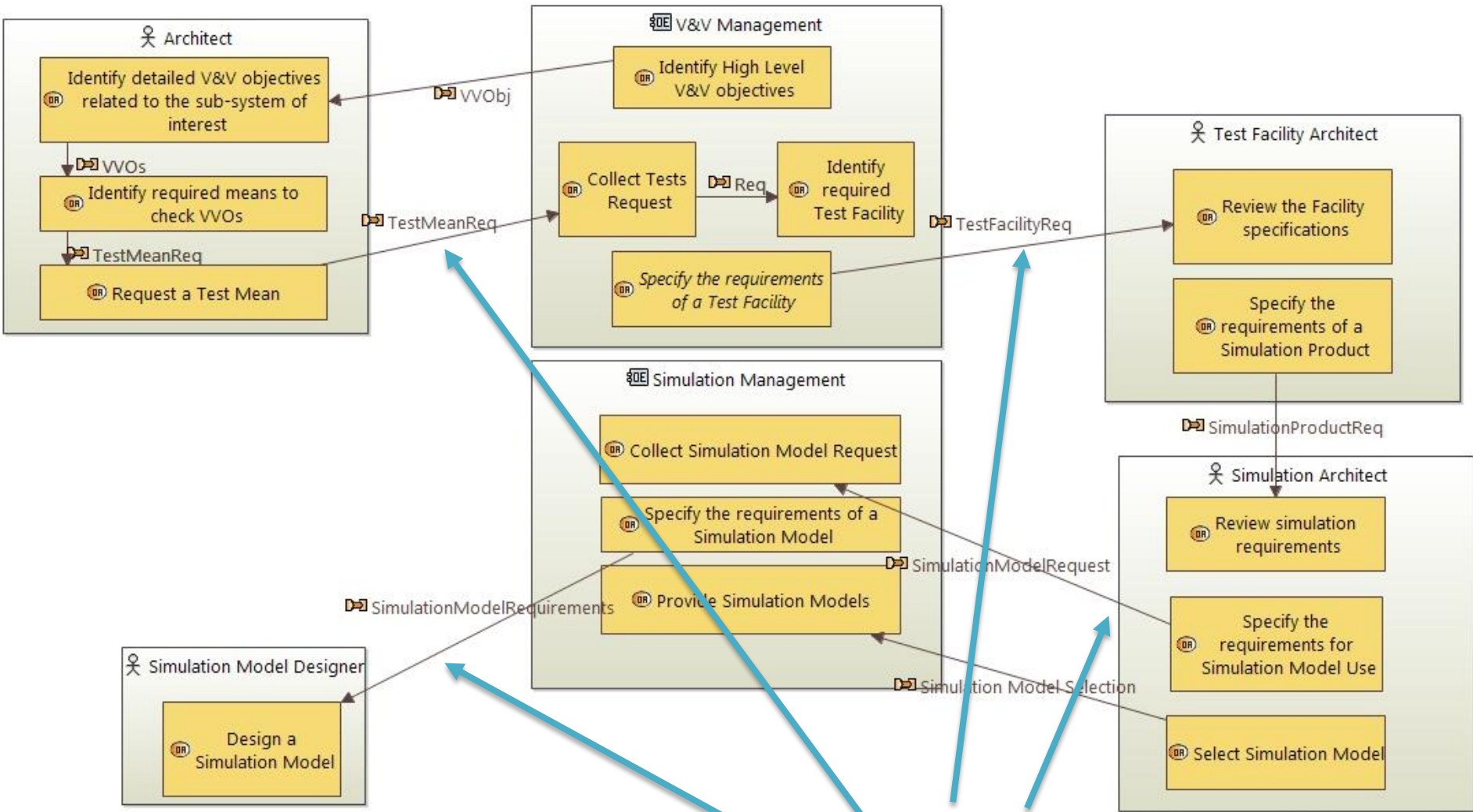


# Specifying and designing a test mean alla Capella



**Document-based process** ... with typical pains

## Specify the Test Means with a strong link to

- the VV Objectives & VV strategy
- the architecture of the system

## Expected benefits

- **Agility:** allow to adapt the VV strategy to
  - feedbacks from planning of deliveries & risk analysis
  - design changes
- **Lead time:**
  - start the development of the test means earlier (Logical)
  - Build the right means : no more / no less
- **Productivity:** work directly on the source information
- **Quality:** less manual error prone activities and interpretations

