33rd Annual INCOSE international symposium
hybrid event
Honolulu, HI, USA
July 15 - 20, 2023

Connecting the Dots: Digital Threads Benefits and Best Practices

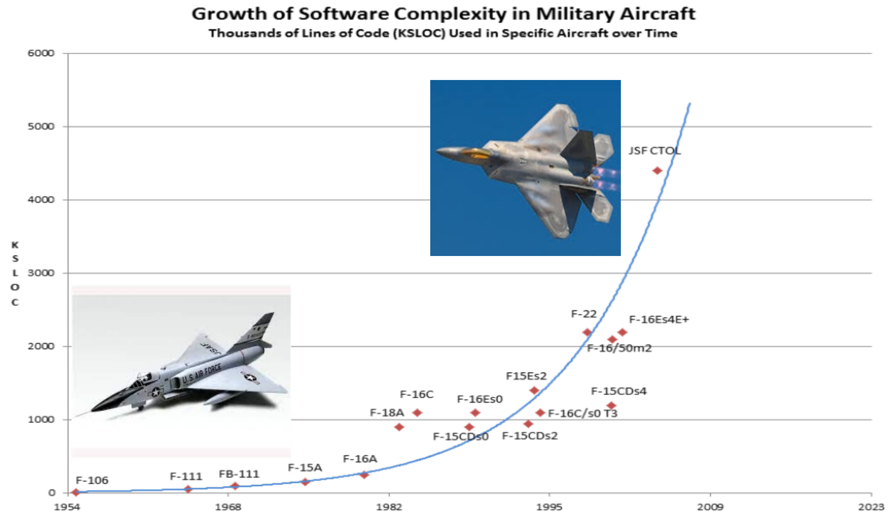
Eran Gery – Global Solutions Lead, IBM Engineering Solutions

Outline

- Today's complex systems challenges
- Digital threads and how they can help
- OSLC as a framework for a digital back bone
    - Digital continuity
    - Global configurations
    - Data exchange
    - Cross lifecycle analysis
- Summary and additional resources

The need for engineering digitization

- Technology of smart products is evolving fast and somewhat unpredictable way... This imposes multiple challenges for the manufacturers with <u>current</u> engineering practices…
- Dealing with increasing complexity with unpredictable technological disruptions
  - Autonomous functions
  - Electrification
- The need for speed – responding to competitive and environmental changes
- Meeting growing industry regulatory demands in areas like safety and cybersecurity
- Lack of skilled engineers requires higher efficiency – doing more with less



The premise of engineering digitization is to increase engineering efficiencies to better meet these challenges!

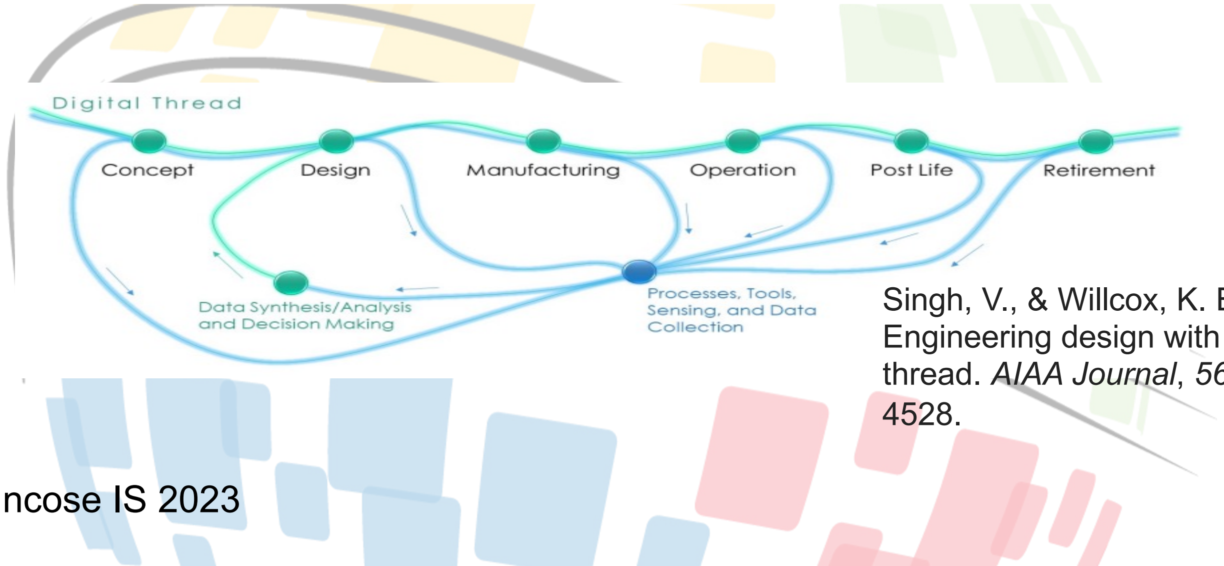
The challenge – effective engineering in context of multidisciplinary development environments…

- Engineering data is siloed across teams and applications (ASOTs)
- Implies deficiencies of key engineering objectives and activities:
  - Continuity and consistency of data across related artifacts
  - Proper assessment of impact and manage changes across all ASOTs
  - Gaining the right insights to conduct engineering assessments by enabling joint digital viewpoints
  - Management of configurations baselines and branches across all datasets
  - effective collaboration across all stakeholders to foster agile engineering

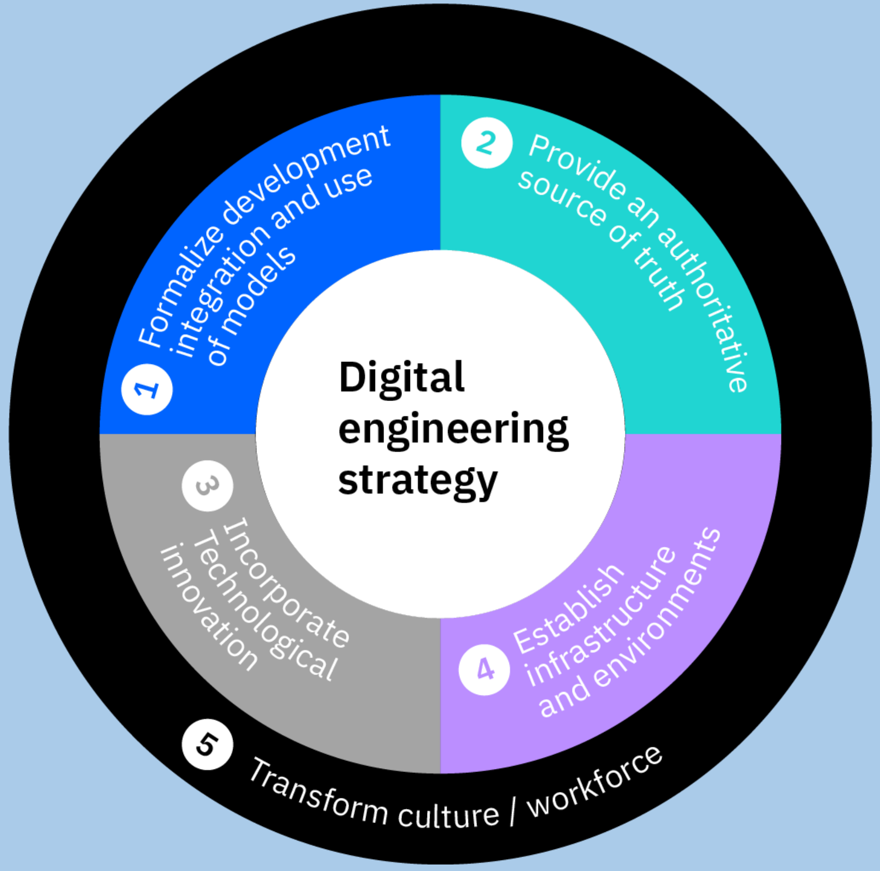
Industry vision: Digital Engineering, Digital threads

Digital threads is "a data-driven architecture that links together information generated from across the product lifecycle and is envisioned to be the primary or authoritative data and communication platform for a company's products at any instance of time."

Digital threads architecture supports the broader vision of digital engineering



Singh, V., & Willcox, K. E. (2018). Engineering design with digital thread. *AIAA Journal*, *56*(11), 4515-4528.



"...such engineering environments will allow DoD and industry partners to evolve designs at conceptual phase, reducing the need for expensive mockups, premature design lock, and physical testing." [1]

Key digital threads related use cases

How do I make sure that my design is consistent with my requirements?

What is the impact of modifying this requirement across all disciplines?

Have I propagated this change to all software and hardware artifacts?

Is my system stable enough to add more functionality?

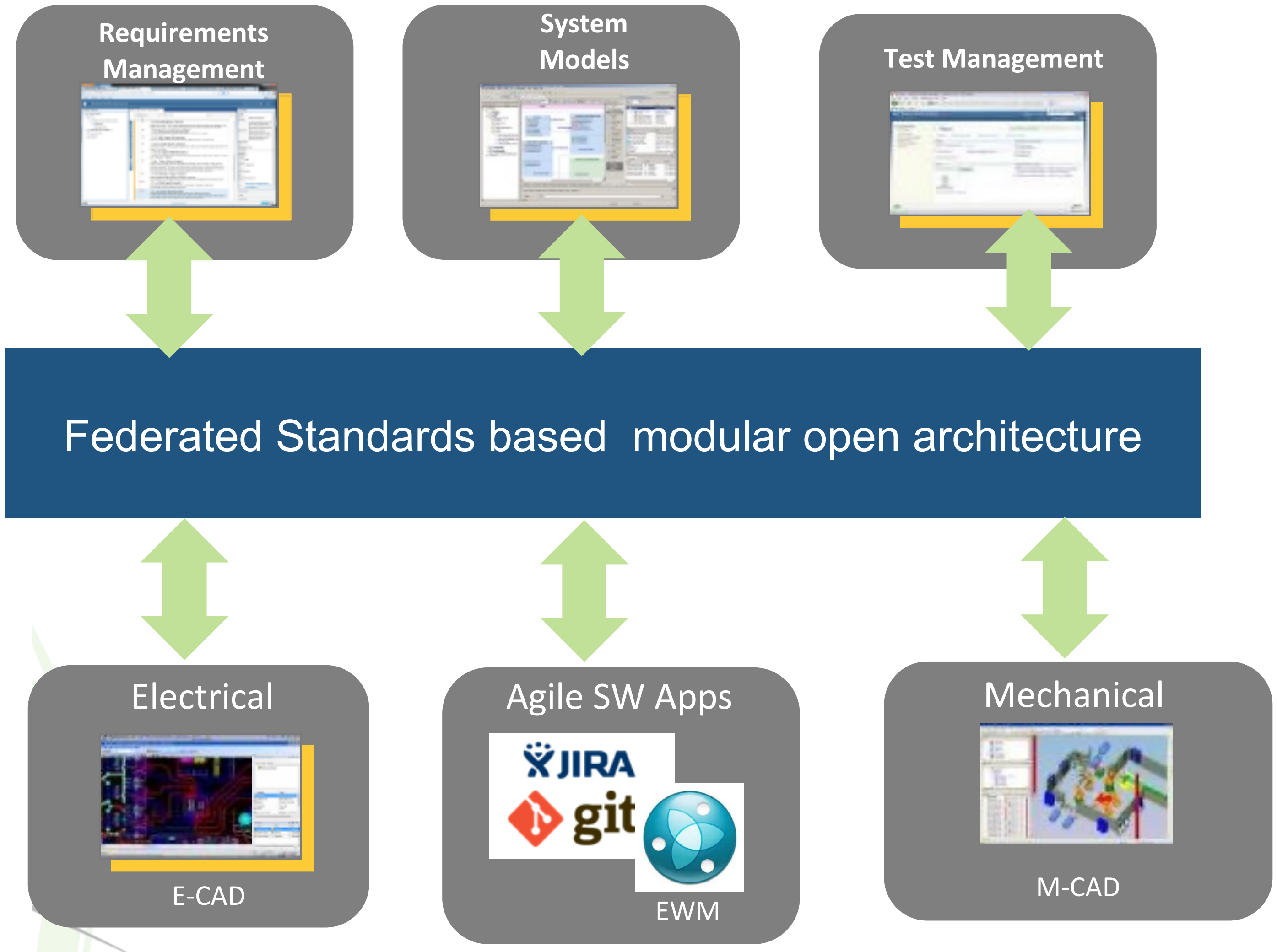
Is my electrical design baseline consistent with my software baseline?

