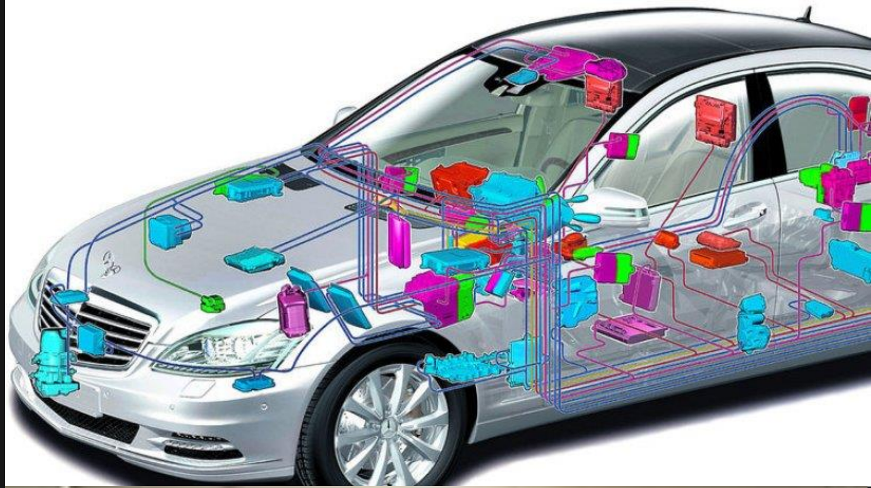
IBM **Engineering**

OSLC as a backbone for digital engineering

OSLCFest 2021

Eran Gery, Global industry solutions lead, IBM ELM



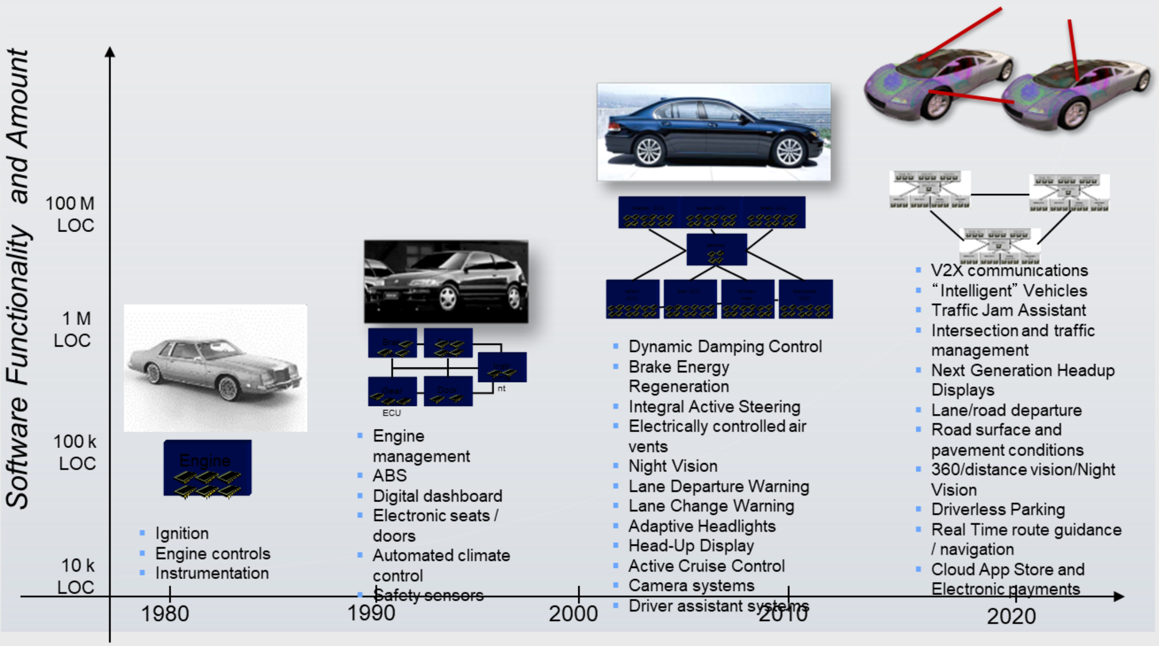
Outline

- Digital engineering challenges
- OSLC
- Realizing OSLC for digital engineering
    - Linking
    - Information exchange
    - Global configurations
    - Lifecycle analytics
- 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

- 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

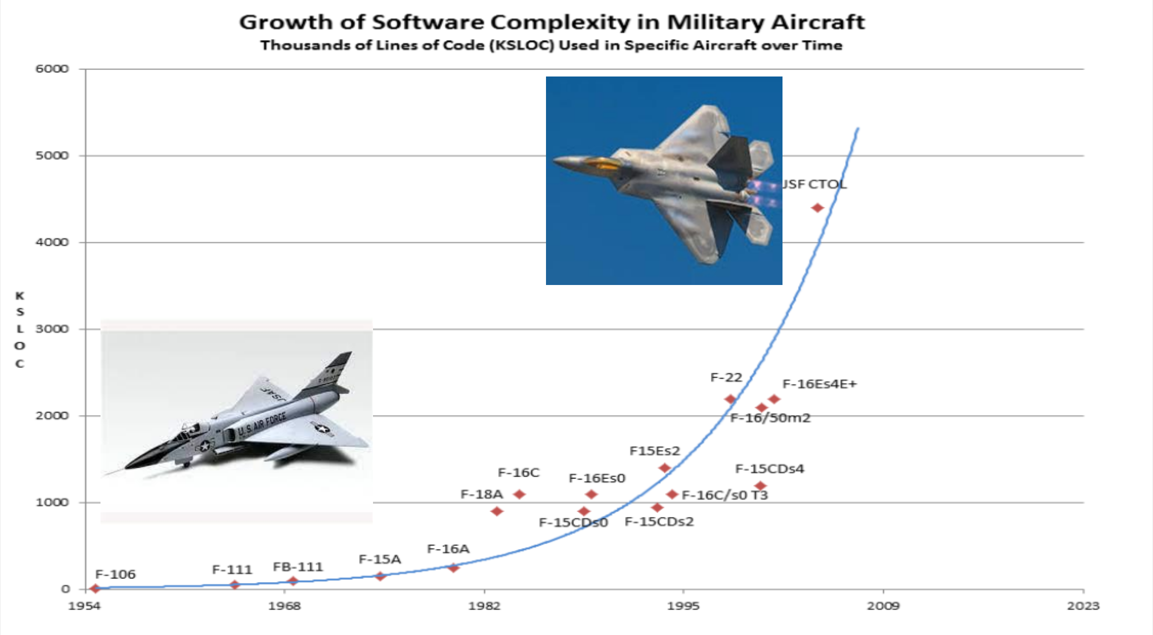


- Ignition
- Engine controls
- Instrumentation

- Engine management
- ABS
- Digital dashboard
- Electronic seats / doors
- Automated climate control
- Safety sensors

- Dynamic Damping Control
- Brake Energy Regeneration
- Integral Active Steering
- Electrically controlled air vents
- Night Vision
- Lane Departure Warning
- Lane Change Warning
- Adaptive Headlights
- Head-Up Display
- Active Cruise Control
- Camera systems
- Driver assistant systems

- V2X communications
- "Intelligent" Vehicles
- Traffic Jam Assistant
- Intersection and traffic management
- Next Generation Headup Displays
- Lane/road departure
- Road surface and pavement conditions
- 360/distance vision/Night Vision
- Driverless Parking
- Real Time route guidance / navigation
- Cloud App Store and Electronic payments

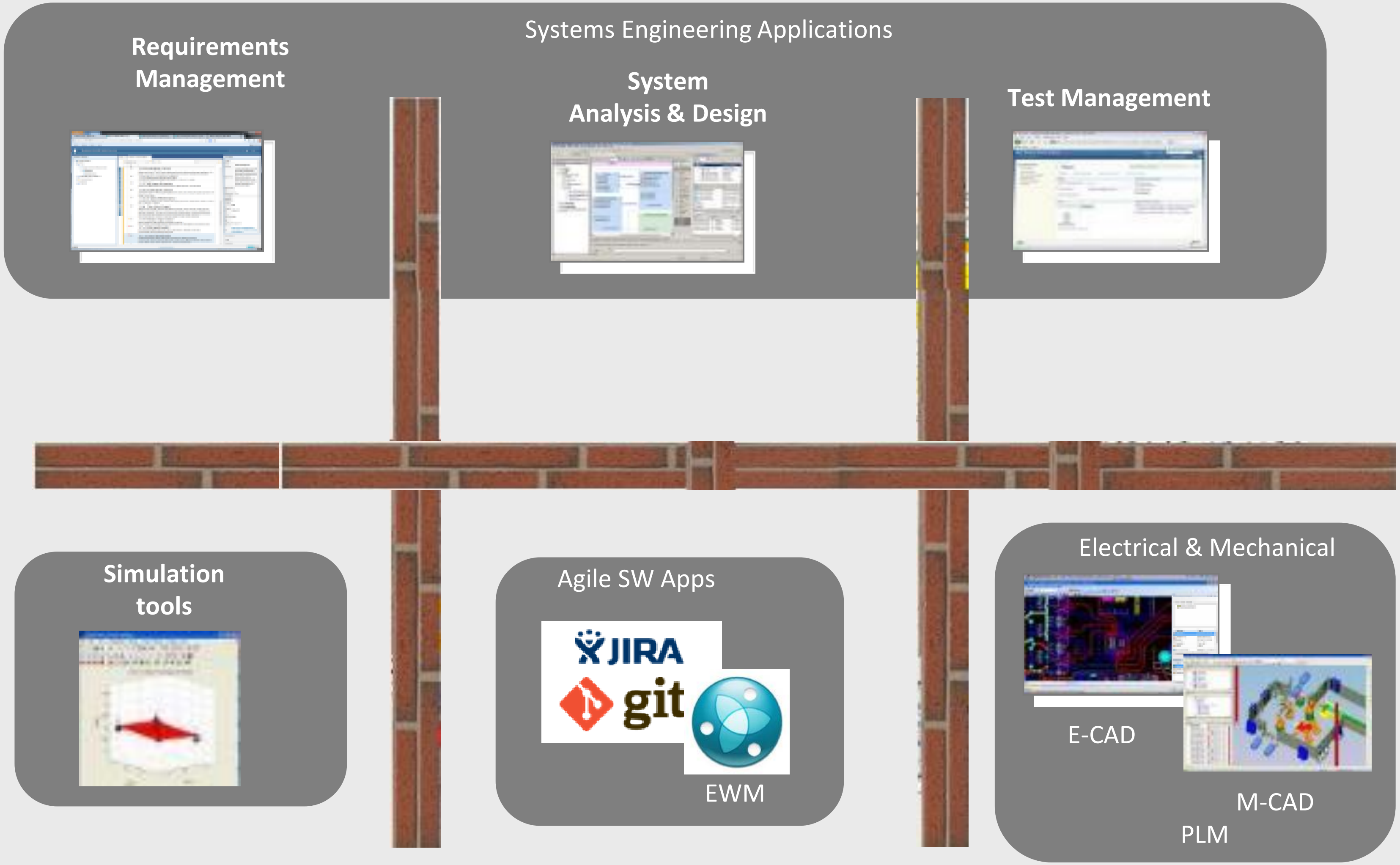

Growth of Software Complexity in Military Aircraft

Thousands of Lines of Code (KSLOC) Used in Specific Aircraft over Time

The current state of the practice...

