100% OPEN SOURCE

EU GDPR COMPLIANT

PEN Test Passed

SNAP4CITY on EUROPEAN OPEN SCIENCE CLOUD MARKETPLACE

SNAP4 Appliances and Dockers Installations

SNAP4CITY

KM4CITY



Be smart in a SNAP!

LIVING LAB

Data Ingestion Processes and Tools



16 January 2021, Course
https://www.snap4city.org/577

SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO | DISIT

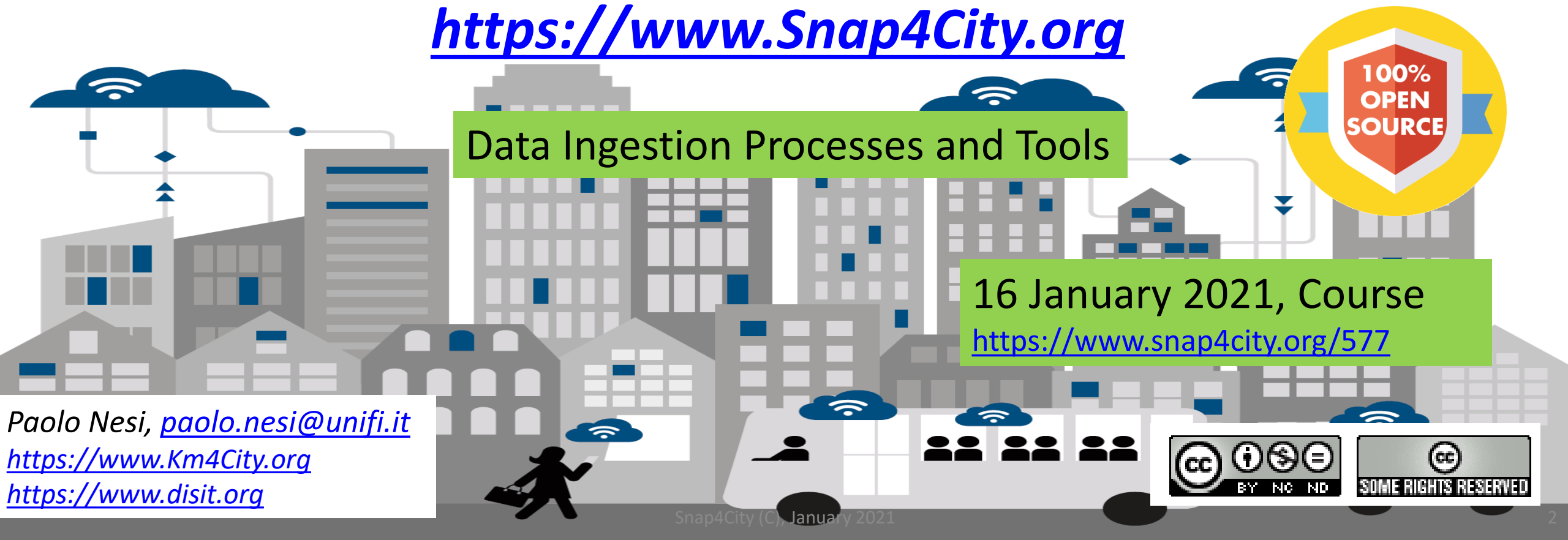
UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

SNAP4CITY

Powered by KM4CITY

scalable Smart aNalytic APplication builder for sentient Cities: for Living Lab and co-working with Stakeholders

https://www.Snap4City.org



100% OPEN SOURCE

Data Ingestion Processes and Tools

16 January 2021, Course
https://www.snap4city.org/577

Paolo Nesi, paolo.nesi@unifi.it
https://www.Km4City.org
https://www.disit.org

CC BY NC ND SOME RIGHTS RESERVED

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

SNAP4CITY
www.snap4city.org

KM 4 CITY

Powered by FIWARE

SNAP4 Appliances and Dockers Installations

FREE TRIAL

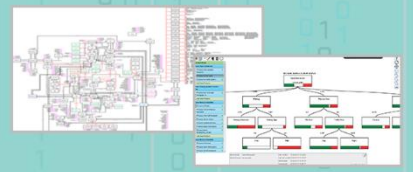
PEN Test Passed

EU GDPR COMPLIANT

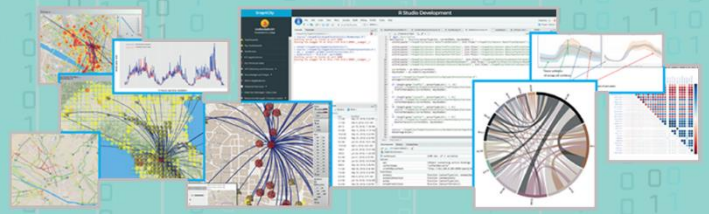
100% OPEN SOURCE



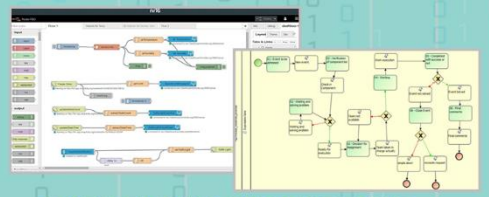
DASHBOARDS AND APPS - CONTROL ROOMS - DECISION SUPPORT SYSTEMS - WHAT-IF ANALYSIS



EXPERT SYSTEM
KNOWLEDGE BASE
STORAGE



BIG DATA ANALYTICS
ARTIFICIAL INTELLIGENCE
BUSINESS INTELLIGENCE
MACHINE LEARNING

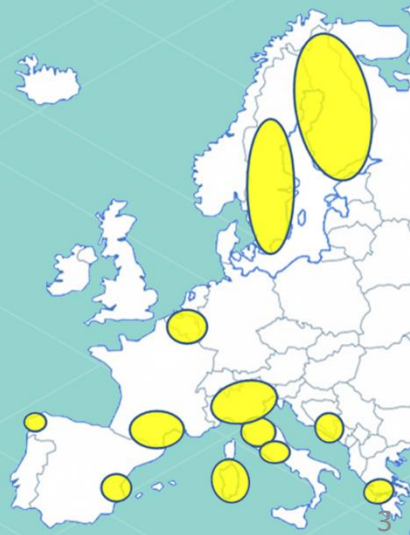


DATA FLOWS, WORKFLOWS
MICROSERVICES
MANAGEMENT



METHODOLOGIES
COURSES AND COMMUNITY
LIVING LABS
DEVELOPMENT TOOLS





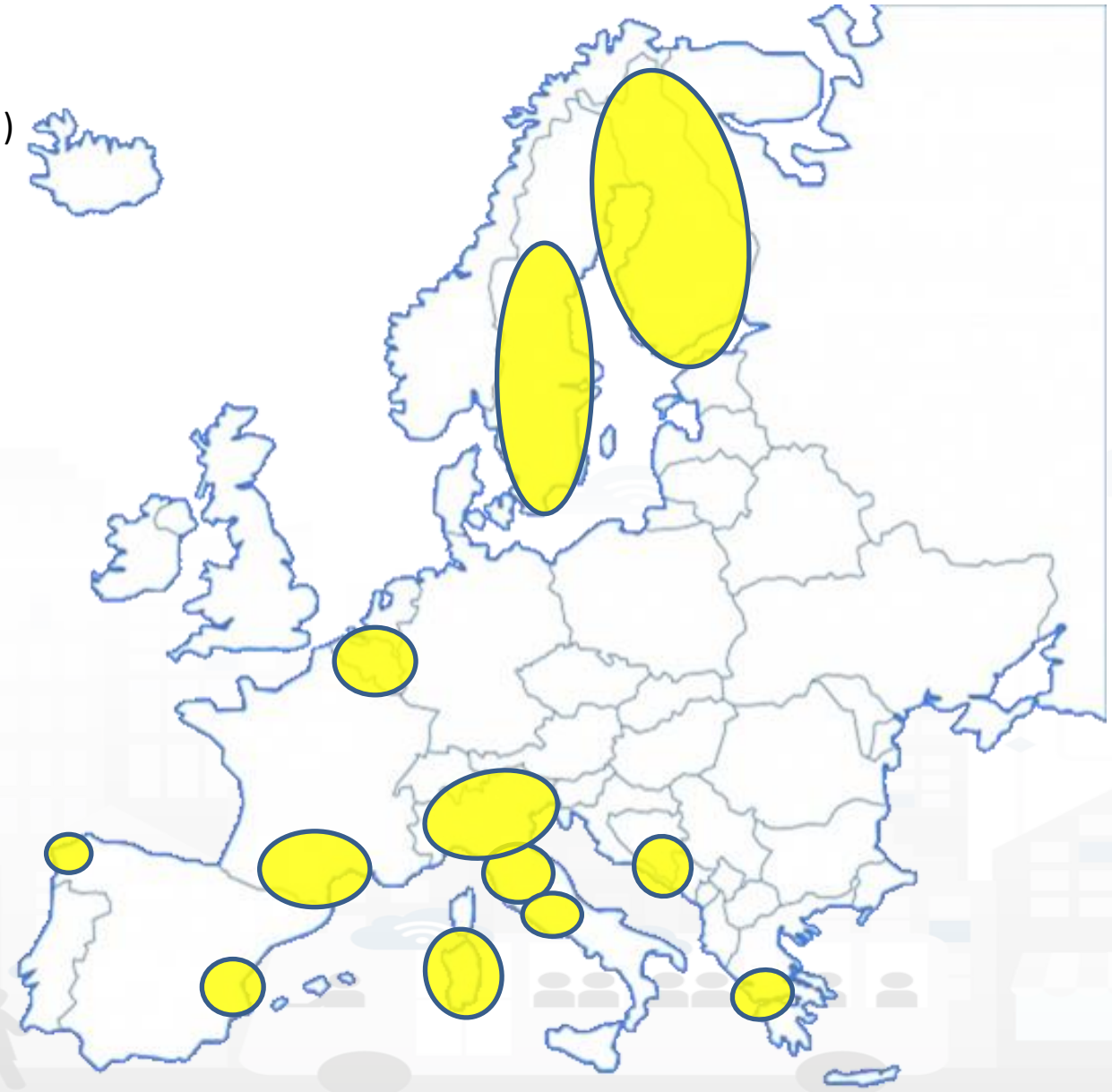
Snap4City/Industry structure

- The **Snap4xxxx** solution is released in Open Source, VM and Docker with fully support of MultiTenant/multiple-Organizations
  - Each Organization may be configured for a separate environment with a set of Maps, Menus, Users, Data, Dashboards, IOT Apps, MicroApplications, Custom Widgets, Models, resources, open data, etc.
- **Https://www.Snap4City.ORG** is the main instance of Snap4xxxx solution managed by DISIT Lab. The main documentation is located and updated on Snap4City.org, GitHUB, dockerHub and Node-Red Library. Snap4City.org is where the last tools are tested and news published.
  - Organizations on Snap4City.org have been created with contracts as for *Platform as a Service*, for testing and for providing *SmartCity as a Service* as well as *Industry 4.0 as a Service*

Coverage 2020

Main Organizations/areas

- Antwerp area (Be)
- Capelon (Sweden: Västerås, Eskilstuna, Karlstad)
- DISIT demo (multiple)
- Dubrovnik, Croatia
- Firenze area (I)
- Garda Lake area (I)
- Helsinki area (Fin)
- Livorno area (I)
- Lonato del Garda (I)
- Modena (I)
- Mostar, Bosnia-Herzegovina
- Pisa area (I)
- Pont du Gard, Occitanie (Fr)
- Roma (I)
- Santiago de Compostela (S)
- Sardegna Region (I)
- SmartBed (multiple)
- Toscana Region (I), SM
- Valencia (S)
- Venezia area (I)
- WestGreece area (Gr)



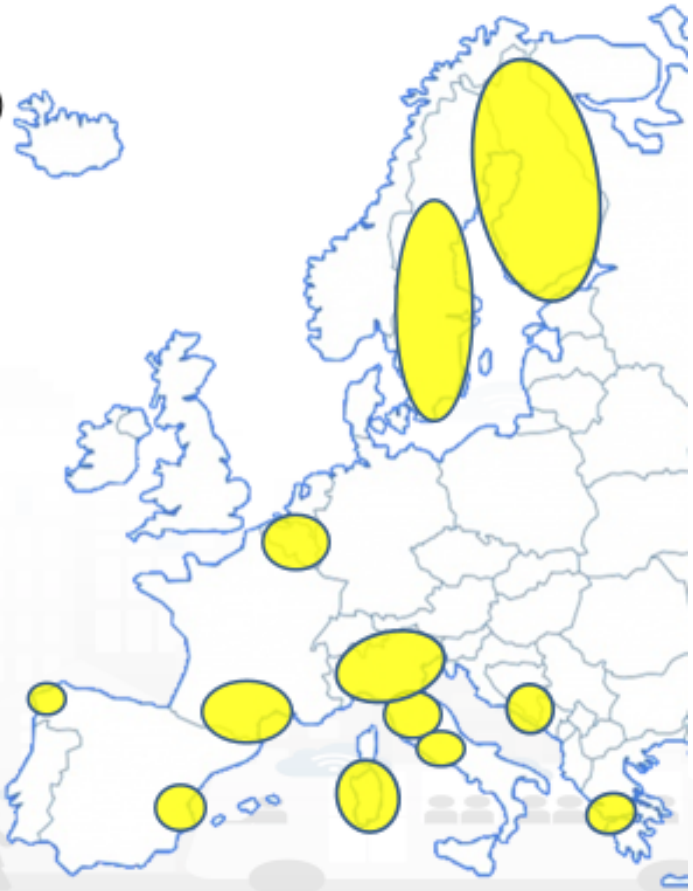
Snap4City/Industry Community

- Most of Organizations on Snap4City.org also correspond to companies or institutions that have an installation of Snap4City tools on their Premise,
  - such as: Pisa, SmartGarda Lake, Snap4, ALTAIR, etc.
- This double way allows them to:
  - test the news,
  - share experiences with other groups,
  - get visibility,
  - work in the collaborative environment, and
  - be better supported by Snap4City.org and DISIT Lab personnel.
- Each instance of Snap4xxxx solution ***can decide to join the federation*** of SmartCity API to exploit shared data.
  - This allows to exploit regional data for city installations applications (web, mobile, dashboards, etc.) without reloading them for example.

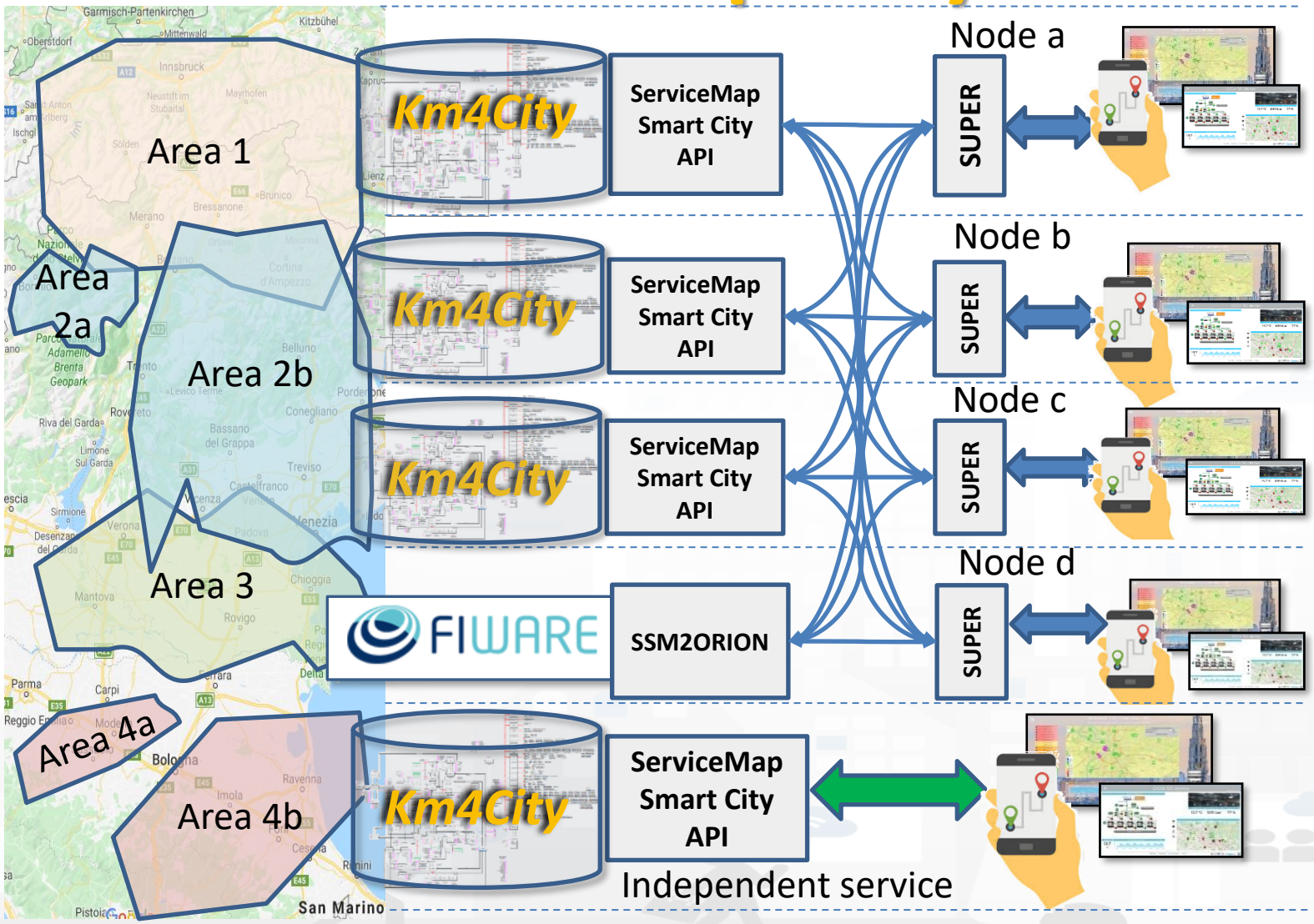
Main Organizations/areas

- Antwerp area (Be)
- Capelon (Sweden: Västerås, Eskilstuna, Karlstad)
- DISIT demo (multiple)
- Dubrovnik, Croatia
- Firenze area (I)
- Garda Lake area (I)
- Helsinki area (Fin)
- Livorno area (I)
- Lonato del Garda (I)
- Modena (I)
- Mostar, Bosnia-Herzegovina
- Pisa area (I)
- Pont du Gard, Occitanie (Fr)
- Roma (I)
- Santiago de Compostela (S)
- Sardegna Region (I)
- SmartBed (multiple)
- Toscana Region (I), SM
- Valencia (S)
- Venezia area (I)
- WestGreece area (Gr)

Snap4City (C), October 2020



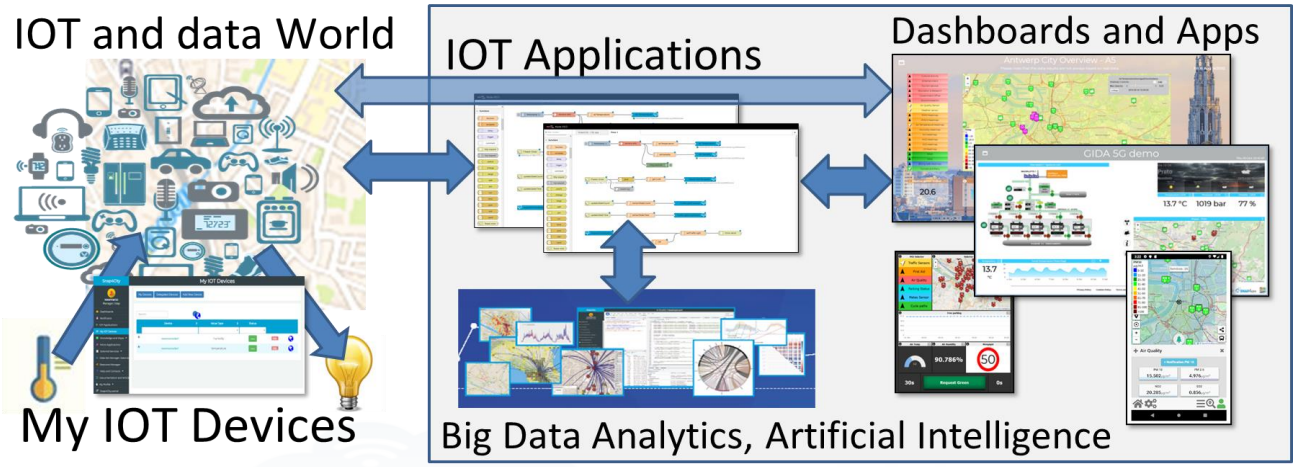
Federation of Snap4City Services



- A Mobile App may refer to one Smart City API Server (for Area 1) via SUPER and receive data from the Federated SUPERS (Area 2) if navigation, queries, etc. are leading to discover out of the addressed KB.
  - SUPER can be used for creating redundant and/or balanced distributed solutions for Federated KB. See Area 2, the two KB in the front.
  - Federated SUPER ServiceMap can have overlapped KB even totally.
  - A Mobile App can be developed to support multiple Smart City API servers, for balancing and
- The usage of Super (ServiceMap) is not mandatory so that separate services can be produced as well
- SuperServiceMap and ServiceMap presents the same Smart City APIs.

Free Trial

- Register on WWW.snap4city.org
  - Subscribe on **DISIT Organization**
- **You can:**
  - Access on basic Tools
  - Access to a large volume of Data
  - Create Dashboards
  - Create IOT Applications
  - Connect your IOT Devices
  - Exploit Tutorials and Demonstrations



IF you need to go more in deep you can ask us to pass at the next Role becoming full AreaManager with full rights of development, also for Data Analytics, machine learning, etc.

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

General Overview of the full Course

- ***1st part:*** *General Overview*
- ***2nd part:*** *Dashboards Creation and Management*
- ***3rd part:*** *IOT Applications development, IOT Devices, IOT Networks*
- ***4th part:*** *Data Analytics, in R Studio, in Python, how to Exploit and Manage Data Analytics in IOT Applications*
- ***5th part:*** *Data Ingestion, Data Warehouse, Data Gate, IOT Device Data ingestion, IOT App for Data Ingestion, etc.*
- ***6th part:*** *Snap4City Architecture, How To Install and Manage Snap4City*
- ***7th part****: Smart city API (internal and external) Web and Mobile App development tool kit*

A number of the training sections include exercitations

Updated versions on: https://www.snap4city.org/577

See also courses in ITALIANO: https://www.snap4city.org/485

Agenda



- GO • **Solution Analysis and Design**
- GO • **Data Ingestion Capabilities and Architecture**
- GO • **Data Ingestion Strategy and Tools:** (Data Inspector, DevDash, Knowledge Base)
- GO • **Upload Data Sets of POIs via IOT Applications** (Map data, KB)
- GO • **Data Ingestion via IOT Brokers**
  - IOT Directory, IOT Broker Registration, IOT Device Model, IOT Device Registration
- GO • **Real Rime Data Ingestion via IOT Applications**
- GO • **Data Ingestion via API: External Services, using HTTP MicroService on IOT Applications**
- GO • **Data Ingestion via IOT Applications towards MyKPI**
- GO • **Data Ingestion via Web Scraping**
- GO • **Data Streams from Smart City API, participatory**
- GO • **Data Streams from Mobile Devices**
- GO • **Data Streams from Dashboards**
- GO • **GIS Data Import and Export**
- GO • **Integration with CKAN, open data manager and portal**
- GO • **Social Media data collection and exploitation**
- GO • **Data Ingestion and Transformation via ETL processes (only for former versions of Snap4City)**
- GO • **Acknowledgements**

Challenges: Requests and Deductions

Public Admin.



Pub. Admin: detection of critical conditions, improving services

Mobility Operators





Tune the service, reselling data and services , prediction







Commercial: customers prediction and profiles, promotions via ads

Tourism Museums



Tune the service, prediction

API for SME

Smart City Engine

User profiling
Collective profiles
User segmentation

Services & Suggestions
Transport, Mobility, Commercial (retail), Tourism, Cultural



Personal Time Assistant
dynamic ticketing, whispers to save time and money, geoloc information, offers, etc.







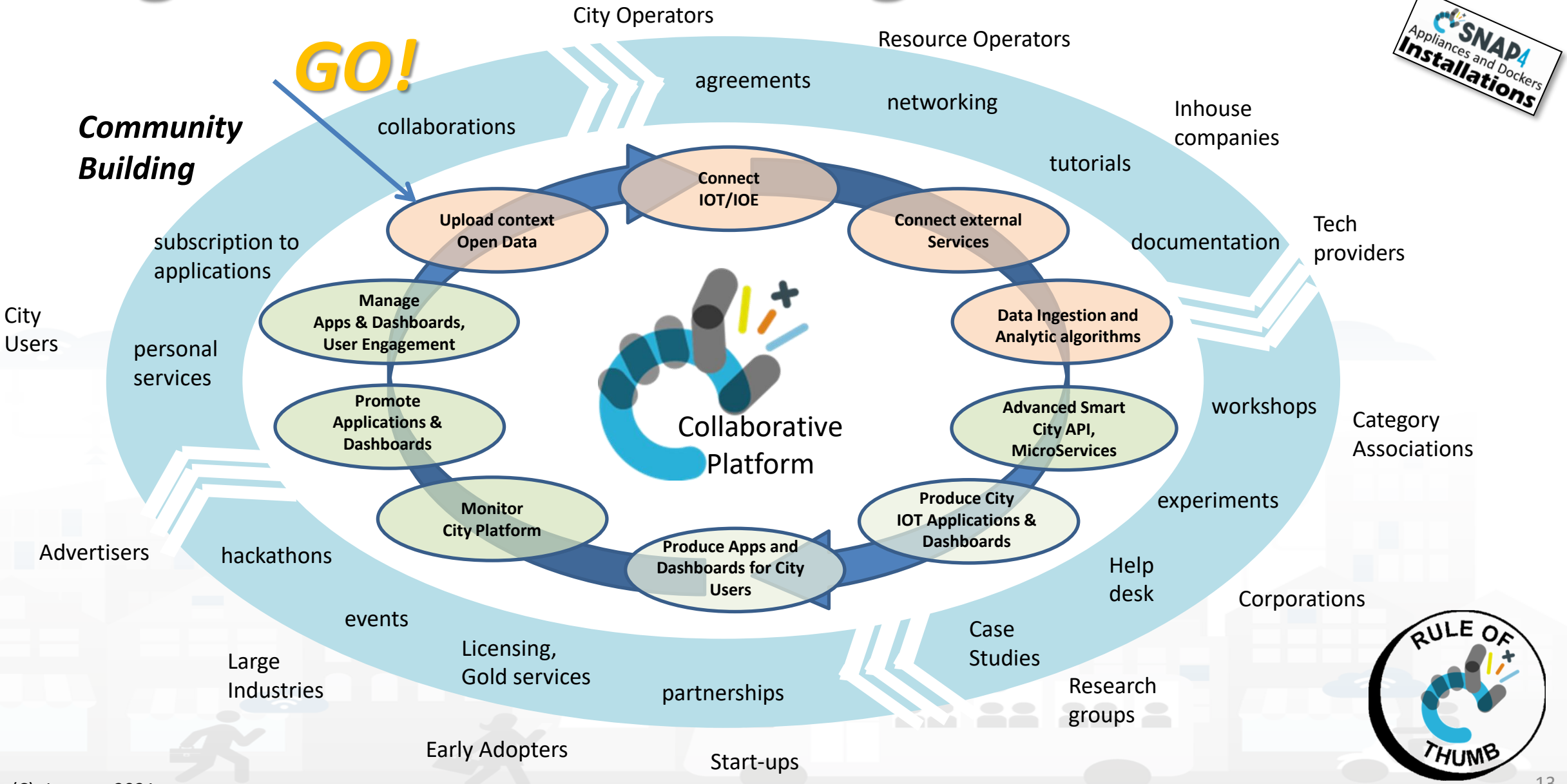
User Behavior
Crowd Sources



Data: Public and Private, Static and Real Time

Private: user movements, social media, crowd sources, commercial (retail)
Public: infomobility, traffic flow, TV cameras, flows, ambient, weather, statistic, accesses to LTZ, services, museums, point of interests, …

