SYNCHRONICITY

Atomic Services

Jose Gato <jose.gato@atos.net>

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No732240

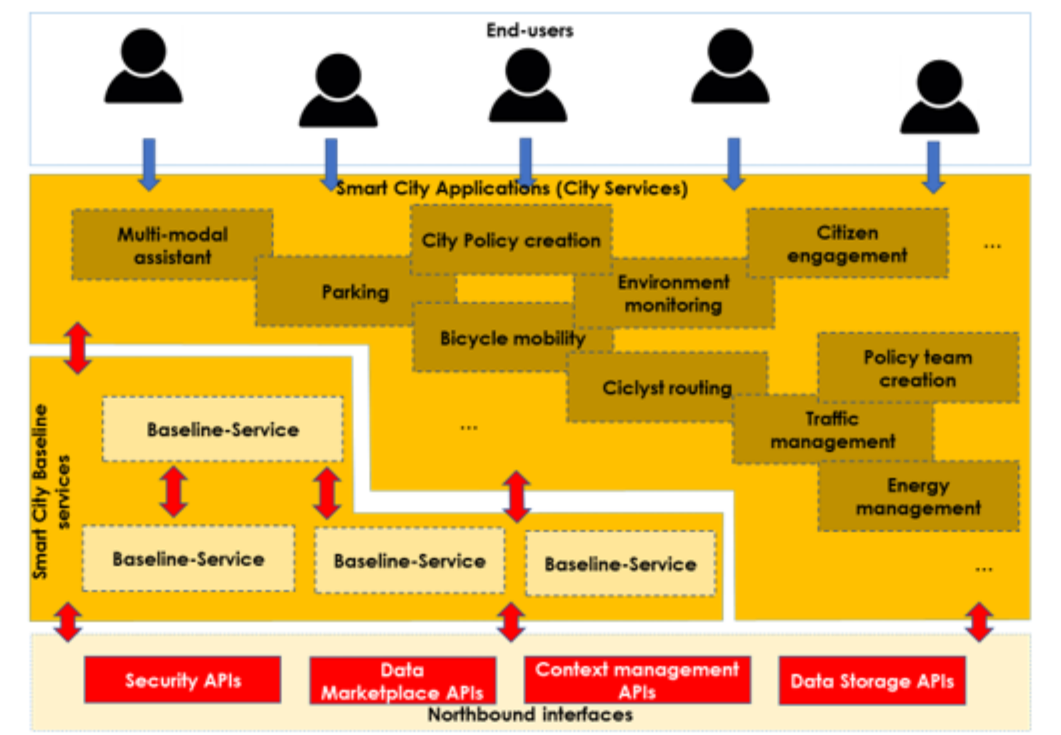
Co-funded by

Switzerland South Korea Mexico

> **The Atomic Service** provides a very specific functionality on top of the Synchronicity Framework integration points (NGSI)

> **The Atomic Service** delivers a more concrete functionality than the raw data provided by the Context or Historical Data APIs (NGSI) that the Synchronicity framework exposes

> **The Atomic Service** is placed above the SynchroniCity framework, and could be straightforwardly instantiated by any other service, orchestrator, application and/or end-user.


End-users
Smart City Applications (City Services)
Multi-modal assistant
Parking
City Policy creation
Environment monitoring
Citizen engagement
...
Bicycle mobility
Ciclyst routing
Policy team creation
Traffic management
Energy management
...
...
Smart City Baseline services
Baseline-Service
Baseline-Service
Baseline-Service
Baseline-Service
Security APIs
Data Marketplace APIs
Context management APIs
Data Storage APIs
Northbound interfaces

> **The Atomic Service** is a good opportunity to test the SynchroniCity framework and OASC principles. It could be easily replicated, accelerating new developments, in many cities that provide and implement these principles.

