Snap4City Platform

Technical Overview

From: DINFO dept of University of Florence, with its
DISIT Lab, Https://www.disit.org with its Snap4City solution

Snap4City**:**

- Web page**:** Https://www.snap4city.org
- https://twitter.com/snap4city
- https://www.facebook.com/snap4city

Contact Person: Paolo Nesi, Paolo.nesi@unifi.it

- o   Phone: +39-335-5668674
- o   Linkedin: https://www.linkedin.com/in/paolo-nesi-849ba51/
- o   Twitter: https://twitter.com/paolonesi
- o   FaceBook: https://www.facebook.com/paolo.nesi2

Access Level: Public.

Date: 01-11-2021

Version: 5.7

Table of Contents

1 - Executive Summary

DISIT Lab is providing Smart City / Industry 4.0 Platform, which covers all your needs, in a suitable form to meet your needs 100% and much more. We are working in this domain since several years providing several modules and solutions with a multitude of partners for creating an integrated platform that we are proposing to you in this document, freely accessible on Https://www.snap4city.org.

DISIT Lab Snap4City platform has a long list of references and successful installations reported in this document, so as to provide you with the evidence that we can solve your problems. In all cases and challenges, we have deployed solutions covering more than what you need now. But, what is non urgent for you now, is or was urgent for other cases, and may be relevant for you in the next future. We work on 100% open-source basis, and according to a continuous innovation and continuous development approach, which means every new development is performed on the platform and made accessible for the whole community and in open source, for all customers and users. This has allowed us to provide for many years a large amount of features for all customers and in several domains, by sharing the Snap4 technology and platform, passing also from Smart City, to Industry 4.0, mobility and transport, environment, security, etc. Snap4City is open and at the same time secure encrypted, scalable, modular and flexible, managed by DISIT Lab of University of Florence, in collaboration with other institutions and companies who have accepted to work on open-source model; it can be used to set up Living Lab and smart city solutions, thus satisfying a large range of requirements by city officers, citizens, and tourists, as well as developers, companies and researchers.

DISIT Lab with Snap4City supports the city in its process of continuous innovation on services and infrastructures, through control and supervision, tools for business intelligence, predictions, anomaly detection, early warning, risk assessment, what-if analysis, and also setting up strategies for increasing city resilience with respect to unexpected unknowns. Indeed, Snap4City is actually capable to keep the real time evolution of your plant and city under control, by reading sensors, services, computing and controlling key performance indicators, detecting critical and unexpected evolutions, performing analytics, performing simulation, computing real time what-if analysis, taking actions from strategies and alarms, thus supporting the whole decision-making process.

DISIT Lab is well known at the international level on Smart Cities for its implementations as: Methodology for Smart Cities and Living Lab setup, Smart City Control Rooms, Smart algorithms and data analytics, Decision support systems. DISIT lab has coordinated the writing of the ICT part of the strategic plan related to Florence Metropolitan area. DISIT lab has a dedicated datacenter for bigdata solutions on which https://www.snap4city.org and Twitter Vigilance are operating, providing services for a large number of cities and regions: Firenze (I), Helsinki (Fi), Antwerp (B), Lonato del Garda (I), Valencia (S), Pont Du Gard (Fr), WestGreece (Gr), Mostar (Bosnia), Dubrovnik (Croatia), Santiago di Compostela (S), Pisa (I), Prato (I), Pistoia (I), Lucca (I), Arezzo (I), Grosseto (I), Livorno (I), Siena (I), Massa (I), Modena (I), Cagliari (I), Venezia (I), Roma (I), etc.; and as far as regions are concerned: Tuscany (I), Garda Lake (I), Sweden (Svealand Region), Sardegna (I), Belgium (B), Finland (Fl), Emilia Romagna (I), Spain (S), WestGreece (Gr), Occitanie (Fr), etc.

Today, the main challenges cities have to face are multidomain services and infrastructures such as mobility and transport, people flows, energy, government, welfare, tourism, culture, environment, weather, education; which are capable of exerting a mutual influence on one another and, in critical conditions, there is bound to be a ripple effect, causing further problems. Snap4City improves city services, security and safety by offering a sustainable solution for Smart City and Living Lab; such a solution fulfils a large range of requirements by city officers, citizens, and tourists, contributing to strengthening the city capacities to be of great appeal to industries and stakeholders.

Snap4City is a sustainable data/service ecosystem where cities and stakeholders can exploit resources to set up valuable services, free from the vendor lock-in problems. And, when cities are ready, the support to set up the Living Lab or co-working environment, is gradually provided, in order to create a collaborative context among

several cities and international experts, where cities' users and stakeholders are involved in order to profitably exploit, produce and share data and services.

This current document is organized as follows:

- **Section 2** describes the Mayor references of DISIT Lab and its solutions, covered scenarios, direct contracts in places, and international projects, prizes and recognitions awarded by the platform;
  - o Including the installations on premise and the size of the installation on cloud and on premise.
  - o Providing a list of accessible scenarios and use cases.
  - o Providing a list of references, and awards we have received
- **Section 3** describes the main concepts of the Snap4City Platform.
  - o In addition, we provide an overview of architecture and innovation methodology, while answering to many specific questions:
    - How to start up the smart city Living Lab (which is substantially what you are planning to have as the environment where your stakeholders can work together for the smart city);
    - how to collect data sources, also via IoT Networks, and how they can be integrated;
    - how it becomes possible to manage data in an integrated manner, to avoid pillars?
    - Which kind of data model and digital twin approach is valid for a smart city?
    - How can I manage the smart city infrastructure taking into account data model, processes, users, applications in a single tool?
    - What the smart city API can do for you and your city?
    - How can city users and visitor get engaged in providing data and using such services?
    - How to exploit data, for mobile app and in order to create services?
    - How to develop applications for the integration of your processes?
    - Which kind of interoperability may you have with Snap4City and the rest of your solutions?
    - How can you federate your smart city with others around?
    - How can you develop Data Analytics tools?
    - What is the situation like in other cities?
    - Would it be easy and powerful for all developers?
    - Which kind of Dashboards and tools may you have, or may you build in short time?
    - What is the development environment to be like? How easy would it be to work with it?
    - How flexible and scalable is the platform?
    - How far the user interface could be customized?
    - Which is the level of security?
    - Is the platform capable to support privacy issues?
    - Is it possible via the platform to manage maintenance of city assets in an easy way?
    - Could it provide tools for a deep management of the platform?
    - Which kind of tools are provided to both developers and integrators, in order to control the smart city platform once deployed?
    - Which kind of automatization degree is available into the smart city platform?
    - etc.
- **Section 4:** List of Terms and Acronyms that may be of use to become familiar with the Snap4City terminology
- **Section 5** a list of additional references, which could be needed to go in deep. Yet, in most cases, within the document we have reported directly the link to make the reading easier and allow the experts to jump directly to additional sources of information.

2 – Snap4City References of DISIT lab

Snap4City improves city services, security and safety by offering a sustainable solution for smart city and Living Lab, thus attracting industries and stakeholders. It can be used for implementing large solutions as well as verticals for smart parking, bike sharing, people flow, industry control, etc. Snap4City is capable to keep under control the real time city evolution: reading sensors; computing and controlling key performance indicators, KPI; detecting unexpected evolutions (early warnings); performing analytics; taking actions on strategies and alarms; providing interfaces to act upon the city by means of control dashboards; performing what-if analysis. Snap4City supports the city in the process of continuous innovation on services, infrastructures, with control and supervision, tools for business intelligence, predictions, anomaly detection, early warning, risk assessment, what-if analysis, also by setting up strategies for increasing city resilience with respect to unexpected or unknowns. Thanks to Knowledge Base support, Snap4City provides flexible solutions to get immediate insights and deductions of the city status and evolution, exploiting ultimate artificial intelligence, semantic computing, data analytics and big data technologies, activating sentient solutions collecting, and exploiting heterogeneous data of any kind, from any data source (open and private; static, real time, event driven, streams, certified and personal).

Snap4City is 100% open source, secure encrypted, scalable, modular and flexible, produced by DISIT Lab of University of Florence, in collaboration with institutions and companies that have accepted to contribute in the full respect of the open source model.

*Snap4City it is an official Platform of FIWARE, a platform of EOSC, a platform of Node-RED, a platform of EO15, winner of Select4Cities PCP Award, winner of the ENEL-X **Challenge on** Open Data. Snap4City is an official solution of FIWARE, but it supports other protocols and brokers as well; **it is a certified to offer** FIWARE **Training Services, and** FIWARE **Consultancy Services, and provide certified FIWARE Experts.***

Snap4City can be used to set up Living Lab and smart city solutions satisfying a large range of requirements by city officers, citizens, and tourists or developers, decision makers, companies and researchers. It can be easily integrated through in place solutions to provide out of data, factors and causes, predictions, anomalies, optimizing resources and understanding social contexts and meanings of facts. It can be installed on cloud and on premise. This capability is obtained by enforcing the new functionalities of Snap4City tools/modules on your premise and/or exploiting them from cloud.

Snap4City is based on **Km4City Ontology** for the **Knowledge Base** and Smart City API, a suite of open-source technologies for developing smart cities, big data aggregators, and analytics adopted in: Ampere, Enterprise, SmartAmbulance, Herit-Data Interreg, MOBIMART Interreg, TRAFAIR CEF, REPLICATE EC H2020, RESOLUTE EC H2020, Sii-Mobility SCN MIUR, WEEE Life, Green Impact, 5G with Wind 3G and ESTRA, MOSAIC, SmartBed, SODA, and others. Snap4City added to Km4City the flexibility to set up Living Labs and co-working environments for IOT/IOE (internet of things/everything) in smart cities and industry 4.0; to cover a large range of scenarios and a large range of IoT Devices, Brokers, IoT Edge, Gateways, mobiles, OBU (on board unit, vehicular kit for bikes, cars and busses), social media and satellite data, and it is open to all standards and custom models. Some of these devices are also produced by Snap4City as Open hardware and Open Software.

Snap4City has been ranked as **1st place Award in the Select4Cities challenges and PCP** (https://www.select4cities.eu/). Select4Cities proposed a number of functional and non-functional requirements and concepts, extending those of **ENOLL Living Lab Association**. See the 1st place Award Snap4City ceremony held at Smart City Expo World Congress for Select4Cities competition. https://www.snap4city.org/558. All the expected requirements and much more have been satisfied by the Snap4City solution which is fully open source, supports cloud and scalability for processing and IOT/IOE. It respects user needs and privacy according to GDPR and to the different kinds of users, provides tools and community for co-creation; mixt data driven, stream and batch processing; it is fully modular and based on microservices, while exploiting easily replaceable tools.

Snap4City solution has been designed to be scalable, flexible, safe and respectful of privacy, endowed of a powerful semantic reasoner based on Km4City multi-domain semantic model and tools (https://www.km4city.org). A special attention has been given to enable the development of applications in multiple domains and not only on mobility and transport, environment, tourism, health & welfare, social, Industry 4.0, etc. The proposed solution is flexible enough to support extensions at different levels of granularity: data, analytics, tools and applications.

> Katrien Lenaert of Digipolis Antwerp stated: "*1st place has been awarded to Florence (snap4city)........the competition was really hard.... frequent changes from our parts..*", "*.. about Florence (snap4city) strong points as always, excellent analytics, excellent data aggregation, very strong point was the utilities on which to work and also the fact that you (snap4city) turn the criticisms in opportunities to improve the tool that made your platform better all the times. It is something that we really appreciated, ... 100% open source, ... and the fact that you (snap4city) commercialize the solution already, ... very good feature... thanks again*".
>
> Hugo Goncalves, Project Manager, Forum Virium Helsinki, said: "*We have seen outstanding solution development in this PCP. Entrance to Phase II was tough and the successful contractors all impressed the Buyers Group by going above and beyond expectations, showing that their solutions not only meet complex technical standards and requirements but also a variety of end-user needs. In this first year of the competition, the innovation levels are very strong, and all of our contractors should be immensely proud of what they have achieved so far*."

Snap4City is operative with **services and data of several cities/Organizations** such as: Firenze (I), Helsinki (Fi), Antwerp (B), Lonato del Garda (I), Valencia (S), Pont Du Gard (Fr), WestGreece (Gr), Mostar (Bosnia), Dubrovnik (Croatia), Santiago di Compostela (S), Pisa (I), Prato (I), Pistoia (I), Lucca (I), Arezzo (I), Grosseto (I), Livorno (I), Siena (I), Massa (I), Modena (I), Cagliari (I), Venezia (I), Roma (I), etc.; and from regions as Tuscany (I), Garda Lake (I), Sweden (Svealand Region), Sardegna (I), Belgium (B), Finland (Fl), Emilia Romagna (I), Spain (S), WestGreece (Gr), Occitanie (Fr), Bosnia-Herzegovina, Croatia, etc. https://www.snap4city.org/download/video/cov/ but also in Israel, Cyprus for the smart city strategies, etc.

UNIVERSITÀ DEGLI STUDI FIRENZE DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB
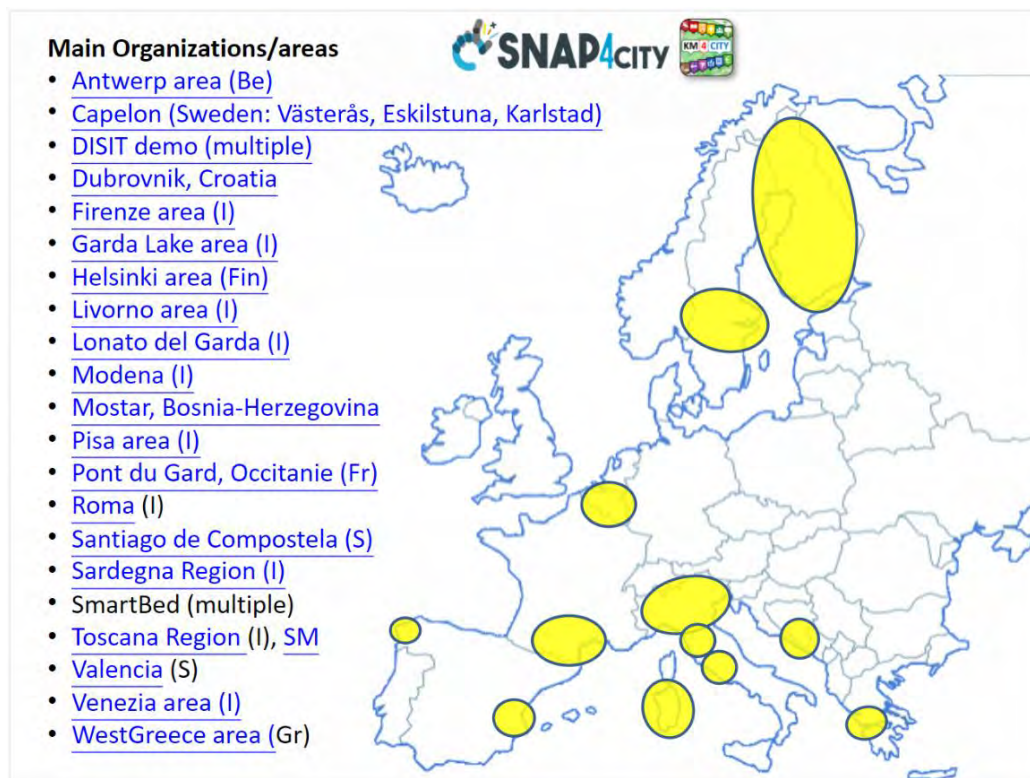
SNAP4CITY

Most of these Organizations keep data and dashboards as private data, this means that even if you can register on them, the access to those resources and Dashboard is decided by the owners, and thus you may not see the Snap4City functionalities in place. For this reasons we suggest to register in Snap4City by selecting DISIT Organization and may be with a other accounts in any other ORG you prefer.

Please note that Snap4City is also adopted in a number of Universities as a tool for Engineers: **University of Florence, University of Milano, University of Verona, University of Cagliari, etc**. and therefore we perform a continuous training.

> *Please refer to Section 4 for the list of acronyms, definitions, and references in section 5.*
>
> *We tried to report links in the text as much as possible so that you can jump to the web page to read more, to see definitions, etc.*

2.1 – Installations on premises and volumes of Snap4City

Snap4City is a platform **adopted in many operative solutions and projects. For this reason, we have a number of Operative in Production installations and a number of ongoing Pilots that use Snap4City Platform as better explained in the following. With the following meaning:**

- **Operative in Production, they are ACTIVE solutions, most of them have also public services and public dashboards. Some of them have only private dashboards and public services.**
  - o **Please note that most of them are federated each other as described in the following**
- **Started in Pilot** for solutions that are performing data ingestion and which going to become visible and in production solutions soon, such as: Dubrovnik, Valencia, etc.
- **Testing training as PoC,** proof of Concept, for understanding if the solution is satisfactory. *Most of the PoC which are presently active have not been reported in detail in this document such as those of: Rome, Prato, etc., and other that are covered by some Non Disclosure Agreement, NDA.*

Snap4City has been designed to support multiple domains, data models, brokers, formats and protocols. This is evident from web page https://www.snap4city.org/4 where a large number of scenarios are reported. They are also listed in **Section 2.2** of the technical proposal document.

In web page: https://www.snap4city.org/661 Snap4City installations are described and reported below. They are only those requested to be listed. There are also other installations not registered to this list, since Snap4City is fully Open Source so that everybody can download and install without informing us that the platform has been used.

Name	URL	Operator	URL of Operator	purpose	location	Shape	Size	KB federated	Living Lab
Snap4City.org	Https://www.snap4city.org	DISIT	Https://www.disit.org	Smart City, Industry 4.0, any domain, training, development	DISIT UNIFI	Snap4City Industry, all	Huge	Yes	Yes
Snap4Altair, SODA	non accessible from Internet	ALTAIR	https://www.altairchimica.com/	managing chemical industry plant	ALTAIR	Industry	medium	no	no
Snap4Pisa, Mobimart	https://snap4city.aedit.it	AEDIT	http://www.aedit.it/	mobility and transport, POI	PISA	Smart City	medium	Yes	Yes
SmartEA SmartGardaLake	https://smartgardalake.snap4.eu/	Snap4	https://www.snap4.eu/	Environment, mobility, transport, POI, Culture, Tourism, etc.	Garda Lake area	Smart City	small	Yes	(yes)
Capelon	non accessible from Internet	CAPELON	--	Smart Light, smart city	Sweden	Smart City	medium	No	(yes)
--	Non accessible	UNISYSTEMS	--	Monitoring experiments	Greek	Smart City	medium	No	Yes

--	Non accessible	SYN-NG-AI	--	Smart city	Israel	Smart City	small	No	No
ITALMATIC	Non accessible	ITALMATIC	--	Industry 4.0	Toscana	Industry	Small	No	No

Please note that a number of organizations are using the Smart City as a service provided and operated by DISIT Lab directly. In the following table, there are data on size of the Organizations and their geographic coverage in terms of inhabitants and visitors. We also reported the number of users, dashboards, content / data models (as data sets), and the status: operative, started if the volume of data and activity has still to reach a minimum level of evidence (according to our approach), testing if the goal of that activity is to train and test the platform before deploying in a larger installation.

Name	Organizations	Inhabitants	Visitors	users	dashboards	Content	status
Snap4City.org	Toscana	3.5 million	>48 million	150	24	252 datasets	Operative
	Firenze	1.5 million	>14 million	40	>100	92 datasets	Operative
	Pisa	90.000	>4 million	50	10	80 datasets	Operative
	Helsinki	630.000	>1.3 million	24	>50	27 datasets	Operative
	Antwerp	498.000	1 million	25	>60	25 datasets	Operative
	GardaLake	250.000	> 30 million		3	4 datasets	Started
	Lonato DG	20.000		10	12	4 datasets	Operative
	DISIT	As Toscana		150	>90	310datasets	Operative
	Sardegna				4	20 datasets	Started
	WestGreece					10 datasets	Operative
	PontDuGard					8 datasets	Operative
	Dubrovnik				3	2 datasets	Operative
	Mostar					2 datasets	Started
	Valencia				2	2 datasets	Operative
	SmartBed				4	1 dataset	Testing training
	Capelon	400.000		10	3	3 datasets	Testing training
Snap4Altair, SODA	Altair	12		12	30	400 items	Operative
Snap4Pisa, Mobimart	Pisa	90.000	>4 million	50	10	80 datasets	Operative
SmartGardaLake	SmartGarda	250.000	>30 million	10	5	15 datasets	Operative
Capelon	Capelon			2	4	2 datasets	Testing training

Main schemas for the **most** relevant **Operative In Production** solutions:

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

SNAP4CITY KM4CITY

Cities of Firenze, Pisa, Livorno for the Tuscany Region

City of Antwerp in Belgium:

City of Lonato del Garda, Italy

Capelon, Sweden

Altair Chemical plant, Tuscany, Italy

Dubrovnik

You can access to a longer list of scenarios from the training course Part 1:
https://www.snap4city.org/download/video/course2020/gen/

The larger installations are operated by DISIT lab as Snap4city.org, even if other installations are growing as well, such as GardaLake (operated by Snap4 srl with SmartTea) and Pisa (operated by AEDIT srl, with PISAMO e and Pisa city).

Snap4city.org as described in the above table, is at present the largest installation that overcome the needs of most of the cities and industry plant since it:

- ingests about 1 million of attributes every 24 hours on HLT sensors;
- generates about 5 million of attributes every 24 hours regarding: MyKPI; Heatmaps for Pisa, Firenze, Livorno; traffic flows; GTFS; user data;
- has more than 4200 users over the different organizations;
- manages more than 17 Organizations / tenants;
- provides more than 1300 Dashboards, among them, more than 100 are public dashboards
    - As to most of them you can access to the related Dashboards from:
        https://www.snap4city.org/dashboardSmartCity/management/dashboards.php
    - more than 100 different kinds of graphic widgets;
    - more than 20 macroclasses for data, and > 520 subclasses;
- manages elastically more than 480 containers with IoT App, Data Analytics (Rstudio, Python), Web Scraping, IoT App on Edge.

Some Details of Data collected / produced in DISIT Org ONLY!

dataset	#devices	#variable	#samples per day	Kind
Air quality Pollution values	199	11	288	collected
Bike sharing	39	14	96	collected
Charging stations	450	9	96	collected
Entertainment events	35	25	96	collected
Events / ordinances	80	25	90	collected
Fuel stations	170	5	1	collected
GTFS (TPL) Stops and timelines	37700	150	1 per month	Collected, 12 GBytes per Month
Hospital Triage	51	15	96	collected
IOT Devices generic	600	3	4	collected
Irrigators status	21	7	144	collected
Mobile App data collected (a part)	2000	40	250	collected
Parking status on structures	111	9	96	collected
Parking status on the roads	17	5	96	collected
Pax counters status	22	2	144	collected
Pollination values	5	17	48	collected
Public transport real time	6	1	96	collected
Rain sensors	6	5	96	collected
RTZ gates	77	10	144	collected
School data (presences)	659	9	2	collected
Smart benches	1	10	24	collected
Smart lighting	71	16	96	collected
Traffic flow sensors	993	15	144	collected
Twitter Data specific	45000	8	1500	Collected for 6 thematic areas
Weather conditions	812	19	6	collected
weather forecast	288	84	1	collected
Wi-Fi data, monitoring people	300	2	144	collected
Smart Building	25	50	144	collected
Bike passages	10	3	144	collected
Copernicus Data	4000x4000 areas	150	Some in test	collected
Heatmap air quality	1000x1000 areas	1	6	produced for 4 locations
Heatmap air NOX	4000x4000 areas	1	48	produced for 4 locations

Heatmap temperature	1000x1000 areas	1	2	produced for 4 locations
Heatmap Humidity	1000x1000 areas	1	2	produced for 4 locations
Traffic flow reconstruction	30000-40000 segments	3	96	produced for 6 locations
Parking prediction	12	8	96	produced
Bike Rack predictions	36	2	96	produced
OD from Mobile data	4 family of devices	5	24	produced for 6 locations
Trajectories	1 family of devices	4	24	produced
Sentiment analysis	4000	6	24	produced for 6 locations

2.2 – Scenarios of Snap4City

Snap4City/Industry covers with a unified platform a large set of possible scenarios, from smart city to IOT and Industry 4.0. In fact, it often occurs that cities present industry level plants in the structure they manage, for instance: water depuration, large storage, fleet management, etc. A list of them is reported in the following, even if they are to be considered only representative of what can be performed, since the platform can define and work with any kind of model, starting from FIWARE Data model to IoT Device Models, or those imported from brokers, etc.

Smart City cases:

- Europe
    - West Greece: https://www.snap4city.org/743
    - Valencia: https://www.snap4city.org/742
    - Mostar: https://www.snap4city.org/744
    - Dubrovnik: https://www.snap4city.org/741
    - Antwerp: https://www.snap4city.org/526
    - Helsinki: https://www.snap4city.org/528
    - Pont du Gard : https://www.snap4city.org/740
- Italy:
    - Florence/Firenze: https://www.snap4city.org/747
    - Pisa: https://www.snap4city.org/746
    - Smart Garda Lake: https://www.snap4city.org/739
    - Livorno: https://www.snap4city.org/751
    - Siena: https://www.snap4city.org/759
    - Prato: https://www.snap4city.org/758
    - Pistoia: https://www.snap4city.org/761
    - Toscana region: https://www.snap4city.org/760
    - Roma: https://www.snap4city.org/753
    - Modena: https://www.snap4city.org/754
    - Bologna: https://www.snap4city.org/750
    - Lonato del Garda: https://www.snap4city.org/712

All Scenarios get more on the following links:

- Global Digital Twin for Florence by Snap4City
- 15MinCityIndex: understanding city areas by means of 13 different aspects
- **Snap4City Impact Story on** FIWARE: https://fiware-foundation.medium.com/snap4city-fiware-powered-smart-app-builder-for-sentient-cities-acfe24df49d5
  PDF version: FF_ImpactStories_Snap4City.pdf
- Scenario: Smart Light Control (CAPELON)
- Scenario: Smart Parking in Lonato del Garda (Reverberi)
- Scenario: SnapBot: Real Time Smart City services via Telegram
- Scenario: Copernicus Satellite Data
- MicroServices Suite for Smart City Applications

- [Scenario: MODBUS for Snap4Industry Snap4City Applications](#)
- [Scenario: MOBIMART Interreg: MOBilità Intelligente MARe Terra](#)
- [Scenario: City of Roma case, mobility and environmental data](#)
- [Scenario: Herit-Data video and aims](#)
- [Scenario: Control Room vs Video Wall](#)
- [Scenario: Snap4Home the case of: Alexa, Philips, Sonoff, TP-link, etc. (Italiano)](#)
- [Scenario: how to manage maintenance and accidents workflows](#)
- [Scenario: Snap4Home, how to exploit Snap4City solution on home automation](#)
- [Scenario: Energy Monitoring](#)
- [Scenario: Multipurpose User Engagement Tools](#)
- [Scenario: 5G Enabled Water Cleaning Control](#) (smart city, industry 4.0)
- [Scenario: High Level Control of Industrial Plant](#) (industry 4.0)
- [Scenario: Vehicle Monitoring via OBD2](#)
- [Data Analytic: Origin Destination Matrices, Algorithms and tools](#)
- [Data Analytic: Traffic Flow Reconstruction](#)
- [Data Analytic: in general, and the cases of Antwerp and Helsinki](#)
- [Data Analytic: Predicting Air Quality](#)
- Data Analytic: Analyzing Public Transportation Offer wrt Mobility Demand, [DORAM Tool](#)
- [Scenario: Events and Museums Monitoring in Antwerp](#)
- [Scenario: High Resolution Prediction of Environmental Data](#)
- [Scenario: Mobility and Transport Analyses in multiple cities](#)
- [Scenario: People Flow Analysis via Wi-Fi](#)
- [Scenario: Antwerp Pilot on Environmental Data](#)
- [Scenario: Helsinki Pilot on Environmental Data](#)
- [Scenario: Firenze Smart City Control Room](#)
- [Scenario: Mobile & Web App: Toscana Where What … Km4City, Toscana in a Snap](#)
- [Scenario: Helsinki Pilot on User Behaviour](#)
- [Scenario: Antwerp Pilot on User Behaviour](#)
- [Smart City Control Room Dashboards: Big Data Infrastructure, from data to decision support](#)
- [Dashboards Touch Screen Support for Situation Rooms and What-iF analysis Operators](#)
- [Data Analytics on Antwerp and Helsinki](#)
- [Decision Support Engines and Models: SmartDS, ResilienceDS](#)
- [IPCAM on Dashboard Widget, Video Cameras](#)
- [Solution: using PAX Counters, monitoring museum and events](#)
- [Solution: selecting the less pollute routing](#)
- [Data Flow Management and Visual Analytic for Big Data Smart City/IOT](#)
- [Snap4City on Social Media](#)
- [AIRQino: Compact low-cost air quality monitoring station, by CNR IBIE](#)
- [Security, Privacy and GDPR Aspects of Snap4City](#)

2.3 – Direct Contracts and Projects Exploiting Snap4City platforms

In this section a list of direct contracts and projects is available for your reference.

Direct contracts using Snap4City/Industry as reference platform:

- **SMART AMBULANCE** Fratelli Mariani, data hub on the ambulance : https://www.snap4city.org/708
- **ALMAFLUIDA**: ALMA FLUIDA activity (ITALMATIC) https://www.snap4city.org/709
- **ENEL-X**: Continuing working Open Data Challenge, 15MinCity Index indicators
- **Capelon** (Sweden) → smart light solutions, center of Sweden area: https://www.capelon.se/
- **Florence Metro City** → ICT aspects of strategic plant of Firenze Metro City

- **Green Field Peas** (soda) ALTAIR Chemical → Industry 4.0, Chemical plant: https://www.altairchimica.com/blog/progetti-di-ricerca-e-sviluppo-green-field-peas-e-soda/
- **Green Impact ALTAIR** Chimica → Industry 4.0, Chemical Plant https://www.altairchimica.com/blog/progetto-di-ricerca-e-sviluppo-green-impact-capacity/
- **ISPRA JRC EC** research center 2019 and 2020 → site management and services, for JRC ISPRA area: https://ec.europa.eu/jrc/en/about/jrc-site/ispra
- **Lonato del Garda city**, Reverberi → smart parking, environment, Lonato del Garda municipality: https://www.reverberi.it/it/application/lonato-del-garda-bs
- **MobiMart**: City of PISA, PISAMO → data aggregation, Living Lab: http://interreg-maritime.eu/web/mobimart
- **Select4Cities** PCP → smart city platform PCP, Helsinki and Antwerp: https://www.select4cities.eu/
- **SmartBed** (laid), Materassificio Montalese → smart health: https://blog.perdormire.com/smart-bed/
- **Cyprus smart city strategy, assigned winner under signature.**
- **Italferr** on Snap4, for data management and business intelligence.

Projects and pilots using Snap4City as reference platform:

- **AMPERE** Project - Furniture, art and fashion in an innovative electrochemical process with remote control and circular eco-friendly: https://www.snap4city.org/707
- **Enterprise**: internal living lab for chemical plat at the support of the internal research and analysis labs
- **5G MIUR** Wind3G→ Industry 4.0 vs SmartCity, for Prato: http://www.confservizitoscana.it/news/tecnologia-5g-a-prato-due-progetti-di-estra-nella-sperimentazione-promossa-da-wind-tre-e-open-fiber
- **GHOST MIUR** → Strategies, smart city, for Cagliari: https://sites.unica.it/ghost/home/
- **Herit Data Interreg** → tourism, culture and management: https://herit-data.interreg-med.eu/
- **MOSAIC RT** → mobility and transport: https://www.disit.org/drupal/?q=node/7174
- **PC4City Ente Cassa** → Civil protection for the City, Tuscany, prediction of slidelands
- **REPLICATE H2020** → ICT, smart City Control room for Florence, Energy, IOT: http://replicate-project.eu/
- **RESOLUTE H2020** → Resilience, ICT, Big Data, for Florence, Athen: http://www.resolute-eu.org
- **Sii-Mobility SCN MIUR** → mobility and transport, sustainability, the whole Tuscany: http://www.sii-mobility.org
- **Smart Garda Lake** → Castelnuovo del Garda municipality: https://smartgardalake.it/progetto/
- **TRAFAIR CEF** → Environment & transport, Florence, Pisa, Livorno, Modena and Santiago de Compostela: http://trafair.eu/
- **WEEE Life EC** → Smart waste, environment: https://www.lifeweee.eu/lifeWeee_en/

2.4 – Recognitions and Awards to Snap4City, DISIT Lab

The Snap4City Platform:

- Snap4City is a Certified FIWARE Platform and Solution
- DISIT lab, Snap4City has two Certified FIWARE Experts
- Snap4City is certified to offer **FIWARE Training Services**: https://marketplace.fiware.org/pages/solutions/03bccd83a0e1b0398ba7a0bf
- Snap4City is certified to offer **FIWARE Consultancy Service**s: https://marketplace.fiware.org/pages/solutions/907f5ecc63927f643dd8421b
- Winner of the Open Data challenge of **ENEL-X**: https://openinnovability.enel.com/projects/New-smart-city-solutions-enabled-by-open-data

- **winner of Select4Cities PCP** https://www.snap4city.org/558 (by Antwerp, Copenhagen and Helsinki), in September 2019;
- **official solution** of **FIWARE**: https://www.snap4city.org/467 (register, the same information is on the public slides of the training course); https://marketplace.fiware.org/pages/solutions/b8905e91973b420189cce972
- **official platform** of **EOSC** (European Open Science Cloud) marketplace of the European Commission. https://marketplace.docker-fid.grid.cyf-kr.edu.pl/services/snap4city
- with UNIFI LABGEO is a **partner of ESRI** the producer of **ArcGIS**, having a common lab since several years.
- **GDPR compliant** and enforces privacy and security for data, **Dashboards**, **IoT Devices, IoT App, personal data, data analytics** and **processes**, etc., which can be private of the user; and the user may delegate the access to them or pass the full control to other users. https://www.snap4city.org/670
- passed **PEN Test from Setek and Vulnerability Test from Thales in the 2019**, they are Mayor expert companies of the Cyber Security sector. https://www.snap4city.org/669
- Snap4City is an official library of **Node-RED**
- Snap4City is a **SigFOX** partner
- Smart City Km4City API of Snap4City are official accessible API via **E015** of Regione Lombardia
- Etc.

3 – Snap4City Platform

Snap4City platform is 100% open source, scalable, and modular with a set of tools accessible according to the user profile. Snap4City can ingest and produce data of any domain and according to any kind of data sources: GIS (ArcGIS, QGIS, etc.), city utilities (water, gas, mobility, light, water, environment, industry, etc.), legacy systems, personal data, mobile data, IoT Network and **IoT Broker**, KPI/MyKPI, industry 4.0 protocols and network, social media, telecom data, etc. All the sources are bidirectional channels since Snap4City can ingest and produce data with protocols suitable for any channel. See for interoperability Https://www.snap4city.org/65 Please note that Snap4City supports any kind of IoT Networks, communication protocols, and data formats, so that **any legacy and vendor solution**. **Data Ingested** with any data model are aggregated on the basis of **Km4City Ontology** into the so-called **Knowledge Base** (which is the **Expert System** of the city/industry) and into the **Big Data storage**. This approach eliminated the problems of data silos and pillars. Any legacy solution, as well as new applications and data may be integrated in a uniform model, establishing semantic relationships of any kind.

Ingested data are immediately usable and searchable for snap4City tool such as the **Data Inspector, Big Data Analytics**, **Dashboards builder**, **Smart City API** for the **Mobile Apps**, and via **MicroServices** for IoT Applications (**IoT Apps, Node-RED**) which are used for integrations. Data can be consumed by data processing/data-analytics, IoT App/integrations, mobile App, and **Dashboard**s, and in real time streaming, that are the so-called data driven processes, end-to-end (from the device to dashboards).

The **developers and qualified operators** on **Snap4City** platform can access to the platform tools via web (without any installations on the local computers) to develop in the **Snap4City** collaborative environment solutions and applications exploiting **Data Analytics**, **Dashboards**, and **IoT Apps (for data ingestion, adaptation, transformation; business logic, data analytic management, etc.)**. Only the **Mobile Apps** developers need of a local development Environment depending on the target mobiles devices. **Final users** can access to the Snap4City services via Web and mobile devices, dashboards, interfaces for digital signages, tables, panels, etc.
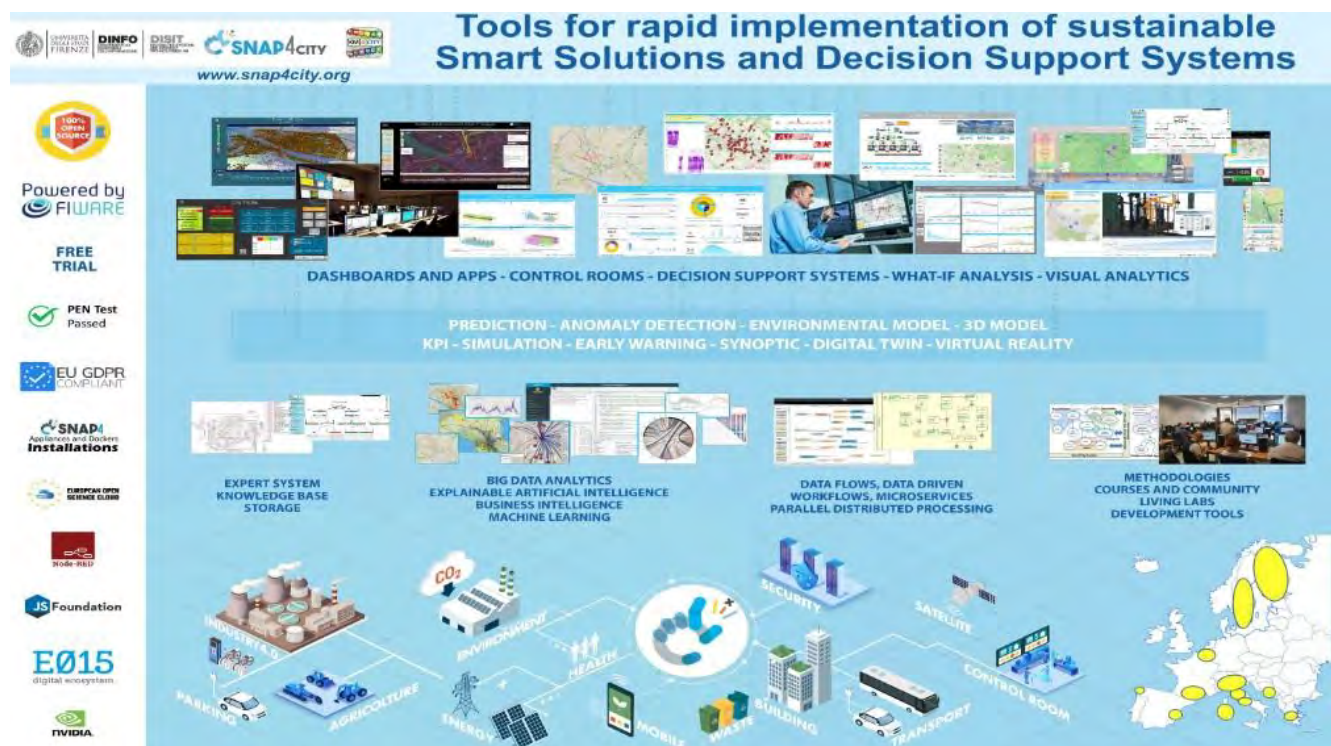
A Snap4City **Living Lab** support allows the **stakeholders** to collaborate for the production of smart solutions and to the innovation of the whole city ecosystem. On this regard, Snap4City provides a methodology for stimulating the innovation identifying the most relevant and effective changes and solutions according to a **quadruple helix**

approach. A **Living Lab** with the web based Snap4City **Development Environment** provides a comprehensive set of tools for developers and stakeholders to implement data ingestion and processing flows, **Data Analytics** algorithms, **Dashboards**, **IoT App**, **Synoptics**, **Custom Widgets**, and **Web and Mobile Apps** [BIgDataService2018].

The Snap4City Platform:

- **official Platform and Solution of FIWARE**: https://www.snap4city.org/467 .
- has been the **winner of Select4Cities PCP** https://www.snap4city.org/558 (by Antwerp, Copenhagen and Helsinki), in September 2019.
- Winner of the Open Data **challenge 2020 of ENEL-X**: https://openinnovability.enel.com/projects/New-smart-city-solutions-enabled-by-open-data
- **official platform of EOSC** (European Open Science Cloud) marketplace of the European Commission. https://marketplace.docker-fid.grid.cyf-kr.edu.pl/services/snap4city
- **GDPR compliant** and enforces privacy and security for data, **Dashboards**, **IoT Devices, IoT App, personal data, data analytics** and **processes**, etc., which can be private of the user; and the user may delegate the access to them or pass the full control to other users. https://www.snap4city.org/670
- provides tools for the administrators to monitor and control the platform status via: auditing, assessment, management, GDPR, network flow monitoring, revoking, request to delete, etc. https://www.snap4city.org/download/video/course2020/sys/
- passed **PEN Tests** from Setek and Vulnerability test from Thales Mayor expert companies. https://www.snap4city.org/669
- can be installed on premise as well as on any public or private cloud. https://www.snap4city.org/471
- can create end-2-end solutions, secure from device to dashboards.
- official library of Node-RED, and official SigFOX partner
- Km4City API of Snap4City are accessible API via E015 of Regione Lombardia
- Etc.

Snap4City.org is operated by DISIT Lab and

- Last hackathon organized by Snap4City with IEEE Intelligent Transportation Systems https://www.snap4city.org/757
- organized successful international Hackathons for coworking: Florence, Tuscany, Helsinki, Antwerp; https://www.snap4city.org/hackathon
- is supported by a number of companies and partners: https://www.snap4city.org/577
- provides continuous training courses with certification. https://www.snap4city.org/622

3.1 – Snap4City Architecture Overview

A **more technical version of Snap4City architecture is reported in the next figure**, where the main elements are reported and described below. **Snap4City** platform consists of a set of 100% open source. **No proprietary components are needed to be installed into the Snap4City platform, and any proprietary solution can be integrated with.**
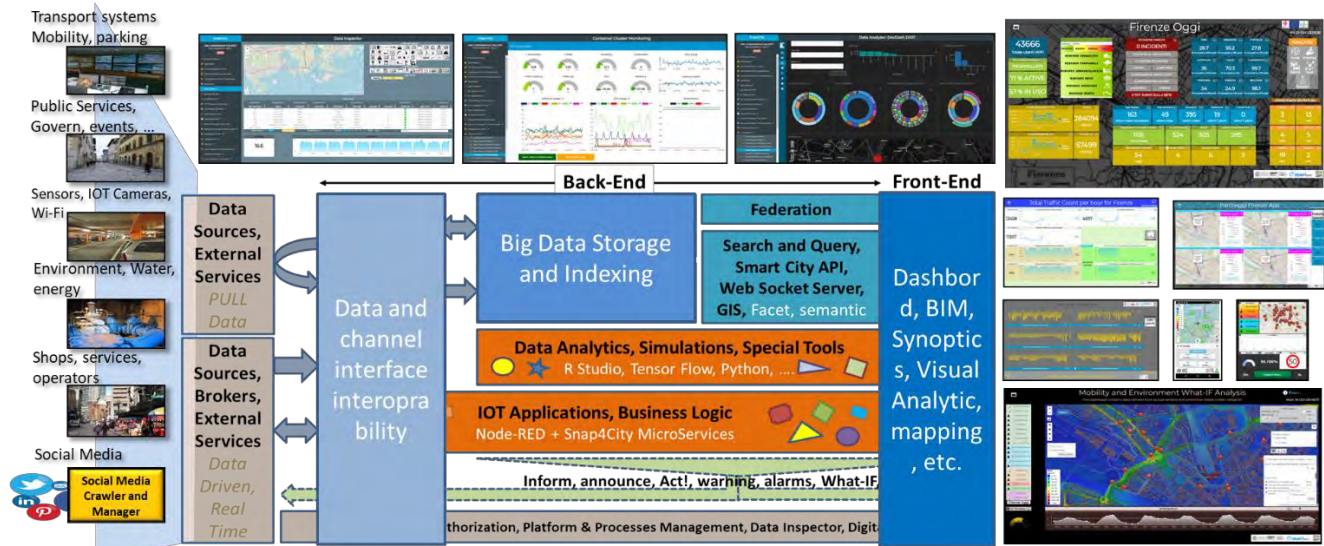
General overview accessible on https://www.snap4city.org/download/video/course2020/gen/

From Left to right of the above figure.

