Transferring Transactional Business Processes To FaaS

Eighth International Workshop on Serverless Computing 2022 (WoSC8)

Kostas Meladakis, Chrysostomos Zeginis, Kostas Magoutis and Dimitris Plexousakis
Foundation for Research & Technology – Hellas (FORTH), Heraklion, Greece

Supported by

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

INSTITUTE OF COMPUTER SCIENCE

A few thing about BPMN2.0...

Community

Managers

Business Analysts

Technical Developers

Application Area

Management

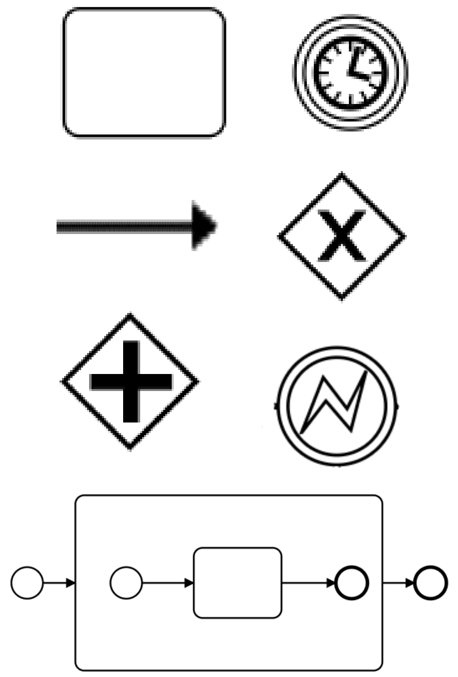
Purchases

Orders

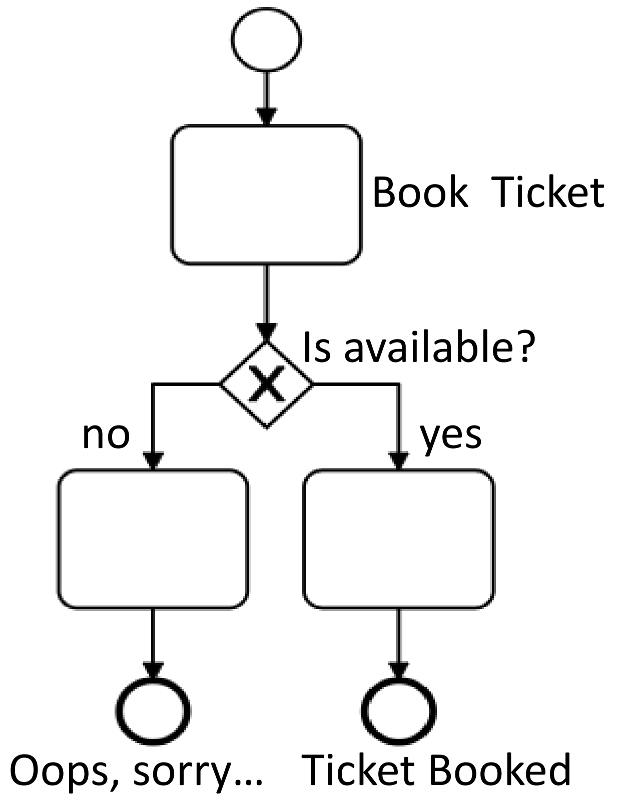
INSTITUTE OF COMPUTER SCIENCE

What is BPMN2.0 ?

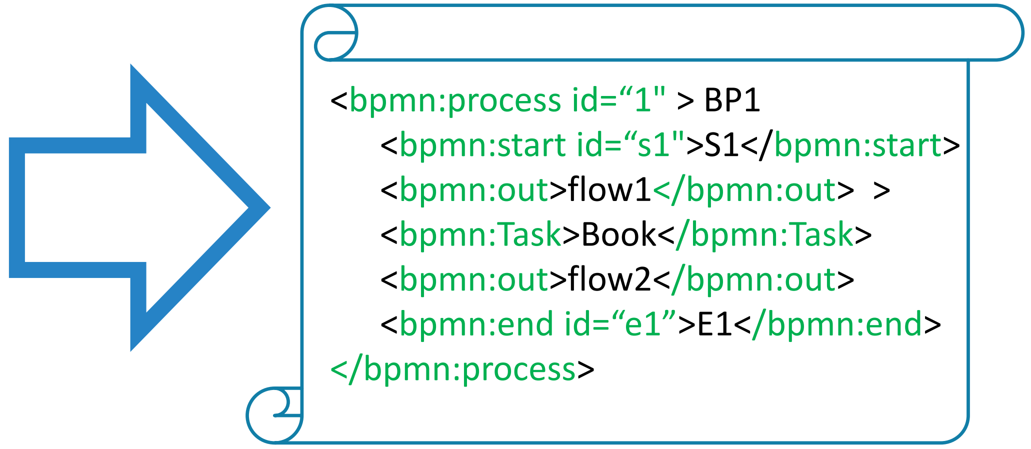
BPMN symbols



Workflows



XML standard



```
<bpmn:process id="1" > BP1
    <bpmn:start id="s1">S1</bpmn:start>
    <bpmn:out>flow1</bpmn:out>  >
    <bpmn:Task>Book</bpmn:Task>
    <bpmn:out>flow2</bpmn:out>
    <bpmn:end id="e1">E1</bpmn:end>
</bpmn:process>
```

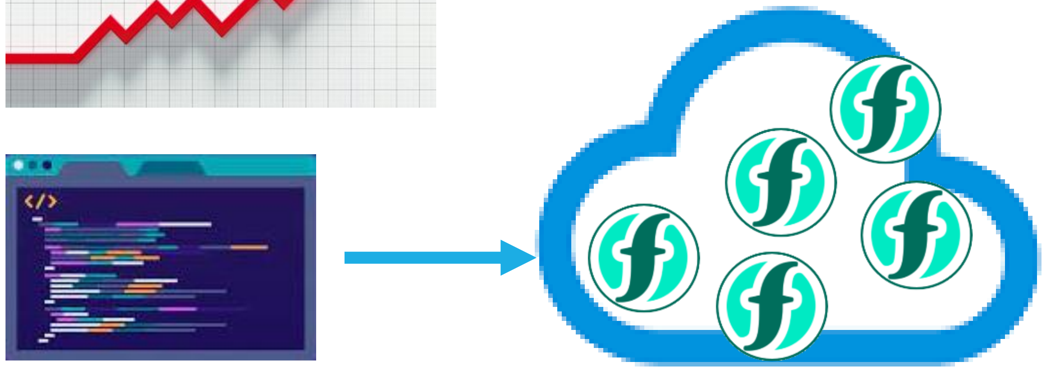
ISO/IEC 19510:2013

BPMN2.0 engines

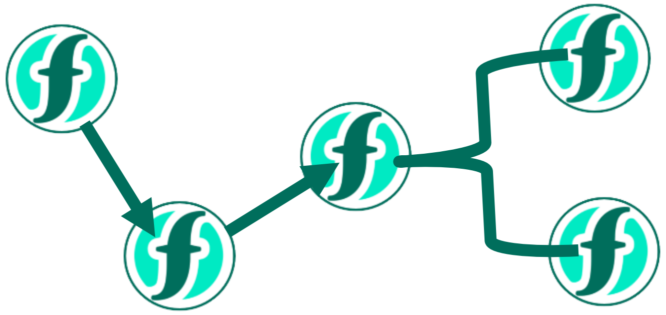
can build whole applications

FORTH
INSTITUTE OF COMPUTER SCIENCE

FaaS is a trend!



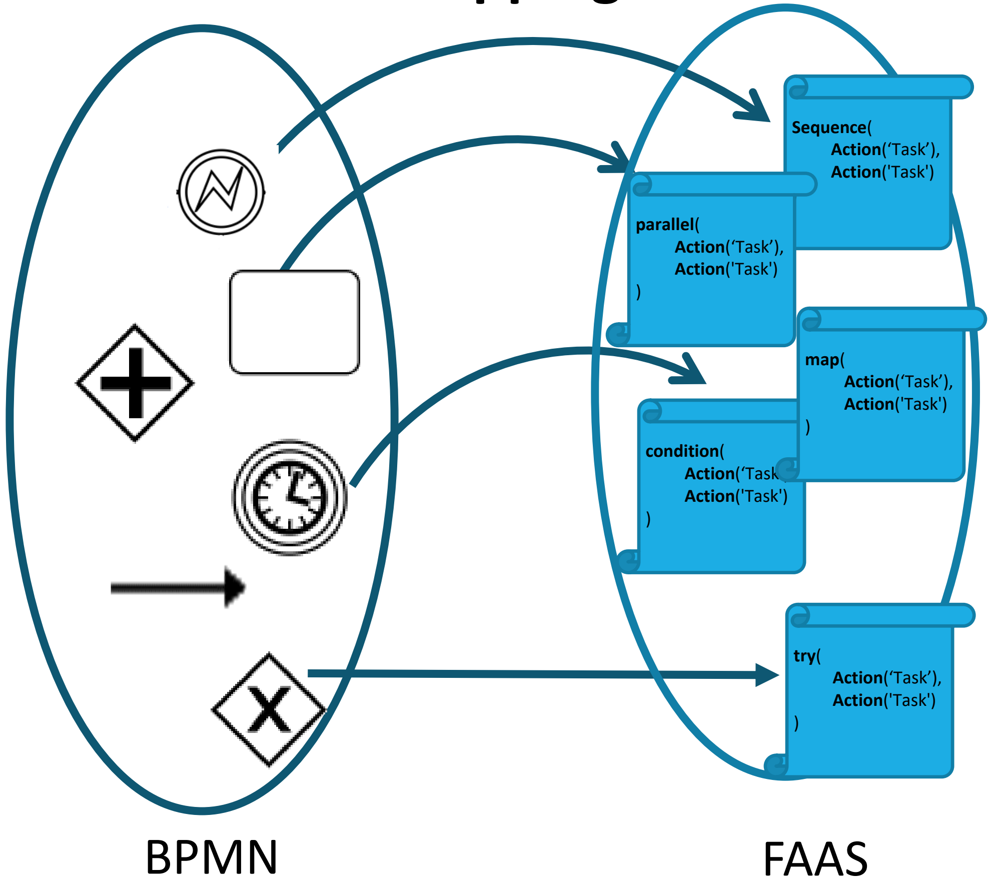
Orchestration → Define Relationships



```
Sequence(
        Action('Task'),
        Action('Task')
)
```

Bridge BPMN2.0 & FaaS

Mapping

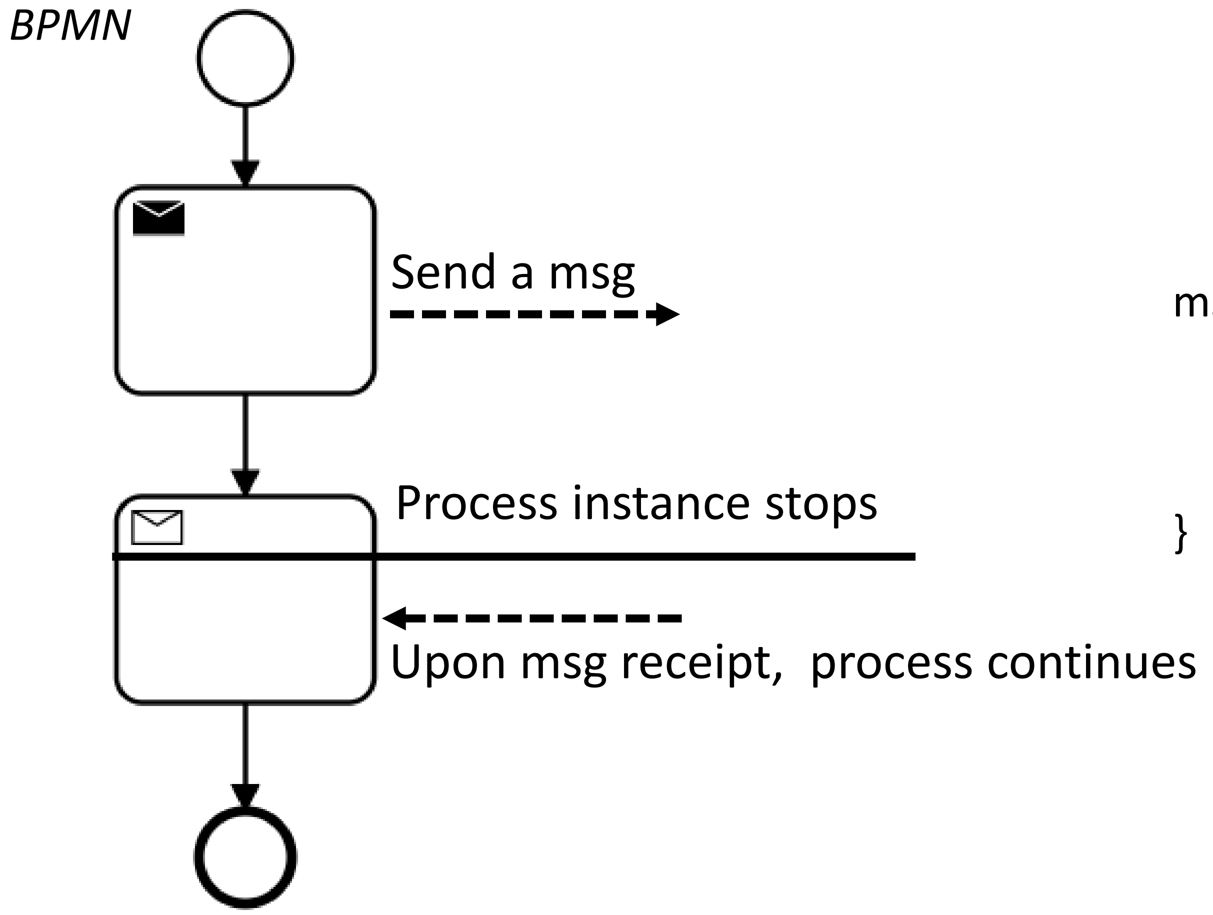

Sequence(Action('Task'), Action('Task')
parallel(Action('Task'), Action('Task'))
map(Action('Task'), Action('Task'))
condition(Action('Task Action('Task'))
try(Action('Task'), Action('Task'))
BPMN
FAAS

