MATLAB EXPO
2021

아키텍처와 요구사항에 대한 할당 워크플로우

홍 혁기

MathWorks

What we’ve heard from YOU

- Model-Based Systems Engineering is a huge improvement over document-based methods

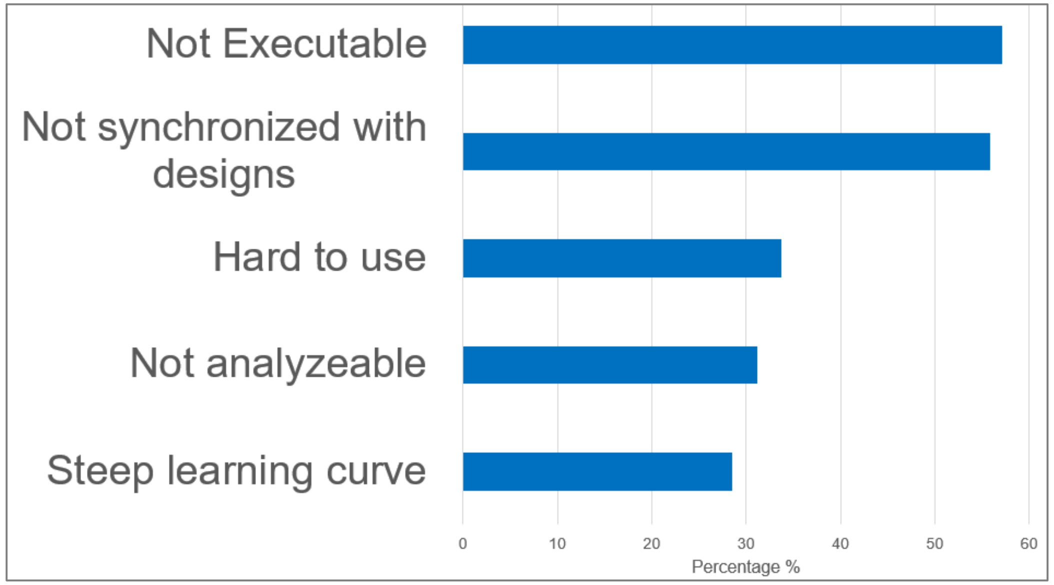
What we’ve heard from YOU

- Model-Based Systems Engineering is a huge improvement over document-based methods

- Existing tools are often missing key capabilities

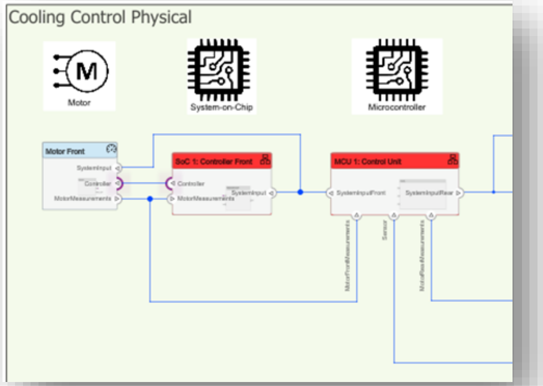
What we’ve heard from YOU

- Model-Based Systems Engineering is a huge improvement over document-based methods
- Existing tools are often missing key capabilities



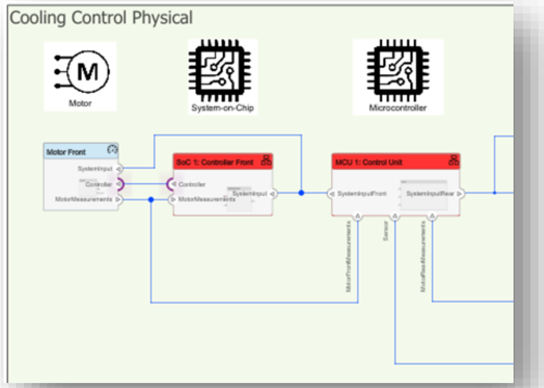
Why MBSE with MathWorks tools?

Be Intuitive

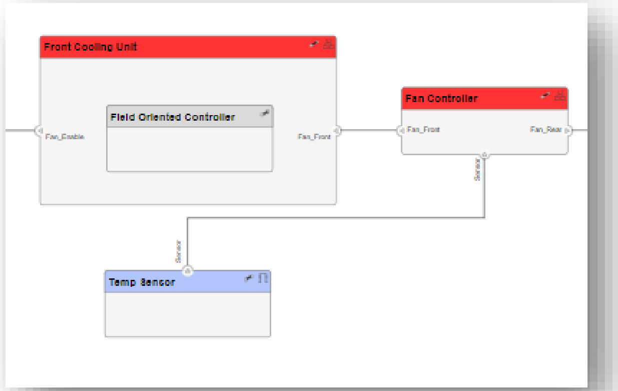


Why MBSE with MathWorks tools?

Be Intuitive

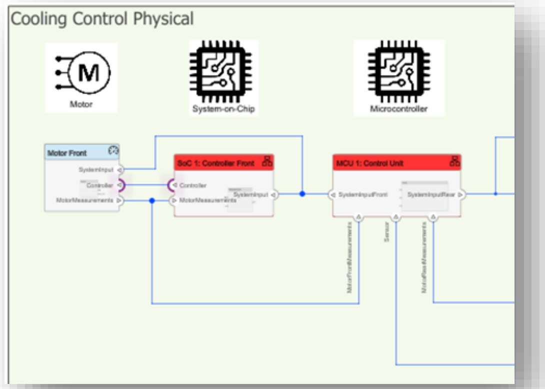


Tackle Complexity

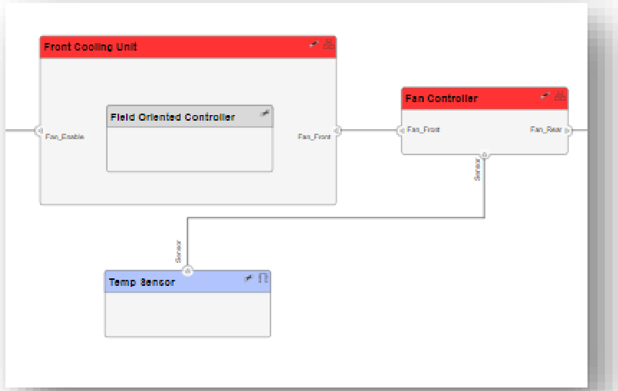


Why MBSE with MathWorks tools?

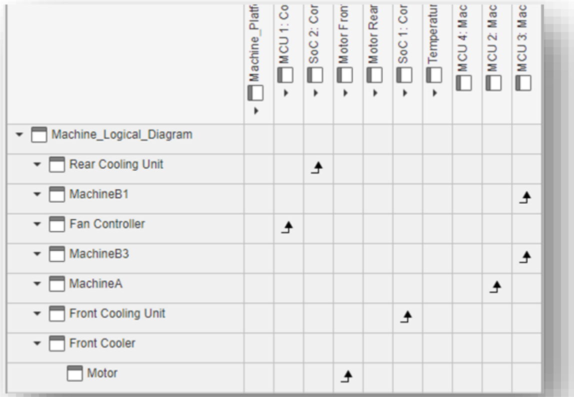
Be Intuitive



Tackle Complexity



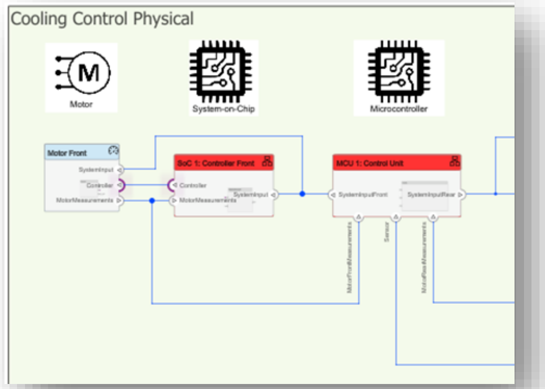
Facilitate Traceability



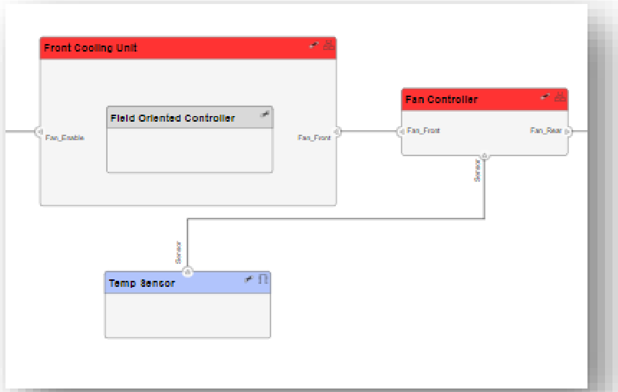
	Machine_Platf	MCU 1: Co	SoC 2: Cor	Motor Fron	Motor Rear	SoC 1: Cor	Temperatur	MCU 4: Mac	MCU 2: Mac	MCU 3: Mac
Machine_Logical_Diagram										
Rear Cooling Unit			↑							
MachineB1										↑
Fan Controller		↑								
MachineB3										↑
MachineA									↑	
Front Cooling Unit						↑				
Front Cooler										
Motor				↑						

Why MBSE with MathWorks tools?

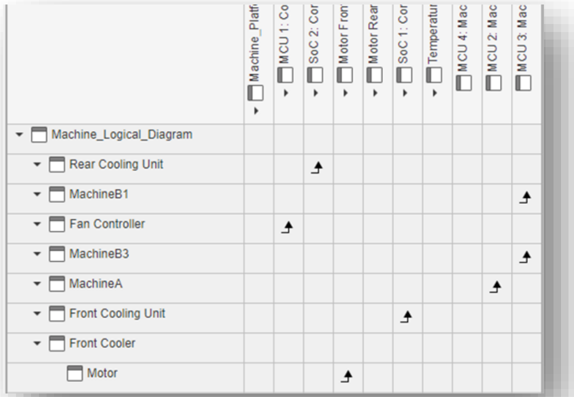
Be Intuitive



Tackle Complexity

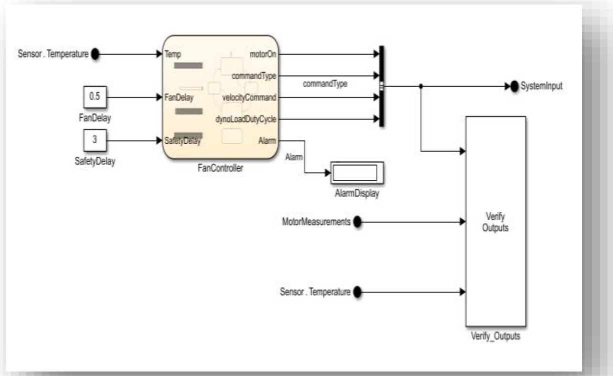


Facilitate Traceability



	Machine_Platf	MCU 1: Co	SoC 2: Cor	Motor Fron	Motor Rear	SoC 1: Cor	Temperatur	MCU 4: Mac	MCU 2: Mac	MCU 3: Mac
Machine_Logical_Diagram										
Rear Cooling Unit			↑							
MachineB1										↑
Fan Controller		↑								
MachineB3										↑
MachineA									↑	
Front Cooling Unit						↑				
Front Cooler										
Motor				↑						

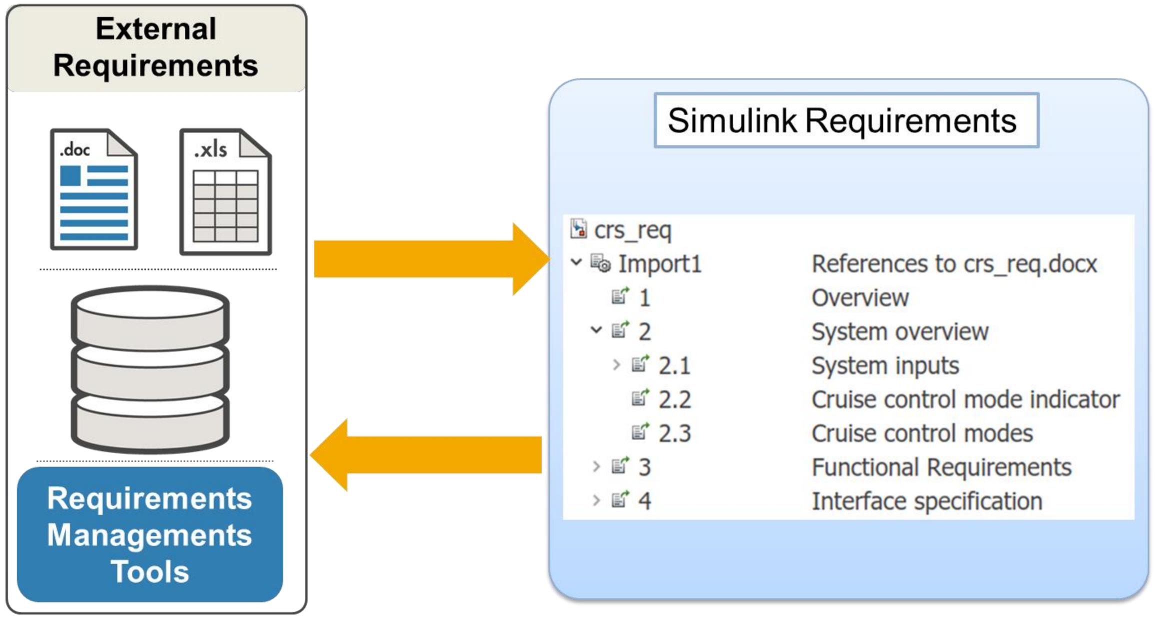
Enable Implementation



Provide Traceability

Key Takeaways

- You can import, write, and store textual requirements right in the same environment as your architecture and design models.

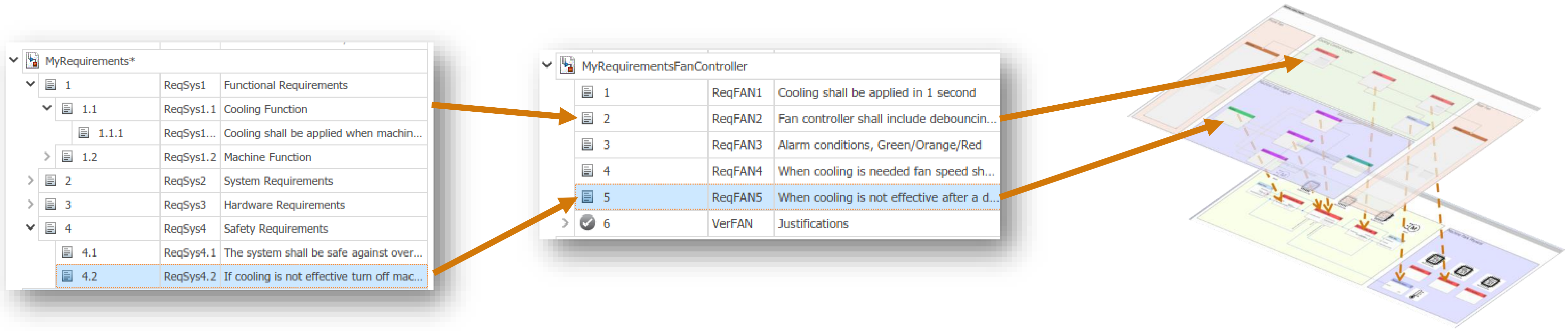


crs_req	
Import1	References to crs_req.docx
1	Overview
2	System overview
2.1	System inputs
2.2	Cruise control mode indicator
2.3	Cruise control modes
3	Functional Requirements
4	Interface specification

Key Takeaways

- You can import, write, and store textual requirements right in the same environment as your architecture and design models.

- You can establish relationships among multiple requirements and architecture artifacts to understand the impact of changes.



MyRequirements*

1	ReqSys1	Functional Requirements
1.1	ReqSys1.1	Cooling Function
1.1.1	ReqSys1...	Cooling shall be applied when machin...
1.2	ReqSys1.2	Machine Function
2	ReqSys2	System Requirements
3	ReqSys3	Hardware Requirements
4	ReqSys4	Safety Requirements
4.1	ReqSys4.1	The system shall be safe against over...
4.2	ReqSys4.2	If cooling is not effective turn off mac...

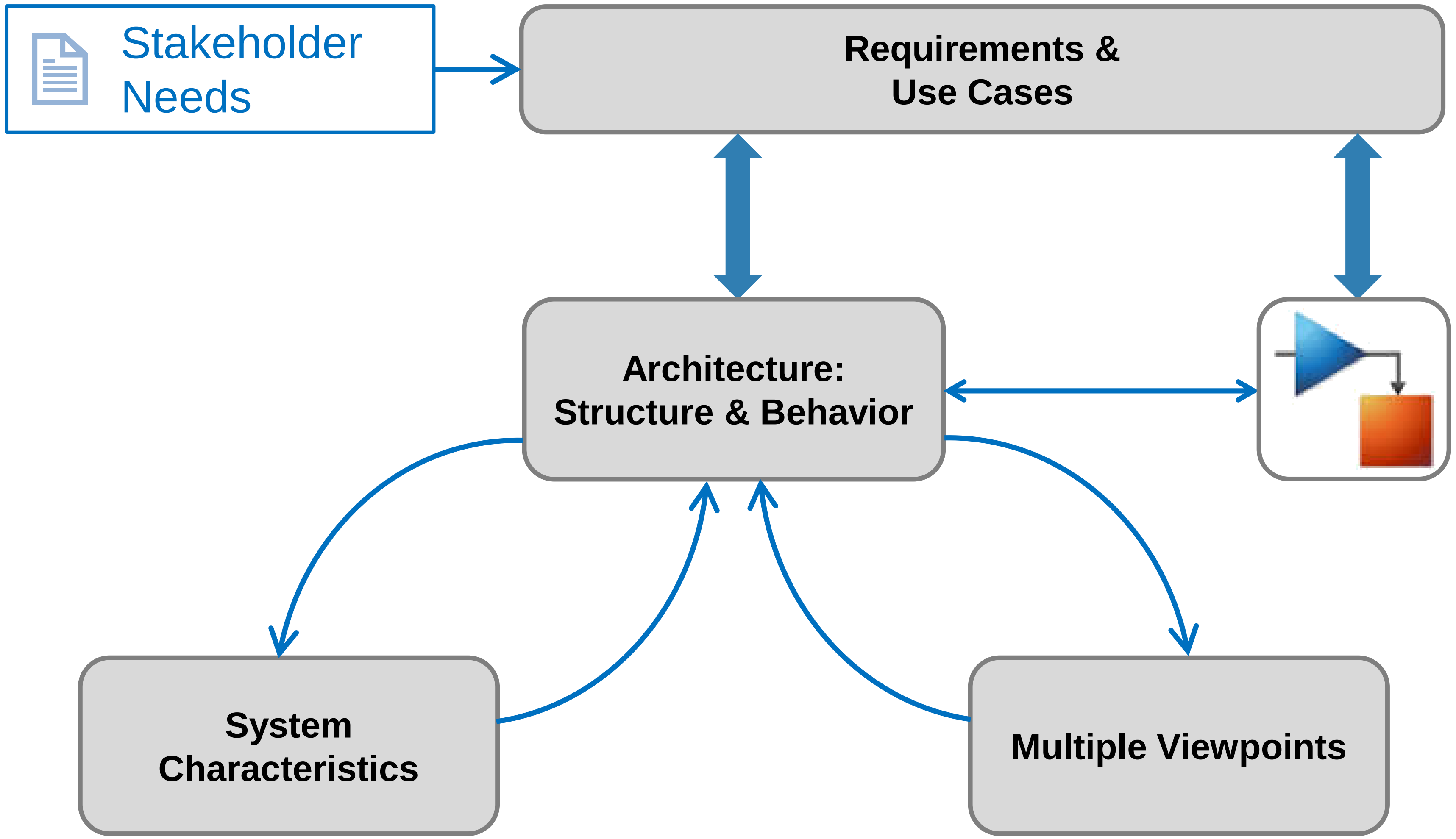
MyRequirementsFanController

1	ReqFAN1	Cooling shall be applied in 1 second
2	ReqFAN2	Fan controller shall include debouncin...
3	ReqFAN3	Alarm conditions, Green/Orange/Red
4	ReqFAN4	When cooling is needed fan speed sh...
5	ReqFAN5	When cooling is not effective after a d...
6	VerFAN	Justifications

Key Takeaways

- You can import, write, and store textual requirements right in the same environment as your architecture and design models.
- You can establish relationships among multiple requirements and architecture artifacts to understand the impact of changes.
- You can visualize those relationships to assess the completeness of your system.

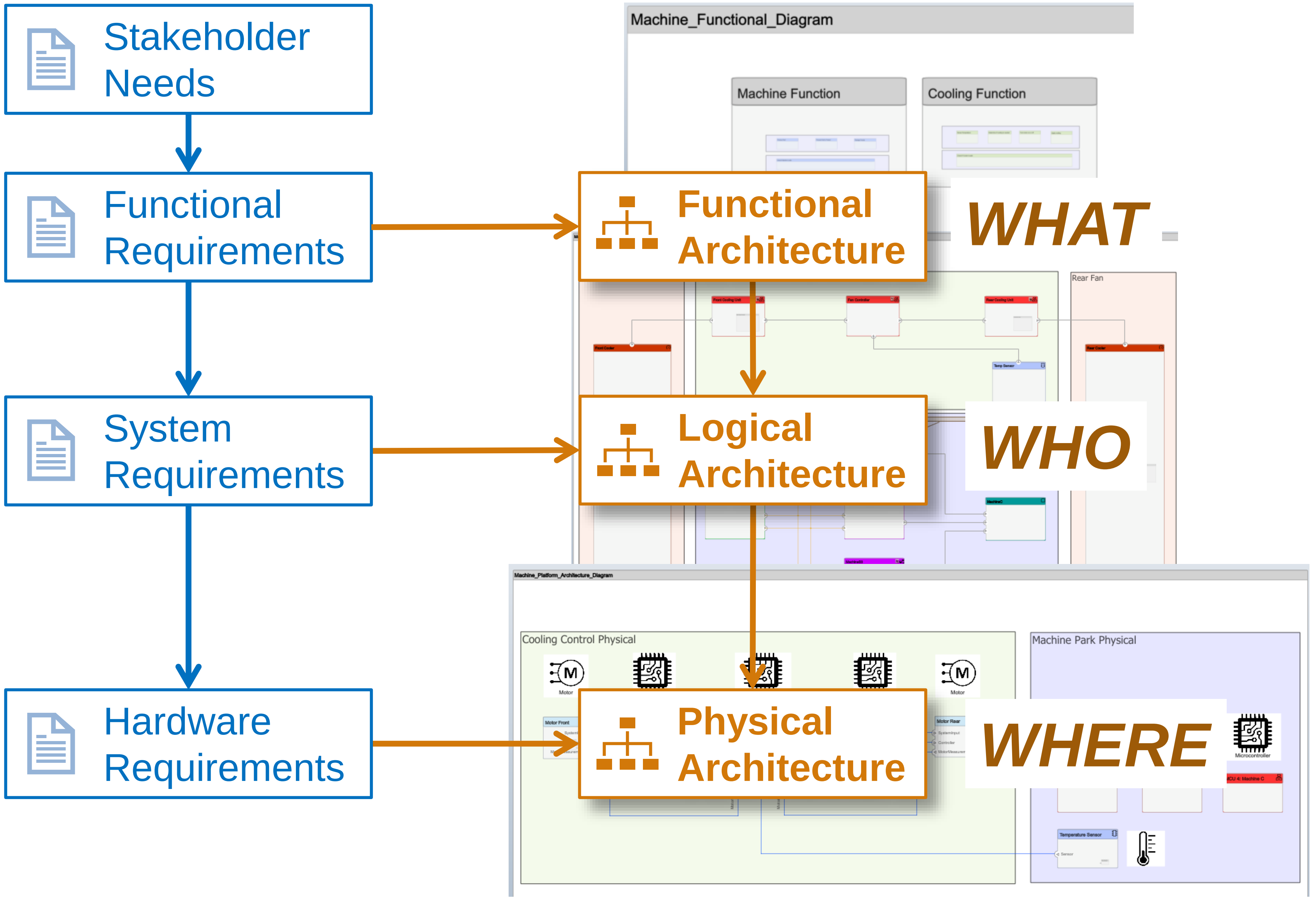
Typical System Engineering Tasks



Stakeholder
Needs

Requirements &
Use Cases

Architecture:
Structure & Behavior


Stakeholder Needs
Functional Requirements
System Requirements
Hardware Requirements
Machine_Functional_Diagram
Machine Function
Cooling Function
Functional Architecture
WHAT
Rear Fan
Logical Architecture
WHO
Machine_Platform_Architecture_Diagram
Cooling Control Physical
Machine Park Physical
Motor
Motor Front
Motor Rear
Physical Architecture
WHERE
Microcontroller
Temperature Sensor

