**Enterprise Integration (MEIC-A, 2019-20, 2º semestre)**

Instituto Superior Técnico – MEIC-A

*Sprint 2 Report*

1. **Definition of the microservices needed for the MaaS functionality**

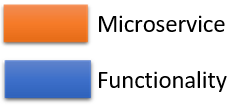


Fig. 1 – Microservices and functionalities

1. **Chosen microservices**

Customer Management and Operator management

1. **Microservices input and output**

Customer Management Service:

**Input**

This service takes as input new user events and trip events that are consumed from the operator topics.

There are 3 types of trip events:

* **Type 0 operators** (Check in and check out method)

These

The new user events contain as main fields eventType and user. The user field contains information about a user: id, email, plan type, first name, last name, balance and a field that says if the user has a pass or not.

{

    "eventType": "new-user",

    "user": {

        "id": "69c594cfdeeaedd220",

        "email": "user@gmail.com",

        "planType": "pre-paid",

        "firstName": "Paulo",

        "lastName": "Neves",

        "balance": "500",

        "hasPass": "true"

    }

}

<root>

  <eventType>new-user</eventType>

  <user>

    <id>69c594cfdeeaedd220</id>

    <email>user@gmail.com</email>

    <firstName>Paulo</firstName>

    <lastName>Neves</lastName>

    <planType>pre-paid</planType>

    <balance>500</balance>

    <hasPass>true</hasPass>

  </user>

</root>

**Output**

{

    "eventType": "trip-cost",

    "info": {

        "baseCost": "23",

        "token": "69c594cfdeeaedd220",

        "hasPass": "true",

        "tripId": "tod89430d",

        "operatorType": "t1",

        "operatorName": "Uber",

        "timeStamp": "2020-02-29 20:57:10.294"

    }

}

<root>

  <eventType>trip-cost</eventType>

  <info>

    <baseCost>23</baseCost>

    <token>69c594cfdeeaedd220</token>

    <hasPass>true</hasPass>

    <tripId>tod89430d</tripId>

    <operatorType>t1</operatorType>

    <operatorName>Uber</operatorName>

    <timeStamp>2020-02-29 20:57:10.294</timeStamp>

  </info>

</root>

Operator Management Service:

Input

{

    "eventType": "trip-cost",

    "info": {

        "baseCost": "23",

        "token": "69c594cfdeeaedd220",

        "hasPass": "true",

        "tripId": "tod89430d",

        "operatorType": "t1",

        "operatorName": "Uber",

        "timeStamp": "2020-02-29 20:57:10.294"

    }

}

<root>

  <eventType>trip-cost</eventType>

  <info>

    <baseCost>23</baseCost>

    <token>69c594cfdeeaedd220</token>

    <hasPass>true</hasPass>

    <tripId>tod89430d</tripId>

    <operatorType>t1</operatorType>

    <operatorName>Uber</operatorName>

    <timeStamp>2020-02-29 20:57:10.294</timeStamp>

  </info>

</root>

Output

{

    "eventType": "debit",

    "info": {

        "token": "69c594cfdeeaedd220",

        "amount": "20"

    }

}

<root>

  <eventType>debit</eventType>

  <info>

    <token>69c594cfdeeaedd220</token>

    <amount>20</amount>

  </info>

</root>

1. **Functional integration of the two microservices with the previous Kafka topics**
2. **Functional integration of the two microservices with the previous Kafka topics**
3. **Implementation of the two microservices**
4. **Functional testing**