



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

# Introduction

# Kitalpha



OPEN  
Version 1.0.0

THALES



- 1 Introduction
- 2 Core Technology Kit
- 3 MBE Workbench
- 4 Studio Environment
- 5 Information

This document is not to be reproduced, modified, adapted, published, translated in any material form in whole or in part nor disclosed to any third party without the prior written permission of Thales. © THALES 2013 – All rights reserved.

# Introduction

**Market**



**DSMs and  
Model-Based  
Engineering  
Workbenches**

- Systems are more complex
- Evolving environments
- Do more... cheaper and faster

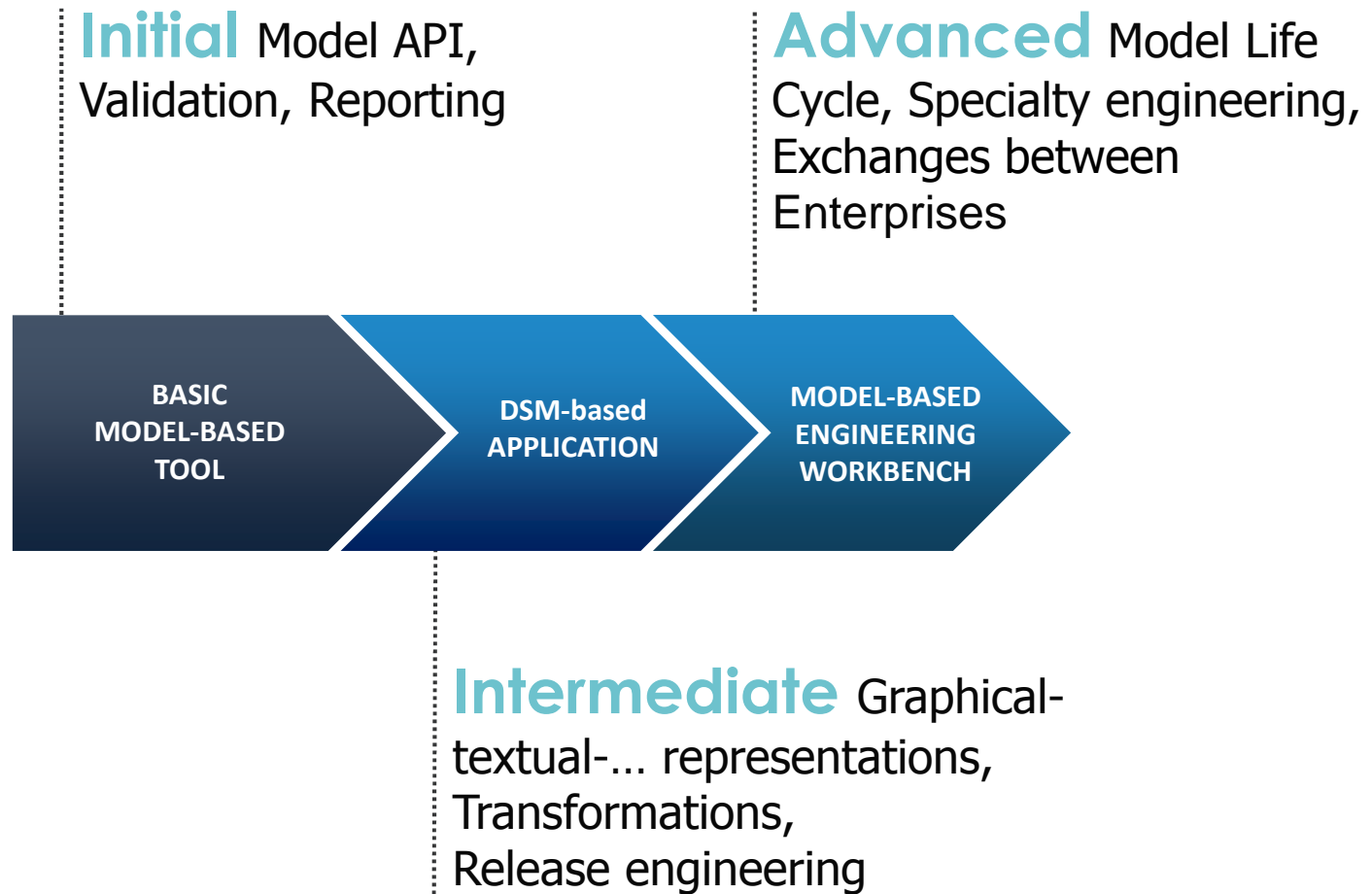
## ■ Objectives

- Providing an environment to develop and execute "classic" DSMs (Domain-Specific Modeling) and MBE workbenches and viewpoints for description of system architecture in system, software and hardware engineering
- Implementing the ISO/IEC-42010 standard for description of system architecture with viewpoints
- Providing Modeling Engineering Commonalities

