Approach Based Patterns for System-of-Systems Reconfiguration

F. Petitdemange[1] I. Borne[1] J. Buisson[2]

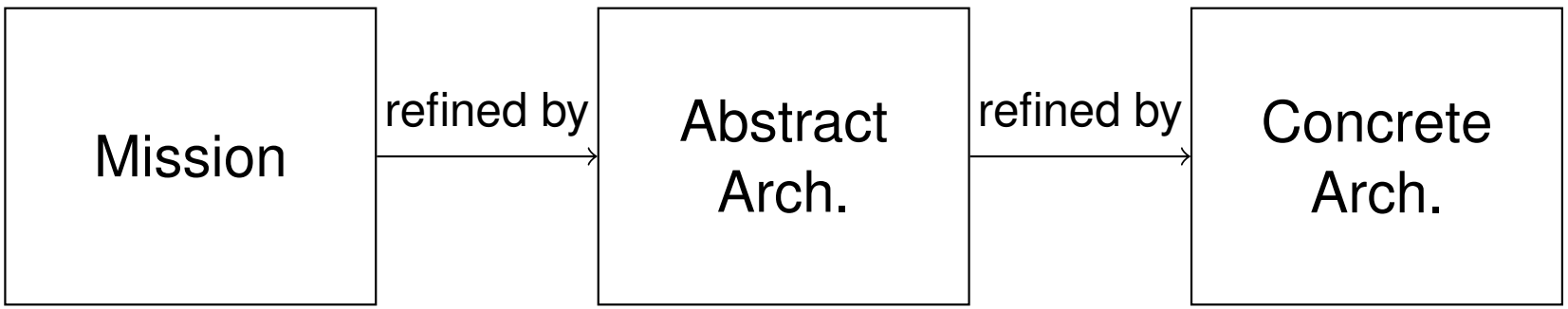
[1]IRISA
University of South Brittany
Vannes, France

[2]IRISA
Military Academy of St-Cyr
Vannes, France

Software Engineering for Systems-of-Systems, 2015

Plan

SoS Engineering Process



Architecture description language : SoSADL

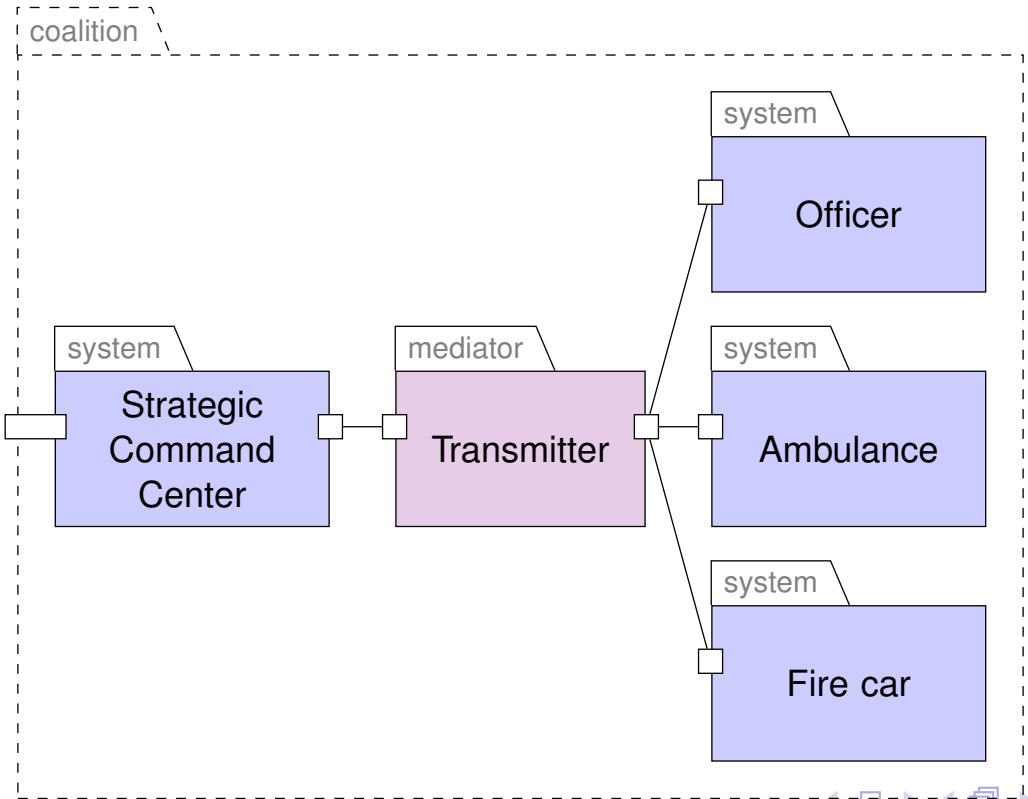
Constituent System (CS) : preexisting systems that are integrated into the SoS in order to contribute to the global mission of the SoS