Siloed tools and document centric, results in:

- Late discovery of issues -> expensive rework
- Slow velocity -> late threat and market response
- Ineffective doc-based supplier collaboration -> schedule & cost
- Reluctance to changes -> non-optimal designs
- Manual reporting and audit trails -> High costs of regulatory compliance



Systems Engineering Applications

Requirements Management

System Analysis & Design

Test Management

Simulation tools

Agile SW Apps

JIRA

git

EWM

Electrical & Mechanical

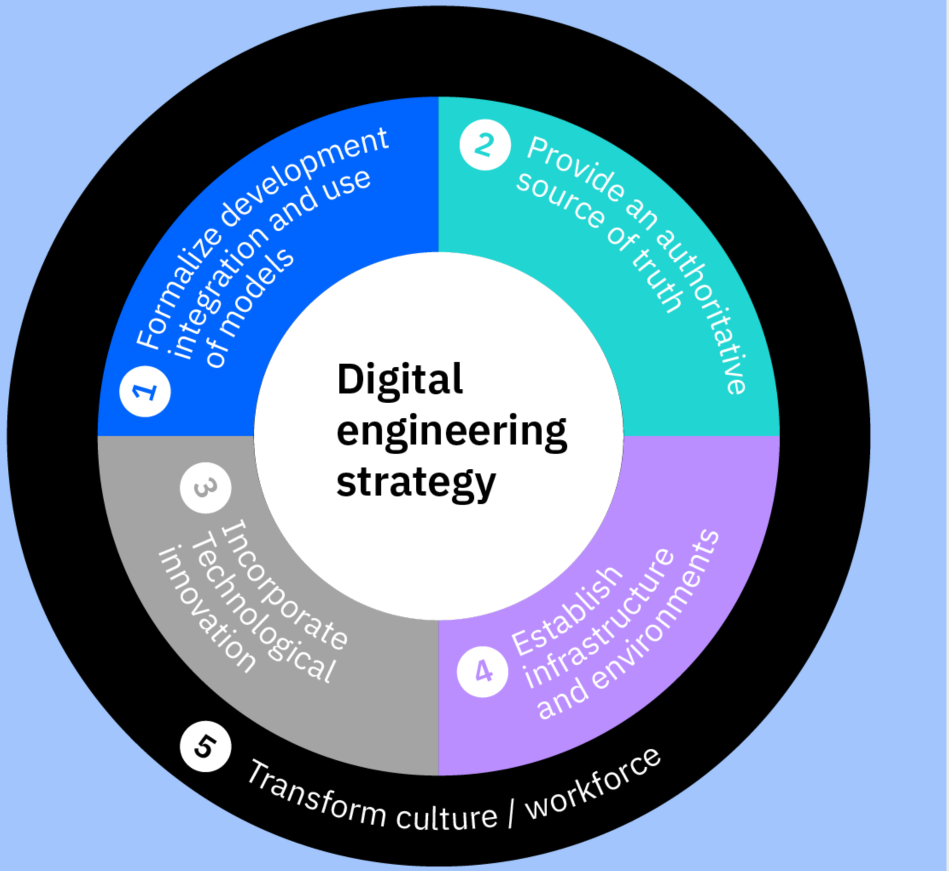
E-CAD

M-CAD

PLM

Industry vision: Digital Engineering

- Shift from document centric to digital representations (aka "models")
- 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 analysis and reporting
- Adopt a digital process with full transparency of planned and performed activity integrated with the data


Digital engineering strategy
1 Formalize development integration and use of models
2 Provide an authoritative source of truth
3 Incorporate Technological innovation
4 Establish infrastructure and environments
5 Transform culture / workforce

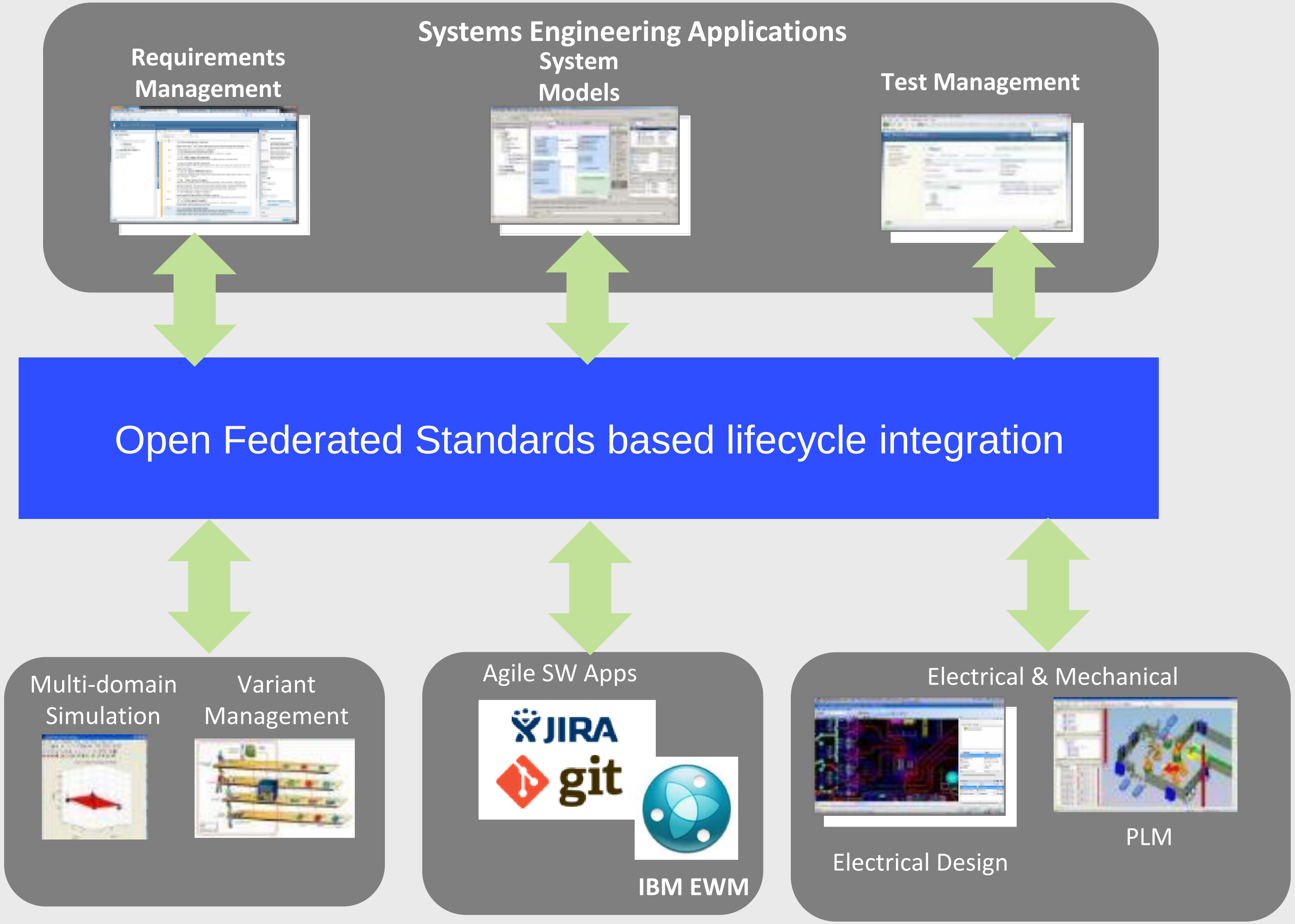
"…such engineering environments will allow DoD and industry partners to evolve designs at conceptual phase, reducing the need for expensive mockups, premature design lock, and physical testing." [1]

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

The need: a digital backbone based on an open and standards-based architecture

Why standards based: allow flexibility and avoid vendor locking around both the infrastructure and the engineering applications

Why federated: integrate the authoritative sources into the process rather than copy their data to a central tool to manage the lifecycle



Systems Engineering Applications

Requirements Management

System Models

Test Management

Open Federated Standards based lifecycle integration

Multi-domain Simulation

Variant Management

Agile SW Apps

IBM EWM

Electrical & Mechanical

Electrical Design

PLM

OSLC to address these digitization challenges

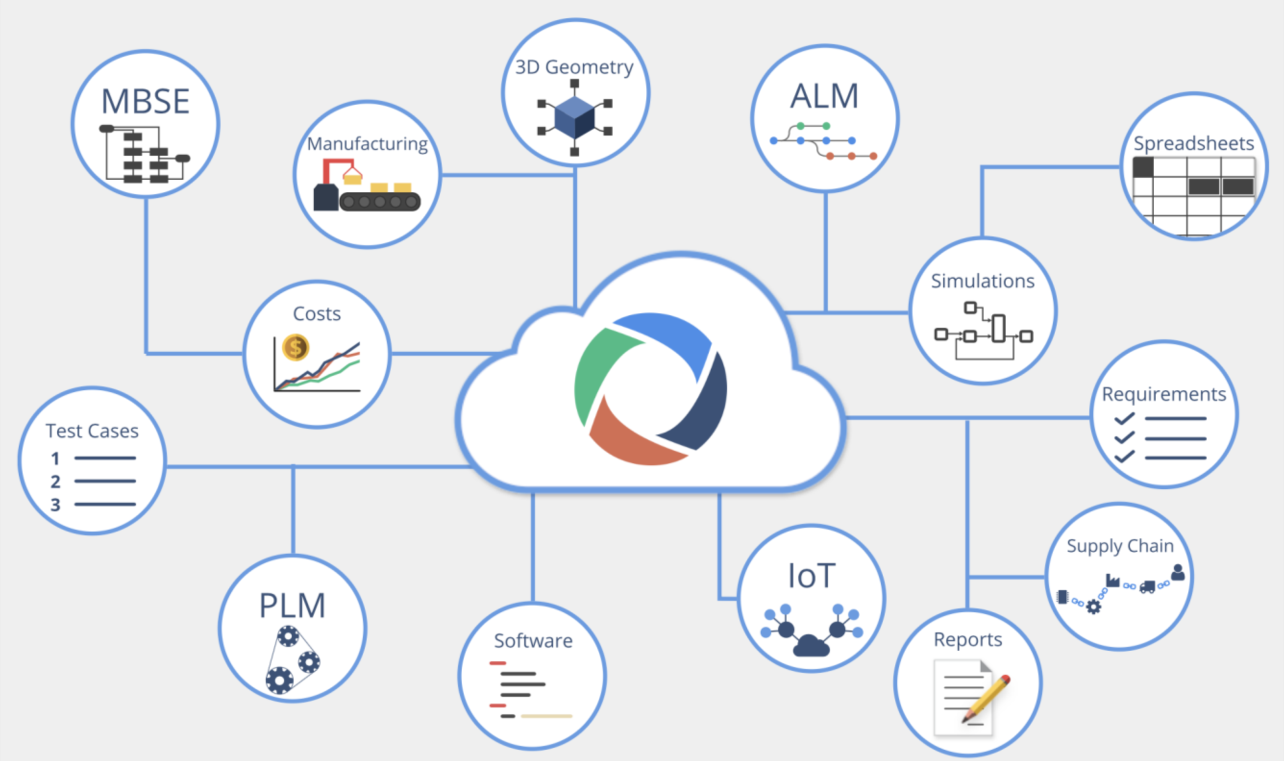
Work seamlessly across tools avoiding complex data synchronization

