



GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

CrEST Story

Alexander Hayward

Helmut-Schmidt-Universität Hamburg

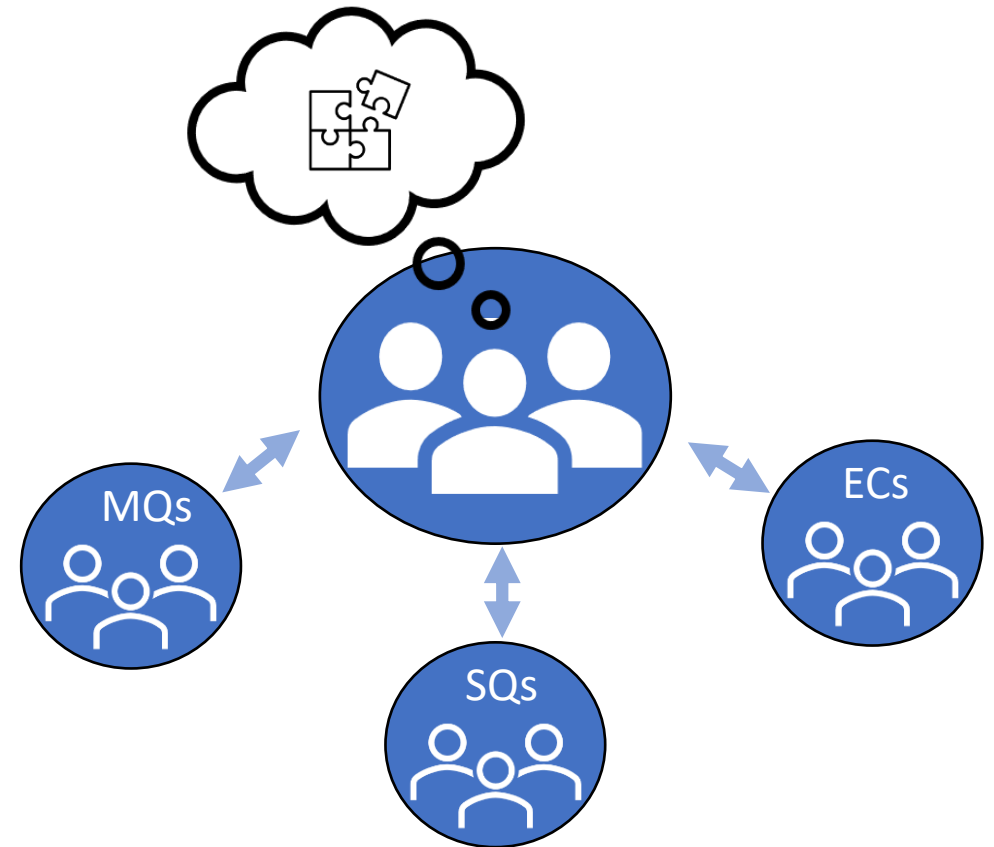
Introduction

- Transition from traditional embedded systems to collaborative embedded systems (CES)
- Ability of a company to develop CESs is becoming a crucial factor in competition
- Transition increases the complexity of information to be considered in the model-based development process of embedded systems
- The focus in the development process shifts from individual systems to dynamic, collaborative system groups (CSG) formed at runtime
- As existing methods to master the complexity of the development process only consider single systems, new methods are needed



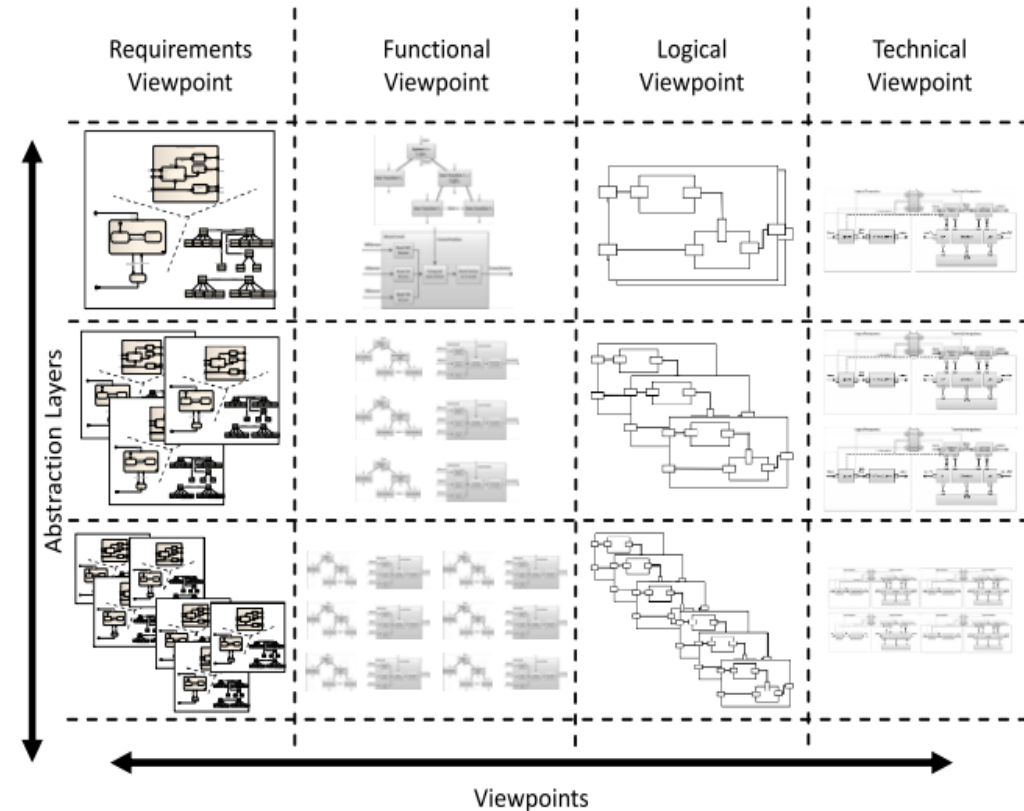
The Idea of the CrEst Story

- Initiated and jointly discussed by representatives of all sub-projects in CrEst
- Establishment of a common understanding of the project objectives and challenges
- Development of a common higher-level framework where all results can be classified and linked together
- Furthermore, the framework serves as a
 - comprehensive overview of the results
 - a reference point for all the methods and models developed in CrEst



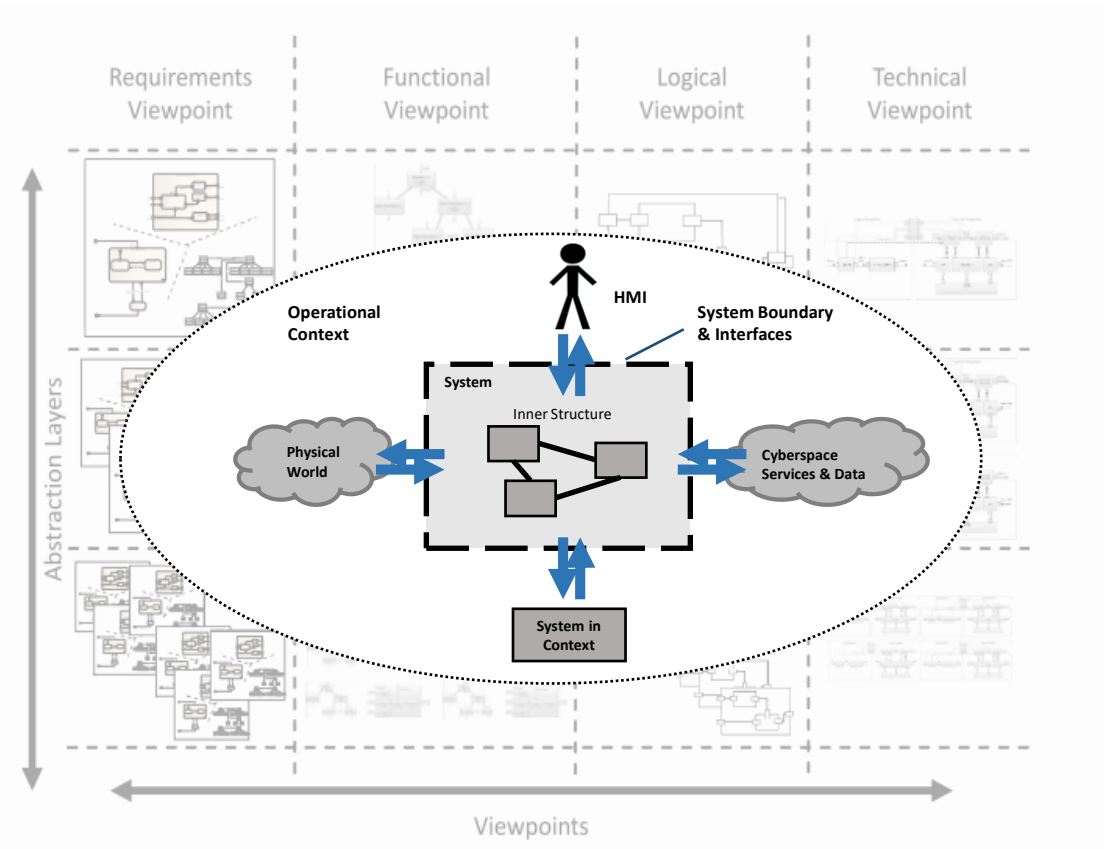
Where it all began

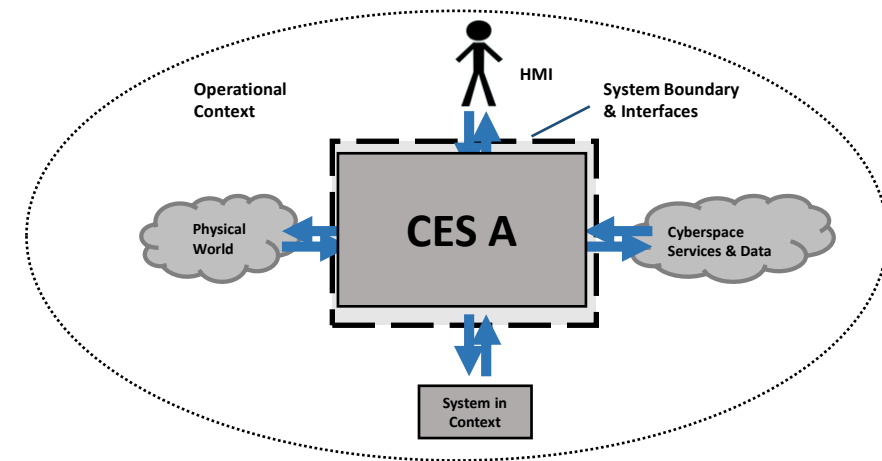
- CrEst is based on the previous projects SPES2020 and SPES_XT
- The SPES Modeling Framework provides a
 - a comprehensive methodological framework
 - for the model-based development of embedded systems
 - and with focus on consistency and semantic coherence

