



Snap4City Platform



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>



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: 24-09-2023

Version: 7.7



UNIVERSITÀ
DEGLI STUDI
FIRENZE

DINFO
DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE

DISIT
DISTRIBUTED SYSTEMS
AND INTERNET
TECHNOLOGIES LAB

Table of Contents

1 - Executive Summary	4
1.1 Change of Terminology, for 2023 version of the platform	5
1.2 Document Organization	5
2 - Snap4City References	7
2.1 - Installations on premise and volumes of Snap4City	9
2.2 - Examples of Scenarios of Snap4City	14
2.3 - Direct Contracts and Projects Exploiting Snap4City Platforms	16
2.4 - Recognitions and Awards to Snap4City, DISIT Lab	17
3 - The Snap4City Platform.....	18
3.1 - Snap4City Architecture Overview	21
3.1 - Snap4City Development Environment and Life Cycle (overview)	29
3.2 - Snap4City Innovation Methodology and Living Lab	30
3.3- Snap4City Data Sources/Recipients, Bidirectional Channels	33
3.3.1 - Main channels for data ingestion/interoperability.....	34
3.3.2 - IoT Network components.....	36
3.4- Snap4City Data Models, Ontology and Digital Twin.....	38
3.4.1 - Snap4City Ontology and Reasoners, a plus in addition to Data Modelling	39
3.4.2 - Snap4City Data Inspector: navigating on data, users and processes via their relationships	42
3.4.2.a - Snap4City: High Level Types.....	43
3.4.2.b - Snap4City: Smart Solutions	45
3.4.2.c - Snap4City: Entity / Data Inspector, Digital Twin	46
3.4.3 - Snap4City High Level Type vs Storage and distribution	49
3.4.4 - Digital Twin Local and Global of the City, 3D City Representation	50
3.4.5 - FIWARE Smart Data Models and Snap4City IoT Device Models	53
3.5 - Snap4City APIs (API Man, Smart City APIs, MicroApplications, Web and Mobile Apps)	55
3.5.1 - Advanced Snap4City APIs (ASCAPI) and Smart City APIs for developers	55
3.5.1.a – Examples of Snap4City Mobile Apps.....	56
3.5.1.b - Snap4City Internal API	57
3.5.2 - Snap4City MicroApplications.....	58
3.5.2.a - Snap4City Proc.Logic / IoT App with Debug Option.....	58
3.5.2.b - Blockchain support on Snap4City	59
3.5.2.c - Snap4City integrated with Milestone X protect VMS	60
3.5.3 - Snap4City Mobile Applications and User Behaviour Understanding	63
3.5.4 - User Engagement via Mobile App	65
3.5.5 - Origin Destination Matrices, Algorithms, and tools	67
3.6 - Snap4City IoT Apps and MicroServices	70
3.6.1 - Snap4City Library of the MicroServices of Smart Cities.....	72
3.6.2 Examples of solutions based on Proc.Logic / IoT App.....	73
3.7 - Snap4City Data Ingestion/Interoperability.....	77
3.8 - Snap4City Federation of Knowledge Base and Smart Cities.....	79
3.9 - Snap4City Data Analytics Development and Exploitation	81

3.9.1 - Snap4City Analytics, the data analytics at your disposal on Snap4City.....	82
3.9.2 - What-IF Analysis Support	86
3.9.3 - An example of Data Analytics integrated with IoT Apps.....	87
3.10 - Snap4City Contextual Dashboards, Business Intelligence, and Control Rooms	89
3.10.0 - Snap4City Dashboard vs Kibana, Grafana Dashboards.....	91
3.10.1 - Snap4City Dashboard Builder.....	92
3.10.2 - Examples of Snap4City Dashboards.....	94
3.10.3 - List of Main Dashboard Widgets for Dashboard Builder	99
3.10.4 - Dashboard Custom Theme/Style	102
3.10.5 - Structure of Snap4City Applications and cohesions among them.....	103
3.10.6 - Snap4City Smart Application, Business Intelligence Tools	104
3.11 - Snap4City Development Environments and Training.....	109
3.12 - Snap4City Security.....	111
3.13 - Snap4City Access Control and GDPR Compliance	112
3.14 - Snap4City Deployed Architecture for Scalability and Control	112
3.14.1 - Snap4City Optional Tools for a typical configuration.....	116
3.15 - Snap4City: Maintenance Management, BPM and BIM.....	117
3.15.2 - Snap4City Business Maintenance Intelligence tool	120
3.16 - Snap4City Back Office: Management Environment and Quality Control	121
3.16.1 - Snap4City User Management	122
3.16.2 - Snap4City Platform Management, quality control	124
3.16.3 - Snap4City Platform Automation	126
3.17 - Comparison with other market and open sources solutions.....	127
4 - Terms and Acronyms of Snap4City.....	128
5- References	139
5.1 - Published Material on TOP level Journals and Conferences with International Reviewers https://www.snap4city.org/426	140
5.2 - Snap4City Main documentation web pages.....	145
5.3 - Scenarios see https://www.snap4city.org/4	146
5.4 - Main "How To" Guidelines https://www.snap4city.org/108	148
5.5 - Other articles: https://www.snap4city.org/78	149

1 - Executive Summary

DISIT Lab provides a Smart City- Industry 4.0 Platform, Snap4City, which covers all the necessities in an urban setting. 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, all of which have been reported in this document, to provide you evidence that we can solve your problems. In all cases and challenges, we have deployed solutions covering all the necessities now, 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.

Snap4City/Industry platform is secure encrypted, scalable, modular, and flexible. Many institutions and companies took up the challenge of working with the DISIT Lab of the University of Florence on this open-source model. Snap4City can be used to set up Living Lab and smart city solutions, thus satisfying a large range of requirements by city officers, citizens, and tourists, as well as developers, companies, and researchers.

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

DISIT Lab is well known at the international level on Smart Cities / Industry domains for many implementations such as Methodology for Smart Cities and Living Lab setup, Smart City Control Rooms, Smart algorithms and data analytics, and Decision support systems. DISIT Lab has coordinated the ICT part of the strategic plan related to the Florence Metropolitan area, the strategy for Cyprus, etc. DISIT Lab has a dedicated data center for big data solutions on which <https://www.snap4city.org> and Twitter Vigilance are operating, providing services for a large number of cities and regions some of them on multiple instances of the Snap4City platform: Firenze (IT), Helsinki (FI), Antwerp (BE), Valencia (ES), Pont du Gard (FR), WestGreece (GR), Mostar (BA), Dubrovnik (KR), Santiago di Compostela (ES), Lonato del Garda, Pisa, Prato, Pistoia, Lucca, Arezzo, Grosseto, Livorno, Siena, Massa, Modena, Cagliari, Venezia, Roma (IT), etc.; and as far as regions are concerned: Tuscany, Garda Lake, Sardegna, Emilia Romagna (IT), Svealand Region (SE), , Belgium (BE), Finland (FI), Spain, WestGreece (GR), Occitanie (FR), etc.

Today, the main challenges cities are facing are multidomain services and infrastructures regarding mobility and transport, people flow, energy, government, welfare, tourism, culture, environment, weather, and education. These domains can exert a mutual influence, and, in critical conditions, a ripple effect is inevitable, causing further problems. Snap4City improves city services, security, and safety by offering a sustainable solution for Smart Cities and Living Labs. Such a solution fulfils a large range of requirements by city officers, citizens, and tourists, contributing to strengthening the city's capacities to be of great appeal to industries and stakeholders.

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, for 2023 version of the platform

After many years we are 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, from 2023	What
<u>IoT Device Model</u>	Entity Model	A data model
<u>IoT Device</u>	Entity Instance	A data instance ready to get message for time series
IoT Device Variable	Entity Variable	A variable of an Entity Instance or of an Entity Model
<u>IoT Device Message</u> , device message	Entity Message	A data message
<u>IoT Directory</u>	Entity Directory	The tool for managing models, entities, data models, etc.
<u>IoT Applications</u> , <u>IoT App</u>	Processing Logic	= Node-RED + Snap4City Libraries The tool for visual programming, node-red JavaScript, data flow, ingestion logic, data transformation, data loading, interoperability, business logic.

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.

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, 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, 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 <https://www.snap4city.org/4>. **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, tools and applications.

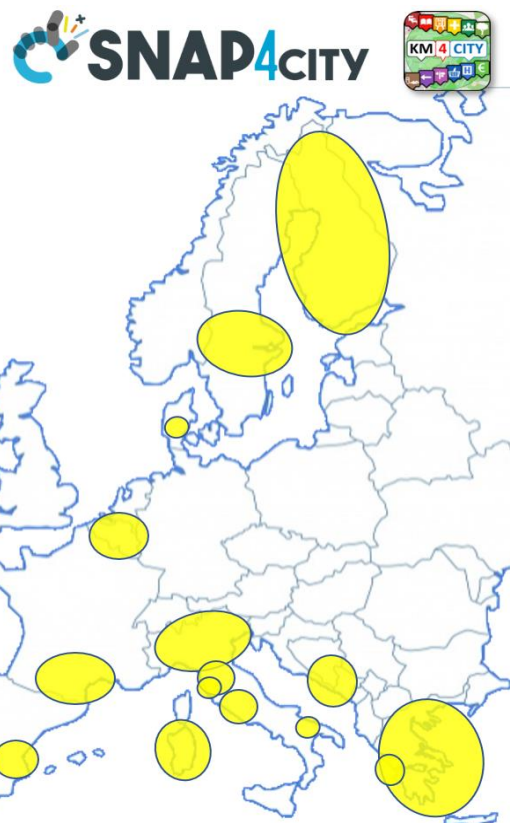
Katrien Lenaert of DigiPolis Antwerp stated: "1st place has been awarded to Florence (Snap4city) ... the competition was really hard ... frequent changes from our parts ..."; "... about Florence (Snap4city) strong points as always, excellent analytics, excellent data aggregation, very strong point was the utilities on which to work and also the fact that you (Snap4city) turn the criticisms in opportunities to improve the tool that made your platform better all the times. It is something that we really appreciated, ... 100% open source, ... and the fact that you (Snap4city) commercialize the solution already, ... very good feature ... thanks again".

Hugo Goncalves, Project Manager, Forum Virium Helsinki, said: "We have seen outstanding solution development in this PCP. Entrance to Phase II was tough and the successful contractors all impressed the Buyers Group by going above and beyond expectations, showing that their solutions not only meet complex technical standards and requirements but also a variety of end-user needs. In this first year of the competition, the innovation levels are very strong, and all of our contractors should be immensely proud of what they have achieved so far."

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

Main Organizations/areas

- [Antwerp area \(Be\)](#)
- [Bologna \(I\)](#)
- [Capelon \(Sweden: Västerås, Eskilstuna, Karlstad\)](#)
- [DISIT demo \(multiple\)](#)
- [Dubrovnik, Croatia](#)
- [Firenze area \(I\)](#)
- [Garda Lake area \(I\)](#)
- [Greece \(Gr\)](#)
- [Helsinki area \(Fin\)](#)
- [Livorno area \(I\)](#)
- [Lonato del Garda \(I\)](#)
- [Modena \(I\)](#)
- [Mostar, Bosnia-Herzegovina](#)
- [Oslo & Padova \(Impetus\)](#)
- [Pisa area \(I\)](#)
- [Pistoia \(I\)](#)
- [Pont du Gard, Occitanie \(Fr\)](#)
- [Prato \(I\)](#)
- [Roma \(I\)](#)
- [Santiago de Compostela \(S\)](#)
- [Sardegna Region \(I\)](#)
- [Siena \(I\)](#)
- [SmartBed \(multiple\)](#)
- [Toscana Region \(I\), SM](#)
- [Valencia \(S\)](#)
- [Venezia area \(I\)](#)
- [WestGreece area \(Gr\)](#)



The map shows several installations and each of them is MultiTenacy. 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 perform **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**, etc., and therefore we perform a continuous training and updates of training material from <https://www.snap4city.org/577>.

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, etc.
- **Testing training as PoC**, (proof of concept), for understanding if the solution is satisfactory. *Most of the PoC which are presently active have not been reported in detail in this document such as those of: Rome, Prato, etc., and other that are covered by some Non-Disclosure Agreement (NDA).*

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

On the web page: <https://www.snap4city.org/661> Snap4City installations reported in the table below are described. 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.**

(data updated in 2021)

Name	URL	Operator	URL of Operator	purpose	location	Shape	Size	KB federated	Living Lab
Snap4City.org	https://www.snap4city.org	DISIT	https://www.disit.org	Smart City, <u>Industry 4.0</u> , any domain, training, development	DISIT University of Florence	Snap4City Industry, all	Huge	Yes	Yes
Snap4Altair, SODA	non accessible from Internet	ALTAIR	https://www.altairchimica.com/	managing chemical industry plant	ALTAIR	Industry	medium	no	no
Snap4Pisa, Mobimart	https://snap4city.aedit.it	AEDIT	http://www.aedit.it/	mobility and transport, POI	PISA	Smart City	medium	Yes	Yes

Name	URL	Operator	URL of Operator	purpose	location	Shape	Size	KB federated	Living Lab
SmartEA SmartGardaLake	https://smartgardalake.snap4.eu/	Snap4	https://www.snap4.eu/	Environment, mobility, transport, POI, Culture, Tourism, etc.	Garda Lake area	Smart City	small	Yes	(yes)
Snap4	https://platform.snap4.eu/	Snap4	https://www.snap4.eu/	Generic MultiOrganization solution for: Environment, mobility, transport, POI, Culture, Tourism, etc.	Italy	Smart City, Industry	Medium	Yes	Yes
Capelon	non accessible from Internet	CAPELON	--	Smart Light, smart city	Sweden	Smart City	medium	No	(yes)
--	Non accessible	UNISYSTEMS	--	Monitoring experiments	Greek	Smart City	medium	No	Yes
--	Non accessible	SYN-NG-AI	--	Smart city	Israel	Smart City	small	No	No
ITALMATIC	Non accessible	ITALMATIC	--	Industry 4.0	Toscana	Industry	Small	No	No
IDTsolutions	not accessible from internet	IDTsolutions				Smart City	Micro X	No	No
IMPETUS	not accessible from internet	IMPETUS				Smart City		No	No
Snap4Asymmetrica	https://www.cityconn.cloud/	Asymmetrica	www.asymmetrica.it	Generic	Italy	Smart City	MICRO X	No	NO

A number of organizations are using the Smart City as a service provided and operated by DISIT Lab directly. The following table reports data on some Organizations' size and their geographic coverage in terms of inhabitants and visitors. It also shows the number of users, dashboards, content/data models (as data sets), and the status (*operative*, *started* - if the volume of data and activity has yet to reach a minimum level of evidence according to our approach, *testing* if it is necessary to train and test the platform before deploying in a larger installation). For some of the above-mentioned installations we do not have information on the data volume; thus, these installations are not included in the following table.

(data updated in 2021)

Name	Organizations	Inhabitants	Visitors	users	dashboards	Content	status
Snap4City.org	Toscana	3.5 million	>48 million	170	44	252 datasets	Operative
	Firenze	1.5 million	>14 million	40	>100	92 datasets	Operative
	Pisa	90.000	>4 million	50	10	80 datasets	Operative
	Helsinki	630.000	>1.3 million	24	>50	27 datasets	Operative
	Antwerp	498.000	1 million	25	>60	25 datasets	Operative
	GardaLake	250.000	> 30 million		3	4 datasets	Operative
	Lonato DG	20.000		10	12	4 datasets	Operative
	DISIT	As Toscana		210	>200	410 datasets	Operative
	Sardegna				4	20 datasets	Started
	WestGreece					12 datasets	Operative
	PontDuGard					10 datasets	Operative
	Dubrovnik				3	6 datasets	Operative
	Mostar					2 datasets	Operative
	Valencia				2	2 datasets	Operative
	SmartBed				4	1 dataset	Testing training
	Capelon	400.000		10	3	5 datasets	Testing training
Snap4Altair, SODA	Altair	12		12	30	400 items	Operative
Snap4Pisa, Mobimart	Pisa	90.000	>4 million	50	10	80 datasets	Operative
SmartGardaLake	SmartGarda	250.000	>30 million	10	5	15 datasets	Operative
Capelon	Capelon			2	4	2 datasets	Testing training

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

Smart City Control Room Florence Metropolitan City

- Multiple Domain Data**
 - Thousands of Open/Private data, POI, IoT, etc.
 - mobility and transport**: accidents, public transport, parking, traffic flow, Traffic Reconstruction, KPI, ...
 - AND**: environment, civil protection, gov KPI, covid-19, social & social media, people flow, tourism, energy, culture, ...
- Multiple dash/tool Levels & Decision Makers**
 - Real Time monitoring, Alerting, quality assess.
 - Predictions, KPI, DSS, what-if analysis
- Historical and Real Time data**
 - Billions of Data
- Services Exploited on:**
 - Multiple Levels, Mobile Apps, API
- Since 2017**



<https://www.snap4city.org/747>

SNAP4CITY (C), August 2022

Herit-Data - Florence Main

<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MxFOA=&v=1>

- Multiple Domains Data**
 - Traffic, environment, People, parking, stock options, Twitter, tc.
- Decision Makers Multiple Locations**
 - NO2 long term predictions
 - Twitter analysis
- Historical and Real Time data**
- Services Exploited on:**
 - Dashboards
 - Social media
 - Sentiment Analysis
- Since 2019, 2020**



SNAP4CITY (C), August 2022

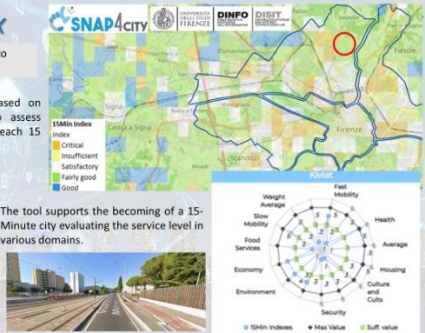
15MinCityIndex

What would support my neighborhood to become a 15-Minute City?

Using the Open Data:
We developed a data analytic tool based on municipal and national open data to assess services adequacy for people living in each 15 minutes areas of the city.

Good public transport services:
bus, new tram line, train stations, cycle paths.

Careggi/Rifredi is a relevant district in Florence because of hosting the main Florence/Tuscany hospitals: Careggi and Meyer, but also university headquarters and many other workplaces.

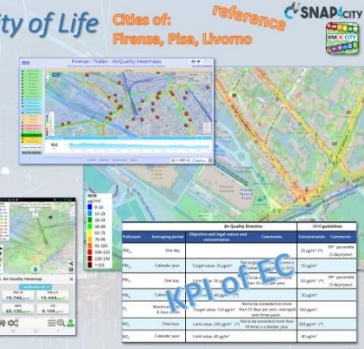


<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MjzQA=&v=1>

SNAP4CITY (C), October 2022

Environment and Quality of Life Air Quality Predictions

- Multiple Domain Data**
 - Traffic Flow data, Pollutant: NOX, CO2, PM10, PM2.5, O3, ...
 - 3D City structure, weather, ...
- Multiple Decision Makers**
 - Pollutant Predictions: NOX, NO2, ...
 - City officers, energy industries
 - Dashboards, What-IF analysis
 - Traffic Flow Reconstruction
- Historical and Real Time data**
 - Billions of Data
- Services Exploited on:**
 - Dashboards, Mobile App
- Since 2020**



SNAP4CITY (C), October 2022

Smart City / Smart Parking + Environment Reverberi, Lonato del Garda

- Multiple Domain Data**
 - Smart Parking, Environment, WiFi
- Multiple Decision Makers**
 - City Officer, operators
- Historical and Real Time data**
 - Dashboards
- Services Exploited on:**
 - Dashboards, API
- Since 2019**



SNAP4CITY (C), January 2022

Mobility and Transport Traffic Flow Analysis

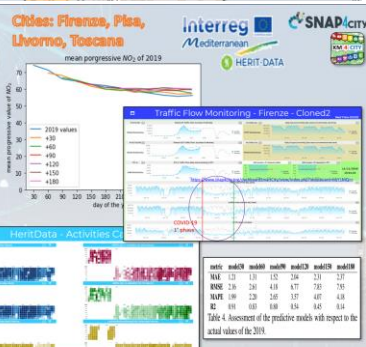
- Multiple Domain Data**
 - Traffic flow sensors, city structure, weather
- Decision Makers Multiple Locations**
 - Real time Monitoring, predictions
 - Traffic Flow Predictions,
 - Traffic Reconstructions, routing
 - Dashboards, What-IF analysis
 - Mobile App, people flows
- Historical and Real Time data**
- Services Exploited on:**
 - Dashboards, Mobile App
- Since 2017, 2019**



SNAP4CITY (C), August 2022

Impact of COVID-19

- Multiple Domains Data**
 - Traffic, environment, People, parking, stock options, Twitter, tc.
- Decision Makers Multiple Locations**
 - NO2 long term predictions
 - Twitter analysis
- Historical and Real Time data**
- Services Exploited on:**
 - Dashboards
 - Social media
 - Sentiment Analysis
- Since 2019, 2020**



SNAP4CITY (C), August 2022

Tuscany Region

- Dashboards & Services:**
 - Mobility**: public transport operators schedule and paths, traffic FI-PI-Li main road, parking status and predictions, traffic sensors, Origin Destination matrix, routing, multimodal routing, etc.
 - Social**: Hospitals and triage, etc.
 - Environment**: sensors, heatmaps, alerting,
 - Pollution Forecast: NOX, NO2
 - Weather Forecast,
 - Culture and Tourism
 - Mobile App and MicroApplications:**
 - Tuscany in a Snap (all stores)
 - Tuscany where what... km4city (all stores)
- Numbers: 1.5 M complex events per day**



SNAP4CITY (C), October 2021

People Control on Pub Services DIGIPOLIS Antwerp

- Multiple Domain Data**
 - PAX Counters: museum, pub services, COVID19
- Multiple Levels & Decision Makers**
- Historical and Real Time data**
 - 20 fixed PaxCounters
 - 2 Mobile PaxCounters
 - Business Intelligence Dashboards
- Services Exploited on:**
 - Dashboards, Mobile Apps, API
 - Fully Controlled Devices by Digipolis
- Since 2019**



SNAP4CITY (C), January 2021

Snap4Altair Decision Support supervision and control, Industry 4.0

- Multiple Domain Data**
 - Distributed Control System: energy, flows, storage, chemical data, settings, ...
 - Cost of energy
 - Orders
 - Production Parameters
 - Maintenance data
- Multiple Levels & Decision Makers**
- Historical and Real Time data**
 - Billions of Data
 - Optimized planning on chemical model
 - Business Intelligence on Maintenance data
- Services Exploited on:**
 - Multiple Levels, Mobile Apps, API
- Since 2020**



SNAP4CITY (C), January 2022

Smart Light Control of CAPELON

- Energy Domain**
 - Smart Light, MQTT, ...
 - IoT Orion Broker FIWARE
- Dashboards**
 - Map coverage on Sweden
 - Monitoring and real time control
 - Energy control, analytics
 - Direct control
- Historical and Real Time data**
- Services Exploited on:**
 - Multiple Levels, API
 - Dashboards
- Since 2020**

SnapshotCity (CL) August 2022

Pont du Gard

- Tourism Domain**
 - KPIs
 - Social Media
 - People Flows
 - Bike Flows
- Dashboards**
 - Monitoring KPI
 - People and bikes flows
 - Twitter Vigilance
- Historical and updated data**
- Services Exploited on:**
 - Dashboard
- Since 2020**

SnapshotCity (CL) August 2022 <https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MzE1Mw==>

Dubrovnik

- Tourism Domain**
 - Counting People
 - TV Cameras and WiFi
 - Social Media
- Dashboards**
 - Monitoring and real time control
 - People flow
 - Twitter Vigilance
- Historical and Real Time data**
- Services Exploited on:**
 - Dashboard
- Since 2020**

SnapshotCity (CL) August 2022 <https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MzE0Ng==>

Valencia, FSMLR

- Tourism Domain**
 - Counting People
 - Environmental data
 - Social Media
- Dashboards**
 - Monitoring and real time control
 - People flow
 - Twitter Vigilance
- Historical and Real Time data**
- Services Exploited on:**
 - Dashboard
- Since 2020**

SnapshotCity (CL) August 2022 <https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MzE1MA==>

Mostar, Bosnia Herzegovina

- Tourism Domain**
 - KPI: business, house prices, investments, stay duration, etc.
 - Profiled POI, Point of Interests
 - People flows: arrivals, overights
- Dashboards**
 - Monitoring KPI
 - POI, flows
- Historical and updated data**
- Services Exploited on:**
 - Dashboard
- Since 2020**

SnapshotCity (CL) August 2022 <https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MzE0OQ==>

West Greece

- Tourism Domain**
 - KPIs: ODM, Flows, ...
 - Social Media
 - People Flows
- Dashboards**
 - Monitoring KPI
 - People flows
 - Twitter Vigilance
- Historical and updated data**
- Services Exploited on:**
 - Dashboard
- Since 2020**

SnapshotCity (CL) August 2022 <https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MzE1NA==>

What-if Analysis

- Definition of scenarios impact on**
 - Traffic, Pollutant, parking, public transport, private flows, etc.
 - KPI analysis

SnapshotCity (CL) August 2022

Progetto Terso - Lavagno

- Traffic Data**
- Environmental Data**
- People counting (pedestrian)**

SnapshotCity (CL) August 2022

Sinottico Impianto

Sinottico Impianto Presse - Autoclave

SnapshotCity (CL) August 2022 <https://dashboard/dashboardSmartCity/view/index.php?iddashboard=MTk=>

Energy monitoring and business intelligence

SnapshotCity (CL) August 2022

You can access to a wider list of scenarios from:

- Scenarios: <https://www.snap4city.org/4>
- training course – Part 1: <https://www.snap4city.org/download/video/course/gen/>
- training in PDF from page <https://www.snap4city.org/577>

The largest installations are operated by DISIT Lab as Snap4city.org. Other installations are also growing as well, such as GardaLake (operated by Snap4 s.r.l. with SmartTea) and Pisa (operated by AEDIT s.r.l., with PISAMO e and Pisa municipality?).

Snap4city.org, as described in the above table, is at present the largest installation that overcome the needs of most of the cities and industry plants since it (data updated in 2023)

- ingests about 2 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 7990 users in different organizations;
- manages more than 20 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 110 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.

Some Details of Data collected / produced in DISIT Org ONLY! (data updated in 2021)

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

dataset	#devices	#variable	#samples per day	Kind
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>

- Europe
 - West Greece: <https://www.snap4city.org/743>
 - Valencia: <https://www.snap4city.org/742>
 - Mostar: <https://www.snap4city.org/744>
 - Dubrovnik: <https://www.snap4city.org/741>
 - Antwerp: <https://www.snap4city.org/526>
 - Helsinki: <https://www.snap4city.org/528>
 - Pont du Gard : <https://www.snap4city.org/740>
 - Scenario: Smart Light Control (CAPELON) <https://www.snap4city.org/713>
<https://www.snap4city.org/816>
- Italy:
 - Arter CER: <https://www.snap4city.org/826>
 - Florence/Firenze: <https://www.snap4city.org/747>
 - Pisa: <https://www.snap4city.org/746>
 - Smart Garda Lake: <https://www.snap4city.org/739>
 - Livorno: <https://www.snap4city.org/751>
 - Siena: <https://www.snap4city.org/759>
 - Prato: <https://www.snap4city.org/758>
 - Pistoia: <https://www.snap4city.org/761>
 - Toscana region: <https://www.snap4city.org/760>
 - Roma: <https://www.snap4city.org/753>
 - Modena: <https://www.snap4city.org/754>
 - Bologna: <https://www.snap4city.org/750>
 - Lonato del Garda: <https://www.snap4city.org/712>

All scenarios are fully described starting from the page: <https://www.snap4city.org/4>

- [Energy Community](#), CER, SELFUSER
- [Smart Light Control](#), CAPELON, [PDF](#)
- [Supporting Decision Makers in Real Time about Quality Lab Analyses on the production process](#), [PDF](#)
- [Herit-Data and Snap4City: to better manage tourism flows](#), PDF
- [Digital Twin Local and Global](#), PDF <https://www.snap4city.org/download/video/tn/3D-DigitalTwin-Snap4City-22.pdf>

- Social Media Analysis: Twitter Vigilance_PDF
<https://www.snap4city.org/download/video/tn/Twitter-Vigilance-22.pdf>
- GDPR Compliant People Detection and Counting using Thermal Cameras_PDF
<https://www.snap4city.org/download/video/tn/ThermalCamera-PeopleCounting-22.pdf>
- [Artificial Intelligence Predicts Landslides in Florence Area](#)
- [Available Parking Slots Prediction](#)
- [Available Bikes and Free Slots Prediction On Bike Sharing Stations](#)
- [Inventory of transferable digital applications and solutions for the tourism ecosystem](#)
- [Scenario: AMPERE Operator Manual](#)
- [Scenario: IMPETUS Intelligent Management of Processes, Ethics and Technology for Urban Safety \(Oslo and Padova\)](#)
- [Scenario: Monitoring Cross Road in Venaria Reale - \(AXIS Camera\)](#)
- [Welcome: how to start using Snap4City for beginners](#)
- [Global Digital Twin for Florence by Snap4City](#)
- [15MinCityIndex: understanding city areas by means of 13 different aspects](#)
- [Snap4City Impact Story on FIWARE: https://fiware-foundation.medium.com/snap4city-fiware-powered-smart-app-builder-for-sentient-cities-acfe24df49d5](#)
PDF version:  [FF ImpactStories Snap4City.pdf](#)
- [Scenario: Smart Light Control \(CAPELON\)](#)
- [Scenario: Smart Parking in Lonato del Garda \(Reverberi\)](#)
- [Scenario: SnapBot: Real Time Smart City services via Telegram](#)
- [Scenario: Copernicus Satellite Data](#)
- [MicroServices Suite for Smart City Applications](#)
- [Scenario: MODBUS for Snap4Industry Snap4City Applications](#)
- [Scenario: MOBIMART Interreg: MOBilità Intelligente MARE Terra](#)
- [Scenario: City of Roma case, mobility and environmental data](#)
- [Scenario: HERIT-DATA video and aims](#)
- [Scenario: Control Room vs Video Wall](#)
- [Scenario: Snap4Home the case of: Alexa, Philips, Sonoff, TP-link, etc. \(Italiano\)](#)
- [Scenario: how to manage maintenance and accidents workflows](#)
- [Scenario: Snap4Home, how to exploit Snap4City solution on home automation](#)
- [Scenario: Energy Monitoring](#)
- [Scenario: Multipurpose User Engagement Tools](#)
- [Scenario: 5G Enabled Water Cleaning Control \(smart city, industry 4.0\)](#)
- [Scenario: High Level Control of Industrial Plant \(industry 4.0\)](#)
- [Scenario: Vehicle Monitoring via OBD2](#)
- [Data Analytic: Origin Destination Matrices, Algorithms and tools](#)
- [Data Analytic: Traffic Flow Reconstruction](#)
- [Data Analytic: in general, and the cases of Antwerp and Helsinki](#)
- [Data Analytic: Predicting Air Quality](#)
- [Data Analytic: Analyzing Public Transportation Offer wrt Mobility Demand, DORAM Tool](#)
- [Scenario: Events and Museums Monitoring in Antwerp](#)
- [Scenario: High Resolution Prediction of Environmental Data](#)
- [Scenario: Mobility and Transport Analyses in multiple cities](#)
- [Scenario: People Flow Analysis via Wi-Fi](#)
- [Scenario: Antwerp Pilot on Environmental Data](#)
- [Scenario: Helsinki Pilot on Environmental Data](#)
- [Scenario: Firenze Smart City Control Room](#)
- [Scenario: Mobile & Web App: Toscana Where What ... Km4City, Toscana in a Snap](#)
- [Scenario: Helsinki Pilot on User Behaviour](#)
- [Scenario: Antwerp Pilot on User Behaviour](#)
- [Smart City Control Room Dashboards: Big Data Infrastructure, from data to decision support](#)
- [Dashboards Touch Screen Support for Situation Rooms and What-iF analysis Operators](#)
- [Data Analytics on Antwerp and Helsinki](#)

- [Decision Support Engines and Models: SmartDS, ResilienceDS](#)
- [IPCAM on Dashboard Widget, Video Cameras](#)
- [Solution: using PAX Counters, monitoring museum and events](#)
- [Solution: selecting the less pollute routing](#)
- [Data Flow Management and Visual Analytic for Big Data Smart City/IOT](#)
- [Snap4City on Social Media](#)
- [AIRQino: Compact low-cost air quality monitoring station, by CNR IBIE](#)
- [Security, Privacy and GDPR Aspects of Snap4City](#)

2.3 - Direct Contracts and Projects Exploiting Snap4City Platforms

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

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

- **Merao for the smart light**
- **Cuneo for the asset management systems**
- **ARTER** → monitoring energy facilities <https://www.snap4city.org/826>
- **SMART AMBULANCE** Fratelli Mariani, data hub on the ambulance: <https://www.snap4city.org/708>
- **ALMAFLUIDA**: ALMA FLUIDA activity (ITALMATIC) <https://www.snap4city.org/709>
- **ENEL-X**: Continuing working on Open Data Challenge, 15MinCity Index indicators
- **Capelon** (Sweden) → smart light solutions, the center of Sweden area: <https://www.capelon.se/>
- **Florence Metro City** → ICT aspects of the strategic plant of Firenze Metro City
- **Green Field Peas** (soda) ALTAIR Chemical → Industry 4.0, Chemical plant: <https://www.altairchimica.com/blog/progetti-di-ricerca-e-sviluppo-green-field-peas-e-soda/>
- **Green Impact ALTAIR** → Industry 4.0, Chemical Plant <https://www.altairchimica.com/blog/progetto-di-ricerca-e-sviluppo-green-impact-capacity/>
- **ISPRA JRC EC** research center, 2019 and 2020 → site management and services, for JRC ISPRA area: <https://ec.europa.eu/jrc/en/about/jrc-site/ispra>
- **Lonato del Garda city**, Reverberi ENETEC → smart parking, environment municipality: <https://www.reverberi.it/it/application/lonato-del-garda-bs>
- **MobiMart**: City of PISA, PISAMO → data aggregation, Living Lab: <http://interreg-maritime.eu/web/mobimart>
- **Select4Cities** PCP → smart city platform PCP, Helsinki and Antwerp: <https://www.select4cities.eu/>
- **SmartBed** (laid), Materassificio Montalese → smart health: <https://blog.perdormire.com/smart-bed/>
- **Cyprus smart city strategy.**
- **Italferr** on Snap4 → data management and business intelligence (via Snap4)
- **ENEL-X** → monitoring energy services (via Snap4)
- **Pretto** → monitoring eVehicles and maintenance (via Snap4)

Projects and pilots in which Snap4City is used as a reference platform:

- PNRR Italian **Centro Nazionale Mobilità e Trasporti**, national center on mobility and transport
- **EDIH Toscana X.0**, reference platform for training, UNIFI
- **AMPERE Project - Furniture, art and fashion in an innovative electrochemical process with remote control and circular eco-friendly**: <https://www.snap4city.org/707>
- **Enterprise**: internal living lab for chemical plant at the support of the internal research and analysis labs
- **5G MIUR** Wind3G → Industry 4.0 vs SmartCity, for Prato: <http://www.confservizitoscana.it/news/tecnologia-5g-a-prato-due-progetti-di-estra-nella-sperimentazione-promossa-da-wind-tre-e-open-fiber>
- **GHOST MIUR** → Strategies, smart city, for Cagliari: <https://sites.unica.it/ghost/home/>
- **Herit Data Interreg** → tourism, culture and management: <https://herit-data.interreg-med.eu/>
- **MOSAIC RT** → mobility and transport: <https://www.disit.org/drupal/?q=node/7174>
- **PC4City Ente Cassa** → Civil protection for the City, Tuscany, prediction of landslides

- **REPLICATE H2020** → ICT, smart City Control room for Florence, Energy, IoT: <http://replicate-project.eu/>
- **RESOLUTE H2020** → Resilience, ICT, Big Data, for Florence, Athen: <http://www.resolute-eu.org>
- **Sii-Mobility SCN MIUR** → mobility and transport, sustainability, the whole Tuscany: <http://www.sii-mobility.org>
- **Smart Garda Lake** → Castelnuovo del Garda municipality: <https://smartgardalake.it/progetto/>
- **TRAFAIR CEF** → Environment & transport, Florence, Pisa, Livorno, Modena and Santiago de Compostela: <http://trafair.eu/>
- **WEEE Life EC** → Smart waste, environment: https://www.lifeweee.eu/lifeWeee_en/

2.4 - Recognitions and Awards to Snap4City, DISIT Lab

The Snap4City Platform:

- 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 Services**:
 - <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

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, 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 and dashboards 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, water, environment, industry, 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, origin destination matrices, satellite data, 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 solution. Entity/Data Ingested with any data/entity model are aggregated on the basis of Km4City Ontology into the so-called Knowledge Base (which is the Expert System of the city/industry) and into the Big Data storage, NoSQL. This approach eliminated the problems of data silos and pillars. Any legacy solution, as well as new applications and data/entity is integrated in a unified model, establishing semantic relationships of any kind and respecting the original semantic, thus avoiding the flatness of data lake that results in poor performance in data/information access when needed for rendering.



Figure: Global Architecture of Snap4City solutions and framework

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 Inspector / Data Inspector, Big Data

Analytics, 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/di/> Data ingestion can be also performed with high performance processes written in Python (provided by Snap4City as configurable script), for example for fast loading historical data from other data base or formats.

Data/Entities can be consumed by data-analytics, Processing Logic / IoT App, mobile App, and Dashboards and Synoptics, 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 can be developed in Python and RStudio, and the platform provide tools and guide to transform them in containers to be used into Processing Logic flow (IoT App) in Node-RED. <https://www.snap4city.org/download/video/course/da/>

Data Rendering via Synoptics, Dashboards, tables, any kind of graphics widgets can be easily used just selecting the data and choosing the preferred graphical representation of data/entities among the several possible: time trends, maps, bars, multiserries, chords, OriginDestination (OD) maps, tables, heatmaps, trajectories, scenarios, paths, shapes, buttons, dimer, sliders, hierarchies, spidernet, donut, comparing trends, staked diagrams, barseries, custom animated elements, etc. <https://www.snap4city.org/download/video/course/das/>

Developers and qualified operators on Snap4City platform 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, 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.

Mobile App developers need a local development Environment depending on the target mobile devices. Final users can access to Snap4City services via Web and mobile devices, dashboards, synoptics, interfaces for digital signages, tables, panels, etc.

A Snap4City Living Lab allows stakeholders to collaborate for the production of smart solutions and to the innovation of the whole smart ecosystem. Snap4City provides a methodology for stimulating the innovation identifying the most relevant and effective changes and solutions according to a quadruple helix approach. A Living Lab with the web based Snap4City Development Environment provides a comprehensive set of tools for developers and stakeholders to implement data ingestion and processing flows, Data Analytics algorithms, Dashboards, Processing Logic / IoT App, Synoptics, Custom Widgets, and Web and Mobile Apps [BigDataService2018].

The Snap4City Platform

- **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> FF_ImpactStories_Snap4City.pdf
- Snap4City **is a Solution Powered by FIWARE**: <https://marketplace.fiware.org/pages/solutions/b8905e91973b420189cce972>
- **Snap4City if an official FIWARE Platform**: <https://marketplace.fiware.org/pages/solutions/d68534ec827500f1bde8720f>
- has been the **winner of Select4Cities PCP** <https://www.snap4city.org/558> (by Antwerp, Copenhagen, and Helsinki), in September 2019.
- Winner of the Open Data **Challenge 2020 of ENEL-X**: <https://openinnovability.enel.com/projects/New-smart-city-solutions-enabled-by-open-data>

- **official platform of EOSC** (European Open Science Cloud) marketplace of the European Commission. <https://marketplace.docker-fid.grid.cyf-kr.edu.pl/services/snap4city>
- **GDPR compliant** and enforces privacy and security for data, **Dashboards, IoT Devices, IoT App, personal data, data analytics** and **processes**, etc., which can be private to the user; and the user may delegate the access to them or pass the full control to other users. <https://www.snap4city.org/670>
- provides tools for the administrators to monitor and control the platform status via: auditing, assessment, management, GDPR, network flow monitoring, revoking, request to delete, etc. <https://www.snap4city.org/download/video/course/sys/>
- passed **PEN Tests** from Setek and Vulnerability test from Thales Mayor expert companies. <https://www.snap4city.org/669>
- can be installed on-premise as well as on any public or private cloud. <https://www.snap4city.org/471>
- can create end-2-end solutions, secure from device to dashboards.
- official library of Node-RED, and official SigFOX partner
- Km4City API of Snap4City are accessible API via E015 of Regione Lombardia
- Etc.