Living Lab Accelerating

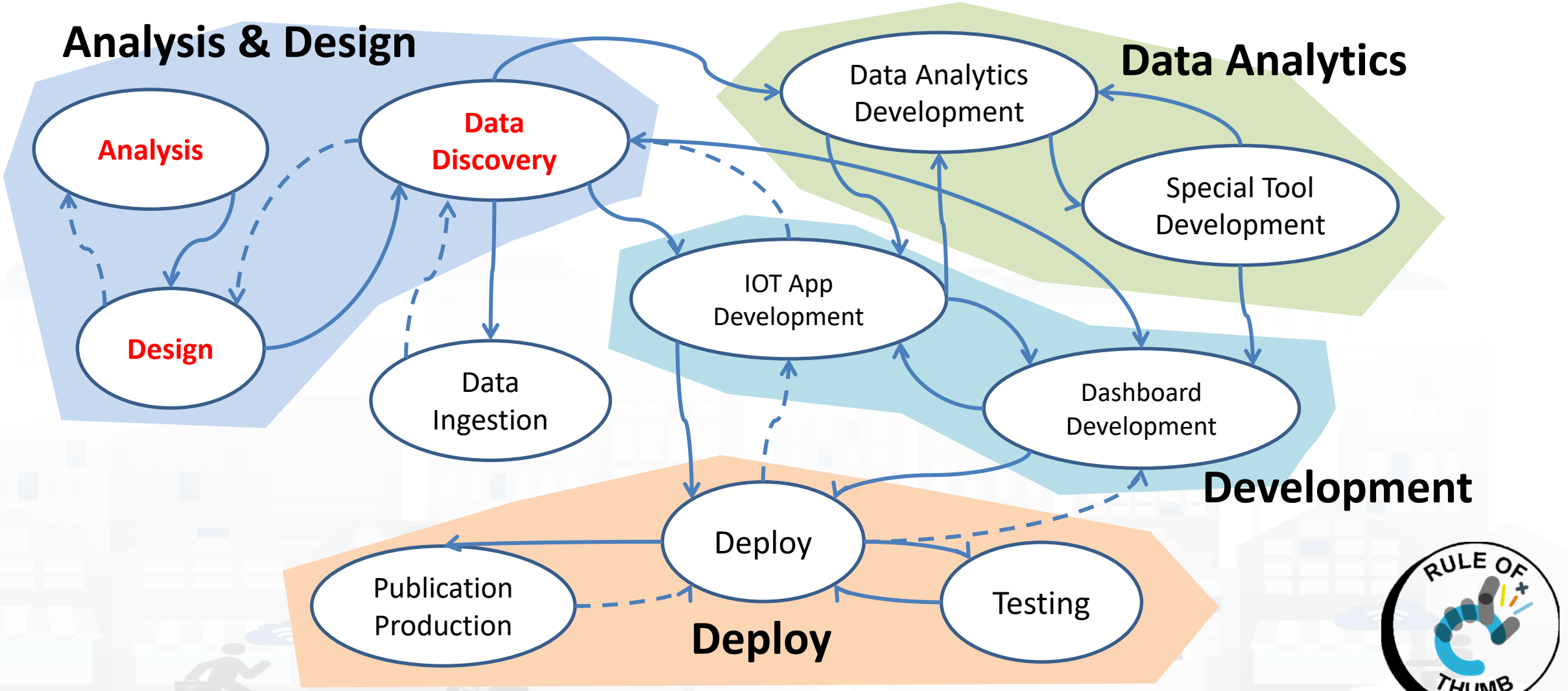


TOP

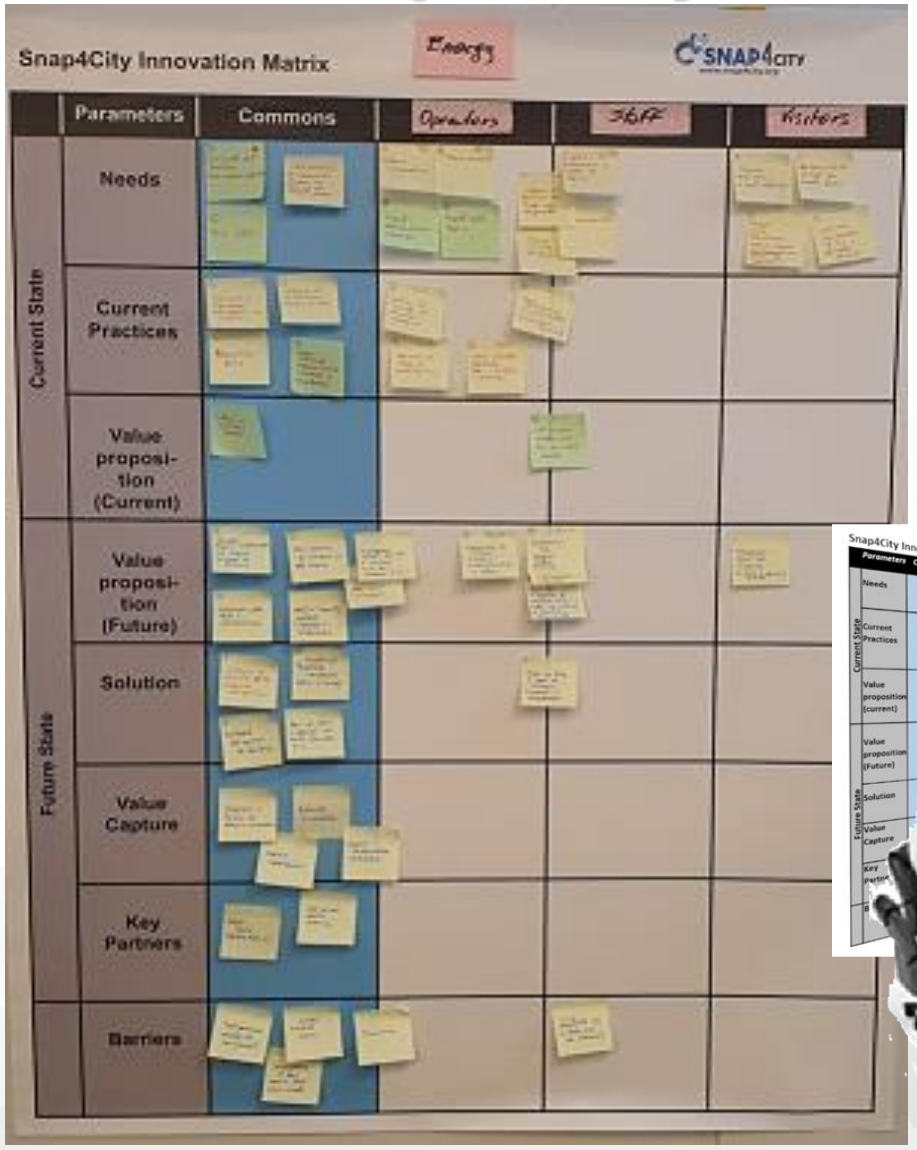
Solution Analysis and Design



Development Life Cycle Smart City Services

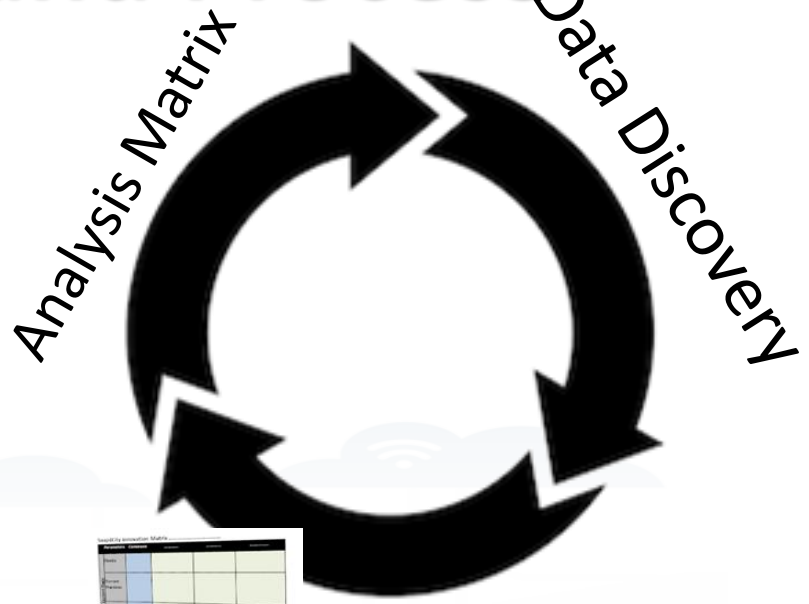


Snap4City Innovation Matrix and Process









see Course Part 6

Data vs Smart Services enabling on Snap4City

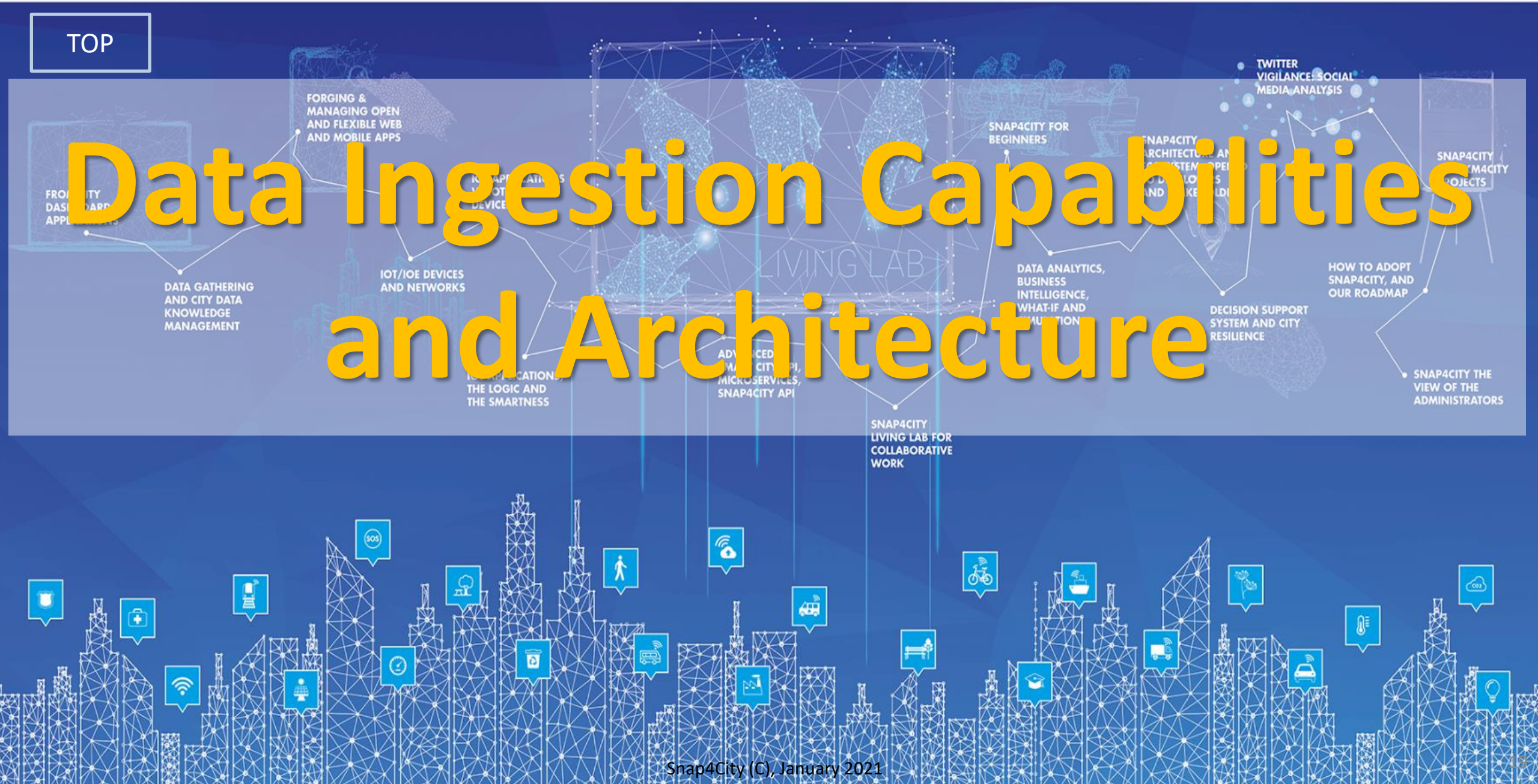
- **Public Transportation and mobility activated services in some where with Snap4City**
  - **Smart parking** (parking locations and real time parking data) … predictions
  - **Smart Fuel pricing** (fuel station locations and real time prices)
  - **Routing** (detailed GIS information, text indexing of streets, POI, etc.)
    - **Quite routing, perfect shopping, etc. etc**. (more data in needed….)
  - **multimodal routing** (detailed GIS information, Public transport time schedule)
  - **Info traffic** (traffic flow sensors, real time Traffic events, their localization, etc.)
  - **Dense info traffic** (traffic flow sensors and traffic flow reconstruction algorithm)
  - **Car/Bike/Scooter Sharing** (position and availability of Cars/Bikes, Scooters) … predictions
  - **Smart Biking** (cycling paths, environmental data) … predictions
  - **E-vehicles** (position, status of recharging stations,. …) … predictions vs booking
  - **Smart river crossing** (position and status of Underpass, Ferry) … prediction
  - **Quality of Public Transport** (actual time of arrival at the bus stops, wrt planned time schedule**)**
  - **Early Warning vs Resilience** (combination of several data including mobility, events, Social to perform early warning…)

Data vs Smart Services enabling on Snap4City

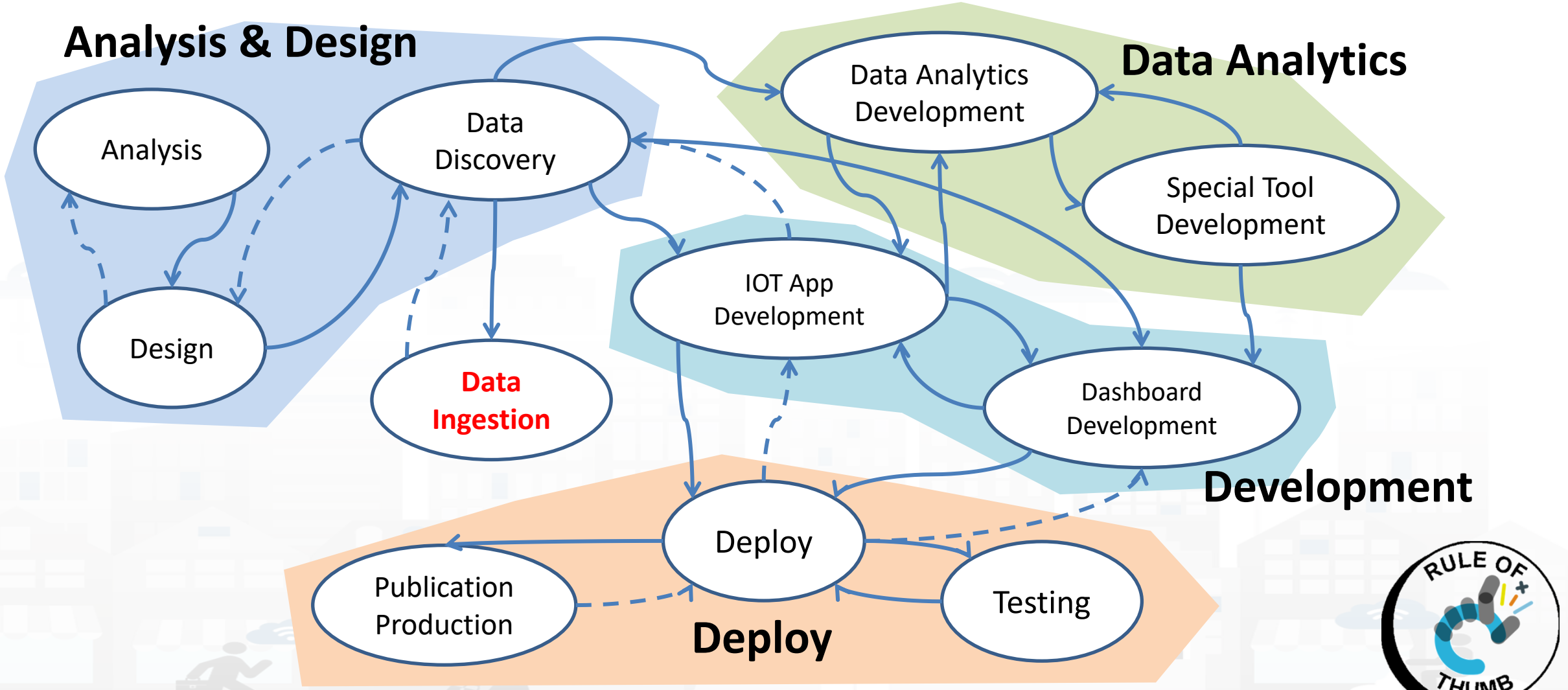
- **Social and Users Behaviour**
  - **Smart First Aid** (Location of First AID, real time status of triage)
  - **search for POI and public transport services** (POI geolocalized, spatial queries, along paths)
  - **Social Media Monitoring and acting** (Identif. of dysfunction, quality of service perceived)
  - **Information to Tourists** (Entertainment Events)
  - **Early Warning, prediction of audience** (Twitter data, social media)
  - **Improvement of services for Tourists** (people flow, usage of services)
    (Origin Destination Matrices, trajectories, heatmaps)
    (People Monitoring, via App, Wifi, PAX Counter)
    (Twitter Data, social mea,….)
- **Weather and environment, quality of life**
  - **Weather forecast/condition** (Weather forecast)
  - **Air quality Pollution** (pollution sensors, PM10, PM2.5, NOX, etc.)
  - **Pollination** (Pollination sensors)
  - **Alerting on Air quality for multiple parameters** (Prediction of parameters time slots, notification)
  - **Information Heatmaps for weather and air quality** (air quality sensors, heatmaps, prediction)
  - **Air quality indexes, and forecast** (…….

TOP

Data Ingestion Capabilities and Architecture



Development Life Cycle
Smart City Services



SNAP4CITY

KM4CITY

URBAN PLATFORM: SMART CITY IOT AS A SERVICE AND ON PREMISE

100% OPEN SOURCE

APPLIANCES CONTAINERS



- LOCAL GOVERN
- STAKEHOLDERS
- CITY USERS
- IN-HOUSE
- ENERGY OPERATORS
- MOBILITY OPERATORS
- COMMERCIAL OPERATORS
- SECURITY OPERATORS
- INDUSTRIES
- RESEARCHERS
- START-UPS
- ASSOCIATIONS

- GDPR
- SECURITY
- PRIVACY
- ASSESSMENT
- AUDITING
- PENTESTED

- OPEN IOT DEVICES
- IOT EDGE
- IOT GATEWAY
- PAX COUNTERS
- IOT BUTTONS

- TEST CASES, SCENARIOUS, VIDEOS, HACKATHONS
- OPEN SOURCES, COMMUNITY OF CITIES
- TRAINING TUTORIALS, COMMUNITY MANAGEMENT

IOT APPLICATIONS - INSTANT APPS



DATA DRIVEN APPLICATIONS • REAL TIME PROCESSING • BATCH PROCESSING • ANY PROTOCOL & FORMAT

DASHBOARDS & APPLICATIONS



CONTROL ROOM • SITUATION ROOM • OPERATOR DASHBORDS • BUSINESS INTELLIGENCE • WHAT-IF ANALYSIS • DECISION SUPPORT • SIMULATIONS • RISK ANALYSIS • RESILIENCE ANALYSIS

MOBILE & WEB APPLICATIONS



DEVELOPMENT KIT • SUGGESTIONS • MOBILE APPS • MONITORING PANELS • PLATFORM UTILITIES • READY TO USE SMART APPLICATIONS

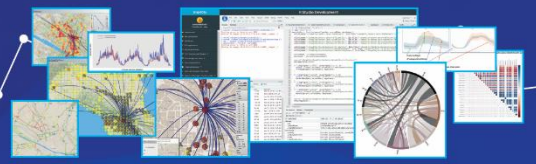
MICROSERVICES & ADVANCED SMART CITY API

LIVING LAB - DEV TOOLS - COWORKING

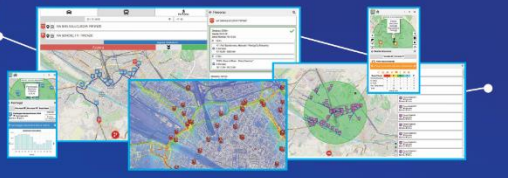


IOT DIRECTORY • SERVICE MAP • RESOURCE MANAGER • DATA GATE • R STUDIO • ETL

BIG DATA - DATA ANALYTICS



DATA ANALYTICS TOOLS - MICRO-APPLICATIONS



PREDICTIONS • ANOMALY DETECTION • WHAT-IF ANALYSIS • TRAFFIC FLOW RECONSTRUCTION • ORIGIN-DESTINATION MATRICES • SOCIAL MEDIA ANALYSIS • OFFER VS DEMAND ANALYSIS • ENVIRONMENTAL DATA ANALYSIS AND PREDICTIONS • REAL TIME HEATMAPS • ROUTING • ALERTING • EARLY WARNING • PERSONAL AND VIRTUAL ASSISSTANTS • SMART SOLUTIONS • SMART SHARING • PARTECIPATORY

KM4CITY DATA AGGREGAT KNOWLEDGE BASE - EXPERT SYSTEM OF THE CITY – BIG DATA STORE

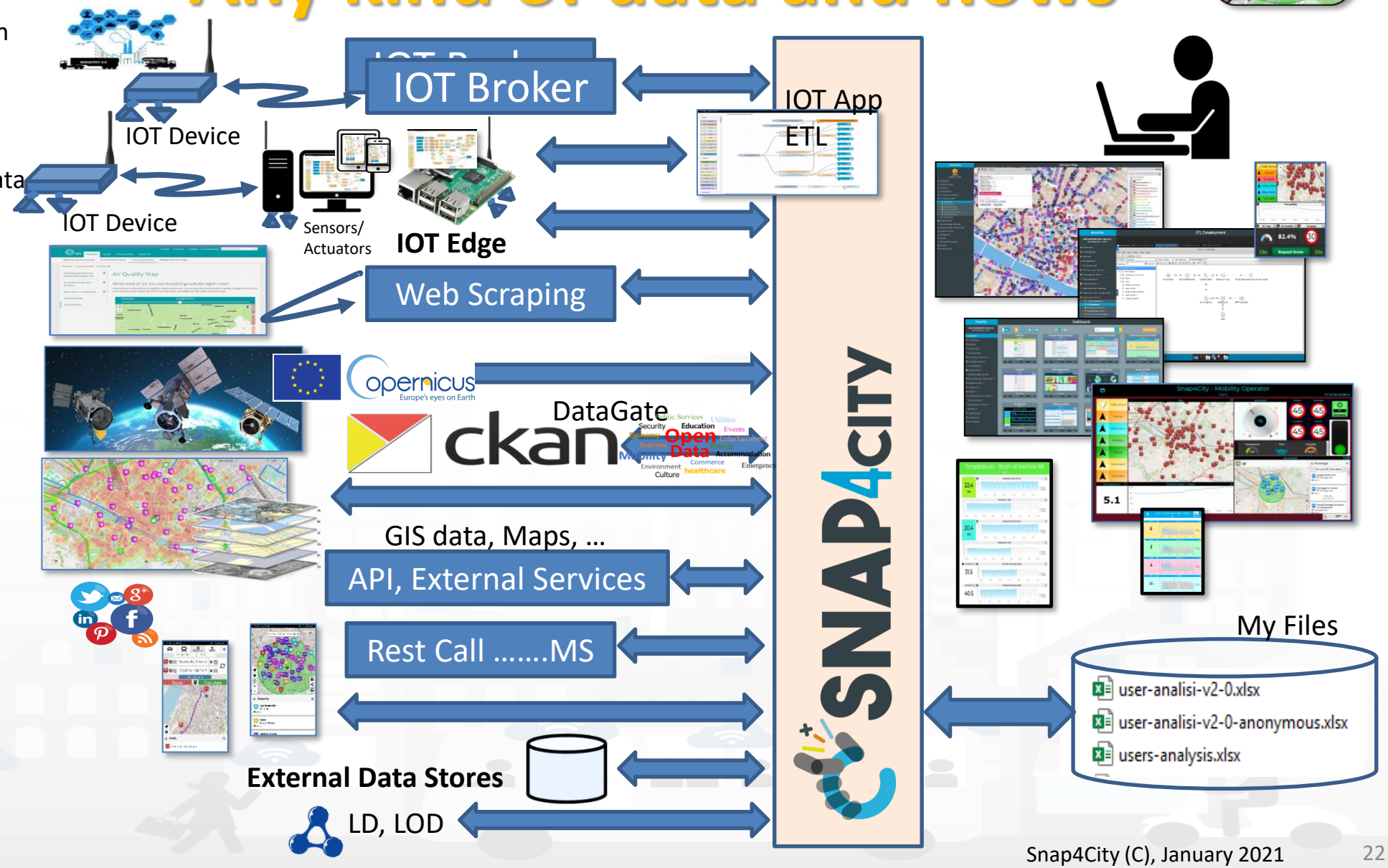
IOT MNG - DATA MNG - DATA INSPECTOR – PROCESS MNG – USER ENGAGEMENT – GDPR MNG ...

GIS	CITY UTILITIES	OPEN DATA	LEGACY & EXTERNAL SERVICES	PERSONAL DATA	IOT / IOE	BROKERS	KPI	INDUSTRY 4.0	SOCIAL MEDIA



Any kind of data and flows

- **Open Data:**
  - Data gate, federation of Open Data Portals
  - IOT App, ETL proc(PULL)
- **IOT Networks:**
  - IOT Application processes, data driven or PULL
  - IOT Brokers (Push) → IOT Shadow
- **Web Pages:**
  - Web scraping, crawling processes
- **Satellite data**
- **Social media: Twitter, Facebook,..**
  - Twitter Vigilance, IOT App
- **Mobile Apps**
  - Smart City API
- **Files upload: CSV, Excel, etc.**
  - IOT Applications, ETL
- **REST API, WS, FTP, LD, LOD, etc.**
  - IOT Applications, ETL
- **Data base accesses**
  - GIS: WFS, WMS
  - ETL, IOT Application



Standards and Interoperability

Compliant with: AMQP, COAP, MQTT, OneM2M, HTTP, HTTPS, TLS, Rest Call, SMTP, TCP, UDP, NGSI, LoRa, LoRaWan, TheThingsNetwork, SigFOX, DATEX II, SOAP, WSDL, Twitter, FaceBook, Telegram, SMS, OLAP, MySQL, Mongo, HBASE, SOLR, SPARQL, EMAIL, FTP, FTPS, WebSocket, WebSocket Secure, ModBUS, OPC, GML, RS485, RS232, WFS, WMS, ODBC, JDBC, Elastic Search, Phoenix, XML, JSON, CSV, db, GeoJSON, Enfuser FMI, Android, Raspberry Pi, Local File System, ESP32, Libelium, IBIMET/IBE, OBD2, SVG, XLS, XLSX, TXT, HTML, CSS, KNX, Enocean, Zigbee, DALI, ISEMC, Alexa, Sonoff, HUE Philips, Tplink, BACnet, TALQ, Copernicus, IFC, etc. https://www.snap4city.org/65

Snap4City vs Formats

- Snap4City is capable to ingest and work with any format:
  - Data **exchange**: JSON, GeoJSON, XML, HTML, HTML5, DATEX, **GTFS**, binary, etc.
  - **GIS formats:** WMF, WFS, heatmaps, ….
  - **Table**: CSV, XLSX, XLS, database, …
  - **archive** file formats: zip, rar, 7z, tgz, …
  - **image** formats: png, gif, tiff, ico, jpg, …
  - **video** formats: mp4, avi, mov, …
- Search the format you need to cope on the search box of Snap4City portal!: **Snap4City Supported Protocols, adding new protocols**

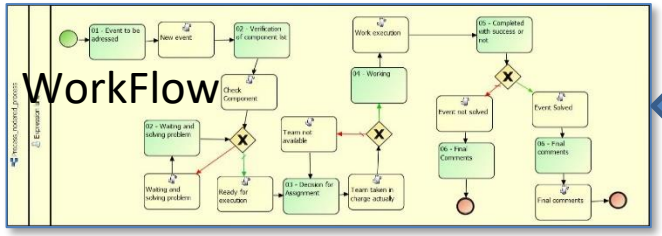
Data Inspector (Digital Twin info) Major Submodels

- **Digital Twin**
  - Device and sensors data
  - Values
  - Healthiness criteria and values
    - Machine learning tools
  - Images and physical world
  - Licensing
  - Users
- **Users**
  - Defined the Data and Devices
  - Defined the processes
  - Create dashboards
  - Etc.
- **Process Views**
  - Device Management tool
  - Data ingestion processes
    - ETL, IOT Apps
  - Data storage access views
    - Index views
    - Relationships view
  - Data Analytics and Transformation
    - IOT App, R Studio, Python
  - Data Rendering Dashboards
    - Synoptics
  - Processes' Developers

KPI, POI, MyKPI, …

API, External Services

Web Scraping





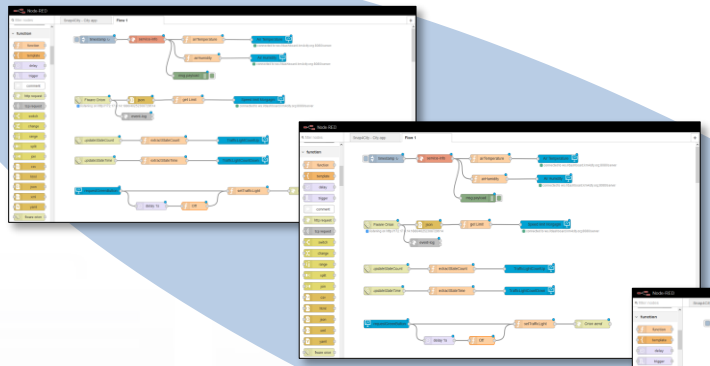


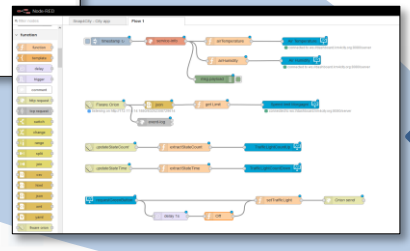


Data Analytics,
Artificial Intelligence





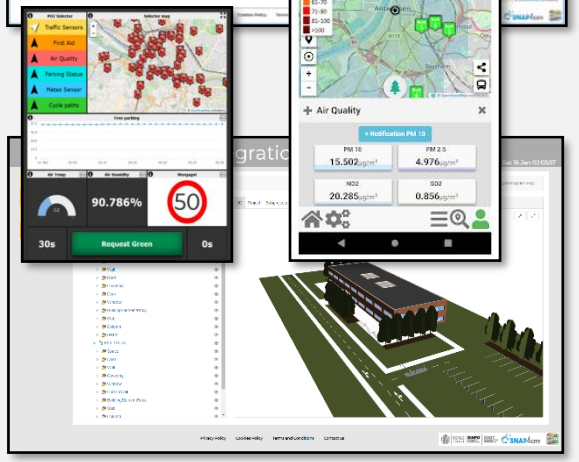




IOT Apps

IOT Broker





ckan

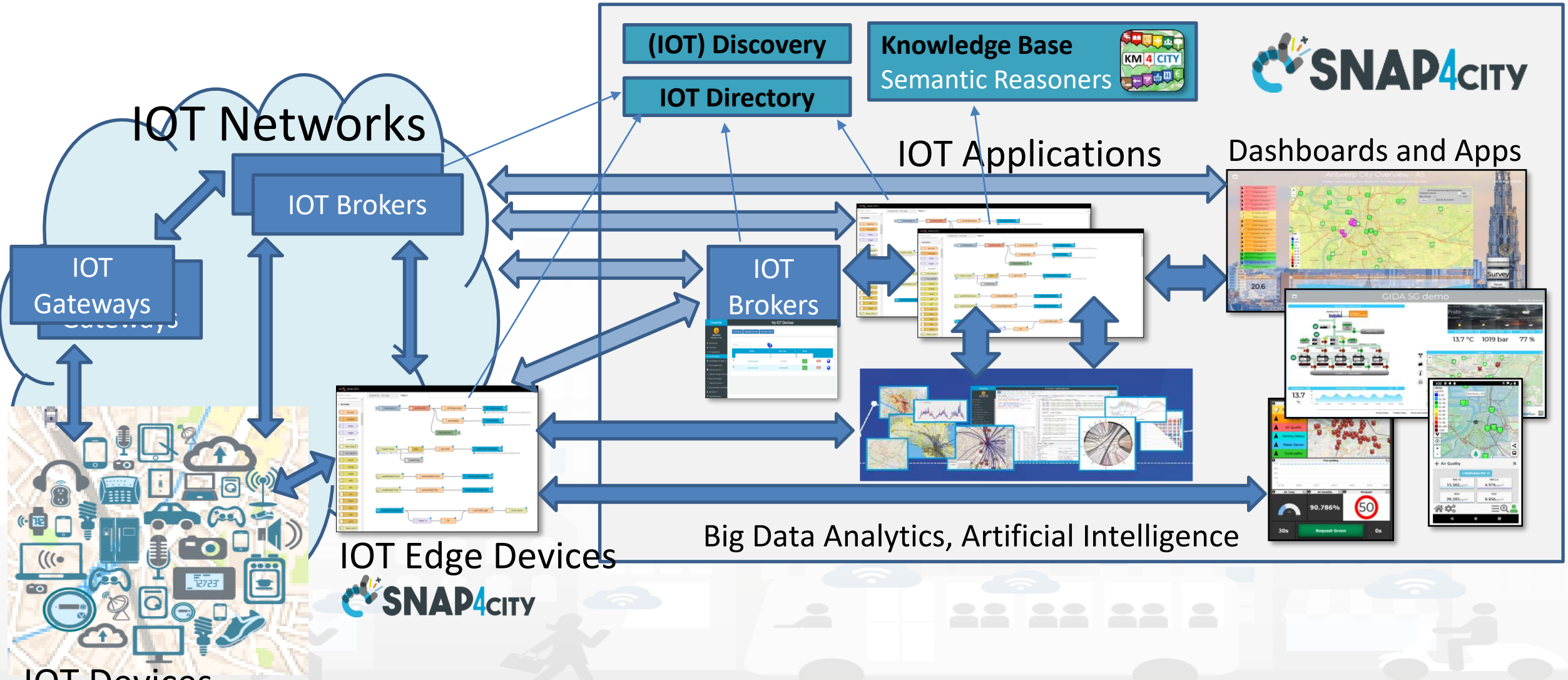
LD, LOD



Big Data

Dashboards and Apps

Snap4City: IOT Directory and data/device Discovery



The usage of IOT Applications

- IOT Applications = Node-RED + Snap4City Libraries
- Used for:
  - **Data Ingestion, Transformation, Extract, Load, and Adaptation** (format and protocol), See Part 5 of the Course (this part)
  - **IOT Edge Devices logic**, for implementing logic on IOT Edge, including **IOT Device control** (see on Part 3 of the course)
  - **Business Logic** control of Dashboards, via Web Sockets secure
    - see Part 2 of the Course
  - **Control and schedule of Data Analytic**, and Machine Learning (see part 4 of the Course)
  - **Firing and condition identification and alerting**.

Further readings

- 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: produce heatmaps, custom heatmaps on any data

Further readings

In Yellow alternative & legacy solutions

- HOW TO: add a device to the Snap4City Platform
- HOW TO: add data sources to the Snap4City Platform
- HOW TO: add IOT Device data source from external broker to the platform.
- TC9.13: How to upload a local file into your IOT Application
- TC9.16 Web Scraping to get data from web pages
- TC6.3. Creating ETL processes for automated data ingestion and data transformation
- TC9.2. Managing heterogeneous File Ingestion, protocols, formats via IOT applications, and open standards
- TC6.1. Managing DataSets via DataGate: ingest, search, download, upload, annotate, share
- TC6.2. Search on DataGate for Data Sets
- TC2.25. Registering external MicroService calling RestCall services, using it on IOT applications

Data Gathering and Knowledge Management

- Data ingestion can be performed by using multiple tools:
  - We suggest:
    - IOT App for static and real time data and flow, from IOT Brokers/Devices
    - DataGate for Static Data, upload them as files, or collected from other CKAN
    - WebScraper for scraping data from Web Pages, when authorized!
    - ETL for static / periodic data in PULL

In Yellow alternative & legacy solutions

- See how to test cases:
  - **HOW TO: add data sources to the Snap4City Platform**
  - **HOW TO: define privacy rules for personal data, produced by the end-users own device**
  - **US6. Developing and using processes for data transformation**
  - **TC6.1 - Managing DataSets via DataGate: ingest, search, download, upload, annotate, share**
  - **TC6.3 - Creating ETL processes for automated data ingestion and data transformation**
  - **TC6.5 - Managing Heterogeneous File Ingestion via ETL processes**
  - **TC6.9 - ETL processes for multiprotocol and format data ingestion, see on GITHUB for library**
  - **TC9.2 - Managing heterogeneous File Ingestion, protocols, formats via IOT applications, and open standards**

Data Ingestion Architecture and Kinds



Snap4City, Snap4Industry Architecture, V2 (2020)







Back-End

Front-End

Data Sources, External Services *PULL Data*

Data Sources, Brokers, External Services *Data Driven, Real Time*

Data Ingestion, aggregation, regularization, reconcile: IOT Directory, NIFI, special tools

Knowledge base
Semantic Reasoners

Indexing and aggregating
Elastic Search

Federation

Search and Query, Smart City API, Web Socket Server, GIS, Facet, semantic

Data Analytics, Simulations, Special Tools
R Studio, Tensor Flow, Python, ...

IOT Applications, Business Logic
Node-RED + Snap4City MicroServices

Rendering Acting, Widgets, Synoptics, MicroApps
User interface, Drill down, maps, heatmaps



Inform, announce, Act!, warning, alarms, What-IF, ..

Authentication, Authorization, Platform & Processes Management, Data Inspector, Digital Twin, ...

Smart City Functional Architecture

Transport systems
Mobility, parking



Public Services,
Govern, events, …



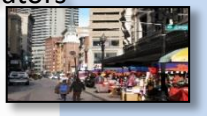
Sensors, IOT Cameras,
Wi-Fi



Environment, Water,
energy



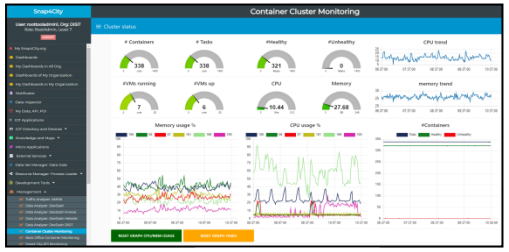
Shops, services,
operators



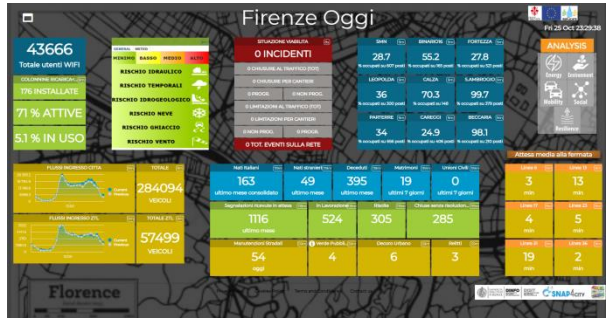
Social Media

Social Media Crawler and Manager









Back-End ← → Front-End

Data Sources, External Services *PULL Data*

Data Sources, Brokers, External Services *Data Driven, Real Time*

Data Ingestion, aggregation, regularizatio n, reconcile: IOT Directory, NIFI, special tools

Knowledge base Semantic Reasoners

Indexing and aggregating Elastic Search

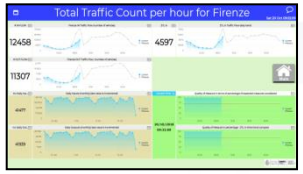
Federation

Search and Query, Smart City API, Web Socket Server, GIS, Facet, semantic

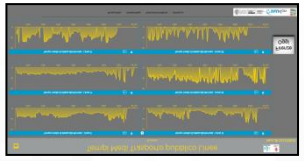
Data Analytics, Simulations, Special Tools R Studio, Tensor Flow, Python, ….

IOT Applications, Business Logic Node-RED + Snap4City MicroServices

Rendering Acting, Widgets, Synoptics, MicroApps User interface, Drill down, maps, heatmaps

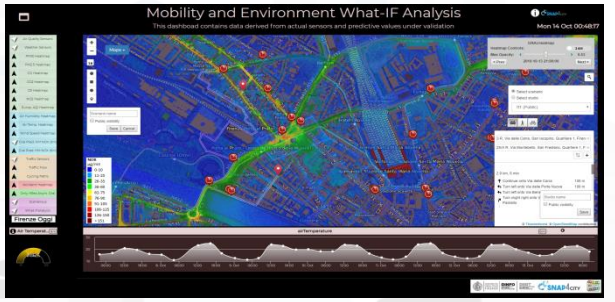












Inform, announce, Act!, warning, alarms, What-IF, ..

Authentication, Authorization, Platform & Processes Management, Data Inspector, Digital Twin, …

Modalities and Strategies for data ingestion

- **Road Graphs**: from GIS, and/or OSM (see Snap4City tool for that), …
- **Data** of any format via any protocol
- **Structured and non-structured** data (tables and free text, mixt)
- **Static data/**metadata descriptors ingested as: a file or manually via user interface:
  - IOT App from file:https://iot-app.snap4city.org/nodered/nr9xtwc/simple
  - IOT Directory from user interface, ingesting them as Sensors
  - IOT Directory from file, ingesting them as a set of Sensors
  - Manually as MyPOIs which have to be transformed in POIs by the Administrator
  - Data Gate, a module of CKAN, allows to process files as POI, Sensors, etc.
  - ETL Process from file which can be used for ingesting any kind of file
- **Simple data and complex data as:** GIS data, heatmaps, tracks, etc.
- **Real Time Event Driven data** can be ingested by:
  - Automatically by NIFI for the Data Shadow construction, or
  - Producing an IOT App, a WebScraping process, a special tool, an ETL, etc.

