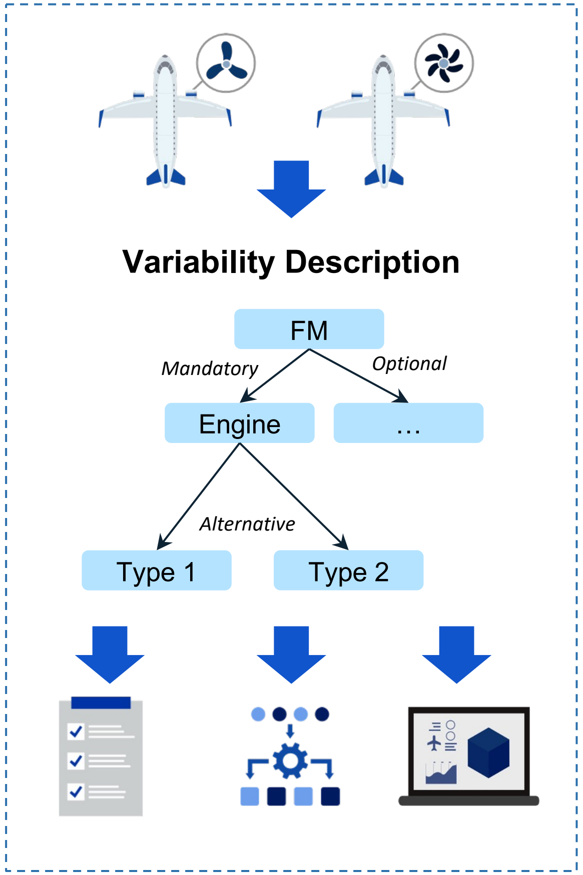


Variability at Co-Development

Variability Propagation through Hierarchical Levels

V

## Feature-Based Product Line Engineering for Holistic MBPLE Framework



Variability Description

FM

*Mandatory*

*Optional*

Engine

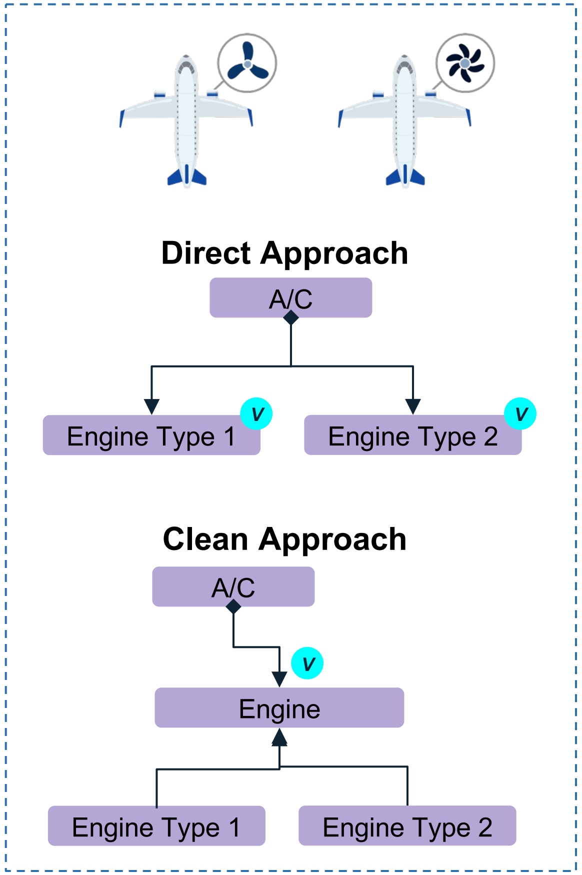
…

*Alternative*

Type 1

Type 2

## Variant Modelling Method: Direct and Clean Approaches



Direct Approach

A/C

Engine Type 1

V

Engine Type 2

V

Clean Approach

A/C

V

Engine

Engine Type 1

Engine Type 2

**End User Experience**

# R-MOFLT Layers of Abstraction

Airbus Method - MOFLT [Mission - Operation - Functional - Logical - Technical] on the MOFLT Layers of Abstraction

M

## Mission

- WHAT is the problem that we need to solve?
- WHAT are the potential ways of solving it?

O

## Operational

- WHAT will our System of Interest (SoI) do to contribute to the Mission?

F

## Functional

- HOW our SoI will work to meet the expectations?