## Find the limit and balance between

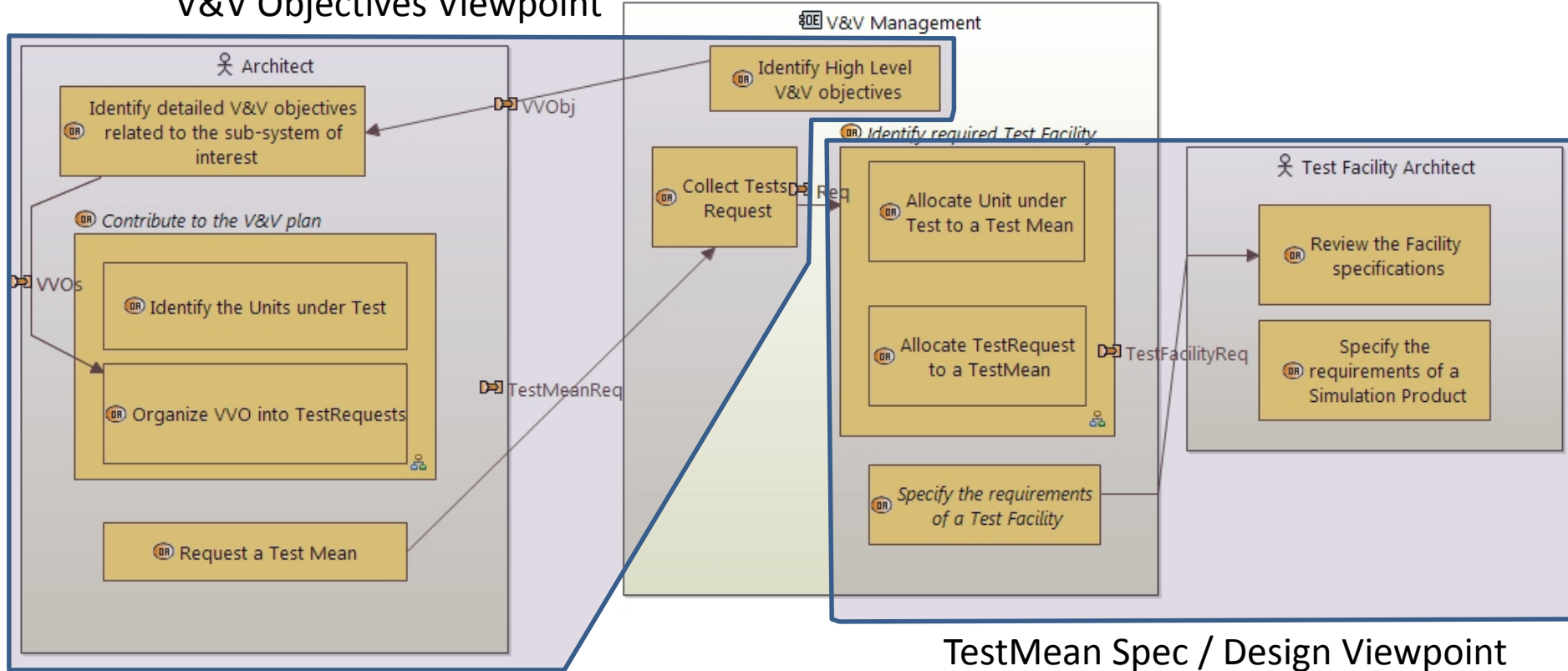
- what shall be implemented in Capella
- what shall be implemented in a dedicated tool