Which variants use this component?

How do I prove that my test plan covers all the requirements?

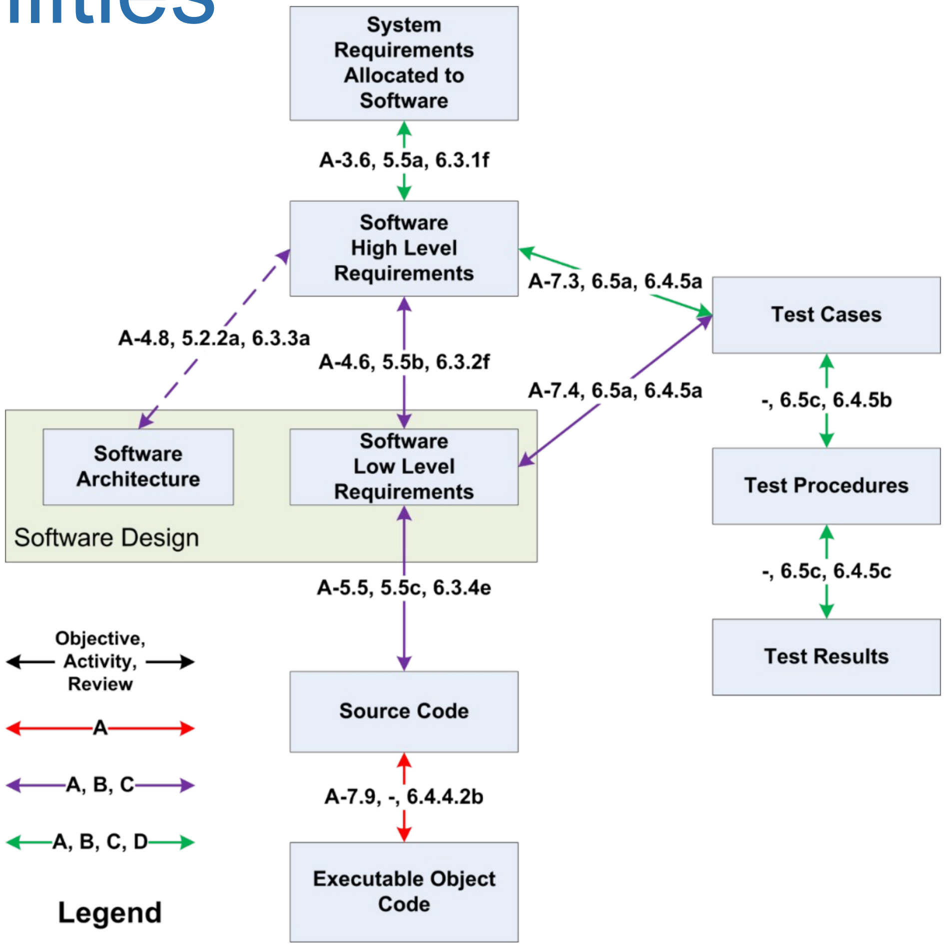
Approaches for digital engineering backbone

- Point to point tool integrations
  - Does not scale, consistency challenges
  - Does not support cross domain digital viewpoints
- Centralized approaches
  - Centralized repository that manages all the data
  - Data replication and consistency with the domain tools
- MBSE backbone
  - Import various domain data to MBSE tool
  - Data replications, consistency challenges, does not scale
- Link management applications
  - Centralized link repository across ASOTs
  - ASOTs not part of the linking model
- OSLC (Linked data Federated)
  - Collaborative management of ASOTs
  - Fosters data consistency across lifecycle
  - Enables digital views points



Key digital threads enabler capabilities

- *Digital continuity*: establish digital information models based on standard resource types and relationships across all domains tools
- Enable *cross domain data exchange* through standard data representations
- *Global configuration management*: manage consistency across all engineering data sources using cross tools configuration management
- *Cross lifecycle analytics and viewpoints:* produce the necessary insights and evidence from across all domain tools
- *Integrated change and process management* across all engineering data and tools

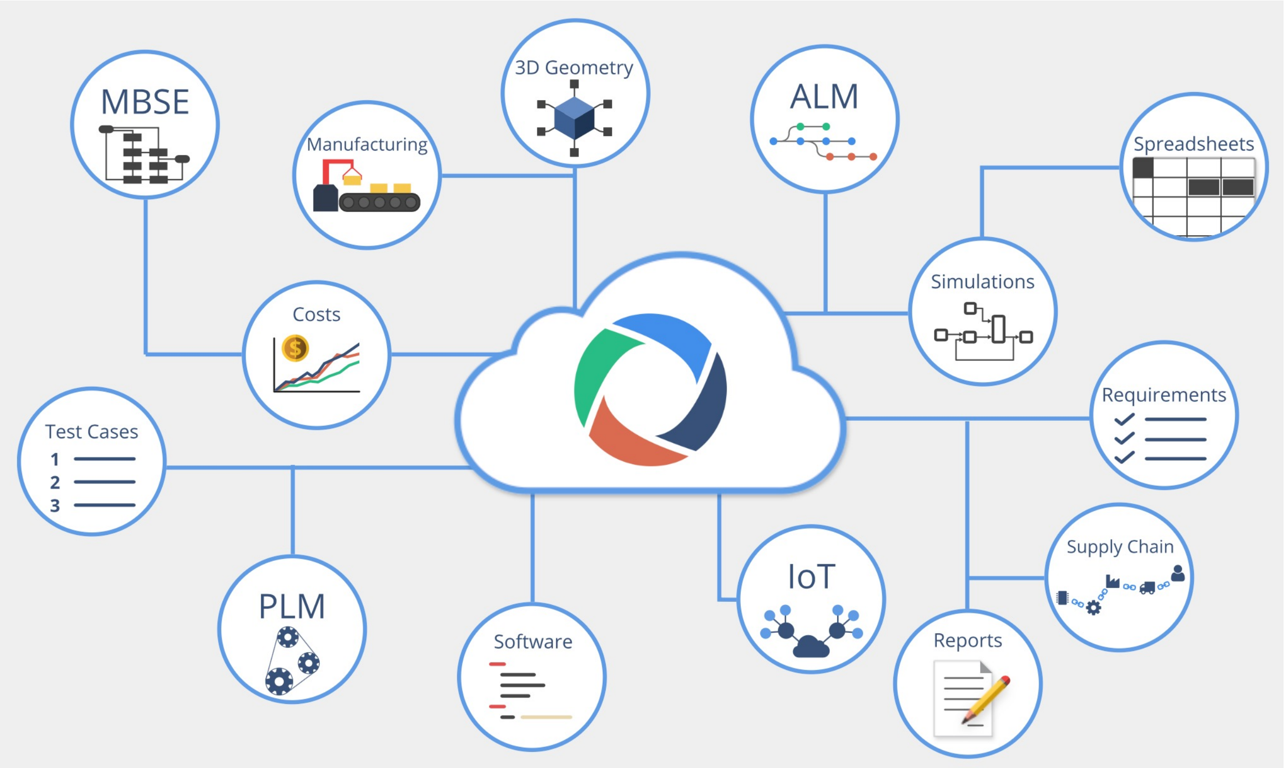


information model required by DO178 DALs

Open Services for lifecycle (OSLC)

A lifecycle integration framework based on open data model and services standards across a federated set of tools

- Specifies standard lifecycle information models based on W3C ontologies
  - Standard resource representation
  - Linking across resources
- Open world assumption: minimal assumptions on data models and services
  - Enable discovery
- Enable collaboration across tools based on standard REST services for complete modularity
- Enable integration of existing tools with no assumption of how they are implemented
- Enable services for cross lifecycle viewpoints and analytics