Modern HTTP/REST integration architecture

W3C Linked data architecture

Data and UI/workflow integration

Standard Domain vocabularies



No vendor lock

Complete lifecycle traceability

Open Standard and Open Source community

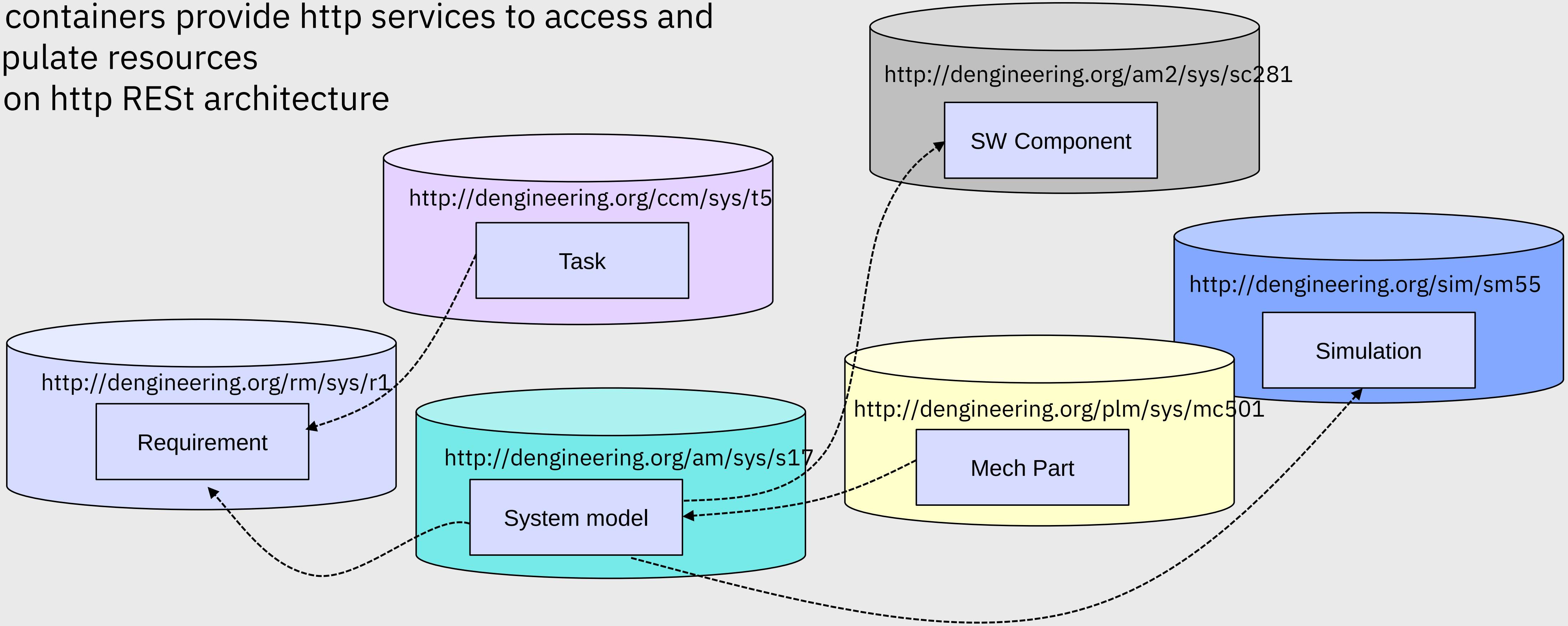
Better visibility

Increased reuse

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)

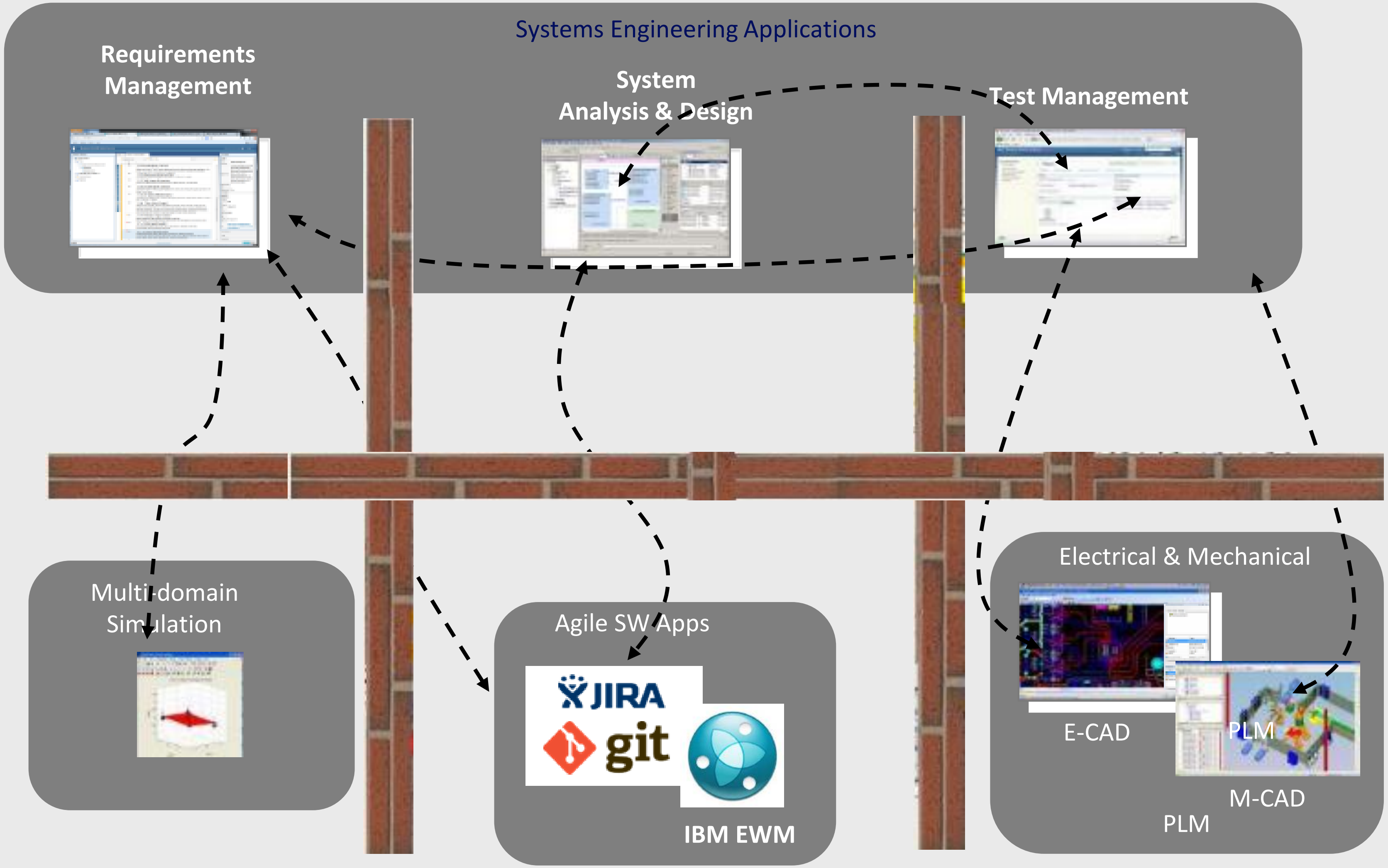
- Generalizing the idea of linked web documents for data
- Lifecycle objects (resources) are identified by http URLS
- Data containers provide http services to access and manipulate resources
- Best on http RESt architecture


http://dengineering.org/am2/sys/sc281
SW Component
http://dengineering.org/ccm/sys/t5
Task
http://dengineering.org/sim/sm55
Simulation
http://dengineering.org/rm/sys/r1
Requirement
http://dengineering.org/plm/sys/mc501
Mech Part
http://dengineering.org/am/sys/s17
System model

Foundation for quality: cross domain traceability with linked data

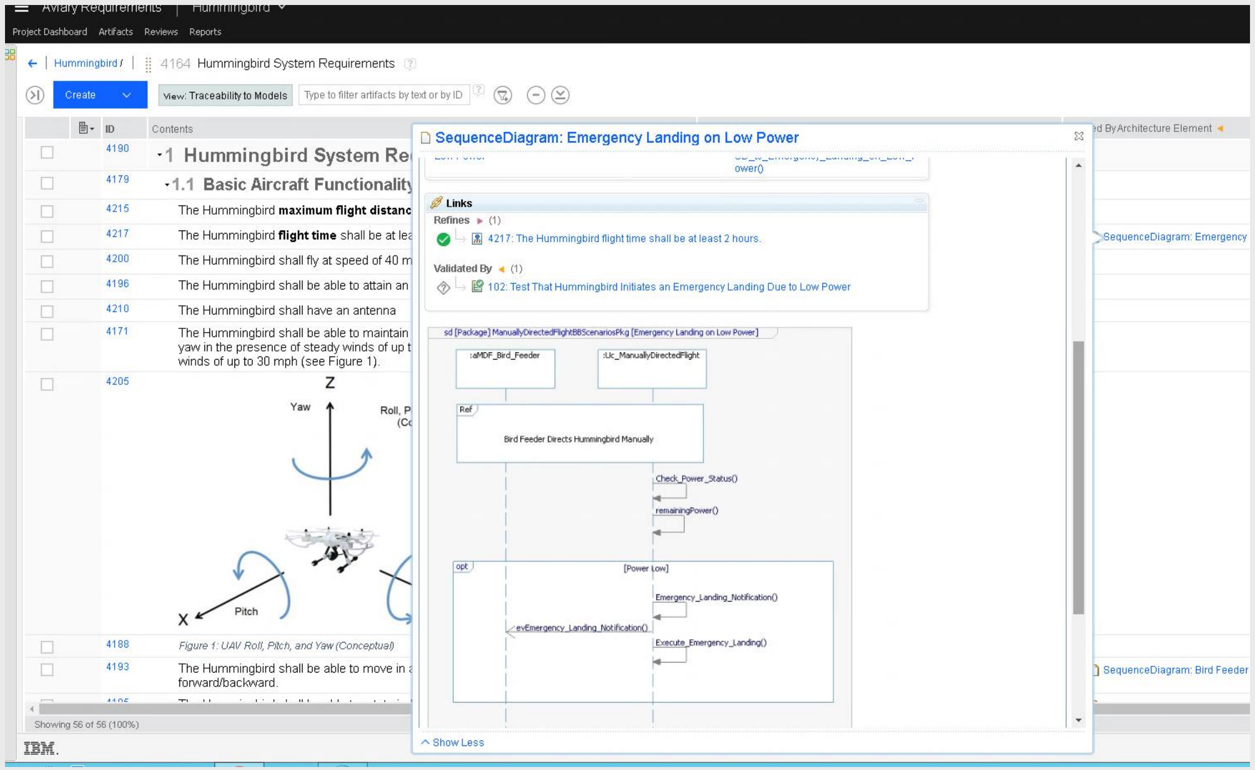
Digital continuity

- Provide digital representation of all lifecycle artifacts
- Establish relationships across lifecycle disciplines and artifacts – digital threads
- Maintain traceability and data consistency across engineering data
- Implements a lifecycle information model
- Lifecycle information model is a foundation for digital engineering



OSLC base linking services

- 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
  - Cross tool navigation
    - The requesting tool switches context to the element page in the hosting tool



4164 Hummingbird System Requirements

4190 1 Hummingbird System Re

4179 1.1 Basic Aircraft Functionality

4215 The Hummingbird **maximum flight distanc**

4217 The Hummingbird **flight time** shall be at lea

4200 The Hummingbird shall fly at speed of 40 m

4196 The Hummingbird shall be able to attain an

4210 The Hummingbird shall have an antenna

4171 The Hummingbird shall be able to maintain yaw in the presence of steady winds of up t winds of up to 30 mph (see Figure 1).

4205

4188 *Figure 1: UAV Roll, Pitch, and Yaw (Conceptual)*

4193 The Hummingbird shall be able to move in a forward/backward.

SequenceDiagram: Emergency

SequenceDiagram: Bird Feeder

Showing 56 of 56 (100%)

SequenceDiagram: Emergency Landing on Low Power

ower()

Links

Refines (1)

4217: The Hummingbird flight time shall be at least 2 hours.

