The Digital Engineering Factory: Considerations, Current Status, and Lessons Learned

INCOSE IS 2024

7/3/2024

PRESENTER Dr. **Joe Gregory** *Postdoctoral Research Associate*
Department of Systems & Industrial Engineering

Co-authors: Dr. Alejandro Salado, Sharon O'Neal, Richardo Larez, Niko Martell, CJ Reda, Evan Martin, Matthew Colson, John Masterson, David Armenta

THE UNIVERSITY OF ARIZONA

Contents

- Digital Engineering (for Students)
- Overview of the ‘Digital Engineering Factory’
- Demo
- Key Considerations and Lessons Learned (so far…)
- Conclusions

© J. Gregory, A. Salado

- Digital Engineering (for Students)
- Overview of the ‘Digital Engineering Factory’
- Demo
- Key Considerations and Lessons Learned (so far…)
- Conclusions

© J. Gregory, A. Salado

Digital Engineering

Software

Electronics

Aerodynamics

Mechanical

© J. Gregory, A. Salado

THE UNIVERSITY OF ARIZONA

Digital Engineering

Software

Electronics

Aerodynamics

Mechanical

This is not good enough in industry!

But what do we do across engineering curricula?

© J. Gregory, A. Salado

THE UNIVERSITY OF ARIZONA

Digital Engineering

SIE 431

SFWE 302

SIE 458

SFWE 403

© J. Gregory, A. Salado

THE UNIVERSITY OF ARIZONA

Digital Engineering

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

© J. Gregory, A. Salado

THE UNIVERSITY OF ARIZONA

Digital Engineering (for Students)

© J. Gregory, A. Salado

THE UNIVERSITY OF ARIZONA

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]."

[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].

© J. Gregory, A. Salado

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]."

[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].

© J. Gregory, A. Salado

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]."

[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].

© J. Gregory, A. Salado

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]."

[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].

© J. Gregory, A. Salado

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]."

[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].

© J. Gregory, A. Salado

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]."

The Objective of the 'Digital Engineering Factory' is to provide the following benefits:

- Students have access to the tools required to support their courses,
- Over multiple courses, students observe a complete end-to-end process,
- Students can observe the consequences of the decisions downstream,
- Students gain experience working collaboratively in a digital environment,
- Students can evaluate the work of others more effectively,
- Instructors can make use of automated grading through model validation pipelines.

[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].

© J. Gregory, A. Salado

- Digital Engineering (for Students)
- Overview of the ‘Digital Engineering Factory’
- Demo
- Key Considerations and Lessons Learned (so far…)
- Conclusions

© J. Gregory, A. Salado

Digital Engineering Factory (DEF) Use Cases

ID	Source	Course Name / Use Case Description	Tool Type / **Specific Tool**
SW-1	SFWE-101	Introduction to Software Engineering	IDE **GitLab**
SW-2	SFWE-301	Software Requirements Analysis and Test	Requirements Test Planning
SW-3	SFWE-401	Software Assurance and Security	IDE Static Code Analyzer
SW-4	SFWE-402	Software DevSecOps	Requirements Project Management Static Code Analyzer Containerization **GitLab**
SW-5	SFWE-403	Software Project Management	IDE **GitLab** Project Management
SW-6	SFWE-498	Senior Design Project	IDE CAD **GitLab** System Architecture Requirements Test Planning
SYS-1	SIE-458	Model-Based Systems Engineering	**Cameo Systems Modeler**
SYS-2	JPL Dragon	Modelica generation from Systems Modeling Language (SysML) v1 state machines	**Cameo Systems Modeler** **OpenMBEE**
SYS-3	JPL Dragon	Document generation from SysML v1 models	**Cameo Systems Modeler** **OpenMBEE**
SYS-4	JPL Dragon	Traceability from test strategy to requirements	Requirements Test Planning
SYS-5	JPL Dragon	Analysis using inputs from SysML v1/v2 models	SysML v1 SysML v2 Analysis Script

Identified 11 use cases

- SysEng x5, SW x6
- 7 reflect current courses
- 4 to be incorporated into courses

