

Automatic Synthesis of Adaptable and Evolving Choreographies

Marco Autili

Massimo Tivoli

Paola Inverardi

University of L'Aquila

**Governing Adaptive and
Unplanned Systems of Systems**

GAUSS Kickoff Meeting
22-23 Feb. 2017 – MILANO BICOCCA

Where do we come from?

(FP7 project)

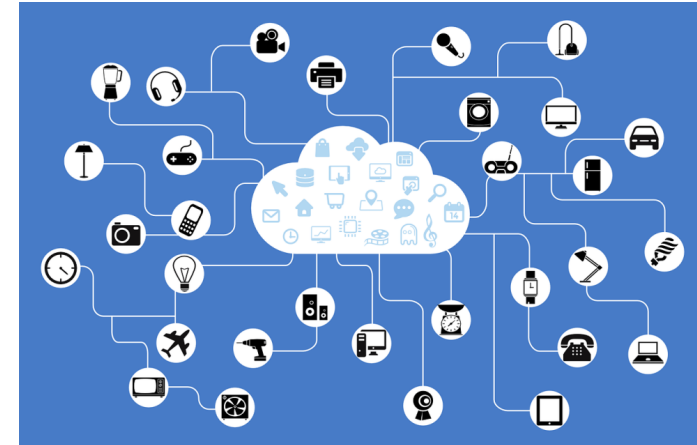


(H2020 project)



Automated Synthesis of Dynamic
and Secured Choreographies
for the Future Internet
(H2020 project)

A man with dark hair is wearing a futuristic blue head-mounted display (HMD). The device has a transparent visor and is connected to a black headband. Overlaid on the visor and around the man's head are various digital interface elements, including a circular radar-like graphic on the left and a vertical bar chart on the right. The man is wearing a white turtleneck sweater and is looking directly at the camera with a neutral expression. His hands are raised, with fingers pointing towards the sides of his head, as if interacting with the device. The background is a solid dark grey.

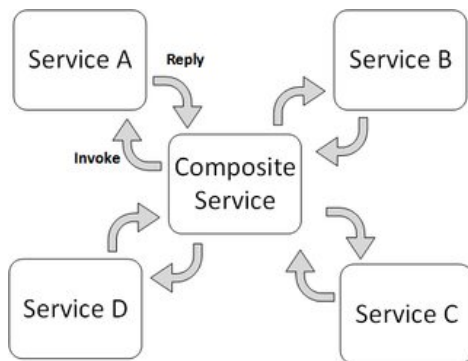


The ability to seamlessly **compose and coordinate** heterogeneous systems is of paramount importance

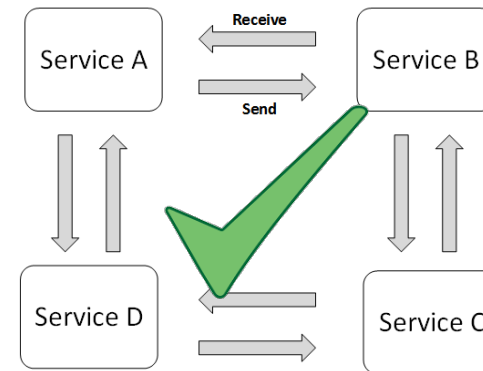
* European Commission
Europe - F

*** European Commission. Digital Agenda for Europe - Future Internet Research and Experimentation (FIRE) initiative, 2015**

Composition approaches



Orchestration (centralized)



Choreography (fully distributed)

Local centralized view
from the perspective of
one participant

Global decentralized view from a
multi-participant perspective (albeit
without a central controller)

OUR GOAL: automatic architecture synthesis (from
specification models to actual code) to support the
realization of choreographies by reusing-existing services

Motivations

Building applications by **reusing** services
(often black-box)

Composing services in a **distributed way**



Support for **automation** is needed
(time-to-market, correctness by construction, etc.)

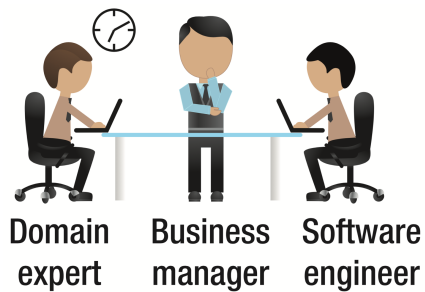


Aiding software producers to **realize**, **deploy**,
execute, and **monitor** choreography-based
systems by **reusing** existing services

Development scenario

Choreography modelers cooperate each other to **set business goals**, e.g.,

- assisting travelers from arrival, to staying, to departure

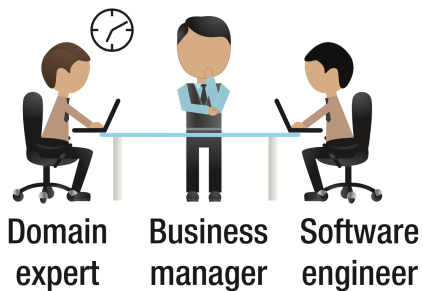


Development scenario

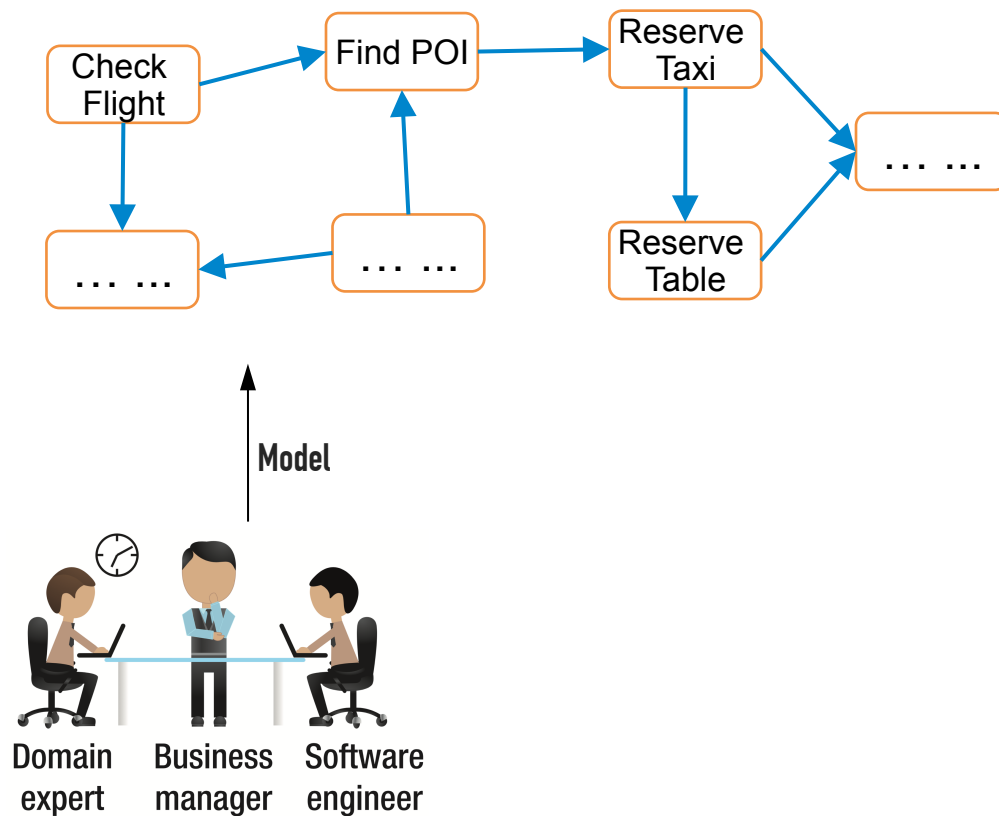
Identify **tasks** and **participants** required to achieve the goal, e.g.,



