Microservices and Choreographies | The **SMAll** Project

Saverio Giallorenzo

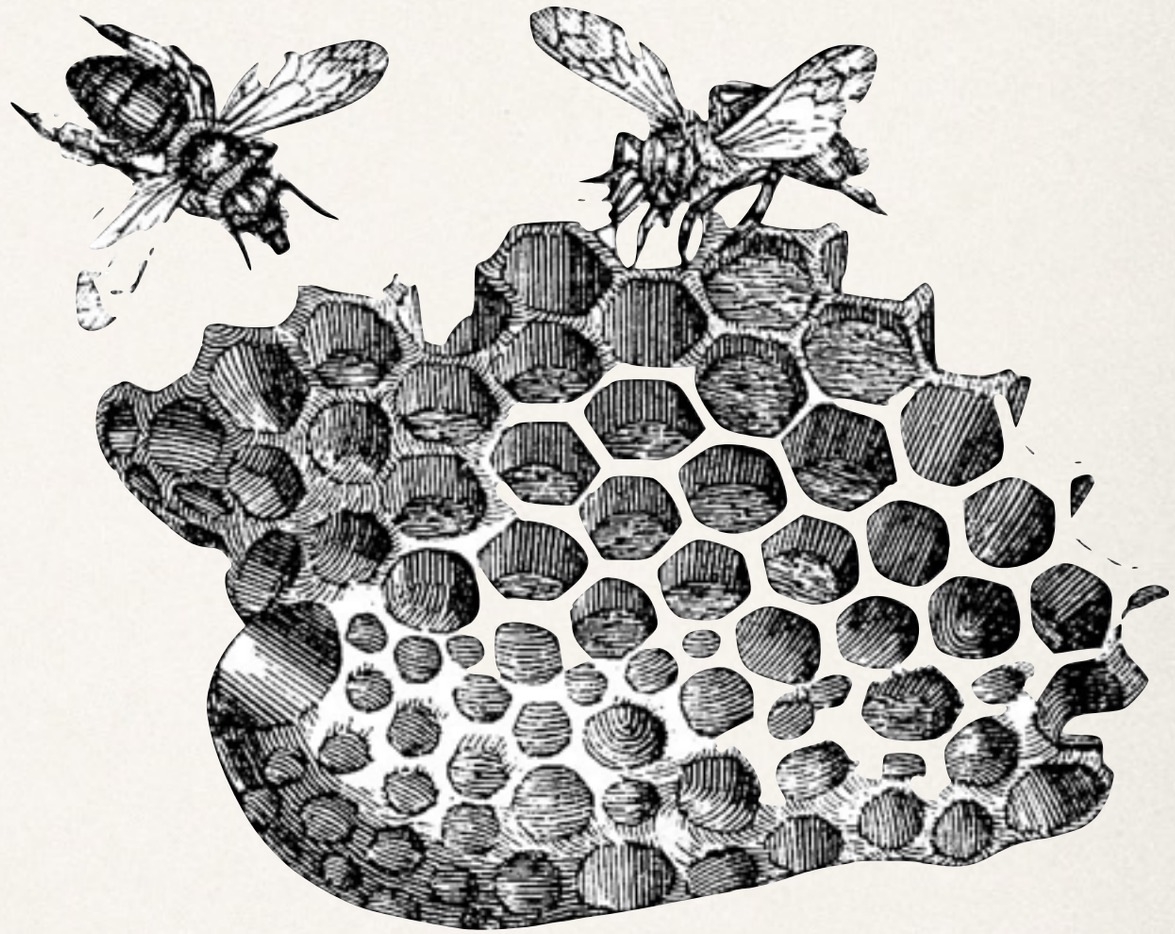
saverio.giallorenzo@gmail.com | *DISI at Unibo* | *Sophia Antipolis* | *Evaluation des Projects*

