Systems Engineering Research Center (SERC)

Collaboration in an Authoritative Source of Truth Environment using OpenMBEE

By:
Benjamin Kruse, Sc.D.
Mary A. Bone, Ph.D.
Mark Blackburn, Ph.D.

SYSTEMS ENGINEERING
Research Center

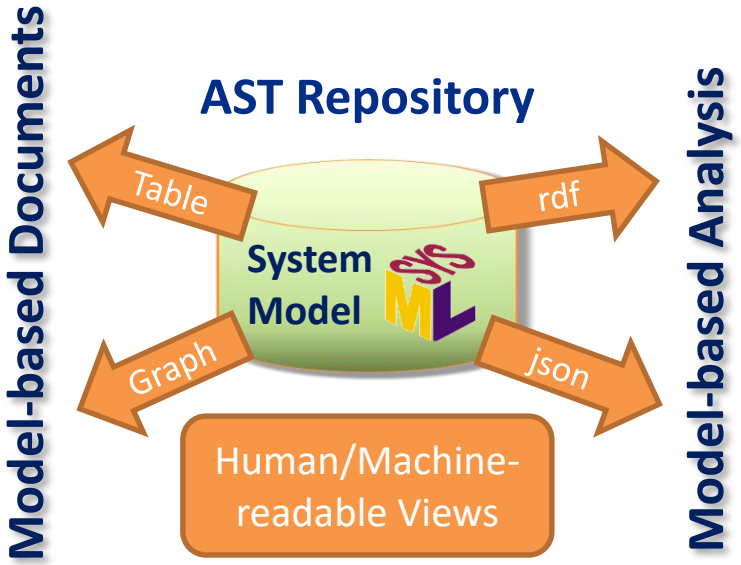
Copyright and Disclaimer

Agenda

- Authoritative Source of Truth & Surrogate Pilot
- OpenMBEE: Lessons Learned
  - Model Development Kit (MDK) & DocGen
  - Model Management System (MMS)
  - View Editor
- Project Usage & Management
- Summary

Introduction

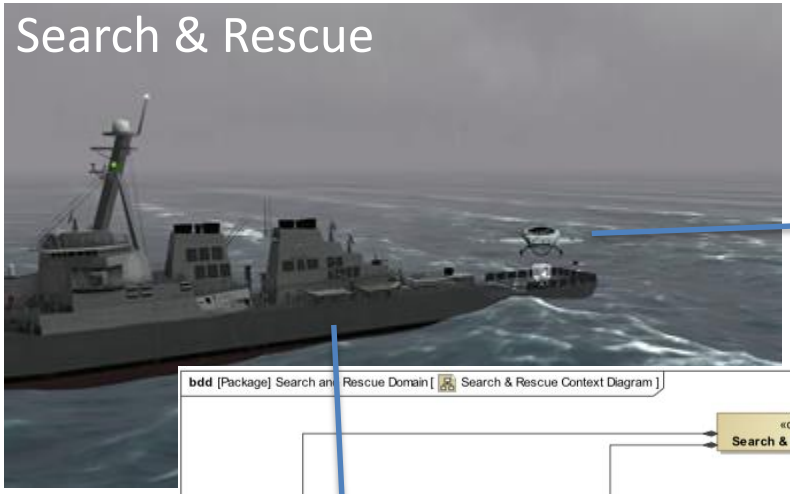
- Authoritative Source of Truth (AST)
  - To provide consistent data in the format necessary for the given task
  - Implemented by **OpenMBEE** = Open Model Based Engineering Environment, developed by NASA/JPL



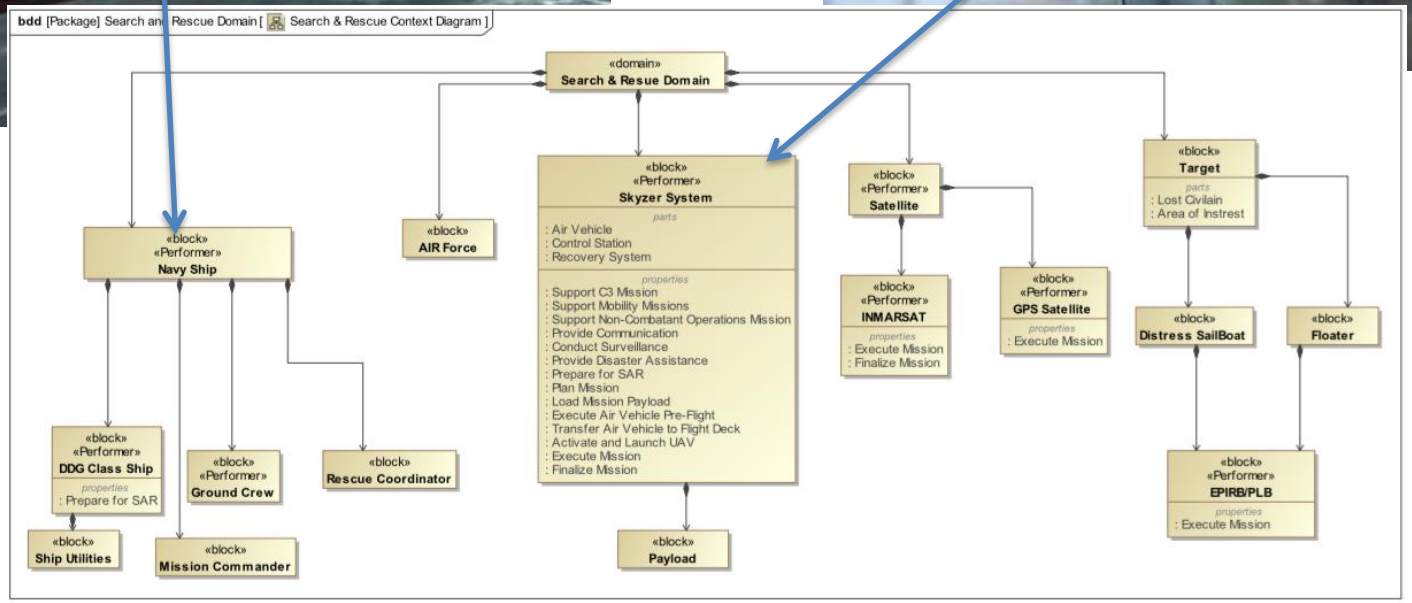
- Surrogate Pilot
  - Execution of NAVAIR's Systems Engineering Transformation (SET) Framework
  - To simulate collaboration in an AST
  - To support new operational paradigm between government and industry
    - By elimination of paper artifacts and large-scale design reviews in favor of continuous insight/oversight via the digital collaborative environment

Surrogate Pilot Scenario: Skyzer UAS

Graphical CONOPS Scenario:



Airfoil designation for a similar Bell tilt-rotor (BA609)

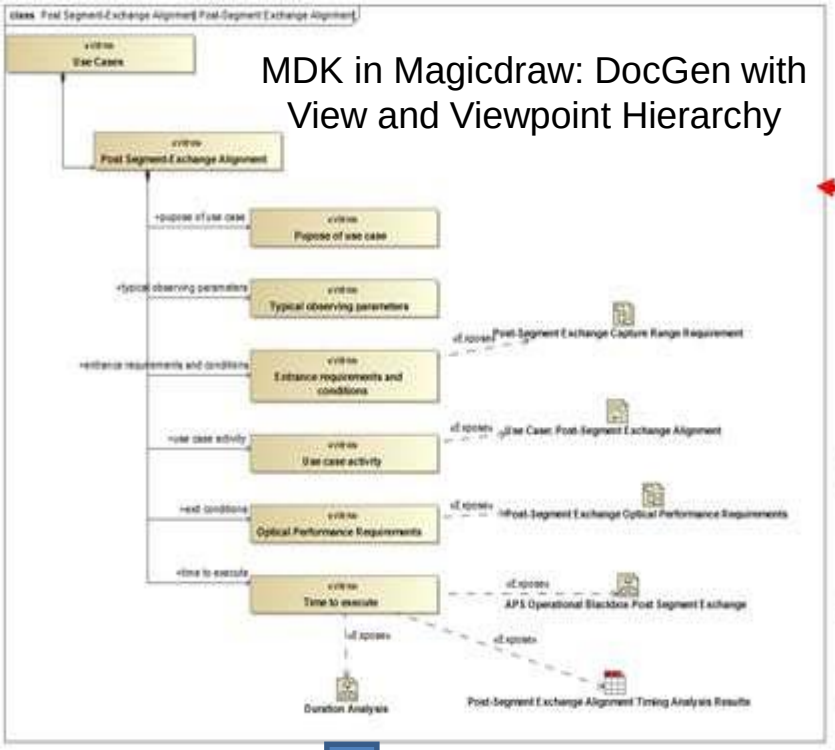


Skyzer System & Mission Models developed using SysML

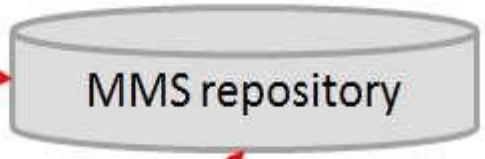
SYSTEMS ENGINEERING
Research Center

OpenMBEE: MDK, MMS & View Editor

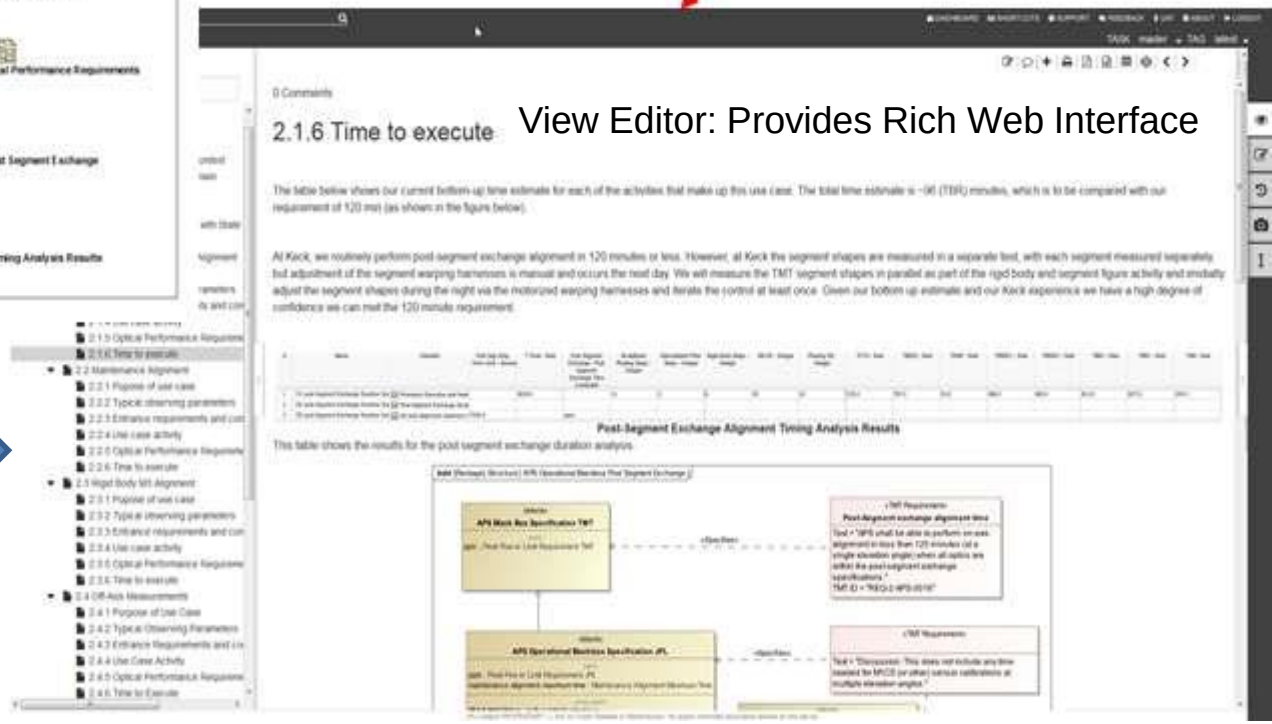
MDK in Magicdraw: DocGen with View and Viewpoint Hierarchy



Model Management System (MMS)