Mediator : communicating elements that specify and coordinate the interactions between CSs

Coalition : defines a set of constraints about the CSs and mediators required to accomplish an emergent behavior.

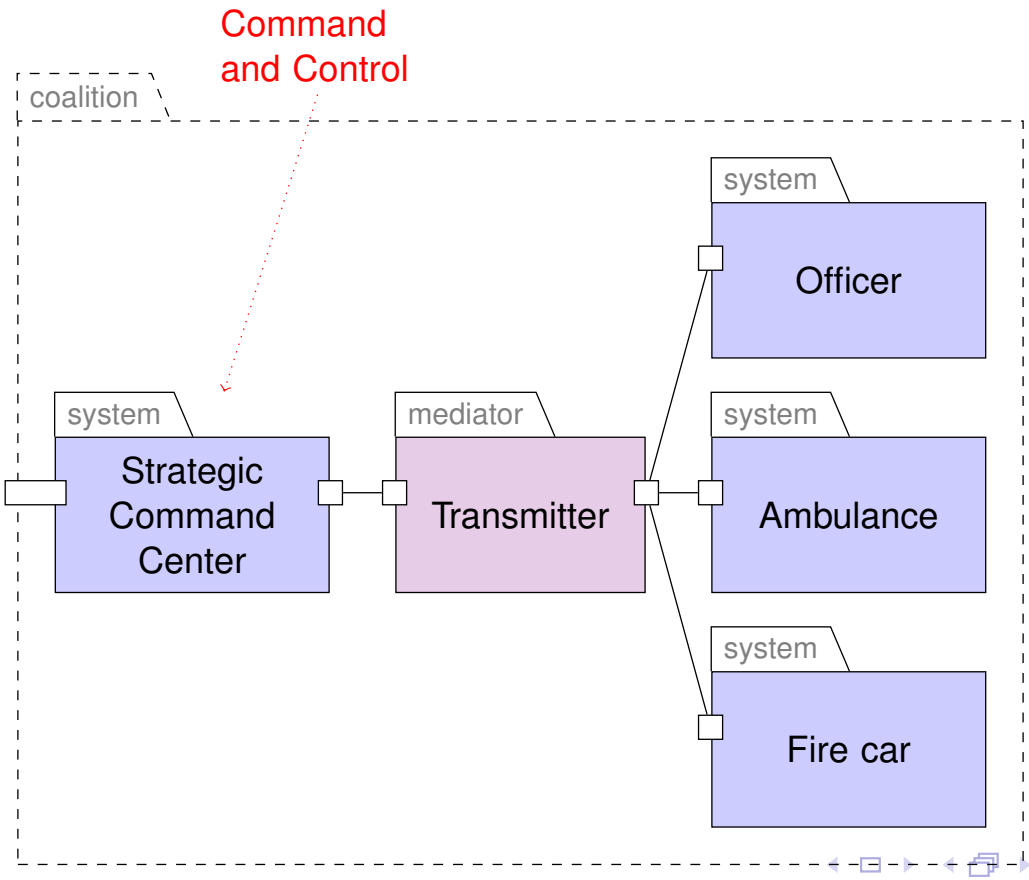
Emergency service SoS

Mission : to optimize the deployment of its resources as fast as possible in order to preserve human life and material