One-to-one mapping of common constructs

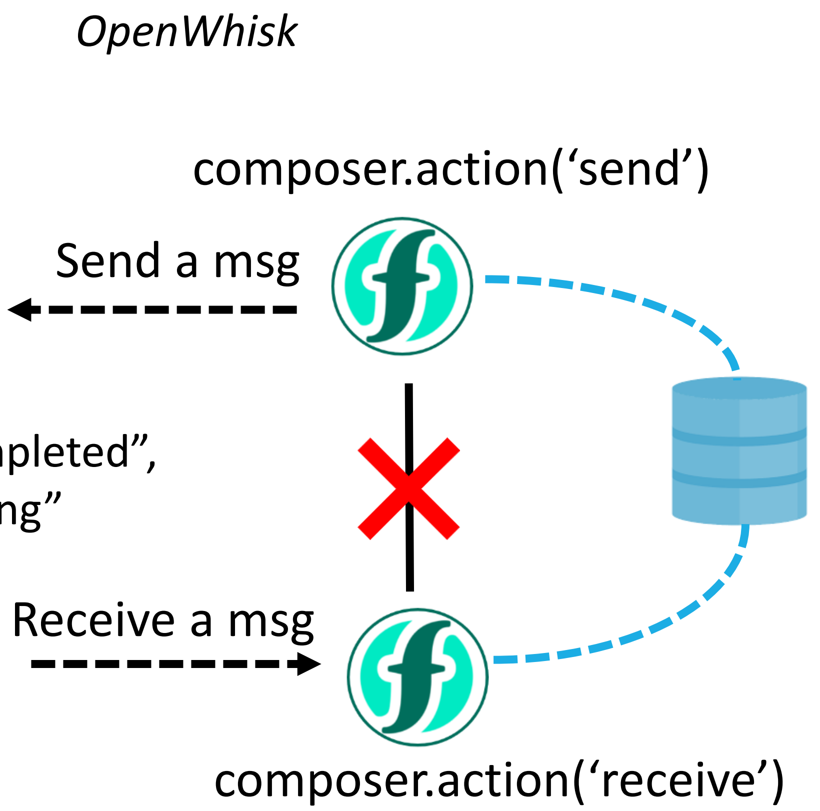
BPMN notation	BPMN representation	OpenWhisk concept	Visual representation
Task	name	composer.action('name')	ƒ
Sequence	→	composer.sequence(action1, action2)	→
Exclusive gateway	⊗	composer.if(condition, path1, path2)	⊗
Parallel gateway	⊕	composer.parallel(action1, action2)	⊕

● ● ●

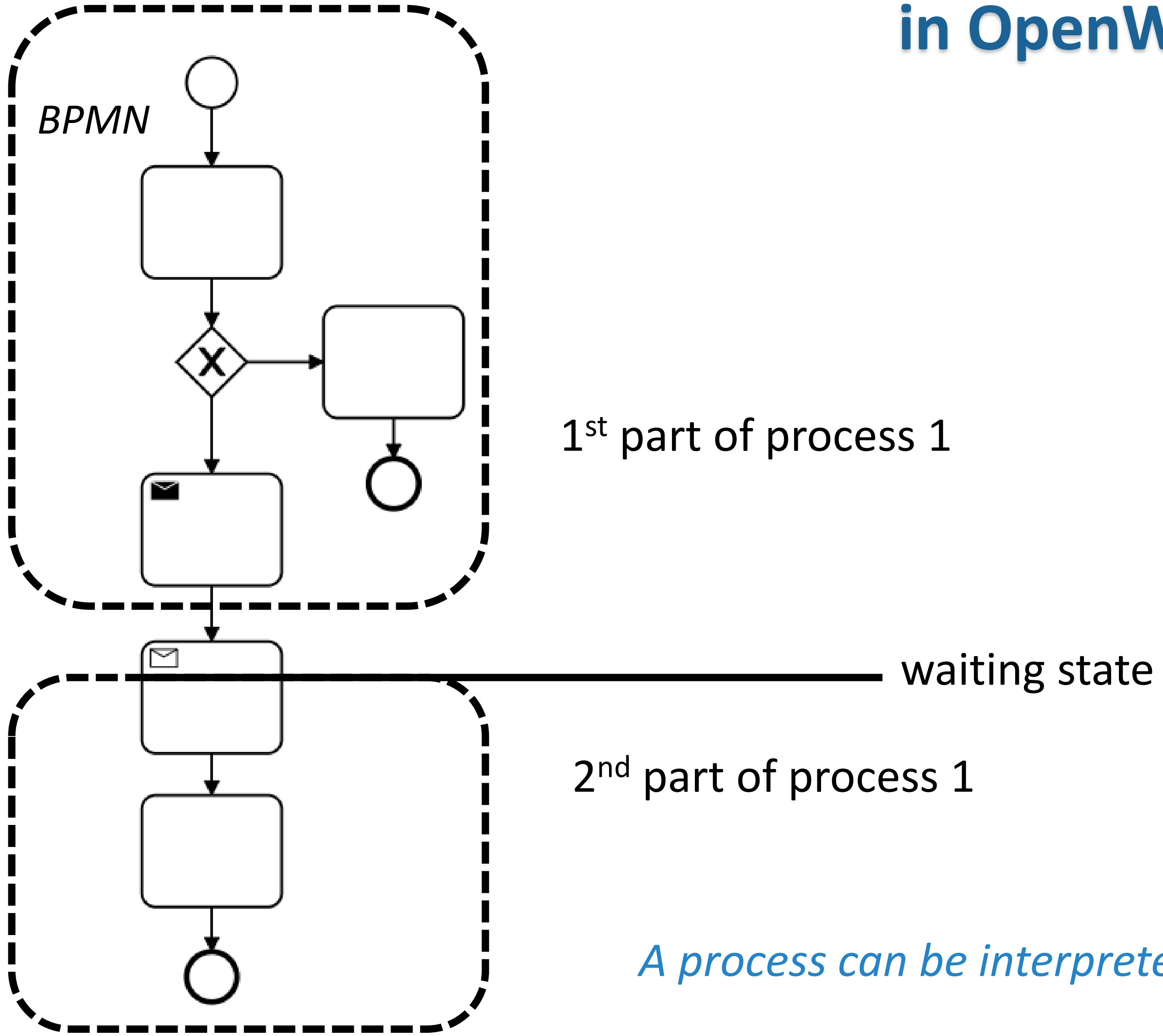
Challenge 1: BPMN blocking vs FaaS event-driven

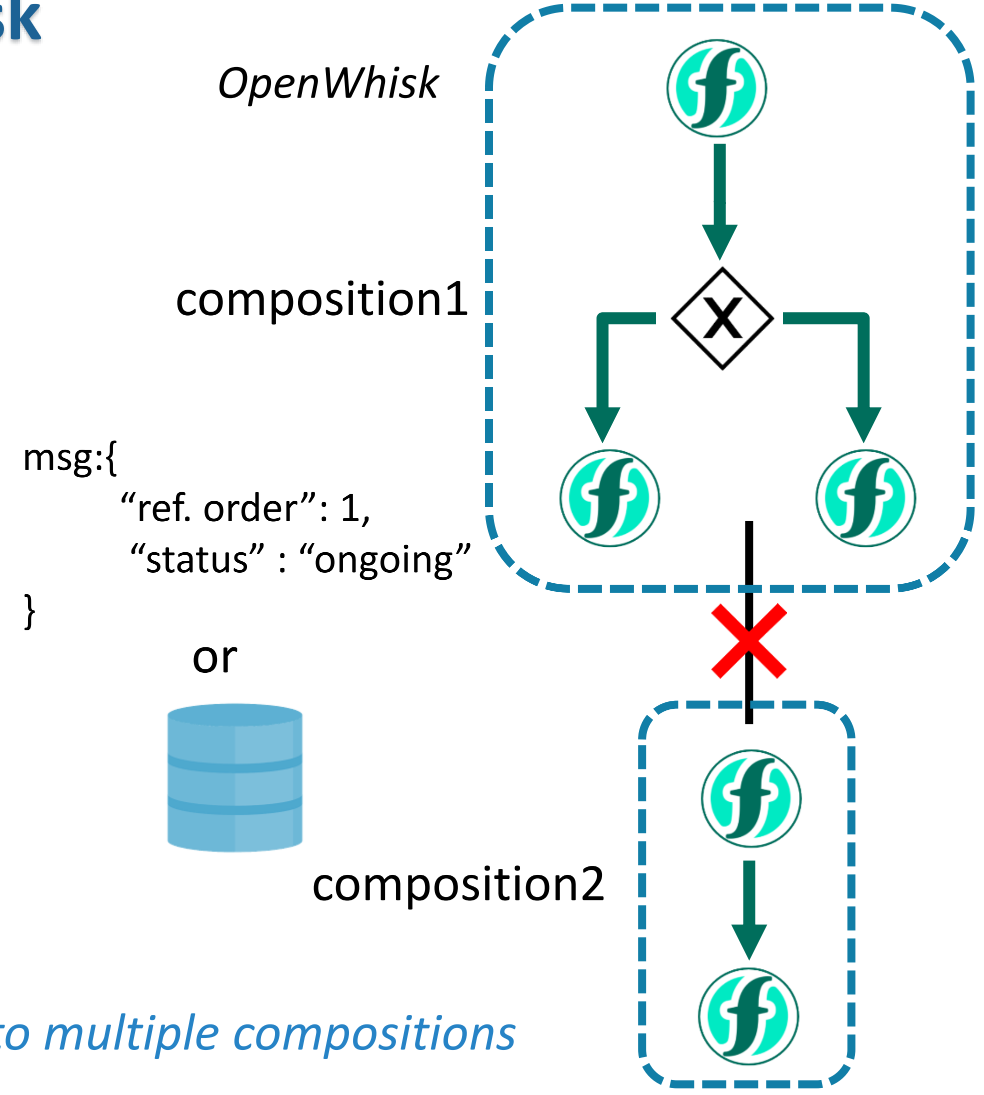


msg:{
"ref. order": 1,
"sendTask": "completed",
"status" : "ongoing"
}



Implementing BPMN waiting states in OpenWhisk



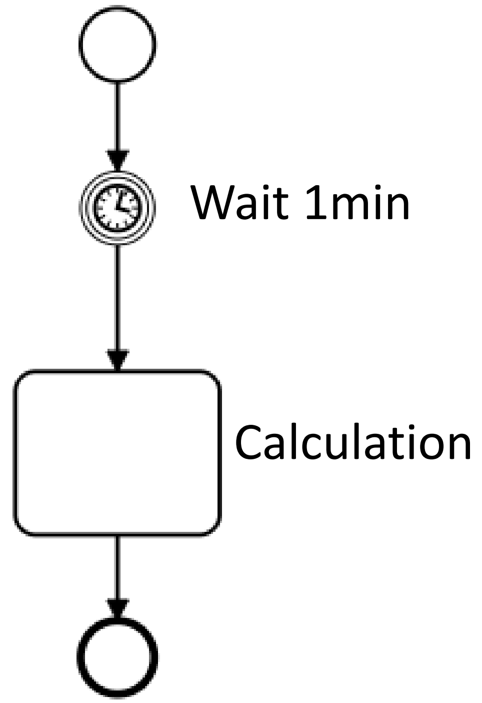


```
msg:{
        "ref. order": 1,
         "status" : "ongoing"
}
```

A process can be interpreted into multiple compositions

Challenge 2 : Implementing BPMN delays in FaaS

BPMN



orchestration definitions

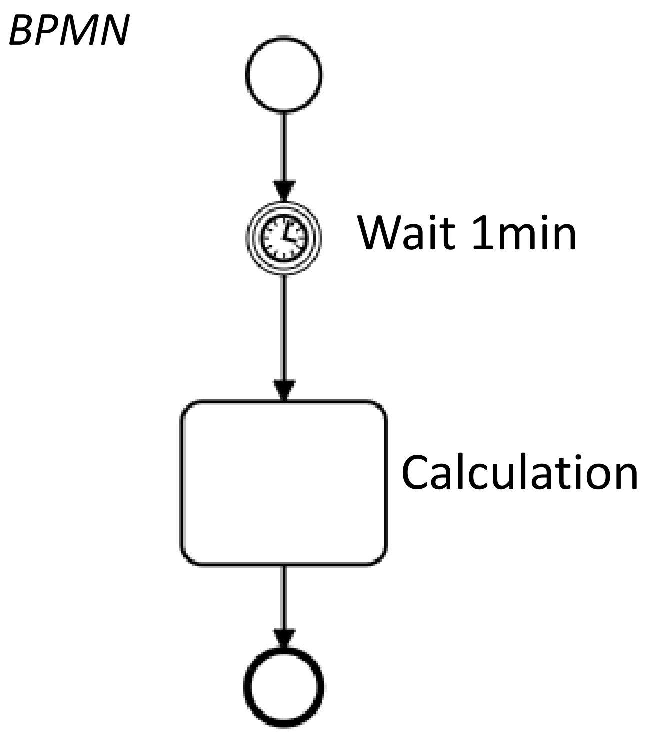
```
Sequence(
          Function( (input) =>{const date = Date.now();
                    let now = null;
                    do { now = Date.now(); }
                    while (now - date < 1000);
                    return input; } ),
          Action('Task')
)
```