Based on the application theme’s needs

Human Centric
Traffic Management

Multi-Modal
Transportation

Community
Policy Suite

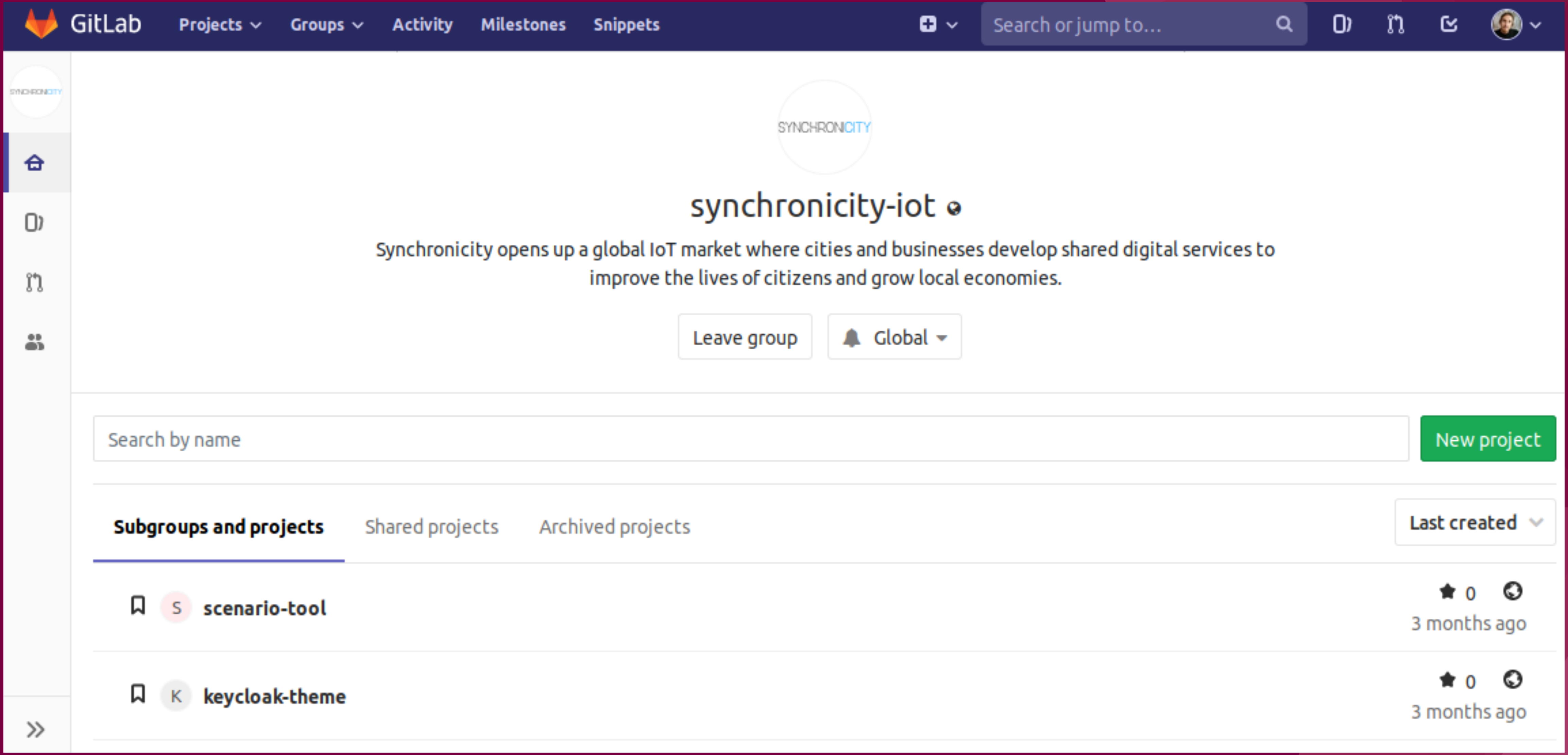
Atomic Services Basics

City Agnostic

Replicable

Easy Deployment

1) Go to the software repository (i.e. GitLab) to get the Atomic Service

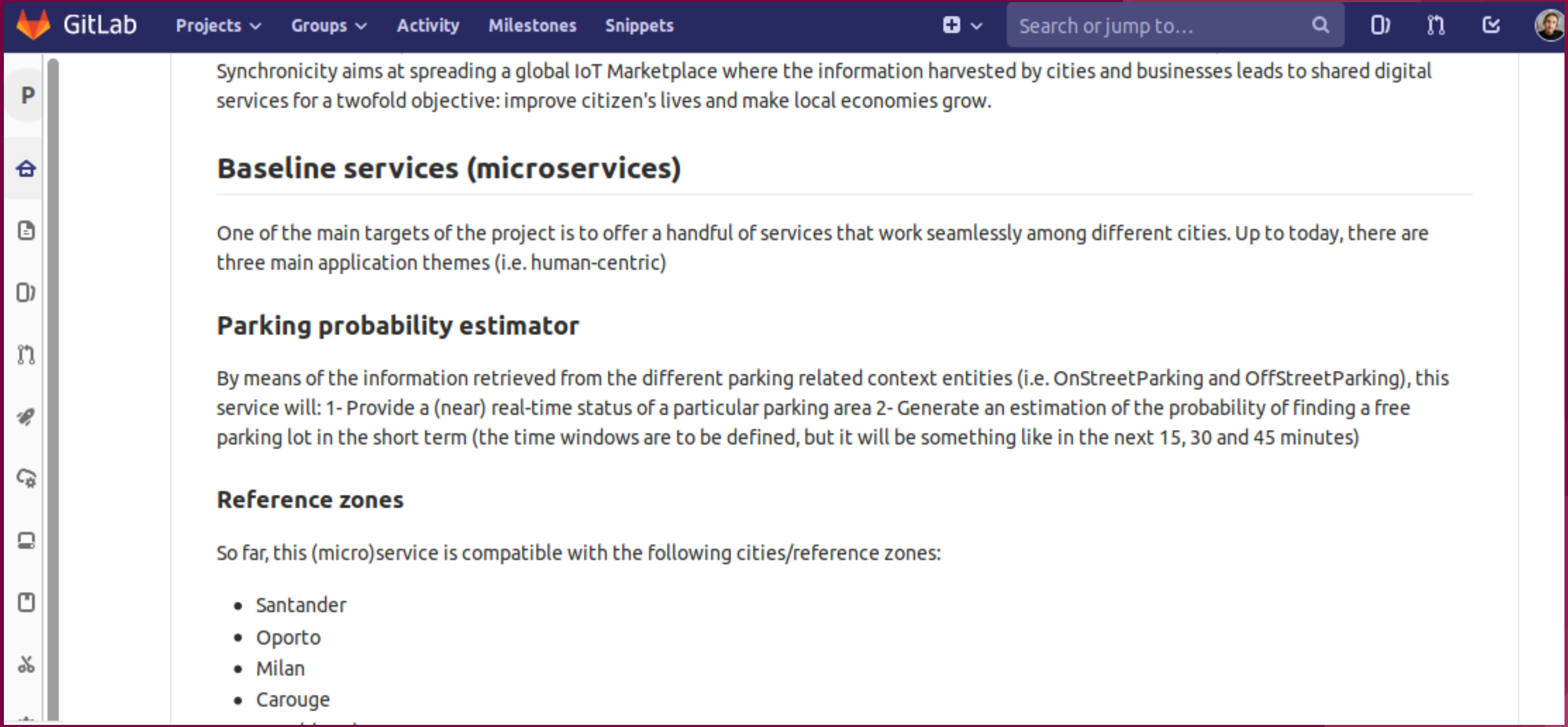


synchronicity-iot

Synchronicity opens up a global IoT market where cities and businesses develop shared digital services to improve the lives of citizens and grow local economies.