Traceability between artifacts

Traceability between artifacts

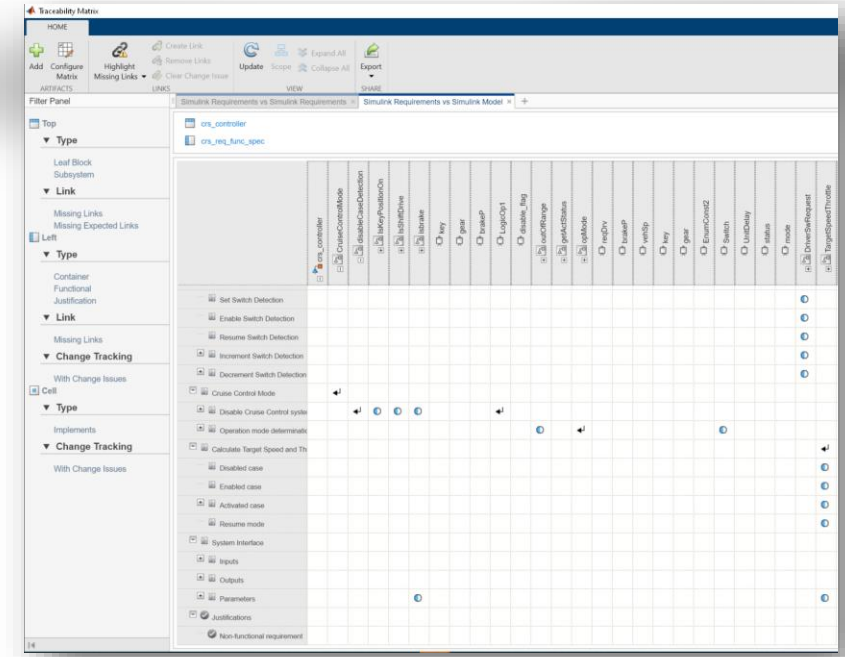
Index	ID	Summary
MyRequirements		
1	ReqSys1	Functional Requirements
2	ReqSys2	System Requirements
3	ReqSys3	Hardware Requirements
4	ReqSys4	Safety Requirements
MyRequirementsFanController		
MyRequirementsFieldOrientedController		

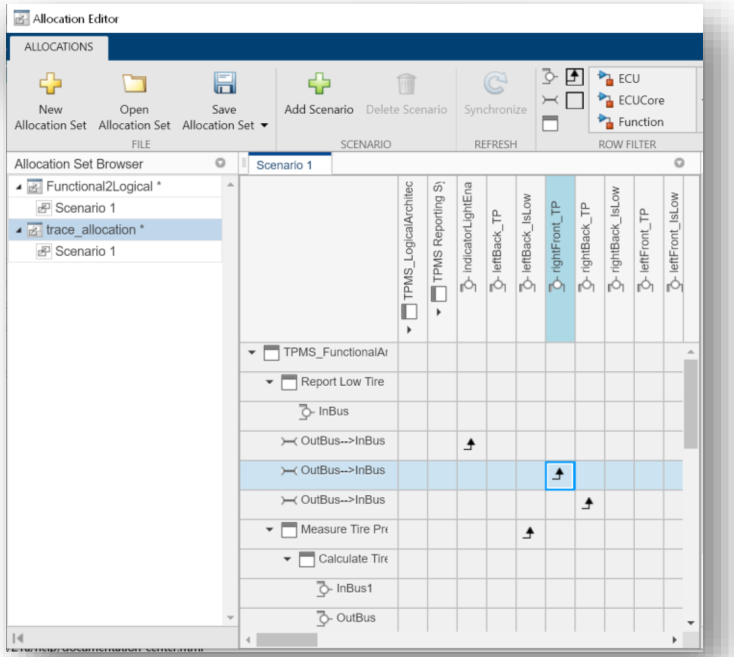
Traceability between artifacts

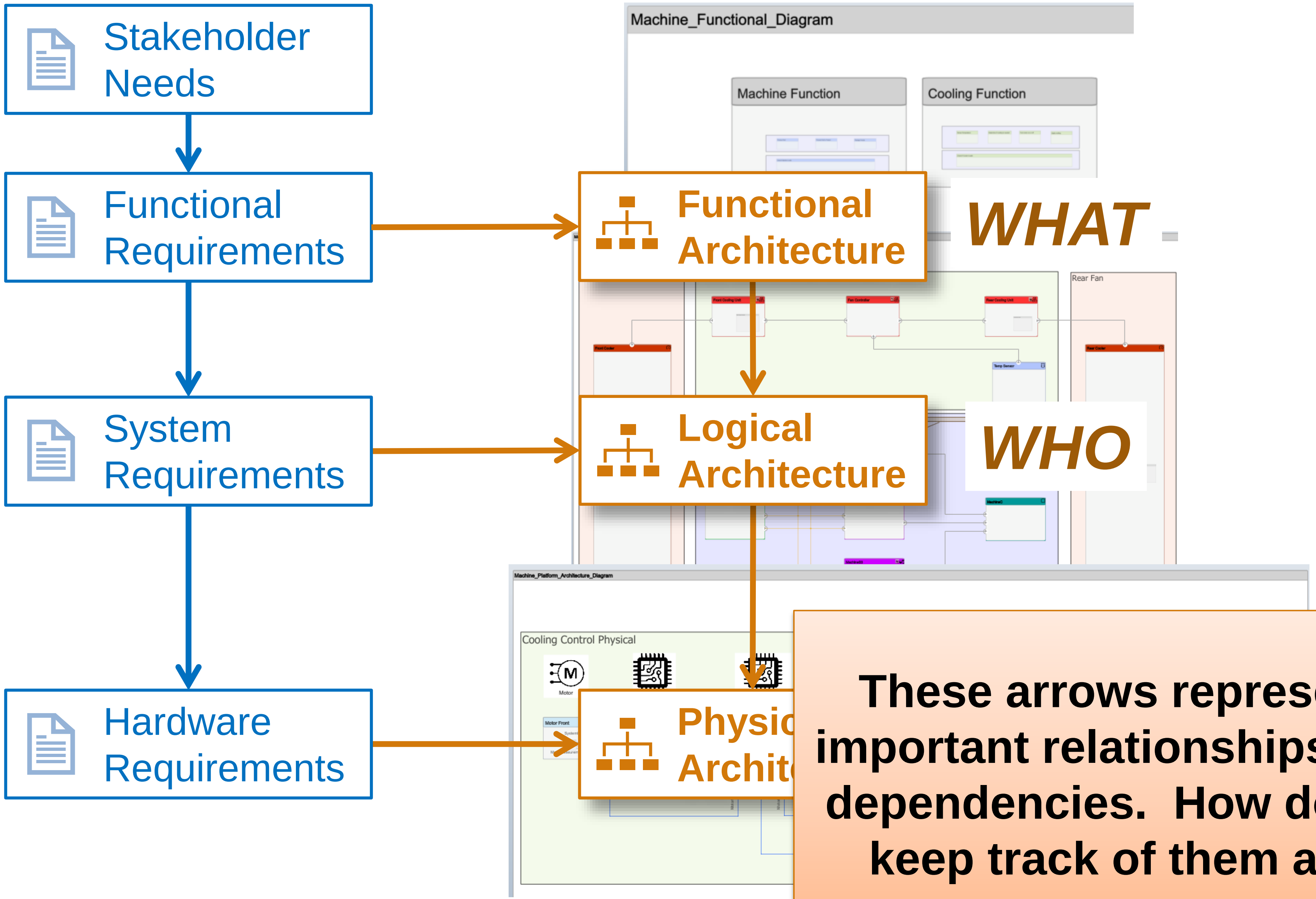
Requirements Editor
REQUIREMENTS
New Requirement Set
Open
Save
Import
Close
Add Requirement
Delete
Promote Requirement
Demote Requirement
FILE
REQUIREMENTS
Index
ID
Summary
MyRequirements
ReqSys1 Functional Requirements
ReqSys2 System Requirements
ReqSys3 Hardware Requirements
ReqSys4 Safety Requirements
MyRequirementsFanController
MyRequirementsFieldOrientedController

Traceability between artifacts

Traceability between artifacts







These arrows represent important relationships and dependencies. How do we keep track of them all?

Stakeholder Needs

Functional Requirements

System Requirements

Hardware Requirements

Functional Architecture

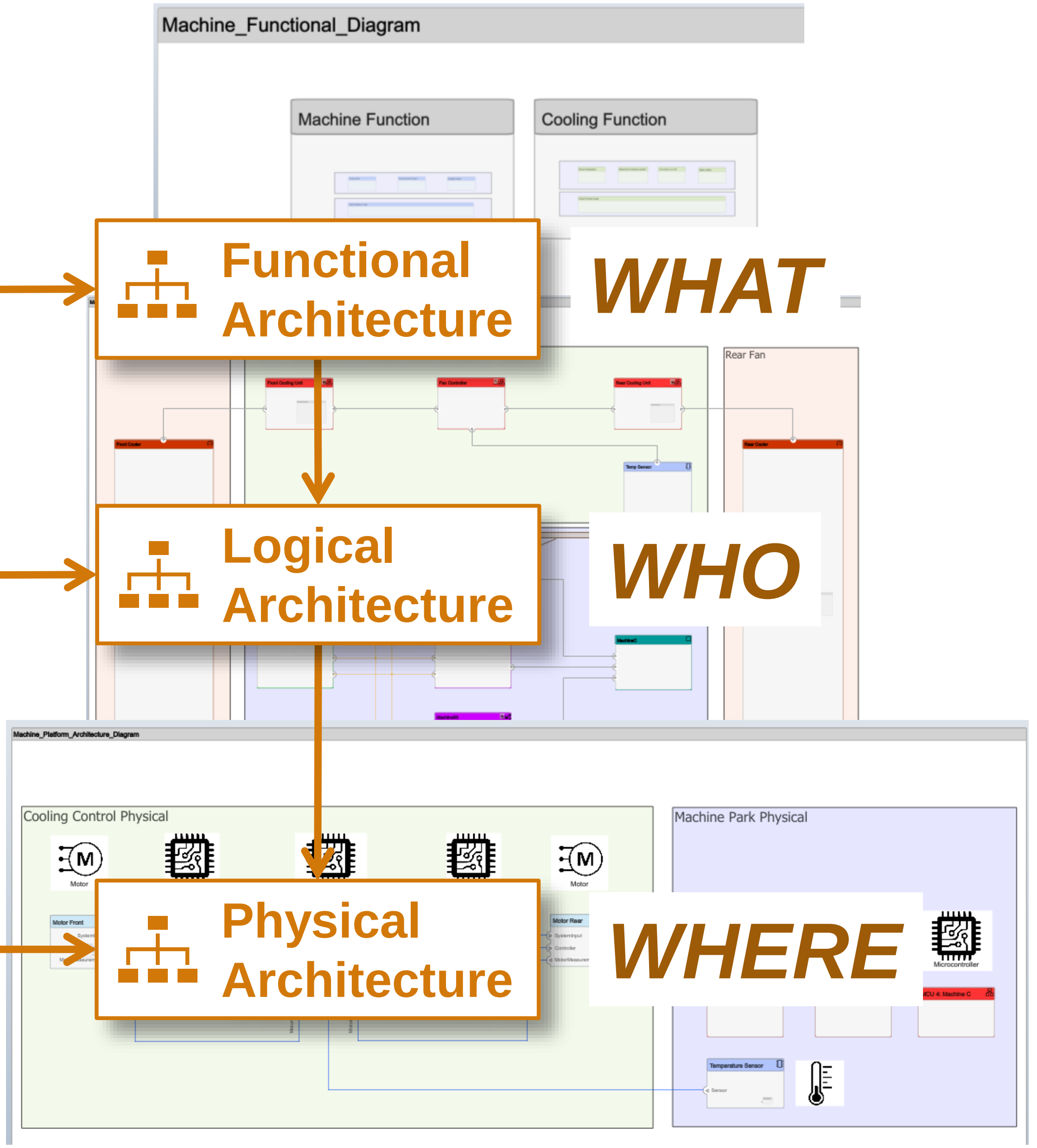
WHAT

Logical Architecture

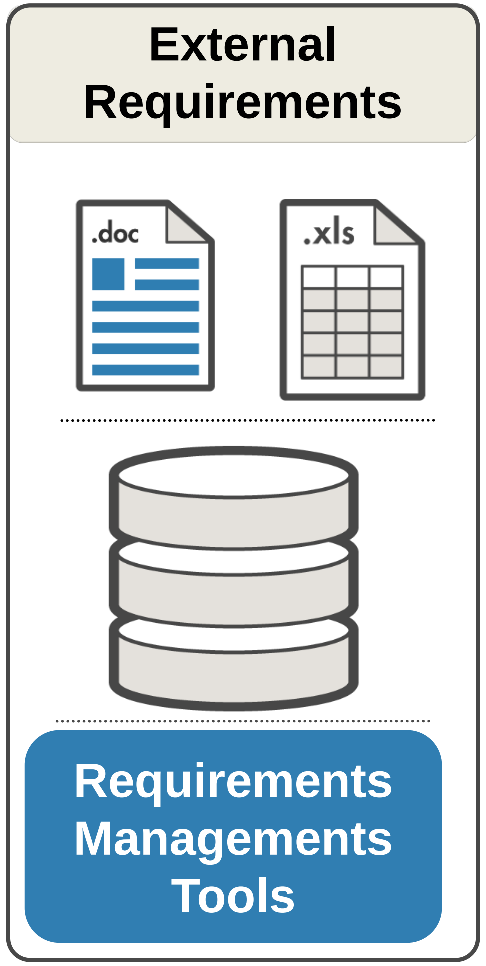
WHO

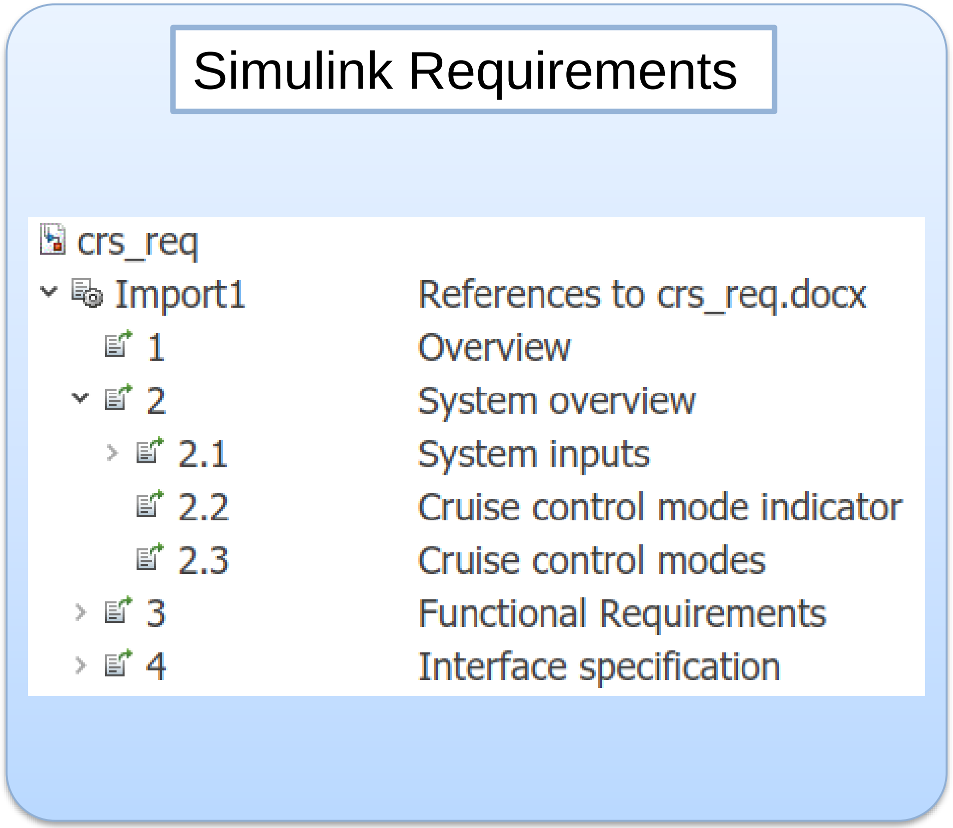
Physical Architecture

WHERE



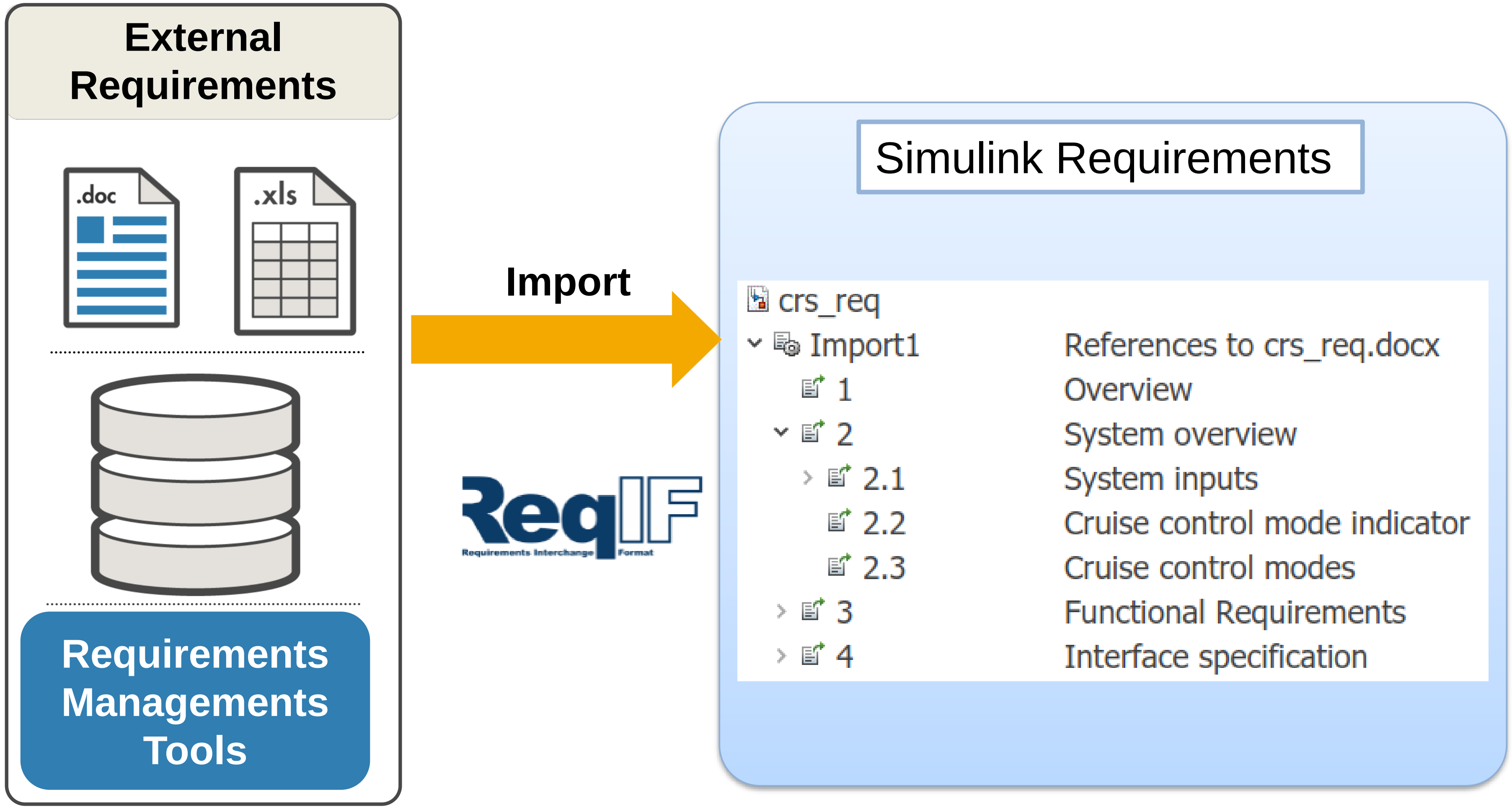
Exchange Data with Third Party Requirements Tools





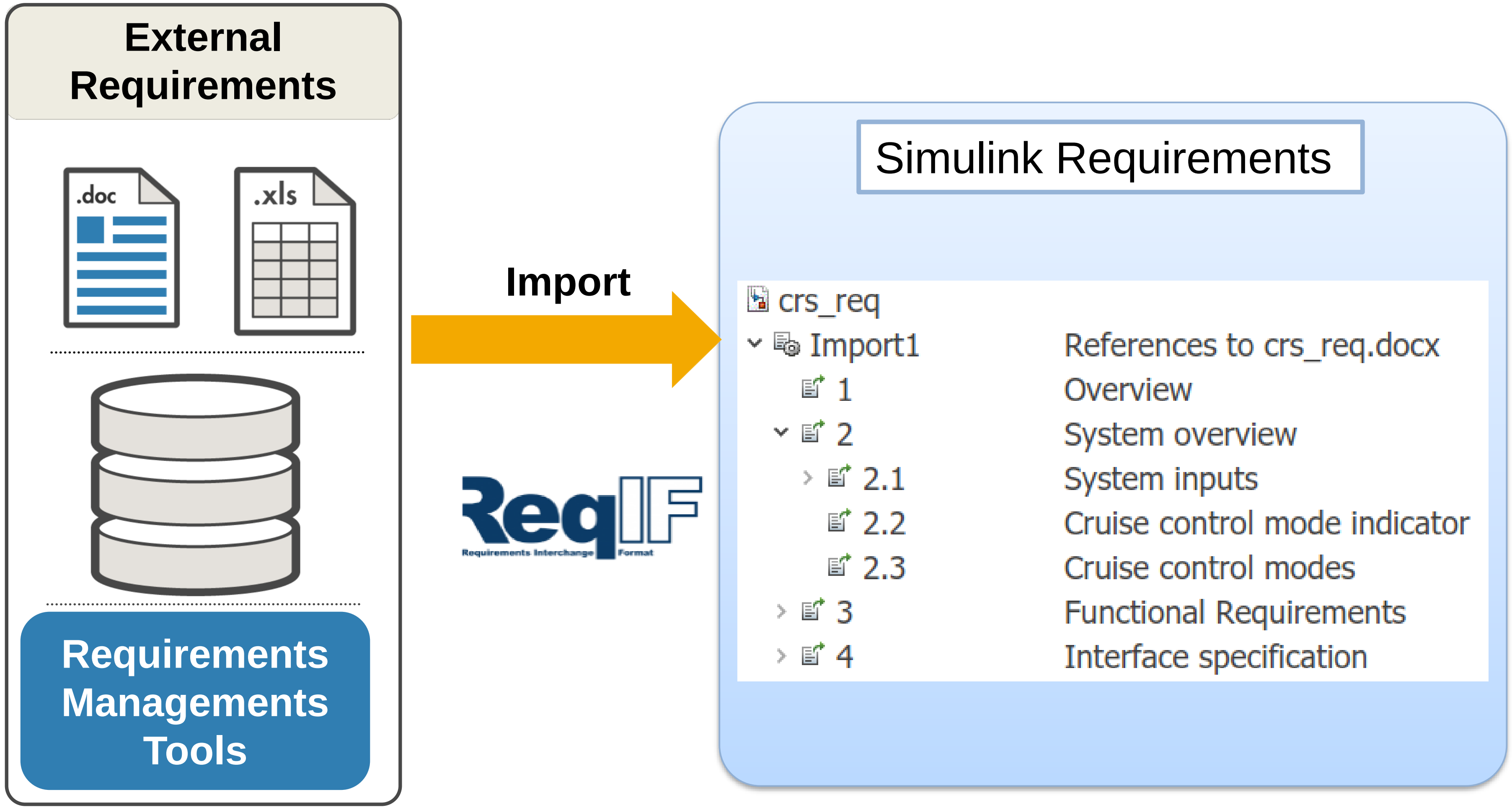
crs_req	
Import1	References to crs_req.docx
1	Overview
2	System overview
2.1	System inputs
2.2	Cruise control mode indicator
2.3	Cruise control modes
3	Functional Requirements
4	Interface specification

Exchange Data with Third Party Requirements Tools


External Requirements
.doc
.xls
Requirements Managements Tools
Import
ReqIF
Requirements Interchange Format
Simulink Requirements

crs_req	
Import1	References to crs_req.docx
1	Overview
2	System overview
2.1	System inputs
2.2	Cruise control mode indicator
2.3	Cruise control modes
3	Functional Requirements
4	Interface specification

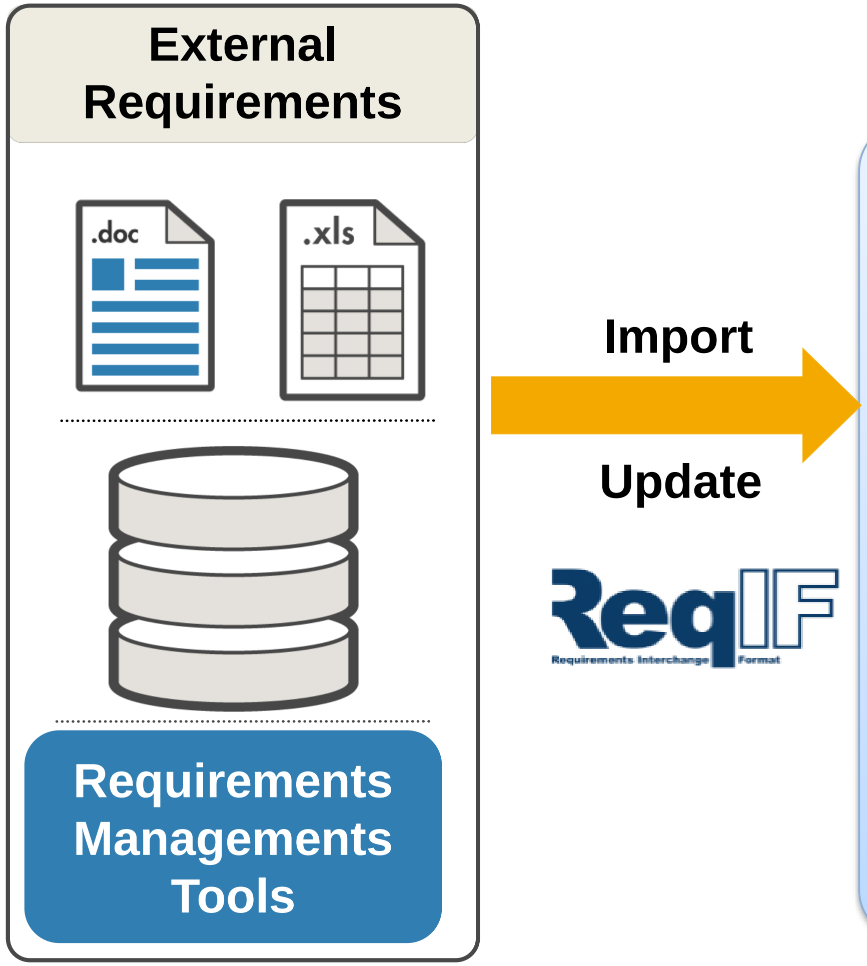
Exchange Data with Third Party Requirements Tools