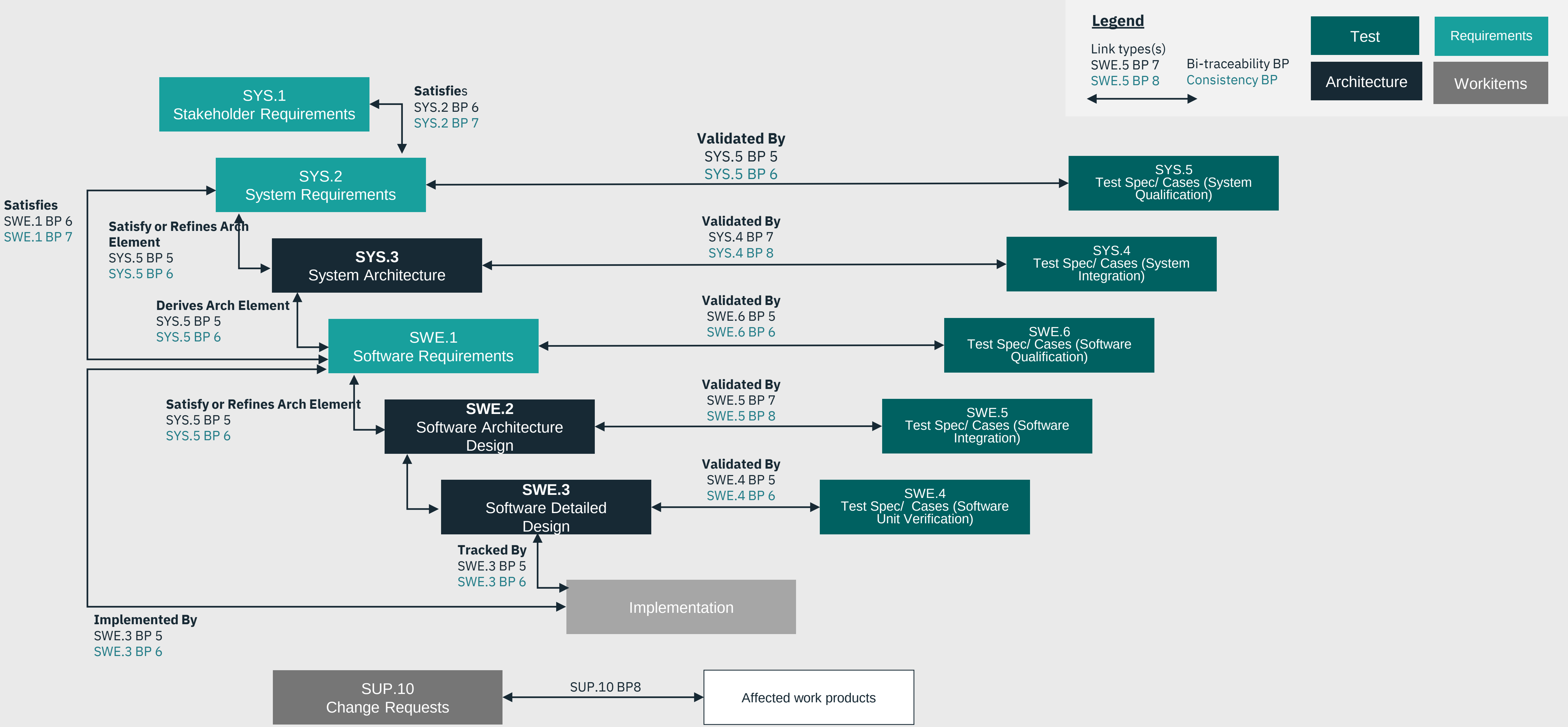
Validated By (1)

102: Test That Hummingbird Initiates an Emergency Landing Due to Low Power

sd [Package] ManuallyDirectedFlightBBScenariosPkg [Emergency Landing on Low Power]
:aMDF_Bird_Feeder
:Uc_ManuallyDirectedFlight
Ref
Bird Feeder Directs Hummingbird Manually
Check_Power_Status()
remainingPower()
opt
[Power Low]
Emergency_Landing_Notification()
evEmergency_Landing_Notification()
Execute_Emergency_Landing()

A model preview page linked to a requirement

Supporting industry information models: Example Automotive SPICE



Legend

Link types(s)

SWE.5 BP 7 Bi-traceability BP

SWE.5 BP 8 Consistency BP

Test

Requirements

Architecture

Workitems

SYS.1
Stakeholder Requirements

Satisfies
SYS.2 BP 6
SYS.2 BP 7

SYS.2
System Requirements

Validated By
SYS.5 BP 5
SYS.5 BP 6

SYS.5
Test Spec/ Cases (System Qualification)

Satisfies
SWE.1 BP 6
SWE.1 BP 7

Satisfy or Refines Arch Element
SYS.5 BP 5
SYS.5 BP 6

SYS.3
System Architecture

Validated By
SYS.4 BP 7
SYS.4 BP 8

SYS.4
Test Spec/ Cases (System Integration)

Derives Arch Element
SYS.5 BP 5
SYS.5 BP 6

SWE.1
Software Requirements

Validated By
SWE.6 BP 5
SWE.6 BP 6

SWE.6
Test Spec/ Cases (Software Qualification)

Satisfy or Refines Arch Element
SYS.5 BP 5
SYS.5 BP 6

SWE.2
Software Architecture Design

Validated By
SWE.5 BP 7
SWE.5 BP 8

SWE.5
Test Spec/ Cases (Software Integration)

SWE.3
Software Detailed Design

Validated By
SWE.4 BP 5
SWE.4 BP 6

SWE.4
Test Spec/ Cases (Software Unit Verification)

Tracked By
SWE.3 BP 5
SWE.3 BP 6

Implementation

Implemented By
SWE.3 BP 5
SWE.3 BP 6

SUP.10
Change Requests

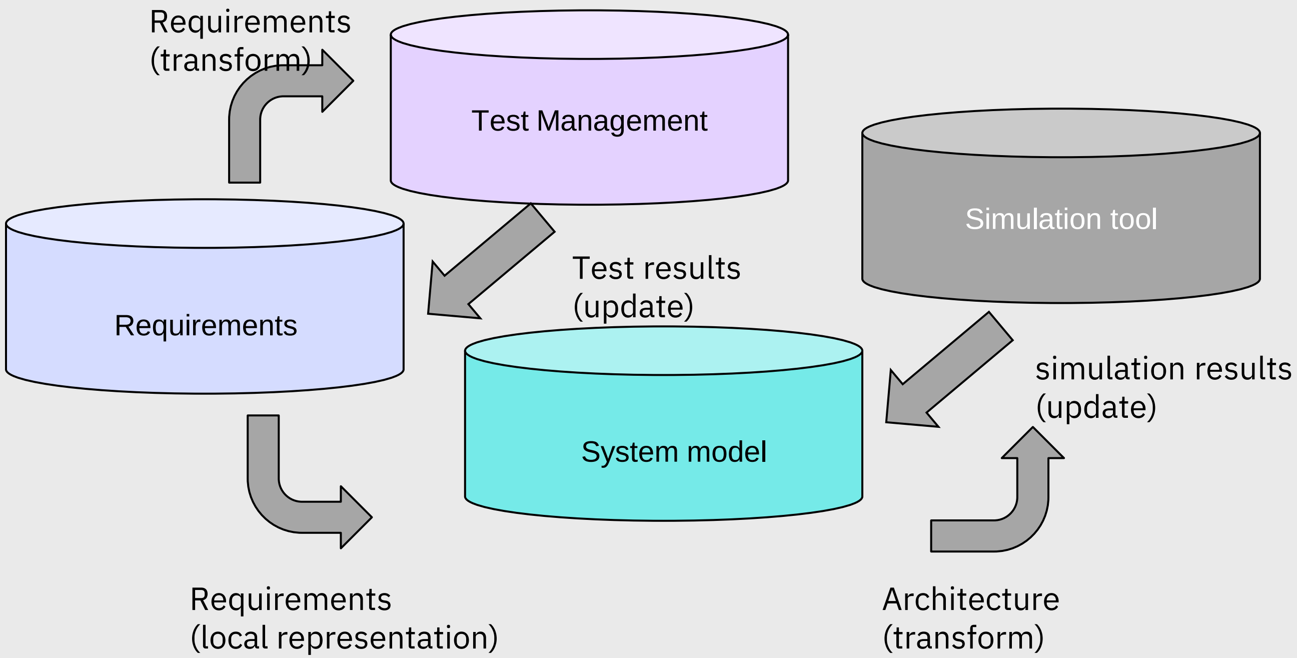
SUP.10 BP8

Affected work products

Data exchange to enable process continuity

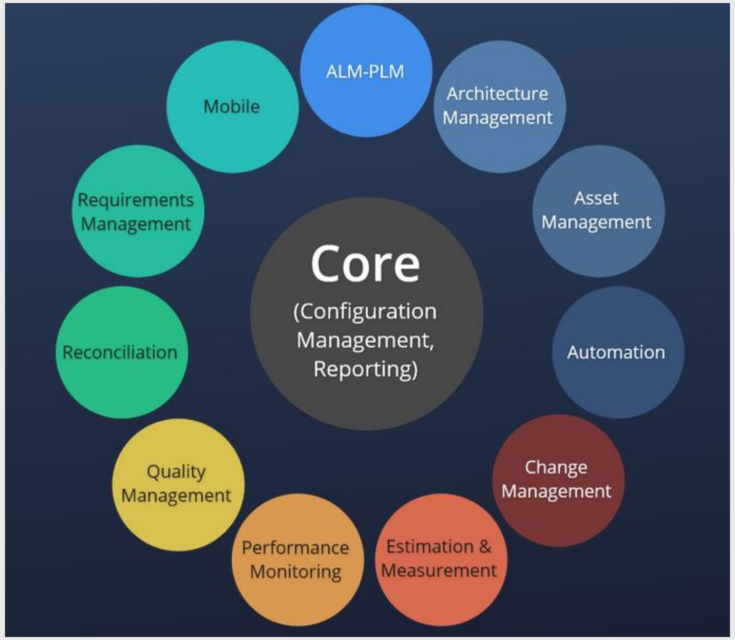
Data exchange

- Data exchange is needed to maintain consistency across domains
- Transform data from one domain to another
- Create local domain representations (e.g. Sysml requirements)
- Back propagation of properties up the chain



Enabling standard data exchange across domains

- OSLC represents resources using RDF
- Standard domain specifications using OSLC vocabularies and shapes
- Data can be discovered using OSLC query
- Data can be fetched and updated using http get/set requests



Standard domain resource representations

A change request RDF (turtle) representation
<https://example.org/ccm/resource/itemName/com.ibm.team.workitem.WorkItem/9>
a oslc_**cm:ChangeRequest** ;
dcterms**:creator** <https://example.org/jts/users/deb> ;
oslc:**modifiedBy** <https://example.org/jts/users/deb> ;
dcterms:**title** "To many messages logged in the console"^^rdf:XMLLiteral .

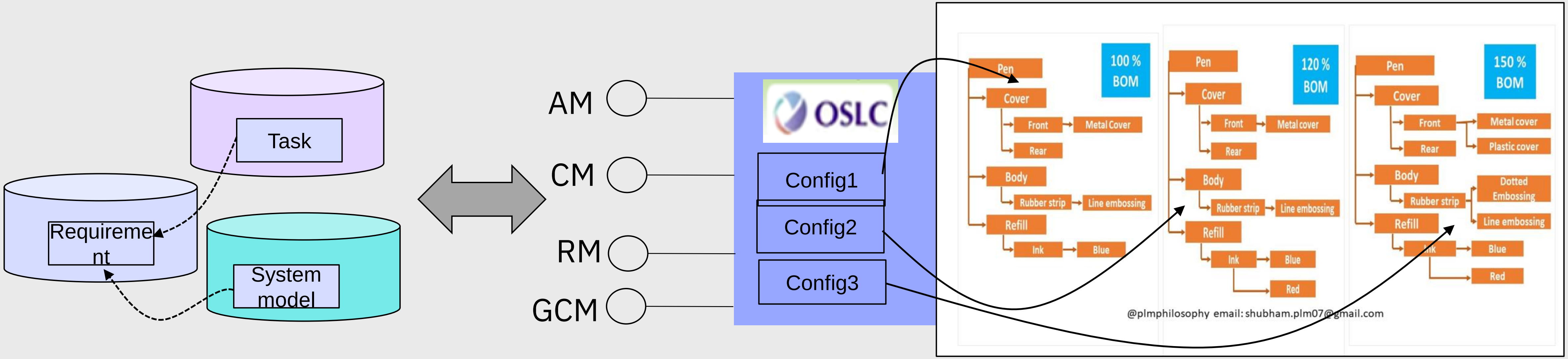
OSLC Query

Select title, creator, modifier of workitems created by deb
GET
https://example.org/ccm/oslc/contexts/_by884MNWEeekg_dNxwf1pg/**workitems**
?**oslc.where**=dcterms%3A**creator**%20%7Bfoaf%3A**name**%3D%22**Deb**%22%7
D
&**oslc.select**=dcterms%3A**title**%2Cdcterms%3A**creator**%2Coslc%3A**modified**
By%3Amodifier%7Bfoaf%3Aname%7D

And what about PLM tools?