L

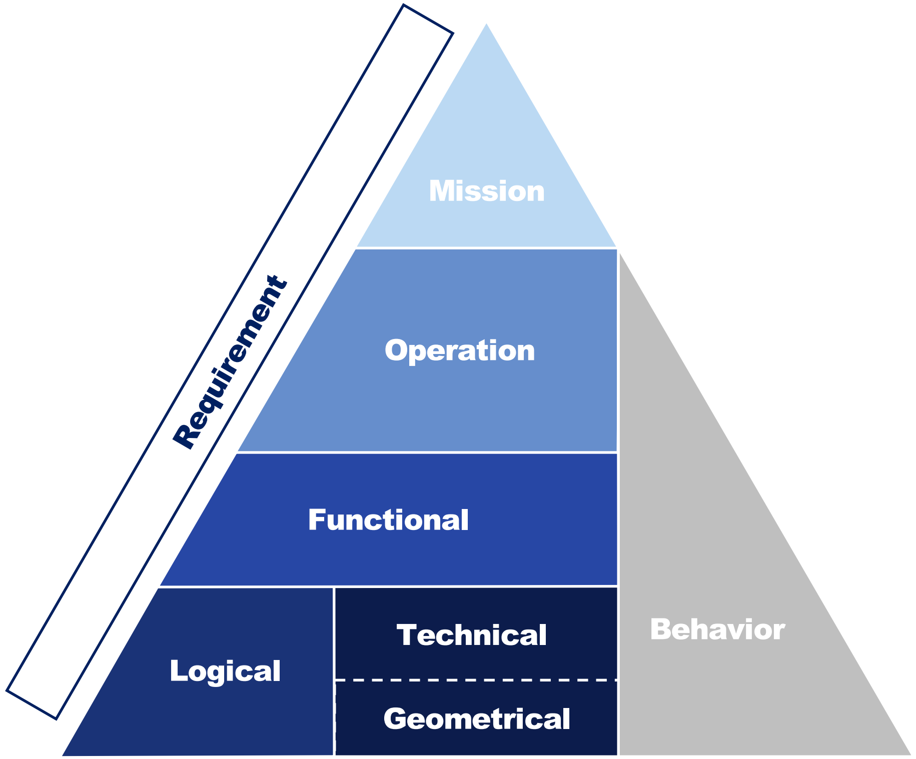
## Logical

HOW our SoI is organized?

T

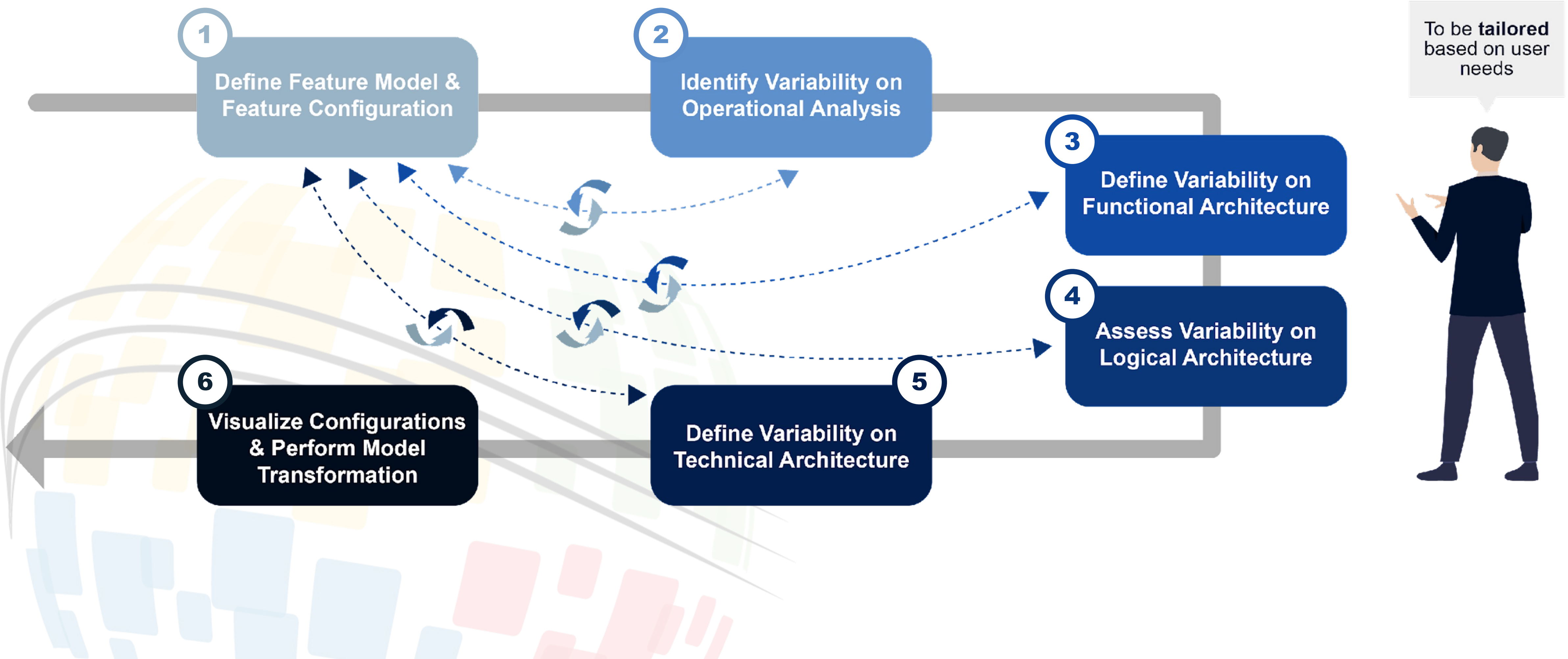
## Technical

- HOW the SoI will be implemented?



# MBPLE for MOFLT: Steps Summary

## Method Overview



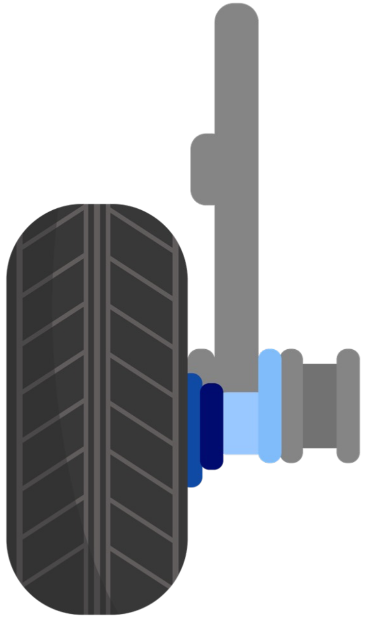
1
Define Feature Model & Feature Configuration
2
Identify Variability on Operational Analysis
3
Define Variability on Functional Architecture
4
Assess Variability on Logical Architecture
5
Define Variability on Technical Architecture
6
Visualize Configurations & Perform Model Transformation
To be **tailored** based on user needs