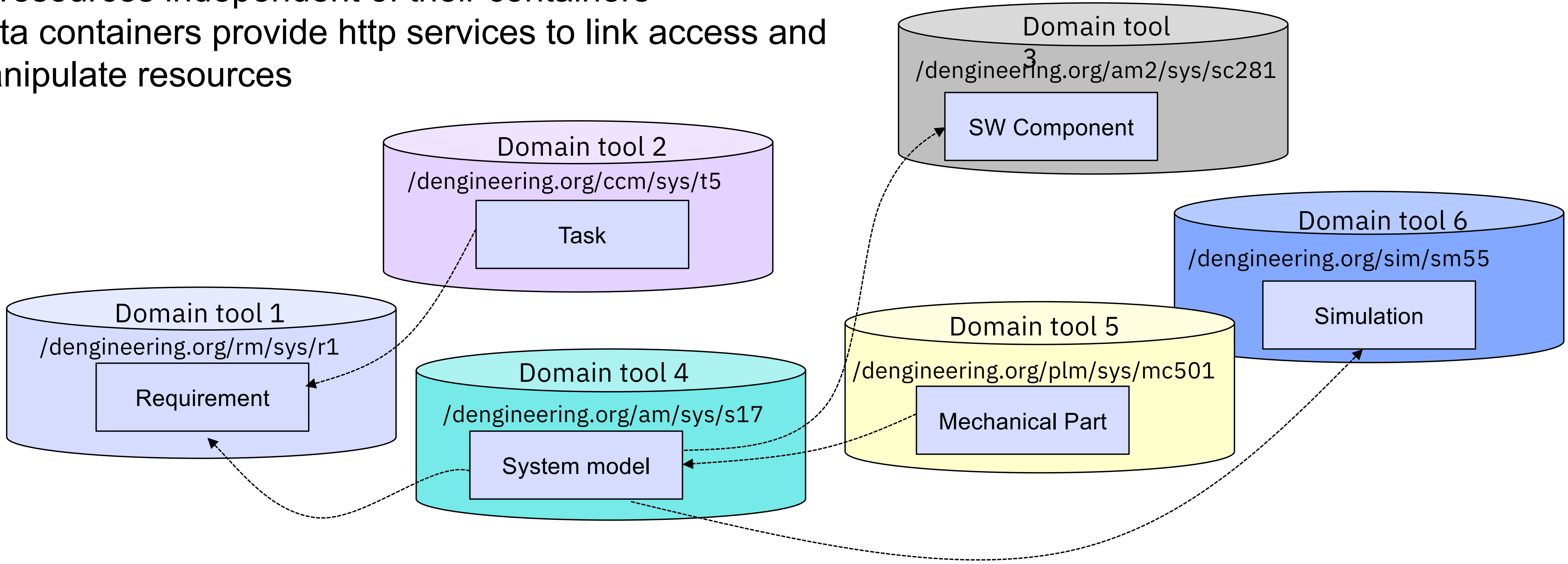
OASIS

OSLC Member Section

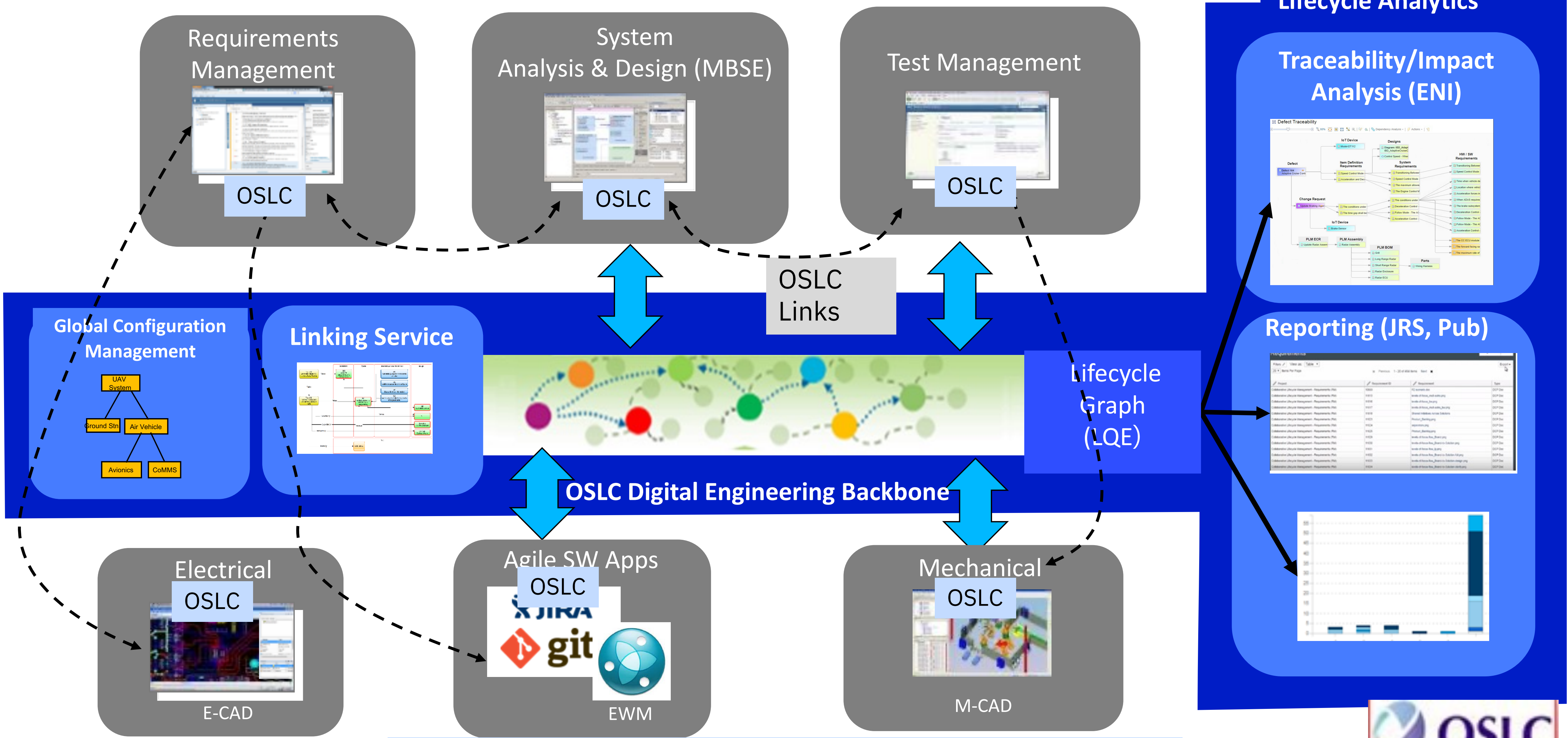
http://oasis-oslc.org

The foundation: Linked data (w3c)

- Lifecycle objects (resources) are identified by http URLs and described using vocabularies (ontologies)
- Enables lifecycle information models with relationships across all resources independent of their containers
- Data containers provide http services to link access and manipulate resources



A Digital Thread architecture based on OSLC services



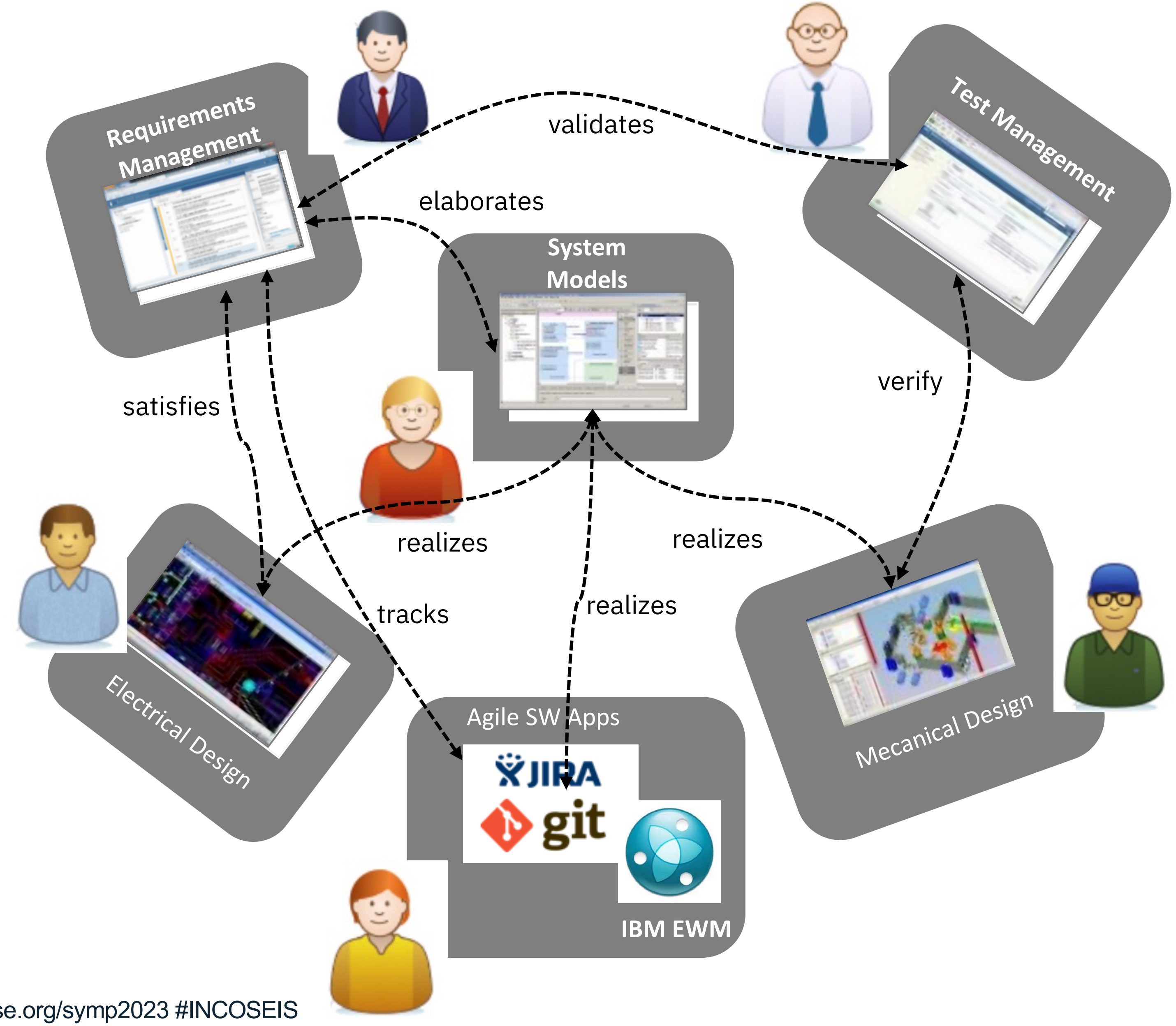
Domain tools provide/consume OSLC APIs

Incose IS 2023

Collaborative linking across domain applications

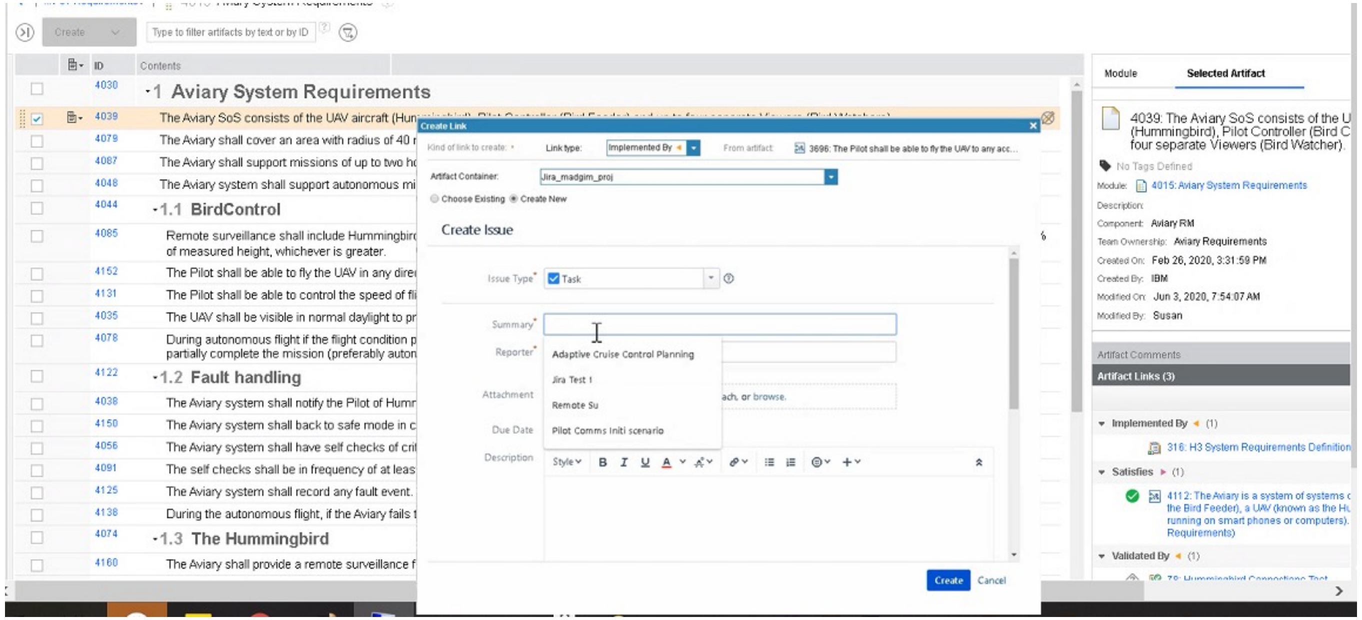
Digital continuity

- Domain tools (ASOTs) extended with OSLC service (adapter)
- Leverage OSLC linking services to establish links
- The linking is a federated concept based on collaboration across tools
- Enables Link discovery
- Enables Link validity



How OSLC linking works?

