Snap4City Platform

Powered by SNAP4Tech

Technical Overview

From Snap4City:

- https://www.snap4city.org
- https://www.snap4solutions.org
- https://www.snap4industry.org
- https://twitter.com/snap4city
- https://www.facebook.com/snap4city
- https://www.youtube.com/channel/UC3tAO09EbNba8f2-u4vandg

Author and coordinator: Paolo Nesi, Paolo.nesi@unifi.it

DISIT Lab, https://www.disit.org
DINFO dept of University of Florence
- Phone: +39-335-5668674

Access Level: Public.

Date: 14-09-2025

Version: 8.5

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

Table of Contents

1 - Executive Summary

DISIT Lab provides a Snap4City platform, which covers all the necessities in an urban setting and industry. We have been working in this domain since several years providing a range of modules and solutions with a multitude of partners for creating an integrated platform described in this document freely accessible on https://www.snap4city.org.

Snap4City/Industry platform has a long list of references and successful installations which can are listed on:

https://www.snap4city.org/661

In all cases and challenges, we have deployed solutions covering all the necessities requested, and considered potential eventualities in the design. We work on a 100% open-source basis, and according to a continuous innovation and development (CI/CD) approach, which means that every new development is performed on the platform and made accessible for the whole community and in open source. This has allowed us to provide for many years a large number 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, energy, security, etc., and in cases improving and integrating all cases in the Snap4City platform for all partners.

Snap4City platform is a scalable solution from single city to regional coverage with multiple cities and areas. The platform is an advanced AI enabled 2D/3D Digital Twin, flexible, real time dynamic and highly interoperable with any device and protocol to increase sustainability and asset control according to city's operational objectives, and capable to support tactic and strategic plans (what-if, optimized and simulated) fully integrated with the territory and stakeholders. **Snap4City exploits** any data/device (**GIS, IoT, BIM, ITS, Sat., OSM, ODM, NGSI, MQTT, CKAN, OGC, OPC-UA, ModBUS, etc.**) to get info and act, and provide hints and controls with respect to the AI Digital Twins of the services of the city/area, providing to decision makers easy to grasp explanations of suggestions (proposed solutions) via eXplainable AI, XAI, and AI-Ethics, and grounded on KPI. In operation, **Snap4City** produces real time notifications (early warning) on multiple channels, and may also perform actions according to profiling and any KPI: decongestion, increment of service quality, reduction of emissions, reduction of costs, etc. (SUMI, SUMP, DUT, SDG, 15MinCityIndex, etc.)

Snap4City provides a set of ready to use tools on multiple domains of the city/area to perform real time operation and/or planning, on: **mobility and transport, energy/light, waste, environmental, government, tourism, security, safety, civil engineering.** The most widely adopted smart tools and dashboards grounded on KPI cover **Control Rooms, Smart Parking, Fines Management, Asset Management, Smart Light, Smart Waste, Traffic and Mobility services optimisation, Tourism management and Optimisation.** As it is on more than 45 cities/areas in countries as Italy, Cyprus, Spain, France, Bosnia-Herzegovina, Greece, Croatia, Sweden, Belgium, Australia, Brazil, China, Romania, Malta, etc., as well-known cities of Florence, Valencia, Merano, Malta, Limassol, Varna, Cuneo, etc. **Snap4City is the reference platform** of Sustainable Mobility Center in Italy (CN MOST), CN HPC big data and quantum computing in Italy, Smart Cyprus, Rhodes's control room, in progress with ISPRA JRC SOC, etc. A number of **Living Labs** adopted Snap4City solution: Florence, Rhodes, CN MOST, etc.

Any Snap4City platform is provided with its main integrated development platform based on Node-RED, JavaScript. A number of solutions for computing and rendering on integrated graphics and for

generating *predictions, heatmaps, ODM, scenarios, synoptics, maps, 2D/3D, animations, etc.,* are provided in open source in basic installations or a simple addon. **Advanced Snap4City AI toolbox includes a set of ready to use modules** based on ML, AI, Generative AI, LLM, RAG LLM, Deep NN, etc., are: *statistics, advanced predictions, early warning, advanced heatmaps, classification, suggestions, simulations, sentiment analysis, optimisation, routing, zoning, reconstruction, origin destination matrices, trajectories, LLM via Snap4Advisor*, etc., by which any smart and business intelligence tool can be created.

Developers of Snap4City may request access to SnapAdvisor which is a Generative AI expert of Snap4City platform which provides answer about how to develop new solutions based on Snap4City tools and help in using standard Snap4City applications and framework. SnapAdvisor is based on a Generative AI LLM. A custom SnapAdvisor can be provided for large Snap4City platforms to assist operators and decision makers on data dependent specific context.

Snap4City is a 100% open-source platform which provides fast and easy start up (on public and private clouds) thanks to its interoperability and automated installation tools. Snap4City is semantically interoperable with Km4City ontology (https://www.km4city.org) grounded on SAREF/S4CITY, WGS84, OTN, DCterms, GTFS, OWL-Time, FOAF, BOT, SSN, etc., is an official FIWARE Platform (https://www.snap4city.org/467 , https://www.fiware.org/), compliant with FIWARE Smart Data Models, IoT Data Models, Data Spaces, and a large range of High-Level Types, official EOSC (European Open Science Cloud) Platform, official Node-RED Library, official E015 API, etc. Snap4City is provided "as a Service" or installed on premise, as well as on pub cloud as AWS, MS-Azure, ARUBA, etc., no licence fee is needed. The list of public **Snap4City installations** is accessible on web portal.

Cybersecurity and Privacy: **Snap4City is GDPR** (General Data Protection Regulation of Europe Union) compliant, and passed **PEN-Test**. With Snap4City you can create your certified entities/data, and sequences of events using the provided Blockchain support. End-2-end secure connections are established from devices to dashboards, web user interfaces, including data processing, storage and data analytics. https://www.snap4city.org/549 . Snap4City is compliant with OpenID Connect, SSO, OIDC, SAML/IAM, European Identity Card, SPID, EU Login, Active Directory, etc. The platform is provided with its **Snap4City Sentinel** application to monitor 24/7 all the functionalities and report any kind of problem over multiple channels. DISIT Lab is well known at the international level on Smart Cities / Industry domains for many implementations such as Methodology for Smart Cities see:

https://www.snap4city.org/download/video/cov/

https://www.snap4city.org/4

Snap4City/Industry is a sustainable data/service ecosystem where stakeholders can exploit resources to set up valuable services, free from vendor lock-in problems. When stakeholders are ready, the support to set up the Living Lab or co-working environment is provided, to create a collaborative context among several cities and international experts, where cities' users and stakeholders are involved to profitably exploit, produce and share data and services.

1.1 Change of Terminology, since 2023 version of the platform

After many years, we have been constrained to change some notations: due to the evolution of Snap4City tools and concepts in terms of their capabilities, features, some of the tools and concepts would be better represented with different names. In particular, we are evolving the names as reported in the following table. The change of the name will be progressively propagated in the platform tools, this document, document of technical overview, and on slides, step by step. In the meanwhile, in most cases the text would report both names, and thus please keep in mind that:

Former name	new name, since 2023	What
IoT Device Model	Entity Model	A data model
IoT Device	Entity Instance	A data instance ready to get message for time series
IoT Device Variable, metric	Entity Variable	A variable of an Entity Instance or of an Entity Model
IoT Device Message, device message	Entity Message	A data message
IoT Directory	Entity Directory	The tool for managing models, entities, data models, etc.
IoT Applications, IoT App	Processing Logic Proc.Logic	= Node-RED + Snap4City Libraries The tool for visual programming, node-red JavaScript, data flow, ingestion logic, data transformation, data loading, interoperability, business logic.
Dashboards	Views and Dashboards	The Snap4City Dashboards are effectively Views of some Web Application, with all the interaction and connection the developer would create among them.
LOG	LOGraph	not all Snap4City platform are provided with the LOGraph, it is optional and can be installed in a second phase

1.2 Document Organization

This document is organized as follows:

- **Section 2** describes the references and solutions, covered scenarios, direct contracts in places, and international projects, prizes, and recognitions awarded by the platform;
  - Including installations on-premise and the size of the installation on cloud and on-premise.
  - Providing a list of accessible scenarios and use cases.
  - Providing a list of references and awards received.
- **Section 3** describes the main concepts of the Snap4City Platform.
  - In addition, it provides an overview of architecture and innovation methodology while dealing with many specific questions/issues:
    - 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 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 to manage smart city infrastructures taking into account data model, processes, users, and applications in a single tool;
    - What smart city APIs can do for you and your city;
    - How city users and visitors can get engaged in providing data and using city services;
    - How to exploit data, for mobile apps and to create services;
    - How to develop applications for the integration of your processes;
    - Which kind of interoperability you may have with Snap4City and the rest of your solutions;
    - How to federate your smart city with others around;
    - How to develop Data Analytics tools;
    - What the situation is like in other cities;
    - If it would be easy and powerful for all developers;
    - Which kind of dashboards and tools you may have, or may build in a short time;
    - What the development environment will be like; How easy it would be to work with it;
    - How flexible and scalable the platform is;
    - How far the user interface could be customized:
    - The level of security;
    - The platform can support privacy issues;
    - How to manage the maintenance of the city assets via the platform in an easy way;
    - How it can provide tools for deep management of the platform;
    - The kind of tools provided to both developers and integrators, to control the smart city platform once deployed;

  - The kind of automatization degree available on the smart city platform;
  - etc.
- **Section 4:** List of Terms and Acronyms that may be of use in the Snap4City terminology
- **Section 5** proposes a list of additional references for further research on these topics. 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

Snap4City/Industry improves city services, security and safety by offering sustainable solutions for smart cities and Living Labs, thus attracting industries and stakeholders. It can be used for implementing large and vertical solutions for smart parking, bike sharing, smart light, smart waste, people flow, industry control, etc. Snap4City is capable of monitoring the city evolution in real time: reading sensors, aggregating data of any kind; computing and controlling key performance indicators (KPIs); 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 and business intelligence tool; performing what-if analysis and simulations.

Snap4City/Industry supports the City/Industry in the process of continuous innovation on services, infrastructures, with control and supervision, tools for decision makers, as decision support systems, risk assessment, what-if analysis, business intelligence tools, based on scenarios, predictions, anomaly detection, early warning, also by setting up strategies for increasing city resilience with respect to unexpected events or unknowns. Thanks to Knowledge Base support (semantic expert system), Snap4City/Industry 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 or 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 agreed to contribute in the full respect of the open-source model. Your artefacts and applications developed using Snap4City can still be protected with your IPR, and can be distributed in non-open-source licensing model as you like.

Snap4City is an official Platform and solution of FIWARE, a platform of EOSC, a platform of Node-RED, a platform of EO15, a platform CN MOST (national center of sustainable mobility in Italy), a platform of CN HPC (national center of HPC, big data and quantum computing of Italy), winner of Select4Cities PCP Award, winner of the ENEL-X Challenge on Open Data. Snap4City supports many kinds of protocols and brokers; it is certified to offer FIWARE Training Services, FIWARE Consultancy Services, and provide certified FIWARE Experts.

Snap4City/Industry can be used to set up Living Labs and smart city solutions satisfying a large range of requirements by city officers, citizens, and tourists or developers, decision makers, operators, companies and researchers. It can be easily integrated through legacy and in place solutions to integrate and provide data, factors and causes, predictions, anomalies, optimizing resources and understanding social contexts and meanings of facts. It can be installed on any public cloud or on your premise, as well as on hybrid solutions, including on Edge, on premise.

Snap4City is based on **Km4City Ontology** for the **Knowledge Base** and on **Smart City API**, a suite of open-source technologies for developing smart cities, big data aggregators, and analytics adopted in several actions such as: ISPRA JRC SOC, CN MOST, CN HPC, Tourism Interreg, SADI-MIAC, CAI4DSA, Energia, AMMIRARE, Museum, ARTER energy monitoring, Ampere, Enterprise, Italmatic, 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, Pretto, Enel-X, and others scenarios and project see the updated list on: https://www.snap4city.org/4 and on news: https://www.snap4city.org/135 and on slides on which also the list of major actions and contracts is accessible.

Snap4City Living Labs is a co-working environment for IoT/IoE (internet of things/everything), smart data models, Digital Twin, in smart cities and industry 4.0; it is open to all standards and custom models to cover a large range of scenarios and a large range of IoT Devices, IoT Brokers, IoT Edge, Gateways, mobiles, OBU (on board unit, vehicular kit for bikes, cars and busses), social media and satellite data. 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 the privacy of a variety of types of users according to GDPR, provides tools and community for collaborative working activities; mixed data driven, stream and batch processing; it is fully modular and based on microservices, using 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, energy, waste, parking, health & welfare, social, Industry 4.0, etc. The proposed solution is flexible enough to support extensions at different levels of granularity: data, analytics, AI, tools and applications.

Dr. Alessandro Monti, General Manager of Foundation for Research and Innovation, FRI. "The Snap4City platform has represented a fundamental boost up for both Herit Data and Tourismo project, thanks to its extraordinary potentialities and flexibility. Snap4City allowed for the use of advanced sensors such as sniffers and thermal cameras, whose data were acquired, elaborated and interpreted, also by means of AI algorithms, one of the many arrows in Sanp4City quiver. The platform merge advanced solutions with a very user friendly dashboarding making it a tool for specialist, but also citizens, and the latter are one of the reasons why many public administrations found Snap4City and ideal instruments to help them in decision and policy making. Keep it up Snap4City team, starboard tiller toward new frontiers!"

Florence Municipality: "tourist-flow system analaysis via Snap4City dashboards and charts will provide us a strong view to understanding of flows and visitor behavior"

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."

Herit-Data: "Snap4City with its tools enabled flexible smart-city applications for managing tourism impacts, had a strong impeact on their management in several cities. Thanks to the solution and strong work of partner, the Herit-Data solution has obtained the flaghips from the EC, as Interreg Good Practice."

Snap4City is operative with services and data of several cities and organizations such as:

https://www.snap4city.org/download/video/cov/

https://www.snap4city.org/4

The map shows several installations and each of them is Multitenancy. The largest installation is www.snap4city.org on which you can access data and dashboards of a relevant number of the above organizations. Most of the Organizations keep data and dashboards as private, this means that even though you can free register on snap4city.org and join any organization among them, the access to their resources and dashboards is decided by the organization manager, and thus you may not see all this data, dashboard and all the Snap4City functionalities in place in any organization. For this reason, if your aim is to understand how Snap4City works, we suggest performing **FREE REGISTRATION** in Snap4City by selecting DISIT Organization and you may use any other Organization accounts 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**, **Politecnico di Bari, University of Naples**, **University of Modena,** etc.,

We perform a continuous training and updates of training material from https://www.snap4city.org/944 .

Refer to Section 4 for the list of acronyms, definitions, and to Section 5 for references.

We tried to include weblinks in the text as much as possible so that you can jump directly to the web page to find out more useful information, definitions, etc.

2.1 – Installations on premise and volumes of Snap4City

***Snap4City is a platform* adopted** in a number of operative solutions and projects (including EC Projects and national projects, funded by public institutions or foundations). For this reason, there are a number of "Operative in Production" installations and a number of ongoing "Pilots" that use Snap4City Platform as better explained in the following:

- **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.
  - Most of them are federated with each other as described further on in this document.
- **Started in Pilot** is used for solutions that are performing data ingestion and which going to become visible and in production soon, such as in the cases of Dubrovnik, Valencia, Malta, Bisevo, Valencia, Rhodes, Varna, 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).*

See for the list: https://www.snap4city.org/661

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**.

On the web page: https://www.snap4city.org/661 Snap4City installations reported. There are other installations unregistered in this list as Snap4City is fully Open Source and everybody can download and install without informing us that the platform is being used. ***Not all Snap4City platforms installations are reported in the following table, as some of them chose not reported and other have not notified us of the installation performed.*** A number of organizations are using the Smart City as a service provided and operated by DISIT Lab directly.

You can access to a wider list of scenarios from:

- DOMAIN: Control and Plan Horizontal Artificial Intelligence Platform Digital Twin for All Domains:
  - **https://www.snap4city.org/1039**
- DOMAIN: Mobility and Transport Operation and Plan Digital Twin**:**
  - **https://www.snap4city.org/1040**
- DOMAIN: Smart Energy and Smart Buildings Operation and Plan Digital Twins:
  - **https://www.snap4city.org/1041**
- DOMAIN: Environment and Waste Management Digital Twin:
  - **https://www.snap4city.org/1042**
- DOMAIN: City Users' Services, Tourism Management and Safety Digital Twin:
  - **https://www.snap4city.org/1043**
- **Scenarios: https://www.snap4city.org/4**
- **training course – Part 1: https://www.snap4city.org/download/video/course/p1/**
- **training in PDF from page https://www.snap4city.org/944**

The largest installation is operated by DISIT Lab as Snap4city.org. And **is at present the largest installation that overcome the needs of most of the cities and industry plants since it (data updated in 2025)**

- ingests about 2,3 million attributes every 24 hours;
- generates about 5 million attributes every 24 hours regarding: MyKPI; Heatmaps for Pisa, Firenze, Livorno; traffic flows; GTFS; user data;
- has more than 9000 users in different organizations;
- manages more than 26 organizations/tenants;
- provides more than 1650 dashboards, of which more than 380 are public dashboards
    - you can access most of them from: https://www.snap4city.org/dashboardSmartCity/management/dashboards.php
    - more than 120 different kinds of graphic widgets;
    - more than 20 macro-classes for data, and more than 520 subclasses;
- manages elastically more than 620 containers with IoT App, Data Analytics (Rstudio, Python), Web Scraping, IoT App on Edge.

Updated in 2021, for example, 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 – Examples of Scenarios of Snap4City

Snap4City/Industry covers a wide range of possible scenarios with a unified platform, from smart city to IoT and Industry 4.0. Cities often have industry-level plants in the structure they manage, for instance: water depuration, large storage, fleet management, etc. A list of the main examples, at European and Italian levels, is reported below, but the list must be considered only representative of what can be performed, as the platform can define and work with any kind of model, starting from the FIWARE Smart Data model to Snap4City IoT Device Models, or others imported from brokers, etc.

Smart City cases: https://www.snap4city.org/4

All scenarios are fully described starting from the main domain pages, get the PDF which are often updated:

- DOMAIN: Control and Plan Horizontal Artificial Intelligence Platform Digital Twin for All Domains:
  - **https://www.snap4city.org/1039**
- DOMAIN: Mobility and Transport Operation and Plan Digital Twin**:**
  - **https://www.snap4city.org/1040**
- DOMAIN: Smart Energy and Smart Buildings Operation and Plan Digital Twins:
  - **https://www.snap4city.org/1041**
- DOMAIN: Environment and Waste Management Digital Twin:
  - **https://www.snap4city.org/1042**
- DOMAIN: City Users' Services, Tourism Management and Safety Digital Twin:
  - **https://www.snap4city.org/1043**

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 in which Snap4City/Industry is used as a reference platform:

https://www.snap4city.org/4

2.4 – Recognitions and Awards to Snap4City, DISIT Lab

The Snap4City Platform:

- Digital Twin paper, Awarded by DMS
- Digital Twin paper Awarded by ICCSA
- DISIT lab, UNIFI is a partner of the Smart City Strategy development for the Cyprus Ministry
- **official solution** of **FIWARE**: https://www.snap4city.org/467
- **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
- **Snap4City is a Powered by FIWARE Solution**:
  - https://marketplace.fiware.org/pages/solutions/b8905e91973b420189cce972
- **Snap4CIty if an official FIWARE Platform**:
  - https://marketplace.fiware.org/pages/solutions/d68534ec827500f1bde8720f
- 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: ù
  - 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 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 for several years.
- **GDPR compliant** and enforces privacy and security for data, **Dashboards**, **IoT Devices, IoT App, personal data, data analytics, processes**, etc., that users can keep private. Users also may delegate access to them or pass the control to other users. https://www.snap4city.org/670
- passed **PEN Test from Setek and Vulnerability Test from Thales in 2019**, they are the leading expert companies in the Cybersecurity 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 – The Snap4City Platform

Snap4City is a digital twin platform designed for decision makers to perform activities of operation and plan in the cities and control rooms. Operation means at least to monitor, control, plan, predicting and react in real time to current operational conditions and related events; and for planning. For this purpose, data are collected, processed and early warning computed by using a range of data analytics and AI processes, and/or simulated. Planning means to assess the city condition in terms of contextual and historical data, the desiderata/objectives and goals/key-performance-indicators o provide support in planning, generating suggestions, solutions to the problems, simulating conditions and make comparison for decisions, etc. The operation is always an activity to be performed in real time, quasi real time, while the plan may take time. Recently, what-if analysis and optimization tools, typically applied on plan are also used on operation for fast reaction to critical conditions and events, and the time to plan has been strongly shortened. Snap4City provides an integrated solution for data gathering, indexing, computing, and information distribution, thus realizing a continuously updated digital twin of the urban environment at global and local scales for monitoring operation and planning. It addresses 3D building models, road networks, Internet of Things entities, points of interest, paths, as well as results from analytical processes for traffic density reconstruction, pollutant dispersion, predictions, and what-if analysis for assessing impact of changes, all integrated into a freely accessible interactive 3D web interface, enabling stakeholder and citizen participation to city decision processes. *In Snap4City, you can have the Digital Twin in operation and potentially an infinite number of Digital Twins in simulation, optimisation, what-if analysis. They can be derived from the one in operation, as well as generated by AI, generated by changing the copy from the one in operation, in the current time or in future conditions.*

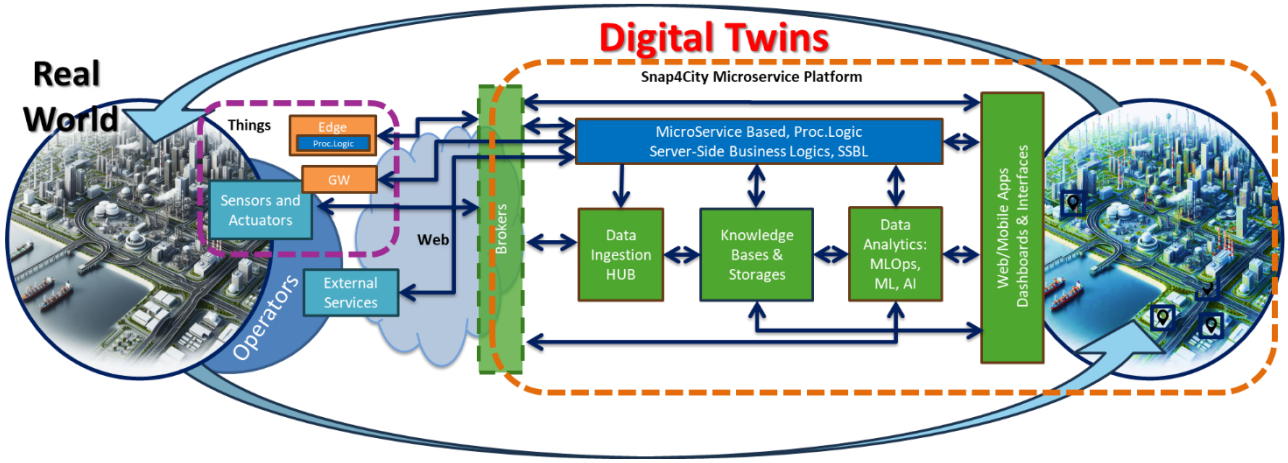


Figure 1a: Main Digital Twin Architecture of Snap4City

To fully understand the Snap4City Architecture and tools we suggest to read the document reporting the **TECHNICAL OVERVIEW:** https://www.snap4city.org/download/video/Snap4City-PlatformOverview.pdf which also includes a glossary of terms (take in mind the change of terminology as reported in **Section 1**). On the other hand, a summary is reported in this section for shortening your learning and training.

Snap4City platform is 100% open source, based on microservices, scalable, and modular with a set of tools accessible according to user profiles. **Smart Applications can be easily developed by producing only: data transformation, data analytics, AI, simulation, and dashboards/views with almost no coding activities.**

Snap4City can ingest and process/produce data/entities of any domain according to any kind of data sources: GIS (ArcGIS, QGIS, etc.), city utilities (water, gas, mobility, energy, light, waste, environment, industry, parking, people flow, beach, etc.), legacy systems, personal data, mobile data, database data, IoT Network and Broker, brokers of any kind, KPI/MyKPI, Industry 4.0 protocols and network, social media, telecom data, trajectories, heatmaps, flow, vector fields, heatmaps, origin destination matrices, satellite data, fleet data ODB2, scenarios,

color maps, floors, BIM, GTFS, etc. All sources are bidirectional channels as Snap4City can ingest and produce data/entities with protocols suitable for any channel. See for interoperability https://www.snap4city.org/65

Snap4City supports any kind of data flow, Entity Networks, communication protocols, and entity/data formats, as well as any legacy and vendor solutions. Entity/Data Ingested with any data/entity models 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, NoSQL. This approach eliminated the problems of data silos and pillars. Any legacy solution, as well as new applications and data/entity is ingested and joined in a unified model automatically, establishing semantic relationships and respecting the original semantic, data sovereignty, data privacy, thus avoiding the flatness of data lake that results in poor performance in data retrieval, data/information access, the occur in Data Lakes when data need to be rendered or acquired by an AI process.

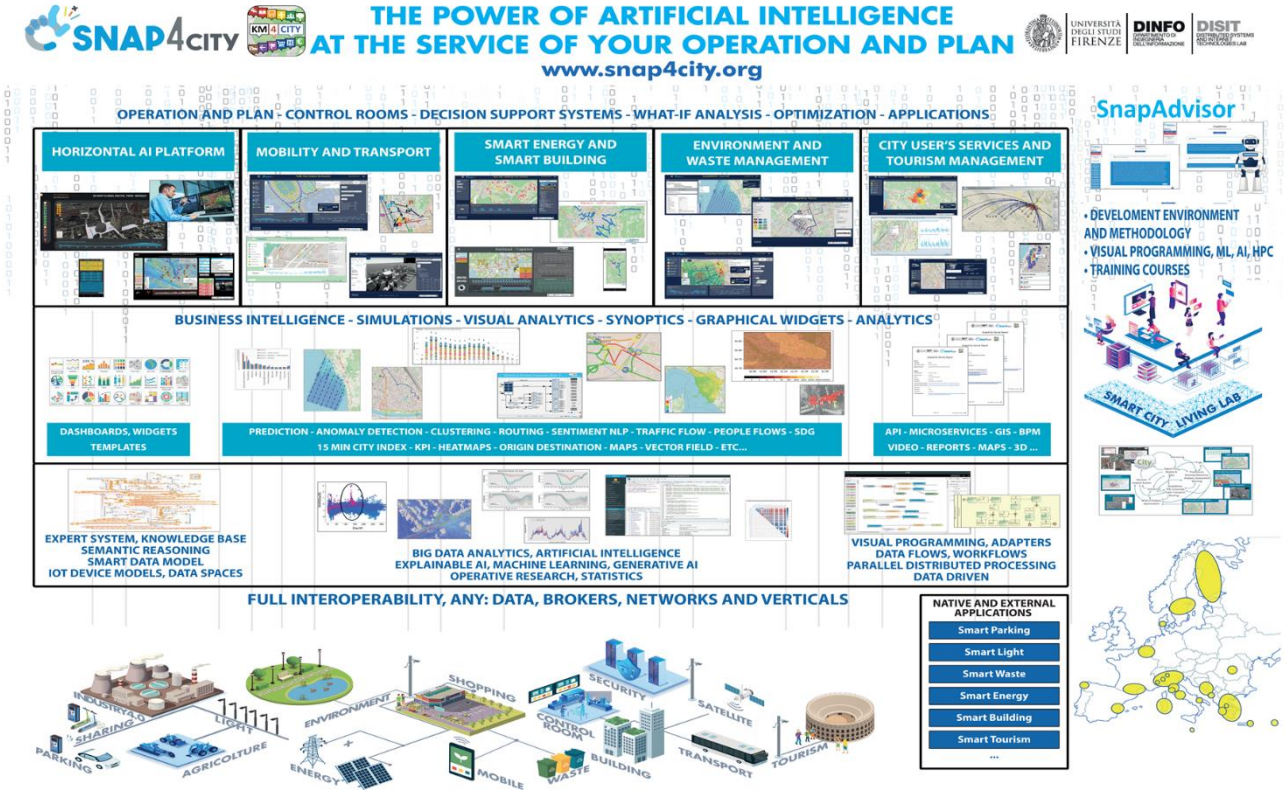


Figure 2: Global Architecture of Snap4City solutions and framework (2025)

Ingested data, just provided in the Brokers are immediately indexed and stored into the long term storage, becoming searchable and usable for snap4City tools such as the Entity / Data Inspector, Big Data Analytics (ML, AI, simulation, statistics, optimization, etc.), Dashboards builder, Smart City APIs and thus for the Mobile Apps, and via MicroServices for Processing Logic / IoT Applications (IoT Apps) which are implemented as Node-RED, used for ingestion, production, integrations, transformation, business logic, server side business logic and intelligence, etc.
https://www.snap4city.org/download/video/course/p5/
Data ingestion can be also performed with high performance processes written in Python (provided by Snap4City as configurable scripts), for example for fast loading of huge amount of historical data from other data bases and/or formats.

Data/Entities can be consumed by data-analytics, Processing Logic / IoT App, mobile App, and Dashboards and Synoptics, AI processes, third party applications via API and in real time streaming, the so-called data driven processes, and end-to-end secure (from the devices to dashboards, back and forwards directions).

Data Analytics (ML, AI, etc.) can be developed in Python and/or Rstudio (using the API, can be developed also

in other languages as GO, C++, CS, etc.). In the case of Python and Rstudio the platform provides tools and guides to transform them automatically in containers to be used into Processing Logic flow (IoT App) in Node-RED. Moreover, if the AI/ML processes are developed according to MLOps of Snap4City (see its manual), they can be deployed on clusters of CPU/GPU / HPC, managed via API, to create AI as a Service, accounting and market place. https://www.snap4city.org/download/video/course/p4/

Data Rendering via Dashboards, Views, Synoptics, tables, any kind of graphics widgets can be easily used just selecting the data and choosing the preferred interactive graphical representation of data/entities among the several possible: time trends, maps, bars, multiseries, chords, OriginDestination (OD) matrices (ODM), tables, heatmaps, trajectories, traffic flow, scenarios, paths, shapes, buttons, dimer, sliders, hierarchies, spidernet, donut, comparing trends, staked diagrams, barseries, custom animated elements, scenario, policies, Time trends, payment models, etc. Moreover, the Dashboards and Views can be connected each other to create Smart Applications, Visual analytics tools, business intelligence tools, also exploiting the Client-Side Business Logic, CSBL, and eventually the Server-Side Business Logic, SSBL, in Node-RED, https://www.snap4city.org/download/video/course/p2/

Developers and qualified operators on Snap4City platforms can access the platform tools via web browser (without any installations on the local computers) to develop solutions and applications in the Snap4City collaborative environment exploiting Data Analytics, AI, ML, Simulations, Dashboards, and Processing Logic / IoT Apps (for data ingestion, adaptation, transformation; business logic, data analytic management, etc.). Data Analytics developers can also develop their application on their local computer if they prefer. When completed the data processes can be loaded to be run in CPU/GPU clusters and/or HPC if available on the Snap4City platform you use.

Mobile App developers need a local development Environment depending on the target mobile devices, an example of mobile app is accessible on Github, the example is based on Cordova. A number of applications have been developed with newer technologies but are not open- source. There are many Web and/or mobiles apps which are currently using the Snap4City model and data, mainly proprietary of their developers. Final users can access to Snap4City services via Web and/or mobile devices, dashboards, Views, synoptics, interfaces for digital signages, tables, panels, etc.

A Snap4City Living Lab allows developers stakeholders to collaborate for the production of smart solutions and to the innovation of the development of their ecosystems. 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, AI, ML, Dashboards, Smart Applications, visual analytics, business intelligence tools, simulators, Processing Logic / IoT App, Synoptics, Custom Widgets, and Web and Mobile Apps [**BIgDataService2018**], see https://www.snap4city.org/426 . **According to Figure 3, the developers can exploit macro-functionalities of Snap4City such as: digital twin support, routing of several different kinds, predictive algorithms, measuring tools for GIS on maps and floors, comparative tools for orthomaps, notification manager, KPI support (SUMI, 15MinCityIndex), scenarios, What-if, Simulators, simulation manager, etc.**

SnapAdvisor is a multilingual, chat-based virtual AI assistant that you can tailor to your own domain by feeding it documents (PDFs, slide decks, code, web pages). Rather than training a model on your data, it uses an **advanced retrieval-augmented generation (RAG)** pipeline so it can answer from the latest content you provide, cite the exact passages it used, and avoid drifting beyond your approved sources. Real-world setups include internal **advisors for Snap4City documentation for Developers**, a hospital legal advisor, industrial operator support, and survey analysis (e.g., TOURISMO). Each is private to its audience and fuelled by that group's own documents. To try it, you can request access and create your collections to make the assistant "expert" in your business. You group materials into **collections** (skills) to keep topics separated—useful when different teams (e.g., legal, operations, marketing) need their own curated knowledge. Organizations can enforce governance

and access controls, personalize answers with tenant/user context, and benefit from lower hallucination rates because responses are grounded in retrieved text with references

The Snap4City Platform

- can be installed on-premise as well as on any public or private cloud. https://www.snap4city.org/738

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 several companies and partners: https://www.snap4city.org/944
- provides continuous training courses with certification. https://www.snap4city.org/622

3.1 – Snap4City Architecture Overview

Smart Applications can be easily developed exploiting the cloud infrastructure by producing only: Processing Logic / IoT App, Data Analytics (ML, AI, simulations) and Dashboards/Views with almost no coding activities (a part for those related to AI, ML). Green parts of Figure 3 are those usually needed to be developed to produce application-level solutions, such as those provided by Snap4City.org or by third party. The rest of the platform modules are provided as microservices for applications. The Snap4City solution and infrastructure can be installed/replicated on private and public clouds, from us or from your technicians. Third party applications can dialog with the platform and each other and can be integrated with the solutions via Smart API, or via Brokers or via Processing Logic / IoT App any protocols.

Third party applications can dialog with the solutions via:

- **Smart City API, Swagger: https://www.km4city.org/swagger/external/**
- **Broker/IoT Brokers API, for example for NGSI context Broker, Entity/IoT Directory: https://www.km4city.org/swagger/external/?urls.primaryName=Orion%20Broker%20K1-K2%20Authentication%20API**
- **Processing Logic/IoT App any protocols: https://www.snap4city.org/65**
  - **And they can also expose some specific API, custom made, not safe and robust enough to be used in production, only for prototypes**
- **Client Side Business Logic, CSBL, directly on the Dashboards. https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf**
- **Exposing API which can be exploited by Processing Logic/IoT App, as well as from CSBL in dashboards.**
- **Exposing API from MLOps clusters to make them accessible to Dashboards, Views, third party applications, smart applications, mobile apps., etc.; also controlled as AI as a Service approach.**
- **Authentication and Authorization APIs.**

Edge and Fog processing tools can be implemented by using Processing Logic / IoT App or other means (custom processes), and may have direct connection with Snap4City Microservices on cloud or locally installed features as local devices, local database, local dashboard, local html pages, etc. API can be those mentioned Smart City API, Broker API. For example, an Edge processing tools may be implemented with Raspberry Pi, windows, Arm, Linux, etc., in which Node-RED is native or can be installed and Snap4City library can be loaded/installed as well. Processing on Edge can exchange data with Snap4City in mutually authenticated manner, TLS, HTTPS, to create secure end-2-end solutions.

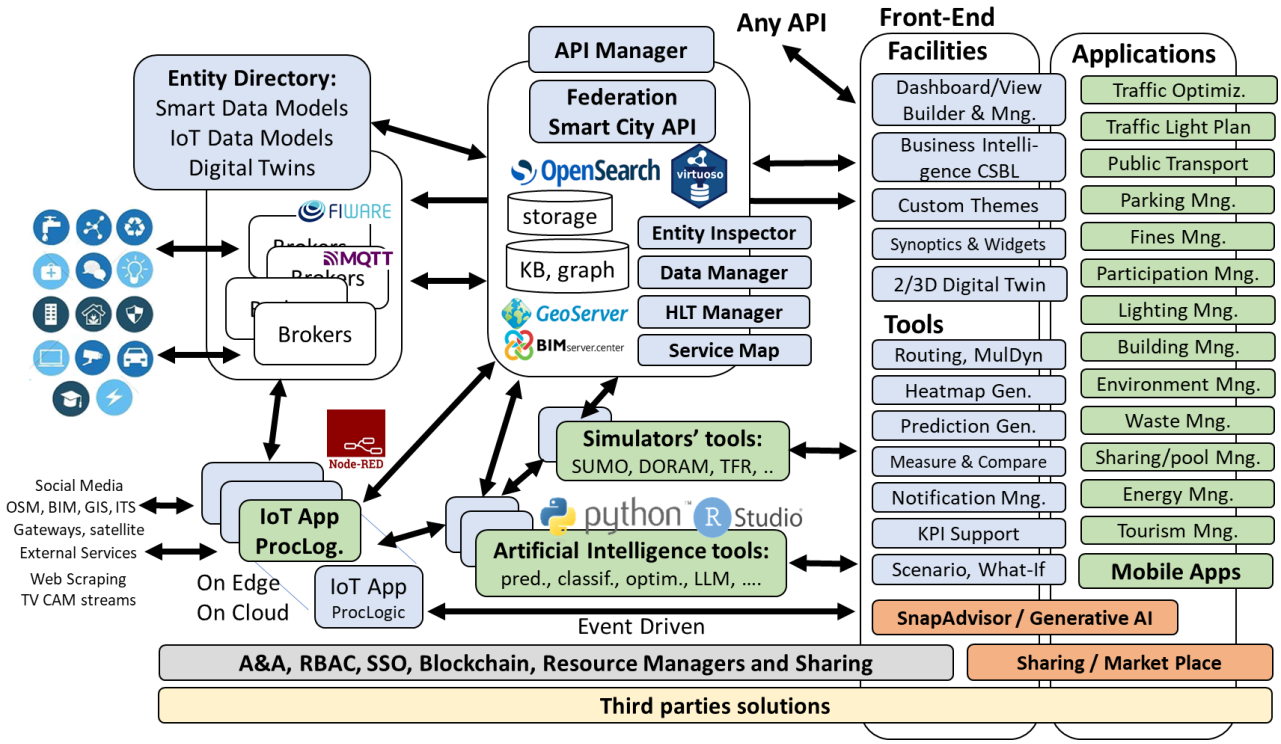


Figure 3: Architectural Overview of Snap4City framework

In particular, for the ***SMart Applications' DEvelopment Life Cycle, SMADELC***, presented in this document we refer to **Figure 3** from the left side:

- ***data ingestion/transformation and interoperability.*** Data may enter/exit in push via brokers of different kinds, and/or via dedicated processes in pull/push (enabling the data/event driven approaches). Data/Entities may be listed in some Entity Directory (IoT Models, Entity Model, Entity Instances, Smart Data Model of FIWARE, data spaces).
    - **Data/Entity may enter/exit from in push via Brokers of different kinds**. All the brokers are managed by the Entity Directory / IoT Directory which registers them and registers any kind of Data/Entity Model including the Smart Data Model of FIWARE, data space models, etc. The brokers can be internally managed (internal Brokers) or external which are managed by third parties. The integration of external brokers with their data/entity model and devices is automated, fast and easy. Please remind that brokers can be also used to act on entities/devices on the field. A Snap4City platform may have multiple brokers: Orion Broker as well as MQTT (both of them can be authenticated according to Snap4City Authentication and Authorization model). Other kinds of Brokers can be added as well.
    - **Data arriving in the platform are stored automatically on the Storage** (Open Search) and / or can be sent directly to the user interface or to data analytics for stream processing. The same flow works as data driven from entering data/entities to dashboards and from dashboards to data streams for acting and sending commands, alarms, etc. Data collected and stored can be browsed by using the **Entity Inspector** / **Data Inspector** and **Data Managers (HLT)** for all the High-Level Types managed in the platform. From Entities, IoT, POI, maps, heatmaps, traffic flow, etc. For some HLT specialized storages can be added to reach higher performances, fully transparent for the users and developers.
- ***Processing Logics*** / IoT Apps could be in some sense used to implement either ETL/ELT scripts. In Snap4City, **Processing Logic** / **IoT App** are Node-RED plus Snap4City microservices or Python processes (to push data in Ni-Fi which is the gate to the storage, OpenSearch). Visual programming languages such as Node-RED, Ni-Fi are typically preferred for rapid implementation. These processes can be also allocated on cloud, fog, edge and on premise/field.

- **Data/Entities may enter/exit in push or pull via Processing Logic / IoT App** as described above to manage data flow from any kind of protocols and formats. Please remind that brokers can be also used to act on devices. Pull/Push connections can be used to work with external services as well.
- **Data arriving in the platform are stored automatically on the Storage** (Open Search) and / or can be sent directly to the user interface or to data analytics for stream processing. The same flow works as data driven from entering data/entities to dashboards and from dashboards to data streams for acting and sending commands, alarms, etc. Data collected and stored can be browsed by using the **Entity Inspector** / **Data Inspector** and **Data Managers (HLT)** for all the High-Level Types managed in the platform. From Entities, IoT, POI, maps, heatmaps, traffic flow, etc. For some HLT specialized storages can be added to reach higher performances, fully transparent for the users and developers.

- ***storage and semantic models and reasoners***. In Snap4City, the arrival of a new entity/data model does not imply for the developer to adequate the storage structure, but to model it on the Entity/IoT Directory once. Once a new Entity Model is registered, any Entity can be created from that model, and data massages of them can be automatically ingested in the platform, any instance of the new data can be directly deployed, and data conformant with the model is automatically ingested. Entities/Data entering in the platform are automatically (i) stored (in **Figure 3**: Open Search), (ii) semantically indexed, (iii) directly sent to the user interface, or accessible for data analytics and/or stream processing. The same data flows work as event driven coming from dashboards to data streams for acting and sending commands, alarms, etc., in the platform and/or to external devices via brokers or processes. Data / entities stored (in some literature, the storage for IoT is called data shadow) can be browsed by using the Entity Inspector/Manager, for all the entity/data types managed in the platform. From simple data entities to IoT devices, maps, heatmaps, flow, matrices, 3D interactive representations, scenarios, etc., as needed by the DT.
- ***Data Analytics*** can be developed in Python, RStudio, MapReduce (only in the presence of early snap4City solutions which were based on Hadoop, then deprecated for their limited performance and reliability), etc., and can be activated from: (i) Processing Logic / IoT App (Node-RED), (ii) scheduled processes, (iii) data driven events; (directly on cloud or on developers' computers). Data Analytics may exploit stored data via some API to access at the data in the storage and may implement any kind of solution based on machine learning, AI, XAI models, tool, simulations, library or dedicated hardware you need, as NVIDIA, using CPU/GPU, HPC, also exploiting MLOps as ClearML manager, etc.
- ***data representation front-end generation with server-side and client-side business logics***. User interface of smart applications can be easily developed by creating HTML generation tools, such as Dashboard Builder [**Dashboards2019, Dashboard2024**], which allows to select the graphics views and connect them to data/entities on storage and to data streams and brokers. Graphic widgets can be configured and scripted to create business intelligence tools by using JavaScript on client- (Client-Side Business Logic, CSBL) and/or on server-sides (Server-Side Business Logic, SSBL). In the latter case of SSBL, the business logic is implemented via Processing Logic as Visual programming Node-RED. Also, the CSBL can be coded using a visual editor. A number of tools are available for the customization of the user interface: dashboard builder editing and composition, wizards, synoptics development tools and templates, templates for the user interface themes, micro applications, etc.
  - **Smart Applications** can be easily developed by creating the user interface with the Dashboard Builder, which allows to select the graphics views and connect them to the data of the storage and to the data streams. It is possible to glue and activate each other the graphic widgets with JavaScript on CSBL or demanding the implementation of the business logic on SSBL or doing a mix. Server-Side Business Logic is implemented by using Processing Logic / IoT App via visual programming. A number of tools are available for the customization of the user interface: dashboard builder editing and composition, wizards, synoptics development tool and templated, templates for the user interface, ready to user micro applications, etc.
  - **According to Figure 3, the developers can exploit macro-functionalities of Snap4City such as: digital twin support, routing of several different kinds, Heatmap productions from data, predictive algorithms, measuring tools for GIS on maps and floors, comparative tools for orthomaps,**

notification manager, KPI support (SUMI, 15MinCityIndex), scenarios, What-if, Simulators, simulation manager, etc.

- **All the data streams (internal and external) are protected, authenticated, and authorized according to [Security2020], Pen Test passed, GDPR compliant, etc.**

In short, conceptually, from left to right of the above figure:

Data Ingestion, transformation, aggregation and unification, as well as **Data Publication** are performed in push/pull by using: https://www.snap4city.org/download/video/course/p3/

- **Processing Logic / IoT Apps** in Node-RED and Snap4City Libraries/palette to perform data ingestion, data transformation, integration, data storage, business logic, and implementing ETL and ELT (extract transform load, extract load transform), including the exploitation of a large number of protocols: WS, FTP, REST Call, etc.
- **Brokers**: FIWARE IoT Connectors, IoT Agents, IoT Adapters, Brokers, IoT Brokers, also integrated with Processing Logic / IoT Apps with dedicated MicroServices and the **Entity Directory** / IoT Directory,
  - **Typical broker of Snap4City is ORION broker which is provided for default** with Snap4City Authentication and Authorisation.
  - **Additionally (optionally) also MQTT Broker** is provided with Snap4City Authentication and Authorisation.
- **Tools** developed in Python/RStudio or other languages which can exploit the Smart City API of Snap4City for data ingestion, transformation, and registering Entity Instances from Entity Models, etc. Some of these tools are accessible as open-source processes to be customized,
- **GIS** data ingestion/publication via WFS/WMS, also integrated with **Processing Logic /** IoT Apps; for example, using **GeoServer**, ArcGIS, etc. **GIS data can be also directly shown (without data ingestion) on dashboards exploiting WFS protocol and formats.**
- **digital twin support,** provided on all Snap4City platform including MicroX via Dashboard Builder on a selector.
- **comparative tools for orthomaps,** provided on all Snap4City platform including MicroX via Dashboard Builder on a selector.
- **measuring tools for GIS on maps and floors,** provided on all Snap4City platform including MicroX via Dashboard Builder on a selector, under beta testing.
- **What-if:** provided on all Snap4City platform including MicroX via Dashboard Builder on a selector.
- **routing of several different kinds**, only accessible on platforms in which the router is installed and the Application is provided.
- **predictive algorithms,** only accessible on platforms in which the predictions modules are installed to make them usable via API or via a dedicated application. They are distributed Open Source.
- **Heatmap computation,** only accessible on platforms in which the predictions modules are installed to make them usable via API or via a dedicated application. They are distributed Open Source.
- **notification manager:** sending notifications via several different methods as SMS, Telegram, email. Something can be directly implemented in Node-RED, partially also on Micro X
- **Participation manager:** collecting complaints from mobile apps, web apps, and managing QR questionnaire; processing text and complains via **SnapAdvisor**, not provided on MicroX.
- **KPI support (SUMI, 15MinCityIndex):** computing SUMI and 15MinCityIndex, not accessible on MicroX
- **Scenarios:** creation of simple and complex scenarios, two different tools and provided, accessible on all Snap4City platforms including Micro X**.**
- **Simulators, simulation manager:**
- data gathering tools such as **DataGate/CKAN** for open data (https://ckan.org) are also integrated with IoT Apps by using dedicated MicroServices, (not present in all Snap4City installations, optional on MicroX)
- **Satellite data service** using dedicated Python processes also integrated with **Processing Logic /** IoT Apps, (not present in all Snap4City installations, optional on MicroX)

- ***Data Table loader*** *and POI Loder for Excel files ingestions including IoT Device Data (Entity Instances) and POI, for bulk data ingestion, also integrated/developed with* ***Processing Logic /*** *IoT Apps which can be customized for each organization or for multiple table/excel kinds (not present in all Snap4City installations, optional on MicroX) (This tool is* ***practically deprecated*** *since a new tool is under development based on* ***SnapAdvisor****)*
- ***Web Scraping via Portia****, also integrated with* ***Processing Logic /*** *IoT Apps. (not present in all Snap4City installations, optional in MicroX) (this tool is* ***deprecated*** *since the crawling can be performed by* ***SnapAdvisor*** *automatically and/or by Node-RED by coding)*

In our experience, **99% of the data ingestion/transformation processes can be easily implemented in Processing Logic / IoT App Node-RED and managed with one or more Processing Logics / IoT Apps**. Processes can be internally scheduled, and automated backup of node-red flows and versioning can be performed using Node-RED features. Only high throughput processes should be activated by using Python that can be controlled by **Processing Logic /** IoT App and deployed in Container, also managed by Snap4City infrastructure, or can be actually put in execution in other manners. The Python processes proposed by Snap4City can expose REST API. A template for developing Custom Tools in Python is also provided by Snap4City. For Performance analysis please see the installation page: https://www.snap4city.org/738

In the context of **big data architectures, the storage is a facility** which does not need to be designed in terms of tables and relationships among tables by developers (as in the old-style software engineering approach). All the data entities are modelled in terms of: Smart Data Models, Entity Models, IoT Device Models, etc. Therefore, in Snap4City, all the data models are **Entity Models /** IoT Device Models, also the FIWARE Smart Data Models are present into Snap4City as **Entity Models /** IoT Device Modes which is a higher level of model abstraction and brings automatically the relationships into a knowledge base which is a semantic graph database (an evolution of reticular databases). So that any data record in the big data platform has a definition in terms of **Entity Models /** IoT Device Model, which is a **Data Model**. Therefore, in the context of IoT, IoE and WoT Snap4City manages different entities such as: FIWARE Smart Data Models, Snap4City **Entity Models /** IoT Device Models, **Entity Instances /** IoT Devices, custom devices, and IoT Brokers/Brokers, that are registered via **Entity Directory** / IoT Directory, a multiprotocol multi-broker tool for IoT Network management. Among the brokers, a major role is played by Orion Broker of FIWARE by which the platform support NGSI with Services/Tenant and Service Paths. The **Entity Directory** / IoT Directory is capable of automatically deploying Orion Brokers on demand. You can connect/register any Broker to Snap4City **Entity Directory** / IoT Directory. In this case, the broker is regarded as an External Broker, not controlled by the Snap4City platform. External Orion Brokers can be harvested for registering their data on the platform in a fast manner, thus reducing connection times to existing infrastructures. A detailed description of data capabilities and networks is provided in: https://www.snap4city.org/download/video/course/p3/
The Snap4City platform supports a very large number of protocols push/pull, sync/async. Among them are MQTT, NGSI, COAP, OneM2M, ModBus, OPC, WMS, WFS, and AMQP. To get a larger list of supported protocols see Https://www.snap4city.org/65 .

For **Data Formats,** the Snap4City supports a large range of **High-Level Types**, *HLT:*

- *Entity Instances, IoT Devices, Sensors/Actuators, POI, KPI, predictions, events of several kinds, social media,*
- *FIWARE Smart Data Models, Entity Models,*
- *Trajectories, paths, etc.*
- *Origin Destination Matrices, demand of mobility, offer of transportation*
- *Vector Field, Flows, high of waves, slope field directions and intensity, wind flows, etc.*
- *Traffic Flow, People Flows, with animations,*
- *Heatmaps, Satellite data, with animations,*
- *3D BIM building, 3D shapes of city,*
- *Building Floors,*

- *trajectories, routes,*
- *TV Camera, TV Cam streams,*
- *public transport offers, from GFS and other formats*
- *synoptics,*
- *scenarios, city scenarios, etc.*
- *predictions,*
- *parking lots, parking areas, etc.*
- *traffic light plans, semaphores plan.*
- *custom formats, user profiles, etc.*
- *entities with geographical shapes and grids,*
- *cabinets, smart light, waste bins,*
- *environmental sensors, pax counters,*
- *thermal cameras, CNC machines,*
- *etc.*

A Digital Twin is an HLT instance and may present several connections to other data kinds, which in turn are instances of other HLTs. Each Digital Twin and element can be accessed via the so-called Data Inspector.

To remark, *the Entity Models (IoT Device Models) can be used as template to create Entity Instances (IoT Devices). The Entity Instances (IoT Devices) are those that can receive messages to update their data over time, and thus to create time series. The Entity Instances and thus most of the instances of the High-Level Types need to be referred from other entities or to see them on Dashboards, etc.*

References to **Entity Instances** are established via the so-called **ServiceURI** in the Knowledge Base, Service Map (also called Device URI in Entity Directory / IoT Directory). We can define the **ServiceURI** as a URI identifier of an Entity in the solution and it is defined according to the international standard definition of an URI: https://en.wikipedia.org/wiki/Uniform_Resource_Identifier In Snap4City, **ServiceURI**, also called **SURI**, for example as: http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO759 if you put the SURI (of a public entity) on a browser its definition appears since it is compliant with the Linked Data standard for the SURI subject, providing predicate P and Object O.

p	o
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/ns/sosa/Sensor
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.disit.org/km4city/schema#Traffic_sensor
http://www.w3.org/ns/ssn/implements	http://www.disit.org/km4city/resource/iot/traffic
http://www.disit.org/km4city/schema#hasAttribute	http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO759/avgDistance
http://www.disit.org/km4city/schema#hasAttribute	http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO759/occupancy
http://www.disit.org/km4city/schema#hasAttribute	http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO759/thresholdPerc
http://www.disit.org/km4city/schema#hasAttribute	http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO759/speedPercentile
http://www.disit.org/km4city/schema#hasAttribute	http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO759/dateObserved
http://www.disit.org/km4city/schema#hasAttribute	http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO759/avgTime
http://www.disit.org/km4city/schema#hasAttribute	http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO759/concentration
http://www.disit.org/km4city/schema#hasAttribute	http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO759/vehicleFlow
http://www.disit.org/km4city/schema#hasAttribute	http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO759/averageSpeed
http://www.disit.org/km4city/schema#hasAttribute	http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO759/congestionLevel
http://www.disit.org/km4city/schema#hasAttribute	http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO759/anomalyLevel
http://www.w3.org/ns/sosa/observes	http://www.disit.org/km4city/resource/value_type/average_vehicle_distance

A large number of details regarding data ingestion are reported in https://www.snap4city.org/download/video/course/p5/ while details regarding data formats are reported in https://www.snap4city.org/download/video/course/p2/ which provides evidence that for each data formats several graphical widgets can be used for its visualization together with other kinds of data.

The Big Data Storage is managed by multiple solutions:

- **Time series** from Entity Instances / IoT Devices and MyKPI are automatically feed into the storage Open Search cluster (which is the new name of Open Distro for Elastic Search of AWS) for storing and indexing data.

  - The Entity Instances / IoT Devices need to be registered into Entity Directory / IoT Directory, which in turn automatically registers them on Knowledge Base for reasoning.
  - The MyKPI can be stored into some SQL database or in Open Search.
  - **We can have time series of: Scenarios, traffic flows, heatmaps, TV cam configurations, Typical time trends, position of vehicles, IoT/WoT, payments, user profiles, moving IoT devices, ODM, Buildings, vector fields, etc.**
- **Knowledge Base**, KB, (based on Km4City ontology) implemented as an RDF store (Virtuoso) which is an index for geospatial, relational, and temporal aspects. The KB can be federated each other, for example for federating different KBs of different organizations. Whenever, a new data model (Entity Instance or Model) is registered in the system, the registration is performed into the KB. Different instances of the KB can be federated via Smart City API by creating a mutual connection among cities/areas of the network, sharing only what each single installation wants to share, and deciding what to share.
- **Heatmaps**, **Orthomaps**, **Traffic Flows Maps**, 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. They can be loaded from Processing Logic / IoT App using specific API. They are also considered time series of heatmaps.
- **Origin Destination Matrices** are managed by the OD/ODM Manager and represented into the Multi Data Map widget in dashboards. They can be loaded from Processing Logic / IoT App using specific API. They are also time series of ODM.
- **Internal Buildings 3D Shapes**, **and Floors** are managed and distributed by BIM Server. Standard BIM tools are used for editing and interchange in IFC formats with standard tools such as AutoDesk Revit, etc. Buildings and Floors are shown in dashboards for their integration with maps and time trends of IoT devices. The BIM are managed on Snap4City by the BIM Manager and can be loaded from the user interface using IFC format.
- **3D City Representation** can be loaded according to different components from which they are composed such as: image patterns, building shapes, 3D shapes, LIDAR data, traffic flows, POI, IoT Devices, etc. [**DigitalTwin1], [DigitalTwin2], [GeneratingDigitalTwin**]. They need to be loaded in agreement with the snap4city platform manager into the Dashboard area.
- **Scenarios, Traffic Flows, Files, TV Cameras, etc., are also specific HLT with special shapes and functionalities which can be managed into the platform.**

Processing Logic / IoT Apps can be used for a range of activities of data ingestion, storage, interoperability, transformation, data driven, management, etc. The activities of **Processing Logic are widely described in other sections.** They are also used for implementing user interface Server-Side Business Logic behind Dashboards to implement smart applications, business intelligence, including interactive Widgets, Custom Widgets and Synoptics, which can be very useful in Smart City Control Rooms. Therefore, with the **Processing Logic /** IoT App it is possible to create end-to-end applications which can get event driven data from the field and rendering data on a dashboard (which can be visualized by many users from different locations and devices and observing the same views/data changes), as well as receive some command from the user to act on **Processing Logic /** IoT App processes and maybe act as actuator. These aspects will be recalled later on this document in the context of Dashboards. https://www.snap4city.org/download/video/course/p2/

Data Transformations (reconciliation, regularization, enrichment, etc.) are implemented through **IoT Apps** (IoT Applications) which are processes for ingesting, integrating and manipulating data with PUSH/PULL protocols from any **External Service** and also from **Broker, and Virtual Entities/IoT Devices** of any kind. **Proc.Logic / 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 190 **Snap4City MicroServices** [MicroServices2019], **Section 3.6.** Data Transformation / Integration may exploit internal and external storage from **any** kind of database, **any** connectors from/to data sources, **any External Service**, **any** direct connection with IoT Devices and Networks, etc. The IoT App can be also used to move data from one Snap4City installation to another. **The IoT Apps may connect multiple user domains and data spaces**. A large number of details regarding the data ingestion/transformation are reported in https://www.snap4city.org/download/video/course/p5/ and in **Section 3.7.**

With Snap4City Terminology, an Proc.Logic/IoT App is a process in the Node-RED plus Snap4City library that is freely accessible on the Node-RED library. The Snap4City Library for Node-RED is based on nodes and call MicroServices, for this reason the nodes are also called MicroServices. Therefore, it is often stated that the Snap4City library in Node-RED provides a large number of MicroServices, meaning that it provides access to the MicroServices via corresponding nodes. Snap4City Libraries for Node-RED can be installed in any Node-RED supported platform including: Linux, Windows, Arm, Android, etc. Then you can install IoT App Snap4City data processing tools on the cloud and Edge, on AXIS cameras, on Raspberry PI, on mobile Phones, etc.

Data Analytics are processes written in RStudio or Python that can: (i) perform data access (data taken from outside or from the Storage/KB of Snap4City Platform), (ii) apply statistics, operating research, Machine Learning, Artificial Intelligence, Explainable AI, deep learning tools, reinforced learning, generative AI, algorithms also exploiting Tensor Flow, CUDA, Keras, Pandas, etc., (ii) produce results as predictions, early warning, hints, new data, heatmaps, simulations, anomaly detection, etc. Snap4City Data Analytics in RStudio, Python and IoT App can be developed by using offline and/or online development environments which also allows putting in execution of the processes in Containers which expose their API/MicroService for Processing Logic / IoT Apps and other tools in the platform. Off-line development environments are viable as well as, if preferred. The recent improvements of the Snap4City platform include the possibility of exploiting clusters of GPU/CPU via MLOps (such as ClearML) and can see processes of AI/ML and any other process activated from CSBL or Node-RED via specific APIs. This approach is very powerful and can be used for extending the capabilities of smart applications. The API could be made accessible from external applications if mapped on Firewall. With Data Analytics processes you can produce/compute: predictions, early warning, traffic flow reconstruction, classifications, clustering, trajectories, anomaly detection, KPIs and indicators, Typical Time Trends, typical trajectories, routing and paths, multimodal routing, 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 cases are described in: https://www.snap4city.org/download/video/course/p4/

Smart City APIs (**SCAPI**) are the main entry point for accessing data from Big Data Storage (KB and Open Search). They 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, etc. All the data and services are accessible via the Smart City APIs which are used by front-end tools such as Dashboards, Web and Mobile Apps, and MicroApplications. Details regarding Smart City APIs are reported in: https://www.snap4city.org/download/video/course/p7/ . **Smart City APIs provide a large set of facilities** to get data filtering them by: area, path, GPS locations, distance; date and time interval; value of the variable of the models; Service URI of the entities, etc. **Advanced Smart City API**s, (**ASCAPI**) may include Smart City APIs and all other APIs for: Entity Directory / IoT Directory management, Orion Brokers direct access (if needed to get the last data and perform subscription on the eventual NGSI events), heatmap management, flow management, dashboard management, user management, process management, etc. Advanced Smart City APIs are used to federate Snap4City platforms, are documented with Swagger, and can be controlled for accounting data access and billing according to different business models. The API are authenticated and provide data according to the authorization accesses of the user as described in the following. So that, they are very useful for implementing services from Mobile App, etc. The Snap4City platforms can be federated each other via Smart City APIs. The Smart City API can be also exposed via **Snap4City API Manager** open-source tool for managing the API accounting. In the API Manager also the APIs from: containers, Ai, ML, MLOps are managed.

Dashboards can be created by using the Dashboard Builder (while Open Search Dashboard (former Kibana, sister of Grafana, is only used for developers to monitor the data flow). The dashboards are used by different kinds of users such as: decision makers, city operators, ICT operators, private users, etc.. They can be suitable for implementing

- Smart City Control Rooms with video wall, and Situation Rooms with touch panels
- operation and planning applications for city Operators on Desktop of multiple monitors,

- mobile apps for mobile operators.

Dashboards can exploit/represent manipulate all kinds of data, HLT, and Data Analytics, legacy services, and special tools such 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 a 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.js, Highcharts JavaScript, etc. https://www.snap4city.org/download/video/course/p2/ Also external web site (services) can be embedded into dashboards. Dashboards can be invoked each other, can be parametrized to show specific data, can provide intelligence coding in CSBL, can access to external data, etc. [**Dashboards2024**].

In Smart City Dashboards, the most critical feature is the capability of the Map Widget to represent integrated data such as: smart data model, entities, IoT Devices, POI, KPI, traffic flows and animations, heatmaps and animations, paths, trajectories, map markers PINs and graphical animated PINs, origin destination matrices, vector fields, 3D shapes, changing orthomaps, comparing orthomaps, performing measures on the maps, and the rendering of all the other time series connected to it in few clicks in multiple rendering manners.

In **Industry 4.0 Dashboards, the most critical feature is the capability to be interactive, real-time, end-to-end secure, and to provide synoptics**, graphic custom widgets, 3D representation of the plant, and the rendering of all the other time series connected to it in few clicks in multiple rendering manners.

Dashboards may be presented with custom rendering styles/themes. Presently more than 20 different styles are made accessible with a manual for their production and customization See **Section 3.10.4**.

Examples of dashboards and how they can be authored are reported in https://www.snap4city.org/download/video/course/p2/ **and Section 3.10.**

In Snap4City, **Global and Local Digital Twins are integrated** with IoT models, maps, heatmaps, etc. The so-called Digital Twin approach implies managing a digital counterpart of the real world, see **Section 3.4.4. Despite the nice 3D effect and immersive feel of the resulting rendering**, the balance of costs with details is fundamental, as well as the connection of details with Local Digital Twin and with all the other HLT of the platform. **Local Digital Twin**: detailed digitally modelled aspects of a physical element, for example a chemical plant with machines, motors, silos, etc., each with 3D representation, detailed components, and real-time values of attached sensors that can be used to understand its behaviour in terms of real-time data and functionalities, mechanical parts to be disassembled for maintenance and inspection, etc. Different levels of resolutions can be provided, and rendering is typically performed on 3D modelling and visualization tools such as the BIM server. The Local Digital Twin also may include a detailed description of floors, detailed description of data, devices, etc. **Global Digital Twin** may represent the 3D shape of the city with roads, buildings, gardens, flows. In the case of the city, the representation can be very detailed as in the Local Digital Twin or simplified in shape and volumes. In the latter case, the modelling can be based on volumes with geometry projections, facades and roofs patterns, exploiting specific attributes such as “height”, “roofShape”, etc.

Aspects of Platform **Management and Quality Control** of back-office are presented in **Sections 3.16** with user activity management, 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/course/p6/

Web and Mobile Applications can be created by developers exploiting **Advanced Smart City APIs - ASCAPI** and may be controlled by Snap4City tools. For example, to send on Mobile Apps engagements, polls, stimulus, and than to monitor user behaviour, create origin-destination matrices, get reactions from the city users, inform user at the onset of critical conditions, etc. Accounting and billing can also be used to control consumption via Mobile and Web Applications. Details regarding Smart City APIs and their usage for the development of Web and Mobile Apps with related development environments are reported in: https://www.snap4city.org/download/video/course/p7/

3.1 – Snap4City Development Environment and Life Cycle (overview)

HOW TO develop using Snap4City/tech tools, please download and read the Development Life Cycle guidelines:

https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf

Smart Applications can be developed by producing: Processing Logics / IoT Apps, Data Analytics and Dashboards with almost no coding activities on the functional aspects and for server side.

Coding may be needed into:
-- Processing Logic / IoT App Node-RED in JavaScript only
-- Dashboard widget Client-Side Business Logic if needed in JavaScript only
https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf
-- Data Analytics: if needed and realized in Python and/or RStudio

Smart Applications Developed on Snap4City implies to create your artefacts and smart solutions, decision support systems, business intelligence tools, simulation, visual analytics tools, etc., in fully control of your Intellectually Property and not open source.

You Applications and IPR in Snap4City are composed by:

- **Data Models**: IoT Device Models, Smart Data Models, Entity Models, etc.
- **Processing Logic / IoT App**: data ingestion, adapter, transformation, wrappers, business logic, transcoding, integration, interoperability, algorithms, etc.
- **Data Analytics:** algorithm and processing in RStudio or Python, ML, AI, XAI, etc.
- **User Interface Design**: Dashboards, client-side business logic, Synoptics, widgets, templates, styles, etc.
- **Client-Side Business Logics (if any)** realized in JavaScript on Dashboard widgets.
  - https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf
- **Server-Side Business Logics (if any)** realized in Processing Logic as Node-RED and JavaScript.
- etc.

The fact that you have developed your solution in Snap4City (which is 100% open source) does not imply that the artefacts and your smart solutions are open source as well. They are 100% under your control/IPR, and you can decide how to release them, which kind of licence to impose on them. This also implies that Snap4City has no rights on what you have developed using Snap4City development tools. In fact, it is totally equivalent of developing a program with any other open-source development tool. For example, if you develop a database with MariaDB, the tables and queries are not open source; another example: if you develop something on NiFi Apache which is open source, your flow is not open source; another example: if you develop something on ECLIPSE which is open source, your Java/PHP/C++ Code is not open source, etc.

If you are developing some smart solutions on some platform labelled as *Powered by Snap4City* technology, the platform manager and owner may have specified some rights on what you have developed according to the terms of use THEY expose on their own platform, please read them in any case. For example, the rights to off and discharge your solution if it creates some problems or violate the rules imposed by the service. For example, ethical rules.

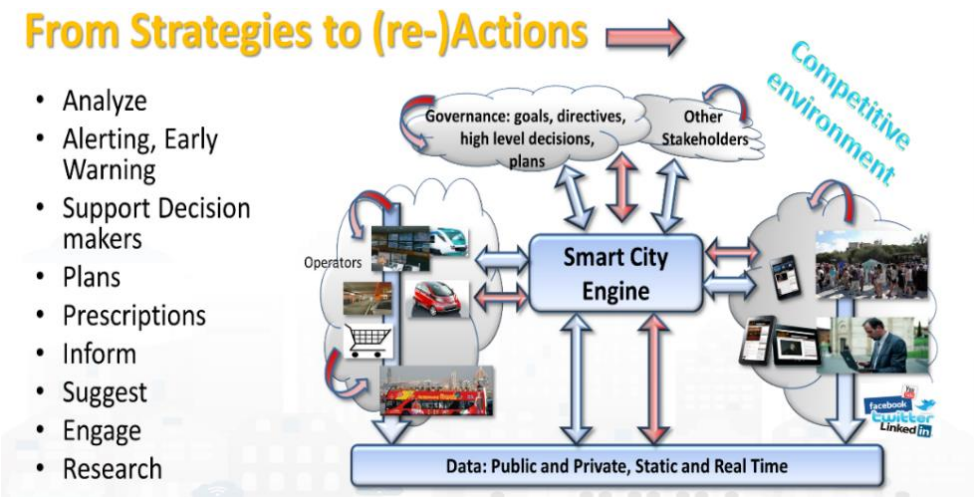
3.2 – Snap4City Innovation Methodology and Living Lab

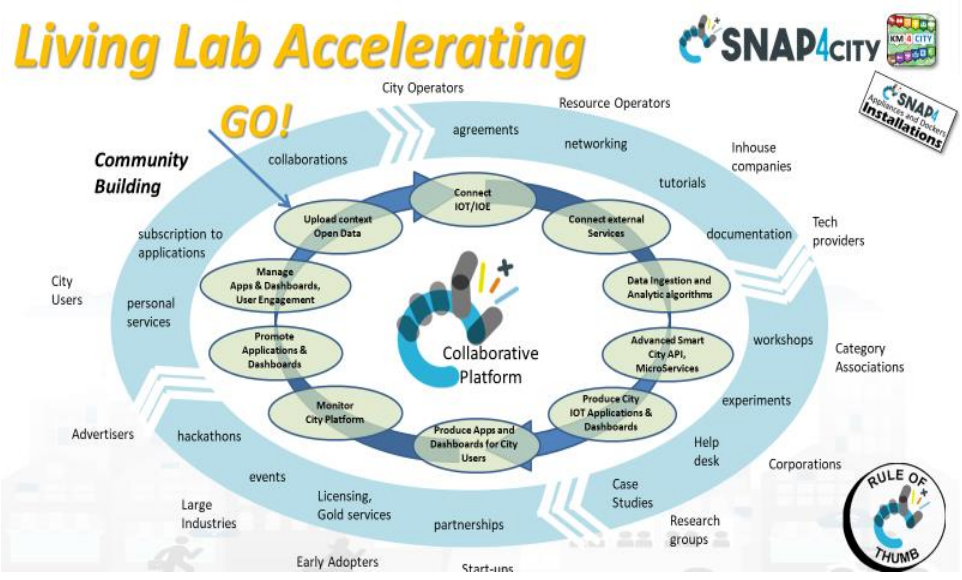
Nowadays most cities must rapidly transform their services to face the ever-changing social, environmental, energy and economic challenges, without losing their character. In recent years, smart vertical solutions have been seen in many cities, some of which have met specific needs with good results; some have proved inadequate to compete on new challenges, in some ways more structural. This approach has proved valid in various contexts, but not in all aspects of city management. Today, the focus has shifted from vertical solutions to infrastructures and to a functional vision focused on the needs of the evolution of the city, ending up asking technology for digital solutions to improve the quality of life. Digital Twin models should be the first to be built to understand the evolution and reactions of cities to natural events or not in synergy with the physical buildings. 3D digital models, proposed to attract tourists or consumers to virtual worlds on the Internet, are unlikely to model and represent the functional aspects of city management. To pursue these goals, cities have invested in smart ecosystems in recent years to provide high-quality services to citizens and businesses. Sensors and/or actuators in strategic contexts and locations help cities optimize their services, reduce costs and improve physical infrastructure performance. But now, there are new challenges to increase the quality of life, such as reducing environmental problems and energy sustainability. From these various trends and indicators seeking to provide measures on the progress: the push towards cities in 15 minutes (for which primary services should be accessible at most in "15 minutes on foot") [**15MinCityIndex2021**]; the Sustainable Development Goals, SDGs, of the United Nations (for which cities can be more committed to achieving some of the 17 SDGs, https://sdgs.un.org/goals); the achievement of the European Commission's objectives in terms of NO2, PM10, PM2.5 emissions (https://environment.ec.europa.eu/topics/air_en).

The drive towards social innovation is growing, with the creation of a virtuous and shared ecosystem for the management of services and therefore of data to allow better decision-making for the government, businesses and residents of the stakeholders. This approach can be seen as an application of the **Quadruple Helix model** to the smart city domain and can attract city actors such as, government, research, industry and society to prepare an environment in which to trigger a continuous collaborative process of innovation in which the needs of citizens and the quality of life are central. Among the different approaches, enabled by the Quadruple Helix model, we can find the Living Lab solutions. These provide a collaborative environment in which the various activities are inserted and activated available to all stakeholders based on their skills, profiles and expectations, and where governance must be shared on consultation tables. The various partners in the area (the stakeholders: industry, trade, universities and research centres, and citizens) can provide real channels in which information and actions can flow in both directions, provide data but also produce innovation and value from which to benefit of ecosystem growth.

In the context of smart cities, the Smart City engine plays a relevant role in 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, to support decision-makers, new plans, and the development of strategies for the city.

Smart Cities need to set up a flexible **Living Lab** to cope with the city's 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 IoT/ IoE technologies and Big Data analytics. The **Snap4City** platform and method have been realized involving different kinds of Organizations (Universities, SMEs and Large Industries, Public Administrations) and city users (City Operators, Resource Operators, Inhouse Companies, Tech Providers, Category Associations, Corporations, Research groups, Strat-ups, Early Adopters, Advertisers, Community builders, etc.), thus reflecting the features, described in **Quadruple Helix** (QH) model, to facilitate the Living Lab approaches in a Smart City (see the next figure). The innovative aspects of the solution proposed are related to the semantic computation of entities for the discovery and search of 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.





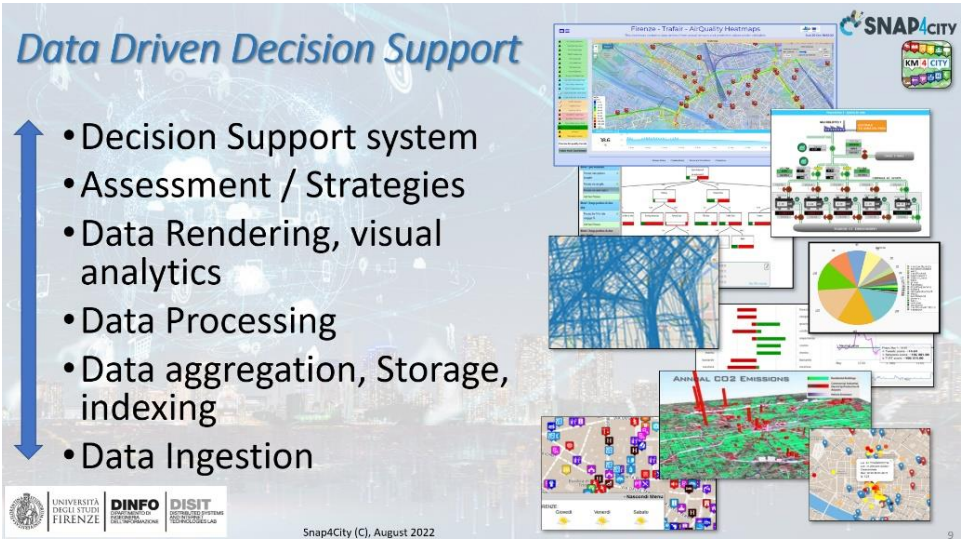
One of the first activities for setting up a **Living Lab** in a city is the creation of the technical infrastructure which in turn is grounded on many valuable enabling tools. They must support the city in: data modelling; uploading contextual and open data; the connection of IoT/IoE sources and external 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 via the web. To this aim, the **Snap4City** platform provides a collaborative environment in which different kinds of stakeholders can mutually collaborate. At the same time as the setup is created, the collaboration among stakeholders can start

by creating: agreements, collaborations, networking, production of tutorials, workshops, hackathons, etc., to get to involve stakeholders around use cases, and finally sign partnership contracts, licenses, etc. Therefore, the provision of specific solutions to city users, operators, etc., is becoming possible. This process must be driven by the municipality itself, and, on the other hand, the municipality needs support for technical aspects in case it is not very large and technology-oriented. Typically, individual companies even if participated by the city or by city operators, do not have the vision and the mission to share such a broad multi-domain multiservice framework and environment. [**LivingLab2018**], [**Gov2018**], [**Gov2020**].

