



# Not (strictly) relying on SysML for Model-Based Systems Engineering

LANGUAGE, TOOLING AND DEVELOPMENT PERSPECTIVES

S. Bonnet, D. Exertier, J.-L. Voirin, V. Normand  
IEEE SysCon, April 19<sup>th</sup>, 2016

[www.thalesgroup.com](http://www.thalesgroup.com)



# Thales : A Wide Spectrum of Complex Systems

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.

N°1  
worldwide



Payloads  
for telecom satellites



Air Traffic Management



Sonars



Security for interbank  
transactions

N°2  
worldwide



Rail signalling systems



In-flight entertainment  
and connectivity

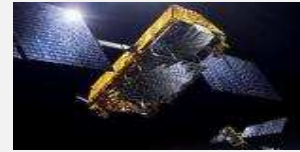


Military tactical  
radiocommunications

N°3  
worldwide



Avionics



Civil satellites

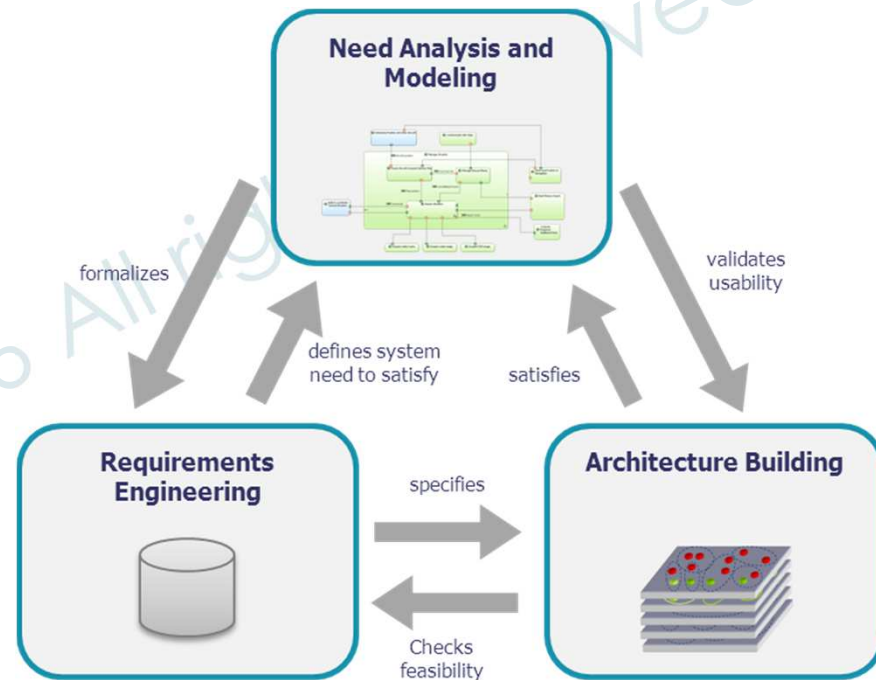


Surface radars

# Model-Based Systems Engineering vs Traditional Systems Engineering

## MBSE does not replace standard Systems Engineering practices

- It formalizes parts of systems engineering
- Combines traditional methods and best practices with rigorous modeling techniques



# Models, What For?

## Answer questions

- About the system
  - What is it, how does it work, is the performance adequate, what happens if something breaks?
- About the design
  - Is it complete, does it support required analyses, does it support impact analysis?

## Ensure consistency

- Across different views, between upstream and downstream engineering, etc.

## Generate artefacts

- Documentation (specification, architecture, interfaces)
- Pieces of code, database schemas, configuration data, deployment data, etc.



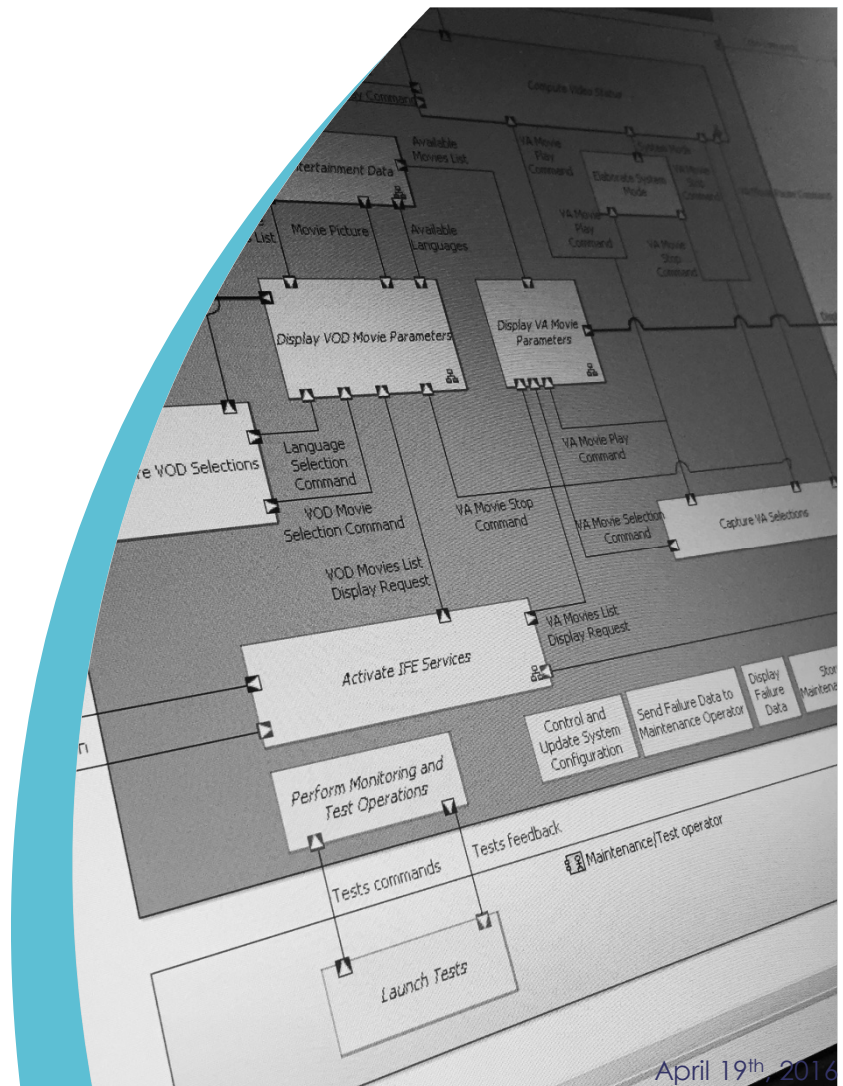
Choose (and adapt)  
the right modeling  
solution for your  
objectives!



## Arcadia and Capella

MODEL-BASED METHOD FOR ARCHITECTURAL DESIGN  
AND ITS SUPPORTING **OPEN SOURCE** MODELING  
WORKBENCH

[www.thalesgroup.com](http://www.thalesgroup.com)

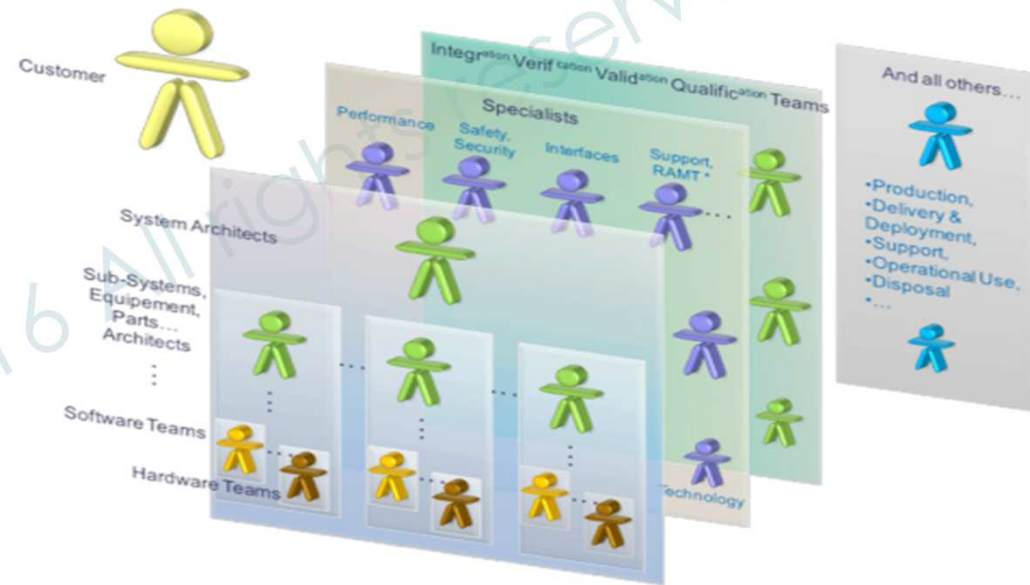


April 19th 2016

# Arcadia : MBSE Scalable and Adaptable Method

## Improving engineering agility and overall performance

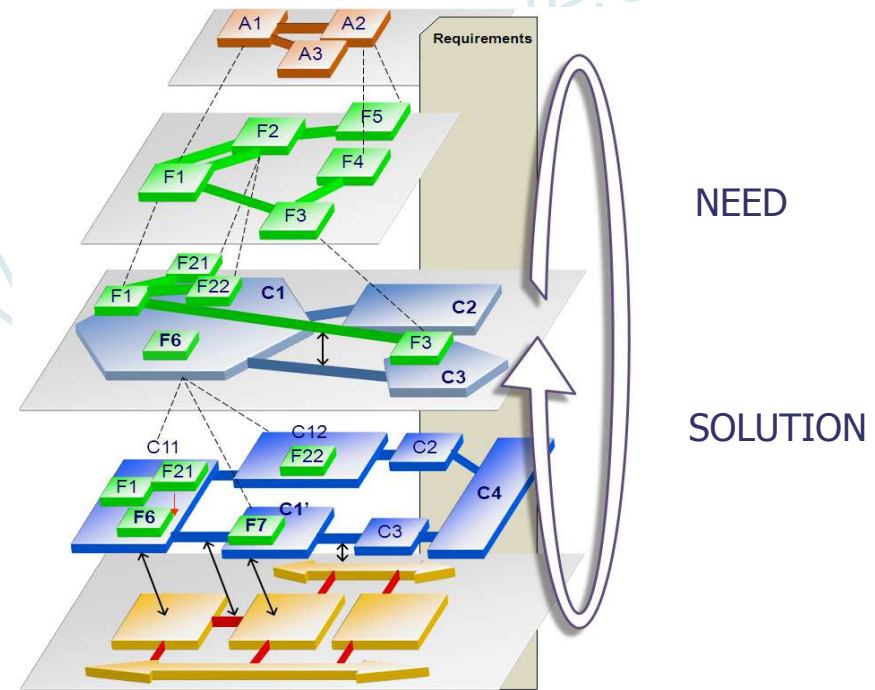
➤ System-wide collaboration



# Arcadia : MBSE Scalable and Adaptable Method

## Improving engineering agility and overall performance

- System-wide collaboration
- Complexity mastering



© Thales - 2016 A'

