Can we link data for System Engineering ?

……***O*** *ye**S** you'l**L** **C** !*

Eclipsecon unconference June 17th 2014 Toulouse

BIENVENUE and WELCOME !

Your team today

- Jean-Luc Johnson (Airbus Group Innovations)

- Gray Bachelor (IBM Rational CTO Office

- Samit Mehta (IBM Rational: ISV enablement and Ready for Rational)

- Harry Reeder (IBM Rational UK)

Welcome to the Crystal tool interoperability session at eclipsecon 2014

- Come and learn about the Crystal project under Artemis and see for yourself how tool interoperability can be achieved today !
- We'll give you an introduction and latest update on the project itself, then go on to walk through the steps to create a tool adapter using OSLC which is adopted under the Crystal Interoperability Specification V1.0
- You'll follow the story from the industrial use case and engineering method to the development based on eclipse Lyo to support Open Modelica data in a Change Impact Analysis scenario
- Come and join Airbus and IBM, who will provide you with a hands on workshop to build an OSLC adapter using the latest eclipse Lyo contribution.
- Our Unconference session allows you to get a good overview in the morning and then you can try for yourself in the afternoon
- Your joint OSLC team: Jean-Luc Johnson (Airbus Group Innovations), Samit Mehta (IBM) and Gray Bachelor (IBM)
- To make best use of the workshop attendees will need to download the requisite software before eclipsecon.
- June 17th 10-17:00 eclipsecon
- Contacts:
  - eclipsecon Gaël Blondelle gael.blondelle@eclipse.org
  - Airbus Jean-Luc.Johnson@eads.com
  - IBM gray_bachelor@uk.ibm.com
-

Crystal tool interoperability agenda

- 10am – Overview and introduction A202
  - Crystal overview and where OSLC fits ?
  - Walkthrough of examples of tool interoperability
  - Summary
- 12:30-14:00 Lunch
  - Poster session in parallel
- 14:00 – Hands-on “Build your own” A203
  - Deeper “Hands on” session with Lyo
- 17:30

Can we link data for System Engineering ?

……***O*** *ye****S*** *you'l****L*** ***C*** *!*

Introduction and Overview

Can we link data for System Engineering ?

……***O*** *ye**S** you'l**L** **C** !*

Overview of Crystal

The Crystal project

- Certain of today's examples relates to work being done under the European Community Artemis project known as Crystal

- Critical System Engineering Acceleration

Acknowledgement: The research leading to these results has received funding from the ARTEMIS Joint Undertaking under Grant Agreement N° 332830 and from specific national programs and / or funding authorities.

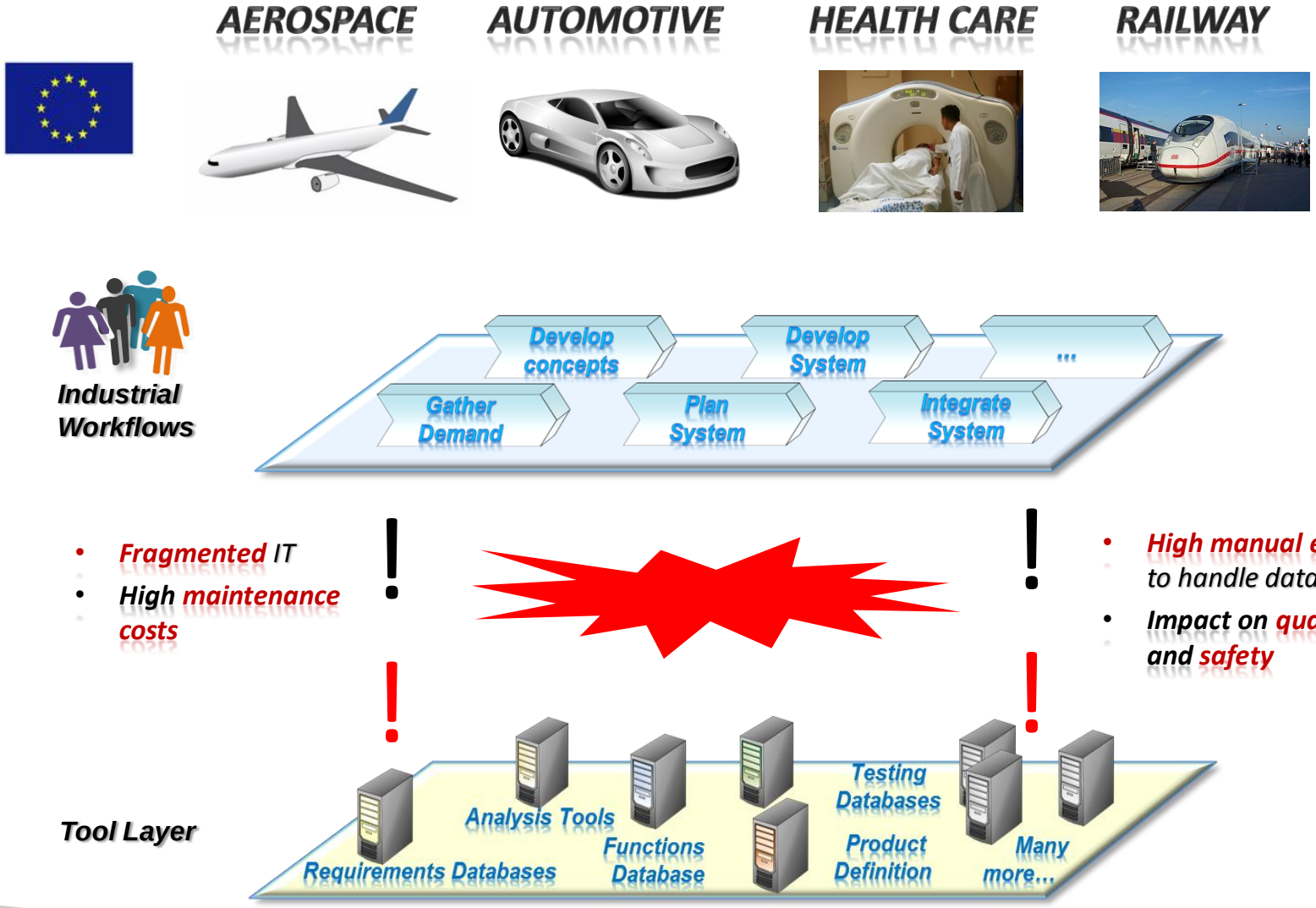
This workshop is a Crystal dissemination event

Why are we turning our attention to tool interoperability ?

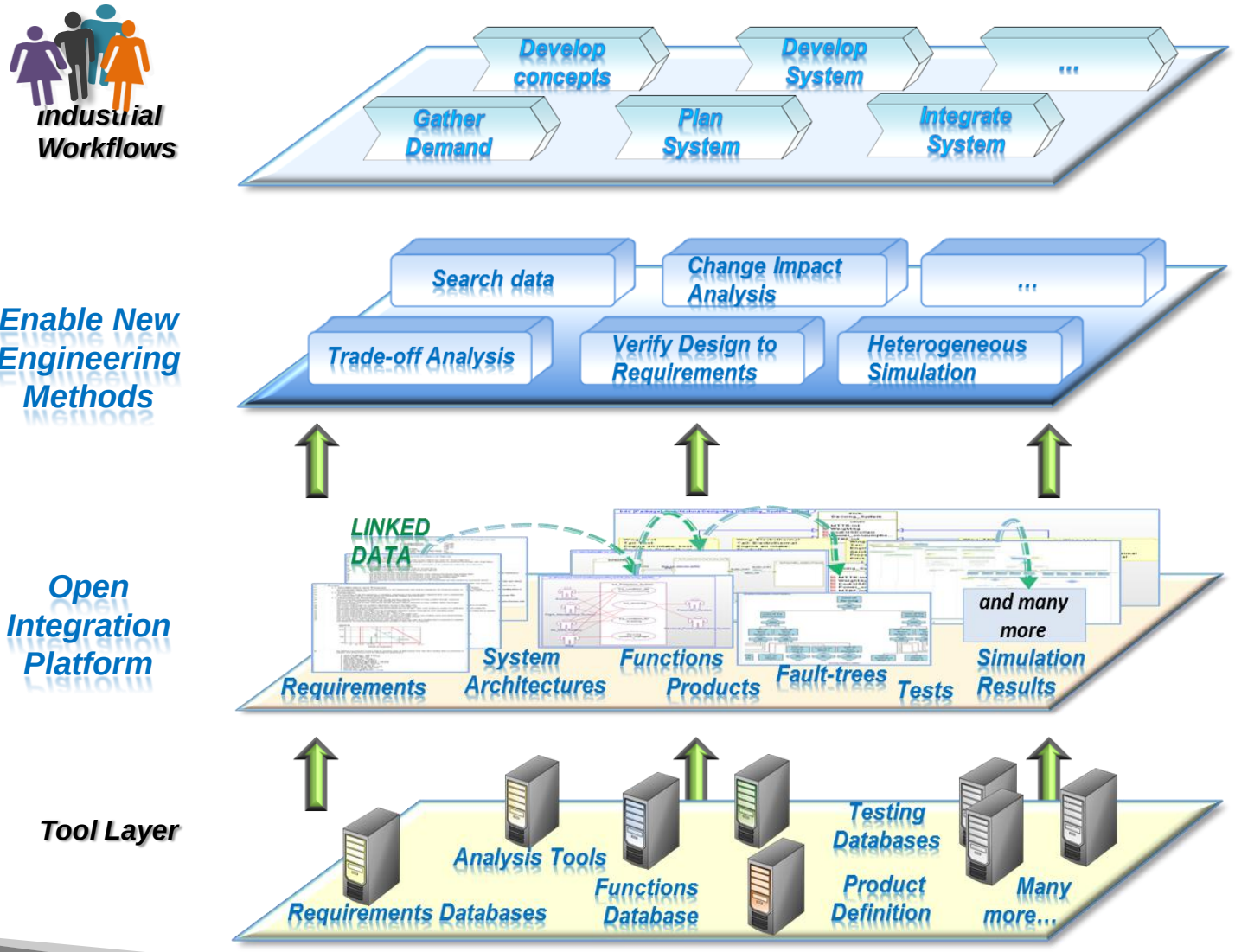
- Tools make us more productive !
- There has never been such a time of change of use of technologies, techniques and tools