EIT Digital Project

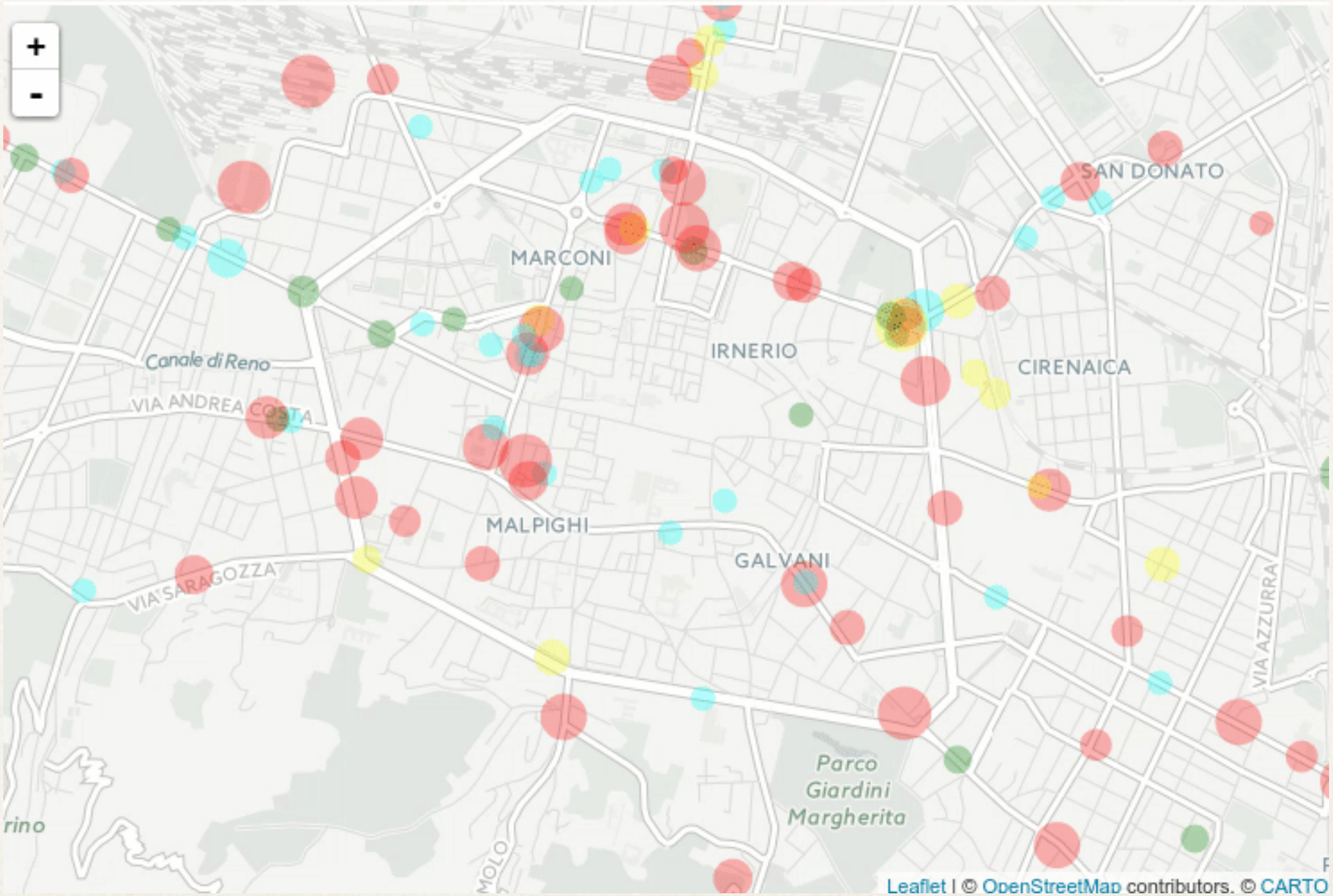
- **S**mart **M**obility for **All**
- Main Objective: creation of a global market of services for transportation;
- Project Partners: University of **Bologna**, FBK@**Trento**, Aalto University/Forum Virium@**Helsinki**
- Business Partners: Reply S.p.A., **Emilia-Romagna Region**, Trento Municipality, MaaS.fi/MaaS.global, …