- reserving a taxi from the local taxi company,
- purchasing digital tickets at the train station,
- performing transactions through services based on near field communication in a shop



Development scenario



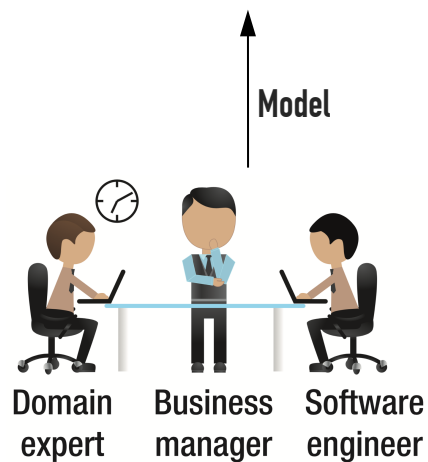
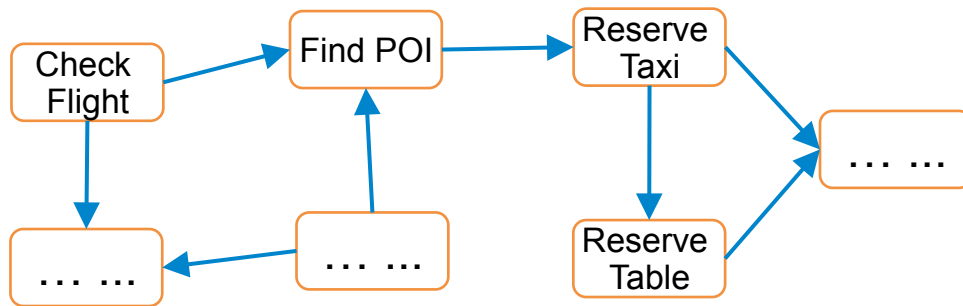
Specify how participants must collaborate as **admissible flows** of the identified business tasks through:

- *BPMN2 Choreography Diagrams*

Development scenario

The inventory contains services published by providers, e.g.,

- transportation companies
- airport retailers



Service providers

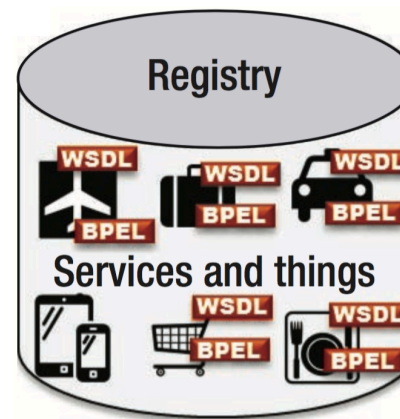


Publish

Register

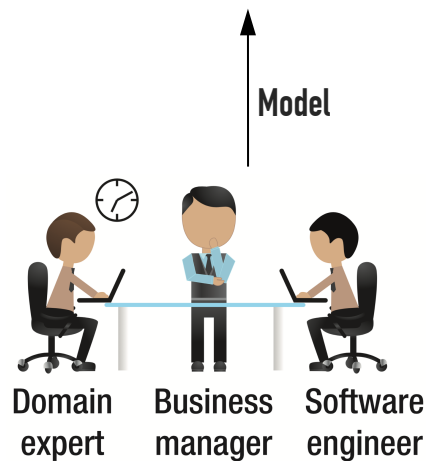
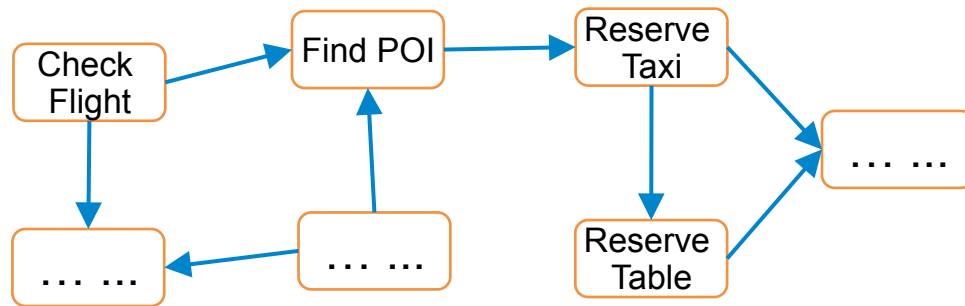


End users

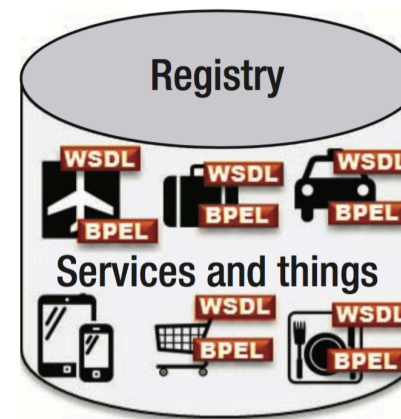
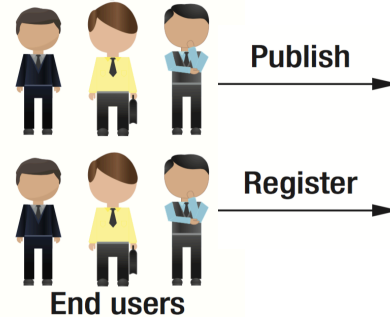