- As new product technologies emerge we *must* use new tools
- As system and product complexity rises we *must* use more able tools
- As we work around the world we *must* have new collaborative tools
- And we *must* improve our existing tools

Today’s situation at industrial companies



The CRYSTAL Vision

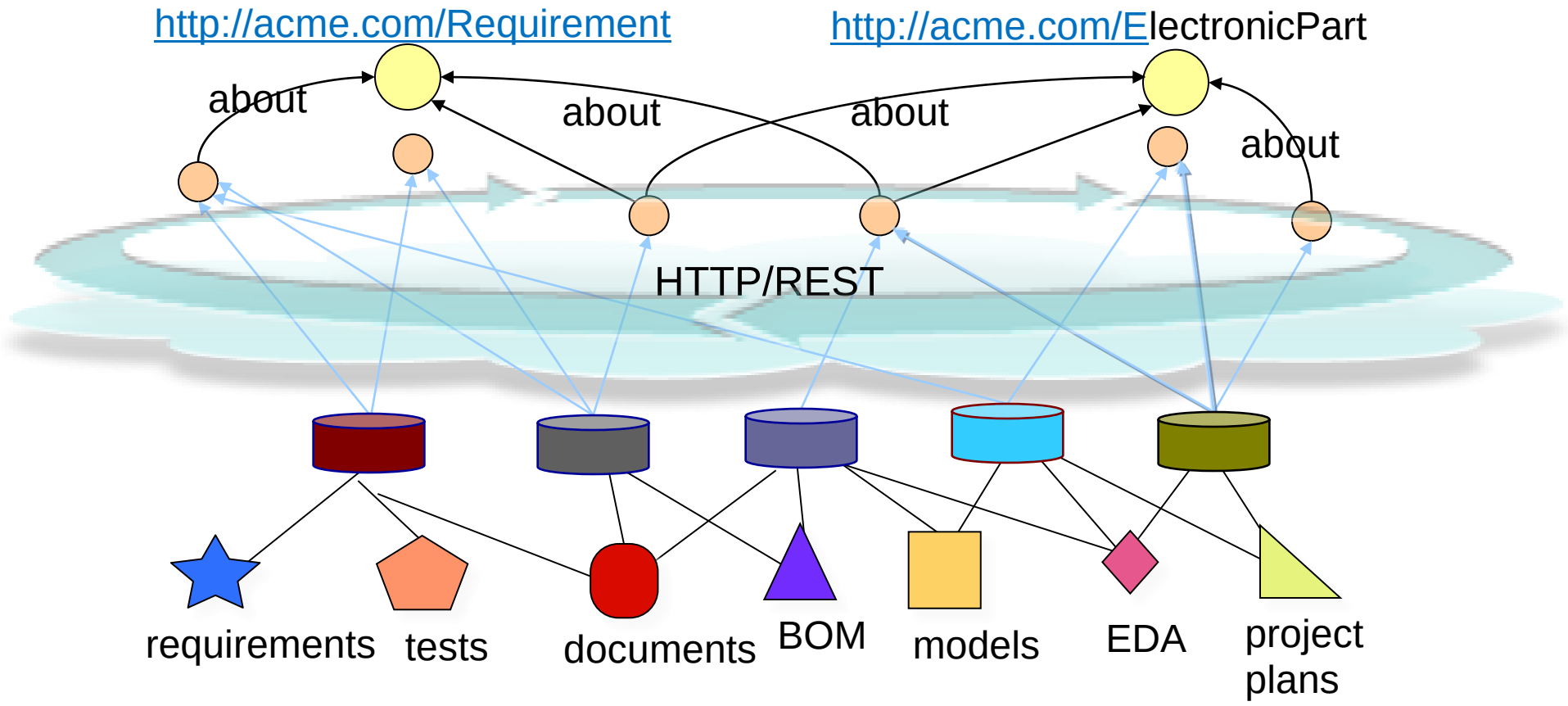


Users get better ways of working

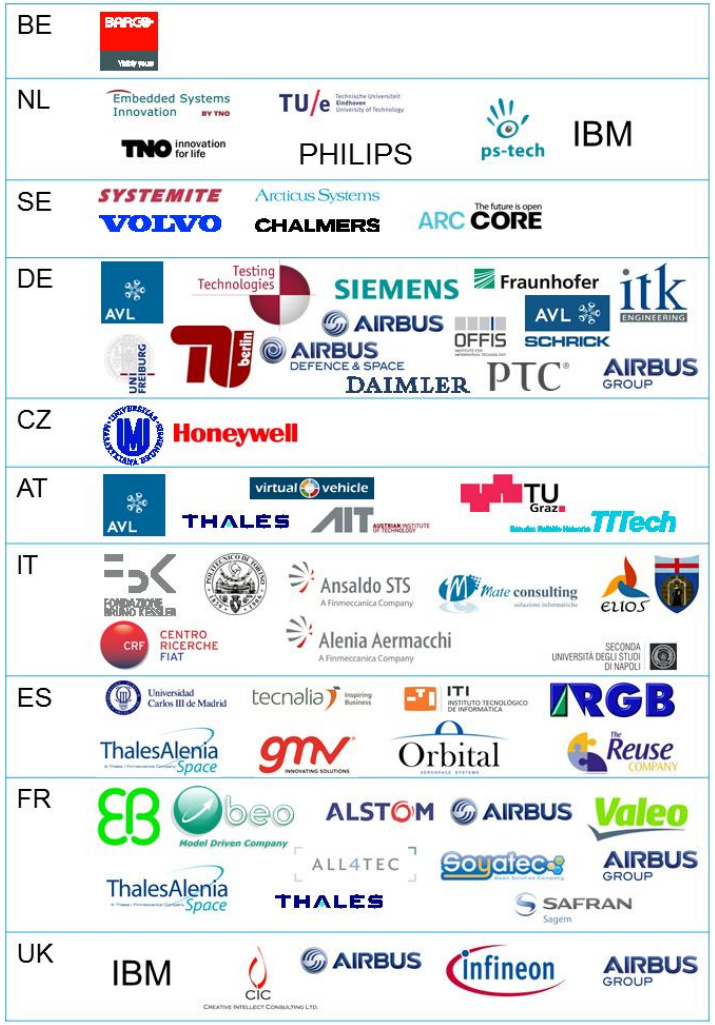
- ***Standardized Interoperability Specification***
- ***Connect tools to expose & link data***

Applying Linked Data Principles along the Product and System Lifecycle

Creating well defined systems of tools

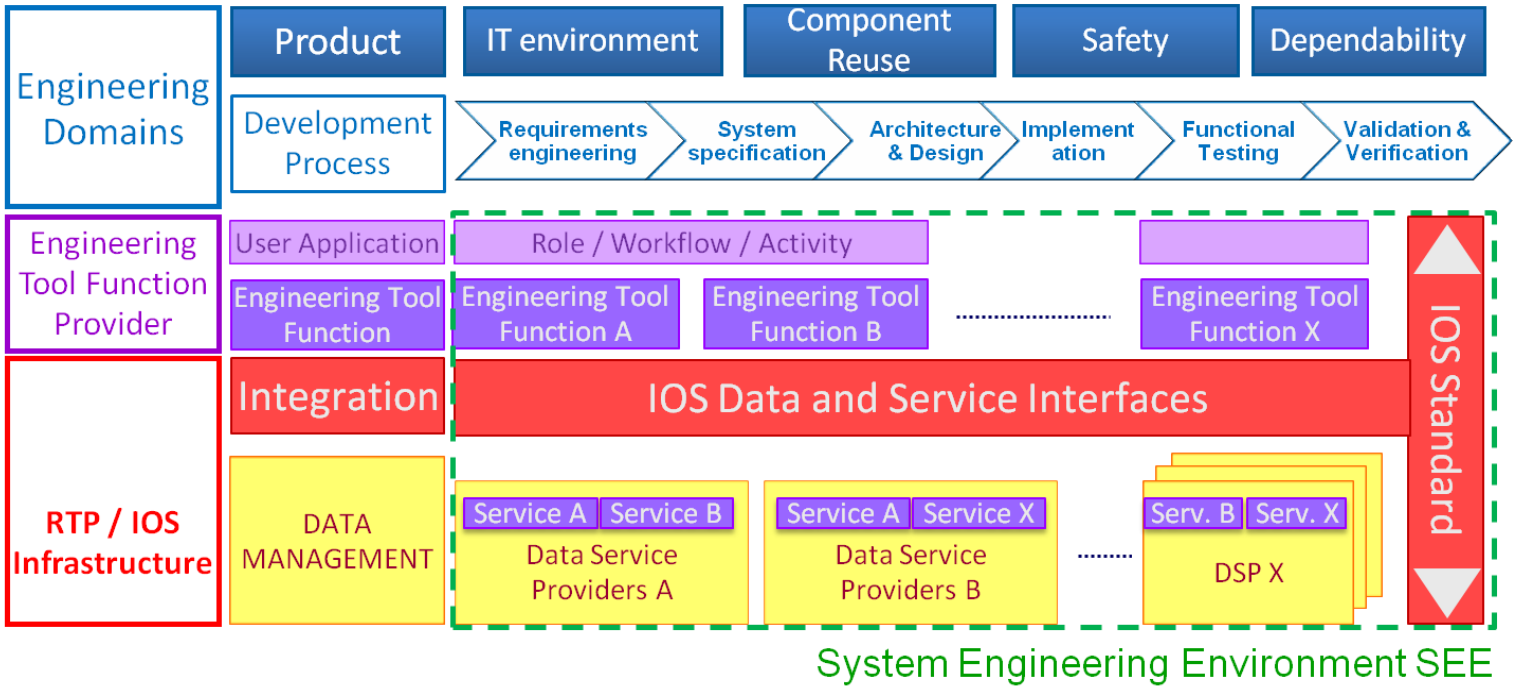


CRYSTAL has the critical mass to generate impact



- **68 partners** from **10 countries**
- **€82M** budget
- **European key players** from different industrial domains
- Large companies developing embedded systems act as **technology users** and case providers
- Large tool providers, SMEs and researchers as **technology providers**

Technical approach



The IOS exploits OSLC

- **<u>Standardize tool interaction, but not a tool's capabilities!</u>**
- Separate data from tool functions
- Apply **Interoperability Specification (IOS)** as the central standard
- Build on existing successful standards where appropriate

What's in for the you, us….for *the world* ?