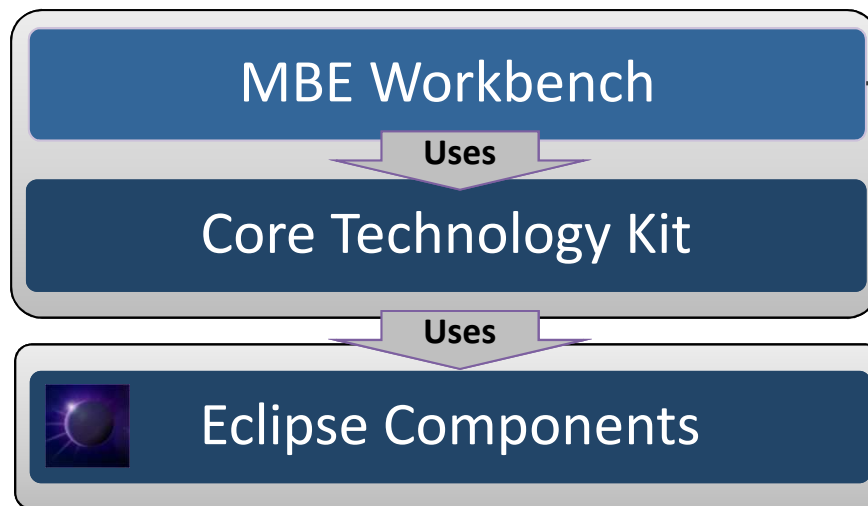
 **Kitalpha**

OPEN

**THALES**



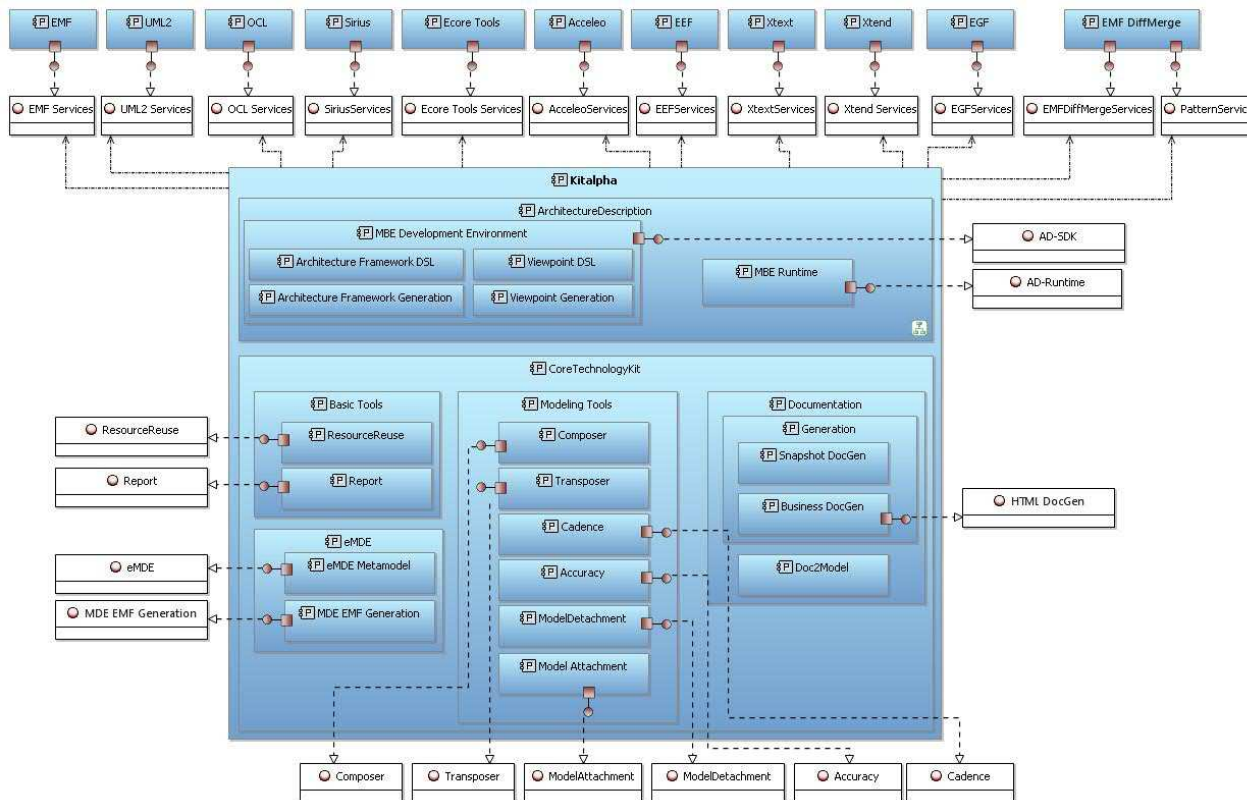
This document is not to be reproduced, modified, adapted, published, translated in any material form in whole or in part nor disclosed to any third party without the prior written permission of Thales. ©THALES 2013 – All rights reserved.



**Engineering Level** for development and execution of MBE Workbench and its viewpoints.

**Technology Level** with components for engineering needs not provided by Eclipse.

# Core Technology Kit (CTK)



<b>Graphical / Textual Representation</b>
Eclipse: Sirius, Ecore Tools, Xtext, Xtend
CTK: -
<b>Documentation</b>
Eclipse: [GenDoc2]
CTK: Business DocGen [, Doc2Model]
<b>Model Exchange</b>
Eclipse: -
CTK: Model Detachment, Model Attachment
<b>Model Comparison</b>
Eclipse: Emf Diff/Merge, Emf Compare
CTK: -
<b>Transformation</b>
Eclipse: Acceleo, EGF
CTK: Transposer, Composer
<b>Basic</b>
Eclipse: EMF, OCL, UML 2
CTK: Resource Reuse, Reports, eMDE, Accuracy