Modelling phase

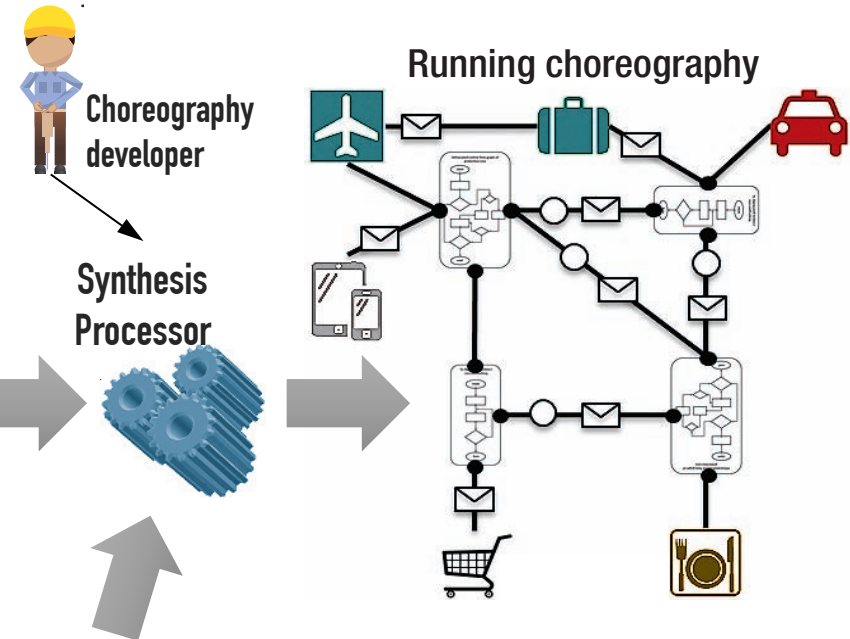
- Out of the **specified business goal**, and
- by **(re-)using** the set of services available in the registry ...



