Erik Herzog, Johan Tingström, Åsa Nordling Larsson

Genesis – an Architectural Pattern for Federated PLM

Saab Aeronautics – the old game

- One customer
- One operations approach – national defence
- One project at a time
- Long development times
- Predictability: Sweden and Saab

1950

1970

1990

The new Game

- Multiple parallel projects
- International operations and interoperability
- Exports
- International collaboration
  - Multi-site Development & Production
- More stringent international regulations
- Speed!
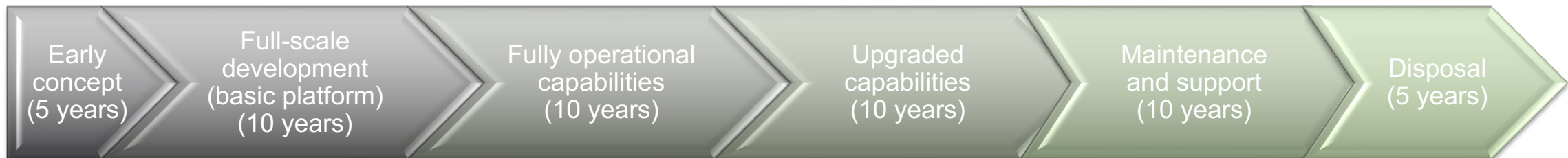 - Product development
  - Enabling systems
- Unpredictable future

2000   2020

System characteristics

- Long lifecycles Safety critical systems
- Continuous development
- Development system life is substantially **shorter** than System life
- Historical observation
  - Need to replace development system **trice** over the life of the system

Early concept (5 years)	Full-scale development (basic platform) (10 years)	Fully operational capabilities (10 years)	Upgraded capabilities (10 years)	Maintenance and support (10 years)	Disposal (5 years)

Consequences

New strategic directions for thriving in the new **unpredictable** world:

- Alignment with best **international practise**
- Need to architect organisation and development environment for **Flexibility**
  - Optimise **overall capability**
  - Ability to adapt the latest **processes, methodology and tools**
- Quick adaptation to **new collaboration scenarios**
  - At **low cost**