Static vs Dynamic Information

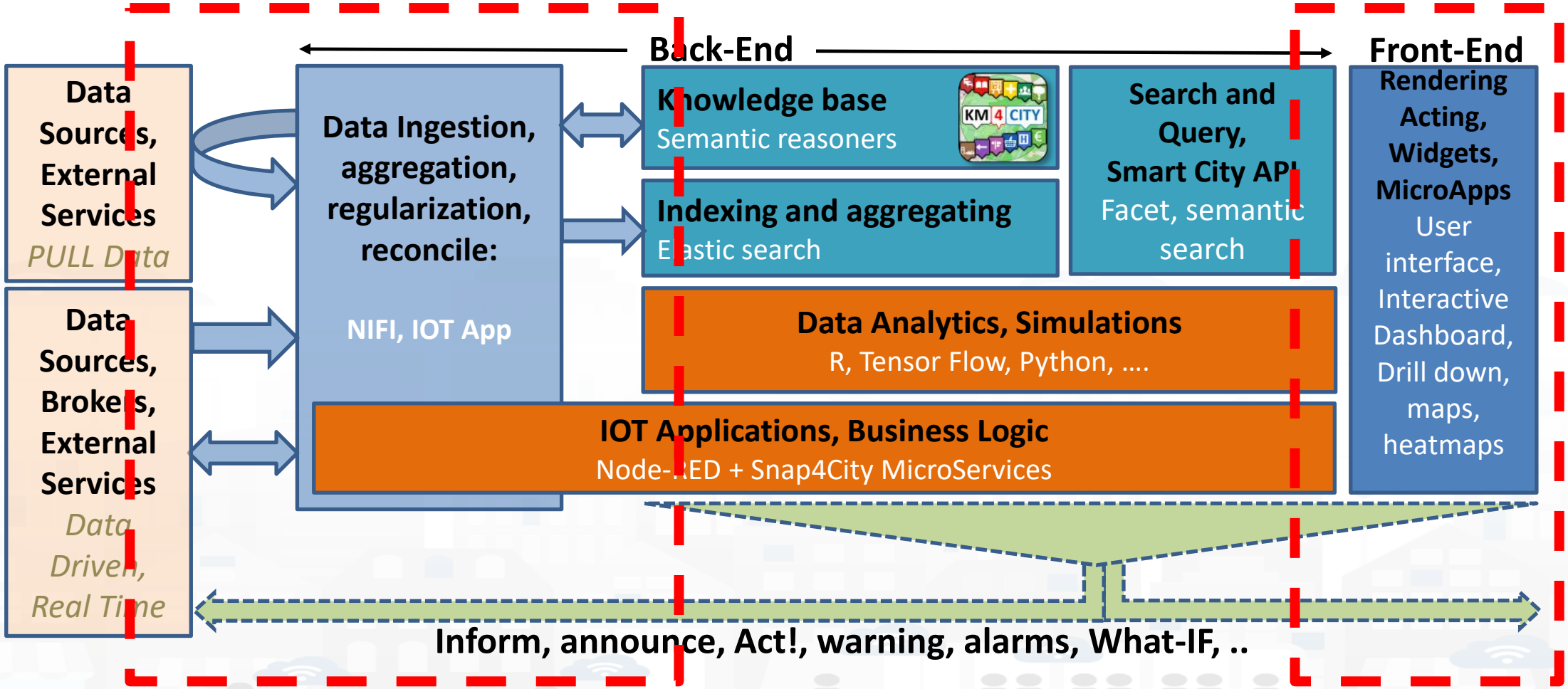
- Most of the valuable data are **Real Time/Dynamic data**, based on **Static** info.
- **For example**, data coming from Smart Park or a Smart Light solution
  - They are composed of a set of Devices,
    - the **Collection of Devices** has in addition common information such as: licensing, gateway access, gateway credentials, description of provider, time of update, etc.
    - **each Device** has **Static** information such as:
      - data structure: device ID, Current consumption, temperature, ……
      - GPS coordinates… (if the devices is not a moving one)
      - Classification: nature, subnature, Healthiness criteria, MTTF, etc.
    - **Each Device** produces a data package/message according to minimum data Structure as **Dynamic** information, in PUSH towards the Gateway.
  - The Gateway in turn can be:
    - a sort of Broker sending data to other consumer in Push or
    - can provide also data in PULL when requested by some consumer.
- This means that data fully describing the situation is composed by **Static** and **Dynamic** aspects that have to described into the platform and reconnected when one access to the device…

What happen to data into the platforms

- **Static** information has to be collected in advance
  - For each **collection of devices**, for each **device**
- **Dynamic** information has to connected to Static one at the ingestion time

- **PLUS**: when data enter into the platform a wide number of information is added/created such as:
  - Internal technical identification, modalities to access to the information
  - Historical data storage
  - Eventual derived data: date and time of ingestion, average, quality level, etc.
  - Rights to access at the data: who can access and to do what ?
  - Visual representation of data, may be an image of the Device how it has been installed and where, by who, telephone number for maintenance, etc.
  - Process ID used to ingest the data, possible IOT Applications that are using the data, …

Snap4City Architecture vs Data Ingestion



Data are **not** so Simple as one can imagine

- Data to be managed into the Smart City IOT are not so simple as one may imagine, and not limited to take into account only IOT Devices.
- THUS, a large number of data TYPES and sources have also to be addressed:
  - E.g.: external services, heatmaps, trajectories, maps, OD matrices, actuators, personal data, KPI, API descriptors, special widgets, events, predictions, Tweets, posts, GIS, mobile devices, etc.
  - With their complexity of managing data, licensing, etc…
- THEY are called **High Level Types: how and which tool / process can cope with them into the Smart City Platform?... See next!**
  - **Thus a unified model is needed.**
- **In 2nd part we have shown how to show specific types of data**

How to Ingest

All of them can be shown on Dashboards, what about manipulate them!!!!

HLT, High Level Types++	GPS	Static	Dynamic	Single	Time Series	Trajectory	HTTP	How to ingest/change/manage/see
POI (Point of Interest)	Yes	Yes		Yes				DataGate, ETL, IOT App, API
MyPOI data	Yes	Yes		Yes				Dashboard, IOT App, UserInterface, API
Sensor data	Yes	Yes	Yes	Yes	Yes			Dashboard, IOT Directory, IOT App, UserInterface, API
Sensor Actuator data	Yes	Yes	Yes	Yes	Yes			Dashboard, IOT App, UserInterface, API
MyKPI data	Yes	Yes	Yes	Yes	Yes	Yes		Dashboard, IOT App, UserInterface, API
Dashboard-IOT App (msg)		Yes	Yes	Yes	Yes			Dashboard, IOT App, API
Dashboard-IOT App real time			Yes	Yes	WS			Dashboard, IOT App, API WS
Synoptics data		Yes	Yes	Yes	Yes			Dashboard, IOT App, UserInterface
My Personal Data		Yes	Yes	Yes	Yes			Dashboard, IOT App, UserInterface, API
KPI (metrics) data		Yes	Yes	Yes	Yes			Dashboard, IOT App, API, SQL calls x Metrics
Special Widget (complex)		Yes	Yes	Yes	Yes		Yes	Dashboard, ETL, special, IOT App, API
Complex Event (msg)	Yes	Yes	Yes	Yes	Yes			Dashboard, ETL, special, IOT App, API
WFS/WMS (GIS data)	Yes	[yes]	[yes]				Yes	Dashboard, GIS tools, or GeoServer, IOT App
GTFS	Yes	Yes	Yes	Yes	Yes			ETL, special [IOT App], MicroApplications
OD Matrices	Yes	Yes	Yes	Yes	Yes			Special tools, MicroApplications

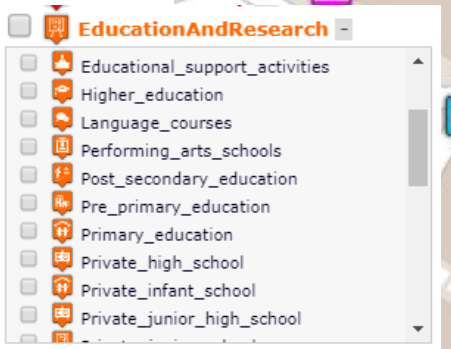
POI: Point of Interest vs Classification

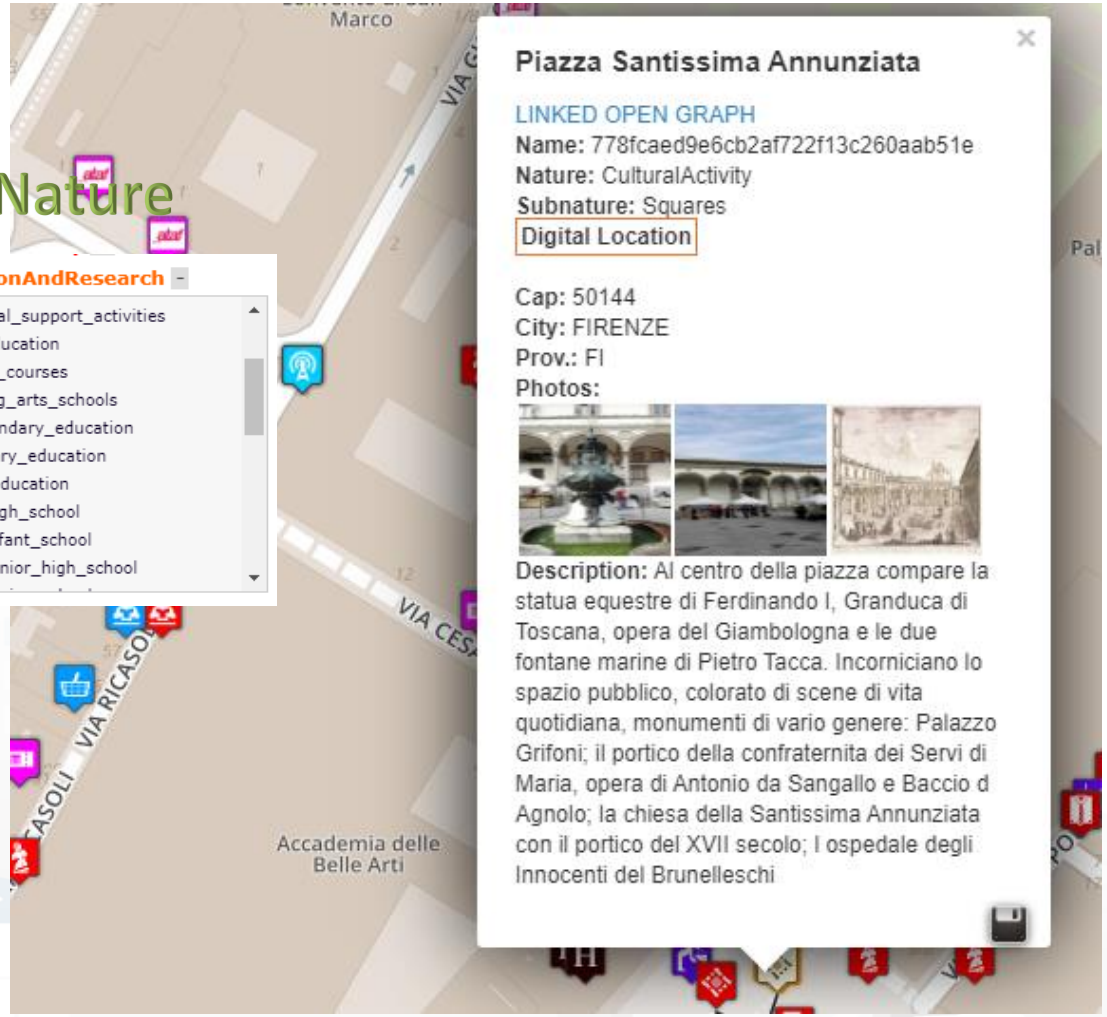
- A POI is defined as an **element of a set (collection)** and with general info:
  - **Nature**: ……..
  - **Subnature**: ….
- Specific infor for each POI
  - **Location**: lat, lon
  - A set of **Attributes**
    - www, email, opening time, phone, cap, address, city, etc.
  - Eventually a link to data

Nature

- Accommodation
- Advertising
- AgricultureAndLivestock
- CivilAndEdilEngineering
- CulturalActivity
- EducationAndResearch
- Emergency
- Entertainment
- Environment
- FinancialService
- GovernmentOffice
- HealthCare
- IndustryAndManufacturing
- IoTDevice
- MiningAndQuarrying
- ShoppingAndService
- TourismService
- TransferServiceAndRenting
- UtilitiesAndSupply
- Wholesale
- WineAndFood

SubNature





MyKPI and Sensors: examples

- **HLT**: MyKPI
- **Nature**: Industry and manufacturing
- **Subnature**: Chemical
- **Value Type**: Density percentage
- **Value Name**: CloroParaffine
- **Data Type**: float mykpi
- **Value Unit**: %
- **Last Date**: 2019-02-25
- **Last Value**: 87.0
- **Healthiness**:
- **Last Check**: 2020-04-03 10:28:12
- **Ownership**: private
- **Organization**: Firenze

- HLT: Sensor
- Nature: From IOT Device to KB
- Subnature: IOT Sensor
- Value Type: Battery Level
- Value name: Irrigator fioriera Gag
- Data Type: float
- Value Unit: V
- Last Date: 2020-04-01 12:59:00
- Last Value: 5.18
- Healthiness:
- Last Check: 2020-04-03 03:28:12
- Ownership: public
- Organization: Firenze

MyKPI vs IOT Device Sensor

- **MyKPI** can be grouped in a Groups of Values. They may have GPS changing
- **Sensors** can be also grouped, but born directly grouped into an **IOT Device** as defined into the IOT Directory. They have GPS static
- Please note that the "dateObserved" has to be in standard ISO to have Time Series

IOT Device may be sending data as:

```
{  "id":"dev45",
   "type":"misura",
   "attributes":[
     { "name":"dateObserved",
       "value":"2020-04-06T17:00:00.000Z",
       "type":"time"   },
     { "name":"temp 1",      "value":28.976,      "type":"float"    },
     { "name":"Hum 1",       "value":35976,       "type":"float"    },
.....
     { "name":"kevi",        "value":721732,      "type":"integer"  }
   ]
}
```

How to Ingest

All of them can be shown on Dashboards, what about manipulate them!!!!

HLT, High Level Types++	GPS	Static	Dynamic	Single	Time Series	Trajectory	HTTP	How to ingest/change/manage
API (Ext. Srv., any prot.)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	ETL, Special, IOT App, …
External Service (web pag)		Yes	--				Yes	ETL, Special, IOT App, Web Scraper, …
MicroApplication (webapp)		Yes	--				Yes	Dashboard, IOT App, API, FTP, …
Heatmap matrix	Yes	Yes	Yes	Yes	Yes			Maps, IOT App, MicroService, UserInterface, API
Synoptics (group)		Yes	Yes	(Yes)	(Yes)		Yes	Dashboard, Special Tools, IOT App, API, …
Special Tools (functional)	(Yes)		(Yes)	(Yes)		(Yes)	Yes	As MyPersonalData, …
Typical Trends (not yet)	(yes)	Yes	(Yes)		Yes		Yes	MicroApp, Special tools, (API) , …

Non HLT

Traffic Flows (are coming)	(yes)	Yes	Yes		Yes	Yes	Yes	Maps, Special tools, API, …
Color Maps		Yes		Yes				Maps, Tables, Special tool, User Interface, API
GTFS (see Sensors, POI)	Yes	Yes	Yes		Yes	Yes		Maps, Special tools, API, …
Typical Trajectory (MyKPI)	Yes	Yes	Yes		Yes	Yes		Maps, Special tools, API, …

- Now, it is more clear about what we intend as:
  - ***High Level Types***

Unified Data and Services Model/Classification

MyKPI	All selected (8)	All selected (15)	All selected (30)		All selected (3)	6 selected			true		All selected (2)	
High-Level Type	Nature	Subnature	Value Type	Value Name	Data Type	Value Unit	Last Date	Last Value	Healthiness	Last Check	Ownership	Organizations
MyKPI	Environment	Weather_sensor	Gradi Centigradi	BatteryTemperatureGalaxyTabS4	float-mykpi	°C	2019-11-21 15:51:25	28		2020-04-03 10:23:02	private	Firenze,dc=ldap,dc=disit,dc=org,
MyKPI	Environment	Weather_sensor	Gradi Centigradi	BatteryTemperatureS6Edge	float-mykpi	°C	2020-01-23 12:37:34.25	299999237060547		2020-04-03 10:23:02	private	DISIT
MyKPI	Environment	Weather_sensor	Gradi Centigradi	BatteryTemperatureSamsungS9	float-mykpi	°C	2019-04-23 05:54:21.24	399999618530273		2020-04-03 10:23:02	private	Firenze,dc=ldap,dc=disit,dc=org,
MyKPI	Environment	Smart_waste_container	Level_percentage	Bin_001	percentage-mykpi	%	1970-01-01 01:00:00			2020-04-03 10:23:02	private	DISIT,dc=ldap,dc=disit,dc=org,
MyKPI	Environment	People_counter	Battery Percentage	borrowed_tbeam_pax_green BATTERY	percentage-mykpi	%	2019-03-13 20:24:52	4.202		2020-04-03 10:23:02	private	Firenze,dc=ldap,dc=disit,dc=org,
MyKPI	Environment	People_counter	Battery percentage	borrowed_ttgov21new_pax_black BATTERY	percentage-mykpi	%	2019-05-06 10:03:55	4.132		2020-04-03 10:23:02	private	Firenze,dc=ldap,dc=disit,dc=org,
MyKPI	Environment	People_counter	Battery Percentage	borrowed_ttgov21new_pax_green BATTERY	percentage-mykpi	%	2019-11-19 08:56:56	4.072		2020-04-03 10:23:02	private	Firenze,dc=ldap,dc=disit,dc=org,
MyKPI	HealthCare	Health_district	tassodicrescita	corkpimemilia	float-mykpi	%	1970-01-01 01:00:00			2020-03-24 13:13:24	private	DISIT

Hide columns | Reset filters | Selected rows: 0 | Previous 1 2 3 4 5 ... 35 Next | Search

High Level Types

Nature

SubNature

Value Type

Value Name

Data Type

Value Unit

Last Date/Time

Last Value

Healthiness

Last Check

Ownership

Organization

Semantic

Technical meaning

RealTime Status

For Admin

Legenda: How to ingest/change/manage

- **Dashboard**: by creating a Dashboard that can act/change the values with actuators (2nd Day Slides)
- **API**: you can use the Smart City API to change / provide the values (see 3rd Day Slides)
- **IOT App**: by developing an IOT Application on Node-RED exploiting Snap4City MicroServices (4th Day Slide, and in part in these slides)

- **DataGate**: you can use the DataGate tool to ingest the Static data (mainly Open), and publish them
- **ETL**: by developing an ETL process, and put it in execution via DISCES
- **IOT Directory**: you can use the IOT Directory tool to change the parameters, and set up the ingestion process, via IOT Brokers, IOT Devices, IOT Edge.
- **Special**: by using a special tool for developing a process, or for creating SVG Synoptics
- **UserInterf**: there is a number of Tools with Graphic User Interface that you can use to change the values, see in the menu on the left.
- **Web Scraper**: by creating a Web Scraping process and exploiting the results into an IOT Application

- **As MyPersonalData**: they are substantialy MyPersonalData
- **From third party tools**: they can be manipulated by using third party tools

IOT App vs ETL

IOT Applications

- Created on browser
- A huge number of Protocols https://www.snap4city.org/65
- Scheduled internally and managed as Container
- Largely diffused approach as Node-RED
- Large number of Snap4City MicroServices, well documented
- PUSH and PULL models
- Simple mechanism to add new Features
- Very diffused in IOT
- Static and Dynamic data models depending on IOT Broker capabilities
- Scalable on Cloud
- Also present in IOT Edge devices

ETL processes

- Created with Spoon editor on VM (on premise or via remote access to VM)
- A Large number of protocols https://www.snap4city.org/65
- Scheduled by DISCES in the back office
- Well known data warehouse model
- Well documented for the process
- Only PUSH models
- Complex mechanism to add a new functionality
- Very diffused in Data transformation
- Static and Dynamic data models well linked ..

Data Ingestion Methods Comparison

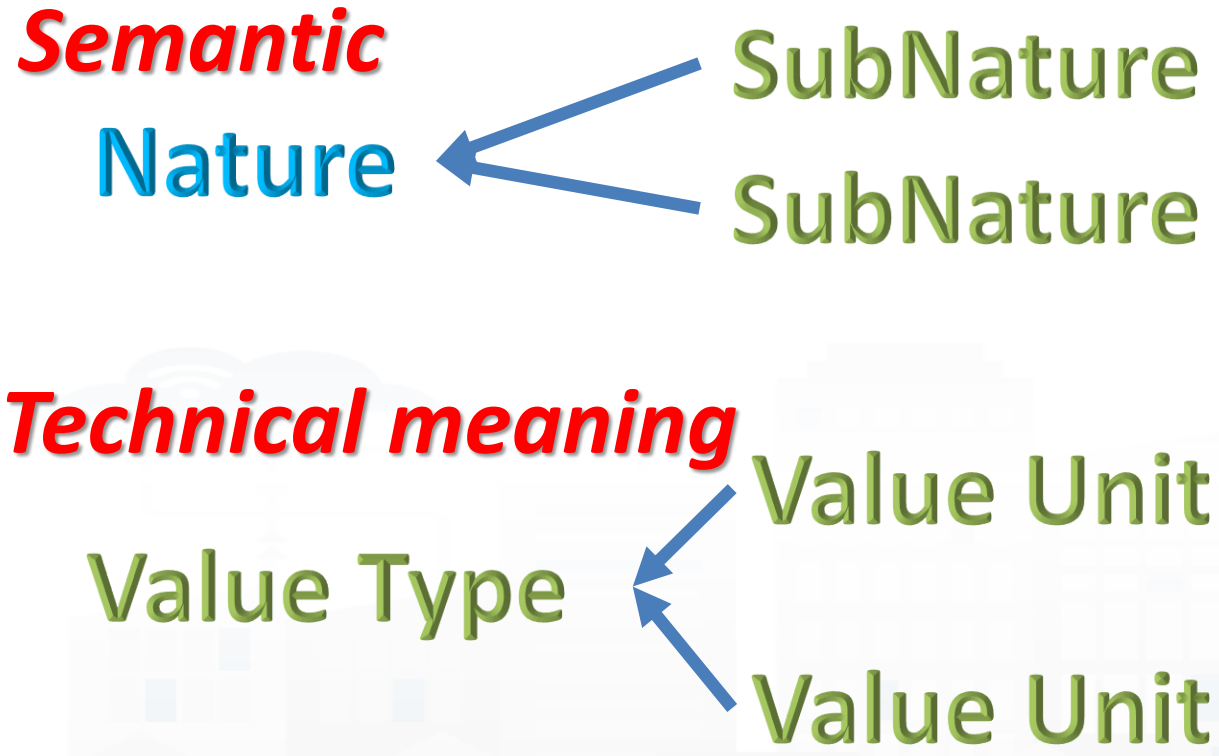
	Datagate	ETL	IoTApp
types of data managed	S	S, P	S, P, RT
Data protocol types managed	PULL	PULL	PULL and PUSH
Scheduling	external	external	internal
Flows to manage N instances of the same dataset	N	N	1
Users' technical level	without	medium/high	low
Development time	1,2 hours	1, 2 weeks	3, 4 days
Semantic (KM4City)	standard template	ad hoc (manual)	ad hoc (semi-automatic)
Developed number	1334 datasets	162	76
Mean number of blocks	0	120.333	27,67
Mean number of lines of code	0	275	229

Dictionary for Data Fields

Semantics and Technical Meaning



Unified Data and Services Model/Classification



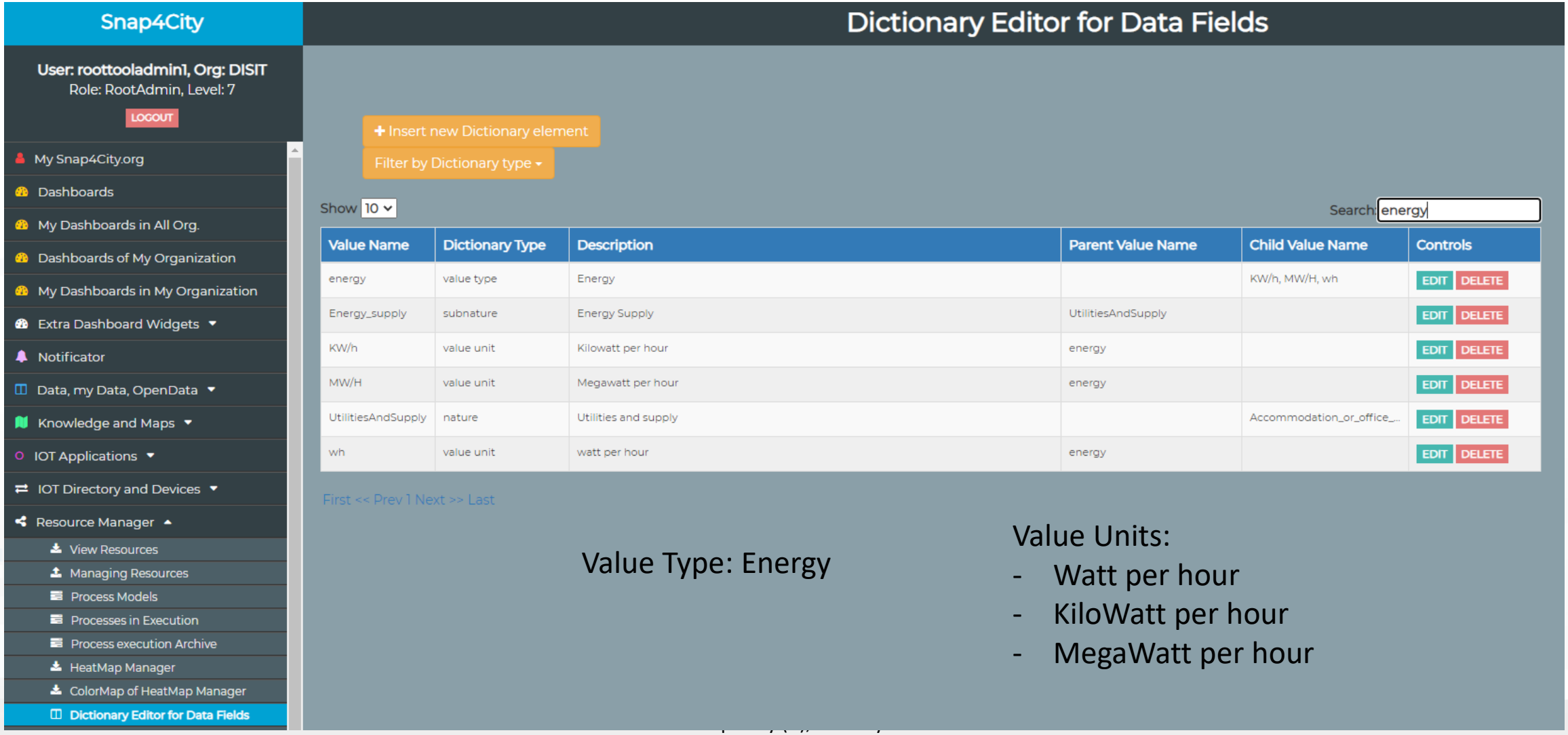
- Exists a Dictionary for the 4 categories
- They are related each other and not all values are possible
- Right setting lead to right rendering on graphs and automated combinations and processing
- The Disctionary is used by many tools

Two Examples

- **HLT: MyKPI**
- **Nature: Industry and manufacturing**
- **Subnature: Chemical**
- **Value Type: Density percentage**
- **Value Unit: %**
- **Data Type**: float mykpi
- **Value name**: CloroParaffine
- **Last Date**: 2019-02-25 *«DateTime»*
- **Last Value**: 87.0
- **Healthiness**:
- **Last Check**: 2020-04-03 10:28:12
- **Ownership**: private
- **Organization**: Firenze

- **HLT: Sensor**
- **Nature: From IOT Device to KB**
- **Subnature: IOT Sensor**
- **Value Type: Battery Level**
- **Value Unit: V**
- **Data Type**: float
- **Value name**: Irrigator fioriera Gag
- **Last Date**: 2020-04-01 12:59:00 *«dateObserved»*
- **Last Value**: 5.18
- **Healthiness**:
- **Last Check**: 2020-04-03 03:28:12
- **Ownership**: public
- **Organization**: Firenze

Example of Energy and its Value Units



Value Type: Energy

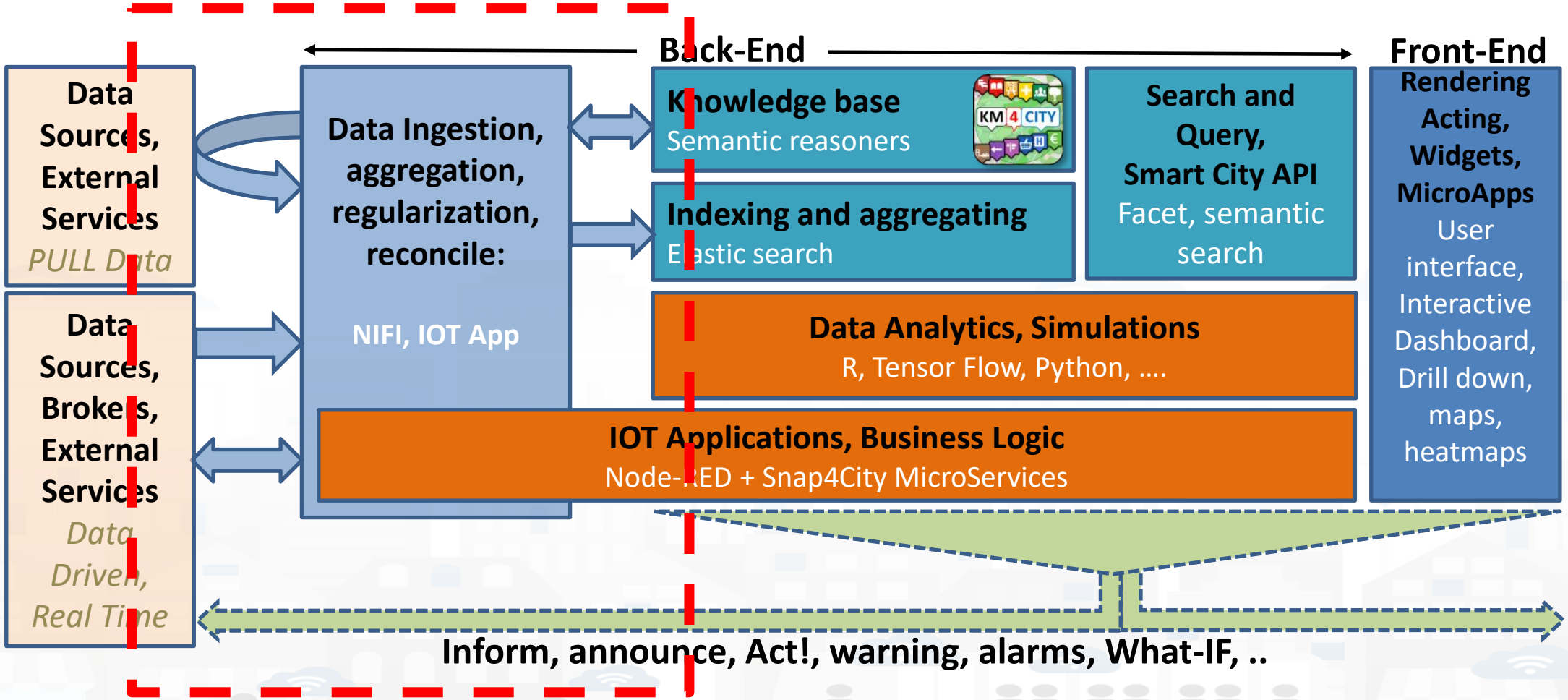
Value Units:
- Watt per hour
- KiloWatt per hour
- MegaWatt per hour

TOP

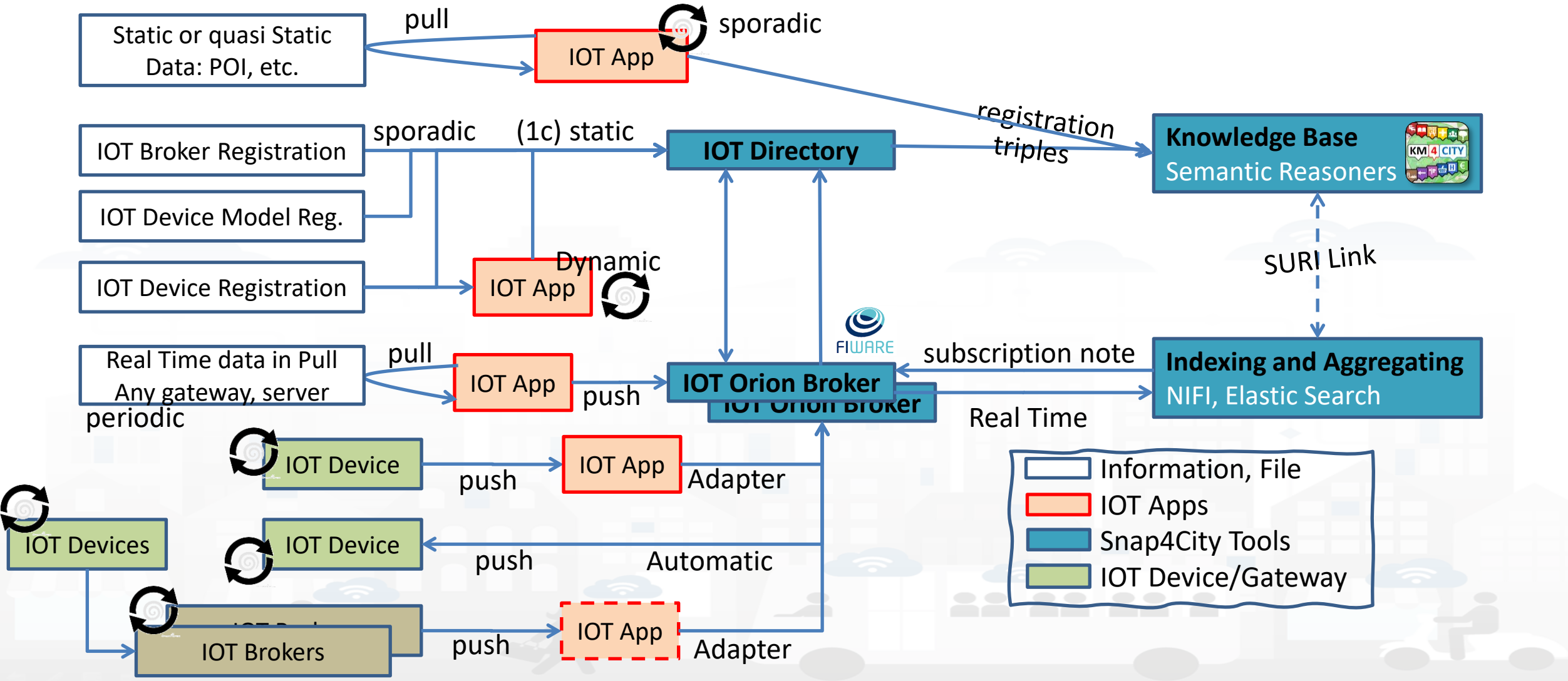
Data Ingestion Strategy and Tools



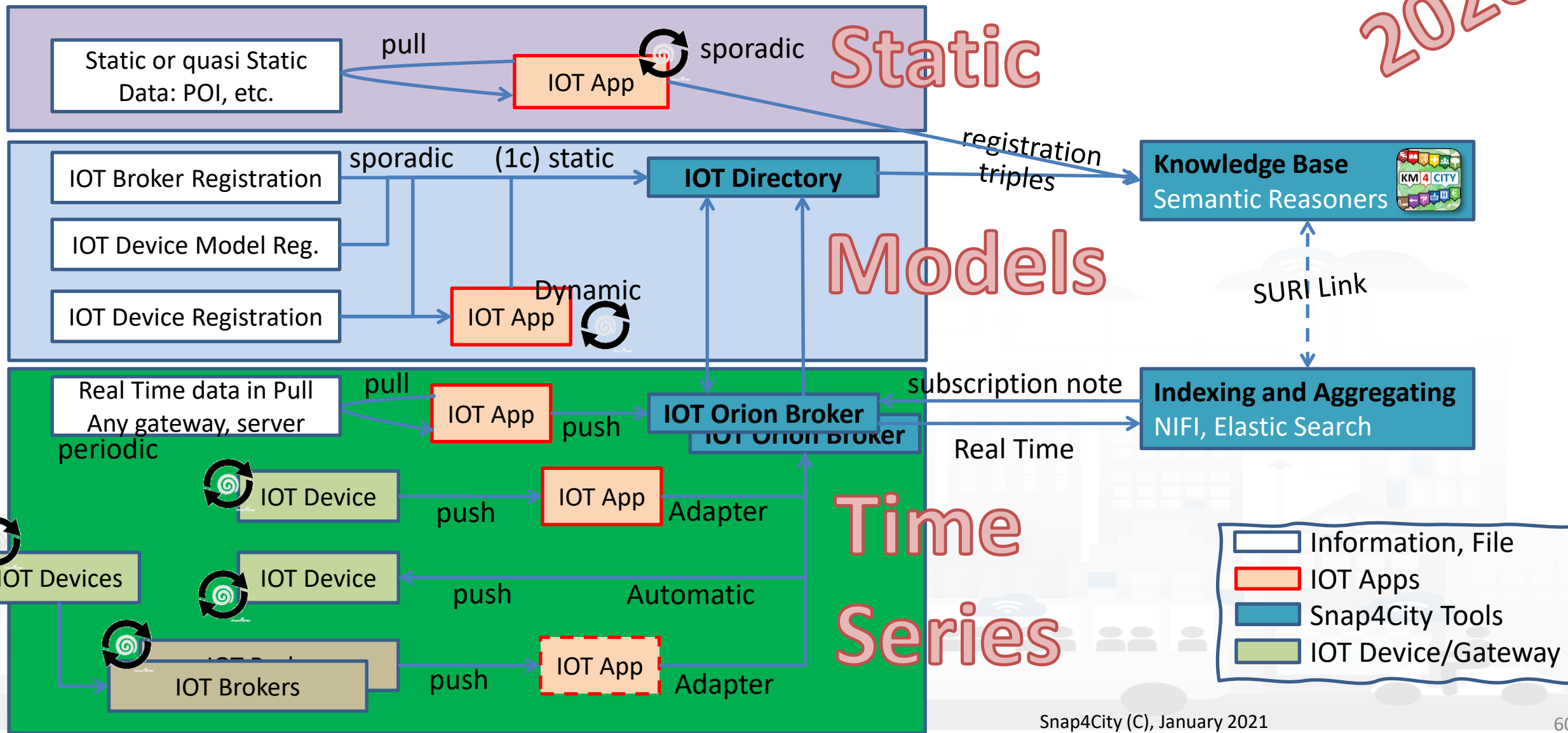
Snap4City Architecture vs Data Ingestion