Snap4City.org is operated by DISIT Lab and

- Last hackathon organized by Snap4City with IEEE Intelligent Transportation Systems <https://www.snap4city.org/757>
- organized successful international Hackathons for coworking: Florence, Tuscany, Helsinki, Antwerp; <https://www.snap4city.org/hackathon>
- is supported by several companies and partners: <https://www.snap4city.org/577>
- 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 and Dashboards with almost no coding activities. Orange parts of Figure 3 are those usually developed, while all the rest is part of the provided microservices and infrastructure. 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.
- Authentication and Authorization APIs.

Edge and Fog processing 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.

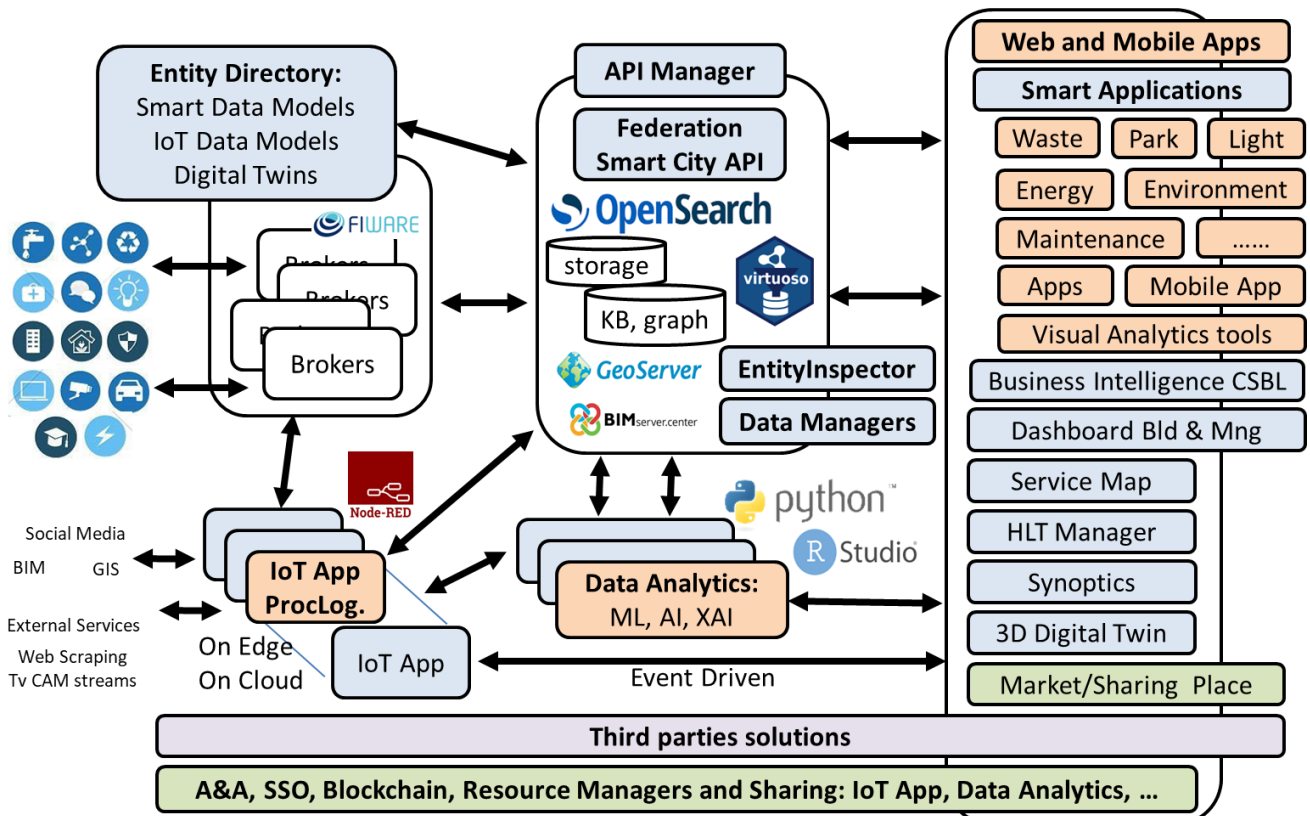


Figure: Architectural Overview of Snap4City framework

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

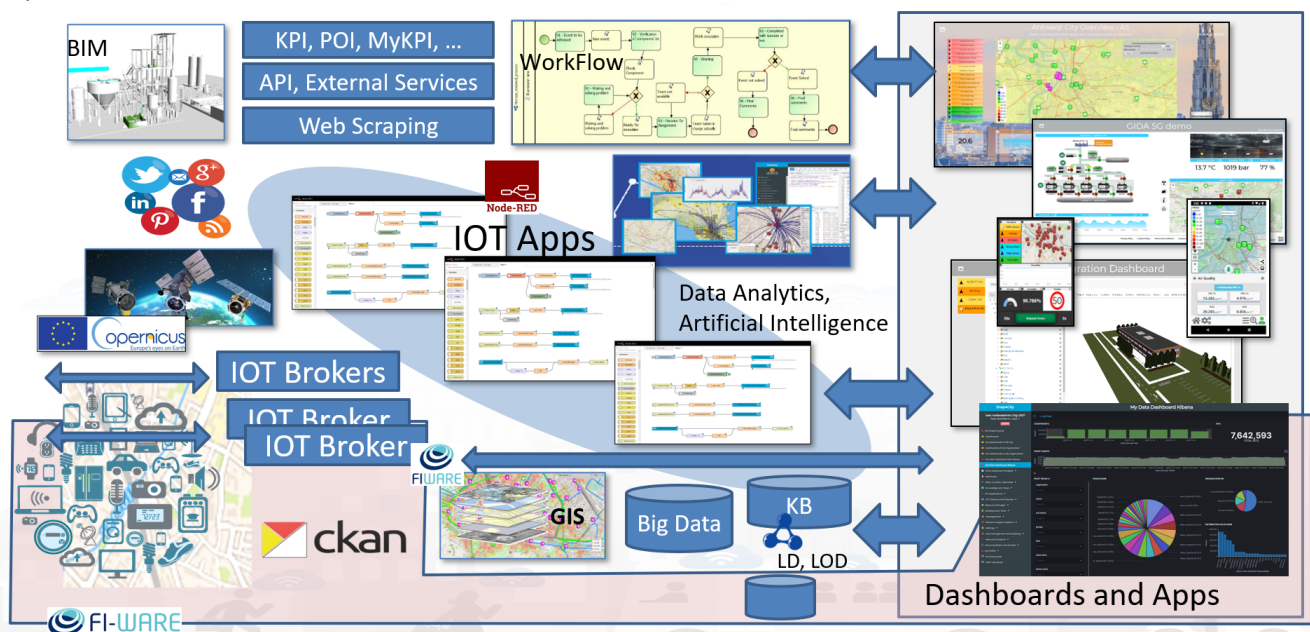
- **data ingestion/transformation and interoperability.** Data may enter/exit in push via brokers of different kinds, and/or via dedicated processes in pull/push (enabling the data/event driven approaches). Data/Entities may be listed in some Entity Directory (IoT Models, Entity Model, Entity Instances, 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. 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.
 - **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.
- **Processing Logics** could be in some ETL/ELT script. In Snap4City, **Processing Logic / IoT App** are Node-RED plus Snap4City microservices or Python processes (to push data in NiFi which is the gate to the storage, OpenSearch). Visual programming languages such as Node-RED, NiFi are typically preferred for rapid prototyping. 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 protocol and format. Please remind that brokers can be also used to act on devices.
 - **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.
- **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 instance can be directly deployed, and specific data loaded. Entities/Data entering in the platform can be automatically (i) stored (in **Figure 3**: Open Search), (ii) semantically indexed, (iii) directly sent to the user interface, or to data analytics for stream processing. The same data flows work as data 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 stored (in some cases, the storage 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 Hadoop based platforms), 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, library or dedicated hardware you need, as NVIDIA, 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], 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- and/or on server-sides. In the latter case, the business logic can be implemented via Processing Logic as Visual programming Node-RED. 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 the graphic widgets with JavaScript on client side or demanding the implementation of the business logic on server side or 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.
- **All the data streams (internal and external) are protected, authenticated, and authorized according to [Security2020], Pen Test passed, GDPR compliant, etc.**

No proprietary components are installed on the official distributions of Snap4City platform, while any proprietary solution can be integrated. So that if you are a solution builder you can add your proprietary solution to do business by leveraging the Snap4City solution and tools.

A **Conceptual version of Snap4City architecture** is reported in the next figure, the main elements of which are reported and described below.



General Concept from <https://www.snap4city.org/download/video/course/gen/>

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/iot/>

- **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,

- **Data Table loader** and POI Loader 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,
- **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,
- Satellite data service using dedicated Python processes also integrated with **Processing Logic** / IoT Apps,
- data gathering tools such as **DataGate/CKAN** for open data (<https://ckan.org>) are also integrated with IoT Apps by using dedicated MicroServices,
- GIS data ingestion/publication via WFS/WMS, also integrated with **Processing Logic** / IoT Apps; for example, using GeoServer, ArcGIS, etc.,
- **Web Scraping via Portia**, also integrated with **Processing Logic** / IoT Apps.

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 internally designed in terms of tables and relationships among tables (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 Models 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 V1, V2, and LD 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/iot/> and in **Section 3.6**.

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> and **Section 3.7**.

For **Data Formats**, the Snap4City supports a large range of **High-Level Types**, *HLT: Entity Instances, IoT Devices, FIWARE Smart Data Models, Entity Models, Sensors/Actuators, POI, Trajectories, Origin Destination Matrices, Flows, Traffic Flow, People Flows, Heatmaps, Satellite data, 3D BIM building, 3D shapes of city, social media, routing, public transport offers, geographical shapes and grids, KPI, TV Camera streams, predictions, animations, synoptics, city scenarios, events of several kinds, user profiles, custom formats, etc.*

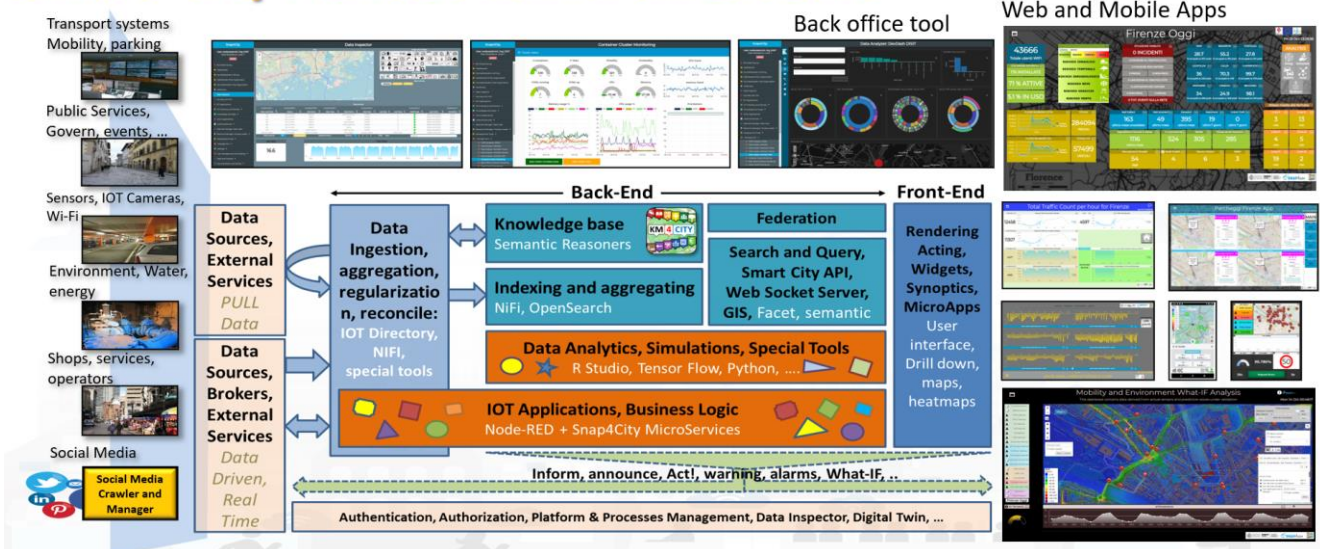
To remark, the Entity Models (IoT Device Models) can be used as template to create Entity Instances (IoT Devices). The Entity Instances (IoT Devices) are those that can receive messages to update their data over time, and thus to create time series. The Entity Instances and thus most of the instances 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 on a browser and the SURI refers to a public entity, its definition appears since it is compliant with the Linked Data standard.

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

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

Smart City Functional Architecture



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 also stored into some SQL database.
- **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, for example for federating different KB or 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, if the 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.
- **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.
- **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. see **Section 3.4.4**.
- Etc.

Processing Logic / IoT Apps can be used for a range of activities in data ingestion, storage, interoperability, transformation, data driven, etc. The activities of **Processing Logic** are **better and 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, 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/das/>. and in **Section 3.10**.

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/di/> 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, algorithms also exploiting Tensor Flow, CUDA, Keras, Pandas, etc., (ii) produce results as predictions, early warning, hints, new

data, heatmaps, 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 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 success stories are provided in: <https://www.snap4city.org/download/video/course/da/> and in **Section 3.9**.

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/app/> and in **Section 3.6**. **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. They 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 some API manager. For the purpose, Snap4City suggests APIMan open-source tool.

Advanced Smart City APIs, (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 Advanced Smart City API can be also exposed via some API manager. For the purpose, Snap4City suggests APIMan open-source tool.

Dashboards can be created by using the Dashboard Builder and/or Open Search Dashboard (former Kibana, sister of Grafana) for different kinds of users such as: decision makers, city operators, ICT operators, private users, etc., suitable for Smart City Control Rooms with video wall, for city Operators on Desktop of multiple monitors, mobile operators, and Situation Rooms with touch panels. Dashboards can exploit all 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/das/>

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, 3D shapes, 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 8 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/das/> 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/sys/>

Web and Mobile Applications can be created by developers using **Advanced Smart City APIs - ASCAPI** and may be controlled by Snap4City tools. For example, to send on Mobile Apps engagements, polls, stimulus, and then 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/app/>

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
- 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, 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.
- **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, 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, PM_{2.5} 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.

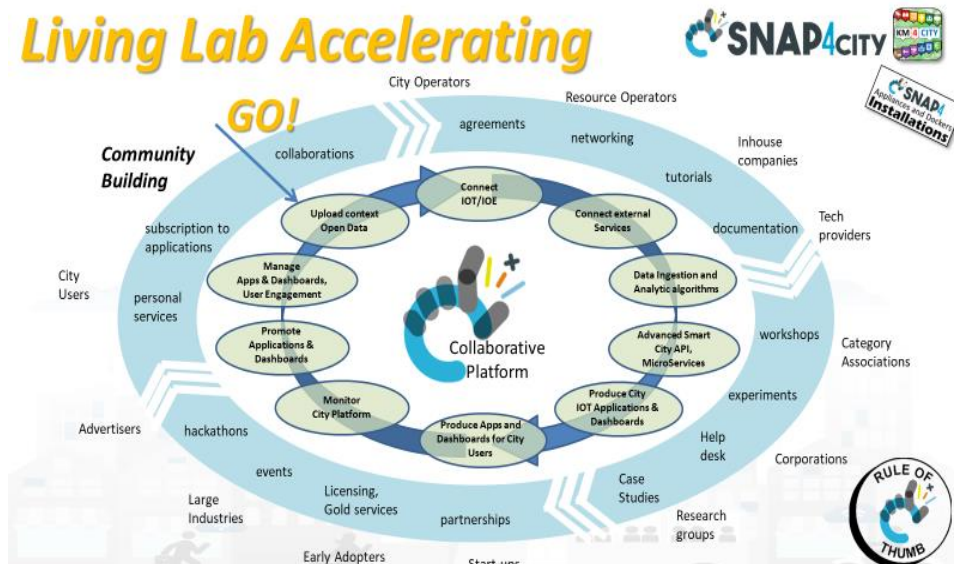
Smart Cities need to set up a flexible **Living Lab** to cope with the city's evolution in terms of services, city users' needs and capabilities. To this end, the **Snap4City** solution provides a set of tools and a flexible method and solution to quickly create a large range of smart city applications exploiting heterogeneous data and stakeholder services also

From Strategies to (re-)Actions →

- Analyze
- Alerting, Early Warning
- Support Decision makers
- Plans
- Prescriptions
- Inform
- Suggest
- Engage
- Research



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 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/sys/>

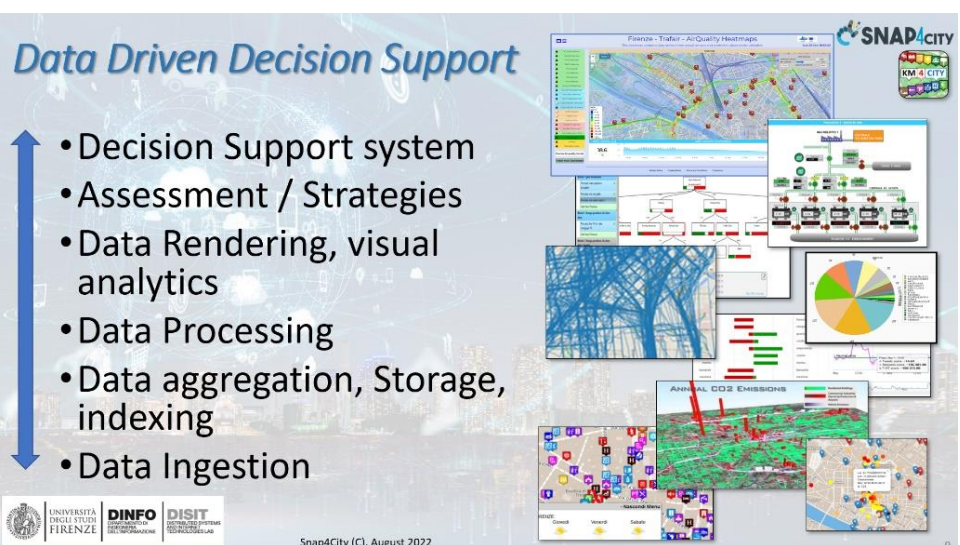
The scenarios identified during the workshops may be focussed on specific domains (mobility, energy, government services, water, environment, etc.) and to improve services to specific user categories. To this end, a process of **Data Discovery** is also carried out in dialogue with stakeholders and their 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, 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.



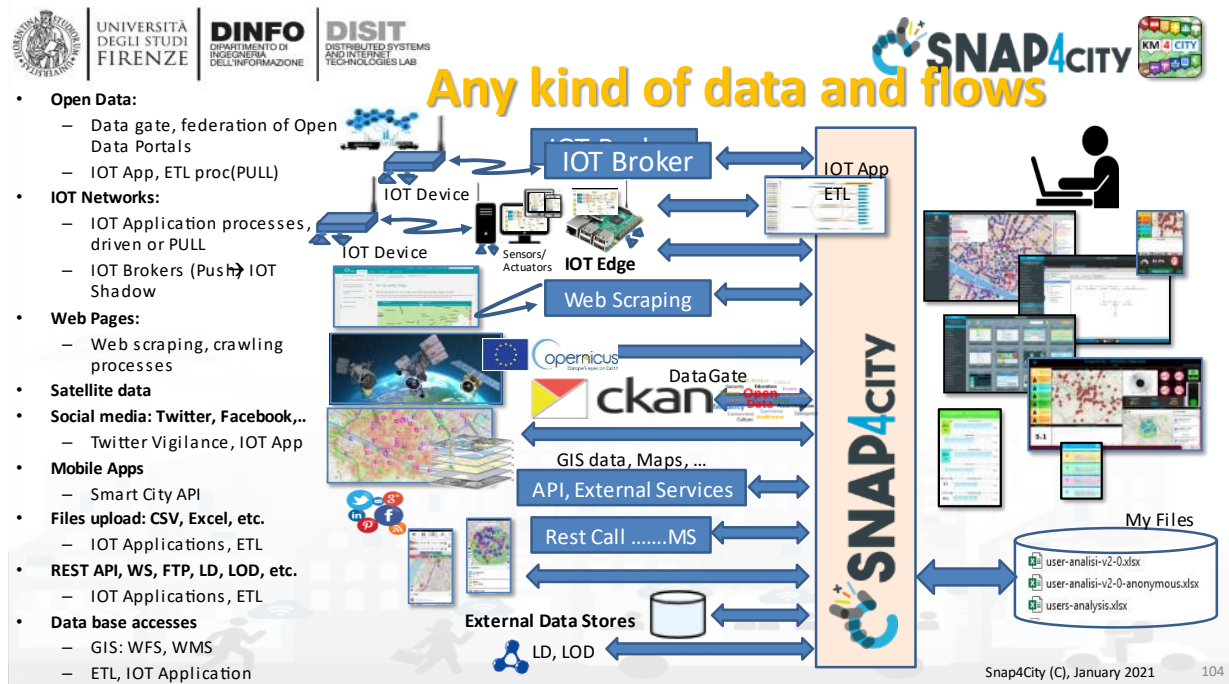
3.3- Snap4City Data Sources/Recipients, Bidirectional Channels

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

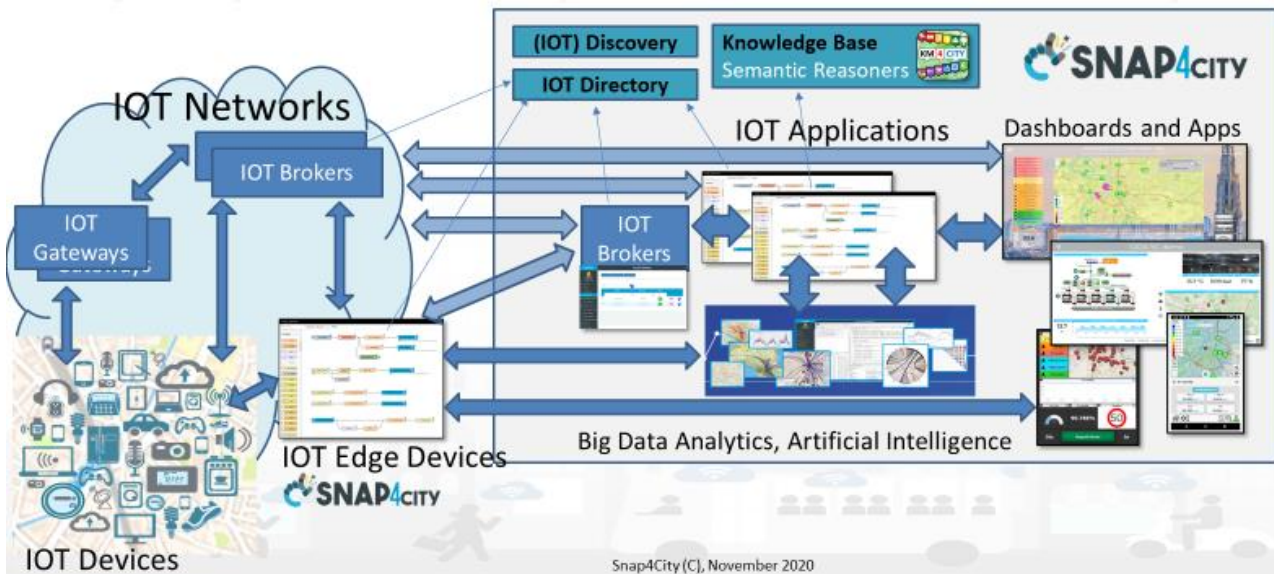
The whole process is described in:

<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.



Snap4City (C), November 2020

29

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/iot/>
- <https://www.snap4city.org/download/video/course/di/>

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**;
 - **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.
- **APIs on external services as:**
 - **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.
 - **GIS data ingestion/publication via WFS/WMS**, also integrated with IoT Apps; for example, using GeoServer, ArcGIS, etc.

- **IoT Gateways, web servers, FTP server**, social media services, linked open data, linked data, ESBs
- **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:**
 - 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,
- **Specific Tools for:**
 - **Web Scraping** via Portia 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.
 - **Data Table Loader** for shortening the activities of registering devices, loading time series, etc., by loading the data from excel files.
 - **POI Loader** for shortening all the activities of ingesting new data for points of interest providing them as excel files.
 -
- **Files:** of any format they can be loaded via: (i) specific IoT App to set up specific data ingestion processes such as via Web page, FTP, SSH, DropBox, Google Drive, etc., (ii) **DataGate/CKAN**. Files loaded can be any format and size.

Direct ingestion and publication via Brokers connect IoT Networks of Entities Devices. A major role is played 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.
- Entity Instances / IoT Devices and Broker 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 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.
- **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?iddashboard=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**.
- **Data Analytic** processes may create new data, which can be saved into the platform and may contribute to the vision of the city via **Dashboards** and to produce innovative and smart services. For example, they can produce: predictions, early warning, traffic flow reconstruction, alarms detecting anomalies, KPIs, Typical Time Trends, typical trajectories, Routing and paths, travel plans, query results, simulations, analysis, calibrated heatmaps, smart parking suggestions, car sharing analysis, etc. See **Section 3.9**.
- **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**.

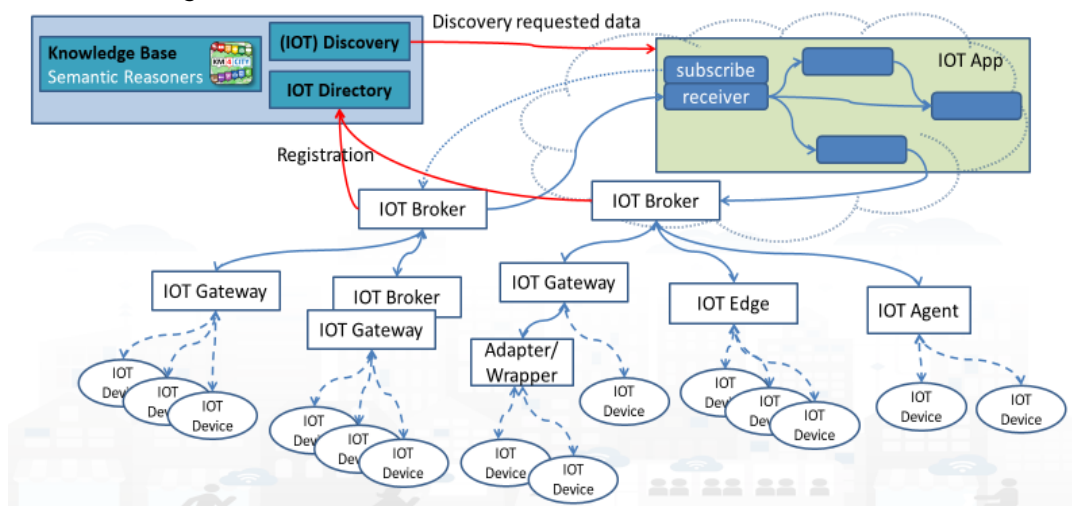
In addition, it is possible to **generate Reports:** as consumptive views on the platform status on specific programmable aspects for users and administrators, for example at the level of **IoT Devices**, and **Dashboards**.

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/iot/>

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 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, 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.

- **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>

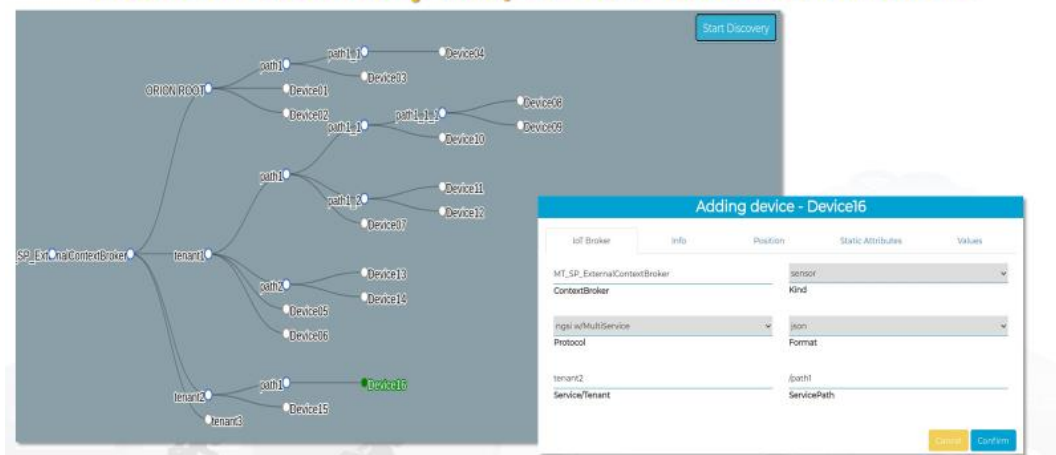
IOT Directory and Devices

- My IOT Sensors and Actuators
- IOT Sensors and Actuators
- IOT Devices
- IOT Devices Management
- IOT Brokers
- IOT Device Models
- IOT Devices Bulk Registration
- IOT Broker Periodic Update settings
- IOT Orion Broker Mapping Rules
- Doc: IOT Directory and Devices
- Create an IOT Device Instance
- Create an IOT Device Model
- Add an IOT Device into Snap4City

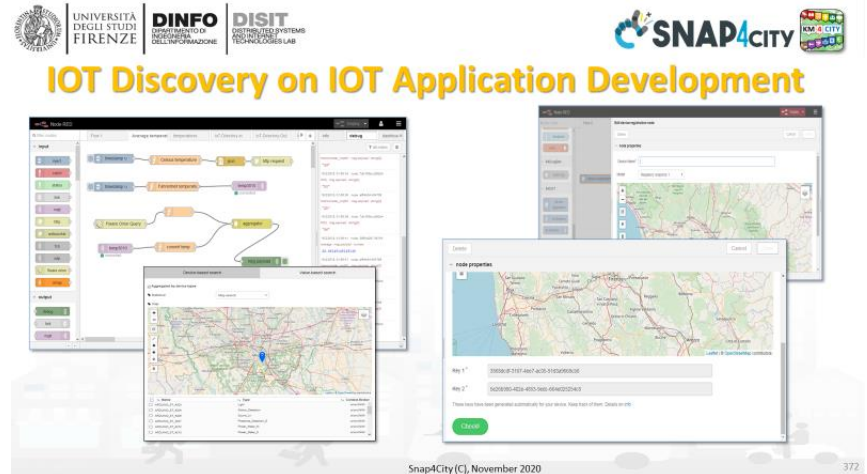
For non admin tools see Training parts 3 and 5: <https://www.snap4city.org/577>

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>

Device Discovery MT/SP on External Brokers

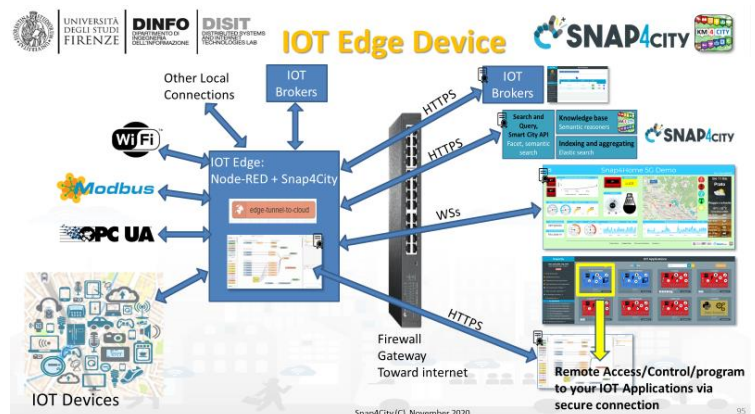


- **Entity / IoT Discovery:** It is a function of Snap4City Entity Directory and Knowledge base which allows discovering the Entities / Devices by several filters: geospatial, by type, by value name, by unit, by nature and sub nature, etc- <https://www.snap4city.org/109>
The same feature is provided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

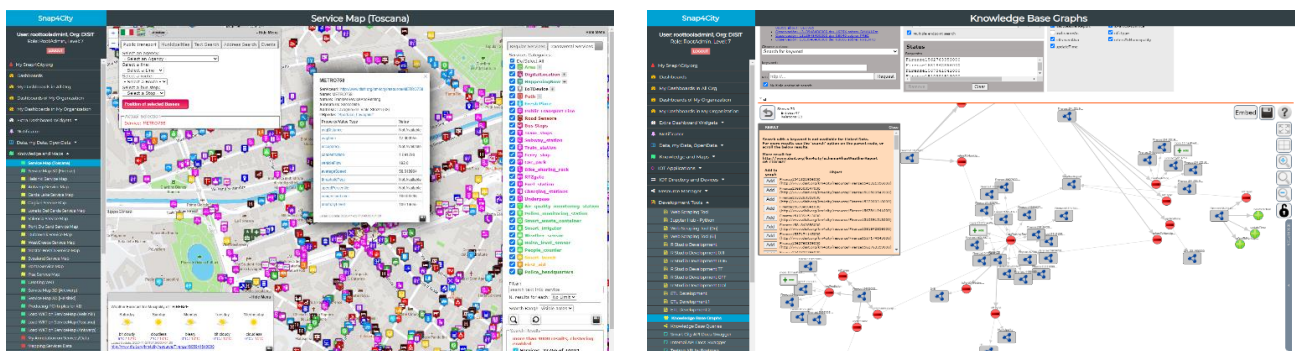
Km4City Ontology has been defined to address in a unified manner any city and industry 4.0-related domains, concepts, structures and data models [<https://www.snap4city.org/19>]. <https://www.snap4city.org/download/video/DISIT-km4city-City-Ontology-eng-v5-1.pdf> Each data element of every single model 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

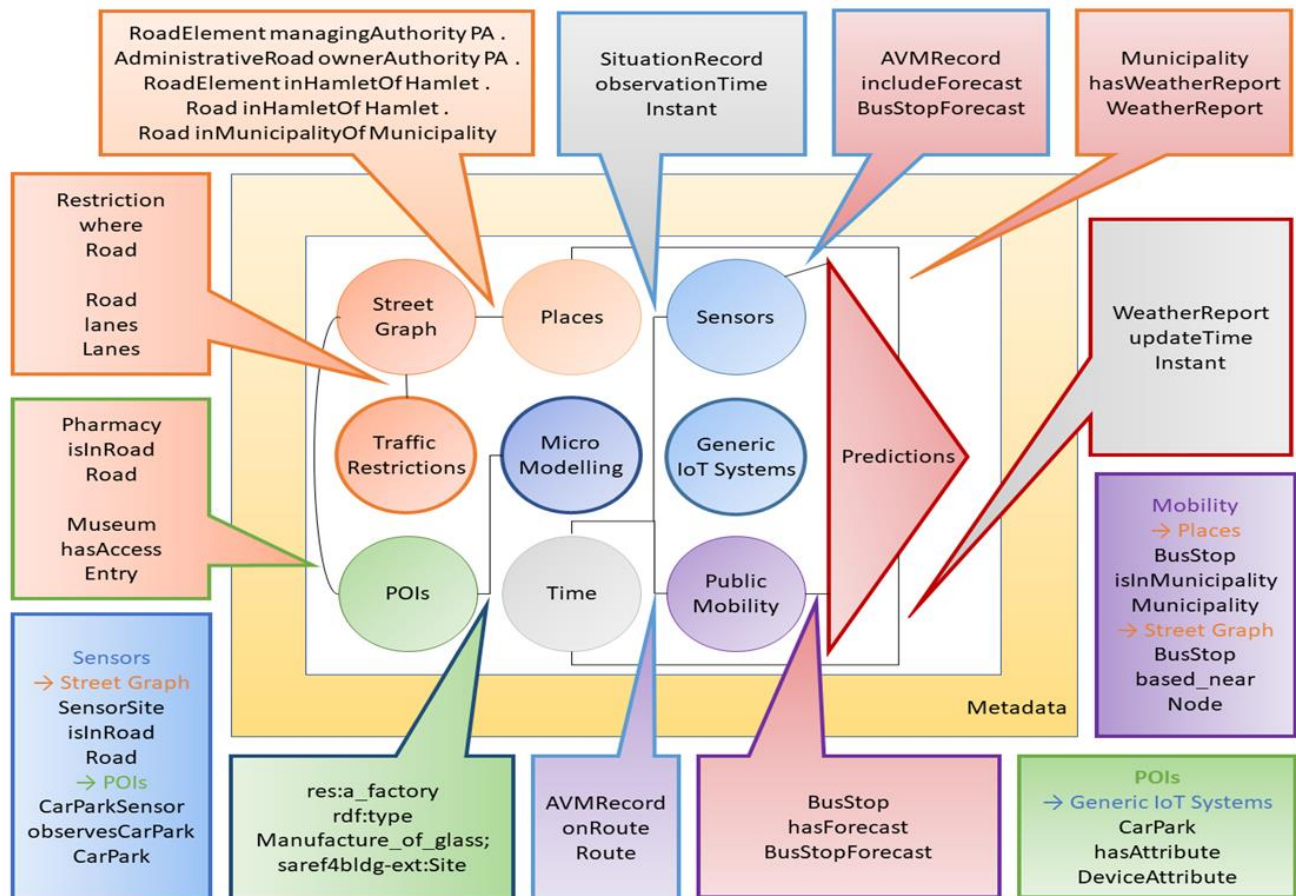
of fuel, and much more. For example, sensor devices can be used to model data collected for monitoring pollution, pollination, traffic, people tracking, parking, RTZ, irrigators, etc. In the same manner, forecast data can be performed for weather, parking, road accidents, people flow, vehicle flows, 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.

The **Km4City ontology** 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;
- *schema.org*, for a description of people and organizations;
- *wgs84_pos*, vocabulary representing latitude and longitude with the WGS84 Datum of geo-objects, since the present RDF store and indexing engine allows to perform geographic queries for example to identify the POI which 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;
- *SSN*, semantic sensor network, a W3C ontology for the modelling of IoT Systems;
- *iot-lite*, aimed at lightweight modelling of the Internet-of-Things;
- *saref4bldg*, SAREF extension for building, aimed at modelling topologies, items, and flows, at a high level of detail.

Other Vocabularies can be added to Km4City Ontology.

Then the view of the Snap4City **Data Inspector** (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.

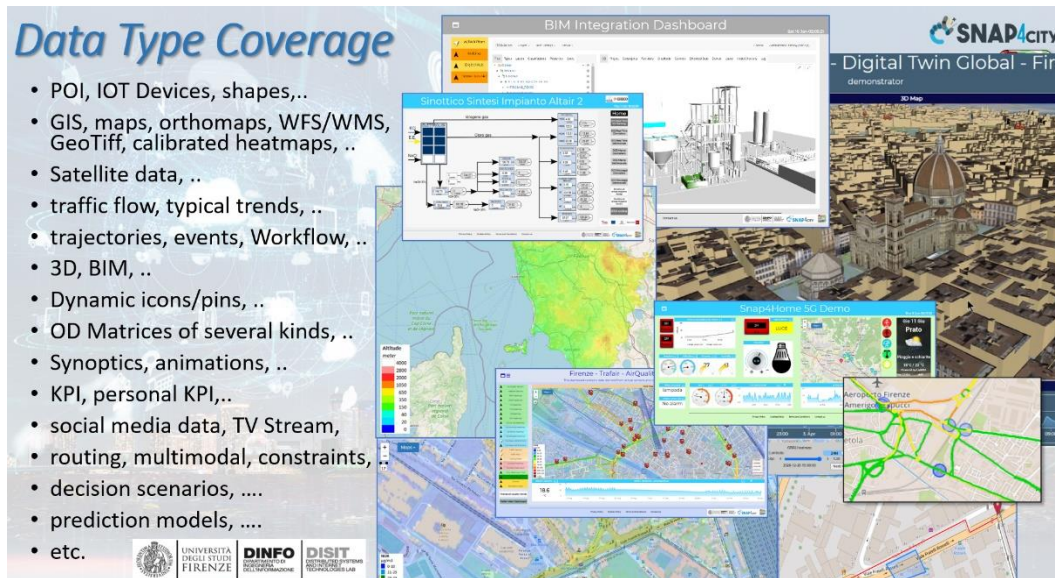
In this regard, **the Entity/Data Inspector**

- 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

Snap4City has the concept of **High-Level Type**, that are unified with a **Unified Data Model**. The HLT are 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**.