32nd Annual **INCOSE**
international symposium

hybrid event

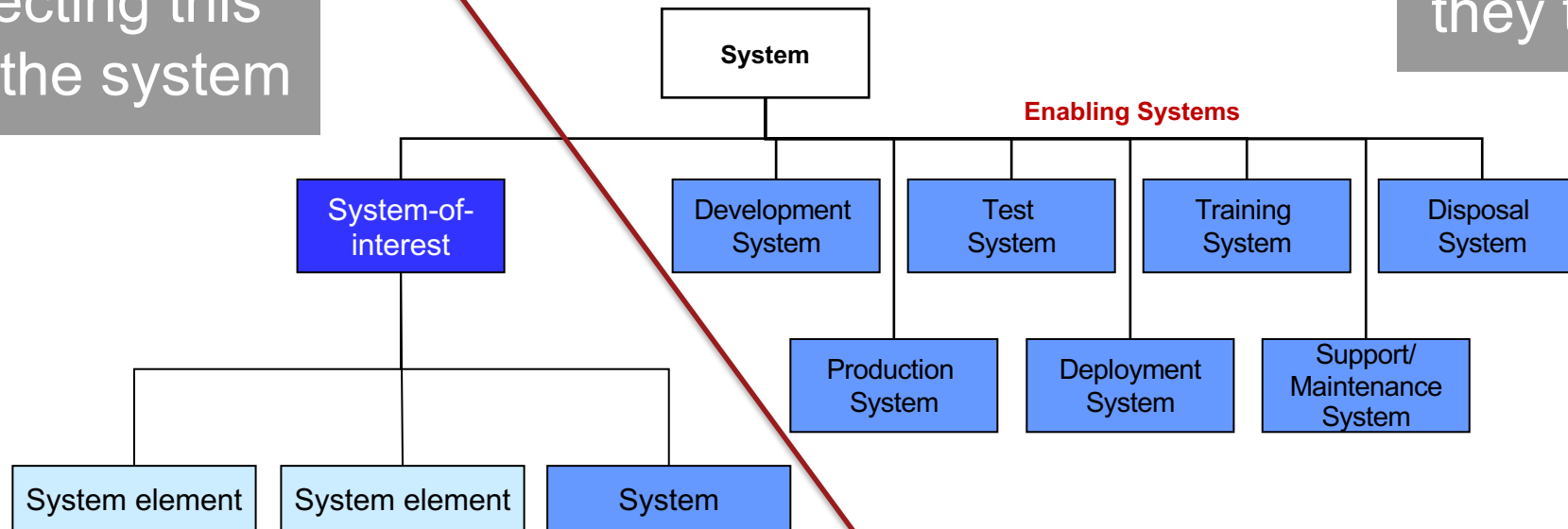Detroit, MI, USA
June 25 - 30, 2022

Architecting the development system
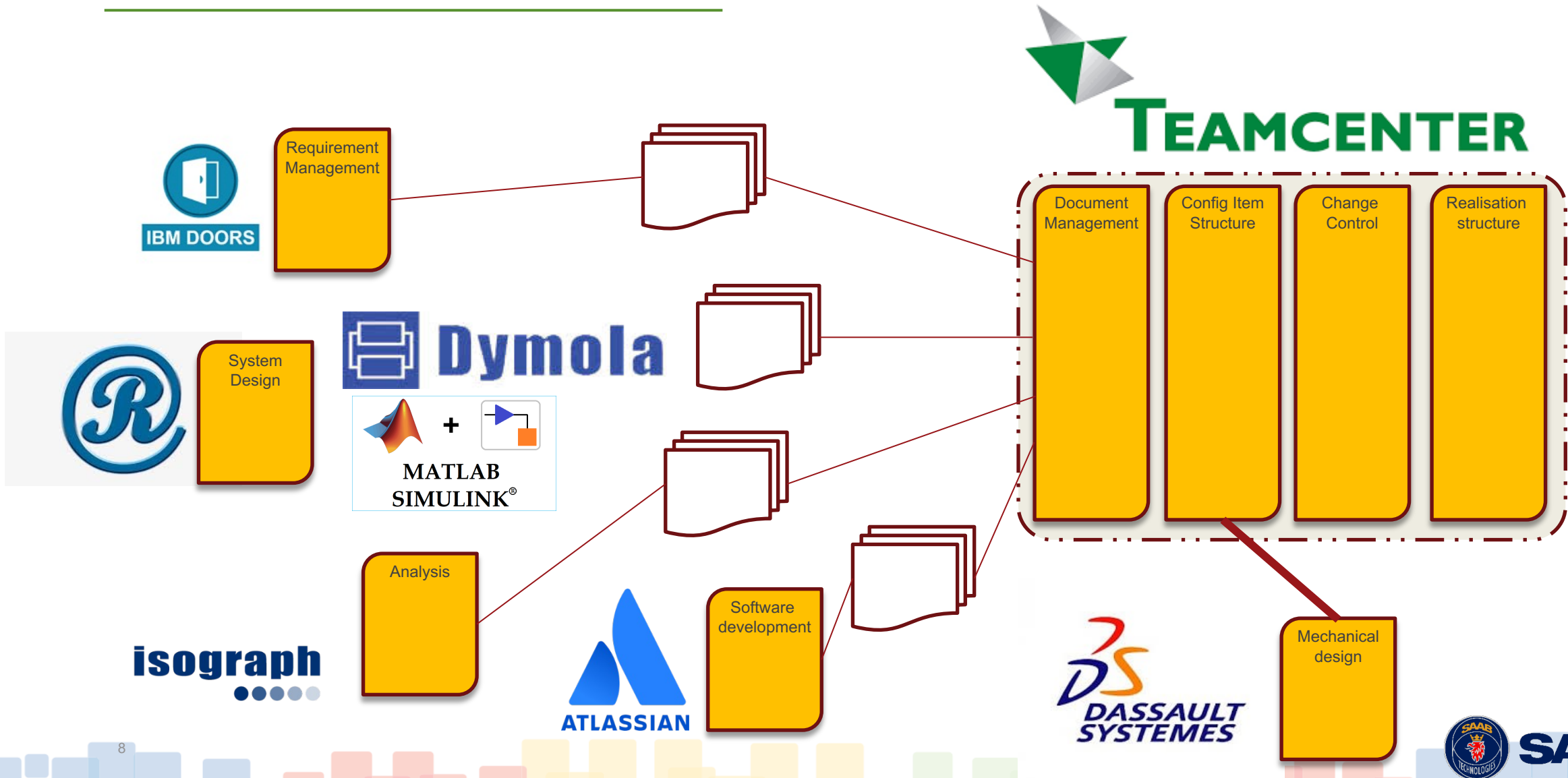
A look at the enabling systems