- Services OSLC providers (tools) offer other providers to create links
  - Selection service
    - Delegated HTML page that allows selection of an element in the provider tool for linking. The requests returns a URL of the linked element.
  - Element preview service
    - Delegated HTML page that provides information on a linked element in the context of the source tool
  - Element view service
    - The requesting tool switches context to the element page in the hosting tool

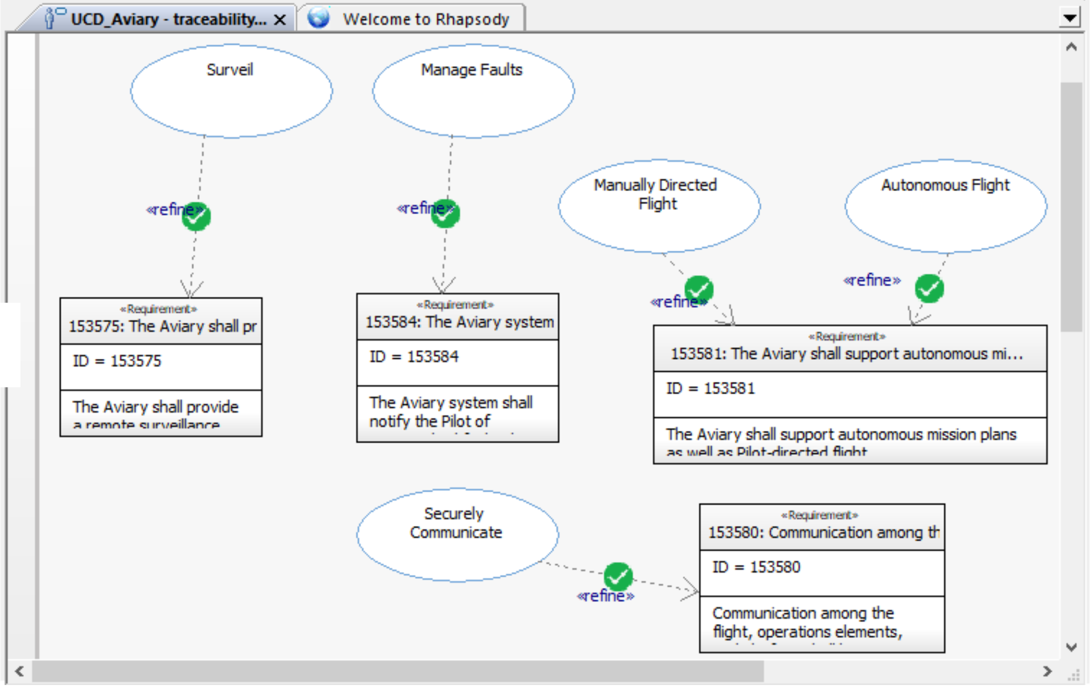


A requirement with a preview page to a Jira issue

Link validity: ensuring data consistency

Contents	Traced By Architecture Element	Refined By Architecture Element
1 Aviary System Requirements		
The Aviary SoS consists of the UAV aircraft (Hummingbird), Pilot Controller (Bird Cage) and up to four separate Viewers (Bird Watcher).		
The SoS system shall cover area with radius of 40 miles		
SoS mission shall take up to two hours		
The SoS system shall support autonomous mission based on pre-defined flight plan.		
1.1 BirdControl	Requirements Diagram: Manually Directed Flight	
Remote surveillance shall include Hummingbird position in latitude, longitude, altitude, to an accuracy of ±1 meter and height above surface, accurate to ±.2 cm or 1% of measured height, whichever is greater.	UseCase: Surveil	SequenceDiagram: Surveil Video Stream
The Pilot shall be able to fly the UAV in any direction with respect to the UAV's "forward looking" position.	UseCase: Manually Directed Flight	SequenceDiagram: Set Rotation
The Pilot shall be able to control the speed of flight from 0 to maximum speed in any direction in a smooth fashion.	UseCase: Manually Directed Flight	
The Aviary flying unit shall be visible in normal daylight to prevent collisions.		
During autonomous flight if the flight condition prevent, from the system, to complete its mission, the Aviary system shall change its flight plan such it will fully or partially complete the mission (preferably automatically or by remote taking over by a pilot)		
1.2 Fault handling	Requirements Diagram: Manage Faults Use Case	
The Aviary system shall notify the Pilot of Hummingbird faults that may negatively impact mission success. Modified within 0.5s of their	UseCase: Manage Faults	

Validity markers in a requirements tool

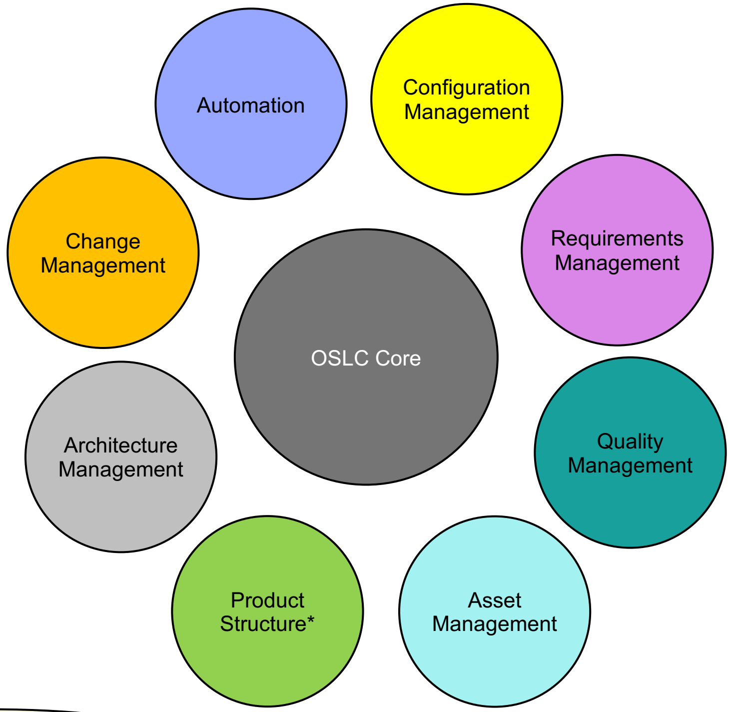


Validity indicators in an MBSE tool

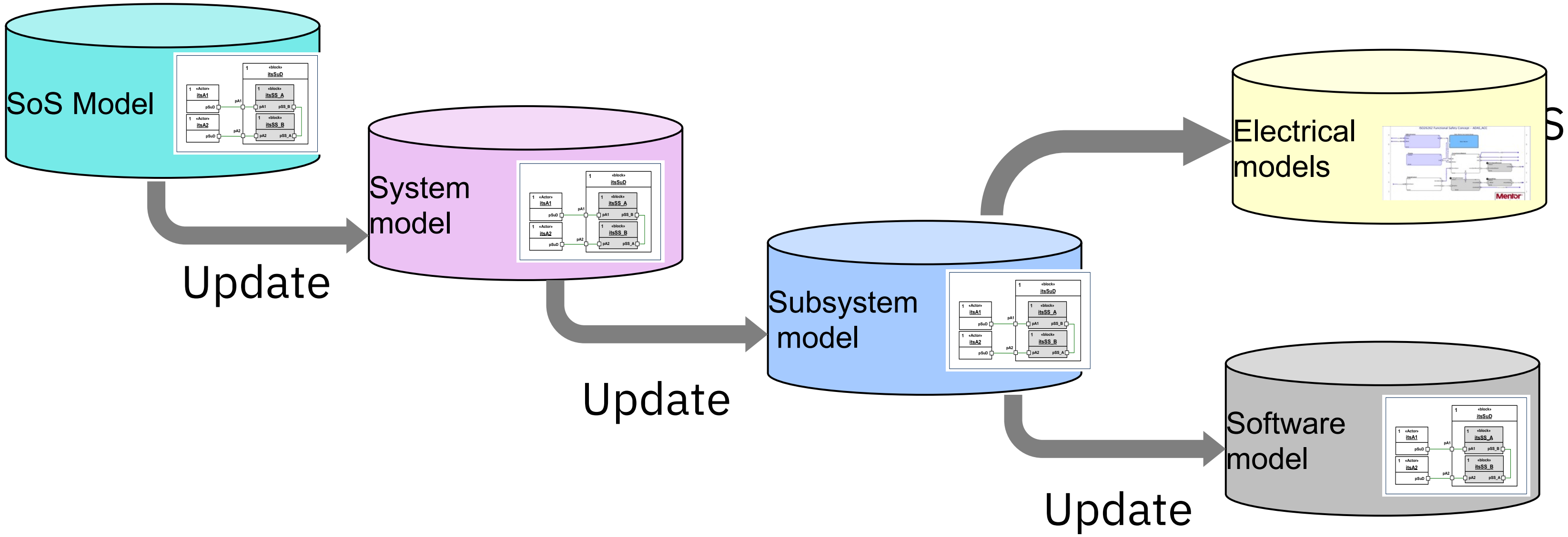
Data exchange to enable process continuity

Data exchange

- Why data exchange?
  - Propagate information across domains and levels of abstraction
  - Parametric and structural information in the form of a model transformation
  - Automation of change propagation across linked artifacts
- OSLC establishes standard **domain vocabularies** as common artifact representations that enables automation of data exchange

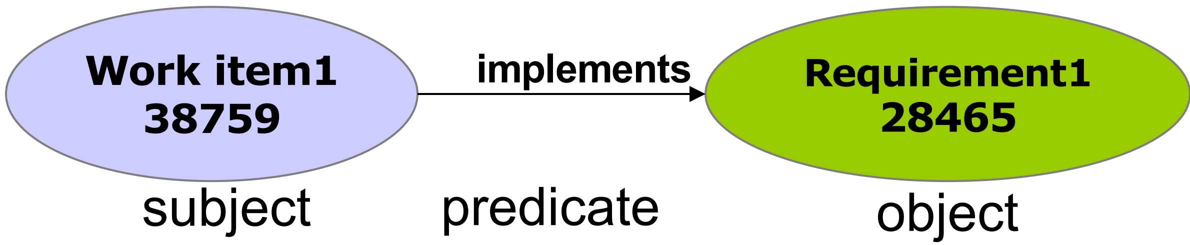


SLC domain vocabularies



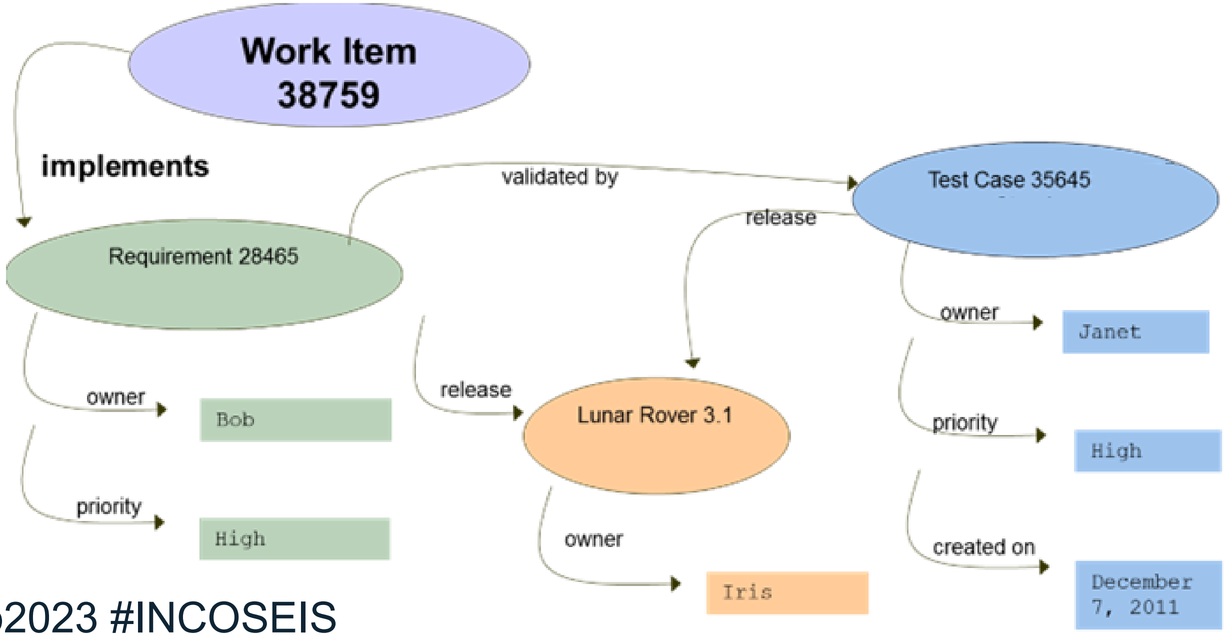
Resource representation with the RDF data model