S	scenario-tool	0 3 months ago
K	keycloak-theme	0 3 months ago

2) Choose your service to get documentation, configuration, installation, contact with developers, etc. Parking Estimator for this demo



Synchronicity aims at spreading a global IoT Marketplace where the information harvested by cities and businesses leads to shared digital services for a twofold objective: improve citizen's lives and make local economies grow.

Baseline services (microservices)

One of the main targets of the project is to offer a handful of services that work seamlessly among different cities. Up to today, there are three main application themes (i.e. human-centric)

Parking probability estimator

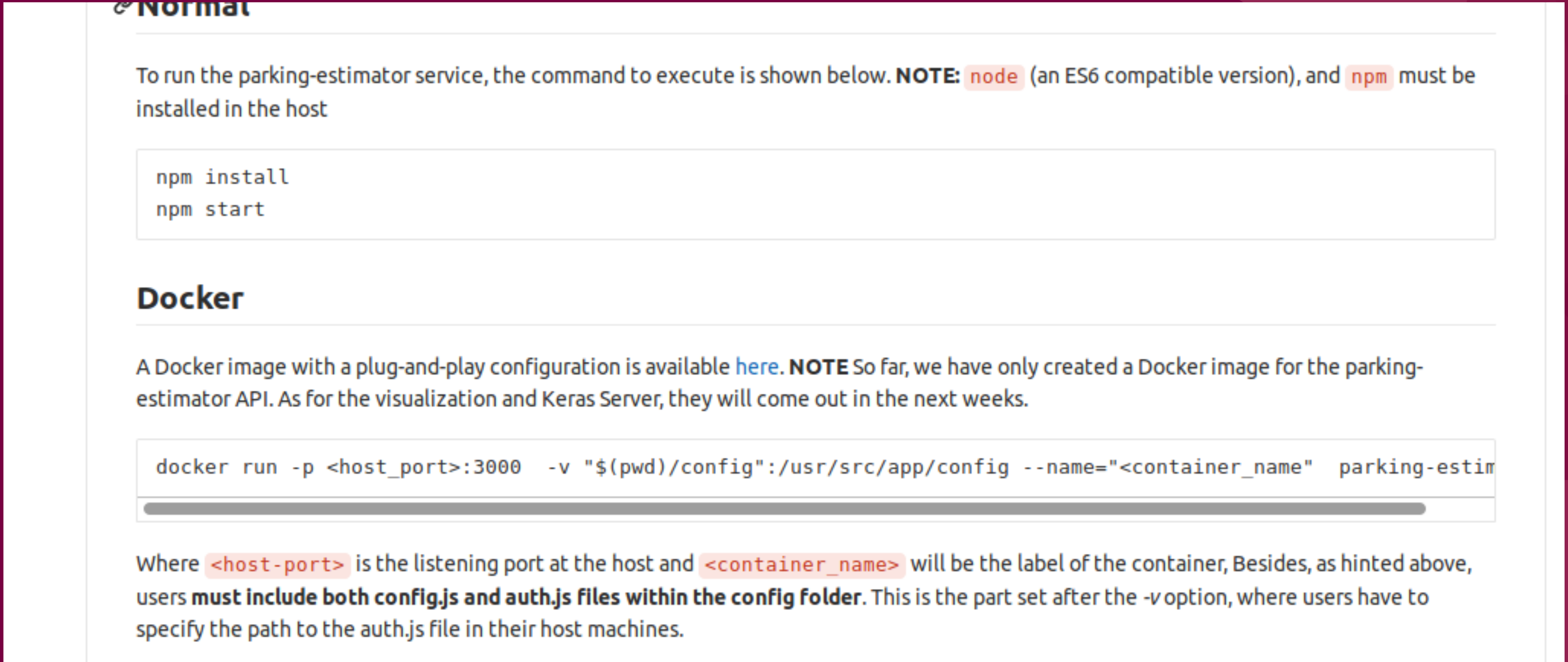
By means of the information retrieved from the different parking related context entities (i.e. OnStreetParking and OffStreetParking), this service will: 1- Provide a (near) real-time status of a particular parking area 2- Generate an estimation of the probability of finding a free parking lot in the short term (the time windows are to be defined, but it will be something like in the next 15, 30 and 45 minutes)

Reference zones

So far, this (micro)service is compatible with the following cities/reference zones:

- Santander
- Oporto
- Milan
- Carouge

3) Installation and deployment instructions: every Atomic Service is dockerized for easy installation, deployment and replication.