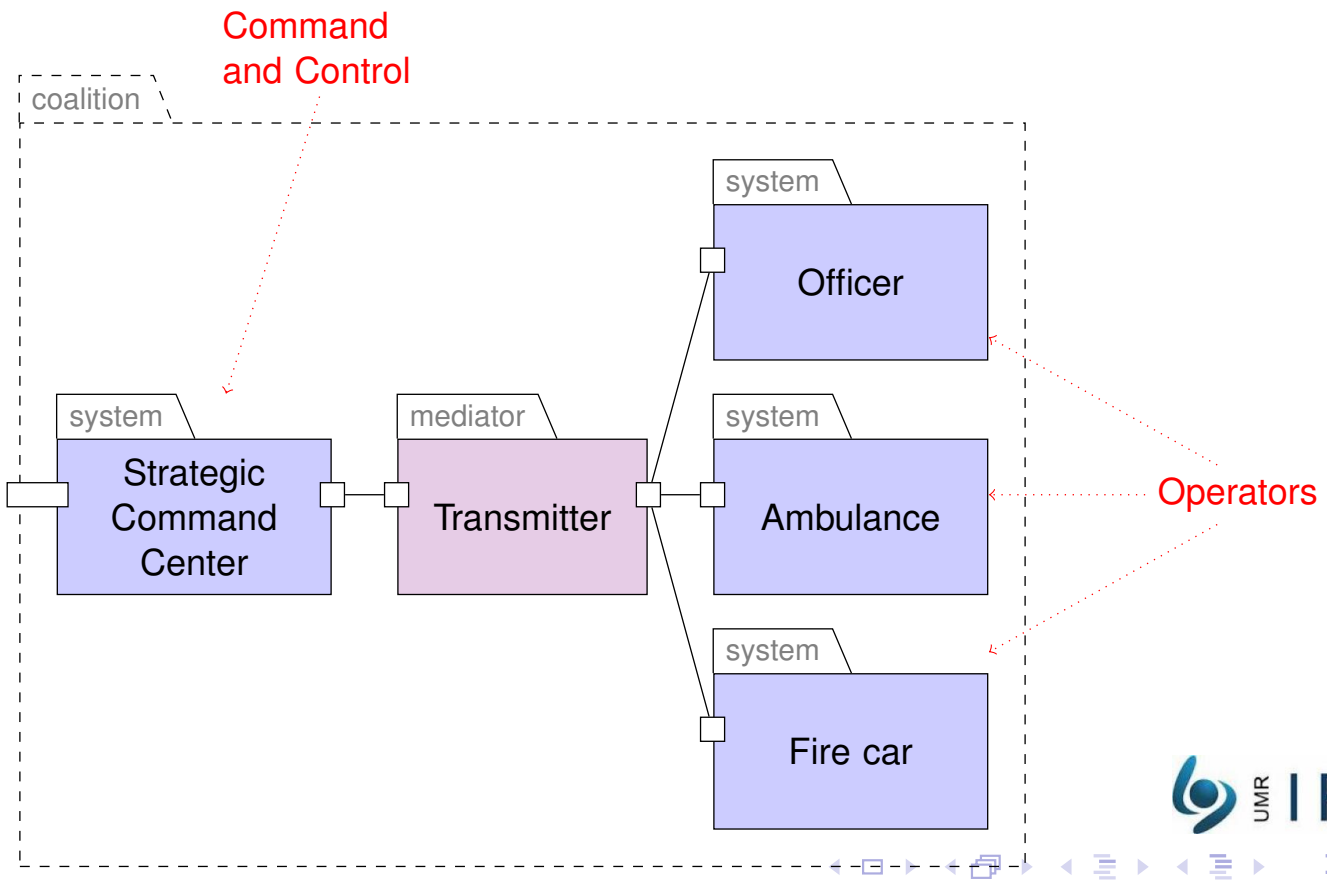
Emergency service SoS

Mission : to optimize the deployment of its resources as fast as possible in order to preserve human life and material