Data Ingestion and aggregation is performed in push/pull by using: IoT Connectors, IoT Agent, **IoT Adapter, IoT Broker, IoT Apps** in Node-RED, and data gathering tools as **DataGate**/CKAN for open data (https://ckan.org), ArcGIS, Data Table loader, POI Loder, and **Portia** for Web Scraping. The **Snap4City** platform supports a very large number of protocols push/pull, sync/async among them: MQTT, NGSI, COAP, OneM2M, ModBus, OPC, WMS, WFS, and AMQP. See for the list of the protocols supported Https://www.snap4city.org/65 **Data Models, IoT Device Models**, **IoT Devices**, and **IoT Broker** are registered via **IoT Directory** which is multiprotocol multibroker tool for IoT Network management. Among the brokers, a mayor role is played by **IoT Orion Broker** of FIWARE by which the platform support NGSI V1, V2, LD with **Services/Tenant** and **Service Paths**. The IoT Directory is also capable to automatically deploy Orion Brokers on demand. And its elements as IoT Agents, etc. A detailed description of IoT capabilities and network are provided in: https://www.snap4city.org/download/video/course2020/iot/ while a large number of detail regarding the data ingestion are reported in https://www.snap4city.org/download/video/course2020/di/

Data Storage is feed by using a **NIFI Cluster** for massive data aggregation/enrichment (collecting data from any **IoT Broker** at which it is automatically registered by the IoT Directory), and saving data to **Open Distro** for **Elastic Search** cluster for data storage and indexing, and to the **Knowledge Base** (based on **Km4City ontology**) implemented as an RDF store (Virtuoso) which is an index for geo-spatial, relational, and temporal aspects. Every time a new data model is registered in the system, the registration is performed into the **Knowledge Base**. Knowledge base and Smart City API can be **federated** creating a mutual connection among friend cities. In the Snap4City Platform, (i) Heatmaps, Orthomaps and Maps are managed by the **Heatmap Server** which is a **GIS** (for example **GeoServer or ArcGIS** if you have one installed with WMS facilities) and can distribute the maps according to WMS/WFS protocols towards Web and Mobiles App and Dashboards; (ii) Buildings 3D shapes, and Floors are managed and distributed into the **OpenMAINT** tool, standard BIM tools are used for editing and interchange in IFC formats by standard tools as AutoDesk Revit, etc. Buildings and Floors are shown in **Dashboards** for their integration with maps and IoT Devices time trends. Thus, creating a full **Digital Twin** model

of any part of the city. The tools for accessing to the **Digital Twin** is the so-called **Data Inspector**. Details can be obtained from: https://www.snap4city.org/download/video/course2020/di/

Data Transformations (reconciliation, regularization, enrichment, etc.) are implemented via **IoT Apps** (IoT Applications) which are processes for ingesting, integrating and manipulating data with PUSH/PULL protocols from any **External Service** and also from **IoT Broker, and Virtual IoT Devices** of any kind. **IoT Apps** are developed by using a visual programming **data flow and/or data driven** flows in **Node-RED** exploiting a large library of more than 150 **Snap4City MicroServices**.

> Please note that, in Snap4City Terminology an IoT App is a process in Node-RED plus Snap4City library freely accessible on Node-RED library. The Snap4City Library for Node-RED is based on nodes and call MicroServices, for this reason also the nodes are called MicroServices. So that often, it is stated that, the Snap4City library in Node-RED provides a large number of MicroServices, in the sense that provide access to MicroServices via corresponding nodes.

IoT Apps are also used for implementing user interface logic **Business Logic** behind smart applications and **Dashboards** also in references of **Custom Widgets** and **Synoptics**, which can be very useful in **Smart City Control Rooms**. https://www.snap4city.org/download/video/course2020/das/

Please note that Data Transformation / Integrations may exploit internal and external storages from **any** kind database, **any** connectors from/to data sources, **any External Service**, **any** direct connection with IoT Devices and Networks, etc. A large number of details regarding the data ingestion are reported in https://www.snap4city.org/download/video/course2020/di/

Data Analytics are processes written in **Rstudio** or **Python** which can perform data access and apply statistics, Machine Learning, Artificial Intelligence, deep learning tools, reinforced learning, algorithms also exploiting Tensor Flow, CUDA, Keras, etc. It is also possible to develop processes in JavaScript directly into IoT App, and in Java scheduling processes with **DISCES**. **Snap4City Data Analytics** in Rstudio, Python and JavaScript are developed with an online development environment which also allows to put in execution the scripts in **Containers** which are deployed as **API** and **MicroService** for **IoT App** and other tools in the platform. Off-line development environments are viable as well, if preferred. **Data Analytics** can produce: predictions, early warning, traffic flow reconstruction, classifications, tracking, alarms detecting anomalies, KPI and indicators, Typical Time Trends, typical trajectories, Routing and paths, travel plans, query results, simulations, analysis, calibrated heatmaps, smart parking suggestions, car sharing analysis, what-if analysis, automated ingestion of satellite data, social media processing and sentiment analysis, etc. A large number of examples are provided in: https://www.snap4city.org/download/video/course2020/da/

Smart City APIs include a large collection of services to: exploit queries and reasoning on the storage and **Knowledge Base**, access/control IoT Network, exploit **Data Analytic** results, exploit **IoT Apps**, etc. All the data and services are accessible via the **Smart City API** which are used by Front End Tools such as **Dashboards**, **Web and Mobile Apps**, **MicroApplications**. Details regarding Smart City API are reported in: https://www.snap4city.org/download/video/course2020/app/ Knowledge base and Smart City API can be **federated** creating a mutual connection among friend cities**.**

Dashboards can be created by **Dashboard Builder** and/or **Kibana (Open Distro)** for different kind of users such as: decision makers, city operators, ICT operators, private for the users, etc., suitable for **Smart City Control Rooms** with video wall, for city Operators on Desktop of multiple monitors, mobile operators, and Situation Rooms with touch panels. **Dashboards** can exploit all kind of data and **Data Analytic**, legacy services, and special tools as traffic flow reconstruction, decision support systems, what-if analysis, scenarios definition, etc. **Dashboards** are created by using a large range of ready to use **Widgets (for rendering data time series, data on map, interacting with data and tools, providing visual analytics),** and also **Custom Widgets** creating **Synoptics** as **SVG** elements and animations, or exploiting other graphic libraries, such as D3, Highcharts, etc. Examples on

dashboard and how they can be authored are reported in https://www.snap4city.org/download/video/course2020/das/

Tools of **Snap4City Platform for BPM/BIM** are discussed in **Section 3.15**. They can be used for passing from 3D view of the **Local Digital Twin** to the map and/or data time series and analysis and viceversa. Snap4City has also the possibility of providing **Global Digital Twin** for navigating into the smart city 3D building.
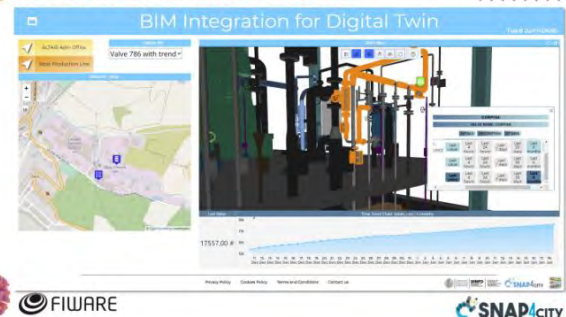
Aspects of Platform **Management and Quality Control** of back-office are presented in Sec**tions 3.16** with managing users' activities, platform setup and control, user auditing, assessment, monitoring, security, process management, and elastic scaling. Details and examples are reported in https://www.snap4city.org/download/video/course2020/sys/

Web and Mobile Applications can be created by developers exploiting **Smart City APIs** and may be controlled by Snap4City tools. For example, to send on the Mobile Apps: engagements, soundages, stimulus, and thus for monitoring user behaviour, creating origin destination matrices, getting reaction from the city users, informing user at the inception of critical conditions, etc. Details regarding Smart City API and their usage for the development of Web and Mobile Apps with related development environment are reported in: https://www.snap4city.org/download/video/course2020/app/

Aspects of Platform **Management and Quality Control** of back-office include managing users' activities, platform setup and control, user auditing, assessment, monitoring, security, process management, and elastic scaling. Details and examples are reported in https://www.snap4city.org/download/video/course2020/sys/

3.2 – Snap4City Innovation Methodology and Living Lab

In the context of Smart Cities, the smart city engine plays a relevant role for increasing city technological level and accelerating the smartening and simplification of city services and processes. To this end, the Smart City engine should play the role of catalyser of the innovation processes, collecting and sharing data (private and public, static and real time), to produce services and information for the stakeholders. All the loops are closed returning information to the providers, suggestion strategies and actions, alerting and early warning on dashboards and control rooms, supporting decision makers, and supporting the new plans and the development of strategies of the city.
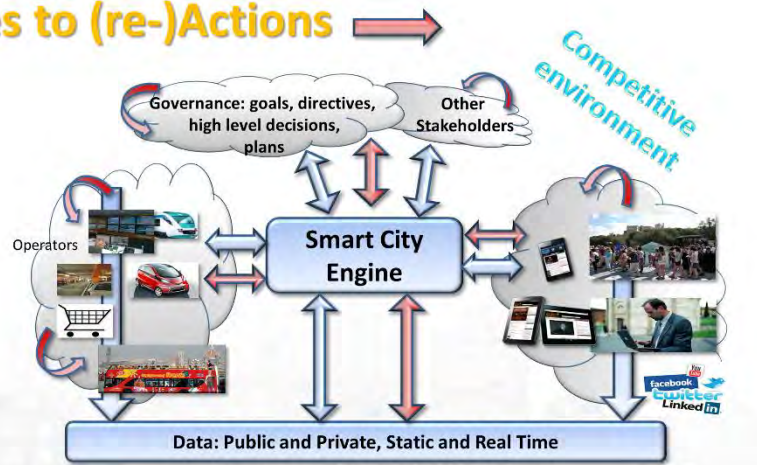
Smart Cities need to set up a flexible **Living Lab** to cope with the city evolution in terms of services, city users' needs and capabilities. To this end, the **Snap4City** solution provides a set of tools and a flexible method and solution to quickly create a large range of smart city applications exploiting heterogeneous data and stakeholder services also enabled by Internet of Thing (IoT)/Internet of Everting (IoE) technologies and Big Data analytic. The **Snap4City** platform and method has been realized involving different kind of Organizations (Universities, SME and Large Industries, Public Administrations) and users (City Operators, Resource Operators, Inhouse companies, Tech providers, Category Associations, Corporations, Research groups, Strat-ups, Early Adopters, large industries, advertisers, City users, Community builders, etc.), thus reflecting the features, described in **Quadruple Helix** (QH), to facilitate the Living Lab approaches in a Smart City (see the next figure). The innovative aspects of the solution proposed are related to semantic computing of entities for discovering and search information, resources management, parallel and distributed computing and cloud management, applications based on microservices and external services, dashboard and development tool kits, etc. The proposed solution is flexible enough to support extensions at distinct levels of granularity: data, analytics, tools and applications.

One of the first activities for setting up a **Living Lab** in a city is the process of creating the technical infrastructure which in turn is grounded on many valuable enabling tools. They must support the city in: the modelling of data; the upload of context data and open data; the connection of IoT/IoE sources and external services; the creation of **IoT Apps** / Integrations processes and **Data Analytics** algorithms, to arrive at producing smart city dashboards and at starting the production of **IoT Apps** based on **MicroServices**. All these phases must be accompanied and supported by the availability of a set of development tools, easy to use, accessible and open. To this aim, the **Snap4City** platform provide a collaborative environment in which different kinds of stakeholders can mutually collaborate. At the same time in which the setup is created, the collaboration among stakeholders can start by creating: agreements, collaborations, networking, producing tutorials, workshops, hackathon, etc., so at to arrive at involving the stakeholders around use cases, and finally to sign contracts of partnership, licensing, etc. Thus, the delivering of specific solutions to city users, operators, etc., is becoming possible. This process must be driven by the municipality and, on the other hand, the municipality needs support for technical aspects if it is not very large and technological oriented. Typically, the single companies even if participated by the city or the city operators, do not have the view and the mission to put in common a so large multi-domain multiservice framework and environment. [LivingLab2018], [Gov2018], [Gov2020].

The approach is a methodology to arrive at producing the **Snap4City innovation matrix**. It is implemented by means of a number of workshops (physical or online) to arrive at producing a deep analysis of the current situations and the needed steps for fast evolution toward the smartening of the processes, by taking the more efficient direction and the most acceptable solutions for the stakeholders.

See **training course part 6**. https://www.snap4city.org/download/video/course2020/sys/

The scenarios identified during the workshops may be focussed on specific domains (mobility, energy, gov services, water, environment, etc.) and with the aim of improving services to specific user categories. To this end, a process of **Data Discovery** is also performed by talking with the stakeholders and their technicians to verify the possibility of collecting the needed / available data as well as to create new IoT Network of Devices to measure phenomena that is presently under focus. Please take in mind that, typically the same Scenario (target smart solution) may be realized by starting from different kind of data. So that the scenario is the target, the data are a tool, and the experience may help you in identify among the different data sources those that may be simpler to be acquired and exploited, technically and legally speaking.

One of the first steps to set up a Smart City or add new services can be the data ingestion, and thus connections with data sources/recipients, that in **Snap4City** can be performed in several manners. All the connections can be bidirectional and based on their own protocols. **Snap4City is capable to ingest data via those protocols but also produce data back with the same or other protocols**.
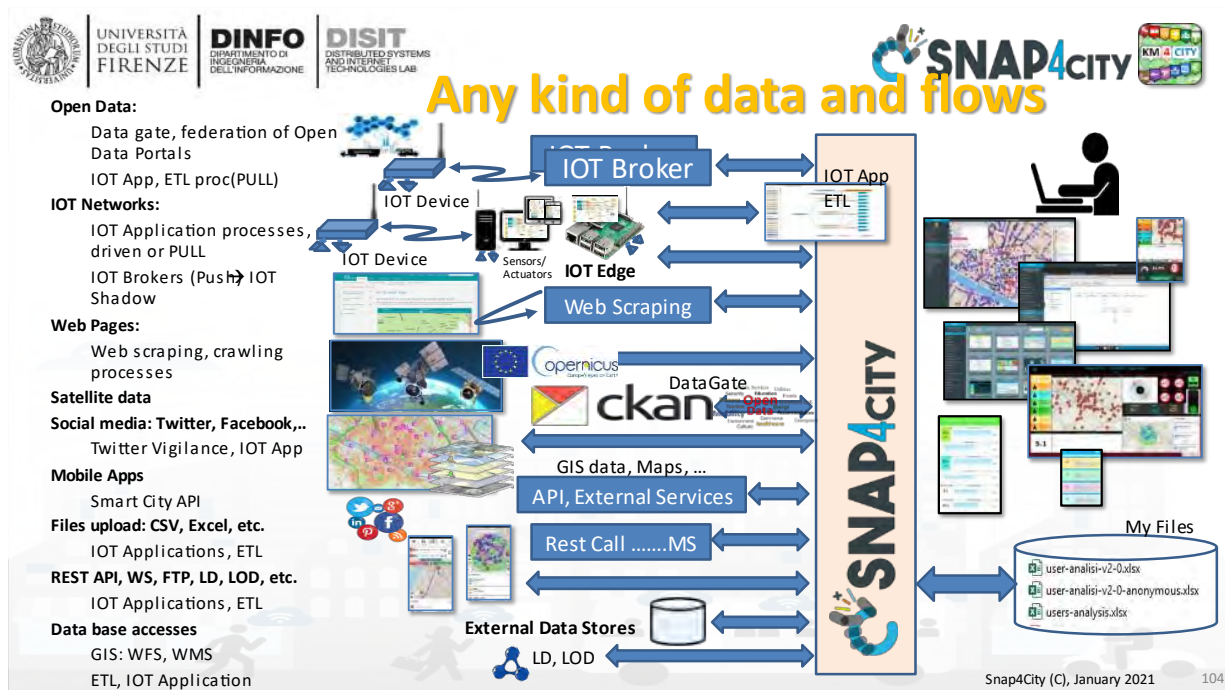
Snap4City platform can interact with legacy (of any vender and format) installations on IoT Networks, and thus with IoT Edge Devices, Gateway, etc. They can communicate via their own protocols or using the direction connection with Snap4City exploiting protected communications. IoT Edge devices with Node-RED can directly install Snap4City library to exploit from the platform: data ingestion processes, data model registration, Data Analytic, dashboards, direct interaction with dashboards and users.
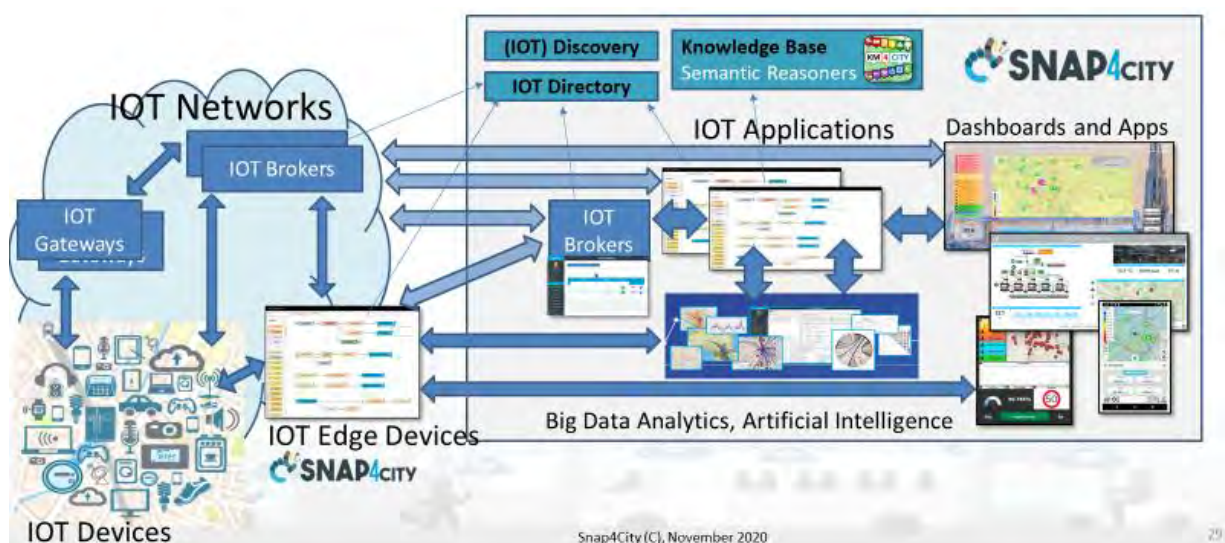
Main channels for data ingestion/interaction are:

- **IoT App** in **Node-RED**: collecting data from **External Services (e.g., IoT Gateways, web servers, ftp server, social media services, linked open data, linked data, ESBs)**, by using any kind of Connector, calling any kind of protocol and/or getting/transforming any kind of format into a Data Model / **IoT Device Model** for the Platform. Ingestion can be in PUSH/PULL to set up periodic/sporadic processes, as well as data driven (real time) processes of data ingestion and reaction/action for the platform towards other internal/external elements including other **IoT App**. See **Section 3.6 and Section 3.7 for interoperability.** https://www.snap4city.org/download/video/course2020/iot/ https://www.snap4city.org/download/video/course2020/di/

- **IoT Brokers**: which connect IoT Networks of **IoT Devices**.
  - A mayor role is played by **IoT Orion Broker** of **FIWARE** by which the platform support NGSI V1, V2, LD with **Services/Tenant** and **Service Paths**.
  - The **IoT Orion Brokers can** be controlled by the same administrators of the Snap4City Platform on your premise (so called **Internal IoT Broker**) or can be **External IoT Broker** controlled by third parties. Brokers may exploit: IoT Agent, **IoT Adapter, wrappers, IoT Edge Devices.**
  - **Snap4City** platform supports many different **IoT Broker** and more than 70 different protocols push/pull, sync/async [Https://www.snap4city.org/65] among them: MQTT, NGSI, COAP, OneM2M, ModBus, OPC, OPC-UA, Copernicus, and AMQP. See for the list of the protocols supported.
  - **Supported data models are: FIWARE Data Model, and IoT Device Models. The former are automatically converted for the Knowledge base.**
  - **IoT Devices and IoT Broker** are registered into **IoT Directory** which is multiprotocol multibroker tool for IoT Network management, also providing features of **IoT Discovery** in collaboration with the **Knowledge Base**. The registration of IoT Devices into Knowledge Base from IoT Orion Brokers is performed automatically.
  - **IoT Brokers can be deployed on demand by using the IoT Directory services.**

- **Acquire data from Satellite Copernicus. https://www.snap4city.org/671 with information and data related to: vegetation, quote, humidity, temperature, rain, and many many others. Which can be taken for large regions and not only on specific sensors points.**
  - https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MzAwNQ==
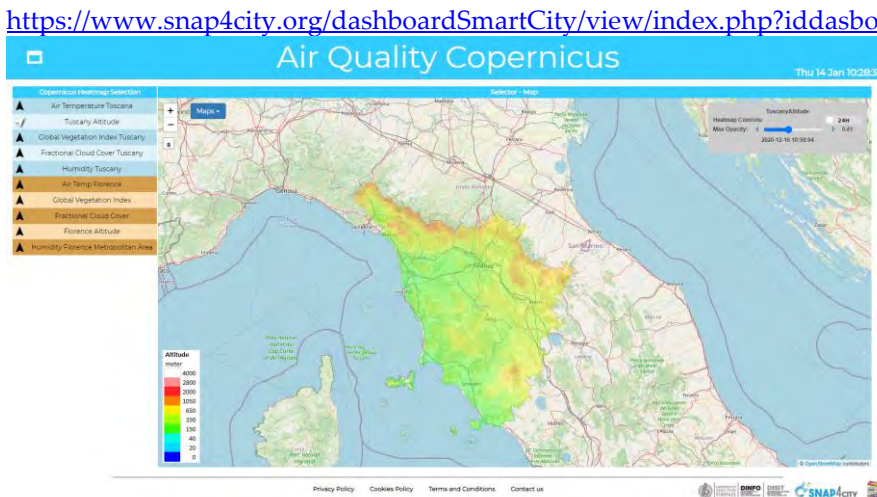
  -
- **Data Table Loader** for shortening the activities of registering devices, loading time series, etc., by loading the data from excel files.
- **POI Loader** for shortening all the activities of ingesting new data for point of interest providing them as excel files.
- **Work with top 10 used DBs** and more, as: Mongo, Azure, MySQL, Microsoft SQL, MariaDB, As400, Cassandra, Elastic Search, Oracle, IBM DB2, Postgres, AWS, etc. and via standard connectors such as: ODBC, JDBC, etc. via IoT App.

- **DataGate/CKAN** for open data (https://ckan.org): collecting open data from the Open Data network of CKAN harvesting via different protocols. And integrating those data into the platform. **DataGate**/CKAN can be also controlled by **IoT App** in **Snap4City**. CKAN support any kind of files in the domain of Open Data and any CKAN instance can connect with a network of CKAN in the world by harvesting data and sharing data as files and metadata descriptors.
- **Web Interface:** to add/change data via web page, for example: **POI, MyKPI**, etc.
- **Web Scraping/Portia**: activating a process to grab data from your and/or third parties web pages and convert them in real time to a Data Model in JSON, **IoT Device Model** for the platform. Web Scraping processes are also controlled by **IoT App** in Snap4City.
- **Files**: of any format they can be loaded via: (i) specific IoT App to set up specific data ingestion processes such as via Web page, FTP, SSH, DropBox, Google Drive, etc., (ii) **DataGate**/CKAN. Files loaded can be any format and size.
- **Web and Mobile Apps:** they typically connect with the platform to get data and suggestions, and also they may send data back as preferences, trajectories, queries on sequences, clicks, etc. (if the user agree, data as sent anonymously or according to a signed consent in agreement to GDPR). This huge amount of data can be used for providing contextual services and tuning the service to the city users. The Mobile App users can be registered on the same pool of platform users or may be decoupled from the platform central user registration. See **section 3.5.**
- **Dashboards:** they typically connect with the platform to get data and suggestions, and also they may send data back. The dashboards may present Actuator (Virtual IoT Device Sensor Actuators) to send control data to the platform such as: dimmer, know, switches, selector, clicks, text, forms, etc. This information can be consumed on the fly by IoT App which are the Business Logic of the User Interface, UI, and/or they can be saved into the platform for further usage. See **Section 3.10**
- **Direct URLs**: such as URI/URLS to perform rest call to external services, such as GIS server. They can be directly accessed to show data on Dashboards, exploiting protocols such as WFS and WMS, which can be also produced by Snap4City **ServiceMap**.
- **Data Analytic** processes may create new data, which can be saved into the platform and may contribute to the vision of the city via **Dashboards** and to produce innovative and smart services. For example, they can produce: predictions, early warning, traffic flow reconstruction, alarms detecting anomalies, KPI and indicators, Typical Time Trends, typical trajectories, Routing and paths, travel plans, query results, simulations, analysis, calibrated heatmaps, smart parking suggestions, car sharing analysis, etc. See **Section 3.9.**
- **Reports:** may create consumptive views on the platform status on specific programmable aspects for users and administrators, for example at level of **IoT Devices**, **Dashboards.**

3.3.1 – IoT Network components

IoT Networks may be composed of several IoT Broker compliant with different protocols and corresponding devices, IoT Adapter, IoT Agents, etc. See training part 3
https://www.snap4city.org/download/video/course2020/iot/

Therefore, the following elements in addition to those above introduced are relevant:
- **IoT Device**: An IoT Device may have sensors and/or actuators. In Snap4City, an IoT Device can be registered on IoT Broker before sending data on the platform. If the Broker is internal, the IoT Device can be registered on IoT Directory that perform all what is needed to register on IoT Broker and also on Knowledge Base and NIFI to automatically perform the Data Shadow of all the data produces by the device. This is possible since NIFI is automatically subscribed to all Devices of the Internal Brokers.
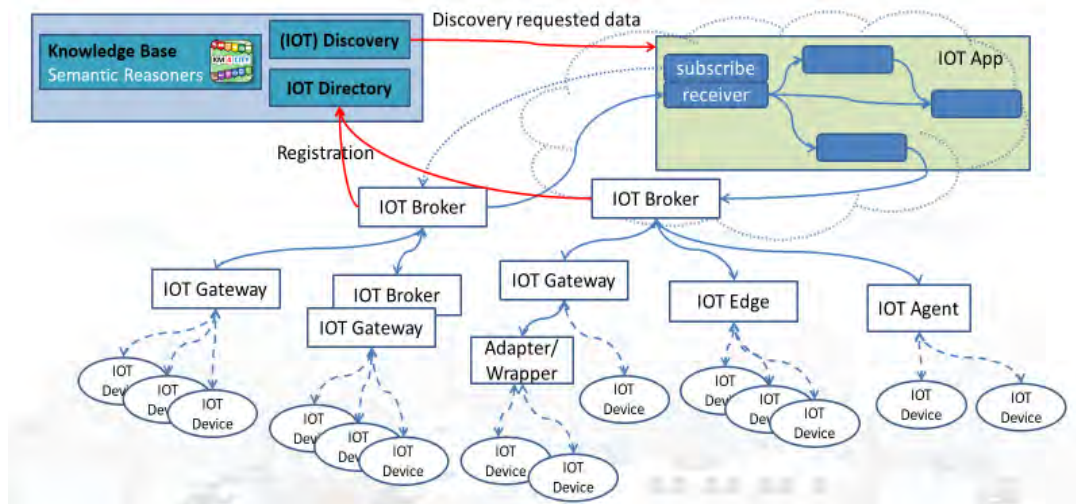
- **IoT Device Model**. **Supported data models are: FIWARE Data Model, and IoT Device Models,** models for IoT Devices, Virtual IoT Devices, time series, moving IoT Devices, etc. **They are automatically converted/ingested for the Knowledge base, creating the unified view of entities coming from any vendors and broker channel.** In Snap4City, an IoT Device Model can be registered once and shared and used many times for instantiating one or many devices with the same model in short time. For example, 40.000 lights of a smart light system. Please note that the Model is a template, and once used can be modified without any impact of the produced instances.

- **IoT Orion Broker is a FIWARE** component: https://FIWARE-orion.readthedocs.io/en/master/ The IoT Orion Broker support natively the NGSI V1, V2 and LD. It usually provides only the last values of the IoT Devices and not the historical data (data shadow). In order to have the Data Shadow, and thus to perform queries on time series, the IoT Orion Broker has to be connected to a storage. In Snap4City, the storage provided is Elastic Search which provides high capabilities for insert, search and retrieval and scalability.

- **IoT Directory:** Snap4City tool for registering IoT Broker and Devices. Snap4City is almost agnostic about the brokers (services for Orion Brokers are very extensive) and accept a number of brokers and protocols registered on IoT Directory of Brokers and Devices. The IoT Directory is capable to browse on internal and external brokers to discover (in the case of Orion Broker) IoT Devices and register them on Knowledge Base. https://www.snap4city.org/115

The IoT Directory is also the tool that manage all the network information about the devices deployed and connected. It is capable to exploit NGSI V1 and V2 protocol aspects to inspecting and managing IoT Orion Brokers of FIWARE, so that to manage a large number of them and automatically performing registration of devices in bulk on Knowledge base. See https://www.snap4city.org/76 https://www.snap4city.org/562, https://www.snap4city.org/647, https://www.snap4city.org/763

Device Discovery MT/SP on External Brokers

- **IoT Discovery**: It is a function of Snap4City IoT Directory and Knowledge base which allows to discover the IoT Devices by a number of filters: geospatial, by type, by value name, by unit, by nature and subnature, etc-https://www.snap4city.org/109 The same Feature is provided in IoT App, IoT Directory, ServiceMap, Data Inspector and Dashboard Builder.
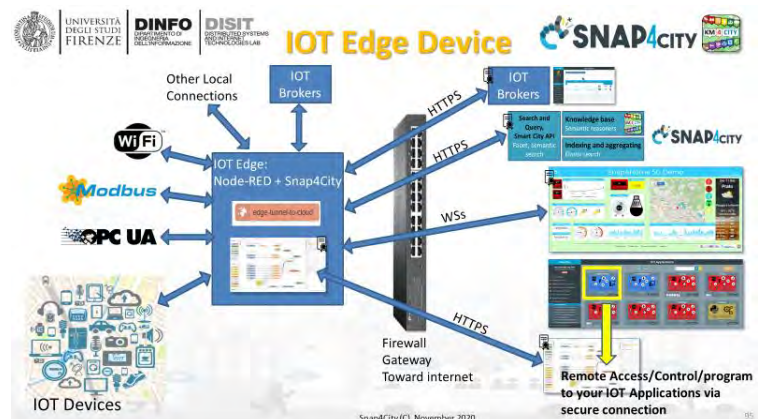

IOT Discovery on IOT Application Development
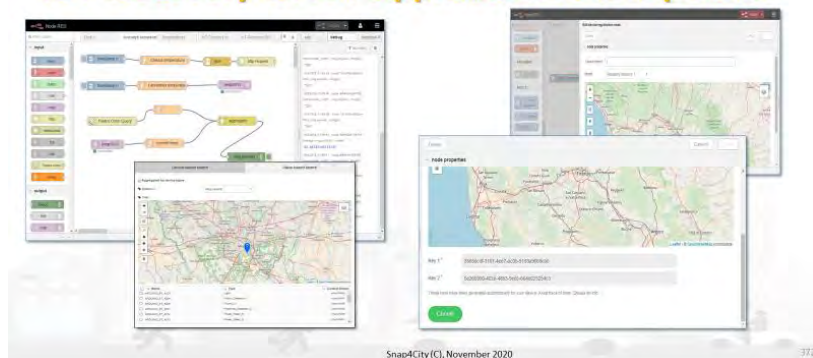
- **IoT Edge Device**: An IoT Device capable to execute processes. In Snap4City terminology, IoT edge Devices are typically endowed of Node-RED process which also has installed Snap4City Libraries of MicroServices. See **section 3.5**, and also
  - o https://www.snap4city.org/646 for the remote control of IoT Edges
  - o Edges on Linux https://www.snap4city.org/298
  - o Android https://www.snap4city.org/278
  - o raspberry pi. https://www.snap4city.org/279
  - o Snap4Home: https://www.snap4city.org/617
  - o Snap4industry: https://www.snap4city.org/369

IOT Edge Device

3.4– Snap4City Data Model, Ontology and Digital Twin

According to the previous section, the data ingested passed automatically from **Data Models** to be registered on **Knowledge Base**. The registering the data model is the first step to enable the possibility of loading data instances which can be: Static, quasi static and dynamic real time data, as even driven processes and streams. This process is totally automatic providing data in Excel Table and exploiting the Data Table Loader and/or the POI Loader.

For manual approach, the first step in the conquering the approach is to understand the **IoT Device Model** which is based on the data **Dictionary** and **Snap4City Ontology** as reported in the next subsection 3.4.1. Then the view of the Snap4City **Data Inspector** (section 3.4.2) which is the major tool for understanding not only the **Digital Twin** information but also the **relationships** of Data with **processes** and **users** which are the tree aspects of any Smart engine for smart city and industry.

In fact, in the management of a Smart Solutions, one of the main problems is to keep trace of the complex network of processes and causes and effects from the data and their usage into Data Analytics, IoT Apps, Dashboards, etc. **Data Inspector is the tools that allows you to keep all the knowledge of the Smart City back office under control.**

3.4.1 - Snap4City Ontology and Reasoners, a plus in addition to IOT Data Modelling

*Snap4City organizes all Data Models according to **Km4City Ontology** to build a **Knowledge Base** which consists of the **Expert System of the city**. It allows to navigate into the unified data model of the city, exploiting geospatial, temporal and relational aspects. The **Knowledge Base** has a number of tools for its access and management such as **ServiceMap, Smart City API**, **Wizards** and **Data Inspector** (**Digital Twin**). The **Data Inspector** go beyond the Digital Twin since it give at the operator access at integrated representations of **city entities, devices, processes and users.***

Km4City Ontology has been defined to address in a uniform manner any city related domains and data models [https://www.snap4city.org/19] . https://www.snap4city.org/download/video/DISIT-km4city-City-Ontology-eng-v5-1.pdf Each single data elements of each single model aims at modelling and establishing the needed relationships among elements, thus making a general data set semantically interoperable at model level. In the definition of a **Knowledge Based** grounded on **Km4City**, the usage of the expert system satisfies to crucial functional needs as:
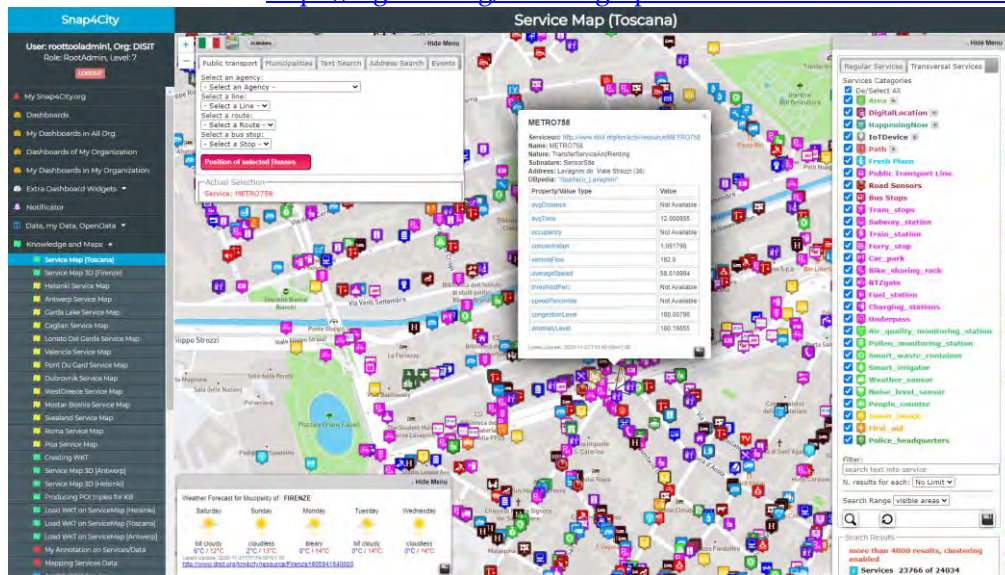- coverage, the data model is detailed enough to represent everything of interest for real applications and services;
- time-efficient, the Knowledge base has to be queried providing smart services, and users/clients request high performance and scalability (this means to apply solution for design for performance);
- space-efficient, given the level of detail to be reached, the less the triples are, the better it is. The result of the deep analysis and practice is the **Km4City Ontology**.

Smart cities domains addressed: mobility and transport, energy, health, economy, key performance indicators, KPI, people flow, commercial, entertainment, culture, sport facilities, industry, govern services, financial, education, tourism, security, emergency, agriculture, accommodation, civil engineering, advertising, etc. On the other hand, a different technical view can see those services modelling as Point of Interest, POI, road structure, civic numbers, car parks, traffic flow, bus timeline, RTZ (restricted traffic zones, passages), tram lines, bus stops, statistics, ordinances and resolutions, events, people moving and arriving in the city, vehicles movements and flows, points of interest, present weather and forecasts, parking and forecast, private data owned by mobility and transport operators, recharging stations, pollutants and pollinations, cost of fuel, and much more. For example, sensor devices can be used to model data collected for monitoring pollution, pollination, traffic, people track, parking, RTZ, irrigators, etc. In the same manner, forecast data can be performed for weather, parking, road
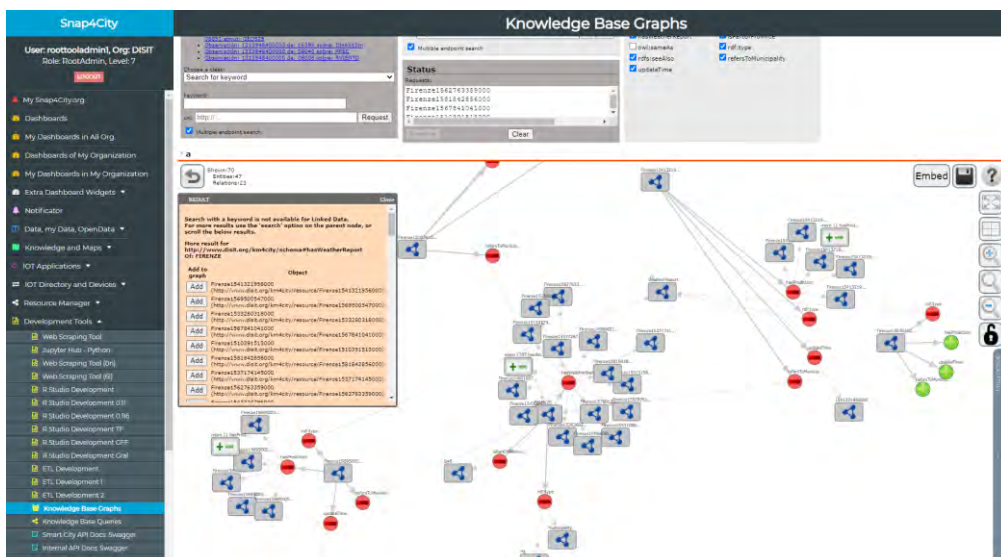
accidents, people flows, vehicles flows, pollutant, temperatures, etc. Typical trends can be used for any data varying over time, which may have a day, week, etc., some seasonality. And addressing Industry 4.0, industry plant such as: water depurator, waste recycling, heating systems for large stadium and large buildings, fleet management, maintenance team management, copy and publication centers, data centers, etc.

Please note that the simple data modelling at level of Data, IoT Device and IoT Broker is not enough to model MyKPI, Point of Interest, GIS information, BIM information, shapes, heatmaps, traffic flow, typical trajectories, typical time trends, cycling paths, time schedule and plans, routing, multimodal routing, GTFS of public transportation, etc., while the Km4City Ontology is capable to integrate in a uniform model all these aspects and models.

The main tool to access at the Knowledge base are the ServiceMap https://servicemap.snap4city.org/ and its **LOG.DISIT.ORG** as RDF browser: https://log.disit.org/service?graph=e7fe1456503401b7bdd69fd3988143ea


Service Map

Knowledge Base graph LOG.DISIT.ORG

The mixture of theory, data analysis, and work on-the-field that has characterized the last few years, has led to an evolution in the data model, that starting from that described in [KnowledgeBase2018], is now made up of the

10 macro-classes plus metadata depicted in the following Figure. A selection of examples of relations among macro-classes is provided as labelled links.

- **Places** (as Administration) enables the modelling of public administrations and their resolutions, it also includes classes for hamlets, suburbs, commercial, industrial, residential areas, that have a key role in context-rich applications, and for which relations, ways and node tagging.

- **Street graph** (as Street-guide) enables the representation of the anatomy of mobility infrastructures through concepts such as Road, Node, RoadElement, AdministrativeRoad, Milestone, StreetNumber, Entry, and their related properties, OTN (Ontology for Transport Network).

- **Points of Interest** (POIs): includes all services, activities, which may be useful to the citizen and who may have the need to "search-for" and to "arrive-at". The two-level categorization of services and activities with more than 30 classes and more than 520 subclasses.

- **Public Mobility (as Local public transport)** the ontology includes concepts for the modelling of public transport infrastructures, time schedules, real-time records. More recently, the General Transit Feed Specification (GTFS) Linked Open Vocabulary has also been imported and it is used with minor integrations, that has allowed to supersede the local dimension of the original modelling.
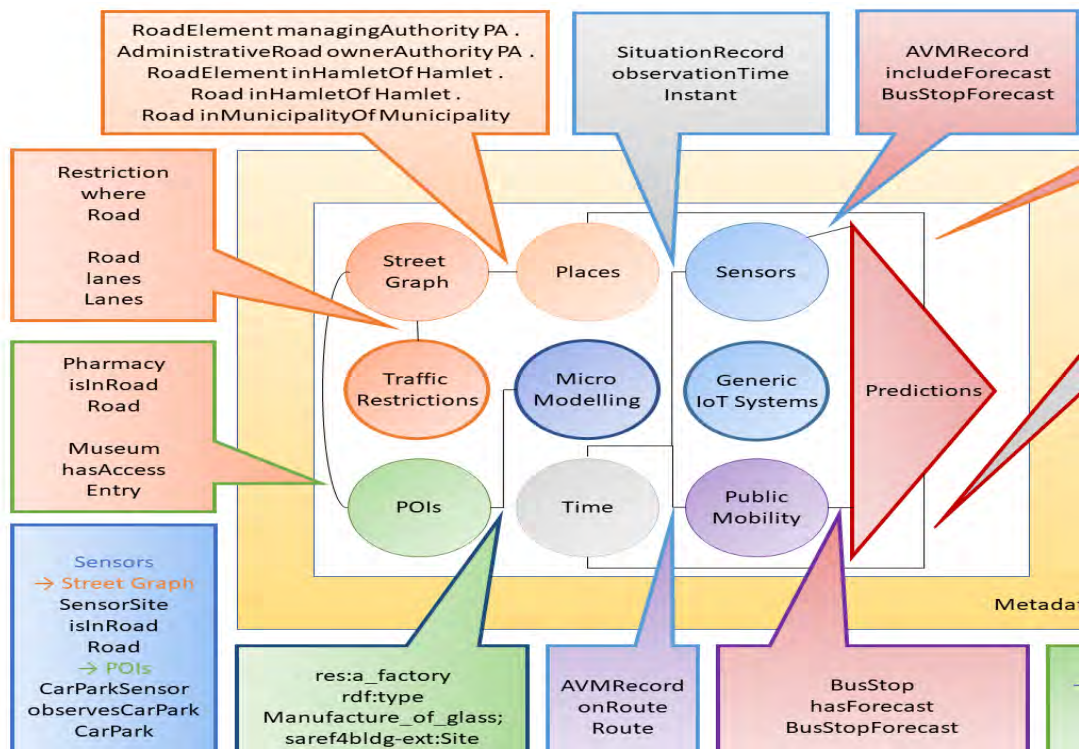
- **Sensors**: concepts for the modelling of specific types of sensor is part of the ontology since the early times, for traffic and car park occupancy.

- **Time (formerly, Temporal)**: concepts and relations wrapped that enable the representation of time intervals and instants of time to be associated to detections and predictions.

- **Metadata**: collection of metadata associated with datasets and dataset status, such as if they have been ingested and integrated. Such as: data of ingestion and update, licenses information, versioning, and so on.

- **Traffic restrictions**: in this macro-class concepts and relations are wrapped that enable the representation of forbidden and mandatory directions at crossroads (possibly at the lane level), access restrictions (even temporary/conditioned ones), size, weight, speed limitations, at the maximum possible level of detail, with the semantic of values that is borrowed from the definition of corresponding Open Street Map tags, and therefore public, maintained, rigorous. Not only routing, but also traffic prediction and reconstruction are so enabled, and therefore advanced routing, air quality prediction, and more.