The approach involves a methodology to produce the **Snap4City innovation matrix**. It is implemented through a series of workshops (in presence or online) in which an n in-depth analysis of the current situations and the needed steps in the direction of a rapid optimization of processes is produced, taking the most efficient direction and the most acceptable solutions for stakeholders.

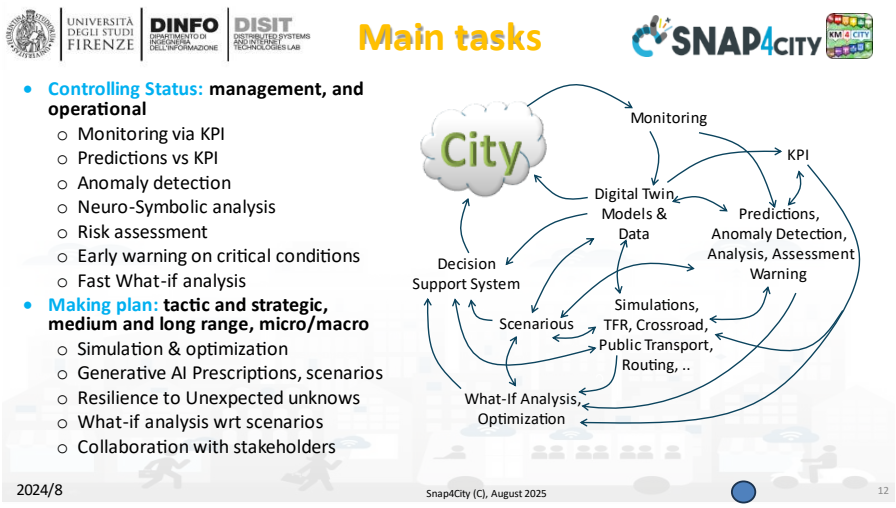
See **training course part 6**. https://www.snap4city.org/download/video/course/p6/

The scenarios identified during the workshops may be focussed on specific domains (mobility, energy, government services, water, environment, etc.) and to improve services to specific user categories. To this end, a process of **Data Discovery** is also carried out in dialogue with stakeholders and their technicians to verify the possibility of collecting the necessary and available data as well as to create new network of IoT Devices to measure the phenomena that are currently under attention. Keep in mind that typically the same Scenario (target smart solution) may be realized starting from different kind of data. In other words, the scenario is the target, the data are tools, and the experience may help you identifying among the different data sources that may be simpler to be acquired and exploit, technically and legally speaking.



Today, smart cities meet new challenges to: (i) better plan the management of tourist flows, (ii) increase efficiency in the management of city maintenance events, (iii) be ready to reduce response times to unexpected events, (iv) optimize the global mobility of the city by harmonizing public transport with sharing solutions, (v) improve services for parking light and waste, (vi) stimulating and managing engagement with city users, etc . They can bring concrete benefits to the city and a certain extent also to operators, such as to motivate them to contribute to the development of innovative solutions. The investments are huge, and only through synergistic actions can the resources to face the new challenges be found.

The new challenges of smart cities are fought with the development of decision support solutions (DSS, Decision Support Systems), which based on objective information try to mediate between multiple objectives such as increasing the quality of life of citizens, improvement of services, cost reduction, innovation, attractiveness for tourists and/or industries and/or commercial activities, etc. To this end, it is necessary to equip oneself with highly interoperable flexible and dynamic analysis models and tools to make the most of the heterogeneous data coming from the territory, the structures and services of the city, and from stakeholders. All information, knowledge, and data must be aggregated in integrated Digital Twins to be usable by simulation tools and short, long, and very long-term prediction algorithms in order to analyse incipient scenarios but also future situations for structural planning, and for what-if analysis (given a scenario not previously studied, the system must be able to respond in real-time with simulations and prescriptions). Consequently, smart city infrastructures must be able to exploit integrated Digital Twin models, apply conditions and scenarios and provide the decision-making process with indications and prescriptions on how the system could / should evolve. In this regard, the Artificial Intelligence solutions with modern techniques allow explaining to the decision maker the reasons for the predictions and prescriptions, and simulations: **XAI** (**Explainable AI**). These findings should be accessible for operators, and decision-makers but also for all stakeholders and citizens, to illustrate and discuss possible solutions and development plans with them. The XAI techniques provide public decision-makers with ethical support and detailed explanations of the motivations underlying the suggestions provided by the model so that they can understand the process and the reasons and reassess the technical, ethical, and social aspects of the choice, as if listening to a trusted expert.



3.3- Snap4City Data Sources/Recipients, Bidirectional Channels

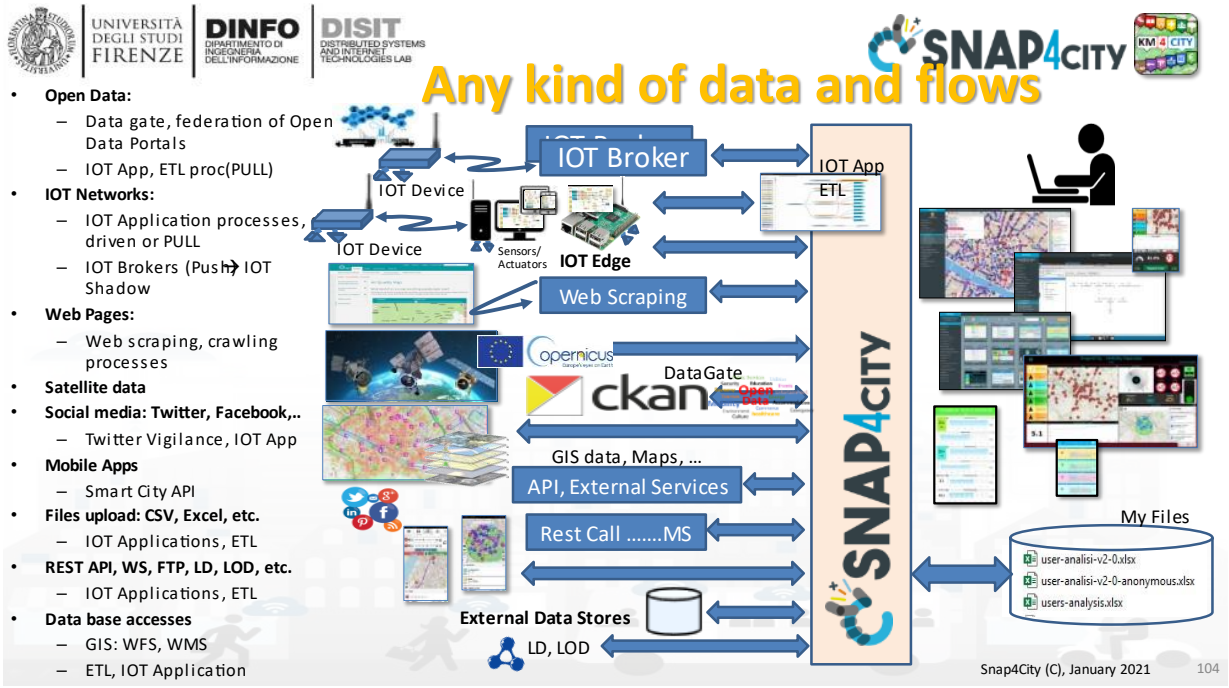
One of the first steps to set up a Smart City infrastructure can be the data ingestion or the addition of new services, applications, tool and connections with data sources/recipients.

The whole development process for smart applications is described in:

https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf

In **Snap4City, the activities of data ingestion can** be performed in several manners, taking data and information from any kind of data course, and also produce them. All the connections could be bidirectional on the basis of the protocols, while complexity is reduced for the developer using of the **almost no coding platform as Snap4City**.

Snap4City platform can interact with legacy installations (of any vendor, protocol and format) on IoT Networks, and therefore with Edge solutions, Edge/Fog Devices, Gateways, etc. They can communicate via their protocols or using the direct connections with Snap4City exploiting secure/protected communications. Edge systems/devices executing Node-RED (e.g., windows, Linux, raspberry pi, android, etc.) can directly install Snap4City library to use from the platform: data ingestion processes, data model registration, data analytics, dashboards, and direct interaction with dashboards and users.



3.3.1 – Main channels for data ingestion/interoperability

See **Section 3.6** and **Section 3.7** for **interoperability**.

- https://www.snap4city.org/download/video/course/p3/
- https://www.snap4city.org/download/video/course/p5/

Proc.Logic / IoT Apps are developed in **Node-RED + Snap4City Libraries** published from the Node-RED foundation. They can be used to automate a large number of back-office activities such as: data collection, transformation, aggregation, and publishing, according to push/pull protocols; 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 Proc.Logic / IoT App implementing ETL and ELT (extract transform load/extract load transform), but also business logic for dashboards.

See the Development Life Cycle Document: https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf

Proc.Logic / IoT Applications can get/send data/messages via any kind of

- **Brokers:**
  - **Orion Broker FIWARE**, NGSI V1, V2, LD, with **Services/Tenant** and **Service Paths**;
  - **MQTT**, with open ID connect or without.
  - **exploit**: IoT Connectors, IoT Agent, IoT Adapter, wrappers, IoT Edge Devices;
  - **any other 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.
- **Storage, save, search, and retrieve:**
  - **Snap4City Storage** which is Open Search and Knowledge Base
  - **Databases DBs** and more, such 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/Proc.Logic.
- **APIs on external services** as:
  - **GIS data ingestion/publication via WFS/WMS**, also integrated with IoT Apps and Dashboards (MultiDataMap widget); for example, using GeoServer, ArcGIS, etc.

    - **IoT Gateways, web servers, FTP servers,** social **media services, linked open data, linked data, ESBs, any kind of data providers** as TIM, Here, etc.
    - **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.
- **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 proc. Logic / IoT App which is the Business Logic of the User Interface, and UI, and/or they can be saved into the platform for further usage. See **Section 3.10.**
- **Data Analytics (in different forms, AI, ML, operating research, statistics, LLM):**
    - Sending data, collecting results and controlling python and/or Rstudio in container via Snap4City Facilities
    - **Custom Tools** developed in Python/R-Studio or other languages which can exploit the Smart City API of Snap4City for data ingestion, transformation, etc. managed by IoT App on Container,
    - **MLOps** for managing AI/ML development life cycle, and on cluster of CPUs/GPUs, HPC
    - **API Manager of Snap4City** can expose any internal APIs (of ASCAPI, MLOps, containers) to be used from internal and external processes such as Web/mobile Apps, Dashboard, smart application, business intelligence tools, AI processes, ML processes, Action LLM, Generative AI.
- **Specific Tools (optional) for:**
    - **Web scraping** can be performed by SnapAdvisor LLM/RAG as well as in Proc.Logic / IoT App
        - ***Web Scraping*** *via Portia (**deprecated**) managed by Proc.Logic / IoT App on Container, 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.*
    - **Excel Files Loader for IoT and POI fast ingestion**, can be performed by: (i) SnapAdvisor LLM/RAG (presently under development), (ii) Proc.Logic / IoT App (accessible in all Snap4City platforms).
        - ***Data Table Loader*** *for shortening the activities of registering devices, loading time series, etc., by loading the data from excel files. (**deprecated**)*
        - ***POI Loader*** *for shortening all the activities of ingesting new data for points of interest providing them as excel files. (**deprecated**)*
- **Files**: of any format and size they can be loaded via toward the **File Manager**:
    - **API**: can permit the load and retrieve of Files. They are also used by the Participation tool to collect media files related to complains. The API can be used by web/mobile applications as one developed by Snap4City.
    - **specific IoT App** to set up specific data ingestion processes such as via Web page, FTP, SSH, DropBox, Google Drive, etc.,
    - **DataGate**/CKAN, for files collection and sharing, API of CKAN can be automatically used into IoT App/Proc.Logic to automatize the retrieve and/or publishing.

Direct ingestion and publication via Brokers connect IoT Networks of Entities Devices. A major role is played by MQTT and by Orion Broker of FIWARE by which the platform support NGSI V1, V2, LD, with Services/Tenant and Service Paths; may exploit: IoT Connectors, Agent, Adapter, wrappers, Edge Devices.

- The 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.
    - Orion Brokers can be deployed on demand by using the Entity / IoT Directory services.
    - Supported data models are FIWARE Smart Data Models, and Snap4City Entity / IoT Device Models. They are automatically registered into the Knowledge base.
- MQTT brokers are also supported without or with OpenID connect compatible with Snap4City.

- Any Entity Instances / IoT Devices and Brokers are registered into Entity / IoT Directory which is a multiprotocol multi-broker tool for IoT Network management, also providing features of IoT Discovery in collaboration with the Knowledge Base. The registration of Entity Instances / IoT Devices into the Knowledge Base from Orion Brokers has performed automatically.

Thus Proc.Logic / IoT App can:

- Create **Entity Instances / IoT Devices** from
  - **scratch** for massive device production/registration;
  - **Entity Models** (FIWARE Smart Data Models, Snap4City IoT Device Models), for massive device production/registration;
- **Generate, load and receive**
  - **Entity Messages / IoT Messages, virtual messages, and simulated data** associated with multiple devices for any purpose in any direction as sensors or actuators.

Other tools for data entry/ingestion are:

- **Web App on a browser user interface on Snap4City or that you can develop**: to add/change data via a web page, for example: **POI, MyKPI**, **IoT Devices**, etc.
  - **See API based development**: https://www.snap4city.org/download/video/course/p7/
  - **See CSBL development**: https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf
- **Web App to request data from Satellite Copernicus on Snap4City.** https://www.snap4city.org/671 with information and data related to vegetation, quotes, humidity, temperature, rain, and many others, which can be taken for large regions and not only on specific sensor points. https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MzAwNQ==
- **Web and Mobile Apps that you can develop:** they typically connect with the platform to get data and suggestions, and they may send data back as preferences, trajectories, queries on sequences, clicks, etc. (if the user agrees, data is 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 city users. The Mobile App users can be registered on the same pool of platform users or may be decoupled from the central user registration of the platform. See **section 3.5.**
- **Participation tool of Snap4City** to collect complains, fines, etc., including Entities and Files which are collected and uploaded on Snap4City.
- **Python processes for Fast Ingestion of Snap4City**, which are available to massive loading of data from other platforms and databased, or for massive ingestion multistream from MQTT brokers or others.
- **Feet Data Ingestion tool from OBU,** and massive data ingestion toward the fleet Management Systems.
- **Data Analytic** processes in any language may create new data and consume, 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, KPIs, Typical Time Trends, typical trajectories, Routing and paths, travel plans, query results, simulations, optimisations, generate scenarios, analysis, calibrated heatmaps, smart parking suggestions, car sharing analysis, etc. See **Section 3.9**.
- **Direct URLs**: such as URI/URLs to perform rest calls 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**.

In addition, it is possible to **generate Reports**: as consumptive views on the platform status on specific programmable aspects for users and administrators, for example at the level of **IoT Devices**, and **Dashboards**.

3.3.2 – IoT Network components

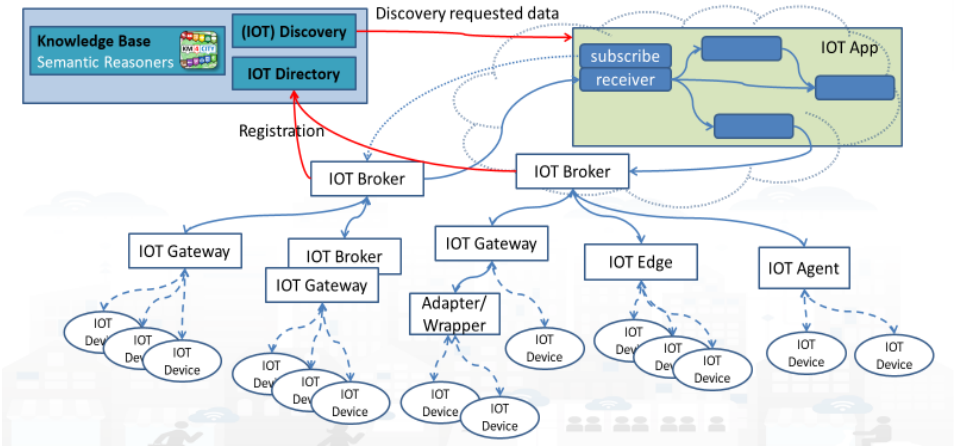
IoT Networks may be composed of one or several IoT Brokers, compliant with different protocols and corresponding devices, IoT Adapters, IoT Agents, etc.

See **training part 3** https://www.snap4city.org/download/video/course/p3/

https://www.snap4city.org/download/video/course/p5/

Therefore, the following elements in addition to those above introduced are relevant:

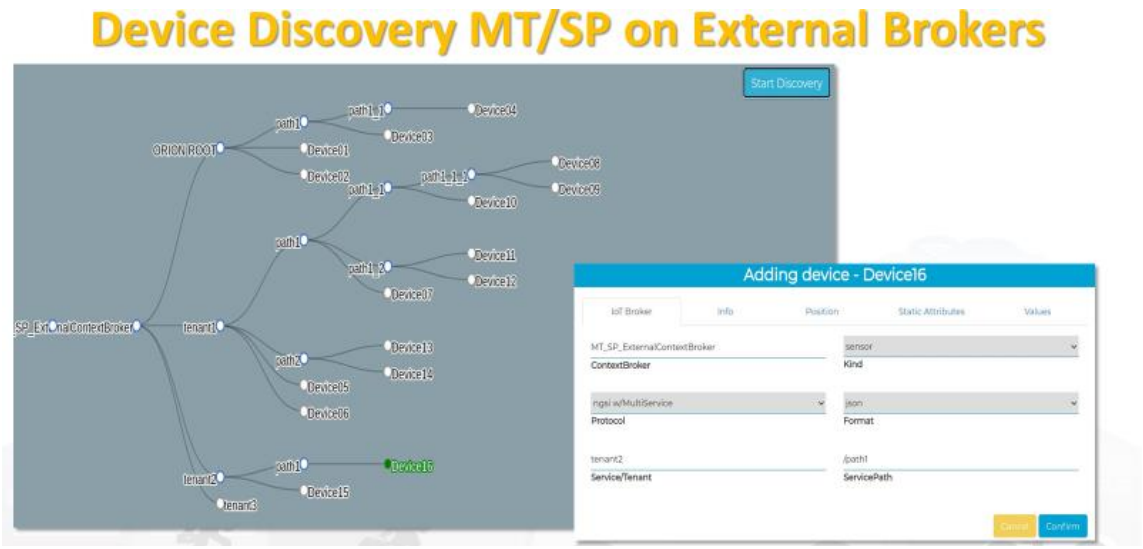
- **IoT Device / Entity Instance**: An Entity Instance / IoT Device may have sensors and/or actuators. In Snap4City, an Entity Instance / IoT Device has to be registered with a Broker before sending data on the platform. If the Broker is internal, the Device is registered on the Entity/ IoT Directory, which registers it on the Broker and the Knowledge Base. While NIFI cluster is subscribed to all Internal Brokers to perform the Data Shadow of all messages.



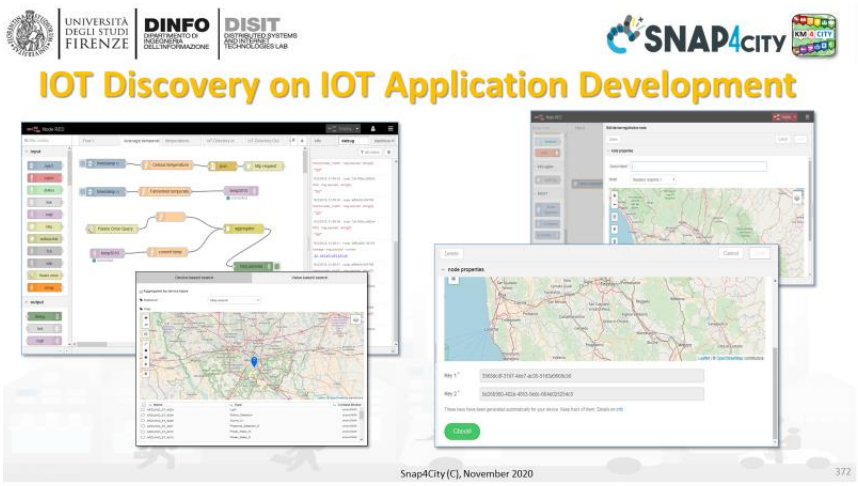
- **Entity Model / IoT Device Model**. Supported data models are FIWARE Smart Data Model, and Snap4City IoT Device Models, which are models for Entity / Devices, Virtual Entities / IoT Devices, time series, moving Devices, POI, TV Cam, Scenario, ODM, etc. They are automatically registered for the KB, creating a unified view of entities from any vendors and broker channel. In Snap4City, an Entity / Device Model can be registered once, shared and used many times for instantiating one or many devices with the same model in a short time, such as 40.000 lights of a smart light system. The Model is a template, and once used can be modified without any impact on the produced device instances.
    - **Concepts of Model and Entities can be also used on MQTT, or other brokers which distribute JSON formats**
- **Orion Broker is a FIWARE** component: https://FIWARE-orion.readthedocs.io/en/master/ The Orion Broker supports 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). To have the Data Shadow, and thus to perform queries on time series, the Orion Broker has to be connected to storage. In Snap4City, the storage provided is Open Search which provides high capabilities for inserting, searching and retrieval and scalability.
- **Entity / IoT Directory:** Snap4City tool for registering Brokers and Entities / Devices. Snap4City is almost agnostic about the brokers (services for Orion Brokers are very extensive) and accepts a number of brokers and protocols registered on the Directory of Brokers and Devices. The Directory is capable to browse on internal and external brokers to

discover (in the case of Orion Broker) Entities / Devices and register them on the Knowledge Base. https://www.snap4city.org/115

The Entity / IoT Directory is the tool that manages all the network information about the deployed and connected devices. It can exploit aspects of the NGSI V1, V2, and LD protocols to inspect and manage FIWARE's Orion Brokers, to manage a large number of them and automatically perform registration of devices in bulk on the Knowledge base. See https://www.snap4city.org/76 https://www.snap4city.org/562, https://www.snap4city.org/647, https://www.snap4city.org/763

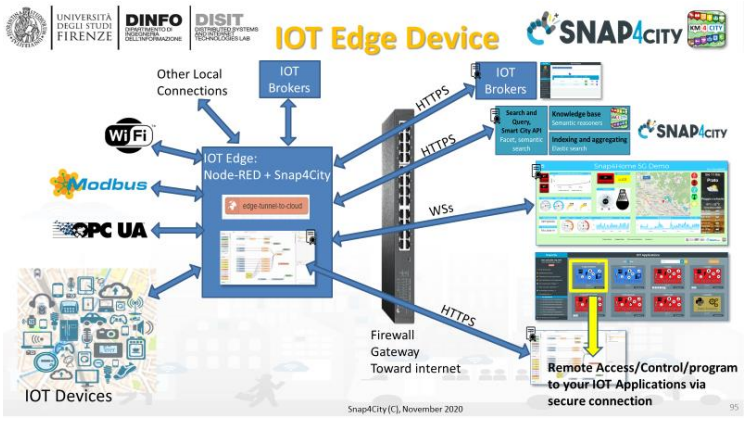


- **Entity / IoT Discovery**: It is a function of Snap4City Entity Directory and Knowledge base which allows discovering the Entities / Devices by several filters: geospatial, by type, by value name, by unit, by nature and sub nature, etc- https://www.snap4city.org/109
  The same feature is provided in Proc.Logic / IoT App, Entity / IoT Directory, ServiceMap, Data Inspector and Dashboard Builder.



- **Edge Processes Entities / Device**: An Edge is typically a Device capable to execute processes. According to Snap4City terminology, Edge Devices are typically endowed with Node-RED process with installed Snap4City Libraries of MicroServices. From Snap4City.org is possible to access the Edge Devices registered with the Snap4City library and specific VPN service active to perform remote programming and maintenance of the Edge Applications.
  See **section 3.5**, and also

- https://www.snap4city.org/646 for the remote control of Edges
- Edges on Linux https://www.snap4city.org/298
- Edges on Android https://www.snap4city.org/278
- Edges on Raspberry pi. https://www.snap4city.org/279
- Snap4Home: https://www.snap4city.org/617
- Snap4industry: https://www.snap4city.org/369



3.4- Snap4City Data Models, Ontology and Digital Twin

According to the previous sections, the data ingested automatically arrives on **Knowledge Base and in the corresponding big data storage**.

The **definition of an Entity Model is** the first step to defining a way for loading data which can be: static, quasi-static and dynamic real-time data, as even driven processes and streams. A model may have an Identifier (the name, the ID), GPS coordinates (Lat and Long, for example for moving devices, or for marking the device position once), the date and time of creation and change, etc. In the Smart Data, IoT world, an Entity / Device (sensors and/or actuator) has a data model. Each **Entity / Device** has associated with a specific broker and organization, in addition to several technical parameters. Moreover, the main differences are based on the so-called **Values**, which are the attributes of the Entity/Device Model. The **Values** are defined in terms of **Value_Name** (the name), **Value_Type** (the kind, for instance, a temperature), **Value_Unit** (for instance Celsius or Fahrenheit), and the **Data_Type** (for instance: float, integer).

See the Development Life Cycle Document: **https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf**

If one has only one Entity Instance / IoT Device with a certain schema/model, the Entity can be manually created, while if you have to create 100.000 of them, or they have to be exchanged in a community or they define a standard model for those entities/devices, thus the need of creating a sort of a template/model for Entity is almost mandatory. The main collections of Data Models templates are:

- **FIWARE Smart Data Models**,
- **Snap4City IoT Device Models**,
- **and Data Spaces** in general

On the Snap4City platform, Models can be defined and **loaded into the Entity/IoT Directory** and can be searched, browsed, inspected and used for creating Entity Instances / IoT Devices on Snap4City Platform.

If the needed model does not match with those accessible in the **Directory**, any developer in Snap4City can **create and register a new Entity Model / IoT Device Model,** which can be published in the Snap4City network, platform. Thus, the Directory can be used to create/register Models, create/register Entity / IoT Devices manually from scratch and from models of any kind.

Once created an Entity Instance / IoT Device, you can associate with it several messages which may differ for: date and time of storage and/or creation, GPS position, values, etc. This the approach to creating Time Series, evolving events, sequences of messages, evolution and versioning of variables, etc. The Entity Instance / IoT Device concept can be equivalent to the main entry for a list of instances, having all the same structure of the main entry. While the Entity Model / IoT Device Model can be regarded as a class defining the data structure. In Snap4City, any Entity / Device can be identified by a **ServiceURI**. In most cases, **an Entity / Device may need to**

refer to other devices, and this can be performed from Values with Value Type = URI, Value Unit as ServiceURI, and Data Type as String. See for details the above-mentioned development life cycle manual.

The ServiceURI is the ID of the main conceptual entity of the Km4City Ontology which is the model of the Knowledge Base in which the entities and models are registered.

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

*Snap4City organizes all Models according to **Km4City Ontology** to build a **Knowledge Base, KB,** that consists of the **Expert System of the city**. It allows us to navigate into the unified data models of the city, exploiting geospatial, temporal and relational aspects. The **Knowledge Base** has several tools for its access and management such as **ServiceMap, Smart City API**, **Wizards** and **Data Inspector**. The **Data Inspector** goes beyond the Digital Twin concept since it gives the operator access to integrated representations of **city entities, devices, processes and users as described in Section 3.4.2.***

Km4City Ontology has been defined to address in a unified manner any city and industry 4.0-related domains, concepts, structures and data models [https://www.snap4city.org/19].

Version 1.6.7 https://www.snap4city.org/download/video/DISIT-km4city-City-Ontology-eng-v5-1.pdf

the update to the 1.6.8 can be recovered from: https://www.snap4city.org/21 and in the following. From that page you can download the ontology as well.

Please note that presently in production on all the KB of Snap4City.org there is the new version of the Km4City ontology and precisely the 1.6.8. This new version includes the concepts for extended Digital Twins and High-Level Types: heatmaps, vector field, building, building 3D, typical time trends, floors, entity groups, garden, traffic flow, etc.
Also, in the MicroX the KB provided is the version 1.6.8, since September 2024.
You can ask to have the new version of KB, while the documentation is under update and publication.

The following text is referring to the former version 1.6.7.

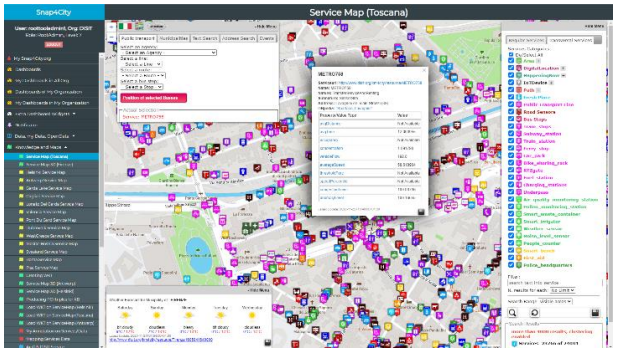
Each data element of every single model aims at modelling and establishing the needed relationships among elements, thus making a general data set semantically interoperable at the model level. In the definition of a **Knowledge Base** grounded on **Km4City**, the usage of the expert system satisfies crucial functional needs as:

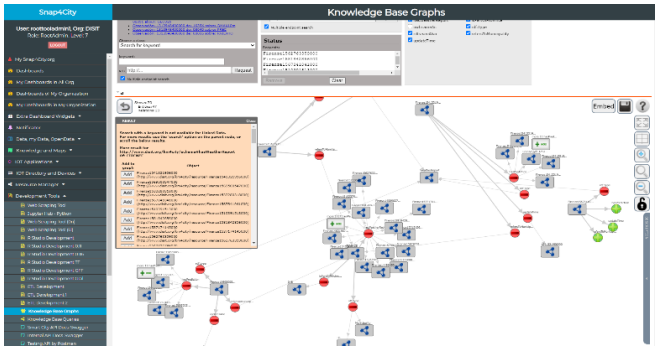
- coverage, the data model is detailed enough to represent everything of interest for real applications and services;
- time-efficient, the Knowledge base has to be queried providing smart services, and users/clients request high performance and scalability (this means applying solutions for design for 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 city domains addressed: mobility and transport, energy, health, economy, key performance indicators, KPI, people flow, commercial, entertainment, culture, sports facilities, industry, government services, financial, education, tourism, security, emergency, agriculture, accommodation, civil engineering, advertising, industry, structure, 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 tracking, parking, RTZ, irrigators, etc. In the same manner, forecast data can be performed for weather, parking, road accidents, people flow, vehicle flows, pollutants, 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 plants such as: water depurators, waste recycling, heating systems for large stadiums and large buildings, fleet management, maintenance team management, copy and publication centers, data centers, etc.

The data modelling at the level of Entities, Data, IoT Device and IoT Broker is not enough to model complex data models of the city such as: GIS information, BIM information, 3D representations, shapes, heatmaps, traffic flow, typical trajectories, typical time trends, cycling paths, schedule and plans, routing, multimodal routing, OD matrices, GTFS of public transportation, etc., while the Km4City Ontology is capable to integrate into a unified model all these aspects and models and POI, IoT, and multiple KB can be federated each other, see section 3.8.





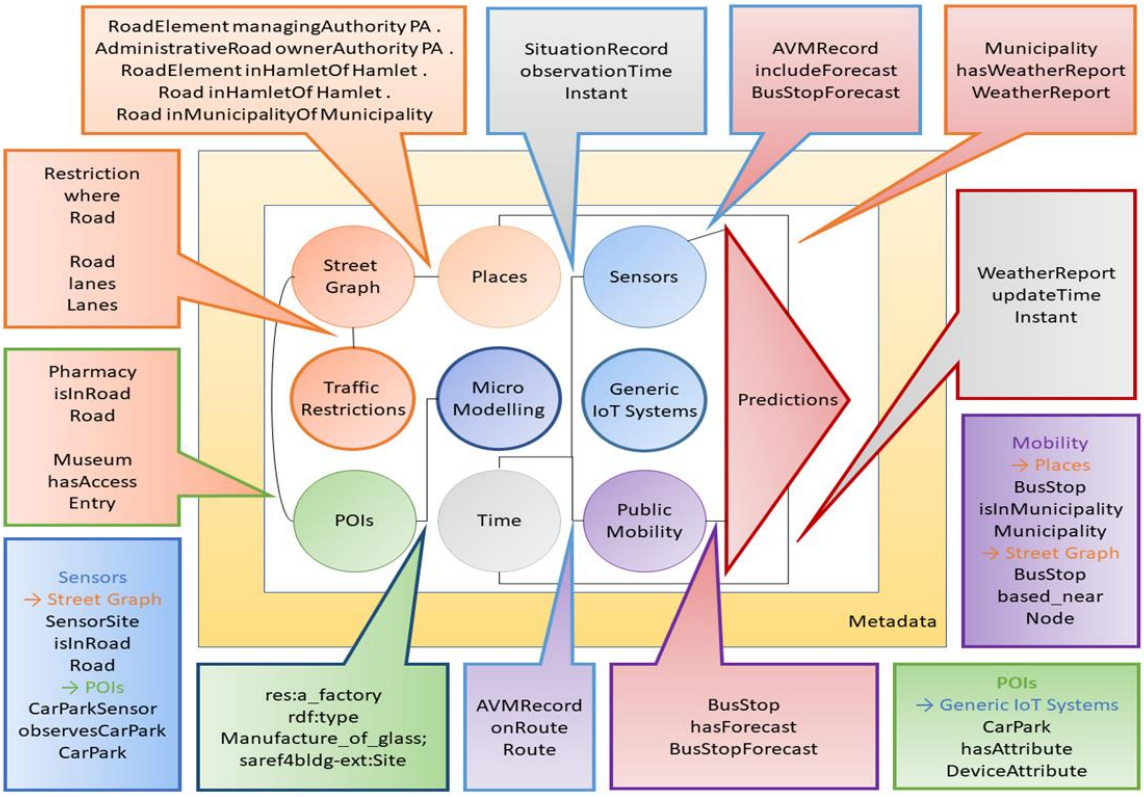
The main tool to access the Knowledge Base (which is implemented into an RDF store) is the ServiceMap https://servicemap.snap4city.org/ which is presenting the KB as an interactive map in which the entities are placed, and its **LOG.DISIT.ORG** which is an RDF browser: https://log.disit.org/service?graph=e7fe1456503401b7bdd69fd3988143ea and thus it represents the KB structure as a graph database.

The mixture of theory, data analysis, and work on-the-field that has characterized the last 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, and 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, and activities, which may be useful to the users 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, and 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 have allowed to supersede the local dimension of the original modelling.



- **Sensors**: concepts for the modelling of specific types of sensors is part of the ontology since the early times, for traffic and car park occupancy. All devices are attached here.
- **Time (formerly, Temporal)**: concepts and relations wrapped that enable the representation of time intervals and instants of time to be associated with 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, license 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 are 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.

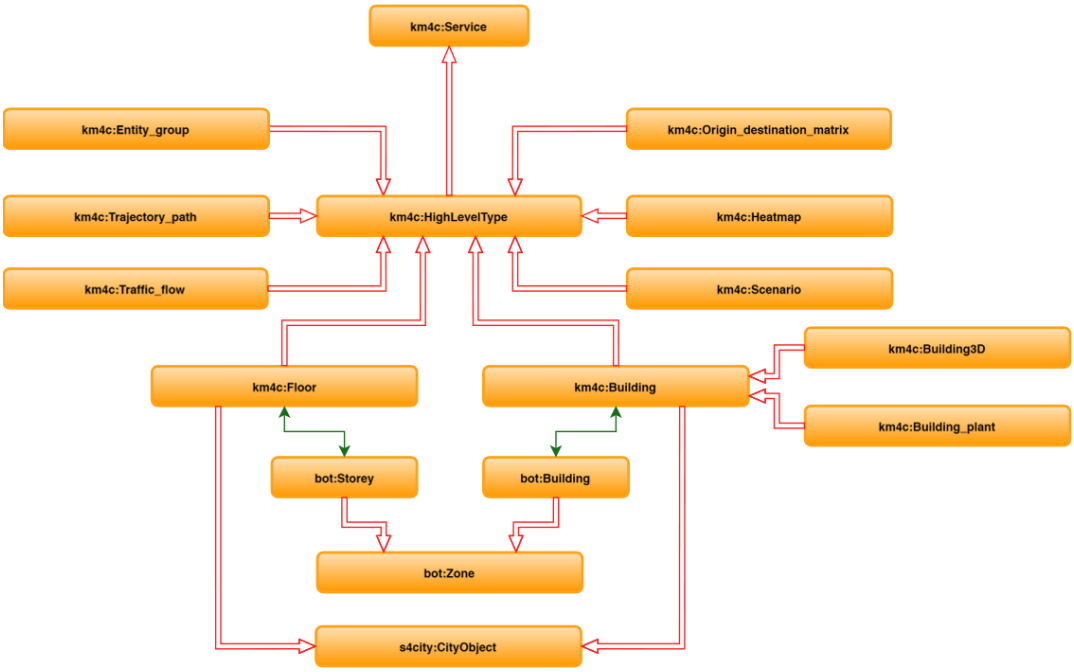
km4city Extensions of 1.6.8 with High Level Types

A novel class, named km4c:HighLevelType (HLT), has been introduced in KM4City version 1.6.8 (https://www.disit.org/km4city/schema/). HLT has been introduced to broadly define any physical or conceptual entity relevant for the Smart City domain, like for example city element (buildings, floors, etc.) or fundamental smart city data like origin-destination matrices (ODM), heatmaps, scenarios, traffic flows, etc. Please note that smart city data such as mobility and transport data, energy data, tourism data are in the focus of the main actions of the European Commission on Data Spaces [EU1, EU2]. HLTs are exploited to quickly retrieve associated entities and to associate them with adequate visualization functionalities. Being KM4City enforced into the Snap4City platform, such characteristics allow to link the semantic representation of the data with their visualization enabling improved automatization in the creation of dashboard for data visual analytics. Note also that each

entity can be put in relation both to an HLT and to a Service class. Thus, exploiting the already defined Service category hierarchy together with the HLTs is possible to associate an entity with a kind description and with its usage destination. For example, a hotel is related to the km4c:Building HLT and to the km4c:Hotel subclass of the km4c:Accommodation, while a hospital is still associated with the km4c:Building HLT and to the km4c:Public_hospital subclass of the km4c:HealtCare. At the same time, is it possible to have more entities with the same category and different HLT. Similar relations can be instantiated with all the HLTs, e.g., to describe the data mapped in a heatmap of the characteristics of the flows represented in an origin-destination matrix.

HLT is a broad concept that can be used to describe different kinds of entities relevant in the smart city context. It can be seen as a sort of label to add semantic information to data of different kinds. To indicate the specific semantic category a series of subclasses have been defined. Currently KM4City HLT includes km4c:Trajectory_path, km4c:Building (with subclasses km4c:Building3D and km4c:Building_plant), km4c:Entity_group, km4c:Floor, km4c:Heatmap, km4c:Origin_destination_matrix, km4c:Scenario, km4c:Traffic_flow. They correspond to classes: **IOT Device/entity**, Heatmap, Traffic flow, Origin Destination Matrix, Trajectory path, **Garden**, Building plant, Building 3D, Floor, **Cycling Path**, Entity Group, **Typical time trend**, Scenario, **Vector Field**, **Parking slot**.

When deemed useful, HLT subclasses have been linked by subclass or equivalence relations to classes of other ontologies to improve general interoperability. km4c:Building and km4c:Floor have been subclassed by the s4city:CityObject [s4city]. Moreover, km4c:Floor and km4c:Building have also been made equivalent to the bot:Storey and bot:Building classes from the Building Topology Ontology [bot], respectively.



Other changes wrt 1.6.7

With respect to version 1.6.7, other main changes are the introduction of two new classes, km4c:IndustrialArea and km4c:Urban_furniture, and the update of the class km4c:Traffic_light.

- km4c:IndustrialArea describes an industrial area composed of multiple sites. Each site is described with the class industry:Site and related to km4c:IndustrialArea through the object property industry:isPartOfIndustrialArea, both defined in the ontology saref4bldg [saref4bldg]. The inverse object property industry:containsSite is also included.
- km4c:Urban_furniture is used to represent any urban furniture like benches, protective elements, accessibility aids, shelters, etc. km4c:Urban_furniture is defined as subclass of km4c:TransferServiceAndRenting and s4city:CityObject.
- km4c:Traffic_light has been modified. The class is still used to represent traffic lights, yet it has been made subclass of s4city:CityObject.

[EU1] European Commission, "Common European Data Spaces." Available at https://digital-strategy.ec.europa.eu/en/policies/data-spaces
[EU2] European Commission, "Data Spaces." Available at https://interoperable-europe.ec.europa.eu/collection/semic-support-centre/data-spaces
[s4city] s4city: SAREF extension for Smart City. Available at https://saref.etsi.org/saref4city/v1.1.2/
[bot] Building Topology Ontology. Available at https://w3c-lbd-cg.github.io/bot/
[saref4bldg] saref4bldg : SAREF extension for building. Available at https://saref.etsi.org/saref4bldg/v1.1.2/

The **Km4City ontology 1.6.8** reuses **standard vocabularies most of the 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;
- **Good Relation:** entities relationships
- ***iot-lite***, aimed at lightweight modelling of the Internet-of-Things;
- ***OTN***: Ontology of Transportation Networks
- ***OWL-Time:*** time reasoning
- **SAREF and *saref4bldg* Smart Appliances REFerence extension for building devices available at** https://saref.etsi.org/saref4bldg/
- **S4CITY:** SAREF extension for Smart City. https://saref.etsi.org/saref4city/v1.1.2/
- ***schema.org***, for a description of people and organizations;
- ***SSN***, semantic sensor network, a W3C ontology for the modelling of IoT Systems;
- ***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 is closer than a given distance concerning a specific GPS position, prior the building of a specific index;
- ***gtfs***, General Transit Feed Specification, dedicated to the modelling of public mobility data, and Transmodel, for public transport infrastructures: lines/rides time schedules, real-time records, paths, etc.;
- **BOT:** Building Topology Ontology. https://w3c-lbd-cg.github.io/bot/

New vocabularies have been added on 1.6.8 version since September 2024.

- ***BOT*:** Building Topology Ontology. https://w3c-lbd-cg.github.io/bot/
- ***S4CITY*:** SAREF extension for Smart City. https://saref.etsi.org/saref4city/v1.1.2/

Other Vocabularies can be added to Km4City Ontology as well.

Then the view of the Snap4City **Data Inspector (see new version of 2025 which use Open Search as engine for faceted searches)** (see section 3.4.2) which is the major tool for understanding not only the **Digital Twin** information but also the **relationships** of Entities / Data with **processes** and **users** which are the three aspects of any Smart engine for a smart city and industry.

3.4.2 - Snap4City Data Inspector: navigating on data, users and processes via their relationships

In the management of a Smart Solutions, one of the main problems is to keep a trace of the complex network of processes, data, causes and effects from the entities / data and their usage in Data Analytics, Proc.Logic / IoT Apps, Dashboards, and developers who created them, etc. This is even more complex when multiple developers and organizations work on the same environment, data and framework. In those cases, the same data can be exploited by multiple users, processes, and dashboards, and thus a failure on it can impact multiple applications and vice versa. For this reason, an extension of the Km4City ontology has been provided to model and keep under control all the relationships among processes, data, users, etc., in all the applications of the framework.

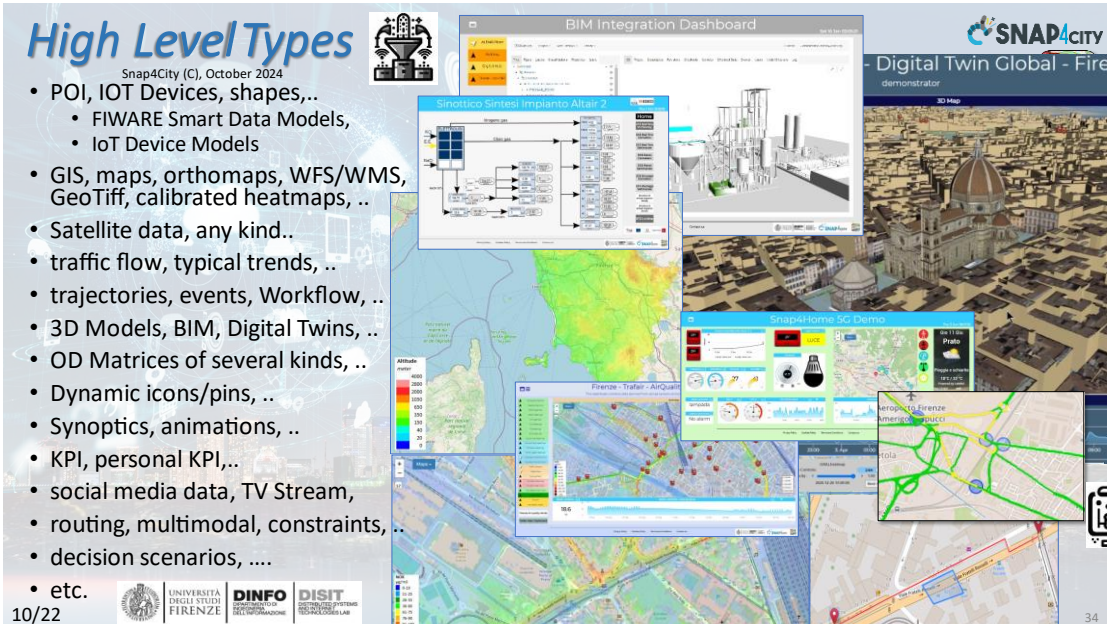
In this regard, **the Entity/Data Inspector, in its former and new version**

- is the tool to keep under control most of the Knowledge about Smart Applications and solutions, from the front end to the back end, via data and HLT.
- provides a table for filtering entities / data which can be observed mainly for **High-Level Types**, HLT. All the elements provide a semantic classification in terms of Nature and SubNature that can be used for thematic retrieval of data and entities.

3.4.2.a - Snap4City: High-Level Types, HLT

Snap4City has the concept of **High-Level Type**, that are unified with a **Unified Data Model**. The list of HLT is the collection of all data kinds of data formats supported by Snap4City system and are listed from the **Data Inspector**, and in the **Dashboard Wizard.**

When in Snap4City is talking about Modelling is referring to Defining and Modelling Entity Models. **Multiple data models can be used for each HLT. A large number of other data types are also managed as follows.**



HLTs are grouped by categories as follows, and most of them have a counterpart of data or metadata described as Entities to have uniform handles to manage them:

- Devices/**Entities a**nd their model and variables (Managed from **Entity/IoT Directory**):
  - **Entity Model (IoT Device Model),** (managed by Entity/IoT Directory)
    - Entity Models can be derived from FIWARE Smart Data Models
  - **Entity Instance (IoT Device), Entity Variable** (IoT Device Variable/Metric) (managed by Entity/IoT Directory)
    - **Entity Instances can be also HLT as described in the following**
  - **Mobile Entity Model (Mobile Device Model), Mobile Entity Instance (Mobile Device)**, **Mobile Entity Variable** (**Mobile Device Variable)** (managed by IoT Directory) The difference with the previous is the possibility for them to move (change GPS) and thus render them on trajectories, track for fleets and sensors, let's say time series moving over time.
    - Mobile Entity Models can be derived from FIWARE Smart Data Models
  - **BIM Entities** are actually Entity Instances (managed by Snap4City **BIM Manager** accessible into the **Dashboard Wizard**), that refer to some BIM View, and the BIM manager may refer to some BIM Views in which the Entity Instance.
  - **Files Entities** are actually Entity Instances (managed by the Snap4City **File Manager** from **Data Management**).

- **Typical Time Trends, TTT:** they are **Entities** managed by the Snap4City **TTT Manager on Data Management**, on which a series of data (typically one for each hour) is modelling the typical time trend for the H24 of a given variable. For example, for: traffic flow, temperature, traffic density, humidity, emissions of CO2, number of people, payments, costs of parking, etc. TTT are de facto Entity Instances of HLT.
- **TV Cam** (managed by **TV Cam Manager** from **Data Management**), they are a specialized Entity Model and Entity Instances. So that, they have GSP location, nature Subnature, variable, etc., plus credentials to access at the protected video stream. They get video streams on RTSP, ONVIF modelled as Entities / Devices. They can be defined and managed by the **TV Cam manager,** in which it is possible to define protocol and access data. The visualization is performed on **WebRTC protocol** via **Kurento on a special Widget**. TV Cam are de facto Entity Instances.
- **KPI multiple variables, POI, Buildings, Buildings floors, parking lots, Vector Fields, heatmaps, traffic flows, ODM, scenarios, etc. ARE ALL defined as Entities as well.**
- ***Data Table Device, Data Table Model, Data Table Variable*** *(managed by IoT Directory and produce by Data Table tool). They are produced by the* ***Data Table Loader or POI Loader*** *automated processes for loading data and generating their models and devices, which automatically produce the model, devices, and instance messages over time, let say time series. (This tool is not accessible in all Snap4City platforms, optional in MicroX, deprecated)*
- ***Sensor, Sensor Device, Sensor-Actuator*** *(deprecated since all the IoT Devices can be both sensors and actuators). They are produced by formers ETL processes, similar to the concept of device and variable (Sensor Device, Sensor). In the present version of Snap4City, all sensors are actually modelled as sensors-actuators.*

- **KPI, key performance indicator** single variables (Managed from **Data Management, HLT**):
  - **MyKPI** (managed by MyKPI manager from **Data Management**) they are all personal data produced by IoT App, Mobile App, or loaded from some process. MyKPIs are mainly used for trajectories, for saving single values time series. They can be: (i) used for long-term saved and retrieved from Proc.Logic / IoT App, (ii) saved on MySQL or Open Search according to a setting, (iii) used as event-driven for synoptics, (iv) created via the MyKPI management via the graphic user interface. (in most cases they can be modelled as **Entity Instances**)
  - **My Personal Data**, **MyData** (the latter are mainly deprecated since can be modelled as Entities). The former is used to save user profile information such as password of hash, scenarios, etc. (in most cases they can be modelled as **Entity Instances**)
  - ***My POI*** *(mainly managed as Entity Models, mainly* ***deprecated****). They are static point of interest that can born private and then made public when approved. POIs are used for marking Points of Interest on maps (museums, banks, schools, parks, etc.), they are points of attractions without time changing variables, without time series and with a relevant number of metadata: web page, email, time of opening, etc. They could be also generated as Devices, see for instance POI Loader. MyPOI are personal POI, which can be created from a Mobile App. The administrator can leverage a MyPOI to a general POI. (in most cases they can be modelled as* ***Entity Instances****)*
  - ***KPI/Direct Metrics****: deprecated since can be modelled as MyKPI and Entities*

- **Map features and tools** in addition to all the above data, most of them and other can be located on map with their dedicated graphic representation and tool (Managed from **Data Management, HLT**), also:
  - **Heatmap** (managed by **Heatmap Manager** from **Data Management, also saved as Entity Instances in some cases**) they are produced by some data analytics or directly loaded on the Heatmap Server. They can be shown on a map, and picket if loaded correspondingly. Some of the heatmaps can be computed on the client (non-calibrated), the others are generated to produce dense and regular maps calibrated according to a **Color Map** standard (see **Color Map Manager**). The latter is distributed by the GeoServer via WMS/WFS protocol and can be animated in 24 hours if instances

over time are present. Heatmaps are actually **time series** of matrices. For example, the Heatmap of CO2 of every day for the whole year.

- **Traffic Flows** (managed by **Traffic Flows Manager** from **Data Management, also saved as Entity Instances in some cases**) they are typically produced from a traffic flow reconstruction algorithm to generate data in terms of Polyline JSON and loaded on Traffic Flow Manager which converts them into geo images for the GeoServer which distributes them via WFS/WMS. They can be **time series** of data to be used for showing animations daily of the traffic flow on the map.
  - They can be (i) generated from data provided, for instance provided by Here, or (ii) generated by SUMO simulation.
  - Traffic Flows can be saved on the platform in different formats: JSON files, GeoTIFF on the GeoServers and recently on OpenSearch as data about the traffic density for road segment.
  - **Traffic Flow Polyline:** they are produced from traffic flow reconstruction data in terms of Polyline JSON and can be used for rendering on 3D City Digital Twin as crests, on a 3D map. They can be time series of data.
- **OD Matrix** (managed by **OD Matrix Manager** from **Data Management, also saved as Entity Instances in some cases**). They are produced from some original data, census data, cell phone, data, mobile data, PAX Counters plus Snap4City algorithms, etc., and loaded in terms of ODM data on the OD Manager/server for distribution on map and Dashboards. They can refer to different geo areas, grids, ace, regional administrative shapes, MGRS, custom shapes, etc. They can be **time series** of data, inflow and outflows, etc., changing over time, inflow and outflows. They can be acquired from data provider such as telecom operators: TIM, Vodafone in Italy.
- **WFS** (**WMS**, etc.), for connection to some GIS for images, maps, orthomaps, shapes, POI, PIN, etc. They can be taken from some QGIS, GeoServer, ArcGIS, etc. They can be acquired from IoT App/Proc.Logic as well as taken and visualized on MultiDataMap Widget on Dashboard without storing them on platform.
- **3D models of the City, Global/Local Digital Twins as single Buildings, floors, etc. https://digitaltwin.snap4city.org/ to compose the whole city, as well as global 3D shapes** generated by**:**
  - a suite of Snap4City tools: starting from Lidar for the high of building, GIS shapes of plant, facades images, orthomaps images, etc.
  - taken from global services as Google 3D tiles
  - taken from IFC format of building, produced by DWG Autodesk.
- **BIM Entities** are actually Entity Instances (managed by Snap4City **BIM Manager** accessible into the **Dashboard Wizard**), that refer to some BIM View, and the BIM manager may refer to some BIM Views in which the Entity Instance.
  - **Can be Floors, plants, etc.**
- **Trajectories**, **Paths**, they are produced from some original data or some data analytics, or **routing**, or **paths**. They can be sent to the map for visualization. Some segments of the Trajectory can be requested to the composed by a new routing. (**also saved as Entity Instances in some cases**)
  - **GTFS:** paths for busses and public transport services **Public Transportation paths and timelines, …**
  - **Routing** paths of several kinds, travel means
  - **Paths** of moving devices, as fleet, people moving, etc.
- **All the Entities (IoT, WoT, devices, etc.) may have shapes (WKT and GeoJSON) as gardens**, **Building's shape plant, Buildings floors, parking lots, admin areas, etc.** They can be sent to the map for visualization with some specific colour for their filling.
- **Vector Fields,** are a grid of vectors representing flows, slops, wind, deep, waves, etc. in terms of magnitude and direction, also saved as Entity Instances. They can be sent to the map for visualization.

- **Scenarios are city areas with:** road graph for vehicles, public and reserved roads, pedestrian paths, tramways, cycling paths, etc., also saved as Entity Instances. They can be sent to the map for visualization. **Two kinds of scenarios are possible on Snap4City Multi Data map:**
  - simple for blocking area, and
  - complex to work on road graph and services included.
- **Tools for the Map**: what-if, scenarios, scenario editor, comparing, measuring, etc. (as tools listed into the **Dashboard Wizard**). They can be recalled from the selector to activate them on demand, and once activated are shown on the Multi Data Map on views and dashboards. They are
  - **Scenario Editor (put *scenary* on the selector):** a complex scenario editor to select a segment of the road graph, editing a large range of details, saving its version: INIT, ACC, etc., to be used for sending the scenario to some other data analytics such as: heatmaps generator, traffic flow generator, etc.
  - **Scenario (put */scenario/* on the selector):** for creating simple scenarios blocking a part of the city, multiple unconnected areas and for multiple time slots. Scenarios can be saved and shared with other operators.
  - **What-if with Routing tool (put */whatif/* on the selector):** for exploiting a scenario and deciding routing and solutions to be taken for private routing. The resulting cases can be saved to be shared with other operators.
  - **Comparing orthomaps with MultiDataMap widget on Dashboards having placed any add on entity on top (put */sideVision/* on the selector)**
  - **Measuring elements on Orthomaps: as a CSBL tool to be activated on dashboard.**
  - **Others are simple viewers that are activated automatically on the basis of the HLT visualized as: heatmap, ODM, traffic flow,**
  - **Other as CSBL tools to be activated on dashboard External Content widgets (and in some cases a counterpart also on MultiDataMap widget): TLP editor, GTFS viewer/editor (in progress), etc.**
- **Synoptics** (managed by the Synoptics Manager from **Extra Dashboard Widgets**) they are special modules provided by Snap4City grounded on SVG format for rendering in real time graphics layouts of energy plat, industrial plants, metro stations, etc. They may present several IoT Device Variable, Constants and MyKPI in event-driven data. They can also provide interactive elements by which the user can send data, and actions to the platform, writing to MyKPI, using the IoT App. Synoptics are instantiated based on a model which can be generated according to guidelines.
  - **SVG synoptics** can be generated from some graphic tool, with the possibility of changing colors and shape of the elements according to some variable, in real time.
  - **SVG of building floors** can be generated from DWG, DXF of Autocad, converting them in GeoJSON and then from GeoJSON with a Snap4City tool converting them in actionable SVG, with the possibility of changing colors and shape of the elements according to some variable, in real time.
- Etc.

- **Third party html pages and tools, only for Dashboard Widgets**
  - **External Service** (managed by the **External Service Manager**) they are websites which registered on Snap4City.org and can be rendered into an Iframe of the Dashboard if they allow to do it. Some restrictions may be applied by the owner of the Web Sites.
  - **MicroApplications** (managed by the MicroApplications Manager from **Extra Dashboard Widgets**) they are Web Applications developed by Snap4City as micro solutions, micro applications solving specific cases and rendered into an Iframe of the Dashboard.
  - **BIM View** for single building and areas (managed by BIM manager from **Data Management**)
    - **BIM View:** are used to represent and show a 3D building, floor or segment derived from some IFC format for BIM. They are collected and provided by the **BIM Manager**.
    - **BIM Device:** are IoT Devices used to represent and show a 3D building, floor or segment with a specific IoT Device, ServiceURI, POI attached. They are derived from some IFC format

for BIM, collected and provided by the BIM Manager, while the BIM editor permits to attach the ServiceURI to the BIM corresponding element.

- **Other Widgets:**
  - **Dashboard-IOT App: the formalization of the SSBL:** representing the data flow, even driven from Proc.Logic / IoT App to Dashboards and vice versa. They can be intercepted and exploited.
  - **A large number of new Widgets for Dashboards are reported on training slides for dashboards and views:**
    - **https://www.snap4city.org/download/video/course/p2/**
- ***Other Deprecated Widgets which now are accessible or can be remade with CSBL/SVG:***
  - ***Special Widget:*** *several special widgets have been created to show specific HLT, for example the* ***weather forecast, civil protection, quality of public transport, status of a specific decision support system process, first aid, clock, Twitter Citations, and Twitter Hashtags.***
  - ***Complex Event:*** *to represent events of different kinds on specific Dashboard Widgets: entertainment events, critical events, operator events, traffic accidents, etc. In the short future, a new version of the* ***Event Device/Device Table*** *and Event Table Widget will be finalized to deprecate this former HLT with a set of more flexible solutions.*

3.4.2.b - Snap4City: Smart Solutions, Smart Applications, Visual Analytics, Business Intelligence tools

The smart solutions, Smart Applications, Visual Analytics, Business Intelligence tools consist of

- **Users:** owners (who register), responsible, providers, developers, etc.
- **Dashboards and their widgets** which provides interactive tools to users, which allows to activate processes on clients and servers, and even AI.
  - The widgets can be orchestrated each other to make drill down/up, slicing, faceting, searching, filtering, etc. https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf
  - The widgets can also refer to Server-Side Business Logic on IoT Apps.
- **Processes/API**: which can be: aperiodic, periodic, event-driven, …. managed by different tools (Brokers, ServiceMap, Heatmap Manager, Web Scrapers, Proc.Logic/IoT Apps, Traffic Flow Manager, AI, LLM, etc.), on servers, on clusters HPC, on MLOps ClearML, as static and/or dynamic processes:
  - **Data Ingestion**: the models and mechanism for data gathering**: IoT App / Node-RED, Proc.Logic,** directly via Broker, Fast Data Loader Python processes, etc.
    - While are deprecated the Data Table Loder, POI Loader.
  - **Data Transformation:** IoT App / Node-RED, Proc.Logic, integrations, ..
  - **Data Storage,** indexing, retrieval, drill down/up, slicing, etc.
  - **Data Analytics:** predictions, anomaly detection, computing, etc., using AI, ML, LLM, etc.
  - **Business Logic behind Dashboards, server- and client-side business logic.**
  - **Data rendering, Dashboarding, interacting.**
  - **Etc.**
- **Entities / Data (and their Digital Twin information)**: data are heterogeneous, large volumes, 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, described above.**
- **Relationships**: among data, processes, users and mixed.

To understand how to develop smart solutions, Smart Applications, Visual Analytics, Business Intelligence tools please read:

- Development Life Cycle: https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf

- **Client-Side Business**
  Logic Widget Manual: https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf

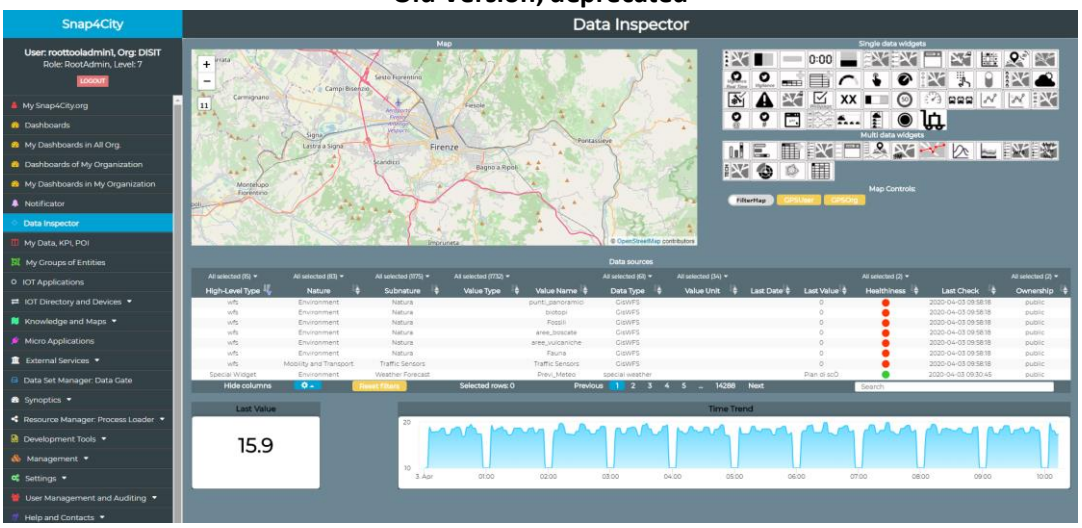
3.4.2.c - Snap4City: Entity / Data Inspector, Digital Twin

The Snap4City Entity / Data Inspector is the primary point to understand the issues and produce reports on those facts. For example, at the level of users, devices, process, dashboard and application. See this document on the early Version [**DI-DataInspectorBase**].

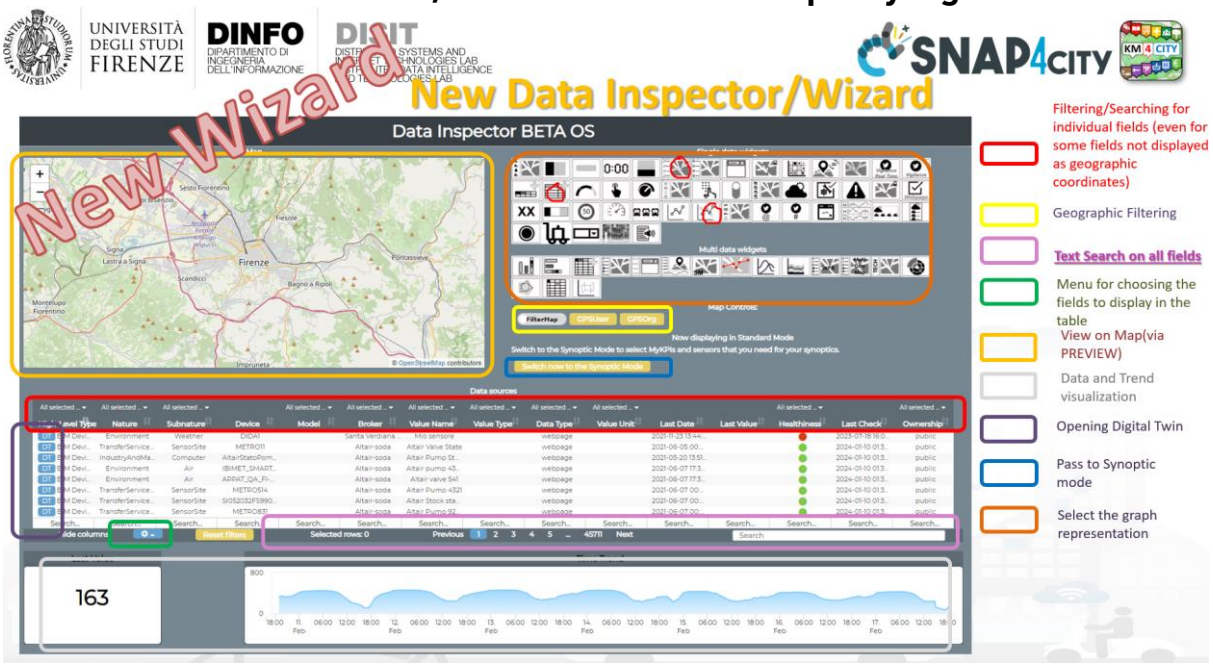
Data Inspector is showing data:

- **Retrieved by any combination of parameters: nature, Subnature, type, model, etc.**
- **Listed by High-Level Types,** and classified as in the Dashboard WIZARD: see Tutorial Part 2 for details: https://www.snap4city.org/download/video/course/p2/
- **Organization** only, that are public within your organization, and your private, and those that have been delegated to you:

Old Version, deprecated



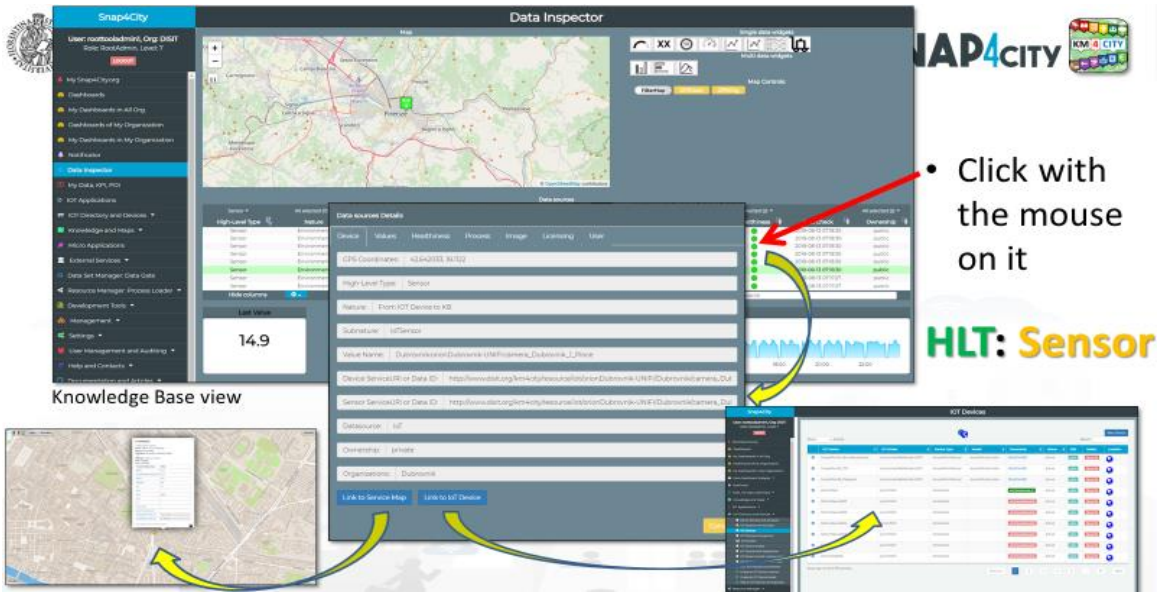
New version, active on MicroX and Snap4City.org



On Data Inspector, as well as on Wizard, you can perform (see https://www.snap4city.org/download/video/course/p5/ **)**

- **Cross Filtering based on**:
  - **MAP**: pan and zoom → lock, center on GPS coordinates of the user
  - **Data Source Classification**: faceted filtering, full-text search, etc.
  - **Click on data source** to see it on the map, and see the graphical representation, just to learn how to "widgetting" it.
  - **Full-Text Search** if you remember some description… for each field
- **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)**

In the former versions of the DI, to open the data inspection of a given Entity you have to click on the microbutton red/green marking the healthiness.
In the new version a blue button is provided on the left of each data line in the faceted search table.



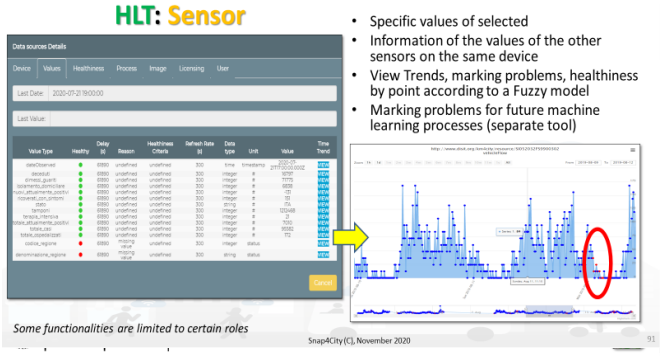
By clicking on the healthiness status micro button (for example the green one on the above figure), the Data Inspector will provide you with different data according to the HLT.

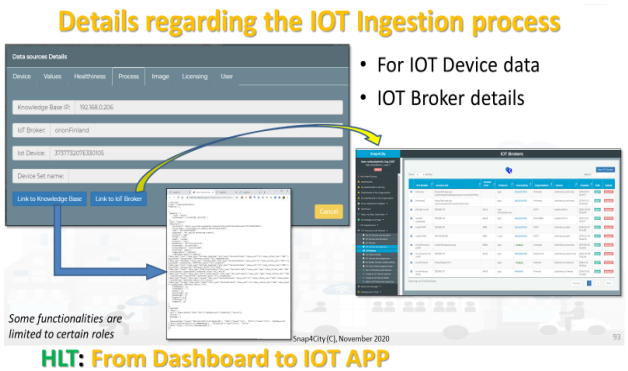
From the Data Inspector, you can

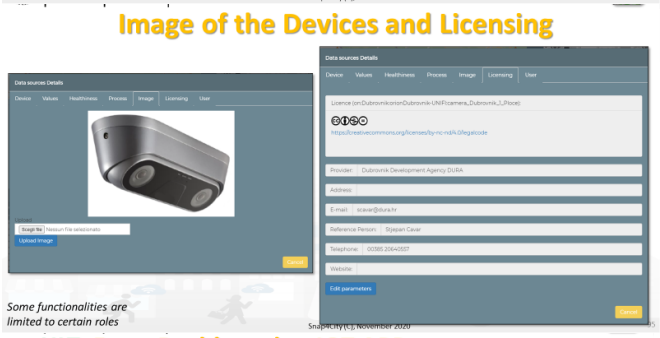
- order your report and get them. reports on devices can be monthly or trimonthly.
- see the data of dashboards and the relationships among Dashboards, IoT App, Data, Data Analytics, etc.

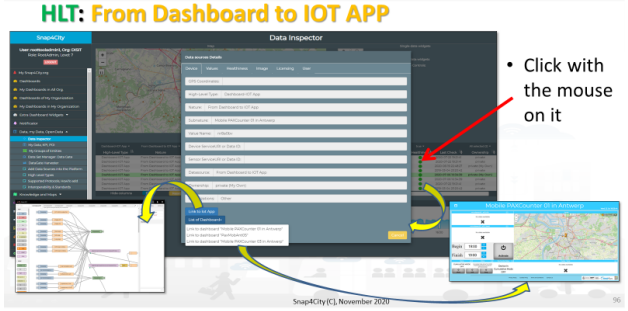
For example, for IoT Devices, MyKPI, etc.: Device data, values, healthiness, process, image, licensing, user info and Reports produced or request form, plus Links to ServiceMap, IoT Directory, IoT App using it, KB graph, IoT broker.

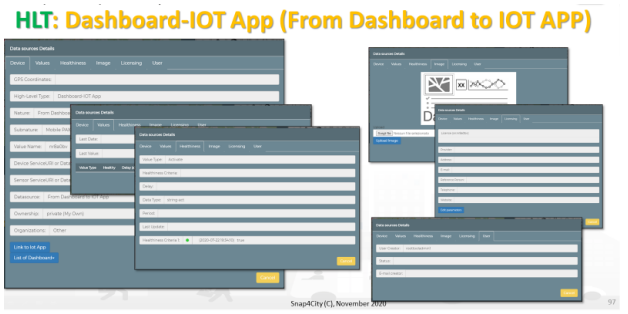
The relationships of data vs dashboards vs widgets can be also browsed starting from the Dashboards structure which is accessible for the Dashboard owner as described in the following.

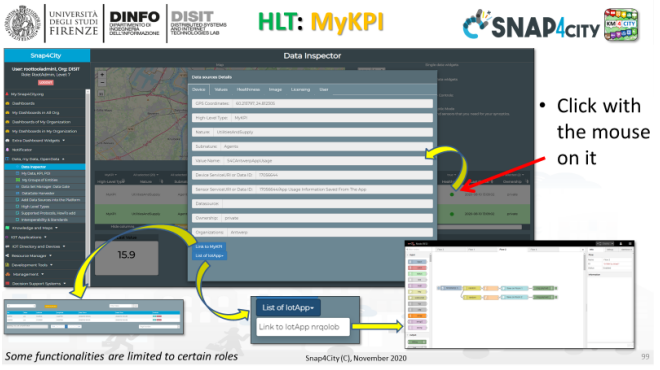


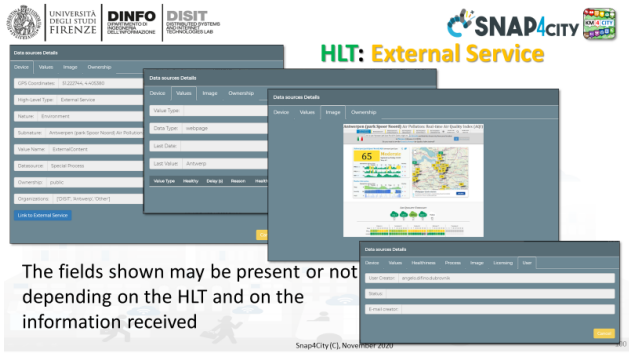












3.4.3 - Snap4City High Level Type vs Storage and distribution

The Storage of Snap4City is organized as follows:

High Level Type data	Ingestion kind: MicroServices, API	Storage	Distribution tool API
Entity / IoT Devices (time series)	Proc.Logic / IoT App, special Python, Brokers, API	OpenSearch, KB	Smart City API, Orion Broker
MyKPI (time series)	Proc.Logic / IoT App, API	OpenSearch (or SQL)	Smart City API
POI	Proc.Logic / IoT App, API	OpenSearch, KB	Smart City API
Heatmaps (time series)	Proc.Logic / IoT App, Python, API	GeoServer	API for WFS, WMS
Maps	API and user interface	GeoServer	API for WFS, WMS

GIS data	API and user interface	GeoServer	API for WFS, WMS
Traffic Flow (time series)	Proc.Logic / IoT App, Rstudio, Python, API	GeoServer, Traffic Flow Manager, KB	Traffic Flow Manager, API for WFS, WMS
Traffic Flow HTL (time series)	Rstudio, Python, + Broker, API	Traffic Flow Manager, KB	Traffic Flow Manager, API for flow segments
OD Matrices (time series) (HLT)	Proc.Logic / IoT App, Python, API	GeoServer, OD Manager	OD Manager, API for WFS, WMS
Scenario Advanced	Proc.Logic / IoT App, API	OpenSearch and Big data space	Smart City API
Simple Scenario	MyPersonal Data	SQL	
TV CAM	Registering and IoT Device, Broker, API	OpenSearch, KB	Smart City API
BIM view (HLT)	BIM Server, IFC, (Broker), API	BIM Server, KB BIM Manager	BIM Manager
3D City Model (HLT)	Python, (broker) API	GeoServer, OpenSearch, KB	Smart City API, API for WFS, WMS; HTTPS
Buildings, floor, parking, etc.	Proc.Logic / IoT App, API	OpenSearch, KB, Files	Smart City API
File, TTT	Proc.Logic / IoT App, API	OpenSearch, KB, Files	Smart City API
Trajectories	Proc.Logic / IoT App, Python, API	OpenSearch, KB	Smart City API
Etc.			