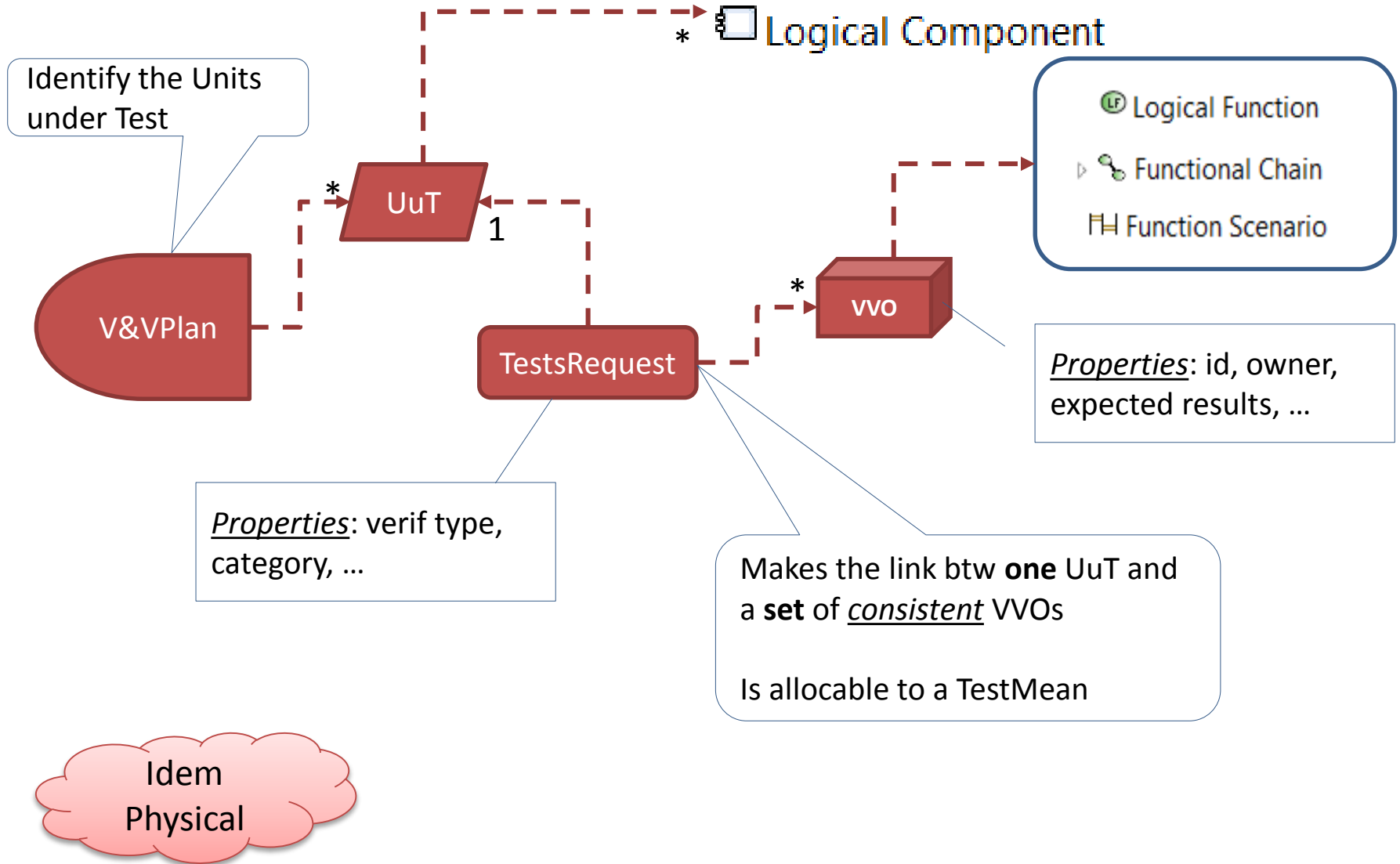


## Guidance:

- Use viewpoints when the additional data shall be consolidated with architecture data in a short-loop
- Use export when the additional data are loosely coupled