- RDF – Resource Definition Framework
  - a standard to describe structured data on the web.
  - Generic description of linked data as a set of **triples**
  - RDF triples inspire a **graph**
- Basic structure of information: a triple
  - consists of a subject, a predicate and an object.
  - The predicate denotes a relationship between the subject and object.
  - Graph nodes are resources or literals (values)
- RDF predicates are defined in RDF vocabularies identified by namespaces
  - e.g. rdf:about

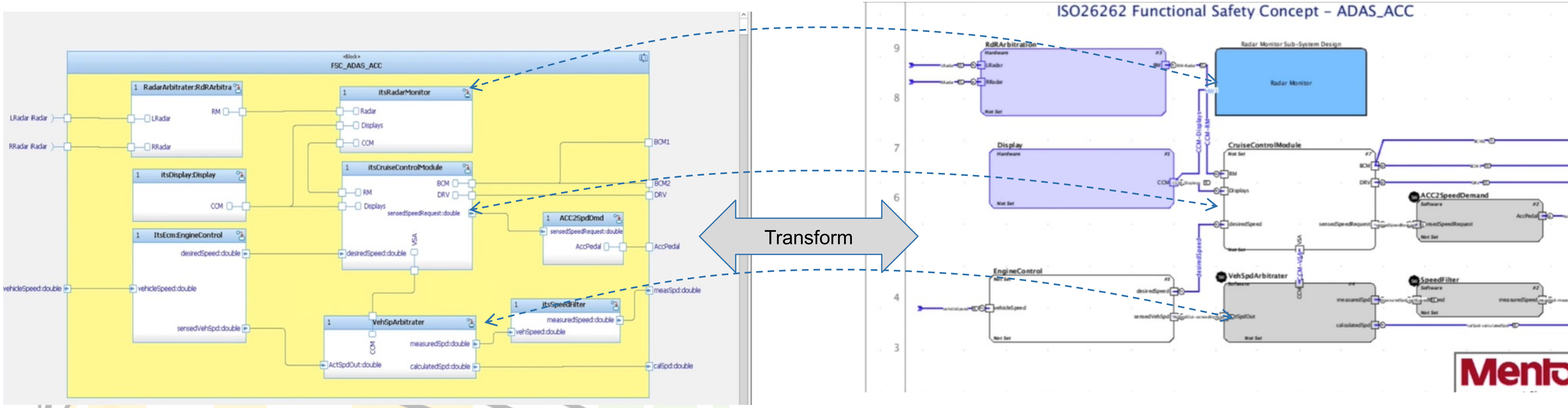


Serialization in turtle format

<http://example.com/Workitems/1> a oslc_cm:Workitem ;
oslc_cm:implements <http://example.com/Requirements/1>.



Example: Transforming a system model to electrical model



OSLC configuration management:
Standardizes access of versioned data across applications

Global configuration management

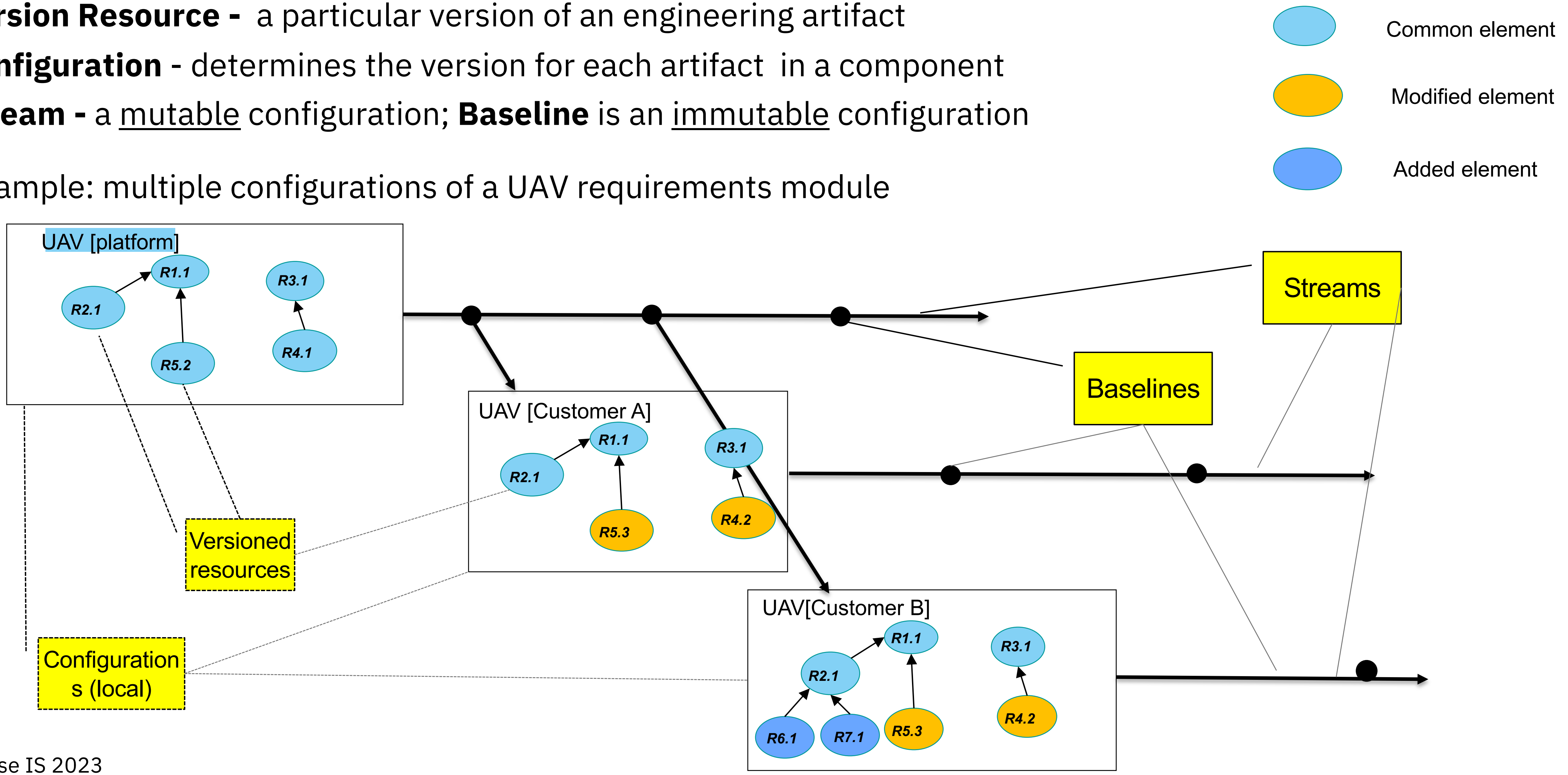
Component - a configuration item; a container of resources

Version Resource - a particular version of an engineering artifact

Configuration - determines the version for each artifact in a component

Stream - a mutable configuration; **Baseline** is an immutable configuration

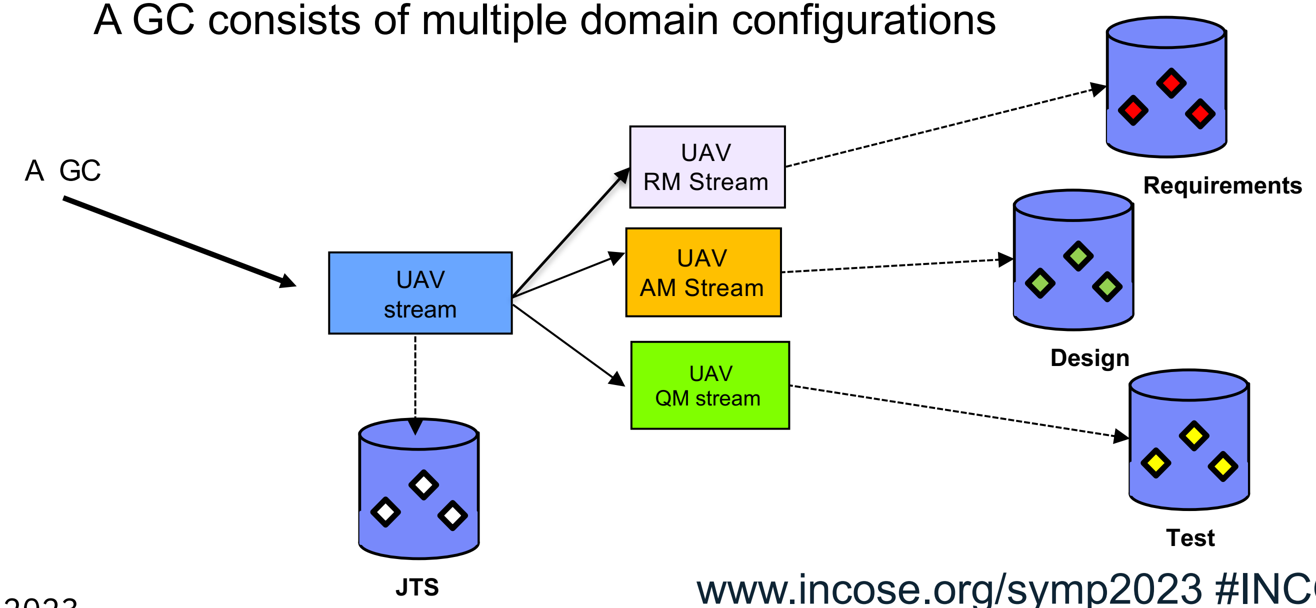
Example: multiple configurations of a UAV requirements module



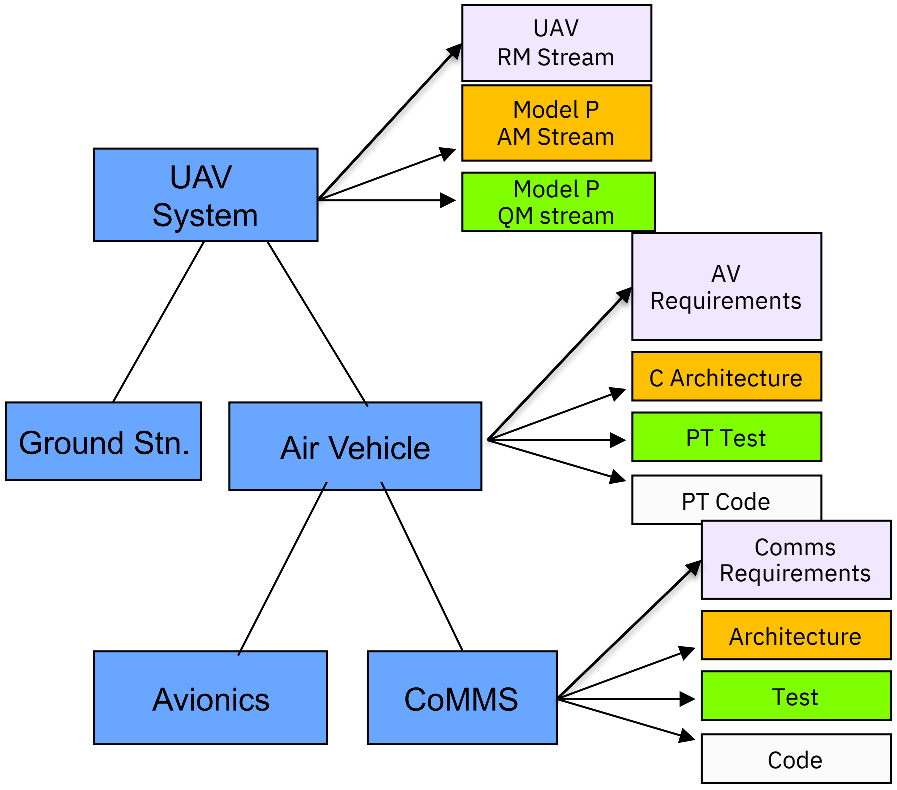
Global configurations: managing configurations across the digital thread