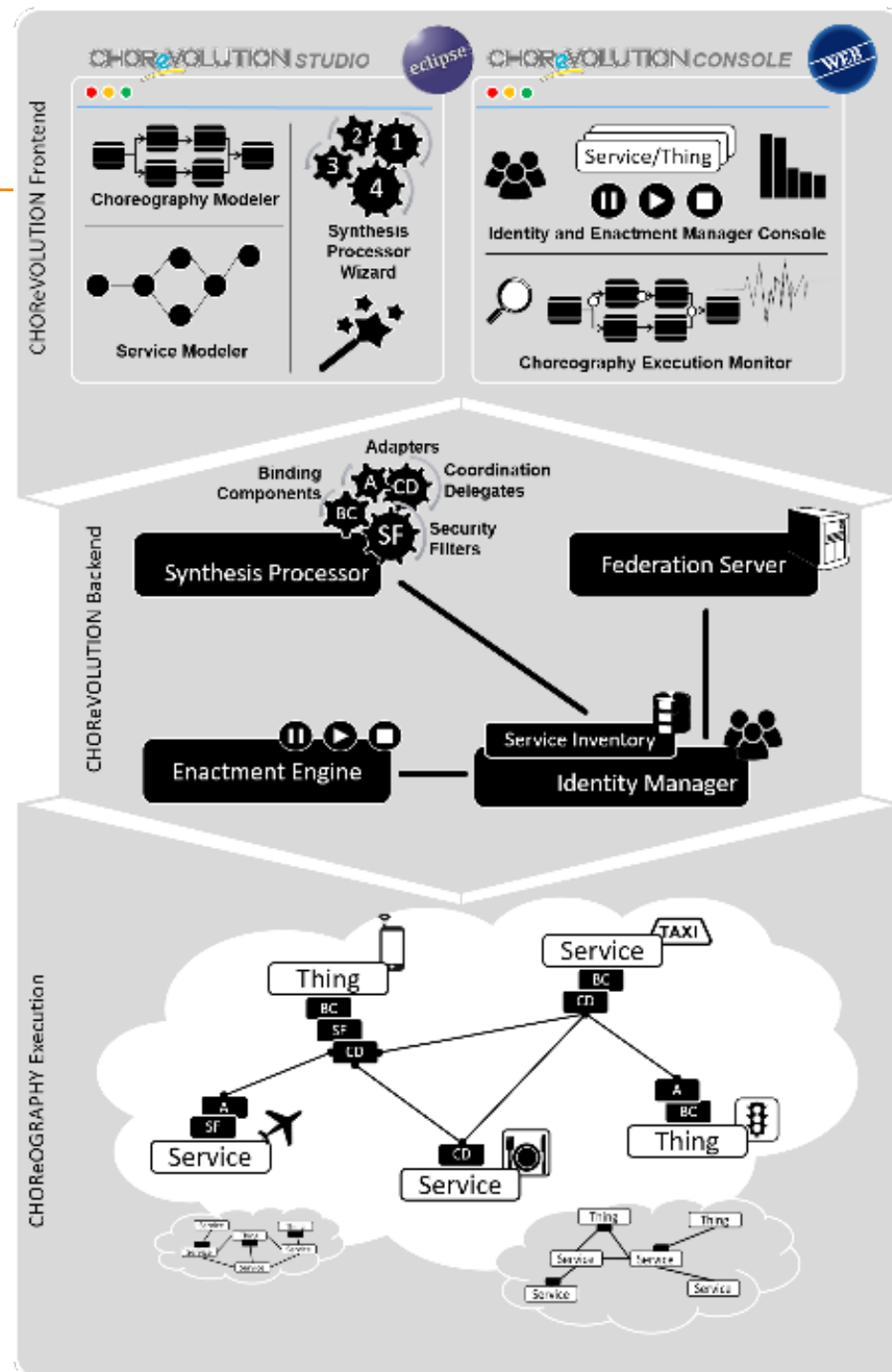
Service providers



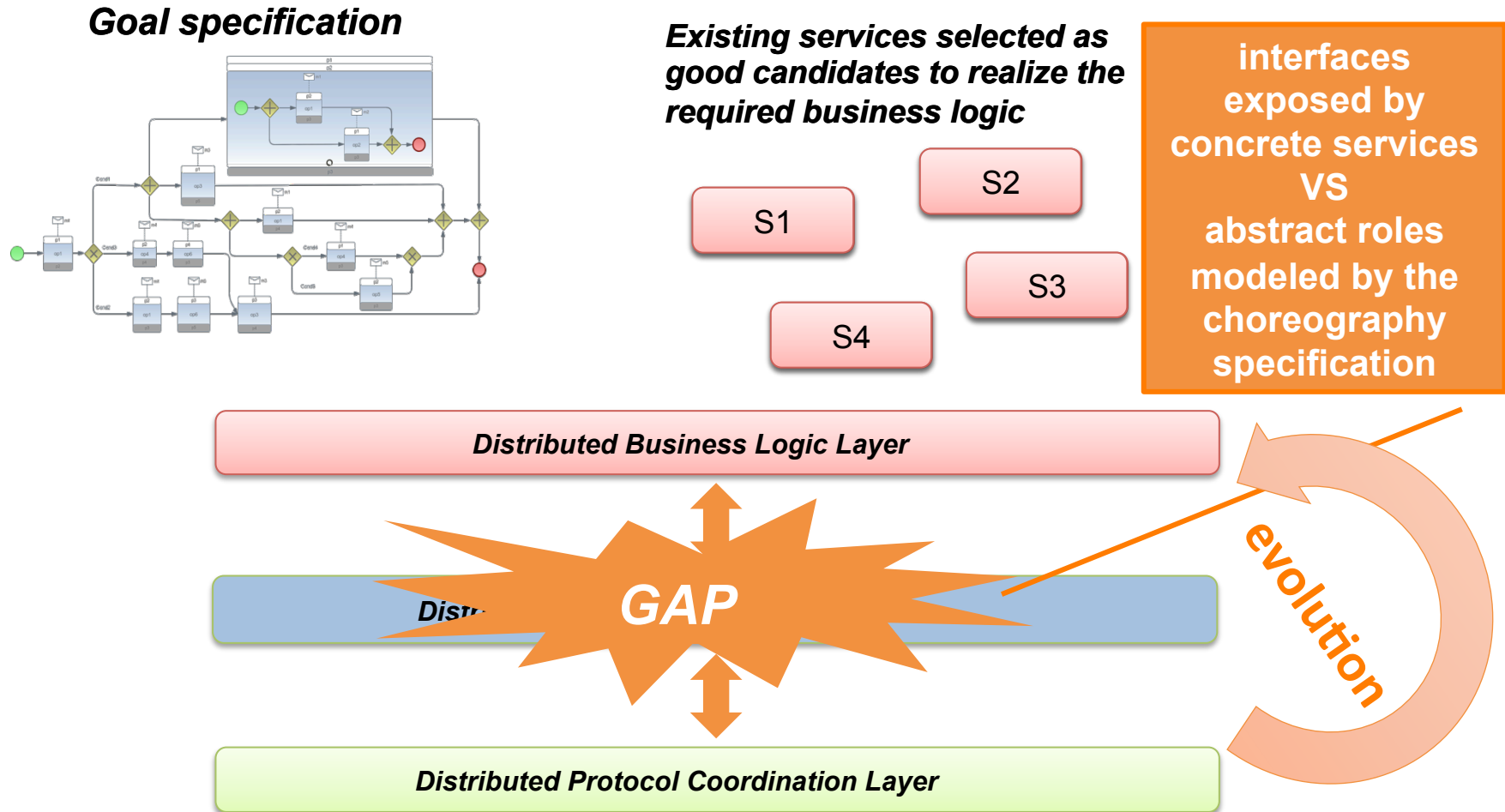
Synthesis phase



... the **Synthesis Processor** automatically produces (if possible) a choreography-based application achieving the specified goal



Main ingredients of our method



Choreography coordination

Problem

Automatic enforcement of choreography realizability

- How to externally coordinate the interaction of existing services so to fulfill the global collaboration prescribed by the choreography specification?

Assumption

BPMN2 Choreography Diagrams

- A choreography-based specification of the system to be realized

Our solution

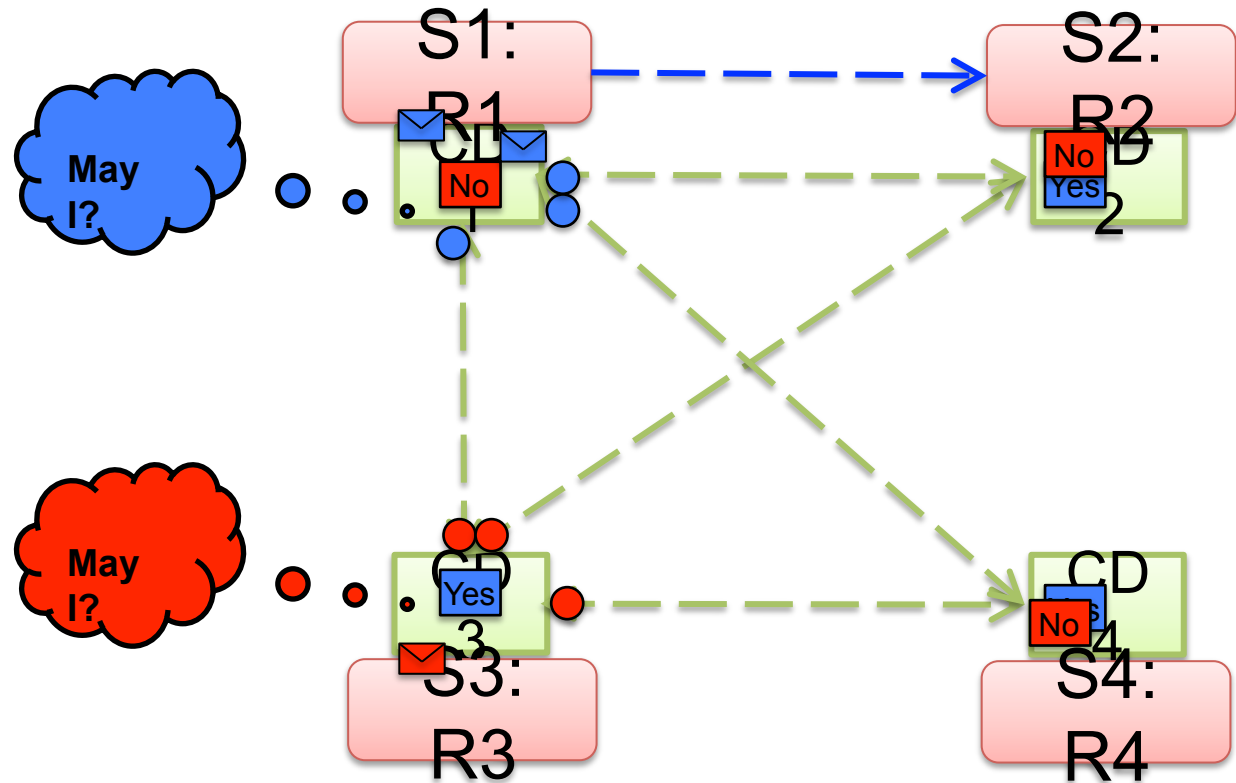
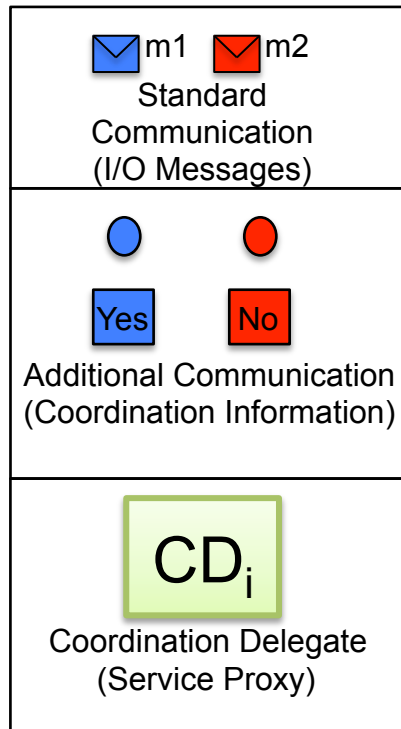
Automated synthesis of Coordination Delegates (CDs)

