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.
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.
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.

```bash
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.

```bash
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 probability of finding a free parking spot in a particular parking lot

**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 functionality.

**Docker Pull Command**

docker pull synchronicityiot/parking-estimator

**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
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
7) Build your own app based on one (or several) Atomic Services
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
- Metrics Visualizer (Graphana)
- Smartcities Dashboard

NGSI data models

- TrafficFlowObserved
- ParkingSpot
- WeatherObserved
- AirQualityObserved
- Geolocated entities
More Atomic Services

- NGSI to GTFS-RT (real-time)
  - DetachedBusArrivalEstimation
  - GTFSTransitFeedFile

- NGSI to GTFS (static)
  - UrbanMobility
  - GTFSTransitFeedFile

- Routing Service (OTP)
More Atomic Services to come…
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