- Global configuration **(GC)** orchestrates configurations of **multiple** components across applications
- GCs are **hierarchical** and can represent the logical structure of lifecycle data
- GCs can also be streams or baselines
- GCs are managed by a global configuration service

A GC consists of multiple domain configurations

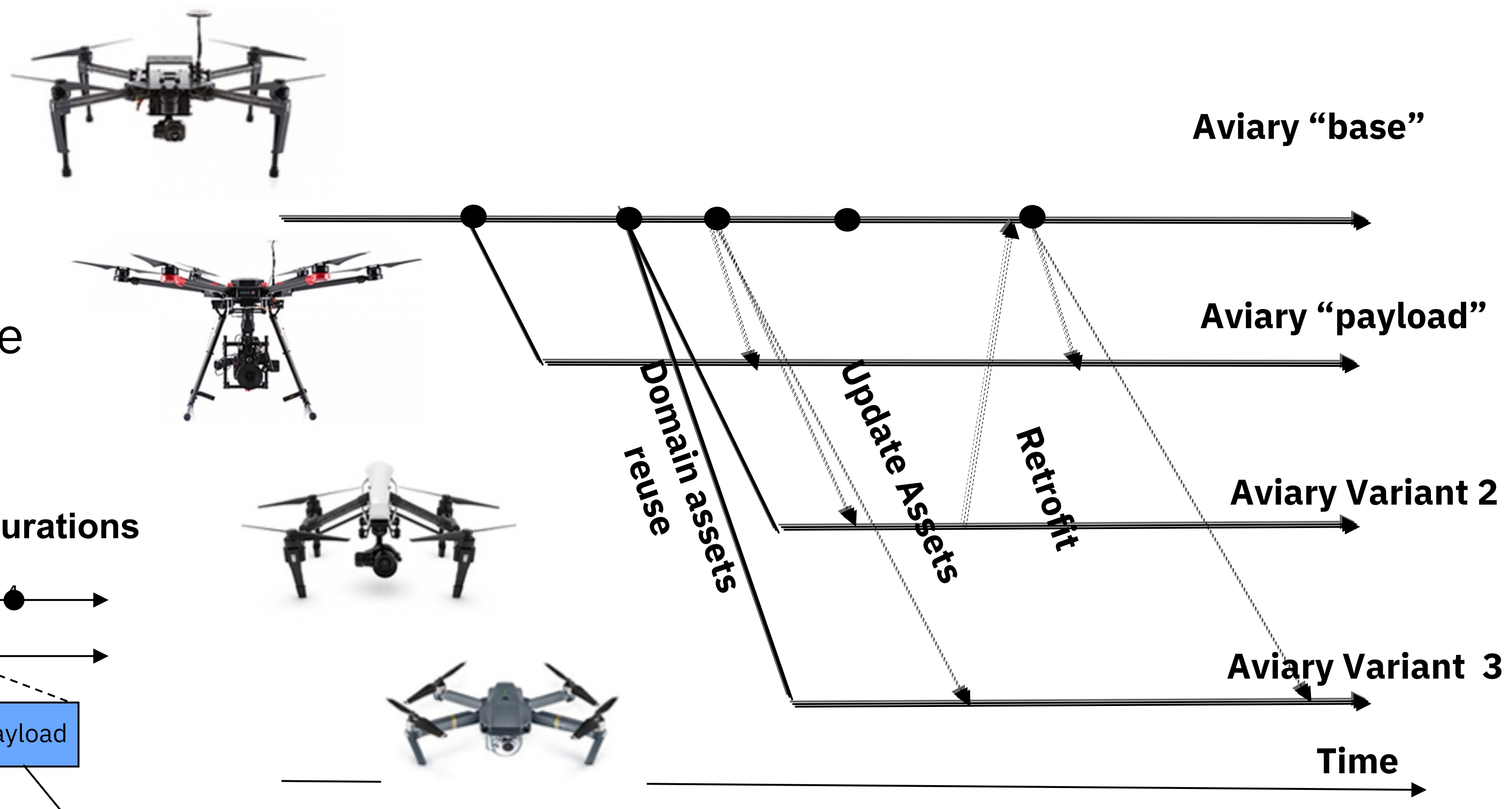


Example: A hierarchical GC of a UAV

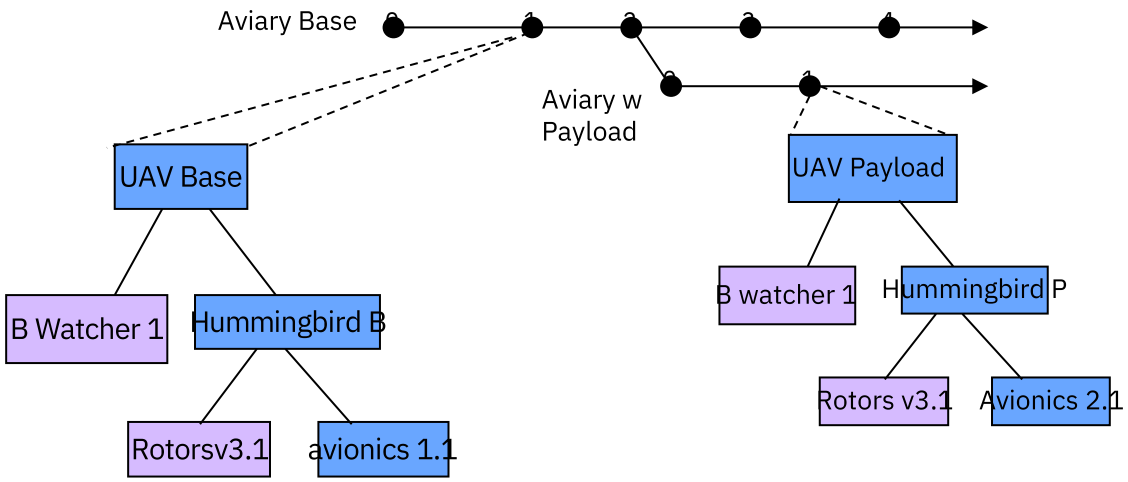


Using GCs to manage system variants

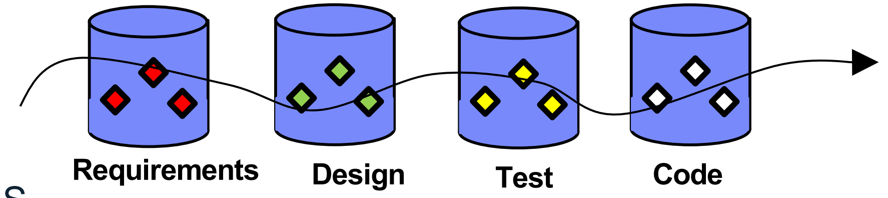
- A GC stream may represent an evolution of a system variant
- GCs enable component reuse across variants
- Changes to a variant can propagate to other variants