Visualization in View Editor

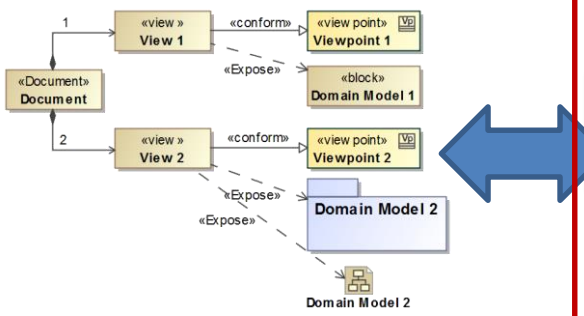


View Editor: Provides Rich Web Interface

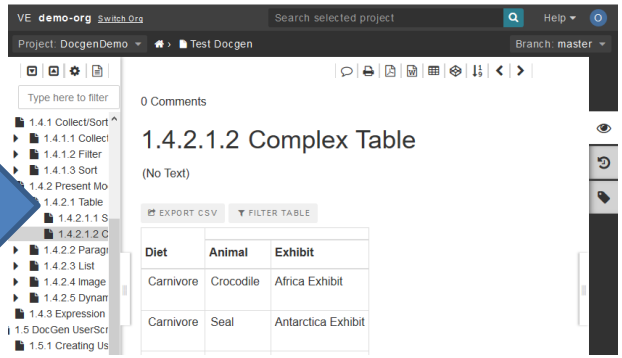
Model Development Kit (MDK)

- What is MDK?
  - Plugin for Magicdraw, to support building system assemblies through modeling augmentation and validation, enable syncing with MMS and using the DocGen language for model-based document creation using views and viewpoints
  - Content
    - Systems Reasoner
    - MMS Sync
    - DocGen

MDK: View and Viewpoint Hierarchy



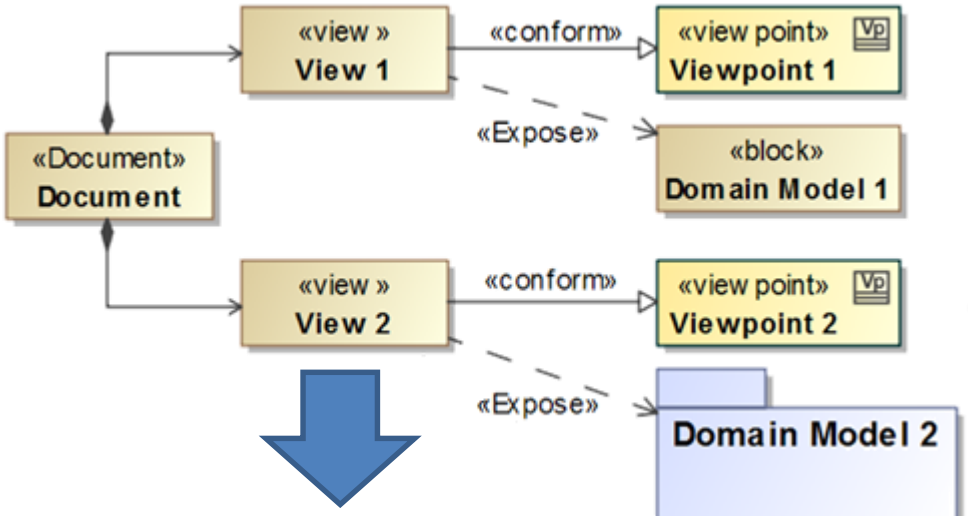
Model Management System (MMS)

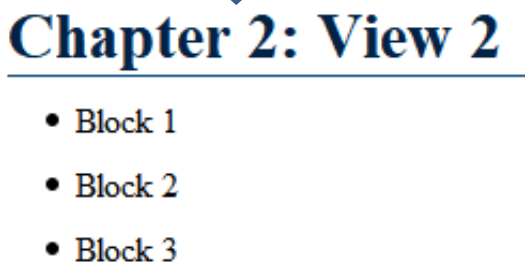


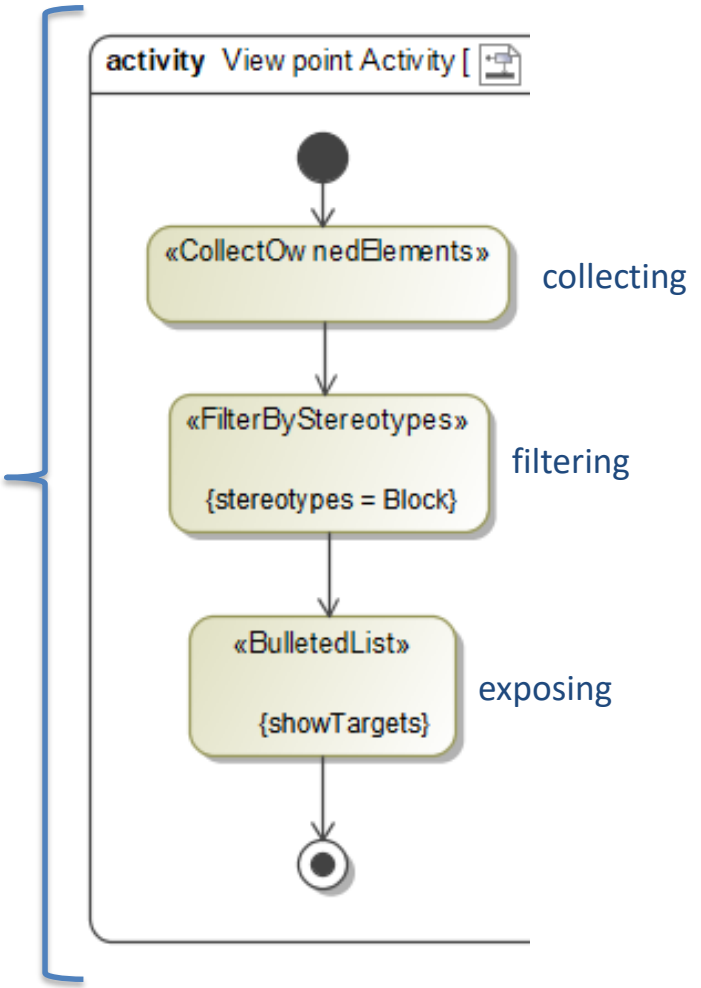
View Editor: Provides Rich Web Interface

MDK: DocGen

- DocGen
  - For model-based document creation based on Views and Viewpoints
    - To generate views for View Editor or pdfs
    - To guide modeling and development







MDK: DocGen Lessons Learned

- Viewpoint Modeling
  - Most viewpoint needs covered by the provided activity diagram elements and some OCL[1] constraints
  - Difficult access on tagged values of custom stereotypes (e.g. for filtering)
- Viewpoint Library
  - Providing standardized viewpoints to quickly create consistent results
  - Few modelers need to create or know how to create viewpoints
  - Allowing pre-planned view hierarchies to define document structure and type of content prior to modeling
  - Included warning messages if no suitable elements are found
  - Requiring some modeling considerations to consistently find correct model elements to be exposed

Modeling Guidance

1) OCL = Object Constraint Language

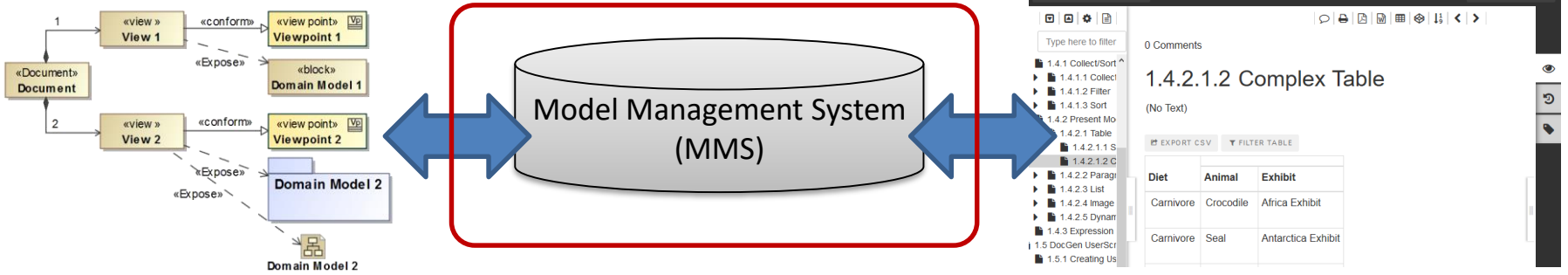
MDK: DocGen Modeling Guidelines

- Using a consistent model structure to allow model elements to be found by viewpoints
- Separate model content suited to be exposed and model content that should not get exposed
  - By using separate workspace package
  - By removing empty diagrams and unused model elements
- Keeping document creation in mind while modeling
  - Limiting the size of diagrams and tables, to be printable
  - Element (e.g. diagram) documentation to be used in documents

Model Management System (MMS)

- What is MMS?
  - —A version control system for structured data, including versioning, workflow management, and controlled access through RESTful web services
  - —Used as central data hub to facilitate multi-tool and multi-repository integration across engineering, computing, and management disciplines
- To store SysML model data
  - —Capturing all model elements (e.g. classes, instances, relations, but not: diagram layout), including their change history and views for View Editor

MDK: View and Viewpoint Hierarchy

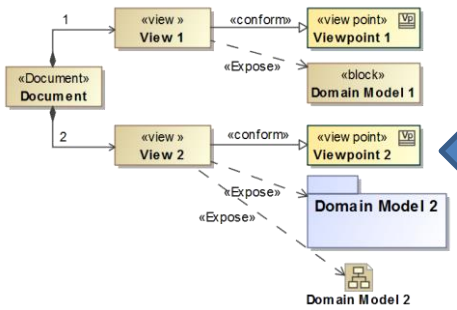


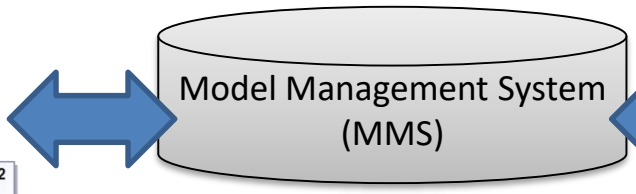
View Editor: Provides Rich Web Interface

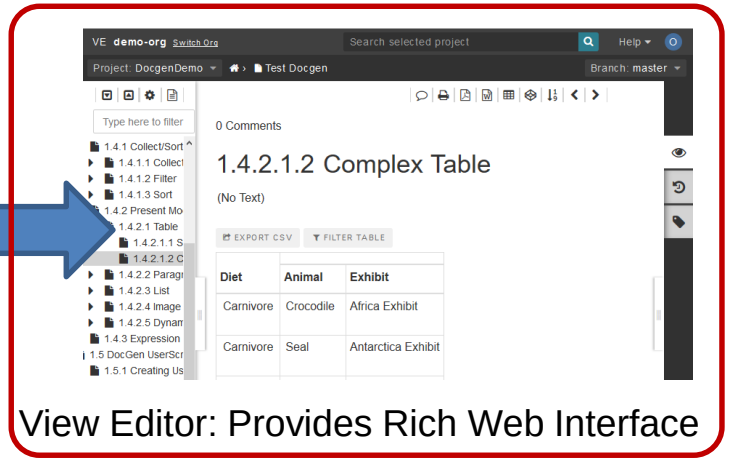
View Editor

- What is the View Editor?
  - A web app to provide consistent data from SysML models, to allow interaction with model elements without having to use the case tool
  - By utilizing exposed data from DocGen views and viewpoints through view instances and model elements on MMS
- To improve communication with non-modelers via live data in stakeholder-specific documents from the AST

