A Digital Engineering Factory for Students

Dr. Joe Gregory
Dr. Alejandro Salado
Prof. Sharon O'Neal

University of Arizona

Agenda

- DE for Students
- The Problem
- A Possible Solution (Semantic Hub and Spoke)
- Digital Engineering Factory
- Use Cases
- Example
- Future Work and Lessons Learned

Digital engineering is defined as:

> ‘‘an integrated digital approach that uses authoritative sources of systems’ data and models as a continuum across disciplines to support lifecycle activities from concept through disposal [1].’’

Digital engineering is defined as:

> ‘‘an integrated **digital** approach that uses authoritative sources of systems’ data and models as a continuum across disciplines to support lifecycle activities from concept through disposal [1].’’

Digital engineering is defined as:

> ‘‘an integrated **digital** approach that uses **authoritative sources** of systems’ data and models as a continuum across disciplines to support lifecycle activities from concept through disposal [1].’’

Digital engineering is defined as:

> ‘‘an integrated **digital** approach that uses **authoritative sources** of systems’ data and models as a continuum **across disciplines** to support lifecycle activities from concept through disposal [1].’’

Digital engineering is defined as:

> ‘‘an integrated **digital** approach that uses **authoritative sources** of systems’ data and models as a continuum **across disciplines** to support lifecycle activities from **concept through disposal** [1].’’

Digital engineering is defined as:

> ‘‘an integrated **digital** approach that uses **authoritative sources** of systems’ data and models as a continuum **across disciplines** to support lifecycle activities from **concept through disposal** [1].’’

Digital Engineering *for Students*

Digital engineering is defined as:

> ‘‘an integrated **digital** approach that uses **authoritative sources** of systems’ data and models as a continuum **across disciplines** to support lifecycle activities from **concept through disposal** [1].’’

Goal is to develop a ‘Digital Environment’ to support students during class and on projects, that can:

- support authoritative representation of system description and simulation,
- host tools to support students across all engineering courses,
- connect this data across courses (lifecycle phases).

Courses – Software Engineering (SFWE), Systems and Industrial Engineering (SIE)

Code	Name	Description	Tools
SFWE-301	Software Requirements Analysis and Test	Students learn how to develop high-quality software requirements, test plans and test procedures. Students also learn the importance of traceability.	**Requirements** **Test Strategy**
SFWE-302	Software Architecture & Design	Students learn how to build, test, and deploy software applications. Students will automate test cases and run static code analysis on their code.	**IDE** **Static Code Analyzer**
SFWE-403	Software Project Management	Students learn how to plan, manage and track the progress of an Agile software management project.	**Project Management**
SIE-431	Simulation Modeling and Analysis	Students learn how to conduct simulations on candidate system architectures. Discrete event simulation, model development, statistical design and analysis of simulation experiments, variance reduction, random variate generation, Monte Carlo simulation.	**Requirements** **System Architecture** **Physics Simulation** **Statistics Tool**
SIE-458	Model-Based Systems Engineering	Students learn how to use a SysML v1 modeling tool to exercise the systems engineering process.	**SysML v1**
SIE-498	Senior Design Project	The Senior Design Project pulls much of this together. Students use material taught across the curriculum to address a customer's needs and help a real-world client design or improve a system.	**Requirements** **Software Architecture** **Test Strategy** **Code Test**
…	…	…	…

Tool Examples

Software Development Tools

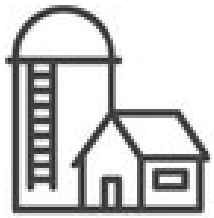
Requirements Analysis	SW Design and Modeling	Code and Unit Test	Continuous Integration	Verification / Validation	Continuous Deployment & Operations
Visual Paradigm	Enterprise Architect, MathWorks	GitHub, Visual Studio	Jenkins, Veracode	TestRail, klocwork	GitLab

Systems Development Tools

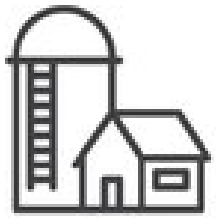
Requirements Engineering	System Architecture	Domain-Specific Analysis	Design and Manufacture	Trade-Offs and Optimization	PLM
jama	SysML v2, Genesys	Modelica	creo, SOLIDWORKS	GoldSim, jupyter	duro, Siemens, ptc

Teamcenter — Jira Software

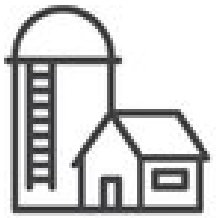
Silos in Engineering Education



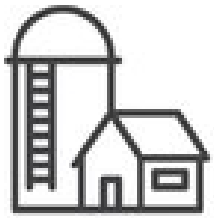
Software



Electronics



Aerodynamics

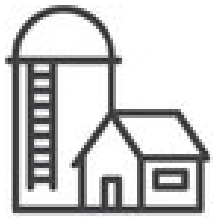


Mechanical

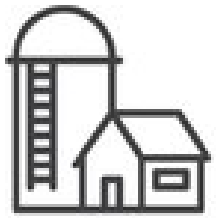
This is not good enough in industry!

But what do we do across engineering curricula?

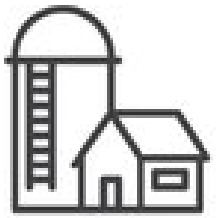
Silos in Engineering Education



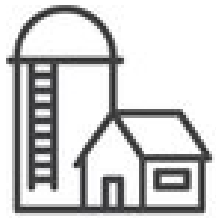
SIE 431



SFWE 302

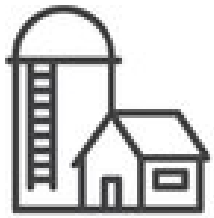


SIE 458

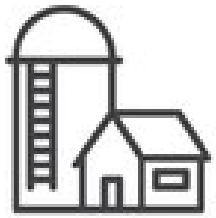


SFWE 403

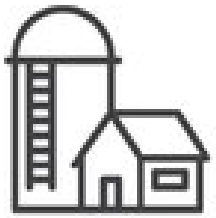
Silos in Engineering Education



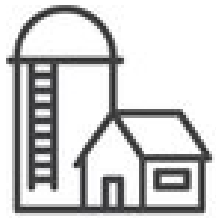
SIE 431



SFWE 302



SIE 458



SFWE 403

Engineering courses are **silos**

- Students do not observe effects downstream
- Students do not gain experience of working in a DE environment
- Data / effort duplication across courses
- There is limited scope for collaborative work

What are the Possible Benefits of DE in Engineering Curricula?

Students no longer submit homework, they **commit models**.

Students have access to data and tools through a single **integrated** environment.

Digital Engineering is **embedded** into the engineering curriculum.

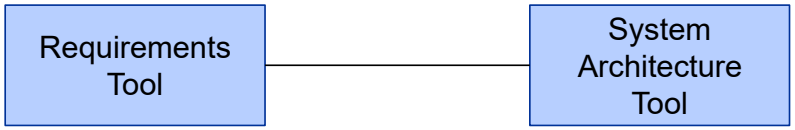
Potential **benefits** include:

- Over multiple courses, students see a complete **end-to-end** process,
- Students see the **consequences** of their decisions downstream,
- Students gain experience working **collaboratively** in a digital environment,
- Instructors (and students) can **evaluate / grade** work more effectively.

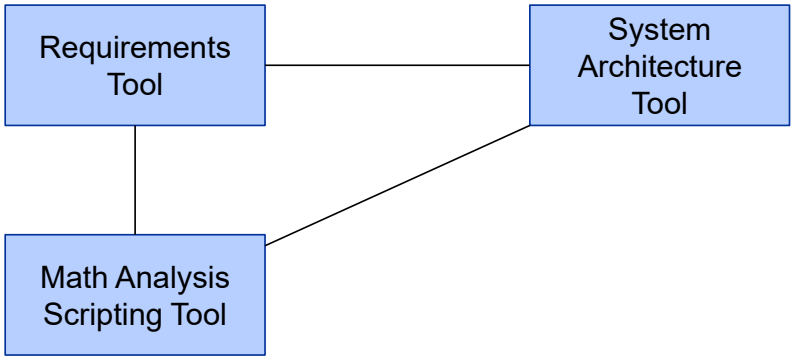
Tool Map

Requirements Tool

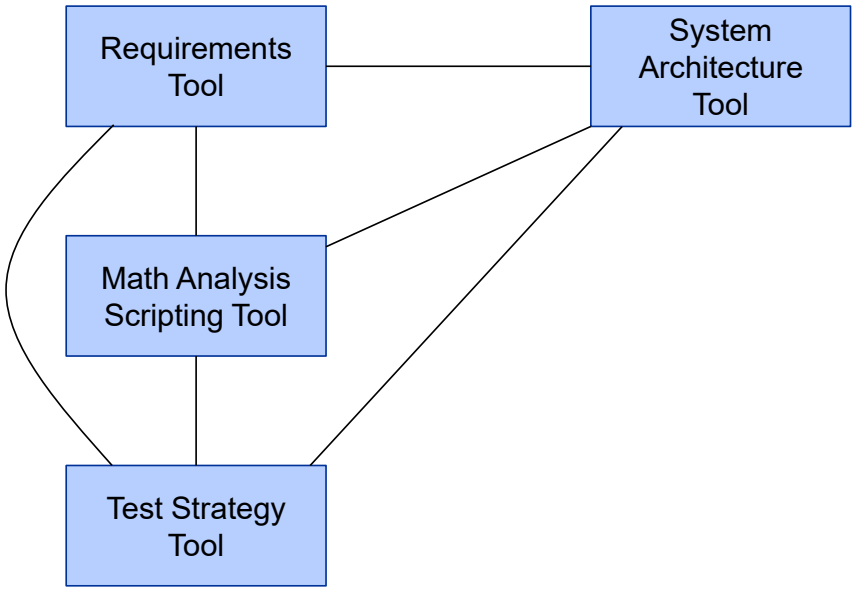
Tool Map



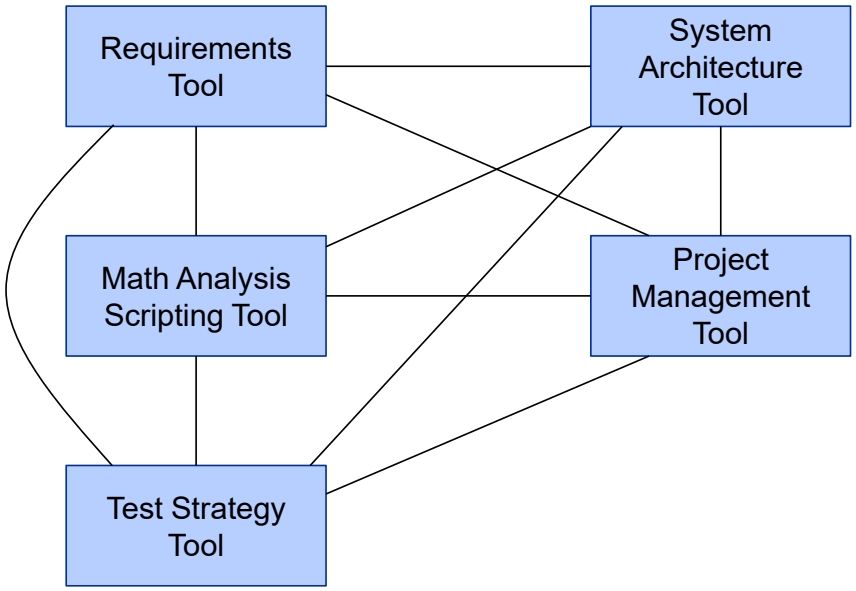
Tool Map


Requirements Tool
System Architecture Tool
Math Analysis Scripting Tool

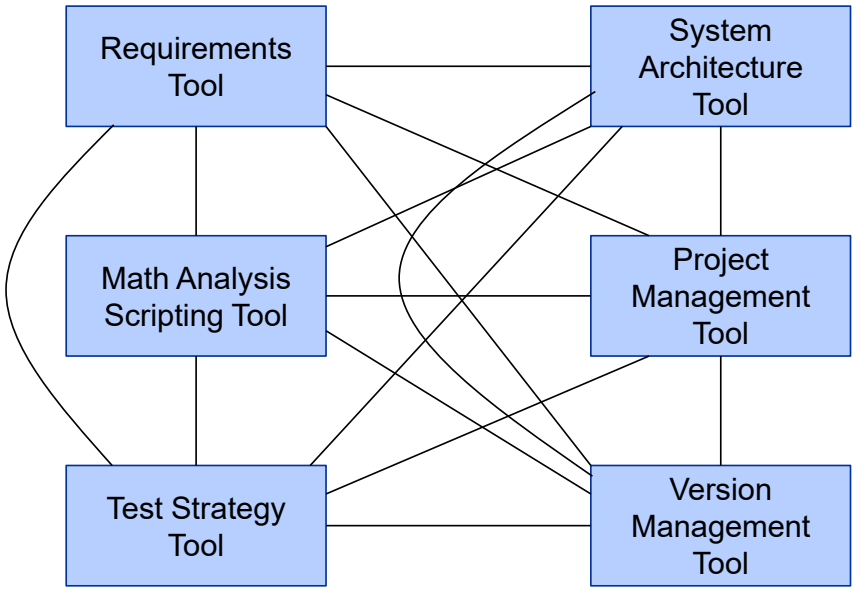
Tool Map


Requirements Tool
System Architecture Tool
Math Analysis Scripting Tool
Test Strategy Tool

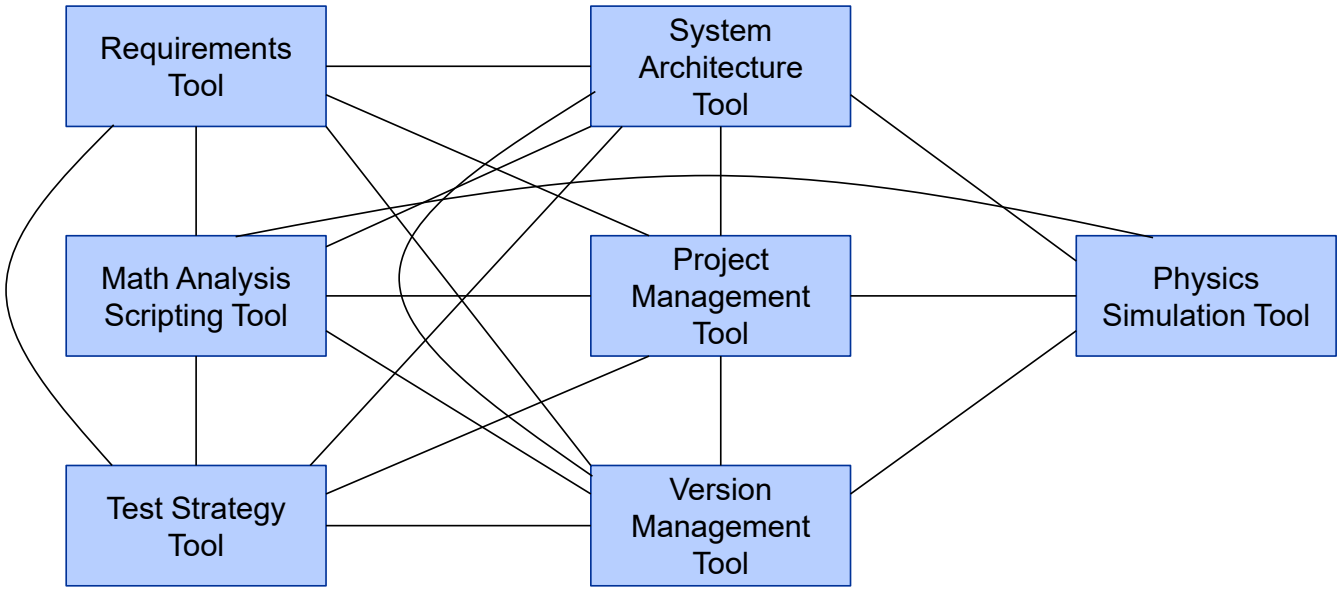
Tool Map


Requirements Tool
System Architecture Tool
Math Analysis Scripting Tool
Project Management Tool
Test Strategy Tool