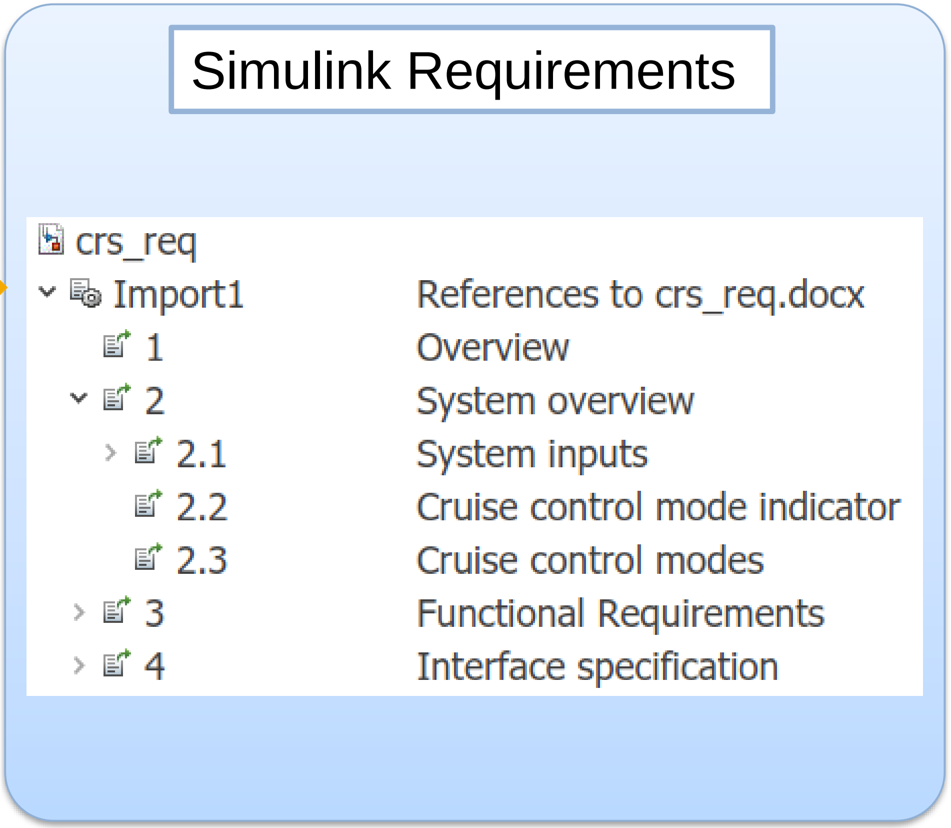

External Requirements
.doc
.xls
Requirements Managements Tools
Import
ReqIF
Requirements Interchange Format
Simulink Requirements

crs_req	
Import1	References to crs_req.docx
1	Overview
2	System overview
2.1	System inputs
2.2	Cruise control mode indicator
2.3	Cruise control modes
3	Functional Requirements
4	Interface specification

- Import from:
  - Word / Excel
  - IBM® Rational® DOORS®
  - DOORS Next
  - ReqIF™ standard
- Synchronize changes from source
- Edit and add further details to import
- Export ReqIF
  - Enables roundtrip with external tools

Exchange Data with Third Party Requirements Tools

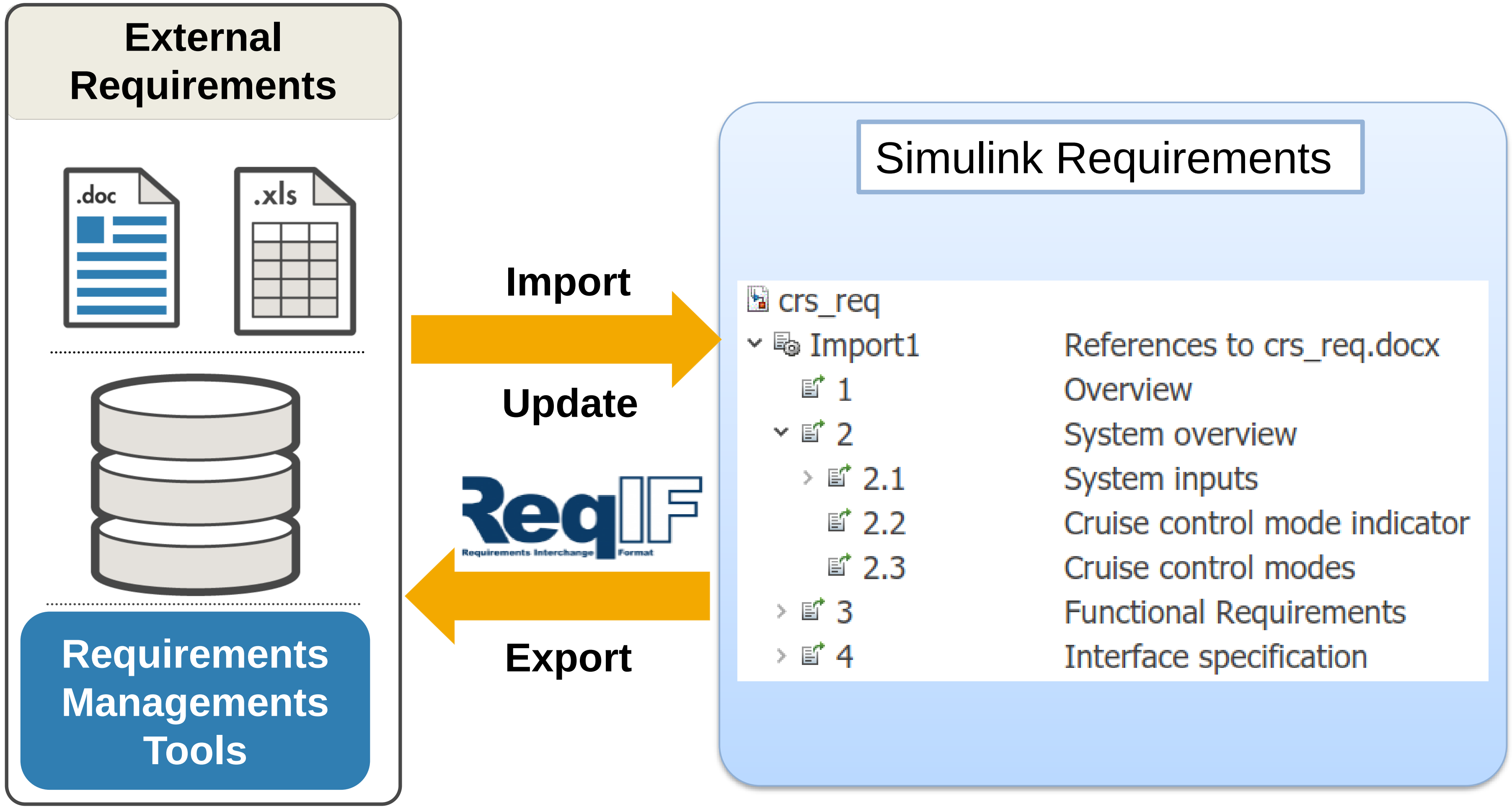

External Requirements
.doc
.xls
Requirements Managements Tools
Import
Update
ReqIF
Requirements Interchange Format



```
crs_req
⌄ Import1        References to crs_req.docx
    1            Overview
  ⌄ 2            System overview
    › 2.1        System inputs
      2.2        Cruise control mode indicator
      2.3        Cruise control modes
  › 3            Functional Requirements
  › 4            Interface specification
```

- Import from:
  - Word / Excel
  - IBM® Rational® DOORS®
  - DOORS Next
  - ReqIF™ standard
- Synchronize changes from source
- Edit and add further details to import
- Export ReqIF
  - Enables roundtrip with external tools

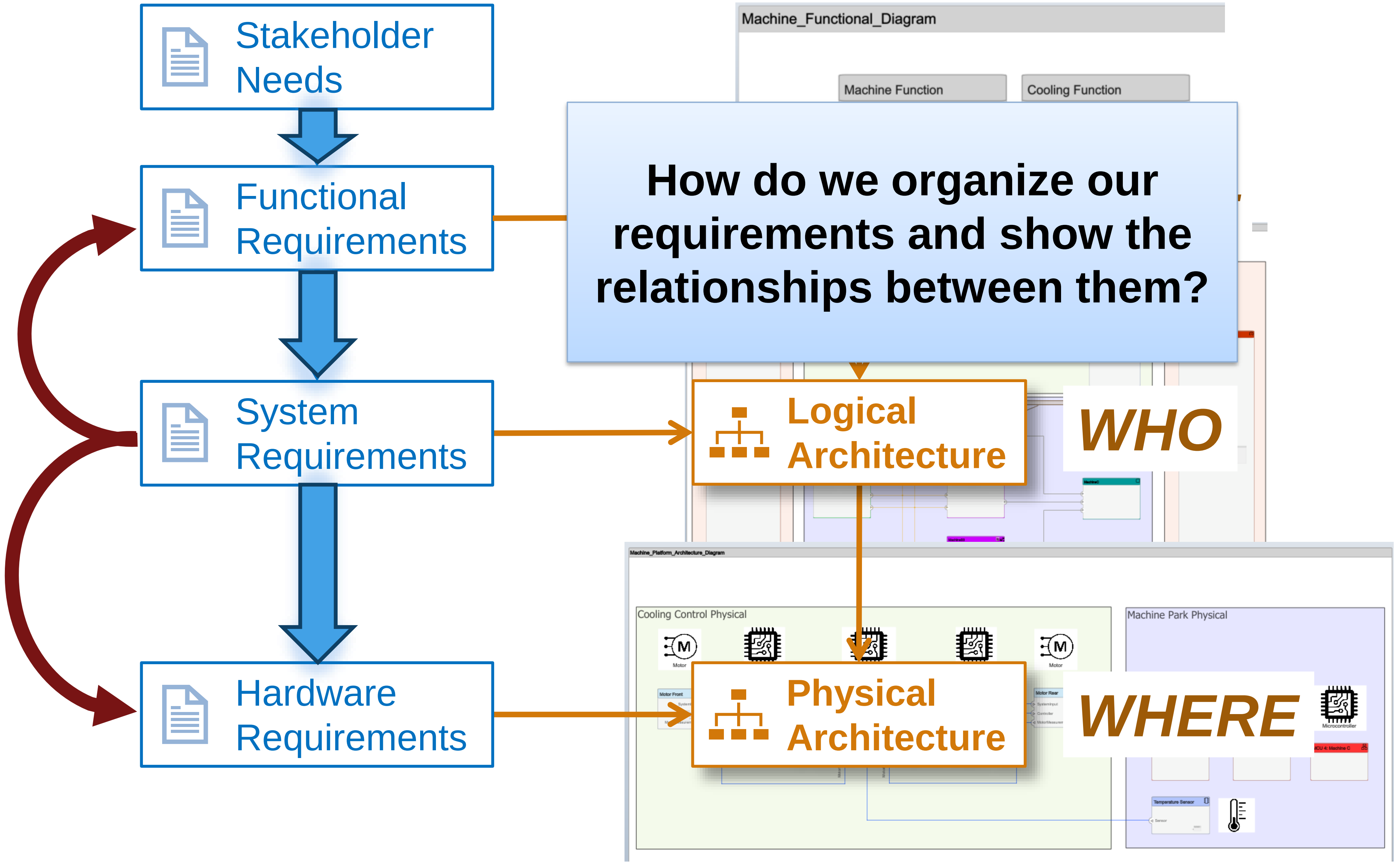
Exchange Data with Third Party Requirements Tools



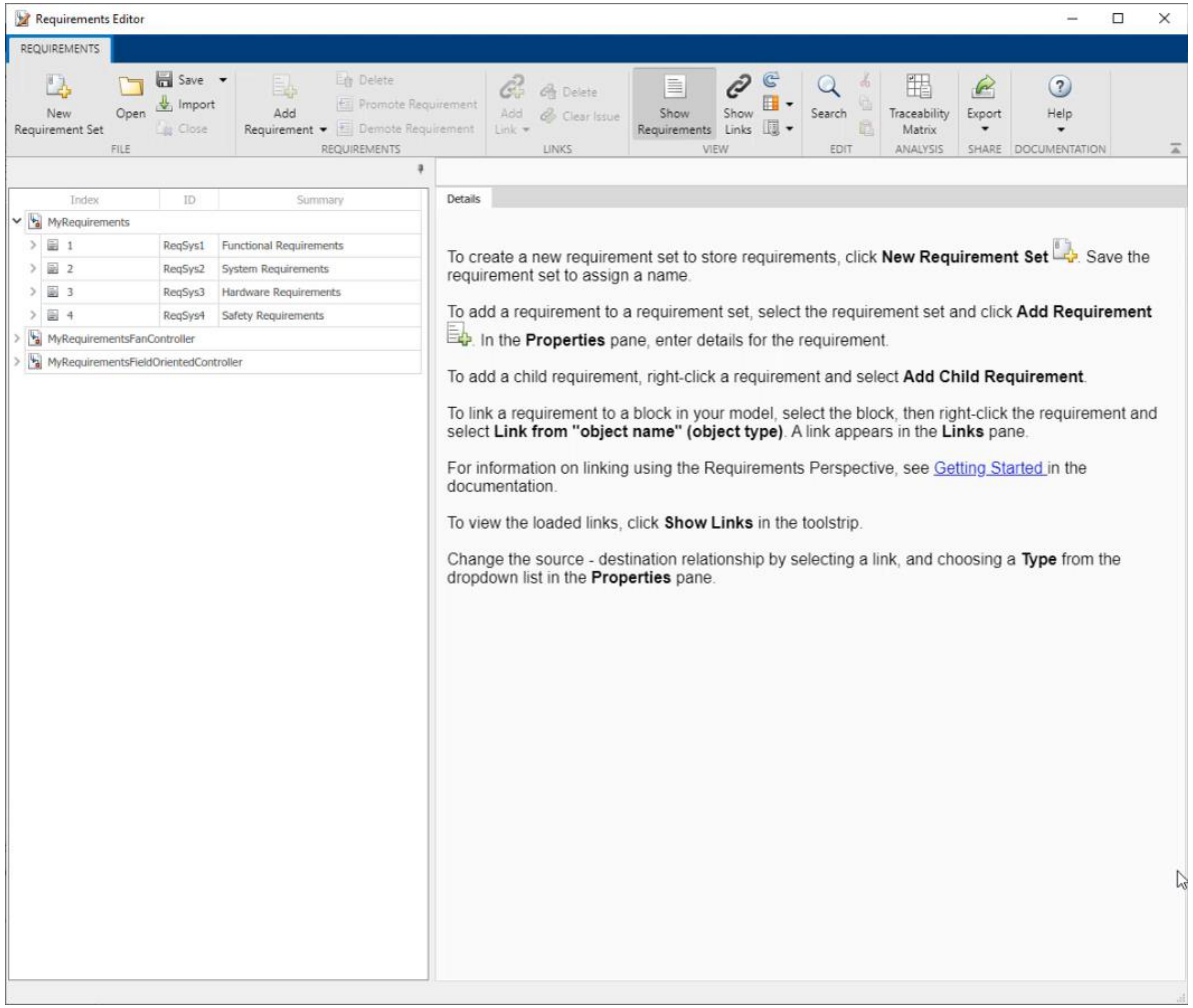
crs_req	
Import1	References to crs_req.docx
1	Overview
2	System overview
2.1	System inputs
2.2	Cruise control mode indicator
2.3	Cruise control modes
3	Functional Requirements
4	Interface specification

- Import from:
  - Word / Excel
  - IBM® Rational® DOORS®
  - DOORS Next
  - ReqIF™ standard
- Synchronize changes from source
- Edit and add further details to import
- Export ReqIF
  - Enables roundtrip with external tools

How do we organize our requirements and show the relationships between them?



Traceability between Requirements



Index	ID	Summary
MyRequirements		
1	ReqSys1	Functional Requirements
2	ReqSys2	System Requirements
3	ReqSys3	Hardware Requirements
4	ReqSys4	Safety Requirements
MyRequirementsFanController		
MyRequirementsFieldOrientedController		

To create a new requirement set to store requirements, click **New Requirement Set**. Save the requirement set to assign a name.

To add a requirement to a requirement set, select the requirement set and click **Add Requirement**. In the **Properties** pane, enter details for the requirement.

To add a child requirement, right-click a requirement and select **Add Child Requirement**.

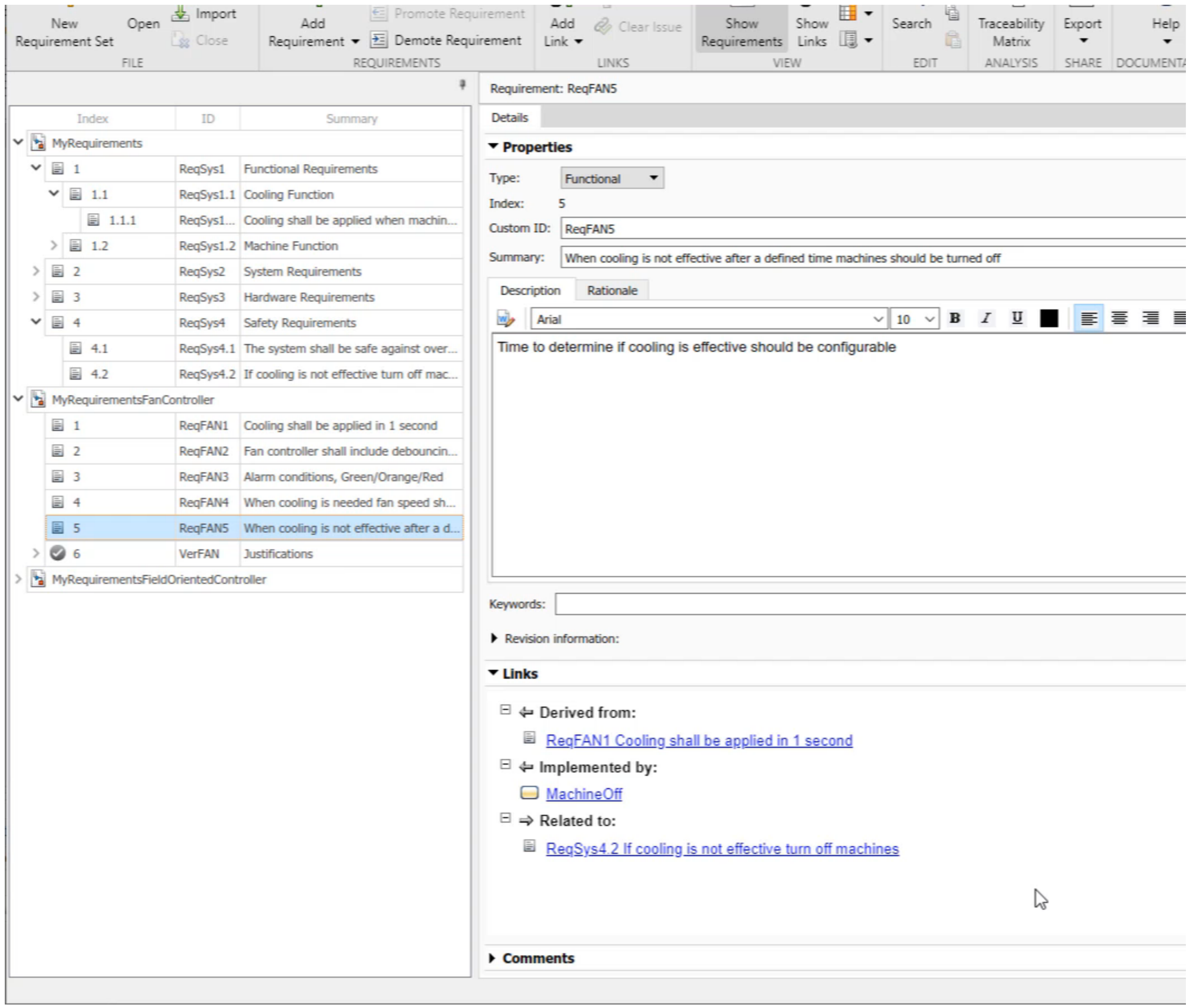
To link a requirement to a block in your model, select the block, then right-click the requirement and select **Link from "object name" (object type)**. A link appears in the **Links** pane.

For information on linking using the Requirements Perspective, see Getting Started in the documentation.

To view the loaded links, click **Show Links** in the toolstrip.

Change the source - destination relationship by selecting a link, and choosing a **Type** from the dropdown list in the **Properties** pane.

Change Management



Index	ID	Summary
MyRequirements		
1	ReqSys1	Functional Requirements
1.1	ReqSys1.1	Cooling Function
1.1.1	ReqSys1...	Cooling shall be applied when machin...
1.2	ReqSys1.2	Machine Function
2	ReqSys2	System Requirements
3	ReqSys3	Hardware Requirements
4	ReqSys4	Safety Requirements
4.1	ReqSys4.1	The system shall be safe against over...
4.2	ReqSys4.2	If cooling is not effective turn off mac...
MyRequirementsFanController		
1	ReqFAN1	Cooling shall be applied in 1 second
2	ReqFAN2	Fan controller shall include debouncin...
3	ReqFAN3	Alarm conditions, Green/Orange/Red
4	ReqFAN4	When cooling is needed fan speed sh...
5	ReqFAN5	When cooling is not effective after a d...
6	VerFAN	Justifications
MyRequirementsFieldOrientedController		

Requirement: ReqFAN5

Details

Properties

Type: Functional
Index: 5
Custom ID: ReqFAN5
Summary: When cooling is not effective after a defined time machines should be turned off

Description | Rationale

Arial 10

Time to determine if cooling is effective should be configurable

Keywords:

Revision information:

Links

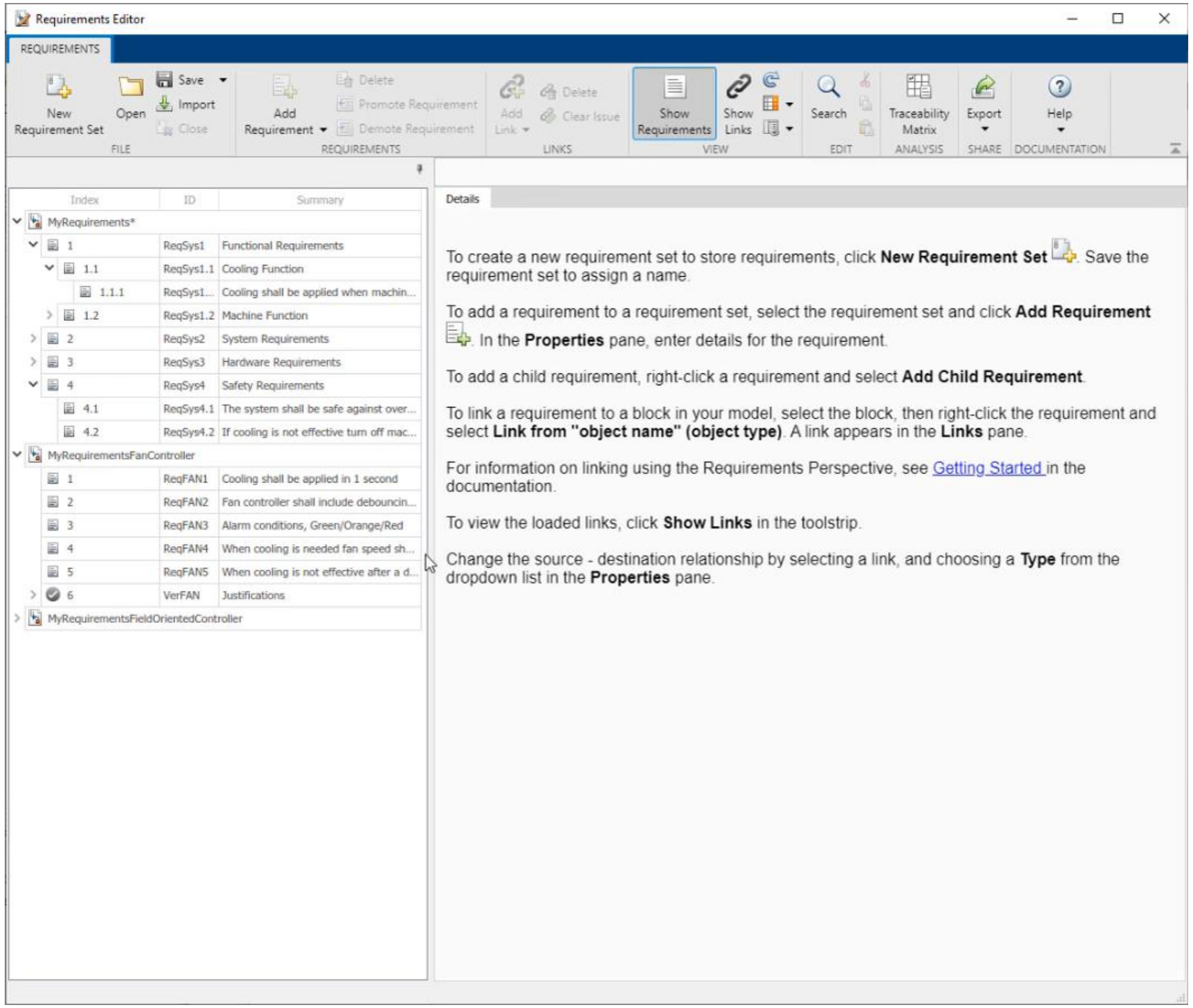
Derived from:
ReqFAN1 Cooling shall be applied in 1 second

Implemented by:
MachineOff

Related to:
ReqSys4.2 If cooling is not effective turn off machines

Comments

Requirements Traceability Metrics



Index	ID	Summary
MyRequirements*		
1	ReqSys1	Functional Requirements
1.1	ReqSys1.1	Cooling Function
1.1.1	ReqSys1...	Cooling shall be applied when machin...
1.2	ReqSys1.2	Machine Function
2	ReqSys2	System Requirements
3	ReqSys3	Hardware Requirements
4	ReqSys4	Safety Requirements
4.1	ReqSys4.1	The system shall be safe against over...
4.2	ReqSys4.2	If cooling is not effective turn off mac...
MyRequirementsFanController		
1	ReqFAN1	Cooling shall be applied in 1 second
2	ReqFAN2	Fan controller shall include debouncin...
3	ReqFAN3	Alarm conditions, Green/Orange/Red
4	ReqFAN4	When cooling is needed fan speed sh...
5	ReqFAN5	When cooling is not effective after a d...
6	VerFAN	Justifications
MyRequirementsFieldOrientedController		

To create a new requirement set to store requirements, click **New Requirement Set**. Save the requirement set to assign a name.