# Simplified Aircraft Example

Variability at Aircraft Ground Operations

Along with the classic **Engine Powered** Ground Operations, a new **Wheel Powered** option can also be offered to airlines to include in their fleet, based on:

- **Purchase Cost**
- **Operational Cost**
- **Emissions Reduction**



## NEW FUNCTIONALITIES

**Autonomous Push-Back**

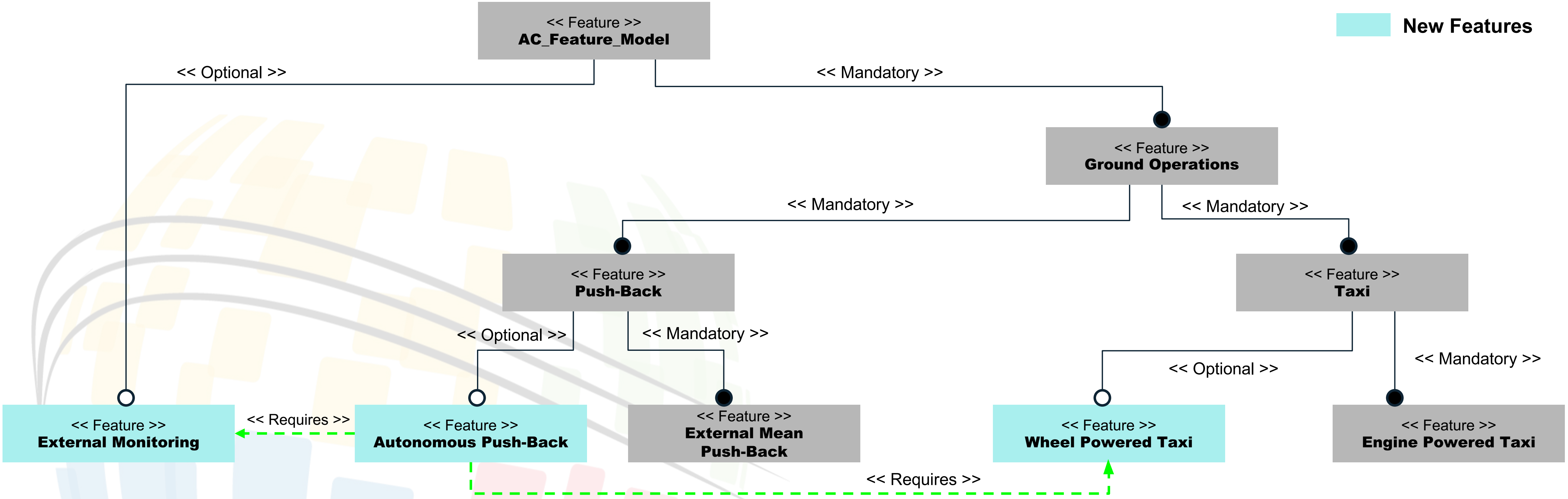
**Wheel Powered Ground Operations**

1 2 3 4 5 6

# Step 1: Feature Model

Feature-Based “decision-tree” like description of the main-drivers of variability

Initial **Feature Model** with the Features describing the new options for **Wheel Powered Ground Operations**



