

# GETTING THE MOST OUT OF TEAMWORK CLOUD (MAGIC COLLABORATION STUDIO)

**Tomas Vileiniskis**

CATIA Systems Modeling Application Senior Manager

Teamwork Cloud Product Manager



## Tomas Vileiniškis

### Teamwork Cloud Product Manager

- 18 years of experience in Computer and Software Engineering
- Leading Teamwork Cloud development for more than 7 years
- 3DS representative at OMG SysML v2 API and Services RTF
- Active member of OMG RAS 3.0 (Reusable Assets) RFC

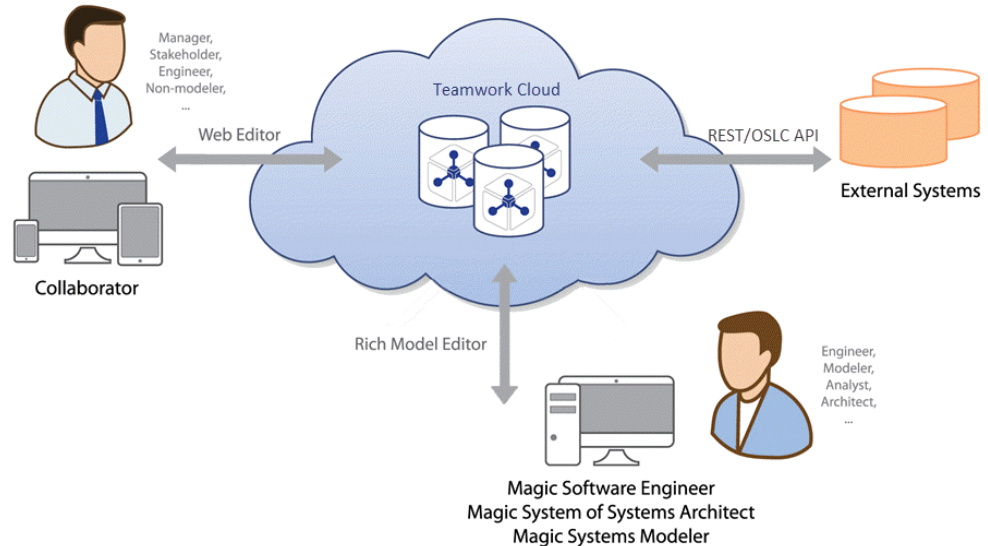
# WHAT IS MAGIC COLLABORATION STUDIO?

- Magic Collaboration Studio is CATIA No Magic's bundle of products for collaborative model development
  - Teamwork Cloud
  - Cameo Collaborator



Teamwork Cloud

CAMEO<sup>®</sup>  
COLLABORATOR



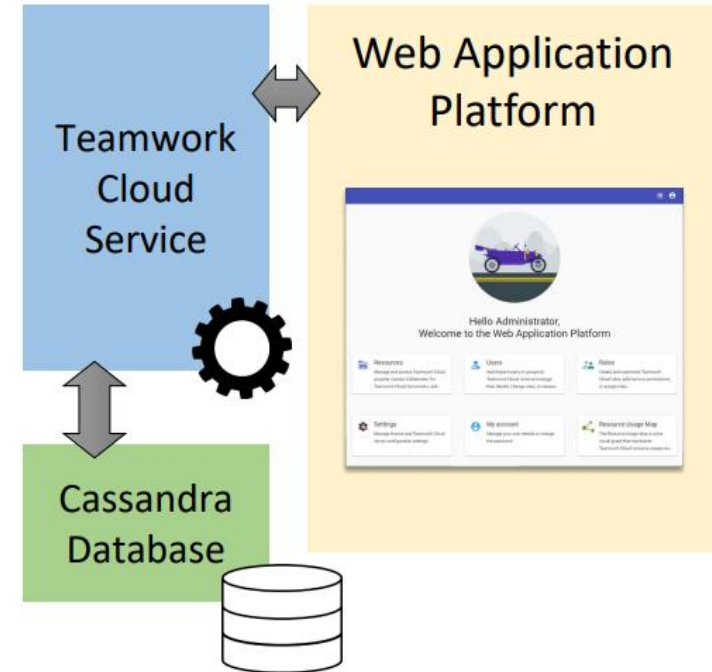
# OUTLINE

- Introduction
- Key Model Management Principles
- APIs and Services
- Recent feature highlights

# INTRODUCTION

# WHAT IS TWC?

- Teamwork Cloud is CATIA No Magic's repository for collaborative model development and model version storage.
- As the name suggests, this product is Cloud-ready, and it can be deployed on a local hardware (VM) or on a hosted server in the Cloud.
- Teamwork Cloud can be accessed both by Cameo modeler and web interface
- Teamwork Cloud also exposes its data via REST and OSLC APIs



# WHAT CAN I DO WITH TWC?

- Collaborative modeling (internal and external)
- Model versioning
- Model history
- Model branching and merging
- Access control
- Model and repository analysis
- Model exchange

# INTERACTING WITH TWC



Hello Administrator,  
Welcome to the Web Application Platform



## Resources

Manage and access Teamwork Cloud projects, Cameo Collaborator for Teamwork Cloud documents, and OSLC resources.



## Users

Add/import users or groups to Teamwork Cloud, remove/manage their details, change roles, or release locked elements.



## Roles

Create and customize Teamwork Cloud roles, add/remove permissions, or assign roles.



## Settings

Manage license and Teamwork Cloud server configuration settings.



## Resource Usage Map

The Resource Usage Map is a live visual graph that represents Teamwork Cloud resource usages as well as identifies potential problem areas.



## My account

Manage your user details or change the password.