- As a niche tool player you can embed your tool in a wider chain
  - Contribute to a bigger picture on "process" efficiency or contribution
- As a portfolio owner you can save the costs of so many interfaces
  - Leverage and promote the growing community towards your suite
- As a user you can contribute more to a project
  - Less effort to keep things on track, easier to show where your work fits
- As a manager you can speed up operations across functions
  - Faster to set up and keep running, work flows more easily across teams

Can we link data for System Engineering ?

……***O*** *ye****S*** *you'l****L*** ***C*** *!*

Overview of Crystal – an Engineering Method example

Change Impact Analysis

… a CRYSTAL Engineering Method example

Sales

"Can we make the product e.g. cheaper, better, faster……?"

"What is needed to make a product change?"

Manager

"What data needs to change?"

Engineer

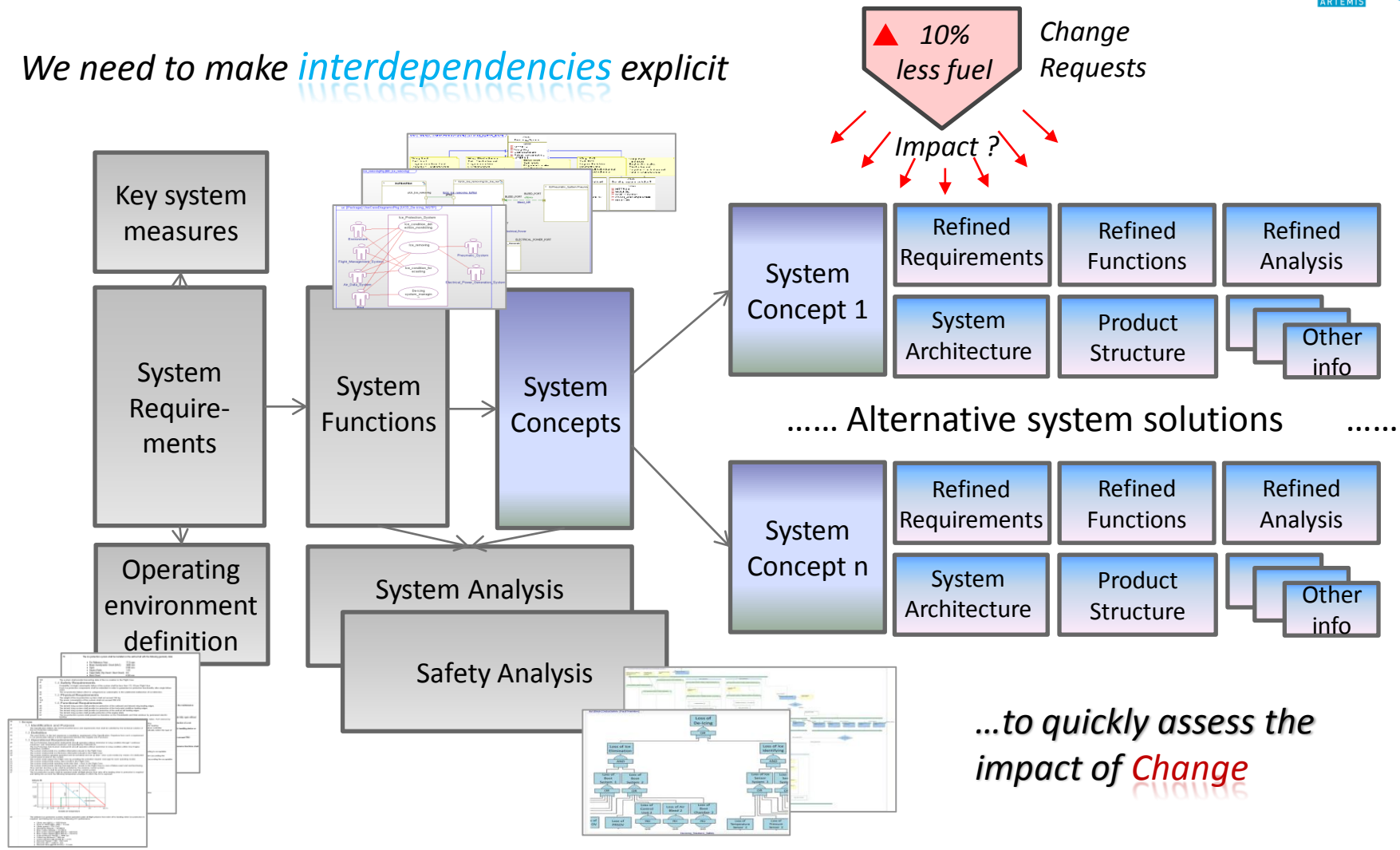
"Do I need to work on this ?"

Me or You

Change Impact Analysis

... a CRYSTAL Engineering Method example

We need to make interdependencies explicit



...to quickly assess the impact of Change

Challenges for our users like Pete and Susan …..struggling to keep up with the Change Request

Pete, Project Manager 

Susan, Systems Engineer 

1 Managing work

What's needed to make this change ?

2 Managing Requirements

How to avoid missing something ?

4 Analysing the Impact of Change

How to avoid too much change ?

What data needs to be changed ?

Which experts need to work on this ?

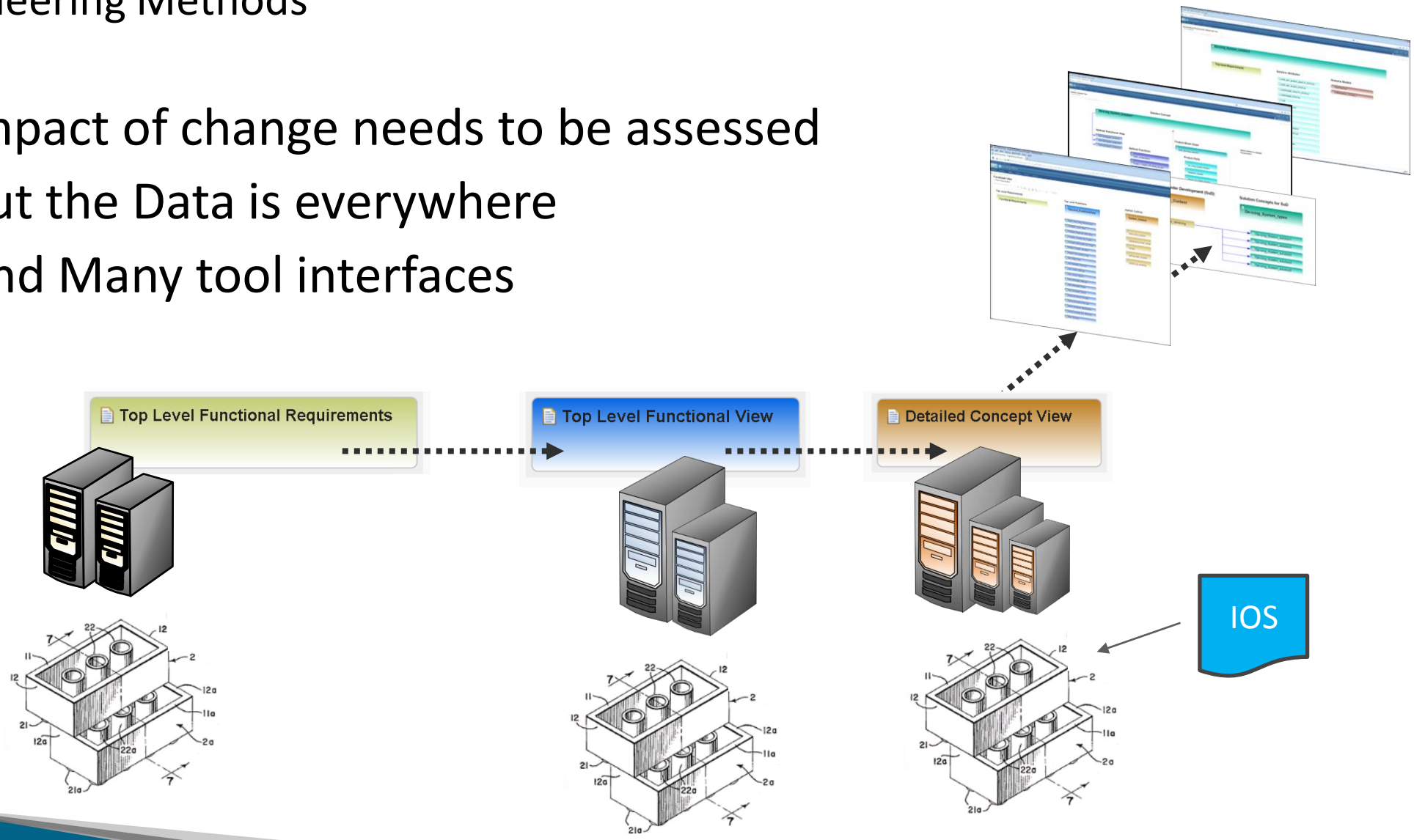
How to work this out quickly ?

3 Collaborating around lifecycle artefacts

How to access the data I need ?