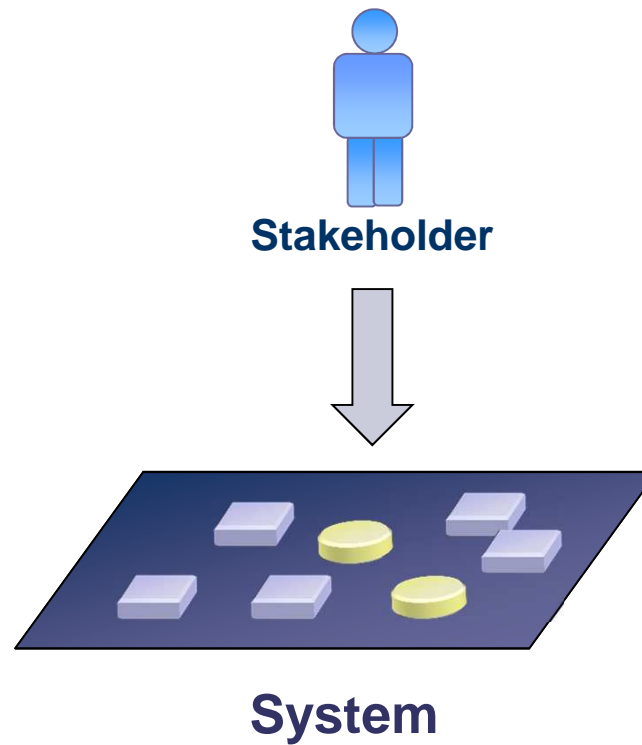


# MBE Workbench

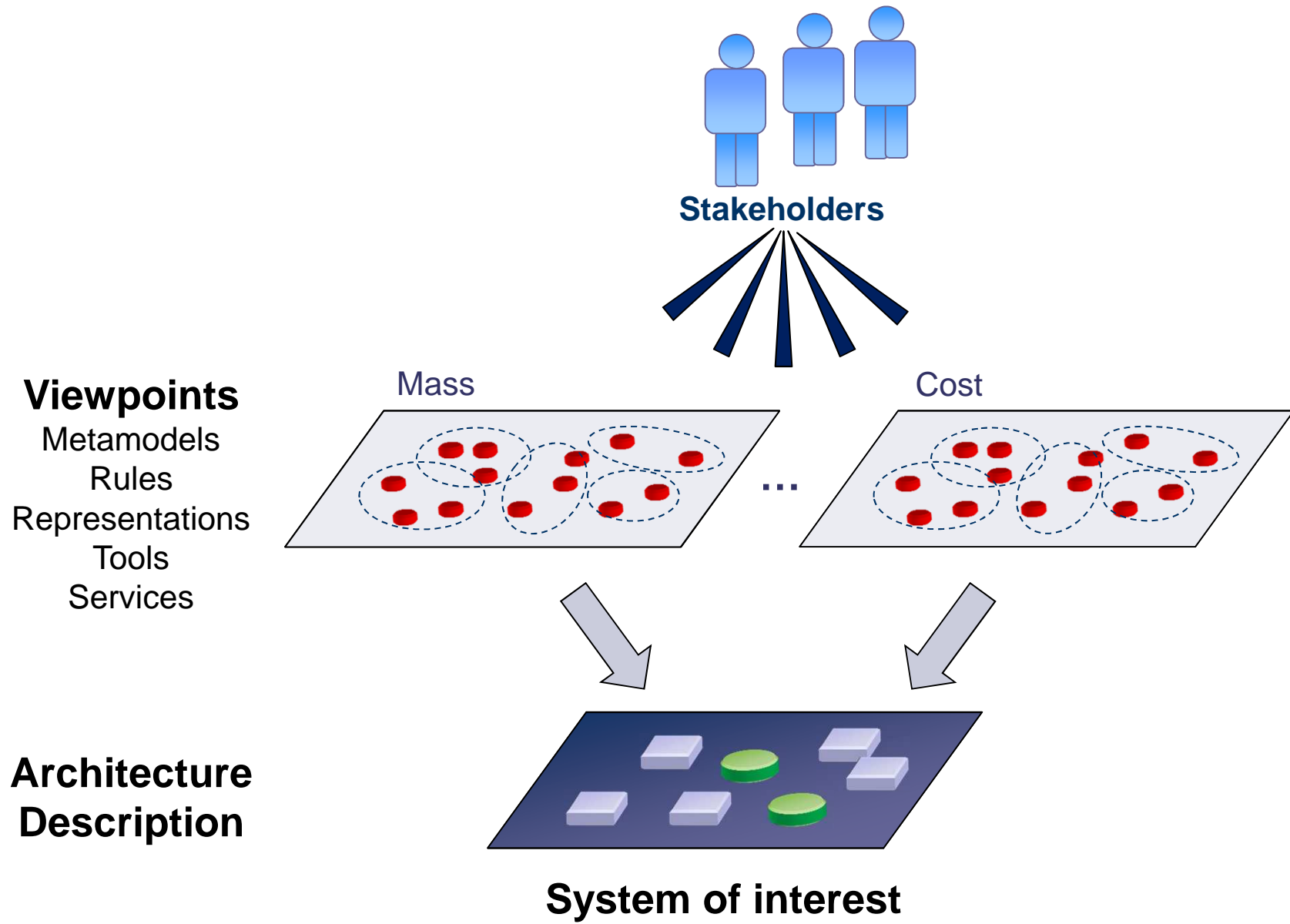
# MBE Workbench

## Architecture Framework & Viewpoint

**Architecture  
Description**



OPEN

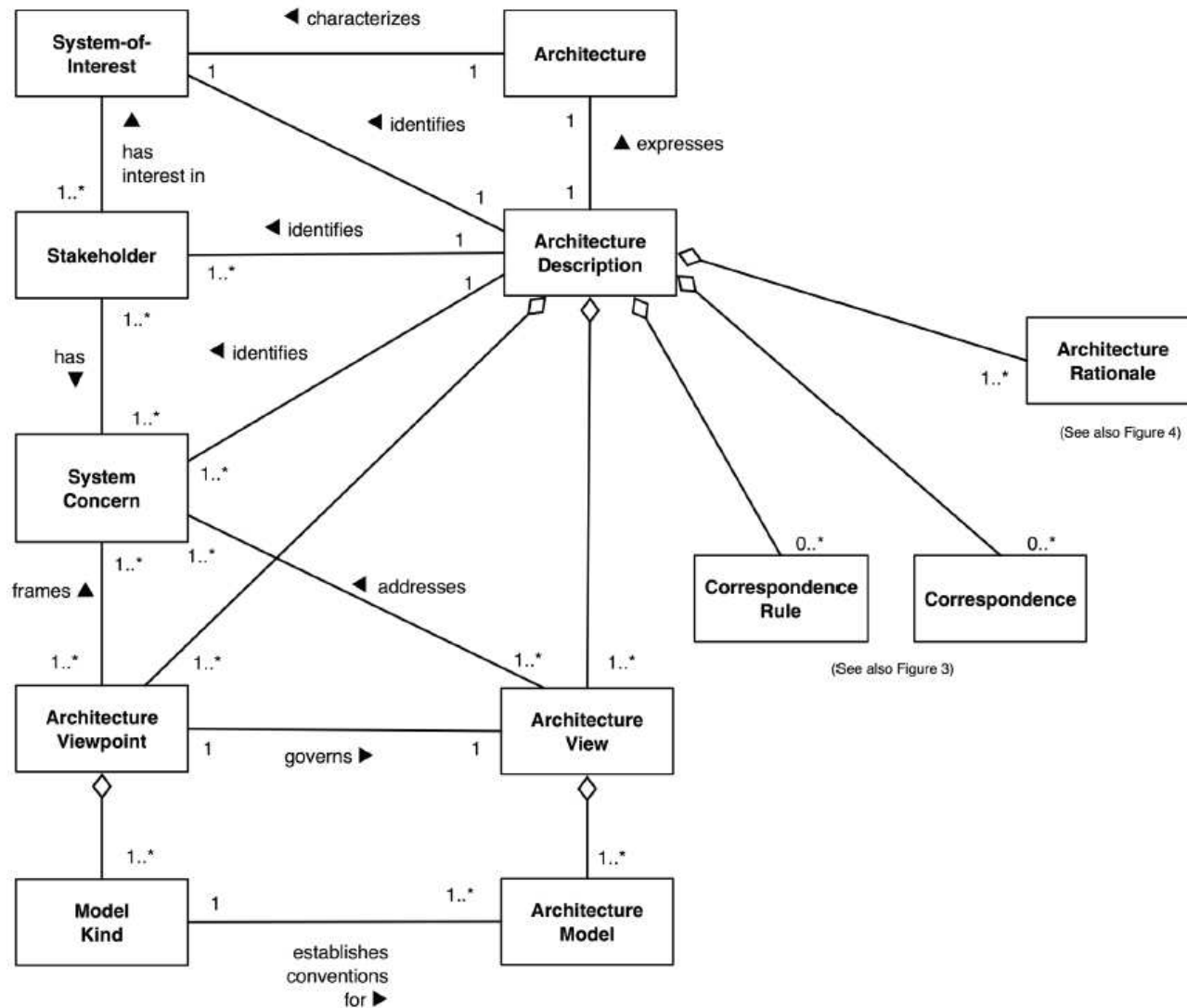


**Viewpoints**  
 Metamodels  
 Rules  
 Representations  
 Tools  
 Services

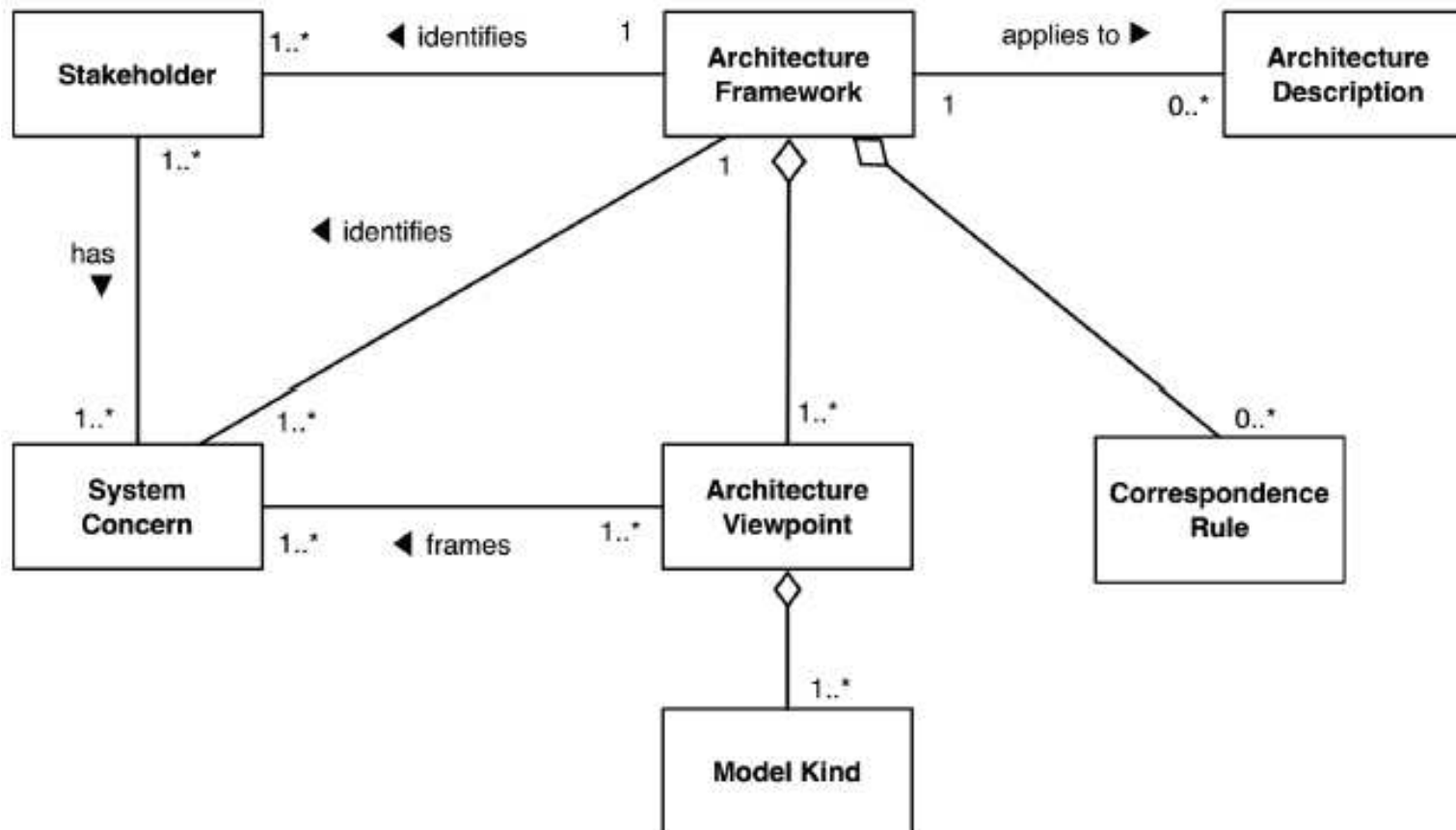
**Architecture Description**

**System of interest**

OPEN



Conceptual model of architectural description (ISO/IEC FCD 42010 – 2010-06-08)



*Architectural framework and application to architecture descriptions (ISO/IEC FCD 42010 – 2010-06-08)*



## ISO/IEC WD3 42010 (2010-06-08)

“An **architecture framework** establishes a common practice for creating, interpreting, analyzing and using **architecture descriptions** within a particular domain of application or stakeholder community.”