Normal

To run the parking-estimator service, the command to execute is shown below. **NOTE:** `node` (an ES6 compatible version), and `npm` must be installed in the host

```
npm install
npm start
```

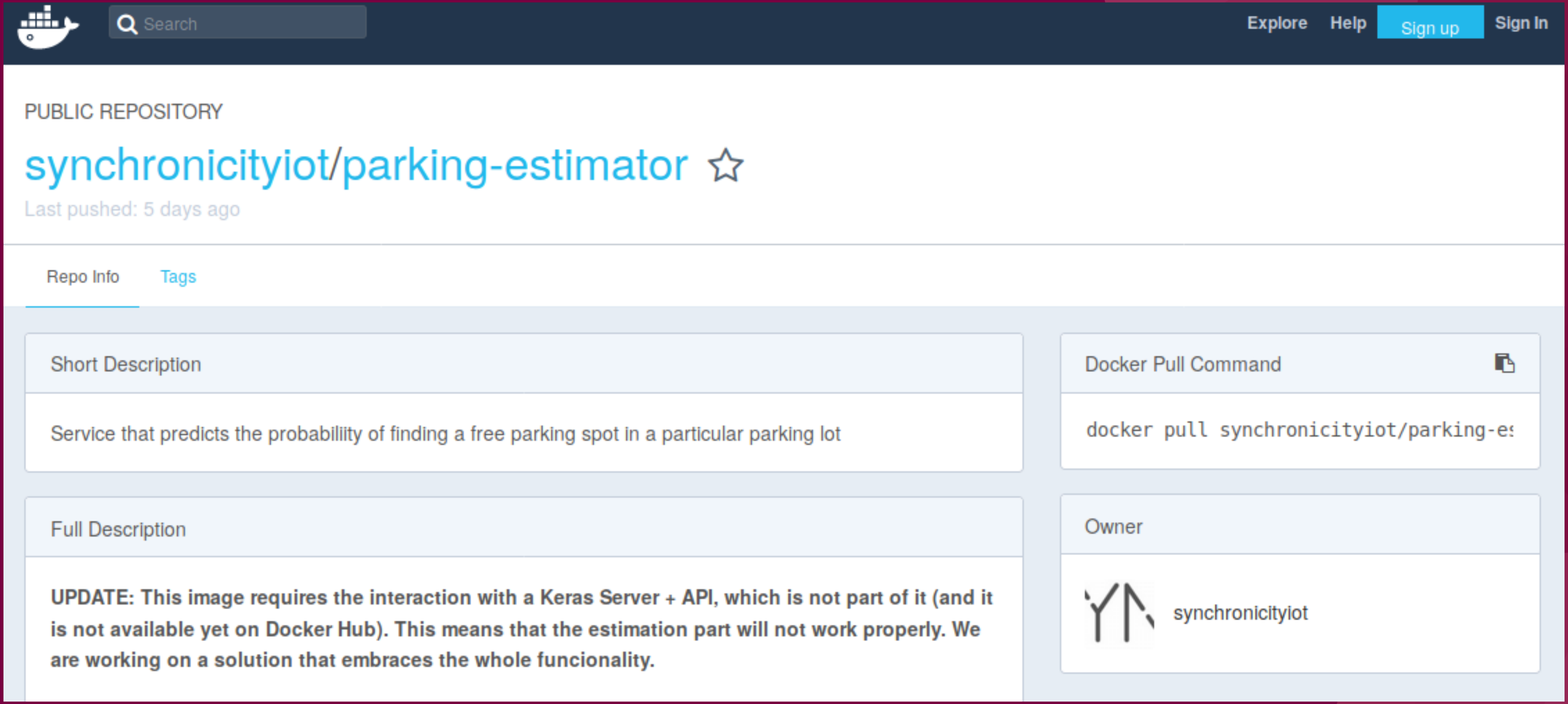
Docker

A Docker image with a plug-and-play configuration is available here. **NOTE** So far, we have only created a Docker image for the parking-estimator API. As for the visualization and Keras Server, they will come out in the next weeks.

```
docker run -p <host_port>:3000  -v "$(pwd)/config":/usr/src/app/config --name="<container_name"  parking-estim
```

Where `<host-port>` is the listening port at the host and `<container_name>` will be the label of the container, Besides, as hinted above, users **must include both config.js and auth.js files within the config folder**. This is the part set after the *-v* option, where users have to specify the path to the auth.js file in their host machines.

4) Dockerhub repository for the project



PUBLIC REPOSITORY

synchronicityiot/parking-estimator ☆

Last pushed: 5 days ago

Short Description

Service that predicts the probabiliity of finding a free parking spot in a particular parking lot

Docker Pull Command

```
docker pull synchronicityiot/parking-es
```

Full Description

UPDATE: This image requires the interaction with a Keras Server + API, which is not part of it (and it is not available yet on Docker Hub). This means that the estimation part will not work properly. We are working on a solution that embraces the whole funcionality.