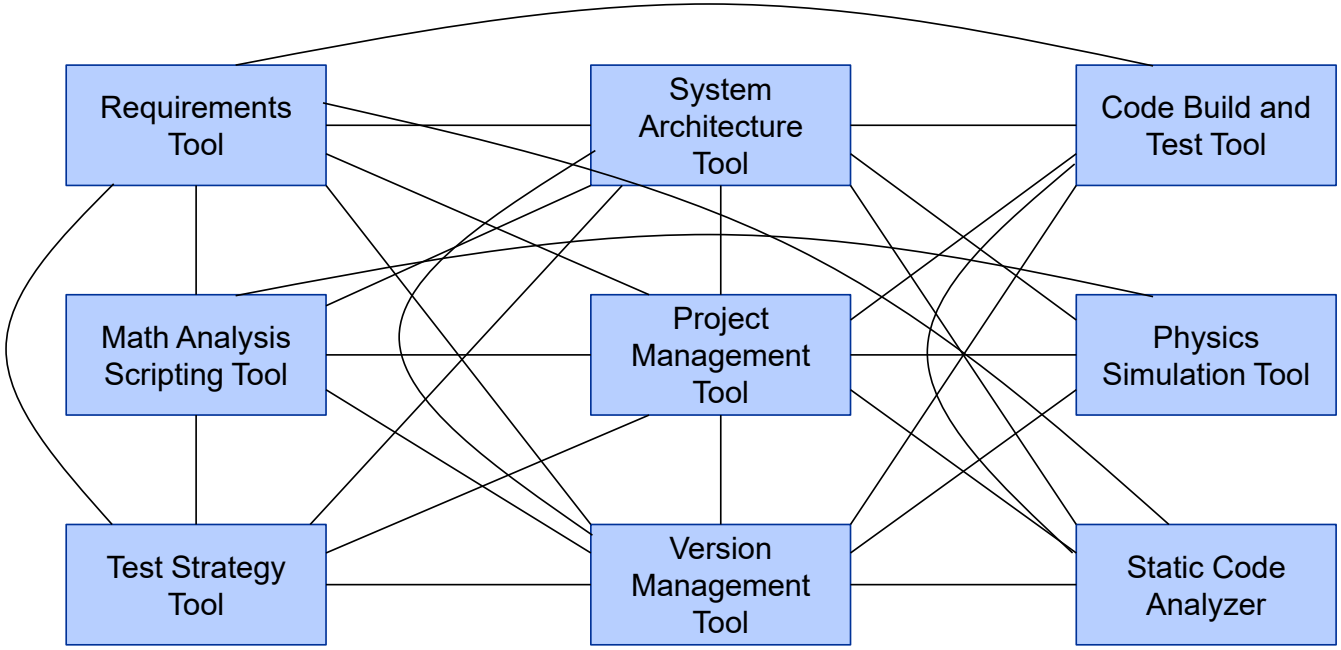
Tool Map


Requirements Tool
System Architecture Tool
Math Analysis Scripting Tool
Project Management Tool
Test Strategy Tool
Version Management Tool

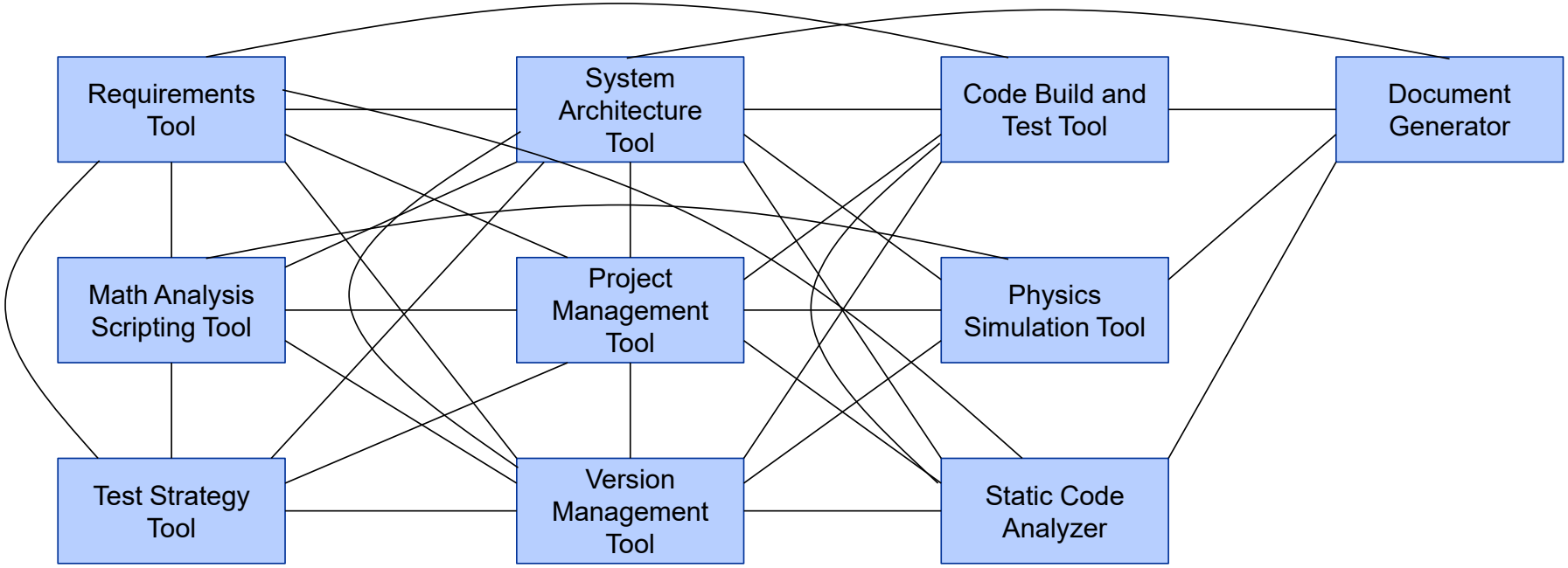
Tool Map


Requirements Tool
System Architecture Tool
Math Analysis Scripting Tool
Project Management Tool
Physics Simulation Tool
Test Strategy Tool
Version Management Tool

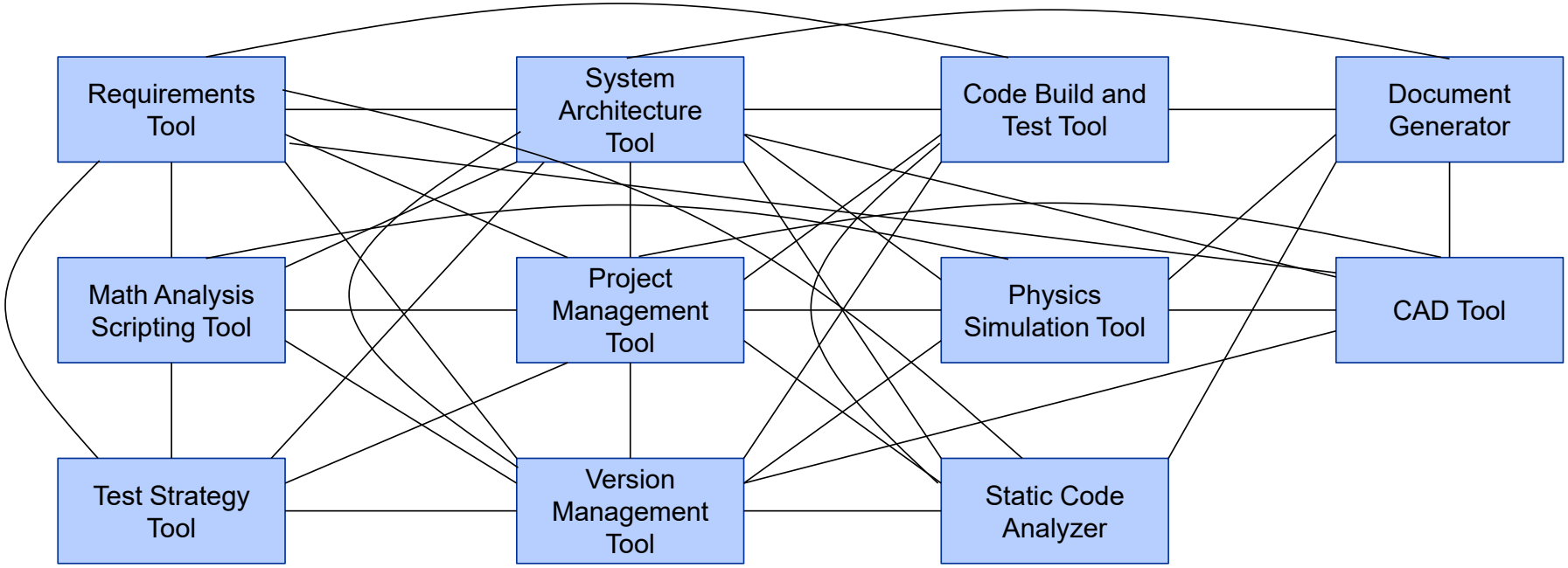
Tool Map


Requirements Tool
System Architecture Tool
Code Build and Test Tool
Math Analysis Scripting Tool
Project Management Tool
Physics Simulation Tool
Test Strategy Tool
Version Management Tool
Static Code Analyzer

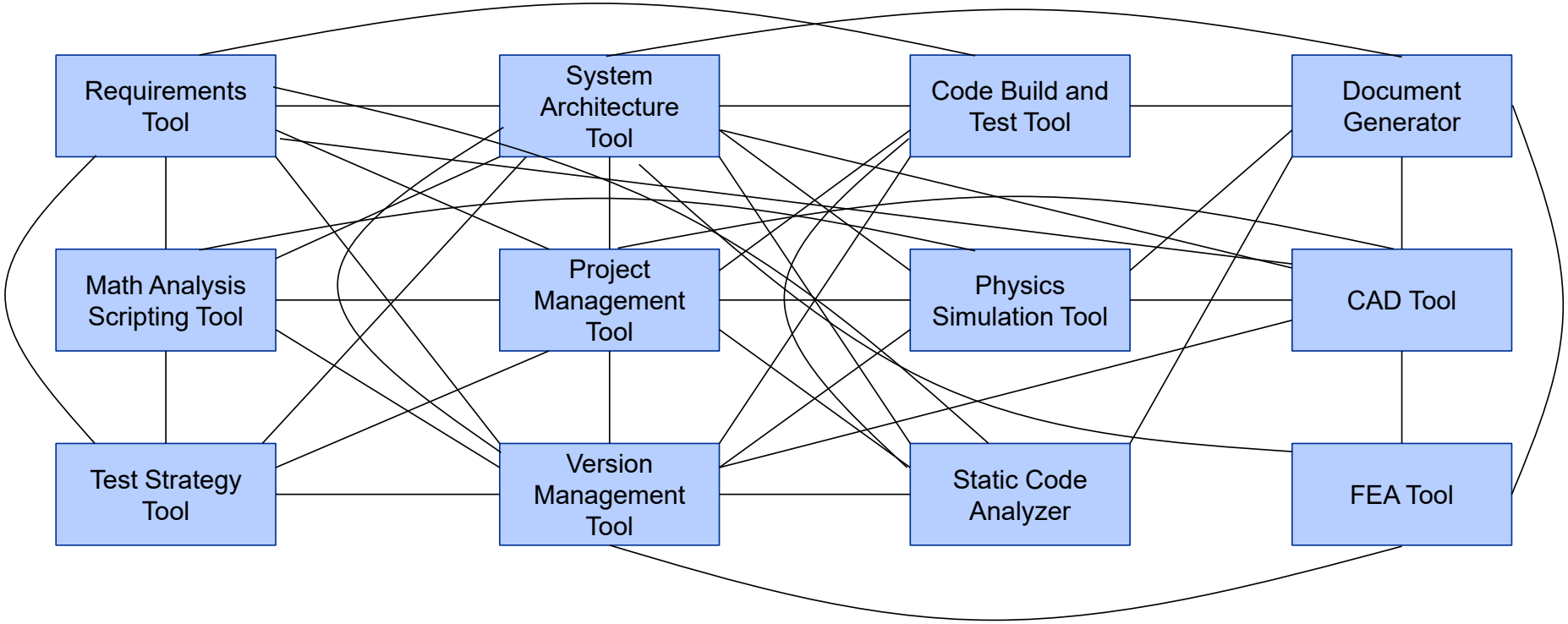
Tool Map


Requirements Tool
System Architecture Tool
Code Build and Test Tool
Document Generator
Math Analysis Scripting Tool
Project Management Tool
Physics Simulation Tool
Test Strategy Tool
Version Management Tool
Static Code Analyzer

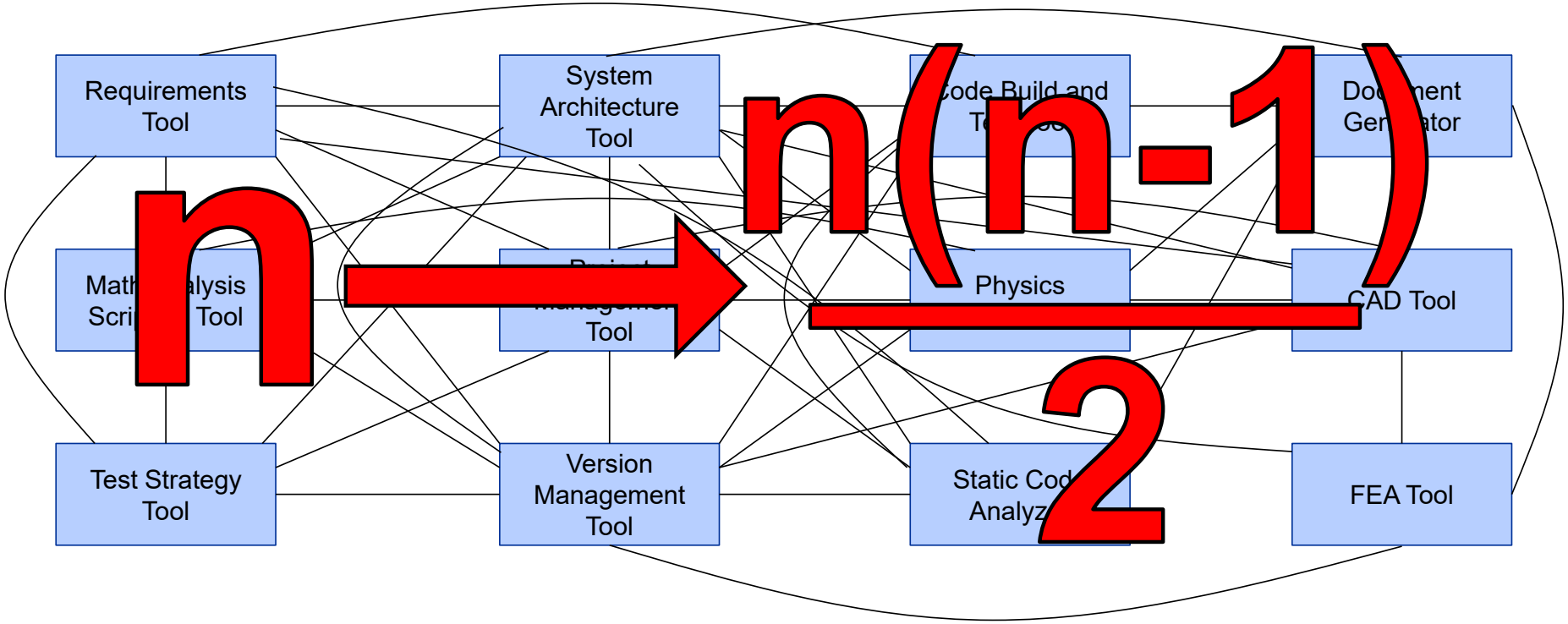
Tool Map


Requirements Tool
System Architecture Tool
Code Build and Test Tool
Document Generator
Math Analysis Scripting Tool
Project Management Tool
Physics Simulation Tool
CAD Tool
Test Strategy Tool
Version Management Tool
Static Code Analyzer

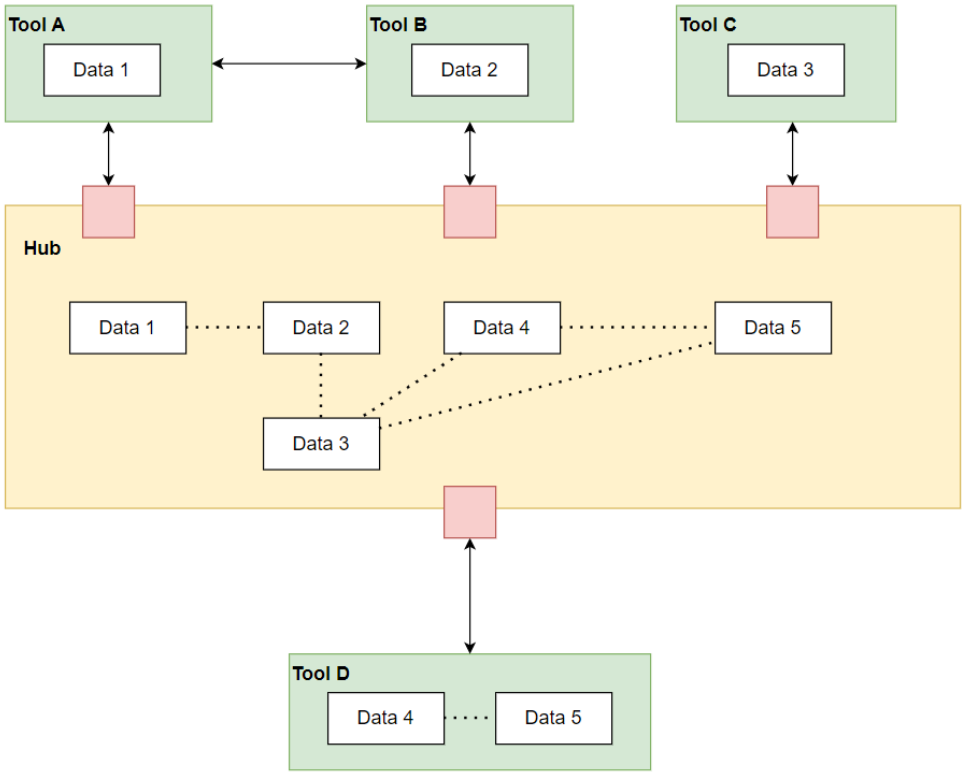
Tool Map


Requirements Tool
System Architecture Tool
Code Build and Test Tool
Document Generator
Math Analysis Scripting Tool
Project Management Tool
Physics Simulation Tool
CAD Tool
Test Strategy Tool
Version Management Tool
Static Code Analyzer
FEA Tool