🟦 **New Features**

<< Feature >> **AC_Feature_Model**

<< Optional >>

<< Mandatory >>

<< Feature >> **Ground Operations**

<< Mandatory >>

<< Mandatory >>

<< Feature >> **Push-Back**

<< Feature >> **Taxi**

<< Optional >>

<< Mandatory >>

<< Optional >>

<< Mandatory >>

<< Feature >> **External Monitoring**

<< Requires >>

<< Feature >> **Autonomous Push-Back**

<< Feature >> **External Mean Push-Back**

<< Feature >> **Wheel Powered Taxi**

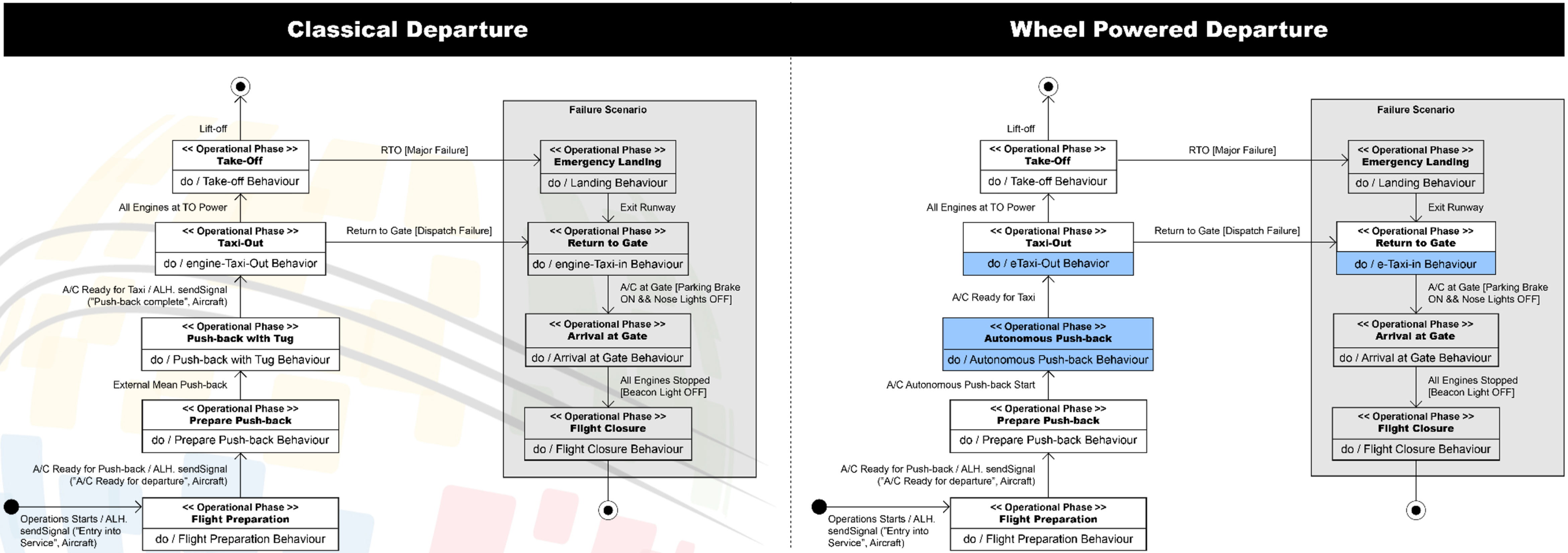
<< Feature >> **Engine Powered Taxi**

<< Requires >>

# Step 2: Operational Analysis

## Identifying Variability in the Operational Analysis

The new **Autonomous Push-Back** and **Wheel Powered Taxi** features has a significant impact on Ground Operations. This variability is captured by having **two distinct Operational Concepts**.



**Classical Departure**

Lift-off

<< Operational Phase >> **Take-Off**
do / Take-off Behaviour

RTO [Major Failure]

All Engines at TO Power

<< Operational Phase >> **Taxi-Out**
do / engine-Taxi-Out Behavior

Return to Gate [Dispatch Failure]

A/C Ready for Taxi / ALH. sendSignal ("Push-back complete", Aircraft)

<< Operational Phase >> **Push-back with Tug**
do / Push-back with Tug Behaviour

External Mean Push-back

<< Operational Phase >> **Prepare Push-back**
do / Prepare Push-back Behaviour

A/C Ready for Push-back / ALH. sendSignal ("A/C Ready for departure", Aircraft)

Operations Starts / ALH. sendSignal ("Entry into Service", Aircraft)

<< Operational Phase >> **Flight Preparation**
do / Flight Preparation Behaviour

Failure Scenario

<< Operational Phase >> **Emergency Landing**
do / Landing Behaviour

Exit Runway

<< Operational Phase >> **Return to Gate**
do / engine-Taxi-in Behaviour

A/C at Gate [Parking Brake ON && Nose Lights OFF]

<< Operational Phase >> **Arrival at Gate**
do / Arrival at Gate Behaviour

All Engines Stopped [Beacon Light OFF]

<< Operational Phase >> **Flight Closure**
do / Flight Closure Behaviour

**Wheel Powered Departure**

Lift-off

<< Operational Phase >> **Take-Off**
do / Take-off Behaviour

RTO [Major Failure]

All Engines at TO Power

<< Operational Phase >> **Taxi-Out**
do / eTaxi-Out Behavior

Return to Gate [Dispatch Failure]

A/C Ready for Taxi

<< Operational Phase >> **Autonomous Push-back**
do / Autonomous Push-back Behaviour

A/C Autonomous Push-back Start

<< Operational Phase >> **Prepare Push-back**
do / Prepare Push-back Behaviour

A/C Ready for Push-back / ALH. sendSignal ("A/C Ready for departure", Aircraft)

Operations Starts / ALH. sendSignal ("Entry into Service", Aircraft)

<< Operational Phase >> **Flight Preparation**
do / Flight Preparation Behaviour

Failure Scenario

<< Operational Phase >> **Emergency Landing**
do / Landing Behaviour

Exit Runway

<< Operational Phase >> **Return to Gate**
do / e-Taxi-in Behaviour

A/C at Gate [Parking Brake ON && Nose Lights OFF]

