32nd Annual INCOSE international symposium

hybrid event

Detroit, MI, USA
June 25 - 30, 2022

Eran Gery – IBM Engineering

Realizing Viewpoints in Digital engineering

www.incose.org/symp2022

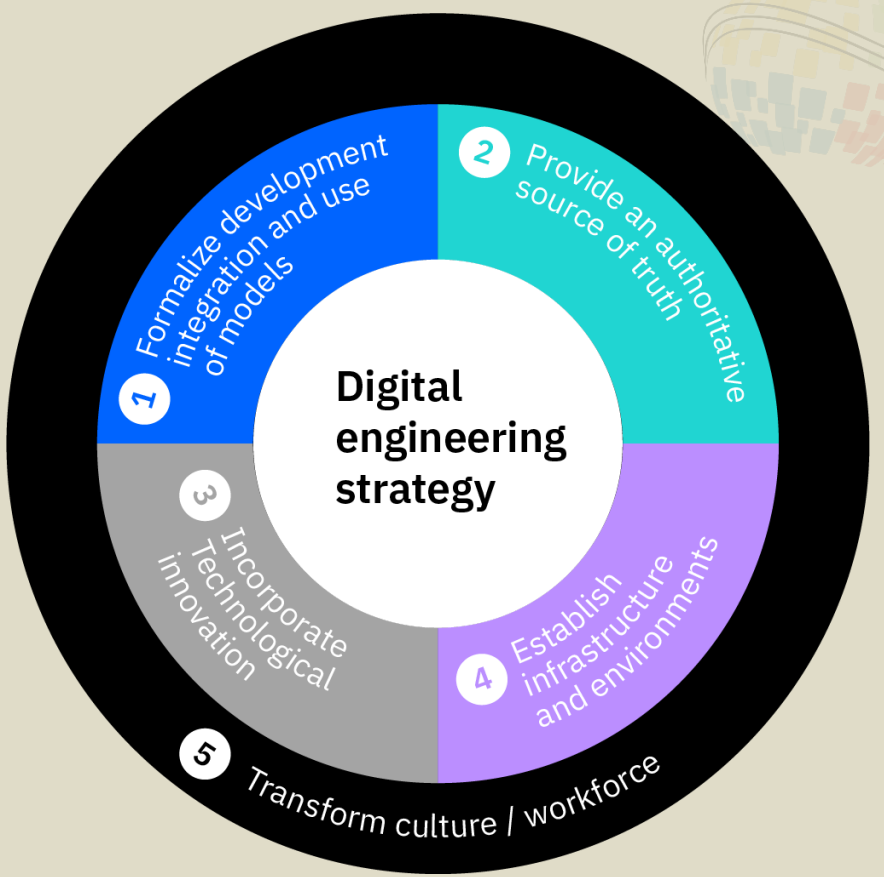
Outline

- Why Digital Engineering viewpoints
- The INCOSE DEIXWG DVM reference model
- OSLC as a framework for a digital back bone
- Specifying viewpoints over the lifecycle graph
  - Lifecycle graph and OSLC TRS
  - The jazz reporting service
  - Engineering insights
- Summary and additional resources

Digital Engineering

an integrated digital approach that uses authoritative sources of system data and models as a continuum across disciplines to support lifecycle activities from concept through disposal.

- Transforming document centric processes
- Eliminating “air gaps”
- Key principles
  - Facilitate digital continuity across providers to form lifecycle information models via digital threads
  - Enable data exchange across domains and providers to foster data consistency and automation
  - Ensure data consistency validity by managing “trusted” data sources
  - Enable cross lifecycle digital viewpoints to support the necessary insights from the data



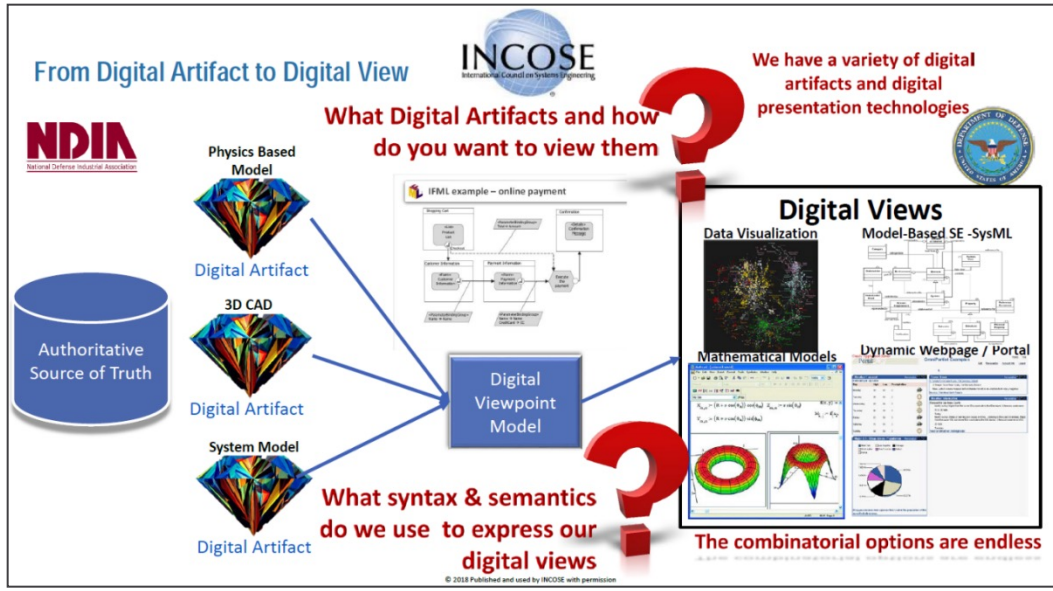
“…such engineering environments will allow DoD and industry partners to evolve designs at conceptual phase, reducing the need for expensive mockups, premature design lock, and physical testing.” [1]

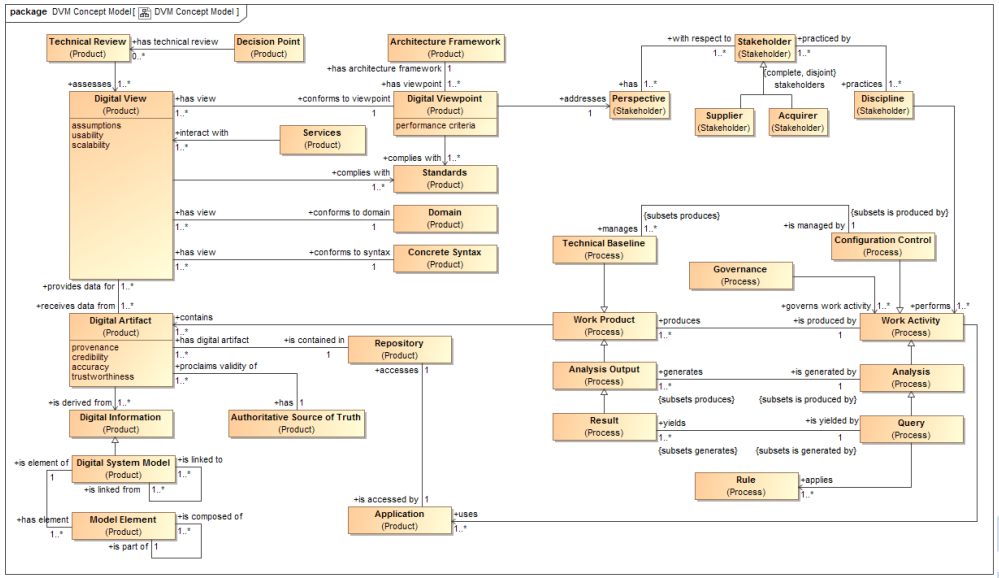
1. DoD Digital Engineering Initiative: https://www.acq.osd.mil/se/initiatives/init_de.html

INCOSE Digital Engineering Information exchange WG (DEIXWG)

Define a conceptual framework for digital engineering

- SE use cases: what information is needed to support certain SE decisions and processes?
- A conceptual Digital Viewpoint Model (DVM) to model the concepts defining both Digital Artifacts and Digital Views and how they are related via Digital Viewpoints
- What standards can be leveraged to enable the necessary viewpoints





Some example challenges for digital viewpoints

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?