# KEY MODEL ORGANIZATION AND MANAGEMENT PRINCIPLES

# RESOURCE ORGANIZATION

- Resource types

- Cameo models



- Collaborator documents

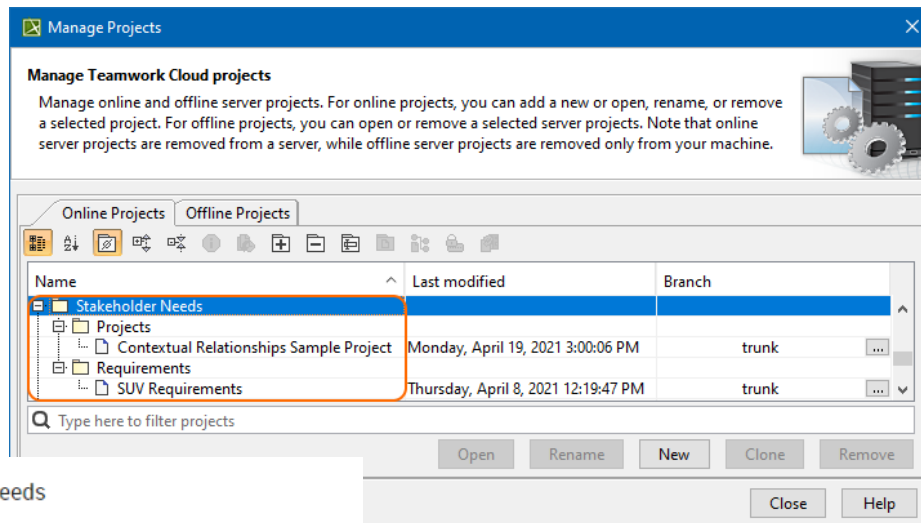
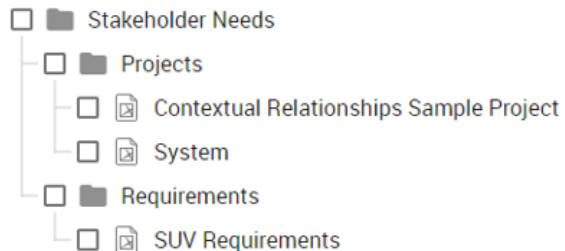


- OSLC documents



- Resources are grouped into categories

- Categories can be nested



# ACCESS CONTROL

- Teamwork Cloud uses Role Based Access Control (RBAC) mechanism to handle access to resources and categories in the repository
- A predefined set of Roles comes with installation by default
- Custom Roles can be created containing custom subset of Permissions
- Roles can be global or custom-scoped
- An end user gets assigned a Role and a Scope (in case of a scoped permission)
- Data Markings as an additional security layer on top



## Resource Manager

Predefined role

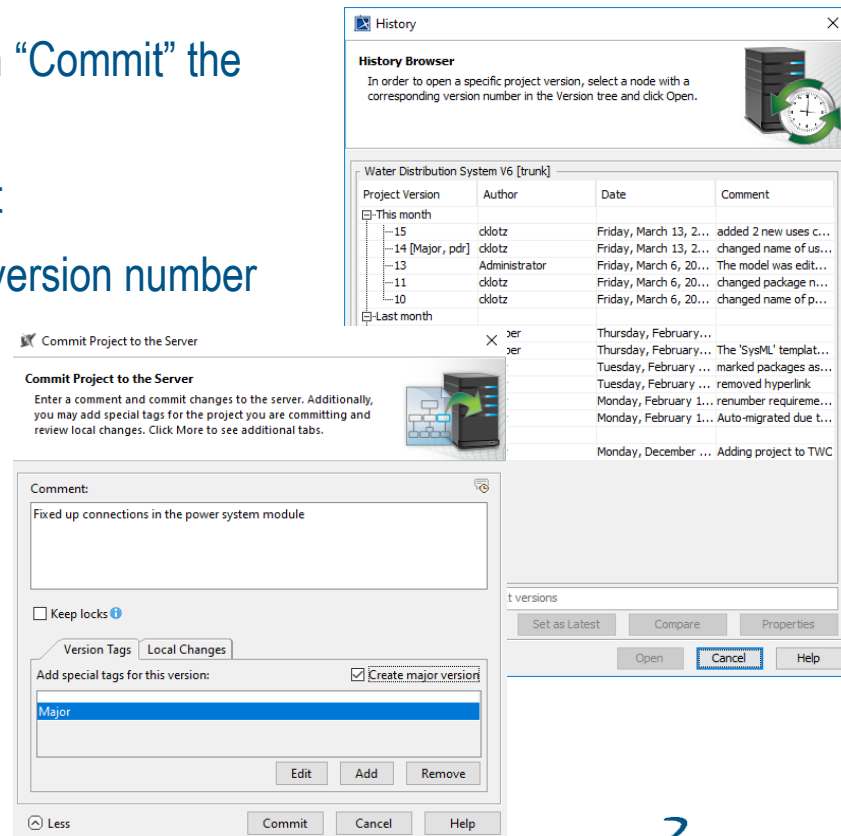
Resource-specific role. Users who are assigned to this role have full permission to manage selected resource, including the ability to grant permissions to other users.

### Permissions

Administer Resources  
Edit Resource Properties  
Edit Resources  
List All Users  
Manage Model Permissions  
Manage Owned Resource Access Right  
Read Resources  
Remove Resource

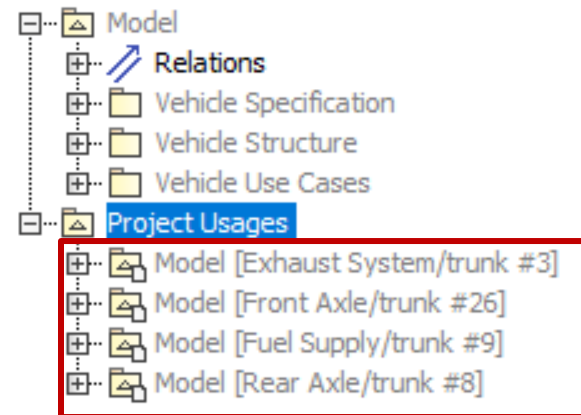
# MODEL VERSIONING

- When a user modifies a project, the user will then “Commit” the changes to Teamwork Cloud
- Each commit creates a new version of the project
- A version will be assigned a unique consecutive version number
- Significant model versions driving model lifecycle state changes can be marked by tags
- Optimistic and pessimistic locking modes



# MODEL COMPOSITION

- When modeling complex large systems, model composition plays a crucial role
- TWC project can be decomposed to consist of other projects (Used Projects)
- Libraries, Sub-systems, Requirements etc.
- Master project can reference (use) different versions and branches of Used Projects
- Utilities available for efficient Used Project configuration and version handling



# APIs and Services

# WHAT KIND OF APIs?

- Administration
- Repository Management
- Model Management
- Model Decomposition and DSL extensions
- Webhooks
- **SysML v2 standard API and Services**

## Model Management

GET	/osmc/resources/{resourceId}/artifacts	List artifacts in the latest revision of the master branch.
POST	/osmc/resources/{resourceId}/artifacts	Upload a blob into a particular resource.
GET	/osmc/resources/{resourceId}/artifacts/{artifact}	Get the artifact information in the latest revision of the master branch.
GET	/osmc/resources/{resourceId}/branches/{branchId}/artifacts	List artifacts in the latest revision of a particular branch.