We are good at architecting this part of the system

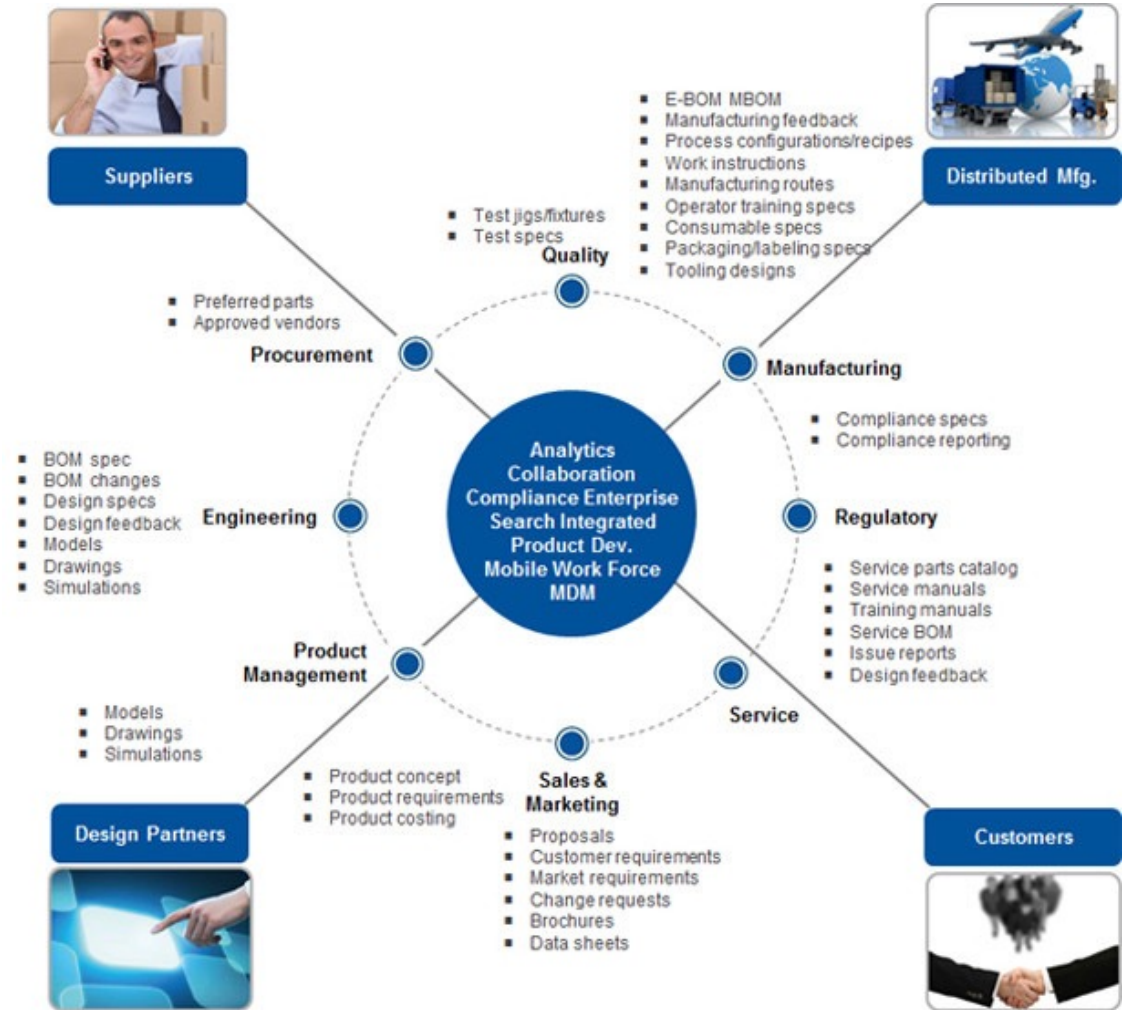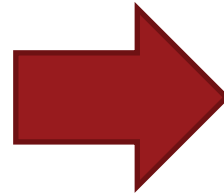Less attention on Enabling systems – they tend to emerge

System

Enabling Systems

System-of-interest

Development System

Test System

Training System

Disposal System

Production System

Deployment System

Support/ Maintenance System
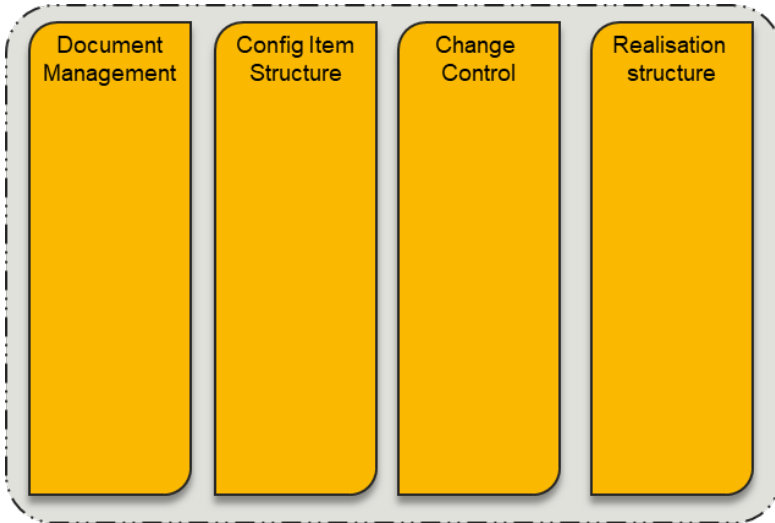
System element

System element

System

Current development tool landscape

Evolution of PLM capabilities

PLM – overview from some large suppliers

Selecting the best PLM environment?