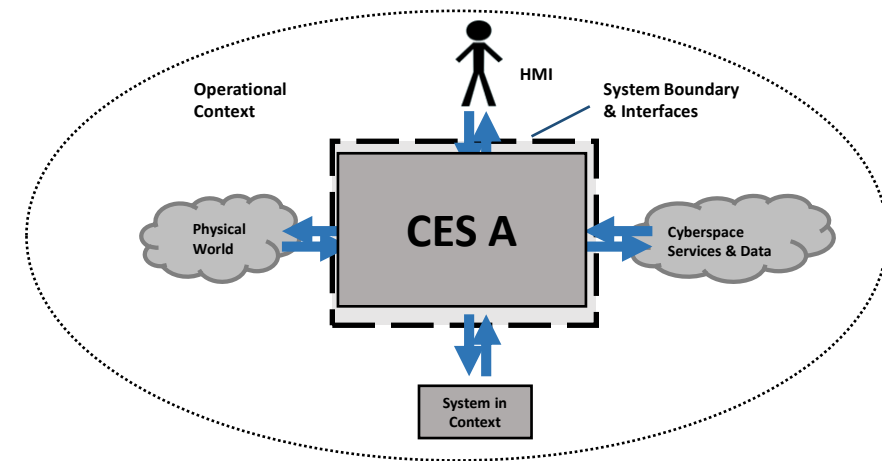


The SPES System Model

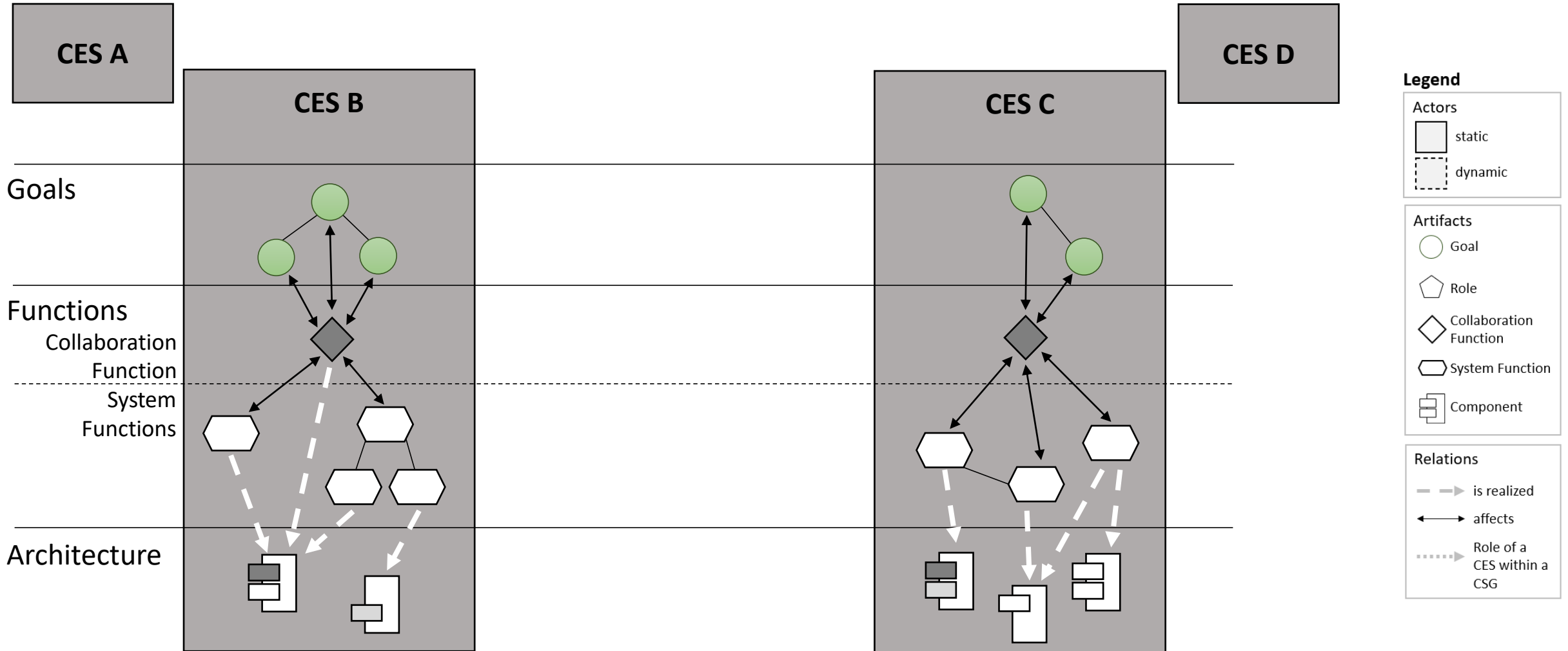
- The basis of the framework is the system model:
 - A generic model to describe a system and its properties
- Assumption of the SPES system model:
 - Elements of the system are static and do not change at runtime
 - Elements in context are assumed to be completely known at design time and static at runtime
- In CrEst, these simplifying assumptions are no longer sufficient, as we consider
 - CESs dynamically forming CSGs
 - in open, dynamic and uncertain context



