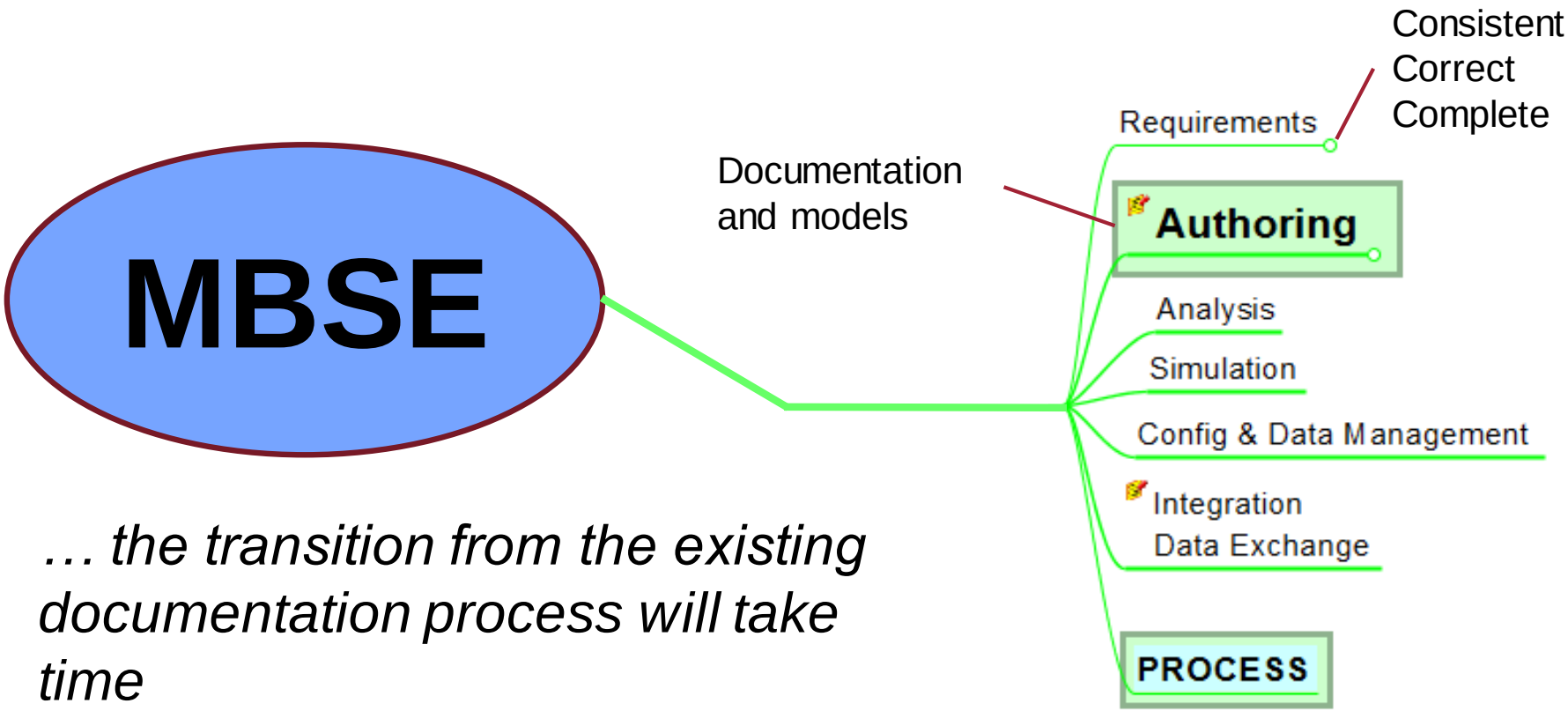
The Transition to MBSE – Maturing the Analysis and Simulation Process

2017 GPDIS Workshop

Mark Williams, Boeing
Hubertus Tummescheit, Modelon

What is MBSE?

Architecture and analysis defined as integrated digital models that are explicit, coherent, and consistent.



… the transition from the existing documentation process will take time

Why do we need MBSE ?

Distributed

Federated

Technology Evolution

CREDIT: IBM Innovate conference in Orlando, 2014. IBM presentation "Continuous Engineering"

90% market penetration	
cars	80 years
TVs	20 years
Smart Phone	5 years
airplanes	60 years