## ISO/IEC WD3 42010 (2010-06-08)

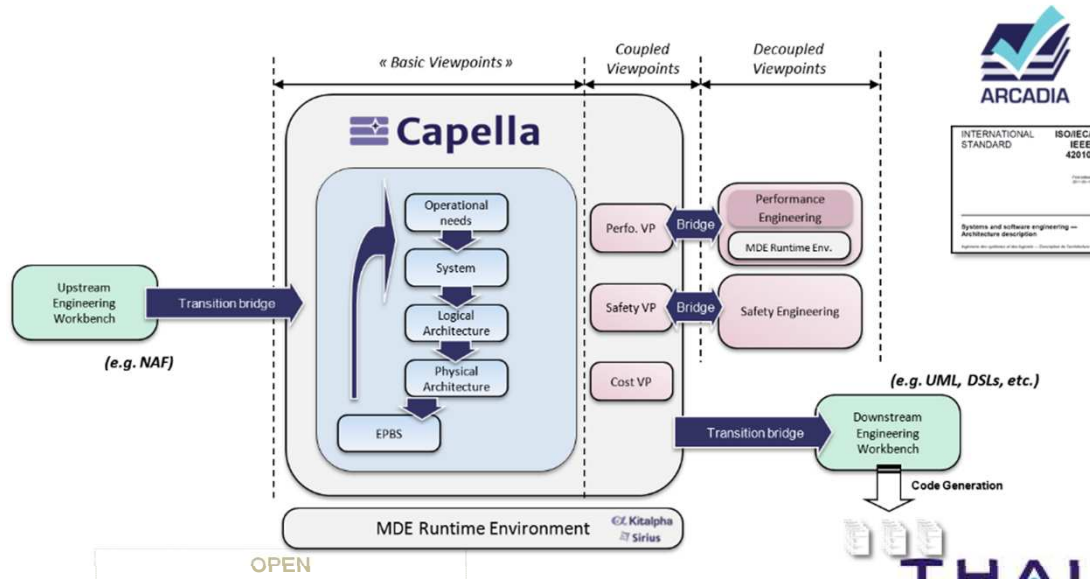
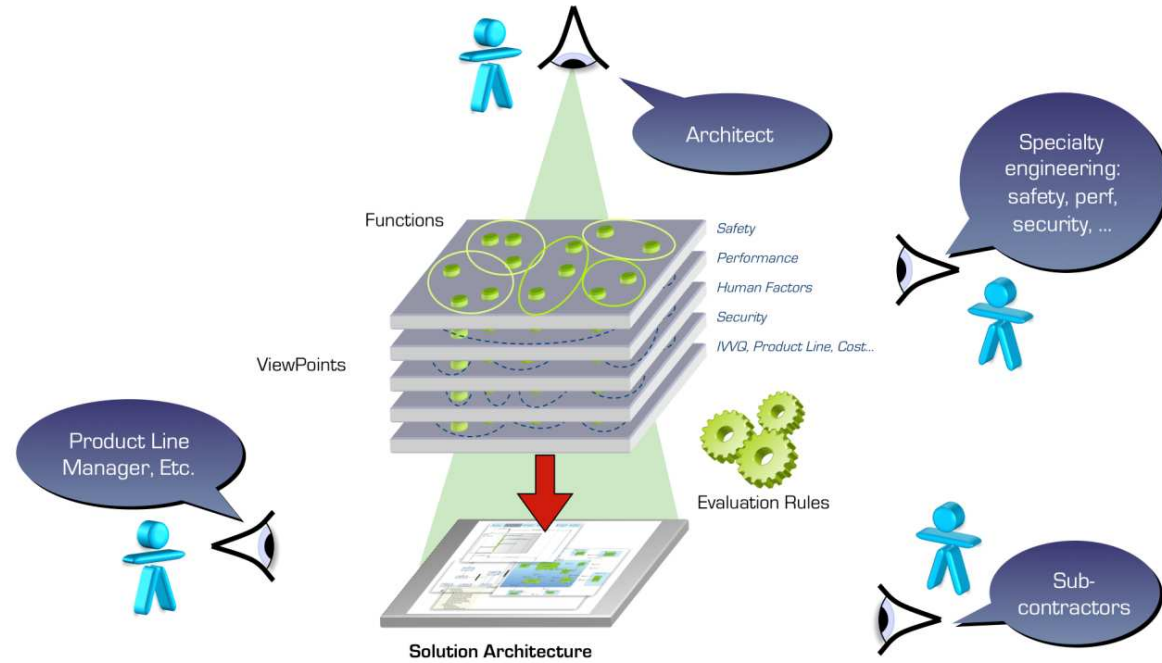
“An **architecture description** includes one or more architecture views. Each architecture view (or simply, view) addresses one or more of the system concerns held by the system’s stakeholders.

Each **architecture view** expresses the architecture of the system-of-interest in accordance with an architecture viewpoint (or simply, viewpoint). Each viewpoint frames one or more system concerns. Each concern can be framed by one or more viewpoints.

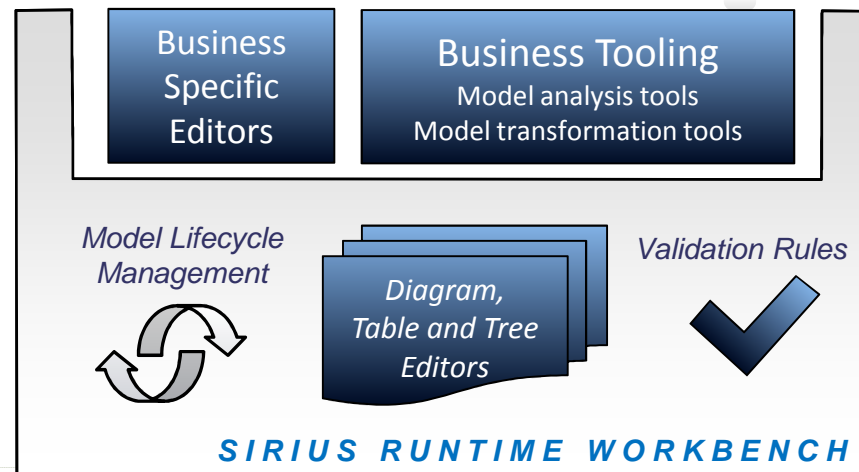
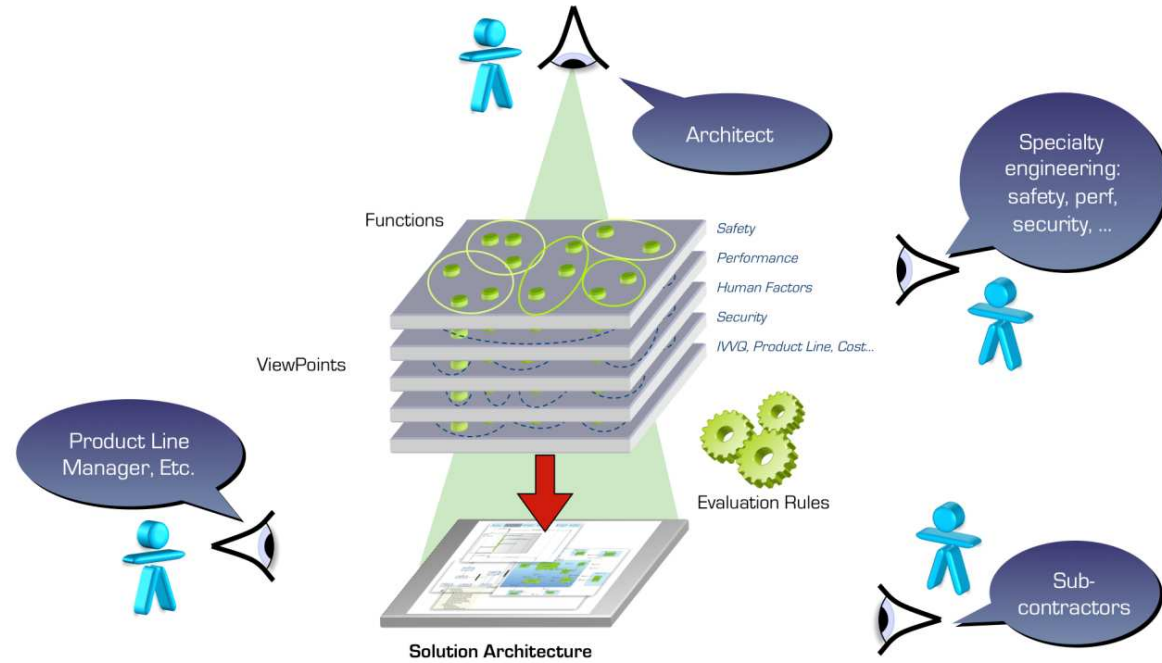
Each view is governed by its viewpoint: the **viewpoint** establishes the conventions for constructing, interpreting and analyzing the view to address concerns framed by that viewpoint. Viewpoint conventions can include languages, notations, model kinds, design rules, and/or modelling methods, analysis techniques and other operations on views.”