To add a requirement to a requirement set, select the requirement set and click **Add Requirement**. In the **Properties** pane, enter details for the requirement.

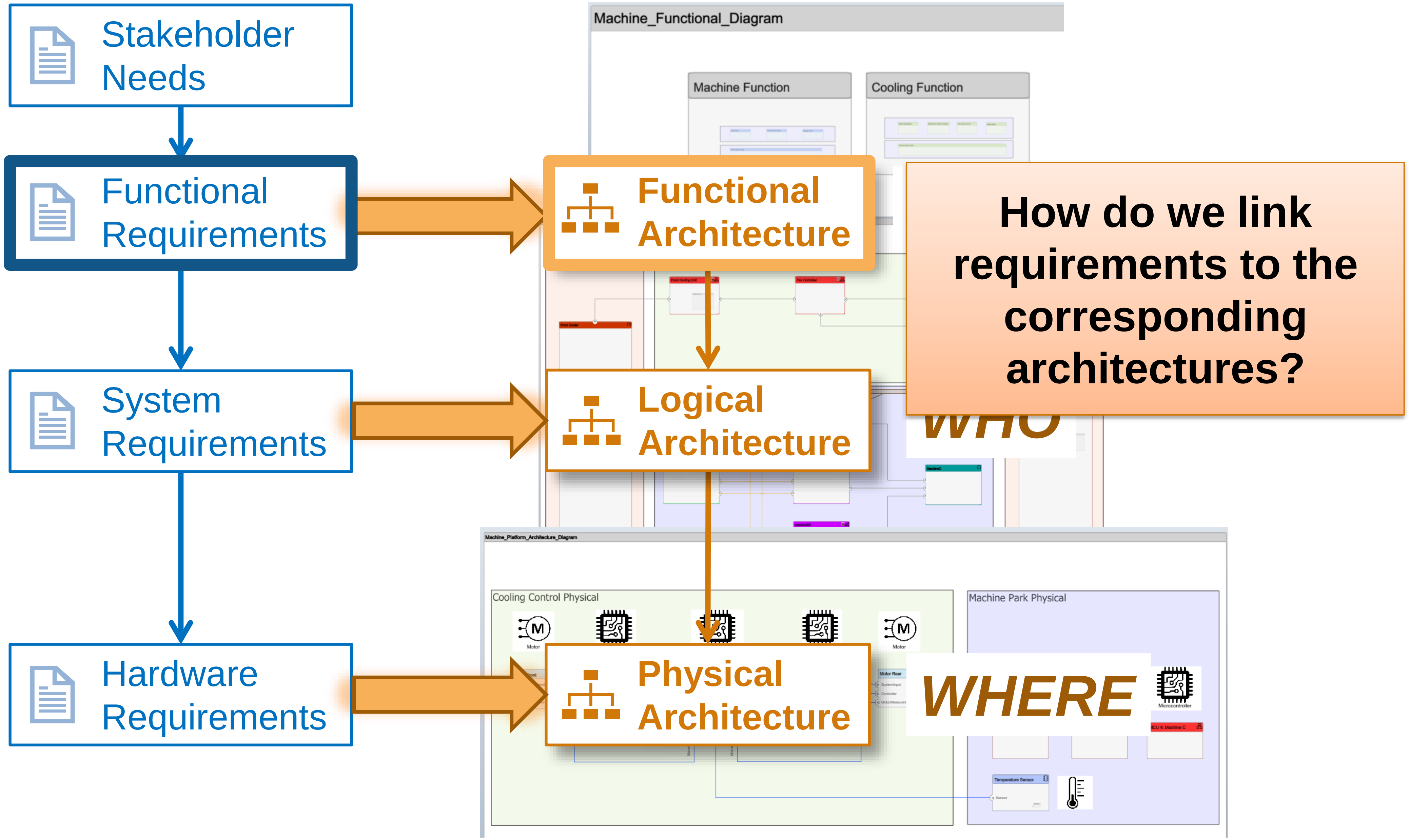
To add a child requirement, right-click a requirement and select **Add Child Requirement**.

To link a requirement to a block in your model, select the block, then right-click the requirement and select **Link from "object name" (object type)**. A link appears in the **Links** pane.

For information on linking using the Requirements Perspective, see Getting Started in the documentation.

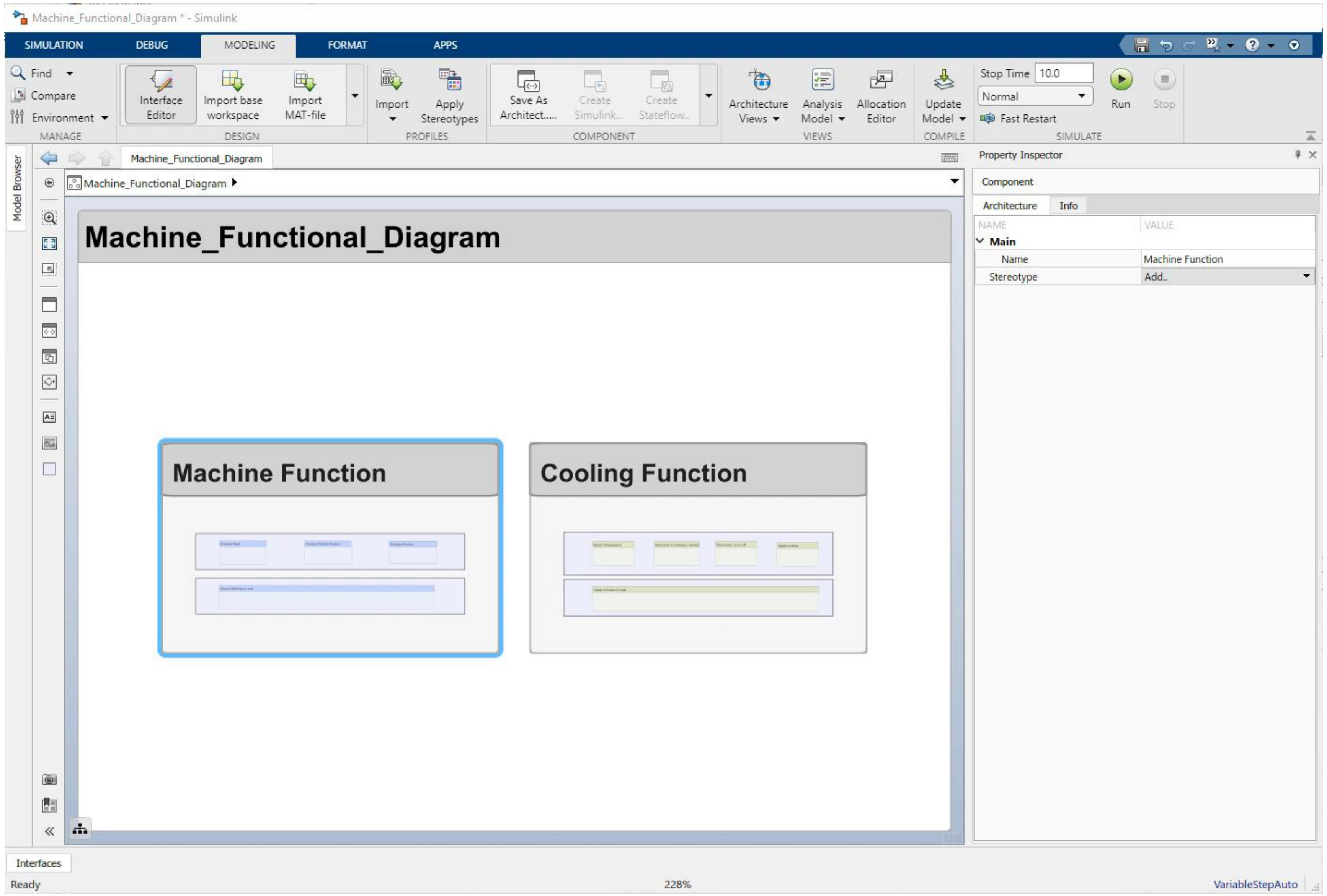
To view the loaded links, click **Show Links** in the toolstrip.

Change the source - destination relationship by selecting a link, and choosing a **Type** from the dropdown list in the **Properties** pane.



How do we link requirements to the corresponding architectures?

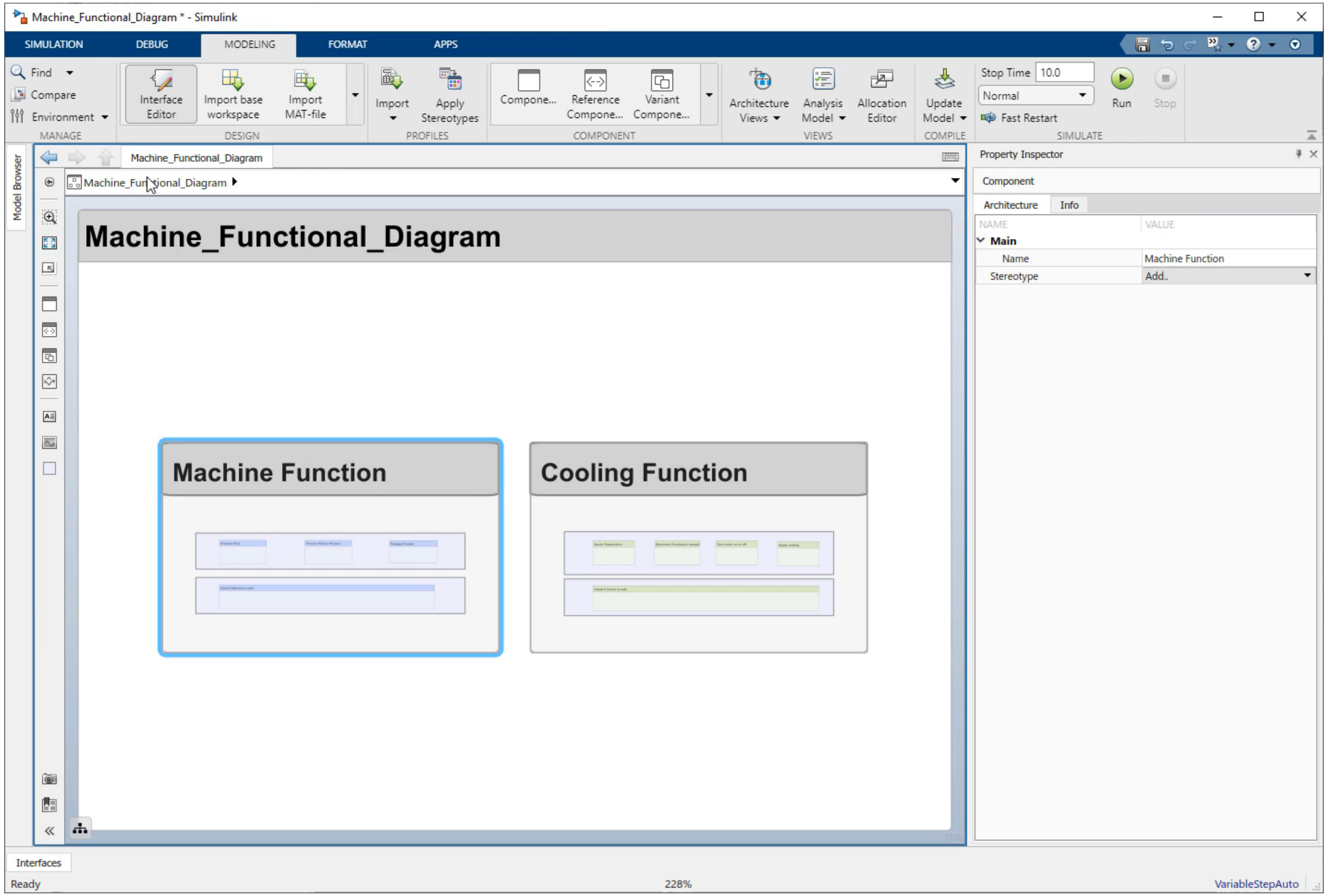
Traceability between Requirement and Architecture



Stop Time: 10.0

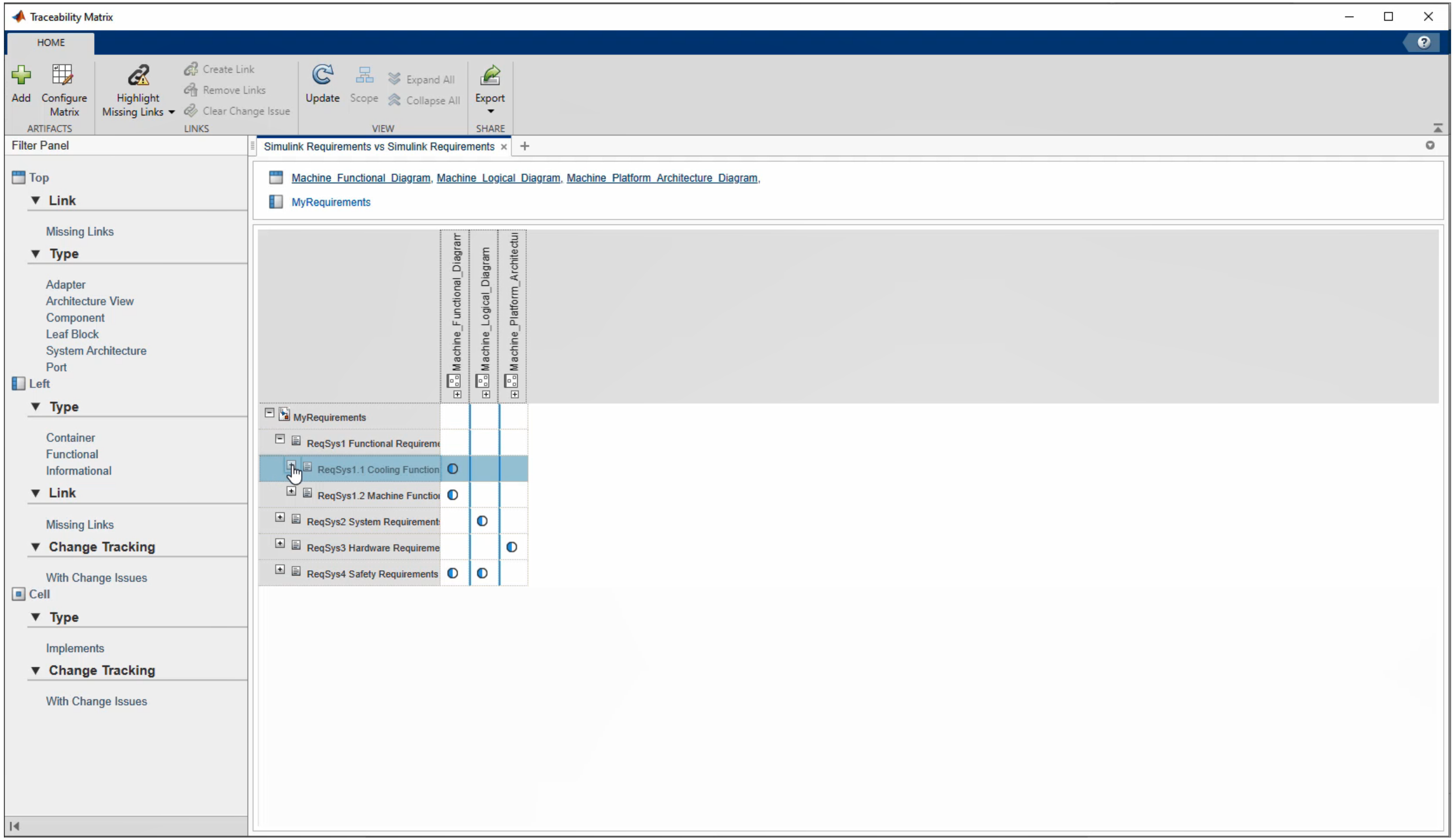
NAME	VALUE
Name	Machine Function
Stereotype	Add..

Hierarchy Diagram



NAME	VALUE
Name	Machine Function
Stereotype	Add..

Traceability Metrics



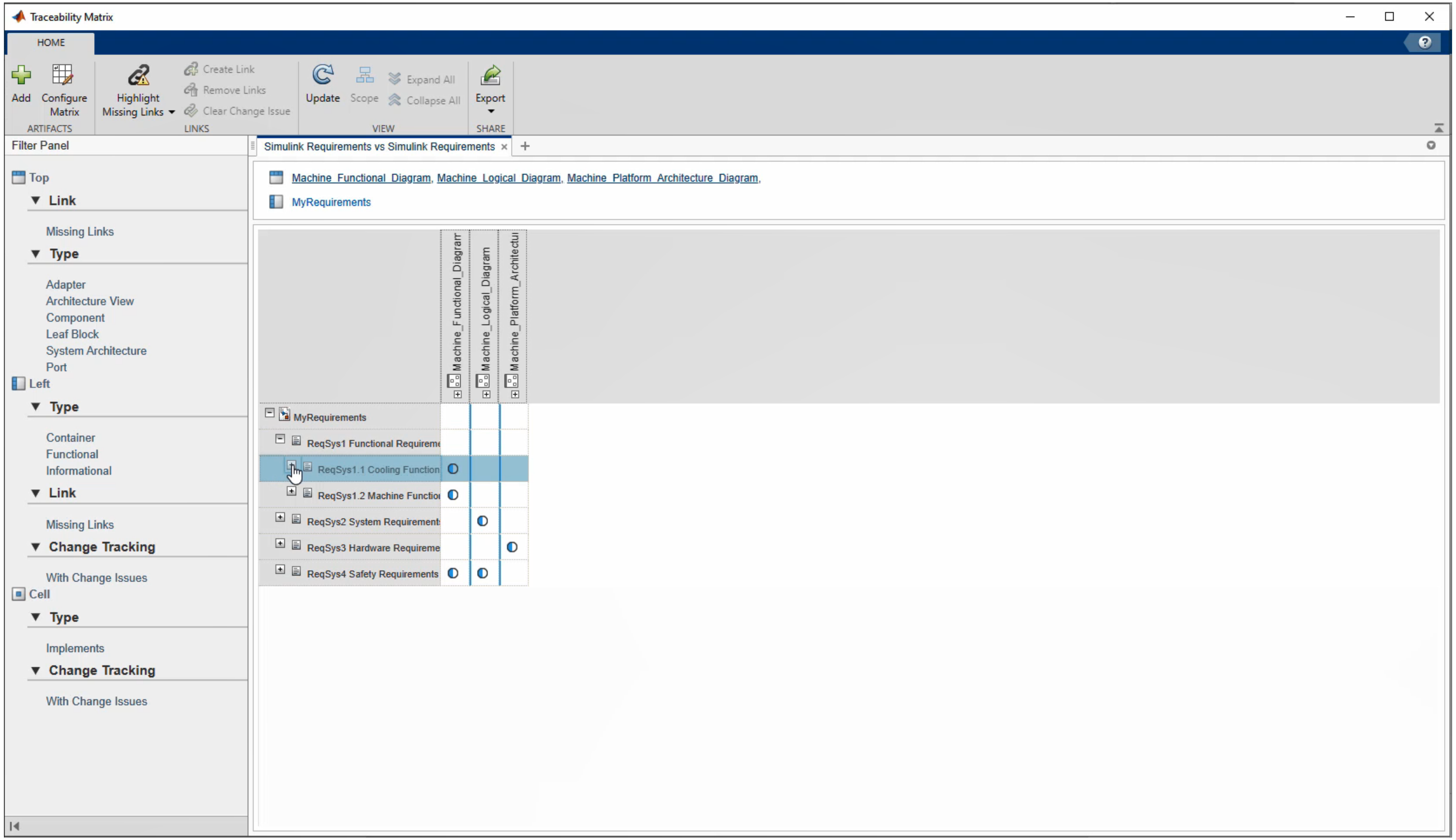
Machine_Functional_Diagram, Machine_Logical_Diagram, Machine_Platform_Architecture_Diagram,

MyRequirements

	Machine_Functional_Diagram	Machine_Logical_Diagram	Machine_Platform_Architectu
MyRequirements			
ReqSys1 Functional Requireme			
ReqSys1.1 Cooling Function	◐		
ReqSys1.2 Machine Functio	◐		
ReqSys2 System Requirement:		◐	
ReqSys3 Hardware Requireme			◐
ReqSys4 Safety Requirements	◐	◐	

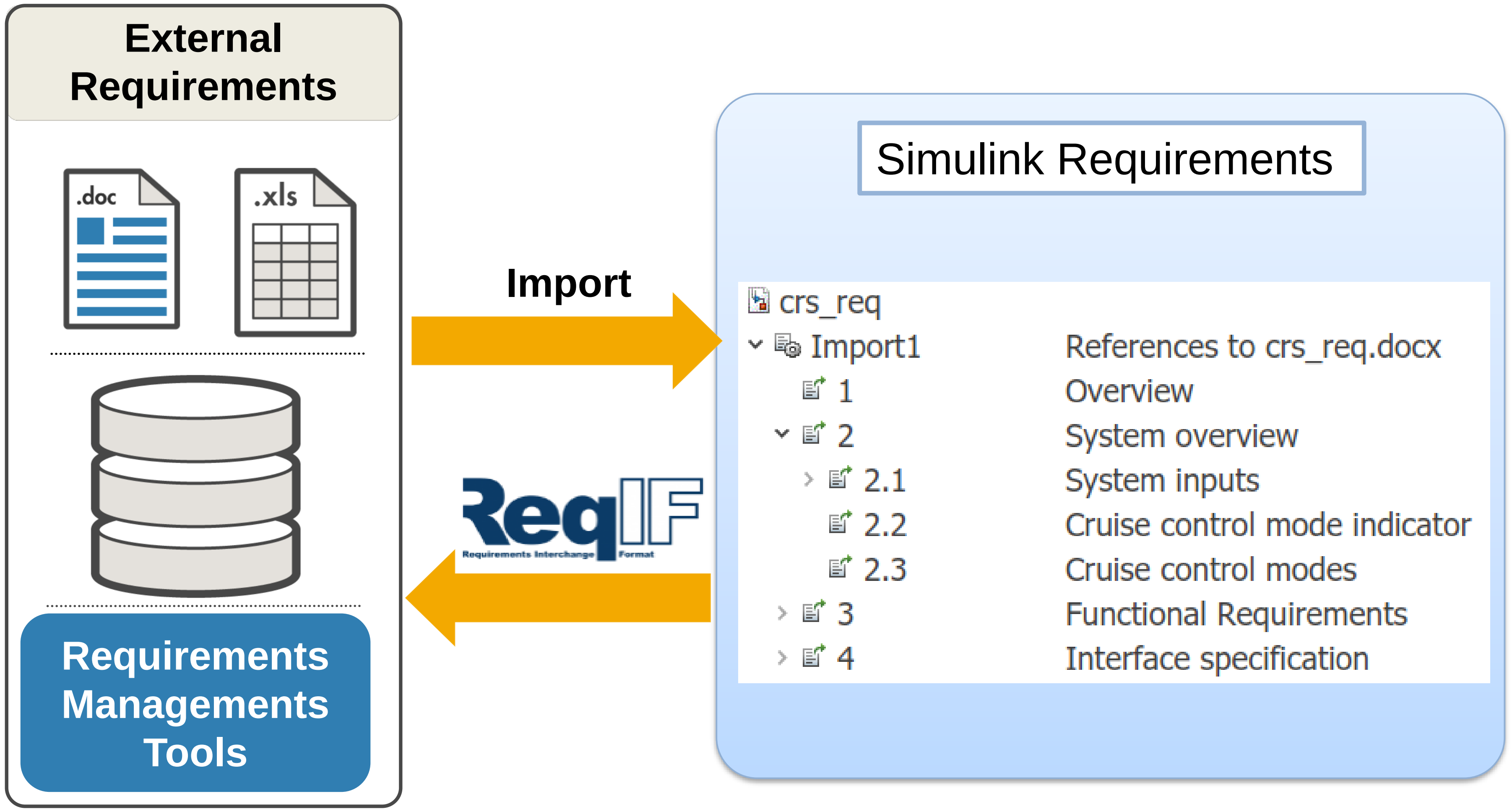
MATLAB EXPO

Traceability Metrics



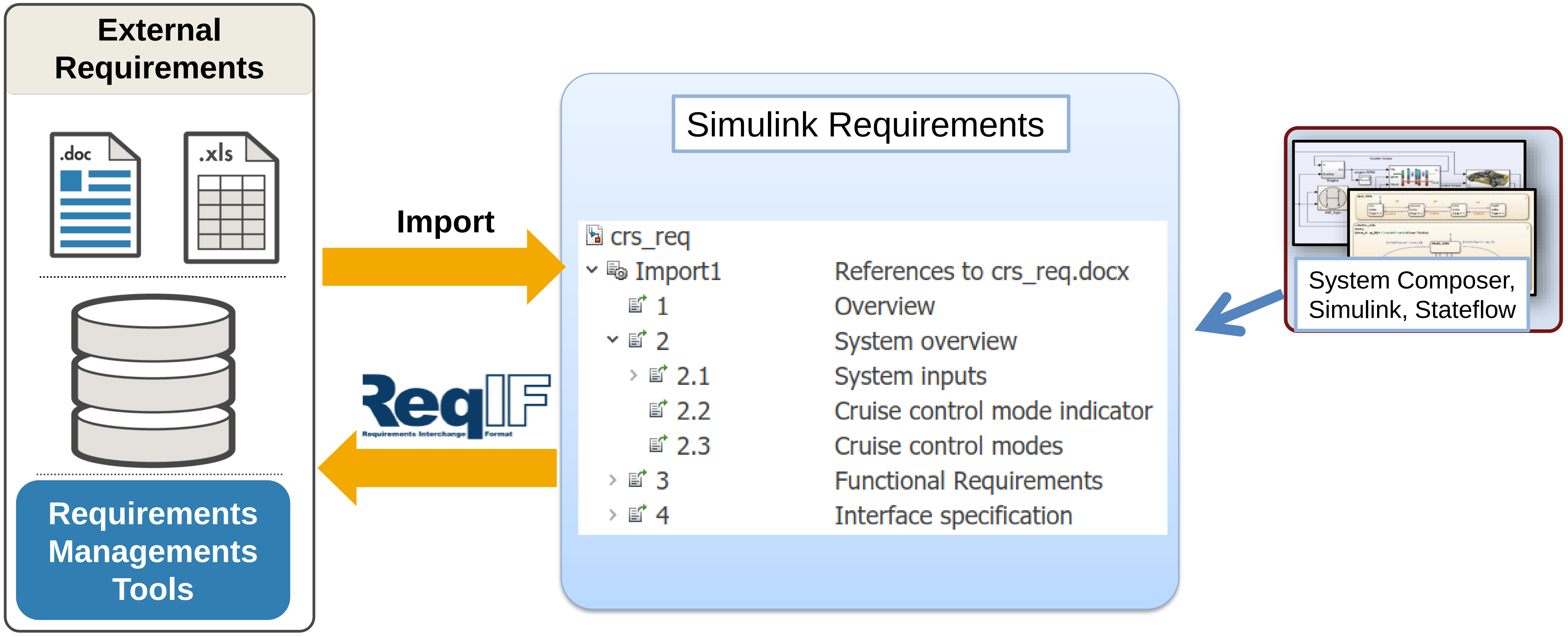
	Machine_Functional_Diagram	Machine_Logical_Diagram	Machine_Platform_Architectu
MyRequirements			
ReqSys1 Functional Requireme			
ReqSys1.1 Cooling Function	◐		
ReqSys1.2 Machine Functio	◐		
ReqSys2 System Requirement		◐	
ReqSys3 Hardware Requireme			◐
ReqSys4 Safety Requirements	◐	◐	