Microservices

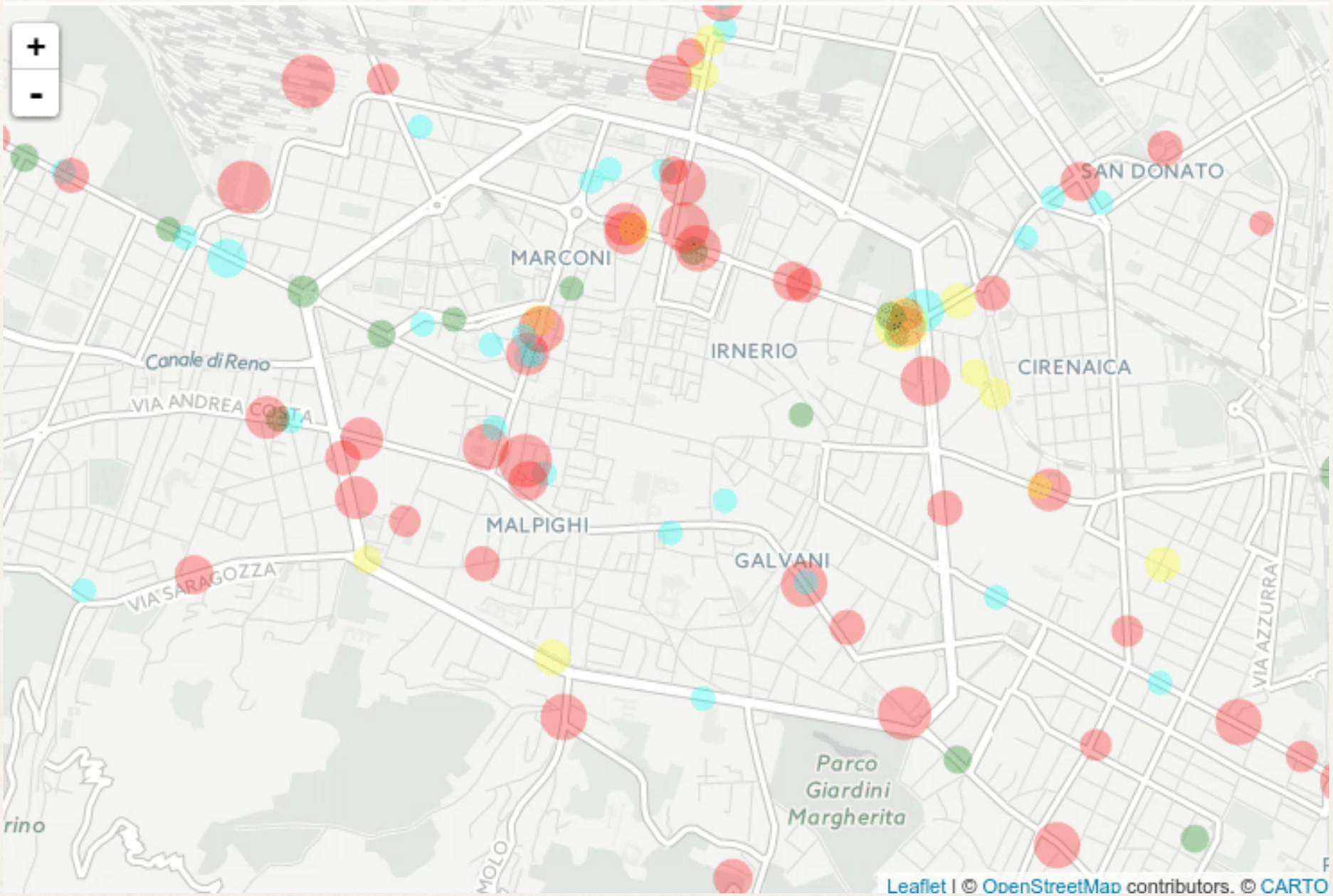
Architectures



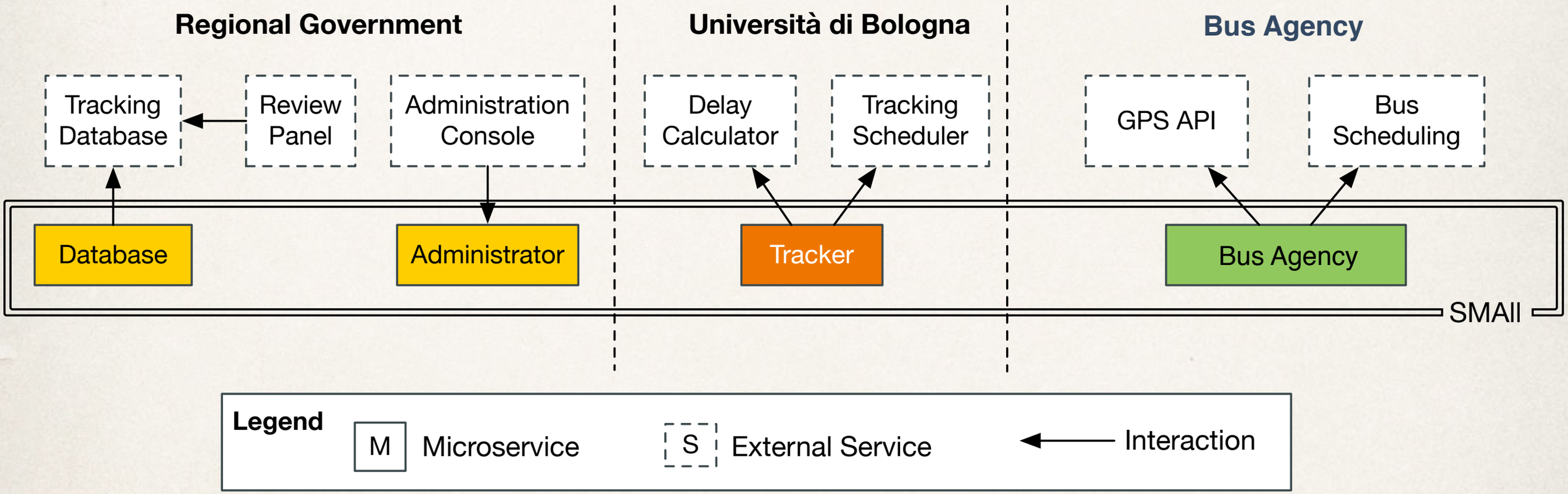
A SMAll Pilot | **BusCheck**



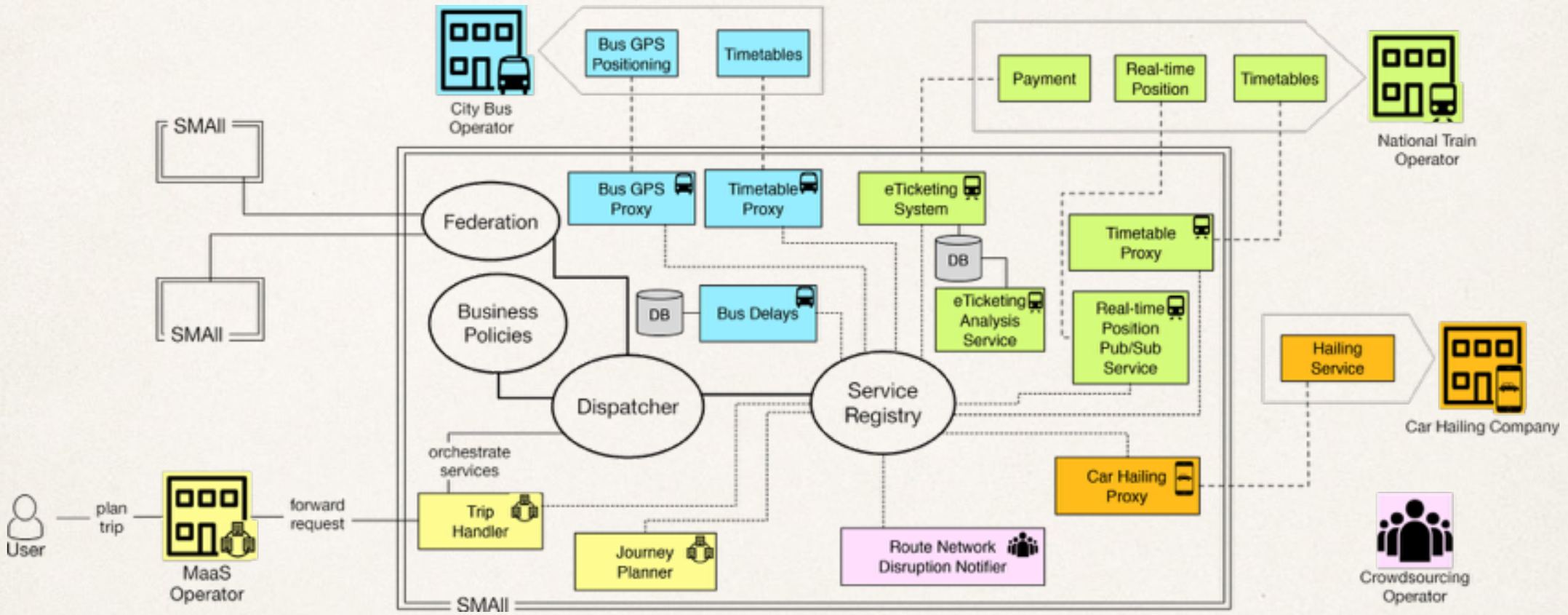
A SMAll Pilot | **BusCheck**



A SMAll Pilot | **BusCheck Architecture**



A Market for Microservices | The SMAll Platform



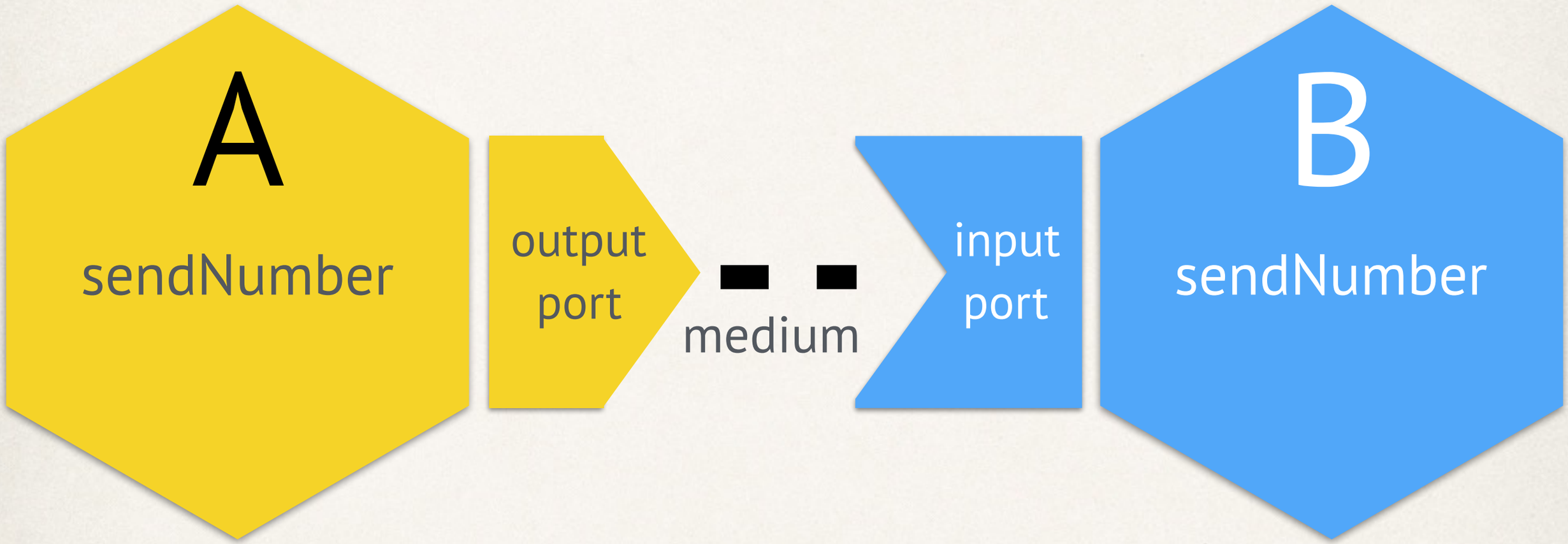
Microservices:

- cohesiveness & minimality
- fine-grained

→ - access policies;
