34th Annual INCOSE international symposium

hybrid event

Dublin, Ireland
July 2 - 6, 2024

Ilsoo Jeong, Shashank Alai, Jaekop Joo, Michael Baloh, Sunkil Yun, Tae Kook Kim, Hwi Seob Park

Hyundai's Modular MBSE Approach to 'Purpose Built Vehicle' Architecture Development

HYUNDAI MOTOR GROUP

SIEMENS

Presenter Bio

Ilsoo Jeong

Driving Comfort Virtual Development Team
Hyundai Motor Company
Seoul, South Korea
isjeong@hyundai.com
+82-31-5172-0091

Shashank Alai

Solution Consultant – MBSE Solutions
Siemens Digital Industries Software
Cincinnati, OH USA
shashank.alai@siemens.com
+1 (513) 576-2888

Authors' Affiliation:

Ilsoo Jeong (Hyundai)

Shashank Alai (Siemens)

Jaekop Joo (Hyundai)

Michael Baloh (Siemens)

Sunkil Yun (Hyundai)

Tae Kook Kim (Siemens)

Hwi Seob Park (Siemens)

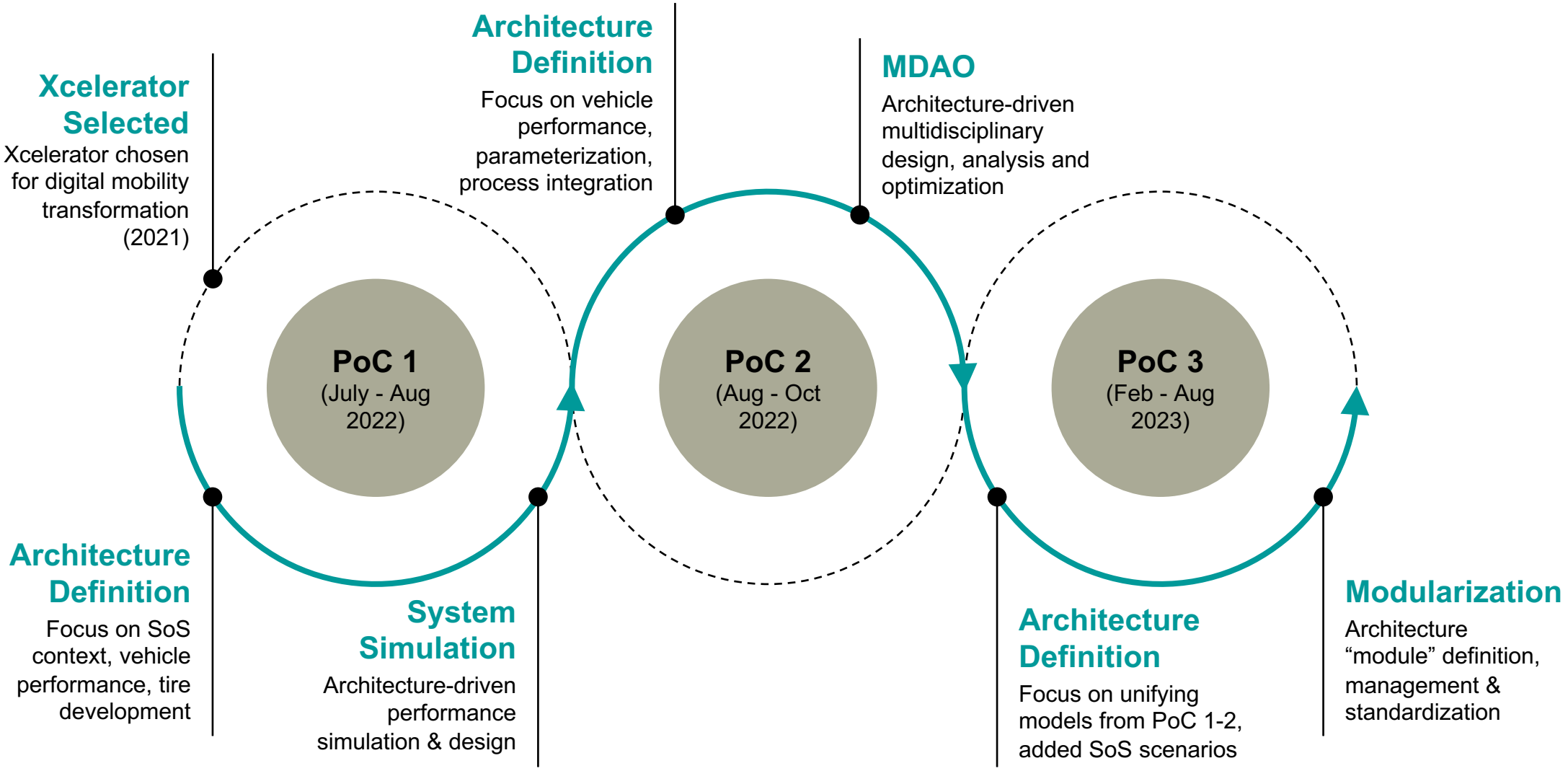
Outline

- Recap
- Introduction
- MBSE LIfecycle
- Modular MBSE
- Conclusion & Observations

Summary of Projects

Recap

Summary of MBSE Projects at Hyundai

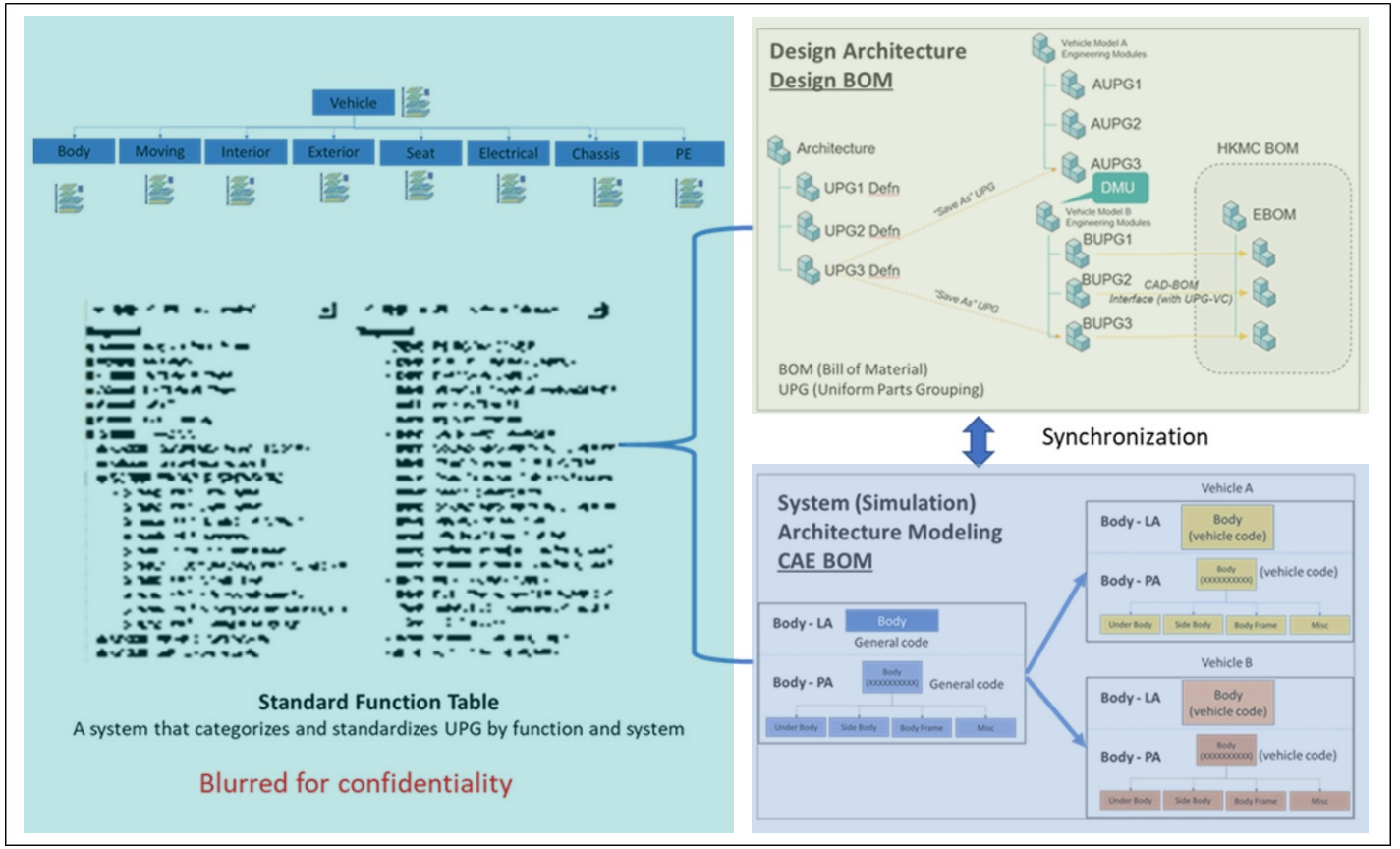


Background & Motivation

Introduction

Relevant Background

- Standard function table (UPG)
- xBOM
  - Design BOM
  - CAE BOM
- Standardized Coding



Design data management system artifacts

Motivation

Dramatic complexity explosion
Automotive systems are SoS
Emergent mobility features

Homogenous system modeling
Complicated representations of system
Limited standardization & reuse

Document-based SE
Communication challenges
Time/cost of data duplication

Lack of digital threads
Parameter/requirement traceability
Disconnect with design infra

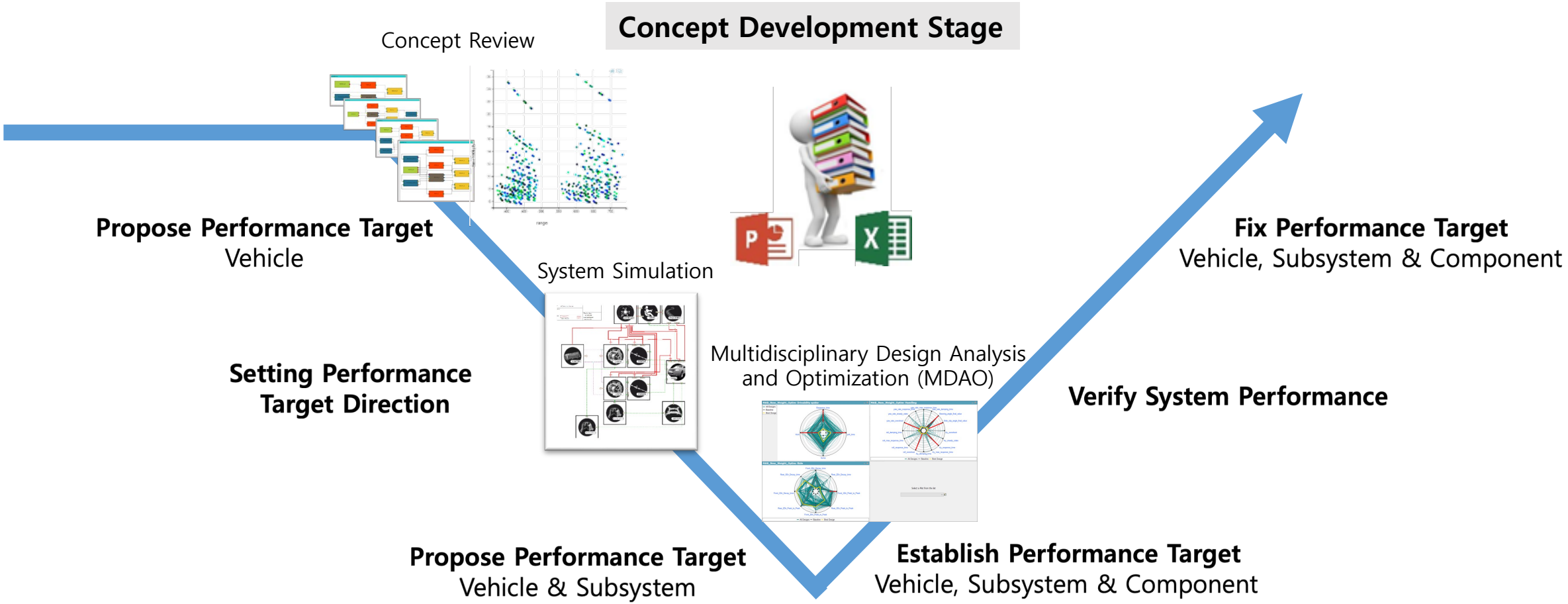
MBSE "business process"
Siloed performance development
Need for a common reference

Capturing knowledge
Legacy module design infrastructure
Legacy design models

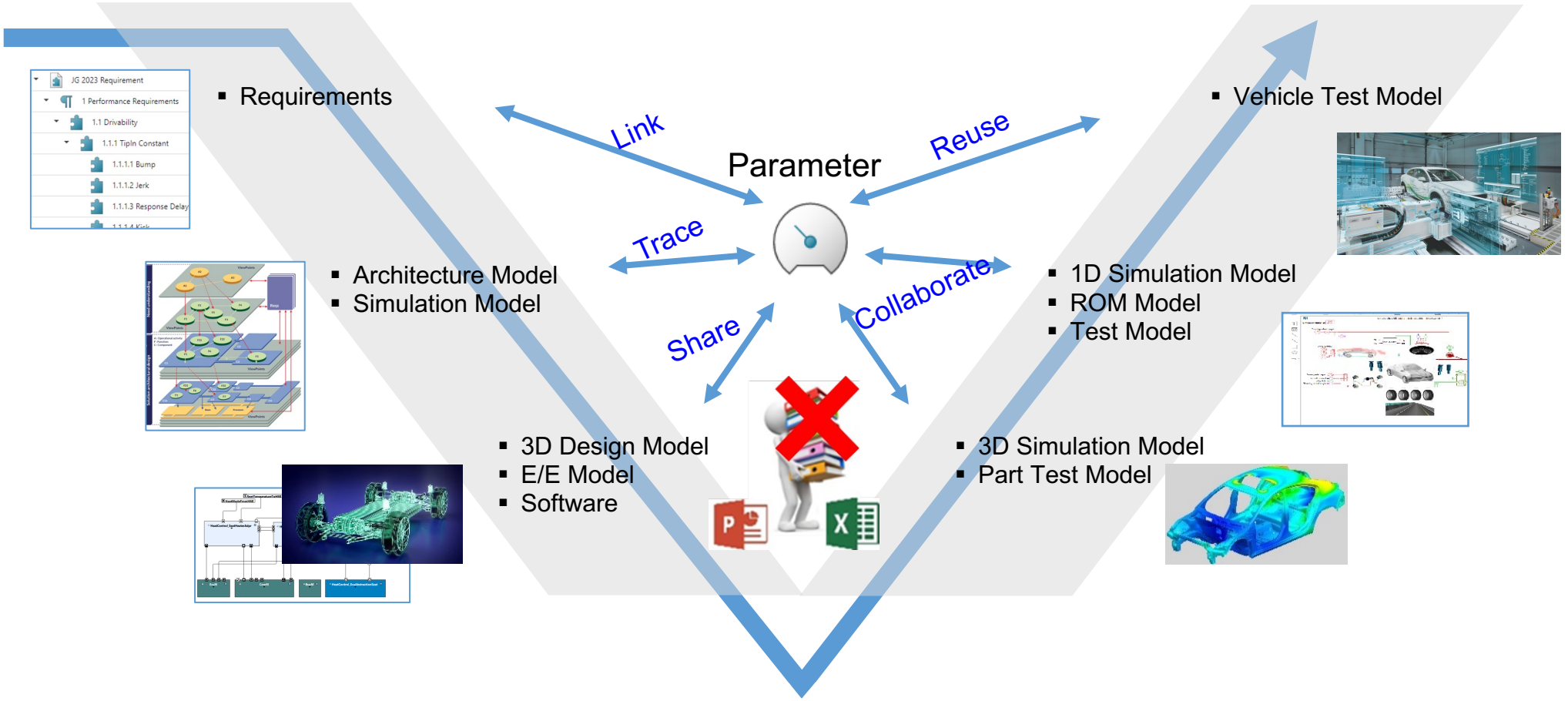
SoS Scalability

MBSE Approach

MBSE Lifecycle (as-is)