The general Snap4City storage is an entry gate for Open Search is based on a **NIFI Cluster** for massive data ingestion, aggregation and enrichment (collecting data from any **Broker** where it is automatically registered by the Entity / Directory), and saving the data to (i) **Open Search** cluster (Open search of AWS) for storing and indexing data, and (ii) the **Knowledge Base, KB,** (based on **Km4City ontology**) implemented as an RDF store (Virtuoso) which is an index for geospatial, relational, and temporal aspects, which can be federated. Whenever a new data model is registered in the system, the registration is performed into the **KB**. Different instances of the KB can be federated via Smart City API by creating a mutual connection among cities/areas of the network, if the single installation wants to share, and deciding what to share.

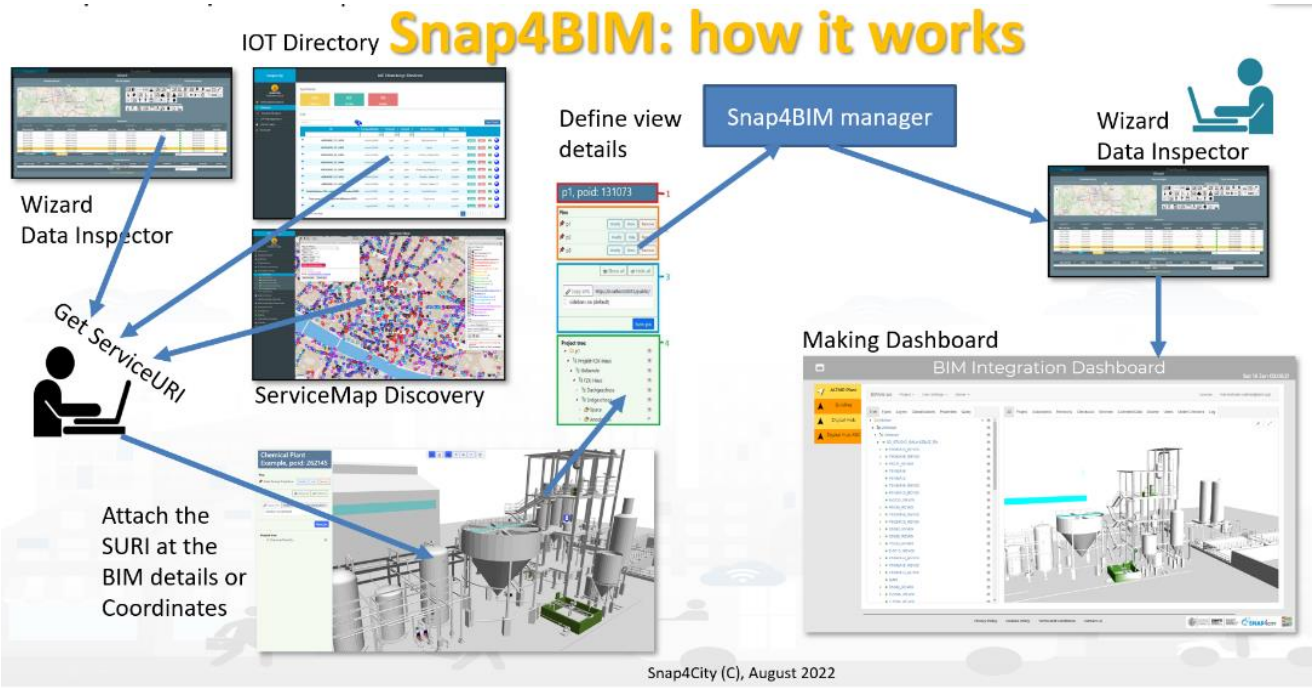
In Snap4City file-based data are managed as well:

- **Heatmaps, Orthomaps, Traffic Flows Maps, 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;
- **Origin Destination Matrices** are managed by the **OD Manager**;
- **Internal Buildings 3D Shapes**, and Floors are managed and distributed by **BIM Server**. Standard BIM tools are used for editing and interchange in IFC formats with standard tools such as AutoDesk Revit, etc. Buildings and Floors are shown in dashboards for their integration with maps and time trends of IoT devices;
- **3D City Representation** can be loaded according to different components from which they are composed such as: image patterns, building shapes, 3D shapes, LIDAR data, traffic flows, POI, IoT Devices, etc. [**DigitalTwin1**], [**DigitalTwin2**], [**GeneratingDigitalTwin**] see **Section 3.4.4.**
- **Etc.**

3.4.4 – Digital Twin Local and Global of the City, 3D City Representation

https://digitaltwin.snap4city.org/

Digital Twins of Smart Cities are fundamental tools for decision makers since they can provide interactive 3D visualizations of the city enriched with real-time information and connected to actual complete digital model of the entities with all their heterogeneous data/info. Such a technology can be exploited to observe the status of the city, and to perform analysis and simulations, and thus to develop strategies. Snap4City provides an open-source framework for creating enhanced Global Digital Twin integrated with Local Digital Twins. The Snap4City Digital Twin is able to manage multiple kind of entities, from 3D representation of the urban infrastructures to information about services, traffic, air quality, parking, environmental data, IoT data, govern data, KPI, SUMI, PUMS, planning, etc., and others.

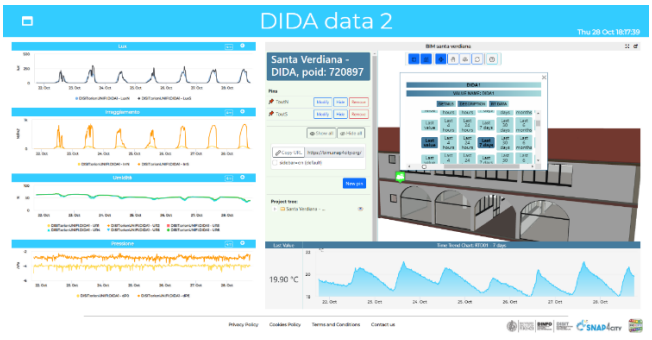


For Digital Twin / CitiVerse we intend the modelling and management of physical entities with digital entities, by monitoring, simulation and control. In the context of cities and industries we can have a digital counterpart describing all aspects including data flow, time series, structural details of the design, control and functional aspects, behaviour, prediction, simulations, and also 3D shapes to see them in the 3D representation in digital https://www.snap4city.org/749 . The Digital Twin local refers to local details of a building or of a single machine/area, while the Global refers to the coverage of the model among several entities. Data, and interaction can be on physical entities and provoke changes in the digital or vice-versa, the two worlds should be actually twined, or merged, or better they should be the same thing, in some cases the digital model can be used to explore other cases, for example in simulation, optimisation, etc. Snap4City's comprehensive suite of tools and technologies offers a robust foundation for creating smart city solutions that are compatible with CitiVerse, adhere to MIMs Plus standards, and utilize the DS4SSCC blueprint and the LDTs Toolbox. This integration promotes the development of innovative, interoperable, and impactful smart city projects. In Snap4City, **Digital Twins** are integrated with IoT Device Models, maps, heatmaps, structure, functional, services, and 3D representations, etc. They can be used for making changes into the virtual world and see the actions of the physical, or just to apply temporary changes to show them at the stakeholders, thus facilitating the decision-making process in close to realistic conditions. Snap4City integrated **Global and Local Digital Twins** with Entity / IoT Device Models, maps, heatmaps, structure, functional, services, and 3D representations, etc. They can be used for making changes into the virtual world and see the action of the physical, or just to apply temporary

changes to show them at the stakeholders, thus facilitating the decision-making process in close to realistic conditions. The so-called Digital Twin approach implies managing a digital counterpart of the real world.

At level of Digital Twin local detailed aspects are modelled in digital of a physical element, for example of a chemical plant with machines, motors, silos, etc., each of them with 3D representation, detailed components, and real-time values of sensors attached which can be used to understand its behaviour in terms of real-time data and functionalities, mechanical parts to be disassembled for maintenance and inspection, etc. Several different levels of resolutions can be provided, and the rendering is typically performed on 3D modelling and visualization tools such as the BIM Manager (https://www.snap4city.org/730). The 3D shape is obtained by starting from formats such as IFC, DXF, SVG, STEP, IGES, STL (Stereolithography), OBJ (Wavefront), DAE (Collada), SCAD (OpenSCAD), and IV (Inventor). The Local Digital Twin also may include the detailed description of floors, detailed description of data, devices, etc.

Snap4City provides support for creating Global Digital Twin of the city with integrated Local Digital Twins, it is part of the Dashboard Builder of Snap4City which is 100% open source. It is possible to pass from Digital Twin to the Local Digital Twin such as "entering in the detailed view". The 3D representation and view of the Digital Twin in Snap4City can manage multiple layers on the map with different levels of opacity. Snap4City 3D City Modelling Framework for Smart City Digital Twin is covering terrain elevation, roads, building planimetries, maps, orthomaps, heatmaps, buildings, high-value buildings with meshes, building extrude from their plant shape, roof reproduction from LiDAR data, pattern extraction and positioning for roofs and facades, traffic flow data, IoT Sensors/actuators, POI, traffic flows, bus routes, cycling paths, building, scenarios, optimisation, what-if analysis, etc. The LDTs Toolbox provides methodologies and tools aim at creating and managing Living Digital Twins of urban environments. Snap4City's features, such as advanced analytics, simulation, and visualization technologies, complement the LDTs Toolbox by enabling the development of dynamic and interactive digital twins. Snap4City can simulate real-world scenarios, support decision-making processes, and facilitate urban planning and management, all while being integrated with the broader smart city ecosystem. Interoperable Data Spaces: Snap4City's platform is built on open standards and supports the creation of interoperable data spaces for smart cities, as a secure and unified marketplace for data that promotes the development of new products and services.

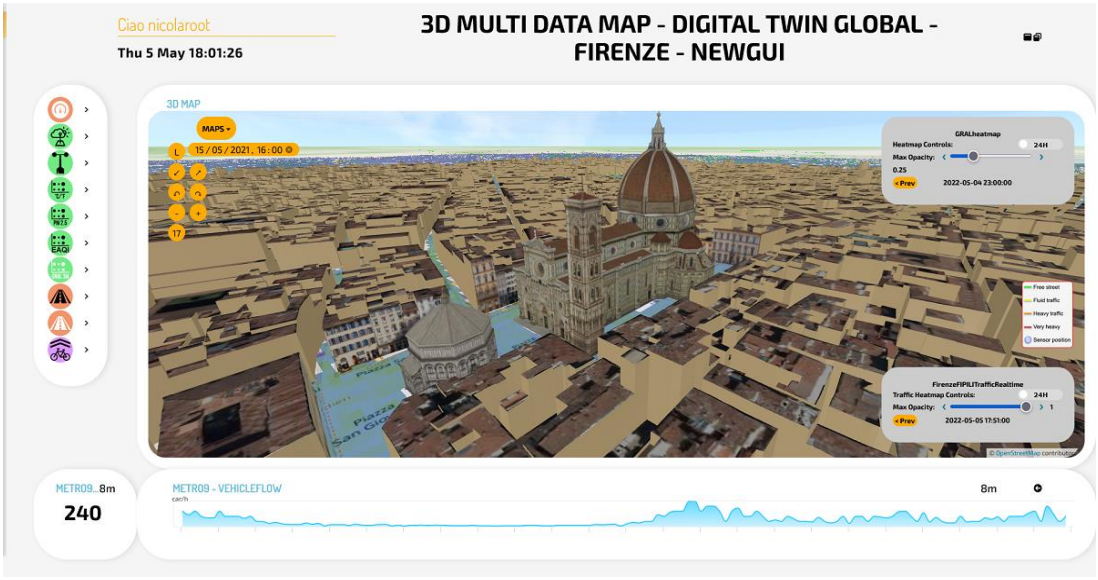


BIM representation in Snap4City provides an application for: Connecting BIM models with ServiceURI data into Snap4City/Industry, so that they can be: Entity / IoT Devices, POI, etc.; Observing PIN (icons representing an entity with its ServiceURI) from any view; Positioning PIN in any place of the 3D space; Accessing to any element of the 3D model and any position with the 3D interactive viewer; Controlling the user interface interaction; Showing the data, and selecting the 3D elements. On the user interface, you can jump from global scale maps to local 3D representations and vice versa.

Global Digital Twin represents the city in its context, indeed, it is the digital counterpart, that includes a 3D representation (https://www.snap4city.org/749), and all associated data and their relationships. Snap4City supports the creation of a Global Digital Twin of the city with integrated Local Digital Twins, it is part of the Dashboard Builder of Snap4City which is 100% open source. It is possible to pass from Digital Twin to the Local Digital Twin such as "entering in the detailed view". The 3D representation and view of the Digital Twin in Snap4City can manage multiple layers on the map with different levels of opacity. Also, heatmaps can be added

in the 3D view since they are shown integrated into the Digital Twin, orthomaps and terrain. POI markers are also integrated into the 3D view with the possibility to see the detailed popup by clicking the corresponding POI marker on the map.

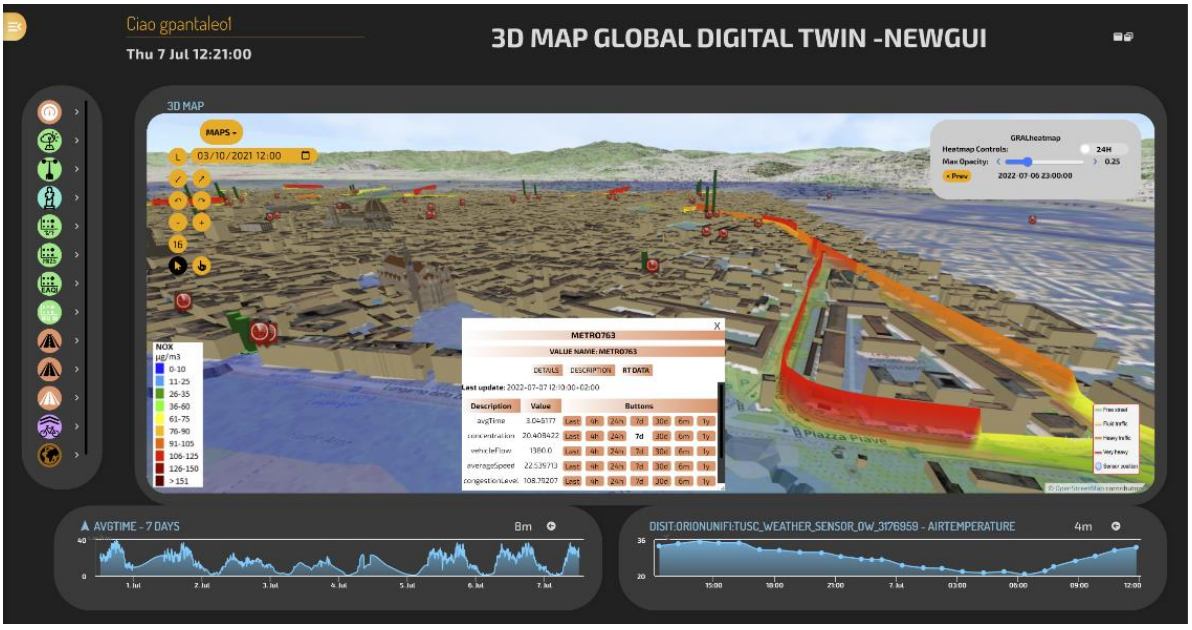
Snap4City 3D City Modelling Framework for Smart City Digital Twin is covering more data concerning those addressed in the LoD classifications and includes **terrain elevation, roads, building planimetries, maps, orthomaps, heatmaps, buildings, high-value buildings with meshes, building extrude from their plant shape, roof reproduction from LiDAR data, pattern extraction and positioning for roofs and facades, traffic flow data, IoT Sensors/actuators, POI, KPI, traffic flows, bus routes, cycling paths**, etc. The following dashboard shows an example of a 3D representation of the City of Florence, with POIs, heatmaps, and geometries that can be activated/deactivated by clicking the corresponding item in the selector widget on the left.



https://www.snap4city.org/dashboardSmartCity/view/Baloon.php?iddasboard=MzQzMg==

The version presented at Smart City Expo World Conference in Barcelona is not publicly accessible. https://www.youtube.com/watch?v=vYeb485nYJ8&t=183s
You can get a demo of the demonstrator by sending a request to snap4city@disit.org

Full access is provided only to contractors and partner in partnerships.



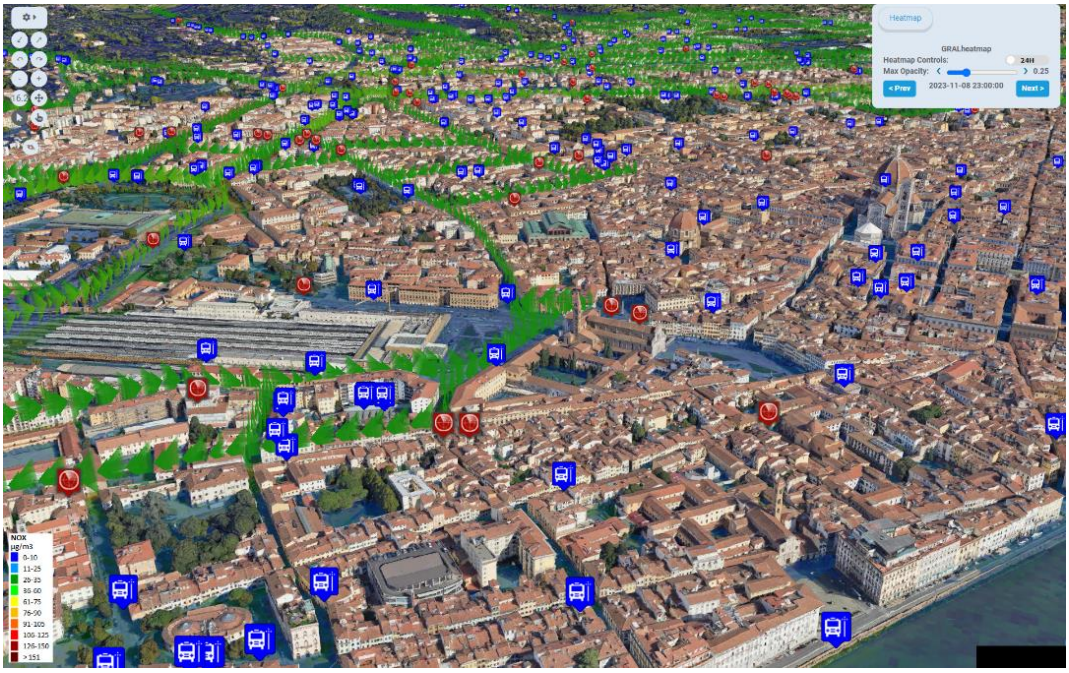
Global Digital Twin for reasoning on impact of changes to the city at low cost

Snap4City provides an efficient and cheap approach for (i) the production process from raw data to 3D DT elements, (ii) the integrated model for DT representation and distribution on the web, (iii) the performance in production and distribution of the resulting integrated DT model. The 3D representation is enriched with: a 3D representation of crests for traffic flows, 3D shapes, and dynamic PINs which can manifest data values for real-time IoT data representation, heatmaps, and animations, picking functionality for building and data elements, and interactivity with all the elements from dashboards [**DigitalTwin1**], [**DigitalTwin2**], [**GeneratingDigitalTwin**].

https://www.snap4city.org/dashboardSmartCity/view/Baloon-Dark.php?iddasboard=MzQzMQ==

Google released photorealistic 3D tiles that can be exploited to represent city buildings. In this case, the Google 3D tiles are loaded using a Tile3DLayer replacing the Terrain, the 3D Buildings, and the Additional 3D elements layers of our architecture (see Figure). All the other entities included into our SCDT can still be represented over the Google 3D tiles effortlessly, thanks both to our layered architecture design which is able to handle any different kinds of data to be displayed, and to the underlying Snap4City Knowledge Base and services that can be queried to retrieve road descriptions, position of POIs, heatmaps, and to perform What-If analysis. As an example, we have reported the usage of the Google 3D tiles into our Snap4City SCDT interface for the cities of Florence and Helsinki. In the first case – Figure – all the elements described in the previous sections (i.e., PINs, 3D Traffic Crests, heatmaps, paths, etc.) can still be represented when using Google tiles, thus demonstrating how the proposed digital twin architecture does not depend on the used 3D building representation. Conversely, in Figure the city of Helsinki is shown in the proposed SCDT framework, together with some PINs indicating the position of POIs, demonstrating how our solution can be effectively deployed in different municipalities. It should be remarked however that the adoption of Google 3D tiles imposes some restrictions, both on usage licenses (that at the time of our writing are not yet well defined by Google) and on the interactivity, since in 3D tiles different buildings are fused in a single mesh. Indeed, in this case, building substitution is no longer possible without elaborating Google tiles to single out individual buildings (something that licenses do not permit). However, we managed to provide a building picking functionality thanks to a smart solution exploiting invisible extruded buildings used to delimit volumes of single structures.

This 3D representation can be offered on all cities in which the Google 3D tiles are active.



3.4.5 - FIWARE Smart Data Models and Snap4City IoT Device Models

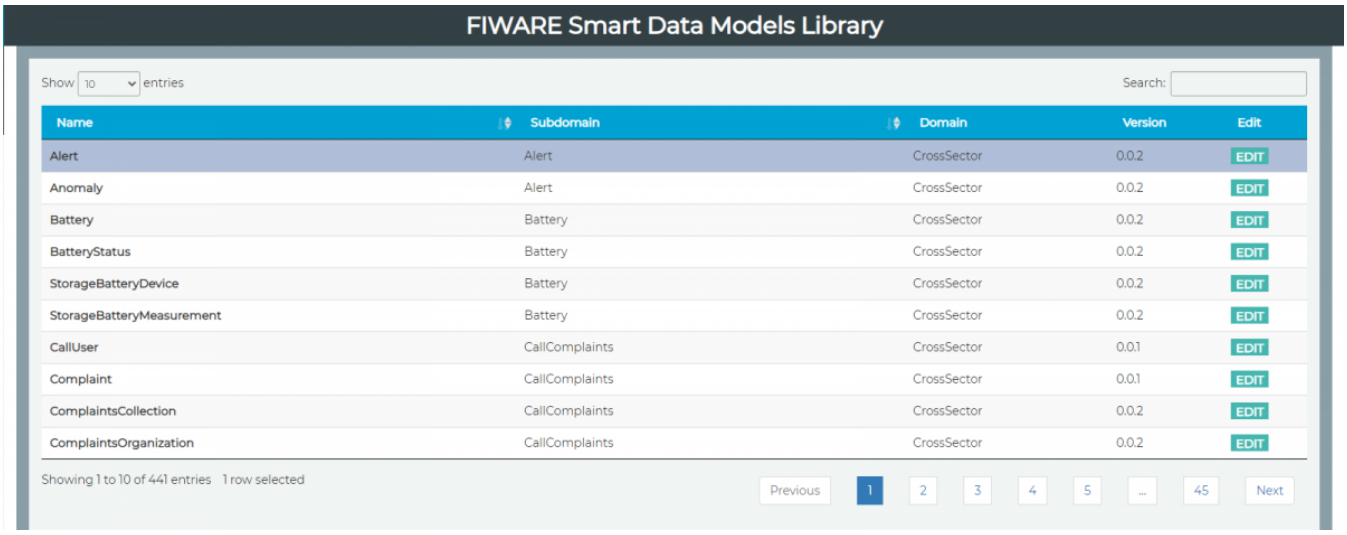
FIWARE Foundation drives the definition – and the Open-Source implementation – of key open standards that enable the development of portable and interoperable smart solutions in a faster, easier and affordable way, avoiding vendor lock-in scenarios. The Foundation also provides to Smart Data Models (SDM) for each domain of interest, such as Smart Cities, Smart Energy and so on.

Snap4City allows to create devices according to:

- **FIWARE Smart Data Models**
- **Entity Models / IoT Device Models of Snap4City**
- **Any custom model**

The solution is based on Snap4City Entity / IoT Directory which allows to:

- Collect the Smart Data Models from the GITHUB, where they are defined, versioned and listed.
- List, search and browse FIWARE Smart Data Models, manage them.
- Edit and customize the Smart Data Models, if needed.
- Automatically register Entity / IoT Devices which are registered in External Brokers, according to their Smart Data Model or any data model they have.
- Recognize automatically Broker Messages of entities/devices which are compliant with Smart Data Models or custom data models.
- Register automatically Smart Data Model, Entity Model, IoT Device Model or custom model to Knowledge base and storage to be immediately used in the creation of Dashboards and tools.
- Any registered device including those of Smart Data Model, Entity Model / IoT Device Model or custom model can be managed into Proc.Logic / IoT Apps for data processing, data transformation, adaptation, and Data Analytics.
- Delegate in read, read/write messages on devices, and also modify the model.
- Define rules for mapping undefined Value Types, Value Units and Data Types of Smart Data Models which can change according to the developers towards formally defined types in the Snap4City Dictionary to make the data directly exploitable in tools and math transformations.



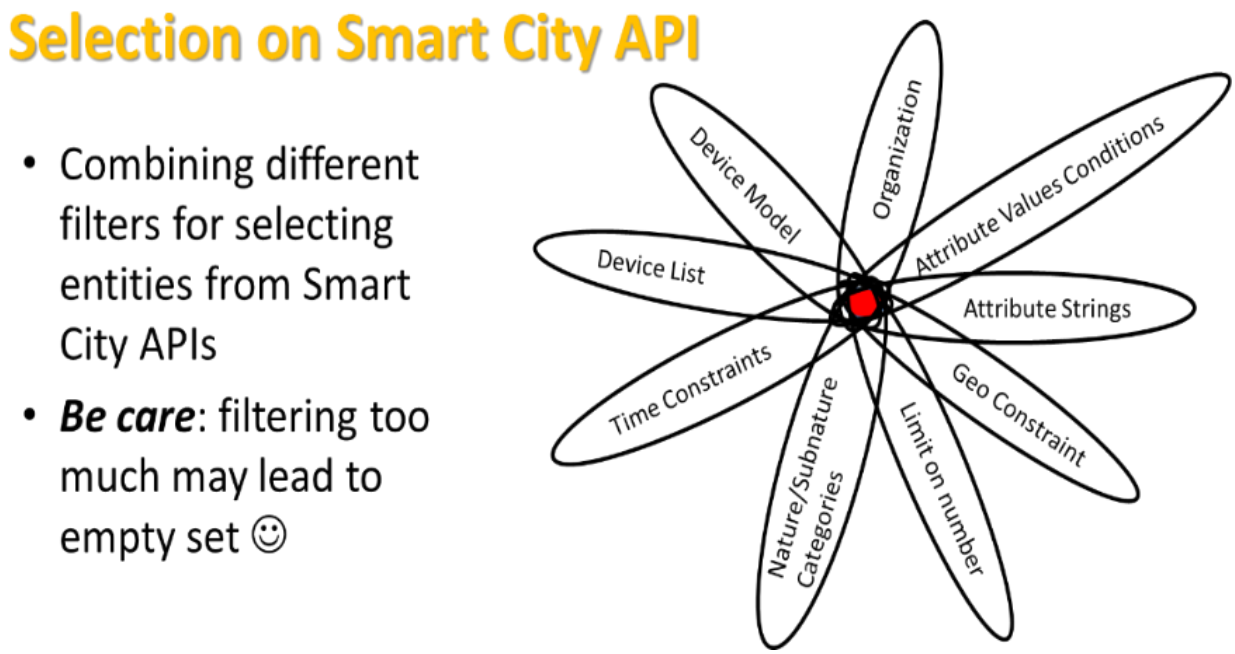
Snap4City Platform provides support to:

1. Manage different kinds of Brokers, Entity Instances / IoT Devices and Edge Devices based on different protocols, formats, and modalities to establish connections with the Platform.

2. Connect External and Internal Brokers. Internal Brokers are deployed, registered and managed, while the External Brokers would be only registered to use them since are managed by some third party. In general, the brokers could be multiservice, for example the Orion Broker with NGSI V2/LD protocol.
3. Register, manage and use messages conformant to any Data Model with any data type. Providing, receiving, managing, storing, and retrieving messages for any Entity Instances / IoT Device (of any Data Model) with its attributes and data types, and related access control. It is difficult to manage the huge variety of data kind. It includes the capability of addressing Smart Data Models, Entity Models, IoT Device Models of Snap4City, and any custom model.
4. Verify the correctness of Entity / IoT Messages of Entity Instances / IoT Devices.
5. Semantic Interoperability. This requirement is fundamental to achieve the coherence among different Entity Instances / IoT Devices (e.g., provided by different builders, addressing the same concepts, information on attributes). The Platform is capable to recognize/classify/retrieve information/attributes and behave accordingly to the semantic data model and types.
6. Support automatics cloud deploy of Internal Brokers, which are directly managed by the Platform to directly performs the registration of Entity Instances / IoT Devices. The result is a simplified experience for the users to populate the network.
7. Register External Brokers. Snap4City can register Entity Instances / IoT Devices / Services of an External Broker into the Platform. Brokers can be single- or multi-tenant and to recover the Entity / IoT Devices model managed by the Broker is the first step to perform their registration.
8. Discover Entity Instances / IoT Devices on Brokers (**External Broker Harvester**). Snap4City can harvest Entity Instances / IoT Devices from Brokers to automate registration and for their classification and search, which is based on their position, nature, value types and units, etc. In other words, it should be possible to discover/search (subscribe, get, send data) to/from Entity Instances / IoT Devices independently from their position/ connection in the IoT Network. The process of discovery is manageable in the sense that its execution time can be scheduled, and possible with external brokers that support a process for device discovery. The result consists of an automated or semi-automated registration process of Entity Instances / IoT Devices and matching the semantic model.
9. Semantic identification and matching. Snap4City recognizes automatically the model's device and its semantic information for attributes. In this way, when the discovery is performed, the devices' messages are automatically understood and add, problems are minimized.
10. Easy management to list and test Brokers, and Entity Instances / IoT Devices and query them for example via a graphic user interface. For each Entity Instances / IoT Device, it has to be possible to perform testing activities.
11. Manage Entity Model, IoT Device Model and Device Data Type ownership and access grant. This permits assignment/changing of the ownership and the creation of access grants to the entities (Brokers, Entities, Devices, and Data Models). In delegation management, it must be possible to grant, list and revoke them. According to GDPR, any entity must start as private of the owner. The delegation should be possible for organizations, groups of users, and single users and can be different types. In particular, it can be about:
    a) Entity Messages / messages - delegated users can read and write or only read the messages of a certain Entity Instances / IoT Device;
    b) Entity Models / IoT Device Models – delegated users can modify or only read the model structure;
    c) Entity Instances / IoT Devices – delegated users can modify or just view the device structure.

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

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

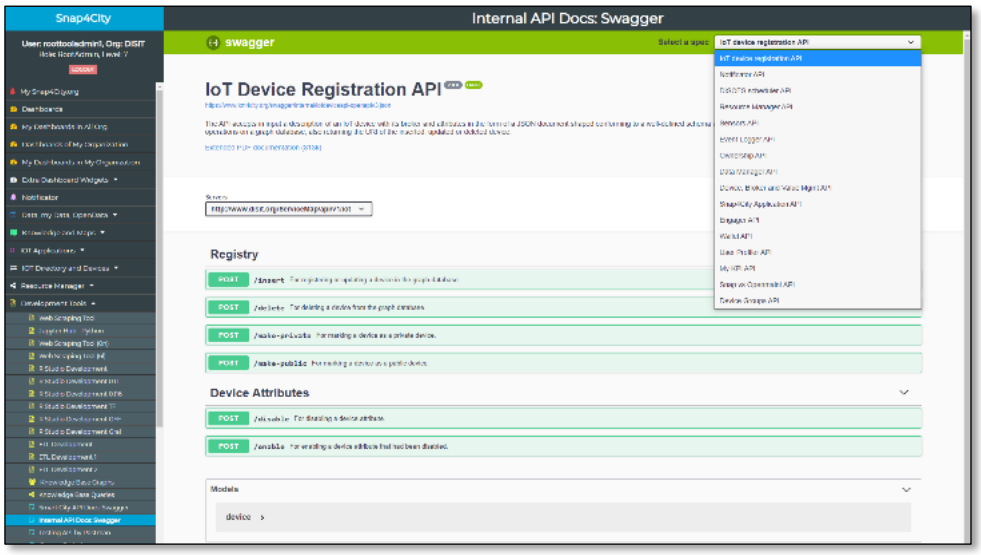


Snap4City solutions can provide data for Dashboards and for interoperability via several kind of APIs:

- **Advanced Smart City API, ASCAPI,** are provided from ServiceMap tool connected to the graph data base with the Km4City ontology and all the data relationships among entities. The requests on Smart City API can allow to perform temporal, spatial, relational and values filtered constraints. The Smart City API provides the first entry point to access data relational and Time Series. They also include the API for ODM, Heatmaps, Files, profiles, payments, etc.
- **IoT Orion Broker API,** which can be used to access at the data regarding the last data arrived to each IoT broker. Please note that in a large Snap4City solution several brokers may be present.
- **API Manager of Snap4City** management tool can be placed in front of the several APIs to (i) provide a unified entry point, (ii) perform accounting on the API consumption, (iii) placing the basis for providing billing on the basis of API usage and data consumption according to different business models: (a) monthly rate, (b) limits, (c) pay per play, etc. **API Manager of Snap4City** has an integrated SSO with the rest of the Snap4City platform or an independent KeyCloak or other Authentication service connected.
    - **The typical APIs managed by the API Manager of Snap4City are those offered from:**
        - **MLOps processes as tools based on AI, ML, LLM, etc.**
        - **API on HPC clusters or CPU/CPU** static, again for AI or simulation.
        - **IoT API of Snap4City**
        - **ASCAPI of Snap4City**

3.5.1 - Advanced Snap4City APIs (ASCAPI) and Smart City APIs 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 well as new modules for your solution. Snap4City APIs are accessible in **Swagger** and are divided into Internal and External.



External APIs: https://www.km4city.org/swagger/external/index.html

- **Smart City API:** To access the **ServiceMap** resources and queries
- **Km4city Web App API:** To exploit **MicroApplications** created as tools for **Dashboards**, totem, web Apps, etc.
- **Orion Broker K1-K2 Authentication:** To communicate with **Orion Brokers** exploiting the Secure Filter of Snap4City to enforce security to NGSI protocols.
- **To save and access HLT. For example, for the Heatmap:** To save and access **HeatMaps** of the Heatmap server, and also Traffic Flows, ODM, Files, etc.
- **Invoke computing and AI solutions: prediction, heatmap production, traffic flow reconstruction, simulation, what-if analysis, traffic optimization, etc. Coming from clusters and MLOps managed.**

For example, the **Smart City APIs** (the specific segment used for the web and mobile Apps) provide support to:

- **search data**: by text, near, along the line, POI, resolving text to GPS and formal city nodes model, resolving from GPS to street civic number, etc.;
- **get information on services** in the territory such as mobility and transport, cultural and tourism, e-health, public transportation, environment, weather, commercial, parking, waste, etc.;
- **get information 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);
  - **Participatory tools, fines tools, etc**.
- **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 to 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 ODM, traffic flow, files, etc.**
- **get suggestions**, assistance, and incentives for a more virtuous behaviour
- **Invoke computing and AI solutions: prediction, heatmap production, traffic flow reconstruction, simulation, what-if analysis, traffic optimization, etc.**

3.5.1.a – Examples of Snap4City Mobile Apps

List of **Mobile Apps** which are exploiting Km4City/Snap4City Smart City API.
https://www.snap4city.org/489

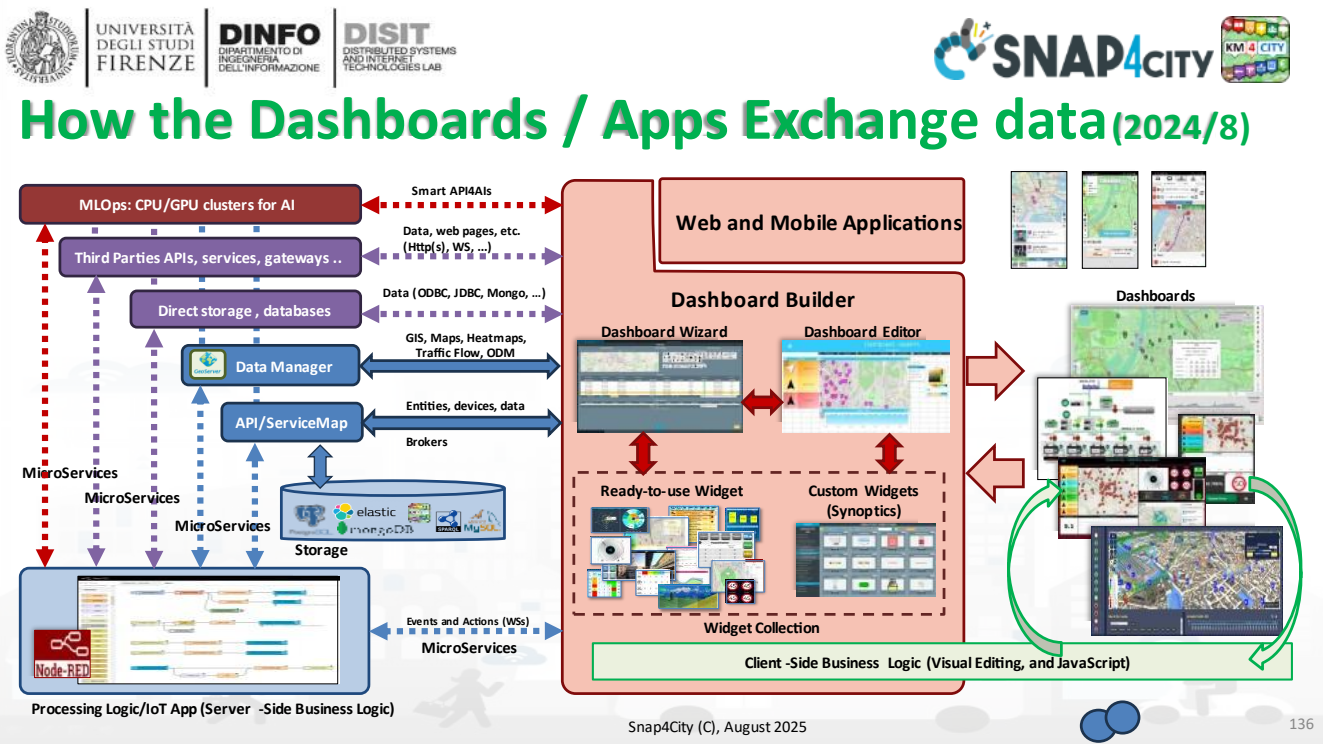


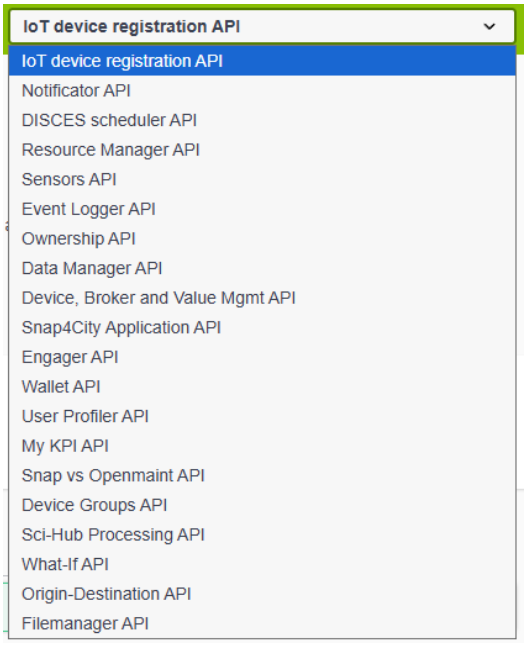
Figure – Dashboards / views and their connections with the platform and other services.

3.5.1.b – Example of Snap4City Internal API

https://www.km4city.org/swagger/internal/index.html

Internal APIs:

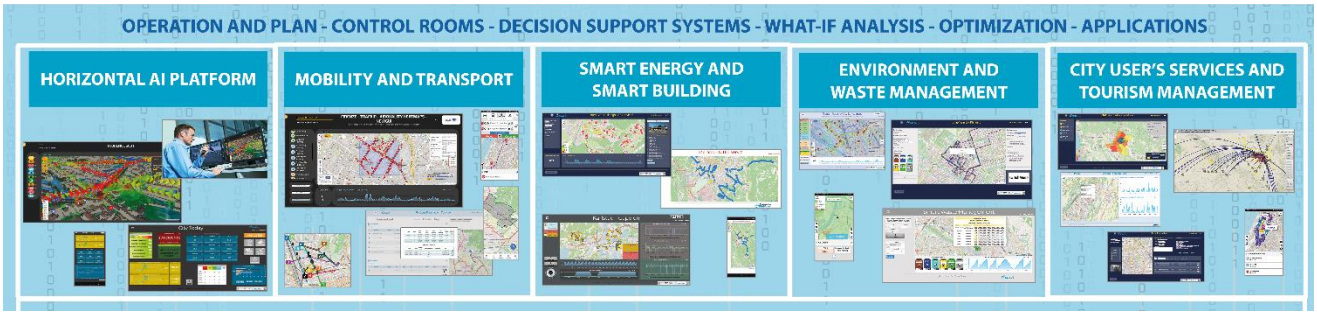
- **IoT Directory API: IoT Devices and tools API:**
  - **IoT Device registration API:** API of the Entity / IoT Directory
  - **Sensors API:** API of the Entity / IoT Directory
  - **Device, Broker, and Value Management API:** API of the Entity / IoT Directory
- **Mobile App Management**
  - **User Profiler API:** To manage the user profile for the **Engager** on Mobile Apps
  - **Engager API:** From the Engager to prepare engagements to the Mobile Apps
  - **Wallet API:** From the Engager to Wallet o the users of Mobile Apps and in general



  - **Snap4City Application API**
- **Resources and entities Management**
  - **Snap4City Application API:** To manage Proc.Logic / IoT Apps
  - **My KPI API:** To manage MyKPI, MyPOI, POI, etc.
  - **Data Manager API:** to access personal data
  - **Resource Manager API:** To manage resources in the marketplace
  - **Ownership API:** To manage ownerships and delegations
  - **Device Groups API:** To manage ownerships and delegations
- **Event Logger API:** to log data
- **Snap vs OpenMAINT API:** Integration with the workflow management, BPM, and ticketing

3.5.2 - Snap4City Applications, ready to use Vertical Applications

The main groups of Snap4City Vertical applications are reported in the main architecture and here.



- SOLUTION: Snap4City Digital Twin Simulation Support
- SOLUTION: SMART3R-FLITS: SMART Transport for TRavellers and Freight Logistics Integration Towards Sustainability
- SOLUTION: Security, Smart City Asset Management for Cuneo, Italy
- SOLUTION: ENERGIA: R&S di autoclave a mandrini multipli nel curing di serbatoi in composito per storage di H2 mediante ottimizzazione energetica con innovativo sistema di recupero Co2, energia termica e tecniche avanzate di fluidodinamica e machine learning.
- SOLUTION: UrbanDT4TF: Urban Digital Twin for Traffic Flow
- SOLUTION: ELLIE: On the UsE of Internet of Senses for the CuLturaL HerItagE
- SOLUTION: Snap4Rhodes: The "Single Smart City & Cyber Security Monitoring Platform" for the Municipality of Rhodes
- SOLUTION: SADI-MIAC: Integrated Decision Support System with Digital Twin Models and Artificial Intelligence for Business
- SCENARIO: City Users' Participation and Engagement with Snap4City, PDF
- OPTIFaaS: Operation and Plan, Transport Infrastructure and Facilities Support as a Service
- SOLUTION: 15MinCityIndex: understanding city areas by means of 13 different aspects, PDF
- SOLUTION: Energy Management and Control, PDF
- SOLUTION: Environment Control, Predictions & Prescriptions, PDF
- SOLUTION: Smart Light Control and Light Adaptive with Traffic Density PDF
- SOLUTION: Smart Tourism Management with Snap4City PDF
- SOLUTION: Traffic Infrastructure Optimisation: reducing travel time and emissions PDF
- SOLUTION: Traffic Light Plan Optimisation: reducing travel time, number of stops for vehicles and tramway lines: PDF
- SOLUTION: Snap4Building: monitoring, managing, controlling infrastructures PDF
- SOLUTION: Snap4City integration with Milestone X Protect, VMS, Video Management System PDF
- SOLUTION: Snap4City Digital Twin, PDF
- SOLUTION: eShare in a Snap - The innovative car sharing and car pooling service, PDF
- SOLUTION: Snap4City Smart Parking Manager and mobile App supports PDF

- SOLUTION: Exploit Snap4City in different Smart Waste use cases, waste manager, PDF
- eShare in a Snap - The innovative car sharing and car pooling service
- Digital Twin Citiverse FAQ to Snap4City
- SASUAM: Solutions for Safe, Sustainable and Accessible Urban Mobility
- SCENARIO: Exploit Snap4City in different Smart Waste use cases
- SCENARIO: Smart City Asset Management for Cuneo, Italy

Snap4City platform can cope with any data and information covering **multiple domains / scenarios** (https://www.snap4city.org/4) in integrated Digital Twins, coverage for requirements of MIM Plus, ENOLL, LDT, CitiVerse, e providing functionalities for operational management, simulation, what-if analysis, optimization and plan:

- **Horizontal Platform, a decision support system, which may integrate all the above plus: governance, security, asset management, solutions and applications**
  - *DOMAIN: Control and Plan Horizontal Artificial Intelligence Platform Digital Twin for All Domains: https://www.snap4city.org/1039*
  - **Goals:**
    - Increasing quality of Life, quality of services,
    - Decongestion, Decarbonization, Sustainability
    - increase efficiency and production optimization
    - Improve accessibility to services: citizens, Tourists, commuters, etc.
    - Improve security/Safety of city users, risk reduction
    - Costs reduction of services, energy consumption reduction
    - Reduction of emissions and EC taxations
  - **Horizontal homogeneous platform Uniform Technology for**
    - **Any Vertical operation/plan:** mobility, transportation, energy, environment, security, tourism, infrastructure and assets control, buildings, etc.
    - **Asset Control:** ICT devices, TV Cams, energy consumption, productivity, Hosts, VM, clusters, traffic gates, UPS, etc.
    - **AI Solutions**: early warning, predictions, simulation, what-if, optimisation, MLOps;
      - AI: Deep Learning, AI, ML, BERT, LLM/RAG, XAI (Shap/Lime), etc.
      - Simulations: SUMO, DORAM, Routing, TFR, Flooding, people flow, etc.
    - **Development Environment for any vertical, Digital Twin**: City Global and Local, IoT, VR, Visual Programming, business intelligence, CSBL, SSBL, Blockchain, etc.
    - **Interoperability**: any format, any protocol, any video management system, any sensor, any device, etc.
  - **KPI:** multidomain KPI, general management, early warning, early detection of critical conditions, 15MinCityIndex, SDG, etc.
  - **Mobile App:** modular applications, operators' modules, multiple cities, etc.
- **Mobility and transport solutions and applications**:
  - *DOMAIN: Mobility and Transport Operation and Plan Digital Twin: https://www.snap4city.org/1040*
  - **Goals**: decongestion, decarbonization, costs reductions, improve accessibility to services, improve Security/Safety of city users.
  - **Operation and Plan:** traffic monitoring, prediction, reconstruction, identification of critical conditions (early warning), dynamic routing, multimodal routing, city user behaviour analysis.
  - **Optimization and what-if analysis**: traffic light plans, traffic infrastructure. Reduction of: travel time, waiting time, # stops, CO2 emissions, consume fuel, travel time for tramways and busses.
  - **Public Transport**: Analysis of Mobility Demand vs Offer of Transportation (SUMO, DORAM2)

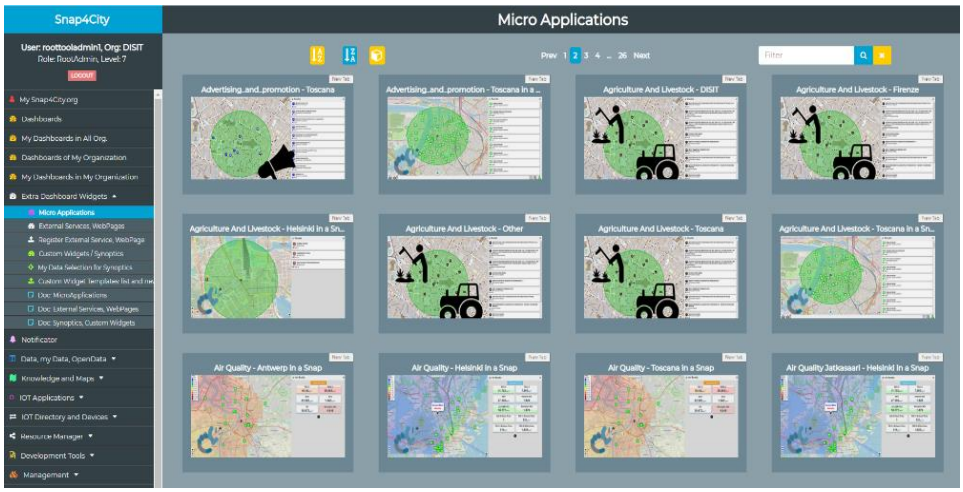
- **Parking Management:** monitoring, prediction, any payments, on-road/off-road.
- **Sharing / Pooling Management**: eShare and mobile app, bike-sharing, smart bike, fleet management.
- **KPI**: SUMI/SUMP, travel time, emissions, traffic status, accessibility, ..
- **Mobile App**: final users and operators: Info Mobility, traffic, parking, payment, fine management, city user participation, charging, overparking, …

- **Energy, eVehicles and Building's solutions and applications**:
  - *DOMAIN: Smart Energy and Smart Buildings Operation and Plan Digital Twins: https://www.snap4city.org/1041*
  - **Goals**: Energy consumption reduction; increment of efficiency; areas and building sustainability; improve accessibility to services, security and safety
  - **Energy Monitoring**: Building, floors, rooms, recharging poles, cabinets, Community of Energy, Data centres, Energy for Hot / cold, air condition, energy vs temperature and usage, etc.
  - **Energy Management:** Predictions, early warning, identification of critical conditions
  - **Smart Light Management**: LED/mixt, cabinets; lights control vs traffic, lights vs security, energy saving, luminaries profiling, group management, ..
  - **Smart Building Management**: consumptions, number of people, communities of energy, photovoltaic plants, sustainability, ..
  - **KPI**: Energy consumption, efficiency; light profiling and adaptation; autoclave industrial plants optimisation, photovoltaic plant simulation; consumption / usage, energy vs temperature, ...
  - **Mobile App**: monitoring, info-recharge, eSharing, booking, ..
- **Tourism, city users' behaviour, services solutions and applications:**
  - *DOMAIN: City Users' Services, Tourism Management and Safety Digital Twin: https://www.snap4city.org/1043*
  - **Goals**: improve Quality of Life and quality of services; Over tourism mitigation, sustainability; Costs reduction of services; Improve accessibility to services: citizens, Tourists, commuters, etc.; Improve Security/Safety of city users
  - **People Flow Analysis / Management:** in/out-door, retail, attractions**:** counting, tracking, flows, origin destination matrices, sentiment, recency/frequency, etc.;
    - Multiple sources: thermal & TV cameras, radar sensors, PAX sniffers, mobile data, Wi-Fi, traffic data, mobile phone data, card data, etc.
    - Suggestions: info Tourism, digital signages, engagement, ..
  - **Tourists Flows & Retail Management:** predictions of presences, services' reputations, suggestions on second offer, over-tourism mitigation, notifications, early warning,
  - **KPI**: 15MinCityIndex, energy vs people, over-tourism, accepted suggestions, precision.
  - **Mobile App:** final users services/informing and operators; Info Tourism, suggestions, people flows, info mobility, sharing, participation, engagement, ..
- **Environment and Waste Management solutions and applications:**
  - *DOMAIN: Environment and Waste Management Digital Twin: https://www.snap4city.org/1042*
  - **Goals:** reduction of emissions and EC taxations; cost reduction for waste collection; reduction of waste collection impact on mobility
  - **AIR quality (Indexes) monitoring and warning:** heatmap, notifications, early warning
  - **Environment Management producing prescriptions:** monitoring, long/short-term predictions, early warning: GHG, emissions, pollutants, aerosol, chemical plants analysis, coastal erosion (blue economy); Traffic Flow impact emissions, emission predictions.
  - **land slide predictions/warning**
  - **Waste Management and Optimisation:** costs reduction, optimal routing production, pay as you throw, avoiding out of bins, predictions of waste production on bins, alarms.

- **KPI:** SDG, 15MinCityIndex, QOS, costs, CO2, collecting time, EC KPI, emissions
- **Mobile App:** final users services/informing and operators, Info Waste for operators, participation, optimal routing, RAEE Collection, ..

Industry 4.0: depuration plants, production plants (monitoring industry plant, control and optimization, digital twin), production plant, predictive maintenance, integrated life cycles among different industry plant, such as on ALTAIR chemical plant, marketing analysis, production of suggestions and engagements;
https://www.snap4city.org/369

3.5.2.a - Snap4City MicroApplications



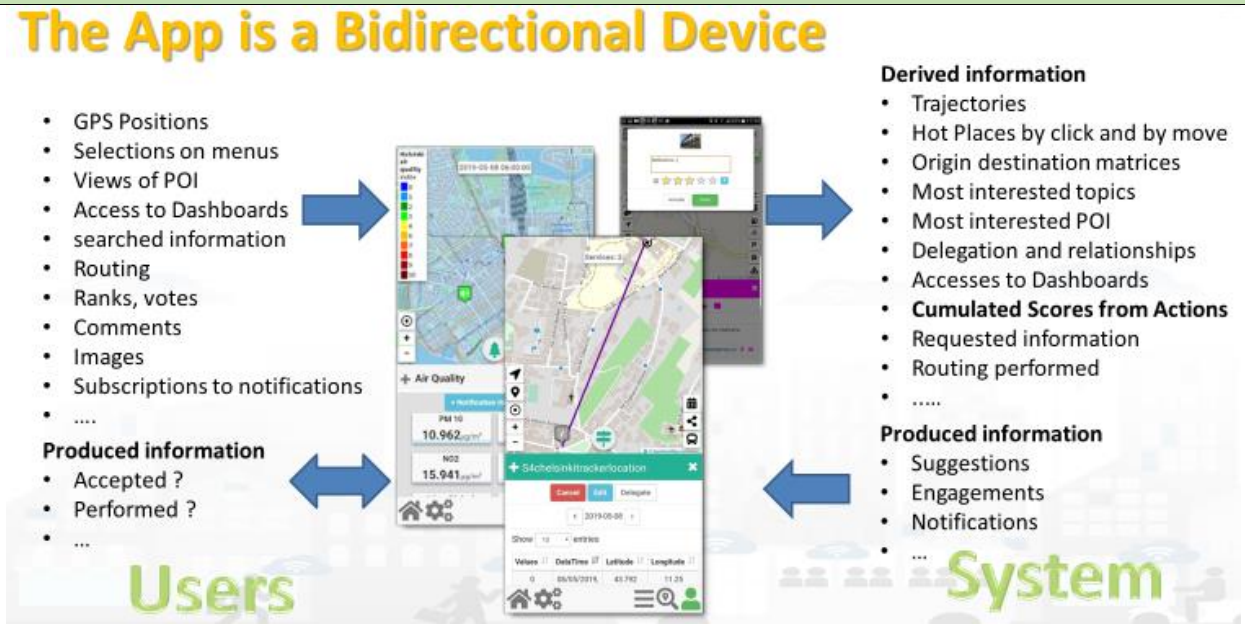
Snap4City also provides via API several **MicroApplications** and corresponding APIs derived from the Advanced Smart City API which can be directly embedded into **Dashboards** of any kind and activated from IoT Application Dashboards. Snap4City provides via API: (i) several **MicroApplications** and corresponding APIs derived from the Advanced Smart City API which can be directly embedded into Dashboards of any kind and activated from Proc.Logic / 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, Entity Instance / 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 publicly accessible.

Using the APIs you can develop your Web Applications.

In alternative also Dashboards can be used to create *Snap4City Smart Applications* since a set of Dashboards can be connected, you can develop a business intelligence tool by using them with some logic of interaction, to make them interactive. You can implement the logic on Proc.Logic / IoT Apps, and/or for advanced users, also on JavaScript directly embedded into the Dashboards.

3.5.3 – Snap4City Mobile Applications and User Behaviour Understanding

The **Snap4City mobile Apps** are bidirectional tools. They provide the users with geolocated and contextualized information and services, allow them to perform selections/queries, view POI, perform routing, get predictions, see images and comments, and receive notifications and alerts, in various domains: mobility, parking, environment, government service, health, weather, waste, light, etc. On the other hand, mobile users also provide information to the Snap4City Platform from which the platform can Derive information: trajectories, hot places, ODM, etc. That information is provided to the users in a 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/course/p7/



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

Snap4City Smart City APIs and services can be used to create several Web and Mobile App of different kinds. 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 prevents (i) creating/modifying their Mobile Apps for exploiting Snap4City API, (ii) starting from Snap4City App to create a new App for your city or area, etc.

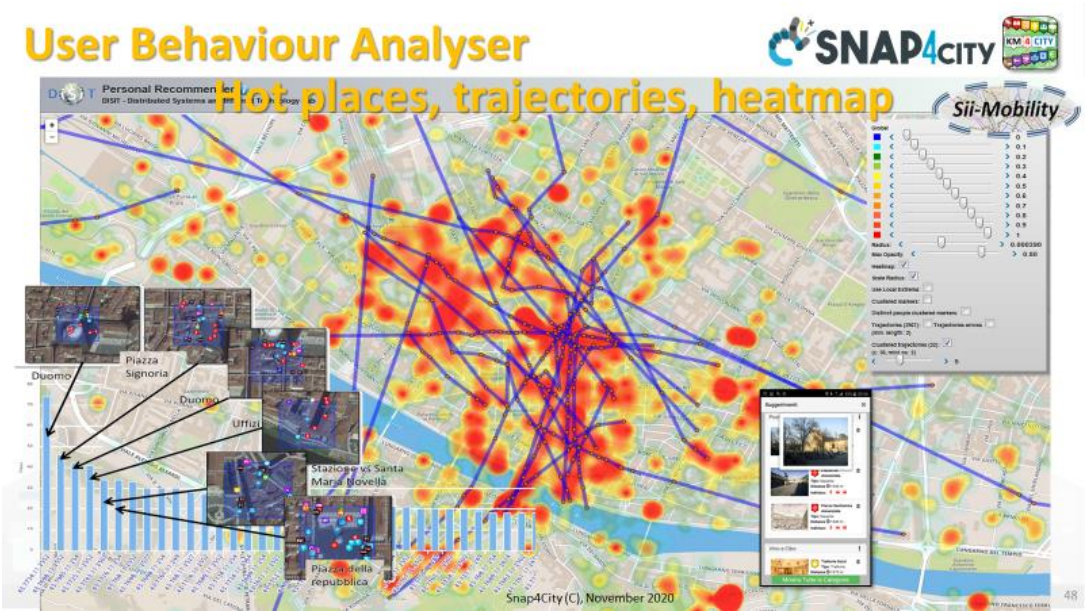
https://play.google.com/store/apps/details?id=org.disit.snap4city.mobileApp.tuscany&hl=en_US

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/492
- Helsinki in a Snap: https://www.snap4city.org/493

Personalization and Profiling, user behaviour analysis

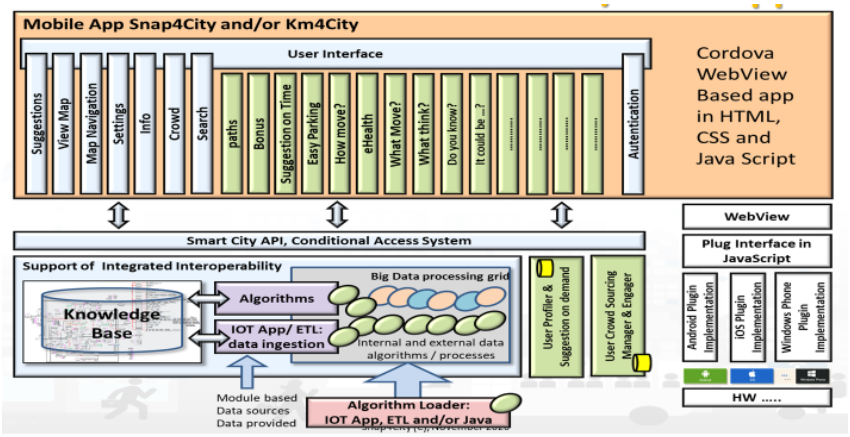


Apps Users are profiled in the server according to the GDPR, collecting a range of data as described above. This approach allows us to tune the service according to the collective and personal user behaviour, to improve the user experience by providing personalized (i) main menu arrangement of the functionalities 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 a further selection). Therefore, the menus offered for each profile kind are modified according to the statistics calculated on the collected data, by defining the positions of 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 with private cars and less for those who travel by public transport or by 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.

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

This kind of Mobile App has been recently updated using Flutter.

The App Development Kit is based on the Apache Cordova framework which allows realizing hybrid applications on multiple platforms. The applications consist of a user interface implemented with HTML, CSS, and JavaScript, and plugins that allow using 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 based on 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, 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 appears and in the desired place according to the server manifest of the module.

3.5.4 – Users' Participation and Engagement

The city users need to be informed about what is going on in the city.

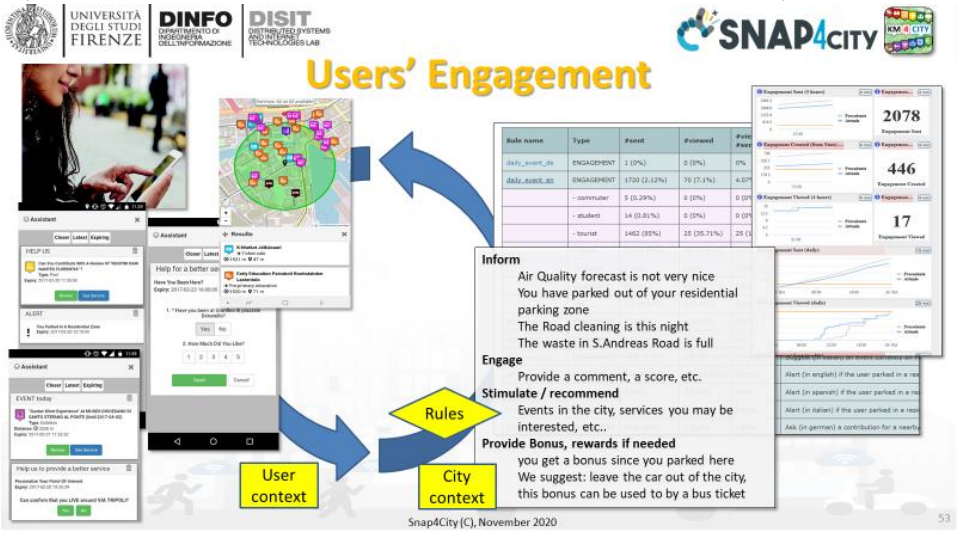
A number of other solutions to engage city users have been developed:

- **Info mobility video stream** with superimposed information regarding traffic, weather, number of people, services conditions (see Malta deploy)
- **Collection and automated processing of questionnaires recalled via QR**, processing via SnapAdvisor to identify critical conditions, service quality and understand complains, etc. (see Malta, Firenze, Varna, Bisevo, etc.). The collected text automatically processed by LLM SnapAdvisor can be used to detect critical conditions and create alerts, alarms.
- **Participation tool** is comprised of a module for the mobile app or a stand alone mobile app on which the city user may provide complains and comments and upload a media. On Server side the Participation Manager collect the complains and the operators may decide to activate some ticket. The collected complains in Text can be automatically processed by LLM SnapAdvisor to detect critical conditions and create alerts, alarms.

More complete solutions for user engagement via Mobile App

On the other hand, most mobile platforms have different approaches for providing asynchronous push notifications to the device when the App is not working in foreground and is not executed by the user. In addition to the alerting, the movements of the device should be also communicated to the server to get new context-based suggestions and alerting, such as: the weather forecast in your area, alarms of civil protection,

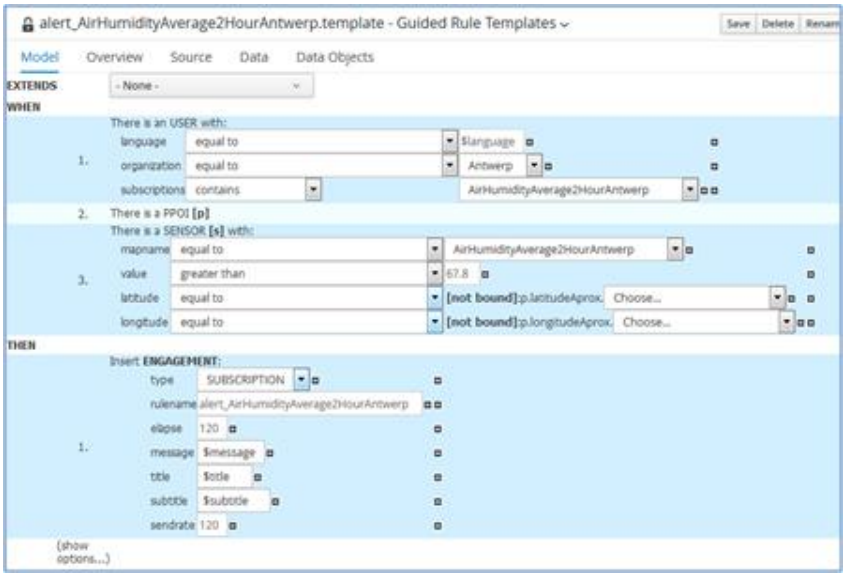
environmental status, closer car park, etc. Some of these innovative and smart features are produced as suggestions, engagements from a personal assistant, predictions produced by the data analytics modules (e.g., parking, the arrival of busses, etc.), and alerts (e.g., civil protections, changes in the traffic, other events). A new and additional feature of notification can be obtained by integrating Telegram, which has to be installed into the App. In this case, the server identifying the critical situation has to interact with the Telegram server.



User Engagement is one of the main challenges in 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 cities need to communicate with the city users and assess their reactions to different purposes. The communication of information is the first step, the real need is User Engagement, which means efficiently and deterministically obtaining reactions and evidence of success, once the city user is

- Informed via messages, for example about the closure of certain streets;
- Involved by providing back comments and opinions on specific aspects (answering 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.



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 takes a 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 the city user that there are parking spaces in an area reserved for other kind of users.
In most cases, when the engagements are sent to stimulate a change in habits or to inform the city user, some incentives/bonuses could be provided to increase acceptance and attraction. The incentives can be in the form of discounts, points, awards, etc. Moreover, they can be assigned based on specific rules, for example, when a certain context or a certain user behaviour is detected (for example, when the system understands that the user is moving with a private car), or when a 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).

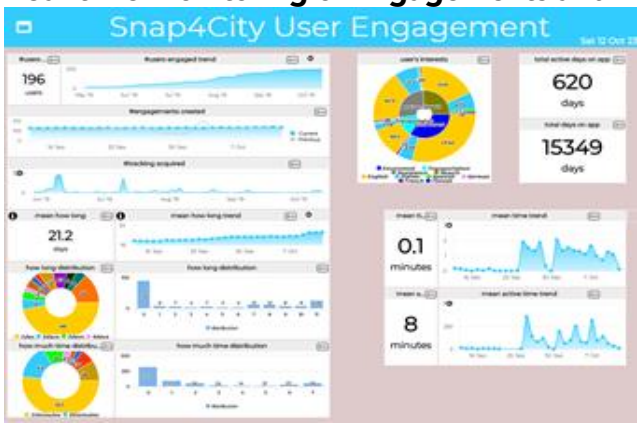
The solution has been used by ATAF, CCTNORD, and BUSITALIA (public transportation operators) in the 2019 campaign in Florence Livorno, and Pisa. The above mechanisms can be implemented/configured in the Snap4City Engagement tools which include:

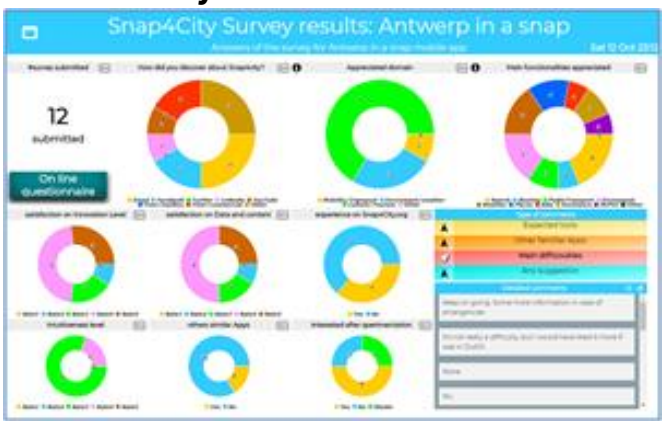
- A tool to easily formalize Conditions at which Actions has to be performed, such as the release of incentive, change of status, assignment of points, etc.;
- An engine for real-time and automated estimation of Conditions and Actions, which is also capable to verify if the message has been received 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, 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 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 collecting users' personal data that is exclusively available to them 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 the creation of a participatory community and a group of active city users who can contribute to the day-by-day activities of the Living Lab.

Real-time Monitoring of Engagements and Provided Surveys





The above solution can be observed at work in the Tuscany region, Antwerp, and Helsinki by downloading and installing the Mobile App from Google Play and the 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 scenarios: https://www.snap4city.org/4
- TC8.10- How to manage the user engagement rules in the Snap4City platform https://www.snap4city.org/486
- Mobile App on all Stores: https://www.snap4city.org/489
- Dashboard: user Engagement for Administrators https://www.snap4city.org/472
- Snap4City mobile App: City in a Snap (available for Antwerp and Helsinki) https://www.snap4city.org/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/20

3.5.5 – Origin Destination Matrices, Algorithms, and tools

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

- **Traffic flow sensors**: reporting the traffic flow in total, and eventually specific flow of touristic busses, and not in the areas close to traffic. If they take plates of the vehicles (or their hash codes, etc.), they can be used for creating ODM. The estimation of ODM is quite simple, the intermediate paths and points are not known.
- **Mobile Apps**: reporting the movement of people, in the city and among the cities, inflow, and outflow. The kind of users to which they refer depends on the App profile, and the number of tracked trajectories. The tracking can be anonymous and anonymized. They are a partial view of the complete people flow since not all have the same App. Moreover, the App users may use different travel means, and the recognition of the travel means is not 100% precise. Snap4City has developed a tool to solve t his problem.
- **Onboard units:** for example, coming from taxis, from insurance black boxes, for fleet management, bus OBU (on Board units), etc. Also, in this case a partial view of the flow is produced according to the different kinds of sources and the number of trajectories. They can be confined to the city (for the taxis) or a 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. The estimation of ODM is quite simple, the intermediate paths and points are not known.
- **People sniffing** tools, for example using Wi-Fi and/or Bluetooth sniffing devices. The estimation of ODM is not simple since the mobile phone produce randomized MAC addresses, the intermediate paths and points are not known.
- **People Tracking by TV cameras**, the estimation of ODM may be not GDPR compliant since the recognition of the person should be done, the intermediate paths and points are not known.
- **Cellular network data**, coming from the Telecom Operators, referring to cells or clustered areas. The mobile operators have the precise data but what they can actually sell are anonymized data on ACE areas which are too large to understand the micro movement, and are good to understand the movement from commuters, and the inflow/outflow of the major city areas.

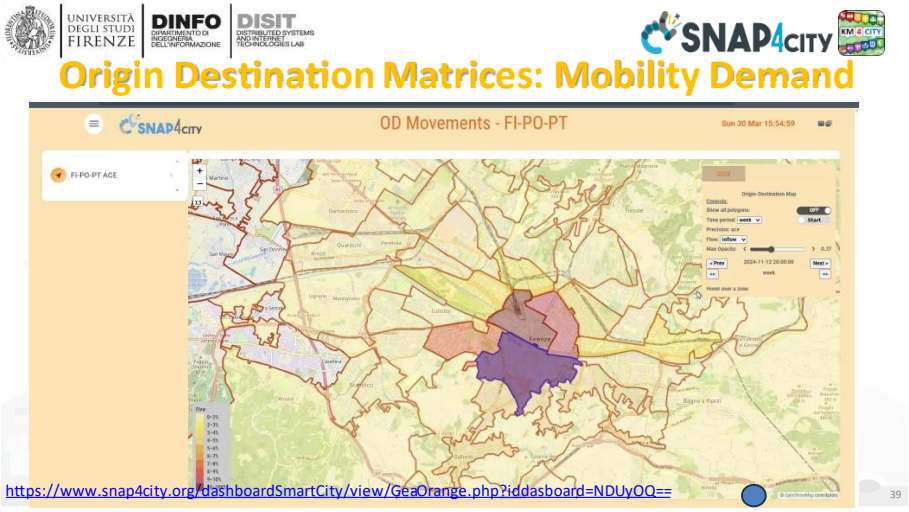
Due to the introduction of GDPR, some of the above data are becoming harder or practically impossible to be obtained. All these kinds of ODMs can be (i) used for simulation and analysis tools, (ii) for direct understanding of flows via visualization. Please note that each of them is producing flow for different kind of population segment: using mobile phone, just moving, only on vehicles, etc. They are not totally interchangeable, some of them may represent a demand of mobility, others the effective movements of people. The time resolution is also different. Census data are seasonal, those of the operators are daily or monthly, PAX counting can be hourly, daily, weekly, etc. etc.

Tools for the simulation and analysis may derive suggestions and optimisations for public transportation service tuning, city consumption of services, city cleaning, service tuning, etc. On the other hand, a direct visualization may allow the decision makers to perceive the critical aspects of the services and of the mobility and transport infrastructure at a glance. For ODM visualization, the classical matrix representation is not very effective since the matrices tend to the very sparse. See for example the figure.