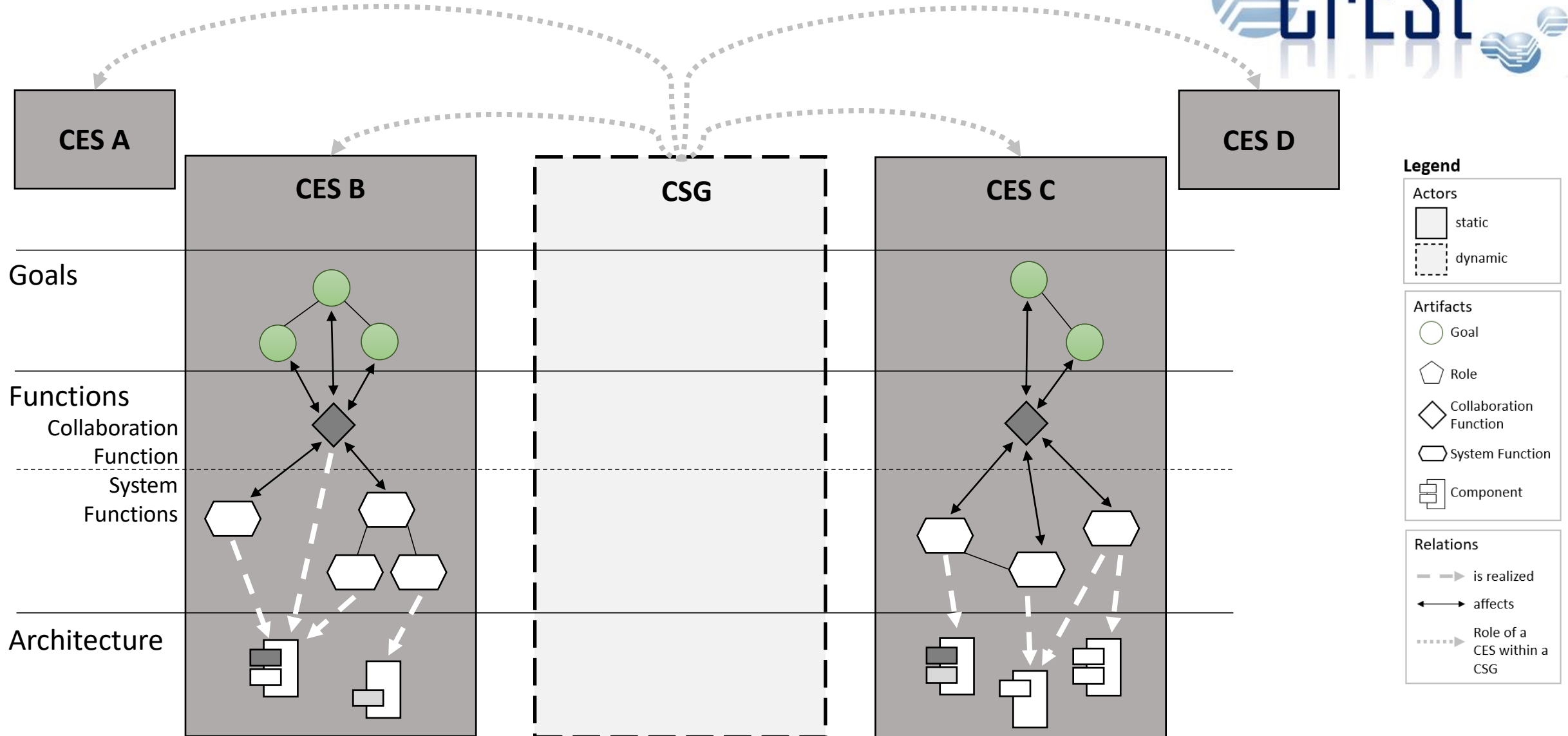




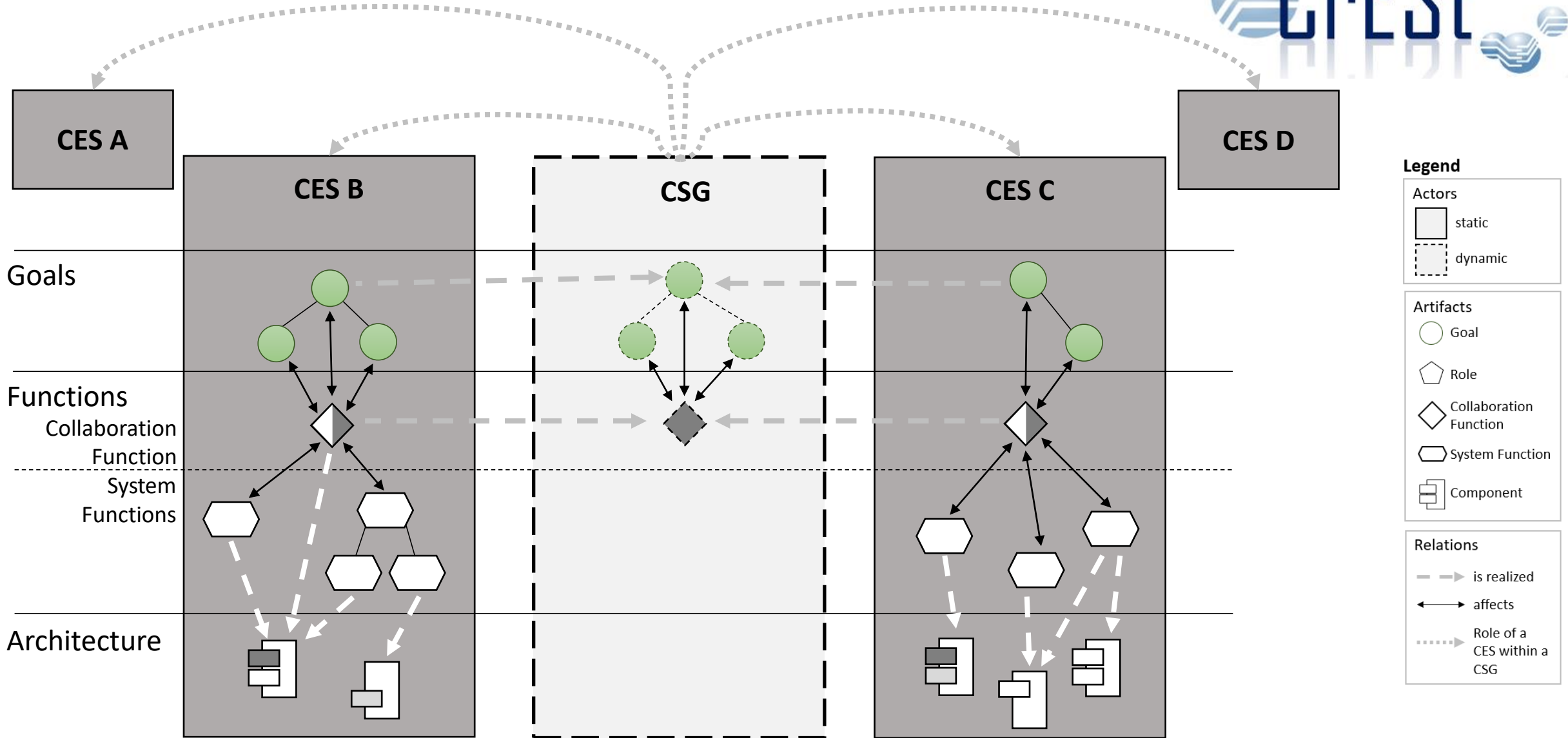
Systems considered in CrEST



Systems considered in CrEst - Collaboration



Systems considered in CrEst - Negotiation



Legend

Actors

- static (solid box)
- dynamic (dashed box)

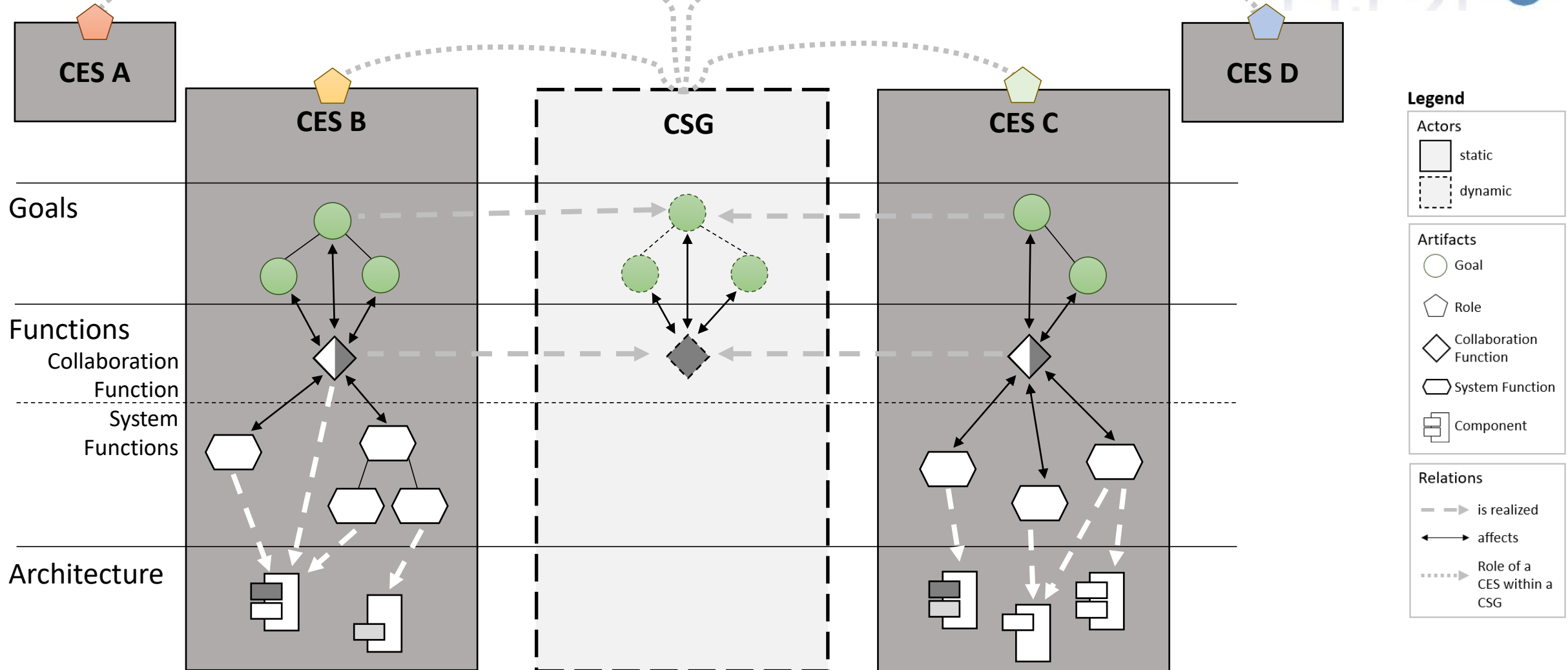
Artifacts

- Goal (green circle)
- Role (pentagon)
- Collaboration Function (diamond)
- System Function (hexagon)
- Component (stacked rectangles)