## Repository Management

GET	/osmc/resources	List all projects.
DELETE	/osmc/resources/{resourceId}	Delete a project.
GET	/osmc/resources/{resourceId}	Get information about the resource (the project).
PATCH	/osmc/resources/{resourceId}	Change workspace or replace the metadata of the resource.
GET	/osmc	

## Query Services

GET	/api/projects/{projectId}/queries/{queryId}
PUT	/api/projects/{projectId}/queries/{queryId}
DELETE	/api/projects/{projectId}/queries/{queryId}
POST	/api/projects/{projectId}/query-results
GET	/api/projects/{projectId}/queries

GET /sb7\blo]ecfz\{blo]ecfzg)\dneL7ez

GET /sb7\blo]ecfz\{blo]ecfzg)\dneL7ez

# WEBHOOKS

1 Webhook

2 Resource scope

## Create new webhook

Webhook title \*

Listen to HSUV model events

Webhook URL \*

https://sendeventshere.web

Webhook scope

Model element

Event

Commit

### Enabled

When enabled, upon creation of webhook it will be listening for new events



### Protect webhook URL with basic authentication

Allows you to add a username and password to secure your webhook URL



CANCEL

NEXT

## Webhooks

Webhooks let you listen for model change events in real time and easily pass data to use in services and integrations

<input type="checkbox"/> Title ↑	URL	Sending event	Status
<input type="checkbox"/> Listen to HSUV model changes	https://sendeventshere.webhook	Commit	Enabled

Webhook

Resource scope

3 Element scope

## Select element scope

- Model
  - Relations
- HSUVModel
  - Relations
  - Explanations
  - HSUV Analysis**
  - HSUV Behavior
  - HSUV Instance Values
  - HSUV Requirements