<< Operational Phase >> **Arrival at Gate**
do / Arrival at Gate Behaviour

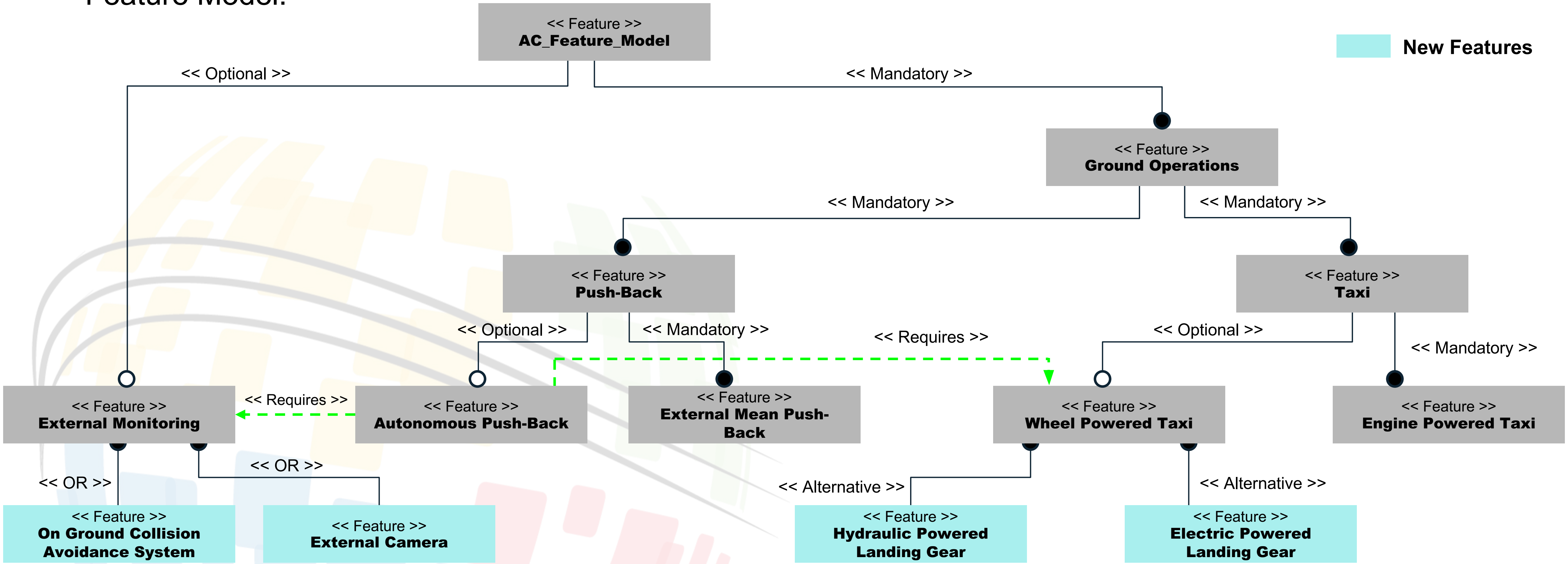
All Engines Stopped [Beacon Light OFF]

<< Operational Phase >> **Flight Closure**
do / Flight Closure Behaviour

# Step 3: Functional Analysis

## Defining Variability in the Functional Architecture

During the Functional Analysis, **solutions** and **architectural decisions** prompted the extension of the Feature Model.

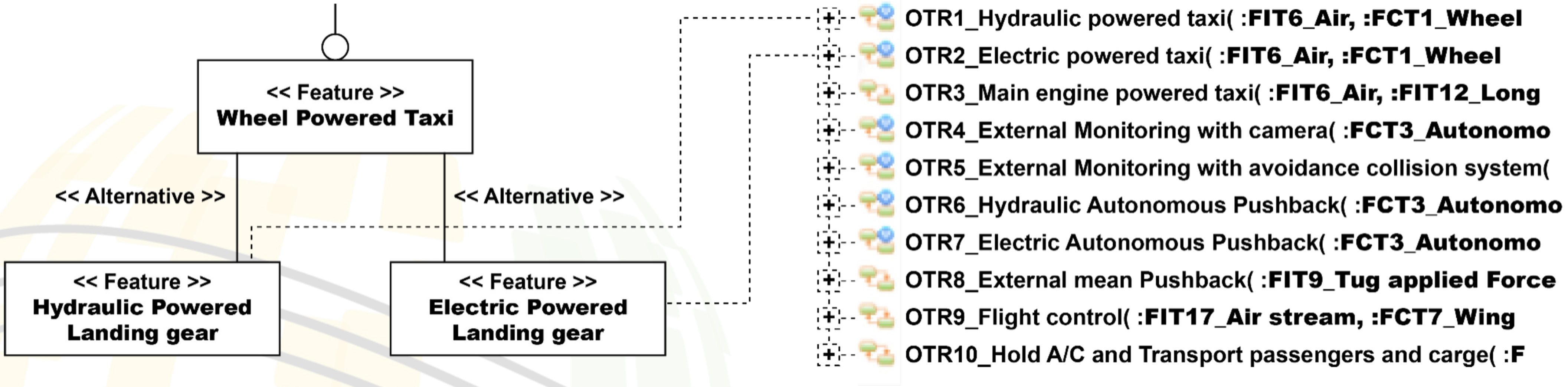


<< Feature >> AC_Feature_Model
🟦 New Features
<< Optional >>
<< Mandatory >>
<< Feature >> Ground Operations
<< Mandatory >>
<< Mandatory >>
<< Feature >> Push-Back
<< Feature >> Taxi
<< Optional >>
<< Mandatory >>
<< Requires >>
<< Optional >>
<< Mandatory >>
<< Feature >> External Monitoring
<< Requires >>
<< Feature >> Autonomous Push-Back
<< Feature >> External Mean Push-Back
<< Feature >> Wheel Powered Taxi
<< Feature >> Engine Powered Taxi
<< OR >>
<< OR >>
<< Alternative >>
<< Alternative >>
<< Feature >> On Ground Collision Avoidance System
<< Feature >> External Camera
<< Feature >> Hydraulic Powered Landing Gear
<< Feature >> Electric Powered Landing Gear

# Step 3: Functional Analysis

## Defining Variability in the Functional Architecture

Operational Task Realisations are defined to cover **all variants SOI Operational Tasks**, common and variable.