MBSE is a Business Evolution

Distributed

Digital Architecture

Integrated Models

Traceable/Linked Requirements

Reusable Data

Integrated Simulations

Federated

Baseline Architecture

Mostly Documents and CAD Models

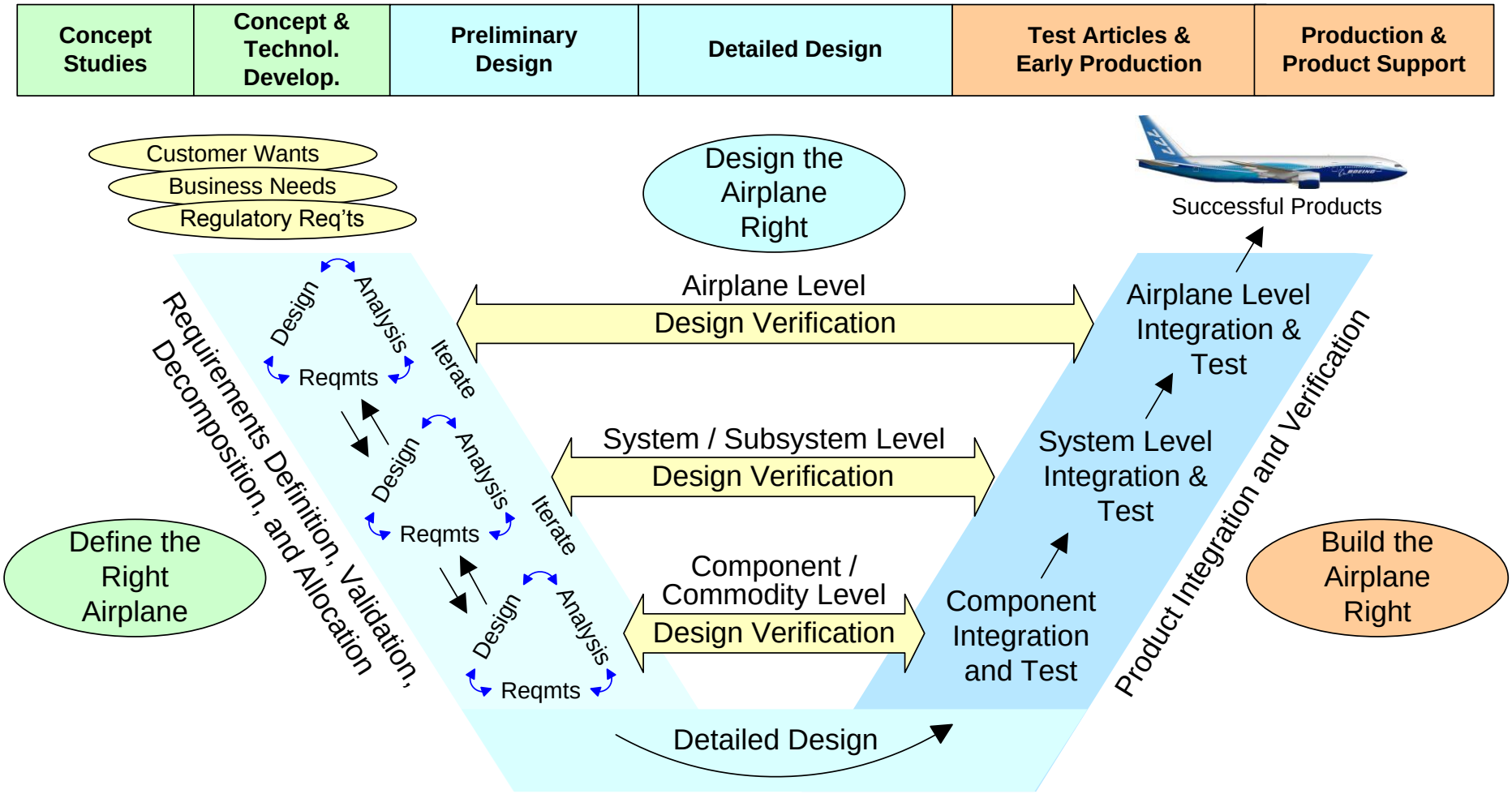
Requirements and Objectives Handbks

Lifecycle until Point of Consumption

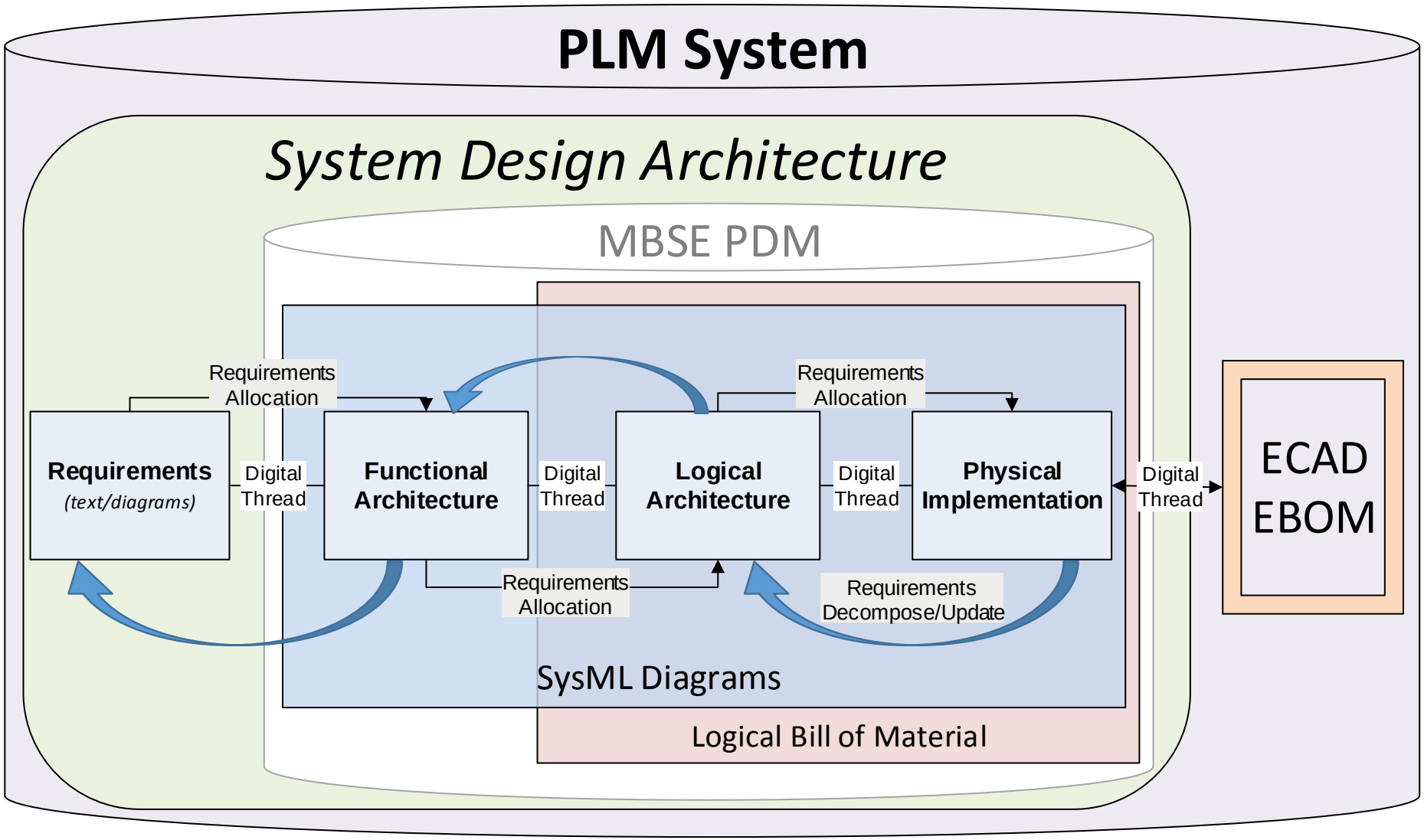
Designs Integrated by Manufacturing

Apply MBSE to the System V

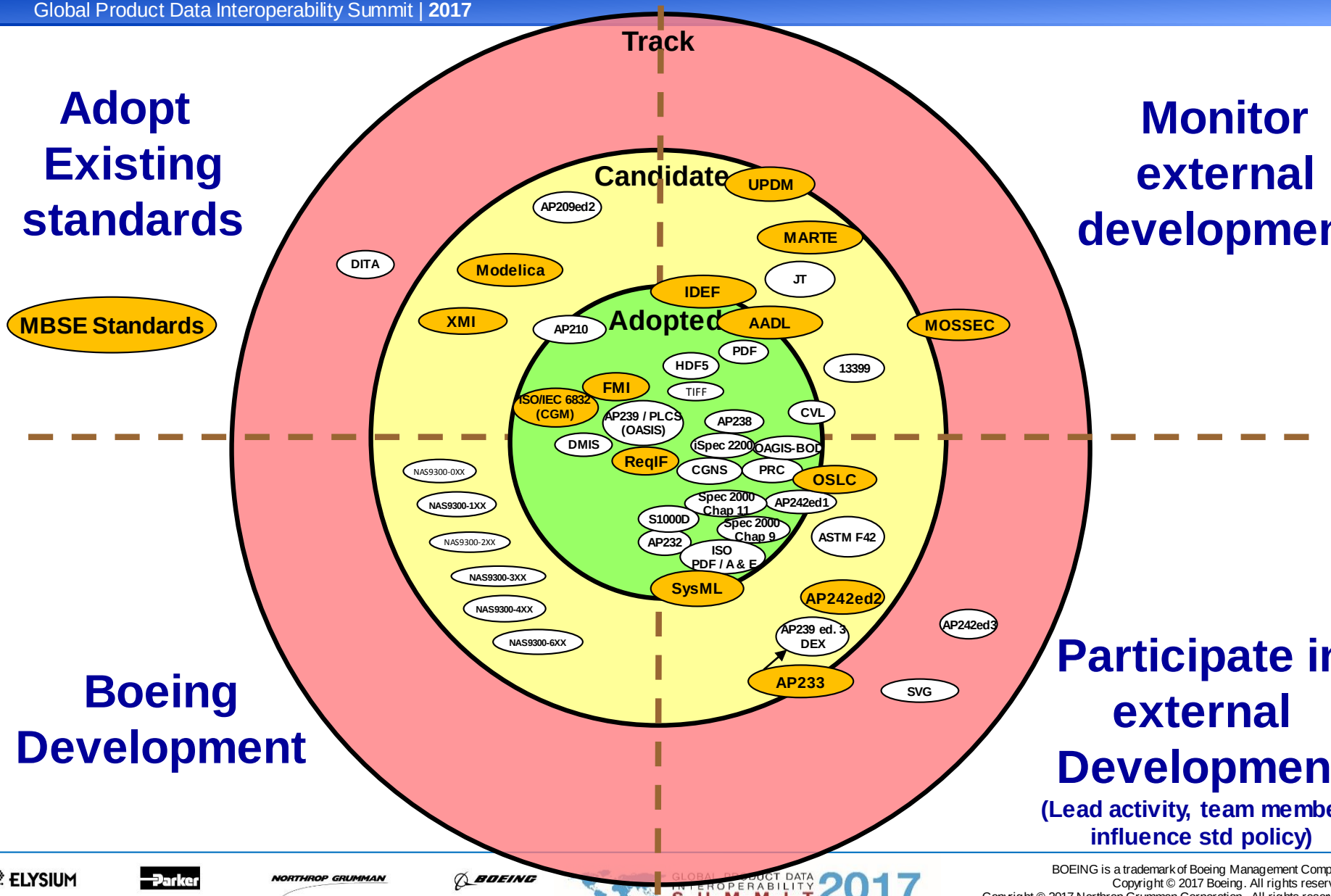
The Systems Engineering 'V'



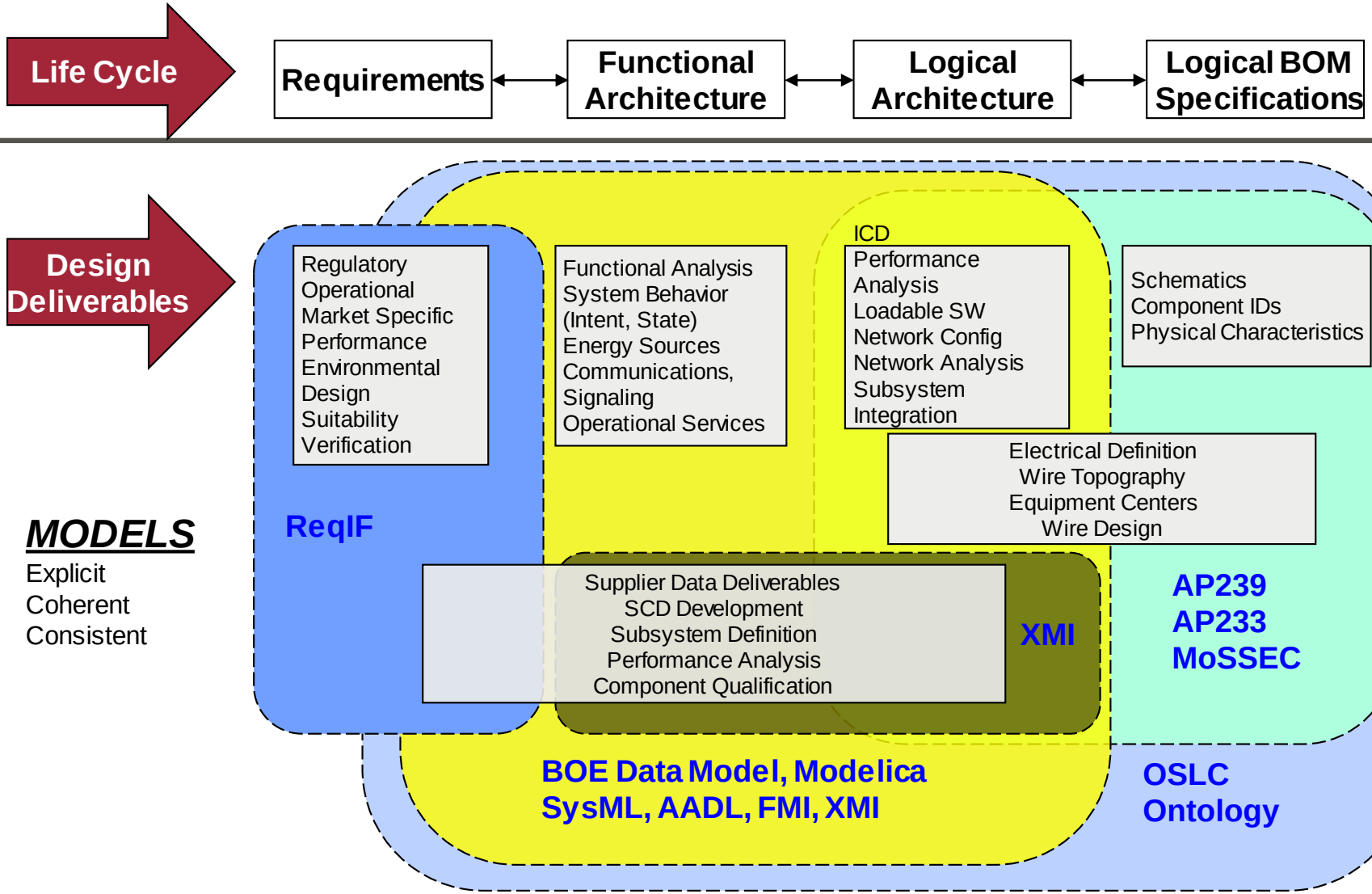
What data is in Scope?
MBSE Design/Business Deliverables