MDK: View and Viewpoint Hierarchy

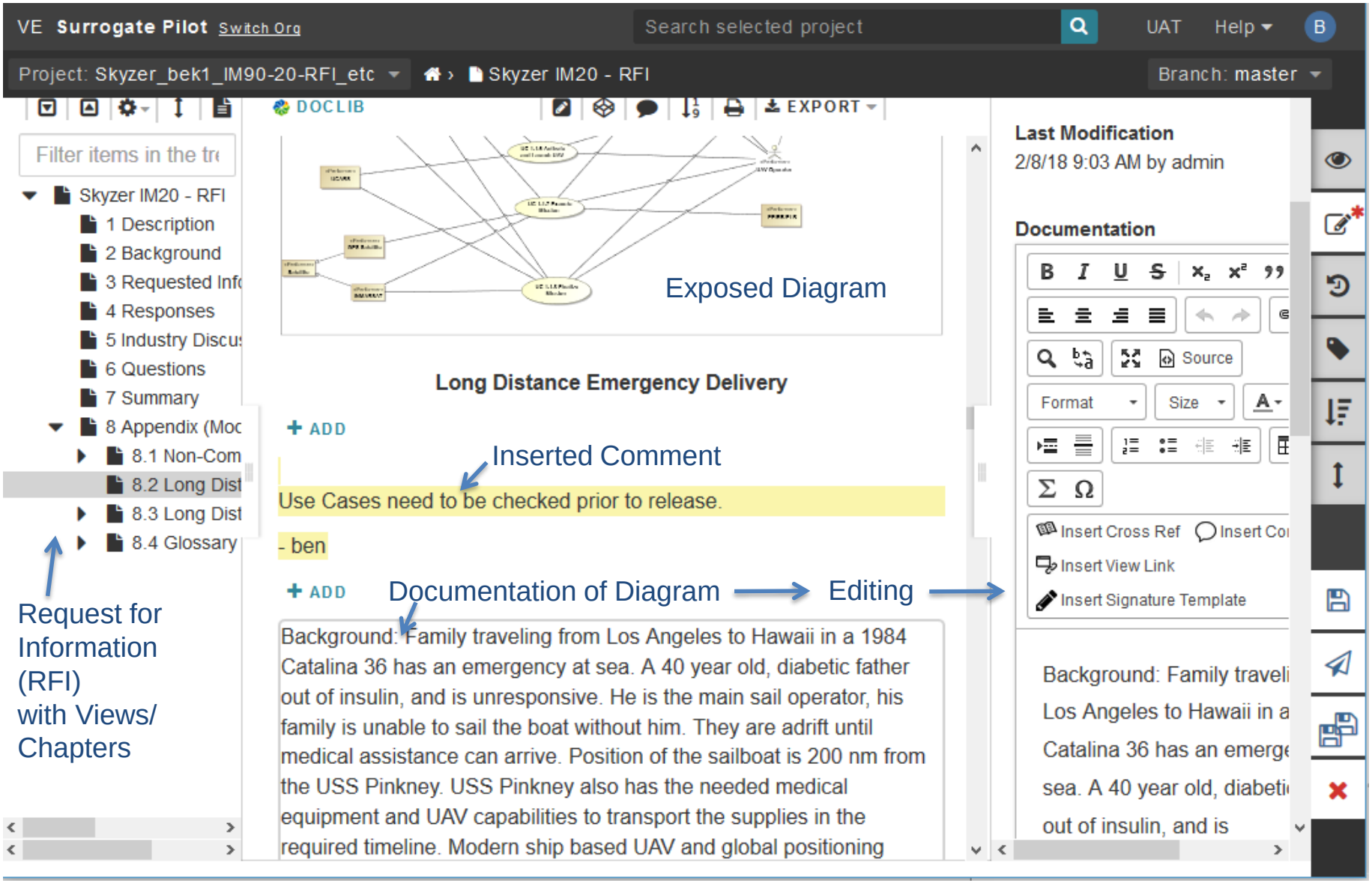






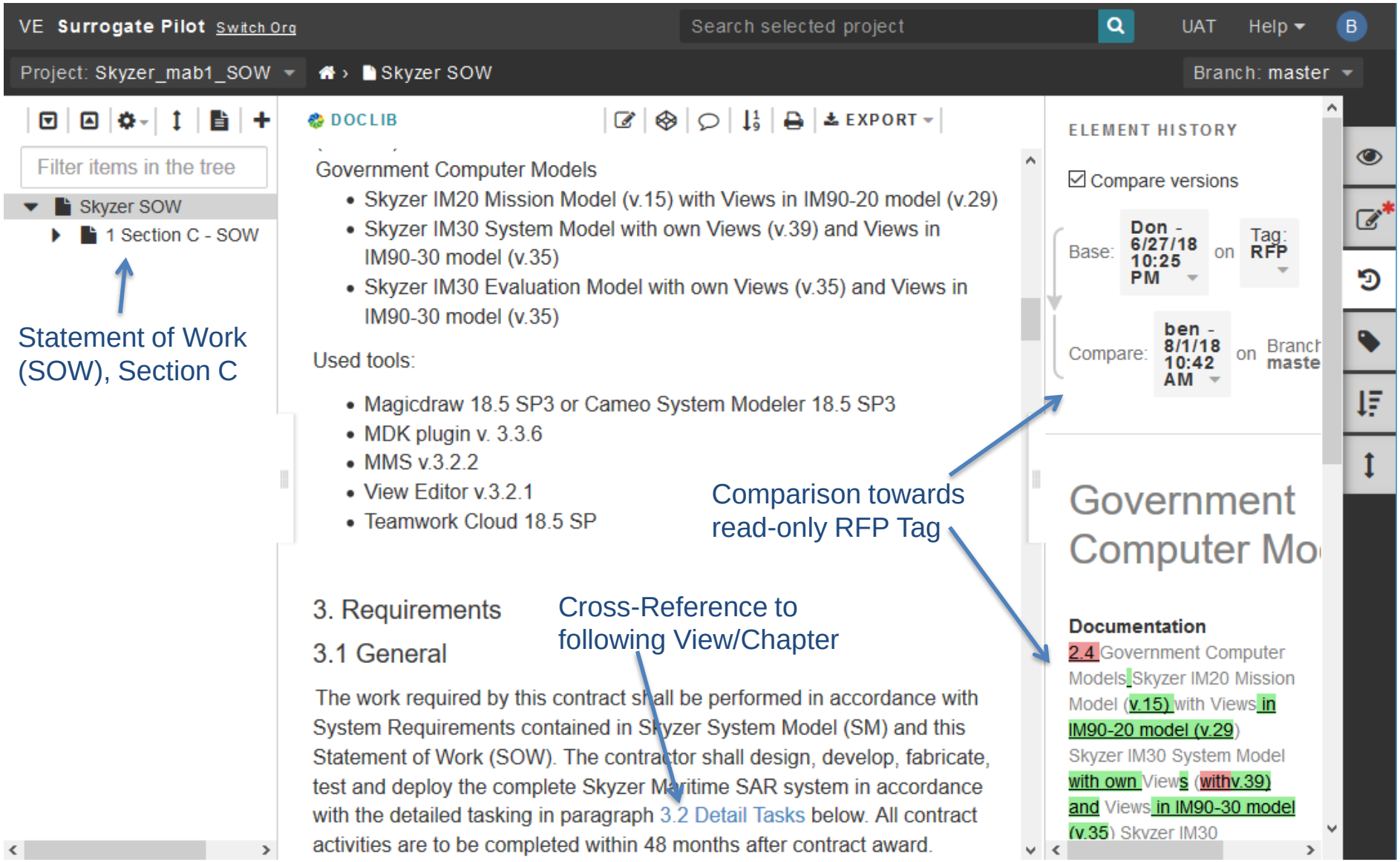
View Editor: Provides Rich Web Interface

View Editor: Commenting & Editing



Request for Information (RFI) with Views/ Chapters

View Editor: History Comparison



View Editor: Lessons Learned (1)

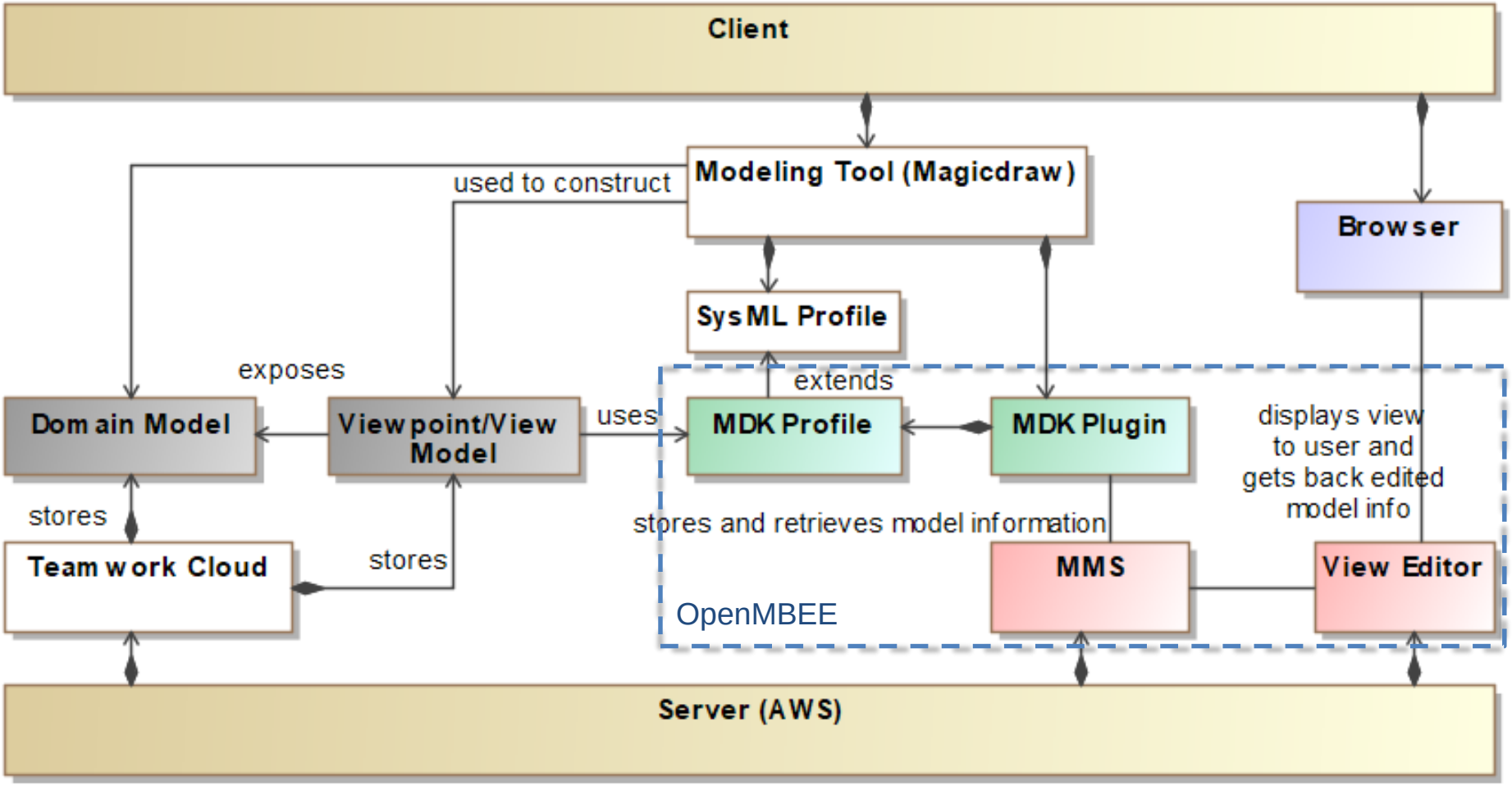
- View Editor editing capability
  - —Addition of presentation elements (e.g. text, images, videos) and comments
  - —Editing names, values and documentation of SysML model elements
  - —Limited creation of new model elements
    - Instead possible to adapt placeholder elements
- Traceability
  - —Searching and cross-referencing of any other elements in MMS
- Creation of read-only Tags in the View Editor
  - —E.g. to capture official RFI and RFP releases
  - —TWC branches committed from SysML modeling tool

View Editor: Lessons Learned (2)

- Model-based document generation from AST
  - Access on consistent AST data without modeling tool or SysML knowledge
    - Enabling transition from document-based to model-based development
    - Enabling fast and useful design iterations
    - Glossary feature of modeling tool to define terms in MMS/View Editor
  - Flexible application of model-based document generation
    - Content created in SysML tool (e.g. textual SOW or System Model) as well as in the View Editor (e.g. most of RFI)
  - Include domain experts & stakeholders to define relevant views/documents
    - Model-generated reports supporting program leadership decisions by providing only relevant information to stakeholders
    - Only modelers will likely know/understand entire model

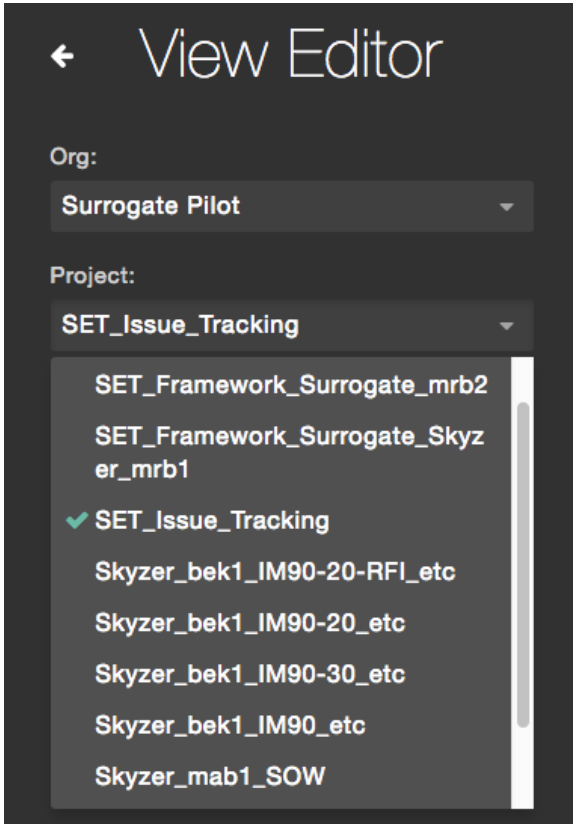
OpenMBEE Implementation

- OpenMBEE Implementation in Docker on AWS:



Project Usage & Management

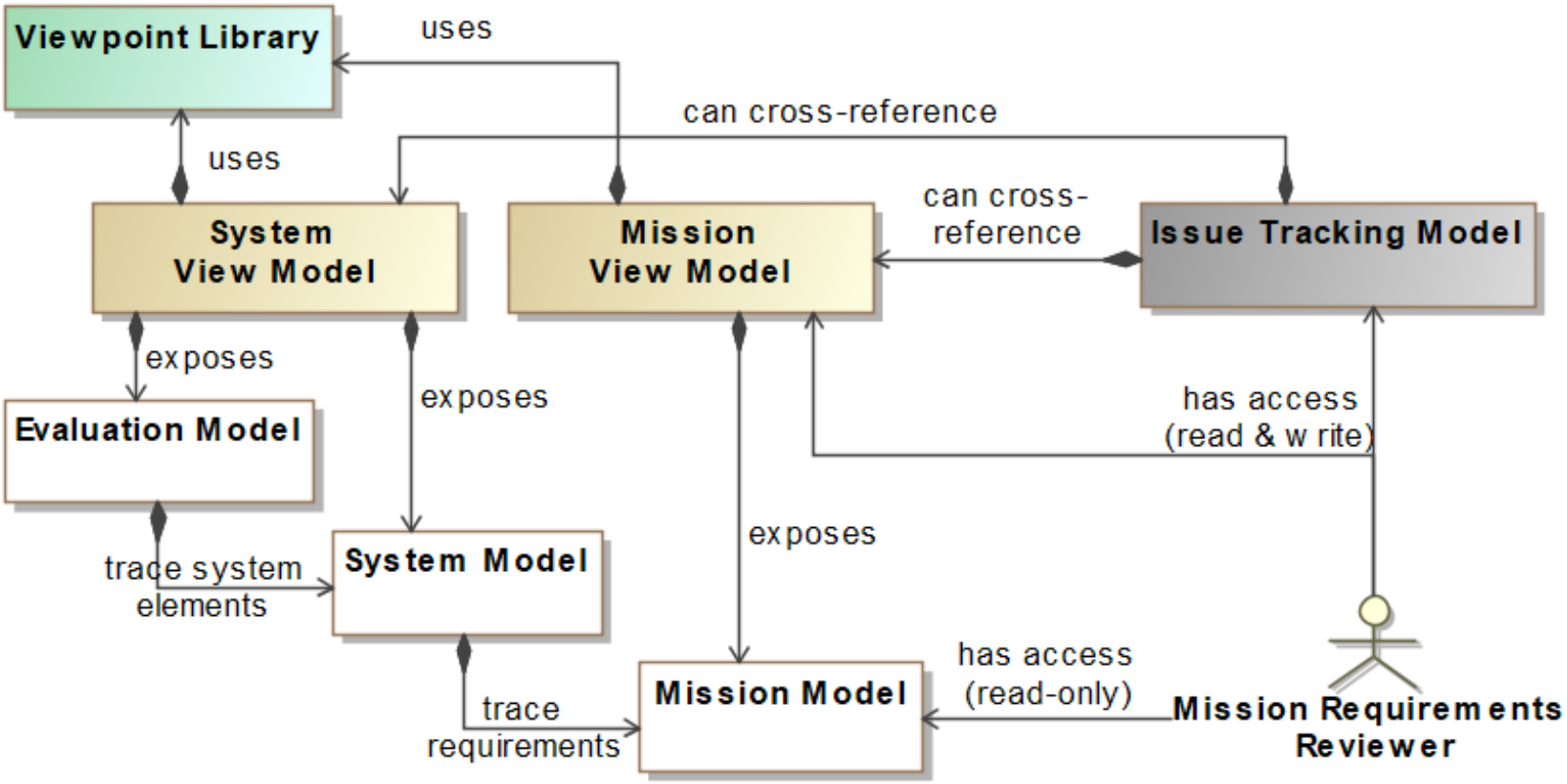
- Project Usage Mechanism
  - —For project modularization to separate distinct parts of the system
    - E.g. internal Evaluation Model separate from provided System Model
  - —Allowing traceability linkages
    - E.g. System Model using Mission Model to trace mission requirements to system model elements
  - —To reuse model libraries (e.g. viewpoints)
    - While hiding used profiles in separate MMS Org



- Issue Tracking in the View Editor
  - —Issues captured in the View Editor and saved in MMS as AST
  - —Allows to cross-reference model elements from used projects
    - E.g. directly using comments in other documents as issues

Project Usage & User Permissions: Example

- Editing & commenting in the Mission View Model, without permission to directly change exposed requirements from Mission Model
- Creating issues in Issue Tracking Model, e.g. using prior comments



Composition ≙ Project Usage

Summary & Future Work

- Promising application of OpenMBEE as an AST environment
  - —Technically feasible to develop everything as a model
    - Project usage for modularization, reuse, partitioning, traceability and user access
    - Quick and consistent model-based document generation through DocGen & Viewpoint Library
    - Enabling fast and useful design iterations
  - —View Editor offering views on consistent data from MMS as AST
    - Improved communication between modelers, experts and other stakeholders
    - Included issue tracking with AST access
- Future Work
  - —Model-centric source selection for RFP response including traceability towards non-SysML data, e.g. multi-physics simulations
  - —Adoption of GitFlow[1] workflow to define and standardize model branching
  - —Addition of ontological reasoning on AST data

1) https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

Further Information

- OpenMBEE http://www.openmbee.org/
  - —GitHub: https://github.com/Open-MBEE
  - —OpenMBEE Public Server Information: https://github.com/Open-MBEE/open-mbee.github.io/wiki/OpenMBEE-public-server-information
  - —Google Group: https://groups.google.com/d/forum/openmbee/
- Surrogate Pilot
  - —Apan SET Surrogate Pilot Group: https://community.apan.org/wg/navair-set/set-surrogate-pilot/
  - —View Editor: http://ime.sercuarc.org/alfresco/mmsapp/mms.html (Login instructions available on Apan)

References

- Blackburn, M., et al.: *Transforming Systems Engineering through Model-Centric Engineering*. Stevens Institute of Technology, Systems Engineering Research Center, 2018. # SERC-2018-TR-103.
- Delp, C., et al.: *Model Based Document and Report Generation for Systems Engineering*, in *Aerospace Conference*. 2013, IEEE.
- Department of Defense: *Digital Engineering Strategy.* Office of the Deputy Assistant Secretary of Defense for Systems Engineering, 2018. www.acq.osd.mil/se
- Zimmerman, P., T. Gilbert, and F. Salvatore: *Digital Engineering Transformation across the Department of Defense.* The Journal of Defense Modeling and Simulation, 2017. doi: 10.1177/1548512917747050

Thank you!

Dr. Benjamin Kruse
Research Assistant Professor
School of Systems & Enterprises
Systems Engineering Research Center
Stevens Institute of Technology

Use Cases for Surrogate Pilot

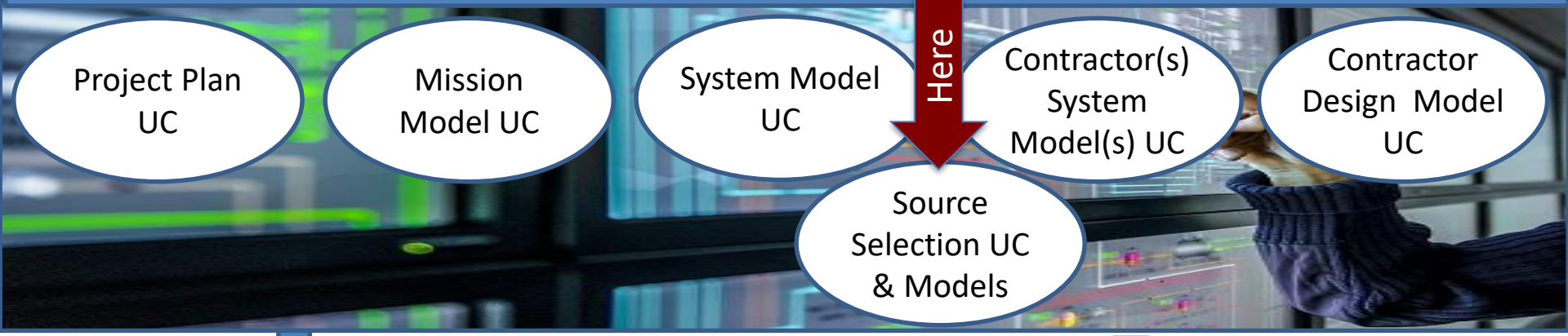
Surrogate Pilot Use Cases (UC)

Collaboration in Authoritative Source of Truth UC

Objectives to Assess SE Framework

How we Collaborate in AST

Surrogate Pilot using Authoritative Source of Truth (AST)



Mission Models
System Models
Evaluation Model
Based on Standards

RFI

RFP

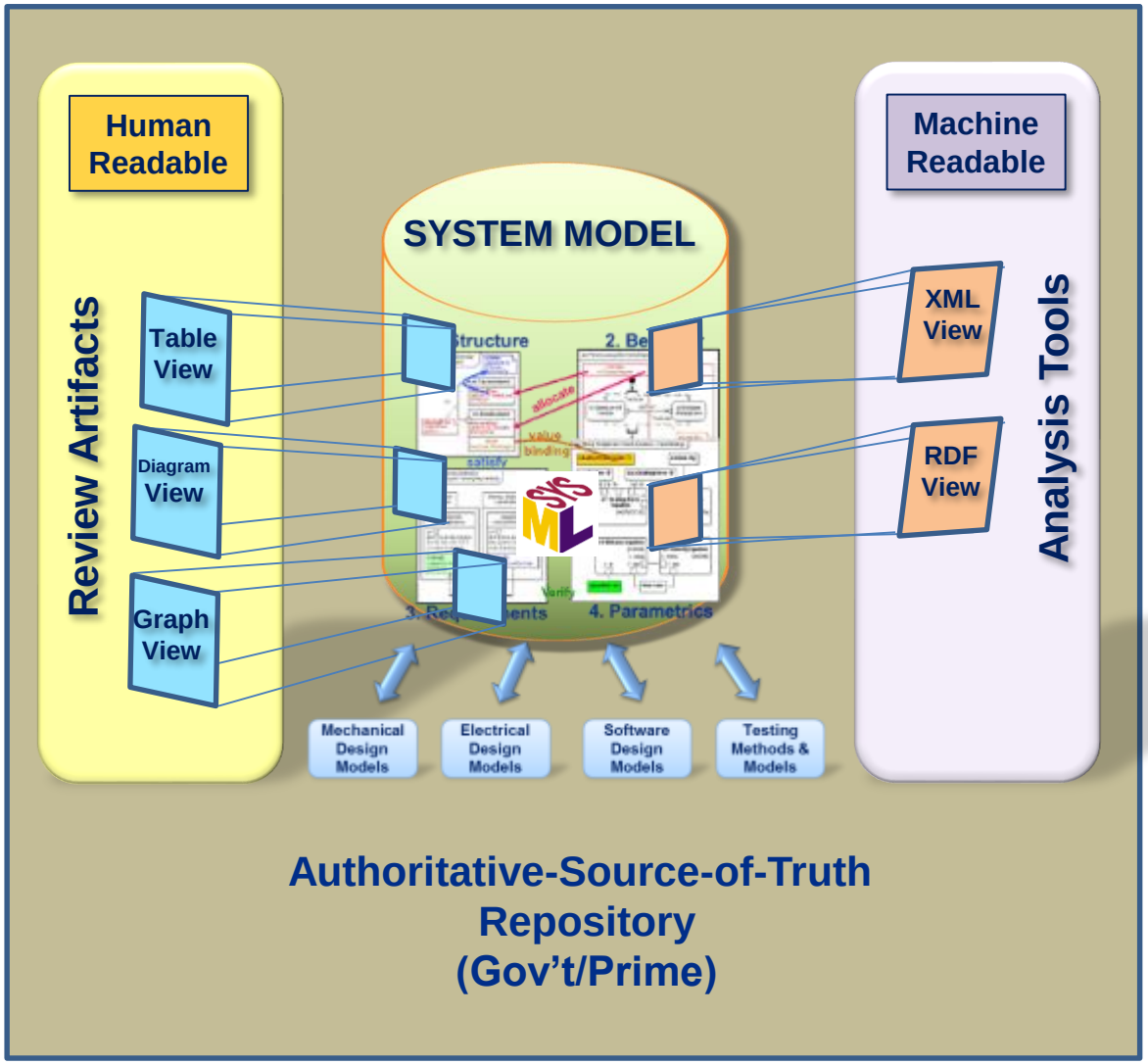
GFI

Proposal Extends GFI

Proposal for Design Models must be able to demonstrate aspects for Producability Decisions involving Multi-physics