They are grouped by categories as follows:



- **Devices/Entities and their model and variables (Managed from IoT Directory):**
 - **Entity Model (IoT Device Model), Entity Instance (IoT Device), Entity Variable (IoT Device Variable)** (managed by IoT Directory)
 - Entity Models can be derived from FIWARE Smart Data Models
 - **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.
 - Entity Models can be derived from FIWARE Smart Data Models
 - **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.
 - BIM devices are actually Entity Instances (managed by 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** devices are actually Entity Instances (managed by the File Manager from **Data Management**)
 - 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.

- **My Personal Data, MyData** (the latter are mainly deprecated). The former are used to save user profile information such as password of hash, scenarios, etc.
- **My POI** (mainly managed as Entity Models). They are static point of interest that can be born private and then made public when approved. POI 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.
- KPI: deprecated HLT
- Map features and tools in addition to all the above data which can be located on map (Managed from **Data Management, HLT**), also:
 - **Heatmap** (managed by Heatmap manager from **Data Management**) 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.
 - **Traffic Flows** (managed by Traffic Flows manager from **Data Management**) 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.
 - **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**). They are produced from some original data, census data, cell phone, data, mobile data, etc., and loaded in terms of OD data on the OD Manager/server for distribution on map. They can refer to different geo areas, grids, ace, regional administrative shapes, etc. They can be time series of data, inflow and outflows, etc.
 - **Tools**: what-if, scenarios, etc. (as tools accessible into the **Dashboard Wizard**) They can be shown on the Multi Data Map on the dashboard and are
 - **Scenario**: for creating critical 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**: 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.
 - **WFS** (WMS, etc.), for connection to GIS for images, maps, orthomaps, shapes, POI, PIN, etc.
 - **3D models of the City, Global Digital Twin**. They are generated with a suite of tools starting from Lidar, GIS shapes, facades images, orthomaps images, etc.
 - **Trajectories**, Paths, shapes, etc. 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.
- 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.
 - **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.

- **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.
- **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.

The following HLT are special elements:

- **Dashboard-IOT App:** representing the data flow, even driven from IoT App to Dashboards and vice versa. They can be intercepted and exploited.
- **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 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

The smart solutions consist of

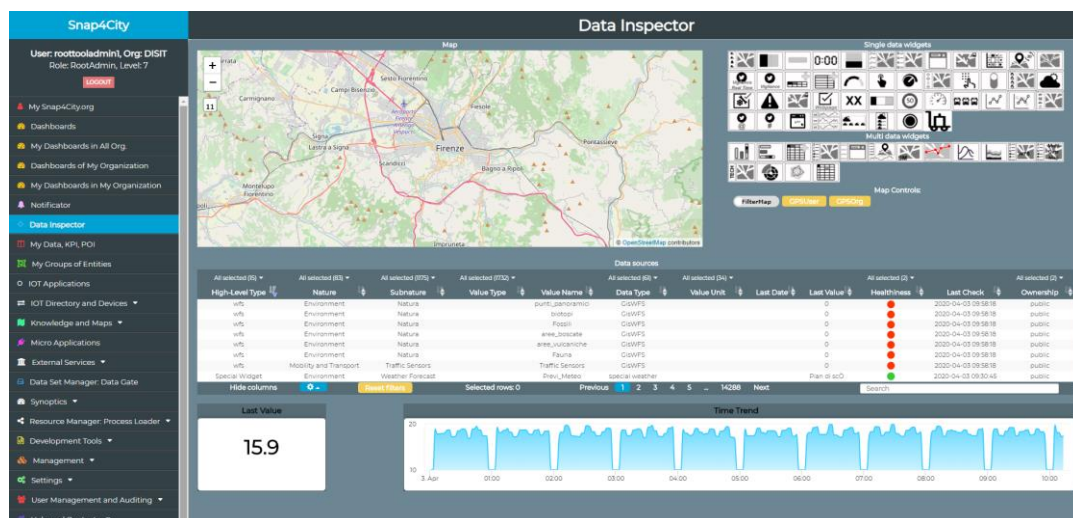
- **Entities / Data (and their Digital Twin information):** data are heterogeneous, large volumes, several protocols, legacy data systems, several semantics, real-time, multiple domains, etc., such as: any kind of data for the smart city. They belong to the so-called High-Level **Types, HLT, described above**.
- **Processes:** which can be: aperiodic, periodic, event-driven, managed by different tools (Brokers, ServiceMap, Heatmap Manager, Web Scrapers, Proc.Logic/IoT Apps, Traffic Flow Manager, etc.):
 - **Data Ingestion:** the models and mechanism for data gathering, Data Table Loader, POI Loader, IoT App, directly via Broker, etc.
 - **Data Transformation: IOT App / Node-RED, integrations, ..**
 - **Data Storage, indexing, etc.**
 - **Data Analytics: predictions, anomaly detection, computing, etc.**
 - **Business Logic behind Dashboards, server and client side.**
 - **Data rendering, Dashboarding, interacting.**
 - **Etc.**
- **Users:** owners (who register), responsible, providers, developers, etc.
- **Relationships:** among data, processes, users and mixed.

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:

- **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/das/>
- **of your Organization** only, that are public within your organization, and your private, and those that have been delegated to you:



To open the data inspector for a given Entity please click on the microbutton red/green marking the healthiness.

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

- **Cross Filtering based on:**
 - **MAP:** pan and zoom → lock , center on GPS coordinates of the user
 - **Data Source Classification:**
 - faceted filtering
 - full-text search
 - **Click on data source** to see it on the map, and see the graphical representation, just to learn how to “widgeting” it.
 - **Full-Text Search** if you remember some description...
- **Selecting a Data Source on Map:** on its Pin you can see:
 - **Real-time data**
 - **Time trend: 4 hours, 12 hours, 1 day, 1 week, 30 days.**
 - **Full status and description (only for Administrators)**

Click with the mouse on it

HLT: Sensor

Knowledge Base view

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.

HLT: Sensor

Some functionalities are limited to certain roles

91

Details regarding the IOT Ingestion process

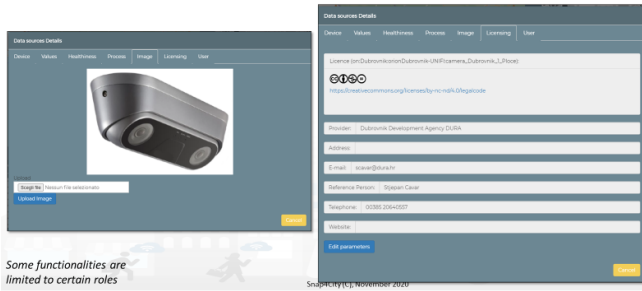
Some functionalities are limited to certain roles

91

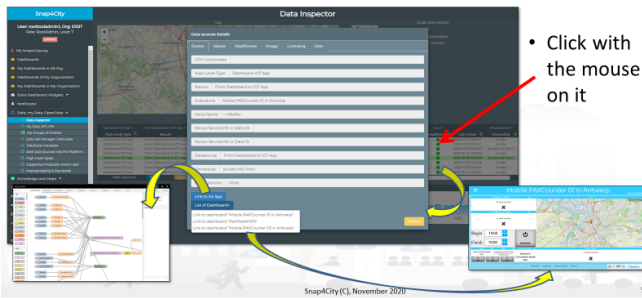
- Specific values of selected
- Information of the values of the other sensors on the same device
- View Trends, marking problems, healthiness by point according to a Fuzzy model
- Marking problems for future machine learning processes (separate tool)

- For IOT Device data
- IOT Broker details

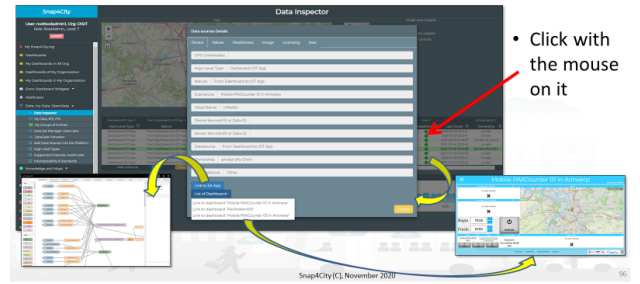
Image of the Devices and Licensing



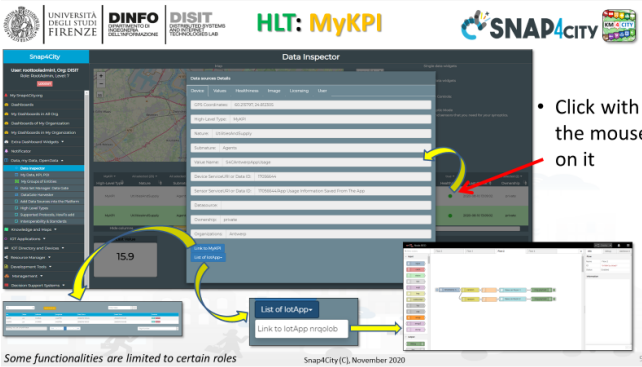
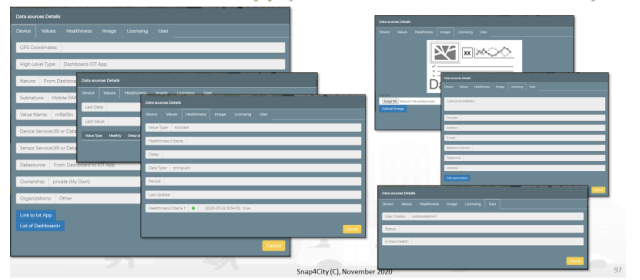
HLT: From Dashboard to IOT APP



HLT: From Dashboard to IOT APP



HLT: Dashboard-IOT App (From Dashboard to IOT APP)



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	OpenSearch, KB	Smart City API, Orion Broker
MyKPI (time series)	Proc.Logic / IoT App	OpenSearch (or SQL)	Smart City API
POI	Proc.Logic / IoT App	OpenSearch, KB	Smart City API
Heatmaps (time series)	Proc.Logic / IoT App, Python,	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	GeoServer, Traffic Flow Manager, KB	Traffic Flow Manager, API for WFS, WMS
OD Matrices (time series)	Proc.Logic / IoT App, Python	GeoServer, OD Manager	OD Manager, API for WFS, WMS
Trajectories	Proc.Logic / IoT App, Python	OpenSearch, KB	Smart City API
TV CAM	Registering and IoT Device	OpenSearch, KB	Smart City API
BIM view	BIM Server, IFC	BIM Server, KB BIM Manager	BIM Manager
3D City Model	Python	GeoServer, OpenSearch, KB	Smart City API, API for WFS, WMS; HTTPS

The Big Data Storage 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**.

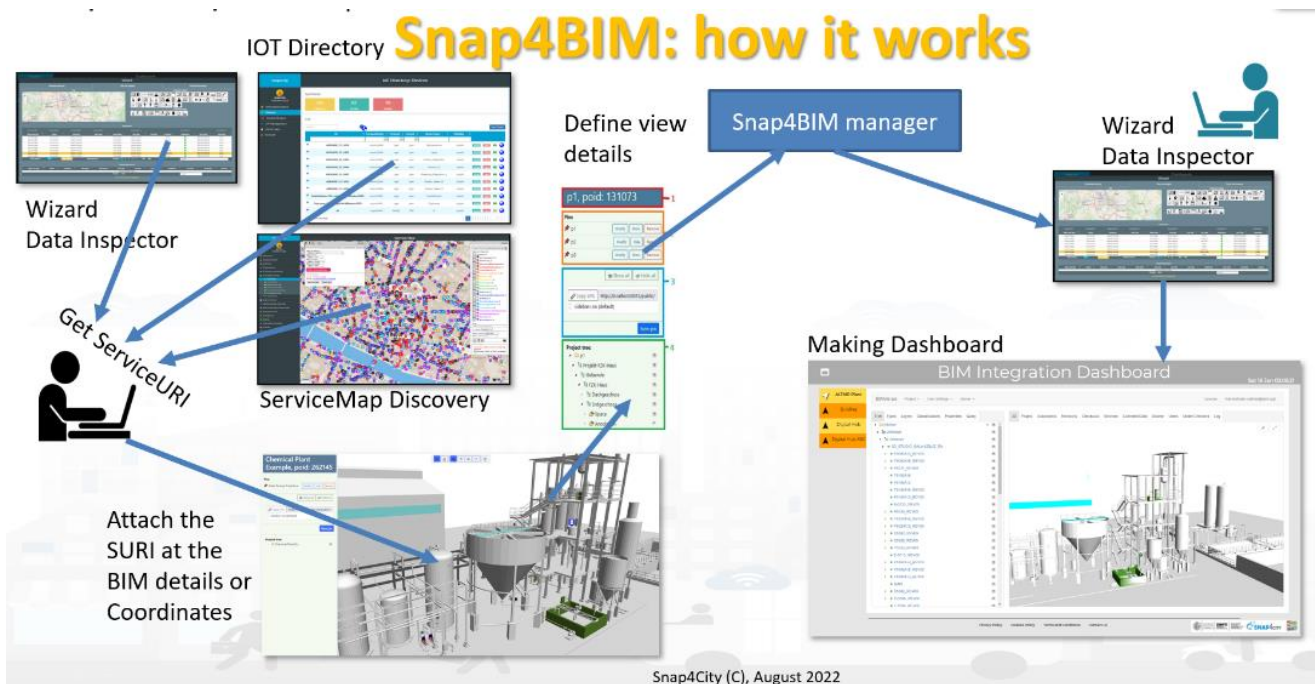
3.4.4 - Digital Twin Local and Global of the City, 3D City Representation

<https://digitaltwin.snap4city.org/>

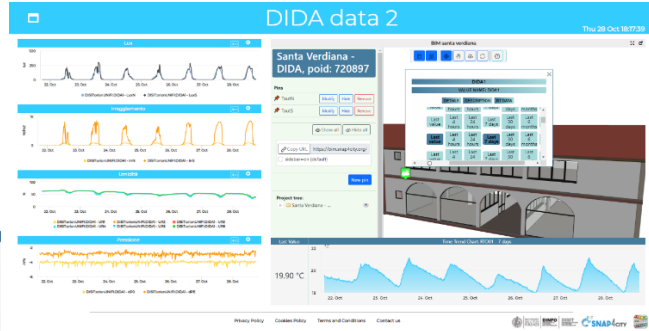
For Digital Twin we intend the integration from physical entities with digital entities. In the context of cities and industries we can have a digital counterpart describing all aspects including data flow, time series, structural details of the design, control aspects, and also 3D shapes to see them in the 3D representation in digital. The Digital Twin local refers to local details of a building or of a single machine, while the Global refers to the general view including the modelling of the interaction among the 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.

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.

Integration of Urban Modeling with Artificial Intelligence and Simulation



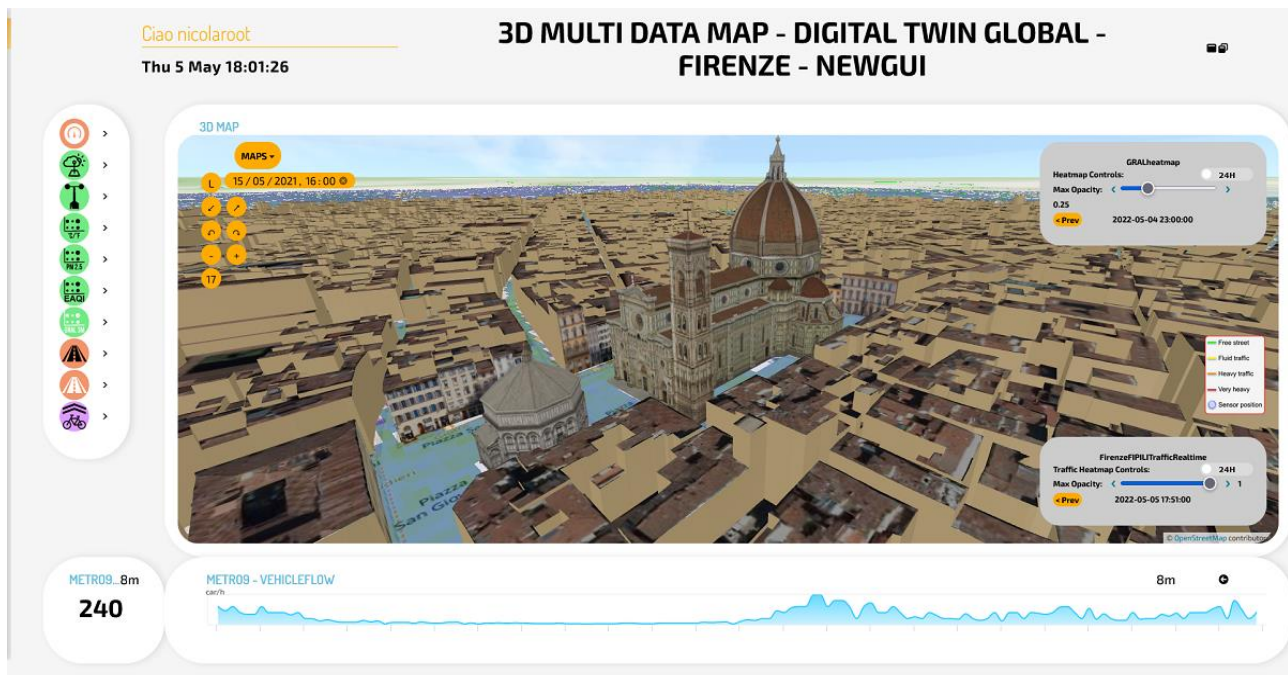
Local Digital Twin: detailed aspects modelled in digital of a physical element, such as a chemical plant with machines, motors, silos, etc., each of them with 3D representation, detailed components, and real-time values attached sensors that can be used to understand its behaviour in terms of real-time data and functionalities, mechanical parts to be disassembled for maintenance and inspection, etc. 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** and **BIM Server** (<https://www.snap4city.org/730>). The 3D shape is obtained by starting from formats such as IFC, DXF, SVG (Scalable Vector Graphics), STEP, IGES, STL (Stereolithography), OBJ (Wavefront), DAE (Collada), SCAD (OpenSCAD), and IV (Inventor). Some of them are open formats, others are proprietary and a large number of converters are present, even if several problems arise during conversions. The Local Digital Twin also may include the detailed description of floors, detailed description of data, devices, etc.



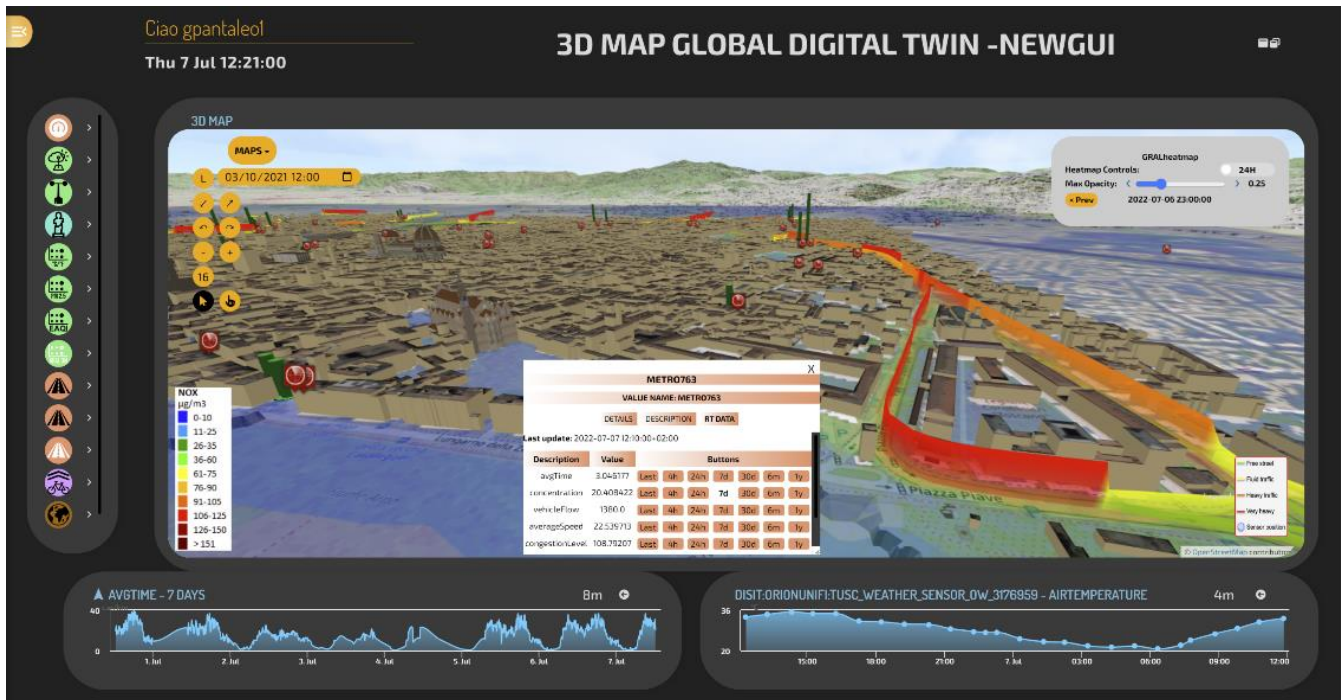
BIM representation in Snap4City provides an application for: Connecting BIM models with ServiceURI data into Snap4City/Industry, so that they can be: Entity / IoT Devices, POI, etc.; Observing PIN (icons representing an entity with its ServiceURI) from any view; Positioning PIN in any place of the 3D 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==>



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=MzQzMg==>

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.

FIWARE Smart Data Models Library				
Show 10 entries	Search:			
Name	Subdomain	Domain	Version	Edit
Alert	Alert	CrossSector	0.0.2	EDIT
Anomaly	Alert	CrossSector	0.0.2	EDIT
Battery	Battery	CrossSector	0.0.2	EDIT
BatteryStatus	Battery	CrossSector	0.0.2	EDIT
StorageBatteryDevice	Battery	CrossSector	0.0.2	EDIT
StorageBatteryMeasurement	Battery	CrossSector	0.0.2	EDIT
CallUser	CallComplaints	CrossSector	0.0.1	EDIT
Complaint	CallComplaints	CrossSector	0.0.1	EDIT
ComplaintsCollection	CallComplaints	CrossSector	0.0.2	EDIT
ComplaintsOrganization	CallComplaints	CrossSector	0.0.2	EDIT
Showing 1 to 10 of 441 entries 1 row selected				
Previous 1 2 3 4 5 ... 45 Next				

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.

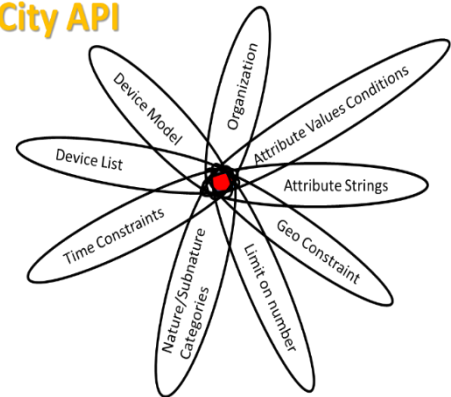
3.5 - Snap4City APIs (API Man, Smart City APIs, MicroApplications, Web and Mobile Apps)

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 API:

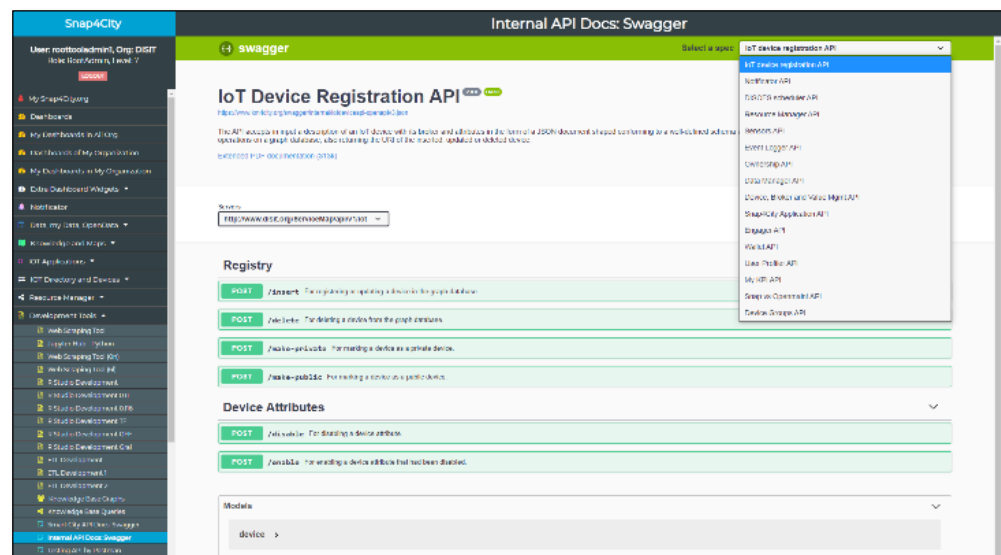
- **Advanced Smart City API** are provided from ServiceMap tool connected to the graph data base with the Km4City ontology and all the data relationships among entities. The requests on Smart City API can allow to perform temporal, spatial, relational and values filtered constraints. The Smart City API provides the first entry point to access data relational and Time Series.
- **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.
- **APIMan** 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. **APIMAN** may have an integrated SSO with the rest of the Snap4City platform or an independent KeyCloak or other Authentication service connected.

Selection on Smart City API



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.



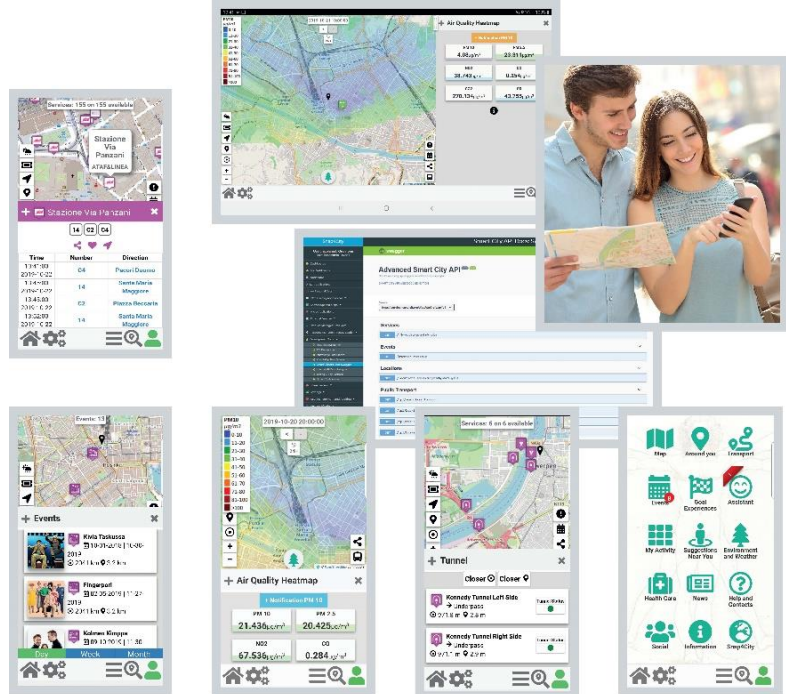
External APIs: <https://www.km4city.org/swagger/external/index.html>

- **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.
- **Heatmap:** To save and access **HeatMaps** of the Heatmap server

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, 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);
- **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 suggestions**, assistance, and incentives for a more virtuous behaviour
- get and set, etc.



3.5.1.a – Examples of Snap4City Mobile Apps

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

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

- Android: <https://play.google.com/store/apps/details?id=org.disit.snap4city.mobileApp.helsinki&hl=en&gl=US>
 - Apple iOS: <https://apps.apple.com/it/app/helsinki-in-a-snap/id1466970280>
- **Tuscany in a Snap**
 - Android: <https://play.google.com/store/apps/details?id=org.disit.snap4city.mobileApp.tuscany&hl=en&gl=US>
 - Apple iOS: <https://apps.apple.com/us/app/toscana-in-a-snap/id1471094480>
- **Snap4Pisa: Pisa in a Snap:**
 - Android <https://snap4city.aedit.it/drupal/sites/default/files/snap.apk>
- **Weee Life Mobile App:**
 - Android: <https://play.google.com/store/apps/details?id=org.disit.lifeweee&hl=en&gl=US>
 - Apple iOS: <https://apps.apple.com/it/app/life-weee/id1470224854>
- **Comune Facile**
 - Android: <https://www.e015.regione.lombardia.it/site/app-detail?id=131>
 - Comune Facile IOS: <https://www.e015.regione.lombardia.it/site/app-detail?id=130>
- **POS by EPSON:**
 - <https://www.e015.regione.lombardia.it/site/app-detail?id=58>
- **SnapBot:**
 - Telegram Bot based on Snap4City data and solution on Tuscany area
 - <https://www.snap4city.org/684>
- **Feel Florence Mobile App**
 - section regarding Herit Data
 - <https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=M...>

3.5.1.b - Snap4City Internal API

Internal APIs: <https://www.km4city.org/swagger/internal/index.html>

- **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 Mgmt 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 of 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

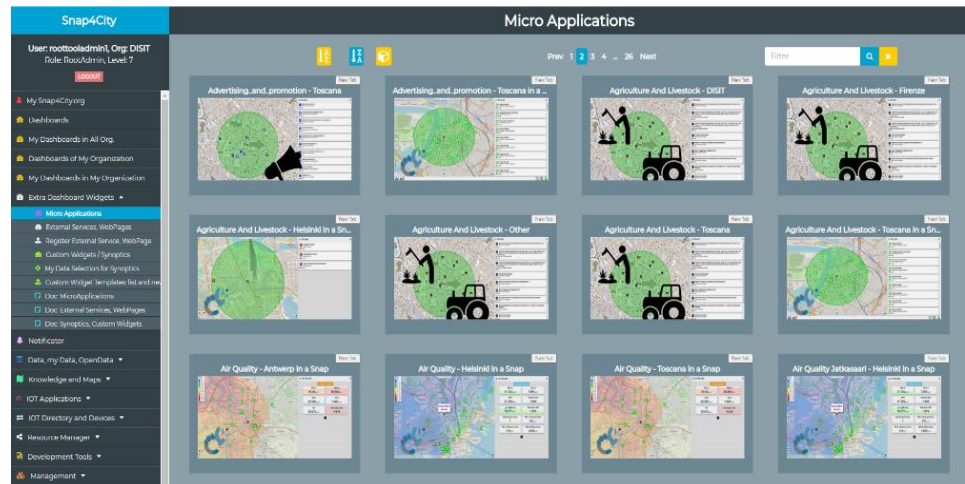
3.5.2 - Snap4City MicroApplications

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.

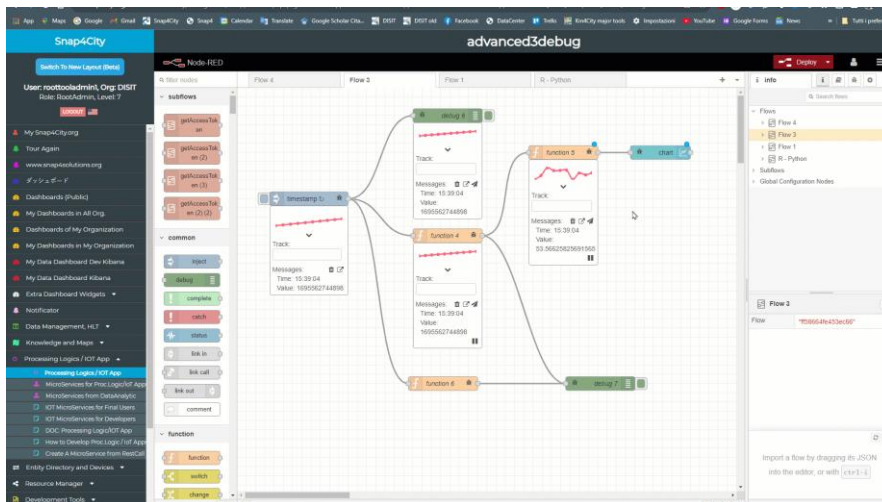


Using the APIs you can develop your Web Applications.

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.

3.5.2.a - 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 analyze 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.

3.5.2.b - 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>

3.5.2.c - 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:

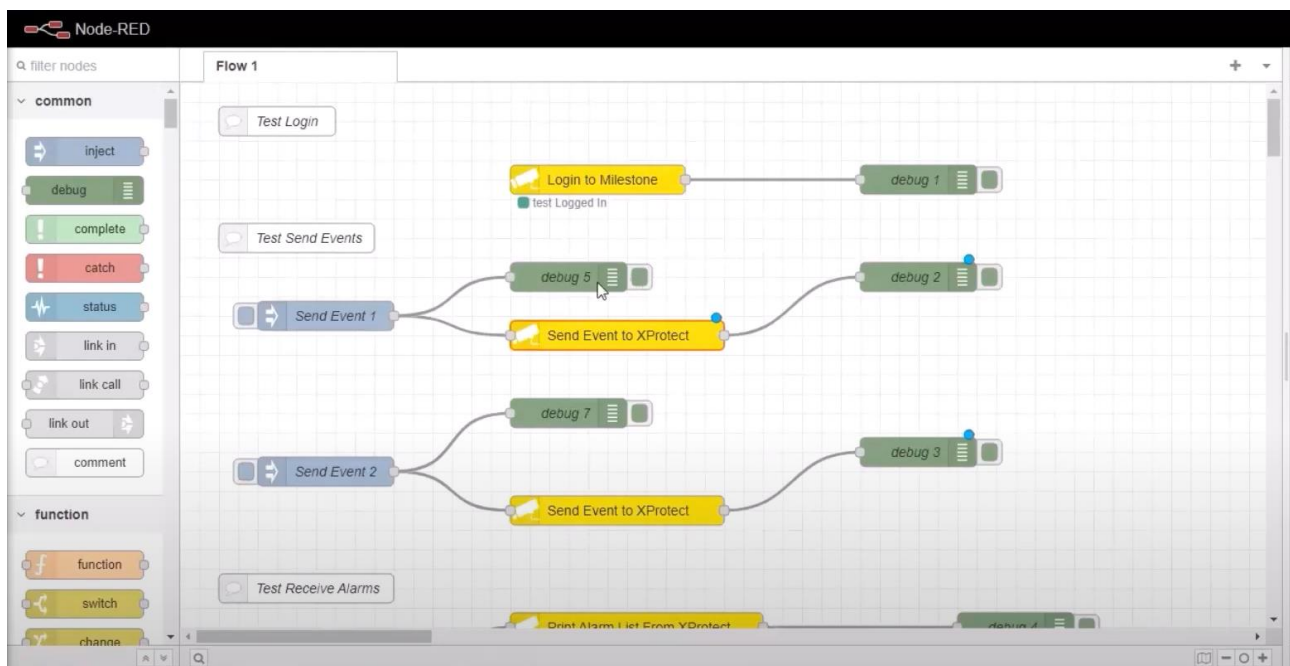
To satisfy the smart city requirements, in Snap4City, a collection of more than 190 MicroServices, as Nodes for the Node-RED programming environment, has been developed.

- <https://flows.nodered.org/node/node-red-contrib-snap4city-milestone>
- See video: <https://www.youtube.com/watch?v=dxYtOKOkvE&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, analyze 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 analyze motion data from Milestone VMS cameras and send only relevant information to Snap4City.
- **Data Analysis:** Node-RED can be used to analyze 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 analyzed 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, analyzing 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.

3.5.3 - Snap4City Mobile Applications and User Behaviour Understanding

The **Snap4City mobile Apps** are bidirectional tools. They provide the users with geolocated and contextualized information, 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, environment, government service, health, weather, etc. On the other hand, mobile users also provide information to the Snap4City Platform from which the platform can Derive information: trajectories, hot places, OD matrices, etc. 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/app/>

The App is a Bidirectional Device



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

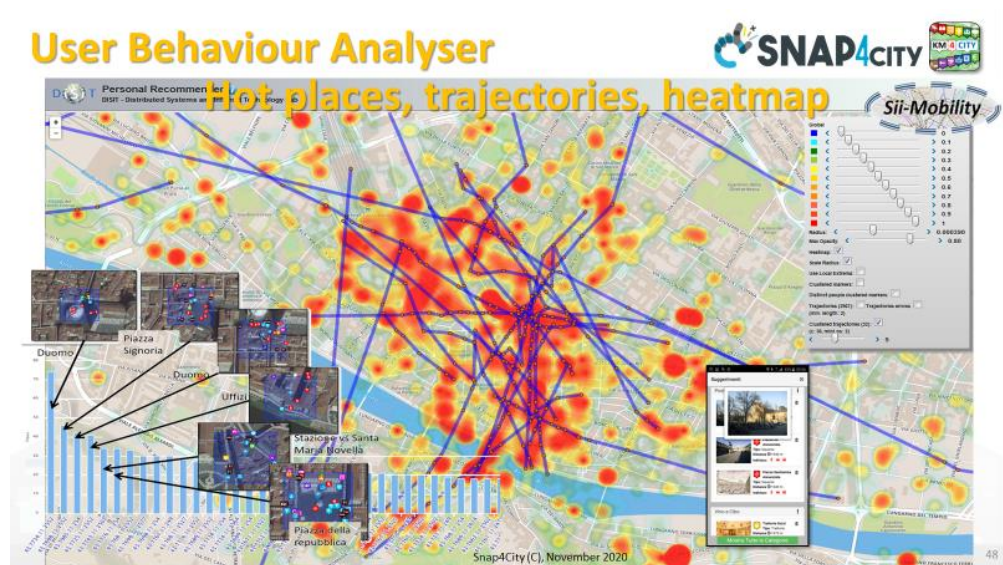
https://play.google.com/store/apps/details?id=org.disit.snap4city.mobileApp.antwerp&hl=en_US

USER MANUAL OF:

- Helsinki in a Snap: <https://www.snap4city.org/drupal/node/492>
- Helsinki in a Snap: <https://www.snap4city.org/drupal/node/493>

Personalization and Profiling, user behaviour analysis

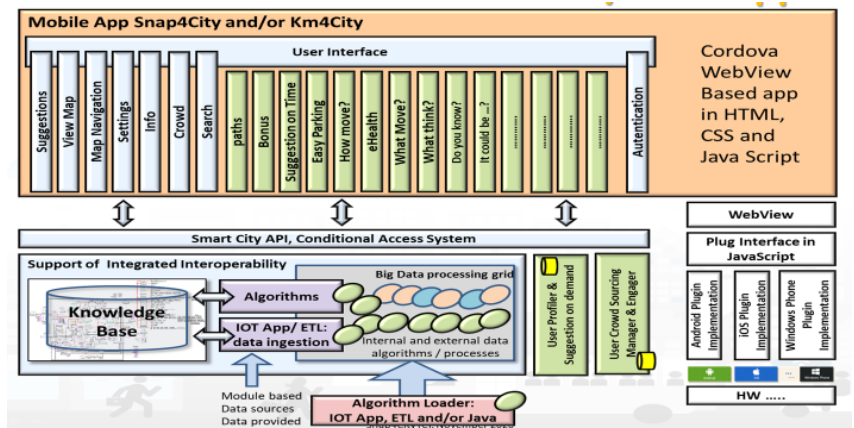
Apps Users are profiled in the server according to the GDPR, collecting a range of data as described above. This approach allows us to tune the service according to the collective and personal user behaviour, to improve the user experience by providing personalized (i) main menu arrangement of the functionalities 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)

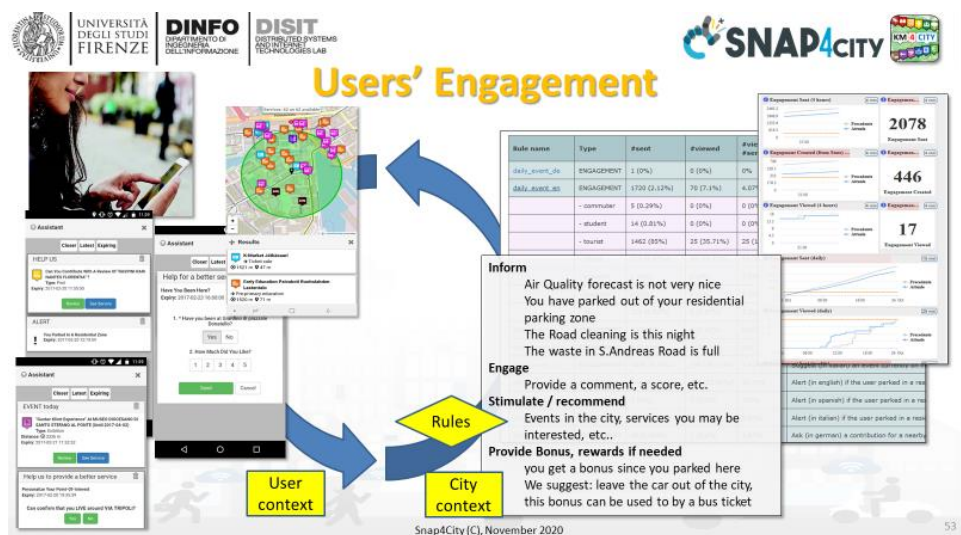
The App Development Kit is based on the Apache Cordova framework which allows realizing hybrid applications on multiple platforms. The applications consist of a user interface implemented with HTML, CSS, and JavaScript, and plugins that allow using the specific hardware functionalities from the different 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 - User Engagement via Mobile App

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

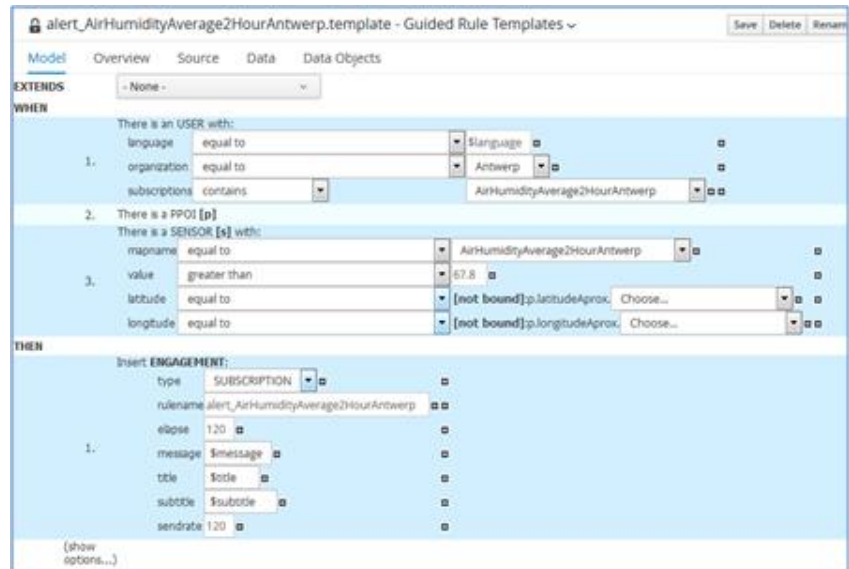


User Engagement is one of the main challenges in Smart Cities. Most of the smart applications imply some involvement of the city users, or at least the evidence of their success can be measured in terms of the city user appreciation and reaction to the proposals performed via Mobile Apps.