OTR1_Hydraulic powered taxi( **:FIT6_Air, :FCT1_Wheel**

OTR2_Electric powered taxi( **:FIT6_Air, :FCT1_Wheel**

OTR3_Main engine powered taxi( **:FIT6_Air, :FIT12_Long**

OTR4_External Monitoring with camera( **:FCT3_Autonomo**

OTR5_External Monitoring with avoidance collision system(

OTR6_Hydraulic Autonomous Pushback( **:FCT3_Autonomo**

OTR7_Electric Autonomous Pushback( **:FCT3_Autonomo**

OTR8_External mean Pushback( **:FIT9_Tug applied Force**

OTR9_Flight control( **:FIT17_Air stream, :FCT7_Wing**

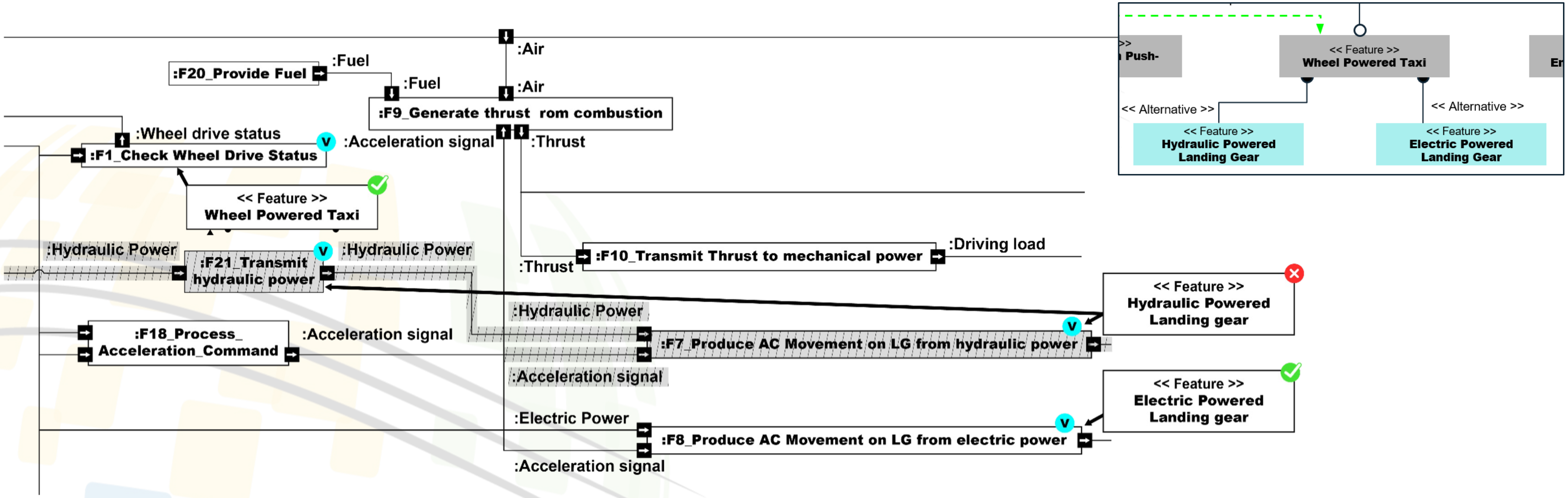
OTR10_Hold A/C and Transport passengers and carge( **:F**

**Variation Points** are formal links between the **Features** and **Model Elements** identified by the blue circle icon with a white V.

# Step 3: Functional Analysis

## Defining Variability in the Functional Architecture

The **150% Functional Architecture** includes the **Functions** realising **both Hydraulic** and **Electric Powered Landing Gear**. Feature Selection determines the changes that will be applied after model transformation.

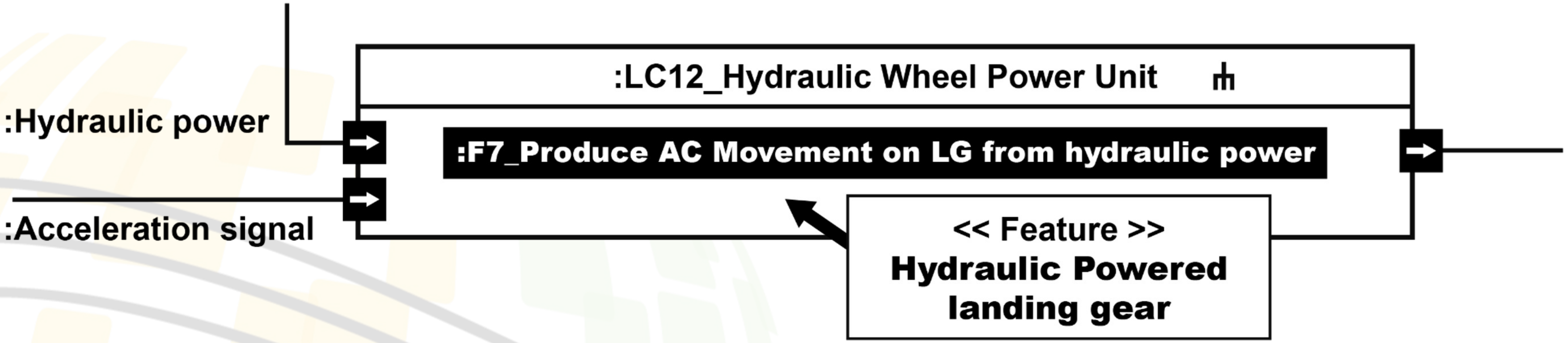


**150% Architectures** might be semantically wrong. In the example, the same **wheel cannot concurrently** have a **Hydraulic** and **Electric power source**.

# Step 4: Logical Analysis

## Assessing Variability in the Logical Analysis

The **150% Logical Architecture** include all Logical Components required to allocate all Functions. **Variability** here is mostly **assessed**, not formally defined by the inclusion of Variation Points. **Variation Points** are only added to the **Functional Parts** themselves, when their **allocation** is **variable**.