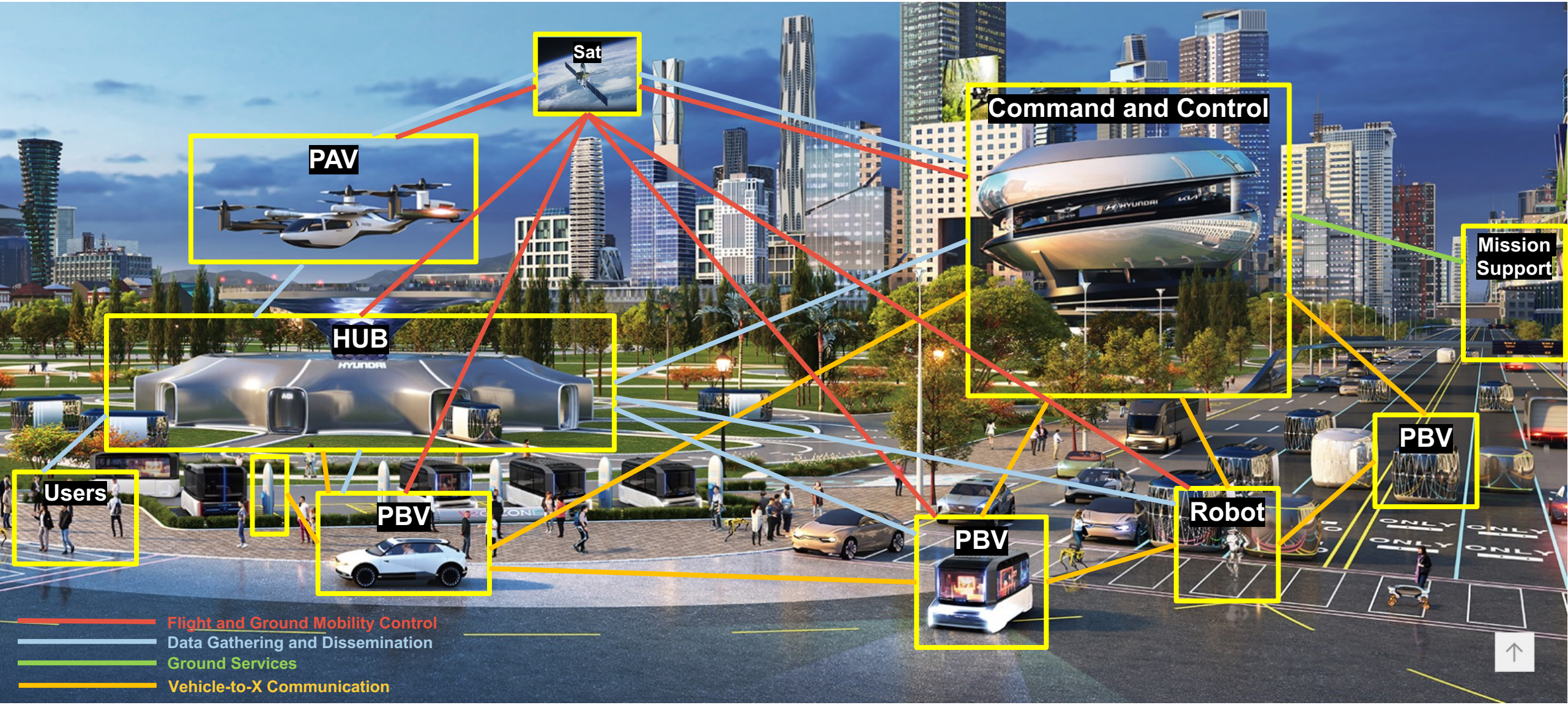
MBSE Lifecycle (to-be)



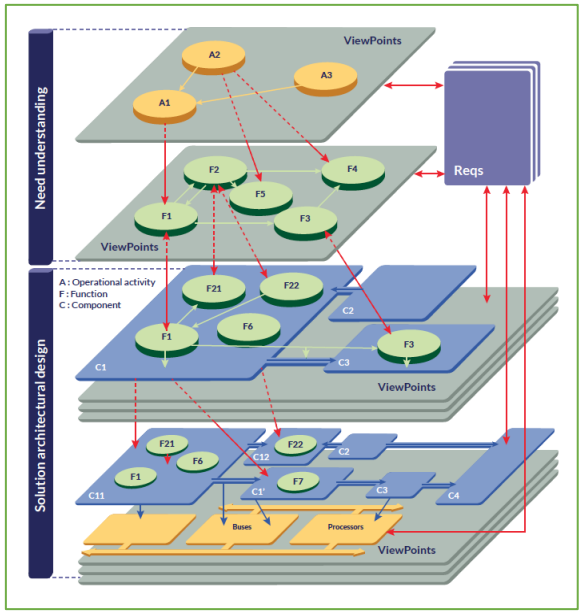
Smart Mobility System of Systems

- SoS challenges on the horizon
- Emergent mobility features
- Significantly higher product line variations
- Faster market response
- Potential for cross-enterprise MBSE

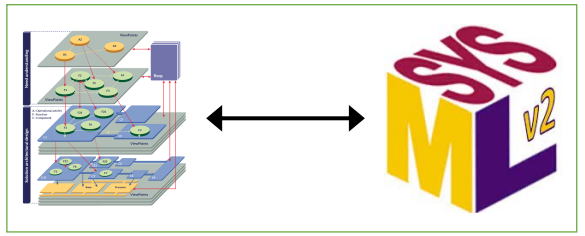

HYUNDAI
HYUNDAI
ONLY
ONLY
ONLY
ONLY


Sat
PAV
Command and Control
Mission Support
HUB
Users
PBV
PBV
Robot
PBV
Flight and Ground Mobility Control
Data Gathering and Dissemination
Ground Services
Vehicle-to-X Communication

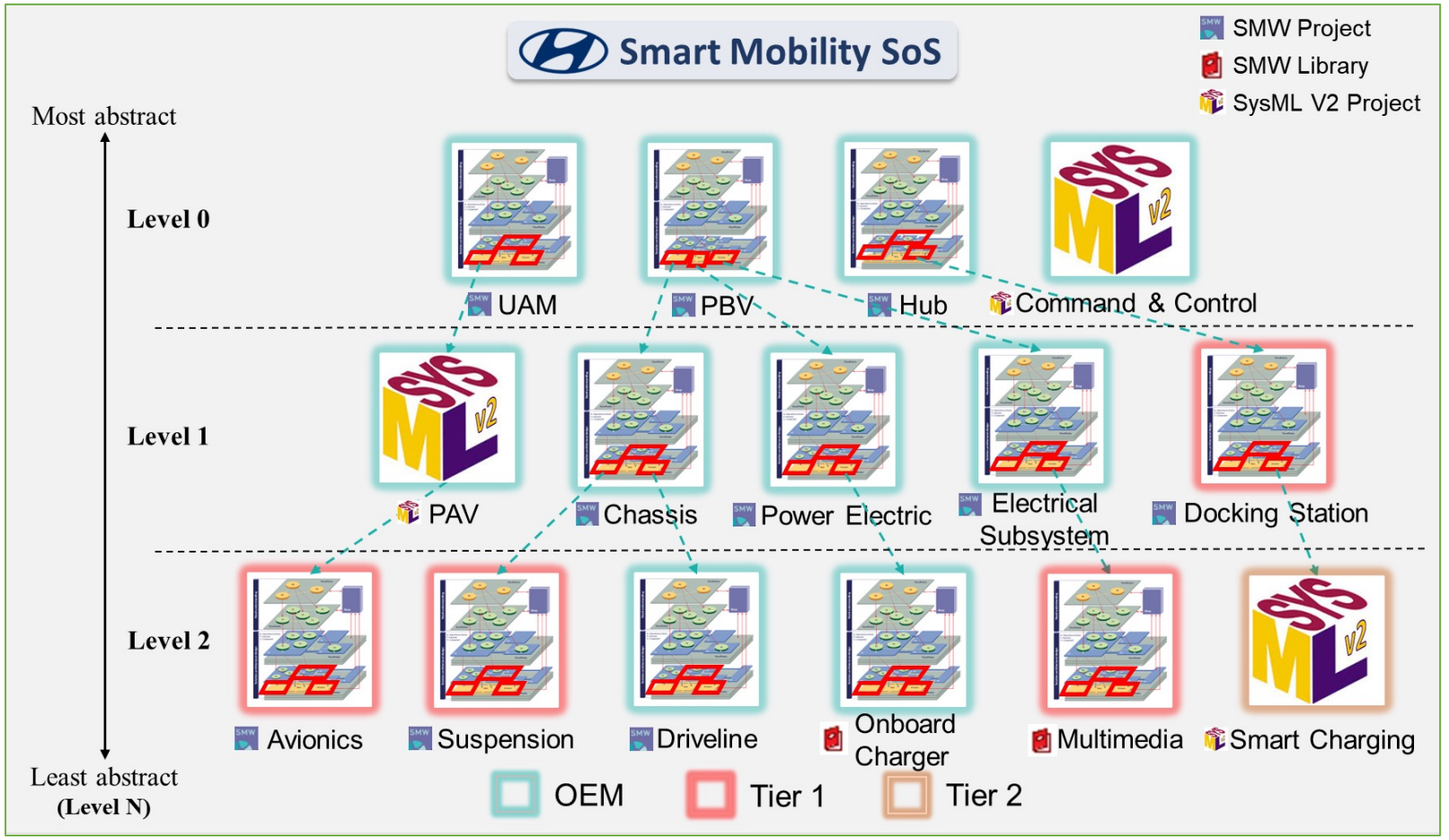
Modularization in MBSE (Vision)



The Arcadia method



Interoperability



Smart Mobility SoS model organization vision

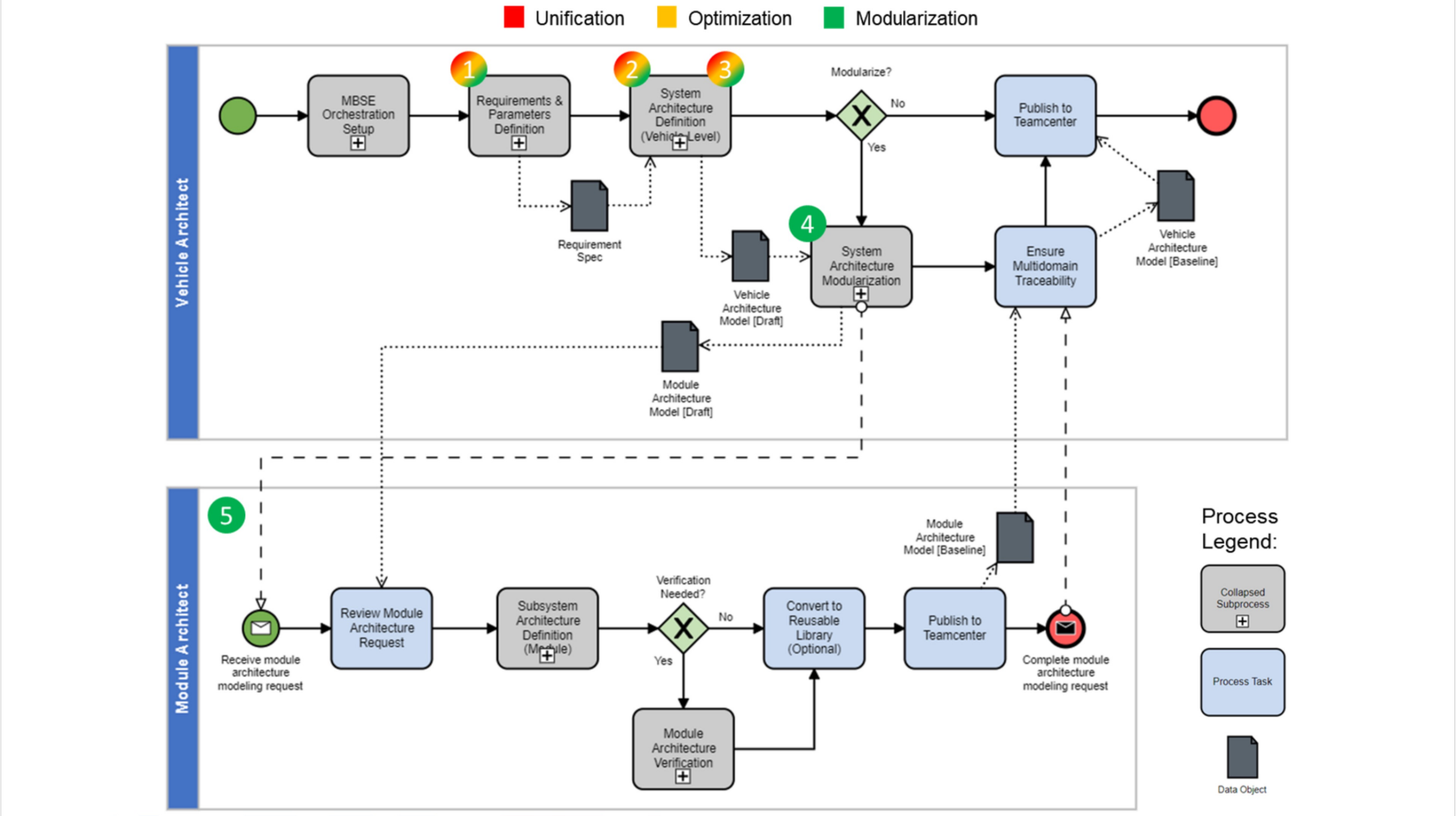
Project Objectives

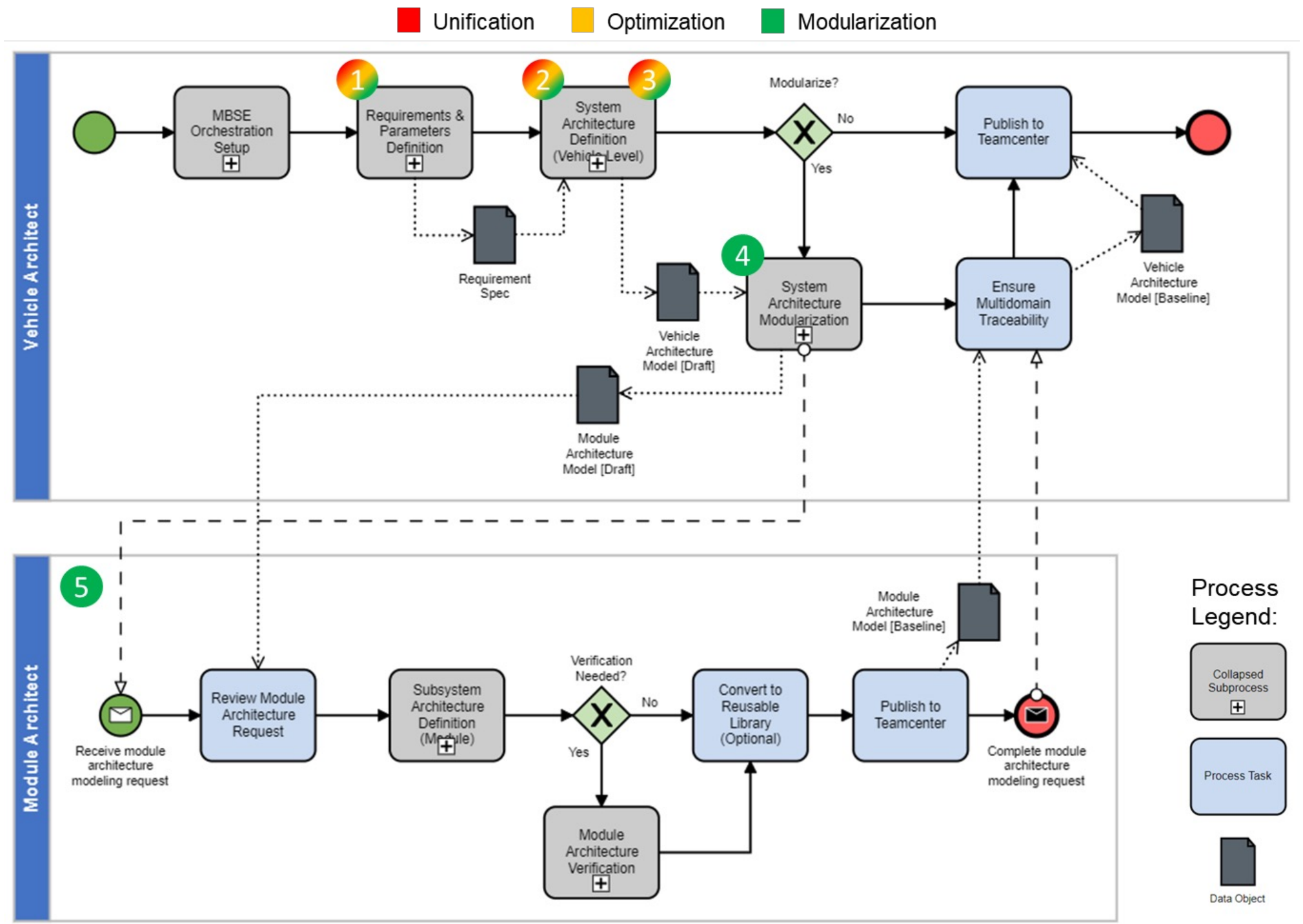
- **Unification**: Unify electric vehicle concept architecture models from the previous two PoCs and model additional operational capabilities required by the smart mobility SoS,
- **Optimization**: Optimize vehicle performance architecture for *drivability*, *ride comfort*, and *handling* and,
- **Modularization**: Modularize levels 0 and 1 of the smart mobility SoS hierarchy.