→ - scalability;
→ - deployment (multistage continuous integration!).

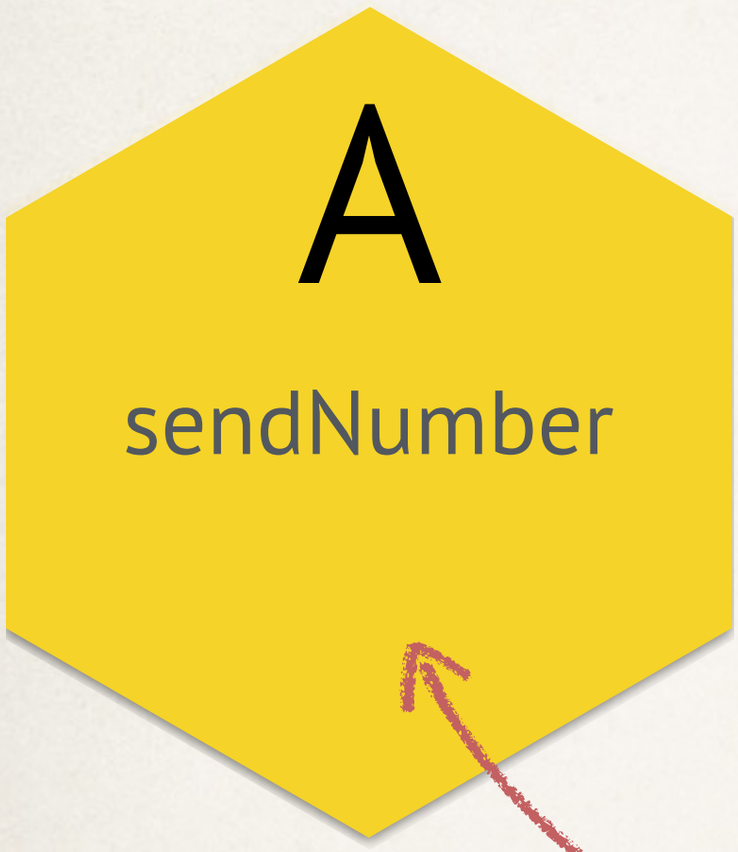
The First Language for Microservices

The Jolie Way



- Services communicate through **ports**.
- **Ports** give access to an **interface**.
- An interface is a set of **operations**.
- An **output port** is used to invoke interfaces exposed by other services.
- An **input port** is used to expose an interface.