:LC12_Hydraulic Wheel Power Unit
:Hydraulic power
:Acceleration signal
:F7_Produce AC Movement on LG from hydraulic power
<< Feature >>
Hydraulic Powered landing gear

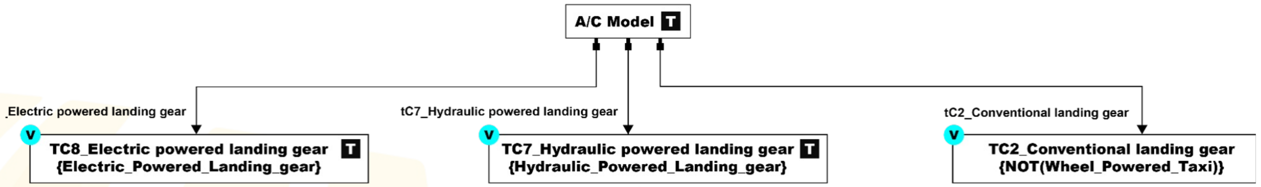
**By displaying functions** in the Logical Components, the variability of the logical architecture can then be assessed even before any model transformation, bringing excellent value to the architect.

# Step 5: Technical Analysis

## Defining Variability in the Technical Analysis

The **150% Technical Architecture** can be built using two distinct approaches: the **Direct** or the **Clean**. Each has its own advantages and disadvantages, depending on the use-case or stage of the modelling.

**Direct Approach**



A/C Model T

.Electric powered landing gear

tC7_Hydraulic powered landing gear

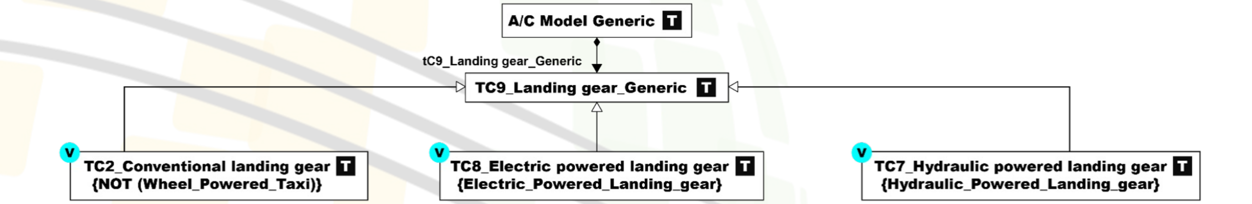
tC2_Conventional landing gear

V TC8_Electric powered landing gear T {Electric_Powered_Landing_gear}

V TC7_Hydraulic powered landing gear T {Hydraulic_Powered_Landing_gear}

V TC2_Conventional landing gear {NOT(Wheel_Powered_Taxi)}

**Clean Approach**



A/C Model Generic T

tC9_Landing gear_Generic

TC9_Landing gear_Generic T

V TC2_Conventional landing gear T {NOT (Wheel_Powered_Taxi)}

V TC8_Electric powered landing gear T {Electric_Powered_Landing_gear}

V TC7_Hydraulic powered landing gear T {Hydraulic_Powered_Landing_gear}

**Direct Approach:** all three alternative landing gear components belong directly to the Architecture.

**Clean Approach:** the three alternative landing gear designs are a specialisation of a generic landing gear.

**Clean Approach** is an alternative way to incrementally build the Technical Architecture starting from a 100% generic architecture. Then, the 100% variant architectures are incrementally built on top by applying SysML realisation relation between the generic and variant architectures. Finally, either the additional variable technical components are included, or the SysML Property Redefinition is applied to replace the generic components with specific ones, which are linked to the generic component by SysML Realisation

# Step 6: Model Transformation

## Realise Variants and Clean-Up

Perform **model transformation** to **generate 100% models** representing the variants defined by feature selection.

**STEP 1**

Select the **Feature Configuration** representing one **variant**.

**STEP 2**

Perform the Model Transformation to generate a 100% model.

**STEP 3**

Clean-up remaining variable elements that did not have Variation Points applied to them (e.g. Operational and Logical layers)

**Pre-Visualisation** is key during the modelling steps, aiding the modellers to to **keep consistent** and **take decisions** regarding variability.