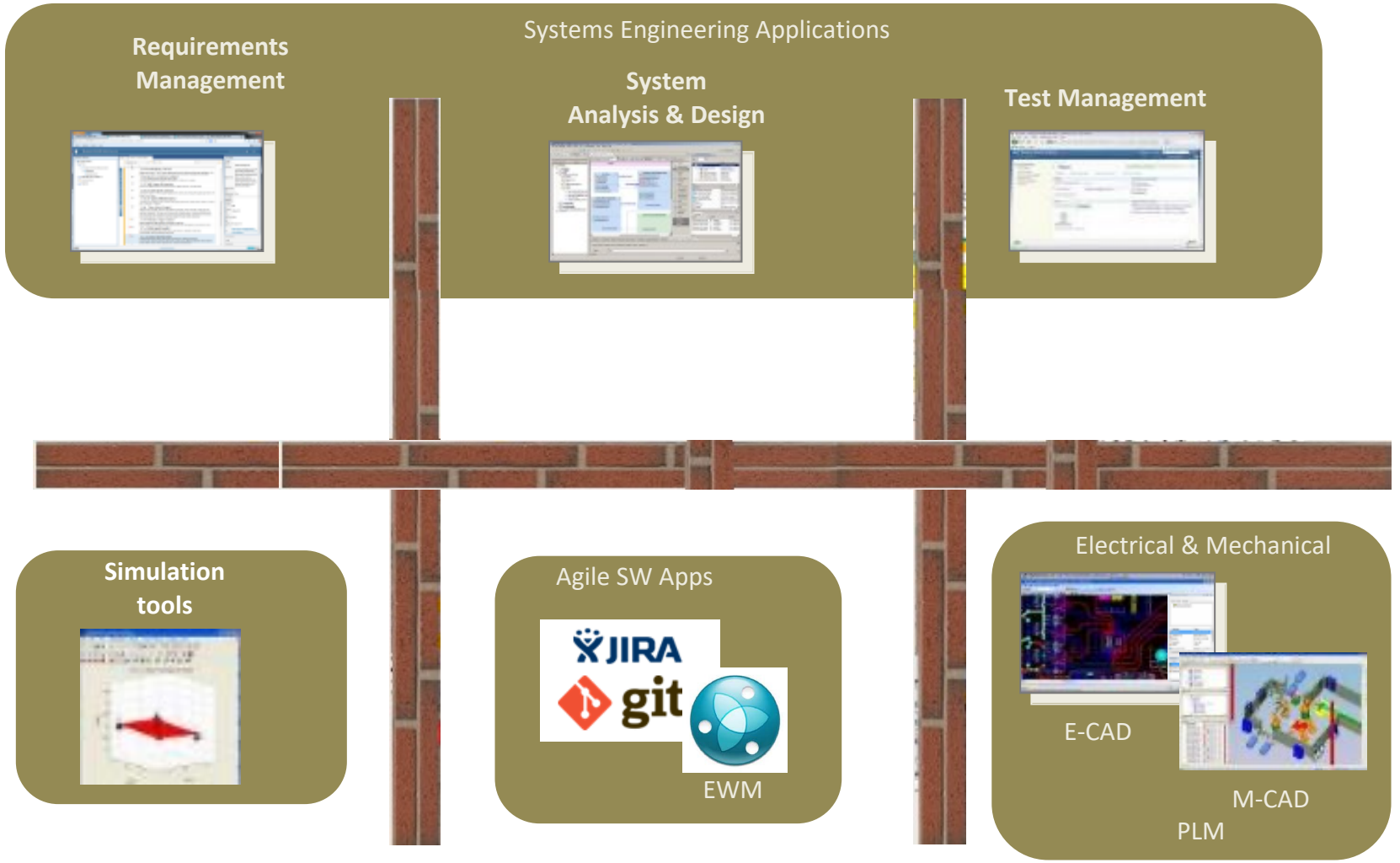
Which variants use this SW component?

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

The siloed nature of lifecycle artifacts…

Digital domain tools exist but siloed,

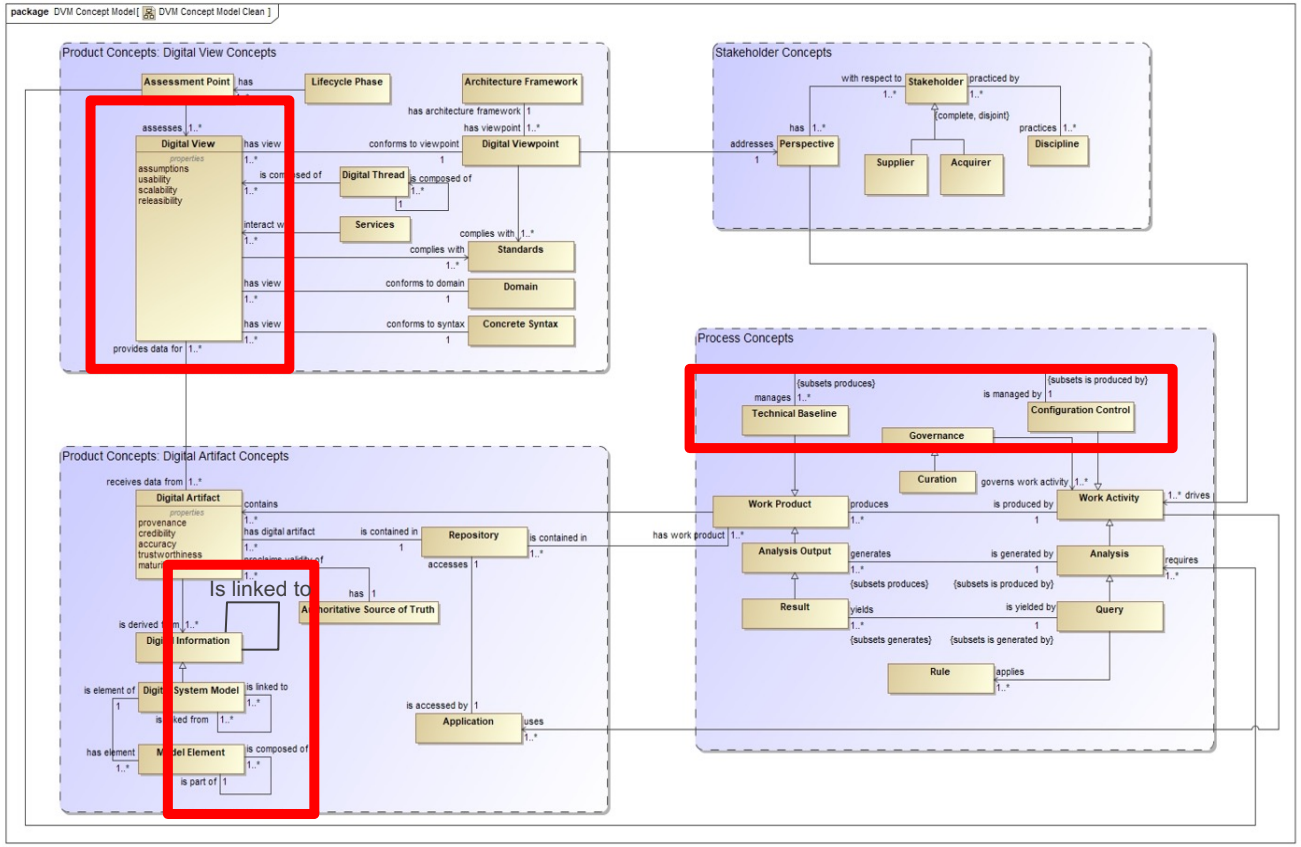
- Key challenges
  - How to maintain consistency across siloed models
  - How to establish traceability (linking) across artifacts
  - How to produce necessary digital viewpoints to support effective SE processes
  - How to maintain authoritative source of truth in such a siloed environment