“native” oslc providers

PLM

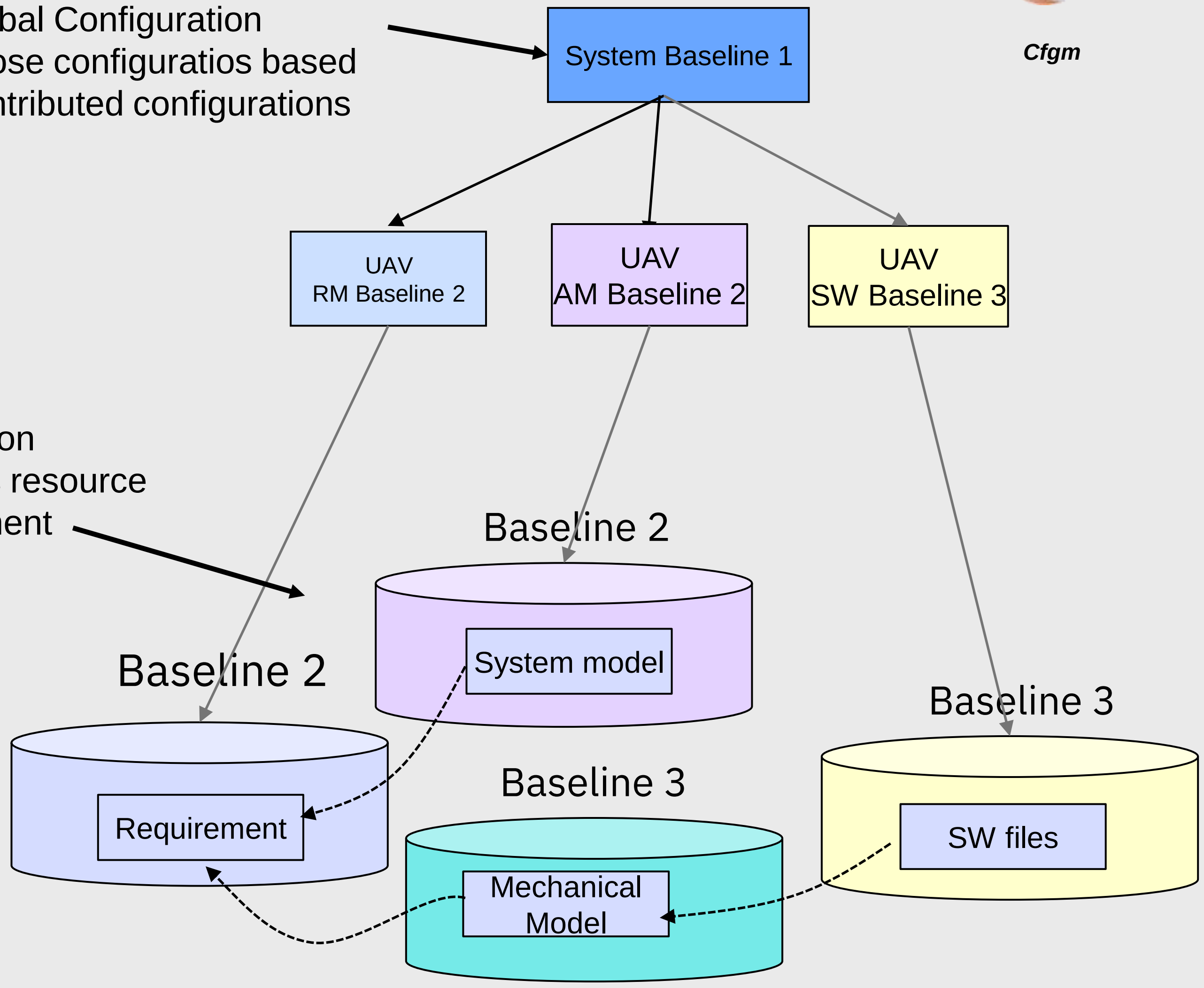


Global configuration management
Managing Data Across the entire digital thread

Data consistency

Cfgm

- Consistent evolution of data across engineering disciplines: common baselining
- Manage platform assets across variants and programs
- Reuse all engineering assets from the platform: requirements, design, implementation, test
- Manage changes across variants and programs
- Harvest innovation in programs for reuse across the product lines



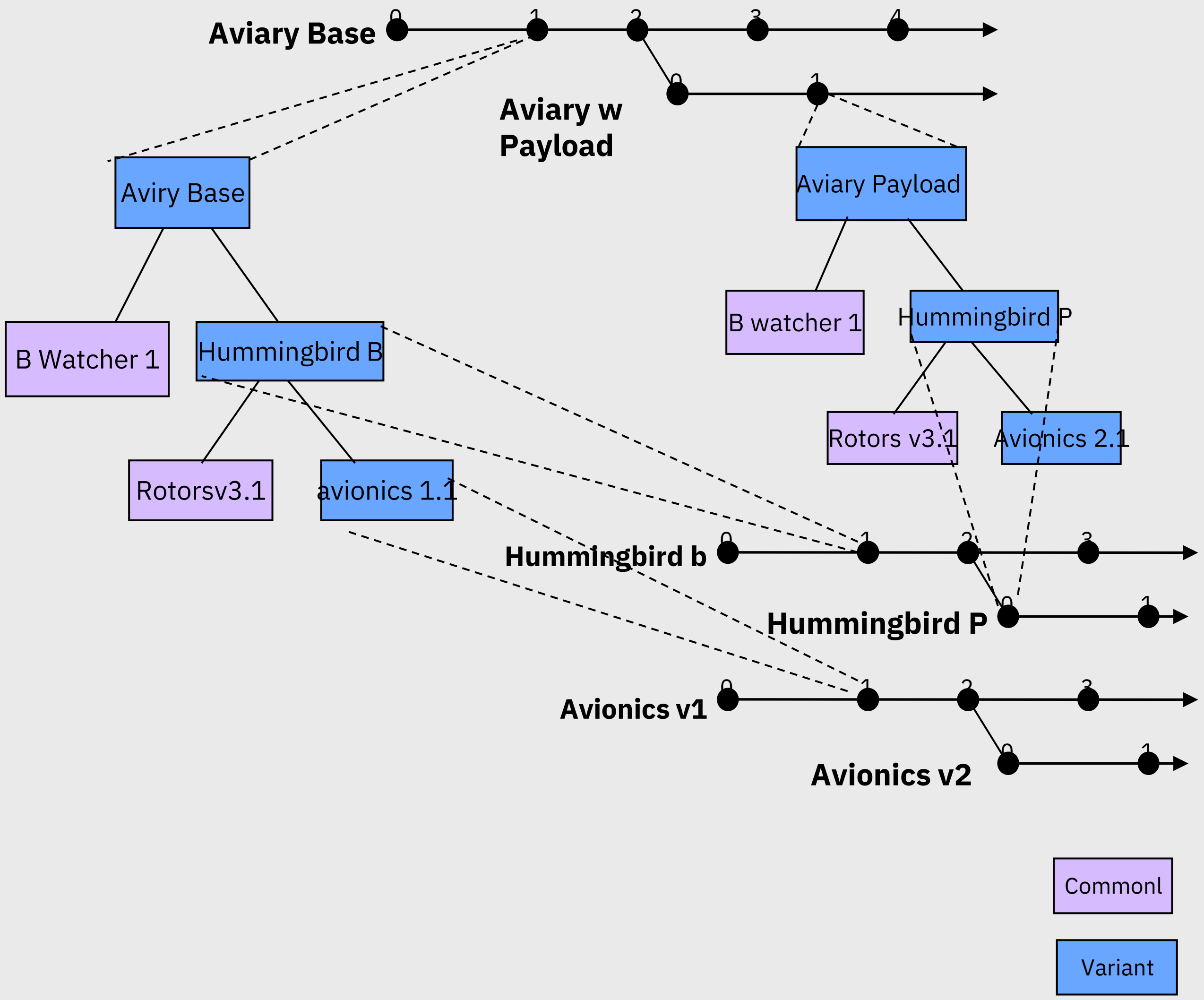
A Global Configuration compose configuratios based on contributed configurations

A Local Configuration provider – manages resource versions as component configurations

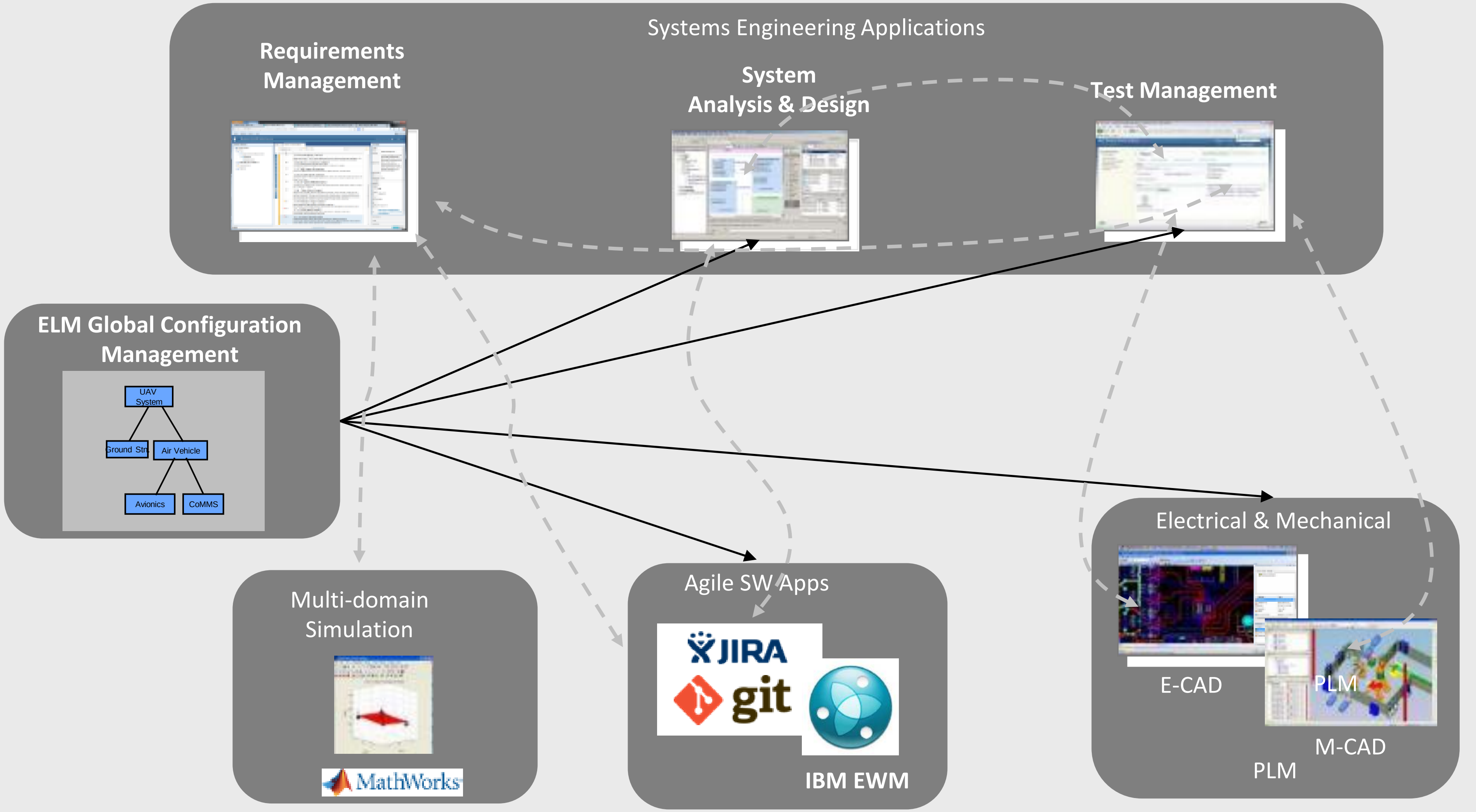
Enabling reuse and variant management at system and component levels