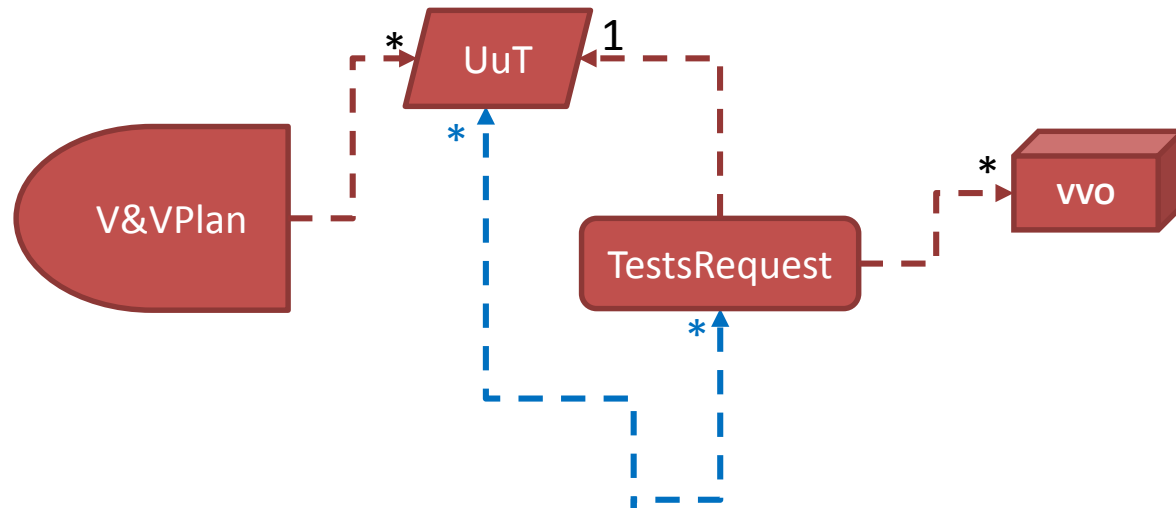
## V&V Objectives Viewpoint







*Demo*





 Logical Component  
 Logical Function



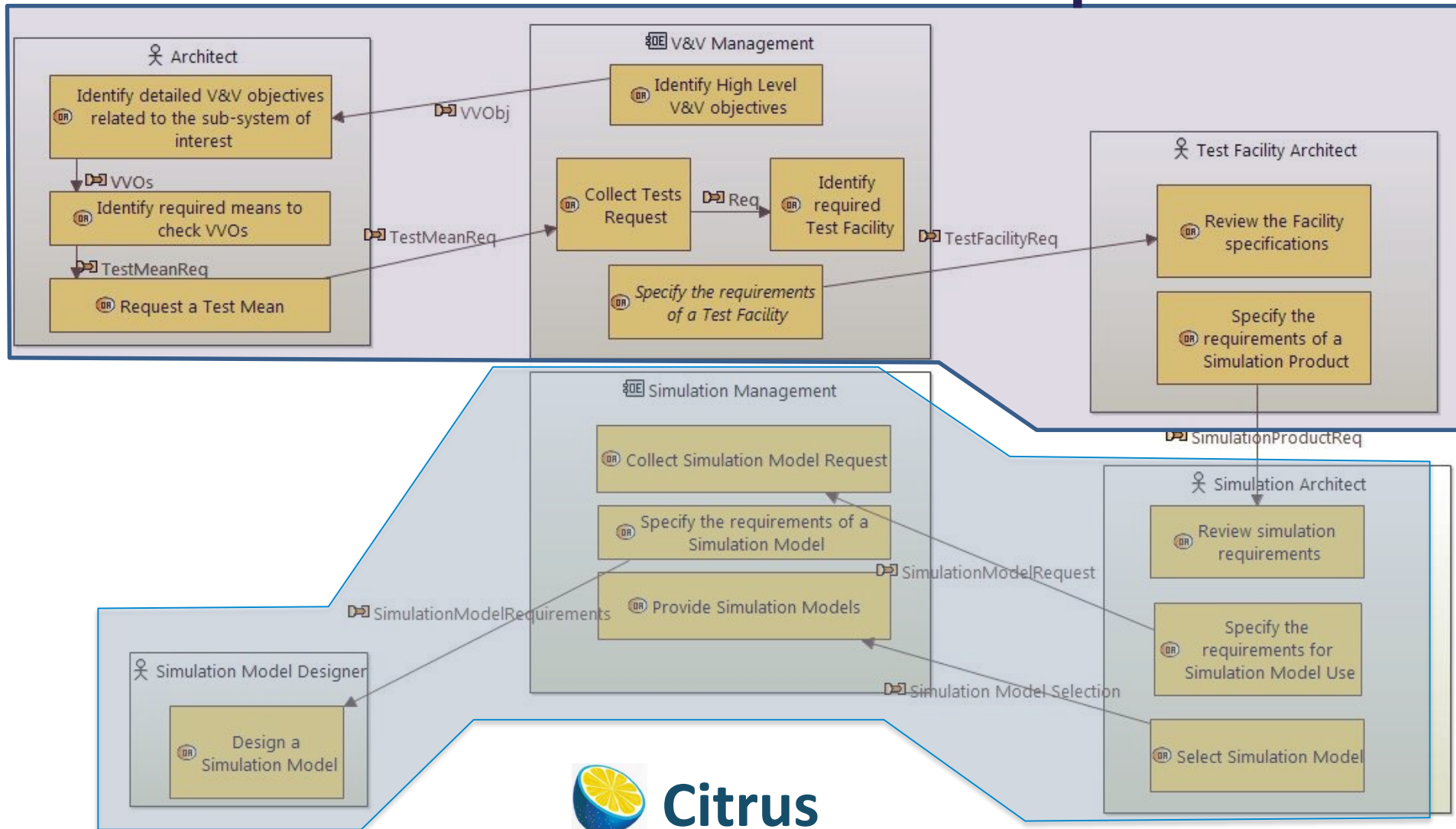
Support TestRequests for multiple UuT

Makes the distinction btw real / sim





*Demo*





## Citrus

### Capella

#### System Architecture

#### V&V Strategy

- V&V Objectives
- V&V Plans
- Test Means identification

#### Test Mean Spec

- Functional scope definition
- Allocation to Simulation

#### Describe the System to be simulated

- Functional and organic architecture
- Real interfaces
- Malfunctions / probes

#### Specify the simulation

- Identify simulation models
- Specification of the simulation models
  - Functional scope & fidelity
  - Interfaces

#### Design the simulation

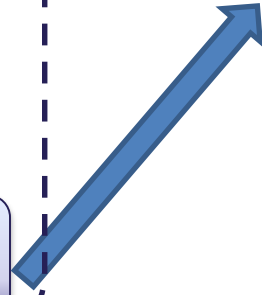
- Design the simulation execution platform
  - Infrastructure
  - Runtime Environment
- Select the models & allocate to Runtime

#### Design simulation models

- Integration of building blocks
- Modeling authoring tools bootstrap
- Compliance checking

#### Integrate & deploy simulation

- Receive simulation models : verif & patch
- Integrate simulation models
  - Connections
    - Format / deformat mgt
    - Automatic algo & scripts
  - Initialisation
- Deploy : configuration generation

