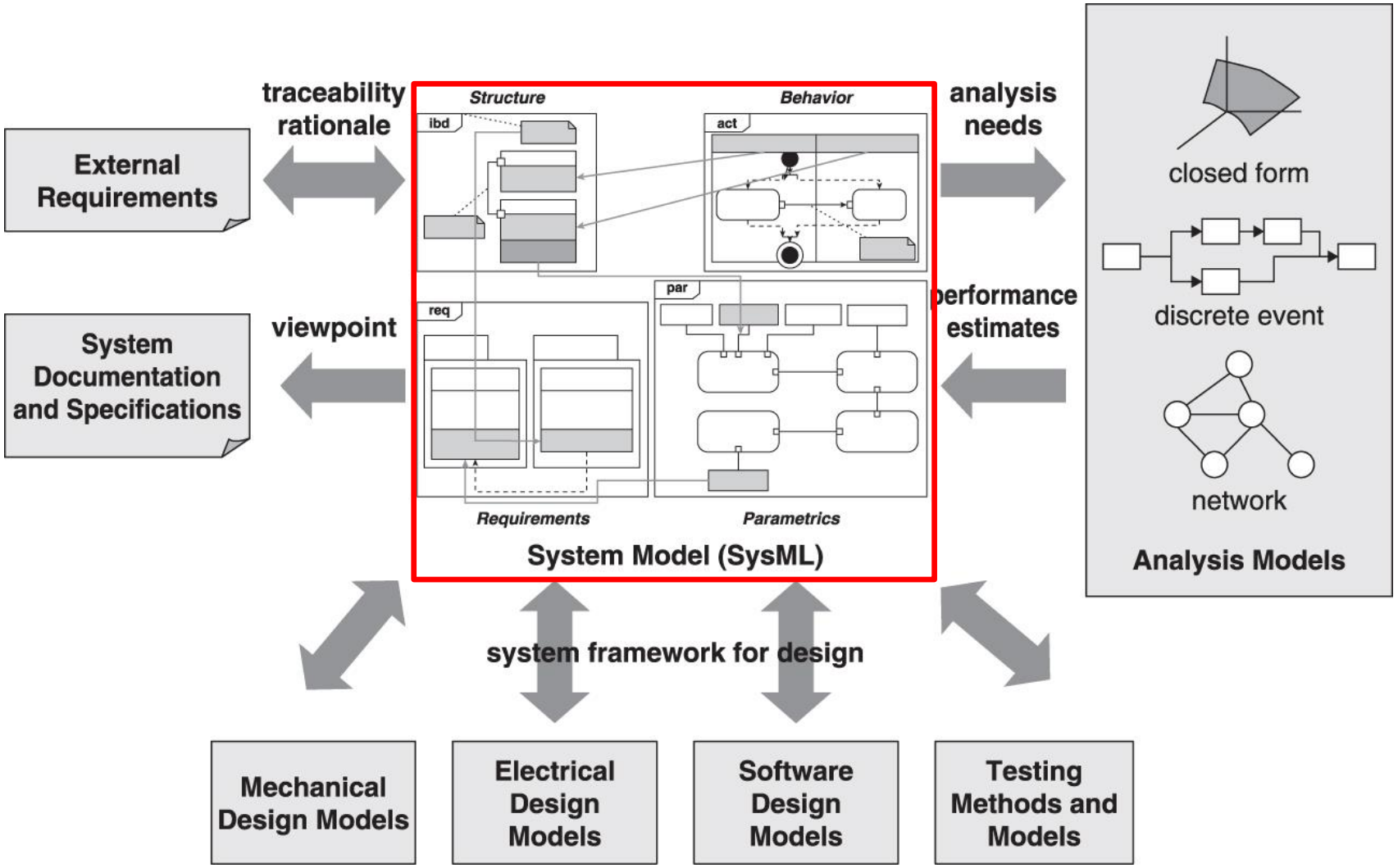
November 2, 2016

Model Based Systems Engineering with MagicGrid

No Magic, Inc.

System Model as an Integration Framework - Need for Ecosystem



The modeling language is just the language, and must be combined with a methodology to be useful

Need for a Method/Framework

This opens discussions of:

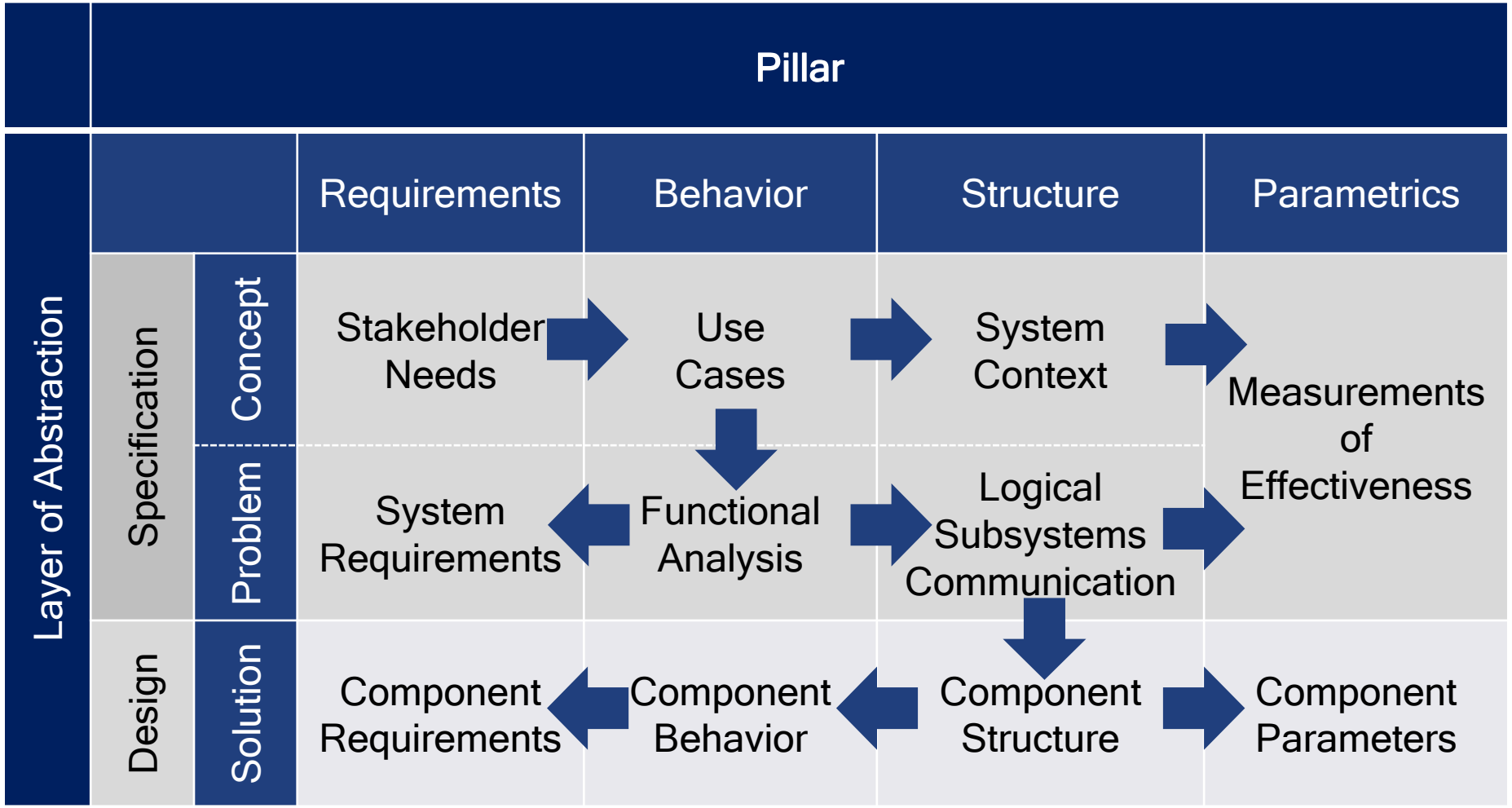
- ✓ how to structure the model
- ✓ what views to build
- ✓ which artifacts to deliver
- ✓ and in what sequence

Every company deals with the same issue differently. Some use:

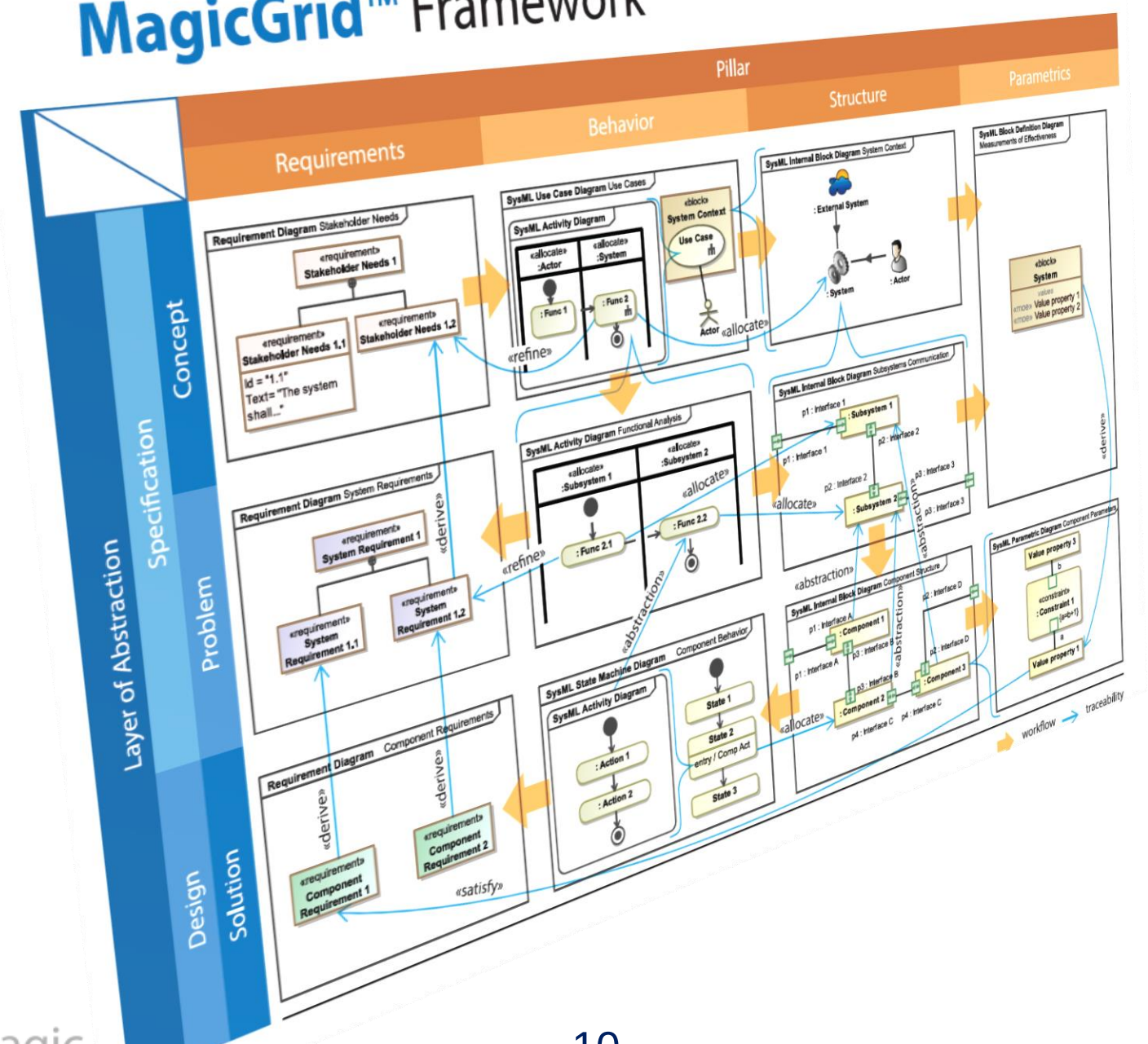
- defense architecture frameworks: DoDAF, NAF, **MODAF**
- MBSE methods: **OOSEM**, **Harmony**, **SYSMOD**, **FAS**; however, saying there is no need for an architectural framework just doesn't work.

You always end-up using an **architecture framework** whether you want one or not, or whether you intend to or not

MagicGrid



MagicGrid™ Framework

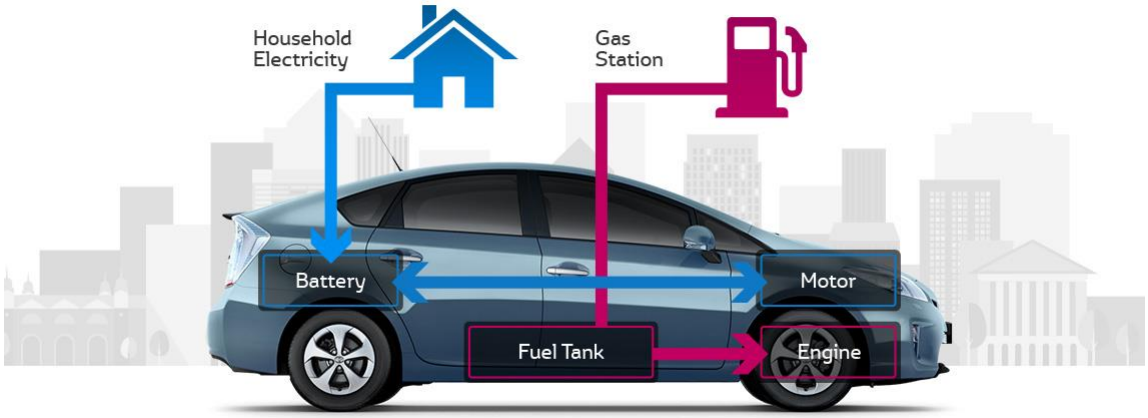


MagicGrid – Problem Domain Definition

Layer of Abstraction	Pillar			
	Requirements	Behavior	Structure	Parametrics
Concept	C1 Stakeholder Needs	C2 Use Cases	C3 System Context	C4-P4 Measurements of Effectiveness
Problem	P1 System Requirements	P2 Functional Analysis	P3 Logical Subsystems Communication	
Solution	S1 Component Requirements	S2 Component Behavior	S3 Component Structure	S4 Component Parameters

Case Study of Hybrid Automobile

- The **Hybrid Automobile** case study follows the **MagicGrid** approach to describe the concept and problem of a hybrid plug-in gas/electric powered vehicle
- The model of the case study is based on **SysML 1.4** and created with **MagicDraw** CASE tool



Stakeholder Needs

Layer of Abstraction	Pillar: Requirements	Pillar: Behavior	Pillar: Structure	Pillar: Parametrics
Concept	C1 Stakeholder Needs	C2 Use Cases	C3 System Context	C4-P4 Measurements of Effectiveness
Problem	P1 System Requirements	P2 Functional Analysis	P3 Logical Subsystems Communication	
Solution	S1 Component Requirements	S2 Component Behavior	S3 Component Structure	S4 Component Parameters

Stakeholder Needs

- The cell represents information gathered from all the stakeholders of the system
- It includes primary user requirements, government regulations, policies, procedures, etc.
- The later refinements in the model make these stakeholder needs structured and formalized

#	Name	Text
1	R SN1 Environmentally friendly	The vehicle should produce less harmful impacts to the environment than vehicles running on gasoline or diesel.
2	⊟ R SN2 Charging	
3	R SN2.1 Regenerative braking	The vehicle should be charging while braking.
4	R SN2.2 Plug-In charge	I want to plug the vehicle into my house current and charge it.
5	R SN2.3 Quick charge	I want to charge the vehicle in 30 minutes and drive at least 90 km.
6	R SN3 Electric-only propulsion	The vehicle should have electric-only propulsion mode.
7	R SN4 Distance on full charge (Electric-only propulsion)	The vehicle should drive at least 120 km on electric-only propulsion mode.

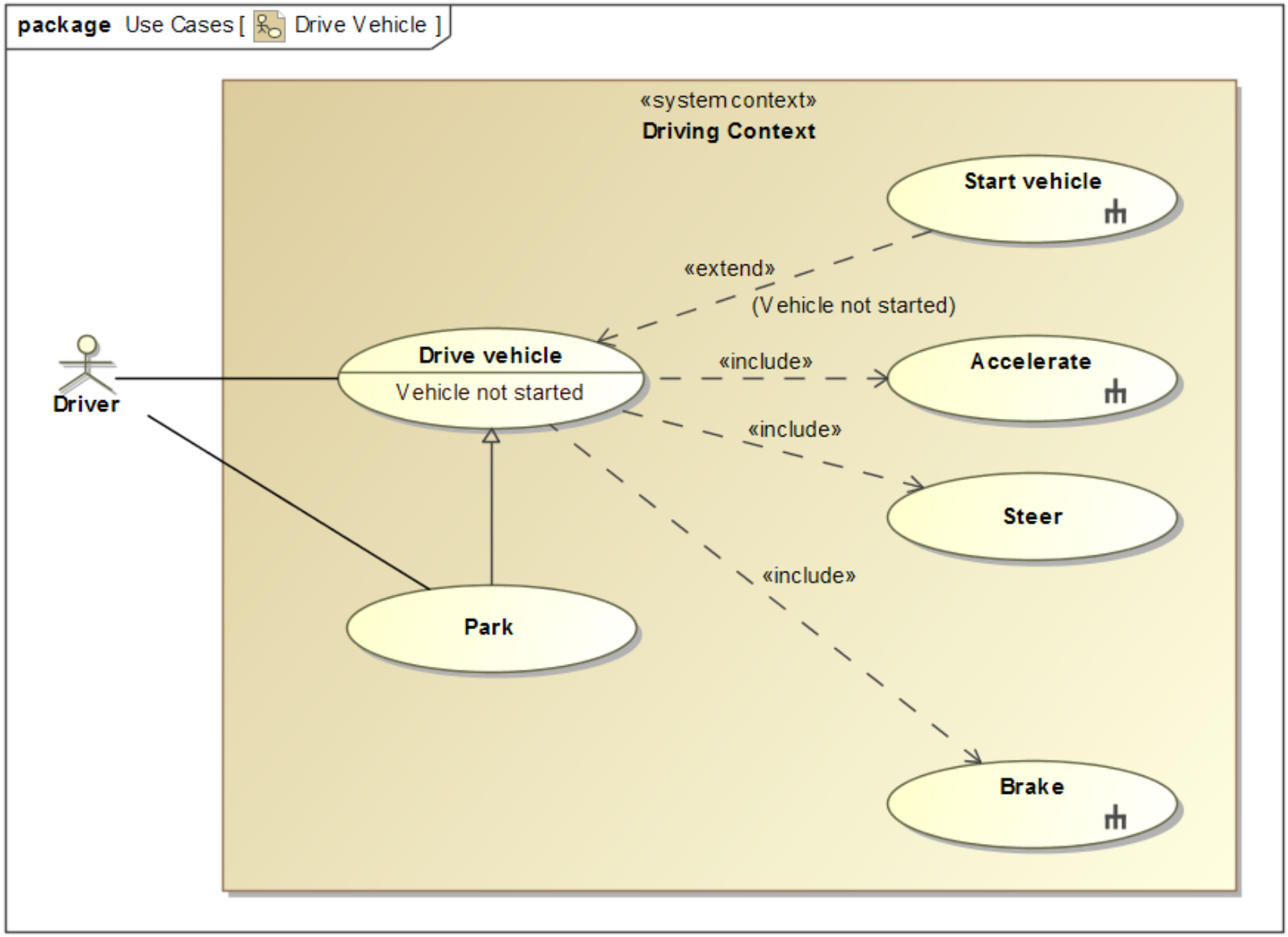
Use Cases

Layer of Abstraction	Requirements	Behavior	Structure	Parametrics
Concept	C1 Stakeholder Needs	C2 Use Cases	C3 System Context	C4-P4 Measurements of Effectiveness
Problem	P1 System Requirements	P2 Functional Analysis	P3 Logical Subsystems Communication	
Solution	S1 Component Requirements	S2 Component Behavior	S3 Component Structure	S4 Component Parameters