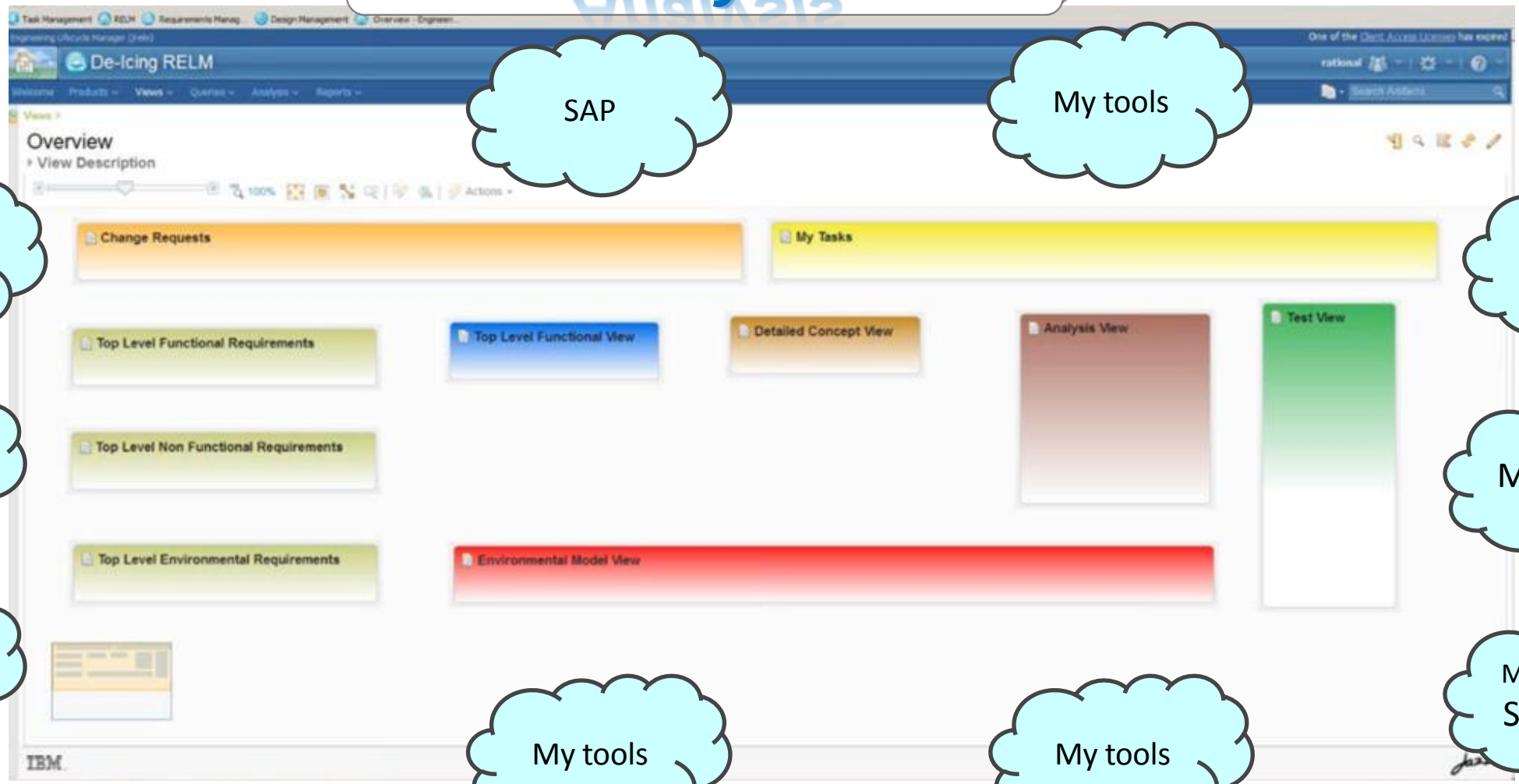
The Crystal Interoperability Spec (IOS) is a key enabler for Engineering Methods

- Impact of change needs to be assessed
- But the Data is everywhere
- And Many tool interfaces



Change Impact Analysis

... a CRYSTAL Engineering Method example



SAP

My tools

Bugzilla

Dassault CATIA

PTC Windchill

Modelica

Siemens Teamcenter

Mathworks Simulink

My tools

My tools

Our “users” Pete and Susan go home on time ……and happy !

Susan, Systems Engineer 

Pete, Project Manager 

1 Managing work

Now I know what needs to change ✓

2 Managing Requirements

Now we can avoid missing something ✓

4 Analysing the Impact of Change

We change just enough ✓

We know what data needs to change ✓

Now I know who needs to work on this ✓

I know what the cost and risk is ? ✓

3 Collaborating around lifecycle artefacts

I can access the data I need ✓

Can we link data for System Engineering ?

……***O*** *ye****S*** *you'l****L*** ***C*** *!*

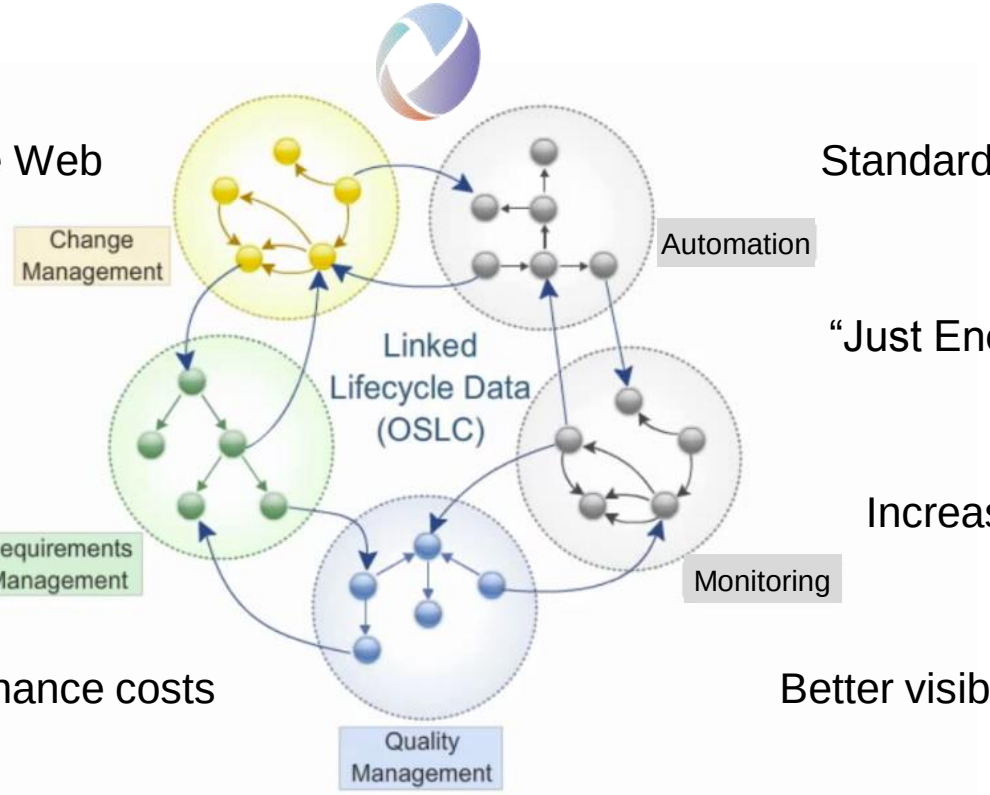
Overview of OSLC

What is OSLC ?
OSLC is cunningly simple !

Users can work seamlessly across their tools
(complex and fragile synchronization schemes not required)