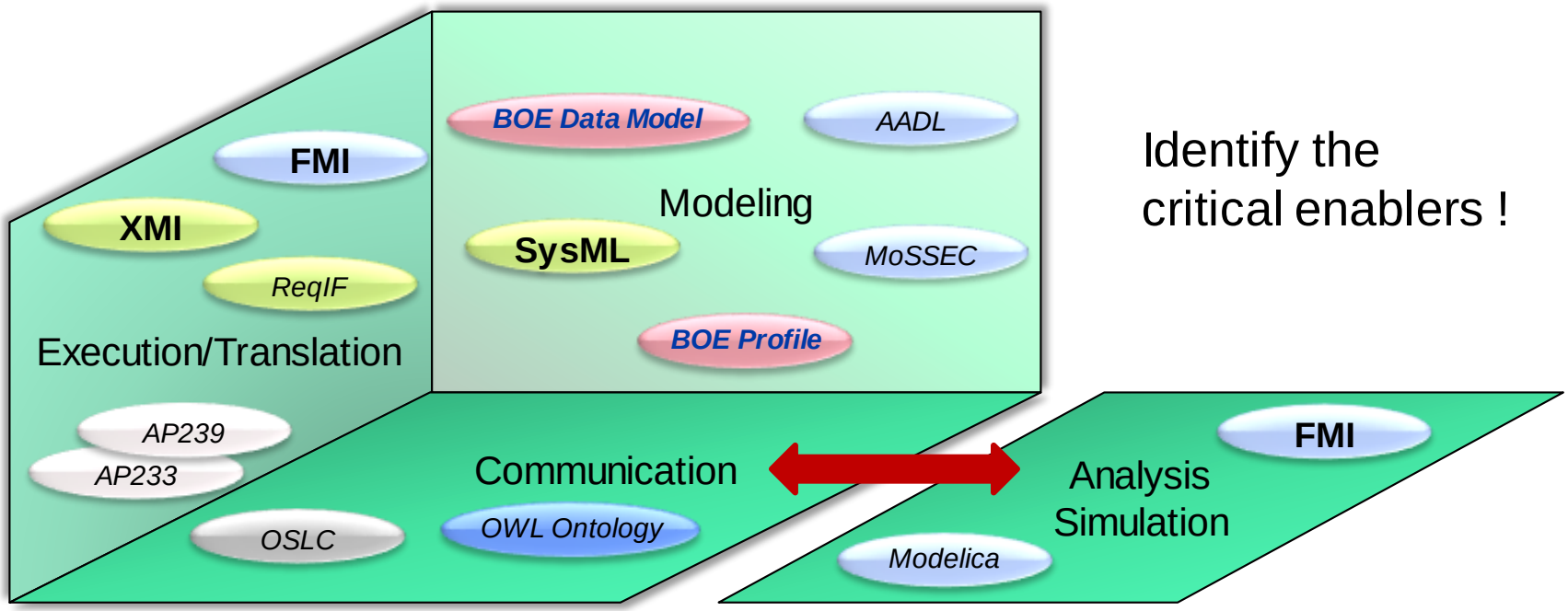
Boeing Data Standards Radar Chart



Align MBSE Standards and Deliverables



Process Deployment of MBSE Standards



Identify the critical enablers !

Standard Body Legend

BOEING | OMG | OASIS | W3C | PDES | SAE/std.org

CREDIT: Bill Chown, Mentor Graphics; MBSE Roundtable, 2015 GPDIS

Intro to FMI

FUNCTIONAL MOCKUP INTERFACE (FMI)

Problems/Needs

- Supplier Components
- OEM as systems integrator
- Many different simulation tools

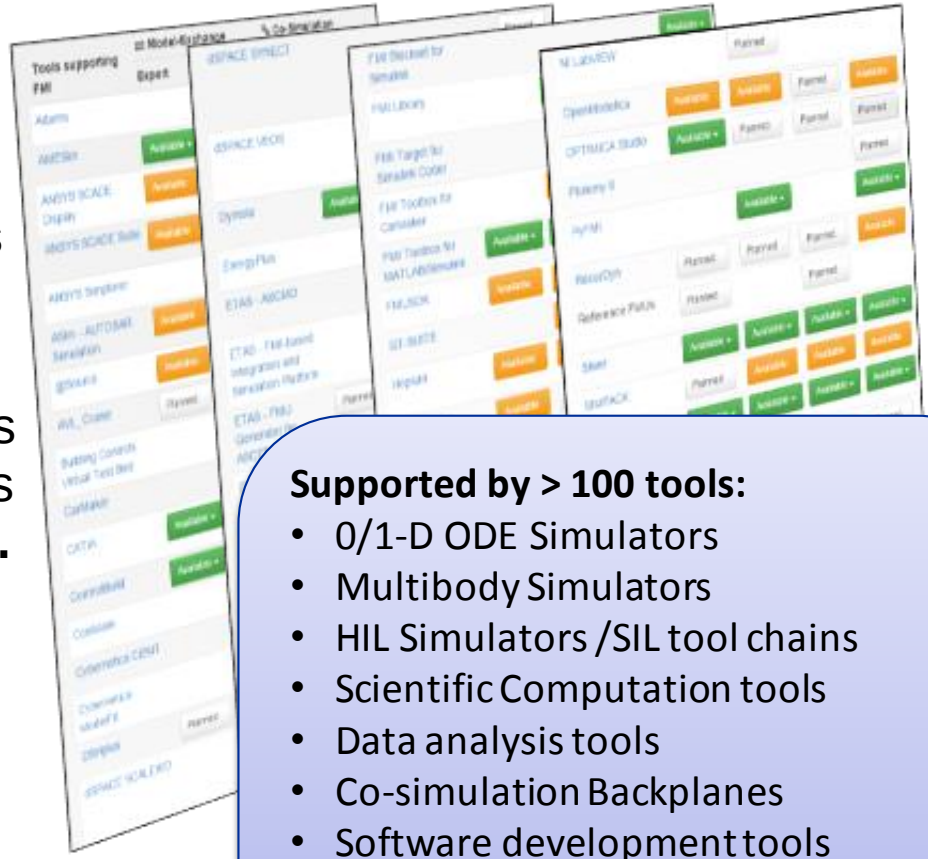
OEM or systems integrator

- Tool independent standard to support both model exchange and co-simulation of dynamic models
- Created to solve model interoperability in automotive industry
- Original development of standard part of EU-funded MODELISAR project led and initiated by Daimler
  First version published in 2010, improved FMI 2.0 in 2014
  Active development as Modelica® Association project

FUNCTIONAL MOCKUP INTERFACE IN A NUTSHELL

- **What is FMI?**
  - an application programming interface and its semantics
  - an xml schema that describes the model structure and capabilities
  - the structure of a zip file that is used to package the model, its resources and documentation**.**

- **> 100 tools support FMI in 10 different categories.**

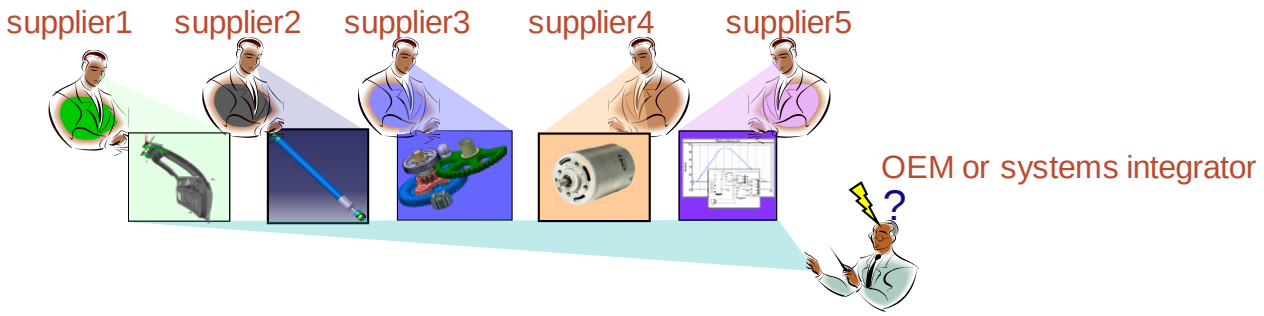


Supported by > 100 tools:

- 0/1-D ODE Simulators
- Multibody Simulators
- HIL Simulators /SIL tool chains
- Scientific Computation tools
- Data analysis tools
- Co-simulation Backplanes
- Software development tools
- Systems engineering tools
- SDKs, legacy integration
- CFD tools

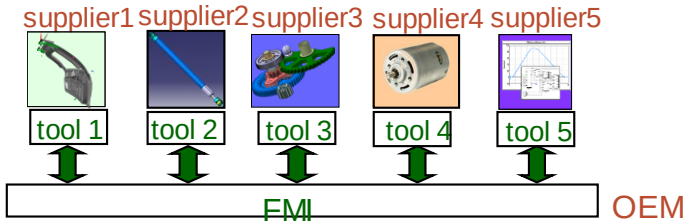
up/-to-date list of tools:
www.fmi-standard.org/tools

USE CASE I:



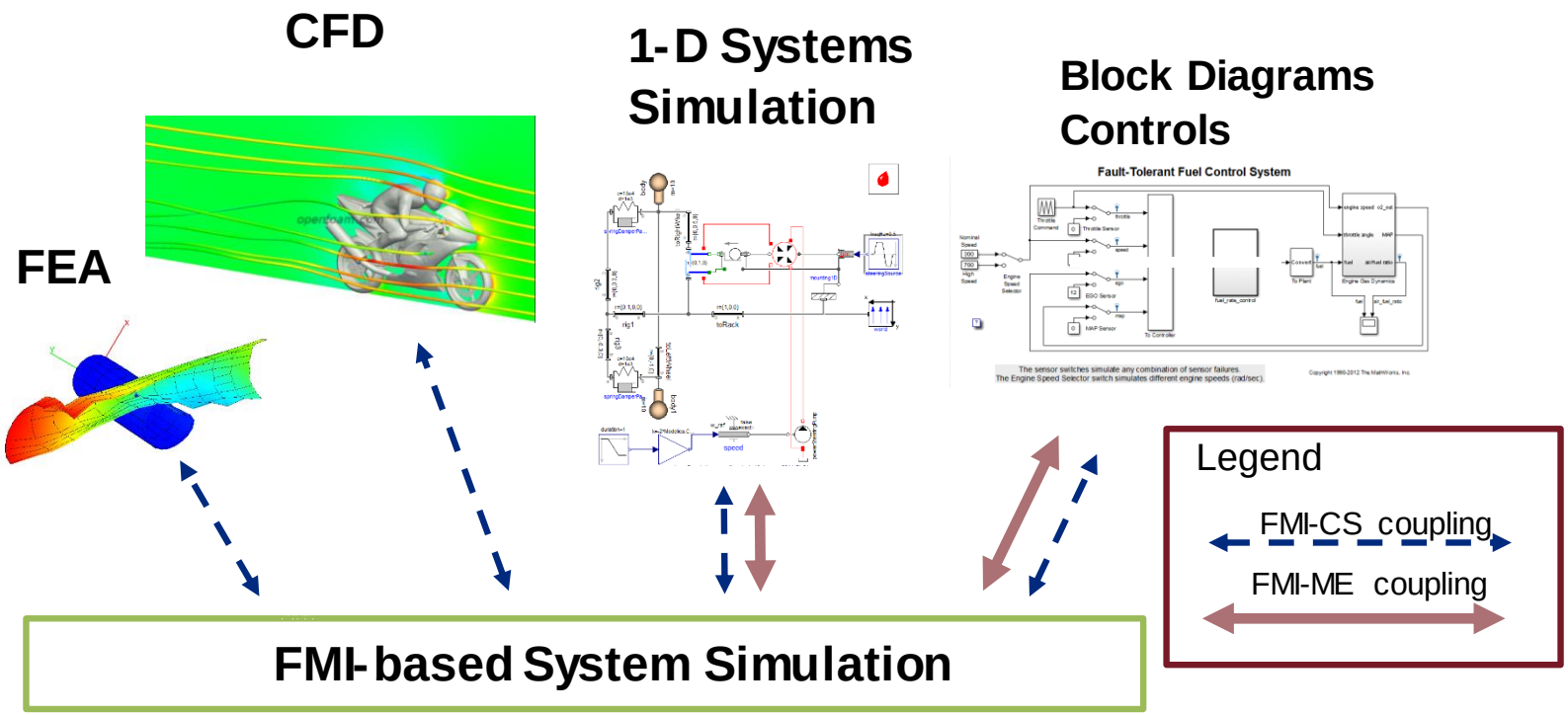
Combined simulation for system integration

- As a universal solution to this problem the Functional Mockup Interface (FMI) was developed by the EU-project MODELISAR, and is now maintained by the FMI project of the Modelica® Association



USE CASE II:

- Combine different modeling formats into coherent co-simulation (cyber-physical systems)
  - Physical models, 1D-3D (not 3D to 3D!)
  - Controls / Software



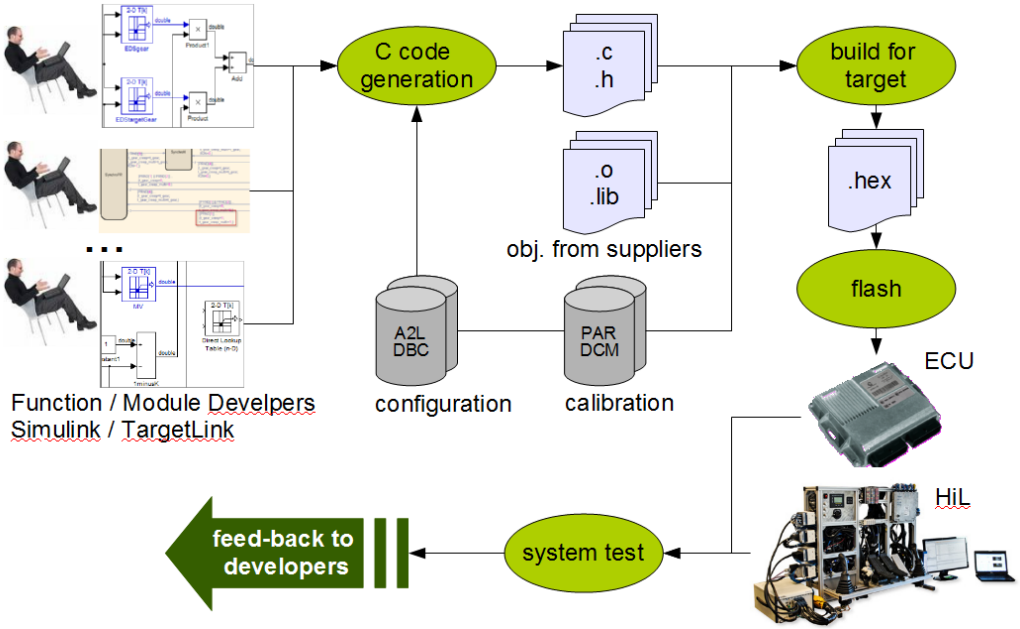
From the official FMI presentation (adapted)

USE CASE III: FMI FOR MIL, SIL AND HIL

- FMI export support from Controls Tools:
  - Matlab/Simulink through FMIT Coder (Modelon)
  - Scade Suite (safety critical applications)

- FMI supported by most major HIL Vendors
  - DSPACE
  - National Instruments
  - Concurrent
  - IPG
  - Speedgoat

- FMI for ECU virtualization
  - Silver by Qtronic
  - ETAS tools (Bosch)