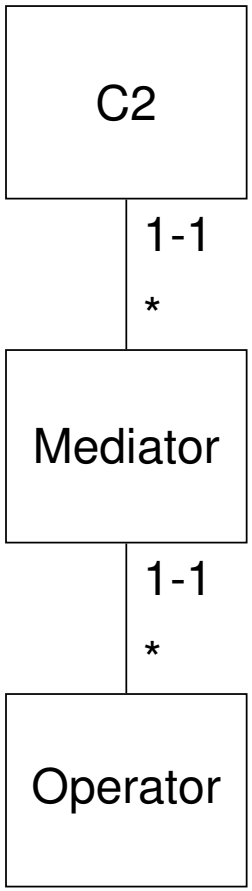


Emergency service SoS

Mission : to optimize the deployment of its resources as fast as possible in order to preserve human life and material



Architectural pattern: Command and Control



Allow the supervision of a complex system by giving an overview of the activities in the system and delegating the governance responsibilities.

Need of architectural reconfiguration

Two kinds of reconfiguration :

- abstract architecture (e.g evolutionary development)
- concrete architecture (e.g environment evolution)

Consequence :

- architecture evolves continuously

How to maintain **architectural consistency**?

- maintain the architectural pattern in the concrete architecture?
- make evolve the architectural pattern?

Patterns for reconfiguration

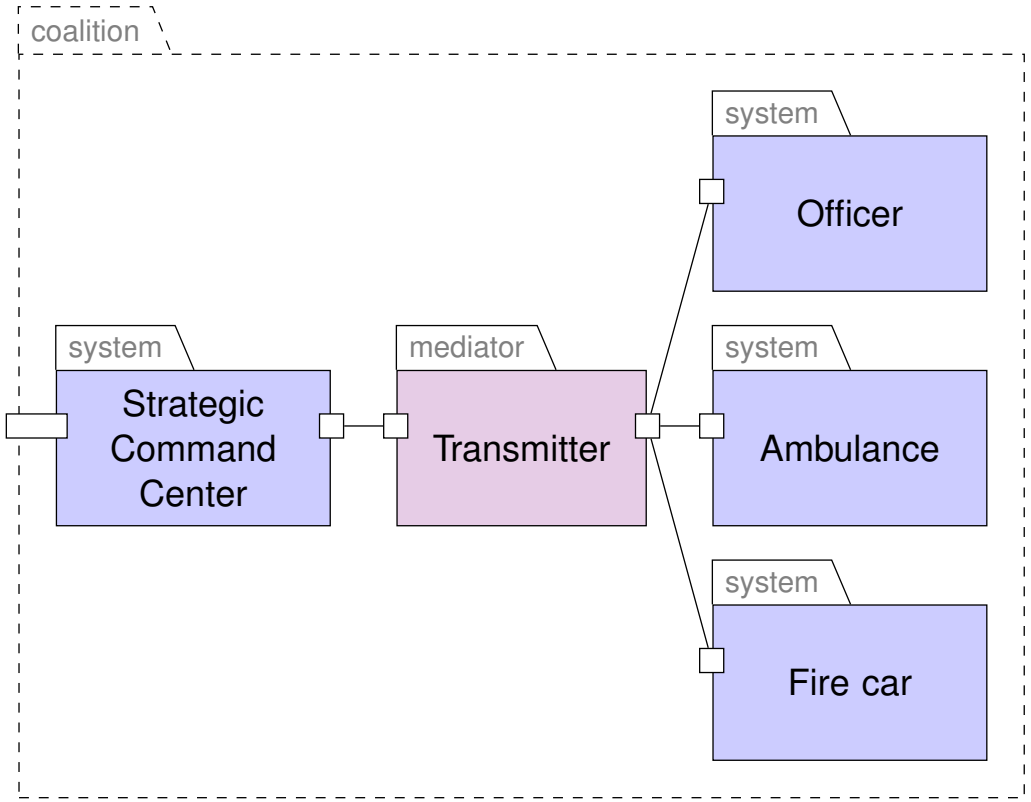
Main idea :