Export Links for Traceability to Model and Test


External Requirements
.doc
.xls
Requirements Managements Tools
Import
ReqIF
Requirements Interchange Format
Simulink Requirements

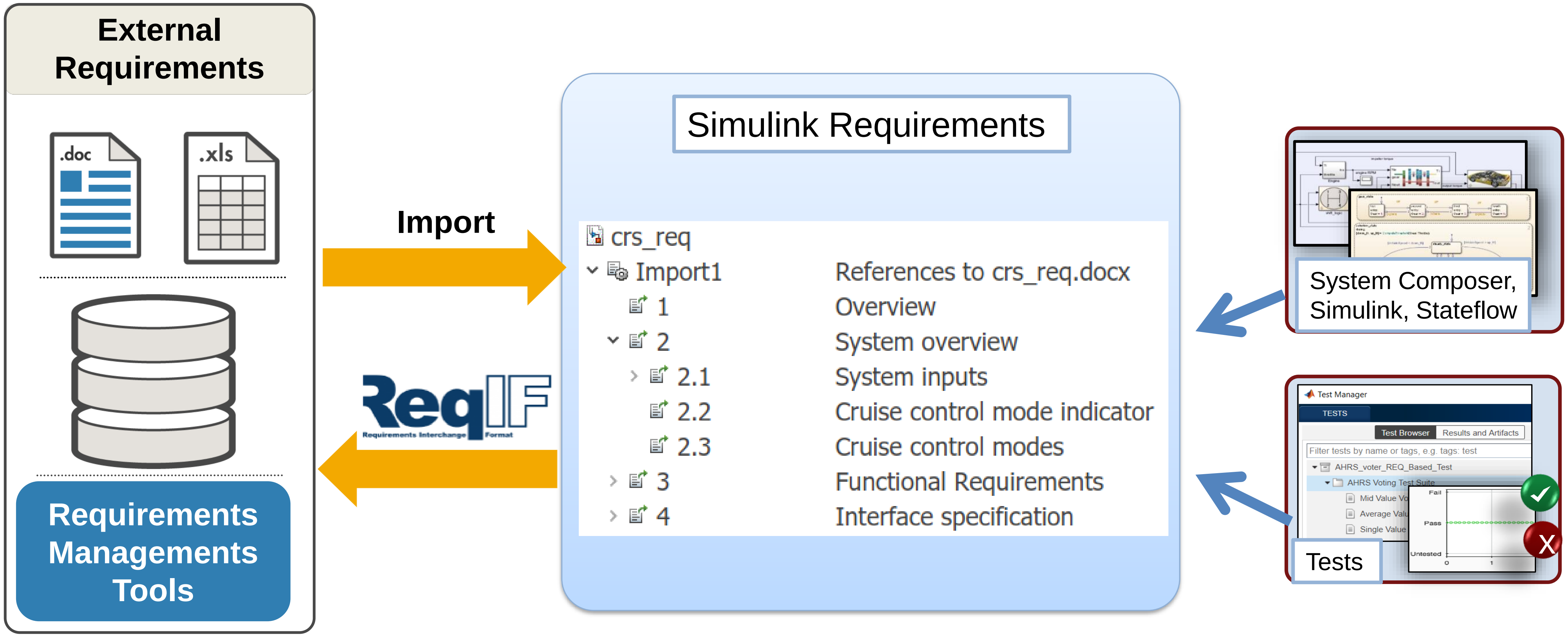
```
crs_req
Import1       References to crs_req.docx
  1           Overview
  2           System overview
    2.1       System inputs
    2.2       Cruise control mode indicator
    2.3       Cruise control modes
  3           Functional Requirements
  4           Interface specification
```

Export Links for Traceability to Model and Test


External Requirements
.doc
.xls
Requirements Managements Tools
Import
ReqIF
Requirements Interchange Format
Simulink Requirements
System Composer, Simulink, Stateflow

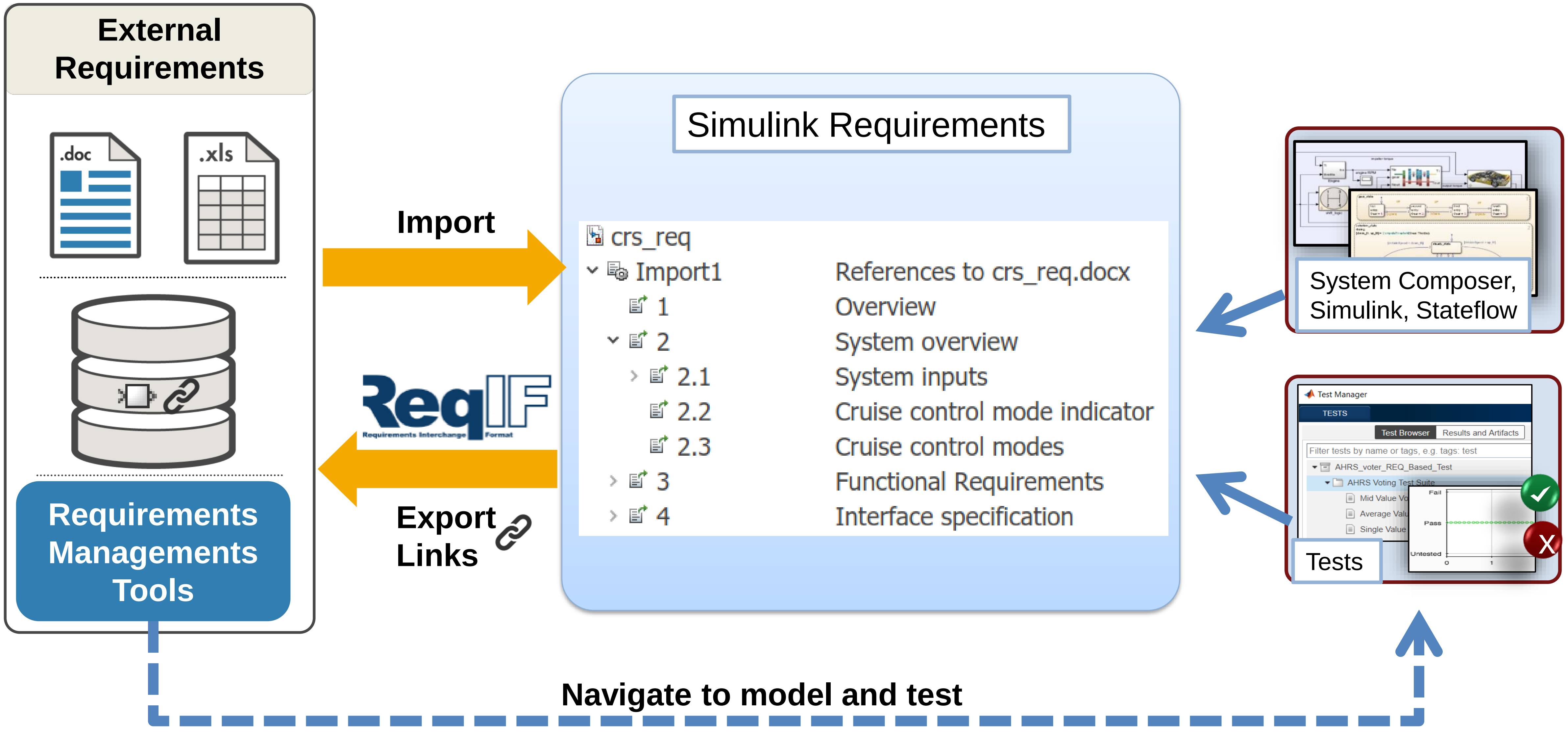
crs_req	
Import1	References to crs_req.docx
1	Overview
2	System overview
2.1	System inputs
2.2	Cruise control mode indicator
2.3	Cruise control modes
3	Functional Requirements
4	Interface specification

Export Links for Traceability to Model and Test

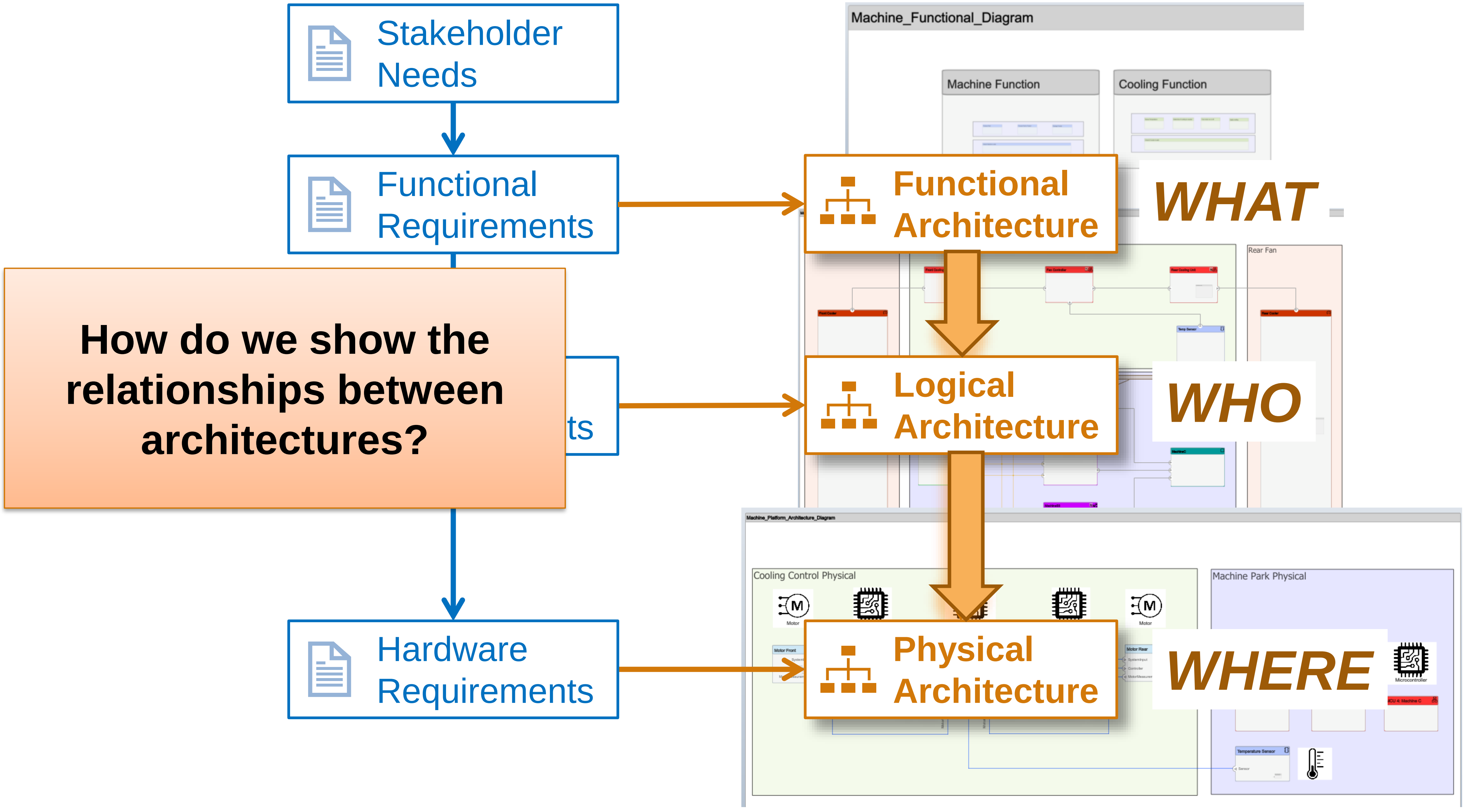


```
crs_req
⌄ Import1          References to crs_req.docx
    1              Overview
  ⌄ 2              System overview
    › 2.1          System inputs
      2.2          Cruise control mode indicator
      2.3          Cruise control modes
  › 3              Functional Requirements
  › 4              Interface specification
```

Export Links for Traceability to Model and Test

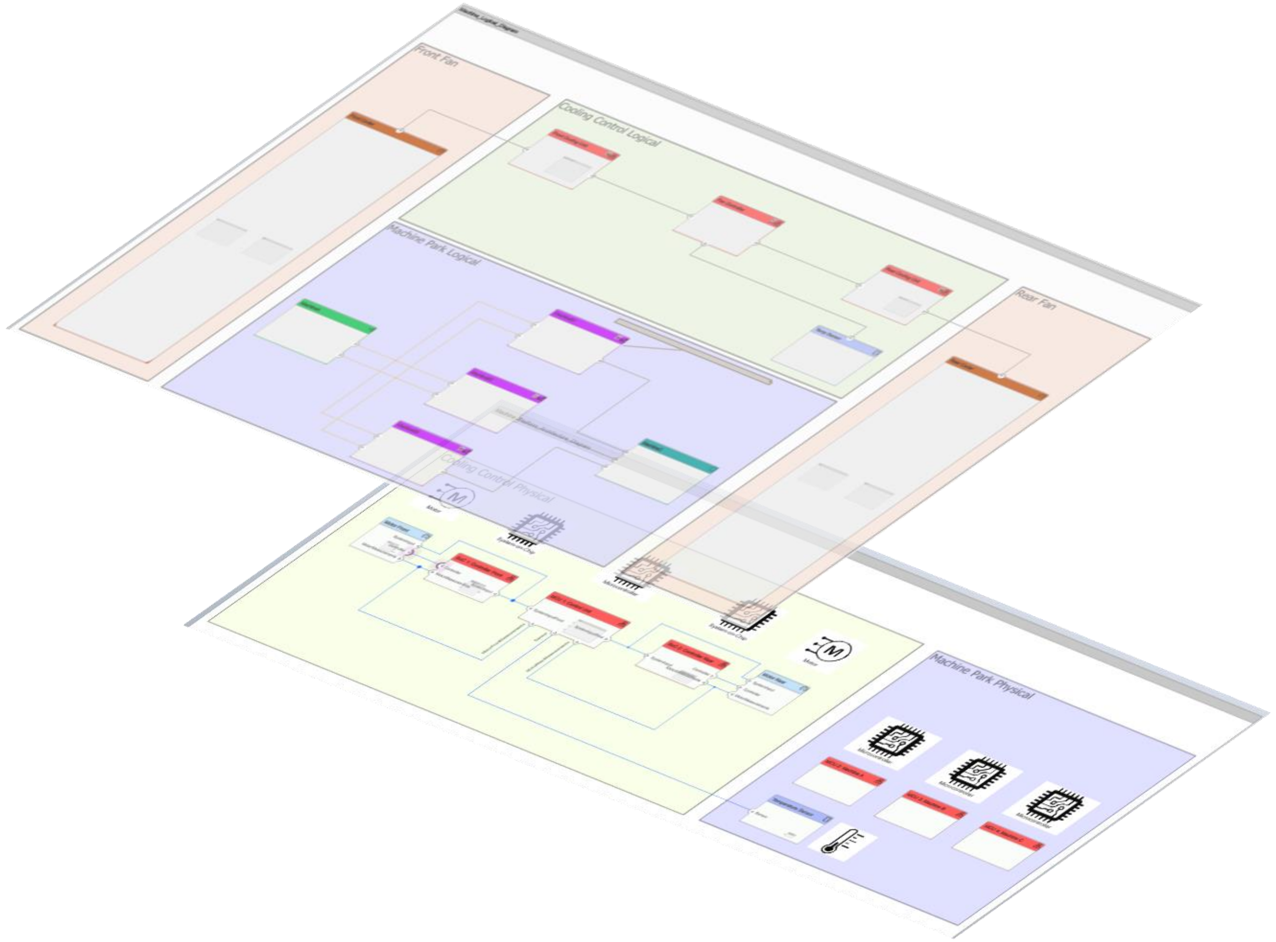

External Requirements
.doc
.xls
Requirements Managements Tools
Import
ReqIF
Requirements Interchange Format
Export Links
Simulink Requirements
System Composer, Simulink, Stateflow
Test Manager
TESTS
Test Browser
Results and Artifacts
Filter tests by name or tags, e.g. tags: test
AHRS_voter_REQ_Based_Test
AHRS Voting Test Suite
Fail
Pass
Untested
Tests
Navigate to model and test

crs_req	
Import1	References to crs_req.docx
1	Overview
2	System overview
2.1	System inputs
2.2	Cruise control mode indicator
2.3	Cruise control modes
3	Functional Requirements
4	Interface specification

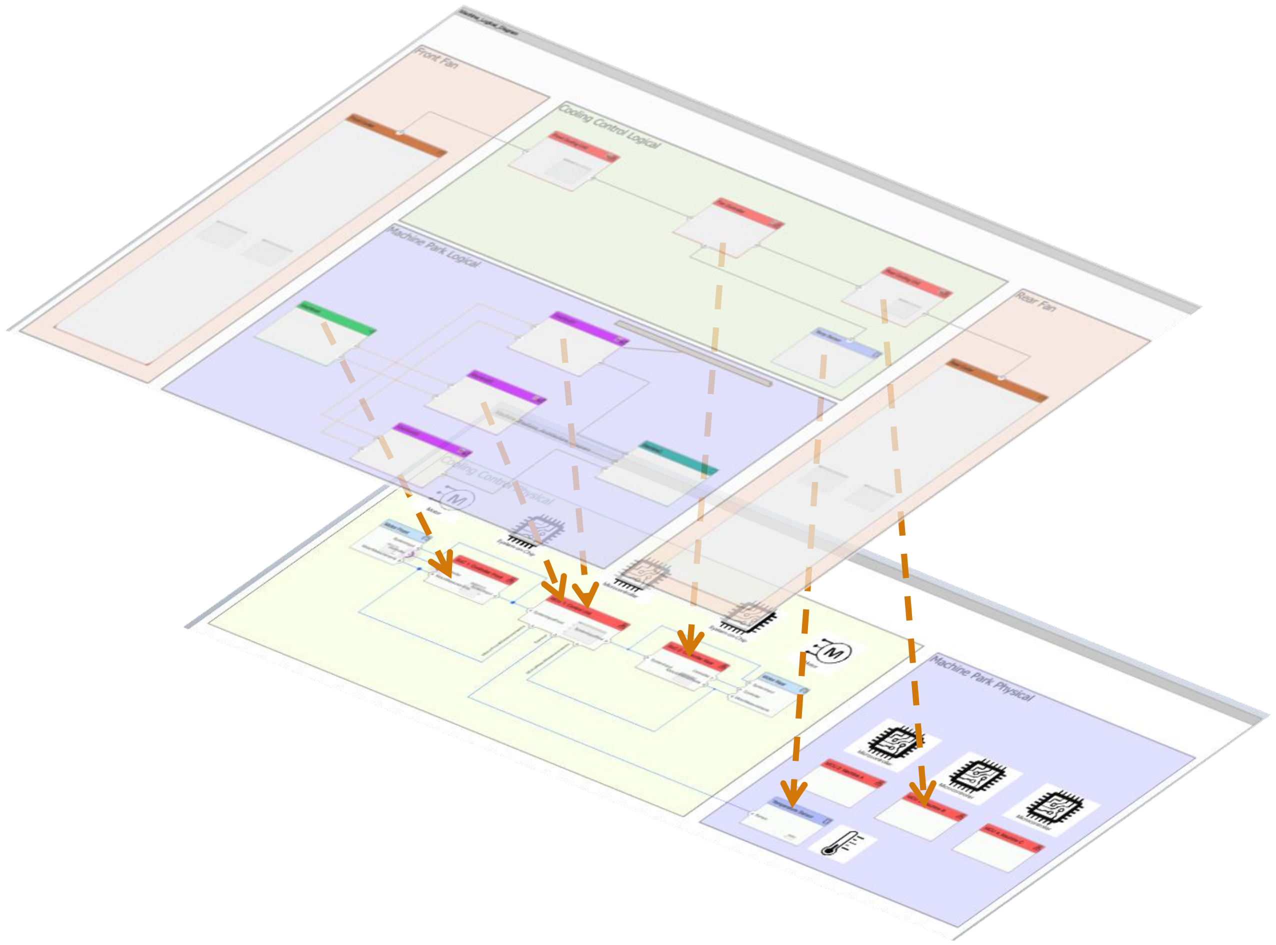


How do we show the relationships between architectures?