- Automatically produce the code of additional software entities that proxy and coordinate the services' interaction so to guarantee the specified global collaboration

Focus

Automatically realizing a choreography by reusing and suitably coordinating third-party services

Enforcing choreography realizability



Choreography adaptation

Problem

Enforcement of service-role bindings

- How to externally adapt the interaction of existing services so to “match” the specification of the choreography roles to be played?

Assumption

LTS-based or BPEL+WSDL-based specification

- A specification of the externally observable behavior of both services and roles in terms of types and sequences of message exchanges

Our solution

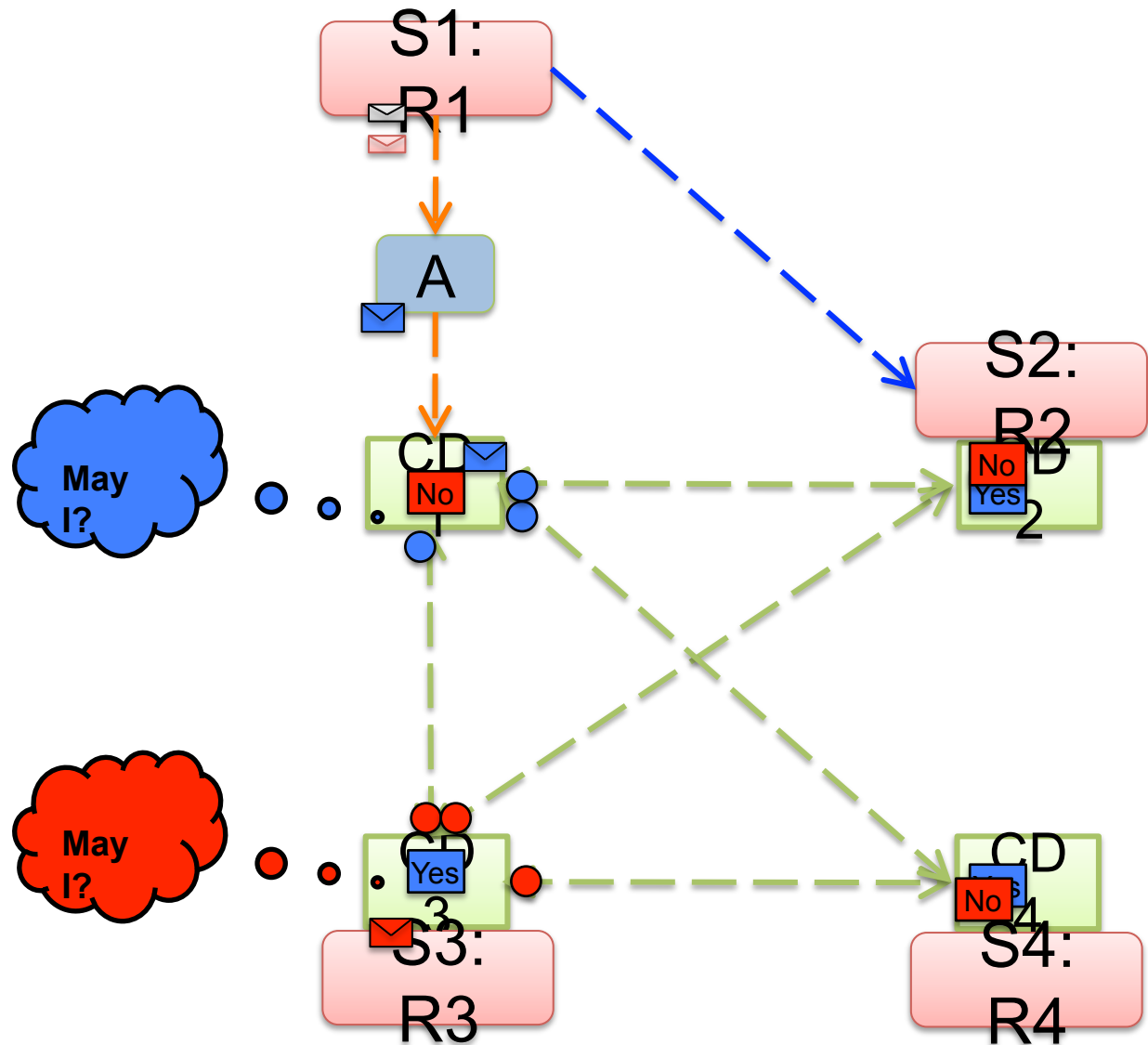
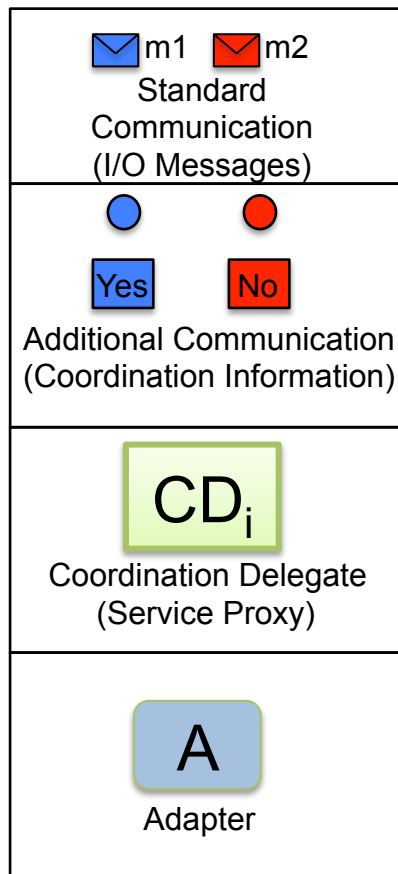
(Partially) automated synthesis of Adapters