Use Cases

- Functional use cases that provide **measurable** value to the user
- Definitions of **system contexts**, wherein these use cases are performed
- Use case scenarios on how the system interacts with the user in the form of action/event flows

Use Cases



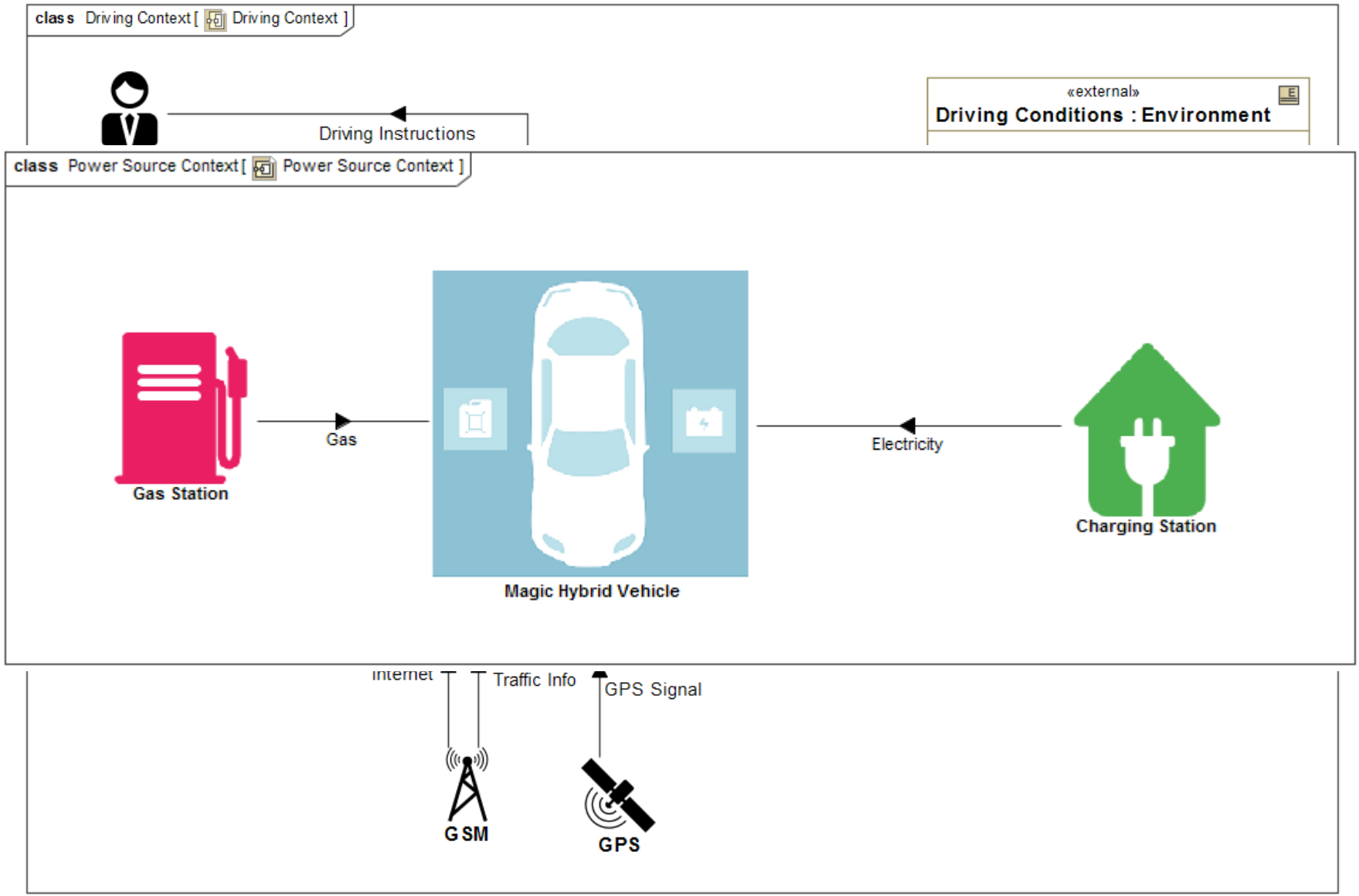
System Context

Layer of Abstraction	Pillar: Requirements	Pillar: Behavior	Pillar: Structure	Pillar: Parametrics
Concept	C1 Stakeholder Needs →	C2 Use Cases →	C3 System Context	C4-P4 Measurements of Effectiveness
Problem	P1 System Requirements	P2 Functional Analysis	P3 Logical Subsystems Communication	
Solution	S1 Component Requirements	S2 Component Behavior	S3 Component Structure	S4 Component Parameters

System Context

- Shows how the system interacts with the actors, external and internal environment
- System context is modeled in the high level of abstraction
- The purpose of this cell is to **identify high level interfaces** needed for the system to communicate with its environment

System Context



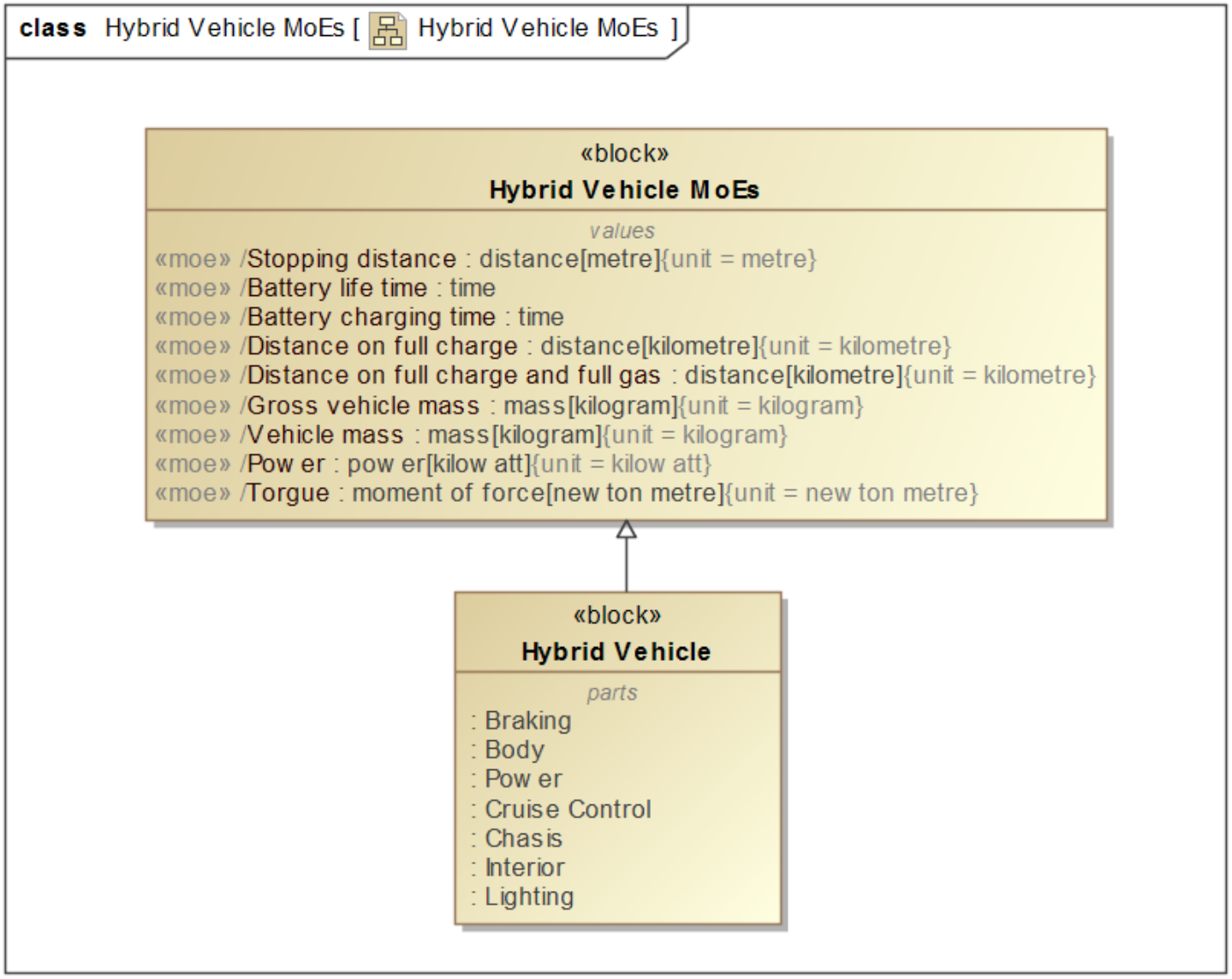
Measurements of Effectiveness (MoEs)

Layer of Abstraction		Pillar			
		Requirements	Behavior	Structure	Parametrics
Concept		C1 Stakeholder Needs →	C2 Use Cases →	C3 System Context →	C4-P4 Measurements of Effectiveness
Problem		P1 System Requirements	P2 Functional Analysis	P3 Logical Subsystems Communication	
Solution		S1 Component Requirements	S2 Component Behavior	S3 Component Structure	S4 Component Parameters

Measurements of Effectiveness (MoEs)

- Measurements of Effectiveness (MoE) are a traditional term widely used in systems engineering and describing how well a system carries out a task within a specific context
- Represents **non-functional** stakeholder needs or objectives for the system expressed in numerical format
- In this abstraction layer it serves as the high level **key performance indicators** that would be automatically checked when the Solution layer is specified

Measurements of Effectiveness (MoEs)