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

3.5.5 - Origin Destination Matrices, Algorithms, and tools

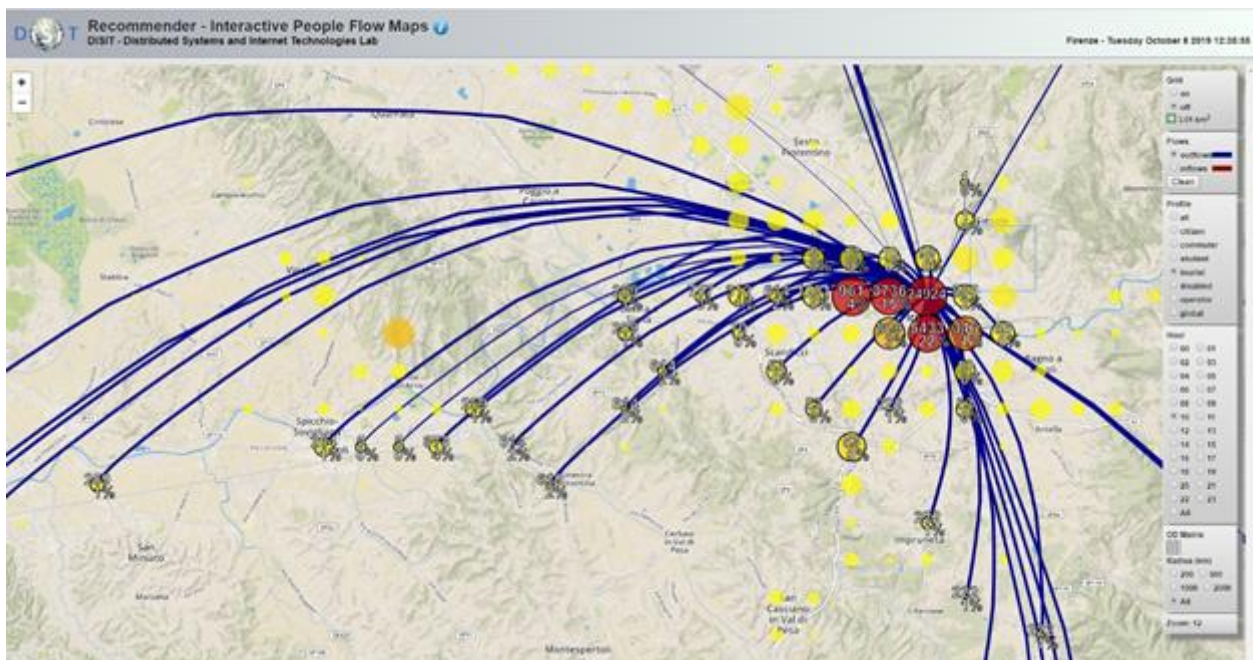
The City Operators and Strategists on mobility, transport, tourism and economic development need to have a precise vision of how the city users are moving in the city. From where they arrive, which kind of travel means are used, how they move in the city and in which time slots, etc. Most of these answers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

- **Traffic flow sensors:** reporting the private traffic flow and eventually traffic from touristic busses, and not in the areas close to traffic. If they take plates of the vehicles, they can be capable to be used for creating matrices.
- **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.
- **Onboard units:** for example, coming from taxis, from insurance black boxes, bus OBU, etc. Also, in this case a partial view of the flow is produced according to the different 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.
- **People tracking** tools, for example using Wi-Fi and/or Bluetooth sniffing devices, TV cameras, etc.
- **Cellular network data**, coming from the Telecom Operators, referring to cells or clustered areas.

Due to the introduction of GDPR, some of the above data are becoming harder or practically impossible to be obtained. All these kinds of Origin-Destination Matrices can be (i) used for simulation and analysis tools, as well as (ii) for direct visualization. Tools for the simulation and analysis may derive suggestions for public transportation, city consumption, city cleaning, service tuning, etc. On the other hand, a direct visualization may allow one to perceive the critical aspects at a glance. For OD matrix visualization, the classical matrix representation is not very effective since the matrices tend to be very sparse. See for example the figure.



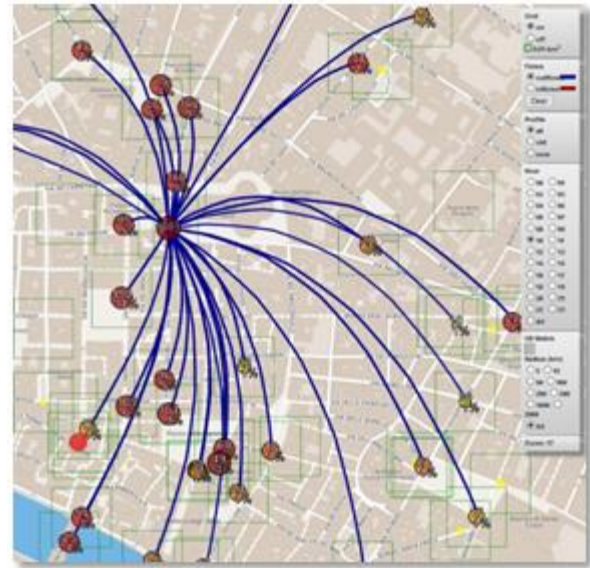
An alternative, directly invented by the DISIT Lab is the Spider Representation. The Visual Analytics tool for OD Matrix representation of the DISIT Lab is rendering OD matrices as Spiders is much more powerful for the direct understanding of the flows in the city.



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



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.

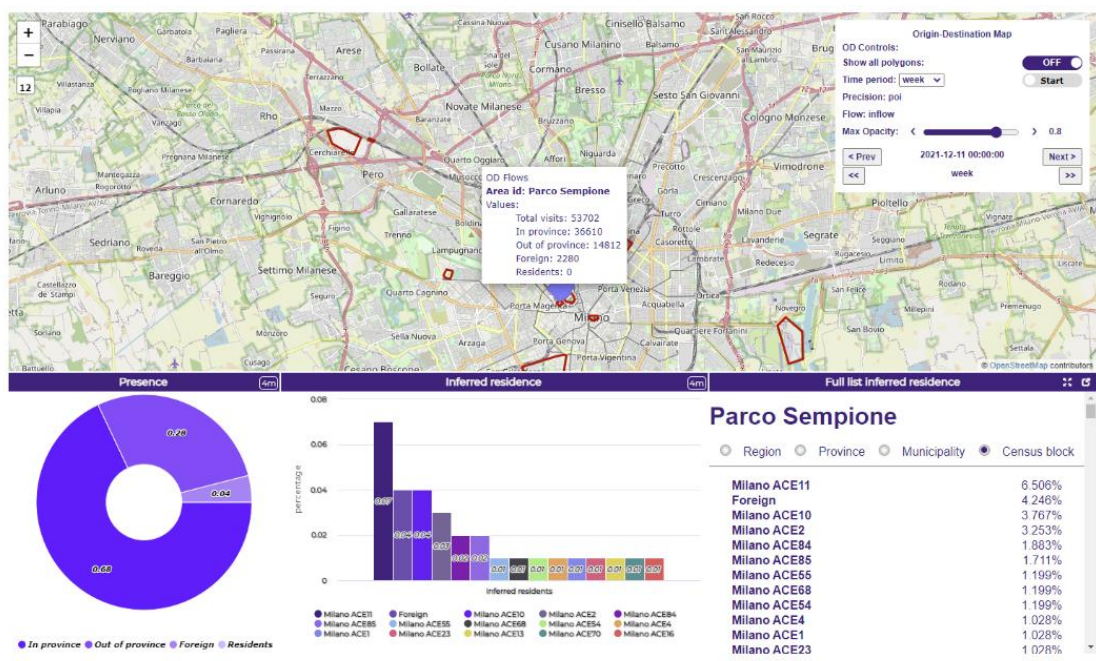
Simplified version of the tools could

be used for communicating with the citizens and business operators of the city, for example, see the Dashboards produced for *Life in Antwerp*, *Life in Helsinki* in which a view of ODM as spider is proposed. And those produced for ENEL-X presented here in the last figure.

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/drupal/node/459>
- Dashboard Life in Helsinki with OD matrix: <https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc1Mg==>
- Dashboard Life in Antwerp with OD matrix: <https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTcwNg==>
- Dashboard Life in Toscana with Origin Destination matrices: <https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc3NA==>



3.6 - Snap4City IoT Apps and MicroServices

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

Proc.Logic (as 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 transformation and processing. The Node-RED platform is based on two components:

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

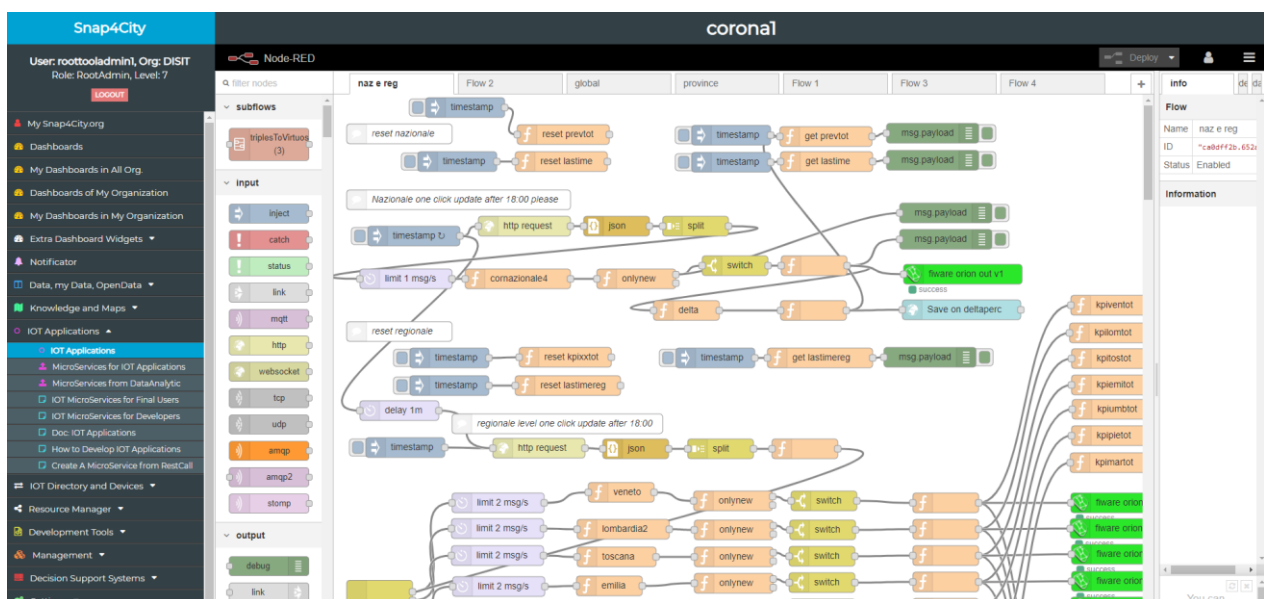
In Snap4City, the **Proc.Logic / IoT App/Node-RED visual editor** is fully integrated into the development environment and it has been improved to:

- communicate with the Snap4City **Resource 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 / IoT 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.

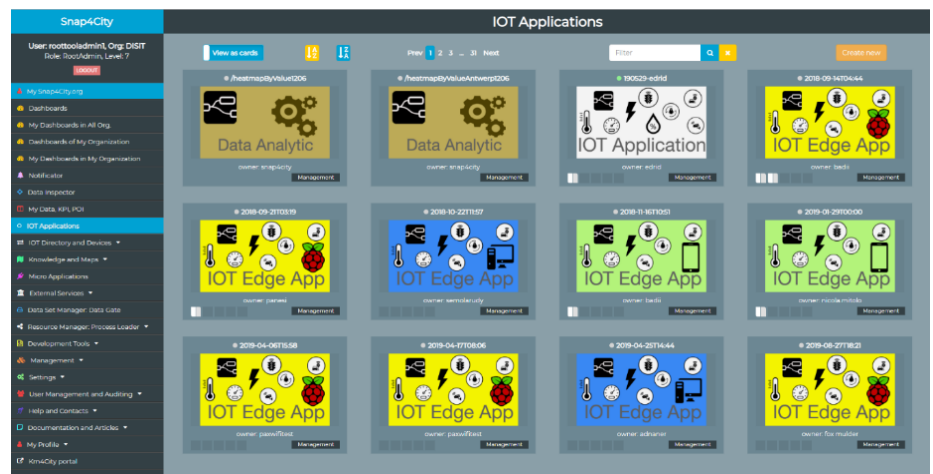


Snap4City platform offers a large set of **MicroServices**, which provide easy and formalized access to all the Smart City services that are available on-cloud from the platform (including the ones to control a part of the platform itself). They are made available in the Node-RED Node.JS environment 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 of Smart Cities

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>
- <https://flows.nodered.org/node/node-red-contrib-snap4city-developer>
- <https://flows.nodered.org/node/node-red-contrib-snap4city-d3-dashboard-widgets>
- <https://flows.nodered.org/node/node-red-contrib-snap4city-tunnel>

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.



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

- **Data ingestion:** more than 100 protocols IoT and Industry 4.0, Web Scraping, external services, any protocol database, etc.
- **Data access:** save/retrieve data, query search on the expert system, geo-reverse solution, search on expert system Km4City ontology, etc.
- **Data Transformation/transcoding:** binary, hexadecimal, XML, JSON, String, any format, etc.
- **Integration:** CKAN, Web Scraping, FTP, Copernicus satellite, Twitter Vigilance, Workflow OpenMaint, Digital Twin BIMServer, any external service REST Call, etc.
- **Manipulated complex data:** heatmaps, scenarios, typical time trends, multi-series, calendar, maps, routing, etc.
- **Access to Smart City Entities and exploitation of Smart City Services:** transport, parking, POI, KPI, personal data, scenarios, etc.
- **Data Analytic:** managing Python native, calling and scheduling Python/RStudio containers as snap4city microservices, etc. (predictions, anomaly detection, statistics, etc.)
- **User interaction on Dashboard:** get data and messages from the user interface, providing messages to the user (form, buttons, switches, animations, selector, maps, etc.)

- **Dashboard widgets event-driven:** D3 any, spider-net, barseries, time series, multi-time series, donut, buttons, knob, switch, single content, data tables, maps, 3D maps, 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, Twitter Vigilance and sentiment analysis, etc.
- **Hardware Specific Devices:** Raspberry Pi, Android, Philips, video wall management, AXIS camera, etc.

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:

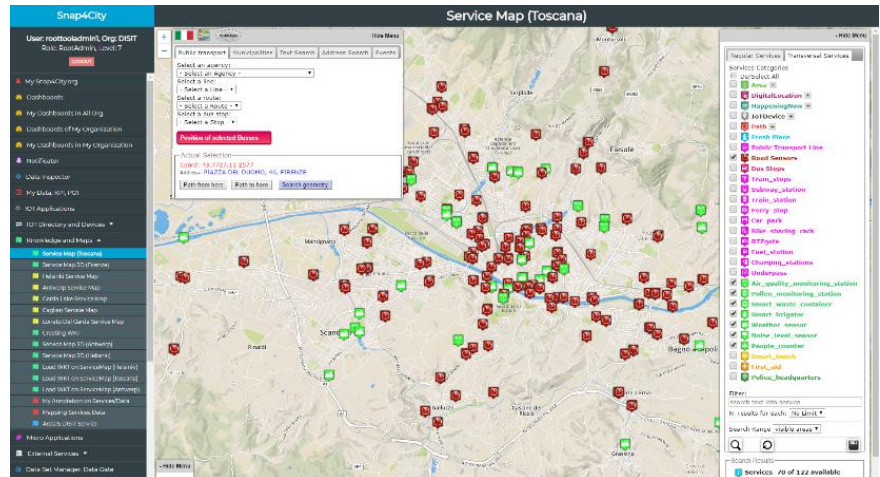
Access to Smart City Entities, which have different data models and, thus, different MicroServices may be required. Some Entities may have simple sets of Metadata, for example, the ones describing the POI, e.g., the title, description, web, email, GPS location, images, opening time, etc.; others may 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.;
- 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**, 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, 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**, or even in **Hadoop**, **HBase**, but **we considered a deprecated solution**. Therefore, the developer of the **Proc.Logic / IoT App** does not need to know all the tiny details about the large variety of adopted storage. Snap4City provides more than 70 different nodes for searching different Smart City entities, providing results in different data types: POIs, time trends, values, events, schedules of buses, bus lines, recommendations, addresses, routes, etc. This approach significantly simplifies the creation of Smart City Applications.

Discovering and Exploiting 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 data-driven 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 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 are provided by Snap4City, while others are managed by third parties, for example, to provide an answer to queries, such as "give me all temperature sensors close to my house" (regardless of the data providers, protocol, source, etc.).

Creation of an advanced Graphic User Interface, including graphics widgets, such as Dashboards, Virtual Devices that act on the 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 also allow for 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:

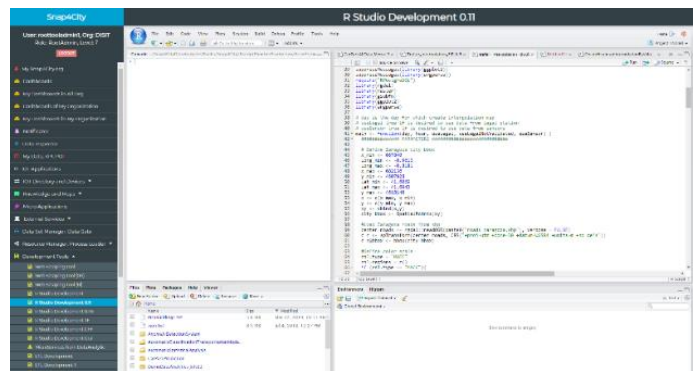
- include Visual Analytics tools, such as External Services, MicroApplications, moving object Trajectories, Origin Destination tools, Heatmaps, Traffic Flows, Scenarios, Routings, Maps of any kind, Orthomaps, etc.
- visualize data from other sources: Orion Brokers, Entity Instances / IoT Devices, data 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 can be connected via specific MicroServices automatically created by the Snap4City solution or provided by Node-RED.

Snap4City allows to

- 1) **create** any kind of **Data Analytic** algorithm in RStudio, Python and they are automatically **transformed into APIs and MicroServices** to be directly exploited in IoT Apps (for example developed in R. See Figure aside).
- 2) **Exploit Data Analytics processes created for the specific Infrastructure, that may be in Java or other services accessible via some APIs which can be mapped into a MicroService as well.** A **REST call** invokes **External Services**, for example, gaining access to Twitter Vigilance API, The Weather Channel API, The Things Network API, etc.



- 3) **Exploit external Data Analytics Services** via Node-RED nodes (Azure, SAP, Watson, etc.).

In addition, the **Snap4City.org** portal provides several ready-to-use RStudio examples, and some of them are ready-to-use MicroService/nodes implementing:

- a. A generic **Data Analytic MicroService**, where Data Analytic algorithms, developed in Rstudio or Python, can be executed, according to specific guidelines.
- b. **Data Analytics** for computing: routing, predictions, anomaly detection, origin-destination matrices, trajectories, suggestions, statistics, heatmaps, 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.

3.7 - Snap4City Data Ingestion/Interoperability

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

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.

Snap4City is widely interoperable since it can:

- 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>
 - **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, SMTP, TCP, UDP, SOAP, WSDL, FTP, FTPS, WebSocket, WebSocket Secure, GML, WFS, WMS, RTSP, ONVIF, AXIS TVCam, CISCO Meraki, OSM, Copernicus, The Weather Channel, Open Weather, OLAP, ...
 - **Formats:** JSON, GeoJSON, XML, CSV, GeoTIFF, OWL, WKT, KML, SHP, db, XLS, XLSX, TXT, HTML, CSS, SVG, IFC, XPD, OSM, Enfuser FMI, Lidar, gITF, 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, ...
 - **Social:** Twitter, FaceBook, Telegram, ...
 - **Events:** SMS, EMAIL, CAP, RSS Feed, ...
 - **OS:** Linux, Windows, Android, Raspberry Pi, AXIS, Local File System, ESP32, etc.



- 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.
- 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**.
- 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.
- **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.

- **Join, publish and harvest Open Data Portals** via DataGate/CKAN.
- **Interoperate with ERP via IoT App:** <https://flows.nodered.org/search?term=ERP>
- **Interoperate with BPM such as OpenMaint:** <https://www.openMAINT.org/en/home> and see <https://www.snap4city.org/625> or others tools.
- **Interoperable with BIM via BIM Server** <https://bimserver.org/> or other BIM manager
- 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>
- **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>

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

- **Smart City API which can be confined into a single Smart City installation or Federated as well as for Super ServiceMap WFS service of Snap4City** on top of Federated Smart City API or simple Smart City API of a single ServiceMap (smart City installation). This solution permits 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>
- **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>

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) disjointed and even far each other (this means that the union of all the areas can be disjointed 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;

Federated ServiceMap and Smart City API

To improve scalability, fault tolerance and federation among cities:

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

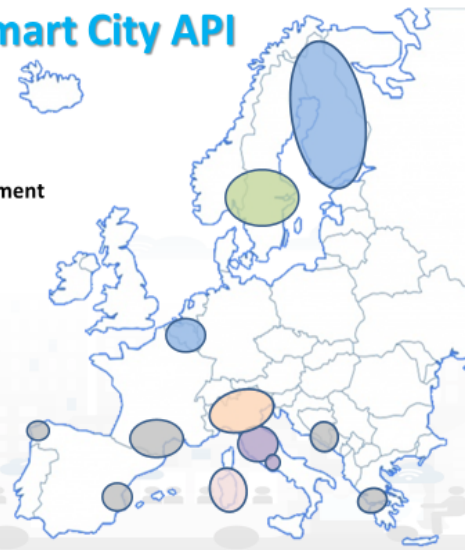
At different levels:

- Among cities/regions
- Among data providers, Operators

By Means of:

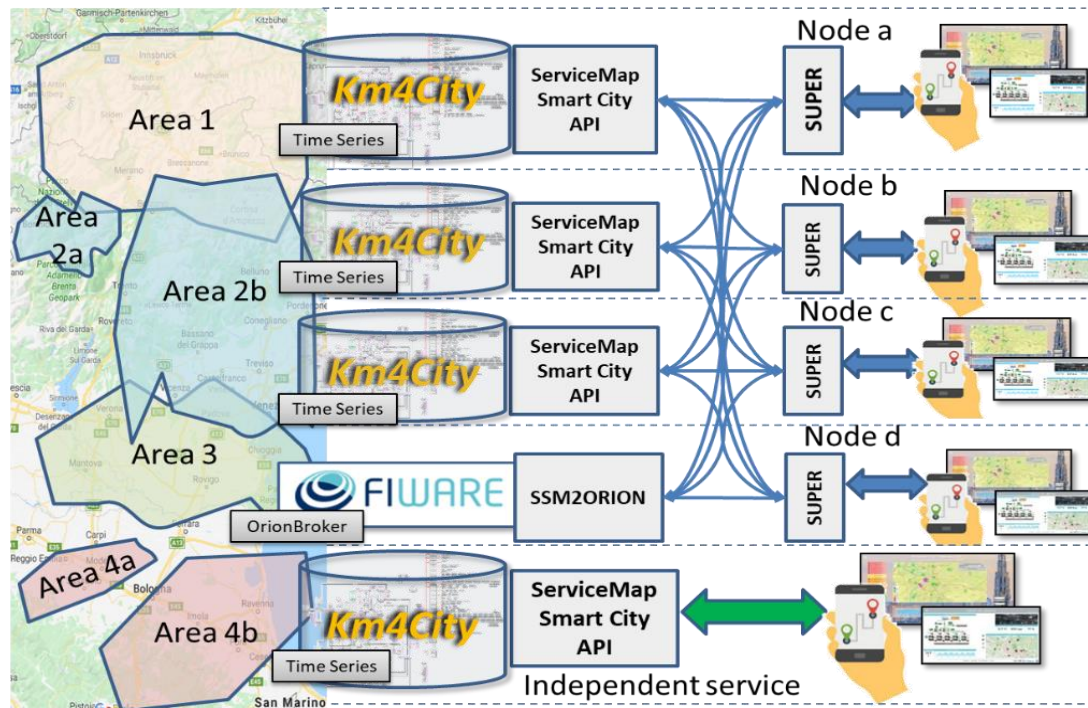
- Smart City API → Apps
- Smart City Ontology
- Dashboards/data analytics
- Organization independent
- CKAN via harvesting

Snap4City (C), November 2020



- 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.

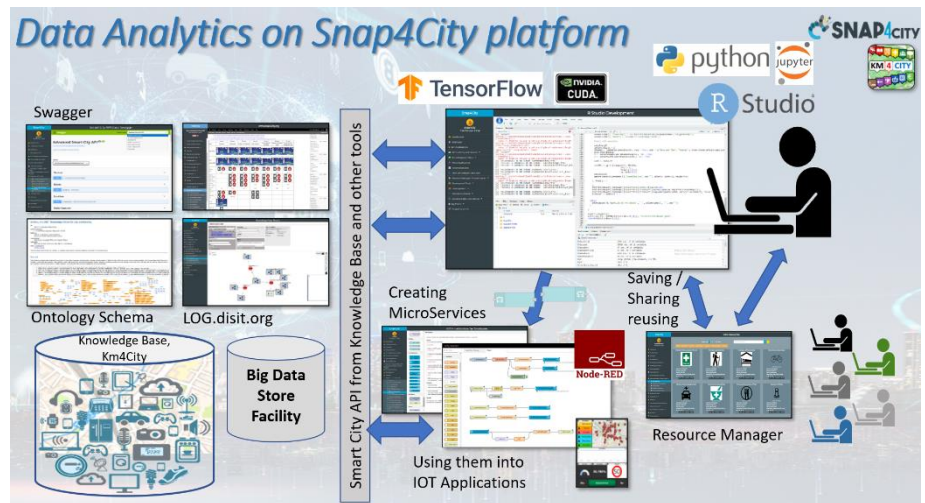
To avoid having a single point of failure, the **Federated Knowledge Bases Supers** can be replicated into each node and the list of Super services on top of ServiceMap can be put accessible on one or more web servers for the update. Each ServiceMap has a representation of the multi-polygon addressed 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

For the development of data analytics, the data scientist and developers can use **Python and/or RStudio** from the online platform and on their premises. Python and RStudio platforms may exploit any kind of libraries such

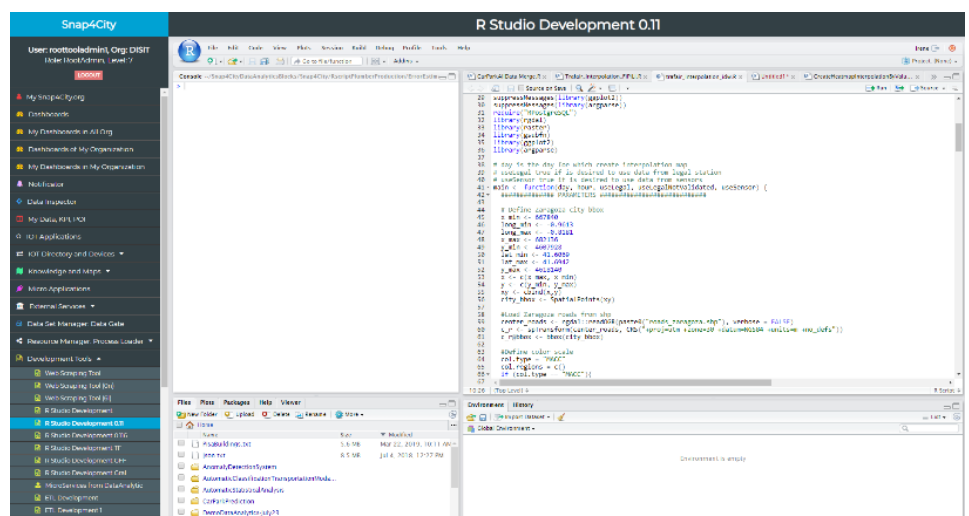
as Keras, Pandas, and hardware accelerator as NVIDIA to use Tensor Flow, etc. The developers can access the KB and Big Data store respecting the privacy and the data licensing by using authenticated Smart City APIs. The access has to permit the reading of historical and real-time data, and saving the resulting data provided by the algorithms, for example, heatmap-related predictions, the assessment of data quality, and labels of detected anomalies. Data scientist' work may be finished once they develop an algorithm (e.g., for computing h



may be finished once they develop the algorithm they should be aware of. On the other hand, the same algorithm (e.g., for computing heatmaps, parking prediction), should allow being:

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

In 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



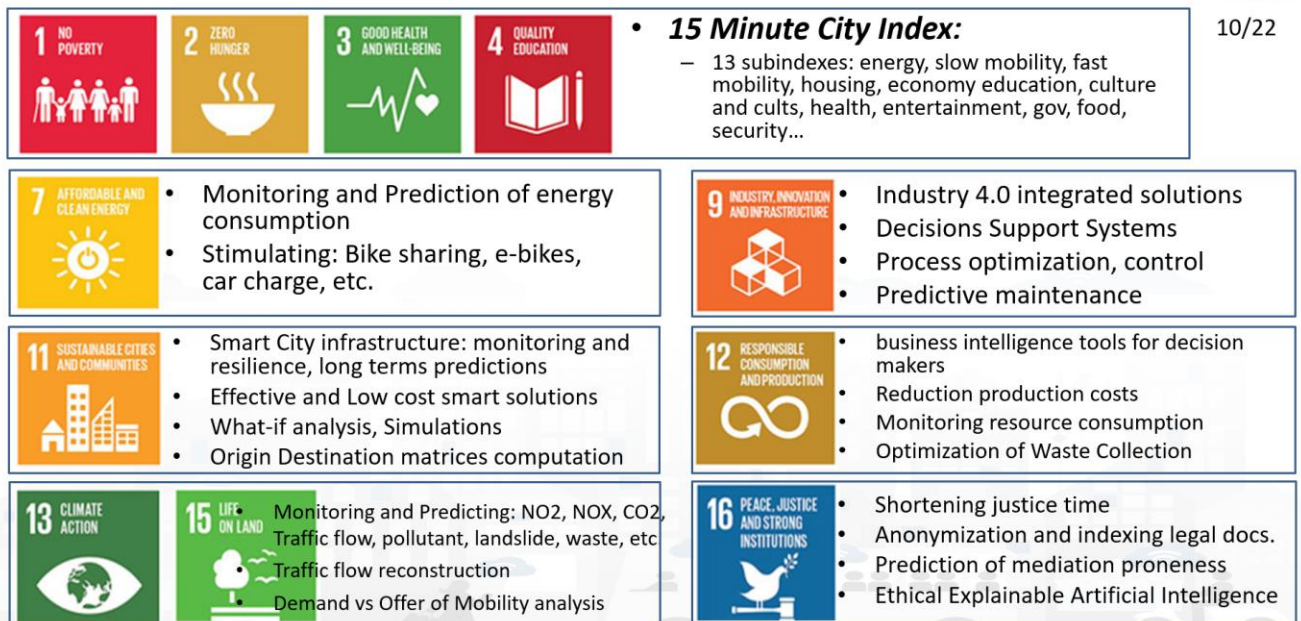
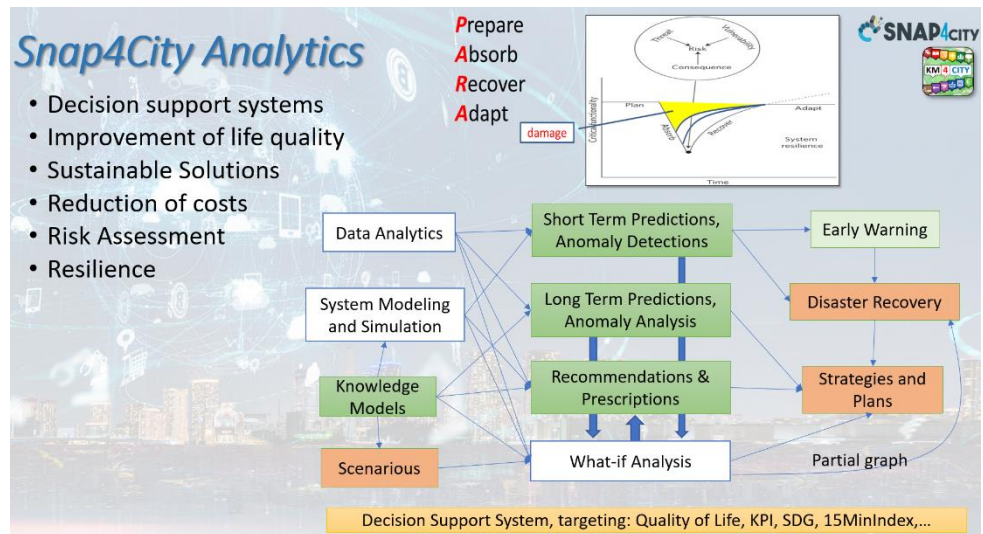
at the Smart City API with the needed permissions. The source code can be shared among developers with the tool "Resource Manager", which also allows the developers to perform queries and retrieve source code made available by other developers.

Read more on: <https://www.snap4city.org/download/video/course/da/>

3.9.1 - Snap4City Analytics, the data analytics at your disposal on Snap4City

The data analytics in Snap4City are focussed on providing support for Decision Makers to improve quality of life, match the Sustainable Development Goals, specific KPIs; and assess the conditions for 15 Min City Indexes.

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/da/>



Snap4City has been designed since the 2017 to be AI enabled, respecting ethics, secure passing the PENTest and GDPR compliant. Snap4City has developed a large number of solutions in the context of Smart City and Industry 4.0. Snap4City fully supports 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, and any kind of library for data analytics, ML and AI. Snap4City is distributing a number of Open-Source data analytics tools and algorithms for: prediction, anomaly detection, classification, detection, constrained routing, optimisation, analysis of demand vs offers of transportation, and many others have been published on international top level journals and can be customized on demand on your cases. Data Analytics is fully integrated into What-IF analysis tools in control rooms and for operators, defining scenarios and solutions. Snap4City has a consolidated

experience in the development, validation and transfer AI/XAI solutions (see course <https://www.snap4city.org/577>). Most of the DISIT lab solutions are based on ML, Deep learning, AI, XAI, natural language processing (NLP), sentiment analysis (SA), semantic reasoning and computing, neuro symbolic, generative AI, 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>

Mobility and Transport

- Impact on planned changes in city viability for: public transport, traffic, parking, people flow, etc.
- 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)
- 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)
- Counting and Tracking people via different kind of sensors and thermals cameras
- Tracking fleets 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)
- 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)

City Users and Social

- People detection and classification: persona, strollers, bikes, etc. (ML, DL)
- people counting and tracking, head counting (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.
 - 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)
- 15 Minute City Index , etc. (modeling and computability)
- Computing SDG, etc., (DP)

Environment and Weather

- Pollutant Predictions: short, long and very long term European Commission KPIs

- NOX, PM10 pollution on the basis of traffic flow, 48 hours (ML, AI, DL)
- Cumulated NO2 average value over the 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)
- 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)
- Optimisation of waste collection schedule and paths (DP, ML)
- Computing SDG, SUMI, PUMS, .. (mainly DP)

Energy and Control, Security

- Smart Light Solutions: monitoring luminaries, profiling luminaries, managing error conditions (DP)
- Design by Simulation of Photovoltaic Plants: using real statistical data from the area (ML, Dp)
- Energy Community: Energy Districts (in Italy, CER) (ML, DP)
 - Monitoring, design and simulation of energy community
- Monitoring and controlling recharging stations, recharging poles
- Monitoring energy production and consumption over: plant, building, floors, offices, server rooms, etc.
- Monitoring healthiness of Smart City Network of devices
- Monitoring critical areas for: people, traffic, boats, etc.
- Etc.

Management, Risk, Resilience and strategies

- Estimation of KPI and local indexes for: quality of life (15MinCityIndex, SUMI, SDG, ..)
- What-if analysis, dynamic routing, origin destination matrices production from a large range of sources
- Planning and Monitoring renovation works via objective KPIs
- Managing Maintenance and teams
- Predictive Maintenance and costs predictions: chemical plant, vehicles, boats (XAI)
- Production Optimization
- Resilience analysis wrt European Guidelines on Resilience of critical infrastructure, and
- Risk analysis: natural and non-natural disaster (XAI)

Semantic Reasoning

- Ontology Modelling and integration, expert system construction
- Multilingual Sentiment Analysis, transformers, BERT, Multilingual
- Knowledge modelling and reasoning on RDF stores: spatial, temporal, relational
- Virtual Assistant construction, virtual expert of the city.