Architecture of the Web

Standard Interfaces



Linked Data

"Just Enough" integration

Increased reuse

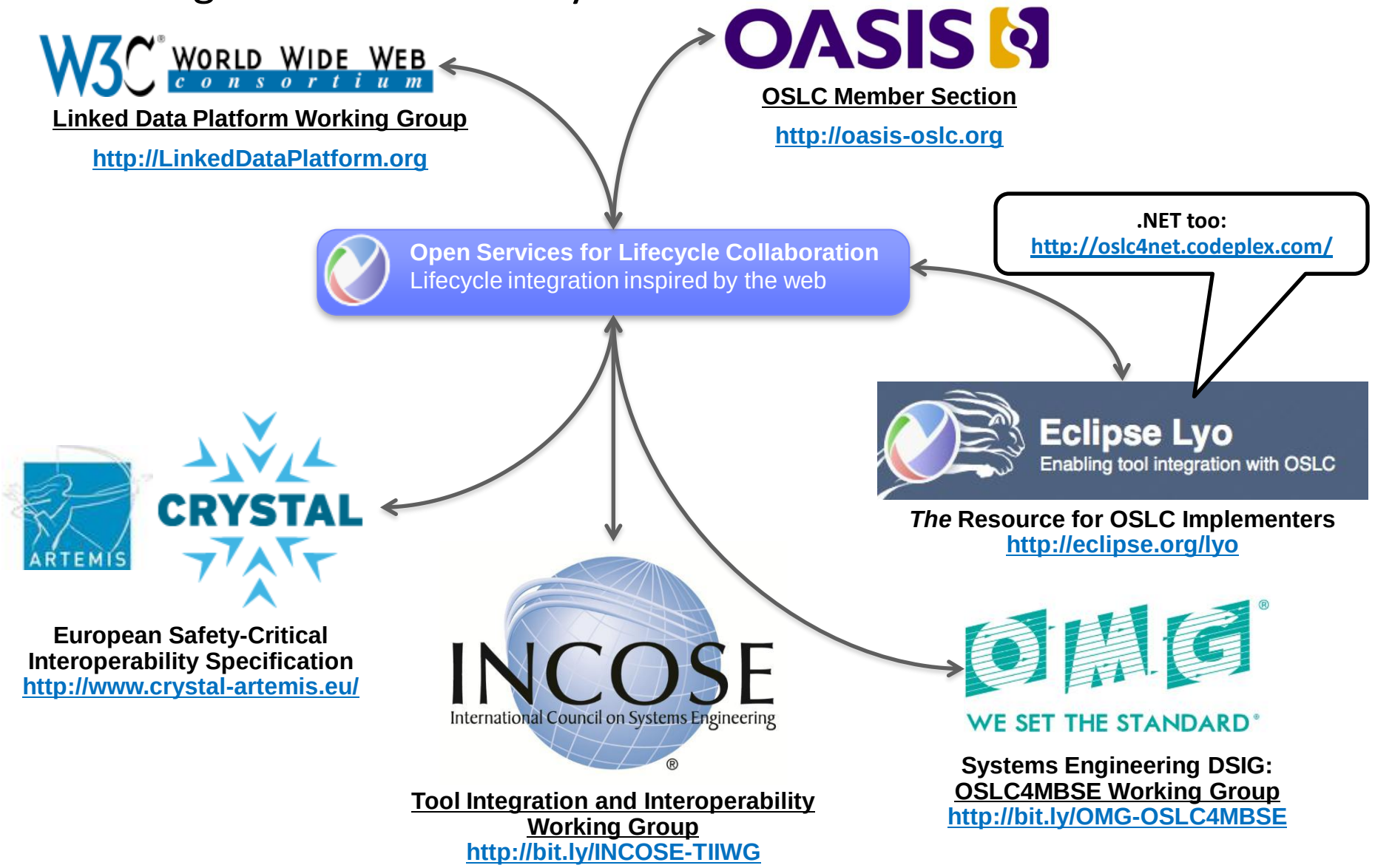
Increased traceability

Decreased maintenance costs

Better visibility

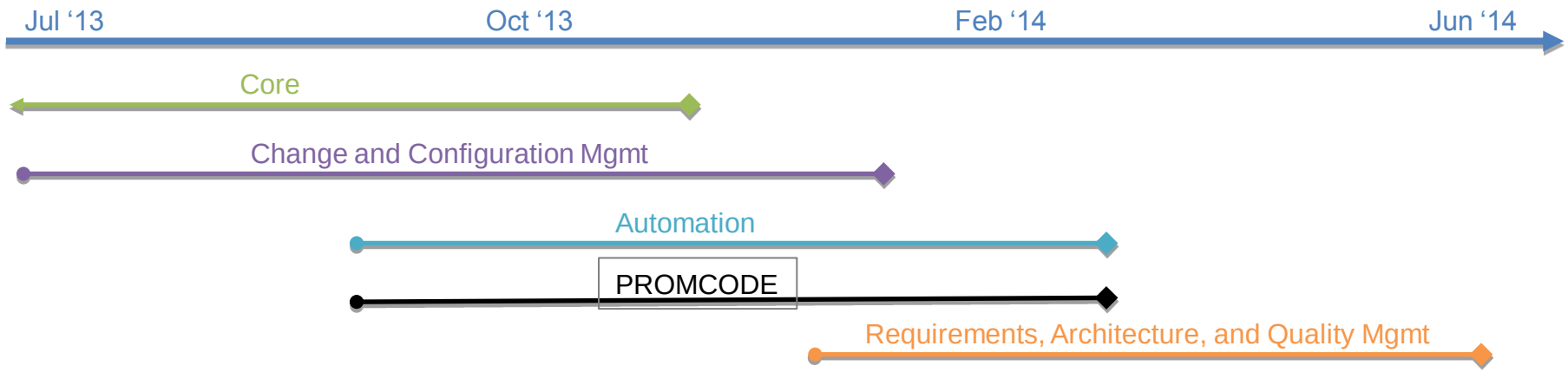
OSLC is an open and scalable approach to lifecycle integration.
It simplifies key integration scenarios across heterogeneous tools

How mature is OSLC today ?
OSLC: Collaborating across the industry



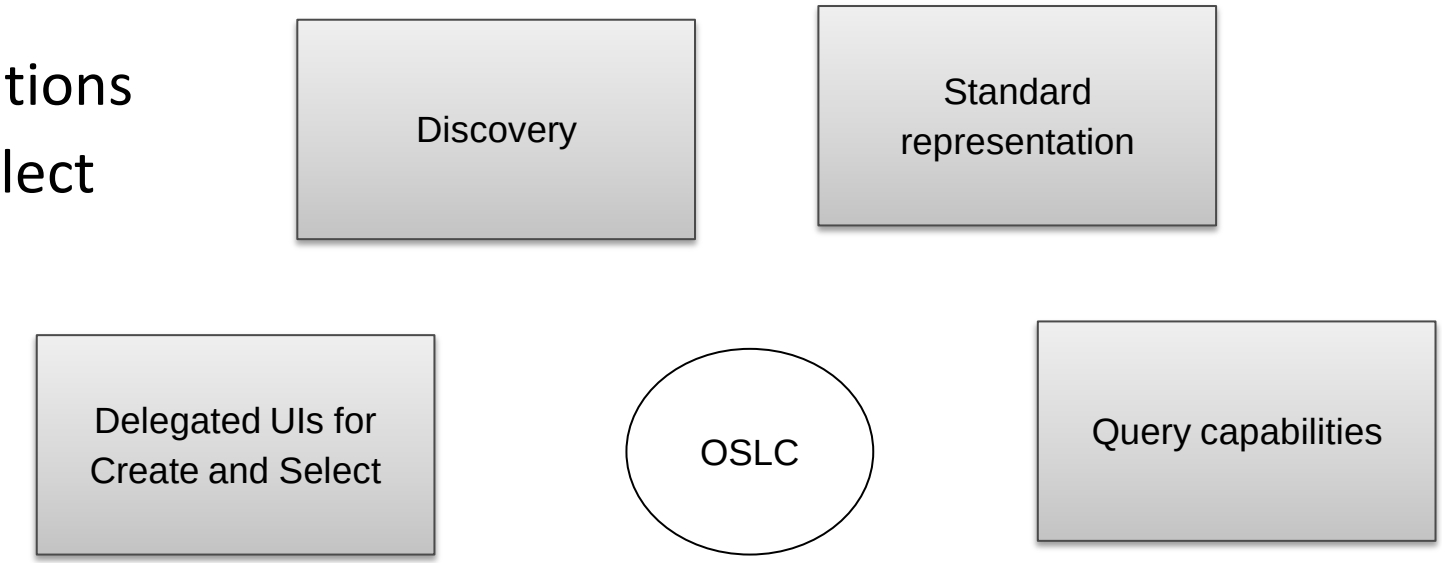
How mature is OSLC today ?
OSLC moves to OASIS

	Prioritization & Preparation with MS members	TC Charter Creation & TC Co-submitter Recruiting	TC Charter Submission & OASIS Process	OASIS Approval & 1st TC meeting
Core TC	✓ Done	✓ Done	✓ Done	✓ Done
OSLC CCM TC	✓ Done	✓ Done	✓ Done	➢ Jan 2014
OSLC Automation TC	✓ Done	✓ Done	➢ Dec-Jan 2014	Mar 2014
OSLC PROMCODE TC	✓ Done	✓ Done	➢ Dec-Jan 2014	Mar 2014
OSLC RAQ Mgmt TC	➢ Dec 2013 – Feb 2014	Mar 2014	Apr-May 2014	Jun 2014

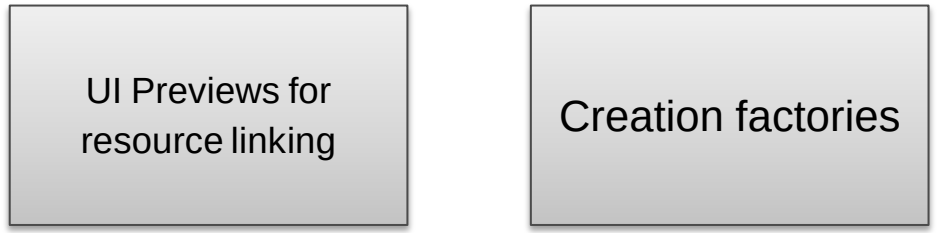


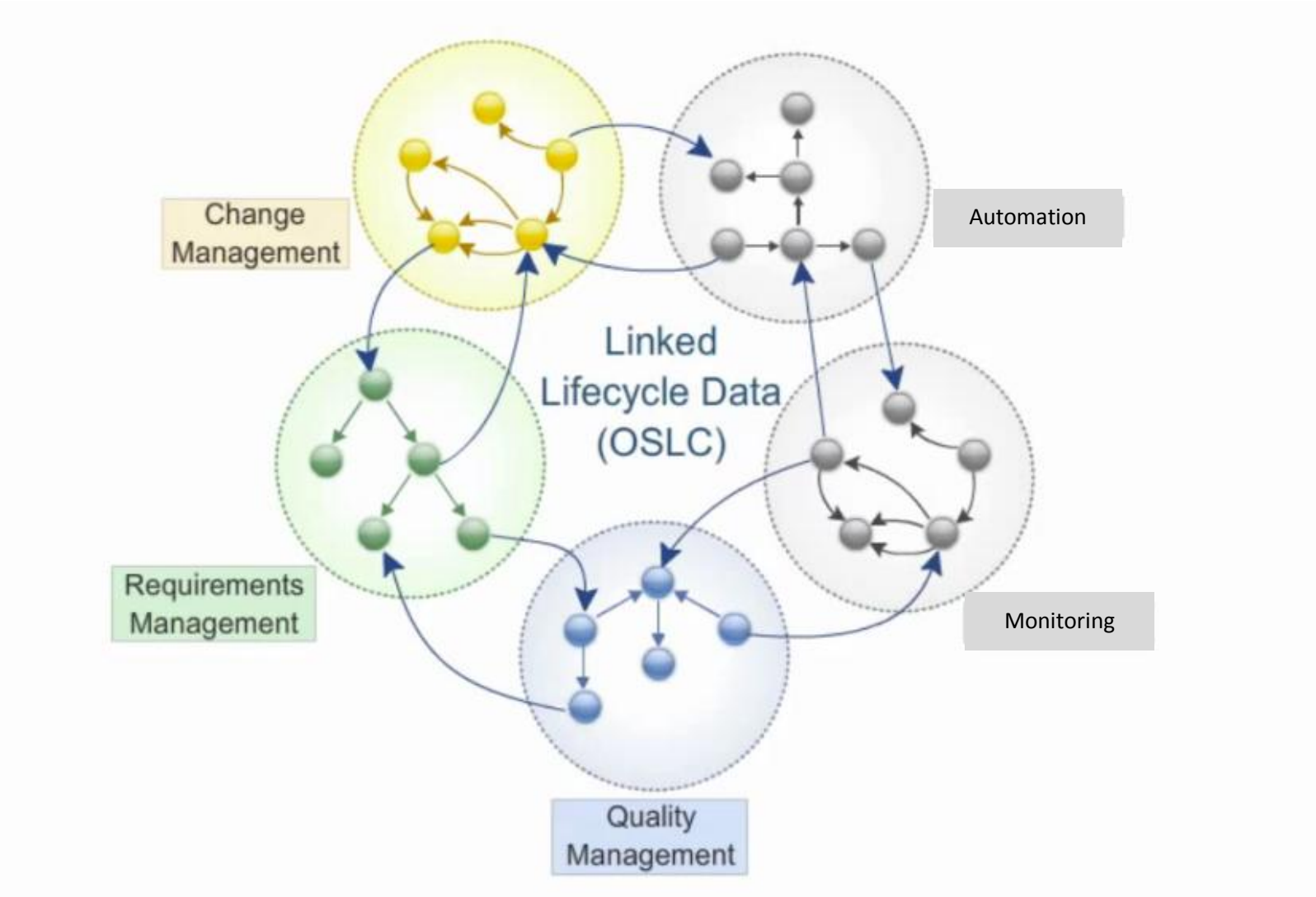
OSLC basic capabilities

- Discovery
- HTTP C.R.U.D for resources
- Standard Resource Representations
- Delegated UI for Create and select
- Query capabilities
- UI Previews for Resource links