# Example: Multi-Viewpoint with Arcadia Method and Capella



# Example: Multi-Viewpoint with Arcadia Method and Capella



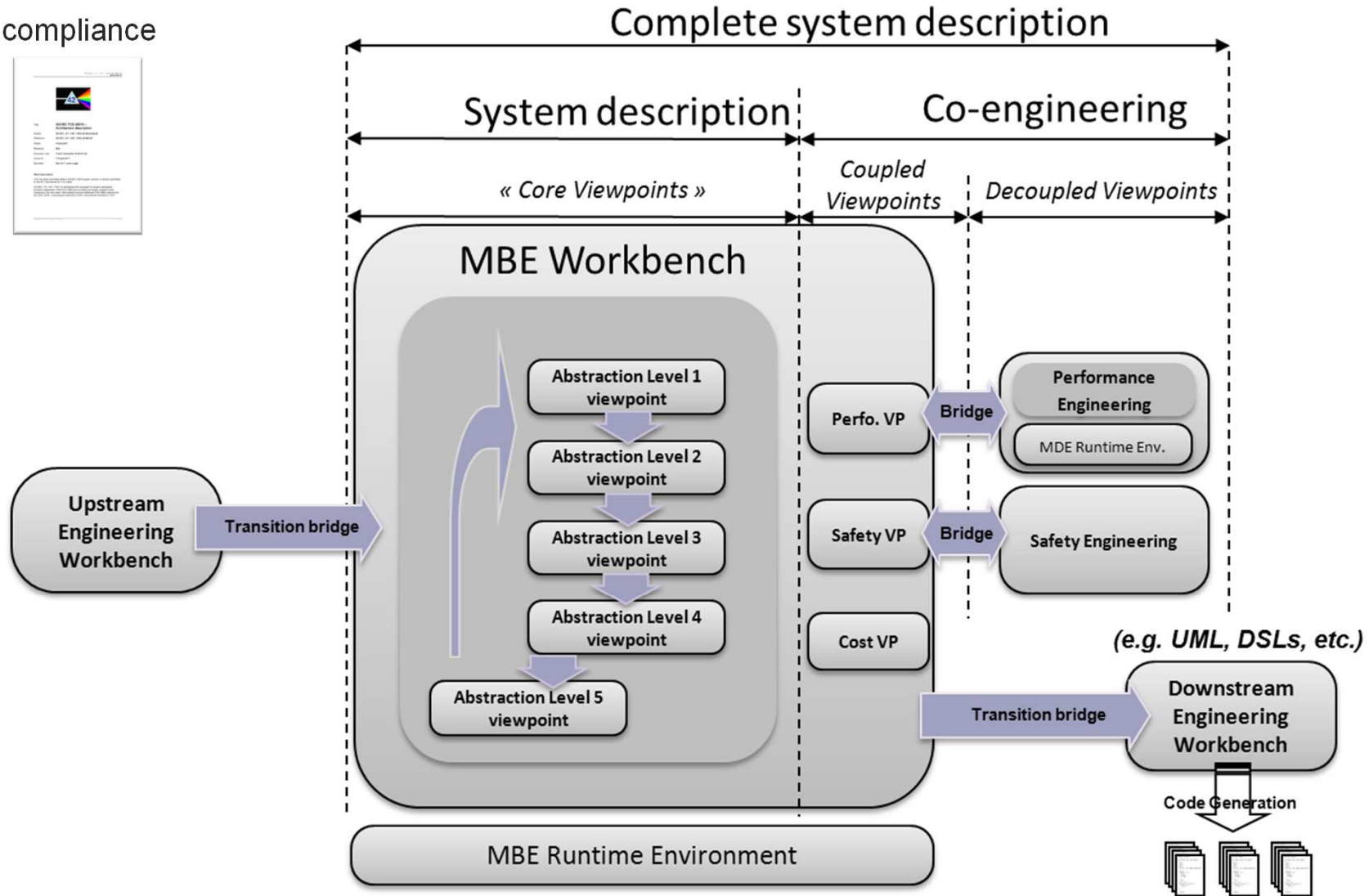
OPEN



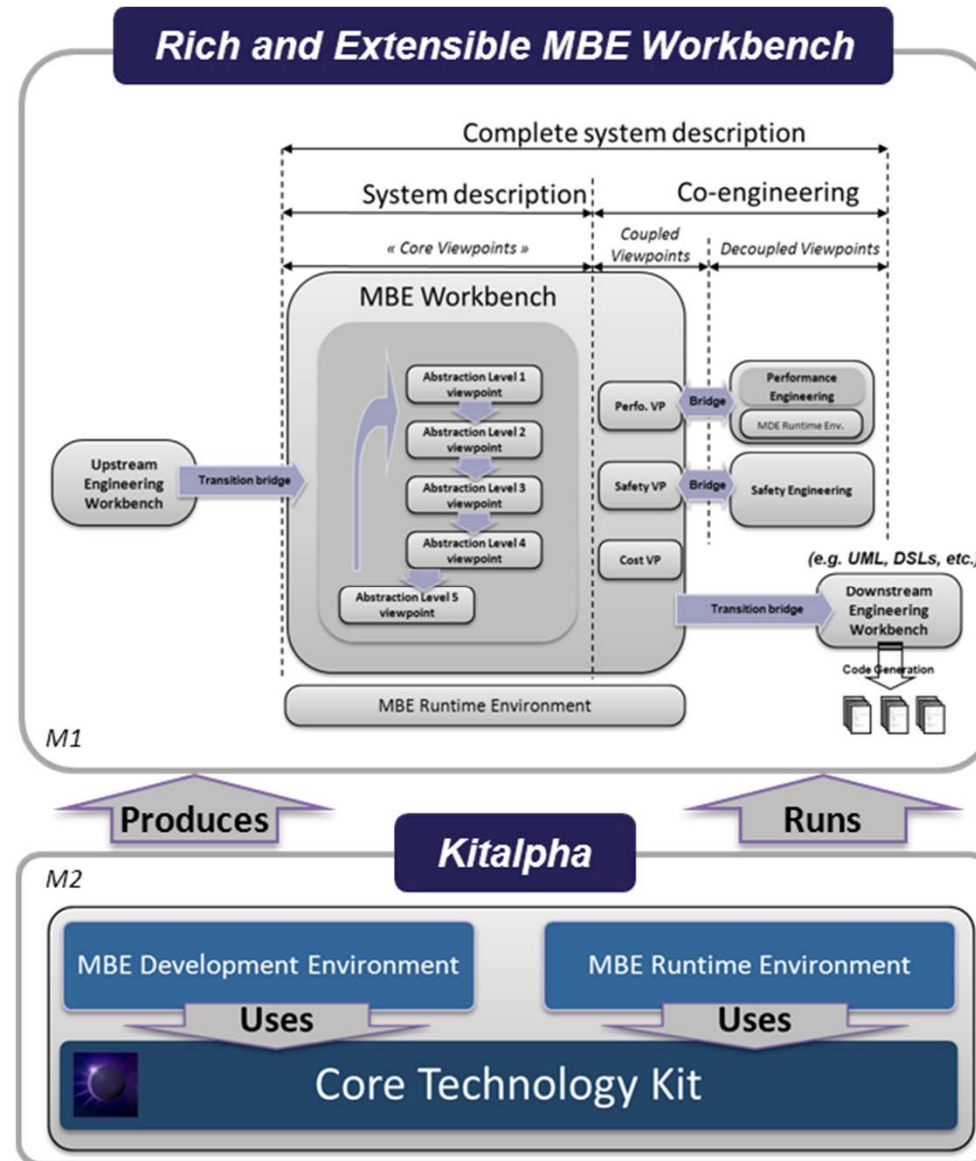
# Kitalpha

**Development and runtime environments for  
viewpoint-based modeling workbenches**

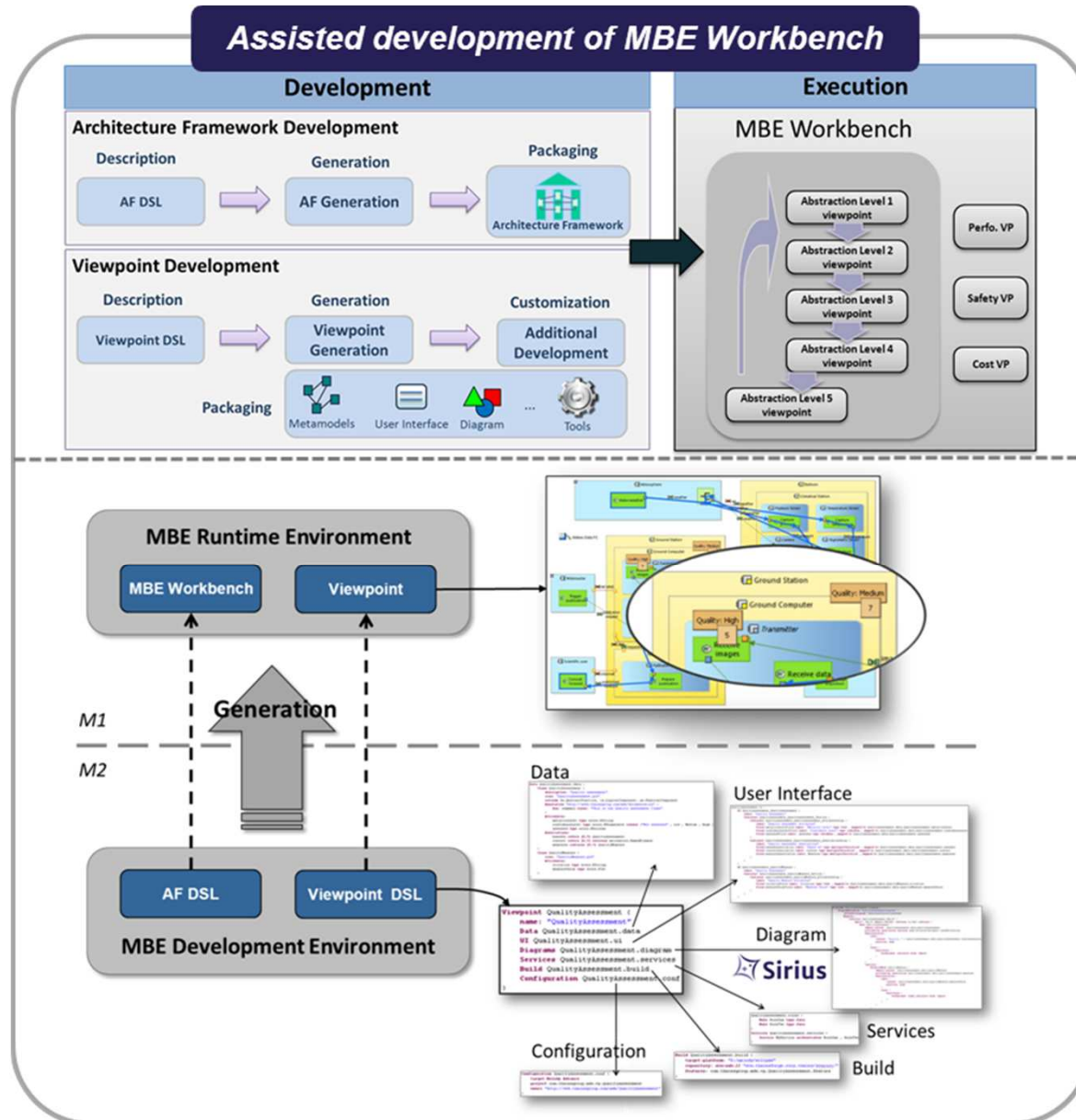
## ISO/IEC 42010 Std compliance



This document is not to be reproduced, modified, adapted, published, translated in any material form in whole or in part nor disclosed to any third party without the prior written permission of Thales. © THALES 2013 – All rights reserved.



This document is not to be reproduced, modified, adapted, published, translated in any material form in whole or in part nor disclosed to any third party without the prior written permission of Thales. © THALES 2013 – All rights reserved.



This document is not to be reproduced, modified, adapted, published, translated in any material form in whole or in part nor disclosed to any third party without the prior written permission of Thales© THALES 2013 – All rights reserved.