Snap4city Data Ingestion Flow Diagram



Snap4city Data Ingestion Flow Diagram



On the Flow Diagram

- **Static** (unified model for multidomain indexing)
  - Geodata, Open Data as POI, Data Bases, records, etc.
  - They change over time sporadically
- **Models** (Registration of IOT Device Models, IOT Devices, IOT Brokers), for entities that have multiple instances in time / space
  - Registration of entities with their metadata and data structures
    - Brokers, Devices, structures of real time data, machine models for IOT Industry 4.0, sensor models/structure, etc.
- **Time Series (dynamic data)**
  - Any instance of IOT Devices over time and space with any kind of entity relationship each other and with other city entities

IOT Device for Time Series



What About IOT Devices, time series

IOT Device



Sends a message

Message (
timestamp: 02-04-2020 at 10:30,
Temperature: 29.34,
Humidity: 35
)

dateObserved	Temp	Humidity
02-04-2020 10:30	34.5	23
02-04-2020 10:40	36.5	24
02-04-2020 10:50	36.0	22.5

- A set of data coming from an IOT Device with multiple sensor become a time series of values for devices.
- For example taking a new measure every 10 minutes (Red Lines)
- Each new measure in Snap4City is conventionally time located in «dateObserved»



Verification of Data Ingestion Process



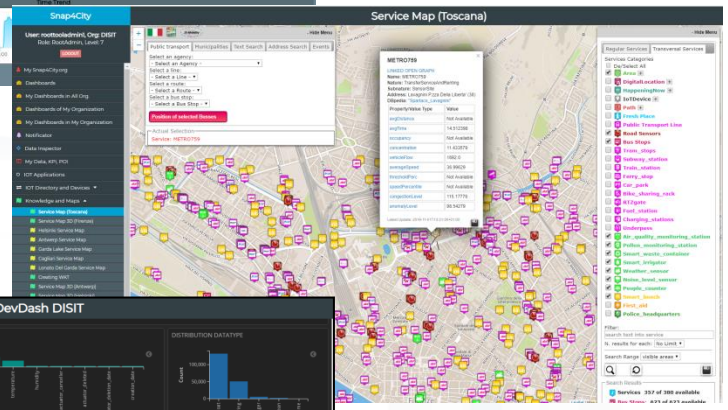
Checking data ingestion results

Knowledge base
Semantic reasoners

- Data Inspector
- ServiceMap, ASCAPI
  - LOG / LOD viewer
  - Super Service Map
- IOT Broker
- ASCAPI: Swagger
- IOT Directory



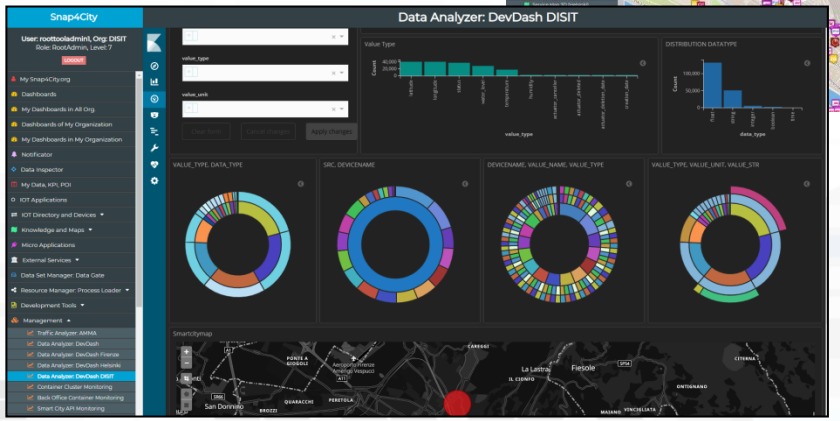
Data Inspector
Digital Twin view



ServiceMap

Indexing and aggregating
Elastic search

- Data Inspector
- ServiceMap, ASCAPI
- My Data Dashboard (Kibana), DevDash
- Elastic Search

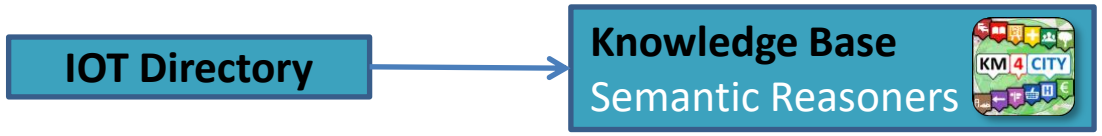


My Data Dashboard
DevDash

Some functionalities are limited to certain roles

Verification of Data Ingestion Process

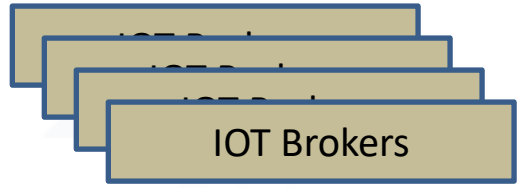
- **Verify that**
  - **IOT Device Creation** see it on
    - IOT Directory, IOT Devices list
    - Service Map if the device is Public
    - Data Inspector if the device is public and/or private
      - You can see the trend to see the time series and last value from the pin on map of the Data Inspector
    - Dashboards…… you can create a specific one using the Dashboard Wizard
  - **IOT Device Data (time series)** see them on
    - IOT Broker, via …… A dedicated services is coming….
    - Service Map if the device is Public
    - Data Inspector, this means that the data are on Storage
      - You can see the trend to see the time series and last value from the pin on map of the Data Inspector
    - Dashboards…… you can create a specific one using the Dashboard Wizard



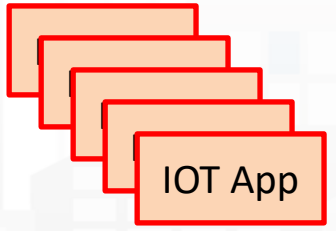
Additional Notes

- The **IOT Orion Brokers** can be feed by means of
  - **IOT App** of Snap4City (to implement Agents and/or Adapters)
  - **IOT Agents** and/or **NGSI Adapters** by FiWare for different protocols
  - **IOT Brokers** of any kind, different protocols and producers, also as Gateways, and they can be located on premise and/or on any cloud





- **IOT App, IOT Agents, Adapters** can
  - be on IOT Edge
  - be implemented as IOT App of Snap4City
  - be on other clouds and services
  - work on a large range of different protocols and kinds
  - have or not Snap4City libraries installed



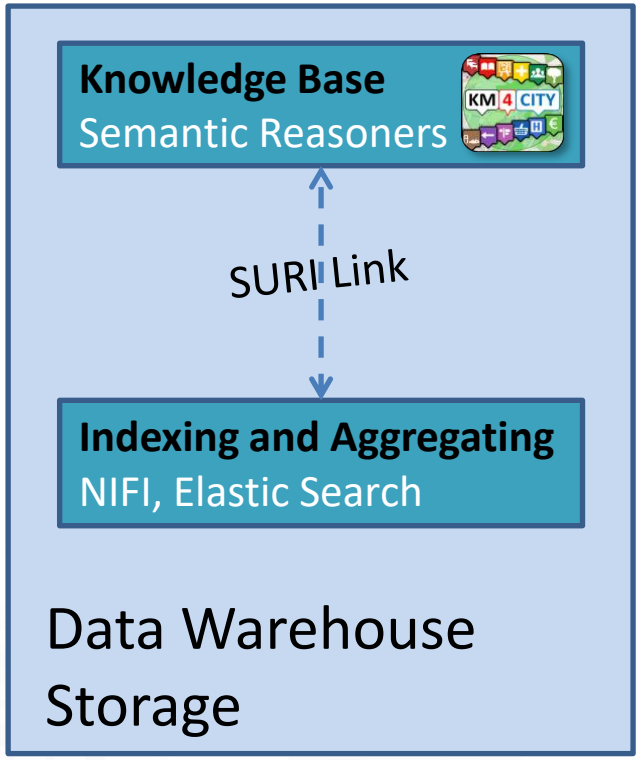
Additional Notes

- The **Internal IOT Orion Brokers** at Snap4City IOT Orion Broker are used as gate for data ingestion and actuations. Since they are
  - connected with the IOT Directory and discovery of the Knowledge Base to make easy the production of Dashboards by wizard, Data Inspector;
  - Synchronized automatically with NIFI/ElasticSearch for the Automated Data Shadow and Indexing
  - Ready to be used by IOT App to subscribe for creating even driven IOT Apps, on IOT Edge and Cloud, etc...
  - Compatible and harmonized with FiWare networks

Direct Ingestion

- **Direct Data Ingestion is also possible:**
  - **From data sources to Data Warehouse Storage of Snap4City, Snap4Industry**

- **Data Warehouse Storage includes:** KB, and I&A, reported on right side can be acted via API REST Call
  - for direct feeding data into store and retrieval,
  - which can be exploited by:
    - IOT Applications
    - applications in Python, R Studio, Java



Notation Terminology

WHERE	Are synonymous at level of service which can be IOT device or entity with data	Are synonymous at level of the single attribute of the entity, device, service, etc.
IOT Directory	IOT Device	Sensor, Actuator, Attributes, Values (value name)
Knowledge Base, ServiceMap, SmartCity API, ASCAPI	Service, ServiceURI, SURI	Attribute, Metric
DataInspector, Wizard, Dashboard	ValueName	Sensor, Sensor Actuator, ValueType
IOT Applications, Node-RED	ServiceURI, SURI	SURI and its real time results of the objects into the data structure

How to access at SURI
See Part 3

ServiceURI, SURI of a sensor device:

- http://www.disit.org/km4city/resource/METRO759

ServiceURI, SURI extended with attribute:

- http://www.disit.org/km4city/resource/METRO759&metric=vehicleFlow
- http%3A%2F%2Fwww.disit.org%2Fkm4city%2Fresource%2FMETRO759&metric=vehicleFlow
- In some cases
  - http://www.disit.org/km4city/resource/METRO759/vehicleFlow

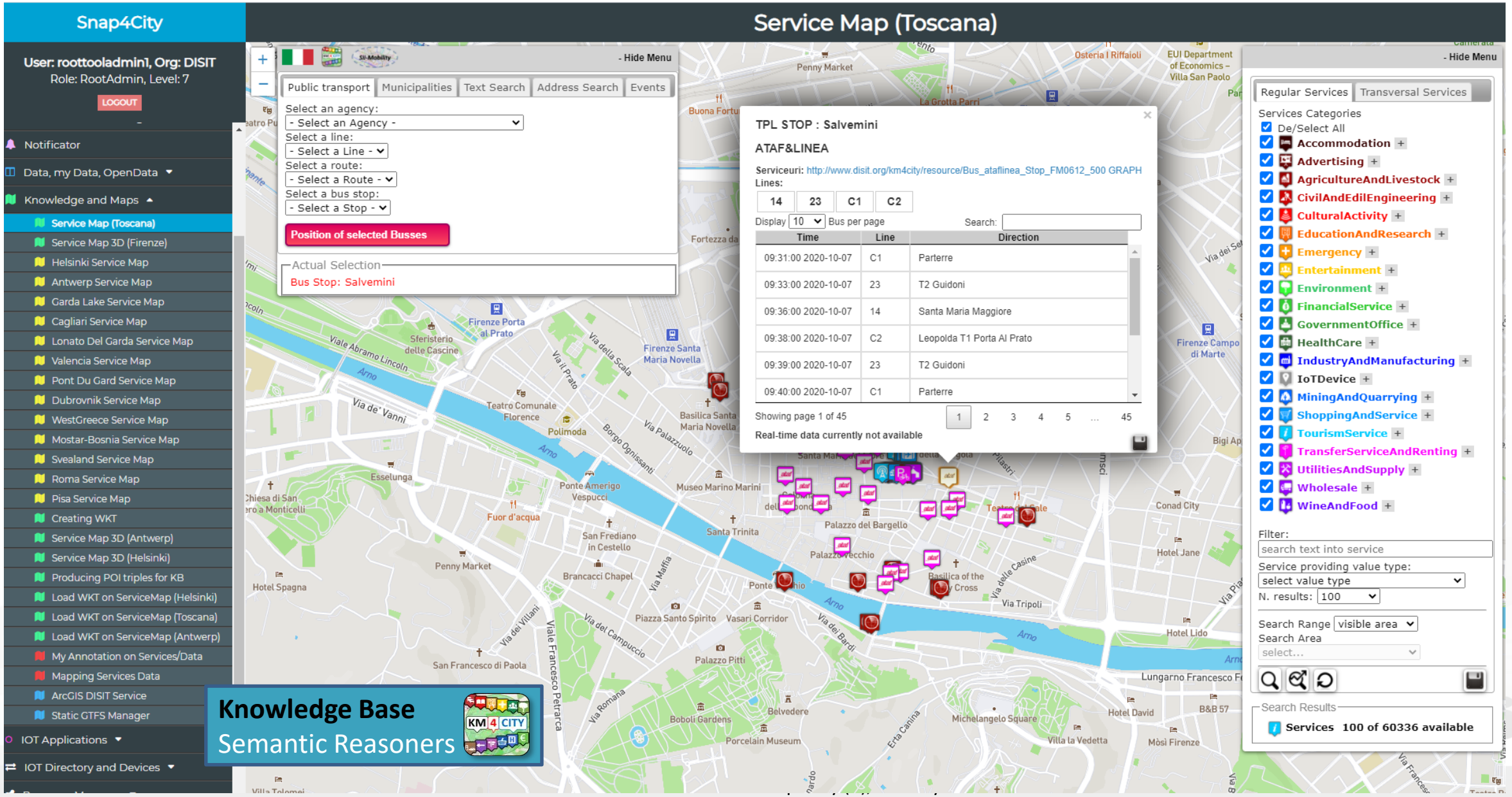
Usage of the ServiceMap and Knowledge Base Browsing



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

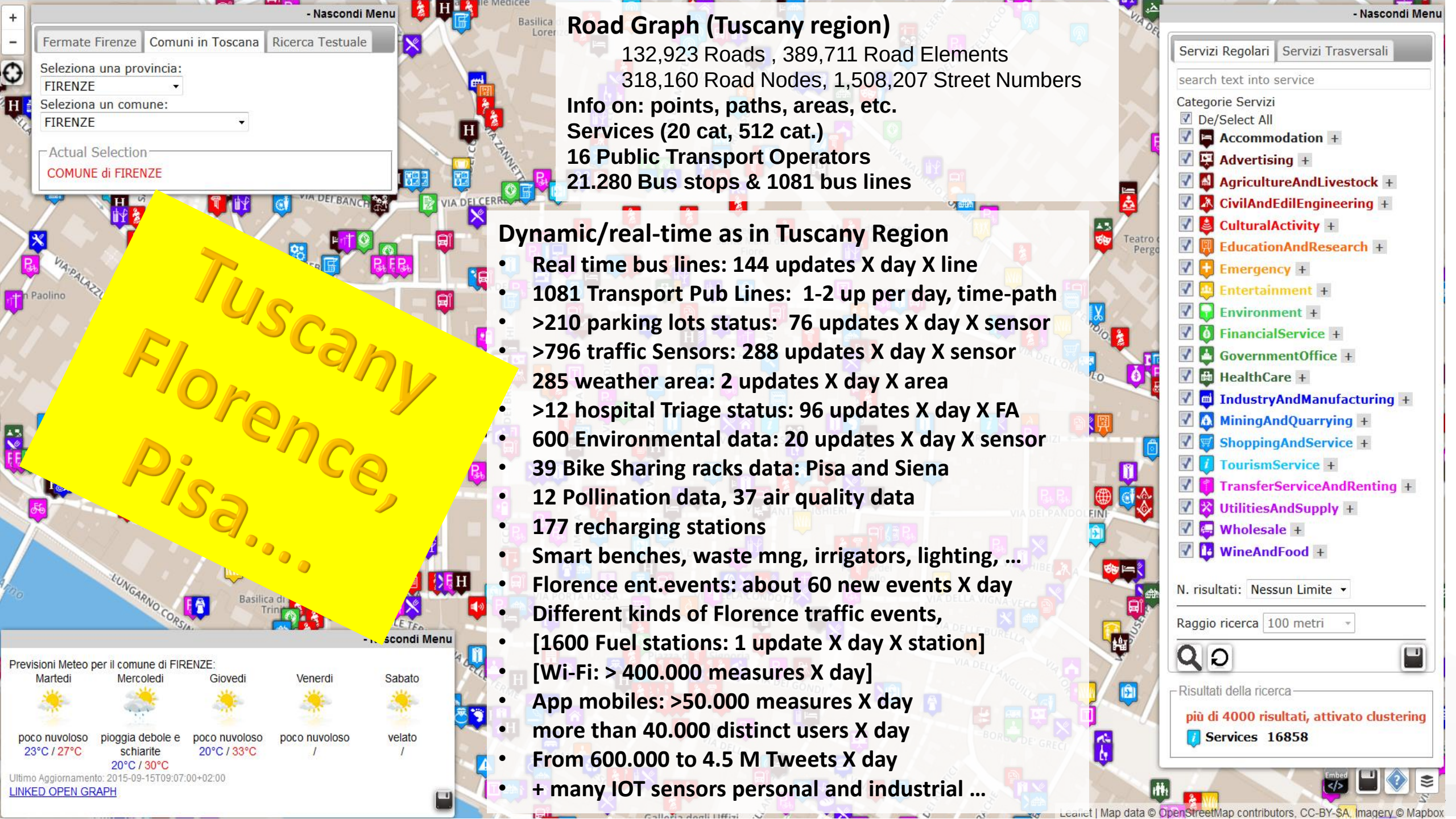
ServiceMaps

SNAP4CITY | KM4CITY



Semantic modeling

- **Km4City is the reference ontology for Snap4City,** It allows to:
  - keep connected city entities each other:
    - Semantic Index, reticular
    - Perform spatial, geo graphic, and temporal reasoning
  - Discover city entities and their relationships via IOT App and Smart City API:
    - IOT devices, IOT sensors, city elements, roads, services, Brokers, etc. etc.
  - Provide access via Advanced Smart City API
  - Federate other Km4City Knowledge Bases, the approach allows to scale geographically and create redundancies, improving performances
- Documentation
  - **TC5.15 - Snap4City Smart City API Collection and overview, real time**
  - **ServiceMap and ServiceMap3D, Knowledge Model, Km4City Ontology**
  - **Knowledge Base Graphs and Queries: browsing and queries into the KB**


- Nascondi Menu
Fermate Firenze
Comuni in Toscana
Ricerca Testuale
Seleziona una provincia:
FIRENZE
Seleziona un comune:
FIRENZE
Actual Selection
COMUNE di FIRENZE
Road Graph (Tuscany region)
132,923 Roads , 389,711 Road Elements
318,160 Road Nodes, 1,508,207 Street Numbers
Info on: points, paths, areas, etc.
Services (20 cat, 512 cat.)
16 Public Transport Operators
21.280 Bus stops & 1081 bus lines
Tuscany Florence, Pisa....
Dynamic/real-time as in Tuscany Region
• Real time bus lines: 144 updates X day X line
• 1081 Transport Pub Lines: 1-2 up per day, time-path
• >210 parking lots status: 76 updates X day X sensor
• >796 traffic Sensors: 288 updates X day X sensor
285 weather area: 2 updates X day X area
• >12 hospital Triage status: 96 updates X day X FA
• 600 Environmental data: 20 updates X day X sensor
• 39 Bike Sharing racks data: Pisa and Siena
• 12 Pollination data, 37 air quality data
• 177 recharging stations
• Smart benches, waste mng, irrigators, lighting, …
• Florence ent.events: about 60 new events X day
• Different kinds of Florence traffic events,
• [1600 Fuel stations: 1 update X day X station]
• [Wi-Fi: > 400.000 measures X day]
• App mobiles: >50.000 measures X day
• more than 40.000 distinct users X day
• From 600.000 to 4.5 M Tweets X day
• + many IOT sensors personal and industrial …
- Nascondi Menu
Previsioni Meteo per il comune di FIRENZE:
Martedi
poco nuvoloso
23°C / 27°C
Mercoledi
pioggia debole e schiarite
20°C / 30°C
Giovedi
poco nuvoloso
20°C / 33°C
Venerdi
poco nuvoloso
/
Sabato
velato
/
Ultimo Aggiornamento: 2015-09-15T09:07:00+02:00
LINKED OPEN GRAPH
- Nascondi Menu
Servizi Regolari
Servizi Trasversali
search text into service
Categorie Servizi
De/Select All
Accommodation
Advertising
AgricultureAndLivestock
CivilAndEdilEngineering
CulturalActivity
EducationAndResearch
Emergency
Entertainment
Environment
FinancialService
GovernmentOffice
HealthCare
IndustryAndManufacturing
MiningAndQuarrying
ShoppingAndService
TourismService
TransferServiceAndRenting
UtilitiesAndSupply
Wholesale
WineAndFood
N. risultati: Nessun Limite
Raggio ricerca 100 metri
Risultati della ricerca
più di 4000 risultati, attivato clustering
Services 16858
Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

ServiceMaps

Knowledge Base
Semantic Reasoners

- ServiceMap is the **main Tool** to:
  - monitor the status of the Knowledge Base
  - test queries and produce query and SmartCity API testing calls for developers
    - Any kind of search (semantic, full text, etc.), routing,
  - Access at the specific Graph Data base via LOG.disit.org
- **ServiceMap is showing**:
  - only **public data**. Private data are not shown via ServiceMap but can be accessed via DataInspector
  - data regarding **a single Knowledge Base** of the federated network of KBs. Each KB may contain multiple Organizations.
  - technical views for developers

Discovery

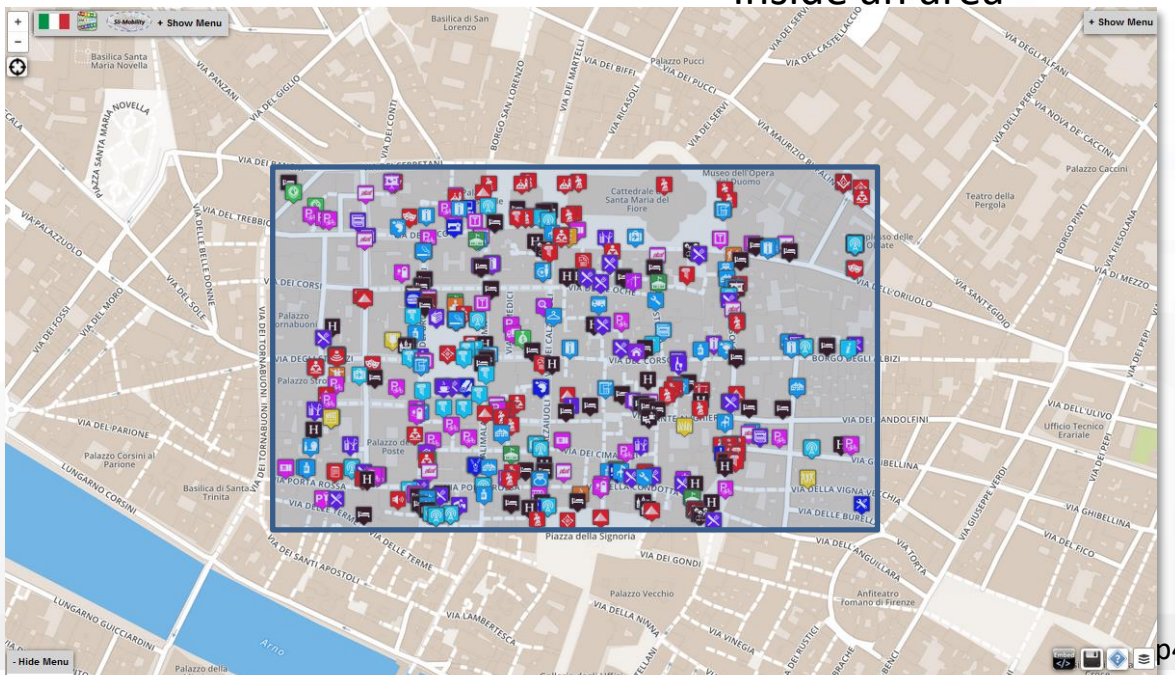
Search by Shape and Distance

Each request or search in the Km4City model can be referred to a point and a ray, to an area, to a polyline