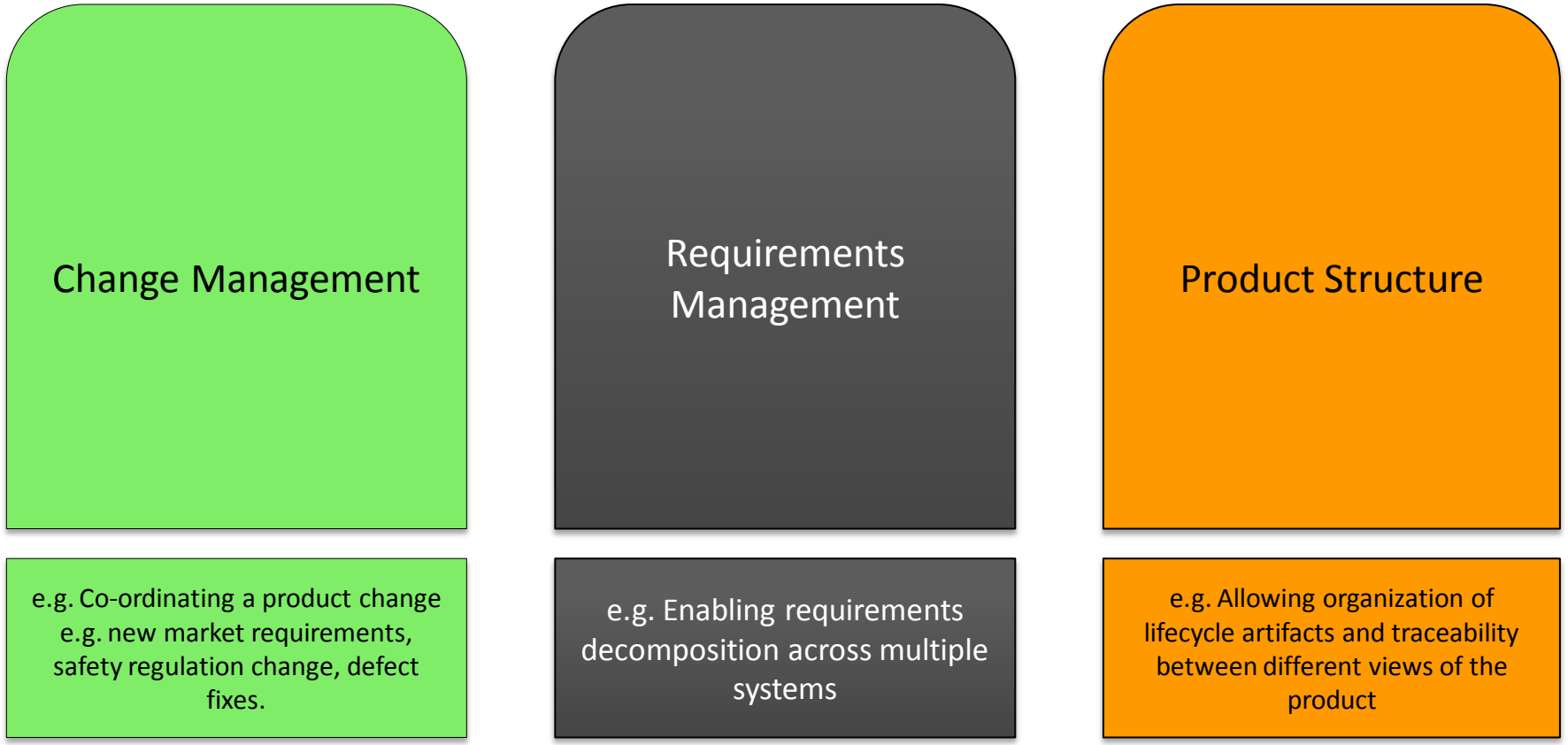


- Now with Tracked Resource Sets




Change Management
Automation
Linked Lifecycle Data (OSLC)
Requirements Management
Monitoring
Quality Management

Prioritization from customer input – Key ALM/PLM integration pain points



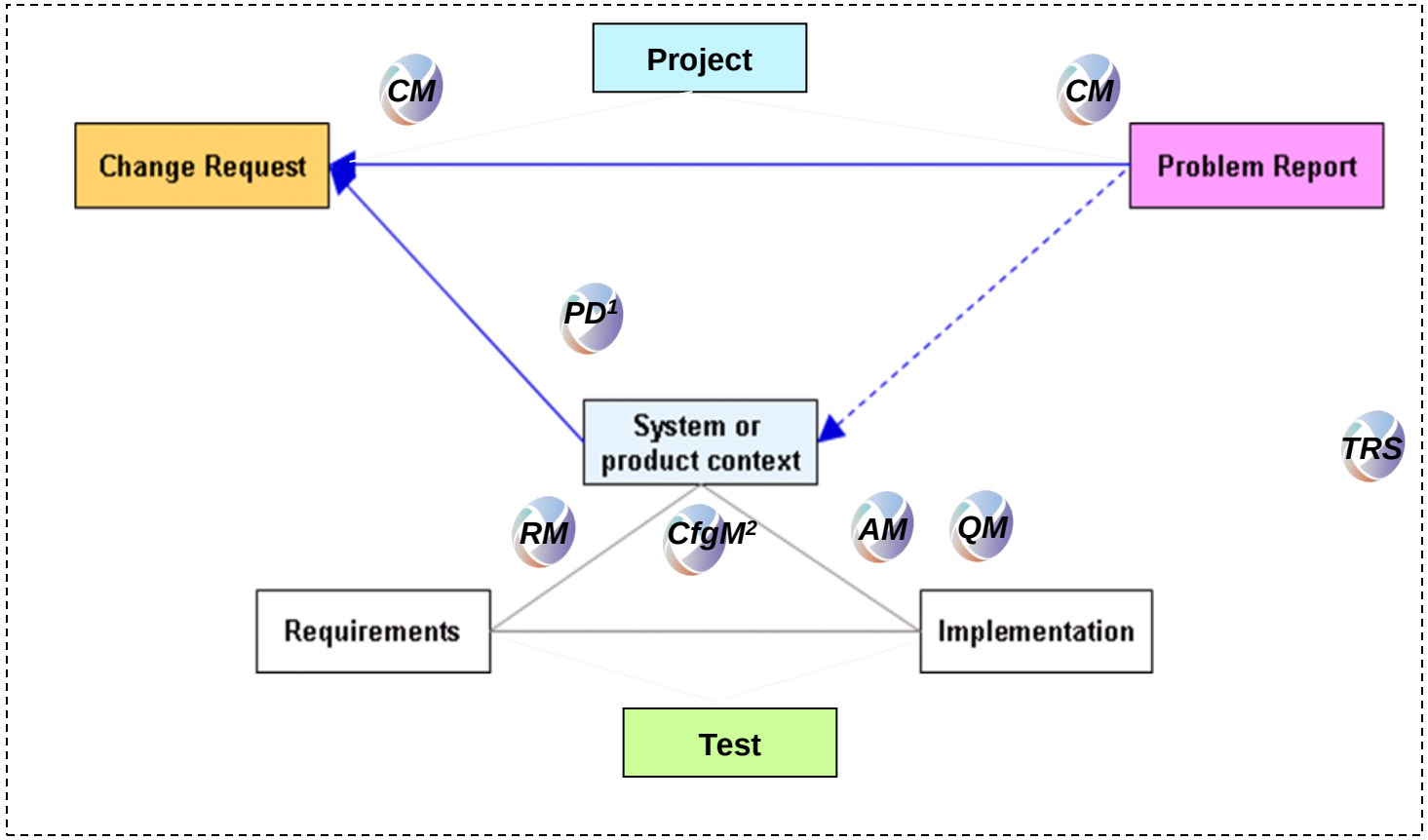
OPEN AND EXTENSIBLE !

OSLC provides support for tool interoperability and is being applied for ALM-PLM workflows and collaboration support

Activities

Configurations

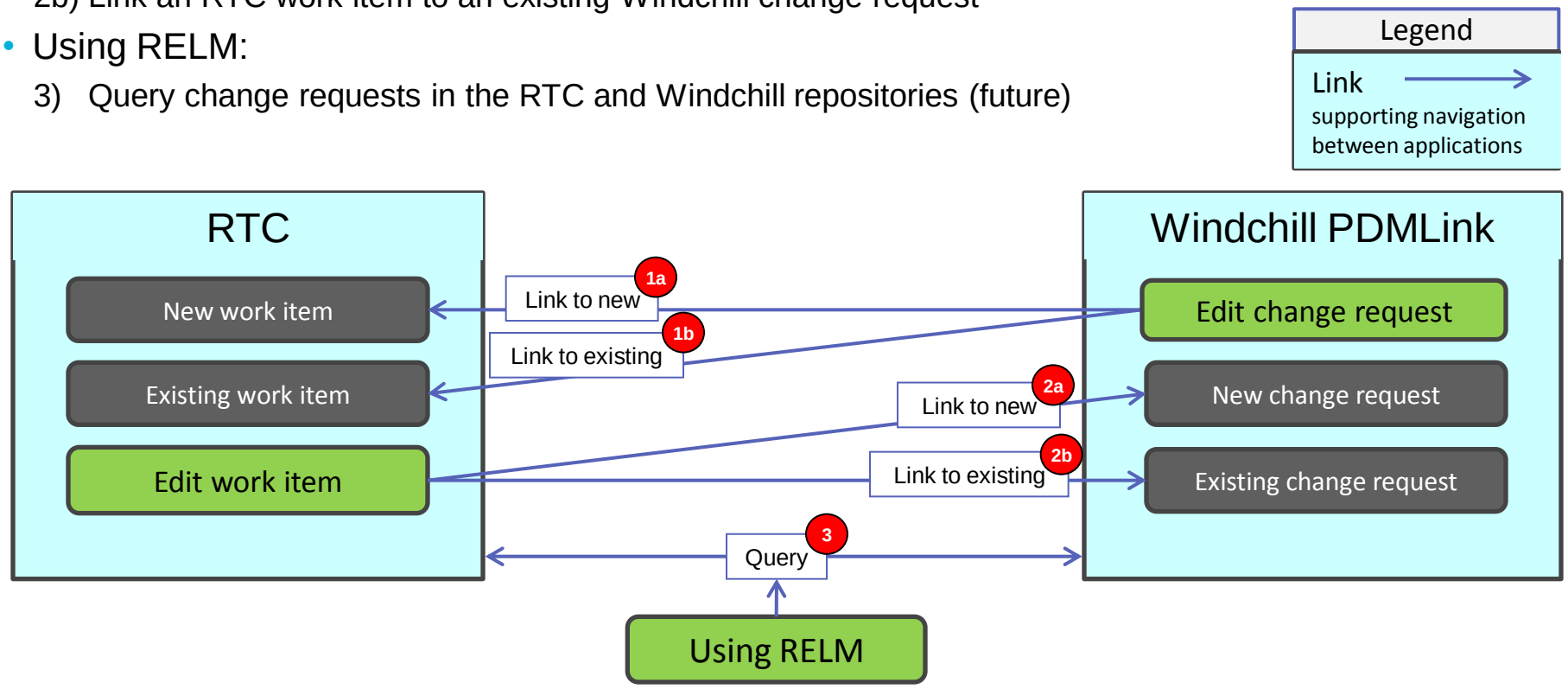
Fulfilment of Requirements



Note 1: Product definition spec in draft
Note 2: Configuration Management spec is in formulation