☐ Select recursively

CANCEL

CREATE

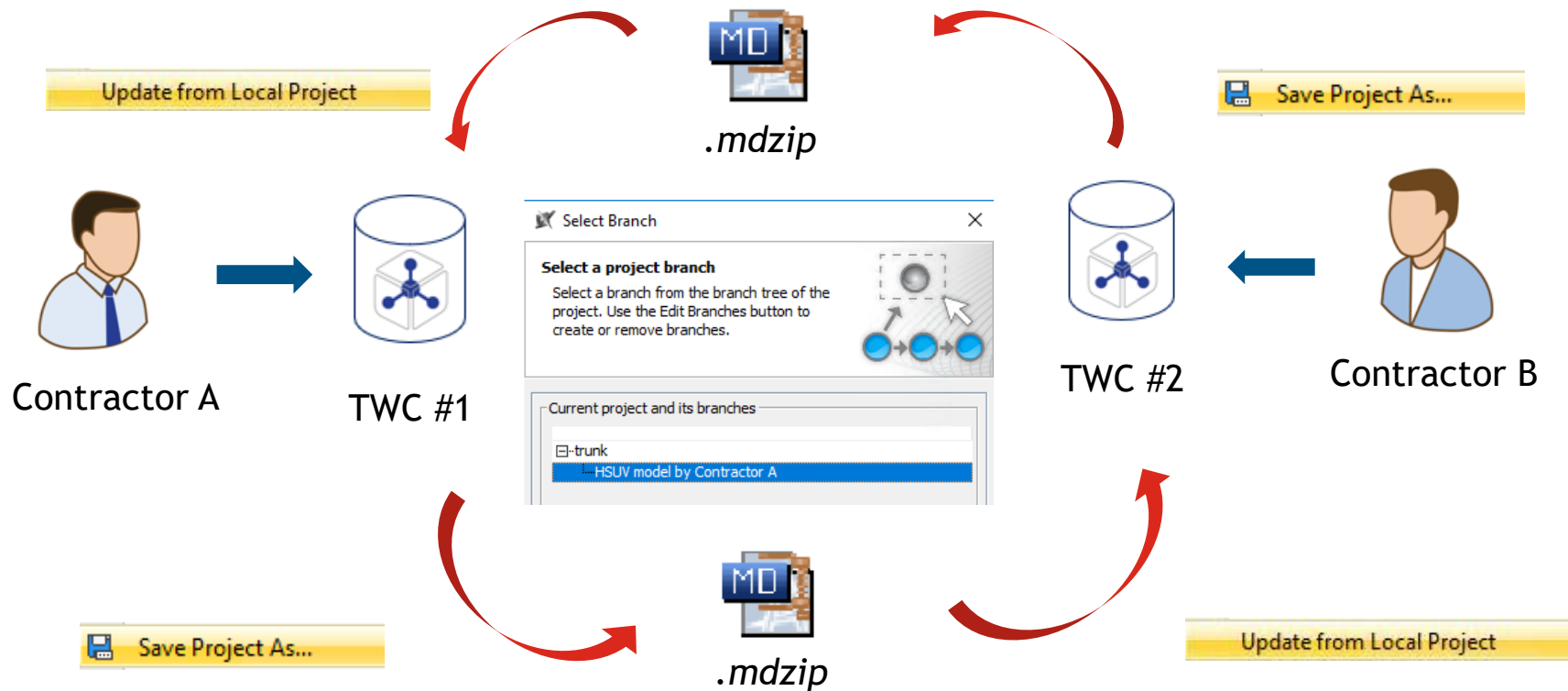
CI/CD Pipeline automation





# MODEL EXCHANGE

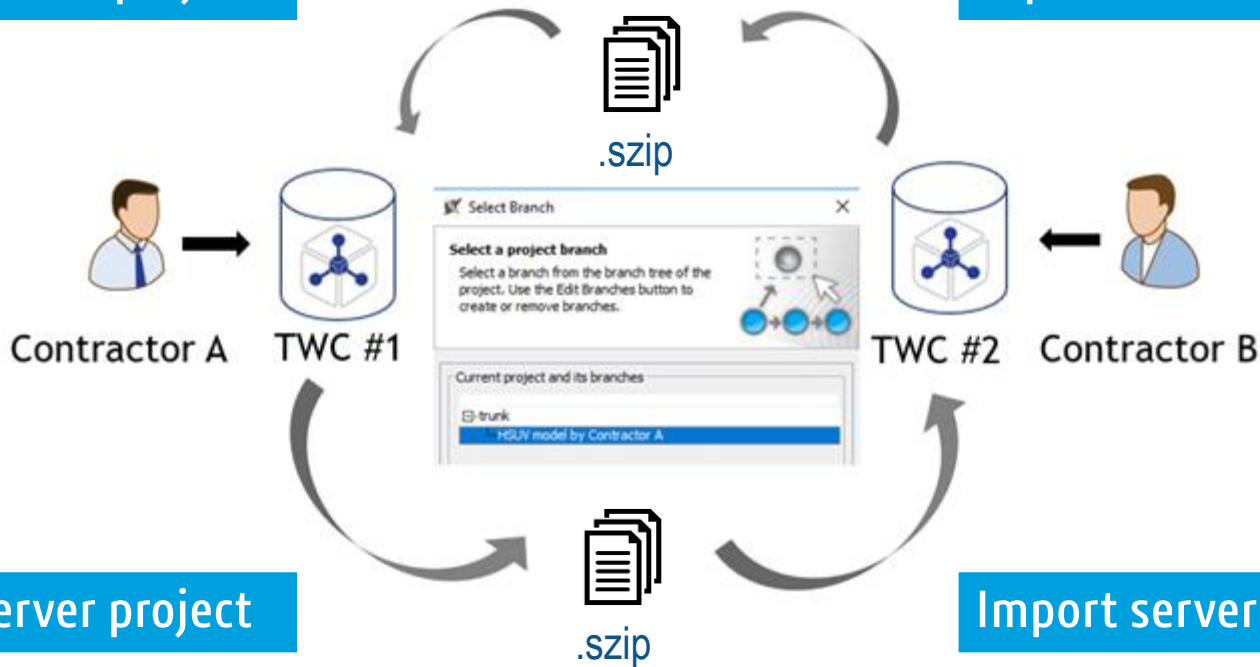
# CLASSICAL MODEL EXCHANGE VIA .MDZIP



# REWORKED DISCONNECTED TEAM COLLABORATION

Import server project

Export server project

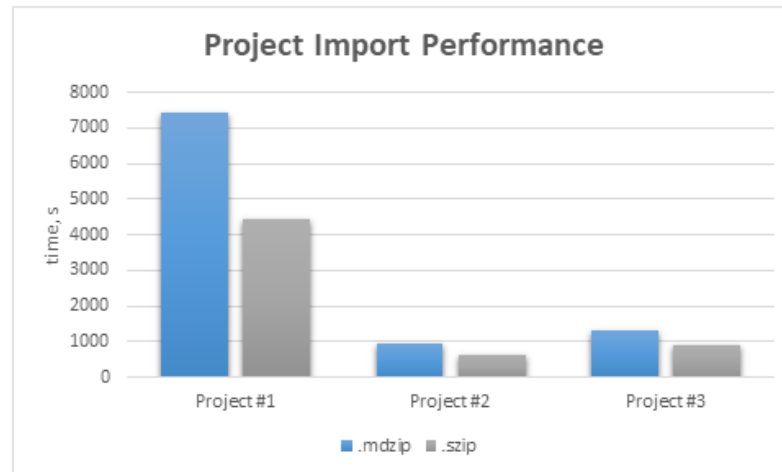
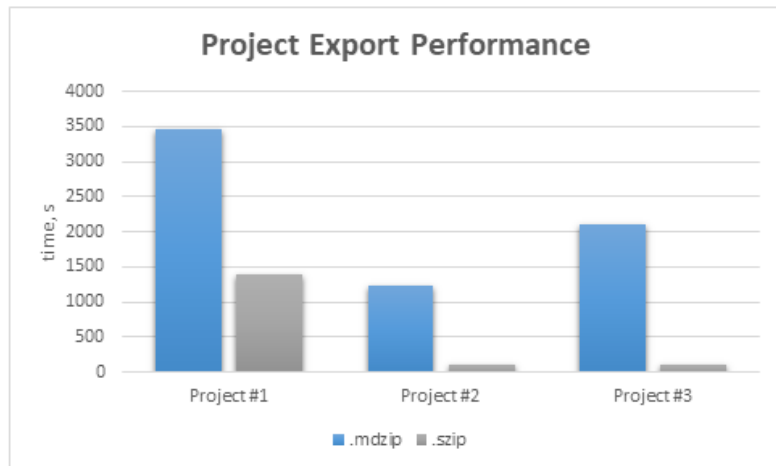


Export server project

Import server project

.mdzip → .zip (server zip)