The **Km4City ontology** reuses **standard vocabularies most of the of W3C**, such as:

- *dcterms*, set of properties and classes maintained by the Dublin Core Metadata Initiative;
- *foaf*, dedicated to the description of the relations among people or groups;
- *schema.org*, for a description of people and organizations;
- *wgs84_pos*, vocabulary representing latitude and longitude with the WGS84 Datum of geo-objects, since the present RDF store and indexing engine allows to perform geographic queries for example to identify the POI which are closer than a given distance with respect to a specific GPS position, prior the building of a specific index;
- *gtfs*, General Transit Feed Specification, dedicated to the modelling of public mobility data;
- *SSN*, semantic sensor network, a W3C ontology for the modelling of IoT Systems;
- *iot-lite*, aimed at a lightweight modelling of the Internet-of-Things;
- *saref4bldg*, SAREF extension for building, aimed at modelling topologies, items, flows, at a high level of detail.

Other Vocabularies can be added to Km4City Ontology.

3.4.2 - Snap4City Data Inspector, integrating: Digital Twin, Users, processes with their relationships

In fact, in the management of a Smart City, one of the main problems is to keep UNDER CONTROL the complex network of processes, with the several producers of information and consumers, causes and effects from the data and the processes, their exploitation and dependencies in Data Analytics, IoT Apps, Dashboards, etc.

Data Inspector is the tools that allows you to search and view all the knowledge of the Smart City services in the back office under control and easy to understand.


Smart city context includes solutions that presents

- **Data (and their Digital Twin information)**: data are heterogeneous, large volume, several protocols, legacy data systems, several semantics, real time, multiple domains, etc., such as: any kind of data for the smart city. They belong to the so called **High Level Types, HLT:**
    - **sensors, actuators, POI, MyKPI, personal data, heatmaps, external services, synoptics, micro applications, traffic flows, OD matrices, etc.**
    - **they can be observed in terms of:**
        - **Models:** which are the templates from which devices are produced
        - **Category** which is the classification in groups as nature and subnature
        - **Devices** which are the single element that take measure which can provide/receive several measures at the same time
        - **Variables** which are the single measure, the single elements/sensors, actuators
- **Processes**: which can be: aperiodic, periodic, event driven, …. Managed by different tools (IoT Brokers, ServiceMap, HeatMap Manager, Web scrapers, IoT Apps, Traffic Flow Manager, etc.):
    - **Data Ingestion**: the models and mechanism for data gathering, Data Table Loder, POI Loader
    - **Data Transformation: IOT App / Node-RED, integrations, ..**
    - **Data Storage, indexing, etc.**
    - **Data Analytic: predictions, anomaly detection, computing, etc.**
    - **Business Logic behind Dashboards**
    - **Data rendering, Dashboarding, interacting**
- **Users:** owners (who register), responsible, providers, developers, etc.
- **Relationships**: among data, processes, users and mixed.

The Snap4City **Data Inspector** is the primary point to understand the issues and produce the reports on those facts. For example, at level of: users, devices, process, dashboard and application. See this document on the early Version [DI-DataInspectorBase]. Now we have deployed since October 2021 a new version of Data Inspector and engine with the extended classification of HLT.

Data Inspector (of the 2021) is showing data:

- **Listed by High Level Types,** and classified as in the Dashboard WIZARD: see Tutorial Part 2 for details: https://www.snap4city.org/download/video/course2020/das/
- **of your Organization** only, that are public of the organization, your private or those that have been delegated to you some how.

You can perform (see https://www.snap4city.org/download/video/course2020/di/ **)**
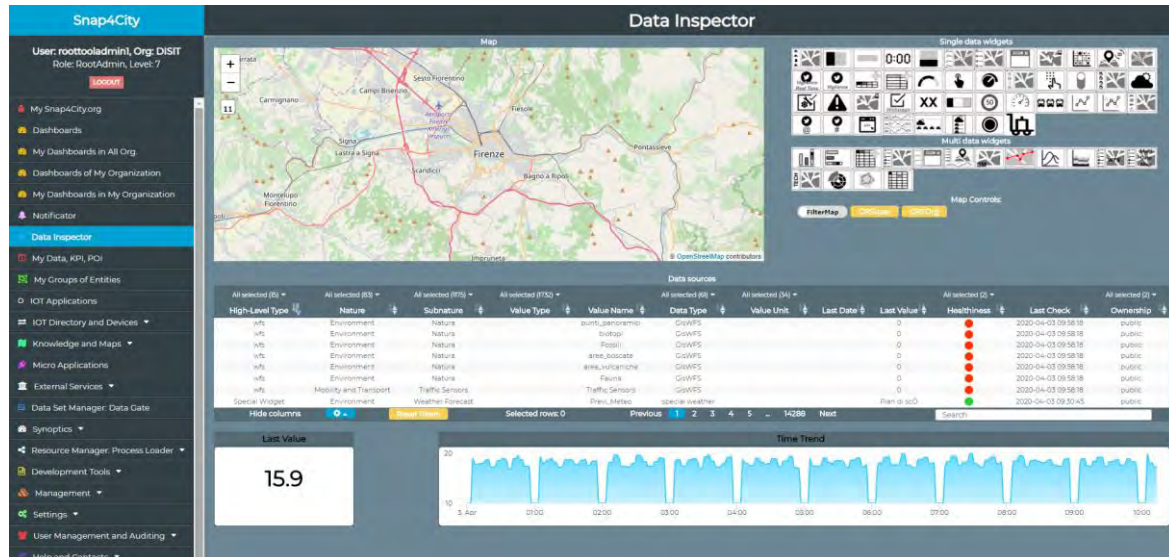
- **Cross Filtering on the basis of**:
    - **MAP**: pan and zoom → lock , center on GPS coordinates of the user
    - **Data Source Classification**:
        – faceted filtering
        – full text search
    - **Click on data source** to see it on map, and see the graphics representation, just to learn how to "widgetting" it.
    - **Full Text Search** if you remember some description…
- **Selecting a Data Source on Map**: on its Pin you can see:
    - **Real time data**
    - **Time trend: 4 hours, 12 hours, 1 day, 1 week, 30 days.**
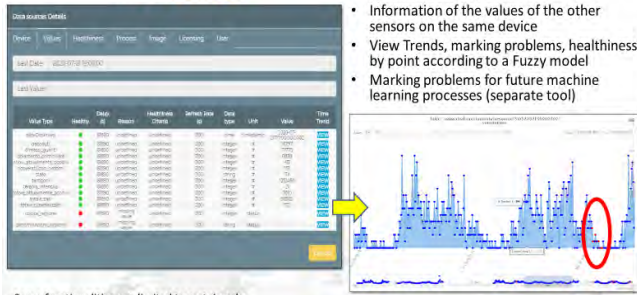   - **Full status and description (only for Administrators)**

From the Data Inspector you can order your report and get them.

From the Data Inspector you can see the data of dashboards and the relationships among Dashboards, IoT App, Data, Data Analytics, etc.

Reports on devices can be monthly or trimonthly.

3.5 – Snap4City APIs and Smart City API, MicroApplications, Web and Mobile Apps

In this section, the smart city API are presented, together with some of their major applications, that is the MicroApplications and the Mobile App with what they can do for the city, in terms of services and data.

3.5.1 - Snap4City APIs and Smart City API, for developers

Snap4City Platform provides a rich set of Smart City APIs and technical APIs, which can be used to develop Web and Mobile Apps and other applications and tools, as wel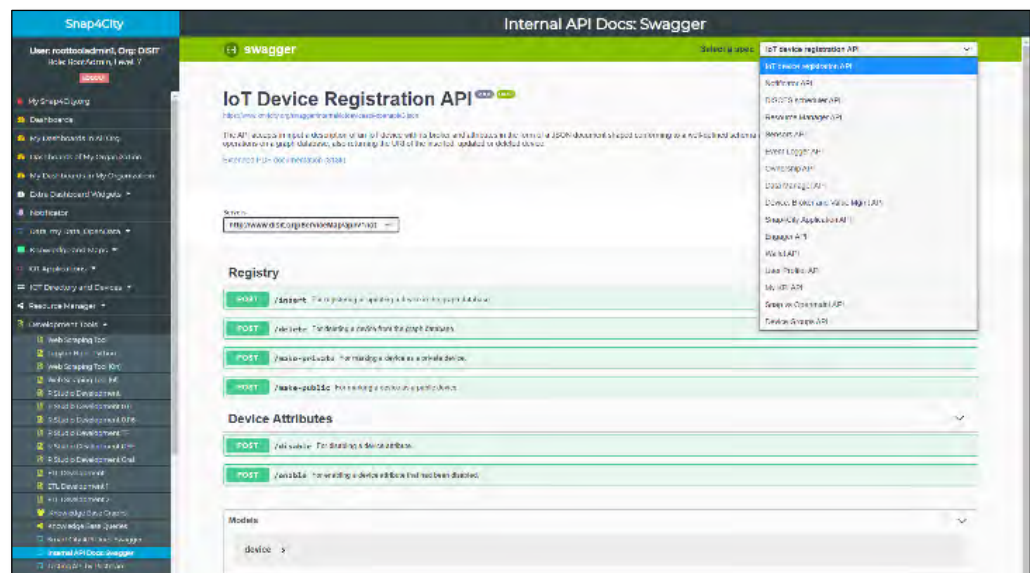l as new modules for your solution. Snap4City APIs are accessible in **Swagger** and are divided in Internal and External.

Internal APIs:
https://www.km4city.org/swagger/internal/index.html
- **IoT Devices and tools API:**
  o **IoT Device registration API:** API of the IoT Directory
  o **Sensors API:** API of the IoT Directory
  o **Device, Broker and Value Mgmt API:** API of the IoT Directory
- **Mobile App Management**
  o **User Profiler API:** To manage the user profile for the **Engager** on Mobile Apps
  o **Engager API:** From the Engager to prepare engagements to the Mobile Apps
  o **Wallet API:** From the Engager to Wallet o the users of Mobile Apps and in general
  o **Snap4City Application API**
- **Resources and entities Management**
  o **Snap4City Application API:** To manage IoT Apps
  o **My KPI API:** To manage MyKPI, MyPOI, POI, etc.
  o **Data Manager API:** to access at personal data
  o **Resource Manager API:** To manage resources on the market place
  o **Ownership API:** To manage ownerships and delegations
  o **Device Groups API:** To manage ownerships and delegations
- **Notificator API:** to register events to be controlled, get events/notifications
- **DISCES scheduler API:** to control scheduling processes
- **Event Logger API:** to log data
- **Snap vs OpenMAINT API:** Integration with the workflow management, BPM, and ticketing

External APIs: https://www.km4city.org/swagger/external/index.html
- **(Advanced) Smart City API:** To access the **ServiceMap** resources and query
- **Km4city Web App API:** To exploit **MicroApplications** created as tools for **Dashboards**, totem, web Apps, etc.

- **Orion Broker K1-K2 Authentication:** To communicate with **IoT Orion Brokers** exploiting the Secure Filter of Snap4City to enforce security to NGSI protocols.
- **Heatmap:** To save and access to **HeatMaps** of the Heatmap server

The **Smart City APIs** (the specific segment used for the web and mobile Apps) provides support to:

- **search data**: by text, near, along line, POI, resolving text to GPS and formal city nodes model, resolving from GPS to street civic number, etc.;
- **get information of services** in the territory such as: mobility and transport, cultural and tourism, e-health, public transportation, environment, weather, commercial, etc.;
- **get info on mobility** such as: pedestrian and car routing, multimodal routing, navigation in critical conditions, bike cycling paths, parking status, bike rack status, bike/car sharing, public transportation operator services: paths and timelines, ..;
- **obtain predictions**, for example on parking status, first aid status, traffic flow, people flow, social events, weather, temperature, environment data, etc.;
- **provide information to the city**: ranking, comments, images, feedback to direct questions (open and closed);
- **receiving notifications and alerts**, for example: alerts of civil protection, events of traffic, connected guide V2V and V2I, OBD2 errors, etc. The users on the move can subscribe on the notifications to receive alerts about a large range of events or when some data overcome the official warning level;
- **get heatmaps**, for example on environmental parameters, weather, hot places, environmental predictions, traffic accidents, etc.;
- **get suggestions**, assistance, incentives to a more virtuous behaviour
- get and set, …etc.

List of Mobile Apps which are exploiting Km4City/Snap4City Smart City API

- **Firenze Dove Cosa, Km4City:**
  - Android https://play.google.com/store/apps/details?id=org.disit.siiMobile&hl=en&gl=US
  - Apple iOS: https://apps.apple.com/it/app/firenze-dove-cosa-km4city/id1028356115
- **Toscana dove cosa, Km4City**
  - Android: https://play.google.com/store/apps/details?id=org.disit.toscana&hl=en&gl=US
  - Apple iOS: https://apps.apple.com/it/app/toscana-dove-cosa-km4city/id1064554200
- **Antwerp in a Snap**
  - Android: https://play.google.com/store/apps/details?id=org.disit.snap4city.mobileApp.antwerp&hl=en&gl=US
  - Apple iOS: https://apps.apple.com/it/app/antwerp-in-a-snap/id1467737363
- **Helsinki in a Snap**
  - Android: https://play.google.com/store/apps/details?id=org.disit.snap4city.mobileApp.helsinki&hl=en&gl=US
  - Apple iOS: https://apps.apple.com/it/app/helsinki-in-a-snap/id1466970280
- **Tuscany in a Snap**

- o Android:https://play.google.com/store/apps/details?id=org.disit.snap4city.mobileApp.tuscany&hl=en&gl=US
  - o Apple iOS: https://apps.apple.com/us/app/toscana-in-a-snap/id1471094480
- **Snap4Pisa: Pisa in a Snap:**
  - o Android https://snap4city.aedit.it/drupal//sites/default/files/snap.apk
- **Weee Life Mobile App:**
  - o Android: https://play.google.com/store/apps/details?id=org.disit.lifeweee&hl=en&gl=US
  - o Apple iOS: https://apps.apple.com/it/app/life-weee/id1470224854
- **Comune Facile**
  - o Android: https://www.e015.regione.lombardia.it/site/app-detail?id=131
  - o Comune Facile IOS: https://www.e015.regione.lombardia.it/site/app-detail?id=130
- **POS by EPSON:**
  - o https://www.e015.regione.lombardia.it/site/app-detail?id=58
- **SnapBot:**
  - o Telegram Bot based on Snap4City data and solution on Tuscany area
  - o https://www.snap4city.org/684
- **Feel Florence Mobile App**
  - o section regarding Herit Data
  - o https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=M...

3.5.2 - Snap4City MicroApplications

Snap4City also provides via API a number of **MicroApplications** and corresponding API derived from the Advanced Smart City API which can be directly embedded into **Dashboards** of any kind, and activated from IoT Applications Dashboards. Snap4City provides via API: (i) a number of **MicroApplications** and corresponding APIs derived

from the Advanced Smart City API which can be directly embedded into Dashboards of any kind, and activated from IoT App Dashboards, (ii) data via WFS and WMS protocols of GIS kind of applications. Snap4City also includes internal APIs to interoperate with legacy solutions and among its modules, those for: GIS with WFS/WMS data exchange, heatmap loading, IoT Device Registration, process management and scheduling, notification and alert management, resource management, inter module communications, data analytics, etc. The whole set of APIs is documented in Swagger and it is public accessible.
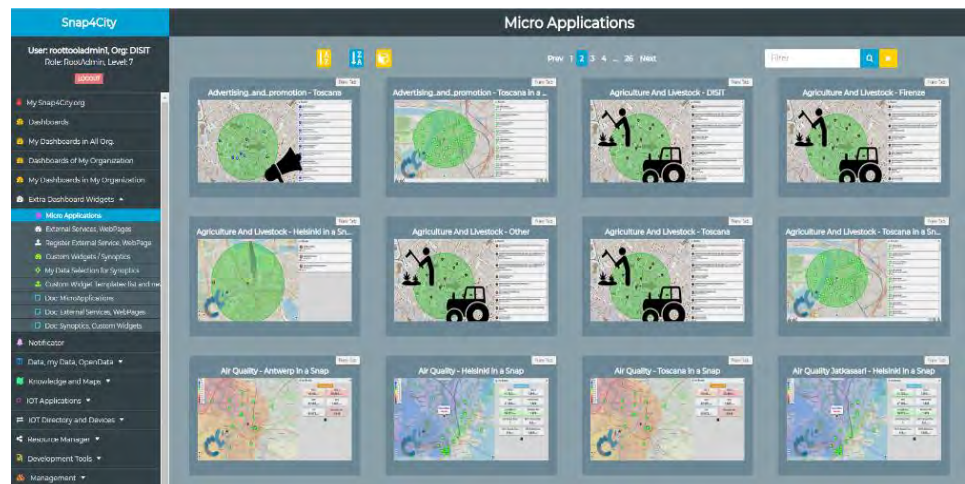
3.5.3 – Snap4City Mobile Applications and User Behaviour Understanding

The **mobile App** are bidirectional tools. They provide at the users geolocated and contextualized information, allow them to perform selections/queries, to view POI, to perform routing, to get predictions, to see images and comments, receive notifications and alerts, in various domain: mobility, environment, gov service, health, weather, etc. On the other hand, mobile users also provide information to the Snap4City Platform from which the platform can Derive information: trajectories, hot places, OD matrices, etc. Those information are provided to the users in contextualized manner and according to the signed consent, GDPR compliant. From the analysis of the data, it is possible to understand how the city is used by the city users as described in the following. See **Tutorial Part 7:** https://www.snap4city.org/download/video/course2020/app/

The App is a Bidirectional Device

Users

- GPS Positions
- Selections on menus
- Views of POI
- Access to Dashboards
- searched information
- Routing
- Ranks, votes
- Comments
- Images
- Subscriptions to notifications
- ....

Produced information
- Accepted ?
- Performed ?
- ...

Derived information
- Trajectories
- Hot Places by click and by move
- Origin destination matrices
- Most interested topics
- Most interested POI
- Delegation and relationships
- Accesses to Dashboards
- **Cumulated Scores from Actions**
- Requested information
- Routing performed
- .....

Produced information
- Suggestions
- Engagements
- Notifications
- ...

System

Therefore, the bidirectional data flow can create mutual benefits from the city users to city and operators.

Snap4City Smart City API and services can be used to create a number of Web and Mobile App of different kind. In the following and above some examples are provided. Those created by DISIT Lab are distributed from Google Play and Apple store: https://www.snap4city.org/489

Nothing prevent to: (i) create/modify their own Mobile Apps for exploiting Snap4City API, (ii) start from Snap4City App to create a new App for your city or area, etc.

https://play.google.com/store/apps/details?id=org.disit.snap4city.mobileApp.tuscany&hl=en_US
https://play.google.com/store/apps/details?id=org.disit.snap4city.mobileApp.antwerp&hl=en_US

- USER MANUAL OF:
    - Helsinki in a Snap: https://www.snap4city.org/drupal/node/492
    - Helsinki in a Snap: https://www.snap4city.org/drupal/node/493

Personalization and Profiling, user behaviour analysis

Apps Users are profiled in the server according to the GDPR, collecting a range of data as described above. This approach allows us to tune the service according to the collective and personal user behaviour, with the aim of improving the user experience providing personalized (i) main menu arrangement of the functionalities according
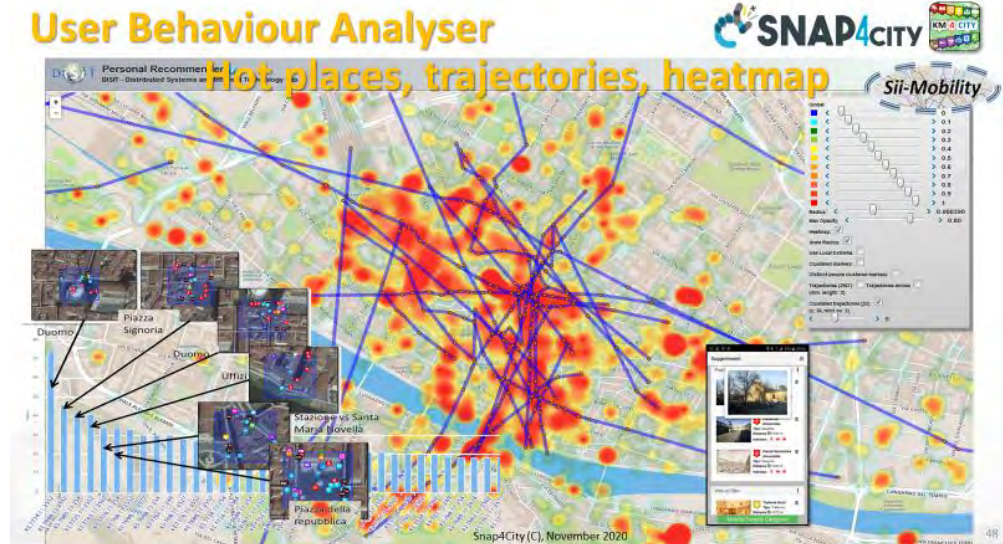
to the user profile (the user also may change it to have precise personalization), (ii) suggestions (the user may decide to ban some of the suggestion or even a category), (iii) menu search among categories (the user may decide to make further selection). Therefore, the menus offered for each profile kind is modified according to the statistics calculated on the collected data, by defining the positions the most researched categories, buttons, etc., and in some cases, putting off/on some functionality. For example, the triage monitoring may be of interest for citizens and operators and less for tourists, the parking status is interesting for users using the private car and less for those moving with the public busses or in bike, etc. The priority on the creation of the menu profiles is given in first place to the updated menu which can be found on the server. If the server is unreachable, the last saved menu for that specific profile is loaded; if the menu has not been saved, the loaded menu is the menu released with the application update, and so on.

Alerting and tracking

The city users need to be informed on what is going on the city. On the other hand, most of the mobile platforms have different approaches for providing asynchronous notifications in push to the device when the App is not in foreground, is not executed by the user. In addition to the alerting, the movements of the device should be also communicated to the server in order to get new context based suggestions and alerting, such as: please take care about the weather forecast in your area, alarms of civil protection, environmental status, closer car park, etc. Some of these innovative and smart features are produced as suggestion, engagements from personal assistant, predictions produced by the data analytics modules (e.g., parking, arrival of busses, etc.), alerts (e.g., civil protections, changes in the traffic, events). A new and additional feature of notification can be obtained integrating Telegram, which has to be installed into the App. In this case, the server identifying the critical situation has to interact with Telegram server.

App Development Kit, ADK (for Web and Mobile Apps)

The App Development Kit, is based on Apache Cordova framework which allows realizing hybrid applications on multiple platforms. The applications consist of user-interface implemented with HTML, CSS and JavaScript, and of plugins which allow to use the specific hardware functionalities from the different platforms (i.e., battery-status, camera, device orientation), through a JavaScript interface. It is possible to start from the ADK source code released as Affero GPL, available on https://github.com/disit/snap4cityAppKit , for all developers that would like to realize new modules for official applications on the basis of data available through Smart City API Km4City and/or from other sources and APIs that may be related to private data of the city operators.

Each module is substantially a mini-application in HTML5. The produced modules may be dynamically loaded inside the official application or provided into the App from the store. In the first case, every time the App is executed, it looks for the availability of new modules to load them from the server site, also updating the former versions when needed. The module loader directly locates the new and updated function and mini-apps on the menu, and when needed new buttons are connected to the new feature that appear and in the desired place according to the server manifest of the module.
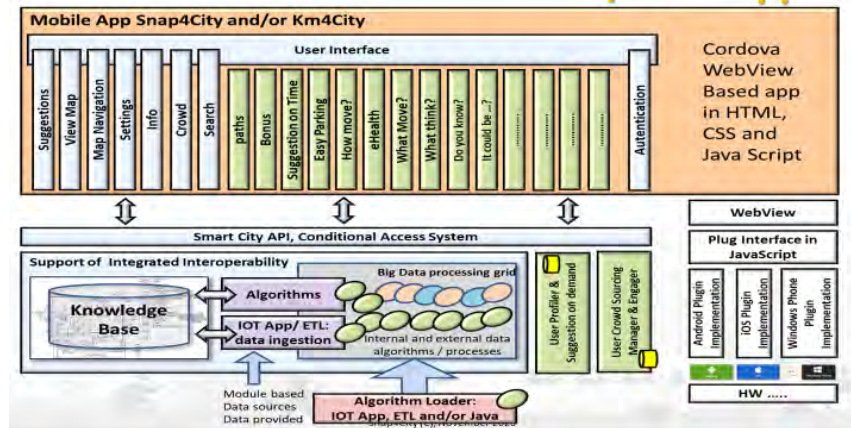

User Engagement via Mobile App

The User Engagement is one of the main challenges in the Smart Cities. Most of the smart applications imply some involvement of the city users, or at least the evidence of their success can be measured in terms of the city user appreciation and reaction to the proposals performed via Mobile Apps. Therefore, most of the cities need to communicate with the city users and to assess their reactions to different purposes. The communication of information is the first step, the real need is the User Engagement, which means to efficiently and deterministically obtaining reactions and evidence of the success, once the city user is
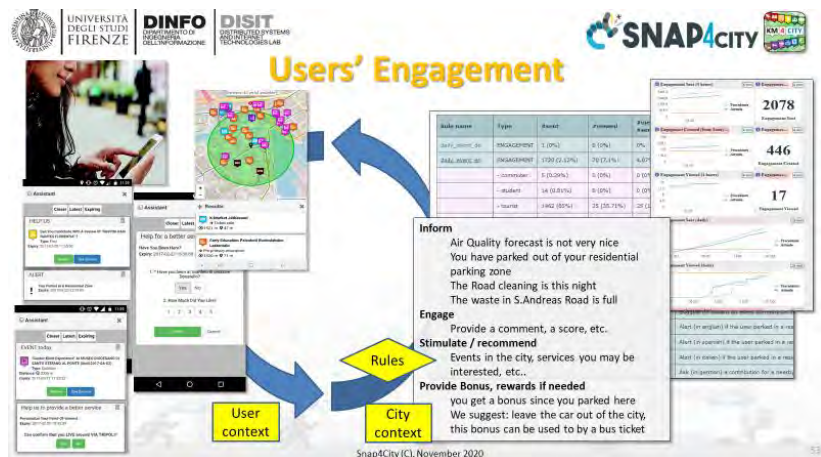
- Informed via a messages, for example about the closure of certain streets;
- Involved by providing back comments and opinions on specific aspects (answering to questionnaires), in a participatory manner, for example, to get suggestions about a moved bus-stop or changed service;
- engaged by performing specific actions, for example providing explicit comments, suggestions, taking pictures, raking, etc., on specific topics according to their context.

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

SNAP4CITY KM4CITY

The city users can be efficiently engaged only if the message arrives at the moment and in the locations in which the request has a sense. For example, suggesting to a commuter that is taking the private car every day to go from home to work, when a simple alternative based on public transportation is possible, inform the commuter in advance that his typical parking slot is not available, remind at the city user that is parking in area which is reserved for other kind of resident users.

In most cases, when the engagements are sent to stimulate a change in the habits or to inform the city user, some incentives/bonus could be provided to increase the acceptance and attraction. The incentives can be in the form of discount, points, awards, etc. Moreover, they can be assigned on the basis of specific rules, for example, when certain context or user behaviours is detected (for example, when the system understand that the user is moving with a private car), or when the change of habits has been verified (when, after to have received a suggestion, this has been accepted, for example, leaving the private car, and taking the bus).
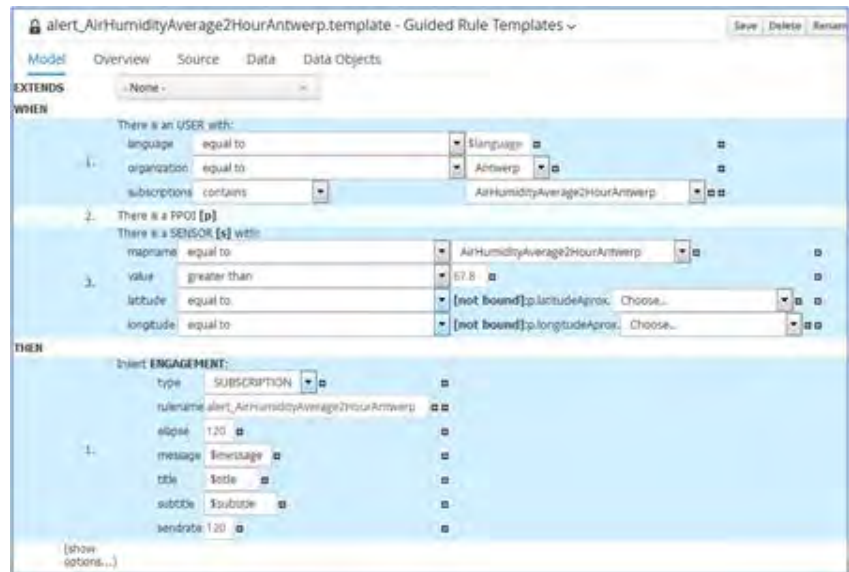
The solution has been used by ATAF, CCTNORD and BUSITALIA (public transportation operators) in the 2019 in campaign in Florence Livorno and Pisa. The above mechanisms can be implemented/configured in the Snap4City Engagement tools which include:

- A tool to easily formalize Conditions at which Actions has to be performed, such as: release of incentive, change of status, assignment of points, etc.;
- An engine for real time and automated estimation of Conditions and Actions, which is also capable to verify if the message has been received by who, where and when;
- A set of Mobile Apps in which those engagements can be received to engage the city users; the same Mobile App can be used to collect data regarding city users movements, preferred points and topics of interest, etc.;
- A Management tool for controlling the Engagement status, which exposes Dashboards for real time monitoring of the Engagement process.

The users of the App may use the App as anonymously or by registration providing or not a signed consent according to GDPR. The collected data are treated according to GDPR. The most relevant data analytics collected from the App are related to the: preferred places of the users, origin destination matrices, trajectories, user behaviours, comments on services, images on services and new POI, ranking and appreciations, etc. The App is a tool for the users for collecting their personal data which are located at their exclusive disposal on the Snap4City platform according to GDPR. They can save trajectories, personal usage, etc., and exploit those data with IOT data of the city or of their personal private IOT devices for creating IOT Applications and personal Dashboards. This allows creating a participatory community and a group of active city users which can contribute to the day by day activity of the Living Lab.

Real time Monitoring of Engagements and Provided Surveys

The above solution can be observed at work on Tuscany region, Antwerp and Helsinki downloading and installing the Mobile App from Google Play and Apple store. The City Operator may control the Engagement and monitor the results by using the above presented tools, such as the Monitoring tool and Dashboards reported.

Download as PDF: https://www.snap4city.org/download/video/tn/user-engagement.pdf

Other examples and connected scenarios:

- List of all scenarious: https://www.snap4city.org/4
- TC8.10- How to manage the user engagement's rules in the Snap4City platform https://www.snap4city.org/drupal/node/486
- Mobile App on all Stores: https://www.snap4city.org/drupal/node/489
- Dashboard: user Engagement for Administrators https://www.snap4city.org/drupal/node/472
- Snap4City mobile App: City in a Snap (available for Antwerp and Helsinki) https://www.snap4city.org/drupal/node/448
- Dashboard Life in Toscana with Origin Destination matrices, computed by Mobile App data: https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc3NA==
- Advanced Smart City API for mobile app development, development tutorial: https://www.snap4city.org/drupal/node/20

## Origin Destination Matrices, Algorithms and tools

The City Operators and Strategists on mobility and transport, on tourism and economic develop need to have precise view about the city users are moving in the city. From where they arrive, which kind of travel means are using, how they move in the city and in which time slots, etc. Most of these answers are provided in terms of Origin Destination Matrices. These can be computed from several different data sources, thus providing different views of the city flows:

- **Traffic flow sensors**: reporting the private traffic flow and eventually traffic from touristic busses, and not in the areas close to traffic. If they take the vehicles plates, they can be capable to be used for creating matrices.
- **Mobile Apps**: reporting the movement of the people, in the city and among the cities, inflow and outflow. The kind of users at which they refer depend on the App profile, and on the number of tracked trajectories.
- **On board units:** for example, coming from taxis, from insurance black boxes, bus OBU, etc. Also, in this case a partial view of the flow is produced according to the different kind of sources and the number of trajectories. They can be confined in the city (for the taxis) or wider area (for example in the case of insurance data).
- **Census data**: data collected from the periodic census in which also some questions related to the commuters: workers, students; which is asked to describe their day travel means to go and return for studying and working.
- **People tracking** tools, for example using Wi-Fi and/or Bluetooth sniffing devices, TV cameras, etc.
- **Cellular network data**, coming from the Telecom Operators, referring to cells or clustered areas.

Due to the introduction of GDPR some of the above data are becoming harder or practically impossible to be obtained. All these kinds of Origin Destination Matrices can be (i) used for simulation and analysis tools, as well as (ii) for direct visualization. Tools for the simulation and analysis may derive suggestion for public transportation, city consumption, city cleaning, service tuning, etc. On the other hand, a direct visualization may allow to perceive the critical aspects in a glance. For OD matrix visualization, the classical matrix representation is not very effective since the matrices tends to the very sparse. See for example the figure.
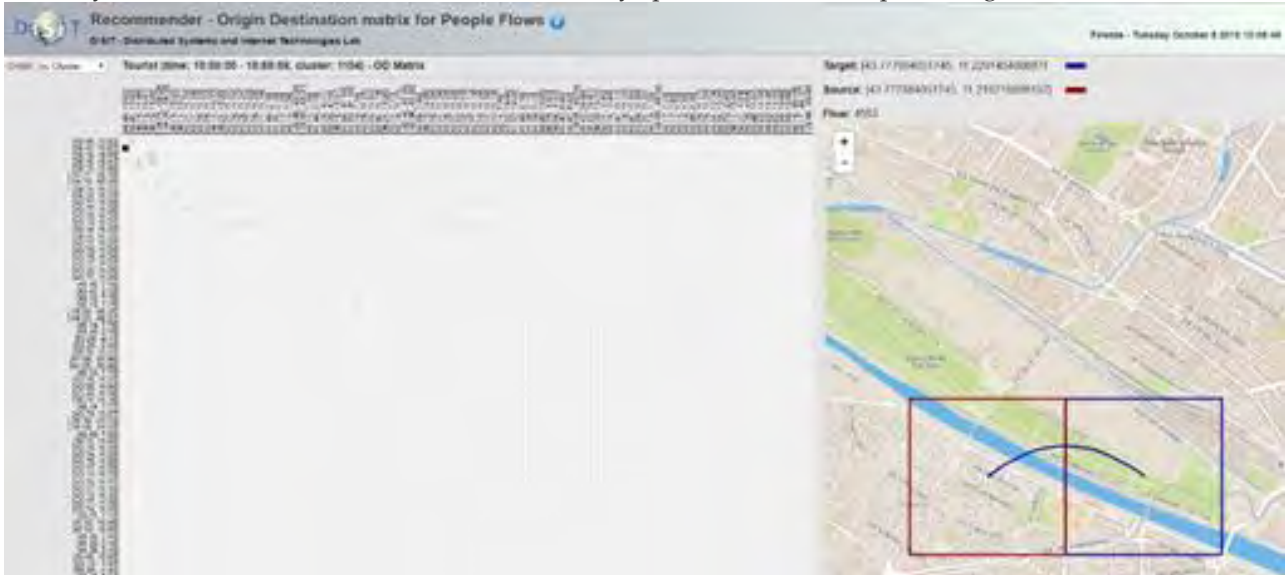
An alternative, directly invented by DISIT lab is the Spider Representation. The Visual Analytics tool for OD Matrix representation of DISIT Lab is rendering OD matrices as Spiders is much more powerful for the direct understanding of the flows in the city.
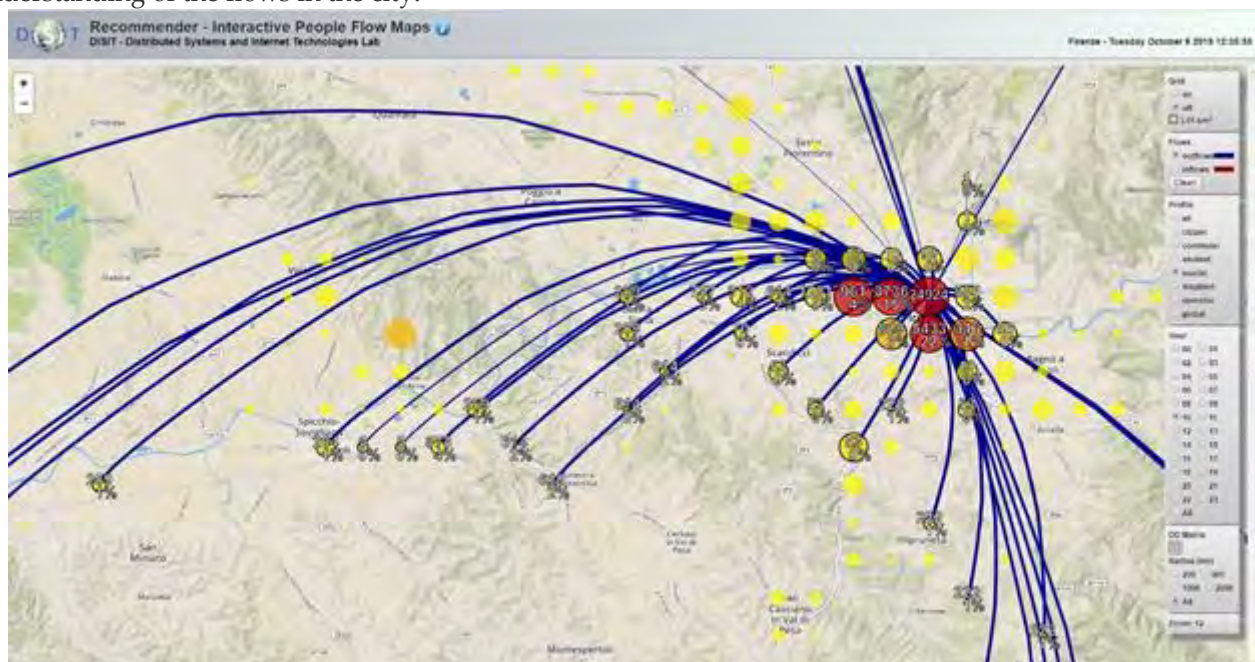

The proposed tool for OD matrix analysis one can:
- Select from Inflows and Outflows
- Hide and show the GRID
- Select the user profiles among: citizens, commuters, students, tourists, etc.
- Change the resolution and thus see the OD Matrix at different level of resolution, that means addressing different size of the areas. The areas can be squared as in the above case (with data coming from mobile

Apps, OBU, etc.) or non-squared as those coming from Wi-Fi, or cellular data, etc. For example, in the figure the OD matrix computed from Wi-Fi data is reported.

- Defining the time slot or see the whole 24 hours
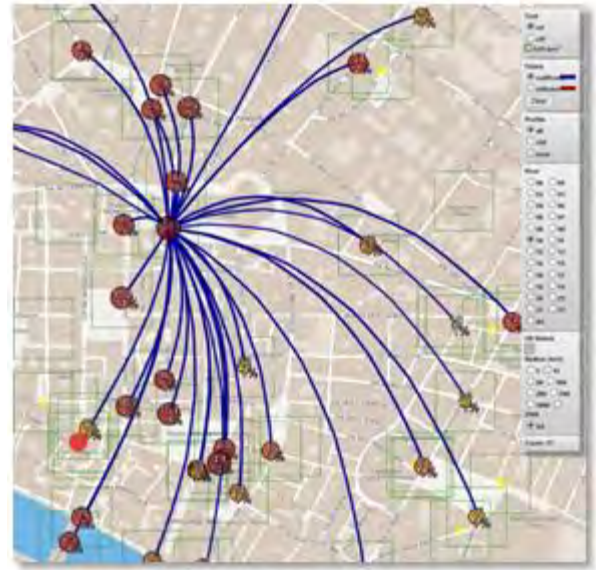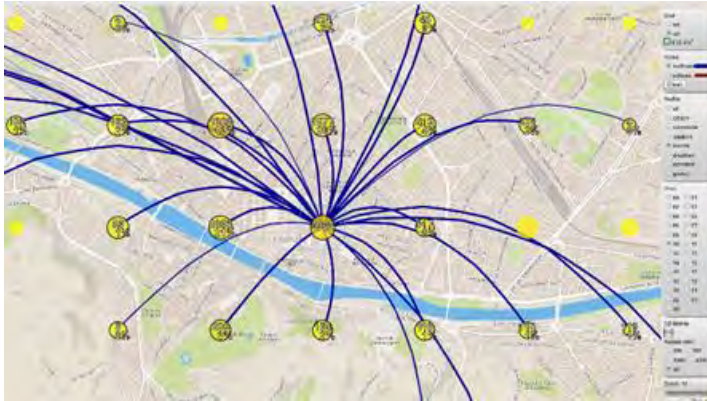
The computation of the scalable OD matrices is time consuming and it is typically updated daily taking into account data of the last months or weeks. Simplified version of the tools could be used for communicating with the citizens and business operators of the city, for example, see the Dashboards produced for Life in Antwerp, Life in Helsinki in which a view of OD matrix as spider is proposed.

Download this note on PDF: https://www.snap4city.org/download/video/tn/Origin-Destination-Matrices.pdf

More details and example are on:

- List of all scenarios: https://www.snap4city.org/4
- TC1.18: Origin Destination Matrix: https://www.snap4city.org/drupal/node/459
- Dashboard Life in Helsinki with OD matrix: https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc1Mg==
- Dashboard Life in Antwerp with OD matrix: https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTcwNg==
- Dashboard Life in Toscana with Origin Destination matrices: https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc3NA==

3.6 – Snap4City IoT Apps and MicroServices

The **IoT Apps** can be executed on-cloud or IoT Edge Devices (on-premise). When IoT Apps are executed on IoT Edge Devices, they may directly communicate with the IoT Apps or Dashboards on-cloud or by means of IoT Broker (to which all other entities can be subscribed). On such grounds, in Snap4City, the IoT Apps are defined as:

<div align="center">

IoT App = Node-RED + Snap4City MicroServices.

</div>

The IoT Apps exploit the basic nodes of Node-RED Node.JS plus Snap4City MicroServices and their visual presentation as nodes/blocks, which are suitable for smart city and IOT transformation and processing. The Node-RED platform is based on two components:

      (1) a web-based **visual editor** to design flows and
      (2) a runtime environment that may execute flows.

In Snap4City, the **IoT App/Node-RED visual editor** is fully integrated into the development environment and it has been improved to:

- communicate with the Snap4City **Resource Manager** to save and load IoT App and flows, which is a marketplace of resources for sharing and marketing them in the community,
- login via SSO (Single Sign On), exploiting Snap4City **LDAP** and **KeyCloak (which is an open IAM)**, using **OpenIDConnect** (OAuth and Access Token)
- manage Smart City / IOT MicroServices, namely, the Snap4City Libraries of Node-RED, which are accessible from the Node-RED official library: https://flows.nodered.org/search?term=snap4city

The runtime engine of Node-RED has also been improved to

- manage the security, according to SSO and the Snap4City model also using Access Token for authenticated M2M communications, and
- execute IoT Apps on Container/Docker, according to the elastic management solution of Snap4City (**Marathon** and **Mesos**, and eventually in **Kubernetes**).

The changes performed on Node-RED have been released as open source and are functional only for large-scale on-cloud use, while the Snap4City MicroServices can be used in any Node-RED installation and thus also on IoT Edge with the standard Node-RED tools, without any restrictions.
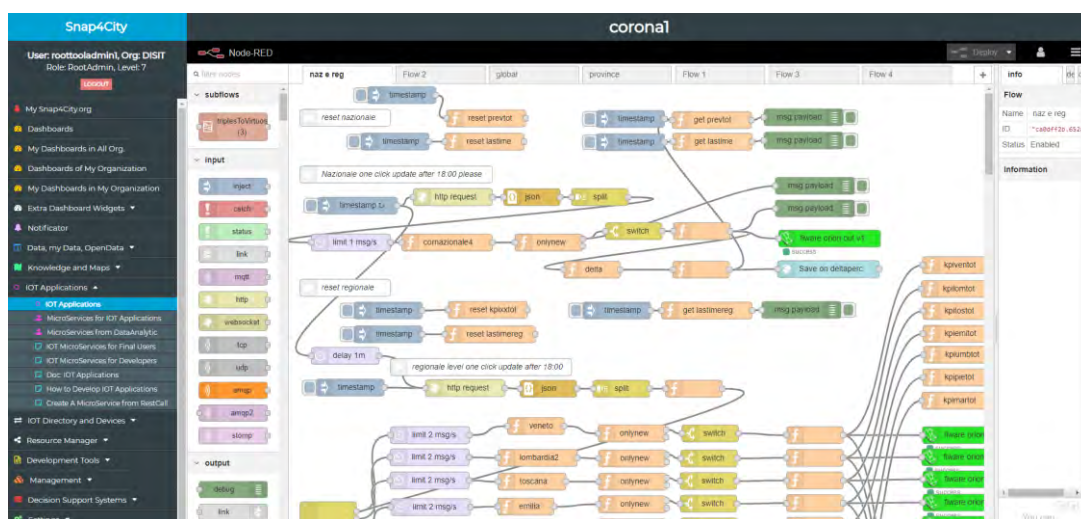
Snap4City platform provides a large set of **MicroServices**, which provide an easy and formalized access to all the Smart City services that are available on-cloud from the platform (including the ones to control a part of the platform itself). They are made available in the Node-RED Node.JS environment to create **IoT App** as visual programming language. Among the **MicroServices**, the **IoT Apps** also need to access such services to allow for

the exploitation of **Data Analytics**, Visual Analytics and Dashboards. The latter two aspects can be employed to create the Graphic User Interface (GUI) of the **IoT Apps**. These tools, orchestrated by the **IoT App** flows, may automatically inform, announce, act and produce alerts and warnings on **IoT Devices**, networks, the user interface, external services, etc., and provide support to close the loop towards the user acting/reacting on the GUI and/or Devices, including notifications.