Business Process

Modular MBSE

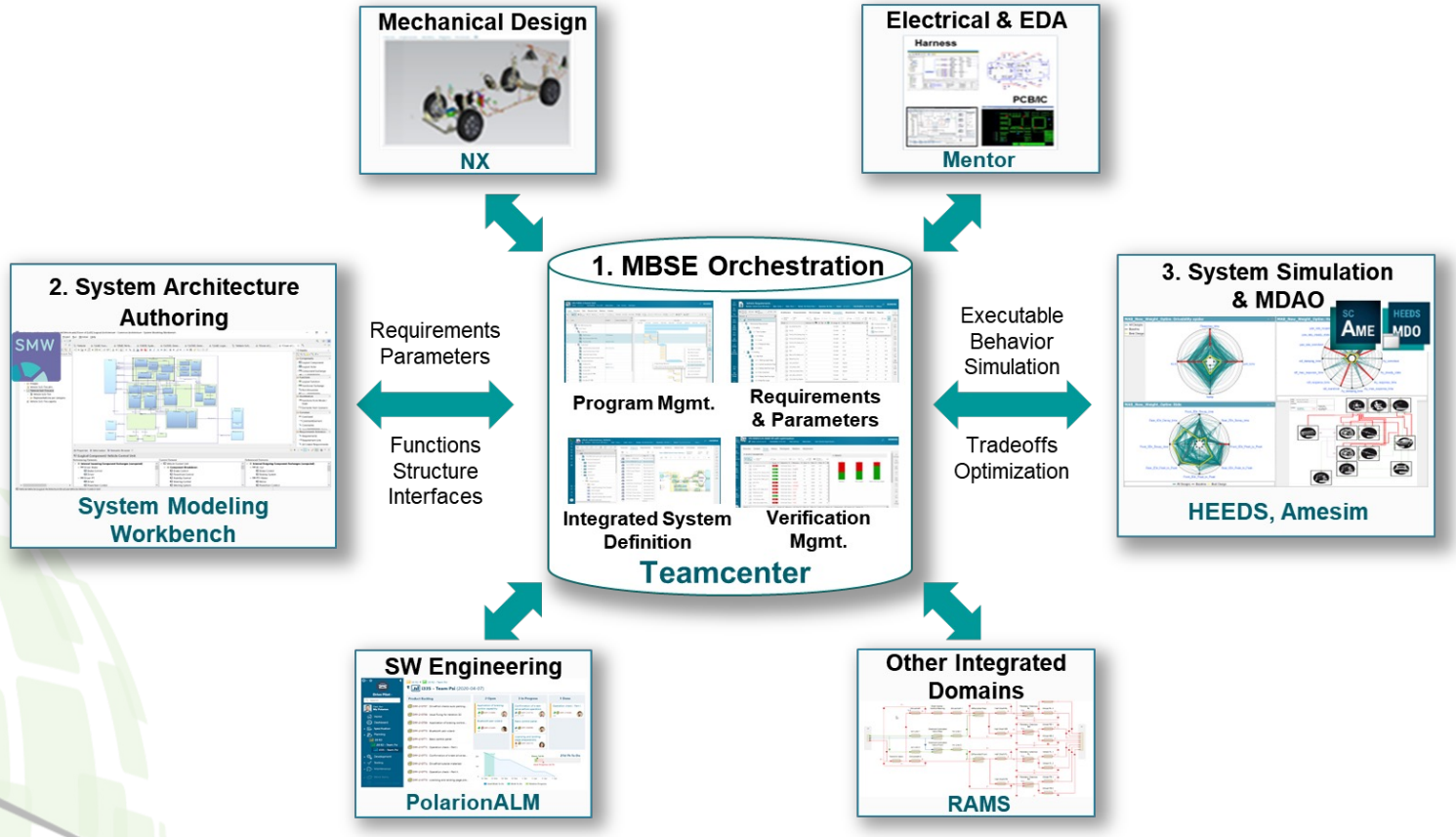
Business Process



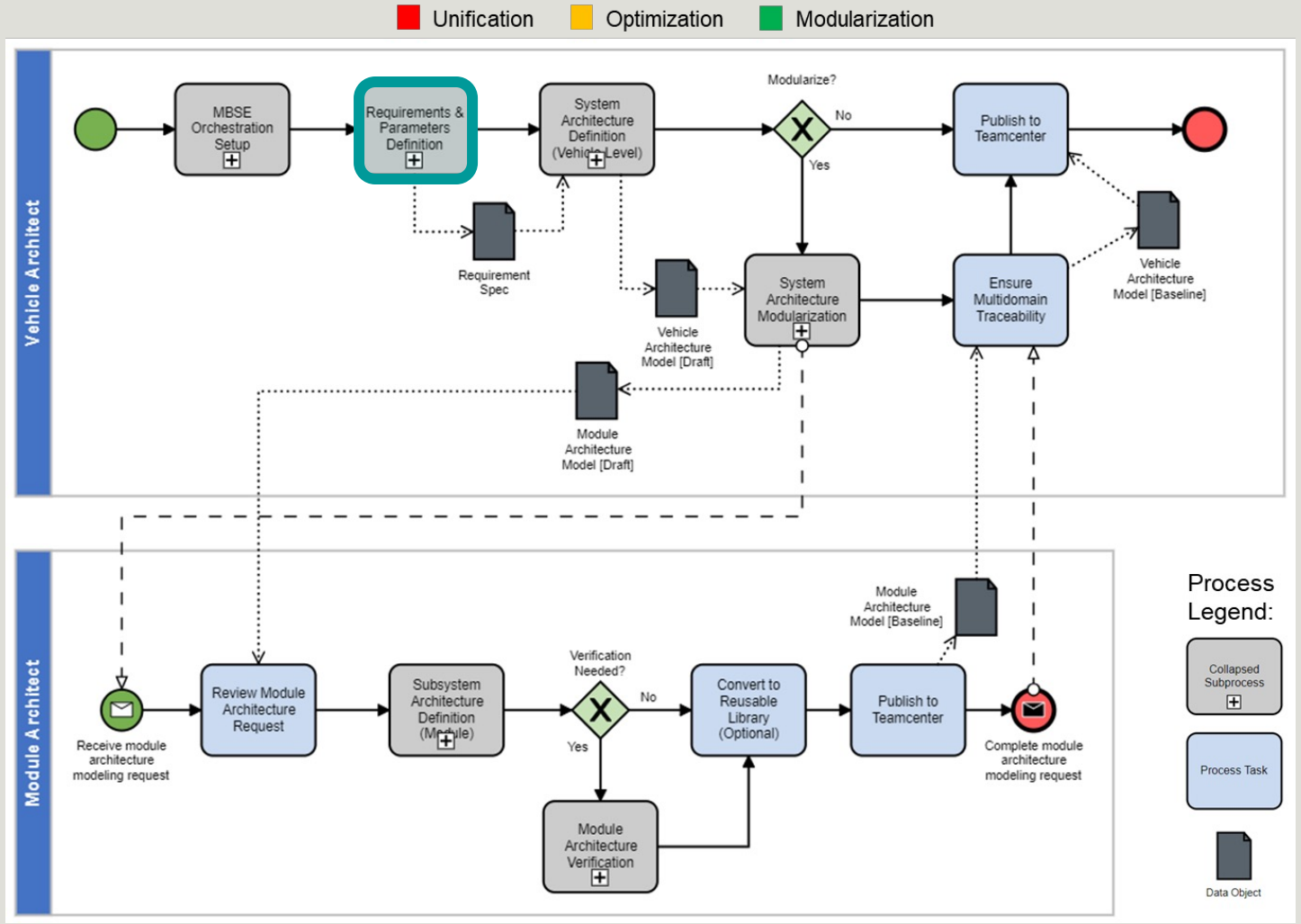

Unification
Optimization
Modularization
Vehicle Architect
MBSE Orchestration Setup
Requirements & Parameters Definition
System Architecture Definition (Vehicle Level)
Modularize?
No
Yes
Publish to Teamcenter
Requirement Spec
Vehicle Architecture Model [Draft]
System Architecture Modularization
Ensure Multidomain Traceability
Vehicle Architecture Model [Baseline]
Module Architecture Model [Draft]
Module Architect
Receive module architecture modeling request
Review Module Architecture Request
Subsystem Architecture Definition (Module)
Verification Needed?
No
Yes
Convert to Reusable Library (Optional)
Module Architecture Verification
Module Architecture Model [Baseline]
Publish to Teamcenter
Complete module architecture modeling request
Process Legend:
Collapsed Subprocess
Process Task
Data Object

Process Tools

1. **MBSE Orchestration:** Teamcenter
2. **System Architecture Authoring**: System Modeling Workbench
3. **Multidisciplinary Design, Analysis & Optimization**: Simcenter HEEDS, Amesim



1 Requirements and Parameters Definition



Inputs:

- SoS vision, problem / opportunity statements, strategic targets
- MBSE orchestration environment readiness

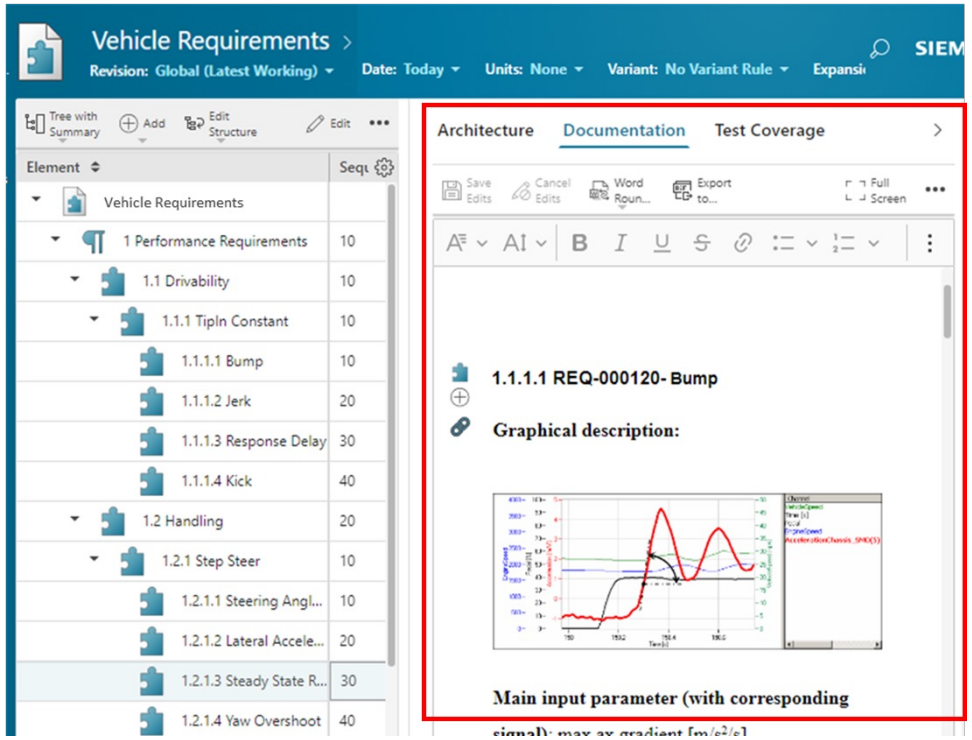
Activities:

- Multi-stakeholder requirement definition
- System / component requirement definition
- Configurable requirements with linked configurable parameter objects
- Traceability to various stages of development

Outputs:

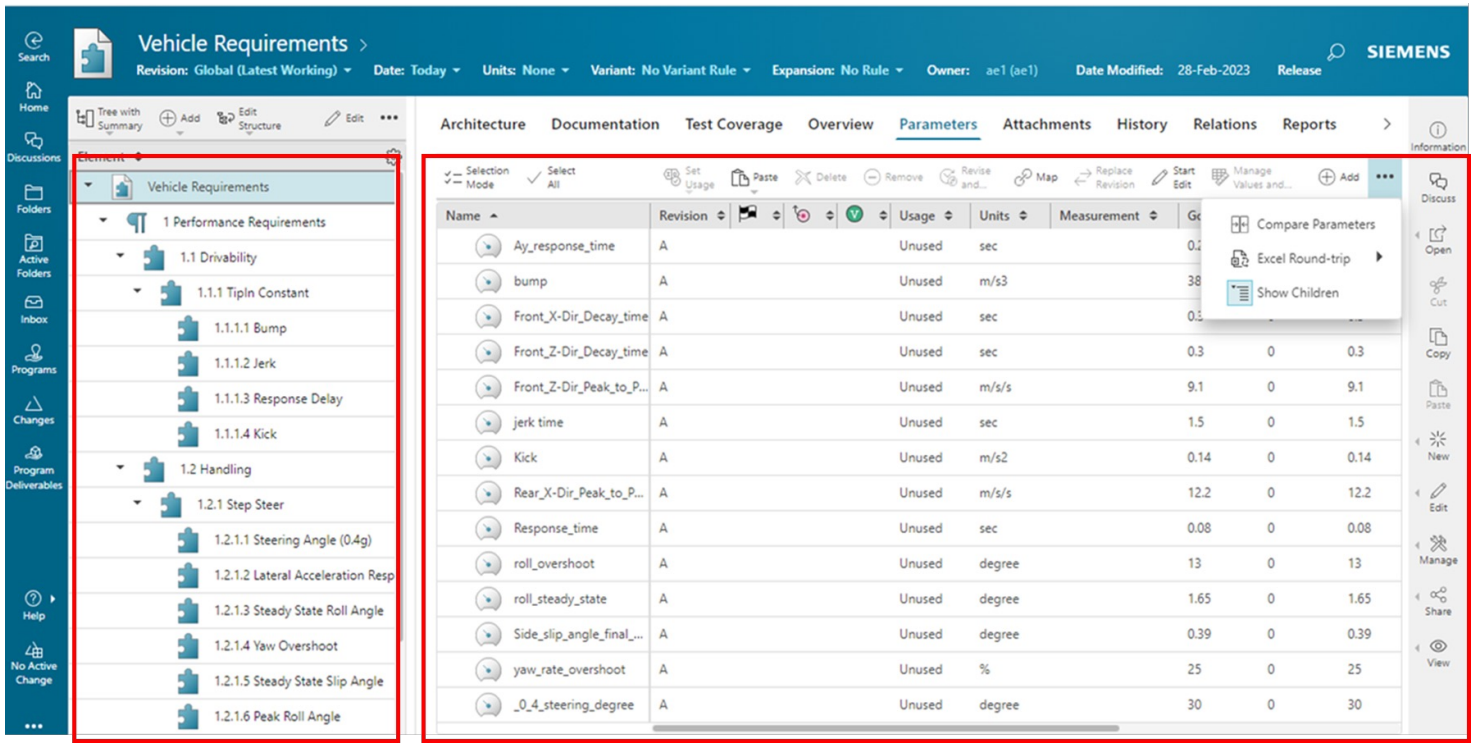
- Multi-level requirements specification
- Parameter definitions
- Requirement text - parameter mapping