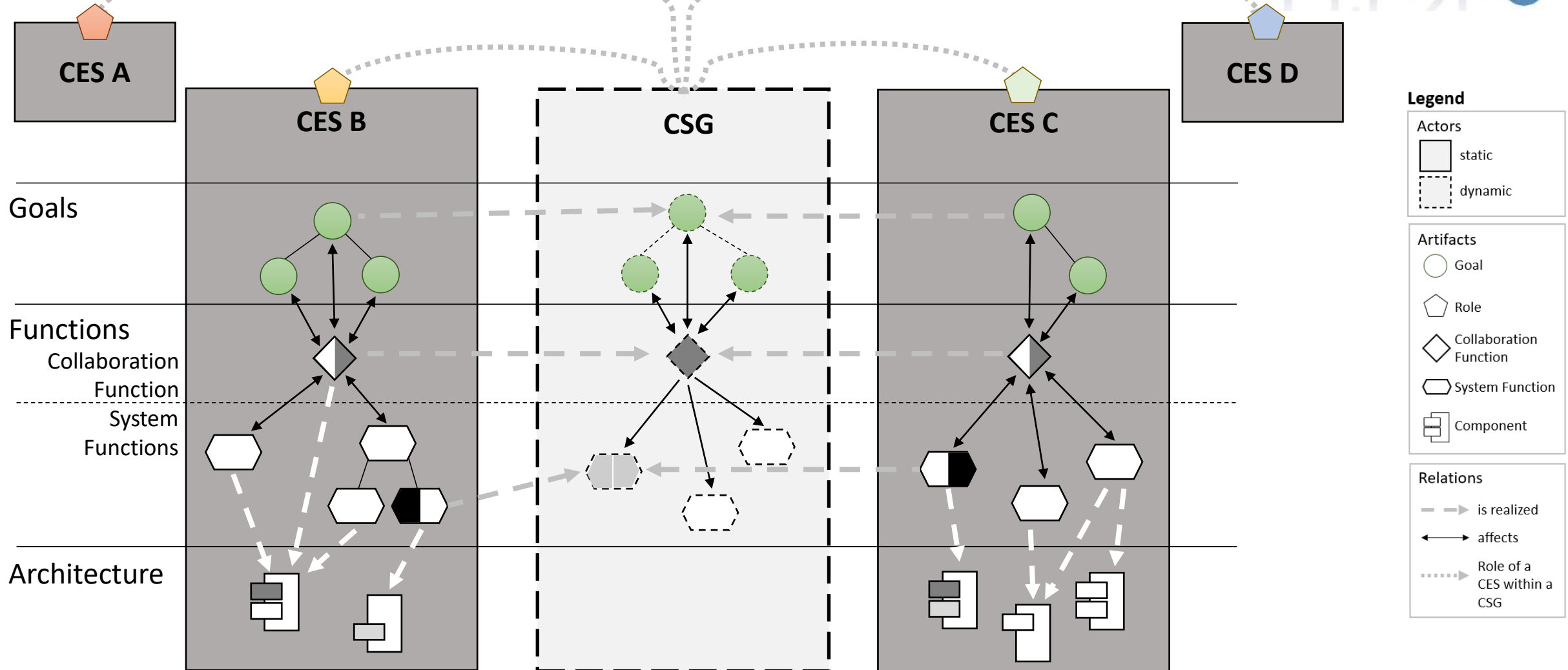
Relations

- is realized (dashed arrow)
- affects (solid arrow)
- Role of a CES within a CSG (dotted arrow)

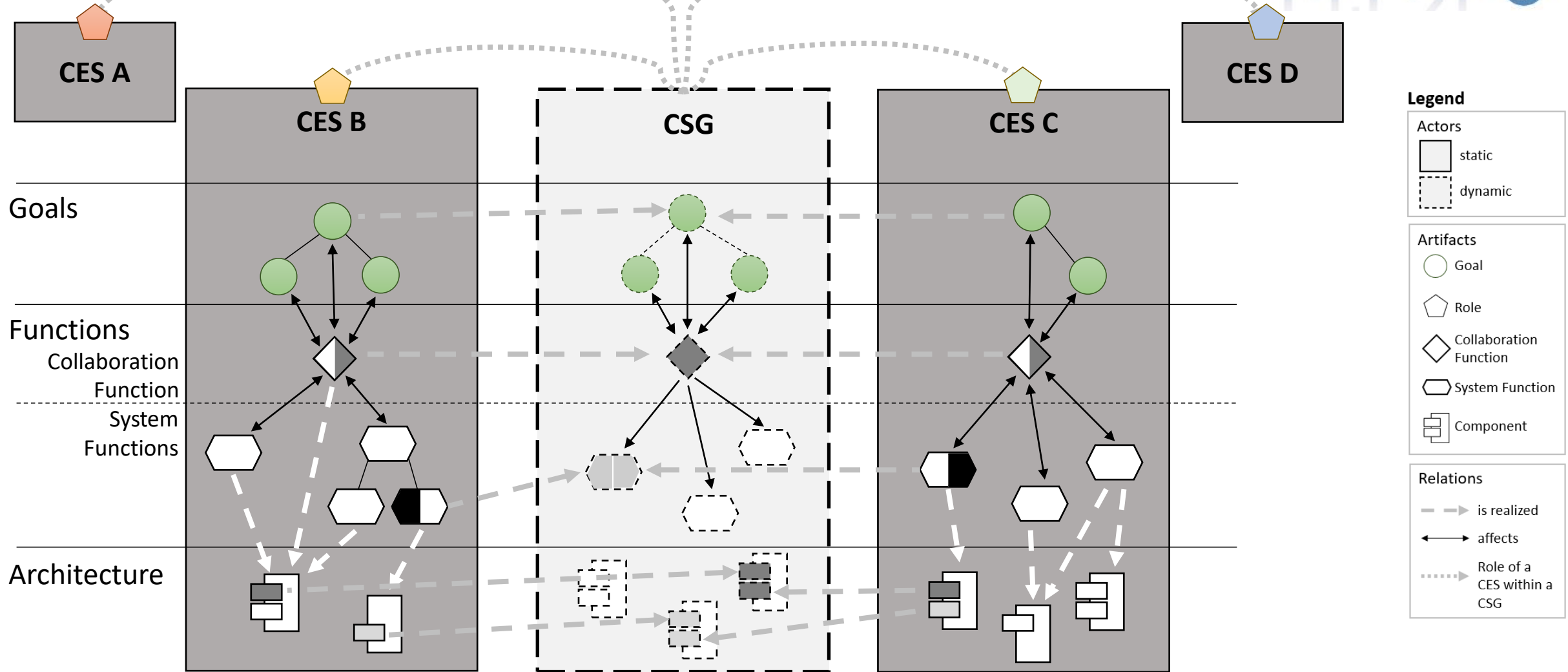
Systems considered in CrEst - Roles



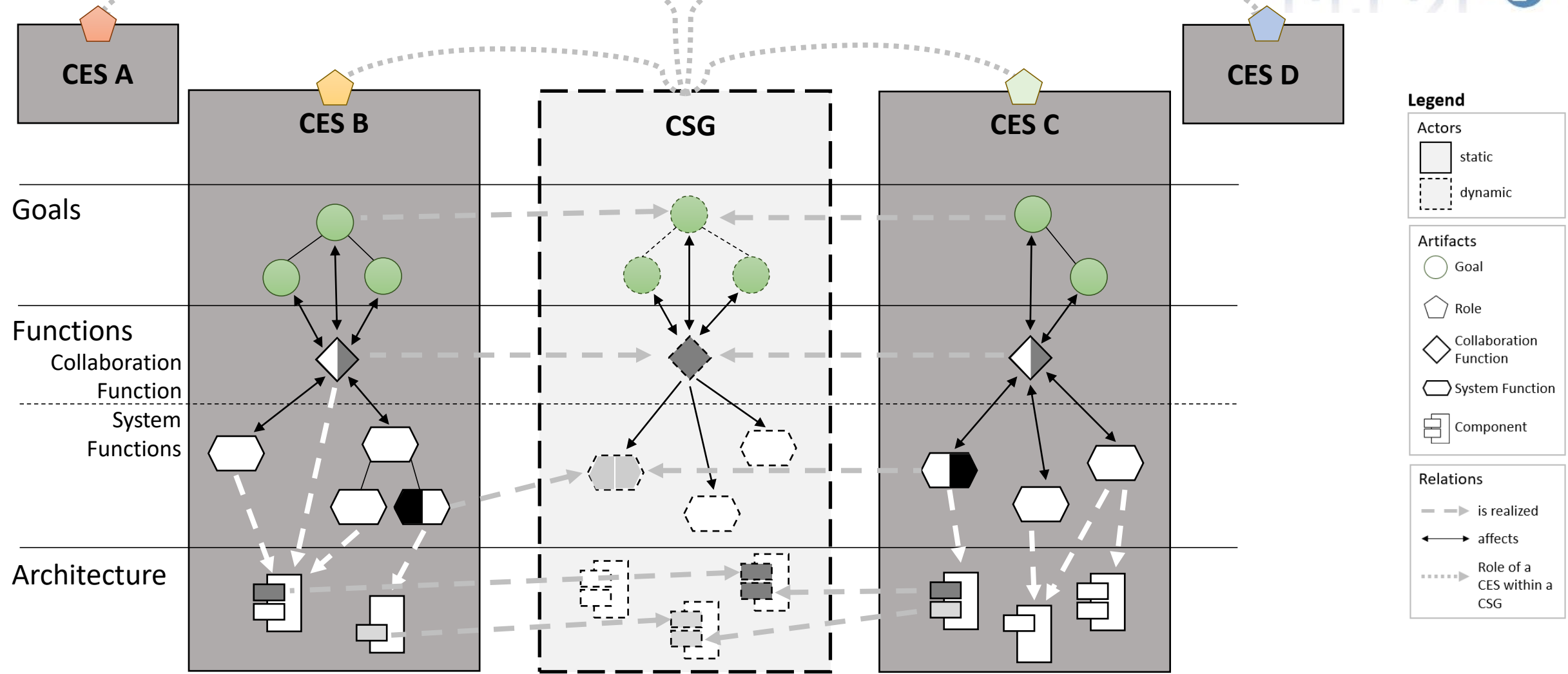
Systems considered in CrEst – Functionality