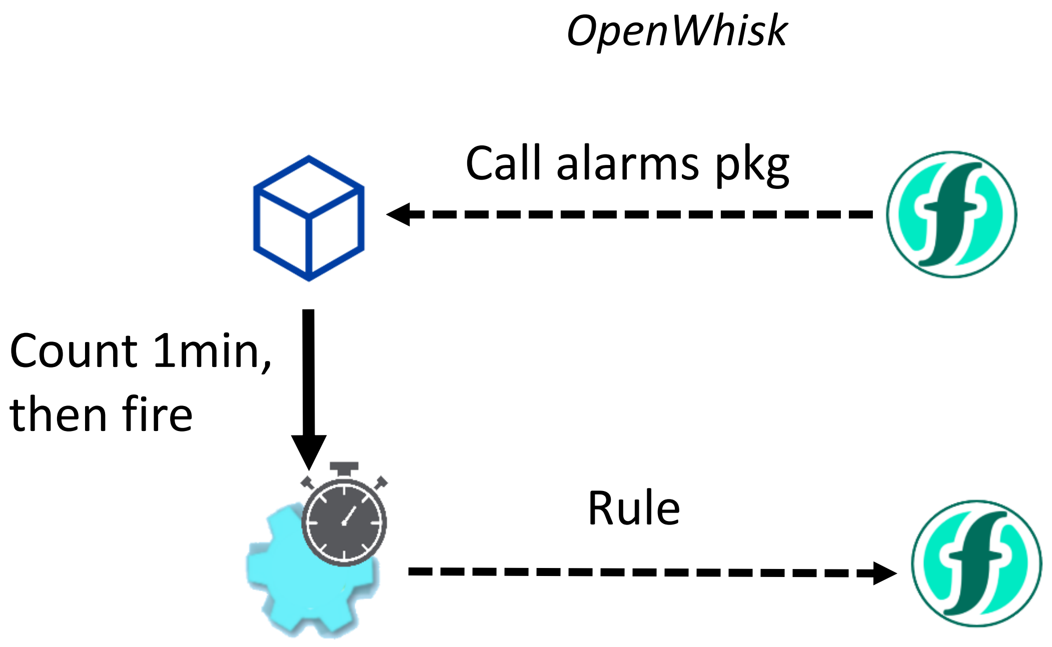
Timeout limit of 6min

Burns our CPU

INSTITUTE OF COMPUTER SCIENCE

Implementing Delay: Use a timed event cloud-native service

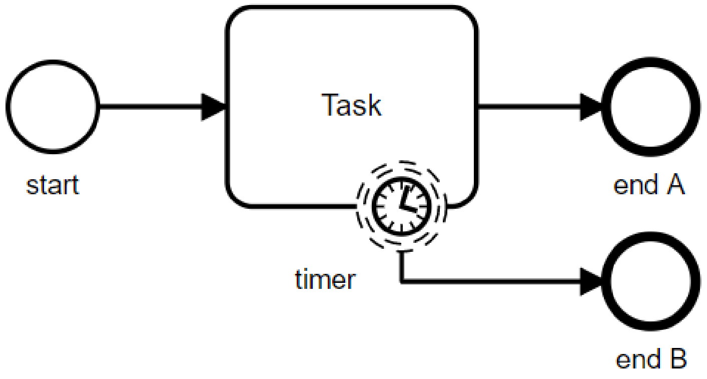




FORTH INSTITUTE OF COMPUTER SCIENCE

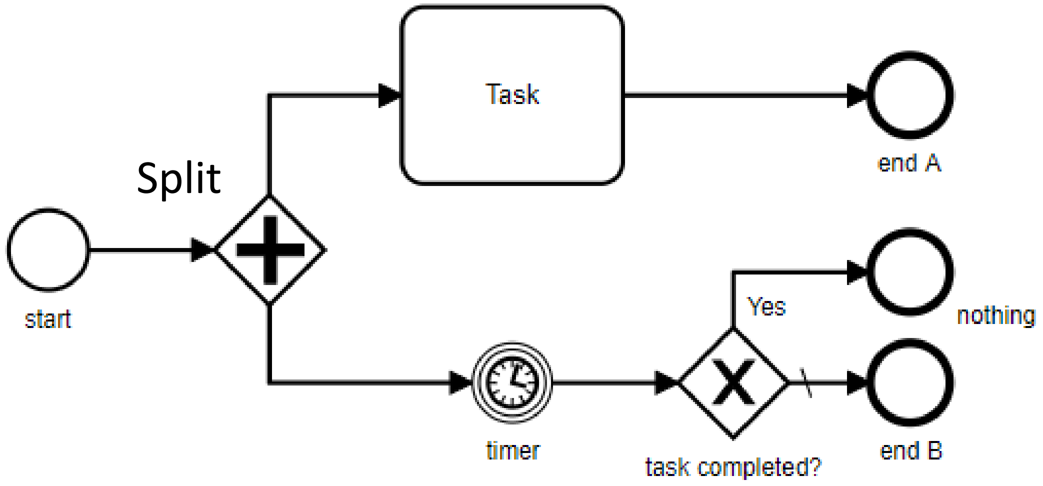
Challenge 3: Boundary Timers

Non-interrupting boundary timer



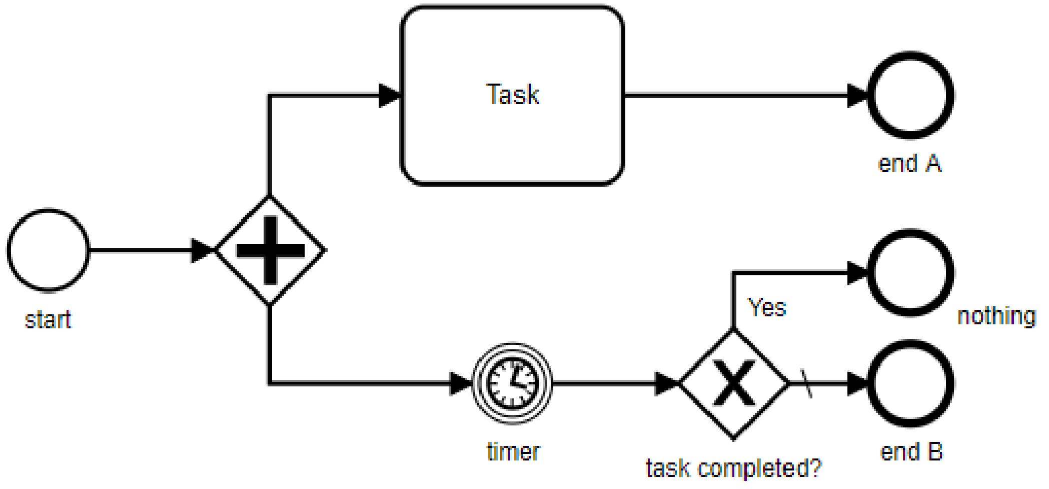
BPMN

Think algorithmically
(reduction to a problem with known solution)