SYSTEMS ENGINEERING
Research Center

~~Single~~ Authoritative Source of Truth

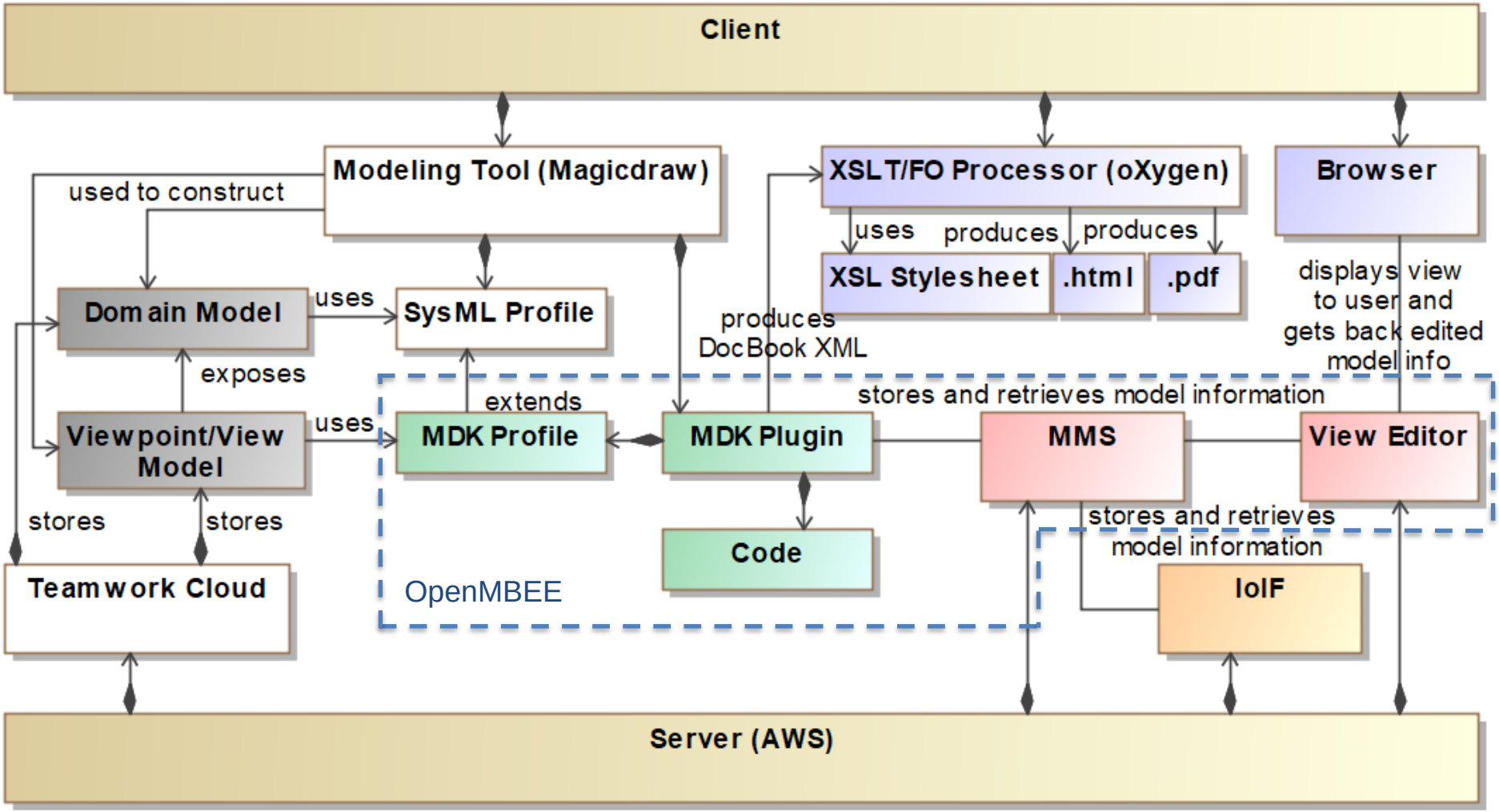


The entire set of models and tools is held in a single repository and becomes the Single-Source-of-Truth for the duration of system development

- Ability to *interrogate the design information* and extract data into the format necessary for the given task
  - Leverages formalism
  - Transformation rules are *reusable*
  - Provides *machine and human readable formats*
- Leverage the model by reviewing *the model itself*
- Stakeholders *focus on the views of the system model* that address their concerns

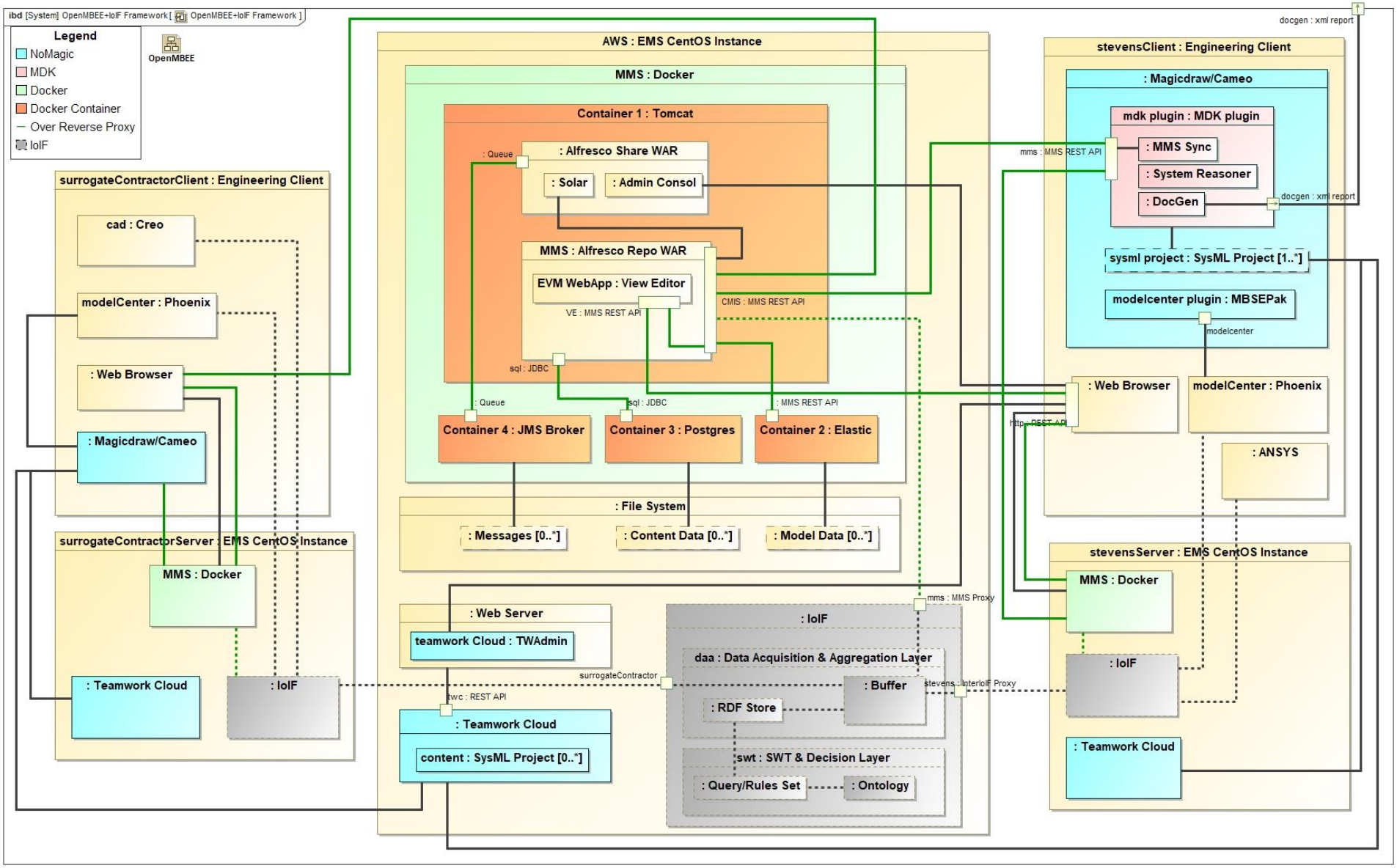
OpenMBEE Implementation

- OpenMBEE Implementation in Docker on AWS:



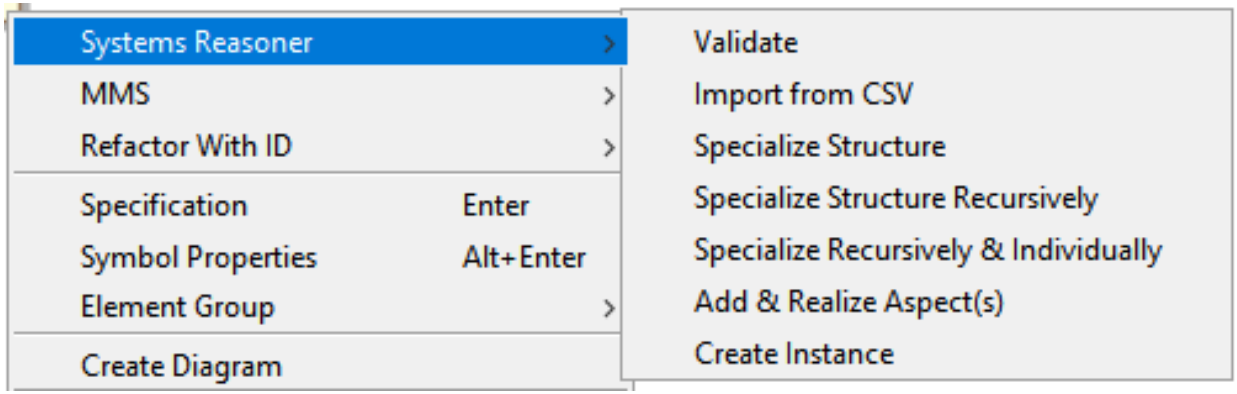
SYSTEMS ENGINEERING
Research Center

OpenMBEE + IoIF Implementation



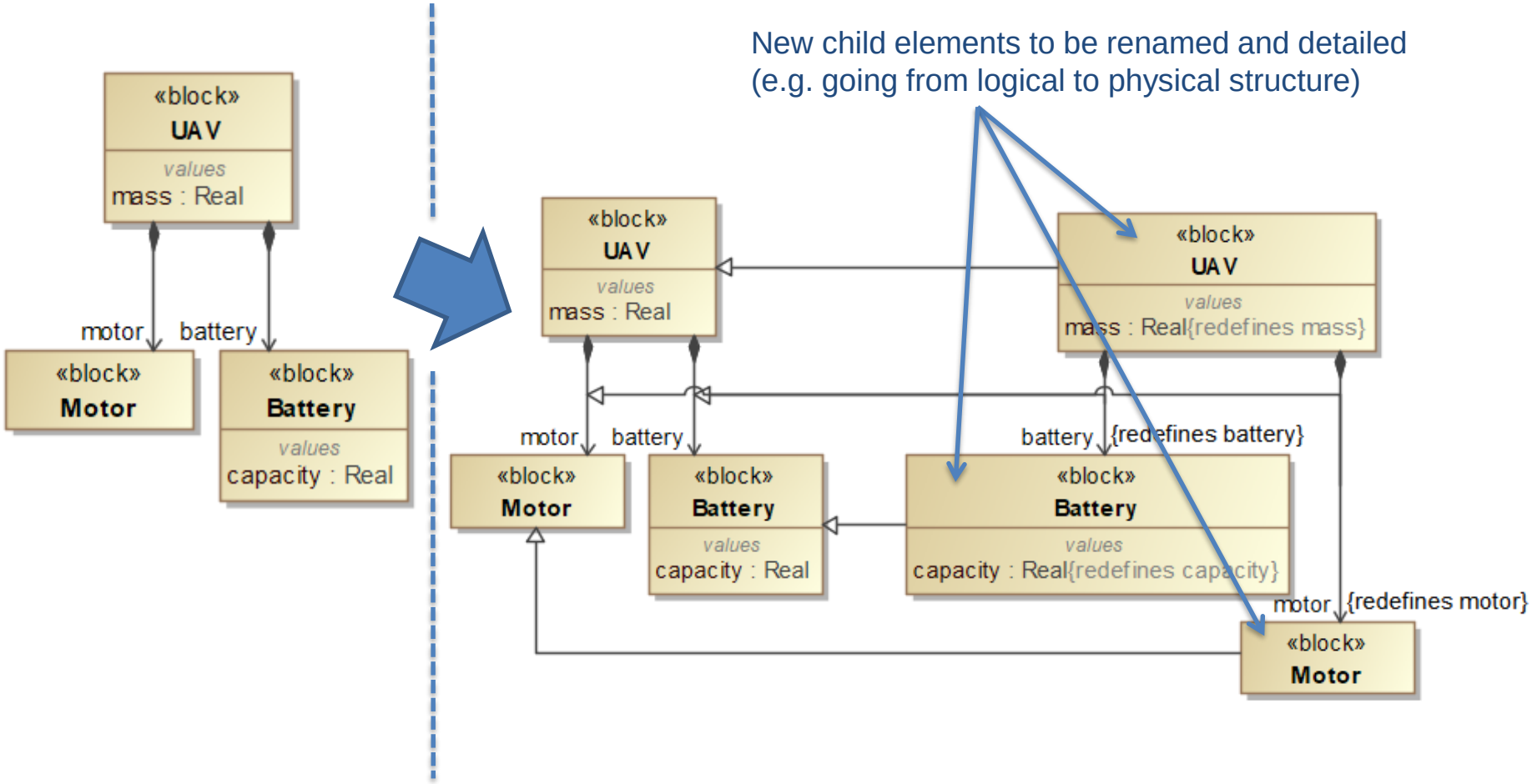
MDK: Systems Reasoner

- Purpose: Support building system assemblies in SysML
  - —Augmented creation of child elements from imported CSV tables
  - —Augmented creation of specialization trees
    - Including propagation and redefinition of properties
  - —Aspect creation to avoid over-use of stereotypes with tagged values
  - —Model validation