Time Series, Matrices, Images, Maps and 3D Digital Models

- Time Series Anomaly detection
- Data quality assessment and control
- Short, long and very long term predictions
- Interpolation of scattered Data on regular grid for calibrated heatmaps
- Conversion of Satellite data images into regular ground images/heatmaps measures
- Extraction information from Orthomaps, LIDAR, etc., regarding city structures

- 3D Digital Twin of Cities and Objects: pattern extraction, 3D model reconstruction.
- Ontology Modelling and integration, expert system construction
- Knowledge modelling and reasoning on RDF stores: spatial, temporal, relational

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

- RF, XGBoost, BRNN, RNN, SVR, MLP, etc.
- Recent AI: DNN, LSTM, CNN-LSTM, Bi-LST, Transformer, Autoencoders, BERT, ..., YOLO, PINN, etc.
- Clustering: K-means, K-Medoid, k-Shape, etc.,
- Optimization: Simulated Annealing, Genetic Search, Taboo Search, etc.
- XAI: Shap, LIME, Gradient, Integrated Gradient, variations

Data analytics have to be targeted concerning to the **user kind** (decision maker, ICT official, Operator, city user, commuter, student, tourist, etc.) 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. The following image reports the mapping of the Data Analytics vs the Antwerp and Helsinki users and the main tools in which they have been made accessible in a transparent manner for the user.

	Antwerp					Helsinki								Where				Main Data Sources
	City official	ICT official	Developer	Citizen, tourist, visitor	Business owner	City officials	City officials Domain experts	City officials City developers	Third party developers	Citizen	Citizens with respiratory problems	Tourists	Business owners	Mobile	MicroApplication	Tool, via Portal (ICT Developers)	Dashboards	
Discovery near to me	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			POI, OSM
Discovery along a path	X	X	X	X	X	X	X	X	X	X	X	X	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	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	X	X		POI, OSM
Routing: pedestrian				X	X		X	X	X	X	X	X	X	X	X			OSM
Routing: pedestrian quite				X	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	X	X	X			OSM
Routing: Multimodal Public Transport				X	X		X	X	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			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			X	Sensors data, OSM
heatmaps: environmental variables, Noise	X	X		X	X	X	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			X	Enfuser data
heatmaps piking values any place	X	X			X	X	X	X	X	X	X	X	X	X			X	Computed Heatmps
heatmaps: GRAL prediction, PM10						X	X	X	X	X	X	X	X	X			X	OSM, Traffic, Weather
Comparsion: Enfuser, Gral, Real Time						X	X	X	X	X	X	X	X	X			X	Enfuser, Sensors, GRAL
Sensors Data Time Trends, & drill down	X	X	X	X	X	X	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			X	Forecast Service
Origin Destination Matrices	X	X	X	X	X	X	X	X	X				X				X	Snap4City Mobile App
Typical trajectories	X	X	X	X	X	X	X	X	X				X			X	X	Snap4City Mobile App
Hot Area in the city	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	Snap4City Mobile App
Hot Places in Smart Zone	X	X	X	X	X									X		X	X	Snap4City PAXcounters
Services Suggestions on mobiles				X						X	X	X		X	X			Snap4City Mobile App
Alerts on critical cases: several variables	X			X	X	X	X			X	X	X	X	X				Sensors data, OSM
The most used services		X		X	X	X				X	X	X	X				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

3.9.2 - What-IF Analysis Support

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. A tool of What-IF analysis has to be capable to answer questions such as: **What is going to happen at certain City Services if certain conditions/cases are going to occur?***



The What-if analysis process is following 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?iddashboard=MjE5MA==>

In the previous section Data Analytics has been reported. In addition, other Simulations can come from:

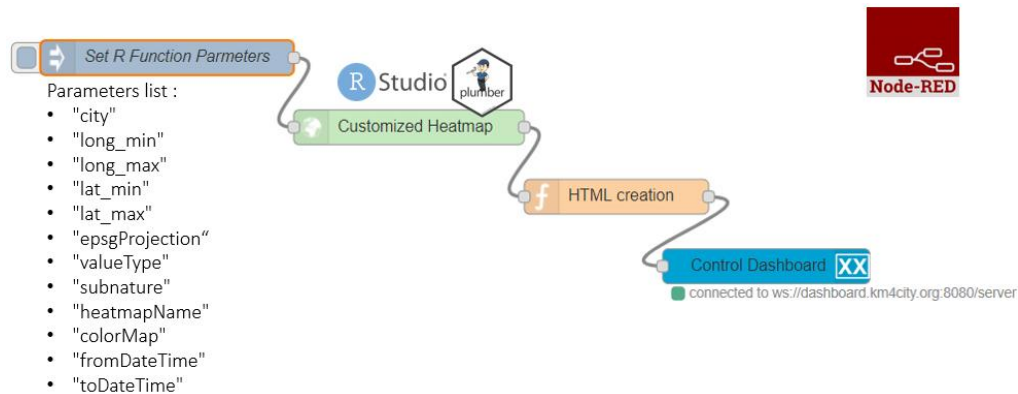
- Traffic Flow reconstruction with conditions and restrictions,
- Predictions with conditions and restrictions: traffic, parking, NOx, people flow, ...
- Analysis of the impact of changes on Public Transportation via DORAM tool,
- Routing with changed conditions and restrictions in real-time,
- Match from offer and demand of public transportation
- simulations via **SUMO** tool, etc.

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

3.9.3 - 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.



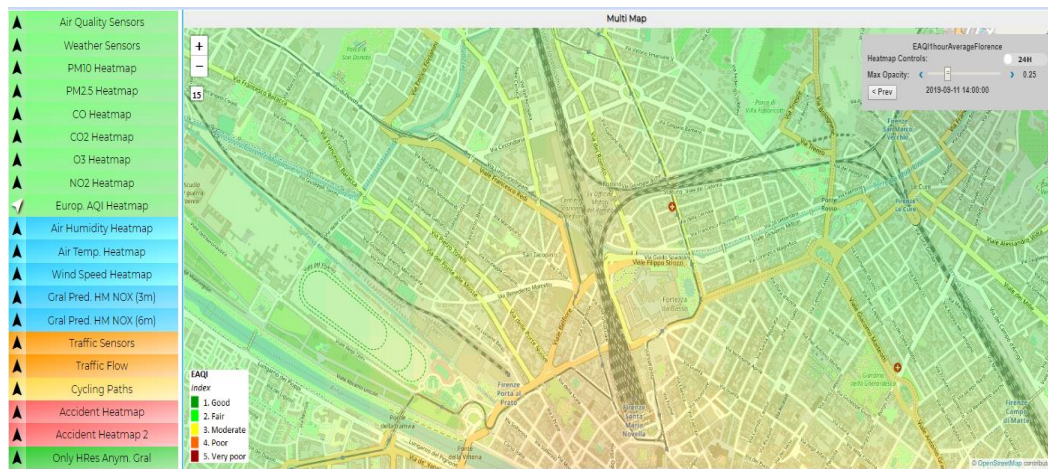
IoT App/Node-RED workflow for automatization of heatmap production.

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 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.



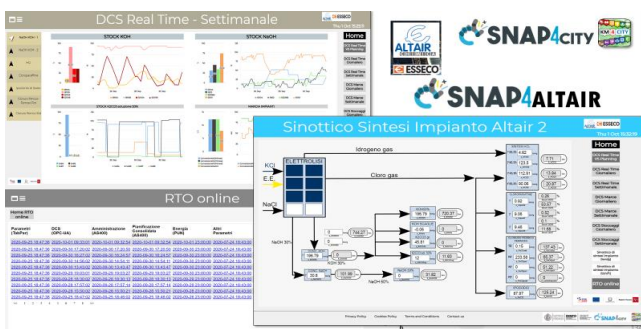
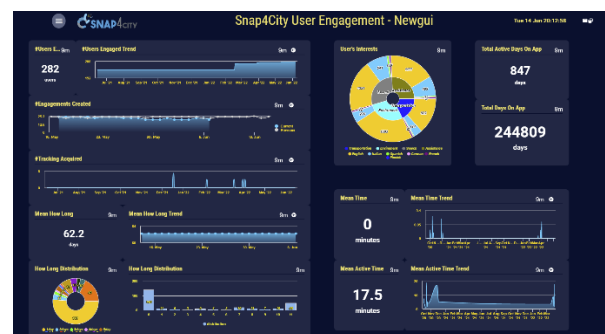
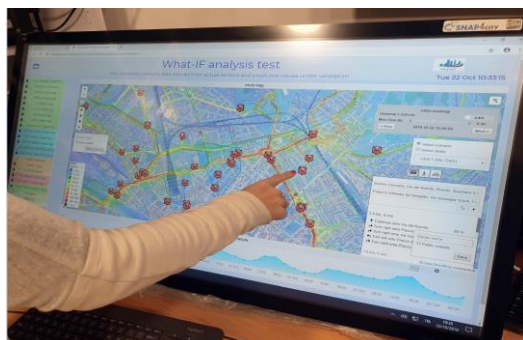
In addition, to automatize the process of the creation and use of heatmaps, there are also the following services:

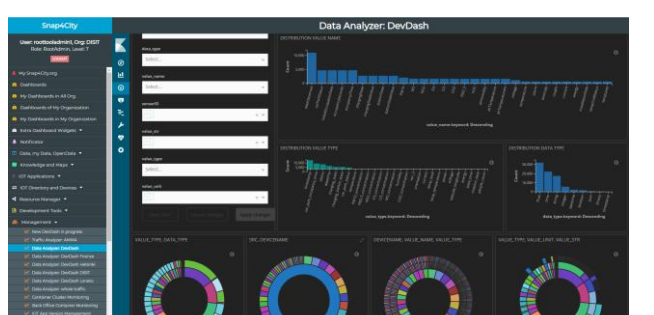
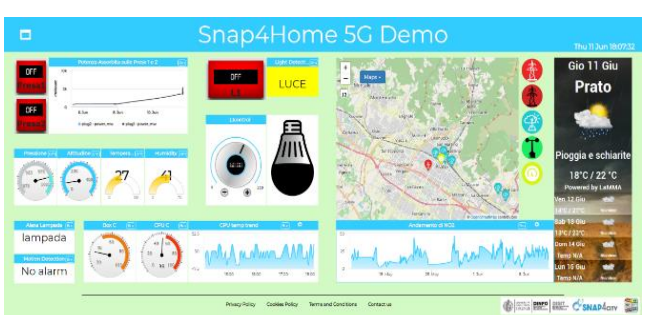
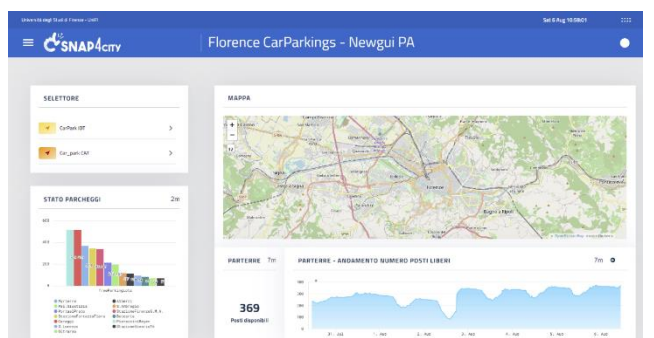
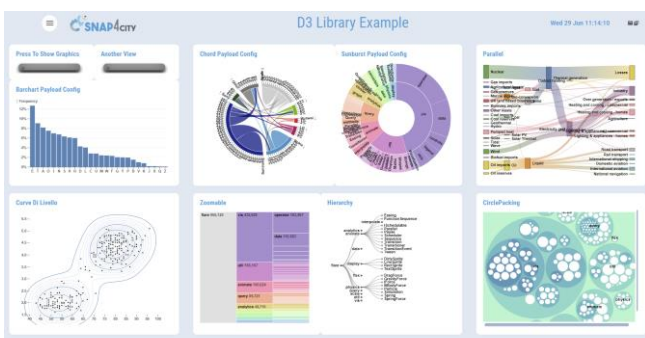
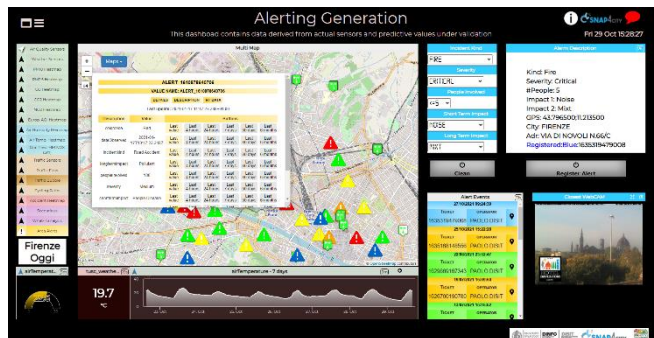
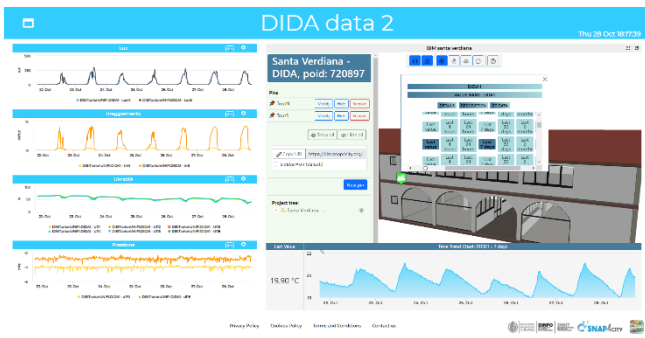
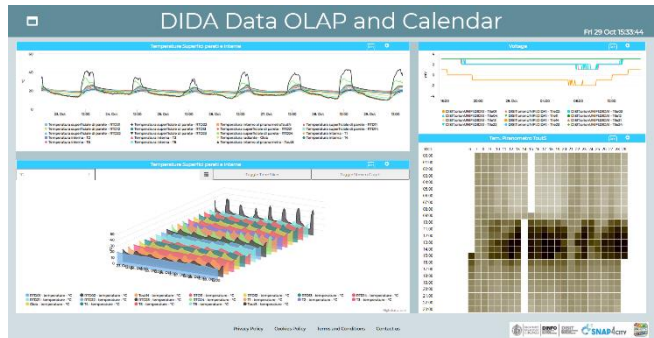
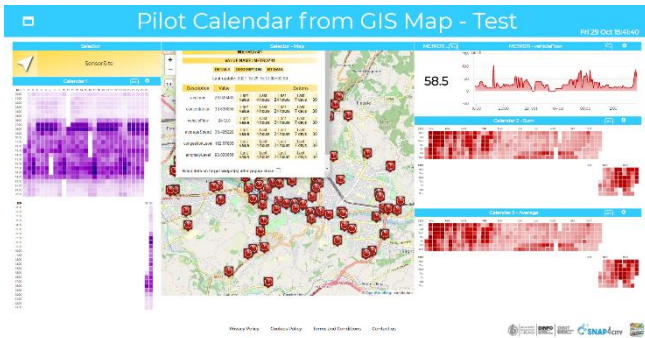
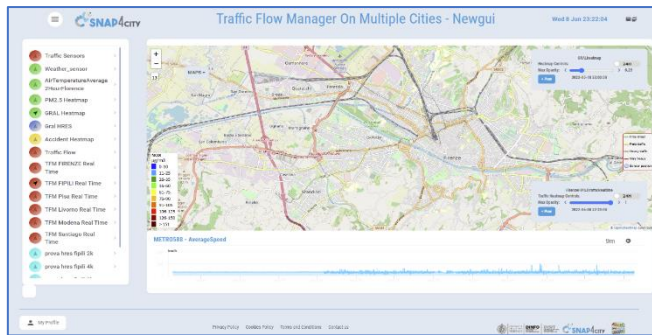
- **Piking the Heatmap value** in any GPS point disregarding the scattered distribution of sensors, exploiting interpolations (Akima, IDW, etc.) with a specific Smart City API and MicroService.
- **Alerting Assessment** has been implemented by using the **Snap4City Engager** tool that, every time a GPS 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.

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

A **dashboard is substantially a Decision Support System** tool, since it may provide evidence of normal and critical conditions, and in some cases may offer solutions, if well designed and connected with Data Analytics tools. 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, 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** of any kind which are interactive data & graphics applications including 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>*

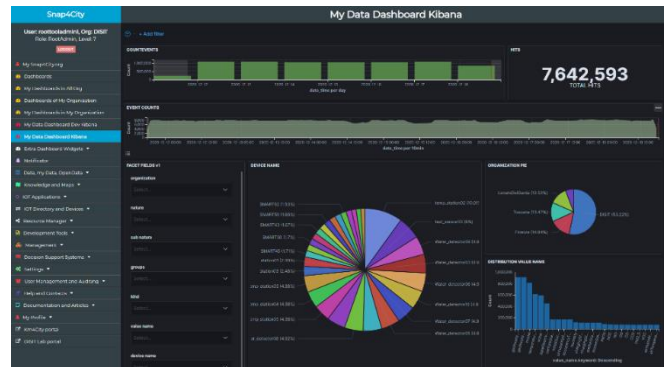




3.10.0 - Snap4City Dashboard vs Kibana, Grafana Dashboards

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
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, 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

3.10.1 - Snap4City Dashboard Builder

Snap4City **Dashboard Builder** enables users of any 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/das/> [Industry4.0-2020], [Dashboards2019], [DashboardProduction2020], [ChemicalPlant2021].

Dashboard Builder allows the creation of Dashboards:

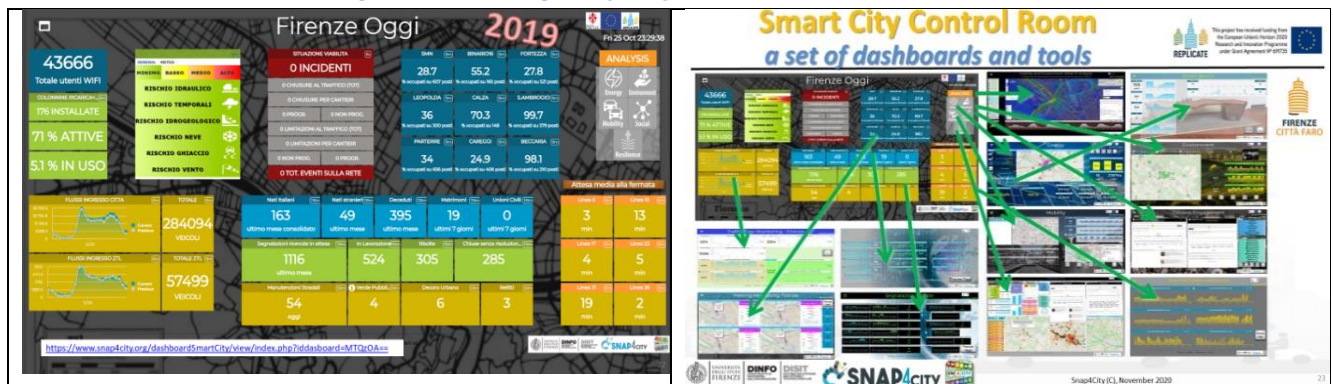
- **manually** by using a large set of graphic and integrated **widgets** that can be customized in deep for: data source, size, colors, shape, font, strings UTF-8 also Arabic, dates, left/right time series, etc.; A list of them is reported in the following. You can edit them manually to change any tiny 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**, which can be customized and enlarged by following a simple tutorial. The passage from one style to another is immediate, and the default style 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 (Dario Nardella) and the second-level dashboards. The third and fourth levels are present as well. <https://www.snap4city.org/525>

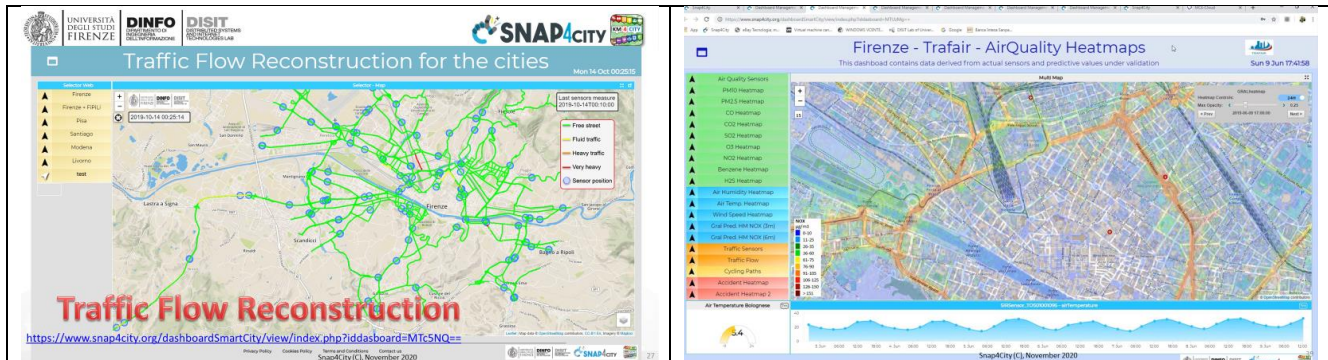
Control room with control video wall: <https://www.snap4city.org/621>

Florence Control Room: <https://www.snap4city.org/531>



Dashboards 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, and **What-IF analysis**.

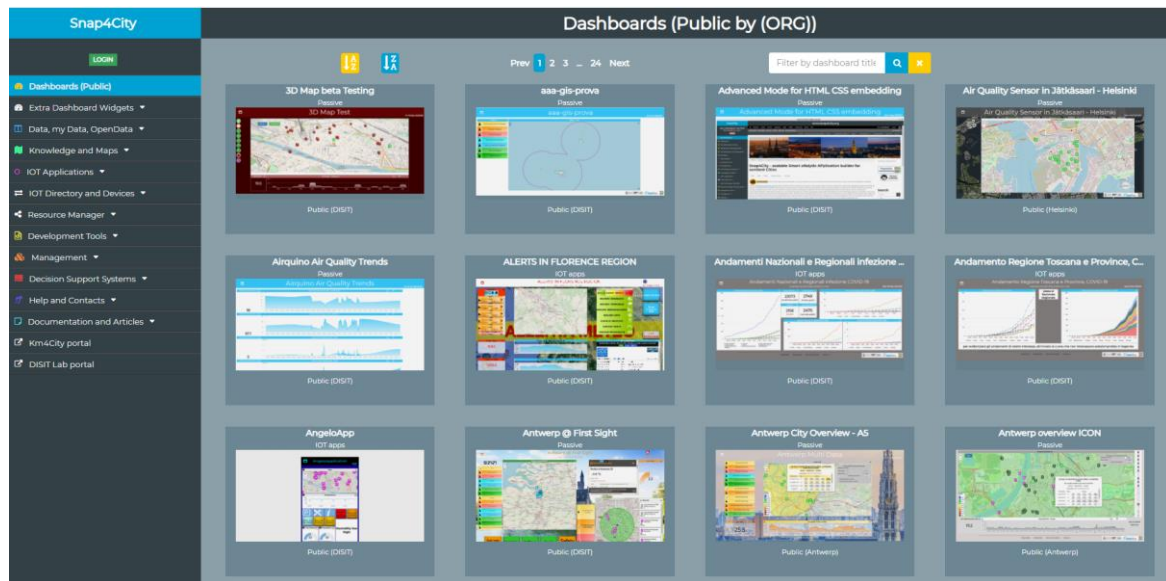


Snap4City Dashboard system is capable to:

- show dashboards 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 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;
- 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 dashboard specific menus;
- Connect to each dashboard a dedicated chat room for discussing problems and events.

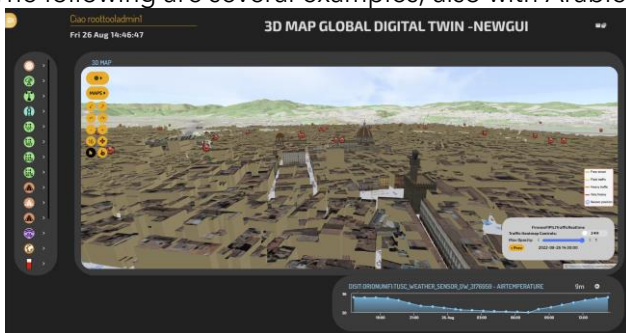
The Snap4City Dashboard system is used in several cities and projects and can be accessed from:

- Scenarios: <https://www.snap4city.org/4>
- **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>

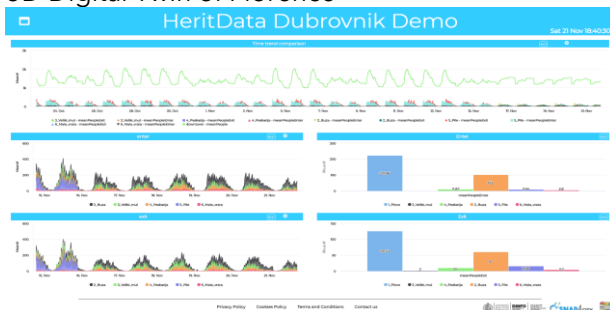


3.10.2 - Examples of Snap4City Dashboards

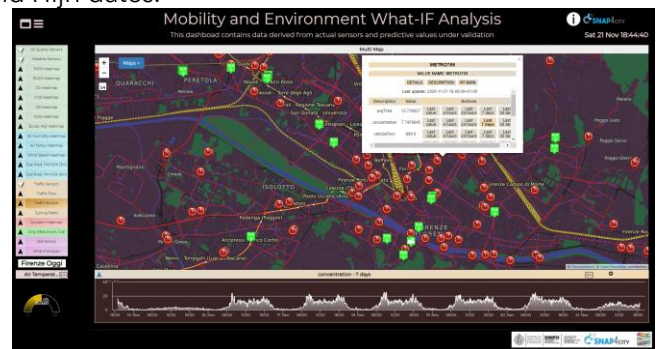
The following are several examples, also with Arabic and Hijri dates.



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



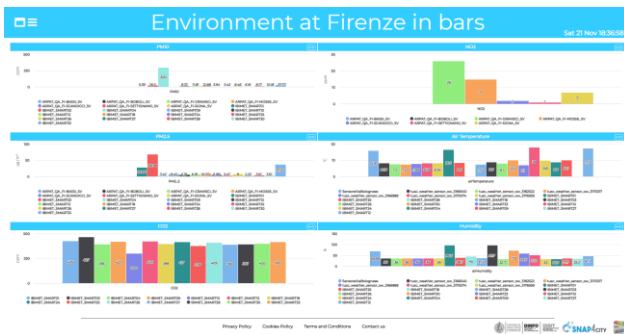
<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=Mjg5Mw==>
Twitter Vigilance on 6 European Cities



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



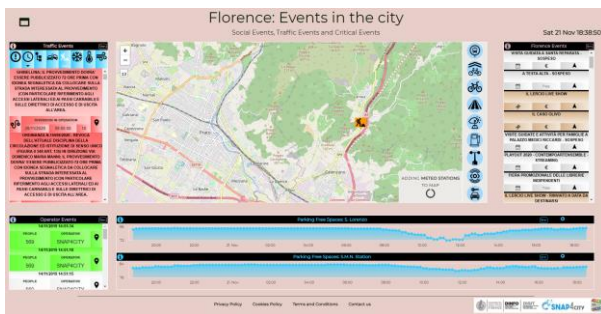
<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MTUzMg==>
Environmental data, NOX predictions and several heat maps



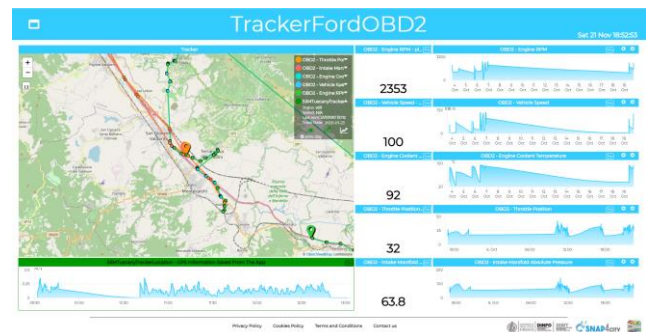
<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MjQwNw==>
Environmental data



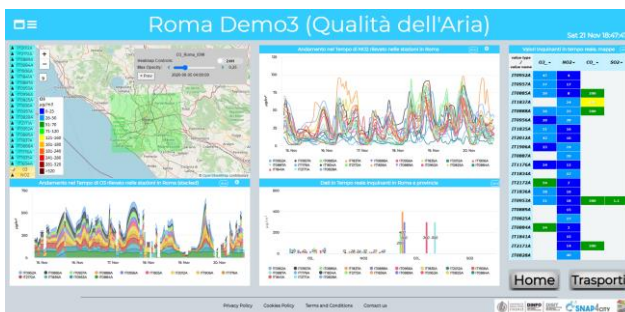
<https://www.snap4city.org/dashboardSmartCity/view/Gea.php?iddashboard=MzQ4OQ==>
Using D3 graphic library on Snap4City Dashboards



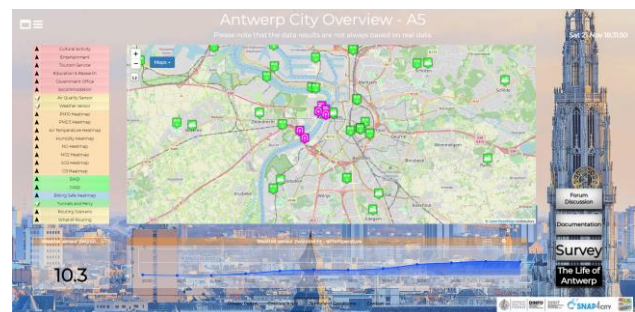
Events and Ordinances in Florence



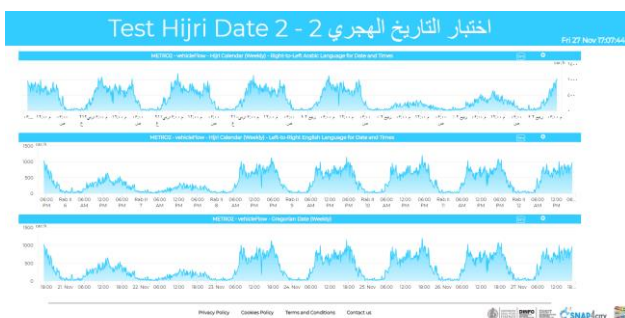
Tracking moving sensors and vehicles



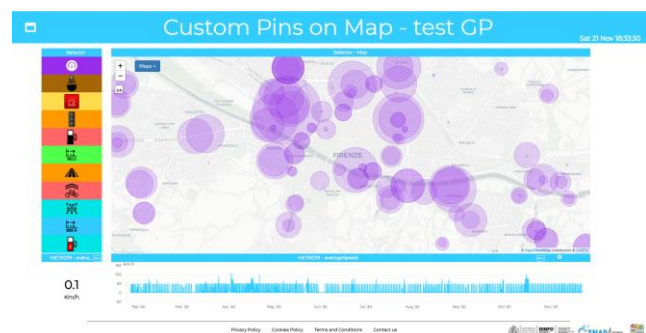
<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MjcyNg==>, Rome



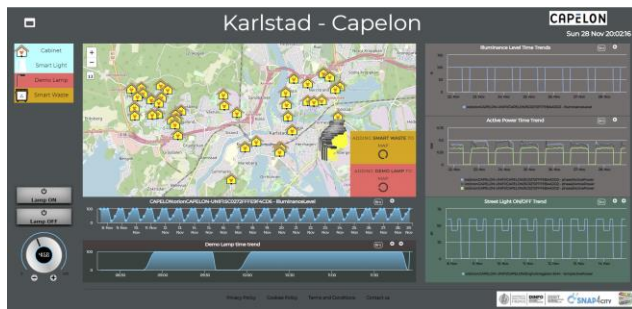
<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MTQwNw==> Antwerp dashboard



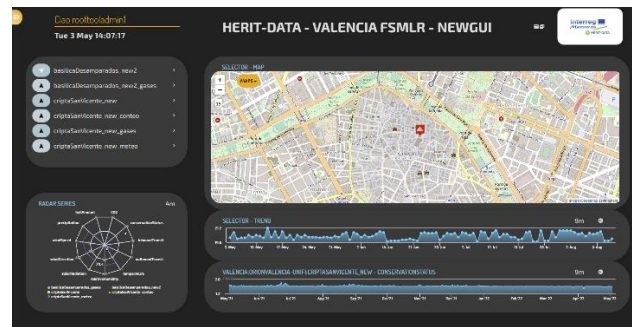
<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MzAyMw==>



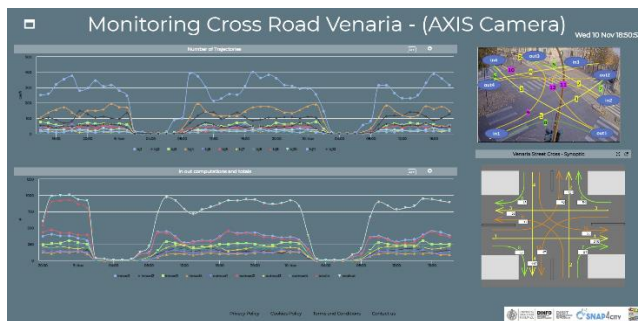
<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=Mjk5MA==> Example of dynamic PIN monitoring real services in Florence



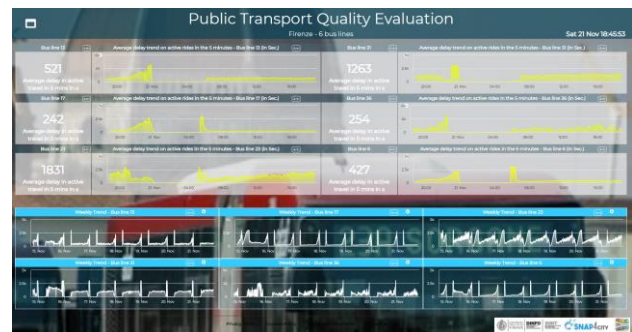
Smart Light in Karlstad, Sweden



Valencia, Spain



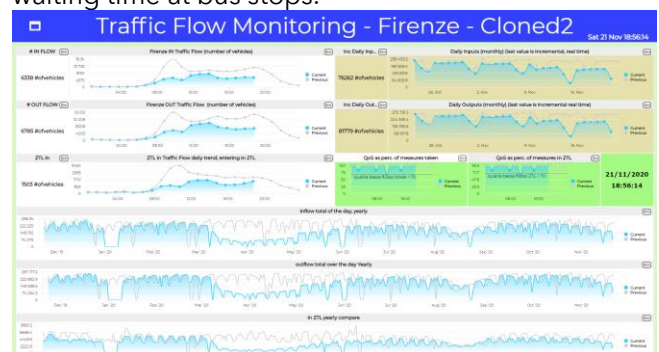
Venaria Reale, Torino, Italia



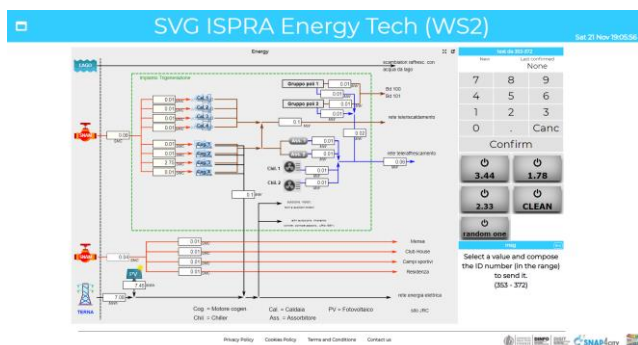
Monitoring of Public Transportation, average waiting time at bus stops.



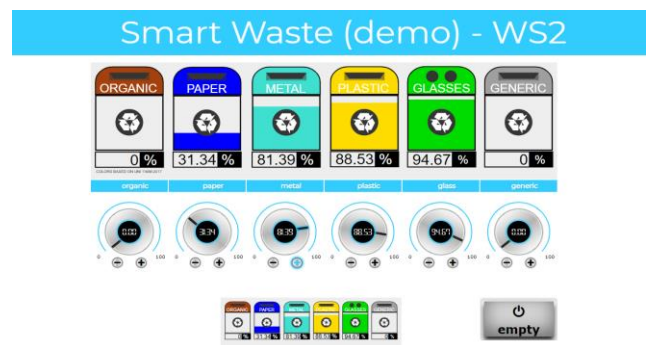
<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MTc5NQ==>
Traffic flow reconstruction in several cities



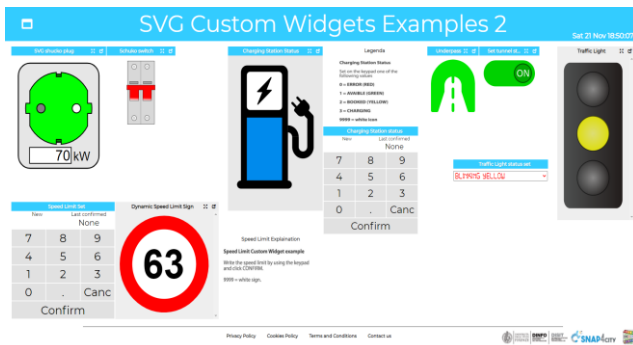
<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MjY1MQ==>
Traffic flow monitoring, in/out flows in Florence



Synoptic, energy consumption



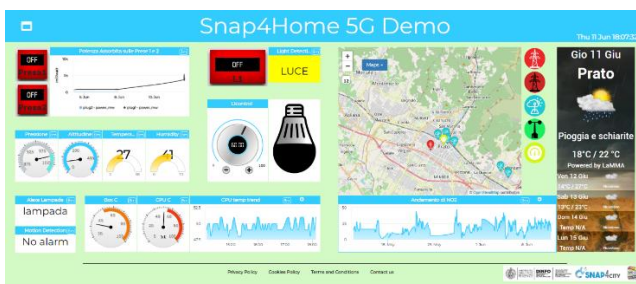
Synoptic for smart waste



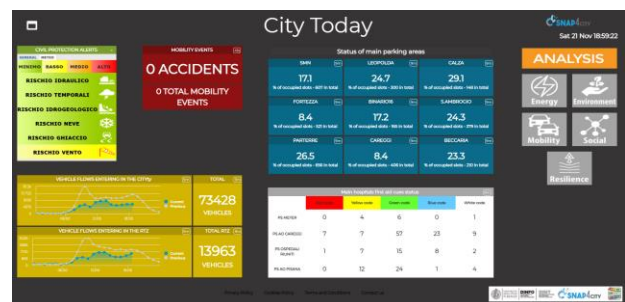
Synoptic SVG custom widgets



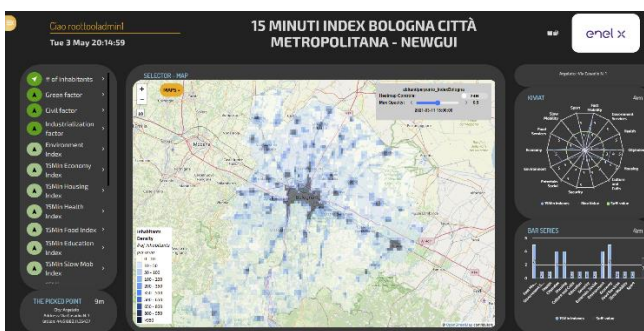
<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjY4MQ==>



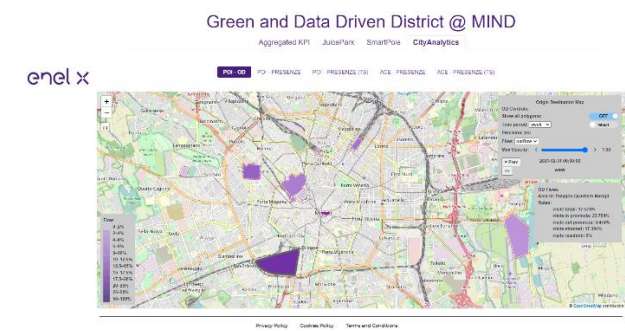
Smart@Home 5G dashboard



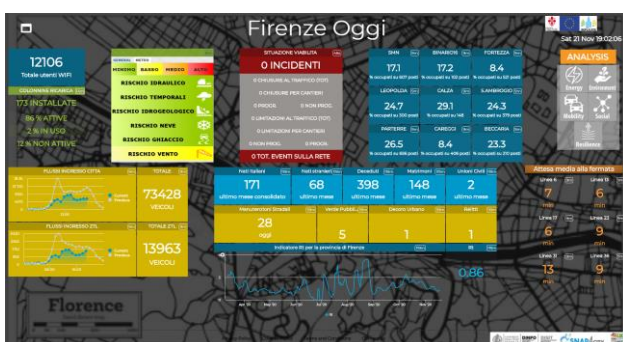
The today's City of Florence



15 Min City Index for Bologna, Italy



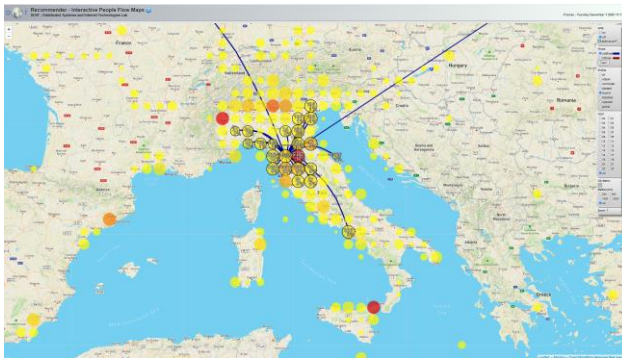
OD Matrics, city Analytics in Milano for ENEL-X



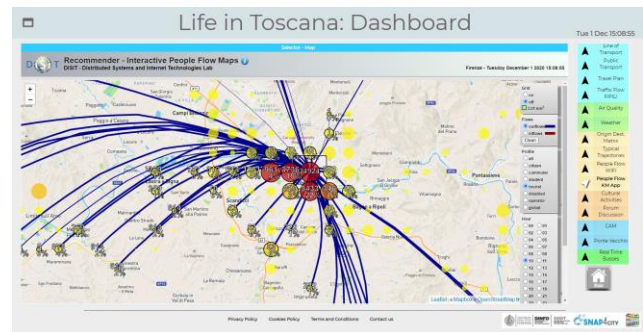
Florence Metro City Control Room main dashboard



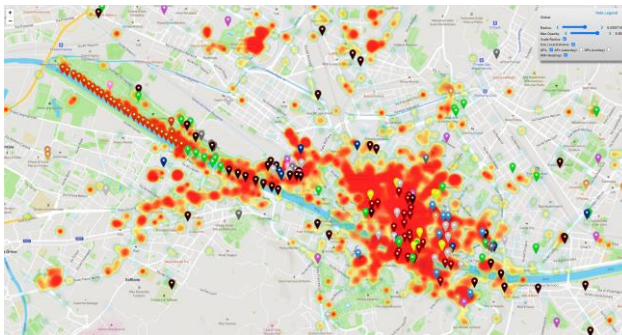
Smart parking in Lonato del Garda



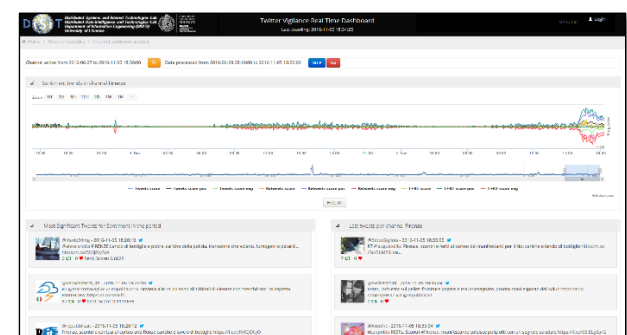
National level OD map



<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MTc3NA==>



Analysis of Wi-Fi Data



Twitter data, volume and sentiment

Some of the above dashboards are private and thus cannot be freely accessed. For these reasons some of the links are also missing, instead of providing links that could bring you to land on the login splash.