Identified required tools

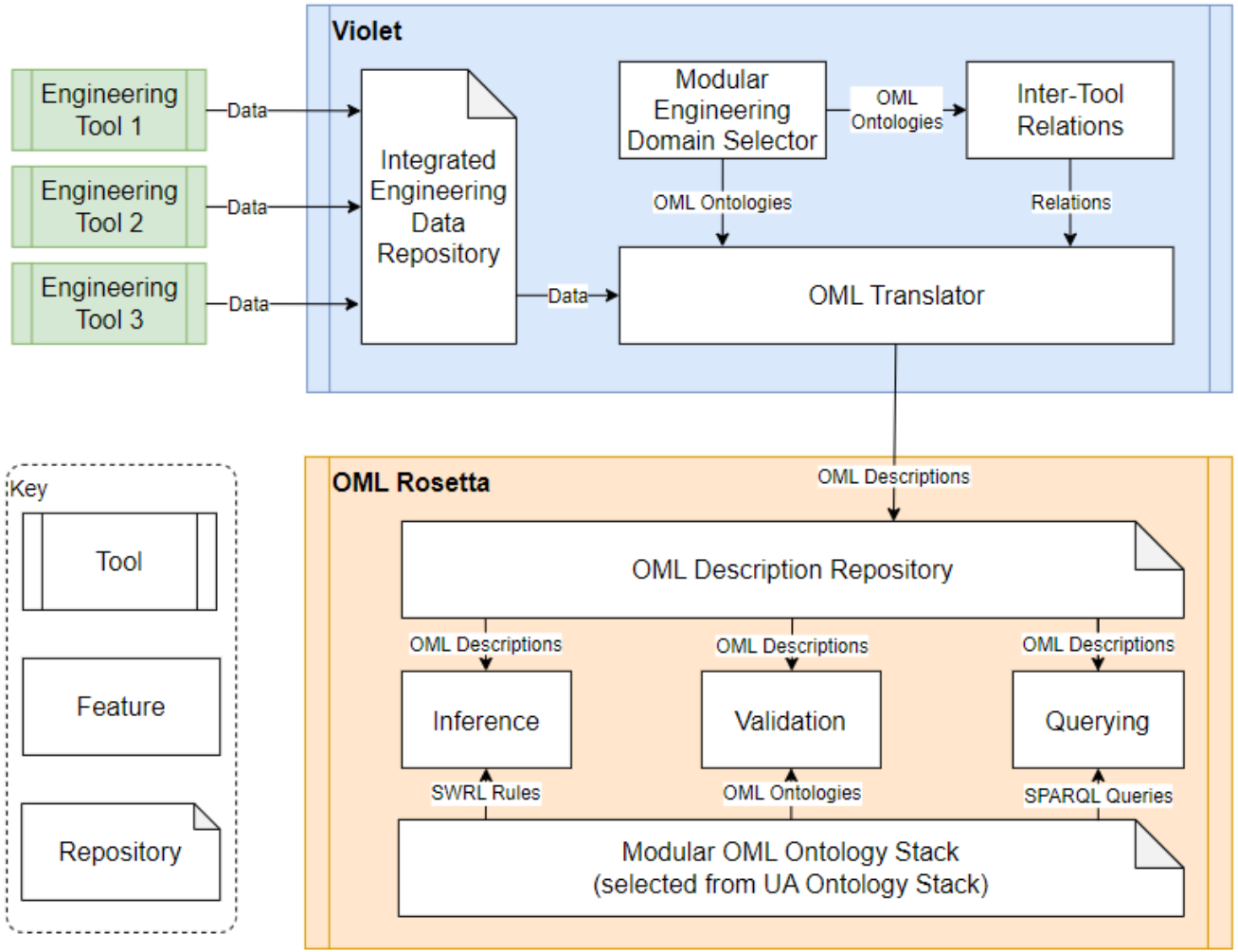
- Some specifically requested
- Others not: trade-off

Identified required connections

- how do we integrate tools?
- how do review entire dataset?