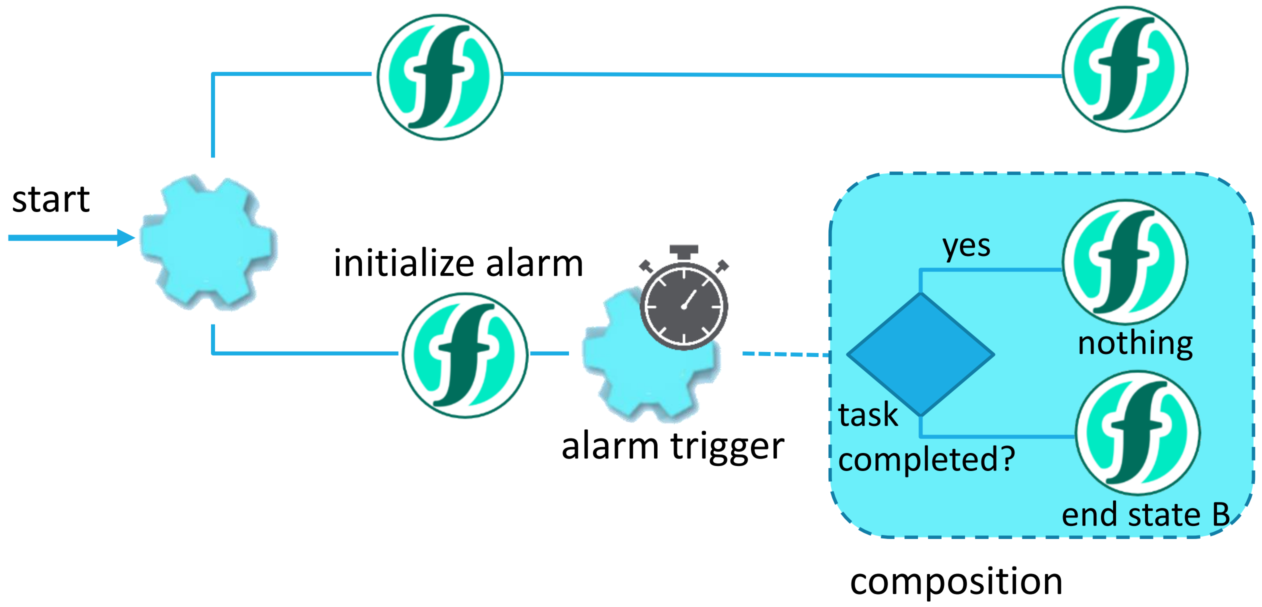


BPMN

Transformation of boundary timer to OpenWhisk

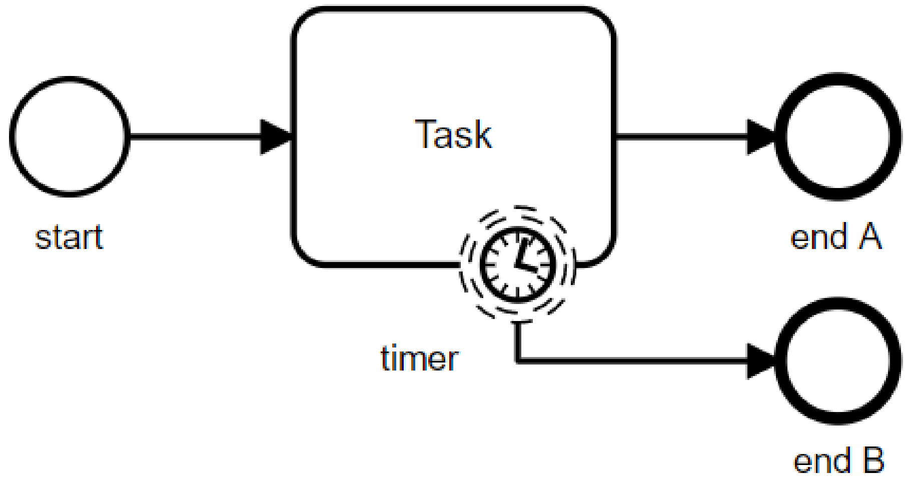


BPMN

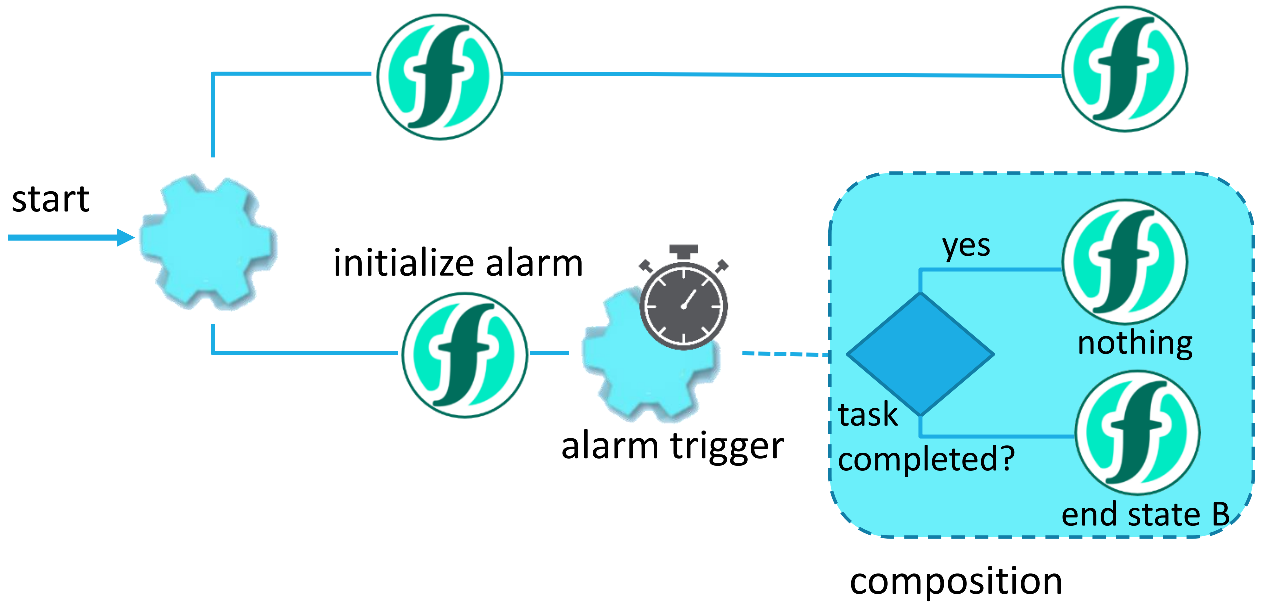


OpenWhisk

Transformation of boundary timer to OpenWhisk

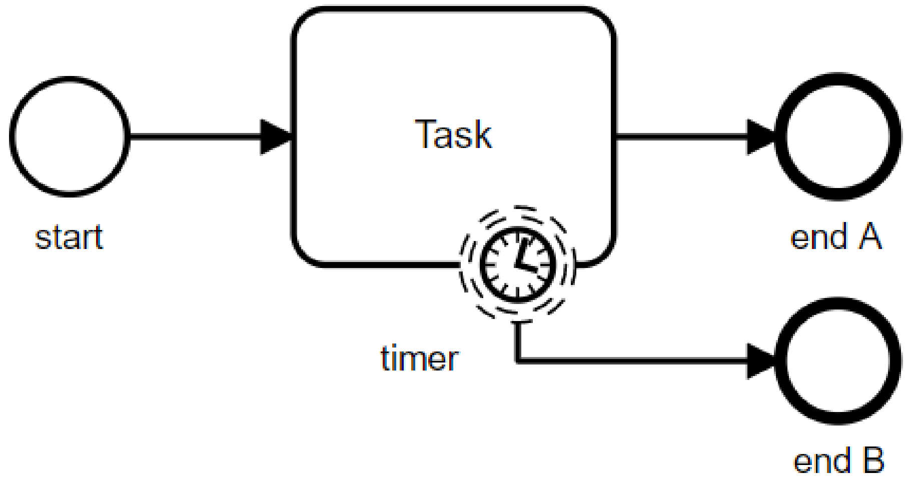


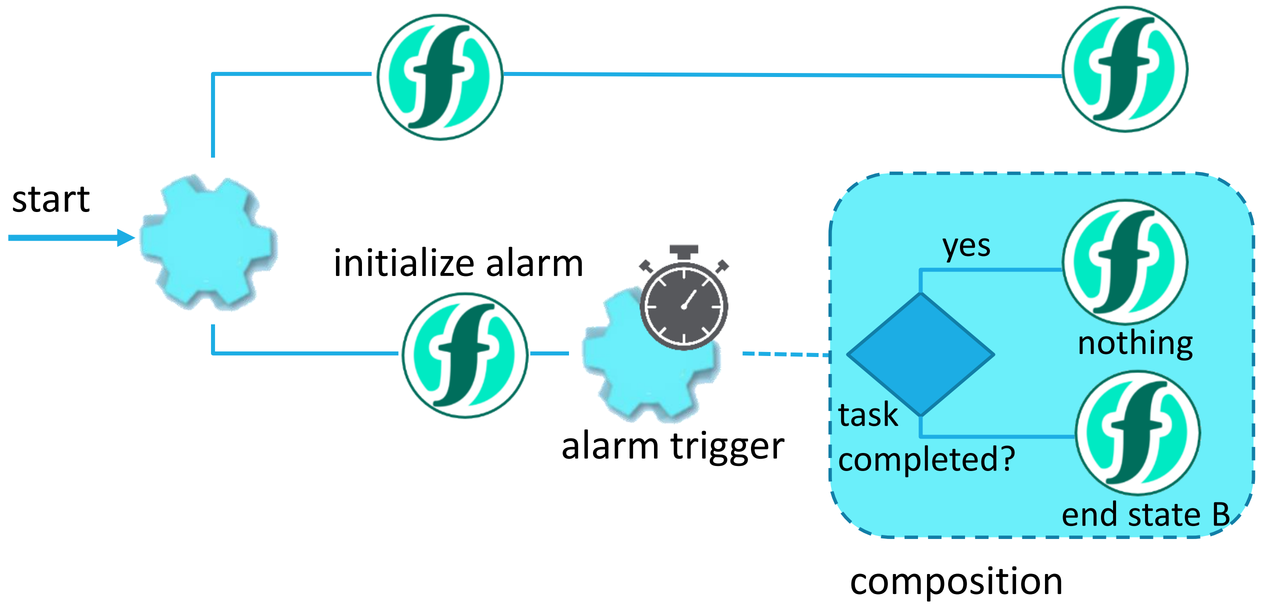
BPMN



OpenWhisk

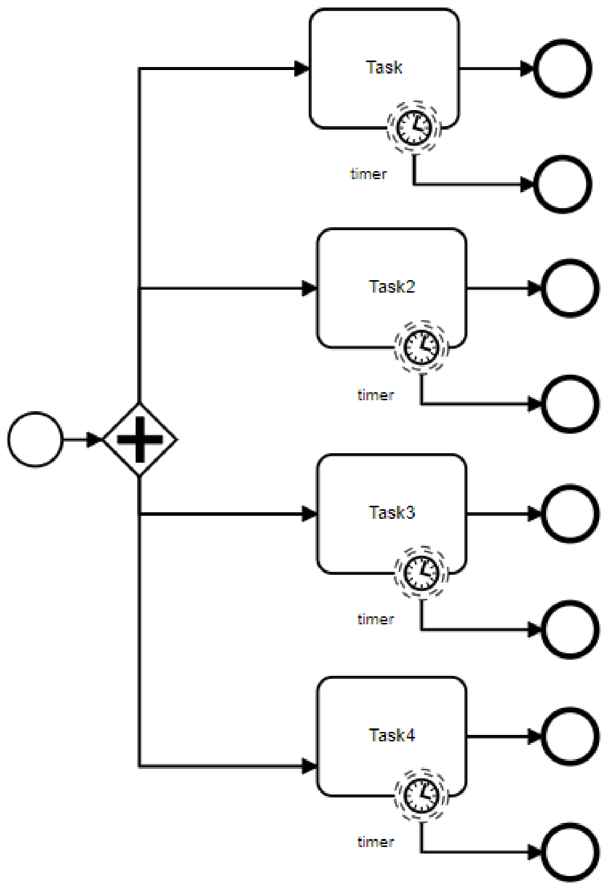
Observation: FaaS is structurally more complex

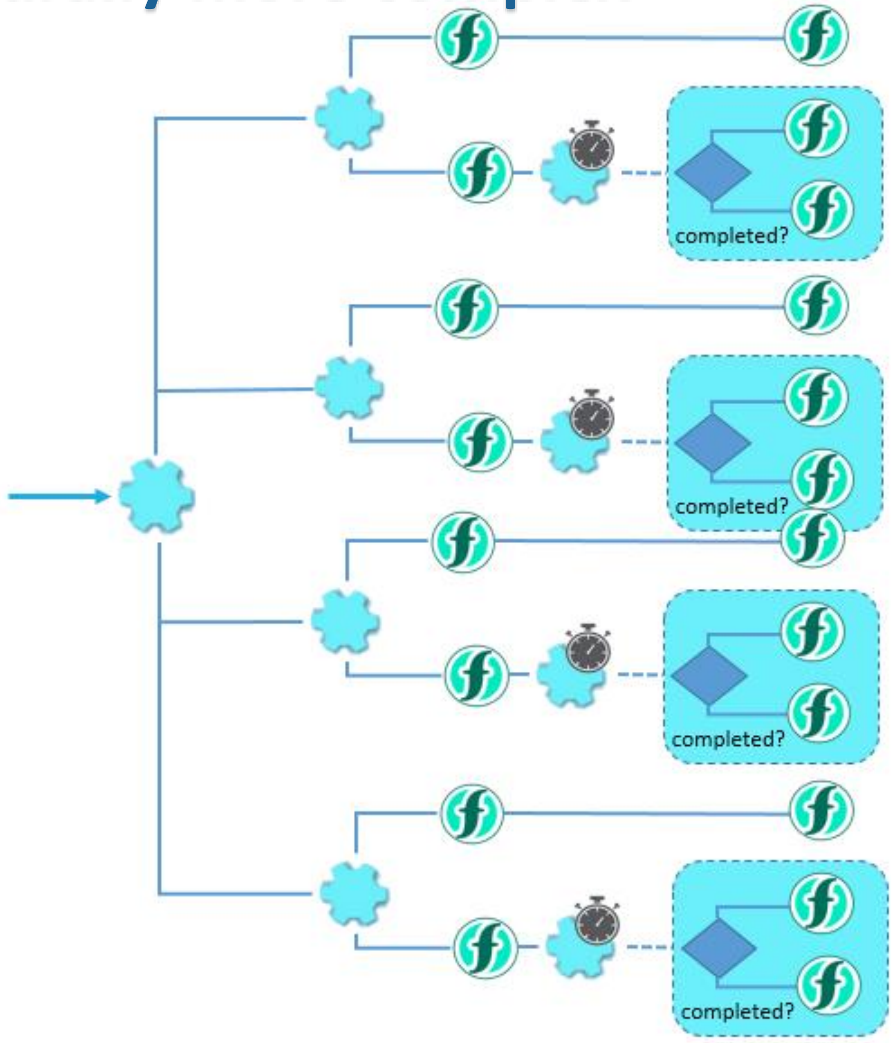




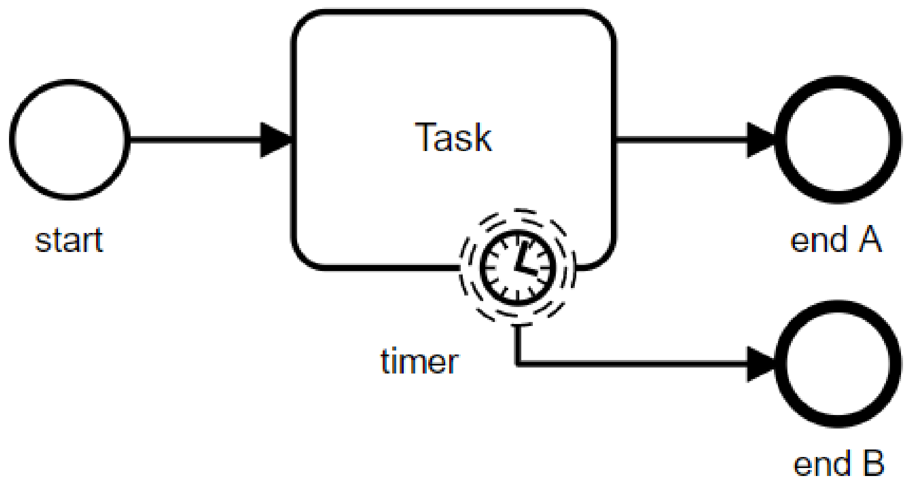
INSTITUTE OF COMPUTER SCIENCE

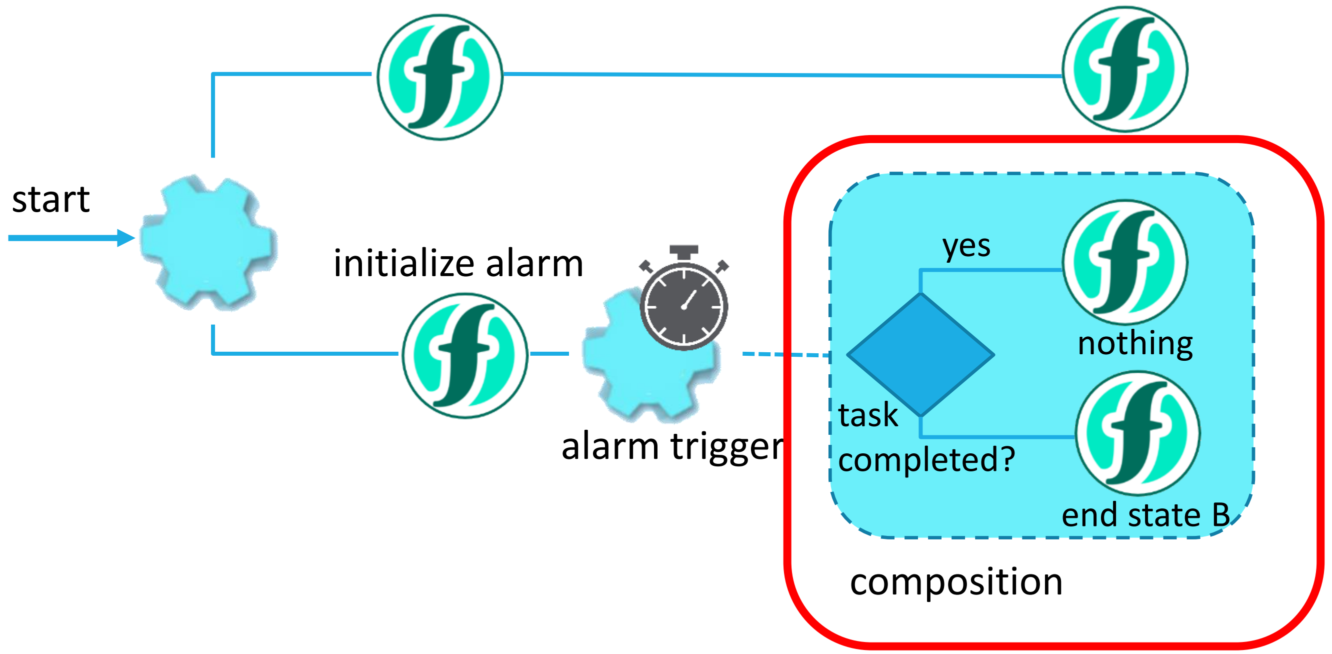
Observation: FaaS is structurally more complex



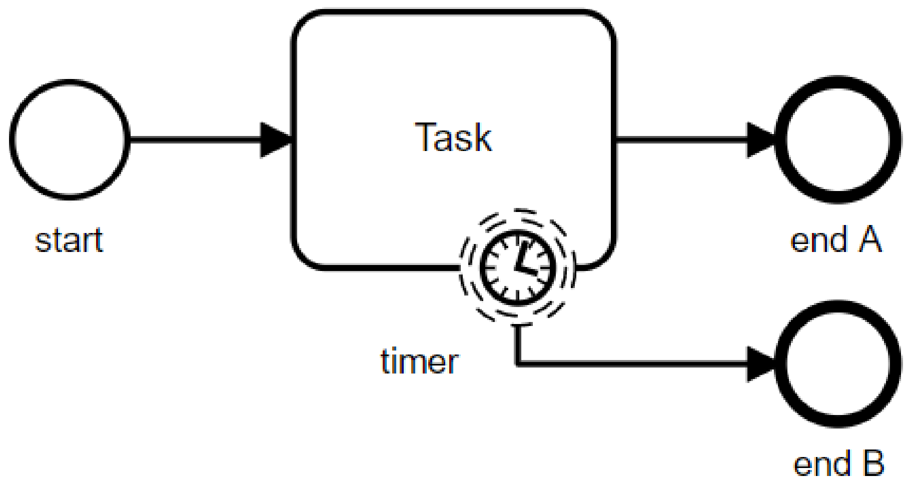


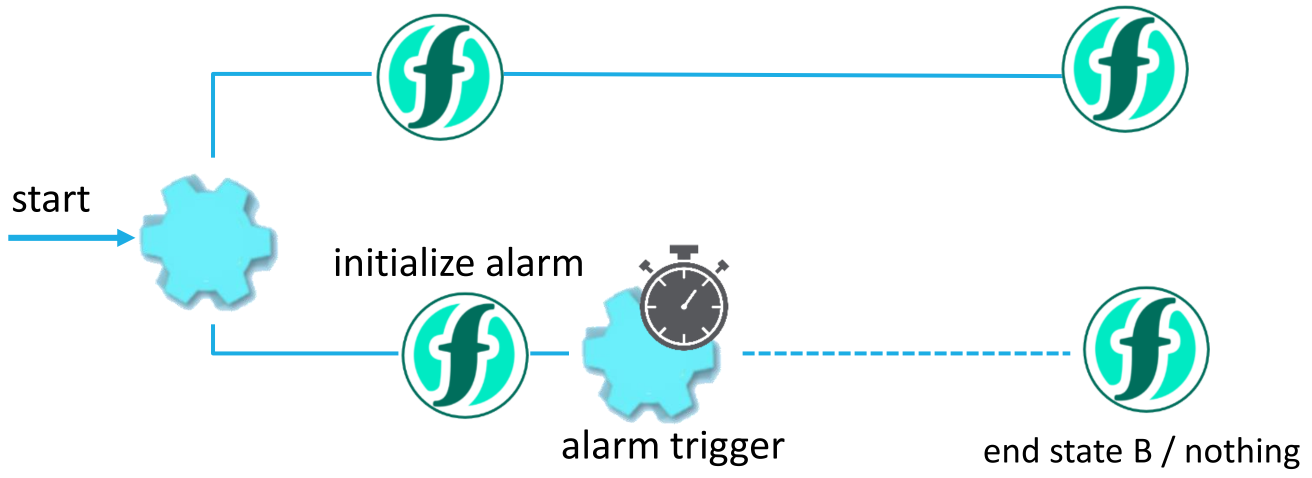
Observation: FaaS is structurally more complex