# Deployment at Airbus

## Beyond the "Traditional" System Development

Product

Industrial

Services

**Successfully Deployed at the Three Axis of Co-Development**

## Proven to Support Diverse Types of Systems and Hierarchical Levels

A/C Level

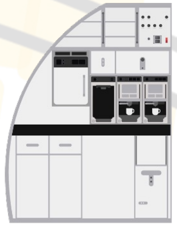
FAL

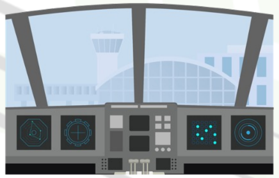
A/C Components & Assemblies

Sub-Systems

**Method proven applicable to different domains**

**Method already delivering value to real-life projects for 1.5+years**





## From Proof of Concept to Real-Life Deployment

**Method developed accounting to the need of the whole Airbus Group**

AIRBUS

Commercial Aircrafts

Helicopters

Defense & Space

## Covering the Needs of Airbus "Big Family"

# Conclusion and Future Development

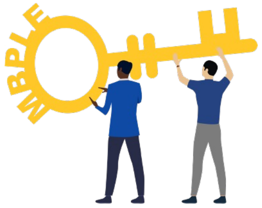
# Takeaways

**MBPLE 4MOFLT**

a solution for

***Architecture Variability Management***

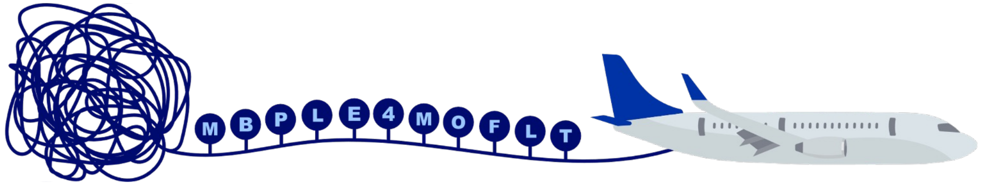
at AIRBUS



**MBSE** combined with **PLE** is a key enabler for **Systematic Reuse** and **Complexity Management**

operations functional logical/technical

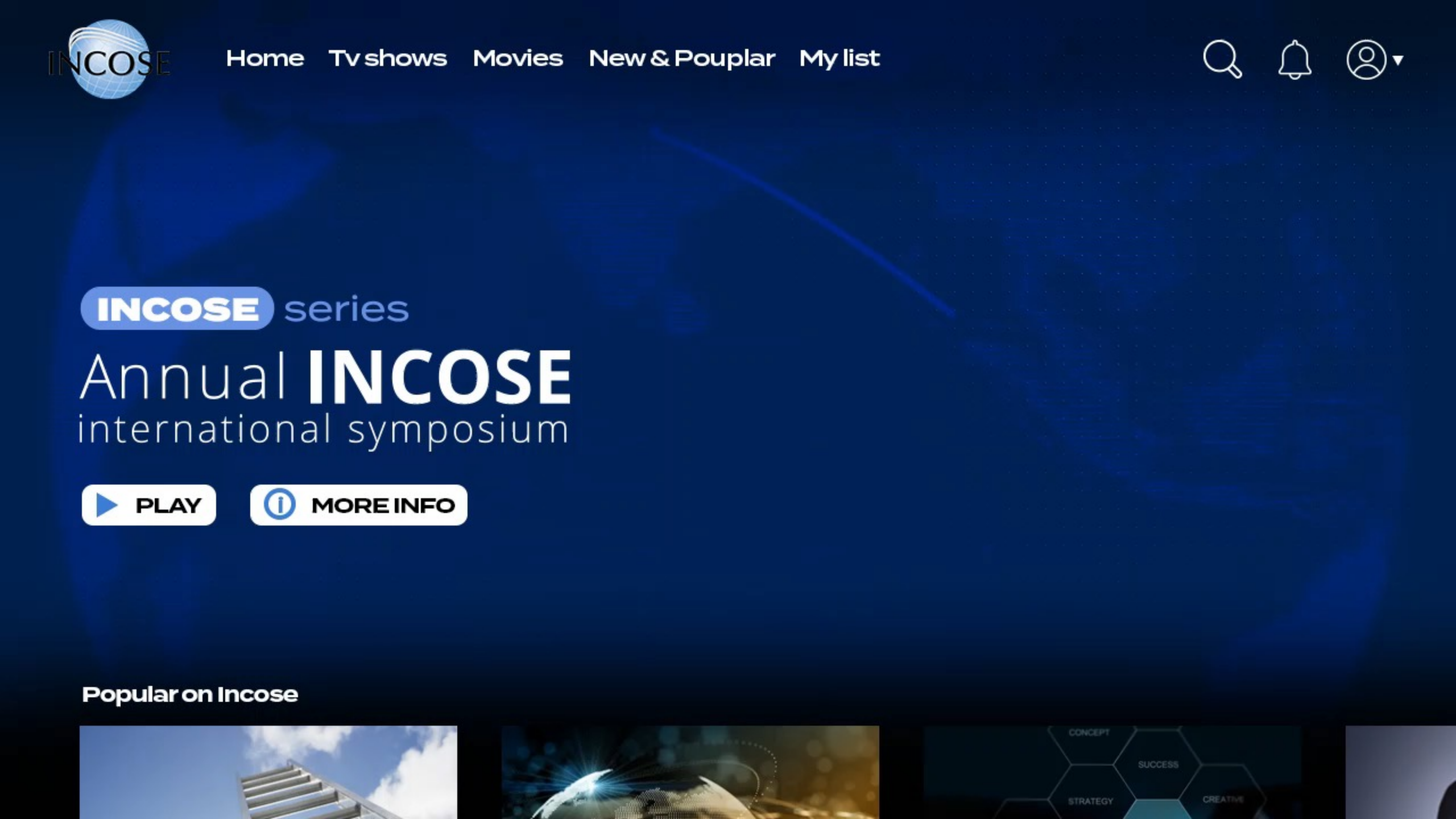
**MBPLE4MOFLT** covers Variability Identification, Definition and Assessment from **Operations** of the System up to the **Functional** and **Logical/Technical** Architectures definitions, with consistent Variability cascade through layers



**MBPLE4MOFLT** was shaped to **simplify** its **deployment and focus on core variability**, **reducing workload** and **consistency issues**, even if this might require post-transformation clean-up

**MBPLE4MOFLT** has been **deployed** for **1.5+ years** in **Airbus** in different **Domains**, **Systems Types** and **Hierarchical Levels**

Adding variability at **MOLFT** is a key enabler for efficient **Product Lines**, **Customization**, **Architectural Trades** at **Airbus**



Annual INCOSE international symposium

Popular on Incose

# Q&A

33rd Annual INCOSE
international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023