Tool Map

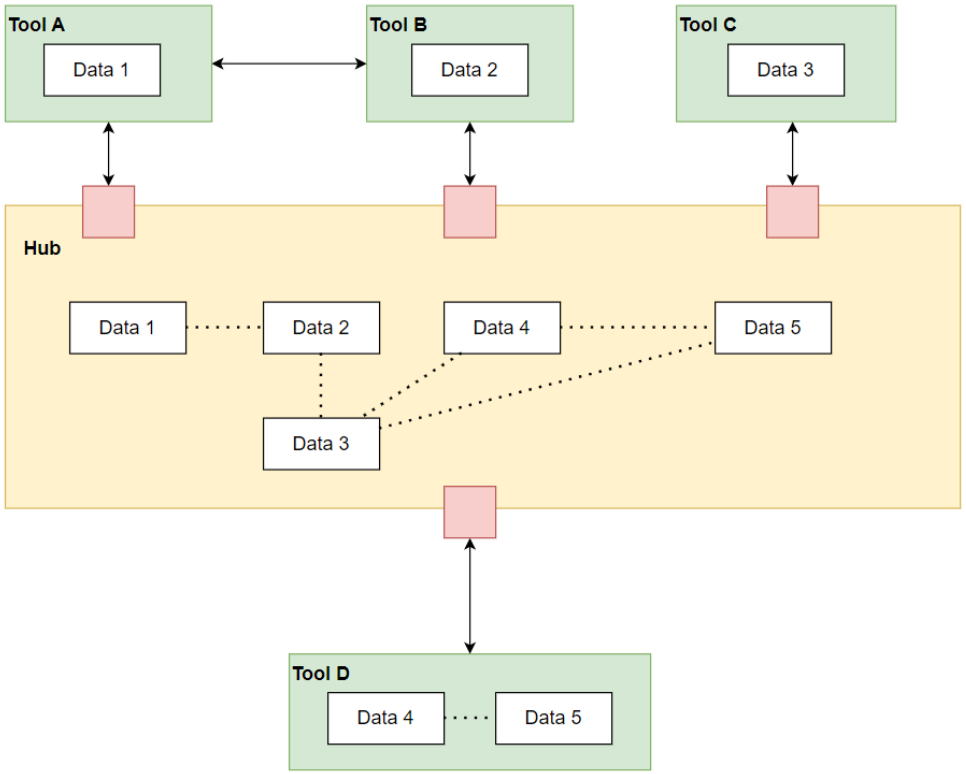

Requirements Tool
System Architecture Tool
Physics
CAD Tool
Test Strategy Tool
Version Management Tool
FEA Tool
n
n(n-1)
2

Hub and Spoke



- Hub and Spoke approach
  - The ‘Hub’ enables integrations from multiple tools
  - Data is connected within the Hub
  - Intra-tool relations are carried
  - Inter-tool relations can be added
  - Comprehensive representation of **entire** dataset in **one** place

Hub and Spoke



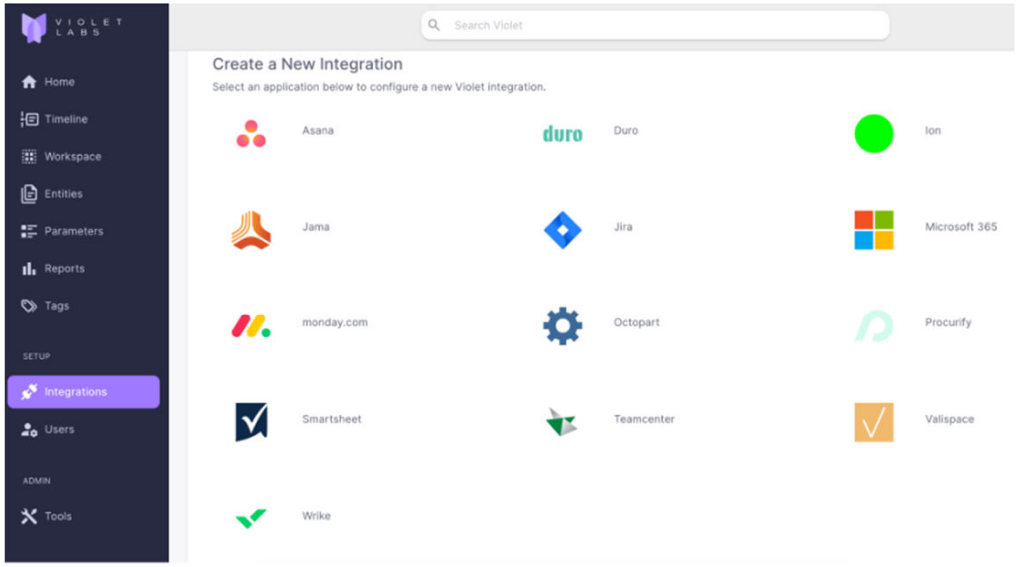
- Hub and Spoke approach
  - The ‘Hub’ enables integrations from multiple tools
  - Data is connected within the Hub
  - Intra-tool relations are carried
  - Inter-tool relations can be added
  - Comprehensive representation of **entire** dataset in **one** place

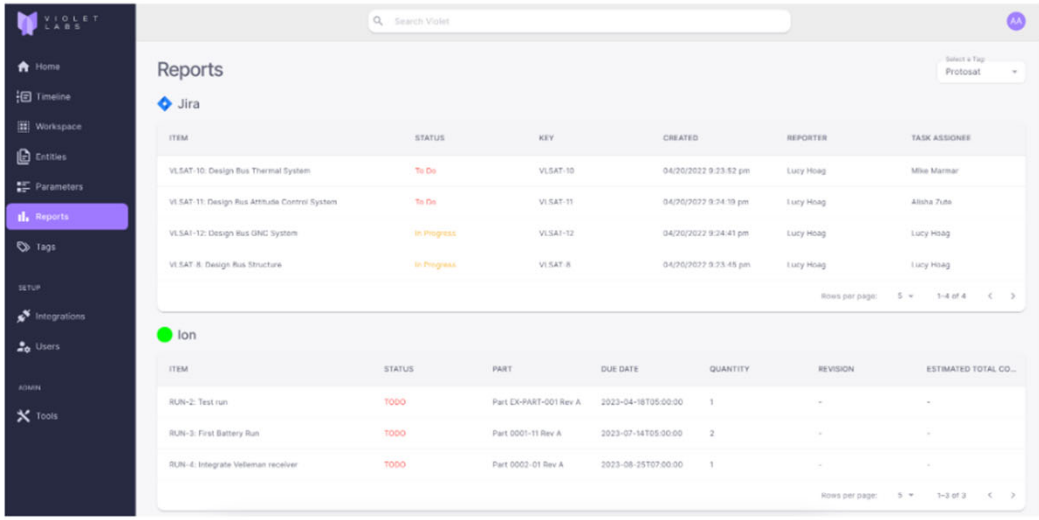
n → n

The Hub – Violet Labs [2]

1 Connect tools

2 Violet continuously fetches and structures the data



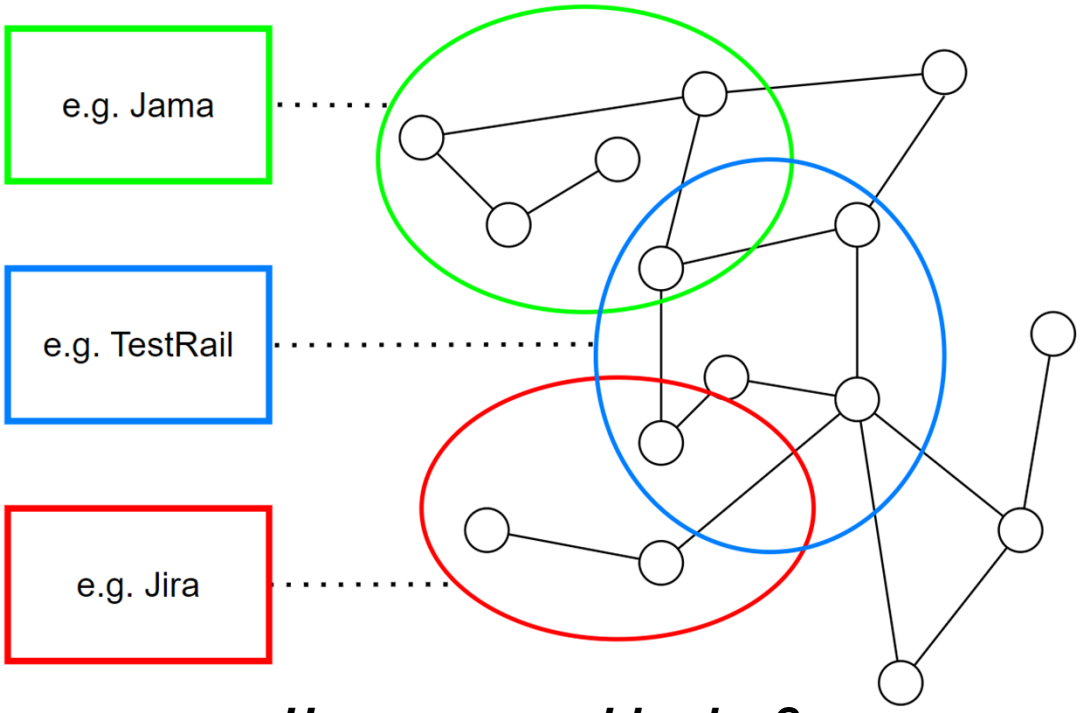


3 Enables bi-directional data sync, parameter backbone, report generation

www.violetlabs.com

The Hub – Violet Labs [2]

So, we have a graph representation of the data…



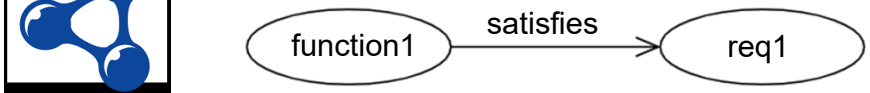
How can we add value?

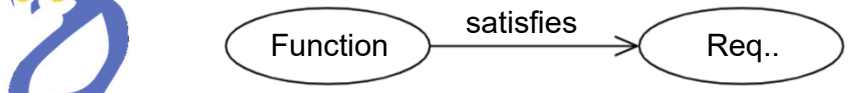
The Hub – OML Rosetta [3]

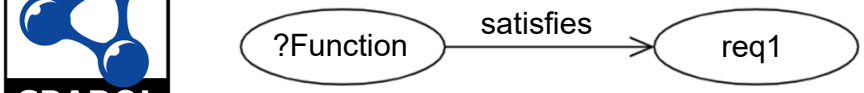
Semantic Web Technologies (SWTs)

- provide an approach to the structuring and understanding of data [4].

- Resource Description Framework (RDF) [5]
  - *Makes information machine-readable*
- Web Ontology Language (OWL) [6]
  - *Provides context to information, enables validation*
- SPARQL [7]
  - *Enables users to query RDF graphs*







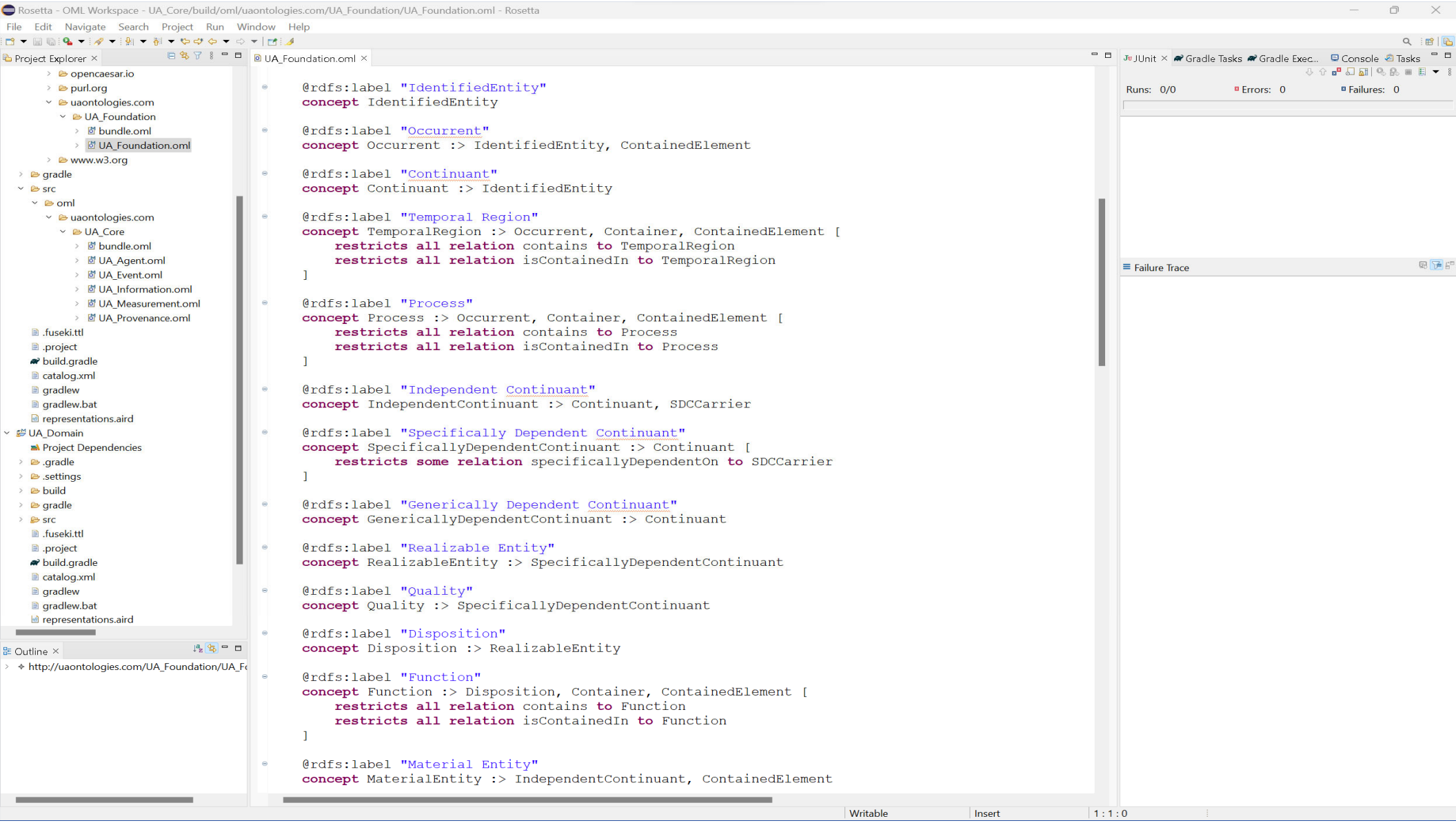
The Hub – OML Rosetta [3]

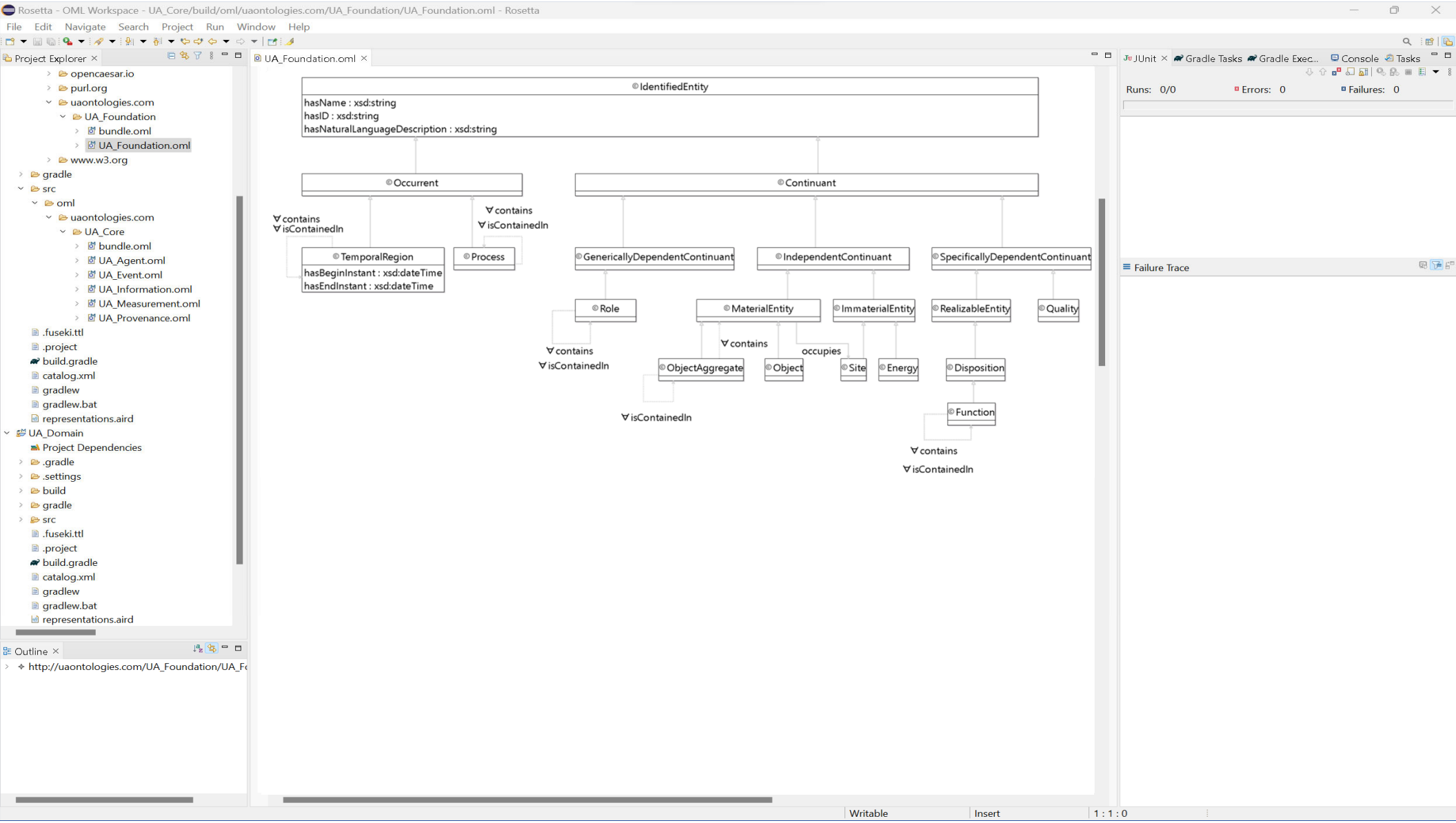
- The Ontological Modeling Language (OML) is a thin extension to OWL 2 DL
  - Part of the 'OpenCAESAR' project [3]
  - enables reification of relations and convenient use of mixin classes (as aspects).
  - enables representation of data as RDF and therefore SPARQL querying.