What about
- Overall capability?
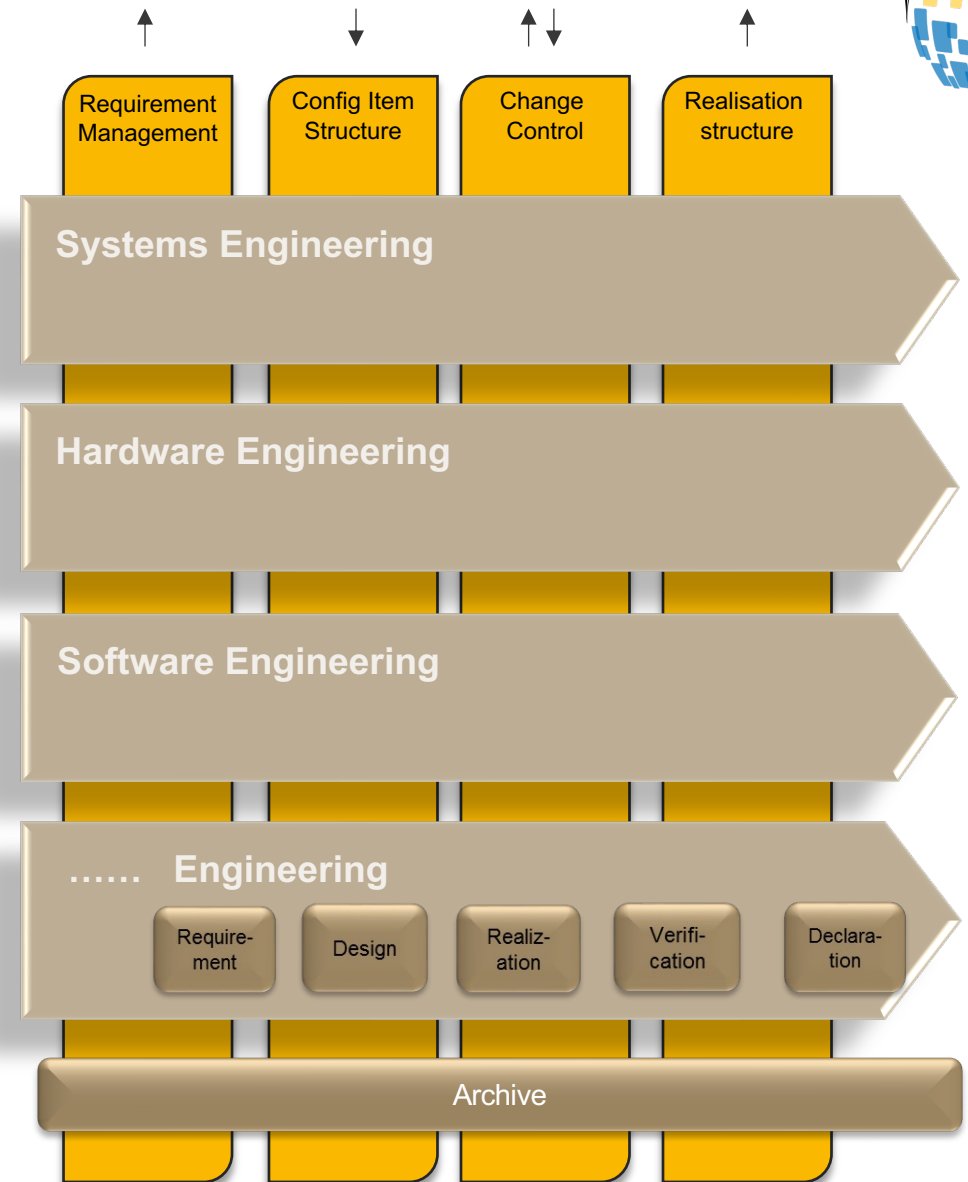- Adaptability?
- Flexibility?