THE UNIVERSITY OF ARIZONA

Digital Engineering Factory (DEF) Architecture



Hub and Spoke

Technical Integration

- Violet Labs

Data Integration

- Ontological Modeling Language
- OML Rosetta

Enables SWTs

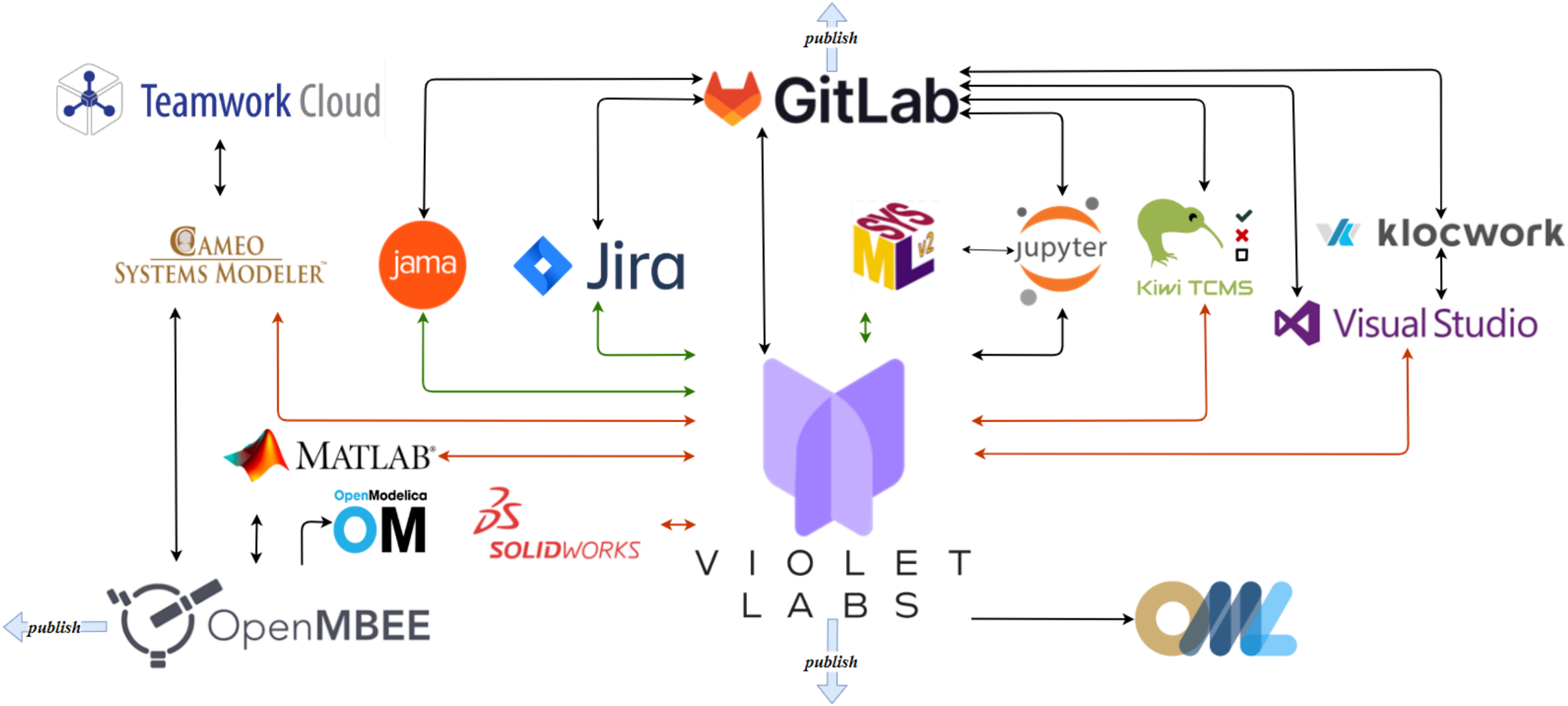
- Inference
- Validation
- Querying

© J. Gregory, A. Salado

THE UNIVERSITY OF ARIZONA

Digital Engineering Factory (DEF)

Integration complete
Integration currently one-way
Integration under development



- Digital Engineering (for Students)
- Overview of the ‘Digital Engineering Factory’
- Demo
- Key Considerations and Lessons Learned (so far…)
- Conclusions