- OML uses the concept of '**bundles**'
  - these allow the modeler to group together multiple descriptions into a closed-world dataset.
  - the **closed-world assumption** allows the reasoner to highlight the absence of expected information.

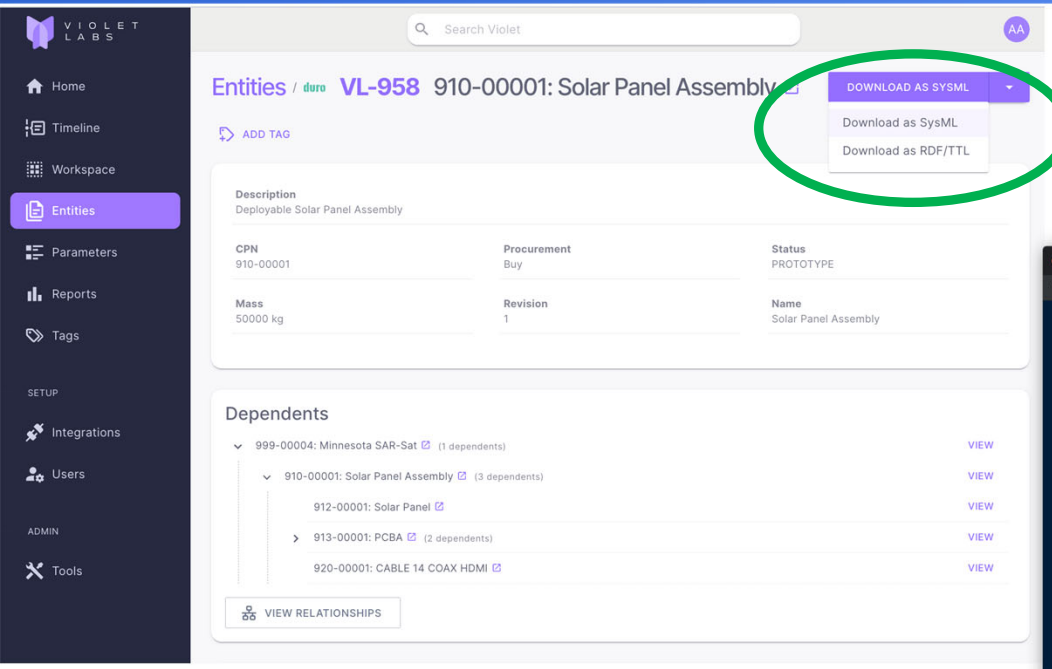
- OML Ontologies and Descriptions can be modeled using OML Rosetta

- OML Rosetta is Eclipse-based
  - supports textual notation
  - supports graphical representation

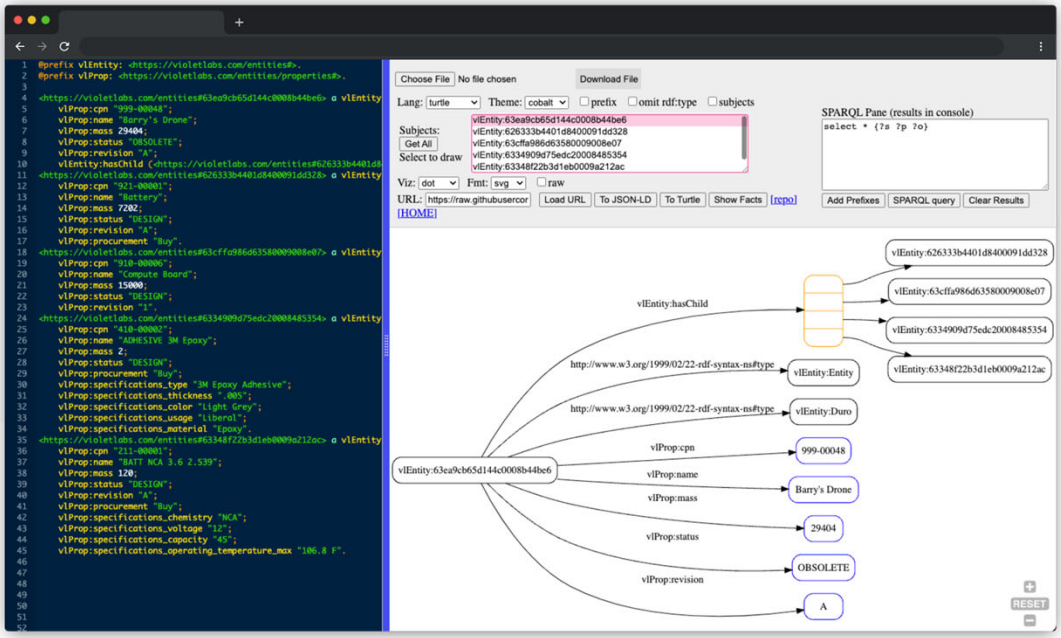

Rosetta - OML Workspace - UA_Core/build/oml/uaontologies.com/UA_Foundation/UA_Foundation.oml - Rosetta
File Edit Navigate Search Project Run Window Help
Project Explorer
opencaesar.io
purl.org
uaontologies.com
UA_Foundation
bundle.oml
UA_Foundation.oml
www.w3.org
gradle
src
oml
uaontologies.com
UA_Core
bundle.oml
UA_Agent.oml
UA_Event.oml
UA_Information.oml
UA_Measurement.oml
UA_Provenance.oml
.fuseki.ttl
.project
build.gradle
catalog.xml
gradlew
gradlew.bat
representations.aird
UA_Domain
Project Dependencies
.gradle
.settings
build
gradle
src
.fuseki.ttl
.project
build.gradle
catalog.xml
gradlew
gradlew.bat
representations.aird
Outline
http://uaontologies.com/UA_Foundation/UA_F
UA_Foundation.oml
@rdfs:label "IdentifiedEntity"
concept IdentifiedEntity
@rdfs:label "Occurrent"
concept Occurrent :> IdentifiedEntity, ContainedElement
@rdfs:label "Continuant"
concept Continuant :> IdentifiedEntity
@rdfs:label "Temporal Region"
concept TemporalRegion :> Occurrent, Container, ContainedElement [
restricts all relation contains to TemporalRegion
restricts all relation isContainedIn to TemporalRegion
]
@rdfs:label "Process"
concept Process :> Occurrent, Container, ContainedElement [
restricts all relation contains to Process
restricts all relation isContainedIn to Process
]
@rdfs:label "Independent Continuant"
concept IndependentContinuant :> Continuant, SDCCarrier
@rdfs:label "Specifically Dependent Continuant"
concept SpecificallyDependentContinuant :> Continuant [
restricts some relation specificallyDependentOn to SDCCarrier
]
@rdfs:label "Generically Dependent Continuant"
concept GenericallyDependentContinuant :> Continuant
@rdfs:label "Realizable Entity"
concept RealizableEntity :> SpecificallyDependentContinuant
@rdfs:label "Quality"
concept Quality :> SpecificallyDependentContinuant
@rdfs:label "Disposition"
concept Disposition :> RealizableEntity
@rdfs:label "Function"
concept Function :> Disposition, Container, ContainedElement [
restricts all relation contains to Function
restricts all relation isContainedIn to Function
]
@rdfs:label "Material Entity"
concept MaterialEntity :> IndependentContinuant, ContainedElement
JUnit
Gradle Tasks
Gradle Exec...
Console
Tasks
Runs: 0/0
Errors: 0
Failures: 0
Failure Trace
Writable
Insert
1 : 1 : 0


Rosetta - OML Workspace - UA_Core/build/oml/uaontologies.com/UA_Foundation/UA_Foundation.oml - Rosetta
File Edit Navigate Search Project Run Window Help
Project Explorer
opencaesar.io
purl.org
uaontologies.com
UA_Foundation
bundle.oml
UA_Foundation.oml
www.w3.org
gradle
src
oml
uaontologies.com
UA_Core
bundle.oml
UA_Agent.oml
UA_Event.oml
UA_Information.oml
UA_Measurement.oml
UA_Provenance.oml
.fuseki.ttl
.project
build.gradle
catalog.xml
gradlew
gradlew.bat
representations.aird
UA_Domain
Project Dependencies
.gradle
.settings
build
gradle
src
.fuseki.ttl
.project
build.gradle
catalog.xml
gradlew
gradlew.bat
representations.aird
Outline
http://uaontologies.com/UA_Foundation/UA_F
UA_Foundation.oml
IdentifiedEntity
hasName : xsd:string
hasID : xsd:string
hasNaturalLanguageDescription : xsd:string
Occurrent
Continuant
∀contains
∀isContainedIn
∀contains
∀isContainedIn
TemporalRegion
hasBeginInstant : xsd:dateTime
hasEndInstant : xsd:dateTime
Process
GenericallyDependentContinuant
IndependentContinuant
SpecificallyDependentContinuant
Role
MaterialEntity
ImmaterialEntity
RealizableEntity
Quality
∀contains
∀isContainedIn
∀contains
occupies
ObjectAggregate
Object
Site
Energy
Disposition
∀isContainedIn
Function
∀contains
∀isContainedIn
JUnit
Gradle Tasks
Gradle Exec...
Console
Tasks
Runs: 0/0
Errors: 0
Failures: 0
Failure Trace
Writable
Insert
1 : 1 : 0

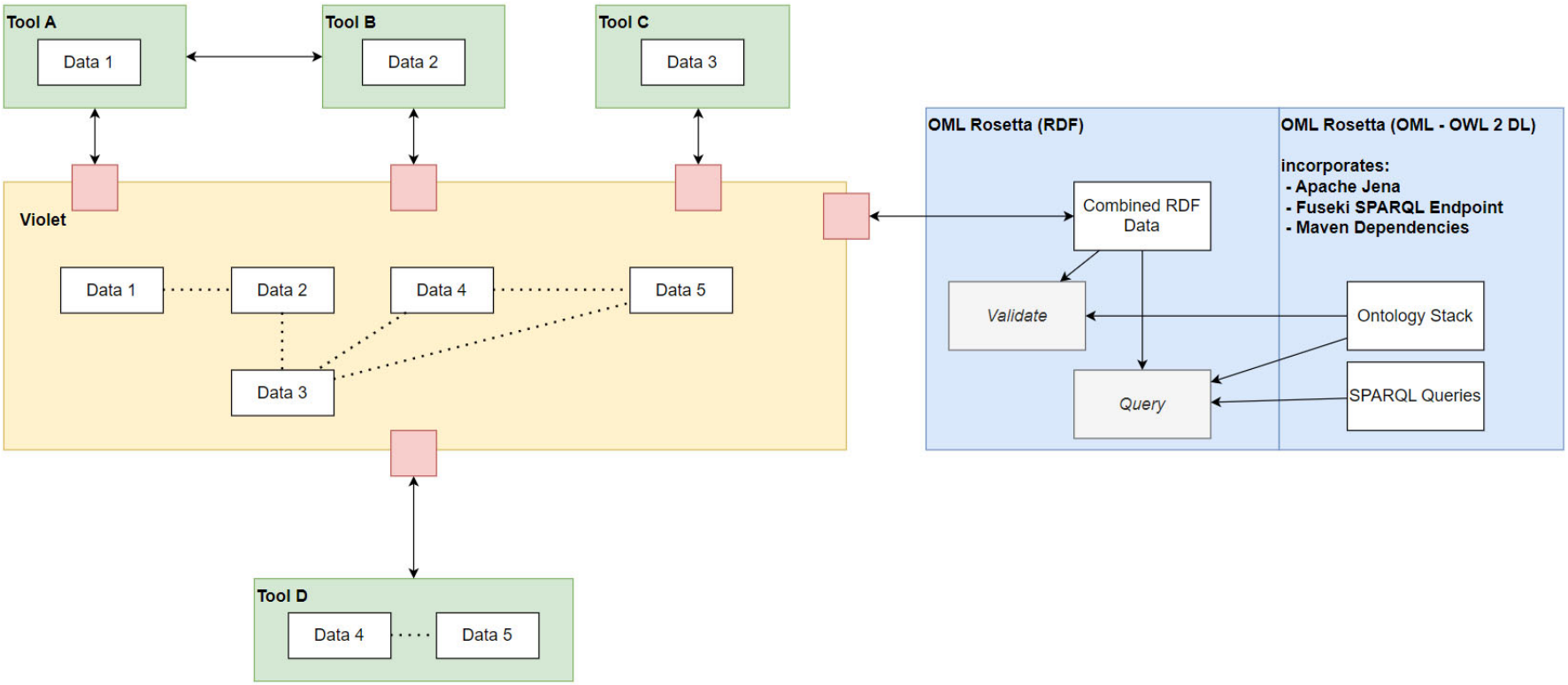
The Hub – Violet Labs [2] and OML Rosetta [4]