Genesis PLM Model

- Engineering Disciplines

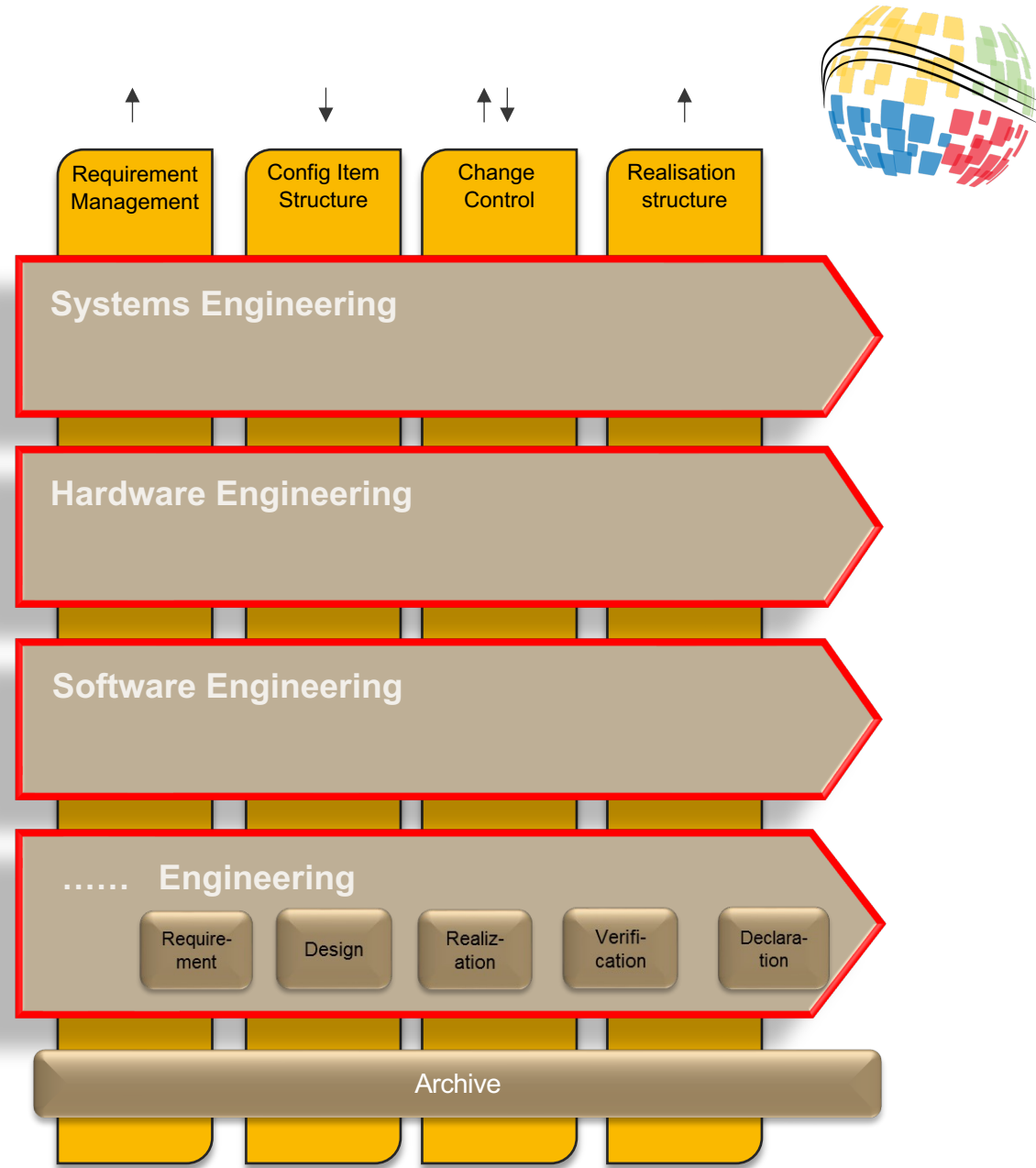- Fine granularity product data

- Design Traceability Dimensions
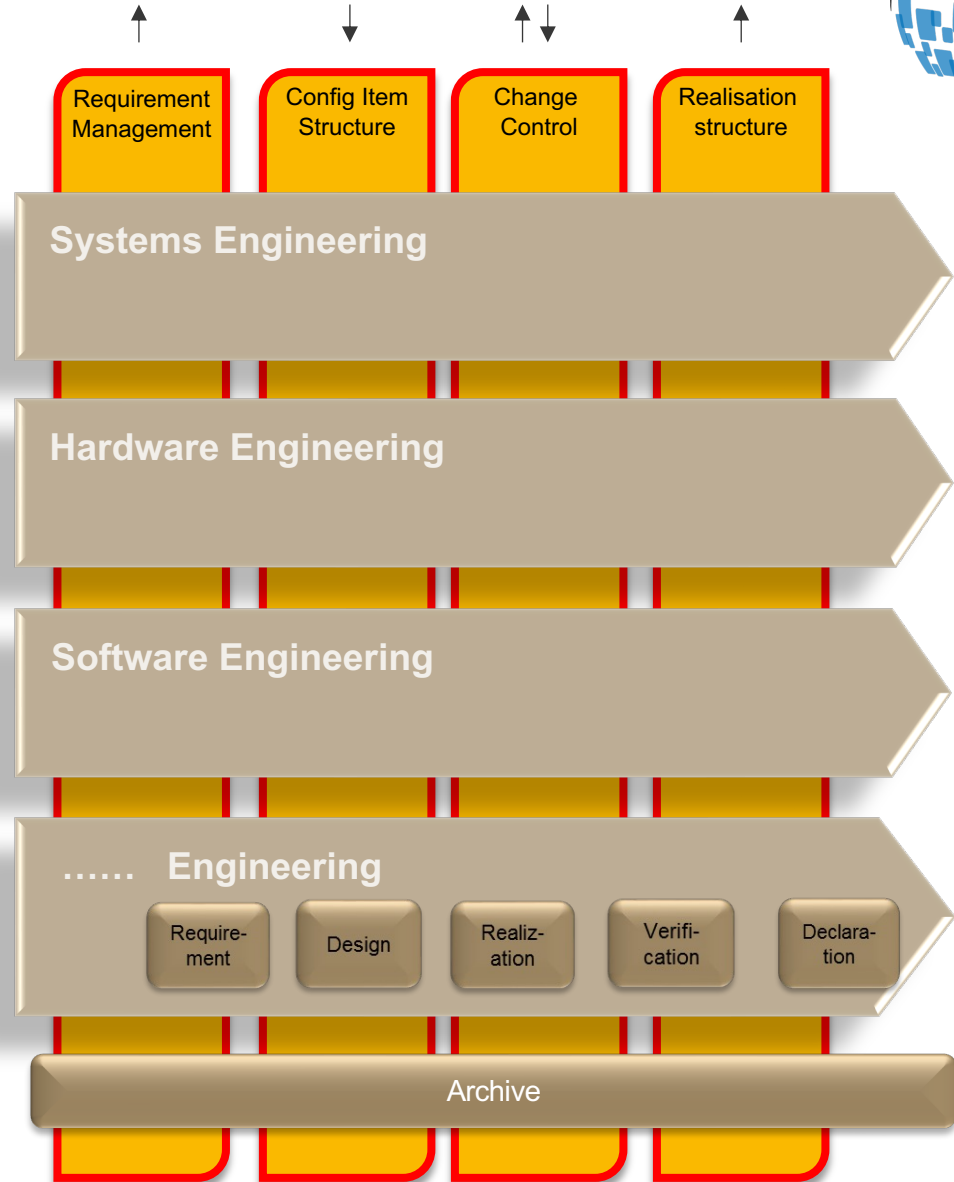 - We believe there are four of them only
- Archiving

Modularity

- Optimise support for each **engineering discipline**
  - **Maximise automation**, as provided by the supplier
  - Minimise application family **switching**
- Bring together **management and engineers** in a single environment
  - E.g., Change management and Status reporting
- **Redundant** capabilities accepted
- Ability to **upgrade or replace** environments without upsetting the complete PLM landscape

Requirement Management	Config Item Structure	Change Control	Realisation structure
↑	↓	↑↓	↑