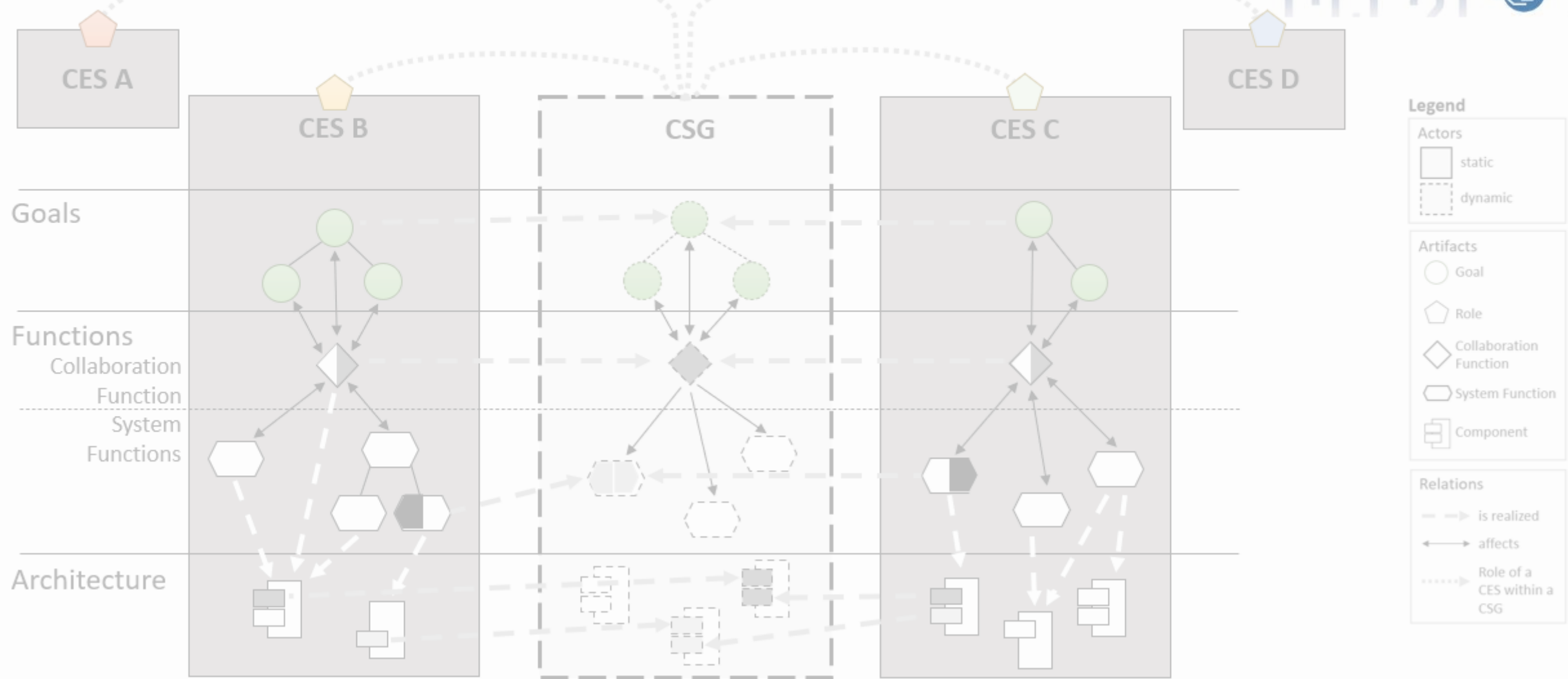
Systems considered in CrEst – CSG Forming