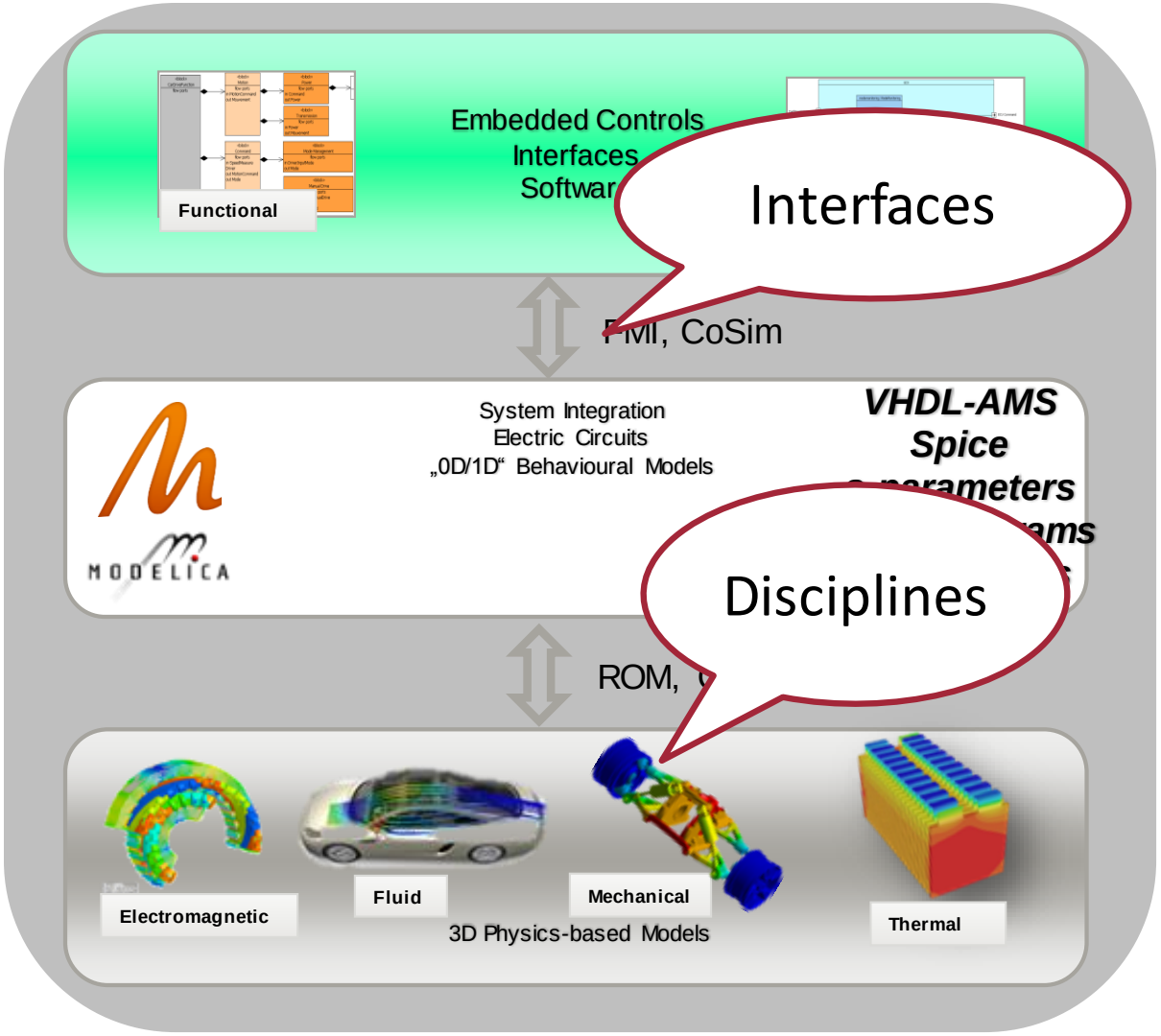


FMU: A MODEL WITH STANDARD INTERFACE

- A component which implements the FMI standard is called *Functional Mockup Unit (FMU)*
- Separation of
  - Description of interface and meta-data (XML file)
  - Functionality (C code or binary)
- A FMU is a zipped file (*.fmu) containing the XML description file and the implementation in source or binary form
- Additional data and functionality can be included
- Information & Interface specification: www.fmi-standard.org

From the official FMI presentation (adapted)

System Level Integration



ROM = Reduced-order model
FMI = Functional mock-up interface
CoSim = Co-Simulation

Interoperability

.dll,
FMI,
CoSim

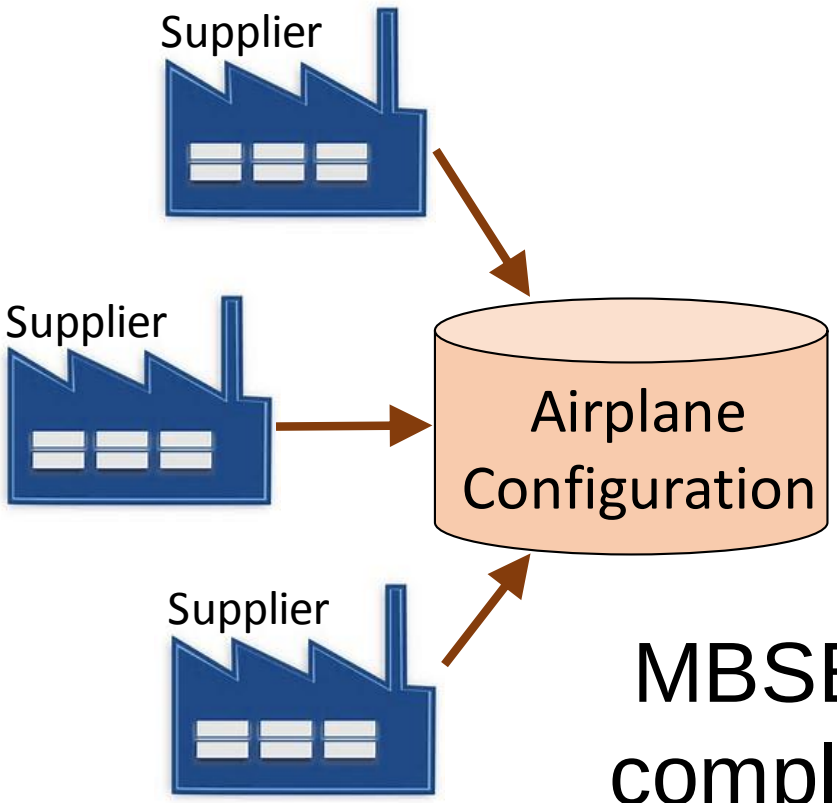
MATLAB SIMULINK
GT
LMS
Domain specific Tools

A BUSINESS MODEL INNOVATION THAT ENABLES COLLABORATION

- FMI-compliant tools often provide an export mode to generate models for license-free distribution in the organisation and to partners
- This is a unique enabler for model exchange inside one company and for OEM/supplier collaboration
- Deployment from few simulation specialists to designers, domain specialists, control engineers
  - One FMU used by many engineers (control design)
  - One FMU run on many cores (robust design)