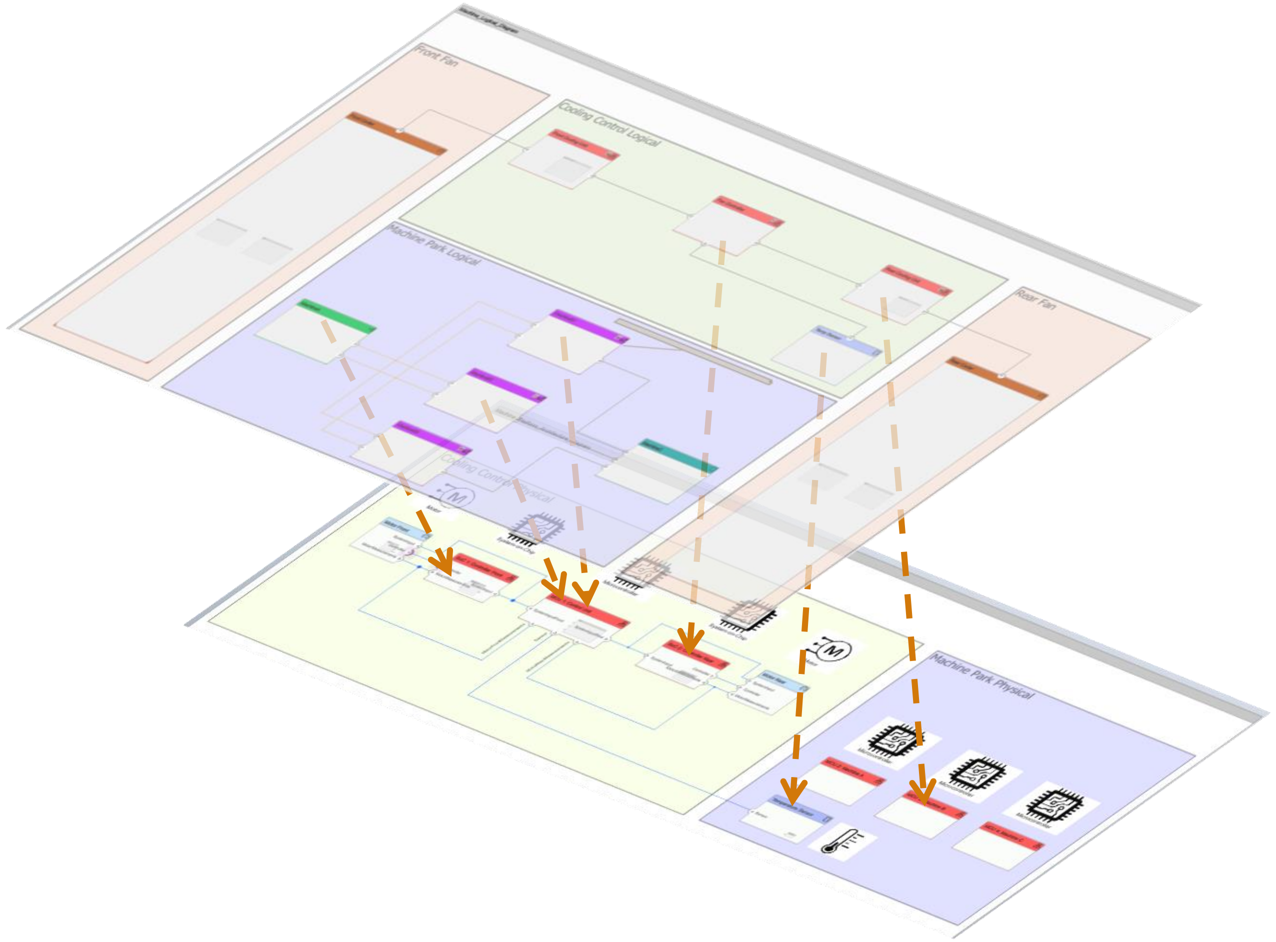
Allocating between Architectures

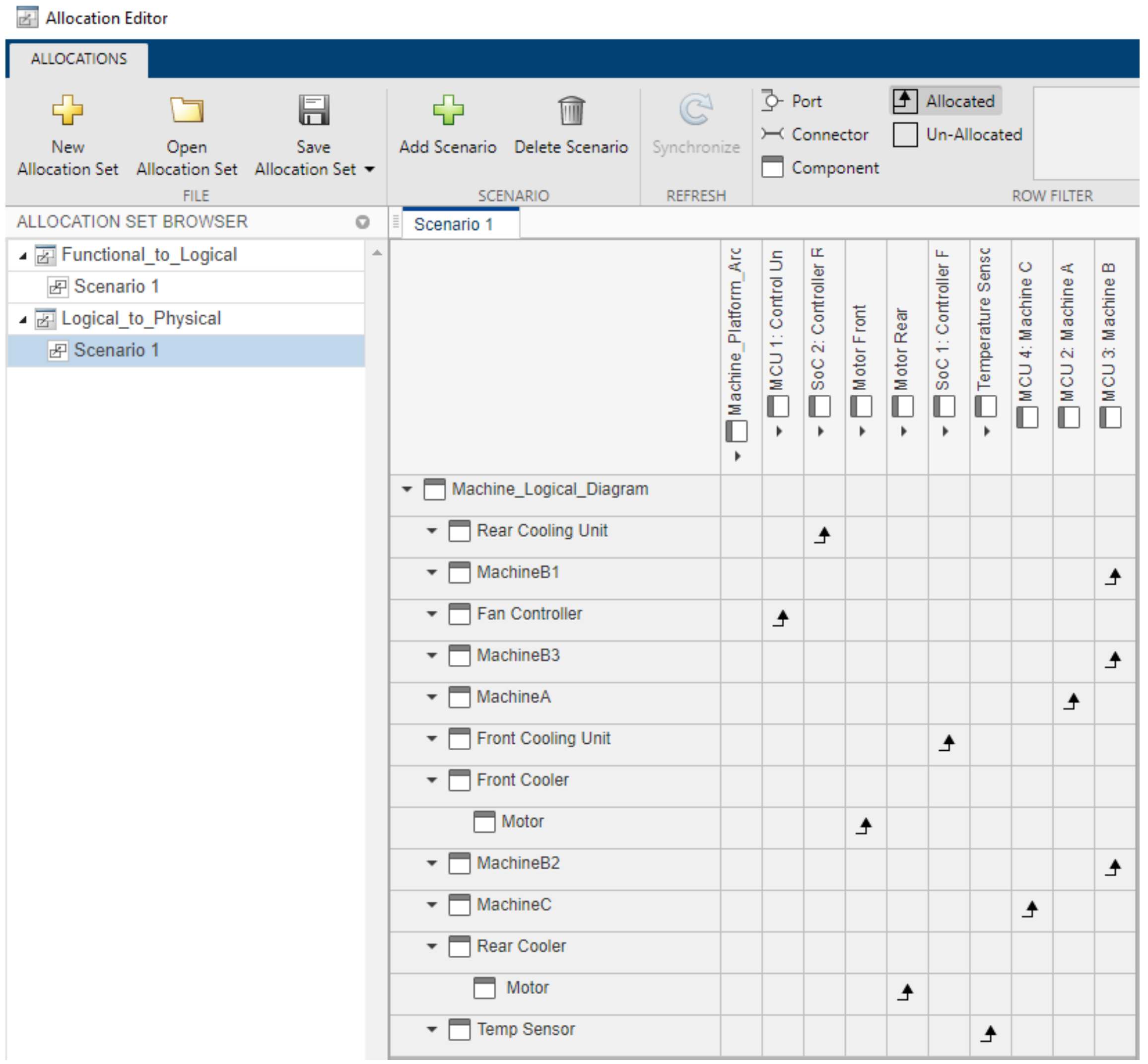


Allocating between Architectures



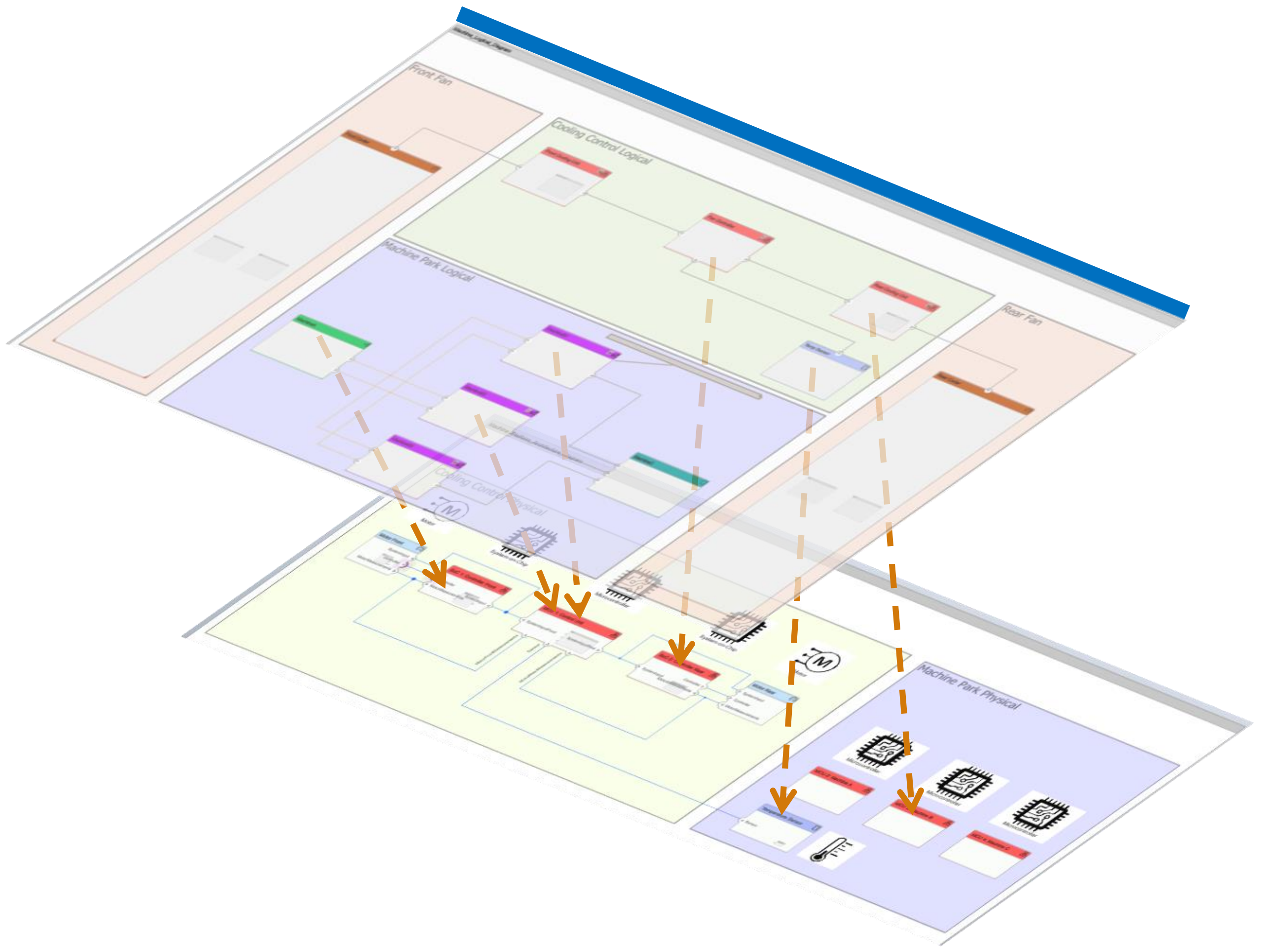
Allocating between Architectures

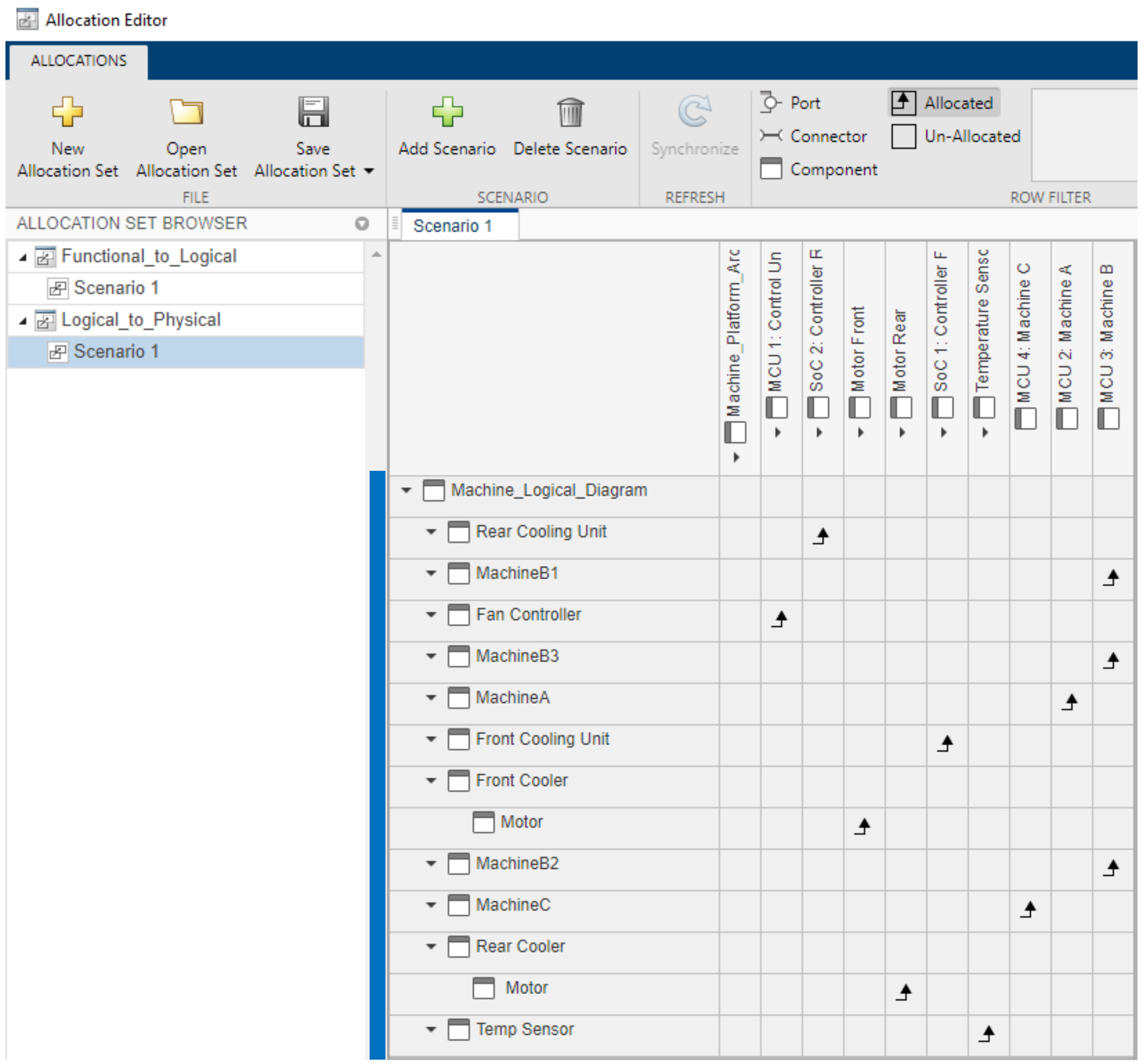




	Machine_Platform_Arc	MCU 1: Control Un	SoC 2: Controller F	Motor Front	Motor Rear	SoC 1: Controller F	Temperature Sensc	MCU 4: Machine C	MCU 2: Machine A	MCU 3: Machine B
Machine_Logical_Diagram										
Rear Cooling Unit			↥							
MachineB1										↥
Fan Controller		↥								
MachineB3										↥
MachineA									↥	
Front Cooling Unit						↥				
Front Cooler										
Motor				↥						
MachineB2										↥
MachineC								↥		
Rear Cooler										
Motor					↥					
Temp Sensor							↥			

Allocating between Architectures





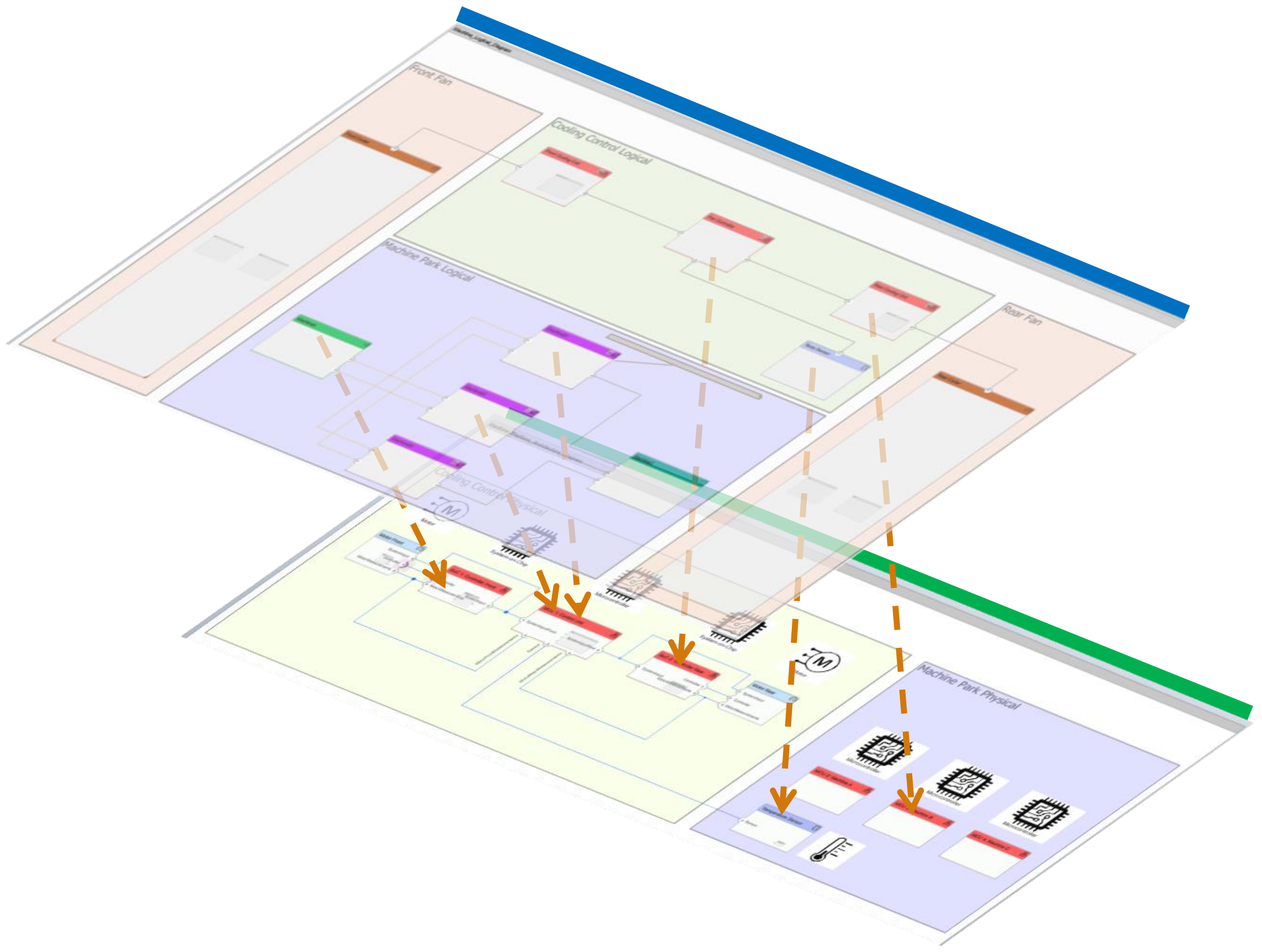
ALLOCATION SET BROWSER

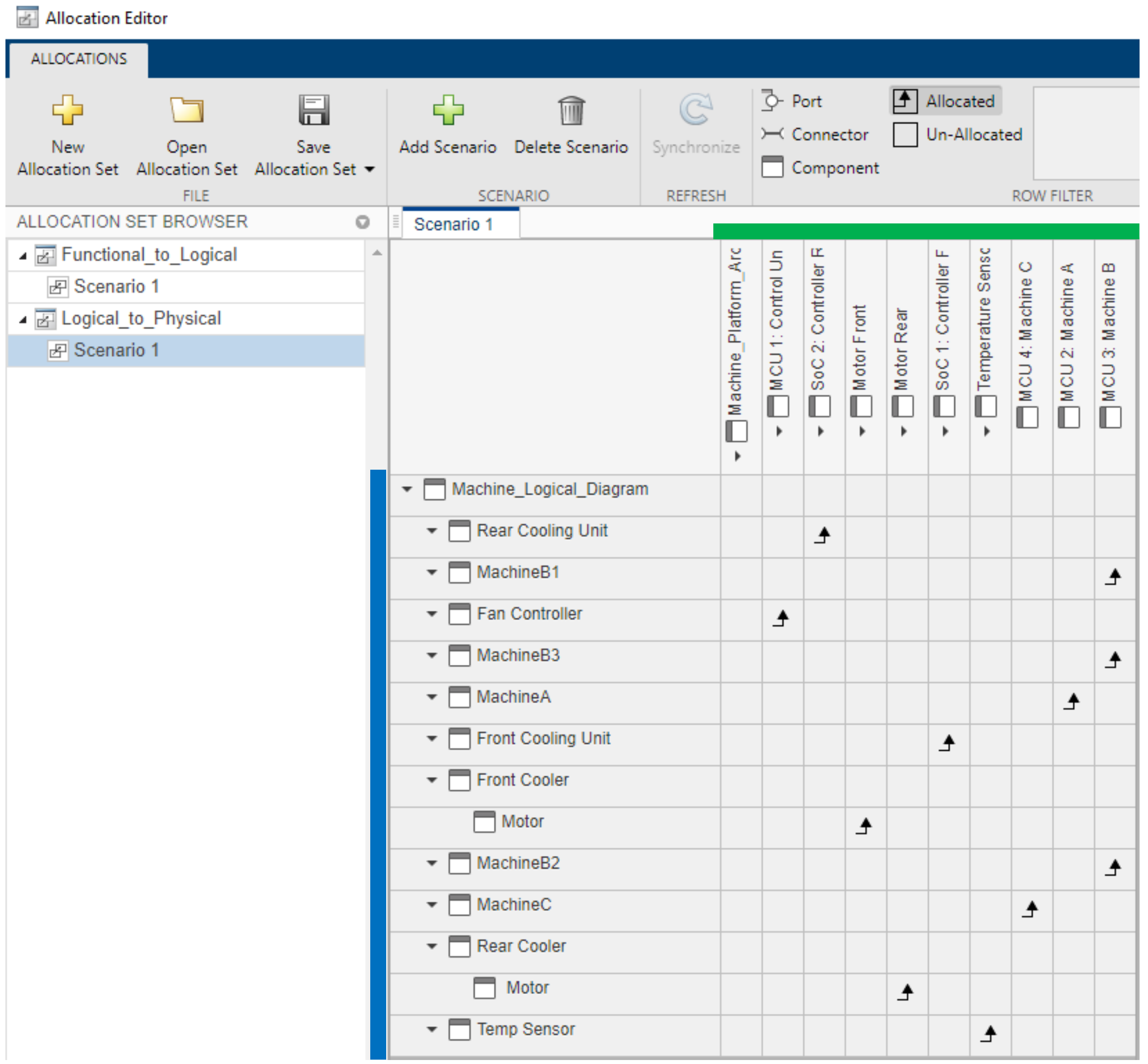
- Functional_to_Logical
  - Scenario 1
- Logical_to_Physical
  - Scenario 1

Scenario 1

	Machine_Platform_Arc	MCU 1: Control Un	SoC 2: Controller F	Motor Front	Motor Rear	SoC 1: Controller F	Temperature Sensc	MCU 4: Machine C	MCU 2: Machine A	MCU 3: Machine B
Machine_Logical_Diagram										
Rear Cooling Unit			↑							
MachineB1										↑
Fan Controller		↑								
MachineB3										↑
MachineA									↑	
Front Cooling Unit						↑				
Front Cooler										
Motor				↑						
MachineB2										↑
MachineC								↑		
Rear Cooler										
Motor					↑					
Temp Sensor							↑			

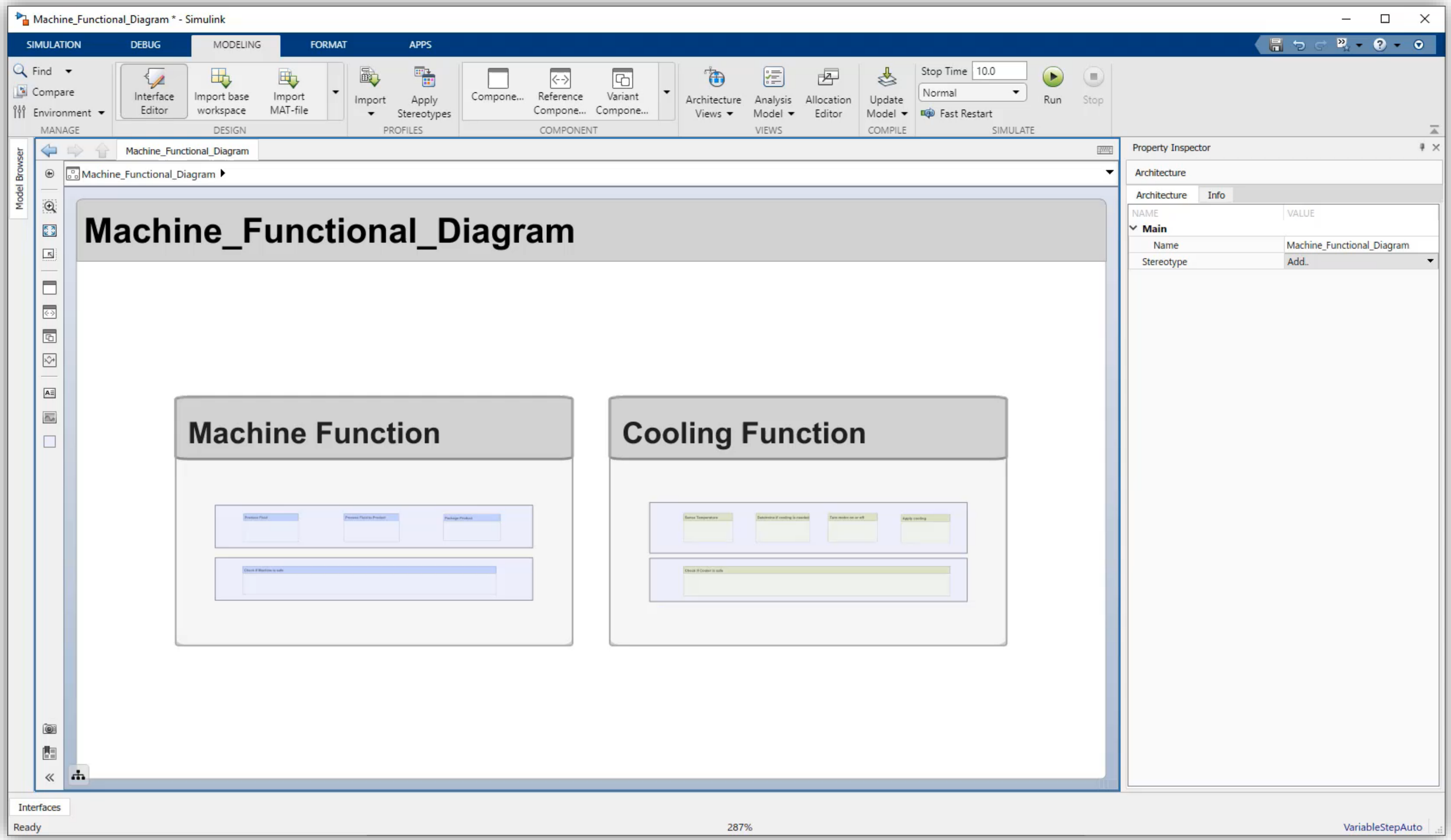
Allocating between Architectures





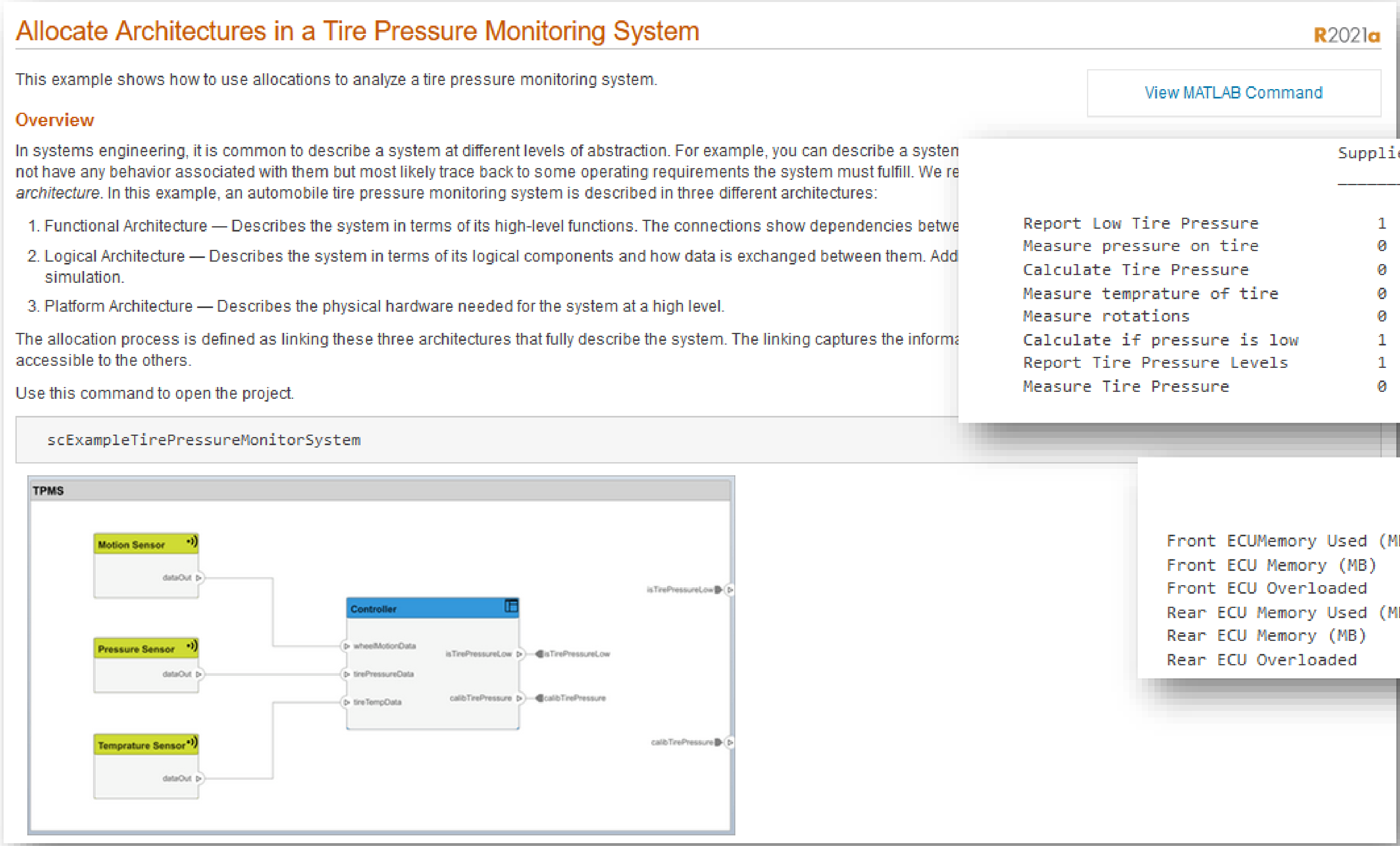
- Functional_to_Logical
  - Scenario 1
- Logical_to_Physical
  - Scenario 1