Snap4City provides a Large set of new tools for ODM management (computing, loading, accessing, and rendering):

- How to: Origin Destination Matrices, ODM, load and view

- How to: get ODM data using OD-API
- How to create ODM: Tests to Insert ODM
- HOW TO: from PAX Counters devices to Origin Destination Matrices, ODM

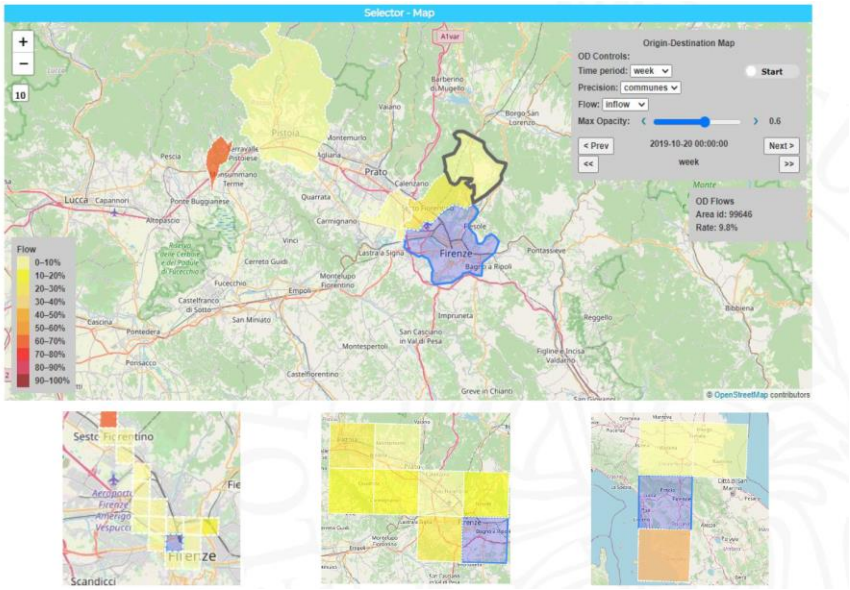


https://www.snap4city.org/dashboardSmartCity/view/Gea-Orange.php?iddasboard=NDUyOQ==

In Snap4City Origin-Destination Matrices (ODM) can be represented in 3 ways, using:

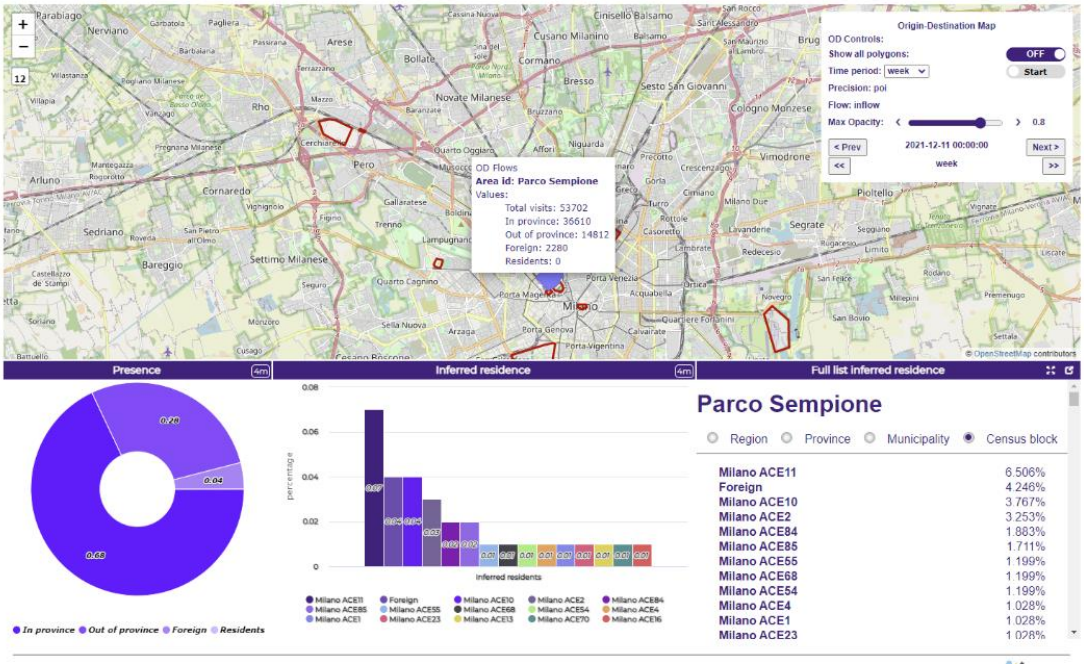
- **MGRS cells** [https://mgrs-data.org/] of different sizes (100 km, 10 km, 1 km, 100 m, 10 m, 1 m);
- **Administrative** areas defined by public organizations.
  - **GADM** data (v3.6) [https://gadm.org/] are used to describe areas worldwide down to the precision of individual municipalities.
  - **ACE** areas for mobile cellular data. Higher precision down to census areas are available for Italy only, exploiting data obtained from ISTAT [https://www.istat.it/notizia/confini-delle-unita-amministrative-a-fini-statistici-al-1-gennaio-2018-2/]
- **Custom shapes**, of any form, justa geojson shape

All geographical representations are reported in EPSG 4326 (WGS84), as latitude and longitude, for an immediate visualization with tools and libraries adopted in the Snap4City dashboards.



A smarter integrated application with ODM

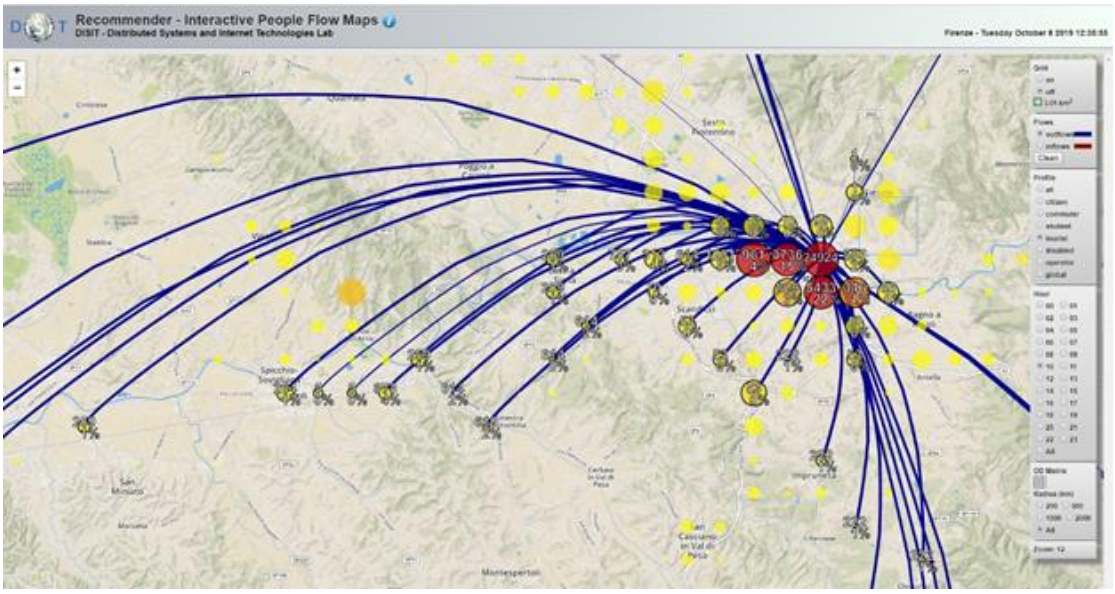
The computation of the scalable ODM is time consuming and it is typically updated daily taking into account data of the last months, years, or weeks. Produced for ENEL-X presented here in the last figure.



Former tools of Snap4City to cope with ODMs:

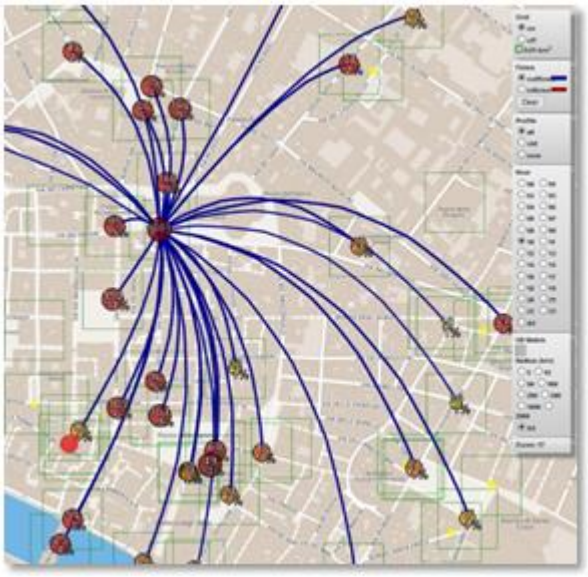
Traditional representation of ODM:

An alternative representation, directly invented by the DISIT Lab is the Spider Representation. The Visual Analytics tool for OD Matrix representation of the 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 ODM Visual Analytic tools, in which you 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 ODM 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 ODM computed from Wi-Fi data is reported.
- Define the time slot or see the whole 24 hours
- Show the distribution on barseries, Pie, and histograms
- Work on different areas: region, province, city, MGRS, ACE, etc.
- Show animations for showing the evolution of the ODM
- See the evolution over time of the ODM sequence



See the Dashboards produced for *Life in Antwerp*, *Life in Helsinki* in which a view of ODM as spider is proposed.

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

More details and examples are on:

- List of all scenarios: https://www.snap4city.org/4
- TC1.18: Origin Destination Matrix: https://www.snap4city.org/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 / Proc.Logic and MicroServices

The Snap4City **Proc.Logic / IoT Apps** can be executed any kind of computer, server, on-cloud, or Edge Devices (on-premise), or mobile phone, etc. When IoT Apps are executed on Edge Devices, they may directly communicate with the IoT Apps or Dashboards on-cloud or storage by using Brokers (to which all other entities can be subscribed) or directly. On such grounds, in Snap4City, the Proc.Logic / IoT Apps are defined as:

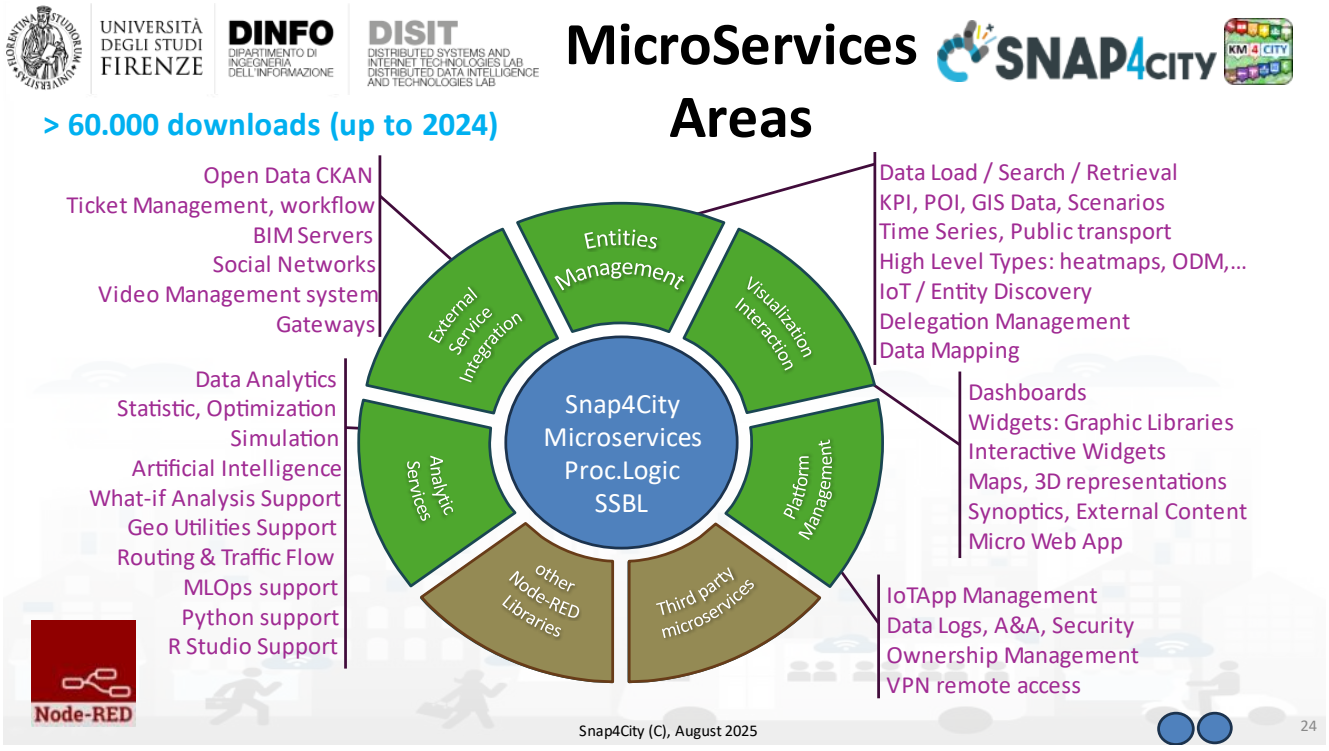
Proc.Logic (alias IoT App) = Node-RED + Snap4City MicroServices.

The Proc.Logic / 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 transformations 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.

We suggest you to read:

PDF FILE: https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf



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

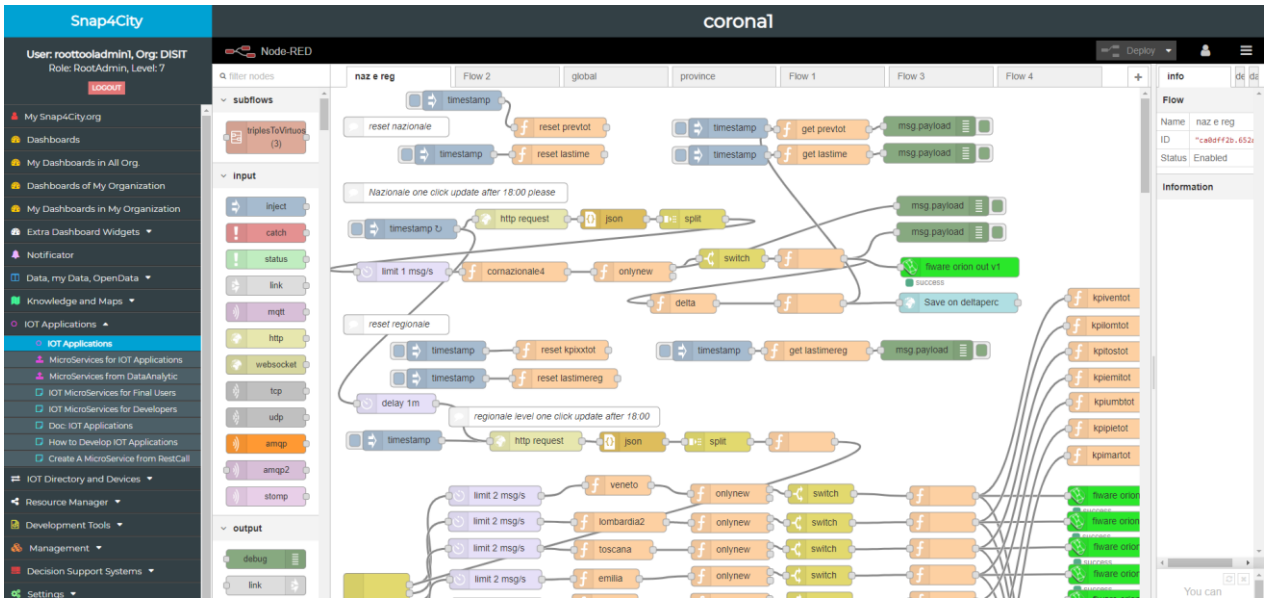
- communicate with the Snap4City **Resource/Data Manager** to save and load IoT App and flows, which is a marketplace of resources for sharing and marketing them in the community,
- login via SSO (Single Sign On), exploiting Snap4City **LDAP** and **KeyCloak (which is an open IAM)**, using **OpenIDConnect** (OAuth and Access Token), which can be connected on SPID, EIC, etc.
- manage Smart City / 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 **Proc.Logic /** oT Apps on Container/Docker, according to the elastic management solution of Snap4City (**Marathon** and **Mesos**, or **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 Edge with the standard Node-RED tools, without any restrictions.



Snap4City platform offers a large set of **MicroServices**, which provide 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 Proc.Logic / IoT App as a visual programming language. Among the MicroServices, the Proc.Logic / 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 Proc.Logic / IoT Apps. These tools, orchestrated by the Proc.Logic / IoT App flow, may automatically inform, announce, act and produce alerts and warnings on Entity / 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, several **Proc.Logic / IoT Apps** may need to be deployed based on the online requests, made by users/organizations. Proc.Logic / IoT Apps are managed and allocated on-demand based on the users requesting them, as well as executed on-cloud. To this end, an elastic (vertical and horizontal) infrastructure is provided to manage – in a scalable manner – the Proc.Logic / 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 Proc.Logic / IoT Apps and Containers irrespective of whether they are on-cloud Proc.Logic / IoT Apps or Edge/field Apps, whether they are child processes/containers for Data Analytics or Web Scraping, etc. Different kinds of Proc.Logic / IoT Apps are represented by different icons.

The Snap4City MicroServices allows the users to develop

Proc.Logic / IoT Apps in Node-RED which is a **standard** and 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, managing Data Analytics, and Web Scraping,
- flows for data exchange among platforms with multiple authentications,
- 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, data analytics, dashboarding, networking among Entity Instances / IoT Devices, data abstraction, etc. The Snap4City **MicroServices** are distributed into four official libraries of Node-RED nodes by the JS Foundation portal). The four libraries are dedicated to final users (basic) and to developers (advanced), plus a library for using D3.JS graphic representations on Dashboards and for tunnelling on Edge devices with Node-RED and Snap4City.

The version dedicated to Users provides outputs of Node-RED nodes that can easily be exploited by non-skilled users on JSON. 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 several nodes/blocks that can accept complex JSON message inputs to create strongly dynamic Proc.Logic / 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, Arm AXIS TV Cameras, 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 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 Entity Instances / IoT Devices and **Edge Devices, Proc.Logic / IoT Apps,** and **Dashboards**.

3.6.1 – Snap4City Library of the MicroServices for Smart Cities and Industries

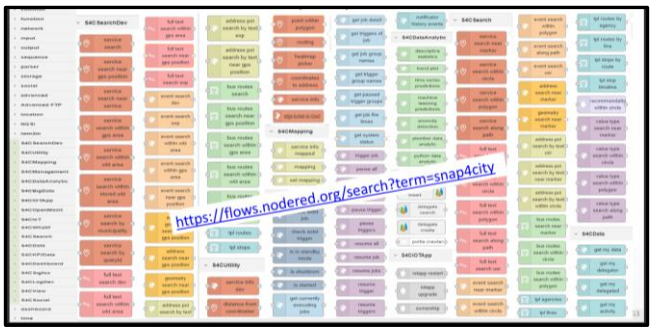
To satisfy the smart city requirements, in Snap4City, a collection of more than 190 MicroServices, as Nodes for the Node-RED programming environment, has been developed.

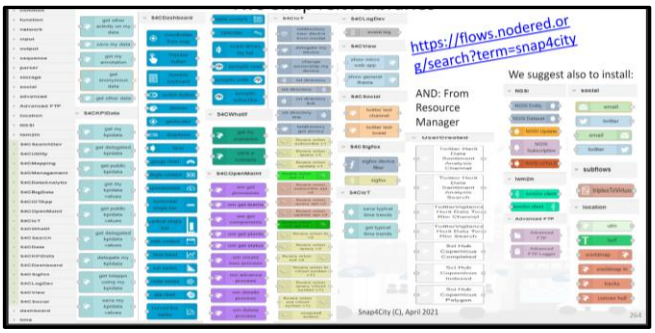
- https://flows.nodered.org/search?term=snap4city
  - https://flows.nodered.org/node/node-red-contrib-snap4city-user
    - Simple library for early users
  - https://flows.nodered.org/node/node-red-contrib-snap4city-developer
    - Advanced library for JavaScript developers
  - https://flows.nodered.org/node/node-red-contrib-snap4city-d3-dashboard-widgets
    - library for developers exploiting D3
  - https://flows.nodered.org/node/node-red-contrib-snap4city-tunnel
    - library for connecting node-red with tunnelling, remote maintenance
  - https://flows.nodered.org/node/node-red-contrib-snap4city-milestone
    - library for connecting with VMS of milestone
  - https://flows.nodered.org/node/node-red-contrib-snap4city-clearml
    - library for controlling MLOps processes in AI, LLM, etc.

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 Proc.Logic / 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 Proc.Logic / 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 several libraries from different providers. Moreover, the Snap4City **MicroServices** are at a level that can allow even non-expert users to easily develop Logic / IoT Apps for smart cities.

See **https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf**





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 Proc.Logic / IoT App construction. **Snap4City library of MicroServices in Node-RED** includes more than 190 nodes, block, microservices.

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 Proc.Logic / 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 Proc.Logic / 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 several libraries from different providers. Moreover, the Snap4City MicroServices are at a level that can allow even non-expert users to easily develop Logic / IoT Apps for smart cities.
Actually, there are more than 190 nodes/blocks in the Snap4City libraries on IoT App which can really facilitate your life and save your time in producing Smart Applications for composition of the following microservices and using those that you can install from internet, thousands of functionalities:

- **Data ingestion**: more than 130 protocols IoT and Industry 4.0, web Scraping, external services, any protocol database, etc.
- **Entity Management**, to create data models, instances, messages, etc., manage the authorizations.
- **Data Management**: save/retrieve data, query search on expert system, georeverse solution, search on expert system Km4City ontology, call to Smart City API, etc.
- **Data Transformation/transcoding:** binary, hexadecimal, XML, JSON, String, any format
- **Integration**: CKAN, Web Scraping, FTP, Copernicus satellite, Twitter Vigilance, Workflow OpenMaint, Digital Twin BIM Server, any external service REST Call, etc.
- **Manipulation of 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 (predictions, anomaly detection, statistics, etc.), managing MLOps via ClearML.
- **User interaction on Dashboard**: get data and message from the user interface, providing messages to the user (form, buttons, switches, animations, selector, maps, etc.), send data to special graphical widgets: D3, Highcharts, etc.
- **Custom Widgets**: SVG, synoptics, animations, dynamic pins on maps, etc.
- **Event Management**: Telegram, Twitter, Facebook, SMS, WhatsApp, CAP, etc.
- **Special tools as**: routing, georeverse, verify if a point is included into an area, Twitter Vigilance and sentiment analysis, get the closest civic number, distance from two GPS points, etc.
- **Hardware Specific Devices**: Raspberry Pi, Android, Philips, video wall management, etc.
- **Etc**. etc.

See for details **https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf**

3.6.2 Examples of solutions based on Proc.Logic / IoT App

Some examples and details are reported below. While a more complete overview about how to create Smart Solutions with Snap4City development environment is reported in the development life cycle document:

https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf

Examples:

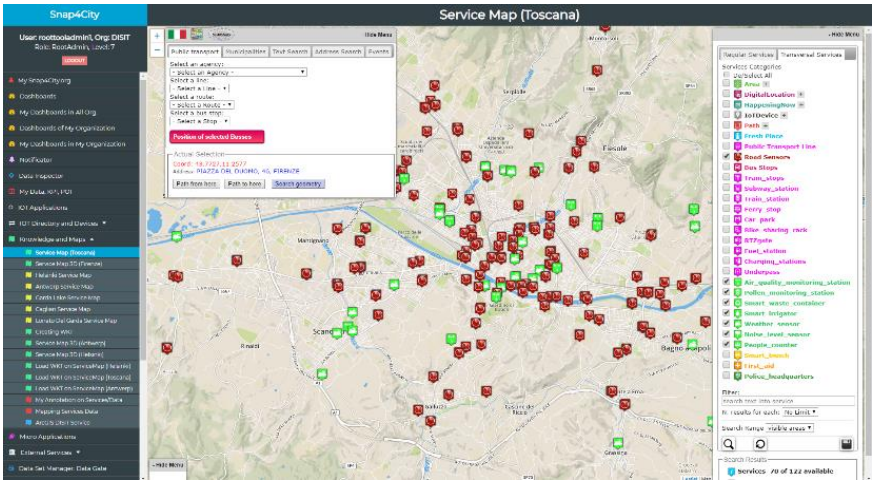
Access to Smart City Entities, which have different data models and, thus, different MicroServices may be required. Some Entities may have simple sets of Metadata, for example, the ones describing the POI, e.g., the title, description, web, email, GPS location, images, opening time, etc.; others may 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 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.;
- ODM, Heatmaps, Traffic Flows, Vector Fields, 3D buildings, etc.
- Entities with shapes: parking, gardens, cycling paths, routings, scenarios, 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.

To simplify this complexity, **MicroServices/nodes**, 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**, from ServiceMap visual tool, which provides a GUI for queries; from IoT Directory, from wizard, from Data Inspector, etc. 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, based on 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, **OpenSearch, etc.**

Therefore, the developer of the **Proc.Logic / IoT App** does not need to know all the tiny details about the large variety of adopted big data 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. In most cases, even complex data types have am Entity with metadata from which it is convenient to start, and it can be accessed with **MicroServices/nodes**, like "*Service Info*" and/or "*Service Info Dev*".

Discovering and Exploiting Entities (sensors and actuators) should not be different from discovering any smart city entity. In Snap4City Entities / 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 Proc.Logic / IoT App exploiting specific Entities / IoT Devices, some specific MicroServices can be used to discover the desired devices, regardless of the Broker, protocol or Entity/Device. To this end, the Entity/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 Entity/IoT Directory, Entity Instance / IoT Devices, and Broker, supporting a large number of different protocols and authentication models. The Entity/IoT Directory automatically registers new Devices in the Knowledge Base, and each new Entity/IoT element receives a ServiceURI, thus becoming a City Entity and POI. Several Broker kinds and actual brokers used are provided by Snap4City.org, 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 Proc.Logic / IoT Apps and Entity/IoT Devices, advanced tools, etc. This means giving developers the possibility to design the Proc.Logic / IoT App user interface. The user interface has to show and permit interactions between users through messages in the IoT Network, including the Proc.Logic, which may implement the logic of the user interface. The user interface is built by composing several graphics widgets in a connected Dashboard to present and collect data.

In Snap4City, the Proc.Logic / IoT Apps may be directly connected with Dashboards using widgets for:

- **Firing events from Dashboard to Proc.Logic / IoT App, and vice versa**. 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 Proc.Logic / IoT Apps as Input Nodes for the flow and can provoke events in the Proc.Logic / IoT App, according to the data-driven approach.

- **Showing MicroApplications and External Services** in a generic iframe widget.
- **Controlling from the Proc.Logic / IoT App the status** and the visualized data into the Dashboards. For example, the Proc.Logic / IoT App can change dynamically the content of the Dashboard widgets on the basis of the context. If an alarm arrives or is detected based on 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.

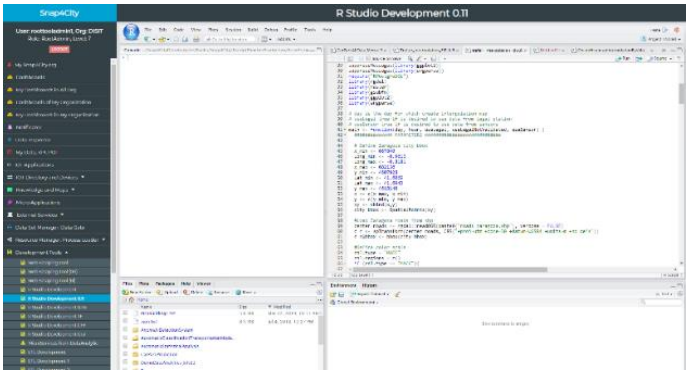
In addition, other Dashboard Widgets may not have a counterpart in the Proc.Logic / IoT App and may be directly added to the Dashboard using the Dashboard Builder, which can:

- 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: Orion Brokers, Entity Instances / IoT Devices, data stores, APIs, etc.
- include Virtual Devices that can send data directly to Broker, as an Entity / Device but from the user interface.

This also means that an Proc.Logic / IoT App may be connected to multiple Dashboards, and a Dashboard may be connected to multiple Entities / Devices and Proc.Logic / IoT Apps.

Data Analytics (AI, ML, etc., in python or R studio) can be:

1) **deployed and connected via specific MicroServices** automatically created by the Snap4City solution or provided by Node-RED. They create a new container which is allocated and managed by Marathon. As any kind of **Data Analytic** algorithm in RStudio, Python and they can be automatically **transformed into APIs and MicroServices** to be directly exploited in IoT Apps (for example developed in R. See Figure aside).



2) **created as static process to expose API in some server and thus also usable from** Proc.Logic / IoT Apps, node-red. **Created for the specific Infrastructure, that may be in Java or other services accessible via some APIs which can be mapped into a MicroService as well.**
3) **created and punt into a container image to be deployed on Kubernetes exposing some API and thus also usable from** Proc.Logic / IoT Apps, node-red.
4) **developed in Python on MLOps ClearML** which is capable to directly deploy them on CPU/GPU clusters, HPC, in sporadic or stable manners, see MLOps of Snap4city user manual.

All of those processes can **Exploit external Data Analytics Services** via Node-RED nodes (Azure, SAP, Watson, etc.) or directly from their code in RStudio or Python. A **REST call** invokes **External Services**, for example, gaining access to Twitter Vigilance API, The Weather Channel API, The Things Network API, etc.

In addition, the **Snap4City**.org portal provides several ready-to-use RStudio examples, and some of them are ready-to-use MicroService/nodes computing: routing, predictions, anomaly detection, trajectories, suggestions, statistics, heatmaps, ODM, optimisation, social media analysis via Twitter Vigilance (not included in the Snap4City suite, but integrated with it), etc. Using statistics, operating research, machine learning, artificial intelligence, explainable artificial intelligence.

Save and Retrieve Personal Data, for instance, time series for motion tracking, values of personal devices, clicks on Mobile Apps, POIs, shapes, KPIs (key performance indicators), Keys to access Entity / Device services, etc. The

possibility of saving and retrieving data from 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. 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 are also several 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, many so-called **GEO-utilities** have been identified and implemented in Snap4City, such as MicroServices/nodes to:

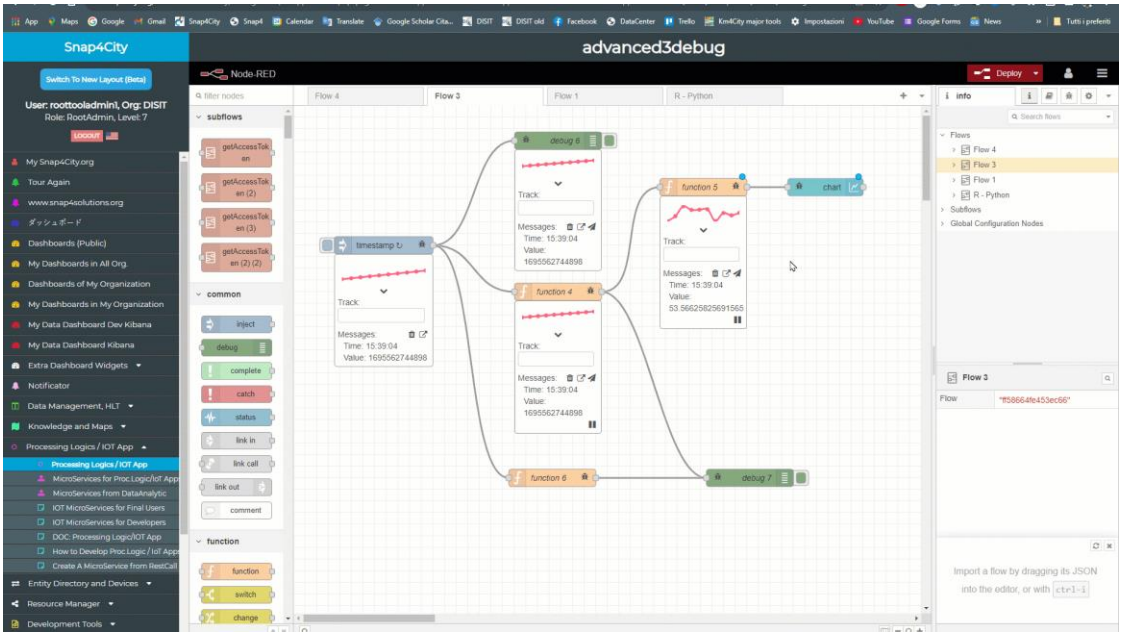
- Calculate the **distance from two GPS** locations.
- Verify if a GPS location is close to a shape/area, polyline
- Verify if a GPS location is inside the area or not to, for instance, verify the match with administrative areas and thus determine whether your dog is in the garden or the monitored bear 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.
- Computing routing: you need to install a specific VM with **Graphhopper** https://www.graphhopper.com/

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

Several 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.

3.6.3 - Snap4City Proc.Logic / IoT App with Debug Option

Snap4City added the extended debug capabilities, by using and extending CAULDRON tool. So that it is possible to attach at each Node extended debug capabilities. The integration of the CAULDRON offers the possibility of monitoring the passage of messages entering the Node-RED nodes, also providing the possibility of using breakpoints to block the flow through a node in order to be able to analyse it, make it proceed step-by- step, release the message queue held by the breakpoint or delete it. Specifically, the nodes of the CAULDRON version using the debug icon it is possible to see the trend of the last 10 messages. The graph draws lines if the plotted field contains an integer to highlight the value trend, otherwise it shows points to highlight the temporal trend. Clicking on the down arrow further modifies the interface. Nodes with both input and outputs provide other functionalities.



https://www.snap4city.org/964

The three icons next to the messages label have the following functions:

- basket icon allows you to delete the entire history of messages received and sent by the node;
- link icon to open a pop-up that shows the history of messages received and sent by the node;
- airplane icon to inject a personalized message into the node.
- pause icon puts the node at breakpoint, allowing you to pause the flow. Messages arriving in the meantime are placed in a queue.
- Fast Forward icon allows you to extract the first message from the queue and let it continue the flow.

The Value field is an editable textbox containing the Json of the first message in the debug queue. Snap4City enhanced version with respect to the original version of Cauldron allows you to select, through a special textbox present in each node, the attribute to be tracked. The permitted syntax is the usual JavaScript one, therefore for an object of the type:

{ A:[{B:'0b', }, {B:”1b”, }] }

Typing the expression A[0].B into the textbox shows the value '0b' while the expression A[1].B shows the value '1b'. If the expression does not match any attribute of the object, no results are shown. This functionality can be used in static ('static') or dynamic ('dynamic') mode: static mode means that the change of the textbox takes effect from the next message that crosses the node, while dynamic mode means that updating the textbox involves an instant update of the value displayed on the screen. If the textbox remains empty, the value displayed will be the default one, usually the message arrival timestamp for generic nodes.

The features just discussed are accessible via a dropdown UI specific to each node, which can be activated via the respective debug button on the node.
For the purposes of extending the graph, please note that it stores and takes into account only the last 10 messages received, and NOT the complete history since the activation of the flow.

3.6.4 - Blockchain support on Snap4City

The Snap4City blockchain can be used in multiple domains. For example, in the context of mobility and transport it could be used to certify the way we interact with the world, connecting everyday devices/entities, in the cities and industries to certify: (i) distribution of goods and services in last mile collaborative framework, such as city hubs; (ii) vehicle identity, mileages, maintenance operations, reparation from incidents, on board unit data for

insurances, on board unit data for professional drivers (to certify their readiness and attention level); (iii) tickets sold on the Mobility as a Service platforms, MaaS, also using it for revenue sharing towards the operators of multimodal traveling; (iv) computation of KPI indicators of the city with the aim of comparing them with other cities and for taxation, etc. In environmental applications, it can be used to certify measured values of pollution metrics, which are used for city taxations and to assess the achievement of the targeted values according to the European Commission. In the context of health to certify the source of organs and blood and their delivery chain of cold, the process for cleaning surgery instruments and hospital sheets, etc. In the context of energy and gas, it can be useful to certify the energy produced and exchanged in the communities of energies. It can also be used to certify, unicity, ownership and/or provenance of digital content, art, collectibles, and more through NFTs (Non-Fungible Tokens) that are unique digital assets authenticated on an immutable ledger. Blockchain can enhance smart home applications in regard to the fundamental security goals of confidentiality, integrity, and availability.
In the literature many specific frameworks have been developed, while the IoT/WoT platforms may be used to implement all those cases. IoT/WoT platforms are capable to manage device messages exchanged with any kinds of data/device structure. In this context, a strong push on defining data models has been realized, for example the FIWARE Smart Data Models, SDM. A data model provides a formal template format for IoT/WoT entity/messages with formalized variables/attributes including data types, units, etc., to produce identical entities, and this process should be certified as well. As to the platforms under analysis, messages from IoT devices are freely shaped, to assure data flexibility. For example, IBM Watson uses formats such as JSON or XML, without supporting FIWARE SDMs, which also are published in several versions over time. The IoT/WoT models are the formal template from which the devices/entities can be generated/registered, and in turn the devices/entities with their registration formalize the structure of the messages which can be received/sent on them by the brokers and storage. Whenever a message arrives from a device (which can partially provide pieces of information into its body, typically not the metadata, since most devices minimize data transmission), the platform is not capable to register the device, nor to correct the message link to former devices.

Snap4City added blockchain on IoT/WoT infrastructure adding a set of general-purpose features for certification/verification, and at the same time leaving flexibility to the developers of IoT/WoT solutions, allowing them to use the blockchain technology to set up their scenarios with a mix of certified and non-certified entities in a federated distributed architecture. The main contributions are on: (i) certification of any kind of IoT/WoT data models, devices and messages (with a particular focus on the certification of data models and its implications); (ii) automation of certifications according to the relationship among models and devices/entities, and among devices/entities and messages to guarantee consistency; (iii) certification of the hash for data messages by defining specific rules to cope for the flexibility of JSON objects of the simple schema validation; (iv) certification of time series, within a certain interval, for example to certify a trip, a mission, a delivery travel, etc.; (v) an architectural multi-organization solution to guarantee satisfactory performance in the certification of messages which is the most critical performance aspects.
The blockchain support is accessible via Entity/IoT Directory in which the developer can:

- Specify which Entity Models have to be certified, implying the production of Entities / Devices which are certified.
- Specify which Entities Instances / IoT Devices have to be certified, implying that messages on them will be certified
- Verify the certifications performed over time, on the time series of the devices / entities.

The developers can use these facilities to implement a large range of applications: voting, access rights, rewards, revenue distribution, healthiness, driver assessment, emission of coupons, digital collection, NFT, complimentary currencies, loyalty tools, crowdfunding, etc.
Ask to Snap4City@disit.org to get information and support to use this advanced feature on Snap4City.org infrastructure and on MicroX installations. See: https://www.snap4city.org/965

3.6.5 - Snap4City integrated with Milestone X protect VMS

The actual integration of Snap4City with Milestone is performed by means of a node-red library of Snap4City:

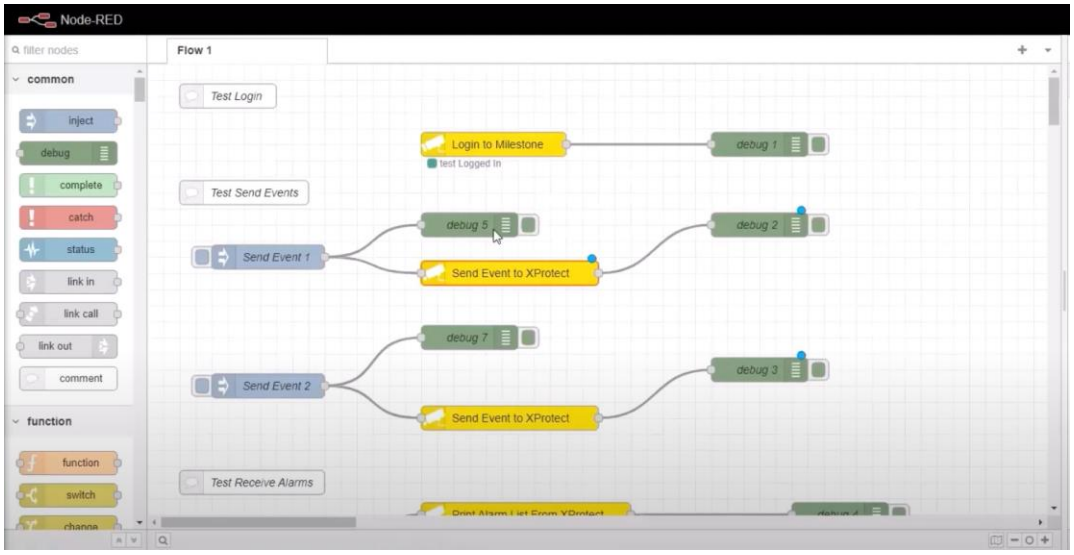
To satisfy the smart city requirements, in Snap4City, a collection of more than 190 MicroServices, as Nodes for the Node-RED programming environment, has been developed.

- https://flows.nodered.org/node/node-red-contrib-snap4city-milestone
- See video: https://www.youtube.com/watch?v=dxxYtOKOkvE&t=7s

The Snap4City node-red library can be used even **without the usage of Snap4City** tools. It allows to:

- send events from Node-RED to VMS X protect of Milestones
- receive events from VMS X protect of Milestones into Node-RED

The library can be installed on AXIS camera, on windows, Linux, raspberry Pi, etc., without the installation of .net.



The Snap4City platform can be integrated with Milestone VMS to create a more complete and integrated surveillance and security system. Here are some examples of how the Snap4City platform could be useful in combination with Milestone VMS:

- **Traffic monitoring and management**: The Snap4City platform can be used to integrate traffic information from Milestone VMS security cameras and display it on an interactive map. This allows you to monitor traffic in real time and manage it more effectively.
- **Incident Management**: In the event of road accidents, the Snap4City platform can be used to integrate information from Milestone VMS security cameras with other data sources, such as traffic data and weather conditions. This allows you to coordinate emergency resources more quickly and improve incident management.
- **Public Safety**: The Snap4City platform can be used to integrate information from Milestone VMS security cameras with other data sources, such as weather data and social media data. This allows you to monitor public events in real time and improve public safety.
- **Parking Management**: The Snap4City platform can be used to integrate information from Milestone VMS security cameras with other data sources, such as traffic data and parking data. This allows parking to be managed more efficiently and improves the availability of parking for citizens.

In summary, the Snap4City platform can be used in combination with Milestone VMS to create a more complete and integrated surveillance and security system. The platform allows information from different sources to be integrated and displayed on an interactive map, improving traffic management, public safety and parking management, among other things.

Node-RED is a visual programming platform used for creating IoT (Internet of Things) applications based on data flows. Node-RED can be used to integrate Milestone VMS and Snap4City, providing a way to process, analyse and visualize Milestone VMS camera data within Snap4City. Here's how I see the use of Node-RED between VMS and Snap4City:

- **Data Processing**: Node-RED can be used to process Milestone VMS camera data and transform it into a format compatible with Snap4City. For example, Node-RED can be used to analyse motion data from Milestone VMS cameras and send only relevant information to Snap4City.
- **Data Analysis**: Node-RED can be used to analyse Milestone VMS camera data along with other Snap4City data sources, such as traffic data and weather conditions. This allows you to extract more meaningful information and provide a more complete view of the situation.
- **Data Visualization**: Node-RED can be used to send processed and analysed data to Snap4City for display on an interactive map. This allows you to monitor traffic and safety in real time and make more informed decision

Furthermore, Node-RED offers a wide range of libraries and tools to easily integrate different data sources and communicate with different platforms. This simplifies the integration between Milestone VMS and Snap4City, reducing development time and improving overall system efficiency. In summary, using Node-RED between VMS and Snap4City can be very useful for processing, analysing and visualizing Milestone VMS camera data within Snap4City. This allows you to integrate different data sources and provide a more complete view of the situation, improving traffic management, public safety and parking management, among other things.

3.7 – Snap4City Data Ingestion/Interoperability

Snap4City is capable to interoperate since it can import and export data in multiple formats and protocols, more than 190. The interoperability is provided at multiple tools as listed in details on the table reported on https://www.snap4city.org/65 :

- **IoT App/Proc.Logic**, Node-RED: several difference protocols **Proc.Logic / 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 implementing Business Logic flow behind Dashboards and user interfaces.
- **Data Analytics**: any API, WS, SUMO, etc., any call
- **Dashboards**: WS, HTTPS, WFS, WMS, GeoJSON, JSON, NGSI, etc.
- **CKAN**: REST call, OAI PMH, OGC, WAF, etc.
- **ServiceMap**: RDF, HTTPS, RDF, ODBC, SPARQL, MySQL, Open Search, WKT, GeoJSON, etc.
- **GeoServer**: heatmaps, orthomaps, traffic flow, TIFF, GeoTiFF, GeoJSON, etc.
- **ODM manager**: multiple formats and shapes

- **IoT:** NGSI V2/LD, LoRa, LoRaWan, MQTT, AMQP, COAP, OneM2M, TheThingsNetwork, SigFOX, Libelium, IBIMET/IBE, Enocean, Zigbee, DALI, ISEMC, Alexa, Sonoff, HUE Philips, Tplink, BACnet, TALQ, Protocol Buffer, KNX, OBD2, Proximus, ..
- **IoT model:** FIWARE Smart Data Model, Snap4City IoT Device Models
- **General**: HTTP, HTTPS, TLS, Rest Call, SNMP, TCP, UDP, SOAP, WSDL, FTP, FTPS, WebSocket, WebSocket Secure, GML, WFS, WMS, WCS, RTSP, ONVIF, AXIS TVCam, CISCO Meraki, OSM, Copernicus, The Weather Channel, Open Weather, OLAP, VMS Milestone, TIM, HERE, OGC, ….
- **Formats**: JSON, GeoJSON, XML, CSV, GeoTIFF, OWL, WKT, KML, SHP, db, XLS, XLSX, TXT, HTML, CSS, SVG, IFC, XPDL, OSM, Enfuser FMI, Lidar, glTF, GLB, DTM, GDAL, Satellite, D3 JSON, …
- **Database**: Open Search, MySQL, Mongo, HBASE, SOLR, SPARQL, ODBC, JDBC, Elastic Search, Phoenix, PostGres, MS Azure, ..
- **Industry**: OPC/OPC-UA, OLAP, ModBUS, RS485, RS232,..
- **Mobility**: DATEX, GTFS, Transmodel, ETSI, NeTEx, ..
- **Social**:Twitter, FaceBook, Telegram, ..
- **Events**: SMS, EMAIL, CAP, RSS Feed, ..
- **OS**: Linux, Windows, Android, Raspberry Pi, Local File System, AXIS, ESP32, etc.

Snap4City is widely interoperable since it can get and send data with a large range of protocols and formats and has specific channels with major City management systems and IoT Networks:

- Acquire and send data from/to several sources/channels (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, databases and tools: https://www.snap4city.org/65
- Access and exploit data coming from the **Copernicus** satellite system.
- **Interoperate with AXIS camera,** in which the Snap4City can be installed, via Node-RED and with specific Plug-Ins for Video Processing, people detection and counting.
- **Interoperate with VMS Milestone** for collecting events and sending events via Proc.Logic Node-RED nodes Snap4City developed.

- 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 in the **Resource Manager**.
    - **Dashboards with CSBL can interoperate with any external APIs.**
- Exploit **External Services Web Applications** (distinguished for the 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 into Dashboards External Services and Tools to produce integrated views.
    - **Dashboards with CSBL can interoperate with any external APIs.**
- **Work with top used DBs** and more, such 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.
- **Interoperate with ERP via Proc.Logic/IoT App:** https://flows.nodered.org/search?term=ERP
- Acquire data from **WEB Portals of third parties web pages, Web crawling**. 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 areas transparently. This solution is presently in place among the Knowledge Bases of Antwerp/Helsinki, Tuscany/Firenze, Sardinia, etc. The resulting Service is called Super Service Map and it is integrated into 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, optional in Micro X
- **Interoperate with GIS API: WFS, WMS. Such as 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
- **Join, publish and harvest Open Data Portals** via DataGate/CKAN (CKAN is optional in Micro X, while the capability is accessible in all versions).
- **Interoperate with BPM such as OpenMaint:** https://www.openMAINT.org/en/home and see https://www.snap4city.org/625 or others tools (Open main is optional in Micro X, while the capability is accessible in all versions).
- **Interoperable with BIM via BIM Server https://bimserver.org/ or other BIM managers** optional in Micro X, while the capability is accessible in all versions).

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

- **Smart City APIs 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 GIS applications and platforms (such as ESRI ArcGIS Online, ArcGIS Enterprise, ArcMap, ArcGIS Pro, QGIS, GeoServer, etc.) to access Snap4City data. See interoperability with ArcGIS and other GIS environments in general https://www.snap4city.org/368
- **Smart City APIs allow to provide access to almost all kinds of HLT listed above.**
- **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**

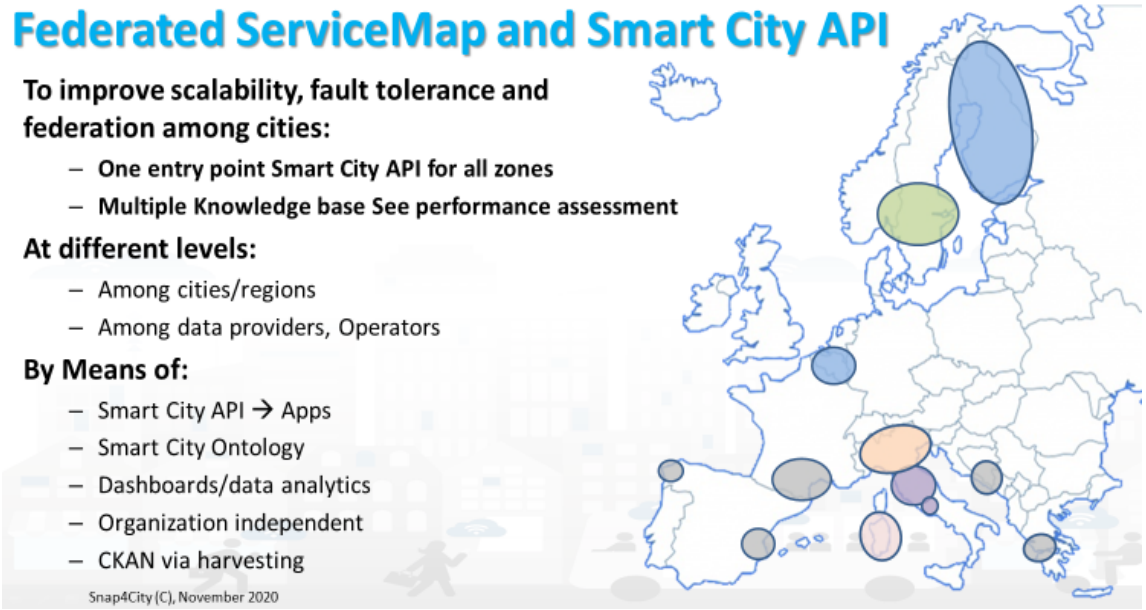
Base, Smart City APIs. This allows the creation of mobile applications that may move from multiple cities and areas accessing data and making queries transparently. This solution is presently in place among the Knowledge Bases of: Antwerp/Helsinki, Tuscany/Firenze, Sardegna, etc. The resulting Service is called **SuperServiceMap** and it is integrated in the Smart City API. See Section 3.8 and [**FederatedKnowledgeBase2020**], [**FederatingSmartCities**].

- **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 manager server and automatically produced based on the set colormap, the maps/heatmaps are provided to dashboards in standard WMS using 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
- **APIMAN (optional in most of the configurations)** can be configured to provide access to those mentioned API (internal and external), by means of some control in terms of business models for counting, for traffic, etc.

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 such as WFS (Web Feature Service), WMS (Web Map Service), for the exchange of Maps and geo-elements such as paths, points of interest, road elements, road graphs, etc. GIS protocols, according to their definition, are typically unsuitable as APIs 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 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 the level of the database.

Snap4City solutions can be federated with each other, creating cross-city environments among cities, regions, and in the same city among different areas if needed. The federation is at the level of Smart City API, and this allows 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****. Snap4City is 100% open source so the solution provided can be easily replicated without licensing costs. Federated Snap4City solutions with their KB may have overlapped and duplicated data, which are automatically removed by the network service of Super ServiceMap, which exposes the same Smart City API of Snap4City.*

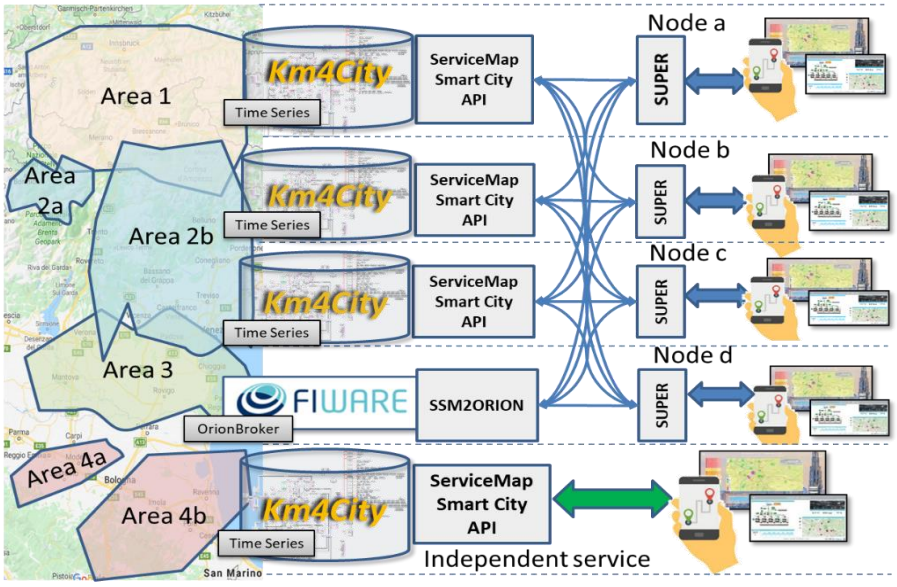


Federated Knowledge Bases, Multiple Snap4City solutions connected via their Knowledge Base (and/or connecting other IoT Orion Brokers via SSM2ORION which do not are integrated into a Snap4City solution), and Smart City APIs allow the creation of mobile applications that may move from multiple cities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

A distributed Smart City API-based solution for a set of cities/areas, let's say a federated network of Smart City APIs may provide several advantages:

1) support **distributed search** on the **Federated Knowledge Bases** network;
2) support **connections with smart cities of any size** in terms of the number and volume of data sets providing services to 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) 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 concerning 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 a **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 authorized 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 private data/services;
7) support the **addition/removal of Federated Knowledge Bases in the network** without the need of fully restructuring the network or 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) **respond in a 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.

To avoid having a single point of failure, the **Federated Knowledge Bases Supers** can be replicated into each node and the list of Super services on top of ServiceMap can be put accessible on one or more web servers for the 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 detail, each Federated Knowledge Bases may register in the Super network the descriptor of the multi-polygon area of your competence. This approach permits 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 perform data fusion. The Supers as well as the ServiceMaps may also implement some query caching solutions like all the other Federated Knowledge Bases.



3.9 – Snap4City Data Analytics Development and Exploitation

The design and development of **Data Analytics, DA, Processes (DAP)** is mainly performed taking in mind that their development cases are performed in Python or Rstudio. For DAP we intend the development of algorithms for some computation, data processing: KPI, predictions, optimization, simulation, etc., exploiting ML (machine learning), AI (artificial intelligence), XAI (explainable AI), operating research, statistics, heuristics, LLM, NLP, etc.

The DAPs can be devoted performing tasks of model training, model execution, computation, simulation, etc., in batch or stream, on demand or as a stable process serving requests on demand from some API. The design of DAP implies to decide their aims, for example, for implementing specific algorithms, or making predictions, anomaly detection, suggestions, statical analysis, clustering, recognition, counting, classification, object detection, KPI estimation, optimization, conversion, etc. Most of these aims can implemented by using techniques as ML, AI, XAI, NLP, LLM, operating research, statistics, etc. To this end they would need to exploit a set of libraries for Python or RStudio to produce a model (in a training phase) which in turn has to be saved to be later exploited in execution/inference. Python and RStudio platforms may exploit any kind of libraries such as Keras, Pandas, and hardware accelerator as NVIDIA to use Tensor Flow, and clusters of CPUs/GPUs, HPC infrastructures, via ClearML, MLOps, etc., exploiting Kubernetes or other solutions.

Moreover, in order to get data, the DAP in Snap4City can access to any kind of storage from external services and can access to the Snap4City KB (knowledge base, service map) and Big Data store. In that case, the access to Snap4City data is GDPR compliant and thus respect the privacy, the data licensing by using authenticated Smart City APIs, via some Access Token as explained in the **Development Life Cycle** Manual referred in the cover. The platform allows the access to historical and real-time data, and permits to save the resulting data produced by the algorithms, for example, heatmap-related predictions, the assessment of data quality, traffic flow data, ODMs, labels of detected anomalies, LLM requests and results, etc., also using some specific APIs.

- **Development Life Cycle: https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf**

For the analysis details are reported in the **Development Life Cycle** and for a **DAP** one should identify:

- What process must be implemented by the DAP?
- Which data models would be consumed and produced?
- Which data are needed, to be consumed?
- The DAP to be implemented is for training or for production?
- How many users are going to exploit the DAP at the same time? How many executions per minute or per day, the usage is sporadic or periodic?
- How many processes for production I am going to have at the same time?
- How big would be the AI model to be loaded for execution the first time?
- From where the DAP is expected to be called, from a Dashboard/view? or simply from a back-office process as a MicroService?
- Which is the expected execution time?
- Which is the expected precision, and which is the state of the art?
- Do I need to execute the DAP exploiting special hardware as NVIDIA since I am going to use CUDA, tensor flow, …?
- How much GPU I need?
- How much memory to load the AI model I need?

How to proceed to design the single DAPs according to its nature?

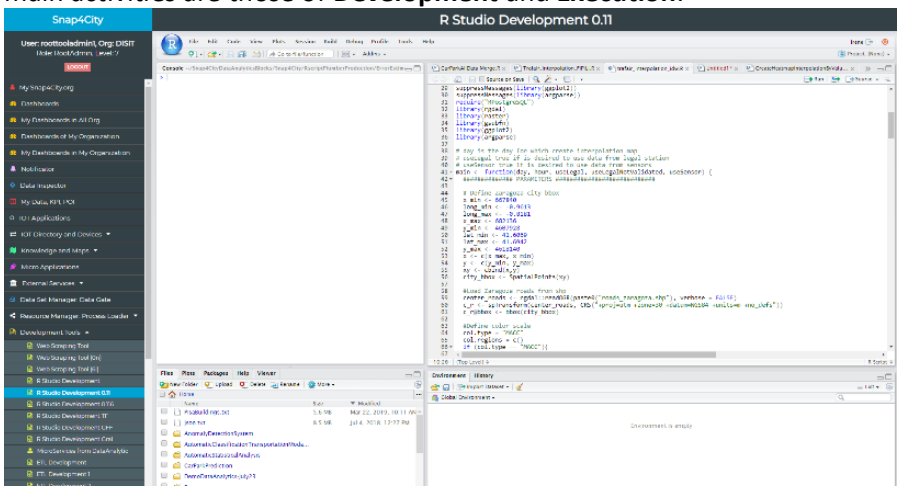
Here in the following the most relevant tasks summarized, just to recall you the main aspects to be addressed:

- Problem analysis, business requirements.
- Data Discovery, Data ingestion, data acquisition (as above presented that can give for granted), data access from Snap4City platform or from other sources.
- Data set preparation, transformation, identification of features, normalization, scaling, imputation,

feature engineering, etc., eventual data ingestion to the Snap4City platform by using Proc.Logic or Python and then storing data in the storage. The process of feature engineering may be performed by also using a PCA (principal component analysis or other techniques), or directly performing the first training and assessing the relevance of the features (for example with some RF, or using some XAI or other tech.), may be discharging those less relevant.

- Target Assessment Model Definition (mandatory to assess the precision of the results, the quality of the solution produced)
  - Identification of metrics for the assessment of model results, KPI.
  - Typically: R2, MAE, MAPE, RMSE, MSE, MASE, MAPE, …
- Screening on Models/Techniques, for each Model/Technique or for the selection Model/Technique perform the
  - Model/Technique Development/testing
  - Performing for each of them some hyper-parametrization, take care about the range of parameters with respect to the obtained best value of each of them;
  - Defining a good loss KPI according to the target results to be achieved and controlling the loss trend.
- Best Model/solution selection among those tested
  - If needed reiterate for different parameters, features, etc.
  - Comparison with state-of-the-art results.
  - Needs of Explainable AI solutions: global and local.
- Deploy best Model/solution in production, monitoring in production. In this phase assumes particular relevant:
  - Assessing performance and resources needed: GPUs, Memory, storage, etc.
  - Security of data and solution
  - Scalability of the solution, in terms of multiple users requesting the same computation,
  - multiple requests of the same computation but working on different spatial area, such difference cities, KB, maps, graphs, time series, etc.

In conclusion, the main activities are those of **Development** and **Execution**.



Figure– Data Analytics development in R-Studio, similar to Python which is in Jupiter HUB

In Snap4City, the activity of data scientists is strongly simplified since the Data Analytic can be implemented in Rstudio or Python exploiting processes that are executed on:

- **A) Dockers Containers** accessing and controlling them via some API, and these can be automatically produced and manage by the platform. The management is typically performed by some Proc.Logic (IoT App). The container is automatically allocated on cluster and maintained alive to be used. In this case, containers are allocated on Marathon or Kubernetes and may exploit the GPU/CPU according to the configurations. They are usually allocated dynamically, and they are moved from one VM to another by the

container manager.

- **B) Dedicated servers (for example for developers, or for exploiting high performance boards as the NVIDIA, in the lack of VGPU on your cloud resources)** and leaving them to access to the storage for using the data and providing results via Snap4City API, in authenticated and authorized manner.
- **C) Snap4City platform with MLOps Support and its integrated DAP Container Manager.** This case represents the most advanced solution for dynamic DAP **development** and execution**.**

3.9.1 - Cases A and B: Static Allocation on Container or Servers

In A) and B) cases, Python and/or RStudio cases, the script code has to include a library for creating a REST Call, namely: Plumber for RStudio 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 the Snap4City cloud environment since it is the best 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", which also allows the developers to perform queries and retrieve source code made available by other developers.

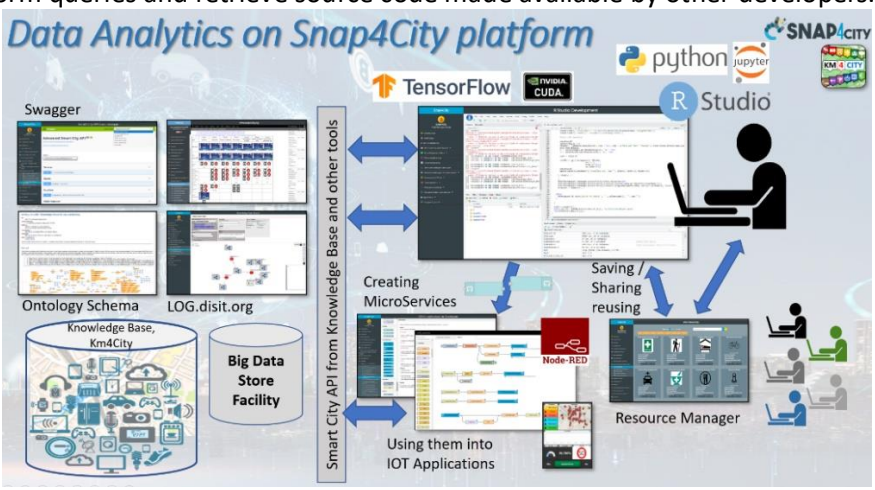
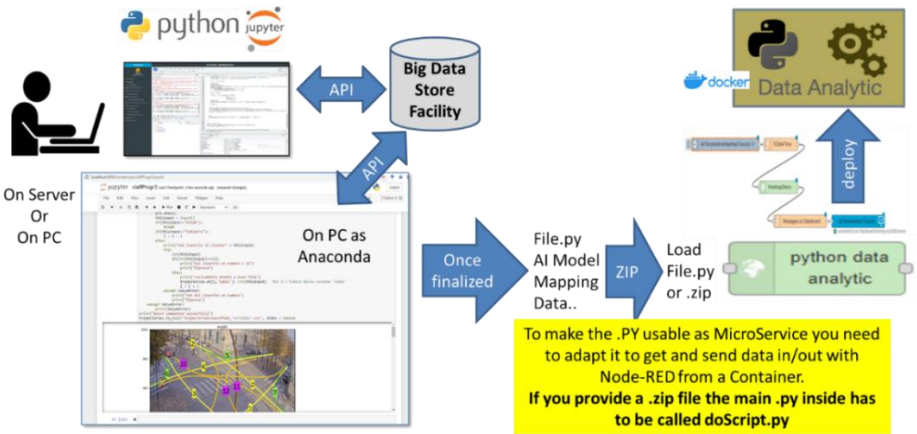


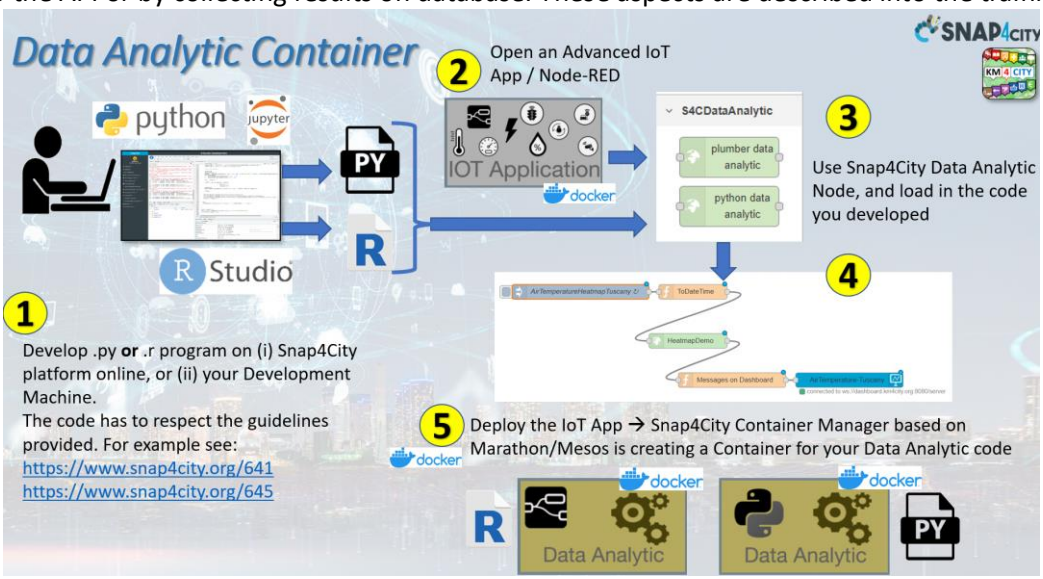
Figure – Schema of Data Analytics (ML, AI) development to be used as stable Containers (exploiting CPU on cloud)

Rstudio and Python data analytics processes may include conceptually any kind of libraries for ML, AI, operative research, simulation, etc. The Development environment may be configured to allow at the single operators to load their own preferred libraries. On the other hand, when the process is adopted to produce a contains as in the next figure, the container has to include the library used in the code. This can be performed by requesting a specific image to the platform manager and indicating the library you would like to have on Container executions.

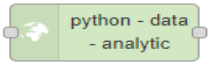


Figure– Data Analytics development flow in Python.

This description of the flow refers to case in which the Python or Rstudio are created to be used as MicroServices as contained from the Proc.Logic / IoT App. An alternative is to develop the Data Analytic to be used as standalone services, working on API, or providing some REST Call, and thus usable from Proc.Logic / IoT App according to the API or by collecting results on database. These aspects are described into the training course.



Figure– Data Analytics development flow in Python and integration into Proc.Logic / IoT App.

- The developer can develop the solution on their Computer as well
  A. a stable container on CPUs via Node-RED, Docker (with Snap4City Node-red library for developers), with node: 

  B. a some server with GPU/CPUs (directly manual operation)

In Snap4City, there is a **specific tutorial for the Data Analytic** development with several examples:
https://www.snap4city.org/download/video/course/p4/
Read the mentioned slide course and/or platform overview to get a list of Data Analytics in place:
https://www.snap4city.org/download/video/Snap4City-PlatformOverview.pdf
We also suggest reading the Snap4City booklet on Data Analytic solutions.
https://www.snap4city.org/download/video/DPL_SNAP4SOLU.pdf

3.9.2 - Case C: Exploiting MLOps: ML, AI, exploiting cluster CPU/GPU, HPC

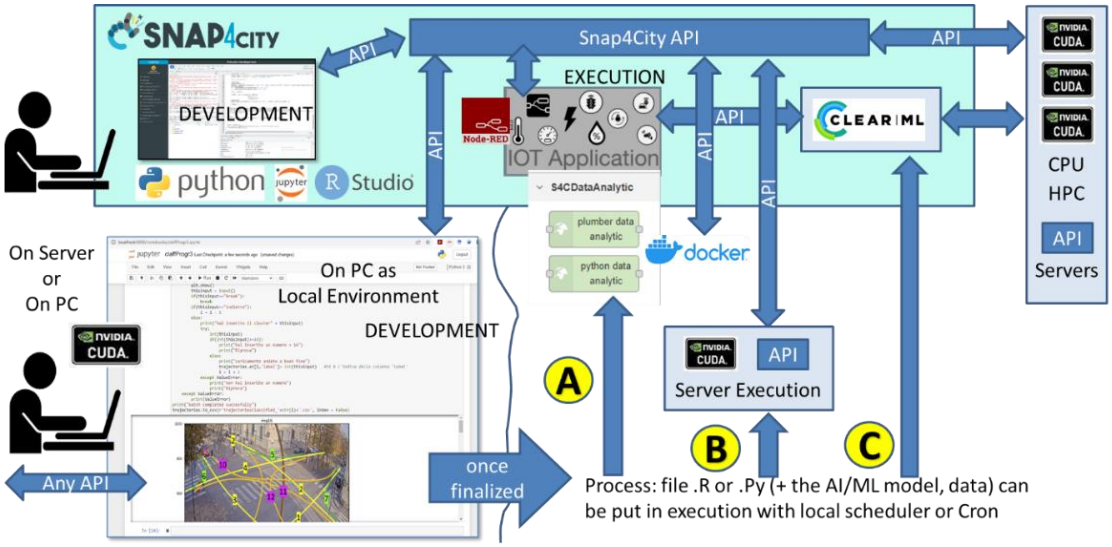
In this case C) of the above list, the solution provided by Snap4City includes the support for MLOps, Machine Learning Operation. In Snap4City, the MLOps is provided by using a custom version of ClearML tool, using a Jupiter HUBs for Python to develop DAPs, using Kubernetes or on former Marathon and Mesos or exploiting direct clusters of CPU/GPU and HPC, or combination of the Kubernetes and the latter. The access to this facility to the DAP developers is provided in any Snap4City platform by the RootAdmin to AreaManager role of users or higher. Snap4City.org is the most complete platform with clusters of CPU/GPU and HPC with both Marathon and Kubernetes; and others. Thus, the management of AI/ML operation/development via MLOps, implies to perform: training and execution

- Training needs: several processes
  - with different parameters and models to be trained, validated and test in batch to find the best results wrt metrics
  - High computational costs, time consuming if the processes have to be sent on GPU/CPU manually
- Execution needs, single executions in most cases singularly cheaper, but expensive for large volume of executions:

- periodically as predictions (saving time if the model is loaded permanent)
- on demand as optimisation, clustering, etc. (loading model, burning time)

In snap4City, the training processes can be performed:

- On Jupyter HUB provided by Snap4City in Python using ASCAPI, in this case the Jupyter HUB can be on CPU or CPU/GPU server
- By using ClearML for the Training and/or Execution, on Cluster of GPU/CPU
- Only Jupiter HUB of Snap4City can access to ClearML and Cluster of GPU/CPU
- The access at ClearML facility has to be authorized by Snap4City Administrator
- On Jupyter HUB provided by Snap4City in Python using ASCAPI, in this case the Jupyter HUB can be on CPU or CPU/GPU server
- On Jupyter HUB in Python using ASCAPI, in this case the Jupyter HUB can be on CPU or CPU/GPU server, not provided by Snap4City, not accessing to CPU/GPU of Snap4City
- On your computer in Python using ASCAPI, not accessing to on cloud CPU/GPU of Snap4City.



The developers can create their AI models using Snap4City data and infrastructure (Jupiter Hub) to put them in execution using the Snap4City Nodes for ClearML in this library, which allows to put in execution the models on a process managed by ClearML on some cluster of GPUs/CPUs, HPC

The **Execution on production** has to guarantee support for:

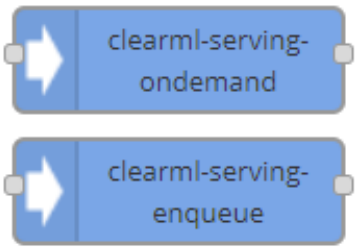
- Security of data and DAP solution access, permitted only to A&A users. Also in this case, the developer, working on Jupyter Hub, can send the code to the MLOps only by using its specific credentials and IDs.
- Scalability of the DAP solution, in terms of multiple users requesting the same computation at the same time,
- multiple requests of the same time working on different spatial area, such difference cities, KB, maps, graphs, time series, etc.
- monitoring the resource consumption in the terms of memory, storage, and CPU/GPU clocks/percentage. Eventual early warning and alarms sent to administrator. Possible the accounting of resource consumed.
- Eventual block and removal of strange / non desired processes.

The Execution on production is enabled **by creating DAPs (with a modality described in the rest of this document) which can be called via some APIs (provided, made accessible) according to TWO Modalities**:

- **SPORADIC services for REST calls, which are allocated and executed every time they are called (former name Enqueue):** to call the API of a DAP which is created as a task and executed at every API call by the

MLOps according to the list of requests put on some queue. The DAP is allocated automatically on some server as temporary container and process (NVIDIA / GPU, clusters or classic CPU clusters, or HPC, or Kubernetes) by the MLOps manager just for the single execution.
    - This means that each DAP Execution includes the loading time on boards/memory, and that the DAP does not remain in memory, and the memory of the servers (CPU/GPU) are not permanently booked for that DAP, the Kubernetes container is deployed, executed and destroyed.
    - This approach is suitable for DAPs which are executed sporadically, and/or periodically for which the overhead time to put them in execution is acceptable with the respect to the time for computing and delivery of the response, and the period of execution.
- **STABLE services static allocated REST Calls performing the on memory and GPU/CPU board allocation once, at the firs Rest call (former name: OnDemand):** to call the API of a DAP which is created as a task into a container and load statically on the server (NVIDIA / GPU, clusters or classic CPU clusters, Kubernetes, HPC).
    - This means that at the first execution the time to load the process on memory of boards will be evident and may be relevant, suggest solution for LLM and large AI Models.
    - This means that, once loaded, the DAP is ready to respond to the API call since is statically (permanently) allocated on the execution server, occupying memory (of CPU mem, GPU video mem) and not the actual CPU/GPU, until is not called (wake up) via API.
    - This modality is particularly suitable to exploit DAPs which need a relevant time to be loaded and put in execution, thus making the usage of the **Sporadic modality** not viable. For example, the usage of a LLM needing 24Gbyte or more would need lot of loading time, with respect to its single execution time by using the OnDemand/Stable modality only in a few seconds. So that in this case, the Sporadic solution is not suggested.
- **On both these modalities**, Snap4City.org provides access to exploit a number of clusters of services and single servers in CPU/GPU with many kinds of NVIDIA boards: H100 NVL, GV 100, RTX 4090, RTX 3090, Titan XP, etc., and Kubernetes HPC based.

The Snap4City platforms with MLOps support, may expose APIs of the DAP in the two modalities of Enqueue/Sporadic and OnDemand/Stable which can be called in authenticated manner via API as well as via IoT App/Proc.Logic nodes, as reported on the right side.



The two nodes are accessible as a separate Node-RED library of Snap4City microservices: https://flows.nodered.org/node/node-red-contrib-snap4city-clearml which can be installed on any IoT App/Proc.Logic on cloud and on Edge.