In the deployment of a Smart City, a number of **IoT Apps** may need to be deployed on the basis of the on-line requests, made by users/organizations. **IoT Apps** are managed and allocated on-demand on the basis of the users requesting them, as well as be executed on-cloud. To this end, an elastic (vertical and horizontal) infrastructure is provided to manage—in a scalable manner—the **IoT Apps** in containers and mechanisms to guarantee end-to-end security [Security2020].

As shown in the next Figure, the user interface allows the user to manage (upgrade, restart, delete) its own **IoT Apps** and **Containers** irrespective of whether they are on-cloud IoT Apps or IoT Edge/field Apps, whether they are child processes/containers for **Data Analytics** or **WebScraping**, etc. Please note that different kinds of **IoT Apps** are represented by different icons.

The Snap4City **MicroServices** allows at the users to develop **IoT Apps** in a **standard** well diffuse **visual programming environment** without being strong programmers hiding the complexity of the sophisticated algorithms and tools:
- data ingestion flows (exploiting connectors, implementing integrations, adapters, etc.),
- data transformation and integrations flows,
- flows for exploiting Data Analytics, and Web Scraping,
- Business Logic flow behind Dashboards and user interfaces,

On the other hand, skilled programmers can fasten their development, with fast access to all data and functions and with the possibility of inserting code in Python, JavaScript, RStudio, etc.

This approach is useful and suitable, for example, for providing routing, a spatio-temporal search and discovery, for data analytics, dashboarding, networking among IoT Devices, IoT data abstraction, etc. The Snap4City **MicroServices** are distributed into two official libraries of Node-RED nodes by the JS Foundation portal). The two libraries are dedicated to final users (basic) and to developers (advanced). The version dedicated to Users provides outputs of Node-RED nodes that can easily be exploited by non-skilled users on JSON. In fact, most of the output produces single variables and not complex JSON-structured messages. On the other hand, the version for **Developers** (to be installed on top of the basic version for final users) presents a number of nodes/blocks that can accept complex JSON message inputs to create strongly dynamic IoT Apps.

Both Libraries of Snap4City Nodes/MicroServices https://flows.nodered.org/search?term=snap4city can be directly installed in any Node-RED tool on any operating system: Linux, Raspberry Pi, Windows, etc. In addition, we have also developed an Android App that executes Node.JS/Node-RED and our libraries to allow for the use of them on IoT Edge and also exploit the mobile device sensors on the above-mentioned operating systems. It has been demonstrated how the Snap4City approach may work on mobility and transport applications, where critical

safety communications and solutions have to be set up, involving IoT Networks with IoT Devices and **IoT Edge Devices, IoT Apps** and **Dashboards**.

3.6.1 – Snap4City Library of the MicroServices of Smart Cities

In order to satisfy the smart city requirements, in Snap4City, a collection of more than 150 MicroServices, as Nodes for the Node-RED programming environment, has been developed.

https://flows.nodered.org/search?term=snap4city

The Node-RED philosophy of visual programming allows for the creation of event-driven data flow applications, where the exchanged messages are in JSON format. On the other hand, periodic processes can also be developed by scheduling one or more internal timers. This means that users can develop IoT Apps as Node-RED flows, exploiting both Push and Pull data protocols, in the same visual programming environment. In the context of smart cities, both protocols are needed, while IoT Apps are capable of creating flows and exploiting a large number of features that are typically not available in the Node-RED open library, nor in a number of libraries from different providers. Moreover, the Snap4City **MicroServices** are at a level that can allow even non-expert users to easily develop IoT Applications for smart cities**.**

The most relevant families of nodes/MicroServices for smart cities are listed below, and they perform different kinds of activities, which are useful in the IoT App construction. **Snap4City library of MicroServices in Node-RED** include more than 150 nodes

https://flows.nodered.org/node/node-red-contrib-snap4city-user   and
https://flows.nodered.org/node/node-red-contrib-snap4city-developer

---

The main categories of Snap4City Node-Red Libraries and MicroServices on IoT Apps are:

- **Data ingestion:** more than 70 protocols IOT and Industry 4.0, web Scraping, external services, any protocol database, etc**.**

- **Data access:** save/retrieve data, query search on expert system, georeverse solution, search on expert system Km4City ontology, etc.

- **Data Transformation/transcoding:** binary, hexadecimal, XML, JSON, String, any format, etc.

- **Integration:** CKAN, Web Scraping, FTP, Copernicus satellite, Twitter Vigilance, Workflow OpenMaint, Digital Twin BIMServer, any external service REST Call, etc.

- **Manipulated complex data:** heatmaps, scenarios, typical time trend, multi series, calendar, maps, etc.

- **Access to Smart City Entities and exploitation of Smart City Services:** transport, parking, POI, KPI, personal data, scenarios, etc.

- **Data Analytic:** managing Python native, calling and scheduling Python/Rstudio containers as snap4city microservices, etc. (predictions, anomaly detection, statistics, etc.)

- **User interaction on Dashboard:** get data and message from the user interface, providing messages to the user (form, buttons, switches, animations, selector, maps, etc.)

- **Custom Widgets:** SVG**,** synoptics, animations, dynamic pins on maps, etc.

- **Event management:** Telegram, Twitter, Facebook, SMS, WhatsApp, CAP, etc.

- **Hardware Specific Devices:** Raspberry Pi, Android, Philips, video wall management, etc.

ONLY some example and details are reported in the following.

Access to Smart City Entities, which have different data models and, thus, different MicroServices may be required. Some Entities may have simple sets of Metadata, for example, the ones describing the POI, e.g., the title, description, web, email, GPS location, images, opening time, etc.; others may have complex information and/or specific data types and structures, for example:

- Status of the first aid: number of people under observation for each color in the triage, waiting time, etc.;
- Bus stops: all the bus lines, including the geometry, their planned time schedule, real-time delays, etc.;
- Sensors, along with their values, measurement units, types, healthiness criteria, etc.;
- Weather forecast associated with an area/region, which consists of a large set of values: temperature, humidity, pressure, wind, etc., for many different time slots in advance;
- Shape of cycling paths, gardens, parks, difficulties, restrictions, etc.;
- Parking areas, with the number of free spaces, predictions, typical daily trends in free spaces, costs, etc.;
- Events: (i) entertainment, with their description, photo, start date, end date, etc.; (ii) police officers on the street; (iii) emergency events, such as a civil protection early warning, according to the CAP standard, etc.

In order to simplify this complexity, **MicroServices**, like "*Service Info*" and/or "*Service Info Dev*", are provided for Final Users and Developers, respectively. In the IoT App, a search/Discovery has to be performed, as described in the next paragraph; otherwise, the developer needs to know the so-called **ServiceURI**, which is the unique identifier of all of the city entities in the **Knowledge Base**. In the Snap4City development environment, the

ServiceURIs can be recovered directly from the graphic configuration of the **MicroService** or searched using the ServiceMap visual tool, which provides a GUI for queries. This means that the IoT App programming is 100% visual, even if a single service is used to access a single element.

Search/Discovery of City Entities and their relationships. The search of city data has to allow users to discover data/device values by a semantic search using a composition of the available query types, which are as follows:

- **Semantic classification**. In Snap4City, all the POI and Services are classified into more than 20 classes (mobility, energy, banking, environment, cultural activities, etc.), including a total of more than 520 subclasses (see, for example, the menu of the **ServiceMap** reported in the Figure aside);

- **Geo spatial references**: close to a point, max. distance from a given point, along a path, into an area/polyline;
- **Textual keyword substrings**: for example, on the basis of the title and descriptions of city entities;
- **Value Types**: for example, all sensors measuring the temperature, all bus paths, etc.;
- **Historical time slot**: for example, data values for the last 7 days;
- **Prediction time slot**: for example, data regarding the predicted parking slots for the next 15 min, 30 min, 1 h, etc.;
- etc.
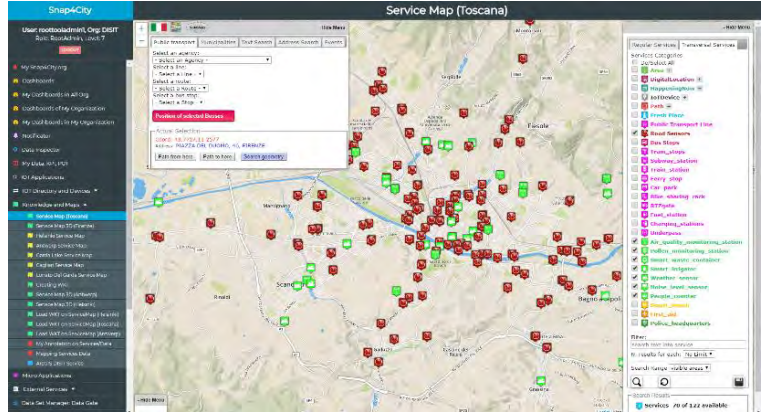
The results of this kind of search can be a single element, with its description, as well as a JSON containing a list of entities. In this latter case, the list can be split into single messages using the Node-RED Split node. Once the list of **ServiceURIs** is accessible, their detailed description can be obtained using the above-mentioned "*Service Info*" node. The search facility for Final Users is provided via the Node interface, simply performed by employing a user interface for setting parameters. On the contrary, as for the nodes for Developers, search parameters can be also prepared and sent to the search Node in JSON. In both cases, the user does not need to know any query language (e.g., SQL or MySQL), and he/she does not need to know if the data are coming from a complex set of queries on SPARQL, for the RDF store, **ElasticSearch** and Phoenix/HBASE. Therefore, the developer of the **IoT App** does not need to know all the tiny details about the large variety of adopted storages. Snap4City provides more than 70 different nodes for searching different Smart City entities, providing results in different data types: POIs, time trends, values, events, schedules of buses, bus lines, recommendations, addresses, routes, etc. This approach significantly simplifies the creation of Smart City Applications.

Discovering and Exploiting IoT Entities (sensors and actuators) should not be different from discovering any smart city entity. In Snap4City **IoT Devices**, data values can be accessed, searched and discovered by the above-presented MicroServices/nodes. On the other hand, when the user/developer would like to create a data-driven IoT App exploiting specific IoT Devices, some specific MicroServices can be used so as to discover the desired devices, regardless of the IoT Broker, IoT protocol or IoT Device. To this end, the **IoT Directory MicroService** exploits the services of the Km4City Knowledge Base for searching, managing and discovering all of the available IoT sensors/actuators. In Snap4City, developers can register on the **IoT Directory**, **IoT Devices** and **IoT Broker**, supporting a large number of different protocols and authentication models. The **IoT Directory** automatically registers new Devices in the **Knowledge Base**, and each new IoT element receives a **ServiceURI**, thus becoming a City Entity and POI. A number of IoT Broker are provided by Snap4City, while others are managed by third parties, for example, to provide an answer to queries, such as "give me all temperature sensors close to my house" (regardless of the data providers, protocol, source, etc.).

Creation of an advanced Graphic User Interface, including graphics widgets, such as Dashboards, Virtual Devices that act on the IoT Apps and IoT Devices, advanced tools, etc. This means giving developers the

possibility to design the IoT App user interface of the IoT App. The user interface has to show and also allow for interactions between users through messages in the IoT Network, including the **IoT App**, which may implement the logic of the user interface. The user interface is built by composing a number of graphics widgets in a connected **Dashboard** to present and collect data.

In Snap4City, the IoT Apps may be directly connected with Dashboards by means of widgets for:

- **Firing events from Dashboard to IO App, and from IoT App to Dashboards**. That is monitoring, controlling and **Acting**.

- **Rendering data**: single content, time trends, bars, histograms, maps, pies, semaphores, dynamic signals, buttons, knobs, clocks, etc.

- **Collecting data** as a switch, a knob or a keypad, which are interactive elements on Dashboards and are represented in IoT Applications as Input Nodes for the flow and are able to provoke events in the IoT App, according to the data-driven approach.

- **Showing MicroApplications and External Services** in a generic iframe widget.

- **Controlling from the IoT App the status and the visualized data into the Dashboards. For example, the IoT App can change dynamically the content of the Dashboards widgets on the basis of the context. If an alarm arrives or is detected on the basis of data in the Smart City Back office, the widgets on dashboards can automatically swap to the context. Also reconfiguring the Video Wall according to the needed alarm and context.**

In addition, other Dashboard Widgets may not have a counterpart in the IoT App and may be directly added to the Dashboard using the **Dashboard Builder**, which can:

- include **Visual Analytics** tools, such as **External Services**, **MicroApplications**, moving object **Trajectories**, **Origin Destination** tools, **Heatmaps**, **Traffic Flows**, **scenarios**, **routings**, maps of any kind, **Orthomaps**, etc.
- **visualize data** from other sources: **IoT Orion Brokers**, **IoT Devices**, data stores, **APIs**, etc.
- include **Virtual IoT Devices** that can send data directly to **IoT Broker**, as an **IoT Device** but from the user interface.

This also means that an **IoT App may be connected to multiple Dashboards**, and a Dashboard may be connected with multiple IoT Devices and IoT Apps.

Data Analytics can be connected via specific MicroServices automatically created by the Snap4City solution or provided by Node-RED.

Snap4City allows to

1) **create** any kind of **Data Analytic** algorithm in RStudio, Python and they are automatically **transformed in APIs and MicroServices** to be directly exploited in IoT Apps (for example developed in R, See Figure aside).

2) **Exploit Data Analytics processes created for the specific Infrastructure, may be in Java or other services accessible via some APIs which can be mapped into a MicroService as well. A REST** call invoking **External Services**, for example, gaining access to Twitter Vigilance API, The Weather Channel API, The Things Network API, etc.

3) **Exploit external Data Analytics Services** via Node-RED nodes (Azure, SAP, Watson, etc.).
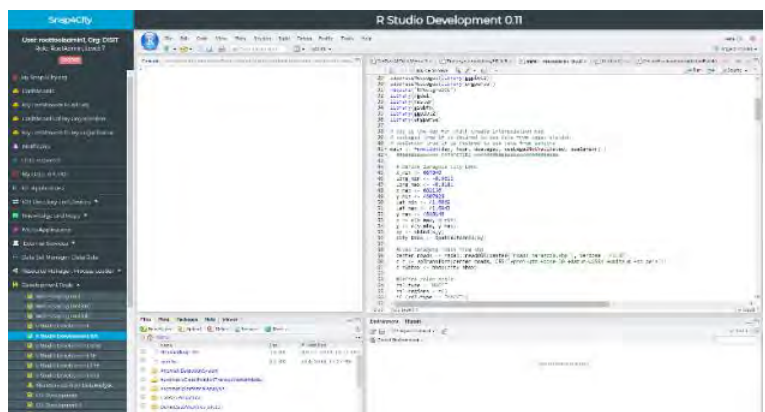
In addition, the **Snap4City**.org portal provides a number of ready to use RStudio examples for, and some of them are ready to use MicroService/nodes implementing:

a. A generic **Data Analytic MicroService,** where Data Analytic algorithms, developed in R-Studio, can be executed, according to specific guidelines.

b. A generic **Data Analytic MicroService,** where Data Analytic algorithms, developed in Python, can be executed, according to specific guidelines

c. **Data Analytics** for computing: routing, auto-ARIMA predictions, anomaly detection, descriptive statistics, machine learning predictions, heatmaps, providing social media data from Twitter Vigilance (not included into the Snap4City suite, but it is also DISIT lab product), etc.

Save and Retrieve Personal Data, for instance, time series for motion tracking, values of personal devices, clicks on Mobile Apps, POI, shapes, KPI (key performance index), Keys to access IoT Device services, etc. The possibility of saving and retrieving data from a safe storage (with the possibility of assigning access delegations according to GDPR) enables a large variety of smart scenarios for the final users and operators, for example, saving personal data from personal health devices (e.g., monitoring glucose), from home statuses, from the location of mobile phones that belong to a specific person. In addition, the Snap4City platform automatically collects all the personal data gathered from the mobile Apps accessed on the Snap4City login (provided that the user authorizes their collection with signed consent, according to GDPR).

Save, retrieve, and publish Data Sets, as those managed in the open data portals. Most public administrations publish their data sets, in the form of open data, and also share their data via federated networks (see, for example, the harvesting mechanisms of CKAN). Having the possibility of creating data sets from the flow of IoT Applications means that city operators can automatize their ingestion/update processes and their production/publication.

In the Snap4City libraries of MicroServices, there is also a number of tools for managing the **DISCES** back office scheduler of processes, as well as for saving LOGS of data accesses and flows. The former can only be accessed by administrators, while the latter can be useful for both developers and administrators. Moreover, a number of so-called GEO-utilities have been identified and implemented in Snap4City, such as MicroServices/nodes to:

- Calculate the **distance from two GPS** locations.
- Verify if a GPS location is close to a shape/area, polyline
- Verify if a GPS location is inside the area or not in order to, for instance, verify the match with administrative areas and thus determine whether your dog is in the garden or the monitored bear is in the forest, etc.
- Obtain the most probable value of a variable represented in a Heatmap from any GPS point, irrespective of the presence or absence of a sensor at that point. This feature is very useful in choosing among different routes: the quietest one, the less polluted one, the one that has the least traffic, the busiest road for meeting people, etc.
- Obtain the closest civic number and street from a GPS point or vice-versa. This is very useful for geo reversing.
- Obtain the closest road map segment (typically called a node in Open Street Map language) from a GPS point or vice-versa. This can be useful for routing and computing precise distances, and it is very useful for geo reversing.

This means that IoT App developers do not need to solve the so-called direct or inverse georeferencing problems, since they are provided by default.

A number of Node-RED/MicroServices that are useful in smart city contexts are also available: they have not been implemented by us, as they are provided by third parties, for instance, interaction via Facebook social media, SMS, email, Telegram, etc.

Snap4City is capable to interoperate since it can import and export data in multiple formats and protocols mainly via **IoT Apps**.

IoT Apps can be used for creating flows/integrations for: data ingestion, connectors, data transformation, integrations, exploiting Data Analytics, exploiting Web Scraping, data store access, BPM vs Data flow integration, and for implementing Business Logic flow behind Dashboards and user interfaces.

Snap4City is strongly interoperable since it can:

- Acquire and send data from/to **several sources (any protocol and format)**. Snap4City supports a very **large number of protocols (and Connectors, adapters)** for connecting (collecting data and for exchanging data) with other services, database and tools: https://www.snap4city.org/65

  o AMQP, COAP, MQTT, OneM2M, HTTP, HTTPS, TLS, Rest Call, SMTP, TCP, UDP, NGSI, LoRa, LoRaWan, TheThingsNetwork, SigFOX, DATEX II, SOAP, WSDL, Twitter, FaceBook, Telegram, SMS, OLAP, MySQL, Mongo, HBASE, SOLR, SPARQL, EMAIL, FTP, FTPS, WebSocket, WebSocket Secure, ModBUS, OPC, OPC-UA, GML, RS485, RS232, WFS, WMS, ODBC, JDBC, Elastic Search, Phoenix, XML, JSON, CSV, db, GeoJSON, Enfuser FMI, Local File System, ESP32, Libelium, IBIMET/IBE, OBD2, SVG, XLS, XLSX, TXT, HTML, CSS, KNX, Enocean, Zigbee, DALI, ISEMC, Alexa, Sonoff, HUE Philips, Tplink, BACnet, TALQ, IFC, Copernicus, DCAT, XPDL, etc.)

  o

- Access and exploit data coming from **Copernicus** satellite system.
- Include **External Services as API Rest Calls:** They can be made accessible into Snap4City IoT Apps as MicroServices. The user can register them via a simple interface. A List of the available MicroServices from External Services is also provided into the **Resource Manager**.
- Exploit **External Services Web Applications** (distinguished for organization of the registered user), exploiting them on Dashboard and MicroServices. For External Services we intend Web based Tools that may expose web pages on which one can interact: a weather portal, a specific view on traffic, the status of the port, etc. This means that the Snap4City Platform can integrate in Dashboards External Services and Tools to produce integrated views.
- **Work with top 10 used DBs** and more, as: Mongo, Azure, MySQL, Microsoft SQL, MariaDB, As400, Cassandra, Elastic Search, Oracle, IBM DB2, Postgres, AWS, etc. and via standard connectors such as: ODBC, JDBC, etc. via IoT App.
- **Join, publish and harvest Open Data Portals** via DataGate/CKAN.
- **Interoperate with ERP via IoT App:** https://flows.nodered.org/search?term=ERP
- **Interoperate with BPM and BIM via OpenMaint:** https://www.openMAINT.org/en/home and see https://www.snap4city.org/625
- Acquire data from **WEB Portals of third parties**. The approach of data scraping/crawling from Web Portals is based on an integrated Web Scraper based on Portia. https://www.snap4city.org/478
- **Federate Snap4City Knowledge Bases**. This allows the creation of mobile applications that may move from multiple cities and area transparently. This solution is presently in place among the Knowledge Bases of: Antwerp/Helsinki, Tuscany/Firenze, Sardegna, etc. The resulting Service is called Super Service Map and it is integrated in the Smart City API. **See Section 3.8.**
- Integrate/command **ISEMC for Video Wall** management, for implementing Map Command and Control Center, Smart City Control Room: (https://www.snap4city.org/621) Control Room as: https://www.snap4city.org/531
- **Interoperate with ArcGIS ESRI via API and protocols such as WFS, WMS, etc. https://www.snap4city.org/368 It is also integrated with ArcGIS ESRI via IoT App and Node-RED:**

https://flows.nodered.org/search?term=arcgis

- Exploit **external Maps via WMS protocols (among them also MAP provided for free** from **ESRI)** https://www.snap4city.org/368 **to show them directly into** Dashboards**, see for example:** https://www.snap4city.org/368 and Exploit **WFS sources to show them directly on Dashboards:** https://www.snap4city.org/368
- Exploit **WFS sources to ingest them into the Snap4City storage via IoT Apps:** https://www.snap4city.org/368

Snap4City is capable to interoperate since it can Expose its data and services via:

- **Smart City API which can be confined into a single Smart City installation or Federated as well as for Super ServiceMap WFS service of Snap4City** on top of Federated Smart City API or simple Smart City API of a single ServiceMap (smart City installation). This solution permits to GIS applications and platforms (such as ArcGIS OnLine ESRI, ArcGIS Enterprise ESRI, ArcGIS Map/pro Desktop, QGIS, GeoServer, etc.) to access at Snap4City data. See interoperability with ArcGIS and GIS in general https://www.snap4city.org/368
- **WMS service of Snap4City** for publishing maps and heatmaps, provided by an installed GIS (GeoServer or ArcGIS) third party open source tool.
- **Federated Knowledge Bases, Multiple Snap4City Smart Cities connected via their Knowledge Base, Smart City APIs**. This allows the creation of mobile applications that may move from multiple cities and area accessing data and making queries transparently. This solution is presently in place among the Knowledge Bases of: Antwerp/Helsinki, Tuscany/Firenze, Sardegna, etc. The resulting Service is called **SuperServiceMap** and it is integrated in the Smart City API. **See Section 3.8** and [FederatedKnowledgeBase2020].
- **Federated Open Data Portals** via DataGate/CKAN that presently presents now more than 13800 data sets linked for the cities of Helsinki and Antwerp with **DCAT protocol**. See for example: https://doc.arcgis.com/en/hub/content/federate-data-with-external-catalogs.htm
- **Interoperability with Heatmaps and WMS** Snap4City is working with Heatmaps and Maps in WMS standard. Heatmaps are saved in terms of points in the Heatmap manger server and automatically produced on the basis of the set colormap, the maps/heatmaps are provided to dashboards in standard WMS by means of the Snap4City **Heatmap Server** which includes a reference to a GIS (GeoServer or ArcGSI) and tools for automated generation of GeoTiff images: https://www.snap4city.org/368

3.8 – Snap4City Federation of Knowledge Base and Smart Cities

The classic GIS interoperability is limited to 1:1 exchange of geographical data for example exploiting protocols as WFS (Web Feature Service), WMS (Web Map Service), for the exchange of Maps and geoelements such as paths, points of interest, road elements, road graphs, etc. GIS protocols, according to their definition, they are typically unsuitable as API for providing data related to smart services, such as smart parking, subscription on alerts on environmental conditions, etc., as needed for smart city applications. Most of vertical Smart City API based solutions are typically focused on a limited range of data by using SQL databases which in turn can provide support for geolocated information and may be federated at level of the database.

*Snap4City solutions can be federated each other, creating a cross-city environments among cities, regions, and in the same city among different areas if needed. The federation is at level of smart city API, and this allows to mobile Apps to pass from one city/area to another without loss of continuity. This feature could be performed to **federate your Cities/Areas or applications**. Please note that, Snap4City is 100% open source, so that the solution provided can be easily replicated without costs of licensing.*

Federated Knowledge Bases, Multiple Snap4City solutions connected via their Knowledge Base (and/or connecting other IoT Orion Brokers via SSM2ORION which do not are integrated into a Snap4City solution), Smart City APIs allows the creation of mobile applications that may move from multiple cities and area accessing data and making queries transparently. This solution is presently in place among the Knowledge Bases of: Antwerp/Helsinki, Tuscany/Firenze, Sardegna, etc. The resulting Service is called **SuperServiceMap** and it is integrated in the Smart City API. [FederatedKnowledgeBase2020].

A distributed Smart City API based solution for a set of cities, let say a federated network of Smart City APIs may provide a number of advantages.

1) support **distributed search** on the **Federated Knowledge Bases** network;
2) support **connection with smart city of any size** in terms of number and volume of data sets providing services of the nodes. In addition, the geospatial size and shape of each node may be: (i) not regular (nor a circle but a shape), and multiple connected (so called multi-polygon), (ii) partially overlapped with other nodes, (iii) totally included into those of other nodes, (iv) disjoined and even far each other (this means that the union of all the areas can be disjoined with respect to the global map of the earth);
3) Support nodes with a **different number of services available**. This implies that not all kinds of services and data may be necessarily available in all **Federated Knowledge Bases**;
4) Support nodes with **georeferenced services or not.** This means that are general for the area addressed and not specifically related to the GPS position;
5) respond to API calls in terms of services in **transparent manner passing from one node to another** or when the service needs to provide results coming from more **Federated Knowledge Bases**;
6) support access control to **prevent access to data and services by not authorised users**. Since the passage of a user from one **Knowledge Base** to another of the **Federated Knowledge Base** network may imply the sending of requests which may try to access at private data/services;
7) support the **addition/removal of Federated Knowledge Bases in the network** without the need of fully restructuring of the network and modifications have an immediate effect without any service reloading or disruption;
8) provide **results in real time also when a large number of Federated Knowledge Bases/areas are involved**. The implementation should also provide support for creating redundant solutions with high resilience;
9) provide the **response in the coherent format** with the expected response of the single services. Thus, the results of the federation may need to be merged to produce the response in any format: JSON, XML or HTML.

In order to avoid having a single point of failure, the **Federated Knowledge Bases** Super can be replicated into each node and the list of Super services on top of **ServiceMap** can be put accessible in one or more web servers for update. Each **ServiceMap** has a representation of the multi-polygon addressed by the nodes (with their data/services) and thus of their partitioning over the nodes of the **Federated Knowledge Bases** network. In more details, each **Federated Knowledge Bases** may register in the Super network the descriptor of the multi-polygon area of your competence. This approach permits at the **Supers** to redirect the queries to the nodes that could provide the service and data. Thus, the **Supers** do not need to hold the data of the nodes and perform the distribution of queries only to the involved nodes, to finally collect the results and performs data fusion. The **Supers as well as the ServiceMaps** may also implement some query caching solution as all the other **Federated Knowledge Bases**.
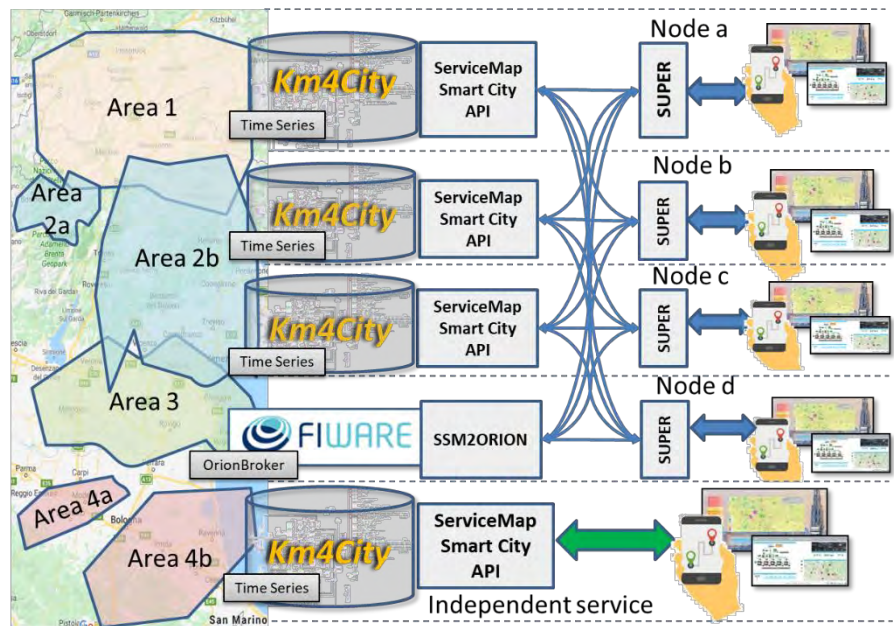
Federated ServiceMap and Smart City API

To improve scalability, fault tolerance and federation among cities:

- One entry point Smart City API for all zones
- Multiple Knowledge base See performance assessment

At different levels:

- Among cities/regions
- Among data providers, Operators

By Means of:

- Smart City API → Apps
- Smart City Ontology
- Dashboards/data analytics
- Organization independent
- CKAN via harvesting

Snap4City (C), November 2020

3.9 – Snap4City Data Analytics Development and Exploitation

For the data analytic development, it is possible to access the Big Data store respecting the privacy and the data licensing by using authenticated Smart City APIs. The access has to permit to read historical and real time data, and to save the resulting data provided by the algorithms, for example, heatmap-related predictions, the assessment of data quality, and labels of detected anomalies. The work of the data scientist could finish once the algorithm has been developed which he/she should be aware of. On the other hand, the same algorithm (e.g., for computing heatmaps, parking prediction), should allow to be:

- Used on different services of the same kind located in different places and on the basis of a number of parameters (e.g., target precision and list of data sources). This means that data analytic itself has to be designed with the needed flexibility and generality;
- put in execution from IoT Applications by passing a set of parameters and collecting the results on the Data Storage or as a result of the invocation. The executions can be periodic or event driven —e.g., the arrival of a request or by the arrival of the new set of data values;
- controlled for collecting eventual errors and mistakes, in debug and at run time for logging. This may be for informing the developer and/or the administrator of eventual mistakes and problems by sending notifications; and
- dynamically allocated on cloud in one or multiple instances to plan a massive computation of the same data analytic process on several data sets and services at the same time.

Therefore, the solution developed for Snap4City satisfies all the above described requirements. Data analytic processes can be developed using **R Studio** or **Python**. In both cases, the code has to include a library for creating a REST Call, namely: Plumber for R Studio and Flask for Python. In this manner, each process presents a specific API, which is accessible from an IoT Application as a MicroService, that is, a node of the above-mentioned Node-RED visual programming tool for data flow.
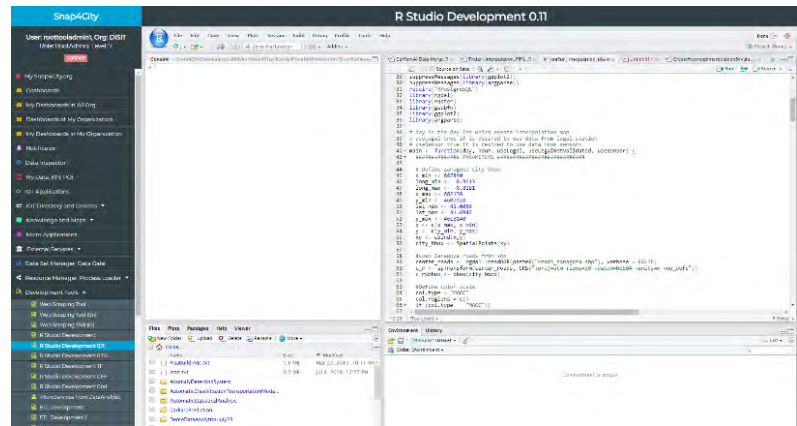
Data scientists can develop and debug/test the data analytic processes on Snap4City cloud environment since it is the only way to access at the Smart City API with the needed permissions. The source code can be shared among developers with the tool "Resource Manager", that also allows the developers to perform queries and retrieve source code made available by other developers.

3.9.1 – Data Analytics developed in the Snap4City

To give you an idea of the Data Analytics we have developed we suggest you to browse the training course: https://www.snap4city.org/download/video/course2020/da/

Data analytics have to be targeted with respect to the user kind and the device in which they are exploited. So that, in Snap4City platform you can find, examples and tools to comply with those aspects. The following examples and those reported in the training course and can give you an idea of the capability of the platform.

The following image report the mapping of the Data Analytics vs the Antwerp and Helsinki users and the main tools in which they have been made accessible in transparent manner for the user.

	Antwerp					Helsinki								Where				Main Data Sources
	City official	ICT official	Developer	Citizen, tourist, visitor	Business owner	City officials	City officials Domain experts	City officials City developers	Third party developers	Citizen	Citizens with respiratory problems	Tourists	Business owners	Mobile	MIcroApplication	Tool, via Portal (ICT Developers)	Dashboards	
Discovery near to me	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			POI, OSM
Discovery along a path	X	X	X	X		X		X	X	X	X	X		X	X			POI, OSM
Discovery in an area, shape	X	X	X	X	x	X	X	X	X	X	X	X	x	X		X		POI, OSM
browsing Public Transport	X	X	X	X	x	X	X	X	X	X	X	X	x	X	X			OSM, GTFS
Full Text search	X	X	X	X	X	X		X	X	X	X	X	X	X		X		POI, OSM
Routing: pedestrian				X	x			X	X	X	X	X	x	X	X			OSM
Routing: pedestrian quite				X	x			X	X	X	X	X	x	X	X			OSM
Routing: private vehicles	X		X	X		X		X	X	X	X			X	X			OSM
Routing: Multimodal Public Transport				X				X	X	X	X			X	X	X		OSM, GTFS
heatmaps: weather (Temp, Humidity)	X	X		X	X	X	X		X	X	X	X	X	X			X	Sensors data, OSM
heatmaps: environmental variables, PM10, PM2.5, NO2, EAQI	X	X		X	X	X	X		X	X	X	X	X	X			X	Sensors data, OSM
heatmaps: environmental variables, Noise						X	X		X	X	X	X	X	X			X	Sensors data, OSM
heatmaps: safe on bike (Antwerp)	X	X		X	X									X			X	Spec. Portal
heatmaps: Enfuser prediction, PM10, PM2.5, AQI						X	X		X	X	X	X	X	X			X	Enfuser data
heatmaps piking values any place	X	X			X	X	X	X	X				X				X	Computed Heatmps
heatmaps: GRAL prediction, PM10						X	X		X	X	X	X	X	X			X	OSM, Traffic, Weather
Comparsion: Enfuser, Gral, Real Time						X	X										X	Enfuser, Sensors, GRAL
Sensors Data Time Trends, & drill down	X	X	X		X	X	X	X					X			X	X	Sensors data, OSM
Weather Forecast	X	X		X	X	X	X		X	X	X	X	X	X			X	Forecast Service
Origin Destination Matrices	X	X	X		X	X	X	X	X				X				X	Snap4City Mobile App
Typical trajectories	X	X	X	X	X	X	X	X	X				X			X	X	Snap4City Mobile App
Hot Area in the city	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	Snap4City Mobile App
Hot Places in Smart Zone	X	X	X	X	X									X		X	X	Snap4City PAXcounters
Services Suggestions on mobiles				X						X	X	X		X	X			Snap4City Mobile App
Alerts on critical cases: several variables	X		X	X	X	X	X			X	X		X	X				Sensors data, OSM
The most used services		X		X	X					X	X	X	X				X	Snap4City Mobile App
Twitter Trends Daily	X	X	X		X	X	X	X	X				X			X	X	Twitter Vigilance
The auditing of user and living lab		X				X		X									X	Snap4City Portal
Self assessment	X	X	X	X	X	X	X	X	X	X	X	X	X			X		Snap4City Portal
Trajectories reg from mobile PAX Counters	X	X	X			X	X	X								X	X	PAX Counters
Engagement real time assessment	X	X	X			X	X	X									X	Snap4City Mobile App

Other examples of Data Analytics could be:

- Mobility and transport
    - Traffic flow reconstruction from sensors and other sources
    - Predictions for: traffic flow, smart parking, smart bike sharing, etc.
    - Analysis of the demand vs offer of mobility according to public transportation and multiple data sources
    - Accidents heatmaps
    - Tracking fleets, people, via devices: OBU, OBD2, mobile apps, etc.
    - Routing and multimodal routing
    - Bike sharing predictions
- Environment and weather
    - NOX, PM10 pollution prediction on the basis of traffic flow, 48 hours
    - Long term prediction of European Commission KPIs on NOX, PM10, etc.
    - Heatmaps production, dense data interpolation
- Resilience
    - Resilience and risk analysis
    - Early warning computation
    - What-if analysis, dynamic routing, origin destination matrices production from a large range of sources
- User and Social
    - People flows prediction and reconstruction, via Wi-Fi, mobile apps, etc.

- User engagement for sustainable mobility
- User's behaviour analysis, origin destination matrices, hot places, time schedule, Recency and frequency, permanence, trajectories, etc.
- People flow analysis from PAX Counters
- Social media analysis on specific channel, specific keywords: see Twitter Vigilance, for NLP and Sentiment Analysis, SA
- Tweet proneness, retweet-ability of tweets, impact guessing
- Audience prediction to TV channels and physical events
- Generic
  - Data quality assessment, prediction, anomaly detection
  - Maintenance prediction and costs predictions
  - Estimation of KPI and local indexes for: quality of life, 15 minutes, etc.
  - COVID-19: user flow, user density, Rt estimation, etc.
- Etc.

3.9.2 – What-IF Analysis Support

The What-IF analysis is a modality of work recently included in the Snap4City suite which aim to exploit in a multicontextual environment most of the former tools as dashboards and Data Analytic with IoT App intelligence.

What-if analysis Support in Snap4City is not a single tool but a modality of work which enables you to create several solutions for city analysis depending on your needs and context. A tool of What-IF analysis has to be capable to answer at questions such as:

What is going to happen at certain City Services if certain conditions/cases are going to occur?

The What-if analysis process is following a number of steps:

- **Create/save/share a Scenarios** on the basis of present or future Conditions/cases, Services, aiming at

- **Produce/save/share a Studio** (which is the actual What-IF analysis) on top of the Scenario. A studio can be a solution, automatically or manually produced with the support of some Data Analytics and Simulation tool

An example in the context of mobility is accessible on:
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjE5MA==

In the previous section the Data Analytics have been reported. In addition to them also the Simulations can come from:

- Traffic Flow reconstruction with conditions and restrictions
- Predictions with conditions and restrictions: traffic, parking, Nox, people flow, ..
- Routing with conditions and restrictions
- Match from offer and demand of public transportation
- simulations via **SUMO** tool
- etc.

The tools and KPI identified can be tracked and exploited in decision support systems such as ResilienceDS and SmartDS (https://www.snap4city.org/520).

3.9.3 – An example of Data Analytics integrated with IoT Apps

In Snap4City solution, the heatmap production can be automatized implementing an IoT App. The heatmaps of different kinds can be produced on the basis of any kind of geolocated data. This feature is present in the **Heatmap Server**/service. https://www.snap4city.org/457 ,   https://www.snap4city.org/641

The automatization of heatmaps production depicted in the following Figure reports an IoT App composed of 4 different blocks. The first block named "Set R function Parameters", is the inject node to insert the R parameters (listed below the block in the figure) in JSON (JavaScript Object Notation) format. The rounded arrow on the node shows that on that node it is possible to send the JSON created at certain frequencies chosen during configuration and modifiable at any time. The green block named "Customized Heatmap" is the Plumber-Data-analytic node to upload the R script and create a plumber instance. Note that, the Plumber-Data-analytic block return specific errors if the settings are not suitable and/or if the process for heatmap computing is already running. The orange node is a function node where it is possible to manipulate the response JSON coming from the node running the R script, since it is a JSON with the results or an error generated when creating the heatmap. The function that manipulates the JSON must be written in JavaScript and it allows to create HTML code to be displayed inside the dashboard of the Snap4city platform. The visualization can be computed using the blue node (single content node) that allows to create a widget inside a dashboard with the HTML content previously created with the function node.
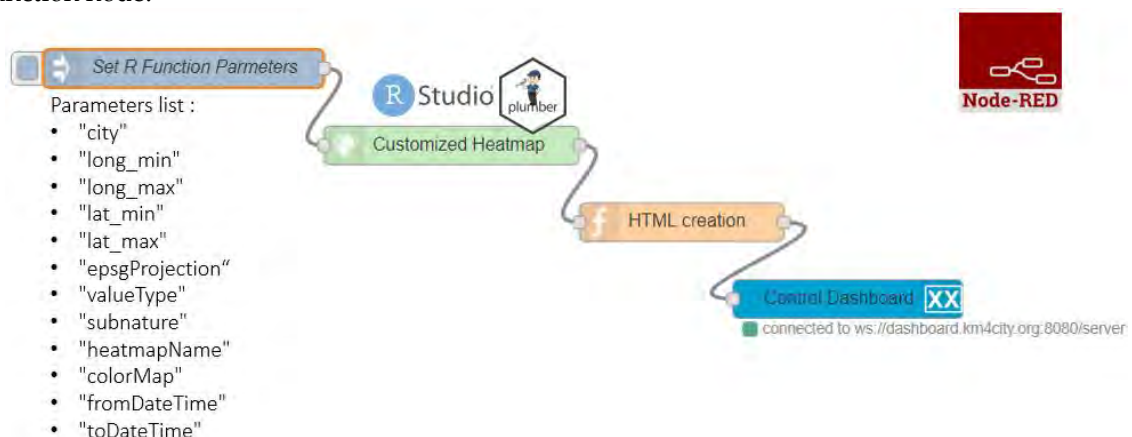
IoT App/Node-RED workflow for automatization of heatmap production.

To create a new customized heatmap from the source code, the principal needed/parameter are:

- The GSP coordinates of the area of interest (min/max latitude, min/max longitudes, etc.);

- The presence of a number of measured values (a set of sensors providing pollution values located in a given area, more than 5 sensors) in a specific time slot determined by a start-date and an end-date parameter in timestamp format. Each sensor has to be identified by a Service URI in the Snap4City Knowledge Base;
- The identification of a colormap that corresponds to the considered pollutant. If a suitable colormap is not available it is possible to create a new one by using the colormap manager positioned in the resource management tool;
- EPSG (Geodetic Parameter Dataset) projection that depends on the location in which the heatmap must be estimated;
- The value type, that is a single name or vector of possible names for the sensor's attributes/pollutant on which compute the interpolation;
- The sub nature of the sensor, that is a single name that corresponds to the nature/type of the sensor of interest; and
- The heatmap name and city of interest. The name of these parameters can be chosen directly by the user.

The production of the heatmap data is only the first step of the process since the heatmaps have to be transformed from a mere grid of points to GeoTIFF according to specific color map adopted for the rendering. This means that the ColorMap has to be available for the GeoTIFF creation. To this end, in Snap4City, a specific ColorMap editor and service has been created and it is exploited by an automated process that transforms the heatmap points into GeoTIFF: The so-called "heatmap GeoTIFF Generator". The heatmap can be very large and at high resolution (millions or billions of areas), the distribution of the maps towards Dashboards and Mobile App is very efficient since it is performed by using a GIS (ArcGIS or GeoServer) according to WMS protocol. The produced Heatmaps can be animated if more than one instance per day is produced.