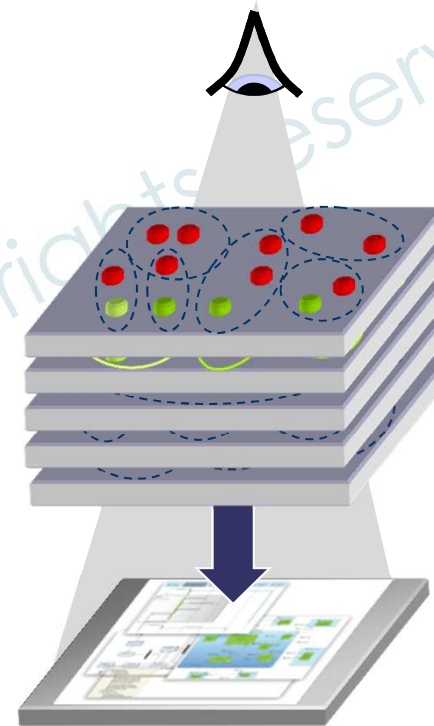




# Arcadia : MBSE Scalable and Adaptable Method

## Improving engineering agility and overall performance

- System-wide collaboration
- Complexity mastering
- Concurrent engineering



### ViewPoints

etc.  
Product Line  
Human Factors  
Performance  
Security  
Safety



Evaluation Rules

**Solution  
Architecture**



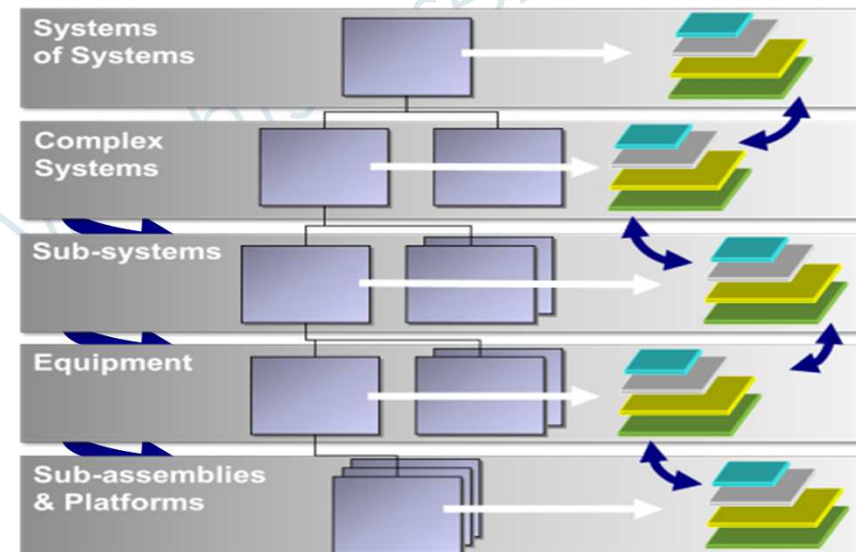
ARCADIA

**THALES**  
April 19<sup>th</sup>, 2016

# Arcadia : MBSE Scalable and Adaptable Method

## Improving engineering agility and overall performance

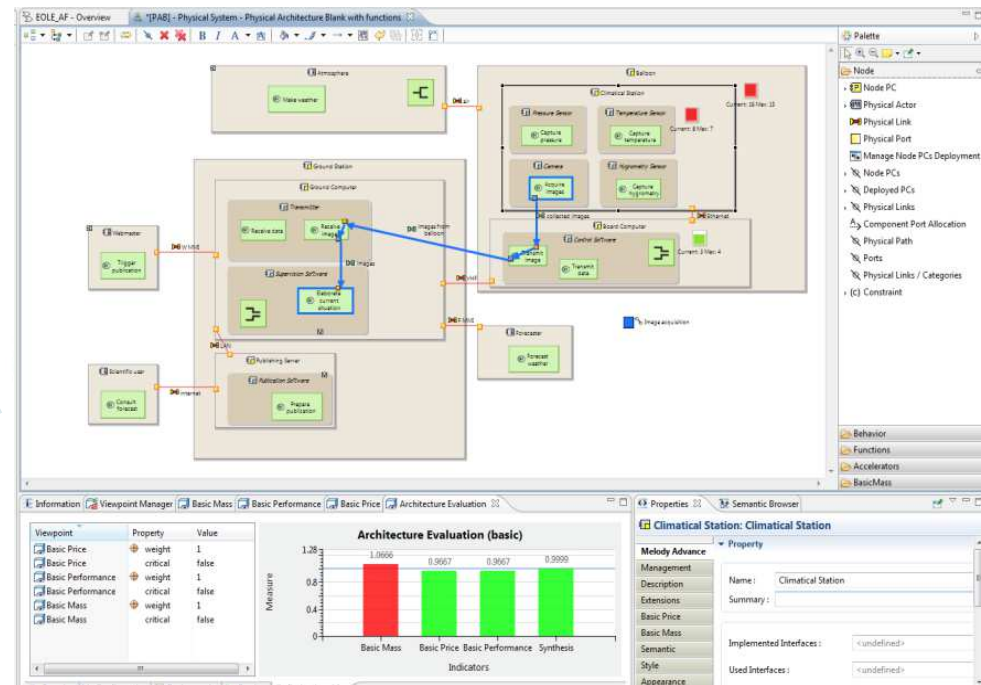
- System-wide collaboration
- Complexity mastering
- Concurrent engineering
- Mastering transitions



# Capella: An Open Source Modeling Workbench Supporting Arcadia

## Not a talk about Capella features, but....

- Methodological browser
- Semantic browser
- Computed graphical views
- Advanced diagram mgt.
- Validation & quick fixes
- Semantic delete
- Replicable elements
- Patterns
- HTML generation
- Transition to sub-systems
- Multi-viewpoint mgt.



This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.

# THALES

## Capella ~~versus~~ and SysML

NOT A DSML BUT A HYBRID APPROACH

[www.thalesgroup.com](http://www.thalesgroup.com)



April 19<sup>th</sup>, 2016

## Preamble

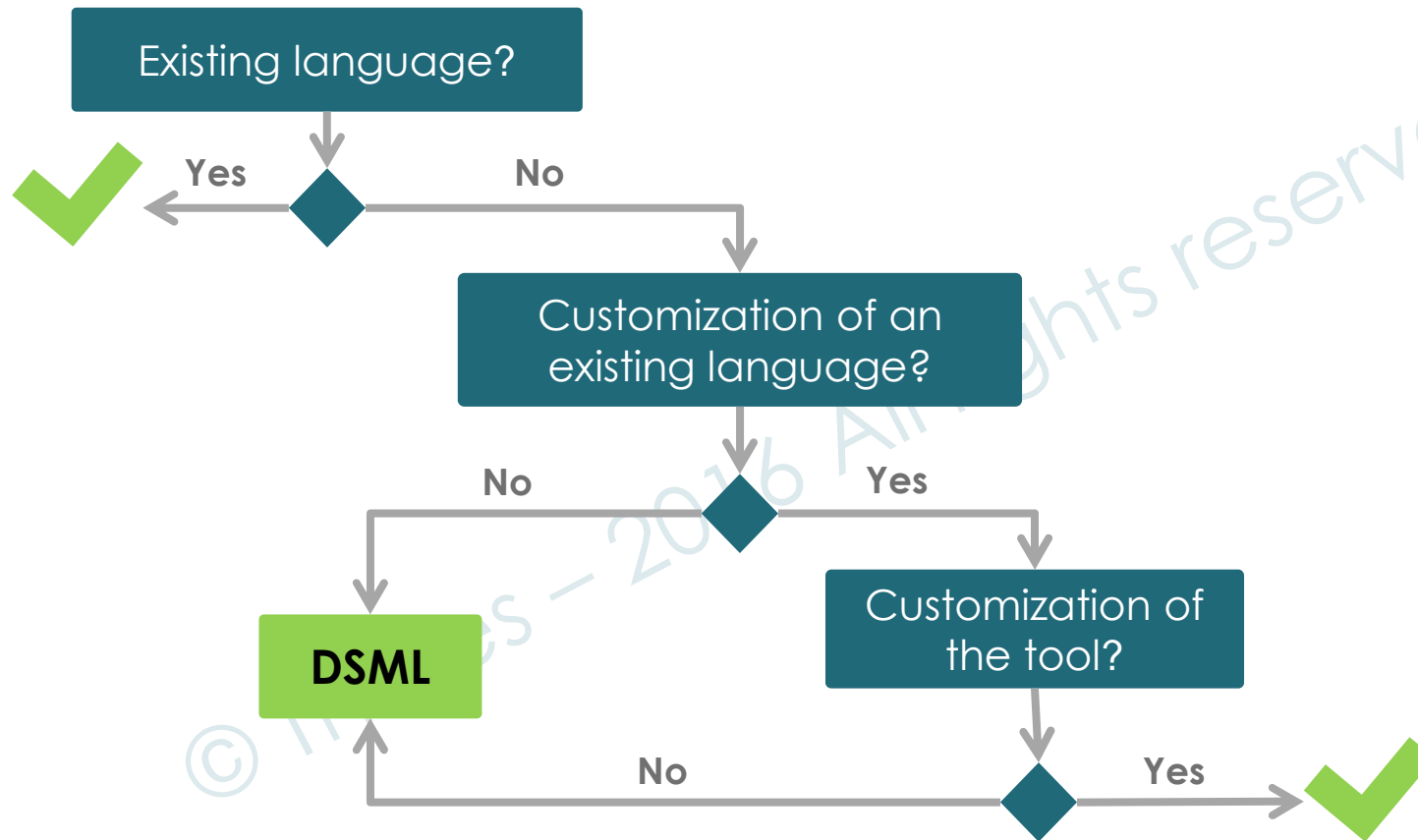
### OMG Systems Modeling Language SysML™