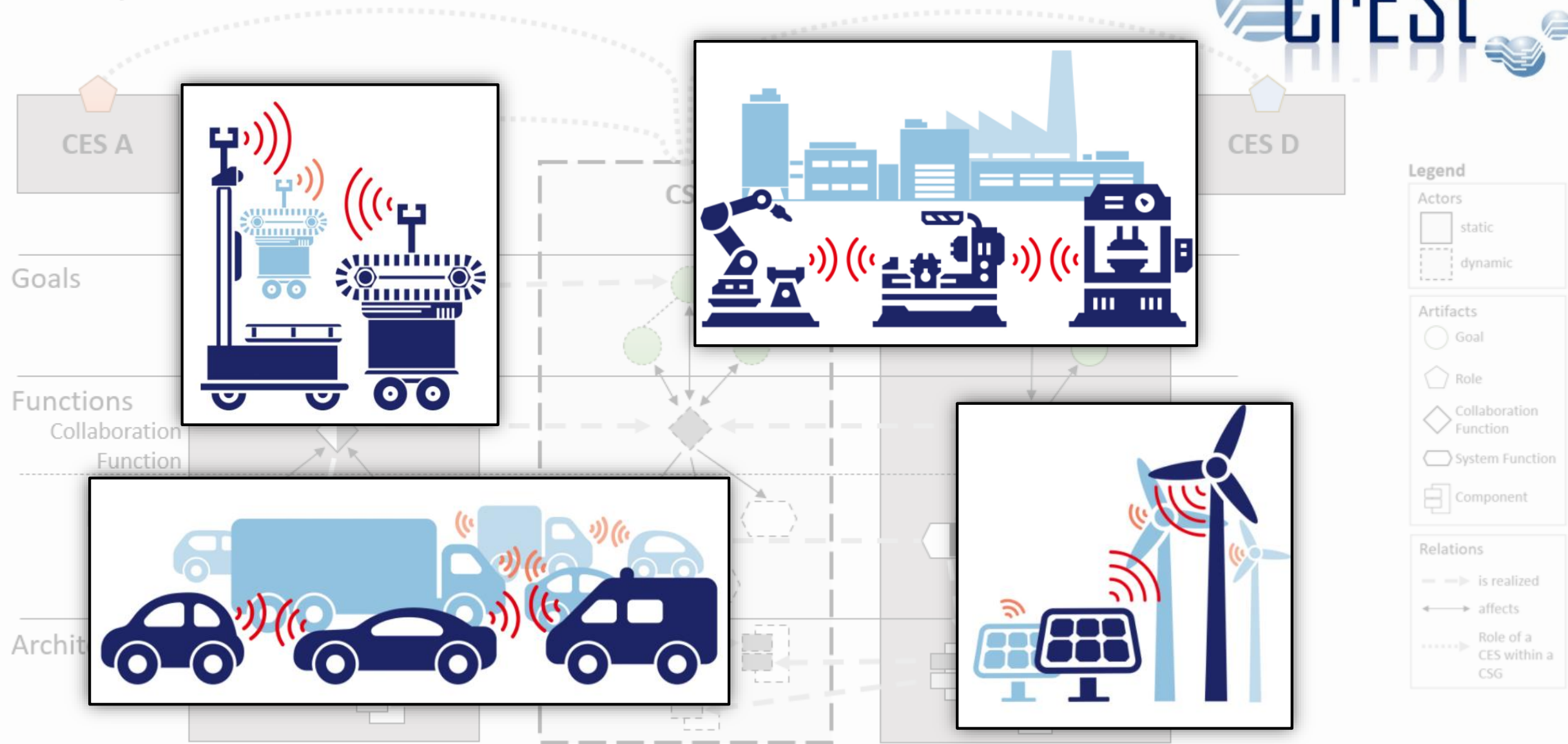
Systems considered in CrEST



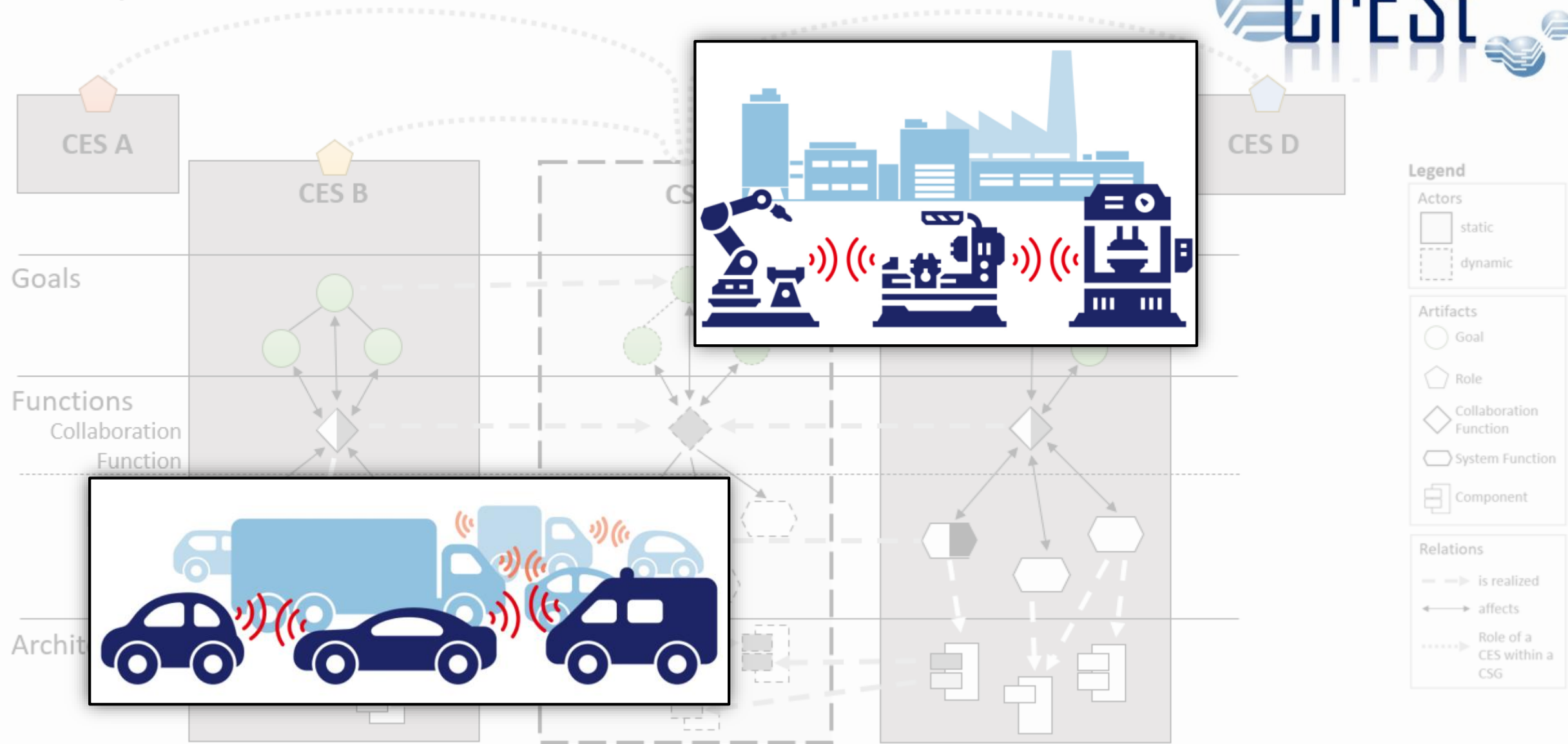
Systems considered in CrEST



Systems considered in CrEst



Systems considered in CrEst

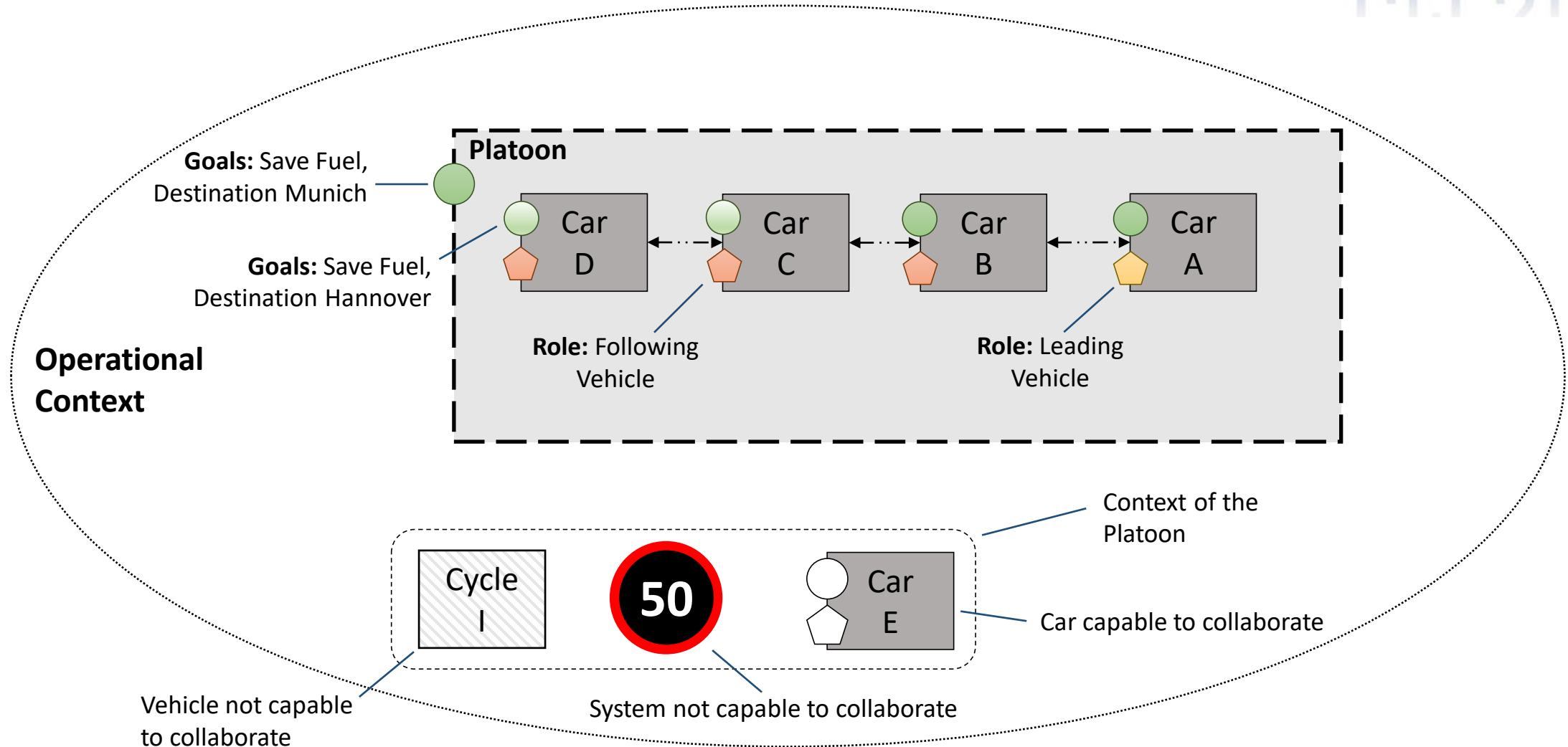


Use Case: Platooning

- Single vehicles coordinate their driving behavior autonomously
- Furthermore, they share information between and collaborate with each other
- They actively involve their environment in the planning and execution process of their behavior
- As part of the collaboration the vehicles form a platoon to:
 - avoid traffic jams
 - reduce energy consumption
 - increase the safety of the driver and the environment

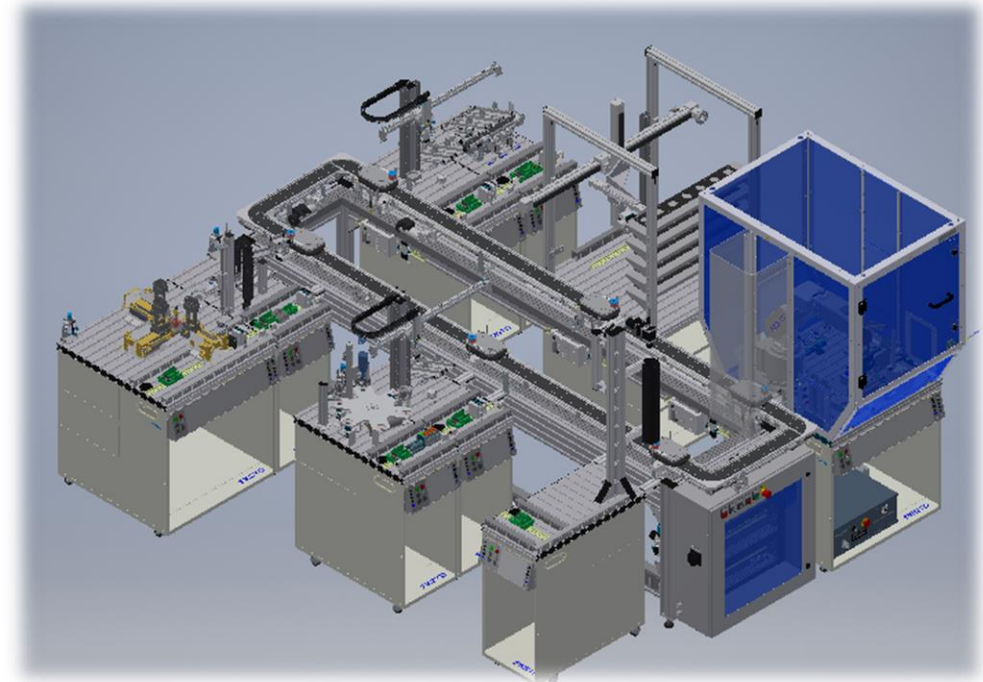


Use Case: Platooning



Use Case: Flexible and Adaptable Factory

- Different modules with various manufacturing functions can contribute to the production of individual products
- With individual production orders, it must be checked whether and how modules can contribute to each single production
- The modules exchange necessary information about the production order (PO) and
 - check their possible contribution
 - plan the production
 - collaboratively manufacture the product



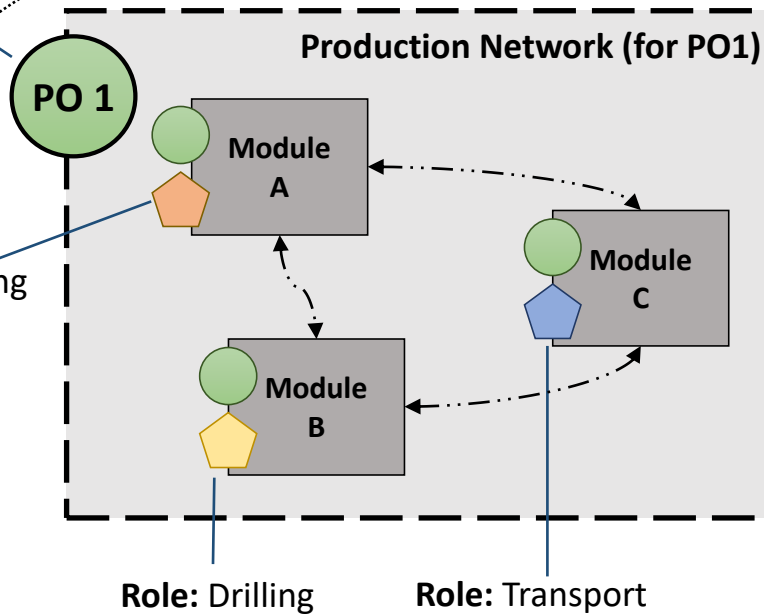
We provided a video at the marketplace to further explain and show the demonstrator