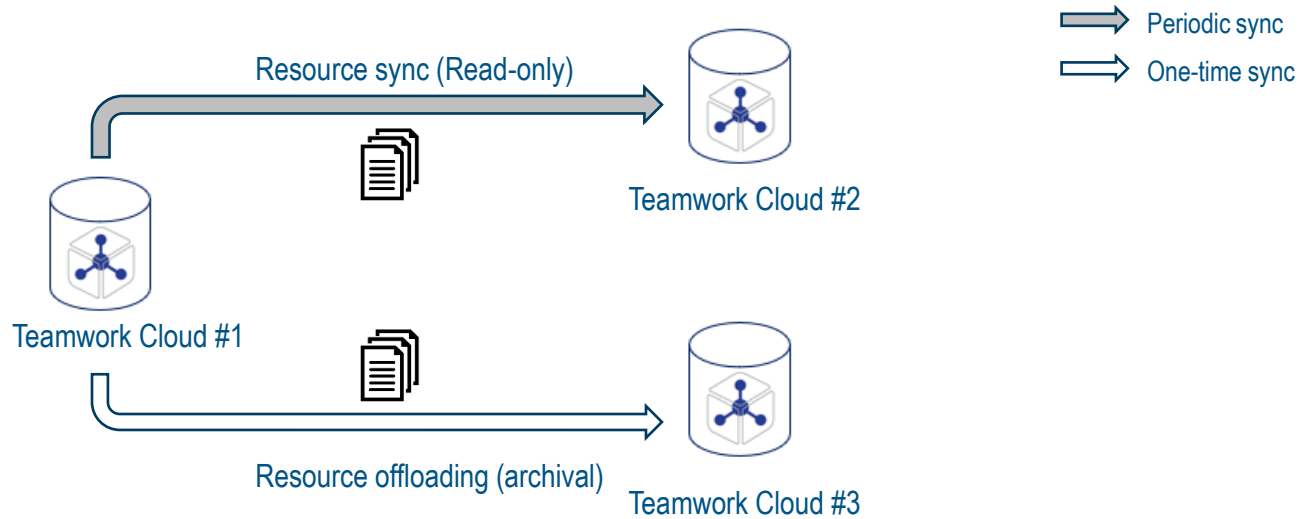
# .MDZIP VS .SZIP PERFORMANCE



*Project #1 (1M elements, 57 Used Projects), Project #2 (2M elements, 7 Used Projects), Project #3 (2M+ elements, 3 Used Projects)*

# CROSS-CLUSTER SYNC

# CROSS-CLUSTER RESOURCE SYNC



Synchronize/move resources across multi-site TWC deployments

# CROSS-CLUSTER RESOURCE SYNC (II)

## Remote repository management

Synchronize resources from another instance of Teamwork Cloud

<input type="checkbox"/>	Category/Resource from Source ↑	Target category	Source address	Synchronization mode	Status	Synchronized	
<input type="checkbox"/>	Vehicle Electrical System	Supplier A Models	twc-a-supplier	On every new commit	Partially synchronized	Sep 24, 2021, 11:23:06 AM	⋮
<input type="checkbox"/>	Climate Control System	From UK Site	uk-twc-site	majorVersion	Synchronized	Sep 20, 2021, 2:33:09 PM	⋮
<input type="checkbox"/>	Temperature Regulation Loop	Supplier B Models	twc-b-supplier	Manual	Synchronized	Sep 24, 2021, 10:11:05 AM	⋮

1 Login to source

2 Select resources

3 Select target

4 Mode

### Log in to the source server

Log in to the Teamwork Cloud server (source server) from which you want to synchronize resources and/or categories. Use the source server credentials to log in.

Source Teamwork Cloud address \*

remote-uk-site-twc

Username \*

Syncer

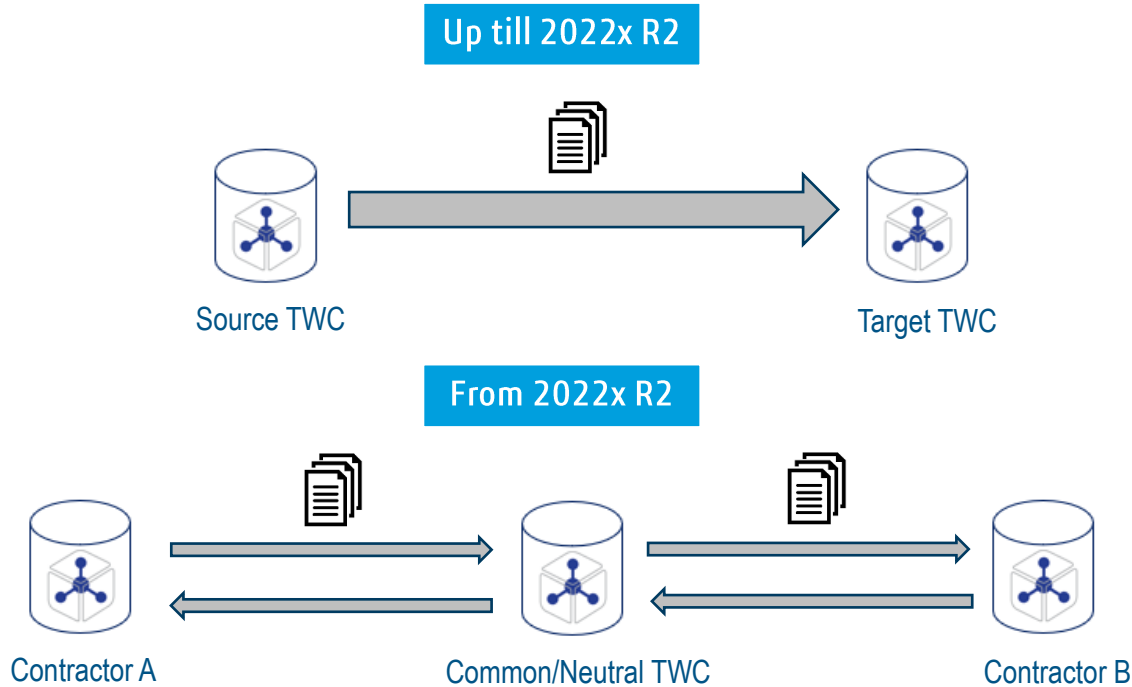
Password \*

•••••

CANCEL NEXT



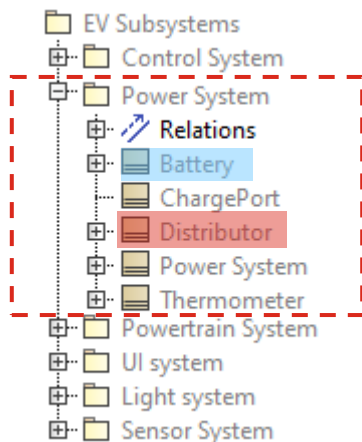
# CROSS-CLUSTER RESOURCE SYNC TOPOLOGIES