- **General-purpose** graphical modeling language for specifying, analyzing, designing, and **verifying complex systems** that may include hardware, software, information, personnel, procedures, and facilities. It provides **graphical representations with a semantic foundation for modeling system: requirements, behavior, structure, parametrics**
- Extends a subset of OMG Unified Modeling Language (OMG UML™) version 2

### DSML (Domain-Specific Modeling Language)

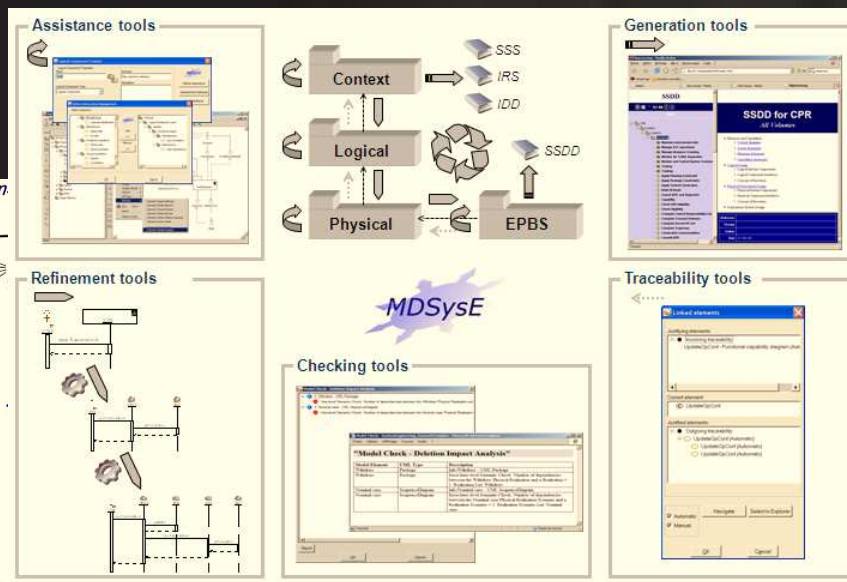
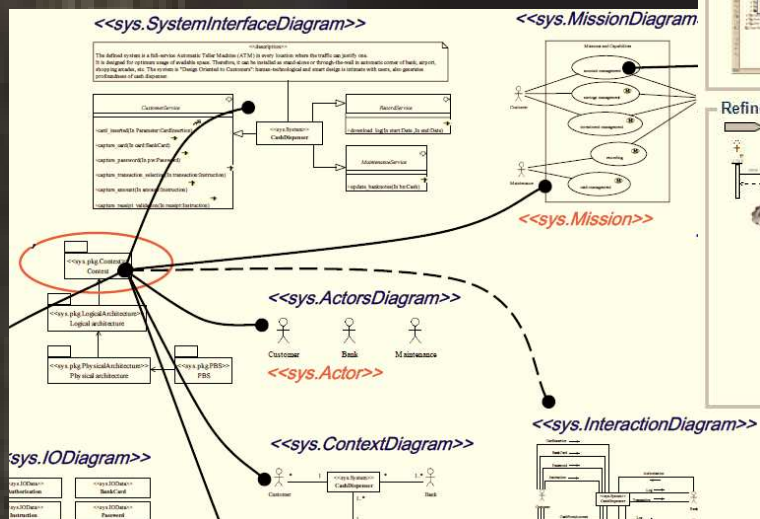
- **Specialized** modeling languages intended to provide solutions for **particular domains**. They typically have **reduced coverage and more focused intentions**

# Tooling a Model-Based Engineering Method

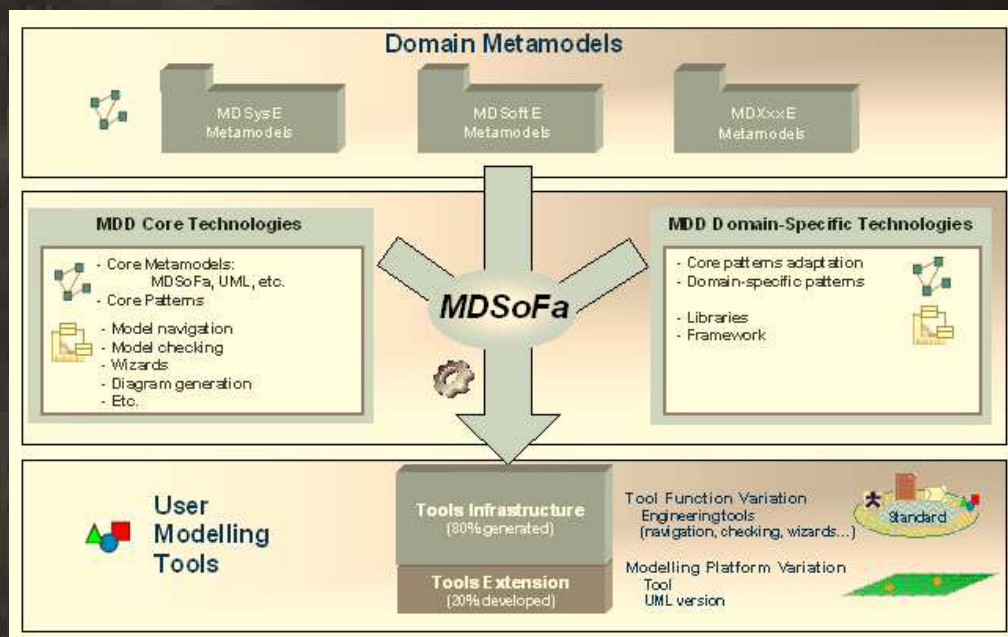


This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.

# Back in the past (2003-2008)



# Back in the past (2003-2008)



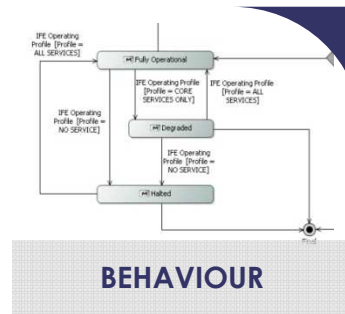
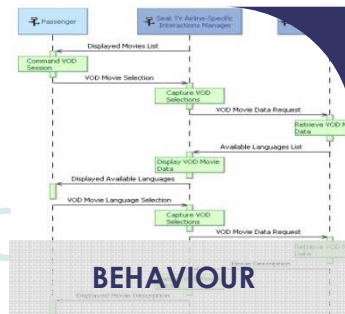
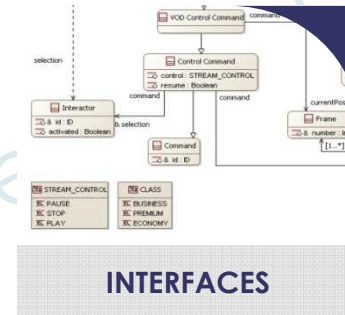
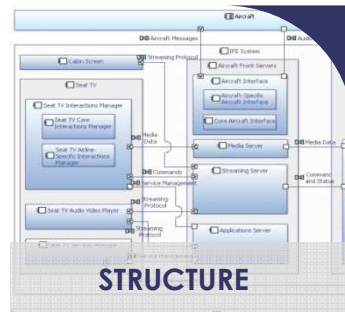
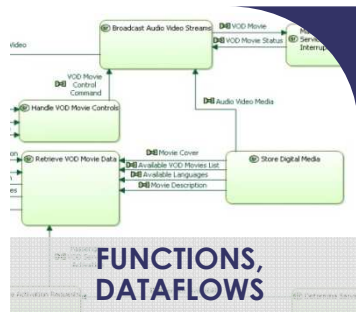
“EMF” outside  
Java & Eclipse

:-)