- Produce adapters that mediate the interaction $\text{Service} \leftrightarrow \text{CD}$

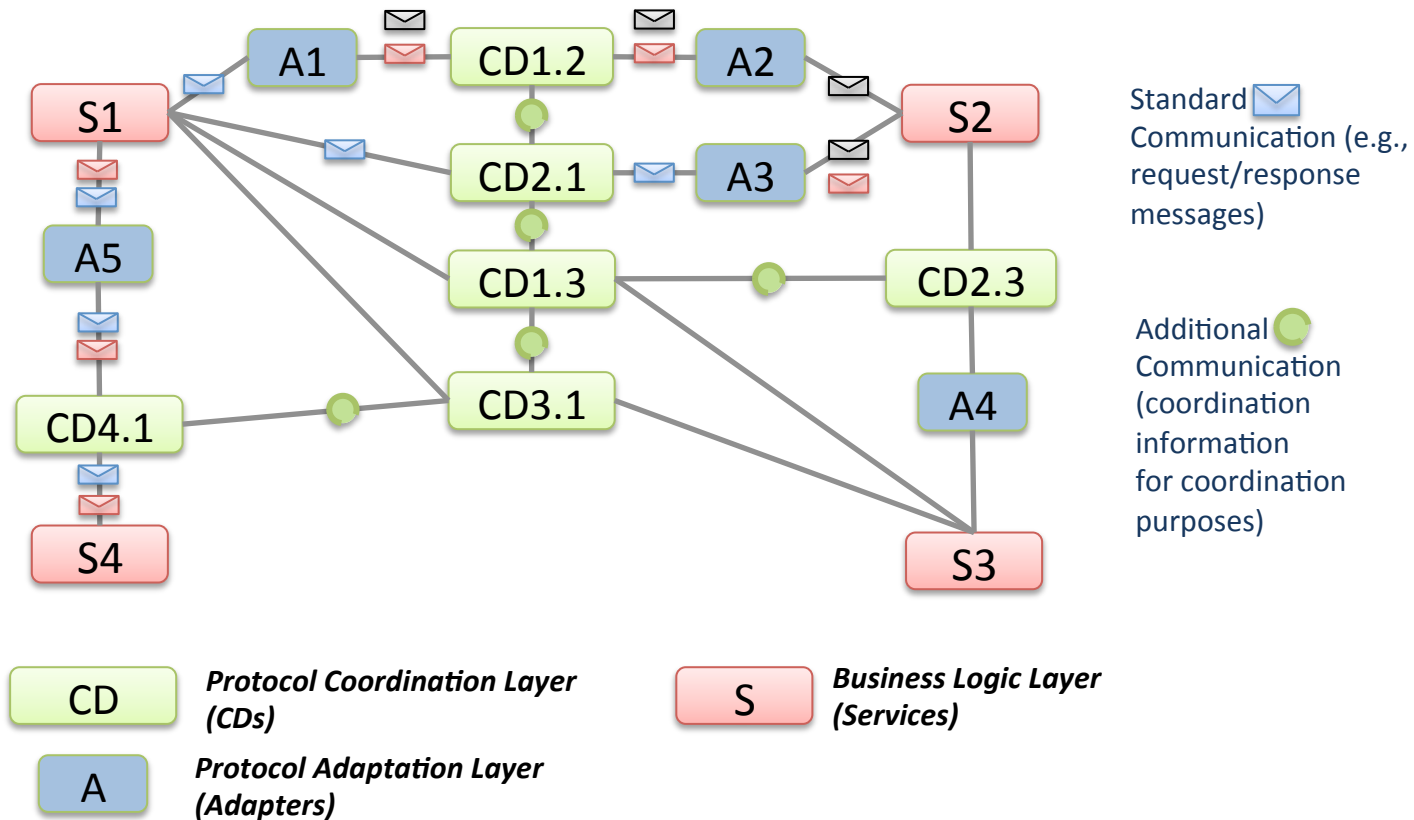
Focus

Realizing correct service-role binding by solving interoperability issues

Enforcing service-role bindings



Adopted architectural style



Focusing on adapters generation

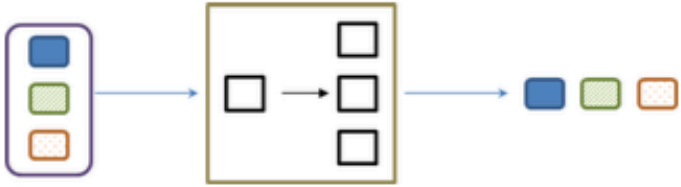
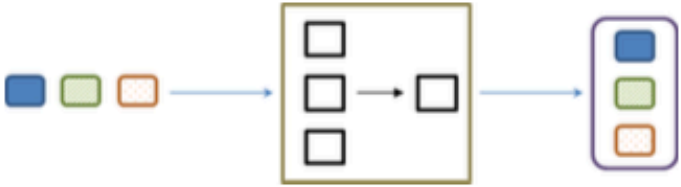
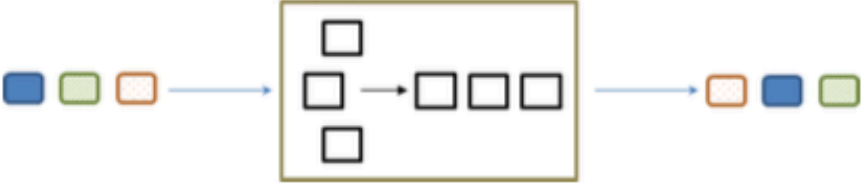
- We exploit **Enterprise Integration Patterns** (EIP)

The generated adaptation logic is realized as a **composition of Message Routing patterns** that realize I/O data mappings

e.g., adapters are able to

- map message data types
- reorder/merge/split the sequence of operation calls and/or related I/O messages

Considered EIP: Message Routing patterns

Name	Description	Figure
Splitter	It splits a message into several ones, and sends the resulting messages to be processed independently	
Aggregator	It receives multiple messages and combines them into a single message. It is stateful and it must buffer the messages to be aggregated and determine when they are completed	
Resequencer	It collects and re-orders messages and put them into an output channel in the specified order. Similarly to the Aggregator, it is stateful	
Message Filter	It decides whether a message should be passed along or dropped based on some criteria respect to the header and/or content of the message	

Choreography evolution

Problem

Choreography evolution

- How to enable choreography evolution in response to goal and context changes?