Systems Engineering

Hardware Engineering

Software Engineering

...... **Engineering**

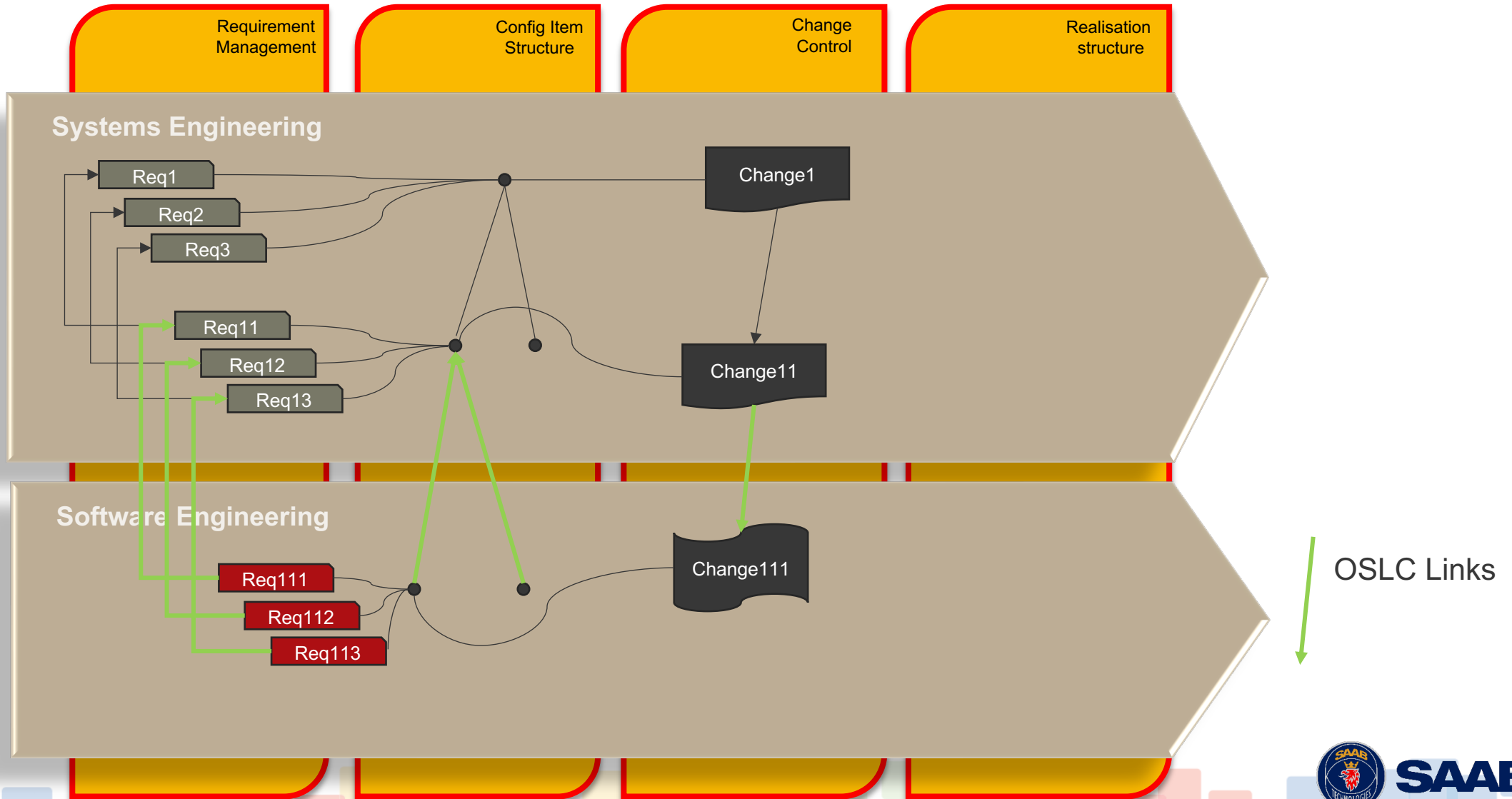| Require-ment	Design	Realiz-ation	Verifi-cation	Declara-tion

Archive

Traceability

- Need capability to ensure **traceability** and **integrity** of product data
- Traceability dimensions between engineering discipline environments
  - **Requirements**
  - **Configuration item structure**
  - **Change management**
  - **Realization**
- Configuration Management capability required for Requirements Traceability, Configuration item structure and Realization structure
  - **Versions and baseline** capabilities
- The **OSLC standard** offers the desired capabilities
  - Exploit for **low** cost and **high quality** integrations