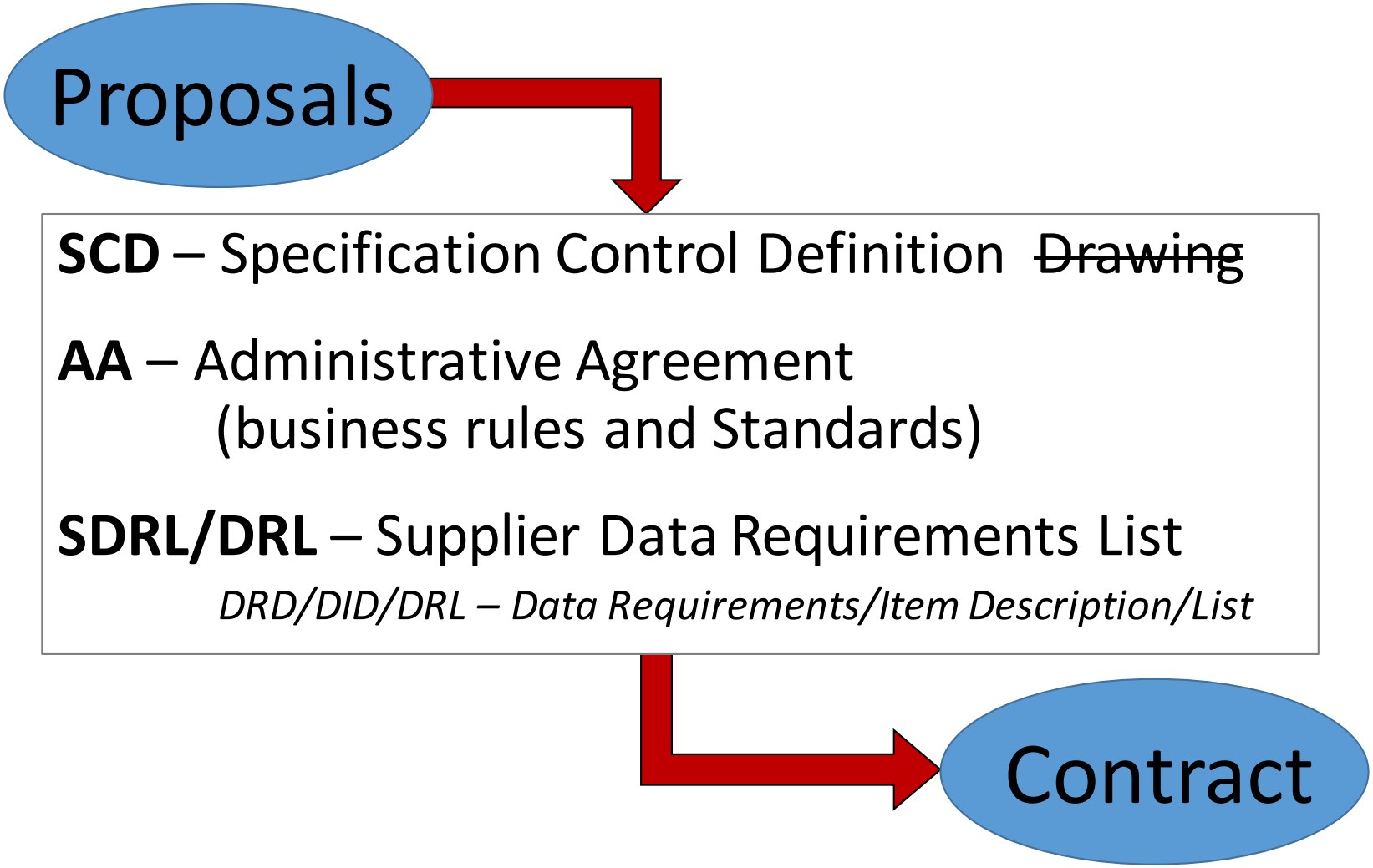
Implementing a Program

OEM – Supplier Dependencies

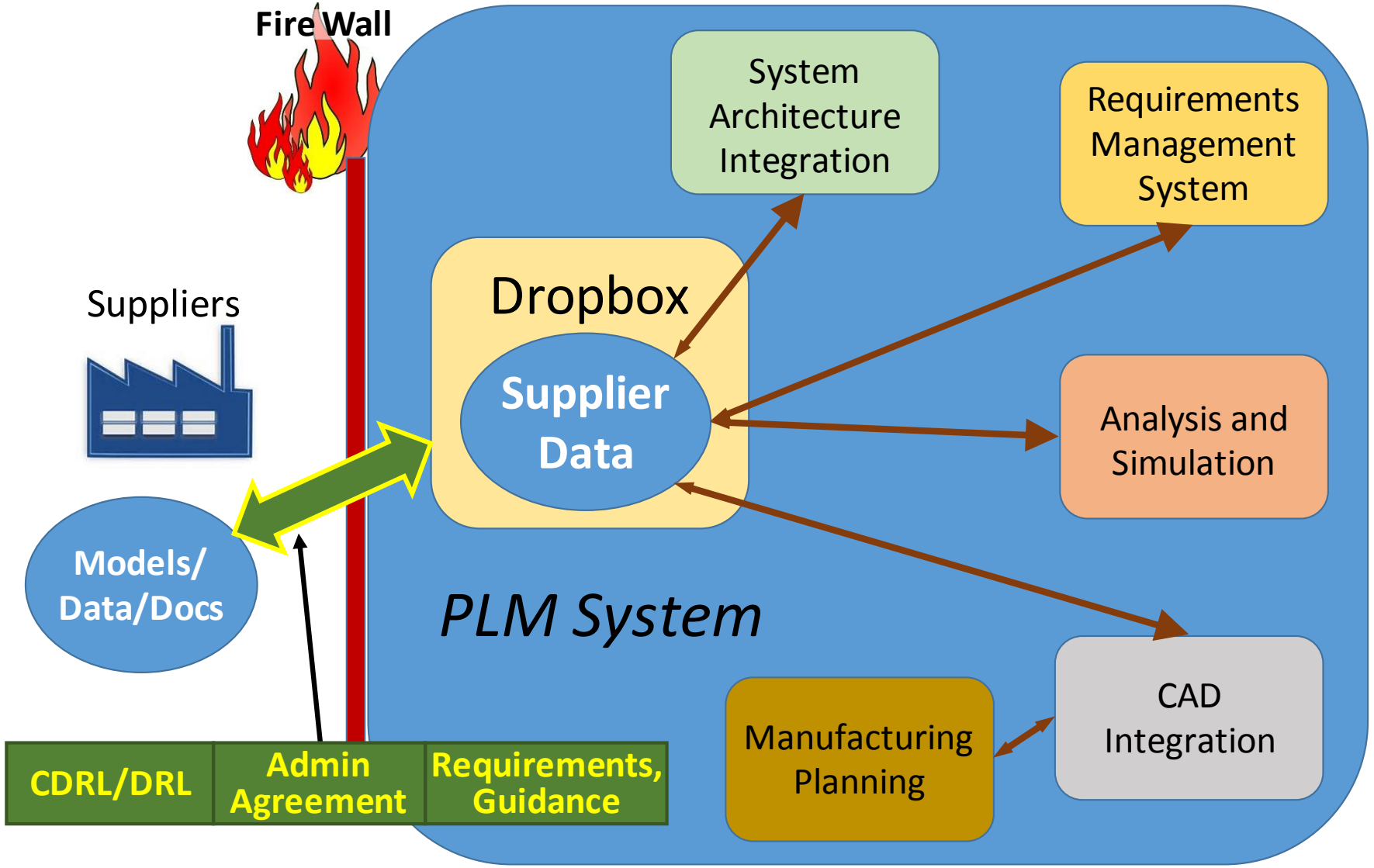


MBSE and Standards get complicated when you add Suppliers!