Requirements and Parameters Definition



Requirement Specification Tree

Rich-text Description

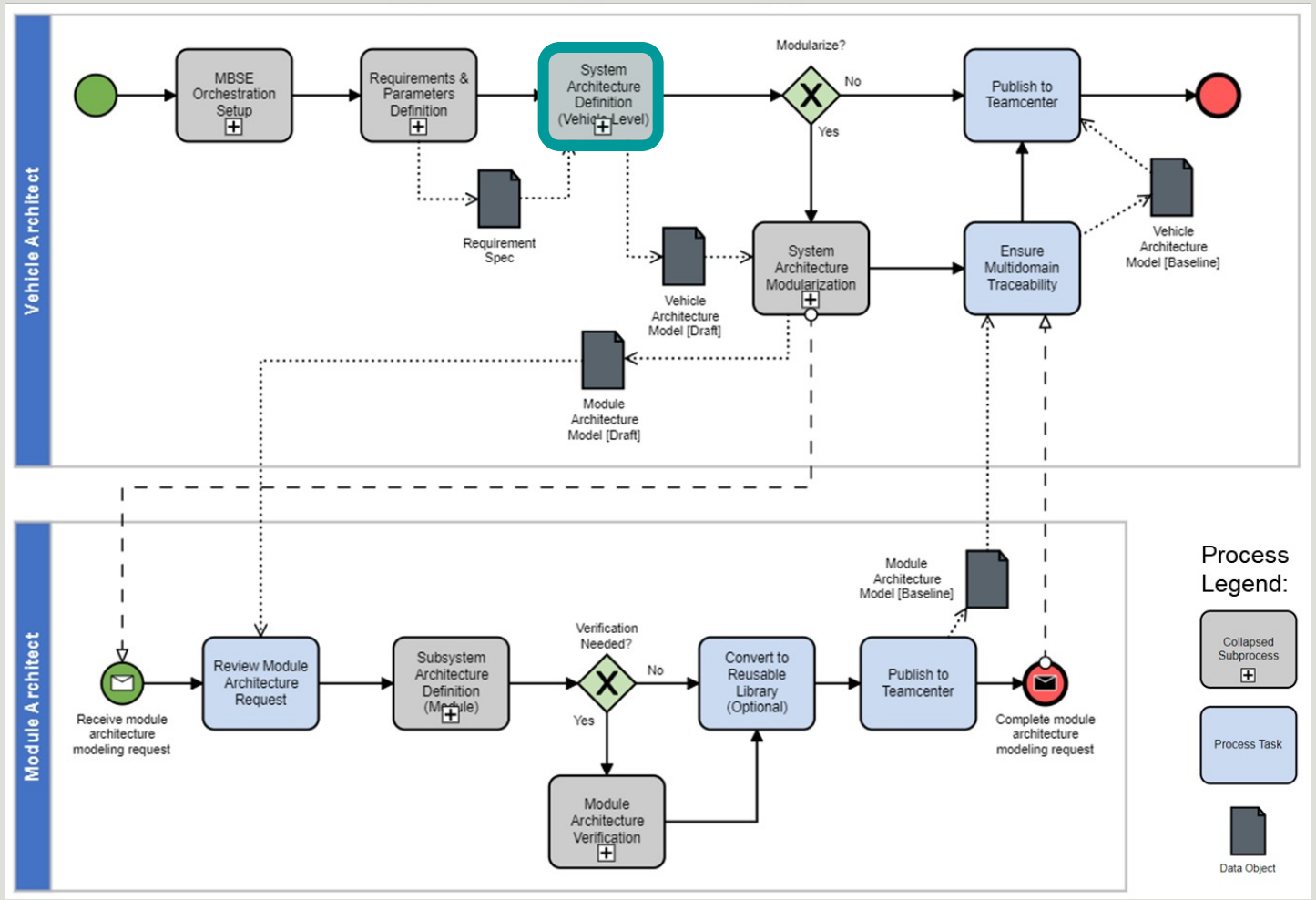


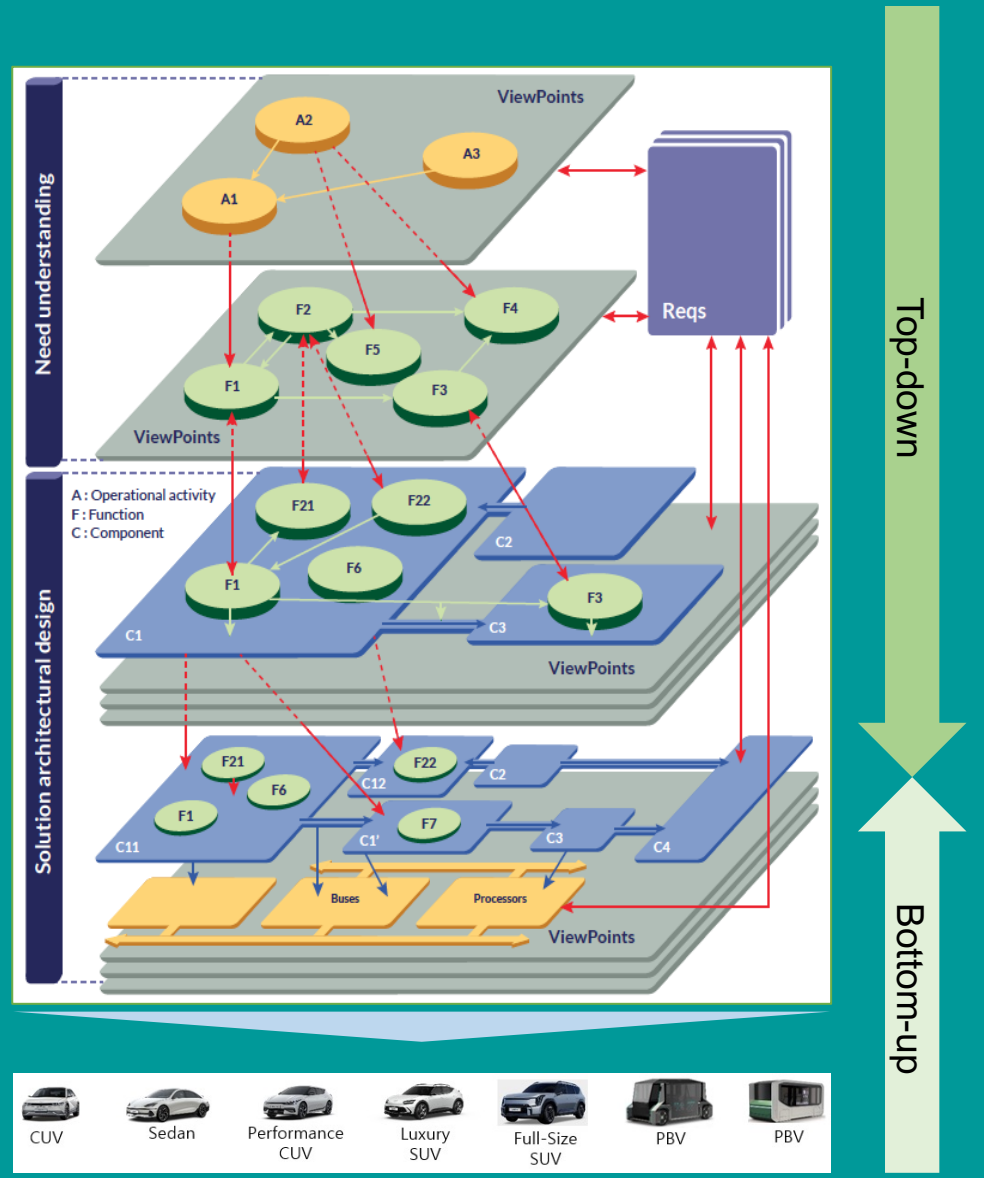
Requirement Specification Tree

Requirement Parameters

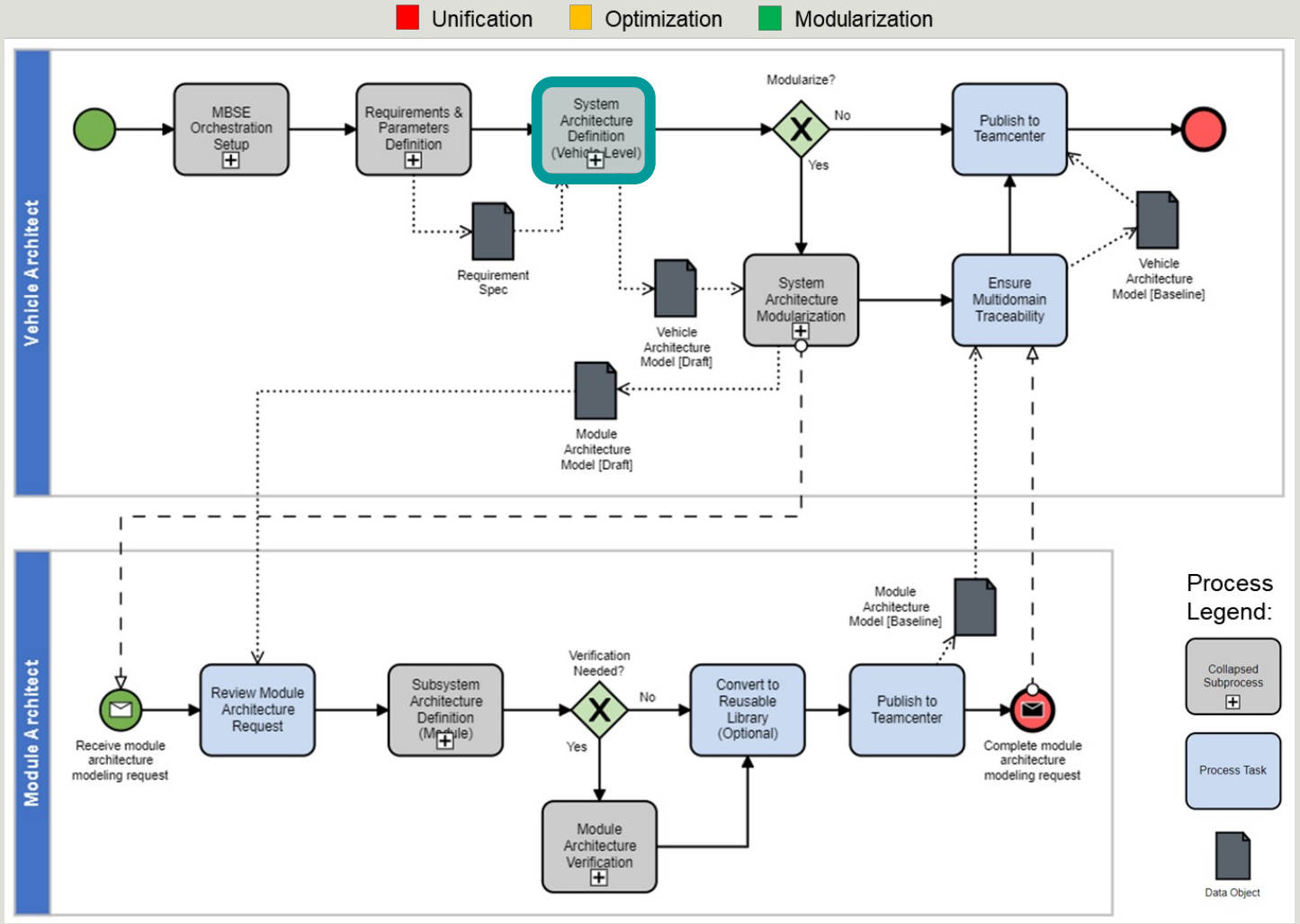
2 System Architecture Definition

Unification Optimization Modularization





2 System Architecture Definition (Operational Analysis)



Inputs:

- SoS needs and constraints
- Preliminary system needs

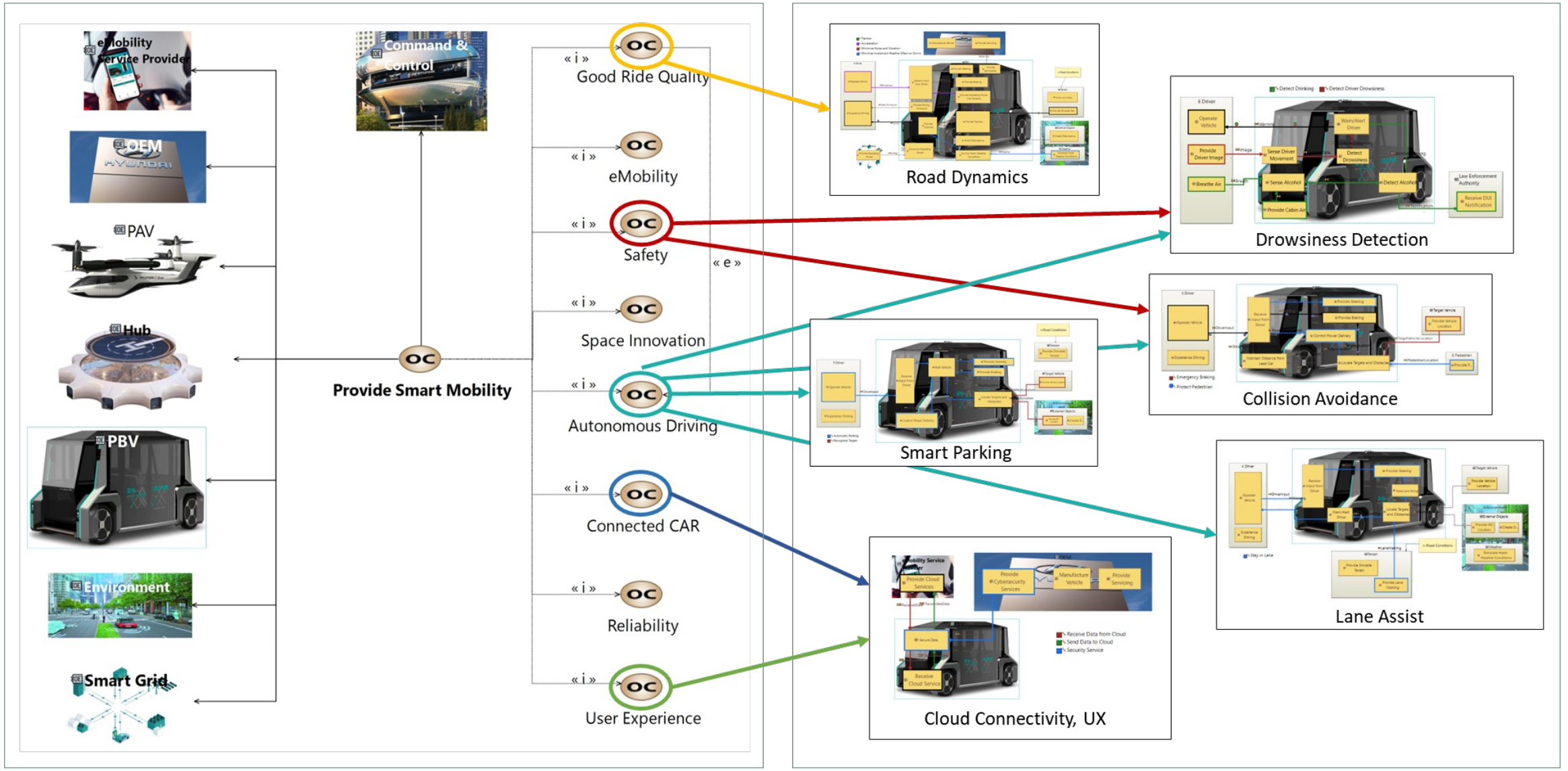
Activities:

- Model SoS mission capabilities
- Define operational behavior
- Define operational architecture
- Refine / allocate stakeholder needs
- Analyze operational model

Outputs:

- SoS interface constraints
- Refined SoS requirements
- Preliminary functional model

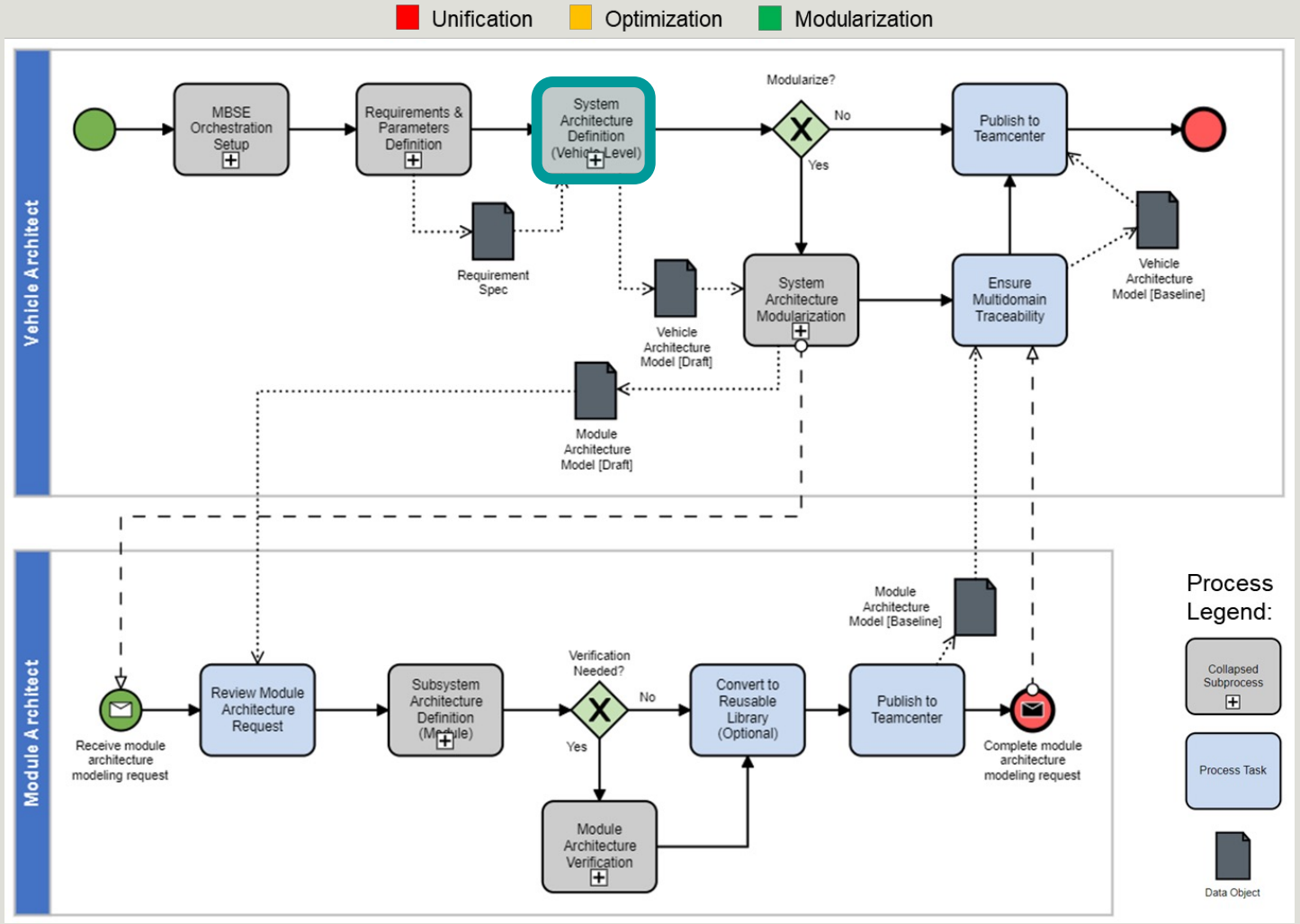
Operational Analysis



SoS Operational Capabilities

SoS Operational Scenarios

2 System Architecture Definition (System Analysis)



Inputs:

- Stakeholder requirements spec
- Operational architecture
- System capabilities

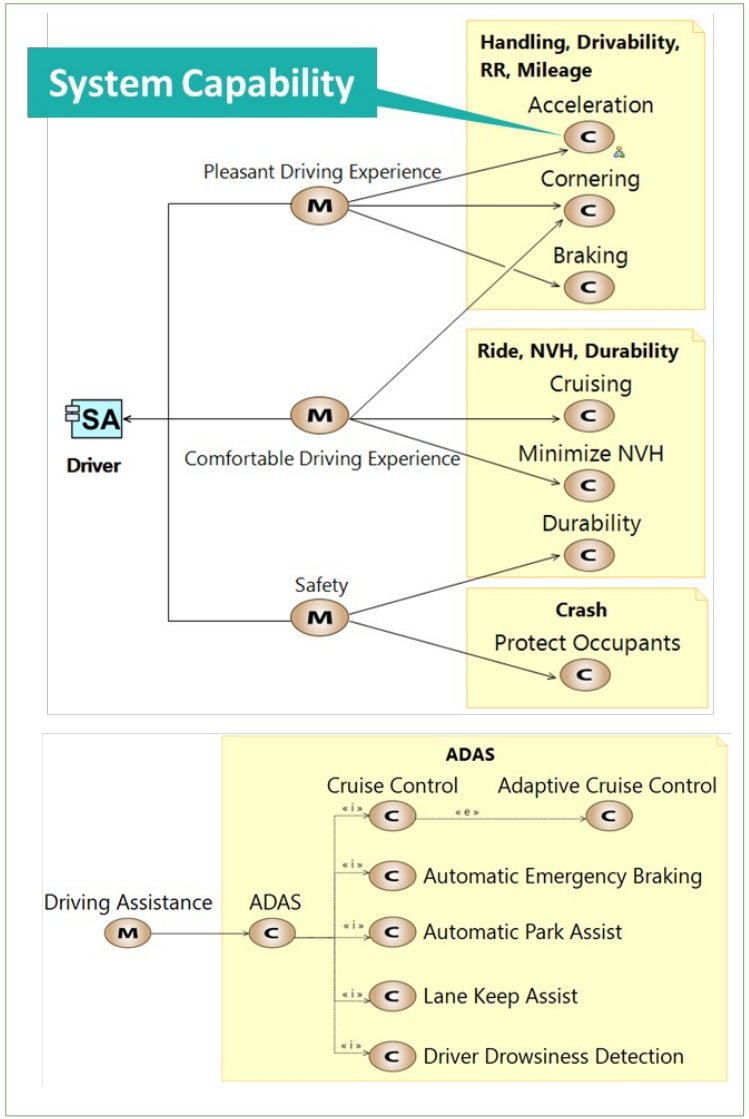
Activities:

- Model system capabilities
- Define and allocate system/actor functions
- Contextualize system-of-interest