Use Case: Flexible and Adaptable Factory

Operational Context

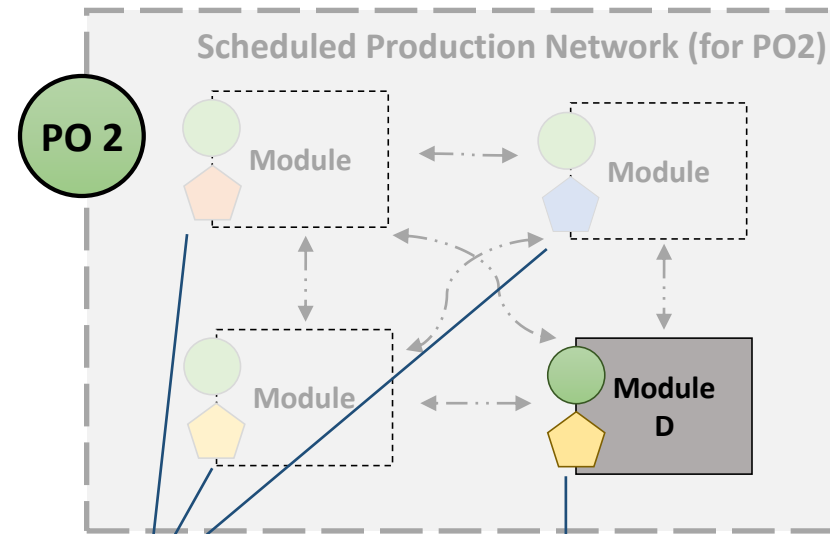
Goal: Production of PO1,
Quality, Cost, Time

Role: Planning



Role: Drilling

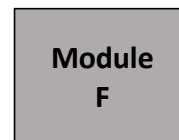
Role: Transport



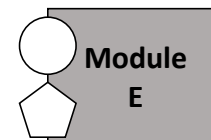
Required Roles
for PO2

Role: Assembly

Module not capable to collaborate



Module capable to collaborate



Categorizing the Project Results

List of all methods developed in CrEst

Results / Methods	Deliverable
Partner network modeling	EC1.AP2.D1
Goal-based strategy exploration	EC3.AP2.D2/D3 EC3.AP3.D1/D2
Matching methods	EC4.AP2.D2/D3/D4
Heterogenous variability modeling	SQ3.AP2.D1/D5
Specification modeling of CSG behavior	EC2.AP3.D1
Modeling CES functional architecture	EC2.AP2.D2 EC2.AP3.D1
Meta-model for functional modeling	SQ1.AP3.D1
Integration CES & CSG for artefact modeling	MQ3.AP2.D2/D3 MQ3.AP3.D1/D2
Enabling co-simulation	MQ3.AP2.D2/D3 MQ3.AP3.D1/D2
Reference architecture for trust-based digital ecosystems	EC6.AP2.D4 MQ2.AP2.D1
Reference architectures	EC1.AP2.D1/D3
Extraction of dynamic architectures	EC2.AP2.D1 EC2.AP4.D1
Safely dealing with shared resources	SQ2.AP1.D2/D3 SQ2.AP2.D1/D2
Goal-based strategy exploration	EC3.AP2.D2/D3 EC3.AP3.D1/D2
Integration CES & CSG for artefact modeling	MQ3.AP2.D2
Code generator composition	MQ1.AP3.D2

Results

Categorizing the Project Results

Results / Methods	Deliverable
Partner network modeling	EC1.AP2.D1
Goal-based strategy exploration	EC3.AP2.D2/D3 EC3.AP3.D1/D2
Matching methods	EC4.AP2.D2/D3/D4
Heterogenous variability modeling	SQ3.AP2.D1/D5
Specification modeling of CSG behavior	EC2.AP3.D1
Modeling CES functional architecture	EC2.AP2.D2 EC2.AP3.D1
Meta-model for functional modeling	SQ1.AP3.D1
Integration CES & CSG for artefact modeling	MQ3.AP2.D2/D3 MQ3.AP3.D1/D2
Enabling co-simulation	MQ3.AP2.D2/D3 MQ3.AP3.D1/D2
Reference architecture for trust based digital ecosystems	EC6.AP2.D4 MQ2.AP2.D1
Reference architectures	EC1.AP2.D1/D3
Extraction of dynamic architectures	EC2.AP2.D1 EC2.AP4.D1
Safely dealing with shared resources	SQ2.AP1.D2/D3 SQ2.AP2.D1/D2
Goal-based strategy exploration	EC3.AP2.D2/D3 EC3.AP3.D1/D2
Integration CES & CSG for artefact modeling	MQ3.AP2.D2
Code generator composition	MQ1.AP3.D2

Reference to deliverables, where methods are described in detail

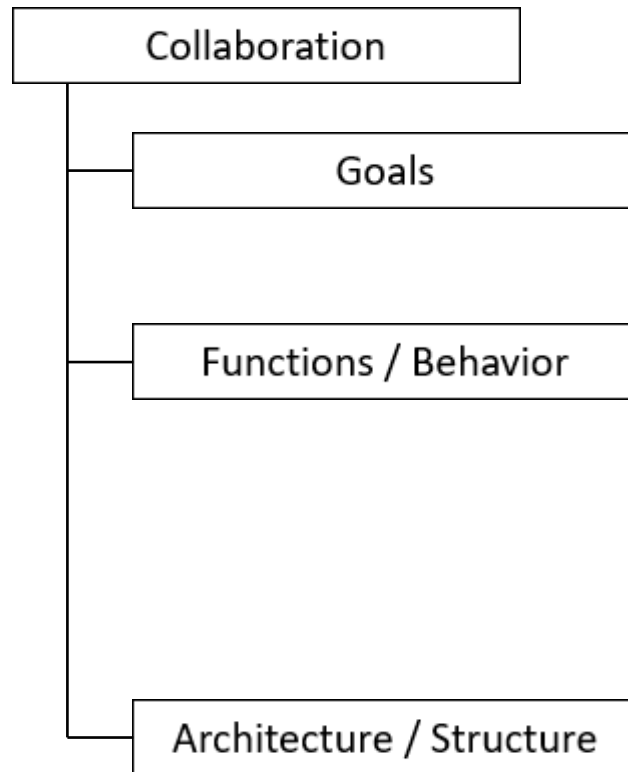
Results

Categorizing the Project Results

Collaborating Systems

Taxonomy of Challenges

Categorizing the Project Results



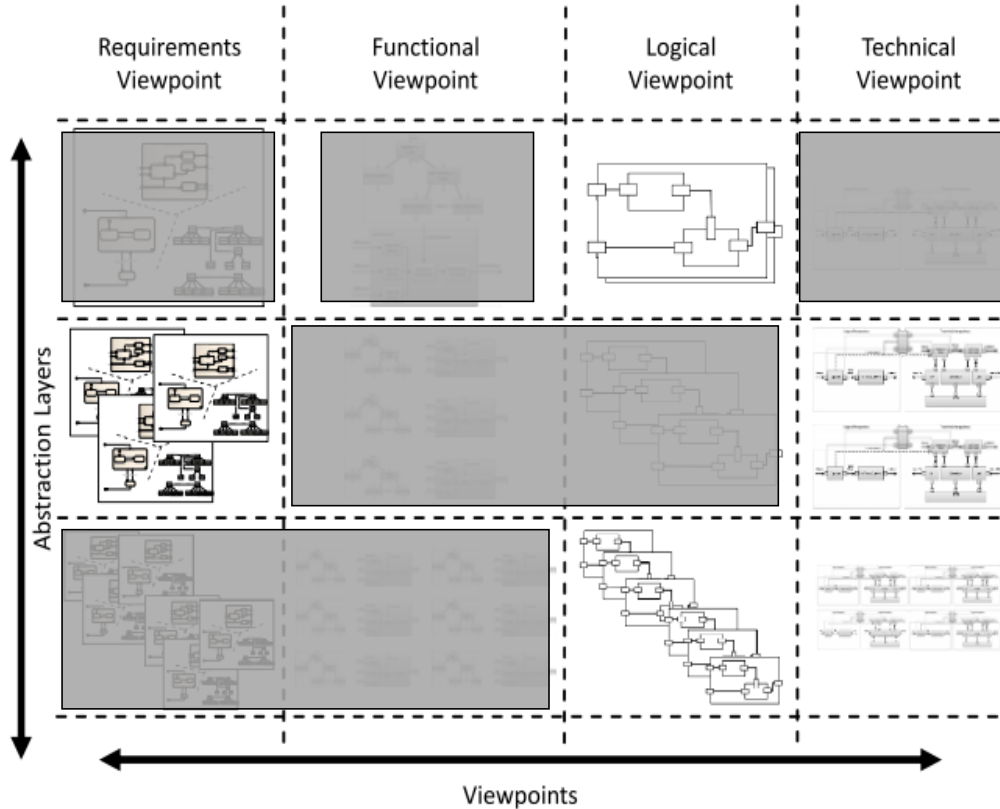
Results / Methods	Deliverable
Partner network modeling	EC1.AP2.D1
Goal-based strategy exploration	EC3.AP2.D2/D3 EC3.AP3.D1/D2
Matching methods	EC4.AP2.D2/D3/D4
Heterogenous variability modeling	SQ3.AP2.D1/D5
Specification modeling of CSG behavior	EC2.AP3.D1
Modeling CES functional architecture	EC2.AP2.D2 EC2.AP3.D1
Meta-model for functional modeling	SQ1.AP3.D1
Integration CES & CSG for artefact modeling	MQ3.AP2.D2/D3 MQ3.AP3.D1/D2
Enabling co-simulation	MQ3.AP2.D2/D3 MQ3.AP3.D1/D2
Reference architecture for trust-based digital ecosystems	EC6.AP2.D4 MQ2.AP2.D1
Reference architectures	EC1.AP2.D1/D3
Extraction of dynamic architectures	EC2.AP2.D1 EC2.AP4.D1
Safely dealing with shared resources	SQ2.AP1.D2/D3 SQ2.AP2.D1/D2
Goal-based strategy exploration	EC3.AP2.D2/D3 EC3.AP3.D1/D2
Integration CES & CSG for artefact modeling	MQ3.AP2.D2
Code generator component	