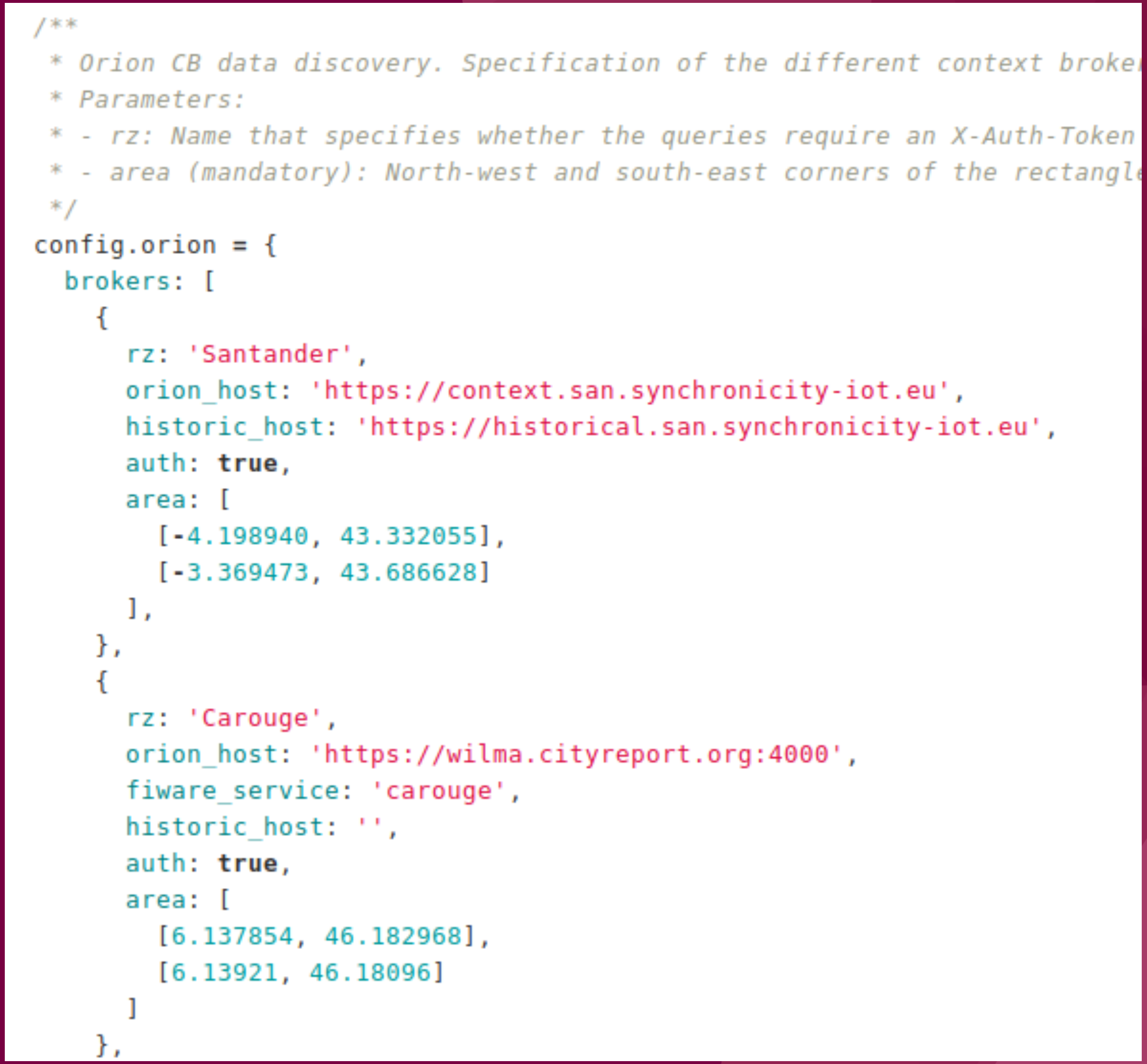
Owner

synchronicityiot

5) One command to deploy. Then configure it against SC Framework.