- The hierarchical nature of GCs enables variant configurations of complete systems, subsystems, and components
- Like LC’s, GCs have streams and baselines
  - Baselining entire structure
  - Manage variants of entire structures
- Supports “component library” reuse strategies and 150% systems/subsystem derivation strategies
- Integrates nicely with parametric variant management approaches such as feature based PLE

Example: component reuse across a UAV system (Aviary) variants



Digital Thread with Global Configuration management



Systems Engineering Applications

Requirements Management

System Analysis & Design

Test Management

ELM Global Configuration Management

Multi-domain Simulation

Agile SW Apps

IBM EWM

Electrical & Mechanical

E-CAD

PLM

M-CAD

PLM

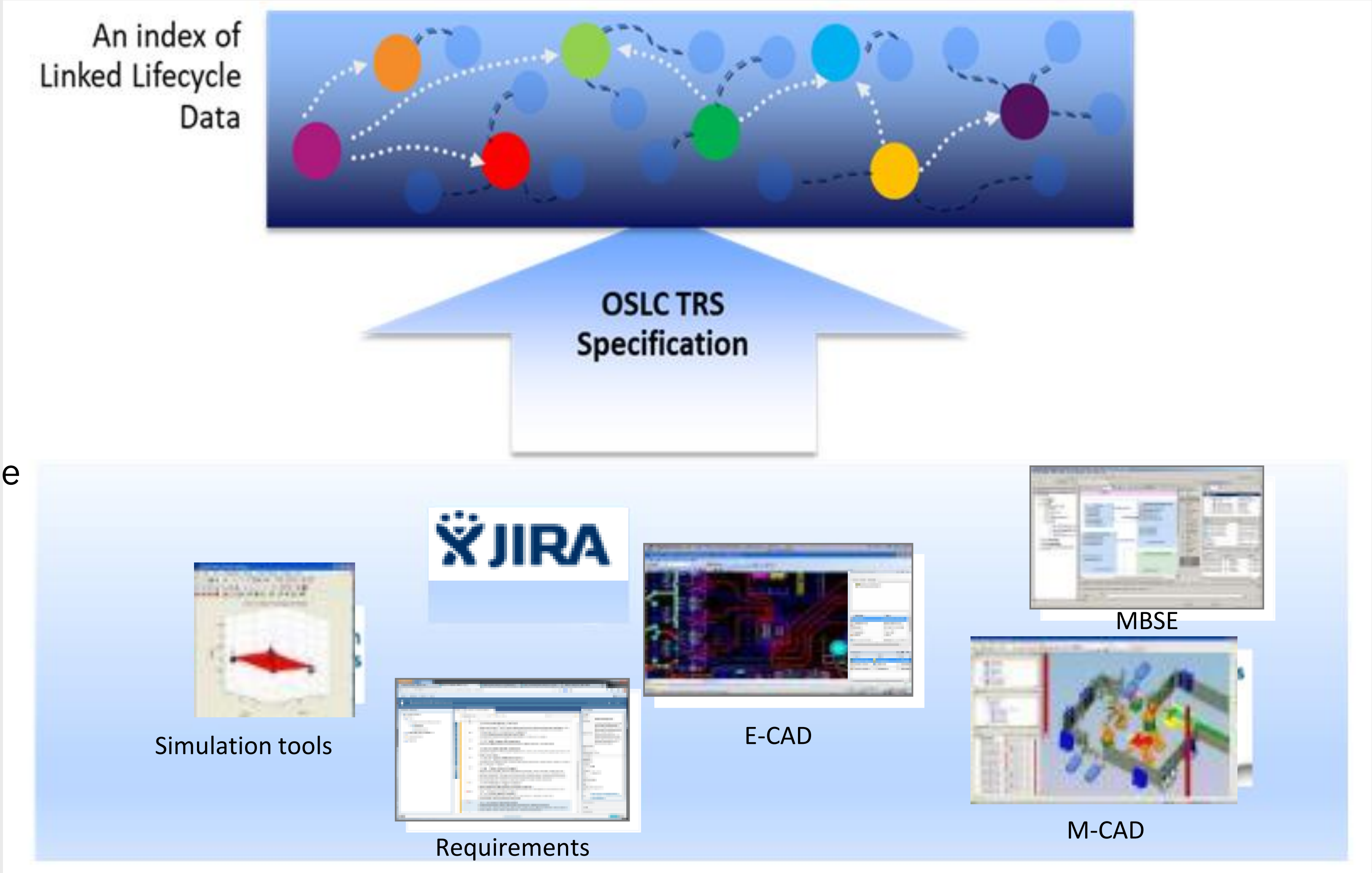
OSLC

OASIS

Enabling Cross Lifecycle Analytics with OSLC TRS

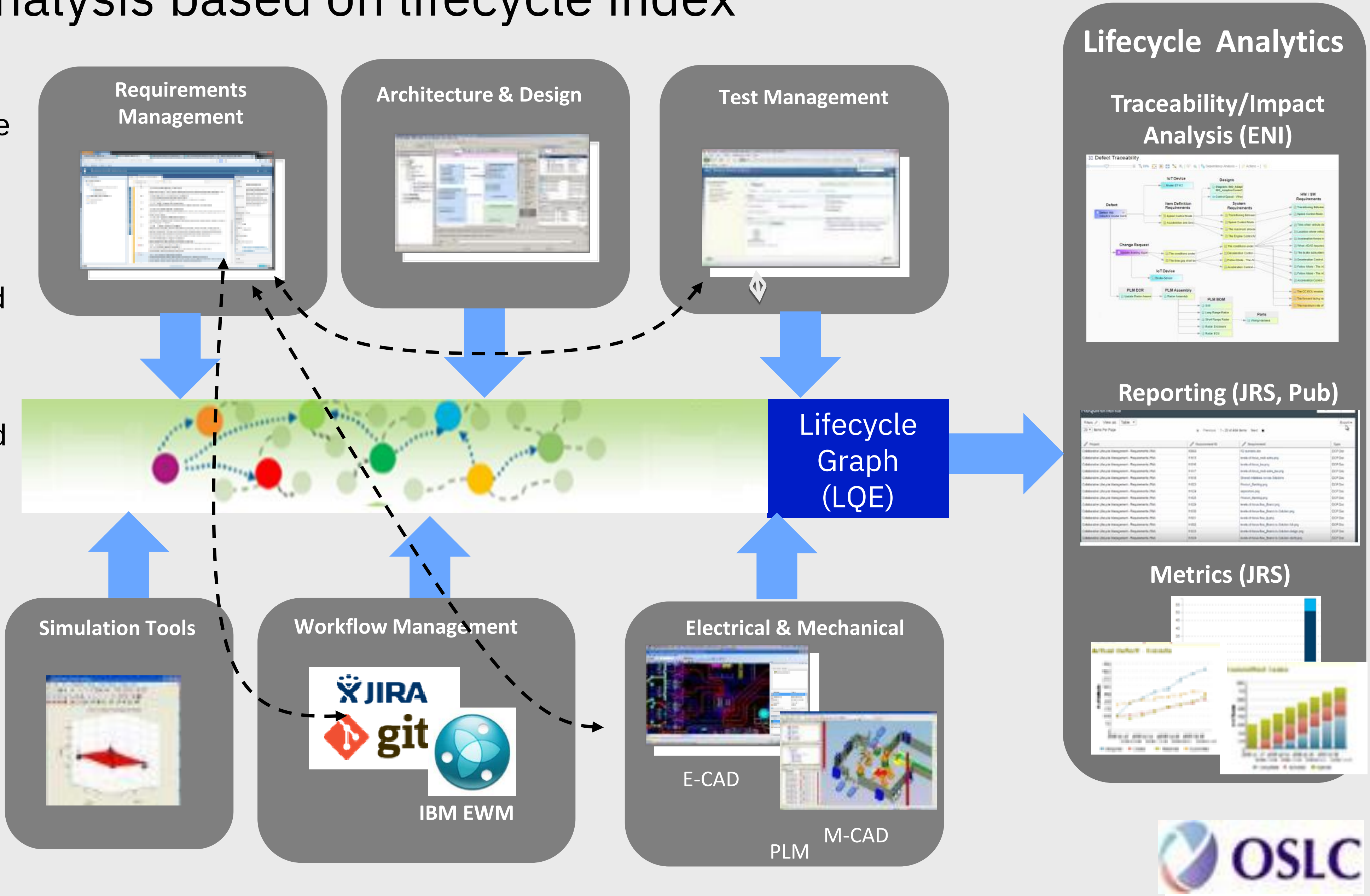
Cross Lifecycle Analysis and reporting

- Sustain a central analytics repository based on TRS subscription
- Supporting decision making
  - What is the impact of change
- Enable lifecycle reporting
- Producing all necessary evidence for engineering regulatory compliance
- Implemented by IBM LQE and others