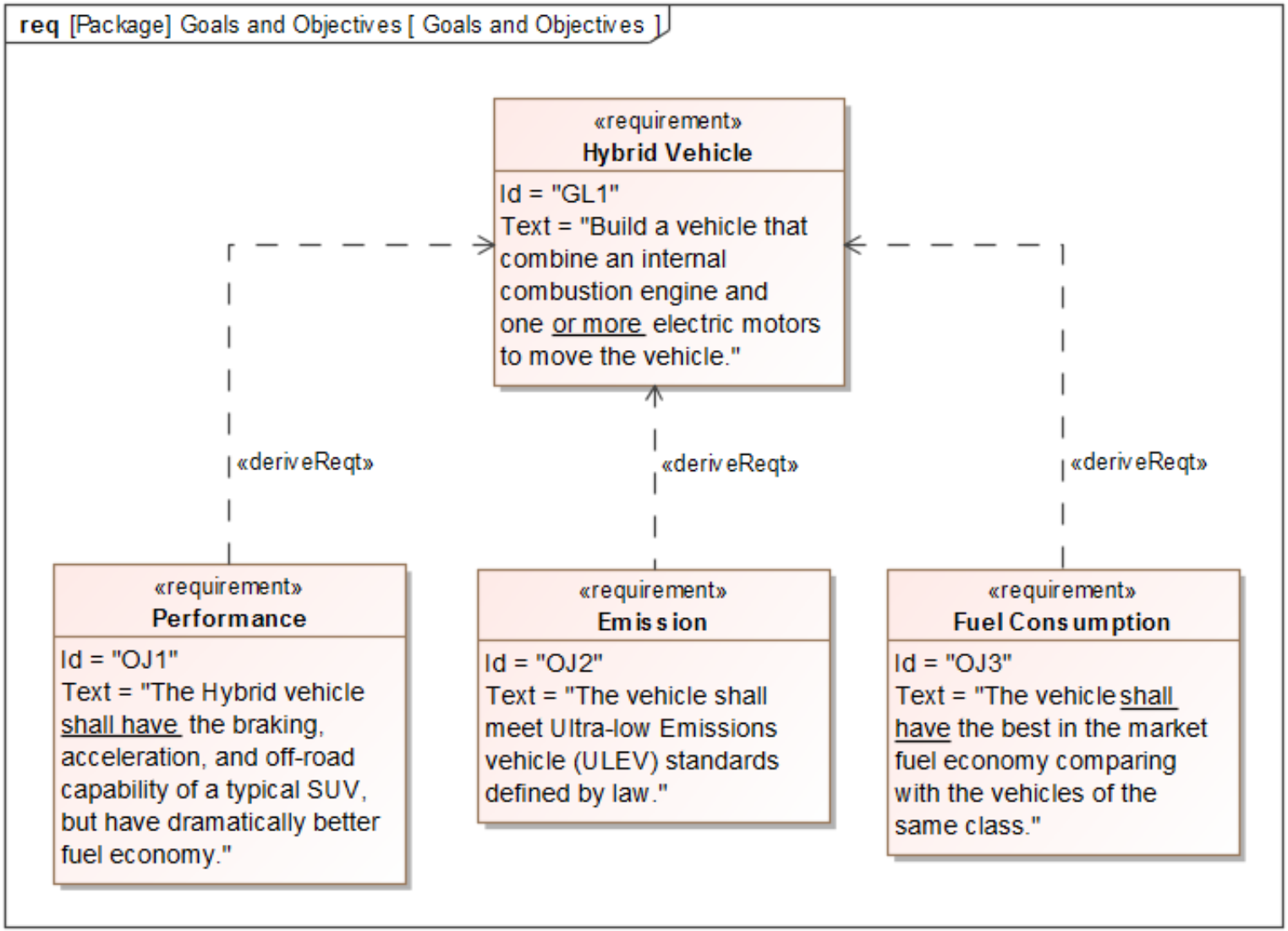
System Requirements

Layer of Abstraction	Pillar: Requirements	Behavior	Structure	Parametrics
Concept	C1 Stakeholder Needs →	C2 Use Cases →	C3 System Context →	C4-P4 Measurements of Effectiveness
Problem	P1 System Requirements	P2 Functional Analysis	P3 Logical Subsystems Communication	
Solution	S1 Component Requirements	S2 Component Behavior	S3 Component Structure	S4 Component Parameters

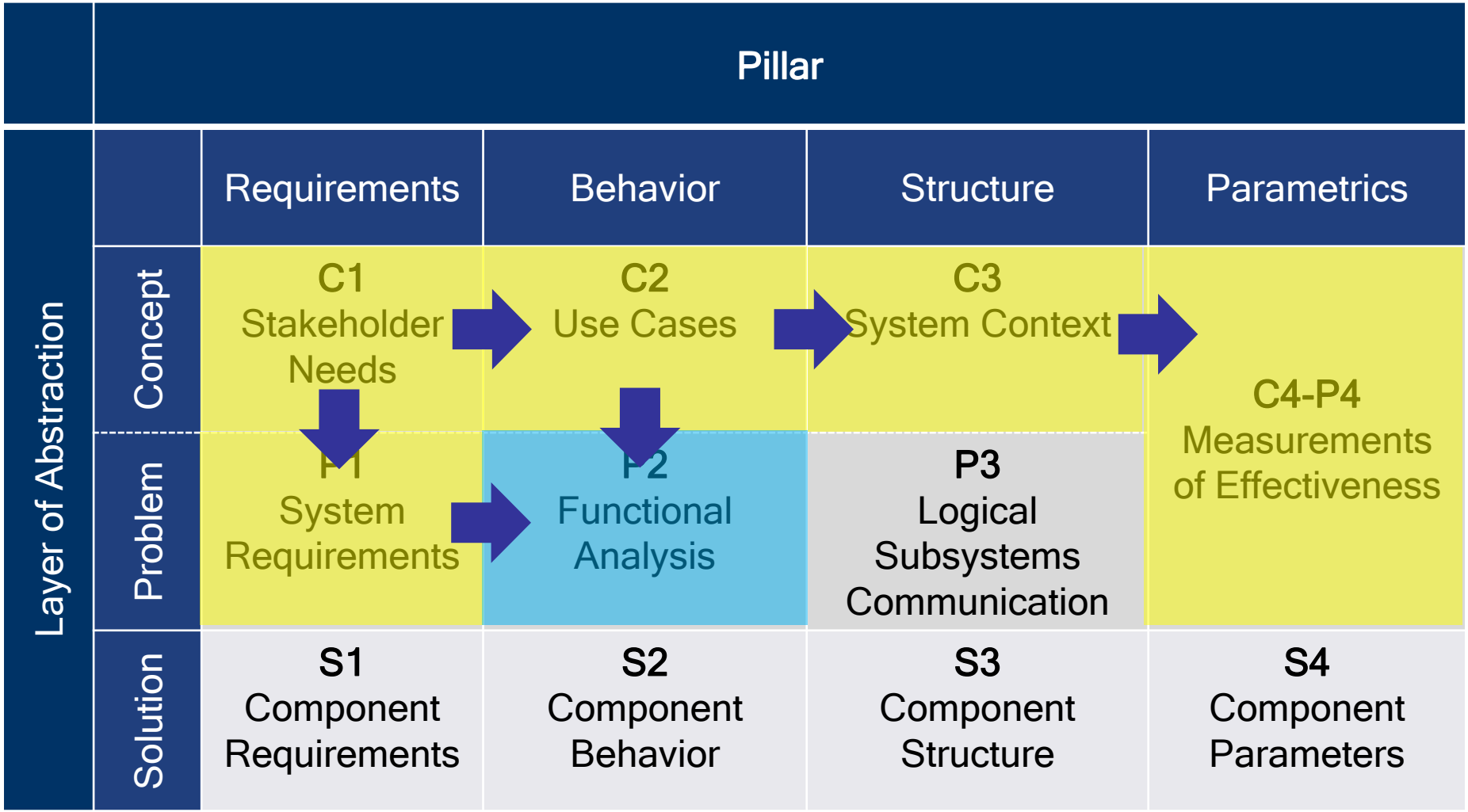
System Requirements

- **Goals** are long-term and global statements that explain what systems engineers' want to achieve and **objectives** define specific, quantifiable, time-sensitive strategies or implementation steps to attain the identified goals
- The goal and objective texts should follow agreed guidelines or standards