Observation: FaaS is structurally more complex





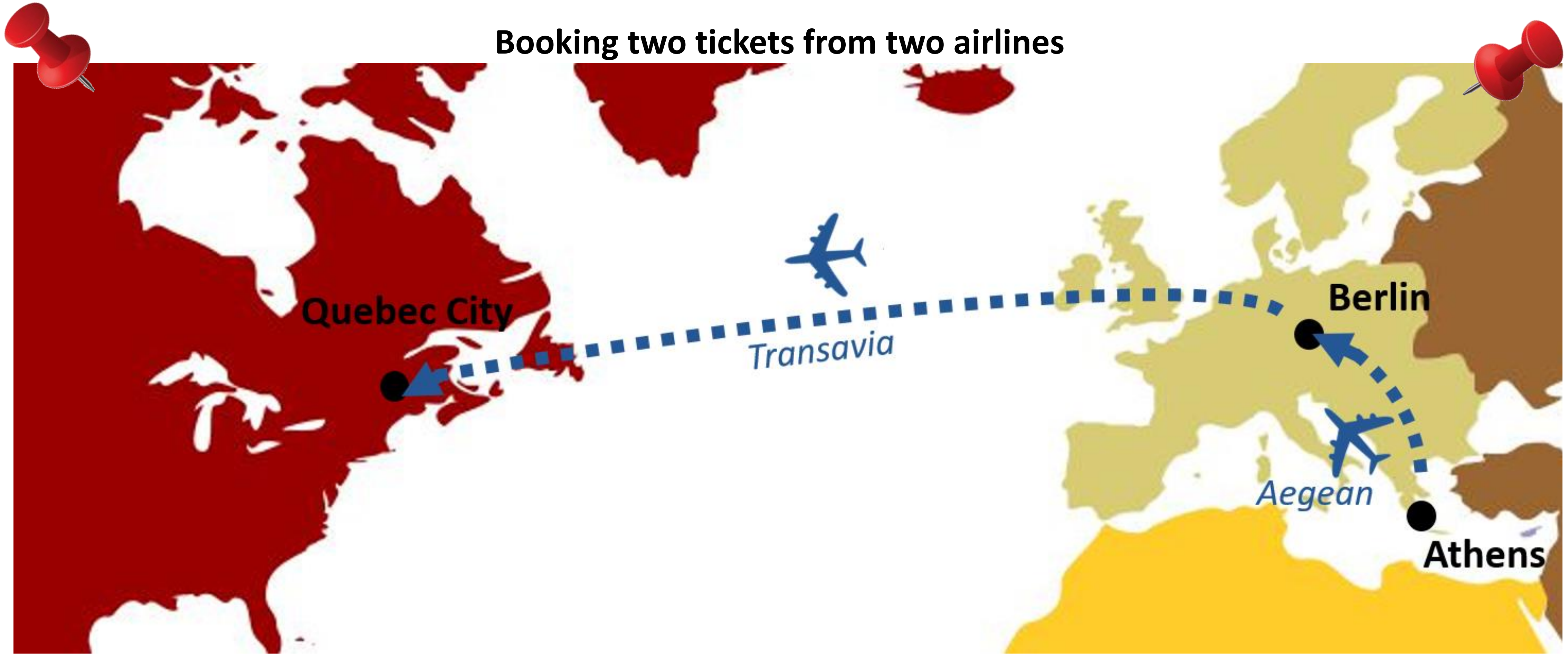
FORTH INSTITUTE OF COMPUTER SCIENCE

Expressing Transactions in BPMN

- Traditionally **BPMN2.0** is used to describe such **transactional** applications
- Since the majority uses saga → **BPMN2.0 supports sagas**
- BPMN2.0 has high expressive power (**semantically strong**)
- Let’s implement an **airline example** using BPMN2.0

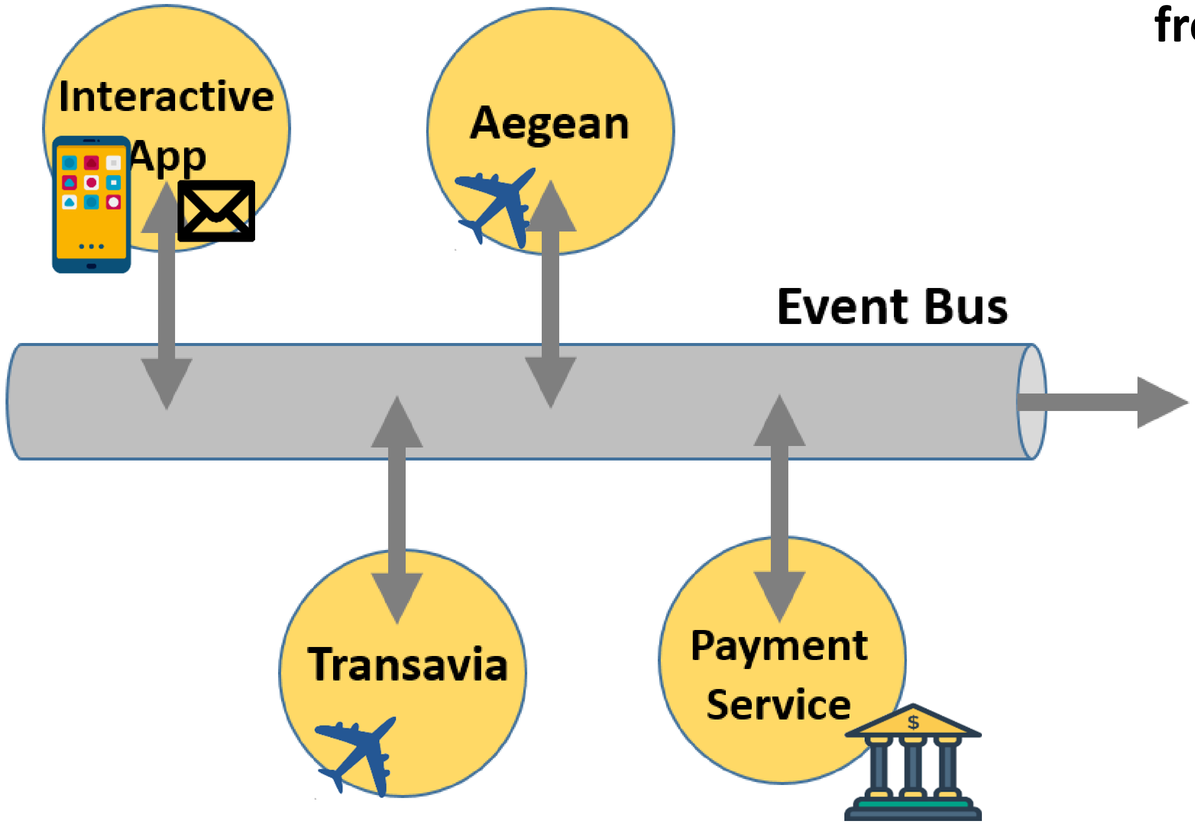
Airline paradigm explanation

Booking two tickets from two airlines



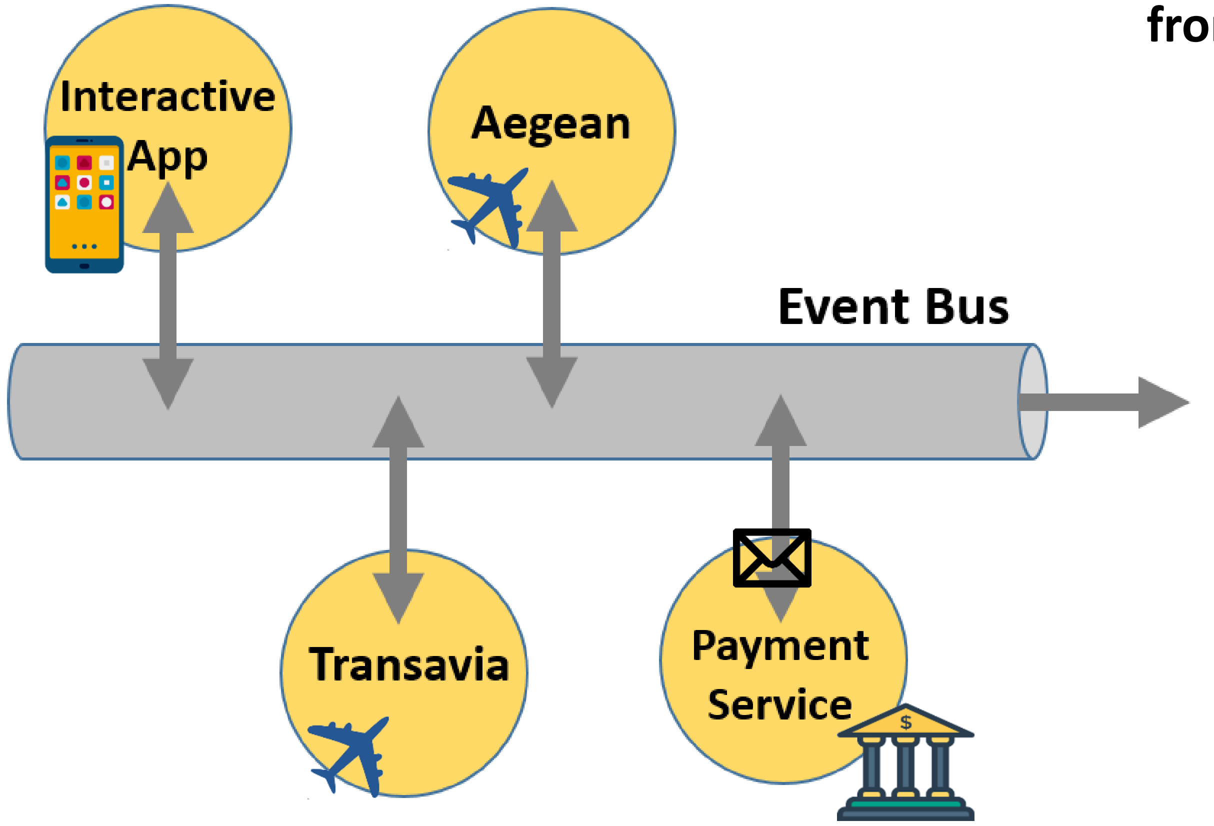
Possible Implementation, multiple participants