The Jolie Way



Deployment:

```
OutputPort B {
  Location: socket://myaddress.org:80
  Protocol: json/rpc (xml/rpc, …)
  Interface: sendNumber( int )
}
```

Behaviour:

```
main {
 sendNumber@B( 5 )
}
```

The Jolie Way

```
interface PositionAPIInterface {
  OneWay: passPosition( PositionType )
}
```

```
include "PositionInterface.iol"
outputPort Tracker {
 Location:
   “socket://Tracker.com:80"
 Protocol: json/rpc
 Interfaces: PositionAPIInterface
}

main
{
 passPosition @ Tracker ( gps )
}
```

Bus Agency

```
include “PositionInterface.iol"
inputPort Tracker  {
 Location:
  “socket://Tracker.com:80"
 Protocol: json/rpc
 Interfaces: PositionAPIInterface
}

main
{
 passPosition ( gps )
}
```

Tracker

The Jolie Way

```
interface PositionAPIInterface {
  OneWay: passPosition( PositionType )
}
```

```
include "PositionInterface.iol"
outputPort Tracker {
 Location:
  “socket://Tracker.com:80"
 Protocol: json/rpc
 Interfaces: PositionAPIInterface
}

main
{
 passPosition @ Tracker ( gps )
}
```

Bus Agency

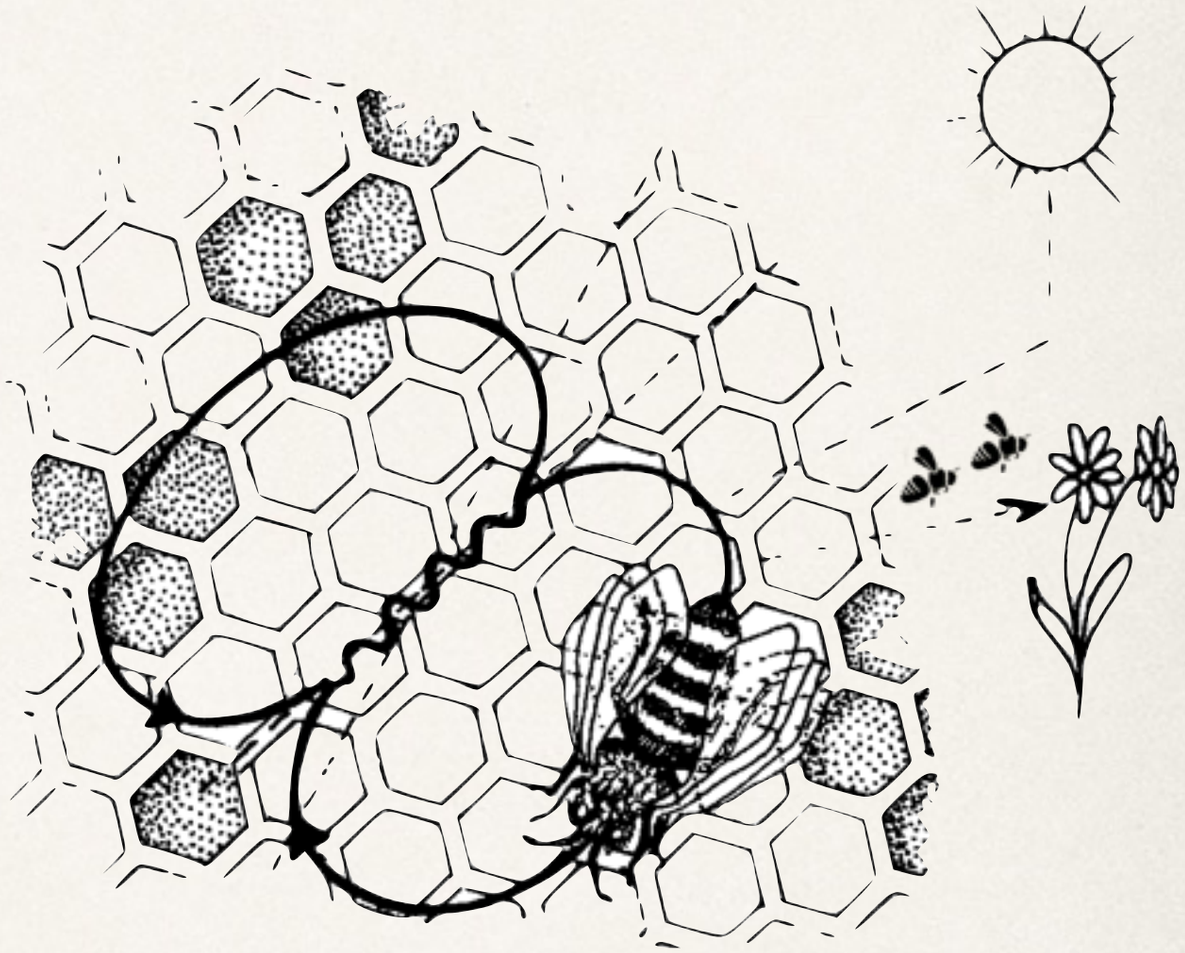
```
include “PositionInterface.iol"
inputPort Tracker  {
 Location:
  “socket://Tracker.com:80"
 Protocol: json/rpc
 Interfaces: PositionAPIInterface
}

main
{
 passPosition ( gps )
}
```