- capture a set of best practices to maintain **consistency** in SoS at runtime

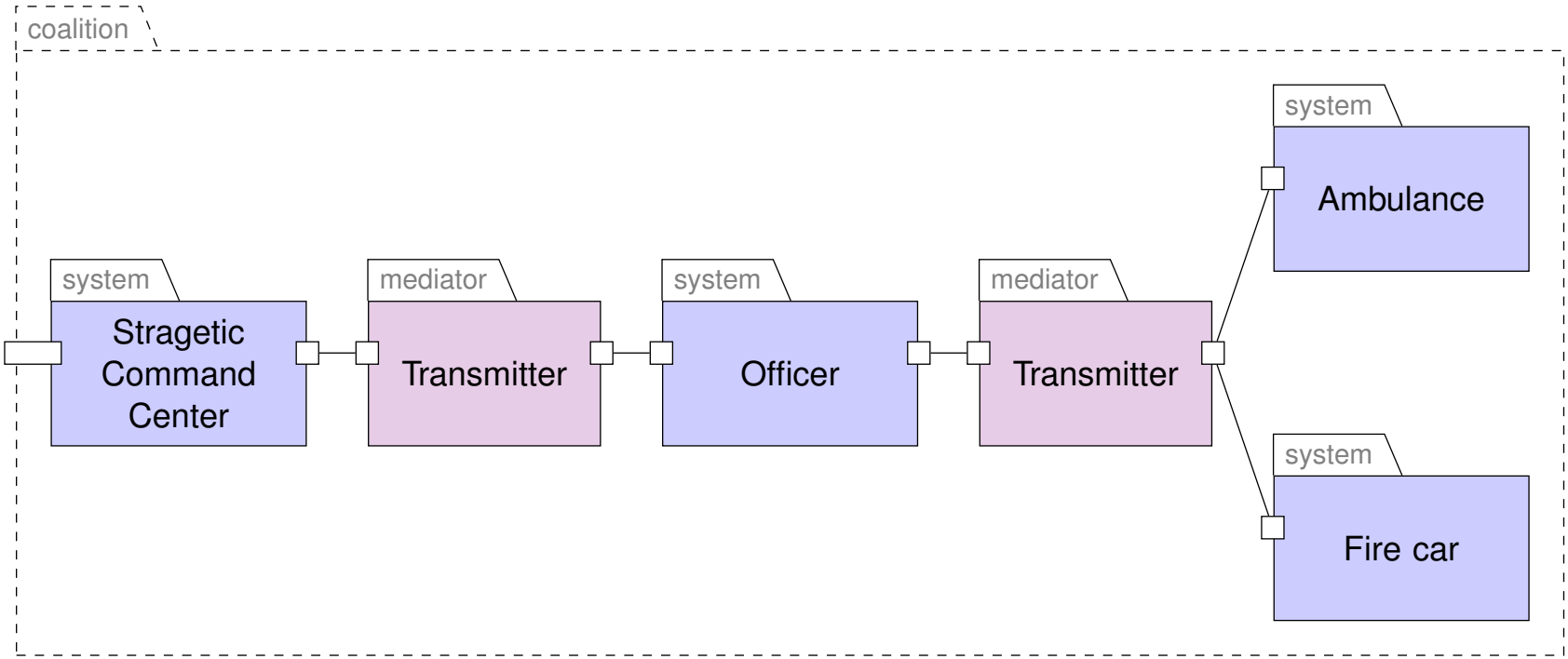
Approach :

- define reconfiguration at **pattern level** combination of multiple smaller reconfiguration patterns
- assist the architect to **maintain** architectural patterns
- assist **evolution** in the architectural pattern