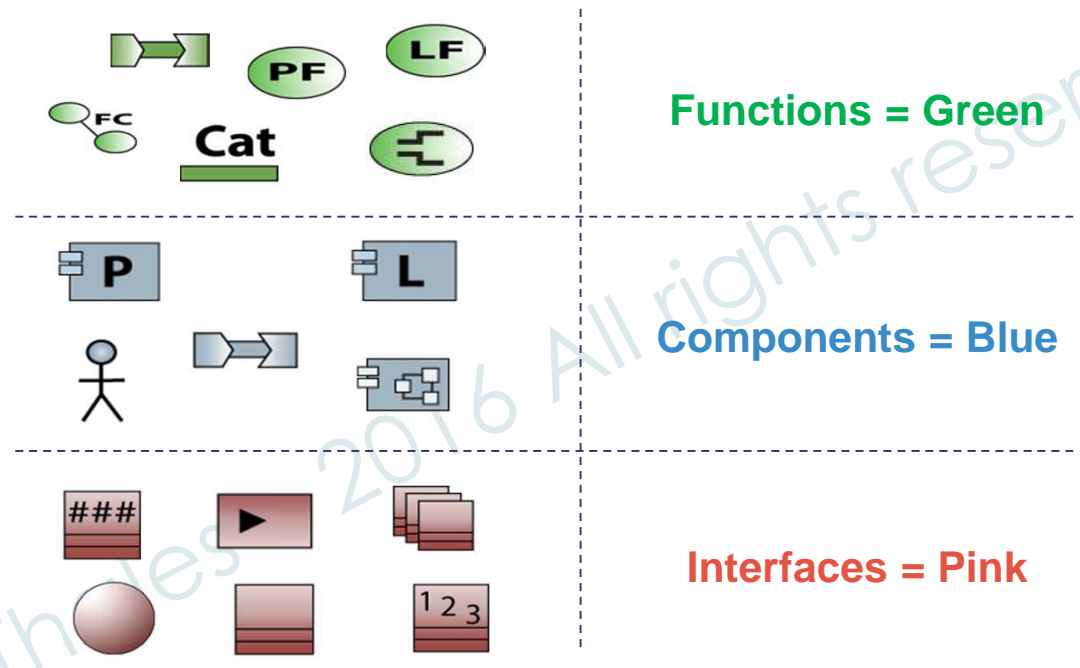
# Capella Core Concepts: The Wheel is Not Reinvented...

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales - 2015. All rights reserved.



# ... Things Are Just Simpler

... when possible

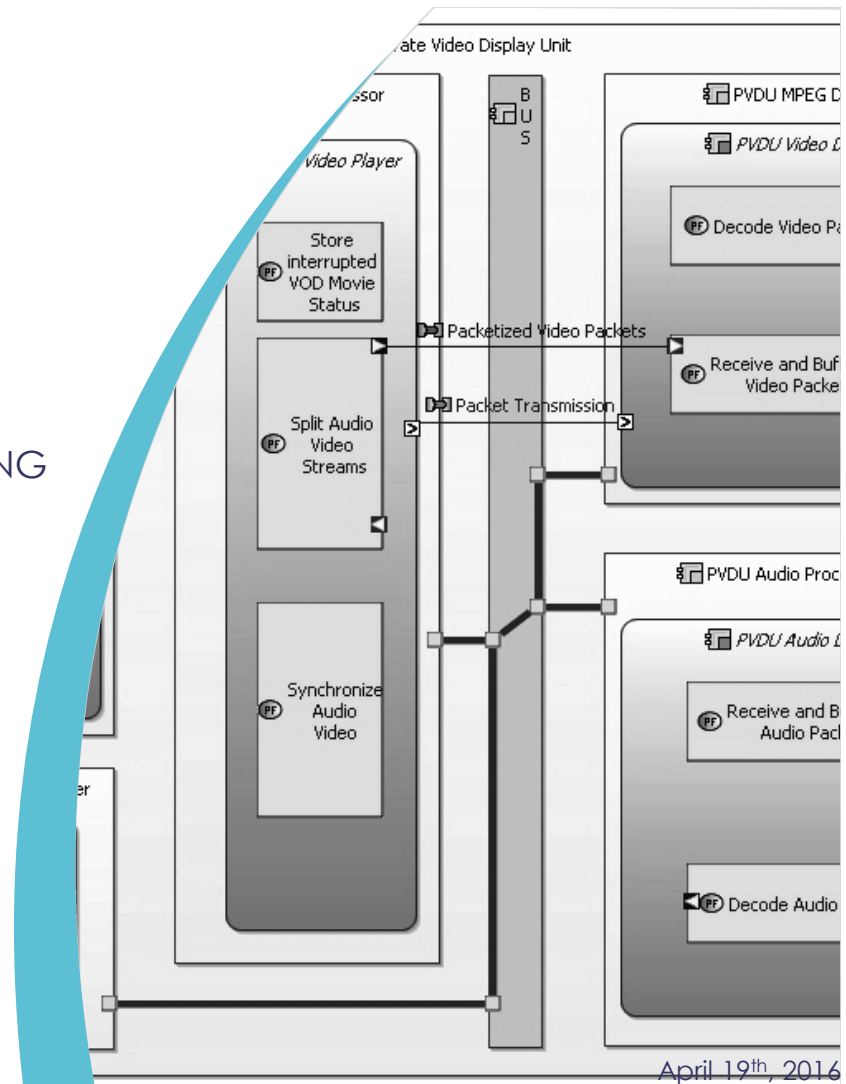


This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.

© Thales 2016 All rights reserved

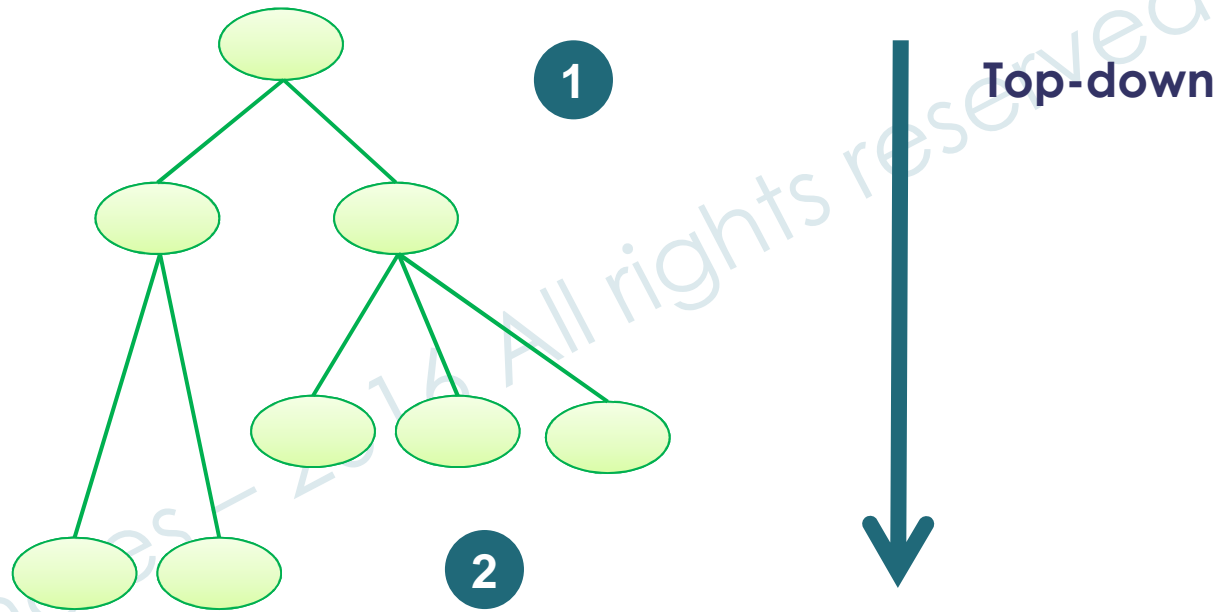
## Two examples of differences

FUNCTIONAL ANALYSIS AND INSTANCE-DRIVEN MODELING



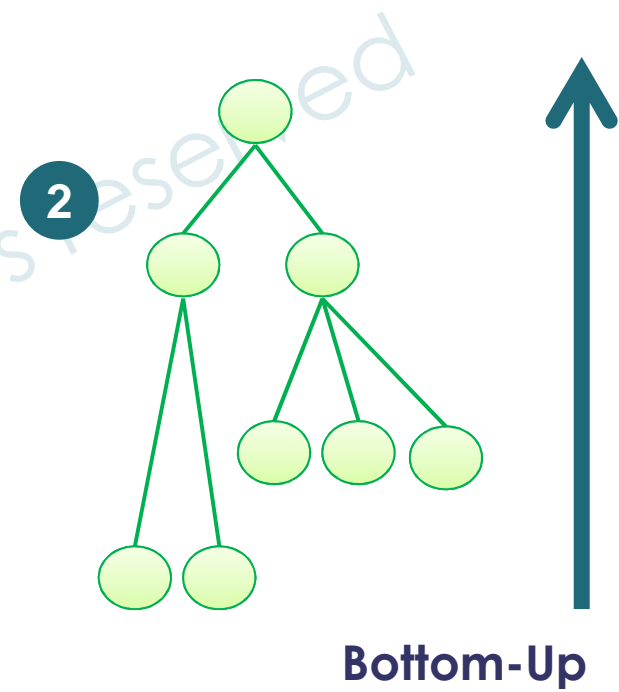
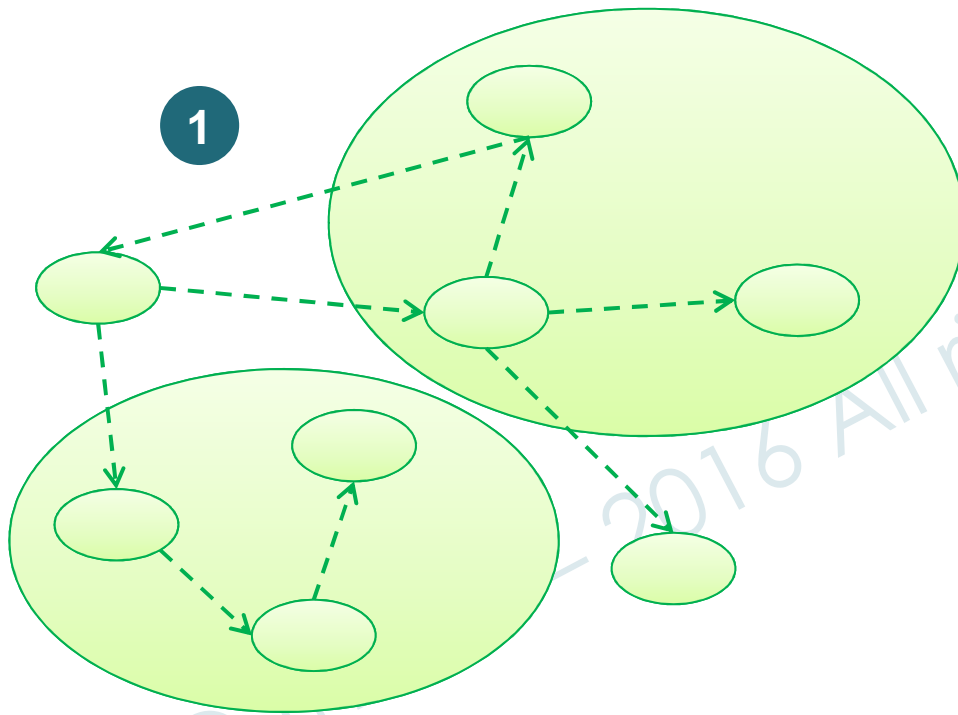
# Functional Analysis Workflows