System Requirements



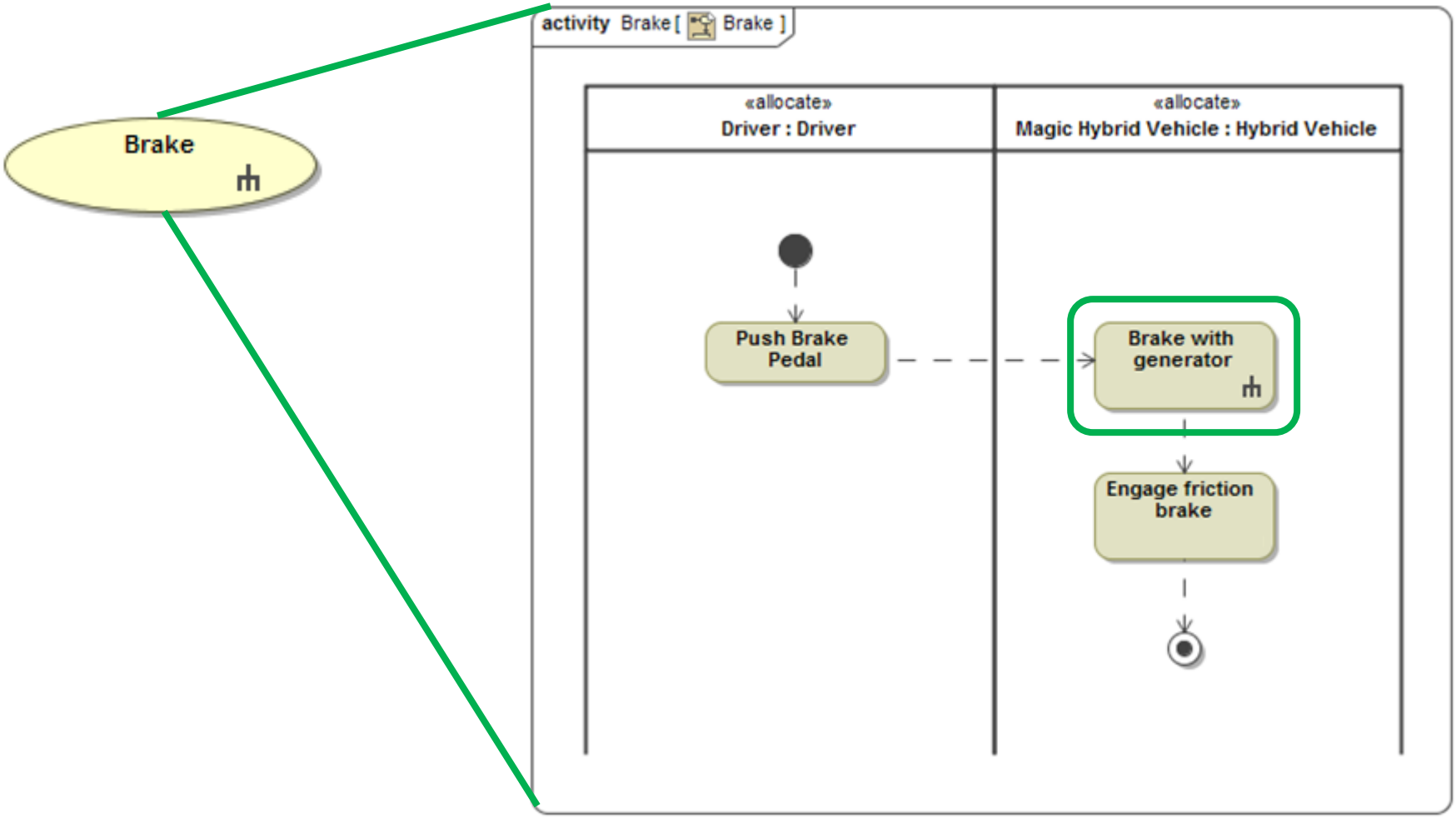
Functional Analysis



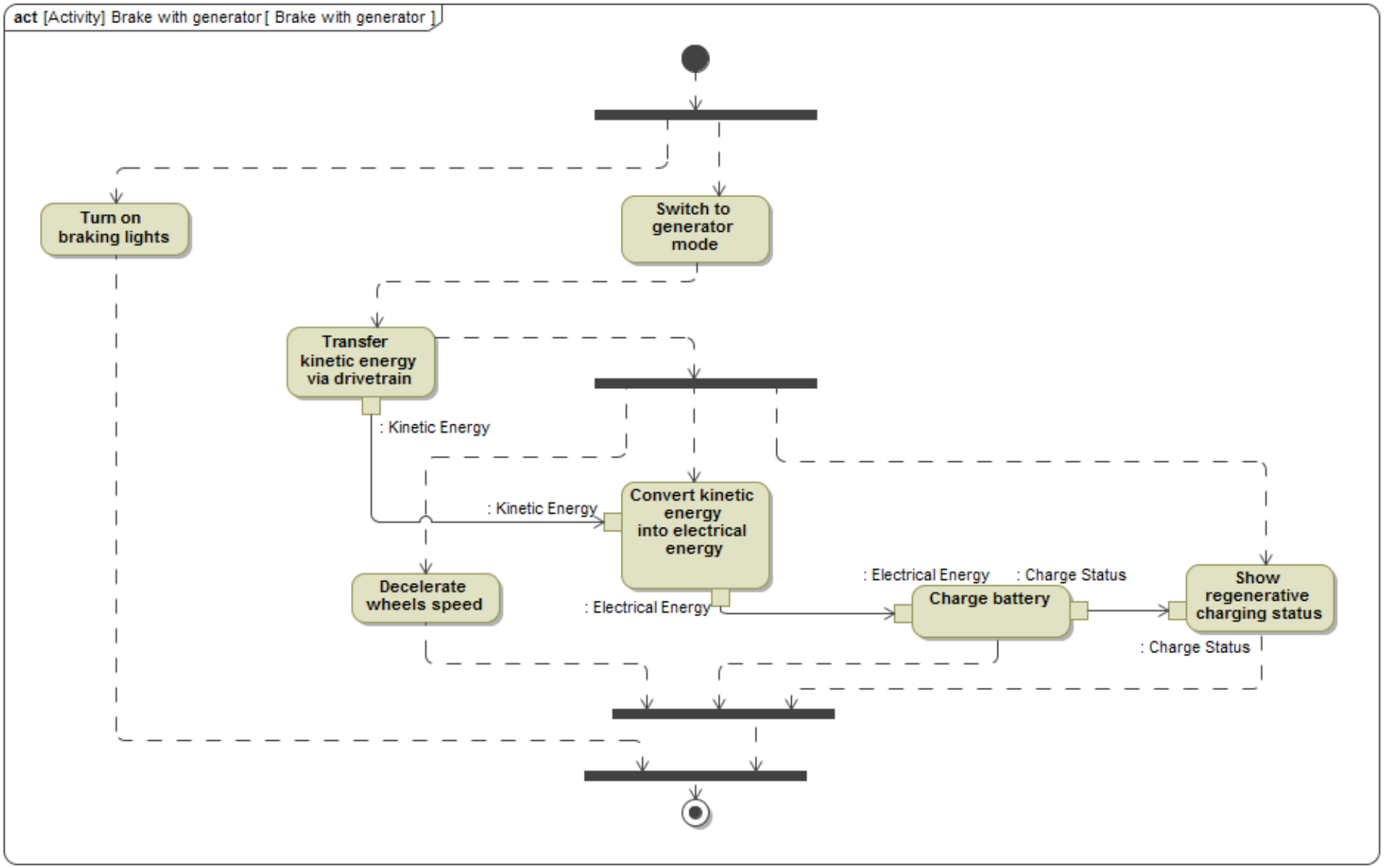
Functional Analysis

- Continuation of functional use case analysis, where focus is internal system functions in some of the techniques known as processes
- Action flows definition requires and stimulates the **identification of logical subsystems**