Example System – Software interface
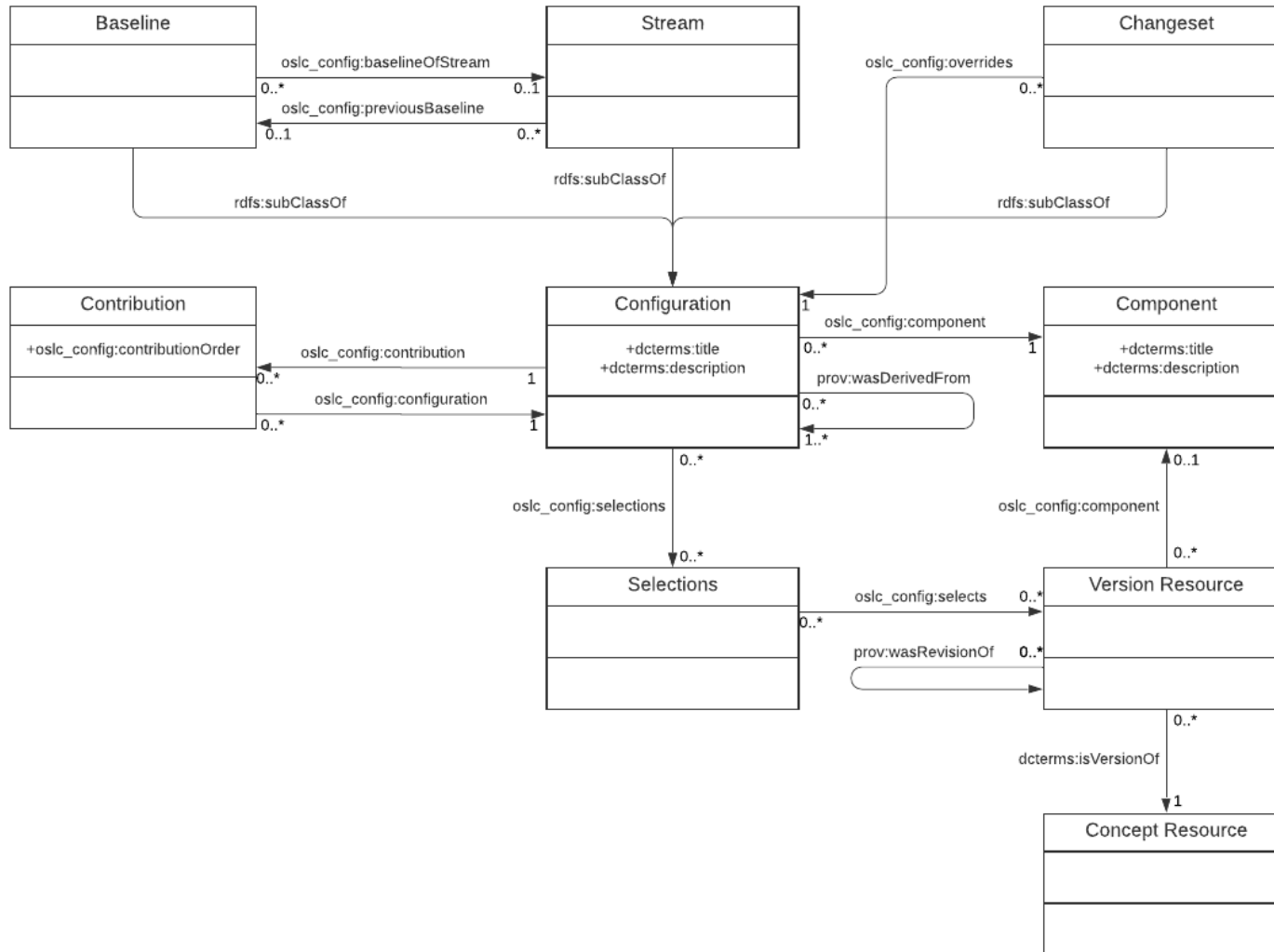
OSLC – does it work?

Scenario:

1. Link an existing requirement in Siemens Polarion to a Defect in IBM ELM

2. Create a new requirement in Polarion and link to the same Defect

Features demonstrated

- Information is linked – not duplicated
- Delegated UI – no need to define a dedicated interface
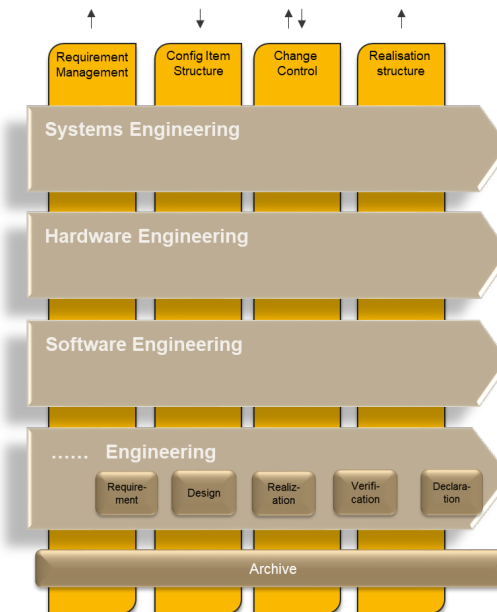- Navigation between applications

OSLC Configuration management

The Heliple project

- Swedish research project to promote the use OSLC – 18 months
- Participants
  - Eurostep
  - KTH
  - Saab
- Scope
  - Promote the Genesis architecture pattern
  - Get experience in OSLC interface creation
  - Improve OSLC interface generation tools
  - Demonstrate the power of OSLC

Conclusions

- Need to prepare for an uncertain future
  - An opportunity, not a threat
- Optimise towards flexibility
- Genesis architecture pattern for federated PLM
  - Optimise process performance
  - Embrace heterogeneity
  - Minimise the number of integration points
- OSLC or equivalent standard is key for enabling plug and play integration
- Heliple – our project for evaluating and promoting OSLC

Contact: Erik.herzog@saabgroup.com for more discussions on OSLC

Any questions?

32nd Annual **INCOSE**
international symposium

hybrid event

Detroit, MI, USA
June 25 - 30, 2022

www.incose.org/symp2022