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.



# Functional Analysis Workflows

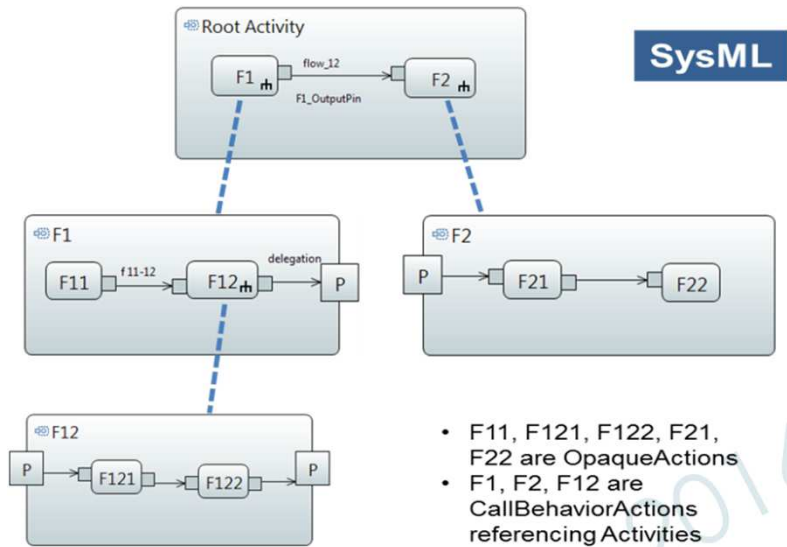
This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.



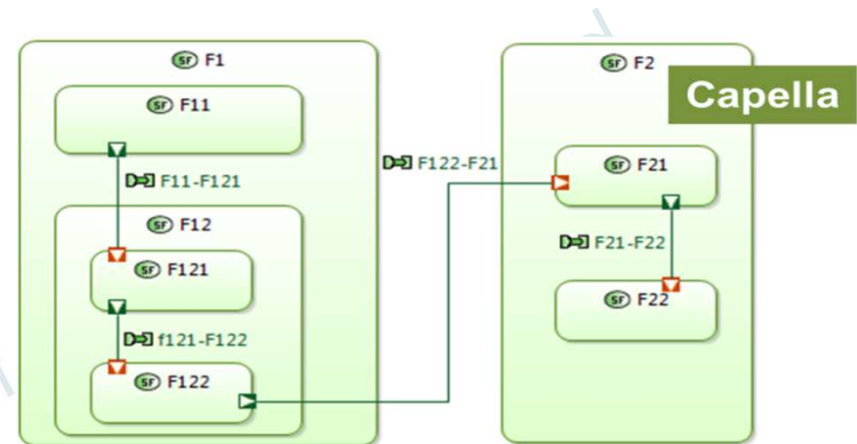


# SysML Activity Diagrams vs Capella Functions

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.