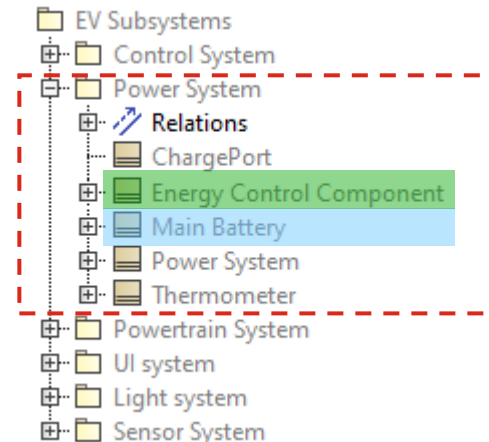
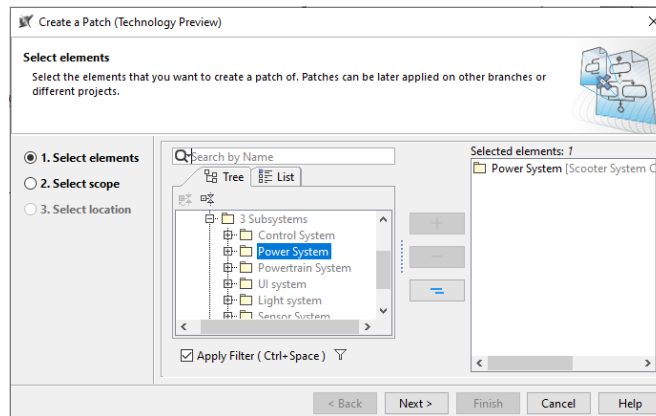
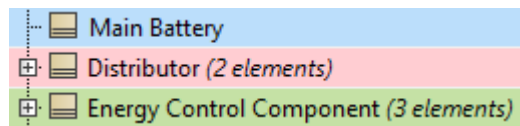


# MODEL PATCHING

# PORTABLE MODEL DIFF & PATCH (EXPORT)



**Source Model v1**

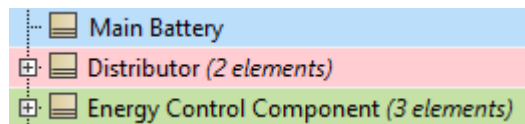


**Source Model v2**

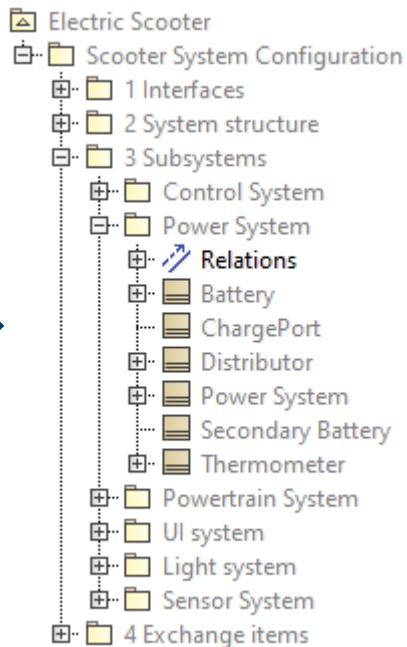


**PowerSystemV1-V2.patch**

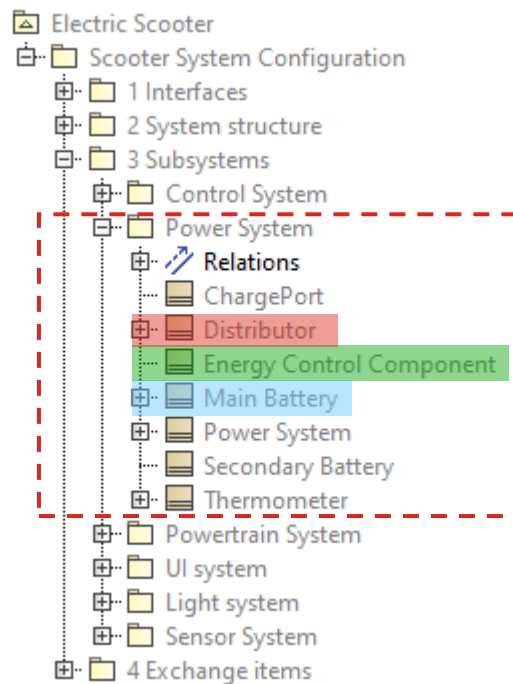
# PORTABLE MODEL DIFF & PATCH (APPLICATION)



*PowerSystemV1-V2.patch*



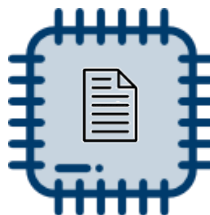
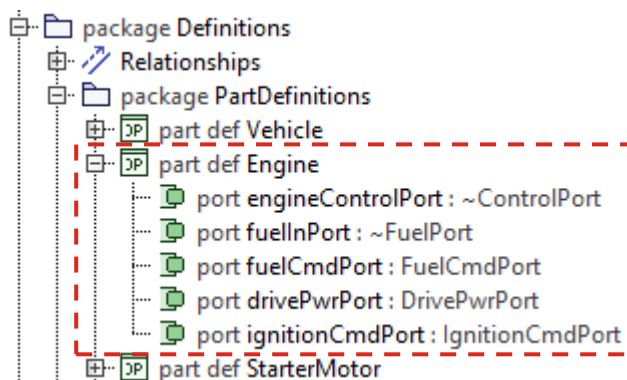
*Target Model*