INCOSE DEIXWG DVM Conceptual data-model

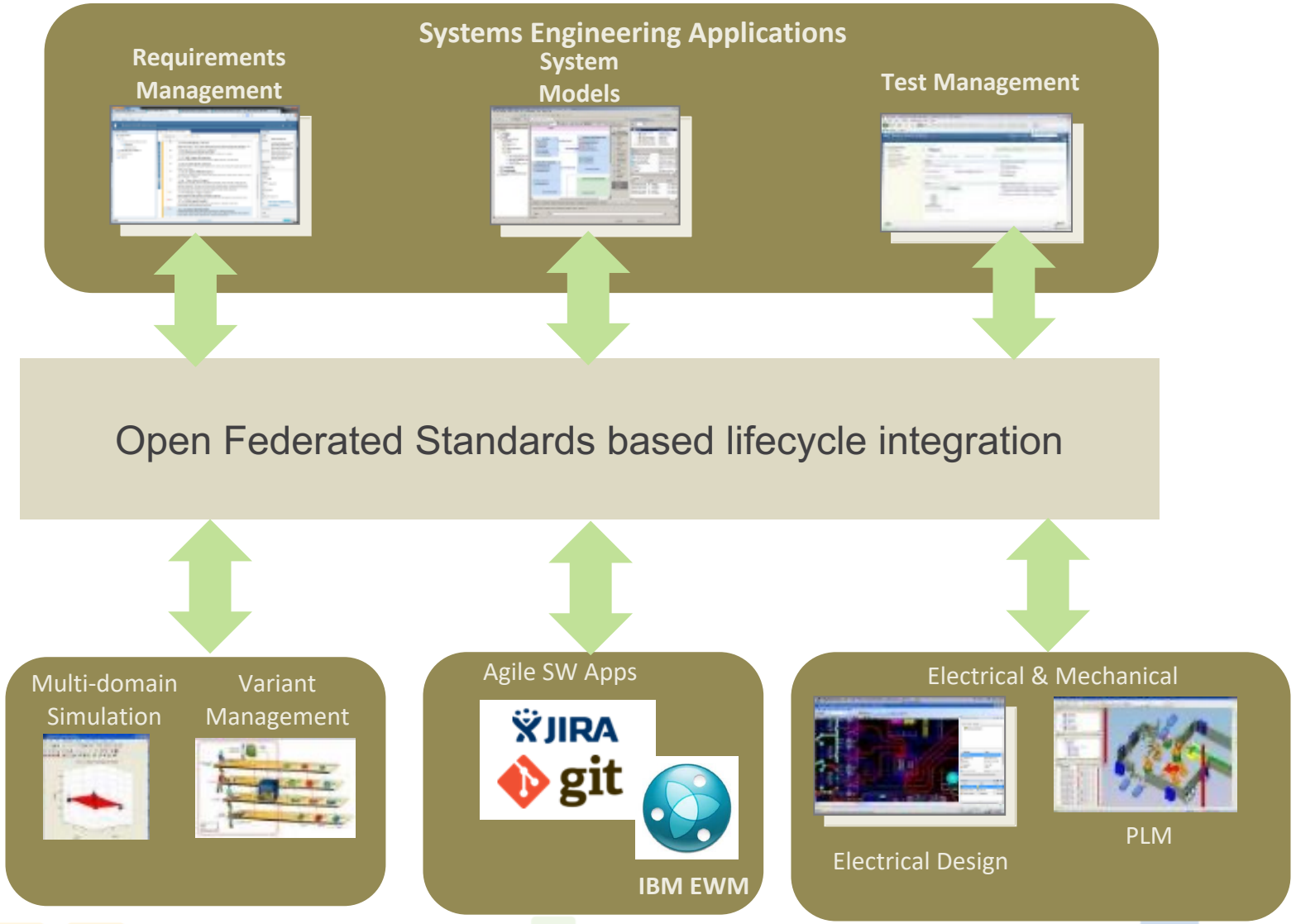
- Challenges..
  - Digital Continuity: How to realize links across models and artifacts
  - Baselining: How to establish “technical baselines” and configuration control across “models” and digital artifacts
  - Lifecycle Views: How to specify a digital view from a collection of artifacts in a “digital” manner?

Requires a digital engineering backbone!



Approaches for digital engineering backbone

- Centralized (“PLM approach”)
  - Centralized repository that manages all the data
  - Often challenges to authoritative source concept, domain tools isolated
  - Proprietary
- MBSE backbone
  - Import various domain data to MBSE – act as centralized repository
  - No ASOT data management
  - Does no scale
- Link management applications
  - Centralized link repository across ASOTs
  - Limited ability to support a DVM
- OSLC (Linked data Federated)
  - Supports ASOTs and DVM



OSLC to address these digitization challenges

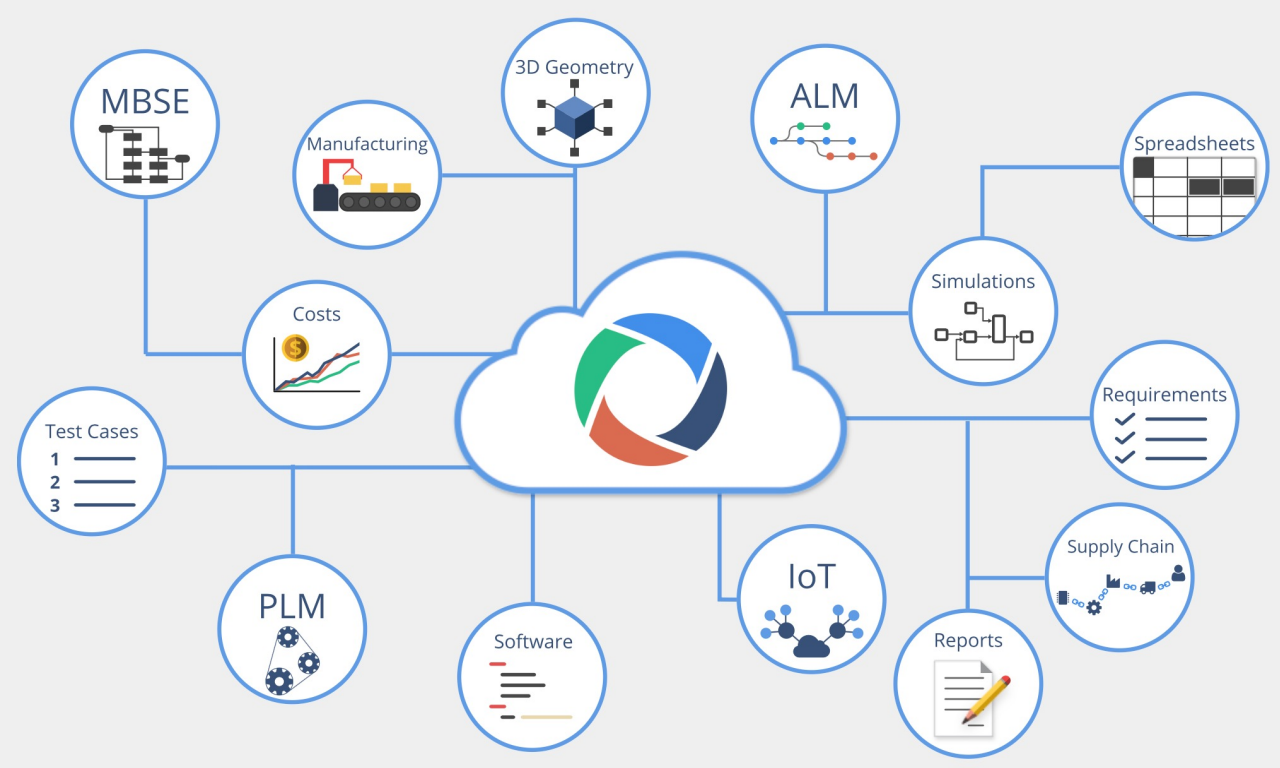
A service-oriented lifecycle integration approach based on W3C semantic web stack