6) **Integration points with SC:**

- Authorization Tokens
- Context Management API
- Historical Data API

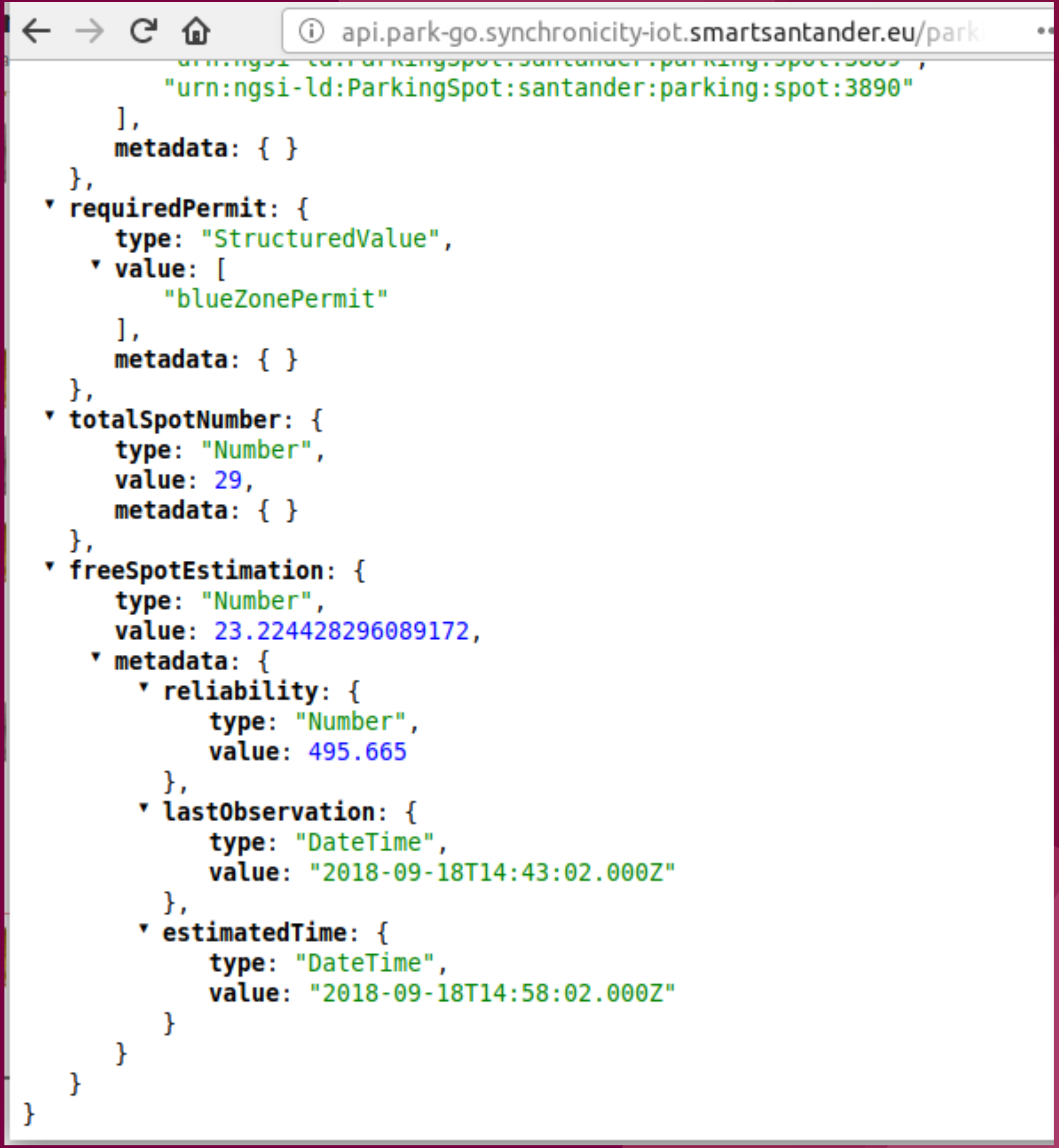


```
/**
 * Orion CB data discovery. Specification of the different context broke
 * Parameters:
 * - rz: Name that specifies whether the queries require an X-Auth-Token
 * - area (mandatory): North-west and south-east corners of the rectangle
 */
config.orion = {
  brokers: [
    {
      rz: 'Santander',
      orion_host: 'https://context.san.synchronicity-iot.eu',
      historic_host: 'https://historical.san.synchronicity-iot.eu',
      auth: true,
      area: [
        [-4.198940, 43.332055],
        [-3.369473, 43.686628]
      ],
    },
    {
      rz: 'Carouge',
      orion_host: 'https://wilma.cityreport.org:4000',
      fiware_service: 'carouge',
      historic_host: '',
      auth: true,
      area: [
        [6.137854, 46.182968],
        [6.13921, 46.18096]
      ]
    },
```

6) Try the Atomic Service via API:

http://api.park-go.synchronicity-iot.smartsantander.eu/

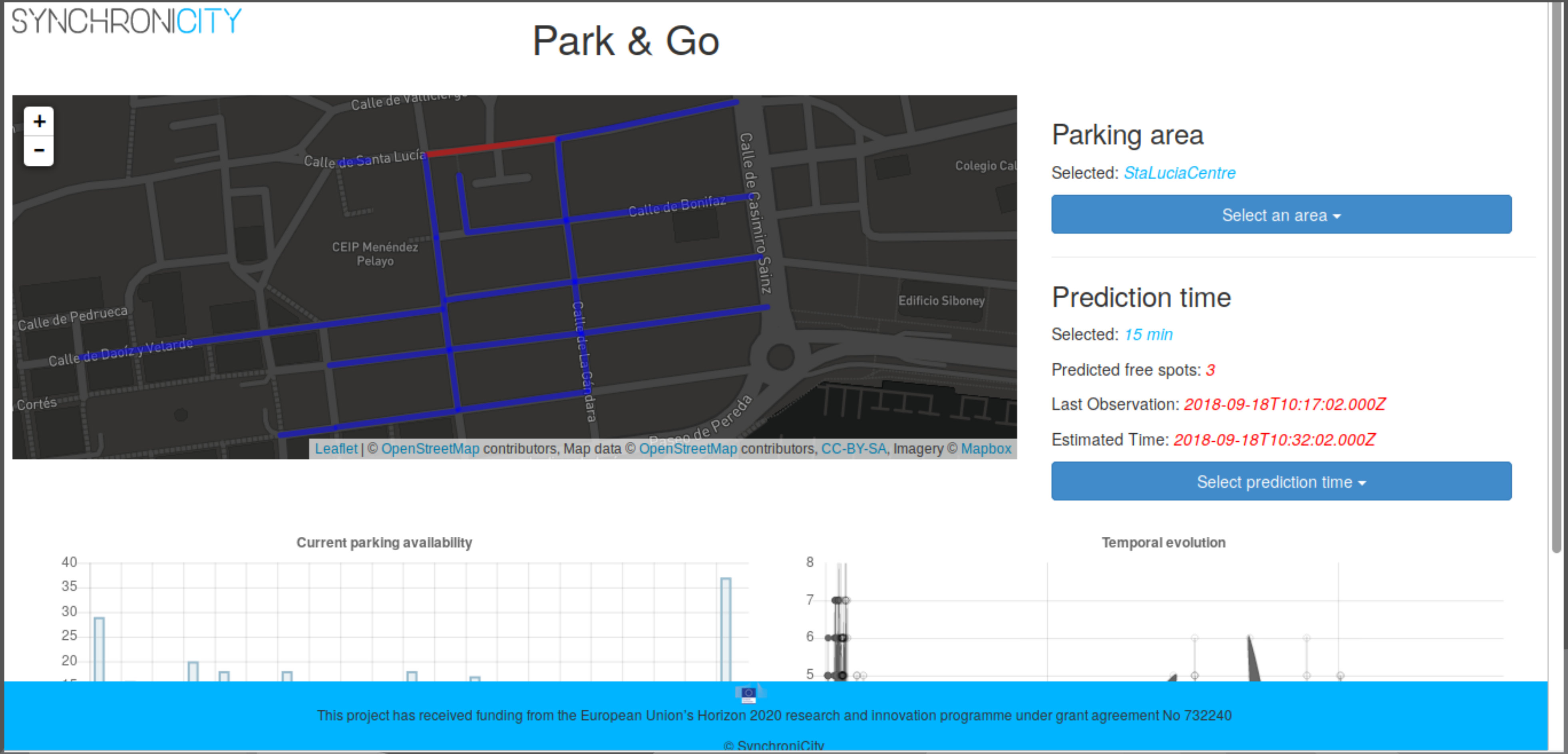
Request 1:
http://api.park-go.synchronicity-iot.smartsantander.eu/parkingareas/urn:ngsi-ld:OnStreetParking:santander:parking:onStreet:StaLuciaEast/freeSpotEstimation?expectedArrivalTime=15