In addition, to automatize the process of the creation and use of heatmaps, there are also the following services:
- **Piking the Heatmap value** in any GPS point disregarding of the scattered distribution of sensors, exploiting interpolations (Akima, IDW, etc.) with a specific Smart City API and MicroService.
- **Alerting Assessment** has been implemented by using the **Snap4City Engager** tool that, every time a GPS location is requested by the Mobile and Web App, is querying the "On demand fine interpolation";
- **Conditioned Routing** has been realized by using an open source router which also queried the same "On demand fine interpolation" service via API, to get the values in specific segment of the possible routes; and
- tool capable to provide the produced **Heatmaps as GeoTIFF** according to tiled exposed in the frame of the Web and Mobile Pages, which are sent to a GIS (ArcGIS or GeoServer) for their distribution to the Apps and to Dashboard via WMS standard protocol.

3.10 – Snap4City Contextual Dashboards, Business Intelligence, and Control Rooms

A **dashboard is substantially a Decision Support System** tool, since it provides evidence of critical conditions, and may offer solutions. On this regard, it may integrate/exploit artificial intelligence algorithms, for example, reporting prediction, identifying anomalies, manifesting early warning, providing relationships among entities exploiting inference geospatial reasoning about what is located in the city: resources, structure, people, areas, critical infrastructures, etc. In the development of a Smart City there is a great emphasis to have a number of different visual rendering tools based on Dashboards, and in particular to satisfy needs of smart city platform **regular users, developers, city operators, operative dashboards for the verticals, data analysis dashboards, decision makers, business intelligence dashboards, up to the system of dashboards to be shown in the Smart City Control Room and in the situation rooms**. [Industry4.0-2020], [Dashboards2019], [DashboardProduction2020].

Snap4City platform allows creating *Dashboards* of any kind which are interactive data & graphics applications including maps, trends, series, tables, gauge, pie, button, dimers, semaphores, time ranges, what-if analysis features, and a large set of special and custom graphic *Widgets* for representing city elements and acting on the city processes: weather forecast, social media and sentiment analysis, traffic flow, event reporting, event selectors, real time vehicle tracking, decision support suggestions, actuators, KPI (key performance indicator), POI (point of interest), IOT/IOE (internet of thing/ everything), trajectories, heatmaps, TV cameras, dynamic routing, etc. Control room with video wall: https://www.snap4city.org/621
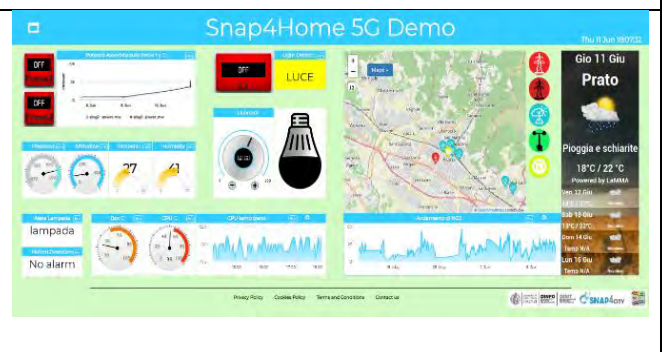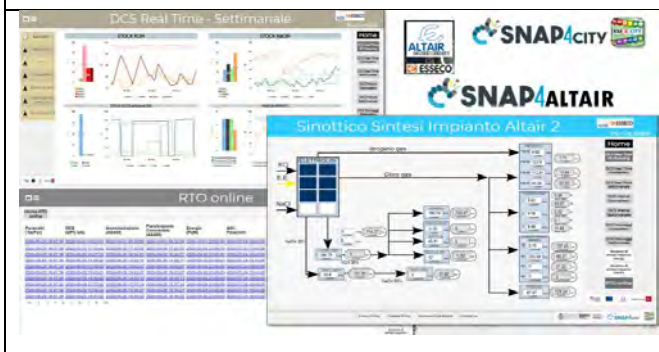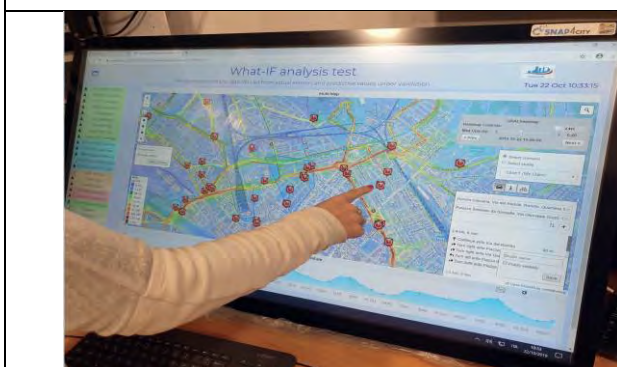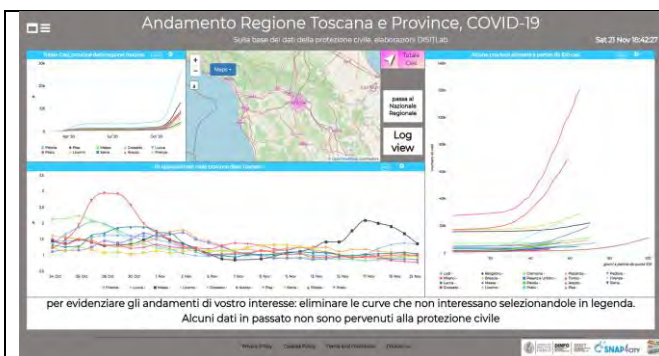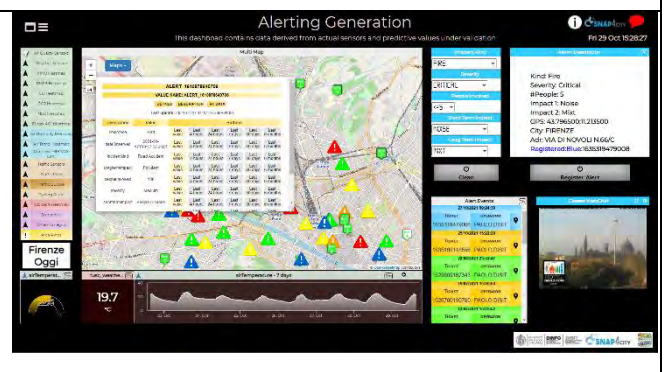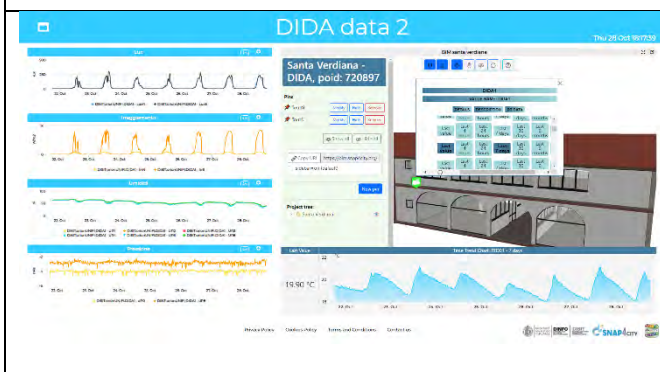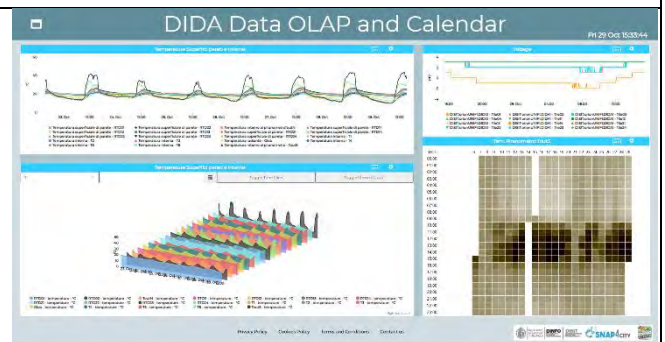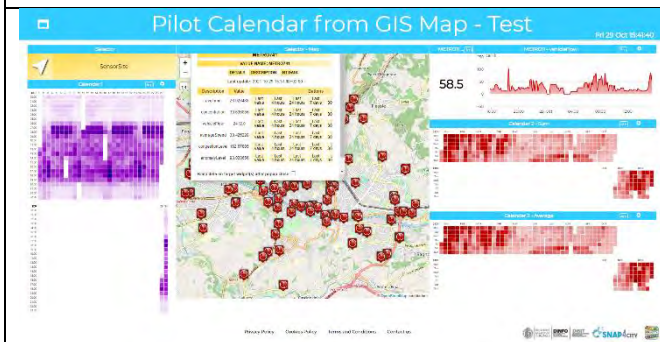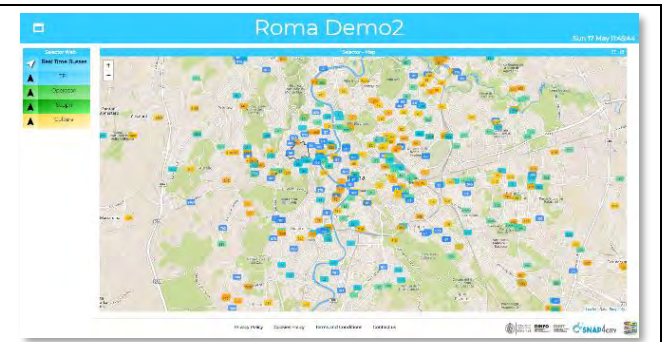
My Data Dashboard (DevDash) Kibana for all kind of users

Snap4City provides two main modalities of creating dashboards, using:

- **Dashboard Builder of Snap4City**
  - For accessing and browsing data on: Elastic Search, Mongo, MySQL, Smart City API, Super and thus from federated Smart City API, etc.
  - Supports sensors/actuators: data driven data, maps in extended manner, data driven widgets, large collection of widgets, direct IoT Connections, custom widgets, animated PIN on maps, a large set of panel/widgets, etc.
  - Very simple to be used for control room, decision makers, situation rooms, operators, etc.
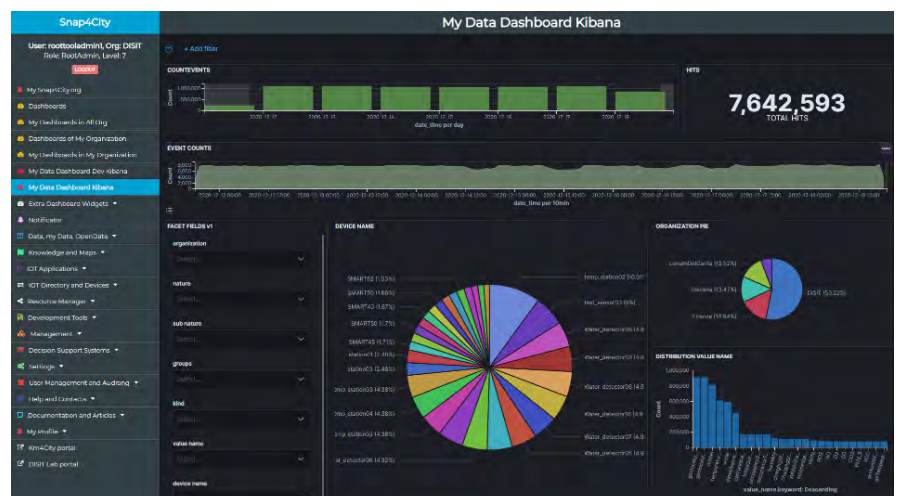
- Very well integrated with IoT App, Custom widgets, animation, external services.
- Very simple to be customized for non programmers since all the tools are visual.
- Support for GDPR and deep control of access.

- **Kibana (so called DevDash, AMMA and recently My Dashboard (Dev) Kibana), also accessible as Grafana**
  - For accessing and browsing data on Elastic Search storage and other sources supported
  - No Support for real time event driven widgets/panels, actuators and synoptics, no sophisticated maps, etc.
  - Not simple for control room, decision makers, etc.
  - Not integrated with IoT App, Custom widgets, animation, external services.
  - Oriented to developers, complex production of custom views, etc.
  - Partial support of GDPR and deep control of access.

3.10.1 – Snap4City Dashboard Builder

Snap4City **Dashboard Builder** enables users of any role in the platform to create and manipulate their own Dashboards of any kind, from mere data visualization tools, to business intelligence tools, to control room dashboards and interactive systems and synoptics. See details on dashboard from the training course part 2: https://www.snap4city.org/download/video/course2020/das/ [Industry4.0-2020], [Dashboards2019], [DashboardProduction2020].

Dashboard Builder allows to created Dashboards:
- **manually** by using a large set of graphic and integrated **widgets** that can be customized in deep for: data source, size, colors, shape, font, strings UTF8 also Arabic, etc.; A list of them is reported in the following. You can edit them manually to change the tiny details of the graphics.
- **automatically** by means of the **Dashboard Wizard** to simplify the production and the connection of data to graphics widget according to models/templates. Once creates, they can be manually edited for defining and changing all detailed aspects.

In the above figure, Dashboard Builder visual editor (left) and **Dashboard Wizard** (next) can be used on any Dashboard to edit and/or add new set of Widgets thus accelerating any productions and tuning.

Dashboards may be single or connected each other. They are typically not a single view, but a view of a set. From a main Dashboard you may need to jump to other views/Dashboards. See the following example of the **Smart City Control Room, SCCR**, of **Florence Metropolitan City** which has more than 1.5 million of inhabitants, and >14 million of Tourists per year, plus students and commuters. The figure reports the main dashboards used by the Florence Mayor (Dr. Dario Nardella) and the second level dashboards. Please note that a third and
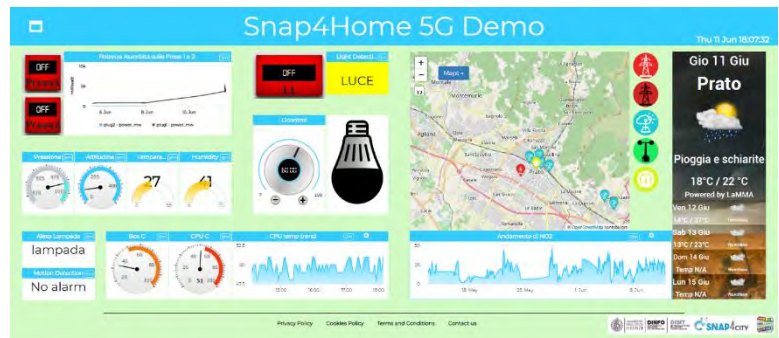
fourth levels are present as well. https://www.snap4city.org/525

Control room with control video wall: https://www.snap4city.org/621

Florence Control Room: https://www.snap4city.org/531
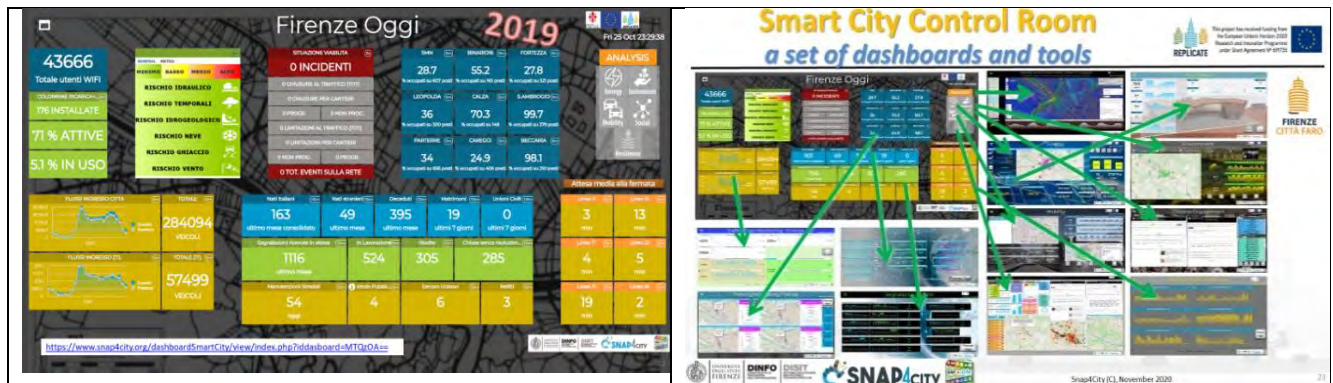
Dashboards are based on a large set of Original Data, derived data and elaborations which are produced in real time for the decision makers and officials. The City Dashboards are typically used and shared with a number of city operators such as fire brigade, 118, civil protection, police, operators, leaders, decision makers, etc., according to a controlled and secure connection on HTTPS, secure WebSocket and GDPR compliant environment. They allow representing and managing critical events, receiving notifications, drill down on data, opening live chats for problem solving, acting in correspondence of alarms by an intelligent monitoring, defining workflows, performing simulation and **What-IF analysis**;
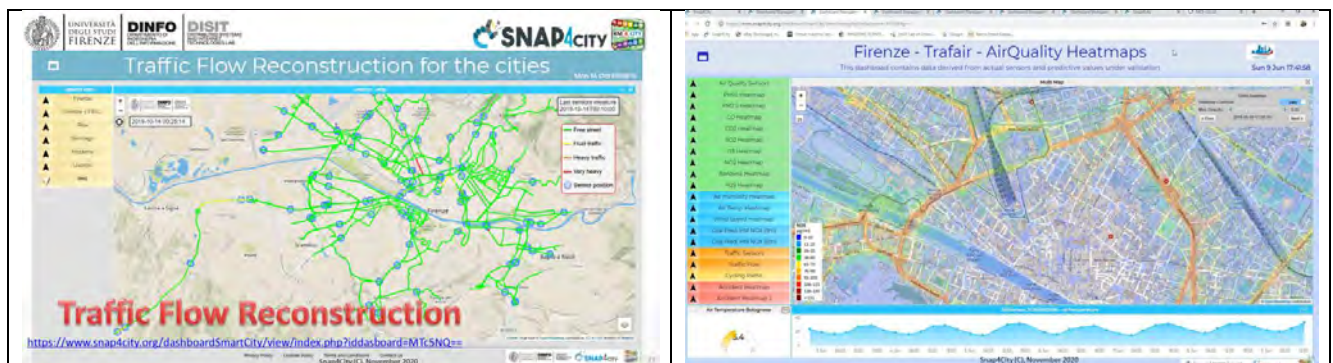
Snap4City Dashboard system is capable to:

- show dashboards on web browser in a H24/7 modality;
- show data on widgets according to several graphic paradigms (tables, graphs, histograms, maps, Kiviat, lists, tv camera, heatmaps, weather, critical city events, etc.) with a level of interactivity and animation;
- **use Arabic language and Hijri date format and conversions;**
- show data on autonomous and connected/ synchronized widgets;
- collect, show and keep update on screen data with automated refresh for each view, and real time data according to the even driven paradigm;
- show data both real time and historical, allowing the drill down on time, space and relationships among data and city entities;
- provide a solution to have dashboards connected with IoT App and platform by using an end-2-end secure connections based on Web Socker secure, TLS, HTTPS;
- collect and show data coming from different big data and classic data sources (SQL, NoSQL, RDF, P2P, API, SOLR, Elastic Search, etc.) also in aggregated manner;
- allow to customize Widgets as Synoptics and also exploiting graphic libraries;

- work with large amount of data providing high performances, as short response time;
- compute alarms, and provide support by a flexible notification system capable to send alerts, activate tickets for maintenance, automate actuators, post on social, etc.;
- provide actuators widgets together with showing graphs, and capable to act on IoT Devices;
- provide support for collaborative production of dashboards and for co-working;
- provide support for embedding dashboards into third party web pages;
- provide data engine for collecting connection response time on different protocols, and for verifying the consistency of web pages via HTTPS;
- allow to clone dashboards;
- allow to give the access to the dashboards to other users;
- integrating with IoT Apps by managing real time data and connecting its actuators to real time IoT Apps, thus having in IoT App the Business logic of one or more dashboards;
- integrate dashboards in more complex dashboard systems;
- Control Video Wall configuration from the Dashboard and/or IoT App Business logic;
- support authentication and authorization with the most general approaches such as LDAP, SSO;
- collect and get data from batch resources and in real time, using a large range of protocols and formats;
- Associate them a dashboard specific menus;
- Connect to each dashboard a dedicated chat room for discussing about problems and events;

The Snap4City Dashboard system is used in several cities and projects as can be accessed from:

- Scenarios: https://www.snap4city.org/4
- **List of Public Dashboards**:
  https://www.snap4city.org/dashboardSmartCity/management/dashboards.php
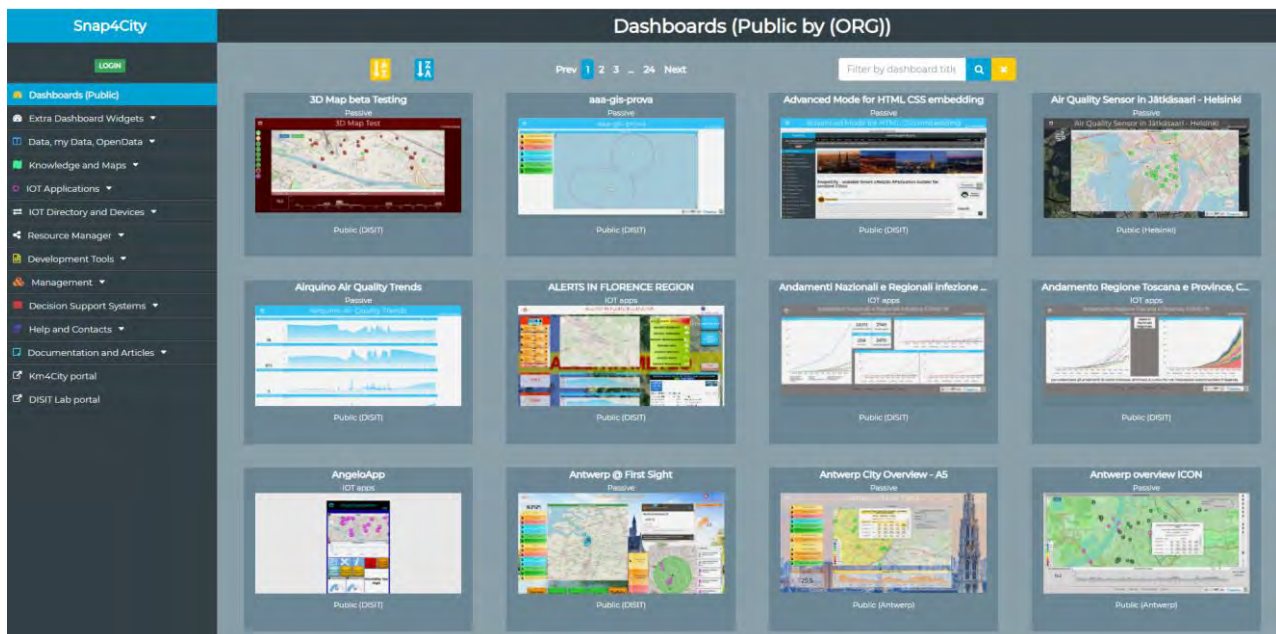
3.10.2 – Examples of Snap4City Dashboards

In the following a number of examples, also with Arabic and Hijri dates.
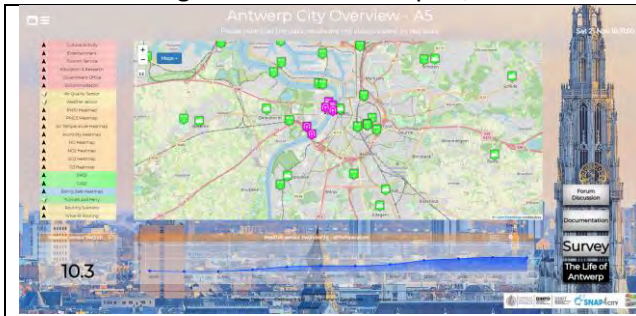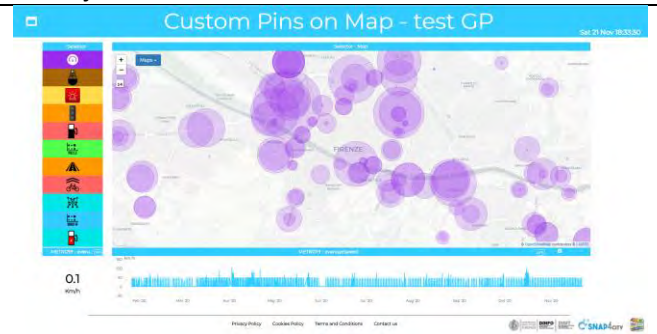
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTQwNw==	https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=Mjk5MA==
Antwerp dashboard	Example of dynamic PIN monitoring real services in Florence
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjQwNw==	https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTUzMg==
	Environmental data, NOX predictions and several heat maps
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjM1Ng==	https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjM1NQ==
Events and Ordinances in Florence	Helsinki dashboard
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=Mjg5Mw==	https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MzAyMw==
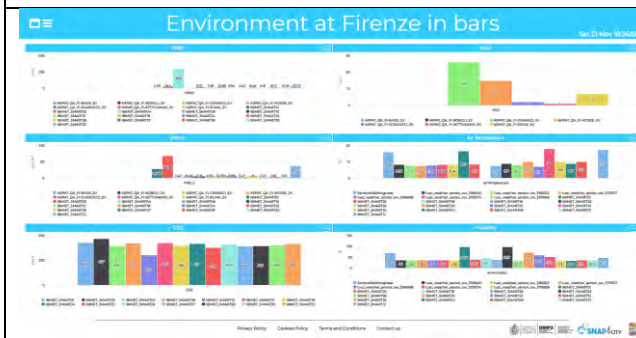
Università degli Studi FIRENZE — DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE — DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

SNAP4CITY — KM4CITY

https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjU3NQ==

COVID-19 data

https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjU2OQ==

COVID-19 data

https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjE5MA==

What-if analysis tool in Florence
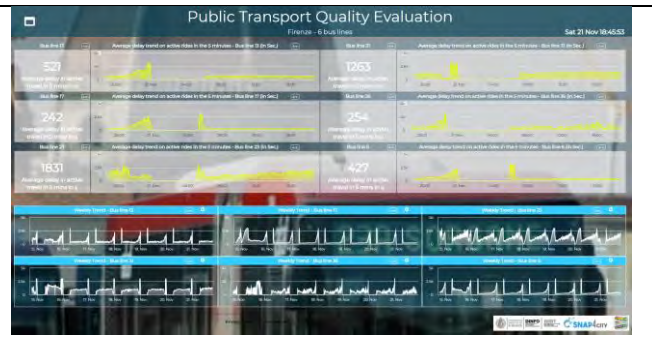
Monitoring of Public Transportation, average waiting time at bus stops.

https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjcyNg==
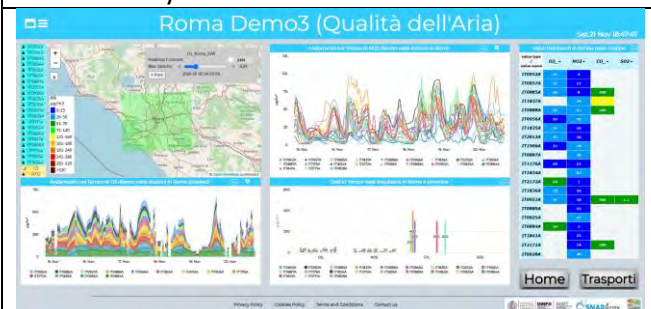
Rome, monitoring environmental data

https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjU2OA==
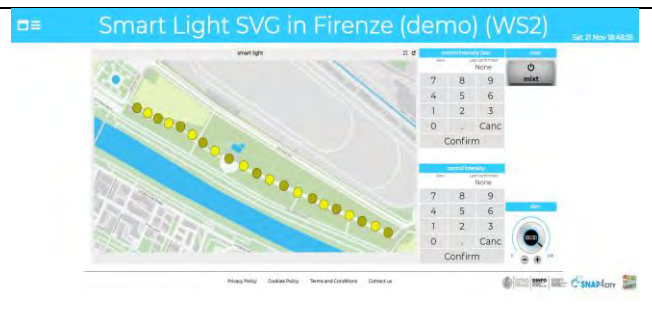
Synoptic for smart lighting

https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjU3MA==

Synoptic for smart waste

https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=Mjk4Ng==
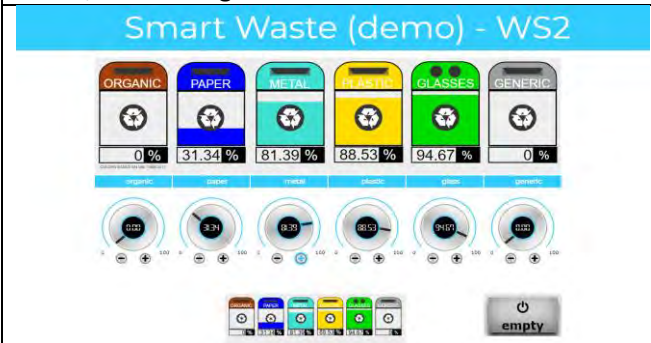
Synoptic examples
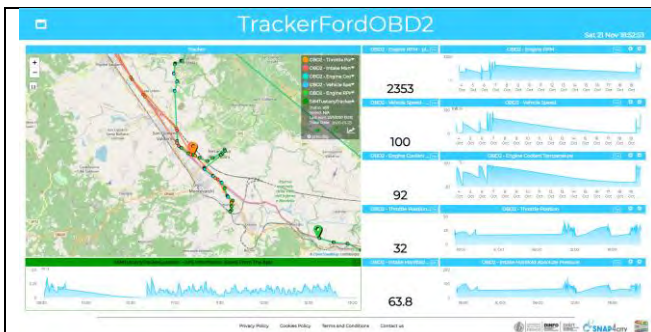
Tracking moving sensors and vehicles	https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjE4Mg==
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjY4MQ==	https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjY1MQ== Traffic flow monitoring, in/out flows in Florence
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc5NQ== Traffic flow reconstruction several cities	ISPRA JRC energy synoptics
Smart@Home 5G dashboard	City today of Florence
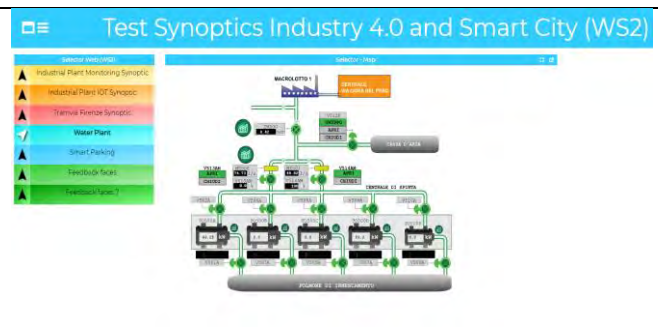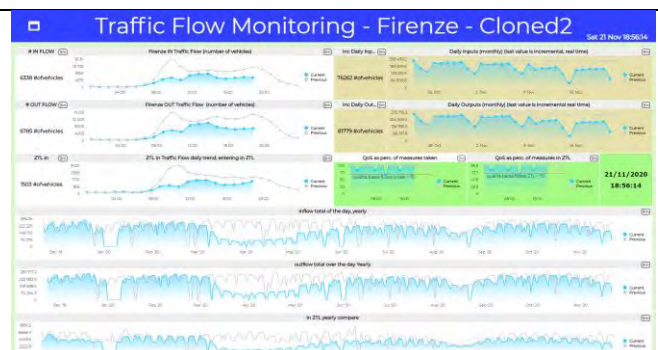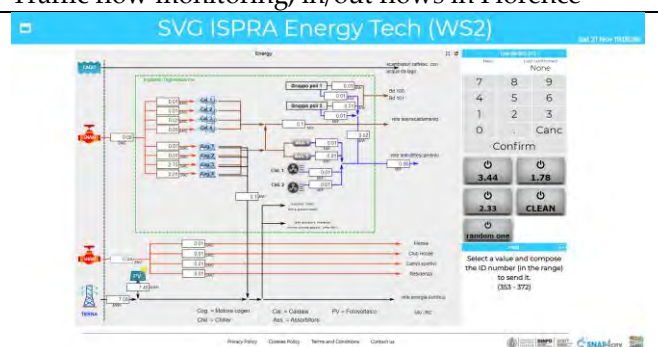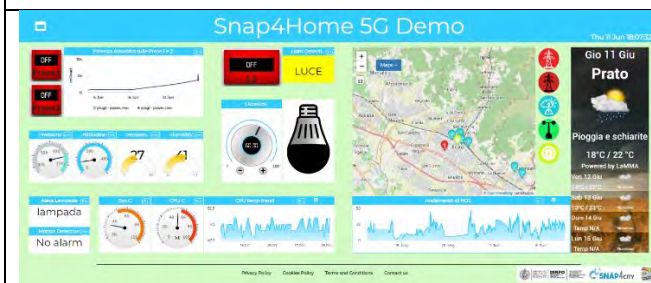
Florence Metro City Control Room main dashboard	Smart parking in Lonato del Garda
National level OD map	https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc3NA==
Analysis of Wi-Fi Data	Twitter data, volume and sentiment
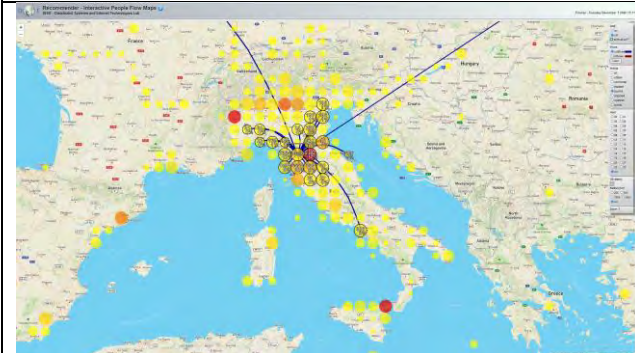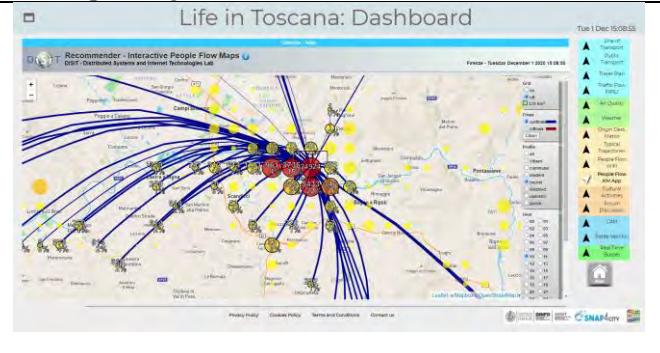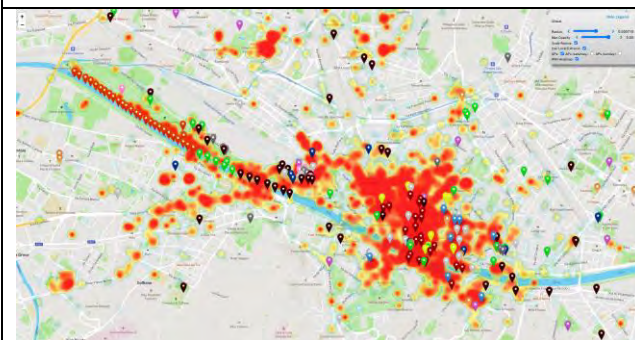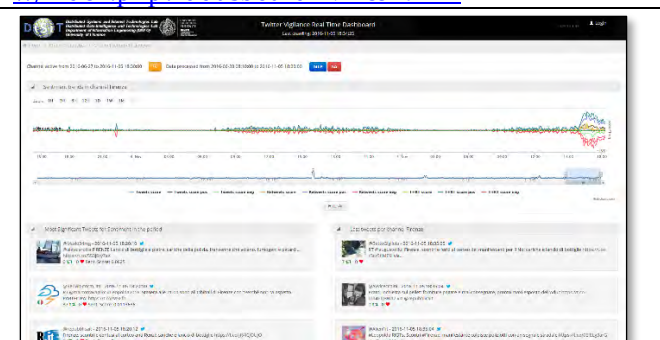
Some of the above dashboards are private and thus cannot be freely accessed. For these reasons some of the links are also missing, instead of providing links that could bring you to land on the login splash.

3.10.3 – List of Main Dashboard Widgets for Dashboard Builder

Snap4City Dashboard Builder provides a very large library of widgets which can be used to create Dashboards. A number of them are visible in the above reported examples. Please note that, a large number of Widgets in the library and the Wizard are shortening the time for creating contextual dashboards. This is a high value since the Dashboards are often requested when needed, typically they are *requested today for yesterday*, this what happen in a large city you have.

Snap4City Dashboard Widgets can be classified into:

- **Time Series and real time data (see also custom widgets for the real time rendering of data via animation, dynamic PINs on map, etc.)**
  - Single content, any value from numbers, strings, to HTML
  - Time trend,
  - Speedometer,
  - Gauge,
  - Single Bars,
  - Kiviat and spidernet,
  - Donuts and Pies,
  - Tables, tables with colormaps,
  - Multi BarSeries: vertical/horizontal, staked or not,
  - Multi Time series: staked/normal, shaded,
  - Time Compare for time series,
    - Versione nuova
  - Events scrolling lists.
  - Typical time trend
  - Calendar
  - 3D Olap time series
- **Maps:**
  - **Multi Data map**:
    - Maps, pan and zoom,
    - 3D buildings on map
    - 3D global digital twin
    - POI, points of Interest
    - Orthomaps, back grounds
    - Heatmaps: calibrated and gaussian
    - GIS information
    - Polylines, shapes
    - Routing, routes
    - Sensors and real time data
    - Definition of scenarios
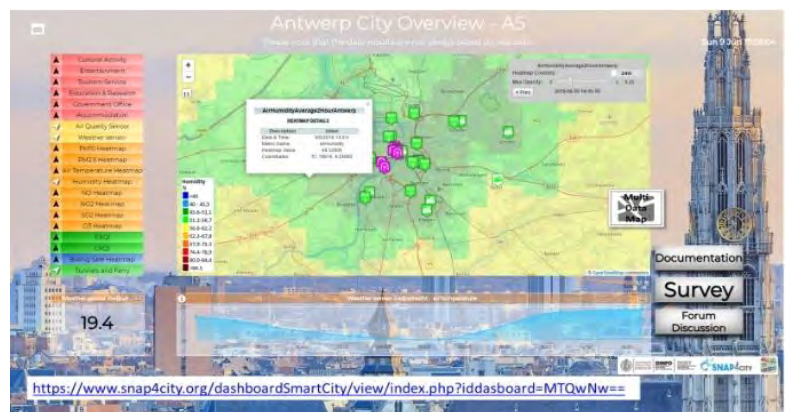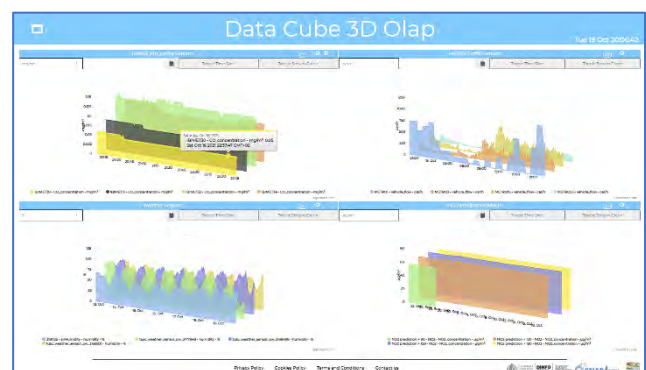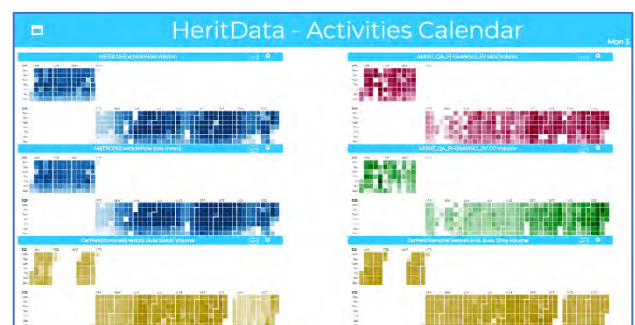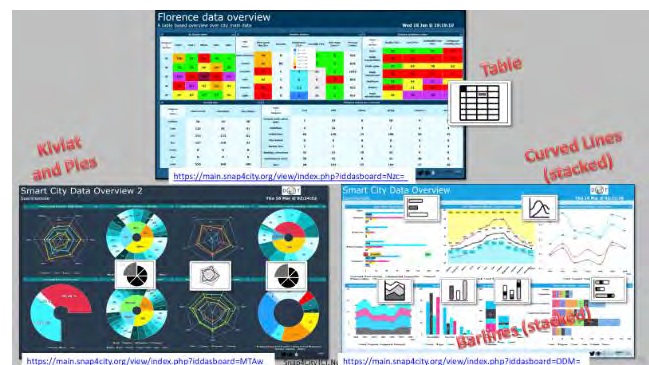
- o **Moving device tracker on maps**
- **Selectors**: (see also custom widgets below for other selector)
  - o Selector from services: icons, text, technical, etc.
  - o Selector Web: from external Services
  - o Selector from IoT App, just multiple choice selector
- **Actuators (see also custom widgets below for other actuators)** from Dashboard to IOT
  - o Buttons, image buttons
  - o Knob, dimers
  - o Switch, impulse button
- **Special**
  - o TV cameras
  - o Public transport status
  - o Decision barline from Smart Decision Support, so called Italian Flag of some SmartDS processes.
  - o Twitter: hashtags, citations
  - o Traffic flows
  - o Weather Forecast of several kinds
  - o Clock (time: HH:MM:SS) real time
  - o Civil protection status and alerts
  - o Digital Twin Local, BIM with sensors pin on the 3D model
  - o First Aid Status: single and multiple triage monitoring
  - o SmartDS, Italian Flags
- **Micro Applications** (huge collection:
  - o https://www.snap4city.org/dashboardSmartCity/management/microApplications.php
  - o bike sharing,
  - o events of entertainment,
  - o favourites POI, marked by the users
  - o my personal POI, market by the users
  - o near to me, search capabilities
  - o personal trajectories,
  - o pollution, pollination, air quality,
  - o Public transport timelines
  - o public transport, GTFS data
  - o routing, travel planner,
  - o search of services by selection,
  - o services browsing: accommodation, parking, wine and food, etc. (about 20 macro categories)
  - o smart parking,
  - o suggestions and goal experiences,
  - o tickets for public transport,
  - o user engagement,
  - o weather forecast, current weather
  - o etc.
- **External Services** (depending on your area)
  - o https://www.snap4city.org/dashboardSmartCity/management/externalServices.php
  - o Any web page with Iframe consensus.
- **Custom Widgets Sensors and Actuators, interactive and with animations and dynamic values** (changing
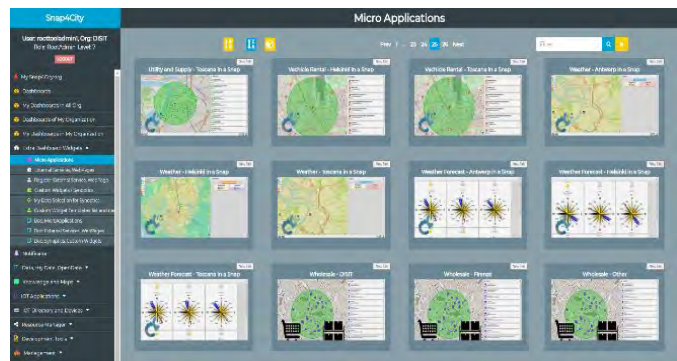
color, changing shape, animating, etc..) (collection): they can be public or private

- o https://www.snap4city.org/dashboardSmartCity/management/synopticTemplates.php
- o Air conditioning, de-humidifier
- o Alarm status
- o Bikes, cars, etc., of different kinds
- o Charger status, level
- o Digital signage status
- o Doors status
- o Energy synoptics, industry plant, of custom layouts
- o Faces feedback of different kinds
- o Fans status
- o Lamps, road lamps, ..
- o Metro status, different layout
- o Plug status
- o Pollution status: PM10, PM2.5, ..
- o Red light
- o Road signals, Speed limits, others..
- o Single slide, double point slider
- o Smart light, of custom layouts
- o Smart parking layout, of custom layouts
- o Smart recharging station
- o Smart waste
- o Switches several formats
- o Tank status/level
- o Thermometer
- o Tunnel status
- o Etc.
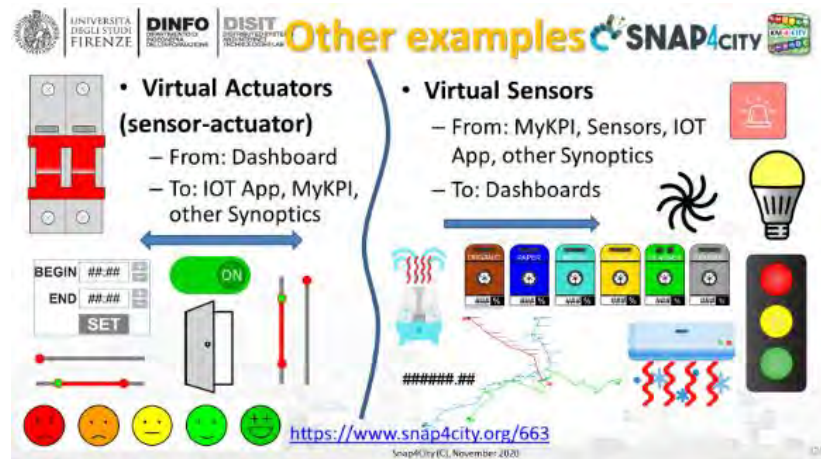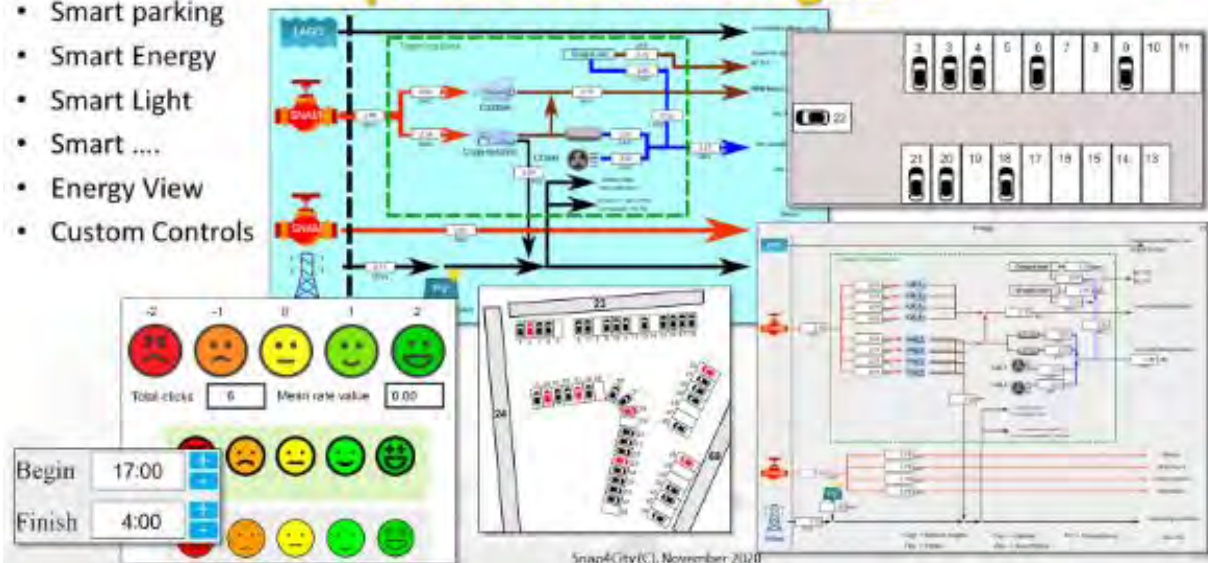
Snap4City.org is a perfect example of the training environment which can be set up using Snap4City Platform. Snap4City.org is a **multitenant full installation of Snap4City platform** in which several Organizations are hosted.