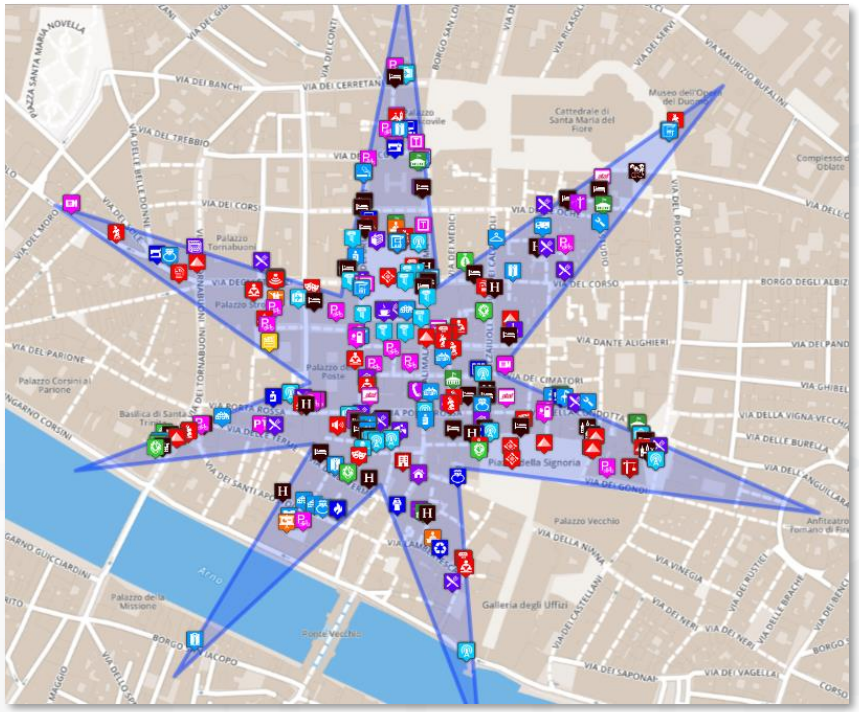
Around a point or POI



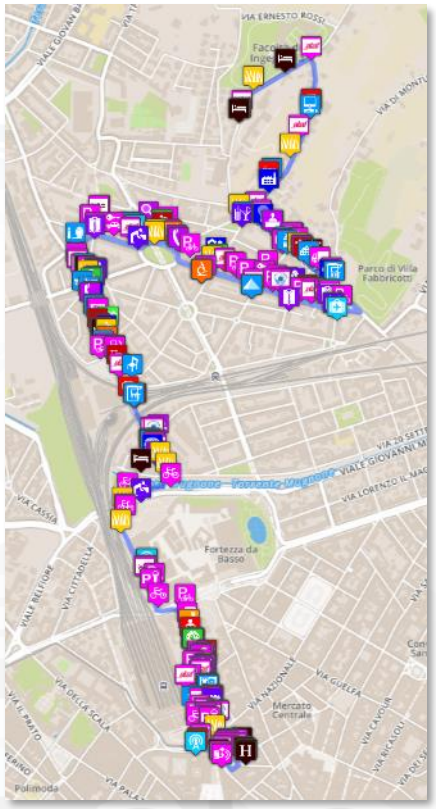
Inside an area



Inside a closed polyline

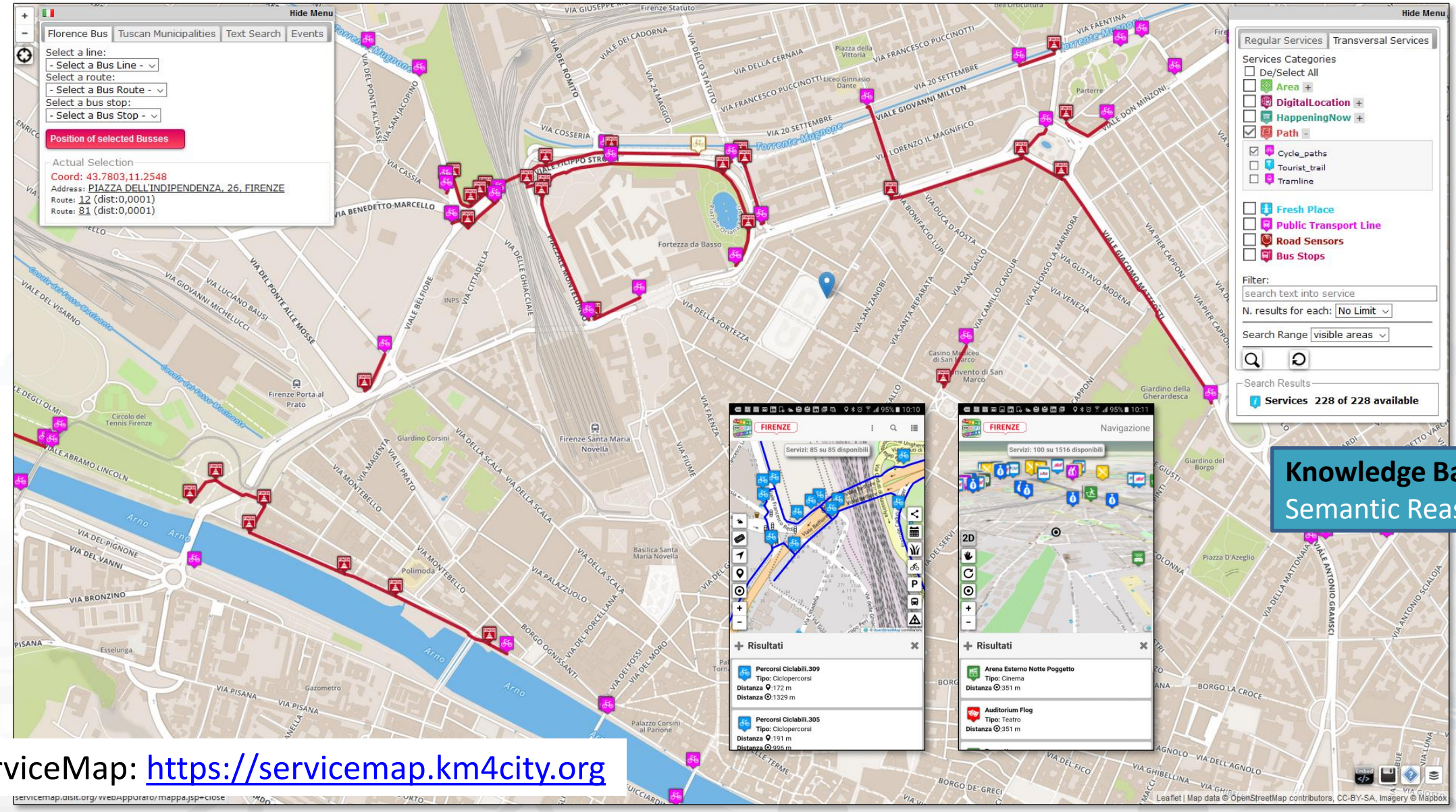


Along a polyline



Knowledge Base
Semantic Reasoners

Cycling Paths

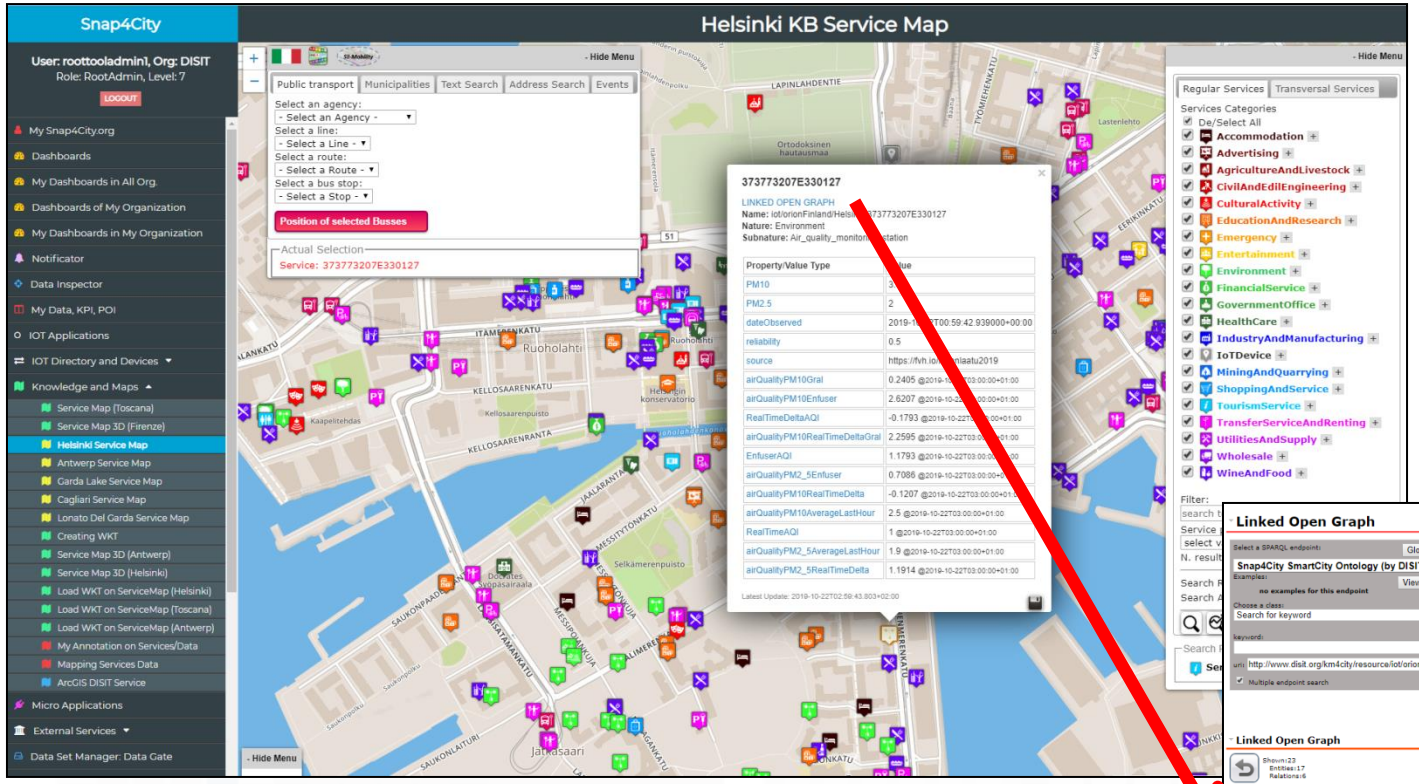


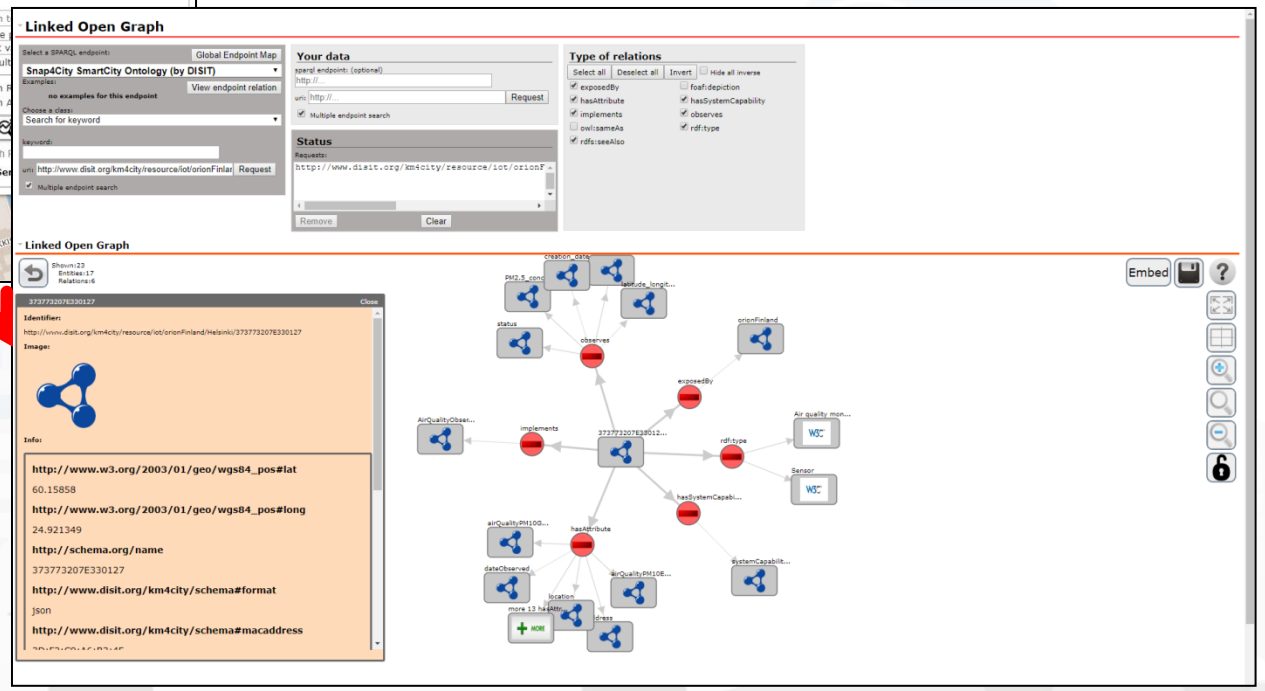
Knowledge Base
Semantic Reasoners

ServiceMap: https://servicemap.km4city.org

Views of the Knowledge Base

Knowledge Base
Semantic Reasoners



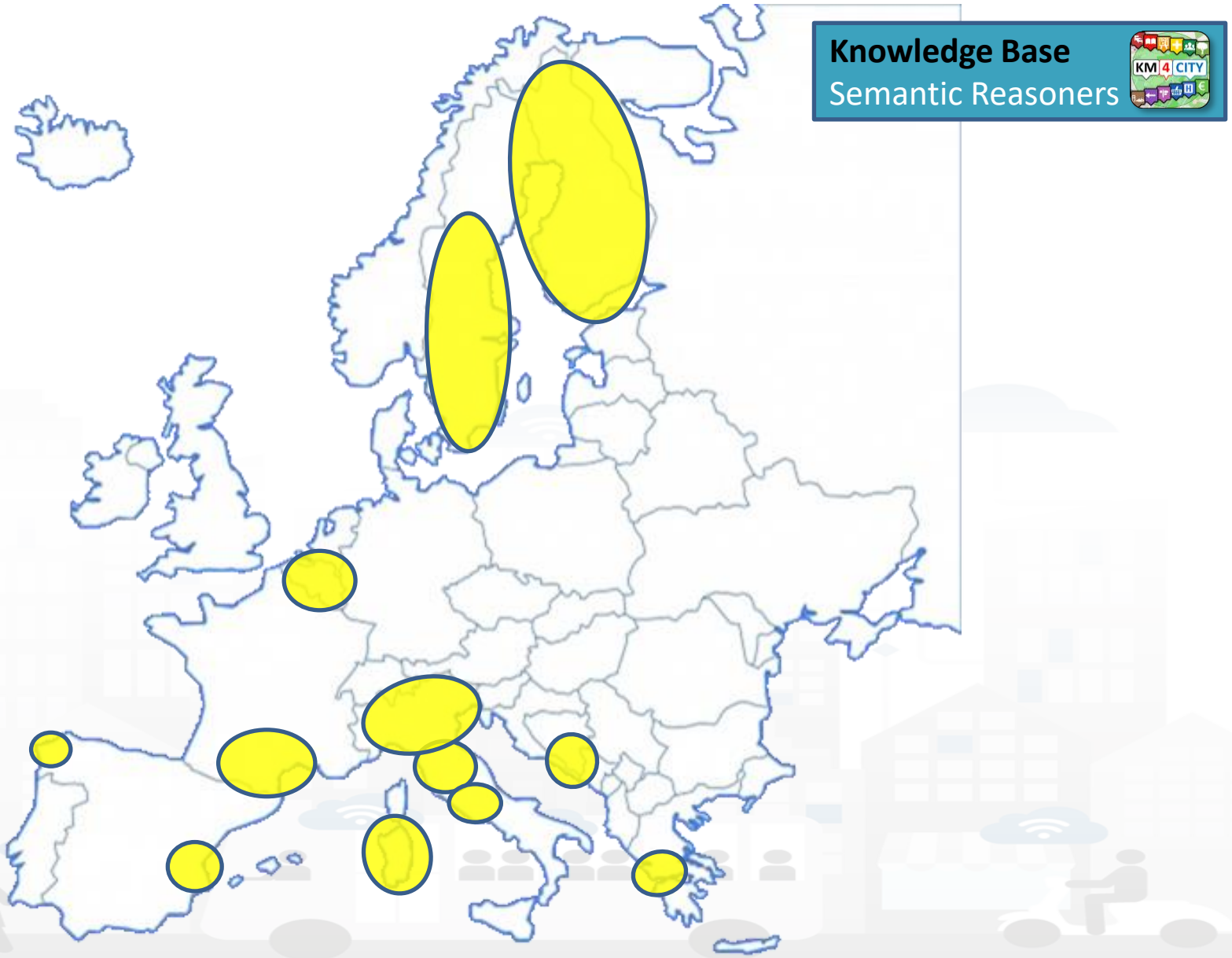


- How pass from ServiceMap to Linked Open Graph, Linket Data view tool

Coverage 2020

Main Organizations/areas

- Antwerp area (Be)
- Capelon (Sweden)
- DISIT demo (multiple)
- Dubrovnik, Croatia
- Firenze area (I)
- Garda Lake area (I)
- Helsinki area (Fin)
- Livorno area (I)
- Lonato del Garda (I)
- Modena (I)
- Mostar, Bosnia-Herzegovina
- Pisa area (I)
- Pont du Gard, Occitanie (Fr)
- Roma (I)
- Santiago de Compostela (S)
- Sardegna Region (I)
- SmartBed (multiple)
- Toscana Region (I), SM
- Valencia (S)
- Venezia area (I)
- WestGreece area (Gr)



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

Km4City Federation

At different levels:

- Among cities/regions
- Among data providers
- Among Operators

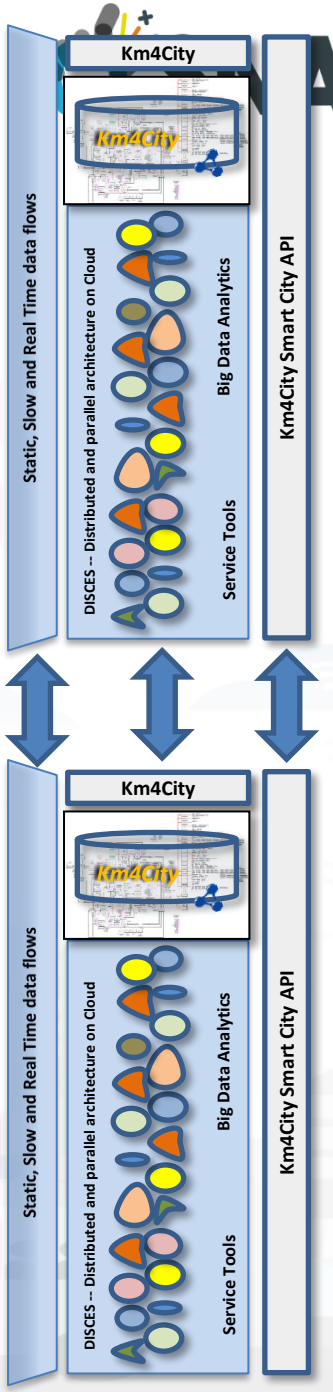
By Means of:

- Smart City API → Apps
- Km4City Smart City Ontology
- Dashboards/data analytics



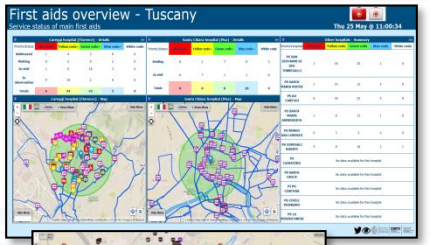


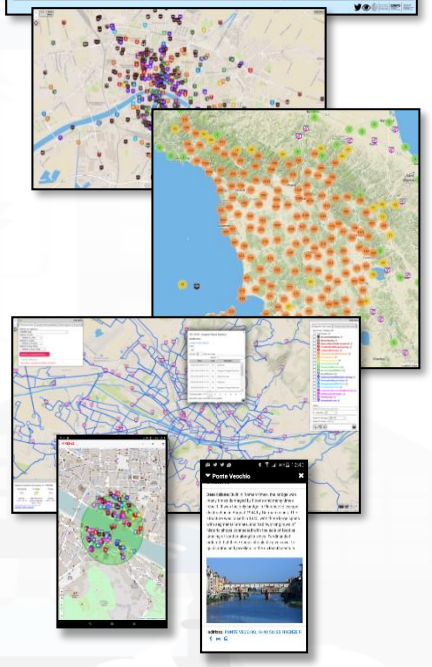




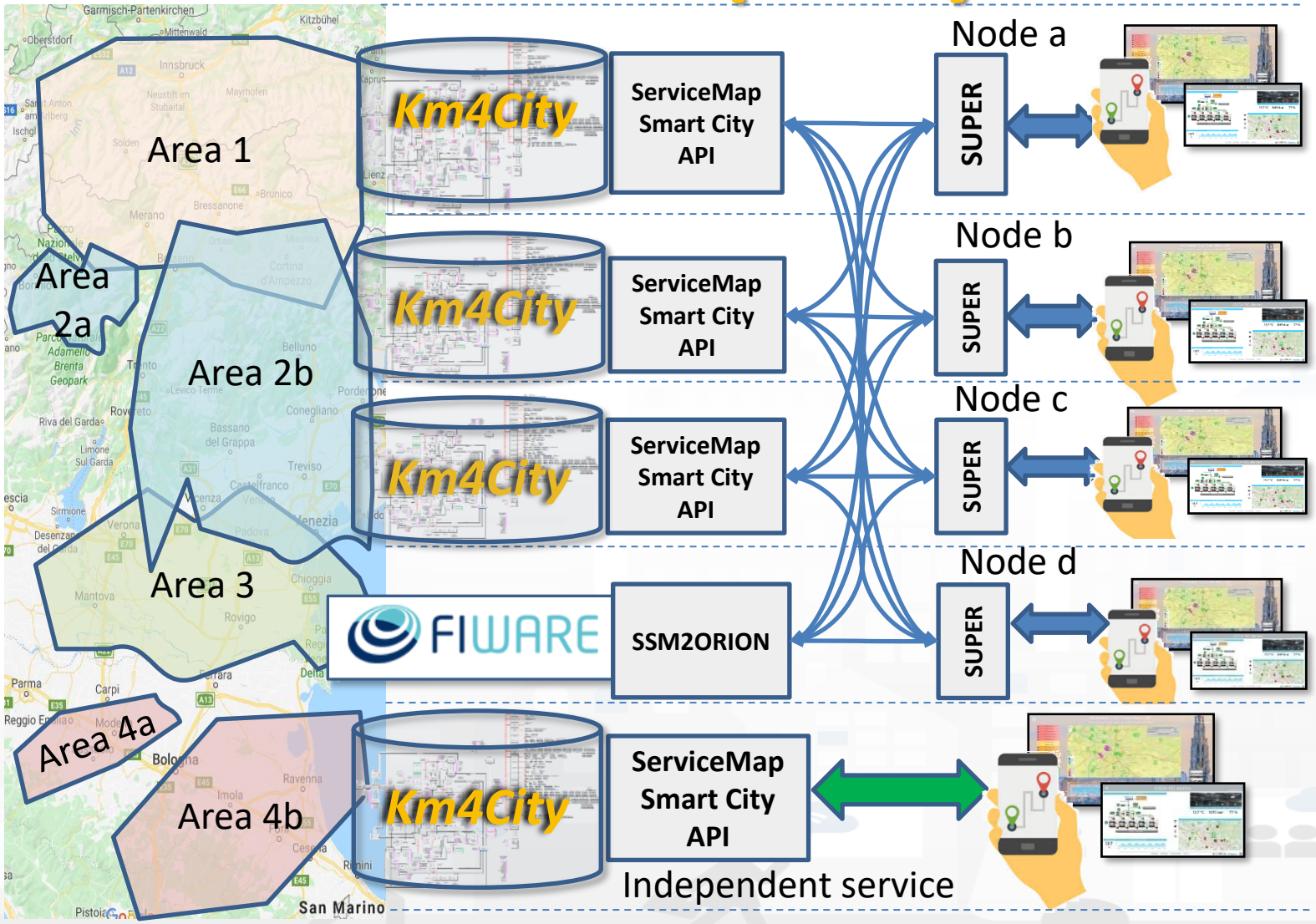








Federation of Snap4City Services



- A Mobile App may refer to one Smart City API Server (for Area 1) via SUPER and receive data from the Federated SUPERS (Area 2) if navigation, queries, etc. are leading to discover out of the addressed KB.
  - SUPER can be used for creating redundant and/or balanced distributed solutions for Federated KB. See Area 2, the two KB in the front.
  - Federated SUPER can have overlapped KB even totally.
  - A Mobile App can be developed to support multiple Smart City API servers, for balancing and
- The usage of Super is not mandatory so that separate services can be produced as well
- Super and Nodes presents the same Smart City APIs.

Super

- Super, Nodes and SSM2ORION presents the same Smart City APIs.
- The **network of Super** can be reconfigured dynamically
  - Multiple networks of Super can be realized as well
  - Distributed Searches via the Federation of Super are performed with o(1) complexity
  - Results from an API rest calls are provided in real time also when the size of the network is large
  - Dashboard widgets and Mobile Apps are enabled to use the Super
  - Clients can pass from one Super to another transparently: moving devices
- Nodes
  - do not need to permanently share data
  - data can be of any size, the data shared is typically public since users of different KB are different and not refer to the same LDAP/KeyCloak authentication/authorization service.
  - may have different number of services
  - Services can be based on KB as well as on Brokers
  - Services managed as HLT of: Sensors, Sensor-Actuators, POI.
    - Data of other HLTs are managed independently from the other SmartCity API such as: MyKPI, External Services, WFS GIS, Heatmaps, special tools, etc. etc.
- The solution support disjoined nodes, federation and independent services

Federated ServiceMap and Smart City API

To improve scalability, fault tolerance and federation among cities:

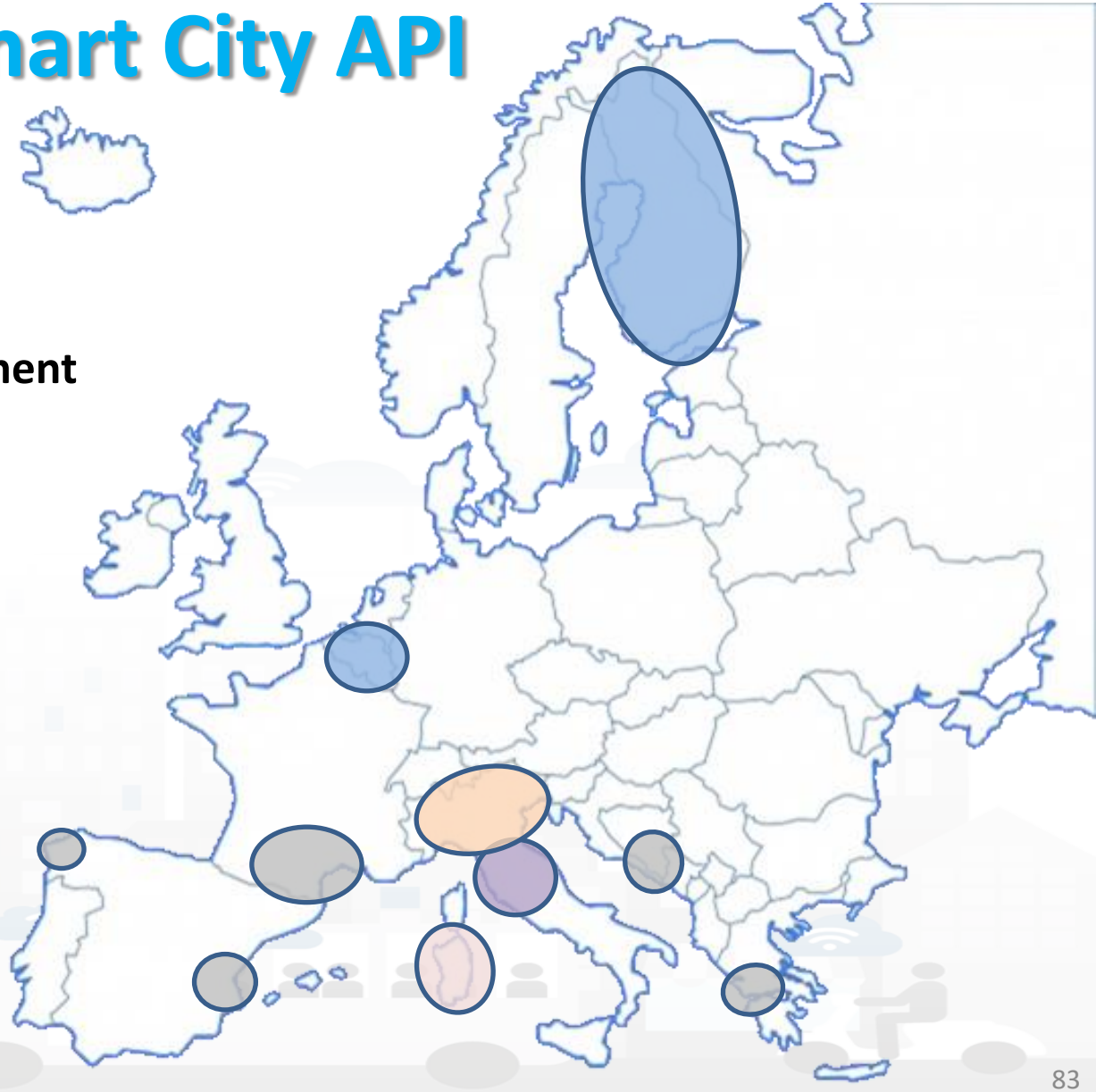
- **One entry point Smart City API for all zones**
- **Multiple Knowledge base See performance assessment**

At different levels:

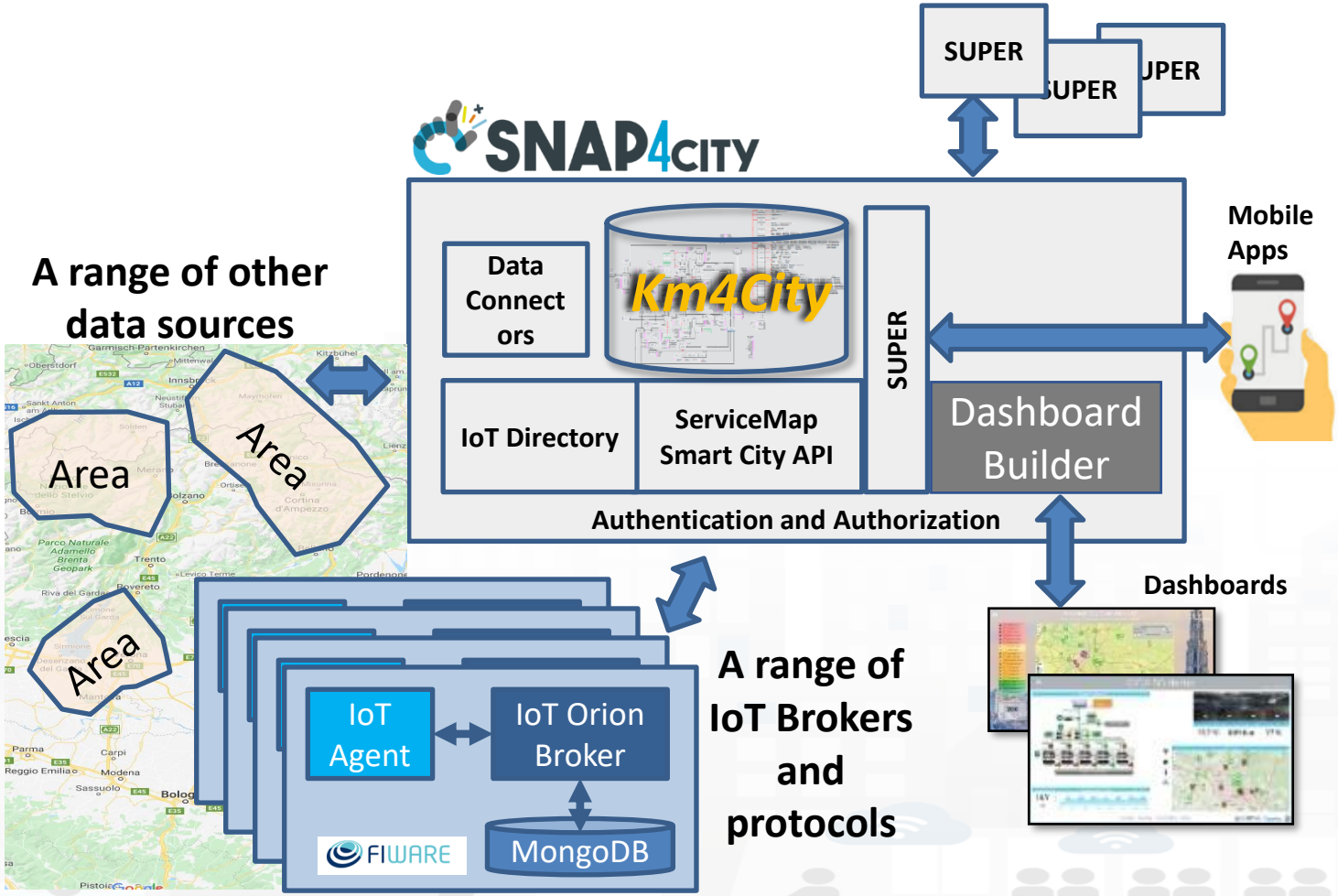
- Among cities/regions
- Among data providers, Operators

By Means of:

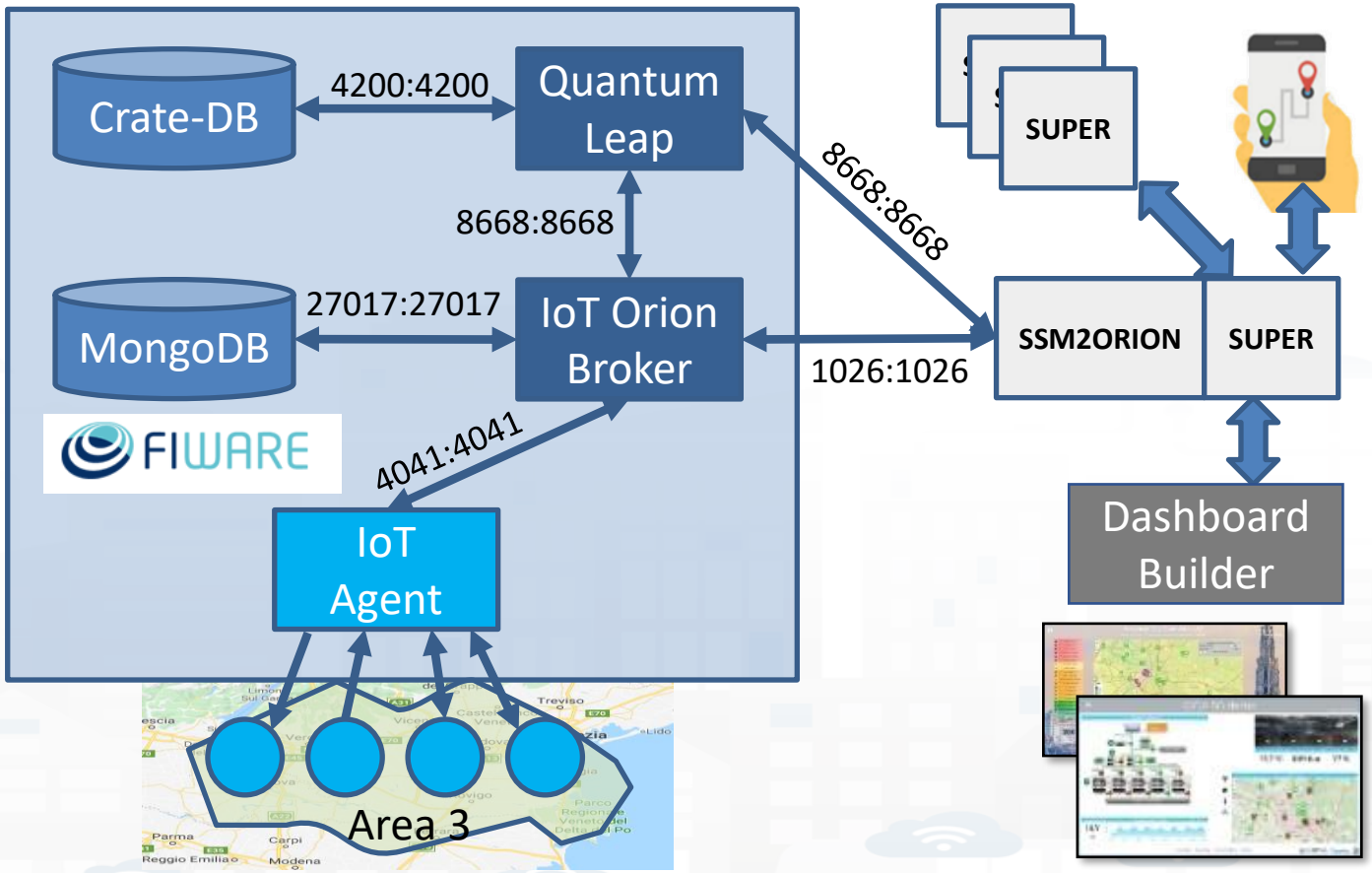
- Smart City API → Apps
- Smart City Ontology
- Dashboards/data analytics
- Organization independent



Snap4City IoT Registration and Access



Federation of Snap4City vs IOT ORION Broker



Data Inspector vs Data Processes Details



Some functionalities are limited to certain roles

Context and Problems

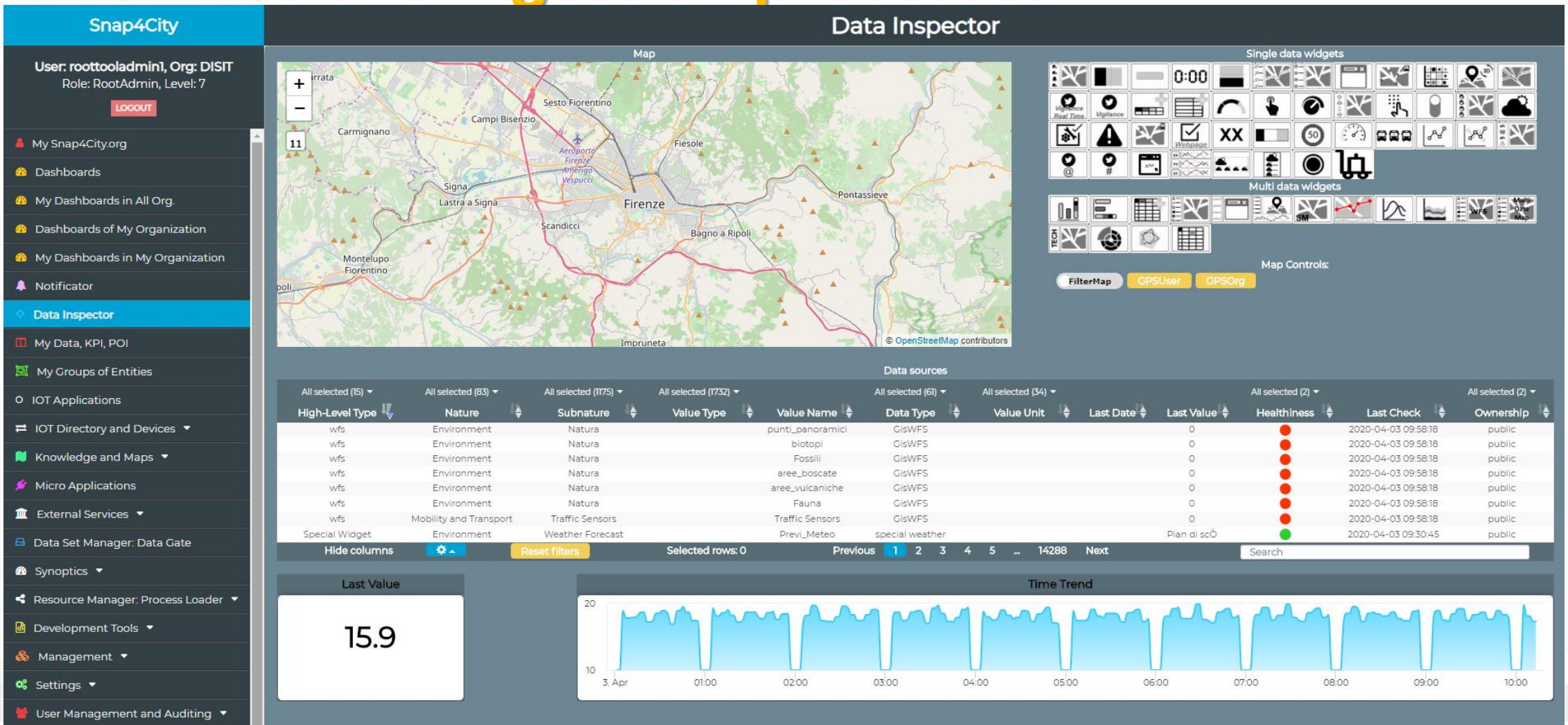
- Smart city context includes solutions that presents
  - **Data**: heterogeneous, large volume, several protocols, legacy data systems, several semantics, real time, multiple domains, etc..
  - **Processes**: several of different kinds, aperiodic, periodic, event driven, ….
  - **Users:** owners, responsibles, developers, final users, etc.
  - **Relationships**: among data, processes, users and mixt
  - **Non Functional Requirements**: security, GDPR, reliability, quality, scalability, etc.
    - Interoperability: legacy, protocols, modularity,

- **Data Ingestion**: the models and mechanism for data gathering
- **Data Inspection**: the model and solution to identify problems, and understand solution

Data Inspector (Digital Twin info) Major Submodels

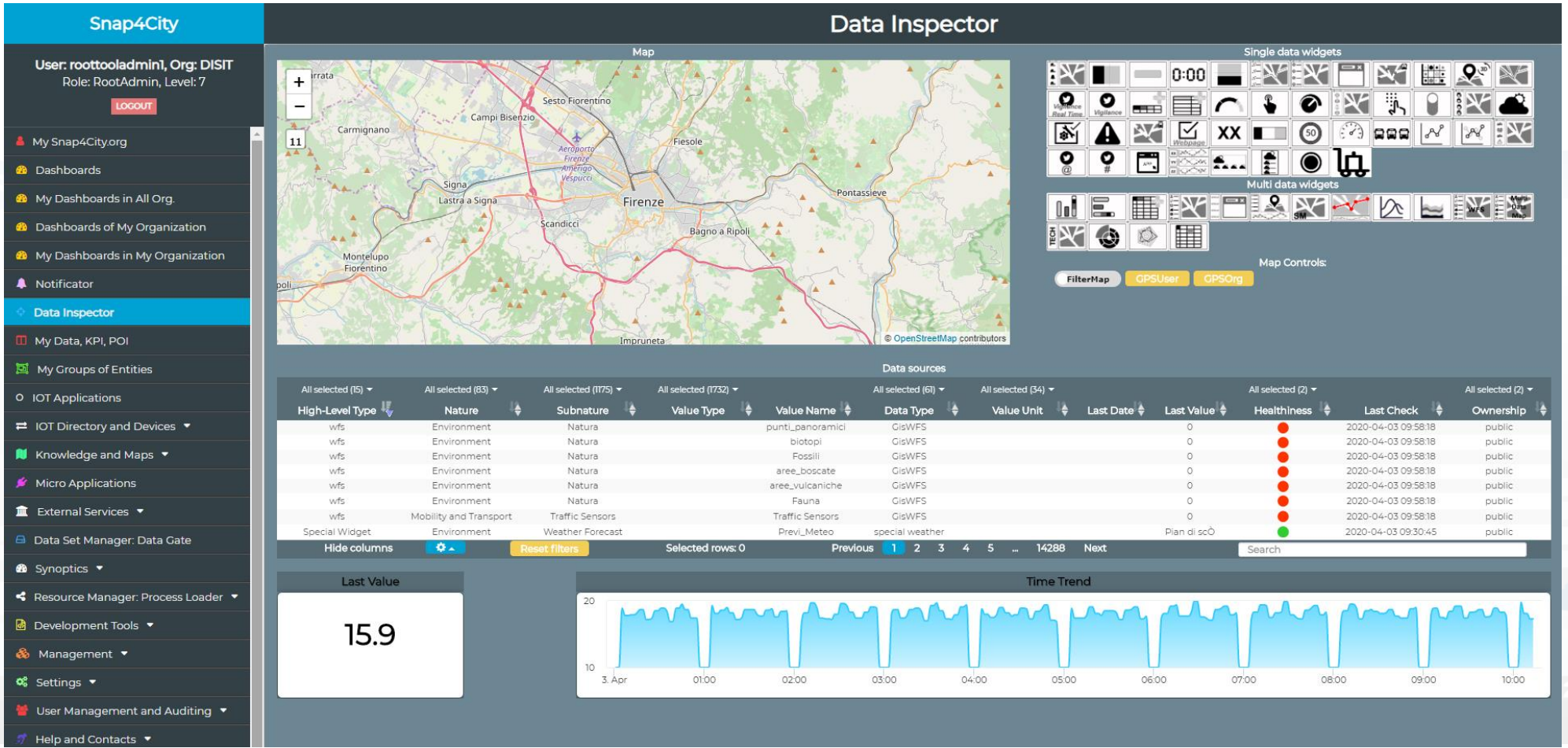
- **Digital Twin**
  - Device and sensors data
  - Values
  - Healthiness criteria and values
    - Machine learning tools
  - Images and physical world
  - Licensing
  - Users
- **Users**
  - Defined the Data and Devices
  - Defined the processes
  - Create dashboards
  - Etc.
- **Process Views**
  - Device Management tool
  - Data ingestion processes
    - ETL, IOT Apps
  - Data storage access views
    - Index views
    - Relationships view
  - Data Analytics and Transformation
    - IOT App, R Studio, Python
  - Data Rendering Dashboards
    - Synoptics
  - Processes' Developers

Data Inspector: all you need to know about data, data sources and ingestion processes



Data Inspector

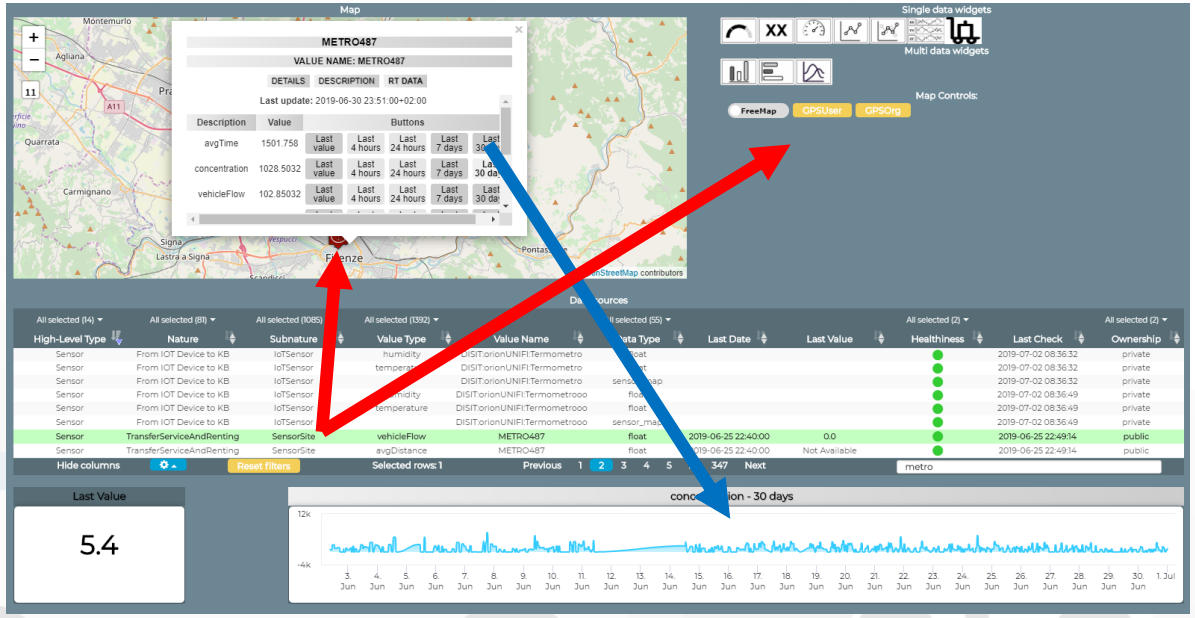
- It is showing data:
  - Listed by High Level Types, and classified as in the **Dashboard WIZARD: see Tutorial Part 2 for details**.
  - of your Organization only, that are public of the organization, your private or those that have been delegated to you some how.