	Machine_Platform_Arc	MCU 1: Control Un	SoC 2: Controller F	Motor Front	Motor Rear	SoC 1: Controller F	Temperature Sensc	MCU 4: Machine C	MCU 2: Machine A	MCU 3: Machine B
Machine_Logical_Diagram										
Rear Cooling Unit			↗							
MachineB1										↗
Fan Controller		↗								
MachineB3										↗
MachineA									↗	
Front Cooling Unit						↗				
Front Cooler										
Motor				↗						
MachineB2										↗
MachineC								↗		
Rear Cooler										
Motor					↗					
Temp Sensor							↗			



NAME	VALUE
Main	
Name	Machine_Functional_Diagram
Stereotype	Add..

Assess Different Allocation Scenarios Quantitatively



Allocate Architectures in a Tire Pressure Monitoring System

R2021a

This example shows how to use allocations to analyze a tire pressure monitoring system.

View MATLAB Command

Overview

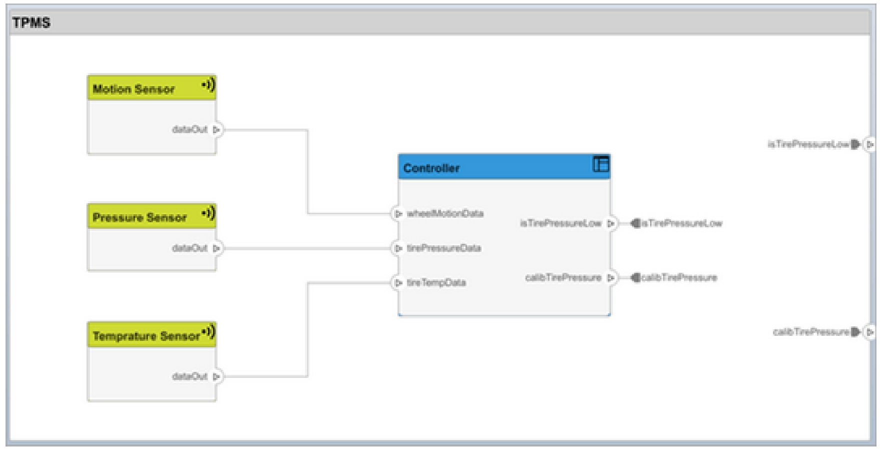
In systems engineering, it is common to describe a system at different levels of abstraction. For example, you can describe a systen
not have any behavior associated with them but most likely trace back to some operating requirements the system must fulfill. We re
architecture. In this example, an automobile tire pressure monitoring system is described in three different architectures:

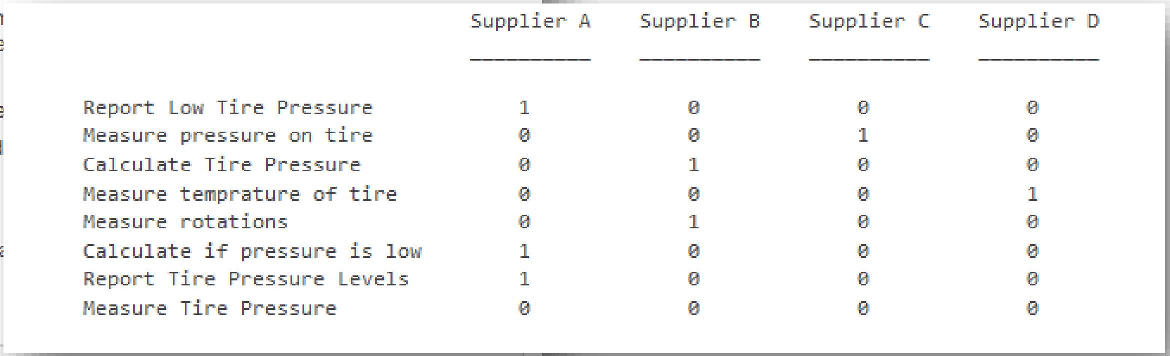
1. Functional Architecture — Describes the system in terms of its high-level functions. The connections show dependencies betwe
2. Logical Architecture — Describes the system in terms of its logical components and how data is exchanged between them. Add
   simulation.
3. Platform Architecture — Describes the physical hardware needed for the system at a high level.

The allocation process is defined as linking these three architectures that fully describe the system. The linking captures the informa
accessible to the others.

Use this command to open the project.

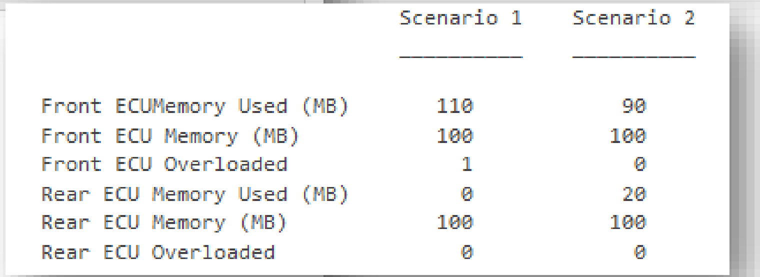
```
scExampleTirePressureMonitorSystem
```





```
                                 Supplier A    Supplier B    Supplier C    Supplier D
                                 __________    __________    __________    __________

    Report Low Tire Pressure         1             0             0             0
    Measure pressure on tire         0             0             1             0
    Calculate Tire Pressure          0             1             0             0
    Measure temprature of tire       0             0             0             1
    Measure rotations                0             1             0             0
    Calculate if pressure is low     1             0             0             0
    Report Tire Pressure Levels      1             0             0             0
    Measure Tire Pressure            0             0             0             0
```



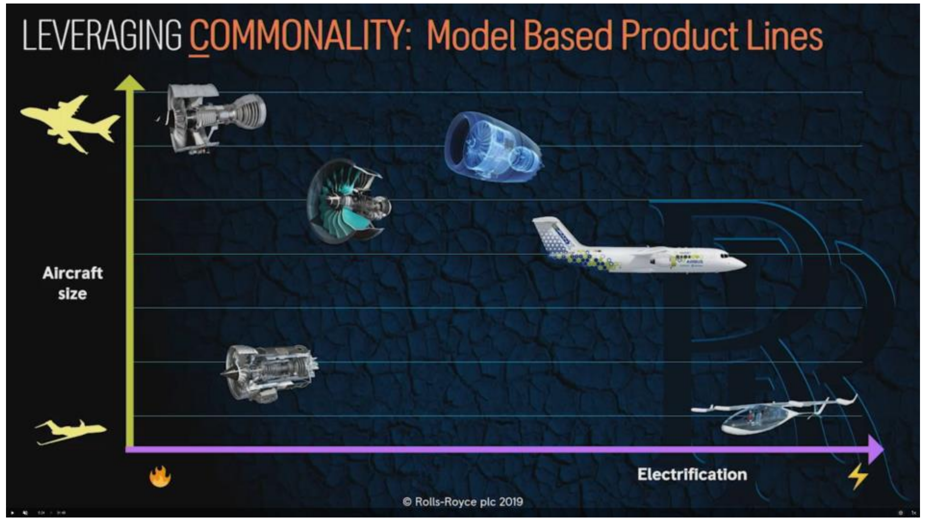
```
                                 Scenario 1    Scenario 2
                                 __________    __________

    Front ECUMemory Used (MB)       110            90
    Front ECU Memory (MB)           100           100
    Front ECU Overloaded              1             0
    Rear ECU Memory Used (MB)         0            20
    Rear ECU Memory (MB)            100           100
    Rear ECU Overloaded               0             0
```

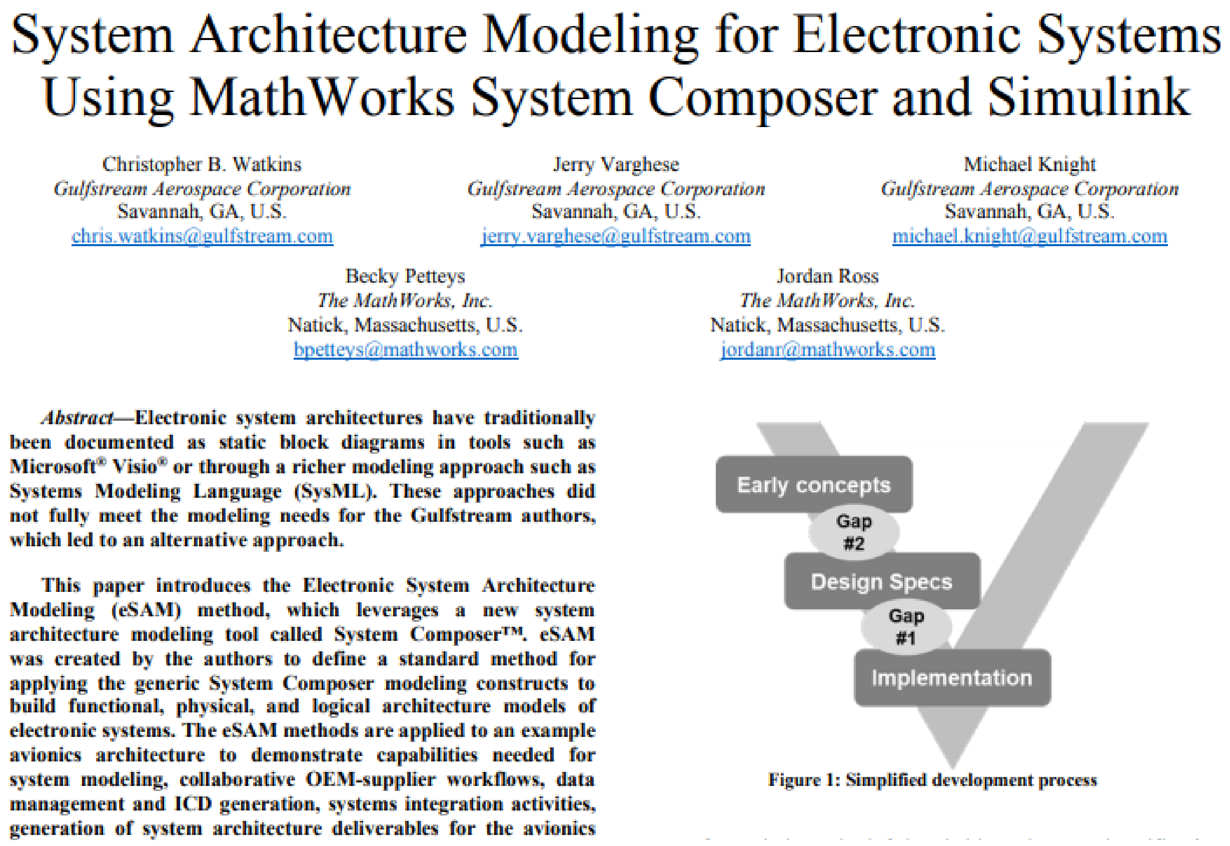
https://www.mathworks.com/help/systemcomposer/ug/allocate-architectures-of-the-tire-pressure-monitoring-system.html

Who is doing Model Based Systems Engineering with MathWorks tools?

Rolls Royce, UK Expo, Oct 2019



https://www.mathworks.com/videos/our-journey-towards-model-based-product-lines-1573233985120.html



System Architecture Modeling for Electronic Systems Using MathWorks System Composer and Simulink

Christopher B. Watkins
Gulfstream Aerospace Corporation
Savannah, GA, U.S.
chris.watkins@gulfstream.com

Jerry Varghese
Gulfstream Aerospace Corporation
Savannah, GA, U.S.
jerry.varghese@gulfstream.com

Michael Knight
Gulfstream Aerospace Corporation
Savannah, GA, U.S.
michael.knight@gulfstream.com

Becky Petteys
The MathWorks, Inc.
Natick, Massachusetts, U.S.
bpetteys@mathworks.com

Jordan Ross
The MathWorks, Inc.
Natick, Massachusetts, U.S.
jordanr@mathworks.com

***Abstract*—Electronic system architectures have traditionally been documented as static block diagrams in tools such as Microsoft® Visio® or through a richer modeling approach such as Systems Modeling Language (SysML). These approaches did not fully meet the modeling needs for the Gulfstream authors, which led to an alternative approach.**

This paper introduces the Electronic System Architecture Modeling (eSAM) method, which leverages a new system architecture modeling tool called System Composer™. eSAM was created by the authors to define a standard method for applying the generic System Composer modeling constructs to build functional, physical, and logical architecture models of electronic systems. The eSAM methods are applied to an example avionics architecture to demonstrate capabilities needed for system modeling, collaborative OEM-supplier workflows, data management and ICD generation, systems integration activities, generation of system architecture deliverables for the avionics

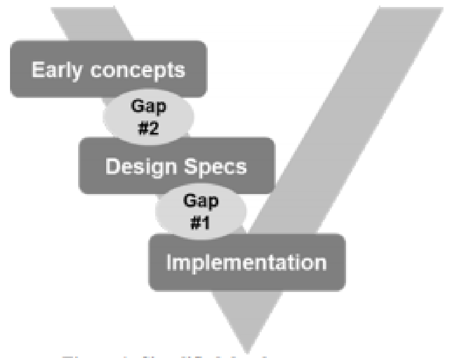


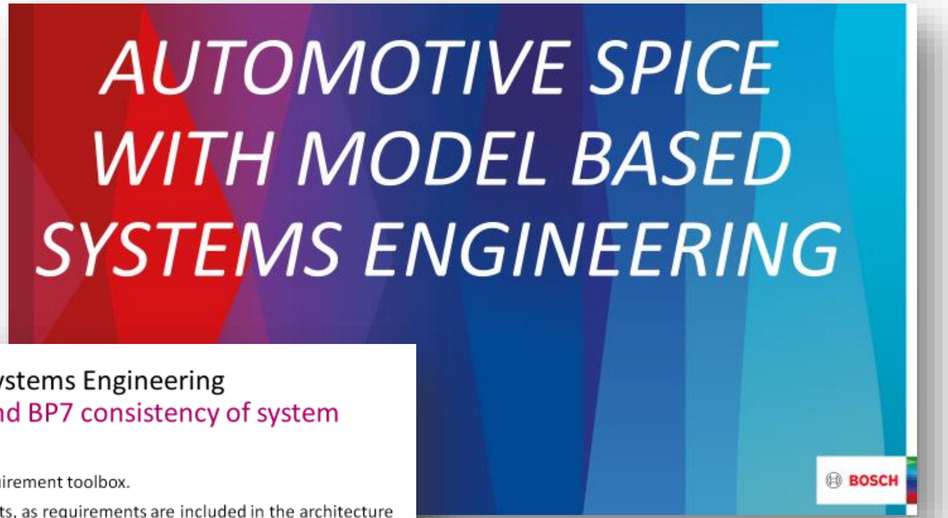
Figure 1: Simplified development process

https://ieeexplore.ieee.org/document/9256753

Who is doing Model Based Systems Engineering with MathWorks tools?

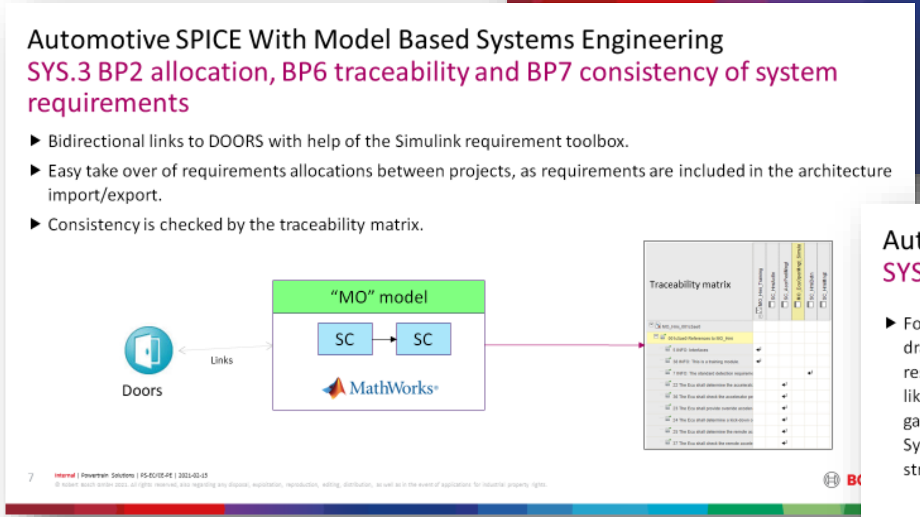
MathWorks Automotive Conference 2021

Felix Raab, Bosch



AUTOMOTIVE SPICE WITH MODEL BASED SYSTEMS ENGINEERING

BOSCH

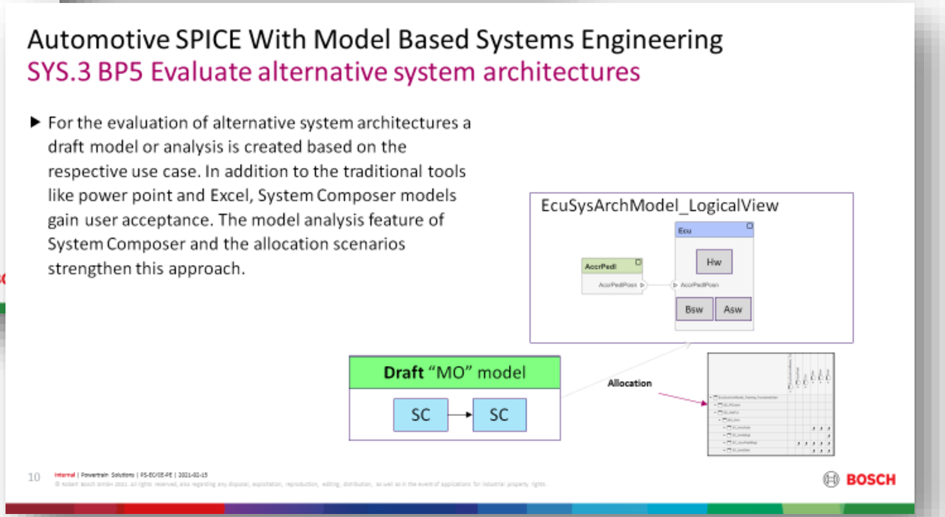


Automotive SPICE With Model Based Systems Engineering

SYS.3 BP2 allocation, BP6 traceability and BP7 consistency of system requirements

- Bidirectional links to DOORS with help of the Simulink requirement toolbox.
- Easy take over of requirements allocations between projects, as requirements are included in the architecture import/export.
- Consistency is checked by the traceability matrix.

7 Internal | Powertrain Solutions | PS-EC/EE-PE | 2021-02-15



Automotive SPICE With Model Based Systems Engineering

SYS.3 BP5 Evaluate alternative system architectures

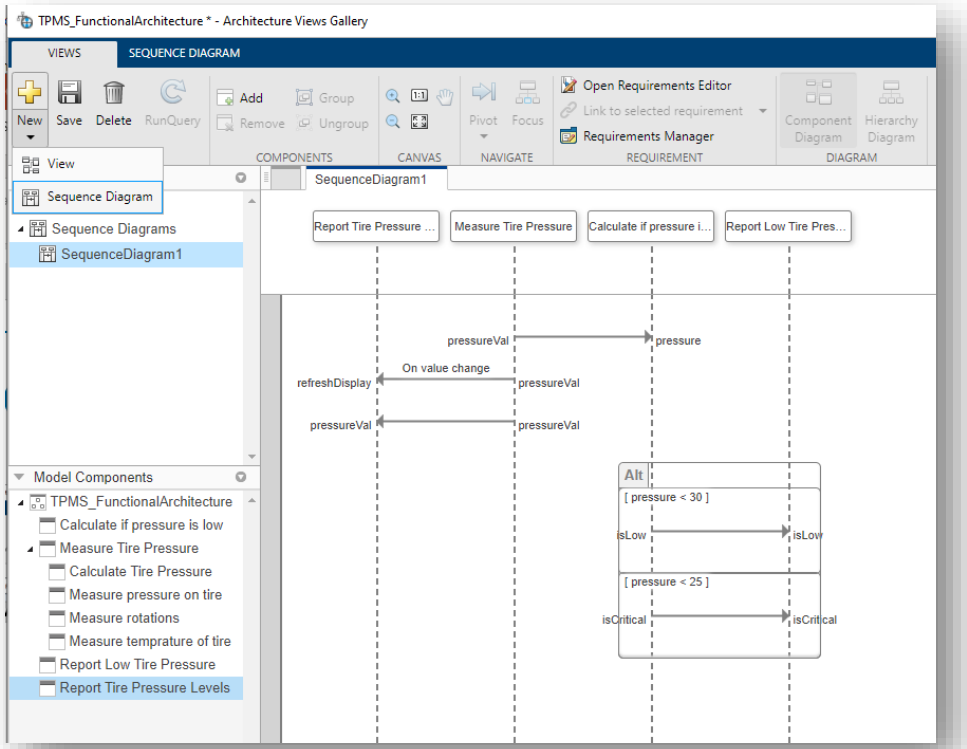
- For the evaluation of alternative system architectures a draft model or analysis is created based on the respective use case. In addition to the traditional tools like power point and Excel, System Composer models gain user acceptance. The model analysis feature of System Composer and the allocation scenarios strengthen this approach.