Development

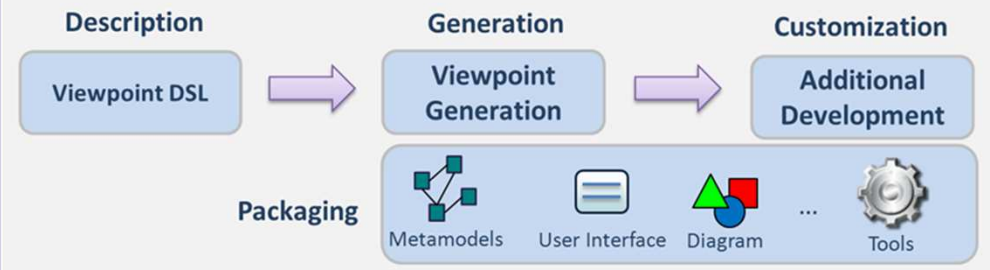
Execution

Process

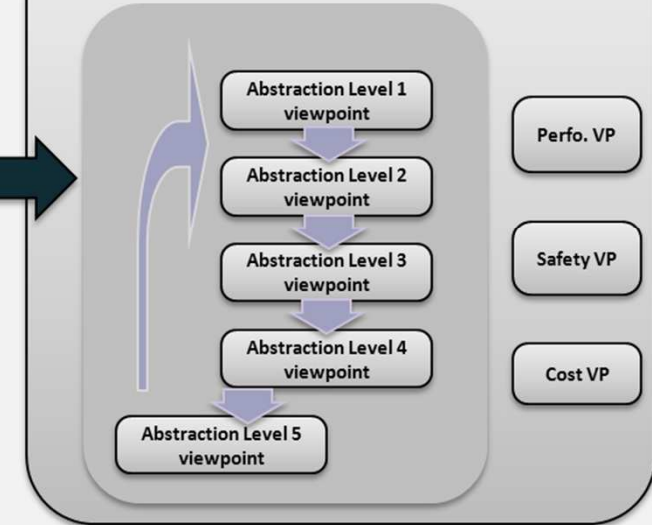
Architecture Framework Development



Viewpoint Development



MBE Workbench



Services

- Creation of Architecture Framework
- Edition of Architecture Framework
- Generation of Architecture Framework
- Packaging of Architecture Framework
- Creation of Viewpoint
- Edition of Viewpoint
- Generation of Viewpoint
- Viewpoint documentation generation
- Packaging of Viewpoint
- Reverse of ecore model to viewpoint dsl
- Versioning
- Collaborative work with viewpoints

Scope of Kitalpha

- System description by viewpoints
- Viewpoint activation
- Viewpoint deactivation
- Viewpoint detachment
- Viewpoint attachment
- Viewpoint migration
- Versioning
- Collaborative work with viewpoints
- Reporting
- Architecture Assessment
- Test, Simulation

This document is not to be reproduced, modified, adapted, published, translated in any material form in whole or in part nor disclosed to any third party without the prior written permission of Thales© THALES 2013 – All rights reserved.