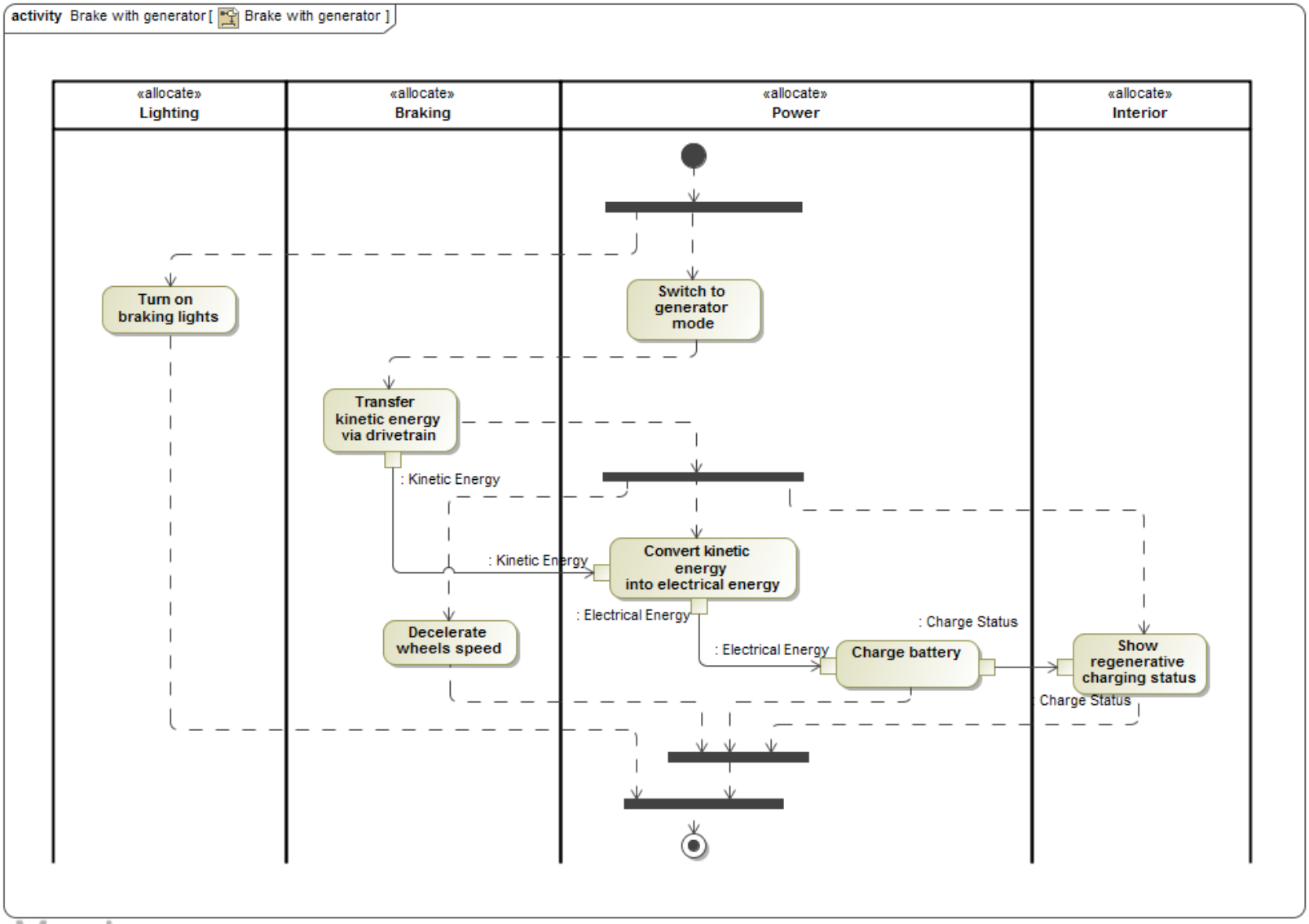
Functional Analysis



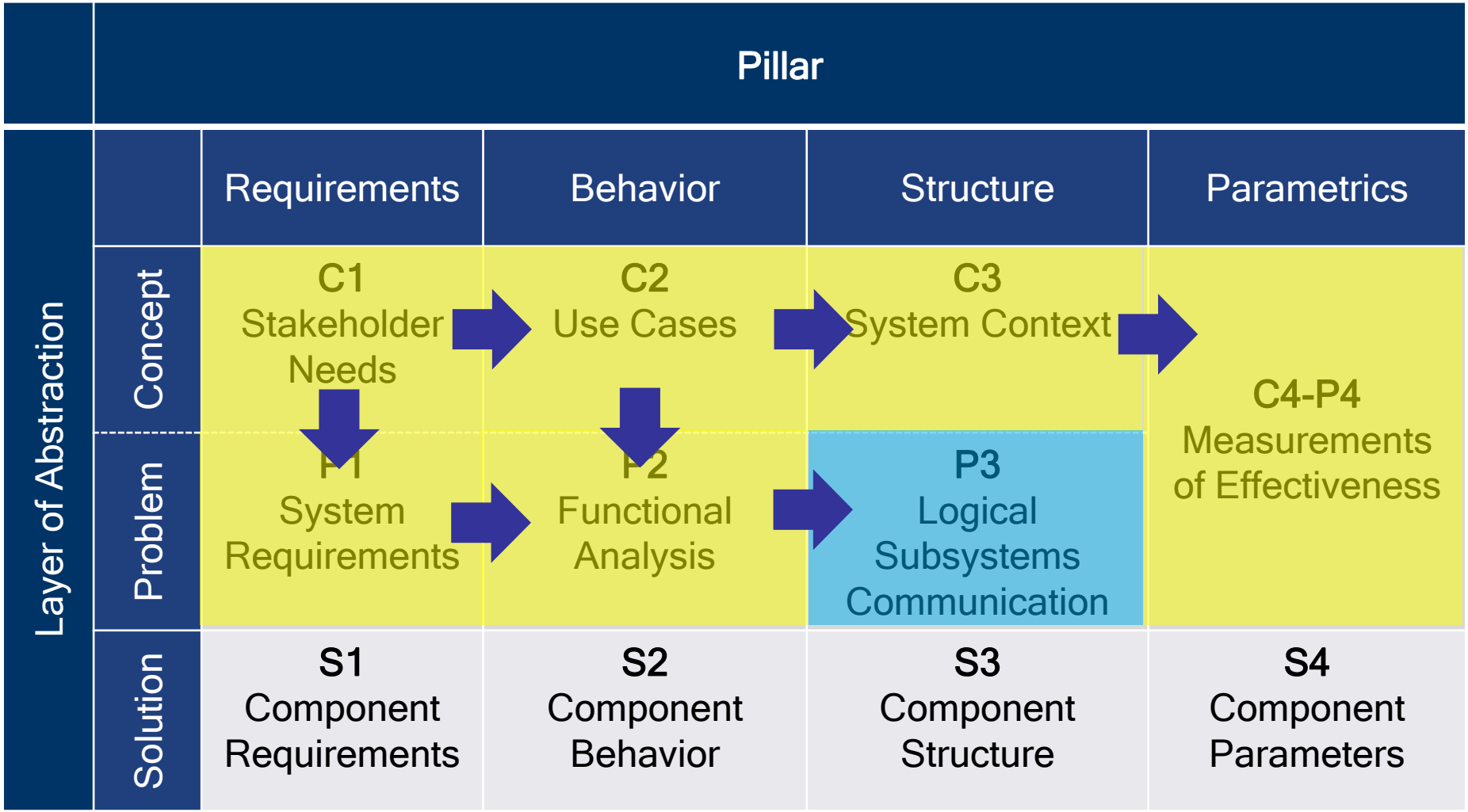
Functional Analysis



Functional Analysis



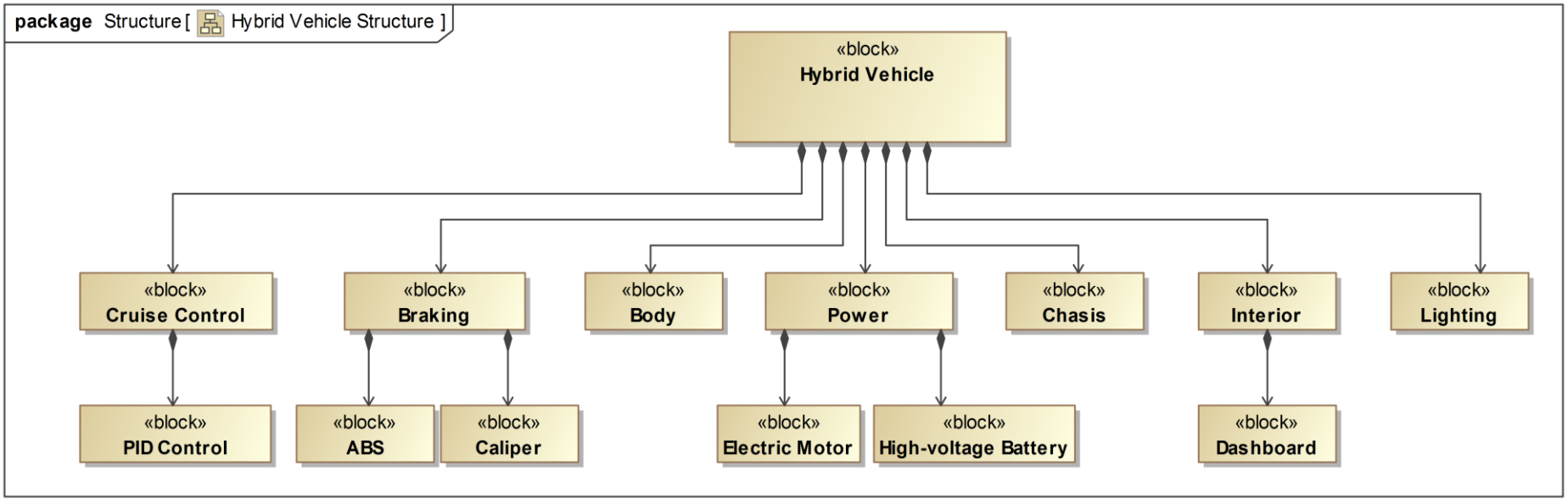
Logical Subsystems Communication



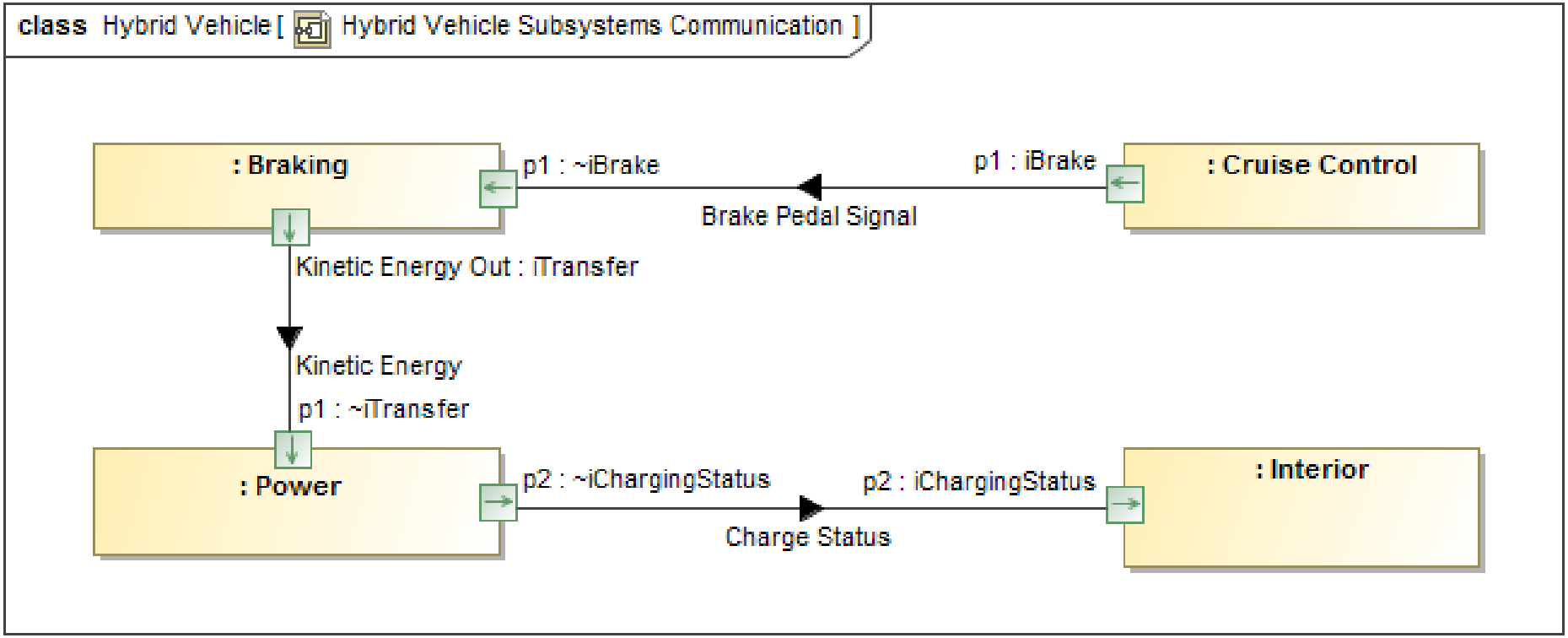
Logical Subsystems Communication

- Identified logical subsystems, based on the control and resource flows captured in the functional analysis model, are connected with one another in terms of logical interfaces
- Logical interfaces are identified and defined
- Interface control documents (ICD) can be generated

Structure


package Structure [Hybrid Vehicle Structure]
«block»
Hybrid Vehicle
«block»
Cruise Control
«block»
Braking
«block»
Body
«block»
Power
«block»
Chasis
«block»
Interior
«block»
Lighting
«block»
PID Control
«block»
ABS
«block»
Caliper
«block»
Electric Motor
«block»
High-voltage Battery
«block»
Dashboard