© J. Gregory, A. Salado

Demo Overview

- SIE 250 – An Introduction to Systems Engineering
- Three teams of 6 students
- Design and Test a Lego Rover

Student	Responsibility	Tool
1	SE Management Planning	Jira Violet
2	Needs and Requirements Validation	Jama SysML v2 (Actions)
3	Architecture Verification and Test Plans	SysML v2 (Parts) SysML v2 (Actions)
4	Detailed HW Design Integration Plan	BrickLink Studio 2.0 SysML v2 (Actions)
5	Detailed SW Design	Jupyter / Visual Studio
6	Tests Integration and Deployment	SysML v2 (Parts)

© J. Gregory, A. Salado

THE UNIVERSITY OF ARIZONA

THE UNIVERSITY OF ARIZONA

College of Engineering IT Support

College of Engineering
Digital Engineering Factory

Home | Resources | Tools

Tools

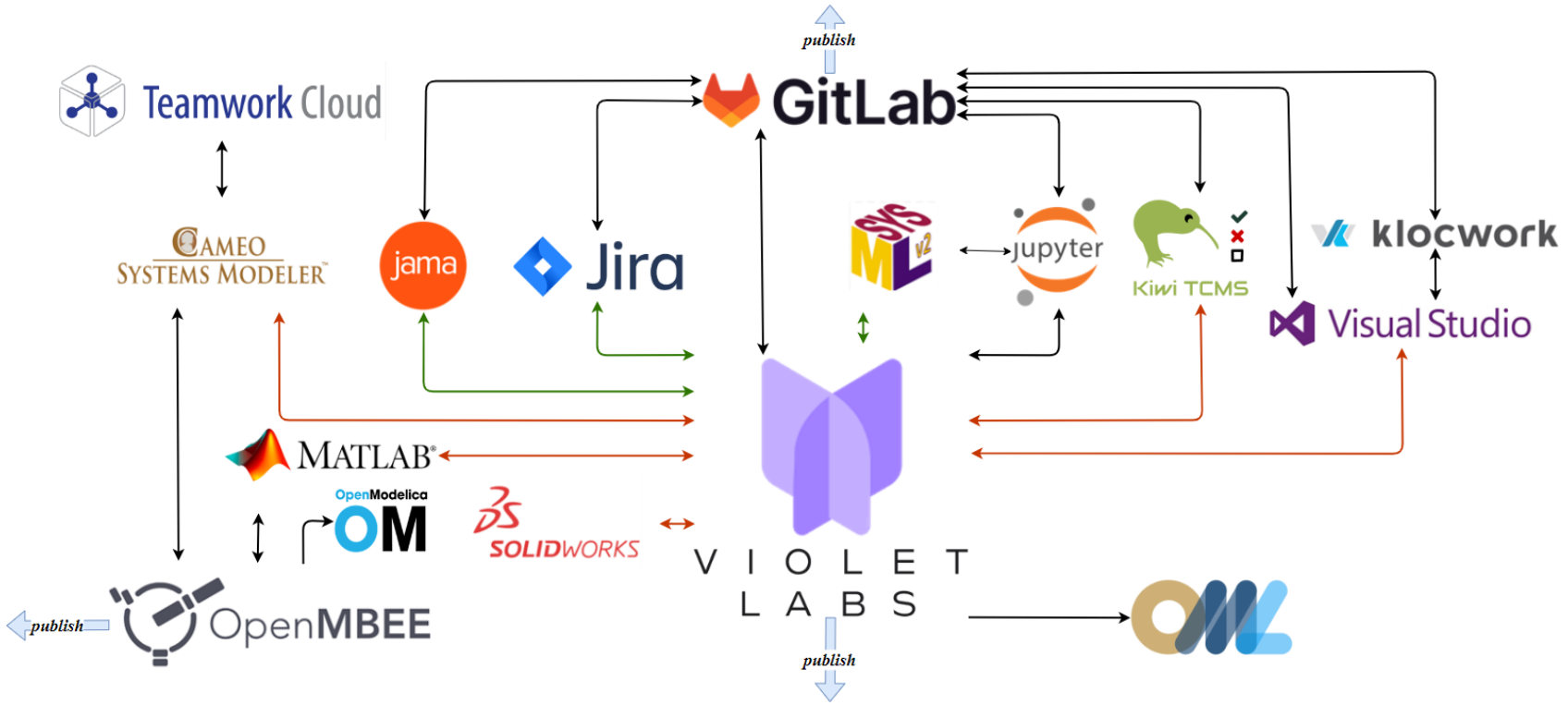
- Digital Engineering (for Students)
- Overview of the ‘Digital Engineering Factory’
- Demo
- Key Considerations and Lessons Learned (so far…)
- Conclusions