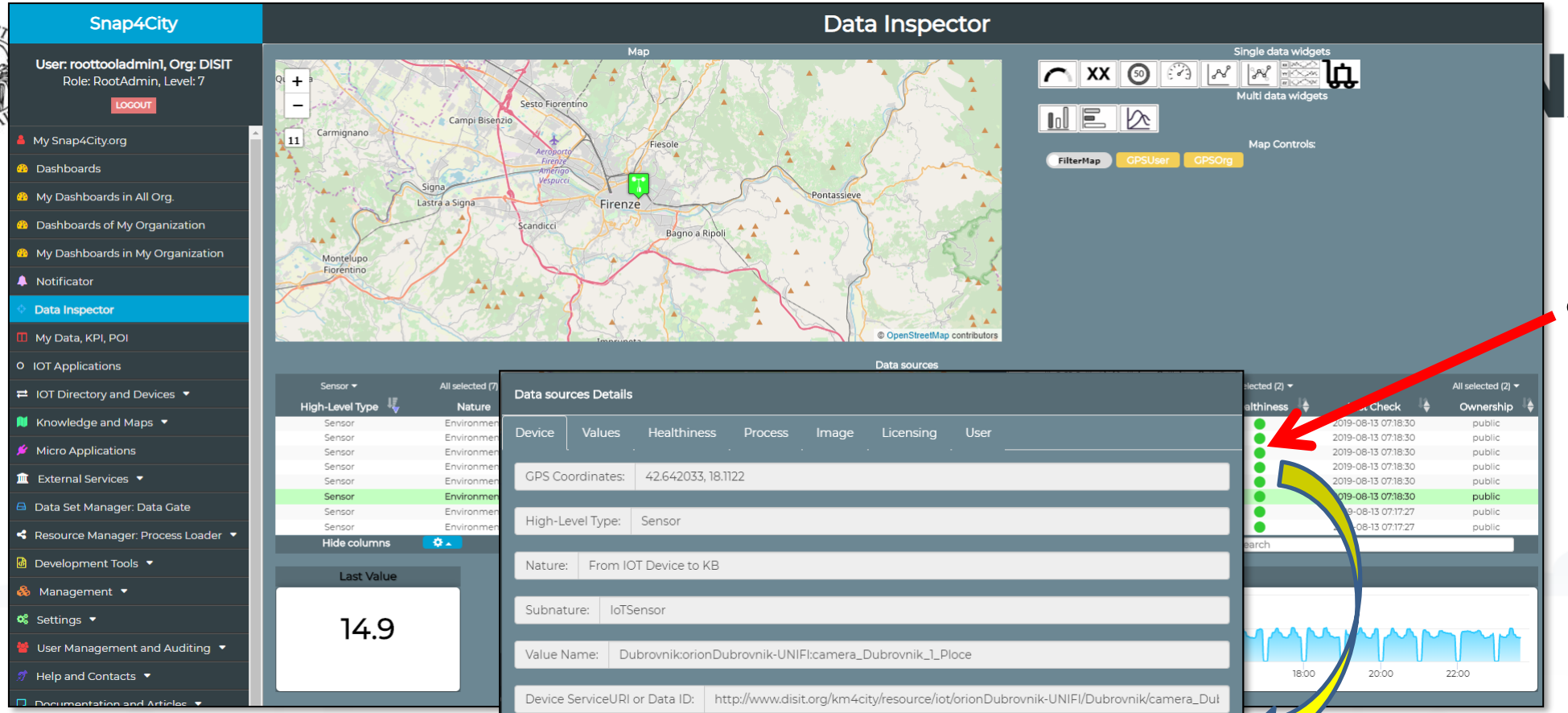


Data Inspector (open on your left side menu)

- **Cross Filtering on the basis of**:
  - **MAP**: pan and zoom → lock , center on GPS coordinates of the user
  - **Data Source Classification**:
    - faceted filtering
    - full text search
  - **Click on data source** to see it on map, and see the graphics representation, just to learn how widgeting it.
  - **Full Text Search** if you remember some desc…

- **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)

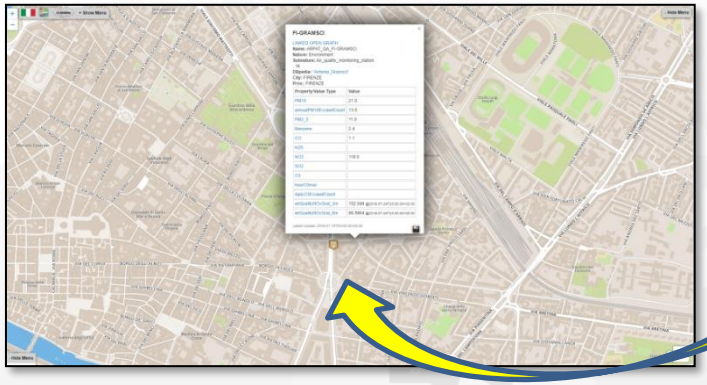


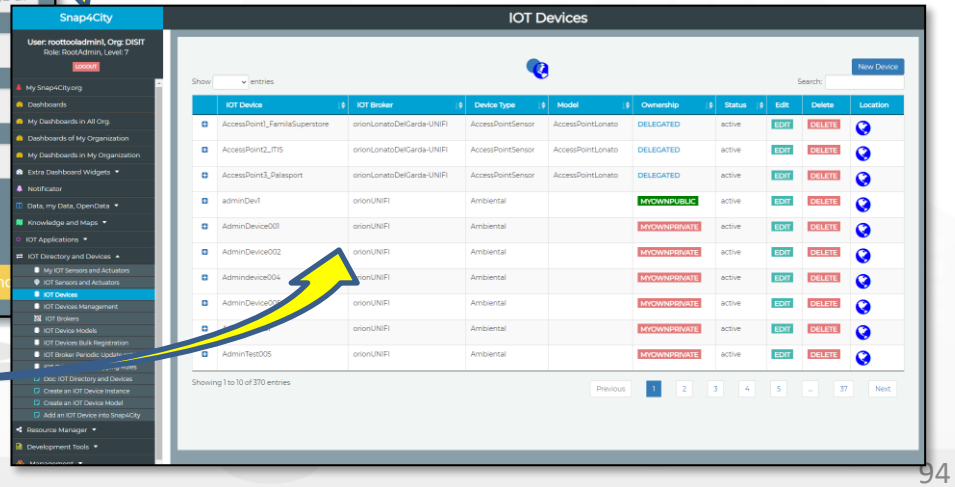


Knowledge Base view

- Click with the mouse on it

HLT: Sensor





Some functionalities are limited to certain roles

HLT: Sensor

Data sources Details

Device | Values | Healthiness | Process | Image | Licensing | User

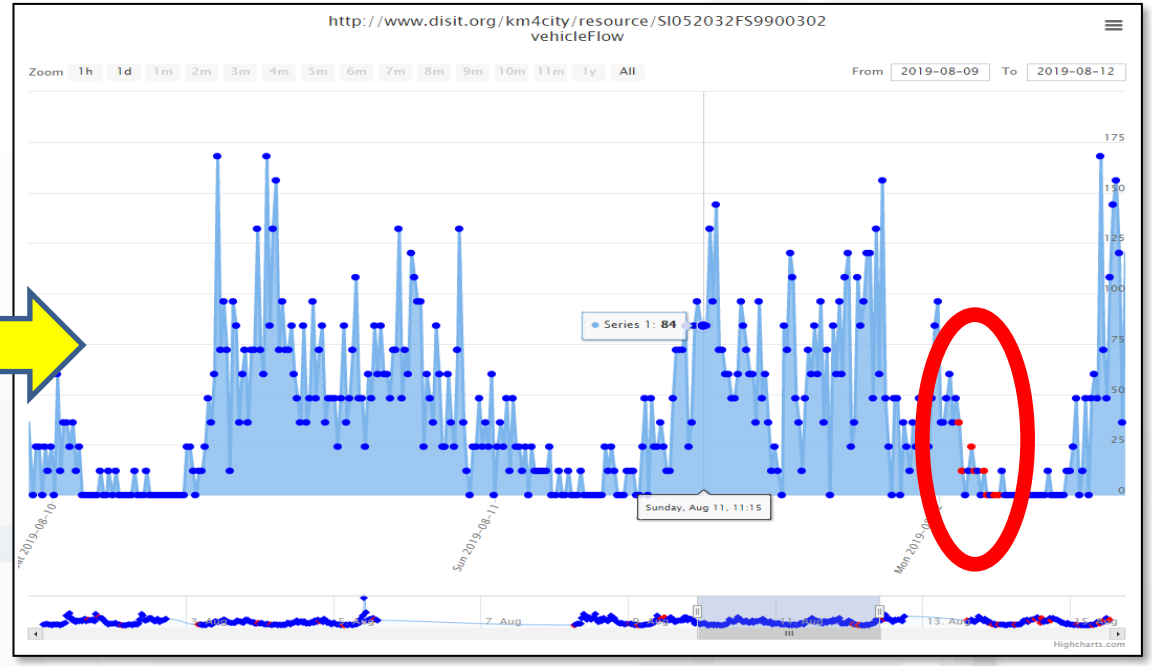
Last Date: 2020-07-21 19:00:00

Last Value:

Value Type	Healthy	Delay (s)	Reason	Healthiness Criteria	Refresh Rate (s)	Data type	Unit	Value	Time Trend
dateObserved	●	61890	undefined	undefined	300	time	timestamp	2020-07-21T17:00:00.000Z	VIEW
deceduti	●	61890	undefined	undefined	300	integer	#	16797	VIEW
dimessi_guariti	●	61890	undefined	undefined	300	integer	#	71775	VIEW
isolamento_domiciliare	●	61890	undefined	undefined	300	integer	#	6838	VIEW
nuovi_attualmente_positivi	●	61890	undefined	undefined	300	integer	#	-131	VIEW
ricoverati_con_sintomi	●	61890	undefined	undefined	300	integer	#	151	VIEW
stato	●	61890	undefined	undefined	300	string	#	ITA	VIEW
tamponi	●	61890	undefined	undefined	300	integer	#	1212468	VIEW
terapia_intensiva	●	61890	undefined	undefined	300	integer	#	21	VIEW
totale_attualmente_positivi	●	61890	undefined	undefined	300	integer	#	7010	VIEW
totale_casi	●	61890	undefined	undefined	300	integer	#	95582	VIEW
totale_ospedalizzati	●	61890	undefined	undefined	300	integer	#	172	VIEW
codice_regione	●	61890	missing value	undefined	300	integer	status		VIEW
denominazione_regione	●	61890	missing value	undefined	300	string	status		VIEW

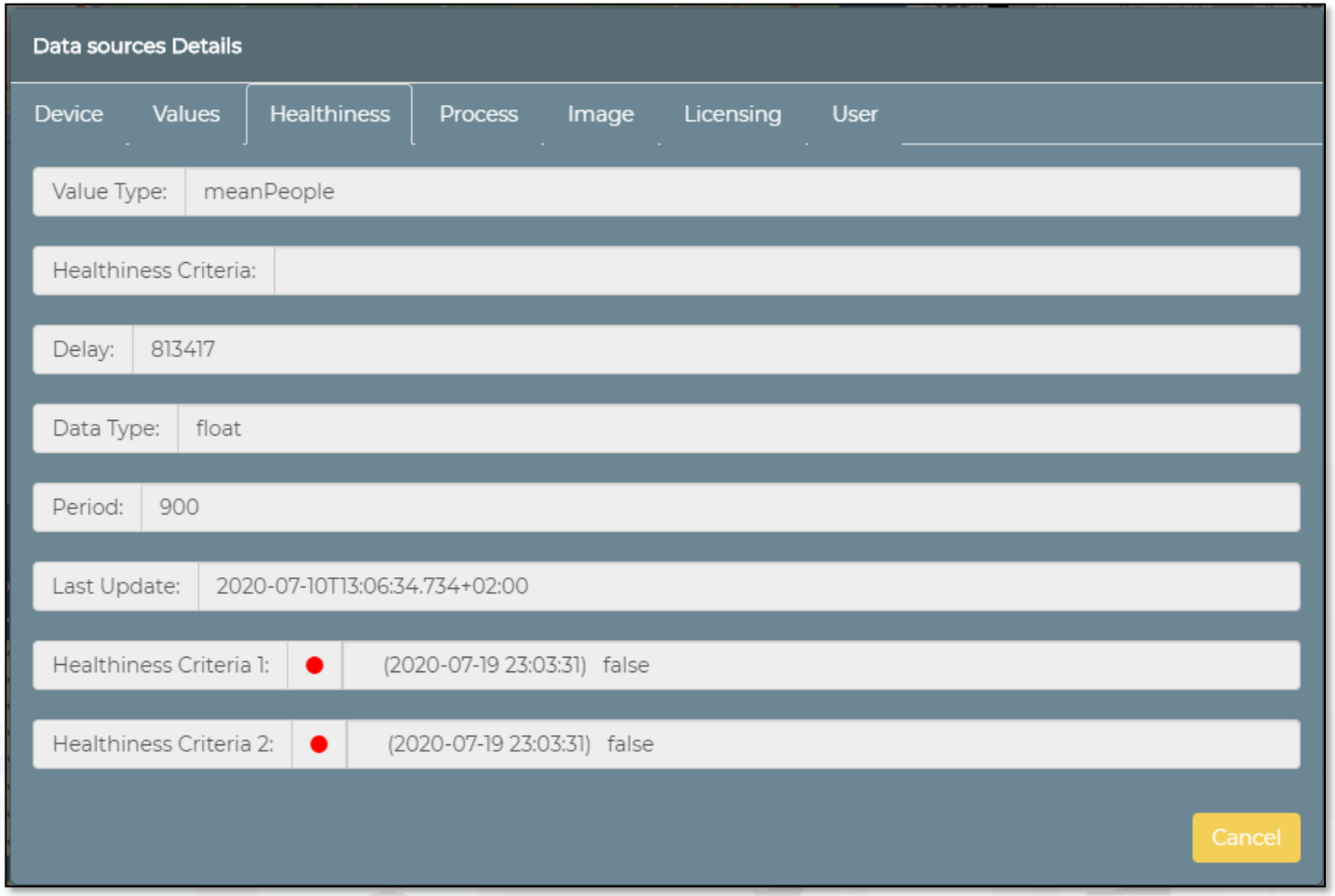
Cancel

- 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)



Some functionalities are limited to certain roles

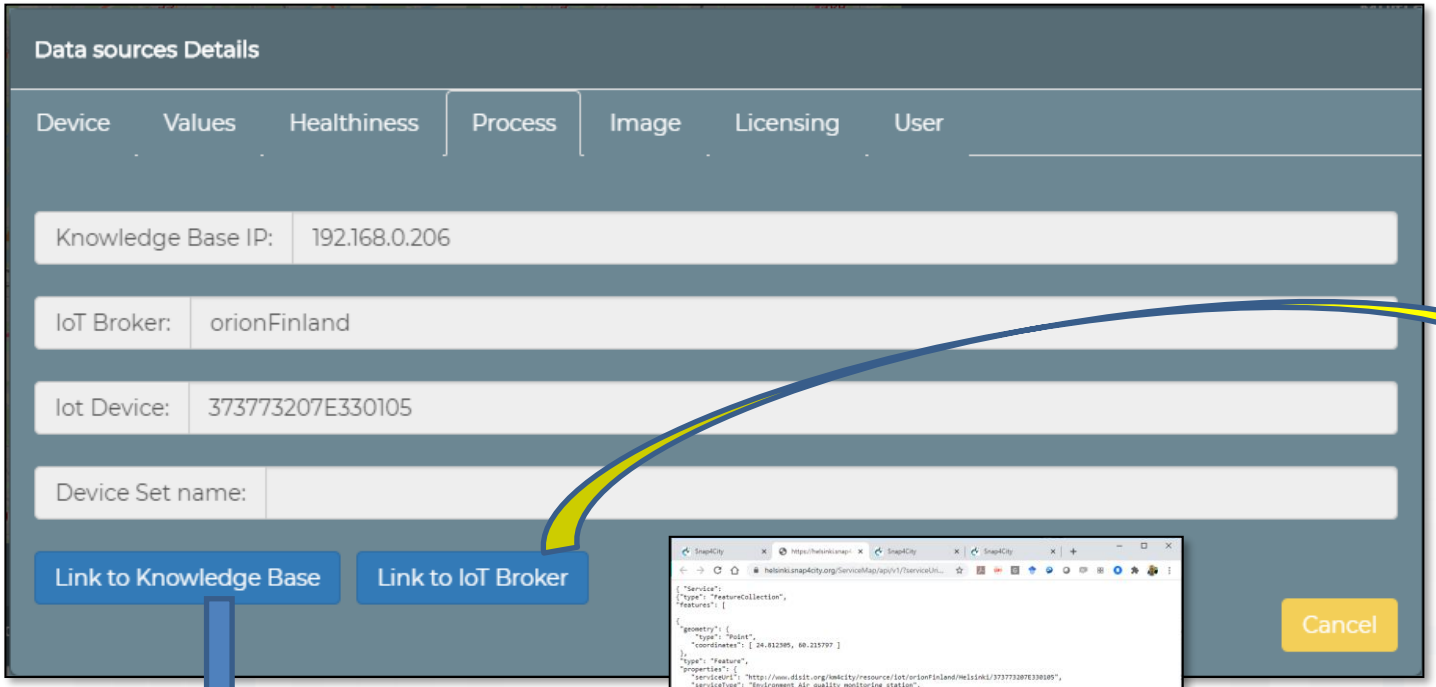
HLT: Sensor Healthiness

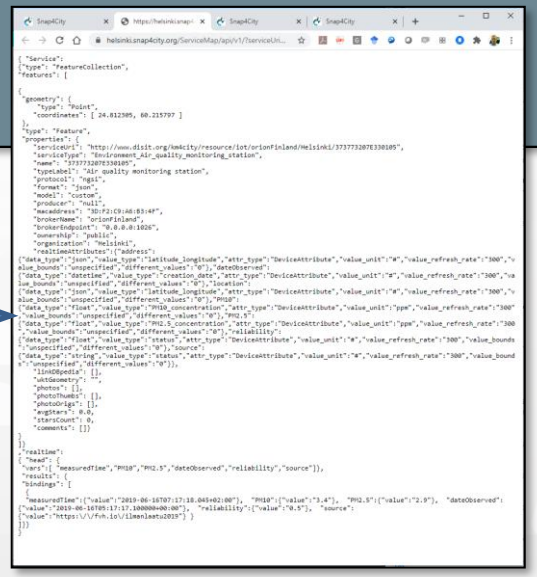


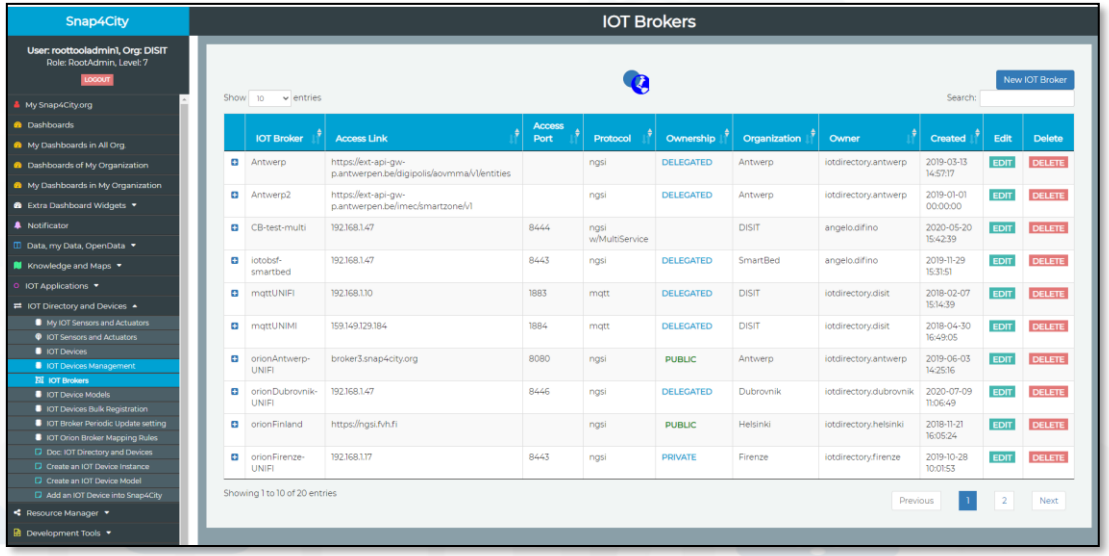
- Two different criteria
  - **H1**: at least an event i nthe last 24 hours
  - **H2**: machine learning for most of Sensors devices