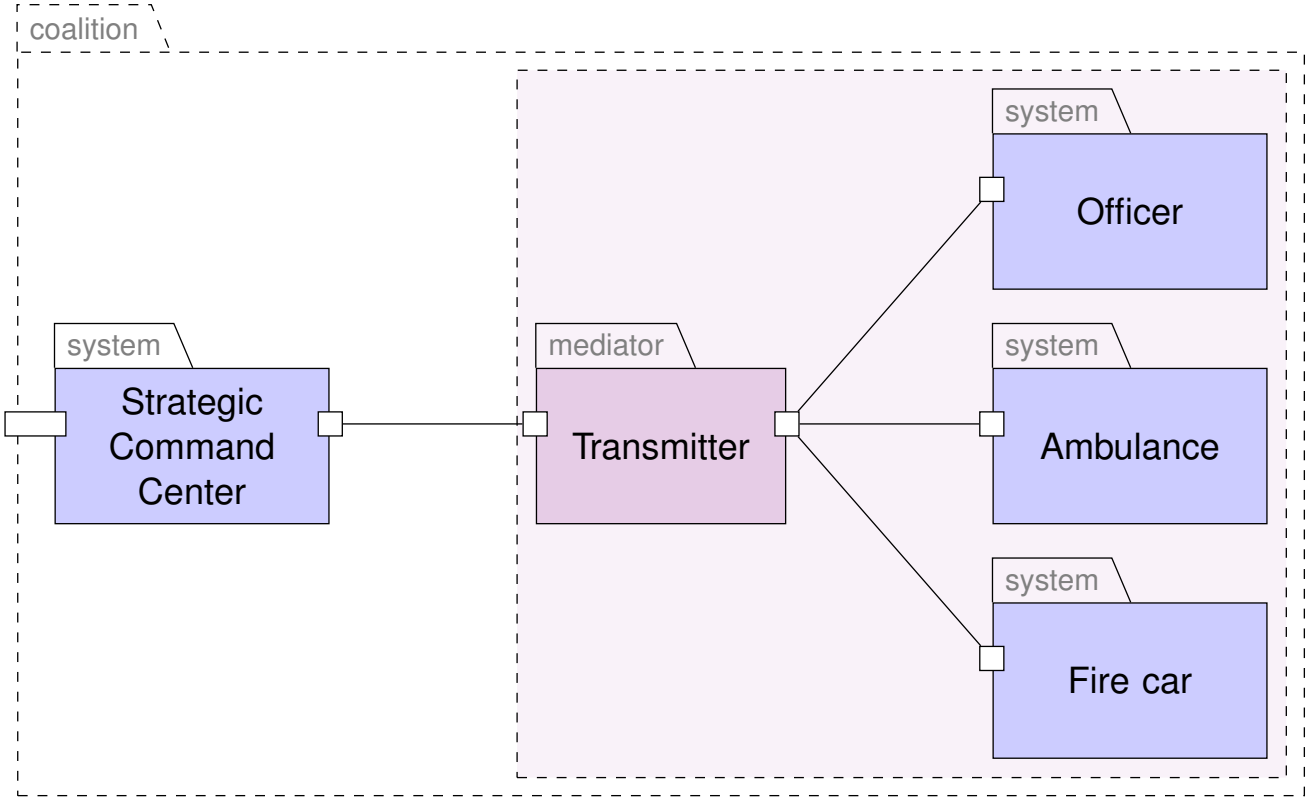
Architectural pattern evolution



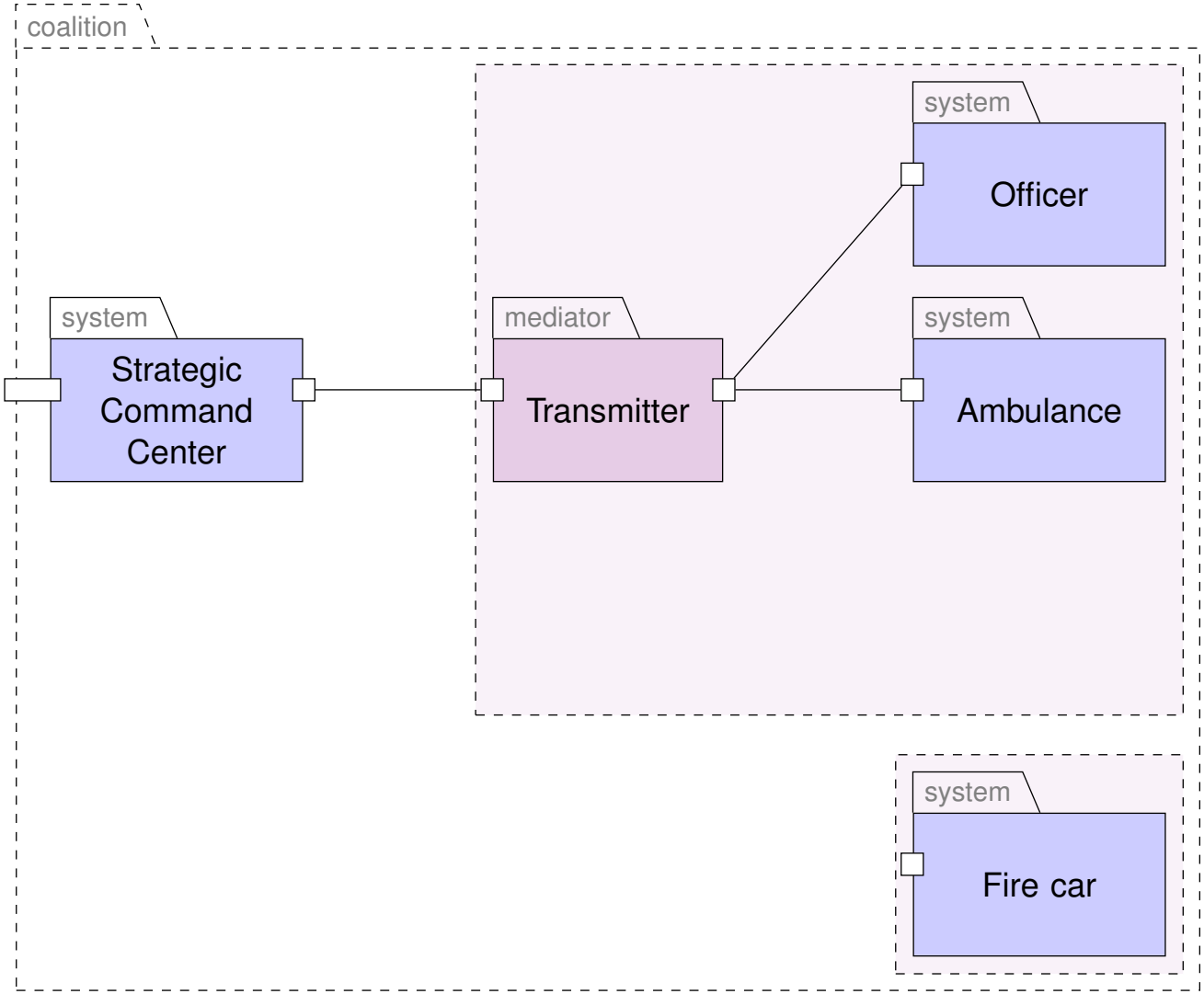
Architectural pattern evolution



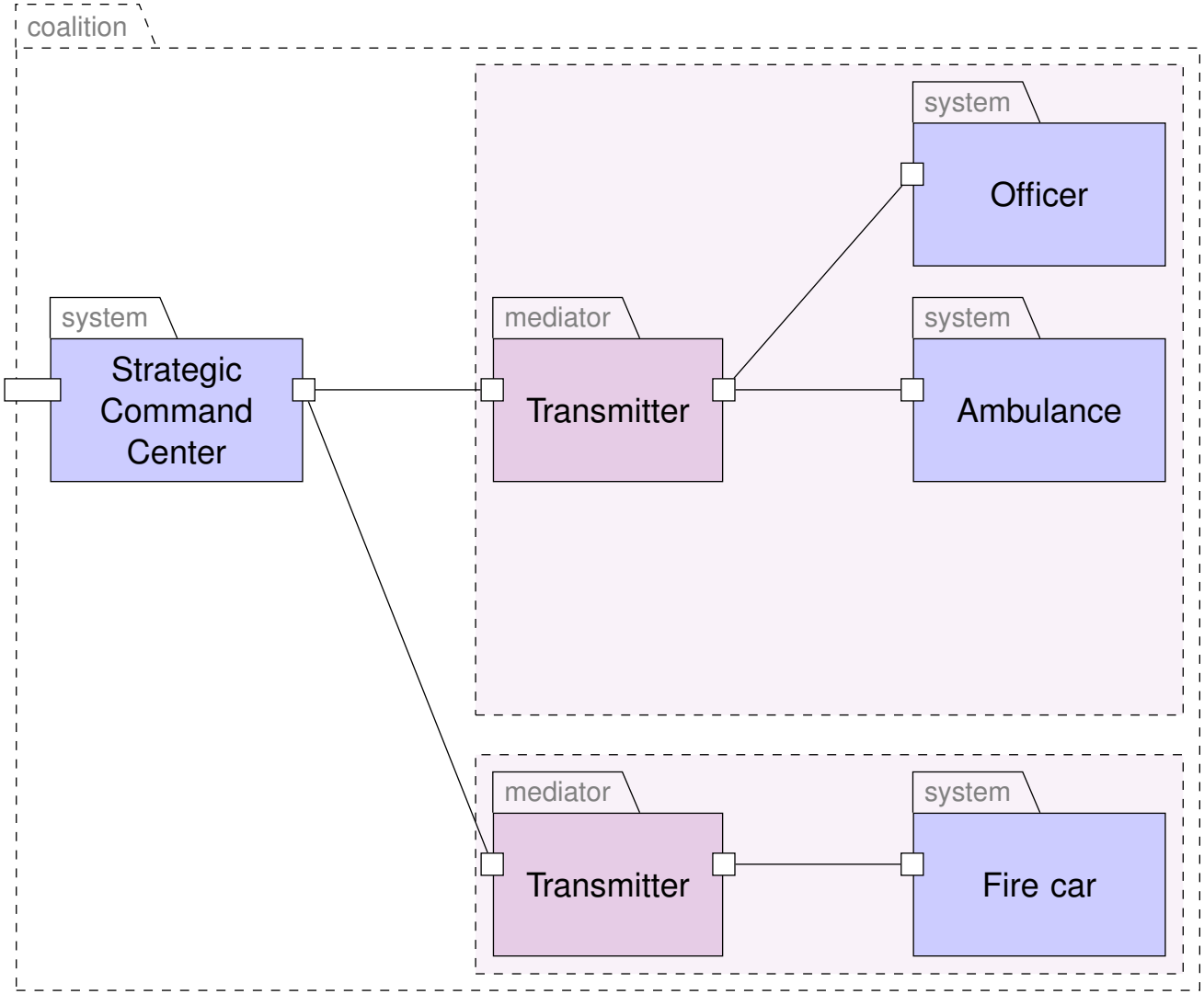
Architectural pattern consistency



Architectural pattern consistency



Architectural pattern consistency



Advantages

- **maintain properties** (emergent behavior, quality model, etc.)
- lead the **evolution** of the architecture
- improve **re-usability** : capture and reuse expert knowledge to solve recurrent problem
- improve verification, documentation and analyses on the architecture

Difficulties and Future work

Difficulties :

- unanticipated evolution in the environment
- lack of classical quiescence mechanisms for this kind of system

Future work :

- develop other reconfiguration examples with bigger case studies
- instrument patterns with SoSADL by defining and organizing reconfiguration patterns in a catalog or a pattern language