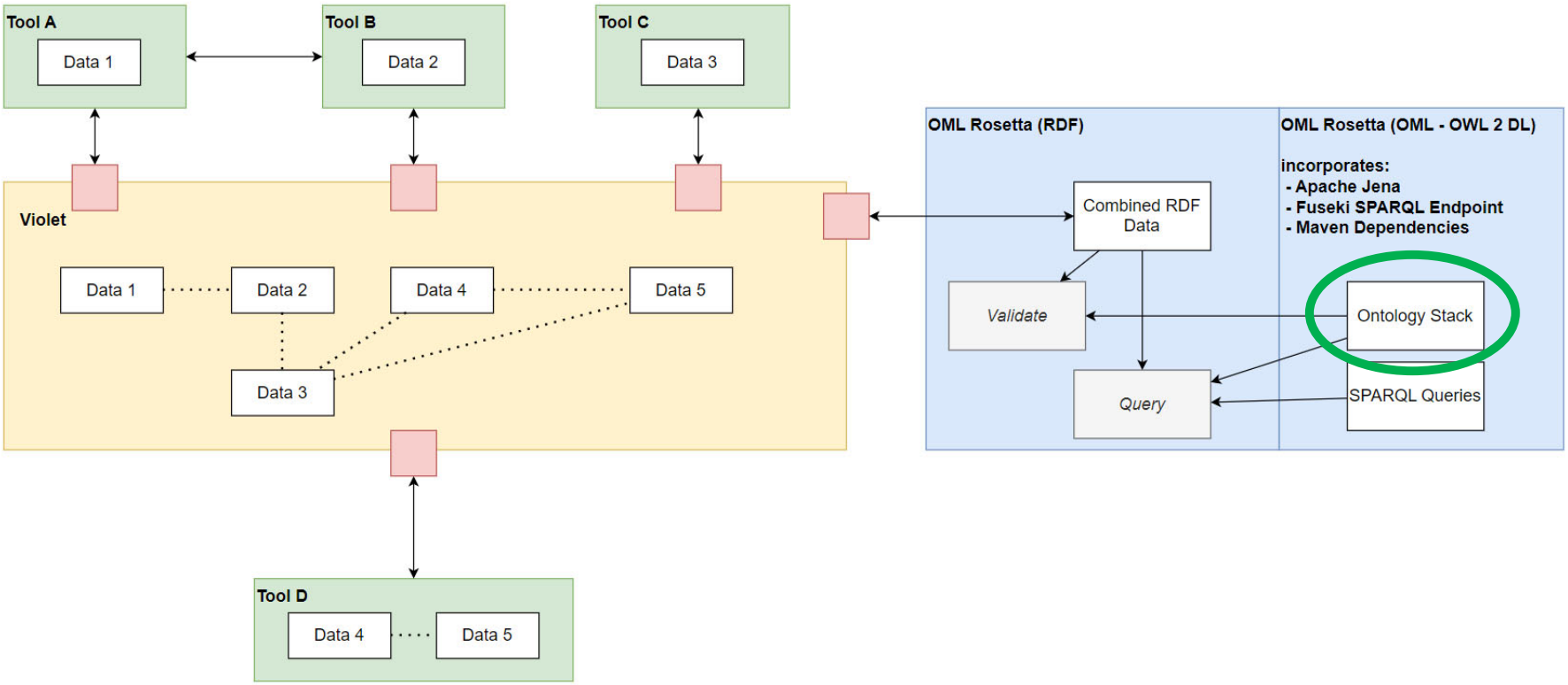
Generate OML/RDF



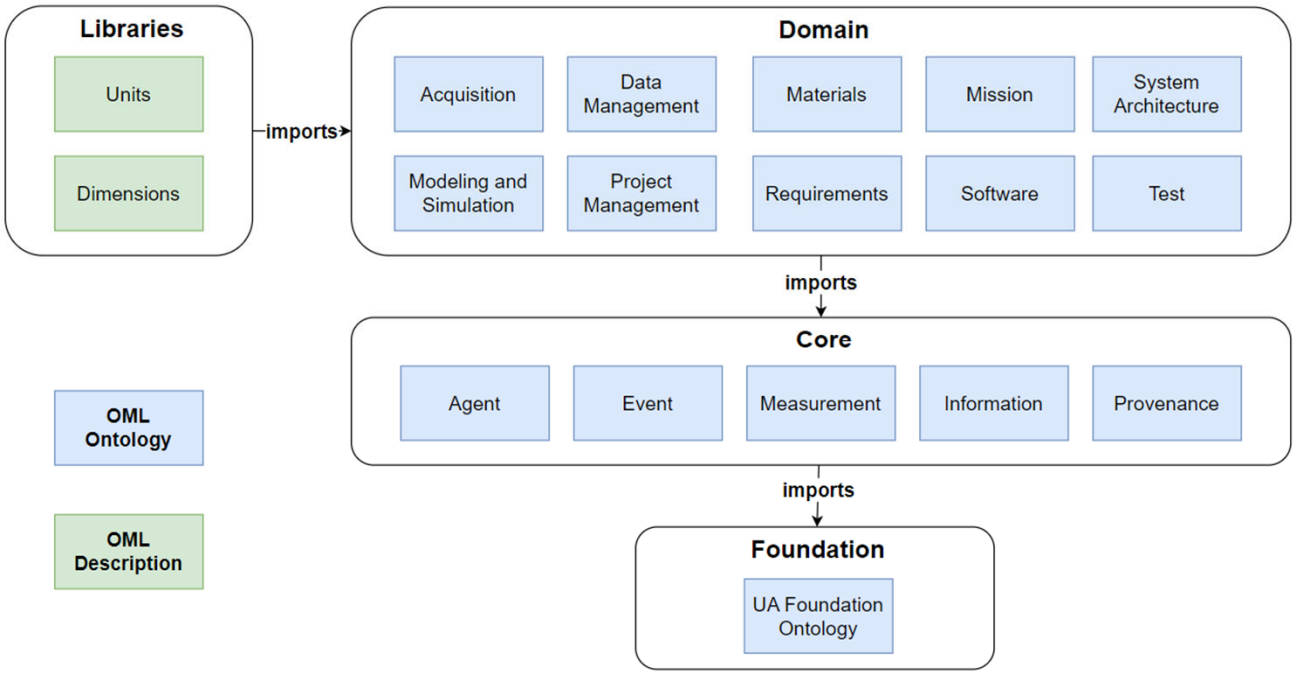
The Hub – Violet Labs [2] and OML Rosetta [4]



The Hub – Violet Labs [2] and OML Rosetta [4]



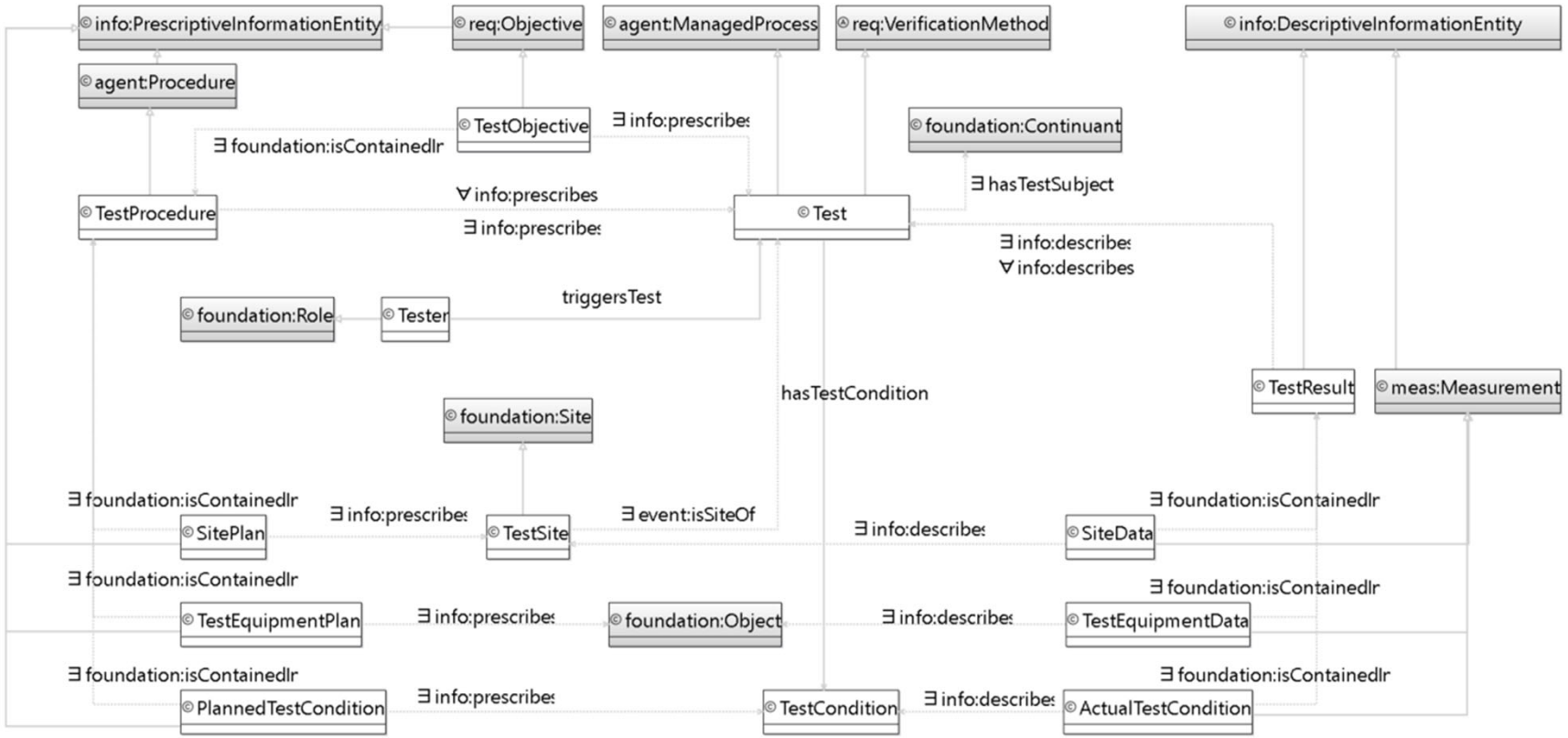
The Hub – UA OML Ontology Stack



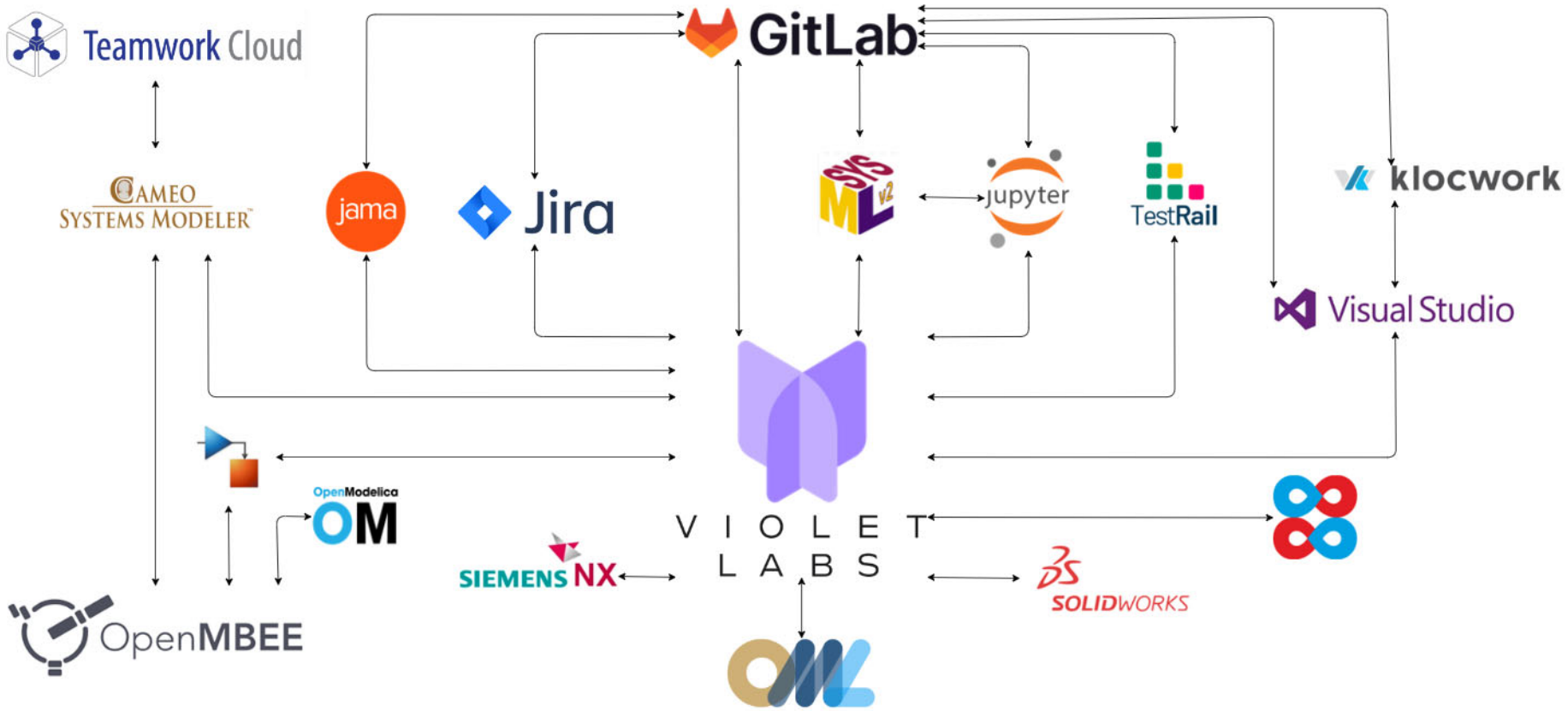
- Foundation Ontology
  - uses Basic Formal Ontology (BFO) [8]
- Core Ontologies
  - uses Common Core Ontologies (CCO) [9]
- Domain Ontologies
  - identified as key to support student work
  - draws from relevant standards and theory
    (e.g. ISO 42010 [10], VIM4 [11])
- Libraries
  - draws from existing libraries

Reuse wherever possible!

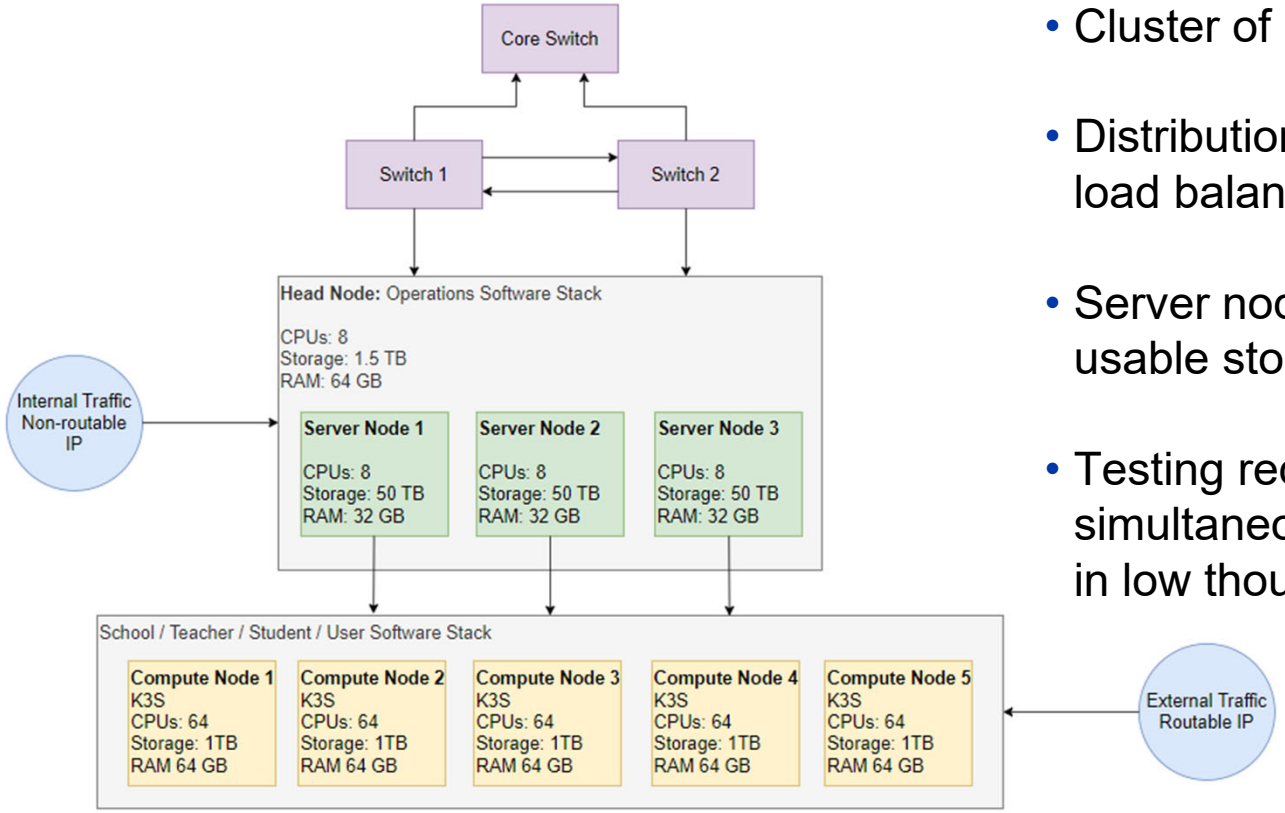
The Hub – Domain-Level 'Test' OML Ontology



Digital Engineering Factory

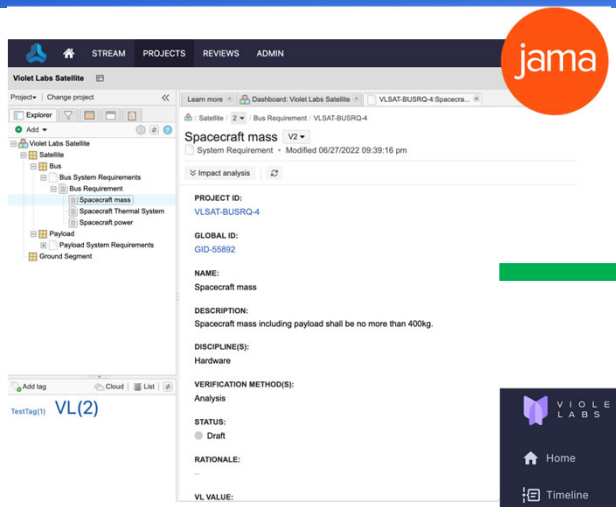


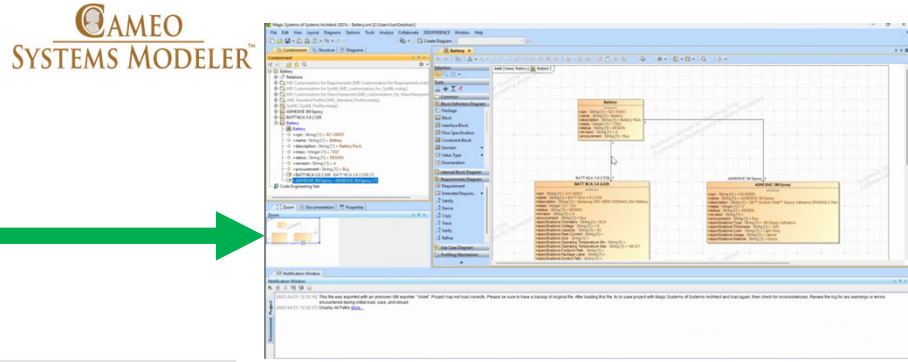
Digital Engineering Factory

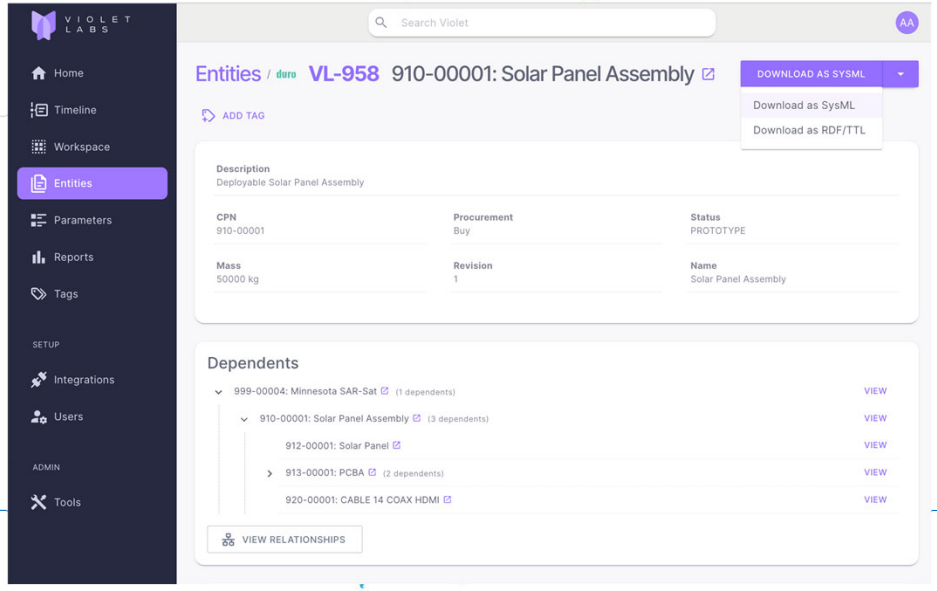


- Cluster of 320 CPUs to support students
- Distribution to support high-availability load balancing and ease of maintenance
- Server nodes will provide ~100TB of usable storage for installations
- Testing required to determine number of simultaneous users supported (expected in low thousands)