Some functionalities are limited to certain roles

Details regarding the IOT Ingestion process





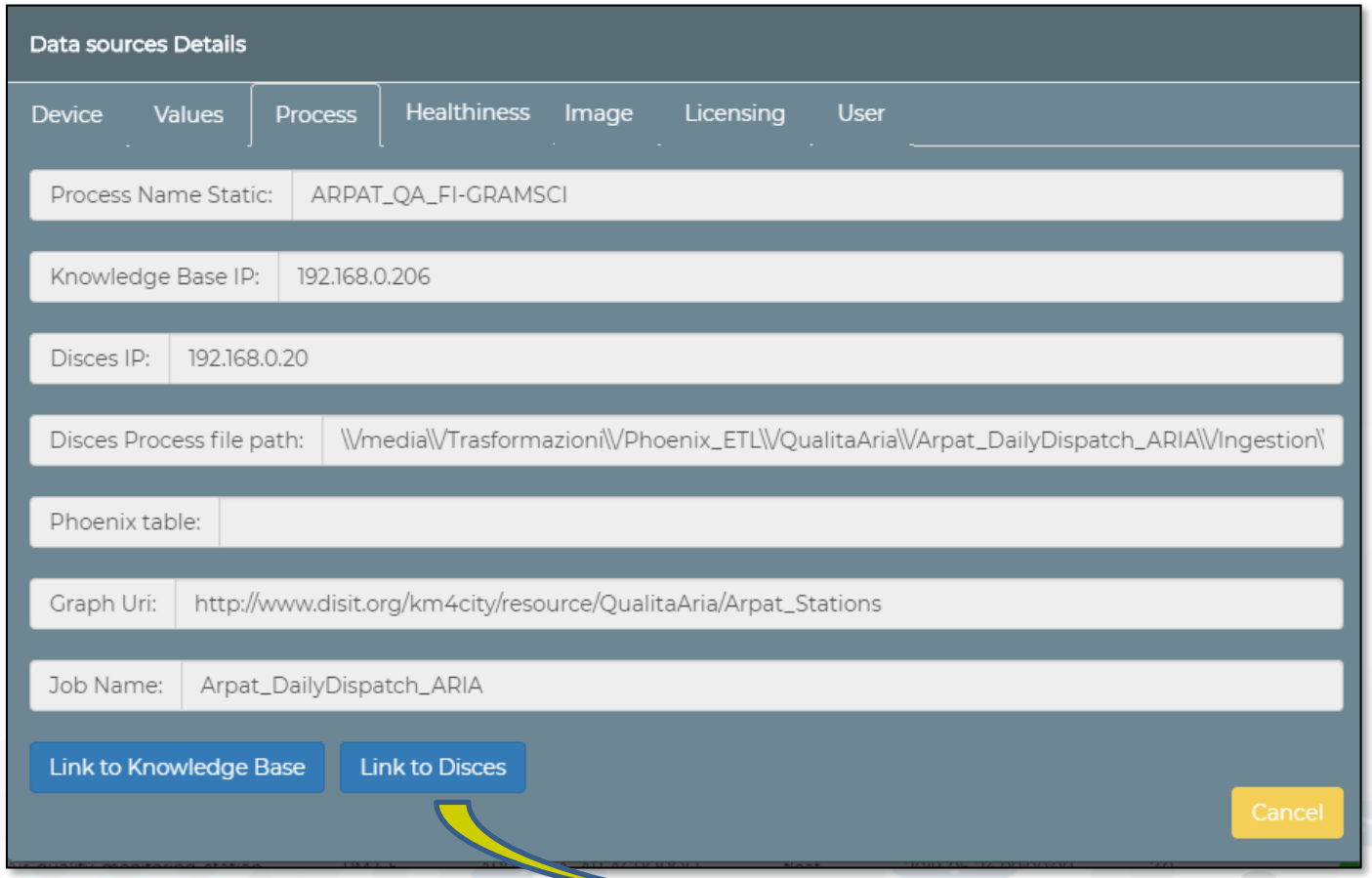


- For IOT Device data
- IOT Broker details

Some functionalities are limited to certain roles

HLT: Sensor

Details regarding the Ingestion process



- For ETL processes
- Scheduling details and status

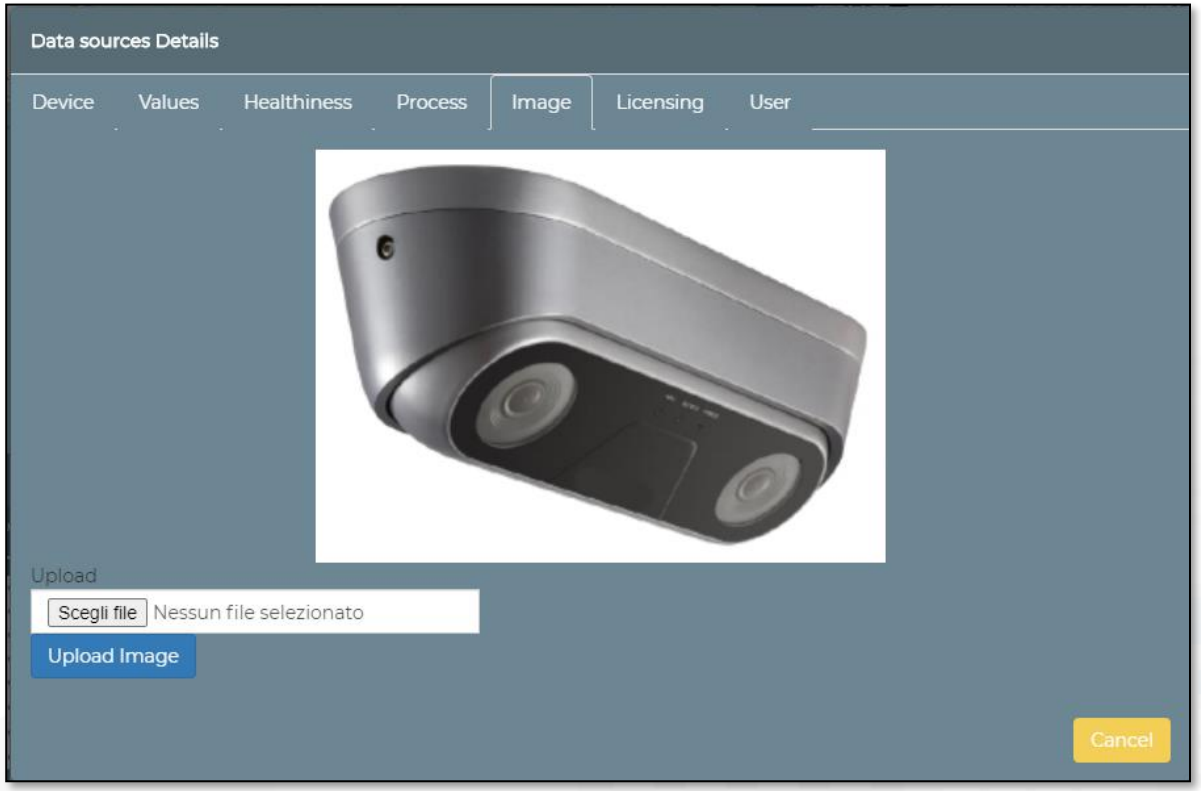
DISCES Process Scheduler view

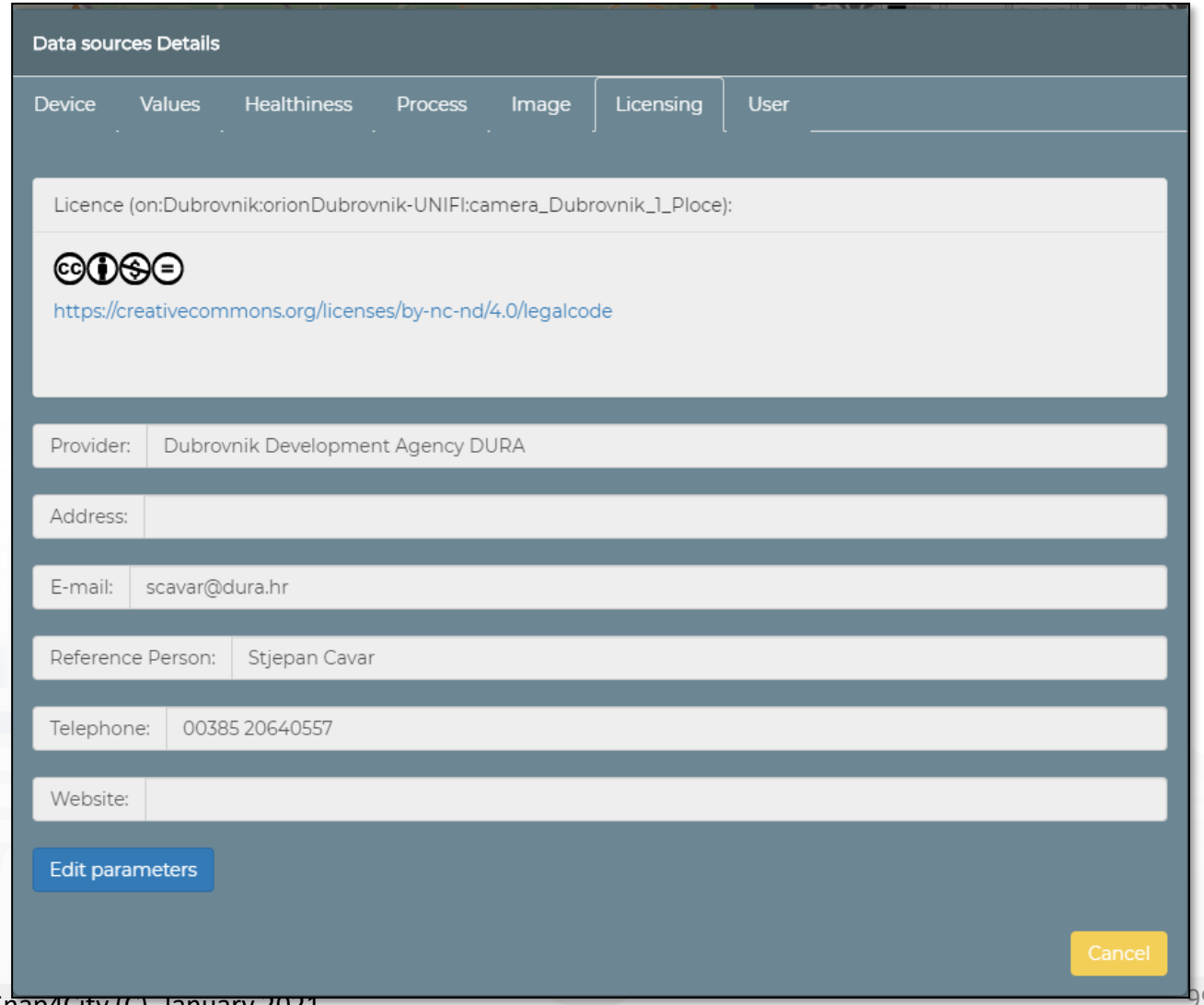


Some functionalities are limited to certain roles

HLT: Sensor

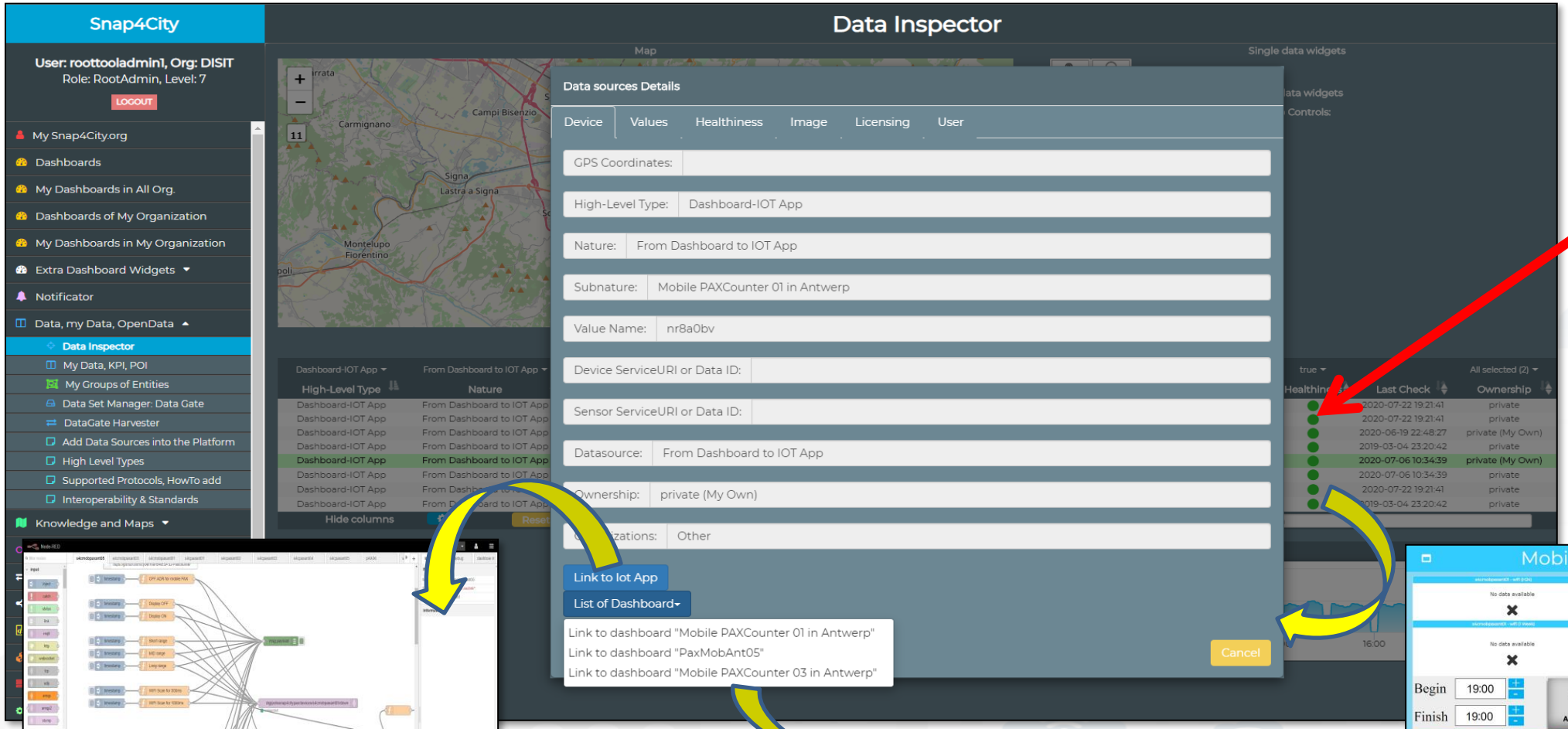
Image of the Devices and Licensing





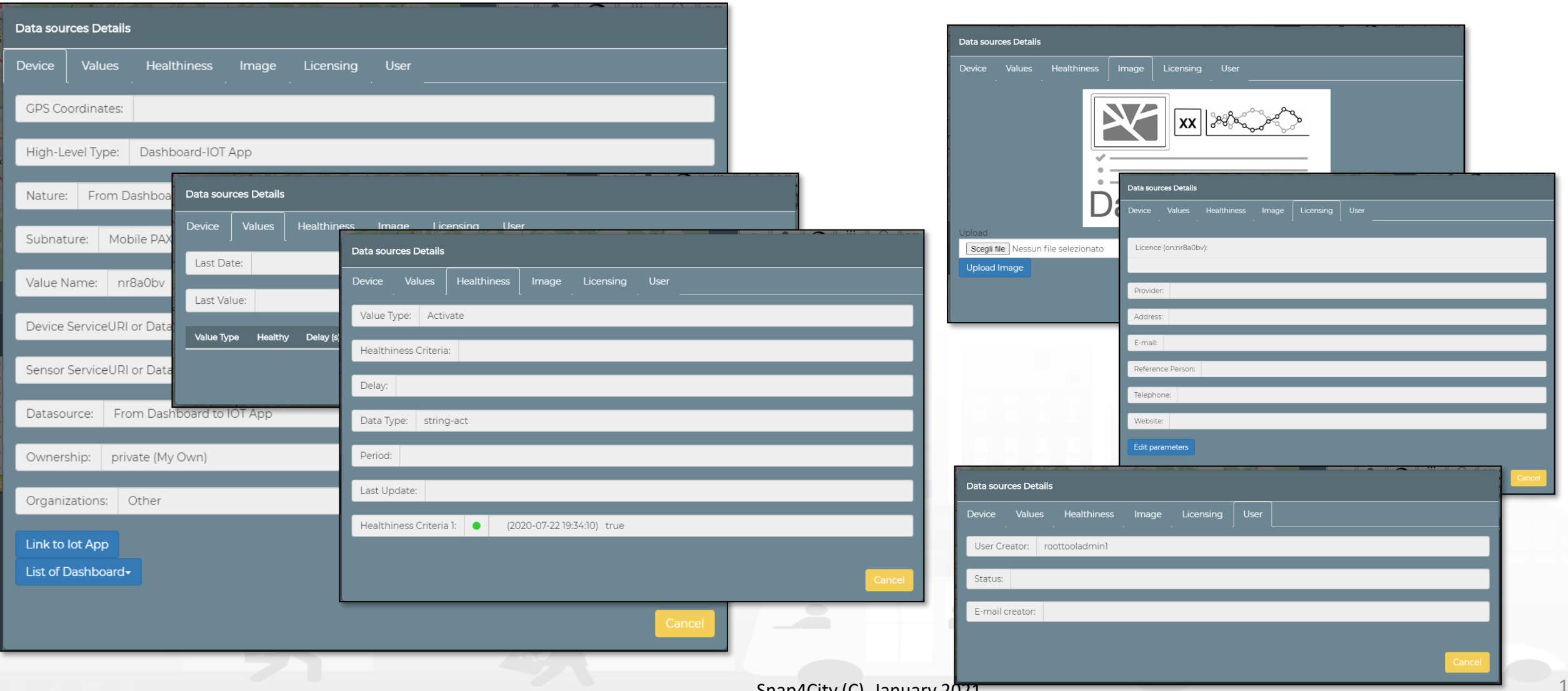
Some functionalities are limited to certain roles

HLT: From Dashboard to IOT APP

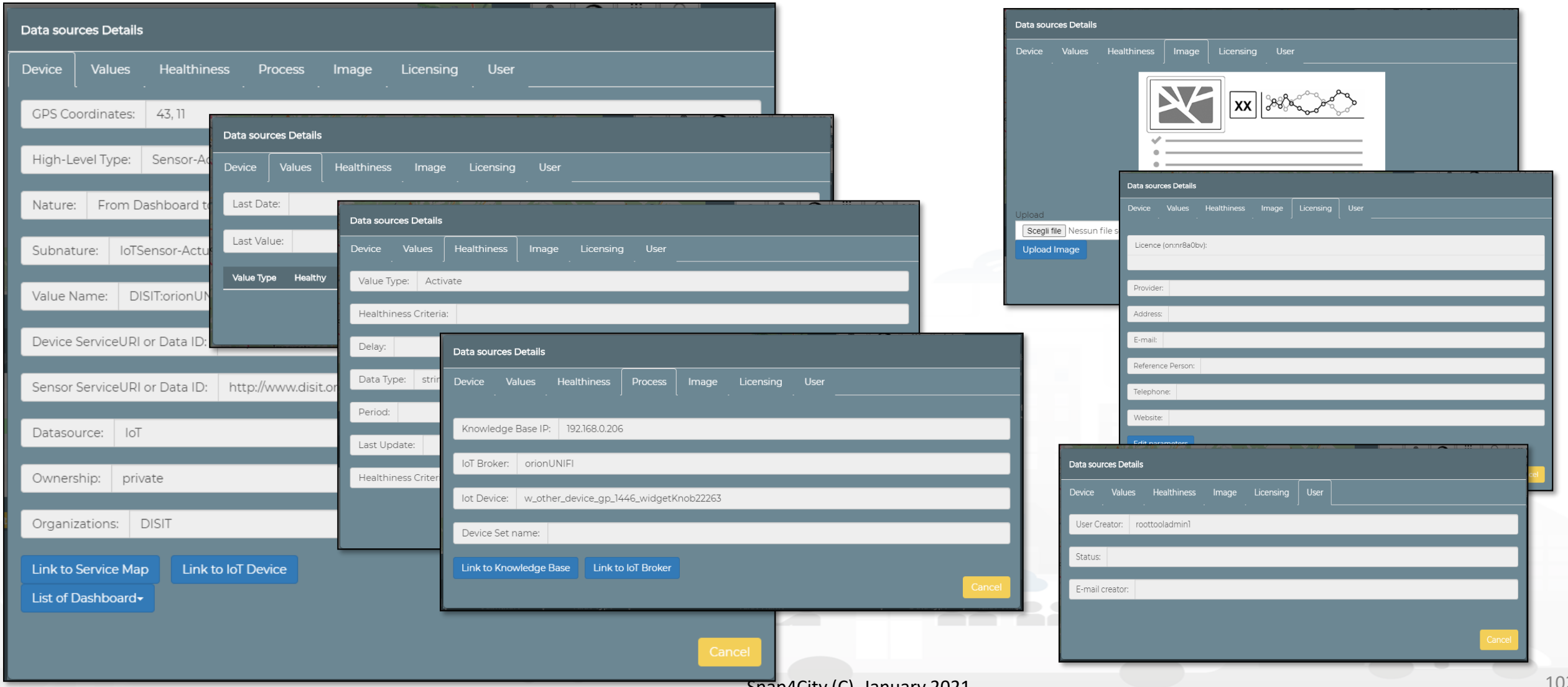


- Click with the mouse on it

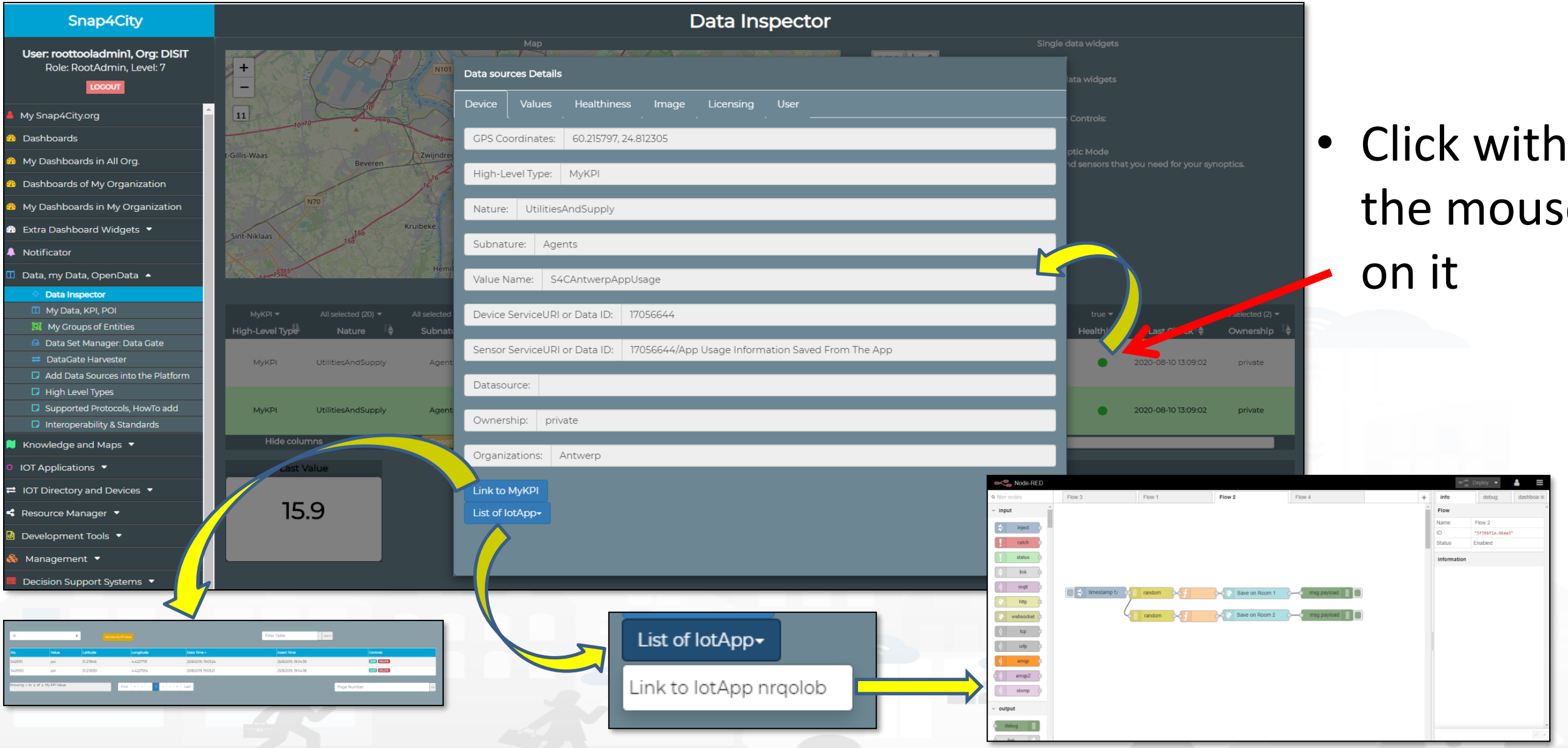
HLT: Dashboard-IOT App (From Dashboard to IOT APP)



HLT: Sensor-Actuator (From Dashboard to IOT Device)



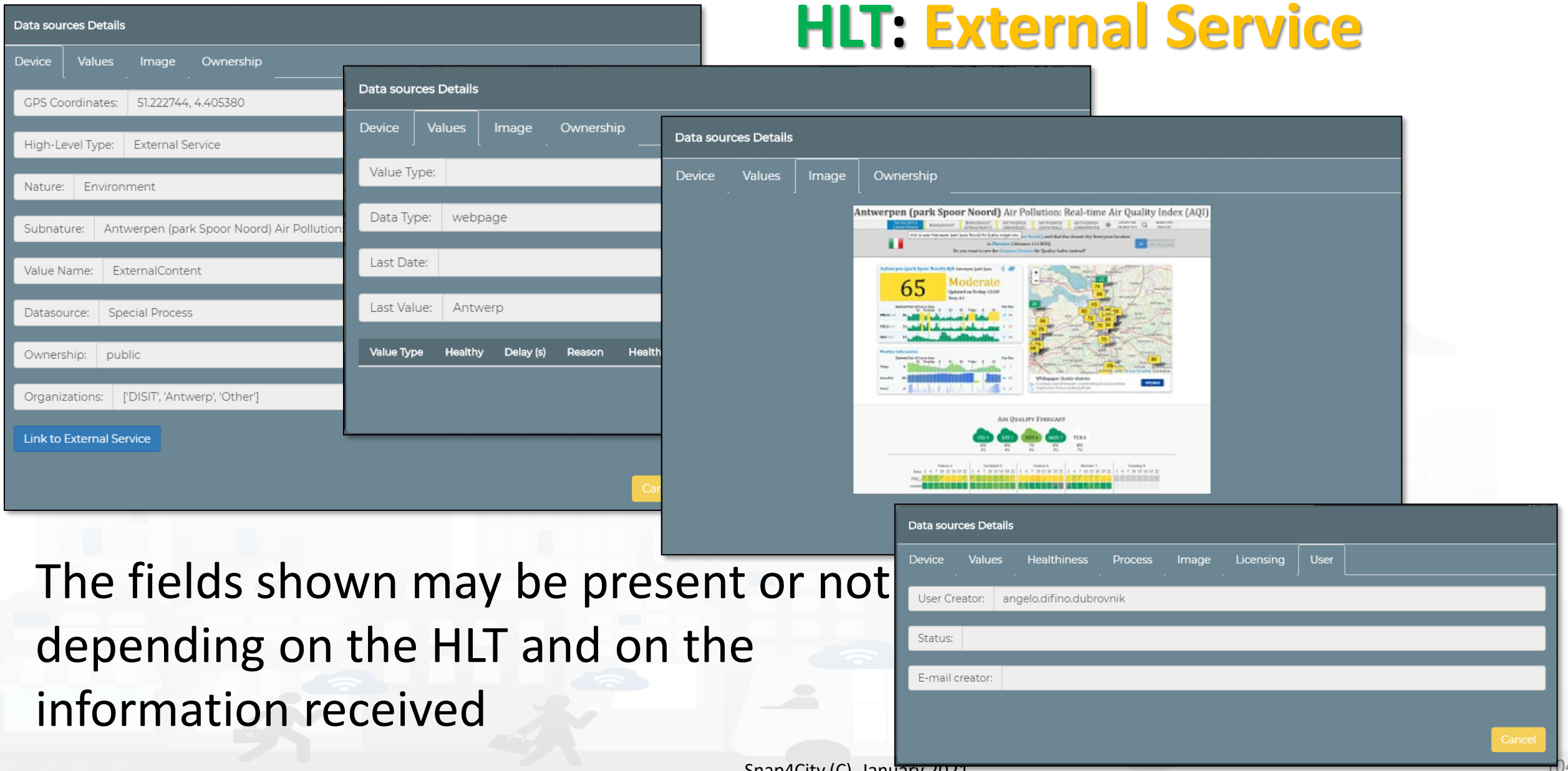
HLT: MyKPI



- Click with the mouse on it

Some functionalities are limited to certain roles

HLT: External Service

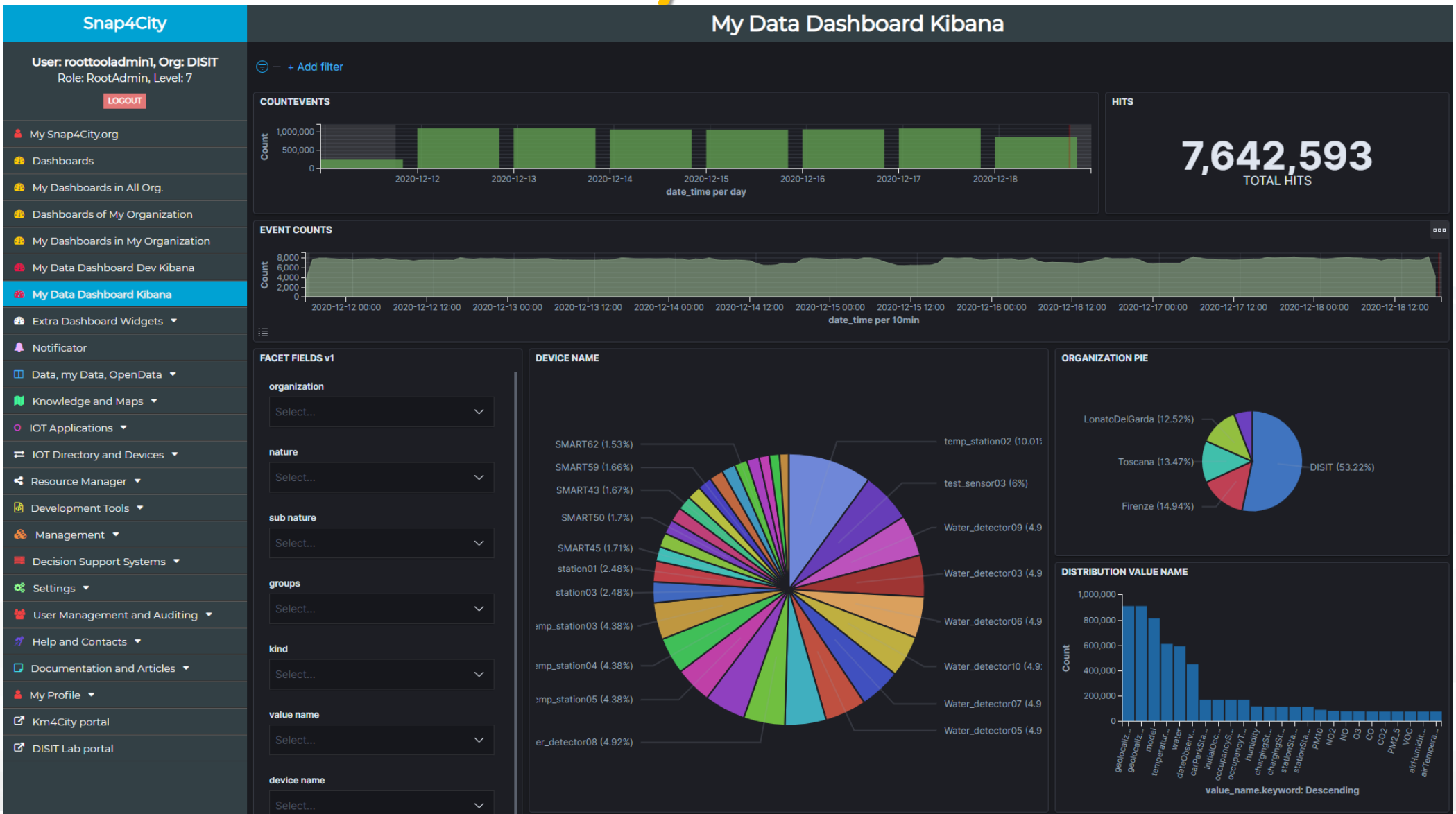


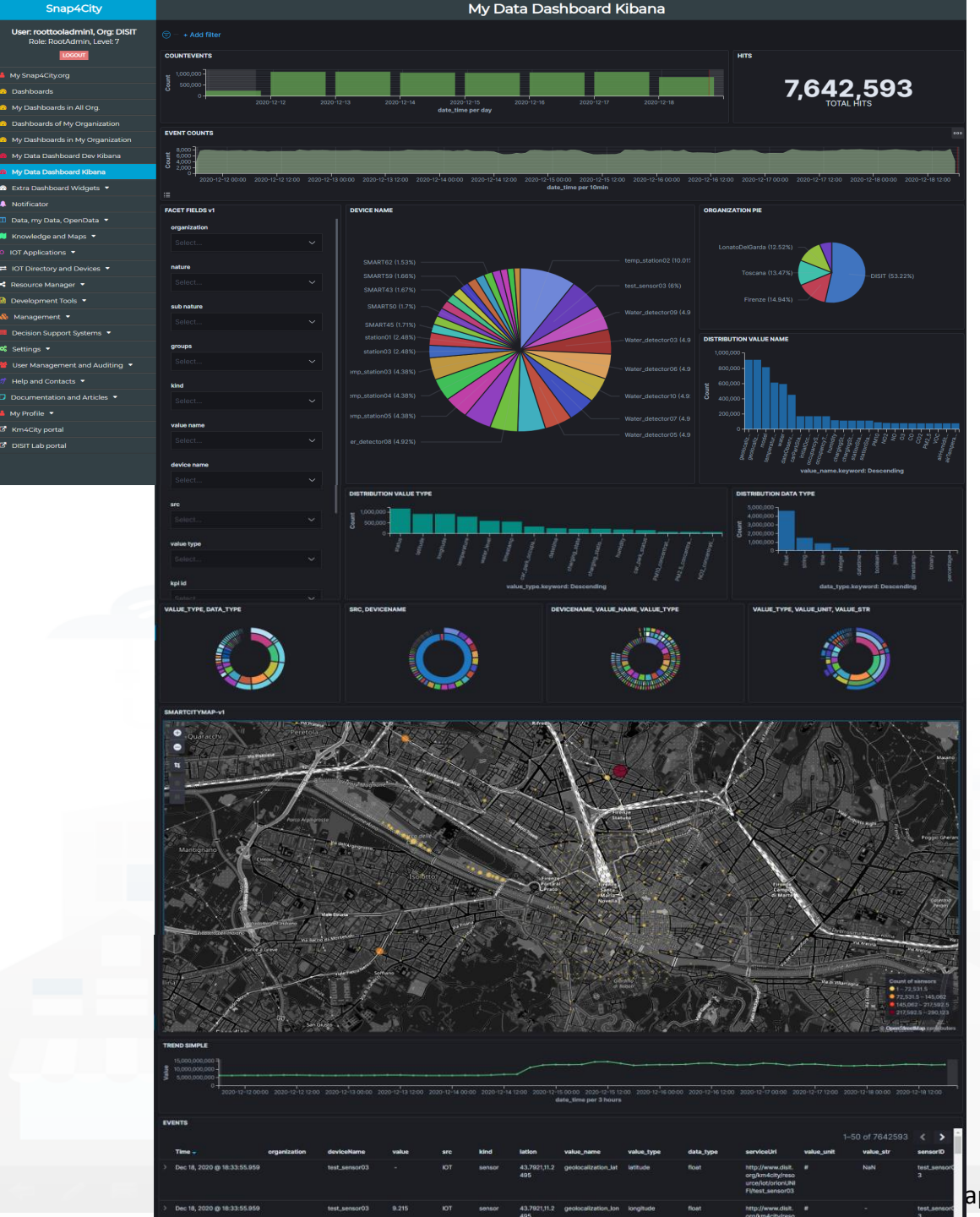
The fields shown may be present or not depending on the HLT and on the information received

DevDash based assessing Elastic Search Data via Kibana



DevDash: My Data Dashboard Kibana



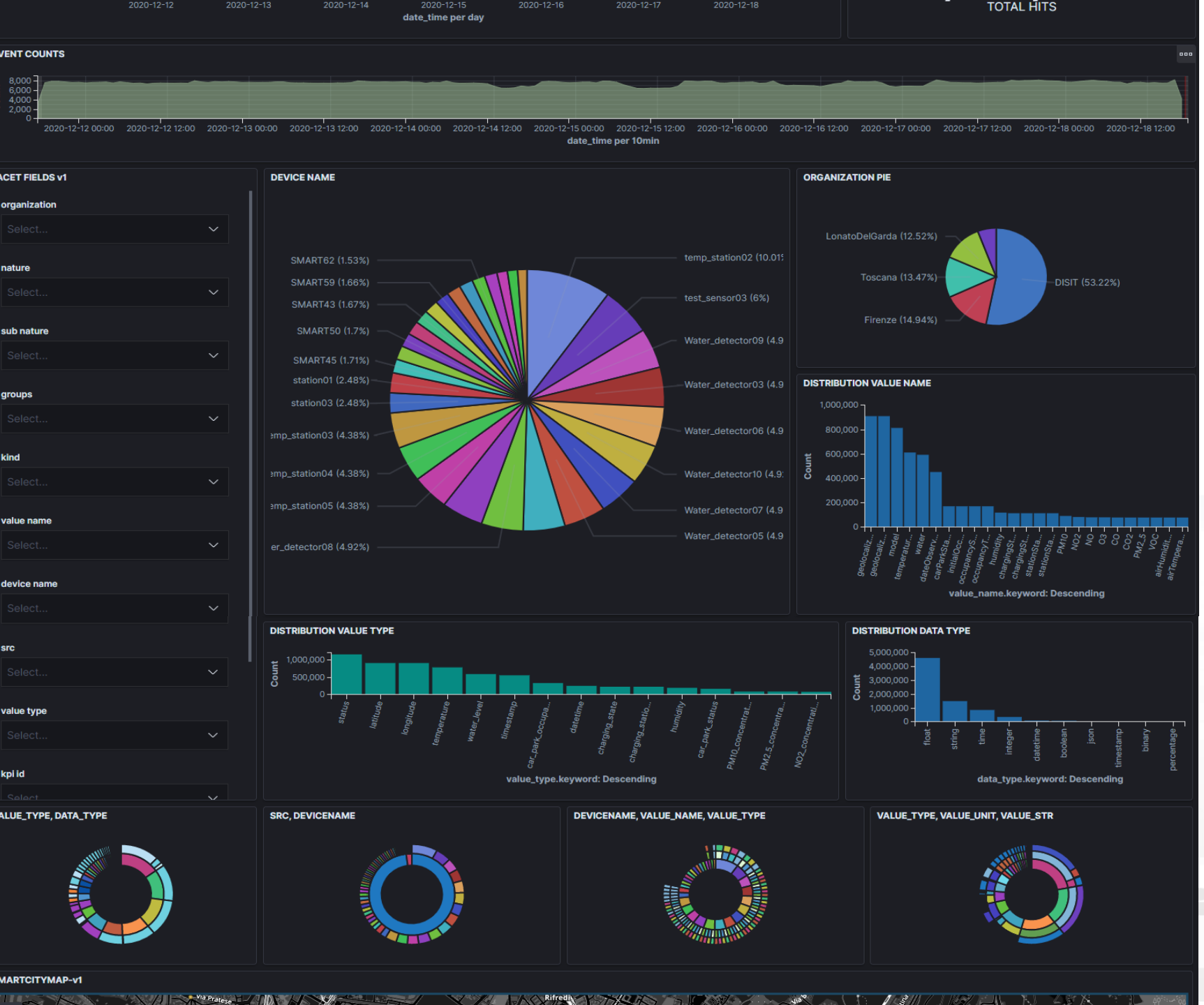


kibana

SNAP4CITY

Business Analysis Dashboards For all kind of users: DevDash

- Dynamic Filtering, Adaptable, …
- Full data details, drill down,…
- Synergic with **Data Inspector** which addresses data relationships, processing and information
- **Only Your Data for**
  - Manager and Area Managers
- **All Accessible Data for**
  - ToolAdmin and RootAdmin



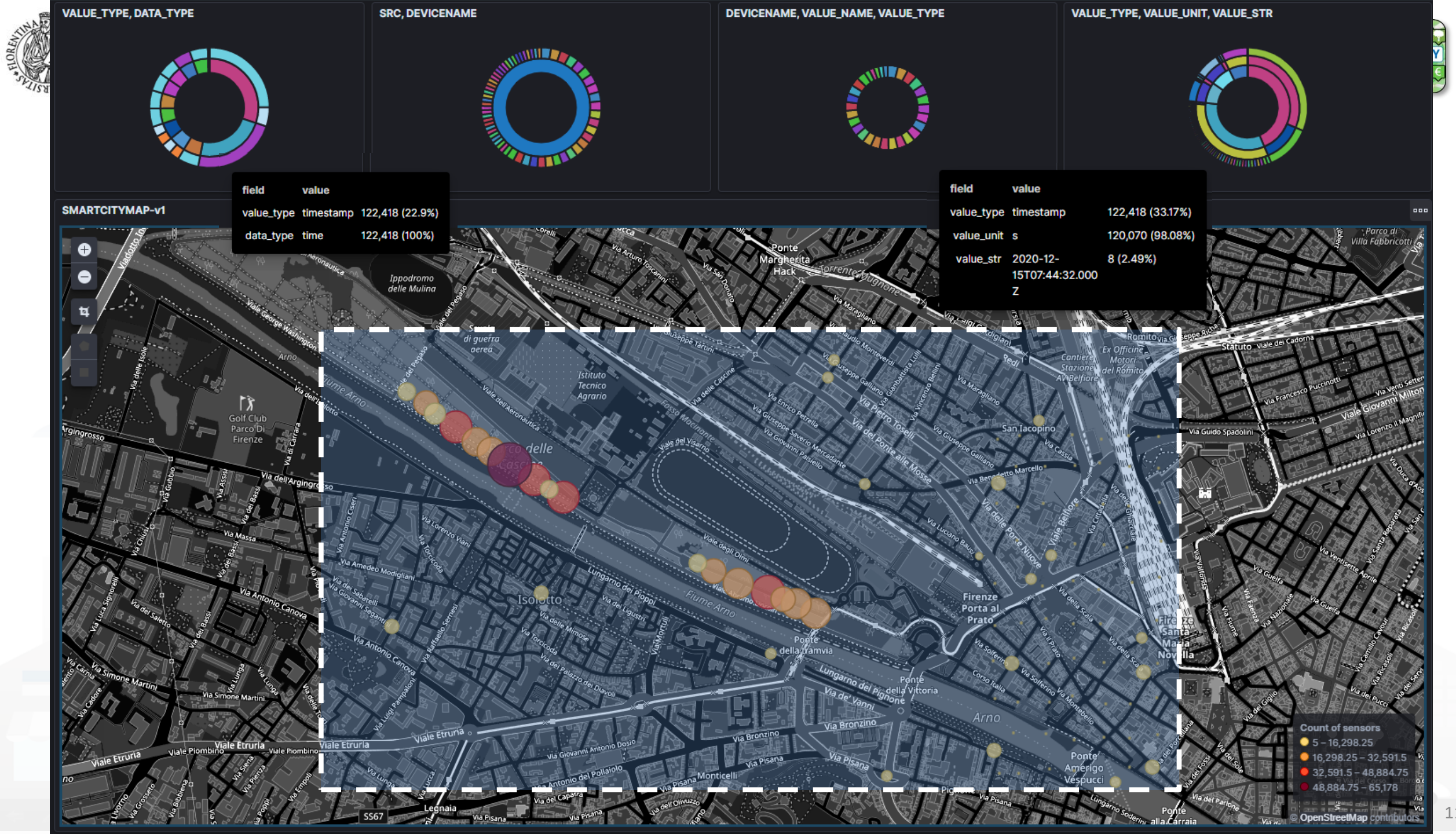
SNAP4CITY

KM4CITY

- Multi faceted Search by
  - Devices
  - Organization
  - Drill on Time
  - Drill on Map
  - Value Types
  - Data Type
  - Value name
  - Data table
  - Etc.