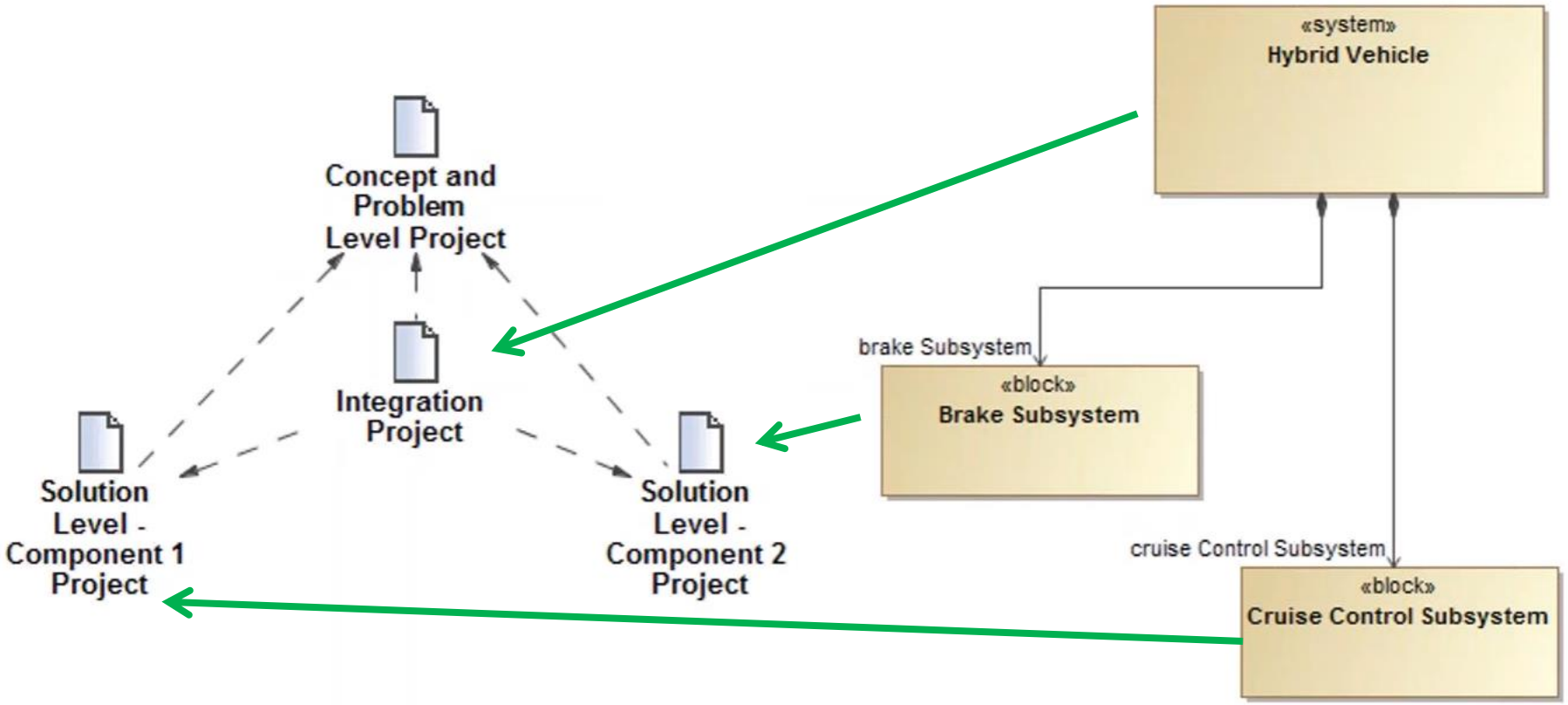
Logical Subsystems Communication



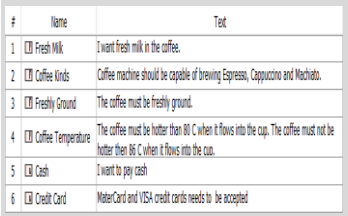
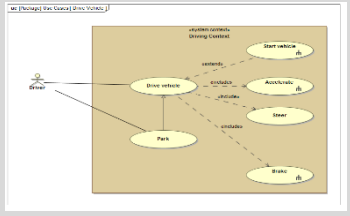
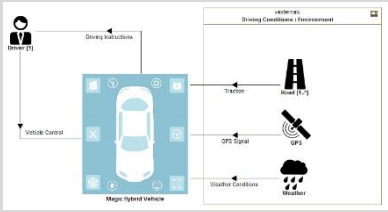
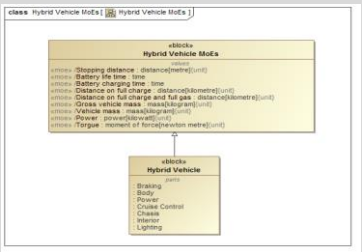
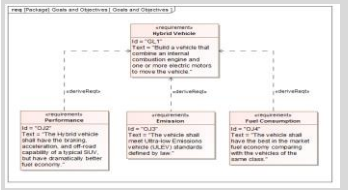
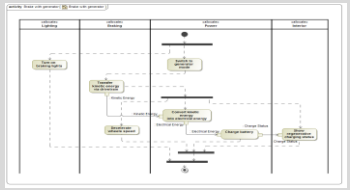
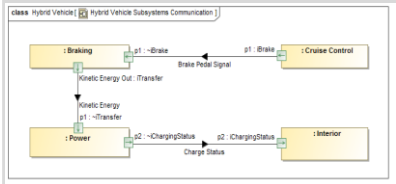
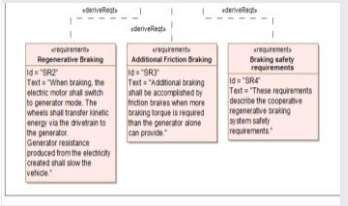
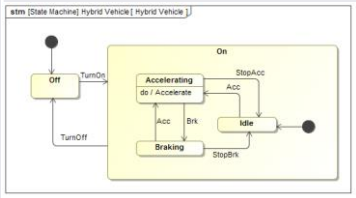
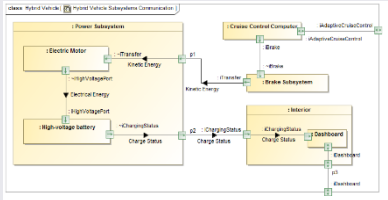
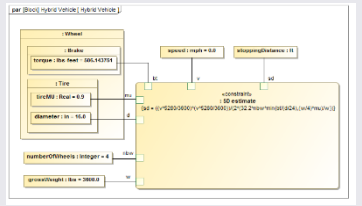
Magic Grid: Solution

		Pillar			
		Requirements	Behavior	Structure	Parametrics
Solution	System	System Requirements	System Behavior	System Assembly	Measurements of Effectiveness (MoEs)
	Subsystem	Subsystem Requirements	Subsystem Behavior	Subsystem Assembly	MoEs for Subsystems
	Component	Component Requirements	Component Behavior	Component Assembly	Physical Component Characteristics