10 Internal | Powertrain Solutions | PS-EC/EE-PE | 2021-02-15

BOSCH

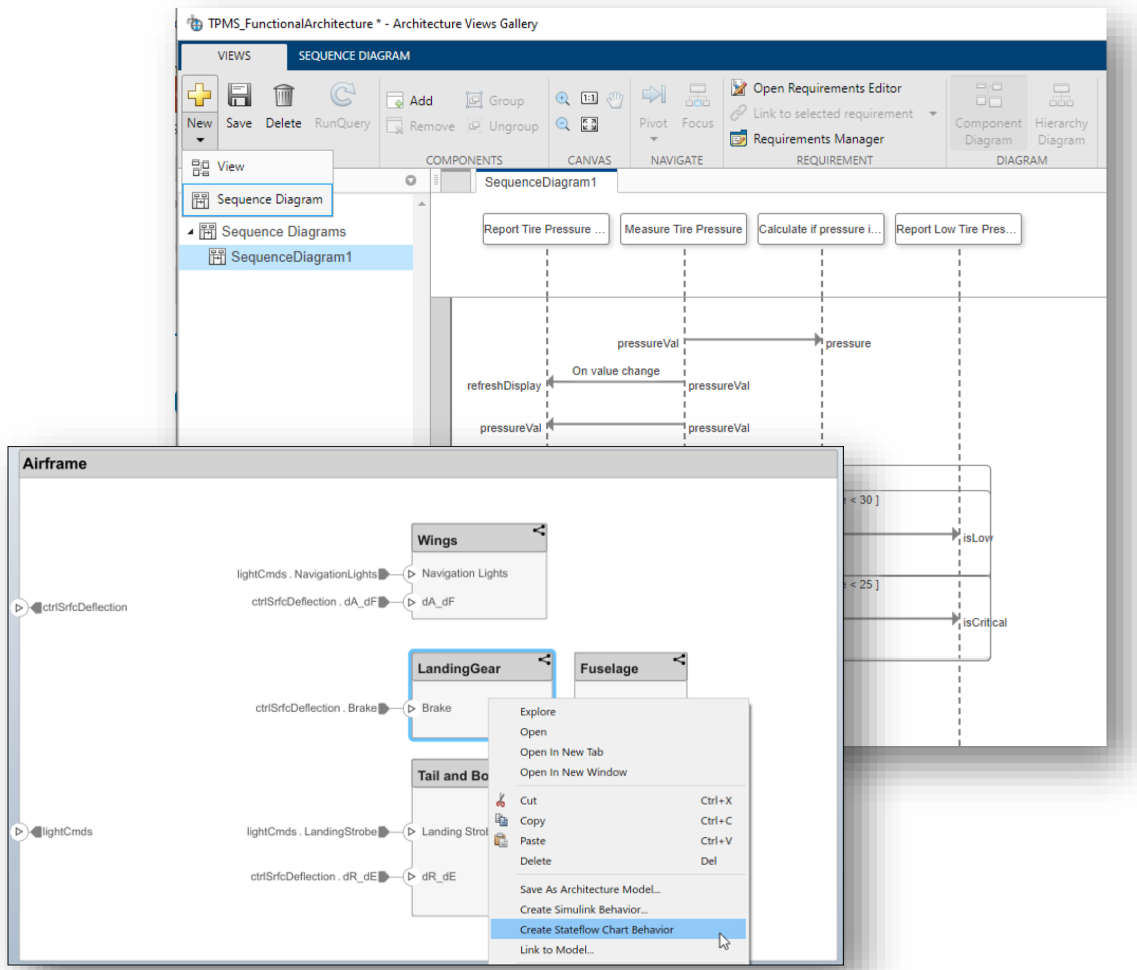
New Features in R2021a

- System Composer
  - Sequence diagrams



New Features in R2021a

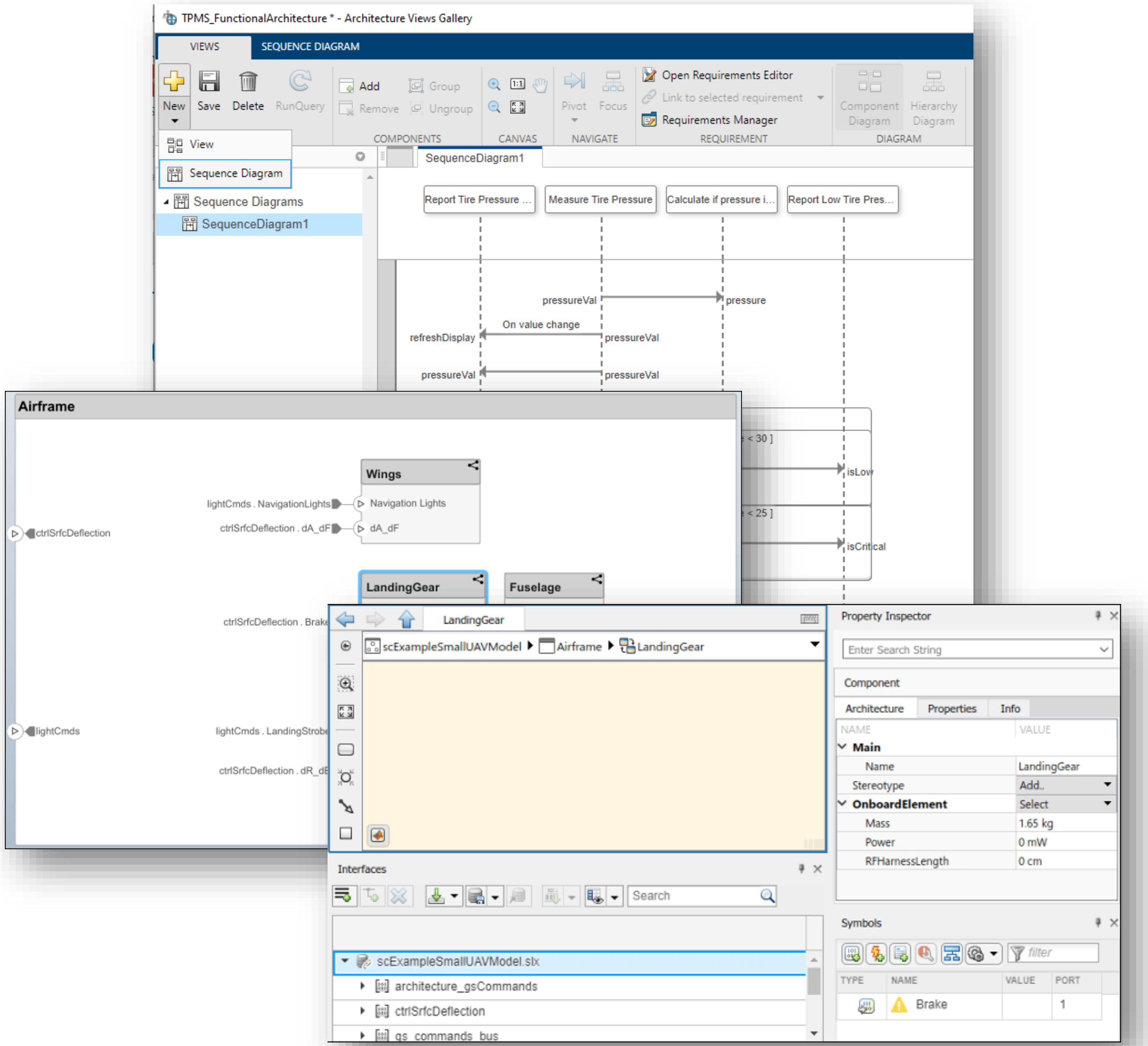
- System Composer
  - Sequence diagrams
  - Stateflow charts in components



MATLAB EXPO

New Features in R2021a

- System Composer
  - Sequence diagrams
  - Stateflow charts in components

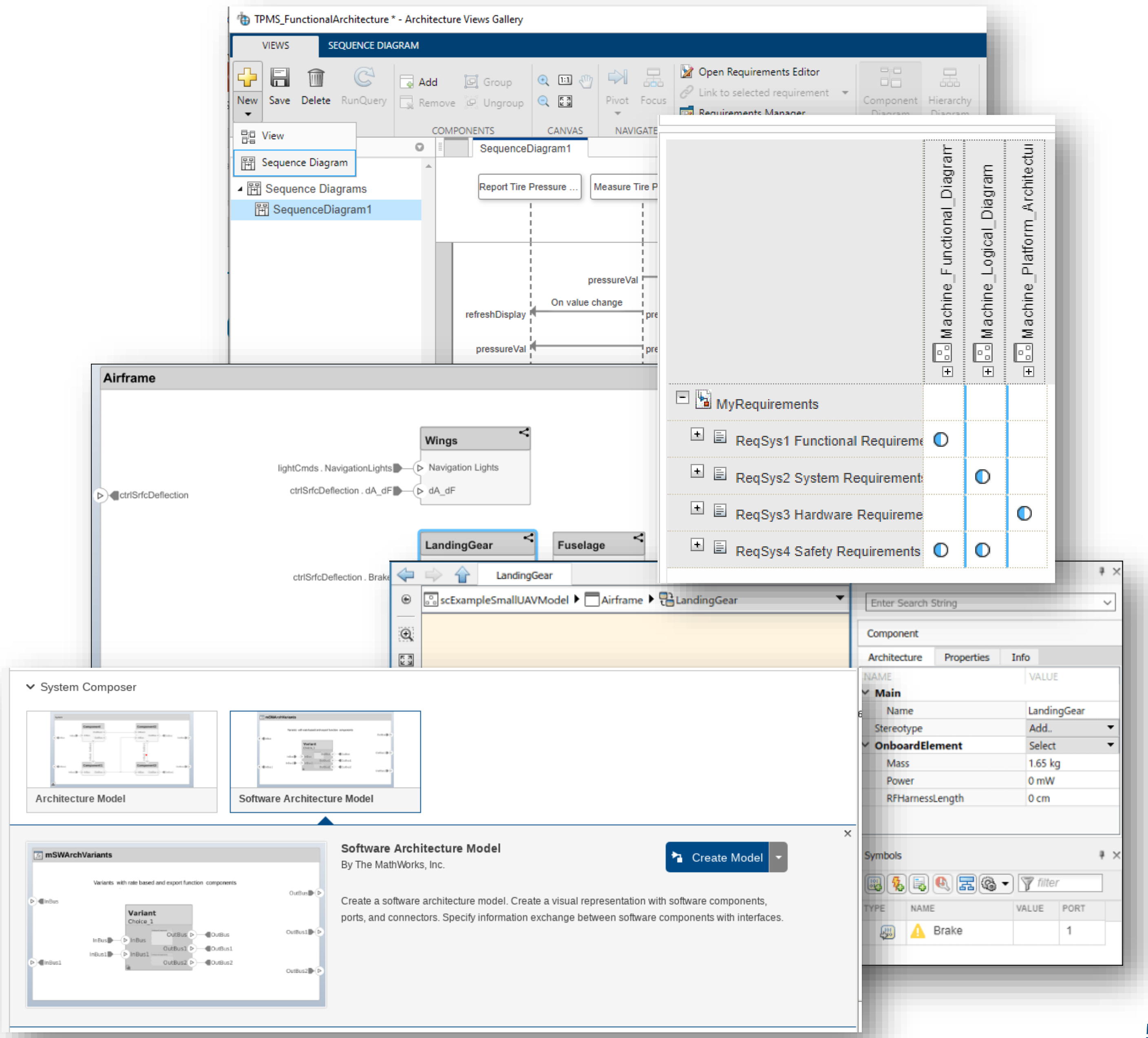


NAME	VALUE
Main	
Name	LandingGear
Stereotype	Add..
OnboardElement	Select
Mass	1.65 kg
Power	0 mW
RFHarnessLength	0 cm

MATLAB EXPO

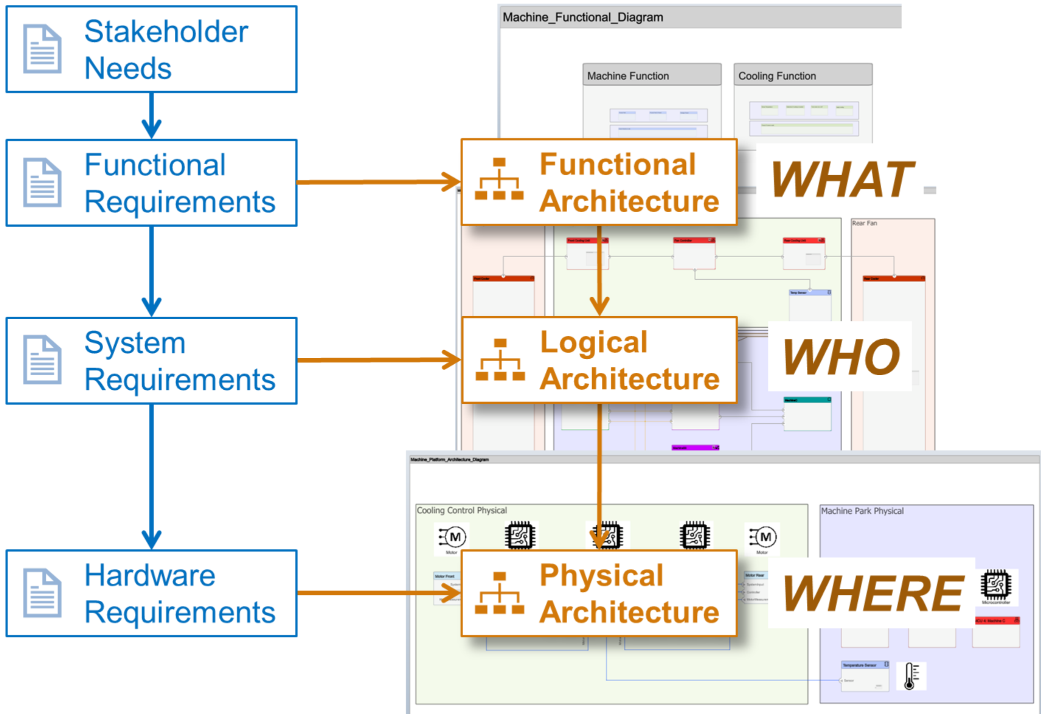
New Features in R2021a

- System Composer
  - Sequence diagrams
  - Stateflow charts in components
  - Software architectures
- Simulink Requirements
  - Editor improvements
  - Multi-artifact traceability matrix

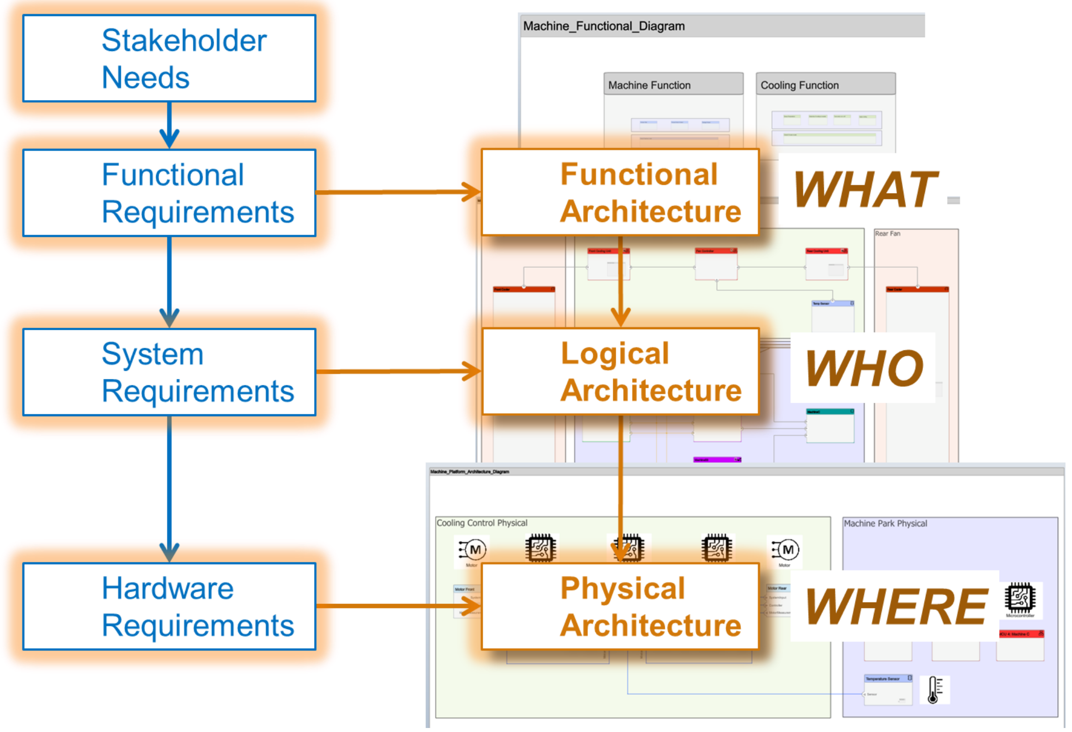


Create a software architecture model. Create a visual representation with software components, ports, and connectors. Specify information exchange between software components with interfaces.

Key Takeaways

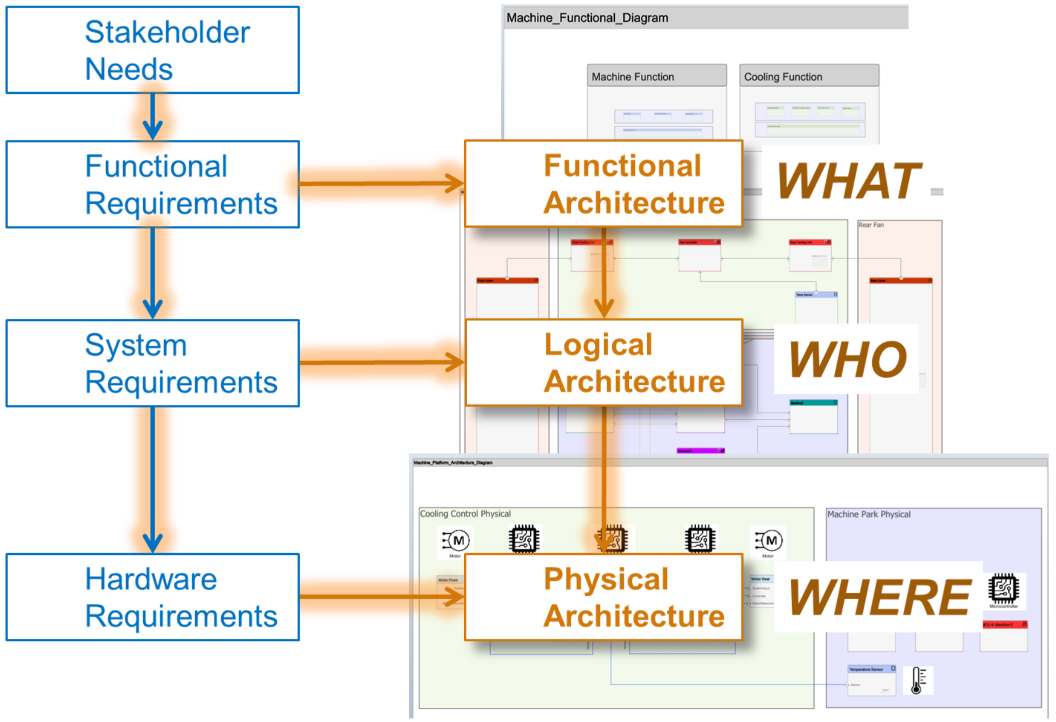


Key Takeaways



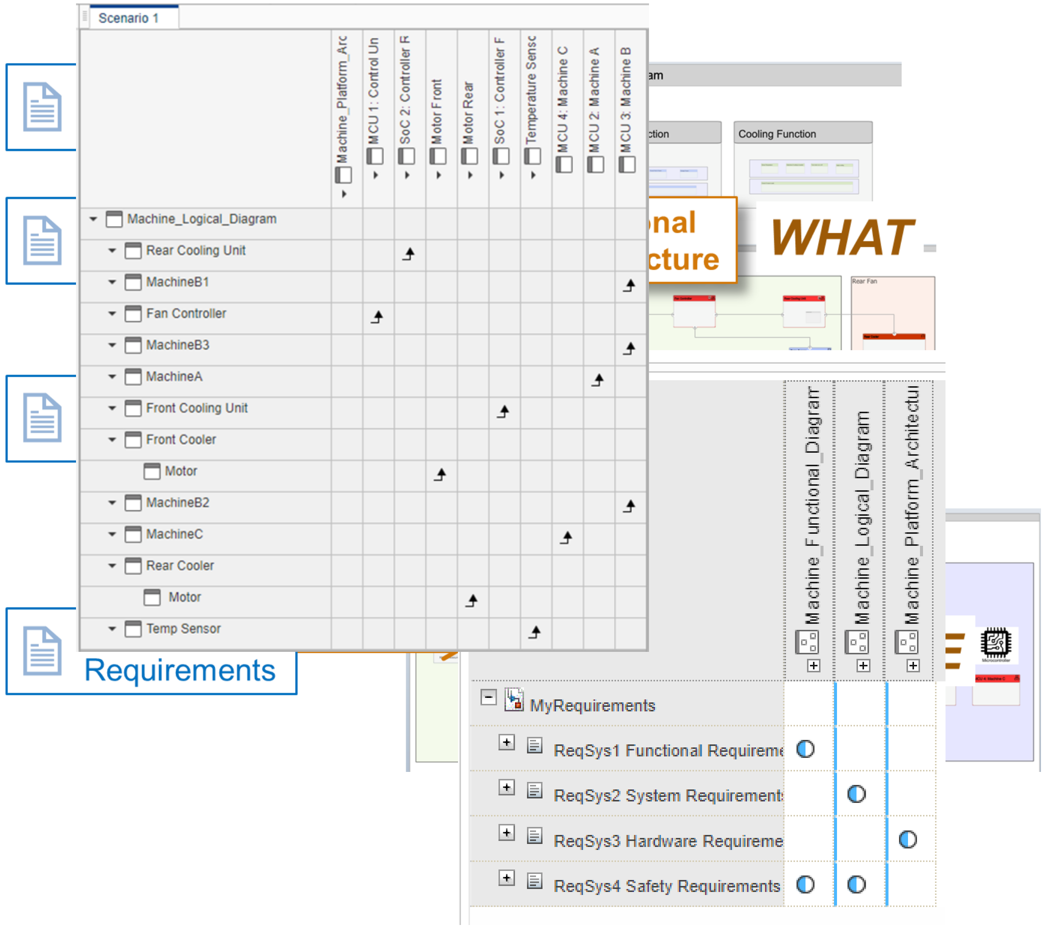
- You can import, write, and store textual requirements right in the same environment as your architecture and design models.

Key Takeaways



- You can import, write, and store textual requirements right in the same environment as your architecture and design models.
- You can establish relationships among multiple requirements and architecture artifacts to understand the impact of changes.

Key Takeaways

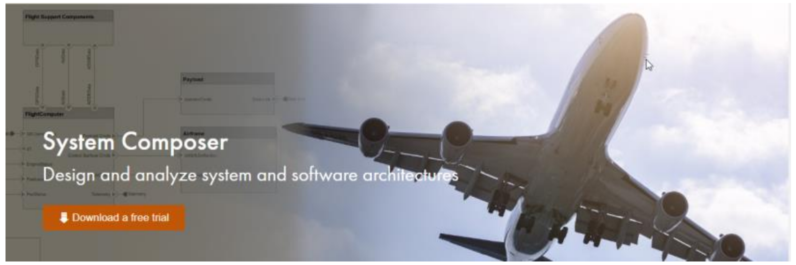


	Machine_Platform_Arc	MCU 1: Control Un	SoC 2: Controller F	Motor Front	Motor Rear	SoC 1: Controller F	Temperature Sensc	MCU 4: Machine C	MCU 2: Machine A	MCU 3: Machine B
Machine_Logical_Diagram										
Rear Cooling Unit			↗							
MachineB1										↗
Fan Controller		↗								
MachineB3										↗
MachineA									↗	
Front Cooling Unit						↗				
Front Cooler										
Motor				↗						
MachineB2										↗
MachineC								↗		
Rear Cooler										
Motor					↗					
Temp Sensor							↗			

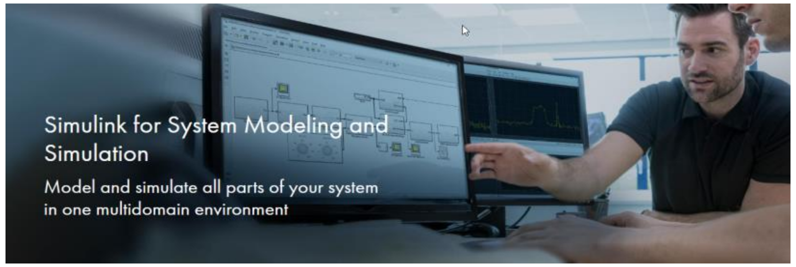
	Machine_Functional_Diagram	Machine_Logical_Diagram	Machine_Platform_Architectu
MyRequirements			
ReqSys1 Functional Requireme	◐		
ReqSys2 System Requirement		◐	
ReqSys3 Hardware Requireme			◐
ReqSys4 Safety Requirements	◐	◐	

- You can import, write, and store textual requirements right in the same environment as your architecture and design models.
- You can establish relationships among multiple requirements and architecture artifacts to understand the impact of changes.
- You can visualize those relationships to assess the completeness of your system.

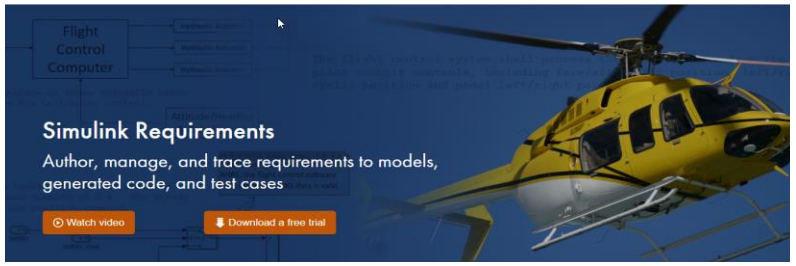
Learn More



System Composer


Simulink for System Modeling and Simulation
Model and simulate all parts of your system in one multidomain environment

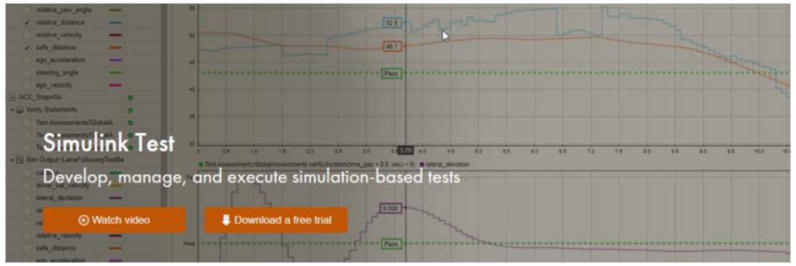
Model-Based Systems Engineering


Simulink Requirements
Author, manage, and trace requirements to models, generated code, and test cases
Watch video
Download a free trial

Simulink Requirements

MATLAB and Simulink for Model-Based Systems Engineering (MBSE)
Design, analyze, and test system and software architectures.

System Modeling and Simulation



Simulink Test

AUTOSAR

MATLAB EXPO
2021

Thank you

MathWorks®