Variants Reuse across hierarchical configurations



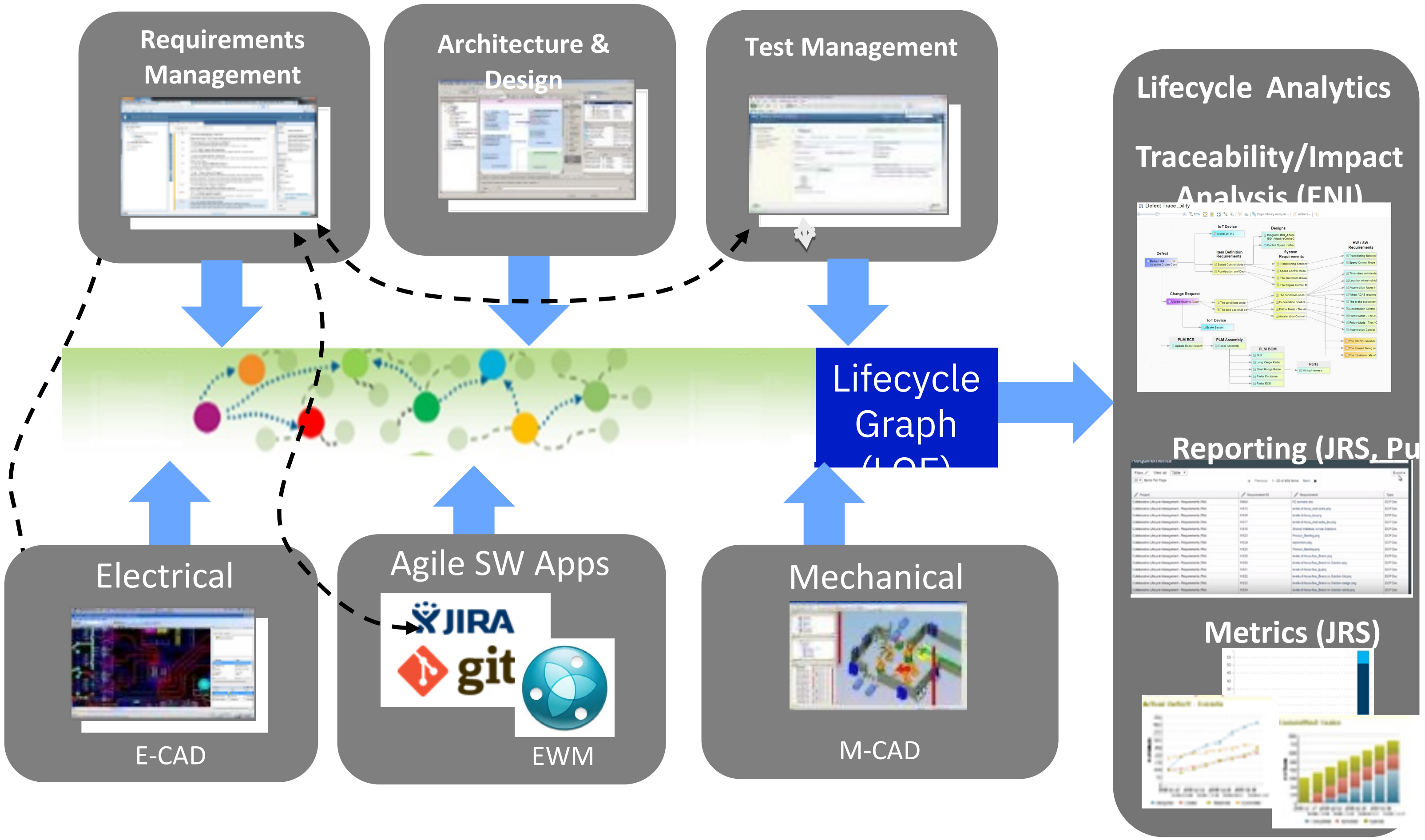
Example: Aviary Variants



Central Reporting and Analysis

Analytics & Reporting

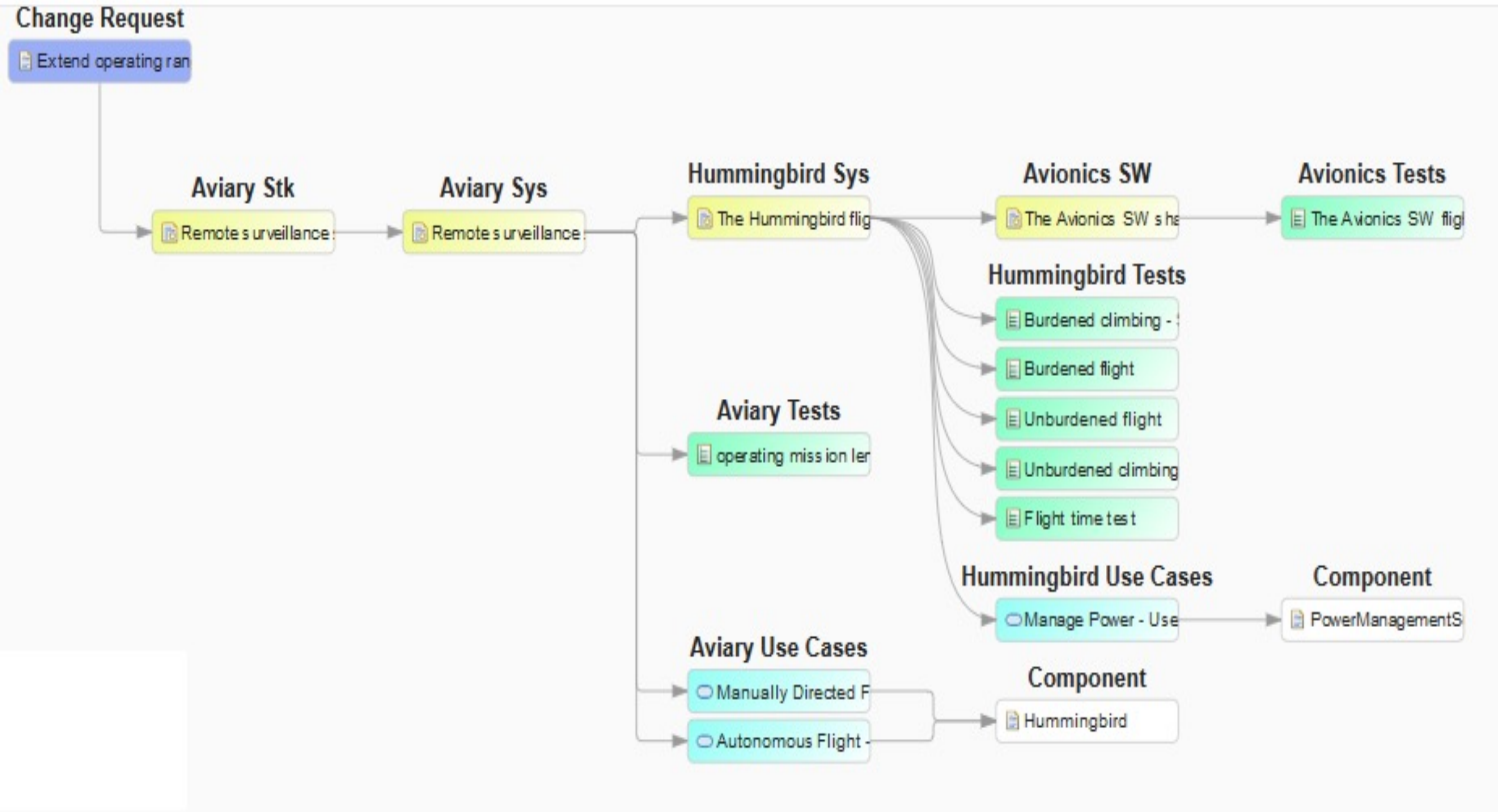
- Central lifecycle graphs enable digital viewpoints across the digital thread
- Such digital viewpoints enable
  - decision making such as change impact analysis
  - necessary KPIs for continuous process improvement
  - provide necessary evidence for regulatory compliance
- Lifecycle graph are continuously maintained using the OSLC data tracking (TRS) service



Impact analysis based on lifecycle graphs visualization

Example: Engineering Insights (ENI)

- ENI is an application renders Graph like viewpoints to study the relationships across lifecycle data
- ENI viewpoints are based on view templates that specify the content of the view
- Conceptually view template represent queries of the lifecycle graph and rendering details
- Impact analysis visualizes the downstream dependencies of an origin node

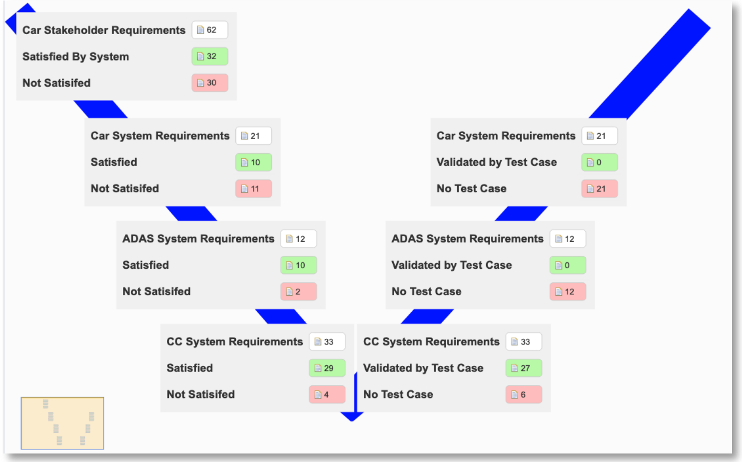


Impact of extending the range requirements for “Aviary”

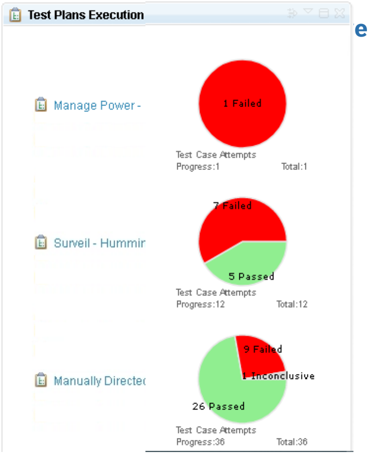
Example: Evidence Reports and KPIs based on lifecycle graph

- All necessary evidence is produced from actual lifecycle records
- All necessary reports automated by predefined templates
- No need for special certification record tracking work!

Lifecycle traceability coverage metrics



Subsystems verification summary



e it!

Aviary System requirements to test traceability report

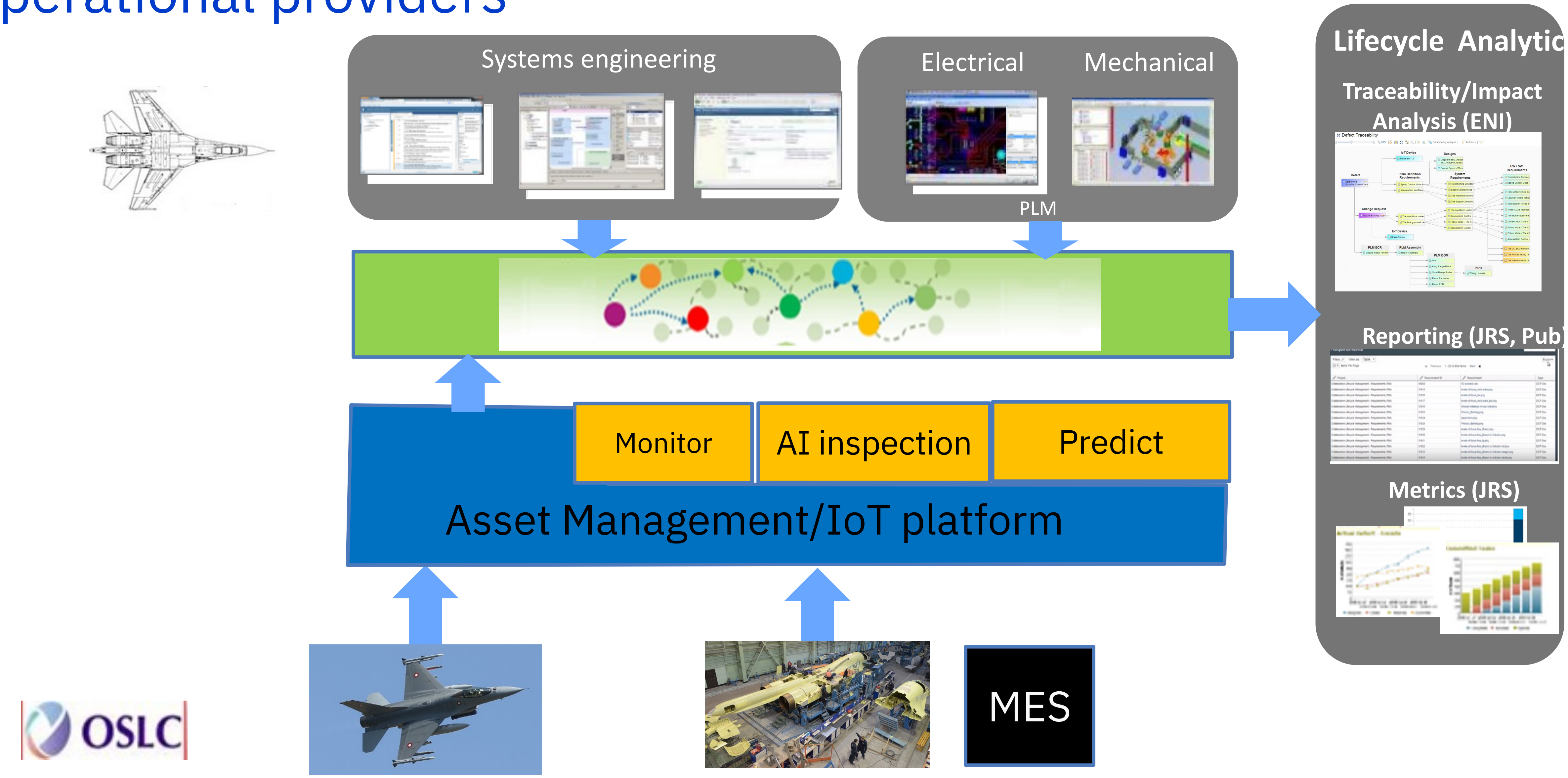
Req Id	Requirement	Tst Id	Test Case
4185	The Hummingbird shall be able to rotate independently of its direction of movement, either to the left or right to any number of degrees.	93	Test Rotation In Various Directions Of Movement
4169	The Hummingbird shall report its altitude above any surface immediately below it in meters with a range of 0 – 1000m and an accuracy of ±2 cm or 1% of the measured height, whichever is greater.	90	Test That Hummingbird Reports Its Altitude Above Any Surface In Meters
4215	The Hummingbird maximum flight distance shall be at least 40 miles.	69	Stress Flight Test
4171	The Hummingbird shall be able to maintain attitude within 5 degrees of arc for roll, pitch, and yaw in the presence of steady winds of up to 20 mph or 20 degrees in the presence of irregular winds of up to 30 mph (see Figure 1).	89	Test That Hummingbird Maintains Roll, Pitch, And Yaw
4177	The Hummingbird shall report its location to the Pilot Controller in response to a command with an accuracy of ±1 meter.	87	Test That Hummingbird Reports Its Location To The Pilot Controller In Response To A Command
4193	The Hummingbird shall be able to move in any combination of directions: up/down, right/left, forward/backward.	86	Test That The Hummingbird Can Move In Any Combination Of Directions: Up/Down, Right/Left, Forward/Backward
4217	The Hummingbird flight time shall be at least 2 hours.	69	Stress Flight Test
4217	The Hummingbird flight time shall be at least 2 hours.	94	Flight Time Test
4184	The hummingbird camera focus shall be settable from 10m to infinity.	95	Camera Focus Test
4176	The aircraft shall support wireless communication using a custom protocol between it and the pilot control and between it and up to 4 separate Viewers.	92	Test Communication With The Pilot Control And The Viewers
4186	The Hummingbird camera zoom shall be commandable via Pilot commands from -4x to + 10x with fidelity	97	Camera Zoom Test