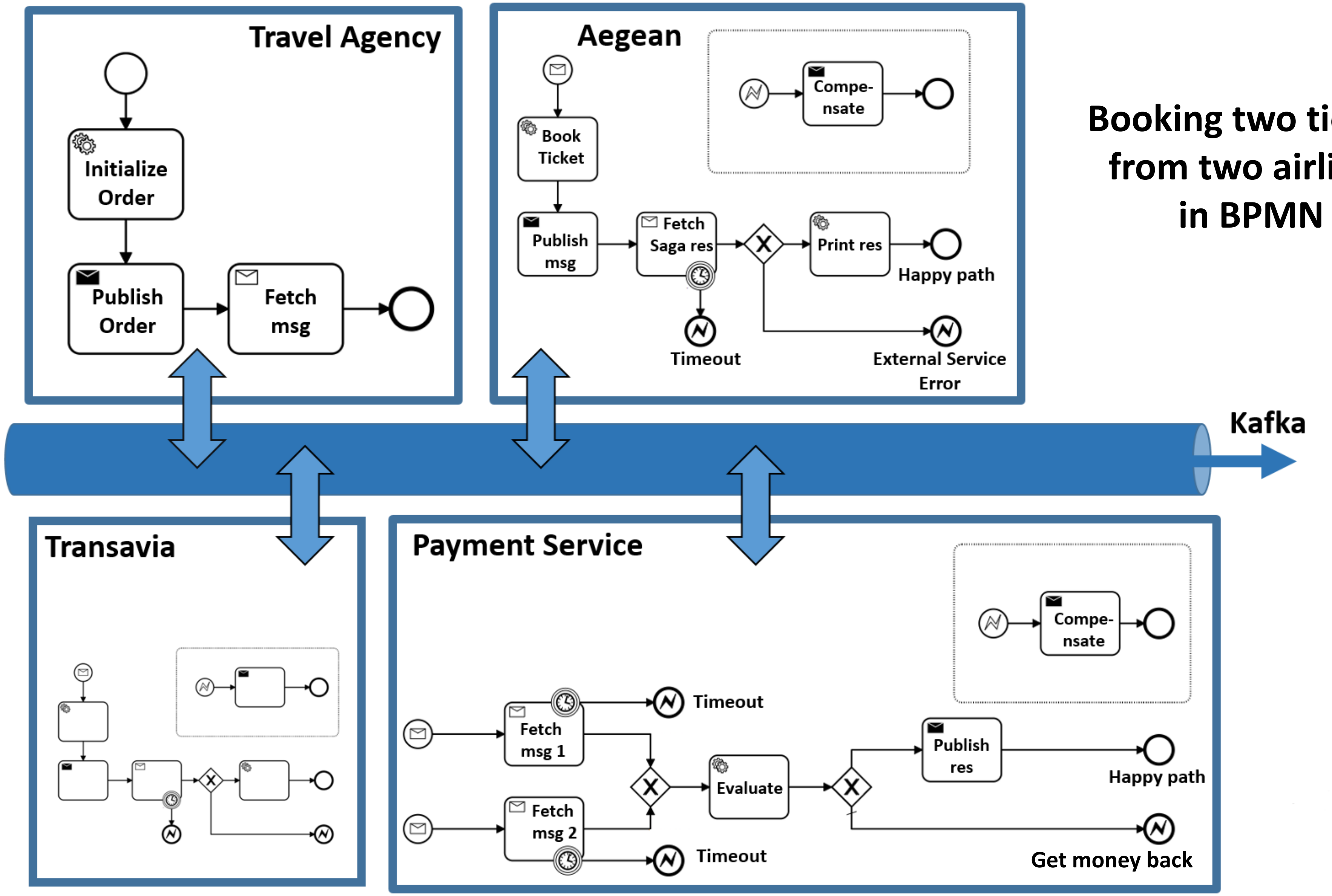
Booking two tickets from two airlines



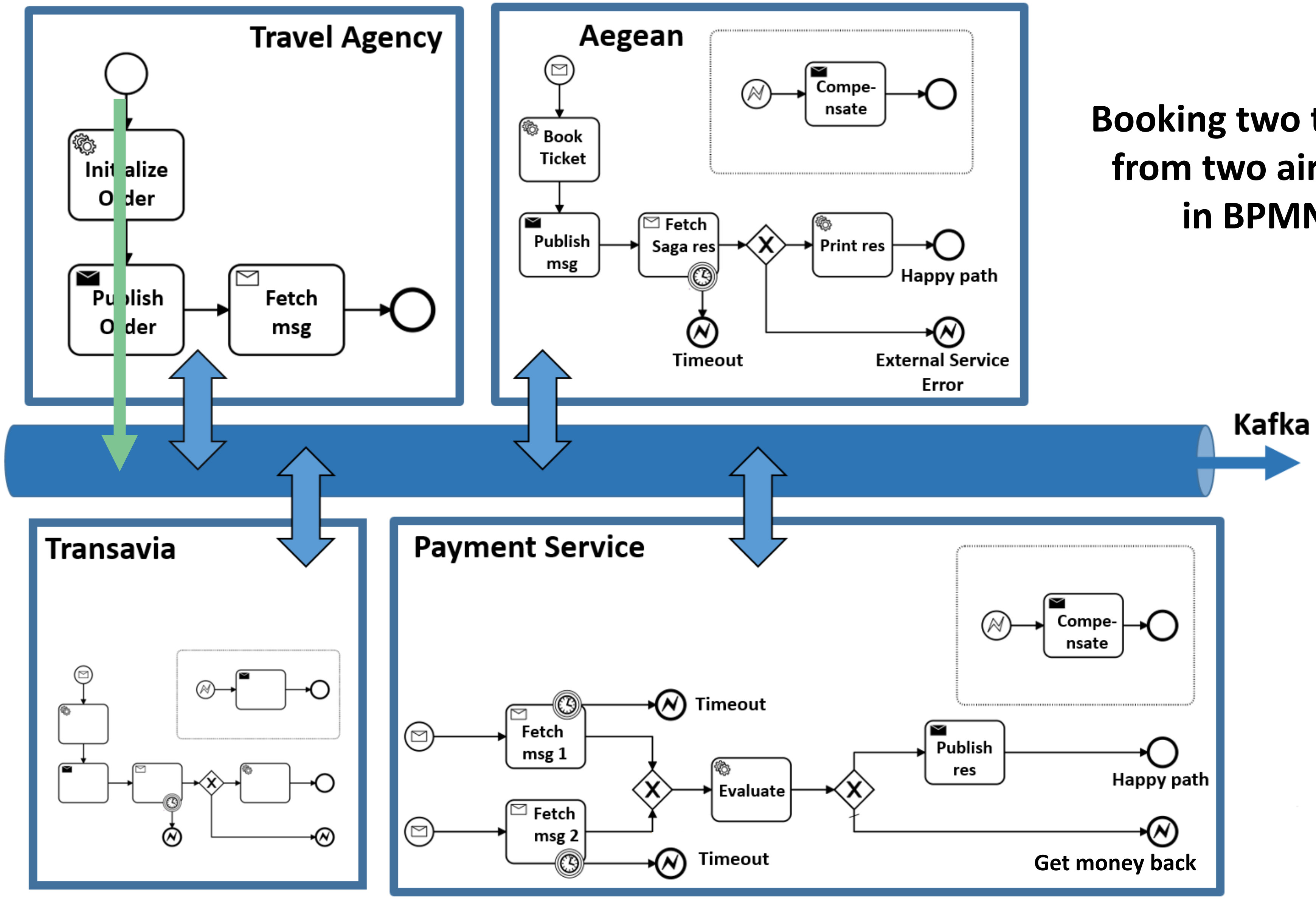
Possible Implementation, multiple participants