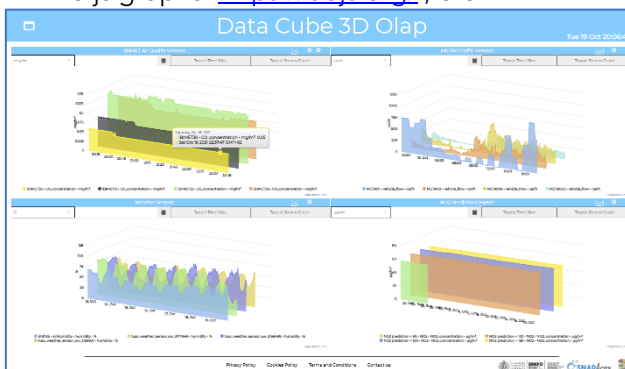
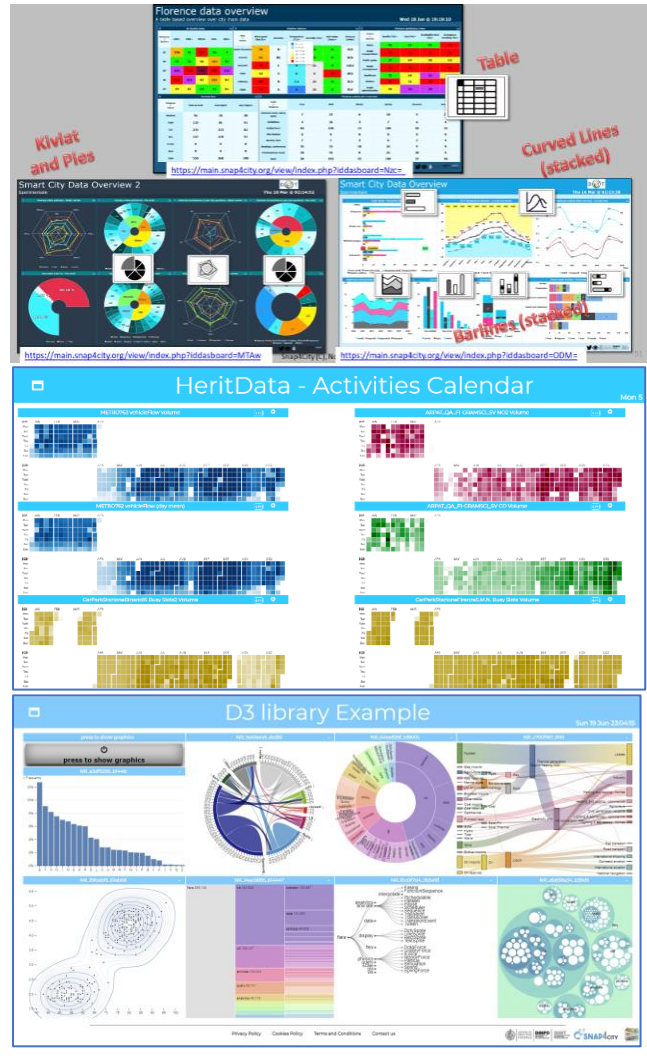
3.10.3 - List of Main Dashboard Widgets for Dashboard Builder

Snap4City Dashboard Builder provides a very large library of widgets that can be used to create Dashboards. A number of them are visible in the above-reported examples. A large number of Widgets in the library and the Wizard reduce the time required to create contextual dashboards. This is a high value because Dashboards are frequently requested when needed, typically *today for yesterday*.

Snap4City Dashboard Widgets can be classified into:

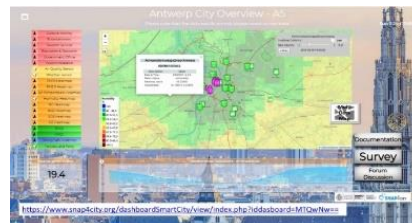
Time Series and real-time data (see also custom widgets for the real-time rendering of data via animation, dynamic PINs on a map, etc.)

- Single content, any value from numbers, strings, to HTML
- Time trend, Typical time trend,
- 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
- Heatmaps: calibrated and gaussian, animations
- GIS information, Polylines, shapes
- Routing, routes, traffic flows,
- IoT Devices, Sensors, and real-time data,
- Definition of scenarios,
- 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.

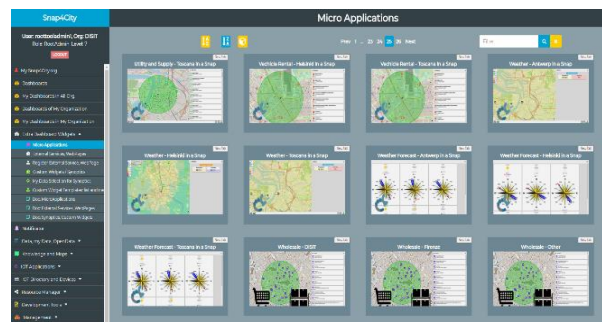
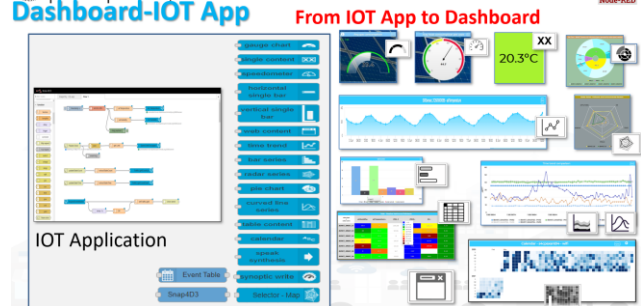
External Services

- <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.



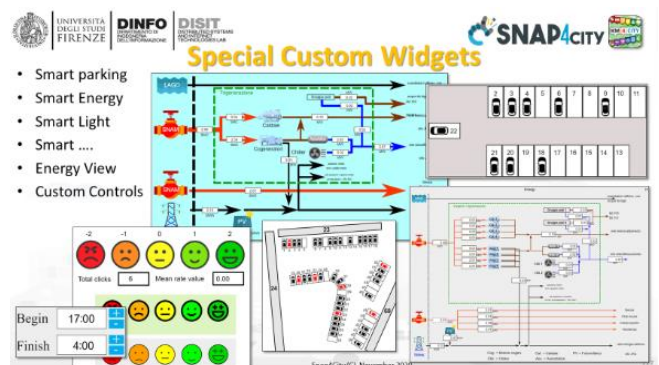
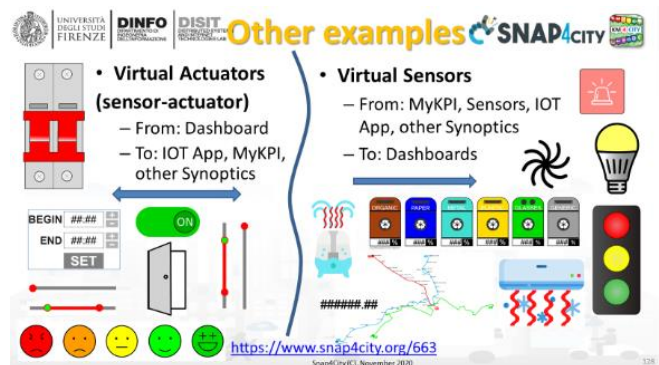
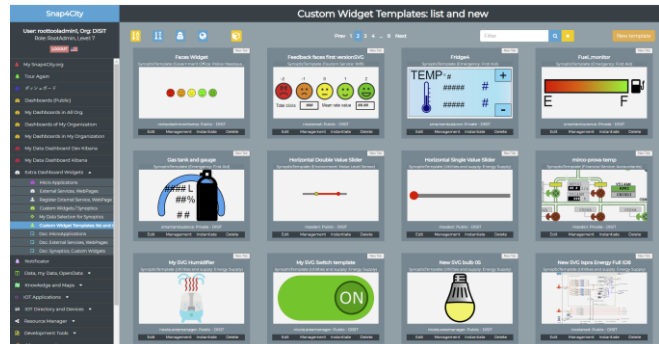
Special widgets:

- 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.

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
- 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
- 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 8 different styles are made accessible 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>

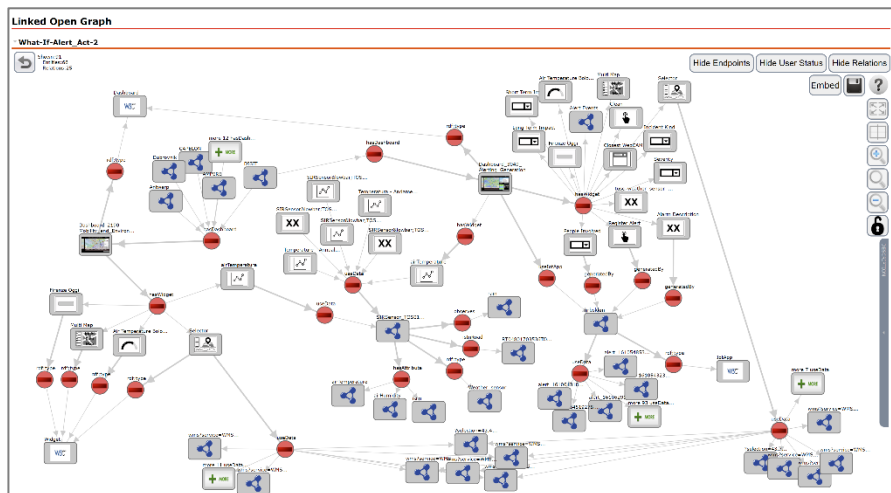
The Theme/style can be changed and customized to create new themes by following the instructions on page <https://www.snap4city.org/793>

The Selection of the style/theme can be performed directly 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 fundamental tools for decision makers 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: (i) data sources in the big data view, and (ii) computing processes. Thus the work of ICT 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.

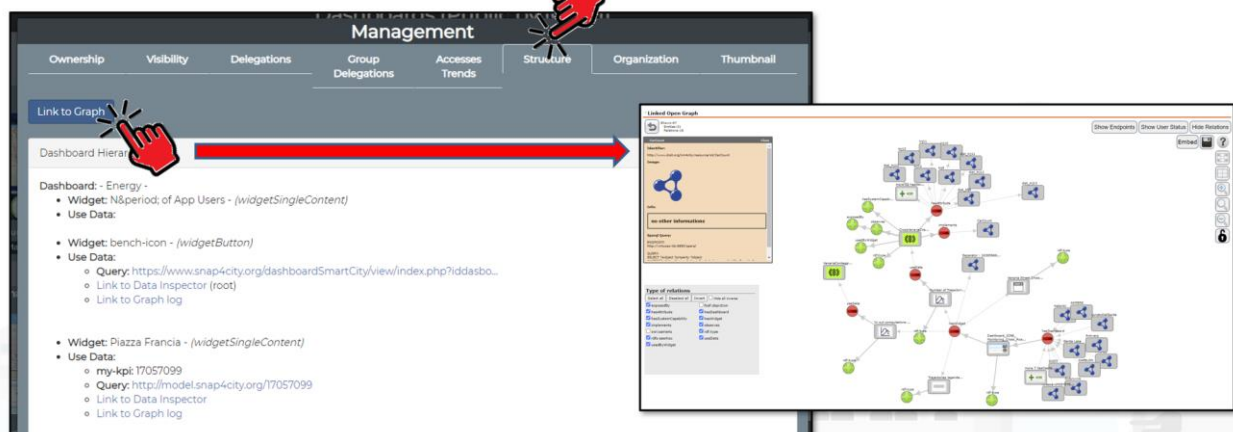
Snap4City developed a unified knowledge model, UKM, integrating data, processes, and users. Each developer can access the graph of its Snap4City Smart Application via any of their Dashboard accessing a graph as represented in the following figure in which the entities involved can be browsed and inspected. 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 provided. 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/sys/>

For All Dashboard owners: Graph and Structure

- Go on Dashboard Management

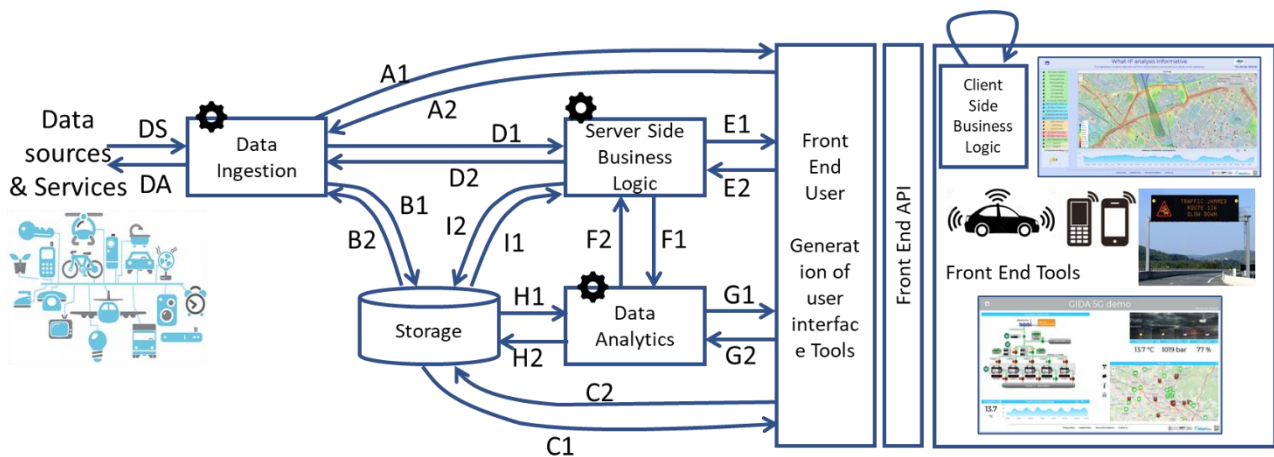


3.10.6 - Snap4City Smart Application, Business Intelligence Tools

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

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.,

In the following figure a conceptual architecture is presented in which the most relevant data flows among the main components are highlighted. Data 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, and the same complexity is present in the Business Logic, Data Analytics and Front-End, and thus also the Storage area may present multiple storages. Identify management and access control is diffused in all the communications among all areas and tools.



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 solutions. In those cases, the IoT data services managed data as Data Sensors (DS) and Data Actuators (DA), to store them in some Storage via B1 (may be indexed and retrieved), and rendered them asynchronously via C1. Data Ingestion / transformation is performed by using IoT App. The arrival of IoT/IoE devices constrained to be more reactive (data driven, managing real-time data) in connecting DS/DA and data services, thus producing the needs of having A1 connections, possibly data driven. In that context, Data Ingestion tools may be implemented by IoT Brokers to manage IoT Devices via bidirectional protocols (C2 and B2). At the same time, A2/A1 connection allows sending data and events from the User Interface to IoT Brokers and Devices as signals/messages in event-driven.

IoT Edge solutions are located on the field and before the Data Ingestion area. They may include a number of elements reported in the Figure, while the substance do not change the general data and control flows. Typically, the IoT Edge includes the Data Ingestion, and may include some data transformation and exchange. In the case of Fog computing some Data Analytics 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.

Slightly more complex smart city applications may need to perform some computing on stored data which can be performed only asynchronously due to their complexity (e.g., periodic processes). This problem can be solved by activating Data Analytic processes and services, exploiting connections H1/2 to read historical data and write results back (for example for computing some machine learning, simulation, predictions, early warning, anomaly detection, heatmaps, origin destination matrices, KPI, etc.). The results can be made accessible on the user interface via C1, once saved in the Storage Area, or via I/E if some Business Logic arrangement is needed. Otherwise, via G1 the analytics results can be provided on the user interface on the

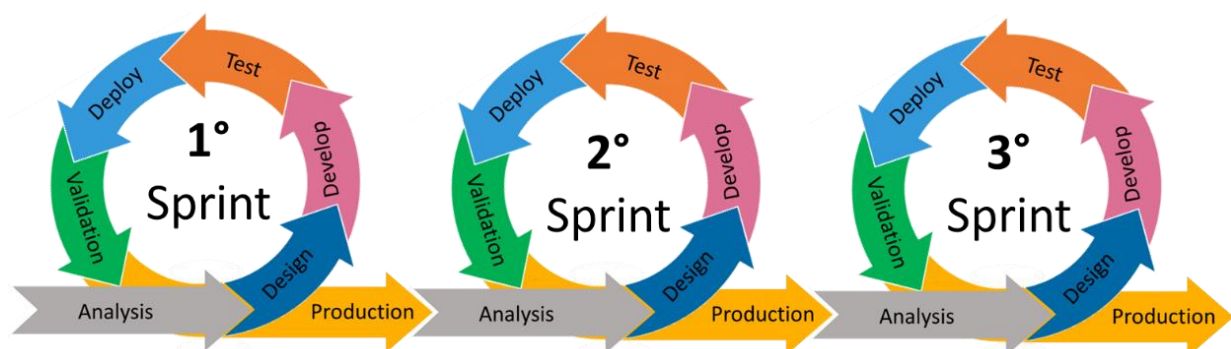
fly without saving values (assuming that the results are just a temporary exploration), that is the classic path of Business Intelligence tools.

Typical smart city applications and services need to compute predictions/simulations on the basis of historical and contextual data. The computation can be performed periodically or on demand. The results are usually saved on Storage, and they can be accessed by user interface devices (web and mobile App, Dashboard, digital signage, etc.) via communication and services, such as H1/2 connections. Examples of smart applications and services exploit computing tools are: smart irrigation (deciding the best irrigation time), smart parking publishing free parking slot predictions, optimizing routing for waste collection, multimodal routing for final users, and for computing origin destination matrices, heatmaps, anomaly detections as early warning, etc.

Recently, in the context of Smart City/IoT, a number of complex smart city applications are demanded by Decision Makers for city management and Control Rooms (with the aim of implementing workflow management, simulation and what-if analysis), developing complex workflows in short time, with frequent updates and modifications of data analytics parameters, data flow, work flow, and business logic of the user interface. To this aim, most of the smart city 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 flows (D1/2, E1/2), as in Figure. To this end, most of the development environments exploit traditional programming languages (such as AWS, MS Azure IoT, etc.), while in some other cases visual languages and MicroServices are used as in Snap4City, where also programming modules can be added. 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. The 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. Client Side Business Logic allows each user to see its own experience on Front-End without loading the server side of many contextual information for each user. Server Side Business Logic may be realized by taking into account the context of the user interaction or without. In the latter case, it can be used to share the user experience with many users for example in control room and decision support contexts.

Thus the implementation of smart solutions can be realized by using micro-services provided by the functional areas of Figure, and taking into account non-functional. The Snap4City framework enables 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 processes, data analytic, and users' tools at the same time.

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.



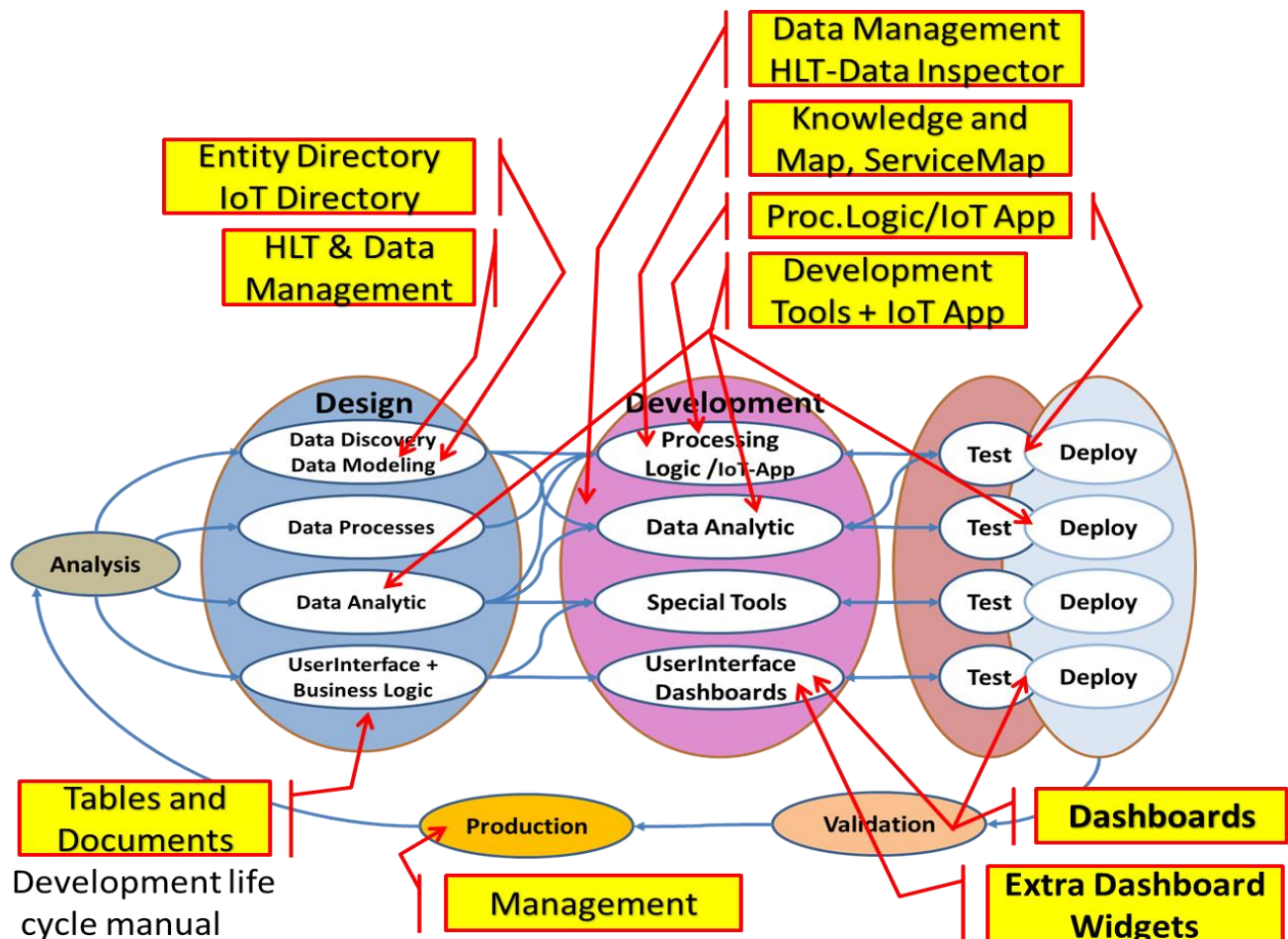
Agile life cycle for developing smart applications on Snap4City, 3 Sprints, not limited to...

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:
 - **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

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.)
- **Interoperability, establishing connections with other services for:**
 - Providing data: very simple call to external API from IoT APP
 - Exposing 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 read 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, suggestion, heatmaps, recommendations, decision support, routing, classification, detection, video processing, etc.

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 IoT Apps with Snap4City Dashboard Nodes
 - Both kind of Business Logic are 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. Example are:

- the click on some button on Dashboard to ask for a computation of some Data Analytics to be shown as a results 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 a 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.

3.11 – Snap4City Development Environments and Training

Snap4City.org is a training environment which you can learn on how to 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. You can replicate the same situation in your premise by using Snap4City open-source versions at your full disposal, license-free.

The list of Organizations and installations is accessible on the map 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, and managing mobile applications, etc. Thus, Snap4City.org is a multi-organization living lab.

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.



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:

- **Living Lab environment** for accessing to training information, video, and examples; discussing solutions, sharing experiences.
- **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.**
- **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 solution, but not limited to FIWARE solutions and protocols.
- **IoT Directory for IoT Network management and abstracting from Brokers, Devices.** Automated deployment of Orion Brokers is authorized only to specific users.
- **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.**
- **Data Analytics** for development in various languages: RStudio, Python, also transformed in API services and MicroServices for IoT Apps. RStudio can call segments in Python, Java, etc.
- **External services as web pages to be integrated into Dashboards;**
- **External services accessible via API, to be integrated as API, and MicroServices in IoT Apps;**
- **Web and Mobile Apps: with standard SDK for Apache Cordova to be installed on local computer**
- **Web Scraping,** and Crawling also transformed into API services and MicroServices for IoT App.
- **Workflows on IoT Apps and OpenMaint/BPM, also integrated.**
- **HeatMap Manager and server:** for automated production of heatmaps and their distribution via other GIS: ArcGIS for example.

- **Report generator:** at levels of users, devices, dashboards, and solutions, for several Organizations, the management is provided only to specific users for each organization.
- **Data Table Loader and POI Loader,** for several Organizations, the management is provided only to specific users for each organization.
- **BIM models and data, created and edited in standard IFC and then integrated into an online BIM Server.**
- **Traffic Flow Manager, and Traffic Flow Reconstruction in** multiple but limited geo areas.
- **OD Manager, and server** in multiple but limited geo areas.
- **TV Cam Manager** for managing view streams from RTSP, ONVIF via Kurento and Turn, and reporting them on the dashboard as WebRTC.
- **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.
- **Routing and MultiModal Routing** based on Snap4City tools, in multiple but limited geo areas.
- **Heatmaps production in** multiple but limited geo areas.
- **Satellite data download and automated reformatting in heatmaps in** multiple but limited geo areas.
- **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]
- **Engager for Mobile Apps,** limited to projects only. [UserBehavior2020]
- **API accounting tools,** limited to projects only.
- **Converters from OSM to KB, Service Map, SSM2ORION,** Limited to projects.
- **Chat manager to discuss on Dashboard events online,** limited to projects.
- **OpenData Gate, CKAN,** limited to projects only.

Online training is accessible from <https://www.snap4city.org/577>

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

You can access the Training Course Slides in PDF and interactive formats. 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 of it 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> 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/sys/>

<https://www.snap4city.org/577>

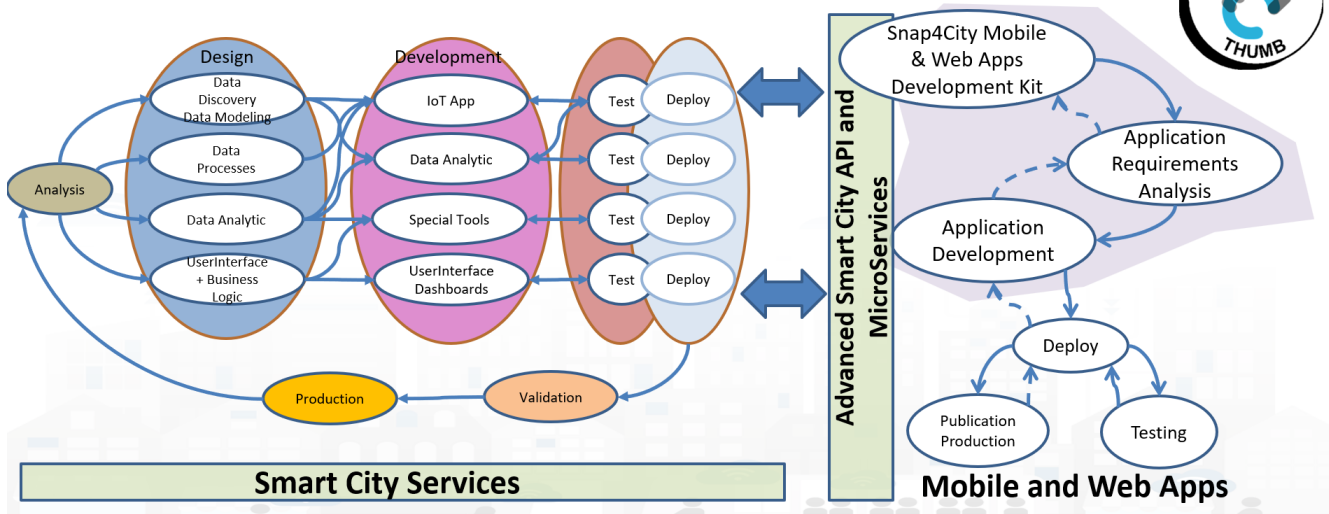


On Line Training Material (free of charge)

	1st part (*)	2nd part (*)	3rd part (*)	4th part (*)	5th part (*)	6th part (*)	7th part (*)
what	General	Dashboards	IOT App, IOT Network	Data Analytics	Data Ingestion processes	System and Deploy Install	Smart City API: Web & Mob. App
PDF							
Inter active							
Video1							
Video2							
Video3							
Video4				none		none	none
duration	2:55	3:16	3:41	2:00	2:48	2:35	1:47

Snap4City (C), November 2020

Develop Mobile & Web Applications Exploiting Snap4City Smart City Services



The **course structure** is detailed on page <https://www.snap4city.org/577> 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. **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.

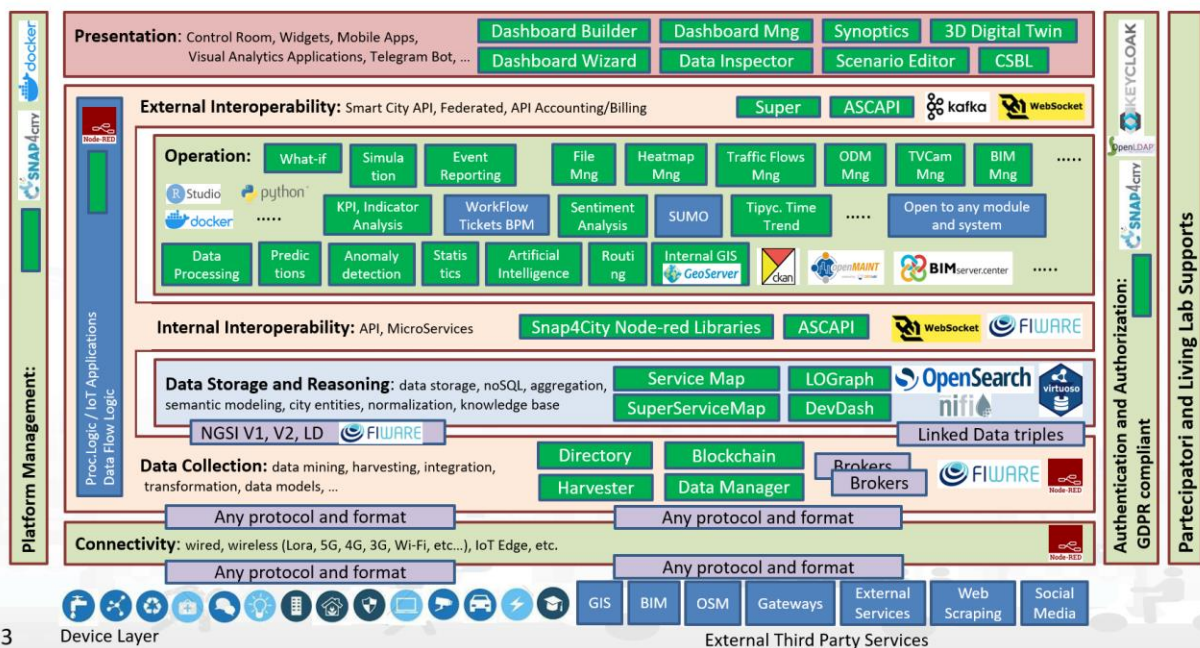
3.13 - Snap4City Access Control and GDPR Compliance

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, but 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/471> **Each VM is an appliance with several components that can be updated directly from GitHub/DISIT.** <https://github.com/disit> The update of the Docker based solutions is more complex but feasible.



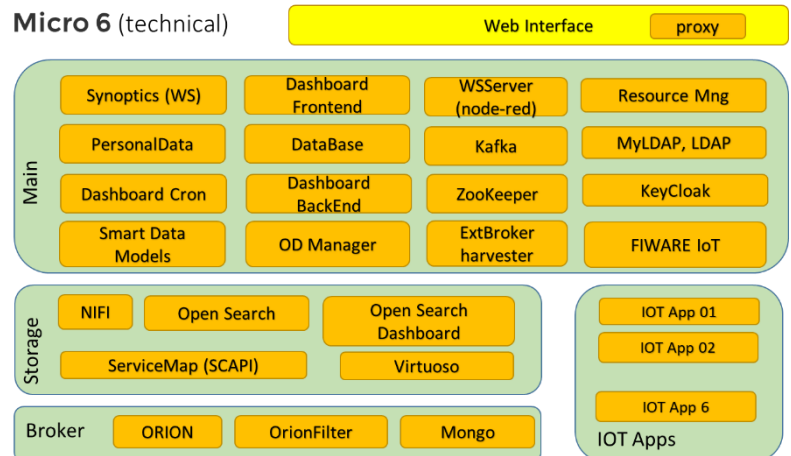
All the components in Green are Snap4City Modules which are familiar to you if you have read all the pages of this document before this one. See <https://www.snap4city.org/661> get the list of the Snap4City installations that are known by us. Snap4City installations have to use the corresponding logo 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/471>:

- VM-based version is the so-called MAIN version 1.6: <https://www.snap4city.org/471>
 - See for installation <https://www.snap4city.org/792>
 - and: <https://www.snap4city.org/487>
- Docker Configuration tool documentation:
 - <https://www.snap4city.org/738>
- 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

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:

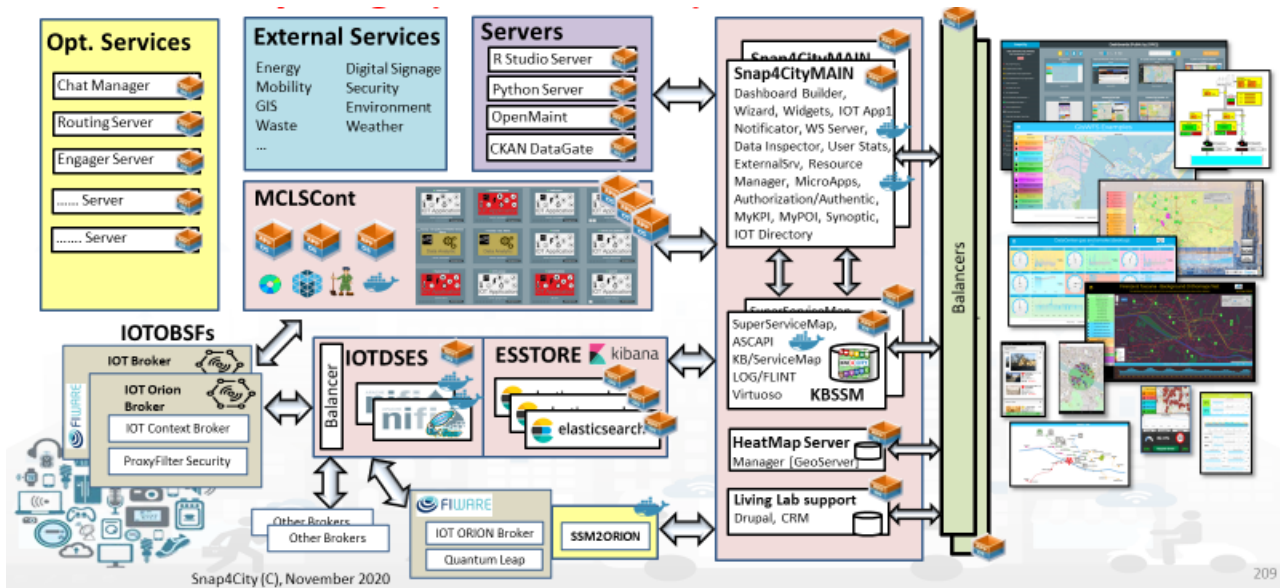
- 24 Gbyte of RAM, more than 50-200 Gbyte of HD (this depends on the data size you would like to have, we suggest 500 GByte in thin provisioning), and 16 cores or virtual cores, faster than 2.1GHz each. A larger size may be needed if you (i) get more IoT Apps (Micro X means the X would be the number 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 area 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

DataCity-Large which is reported in the following figure, and it is mentioned quite similarly into Tutorial **Part 6** <https://www.snap4city.org/download/video/course/sys/>



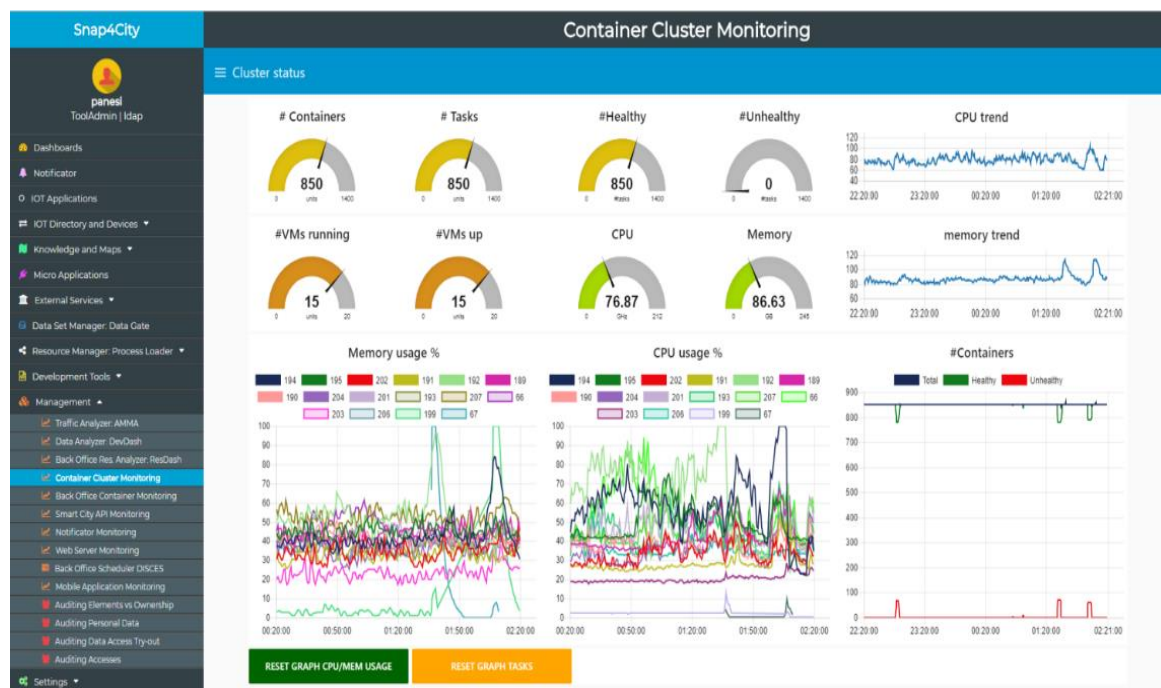
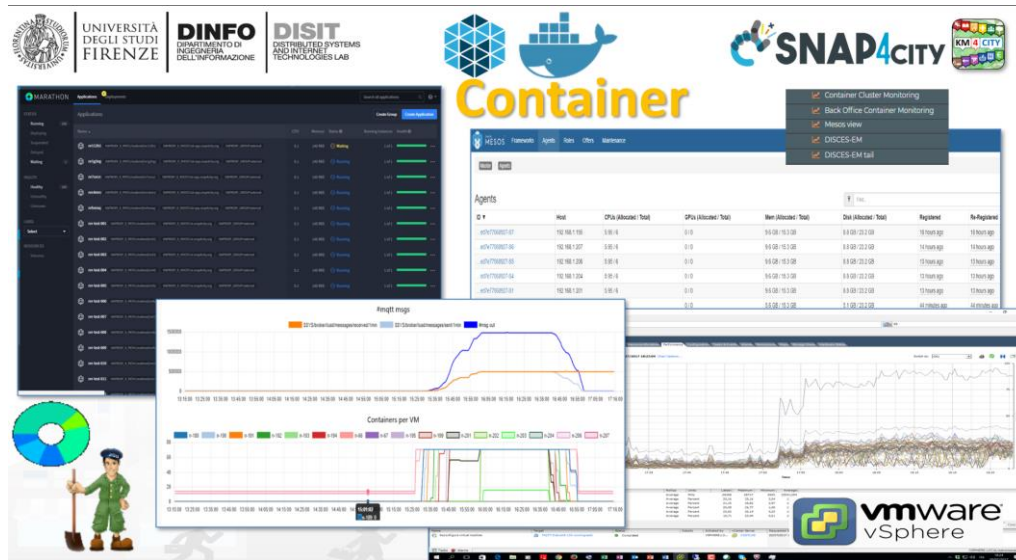
No proprietary components are needed to be installed into the Snap4City platform.