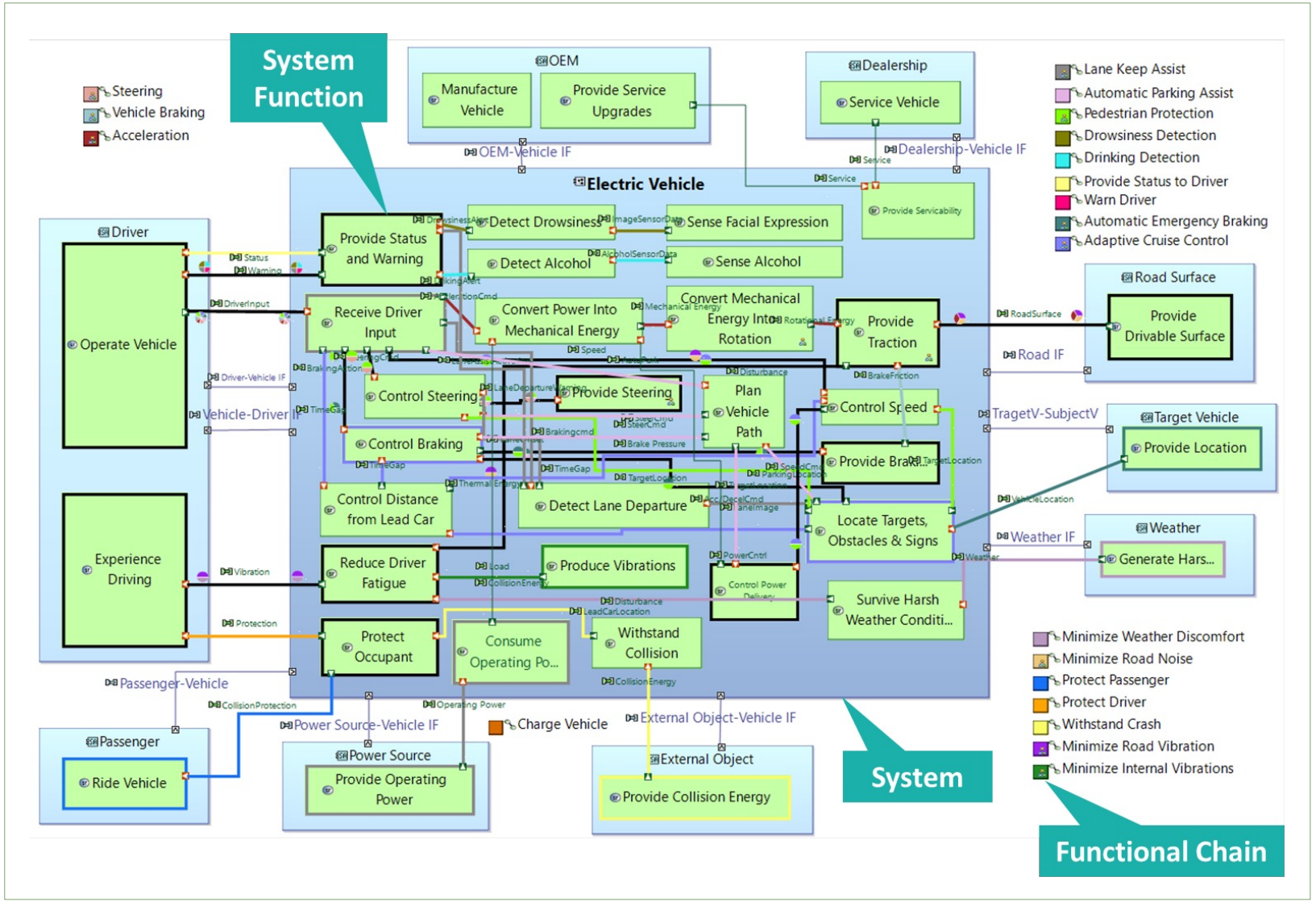
Outputs:

- Blackbox architecture model (configurable)
- System requirements spec
- Functional scenarios spec

System Analysis

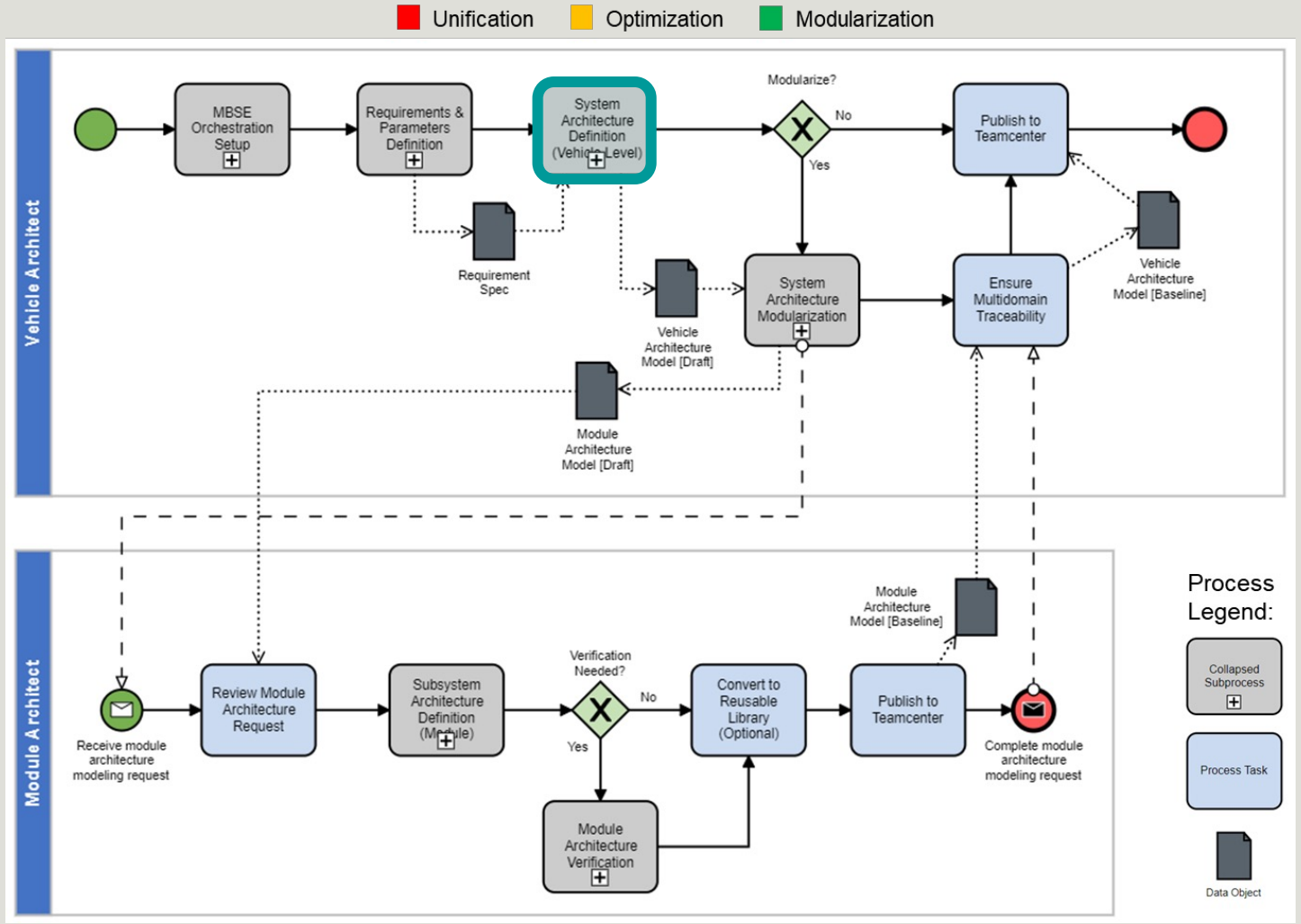


System Capabilities



System Blackbox Architecture (Integrated View)

2 System Architecture Definition (Logical Architecture)



Inputs:

- Blackbox system architecture
- Vehicle performance requirements

Activities:

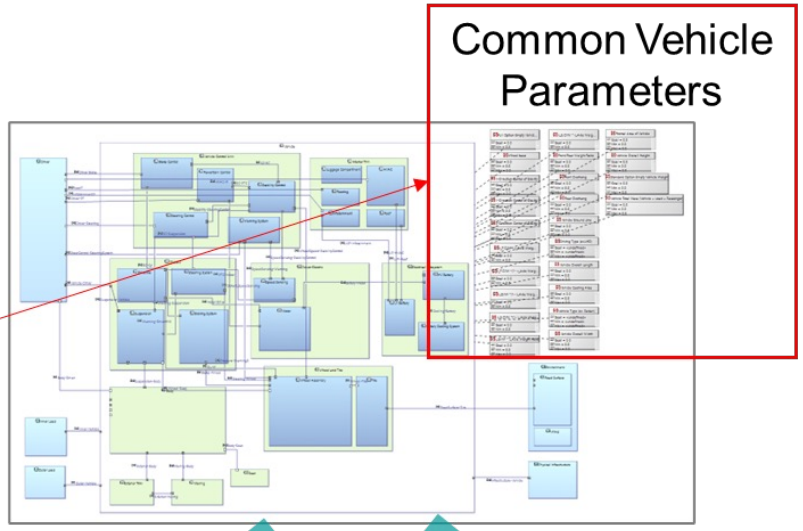
- Decompose logical system
- Allocate functions to logical components
- Define internal interfaces (white box)
- Generate stakeholder concerns (views)
- Requirement / parameter traceability

Outputs:

- Logical architecture model (configurable)
- Performance allocation & traceability
- Performance verification views

Logical Architecture

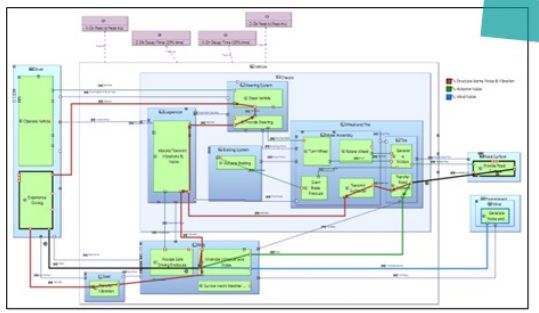
- Common Parameters
  - Driving Type (ex.LHD)
  - Vehicle Overall Height
  - Front Overhang
  - Wheel base
  - Y-Direction Center of Gravity
  - v_GVW *(1- v_Axle Weight Ratio)
  - Vehicle Overall Length
  - Full Option Empty Vehicle Weight
  - Vehicle Total Mass (Vehicle + Load + Passenger)
  - Standard Option Empty Vehicle Weight
  - Vehicle Ground Line
  - Vehicle Overall Width
  - Front/Rear Weight Ratio
  - Frontal Area of Vehicle
  - Rear Overhang
  - Vehicle Cooling Area
  - v_GVW * v_Axle Weight Ratio
  - v_F.CVW * v_Axle Weight Ratio
  - v_S.CVW * v_Axle Weight Ratio
  - Z-Direction Center of Gravity
  - X-Direction Center of Gravity
  - Vehicle Type (ex. Sedan)
  - v_S.CVW *(1- v_Axle Weight Ratio)
  - v_F.CVW *(1- v_Axle Weight Ratio)

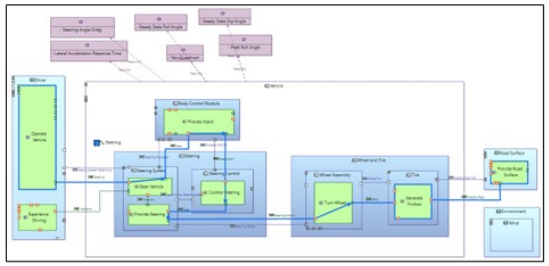


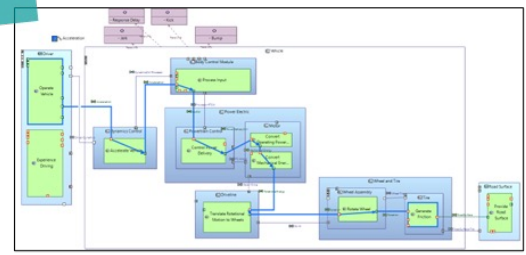
Ride Comfort

Handling

Drivability

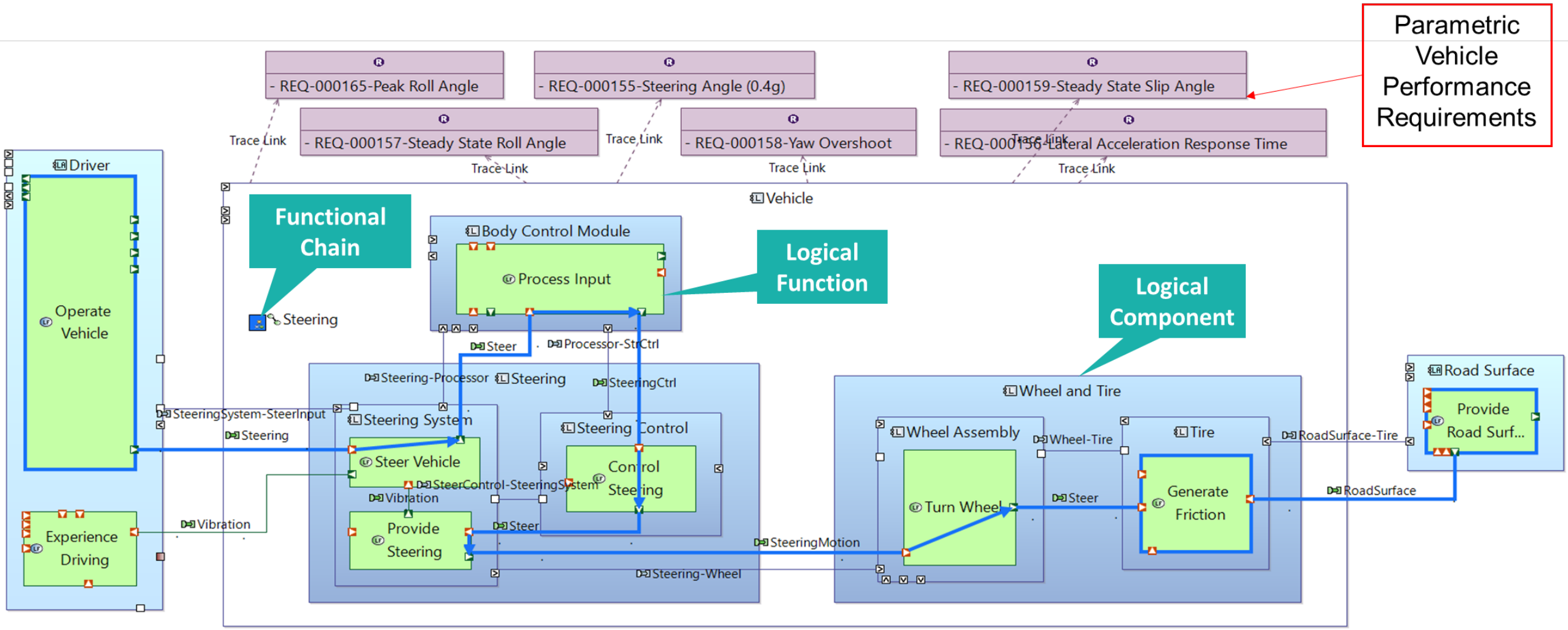






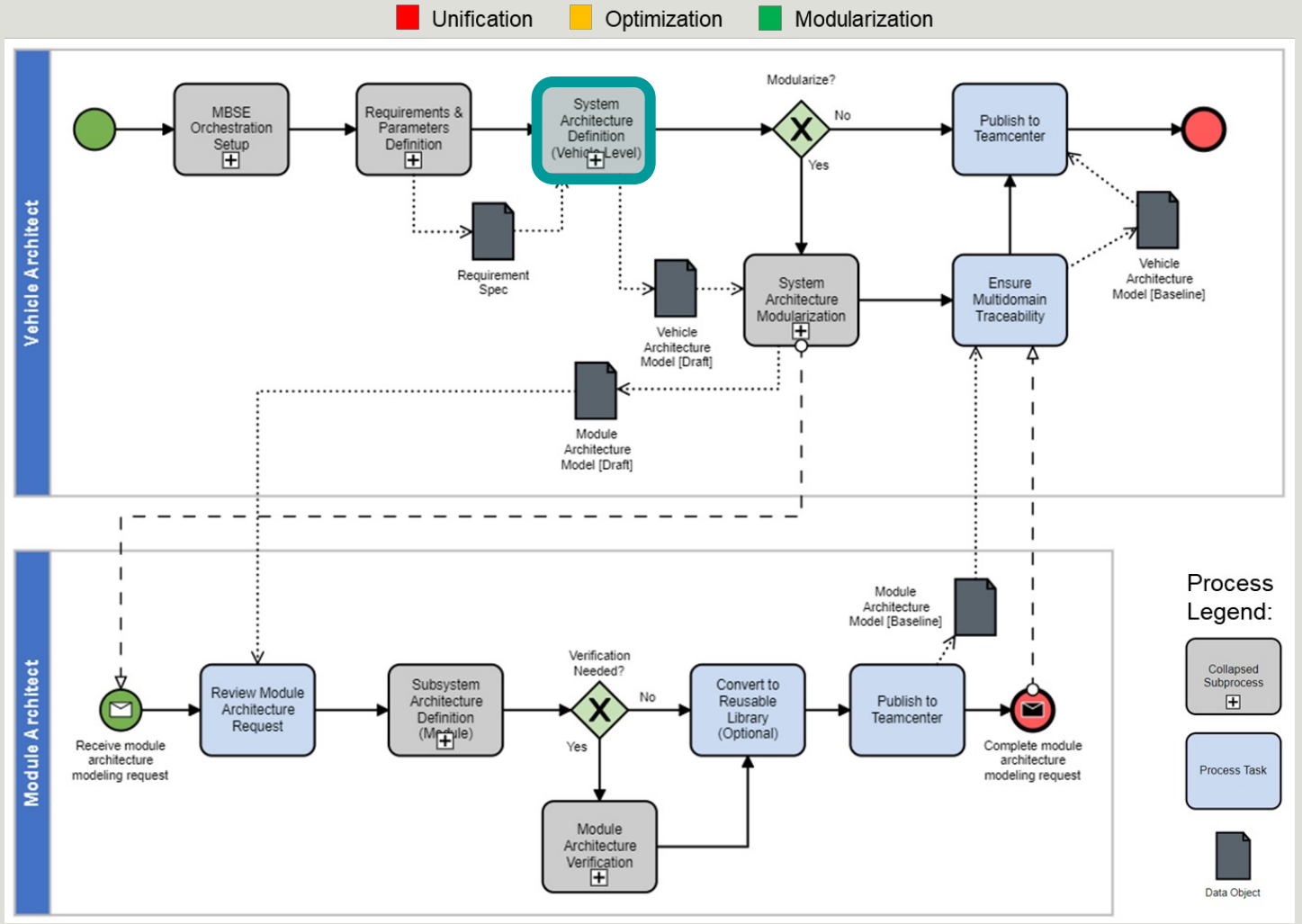
Common Vehicle Parameters, Logical Architecture Views, Traceability