Booking two tickets
from two airlines



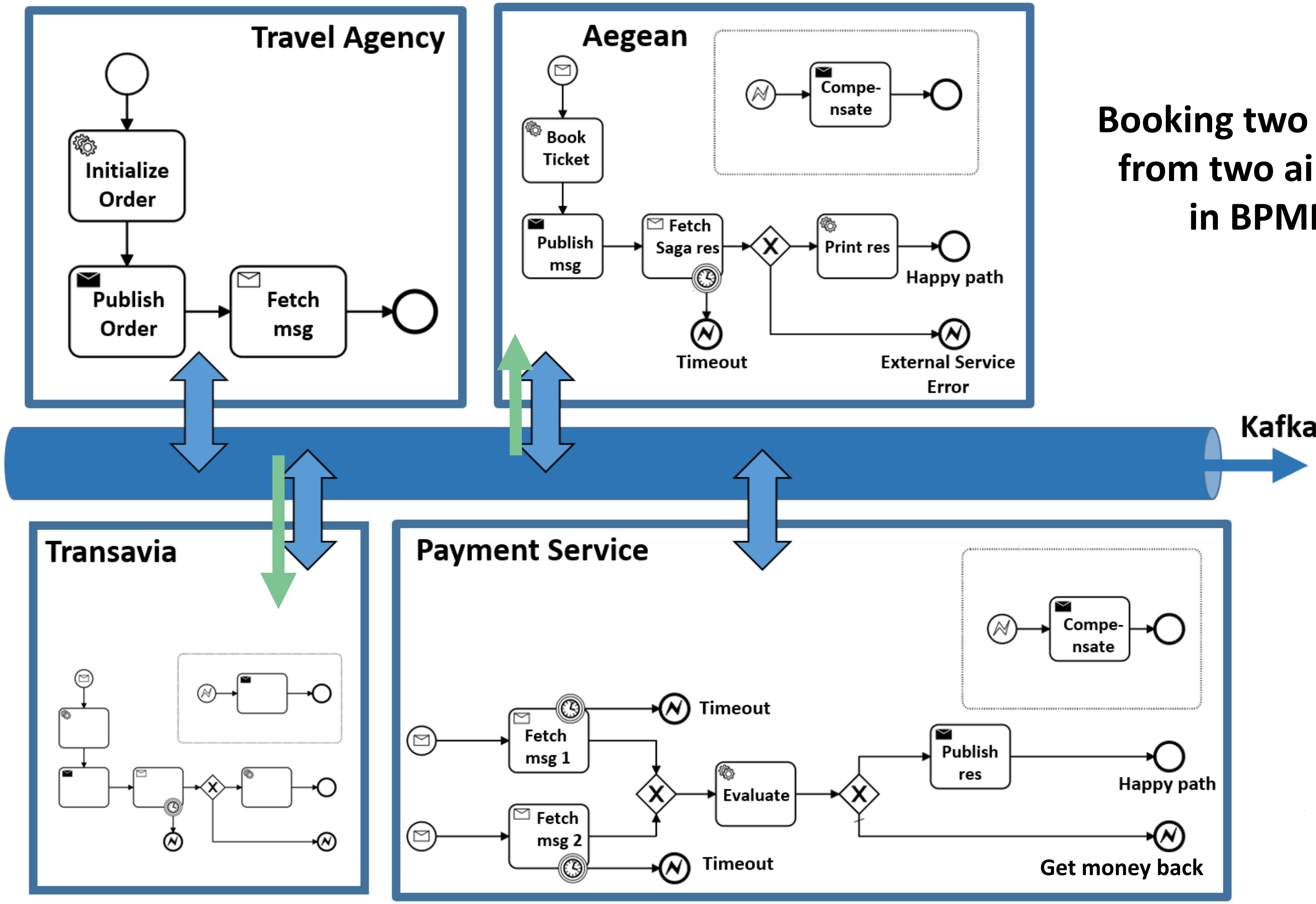


Booking two tickets from two airlines in BPMN

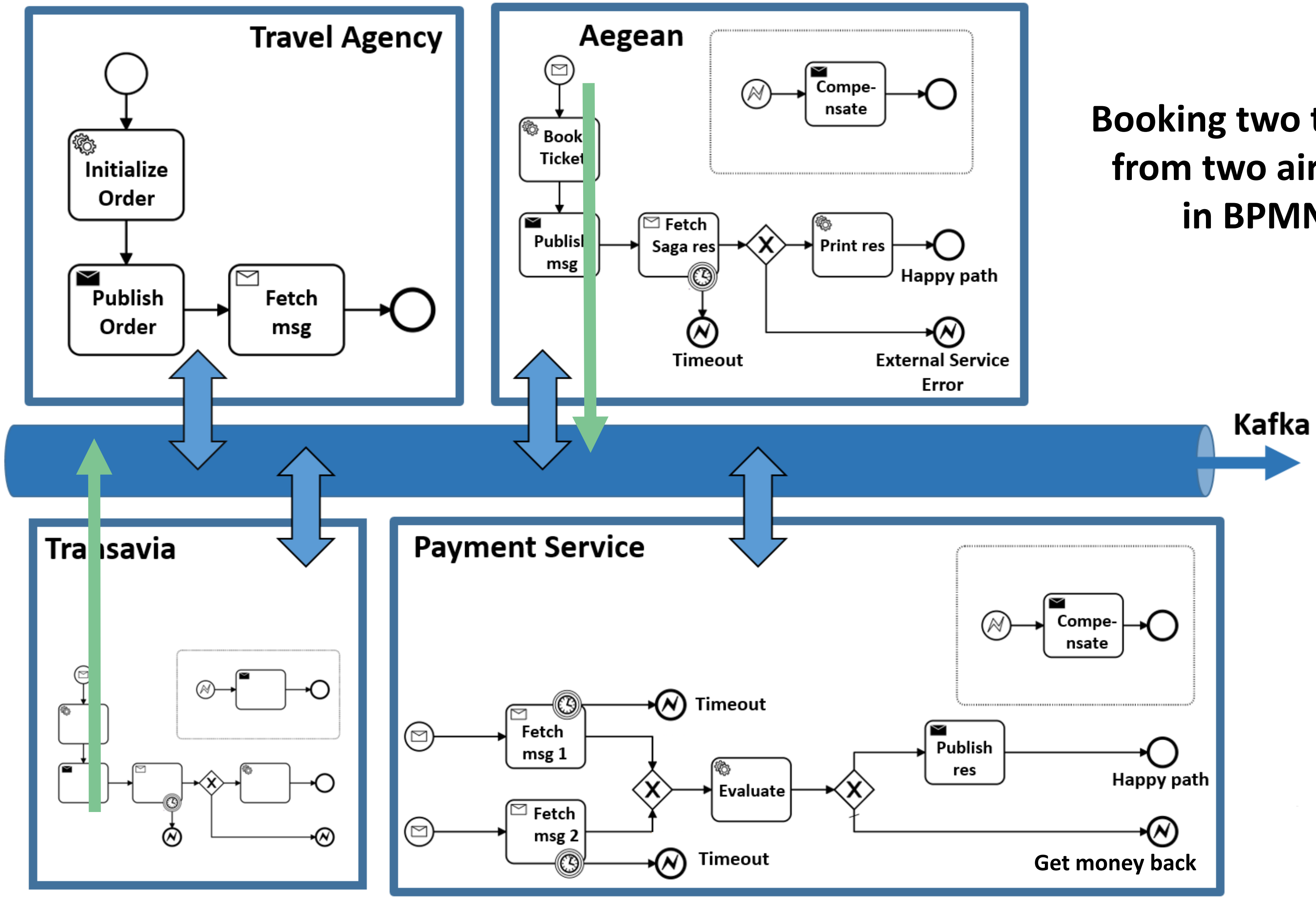


Booking two tickets from two airlines in BPMN

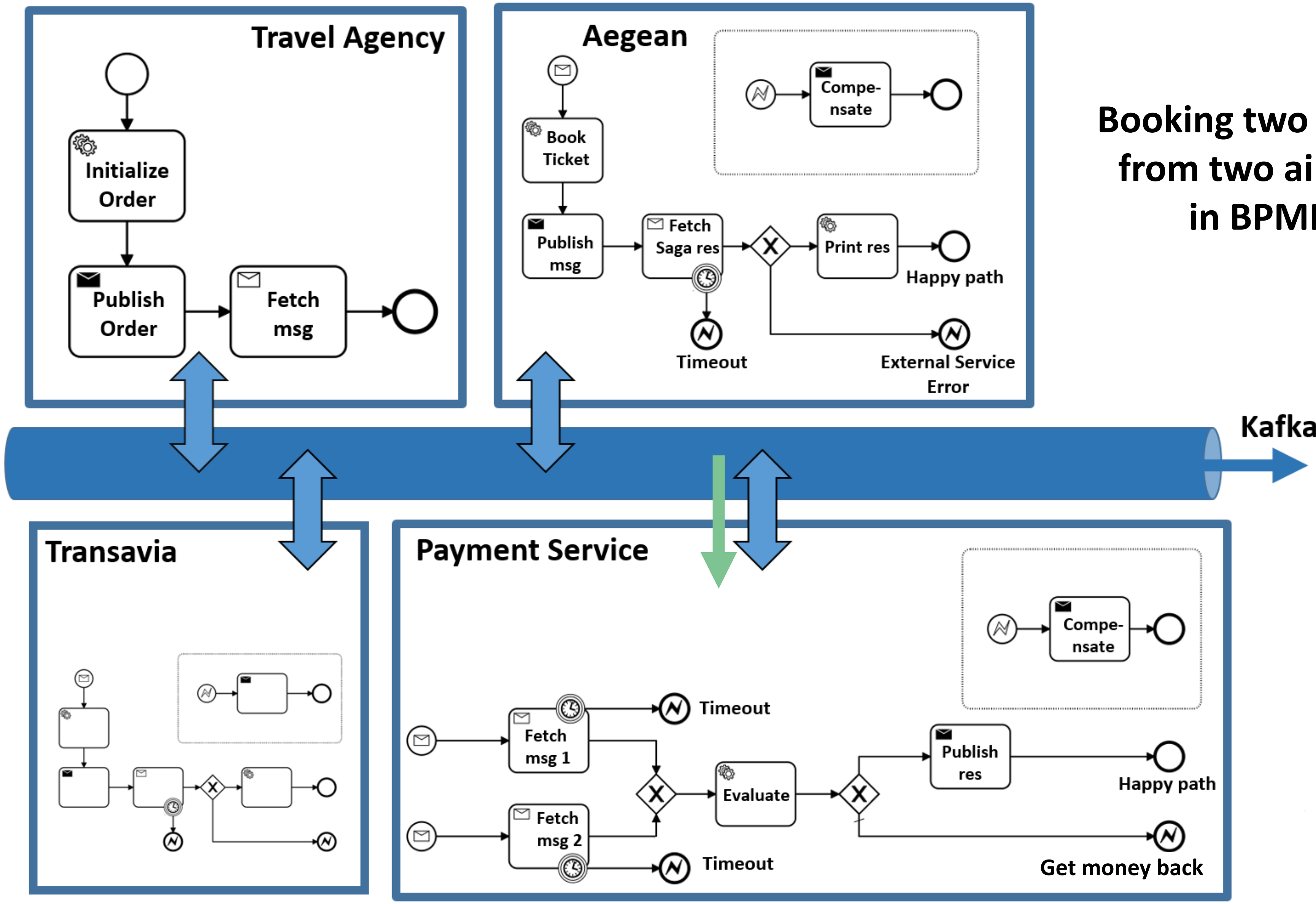
Booking two tickets from two airlines in BPMN



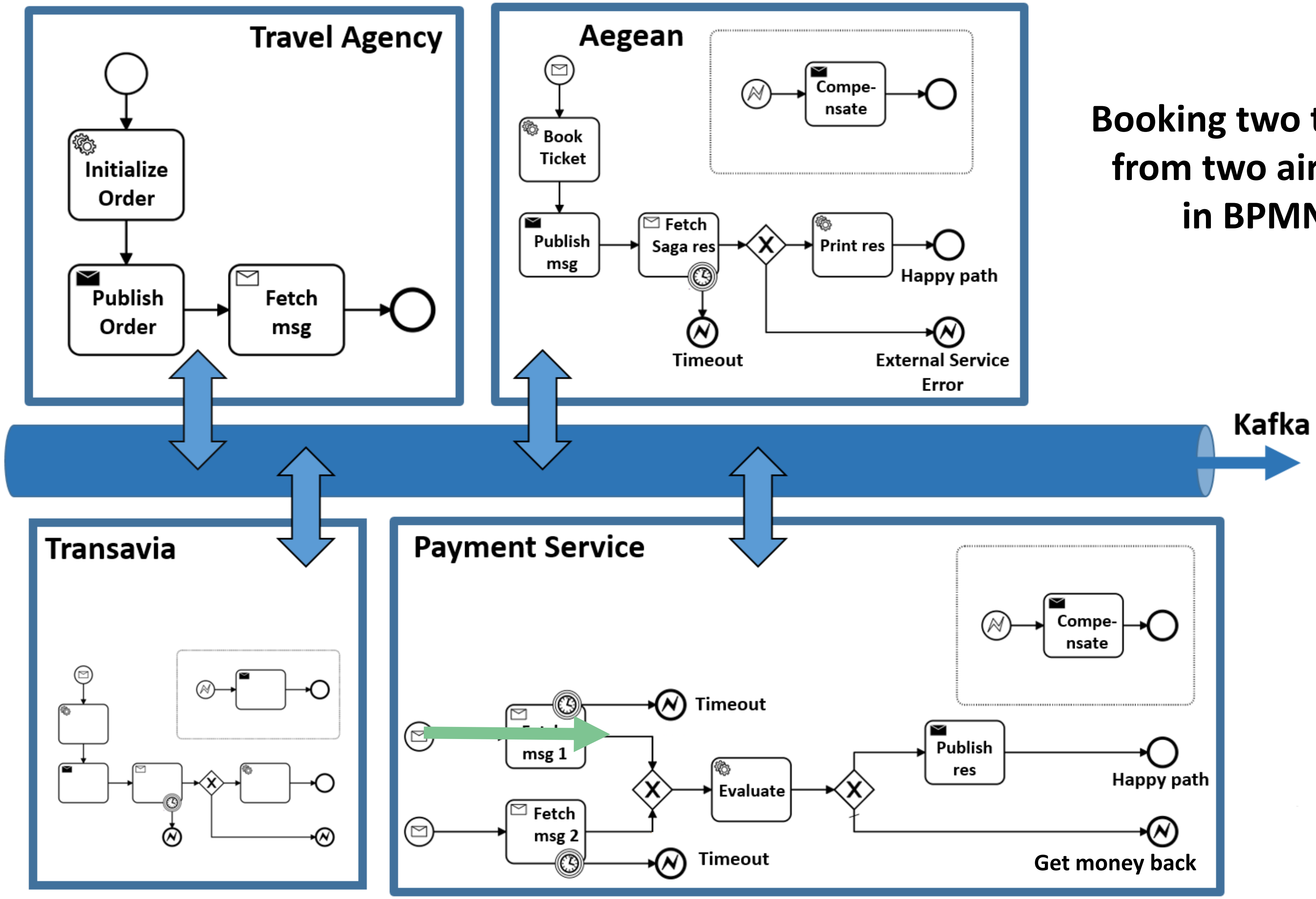
Booking two tickets from two airlines in BPMN



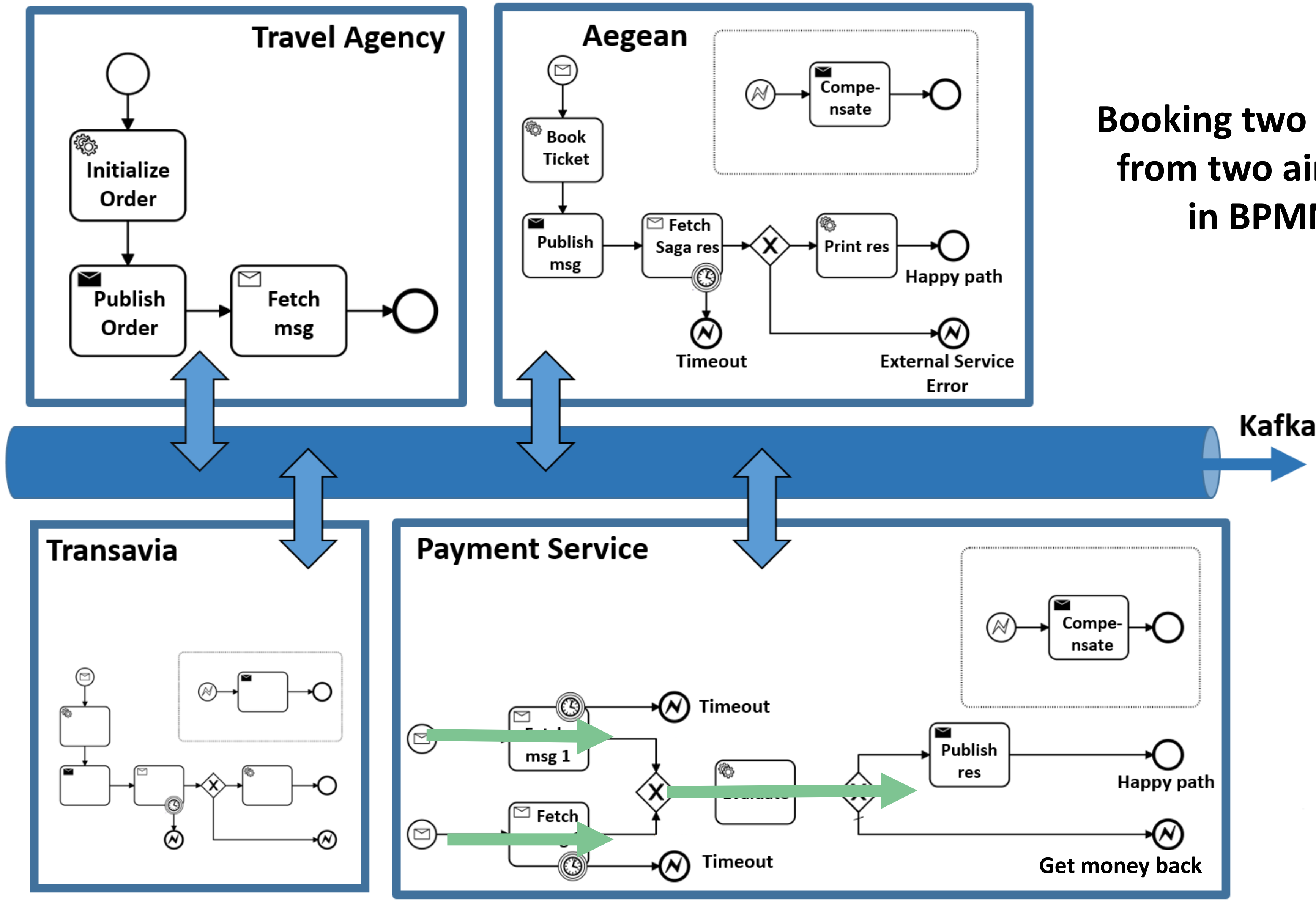
Booking two tickets from two airlines in BPMN