DCL: DataCity-Large configuration includes installations of:

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

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

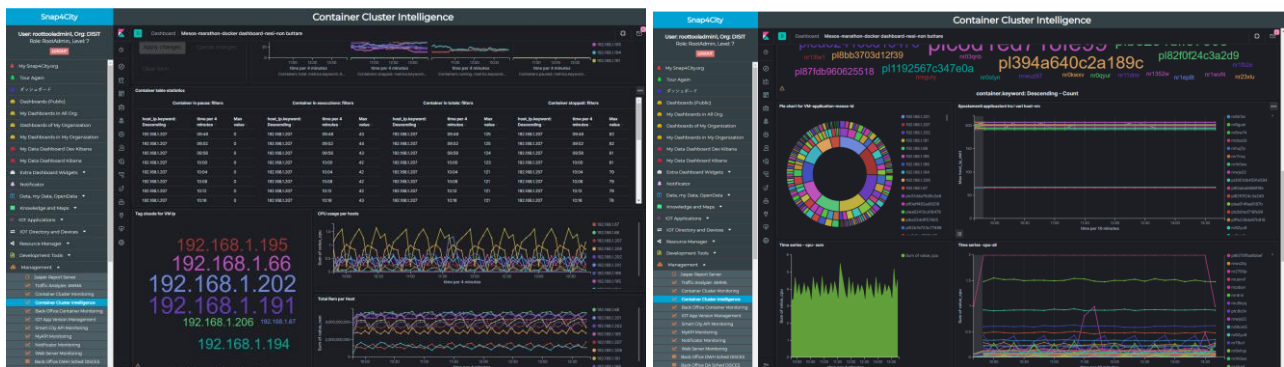
Example of MCLSCount IoT App and Docker cluster Monitoring. The DISCES-EM, which is the DISCES for Elastic Management, is the engine that continuously monitors the Mesos, Marathon, and Vmware resources to decide about the turn on/off of VM from the pool of **MCLSCount** VMs performing elastic management. 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



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:

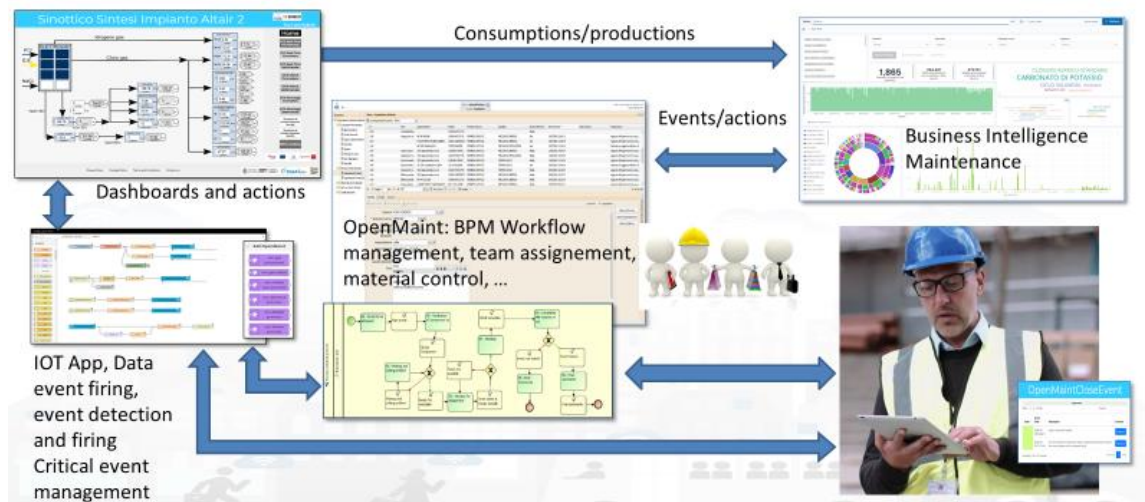
- **Mobile App staff:**
 - **Mobile App** "In a Snap" series, APP; based on [MobileDevKit2017]
 - **User profile manager** and suggestions VM; [UserBehavior2020]
 - **User Engager** system for mobile Apps, VM; [UserEngagement2017], [UserBehavior2020]
- **Data Analytics:**
 - **Traffic Flow predictions**, [TrafficPredictions2020]
 - **Traffic Flow reconstruction and simulation VM**; [TrafficFlowReconstruction2020]
 - **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]
 - **Social Media Analysis**, via Twitter Vigilance,
 - **Predicting Landslide events** [PredictingLandslideEvents],
 - **Single Camera traffic flow analysis of complex crossroads**,
 - **People Detection and counting via thermal cameras**,
 - **Wi-Fi user behaviour analysis**, DA, [UserBehaviour2017]
 - **COVID-19 data monitoring**, DA; [COVID-19]
 - **15 Min City Index**, [15MinCityIndex2021]
 - **CO2 computing from traffic flow and other data** , [EstimatingCO2Emissions]

- **3D production of Digital Twin Global, [GeneratingDigitalTwin]**
- **GTFS and Transmodel ingestion processes;**
- **Computing Trajectories, OD, Typical Trends (DA);**
- **Anomaly Detection, DA;**
- **SUMO Simulator VM;**
- **Features and tools:**
 - **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.
 - **DORAM tool for the analysis of demand of mobility vs offer of transportation;** Tools;
 - **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, Hyderabad.**
 - **DISCES Scheduler and cluster VM;**
 - **Resilience DS VM/Docker [Resilience2017], [Resilience2019].**
 - **Smart DS VM/Docker; [SmartDS]**
 - **Twitter Vigilance VM** in different forms. For early warning and moods assessment: English, Italian, France, and a few other languages; [TwitterVigilance2017]
 - **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.
 - **Large file sender in addition to FTP: <https://filesender.org/>, VM**
- **Cloud SW:**
 - **All Orchestrators and managers for the cloud are not included, such as Vmware vSphere, Vmware converter, etc.**
 - **VEEAM software for backup of VM is not included**
 - **HUB and Firewall software if any is not included**

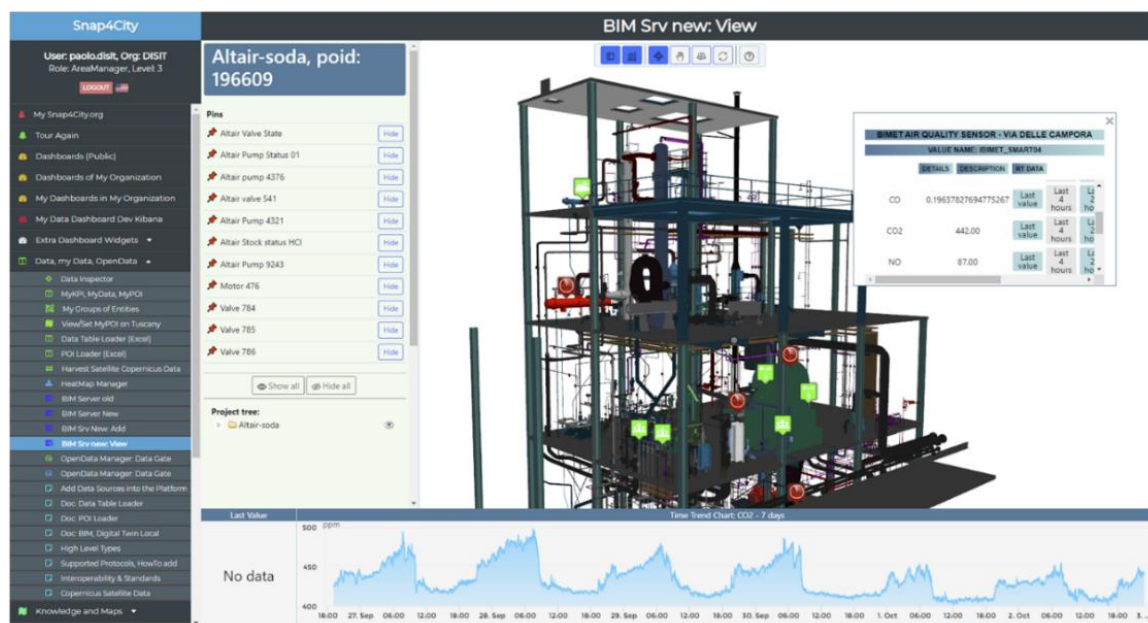
3.15 – Snap4City: Maintenance Management, BPM and BIM

Snap4City Maintenance Solution includes:

- **OpenMaint** open-source solution for property & facility management which is a BPM;
 - Inventory of industry assets (movable, logistics, equipment, etc.)
 - Tickets management for corrective maintenance.
 - User management with different levels of access.
 - BIM Server integrated with OpenMaint;
- **Snap4City OpenMaint Extension**
 - **Extended API** developed by Snap4City:
 - Create new tickets, Manage steps, workflow.
 - Collecting 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.

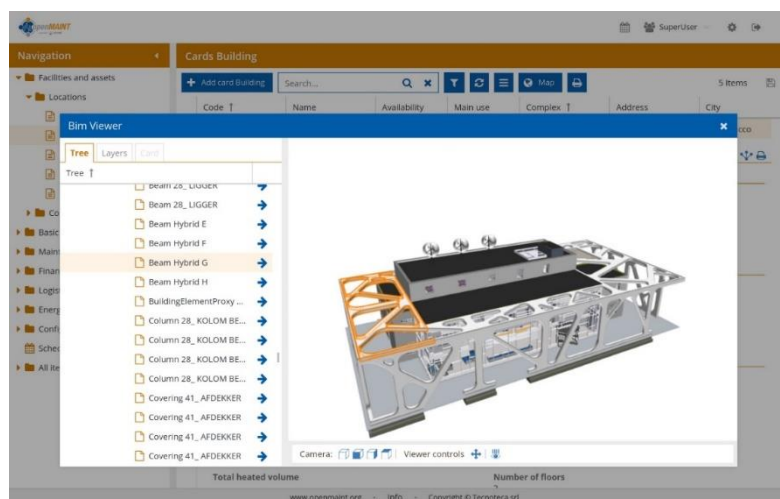
Functional areas of openMAINT includes:

- **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

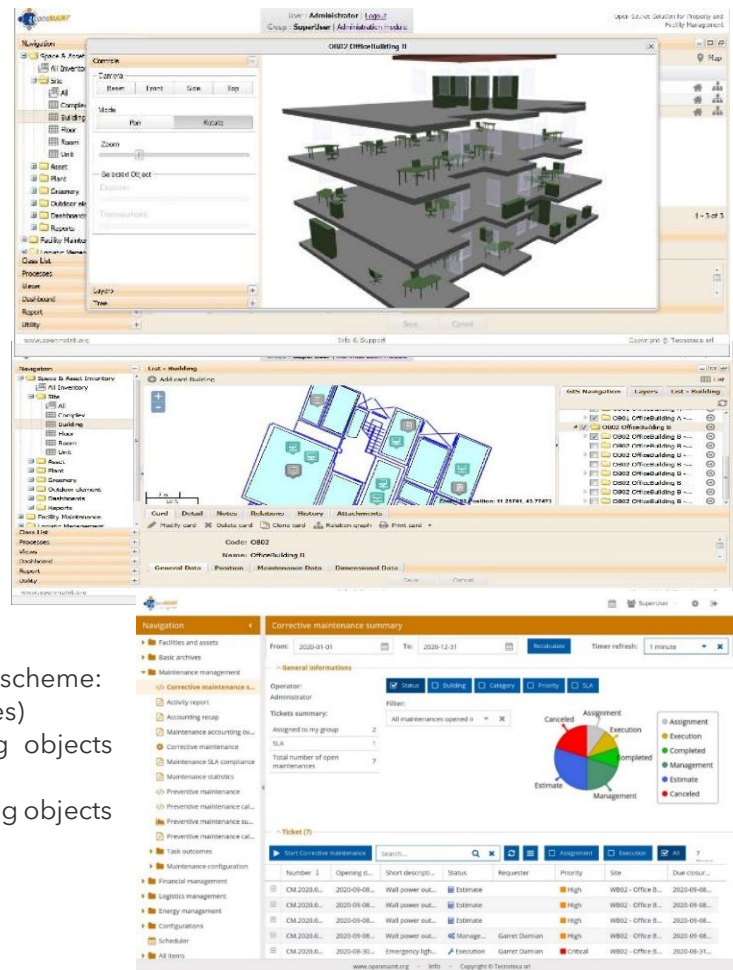
So that you do not need to install OpenMaint to have 3D representation on BIM for the Digital Twin of buildings. Snap4City uses OpenMaint only for collection and maintenance asset management in connection with Snap4City processes and facilities.

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.

3.15.2 - Snap4City Business Maintenance Intelligence tool

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

Snap4City has several user roles:

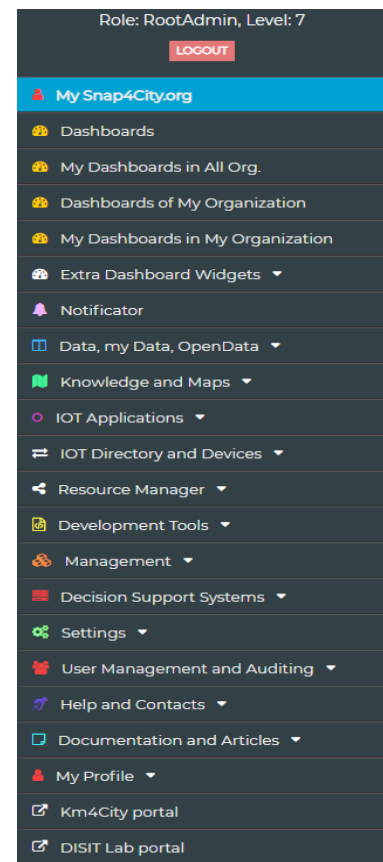
- **Users of any Role** have full control of their own resources: data, devices, dashboards, IoT App, etc., which may control according to GDPR rules, providing access, revoking, etc.
 - **Any new user starts at the lowest role: Manager.**
- **Users Roles:**
 - **RootAdmin:** The gods of the specific 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

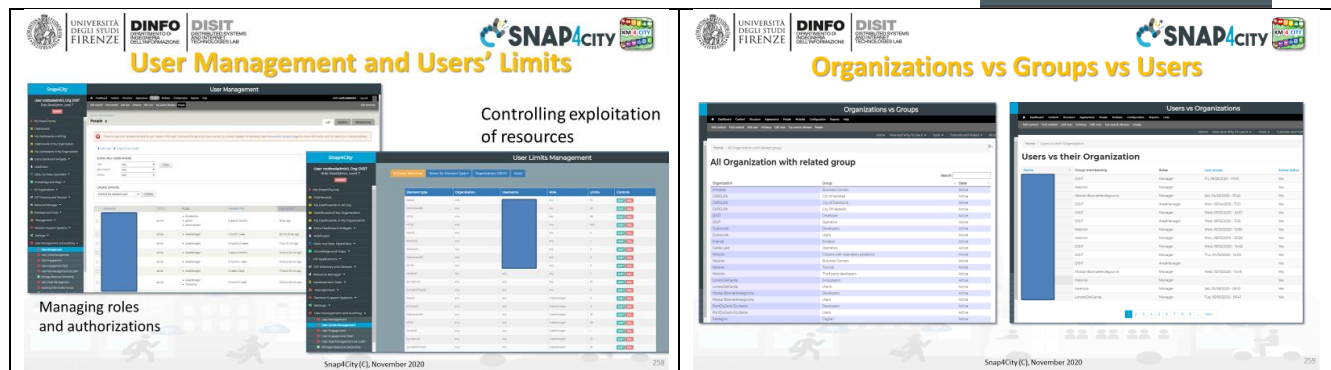
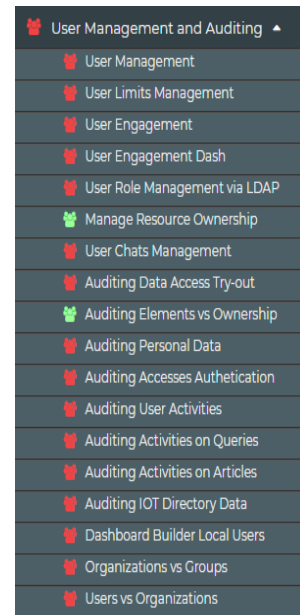
- Dashboards
- Extra Dashboard Widgets
- Notificator (only for solution on premise)
- Data, My Data, OpenData
- Knowledge and Maps
- IoT Applications
- IoT Directory and Devices
- Resource Manager
- Development Tools
- Management
- Decision Support Systems
- Super Settings
- User Management and Auditing
- 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 7 we have

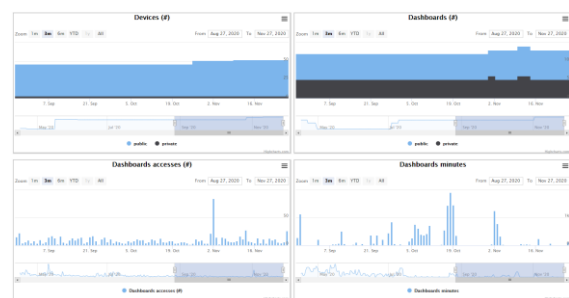
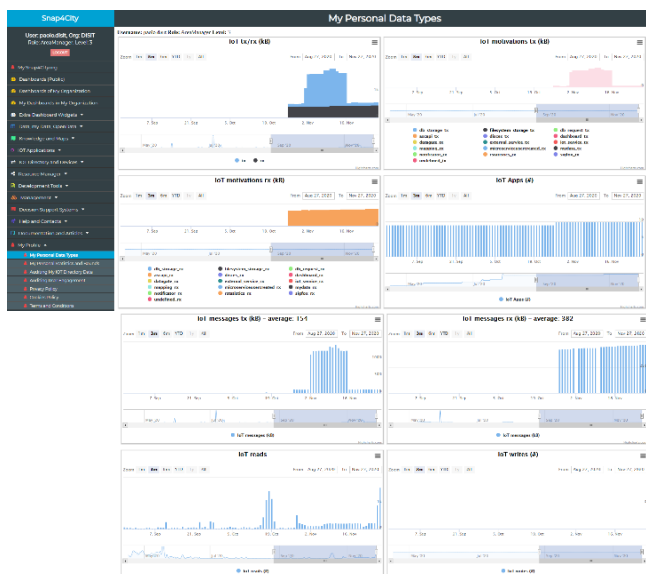
- **User Management** registration and mailing
 - LDAP connection role management
 - KeyCloak Authentication (OpenID Connect)
 - Management of user profile
- **Content management** for Organizations and Groups
 - Indexing of all content and search
 - Distribution of content and Video
 - Reports and views
 - 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

- **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

Management
Traffic Analyzer: AMMA
Data Analyzer: DevDash
Data Analyzer: DevDash Firenze
Data Analyzer: DevDash Helsinki
Data Analyzer: DevDash DISIT
Data Analyzer: DevDash Lonato
Data Analyzer: whole traffic
Container Cluster Monitoring
Back Office Container Monitoring
IOT App Version Management
Smart City API Monitoring
MyKPI Monitoring
Notificator Monitoring
Web Server Monitoring
Back Office DWH Sched DISCES
Back Office DA Sched DISCES
Back Office DISCES monitor
Mobile Application Monitoring
Mng Anonym. Photos Comments
Mng Photos Comments HelAnt
Mng Online Helps
Config ResDash
Mesos view
DISCES-EM
DISCES-EM tail
IOT App for Conf Clust Monitor

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**.

AMMA

- **Managing and Monitoring Data-Traffic in the BackOffice**
- Data Traffic Analyzer
 - Business intelligence
 - Faceted searches
 - Drill down on time
- Several different views and details on data traffic among the main entities in the platform:
 - IOT APP
 - Storage
 - Data sources,
 -

Event Logger

- There are API for Event Logger, REST API
 - They are automatically used by most of the Snap4City MicroServices
 - They log in standard Rsyslog API
 - <https://www.snap4city.org/56>
- The Logs regarding messages passed and usage are logged and accessed with the AMMA tools that is based on Elastic Search and Kibana.
 - Former version was made in Hbase and SOLR, and Banana
- Additional Logs events can be logged by using a dedicated MicroService in Node-RED, IOT Apps

Controlling the Open Search Storage and flow

DevDash

- monitoring and browsing data ingested into Elastic Search, via Kibana
- Faceted search
- Organization View
- Personal User View

Data Analyzer: DevDash

Monitoring the Resources and API Traffic, quality control

Monitoring Smart City API Usage

<http://www.disit.org/dashboardSmartCity/view/index.php?iddashboard=MTkw>
<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MTYONA==>

Monitoring Resources and Traffic

<http://www.disit.org/dashboardSmartCity/view/index.php?iddashboard=MTQ4>
<http://www.disit.org/dashboardSmartCity/view/index.php?iddashboard=MTQ5>

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

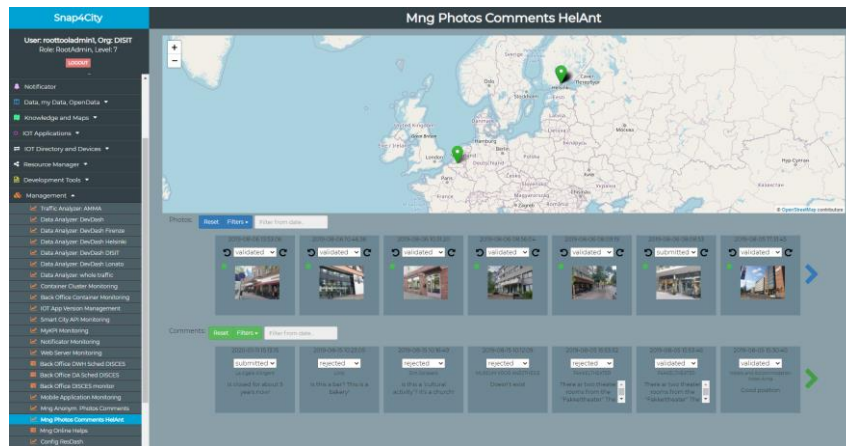
MyKPI

monitoring:

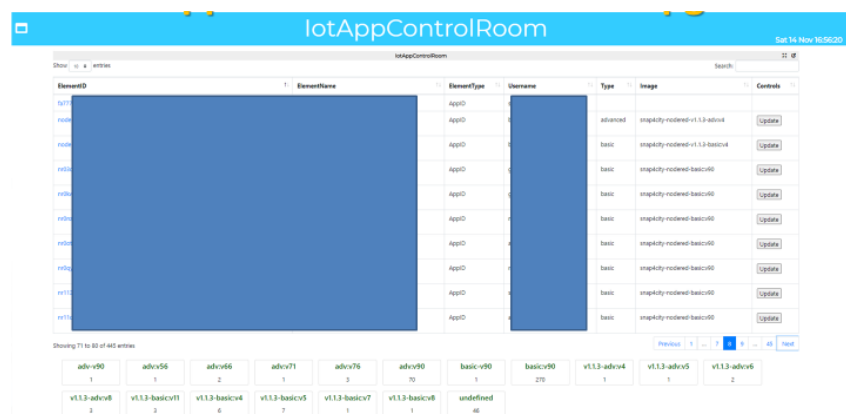
<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTY0NA==>

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

Monitoring Photos and Comments from Web and Mobile App



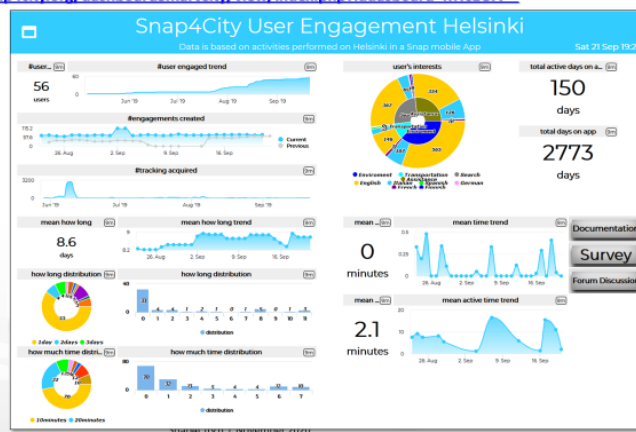
IoT App versions management (blue patches hide private data)



Mobile App monitoring: only internally accessible (for example)

<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc1OA==>

- Dashboard monitoring the Mobile App:
- Collecting the clicks
 - Describing the community of users in terms of the profile aspects
 - Measuring the time spend, and topics of interest of the users, etc.



<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc1OA==>

Other Monitoring and quality controls:

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

3.16.3 - Snap4City Platform Automation

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

Examples of automated activities:

- Harvesting of IoT Broker Devices and their registration in the platform
- registration of data models into **Knowledge Base**
- registration of data model to **Dashboard Wizard** and **Data Inspector**
- Production of 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
Smarthings 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.
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	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/
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
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 direct Dashboards with Kibana.
BIM	Business Information Modelling. In Snap4City, the tool for BIM is performed by BIMServer. https://github.com/opensourceBIM/BIMserver .
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.
BIM Server	A tool for managing BIM representation and distribution. https://github.com/opensourceBIM/BIMserver .
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 .
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 IoT App or JavaScript 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.
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.
Connectors	Connectors can be found/implemented in the Snap4City platform using different approaches, by using: <ul style="list-style-type: none"> 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.
Copernicus	Satellite data: pollution, weather, land, climate, atmosphere, security, emergency, etc. https://www.copernicus.eu/en
Custom Widget	In Snap4City, widgets are the components of the Dashboard to visualize views on data (with animation, graphics, synoptics, etc.) and also to collect interaction from the users (buttons, faces to be clicked, sliders, keypad, text pad, etc.). see HOW to create custom widgets in SVG, and examples: https://www.snap4city.org/651 https://www.snap4city.org/595 https://www.snap4city.org/644 https://www.snap4city.org/663 How to create widgets by programming: https://www.snap4city.org/153
D3.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/das/
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/das/

Dashboard Structure	The structure of an applications starting from the Structure of the single Dashboard. The model is represented by a LOG graph, and/or a hierarchy of the model. A Dashboard included a number of Widgets. Each of which may refer to one or more data sources 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/da/
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/di/
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	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 and network of services
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
DISCES-EM	Distributed scheduler for elastic management for Smart City by Snap4City. 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/sys/

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 TLS, 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 which 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.
External Services, 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.
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 or 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/iot/
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/iot/
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/iot/
IoT App, IoT Application	Node-RED process + Snap4City Library of MicroServices SEE IoT App, see section 3.6 in this document. See training part 3 https://www.snap4city.org/download/video/course/iot/
IoT Application, IoT App	Node-RED process + Snap4City Library of MicroServices SEE IoT App, see section 3.6 in this document. See training part 3 https://www.snap4city.org/download/video/course/iot/
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. 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/iot/
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/iot/
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/iot/
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/iot/
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/iot/
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 <ul style="list-style-type: none"> ■ 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/iot/
IoT Orion Broker	FIWARE component: https://fiware-orion.readthedocs.io/en/master/ The IoT Orion Broker support natively the NGSI V1, V2 and LD. It usually provides only the last values of the IoT Devices and not the historical data (data shadow). 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/iot/
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 (/ˈlaɪdər/, 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	Linked Open Graph tool (nothing to do with Logs). 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/
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.
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	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	New name for <u>IoT Applications</u> , <u>IoT App</u>
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.
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	<p>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:</p> <ul style="list-style-type: none"> • API reachability/availability performed by E015 external service: https://www.snap4city.org/388 • Smart City API performance: https://dashboard.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTkw • DISCES performance: https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjE3Mw== • MyKPI monitoring: https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTY0NA== • Traffic Analyzer: see AMMA • Data Flow Global analyzer: see DevDash • Container Cluster healthiness: see section 3.14, only internally accessible • Mobile App monitoring: only internally accessible • WEB Server Performance and monitoring: only internally accessible • Marathon and Mesos Monitoring: only internally accessible • Cloud Services Monitoring and control: from Vmware, only internally accessible • NIFI Monitoring: an IoT App which uses NIFI API to monitor critical conditions and send alerts • IoT Orion Broker Monitoring: the broker provides some statistics on notifications and logs. https://FIWARE-orion.readthedocs.io/en/master/admin/perf_tuning/index.html • HTTP server tuning + Notification modes and performance
Quality Assessment of IOT Data and services	<p>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.</p> <p>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.</p>
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.
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.
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/app/
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).

Snap4City API	See API
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/sys/
Snap4City IoT App	See IoT App, IoT Application Node-RED process + Snap4City Library of MicroServices
Snap4City Smart Application	Snap4City Smart Application is an applications 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.
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 can be integrated with the solution provided. https://www.eclipse.org/sumo/ https://sumo.dlr.de/docs/ https://en.wikipedia.org/wiki/Simulation_of_Urban_Mobility
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 Reconstruct ion	Algorithm and Tool for computing the traffic flow in any 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 Manageme nt and control	<ul style="list-style-type: none"> • 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>

- [ParkingPredDEEP] S. Bilotta, L.A. Ipsaro Plaesi, P. Nesi, "Predicting free parking slots via deep learning in short-mid terms explaining temporal impact of features", accepted for publication on IEEE Access, 2023.
- [DigitalTwinMTAP] L. Adreani, P. Bellini, C. Colombo, M. Fanfani, P. Nesi, G. Pantaleo, R. Pisanu, "Implementing Integrated Digital Twin Modelling and Representation into the Snap4City Platform for Smart City Solutions", accepted for publication on Multimedia Tools and Applications, Springer, 2023
- [ThermalCAM1] E. Collini, L. A. Ipsaro Palesi, P. Nesi, G. Pantaleo, W. Zhao, "Flexible Thermal Camera Solution for Smart City People Detection and Counting", Multimedia Tools and Applications, Springer, 2023. 10.1007/s11042-023-16374-x <https://link.springer.com/content/pdf/10.1007/s11042-023-16374-x.pdf>
- [EntityModel and Brokers] P. Bellini, L. A. Ipsaro Palesi, A. Giovannoni, P. Nesi, "Managing Complexity of Data Models and Performance in Broker-Based Internet/Web of Things Architectures", accepted for publication on Elsevier Journal: Internet of Things, 2023 <https://www.sciencedirect.com/science/article/pii/S2542660523001579>
- [DSS-bigdataVisualAnalytics] P. Bellini, E. Collini, P. Nesi, L. A. Ipsaro Palesi, G. Pantaleo, "Decision Support Enhancement Through Big Data Visual Analytics", Journal of Visual Language and Computing, pp.1-7, N.1, 2023. 10-18293/JVLC2023-N1-026 <https://ksiresearch.org/jvlc/journal/JVLC2023N1/paper026.pdf>
- [What-IF-trafficflow] P. Bellini, S. Bilotta, L. A. Ipsaro Palesi, P. Nesi, G. Pantaleo, "Vehicular Traffic Flow Reconstruction Analysis to Mitigate Scenarios with Large City Changes", IEEE Access, 2022, ISSN: 2169-3536. <https://ieeexplore.ieee.org/document/9984661>
- [TrafficFlowPrediction-DeepLearning] S. Bilotta, E. Collini, P. Nesi, G. Pantaleo, Short-Term Prediction of City Vehicle Flow via Convolutional Deep Learning, IEEE Access, 2022, 10.1109/ACCESS.2022.3217240.
- [DORAM] A. Arman, C. Badii, P. Bellini, S. Bilotta, P. Nesi, M. Paolucci, Analyzing demand with respect to offer of mobility, Applied Science, MDPI, 2022. <https://www.mdpi.com/2076-3417/12/18/8982>
- [EstimatingCO2Emissions] S. Bilotta, P. Nesi, "Estimating CO2 Emissions from IoT Traffic Flow Sensors and Reconstruction", Sensors, MDPI, 2022. <https://www.mdpi.com/1424-8220/22/9/3382/>
- [PredictingLandslideEvents] E. Collini, L. A. Ipsaro Palesi, P. Nesi, G. Pantaleo, N. Nocentini, A. Rosi, "Predicting and Understanding Landslide Events with Explainable AI", IEEE Access, 2022. 10.1109/ACCESS.2022.3158328 <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9732490>
- [IoT-enabledSmartCities] P. Bellini, P. Nesi, G. Pantaleo, "IoT-enabled Smart Cities: a review of concepts, frameworks and key technologies", Applied Science, MDPI, 2022. <https://www.mdpi.com/2076-3417/12/3/1607>
- [MultiClusteringRecommendation] P. Bellini, L. A. Ipsaro Palesi, P. Nesi, G. Pantaleo, "Multi Clustering Recommendation System for Fashion Retail", Multimedia Tools and Applications, Springer, 2022. <https://link.springer.com/article/10.1007/s11042-021-11837-5>
- [AutomatingIoTDataIngestion] A. Arman, P. Bellini, D. Bologna, P. Nesi, G. Pantaleo, M. Paolucci, "Automating IoT Data Ingestion Enabling Visual Representation", Sensors, MDPI, 2021. <https://www.mdpi.com/1424-8220/21/24/8429>
- [DigitalTwin1] L. Adreani, P. Bellini, C. Colombo, M. Fanfani, P. Nesi, G. Pantaleo, R. Pisanu, "Digital Twin Framework for Smart City Solutions", DMSVIVA 2022, The 28th International DMS Conference on Visualization and Visual Languages, KSIR Virtual Conference Center, Pittsburgh, USA, June 29 - 30, 2022, <http://ksiresearch.org/seke/dmsviva22.html>, Best Paper Award
- [IOT-Directory] P. Bellini, C. Camerota, P. Nesi, "Automating Heterogeneous IoT Device Networks from Multiple Brokers with Multiple Data Models", 2022 Global Internet of Things Summit (GloTS), Dublin, 20-23 June, 2022, IEEE Press. <https://globaliotsummit.org/>
- [DigitalTwin2] L. Adreani, C. Colombo, M. Fanfani, P. Nesi, G. Pantaleo, R. Pisanu, "Rendering 3D City for Smart City Digital Twin", IEEE SMARTCOMP, June 2022, Aalto University, Espoo, Finland. <https://smartcomp.aalto.fi/>
- [SmartSpecialization] S. Chiordi, G. Desogus, C. Garau, P. Nesi, P. Zamperlin, "A Preliminary Survey on Smart Specialization Plat-forms: Evaluation of European Best Practices", ICCSA2022, 22nd Int. Conf. on Computational Science and its Applications, Malaga, Spain, July, 2022, LNCS Springer Verlag.