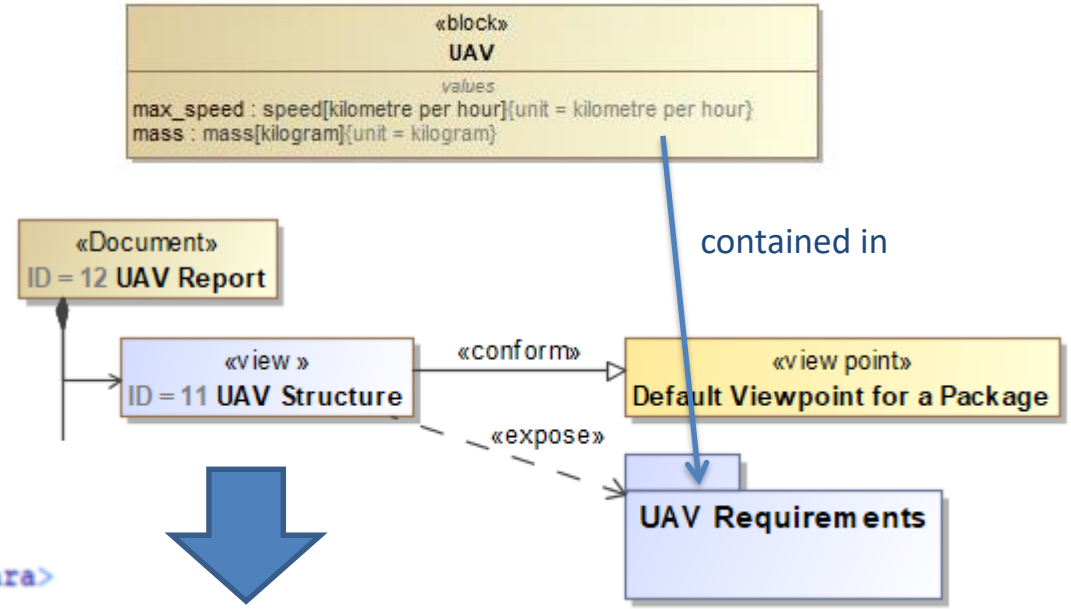
MDK: System Reasoner Example

- "Specialize Structure Recursively": modeling support to create sub-elements together with redefined properties and interrelations



MDK: DocGen Example

- Model: UAV block
- View Hierarchy:
- Result:
  - —XML file:




```
<tbody>
    <row>
        <entry><para>UAV</para>
        </entry>
        <entry><para>The unmanned air vehicle (UAV) has to navigate without a human
                pilot on board.</para>
        </entry>
    </row>
```

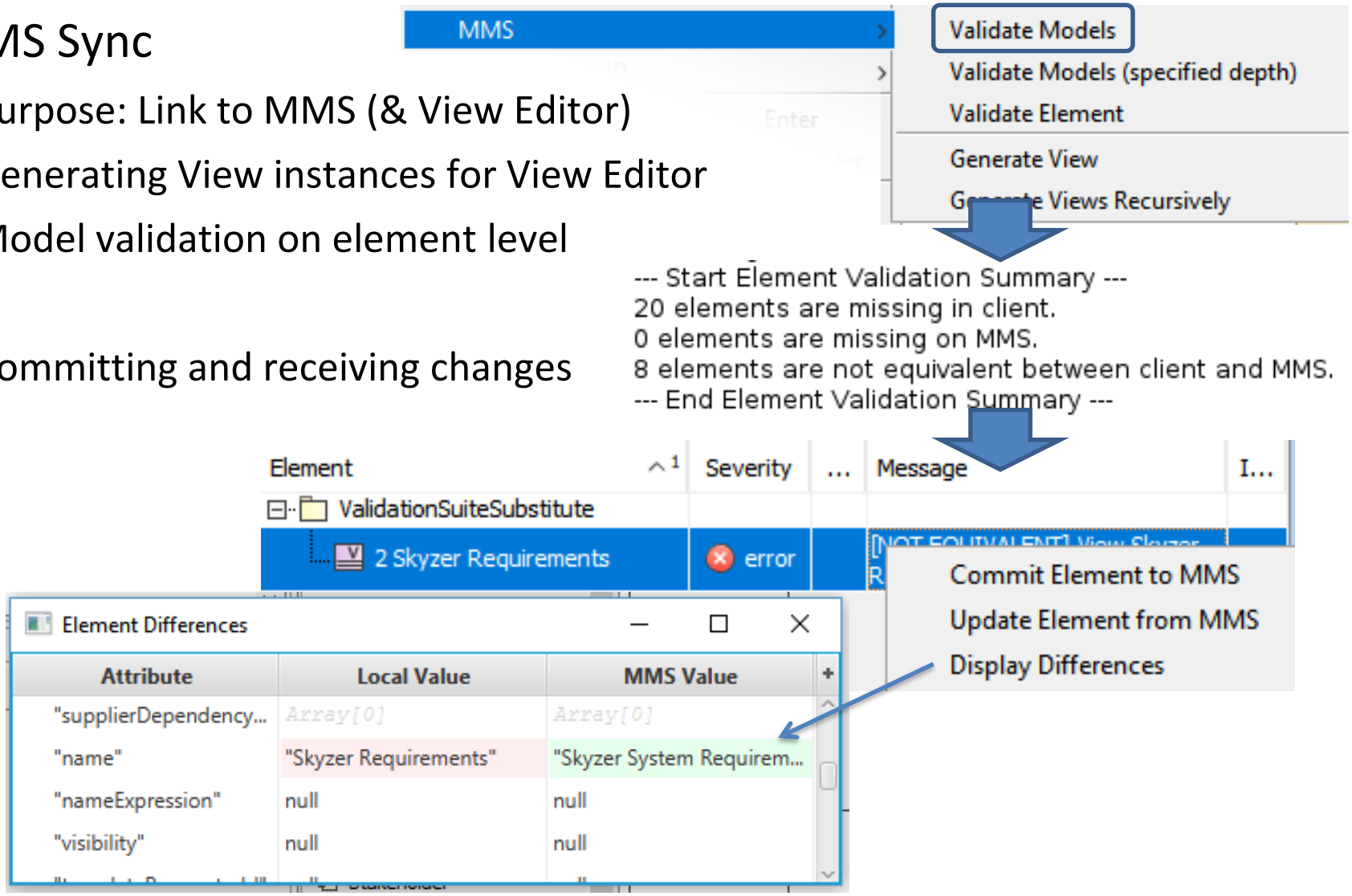
  - —Converted PDF:

Table 1.1. UAV Structure

Model Element	Documentation
UAV	The unmanned air vehicle (UAV) has to navigate without a human pilot on board.

MDK: MMS Sync

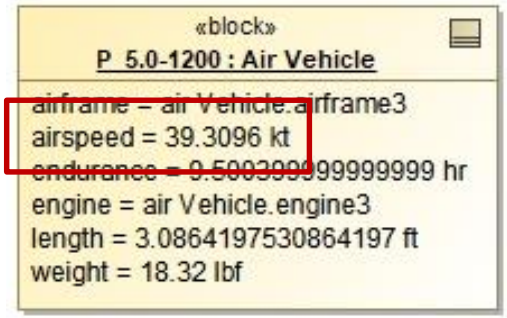
- MMS Sync
  - Purpose: Link to MMS (& View Editor)
  - Generating View instances for View Editor
  - Model validation on element level
  - Committing and receiving changes



SYSTEMS ENGINEERING
Research Center

MMS: Model Data Example (.json)

Element in Magicdraw:




```
"elements": [
  {
    "_modifier": "admin",
    "documentation": "",
    "mdExtensionsIds": [],
    "appliedStereotypeInstanceId": null,
    "_editable": true,
    "type": "Slot",
    "ownerId": "_18_5_2_8db028d_1521039686311_347855_34924",
    "_commitId": "1191cce7-58a3-4d5e-a309-550d83ab7851",
    "_inRefIds": [
      "master"
    ],
    "_creator": "admin",
    "syncElementId": null,
    "_created": "2018-03-16T19:38:42.810+0000",
    "id": "_18_5_2_8db028d_1521039686311_347855_34924-slot-_18_5_2_8db028d_1521039686144_127895_33281",
    "_elasticId": "5bbdfb63-f1d0-462a-a5ef-a4507954a33b",
    "value": [
      {
        "visibility": null,
        "documentation": "",
        "mdExtensionsIds": [],
        "appliedStereotypeInstanceId": null,
        "templateParameterId": null,
        "type": "LiteralReal",
        "ownerId": "_18_5_2_8db028d_1521039686311_347855_34924-slot-_18_5_2_8db028d_1521039686144_127895_33281",
        "clientDependencyIds": [],
        "syncElementId": null,
        "name": "",
        "typeId": null,
        "id": "_18_5_2_8db028d_1521039686311_347855_34924-slot-_18_5_2_8db028d_1521039686144_127895_33281-slotvalue-0-l
        "supplierDependencyIds": [],
        "value": 39.3096,
        "_appliedStereotypeIds": [],
        "nameExpression": null
```