Please note that, the DAPs accessible via **Enqueue/Sporadic or OnDemand/Stable** modalities can be called from external services as well if the API are mapped on **API Manager**, or statically mapped on some FW (not suggested). For security reasons they can be called only by using the current Access Token of the section for the user. This allows to access at the DAPs from CSBL and any Web Application which is developed according to the Snap4City Development model and CSBL approach on Dashboards and views. This approach allows to implement much smarter and dynamic business intelligence tools and smart applications.

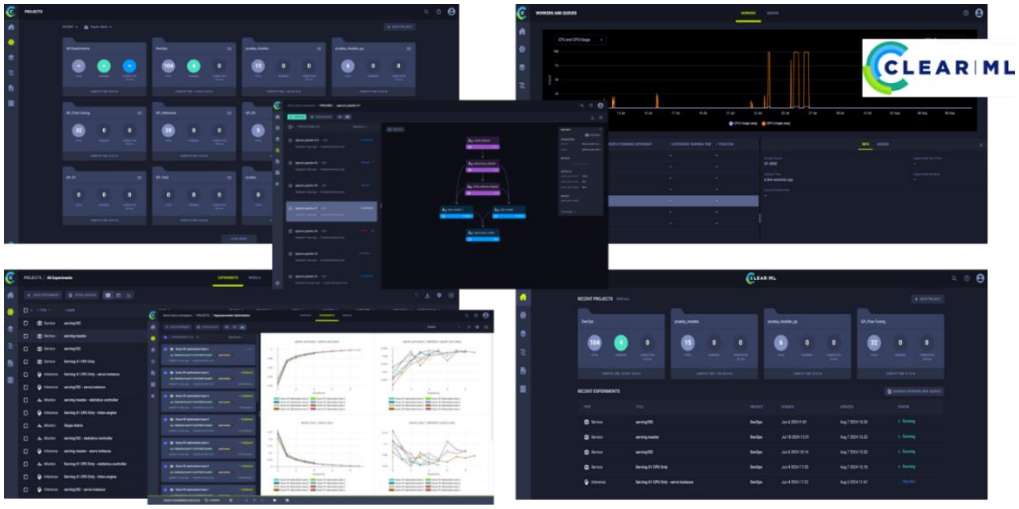
The request of analytic execution is performed: (i) on demand, which implies the allocation of a process on container allocated on demand on some cluster and perform a single execution; (ii) on a stable analytic process which expose some API to respond in fast manner to requests without any overhead of deploying a container and the process in memory.

ClearML main features are

- **Experiment Tracking:** Provides advanced features for experiment tracking, including automatic logging of metrics, output, source code, and the execution environment. This ensures that each experiment is

reproducible, and its results are easily shareable and comparable.

- **Data and Model Management:** Provides tools for efficient management of datasets and models, allowing for easy versioning, archiving, and sharing. Users can track model versions and easily associate them with corresponding experiments.
- **Integration and Compatibility:** ClearML is designed to integrate with existing development environments and tools, such as **Jupyter Notebooks, TensorFlow, PyTorch,** and many others, thus supporting a wide variety of workflows and technology stacks.
- **User Interface and Dashboard:** offers an intuitive dashboard that allows users to monitor the status of experiments in real time, view metrics and outputs, and manage resources and execution queues, all from a single interface.
- **Automation and Orchestration:** It allows the remote execution of experiments on any machine and distributes the tasks to be executed according to a system of queues and priorities. Also automating Hyper-parametrization via **Optuna.**



Read more on: https://www.snap4city.org/download/video/course/p4/

If you are interested to develop ML/AI processes with or without MLOps support, there is Python library which can be obtained only via subscription please contact snap4city@disit.org

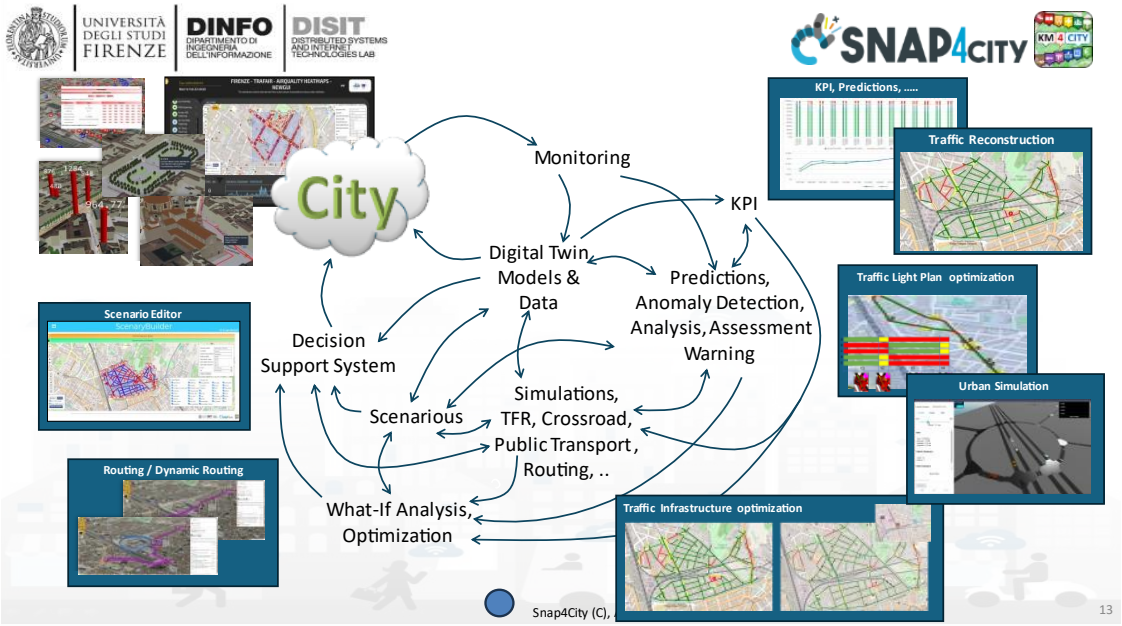
MLOps **for AI and DA development:**

https://www.snap4city.org/download/video/Snap4City-MLOps-Manual.pdf

3.9.3 - Snap4City Analytics, the smart tools at your disposal on Snap4City

The smart tools in Snap4City are focussed on providing support for Decision Makers to improve quality of life, match any specific KPIs, or SDG, assess the conditions for 15 Min City Indexes.

The smart tools are grounded on Data Analytics which are intended as solutions and tools for: predictions, anomaly detections, simulations, optimisations, production of suggestions, processing text for understanding complains and city users' needs, to enable what-if analysis and decision support tools for planning.



The following examples and those reported in the training course can give you an idea of the capability of the platform. We recommend that you browse the training course:
https://www.snap4city.org/download/video/course/p3/

Snap4 Technology has been created by DISIT Lab since the 2017 grounded on AI and respecting ethics and GDPR compliant, and end-2-end secure (passing the PENtest). In these years, DISIT Lab has developed a large number of solutions in the context of Smart City and Industry 4.0. Snap4 solutions fully support the development of real time data analytic processes through ML, AI, ethic trustworthy, XAI via languages such as Python, R-Studio, also exploiting Tensor Flow, Pandas, Keras, BERT, LLM, MLOps, and any kind of library for data analytics, ML and AI. DISIT lab via Snap4City, Snap4Industry, etc., is distributing a number of Open-Source and licensed AI solution and tools for: prediction, anomaly detection, classification, critical condition detection, constrained routing, optimization, analysis of demand vs offers of transportation, production of prescriptions and suggestions, and many others. Most of them have been published on international top-level journals and are tuned on demand on your cases. AI and smart application support are fully integrated into What-If analysis and optimisation tools in control rooms and for the operators, at the service and to simplify the solutions of complex scenarios. DISIT lab has a consolidated experience in the development, validation and transfer AI/XAI solutions (see course https://www.snap4city.org/944). Most of the DISIT lab solutions are based on ML, Deep learning, AI, XAI, LLM, natural language processing (NLP), sentiment analysis (SA), semantic reasoning and computing, neuro symbolic AI, ontological reasoning, generative AI, reinforced learning, federated learning, etc. In the following, a number of examples are listed, while more details can be recovered from the Snap4City course and from technical notes: https://www.snap4city.org/4 and a number of AI solution is accessible: https://www.snap4city.org/997

At page https://www.snap4city.org/997 a list of artificial intelligence Snap4City tools

Mobility and Transport Domain

One of the key factors on transportation infrastructure improvement is the effective traffic decongestion, reduction of CO2 emissions, reduction of travel time and of fuel consumption. The computing of traffic flow reconstruction is at the basis of a large number of traffic infrastructure computations and optimizations such as: routing constrained on traffic conditions, computation of crossroad timing, computation of the match from offers of transportation and the demand of mobility, and thus of what-if analysis and optimization. Snap4City is a platform designed for decision makers to perform activities of operation and plan in the cities and at the support of control rooms. The actions have been mapped on the activities of CN MOST (national center on sustainable mobility), PNRR, and by its Flagships OPTIFaaS, Leverage OPTIFaaS, Scalability SASUAM.

- **SASUAM**: Solutions for Safe, Sustainable and Accessible Urban Mobility
  - https://www.snap4city.org/999
- **OPTIFaaS**: Operation and Plan, Transport Infrastructure and Facilities Support as a Service
  - https://www.snap4city.org/1008
- **CN MOST**: https://www.centronazionalemost.it/

Traffic decongestion, reduction of CO2 emissions, reduction of travel time and of fuel consumption can be addressed with traffic optimization tools in which the functional is a combination of the mentioned goals. They estimation depend on the actual traffic flow distribution, the knowledge about the graphs and about the typical distributions at the crossroads. The scientific research towards mitigating traffic congestion focuses on multiple aspects and they can be classified into two main categories. Solutions that (a) work within the existing road infrastructure such as navigation routing algorithms for trip route optimization, public transport services planning, and (b) those that aim at modifying the existing transportation infrastructure itself (changing the road graph structure as well as the semaphore cycles). Solutions (a) in support of the infrastructure may impact specific users (public transport, drivers that use routing applications, etc...), impacting overall mobility indirectly. Modifying the infrastructure, as in (b), would require investments from cities and would impact would affect all road users.

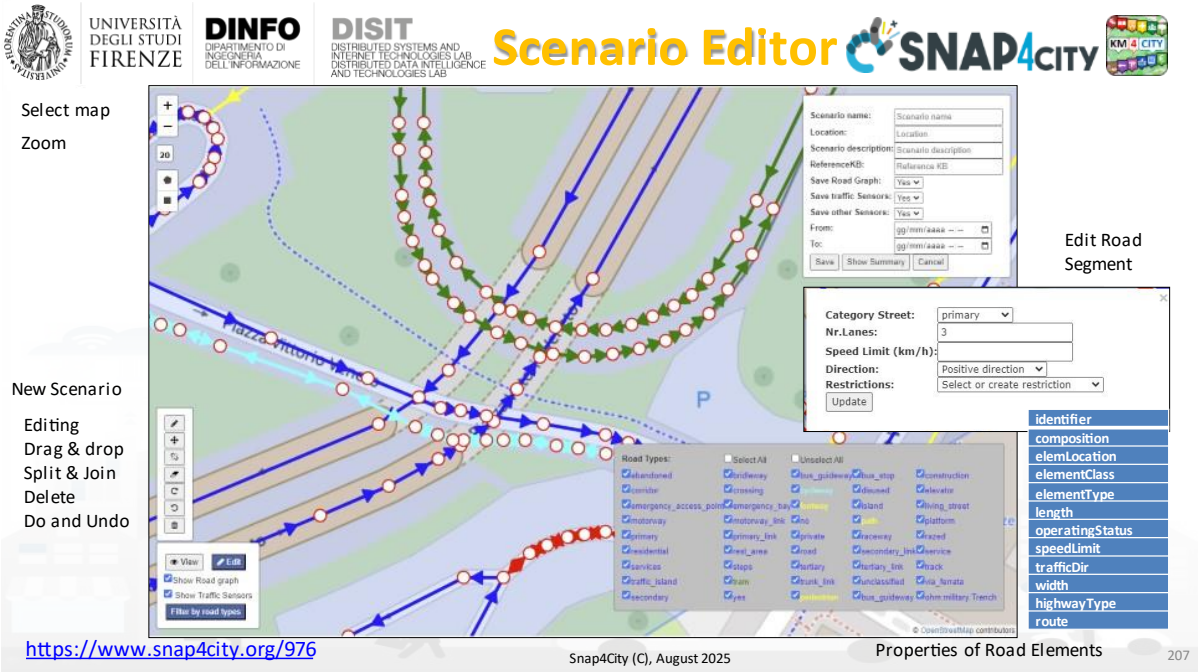
Urban traffic congestion and derived pollution represent critical challenges for sustainable city development. Snap4City presents a novel approach to optimize urban mobility infrastructure using AI as reinforced learning

and/or stochastic relaxation algorithms, which can be regarded as probabilistic reasoner to support decision makers. The method is designed to address the growing challenges of reducing traffic congestion, fuel consumption, and CO2 emissions in rapidly urbanizing cities. The system automates the generation and evaluation of road network modifications, drastically reducing the time and effort required for traditional simulation-based approaches. By utilizing real-time traffic data and traffic flow reconstruction algorithms within the Snap4City platform, the proposed system allows for the efficient exploration of "what-if" scenarios, and providing optimization support on the bases of key performance indicators such as travel time, fuel use, and CO2 emissions. A case study conducted in the city of Florence, Italy, demonstrates significant improvements in traffic conditions, fuel efficiency, and emissions reduction, underscoring the system's potential to enhance urban mobility in a sustainable manner. This system has been developed exploiting the Snap4City platform Scenario Editor and platform.

The solution automatically produces/computes a near-optimal solution in terms of suggested changes by minimizing the KPIs selected, and provides support to:

- specifies a scenario with geographic scope, designs the boundaries of the urban area, associates specific traffic data, period of observation, making changes if needed, etc.
- generates traffic data in all segments and time slots of any scenario (manually defined or generated).
- specifies and computes a set of KPI objectives: fuel consumption, congestion levels, CO2 emissions, and travel time over specific paths in the scenario.
- specifies constraints, such as: (i) roads to remain unchanged, (ii) changes on traffic flow data, to assess the effects of more critical conditions.
- optionally specifies the max number of changes that can be made by the optimization process to maintain any operational feasibility.
- supports the man in the loop, the operators may propose road changes. For example, starting from a modified scenario with additional changes with respect to actual conditions.
- generates a set of feasible improvements / changes to the road network and adhering to constraints and regulations, in agreement to the KPI chosen.

In the following figure, an example of the Snap4City Scenario Editor (Scenary Builder) which is accessible on all recent instances of the Snap4City platform. Read more on Scenario Editor on: https://www.snap4city.org/977



The framework proposed based on AI as Deep Reinforced Learning explores the solution space towards a near-optimal solution in a computationally efficient way. Based on the need for this type of problem of limiting the number of changes in the transportation infrastructure to maximize the investments in terms of roads in congested states at peak hours, fuel consumption, and CO2 emissions, the system has been developed to be able to propose a modified scenario on a limited number of changes. The results reported a limited number of changes set to 4 on a middle urban city scenario improvement in terms of fuel consumption and CO2 emission in the range from 20% to 30% modifying some roads configuration (see the following figure, on the left the original conditions, and on right the solution with proposed changes. The changes are on roads marked with dashed lines). The solution has been designed to assess also the travel time on some specifiable main roads and enables the operator to set arbitrary unchangeable constraints to be flexible based on the requirements specific to the case study. The solution with the limited number of changes set to 4, proposed a road infrastructure configuration relaxing the initial solution towards improving the efficiency and environmental impact of the urban scenario considered.

- **Goals:**
  - decongestion, decarbonization, costs reductions, improve accessibility to services, improve Security/Safety of city users.
- **Solutions for Operation (monitoring, managing, mobile apps, digital signages, control rooms)**
  - Monitoring traffic, parking, people flow, services, boats, ports, beaches, etc.
  - Early detection/warning of critical conditions: traffic, congestion, security/safety
  - Managing Smart Parking, transportation services, fines, etc.
  - Managing fleets: personal, sharing, waste collection, maintenance, etc.
  - Managing E-sharing, pooling services, MaaS, etc.
  - Managing entrances in city areas: restricted areas, touristic busses, etc.
  - performing routing, multimodal routing, routing for operators, routing for public transport
  - Production of suggestions, recommendations, nudging
  - Computing predictions of any kind
- **Solutions for Planning (optimization and what-if analysis)**
  - Reduction of traffic congestion, via optimization: traffic light plans, viability, routing
  - Reduction of Pollutant Emissions, via optimization: traffic light plans, viability
  - what if analysis, routing Traffic light optimisation
  - Optimization of transportation offers wrt multimodal mobility demand
- **Algorithms and computational solutions**
  - **Optimisation of viability of an area for reducing congestion, waiting time, stops**
  - **Optimisation of Traffic Light Plans, synchronization, in an area for reducing congestion, waiting time, stops**
  - **Predictions** for: traffic flow, smart parking, smart bike sharing, people flows, etc. (ML, DL)
  - **What if analysis**: routing, traffic flow, demand vs offer, pollutant, etc. (Simulation + ML)
  - **Traffic flow reconstruction** from sensors and other sources (simulation + ML)
  - **Public Transportation**: Ingestion and modelling of GTFS, Transmodel, NeTEx, etc. (DP)
    - Analysis of the **demand mobility vs offer transport** of according to public transportation and multiple data sources (Simulation)
    - Assessing **quality of public transportation** (analysis)
  - **Accidents** heatmaps, anomaly detection (analysis, ML)
  - **Road light controlled by traffic conditions**
  - **Tracking fleets**, people, via devices: OBU, OBD2, mobile apps, etc. (DP)
  - **Routing** and multimodal routing (multistop travel planning), constrained routing, dynamic routing (DA)
  - Computing **Origin Destination Matrices** from different kind of data (analysis, DP, DP)
  - Computing **typical trajectories** on the basis of tracks (analysis, ML)
  - Fleet management, monitoring, booking, allocation, maintenance

- Computing Messages for Connected drive (DP)
- Slow and Fast Mobility **15 Minute City Indexes** (analysis, DP, …ML)
- Computing and comparing traffic flow on devices and at the city border (analysis)
- **Typical time trends** for traffic flow and IoT Time series. (analysis, ML)
- **Impact of COVID-19** on mobility and transport
- Computing **SUMI, PUMS**, etc. (mainly DP)
- **Definition of Scenarios:** traffic, road graph, conditions, etc.

City User Behaviour/services, Tourism and Safety

- **Goals:**
  - Quality of Life, quality of services, over tourism mitigation, sustainability
  - Costs reduction of services
  - Accessibility to services: citizens, Tourists, commuters, etc.
  - Security/Safety of city users
- **Solutions for Operation (monitoring, managing, mobile apps, digital signages, control rooms)**
  - Monitoring services: tickets, reputation, usages, areas, etc.
  - Monitoring user behaviour (counting, trajectories): indoor/outdoor, hot places/services, ports, beaches,
  - Computing: origin destination, trajectories, travel means, reputation, predictions, etc.
  - Early detection/warning of critical conditions, connection with Video Management Systems
  - Managing entrances in city areas: restricted areas, touristic busses, etc.
  - Production of info-tourism, recommendations, nudging to city users and operators, second offer promotion
  - Providing Virtual Assistants for City Services, Tourist Offices, etc.
  - Monitoring reputation of services via: social media, blogs, etc.
  - Collecting complains, requests, participations from City users via mobile apps
  - Computing predictions of any kind: people coming/moving, services and sites reputation, advertising impact and people reactions.
- **Solutions for Planning (optimization and what-if analysis)**
  - prediction of the effect of certain changes on the offer;
  - Reduction of Pollutant Emissions, via optimization
  - Optimization plan to distribution of workload on multiple touristic offers/services, area cleaning, etc.
  - Predicting reputation of services, touristic and operative
- **Algorithms and computational solutions**
  - People detection and classification: persona, strollers, bikes, etc. (ML, DL)
  - people counting and tracking, head counting, people trajectories (via thermal cameras, ML, DL)
  - People flows prediction and reconstruction, (ML, DL)
    - Wi-Fi data, mobile apps data, Mobile Data, etc.
  - User's behaviour analysis, People flow analysis from PAX Counters and heterogenous data sources (ML, AI)
    - origin destination matrices, hot places, time schedule,
    - Recency and frequency, permanence, typical trajectory, etc.
  - Computing User engagement and suggestions for sustainable mobility (Rule Based, ML)
  - Social media analysis on specific channel, specific keywords: see Twitter Vigilance,
    - Reputation, service assessment: MultiLingual NLP and Sentiment Analysis, SA
    - Tweet proneness, retweet-ability of tweets, impact guessing
    - Audience predictions on TV channels and physical events, locations
    - Prediction of attendance of events and on attractions

- Virtual Assistant construction, LLM, NLP, Sentiment Analysis (DL, NLP)
- Video management System integration for security
- 15 Minute City Index, etc. (modelling and computability)
- Computing SDG, etc., (DP)

Environment, waste, land, etc., Domain

- **Goals:**
  - Reduction of emissions and EC taxations
  - Cost reduction for waste collection, reduction of waste collection impact on mobility
- **Solutions for Operation (monitoring, managing, mobile apps, digital signages, control rooms)**
  - Monitoring emissions, weather, waste, water, etc.: sensors, traffic, flows, ….
  - Early detection/warning of critical conditions on *emissions, weather, waste, water, fire, animals, …*
  - Early detection/warning of critical conditions for *landslides, water flooding, beach*
  - **Smart Waste Management**: bins/lockers, waste collection daily plan, pay as you throw, PAYT, etc.
  - Short terms prediction of emissions: CO2, NO2, etc.
  - Production of suggestions, nudging
  - Computing and predicting of long terms KPI indicators of the European Commission
- **Solutions for Planning (optimization and what-if analysis)**
  - Identification of main CO2/NO2 emissions locations in the city, total production from traffic
  - Reduction of Pollutant Emissions, via optimization: semaphore cycles, viability
- **Algorithms and computational solutions**
  - **Pollutant Predictions: short, long and very long-term** European Commission KPIs
    - NOX, PM10, PM2.5 pollution on the basis of traffic flow, 48 hours (ML, AI, DL)
    - Cumulated NO2 average over year (ML, AI, DL)
  - **Computation of CO2** on the basis of traffic flows (DP), computing emission factor (DA)
    - each road for each time slot of the day
  - **Prediction of MicroClimate** conditions for diffusion (ML, AI)
    - NO2, PM10, PM2.5, etc.
  - **Prediction of landslides**, 24 hours in advance (AI, DL)
  - prediction of **waste collection, & o**ptimisation of schedule and paths (DP, ML)
  - **Heatmaps production** dense data interpolation (DP) for
    - Weather conditions: temperature, humidity, wind, DEW
    - Pollutants and Aerosol: NO, NO2, CO2, PM10, PM2.5, etc.
  - **Impact of COVID-19** on Environmental aspects (DP)
  - Computing **SDG, SUMI, SUMP,** .. (mainly DP)
  - Etc.

Snap4Building Domain

- **Goals:**
  - increase efficiency, cost reduction, sustainability
  - Accessibility to services, Security/Safety
- **Solutions for Operation (monitoring, managing, mobile apps, digital signages, control rooms)**
  - Monitoring: usage, energy, environmental conditions, people flows, services, etc.
  - Early detection/warning, alarm, of critical conditions, notifications, decision support
  - Production of suggestions/prescriptions, nudging
  - Managing smart services: cabinets, dispenser, lockers, etc.
  - Global and local 3D/2D representations of area and buildings
  - Integration with Video Management Systems

- Computing predictions of any kind
- **Solutions for Planning (optimization and what-if analysis)**
  - Reduction of energy costs via optimization
- **Algorithms and computational solutions**
  - **Digital Twin for monitor, control and manage distributed infrastructures**
    - 2D/3D representations of the whole set of buildings, BIM modelling
    - Entities (building, floors, rooms, parking, charging stations, gates, etc.) with their shapes and descriptors, and data monitoring the allocation to office, meeting, cafeteria, storage, stairs, elevator, etc.
  - **Monitoring and computing KPIs on real time for**
    - **energy** consumed or produced (hot/cold)**, parking, logistic, presences, cleaning, air quality, departments, subareas, maintenance, etc.**
    - **allocation/designation**, dispositions, heating, cooling, temperature, equipment, etc.
    - **grouped in Zones**

Energy Domain

- **Goals:**
  - Energy consumption reduction, increment of efficiency, sustainability
  - improve area and building sustainability
  - improve accessibility to services
- **Solutions for Operation (monitoring, managing, mobile apps, digital signages, control rooms)**
  - Monitoring energy consumption (heating, cooling, prod.,..), conditions, charging stations, etc.
  - **Managing Smart Light** for city: dimering, programming, traffic control, controllers, legacy, etc.
  - Early detection/warning, alarm, of critical conditions
  - Managing smart services: cabinets, lockers, etc.
  - Production of suggestions, nudging
  - Global and local 3D/2D representations of area and buildings
  - Managing Communities of Energy, certification via Blockchain
  - Computing predictions of any kind
- **Solutions for Planning (optimization and what-if analysis)**
  - Reduction of energy costs, via optimization
  - Identification of roofs with better orientation
  - Optimization of battery storage size for PV plants
  - Community of Energy planning and viability
- **Algorithms and computational solutions**
  - **Monitoring Energy Consumption in single building, area and per zone**
  - **Smart Light management**, unicast and multi cast management, smart light controlled by **traffic flow data**
  - Monitoring Energy provisioning on **recharging station**
  - **Matching Energy consumption with respect to the actual usage**
  - **Computing Roof orientation for Photovoltaic installations**
  - **Optimisation of Photovoltaic installations** to identify the best parameters of size and storage
  - Collecting and managing **Communities of Energy**
  - Computing **KPI**
  - Etc.

Assets Control Domain

- **Goals:**
  - Costs reduction, increase service availability, risk reduction,
  - Improve quality Level

- **Solutions for Operation (monitoring, managing, mobile apps, digital signages, control rooms)**
  - Monitoring:
    - **Assets**: switches, Wi-Fi, servers, UPS, sensors, building, TV Cams, etc.
    - **Energy**: consumption, operative conditions, etc.
    - **Production**: continuous quality analysis
    - Etc.
  - Early detection/warning, alarm, of critical conditions
    - **Multichannel** Event reporting: email, Telegram, mobile apps, SMS, etc.
  - Managing maintenance operation
  - Computing predictions of any kind
- **Solutions for Planning (optimization and what-if analysis)**
  - Reduction maintenance costs, reduction of critical SLA conditions, improvement of quality level

Industry production Domain

- **Goals:**
  - Cost reduction, increase control on production, Production optimisation
  - Quality Level
- **Solutions for Operation (monitoring, managing, mobile apps, digital signages, control rooms)**
  - Monitoring KPI: administration, production, commercial, faults, etc.
  - Early detection/warning, alarm, of critical conditions
    - **Multichannel** Event reporting: email, Telegram, mobile apps, SMS, etc.
  - Managing maintenance operation, predictive maintenance
  - Computing predictions on KPI
  - Computing predictive maintenance
  - generation of patterns in production, design, etc.
  - solving complex physical equations with Physically Informed Neural Networks, PINN, for Simulating fluid-dynamic flows in autoclaves and machineries, cost reduction, time reduction.
- **Solutions for Planning (optimization and what-if analysis)**
  - Generative AI and predictive AI for production plan optimisation
  - Reduction maintenance costs, reduction of critical SLA conditions, improving quality level

They are developed by using a large range of statistics, operating research, ML, AI, XAI techniques such as:

- Predictions, classifications, anomaly detection
  - RF, XGBoost, BRNN, RNN, SVR, MLP, etc.
  - DNN, LSTM, CNN-LSTM, Bi-LST, Transformer, Autoencoders, BERT, …, YOLO, GNN, etc.
- Tech: Reinforced Learning, PINN, Federated Learning, Continuous Learning, etc.
- Generative AI: LLM, LLM RAG, Graph RAG LLM, etc., see SnapAdvisor
- Clustering: K-means, K-Medoid, k-Shape, etc.,
- Optimization: Reinforced Learning, Simulated Annealing, Genetic Search, Taboo Search, etc.
- XAI: Shap, LIME, Gradient, Integrated Gradient, variations

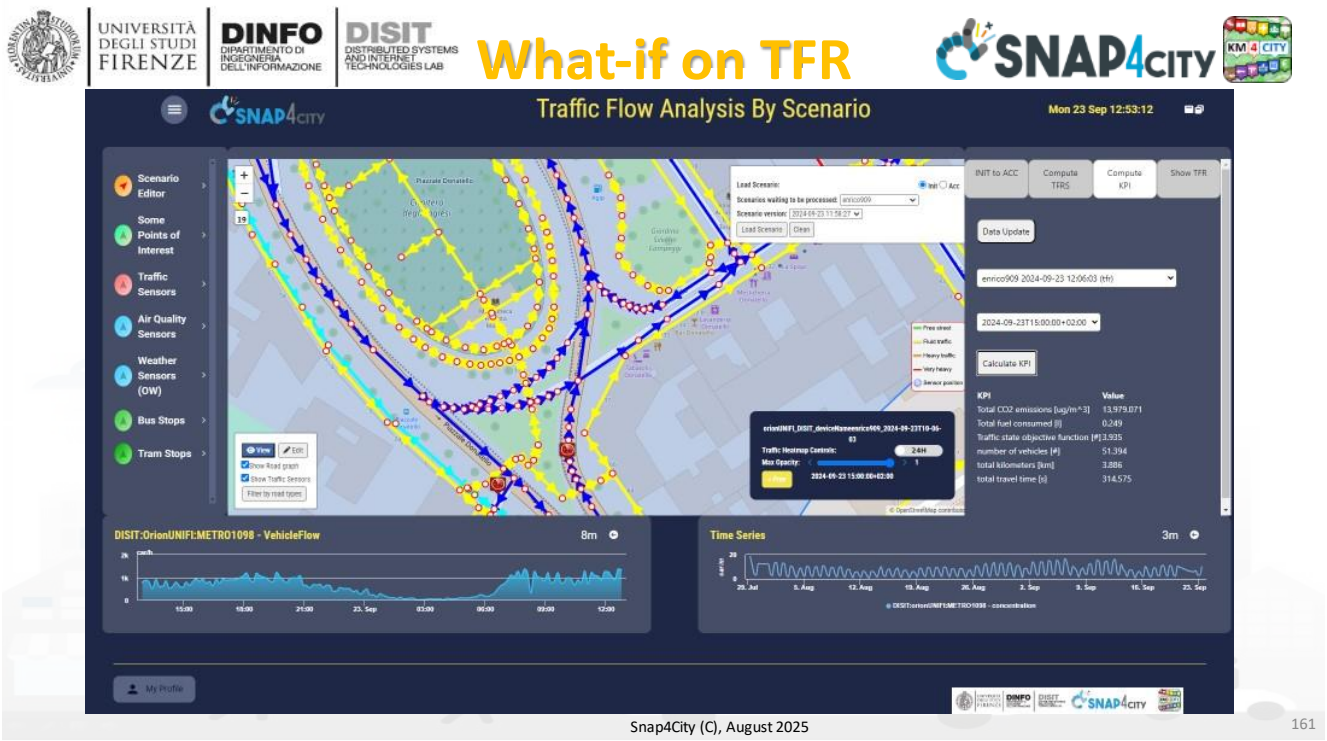
Data analytics have to be targeted to the **user kinds** (decision makers, ICT official, Operators, city users' kinds, …) and the device in which they are exploited. **For devices,** we intend a video wall in the control room, situation room touch screen, web pages, mobile apps, tablets for operators, etc. So, in the Snap4City platform, you can find, examples and tools to comply with those aspects. For example, the image aside reports the mapping of DA vs the Antwerp and Helsinki users and the main tools in which they have been made accessible in a transparent manner for the user, in the 2019. So that most of the above-mentioned and following AI tools were not invented yet.

	Antwerp					Helsinki								Where				
	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	Main Data Sources
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	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
Comparsison: 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				Sensors data, OSM
The most used services		x		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

3.9.4 - What-IF Analysis and elements for Simulation

The What-IF analysis is a modality of work included in the Snap4City suite which aims to exploit in a multi-contextual environment most of the former tools such as dashboards and Data Analytics, business intelligence with IoT App intelligence.

What-If analysis Support in Snap4City is not a single tool but a modality of work that enables you to create several solutions for city analysis depending on your needs and context. The Snap4City What-IF analysis tool is capable to answer questions such as: ***What is going to happen at certain City Services if certain conditions/cases are going to occur?***



Snap4City (C), August 2025 161

The What-if analysis process is following many steps:

- **Create/save/share Scenarios** based on present or future Conditions/cases, Services, aiming at
- **Producing/saving/sharing 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 at:

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

Simulators in Snap4City Digital Twins can be used in What-If analysis, plan and Optimisations
https://www.snap4city.org/1053

- **Mobility and Transport: infrastructure, data, TLP, by Scenario or Wall City/area**
  - Traffic flow reconstruction: from sensors to full traffic flows (Meso, Macro) (Snap4City)
    - https://www.snap4city.org/1078
    - https://www.snap4city.org/1022
  - Multimodal Traffic: cars, tram, busses, pedestrian (SUMO micro-meso + Snap4City meso-macro)
    - https://www.snap4city.org/1084

        - also for generating traffic flow data, for optimisation, dynamic routing, optimisation of semaphores, etc.
      - https://www.snap4city.org/1022
    - Public Transport: match demand vs offer (Snap4City:DORAM1, DORAM2 beta + SUMO: micro-meso-macro)
      - https://www.snap4city.org/1084
      - https://www.snap4city.org/633
      - https://www.snap4city.org/636
    - Routing, dynamic routing: micro, meso, macro scales (GraphHopper + Snap4City)
      - https://www.snap4city.org/506
      - https://www.snap4city.org/531
    - Multimodal routing: micro, meso, macro scales (GraphHopper + Snap4City)
      - https://www.snap4city.org/506
    - Road Parking simulator and management system (Snap4City)
      - https://www.snap4city.org/1013
      - https://www.snap4city.org/802
    - KPI: travel time, emissions, waiting time, etc. (Snap4City, SUMO)
      - https://www.snap4city.org/1078
      - https://www.snap4city.org/1084
- **Environment**
    - Impact on NO2, CO2 by traffic flow, heating emissions, wind, weather, etc. (Snap4City)
      - https://www.snap4city.org/500
    - Waste Collection Simulator and Management system (Snap4City)
      - https://www.snap4city.org/1042
      - https://www.snap4city.org/982
- **Energy**
    - Photovoltaic plant: home level, area data (Snap4City)
      - https://www.snap4city.org/940
    - Computing roof orientations and patterns from LIDAR data, positioning of PV panels (Snap4City)
      - https://www.snap4city.org/749
    - Smart Light management systems (by Snap4 srl, on top of Snap4City
      - https://www.snap4city.org/968
      - https://www.snap4city.org/1038
- **Other:**
    - 15 Min City Index: 13 subindexes, (Snap4City)
      - https://www.snap4city.org/652
    - SUMI indexes, partial; SUMP indexes, partial; SDG…; etc. (Snap4City)
      - https://www.snap4city.org/951
    - 3D Tiles from plant shape extrusions and patterning, Google 3D Tiles, working on other formats (Snap4City)
      - https://www.snap4city.org/749

SUMO is integrated into Snap4City. So that you can create data for SUMO and see simulation on Snap4City dashboards and views.

Other simulators can be added and put in synergic execution with those reported above, all running on Kubernetes/containers, and exploiting CPU/GPU clusters (on your servers or on HPC infrastructure, and/or exploiting ClearML MLOps) and eventually other services via any kind of APIs.

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

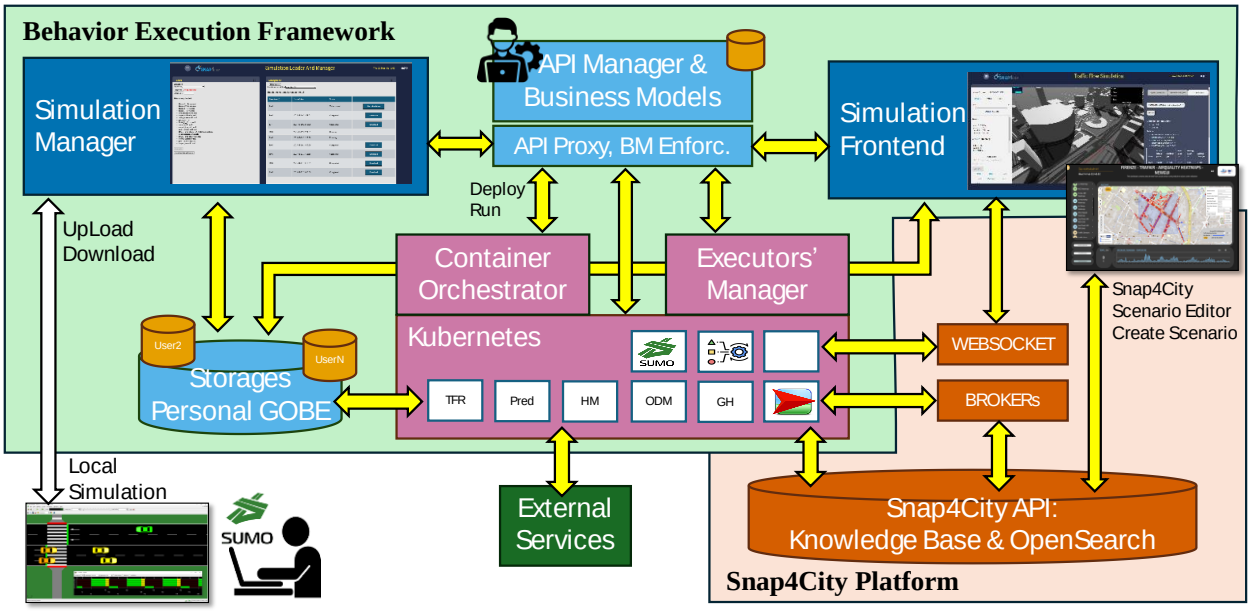
3.9.5 – Simulation Manager and Support

DAP and Simulation is a methodological approach for the analysis, design, and optimization of complex systems in domains such as industrial process design and urban mobility planning.

The Snap4City BEFDIT (**Behavior Execution Framework for Digital Twins**) framework addresses key limitations of most of complex digital twin behavioral supports through the following contributions by providing:

- **modular architecture** to support multiple DAP and simulation tools through a range of interfaces (e.g., API, WS, files), enabling co-simulation and co-execution of different tools which can contribute to the same digital twin behavior. It is a way to ensure seamless integration of heterogeneous behavioral components embedded in simulation scenarios, enabling dynamic interactions, agent-level intelligence, and feedback mechanisms across different simulation scales.
- **support for the integration** on the front-end of the simulations / DAP executors in "real time" and off-line via Web Interface. This point and previous are enabling solutions for co-working among experts on different fields/simulations models.
- **support for the scalable management** of the behavioral executions of DAP according to different business models in the range of "*as a service*" approach based on cloud/containers. Thus, opening the space to add other executors/simulations in the framework, supporting both open source and proprietary solutions. This is driver for reducing costs.

BEFDIT support the concept of **Behavioral Executors, BEX, or DAPs** are capable of moving vehicle agents, computing routing for each vehicle, computing pollutant diffusion, making traffic flow predictions, computing emissions and costs, computing thermal evolution in building, computing the energy produced by irradiation, etc. Groups of **BEX can be activated on demand.**



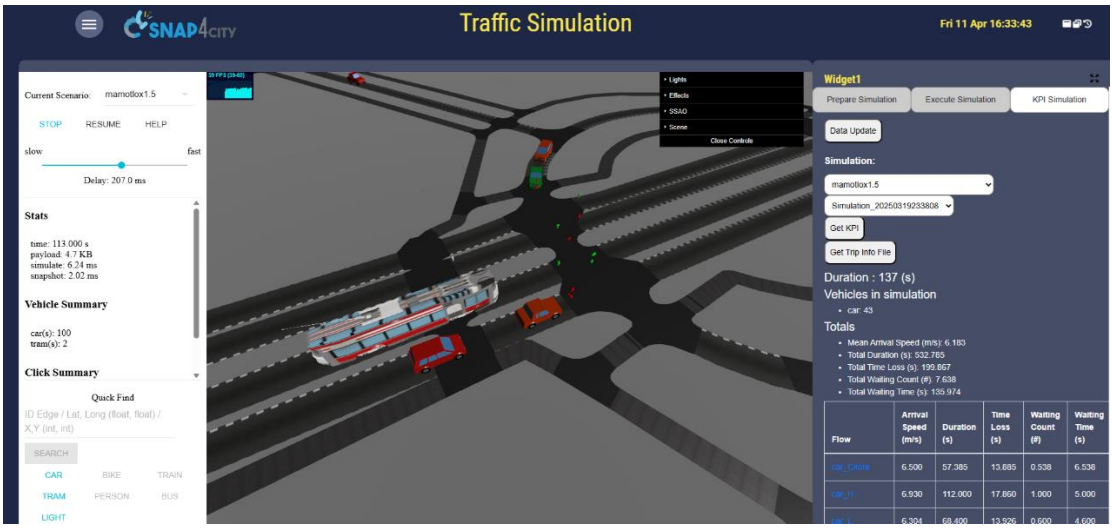
The solution enables users to configure and execute digital twins taking into account the behavioral aspects in the mobility and transport domain and it has been developed to extend the Snap4City open-source platform. **BEFDIT** consists of the following main components.

Simulation Manager, which manages the data regarding simulation as they are the setup and results. It provides a user interface to allow the upload of files coming from local preparation of simulation and of download of results.

Kubernetes, Container Orchestrator and Executors' Manager (to manage BEXs). This area provides support on managing containers on which any kind of simulator and executor (BEX) can be deployed and run (in our cases: SUMO, traffic flow reconstruction, routing engines, predictions, etc. Each of them, may exploit services such as: those provide by Snap4City API, any other **External Services** via any protocol or API, IoT/WoT (internet/web of things) brokers, WebSocket (the latter two: to provide real time data/updates from/to the front- and back-end); and may access to the user's personal storage.

API Manager & Business Models provide support for defining which API can be exploited as well as by front end simulations and tools, according to a range of business models.

Simulation Frontends is an exemplification of a large range of Snap4City Dashboards/Tools which can be built by exploiting API via Client-Side Business Logic in JavaScript, CSBL. Among the simpler front-end tools of Snap4City: the **Scenario Editor** of Snap4City which allows to extract/change a portion of the Digital Twin and save it on Snap4City storage, the **MultiDataMap** for rendering on maps/orthomaps: traffic, ODM, heatmaps, trajectories, sensor data, flows, vector fields, etc.; **Synoptic** frames in SVG; **3D rendering** tools for cities; **BIM viewers**; **Traffic Light Planer**; **Waste Management**, **Smart Light** management; **eShare** for car sharing; etc.



For example, BEFDIT allows the integration of simulated traffic data into the solution making it directly usable by existing tools for visualization, analytics, event management, or as input for new simulation and decision-support systems (see **Figure**). The use of simulated sensors enables several applications, including the validation of predictive models, testing of smart city services in controlled environments, provision of fallback data in case of real sensor failures, and support for digital twin configurations in urban mobility scenarios.

The BEFDIT is a scalable, modular, and interoperable environment designed to address the increasing complexity and fragmentation of simulation tools within smart city digital twin infrastructures, particularly for mobility and transport domains. BEFDIT successfully enables the integration and co-execution of heterogeneous behavioral models, supporting agent-based, discrete-event, continuous, and hybrid simulations, across multiple spatial and temporal scales. The framework's key contributions include: interoperability across diverse simulation tools through API-based communication and containerized deployments; Behavior execution as a service, leveraging business model integration for scalable and cost-effective simulation management; Real-time and offline co-simulation capabilities, supporting both human-driven what-if analyses and AI-driven optimization processes; Persistent, versioned simulation management, enabling reproducibility, scenario evolution, and extensive experiment tracking. BEFDIT's integration within the open-source Snap4City platform and its deployment in operational smart city contexts demonstrate its practical effectiveness. Use cases involving dynamic congestion management, real-time data generation, multi-simulator co-execution, and high-performance simulation orchestration highlight its adaptability, scalability, and robustness. Experiments and operational data collected over large-scale urban environments (e.g., Florence, Helsinki, Sweden, Greece) confirmed BEFDIT's capacity to handle millions of road elements and thousands of behavioral executions, with dynamic orchestration of cloud resources via Kubernetes clusters. These results validate the framework's capability to enhance smart city decision-support systems by providing tactical and strategic planning tools based

on automated, interoperable, and scalable digital twin simulations. Future developments will focus on expanding the ecosystem of integrated simulators and behavioral executors, strengthening AI-driven optimization loops, and further automating multi-domain what-if analysis processes. Additionally, efforts will be made to enhance multi-user collaboration features, privacy-preserving simulations, and broader exploitation across national and international smart city initiatives.

This result has been produced for the SASUAM scalability project, and for the OPTIFaaS Flagship project of CN MOST, the National Center on Sustainable Mobility in Italy (https://www.centronazionalemost.it/), and Snap4City as official infrastructure framework on which the experiments were conducted (https://www.snap4city.org). Snap4City is an open-source technology of DISIT lab.

This tool is accessible on Snap4City platform and some of its instances.

- This page: https://www.snap4city.org/1014
- **SASUAM**: Solutions for Safe, Sustainable and Accessible Urban Mobility
  - https://www.snap4city.org/999
- **OPTIFaaS**: Operation and Plan, Transport Infrastructure and Facilities Support as a Service
  - https://www.snap4city.org/1008
- **CN** MOST: https://www.centronazionalemost.it/

3.9.6 -Traffic light Optimisation

Urban traffic congestion is a critical issue due to increasing city populations and economic activities, traditional methods of traffic management have proven inadequate, particularly in complex and heavily trafficked areas. The primary objective of the Snap4City proposed solution has been to reduce congestion, in areas where tramways need priority by providing innovative traffic light optimization strategies. Thus, two novel approaches have been developed: (a) Multi Agent Reinforced Learning, (b) Genetic Algorithm-based Multi-Objective Traffic Light Optimization (MaMoTLO). They take into account multiple factors such as minimizing stops, reducing travel and waiting times (private and public), and ensures the synchronization of tramway schedules. These approaches are particularly useful in urban area where multiple tramways intersect with regular traffic, creating potential bottlenecks for private flows. The solutions proposed outperform and has been compared with existing state-of-the-art methods, including those based on Non-Dominated Sorting Genetic Algorithm II (NSGA-III), SUMO Actuated solutions, and Webster's formula for traffic light timing. The Snap4City solutions have been tested using real traffic data, and simulated scenarios to measure their effectiveness in reducing congestion and improving traffic flow on Snap4City open platform. The findings indicate that the Snap4City solutions outperform state of the art methods by providing a more balanced and efficient traffic light schedule, which is crucial for urban areas with high tramway traffic. The proposed optimization improves the flow of vehicular traffic and ensures that public transportation systems like tramways operate smoothly without unnecessary delays.

The traditional methods for traffic light management at crossroads can be essentially classified in:

- **Fixed-cycle** solutions follow a fixed cycle of green/yellow and red time durations (Traffic Light Plan, TLP) for each direction of traffic to prevent collisions and following normative.
- **Actuated solutions** adapt the cycles durations in real time according to local traffic density conditions of single junctions. This type of solutions exploits the possibility of measuring traffic density via sensors to take into account of number of vehicles which are present in the roads arriving at junction under control. This approach can be effective in certain traffic situations, while they are not usually capable to take into account the traffic status in the whole area, and of the arrival of the tramways with regular rates.

Both fixed-cycle and actuated methods may improve their performance by means of techniques such as:

- **Adjusted on Demand algorithms:** this add-on allows the traffic signal controller to receive requests about the presence of vehicles (ambulance, busses, police, etc.) or pedestrians and adjust the green duration (making it longer or red shorter) accordingly. This approach can reduce waiting times and improve traffic efficiency.

- **Traffic Prediction algorithms**: this add-on allows the traffic signal controller to use the previous traffic data to predict future traffic flows in the roads arriving at the junctions, as provided by Snap4City predictions tools.

The Snap4City family of solutions are based on Dynamic or Fixed-cycle approaches to produce optimized TLPs of synchronized connected systems/groups of junctions with and without adjusting within urban areas. They provide effective strategies to reduce traffic congestion, mitigating at the same time:

- **Combining Multiple-Objectives:** to generate optimal TLPs considering multiple objectives and constraints, such as: high priority of public transport travel time, reduction of waiting time, reduction of number of stops, and take into account priority in multiple traffic directions (reduction of travel time), supporting green waves, and mitigating congestion at intersections, taking into account the lengths of the queues and of the tramway. It uses of multi-criteria decision-making techniques to provide public administrations a flexible tool to adapt to variations in traffic or different situations and prepare TLP, which can be also Adjusted in real time.
- **Prioritizing public transportation and particularly tramway** should be given the priority to guarantee the travel time to a higher number of people and for sustainability, and thus to avoid collapsing the transport systems since the rate on rides is high (tramway presents irregular velocity in segments due to the time needed at stops), and to minimize the number of stop for tramways at the traffic lights.
- **Working at macroscale**: the solution offers the ability to select a specific urban zone, as well as the entire city, to identify the real or simulated traffic flow workload and to optimize traffic light systems and produce all the TLPs for all the included junctions. This functionality allows for satisfying specific needs of the areas under consideration.
- **Computing solution** by using real traffic flow and/or typical values: the solution may use real traffic density and the related average vehicle speed as well as typical values to consider traffic density in the road network over time.

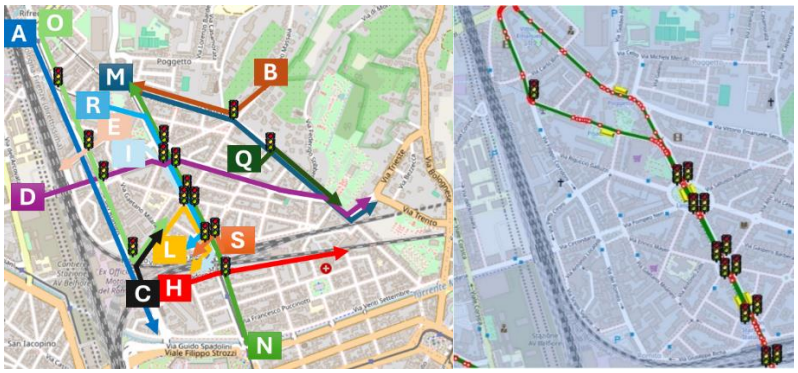


Figure: (left) Paths/directions in the study area of Florence, (right) paths of the tramways and corresponding traffic lights.

The Snap4City approaches incorporate innovative features (penalty, adjust, multiple constraints) and it have been demonstrated to produce traffic light plans which can be applied in areas in which the traffic light control does not monitor the traffic flows, and are not ready for the adjust, making the adoption of the solution cheaper and more sustainable. The Adjust on demand feature allows the control system to dynamically respond to real-time traffic conditions and the specific needs of tramway schedules. The solutions ensure that tramways can cross intersections without unnecessary delays, while the Adjust on Demand rules allow for real-time adjustments to traffic light cycles to accommodate sudden changes in traffic flow. These capabilities make the **Snap4City for traffic light optimisation framework** highly flexible, adaptive and robust, capable of handling the dynamic and often unpredictable nature of urban traffic in a large range of conditions.
The Snap4City solutions have been tested using real traffic data, and simulated scenarios to measure their effectiveness in reducing congestion and improving traffic flow on Snap4City open platform. The findings indicate that the new approach outperforms traditional methods by providing a more balanced and efficient traffic light schedule, which is crucial for urban areas with high tramway traffic. **Snap4City for traffic light optimisation framework**, is grounded on a composable multi-objective approach and real-time adaptability, provides a significant improvement over traditional traffic management methods. By prioritizing tramways and optimizing traffic light schedules, it enhances the efficiency of public transportation and contributes to create a more sustainable and liveable urban environment. The successful validation of the proposed solutions using real traffic data underscores their practical applicability and potential for widespread adoption in cities worldwide.
The solution has been capable to avoid stops for tramway lines (in the above described scenario), and to reduce the travel time, and waiting times of about the 7% with respect to the state of the art solutions.
The solution is based on Snap4City Scenario Editor which is accessible on all recent instances of the Snap4City platform. Read more on Scenario Editor on: https://www.snap4city.org/977

3.9.7 - Routing tools

The Snap4City Router with What-if functionality has been developed and enforced in Snap4City with the aim of providing a versatile and easy-to-use tool for finding routes among points on map. It also let user to study possible outcomes of making some changes in "real-life scenarios" context, trying to respond to the answer "What if ... ?". It's based on Graphhopper v7.0-pre2.

Red more on: https://www.snap4city.org/1106

The main functionalities of Snap4City routing solutions are:

- modal routing: private vehicles, bikes, pedestrian
  - with start, end and multiple intermediate points
  - selecting: shorter, faster, quitter, etc..
  - dynamic conditional routing taking into account the effective **traffic flow** status, or typical **traffic flow** status
  - dynamic conditional routing taking into account eventual blocked areas (by scenario) for example for street working, restoring, etc. (**what-if** cases and analysis)
- multimodal routing for the city users to walk and take the public collective transport
- modal routing for public administrations (ambulance, fire brigade, police, busses, etc.) exploiting the reserved lanes, etc.
- a combination of the above cases.
- Full **API** for the exploitation of your applications: **https://www.snap4city.org/1107**

In figure below is presented a screenshot of a Snap4City dashboard/application that let user to interact with these tools. As you can see, a map is used as benchwork while on the left and right sides of the map are placed interactive panels and buttons. All the functionalities are accessible only to registered users, and the registration is free of charge, some authorisation may be needed in some platforms.

Requirements (only for multimodal routing) Router needs some geographical and gtfs data to work with, provided from Snap4City. User needs only to pay attention to use the same context of that data to make things work. (e.g. if router works with Florence map and public transport, user should not search routes outside the city).

What-if while connecting two points, how can user avoid certain areas? How is the route modified? The first action that user can take is to open the scenario panel to draw on the map polygons that represent the to-be-avoided areas. They can be points, circles and polygons.

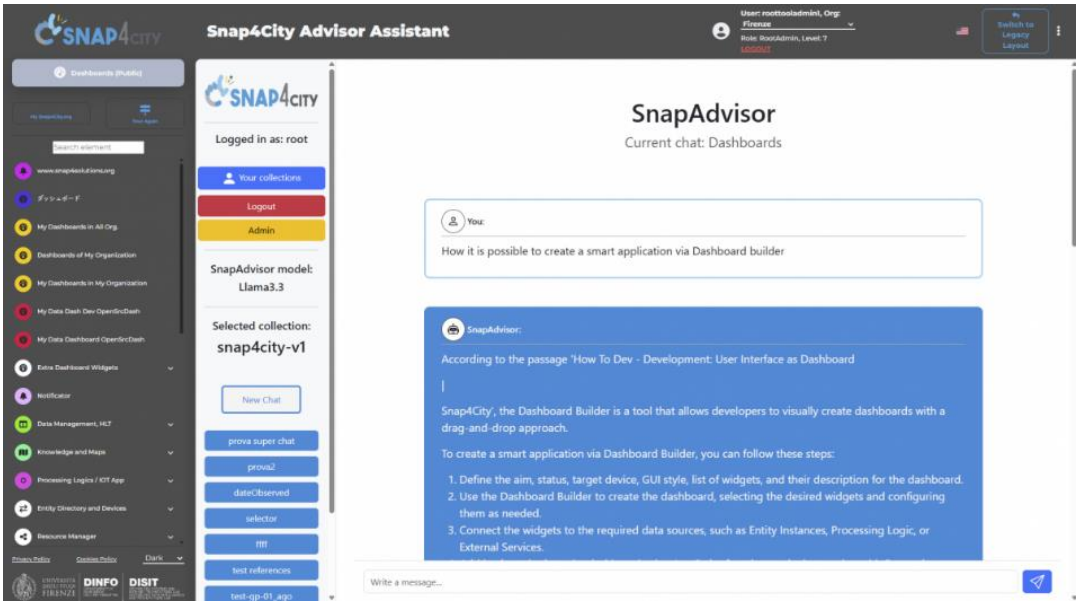
3.9.8 - SnapAdvisor: the expert at your disposal on Snap4City and your private content

SnapAdvisor of Snap4City is a multilingual chat-based virtual AI assistant which can be tailored on your domain by providing a content based on PDF documents, PDF slides, coding, and web pages. SnapAdvisor preserves the privacy since it uses the content you provide to perform temporary in the reserved chat you establish with it, and not use the data/files and chants to perform additional training. It can answer in your language, even if the content has been provided in other languages. SnapAdvisor can be classified as an Advanced RAG (retrieval-augmented generation) built by the DISIT Lab on top of Llama 3.3 Large Language Model, LLM, which can be put at work on other LLMs.

SnapAdvisor key advantages are:

- working on up-to-date content and thus producing answers pulls the latest docs records without retraining the model, great for policies, catalogues, dashboards, tickets, etc.;
- lower hallucinations: responses cite retrieved passages, keeping the model anchored to verifiable text, also providing text references and segments;
- domain control: you can decide the knowledge base (internal wikis, PDFs, APIs), easy to scope different "skills" by data set (which are called collections in SnapAdvisor terminology);
- costs & speed vs. fine-tuning: no expensive retraining, update results by updating the index, may also work with a smaller/cheaper LLM;
- explainability capability, which is very important for ethics and decision makers, but also for trainers and producing reports: the user receive answers with references to its own provided sources (for example "see document XXX at page 3…"), which helps users trust the output and prepare reliable reports;
- compliance & governance: it is possible to restrict retrieval to approved content, also apply access controls per user or team;
- personalization: retrieve user- or tenant-specific context (their contracts, past interactions) at runtime;
- multilingual via content: your corpus has multiple languages, retrieval naturally feeds the model the right language passages of the chat;

- modularity: it is possible to pass from one collection of documents to another, and multiple users can work on the advisor asking for different topic on different collections/domains at the same time, independently as needs change, without any interferences among them.



The direct access to SnapAdvisor is: https://snapadvisor.snap4city.org

The SnapAdvisor is a Virtual AI Assistant with chatting capabilities in multilingual. It is an expert of several domains which can be specifically set up from you to become expert of your domain and business. It is grounded on an Advanced RAG approach of DISIT Lab and it is presently based on LLama3.3 model, it can be activated to other LLM models, and in any domain. Each authorized user can actually provide its own set of documents, slides, pages, etc., to create its private domain in which the SnapAdvisor will become immediately skilled.

Examples of domains and communities on SnapAdvisor are:

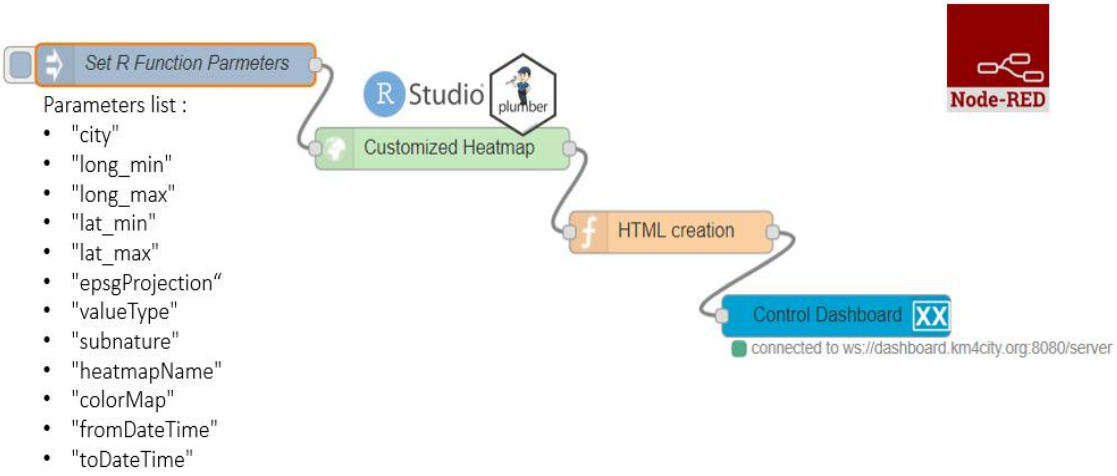
- **Sna4CityAdvisor:** Developers and promoters: all Snap4City and DISIT Lab literature, articles, manuals, slides, codes, etc.: The SnapAdvisor with this skill is accessible only for selected Snap4City users.
- **Legal Advisor** on Disputes for a major Hospital: The SnapAdvisor with this skill is accessible only for legal department of the hospital. It has been set up for the THE project with Careggi Hospital in Florence.
- **GolfAdvisor**: a tool for answering to complex queries about a sport discipline.
- **Operators** of a category of industrial machines. Accessible only for that specific industry and their operators.
- **Questionnaire analysis**, performed for TOURSIMO project on the Questionnaires collected via QR generator and manager of Snap4City. The results on Malta for example, are accessible from: https://www.snap4city.org/1120
- etc.

3.9.9 – An example of Data Analytics integrated with IoT Apps

In the Snap4City solution, the heatmap production can be automatized by implementing a Proc.Logic /n IoT App. The heatmaps of different kinds can be produced based on 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 returns 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 the creation of 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 the creation of a widget inside a dashboard with the HTML content previously created with the function node.

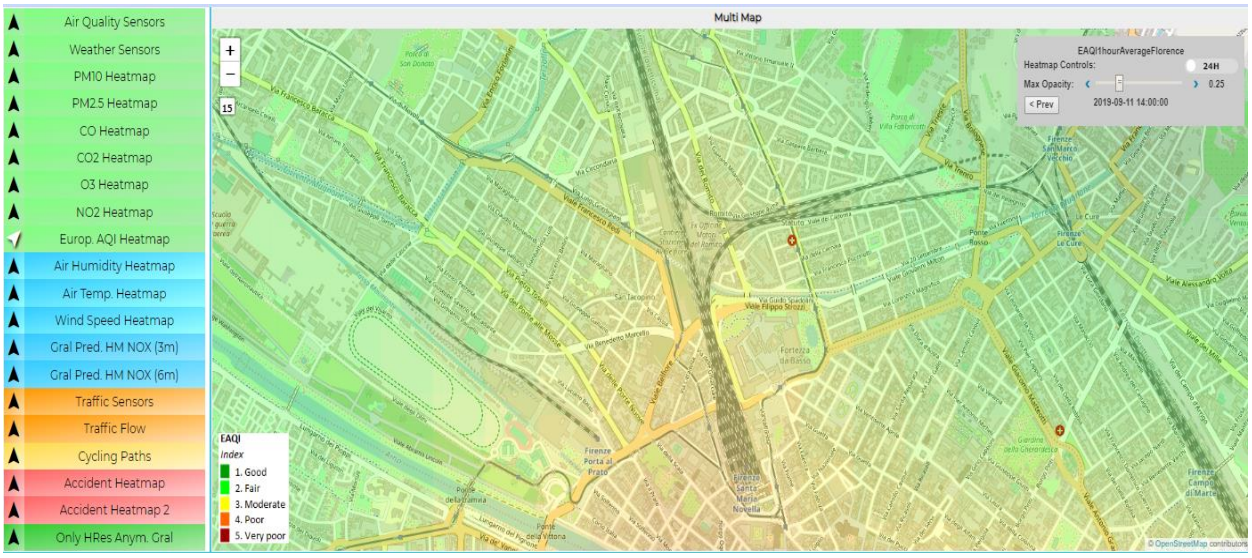


IoT App/Node-RED workflow for automatization of heatmap production.

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

- The GSP coordinates of the area of interest (min/max latitude, min/max longitudes, etc.);
- The presence of some 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 Snap4City 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.

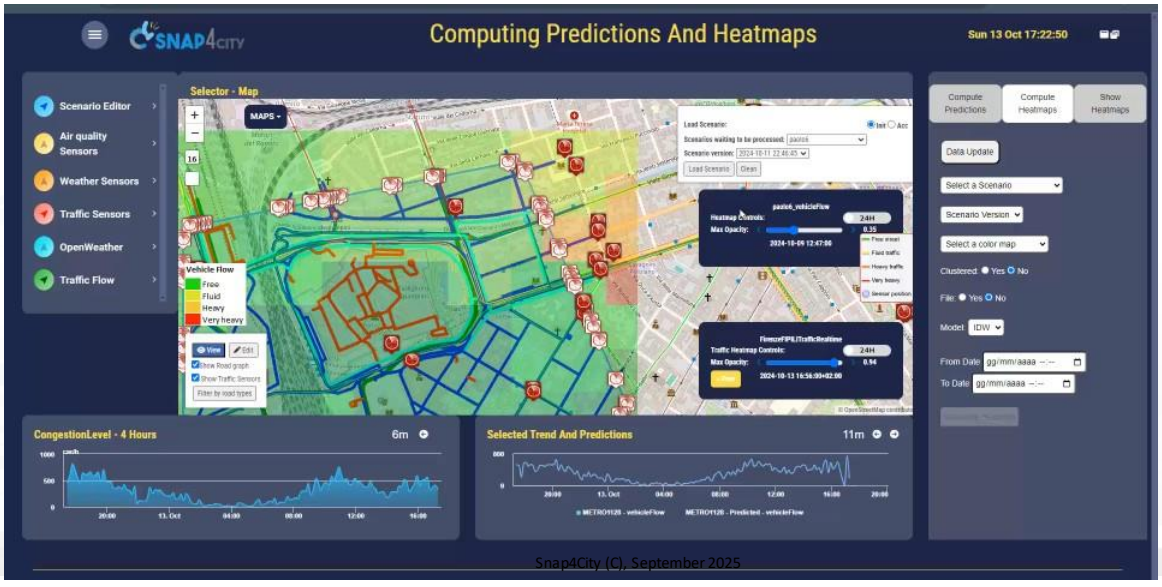
The production of the heatmap data is only the first step of the process since the heatmaps have to be transformed from a mere grid of points to GeoTIFF according to the specific color map adopted for the rendering. This means that the ColorMap has to be available for the GeoTIFF creation. To this end, 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 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 a 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 Dashboards via WMS standard protocol.

Predictions and Heatmaps in Real Time



3.10 – Snap4City Dashboards/views, for Operation and Plan, Business Intelligence, and Control Rooms

A **dashboard is substantially a view of a smart tool to be used in both Operation and Plan, since it may include:**

- **operational services** such as: monitoring data values, trends, events, alarms, conditions, etc., but also providing predictions, early warning, etc.
- **Decision Support System** tool, providing evidence of normal and critical conditions, and in some cases may offer solutions (exploiting AI, Generative AI, etc.), via well designed and connected Data Analytics tools.
- **What-if analysis:** a tool to understand what could happen if something occurs or is going to be changed (in traffic, road infrastructure for example), providing the evidence by prediction and simulations of what would be the effect of changes on the Digital Twin taking to account, for example: traffic, emissions, travel time, congestion, etc. (it can be used on Operation and Plan)
- **Simulation is a class of tools** to simulate the city conditions on the basis of changes: on traffic flow in specific parts, in road graph setting (adding/changing a road parameter as lanes, direction, position, velocity, etc.) (it can be used in Operation and Plan)
- **Optimization is a class of tools** which are capable on the basis of an initial scenario and a set of parameters to provide one or more possible solutions to solve the problem. For example, to reduce congestion, emissions, pollution, travel time, the number of stops at the traffic lights, etc.
- **Scenarios is a class of tools** to select an area of the map and define a context including: road graph perimeter, eventual changes on the road graph, traffic in/out flux, included sensors of any kinds, etc.

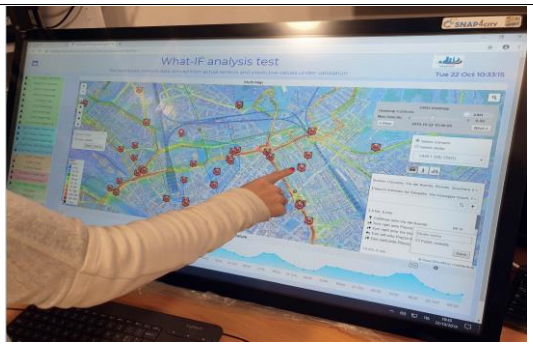
In this regard, it may integrate/exploit Data Analytics, for example, reporting prediction, identifying anomalies, manifesting early warning, providing relationships among entities exploiting inference geospatial reasoning about what is located in the city: resources, structure, people, areas, critical infrastructures, etc.