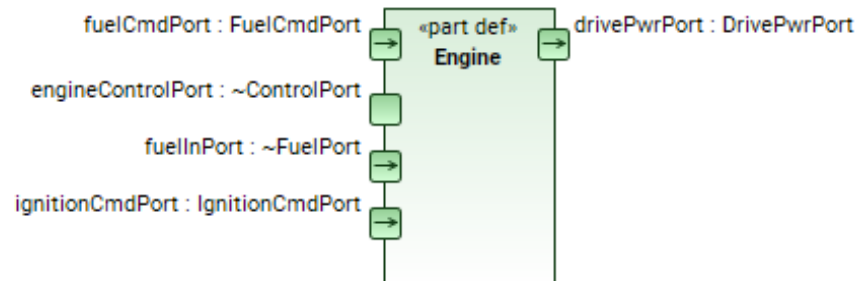
*Resulting Target Model*

# MODEL PATCHING (BB)

## A Serializable Model Change Set

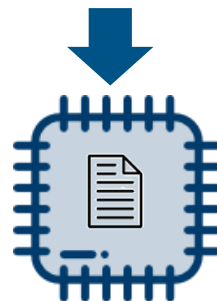
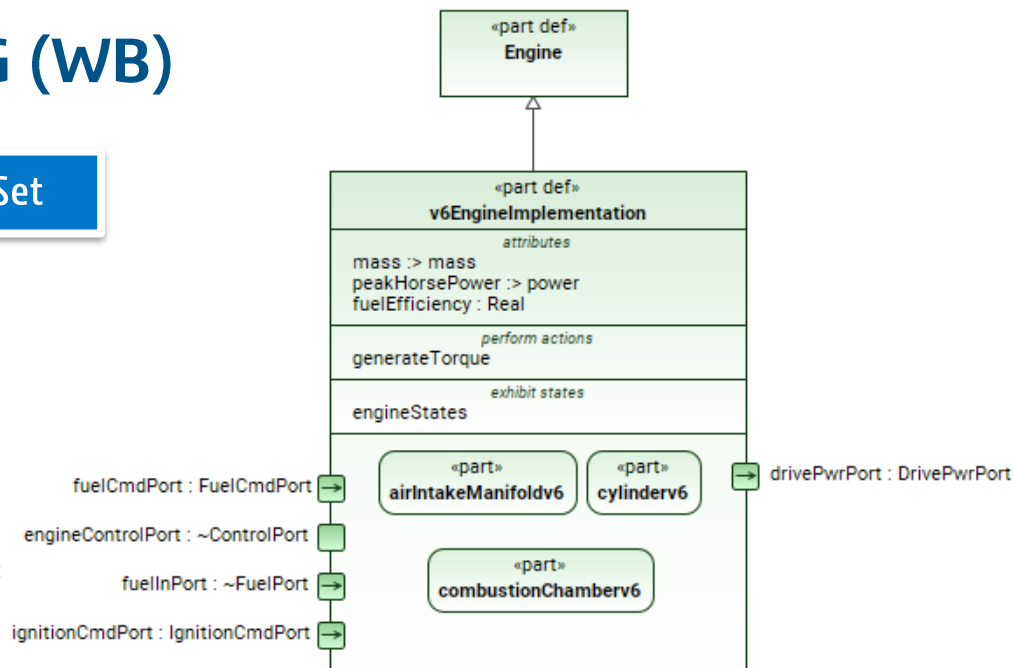
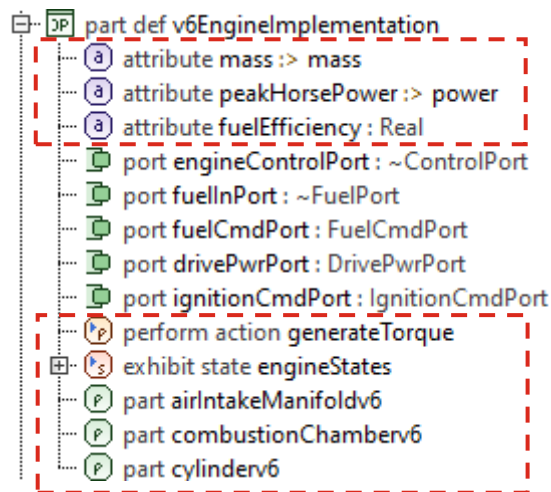


EngineBlackBox.patch



# MODEL PATCHING (WB)

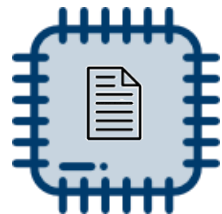
## A Serializable Model Change Set



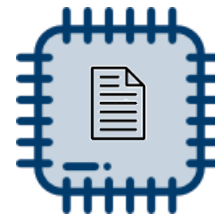
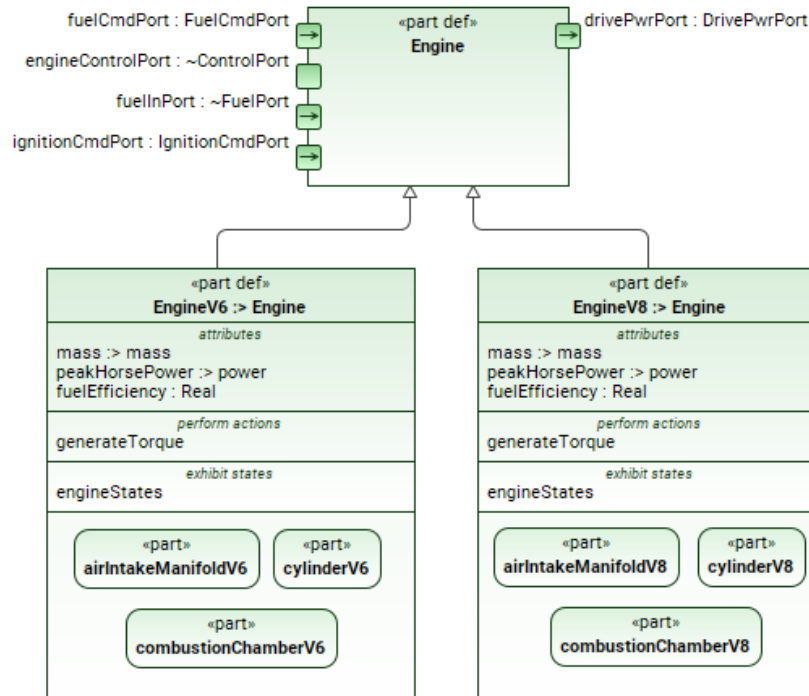
EngineImpl1.patch