The list of Organizations accessible from: https://www.snap4city.org/download/video/cov/ The different Organizations have different levels of privacy and are exploiting the Snap4City.org service and platform in different manners for different purposes. Some of them for tourism, other for mobility and transport, environmental aspects, others for full smart city control room with several domains and smart solutions applied, and others again for verticals as smart parking, smart lighting, and managing mobile applications, etc. So that the Snap4City.org is a multi-organization living lab.

Some of those organizations have also their own installation of Snap4City Platform on their premise, and use their tenant on Snap4City.org as the sand box in which new features are tested, training courses are given, documentation is published and new examples are accessible.

A Snap4City platform provides to developers, entrepreneurs, public and private sectors, non-profit organizations, researchers, etc. an SDK which includes all the instruments needed to **ONLINE development** (if not otherwise specified) for creating, saving, sharing and editing, their artefacts:

- **Living Lab environment** for accessing to training information, video, examples; discussing about solution, sharing experiences.
- **Resource Manager:** to search for solutions, sharing resources, saving back up of their solutions.
- **Dashboards** and **Widgets on Dashboards, widget collections.**
- **Synoptics**, **Custom Widgets (custom widgets may need to install open source SVG editor on local computer), custom PINs for animated and dynamic PIN on Maps.**
- **IoT Apps**: for connectors, integrations, data ingestion, data transformation, business logic behind Dashboards, business intelligence, services. Also managing versioning on GitHub of their flows.
- **IoT Brokers** and connectors of any kind full features of FIWARE based solution, but not limited to FIWARE solutions and protocols.
- **IoT Directory for IoT Network management and abstracting from Brokers, Devices.**
- **IoT Device Models, and SDK to develop code to be placed on physical IoT Devices based on Android, Raspberry Pi, Arduino, ESP32, window, Linux.**
- **Data Analytics** in various languages: RStudio, Python, also transformed in API services and MicroServices for IoT App. Please note that RStudio can call segments in Python, Java, etc.
- **External services as web pages to be integrated into Dashboards;**
- **External services accessible via API, to be integrated as API, and MicroServices in IoT App;**
- **Web and Mobile Apps: with standard SDK for Apache Cordova to be installed on local computer**
- **Web Scraping**, Crawling also transformed in API services and MicroServices for IoT App.
- **Workflows on IoT App and OpenMaint/BPM, also integrated each other.**
- **HeatMap server:** for automated production of heatmaps and their distribution via other GIS: ArcGIS for example.
- **Report generator:** at levels of users, devices, dashboard, solution.
- **BIM models and data, created and edited with third part tools in standard IFC and then integrated into on line OpenMAINT.**

On line training is accessible from Https://www.snap4city.org/577

From the training slides you can see a large number of real scenarios with real dashboards, IoT App, business intelligence solution, full applications and Data Analytics in place.

Since now you can access at the Training Course 2020 Slides in PDF and in the interactive formats. The Videos accessible (at the time of writing, November 2020) are those of the last recordings of April 2020. A lot of new material and improvements have been deployed and distributed, so that, you can see the differences by comparing the present slides with respect to the videos.


When the course is given, a number of practical exercises are requested to be personally performed exploiting the online platform, and the teachers assess the proficiency of the participants on the basis of questionnaires and by assessing the artefacts produced on the platform. Example of certificates of participation and/or proficiency are accessible on: https://www.snap4city.org/622 The training course is at the University level; the Teachers are University Professors or Assistants.

The online course is based on the full life cycle of the Smart City development environment described in the next figure, and described in detail into the **Training Course**, several parts, and at general overview in **Part 6** https://www.snap4city.org/download/video/course2020/sys/
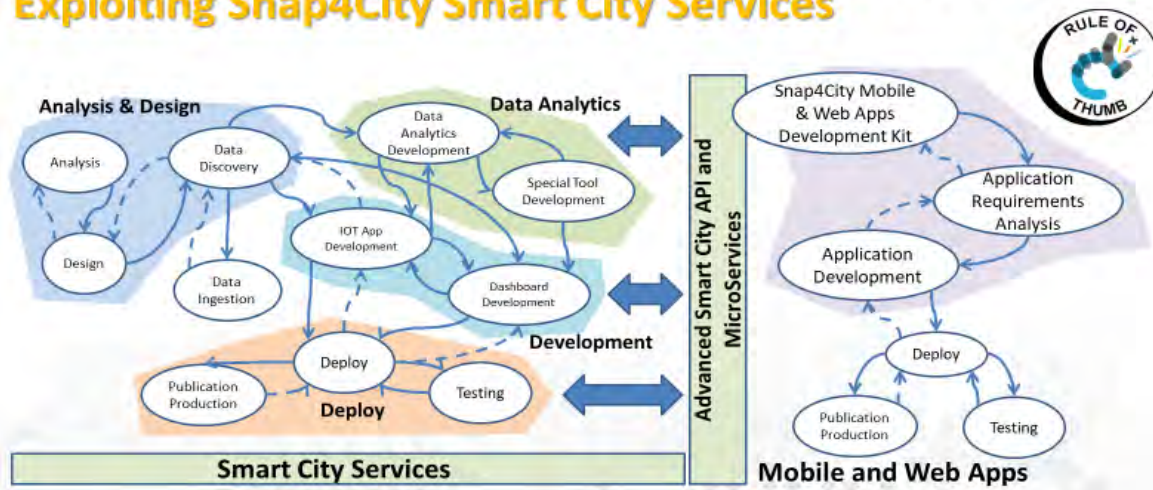
The **course of 2020 structure is detailed at page** Https://www.snap4city.org/577 as follow:

- **first part** provides an overview of the Snap4City platform.
- **Second part** provides a detailed view on Dashboards capabilities and mechanisms for their production. The Exercises are on dashboards production using the Wizard and manually.
- **third part** goes inside the IoT Network, device registration, brokers, and the development of IoT App for Connectors, Integration, data transformation and business logic behind Dashboards. The exercises are mainly on IoT App, JavaScript, Node-red.

- **fourth part** addresses the problems of Data Analytics, providing a large number of examples: smart parking, bike sharing, heatmaps, traffic flow reconstruction, user behaviour analysis, user engagement, etc. The exercises are mainly in RStudio and/or Python.
- **fifth part** describes in details the procedures for data ingestion, how to ingest data, using IoT App and the other tools, how to verify the correct ingestion, etc. The exercises are on registering data, monitoring the flow, controlling the process of ingestion, in most of the cases and data models.
- **sixth part** includes an overview/description of: (i) development environment and process, (ii) living lab processes and guidelines, (iii) configurations for deployment of the Snap4City on premise, (iv) usage of Snap4City technology for Industry 4.0, (v) a view of the Administrative tools. Most of these aspects are producing specific training courses. For example, for administrators, for Living Lab managers, for trainers, for installers, etc.
- **seventh part** describes the smart city APIs and how they can be exploited for developing Web and Mobile Apps.

3.12 – Snap4City Security

Snap4City platform has successfully passed PENtest performed by SETEK and Vulnerability Test performed by Thales Italy in the 2019. **Corrective Actions** performed on **Snap4City** main platform are reported on https://www.snap4city.org/669 . The security of the **Snap4City** solution has been validated in international top journal and publication IEEE Access and it is freely accessible from **[Security2020].**

The OWASP IOT mainly refers to the IoT Network aspects, which are out of the platform control. On the other hand, the aspects referring to the server side of the top 10 problems (e.g., 1) passwords, 2) network services, 3) interface, etc. https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project) are practically also verified by PENtest performed by SETEK and Vulnerability Test performed by Thales.

3.13 – Snap4City Access Control and GDPR Compliance

Snap4City platform has been designed and it is GDPR compliant as an IOT/IOE platform. In Snap4City, all devices start as **personal private IoT Devices,** such as: Mobile App data, personal devices data, SmartBeds data, Cars data, hearth monitoring, glucometers, etc**.** They may become public or accessible to one or many users, groups, orgs, on the basis of grant authorizations, defined by the owner**. Snap4City** platform is GDPR compliant and allows to collect and manage data according to the rights of the data owners, and to control the data exploitation when they are distributed. GDPR aspects are presented on Snap4City on https://www.snap4city.org/670

See for security and GDPR aspects: [**Security2020]**

Snap4City platform can be installed in several different configurations exploiting the same modules. The different configurations may be suitable from small cities to large cities with a high number of users and huge volume of data. The different modules can be provided as VM (Virtual Machines) Appliances or Dockers. The different Snap4City configurations and each component mentioned in the following are described in detail with the instructions for their installation at page: https://www.snap4city.org/471 **Please note that each VM is an appliance with a number of components which can be updated directly from GITHUB/DISIT.** **https://github.com/disit**
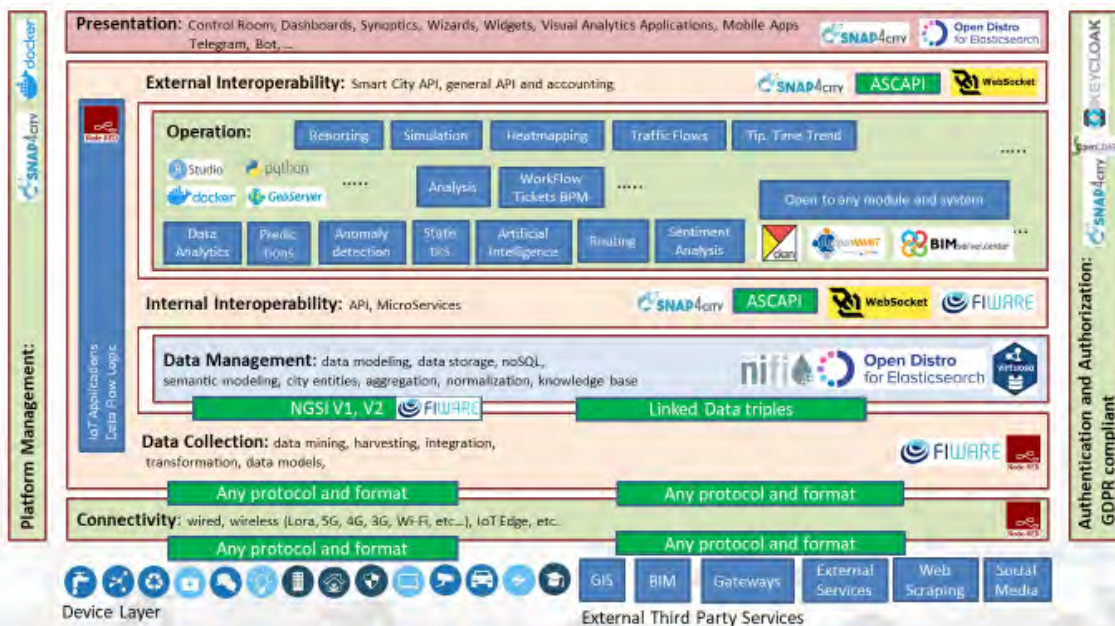
The Edition of 2021 is based on VM and Containers:

- Docker configuration tool
- Docker Configuration tool documentation: https://www.snap4city.org/738

For example, a small deploy could include:
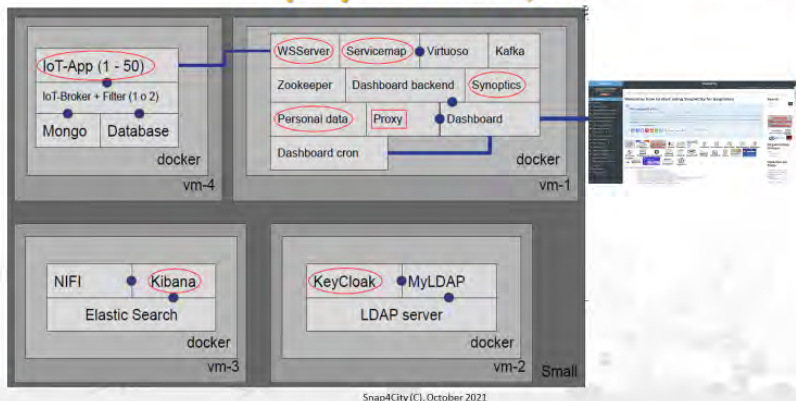
DataCity-Large which is reported in the following figure, and it is mentioned quite similarly into Tutorial **Part 6** https://www.snap4city.org/download/video/course2020/sys/
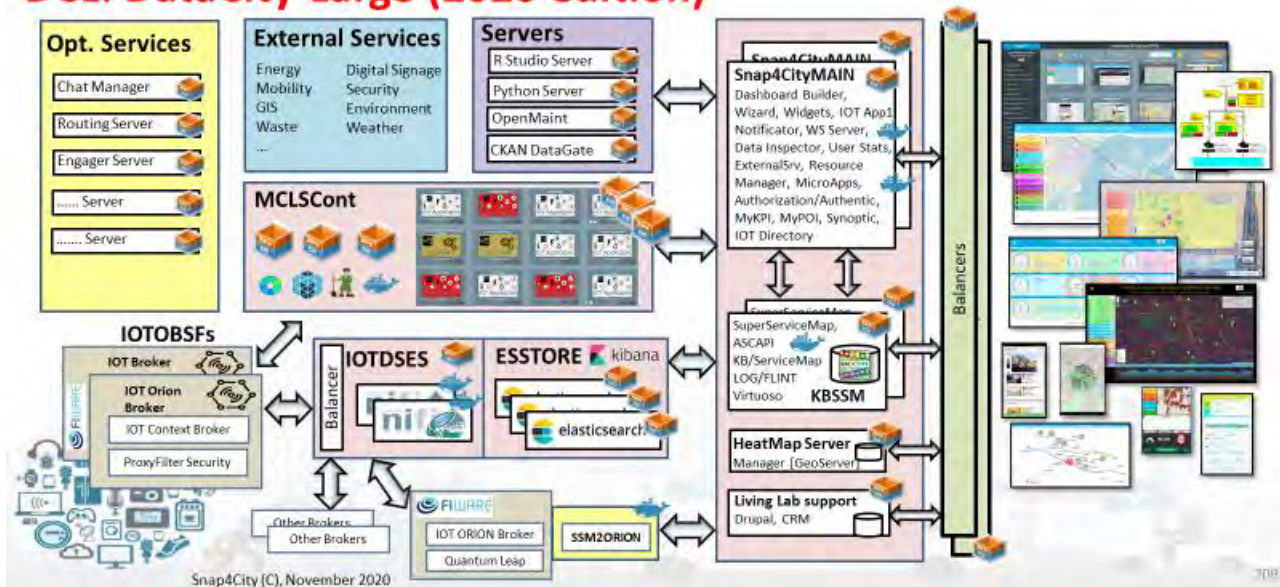
No proprietary components are needed to be installed into the Snap4City platform.

DCL: DataCity-Large configuration includes installations of:

- **Snap4CityMAIN VM/Docker compose with**: Dashboard Builder, Wizard, library of Widget, WebSocket Servers, Data Inspector, User Statistics, External Service manager, Resource Manager, Micro Applications manager, LDAP/KeyCloak for Authorization/Authentication, Kafka (which has to be scaled up if the traffic on synoptic is high), MyKPI, MyPOI, Synoptic and Custom Widget/PINs, and IoT Directory;
- **KBSSM VM**: **ServiceMap**, Smart City API, Knowledge Base Km4City (LOG, Flint, Virtuoso): which can be duplicated in federated configurations to increase robustness/volume of traffic or in FT;
- **Heatmap Server VM** with an heatmap manager to generate the GeoTIFF which are loaded into a GIS for their distribution via WFM protocol such as ArcGIS or GeoServer; which can be started in FT;
- **Living Lab Support VM** with Drupal 7 with Snap4City extensions; which can be started in FT;
- **MCLSCount VM** which is cluster of VM to manage Docker Containers with Marathon, Mesos and DISCES-EM. They are containers for IoT Apps, DataAnalytics, Web Scraping; This cluster has 3 VM for the masters and a number of VM for the Dockers/containers. It may have sense to have a true cluster if the Dockers are more than 200 Containers, otherwise it may be enough to have a static allocation of them in a few VMs in FT. The Snap4City solution has been tested on thousands of Dockers/Containers and the larger cluster of VM consists presently of 16 VMs for Dockers elastically managed. **[Scalable2018], [DataFlow2019].**
  - Typically, we have an average of 70 Containers for each VM depending on the image size of the container, and VM size. The Dockers for Data Analytics cost about 4 times bigger in terms of CPU and Memory with respect to those for IoT Apps, and have a more sporadic usage of the resources. Their timing is more relaxed.
- **IOTDSES** a cluster of federated NIFI on Dockers and their balancer, they are allocated on VM as well;
- **ESSTORE VM** a cluster of Elastic Search and Kibana VMs. The number of VM depends on the size of the global storage which can increase elastically with the need along the operation. In alternative the usage of a shared common SAN and docker could be viable if it is fast enough:
- **IOTOBSF VM**: IoT Orion Brokers (V1/V2) also on Dockers or VM, internal or external, etc.;

- **Rstudio Server VM:** for development of R Studio processes; Their size depends on the number of developers you have at the same time working on them;
- **Python Server VM**: for development of Python processes; Their size depends on the number of developers you have at the same time working on them;
- **CKAN DataGate VM**: for the connection to CKAN open data network;
- **OpenMAINT VM**: including OpenMAINT and BIM, for the development of Workflow, BPM and BIM;

See more on: TC10.13- **Back office Cluster Container Scalability and Monitoring Scalable** Infrastructure with Elastic Management of applications and processes. https://www.snap4city.org/232

Example of MCLSCount IoT App and Docker cluster Monitoring. The DISCES-EM, which is the DISCES for Elastic Management, if the engine that continuously monitoring the Mesos, Marathon and VMware resources to take decision about the turn on/off of VM from the pool of **MCLSCount** VMs performing elastic management. It also reacts to eventual failure of the VM restarting new VM and relocating containers.
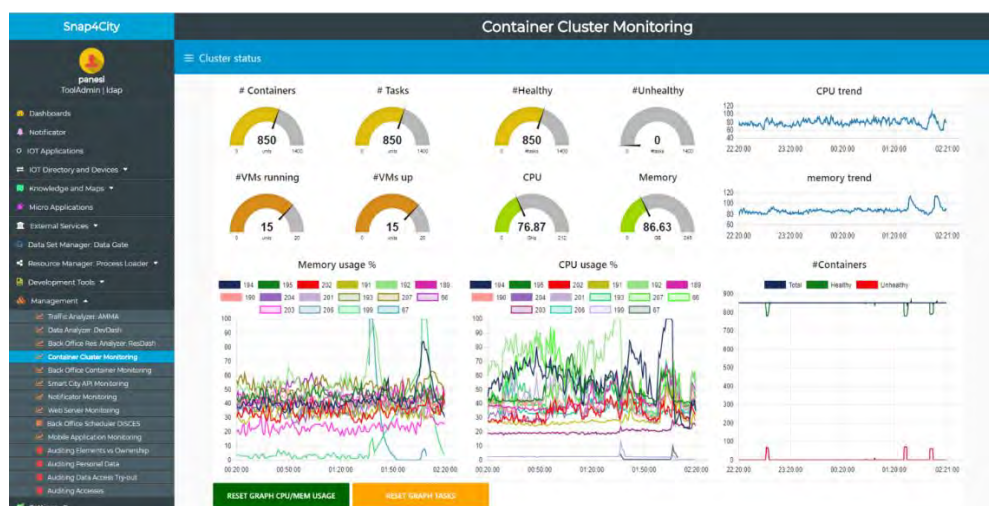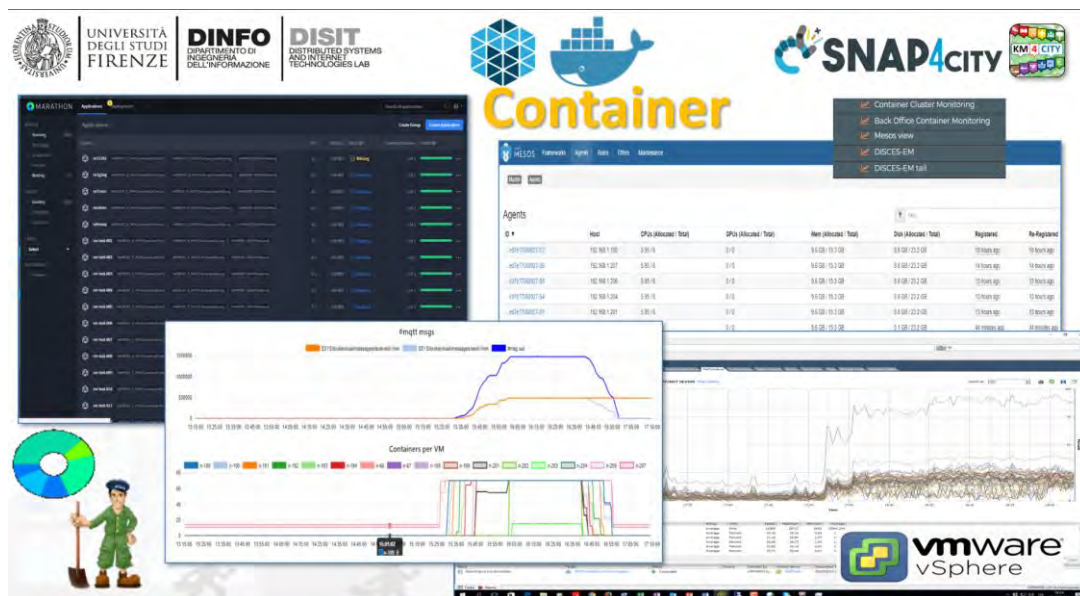
- #**Containers**: a gauge reporting the number of Containers/IoT Applications
- #**Tasks**: a gauge reporting the number of Containers/IoT Applications which are managed by Marathon. When a Container is moved, temporary two instances are allocated so that the #Tasks can be higher than #Containers

- **#Healthy**: number of Containers that are in the Healthy status
- **#Unhealthy**: number of Containers that are in the Unhealthy status
- **CPU Trend**: the value of total CPU computer in GHz of the cluster
- **#VMs up**: number of VMs which have been requested to be running from the scaling manager
- **#VMs running**: number of VMs which are up and running for the container manager
- **CPU**: mean amount of CPU consumed in the last minutes in terms of GHz
- **Memory**: mean amount of Memory consumed in the last minutes in terms of GByte
- **Memory Trend**: the value of total Memory used by the VM activated in the cluster, in GByte
- **Memory usage % graph**: a time trend of the percentage of Memory Usage for each VM that has been activated. The list is dynamic, the ID of the VM correspond to the IP on the 1.X cluster.
- **CPU usage % graph**: a time trend of the percentage of CPU Usage for each VM that has been activated. The list is dynamic, the ID of the VM correspond to the IP on the 1.X cluster.
- **#Containers graph**: is the time trend of the #Container: Total, Healthy and Unhealthy
- The two buttons: RESET CPU/MEM USAGE and RESET TASKS are used only for resetting the graphics.

Container Cluster intelligence
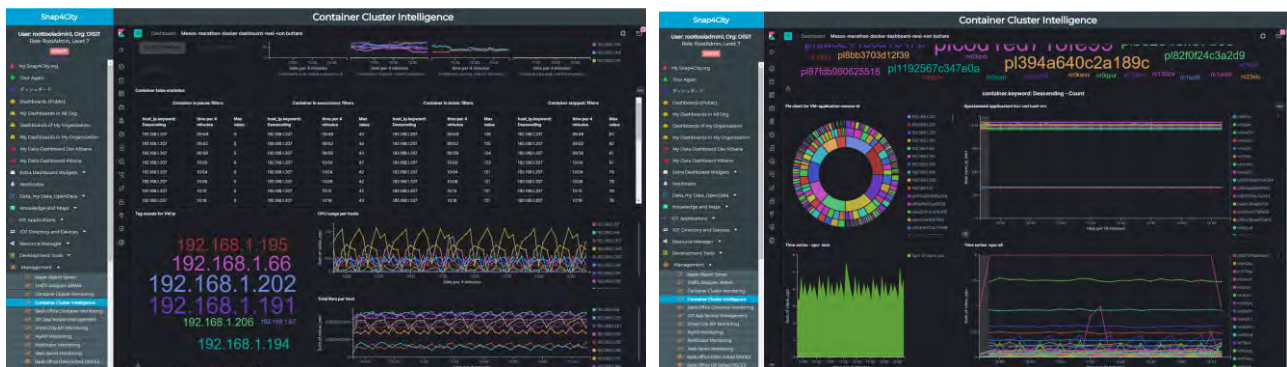
3.14.1 – Snap4City Optional Tools for a typical configuration

In this section, other **VMs/appliances/dockers and data analytics (DA)** which are **not included into the typical configuration**. They are:

- **Mobile App staff:**
  - **Mobile App** "In a Snap" series, APP; based on [MobileDevKit2017]
  - **User profile manager** and suggestions VM; [UserBehavior2020]
  - **User Engager** system for mobile Apps, VM; **[UserEngagement2017], [UserBehavior2020]**
- **Data Analytics:**
  - **SUMO Simulator VM;**
  - **Traffic Flow predictions, [TrafficPredicitons2020]**
  - **Traffic Flow reconstruction and simulation** VM; [TrafficFlowReconstruction2020]
  - **Parking** predictions, DA; **[SmartParking2018]**
  - **bike** sharing rack predictions DA;
  - **Computing Trajectories, OD, Typical Trends** (DA);
  - **NOX predictions** VM/DA; [NOX prediction 2020]
  - **Wi-Fi user behaviour analysis, DA**, **[UserBehaviour2017]**
  - **Anomaly Detection, DA;**

- o **Routing Server VM** for routing, constrained routing, multimodal routing;
- o **COVID-19 data monitoring, DA;** [COVID-19]
- **Features and tools:**
  - o **DORAM tool for the analysis of demand of mobility vs offer of transportation;** Tools;
  - o **Chat manager** for dashboards, specifically for Control Room and Early warning management, VM;
  - o **GTFS editor VM/Docker; Covers also formats as also other formats: Kochi, Hyderaband.**
  - o **DISCES Scheduler and cluster VM;**
  - o **Resilience DS** VM/Docker [**Resilience2017], [Resilience2019].**
  - o **Smart D**S VM/Docker; [**SmartDS]**
  - o **Twitter Vigilance VM** in different forms. For early warning and moods assessment: English, Italian, France, and a few of other languages; [TwitterVigilance2017]
  - o **SSM2ORION** Docker which is a module for connecting an IoT Orion Brokers with its Data Shadow implemented by using **Quantum Leap** with a Snap4City **Federation of Knowledge Base**. This solution allows to connect at the Smart City also other already in place FIWARE solutions which may have local storage. The queries performed on Snap4City API provide seamlessly the result also providing the geo data which are stored into that Orion Broker without the need to register the IoT Devices of that IoT Orion Broker into the Knowledge Base. Some limitations are present on the security aspects.
  - o **Large file sender in addition to FTP:** https://filesender.org/, **VM**
- **Cloud SW:**
  - o **All Orchestrator and managers for cloud are not included, such as VMware vSphere, VMware converter, etc.**
  - o **VEEAM software for backup of VM is not included**
  - o **HUB and Firewall software if any is not included**

3.15 – Snap4City: Maintenance Management, BPM and BIM

Snap4City Maintenance Solution includes:

- **OpenMaint** open source solution for property & facility management which is a BPM;
  - Inventory of industry assets (movable, logistics, equipment, etc.)
  - Tickets management for corrective maintenance
  - User management with different levels of access
  - BIM Server integrated with OpenMaint;
- **Snap4City OpenMaint Extension**
  - **Extended API** developed by Snap4City
    - Create new tickets
    - Manage steps, workflow
    - Collecting feedbacks and results from teams
    - Manage all phases of the workflow on the fields via IOT Apps and logics
    - The integration if via API and MicroServices into IOT App.
  - **MicroServices** integrated with Snap4City via IOT Applications;
- **Business Intelligence** which is the **Snap4City tool based on** Elastic Search: which work on top of the database of tickets collected on OpenMaint;
- **BIMServer** integration with Snap4City Dashboards.
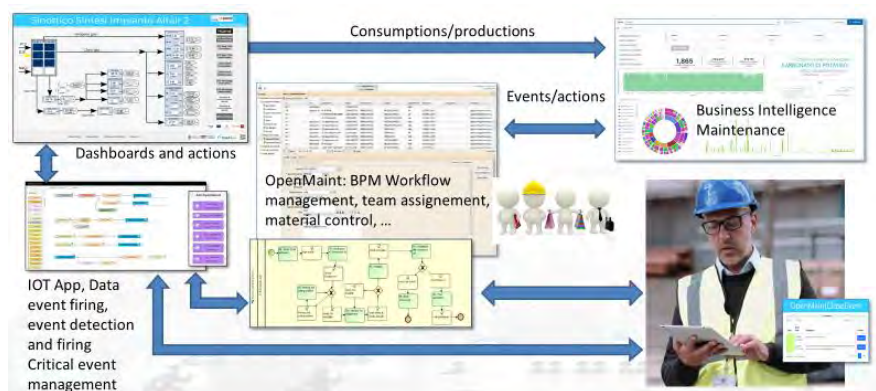
- o **From map to BIM, from bIM to sensors**
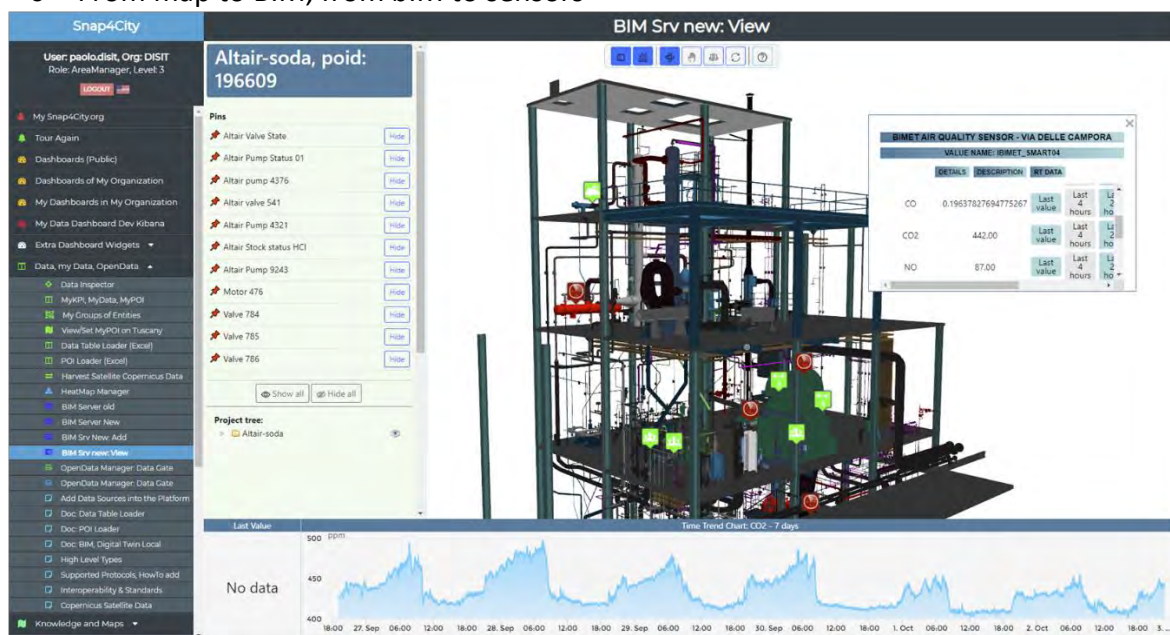
OpenMaint is an open source solution and it

- is a reliable and scalable web solution that can be completely used through a standard browser. openMAINT implements **Business Process Management** (BPM) functions to draw workflows and assign specific operations to users/offices, and assist them to check the development and the correct working
- includes BIM capabilities with its integration with **BIM Server** as described in the following.
- is available also for mobile devices (for Android and iOS smartphones and tables).
- allows the management of mobile assets (buildings, infrastructure, etc.), plants and technical devices (electric panels, heating and cooling, light fixture, fire extinguishers, etc.), furniture, etc., and the related logistical, economical and maintenance activities (scheduled and corrective ones).
- helps the managing authority to know, organize and update those data related to the asset inventory, in order to support their decision and operative procedures.
- can be customized in order to shape over the environment where it operates, usually made up of procedures, documents, roles and responsibilities, technical restrictions, textual data and georeferences, external systems to interact, etc.
- provide features based on a unique and innovative engine **CMDBuild**, which allows to remodel data models, workflows and reports without modifying the code of the application software.

Functional areas of openMAINT include:

- **Space & Asset Inventory**: Inventory of real estate assets, plants and related components, for a complete knowledge of their technical and functional characteristics.
- **Facility Management**: Management processes of Preventive Maintenance and Corrective Maintenance of the assets registered in the system.
- **Logistic Management**: Database of warehouses and warehouse items, handling of materials used in maintenance activities.
- **Economic Management**: Management of budget, suppliers and contracts for the purchase of goods and services, registration of purchase orders and related costs.
- **Energy & Environment**: Recording and analysis of energy consumption data of buildings.
- **BIM support**: 2D vectorial plans and on 3D BIM models, synchronized with the design tools using IFC standard files

Main features of openMAINT, BMP, BIM are:

- web based user interface
- text and graphic browsing through the objects
- complete history of the changes on any data card (versioning)
- rapid and analytical searching functions for every card attribute
- definition of filters and views for a customised access to data
- classifications and lookups which can be autonomously customised
- document file used to match every kind of file with every kind of card
- workflow engine integrated with a visual editor for the graphic design of processes

- report engine integrated with the visual editor for the report design
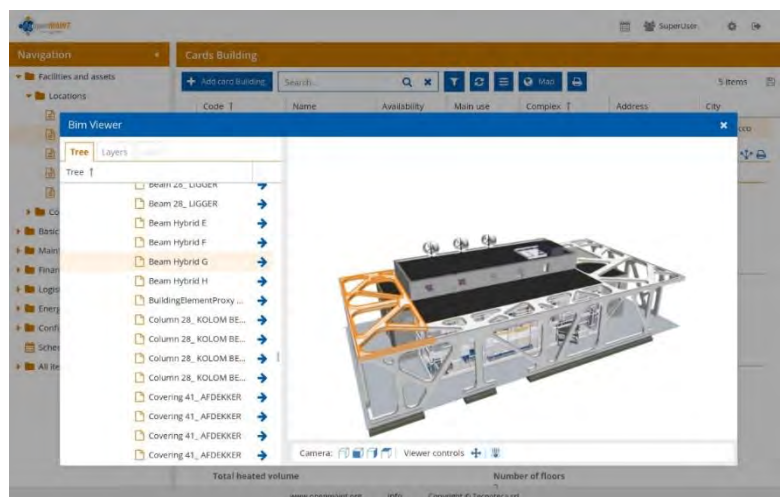
- possibility to define customised web custom pages
- complete outline of users, groups, permissions, and multitenant configuration
- interoperability with other applications through webservice
- CSV data import / export functionalities
- possibility to configure and manage (task manager) automatic operations
- possibility to configure e-mail for notifications and as communication tool
- use of barcode and QR code to automatically recognize the items
- possibility to configure connectors through external systems
- Self-Service portal for non-technical users
- mobile APP for the execution of the main features through tablets and smartphones
- possibility to autonomously model the data scheme:
- creation of new typologies of objects (classes)
- creation of new attributes on pre-existing objects (attributes)
- creation of new typologies of relations among objects (domains)
- Preventive and Corrective Maintenance
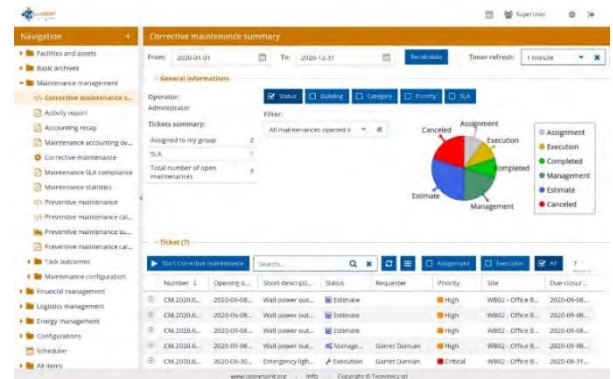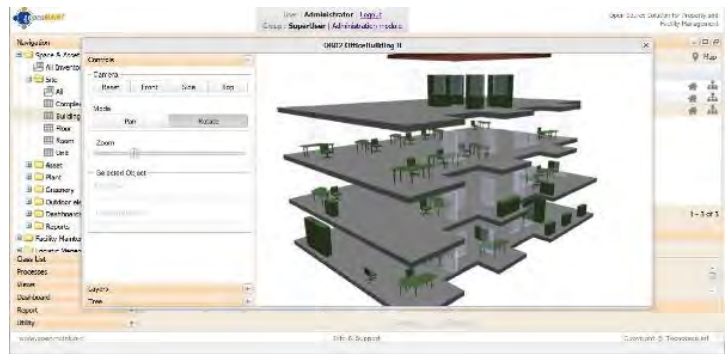
Buildings 3D shapes, and Floors are managed and distributed into the **OpenMAINT** tool, standard BIM tools are used for editing and interchange in IFC formats by standard tools as AutoDesk Revit, etc. Buildings and Floors are shown in **Dashboards** for their integration with maps and IoT Devices time trends

3.15.2 – Snap4City Business Maintenance Intelligence tool

The collected data into the OpenMaint related to tickets and interventions performed on the city or industry plant are collected into an Elastic Search which allows the decision makers to perform:

- Faceted search on the basis of the maintenance actions classifications, kind of elements substituted or repaired, kind of interventions, codes, etc.
- Impact analysis of faults with respect to the plant functionalities and production
- Drill down on time and 100% of the working plants

Snap4City has a number of rules:
- **Users Roles:**
  - **RootAdmin:** The gods of the specific installation, access to all tools for all Organizations
  - **ToolAdmin:** The administrators of an Organization with some capabilities on single tools
  - **AreaManager:** Typical developer capabilities, access to development tools, access to a wider number of resources, IOT with both basic and advanced, IOT Models, etc.
  - **Manager:** Final users, limited access to development, IOT App development with Basic library.
- **Users of any Role** have full control on their own resources: data, devices, dashboards, IOT App, etc., which may control according to GDPR rules, providing access, revoking, etc.
- **Other Roles may be defined**
- **Specific Accesses and features can be granted by the RootAdmin**
- **All users start as Manager roles:** All users have also a Level (numeric). A score about what they have exploited in the platform. Higher scores correspond to wider exploitation of capabilities.
- **Organizations** may have
  - name, ID, GPS center, a number of Groups on Snap4City.org (living lab support Drupal)
  - users of different kinds and may impose early bounds on the resourced used by users (IOT Dev, IOT App, Dash)
  - on cloud user kinds up to level of Tool Administrator
  - One or more ServiceMap and boundaries for the federation
- **ToolAdmin** users (requested by Organizations) may
  - control processes, consumption of resources, healthiness, etc.
  - manage tools exploited in your configuration

RootAdmin users may
- pass Users to higher roles. Ask to snap4city@disit.org to become an AreaManager for testing
- **Provide**/grant specific authorizations to data access on Tool usage
- In the Installation onPremise, you become the RootAdmin of it, you decide ALL.

RootAdmin users have a large set of tools in every field
- **Dashboards**
- **Extra Dashboard Widgets**
- **Notificator**
- **Data, My Data, OpenData**
- **Knowledge and Maps**
- **IOT Applications**
- **IOT Directory and Devices**
- **Resource Manager**
- **Development Tools**
- **Management**
- **Decision Support Systems**
- **Settings**
- **User Management and Auditing**
- **Help and Contacts**
- **Documentation and Articles**
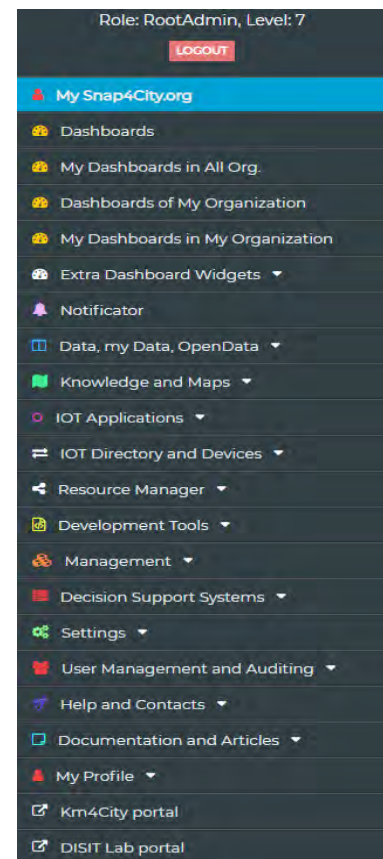
3.16.1 - Snap4City User Management

All that the RootAdmin needs to manage:

- **User Management: for managing**
  - accounts and profiles
  - limits of the users in exploiting resources
  - Accesses and providing special authorization
  - Organization vs Groups of users
  - Users vs Organizations
- **Users vs Web and Mobile Applications**
  - Engaging and monitoring users on platform and devices
- **Users on Chats room of Dashboards**
  - Managing Users on Chats of Dashboards
- **Auditing of the data and resource accesses**
  - Auditing all the activities on the platform (see next section)
  - Personal auditing

User Management and Auditing
- User Management
- User Limits Management
- User Engagement
- User Engagement Dash
- User Role Management via LDAP
- Manage Resource Ownership
- User Chats Management
- Auditing Data Access Try-out
- Auditing Elements vs Ownership
- Auditing Personal Data
- Auditing Accesses Authetication
- Auditing User Activities
- Auditing Activities on Queries
- Auditing Activities on Articles
- Auditing IOT Directory Data
- Dashboard Builder Local Users
- Organizations vs Groups
- Users vs Organizations

Customer Relationships management:

On CRM as Drupal 7 we have
- **User Management r**egistration and mailing
  - LDAP connection role management
  - KeyCloak Authentication (OpenID Connect)
  - Management of user profile
- **Content management** for Organizations and Groups
  - Indexing of all content and search
  - Distribution of content and Video
  - Reports and views
  - Tracking and monitoring
- **Open to full contributions and comments**
  - **Workflow, expandable as all the CRM**
- Etc.