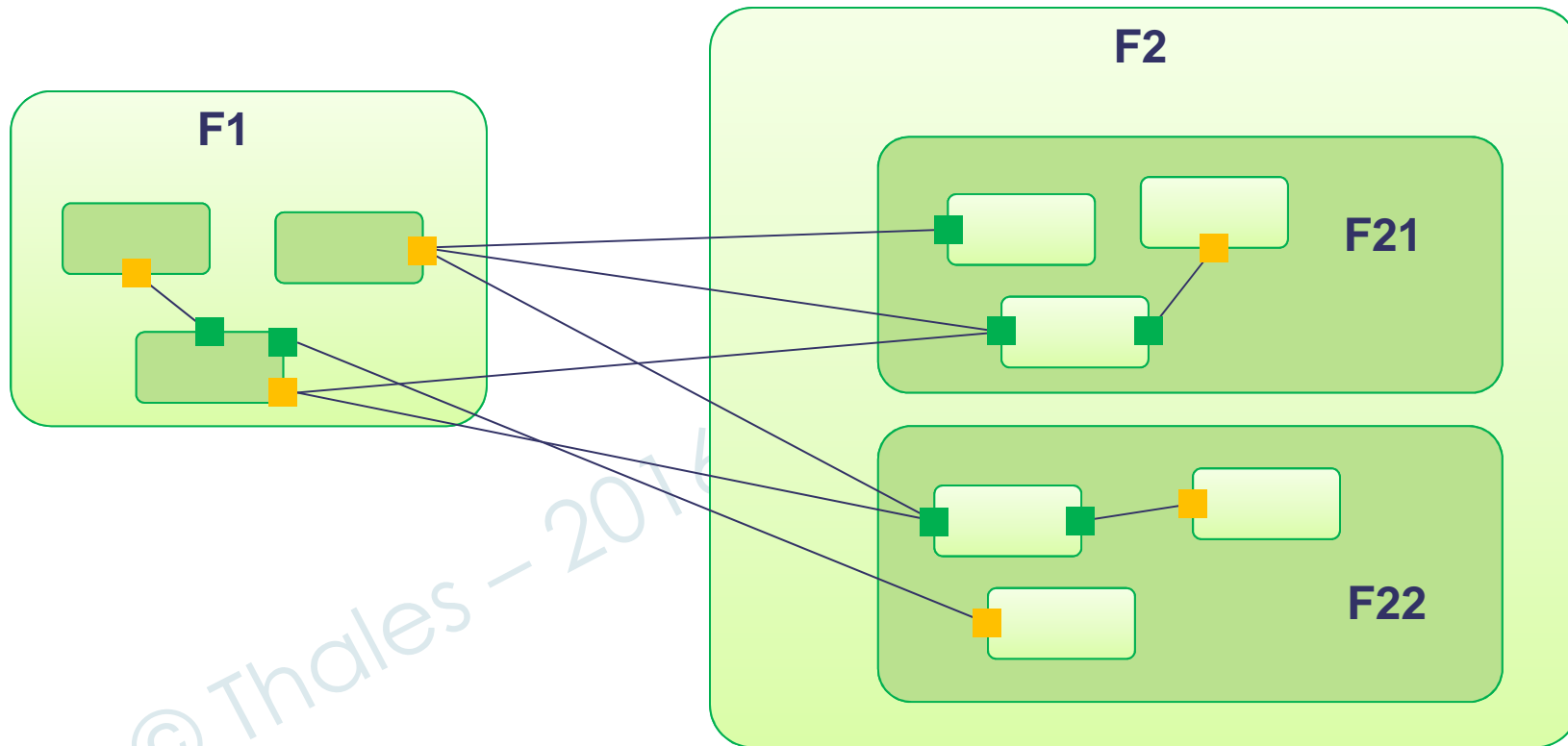


**Rigid encapsulation and delegation mechanism, with three different kinds of « functions »**



**No delegation. When the design is complete, only leaf functions are supposed to have incoming/outgoing exchanges.**

# Functional Analysis with Capella



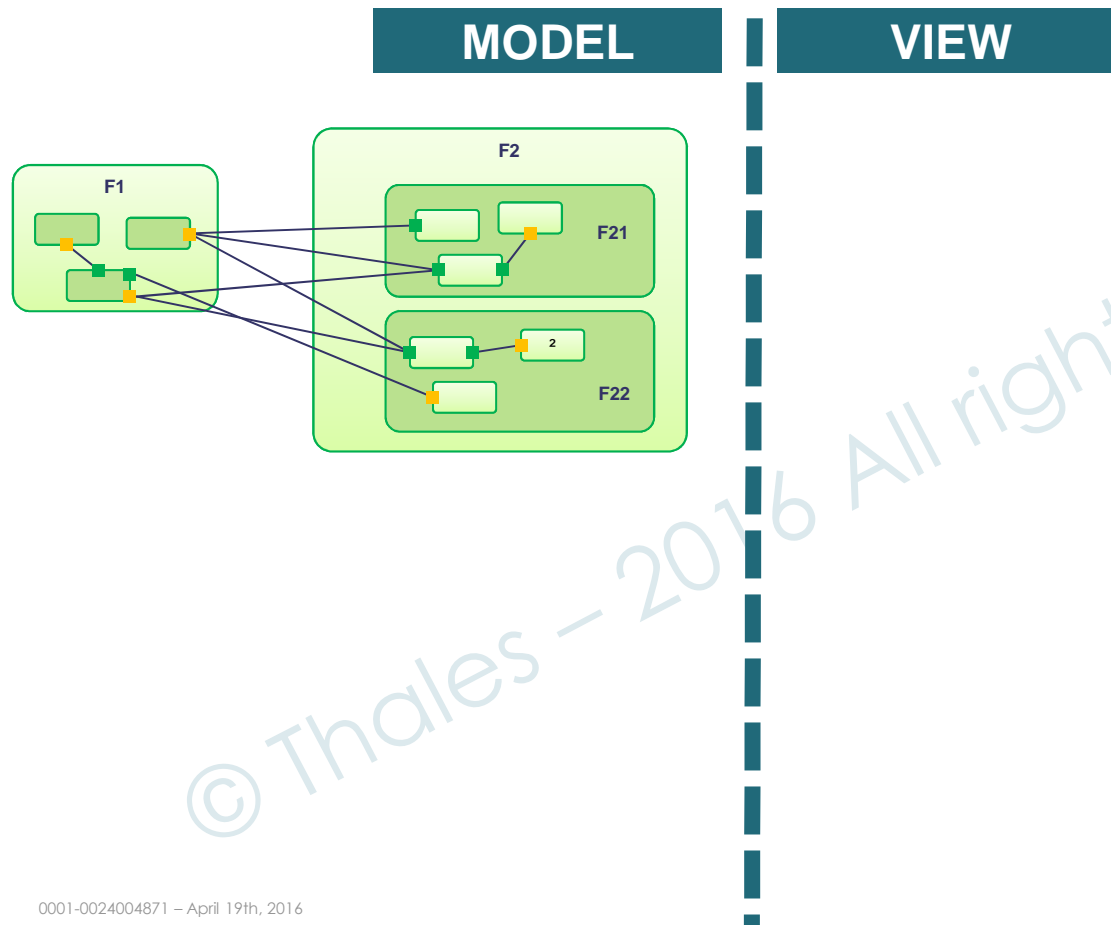
This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.

© Thales - 2014



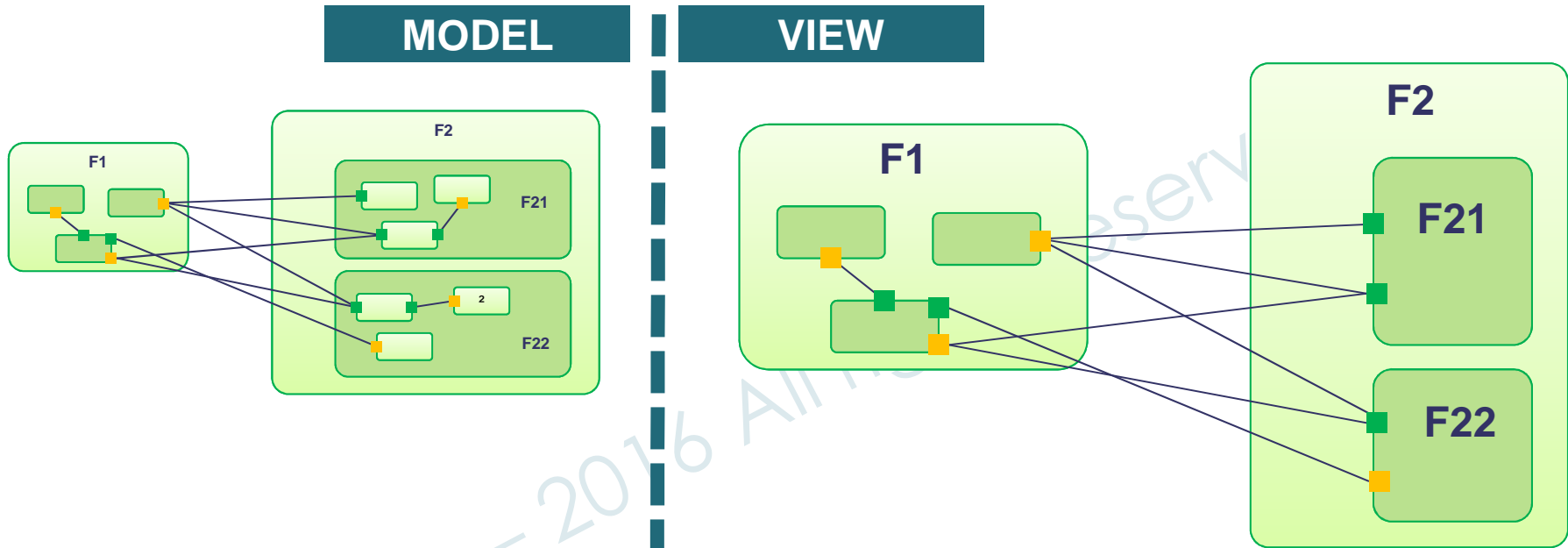
# Functional Analysis with Capella

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.



# Functional Analysis with Capella

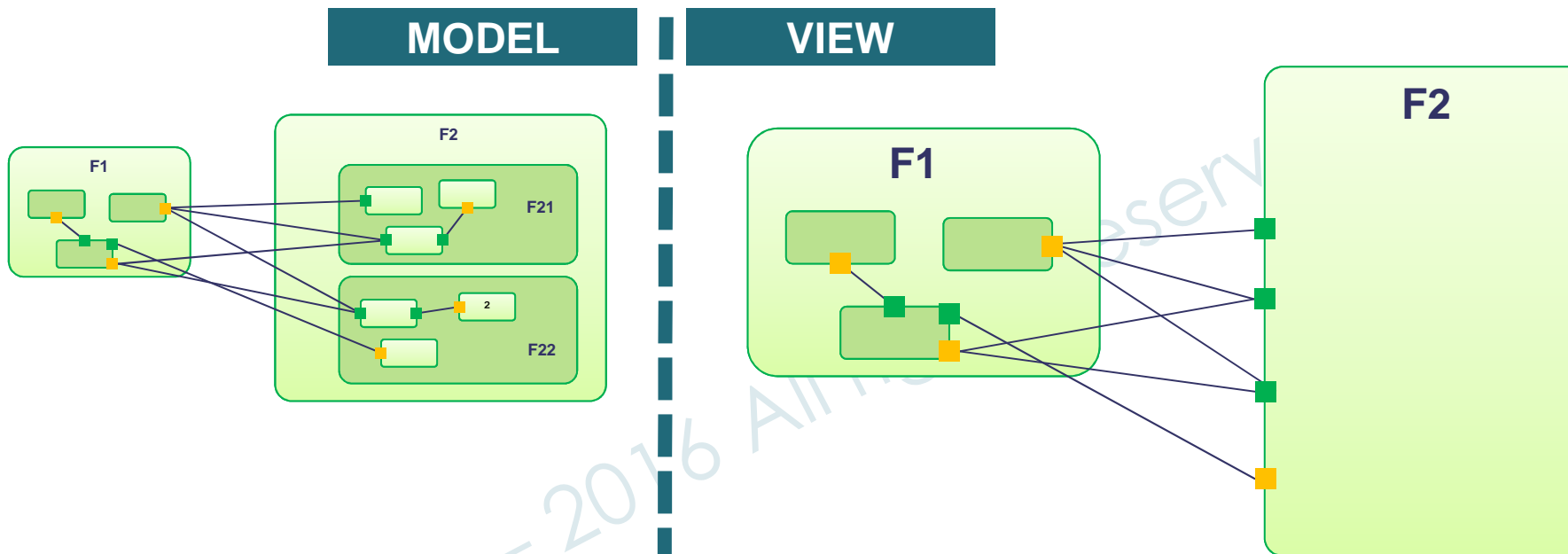
This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.



Graphical simplification: Ports on F21 and F22 do not actually « belong » to F21 and F22 but to their children functions.

# Functional Analysis with Capella

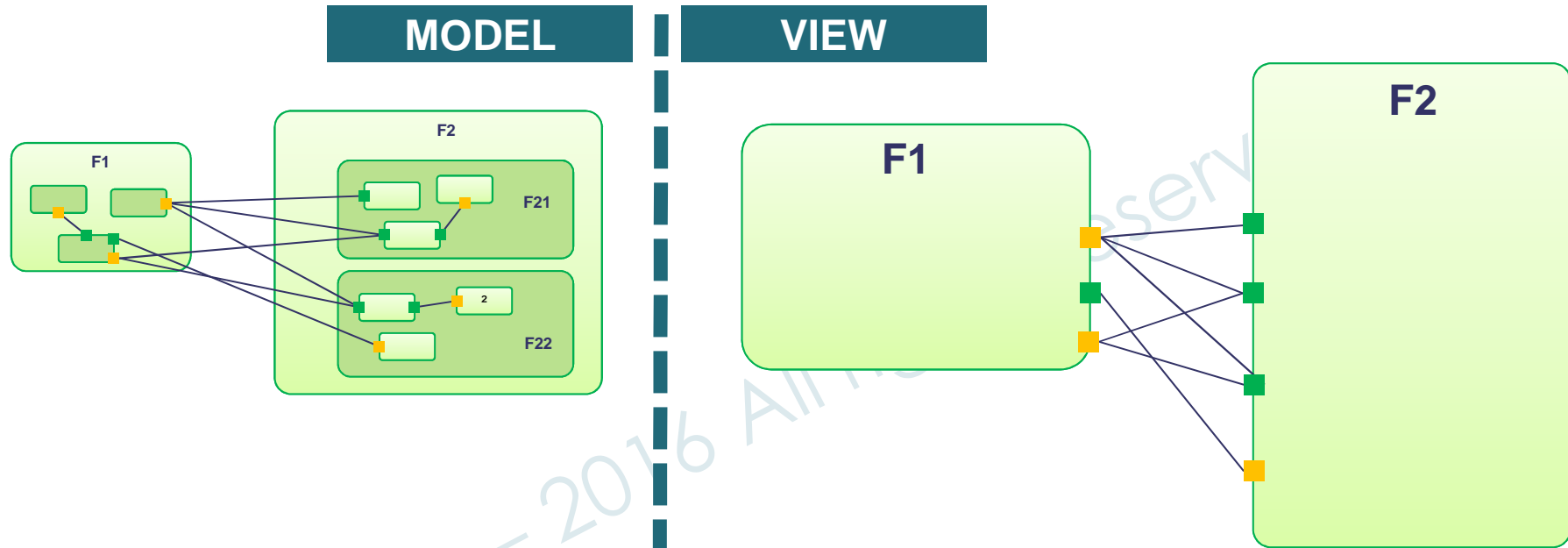
This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.