Modern HTTP/REST integration architecture

W3C Linked data architecture

Data and UI/workflow integration

Standard Domain vocabularies



Modular Systems Approach

Complete lifecycle traceability

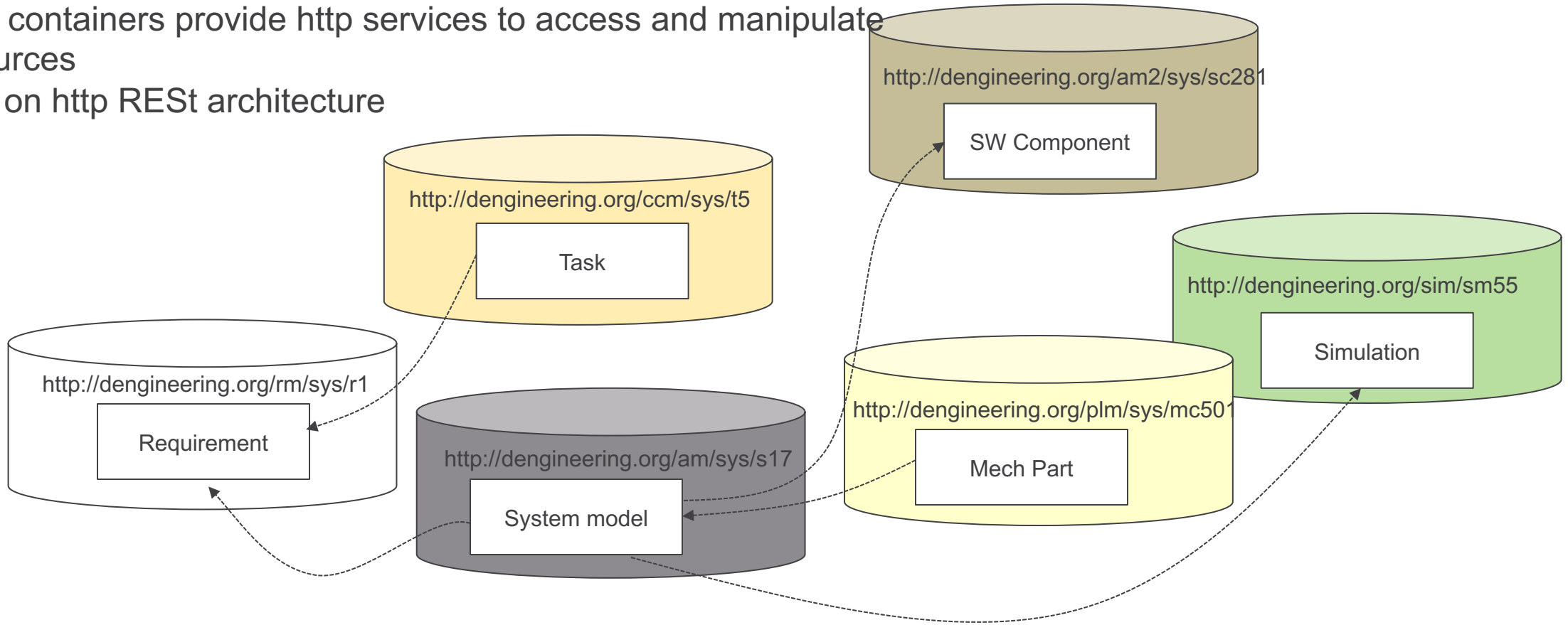
Open Standard and Open Source community

Reference implementation: Eclipse Lyo

OSLC is an open and scalable approach to lifecycle integration. It simplifies key integration scenarios across heterogeneous tools. Does not replace other data interchange standards.

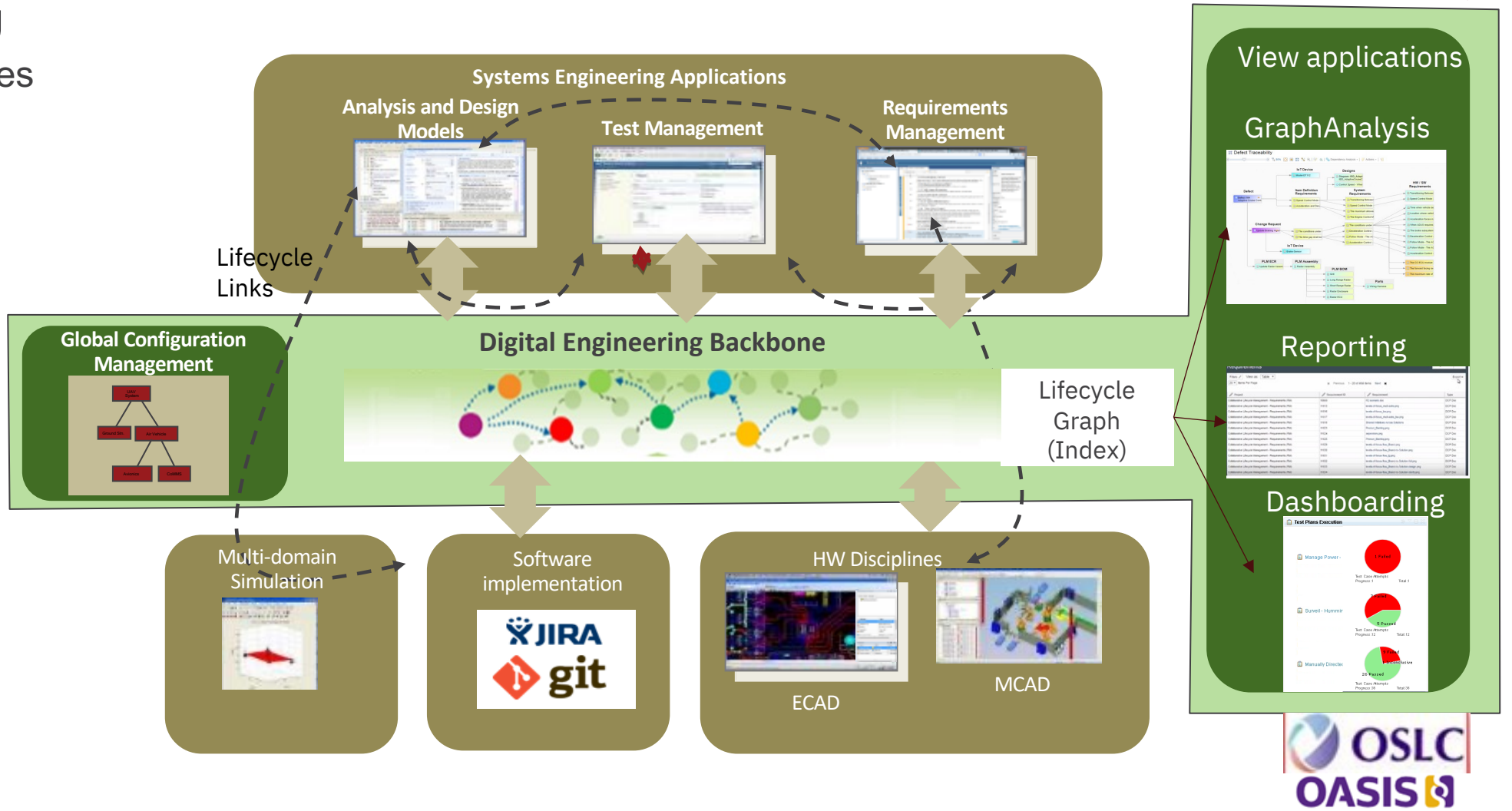
The foundation: Linked data (w3c)

- Generalizing the idea of linked web documents for data
- Lifecycle objects (resources) are identified by http URLS
- Data containers provide http services to access and manipulate resources
- Best on http RESt architecture



A digital backbone architecture based on OSLC

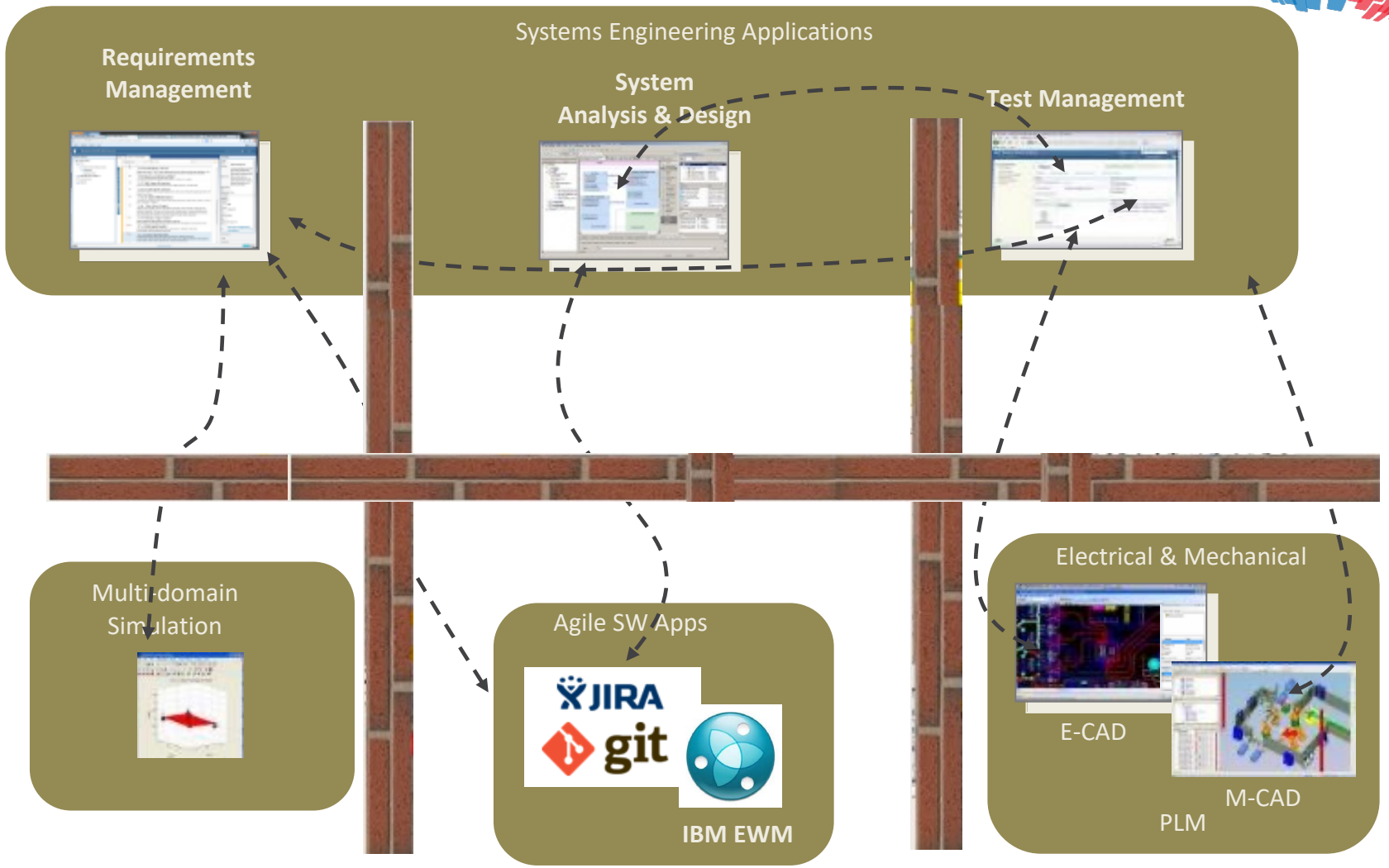
- Collaborative linking
  - OSLC linking services
- Global configuration service
  - OSLC Global configurations
- Lifecycle Graph
  - OSLC TRS
- View applications