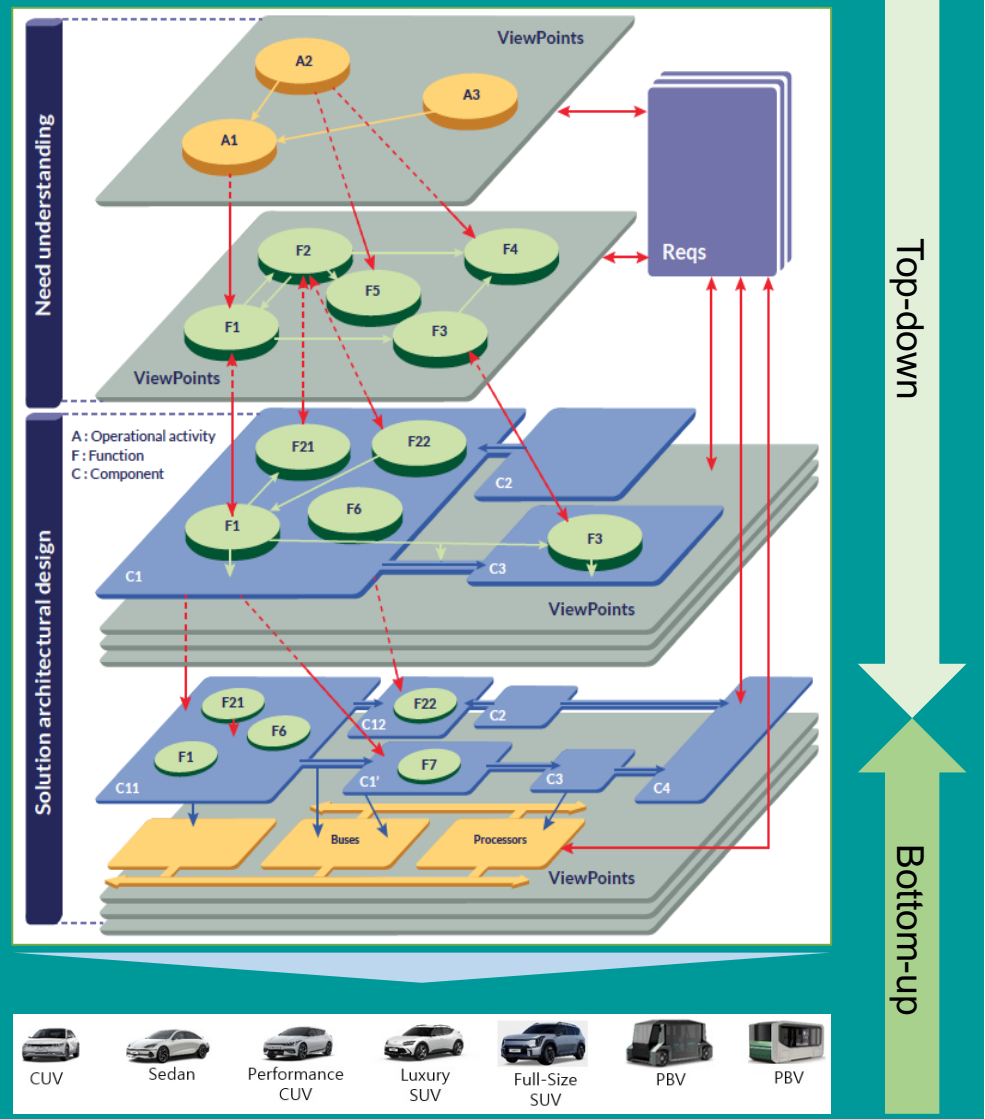
Logical Architecture



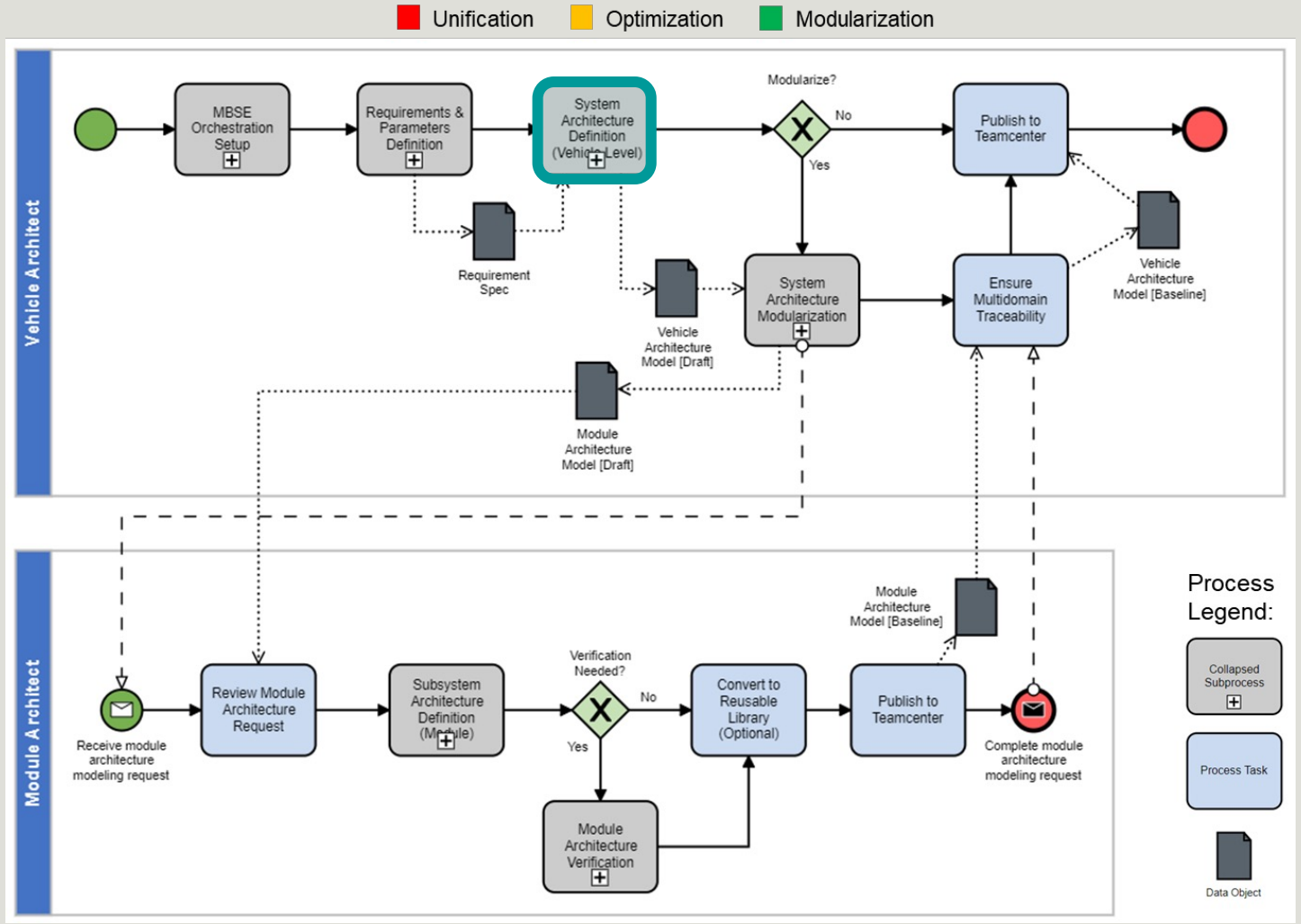
Logical architecture “view” representing handling functionality

3 System Architecture Definition (Physical Architecture)





3 System Architecture Definition (Physical Architecture)



Inputs:

- Logical architecture alternatives
- Performance analysis results
- Existing product BOM

Activities:

- Allocate logical components to physical components (from existing BOM)
- Model physical interfaces
- Orchestrate MDAO
- Evaluate logical alternatives and fix technology choices (tradeoffs, opt.)
- Trace component design requirements and parameters to architecture elements

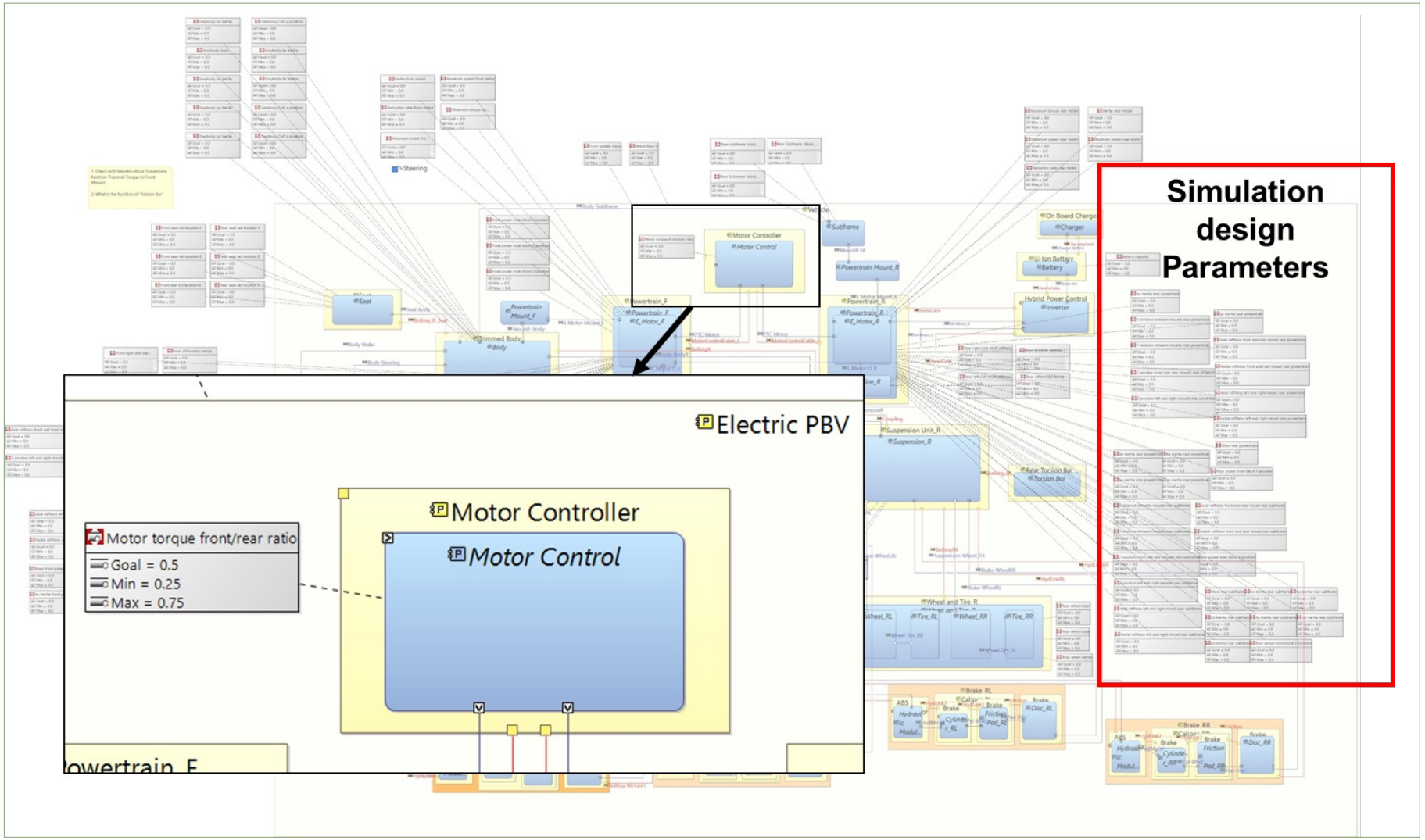
Outputs:

- Physical architecture model (configurable)
- Product breakdown structure / preliminary EBOM
- Preliminary component design specification

Physical Architecture

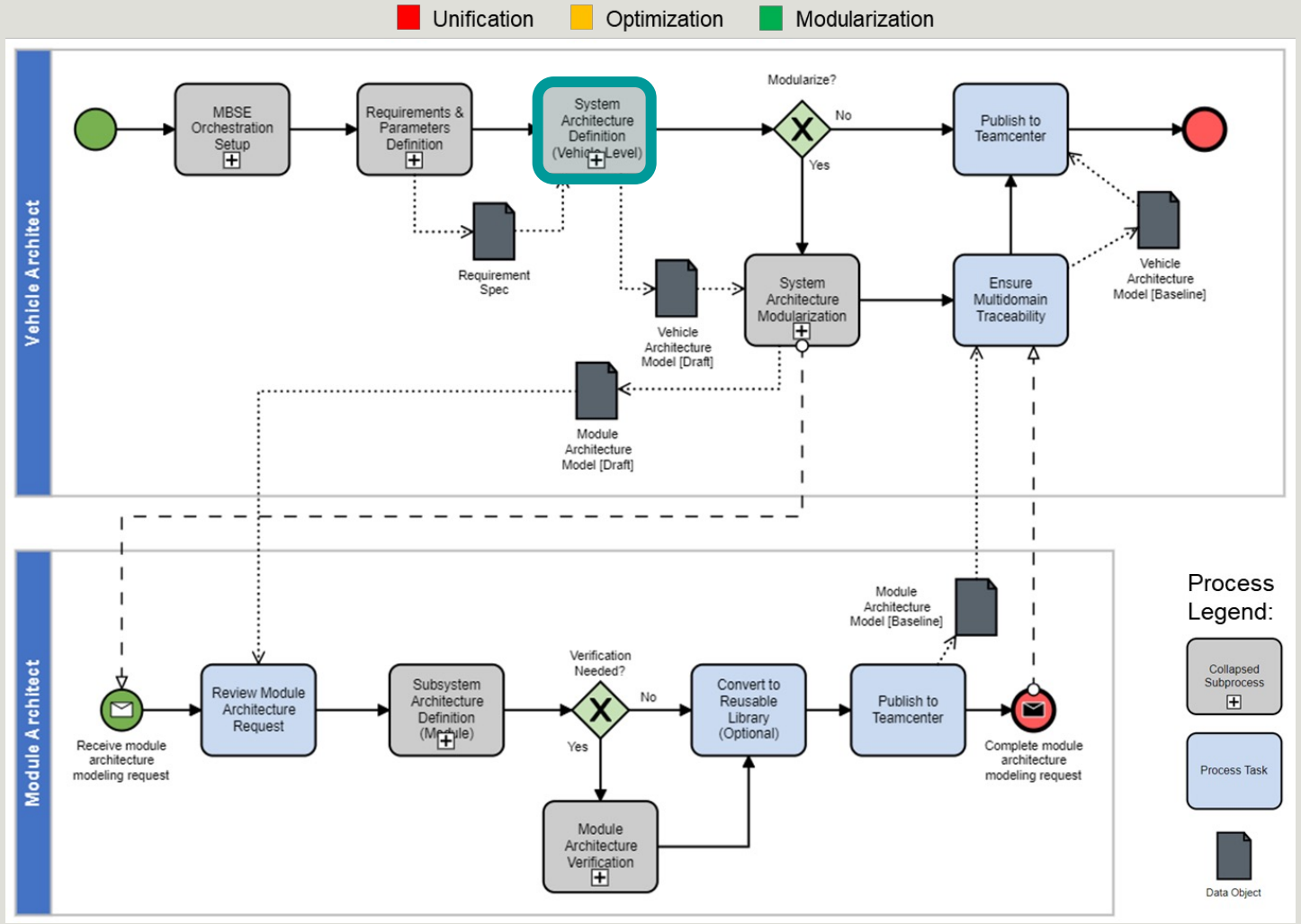
- Physical Architecture
  - Physical Functions
  - Capabilities
  - Interfaces
  - Data
  - Structure
    - Physical System
      - Vehicle Structure-Lv0
        - Body Structure-Lv1
        - Seat-Lv1
        - Power Electric-Lv1
        - Chassis-Lv1
          - General Code = G00000
          - Suspension Accessory-Lv2
          - Suspension Axle-Lv2
            - General Code = GB0000
            - Wheel and Tire-Lv3
              - General Code = GB1000
              - Wheel and Tire_R
              - Wheel and Tire_F
            - Front Axle-Lv3
            - Rear Axle-Lv3
          - Front Suspension-Lv2
          - Rear Suspension-Lv2
          - Air Suspension-Lv2
          - Muffler-Lv2
          - Brake-Lv2
          - Fuel-Lv2
          - MTG-Lv2
          - Driveline-Lv2
          - Control-Lv2
          - Cooling-Lv2
          - Steering-Lv2

Physical system breakdown



Performance simulation “view” with linked simulation parameters

3 System Architecture Definition (Performance Verification)



Inputs:

- Physical architecture model for perf. sim
- Performance verification requests
- Reusable multiphysics simulation models and libraries