Categorized Results

Identifying Viewpoints Extensions

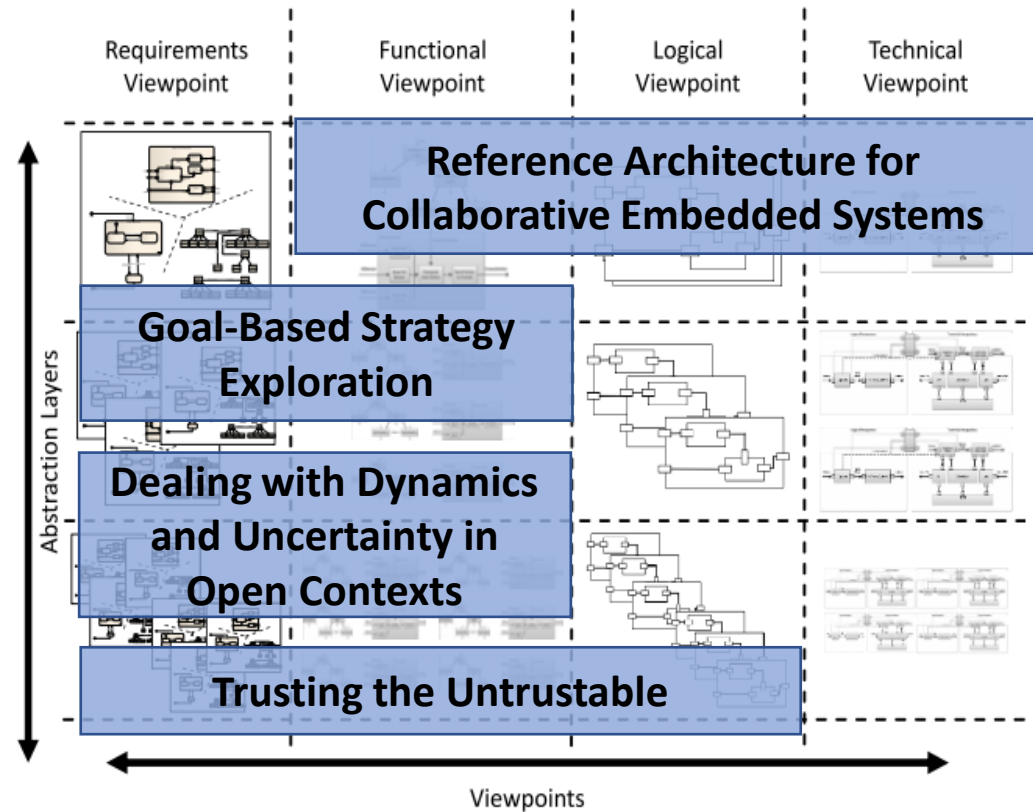
Crest Process Building Block - Template	
Description	
ID	EC3_Goals
Name	Crest Process Building Block - Template
Description	
ID	tbd
Name	Ontology Building
Description	<p>have to cope with open contexts, i.e. the context that these systems interact with through sensors and actuators or communication networks changes its morphology during runtime. In order to be able to cope with open contexts, CPSSs rely on ontologies that represent the information being shared among them. This method contains sub-methods for requirements specification, lightweight and heavyweight ontology building for project requirements, reusable ontologies, information resources (e.g. DL) and Heavyweight Ontology (e.g. OWL-DL) reusable Ontology Design Patterns (e.g. OVL-DL)</p>
Input	
Output	
Precondition	
Postcondition	
Tools & Templates	
Type	
Method automation	
Related Cr	
Type	
Method automation	
Requirement	
Functional	
Logical	
Technical	
Related CrEst Viewpoints	
Uncertainty	
Dynamic Systems and Contexts	
Collaboration	
Distributed Development Processes	
Related MQ1 Topic Clusters	
Cooperative Vehicle Automation	
Adaptable and Flexible Factory	
Distributed Power Generation	
Cooperating Transport Robots	
Related CrEst Use Cases	
Source (Document/ID)	EC4.AP2.D1 (pp. 31-35, 85-87) & EC4.AP2.D3 (pp. 31-35, 85-87)
Contact	Constantin Hildebrandt

Results

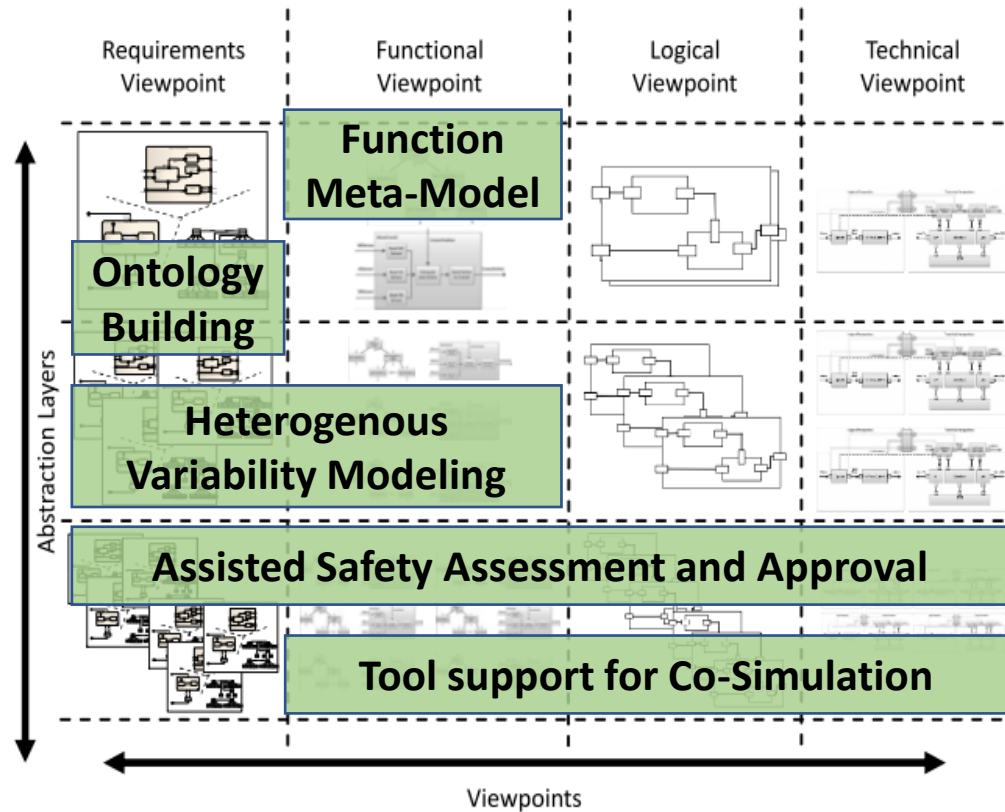


Viewpoint Extensions

Viewpoints Extensions - Examples

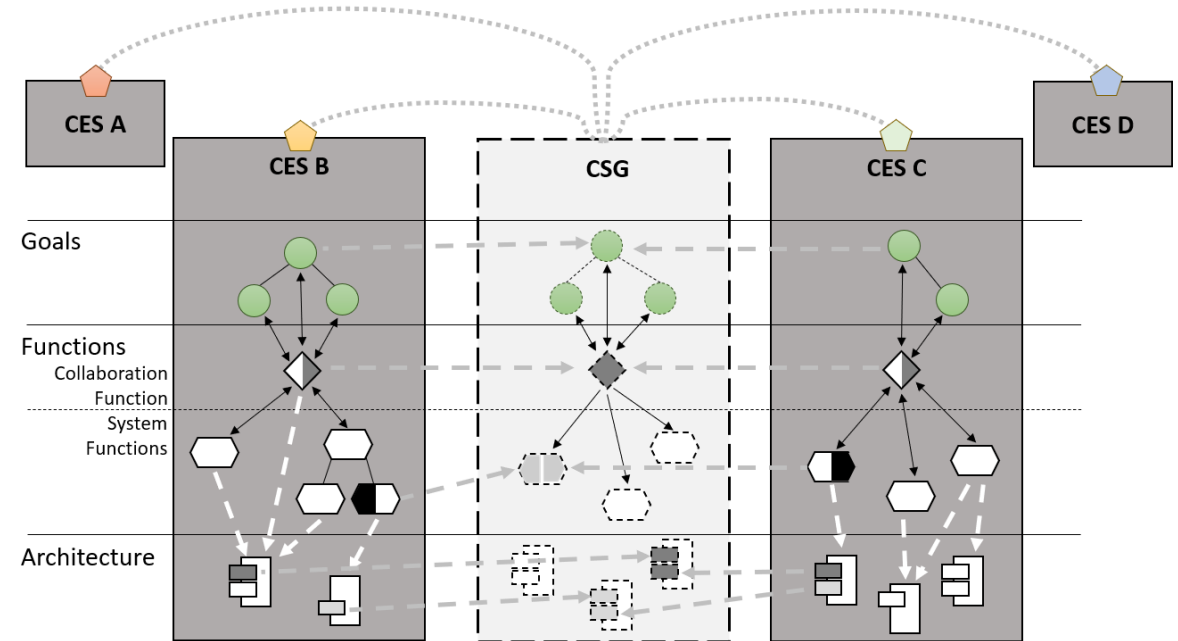


Viewpoints Extensions - Examples



Conclusion

- The CrEST Story describes
 - the different systems (CES and CSG) considered in CrEST
 - the challenges to address in the model-based development of these systems
- In CrEST various methods and models were developed to meet these challenges
- The story provides a comprehensive framework in which all the results are ordered
- This framework is an extension of the SPES Modeling Framework
- The CrEST Modeling Framework for the model-based development of collaborative embedded systems serves as a basis for the industry and further research



*Viele
Dank!*