Graphical simplification: Ports on F2 do not actually « belong » to F2 but to its children functions.

# Functional Analysis with Capella

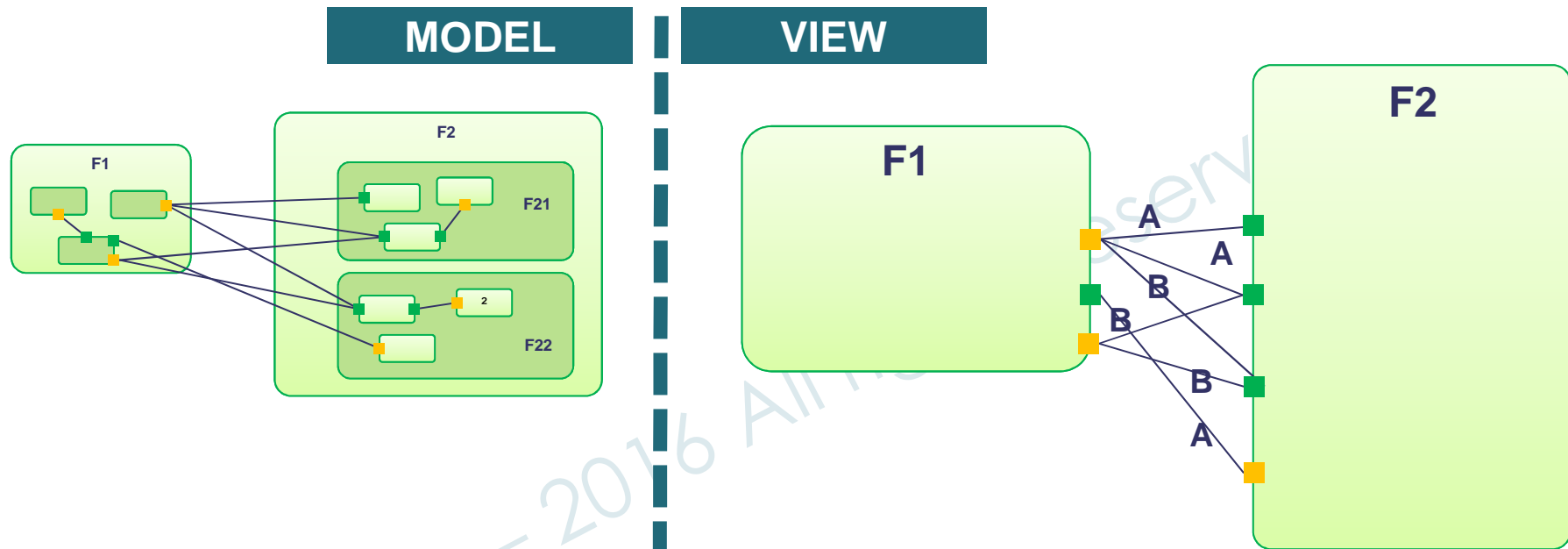
This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.



Graphical simplification: Ports on F1 and F2 do not actually « belong » to F1 and F2 but to their children functions.

# Functional Analysis with Capella

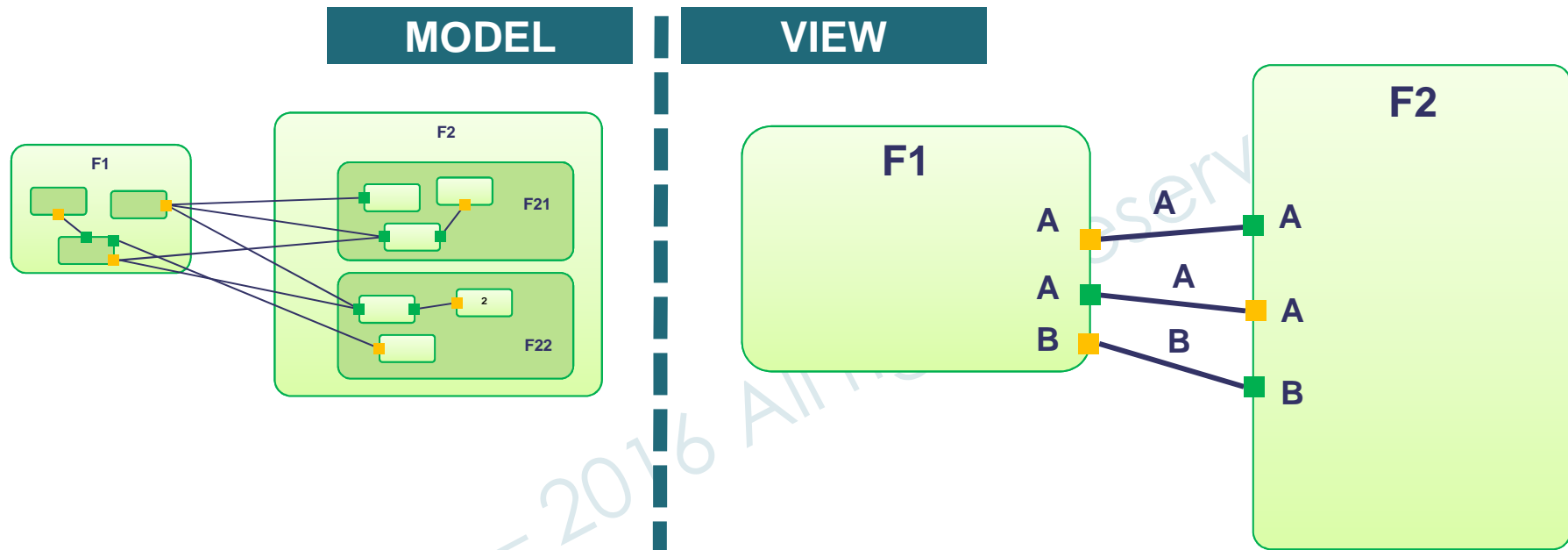
This document may not be reproduced, modified, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.



Introduction of the « Category » concept

# Functional Analysis with Capella

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.



Graphical simplification based on the « Category » concept

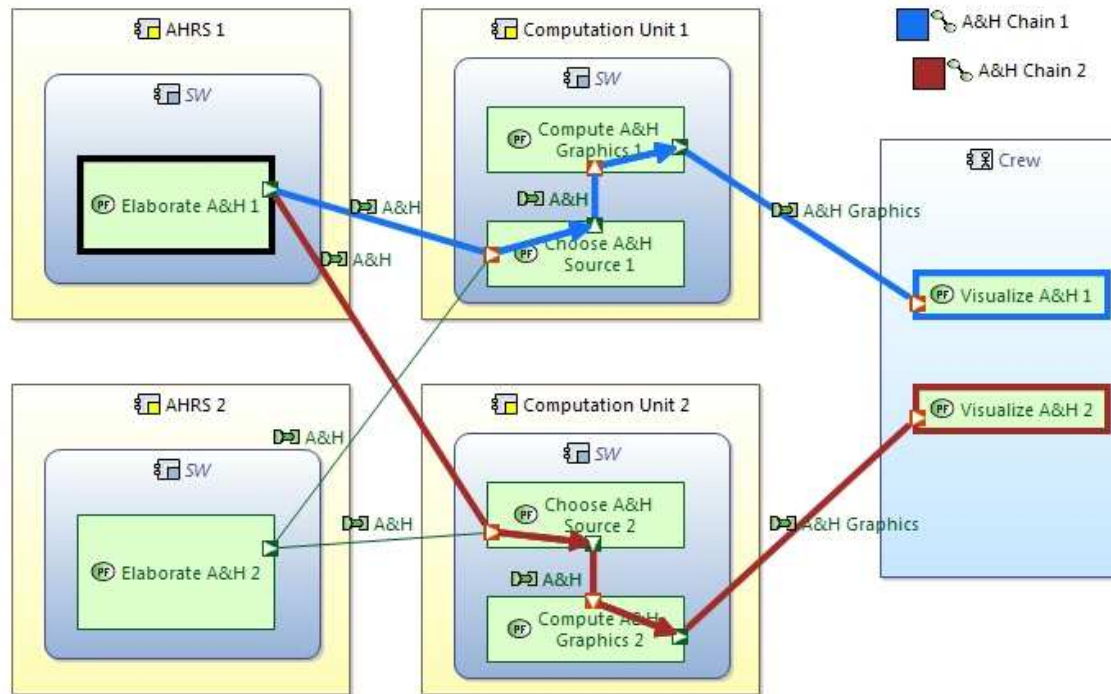


Language and tooling work together to address practitioner's engineering challenges and support different workflows

Computed graphical simplifications are key to manage complexity

# Instance-Driven Modeling

Most systems engineers think in term of instances, not types!