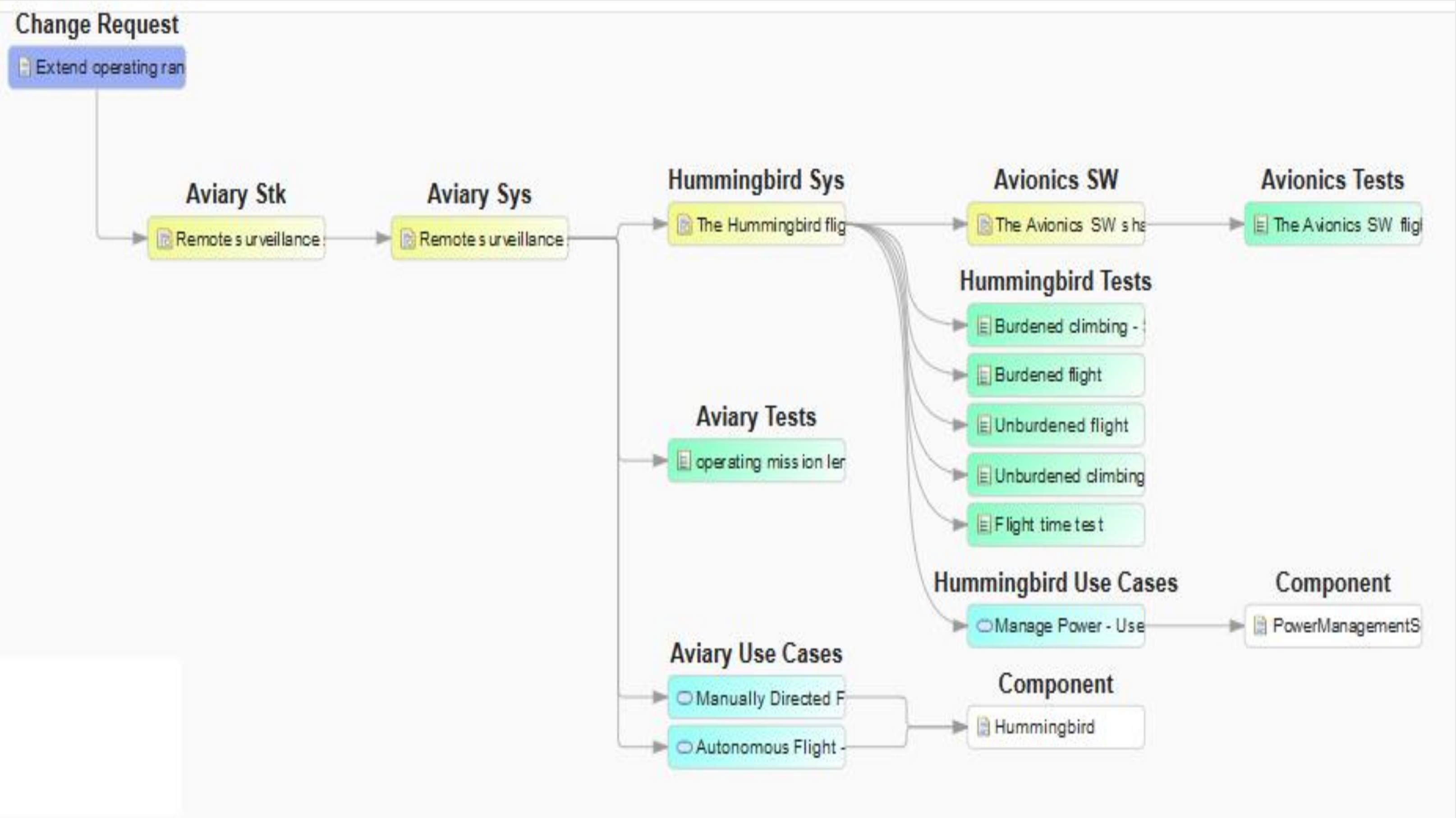


Reporting and Analysis based on lifecycle index

- Reporting, analysis, and visualization tools leverage the TRS based lifecycle index
- production of documents such as device history and risk management files
- Provides continuous visibility to compliance and progress metrics
- Supports decision making such as change impact analysis



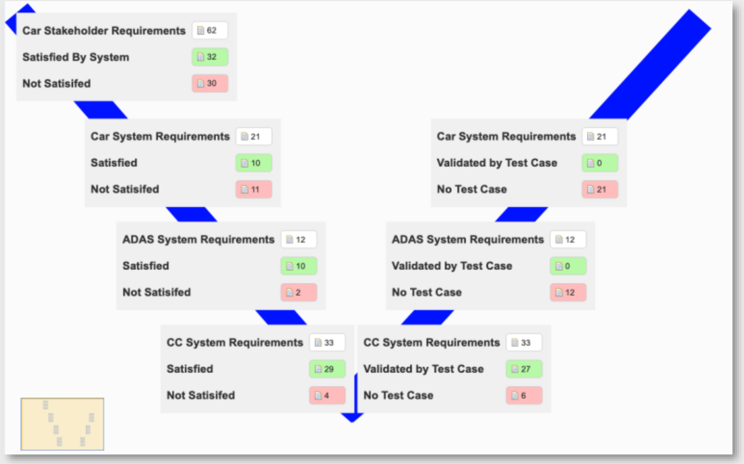
Example - Visualizing the digital thread: ELM Engineering insights



Example: Evidence Reports and KPIs with JRS

- 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



Car Stakeholder Requirements 62
Satisfied By System 32
Not Satisifed 30

Car System Requirements 21
Satisfied 10
Not Satisifed 11

ADAS System Requirements 12
Satisfied 10
Not Satisifed 2

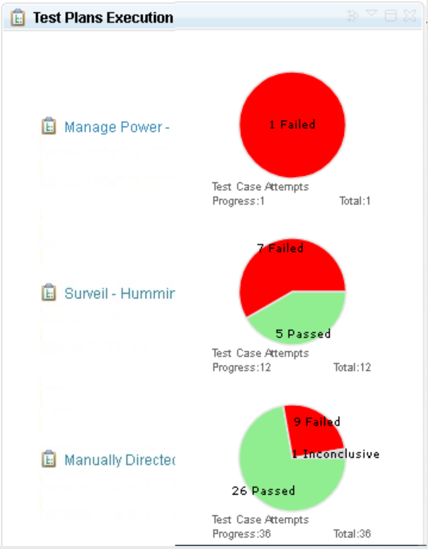
CC System Requirements 33
Satisfied 29
Not Satisifed 4

CC System Requirements 33
Validated by Test Case 27
No Test Case 6

ADAS System Requirements 12
Validated by Test Case 0
No Test Case 12

Car System Requirements 21
Validated by Test Case 0
No Test Case 21

Subsystems verification summary



Test Plans Execution

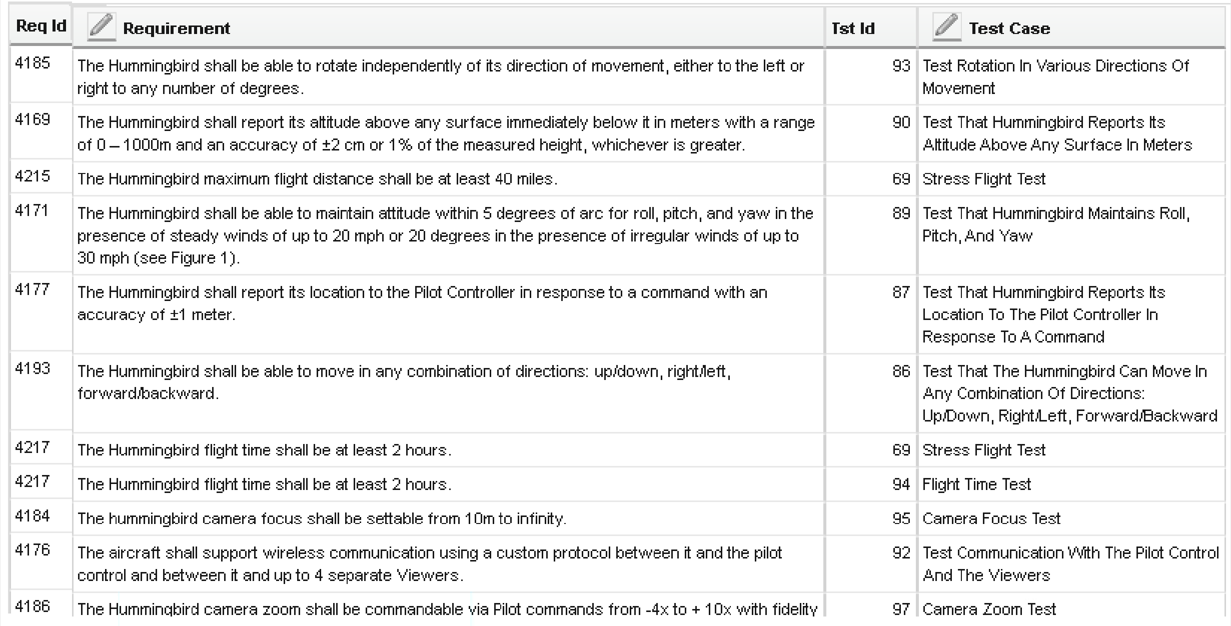
Manage Power -
1 Failed
Test Case Attempts
Progress:1 Total:1

Surveil - Hummir
7 Failed
5 Passed
Test Case Attempts
Progress:12 Total:12

Manually Directe
9 Failed
1 Inconclusive
26 Passed
Test Case Attempts
Progress:36 Total:36

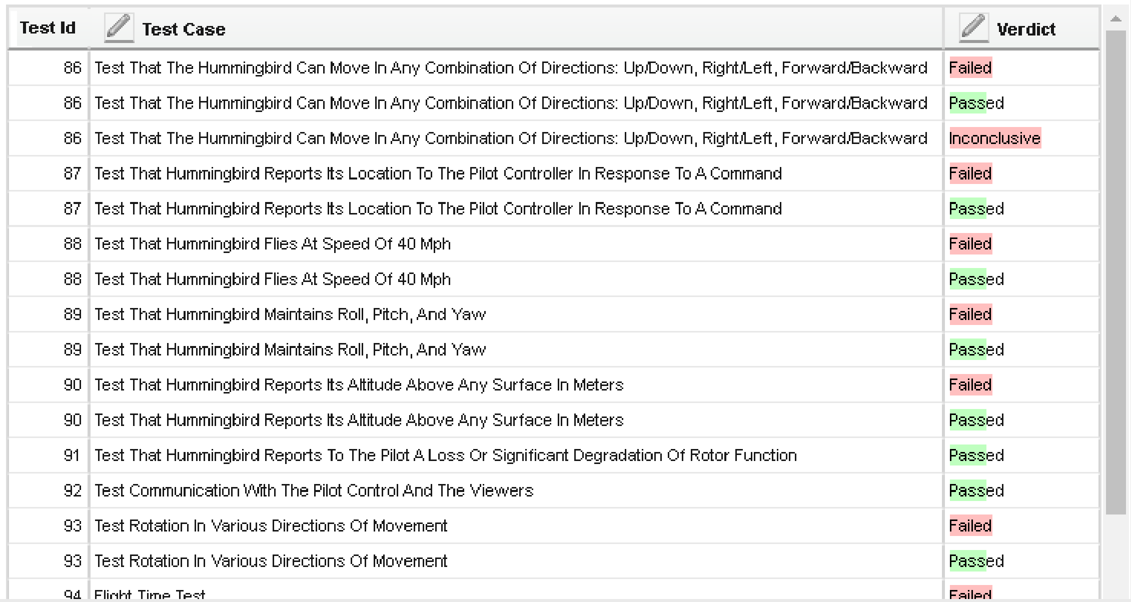
e it!

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

System verification report

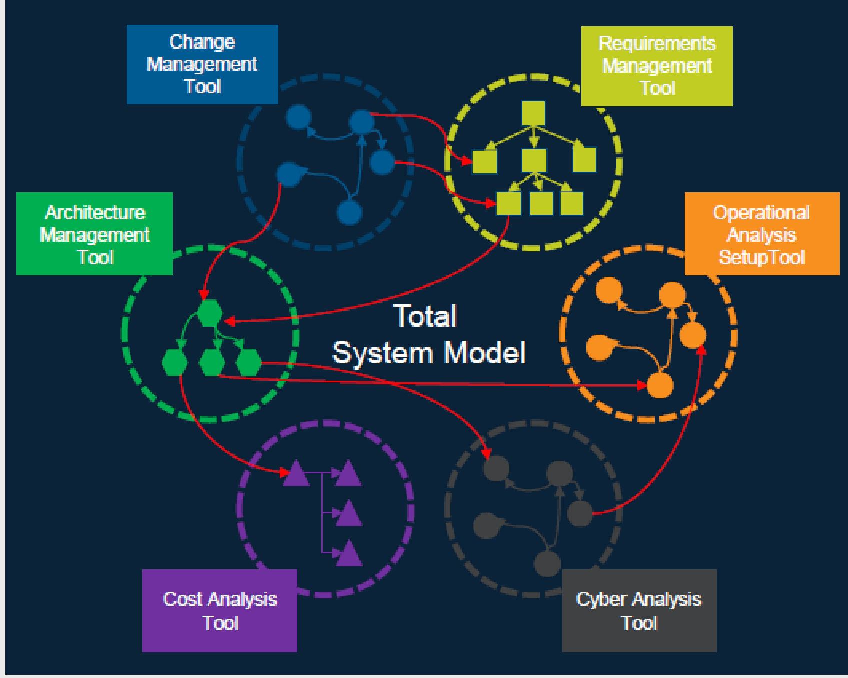


Test Id	Test Case	Verdict
86	Test That The Hummingbird Can Move In Any Combination Of Directions: Up/Down, Right/Left, Forward/Backward	Failed
86	Test That The Hummingbird Can Move In Any Combination Of Directions: Up/Down, Right/Left, Forward/Backward	Passed
86	Test That The Hummingbird Can Move In 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