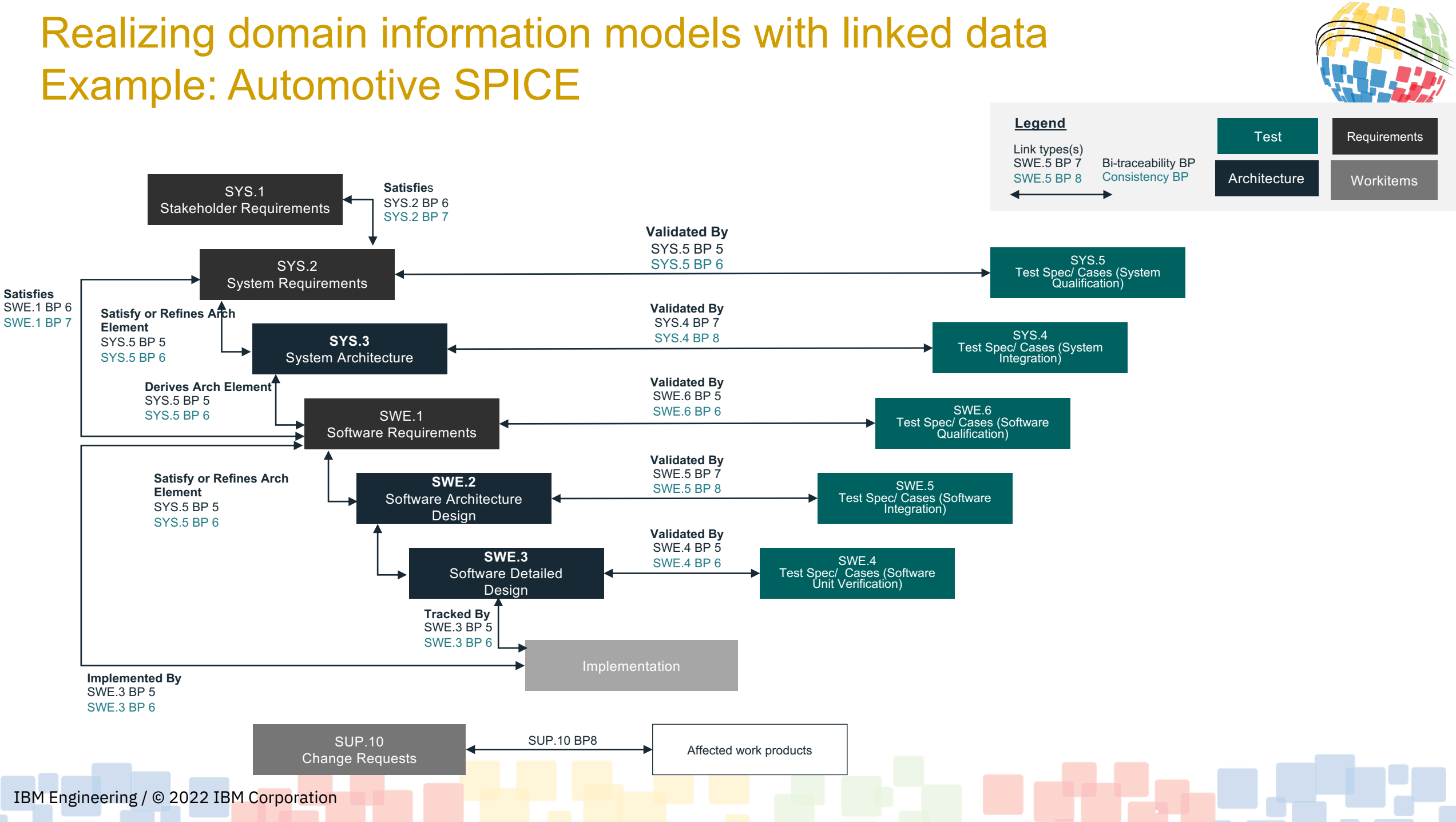
Collaborative linking across domain applications

Digital continuity

- Domain tools (ASOTs) extended with OSLC service (adapter)
- Leverage OSLC linking services to establish links
  - Selection service via delegated HTML
  - Storage (HTTP put)
- The linking is a federated concept based on collaboration across tools

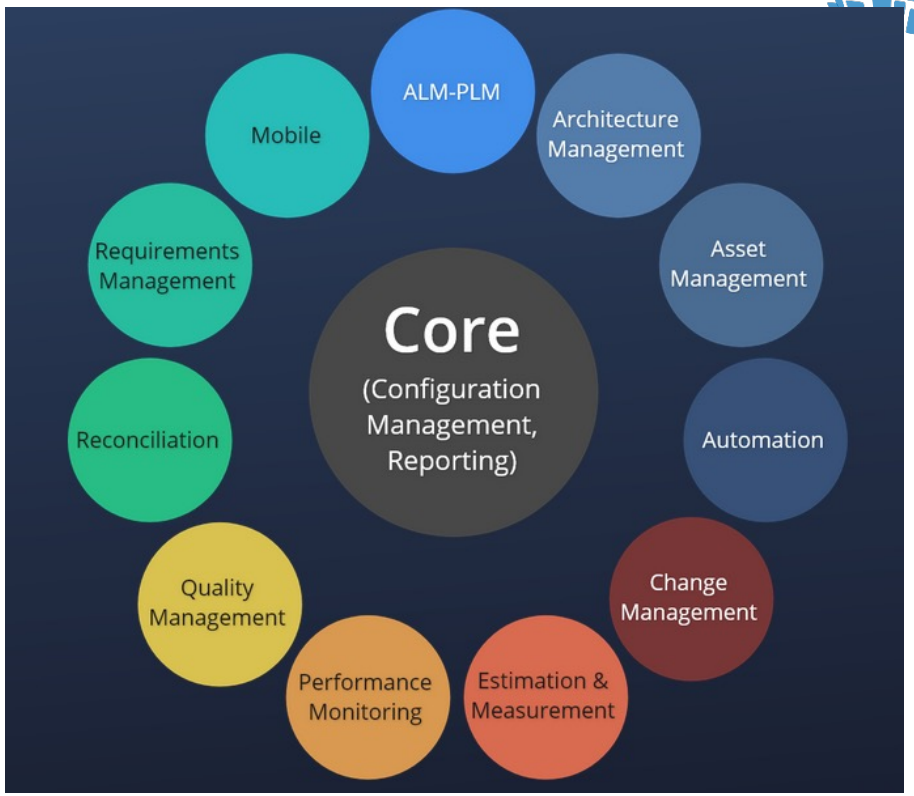


Realizing domain information models with linked data
Example: Automotive SPICE



Common artifacts representation using OSLC vocabularies

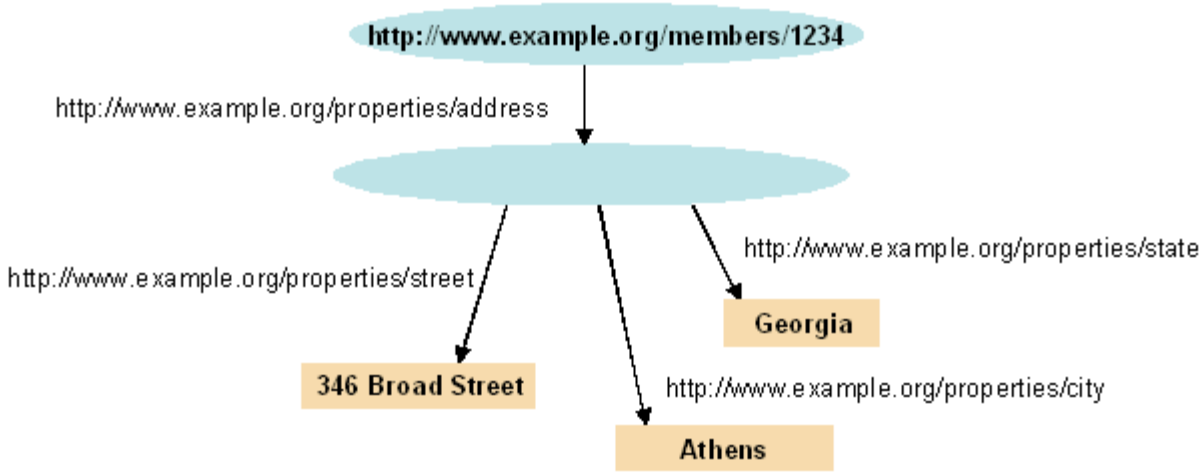
- Establish standard domain vocabularies as common representations across different tools based on a common metadata
- Enables a common representation of data for digital automation
- Enables data exchange across domain applications to maintain consistency
- Transform data from one domain to another based on standard vocabularies
- Propagate changes across linked artifacts