# Instance-Driven Modeling

## SysML

- **Blocks** have **Parts**, typed by other **Blocks**
- **Blocks** can have the “**PropertySpecificType**” stereotype, emulating an instance-level modeling
- **Activities** have **Partitions**
- **CallBehaviorActions** belong to **Partitions** and invoke **Activities**
- **Partitions** represent either by **Blocks** or **Parts**
- **Activities** have **ParameterNodes**
- **Actions** have **Pins**
- **Blocks** have **FlowPorts**
- **Blocks** are related to each other via **Associations**
- **Parts** do not have their own **FlowPort** “instances”
  
- No diagram showing simultaneously Component and Activity/Actions

## Arcadia-Capella

- **Functions** are allocated to **Components**
- By default, one **Component** == one **Part**
- **Functions** and **Components** have **Ports**
- Any set of element can be part of a **rREC** (record) or a **RPL** (replica)
- Content is synchronized between **RPL** and **RECs**

# Instance-Driven Modeling

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015 All rights reserved.

## SysML

- Blocks have Parts, typed by other Blocks
- Blocks can have the "PropertySpecificType" stereotype, emulating an instance-level modeling
- Activities have Partitions
- CallBehaviorActions belong to Partitions and

Extremely ~~complex~~ rich language

- Parts do not have their own HOWFOR "instances"
- No diagram showing both Component and Activity/Actions

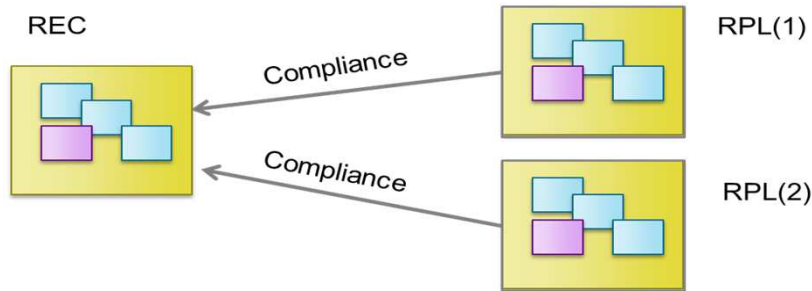
## Arcadia-Capella

- Functions are allocated to Components
- By default, one Component == one Part
- Functions and Components have Ports
- Any set of element can be part of a REC or a RPL
- Content is synchronized between RPL and RECs

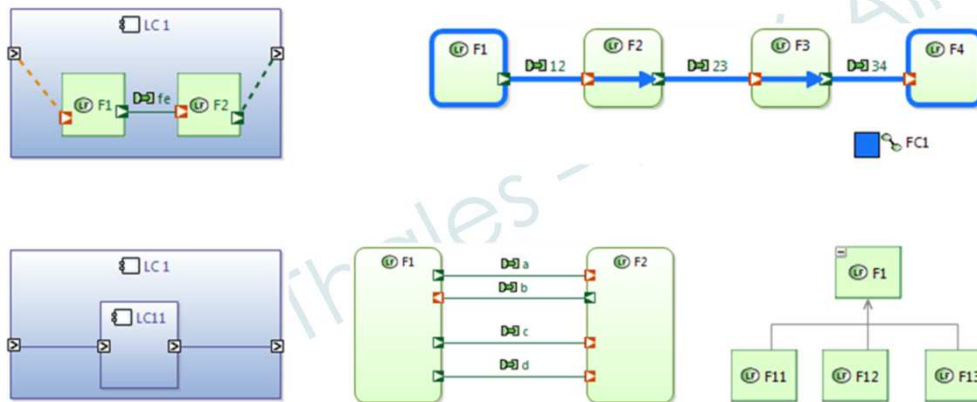
The same language concepts are used both for type and instance modeling

# Instance-Driven Modeling: The Capella Solution

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.



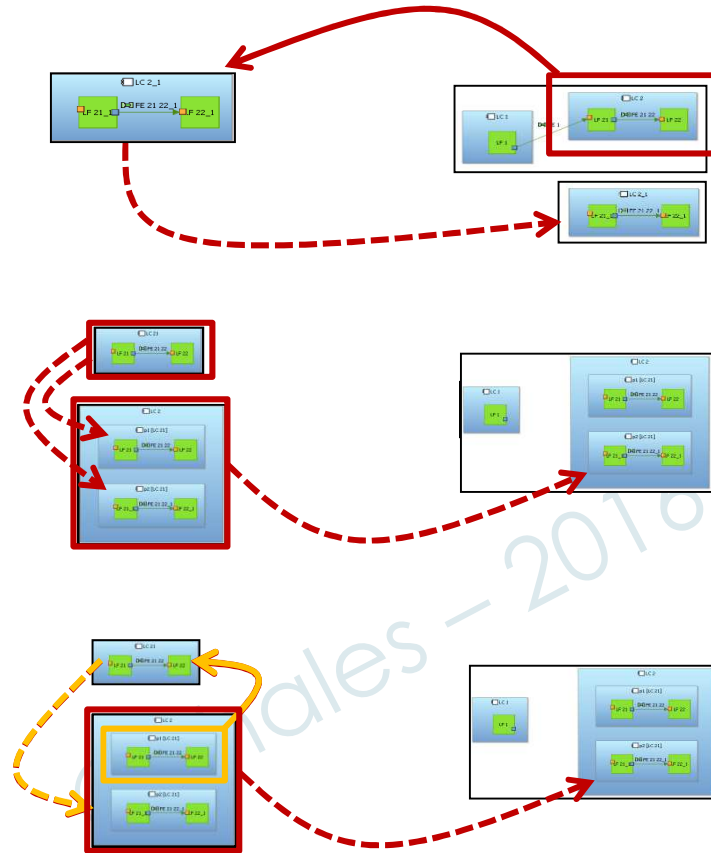
REC-RPL mechanism. The same language concepts are used for both types and instances



A « type » can be anything, including multi-root sets of elements

# Instance-Driven Modeling: The Capella Solution

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.



Support of multiple workflows

# THALES

## Development Perspective

MODEL EXTENSION AND EXPLOITATION

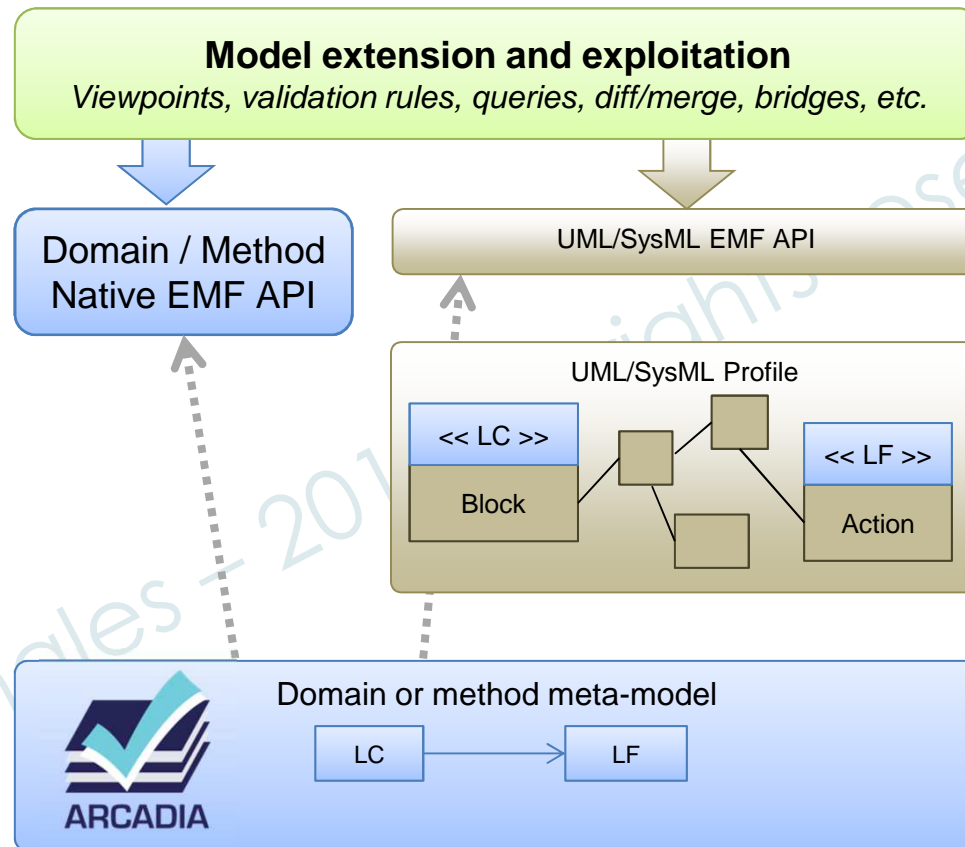
[www.thalesgroup.com](http://www.thalesgroup.com)



# Tooling and API Perspective

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2015. All rights reserved.

<< Give me all the Logical Components and their lists of Functions >>



<< Give me all the Blocks with << LC >> stereotype and their lists of Actions with << LF >> stereotypes >>



**Take  
Away**

April 19<sup>th</sup>, 2016

# THALES

## Visit us on Capella tabletop!

Capella website:

<http://www.polarsys.org/capella/>

LinkedIn 

<http://www.linkedin.com/company/capella-modelling-workbench>

Twitter 

[https://twitter.com/capella\\_arcadia](https://twitter.com/capella_arcadia)

Arcadia forum:

<https://polarsys.org/forums/index.php/f/12/>

Capella forum:

<https://polarsys.org/forums/index.php/f/13/>

IFE model & doc.:

<http://www.polarsys.org/capella/start.html>

[www.thalesgroup.com](http://www.thalesgroup.com)