- <https://sites.unica.it/weaki-transit/2022/01/21/speed2022/>
https://www.snap4city.org/download/video/ICCSA22_Chiordietal.pdf
- [FederatingSmartCities] A. Arman, P. Bellini, P. Nesi, "Searching for Heterogeneous GeoLocated Services via API Federation", ICCSA2022, 22nd Int. Conf. on Computational Science and its Applications, Malaga, Spain, July, 2022, LNCS Springer Verlag. <https://sites.unica.it/weaki-transit/2022/01/21/speed2022/>
https://www.snap4city.org/download/video/LNCS_FederatedKB-SuperServiceMap-v0-4.pdf
- [GeneratingDigitalTwin] L. Adreani, C. Colombo, M. Fanfani, P. Nesi, G. Pantaleo, R. Pisanu, "A Photorealistic 3D City Modeling Framework for Smart City Digital Twin", SCC workshop at IEEE SMARTCOMP, June 2022, Aalto University, Espoo, Finland. <https://smartcomp.aalto.fi/>
<https://www.snap4city.org/download/video/SCC2022.pdf>
- [ShortTermPredictionBikeSharing2021] E. Collini, P. Nesi, G. Pantaleo, "Deep Learning for Short-Term Prediction of Available Bikes on Bike-Sharing Stations", under publication on IEEE Access, pp.1-11, Print ISSN: 2169-3536, Online ISSN: 2169-3536, DOI: <https://doi.org/10.1109/ACCESS.2021.3110794>
- [ChemicalPlant2021] P. Bellini, D. Cenni, N. Mitolo, P. Nesi, G. Pantaleo, M. Soderi, "High Level Control of Chemical Plant by Industry 4.0 Solutions", Journal of Industrial Information Integration, Elsevier. <https://doi.org/10.1016/j.jii.2021.100276>
- [LongTermPredictionBikeSharing2021] D. Cenni, E. Collini, P. Nesi, G. Pantaleo, I. Paoli, "Long-Term Prediction of Bikes Availability on Bike-Sharing Stations", Journal of Visual Languages and Computing, KSI. <https://doi.org/10.18293/jvlc2021-n1-001>
- [SatelliteData2021] P. Bellini, D. Cenni, N. Mitolo, P. Nesi, G. Pantaleo, "Exploiting Satellite Data in the Context of Smart City Applications", The 5th IEEE International Conference on Smart City Innovations - Track 1: Theory, Modeling and Methodologies, 2021. <http://ieeesmartworld.org/sci/>
- [PredictiveMaintenance2021] P. Bellini, D. Cenni, L. A. Ipsaro Palesi, P. Nesi, G. Pantaleo, "A Deep Learning Approach for Short Term Prediction of Industrial Plant Working Status", 7th IEEE International Conference on Big Data Service and Machine Learning Applications, 23-26 August, 2021. <http://www.big-dataservice.net/index.html>
- [15MinCityIndex2021] C. Badii, P. Bellini, D. Cenni, S. Chiordi, N. Mitolo, P. Nesi, M. Paolucci, "Computing 15MinCityIndexes on the basis of Open Data and Services", Proc. of the 2021 International Conference on Computational Science and Its Applications. Published on LNCS Springer. <https://iccsa.org/>
https://www.snap4city.org/drupal/sites/default/files/files/computing15minCityIndex_ICCSA_v0-3.pdf
web page: <https://www.snap4city.org/drupal/node/652>
- [LongTermPredictionNO2-2021] P. Bellini, S. Bilotta, D. Cenni, E. Collini, P. Nesi, G. Pantaleo, M. Paolucci, "Long Term Predictions of NO2 Average Values via Deep Learning", Proc. of the 2021 International Conference on Computational Science and Its Applications. Published on LNCS Springer. <https://iccsa.org/>
- [UserBehaviourModalities2021] C. Badii, A. Difino, P. Nesi, I. Paoli, M. Paolucci, "Classification of Users Transportation Modalities from Mobiles in Real Operating Conditions", Multimedia Tools and Applications, Springer, 2021. <https://doi.org/10.1007/s11042-021-10993-y> or as a PDF here <https://link.springer.com/content/pdf/10.1007/s11042-021-10993-y.pdf>
- [Resilience2021] E. Bellini, P. Bellini, D. Cenni, P. Nesi, G. Pantaleo, I. Paoli, M. Paolucci, "An IoE and Big Multimedia Data approach for Urban Transport System resilience management in Smart City", Sensors, MDPI, 2021, <https://www.mdpi.com/1424-8220/21/2/435/pdf>
- [AnomalyDetection2020] P. Bellini, D. Cenni, P. Nesi, "Anomaly Detection on IOT Data for Smart City", Proc of 6th IEEE International Workshop on Sensors and Smart Cities, with IEEE SmartComp, 14-17 Sept. 2020, Bologna, Italy. <http://ssc2020.unime.it/>
<https://www.snap4city.org/download/video/AnomalyDetection2020.pdf>
- [BlgDataService2018] P. Nesi, P. Bellini, M. Paolucci, I. Zaza, "Smart City architecture for data ingestion and analytics: processes and solutions", IEEE BigDataService 2018, Bamberg, Germany, March 26 - 29, 2018. <https://www.snap4city.org/download/video/BlgDataService2018.pdf>
- [COVID-19] C. Badii, P. Bellini, S. Bilotta, D. Bologna, D. Cenni, A. Difino, A. Ipsaro Palesi, N. Mitolo, P. Nesi, G. Pantaleo, I. Paoli, M. Paolucci, M. Soderi, "How COVID-19 Lockdown Impacted on Mobility and Environmental data", Bollettino della Società Geografica Italiana, FuPress, June 2020, DOI: 10.13128/bsgi.v2i2.932
<https://drive.google.com/file/d/1hN4dFuEXLMLq4rMY8u88ilZnrVk3HPil/view>
<https://www.snap4city.org/download/video/COVID-19.pdf>

- [DashboardProduction2020] Q. Han, P. Nesi, G. Pantaleo, I. Paoli, "Smart City Dashboards: Design, Development and Evaluation", Proc. Of the IEEE ICHMS 2020, International Conference on Human Machine Systems, September 2020. <http://ichms.dimes.unical.it/>
<https://www.snap4city.org/download/video/DashboardProduction2020.pdf>
- [Dashboards2019] P. Bellini, D. Cenni, M. Marazzini, N. Mitolo, P. Nesi, M. Paolucci, "Smart City Control Room Dashboards: Big Data Infrastructure, from data to decision support", Journal of Visual Languages and Computing, 10.18293/VLSS2018-030
<https://ksiresearchorg.ipage.com/vlss/journal/VLSS2018/paper%2030.pdf>
<https://www.snap4city.org/download/video/Dashboards2019.pdf>
- [DataFlow2019] P. Bellini, F. Bugli, P. Nesi, G. Pantaleo, M. Paolucci, I. Zaza, "Data Flow Management and Visual Analytic for Big Data Smart City/IOT", 19th IEEE Int. Conf. on Scalable Computing and Communication, IEEE SCALCOM 2019, Leicester, UK <https://www.slideshare.net/paolonesi/data-flow-management-and-visual-analytic-for-big-data-smart-cityiot>
<https://www.snap4city.org/download/video/DataFlow2019.pdf>
- [DI-DatalInspectorBase] P. Bellini, D. Bologna, Q. Han, P. Nesi, G. Pantaleo, M. Paolucci, "Data Ingestion and Inspection for Smart City Applications", proc. Of IEEE International Conference SMARTCOMP 2020, Bologna, Italy, 2020. <http://www.smart-comp.org/>
<https://www.snap4city.org/download/video/DI-DatalInspectorBase.pdf>
- [DORAM2020] A. Arman, P. Bellini, P. Nesi, M. Paolucci, "Analysing Public Transportation Offer wrt Mobility Demand", ACM Workshop on Technology Enablers and Innovative Applications for Smart Cities and Communities (TESCA 2019), November 10, 2019 at Columbia University, New York, USA.
<https://dl.acm.org/citation.cfm?id=3364828> <https://doi.org/10.1145/3364544.3364828>
https://dl.acm.org/ft_gateway.cfm?id=3364828&ftid=2101673&dwn=1&CFID=180...
<https://www.snap4city.org/download/video/DORAM2020.pdf>
- [FederatedKnowledgeBase2020] P. Bellini, D. Nesi, P. Nesi, M. Soderi, "Federation of Smart City Services via APIs", Proc of 6th IEEE International Workshop on Sensors and Smart Cities, with IEEE SmartComp, 14-17 Sept. 2020, Bologna, Italy. <http://ssc2020.unime.it/>
<https://www.snap4city.org/download/video/FederatedKnowledgeBase2020.pdf>
- [Gov2018] M. Azzari, C. Garau, P. Nesi, M. Paolucci, P. Zamperlin, "Smart City Governance Strategies to better move towards a Smart Urbanism", The 18th International Conference on Computational Science and Its Applications (ICCSA 2018), July 2 - 5, 2018 in Melbourne, with the Monash University, Australia.
<https://www.snap4city.org/download/video/Gov2018.pdf>
- [Gov2020] C. Garau, P. Nesi, I. Paoli, M. Paolucci, P. Zamperlin, A Big Data Platform for Smart and Sustainable Cities: Environmental Monitoring case studies in Europe. Proc. Of International Conference on Computational Science and its Applications, ICCSA2020. Cagliari, Italy, 1-4 July 2020.
<http://www.iccsa.org/> https://link.springer.com/chapter/10.1007%2F978-3-030-58820-5_30
<https://www.snap4city.org/download/video/Gov2020.pdf>
- [Heatmap2020] C. Badii, S. Bilotta, D. Cenni, A. Difino, P. Nesi, I. Paoli, M. Paolucci, "High Density Real-Time Air Quality Derived Services from IoT Networks", Sensors, MDPI, 2020. <https://www.mdpi.com/1424-8220/20/18/5435/htm>
<https://www.snap4city.org/download/video/Heatmap2020.pdf>
- [Industry4.0-2020] C. Badii, P. Bellini, D. Cenni, N. Mitolo, P. Nesi, G. Pantaleo, M. Soderi, "Industry 4.0 Synoptics Controlled by IoT Applications in Node-RED", Proc. Of the IEEE iThings, Nov 02-06, 2020.
<https://www.snap4city.org/download/video/Industry40-2020.pdf>
- [KnowledgeBase2018] P. Bellini, P. Nesi, "Performance Assessment of RDF Graph Databases for Smart City Services", Journal of Visual Language and Computing, Elsevier, 2018.
<https://doi.org/10.1016/j.jvlc.2018.03.002>
<https://www.snap4city.org/download/video/KnowledgeBase2018.pdf>
- [LivingLab2018] P. Nesi, M. Paolucci, "Supporting Living Lab with Life Cycle and Tools for Smart City Environments", The 24th International DMS Conference on Visualization and Visual Languages, DMSVIVA 2018, Hotel Pullman, Redwood City, San Francisco Bay, California, USA, June 29 - 30, 2018
<https://www.snap4city.org/download/video/LivingLab2018.pdf>
- [MicroServices2019] C. Badii, P. Bellini, A. Difino, P. Nesi, G. Pantaleo, M. Paolucci, "MicroServices Suite for Smart City Applications", Sensors, MDPI, 2019. <https://doi.org/10.3390/s19214798>
<https://www.snap4city.org/download/video/MicroServices2019.pdf>
- [MobileDevKit2017] C. Badii, P. Bellini, P. Nesi, M. Paolucci, "A Smart City Development kit for designing Web and Mobile Apps", IEEE international Conference on Smart City and Innovation, 2017, San

- Francisco. <https://www.slideshare.net/paolonesi/a-smart-city-development-kit-for-designing-web-and-mobile-apps>
<https://www.snap4city.org/download/video/MobileDevKit2017.pdf>
- [MobilityMicroServices2019] C. Badii, P. Bellini, A. Difino, P. Nesi, "Sii-Mobility: an IOT/IOE architecture to enhance smart city services of mobility and transportation", Sensors, MDPI, 2019.
<https://doi.org/10.3390/s19010001> <https://www.mdpi.com/1424-8220/19/1/1/pdf>
<https://www.snap4city.org/download/video/MobilityMicroServices2019.pdf>
- [NOX prediction 2020] C. Badii, S. Bilotta, D. Cenni, A. Difino, P. Nesi, G. Pantaleo, I. Paoli, M. Paolucci, "Real-Time Automatic Air Pollution Services from IOT Data Network", proc. Of IEEE Symposium on Computers and Communications (ISCC), MOCS track, 10th Workshop on Management of Cloud and Smart City System, 2020 July 7th, Rennes, France. <https://conferences.imt-atlantique.fr/iscc2020/>
<https://www.snap4city.org/download/video/NOXprediction2020.pdf>
- [Resilience2017] E. Bellini, P. Ceravolo, P. Nesi, "Quantify resilience enhancement of UTS through exploiting connected community and internet of everything emerging technologies", 2017,
<http://hdl.handle.net/2158/1105460>, ACM TRANSACTIONS ON INTERNET TECHNOLOGY
<https://dl.acm.org/citation.cfm?id=3137572>
<https://www.snap4city.org/download/video/Resilience2017.pdf>
- [Resilience2019] E. Bellini, L. Cocone, P. Nesi, "A Functional Resonance Analysis Method driven Resilience Quantification for socio-technical System", IEEE Systems Journal, DOI 10.1109/JSYST.2019.2905713, ISSN: 1932-8184, ISSN: 1937-9234, pp.1-11, 2019 <https://ieeexplore.ieee.org/document/8686178>
<https://www.snap4city.org/download/video/Resilience2019.pdf>
- [Scalable2018] C. Badii, E. G. Belay, P. Bellini, D. Cenni, M. Marazzini, M. Mesiti, P. Nesi, G. Pantaleo, M. Paolucci, S. Valtolina, M. Soderi, I. Zaza, "Snap4City: A Scalable IOT/IOE Platform for Developing Smart City Applications", Int. Conf. IEEE Smart City Innovation, Cina 2018, IEEE Press.
<https://ieeexplore.ieee.org/document/8560331/>
<https://www.snap4city.org/download/video/Scalable2018.pdf>
- [SCAPI-KB2017] C. Badii, P. Bellini, D. Cenni, A. Difino, P. Nesi, M. Paolucci, "Analysis and Assessment of a Knowledge Based Smart City Architecture Providing Service APIs", Future Generation Computer Systems, Elsevier, 2017, <http://dx.doi.org/10.1016/j.future.2017.05.001>
<https://www.snap4city.org/download/video/SCAPI-KB2017.pdf>
- [Security2020] C. Badii, P. Bellini, A. Difino, P. Nesi, "Smart City IoT Platform Respecting GDPR Privacy and Security Aspects", IEEE Access, 2020. 10.1109/ACCESS.2020.2968741
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8966344>
<https://www.snap4city.org/download/video/Security2020.pdf>
- [SmartDS] M. Bartolozzi, P. Bellini, P. Nesi, G. Pantaleo and L. Santi, "A Smart Decision Support System for Smart City," 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, 2015, pp. 117-122, December 2015, Cina, IEEE press, doi: 10.1109/SmartCity.2015.57
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7463711&isnumber=7463653>
<https://www.snap4city.org/download/video/SmartDS.pdf>
- [SmartParking2018] C. Badii, P. Nesi, I. Paoli, "Predicting available parking slots on critical and regular services exploiting a range of open data", IEEE Access, 2018,
<https://ieeexplore.ieee.org/abstract/document/8430514/>
<https://www.snap4city.org/download/video/SmartParking2018.pdf>
- [TrafficFlowReconstruction2020] S. Bilotta, P. Nesi, "Traffic Flow Reconstruction by Solving Indeterminacy on Traffic Distribution at Junctions", Future Generation Computer Systems, Elsevier, 2020, 2021.
[https://authors.elsevier.com/sd/article/S0167-739X\(20\)30835-9](https://authors.elsevier.com/sd/article/S0167-739X(20)30835-9)
<https://www.snap4city.org/download/video/TrafficFlowReconstruction2020.pdf>
- [TrafficPredictions2020] S. Bilotta, P. Nesi, I. Paoli, "Real-Time System for Short- and Log-Term Prediction of Vehicle Flow", IEEE Smart Data Service as Proc of IEEE Services, 2020.
<https://www.snap4city.org/download/video/TrafficPredictions2020.pdf>
- [TwitterVigilance2017] D. Cenni, P. Nesi, G. Pantaleo, I. Zaza, "Twitter Vigilance: a Multi-User platform for Cross-Domain Twitter Data Analytics, NLP and Sentiment Analysis", IEEE international Conference on Smart City and Innovation, 2017, San Francisco. <https://www.slideshare.net/paolonesi/twitter-vigilance-a-multiuser-platform-for-crossdomain-twitter-data-analytics-nlp-and-sentiment-analysis>
<https://www.snap4city.org/download/video/TwitterVigilance2017.pdf>
- [UserBehavior2020] C. Badii, A. Difino, P. Nesi, I. Paoli, M. Paolucci, "Classification of Users' Transportation Modalities in Real Conditions", Proc. Of the 26th International DMS Conference on Visualization and



Visual Languages, DMSVIVA, Wyndham Pittsburgh University Center, Pittsburg, USA, July 7-8, 2020.

<http://ksiresearch.org/seke/dmsviva20.html>

<https://www.snap4city.org/download/video/UserBehavior2020.pdf>

[UserBehaviour2017] P. Bellini, D. Cenni, P. Nesi, I. Paoli, "Wi-Fi Based City Users' Behaviour Analysis for Smart City", Journal of Visual Language and Computing, Elsevier, 2017.

<http://www.sciencedirect.com/science/article/pii/S1045926X17300083>

<https://www.snap4city.org/download/video/UserBehaviour2017.pdf>

[UserEngagement2017] C. Badii, P. Bellini, D. Cenni, A. Difino, P. Nesi, M. Paolucci, "User Engagement Engine for Smart City Strategies", IEEE International Conference on Smart Computing, IEEE SMARTCOMP 2017, Hong Kong.

<https://www.snap4city.org/download/video/UserEngagement2017.pdf>

5.2 - Snap4City Main documentation web pages

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>

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

Training Course page	https://www.snap4city.org/577 see this link to get the updated versions
Training 2023	https://www.snap4city.org/drupal/node/944
Training Part 1 - overview	
Training Part 2 - Dashboard	
Training Part 3 - IoT App and networks	
Training Part 4 - Data Analytics	
Training Part 5 - Data Ingestion	
Training Part 6 - System deploy, summary	
Training Part 7 - API and Mobile Apps	
Training Part 8 - Design and Develop Smart Solutions	




Snap4City Impact Story at FIWARE	https://www.snap4city.org/drupal/sites/default/files/files/FF_ImpactStories_Snap4City.pdf
Snap4Tech Design and Develop smart Solution	https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf
Client Side Business Logic	https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf
5G experimentation and trial in Italy for Prato city with Wind-3G Open Fiber Operator:	https://www.snap4city.org/547
DockerHub DISIT	https://cloud.docker.com/u/disitlab
E015 official APIs of Digital Ecosystem of Regione Lombardia	https://www.e015.regione.lombardia.it/site/app-detail?id=156
Elastic	https://www.snap4city.org/232
EOSC European Open Science Cloud official service:	https://marketplace.eosc-portal.eu/services/snap4city
Fi-Ware Official Solution	https://marketplace.FIWARE.org/pages/solutions/b8905e91973b420189cce972
Flexibility	https://www.snap4city.org/4
GitHub DISIT lab	https://github.com/disit
Hackathon 2019	https://www.snap4city.org/hackathon
Hackathon 2021	https://www.snap4city.org/757
Interoperability	https://www.snap4city.org/283

Km4City Smart City IOT Ontology	https://www.snap4city.org/download/video/DISIT-km4city-City-Ontology-ita-v5-1.pdf
Node-RED official library Snap4City	https://flows.nodered.org/search?term=snap4city
NPM	https://www.npmjs.com/~disit https://www.npmjs.com/package/node-red-contrib-snap4city-user
Open sources	https://www.snap4city.org/7
Protocols	https://www.snap4city.org/65
Scenariious	https://www.snap4city.org/4
Security	https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8966344
SmartBED	https://www.ifc.cnr.it/images/Ricerca/progetti/LAID.pdf https://www.snap4city.org/668

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

- [Herit-Data and Snap4City: to better manage tourism flows, PDF](#)
- [Digital Twin Local and Global, PDF](#)
- [Social Media Analysis: Twitter Vigilance](#)
- [GDPR Compliant People Detection and Counting using Thermal Cameras,](#)
- [Artificial Intelligence Predicts Landslides in Florence Area](#)
- [Available Parking Slots Prediction](#)
- [Available Bikes and Free Slots Prediction On Bike Sharing Stations](#)
- [Inventory of transferable digital applications and solutions for the tourism ecosystem](#)
- [Scenario: AMPERE Operator Manual](#)
- [Scenario: IMPETUS Intelligent Management of Processes, Ethics and Technology for Urban Safety \(Oslo and Padova\)](#)
- Welcome pages of Organizations in no particular order:
 - [Welcome to Florence](#)
 - [Welcome to West Greece \(Herit-Data\)](#)
 - [Welcome to Valencia, Spain \(Herit-Data\)](#)
 - [Welcome to Pont du Gard, France \(Herit-Data\)](#)
 - [Welcome to Mostar \(Herit-Data\)](#)
 - [Welcome to Dubrovnik \(Herit-Data\)](#)
 - [Welcome to Tuscany](#)
 - [Welcome to Helsinki developers](#)
 - [Welcome to Pistoia](#)
 - [Welcome to Siena](#)
 - [Welcome to Bologna](#)
 - [Welcome to Prato](#)
 - [Welcome on DISIT organization](#)
 - [Welcome to Santiago de Compostela](#)
 - [Welcome to Modena](#)
 - [Welcome to Rome](#)
 - [Welcome to Smart Garda Lake](#)
 - [Welcome to Pisa](#)
 - [Welcome to Livorno](#)
 - [Welcome to Antwerp List of Services and OnLine Data](#)
 - [Welcome to Smart Parking in Lonato del Garda](#)
 - [Scenario: Smart Light Control \(CAPELON\)](#)
- [Scenario: Monitoring Cross Road in Venaria Reale - \(AXIS Camera\)](#)
- [Welcome: how to start using Snap4City for beginners](#)

- [Global Digital Twin for Florence by Snap4City](#)
- [15MinCityIndex: understanding city areas by means of 13 different aspects](#)
- [Snap4City Impact Story on FIWARE: https://fiware-foundation.medium.com/snap4city-fiware-powered-smart-app-builder-for-sentient-cities-acfe24df49d5](https://fiware-foundation.medium.com/snap4city-fiware-powered-smart-app-builder-for-sentient-cities-acfe24df49d5)
 - PDF version: [FF_ImpactStories_Snap4City.pdf](#)
- [Scenario: Smart Parking in Lonato del Garda \(Reverberi\)](#)
- [ALMA FLUIDA activity \(ITALMATIC\)](#)
- [SMART AMBULANCE activity started](#)
- [AMPERE Project - Furniture, art and fashion in an innovative electrochemical process with remote control and circular eco-friendly](#)
- [Scenario: SnapBot: Real-time Smart City services via Telegram](#)
- [Scenario: Copernicus Satellite Data](#)
- [Scenario: SmartBed, Materasso Intelligente](#)
- [Snap4City & Snap4Industry Registered Instances](#)
- [MicroServices Suite for Smart City Applications](#)
- [Scenario: MODBUS for Snap4Industry Snap4City Applications](#)
- [Scenario: MOBIMART Interreg: MOBilità Intelligente MARE Terra](#)
- [Scenario: City of Roma case, mobility and environmental data](#)
- [Scenario: Herit-Data video and aims](#)
- [Scenario: Control Room vs Video Wall](#)
- [Scenario: Snap4Home the case of: Alexa, Philips, Sonoff, TP-link, etc. \(Italiano\)](#)
- [Scenario: how to manage maintenance and accidents workflows](#)
- [Scenario: Snap4Home, how to exploit Snap4City solution on home automation](#)
- [Scenario: Energy Monitoring](#)
- [Scenario: Multipurpose User Engagement Tools](#)
- [Scenario: 5G Enabled Water Cleaning Control \(smart city, industry 4.0\)](#)
- [Scenario: High Level Control of Industrial Plant \(industry 4.0\)](#)
- [Scenario: Vehicle Monitoring via OBD2](#)
- [Data Analytic: Origin Destination Matrices, Algorithms and tools](#)
- [Data Analytic: Traffic Flow Reconstruction](#)
- [Data Analytic: in general, and the cases of Antwerp and Helsinki](#)
- [Data Analytic: Predicting Air Quality](#)
- [Data Analytic: Analyzing Public Transportation Offer wrt Mobility Demand, DORAM Tool](#)
- [Scenario: Events and Museums Monitoring in Antwerp](#)
- [Scenario: High Resolution Prediction of Environmental Data](#)
- [Scenario: Mobility and Transport Analyses in multiple cities](#)
- [Scenario: People Flow Analysis via Wi-Fi](#)
- [Scenario: Antwerp Pilot on Environmental Data](#)

-  [Scenario: Helsinki Pilot on Environmental Data](#)
-  [Scenario: Firenze Smart City Control Room](#)
-  [Scenario: Mobile & Web App: Toscana Where What ... Km4City, Toscana in a Snap](#)
-  [Scenario: Helsinki Pilot on User Behaviour](#)
-  [Scenario: Antwerp Pilot on User Behaviour](#)
- [Smart City Control Room Dashboards: Big Data Infrastructure, from data to decision support](#)
- [Dashboards Touch Screen Support for Situation Rooms and What-iF analysis Operators](#)
- [Data Analytics on Antwerp and Helsinki](#)
- [Decision Support Engines and Models: SmartDS, ResilienceDS](#)
- [IPCAM on Dashboard Widget, Video Cameras](#)
- [Solution: using PAX Counters, monitoring museum and events](#)
- [Solution: selecting the less polluted routing](#)
- [Data Flow Management and Visual Analytic for Big Data Smart City/IOT](#)
- [Snap4City on Social Media](#)
-  [AIRQino: Compact low-cost air quality monitoring station, by CNR IBIE](#)
- [Security, Privacy and GDPR Aspects of Snap4City](#)
- [How to: Develop Smart Applications, Snap4City development Life Cycle](#)

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

- [HOW TO: create a Dashboard in Snap4City](#)
- [HOW TO: add a device to the Snap4City Platform](#)
- [HOW TO: add data sources to the Snap4City Platform](#)
- [HOW TO: define privacy rules for personal data, produced by the end-users own device](#)
- [HOW TO: Develop Smart Applications, Snap4City development Life Cycle](#)
- [HOW TO: HLT vs Ingestion, and HLT vs Widgets](#)
- [HOW TO: Develop an IoT Application for Data Ingestion](#)
- [HOW TO: Upload data into Knowledge Base, ServiceMap \(triple upload\)](#)
- [HOW TO: Create as set of Devices with BulkProcessing](#)
- [HOW TO: Create an IoT Device Model](#)
- [HOW TO: Create an IoT Device Instance from IoT Directory tool](#)
- [HOW TO: save / export a MyKPI data into a CSV file](#)
- [HOW TO: develop DataAnalytic in Python and manage them via IoT App](#)
- [HOW TO: customize AIRQino IoT Device to work on Snap4City](#)
- [HOW TO: setup Libelium Waspote Plug&Sense Smart Environment IoT Device to work on Snap4City](#)
- [MicroServices Suite for Smart City Applications](#)
- [HOW TO: create subflow and groups into IoT Applications, Node-RED](#)
- [HOW TO: define Device in mobility, Mobile IOT Devices, Sensor Mobile](#)
- [HOW TO: Manage IOT Network Components on Snap4City](#)
- [HOW TO: Federation of Smart Cities, by federating Smart City API and Knowledge Base](#)
- [HOW TO: extend Snap4City Platform with new modules and tools](#)
- [HOW TO: Manage BMP and BIM: main features of openMAINT, BMP, BIM](#)
- [HOW TO: Support Multiple Cities/Area with a single Snap4City Platform](#)
- [HOW TO: Time Trend Compare Widget](#)

- [HOW TO: manage a calendar Widget](#)
- [Form Widget on Dashboard and IoT Applications](#)
- [HOW TO: use and control the Widget Data Cube, OLAP](#)
- [HOW TO: work on ServiceMap for developers](#)
- [HOW TO: sizing your Snap4City Solution on private or public Clouds](#)
- [UserManual of SelectorMap block/node on Node-RED, controlling Maps from Node-RED](#)
- [HOW TO: send voice messages on dashboards](#)
- [HOW TO: Federating Smart Cities, Super Servicemap, Knowledge Bases](#)
- [HOW TO: Exploit the D3.js graphic library in Snap4City dashboards](#)
- [HOW TO: Develop a Theme for Dashboards](#)
- [HOWTO: Extended delegation capabilities of IOT Devices](#)
- [HOW TO: Use a FIWARE Smart Data Model for creating devices](#)
- [HOWTO: MultiDataMap for Origin-Destination Matrices](#)
- [HOWTO: Exploit the new DeviceTable widget to show data on a table format.](#)

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

- [Social Media Analysis: Twitter Vigilance](#)
- [GDPR Compliant People Detection and Counting using Thermal Cameras](#)
- [Artificial Intelligence Predicts Landslides in Florence Area](#)
- [Available Parking Slots Prediction](#)
- [Available Bikes and Free Slots Prediction On Bike Sharing Stations](#)
- [Inventory of transferable digital applications and solutions for the tourism ecosystem](#)
- [Scenario: IMPETUS Intelligent Management of Processes, Ethics and Technology for Urban Safety](#)
- [Una piattaforma acciappa-dati per Firenze](#)
- [Snap4City for Sustainable Development Goals](#)
- [UserManual of SelectorMap block/node on Node-RED, controlling Maps from Node-RED](#)
- [Twitter Vigilance: Tweet Annotation tool](#)
- [Global Digital Twin for Florence by Snap4City](#)
- [Doc: Translation Manager Tool](#)
- [Calls of the European Commission of interest for Snap4City, DISIT lab](#)
- [Doc: POI Loader Tool \(load of Excel file\)](#)
- [Doc: BIM integration in Snap4City: Digital Twin Local](#)
- [Doc: Data Table Loader tool \(load of Excel File\)](#)
- [Report management on devices and other aspects](#)
 - [Report Templates Generation and Management \(Reports 1/3\)](#)
 - [Report Activation and Access \(Report 2/3\)](#)
 - [Report Generator management \(3/3\)](#)
- [Digital Twin: Snap4City view](#)
- [Overview Snap4City 2021: Smart aNalytic APp builder for sentient Cities and IOT](#)
- [Why Snap4City should be adopted and why it can be easy replicated / installed](#)
- [Course: Big Data and Analytics, Master Industry 4.0 Design, PISA](#)
- [Comparing Snap4City Dashboard Builder wrt Kibana, Grafana](#)
- [TRAFAIR nella Città di Livorno](#)
- [TRAFAIR nella Città di Firenze](#)
- [TRAFAIR nella Città di Pisa](#)
- [Example of Hijri Date Support on Dashboards](#)
- [Scenario: SnapBot: Real-time Smart City services via Telegram](#)
- [DCL: DataCity-Large Structure and Configuration](#)
- [List of Dashboard Widgets, for Dashboard Builder](#)
- [Scenario: Copernicus Satellite Data](#)
- [Snap4City: GDPR compliance and requirements](#)
- [Snap4City: Security and Privacy Assessment](#)
- [Custom Widgets: Table explanation, as SVG](#)

- [Snap4City/Industry Registered Instances](#)
- [Scenario: MODBUS for Snap4Industry Snap4City Applications](#)
- [IoT Directory - DEV \(Technical Reference Manual\)](#)
- [IoT Directory - API \(development manual\)](#)
- [IoT Directory - UI \(user Interface\), HOW TO: work on IoT Directory User Manual](#)
- [Scenario: MOBIMART Interreg: MOBilità Intelligente MARE Terra](#)
- [Custom Synoptics and Widgets for Dashboards](#)
- [DORAM: how to create alternative scenarios \(mobility demand vs transport offer\)](#)
- [Scenario: City of Roma case, mobility and environmental data](#)
- [Scenario: Herit-Data video and aims](#)
- [Twitter Vigilance, social media monitoring and early warning](#)
- [Snap4City Certificates of Proficiency and/or Participation](#)
- [Scenario: Control Room vs Video Wall](#)
- [DISIT Lab Traffic Flow Reconstruction V1.0 Tool License Agreement, version 1.2, 2020](#)
- [Scenario: Snap4Home, how to exploit Snap4City solution on home automation](#)
- [Snap4City info su Andamenti Nazionali e Regionali infezione COVID-19](#)
- [Snap4City Harmonized Data Ingestion process](#)
- [Hotels in Florence having an agreement with UNIFI](#)
- [New Support Widgets for Bars and series](#)
- [How to register automatically an uri callback on Orion Context Broker](#)
- [Installing Snap4City: Appliances, Virtual Machines/Containers ready to use on your premise for Smart City and IoT Applications](#)
- [Snap4City 1st Place award by Select4Cities PCP](#)
- [Snap4City Business Model](#)
- [Snap4City at Smart City Expo World Congress 2019](#)
- [Article on Sensors: <https://www.mdpi.com/1424-8220/19/21/4798/htm>, MicroServices Suite for Smart City Applications](#)
- [Scenario: Energy Monitoring](#)
- [Data Analytics: Analyzing Public Transportation Offer wrt Mobility Demand](#)
- [Security, Privacy and GDPR Aspects of Snap4City](#)
- [How to: Develop Smart Applications, Snap4City development Life Cycle](#)
- [Smart City Control Room Dashboards: Big Data Infrastructure, from data to decision support](#)
- [DataAnalytics on Antwerp and Helsinki](#)
- [What they state about Snap4City tools](#)
- [HOW TO: activate the zoom of External Services embedded into Dashboard Widget](#)
- [Dashboard info: Mobility and Environment What-IF Analysis, Florence](#)
- [Decision Support Engines and Models: SmartDS, ResilienceDS](#)
- [IPCAM on Dashboard Widget, Video Cameras](#)
- [Solution: using PAX Counters, monitoring museum and events](#)
- [Solution: selecting the less pollute routing](#)
- [How to Install: IoT Directory](#)
- [Data Flow Management and Visual Analytic for Big Data Smart City/IOT](#)
- [Dashboards: Since today all of them have Touch Screen Support](#)
- [Snap4City on Social Media](#)
- [AIRQino: Compact low-cost air quality monitoring station, by CNR IBIE](#)
- [Changing the background map, loading Ortomaps, on Dashboard](#)
- [Custom Synoptics and Widgets for Dashboards](#)
- [Snap4City on the European Open Science Cloud Marketplace, EOSC](#)
- [Tutorial on the Bulk Load Approach for IoT Devices on Snap4City](#)
- [Training Snap4City: from data to sentient City, Smart City and IOT](#)
- [Snap4City Mobile Apps: Helsinki in a Snap, Antwerp in a Snap](#)
- [HOW TO: configure/install StartSNAP4CITYVM: starting appliance of Snap4City](#)
- [Node-RED IoT Applications for Smart City](#)
- [Snap4City Appliance: Virtual Machines ready to use on your premise for Smart City and IoT Applications](#)

- [Phase 3: Iteration 3: Trial on Antwerp and Helsinki \(links\)](#)
- [Snap4City Appliance, Virtual Machines](#)
- [Snap4City vs Fi-Ware](#)
- [Environmental Data Definitions, and Predictive Models](#)
- [Snap4City Hackathon 2019: The Winners](#)
- [Snap4City vs GDPR](#)
- **Articoli in Italiano:** Articoli e Descrizioni in Italiano
 - [DISIT Lab Overview 2021](#)
 - [Course: Big Data and Analytics, Master Industry 4.0 Design, PISACourse: Big Data and Analytics, Master Industry 4.0 Design, PISACourse: Big Data and Analytics, Master Industry 4.0 Design, PISA](#)
 - [Traffico, meteo e qualità dell'aria: l'importanza dei dati per gestire e vivere le città](#)
 - [Corso su IOT, programmazione in Node-RED snap4City in Italiano](#)
 - [Scenario: Snap4Home the case of: Alexa, Philips, Sonoff, TP-link, etc. \(Italiano\)](#)
 - [Smart City Lighthouse ,snap4city, SDLM in Advanced Design LM12, Bologna: SLIDE and VIDEO \(italiano\)](#)
 - [Master Industry 4.0 Design: Big Data and Analytics](#)
 - [Training Snap4City: dai dati alla Città Senziente, Smart City and IOT](#)
 - [La Smart City Control Room di Snap4City \(ITA\)](#)
 - [Data Gathering: acquisizione dati \(ITA\)](#)
 - [Big Data Analytics, analitica \(ITA\)](#)
 - [Snap4City per Smart City e Industria 4.0 \(ITA\)](#)
 - [Sindaco Nardella allo Smart City Expo World Conference 2018](#)
 - [Un Hackathon promosso dal DISIT Lab del Dinfo a livello internazionale \(ITA\)](#)
 - [Training Snap4City: dai dati alla Città Senziente, Smart City and IOT](#)
 - [Una App per smaltire bene i rifiuti elettronici ed elettrici, WEEE](#)
- [Most popular content on Snap4City](#)
- [Snap4City Hackathon 2019 finalists for Antwerp and Helsinki](#)
- [Data Analytic: HDOP as GPS Position Quality Assessment](#)
- [Snap4City: best show cases](#)
- [Air Quality Indexes](#)
- [Codemotion Articles:](#)
 - [SELECT for Cities contest: solve challenges and create value for citizens](#)
 - [How Snap4City can bring your vision of a Smart City to reality](#)
 - [The Snap4City API: a tool for smart city developers](#)
 - [Tutorial: connecting an ESP-12 device to the snap4city network](#)
 - [Tutorial: connecting a raspberry Pi to the Snap4city network](#)
 - [The integration between data and devices: the Snap4City solution](#)
 - [Snap4City Dashboards: Overlook Your City](#)
 - [Toward the Sentient City with Snap4City](#)
 - [Using Snap4City, Innovation and Technology to Improve City](#)
 - [IOT Development: the Snap4City way](#)
 - [Data Gathering and Knowledge Management for Smart Cities in Snap4City Platform](#)
- [Km4City Urban Graph Representation](#)
- [How to Access at R Studio](#)
- [How to Access to ETL development environment](#)
- [Km4City /Snap4City Smart City API Service Availability](#)
- [Snap4Industry: Snap4City for Industry 4.0](#)
- [Snap4City per Smart City e Industria 4.0](#)
- [Snap4City vs GIS solutions and connections](#)
- [Certificate of the Certification Authority](#)
- [Karma - in practice: triplification process](#)
- [Sindaco Nardella allo Smart City Expo World Conference 2018](#)
- [Snap4City Abstract: the most and the advanced aspects](#)

- [Snap4City: Arduino & ESP8266 IoT Device NGSI](#)
- [Snap4City IoT Edge on VM Ubuntu: IoT Application](#)
- [My Personal Statistics](#)
- [User Limits and Consumption of Resources](#)
- [Snap4City platform: from cloud to data analytic as a Living Lab for Student and Researchers](#)
- [Partners and Interoperability](#)
- [Registering IoT Edge: example of Raspberry Pi, total security](#)
- [Mobile Phone Android as IoT Edge with IoT Applications in Node-RED](#)
- [Snap4City Terminology](#)
- [Snap4City Living Lab Life Cycle suggested](#)
- [Architecture of Snap4City with Internal API, documented in Swagger](#)
- [searchable: the data, both structured and unstructured](#)
- [The daisy of MicroServices for Snap4City Dashboard and IoT App.](#)
- [IoT Applications, MicroService Directory, data bidirectional data flows](#)
- [HTML5 MicroApplications Smart City API](#)
- [Snap4City IoT Devices Registration](#)
- [Creating: IoT Device, Raspberry Pi based totally compliant with Snap4City](#)
- [Snap4All IOT Button: based on ESP32](#)
- [Users' Roles on Snap4City](#)
- [Contributing as city stakeholder: company, citizen/developer, researcher, etc., to the Living Lab](#)
- [Real-time and Data-driven Capabilities of API](#)
- [Acting on the basis of Data and Data Analytics](#)
- [Exploring and Drill-Down on City Data](#)
- [Supported file and message formats](#)
- [ETL processes for massive Data Ingestion and Transformation](#)
- [Snap4City Supported Protocols, adding new protocols](#)
- [Testing Snap4City API with Postman](#)
- [High Level supported IOT Protocols](#)
- [Snap4City Collaborative Work and Co-working](#)
- [Snap4City Dictionary of Tools and Modules](#)
- [Snap4City Overview and Architecture vs it users](#)
- [EventLog API](#)
- [References](#)
- [Snap4City Logo](#)
- [Acronyms](#)
- [Doc:Dashboards <https://www.snap4city.org/drupal/node/17>](#)
- [Doc:Notificator <https://www.snap4city.org/drupal/node/18>](#)
- [Doc:IOTApplications <https://www.snap4city.org/drupal/node/22>](#)
- [Doc:IOTDirectoryandDevices <https://www.snap4city.org/drupal/node/40>](#)
- [Doc:ServiceMap&3D <https://www.snap4city.org/drupal/node/19>](#)
- [Doc:MicroApplications <https://www.snap4city.org/drupal/node/54>](#)
- [Doc:ExternalServices <https://www.snap4city.org/drupal/node/55>](#)
- [Doc:DataSetManager <https://www.snap4city.org/drupal/node/26>](#)
- [Doc:ResourceManager <https://www.snap4city.org/drupal/node/27>](#)
- [Doc:RStudioDevelopment <https://www.snap4city.org/drupal/node/25>](#)
- [Doc:ETLDevelopment <https://www.snap4city.org/drupal/node/24>](#)
- [Doc:KnowledgeBaseGraph&Query <https://www.snap4city.org/drupal/node/21>](#)
- [Doc:SmartCityAPI,Swagger <https://www.snap4city.org/drupal/node/20>](#)
- [Doc:TrafficAnalyzer,AMMA <https://www.snap4city.org/drupal/node/43>](#)
- [Doc:DataAnalyzer,DevDash <https://www.snap4city.org/drupal/node/23>](#)
- [Doc:ResourceAnalyzer,ResDash <https://www.snap4city.org/drupal/node/42>](#)
- [Doc:SchedulerDISCES <https://www.snap4city.org/drupal/node/28>](#)