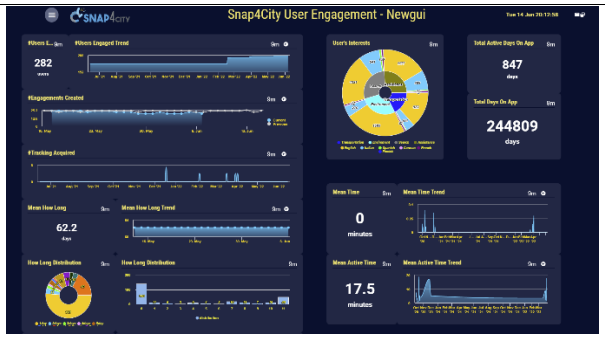
In the development of a Smart City there is a great emphasis to have many different visual rendering tools based on Dashboards, and in particular to satisfy the needs of smart city platform **regular users, developers, city operators, operative dashboards for the verticals, data analysis dashboards, decision-makers, business intelligence dashboards, visual analytics tools, up to the system of dashboards to be shown in the Smart City Control Room and the Situation Rooms, and on Mobile Devices for H24/7 monitoring early warning and collaborative control**. **[Industry4.0-2020], [Dashboards2019], [DashboardProduction2020].**

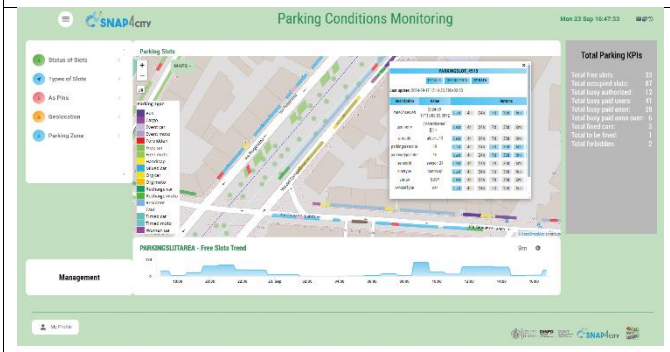
Snap4City *platform allows the creation of* ***Dashboards/Views*** *of any kind which are interactive data & graphics applications including interactive elements, maps, trends, series, tables, gauges, pies, buttons, dimers, semaphores, time ranges, what-if analysis features, and a large set of special 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

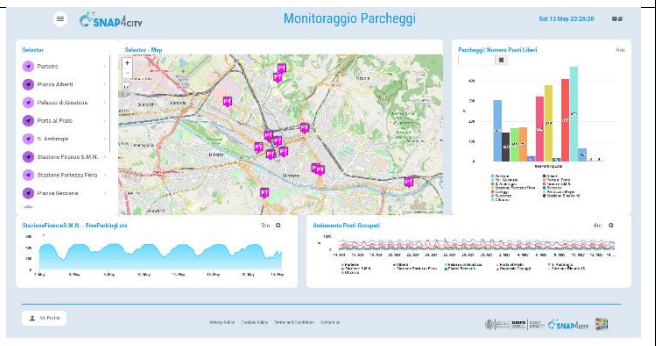

Control Room
SNAP4CITY
https://www.snap4city.org/621

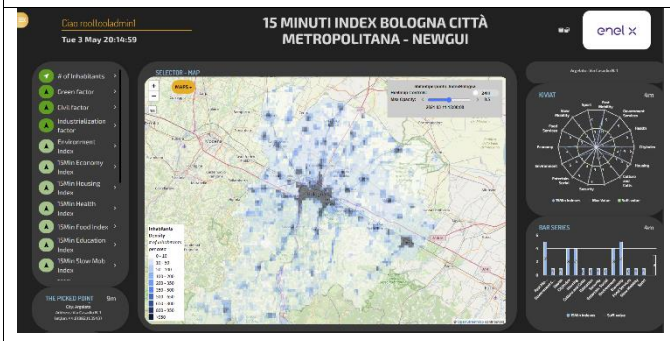

MAPPA FIRENZE

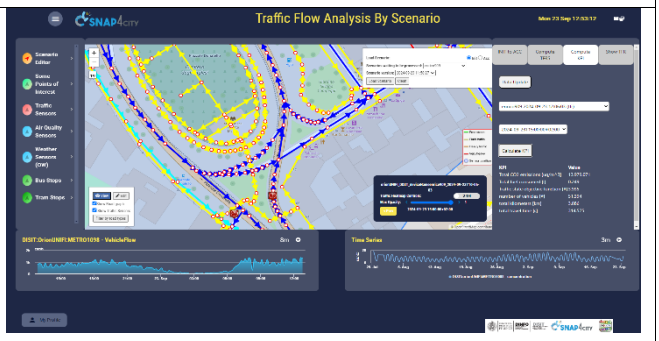

What-IF analysis test

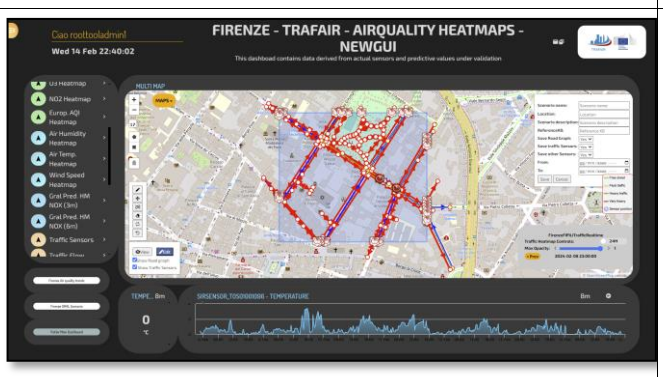

Snap4City User Engagement - Newgui
282
847
244809
62.2
0
17.5

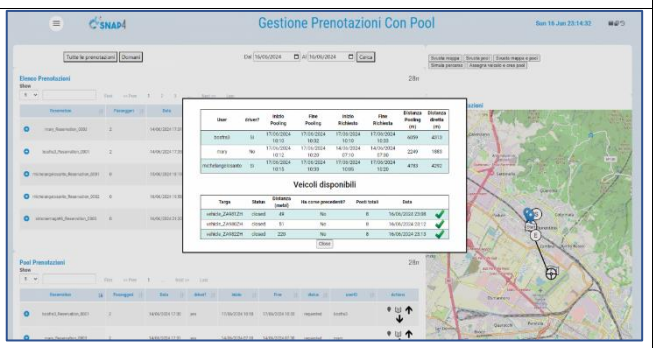

Parking Conditions Monitoring
Total Parking KPIs
Management

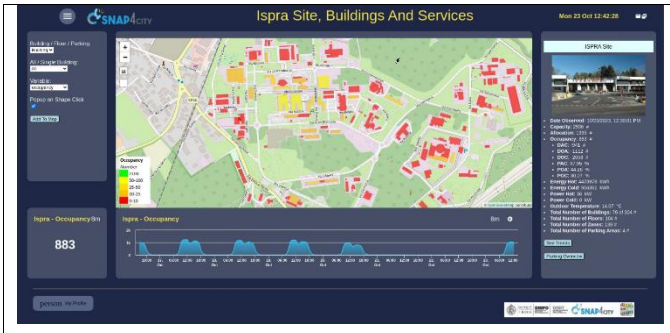

Monitoraggio Parcheggi

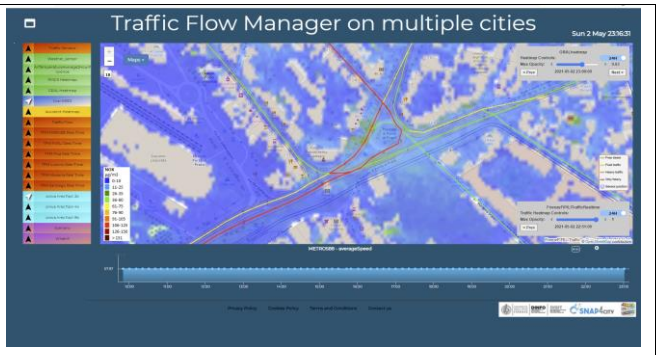

15 MINUTI INDEX BOLOGNA CITTÀ METROPOLITANA - NEWGUI
enel x


Traffic Flow Analysis By Scenario


FIRENZE - TRAFAIR - AIRQUALITY HEATMAPS - NEWGUI

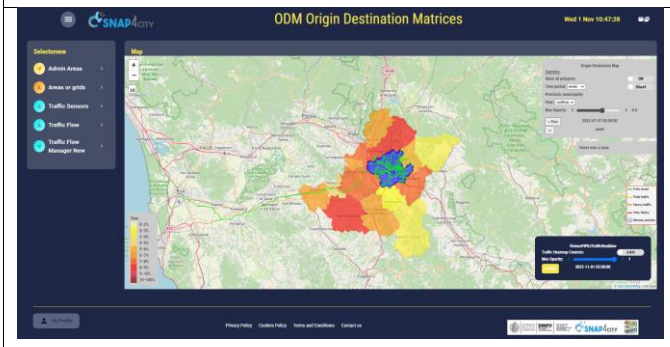

Gestione Prenotazioni Con Pool
Veicoli disponibili

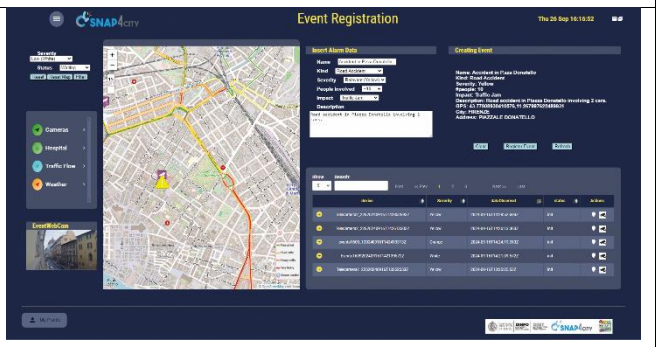

Ispra Site, Buildings And Services
Ispra - Occupancy/0m
883
Ispra - Occupancy

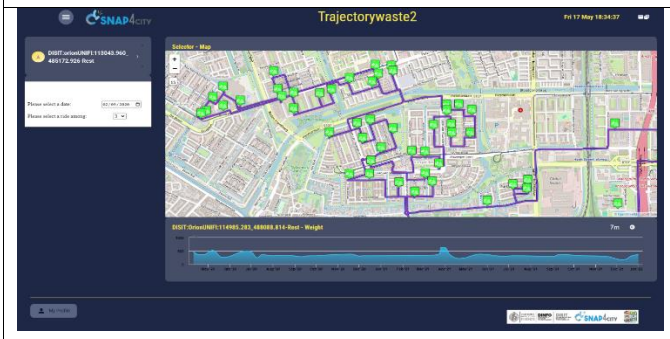

Traffic Flow Manager on multiple cities

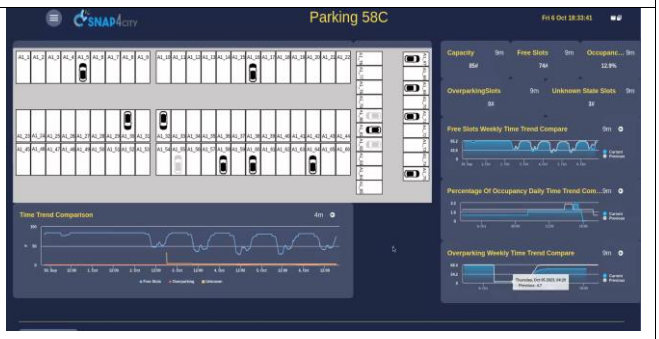

Building 27B Trends


Ispra Floor, Zone And Room Details


ODM Origin Destination Matrices

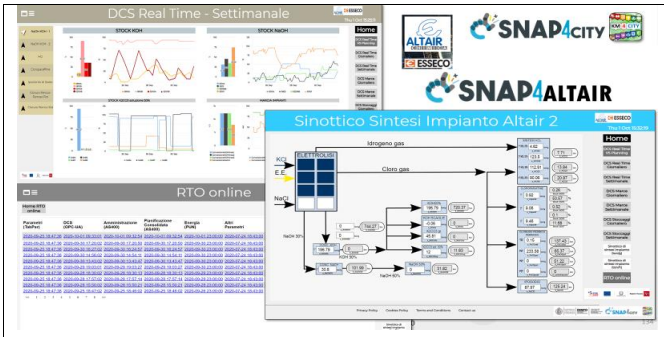

Event Registration


Trajectorywaste2

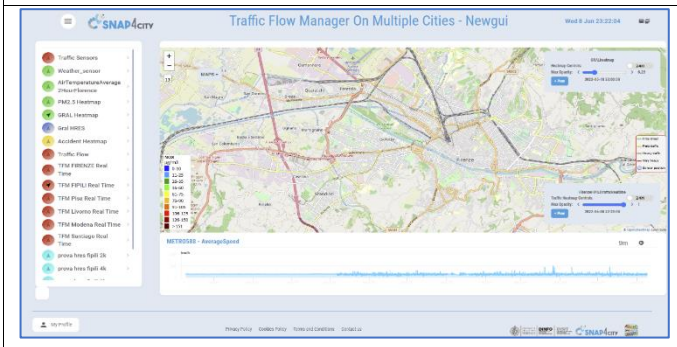

Parking 58C

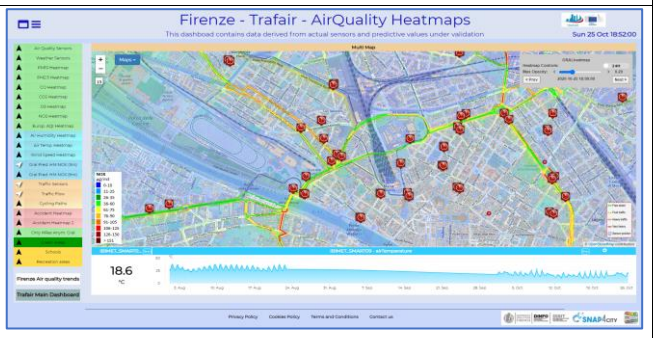

Merano - tutti i servizi
Powered by SNAP4Tech

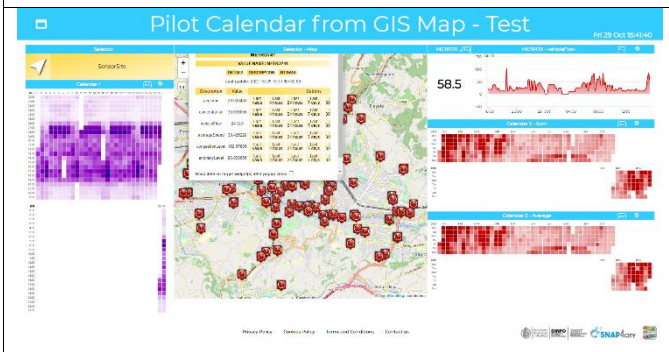

HELSINGBORG H22 DEMO - C3PO-NEWGUI
CAPELON

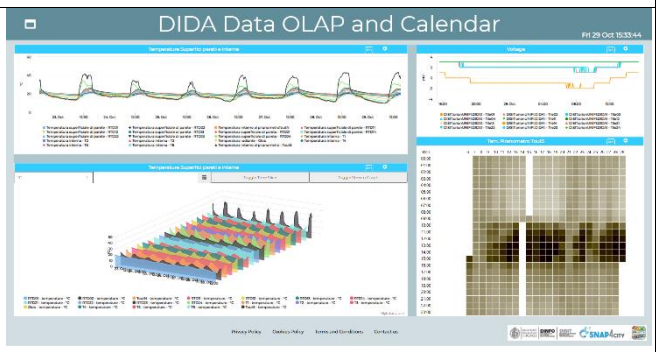

DCS Real Time - Settimanale
SNAP4CITY
SNAP4ALTAIR
Sinottico Sintesi Impianto Altair 2
RTO online

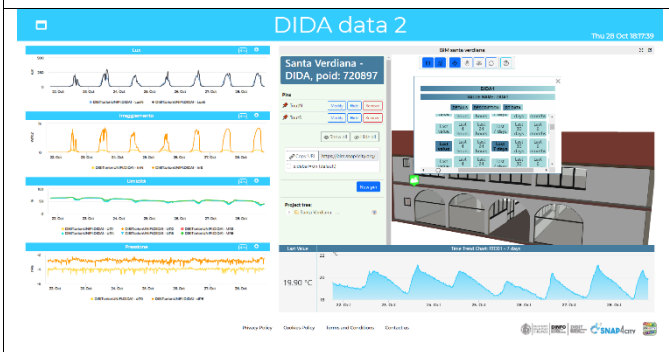

3D MAP GLOBAL DIGITAL TWIN -NEWGUI

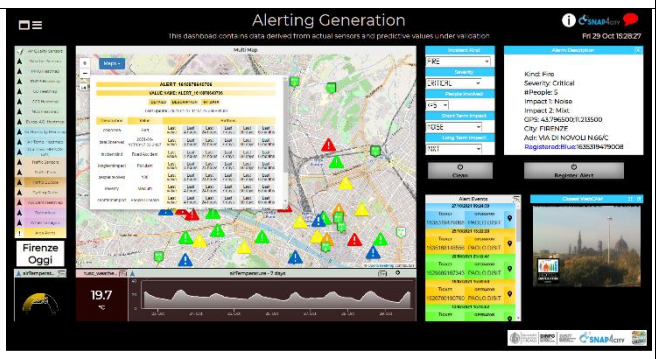

SNAP4CITY
Traffic Flow Manager On Multiple Cities - Newgui

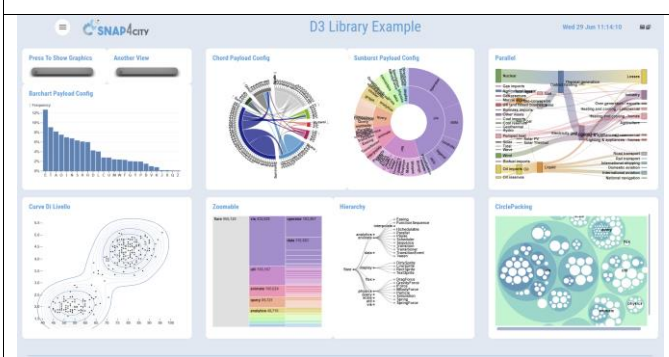

Firenze - Trafair - AirQuality Heatmaps

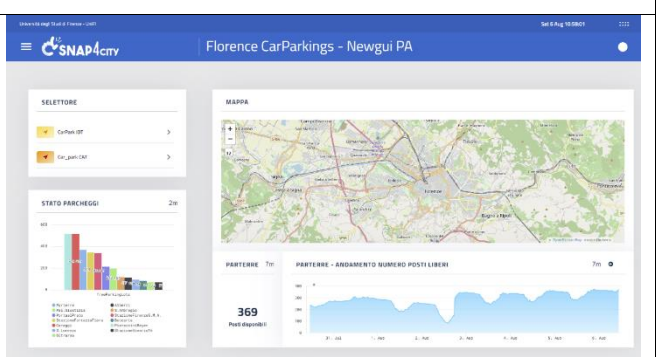

Pilot Calendar from GIS Map - Test

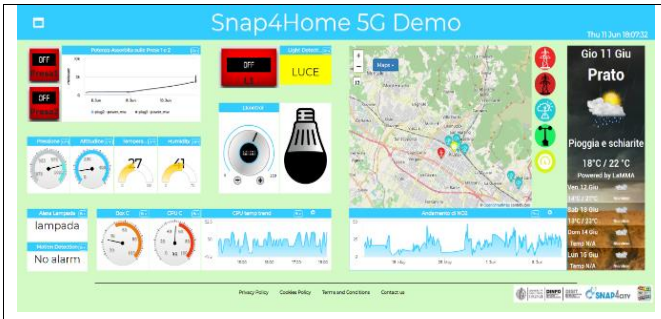

DIDA Data OLAP and Calendar


DIDA data 2
Santa Verdiana - DIDA, poid: 720897


Alerting Generation


SNAP4CITY
D3 Library Example

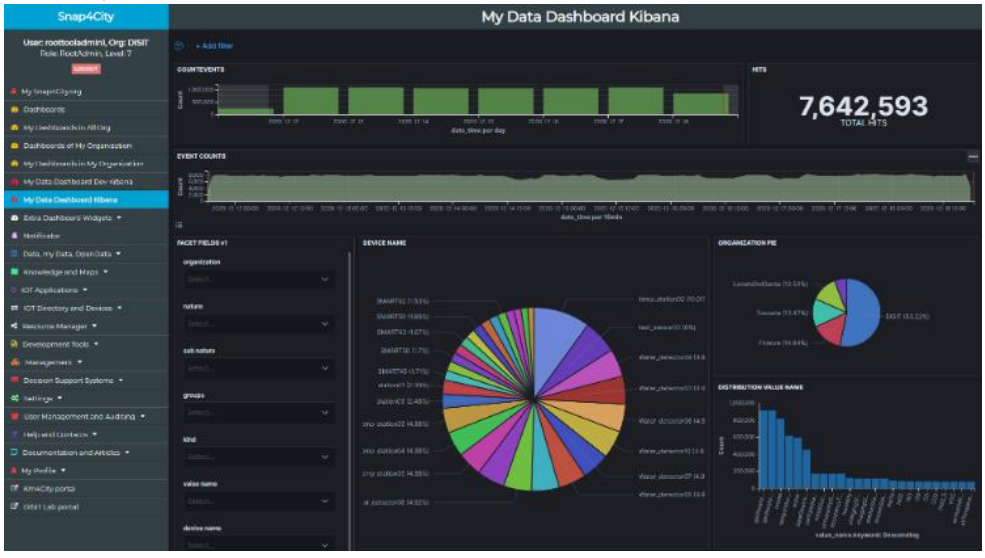

SNAP4CITY
Florence CarParkings - Newgui PA



3.10.0 – Snap4City Dashboard vs Kibana, Grafana Dashboards

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 an extended manner, data-driven widgets, a 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 the control room, decision-makers, situation rooms, operators, etc.
  - Very well integrated with IoT App, Custom widgets, animation, and 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 (which in last versions of Snap4City platform is found as Open Search Dashboard, the new name of Kibana, from AWS fork of Elastic Search)**



  - 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 rooms, decision-makers, etc.
  - Not integrated with IoT Apps, Custom widgets, animations, external services.
  - Oriented to developers, complex production of custom views, etc.
  - Partial support of GDPR and deep control of access.

Snap4City Dashboard presents several differences concerning the Kibana family of dashboards as reported in the following table. For this reason, Snap4City suggests using its dashboards for control rooms and applications.

Features	Snap4City Dashboard Builder	Kibana, Grafana
Large Collection of Widgets, also from D3 library	YES	Nothing
Custom Widgets SVG of any kind, full defined process for customization	YES	Nothing
Real time event driven widgets and data	YES	Nothing
Server/Client Side Business Logic for data transformation with visual programming: Node-RED	YES: visual/coding	coding
Maps with custom PIN, bubbles, animated and moving, etc.	YES	Nothing
Maps with paths, shapes, traffic flow, scenarios, routing, heatmaps, what-if, Origin Destination Matrix, …	YES	Nothing
Maps with Orthomaps from WFS, WMS, GIS connection, etc.	YES	Nothing
TV camera integration and selection	YES	Nothing
Widgets for business logic integration on real time: buttons, selector, switch, etc.	YES	Nothing
Kiviat, Spider net, Calendar (also any other D3 Widgets)	YES	Nothing
Typical Time Trends: day hours, month week, month days, …..	YES	Nothing
Time Trend Compare: day, week, month, year	YES	Nothing
Selectors/Menus: text, icons, etc., also in connection with IOT APP, Node-RED	YES	Nothing
Full control of graphic layout, font, colours, refresh per widget, etc.	YES	Nothing
Iframe integration of third-party widgets and web pages, nesting dashboards, embedding Kibana	YES	Nothing
Connection among multiple Dashboards and Widgets	YES	Nothing
Synchronization with Video Wall, and Operators Views	YES	Nothing
Multiseries, bar lines, charts, pie, donut, simple selectors, trends, etc., also from business logic	YES	Limited
Single content, string, html, any data, etc.	YES	Limited
Special widgets: Weather forecast, civil protection, road plates, Twitter, SVG, etc…	YES	Nothing
Digital Twin Local (BIM) and Global (3D city representation) with 3D traffic, Heatmaps, Devices, …	YES	Nothing
Faceted search	YES: selectors, forms, buttons	YES
Full custom access to third party APIs from CSBL	YES	Limited

3.10.1 – Snap4City Dashboard Builder

The Dashboards are composed by widgets. Each widget may represent several data and has a specific graphic representation, user interaction, and the capability of sending control events to other widgets when some action is performed by the user, for example. Before stating the design of the user interface, you have to be aware of the capabilities of the Snap4City Dashboards and Widgets which are very wide providing almost any kind of widgets and graphic representation for your data, and relationships among them to create not only good representations but also a powerful interaction design, to specific what is going to happen interacting with the graphic elements and data on your user interface.
In Snap4City, there is a specific tutorial for the Dashboard development with several examples and the full list of capabilities in SLIDES: https://www.snap4city.org/download/video/course/p2/

Dashboard Widgets:

- are the main components of the Dashboards/views of smart applications.
- can be created/edited from the Dashboard Builder, resized, placed, changed in colour and theme, etc.
- can be configured to perform a periodic refresh of their data recollecting them from storage/API, third party API, etc.
- can be created/connected to Proc.Logic / IoT App, server-side business logic.
    - can be controlled by the Proc.Logic / IoT App which can command the widget to show specific data from the storage, specific values, etc., and their combinations (Server-Side Business Logic).
- can be event driven, so that they are capable to update their data without forcing any refresh to their data.
- can collected interaction from the user to send them on Client-/Server-Side Business Logic.
    - can be controlled by other widgets.
    - can be controlled by the Client-Side Business Logic in JavaScript coded in other on in the same widget to send/receive command to show specific data from the storage, specific values, etc., and their combinations, and also some computation, etc.
    - can presents dynamically data on the basis of a parameter in the call itself via CSBL
- can open other dashboards thus to create actually powerful smart applications of any kind such as those ready: smart parking, smart light, smart waste, participation, routing, what-if, optimisation, gen

predictions in real time, gen heatmaps, etc.
- etc.

The architecture of the Dashboard Builder is represented in the following Figure. The Dashboard Builder is composed by three main blocks: the **Widget Collection, the Dashboard Wizard, and the Dashboard Editor (which includes the CSBL Editor).**

The **Dashboard Editor** is used to create/modify dashboards (including their logic, visual analytics, what-if tools, etc.), by collecting and configuring Widgets and their relationships, sizing and placing them into dashboard canvas]. Each widget has a number of capabilities in presenting data, collecting data and interacting with users and protocols. The **Widget Collection** includes several ready-to-use widgets and custom widgets (that can be created for implementing new interactive graphic representations and Synoptics by using any SVG graphic editor). Each Widget is realized as an independent module which can: (i) present information to the user, (ii) get actions/interactions from the user, and (iii) interact back and forward with different channels. Channels are implemented as protocols and formats and allow to exploit storage systems (e.g., knowledge bases, relational DB, ODBC, JDBC, NoSQL API), any heterogeneous data sources, connection protocols such as HTTP/ HTTPs, API REST, WebSocket, IoT Brokers API, etc. Therefore, widgets can work/react in an event driven way by Web sockets, and also access the historical data (time series) of sensors, maps, heatmaps, traffic flows, origin-destination matrices (ODMs), as well as query GIS servers (e.g., a GeoServer via WMS, WFS protocols). Such dashboard editing/creation is simplified by the **Dashboard Wizard**, by means of which users can create/connect dashboards in a few steps, exploiting pre-build templates. Moreover, the related wizard guides users in the selection of the most appropriate widgets for displaying the data of interest, or stating from the preferred widget to identify the data which can be used for populating it, or stating from the map to identify the data which are present in the area and the widgets for their rendering, etc. The Wizard assists users by reducing complexity, providing suggestions on finding combinations between data types (time series, vectors fields, array, maps, trajectories, heatmaps, origin destination matrices, point of interest, typical trends, histograms, heatmap calendar, etc.), and graphic representations (trends, multi-trends, pie, donut, maps, chords, hierarchies, solar, dendrograms, single content, Italian flag, traffic flow, 3D building, etc.). Once the editing operation has been completed, users can save the related dashboard (with the possibility to delegate it or grant access to different users) and it is made available in the dashboard collection.

Moreover, with the aim of enabling developers in using the Dashboard Builder to create custom visual analytics, business intelligence, and what-if analysis tools, a flexible approach for modeling any **business logic** is provided with two different manners: **Server-Side Business Logic (SSBL) and Client-Side Business Logic (CSBL**). According to the SSBL approach, some graphic Widgets of dashboards have a counter part in the Node-RED nodes and thus are regarded as MicroServices which the Node-RED can send data and controls to, and which the Node-RED can receive events/actions from, as provided by users. This approach allows the dashboard designer to create SSBL by using the visual programming in Node-RED. This approach also implies that once a new widget node is deployed on a Node-RED flow, the related widget is automatically created into the selected dashboard and a WebSocket secure connection is established. The integration of Dashboards with Node-RED is also used to activate Data Analytics (data processing with machine learning and artificial intelligence algorithms) based on user actions on dashboards and/or scheduling in Node-RED. The CSBL approach is realized by coding segments of JavaScript directly into the graphic interface configuration of widgets (green block in Figure). The CSBL code can call: (i) any external APIs (purple blocks and arrows in Figure), (ii) any API and data base services of the Snap4City platform (blue blocks and arrows in Figure), and (iii) specific functions to send/receive commands and data to other widgets (green block in Figure). This approach allows users who can interact with some widget graphic element (a line, a legend, a bar, a pin on map, etc.) to activate a rendering, a computing, or a visualization on one or more widgets in the dashboard, and even open another dashboard with some parameters. With a minimal JavaScript programming capability to code the logic in these dashboards, a user can add intelligence functionalities to any widget to retrieve data directly from internal and external sources and generate and catch messages from other widgets in an event-driven way.

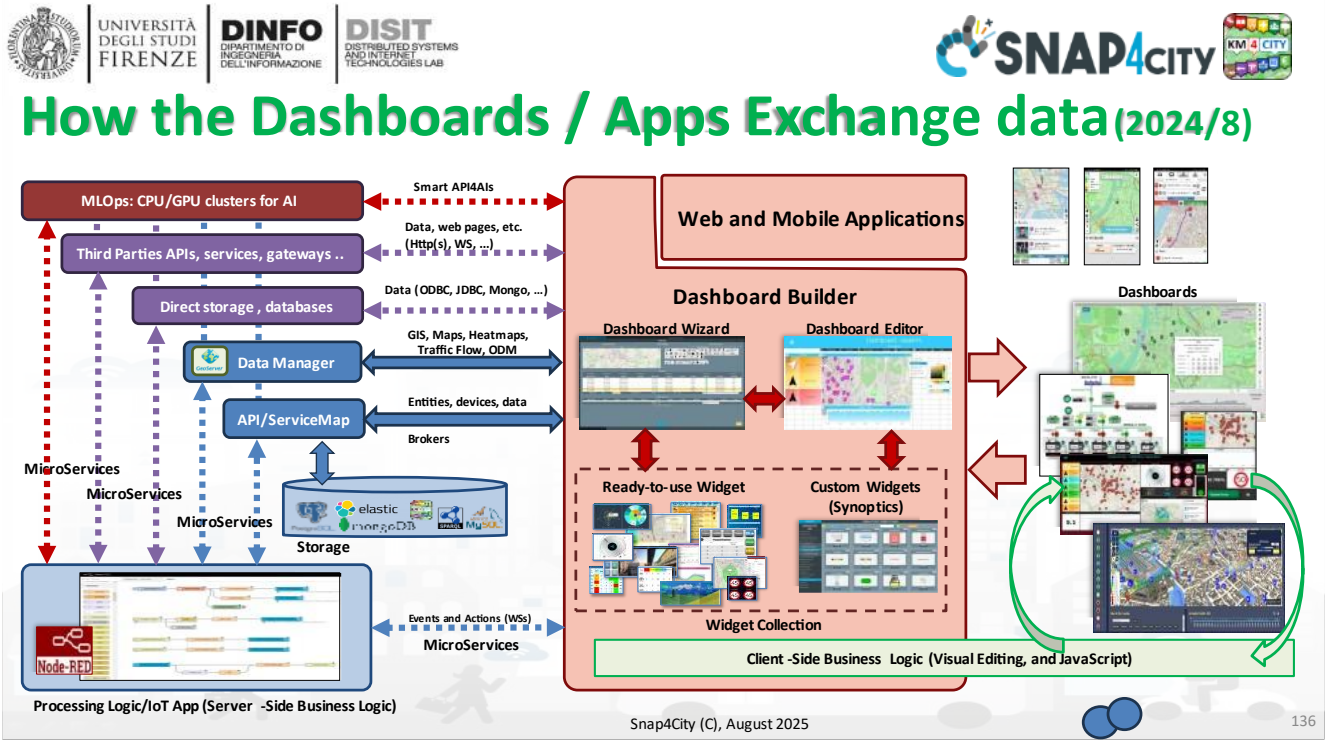


Figure – Dashboards / views and their connections with the platform and other services.

The dashboards can be classified into

- **Passive Dashboards**: **showing data taken from Storage** only, no actions toward Processing Logic (IoT App) node-RED neither on custom JavaScript (ONLY solid BLUE, and some dashed PINK ARROWs in the above figure):
  - Passive dashboards may have widgets of any kind, and a lot of visualization tools without changing the status of Entities on platform, nor sending commands to the Server Side.
  - Passive dashboards are used to creates rendering views of the data in the storage and event driven from their changes with some limited logic pre-coded in the dashboard. For example, a Dashboard with a map and a menu from which the user may decide what is going to be visualized in the page, to browse the data, and see the historical time trend of the time series, etc.
    - https://www.snap4city.org/download/video/course/p2/
  - In passive Dashboard the only external API which can be used are those provided by GIS in WFS, WMS. They are mainly accessed from Selector and MultiDataMap Widget (ping dashed ARROWs).
- **Active Dashboards**, are those that show that from the storage and in addition send/receive commands to/from the logic coded somehow (dashed BLUE and solid GREEN ARROWS in the above figure) and in particular for
  - **Server-Side Business Logic** → logic on Processing Logic (IoT Apps) with Snap4City Dashboard Nodes (dashed BLUE), which is easier to be programmed begin based on Node-RED visual programming.
    - https://www.snap4city.org/download/video/course/p2/
    - https://www.snap4city.org/download/video/course/p3/
  - **Client-Side Business Logic** → logic on JavaScript on specific Dashboard Widgets only for authorized Area Managers developers of Snap4City Platforms. We suggest first prototype by using Server-Side Business Logic, then pass to Client-Side Business Logic in JavaScript. Client-Side Business Logic is coded into the Widgets providing events via the Dashboard Builder in Editing Mode. Client-Side Business Logic may exploit a large number of events provoked by the user on the Dashboard Widgets. **See development manual for CSBL:**
    - https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf
  - **Third party APIs, services, gateways, pages** accessed by CSBL via some API to fill the Dashboards/views, which can be provided by: MLOps, GIS, any GW, etc. (red and pink dashed

ARROWS))
- **Third party databases** accessed by CSBL via some API to fill the Dashboards/views.
- Both CSBL and SSBL may be active together on the same Active Dashboard.

The Active Dashboards are used to implement Business Intelligence, visual intelligence, smart applications and solutions with high interactivity and the possibility of changing the data and the representation of data on the Dashboard/View dynamically on the basis of the user actions. Examples are:

- the click on some button or widget on Dashboard to activate a computation on Client/Server side. The computation on server side can include the activation of complex Data Analytics to be shown as a result on the same or other dashboard as event drive actions.
- To select/collect some data and perform a query showing them on map and barseries, pie, multi-trend, etc.
- To move a slider and see the light on dashboard changing.
- To click on a widget and activate / send control commands to other widgets and maps to drill down, drill up, zoom, on data, maps, etc.
- To filter data from a certain time window and see the changes also on other data time-based representations.
- Select a time period on a time series and see all the other time trends aligned to the same period.
- Select a PIN on map and see a barseries proving last year average data regarding that element.
- To invoke Data Analytics via some API.
- To include HTML/CSS custom widget with CSBL
- Etc. the limits is your fantasy.

Snap4City **Dashboard Builder** enables users of any operative role in the platform to create and manipulate their Dashboards of any kind, passing from simple data visualization tools to business intelligence tools, to control room dashboards and interactive systems and synoptics, including full Digital Twin solutions with sophisticated integrated 3D representation at low cost. See details on dashboard from the training course part 2: https://www.snap4city.org/download/video/course/p2/ [**Industry4.0-2020], [Dashboards2019], [DashboardProduction2020], [ChemicalPlant2021].**

Dashboard Builder allows the creation of 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 UTF-8 also Arabic, dates, left/right time series, etc.; A list of them is reported in the following. You can edit them manually to change any tiny detail of the graphical representation.
- **automatically** using the **Dashboard Wizard** to simplify the production and the connection of data to graphical widgets according to models/templates. Once creates, they can be manually edited for defining and changing all detailed aspects.
- **styling using a large set of styles/theme,** which can be customized and enlarged by following a simple tutorial. The passage from one style/theme to another is immediate, and the default one can be chosen for each of your Dashboard.

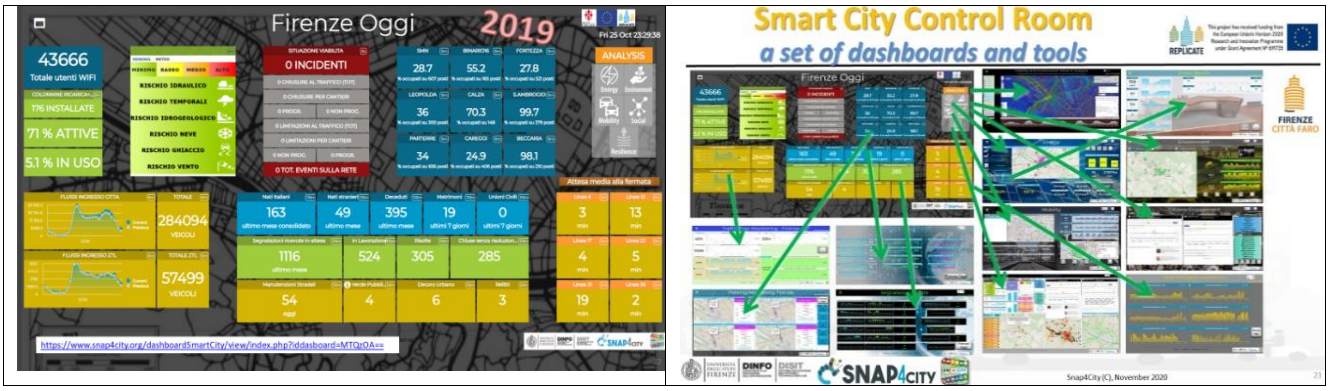
In the Dashboard Builder the visual editor and **Dashboard Wizard** (next figure) can be used on any Dashboard to edit and/or add a new set of Widgets thus accelerating any productions and tuning.

Dashboards may be single or connected. They are typically not a single view, but a view of a set. From the main Dashboard you may need to jump to other views/Dashboards. See the following

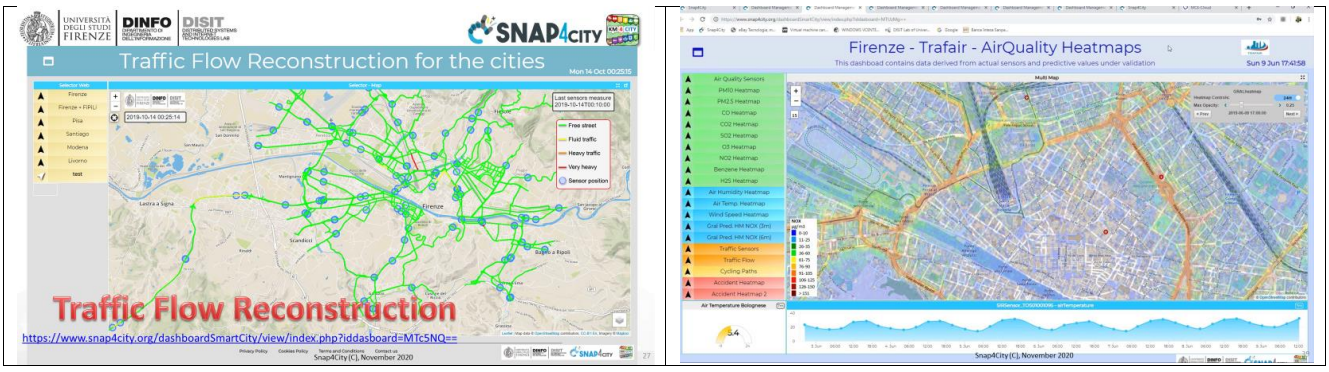
example of the **Smart City Control Room, SCCR**, of **Florence Metropolitan City** (since 2017) which has more than 1.5 million inhabitants, and >14 million Tourists per year, plus students and commuters. The figure shows the main dashboards used by the Florence Mayor and the second-level dashboards. The 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/views may show data coming from Big Data stores, IoT Apps and other sources/databases. They are produced in real-time and may show real-time event-driven data for the decision-makers, officials, city users, totem, operators, fire brigade, emergency, civil protection, police, operators, leaders, 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, drilling down on data, opening live chats for problem-solving, acting in correspondence of alarms by an intelligent monitoring, defining workflows, performing **simulation, optimisation**, and **What-IF analysis**.



Snap4City Dashboard/view system is capable to:

- show dashboards/views on the web browser in an H24/7 modality;
- show data on widgets according to several graphic paradigms (tables, graphs, histograms, maps, Kiviat charts, lists, tv camera, heatmaps, weather, critical city events, 3D, D3.JS, etc.) with a level of interactivity and animation;
- use the Arabic language and Hijri date format and conversions;
- show data on autonomous and connected/ synchronized widgets;
- collect, show, and keep data updated on the screen with an 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 (SSBL) and platform by using 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 a large amount of data providing high performances, such as short response time;
- compute alarms, and provide support by a flexible notification system capable to send alerts, activating tickets for maintenance, automating 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 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 cloning dashboards/views;
- allow export and import dashboards/views;
- dynamically produce dashboards on the basis of some parameter, CSBL;
- allow giving 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;
- script business logic of the Dashboard into the dashboard in JavaScript and/or into IoT App;
- 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, and SSO;
- collect and get data from batch resources and in real-time, using a large range of protocols and formats;
- Associate them a single dashboard specific menus, or a menu for the whole organization;
- Connect to each dashboard a dedicated chat room for discussing problems and events.
- Etc.

The Snap4City Dashboard system has been used for implementing several smart applications, in several cities, actions, projects. A good overview of them can be accessed from:

- Scenarios: https://www.snap4city.org/4
- slides of the training: https://www.snap4city.org/944
- slides focussed on the domains:
    - DOMAIN: Control and Plan Horizontal Artificial Intelligence Platform Digital Twin for All Domains: https://www.snap4city.org/1039
    - DOMAIN: Mobility and Transport Operation and Plan Digital Twin: https://www.snap4city.org/1040
    - DOMAIN: Smart Energy and Smart Buildings Operation and Plan Digital Twins: https://www.snap4city.org/1041
    - DOMAIN: Environment and Waste Management Digital Twin: https://www.snap4city.org/1042
    - DOMAIN: City Users' Services, Tourism Management and Safety Digital Twin: https://www.snap4city.org/1043
- **A large List of *Public Dashboards on Snap4City.org* (many others are in other installations of Snap4City, and most of them are not public)**: https://www.snap4city.org/dashboardSmartCity/management/dashboards.php

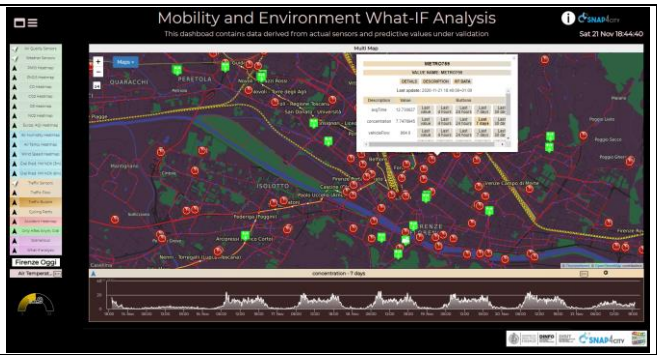
3.10.2 – Examples of Public Snap4City Dashboards

The most complex tools and dashboards **are not publicly accessible** such as:

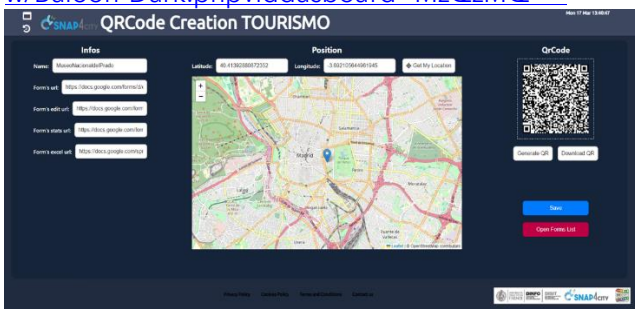
- SnapAdvisor: The Expert at your dispostal H24/7 pn Snap4City and your private content.
- SOLUTION: Snap4City Digital Twin Simulation Support
- SOLUTION: Security, Smart City Asset Management for Cuneo, Italy PDF
- Snap4Rhodes: The "Single Smart City & Cyber Security Monitoring Platform" for the Municipality of Rhodes
- SOLUTION: HOW to: Use of Routing API and What-if
- SnapAdvisor: starting trials from September 2025
- How to control you city and solution: KPI suggestions from Snap4City
- SCENARIO: City Users' Participation and Engagement with Snap4City, PDF
- OPTIFaaS: Operation and Plan, Transport Infrastructure and Facilities Support as a Service
- SOLUTION: 15MinCityIndex: understanding city areas by means of 13 different aspects, PDF
- SOLUTION: Energy Management and Control, PDF
- SOLUTION: Environment Control, Predictions & Prescriptions, PDF
- SOLUTION: Smart Light Control and Light Adaptive with Traffic Density PDF
- SOLUTION: Smart Tourism Management with Snap4City PDF
- SOLUTION: Traffic Infrastructure Optimisation: reducing travel time and emissions PDF
- SOLUTION: Traffic Light Plan Optimisation: reducing travel time, number of stops for vehicles and tramway lines: PDF
- SOLUTION: Snap4Building: monitoring, managing, controlling infrastructures PDF
- SOLUTION: Snap4City integration with Milestone X Protect, VMS, Video Management System PDF
- SOLUTION: Snap4City Digital Twin, PDF
- SOLUTION: eShare in a Snap - The innovative car sharing and car pooling service, PDF
- SOLUTION: Snap4City Smart Parking Manager and mobile App supports PDF
- SOLUTION: Exploit Snap4City in different Smart Waste use cases, waste manager, PDF
- See more on https://www.snap4city.org/4

The following are simpler public examples, also with Arabic and Hijri dates.

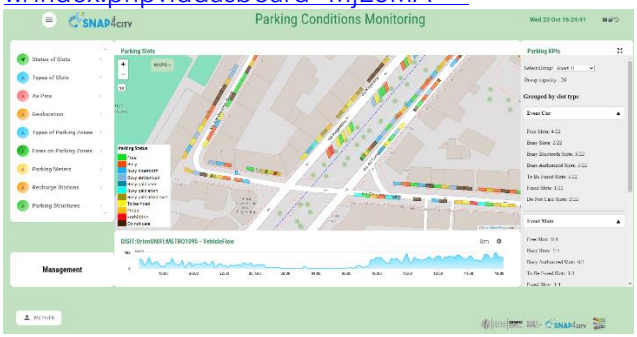
3D Digital Twin of Florence
https://www.snap4city.org/dashboardSmartCity/view/Baloon-Dark.php?iddasboard=MzQzMQ==



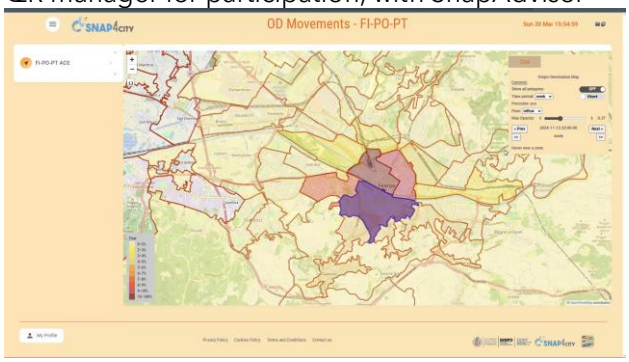
What-if analysis tool in Florence
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjE5MA==



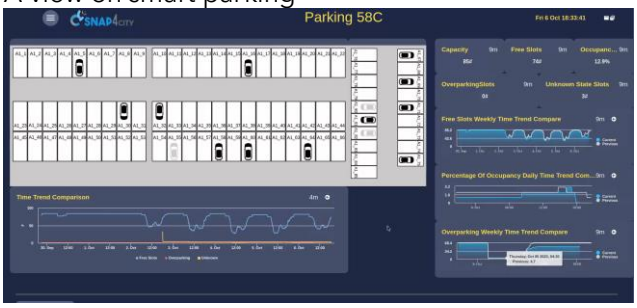
QR manager for participation, with SnapAdvisor



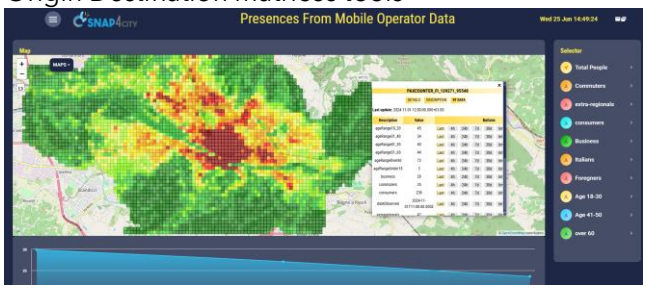
A view on smart parking



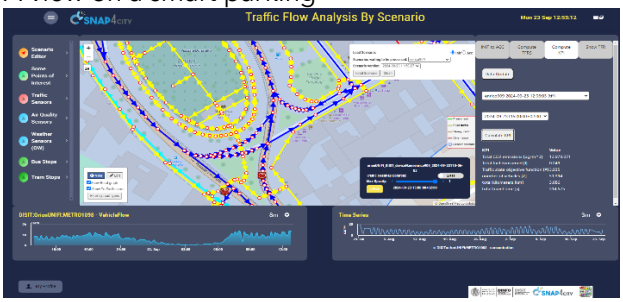
Origin Destination matrices tools



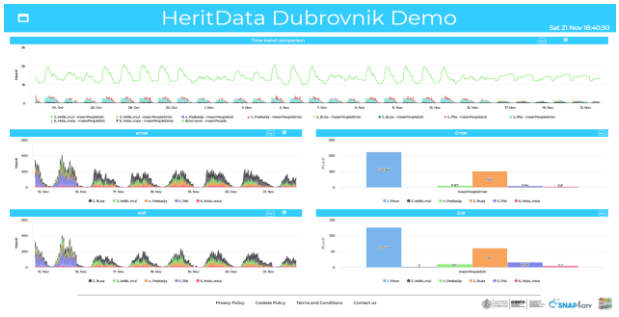
A view on a smart parking



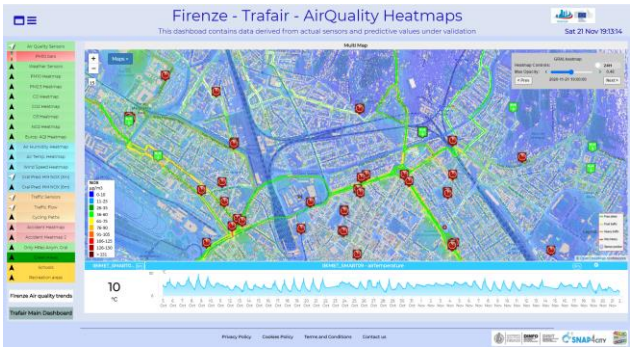
Presence analysis tool



Scenarior based TFR and mobility KPI computation



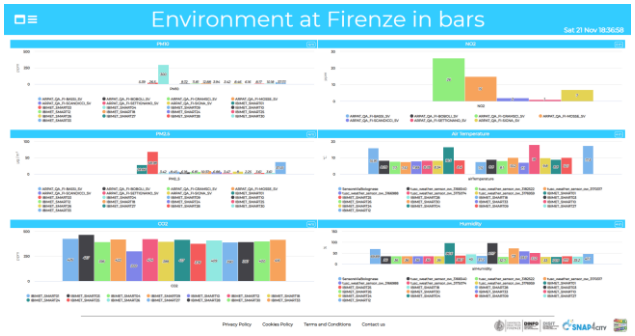
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=Mjg5Mw==
Twitter Vigilance on 6 European Cities



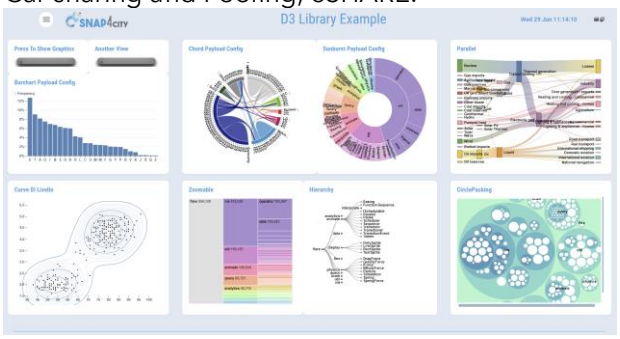
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTUzMg==
Environmental data, NOX predictions and several heat maps

Simulation in SUMO and KPI estimations

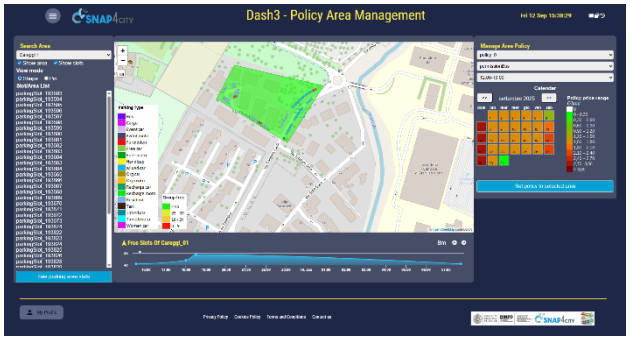
Car sharing and Pooling, eSHARE.



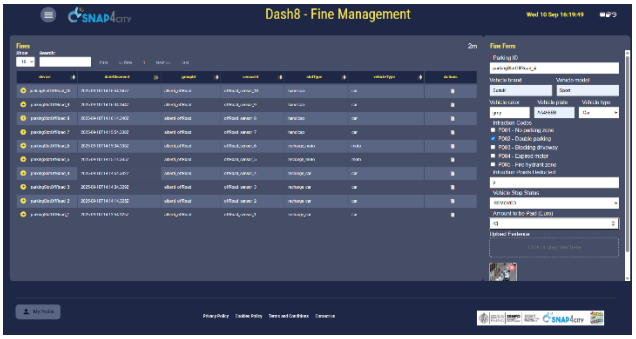
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjQwNw==
Environmental data



https://www.snap4city.org/dashboardSmartCity/view/Gea.php?iddasboard=MzQ4OQ==
Using D3 graphic library on Snap4City Dashboards

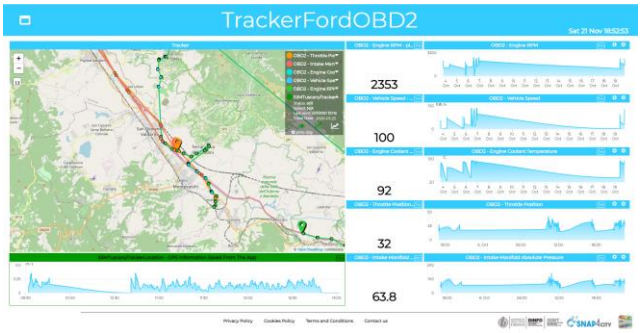


Smart parking price policy areas

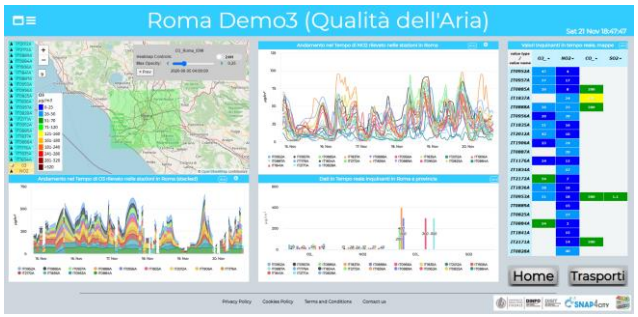


Fines management

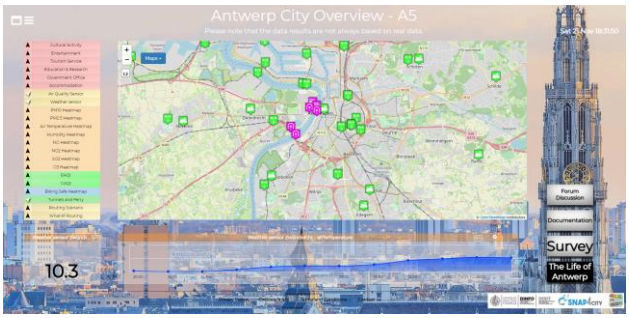
Events and Ordinances in Florence



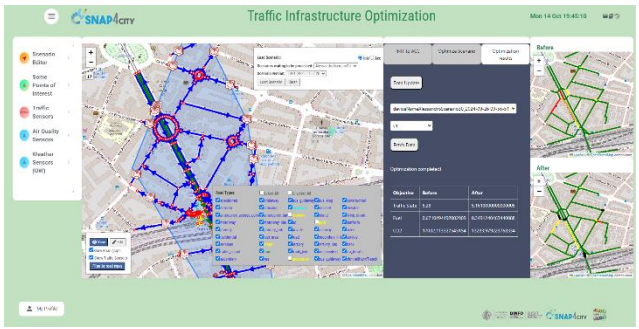
Tracking moving sensors and vehicles



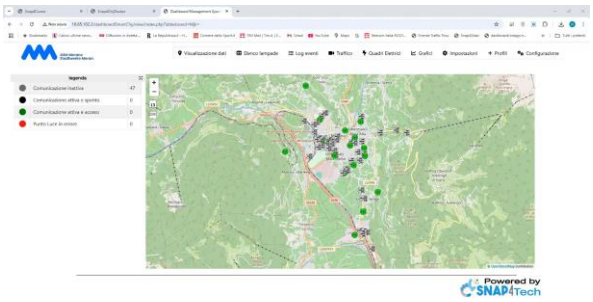
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjcyNg==, Rome



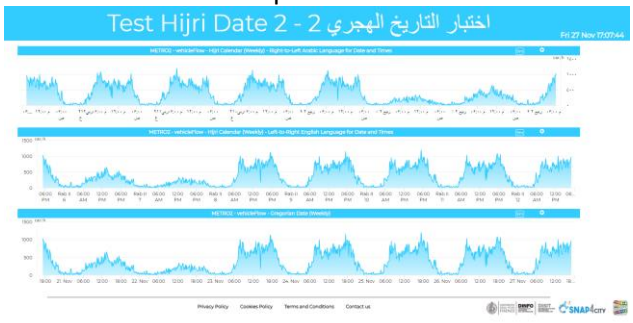
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTQwNw== Antwerp dashboard



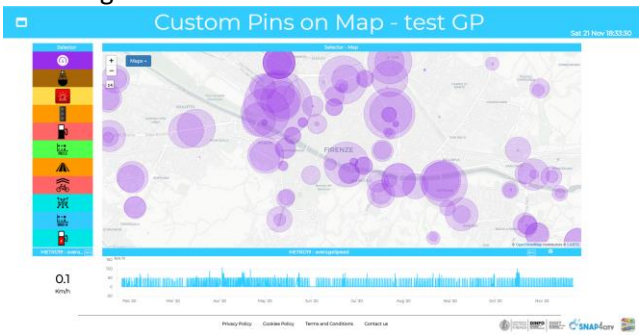
Traffic infrastructure optimisation



Smart Light in Merano

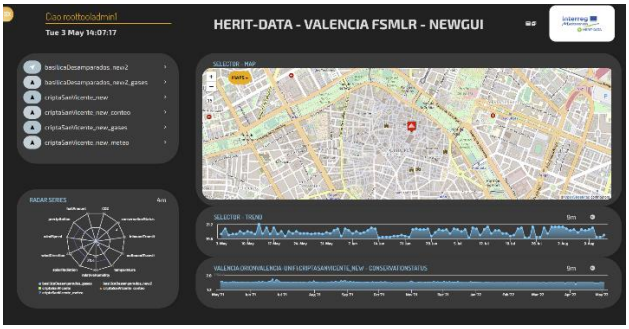


https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MzAyMw==

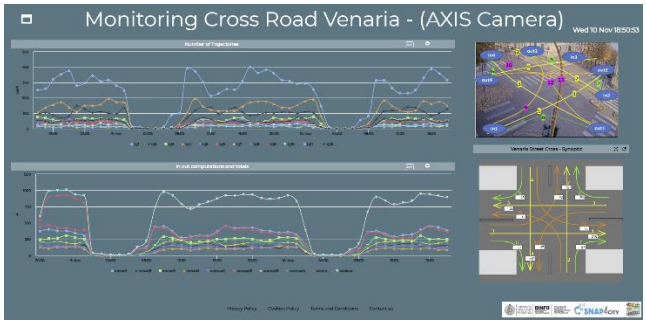


https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=Mjk5MA== Example of dynamic PIN monitoring real services in Florence

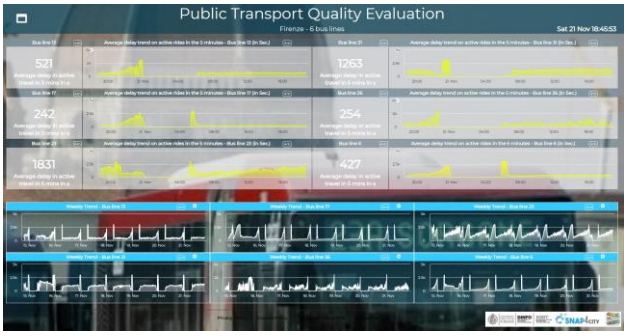
Smart Light in Karlstad, Sweden



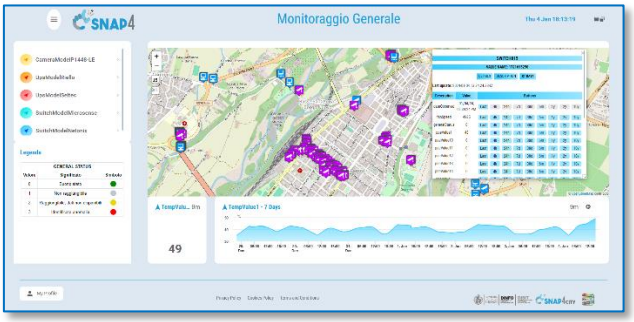
Valencia, Spain



Venaria Reale, Torino, Italia

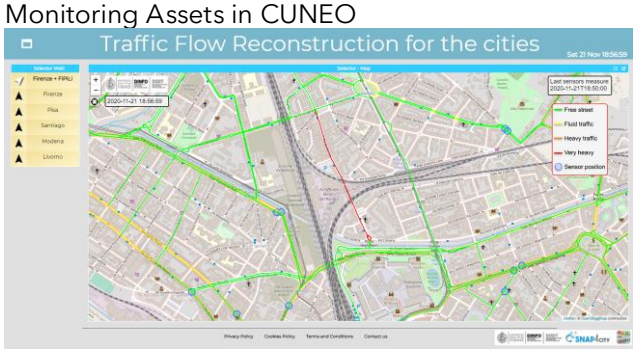


Monitoring of Public Transportation, average waiting time at bus stops.

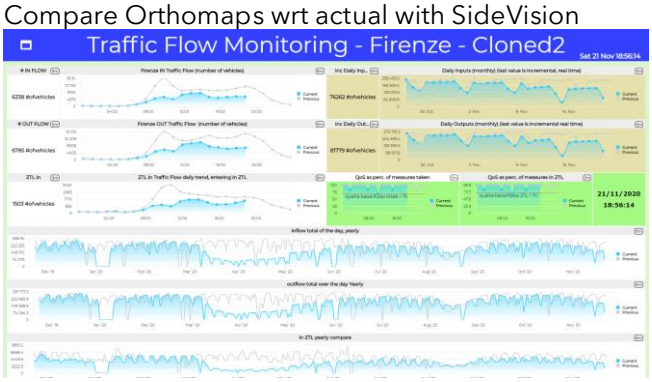


Monitoring Assets in CUNEO

Compare Orthomaps wrt actual with SideVision



https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc5NQ==
Traffic flow reconstruction in several cities

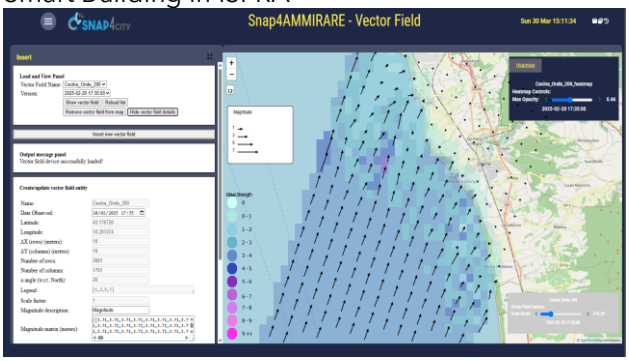


https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjY1MQ==
Traffic flow monitoring, in/out flows in Florence

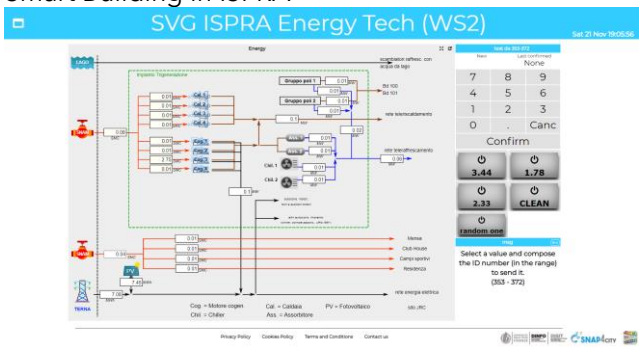
Smart Building in ISPRA

Smart Building in ISPRA

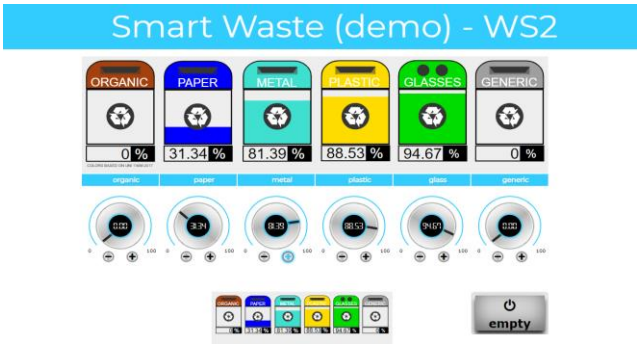
Smart Building in ISPRA



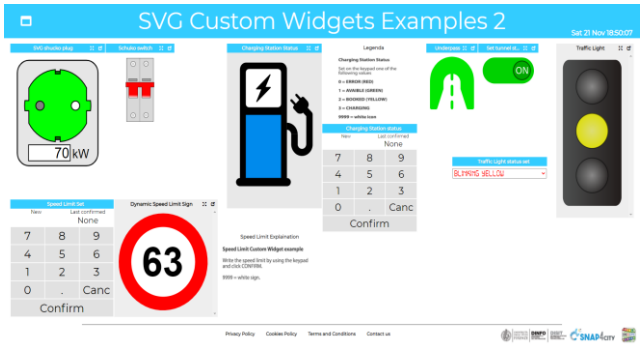
Monitor beach in AMMIRARE



Synoptic, energy consumption

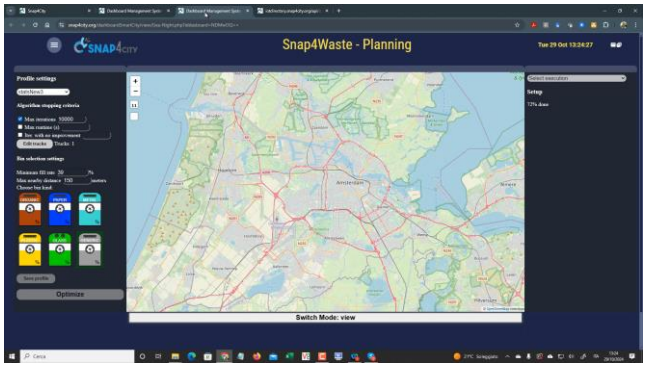


Synoptic for smart waste

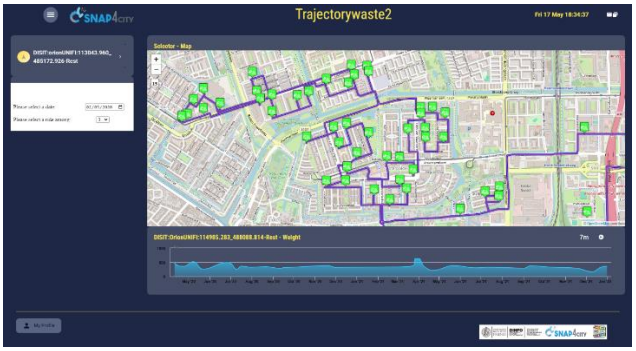


Synoptic SVG custom widgets

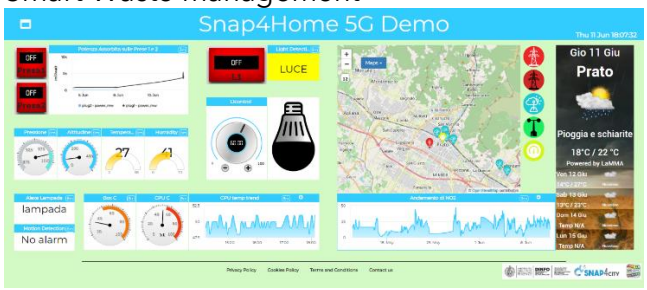
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjY4MQ==



Smart Waste management

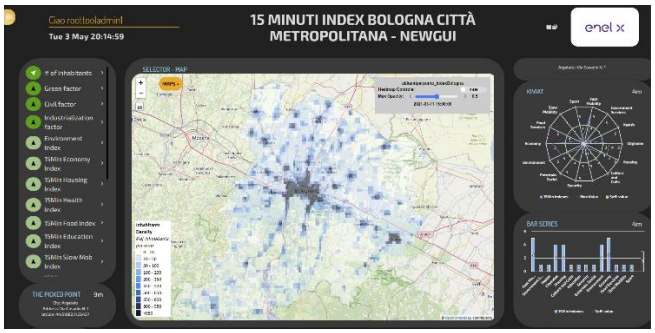


Smart Waste management

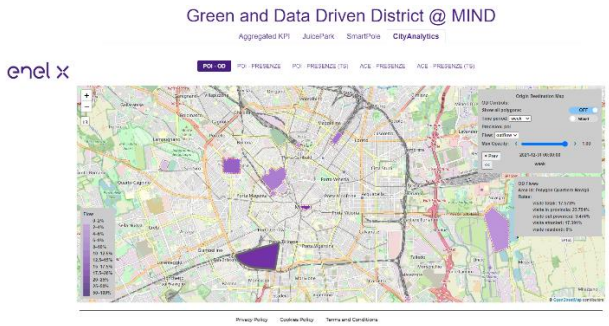


Smart@Home 5G dashboard

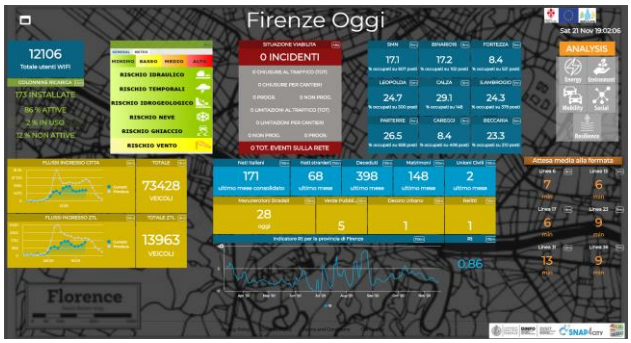
The today's City of Florence



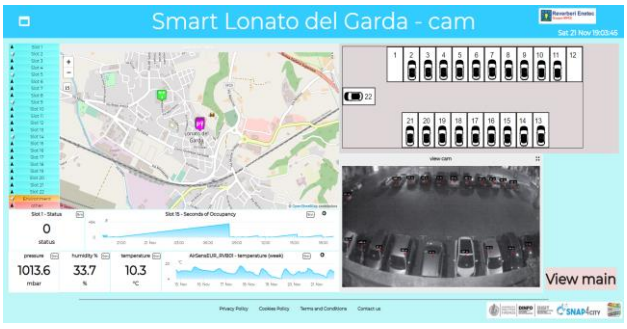
15 Min City Index for Bologna, Italy



OD Matrics, city Analytics in Milano for ENEL-X

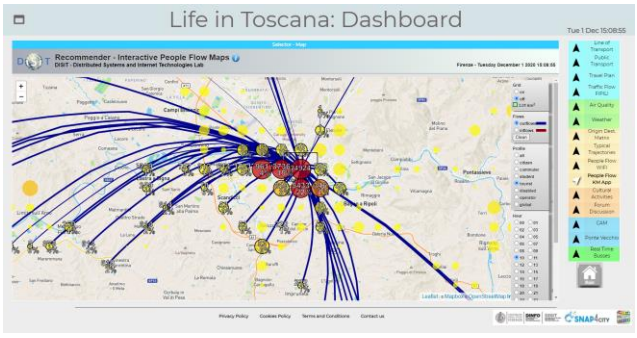


Florence Metro City Control Room main dashboard

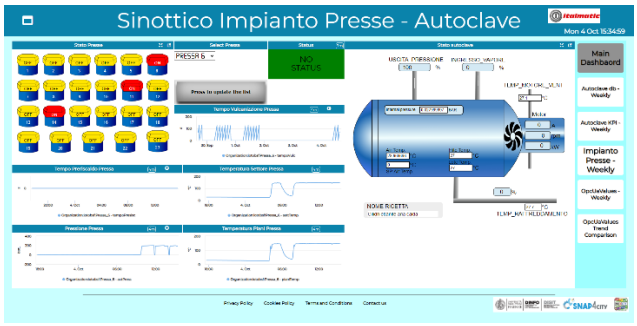


Smart parking in Lonato del Garda

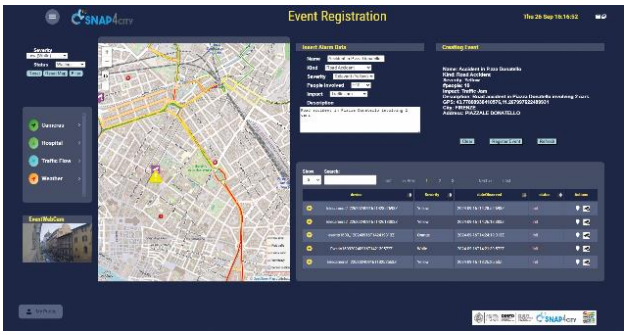
Scenario, production of predictions and heatmaps



https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc3NA==

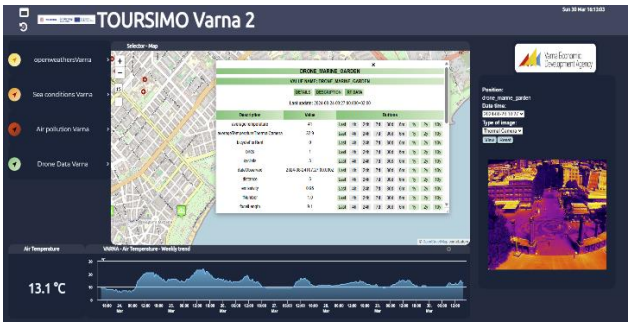


Industrial plant in Italmatic



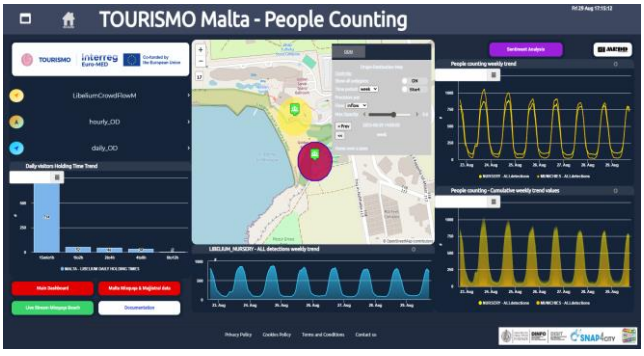
Critical events management

Mobility4Future for Optifaas.



Monitoring people flow with drone, tv cam and pax counters.

People counting and tracking



People flow analysis in Malta, ODM, etc.

Unfortunately, most of the above views are part of complete running application and thus are private since they are actually in operation, 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.

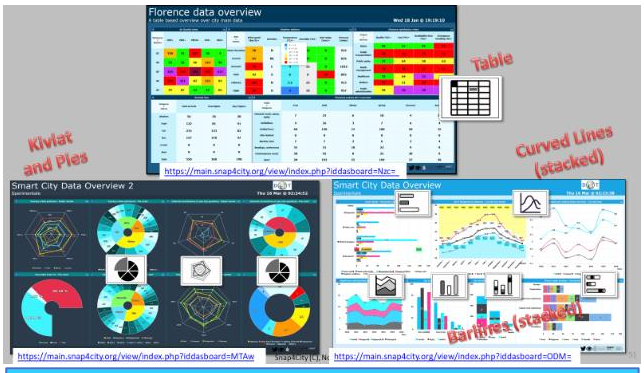
3.10.3 – List of Main Dashboard Widgets for Dashboard Builder

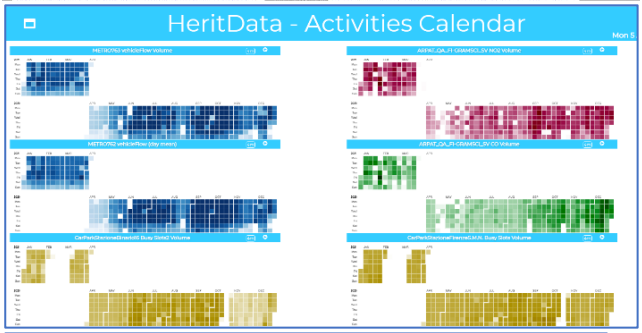
Snap4City Dashboard Builder provides a very large library of widgets that can be used to create smart applications views, business intelligence views, visual analytics views, and also simple Dashboards. A number of them are visible in the above-reported examples. A large number of Widgets in the library and the snap4city Wizard reduce the time required to create views / dashboards, including special widgets for manipulating list sand tables of data. This is a high value because dedicated views are frequently requested when needed, typically *today for yesterday*.

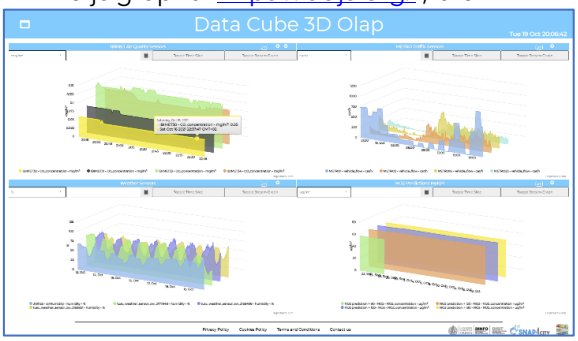
Snap4City Dashboard Widgets can be classified into:

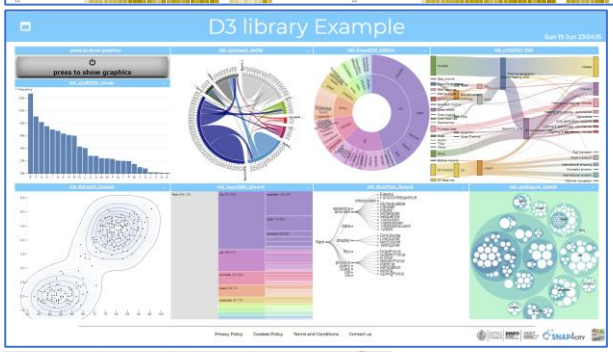
Time Series and real-time data (see also custom widgets for the real-time rendering of data via animation, dynamic PINs on a map, etc.)

- Single content, any value from numbers, strings, to HTML
- Time trend, Typical time trend, double scale
- Speedometer, Gauge, Single Bars,
- Kiviat charts and spider net, Donuts, and Pies,
- Tables, tables with colormaps,
- Multi BarSeries: vertical/horizontal, staked or not, Histograms,
- Multi Time-series: staked/normal, shaded,
- Time Compare for time series,
- Events scrolling lists,
- Heatmap Calendar, bubble charts,
- 3D OLAP time-series, streamgraph,
- Hierarchies, SunBurst, Scatter plots, tree maps,
- Origin-destination maps, chords,
- Ven Diagrams, radials,
- D3.js graphs: https://d3js.org/ , etc.







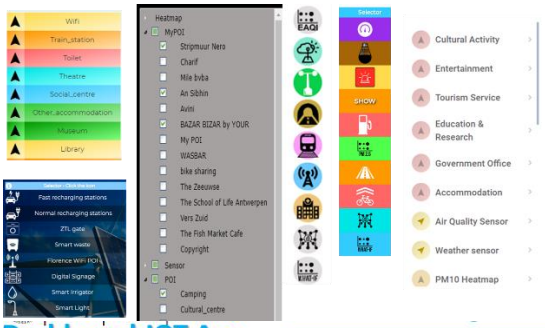


Multi Data Map:

- Maps, pan and zoom, 3D buildings on maps
- 3D global digital twin, Terrain,
- POI, points of Interest, bubble pin,
- SVG Pin, Dynamic animated PINs,
- Orthomaps, backgrounds, comparing
- Heatmaps: calibrated and gaussian, animations
- GIS information, Polylines, shapes
- Routing, routes, traffic flows,
- IoT Devices, Sensors, and real-time data,
- Definition of scenarios,
- Taking measures: maps and floors
- Moving device tracker on maps
- Trajectories, paths, GTFS, cycling paths, etc.

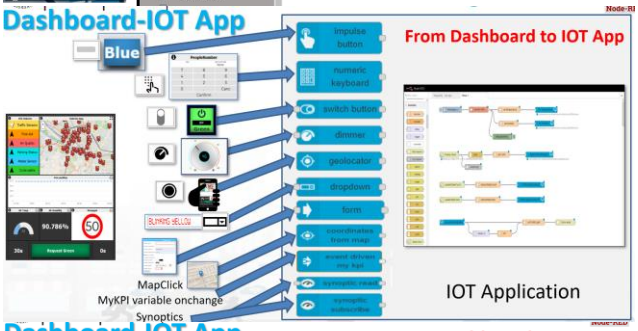
Selectors are Menus for Map and External Content Controllers:

- Selector from the menu to Multi Data Map with icons, text, technical, etc.
- Selector Web: from external Services
- Selector from IoT App, just multiple choice selector to send command from IoT App to Map



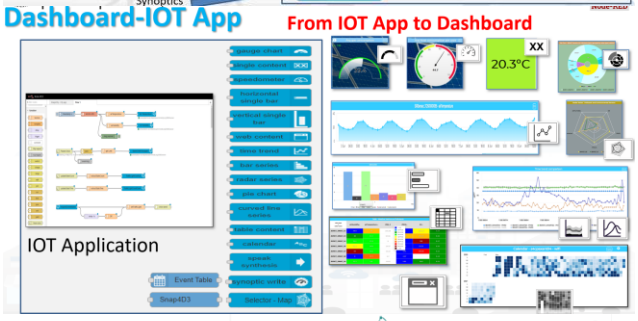
Actuators (see also custom widgets below for other actuators) ***from Dashboard to IoT App***

- Buttons, image buttons
- Knob, dimers
- Switch, impulse button



From IoT App to Dashboard

- Single Content, Time trend, Multiseries,
- Speedometer, Gear,
- Bar series, Spider net, Donut and Pie,
- tables, device table, event table,
- Multi Data Map,
- D3 widgets, ANY!
- Heatmap Calendar,
- Web Pages and URLs, HTML pages.