OSLC domain vocabularies

RDF – Cross lifecycle ontology and data representation

- Part of w3c semantic web stack
- Can describe any set of resources and their inter relationships across all domains
- Does not require a fixed schema
- Can be augmented by schema definitions to check consistency and apply reasoning
  - RDFS



RDF Graph expressed in Turtle notation

```
@prefix eric:    <http://www.w3.org/People/EM/contact#> .
@prefix contact: <http://www.w3.org/2000/10/swap/pim/contact#> .
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

eric:me contact:fullName "Eric Miller" .
eric:me contact:mailbox <mailto:e.miller123(at)example> .
eric:me contact:personalTitle "Dr." .
eric:me rdf:type contact:Person .
```

SPRQL query language

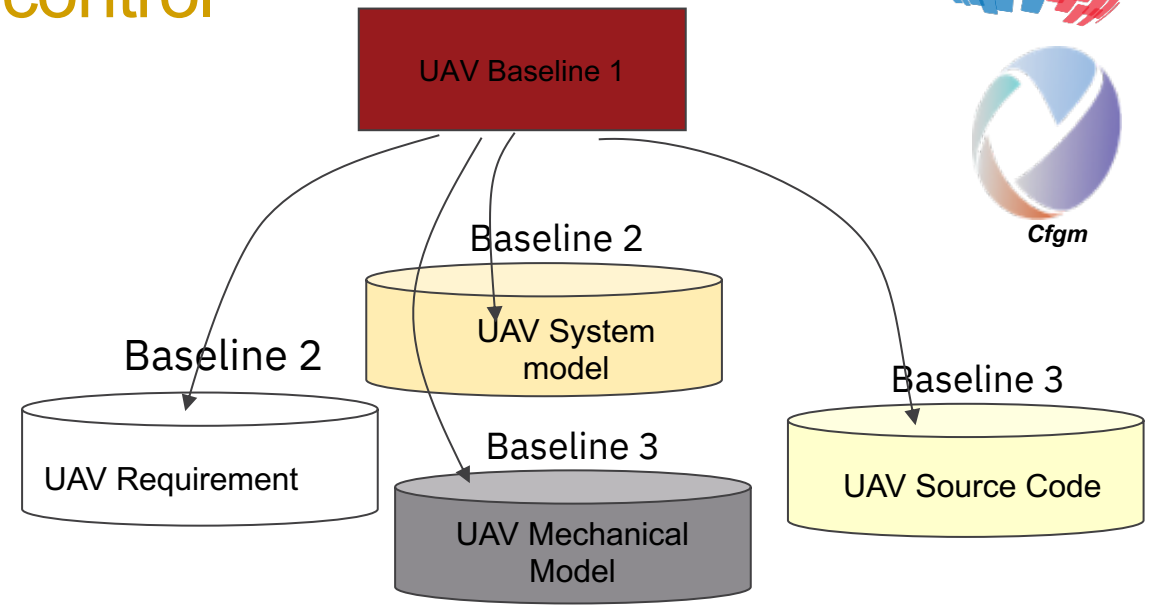
```
PREFIX ex: <http://example.com/exampleOntology#>
SELECT ?capital ?country
WHERE {
  ?x ex:cityname ?capital ;
     ex:isCapitalOf ?y .
  ?y ex:countryname ?country ;
     ex:isInContinent ex:Africa .
}
```

Global configuration management
Cross lifecycle baselines and configuration control

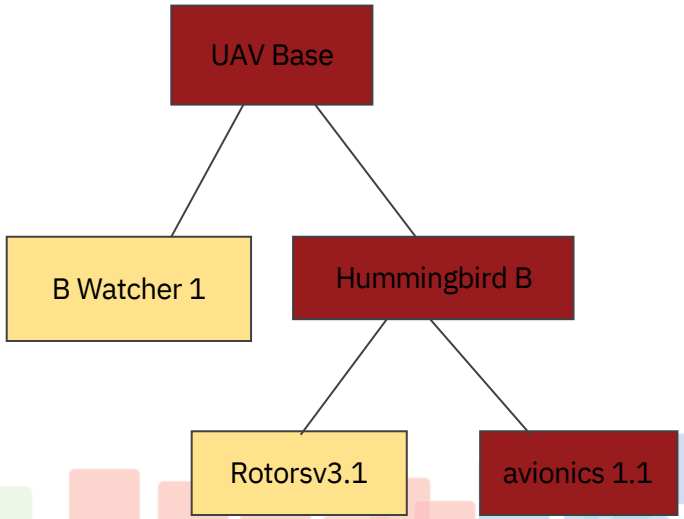
Global baselining

- Realizing “technical baselines” and “configuration control” across digital artifacts as per DVM model
- Establish a global configuration from a set of federated tools and models
- Orchestrating tool specific local configuration controls into a set of consistent configurations
- Global configurations are hierarchical recursive
- Key concepts
  - Global Streams (“branches”)
  - Global baselines