© J. Gregory, A. Salado

Key Considerations

Software Architecture

- Minimize required connections: $\frac{n(n-1)}{2} \rightarrow n$
- While maintaining publication options and leveraging existing connections

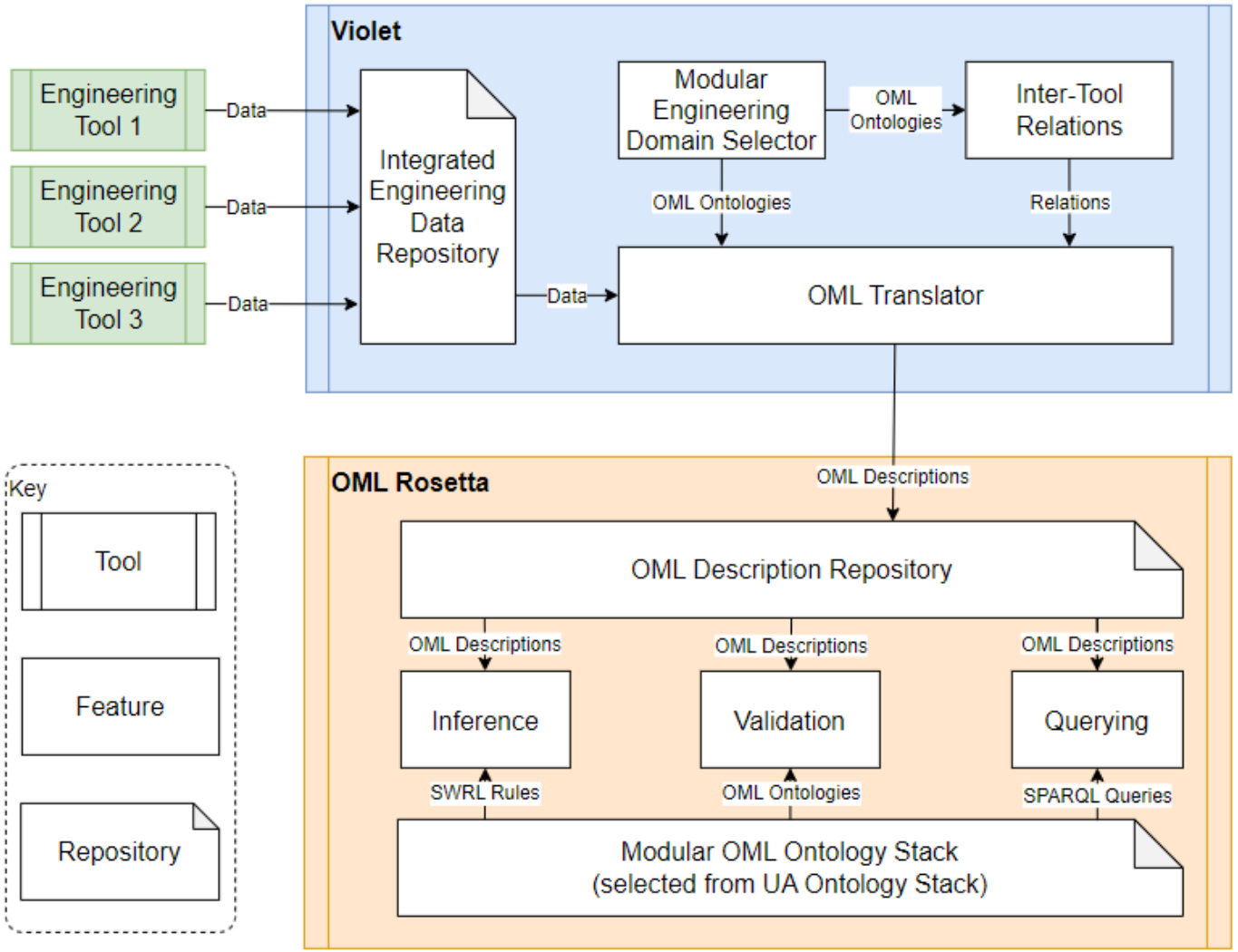


THE UNIVERSITY OF ARIZONA

Key Considerations

Hub Selection

- Traceability of entities, analysis workflows, graph database.