Access to Personal Data types.

Control of resource consumption

Control of both resource consumption and data accesses, while allowing to set limits to the allocation of resources / artefacts to force the user to avoid unregulated consumption of resources.

Each user may monitor the consumption of resources with respect to the limits imposed on its account.

The user is informed when the limits are reached so that to stimulate online acquisition of new resources.
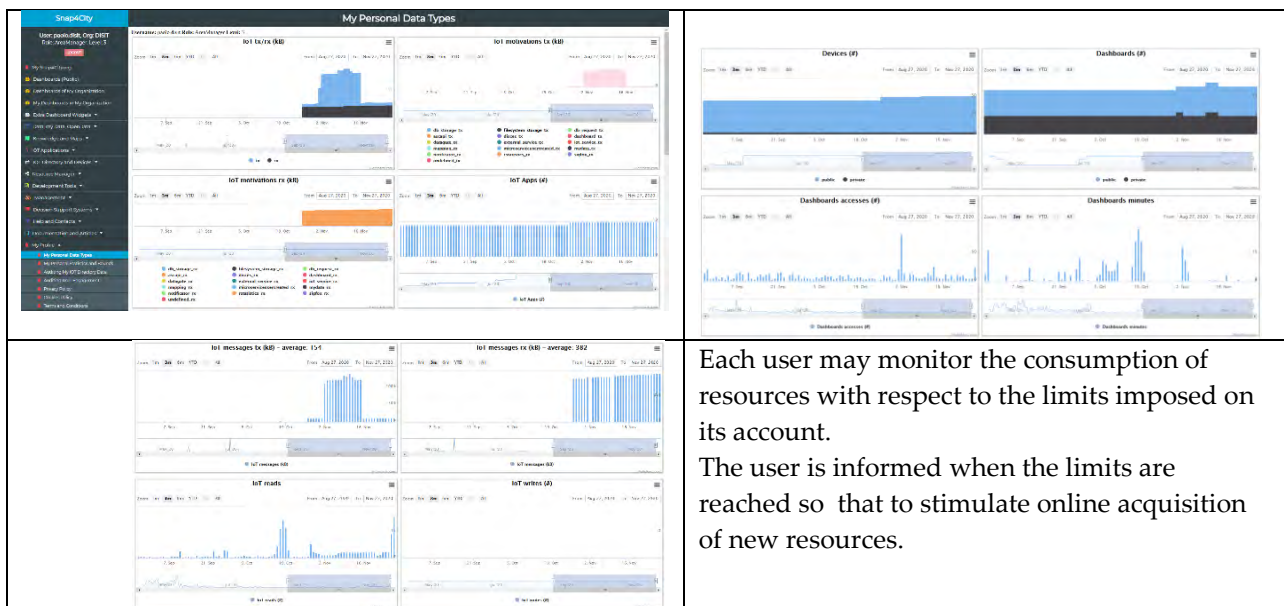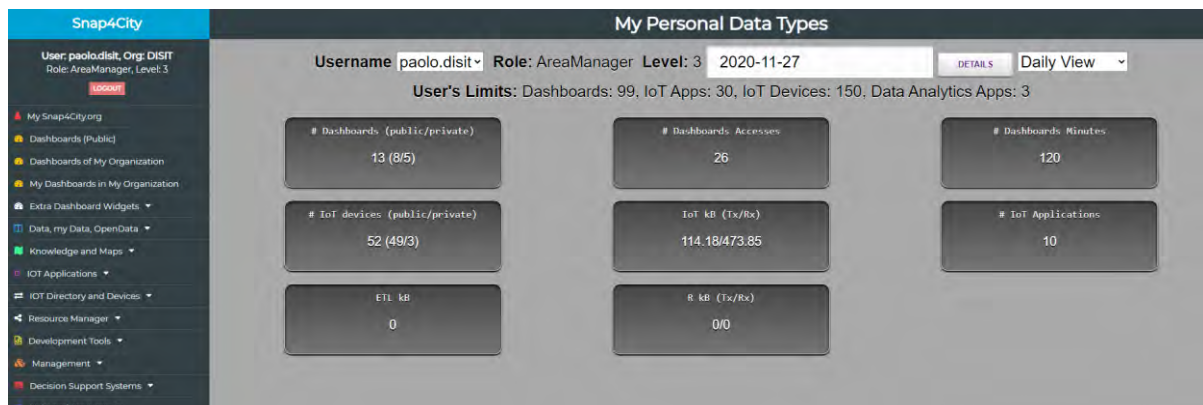
3.16.2 – Snap4City Platform Management, quality control

- **Tools for Platform Management.**
  - Most of tools are only accessible for RootAdmin and OnPremise
- **Tools can be grouped in the following families**
  - **AMMA** Traffic Analyzer (Elastic Search, Kibana), also log analysis.
  - DataAnalyzer (**DevDash**): monitoring and browsing data ingested into Elastic Search, via Kibana
  - Container Monitoring and Management
  - IOT App Version Management of Snap4City tools
  - Smart City API traffic monitoring
  - MyKPI Monitoring
  - DISCES schedulers monitoring and management (V1 infrastructure versions)
  - Mobile Applications Monitoring
  - Management of Images and Comments from Smart City API, Mobile and Web Apps
  - Management of OnLine Helps (not active)

AMMA and Log Analysis, quality control

The LOGs are coded in standard **SYSlog** format and can be monitored / accessed by using **AMMA** tool based on NIFI and Elastic Search, or **LogStash.**

DevDash and controlling the Elastic Search Storage and flow

Monitoring the Resources and API Traffic, quality control

http://www.disit.org/dashboardSmartCity/view/index.php?iddasboard=MTQ4

MyKPI monitoring:
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTY0NA==

Smart City API performance: http://www.disit.org/dashboardSmartCity/view/index.php?iddasboard=MTkw
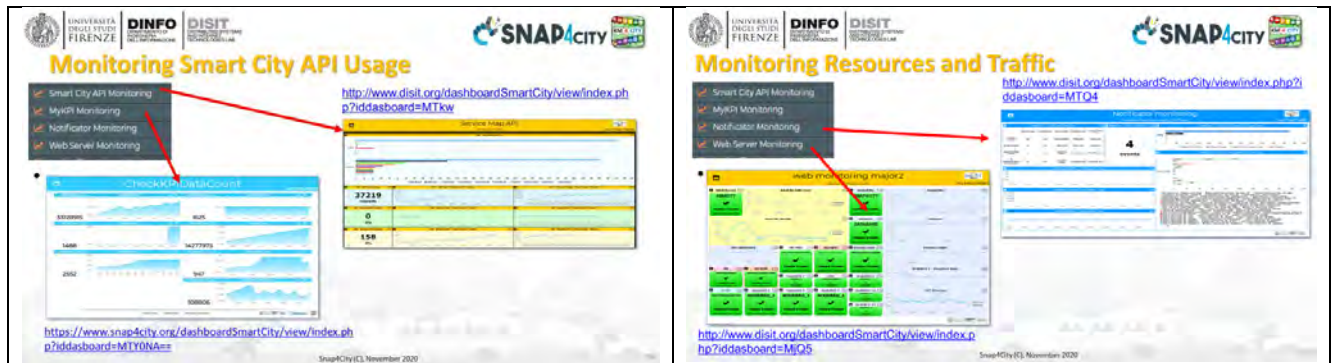
Monitoring Photos and Comments from Web and Mobile App
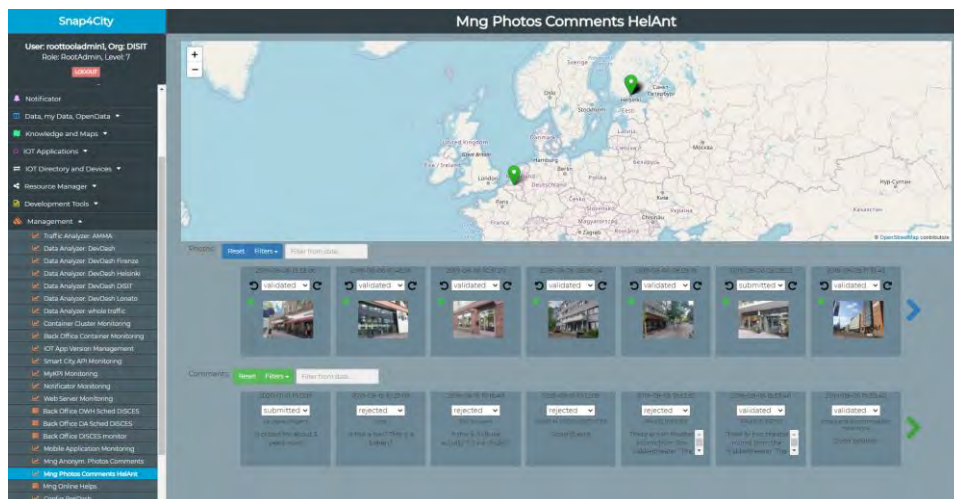
IOT App versions management (blue patches hide private data)

Mobile App monitoring: only internally accessible (for example)

https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc1OA==
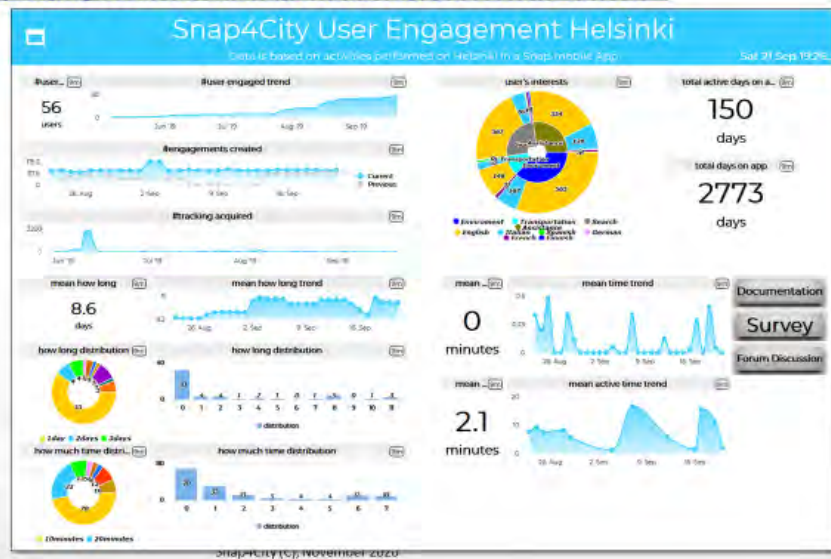
Other Monitoring and quality controls:

- ■ **Marathon and Mesos** Monitoring: only internally accessible (see Section 3.14)
- ■ **Cloud Services** Monitoring and control: from VMware, only internally accessible (see Section 3.14)
- ■ **API reachability / availability** performed by **E015 of Regione Lombardia**: https://www.snap4city.org/388
- ■ **DISCES** performance: https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjE3Mw==
- ■ WEB Servers Performance and monitoring: only internally accessible
- ■ **NIFI Monitoring**: an IoT App which uses NIFI API to monitor critical conditions and send alerts
- ■ **IoT Orion Broker Monitoring**: the broker provides some statistics on notifications and logs. https://FIWARE-orion.readthedocs.io/en/master/admin/perf_tuning/index.html
- ■ **HTTP server tuning** + Notification modes and performance

3.16.3 – Snap4City Platform Automation

Back office control and management most of the activities are automated: **Snap4City back office management, DRUPAL CMS Living Lab, Marathon, Mesos, IoT Orion Broker, DISCES-EM, IoT Directory, NIFI, Elastic Search for data shadow, OpenMAINT, user management, etc.**

Examples of automated activities:

- Harvesting of IoT Broker Devices and their registration in the platform
- registration of data models into **Knowledge Base**
- registration of data model to **Dashboard Wizard** and **Data Inspector**
- production of widgets from IoT App into Dashboards
- production of widget into dashboards from wizard and templates
- creation of the Data Shadow for IoT modelled data
- creation of the geo-spatial, temporal, and relational indexing
- creation and deploy of containers from Data Analytics in RStudio and Python
- Creation and deploy of containers with IoT Apps
- Production of Rest API Call for training developer in using the Smart City API

- Elastic management of Container over a group of VM
- Limiting the consumption of platform resources assigned to users
- Monitoring of API, scheduler, dashboard accesses
- Collection of data regarding user and platform auditing
- Logging of activities performed by IoT App, toward IoT Devices, MyKPI, etc.
- Production of reports
- Conversion of Data Model FiWare into IOT Data Model and knowledge base
- Production of GeoTiff from set of point heatmaps, for creating calibrated heatmaps
- Automated fitting of time trend on the basis of the data series available
- Etc………………..

3.17 – Comparison with other market and open sources solutions

Extracted from: Badii, C., Bellini, P., Difino, A., & Nesi, P. (2020). Smart City IoT Platform Respecting GDPR Privacy and Security Aspects. *IEEE Access*, *8*, 23601-23623. **[security2020]**

	IOT Discovery Abstraction	Authentication, Authorization	Security end-2-end, secure on IOT and Dashboards	Open HW and Open SW	Integrated Community management	Data Types: IOT Devices, IOT App, Dashboard, Data	Data Type: Publish/share, Delegation, Consent and change	Data Type: Download and Delete	Auditing on Data Type Access	Open Source end-to-end	Scalability IOT	Visual Programming end-to-end applications	Advanced Smart City API, MicroServices	Multi Domain Semantic Platform	Standard based Modules and IOT, Open Devices	Resource Sharing	Data Analytics integrated	Dashboard H24/7, protected connection	Multi-protocol on IOT
		G				G	G	G	G										
Snap4City	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
KAA [53]	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	N	Y	N	(Y)	N	N	Y	Y
Thingsboard [55]	Y	Y	Y	Y	N	Y	N	Y	Y	Y	Y	N	N	N	N	N	N	Y	MQTT, coap, http
IOT eclipse.org [56]	N	N	N	(Y)	N	Y	N	N	N	Y	Y	N	N	Y	N	N	N	N	Y
IOT IGNITE [57]	N	Y	N	Y	N	Y	N	Y	Y	Y	Y	Y	N	N	N	N	N	Y	MQTT
FIWARE [47]	N	Y	N	Y	N	N	N	Y	N	Y	(Y)	(N)	Y	N	Y	N	N	N	Y
ARM mbed IoT [48]	Y	Y	Y	Y	Y	N	(N)	N	Y	Y	Y	N	N	N	Y	N	N	Y	Limited
Airvantage [51]	Y	Y	Y	Y	N	Y	N	Y	Y	Y	Y	N	N	N	N	N	N	Y	MQTT, HTTP
AWS [43]	Y	Y	Y	Y	N	Y	(N)	Y	Y	N	Y	N	N	N	Y	Y	(Y)	Y	Limited
Azure IOT [44]	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	N	N	N	Y	Y	(Y)	Y	Limited
PTC ThingWorkx [59]	N	Y	Y	Y	N	Y	N	N	Y	N	Y	Y	N	N	Y	N	N	Y	Y
Bosch IoT Suite [58]	Y	Y	Y	Y	Y	(Y)	(N)	Y	Y	N	Y	Y	N	Y	N	Y	Y	Y	Y
CISCO Jasper [55]	Y	Y	Y	Y	N	(Y)	(N)	N	Y	N	Y	N	N	N	N	--	(Y)	Y	N
Siemens MindSphere [60]	Y	Y	Y	(Y)	N	Y	(N)	Y	Y	N	Y	Y	N	N	Y	N	Y	Y	Y
Carriots [54]	Y	Y	Y	(Y)	N	Y	N	N	Y	N	Y	N	N	N	--	N	N	Y	MQTT
Google IOT [45]	Y	Y	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	N	N	N	(Y)	(Y)	MQTT, HTTP
Homekit Apple [50]	Y	Y	Y	Y	N	Y	N	N	Y	N	(Y)	N	N	N	N	Y	N	Y	Limited
Smarthing Samsung [52]	Y	Y	Y	Y	N	Y	(Y)	Y	Y	N	(Y)	N	N	N	N	N	N	Y	Limited

4 – Terms and Acronyms of Snap4City

Term	Description
Access Token	An access token is an object encapsulating the security identity of a process or thread or of a user. In Snap4City is used for the M2M authentication. For example, when a process needs to access a smart city API via data analytics https://www.snap4city.org/650 or when your IoT App needs to access your devices.
AMMA	Application and MicroService Monitor and Analyzer. A Snap4City tool to perform the analysis of data flows among IoT Apps and the several MicroServices. See more on https://www.snap4city.org/198, https://www.snap4city.org/43 In alternative, standard tools may be used.
API	Snap4City API are classified in Internal and External. They are all API Rest and are documented in Swagger: https://www.km4city.org/swagger/external/index.html https://www.km4city.org/swagger/internal/index.html
ArcGIS	Is a commercial GIS solution, which has to provide WFS and WMS modules if not installed. https://www.arcgis.com/index.html ArcGIS is produced by ESRI that has a joined lab with **University of Florence LabGeo** and has been in collaboration with **University of Florence DISIT Lab since long time on Snap4City platform**.
BI	Business Intelligence. In Snap4City, the tool for BI is composed by Dashboard Builder, Wizard, IoT App, and for direct Dashboards with Kibana.
BIM	Business Information Modelling. In Snap4City, the tool for BIM is performed by OpenMAINT (https://www.openMAINT.org/en/home) and its integration with the BIMServer. https://github.com/opensourceBIM/BIMserver OpenMaint can be controlled by IoT Apps.
BPM	Business Process Management. In Snap4City this activity of defining business process workflows is modelled and performed by means of OpenMAINT tool. See OpenMAINT definition.
Calendar Widget	A widget for showing hourly or daily heatmaps about time series
Calibrated Heatmap	Calibrated heatmaps are heatmap with stable colors based on a color map not depending on the zoom level or on the different view colors are represented. In Snap4City, both calibrated and gaussian heatmap can be produced and distributed. Calibrated Heatmap are distributed via a GIS with WMS (for example as ArcGIS or GeoServer) and may have billions of points. https://www.snap4city.org/457
City Map Command and Control Center	See Smart City Control Room
CKAN	CKAN is an open source solution for Open Data management and distribution https://ckan.org/ In Snap4City is integrated with **DataGate** module and goes with SSO with the rest of tools. CKAN/DataGate can be controlled by IoT Apps.
Connectors	Connectors can be found/implemented in the Snap4City platform by means of different approaches, by using: • IoT App, since a large number of connectors with several protocols and formats are accessible on the Node-RED community. A large number is ready to use in the Snap4City IoT App on cloud and on IoT Edge as well. https://flows.nodered.org/ • IoT Agents of IoT Orion Broker of FIWARE, when they are present, they convert the format into NGSI. https://www.FIWARE.org/developers/catalogue/ For Snap4City, IoT App possibility is more flexible, since it permits to map and filter models and transform them, as well, when needed. The IoT App can be put in execution on Cloud and on IoT Edge as well, but all controlled with a visual environment from the main Snap4City interface.
Containers	In Snap4City Container are implemented as Dockers. They are used for Snap4City platform tools as described in https://www.snap4city.org/471 as well as for managing IoT App, Data Analytics in RStudio, Data Analytic in Python, and WebScraping processes.

Copernicus	Satellite data: pollution, weather, land, climate, atmosphere, security, emergency, etc. https://www.copernicus.eu/en
Custom Widget	In Snap4City, widgets are the components of the Dashboard to visualize views on data (with animation, graphics, synoptics, etc.) and also to collect interaction from the users (buttons, faces to be clicked, sliders, keypad, text pad, etc.). see HOW to create custom widgets in SVG, and examples: https://www.snap4city.org/651 https://www.snap4city.org/595 https://www.snap4city.org/644 https://www.snap4city.org/663 How to create widgets by programming: https://www.snap4city.org/153
D3	A library for JavaScript graphic representation: https://d3js.org/ This library may have some costs according to the licensing and usage.
Dashboard	is substantially a Decision Support System tool, since it provides evidence of critical conditions, and may offer solutions. On this regard, it may integrate/exploit artificial intelligence algorithms, for example, reporting prediction, identifying anomalies, manifesting early warning, providing relationships among entities exploiting inference geospatial reasoning about what is located in the city: resources, structure, people, areas, critical infrastructures, etc. See details on dashboard in section 3.10 of this document and from the training course part 2: https://www.snap4city.org/download/video/course2020/das/
Dashboard Structure	The structure of an applications starting from the Structure of the single Dashboard. The model is represented by a LOG graph, and/or a hierarchy of the model. A Dashboard included a number of Widgets. Each of which may refer to one or more data sources.
Dashboard Builder	In Snap4city, it is the main tool for creating Dashboards and connect them with IoT Apps, and other dashboards and custom widgets. See details on dashboard in section 3.10 of this document and from the training course part 2: https://www.snap4city.org/download/video/course2020/das/
Data Analytic	For Data Analytic we intend all deep data transformation on data that produce a new data kind: prediction, heatmap, anomaly detection alarm, traffic flow reconstruction, origin destination matrices, etc. In Snap4City, in the several installations, a large number of them have been developed in Rstudio, Python, Java, JavaScript, etc. In most cases, they exploit statistic, machine learning, data mining, artificial intelligence, semantic computing, etc. See for more info section 3.9 of this document and training course part 4: https://www.snap4city.org/download/video/course2020/da/
Data Dictionary	A Snap4City tool which allows to define Value_Type, Value_Unit and their relationship; and Nature and Subnature in their relationships. Snap4City platform is also provided with a set of more than 200 different attribute models into the Dictionary.
Data Inspector	In Snap4City, the Data Inspector is the main tool for the browsing of the information about a data stream entering into the platform. It is directly accessible from the main menu and give access to the Digital Twin representation of data, devices, and all **High Level Types** of the platform. See more details on part 5 of the training course: https://www.snap4city.org/download/video/course2020/di/
Data Shadow	A term adopted to describe the historical data of and IoT Device (sensors and actuators). In most of the platforms (for example: MS Azure, AWS), this feature is optional (please note that most of the Brokers provide only the last values of the IoT Devices). In Snap4City, it is a main feature to save all data messages of devices, and it is implemented saving data into Elastic Search cluster. See training part 3 https://www.snap4city.org/download/video/course2020/iot/
Data Table Loader	Data Table Loader for shortening the activities of registering devices, loading time series, etc., by loading the data from excel files. https://www.snap4city.org/729
POI Loader	POI Loader for shortening all the activities of ingesting new data for point of interest providing them as excel files.
Data_Type	Each Attribute/variable in Snap4City platform is defined in term of Value_Type, Value_Unit and Data_Type (e.g., Energy Power, Kw/h, Float).
DataGate, Data Gate	Module of Snap4City to integrate the solution with CKAN Open Data management and network of services
DevDash	A Snap4City tool for monitoring the global flow of data entering into the platform and reaching the Elastic Search. https://www.snap4city.org/198 https://www.snap4city.org/152 https://www.snap4city.org/145
Digital Twin	See Data Inspector which is the main tool for navigating into the information associated at each Digital Twin. The Digital Twin can be Local (BIM) and Global (3D modelling of the city buildings)
DISCES	Distributed scheduler for Smart City by Snap4City. It is used for scheduling Java processes in a distributed set of Nodes. It is part of the additional Suite for managing ETL processes. https://www.snap4city.org/236

DISCES-EM	Distributed scheduler for elastic management for Smart City by Snap4City. This tool is used for the elastic management of Container on the Marathon/Mesos cluster of containers. Read more on section 3.14 and on https://www.snap4city.org/232
Docker	Specific kind of Containers adopted by Snap4City, https://www.docker.com/
DockerHub	Socker Hub for Snap4City tools via DISIT lab https://cloud.docker.com/u/disitlab ,
Drupal CMS	A CMS, Content Management System, for Living Lab. See for its usage Https://www.snap4city.org and part 6 of the training course since it is the main support for the Living Lab: https://www.snap4city.org/download/video/course2020/sys/
End-2-End	Solution that allows to connect devices to the dashboards full stack. Snap4City can create end-2-end solutions, full stack, and secure with TSL, HTTPS, and Web Socket secure from devices to Dashboards.
EOSC	Snap4City is an official platform of EOSC (European Open Science Cloud) marketplace of the European Commission. https://marketplace.docker-fid.grid.cyf-kr.edu.pl/services/snap4city
ERP	Enterprise Resource Planning. Typically, the ERP may include BPM functionalities. Snap4City is integrated with openMAINT BPM which in turn is integrated with several ERP. Also IoT App are integrated with several ERP among them: https://flows.nodered.org/search?term=erp
ESB	Enterprise Service Bus, a modality for exchanging information among services. It has been replaced in most cases with MicroServices.
ExternalServices, External Services	Snap4City external services can be any external services reachable with some protocol to get data and services. REST Call can be automatically transformed in MicroServices for the IoT App. The other can be called from the IOT Ap directly using a large range of protocols: https://www.snap4city.org/65
Federated Knowledge Base	A set of Snap4City knowledge Basis connected each other via the so-called SuperServiceMap API. This allows the creation of mobile applications that may move from multiple cities and area accessing data and making queries transparently. This solution is presently in place among the **Knowledge Bases**: Antwerp/Helsinki, Tuscany/Firenze, Sardegna, etc. The resulting Service is called **SuperServiceMap** and it is integrated in the Smart City API. See **Section 3.8**. **Km4City ontology** data model https://www.snap4city.org/download/video/DISIT-km4city-City-Ontology-ita-v5-1.pdf
Federated Smart Cities	See Federated Knowledge Base
GDPR	General Data Protection Regulation of the European Commission. https://ec.europa.eu/info/law/law-topic/data-protection/reform/what-does-general-data-protection-regulation-gdpr-govern_en Snap4City is compliant with this directive and passed the assessment with a number of cities https://www.snap4city.org/670
GeoServer	GeoServer is an open source solution/tool for GIS data distribution. http://geoserver.org/ In Snap4City, the Geo Server is used into the HeatMap Server for distributing Heatmaps in GeoTiFF format according to tiled. https://www.snap4city.org/536 https://www.snap4city.org/507 GeoTiFF are used for distributing Orthomaps, Map and Heatmaps in form of images / tiles.
GIS	Geographic Information Server/service. A tool for modelling geo information. In Snap4City this role is covered by ServiceMap, and by the HeatMap Manager which includes a connection to a GIS (ArcGIS of GeoServer) or directly the GeoServer if needed. See https://www.snap4city.org/368 to see all relationships from GIS and Snap4City including interoperability.
Group	A Snap4City Group of User is a community into an Organization. Grant authorizations to resource access can be provided at level of single user, Group and/or Organization.
GTFS	General Transit Feed Specification, https://developers.google.com/transit/gtfs It is a standard file format by used to formalize the public transport information, trips, paths, busstops, time schedule, etc. Snap4City is compliant and can ingest GTFS files using ETL processes, the corresponding information is feed into Knowledge Base.
Heatmap	Are maps of points into Heatmap server, and/or directly images representing data in regular and non regular matrices. When they are in images are distributed via a GIS in WMS protocol as tiles (for example via ArcGIS or GeoServer). See https://www.snap4city.org/457 and https://www.snap4city.org/641 see Calibrated Heatmaps
HeatMap Server,	HeatMap Server exposes API for (i) colleting data regarding Heatmaps, (ii) providing information about the value of the map in any GPS point included, the so called heatmap picking, (iii) automated generation of Heatmaps in GeoTiFF format according to tiled which are distributed by a GIS via WMS

Heatmap Manager	protocol (they can be ArcGIS or GeoServer). https://www.snap4city.org/536 https://www.snap4city.org/507
High Level Types, HLT	They are the main data entity type managed by Snap4City Platform. They are: sensor, sensor actuator, virtual sensors, external services, MicroApplications, synoptics, MyKPI, personal data, WFS, Complex event, heatmaps, traffic flow, etc. see https://www.snap4city.org/583
HighCharts	A library for JavaScript graphic representation: https://www.highcharts.com/ This graphic library may have according to the usage some licensing conditions.
HLT	See High Level Types
IAM	Identity and Access Management. In Snap4City, this function is solved by KeyClock and LDAP open source tools, and it also provide SSO.
Innovatrix	A methodology for innovation at the ground of Snap4City Innovation model: https://www.imec-int.com/en/innovation/innovatrix
integrations	The instrument to implement inegrations is what is called in Snap4City terminology the IoT App, IoT Applications. SEE IoT App, see section 3.6 in this document. See training part 3 https://www.snap4city.org/download/video/course2020/iot/
IoT Adapter	A term adopted in IOT to indicate a remote converter of protocol located from the IoT Device and the IoT Broker. See training part 3 https://www.snap4city.org/download/video/course2020/iot/
IoT Agent	A term adopted in IOT to indicate a remote converter of protocol located from the IoT Device and the IoT Broker. See training part 3 https://www.snap4city.org/download/video/course2020/iot/
IoT App, IoT Application	Node-RED process + Snap4City Library of MicroServices SEE IoT App, see section 3.6 in this document. See training part 3 https://www.snap4city.org/download/video/course2020/iot/
IoT Application, IoT App	Node-RED process + Snap4City Library of MicroServices SEE IoT App, see section 3.6 in this document. See training part 3 https://www.snap4city.org/download/video/course2020/iot/
IoT Broker	An IoT Broker which may support one or more protocols, typically only one. It can support IoT Adapter, it can implement solution for MultiTenant and paths, such as IoT Orion Broker. Other IoT Broker can be: Mosquitto, HiveMQ, etc. https://en.wikipedia.org/wiki/Message_broker There are several brokers that may implement the same protocol such as https://en.wikipedia.org/wiki/Comparison_of_MQTT_implementations Snap4City is almost agnostic about the brokers and accept a number of brokers and protocols registered on IoT Directory of Brokers and Devices. See training part 3 https://www.snap4city.org/download/video/course2020/iot/
IoT Connector	A component to connect to a service using a protocol. In Snap4City, a large number of connectors and protocols are supported into the IoT App microservices. See Section 3.7, and interoperability web compliant page Https://www.snap4city.org/65
IoT Device	An IoT Device with sensors and/or actuators. In Snap4City, an IoT Device can be registered on IoT Broker before sending data on the platform. If the Broker is internal, the IoT Device can be registered on IoT Directory that perform all what is needed to register on IoT Broker and also on Knowledge Base and NIFI to automatically perform the Data Shadow of all the data produces by the device. This is possible since NIFI is automatically subscribed to all Devices of the Internal Brokers. See training part 3 https://www.snap4city.org/download/video/course2020/iot/
IoT Device Model	A model for an IoT Device, Virtual IoT Device, etc. In Snap4City, an IoT Device Model can be registered once and shared and used many times for instantiating one or many devices with the same model in short time. For example, 40.000 lights of a smart light system. Please note that the Model is a template, and once used can be modified without any impact of the produced instances. See training part 3 https://www.snap4city.org/download/video/course2020/iot/
IoT Device Mobile	Are IoT Devices which can change their GPS locations over time: https://www.snap4city.org/660
IoT Directory	Snap4City tool for registering IoT Broker and Devices. Snap4City is almost agnostic about the brokers and accept a number of brokers and protocols registered on IoT Directory of Brokers and Devices. The IoT Directory is capable to browse on internal and external brokers to discover IoT Devices and register them on Knowledge Base.

	https://www.snap4city.org/115 The IoT Directory is also the tool that manage all the network information about the devices deployed and connected. It is capable to exploit NGSI V1 and V2 protocol aspects to inspecting and managing IoT Orion Brokers of FIWARE, so that to manage a large number of them and automatically performing registration of devices in bulk on Knowledge base. See https://www.snap4city.org/76 https://www.snap4city.org/562, https://www.snap4city.org/647 See training part 3 https://www.snap4city.org/download/video/course2020/iot/
IoT Discovery	It is a function of Snap4City IoT Directory and Knowledge base which allow to discover the IoT Devices by a number of filters: geospatial, by type, by value name, by unit, by nature and subnature, etc- https://www.snap4city.org/109 The same Feature is provided in IoT App, IoT Directory, ServiceMap, Data Inspector and Dashboard Builder. See training part 3 https://www.snap4city.org/download/video/course2020/iot/
IoT Edge Device	An IoT Device capable to execute processes. In Snap4City terminology are typically endowed of Node-RED process which also has installed Snap4City Libraries of MicroServices. See section 3.5, and also ■ https://www.snap4city.org/646 for the remote control of IoT Edges ■ Edges on Linux https://www.snap4city.org/298 ■ Android https://www.snap4city.org/278 ■ raspberry pi. https://www.snap4city.org/279 ■ Snap4Home: https://www.snap4city.org/617 ■ Snap4industry: https://www.snap4city.org/369 ■ See training part 3 https://www.snap4city.org/download/video/course2020/iot/
IoT Orion Broker	FIWARE component: https://FIWARE-orion.readthedocs.io/en/master/ The IoT Orion Broker support natively the NGSI V1, V2 and LD. It usually provides only the last values of the IoT Devices and not the historical data (data shadow). In order to have the Data Shadow, and thus to perform queries on time series, the IoT Orion Broker has to be connected to a storage. In Snap4City, the storage provided is Elastic Search which provides high capabilities for insert, search and retrieval and scalability. See training part 3 https://www.snap4city.org/download/video/course2020/iot/
ISEMC for Video Wall	ISEMC for Video Wall management integrated with IoT App. (https://www.snap4city.org/621)
KB	See Knowledge Base
KeyCloak	For identify and access management, authentication and SAML, and SSO. https://www.keycloak.org/
Km4City Ontology	Km4City means Knowledge Model for the City. It is an Ontological model for the smart city and IoT Applications in smart city and many other domains. It is the core model adopted in Snap4City to design and implement the Knowledge Base, ServiceMap and the federation of **ServiceMaps** and smart city APIs. https://www.snap4city.org/19 https://www.snap4city.org/download/video/DISIT-km4city-City-Ontology-eng-v5-1.pdf
Knowledge Base, KB,	**Km4City Ontology** at the basis of the RDF Store. It is an expert system on the city data and entities in which smart city API can perform queries with spatial, temporal and relational reasoners. https://www.snap4city.org/19 https://www.snap4city.org/download/video/DISIT-km4city-City-Ontology-eng-v5-1.pdf
KPI	Key Performance Indicator, see MyKPI
LD	See LOD
LDAP	Lightweight Directory Access Protocol, for user registration role management https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol
Living Lab	Snap4City Living lab Support and Methodology, see Section 3.2. https://www.snap4city.org/206 https://www.snap4city.org/82
LOD, LD	Linked Open Data, Linked Data, see LOG.DISIT, LOG.DISIT.ORG tool for their reading https://en.wikipedia.org/wiki/Linked_data https://lod-cloud.net/
Log, LOGS	In Snap4City (as in many professional solutions and operating systems), LOGS/logs are produced reporting errors with different level of severity (e.g., warning, severe, etc.), they are produced in standard format as **SYSlog**. Those logs can be browsed, queried with tools, for example: standard **LogStash** which is based on Elastic Search, and Snap4City AMMA also based on Elastic Search.
LOG.DISIT, Log.disit.org	Linked Open Graph tool (noting to do with Logs). LOG.DISIT is a tool for accessing and browsing Linked Open Data in the world and in the local Knowledge base, KB. https://log.disit.org/service/

LogStash	See Log, https://www.elastic.co/logstash an open source tool for inspecting logs
Marathon	A Tool Open Source, standard for the management of containers https://mesosphere.github.io/marathon/
Market Place	Resource Manager of Snap4City https://www.snap4city.org/205
MESOS	A Tool Open Source, standard for the management of containers http://mesos.apache.org/
MicroApplications	A snap4City set of views implemented in HTML5 JavaScript for realizing specific functionalities. They are substantially views of Web and/or Mobile Apps which can be called independently and placed into Dashboard external content Widget as well as into Totems. https://www.snap4city.org/dashboardSmartCity/management/microApplications.php https://www.snap4city.org/54 https://www.snap4city.org/99
MicroServices	Snap4City platform is based on MicroServices. They are realized on the basis of the API (both internal and external). In Snap4City, the term MicroService is a synonym of Node in the Node-RED terminology. Each Snap4City node in the Snap4city Libraries for Node-RED is a MicroService of the platform. Other MicroServices can be easily added for customization and mapping of REST CALL APIs internal or of third party or of additional services, or from Container with Data Analytics. https://www.snap4city.org/22 https://www.snap4city.org/106 https://www.snap4city.org/129 For the recent list of MicroServices you have to see the documentation into the Library: https://flows.nodered.org/node/node-red-contrib-snap4city-user https://flows.nodered.org/node/node-red-contrib-snap4city-developer
MultiTenant	is a reference to the mode of operation of software where multiple organizations with their applications operate in a shared environment. The instances (tenants) are logically isolated, but physically integrated.
MyKPI, MyPOI	Snap4City tool for collecting and managing personal KPI, POI. A MyKPI is a variable with may change over time determining a TimeSeries with variable GPS position at each time instant. https://www.snap4city.org/396 https://www.snap4city.org/414
MyPOI	See MyKPI
NIFI Apache	Apache NIFI supports powerful and scalable directed graphs of data routing, transformation, and system mediation logic. https://nifi.apache.org/
Node-RED	A visual Editor for Node.JS processes from JS Foundation. https://nodered.org/ https://flows.nodered.org/
OpenMAINT	Workflow Management System, Incident Management, Business process Management, BIM and GIS integrated with Snap4City. The tool is capable to define workflow integrating activities of humans and machines with the main focus on maintenance and ticketing. It is presently integrated with IoT App of Snap4City. https://www.openMAINT.org/en/home
Organization	Snap4City Organizations inside the platform represent a tenant partitioning of maps and data and users. Typically, the users may belong to a single Organization with their email address and nickname. An Organization may have multiple Groups. A user may belong to multiple Groups. Grant authorizations to resource access can be provided at level of single user, Group and/or Organization.
Out of the Box	Component ready to be used without development, already in use in Snap4City applications and/or solutions.
Platform Management	See Quality Assessment and User Management and Control. **See Section 3.16.**
POI	Point of Interest, services on map, with some GPS location and service classification
Portia (web scraper)	A Tool for Web Scraping, extracting data from web pages. Processes of Portia in Snap4City are converted in MicroServices executed on containers. https://portia.readthedocs.io/en/latest/index.html
Python Server	Snap4City Python server for developing Python processes for machine learning, AI and statistical purpose on data. It can access to data via Smart City API and the Python processes can be transformed into Containers. They can exploit Tensor Flow and Keras and CUDA provided that specific NVIDIA boards are present on the servers, and VM can exploit them with some VGPU of the virtualization environment.
Quadruple Helix	describes university-industry-government-public-environment interactions within a knowledge economy. In innovation helix framework theory, first developed by Henry Etzkowitz and Loet

	Leydesdorff and used in innovation economics and theories of knowledge, such as the knowledge society and the knowledge economy, each sector is represented by a circle (helix), with overlapping showing interactions. https://en.wikipedia.org/wiki/Quadruple_and_quintuple_innovation_helix_framework
Quality Assessment	A process of quality control regarding several aspects of the platform. A quality control is also performed when the Reports are produced according to specific KPI. See **Section 3.16.** They are default KPI and Dashboards for quality control on Snap4City platform the views on: ■ API reachability / availability performed by E015 external service: https://www.snap4city.org/388 ■ Smart City API performance: https://dashboard.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTkw ■ DISCES performance: https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjE3Mw== ■ MyKPI monitoring: https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTY0NA== ■ Traffic Analyzer: see **AMMA** ■ Data Flow Global analyzer: see **DevDash** ■ Container Cluster healthiness: see section 3.14, only internally accessible ■ Mobile App monitoring: only internally accessible ■ WEB Server Performance and monitoring: only internally accessible ■ Marathon and Mesos Monitoring: only internally accessible ■ Cloud Services Monitoring and control: from VMware, only internally accessible ■ NIFI Monitoring: an IoT App which uses NIFI API to monitor critical conditions and send alerts ■ IoT Orion Broker Monitoring: the broker provides some statistics on notifications and logs. https://FIWARE-orion.readthedocs.io/en/master/admin/perf_tuning/index.html ■ HTTP server tuning + Notification modes and performance
Quality Assessment of IOT Data and services	A process of quality control regarding IOT data. Snap4City platform performs a control on the healthiness of the IoT Devices automatically on the basis of criteria set up on the IoT Directory. In this case, the control is performed at level of ServiceMap. An additional quality control may be performed in parallel to a dedicated process quality assessment. It is based on Machine Learning. This latter solution is optional for the large amount of resources needed to adopt it. A quality control can be also performed when the Reports are produced according to specific KPI which has to be defined and computed. **See Section 3.16.**
Reports	In Snap4City, the report generator can create consumptive views on the platform status on specific programmable aspects for users and administrators, for example at level of IoT Devices, Dashboards. Specific Reports can be created to produce quality assessment aspects.
Report Generator	A tool for generating reports about Device status over time: https://www.snap4city.org/720 It can be extended to generate reports on Dashboards and whole applications.
ResilienceDS	Resilience and sustained adaptability in urban transport systems (UTS) Today, enhancing resilience in Urban Transport Systems is considered imperative for two main reasons: a) such systems provide critical support to every socio-economic activity and are currently themselves one of the most important economic sectors in Europe; b) the paths that convey people, goods and information, are the same through which risks are propagated. ResilienceDS (https://www.snap4city.org/520).
Resource Manager	Is a Market Place of Snap4City artefacts: IoT App, Flows/subflows, data analytics, ETL, Kibana Dashboards, etc. They can be searched, shared and promoted via a web portal. https://www.snap4city.org/27 https://www.snap4city.org/205 https://www.snap4city.org/188 https://www.snap4city.org/134
Roles	Snap4City users are classified in Roles. Typical Roles are RootAdmin, ToolAdmin, AreaManager and Manager. Other Roles can be defined as well.
Routing, Travel plans	A detailed set of travel segments to start from Point A and reach Point B in the map at the certain time and day. It may be done by car, bus, and multimodal. A specific tool is needed to compute the travel plan, and detailed data are needed on the Knowledge Base to take decision about the planning. Some of the data can be recovered from OSM, Open Street Map, and their availability may depend on the geo zone.
Rstudio Server	Snap4City RStudio server (open source tool) for developing RStudio processes for machine learning, AI and statistical purpose on data. It can access to data via Smart City API and the RStudio processes can be transformed into Containers. They can exploit Tensor Flow and Keras and CUDA.

	https://rstudio.com/
SDK Mobile App	Software Development Kit for the production of Mobile Apps, exploiting the smart City API of Snap4City. Also this kit is provided in Open Source.
ServiceMap	Visual map interface to make Smart City API query on RDF store and test queries, and request samples of queries via email https://www.snap4city.org/19 https://www.snap4city.org/184 https://www.snap4city.org/155 https://www.snap4city.org/180
ServiceURI	In Snap4City terminology the service URI is the unique identifier of the Service and it is in substance an URI in the Linked Data model and Km4City Ontology and Expert system. It allows to identify univocally any entity of the city.
Smart City API	In Snap4City, a large collection of services to: exploit queries and reasoning on the storage and Knowledge Base, access/control IoT Network, exploit Data Analytic results, exploit IoT Apps, etc. All the data and services are accessible via the Smart City API which are used by Front End Tools such as Dashboards, Web and Mobile Apps, MicroApplications. Details regarding Smart City API are reported in: https://www.snap4city.org/download/video/course2020/app/
Smart City Control Room	A solution for centralized control of the smart city via a set of view wall and operator console of 3-4 monitors. See Florence Control Room https://www.snap4city.org/531 ISEMC for Video Wall management integrated with IoT App. (https://www.snap4city.org/621) See the following example of the Smart City Control Room, SCCR, of Florence Metropolitan City which has more than 1.5 million of inhabitants. The figure reports the main dashboard used by the Mayor (namely: Dario Nardella) and the second level dashboards. Please note that a third and a fourth level are present as well. https://www.snap4city.org/525 control room with video wall: https://www.snap4city.org/621
SmartDS	The **SmartDS** (Smart Decision System) of DISIT is an Advanced System Thinking solution for Decision Support System, DSS, on smart city problems and data. SmartDS is a tool presently in trial that allows you to model decision processes by using an Advances System Thinking formalism defining weights on branches and value of the Italian Flags probabilities on processes, etc. (the application of verification and validation algorithms on data are also provided). http://smartds.km4city.org/dss/ see (https://www.snap4city.org/520).
Snap4City Innovation Matrix.	The methodology for innovation of Smart City derived from the **Innovatrix** method and it has been adopted for the leveraging the Innovation into the European Commission JRC ISPRA in the 2019, in Pisa 2020, and other locations. A few details about its implementation and processes are reported in section 3.2 regarding the Living Lab in which the Methodology is largely used. See training course 2020 part 6. https://www.snap4city.org/download/video/course2020/sys/
Snap4Home	■ Snap4Home: https://www.snap4city.org/617 A subset of the Snap4City solution/platform which is suitable for smart home control and automation. It may include, Snap4City IoT App also installed at home, in some IoT Edge Device hosting Node-RED. See the list of supported protocols and those in Node-RED https://www.snap4city.org/65
Snap4industry	■ Snap4industry: https://www.snap4city.org/369 A subset of the Snap4City solution/platform which is suitable for smart industry/industry4.0 control and automation. It may include, Snap4City IoT App also installed in the factor or into the retail shops, and in particular in IoT Edge Device hosting Node-RED. They can be: raspberry Pi, Linux, windows, etc. See the list of supported protocols and those in Node-RED https://www.snap4city.org/65
Snp4City IoT App	See IoT App, IoT Application Node-RED process + Snap4City Library of MicroServices
SSM2ORION	SSM2ORION a module for connecting an IoT Orion Broker with its Data Shadow implemented by using Quantum Leap with the Federation of Knowledge Base. This solution allows to connect at the Smart City also other already in place FIWARE solutions which may have local storage. The queries performed on Snap4City Smart City API provide seamlessly the results also providing the geo data which are stored into that Orion Broker without the need to register the IoT Devices of that IoT Orion Broker into the Knowledge Base. Some limitations are present on the security aspects.