Tracker

Choreographies

Protocols,

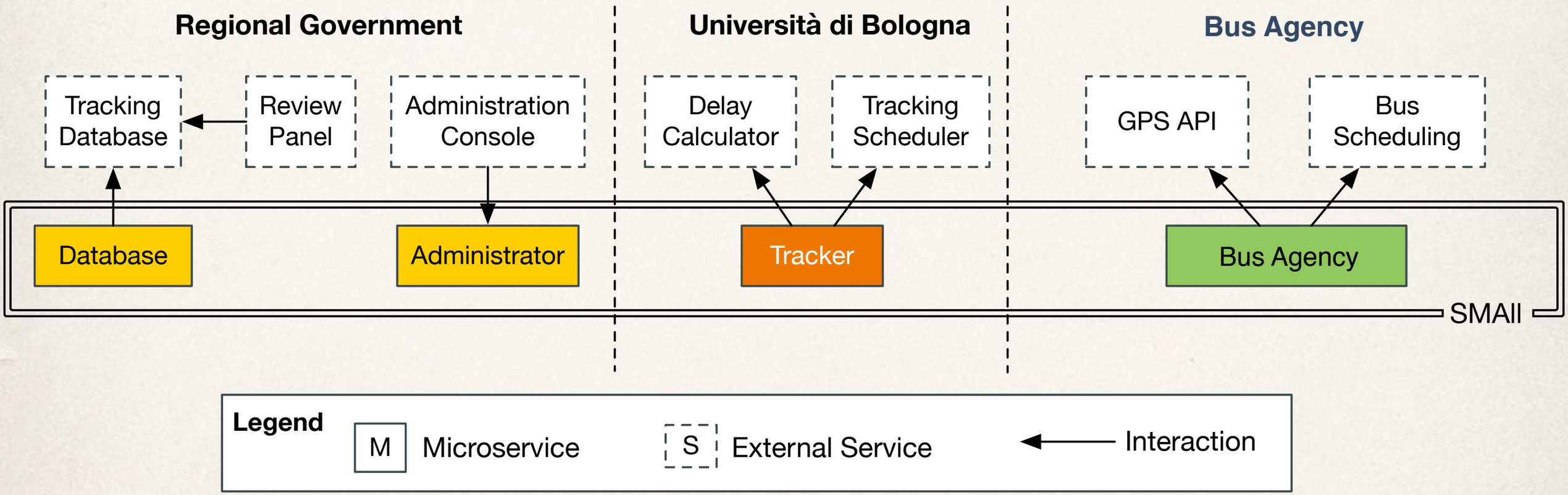
Correct implementations



Choreography

```
include getInput from Region.AdministrationConsole
include insertDelay from Region.TrackingDatabase
include hasNextStop from UniBo.TrackingScheduler
include calculateDelay from UniBo.DelayCalculator
include getBusSchedule from BusAgency.BusScheduling
include getPosition from BusAgency.GpsAPI

preamble{ starter: Admin }

aioc {
 line@Admin = getInput( "Insert line to track" );
 {
  setLine: Admin( line ) -> DB( line )
  |
  setLine: Admin( line ) -> BusAgency( line )
 };
 schdl@BusAgency = getBusSchedule( line );
 passSchdl: BusAgency( schdl ) -> Tracker( schdl );
 hasNext@Tracker = hasNextStop( schdl );
 while( hasNext )@Tracker {
  gps@BusAgency = getPosition( line );
  passPosition: BusAgency( gps ) -> Tracker( gps );
  delay@Tracker = calculateDelay( schdl, gps );
  storeDelay: Tracker( delay ) -> DB( delay );
  {
   _@DB = insertDelay( line, delay )
   |
   hasNext@Tracker = hasNextStop( sched )
  }
 }
}
```

A SMAll Pilot | BusCheck Architecture



Correctness by design and by construction

Choreography —*EPP*→ Endpoint Projection

(Correct by design) *(Correct by construction)*

```
line@Admin = getInput( "Insert line to track" );
{
 setLine: Admin( line ) -> DB( line )
  |
 setLine: Admin( line ) -> BusAgency( line )
};
```

EPP

Admin

```
getInput@UI( "Insert line to track" )( lin
{ setLine@Database( line )
  |
  setLine@BusAgency( line ) };
...
```