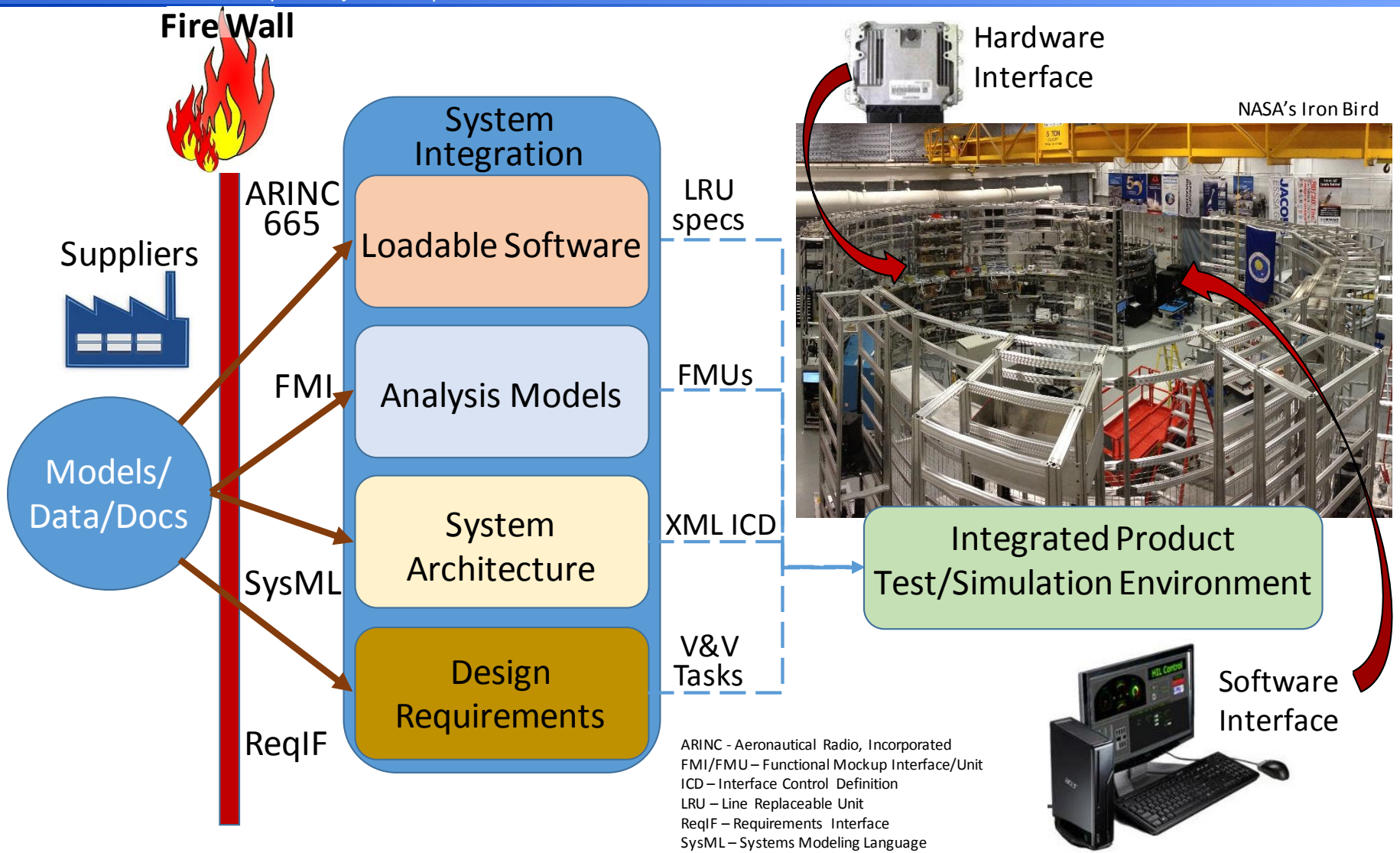
OEM – Supplier Engagement



Manage MBSE Infrastructure



Simulation is KEY to Design Integration



Together we must Influence the Industry

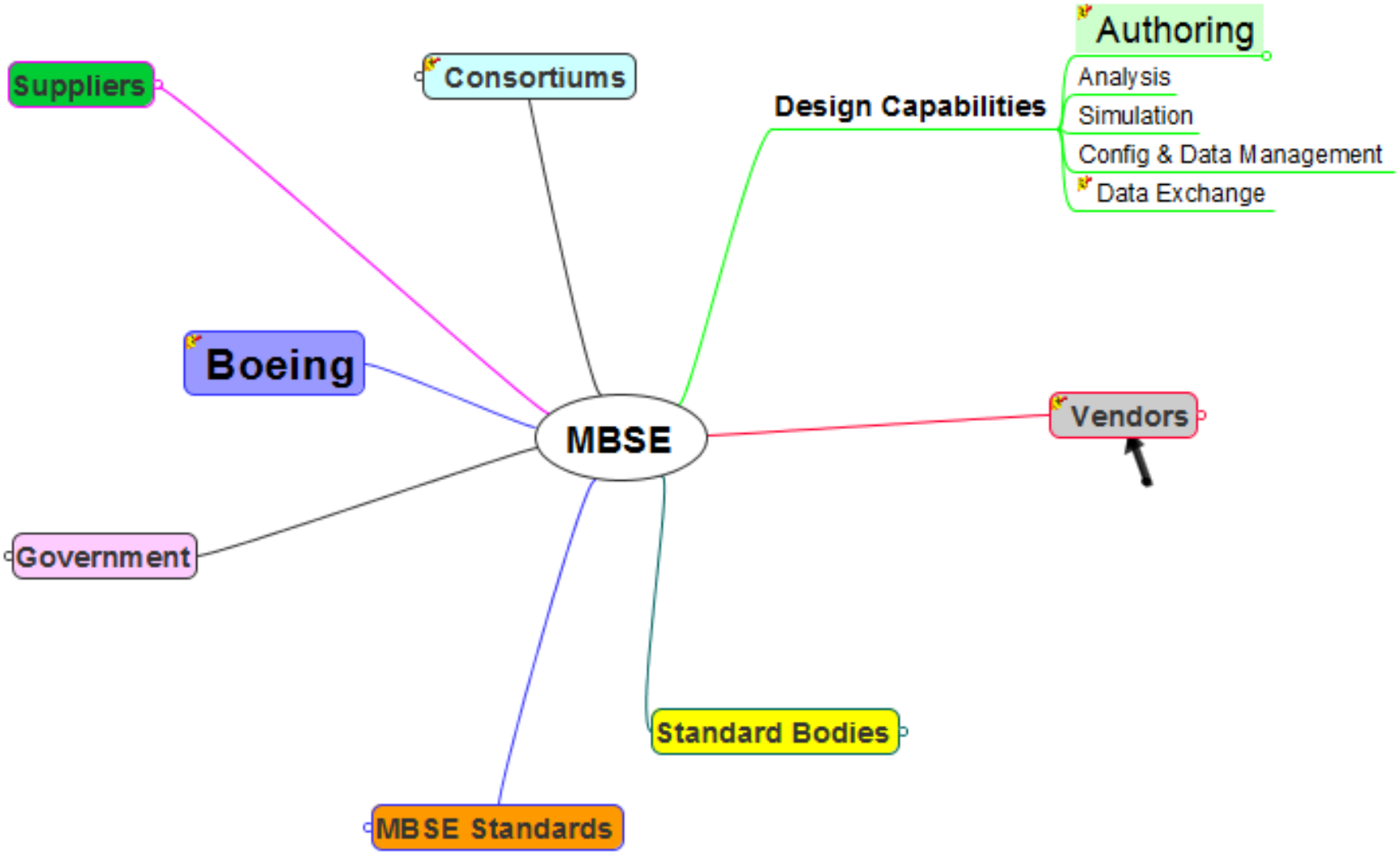
WHO:

Partners, Suppliers, Vendors, Standard Bodies

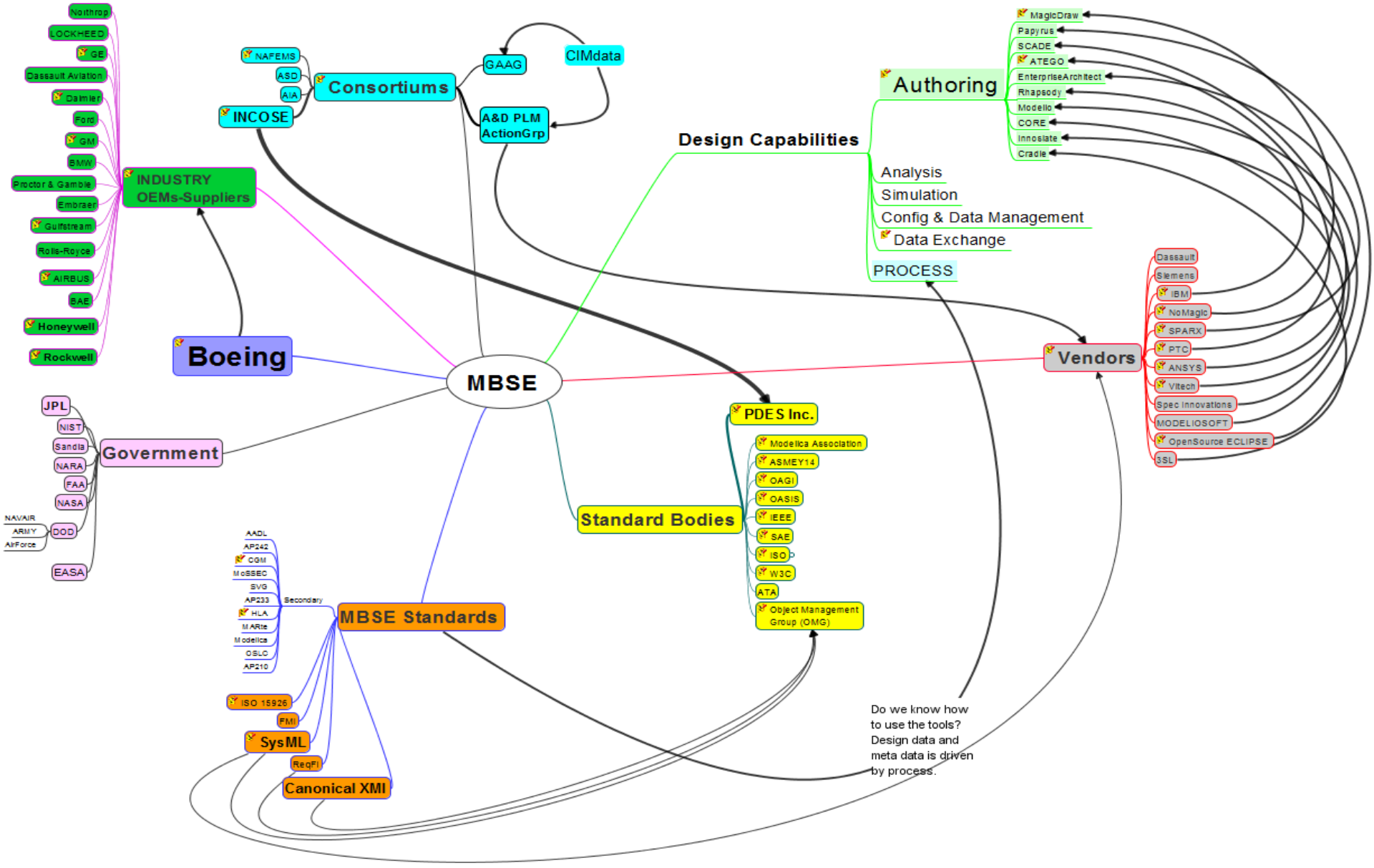
WHAT:

- **Requirements Traceability – StableID, source, revisions**
- **Exchange SysML – Canonical XMI, and Linked Data**
- **Standard Data model – ports, network, software**
- **Analysis integration – FMU meta-data, multi-OS**
- **All Model meta-data – support reuse (MoSSEC spec)**

Industry Players



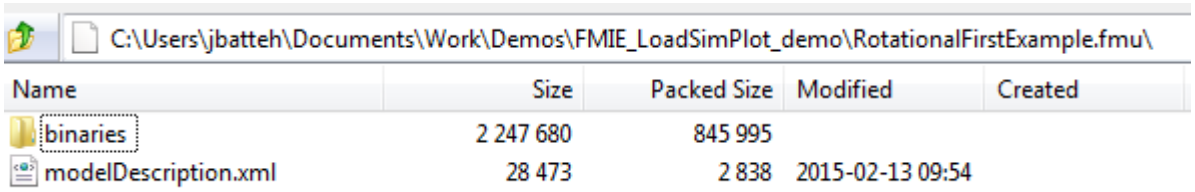
Influence Strategy



Implementing the Analysis

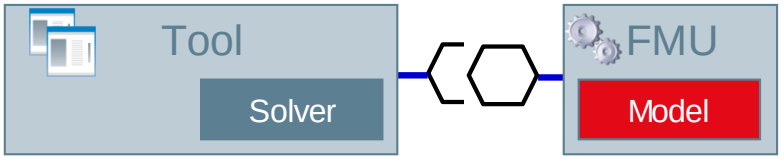
WHAT IS AN FMU?