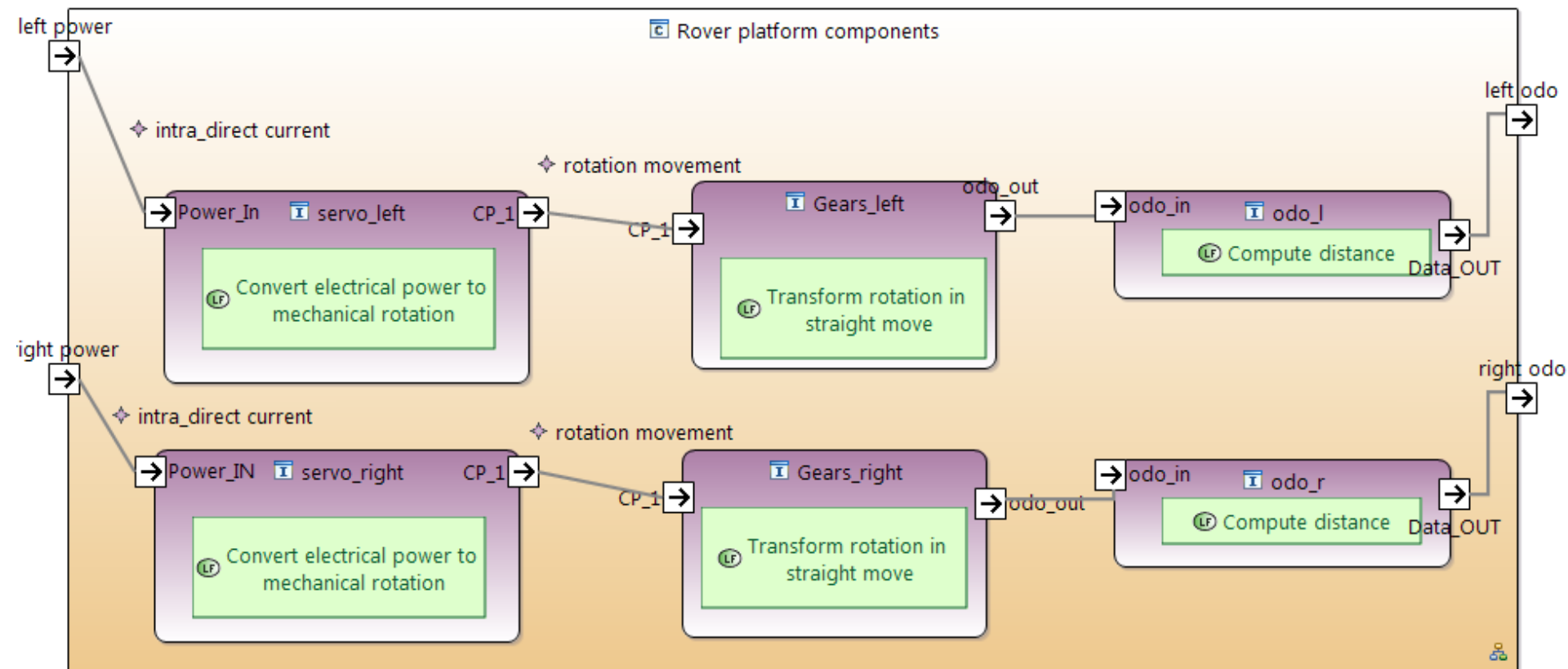


System parts allocated to simulation models

The screenshot displays a simulation design environment. At the top, a diagram shows various system parts (actuators and valves) connected to a central 'Specification' block, which is further linked to simulation models. Below this, a window titled 'FMI\_Internal' is open, showing a 'Parameters' table.

	Name	Description	Type	Causality Type	Variability Type	Can Handle Mu...	Initial Type	B
1	nand1.y	Boolean output signal	Modelica.Blocks....	local	discrete	<input type="checkbox"/>	exact	
2	booleanTable.table[2]	Vector of time points. At every tim...	Real [fmiTypesLi...	parameter	fixed	<input type="checkbox"/>	exact	
3	and1.u[1]	Vector of Boolean input signals	Boolean [fmiTyp...	local	discrete	<input type="checkbox"/>	exact	
4	rSFlipFlop.pre.pre_u_start	Start value of pre(u) at initial time	Boolean [fmiTyp...	local	fixed	<input type="checkbox"/>	exact	
5	set1.expr[2]	y = if u[i] then expr[i] else y_defau...	Boolean [fmiTyp...	local	constant	<input type="checkbox"/>	exact	
6	booleanPulse1.period	Time for one period	Modelica.SIunits....	parameter	tunable	<input type="checkbox"/>	exact	
7	falling.y	Boolean output signal	Modelica.Blocks....	local	discrete	<input type="checkbox"/>	exact	
8	sampleTriggerSet.y	Connector of Boolean output signal	Modelica.Blocks....	local	discrete	<input type="checkbox"/>	exact	
9	xor1.y	Boolean output signal	Modelica.Blocks....	local	discrete	<input type="checkbox"/>	exact	
10	rising.u	Boolean input signal	Modelica.Blocks....	local	discrete	<input type="checkbox"/>	exact	
11	or2.y	Boolean output signal	Modelica.Blocks....	local	discrete	<input type="checkbox"/>	exact	
12	showValue3.showActive		Modelica.Blocks....	local	discrete	<input type="checkbox"/>	exact	
13	or1.u[1]	Vector of Boolean input signals	Boolean [fmiTvp...	local	discrete	<input type="checkbox"/>	exact	

At the bottom of the window, there are tabs for 'Identification', 'Types', 'Units', 'FMI\_IN', 'FMI\_OUT', and 'FMI\_Internal'.



## ***Backlog***

- Transition to a dedicated project for TestMean design
- Malfunctions & Observability points
- VVO refinement & cascading
- Compliance btw TestRequest and TestMeans
- TestRequest & TestMeans versions
- Test procedures specifications
- Integration V&V Planning