# MODEL PATCHING (MULTIPLE WBs)

## A Serializable Model Change Set

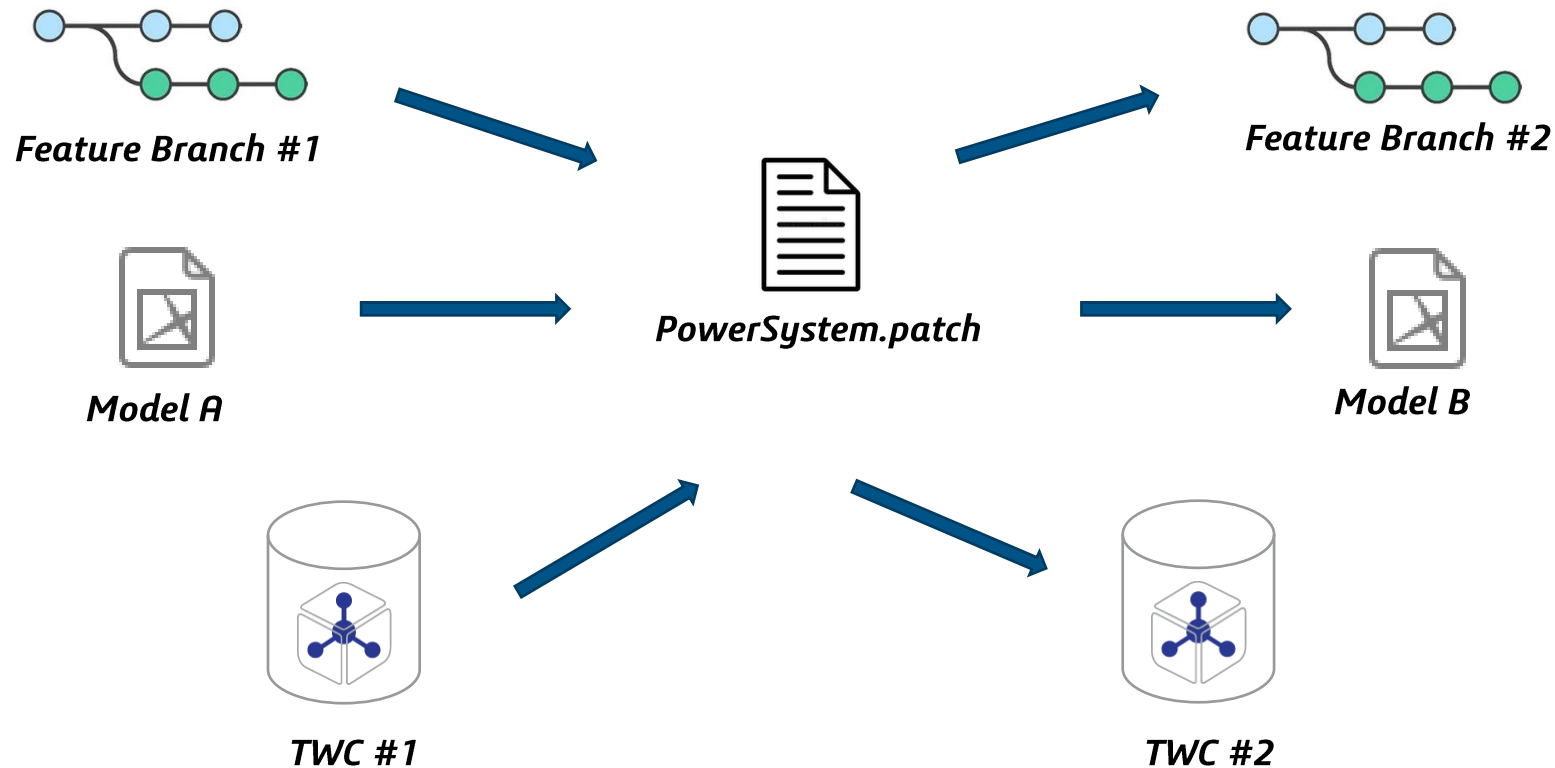


EngineImpl1.patch



EngineImpl2.patch

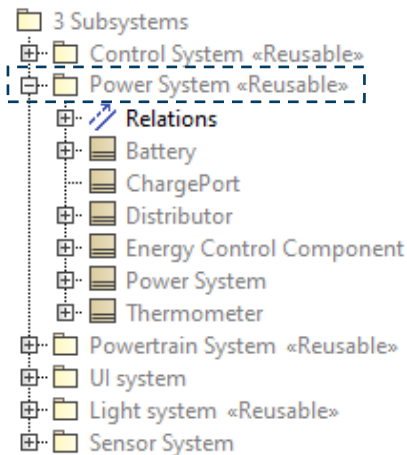
# PORTABLE MODEL DIFF & PATCH



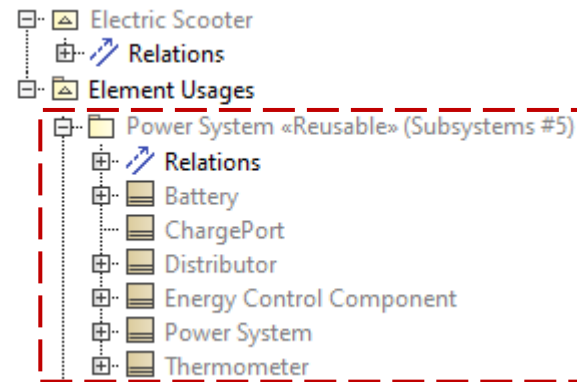
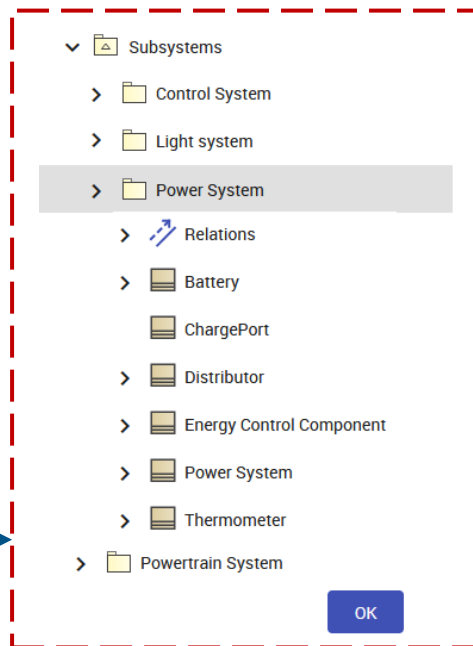
# USED PROJECTS 2.0



# LIGHTWEIGHT ELEMENT REUSE (TECH. PREVIEW)



1. Light model tree browser



1. Select elements to reuse
2. Import the scoped selection only