Assumption

Specification of variation points in terms of Call Choreographies

- Each variation point specifies behavioral alternatives in term of (set of) sub-choreographies

Our solution

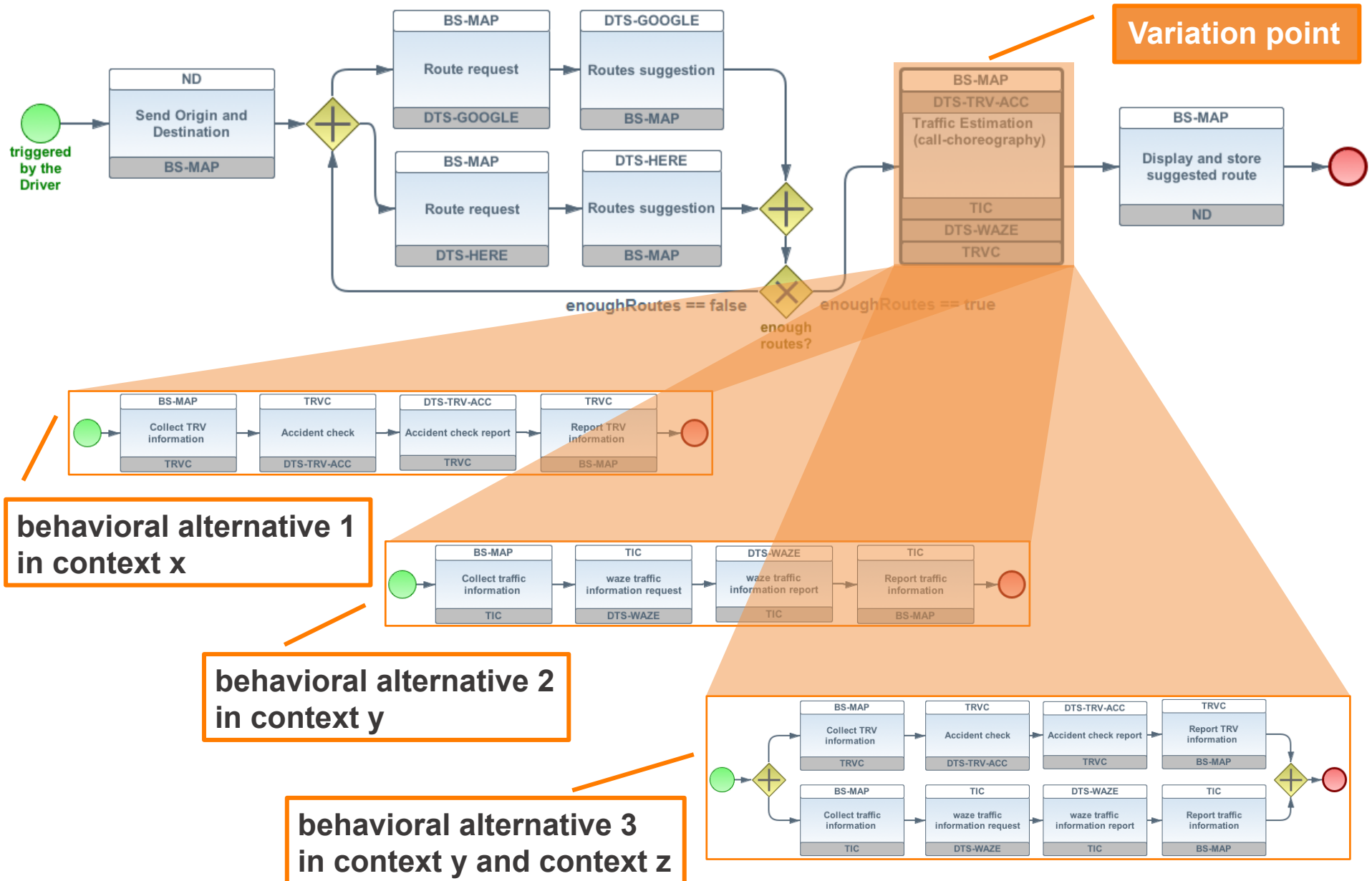
Automated synthesis of autonomic CDs (aCDs)

- Automatically produce the code of CDs that are now are external controllers realizing **multiple interacting feedback loops**