A Global Configuration of UAV assets



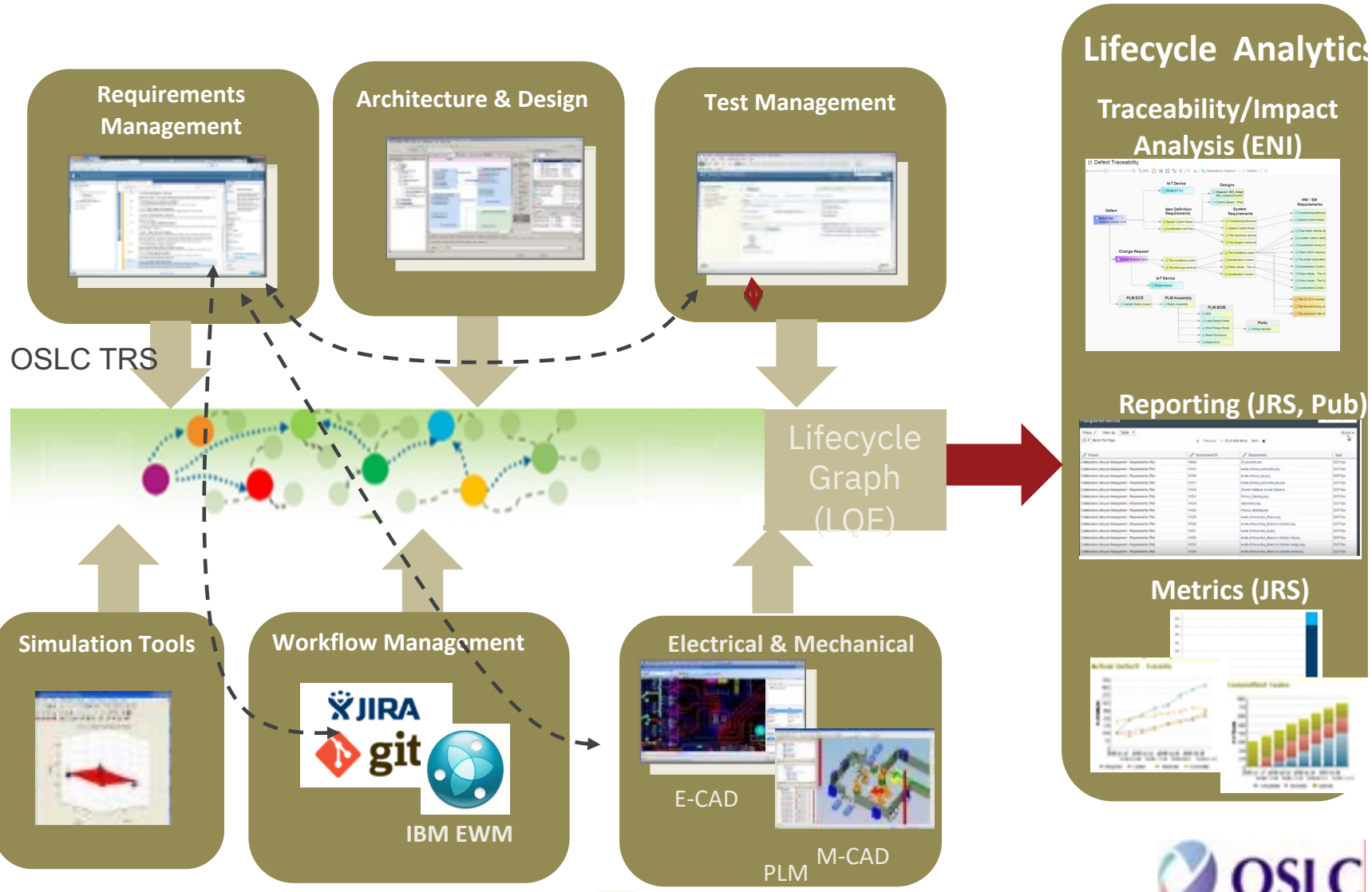
A hierarchical global configuration



Synthesizing digital viewpoints

Cross lifecycle viewpoints and analytics

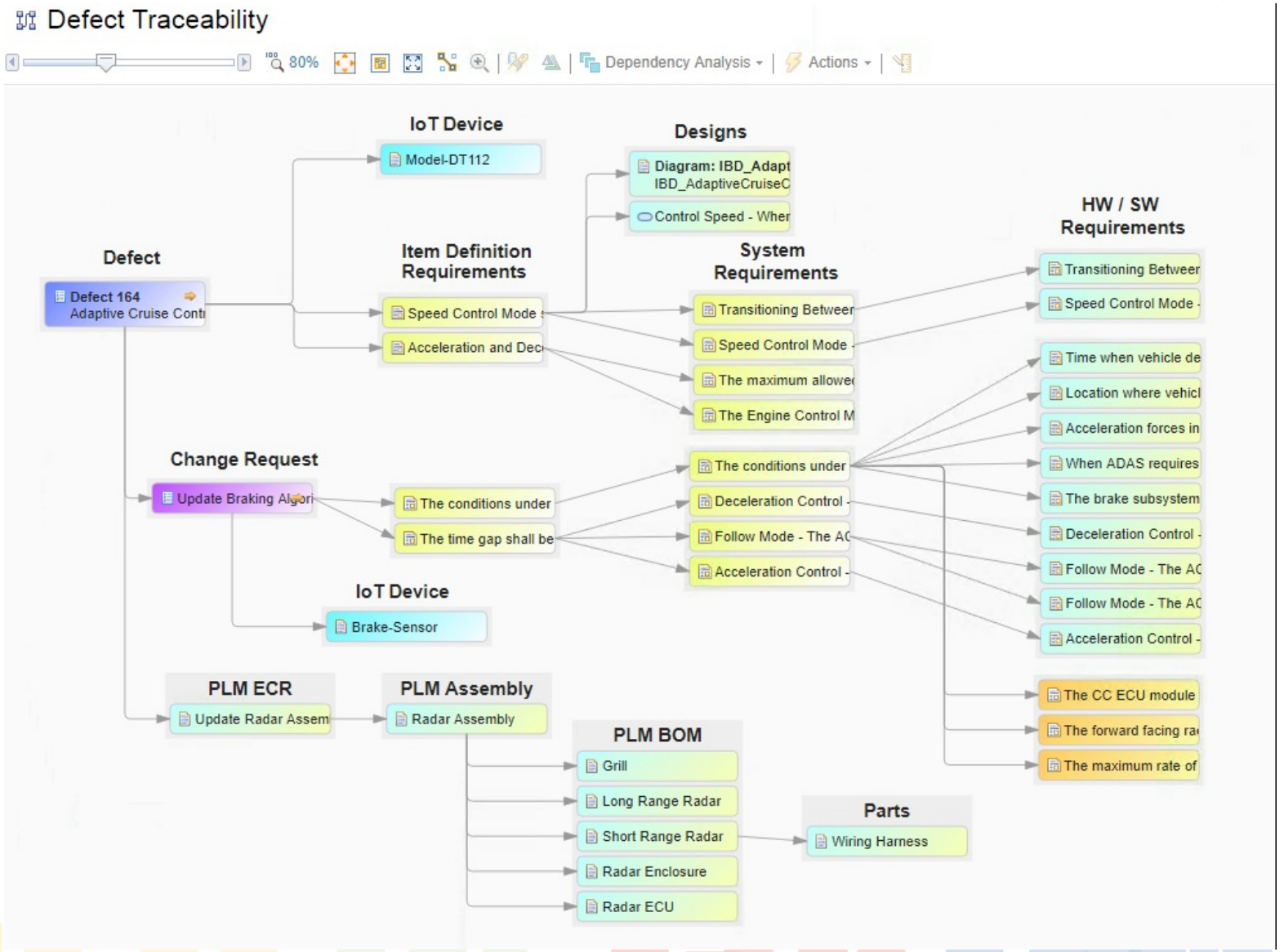
- Digital views are produced by dedicated reporting applications
- Reporting applications leverage a central lifecycle graph to generate the necessary views
- Lifecycle graph is continuously maintained using OSLC data tracking (TRS)
- The lifecycle Graph is an RDF graph and provides a SPRQL query API for view applications
- Supports decision making such as change impact analysis
- Generates necessary evidence for regulatory compliance
- Produces necessary KPIs for continuous process improvement