- Are relevant domains supported (consider domain-specific tools),
- Freedom to define underlying data model?

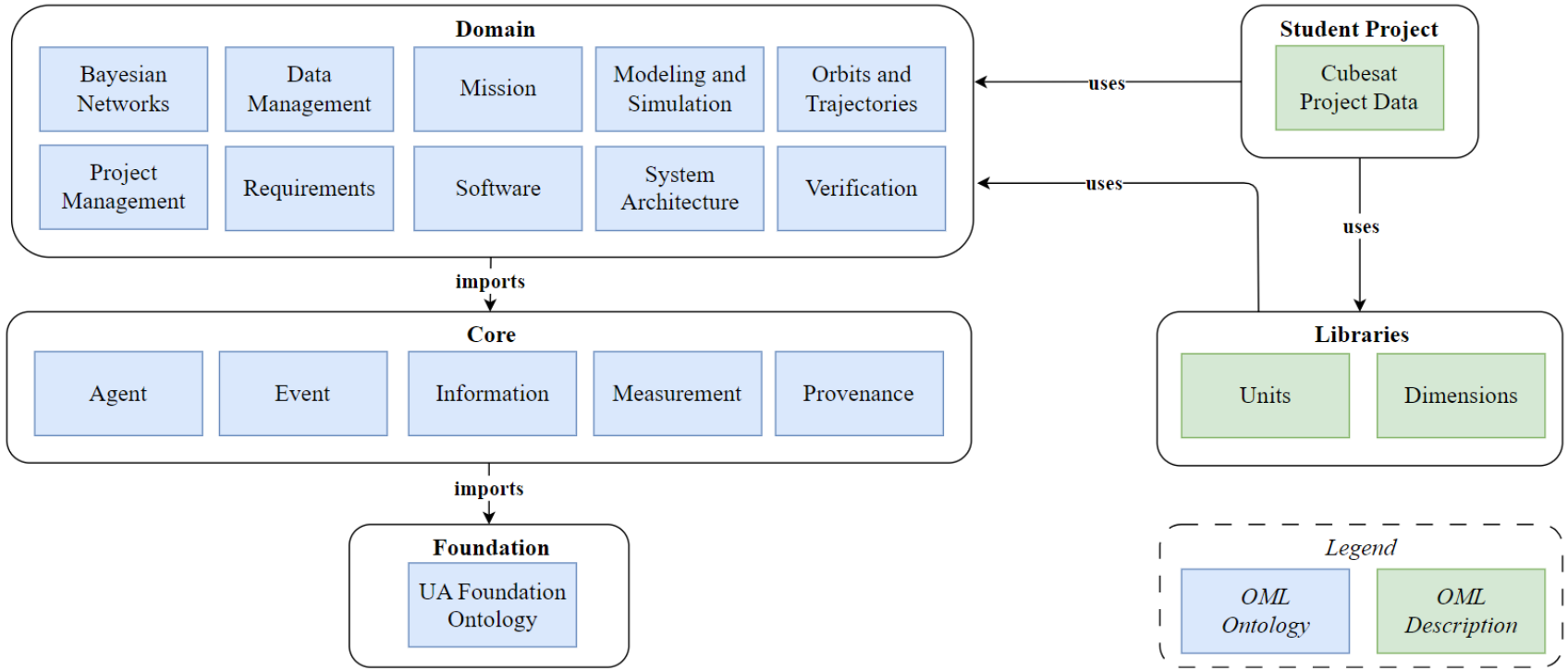


THE UNIVERSITY OF ARIZONA

Key Considerations

Underlying Data Model

- Expressivity requirements vs. Reasoning capability,
- Foundation provides consistency, but locks in assumptions,
- Reuse of existing domain ontologies, relevant standards.