Change Management Use-Cases to support scenarios

- Using Windchill PDMLink:
  1a) Link a Windchill change request to a new RTC work item
  1b) Link a Windchill change request to an existing RTC work item
- Using RTC:
  2a) Link an RTC work item to a new Windchill change request
  2b) Link an RTC work item to an existing Windchill change request
- Using RELM:
  3) Query change requests in the RTC and Windchill repositories (future)



How is OSLC doing today ?
Implementation maturity examples

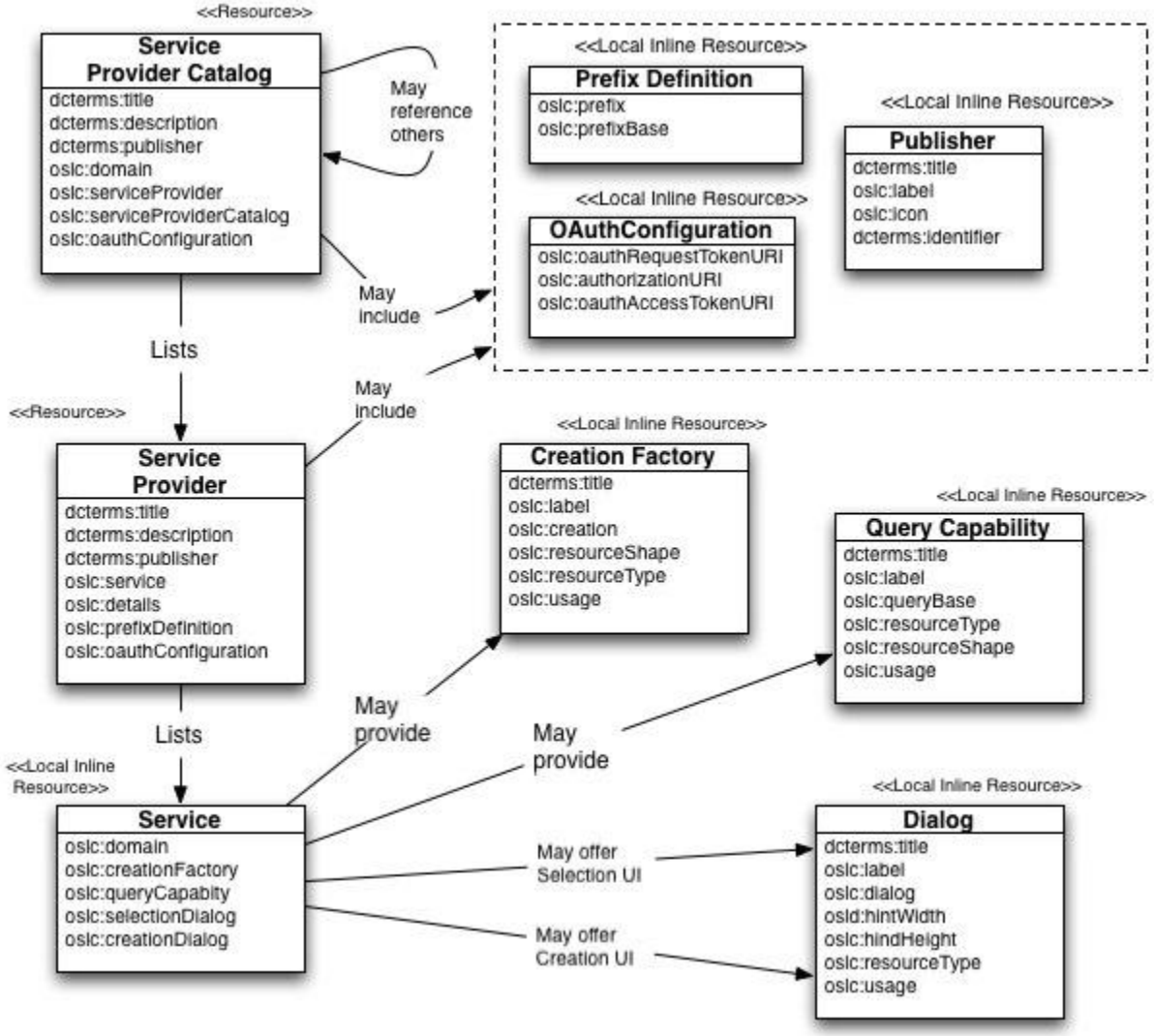
- Multiple domains across multiple IBM product families
  - The whole Rational Jazz family
  - Classic Rational family
  - Across the Tivoli family e.g. Smart Cloud Monitoring
- 3rd party support
  - Mentor Graphics Capital and Context
  - National Instruments
  - Tasktop
  - SAP
  - JIRA
  - PTC Windchill
- Prototypes and demonstrator by customers, ISVs and IBM
  - SAP, Siemens Teamcenter
  - Dassault Enovia
  - Sharepoint
  - Open Modelica
- Other interesting examples e.g. http://www.engisis.com/using-oslc-step-plm-data-integration/ OSLC and STEP 10303

Can we link data for System Engineering ?

……***O*** *ye**S** you'll **L** **C** !*

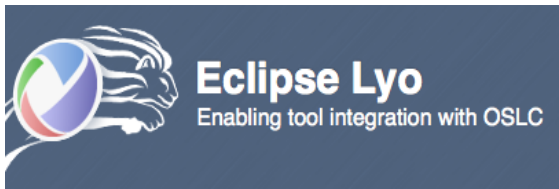
Closer look at implementing OSLC services

Accessing OSLC services



Eclipse Lyo: http://eclipse.org/lyo

- Eclipse project goal: Provide tools to enable adoption of OSLC specifications.
- Content includes
  - SDK - Code libraries (Java, Perl, others under development)
    - Give developers tools to ease OSLC implementations
  - Reference implementations of specifications
    - Provide a starting point for new integrations
  - Test suites and test reporting.
    - Accelerate and assess development
    - Regression testing
  - Samples, tutorials and documentation
    - Working samples of OSLC integrations with Bugzilla, Excel, Jazz tools and more.



What the latest on Lyo?

Main Features released as version 2.1.0

- OSLC4J – Java SDK for OSLC implementations (1.0, 1.1, 2.0.0)
  - Annotations to add OSLC attributes and meta-data to Java objects representing OSLC resources
  - Serialize/deserialize Java objects as OSLC resources
- Tracked Resource Set (TRS) SDK and reference implementation (2.0.0)
  - SDK and reference implementation for building OSLC
  - Tracked Resource Set implementations
- OSLC assessment Test Suite (1.0, 1.1, 2.0.0)
  - Tests cover OSLC Core spec + Change Management,
  - Quality Management, Asset Management, Requirements Management, and Automation specifications
  - (New) Coverage for TRS 2.0 specification in 2.0.0
- OAuth provider library (1.1, 2.0.0)
  - Core, persistence and webapp libraries to ease OAuth provider implementation

Summary of using Eclipse Lyo to make the Bugzilla adapter

- Publish the Service Provider Catalog
- Provide Delegated User Interface Dialogs:
  - Creation Dialog
  - Selection Dialog
- Provide User Interface Preview HTML Representation
- Provide resource creation and update services
- Provide resource query service

Implementing Service Provider Catalog and Service Provider

- The HTTP GET service for Service Provider Catalog

```
@GET
@Path("{serviceProviderCatalogId}") // Required to distinguish from array result.  But, ignored.
@Produces({OslcMediaType.APPLICATION_RDF_XML, OslcMediaType.APPLICATION_XML, OslcMediaType.APPLICATION_JSON})
public ServiceProviderCatalog getServiceProviderCatalog()
{
    ServiceProviderCatalog catalog =  ServiceProviderCatalogSingleton.getServiceProviderCatalog(httpServletRequest);
```

- The HTTP GET service for Service Provider

```
@GET
@Path("{serviceProviderId}")
@Produces({OslcMediaType.APPLICATION_RDF_XML, OslcMediaType.APPLICATION_XML, OslcMediaType.APPLICATION_JSON})
public ServiceProvider getServiceProvider(@PathParam("serviceProviderId") final String serviceProviderId)
{
    httpServletResponse.addHeader("Oslc-Core-Version","2.0");
    return ServiceProviderCatalogSingleton.getServiceProvider(httpServletRequest, serviceProviderId);
}
```

Implementing Delegated User Interface Dialogs

- Dialog Service

```
@GET
@Path("selector")
@Consumes({ MediaType.TEXT_HTML, MediaType.WILDCARD })
public void changeRequestSelector(@QueryParam("terms")     final String terms,
                                  @PathParam("productId")  final String productId) throws ServletException, IOException
{
    int productIdNum = Integer.parseInt(productId);
    httpServletRequest.setAttribute("productId", productIdNum);
    httpServletRequest.setAttribute("bugzillaUri", BugzillaManager.getBugzillaUri());
    httpServletRequest.setAttribute("selectionUri",uriInfo.getAbsolutePath().toString());

    if (terms != null ) {
        httpServletRequest.setAttribute("terms", terms);
        sendFilteredBugsReponse(httpServletRequest, productId, terms);

    } else {
        try {
            RequestDispatcher rd = httpServletRequest.getRequestDispatcher("/cm/changerequest_selector.jsp");
            rd.forward(httpServletRequest, httpServletResponse);
```

Implementing Delegated User Interface Dialogs

- Dialog Service Location

```
@OslcDialogs(
{
    @OslcDialog
    (
        title = "Change Request Selection Dialog",
        label = "Change Request Selection Dialog",
        uri = "/{productId}/changeRequests/selector",
        hintWidth = "525px",
        hintHeight = "325px",
        resourceTypes = {Constants.TYPE_CHANGE_REQUEST},
        usages = {OslcConstants.OSLC_USAGE_DEFAULT}
    )

})
```

Can we link data for System Engineering ?