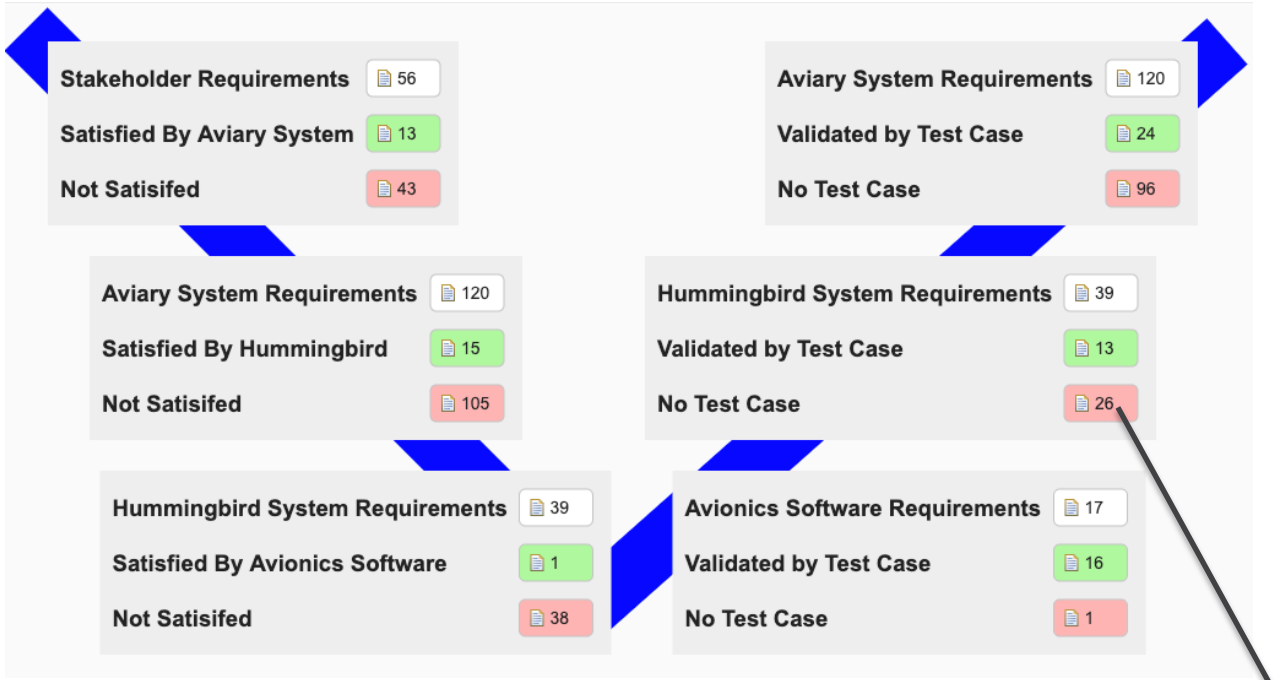
OSLC OASIS

Example: ELM Engineering insights – graph visualization

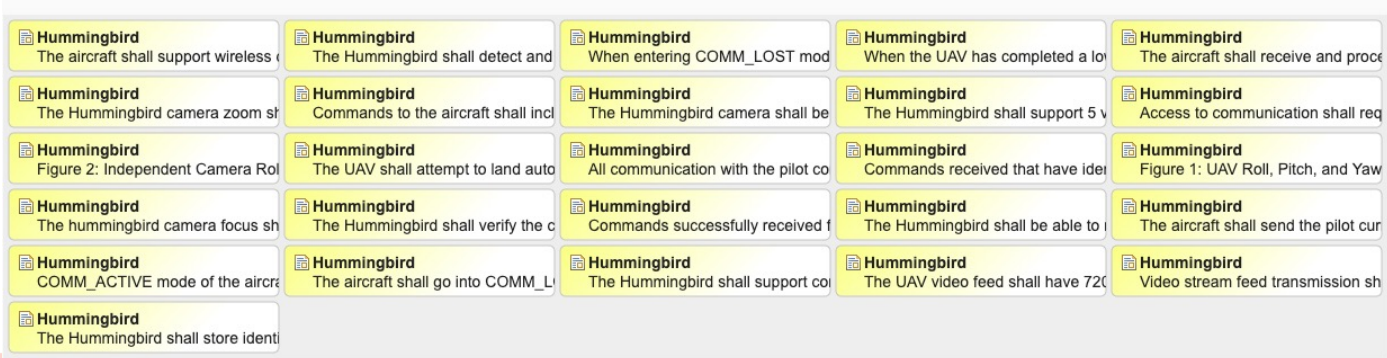
- An application that renders Graph like views to study the relationships across lifecycle data based on view templates
- ENI view templates are custom “self serve” defined by “drag and drop” gestures of element categories to the canvas
- Common usages for such graph like views are
  - Impact of change analysis based on a certain origin node (e.g. a requirement)
  - Coverage views that show how downstream elements realize upstream, elements



Lifecycle KPIs



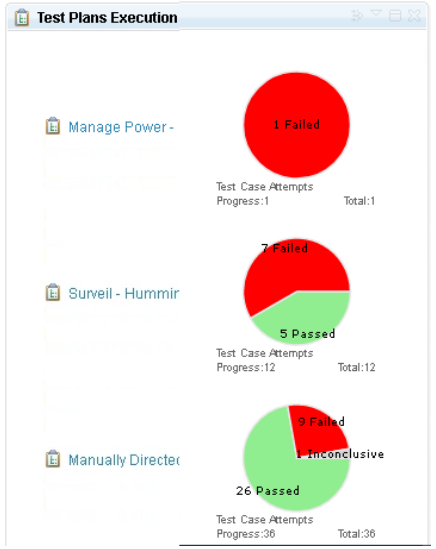
Drill down



JRS: Evidence Reports and KPIs

- An application that renders tabular reports and data aggregation graphs
- Reports are defined based on “self serve” templates using an interactive query builder, translated into SPARQL queries
- Reports are produced from actual lifecycle records fetched from LQE
- Common usages of JRS reports
  - Traceability coverage tables
  - Requirements Test coverage tables
  - System readiness metrics reports

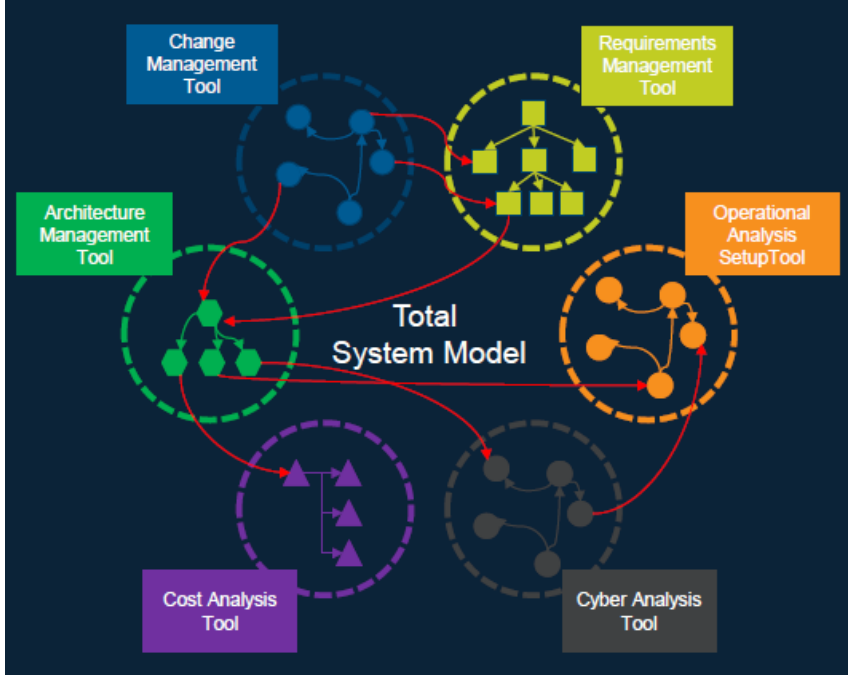
Subsystems verification summary



System requirements to test traceability report

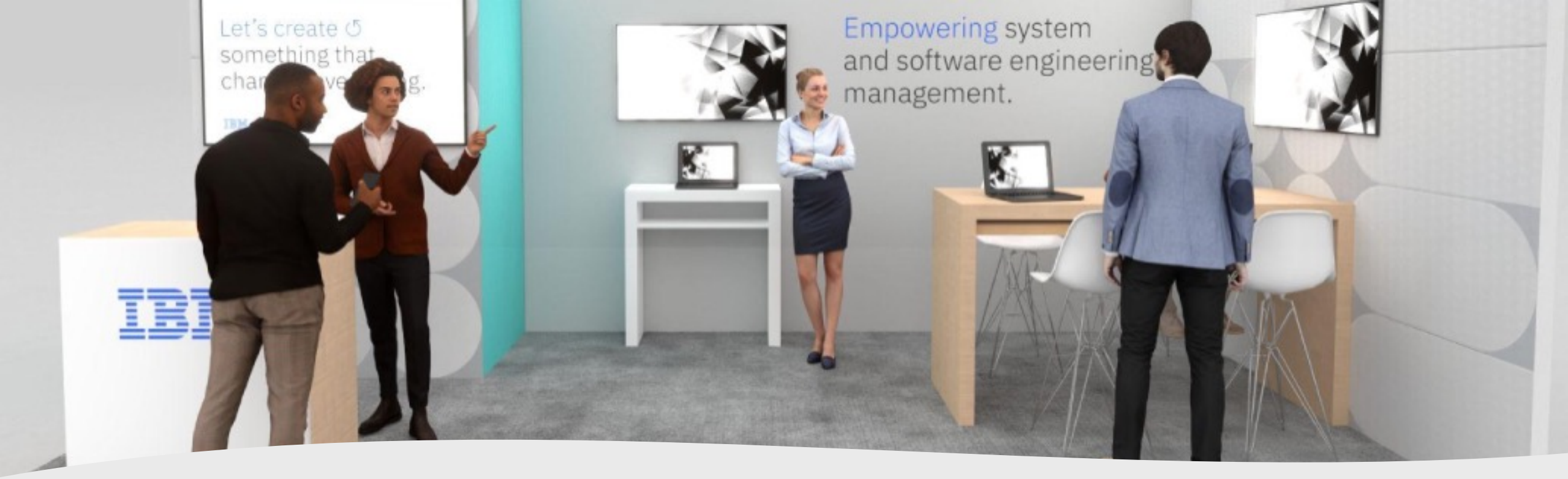
Req Id	Requirement	Tst Id	Test Case
4185	The Hummingbird shall be able to rotate independently of its direction of movement, either to the left or right to any number of degrees.	93	Test Rotation In Various Directions Of Movement
4169	The Hummingbird shall report its altitude above any surface immediately 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

Summary: Realizing Digital Viewpoints with OSLC



OSLC enables a digital fabric with:

- Connectivity across lifecycle models and Heterogeneous environments – digital threads
- Orchestrating “authoritative sources” for the entire system: OSLC Global Configurations
- Enabling cross domain data exchange based on standard vocabularies and queries
- Enabling cross lifecycle analytics and reporting: maintaining lifecycle graphs based on OSLC TRS



See it in action....

- Come by the IBM Expo Center booth
- Engage our team in conversation
  - Products
  - Challenges
  - Strategy
  - Trends & direction

32nd Annual INCOSE international symposium

hybrid event

Detroit, MI, USA
June 25 - 30, 2022

www.incose.org/symp2022