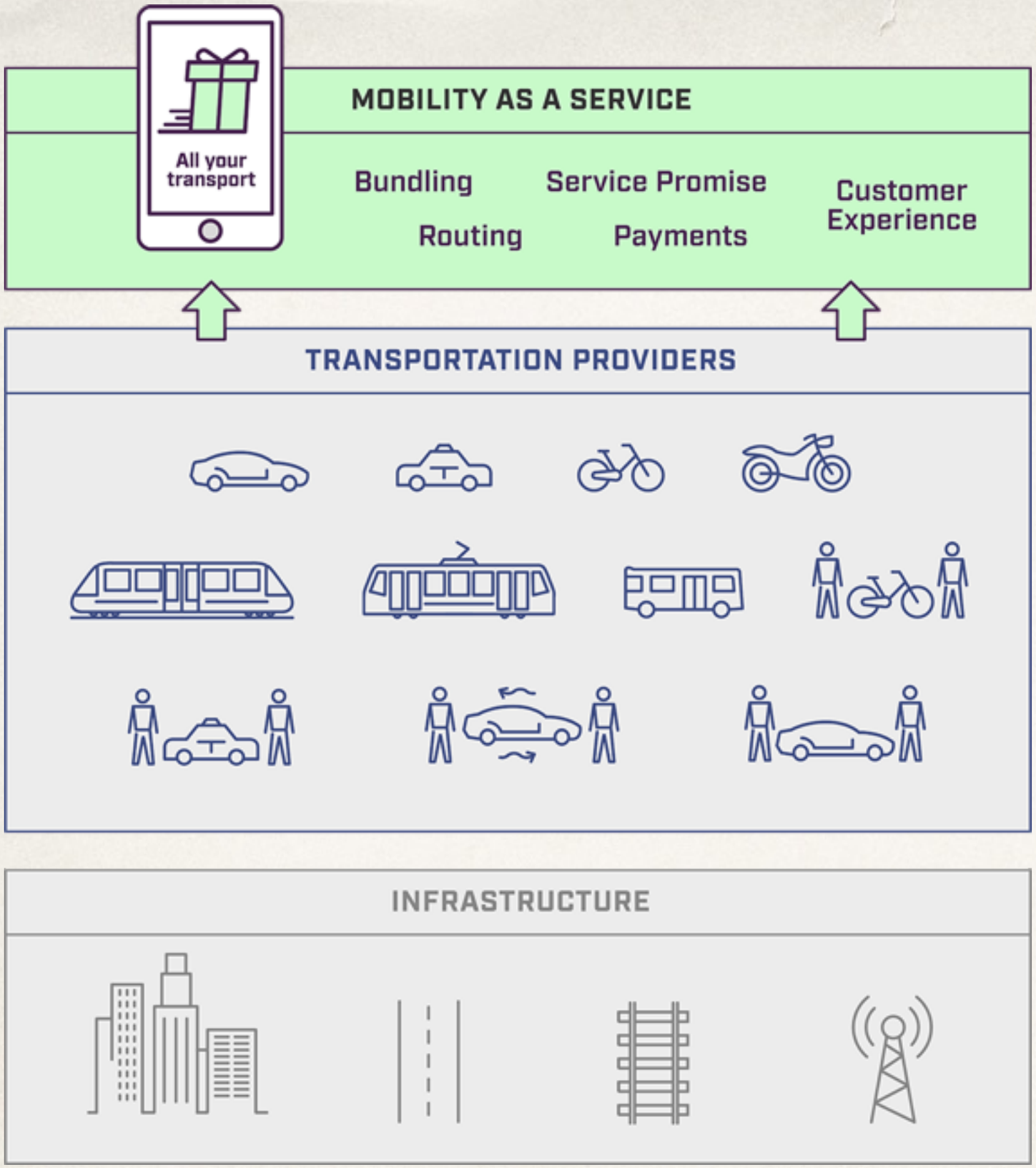
Database

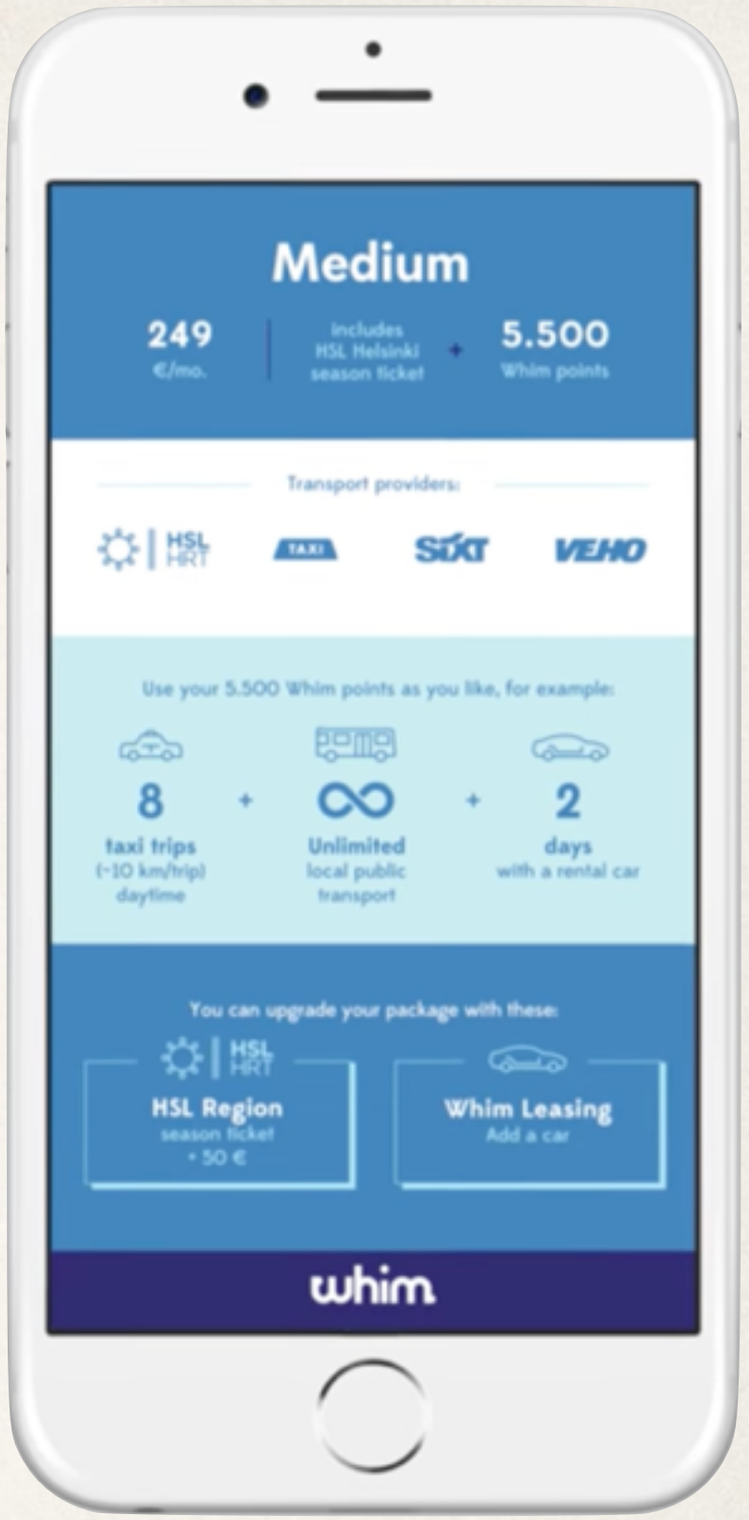
```
setLine( line );
...
```

BusAgency

```
setLine( line );
...
```

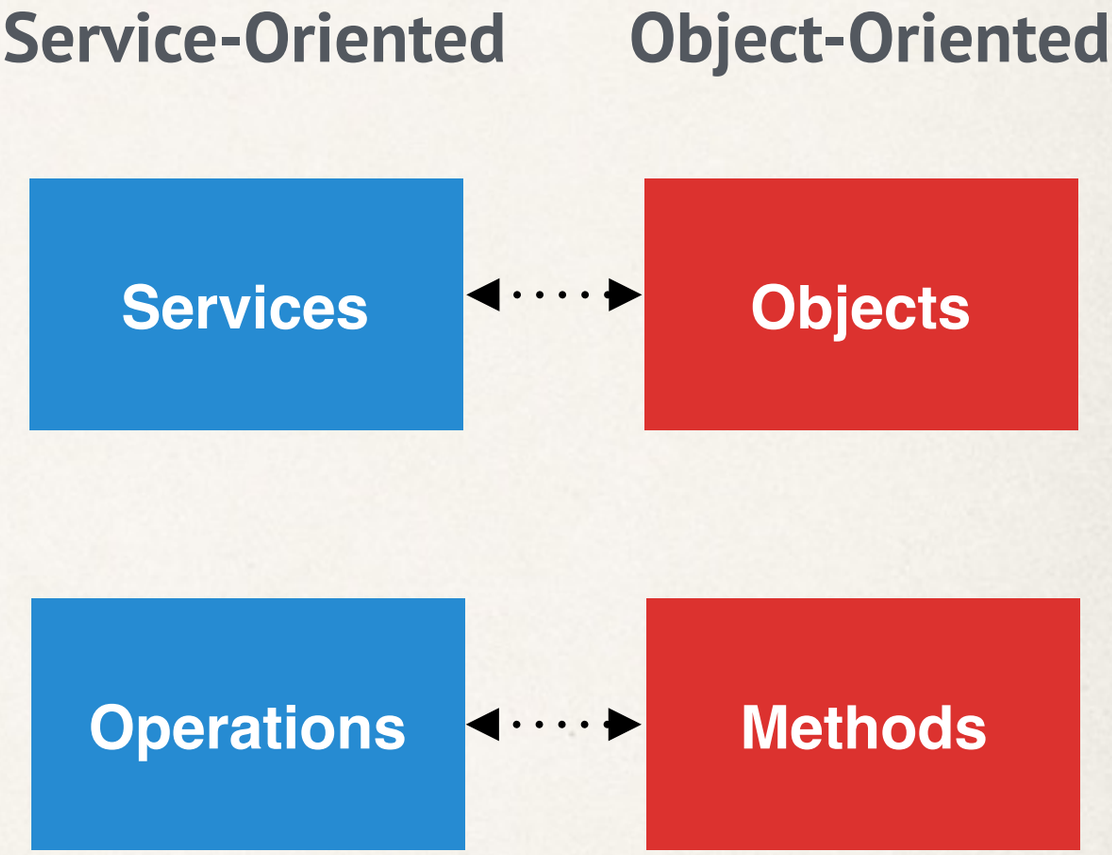
Time for discussion!


All your transport
MOBILITY AS A SERVICE
Bundling
Service Promise
Customer Experience
Routing
Payments
TRANSPORTATION PROVIDERS
INFRASTRUCTURE


Medium
249
€/mo.
includes HSL Helsinki season ticket
+
5.500
Whim points
Transport providers:
HSL HRT
TAXI
SIXT
VEHO
Use your 5.500 Whim points as you like, for example:
8
taxi trips
(~10 km/trip)
daytime
+
Unlimited
local public transport
+
2
days
with a rental car
You can upgrade your package with these:
HSL HRT
HSL Region
season ticket
- 50 €
Whim Leasing
Add a car
whim

Microservices

- Cohesiveness & Minimality
- API design is paramount;
- Partition of work and parallel development;
- Breakdown of complexity into “simple” and specialised services;
- Integrate ESB-like functionalities;
- Lightweight and human-oriented protocols (REST, JSON, etc.).



AIOCJ

Triggers compilation to Jolie