"Slot" type element

Commit ID for element history

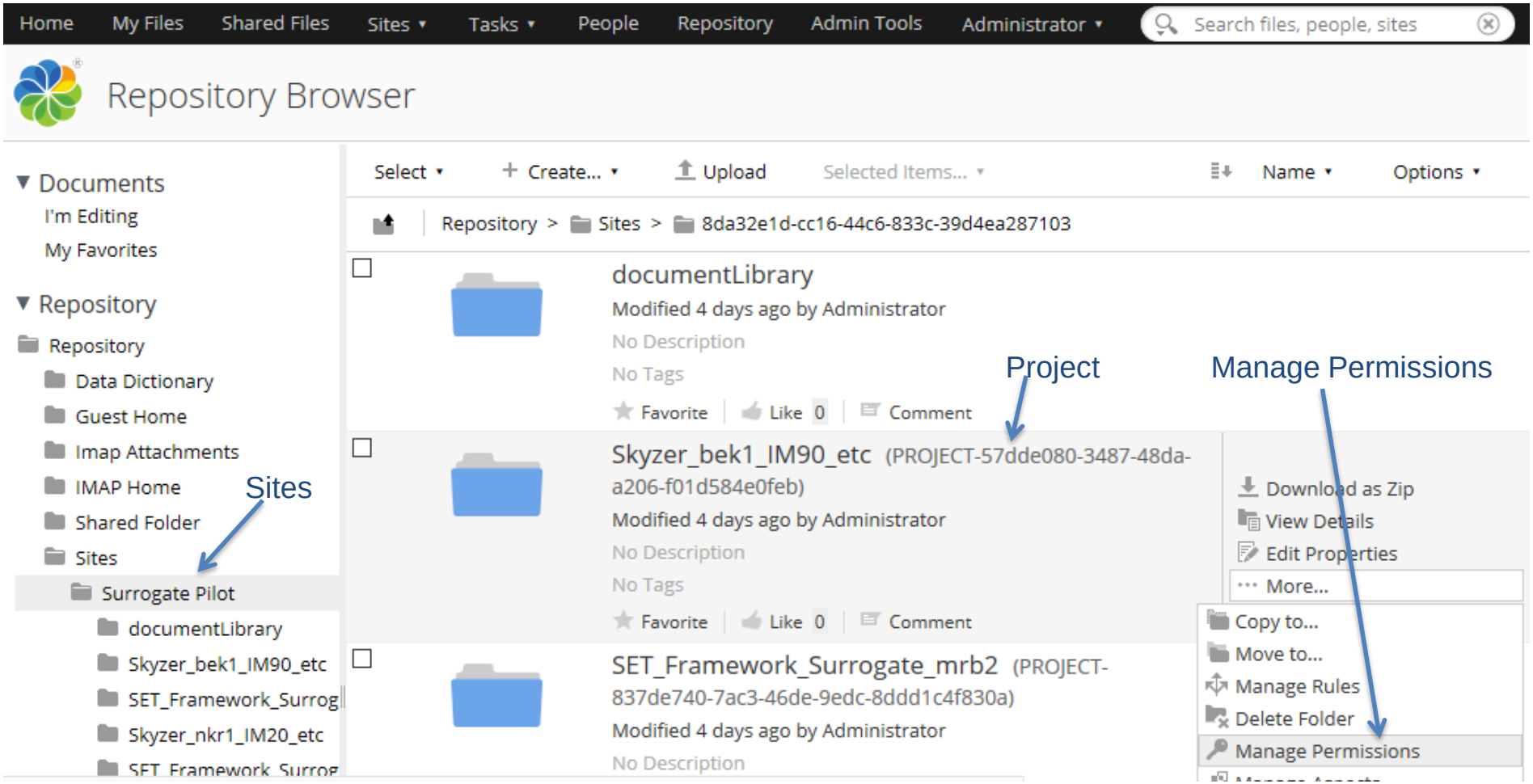
Element ID

Owner ID of "P_5.0-1200 : Air Vehicle" instance

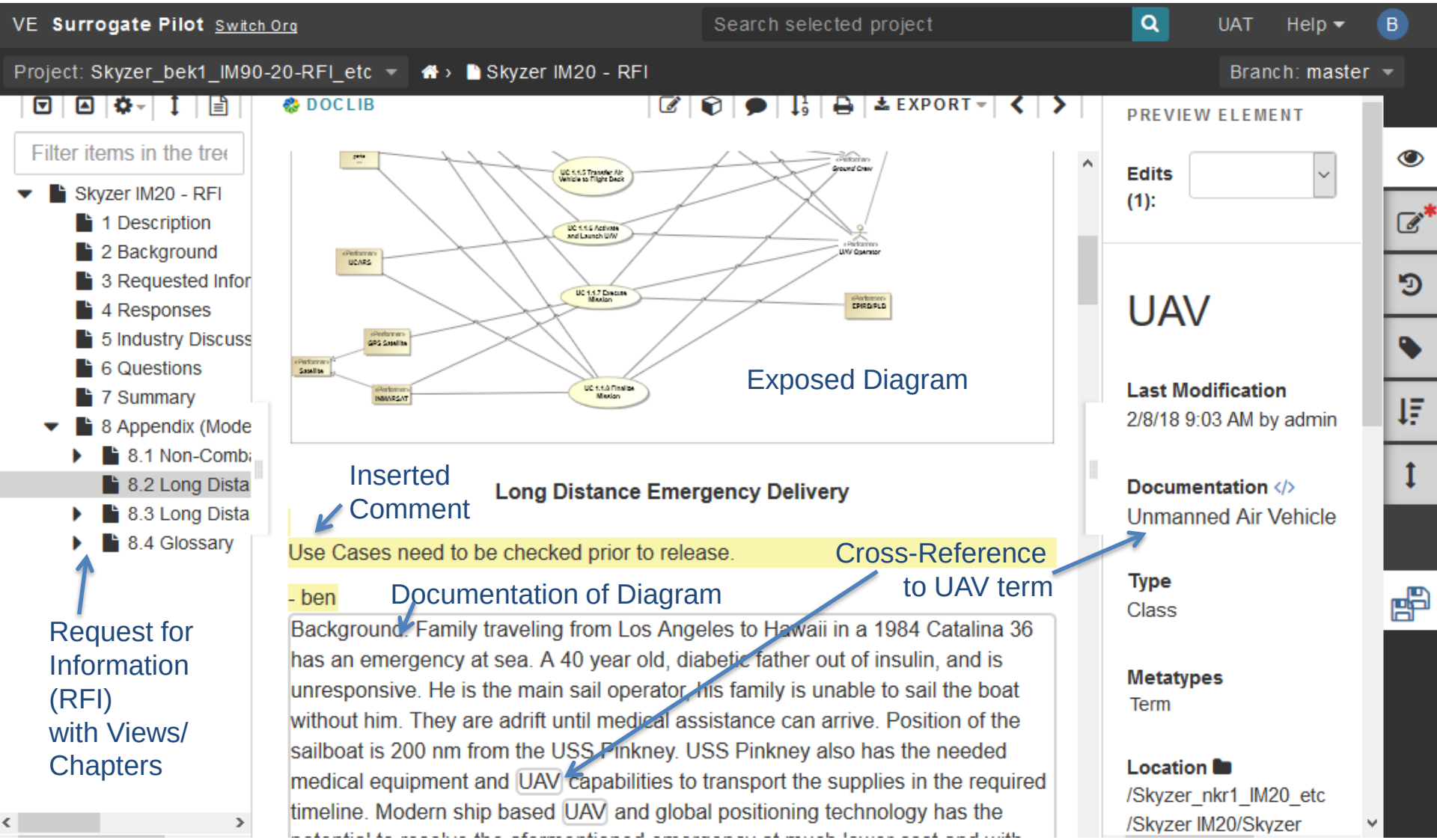
Value of slot element

MMS: Alfresco Content Manager

- For View Editor images and user access management:

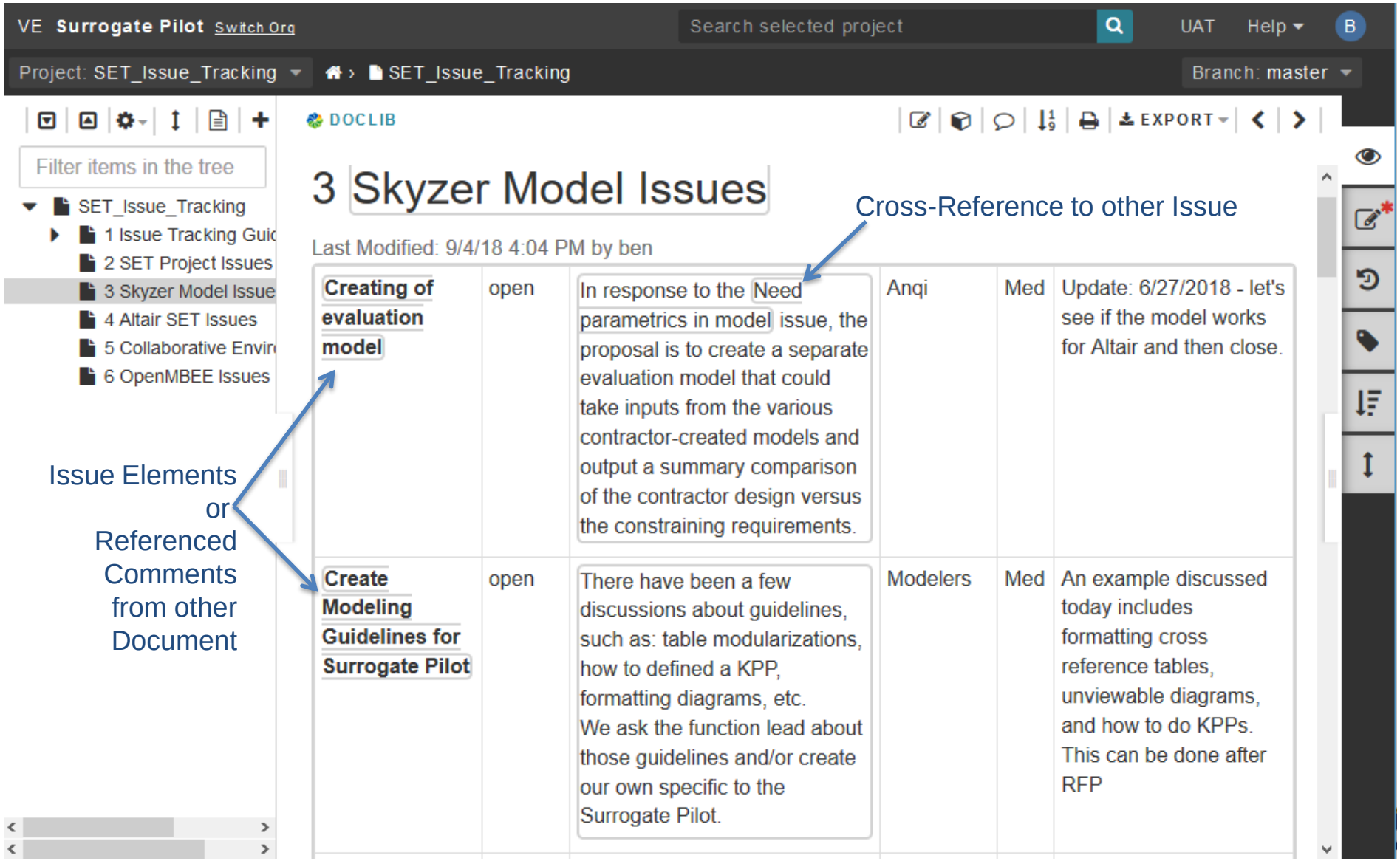


View Editor: Cross-referenced Glossary Term

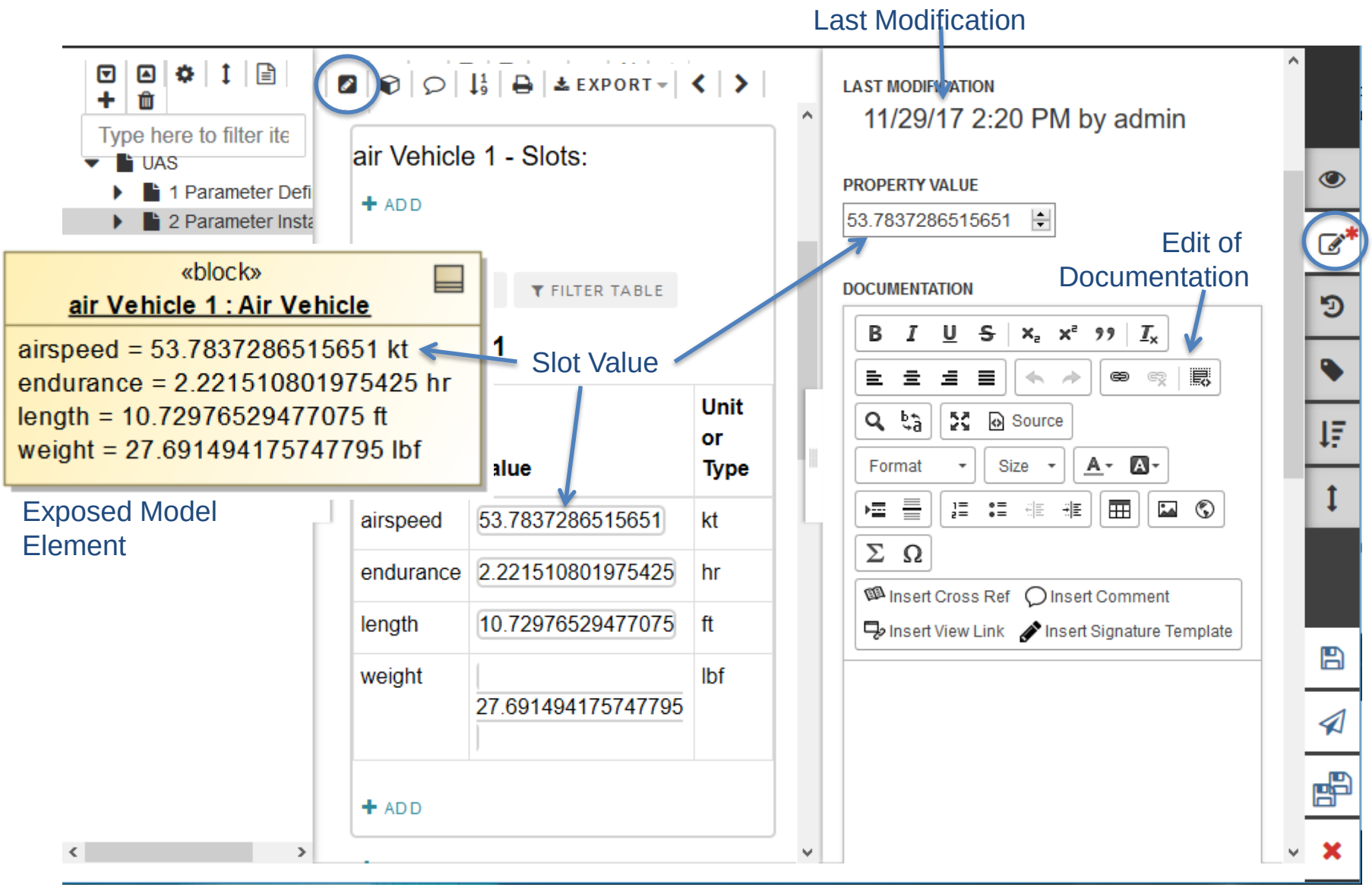


SYSTEMS ENGINEERING Research Center

View Editor: Model-based Issue Tracking

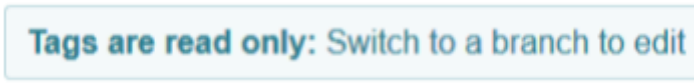


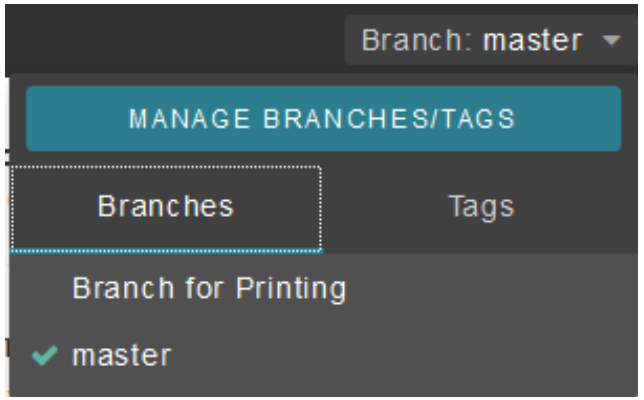
View Editor: Editing Exposed Data



View Editor: Branches and Tags

- Branches/Tasks (similar to GitHub branches)
  - —To create a separate workspace built by copying data at a specified time (any following changes do not affect the "master" branch)
  - —Able to merge with TWC Branches of the same name
    - Merging only if created in Cameo! Or merging of single elements through comparison
    - New branches still include added presentation elements from View Editor
- Tags (sets of permanently saved read-only data with a timestamp)
  - —To create "snapshots" of all the data on View Editor at specified times, e.g. for reviews



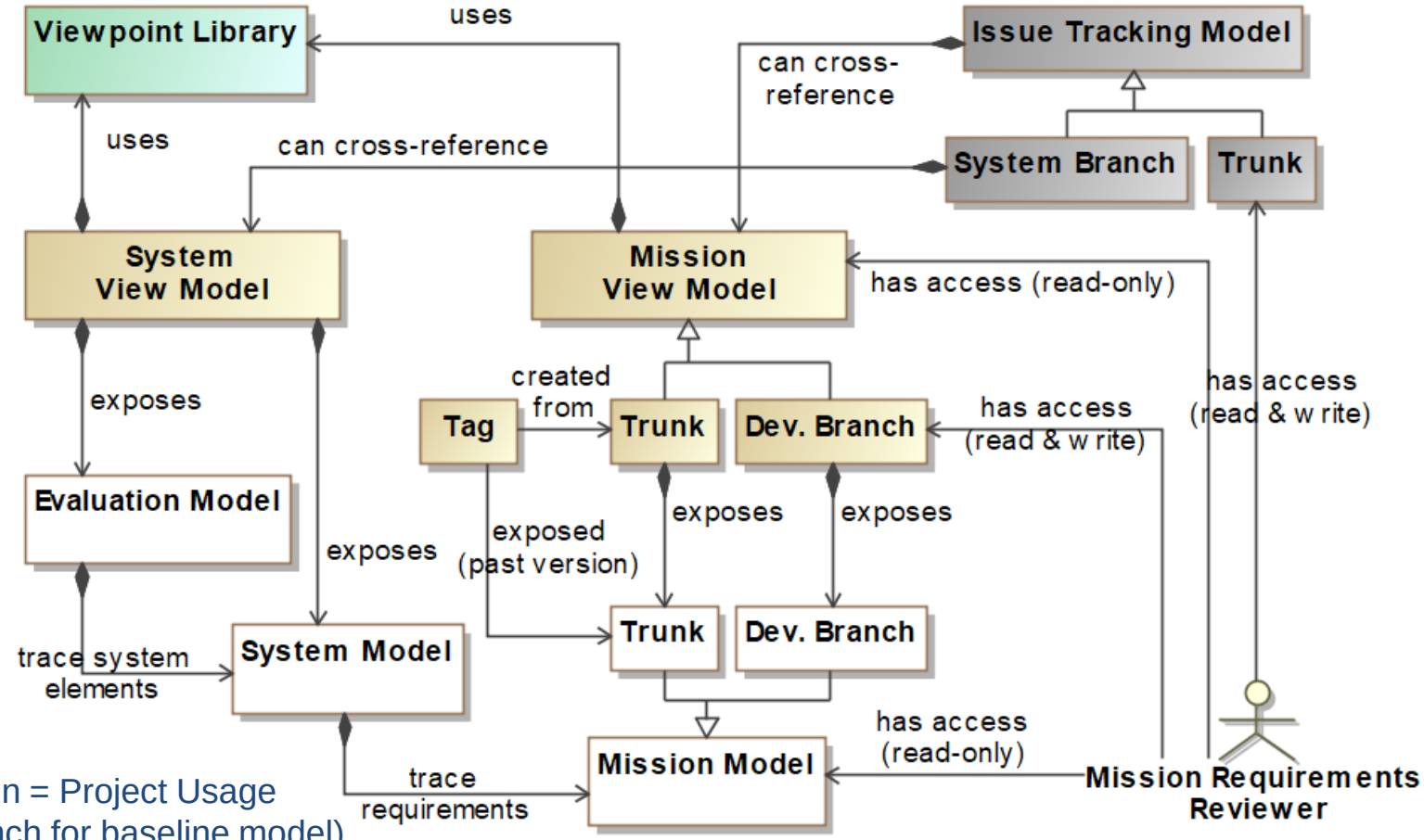


- Manage Branches/Tags
  - —To Create, Delete & Switch between
  - —Deleted branches leave data in Alfresco & cannot be recreated from Cameo

SYSTEMS ENGINEERING
Research Center

Project Usage & User Permissions: Example

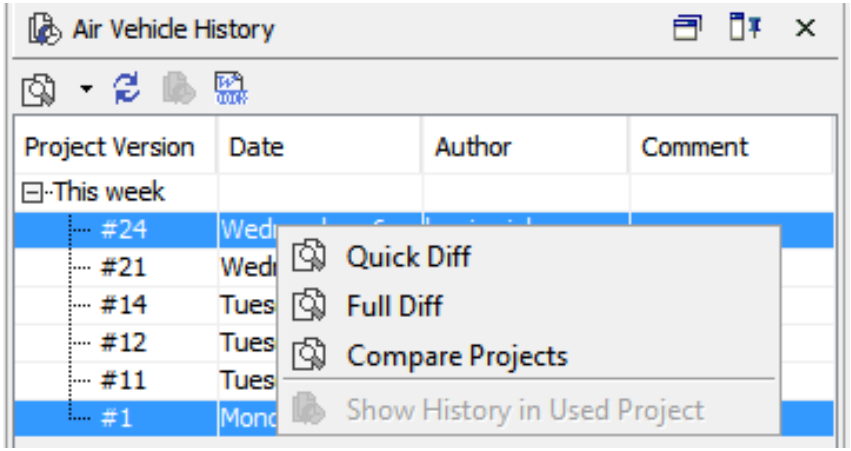
- Permissions to edit/comment in the Development Branch of the Mission View Model, without being able to directly change requirements in Mission Model
- Addition of comments as issues in Issue Tracking Model Trunk branch
- No access on any System Model and Evaluation Model related information



Composition = Project Usage
(Trunk branch for baseline model)

Teamwork Cloud (TWC)

- What is Teamwork Cloud?
  - —Server to work on the same Magicdraw/Cameo projects and merge the concurrent work of all modelers together
- Collaboration Features
  - —Editing server projects locally and commit changes/receive updates
  - —Locking elements for edit, to prevent simultaneous changes by other users
  - —Project and Element History (including comparison between two versions)
  - —Branching and Merging projects



MMS vs. TWC

MMS

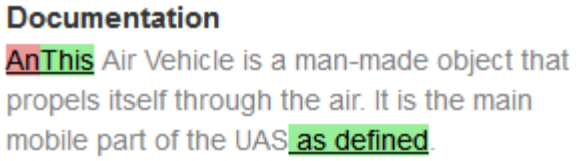
- Purpose: A central structured data hub for multi-tool, multi-repository, and multi-discipline integration
- Storing:
  - All model elements of a project, including their change history; view instances and tags for VE
  - Non-Magicdraw data
- Interfaces:
  - RESTful web services
  - Login through Magicdraw, VE and Alfresco (for user/model mgmt)

TWC

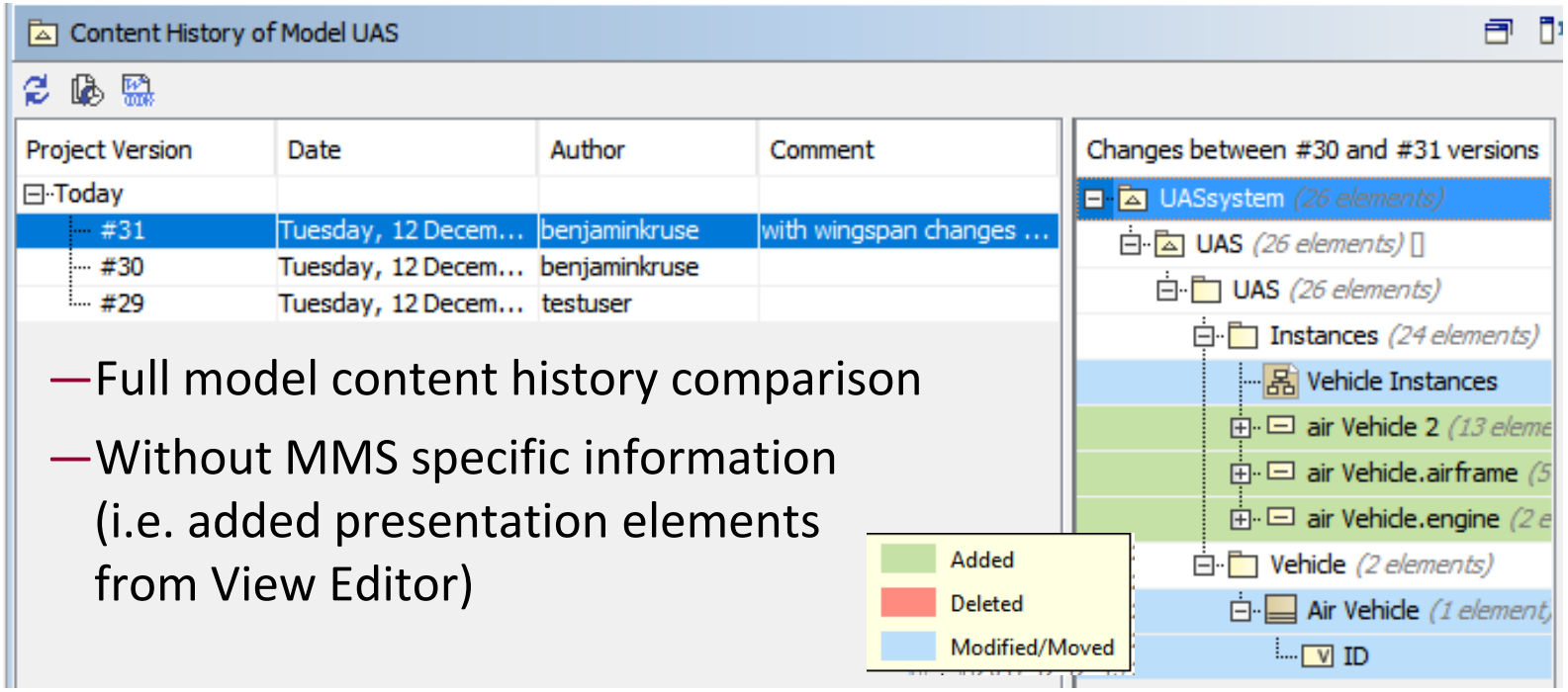
- Purpose: Concurrent & distributed Magicdraw modeling, including versioning and branches
- Storing:
  - Magicdraw projects, including their branches, model versions and element history (e.g. for merging branches)
  - (Collaborator data stored separately)
- Interfaces:
  - REST API (by NoMagic)
  - Login through Magicdraw and Admin Console (& Alfresco for Collaborator)

View Editor vs. TWC: History Comparison

- View Editor: For quick checks with limited scope
  - —History of single elements only
  - —Including all added presentation elements
  - —(Model validation in Cameo: open model vs. current state in MMS)



- TWC: For broad-scale merging of heavily changed models



  - —Full model content history comparison
  - —Without MMS specific information (i.e. added presentation elements from View Editor)