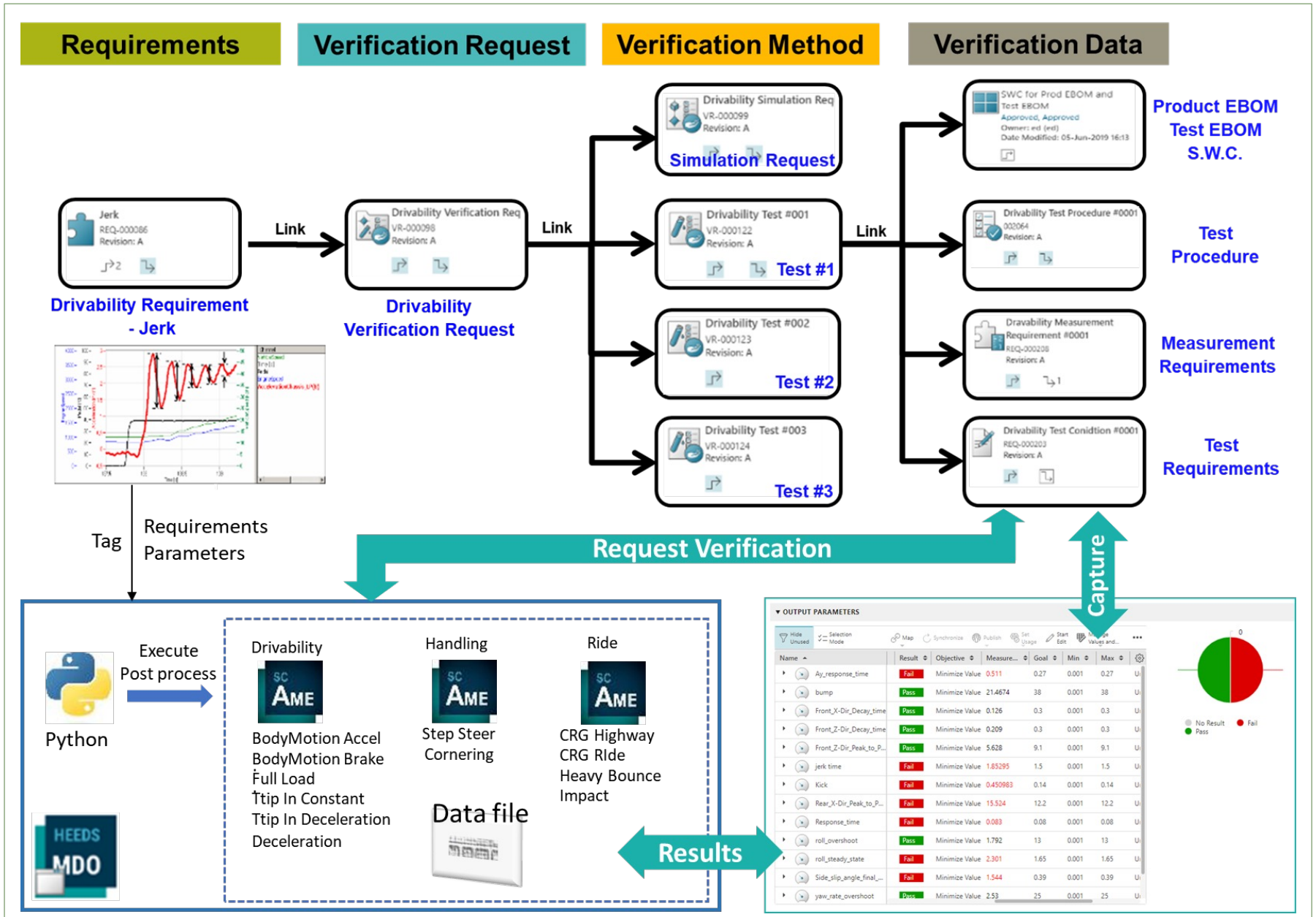
Activities:

- Initiate performance verification requests
- Tradeoff studies, optimization
- Submit verification request results
- Fix technology design parameters and baseline physical architecture
- Share granular vehicle architecture model to the LM repository (integrated system definition)

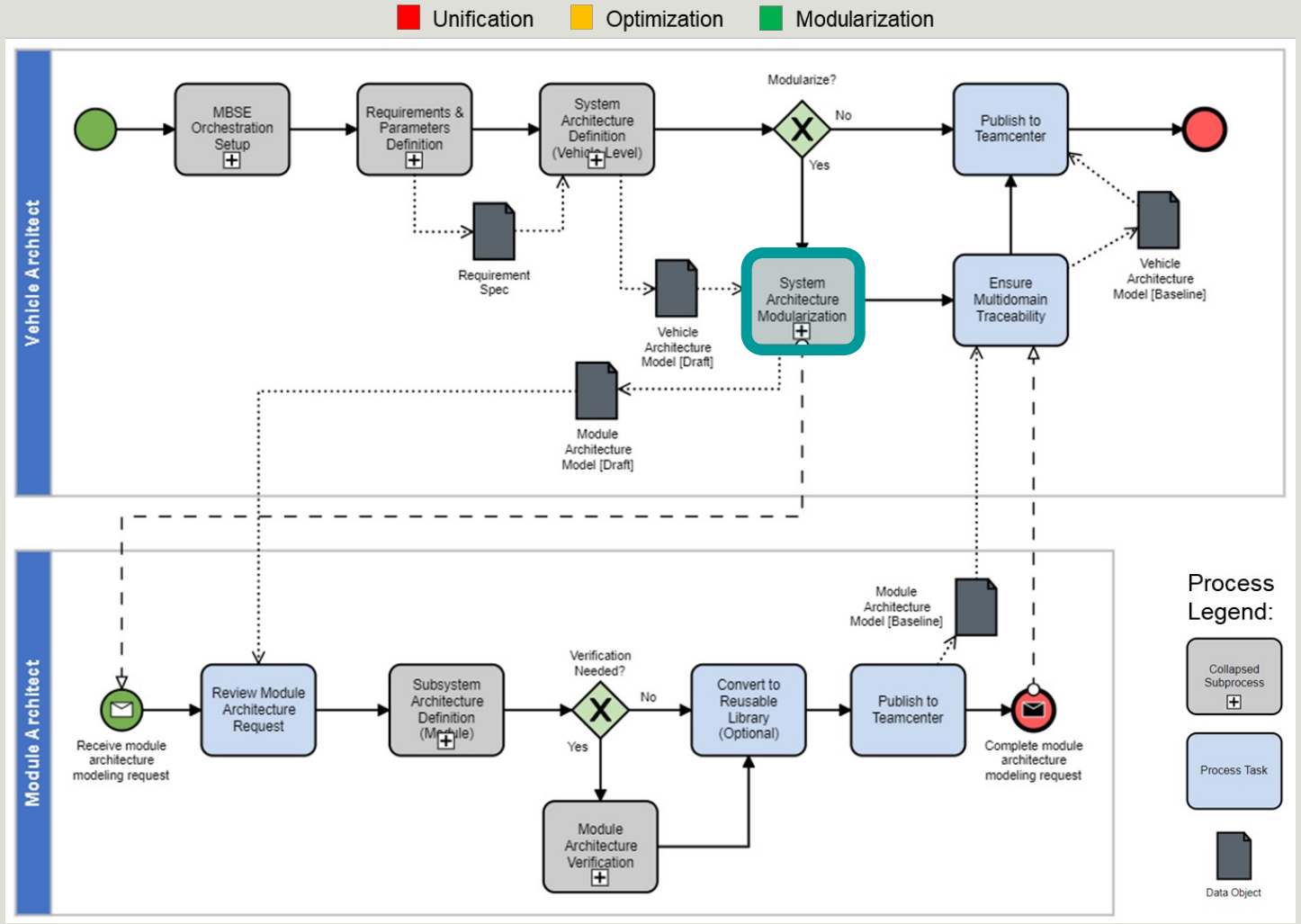
Outputs:

- Baselined vehicle architecture model (configurable)
- Product breakdown structure / preliminary EBOM
- Preliminary component design specification

Performance Verification



4 Architecture Modularization

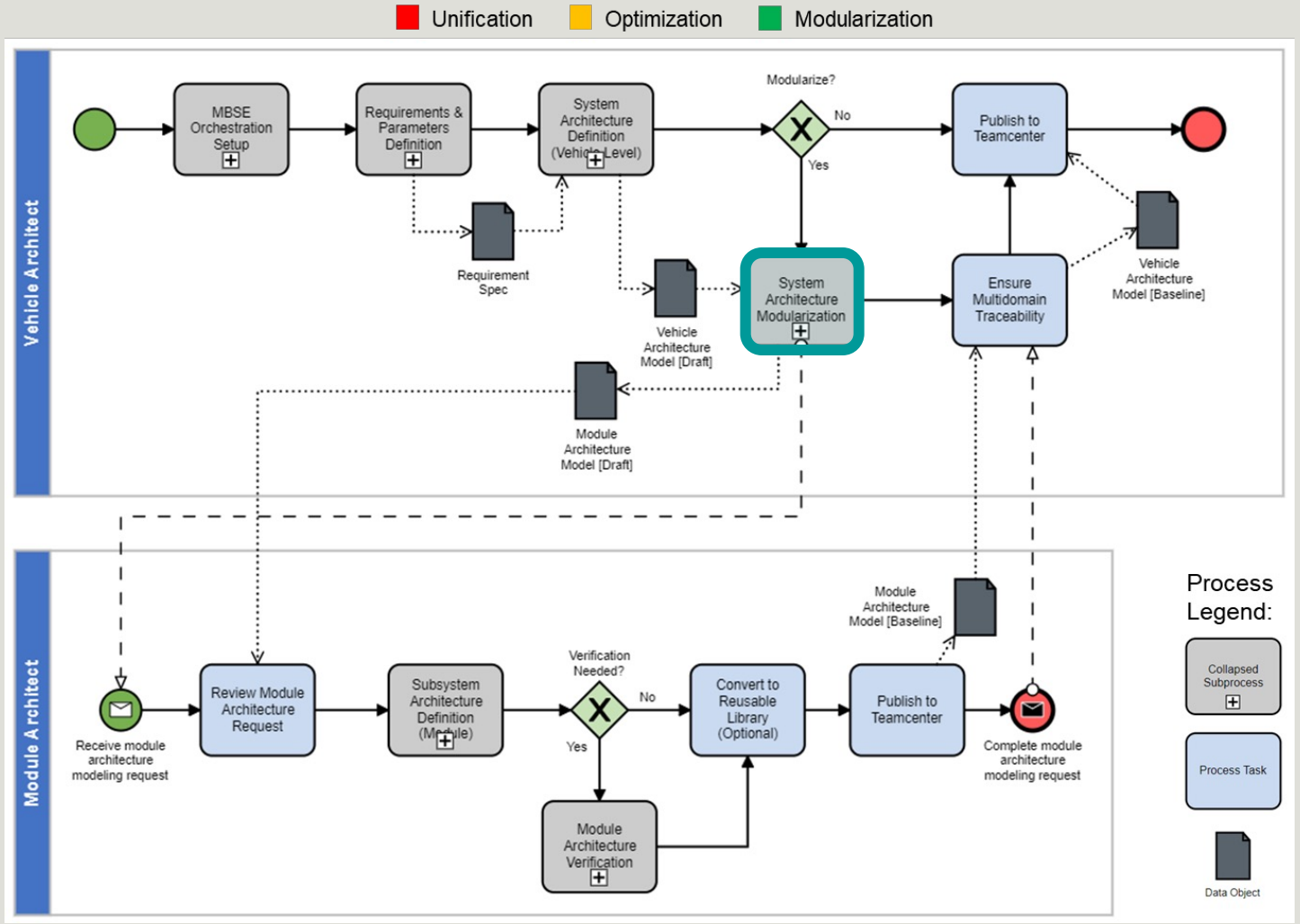


Several criteria pertaining to the development teams responsible for architecture definition are crucial for achieving modularity.

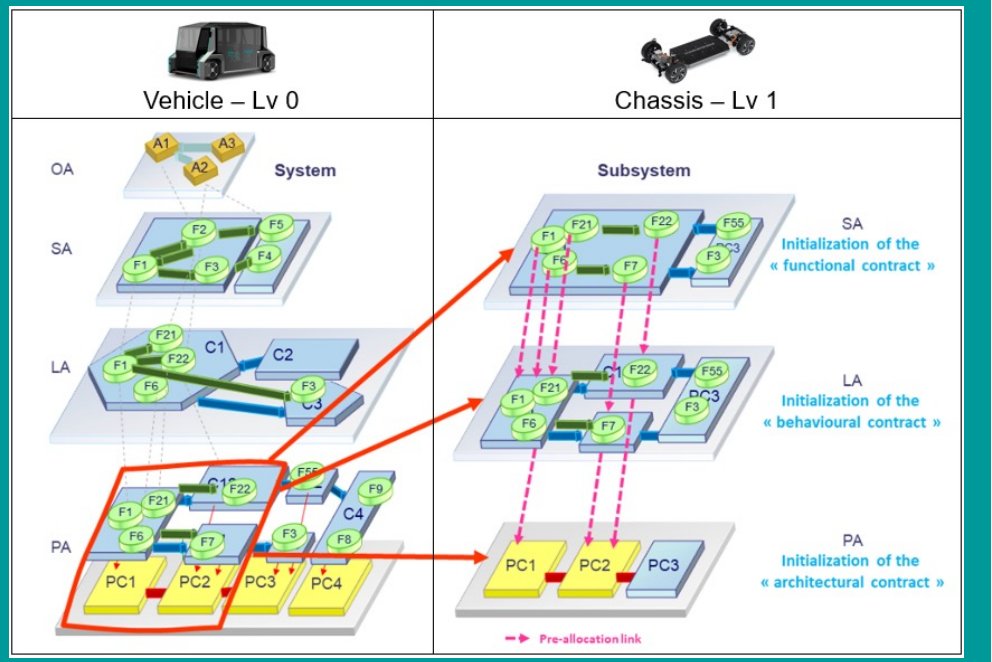
For this study, following 3 criteria are deemed essential:

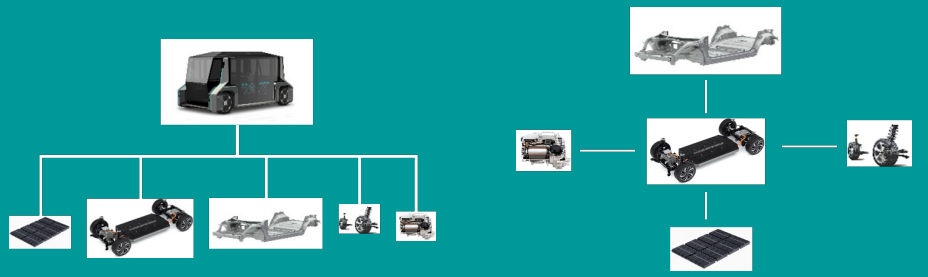
1. **Module Standardization**
2. **Interfaces**
3. **Reusability**

4 Architecture Modularization

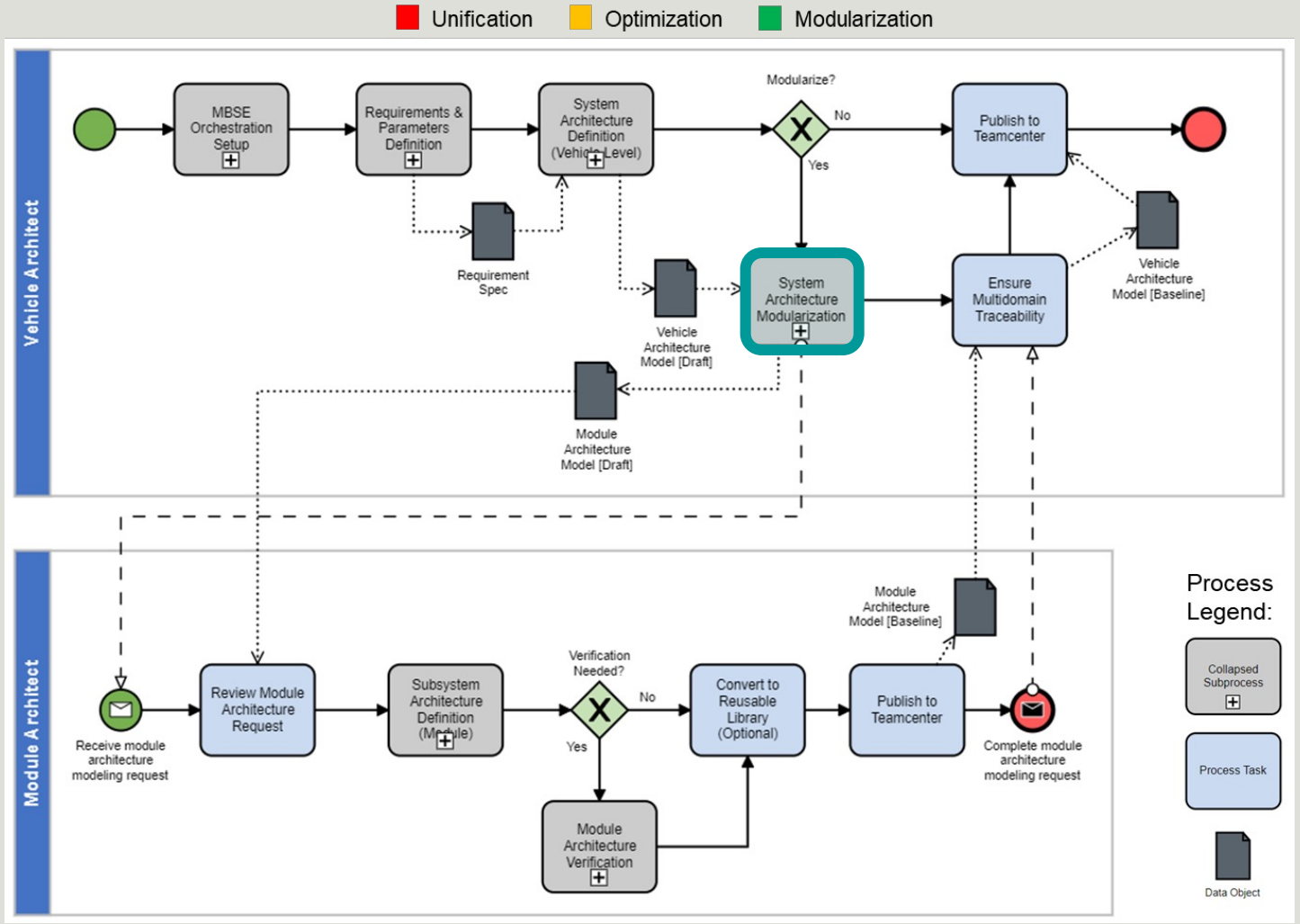


Automated system to subsystem transitions





4 Architecture Modularization



Inputs:

- Vehicle architecture model
- Requirement / parameter traceability
- Existing module classification data

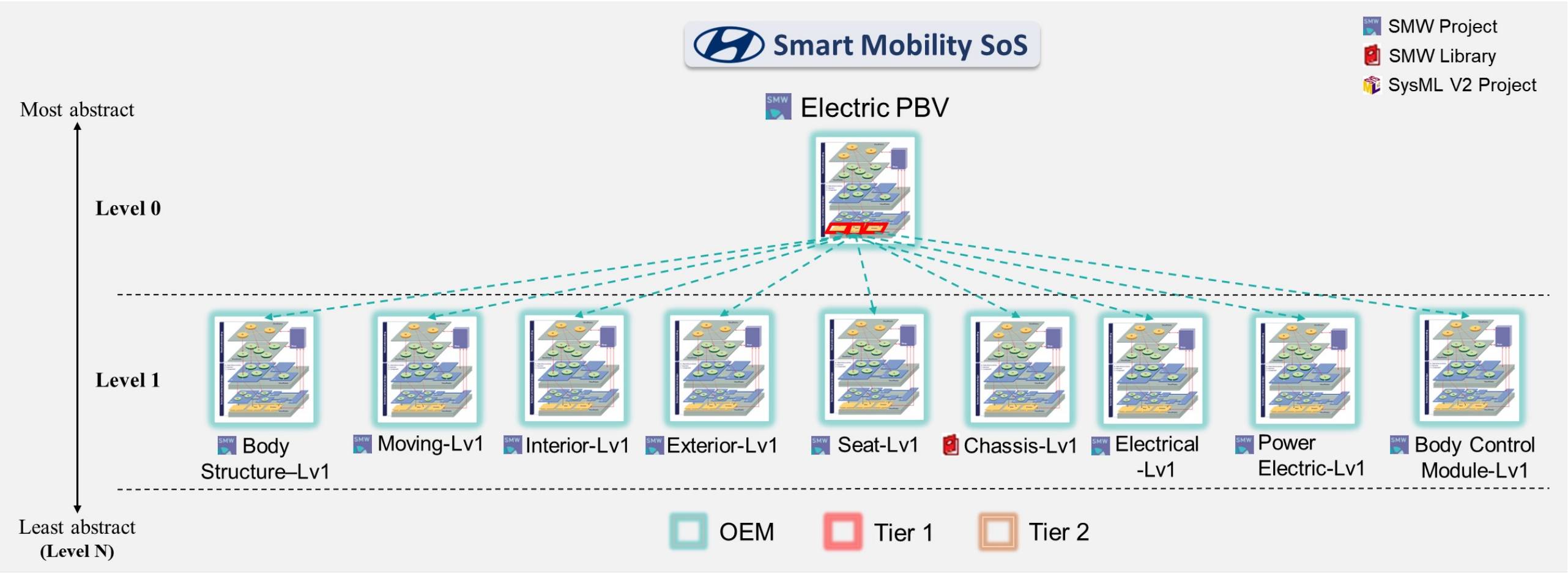
Activities:

- Assign generic vehicle coding to system breakdown
- Define module boundaries
- Generate module architecture models from vehicle architecture model
- Save vehicle / module architectures in enterprise catalog

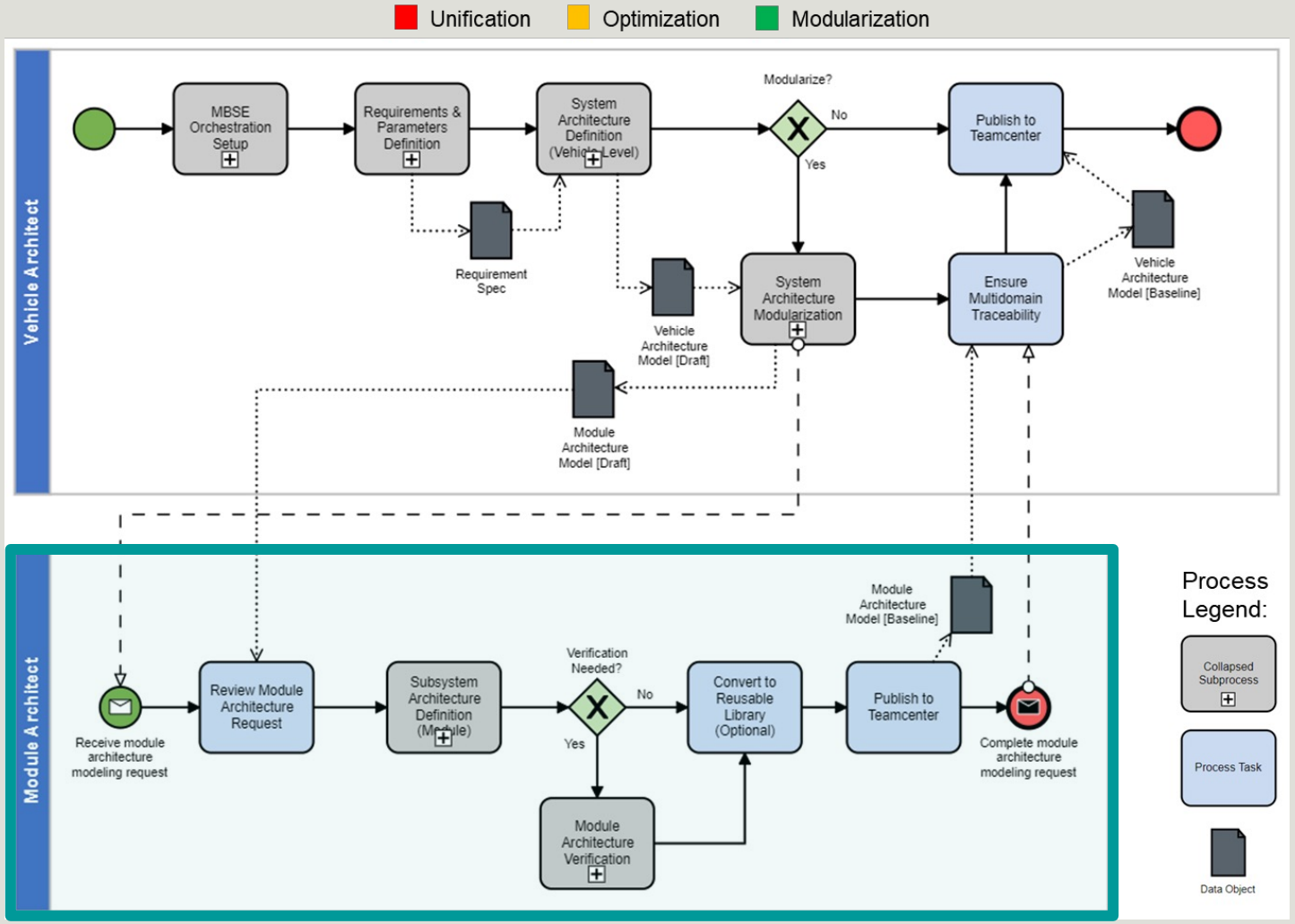
Outputs:

- Draft vehicle / module architecture models (configurable)
- Draft module requirement spec
- Traceability to domain-specific models and structures

Architecture Modularization



5 Module Architecture Development



Inputs:

- Module architecture draft
- Requirement / parameter traceability
- Module usage context

Activities:

- Refine module architectures
- Create reusable module libraries
- Define module interfaces, assign generic module coding
- Save vehicle / module architectures in enterprise catalog
- Release module architecture

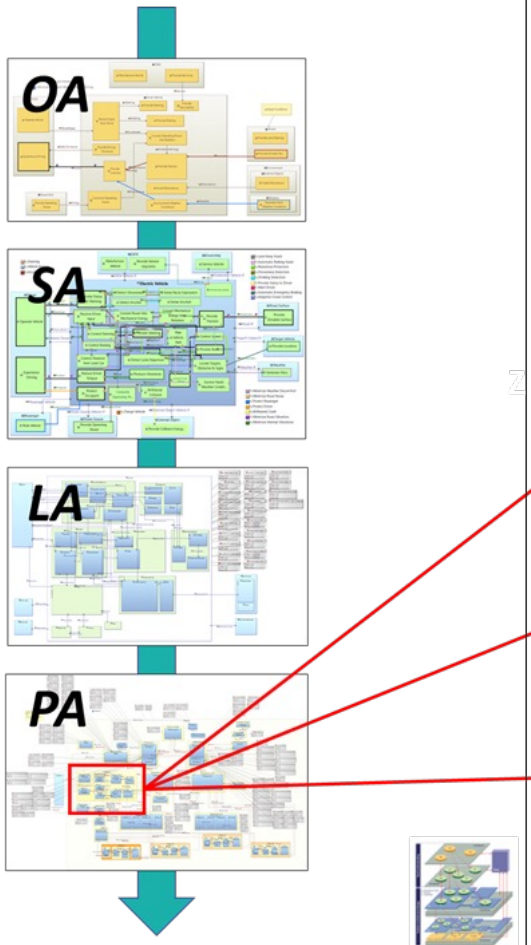
Outputs:

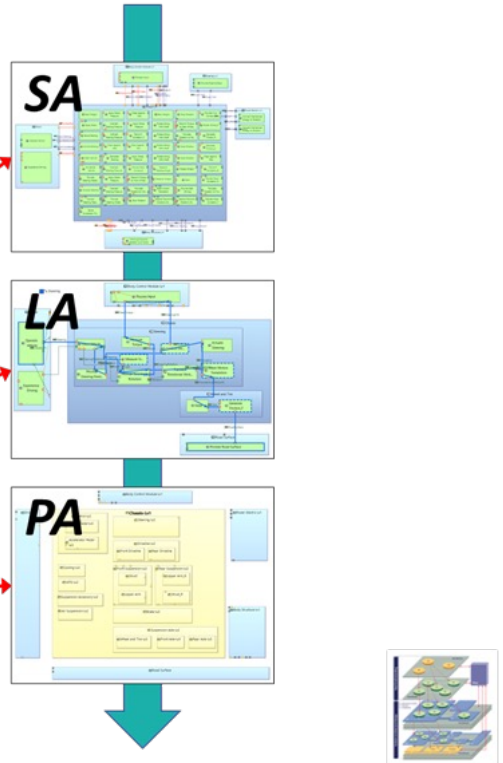
- Baselined module architecture model (configurable)
- Model-based module interface definition
- Reusable libraries of module architectures

Module Architecture Development

PBV – Lv 0

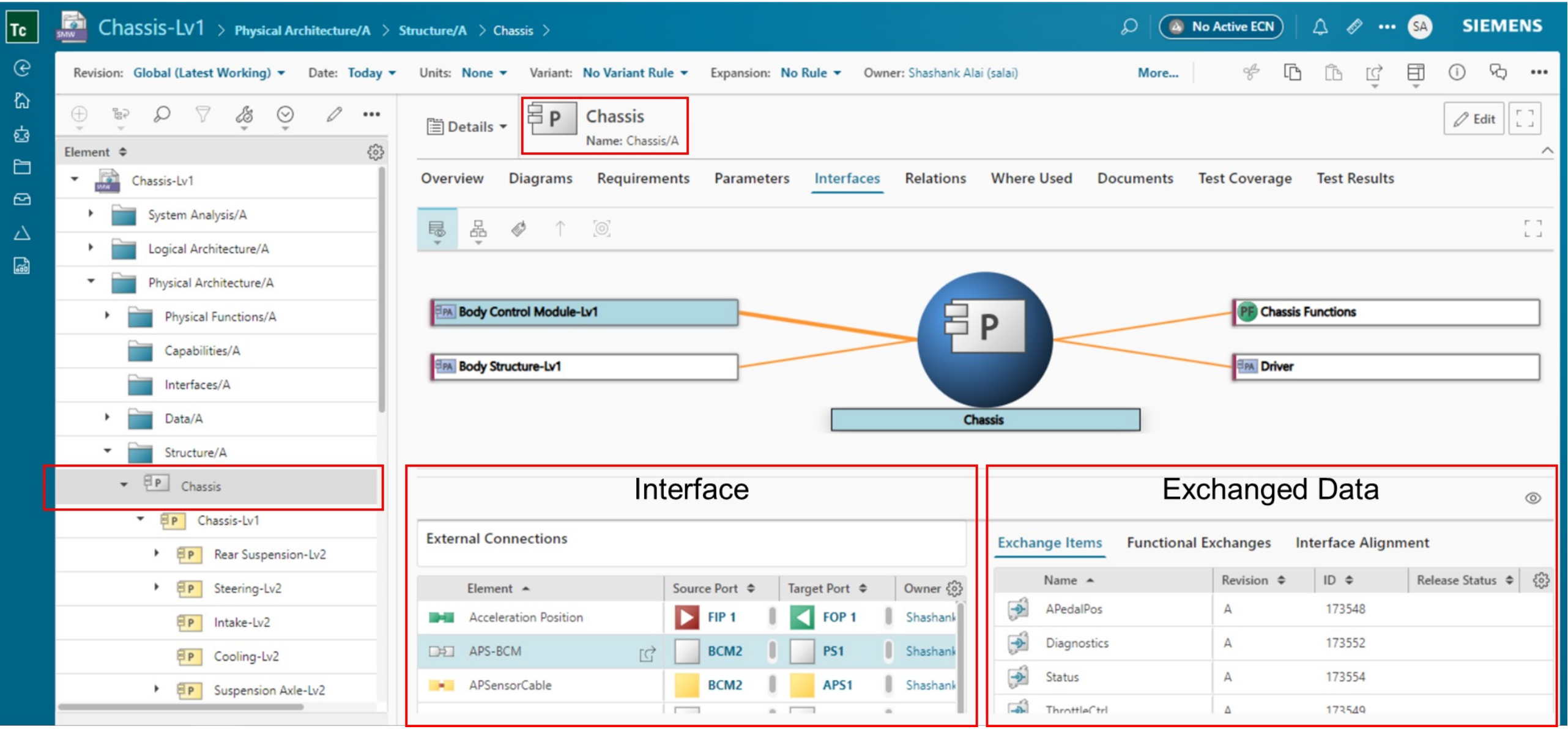
Chassis – Lv 1





Chassis module architecture definition

Module Architecture Development



Model-based module interface definition in Teamcenter

Lessons Learned

Conclusion & Observations

Conclusion

Iterative knowledge capture
Legacy product knowledge
Design evolution

Distributed development
Designated module teams
Designers and vendors

Configurability
Module standardization
Interface definition

Improved reuse
General vehicle module libraries
Program-specific usage

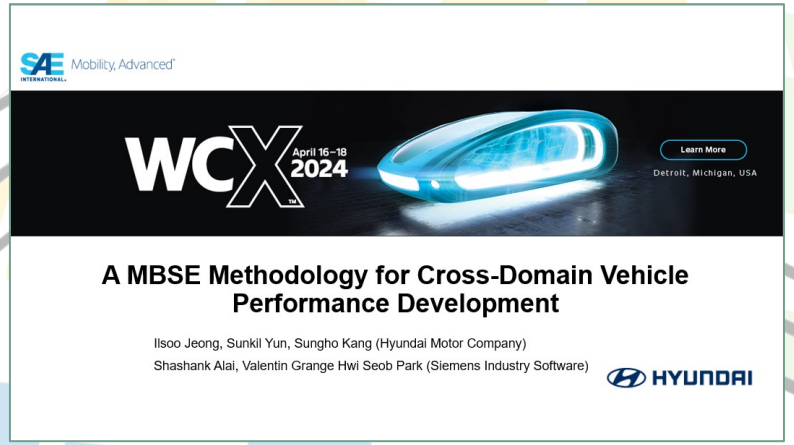
Managed complexity
Modular architecture decomposition
Granular model management

Cross-enterprise collaboration
Multi-stakeholder requirements mgmt.
Model-based supplier collaboration

Observations

- Modular architecture development a possible way to manage model development complexity
- Potential for module-based product line engineering
- Significant upfront architecture definition efforts needed for true ROI
- Pilot MBSE initiatives must extend beyond organizational silos
- When will MBSE-driven digital twins of entire product lines be a reality? Or will it?

Related Publication



1. Yun, S., Alai, S., Kim, Y., Kim, T. K., Jo, J., Lee, D., & Baloh, M. (2024). Applying a System of Systems Perspective to Hyundai-Kia's Virtual Tire Development. *INSIGHT*, *27*(1), 61-74. (Presented at INCOSE IS 2023, July 15-20, Honolulu, HI)

2. Jeong, I., Yun, S., Alai, S., Grange, V., Park, H., & Kang, S. (2024). An MBSE Methodology for Cross-Domain Vehicle Performance Development (No. 2024-01-2499). SAE Technical Paper. (Presented at SAE WCX 2024, April 16-18, Detroit, MI)

Thank you!

34th Annual INCOSE international symposium

hybrid event

Dublin, Ireland
July 2 - 6, 2024

www.incose.org/symp2024
#INCOSEIS