Use Case 1 – Transferring Project Data Between Courses

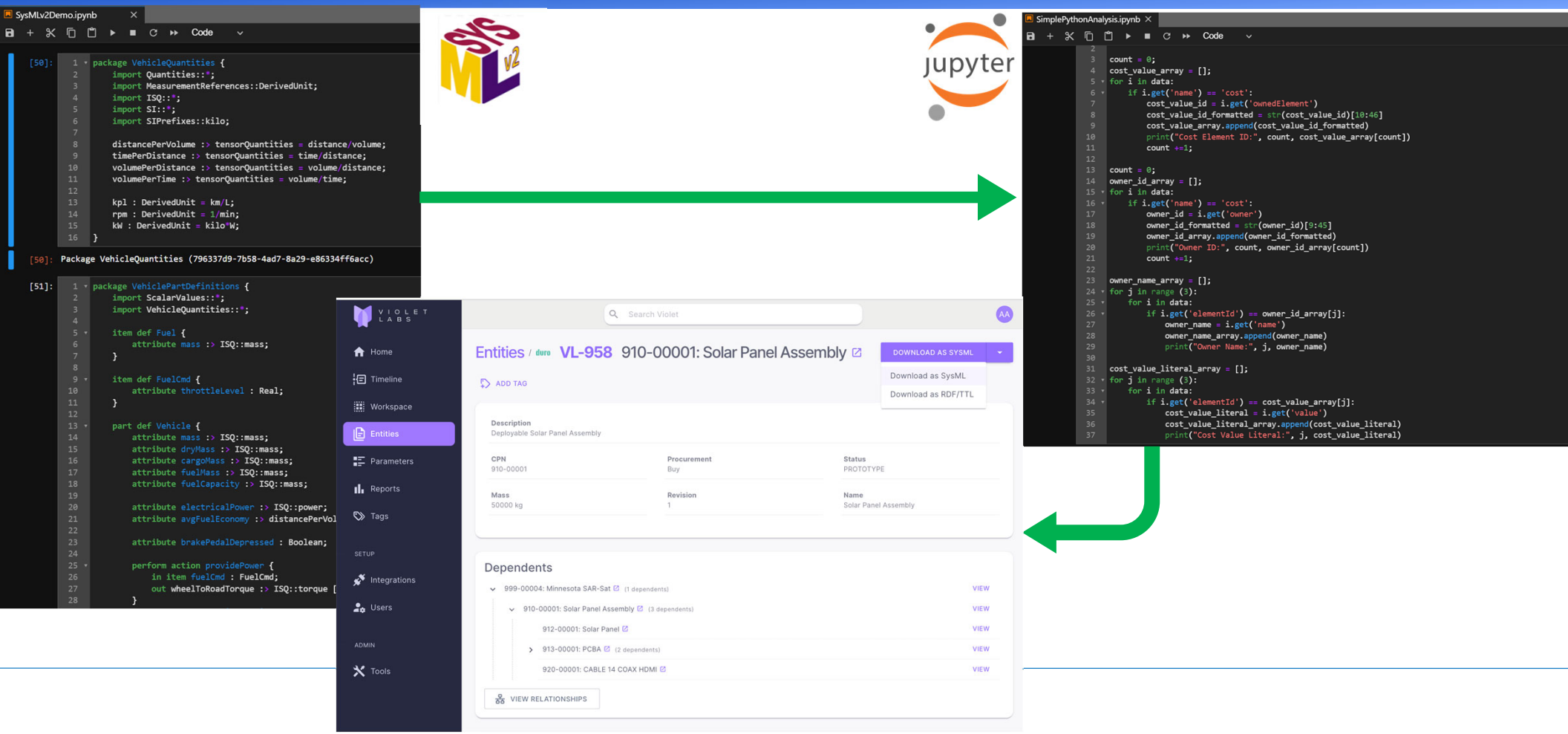






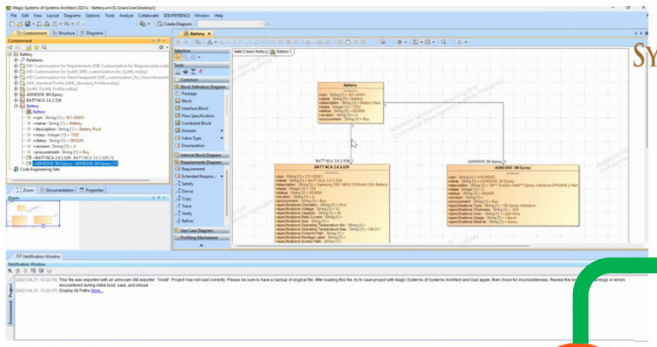
Use Case 1 – Transferring Project Data Between Courses

Global Product Data Interoperability Summit | **2023**

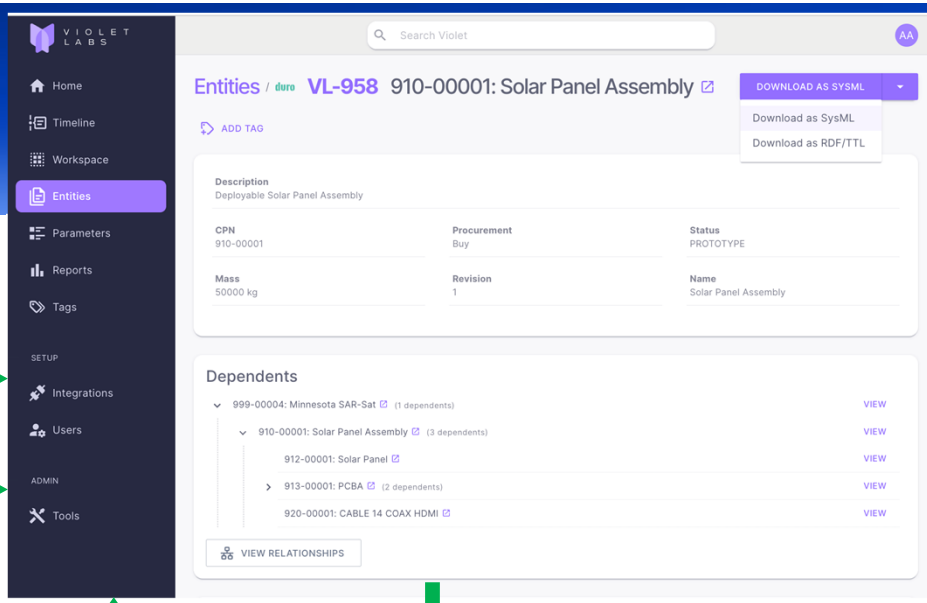


Use Case 2 – Static Data Integration and Evaluation

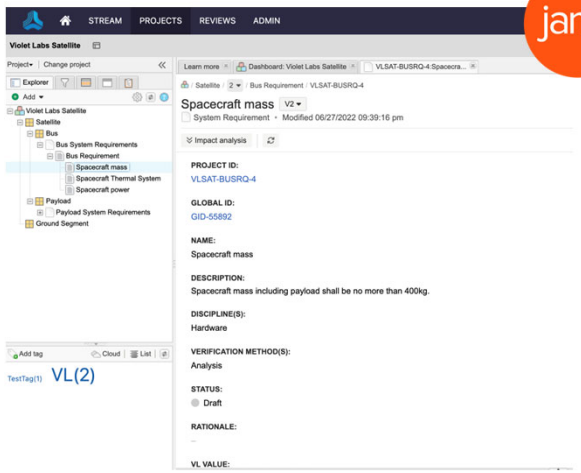
Global Product Data Interoperability Summit | **2023**



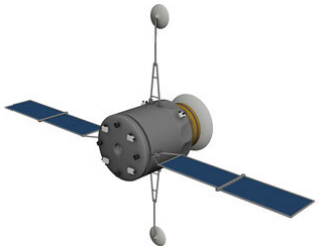
CAMEO SYSTEMS MODELER™

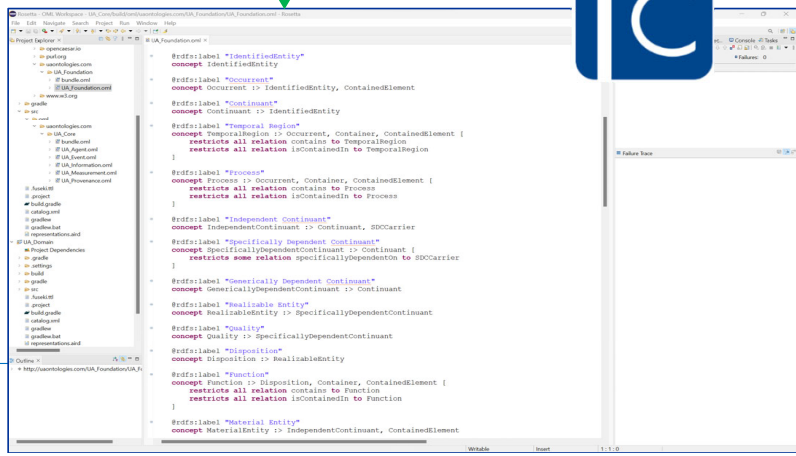


jama



SIEMENS NX

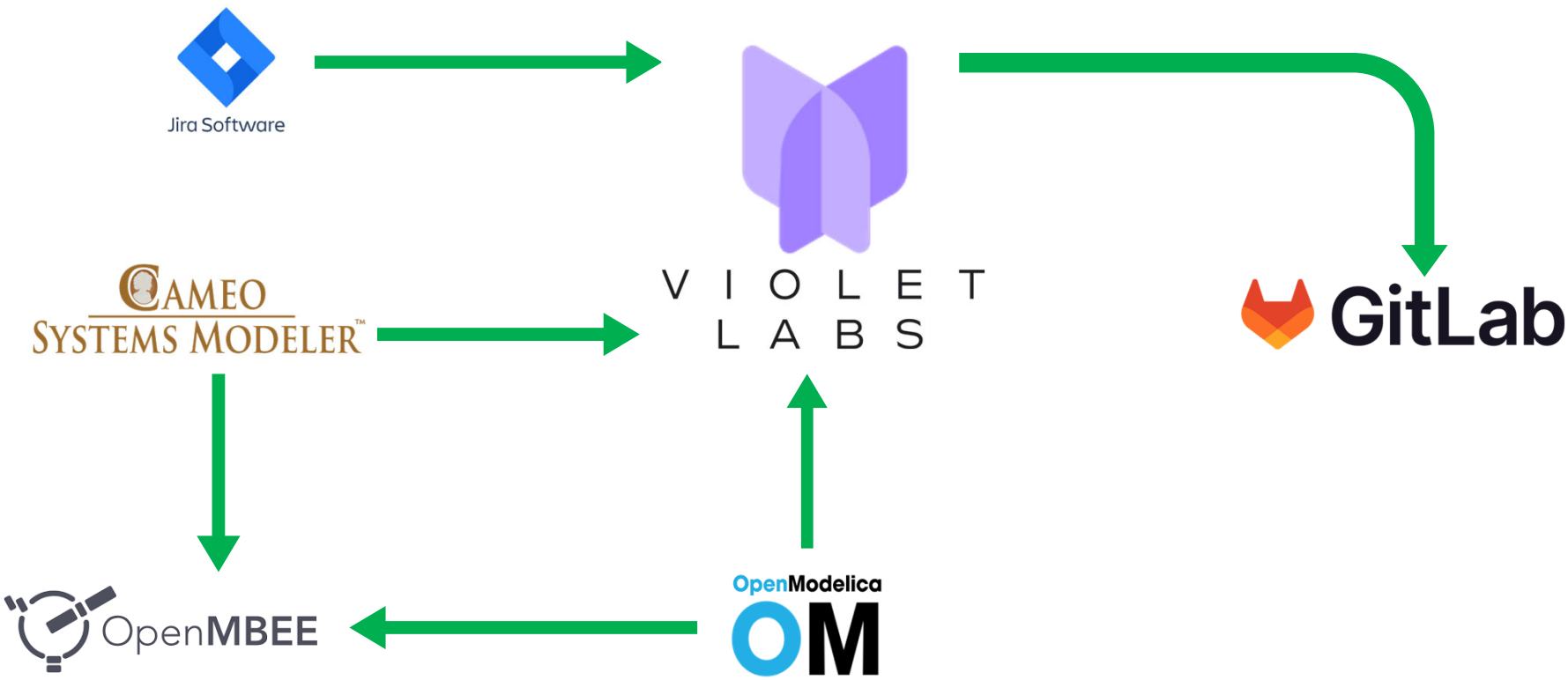




GLOBAL PRODUCT DATA INTEROPERABILITY SUMMIT 2023

Use Case 3 – Continuous Requirement Verification

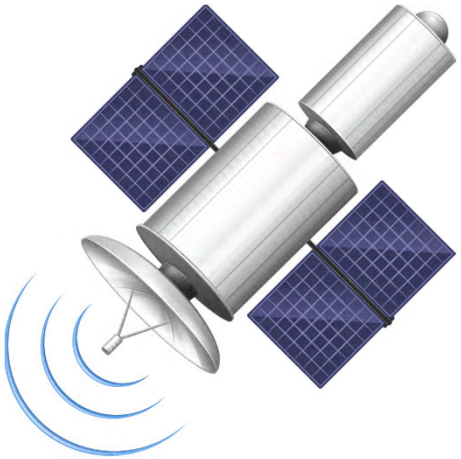
Use Case 4 – Team Project Submission



Example – Overview

"Arizona Sat"

- Students are developing a Low-Earth Orbit satellite
- Students have requirements defined in jama
- Students have components defined in duro
- Students wish to review and submit their entire dataset

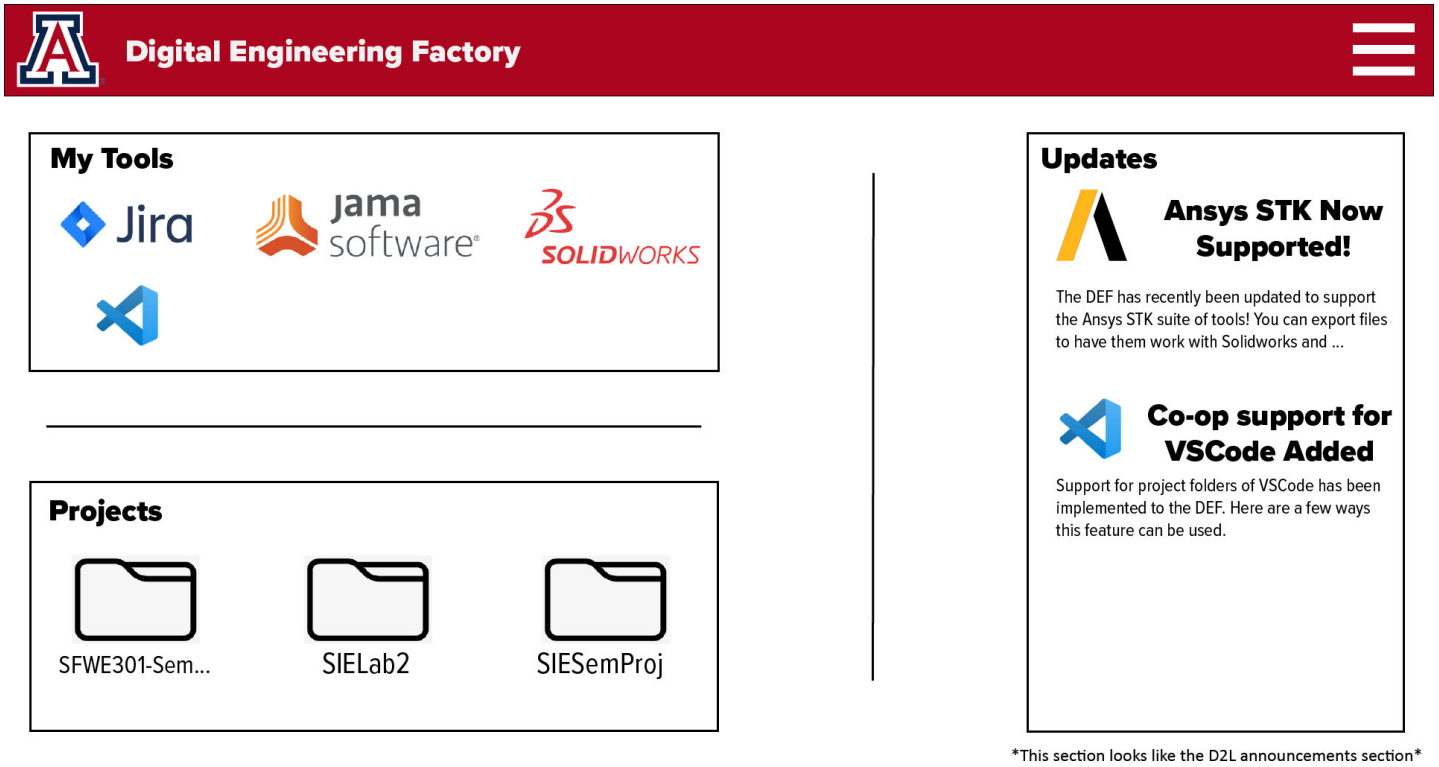


Example – Sign-on

Students will log into the DEF to:

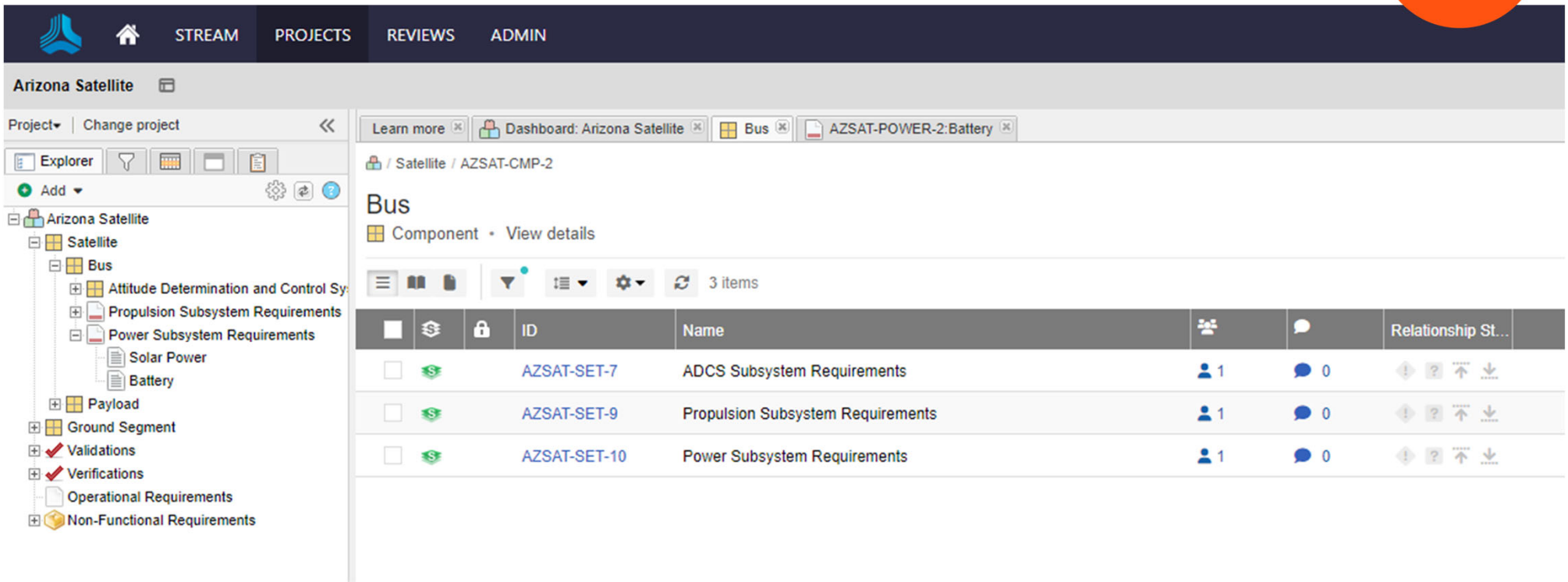
- Access tools
- Access projects
- Work collaboratively