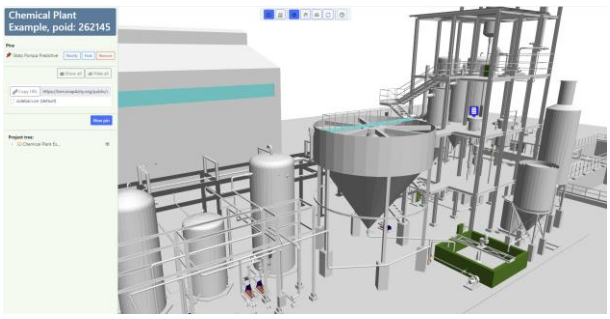


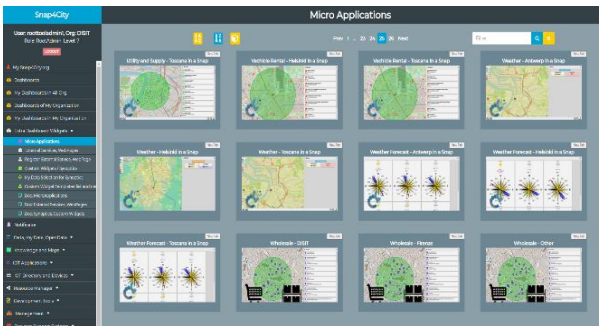
External Services

- https://www.snap4city.org/dashboardSmartCity/management/externalServices.php
- Any web page with Iframe consensus
- TV cameras on WebRTC
- Twitter: hashtags, citations
- Digital Twin Local, BIM with sensors pin on the 3D model

Micro Applications huge collection: https://www.snap4city.org/dashboardSmartCity/management/microApplications.php

- bike sharing, events of entertainment,
- favourite POIs, my personal POIs, marked by the users, near to me, search capabilities, personal trajectories, pollution, pollination, air quality,
- Public transport timelines, public transport, GTFS data, routing, travel planner,
- search of services by selection,
- services browsing: accommodation, parking, wine, food, etc. (about 20 macro categories)
- smart parking, suggestions and goal experiences, tickets for public transport, user engagement, weather forecast, current weather, etc.

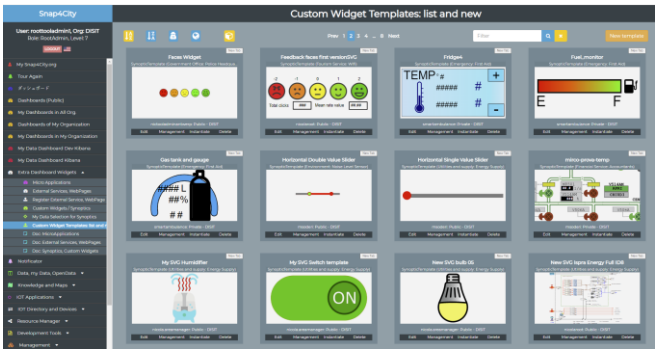




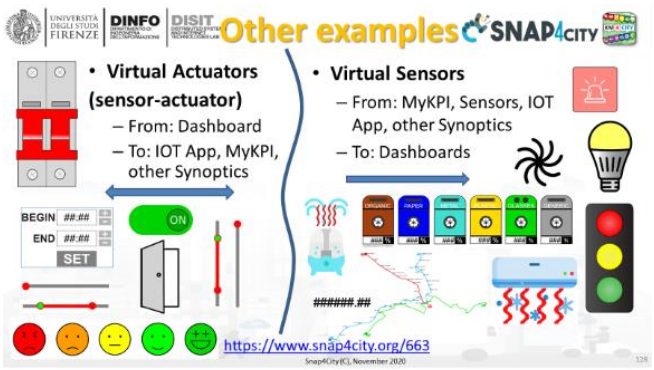
Special widgets:

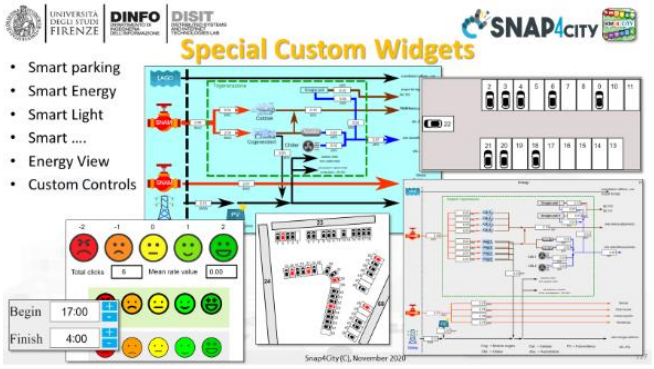
- recently implemented in SVG
- Weather Forecast of several kinds
- Clock (time: HH:MM:SS) real-time
- Civil protection status and alerts
- First Aid Status: single and multiple triage monitoring
- Public transport status
- Decision bar line from Smart Decision Support, the so-called Italian Flag of some SmartDS processes.
- UVA status

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
https://www.snap4city.org/dashboardSmartCity/management/synopticTemplates.php



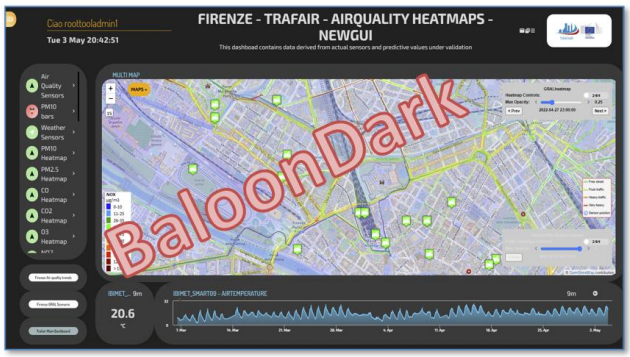
- Air conditioning, de-humidifier
- Alarm status
- UVA status
- Bikes, cars, etc., of different kinds
- Charger status, level
- Digital signage status
- Doors status
- Energy synoptics, industrial plant, custom layouts
- Faces feedback of different kinds
- Fans status
- Lamps, road lamps, …
- Metro status, different layout
- Plug status
- Pollution status: PM10, PM2.5, …
- Red light
- Road signals, Speed limits, etc..
- Single slide, double point slider
- Smart light, of custom layouts
- Smart parking layout, of custom layouts
- Smart recharging station
- Smart waste status
- Switches several formats
- Tank status/level
- Thermometer
- Tunnel status
- Etc.

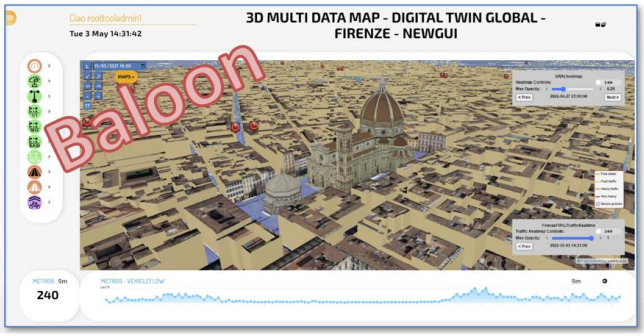


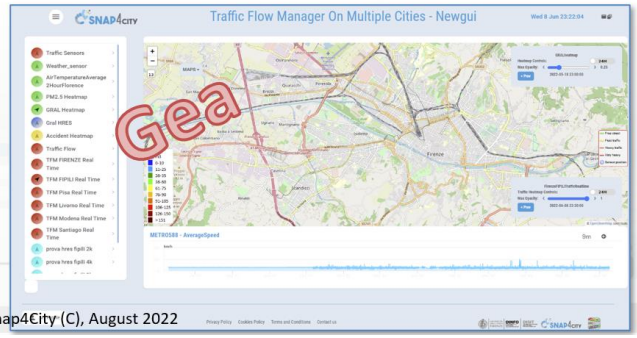


3.10.4 - Dashboard Custom Theme/Style

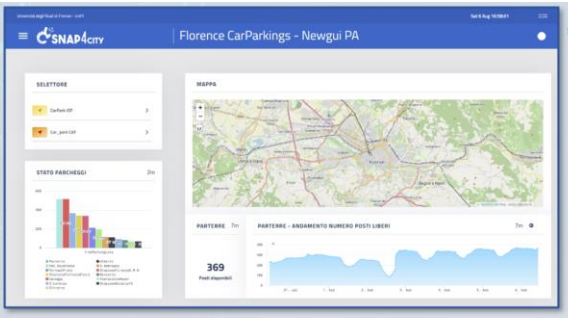
Dashboards may be presented with custom rendering styles/themes. Presently more than 20 different styles have been made and most of them are accessible (also on GitHub) with a manual for their production and customization See section 3.10.4.

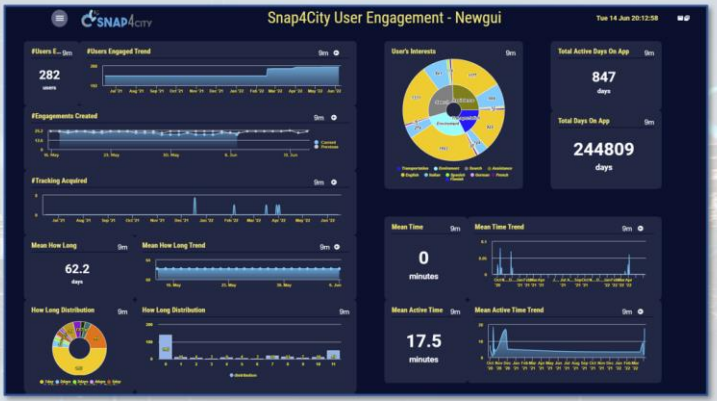






Snap4City (C), August 2022





New styles/themes can be developed by specializing a few files from open source

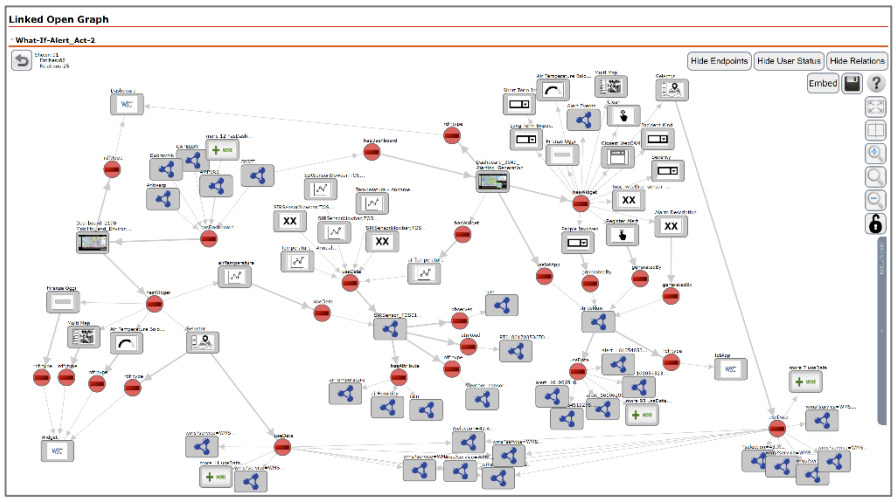
https://www.snap4city.org/793

The Theme/style can be changed and customized to create new themes by following the instructions on page https://www.snap4city.org/793 The Selection of the style/theme can be performed from the Dashboard management panel by each Dashboard owner. In the alternative, you can test the application of any style by changing the URL call of the dashboard in which you can state the name of the theme.

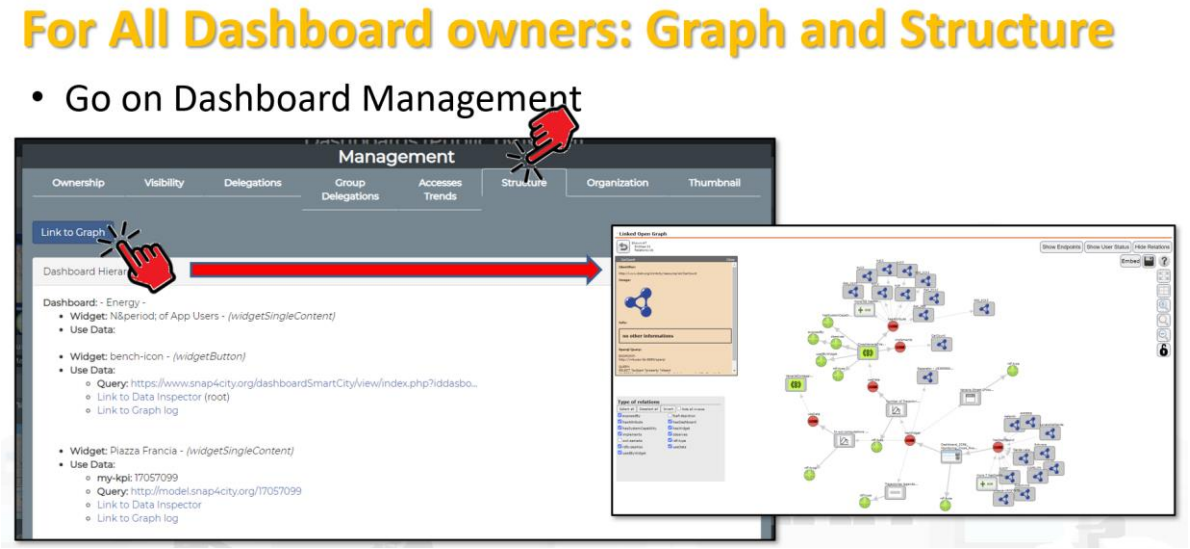
3.10.5 – Structure of Snap4City Applications and cohesions among them

Smart cities are becoming more and more oriented on exploiting technical solutions as tools for decision makers in operation and plan to: (i) enable real-time control of what is happening, (ii) make short, medium and long-term forecasts for planning, and (iii) perform simulations for studying possible new interventions. A modern smart city infrastructure needs to support a multitude of data providers, several data exchange modalities, and many data transformations and services which are executed on the same framework. Data may be very complex, from all points of view (formats, protocols, subject matters and coverage, open and/or private data by respecting specific licensing and access rules, etc.). Thus, several **smart and control applications** need to exploit cross-sector data and services coexisting on the same smart city infrastructure. The different smart services may need to access: (a) data sources in the big data view, (b) computing processes, and (c) representing results/simulations and effect in a proper concise manner. The work of City Operators is not trivial, and may be largely simplified if the smart city infrastructure provides an *integrated framework for navigating and controlling data, processes (microservices), and users' activities in the multitenant infrastructure*.

Smart City ICT Operators need solutions and tools to (I) identify the causes of problems and dysfunctions that may occur in the framework since their inception, (II) provide precise references to data, processes/services, and related APIs when they have to develop new scenarios in the infrastructure, to minimize the costs and effort, (III) enable and monitor and support the developers operating in the same infrastructure reusing data and processes/services.



Snap4City developed a unified knowledge model, UKM, integrating data, processes, and users. Each developer can access the graph of its Snap4City Smart Application via any of their Dashboard accessing a graph as represented in the following figure in which the entities involved can be browsed and inspected. This approach allows the manager to keep under control the impact of changes in the platform on all the other processes, and applications that are using them. For RootAdmin a panel that put in evidence a certain number of indicators is also provider. For example, to easily identify the critical applications, those that are not used, empty dashboards, etc. See more details in training course slides: https://www.snap4city.org/download/video/course/p6/



3.10.6 – Snap4City Smart Application, Business Intelligence Tools

See for details: https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf

This is just a summary.

The design and development of a smart city is much more complex than regular software developments approaches, and thus we need a specific development life cycle. Peculiar aspects are the need of: (i) fast prototyping, (ii) collaborative activities exploiting and reusing the same component by multiple developers, (iii) integrating data flow and control flow in event driven, (iv) integrating in the development life cycle also data analytics, (v) developing front-end tools which may request further data analytic and sophisticated business logic, etc. The Development Life Cycle Smart Solutions involves a number of specialized tools such as those for creating flows and dashboards, etc.,

The architectures for IoT (Internet of Things), IoE (internet of Every Things), WoT (web of things) and big data development tools based on microservices can be generalized and can be classified/assessed according to the following model in which the most relevant areas and connection among them are identified and described. In **Figure 1**, a *conceptual architecture* is presented in which the most relevant data flows among the main components are highlighted. Data/Control Flows are also drawn by corresponding control flow in the opposite direction, or identical in the case of event driven flows. Each module/block of the figure represents a functional area and not a single tool. For example, the Data Ingestion includes the set of tools, processes and services for data ingestion/production, and the same complexity is present in the Business Logic area, Data Analytics and Front-End areas, and thus also the Storage area may present multiple storages and processing. Identify management and access control are diffused in all the communications among all areas and tools.

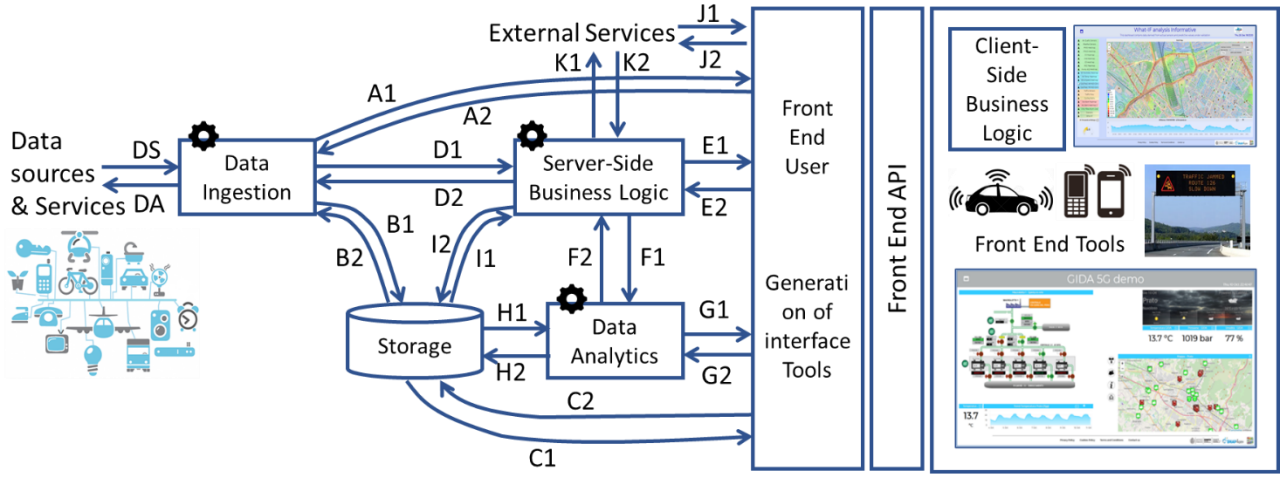


Figure – Conceptual Architecture, from simple to Advanced Smart City Solutions.

The early generation of smart city/IoT applications have been implemented by tools for open data management as well as by GIS (Geographic Information Systems) solutions. In those cases, the IoT data services managed data as Data Sensors / inflows (DS) and Data Actuators / outflows (DA), to store them in some Storage via **B1** (may be indexed and info retrieved via **B2** or other channel) and rendered them asynchronously on the user interface via **C1**. Typically, the Data Ingestion / transformation is performed with the so called ETL/ELT (extract transform/load load/transform) processing [MicroServices2019, MicroServices2024]. The arrival of IoT/IoE devices constrained to be more reactive (data/event driven, managing real-time data) in connecting DS/DA and data services, thus producing the needs of having **A1** connections, possibly data/event driven. In that context, Data Ingestion tools have to be implemented by Brokers (also called IoT Brokers) to manage IoT Devices /Entities via bidirectional protocols (**C1/2** and **B1/2**). At the same time, for acting on the field, **A2/A1** connection was activated to allow sending data and events from the User Interface to Brokers and Devices/Entities as signals/messages in event-driven mode.

Data Ingestion area is the area in which a large number of data ingestion processes can be implemented and put in place to cope with many kinds of data/entities gathered and produced: Entities Instances/IoT Devices

(sensors and actuators, may be based on Smart Data Models, IoT Device Models, etc.), POI (Point of Interest), typical time trends, KPI, SUMI, heatmaps, constrained scenarios, traffic flows, personal data, user profiles, maps, orthomaps, shapes, GIS data, trajectories, origin destination matrices, scenarios, vector fields, routing, payment models and costs, TV cams, synoptics, 2D/3D objects, BIM buildings, floors, etc. They are produced exploiting a large number of protocols Https://www.snap4city.org/65.

IoT Edge / Edge solutions are located on the field and before the Data/Entity Ingestion area. They may include a number of the functionalities reported in **Figure 1 (e.g., ingestion, local storage, business logic)**, while the substance does not change the general data and control flows. Typically, the Edge only includes the Data collection/Ingestion, which as described above, may include some data/entity transformation and exchange. In the case of Fog computing, some Data Analytics (generically intended as any process consuming data and producing some results, which can be via statistic, ML, AI, simulation, optimization, etc.) mainly far from the Front End and may be not storing the results on some Storage but passing them to some other entity in the network to reach the actual platform on cloud.

Slightly more complex solutions could be those that need to perform some computing on stored data which can be performed only asynchronously due to their complexity (e.g., via some periodic processes), or even on the streams. The first problem can be solved by activating Data Analytic/AI processes and services, exploiting connections **H1/2** to read historical data entities and write results back (for example for computing some machine learning, simulation, predictions, early warning, anomaly detection, heatmaps, origin destination matrices, KPI, etc., saving results or derived trained models). The results can be made accessible on the user interface via **C1**, once saved in the Storage Area, or via **I1/E2** if some Business Logic arrangement is needed. In some cases, the results can be passed back to the Business Logic without saving via **F1/F2**. Otherwise, via **G1** the analytics results can be provided on the user interface on the fly without saving values (assuming that the results are just a temporary exploration), that is the classic path of Business Intelligence tools.

Typical smart applications and services need to compute predictions/simulations, optimisation, suggestions, prescriptions, analysis on the basis of historical and contextual data. Examples of smart applications and services exploit computing tools are: *smart waste for optimising waste collection, smart irrigation (deciding the best irrigation time), smart parking publishing free parking slot predictions, optimizing routing for waste collection, optimising traffic light timing, multimodal routing for final users, and for computing origin destination matrices, heatmaps, anomaly detections as early warning, optimisation of the transportation infrastructure, etc.* The computations could be performed periodically or on demand. The results are usually saved on Storage via communication and services, such **H1/2** connections. The Data Analytics processes may also need to directly interact (via **G1/2**) with the front-end layer in which the user interface content is produced (production of HTML pages, streams, etc.) for the devices such as web and mobile App, Dashboard, View, digital signage, etc. This also means that specific APIs **G1/2** must be exposed from Data Analytic/AI processes and make accessible them for the front end (in authenticated and protected manner). Please note that from the front-end, it is also possible to call any **external services (J1/J2) provided by third party and even not included into the Snap4City framework such as** storage or other, via some amy APIs, WSs, etc.

Recently, in the context of Smart City, IoT/IoE, industry 4.0, WoT, a number of complex smart applications are demanded by Decision Makers for city management and Control Rooms (with the aim of implementing workflow management, optimisation, plan, simulation and what-if analysis), developing complex workflows in short time, with frequent updates and modifications of data analytics parameters, data flow, workflow, and business logic / Intelligence on the user interface. To this aim, most of the smart solutions integrated the possibility of implementing a Business Logic module with workflows/dataflows (exploiting **E1/2, F1/2** and **I1/2** connections**),** thus including multiple connections for real time data driven and event driven flows (**D1/2, E1/2, also with external services via K1/2**), as in **Figure** 1. To this end, Snap4City is much more complete with respect to most of the development environments (such as AWS, MS Azure IoT, etc.), in addition, Snap4City uses visual languages and MicroServices as Node-Red by JS foundation [MicroServices2019, MicroServices2024]. Thus, the capability

of the solution highly depends, in practice, on the flexibility of the Business Logic/Workflow, which in turn exploits API and MicroServices, and thus on the integration of the user interface to create smart applications with the development environment and data flow. These smart applications can be regarded as **custom Business Intelligence** tools. In the case of Business Intelligence tool, as well as in Visual Analytics the Business Logic defining the actions in response to the user interaction can be on client side or on server side. Moreover, Snap4City **Client-Side Business Logic, CSBL,** allows each user to see its own experience on Front-End without loading the server side of many contextual information for each user. Snap4City **Server-Side Business Logic, SSBL,** may be exploited by taking into account the context of the user interaction or without it. In the former case, it can be used to share the user experience with many users for example in control room and decision support contexts.

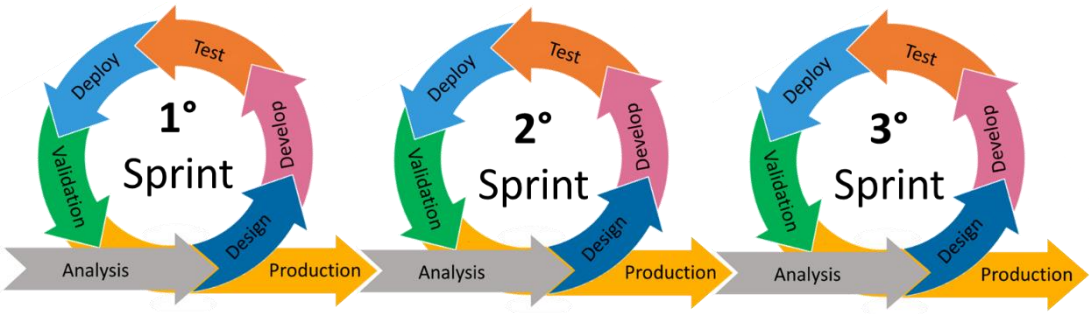
For CSBL see: **https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf**

Thus, on Snap4City, the implementation of smart applications and solutions can be realized by using microservices provided by the functional areas of **Figure 1** and taking into account non-functional requirements. The development framework must enable developers to create advanced applications, and at the same time to support operators to keep the infrastructure under control, despite the complexity generated by multiple applications that may share the same platform, data, data driven and event driven processes, data analytic, and users' tools at the same time.

The **system interoperability** with other solutions can be implemented via a number of API and formats that in Snap4City are more than 150:

- **DA/DS (data input and actions)**: In Snap4City, Https://www.snap4city.org/65
  - API of IoT Brokers of any kind, push/pull,
  - API for data ingestion and data submission, data actions,
  - API for GIS, satellites, BIM, etc.,
  - API for AI, predictions, ODM, Heatmaps, etc. and any MLOPs developed algorithms
  - etc.
- **Front-End API** can provide access to Storage, Data Analytics and Business Logic to:
  - **C1/C2** API which are called Smart City API to query data from the storage,
    - **In push via WS for example.**
  - **E1/E2** API for Server-Side Business Logic, with WebSocket connections
  - **A1/A2** API with brokers to send data directly on broker, and to recover data from brokers.
    - **Broker, IoT Broker** API, IoT Devices / Entity Instances, IoT Edge Devices / Edge processing tools and devices, etc.
    - **In push via WS for example.**
  - **G1/G2** API for exploiting the Data Analytics of any kind. They are typically exploited via **API and call back for Client-Side Business logic**,
  - **J1/J2 API from external services also accessible via CSBL**
  - **Any external applications**: web and mobile App, digital signage, navigators,
- **Processing Logic interoperability, which is in Node-RED + Snap4City Libraries of MicroServices:**
  - Any API with **third party services and applications, which is also a connection with J1/J2**
  - **API and services of DA/DS** as above described,
  - **Front-End API** as above described.

The development life cycle for Snap4City solutions is Agile, and can be function to develop simple and complex applications composed by tens of dashboards and thousands of data sources, and several IoT App according to the different purpose.



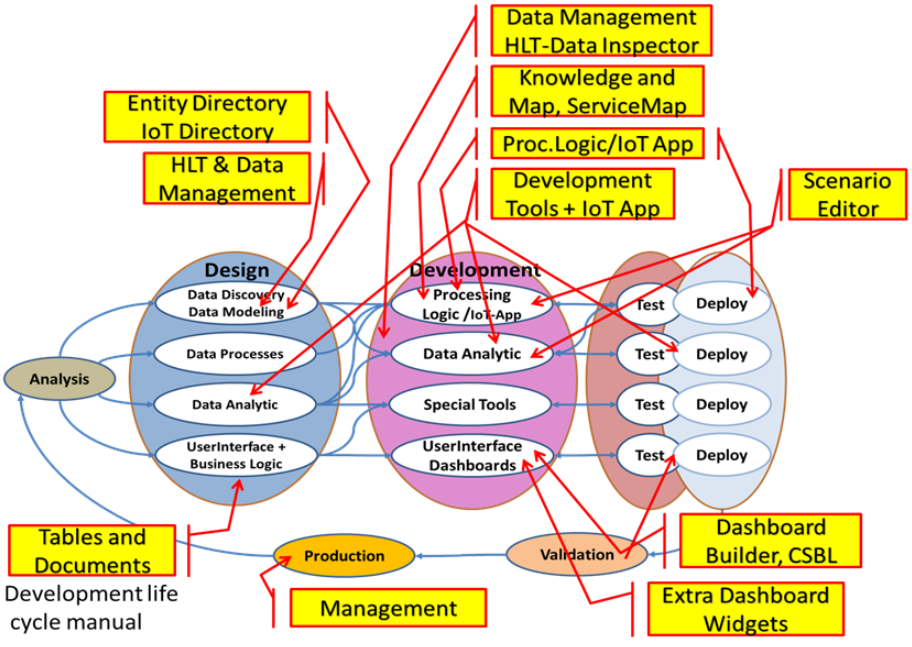
Agile life cycle for developing smart applications on Snap4City, 3 Sprints, not limited to…

Each round of Analysis, Design, Develop, Test, Deploy, Validation and Production can be performed in parallel on different parts and sub-solutions with some interchanges for Data, Data Analytic, User Interface since the developers are typically specialized on different aspects.

Analysis:

Is the phase in which the comprehension of the problem/subproblem, extraction of requirements and definition of the main elements for the specification is performed. Please take in mind that the analysis phase can be performed by subsystems or just incremental for adding new feature to your former solution. Thus, do not be afraid to produce exhaustive requirements, or scenarios, or use cases, focus on the most relevant and start design them and developed the prototype with those elements. The analysis is performed by

- **Performing Innovation Matrix by domain: mobility, energy, parking, gov services, etc.**
- **Entity Identification**: which is the **Dictionary** to be used into the Use Cases and Scenarios. Classic entities and specialized entities are respectively as follows.
- **Scenarios** describing the application/task, textual definition, with some standard table as UML. Refer to Entities
- **Use Cases** describing the different cases into the single applications, by using UML formalization, there are specific Use Cases for each Scenario. Please focus on the most relevant, those that are adding value to your solutions.
- **Requirements** by using standard tables, using identified Dictionary of Entities, prioritizing them, setting mandatory/preferred/optional, functional and non-functional, first/second/third release, etc.
- **Sequence Diagrams:** for some of the critical aspects describing the interaction among them, who is starting the dialogue, which entities are involved, etc. UML diagrams are suitable



A single cycle of the Agile Development life cycle for Snap4City Solutions

Design of the Several Aspects:

- **Data Discovery**: identification of suitable data sources.
- **Data Modelling Design**: only data analysis and modelling.
- **Data Processes Design**, including: data ingestion, transformation, production, publication, etc. This phase is performed by identifying Data Analytics / Transformations. It is a relevant steps since it allows to focus on the functional aspects. In Snap4City this aspect is strongly simplified since the functional aspects are all developed via three main tools:
  - **Proc.Logic/IoT App** which are Node-RED flow with Snap4City libraries:
    - Data Ingestion, integration
    - Data Transformation (save, retrieve, and integration, etc.)
  - **Data Analytics** developed in Rstudio or Python and then IoT App for management
- **User Interface Design**: dashboard design, and user interaction design
  - **Server Side Business Logic** → developed as processes into IoT App.
  - **Client Side Business Logic** → JavaScript on specific Dashboard Widgets only for skilled developers of Snap4City Platform. We suggest first prototype by using Server Side Business Logic, then pass to Client Side Business Logic in JavaScript.

Development of the Several Aspects

Data Ingestion process can be performed in several different manners with the aim of taking data, may be converting them in some of the above Data Models and storing them on some Storage and directly bring them on Graphic User Interface if possible

Proc.Logic/IoT App Development is the activity to implement data flows via visual programming tool. It allow compounding a large collection of macros/nodes (Node-RED Nodes) and most of them refer to MicroServices which are: the standard nodes provided into the Node-RED editor also called IoT App Editor in the context of Snap4City. Those provided by Snap4City (more than 180 MicroServices, which can be installed directly from the Node-RED library, repository of Palette); any other MicroServices/Node from other Libraries of Node-RED nodes and from the web since Node-RED is very diffuse. Thus IoT App allows to develop processes to perform Data_

- **Extract**
  - **Data ingestion:** more than 70 protocols IOT and Industry 4.0, web Scraping, external services, any protocol database, etc.
  - Access to Smart City Entities and exploitation of Smart City Services: transport, parking, POI, KPI, personal data, scenarios, etc.
  - Web scraping
- **Transform**:
  - Data Transformation/transcoding: binary, hexadecimal, XML, JSON, String, any format
  - Manipulation of complex data: heatmaps, scenarios, typical time trend, multi series, calendar, maps, etc.
- **Store and Retrieval**
  - Data access: save/retrieve data, query search on expert system, geo-reverse solution, search on expert system Km4City ontology, etc.
- **Publishing and share**
  - Integration with any service, servers, provider, gateway, brokers, etc.
  - Integration: CKAN, Web Scraping, FTP, Copernicus satellite, Twitter Vigilance, Workflow OpenMaint, Digital Twin BIMServer, any external service REST Call, etc.
- **Data Analytic management**
  - Data Analytic: managing Python native, calling and scheduling Python/Rstudio containers as snap4city microservices (predictions, anomaly detection, statistics, etc.)
  - This can be performed from CSBL
- **Interoperability, establishing connections with other services for:**
  - Providing data: very simple call to external API from IoT APP
  - Expositing data, for example by providing an API interface to get temporary or final data. Those data could be also provided via API and API can be generated from IoT App using HTTP blocks/nodes. Alternatively, the exchange data status can be realized directly into the IoT Device data with a status variable, and IoT Device Status can be red from Smart City API or even from IoT

Device Broker getting the current context status.

- **Business Logic (Server Side) of GUI**
  - 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
  - Generate HTML pages, and collect data entry from those HTML pages
- **Event Management and production:**
  - Create a listener to get event driven
  - Send/post messages to: Telegram, Twitter, Facebook, SMS, emails, etc.
- **Hardware Specific Devices:**
  - Getting local IoT Data from Raspberry Pi, Android, Philips, video wall management, etc.
  - Sending events to HW specific such as: audio, vibrations, SMS, etc.

Data Analytic Development means to exploit collected and accessible data to produce data hints: descriptive analysis, prescriptions, predictions, early warning, anomaly detections, suggestions, heatmaps, recommendations, decision support, optimisation, simulation, routing, classification, detection, video processing, etc.

- **All these kinds of tools can be exploited from Proc.Logic/IoT App and directly from Dashboard by using CSBL**

Graphic User Interface as Dashboard Development means to develop Snap4City Dashboards which are composed by several graphical widgets accessing to: Storage, IoT App data/nodes, External Services, and Brokers. The dashboard can be

- **Passive Dashboards**: showing data taken from Storage only, no actions toward IoT App
  - Passive dashboards may have Selectors, maps, etc., and a lot of visualization without changing the status on Server, no sending commands to the Server Side.
- **Active Dashboards**, which can be those sending or receiving commands to/from the logic coded somehow and in particular for
  - Client Side Business Logic → JavaScript on specific Dashboard Widgets
  - Server Side Business Logic on Proc.Logic/IoT Apps with Snap4City Dashboard Nodes
  - Both kinds of Business Logic may be active

The Active Dashboards are used to implement Business Intelligence solutions with high interactivity and the possibility of changing the data and the representation of data on the Dashboard dynamically on the basis of the user actions. Examples are:

- the click on some button on Dashboard to ask for a computation of some Data Analytics to be shown as a result on the same or other dashboard as event drive actions.
- To select some data and perform a query showing them on map and barseries, pie, multitrend, etc.
- To move a slider and see the light on dashboard changing
- To filter data from a certain time windows, and see the changes also on other data representations.
- Etc.

Testing and Deploy:

It is the phase in which all the needed testing are performed. Unit testing, regression testing, etc. The test is different for the different development parts. The Proc.Logic / IoT App testing is performed once deployed the Proc.Logic / IoT App, the flow is saved and the Proc.Logic / IoT App can be tested using Inject and Debug functions, but also controlling the effects to the external services of the Proc.Logic / IoT App via corresponding microservices.

Please note that the Snap4City development environment allows to have on the same environment applications and solutions which are both in phase of development and finalized. In alternative you can use provided Snap4City Development environment which can be installed in a single VM or via Docker compose.

Validation on Production are the final steps.

Validation is the verification of the solution wrt complete use cases, the process should be performed by different personnel with respect to the developers which also performed the testing and deploy. For this phase actual data are used and actual connector and user interfaces. It is the final phase, which is reached after the acceptance testing, thus the specific smart solution is put in production and made accessible for the users.

3.10.7 – Client-Side Business Logic on Dashboards

when the users expect to play with the data rendered on dashboards for implementing some business intelligence tool, their experience and the evolving data represented in the interface according to their activities is going to change. Those aspects are personal, and context based as one expects to have on a smart application, leading to an evolving user interface that should have a CSBL. In Snap4City, the development of CSBL can be implemented by adding JavaScript functions embedded into the graphic widgets as call back actions, which can perform actions on other widgets and on the platform, and also on third party services, such as a REST Call to API.

To develop this phase one has to follow the Dashboard descriptions performed in the design phase. And at the same time, he/she has to answer questions such as:

- The user interface has to provide some dynamic changes on the basis of the users' actions? Which kind of changes? (CSBL would be need in the affirmative case)
- How many users are using it? (CSBL would be need in the multiple users performing their own private analysis)
- Is it a view for the control room and decision makers of a business intelligence tool for playing with data and solutions? (SSBL would be need if the idea is to create a dashboard in which all the users are going to see the evolution of data even if they are requested by only one of them: sharing the experience or the view)
- Etc.

How to proceed: The developers on Snap4City can visually create dashboards with a drag and drop tool by using Dashboard Builder which is assisted by a number of tools:

- **Wizard** to match data with widgets. It is an expert system for immediate matching HLT (High Level Types: IoT Devices, heatmaps, traffic flows, POI, ODM, etc.) data vs graphics representation for creating Dashboards by rendering and acting on data with a large range of graphics widgets, which may have intelligence in the back by means of:
  - **SSBL**: Processing Logic (IoT App) on data flow combining powerful Microservices/nodes, Data Analytics and API, to implement SSBL.
  - **CSBL**: implemented as visual programming and JavaScript on specific Dashboard Widgets as described in the following of the life cycle document mentioned above.
- **Custom Widget** production tool for creating new widgets and Synoptics by using visual tools and templates: for real time rendering data on graphical scenographic tools, and for graphic interaction on the systems from dashboard to actuators through end-to-end secure connection.
- **Style Theme** modeling for deciding which style to use on the front-end dashboards. It is easy to change the graphical theme and style of the dashboard to have your precise fitting on your applications and portals.
- **External Services** for integrating external tools via SSBL / Proc.Logic and/or via IFRAME into an External Content Widget, or directly calling them as rest API or other means from CSBL.

The list of graphical widgets available in Snap4City to compose the user interface is accessible at the following link: https://www.snap4city.org/download/video/course/p2/

In Snap4City, there is a specific tutorial for the Dashboard development with several examples: https://www.snap4city.org/download/video/course/p2/

Thus, in CSBL we have IN, OUT and IN/OUT widgets (quite similar to SSBL):

- **IN Widgets** are those that are prepared to **receive some actions/commands** (inputs) from the Users or from other widgets as events/messages/commands. For example, a click on a button, a click on the map, etc. They are in some way passive, in the sense that they do not produce any action on other widgets. They do not have Custom CSBL JavaScript inside.

- **OUT Widgets** are those that are prepared to produce **send some messages / triggers to other widgets**. For example, a view of a bar series on some other data, a rendering of a time series, a rendering of a set of Entities on the map, etc. They can be active in the sense that they are capable to produce actions on other widgets, via the Custom CSBL JavaScript they have inside.
- **IN/OUT Widgets** are those that provide capabilities of both IN and OUT Widgets. For example, a map can receive an IN commands to show a PIN, and can send an OUT command to show the data of a selected set of PINs on some barseries.

In the development of the CSBL, two modalities are possible: (i) mixt of visual programming and JavaScript, (ii) fully JavaScript (also called former full CSBL). The JavaScript may be activated by events / actions performed by the User on the GUI of the Widget, and may also send messages/events on other widgets as well perform some API call to Snap4City services or to external services of any kind. While the OUT Widgets are ready to receive commands/messages from CSBL JavaScript (other widgets), similarly to what they do when receiving commands from SSBL via Web Socket to perform actions on the widget or any other JavaScript action/activity.

The Developers that would like to develop CSBL have to be singularly authorized, please ask to snap4city@disit.org, and in MicroX local installation ask to us for the location on your local database for the activation of the functionality to your users. They will be entitled to go in the widget More Options tab and to find and edit a the CK editor to insert the JavaScript code according to this manual, and examples.

When working in **SSBL**, widgets can be created and edited from Node-RED Processing Logic. When working in a CSBL context, widgets have to be created through the Dashboard Wizard as well as from New Dashboard Wizard which provides much more capabilities, as shown in *Figure*.

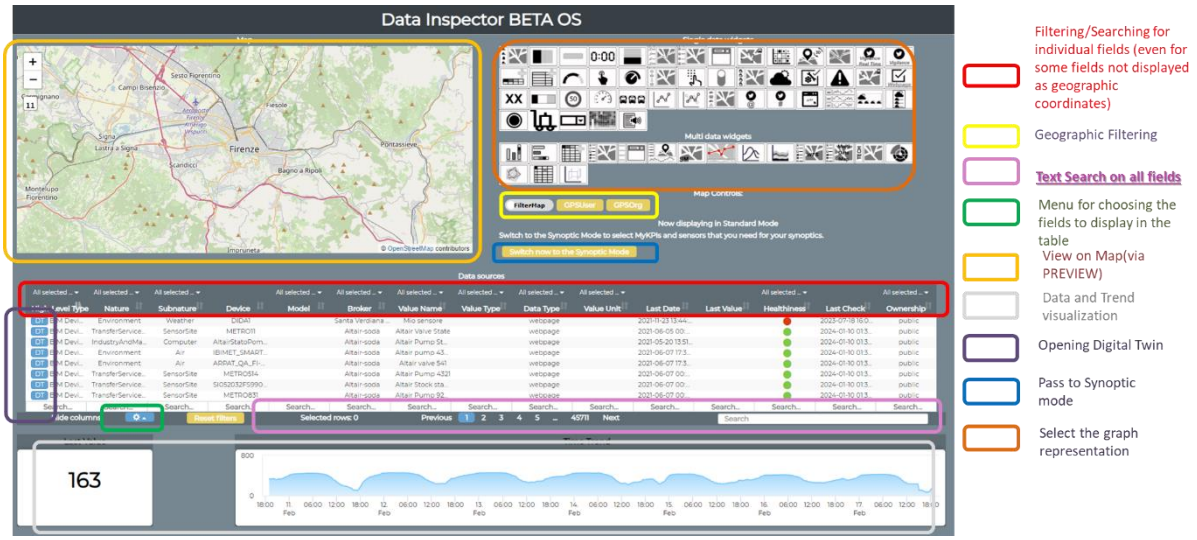


figure : Dashboard Wizard of Snap4City, former and newer. The Dashboard wizard is also accessible as: (i) Data Inspector to search and navigate on data of any kind, (ii) tools for selecting variables for Synoptics panels.

The desired widget type can be chosen from the up-right section of the Wizard showing the available widgets. The desired data source can be chosen from the "Data Sources" table, exploiting the filters on column headers, as well as the text search box in the bottom-right corner of the same table. When the desired choices have been performed, by clicking on the "Next" button a final "Check and Summary" tab is shown for final check. At last, clicking on the "Create widget" magic wand the desired widget(s) are created in the current user dashboard.

In this way, the created widget(s) are in the form of IN widgets, and they are ready to receive and show data and actions by JavaScript, as described in the table "Commands which are ready to execute from JavaScript" in the next pages.

In order to create OUT widgets (those widgets which have this capability are listed in the following table "*Users' Action Description and effects*"), an authorized developer can add the desired JavaScript code in the CK Editor box in the widget "More Options" box, as shown in *Figure 2*. The JavaScript code should be provided as a single JavaScript function named "execute". Please refer to the specific user manual for more detailed instructions.

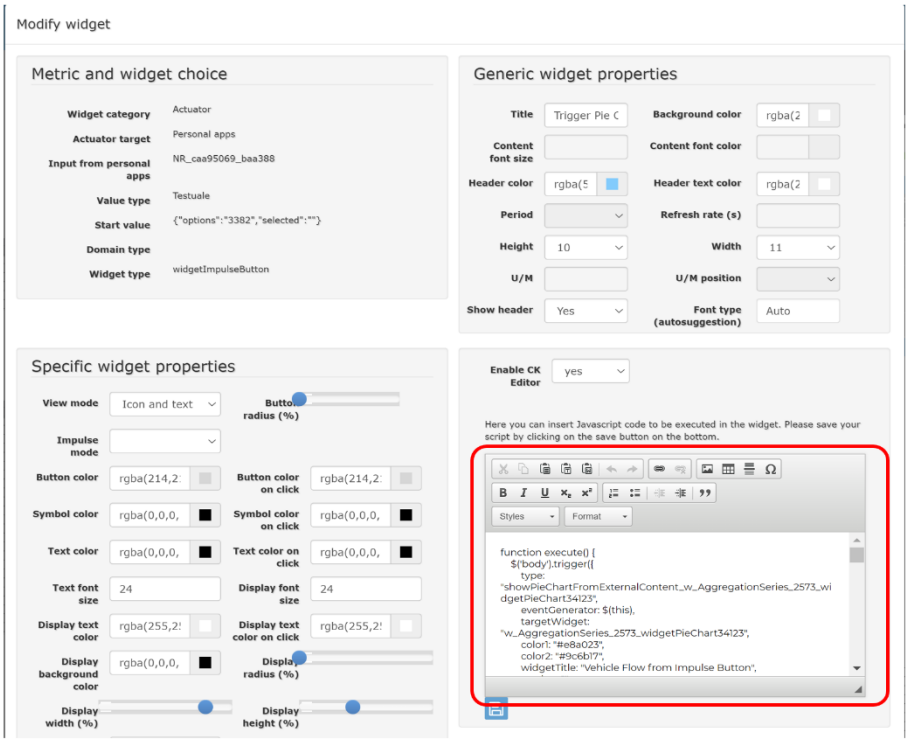
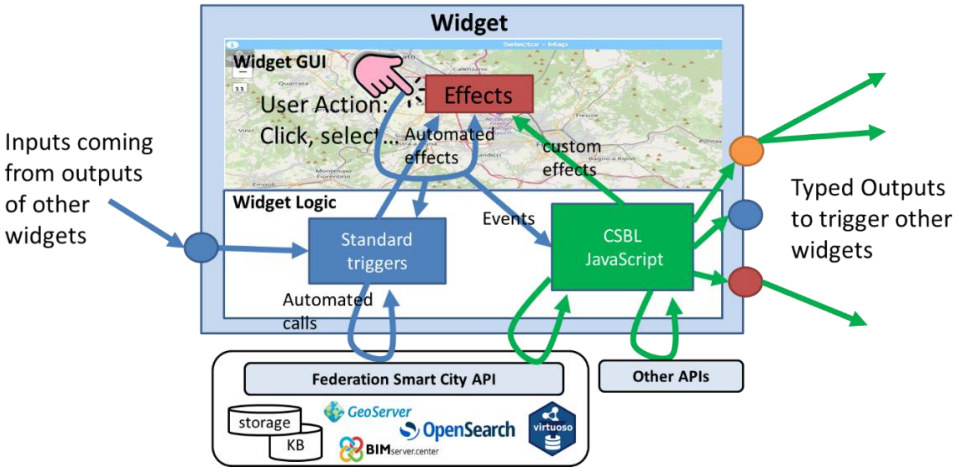
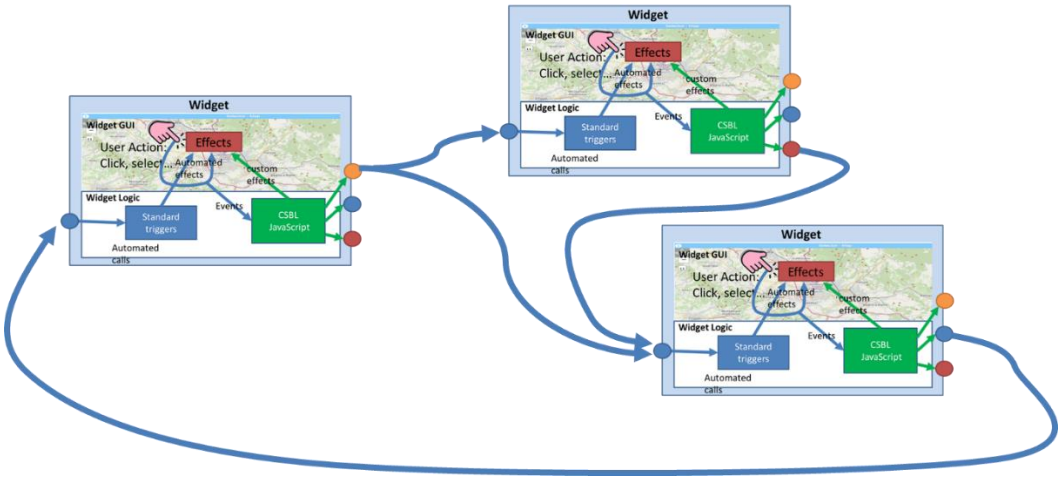


Figure : More Options of Widget with activated CK Editor for CSBL.

OUT and IN/OUT Widgets which present the possibility of scripting in JavaScript when an action is performed on their graphic user interface are reported in the following table. The performed action by the user provokes the activation of a call back that can be filled in the JavaScript editor of the Widget to formalize the action to be performed. At the moment in which an action is triggered, a number of parameters can be provided. For example, geographic coordinates can be passed at a click on map, etc. Into the JavaScript, the developer can code how this information can be used to command IN Widgets, and also REST Calls to Smart City API and other activities.

In general, a widget may receive commands/events (IN) (from other widgets (triggers) and from the user) and may send commands/events to the GUI part of the widget automatically and to the CSBL JavaScript part for custom actions/effects. Moreover, the received inputs commands and events my provoke changes on its own representation as well as to other widgets by sending events, triggers commands outside. The following figure represents a schema.





Figure– The CSBL concepts on Widgets: single and composition.

The inputs received by a widget (coming from others) provoke standard triggers and thus automated effects on the GUI counterpart of the widget. This means that standard triggers cannot be used to execute CSBL JavaScript and custom effects in the same widget in which they are received, but the Effect on GUI are constrained to be compliant with the standard automated effects. In summary: IN widgets provide only blue and red parts of the above figure; pure OUT widgets provide red and green; IN/OUT provide all parts.

This approach avoids the possibility to perform indefinite loops among widgets. On the other hand, it is not a limitation on logic since each widget can control any number of other widgets with its outputs. This mechanism allows the developers to create business intelligence tools in which the action on some GUI Widget (e.g., click, drill down, zoom on map, etc.) can produce a result directly shown on other widgets.

Outputs are usually typed, in the sense that the JSON in out should be compliant with some standard message format, plus other custom messages can be defined as well.

The OUT widgets provide space for Custom JavaScript editable via CKEditor with the following structure, in which is evident that the widget is going to execute function **execute (),** in which the Events may be processed with specific JavaScript segments. The CSBL JavaScript is activated by Events coming from the Users' Actions. The CSBL JavaScript segment may include:

- calls to ASCAPI (Advanced Smart City API of Snap4City), usage of AJAX, authentications, etc.
- call to external API, usage of AJAX, authentications, etc.
- production of Typed Outputs to trigger other widgets (one or more) of the dashboard provided that the widget IDentifier is known.
- commands to open other dashboards/views passing also parameters to them.

In the Snap4City tools there are two modalities of producing Business Intelligence dashboards. In particular they are called:

- **CSBL Visual Editor:** create the CSBL JavaScript in assisted manner via CSBL Visual editor in which the structure of the code and the connections / messages are automatically predefined and generated. **See Section 3.**
- **Full Custom CSBL:** create the CSBL JavaScript according to the user manual reported in **Sections 4, 5**, etc.

Please note that not all widgets provide IN/OUT capabilities. Some of them are limited to IN other to OUT and most of them are IN/OUT. The following two Tables provide you a short summary. If you need to have more IN/OUT CSBL functionalities on widgets, please send an email to snap4city@disit.org

The Snap4City **CSBL Visual Editor** is a graphical interface which is integrated in the Snap4City Dashboard Builder to make easier to implement business intelligence dashboards and thus smart applications. CSBL Visual Editor has been designed to reduce the coding for producing high valuable results and it is accessible for non-programming skilled users. On t the other hand it also enables fully skilled JavaScript programmers to exploit more functionalities, by scripting, and may be passing at the Full Custom CSBL to exploit full potentialities of snap4City Dashboard Builder.

The **CSBL Visual Editor** can be opened in the dashboard page (from the edit mode), by clicking on the "CSBL Editor" button in the upper menu bar. A detail of the graphic interface of the CSBL Editor is displayed in **Figure**. In the CSBL Visual Editor, each node of the flow chart represents a widget of the dashboard (identified by its unique ID displayed in the "WidgetName" box). In these nodes/widgets the user can write the JavaScript code (in the "Code" box) that implements the desired CSBL for each event related to the specific widget (shown in the "Event" box), so that each event and action of the current widget can trigger the execution of a specific JavaScript code (check **Section 2.1** for a list of the available events). Therefore, changing the event selection in the "Event" changes also the visualization of the specific code associated to that event, which is displayed in the "Code" box.

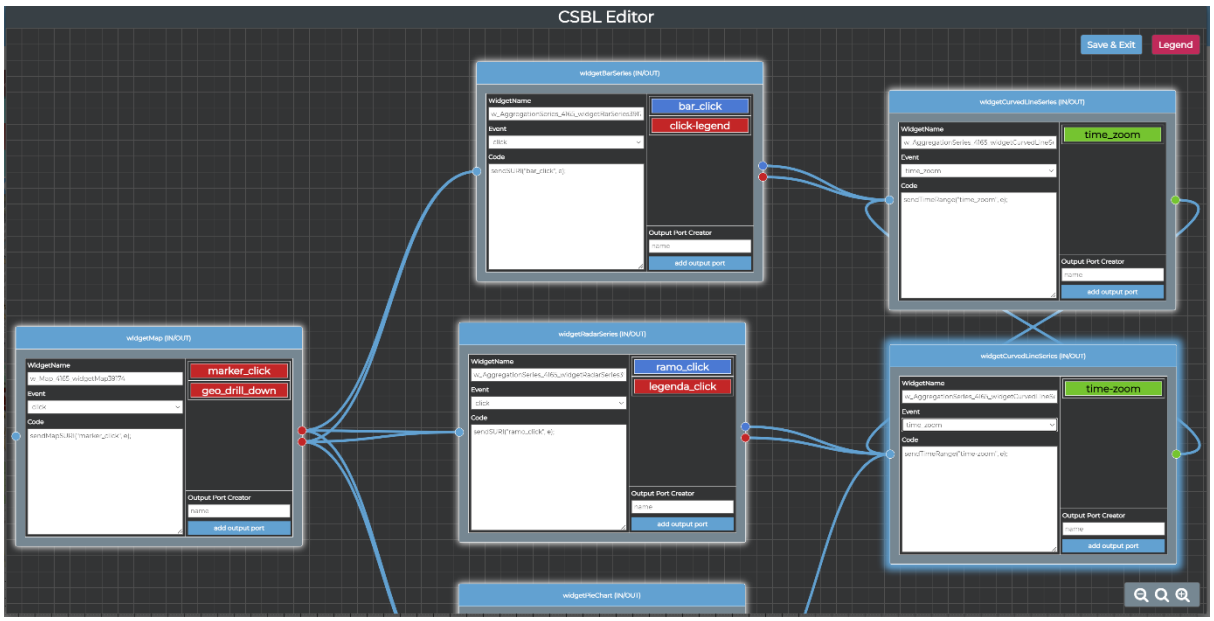


Figure – The CSBL Visual Editor Graphical User Interface: a number of single widget view

3.11 – Snap4City Development Environments and Training

Snap4City.org is a development and training environment which you can learn on how to develop Snap4City applications and use the Snap4City Platform by exploiting all the Ultimate Solutions developed by the Snap4City team and community. Snap4City.org is a **multitenant full installation of the Snap4City platform** in which several Organizations are hosted, it not the only one. You can replicate the same situation in your premise by using Snap4City open-source versions at your full disposal, license-free. Thus, Snap4City.org is a multi-organization living lab.

The list of Organizations and installations is 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 are for tourism, others for mobility and transport, environmental aspects, others for a fully smart city control room with several domains and smart solutions applied, and others again for verticals such as smart parking, smart lighting, waste, and managing mobile applications, etc.

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

See https://www.snap4city.org/661 to get the list of the Snap4City installations that are known by us. Snap4City installations have to use the corresponding logo linked to Https://www.snap4city.org at the bottom right of each dashboard and on the main page of the service. This is the only Legal constraint that you have to satisfy for providing credit to us.

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

- **Dashboards** and **Widgets on Dashboards, widget collections, and D3 widgets.** Business Intelligence facility via IoT App is accessible for all. While the Business Intelligence development tool via JavaScript on Dashboard is accessible for selected developers via specific agreements.
- **Resource Manager:** to search for solutions, share resources, save back up of their solutions.
- **Synoptics**, **Custom Widgets (custom widgets may need to install open source SVG editor on the local computer), custom PINs for animated and dynamic PIN on Maps.**
- **Proc.Logic/IoT Apps as Node-red + Snap4City Libraries**: for connectors, integrations, data ingestion, data transformation, business logic behind Dashboards, business intelligence, and services. Also managing versioning on GitHub of their flows. Snap4City Node-RED Libraries distributed via https://flows.nodered.org/
- **IoT Brokers** and connectors of any kind full features of FIWARE-based and MQTT solutions, but not limited to these solutions and protocols.
- **Entity/IoT Directory for IoT Network and WoT management and abstracting from Brokers, Devices.** Automated deployment of Orion Brokers is authorized only to specific users.
- **Entity Models/IoT Device Models, and SDK to develop code to be placed on physical IoT Devices based on Android, Raspberry Pi, Arduino, ESP32, Windows, and Linux.**
- **External services as web pages to be integrated into Dashboards;**
- **External services accessible via API, to be integrated as API, and MicroServices in IoT Apps;**
- **Web and Mobile Apps: with standard SDK for Apache Cordova to be installed on local computer. Other solutions are also viable.**
- **OD Manager, and server** in multiple geo areas.
- **HeatMap Manager and server:** for automated production of heatmaps and their distribution via other GIS: ArcGIS for example.
- **Heatmaps production in** multiple geo areas (provided in open source and optimally deployed).
- **predictions production in LSTM and Prophet** (provided in open source and optimally deployed).

- **Data Analytics** for development ML/AI, simulations/Optimisation, in various languages: RStudio, Python, also transformed in API services and MicroServices for IoT Apps. RStudio can call segments in Python, Java, etc. (some of the DA are provided in source code others are optional). Jupiter Hub is optionally installed, it is an open source as the RStudio.
- **Living Lab environment** for accessing to training information, video, and examples; discussing solutions, sharing experiences, optionally based on Drupal open source.
- **Workflows on IoT Apps and OpenMaint/BPM, also integrated.** OpenMain is open source and optionally installed.
- **Report generator:** at levels of users, devices, dashboards, and solutions, for several Organizations, the management is provided only to specific users for each organization. (optional)
- **BIM models and data, created and edited in standard IFC and then integrated into an online BIM Server**, (Optional).
- **Traffic Flow Manager, and Traffic Flow Reconstruction (optional) in** multiple geo areas.
- **TV Cam Manager** for managing view streams from RTSP, ONVIF via Kurento and Turn, and reporting them on the dashboard as WebRTC (Optional).
- **Routing** based on GraphHopper for What-If analysis and constrained routing on Multi Data map, in multiple but limited geo areas, supporting What-If analysis tools on Snap4City.org. (Optional)
- **Routing** and **MultiModal Routing** based on Snap4City tools, in multiple but limited geo areas. (Optional)
- **Satellite data download and automated reformatting in heatmaps in** multiple geo areas (Optional).
- **API accounting tools Snap4City API Manager,** limited to projects only, (Optional).
- **MLOps support in advanced ClearML**, to manage clusters of CPU/GPU, HPC (Optional)
- **Converters from OSM to KB, Service Map, SSM2ORION**, (Optional).
- **Chat manager to discuss on Dashboard events online**, (Optional).
- **OpenData Gate, CKAN**, (Optional).
- **Engager for Mobile Apps,** limited to projects only. [UserBehavior2020] (Optional)
- **Twitter Vigilance daily and/or Real-time at 5 minutes, in multiple languages (Italian, France, Spanish, Greek, English, and any Unicode charset),** totally as a service basis in any area. [**TwitterVigilance2017**] (optional, and limited to the access at X APIs)
- **Data Table Loader and POI Loader**, for several Organizations, the management is provided only to specific users for each organization. (deprecated since it can be better realized with other tools)
- **Web Scraping**, and Crawling also transformed into API services and MicroServices for IoT App. (deprecated since it can be better realized with other tools)

When is Optional means that additional tools have to be installed with respect to the initial installation of the Snap4City. Do not the process is not complex.

Online training is accessible from Https://www.snap4city.org/944

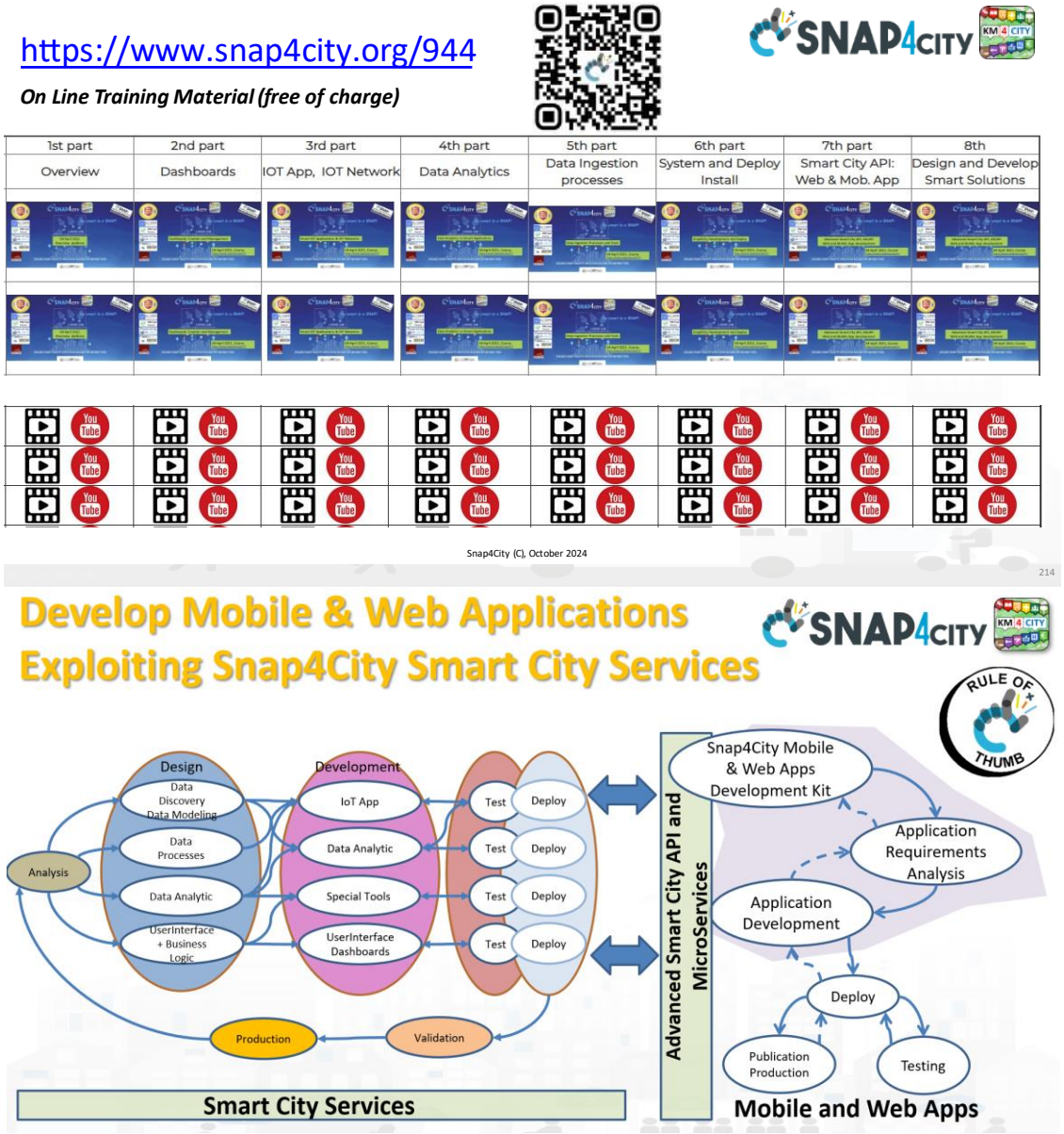
From the training slides you can see a large number of real scenarios with real dashboards, IoT Apps, business intelligence solutions, full applications and Data Analytics in place.

You can access the Training Course Slides in PDF and interactive formats, and video (not so often updated). The version in Slide is the most updated, typically once per month. The date of the last update is reported on top of the table with access to the PDF files.

A lot of new material and improvements have been deployed and distributed, so that, you can see the differences by comparing the present slides concerning the videos.

The whole Snap4City course or segment in live with a tutor is provided on demand. So that multiple videos of different versions in different languages are accessible. When the course is given, several practical exercises are requested to be personally performed exploiting the online platform, and the teachers assess the proficiency of the participants based on questionnaires and by assessing the artifacts produced on the platform. Examples of certificates of participation and/or proficiency are accessible at: https://www.snap4city.org/622 which are provided on request to those that passed the live course. 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 in the **Training Course**, several parts, and a general overview in **Part 6** https://www.snap4city.org/download/video/course/p6/



The **course structure is detailed on page** Https://www.snap4city.org/944 as follow:

- **first part** provides an overview of the Snap4City platform.
- **Second part** provides a detailed view of 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, and 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 detail 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, and 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, Living Lab managers, trainers, installers, etc.
- **seventh part** describes the smart city APIs and how they can be exploited for developing Web and Mobile Apps.
- **Eight part** describes how to design and develop smart solutions via snap4tech framework.

3.12 – Snap4City Security

Snap4City platform has successfully passed PENtest performed by SETEK and Vulnerability Test performed by Thales Italy in 2019, and since that time made again periodically. **Corrective Actions** performed on the **Snap4City** main platform are reported on https://www.snap4city.org/669. The security of the **Snap4City** solution has been validated in international top journals 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's 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

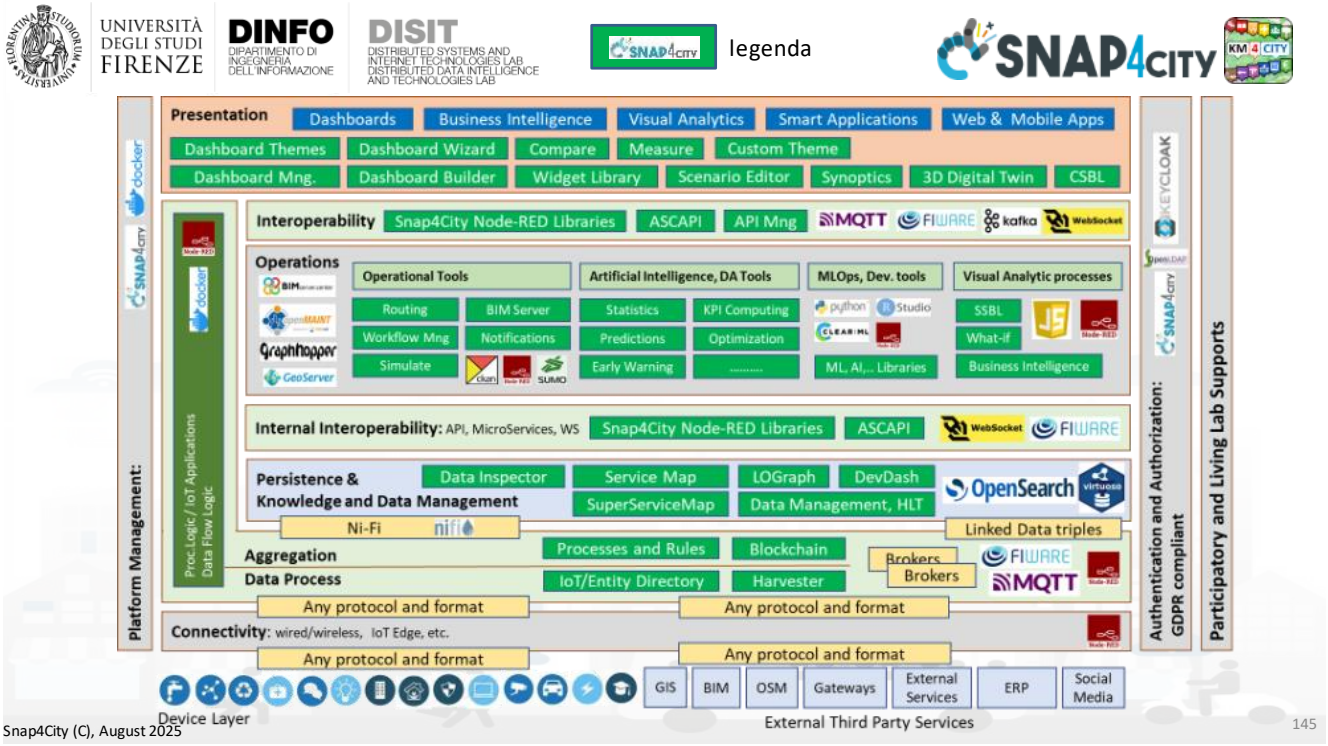
Snap4City platform has been designed and it is GDPR compliant as an IoT/IoE platform. In Snap4City, all devices start as **personal private IoT Devices,** such as Mobile App data, personal device data, SmartBeds data, Car data, hearth monitoring, glucometers, etc. They may become public or accessible to one or many users, groups, organizations, based on 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 control the data exploitation when they are distributed. GDPR aspects are presented on Snap4City at https://www.snap4city.org/670

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

3.14 – Snap4City Deployed Architecture for Scalability and Control

Snap4City platform can be installed in several different configurations exploiting the core modules plus some additional tools among those listed above of the Snap4City.org platform. The different configurations may be suitable from small cities/industries and students to large cities with a high number of users and huge volume of data and data analytics flows. The different modules are provided as VM (Virtual Machines) Appliances or Dockers (the Docker-based solution is typically the most complete, and depends on the moment since there are updated alternately, please ask and see the date of their publication accessible from the portal). The different Snap4City configurations and each component mentioned in the following are described in detail with the instructions for their installation on page: https://www.snap4city.org/738 which is Docker based.

Each VM is an appliance with several components that can be updated directly from GitHub/DISIT. https://github.com/disit



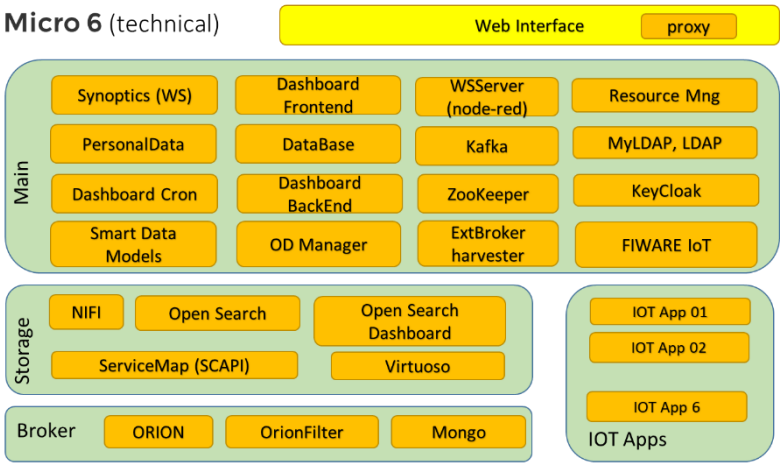
All the components in Green are Snap4City Modules which are familiar to you if you have read all the pages of this document before this one. See https://www.snap4city.org/661 get the list of the Snap4City installations that are known by us. Snap4City installations have to use the corresponding logo linked to Https://www.snap4city.org at the bottom right of each dashboard and on the main page of the service.