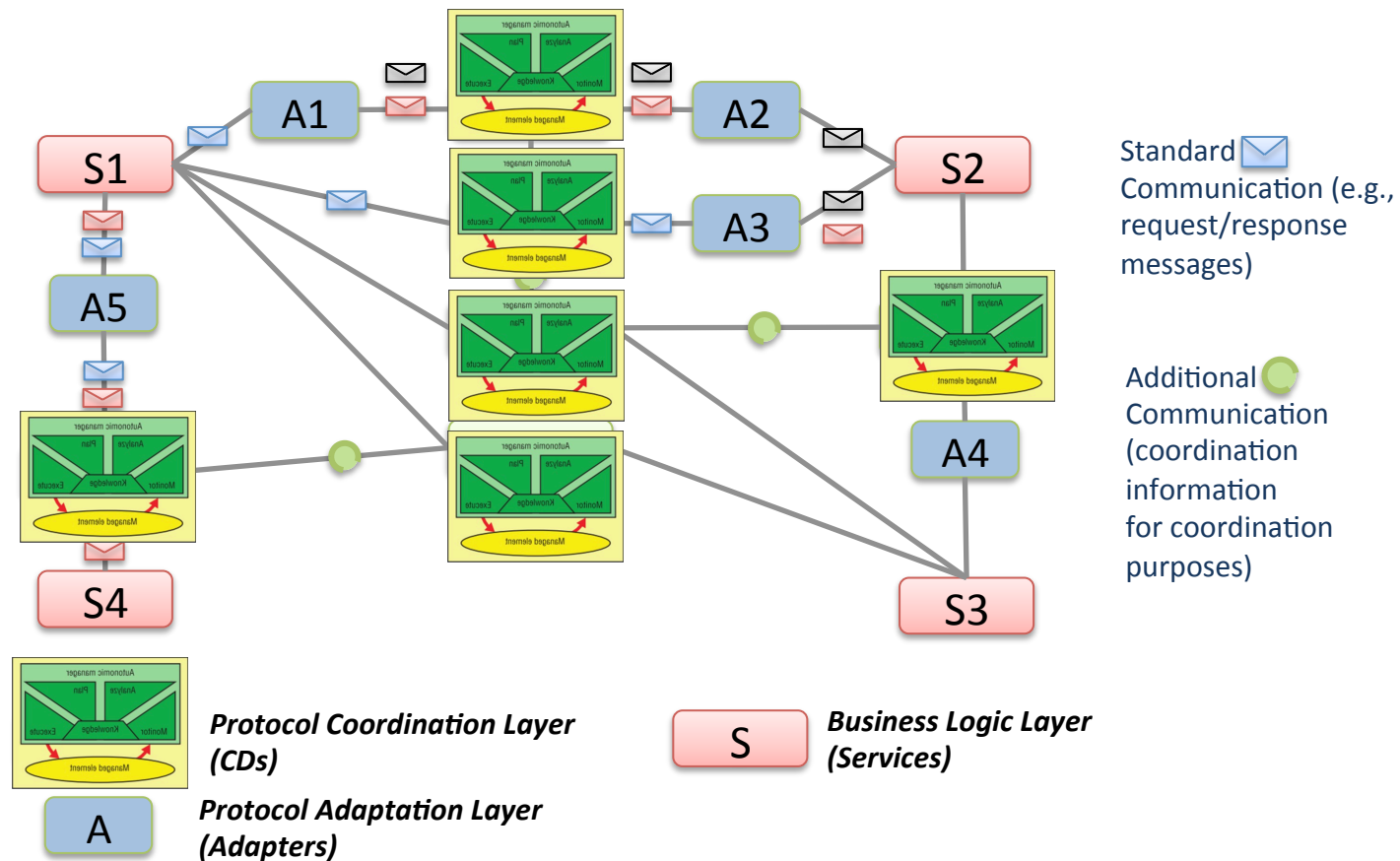
Focus

Enabling (a form of) choreography evolution to face well-confined goals and context changes

Choreography evolution



Adopted architectural style



Conclusions

Put the **bases to support dynamic choreography evolution** in response of goal and context changes

- **Separation of concerns** between application, coordination, and adaptation logic
- **Adapters as a composition of different EIP** depending on a notion of I/O data mappings inference
(Message Filter, Aggregator, Splitter, and Resequencer)

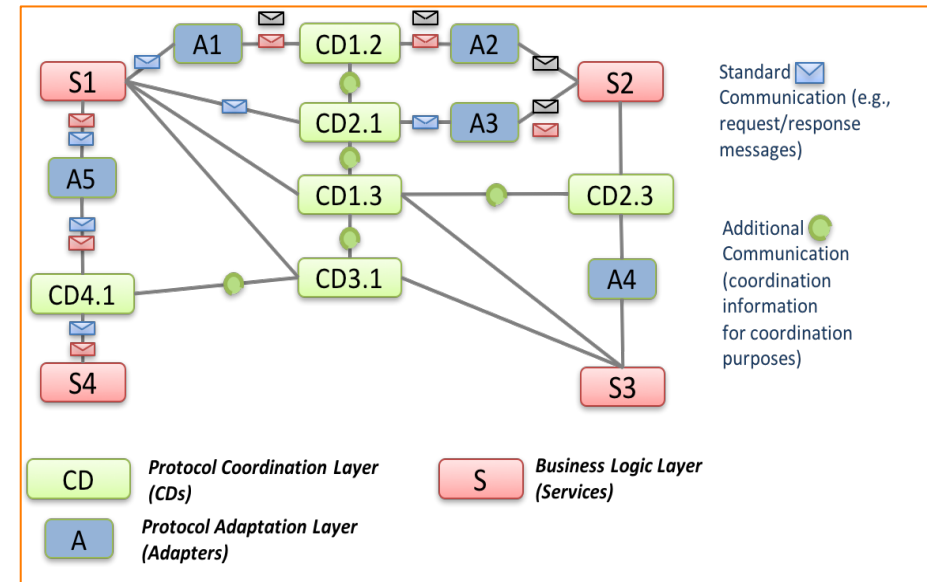
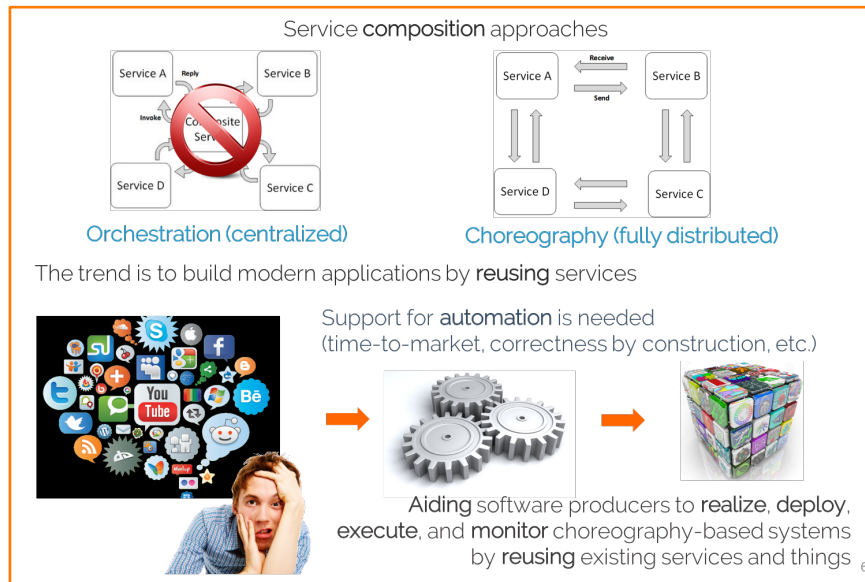
Relevance of exploiting EIP

- Modular adapters
- Dynamic evolution
- Automatic generation and easier maintenance of adapters' code

Future research directions

→ GAUSS needs

- ✓ Extension of the approach to deal with **governance issues**
 - Multiple systems belonging to different security domains/federations governed by different authorities
 - Usage of different identity attributes that are utilized in their access control policies
- ✓ How to support **dynamic evolution** via the automated and **on-the-fly synthesis of, e.g., more complex adapters** realized by combining additional classes of EIP, e.g.,
 - Message Transformation Patterns such as Content Enricher, Content Filter, and Transformer
 - Semantic interoperability
 - Enabling a finer form of adaptation concerning mismatches at the level of the semantics of the exchanged messages



THANK YOU