## Architecture Framework Structure

A set of viewpoints



## Viewpoint Structure

A set of metamodels

A set of rules (check, transformation...)

A set of notations

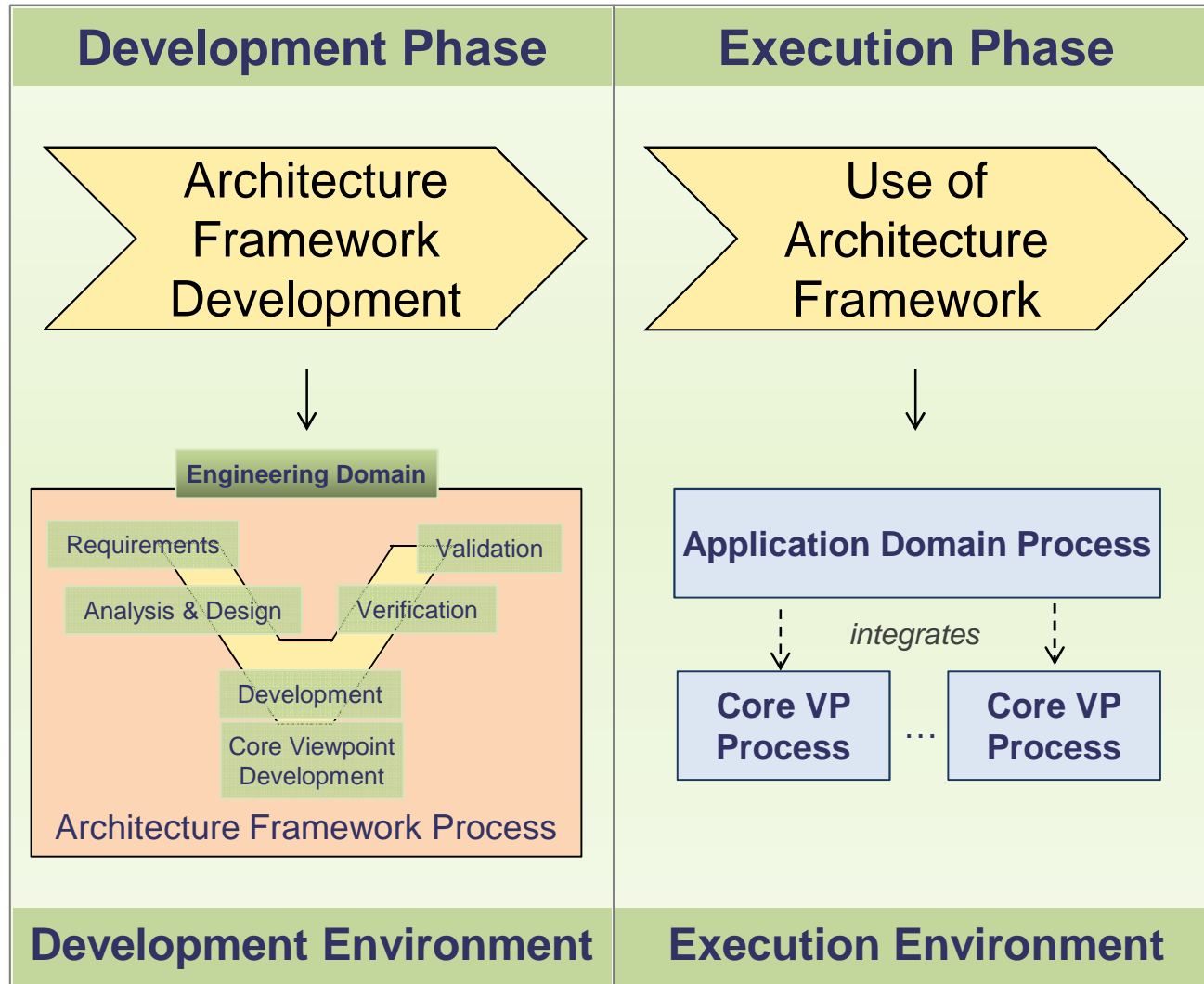
A set of representations (textual, graphical...)