Summary: Digital Engineering 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


Change Management Tool
Requirements Management Tool
Architecture Management Tool
Operational Analysis SetupTool
Total System Model
Cost Analysis Tool
Cyber Analysis Tool

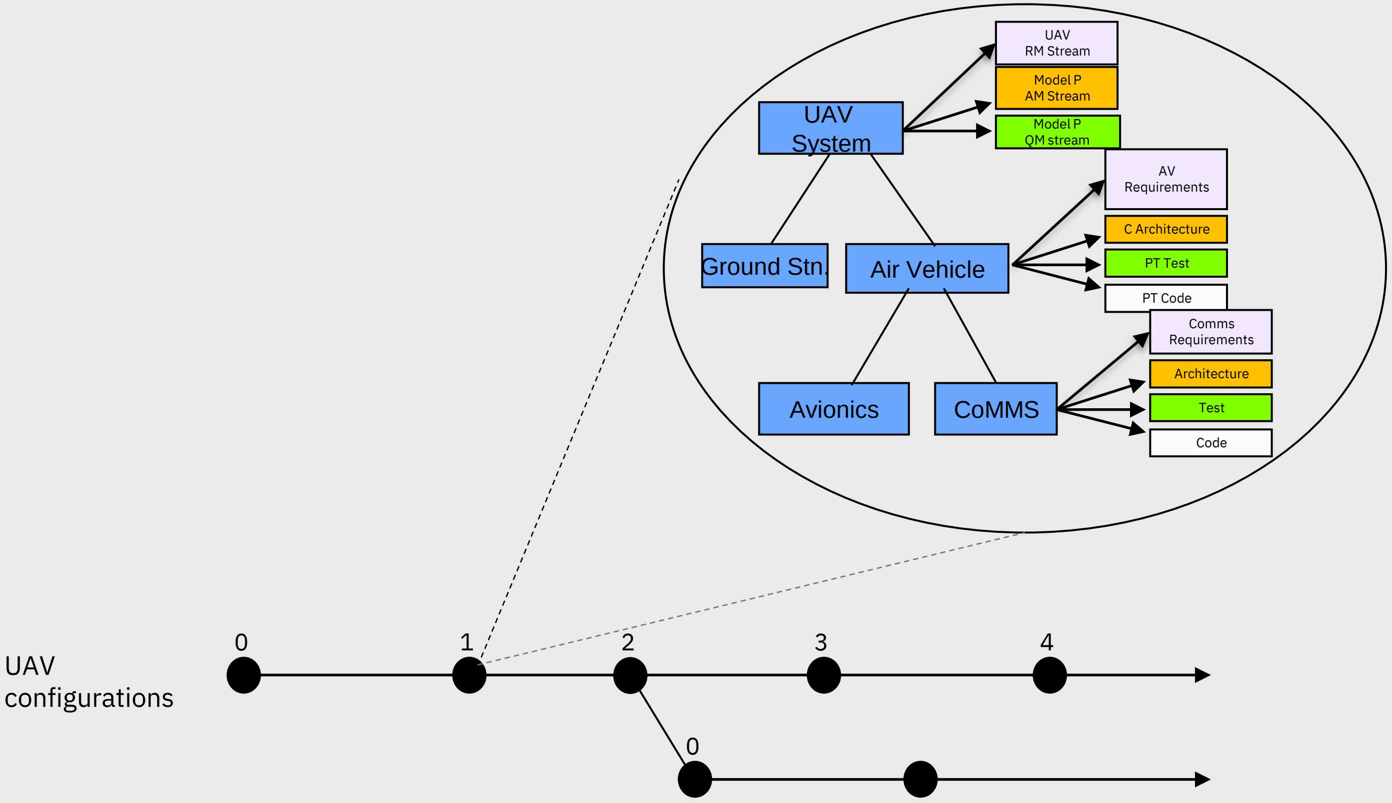
Future work needed:

- Standardize additional used functionalities
  - Link index (reverse linking)
  - Link validity
- Standardize more domains!

IBM

Creating design breakdown structures based on hierarchical configuration structure

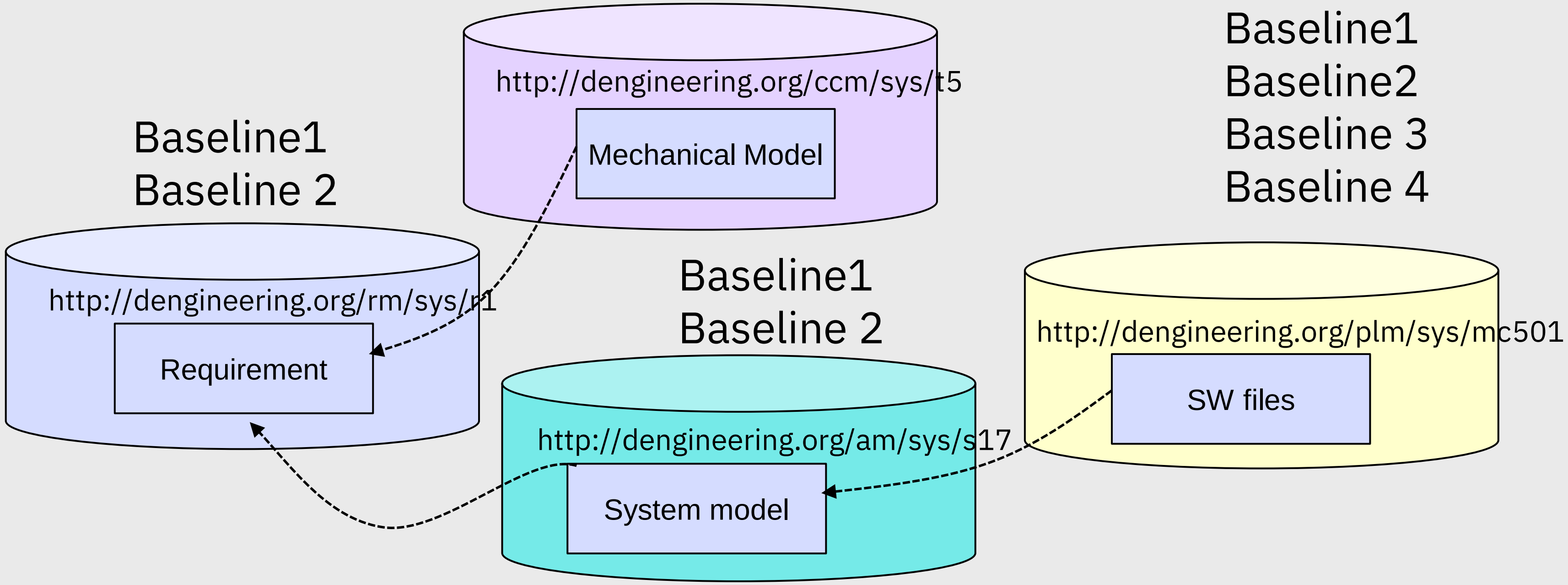
- Sreams (branches) – a mutable (concurrent) configuration
- Baselines – immutable configuration
- Component – a group of elements managed is one configuration item
- Element version – every element (e.g. a requirement) can have many versions



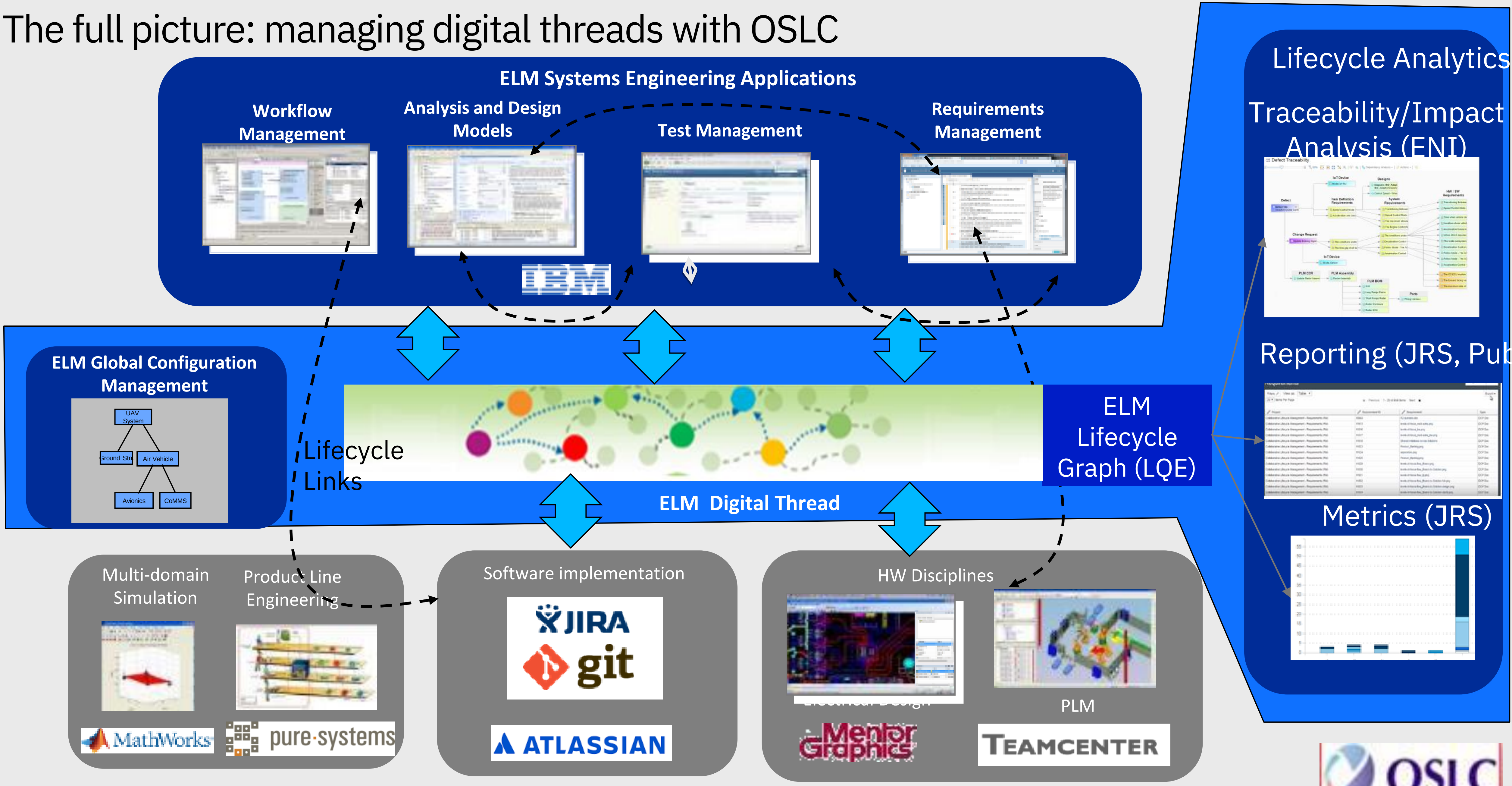
Global configuration management
Managing Data Across the entire digital thread

Data consistency

- Consistent evolution of data across engineering disciplines: common baselining
- Manage platform assets across variants and programs
- Reuse all engineering assets from the platform: requirements, design, implementation, test
- Manage changes across variants and programs
- Harvest innovation in programs for reuse across the product lines


http://dengineering.org/ccm/sys/t5
Mechanical Model
Baseline1
Baseline 2
http://dengineering.org/rm/sys/r1
Requirement
Baseline1
Baseline 2
http://dengineering.org/am/sys/s17
System model
Baseline1
Baseline2
Baseline 3
Baseline 4
http://dengineering.org/plm/sys/mc501
SW files

The full picture: managing digital threads with OSLC



- **Standards based**
- **End to End traceability with Link**
- **Central analytics based on knowledge**
- **Federated configuration**