api.park-go.synchronicity-iot.smartsantander.eu/park

```
        "urn:ngsi-ld:ParkingSpot:santander:parking:spot:3889",
        "urn:ngsi-ld:ParkingSpot:santander:parking:spot:3890"
      ],
      metadata: { }
    },
  requiredPermit: {
      type: "StructuredValue",
    value: [
        "blueZonePermit"
      ],
      metadata: { }
    },
  totalSpotNumber: {
      type: "Number",
      value: 29,
      metadata: { }
    },
  freeSpotEstimation: {
      type: "Number",
      value: 23.224428296089172,
    metadata: {
      reliability: {
          type: "Number",
          value: 495.665
        },
      lastObservation: {
          type: "DateTime",
          value: "2018-09-18T14:43:02.000Z"
        },
      estimatedTime: {
          type: "DateTime",
          value: "2018-09-18T14:58:02.000Z"
        }
      }
    }
}
```

7) Build your own app based on one (or several) Atomic Services



SYNCHRONICITY

Park & Go

Leaflet | © OpenStreetMap contributors, Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

Parking area

Selected: *StaLuciaCentre*

Select an area ▾

Prediction time

Selected: *15 min*

Predicted free spots: *3*

Last Observation: *2018-09-18T10:17:02.000Z*

Estimated Time: *2018-09-18T10:32:02.000Z*

Select prediction time ▾

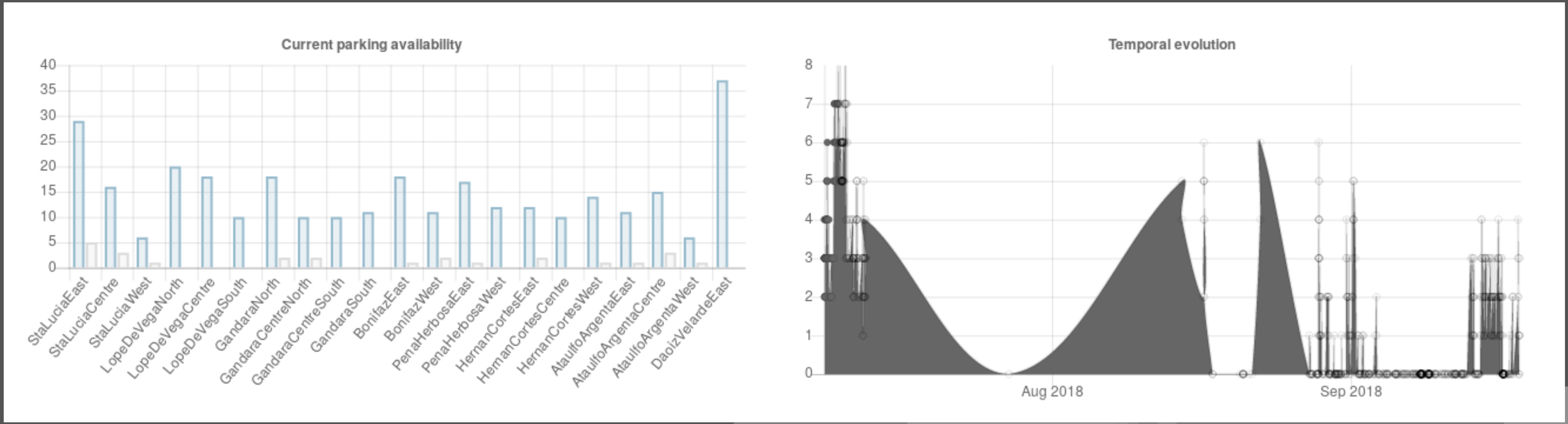
Current parking availability

Temporal evolution

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732240

8) Integration points with SC:

- SC Framework: Historical Data and Context Management API, Authorization tokens
- Atomic Service: Parking Estimator API