SFWE students are currently helping to develop the DEF Landing Page

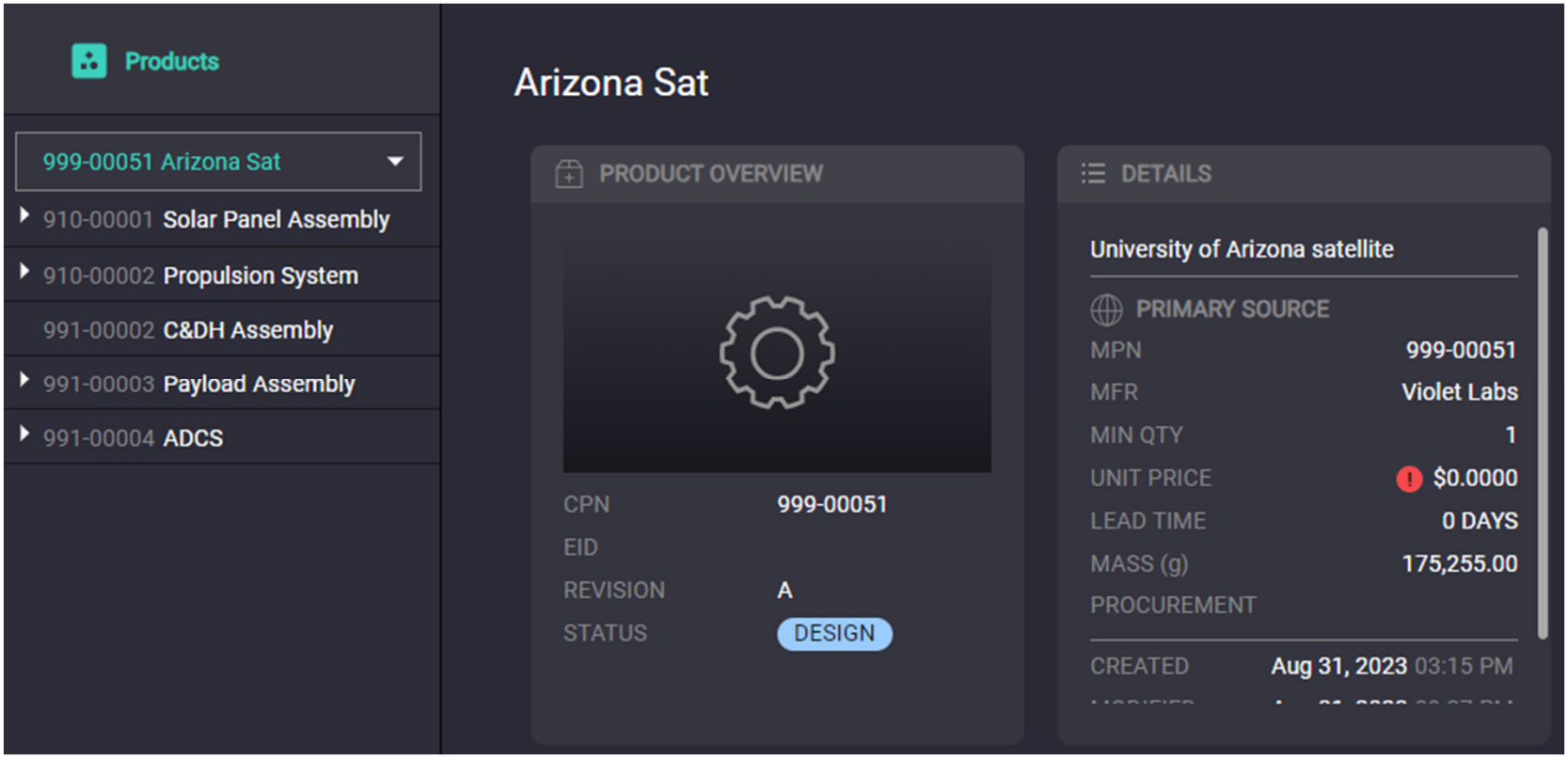


This section looks like the D2L announcements section

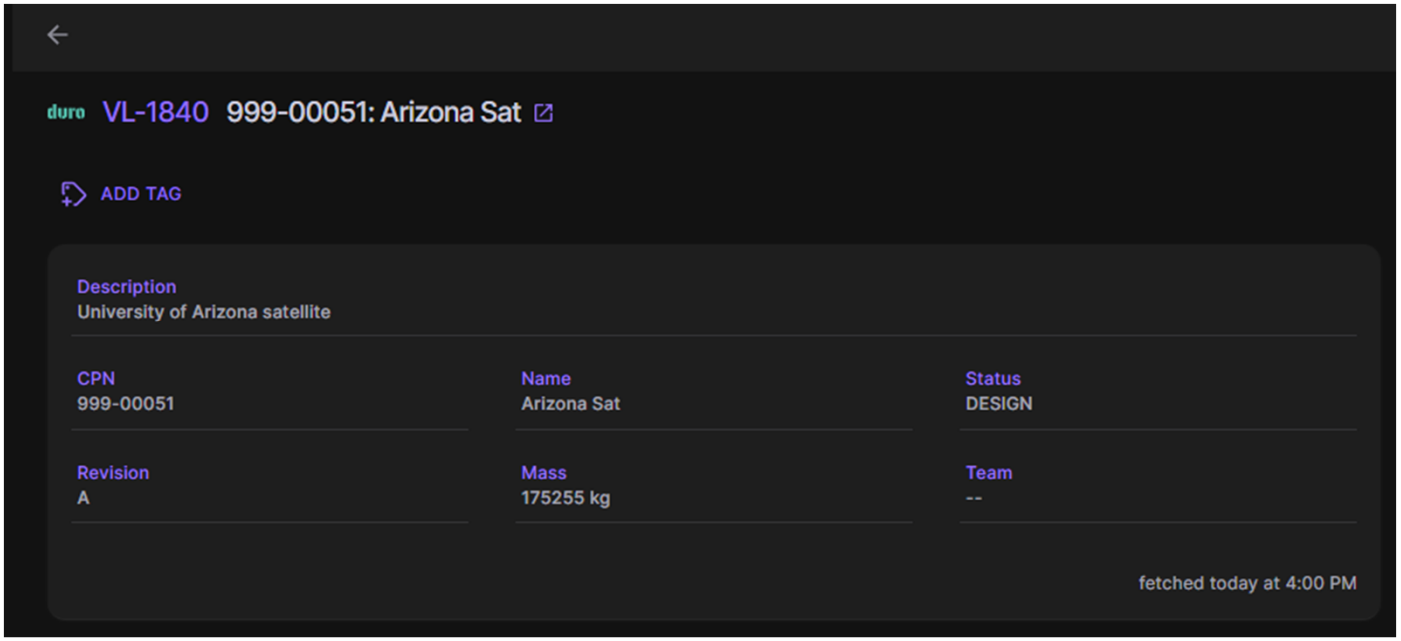
Example – Student Project Data



Example – Student Project Data

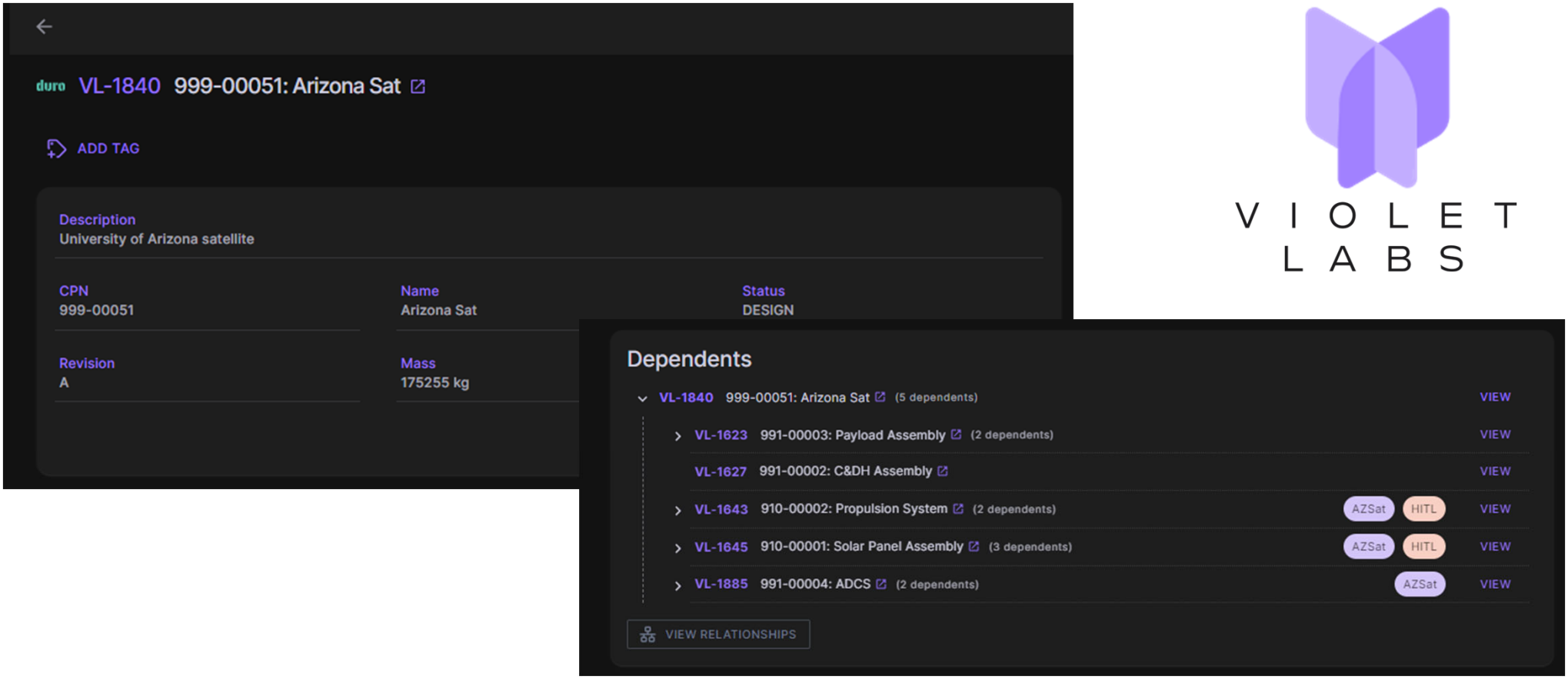


Example – Student Project Data



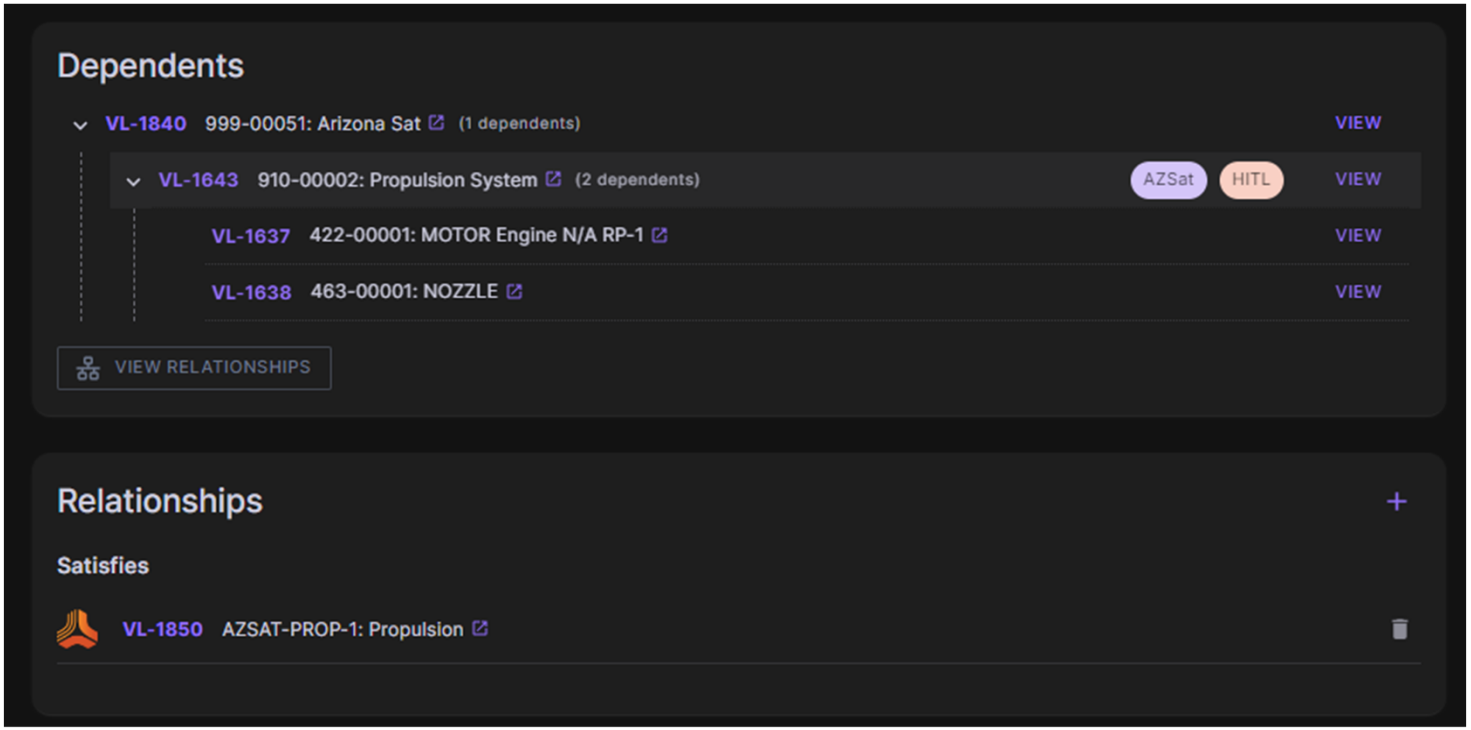
VIOLET LABS

Example – Student Project Data



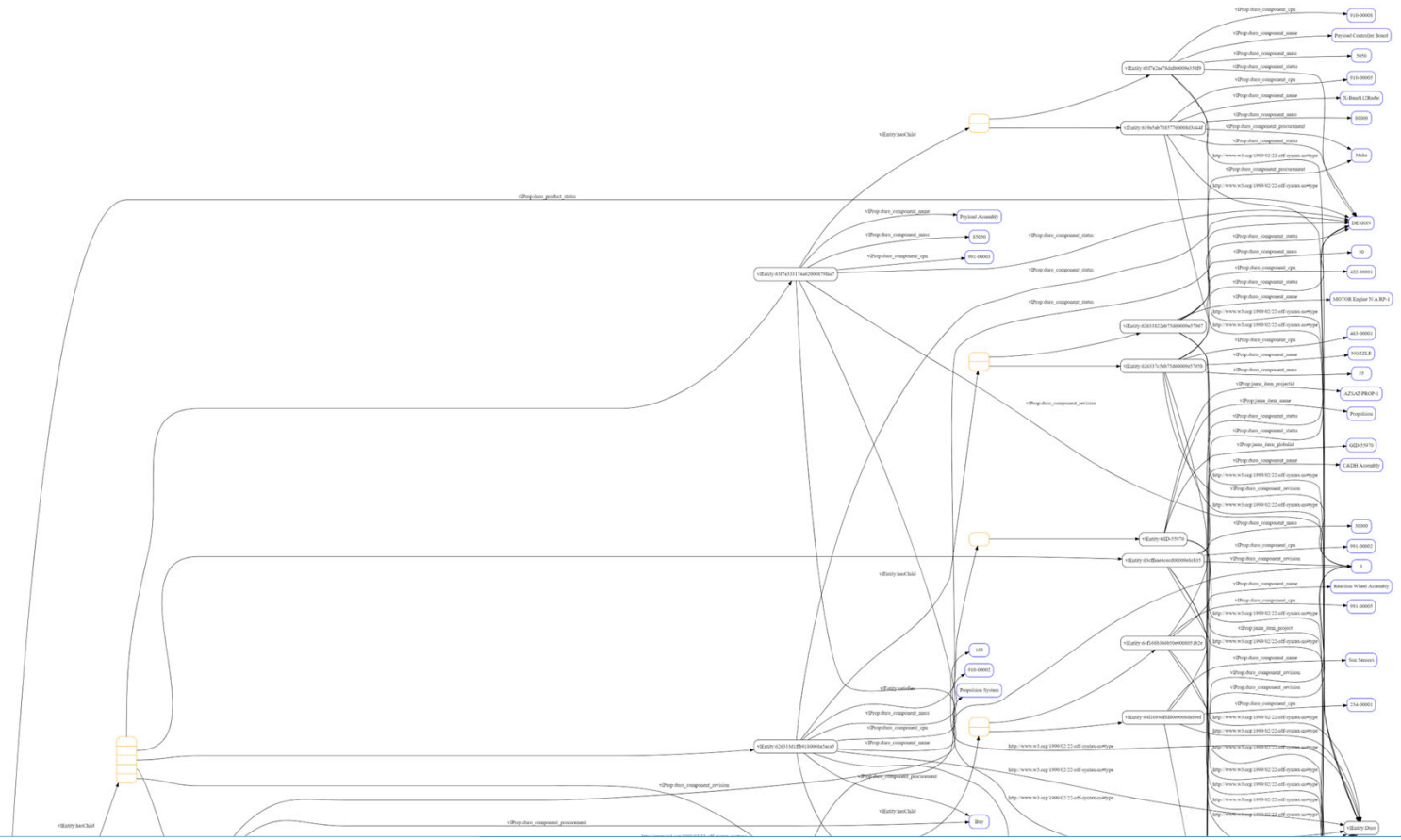
VIOLET LABS

Example – Student Project Data



VIOLET LABS

Example – RDF Representation



Example – Validation

Example – Validation

Initial issue with build – validation failed

Operation	Duration
Run build	1.401 s
Configure build	0.196 s
Calculate build tree task graph	0.016 s
Load build	0.010 s
Build finished for file system watching	0.001 s
Run main tasks	1.156 s
Run tasks	1.155 s
Build started for file system watching	0.013 s

Example – Validation

Antisymmetric property isAbout

```
<?xml version="1.0"?>
<rdf:RDF xmlns="urn:unnamed:ontology#ont395285532729201#"
    xml:base="urn:unnamed:ontology#ont395285532729201"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:swrl="http://www.w3.org/2003/11/swrl#"
    xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
    xmlns:UA_Foundation="http://uaontologies.com/UA_Foundation/UA_Foundation#"
    xmlns:UA_Information="http://uaontologies.com/UA_Core/UA_Information#">
   <owl:Ontology rdf:about="urn:unnamed:ontology#ont395285532729201"/>
```

Example – Validation

Antisymmetric property isAbout

```
<?xml version="1.0"?>
<rdf:RDF xmlns="urn:unnamed:ontology#ont395285532729201#"
    xml:base="urn:unnamed:ontology#ont395285532729201"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:swrl="http://www.w3.org/2003/11/swrl#"
    xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
    xmlns:UA_Foundation="http://uaontologies.com/UA_Foundation/UA_Foundation#"
    xmlns:UA_Information="http://uaontologies.com/UA_Core/UA_Information#">
   <owl:Ontology rdf:about="urn:unnamed:ontology#ont395285532729201"/>
```

```
<!-- http://uavl.com/UAVL_Example/tag1#AZSAT-SET-7 -->

<owl:NamedIndividual rdf:about="http://uavl.com/UAVL_Example/tag1#AZSAT-SET-7">
   <UA_Information:prescribes rdf:resource="http://uavl.com/UAVL_Example/tag1#Arizona_Sat"/>
</owl:NamedIndividual>


<!-- http://uavl.com/UAVL_Example/tag1#Arizona_Sat -->

<owl:NamedIndividual rdf:about="http://uavl.com/UAVL_Example/tag1#Arizona_Sat">
   <UA_Foundation:contains rdf:resource="http://uavl.com/UAVL_Example/tag1#AZSAT-SET-7"/>
</owl:NamedIndividual>
```

Example – Validation

After identifying the issue using the Reasoning report, the issue is rectified

Operation	Duration
Run build	0.272 s
Calculate build tree task graph	0.016 s
Configure build	0.180 s
Build finished for file system watching	0.001 s
Build started for file system watching	0.012 s
Run main tasks	0.042 s
Run tasks	0.042 s
Load build	0.013 s

Example – Querying (as Grading)

Prefixes – UA Ontology Stack

```
PREFIX foundation:  <http://uaontologies.com/UA_Foundation/UA_Foundation#>

PREFIX agent:   <http://uaontologies.com/UA_Core/UA_Agent#>
PREFIX event:   <http://uaontologies.com/UA_Core/UA_Event#>
PREFIX info:    <http://uaontologies.com/UA_Core/UA_Information#>
PREFIX meas:    <http://uaontologies.com/UA_Core/UA_Measurement#>
PREFIX prov:    <http://uaontologies.com/UA_Core/UA_Provenance#>

PREFIX acq:     <http://uaontologies.com/UA_Domain/UA_Acquisition#>
PREFIX data:    <http://uaontologies.com/UA_Domain/UA_DataManagement#>
PREFIX mission: <http://uaontologies.com/UA_Domain/UA_Mission#>
PREFIX ms:      <http://uaontologies.com/UA_Domain/UA_ModelingAndSimulation#>
PREFIX pm:      <http://uaontologies.com/UA_Domain/UA_ProjectManagement#>
PREFIX req:     <http://uaontologies.com/UA_Domain/UA_Requirements#>
PREFIX sw:      <http://uaontologies.com/UA_Domain/UA_Software#>
PREFIX sa:      <http://uaontologies.com/UA_Domain/UA_SystemArchitecture#>
PREFIX test:    <http://uaontologies.com/UA_Domain/UA_Test#>
PREFIX mat:     <http://uaontologies.com/UA_Domain/UA_Materials#>
```

Example – Querying (as Grading)

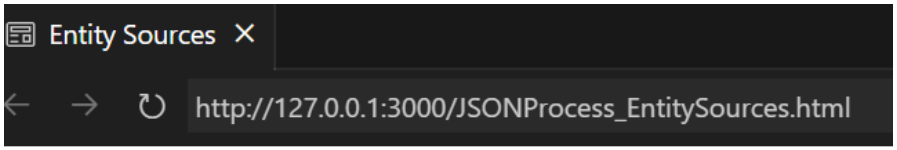
List all entities, their IDs and their sources:

```
SELECT DISTINCT ?EntityName ?EntityID ?EntitySource

WHERE {
    ?Entity sw:hasEntitySource ?EntitySource .
    ?Entity foundation:hasName ?EntityName .
    ?Entity foundation:hasID ?EntityID .
}
```

Example – Querying (as Grading)

List all entities, their IDs and their sources:

```
SELECT DISTINCT ?EntityName ?EntityID ?EntitySource

WHERE {
    ?Entity sw:hasEntitySource ?EntitySource .
    ?Entity foundation:hasName ?EntityName .
    ?Entity foundation:hasID ?EntityID .
}
```



Entity Name	Entity ID	Entity Source
Payload Assembly	991-00003	Duro
Arizona Sat	999-00051	Duro
CDH Assembly	991-00002	Duro
Propulsion System	910-00002	Duro
ADCS	991-00004	Duro
Solar Panel Assembly	910-00001	Duro
Propulsion	AZSAT-PROP-1	Jama
ADCS Subsystem Pointing	AZSAT-ADCS-1	Jama
Power Generation	AZSAT-POWER-1	Jama

Example – Querying (as Grading)

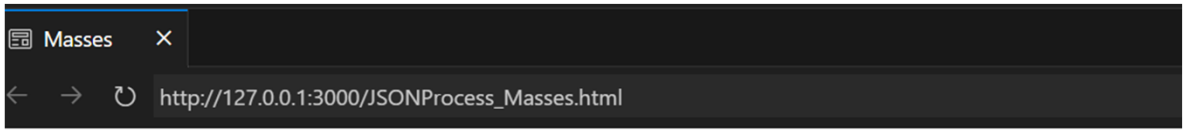
List all components, their descriptions and their masses:

```
SELECT DISTINCT ?SystemName ?SystemDescription ?MassValue ?MassUnit

WHERE {
    ?System a sa:System .
    ?System foundation:hasName ?SystemName .
    ?System foundation:hasNaturalLanguageDescription ?SystemDescription .
    ?SystemMass a meas:Quantity .
    ?SystemMass foundation:specificallyDependentOn ?System .
    ?SystemMassMeasurement a meas:SimpleMeasurement .
    ?SystemMassMeasurement meas:measurementOf ?SystemMass .
    ?SystemMassMeasurement meas:hasValue ?MassValue .
    ?SystemMassMeasurement meas:hasUnit ?MassUnit .
}
ORDER BY ASC (?SystemName)
```

Example – Querying (as Grading)

List all components, their descriptions and their masses:

```
SELECT DISTINCT ?SystemName ?SystemDescription ?MassValue ?MassUnit

WHERE {
    ?System a sa:System .
    ?System foundation:hasName ?SystemName .
    ?System foundation:hasNaturalLanguageDescription ?SystemDescription .
    ?SystemMass a meas:Quantity .
    ?SystemMass foundation:specificallyDependentOn ?System .
    ?SystemMassMeasurement a meas:SimpleMeasurement .
    ?SystemMassMeasurement meas:measurementOf ?SystemMass .
    ?SystemMassMeasurement meas:hasValue ?MassValue .
    ?SystemMassMeasurement meas:hasUnit ?MassUnit .
}
ORDER BY ASC (?SystemName)
```



Component Name	Component Description	Component Mass	Mass Unit
ADCS	ADCS for University of Arizona satellite	100	kilogram
Arizona Sat	University of Arizona satellite	175255	kilogram
CDH Assembly	CDH assembly for University of Arizona satellite	30000	kilogram
Payload Assembly	Payload assembly for University of Arizona satellite	85050	kilogram
Propulsion System	Propulsion system for University of Arizona satellite	105	kilogram
Solar Panel Assembly	Solar panel assembly for University of Arizona satellite	60000	kilogram

Example – Querying (as Grading)

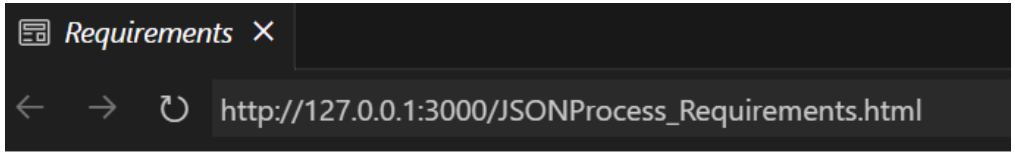
List all requirements and the entities that satisfy them:

```
SELECT DISTINCT ?ReqID ?ReqName ?SystemName

WHERE {
    ?Requirement a req:Requirement .
    ?Requirement foundation:hasID ?ReqID .
    ?Requirement foundation:hasName ?ReqName .
    ?Requirement req:isSatisfiedBy ?System .
    ?System foundation:hasName ?SystemName .
}
```

Example – Querying (as Grading)

List all requirements and the entities that satisfy them:

```
SELECT DISTINCT ?ReqID ?ReqName ?SystemName

WHERE {
    ?Requirement a req:Requirement .
    ?Requirement foundation:hasID ?ReqID .
    ?Requirement foundation:hasName ?ReqName .
    ?Requirement req:isSatisfiedBy ?System .
    ?System foundation:hasName ?SystemName .
}
```



Requirement ID	Requirement Name	Satisfied By
AZSAT-PROP-1	Propulsion	Propulsion System
AZSAT-ADCS-1	ADCS Subsystem Pointing	ADCS
AZSAT-POWER-1	Power Generation	Solar Panel Assembly

Future Work

We are at the beginning of this journey!

- We have tools ready to go: 

- Others will be online soon: 


- And we are in the process of developing the automation pipelines
  - Elaborating use cases: change management, consistency checking, student submission.
  - Reviewing IncQuery.

We are interested in your **use cases**!

- What would **you** like your future engineering grads to have experience with?
- How can we use the DEF to provide those opportunities?

Lessons Learned

- Selection of hub ‘type’ and data representation is key
  - Are you interested in workflows?
  - Are you interested in mappings?
  - Are you interested in translations?
- Not all of the ***n*** connections are necessary
  - Leverage existing connections
  - OpenMBEE, Plug-ins, Repositories
- Tool selection criteria
  - Open-source?
  - License availability?
  - Existing integrations with other tools to be exploited?
- HW Considerations
  - Implement locally, or use a service?
  - Data volume, port requirements, etc. impacts configuration

Acknowledgments

- This material has been produced using funds provided by the *Arizona Technology and Research Initiative Fund*

- Violet Labs
  - for continued support of the DEF project
  - www.violetlabs.com
  - hello@violetlabs.com
- UA IT Team
  - for continued work on hardware configuration
- UA SFWE Students
  - for continued work on DEF development

References

[1] Office of the Deputy Assistant Secretary of Defense (Systems Engineering) [ODASD (SE), "DAU Glossary: Digital Engineering," Defense Acquisition University (DAU), 2017. [Online]. Available: https://www.dau.edu/glossary/Pages/Glossary.aspx. [Accessed 24 August 2023].

[2] Violet Labs, 2023. [Online]. Available: https://www.violetlabs.com. [Accessed 24 August 2023].

[3] OpenCAESAR, "Ontological Modeling Language 2.0.0 Living Standard", 2023. [Online]. Available https://www.opencaesar.io/oml/. [Accessed 24 August 2023].

[4] A. Patel and S. Jain, "Present and future of semantic web technologies: a research statement," Int. J. Comput. Appl., pp. 1–10, 2019.

[5] W3C, "Resource Description Framework (RDF) Concepts and Abstract Syntax", 2014. [Online]. Available https://www.w3.org/TR/rdf11-concepts/. [Accessed 24 August 2023]

[6] W3C, "OWL 2 Web Ontology Language Document Overview (Second Edition)", 2012. [Online]. Available https://www.w3.org/TR/owl2-overview/. [Accessed 24 August 2023]

[7] W3C, "SPARQL 1.1 Query Language", 2013. [Online]. Available https://www.w3.org/TR/sparql11-query/. Accessed [24 August 2023]

[8] R. Arp, B. Smith, and A. Spear, "Building Ontologies with Basic Formal Ontology". MIT Press, 2015.

[9] R. Rudnicki, "An Overview of the Common Core Ontologies," CUBRC Report, 2019.

[10] ISO, IEC, and IEEE, "ISO/IEC/IEEE 42010:2011(E) - Systems and software engineering - Architecture description." Geneva, Switzerland, 2011.

[11] Joint Committee for Guides in Metrology, "International Vocabulary of Metrology (Fourth Edition)", 2021. JCGM-WG2-CD-01

Thank you.
Questions?

joegregory@arizona.edu