SSO	Single Sign On, In Snap4City, this function is performed by LDAP and KeyCloak at which all tool are referring to exploiting OpenID Connect.
SUMO	Simulation of Urban Mobility, open source tool, which can be integrated with the solution provided. https://www.eclipse.org/sumo/ https://sumo.dlr.de/docs/ https://en.wikipedia.org/wiki/Simulation_of_Urban_Mobility
SuperServiceMap	See **Federated Knowledge Base**. A tool on top of Smart City API, which is the API interface of ServiceMap, **Knowledge Base**
Swagger	Standard API documentation and design tool https://swagger.io/
Synoptics	See **Custom Widgets**
SYSLog	Standard format for Logs: https://en.wikipedia.org/wiki/Syslog
Time Series	Is a series of data values over time associated to a variable, sensor, actuator. A Time Serie may have change GPS location of measure, and in that case is a Moving Object Time Series. In Snap4City, all the variable of IoT Devices, area HTL Sensors/Actuator and may be Time Series; also MyKPI variable may be Moving Object Rime Series.
Tracker	A Specific Widget for tracking moving devices, which can be located into Snap4City Dashboards.
Traffic Flow Reconstruction	Algorithm and Tool for computing the traffic flow in ay point of the city on the basis of the data collected from a limited number of sensors located on roads and scattered in the city.
Traffic Flow Manager	A tool for collecting and distributing Traffic Flows. Similar to Heatmap Manager but for traffic flows.
Typical Time Trends	A typical trend of the data, highlighting one or more seasonality aspects of the data. They can be daily, weekly, weekly on day, monthly, etc.
User Management and control	■ User vs registration: LDAP and KeyCloak ■ User Limits management ■ User vs consumption of resources ■ Content vs publication ■ Auditing data access try-out, Elements and Ownerships, personal data, accesses authentications, user activities, queries, articles, web pages, dashboards, IoT Directory, etc- ■ Org vs Groups, user vs orgs ■ Chat management **See Section 3.16.**
Value_type	Each Attribute/variable in Snap4City platform is defined in term of Value_Type, Value_Unit and Data_Type (e.g., Energy Power, Kw/h, Float). They can be defined with the Data Dictionary.
Value_Unit	Each Attribute/variable in Snap4City platform is defined in term of Value_Type, Value_Unit and Data_Type (e.g., Energy Power, Kw/h, Float). They can be defined with the Data Dictionary.
Virtual IoT Devices	An IoT Device which has not a physical counterpart. It can be a just defined in the IoT Broker for passing data from one service to another. Virtual IoT Devices having sensors and actuators can be also the Widgets on Dashboards by which a user can see value and act on them to send values.
VM	Virtual Machine
Web Socket Secure, WSs	A TLS version of the WS. In Snap4City it is used for communicating from Client Dashboards, Custom widgets, event driven widgets, etc., to the platform on which IoT App and MyKPI/storage are connected in real time.
Web Socket Variable	A single Variable connected via WSs
Web Socket. WS	A communication protocol for real time connection. In Snap4City is only used in its TLS version Web Socket Secure.
What-IF analysis	The What-IF analysis is a modality of work recently included in the Snap4City suite which aim to exploit in a multicontextual environment most of the former tools as dashboards and Data Analytic with IoT App intelligence. See Section 3.9.2
WS	See Web Socket
WS Secure	See Web Socket Secure, WSs

5– References

5.1 - Published Material on TOP level Journals and Conferences with International Reviewers

[ShortTermPredictionBikeSharing2021] E. Collini, P. Nesi, G. Pantaleo, "Deep Learning for Short-Term Prediction of Available Bikes on Bike-Sharing Stations", under publication on IEEE Access, pp.1-11, Print ISSN: 2169-3536, Online ISSN: 2169-3536, DOI: https://doi.org/10.1109/ACCESS.2021.3110794

[ChemicalPlant2021] P. Bellini, D. Cenni, N. Mitolo, P. Nesi, G. Pantaleo, M. Soderi, "High Level Control of Chemical Plant by Industry 4.0 Solutions", Journal of Industrial Information Integration, Elsevier. https://doi.org/10.1016/j.jii.2021.100276

[LongTermPredictionBikeSharing2021] D. Cenni, E. Collini, P. Nesi, G. Pantaleo, I. Paoli, "Long-Term Prediction of Bikes Availability on Bike-Sharing Stations", Journal of Visual Languages and Computing, KSI. https://doi.org/10.18293/jvlc2021-n1-001

[SatelliteData2021] P. Belllini, D. Cenni, N. Mitolo, P. Nesi, G. Pantaleo, "Exploiting Satellite Data in the Context of Smart City Applications", The 5th IEEE International Conference on Smart City Innovations - Track 1: Theory, Modeling and Methodologies, 2021. http://ieeesmartworld.org/sci/

[PredictiveMaintenance2021] P. Bellini, D. Cenni, L. A. Ipsaro Palesi, P. Nesi, G. Pantaleo, "A Deep Learning Approach for Short Term Prediction of Industrial Plant Working Status", 7th IEEE International Conference on Big Data Service and Machine Learning Applications, 23-26 August, 2021. http://www.big-dataservice.net/index.html

[15MinCityIndex2021] C. Badii, P. Bellini, D. Cenni, S. Chiordi, N. Mitolo, P. Nesi, M. Paolucci, "Computing 15MinCityIndexes on the basis of Open Data and Services", Proc. of the 2021 International Conference on Computational Science and Its Applications. Published on LNCS Springer. https://iccsa.org/
https://www.snap4city.org/drupal/sites/default/files/files/computing15minCityIndex_ICCSA_v0-3.pdf
web page: https://www.snap4city.org/drupal/node/652

[LongTermPredictionNO2-2021] P. Bellini, S. Bilotta, D. Cenni, E. Collini, P. Nesi, G. Pantaleo, M. Paolucci, "Long Term Predictions of NO2 Average Values via Deep Learning", Proc. of the 2021 International Conference on Computational Science and Its Applications. Published on LNCS Springer. https://iccsa.org/

[UserBehaviourModalities2021] C. Badii, A. Difino, P. Nesi, I. Paoli, M. Paolucci, "Classification of Users Transportation Modalities from Mobiles in Real Operating Conditions", Multimedia Tools and Applications, Springer, 2021. https://doi.org/10.1007/s11042-021-10993-y or as a PDF here https://link.springer.com/content/pdf/10.1007/s11042-021-10993-y.pdf

[Resilience2021] E. Bellini, P. Bellini, D. Cenni, P. Nesi, G. Pantaleo, I. Paoli, M. Paolucci, "An IoE and Big Multimedia Data approach for Urban Transport System resilience management in Smart City", Sensors, MDPI, 2021,
https://www.mdpi.com/1424-8220/21/2/435/pdf

[AnomalyDetection2020] P. Bellini, D. Cenni, P. Nesi, "Anomaly Detection on IOT Data for Smart City", Proc of 6th IEEE International Workshop on Sensors and Smart Cities, with IEEE SmartComp, 14-17 Sept. 2020, Bologna, Italy. http://ssc2020.unime.it/
Https://www.snap4city.org/download/video/AnomalyDetection2020.pdf

[BIgDataService2018] P. Nesi, P. Bellini, M. Paolucci, I. Zaza, "Smart City architecture for data ingestion and analytics: processes and solutions", IEEE BigDataService 2018, Bamberg, Germany, March 26 - 29, 2018. Https://www.snap4city.org/download/video/BIgDataService2018.pdf

[COVID-19] C. Badii, P. Bellini, S. Bilotta, D. Bologna, D. Cenni, A. Difino, A. Ipsaro Palesi, N. Mitolo, P. Nesi, G. Pantaleo, I. Paoli, M. Paolucci, M. Soderi, "How COVID-19 Lockdown Impacted on Mobility and Environmental data", Bollettino della Società Geografica Italiana, FuPress, June 2020, DOI: 10.13128//bsgi.v2i2.932   https://drive.google.com/file/d/1hN4dFuEXLMLq4rMY8u88iIZnrVk3HPIl/view
Https://www.snap4city.org/download/video/COVID-19.pdf

[DashboardProduction2020] Q. Han, P. Nesi, G. Pantaleo, I. Paoli, "Smart City Dashboards: Design, Development and Evaluation", Proc. Of the IEEE ICHMS 2020, International Conference on Human Machine Systems, September 2020. http://ichms.dimes.unical.it/ Https://www.snap4city.org/download/video/DashboardProduction2020.pdf

[Dashboards2019] P. Bellini, D. Cenni, M. Marazzini, N. Mitolo, P. Nesi, M. Paolucci, "Smart City Control Room Dashboards: Big Data Infrastructure, from data to decision support", Journal of Visual Languages and Computing, 10.18293/VLSS2018-030 https://ksiresearchorg.ipage.com/vlss/journal/VLSS2018/paper%2030.pdf Https://www.snap4city.org/download/video/Dashboards2019.pdf

[DataFlow2019] P. Bellini, F. Bugli, P. Nesi, G. Pantaleo, M. Paolucci, I. Zaza, "Data Flow Management and Visual Analytic for Big Data Smart City/IOT", 19th IEEE Int. Conf. on Scalable Computing and Communication, IEEE SCALCOM 2019, Leicester, UK https://www.slideshare.net/paolonesi/data-flow-management-and-visual-analytic-for-big-data-smart-cityiot Https://www.snap4city.org/download/video/DataFlow2019.pdf

[DI-DataInspectorBase] P. Bellini, D. Bologna, Q. Han, P. Nesi, G. Pantaleo. M. Paolucci, "Data Ingestion and Inspection for Smart City Applications", proc. Of IEEE International Conference SMARTCOMP 2020, Bologna, Italy, 2020. http://www.smart-comp.org/ Https://www.snap4city.org/download/video/DI-DataInspectorBase.pdf

[DORAM2020] A. Arman, P. Bellini, P. Nesi, M. Paolucci, "Analysing Public Transportation Offer wrt Mobility Demand", ACM Workshop on Technology Enablers and Innovative Applications for Smart Cities and Communities (TESCA 2019), November 10, 2019 at Columbia University, New York, USA. https://dl.acm.org/citation.cfm?id=3364828 https://doi.org/10.1145/3364544.3364828 https://dl.acm.org/ft_gateway.cfm?id=3364828&ftid=2101673&dwn=1&CFID=180... Https://www.snap4city.org/download/video/DORAM2020.pdf

[FederatedKnowledgeBase2020] P. Bellini, D. Nesi, P. Nesi, M. Soderi, "Federation of Smart City Services via APIs", Proc of 6th IEEE International Workshop on Sensors and Smart Cities, with IEEE SmartComp, 14-17 Sept. 2020, Bologna, Italy. http://ssc2020.unime.it/ Https://www.snap4city.org/download/video/FederatedKnowledgeBase2020.pdf

[Gov2018] M. Azzari, C. Garau, P. Nesi, M. Paolucci, P. Zamperlin, "Smart City Governance Strategies to better move towards a Smart Urbanism", The 18th International Conference on Computational Science and Its Applications (ICCSA 2018), July 2 – 5, 2018 in Melbourne, with the Monash University, Australia. Https://www.snap4city.org/download/video/Gov2018.pdf

[Gov2020] C. Garau, P. Nesi, I. Paoli, M. Paolucci, P. Zamperlin, A Big Data Platform for Smart and Sustainable Cities: Environmental Monitoring case studies in Europe. Proc. Of International Conference on Computational Science and its Applications, ICCSA2020. Cagliari, Italy, 1-4 July 2020. http://www.iccsa.org/   https://link.springer.com/chapter/10.1007%2F978-3-030-58820-5_30 Https://www.snap4city.org/download/video/Gov2020.pdf

[Heatmap2020] C. Badii, S. Bilotta, D. Cenni, A. Difino, P. Nesi, I. Paoli, M. Paolucci, "High Density Real-Time Air Quality Derived Services from IoT Networks", Sensors, MDPI, 2020. https://www.mdpi.com/1424-8220/20/18/5435/htm Https://www.snap4city.org/download/video/Heatmap2020.pdf

[Industry4.0-2020] C. Badii, P. Bellini, D. Cenni, N. Mitolo, P. Nesi, G. Pantaleo, M. Soderi, "Industry 4.0 Synoptics Controlled by IoT Applications in Node-RED", Proc. Of the IEEE iThings, Nov 02-06, 2020. Https://www.snap4city.org/download/video/Industry40-2020.pdf

[KnowledgeBase2018] P. Bellini, P. Nesi, "Performance Assessment of RDF Graph Databases for Smart City Services", Journal of Visual Language and Computing, Elsevier, 2018. https://doi.org/10.1016/j.jvlc.2018.03.002 Https://www.snap4city.org/download/video/KnowledgeBase2018.pdf

[LivingLab2018] P. Nesi, M. Paolucci, "Supporting Living Lab with Life Cycle and Tools for Smart City Environments", The 24th International DMS Conference on Visualization and Visual Languages, DMSVIVA 2018, Hotel Pullman, Redwood City, San Francisco Bay, California, USA, June 29 – 30, 2018

Https://www.snap4city.org/download/video/LivingLab2018.pdf

[MicroServices2019] C. Badii, P. Bellini, A. Difino, P. Nesi, G. Pantaleo, M. Paolucci, "MicroServices Suite for Smart City Applications", Sensors, MDPI, 2019. https://doi.org/10.3390/s19214798
Https://www.snap4city.org/download/video/MicroServices2019.pdf

[MobileDevKit2017] C. Badii, P. Bellini, P. Nesi, M. Paolucci, "A Smart City Development kit for designing Web and Mobile Apps", IEEE international Conference on Smart City and Innovation, 2017, San Francisco. https://www.slideshare.net/paolonesi/a-smart-city-development-kit-for-designing-web-and-mobile-apps
Https://www.snap4city.org/download/video/MobileDevKit2017.pdf

[MobilityMicroServices2019] C. Badii, P. Bellini, A. Difino, P. Nesi, "Sii-Mobility: an IOT/IOE architecture to enhance smart city services of mobility and transportation", Sensors, MDPI, 2019. https://doi.org/10.3390/s19010001   https://www.mdpi.com/1424-8220/19/1/1/pdf
Https://www.snap4city.org/download/video/MobilityMicroServices2019.pdf

[NOX prediction 2020] C. Badii, S. Bilotta, D. Cenni, A. Difino, P. Nesi, G. Pantaleo, I. Paoli, M. Paolucci, "Real-Time Automatic Air Pollution Services from IOT Data Network", proc. Of IEEE Symposium on Computers and Communications (ISCC), MOCS track, 10th Workshop on Management of Cloud and Smart City System, 2020 July 7th, Rennes, France. https://conferences.imt-atlantique.fr/iscc2020/
Https://www.snap4city.org/download/video/NOXprediction 2020.pdf

[Resilience2017] E. Bellini, P. Ceravolo, P. Nesi, "Quantify resilience enhancement of UTS through exploiting connected community and internet of everything emerging technologies", 2017, http://hdl.handle.net/2158/1105460 , ACM TRANSACTIONS ON INTERNET TECHNOLOGY https://dl.acm.org/citation.cfm?id=3137572
Https://www.snap4city.org/download/video/Resilience2017.pdf

[Resilience2019] E. Bellini, L. Coconea, P. Nesi, "A Functional Resonance Analysis Method driven Resilience Quantification for socio-technical System", IEEE Systems Journal, DOI 10.1109/JSYST.2019.2905713, ISSN: 1932-8184, ISSN: 1937-9234, pp.1-11, 2019 https://ieeexplore.ieee.org/document/8686178
Https://www.snap4city.org/download/video/Resilience2019.pdf

[Scalable2018] C. Badii, E. G. Belay, P. Bellini, D. Cenni, M. Marazzini, M. Mesiti, P. Nesi, G. Pantaleo, M. Paolucci, S. Valtolina, M. Soderi, I. Zaza, "Snap4City: A Scalable IOT/IOE Platform for Developing Smart City Applications", Int. Conf. IEEE Smart City Innovation, Cina 2018, IEEE Press. https://ieeexplore.ieee.org/document/8560331/
Https://www.snap4city.org/download/video/Scalable2018.pdf

[SCAPI-KB2017] C. Badii, P. Bellini, D. Cenni, A. Difino, P. Nesi, M. Paolucci, "Analysis and Assessment of a Knowledge Based Smart City Architecture Providing Service APIs", Future Generation Computer Systems, Elsevier, 2017, http://dx.doi.org/10.1016/j.future.2017.05.001
Https://www.snap4city.org/download/video/SCAPI-KB2017.pdf

[Security2020] C. Badii, P. Bellini, A. Difino, P. Nesi, "Smart City IoT Platform Respecting GDPR Privacy and Security Aspects", IEEE Access, 2020. 10.1109/ACCESS.2020.2968741 https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8966344
Https://www.snap4city.org/download/video/Security2020.pdf

[SmartDS] M. Bartolozzi, P. Bellini, P. Nesi, G. Pantaleo and L. Santi, "A Smart Decision Support System for Smart City," 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, 2015, pp. 117-122, December 2015, Cina, IEEE press, doi: 10.1109/SmartCity.2015.57 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7463711&isnumber=7463653
Https://www.snap4city.org/download/video/SmartDS.pdf

[SmartParking2018] C. Badii, P. Nesi, I. Paoli, "Predicting available parking slots on critical and regular services exploiting a range of open data", IEEE Access, 2018, https://ieeexplore.ieee.org/abstract/document/8430514/
Https://www.snap4city.org/download/video/SmartParking2018.pdf

[TrafficFlowReconstruction2020] S. Bilotta, P. Nesi, "Traffic Flow Reconstruction by Solving Indeterminacy on Traffic Distribution at Junctions", Future Generation Computer Systems, Elsevier, 2020, 2021. https://authors.elsevier.com/sd/article/S0167-739X(20)30835-9

Https://www.snap4city.org/download/video/TrafficFlowReconstruction2020.pdf

[TrafficPredicitons2020] S. Bilotta, P. Nesi, I. Paoli, "Real-Time System for Short- and Log-Term Prediction of Vehicle Flow", IEEE Smart Data Service as Proc of IEEE Services, 2020.
Https://www.snap4city.org/download/video/TrafficPredicitons2020.pdf

[TwitterVigilance2017] D. Cenni, P. Nesi, G. Pantaleo, I. Zaza, "Twitter Vigilance: a Multi-User platform for Cross-Domain Twitter Data Analytics, NLP and Sentiment Analysis", IEEE international Conference on Smart City and Innovation, 2017, San Francisco.https://www.slideshare.net/paolonesi/twitter-vigilance-a-multiuser-platform-for-crossdomain-twitter-data-analytics-nlp-and-sentiment-analysis
Https://www.snap4city.org/download/video/TwitterVigilance2017.pdf

[UserBehavior2020] C. Badii, A.Difino, P. Nesi, I. Paoli, M. Paolucci, "Classification of Users' Transportation Modalities in Real Conditions", Proc. Of the 26th International DMS Conference on Visualization and Visual Languages, DMSVIVA, Wyndham Pittsburgh University Center, Pittsburg, USA, July 7-8, 2020.
http://ksiresearch.org/seke/dmsviva20.html
Https://www.snap4city.org/download/video/UserBehavior2020.pdf

[UserBehaviour2017] P. Bellini, D. Cenni, P. Nesi, I. Paoli, "Wi-Fi Based City Users' Behaviour Analysis for Smart City", Journal of Visual Language and Computing, Elsevier, 2017.
http://www.sciencedirect.com/science/article/pii/S1045926X17300083
Https://www.snap4city.org/download/video/UserBehaviour2017.pdf

[UserEngagement2017] C. Badii, P.Bellini, D.Cenni, A. Difino, P. Nesi, M. Paolucci, "User Engagement Engine for Smart City Strategies", IEEE International Conference on Smart Computing, IEEE SMARTCOMP 2017, Hong Kong.
Https://www.snap4city.org/download/video/UserEngagement2017.pdf

5.2 – Snap4City Main documentation web pages

Training Course page	https://www.snap4city.org/577 see this link to get the updated versions
Training Part 1 - overview	https://www.snap4city.org/download/video/course2020/gen/Snap4City-1st-Overview-2021-v3-4.pdf
Training Part 2 - Dashboard	https://www.snap4city.org/download/video/course2020/das/Snap4City-2nd-slot-dashboard-building-v7-9.pdf
Training Part 3 – IoT App and networks	https://www.snap4city.org/download/video/course2020/iot/Snap4City-3rd-slot-IOT-Applications-v8-0.pdf
Training Part 4 – Data Analytics	https://www.snap4city.org/download/video/course2020/da/Snap4City-4th-slot-Data-Analytic-v5-2.pdf
Training Part 5 – Data Ingestion	https://www.snap4city.org/download/video/course2020/di/Snap4City-5th-slot-data-ingestion-v7-1.pdf
Training Part 6 – System deploy, summary	https://www.snap4city.org/download/video/course2020/sys/Snap4City-6th-slot-system-deploy-v6-7.pdf
Training Part 7 – API and Mobile Apps	https://www.snap4city.org/download/video/course2020/app/Snap4City-7th-day-Mobile-Applications-v3-7.pdf

5G experimentation and trial in Italy for Prato city with Wind-3G Open Fiber Operator:	https://www.snap4city.org/547
DockerHub DISIT	https://cloud.docker.com/u/disitlab
E015 official APIs of Digital Ecosystem of Regione Lombardia	https://www.e015.regione.lombardia.it/site/app-detail?id=156
Elastic	https://www.snap4city.org/232
EOSC European Open Science Cloud official service:	https://marketplace.eosc-portal.eu/services/snap4city

Fi-Ware Official Solution	https://marketplace.FIWARE.org/pages/solutions/b8905e91973b420189cce972
Flexibility	https://www.snap4city.org/4
GitHub DISIT lab	https://github.com/disit
Hackathon 2019	https://www.snap4city.org/hackathon
Hackathon 2021	https://www.snap4city.org/757
Interoperability	https://www.snap4city.org/283
Km4City Smart City IOT Ontology	https://www.snap4city.org/download/video/DISIT-km4city-City-Ontology-ita-v5-1.pdf
Node-RED official library Snap4City	https://flows.nodered.org/search?term=snap4city
NPM	https://www.npmjs.com/~disit https://www.npmjs.com/package/node-red-contrib-snap4city-user
Open sources	https://www.snap4city.org/7
Protocols	https://www.snap4city.org/65
Scenarious	https://www.snap4city.org/4
Security	https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8966344
SmartBED	https://www.ifc.cnr.it/images/Ricerca/progetti/LAID.pdf https://www.snap4city.org/668

5.3 – Scenarios

- Scenario: Smart Light Control (CAPELON)
- Scenario: Smart Parking in Lonato del Garda (Reverberi)
- ALMA FLUIDA activity (ITALMATIC)
- SMART AMBULANCE activity started
- AMPERE Project - Furniture, art and fashion in an innovative electrochemical process with remote control and circular eco-friendly
- Scenario: SnapBot: Real Time Smart City services via Telegram
- Scenario: Copernicus Satellite Data
- Scenario: SmartBed, Materasso Intelligente
- Snap4City & Snap4Industry Registered Instances
- MicroServices Suite for Smart City Applications
- Scenario: MODBUS for Snap4Industry Snap4City Applications
- Scenario: MOBIMART Interreg: MOBilità Intelligente MARe Terra
- Scenario: City of Roma case, mobility and environmental data
- Scenario: Herit-Data video and aims
- Scenario: Control Room vs Video Wall
- Scenario: Snap4Home the case of: Alexa, Philips, Sonoff, TP-link, etc. (Italiano)
- Scenario: how to manage maintenance and accidents workflows
- Scenario: Snap4Home, how to exploit Snap4City solution on home automation
- Scenario: Energy Monitoring
- Scenario: Multipurpose User Engagement Tools
- Scenario: 5G Enabled Water Cleaning Control (smart city, industry 4.0)
- Scenario: High Level Control of Industrial Plant (industry 4.0)
- Scenario: Vehicle Monitoring via OBD2
- Data Analytic: Origin Destination Matrices, Algorithms and tools
- Data Analytic: Traffic Flow Reconstruction

- Data Analytic: in general, and the cases of Antwerp and Helsinki

- Data Analytic: Predicting Air Quality
- Data Analytic: Analyzing Public Transportation Offer wrt Mobility Demand, DORAM Tool

- Scenario: Events and Museums Monitoring in Antwerp

- Scenario: High Resolution Prediction of Environmental Data

- Scenario: Mobility and Transport Analyses in multiple cities

- Scenario: People Flow Analysis via Wi-Fi

- Scenario: Antwerp Pilot on Environmental Data

- Scenario: Helsinki Pilot on Environmental Data

- Scenario: Firenze Smart City Control Room

- Scenario: Mobile & Web App: Toscana Where What … Km4City, Toscana in a Snap

- Scenario: Helsinki Pilot on User Behaviour

- Scenario: Antwerp Pilot on User Behaviour
- Smart City Control Room Dashboards: Big Data Infrastructure, from data to decision support
- Dashboards Touch Screen Support for Situation Rooms and What-iF analysis Operators
- Data Analytics on Antwerp and Helsinki
- Decision Support Engines and Models: SmartDS, ResilienceDS
- IPCAM on Dashboard Widget, Video Cameras
- Solution: using PAX Counters, monitoring museum and events
- Solution: selecting the less polluted routing
- Data Flow Management and Visual Analytic for Big Data Smart City/IOT
- Snap4City on Social Media

- AIRQino: Compact low-cost air quality monitoring station, by CNR IBIE
- Security, Privacy and GDPR Aspects of Snap4City
- How to: Develop Smart Applications, Snap4City development Life Cycle

5.4 – Main "How To" Guidelines

- HOW TO: create a Dashboard in Snap4City
- HOW TO: add a device to the Snap4City Platform
- HOW TO: add data sources to the Snap4City Platform
- HOW TO: define privacy rules for personal data, produced by the end-users own device
- HOW TO: Develop Smart Applications, Snap4City development Life Cycle
- HOW TO: HLT vs Ingestion, and HLT vs Widgets
- HOW TO: Develop an IoT Application for Data Ingestion

- [HOW TO: Upload data into Knowledge Base, ServiceMap (triple upload)](#)
- [HOW TO: Create as set of Devices with BulkProcessing](#)
- [HOW TO: Create an IoT Device Model](#)
- [HOW TO: Create an IoT Device Instance from IoT Directory tool](#)
- [HOW TO: save / export a MyKPI data into a CSV file](#)
- [HOW TO: develop DataAnalytic in Python and manage them via IoT App](#)
- [HOW TO: customize AIRQino IoT Device to work on Snap4City](#)
- [HOW TO: setup Libelium Waspmote Plug&Sense Smart Environment IoT Device to work on Snap4City](#)
- [MicroServices Suite for Smart City Applications](#)
- [HOW TO: create subflow and groups into IoT Applications, Node-RED](#)
- [HOW TO: define Device in mobility, Mobile IOT Devices, Sensor Mobile](#)
- [HOW TO: Manage IOT Network Components on Snap4City](#)
- [HOW TO: Federation of Smart Cities, by federating Smart City API and Knowledge Base](#)
- [HOW TO: extend Snap4City Platform with new modules and tools](#)
- [HOW TO: Manage BMP and BIM: main features of openMAINT, BMP, BIM](#)
- [HOW TO: Support Multiple Cities/Area with a single Snap4City Platform](#)
- [HOW TO: Time Trend Compare Widget](#)
- [HOW TO: manage a calendar Widget](#)
- [Form Widget on Dashboard and IoT Applications](#)
- [HOW TO: use and control the Widget Data Cube, OLAP](#)
- [HOW TO: work on ServiceMap for developers](#)
- [HOW TO: sizing your Snap4City Solution on private or public Clouds](#)

5.5 – Test Cases

- **US1.   Using City Dashboards**
  - [TC1.1.   The main entry of the Snap4City service: www.snap4city.org](#)
  - [TC1.2.   Different Tools and Functionalities Accessible for user category](#)
  - [TC1.3.   Accessing and using Dashboards with any device](#)
  - [TC1.4.   Dashboards Showing Data Real Time and Historical/trends, comparison](#)
  - [TC1.5.   Dashboards showing a range of different High Level Types: KPI, POI, IOT dev, MicroApp, Maps using different kinds of Graphics Widget. Monitor city status with Dashboards](#)
  - [TC1.6.   Dashboards using different kinds of graphics Widgets matching with High Level Types, Monitor city status with Dashboards](#)
  - [TC1.7.   Dashboard Builder getting data from data sources](#)
  - [TC1.8.   Visual production of Dashboard via Wizard](#)
  - [TC1.9.   Search on Wizard for any kind of data managed into the platform, from POI to sensors, KPI, social, etc.](#)
  - [TC1.10.   Dashboard delegation to access, and passage of ownership, and/or cloning](#)
  - [TC1.11.   IoT Discovery, on Dashboard Wizard](#)
  - [TC1.12- Dashboard with MicroApplications](#)
  - [TC1.13- Dashboard Builder External Services and Widgets](#)
  - [TC1.14- Dashboards uniform model for: simple dashboard, those with IoT Broker connection and those with logic in IoT Application](#)
  - [TC1.15- One Dashboard multiple IoT Applications, data and IoT Devices/brokers](#)
  - [TC1.16- One IoT Application, multiple Dashboards](#)
  - [TC1.17. Object tracking, widget tracker, personal tracking/trajectories, moving sensors](#)
  - [TC1.18: Origin Destination Matrix](#)
  - [TC1.19: Creating and Exploit heatmaps for Dashboards and as reference data services](#)
  - [TC1.20: Add a private chat room to your dashboard (public or private)](#)
  - [TC1.21- Cross-widget connection and interaction](#)
  - [TC1.22: Create and configure a Snap4City SVG Custom Widget for real-time interaction](#)
  - [TC1.23 – Dashboard Menu management per Organization](#)
  - [TC1.24 – Integrated Ticketing and Facility Management system](#)
  - [TC1.25 – Embed Snap4City dashboards and widgets in external Websites](#)

- **US4. Creating City Dashboards and related Event Monitoring and Actions**
    - TC4.1. Dashboard and Notificator
    - TC4.2. Managing Notifications on IoT Application: SMS, Telegram, email, Twitter, Dashboard events, in Push on other IoT Applications
    - TC4.3. Alerting about critical events involving people in a specific area, IoT Application, Acting to IoT Application from City Dashboards and vice versa
    - TC4.4. Notificator Engine Monitoring
    - TC4.5. MicroApplication for suggestions
    - TC4.6. Dashboard with city events, ESB, police, traffic, etc.
    - TC4.7: Vehicles and Fleets Monitoring Telemetry via OBD2 and Tracker
    - TC4.8: Widget Single Content on IoT Applications
    - TC4.9: New Support Widgets for Bars and Series, on Dashboards and IoT Applications
    - TC4.10: Bubble charts on maps for Sensor Data
- **US5. Discovering City Services Exploiting Knowledge Base via ServiceMap**
    - TC5.1. Knowledge Base Semantic Search via ServiceMap, geolocation: around a point, along a path, into an area
    - TC5.2. Knowledge Base Semantic Search via Log.DISIT.org
    - TC5.3. Knowledge Base Semantic Search via SPARQL
    - TC5.4. Knowledge Base Semantic Search via full text: ServiceMap
    - TC5.5. Knowledge Base Semantic Search via relationship category, sub cat, value type, value unit: ServiceMap
    - TC5.6. Drill Down on entity relationships (via ServiceMap and LOG). Discover City Entities as Linked Open Data, triples, semantic search.
    - TC5.7. Dashboard for Data Comparison: Multiple Selector, and cross communication among widgets
    - TC5.8. Dashboard for Data Comparison: multiple servicemap
    - TC5.9. Developer Dashboard tool for comparison analysis
    - TC5.10. Open Street Map ingestion process
    - TC5.11. DataMapper, DataMapping, on knowledge and MicroService IoT Applications
    - TC5.12. Anonymous annotation via Smart City API and MicroApplications
    - TC5.13. Personal Annotation Services from web page, MyAnnotation from MicroService, Saving and Accessing to "MyAnnotations" from IoT App
    - TC5.14. Sharing Knowledge base views and queries on ServiceMap and Log.DISIT.org Discover City Entities as Linked Open Data, triples, semantic search.
    - TC5.15. Snap4City Smart City API Collection and overview, real time
    - TC5.16. Exploiting Smart City API for developing Mobile and Web Apps
    - TC5.17. Search on Services via Smart City API: MicroApplication, Exploiting Micro Applications in HTML5 based on Advanced Smart City API
    - TC5.18. Snap4City APIs are documented in Swagger, and tested in Postman
    - TC5.19. Using ServiceMap as a Tool for Developing web and mobile apps and micro applications
- **US6. Developing and using processes for data transformation**
    - TC6.1. Managing DataSets via DataGate: ingest, search, download, upload, annotate, share
    - TC6.2. Search on DataGate for Data Sets
    - TC6.3. Creating ETL processes for automated data ingestion and data transformation
    - TC6.4. Managing ETL processes via Resource Manager, upload, execute, monitor
    - TC6.5. Managing Heterogeneous File Ingestion via ETL processes
    - TC6.6. Producing data-sets in Bundle via ETL
    - TC6.7. Producing data-set in Bundle/Bulk via IoT Application
    - TC6.8. ETL processes for data transformation, and exploiting MicroServices/API/RestCall
    - TC6.9. ETL processes for multiprotocol and format data ingestion, see on GITHUB for library
    - TC6.10. ETL Applications using multiple protocols, and formats for files and to calling services using REST and WS
    - TC6.11. Add a new ETL coping with a new Protocol
    - TC6.12 How to Create GTFS File Ingestion via ETL (Extract Transform Load): the case of Helsinki
- **US7. Data Analytics and related integration aspects**
    - TC7.1. Exploiting data analytics and machine learning in IoT Applications as MicroService
    - TC7.2. R Studio for Analytics, exploiting Tensor Flow
    - TC7.3. Download data from AMMA (Application and MicroService Monitor and Analyser), ResDash (Resource Dashboard) and DevDash (Development Dashboard) tools
    - TC7.4. From R Studio process to MicroService for IoT Application, data analytics, machine learning
    - TC7.5. Developing Data Analytics Processes

14.6 - All the other articles:

- IoT Directory - UI (user Interface), HOW TO: work on IoT Directory User Manual
- Scenario: MOBIMART Interreg: MOBilità Intelligente MARe Terra
- Custom Synoptics and Widgets for Dashboards
- DORAM: how to create alternative scenarious (mobility demand vs transport offer)
- Scenario: City of Roma case, mobility and environmental data
- Scenario: Herit-Data video and aims
- Twitter Vigilance, social media monitoring and early warning
- Snap4City Certificates of Proficiency and/or Participation
- Scenario: Control Room vs Video Wall
- DISIT Lab Traffic Flow Reconstruction V1.0 Tool License Agreement, version 1.2, 2020
- Scenario: Snap4Home, how to exploit Snap4City solution on home automation
- Snap4City info su Andamenti Nazionali e Regionali infezione COVID-19
- Snap4City Harmonized Data Ingestion process
- Hotels in Florence having an agreement with UNIFI
- New Support Widgets for Bars and series
- How to register automatically an uri callback on Orion Context Broker
- Installing Snap4City: Appliances, Virtual Machines/Containers ready to use on your premise for Smart City and IoT Applications
- Snap4City 1st Place award by Select4Cities PCP
- Snap4City Business Model
- Snap4City at Smart City Expo World Congress 2019
- Article on Sensors: https://www.mdpi.com/1424-8220/19/21/4798/htm , MicroServices Suite for Smart City Applications
- Scenario: Energy Monitoring
- Data Analytics: Analyzing Public Transportation Offer wrt Mobility Demand
- Security, Privacy and GDPR Aspects of Snap4City
- How to: Develop Smart Applications, Snap4City development Life Cycle
- Smart City Control Room Dashboards: Big Data Infrastructure, from data to decision support
- DataAnalytics on Antwerp and Helsinki
- What they state about Snap4City tools
- HOW TO: activate the zoom of External Services embedded into Dashboard Widget
- Dashboard info: Mobility and Environment What-IF Analysis, Florence
- Decision Support Engines and Models: SmartDS, ResilienceDS
- IPCAM on Dashboard Widget, Video Cameras
- Solution: using PAX Counters, monitoring museum and events
- Solution: selecting the less pollute routing
- How to Install: IoT Directory
- Data Flow Management and Visual Analytic for Big Data Smart City/IOT
- Dashboards: Since today all of them have Touch Screen Support
- Snap4City on Social Media
- AIRQino: Compact low-cost air quality monitoring station, by CNR IBIE
- Changing the backgroud map, loading Ortomaps, on Dashboard
- Custom Synoptics and Widgets for Dashboards
- Snap4City on the European Open Science Cloud Marketplace, EOSC
- Tutorial on the Bulk Load Approach for IoT Devices on Snap4City
- Training Snap4City: from data to sentient City, Smart City and IOT
- Snap4City Mobile Apps: Helsinki in a Snap, Antwerp in a Snap
- HOW TO: configure/install StartSNAP4CITYVM: starting appliance of Snap4City
- Node-RED IoT Applications for Smart City
- Snap4City Appliance: Virtual Machines ready to use on your premise for Smart City and IoT Applications
- Phase 3: Iteration 3: Trial on Antwerp and Helsinki (links)
- Snap4City Appliance, Virtual Machines
- Snap4City vs Fi-Ware
- Environmental Data Definitions, and Predictive Models
- Snap4City Hackathon 2019: The Winners

- Snap4City vs GDPR
- **Articoli in Italiano:** Articoli e Descrizioni in Italiano
  - DISIT Lab Overview 2021
  - Course: Big Data and Analytics, Master Industry 4.0 Design, PISACourse: Big Data and Analytics, Master Industry 4.0 Design, PISACourse: Big Data and Analytics, Master Industry 4.0 Design, PISA
  - Traffico, meteo e qualità dell'aria: l'importanza dei dati per gestire e vivere le città
  - Corso su IOT, programmazione in Node-RED snap4City in Italiano
  - Scenario: Snap4Home the case of: Alexa, Philips, Sonoff, TP-link, etc. (Italiano)
  - Smart City Lighthouse ,snap4city, SDLM in Advanced Design LM12, Bologna: SLIDE and VIDEO (italiano)
  - Master Industry 4.0 Design: Big Data and Analytics
  - Training Snap4City: dai dati alla Città Senziente, Smart City and IOT
  - La Smart City Control Room di Snap4City (ITA)
  - Data Gathering: acquisizione dati (ITA)
  - Big Data Analytics, analitica (ITA)
  - Snap4City per Smart City e Industria 4.0 (ITA)
  - Sindaco Nardella allo Smart City Expo World Conference 2018
  - Un Hackathon promosso dal DISIT Lab del Dinfo a livello internazionale (ITA)
  - Training Snap4City: dai dati alla Città Senziente, Smart City and IOT
  - Una App per smaltire bene i rifiuti elettronici ed elettrici, WEEE
- Most popular content on Snap4City
- Snap4City Hackathon 2019 finalists for Antwerp and Helsinki
- Data Analytic: HDOP as GPS Position Quality Assessment
- Snap4City: best show cases
- Air Quality Indexes
- Codemotion Articles:
  - SELECT for Cities contest: solve challenges and create value for citizens
  - How Snap4City can bring your vision of a Smart City to reality
  - The Snap4City API: a tool for smart city developers
  - Tutorial: connecting an ESP-12 device to the snap4city network
  - Tutorial: connecting a raspberry Pi to the Snap4city network
  - The integration between data and devices: the Snap4City solution
  - Snap4City Dashboards: Overlook Your City
  - Toward the Sentient City with Snap4City
  - Using Snap4City, Innovation and Technology to Improve City
  - IOT Development: the Snap4City way
  - Data Gathering and Knowledge Management for Smart Cities in Snap4City Platform
- Km4City Urban Graph Representation
- How to Access at R Studio
- How to Access to ETL development environment
- Km4City /Snap4City Smart City API Service Availability
- Snap4Industry: Snap4City for Industry 4.0
- Snap4City per Smart City e Industria 4.0
- Snap4City vs GIS solutions and connections
- Certificate of the Certification Authority
- Karma - in practice: triplification process
- Sindaco Nardella allo Smart City Expo World Conference 2018
- Snap4City Abstract: the most and the advanced aspects
- Snap4City: Arduino & ESP8266 IoT Device NGSI
- Snap4City IoT Edge on VM Ubuntu: IoT Application
- My Personal Statistics
- User Limits and Consumption of Resources
- Snap4City platform: from cloud to data analytic as a Living Lab for Student and Researchers
- Partners and Interoperability
- Registering IoT Edge: example of Raspberry Pi, total security
- Mobile Phone Android as IoT Edge with IoT Applications in Node-RED

- Snap4City Terminology
- Snap4City Living Lab Life Cycle suggested
- Architecture of Snap4City with Internal API, documented in Swagger
- searchable: the data, both structured and unstructured
- The daisy of MicroServices for Snap4City Dashboard and IoT App.
- IoT Applications, MicroService Directory, data bidirectional data flows
- HTML5 MicroApplications Smart City API
- Snap4City IoT Devices Registration
- Creating: IoT Device, Raspberry Pi based totally compliant with Snap4City
- Snap4All IOT Button: based on ESP32
- Users' Roles on Snap4City
- Contributing as city stakeholder: company, citizen/developer, researcher, etc., to the Living Lab
- Real Time and Data Driven Capabilities of API
- Acting on the basis of Data and Data Analytics
- Exploring and Drill-Down on City Data
- Supported file and message formats
- ETL processes for massive Data Ingestion and Transformation
- Snap4City Supported Protocols, adding new protocols
- Testing Snap4City API with Postman
- High Level supported IOT Protocols
- Snap4City Collaborative Work and Co-working
- Snap4City Dictionary of Tools and Modules
- Snap4City Overview and Architecture vs it users
- EventLog API
- References
- Snap4City Logo
- Acronyms
- Doc:Dashboards https://www.snap4city.org/drupal/node/17
- Doc:Notificator https://www.snap4city.org/drupal/node/18
- Doc:IOTApplications https://www.snap4city.org/drupal/node/22
- Doc:IOTDirectoryandDevices https://www.snap4city.org/drupal/node/40
- Doc:ServiceMap&3D https://www.snap4city.org/drupal/node/19
- Doc:MicroApplications https://www.snap4city.org/drupal/node/54
- Doc:ExternalServices https://www.snap4city.org/drupal/node/55
- Doc:DataSetManager https://www.snap4city.org/drupal/node/26
- Doc:ResourceManager https://www.snap4city.org/drupal/node/27
- Doc:RStudioDevelopment https://www.snap4city.org/drupal/node/25
- Doc:ETLDevelopment https://www.snap4city.org/drupal/node/24
- Doc:KnowledgeBaseGraph&Query https://www.snap4city.org/drupal/node/21
- Doc:SmartCityAPI,Swagger https://www.snap4city.org/drupal/node/20
- Doc:TrafficAnalyzer,AMMA https://www.snap4city.org/drupal/node/43
- Doc:DataAnalyzer,DevDash https://www.snap4city.org/drupal/node/23
- Doc:ResourceAnalyzer,ResDash https://www.snap4city.org/drupal/node/42
- Doc:SchedulerDISCES https://www.snap4city.org/drupal/node/28