More Atomic Services

- **Traffic Flow Estimator**. Based on the same technology than Parking Estimator. Using NGSI TrafficFlowObserved data model.
- **Metrics Visualizer** (based on Graphana). Well known opensource project for graphics visualization. Integrated to support NGSI ParkingSpot, WeatherObserved/AirQualityObserved
- **Smartcities Dashboard**. Map visualization tool.
- **NGSI to GTFS-RT**. Specific adaptor to translate from an NGSI source to the GTFS-RT format.
- **NGSI to GTFS static data**. Translator from an NGSI source to GTFS format for time tables (e.g. at bus stops).
- **Routing Service, OpenTripPlanner (OTP)**. Well known opensource project for route planning. Orchestrated to work easily with NGSI to GTFS and GTFS-RT adaptors

More Atomic Services

Traffic Flow Estimator

TrafficFlowObserved

Metrics Visualizer (Graphana)

ParkingSpot
WeatherObserved
AirQualityObserved

Smartcities Dashboard

Geolocated entities

NGSI
data models

More Atomic Services

NGSI to GTFS-RT (real-time)

DetachedBusArraivalEstimation
GTFSTransitFeedFile

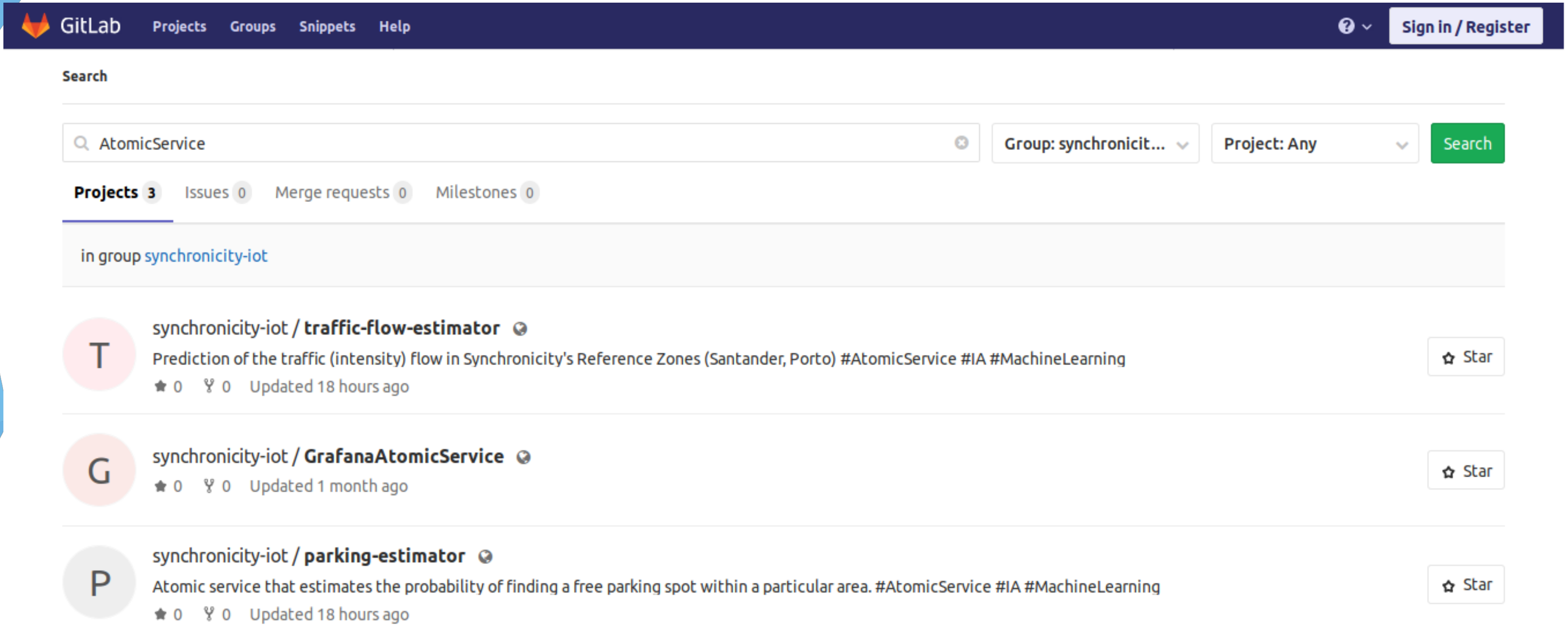
NGSI to GTFS (static)

UrbanMobility
GTFSTransitFeedFile

Routing Service (OTP)

NGSI
data models

More Atomic Services to come…



Search

AtomicService

Group: synchronicit…

Project: Any

Search

in group synchronicity-iot

synchronicity-iot / **traffic-flow-estimator**

Prediction of the traffic (intensity) flow in Synchronicity's Reference Zones (Santander, Porto) #AtomicService #IA #MachineLearning

★ 0 0 Updated 18 hours ago

synchronicity-iot / **GrafanaAtomicService**

★ 0 0 Updated 1 month ago

synchronicity-iot / **parking-estimator**

Atomic service that estimates the probability of finding a free parking spot within a particular area. #AtomicService #IA #MachineLearning

★ 0 0 Updated 18 hours ago

SYNCHRONICITY

Visit our website
synchronicity-iot.eu

Follow us on Twitter
@SyncCityIoT

Follow us on Facebook
@SynchroniCityiot

General information
info@synchronicity-iot.eu

Open Call enquiries
helpdesk@synchronicity-iot.eu