A set of tools

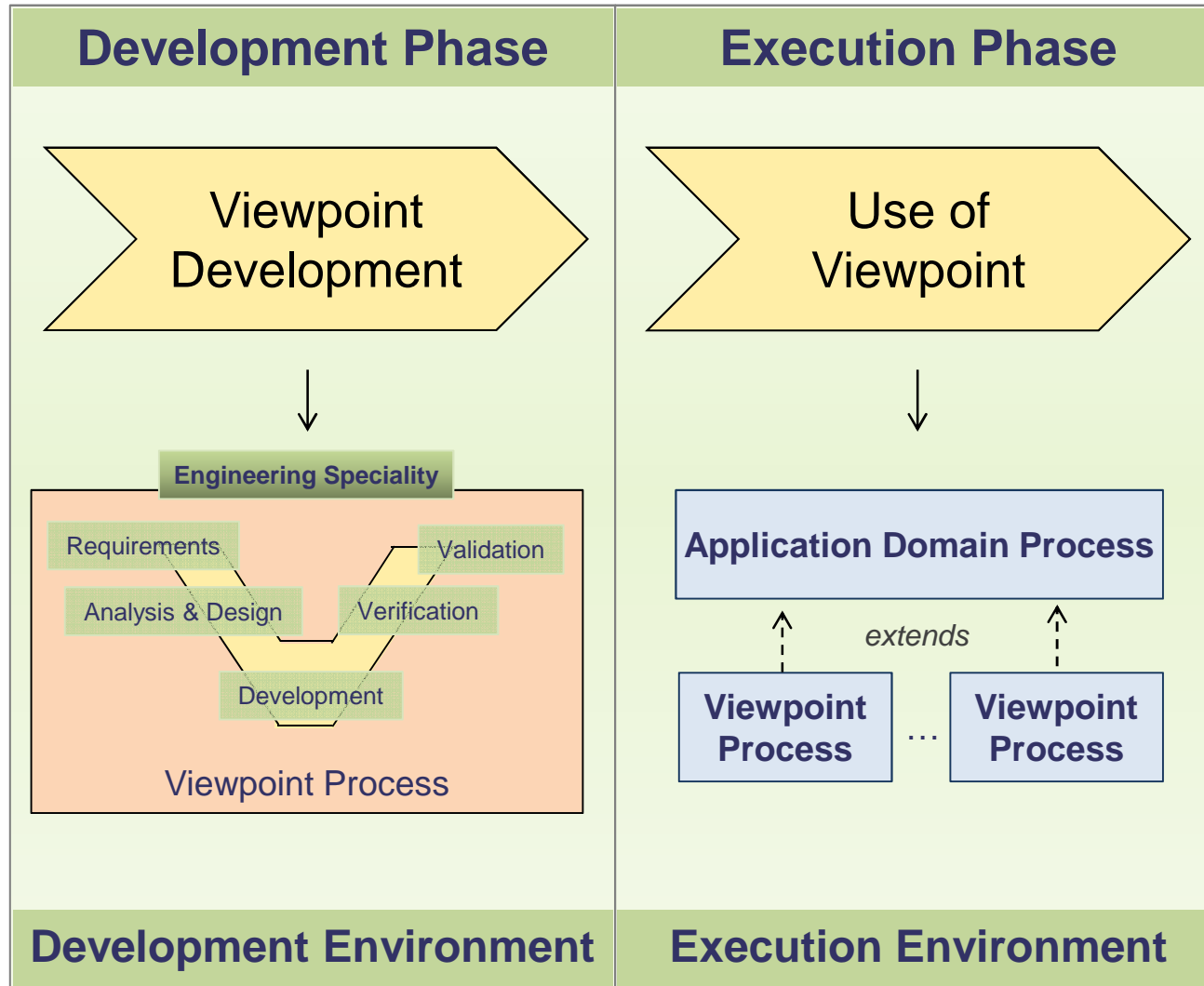
A set of services

# Process

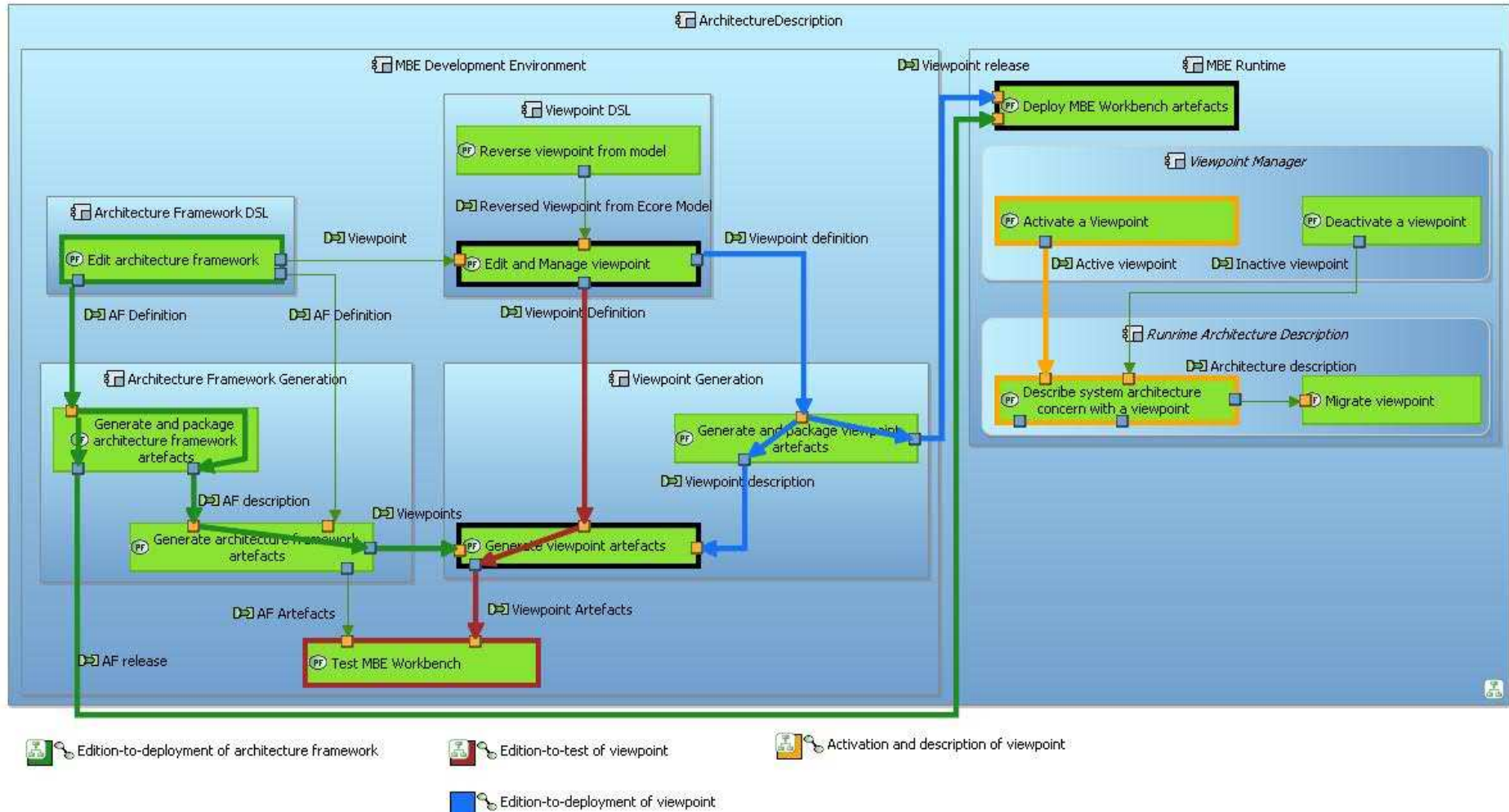
Process adaptation to viewpoint-based modeling development



This document is not to be reproduced, modified, adapted, published, translated in any material form in whole or in part nor disclosed to any third party without the prior written permission of Thales. © THALES 2013 – All rights reserved.



This document is not to be reproduced, modified, adapted, published, translated in any material form in whole or in part nor disclosed to any third party without the prior written permission of Thales. © THALES 2013 – All rights reserved.



This without the prior written permission of Thales © THALES 2013 – All rights reserved.

## Development of Architecture Framework

### Development phase

1. The Developer creates an architecture framework
2. The Developer develops a set of core viewpoints
3. The Developer aggregates the core viewpoints
4. The Developer generates all the architecture and core viewpoint artifacts
5. The Developer verifies the architecture framework
6. The Developer packages the architecture framework

### Use phase

1. The User deploys the architecture framework
2. The User uses the architecture framework

### Extensibility

1. The Developer develops a new viewpoint
2. The User deploys the viewpoint
3. The User uses the architecture framework with the new viewpoint

## Development of Viewpoint

### Development phase

1. The Developer creates a viewpoint
2. The Developer develops the different aspects of the viewpoint (e.g., data, diagram...)
3. The Developer generates the viewpoint artifacts
4. The Developer verifies the viewpoint
5. The Developer packages the architecture framework

### Use phase

1. The User deploys the viewpoint
2. The User uses the viewpoint in an architecture framework

# Studio Environment



**Objective:** availability of a development environment to extend a solution, especially for viewpoint development.



**Typical integration work:**

1. Integration of the [Solution] in Kitalpha
2. Definition of a new Target Application for the [Solution]
3. Integration of [Solution]-specific generators
4. Extensions of the textual editors
5. Customization of the html documentation generation



1. Integration of Capella in Kitalpha
2. Definition of a Capella Target Application
3. Integration of Capella-specific generators
4. Extensions of the textual editors
5. Customization of the html documentation generation

# Information

## Bugzilla

PolarSys Bugzilla: <https://polarsys.org/bugs/>

## Access to PolarSys Git repository

PolarSys: <http://polarsys.org/c/>

Kitalpha: <http://polarsys.org/c/kitalpha/kitalpha.git/>

## Access to PolarSys Gerrit

PolarSys: <https://polarsys.org/r>

## Kitalpha jobs

All jobs: <https://hudson.polarsys.org/kitalpha/>

Nightly build: [hudson.polarsys.org/kitalpha/job/Kitalpha\\_Nightly/](https://hudson.polarsys.org/kitalpha/job/Kitalpha_Nightly/)

## Kitalpha mailing list

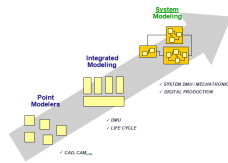
Name: kitalpha-dev

## Kitalpha Forum

<https://polarsys.org/forums/index.php/f/8/>

## Kitalpha Wiki

<https://polarsys.org/wiki/kitalpha>



Kitalpha is supported by **Sys2Soft**  
and **Crystal**, respectively French  
and European projects



# Th $\alpha$ nk You!

<https://www.polarsys.org/projects/polarsys.kitalpha>

[benoit.langlois@thalesgroup.com](mailto:benoit.langlois@thalesgroup.com)

[#LangloisBenoit](#)