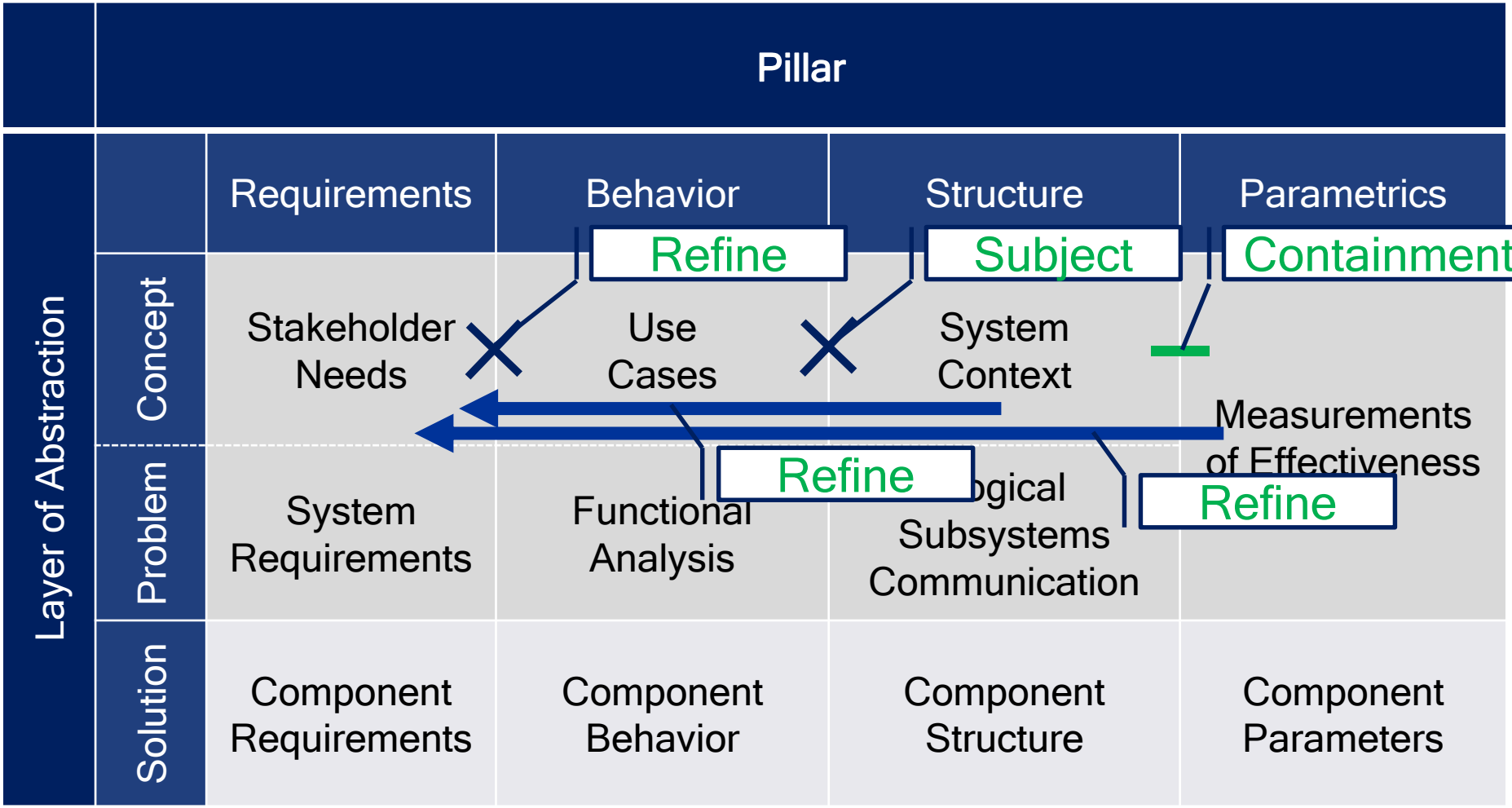
Solution Project Structure



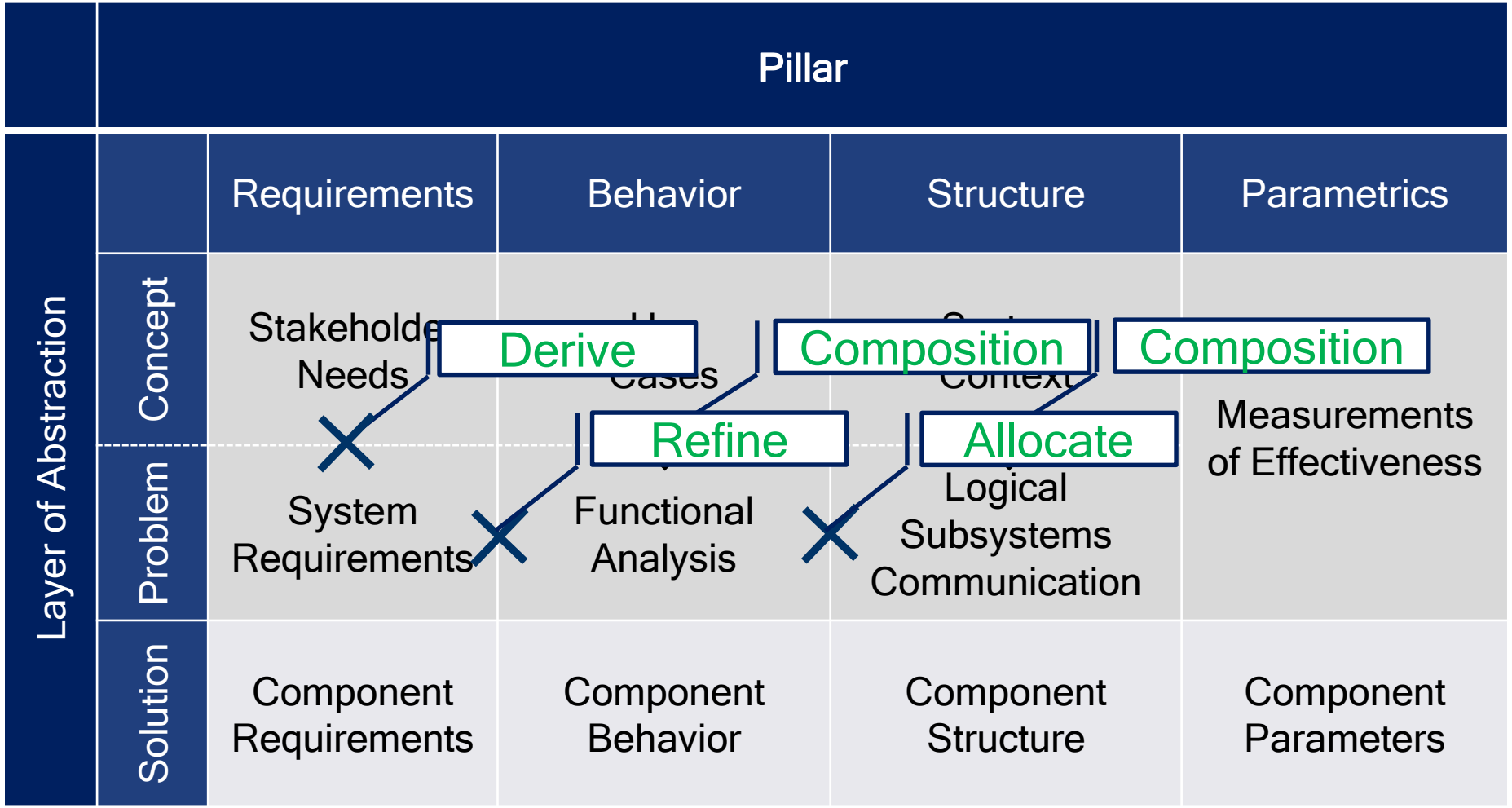
MagicGrid (2)

Layer of Abstraction		Pillar: Requirements	Pillar: Behavior	Pillar: Structure	Pillar: Parametrics
Specification	Concept				
	Problem				
Design	Solution				

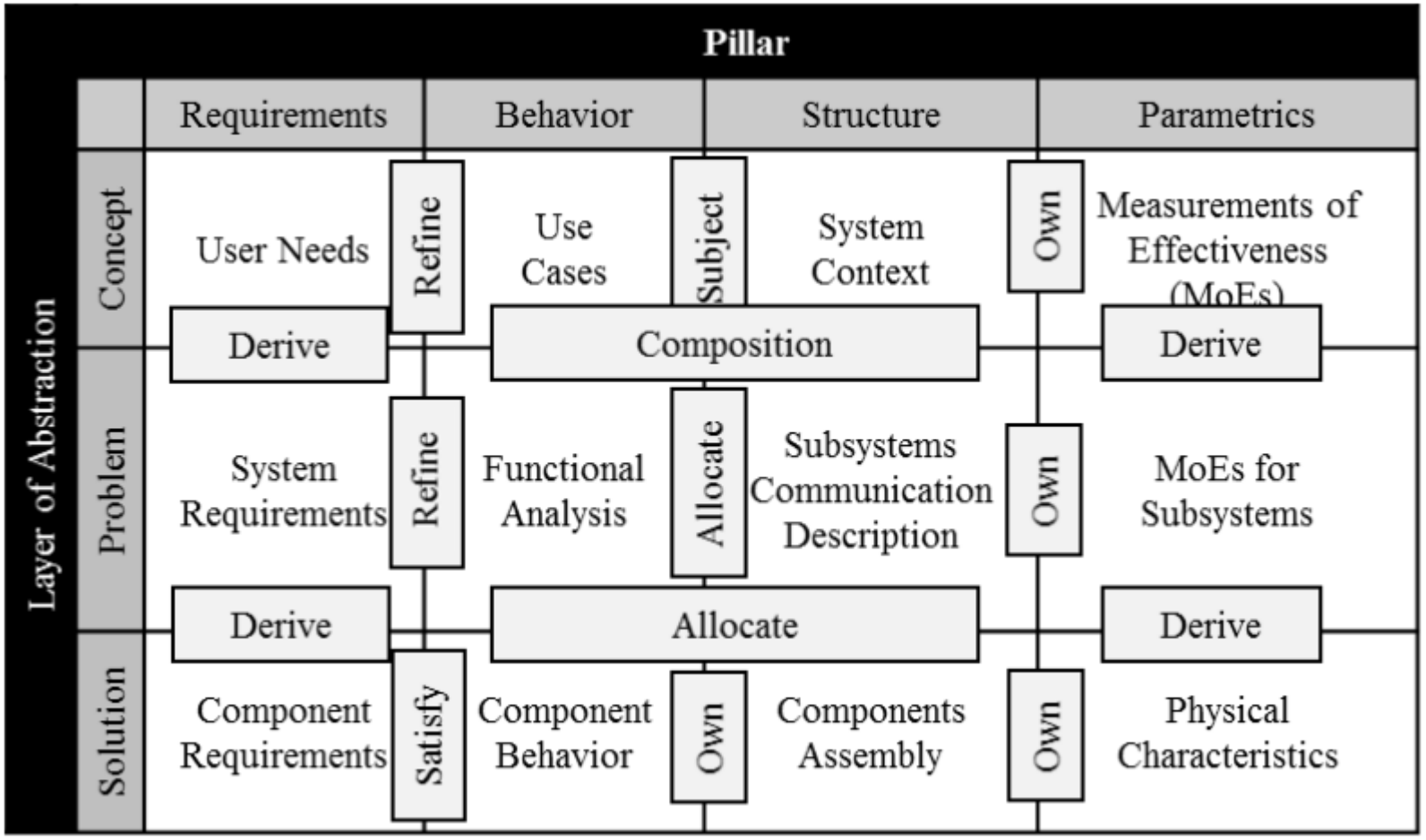
Traceability - Concept



Traceability - Problem

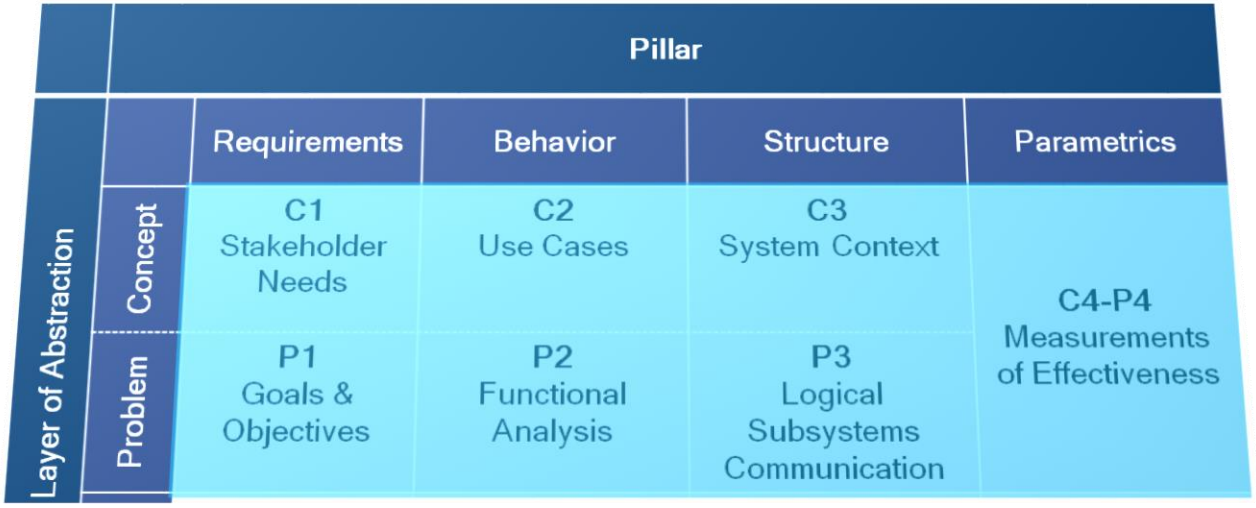


Traceability



Conclusions

- **MagicGrid** proposes a simplified framework
  - Clearly defines the modeling process
  - Reveals what models should be produced going from the highest to the lowest abstraction layers of the system analysis and design
  - Gives rules for managing relations among these layers
- Successful adopted on real-world projects



Questions and Answers