……***O*** *ye**S** you'l**L** **C** !*

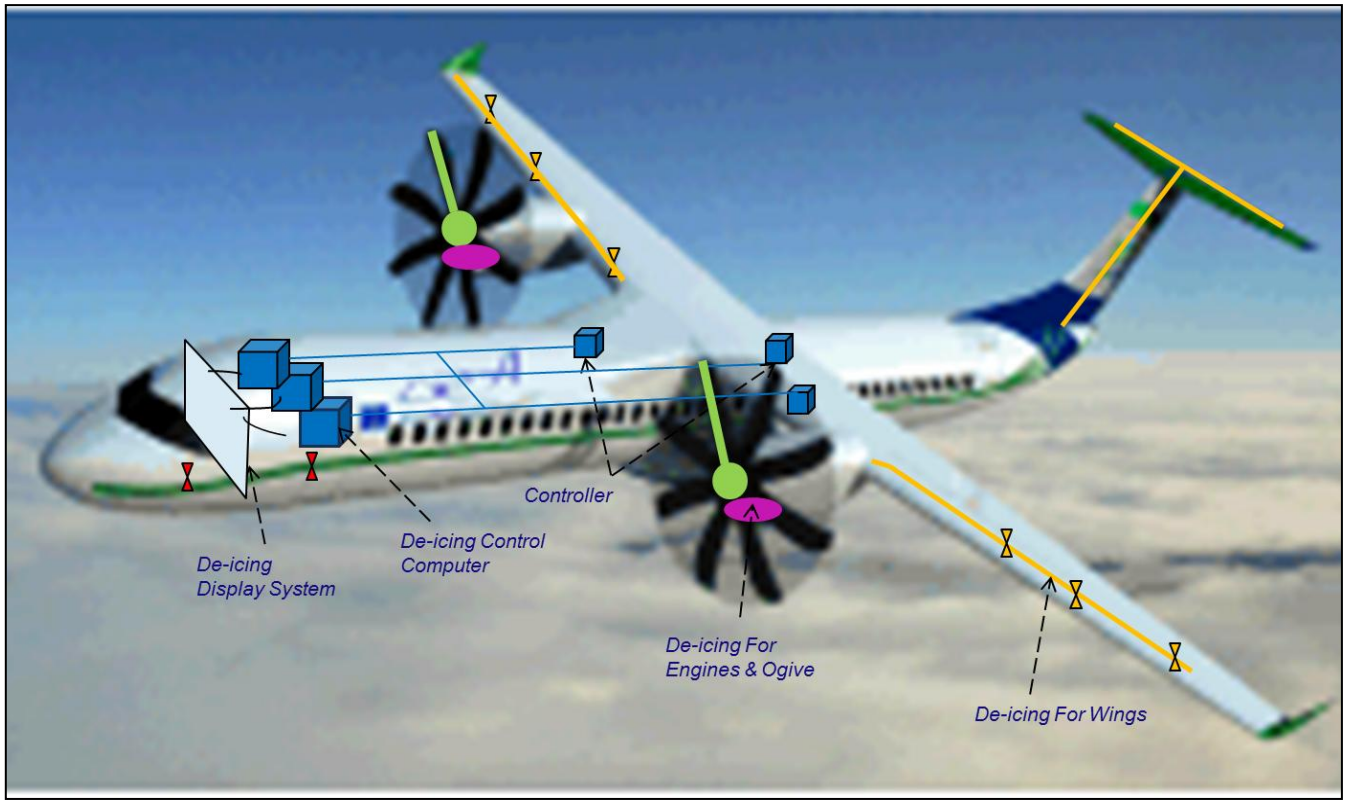
Example for Open Modelica

CRYSTAL OSLC adapter & Open Modelica

- Overview and introduction
  - What is the de-icing system?
  - Where Open Modelica fits in?
  - The Open Modelica OSLC Adapter

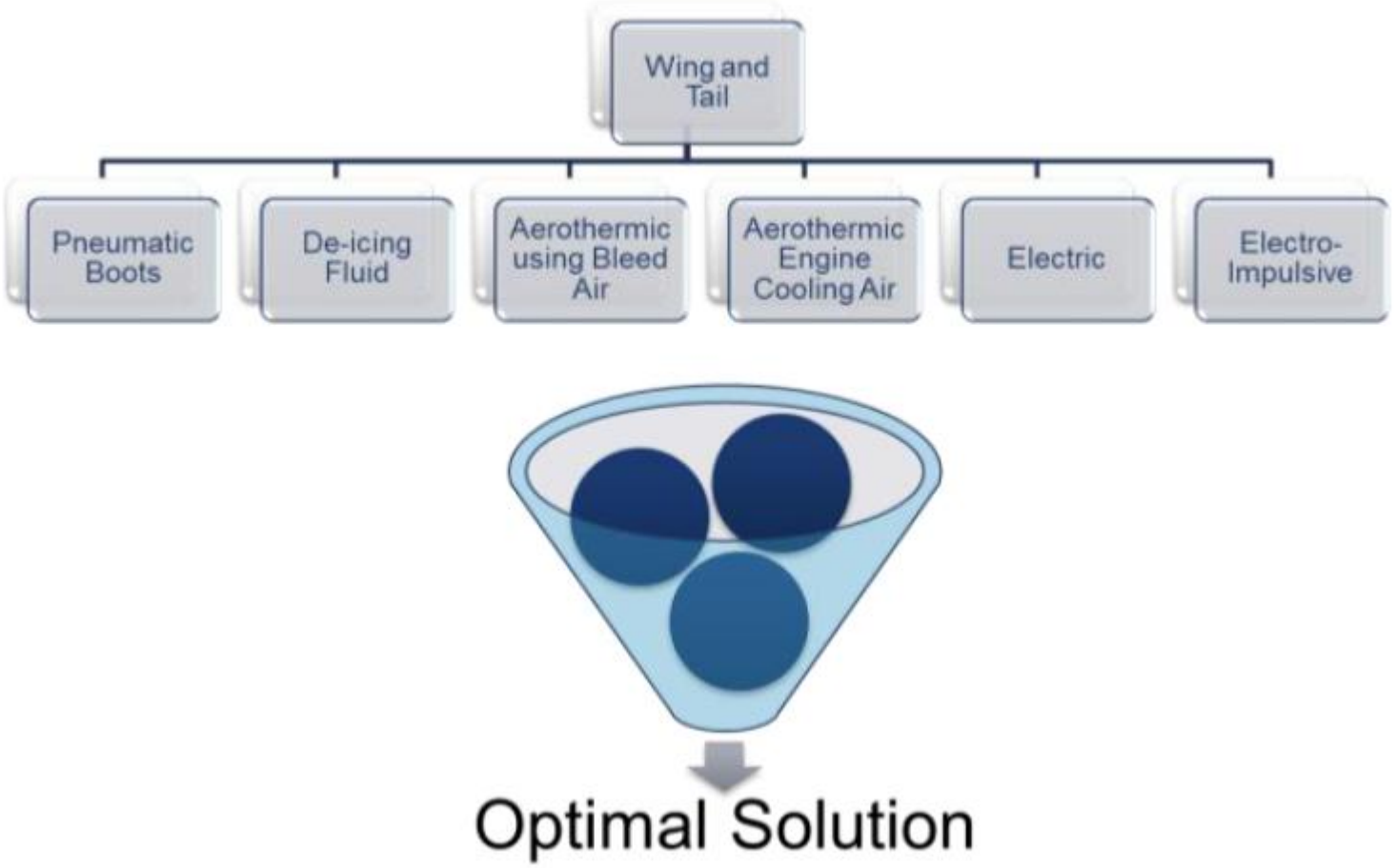
Regional aircraft de-icing system

- Purpose:
  - Prevent the ice creation on
    - Sensors
    - Wings
    - Engines
- Objective:
  - Design the system
    - At minimum cost,
    - weight and
    - Power consumption



De-icing system alternatives

- Pneumatic boots
  - Permanent inflation and
  - Deflation
- Bleed air
  - To melt ice
- Electrical heating



Open modelica Simulation tool

- Open modelica (Simulation tool)
  - To asset the physical behavior of the system
- Contribute to few engineering methods
  - Verify the design against the requirements
  - Change impact analysis
  - Trade-off analysis

Can we link data for System Engineering ?

……***O*** *ye****S*** *you'l****L*** ***C*** *!*

Morning Q&A

Crystal tool interoperability agenda

- 10am – Overview and introduction A202
  - Crystal overview and where OSLC fits ?
  - Walkthrough of examples of tool interoperability
  - Summary
- 12:30-14:00 Lunch
  - Poster session in parallel
- 14:00 – Hands-on “Build your own” A203
  - Deeper “Hands on” session with Lyo to continue
- 17:30

What will you do this afternoon ?

- Preview: Java Swing GUI (OSLC consumer)

- Workshop1: Resource class overview – Rootservices document parser
- Workshop2: Discover Service providers and list services available
- Workshop 3: Consuming services – query capability and creation factory
  Get resource, update and create

- Demo: Java Swing GUI (OSLC consumer)
- Demo: Bugzilla OSLC provider

Labs

- See separate material

Can we link data for System Engineering ?

……***O*** *ye**S** you'l**L** **C** !*

Build you own !

Crystal tool interoperability agenda

- 10am – Overview and introduction A202
  - Crystal overview and where OSLC fits ?
  - Walkthrough of examples of tool interoperability
  - Summary
- 12:30-14:00 Lunch
  - Poster session in parallel
- 14:00 – Hands-on “Build your own” A203
  - Deeper “Hands on” session with Lyo to continue
- 17:30

What will you do this afternoon ?

- Preview: Java Swing GUI (OSLC consumer)

- Workshop1: Resource class overview – Rootservices document parser
- Workshop2: Discover Service providers and list services available
- Workshop 3: Consuming services – query capability and creation factory
  Get resource, update and create

- Demo: Java Swing GUI (OSLC consumer)
- Demo: Bugzilla OSLC provider

Can we link data for System Engineering ?

……***O*** *ye**S** you'l**L** **C** !*

Your feedback of the lab, please….

Can we link data for System Engineering ?

……***O*** *ye****S*** *you'l****L*** ***C*** *!*

Afternoon Q&A

Can we link data for System Engineering ?

……***O*** *ye****S*** *you'l****L*** ***C*** *!*

Close