- Component which implements the FMI standard is called Functional Mockup Unit (FMU)
- Zipped file (*.fmu) containing the XML description file and the implementation in source or binary form
- Separation of description of interface data (XML file) from functionality (C code or binary)
- Additional data and functionality can be included in file
- Information and interface specification: www.fmi-standard.org

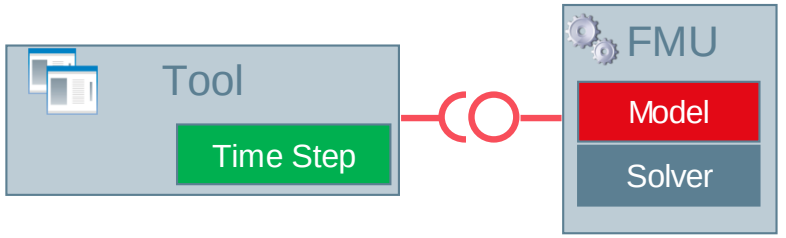


FMI FLAVORS

- The Functional Mock-up Interface (FMI) is a tool independent standard for
  - Model Exchange (ME)



  - Co-Simulation (CS)



- The FMI defines an interface to be implemented by an executable called Functional Mock-up Unit (FMU)

FMU=Model w/ Standard Interface

FMI IS GREAT: ENABLES COLLABORATIVE MBSE

- Standardized, open, vendor neutral API
- Convenient container for handling simulation artefacts: storing, sharing, long term archiving…
- Free simulation users from modeling/generation tool knowledge
- Reduce IP sharing

- A new quality of simulations is attainable now, because:
  - Producing, sharing and using simulation components is simpler than ever
  - Coupling multi-disciplinary simulations is now more efficient than ever

Before FMI

With FMI

What we want

FMI IS GREAT – BUT NOT MAGIC!

- Standardized, open, vendor neutral API
- Convenient container for handling simulation artefacts: storing, sharing, archiving…
- Free simulation users from modeling/generation tool knowledge
- Reduce IP sharing

- A new quality of simulations is attainable now, because:
  - Producing, sharing and using simulation components is simpler than ever
  - Coupling multi-disciplinary simulations is now more efficient than ever

Before FMI

With FMI

What some of us get

FMI IS GREAT, BUT CO-SIMULATION IS HARD

- FMI should be part of every companies virtualization strategy, but:
- Co-simulation is difficult
  - Co-simulation is simpler than model exchange
  - Co-Simulation introduces time delays at couplings
    - Can cause instabilities and cause deviations
    - Can be reduced by sophisticated Master Algorithms
  - Numerical properties of bad coupling choices can easily lead to failure!

- Guidance and process needed for successful separation of models into subsystems

WHICH FMI FLAVOR IS BEST?

- Combined FMI for Co-simulation
  - Use if model exchange is not supported
  - Use for sampled data/ discrete time systems
  - Use to couple plant with controller
  - Needed if combining CFD or FEM models

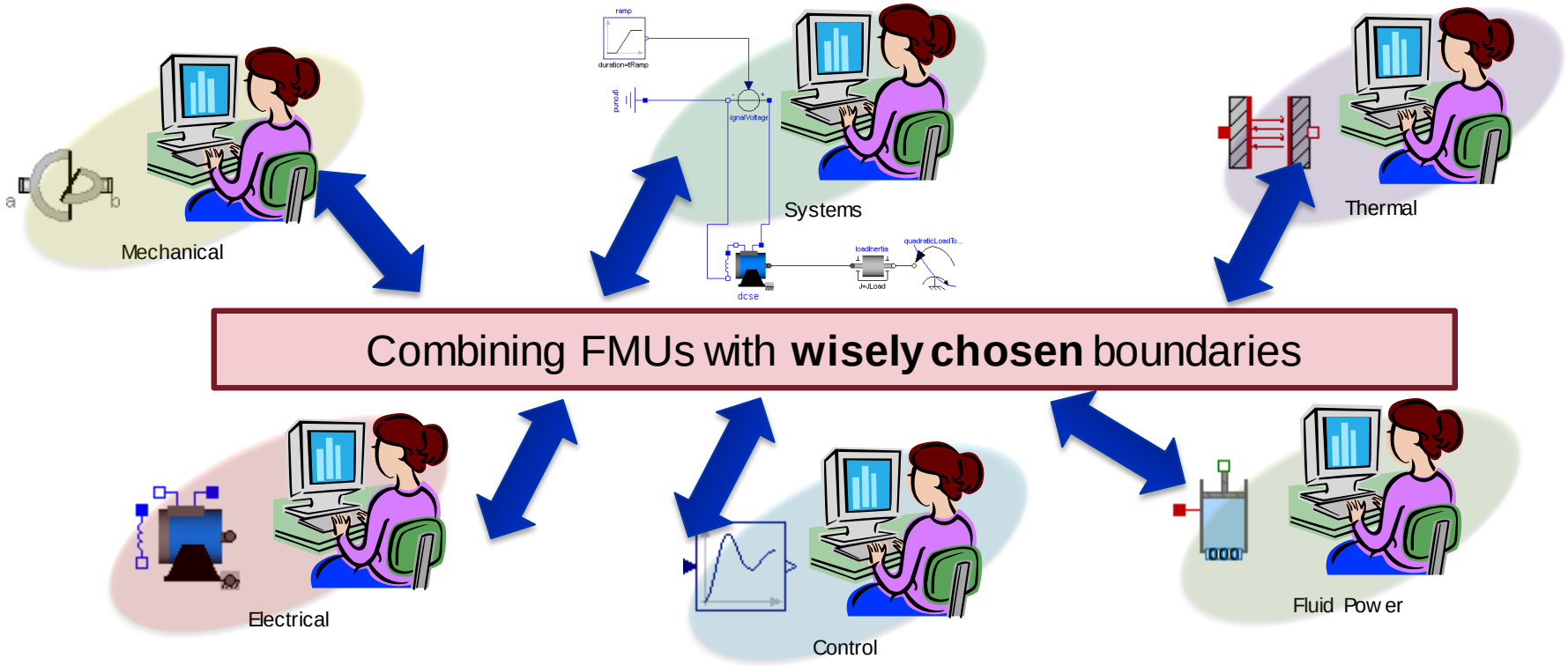
- Combined FMI for model exchange
  - use a single solver: no co-simulation problems, no delays (controls analysis)
  - Use for combining different engineering domains, if possible

MODELING GUIDELINES

1. Keep the number of components low: FMI is not built for libraries of small components
2. Avoid algebraic loops through multiple FMU's
3. Separate discrete time vs continuous time subsystems
4. Do not introduce model boundaries into connections with fast dynamics and tight coupling
5. Carefully design input-output selection for physical connections

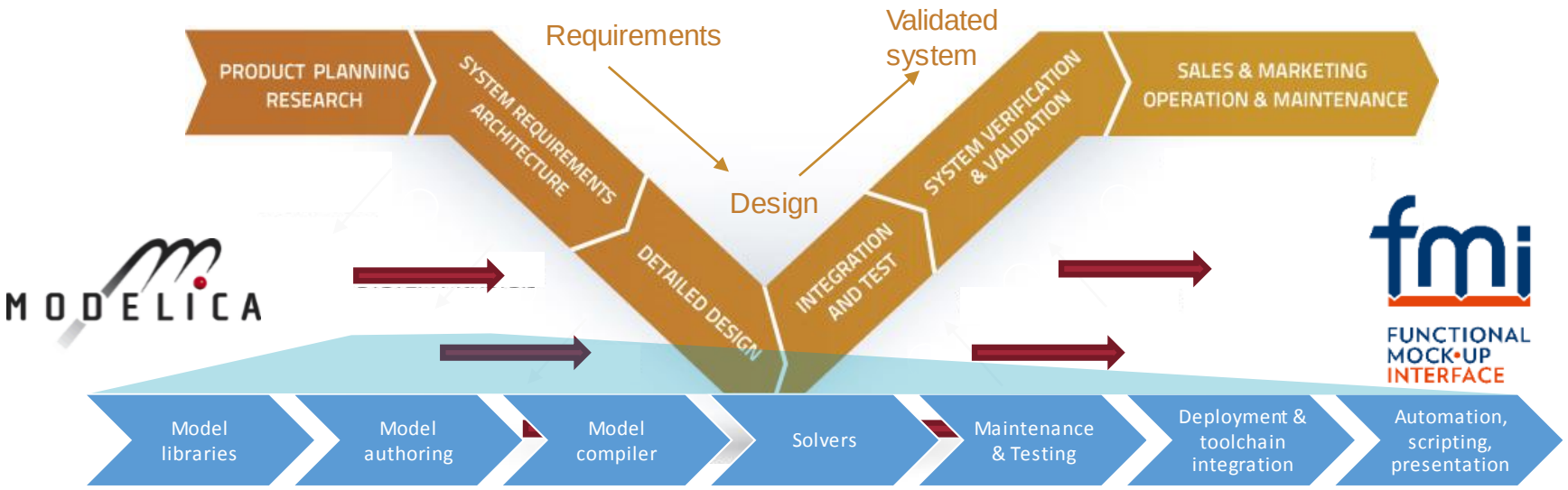
MULTI-DOMAIN & INTER-COMPANY COLLABORATION

- Engineers in different domains work **with FMUs**
  - Share models, distributed collaboration, work in tool of choice, reduced license costs, protect IP, couple carefully!!



FMI SUPPORTS THE ANALYS AND SIMULATION PARTS OF A COMPLETE MBSE WORKFLOW

Proven model-based methodologies and technology to decrease development time and costs while making it possible to manage product complexity and enhance innovation and performance



Engineering tasks executed/solved using modeling & simulation