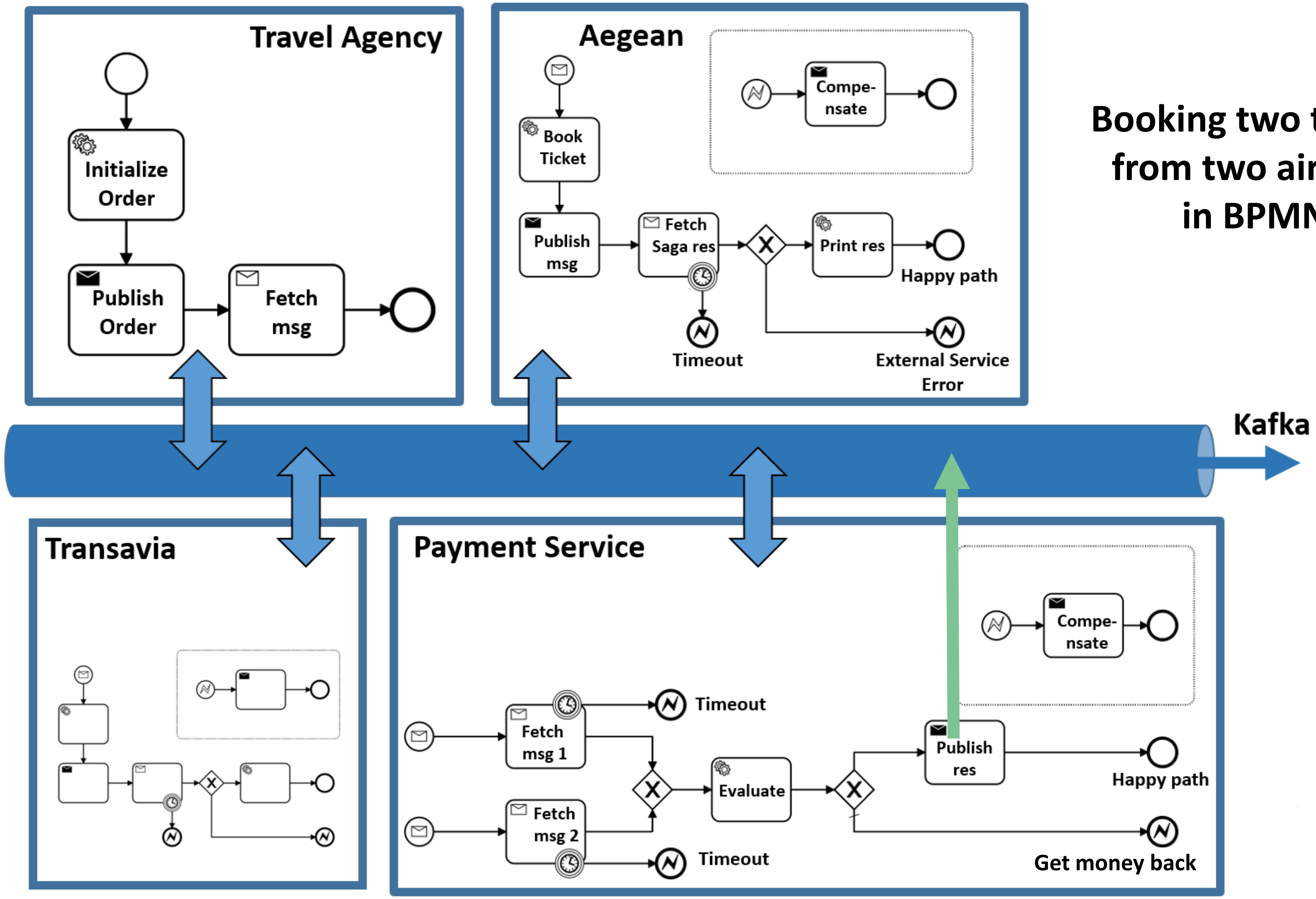


Booking two tickets from two airlines in BPMN

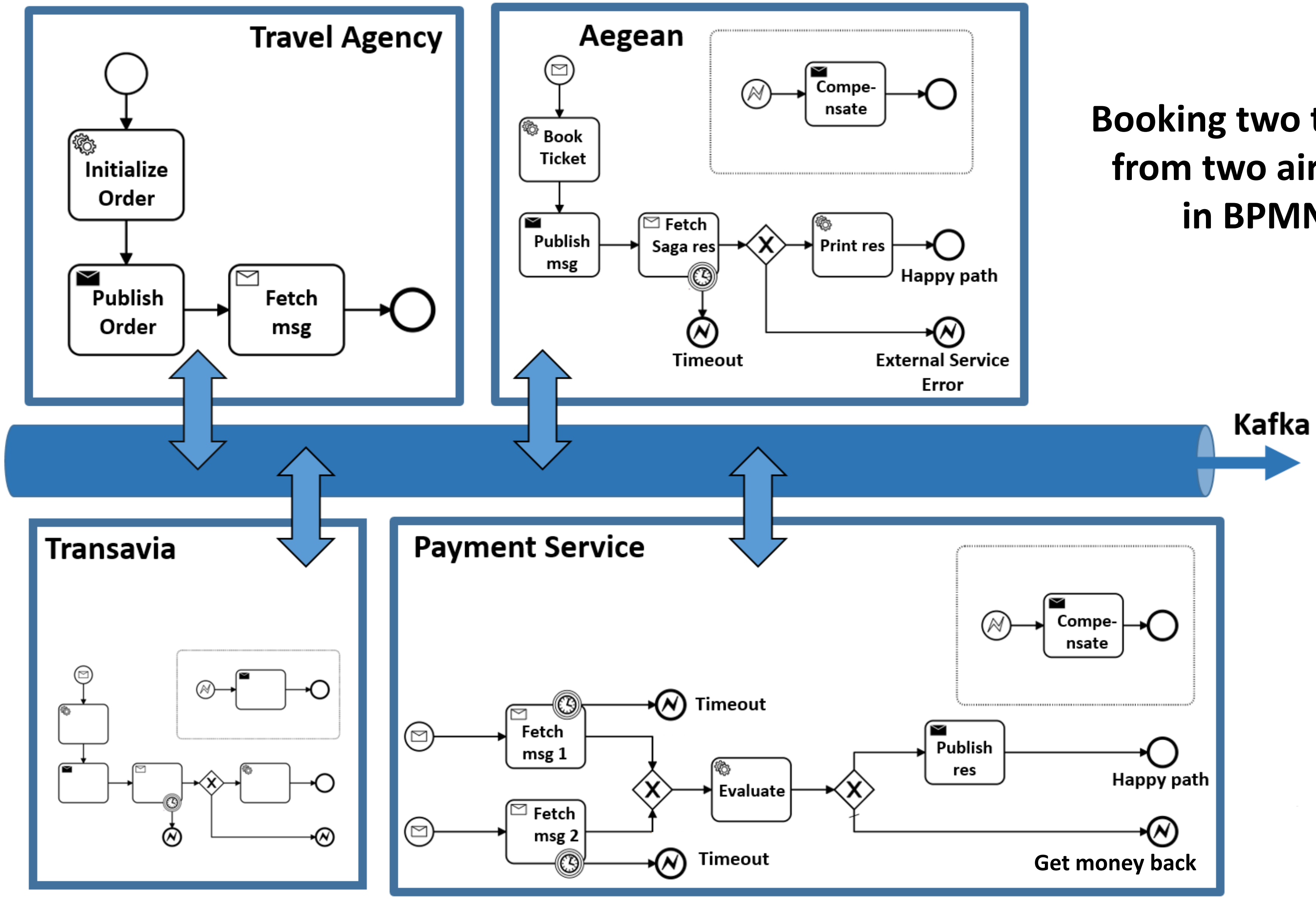


Booking two tickets from two airlines in BPMN

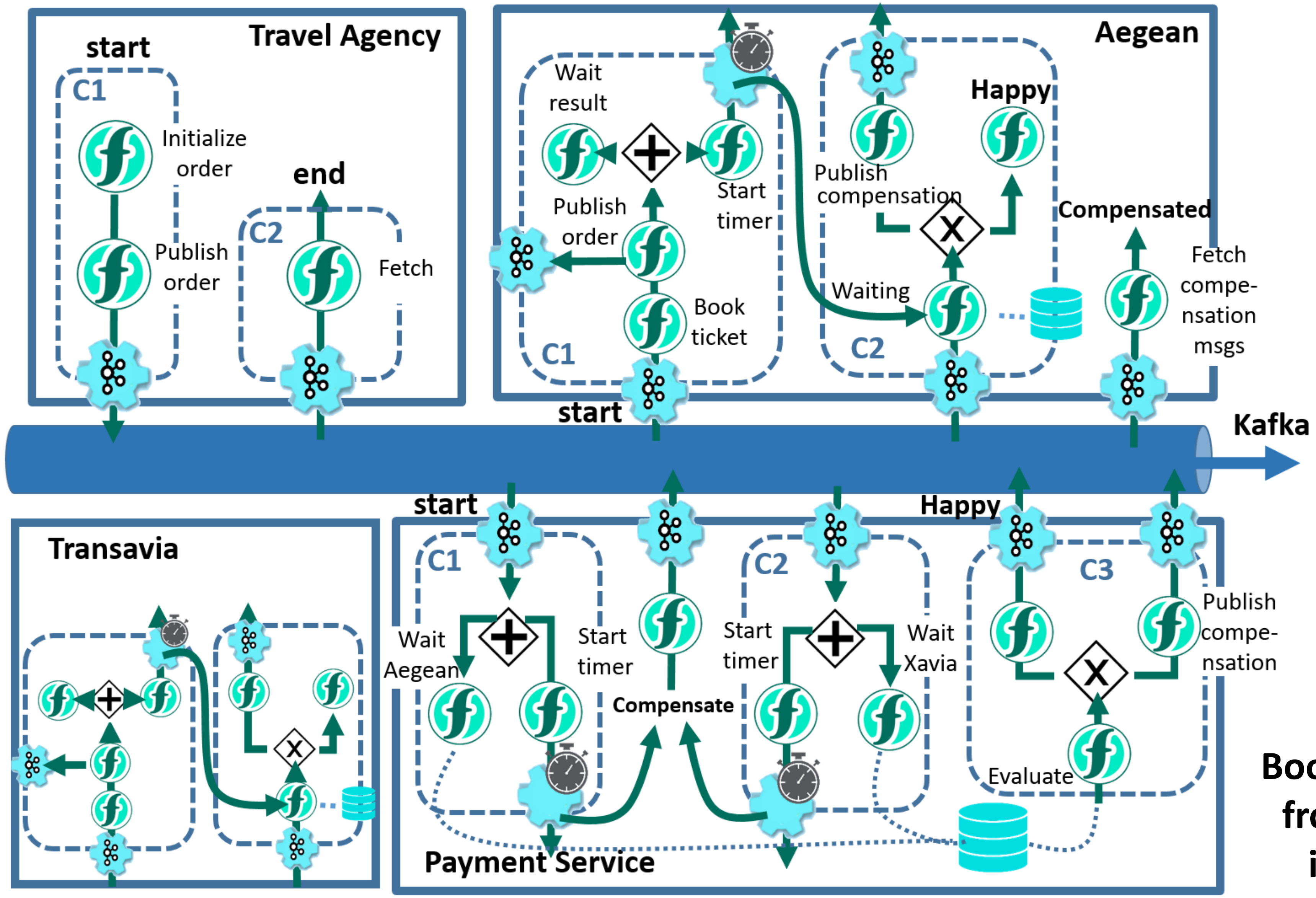




Booking two tickets from two airlines in BPMN



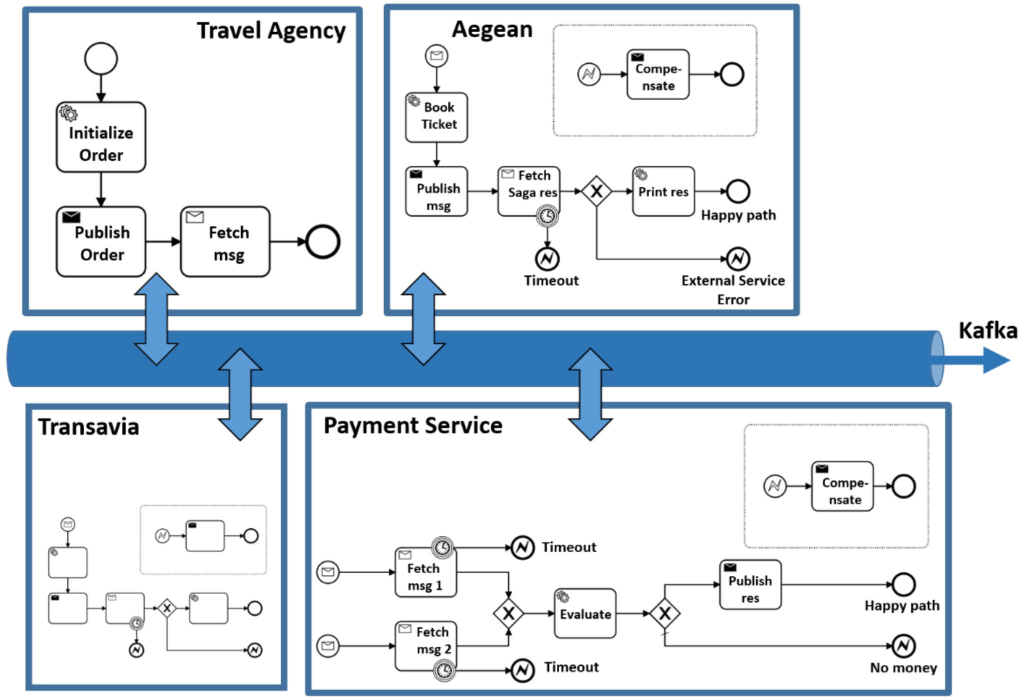
Booking two tickets from two airlines in BPMN



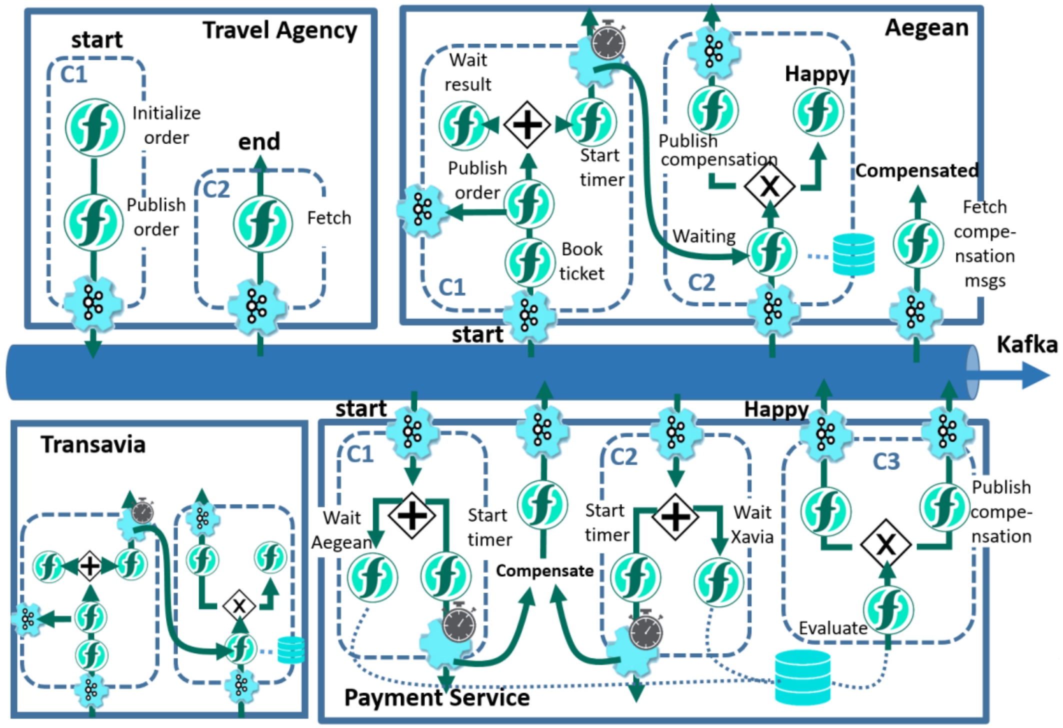
Booking two tickets from two airlines in OpenWhisk

FaaS workflow is typically more complex

BPMN



FAAS



FORTH
INSTITUTE OF COMPUTER SCIENCE

Conclusions

- **Addressed** challenges in **mapping** BPMN2.0 to OpenWhisk
- **Saga** transactions expressed in BPMN **straightforwardly** carry over to OpenWhisk
- This enables business analysts to express processes in a simpler BPMN format compared to the more complex OpenWhisk workflows
- Bridges simplicity of BPMN to ubiquity and power of FaaS platforms

FORTH
INSTITUTE OF COMPUTER SCIENCE

QUESTIONS ?