THE UNIVERSITY OF ARIZONA

Key Considerations

Tool Selection

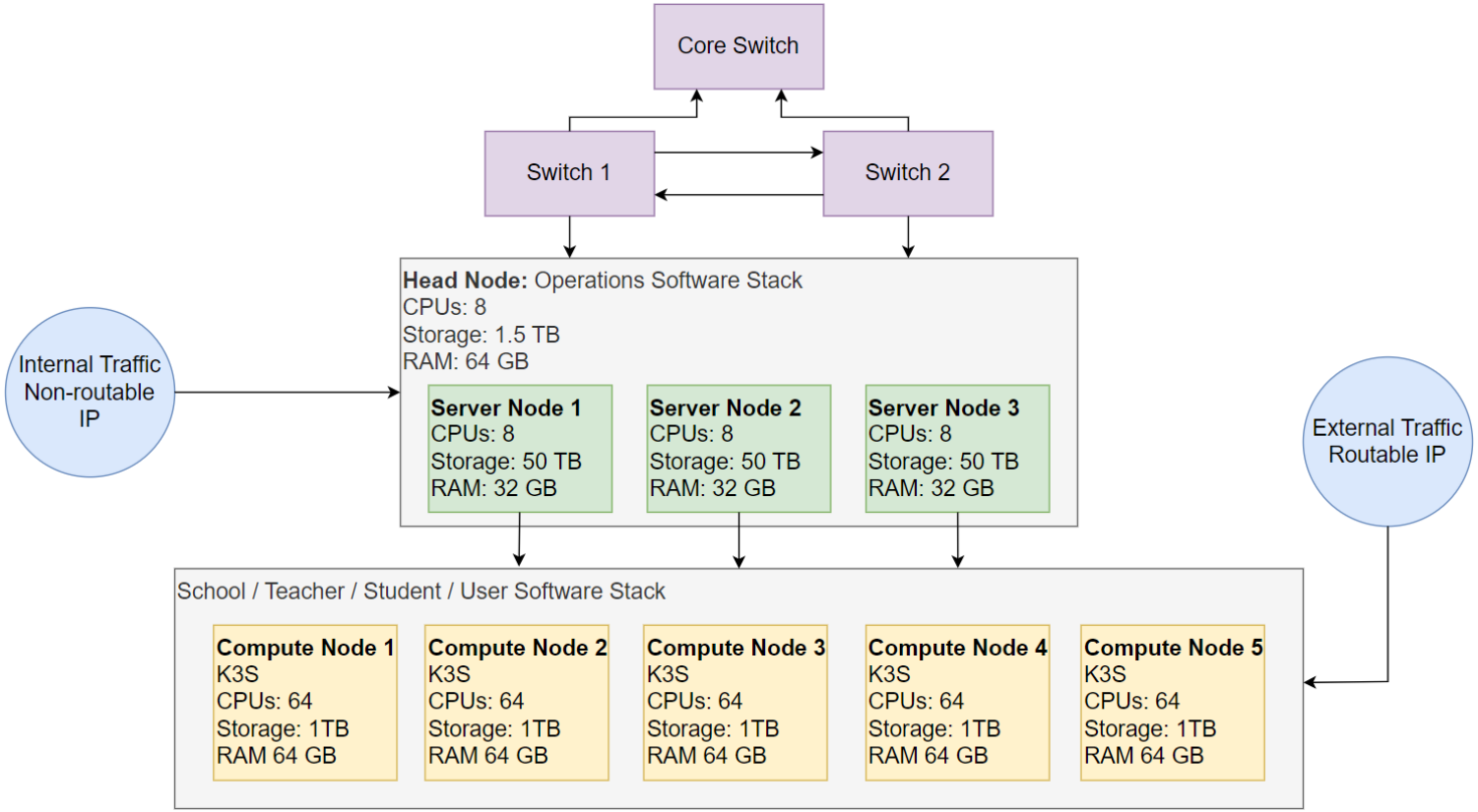
- Are class requirements met?
  - Often demonstrating few tool capabilities.
- Open-source?
  - UA license availability
- Integrations
  - Existing integration with Violet?
  - Existing direct (tool-to-tool) integrations?
  - Conformance with integration standards (REST API, Graph QL)