Aviary System verification report

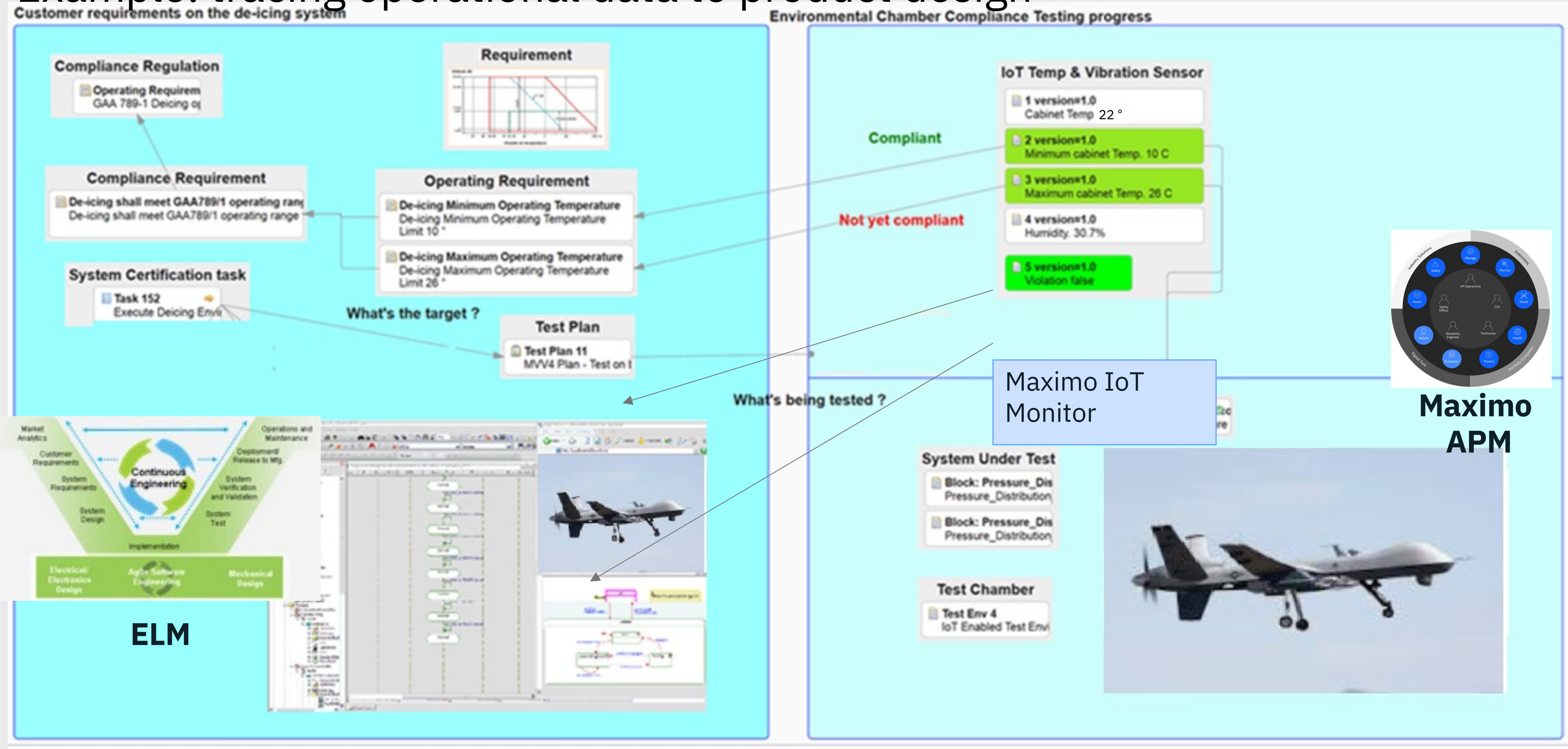
Test Id	Test Case	Verdict
86	Test That The Hummingbird Can Move In Any Combination Of Directions: Up/Down, Right/Left, Forward/Backward	Failed
86	Test That The Hummingbird Can Move In Any Combination Of Directions: Up/Down, Right/Left, Forward/Backward	Passed
86	Test That The Hummingbird Can Move In Any Combination Of Directions: Up/Down, Right/Left, Forward/Backward	Inconclusive
87	Test That Hummingbird Reports Its Location To The Pilot Controller In Response To A Command	Failed
87	Test That Hummingbird Reports Its Location To The Pilot Controller In Response To A Command	Passed
88	Test That Hummingbird Flies At Speed Of 40 Mph	Failed
88	Test That Hummingbird Flies At Speed Of 40 Mph	Passed
89	Test That Hummingbird Maintains Roll, Pitch, And Yaw	Failed
89	Test That Hummingbird Maintains Roll, Pitch, And Yaw	Passed
90	Test That Hummingbird Reports Its Altitude Above Any Surface In Meters	Failed
90	Test That Hummingbird Reports Its Altitude Above Any Surface In Meters	Passed
91	Test That Hummingbird Reports To The Pilot A Loss Or Significant Degradation Of Rotor Function	Passed
92	Test Communication With The Pilot Control And The Viewers	Passed
93	Test Rotation In Various Directions Of Movement	Failed
93	Test Rotation In Various Directions Of Movement	Passed
94	Flight Time Test	Failed

+Using ELM reporting

Extending the span from engineering to operations with operational providers



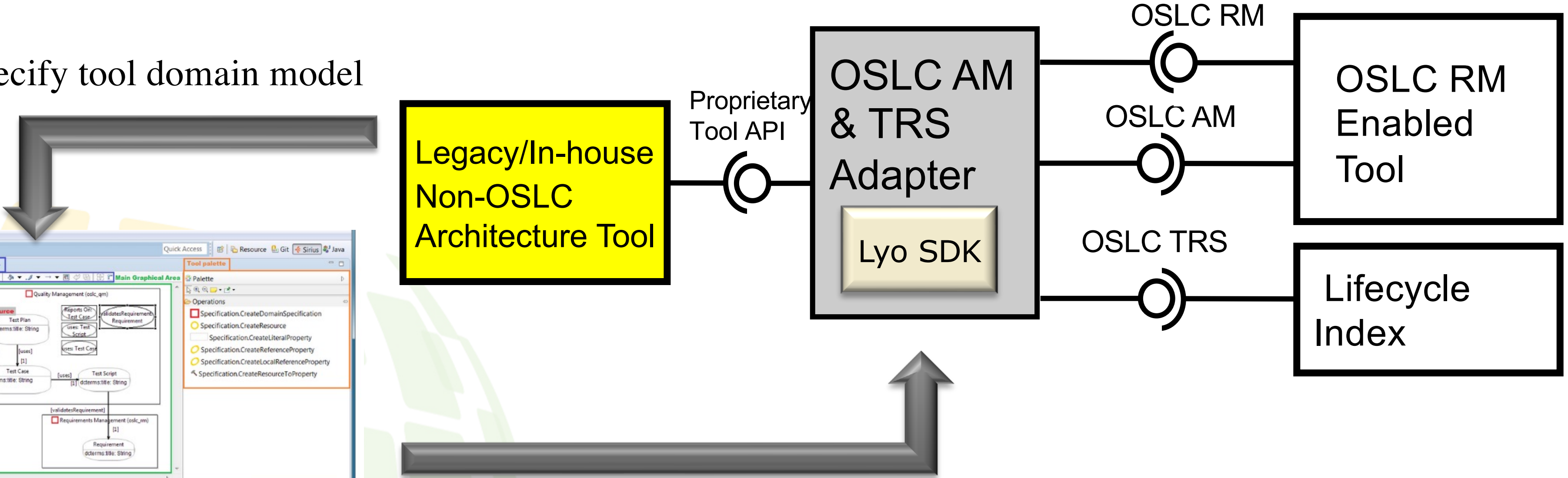
Example: tracing operational data to product design

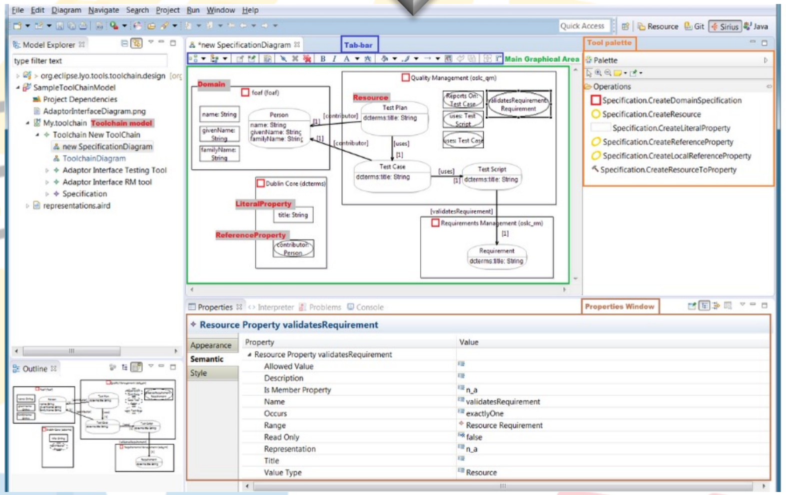


Eclipse Lyo – automating creation of OSLC APIs

- Open source system to develop OSLC servers
- Streamlines creation of OSLC adapters to existing tools
- Includes an IDE (Lyo designer) and an SDK

Specify tool domain model



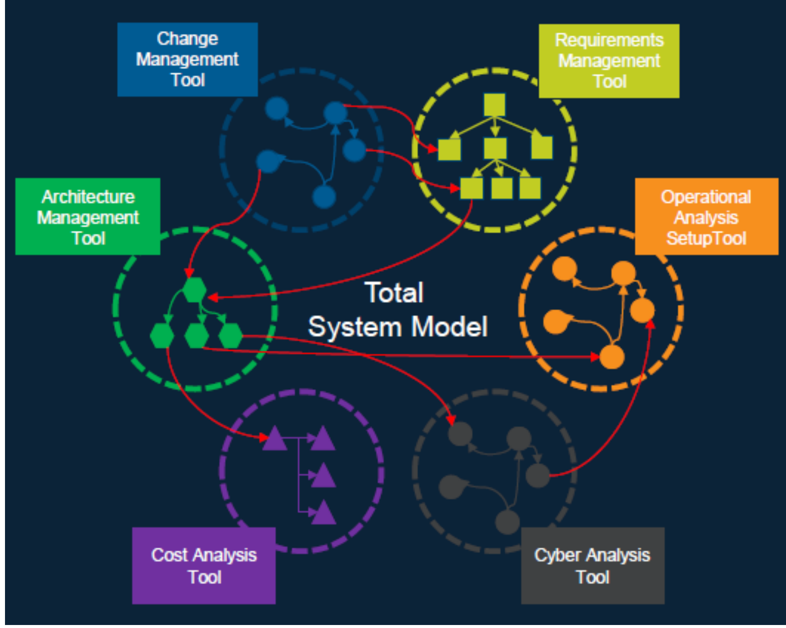


Lyo Designer

A typical flow of an OSLC adapter creation

http://eclipse.org/lyo

Summary: Digital threads benefits and best practices



- Digital threads provide engineers with the data, insights, and collaboration needed to optimize a product's environmental impact across its entire lifecycle.
- OSLC is an open standards-based architecture that enables key engineering capabilities:
  - Traceability throughout the product development process
  - Data-driven decision-making: provide engineers with access to real-time data throughout the development process
  - collaboration and transparency across different teams and stakeholders, allowing for more efficient and effective decision-making around sustainability
  - continuous improvement - allowing engineers to identify areas for improvement incorporating operational data
  - Foster reuse through global configuration management

Where to find out more….

- Open-services.net
- Eclipse LYO – https://www.eclipse.org/lyo
- Jazz.net - https://jazz.net
- W3C Linked data -
- https://www.w3[illegible]ds/seman[illegible]

Thank You!

33rd Annual INCOSE
international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023