The last version is based on VM and Containers, all accessible from https://www.snap4city.org/738:

- **Docker Based Configuration tool** documentation: https://www.snap4city.org/738
  - **The docker/Kubernetes based installation is simplified by a tool which generates the installation filed for you.** https://www.snap4city.org/docker-generator/selecting_model
  - MicroX troubleshooting ENG:
    - https://www.snap4city.org/download/video/TroubleShooting-MicroX-on-premise-en.pdf
  - Monitoring Tool for MicroX --> MicroX Sentinel:

        - Sentinel Activation and manual: https://www.snap4city.org/drupal/sites/default/files/files/Snap4Sentinel-Documentation-for-users-v0-7.pdf
        - https://www.snap4city.org/download/video/MIcroX-monitoring tool with images-v0-1.pdf
    - How To add a Node to OpenSearch: https://www.snap4city.org/download/video/snap4city_elasticsearch_add_node.pdf
    - Note on Kubernetes: if you decide to generate the version for Kubernetes, please download and follow the instructions into the single folders of the installation files provided
        - download the notes: https://www.snap4city.org/download/video/Snap4City-Generation-Kubernetes-V0-1.pdf
- Docker configuration tool helps you in a few steps to get the Docker Compose files for the installation
    - https://www.snap4city.org/docker-generator/selecting_model
- VM-based version is the so-called MAIN version 1.6 (almost deprecated): https://www.snap4city.org/471
    - See for installation https://www.snap4city.org/792
    - and: https://www.snap4city.org/487

For example, a small deployment would be **Micro 6 (X=6):** configurations suitable for solutions for small verticals and industries, single VM. It is similar to the **Alone** configuration as MAIN 1.6, depending on the version, but most of the modules and services are the same. **We suggest that you install the Micro X (6) in a VM with all in with**:



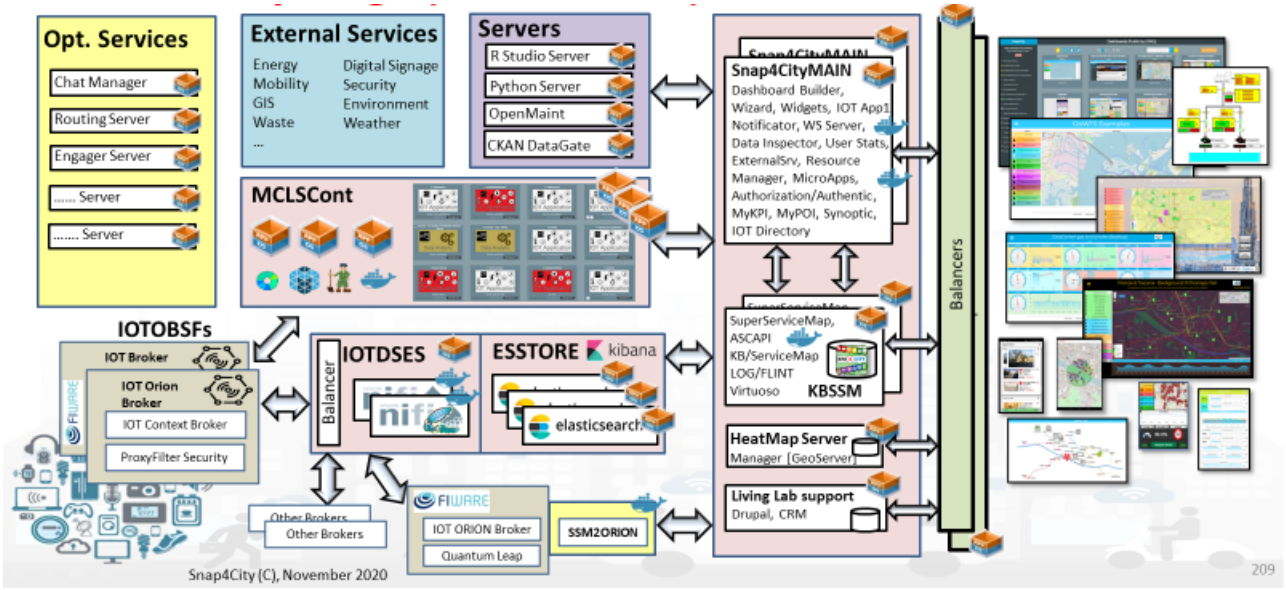
- 32 Gbyte of RAM, more than 50-200 Gbyte of HD (this depends on the data size you would like to have, we suggest 500 GByte in thin provisioning), and 24 cores or virtual cores, faster than 2.1GHz each. A larger size may be needed if you (i) get more IoT Apps (Micro X means the X would be the number of IoT Apps you request, (ii) need to have multiple users / developers and processes and storage.
- Estimated Time to
    - install: 3 hours (Micro 3), time estimated when the VM is ready, the Domain is known, the mailing server is known, and the rest is of default.
    - test and validate: 1 day (Micro 3)
- the step by step installation, and troubleshooting are accessible from https://www.snap4city.org/738
- in https://www.snap4city.org/docker-generator/micro_components_x you can get the flow and ports among the services of the MICRO X version

The different versions for the installations are:

- **Micro X:** 1 VM of dockers all in:
    - It is suitable for micro experiments in smart cities and industries mainly
    - Not includes: not heatmap, no OD Manager, no GeoServer, no data table loader
- **Normal X,Y:** 2 VM of dockers:
    - It is suitable for normal experiments in smart city
    - includes: heatmap manager, OD manager, GeoServer, Data Table Loader, POI Loader

  - 1VM for main, 1 VM with X IoT App, Y brokers
- **Small X,Y:** scalable, 4 VM of dockers:
  - suitable for starting a scalable experiment in smart city and industry
  - includes: heatmap manager, OD manager, GeoServer, Data Table Loader, POI Loader
  - 1VM for Main, 1 VM for X IoT App and Y brokers; 1 VM for storage, 1 VM for Authentication and Authorization
- **DataCitySmall X,Y,Z:** scalable, 6 VM of dockers
- **DataCityMid X,Y,Z,T:** scalable, # VM + X/70 VM + Y/3 VM + Z VM + T VM of dockers
- **DataCityLarge:** scalable, depending on your needs

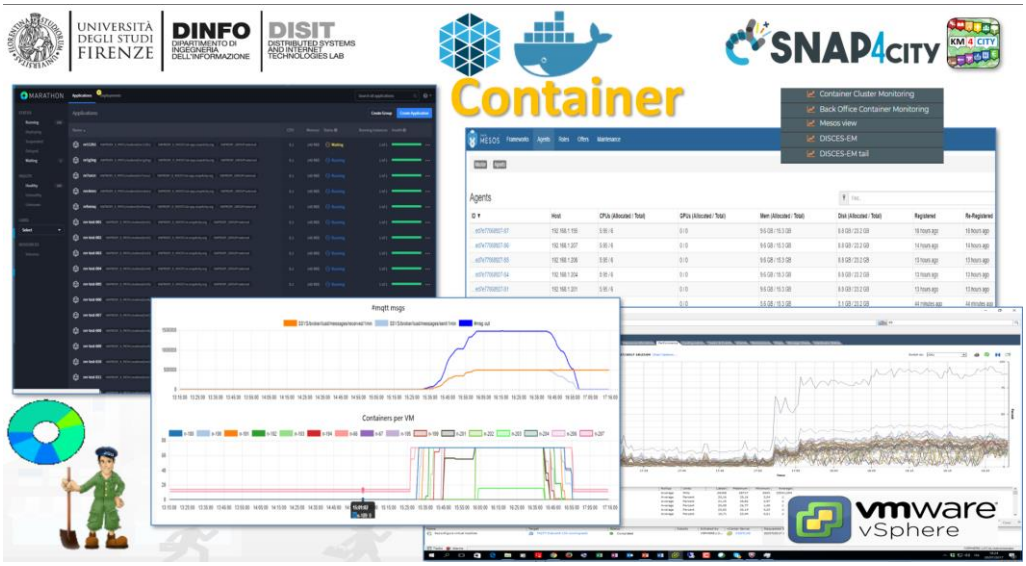
Conceptually a DataCity-Large based on VM is reported in the following figure, and it is mentioned quite similarly into Tutorial **Part 6** https://www.snap4city.org/download/video/course/p6/



No proprietary components are needed to be installed into the Snap4City platform.

We suggest you to start from a MICRO X, or Normal X,Y, or Small X,Y and then increase the number of tools. Most of the tools reported above in VM are actually today "*Dockerized*" so that the installation can be much more flexible and the number of VM reduced.

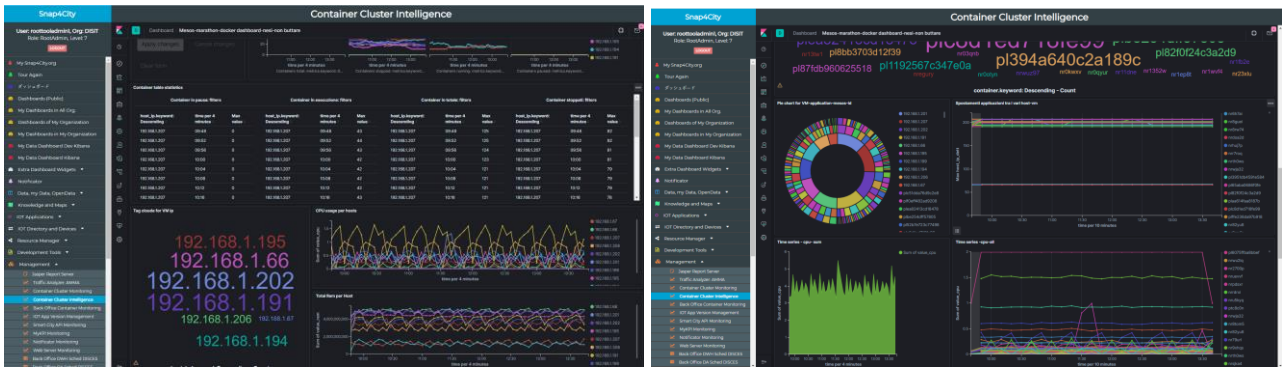
Example of MCLSCount IoT App and Docker cluster Monitoring. The DISCES-EM, which is the DISCES for Elastic Management, is the engine that continuously monitors the Mesos, Marathon, and Vmware resources to decide about the turn on/off of VM from the pool of **MCLSCount** VMs performing elastic management. It also reacts to the eventual failure of the VM by restarting the 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 that 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**: several 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**: the mean amount of CPU consumed in the last minutes in terms of GHz
- **Memory**: the 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, and the ID of the VM corresponds to the IP on the 1.X cluster.

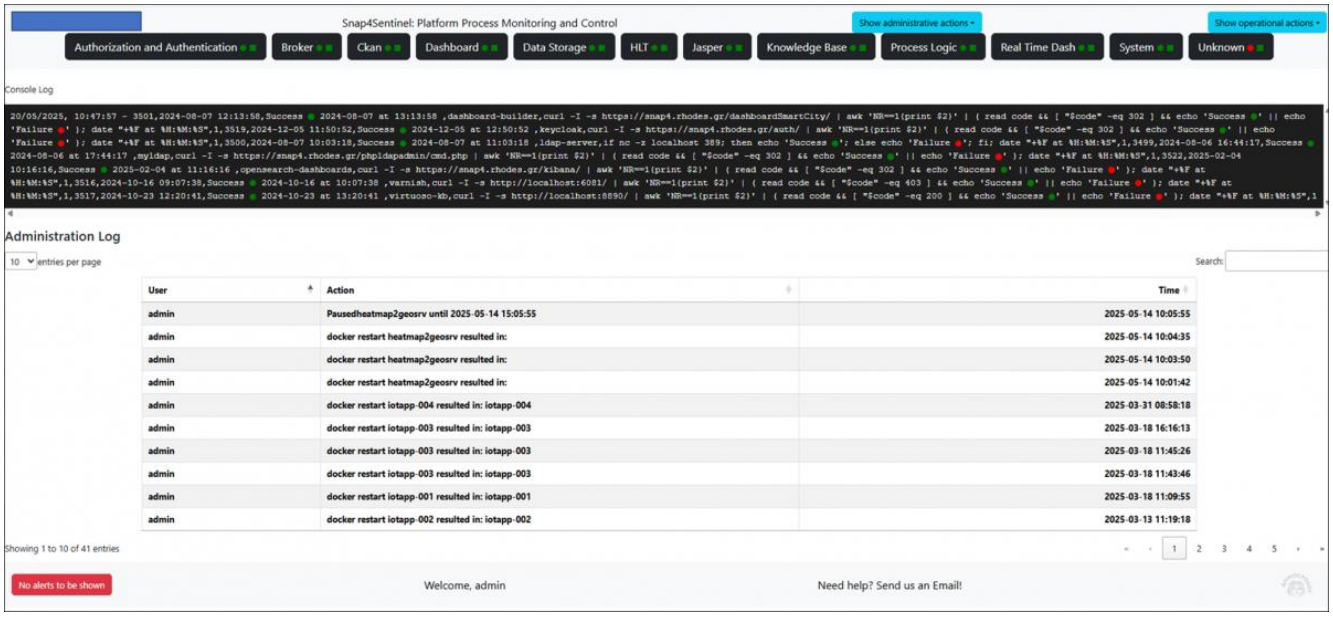
- **CPU usage % graph**: a time trend of the percentage of CPU Usage for each VM that has been activated. The list is dynamic, the ID of the VM correspond to the IP on the 1.X cluster.
- **#Containers graph**: is the time trend of the #Container: Total, Healthy and Unhealthy
- The two buttons: RESET CPU/MEM USAGE and RESET TASKS are used only for resetting the graphics.

Container Cluster intelligence



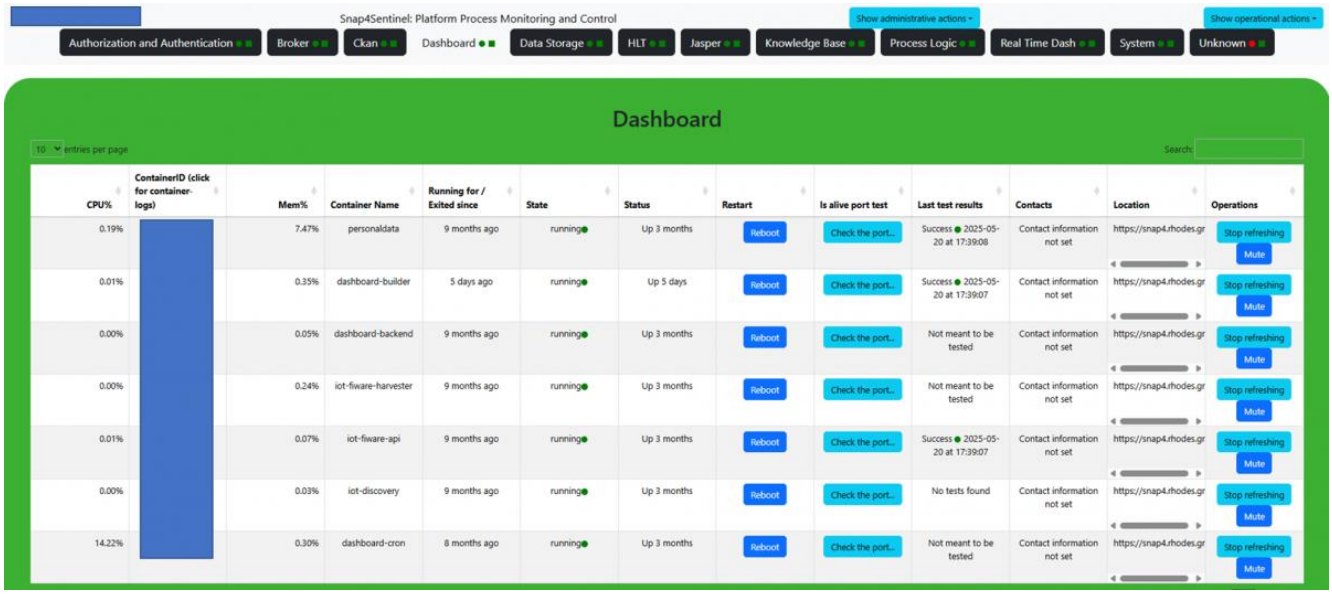
SENTINEL a recently added new tool for monitoring Snap4City platform based on container, Kubernetes, VM, Hosts. https://www.snap4city.org/1101

Sentinel is the tool for troubleshooting, monitoring, performing tests, gathering logs from on installations based on Snap4City technology. This tool is provided in the Micro X Sentinel version. It usually has to be installed in a separate VM accessing to all the infrastructure. The Sentinel tool is available for users that have to manage and supervise the snap4city infrastructure, typically the RootAdmin. By visiting the Sentinel main web page the admin user can see the status of all the containers of the cluster, divided in multiple categories. By default, Sentinel reduces the data gathered by showing the most relevant columns avoiding to provide too information. This can be toggled by accessing the menu “Show operational actions” on the top navbar and clicking on “Show additional columns” (The button changes its text to reflect the toggling operation).



The summary of each category of components on top of the webpage, provides a short description of the current status. A green dot indicates that the category has all of its components in the intended execution status (usually running). A red dot indicates that at least one component is not in the intended execution status. There are also green and red squares representing the results of a “is alive” test.

Clicking on the button of the desired category at the top of the page opens a view showing the details of all active services, with the possibility to check their status, resource usage, and perform actions such as rebooting the service and running an "is alive" check. In order to see the logs of a docker container, the Sentinel tool exposes webpages to see the logs.



Sentinel tool also runs sanity checks on the containers, on startup and periodically. If a container is found not to be its expected status, the backend takes notice of that and takes notice of each test that failed as well. If any unexpected status is found or if any “is alive” test has failed, the backend sends an email to a list of recipients defined in the configuration, then it also sends the same alert with a Telegram Bot to a channel specified in the configuration.

3.14.1 – Snap4City Optional Tools for a typical configuration

In this section, other **VMs/appliances/dockers and data analytics (DA) are listed**, which **are not included in the typical configurations**. Most of them can be obtained on demand, activating specific agreements:

- **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, only the main are listed see the following reference for the full list:**
  - **Snap4City 8/2025 Artificial Intelligence Digital Twin services and tools offering catalogue https://www.snap4city.org/997**
  - **Simulation tools: https://www.snap4city.org/1053**
  - **Applications see: https://www.snap4city.org/4**
  - Traffic light plan optimisation
  - Traffic infrastructure optimization
  - Waste collection optimal planning computation
  - Traffic Flow reconstruction and simulation VM; [TrafficFlowReconstruction2020]
  - ODM production from several kind of sources
  - Computing Trajectories, Typical Trends;
  - Smart light management and traffic depending lighting
  - People Detection and counting via thermal cameras,
  - Wi-Fi user behaviour analysis, DA, [UserBehaviour2017]
  - 15 Min City Index, [15MinCityIndex2021]

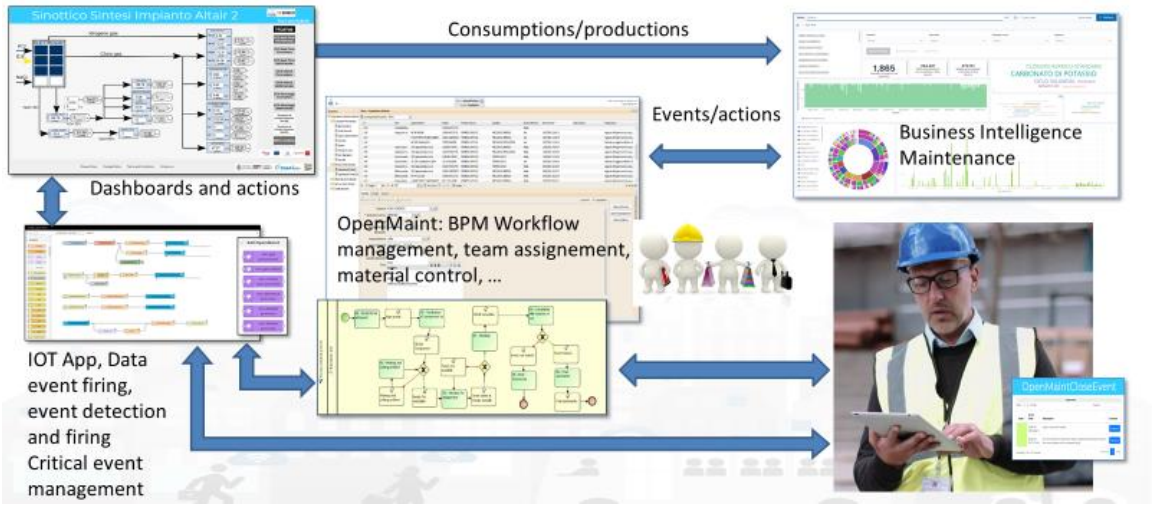
- Business intelligence tools on all data
- CO2 computing from traffic flow and other data, emission factors [EstimatingCO2Emissions]
- Single Camera traffic flow analysis of complex crossroads,
- Predictions and anomaly detections
  - Traffic Flow predictions, [TrafficPredicitons2020]
  - Parking predictions, DA; [SmartParking2018]
  - bike sharing rack predictions DA; [ShortTermPredictionBikeSharing2021], [LongTermPredictionBikeSharing2021]
  - NOX predictions and dense heatmaps production, VM/DA; [NOX prediction 2020]
  - Long Term prediction of NO2, [LongTermPredictionNO2-2021]
  - Predicting Landslide events [PredictingLandslideEvents],
- Social Media Analysis, via Twitter Vigilance,
- COVID-19 data monitoring, DA; [COVID-19]
- 3D production of Digital Twin Global, [GeneratingDigitalTwin]
- GTFS and Transmodel ingestion processes;
- Anomaly Detection, DA;
- SUMO Simulator;

- **Features and tools to be additionally installed:**
  - **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 us 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 in the security aspects.
  - **MLOps as ClearML** advanced, to manage with Snap4City tools the development and the exploitation of AI processes.
  - **Routing Server VM** for routing based on GraphHopper (accessible from github), constrained routing, multimodal routing; Routing Server based on Snap4City solution is accessible from Github.
  - **Smart DS DSS based on System Thinking**; [**SmartDS],** more deeply integrated and distributed in open source.
  - **DORAM tool for the analysis of demand of mobility vs offer of transportation;** Tools, a new DORAM 2 has been released;
  - **Chat manager** for dashboards, specifically for Control Room and Early warning management, VM;
  - **GTFS editor VM/Docker; Covers also formats as also other formats: Kochi, Hyderaband.**
  - **Resilience DS** VM/Docker [**Resilience2017], [Resilience2019].**
  - **Twitter Vigilance VM** in different forms. For early warning and moods assessment: English, Italian, France, and a few other languages; [TwitterVigilance2017]
  - *DISCES Scheduler and cluster VM (deprecated);*
  - *Large file sender in addition to FTP: https://filesender.org/, VM (deprecated, see the integrated File Manager)*
- **Cloud SW:**
  - All Orchestrators and managers for the cloud are not included, such as VMware vSphere, VMware converter, etc.
  - VEEAM software for backup of VMs is not included
  - HUB and Firewall software if any is not included
  - Kubernetes is usually not included, please request we can provide.

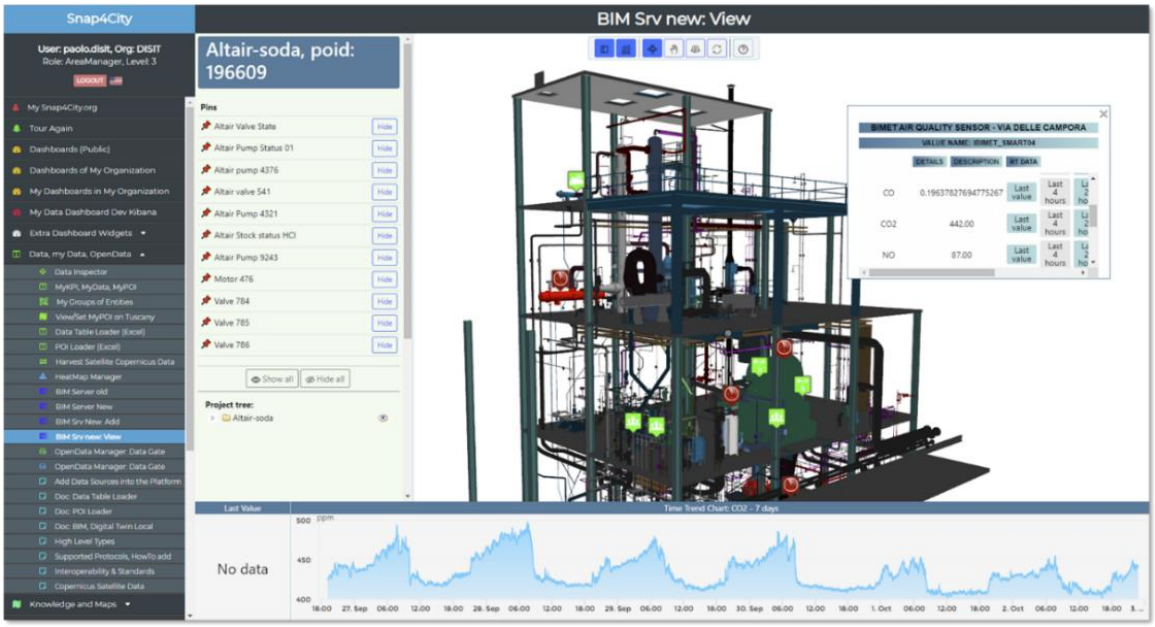
3.15 – Snap4City: Maintenance Management, BPM and BIM

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 feedback and results from teams
    - Manage all phases of the workflow on the fields via Proc.Logic / IoT Apps.
    - The integration if via API and MicroServices into Proc.Logic / IoT Apps.
  - **MicroServices** integrated with Snap4City via Proc.Logic / IoT Apps;
- **Business Intelligence** which is the **Snap4City tool based on** Elastic Search: which works on top of the database of tickets collected on OpenMaint;
- **BIMServer** integration with Snap4City Dashboards.



- 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 tablets).
- 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, to support their decision and operative procedures.
- can be customized 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 remodeling data models, workflows and reports without modifying the code of the application software.

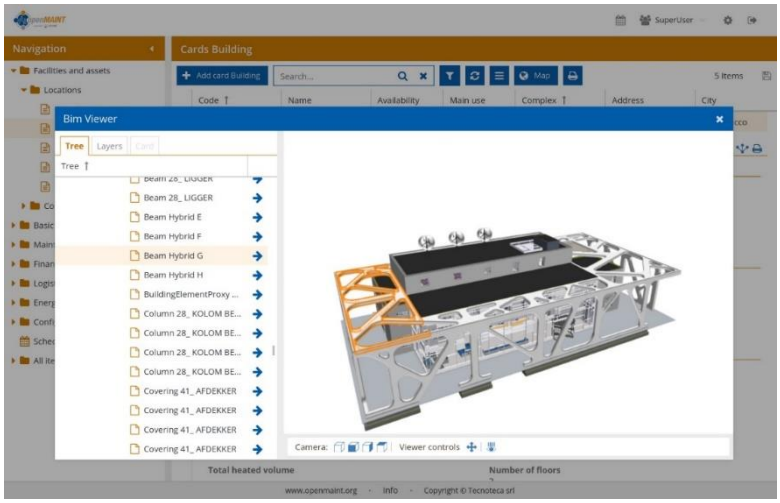
Functional areas of openMAINT includes:

- **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

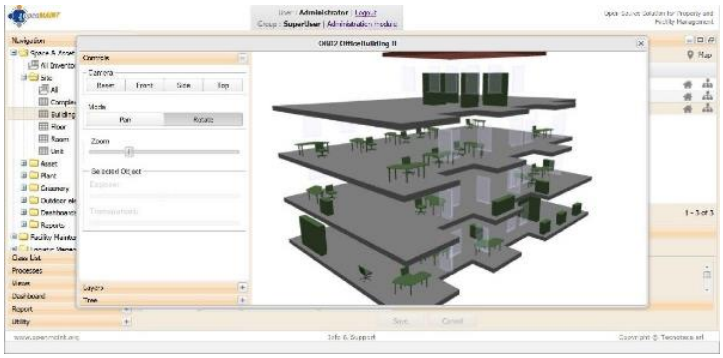
So that you do not need to install OpenMaint to have 3D representation on BIM for the Digital Twin of buildings. Snap4City uses OpenMaint only for collection and maintenance asset management in connection with Snap4City processes and facilities.

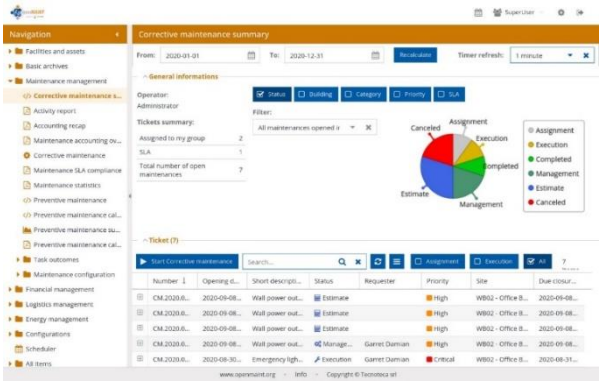
The main features of openMAINT, BMP, and **BIM** are:

- 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 customized access to data
- classifications and lookups which can be autonomously customized
- 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 customized web custom pages
- complete outline of users, groups, permissions, and multitenant configuration
- interoperability with other applications through web services
- CSV data import / export functionalities
- possibility to configure and manage (task manager) automatic operations
- possibility to configure e-mail for notifications and as a 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

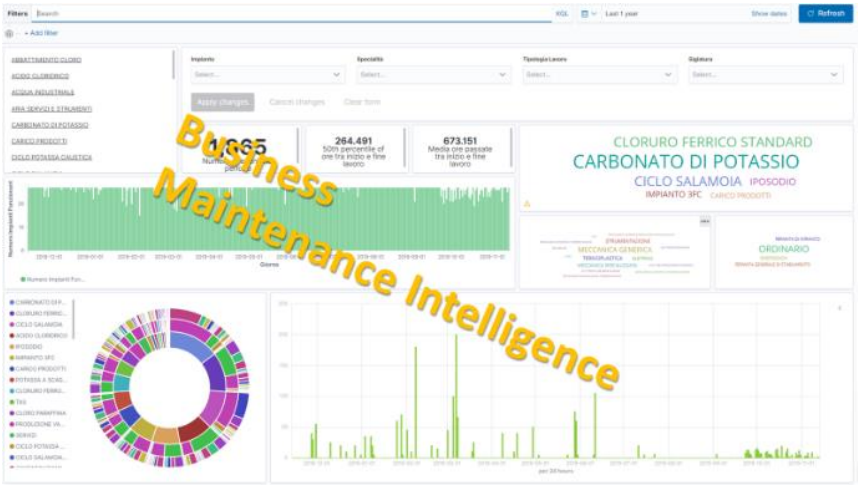




Buildings 3D shapes, and Floors are managed and distributed into the **BIM Server**, standard BIM tools are used for editing and interchange in IFC formats by standard tools such as AutoDesk Revit, etc. Buildings and Floors are shown in **Dashboards** for their integration with maps and IoT Devices' time trends.

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

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



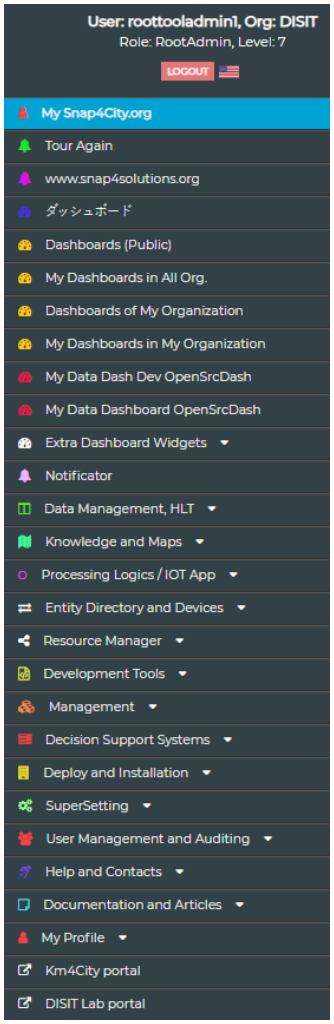
3.16 – Snap4City Back Office: Management Environment and Quality Control

This is just an overview since the actual number and specific tools for user and platform management depend on the size of the platform and of the number of tools installed. See the previously presented SENTINEL tool.

Snap4City has several user roles:

- **Users of any Role** have full control of their own resources: data, devices, dashboards, IoT App, etc., which may control according to GDPR rules, providing access, revoking, etc.
  - **Any new user starts at the lowest role: Manager.**
- **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 such as:
    - control processes, consumption of resources, healthiness, etc.
    - manage tools exploited in your configuration
  - **AreaManager:** Typical developer capabilities, access to development tools, access to a wider number of resources, IoT App with both basic and advanced, IoT Models, etc.
    - They are entitled to get specific access to Rstudio, Python development, Portia Web Scraping, and Business Intelligence script on Dashboards, only when authorized.
  - **Manager:** Final users, limited access to development, IoT App development with the Basic library.
  - **Other Roles may be defined**
- **Specific Accesses and features can be granted by the RootAdmin**
- **All** users have 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, several Groups on Snap4City.org (living lab support Drupal)

  - users of different kinds and may impose early bounds on the resources used by users (IoT Dev, IoT App, Dash)
  - on cloud user kinds up to the level of Tool Administrator
  - One or more ServiceMap and boundaries for the federation



RootAdmin users may

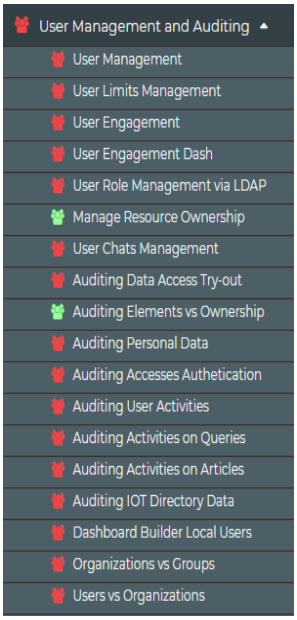
- pass Users to higher roles. Ask snap4city@disit.org to become an AreaManager for testing on Snap4City.org ;
- **Provide**/grant specific authorizations to data access on Tool usage
- In the Installations that you perform on Premise from the Docker compose or from VM, you become the RootAdmin of it, you are going to decide ALL. Full control of all the platforms.

RootAdmin users have a large set of tools in every field

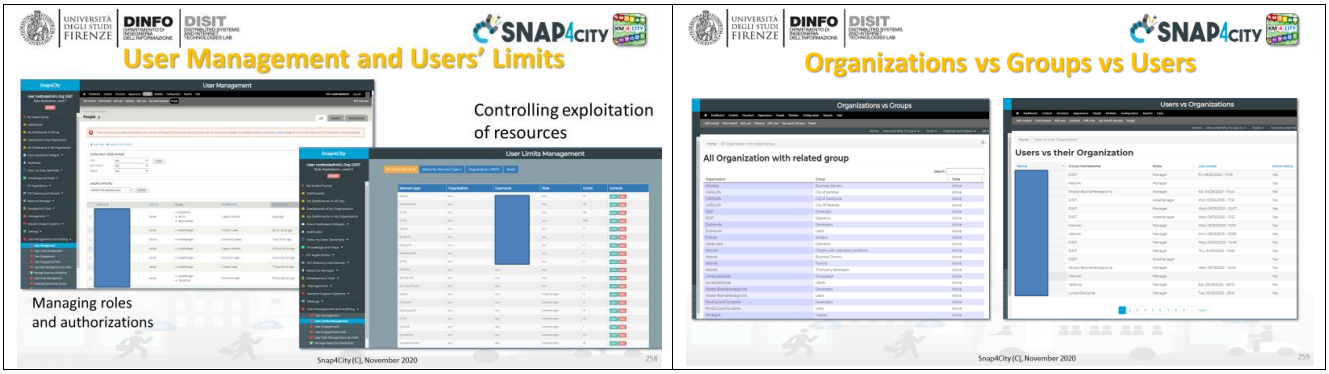
- Dashboards
- Extra Dashboard Widgets
- Notificator (only for solution on premise, and in most cases deprecated since it can be substituted with a the more flexible Proc.Logic/IoT App)
- Data Management, HLT
- Knowledge and Maps
- Processing Logic/IoT App
- Entity Directory and Devices
- Resource Manager
- Development Tools
- Management
- Decision Support Systems
- Deploy and Installation
- Super Settings
- User Management and Auditing
- Help and Contacts
- Documentation and Articles
- My Profile

3.16.1 – Snap4City User Management

The RootAdmin needs to manage:



- **User Management: for managing**
  - accounts and profiles
  - limits of the users in exploiting resources
  - Accesses and special authorizations
  - 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 access**
  - Auditing all the activities on the platform (see next section)
  - Personal auditing

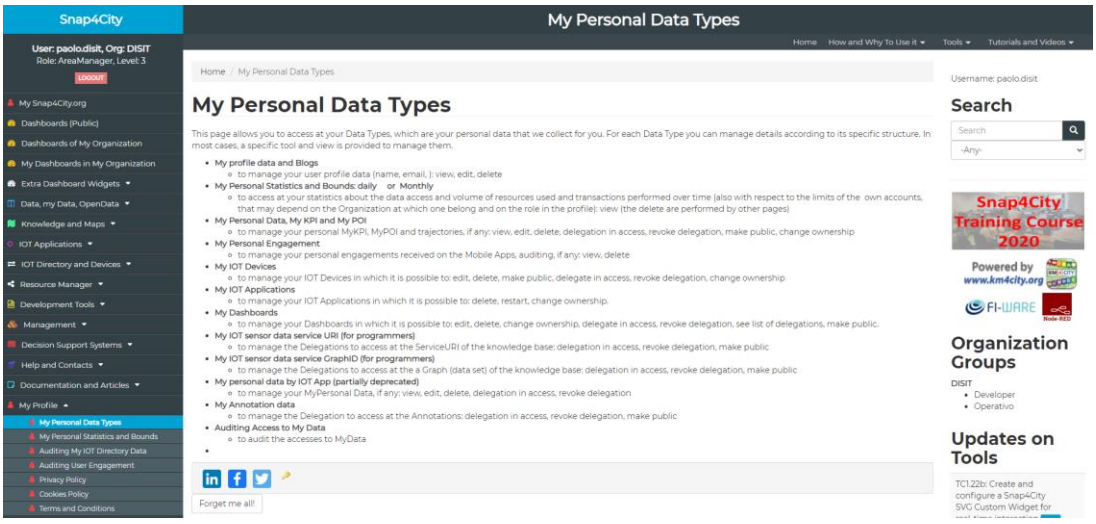


Customer Relationships management:

On CRM as Drupal we have

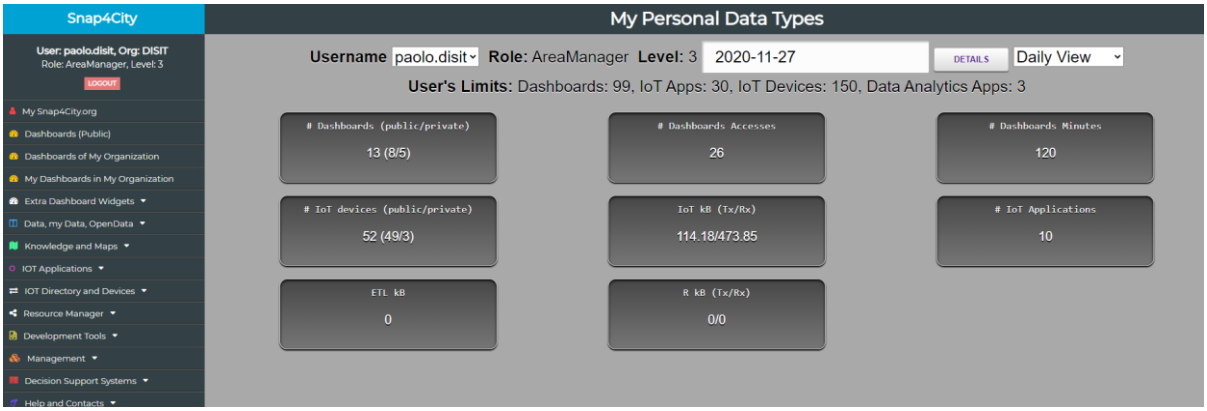
- **User Management** registration and mailing
  - LDAP connection role management
  - KeyCloak Authentication (OpenID Connect)
  - Management of user profile
- **Content management** for Organizations and Groups
  - Indexing of all content and search
  - Distribution of content and Video
  - Reports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

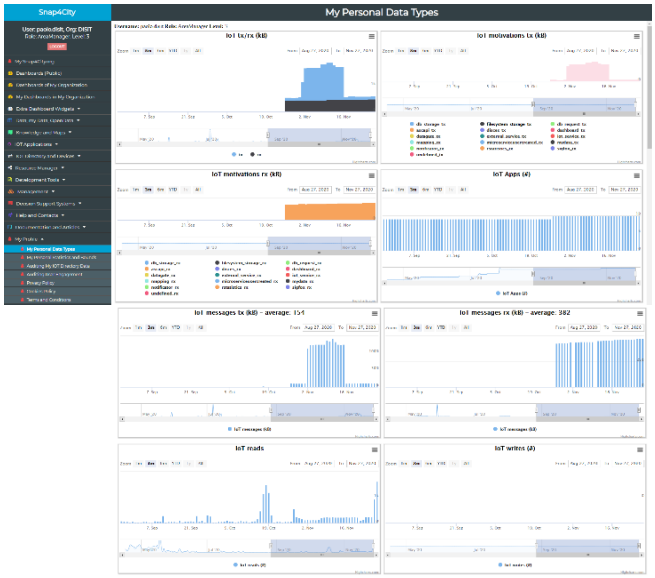
Access to Personal Data types.



Control of resource consumption

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

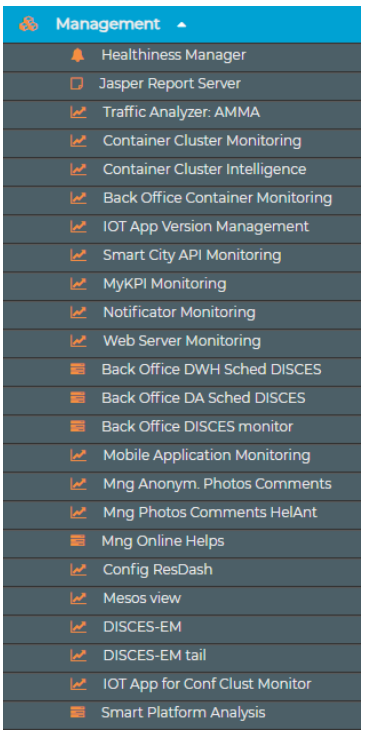




Each user may monitor the consumption of resources concerning the limits imposed on its account.
The user is informed when the limits are reached to stimulate the online acquisition of new resources.

3.16.2 – Snap4City Platform Management, quality control

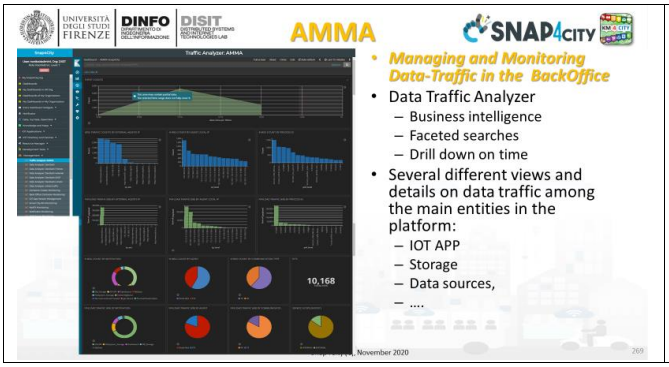
This is just an overview since the actual number and specific tools for user and platform management depend on the size of the platform, and on the kind of platform installation (which can be based on VM, Kubernetes, or a mixt, etc.) and of the number of tools installed. See the previously presented SENTINEL tool.

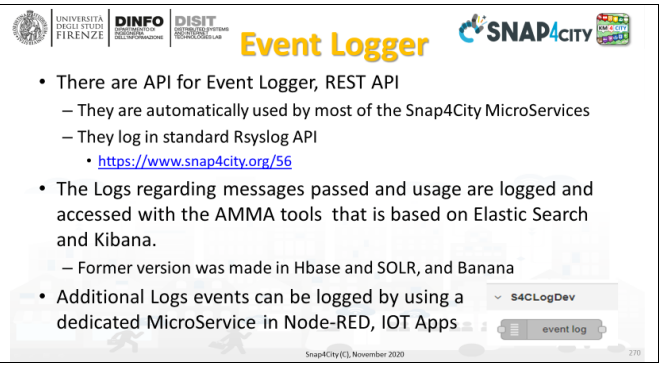


- **Tools for Platform Management.**
  - Most of the tools are only accessible for RootAdmin and OnPremise
- **Tools can be grouped in the following families**
  - DataAnalyzer (**DevDash**): monitoring and browsing data ingested into Elastic Search, via Kibana
  - Container Monitoring and Management
  - IOT App Version Management of Snap4City tools
  - Smart City API traffic monitoring
  - MyKPI Monitoring
  - Mobile Applications Monitoring
  - Management of Images and Comments from Smart City API, Mobile and Web Apps

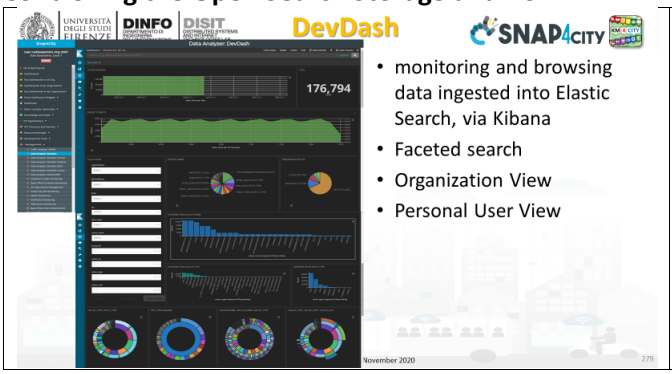
AMMA and Log Analysis, quality control (almost deprecated)

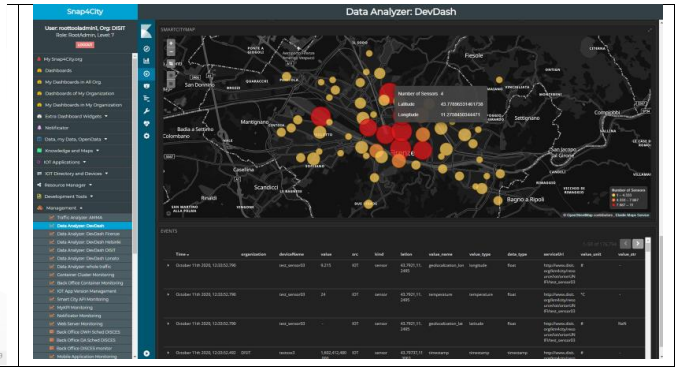
The LOGs are coded in standard **SYSlog** format and can be monitored/accessed by using the **AMMA** tool based on NIFI and Elastic Search, or **LogStash**.



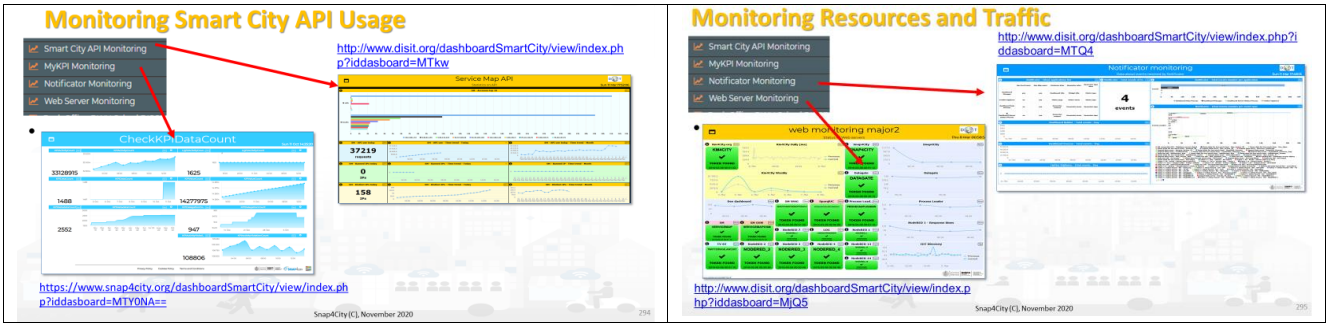


Controlling the Open Search Storage and flow





Monitoring the Resources and API Traffic, quality control

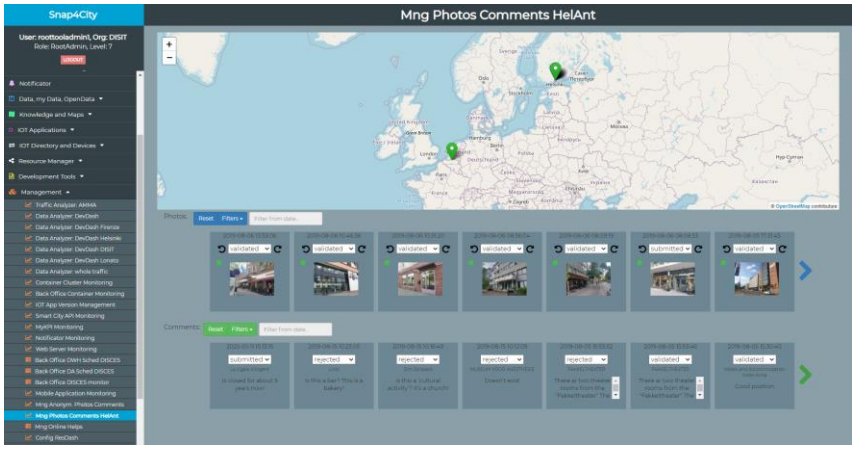


http://www.disit.org/dashboardSmartCity/view/index.php?iddasboard=MTQ4

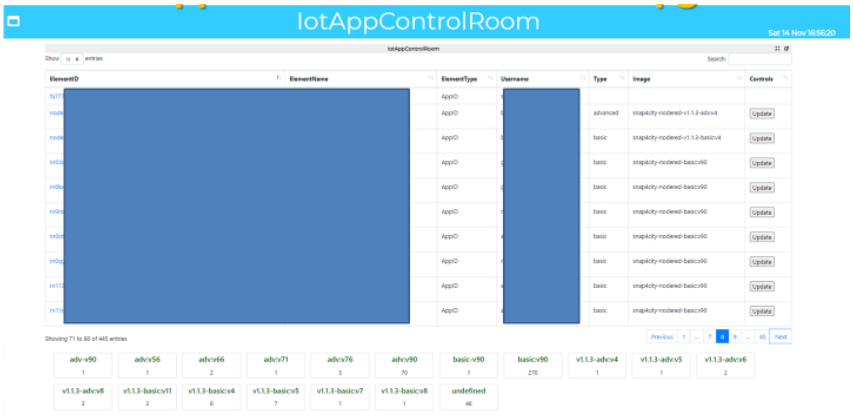
MyKPI monitoring: https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTY0NA==

Smart City API performance: http://www.disit.org/dashboardSmartCity/view/index.php?iddasboard=MTkw

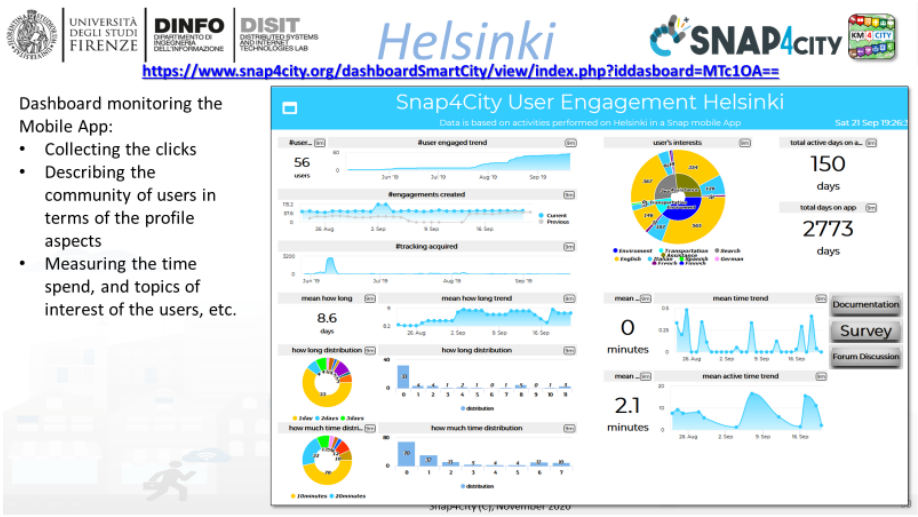
Monitoring Photos and Comments from Web and Mobile App



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:

- **Snap4City Sentinel tool to monitor** processes and tools: https://www.snap4city.org/1101
- **Marathon and Mesos** Monitoring: only internally accessible (see Section 3.14), see also version with Kubernetes.
- **Cloud Services** Monitoring and control: from VMware, only internally accessible (see Section 3.14)
- **API reachability/availability** performed by **E015 of Lombardia Region**: 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 of Kubernetes, IoT Orion Broker and Brokers in general, IoT/Entity Directory, NIFI, Open Search for data shadow, OpenMAINT, User Management, KeyCloak, LDAP, etc.**

Examples of automated activities:

- Healthiness manager (not accessible in all versions)
- 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 device models, device instances and time-series from EXCEL file via Data Table Loader and POI Loader.
- Ingestion and usage of FIWARE Smart Data Model as Snap4City IoT Data Model and knowledge base
- production of widgets from IoT App into Dashboards
- creation of the Data Shadow for IoT modelled data
- production of the widget into dashboards from wizard and templates
- creation of the geo-spatial, temporal, and relational indexing
- creation and deployment of containers from Data Analytics in Rstudio and Python

- Creation and deployment of containers with IoT Apps
- Production of Rest API Call for training developers 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 APIs, scheduler, and dashboard access
- Collection of data regarding user and platform auditing
- Logging of activities performed by IoT App, toward IoT Devices, MyKPI, etc.
- Production of reports
- Production of GeoTiff from a set of point heatmaps, for creating calibrated heatmaps
- Automated fitting of time trend based on 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	N	Y	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	Y	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	Y	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	Y	N	Y	N	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	Y	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 a digital object encapsulating the security identity of a process or thread or 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.
Advanced Smart City API	Advanced Smart City APIs include several APIs in addition to the SCAPI, also those for IoT Directory and IoT management, IoT App management, resource management, etc. They are classified as 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
AI	See Artificial Intelligence
AMMA	Application and MicroService Monitor and Analyzer. A Snap4City tool to perform the analysis of data flows among IoT Apps and several MicroServices. See more on https://www.snap4city.org/198, https://www.snap4city.org/43 In the alternative, standard tools may be used. It is almost deprecated in the last Snap4City versions. (tool deprecated)
API	Snap4City APIs are classified as 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
APIMAN (deprecated)	**APIMan is a** management API tool can be placed in front of the several APIs to (i) provide a unified entry point, (ii) perform accounting on the API consumption, (iii) placing the basis for providing billing on the basis of API usage and data consumption according to different business models: (a) monthly rate, (b) limits, (c) pay per play, etc. https://www.apiman.io/latest/
Snap4City API Manager	**A tools for** management API tool can be placed in front of the several APIs to (i) provide a unified entry point, (ii) perform accounting on the API consumption, (iii) placing the basis for providing billing on the basis of API usage and data consumption according to different business models: (a) monthly rate, (b) limits, (c) pay per play, etc.
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 which has joined lab with the **University of Florence LabGeo** and has been in collaboration with the **University of Florence DISIT Lab for long time on the Snap4City platform**.
Artificial Intelligence	https://en.wikipedia.org/wiki/Artificial_intelligence
ASCAPI	See Advanced Smart City API of Snap4City
AXIS	A top-level producer of TV Cameras and other complex sensors kinds. https://www.axis.com/it-it
BI	Business Intelligence. In Snap4City, the tool for BI is composed of Dashboard Builder, Wizard, IoT App, and for simple direct Dashboards with Kibana.
BIM	Business Information Modelling. In Snap4City, the tool for BIM is performed by BIMServer. https://github.com/opensourceBIM/BIMserver. (optional)
BIM Device	In snap4city they are used to represent and show a 3D building, floor or segment with a specific IoT Device, ServiceURI, or POI attached.
BIM Manager	A Snap4City tool for collecting and managing the relationships among BIM representations and elements of the smart city as ServiceURI. (optional)
BIM Server	A tool for managing BIM representation and distribution. https://github.com/opensourceBIM/BIMserver. (optional)
BIM View	In Snap4City they are used to represent and show a 3D building, floor or segment derived from some IFC format for BIM. They are collected and provided by the **BIM Manager**.(optional)

BPM	Business Process Management. In Snap4City this activity of defining business process workflows is modelled and performed using OpenMAINT tool. See OpenMAINT definition. OpenMaint can be controlled by IoT Apps and vice versa.
Business Logic	The logic of interaction behind a smart application or a Dashboard. For example, what is going to happen at: a certain click on a dashboard button, acting/selecting elements on some widget, selecting elements on some map, etc. The typical actions are Drill Down, Slicing, return back, zooming in space and time, etc. The business logic in Snap4City can be formalized in Proc.Logic/IoT App (SSBL) or JavaScript (CSBL) directly associated with specific Widget of the Dashboard.
Calendar Widget	A widget for showing hourly or daily heatmaps about time series, from the D3.js library
Calibrated Heatmap	Calibrated heatmaps are heatmaps 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 Heatmaps are distributed via a GIS with WMS (for example ArcGIS or GeoServer) and may have billions of points. https://www.snap4city.org/457
Cam Manager	See TV Cam Manager
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 the **DataGate** module and goes with SSO with the rest of tools. CKAN/DataGate can be controlled by IoT Apps. (optional)
Color Map	A color map table is managed in Snap4City via Color Map manager. The Color Maps are used by: Heatmaps, Tables, OD matrices, and Traffic Flows. They may be used by other tools as well.
Connectors	Connectors can be found/implemented in the Snap4City platform using 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, the 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 Containers 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. The installation based on container is accessible from: https://www.snap4city.org/738
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 A more updated way to create Custom Widgets is to use an External Content Widget and load in it HTML/CSS custom widget exploiting the CSBL for exchanging data with the platform 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 (possible but more complex than the CSBL approach): https://www.snap4city.org/153

CSBL	Client Side Business Logic
Client Side Business Logic	Business Logic associated with dashboard widgets to create business intelligence tools and visual intelligence tools, and any kind of optimisation and AI endowed tools and smart applications. See the development manual for CSBL: **https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf**
D3.js	A large library of graphical representations https://d3js.org/ . Accessible on Snap4City Dashboard widget via IoT Applications.
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/course/p2/
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/course/p2/
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 and IoT Apps.
Data Analytic	For Data Analytic we intend all deep data transformations 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 statistics, 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/course/p4/
Data Dictionary	A Snap4City tool that allows to define Value_Type, Value_Unit, and Data_Type Data 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 in the Dictionary.
Data Inspector	In Snap4City, the Data Inspector is the main tool for browsing the information about a data stream entering the platform. It is directly accessible from the main menu and gives 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/course/p5/
Data Shadow	A term adopted to describe the historical data of an IoT Device (sensors and actuators). In most of the platforms (for example: MS Azure, AWS), this feature is optional (most of the Brokers provide only the last values of the IoT Devices). In Snap4City, it is the main feature to save all data messages of devices, and it is implemented saving data into the Elastic Search cluster. See training part 3 https://www.snap4city.org/download/video/course/iot/
Data Table Loader	(deprecated) 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
Data_Type	Each Attribute/variable in the Snap4City platform is defined in terms of Value_Type, Value_Unit and Data_Type (e.g., Energy Power, Kw/h, Float).
DataGate, Data Gate	The module of Snap4City to integrate the solution with CKAN Open Data management (open source) and network of services (optional)
DevDash	A Snap4City tool for monitoring the global flow of data entering into the platform and reaching the Open Search. The new name for us is "My Data Dashboard Dev Kibana" 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 the information associated with each Digital Twin. The Digital Twin can be Local (BIM) and Global (3D modelling of the city buildings)

	A digital counterpart of a physical entity which could be a simple device, object, building or a complex city. The resolution of representation with 3D or less can be disputable.
Digital Twin Global	Typical intended as the Digital Twin of City. The resolution of representation with 3D or less can be disputable.
Digital Twin Local	Typical intended as the Digital Twin of a single Building or large complex local object, a Complex Machine. The resolution of representation with 3D or less can be disputable.
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. It is deprecated in the recent versions of Snap4City. https://www.snap4city.org/236 (deprecated)
DISCES-EM	Distributed scheduler for elastic management for Smart City by Snap4City (deprecated). This tool is used for the elastic management of Containers 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	Docker 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/course/p6/
End-2-End	The solution that allows connecting 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.
Entity Directory	New name for **IoT Directory**
Entity Instance	New name for **IoT Device**
Entity Message	New name for **IoT Device Message**, device message
Entity Model	New name for **IoT Device Model**, smart data model
Entity Variable	New name for **IoT Device Variable**, value, attribute
EOSC	Snap4City is an official platform of the 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 Apps 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.
External Broker	are brokers whichs are not under the control of the same administrator of the platform. They are for example connected to get data from them without having the possibility of browsing them in deep, near of registering devices, etc. Snap4City External Broker Harvester allows to register and get data of Devices which are already register in external brokers
External Broker Harvester	Snap4City External Broker Harvester allows to register and get data of Devices which are already register in external brokers.
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 areas 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 into 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. It is a term used for federated Snap4City infrastructures.
File Manager	A snap4City tool for collecting and managing files of any kind, according to the ownership and GDPR compliance
FIWARE	https://www.fiware.org/
FIWARE broker	See Orion Broker https://fiware-orion.readthedocs.io/en/master/
FIWARE Smart Data Model	IoT models defined by FIWARE, https://www.fiware.org/smart-data-models/
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 several 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 in the HeatMap Server for distributing Heatmaps, Traffic Flows maps, etc., in GeoTiFF format according to tiled. https://www.snap4city.org/536 https://www.snap4city.org/507 GeoTiFF format is used for distributing Orthomaps, Maps 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 to the GeoServer if needed. See https://www.snap4city.org/368 for all the relationships from GIS and Snap4City including interoperability.
Group	A Snap4City Group of Users is a community into an Organization. Grant authorizations to resource access can be provided at the level of a single user, Group and/or Organization.
GTFS	General Transit Feed Specification, https://developers.google.com/transit/gtfs It is a standard file format used to formalize the public transport information, trips, paths, bus stops, schedules, etc. Snap4City is compliant and can ingest GTFS files using ETL processes, the corresponding information is fed into Knowledge Base.
Heatmap	Are maps of points into Heatmap server, and/or directly images representing data in regular and nonregular 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 Manager	HeatMap Server exposes API for (i) collecting 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 protocol (they can be ArcGIS or GeoServer). https://www.snap4city.org/536 https://www.snap4city.org/507
HighCharts	A library for JavaScript graphic representation: https://www.highcharts.com/ This graphic library may have according to some licensing conditions.
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, MyKPIs, personal data, WFS, Complex events, heatmaps, traffic flow, etc. see https://www.snap4city.org/583
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 provides SSO.
Industry 4.0	https://it.wikipedia.org/wiki/Industria_4.0
Innovatrix	A methodology for innovation at the ground of the Snap4City Innovation model: https://www.imec-int.com/en/innovation/innovatrix

integrations	The instrument to implement integrations 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/course/p3/ https://www.snap4city.org/download/video/course/p5/
IoT Adapter	A term adopted in IoT to indicate a remote converter of protocol located between the IoT Device and the IoT Broker. See training part 3 https://www.snap4city.org/download/video/course/p3/
IoT Agent	A term adopted in IoT to indicate a remote converter of protocol located between the IoT Device and the IoT Broker. See training part 3 https://www.snap4city.org/download/video/course/p3/
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/course/p3/
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/course/p3/
IoT Broker	An IoT Broker which may support one or more protocols, typically only one. It can support IoT Adapters, it can implement solutions for MultiTenant and paths, such as IoT Orion Broker and/or MQTT. Other IoT Broker can be: Mosquitto, HiveMQ, etc. https://en.wikipedia.org/wiki/Message_broker Several brokers may implement the same protocol such as https://en.wikipedia.org/wiki/Comparison_of_MQTT_implementations Snap4City is almost agnostic about the brokers and accepts a number of brokers and protocols registered on the IoT Directory of Brokers and Devices. See training part 3 https://www.snap4city.org/download/video/course/p3/
IoT Connector	A component to connect to a service using a protocol. In Snap4City, a large number of connectors and protocols are supported in 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 the IoT Directory that performs 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 produced 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/course/p3/
IoT Device Mobile	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 a short time. For example, 40.000 lights of a smart light system. The IoT Device Model is a template, and once used can be modified without any impact on the produced devices. See training part 3 https://www.snap4city.org/download/video/course/p3/
IoT Directory	Snap4City tool for registering IoT Broker and Devices. Snap4City is almost agnostic about the brokers and accepts several brokers and protocols registered on the 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 manages all the network information about the devices deployed and connected. It is capable to exploit NGSI V1 and V2 protocol aspects to inspect and manage IoT Orion Brokers of FIWARE, so that to manage a large number of them and automatically performing registration of devices in bulk on the 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/course/p3/
IoT Discovery	It is a function of Snap4City IoT Directory and Knowledge base which allow discovering the IoT Devices by many filters: geospatial, by type, by value name, by unit, by nature and sub nature, 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/course/p3/

IoT Edge Device	An IoT Device is capable to execute processes. In Snap4City terminology is typically endowed with the 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/course/p3/
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). To have the Data Shadow, and thus to perform queries on time series, the IoT Orion Broker has to be connected to 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/course/p3/
IoT Orion Broker	See Orion Broker https://fiware-orion.readthedocs.io/en/master/
ISEMC for Video Wall	ISEMC for Video Wall management integrated with IoT App. (https://www.snap4city.org/621)
KB	See Knowledge Base
KeyCloak	For identity 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 the smart cities 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
LIDAR	Lidar (/ˈlaɪdɑːr/, also LIDAR, or LiDAR; sometimes LADAR) is a method for determining ranges (variable distance) by targeting an object or a surface with a laser and measuring the time for the reflected light to return to the receiver. https://en.wikipedia.org/wiki/Lidar#:~:text=Lidar%20(%2F%CB%88la%C9%Aad,to%20return%20to%20the%20receiver.
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 levels of severity (e.g., warning, severe, etc.), they are produced in standard format as **SYSlog**. Those logs can be browsed, and 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 also	Linked Open Graph tool (nothing to do with Logs), LOGraph. LOG.DISIT is a tool for accessing and browsing Linked Open Data in the world and the local Knowledge base, KB. https://log.disit.org/service/

called LOGraph	
LogStash	See Log, https://www.elastic.co/logstash an open-source tool for inspecting logs
Marathon	A Tool Open Source, the standard for the management of containers, in connection with Mesos https://mesosphere.github.io/marathon/
Market Place	Resource Manager of Snap4City https://www.snap4city.org/205
MESOS	A Tool Open Source, the standard for the management of containers, in connection with Marathon. http://mesos.apache.org/
MicroApplications	A snap4City set of views is implemented in HTML5 JavaScript for realizing specific functionalities. They are substantial views of Web and/or Mobile Apps which can be called independently and placed into Dashboard external content Widgets 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 based on the APIs (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 the third-party or 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 in 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.
My Data Dashboard Dev Kibana	Synonym of DevDash.
MyKPI, MyPOI	Snap4City tool for collecting and managing personal KPIs and POIs. 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/
OD Manager	Synonym of Origin-Destination Matrix Manager
Open Search	The new name of Elastic Search fork of AWS https://aws.amazon.com/opensearch-service/the-elk-stack/what-is-opensearch/
Open Search Dashboard	New name of the Kibana fork of AWS https://aws.amazon.com/opensearch-service/the-elk-stack/what-is-opensearch/
OpenMAINT	Workflow Management System, Incident Management, Business Process Management, BIM and GIS integrated with Snap4City. The tool is capable to define workflow by integrating activities of humans and machines with the main focus on maintenance and ticketing. It is presently integrated with the IoT App of Snap4City mainly for workflow management and ticketing management. 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 the level of a single user, Group and/or Organization.
Origin Destination Matrix Manager, ODM Manager	A Snap4City tool for collecting and managing Origin Destination Matrices which are collected according to different kinds of shapes for Origin and Destination: grid, administrative shapes as ACE areas. At each ODM the OD Manager assigns several metadata and a Color Map. They are distributed via OD Server.
Origin Destination Matrix	A map that describes inflow or outflow from certain points or areas over time in some geo area. They are used for describing the flow of people, by cellular data, mobile data, census data, etc.
Orion Broker	The IoT Broker of FIWARE official CEF of the EC. It has NGSI V2 and NGSI LD as native formats. https://fiware-orion.readthedocs.io/en/master/
Orion Context Broker	See Orion Broker https://fiware-orion.readthedocs.io/en/master/
Out of the Box	Component ready to be used without development, already in use in Snap4City applications and/or solutions.
PAX counter	A kind of tools for counting people typically by sniffing Wi-Fi, Bluetooth. They can be also developed by using other solutions as TV cams, laser, radar, etc.
Platform Management	See Quality Assessment and User Management and Control. **See Section 3.16.**
POI	Point of Interest, services on the map, with some GPS location and service classification
POI Loader	(deprecated) POI Loader for shortening all the activities of ingesting new data for points of interest providing them as excel files.
Portia (web scraper)	A Tool for Web Scraping, extracting data from web pages. Processes of Portia in Snap4City are converted to MicroServices executed on containers. https://portia.readthedocs.io/en/latest/index.html
Processing Logic	In Short Proc.Logic. a new name for IoT Applications, IoT App
Python Server	Snap4City Python server for developing Python processes for machine learning, AI and statistical purpose on data. It can access 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.
QR Manager	A tool to produce QR for questionnaires, collect them and processing them using SnapAdvisor LLM.
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. 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: • **Sentinel**: https://www.snap4city.org/1101 • 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 (deprecated) • 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 control on the healthiness of the IoT Devices automatically based on criteria set up on the IoT Directory. In this case, the control is performed at the 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 number of resources needed to adopt it. Quality control can be also performed when the Reports are produced according to a specific KPI which has to be defined and computed. **See Section 3.16.**
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.
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 the level of IoT Devices, Dashboards. Specific Reports can be created to produce quality assessment aspects.
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 artifacts: 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 into 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 of the map at a 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 decide on 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 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/
SCAPI	See Smart City API
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. (the version provided is presently developed in former technology with respect to the last mobile apps in distributions)
SENTINEL	A Snap4City for monitoring Snap4City platform tools and processes, producing alarms, detecting anomalies, providing notifications, etc.
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 a URI in the Linked Data model and Km4City Ontology and Expert system. It allows to identify univocally any entity of the city.
Simulation Manager	A tool for loading simulation and monitoring their progresses. For example SUMO simulations.
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 APIs which are used by Front End Tools such as Dashboards, Web and Mobile Apps, and MicroApplications. Details regarding Smart City API are reported in: https://www.snap4city.org/download/video/course/p7/
Smart City Control Room	A solution for centralized control of the smart city via a set of view walls and operator consoles 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 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. 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
Smart Data Model	Synonym of **FIWARE Smart Data Model,** https://www.fiware.org/smart-data-models/
Smart Data Models	FIWARE smart data model, a set of data models produced by FIWARE foundation as standard for IoT modelling and other applications. Snap4City can use the FIWARE Smart Data Model as IoT Model for data ingestion and management
SmartDS	The **SmartDS** (Smart Decision System) of DISIT is an Advanced System Thinking solution for Decision Support Systems, DSS, on smart city problems and data. SmartDS is a tool presently in a 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).
SnapAdvisor	A LLM Expert at your disposal H24/7 on Snap4City and your private content. SnapAdvisor of Snap4City is a multilingual chat-based virtual AI assistant which can be tailored on your domain by providing a content based on PDF documents, PDF slides, coding, and web pages. SnapAdvisor preserves the privacy since it uses the content you provide to perform temporary in the reserved chat you establish with it, and not use the data/files and chants to perform additional training. It can answer in your language, even if the content has been provided in other languages.
Snap4City API	See API, API Manage, ASCAPI, etc.
Snap4City Innovation Matrix.	The methodology for innovation of Smart City derived from the **Innovatrix** method and it has been adopted for leveraging the Innovation into the European Commission JRC ISPRA in 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/course/p6/
Snap4City IoT App	See IoT App, IoT Application Node-RED process + Snap4City Library of MicroServices
Snap4City Smart Application	Snap4City Smart Application is an application composed by one or more Dashboards in which the business logic is defined into IoT App and/or directly in JavaScript associated with some widget of the Dashboards. They can be realized only by authorized AreaManagers.

Snap4City Smart Dashboard	Synonym of Snap4City Smart Application
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 Devices 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
Snap4Tech	The set of technologies and tools underlined for Snap4City, Snap4Home and Snap4Industry solutions, most of them are described in this document. Who is going to install the Snap4City tools has to provide a reference on their applications and visible web page to a logo "Powered by Snap4Tech" logo. https://www.snap4city.org/296 **see platform installations**: https://www.snap4city.org/661
SSM2ORION	SSM2ORION is 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 us 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 in the security aspects.
SSO	Single Sign On, In Snap4City, this function is performed by LDAP and KeyCloak at which all tools are referring to exploiting OpenID Connect.
SUMO	Simulation of Urban Mobility, an open source tool, which is fully integrated with Snap4City and with the solution provided. https://www.eclipse.org/sumo/ https://sumo.dlr.de/docs/ https://en.wikipedia.org/wiki/Simulation_of_Urban_Mobility
Super	See **SuperServiceMap**
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 with a variable, sensor, or actuator. A Time Serie may have changed GPS location of measure, and in that case is a Moving Object Time Series. In Snap4City, all the variables of IoT Devices, area HTL Sensors/Actuator and maybe Time Series; also MyKPI variable may be Moving Object Rime Series.
Tracker	A Specific Widget for tracking moving devices, that can be located in Snap4City Dashboards.
Traffic Flow Manager	A Snap4City tool for collecting and distributing Traffic Flow data, for a given area, for a specific segment of roads, etc. At each Traffic Flow the Traffic Flow Manager assigns several metadata and a Color Map. They are distributed via GeoServer. Similar to Heatmap Manager but for traffic flows.
Traffic Flow PolyLine	Traffic Flow data is produced as a polyline. They are distributed for showing them on the 3D representation of the city as Crests. See Traffic Flow.
Traffic Flow Reconstruction	Algorithm and Tool for computing the traffic flow in ay point of the city based on the data collected from a limited number of sensors located on roads and scattered in the city.
Transmodel	A format for transportation data https://www.transmodel-cen.eu/

TV Cam Manager	Video Cam manager, to collect information about TV Cam stream, manage them, and redirect them on Dashboards via Kurento.
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 the 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 the Snap4City platform is defined in terms 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 the Snap4City platform is defined in terms 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 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 values 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 multi contextual 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
XAI	Explainable AI, a set of technologies to explain the results of AI solutions, that is to produce AI solutions being ethical and nonbiased.

5- References

5.1 - Published Material on TOP level Journals and Conferences with International Reviewers https://www.snap4city.org/426

5.2 – Snap4City Main documentation web pages

TECHNICAL OVERVIEW: https://www.snap4city.org/download/video/Snap4City-PlatformOverview.pdf
Dev paper: https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf
Client Side Business Logic: https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf

- **DOMAIN: Control and Plan Horizontal Artificial Intelligence Platform Digital Twin for All Domains:** https://www.snap4city.org/1039
- **DOMAIN: Mobility and Transport Operation and Plan Digital Twin:** https://www.snap4city.org/1040
- **DOMAIN: Smart Energy and Smart Buildings Operation and Plan Digital Twins:** https://www.snap4city.org/1041
- **DOMAIN: Environment and Waste Management Digital Twin:** https://www.snap4city.org/1042
- **DOMAIN: City Users' Services, Tourism Management and Safety Digital Twin:** https://www.snap4city.org/1043

	1st part	2nd part	3rd part	4th part	5th part	6th part	7th part	8th
what	Overview	Dashboards	IOT App, IOT Network	Data Analytics	Data Ingestion processes	System and Deploy Install	Smart City API: Web & Mob. App	Design and Develop Smart Solutions
PDF 2022								
Interactive (2022) with video and animations								

Training Course page	https://www.snap4city.org/944 **use this link to get PDF updated versions of the following training sections** **and interactive versions cited in the paper.**
Training Part 1 – overview / for developers	https://www.snap4city.org/download/video/course/p1-rnd/
Training Part 1 - overview	https://www.snap4city.org/download/video/course/p1/
Training Part 2 - Dashboard	https://www.snap4city.org/download/video/course/p2/
Training Part 3 – IoT App and networks	https://www.snap4city.org/download/video/course/p3/
Training Part 4 – Data Analytics	https://www.snap4city.org/download/video/course/p4/
Training Part 5 – Data Ingestion	https://www.snap4city.org/download/video/course/p5/
Training Part 6 – System deploy, summary	https://www.snap4city.org/download/video/course/p6/
Training Part 7 – API and Mobile Apps	https://www.snap4city.org/download/video/course/p7/
Training Part 8 – Design and Develop Smart Solutions	https://www.snap4city.org/download/video/course/p8/

Snap4City Impact Story at FIWARE	https://www.snap4city.org/drupal/sites/default/files/files/FF_ImpactStories_Snap4City.pdf

Km4City Smart City IOT Ontology	https://www.snap4city.org/download/video/DISIT-km4city-City-Ontology-ita-v5-1.pdf see also for the new version: https://www.snap4city.org//21
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
Scenarios	https://www.snap4city.org/4

5.3 - Scenarios see https://www.snap4city.org/4

5.4 - Main "How To" Guidelines https://www.snap4city.org/108

5.5 - Other articles: https://www.snap4city.org/78