© J. Gregory, A. Salado

Key Considerations

Hardware

- Number of users?
  - Total, Simultaneous,
  - Volume of data transfer.
- Storage requirements?
  - Storage per user,
  - Installation storage.
- What nodes and switching capabilities are required?
- VM, Containerization and distribution?
  - e.g., Proxmox, Kubernetes, K3s, OpenStack, Docker, RedHat
  - Up-front vs. recurring costs?



© J. Gregory, A. Salado

THE UNIVERSITY OF ARIZONA

- Digital Engineering (for Students)
- Overview of the ‘Digital Engineering Factory’
- Demo
- Key Considerations and Lessons Learned (so far…)
- Conclusions

© J. Gregory, A. Salado

Conclusions

- The DEF remains under development:
  - We continue to integrate tools (and make bi-directional),
  - We continue to develop the UAOS,
  - We will need to optimize as we scale up.

- Key Decisions
  - SW Architecture, Data Model, Selection of Hub and Tools, HW Setup

- We have deployed on multiple, small-scale use cases.

- Next step will be to deploy on a larger scale:
  - Multiple tools, larger teams, change management, version control.
  - Incorporate Dashboards and Query capabilities.

© J. Gregory, A. Salado

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 Eng IT Team
for continued work on hardware configuration

UA SFWE Students
for continued work on DEF development

© J. Gregory, A. Salado

THE UNIVERSITY OF ARIZONA

THANK YOU

joegregory@arizona.edu

THE UNIVERSITY OF ARIZONA

Requirements Table

Req ID	Req Name	Req Text	Satisfied by	Verified by
SIE250_1-SYS-1	Terrain Requirement	The system shall be capable of operating in both indoor and outdoor terrains.		
SIE250_1-SYS-2	Remote Control Requirement	The system shall have remote control capacity between 3-20 ft.		
SIE250_1-SYS-3	Autonomous Requirement	The system shall become autonomous if there is no signal through the remote control after 30 seconds.		
SIE250_1-SYS-4	Cargo Requirement	The system shall be able to carry cargo up to 40 lbs.		CargoTest
SIE250_1-SYS-5	Spinning Requirement	The system shall observe 360 degrees in 30 second intervals when the system is turned on.		
SIE250_1-SYS-6	Trash Requirement	The system shall grab food container-type trash and dispose of it.		
SIE250_1-SYS-7	Mass Requirement	The system mass shall be less than 5kg	Group1_Rover_Mass	MassTest

© J. Gregory, A. Salado

Tasks Table

TaskID	Description	StudentName	Outputs
SIE2501-10	Import SYSMLV2 Files into Violet	Dominick D'Emilio	
SIE2501-16	Create a Verification Plan for the System	Ashley Ferrell	Rover_Verification_Strategy
SIE2501-17	Create a detailed Hardware design (CAD model) for System	Jace Cascarini	SIE250_CADModel
SIE2501-22	Construct Rover and all Hardware Components	Sebastian Alcantar	
SIE2501-23	Integrate all hardware and software to complete system architecture	Sebastian Alcantar	
SIE2501-24	Run Verification Tests according to Verification Plan	Sebastian Alcantar	
SIE2501-27	Complete mass analysis verification activity	Katelyn Miller	
SIE2501-3	Define System Requirements	Katelyn Miller	Terrain Requirement, Trash Requirement, Autonomous Requirement, Remote Control Requirement, Cargo Requirement, Spinning Requirement, Mass Requirement
SIE2501-4	Refine Stakeholder Needs	Katelyn Miller	
SIE2501-7	Create a System Architecture	Jace Cascarini	ProvideRoverPropulsion, Group1_Rover
SIE2501-9	Create and Assign Jira Tasks	Carter Buss	

© J. Gregory, A. Salado

THE UNIVERSITY OF ARIZONA