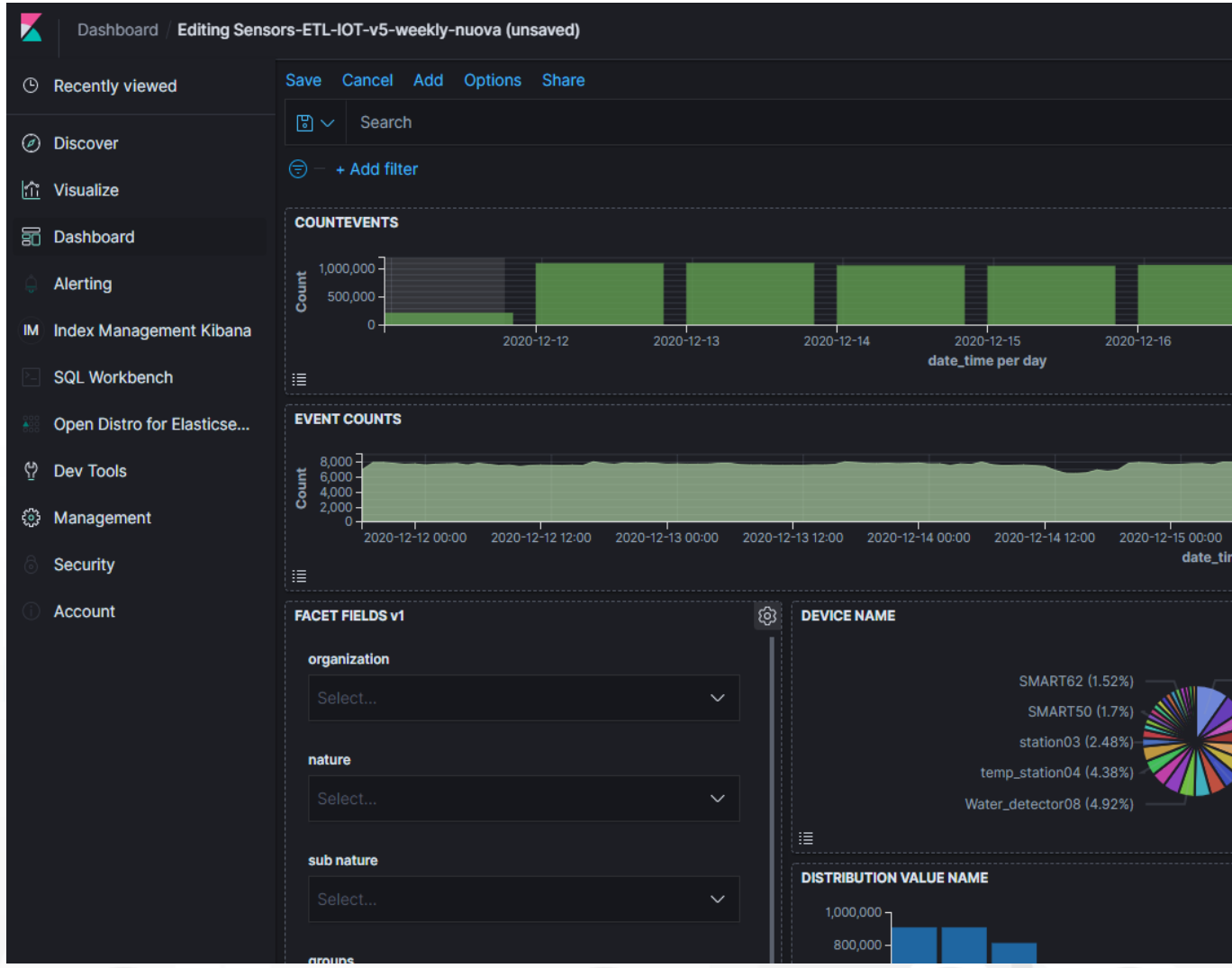
kibana

- Respect Privacy and GDPR


VALUE_TYPE, DATA_TYPE
SRC, DEVICENAME
DEVICENAME, VALUE_NAME, VALUE_TYPE
VALUE_TYPE, VALUE_UNIT, VALUE_STR
field value
value_type timestamp 122,418 (22.9%)
data_type time 122,418 (100%)
field value
value_type timestamp 122,418 (33.17%)
value_unit s 120,070 (98.08%)
value_str 2020-12-15T07:44:32.000Z 8 (2.49%)
SMARTCITYMAP-v1
Count of sensors
5 – 16,298.25
16,298.25 – 32,591.5
32,591.5 – 48,884.75
48,884.75 – 65,178
© OpenStreetMap contributors

My Data Dashboard Customization

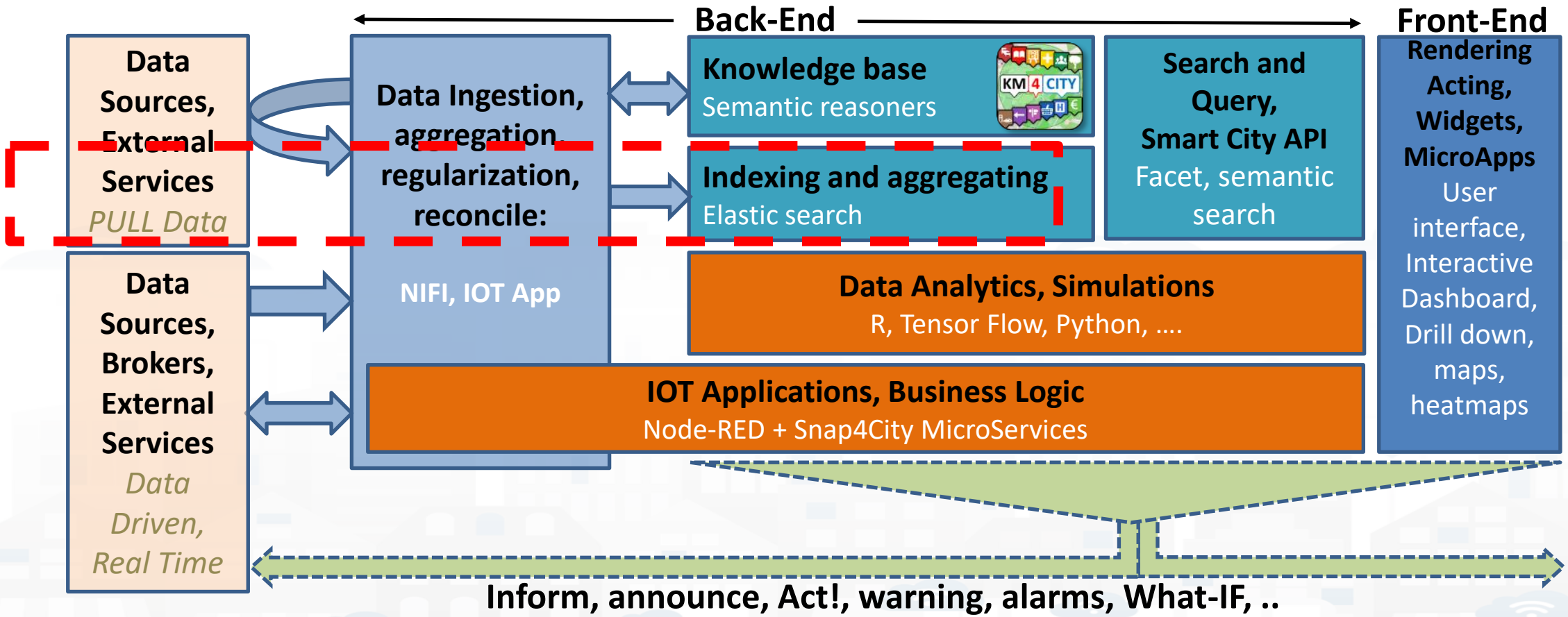
- My Data Dashboards (for Developers and for Managers) can be customized by RootAdmin.
  - Authority for Customizzation can be also extended to other role on Premise solutions
- Customizations is based on Full editing Capabilities of Kibana Elastic Search 7, and with multiple indexes of Snap4City



Set up of the Road Graph on Knowledge Base, performed with an open source tool



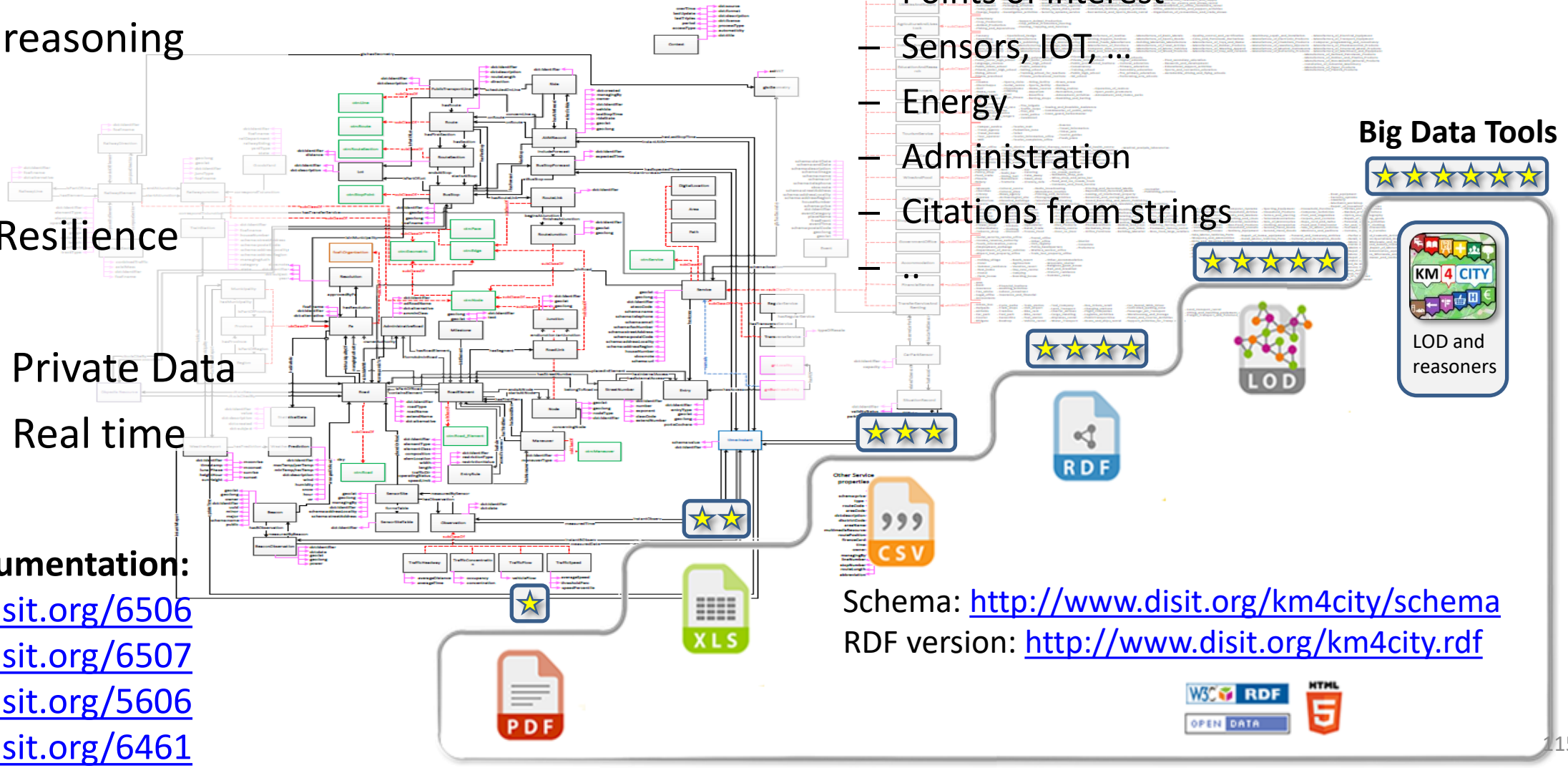
Snap4City Architecture vs Data Ingestion



Km4City: Knowledge Base

- **Multiple DOMAINS**
- Geospatial reasoning
- Temporal reasoning
- Metadata
- Statistics
- Risk and Resilience
- Licensing
- Open and Private Data
- Static and Real time
- IOT/IOE

- Street-Guide
- Mobility and transport
- Points of interest
- Sensors, IOT, …
- Energy
- Administration
- Citations from strings
- ..



Ontology Documentation:
http://www.disit.org/6506
http://www.disit.org/6507
http://www.disit.org/5606
http://www.disit.org/6461

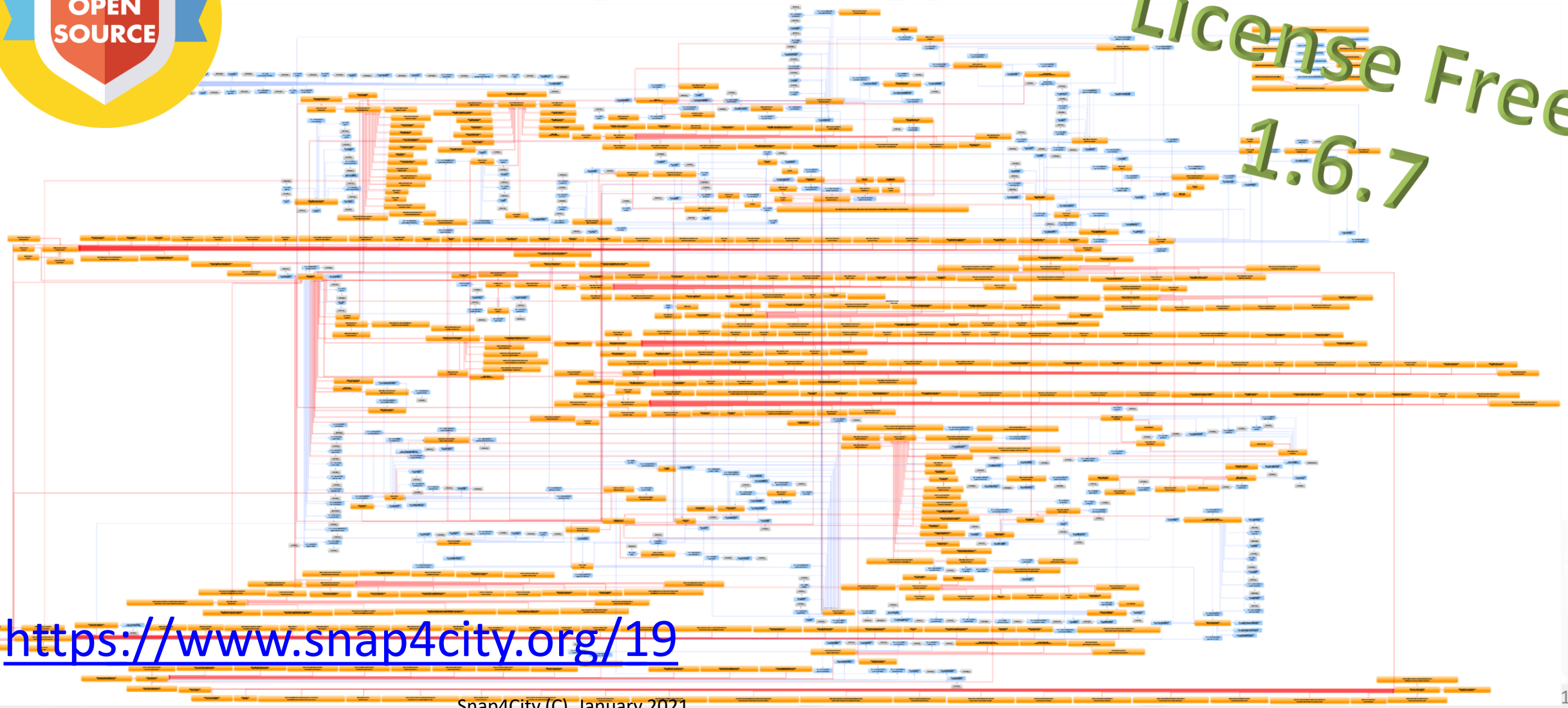
Schema: http://www.disit.org/km4city/schema
RDF version: http://www.disit.org/km4city.rdf

100% OPEN SOURCE

Smart-city Ontology km4city

License Free

1.6.7

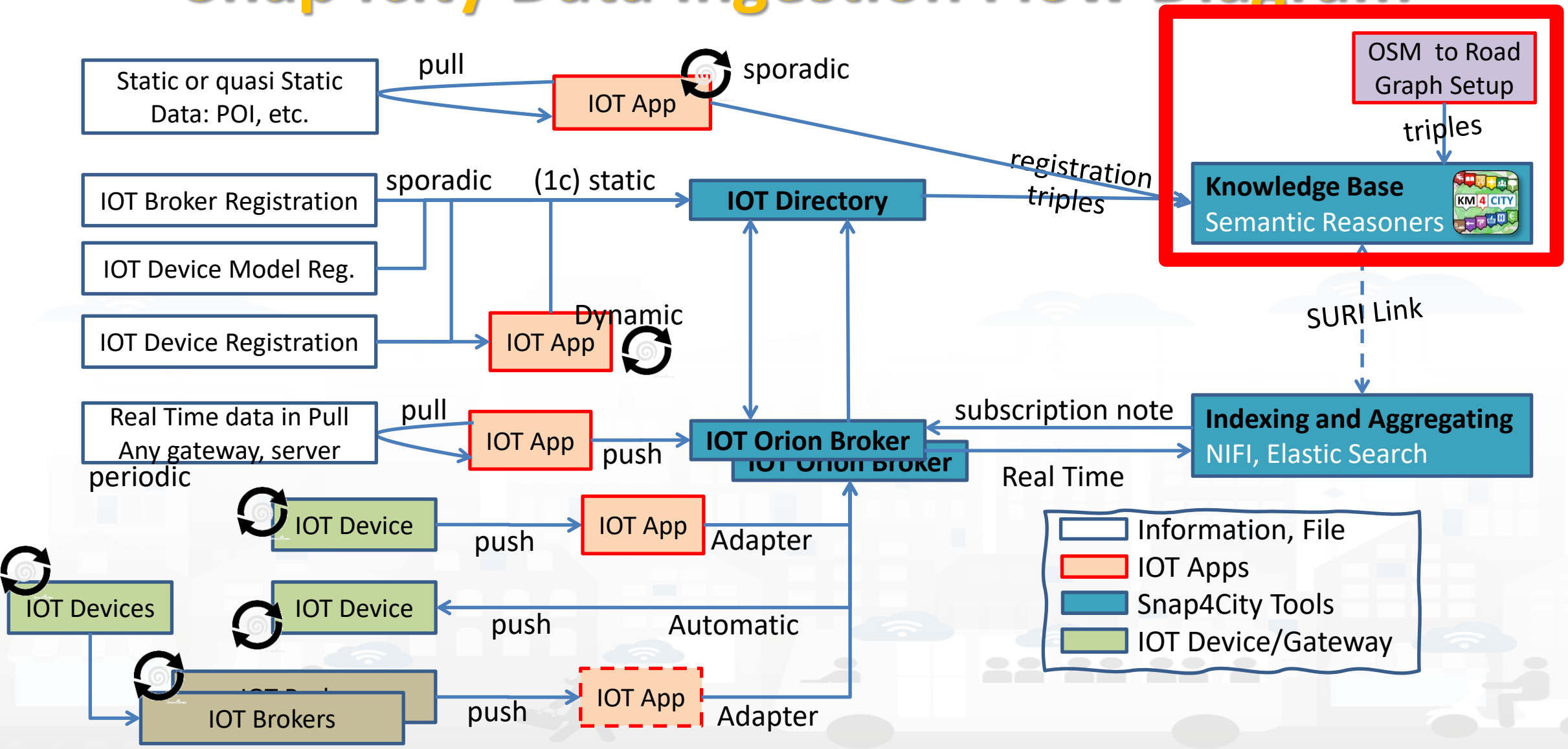


https://www.snap4city.org/19

Set up of the Knowledge Base, KB

- The **KB starts empty**, it has to be **initialized** with the Road Graph(s) of interest.
- **Road Graphs** can be obtained from:
  - GIS of the municipalities, regional govern, etc.
  - Open Street Map, OSM
  - Etc.
  - See this note on KM vs OSM: https://www.snap4city.org/397
- Snap4City provides a tool for **feeding the KB with OSM**
  - TC5.10- Open Street Map ingestion process
  - **OSM2KM4C** tool is included into KBSM, VM and Docker https://www.snap4city.org/471
  - Tool: https://github.com/disit/osm2km4c
- The load of a city of 1 Million of inhabitants can be done in few hours.

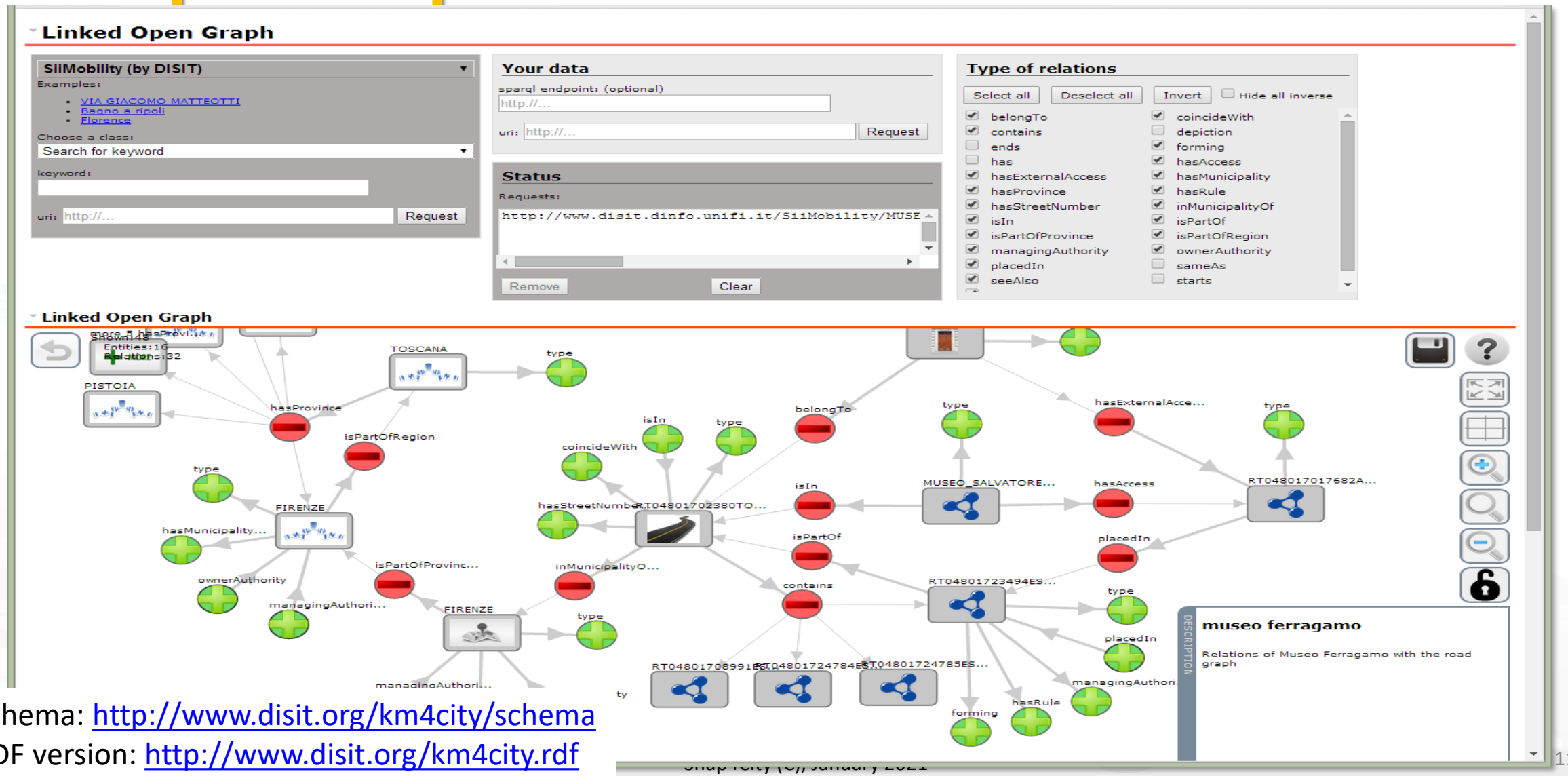
Snap4city Data Ingestion Flow Diagram



Linked Open Data

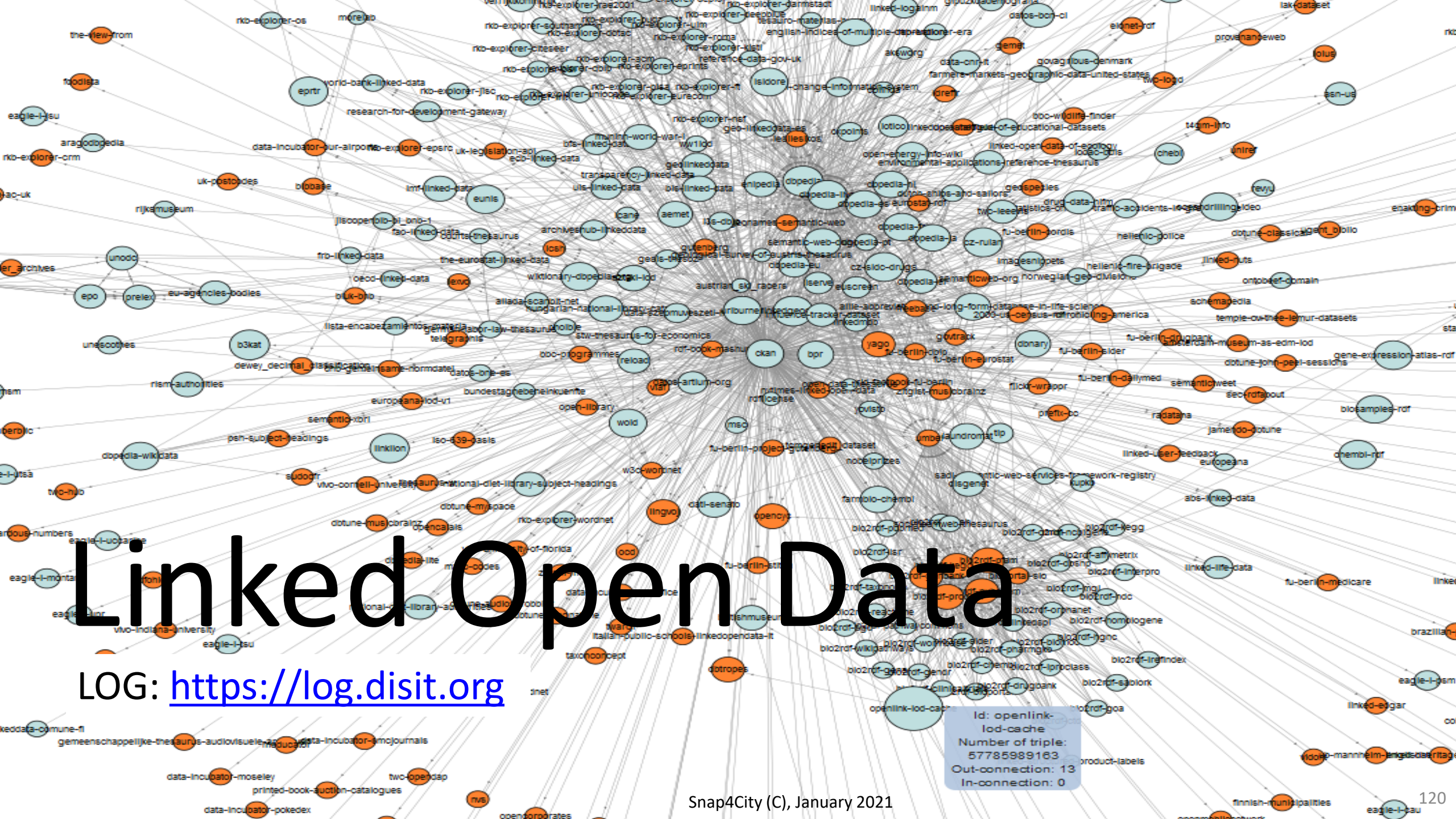
Linked Open Graph

LOG: https://log.disit.org

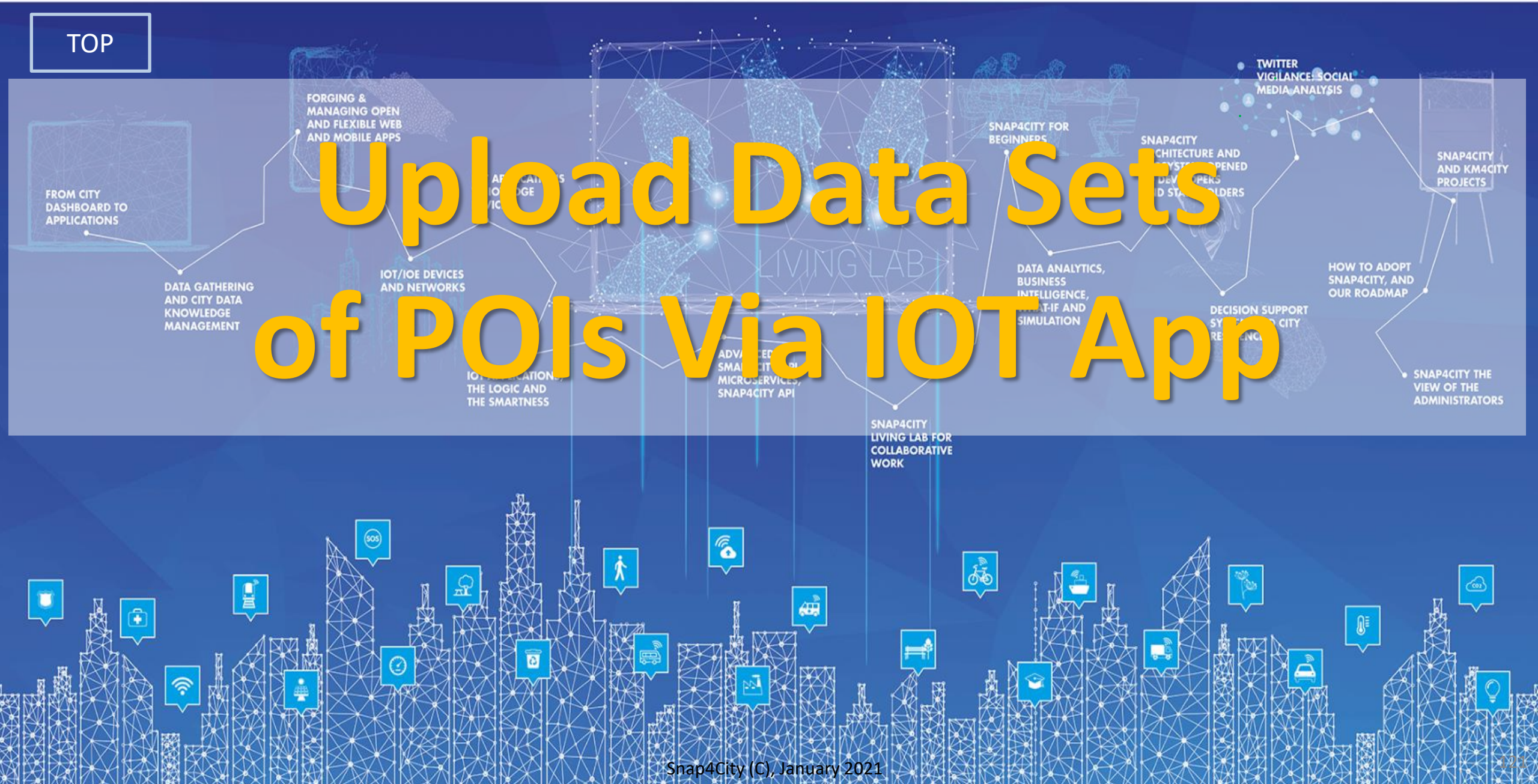


Schema: http://www.disit.org/km4city/schema

RDF version: http://www.disit.org/km4city.rdf

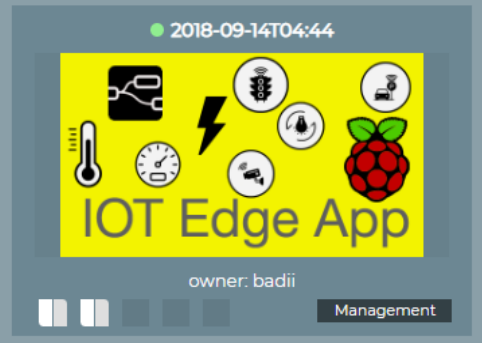

Linked Open Data
LOG: https://log.disit.org
Id: openlink-lod-cache
Number of triple: 57785989163
Out-connection: 13
In-connection: 0

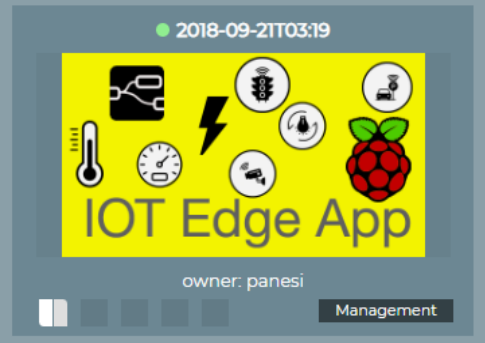
TOP

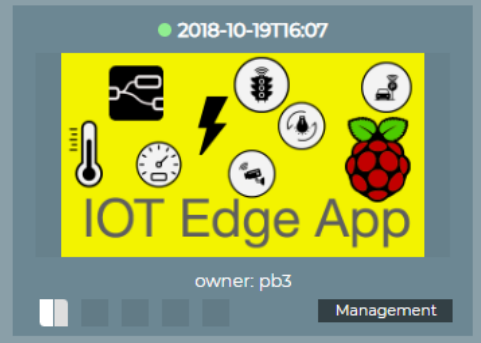


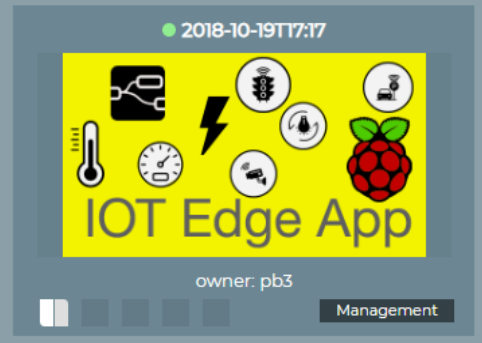
Upload Data Sets of POIs Via IOT App

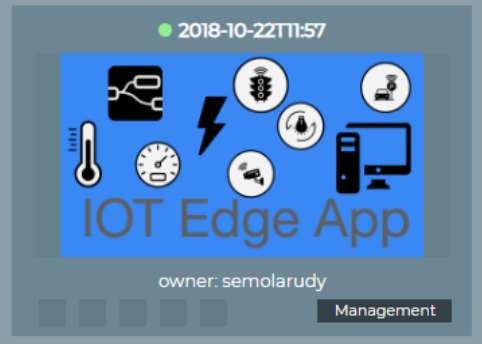
IOT Applications

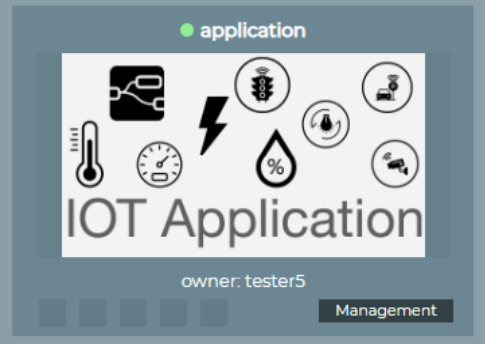


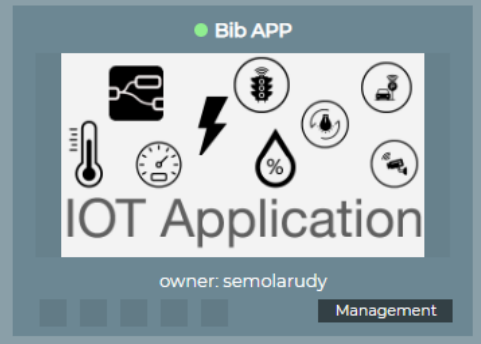


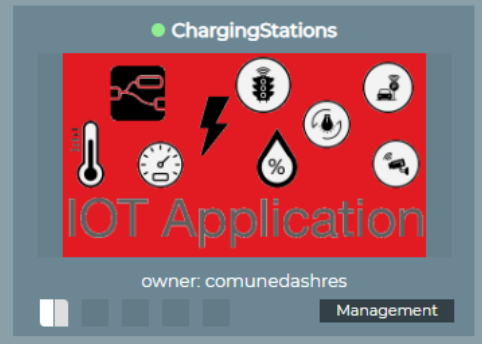


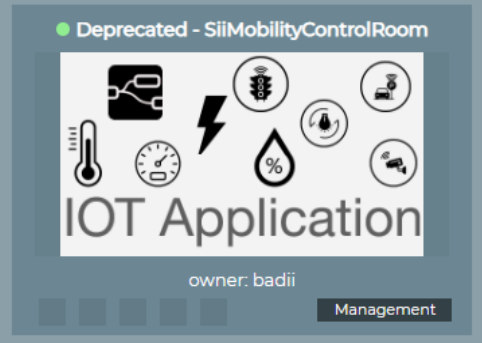






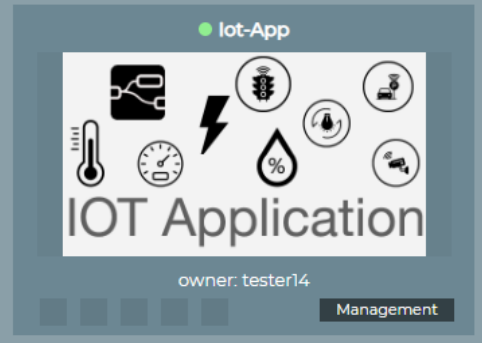








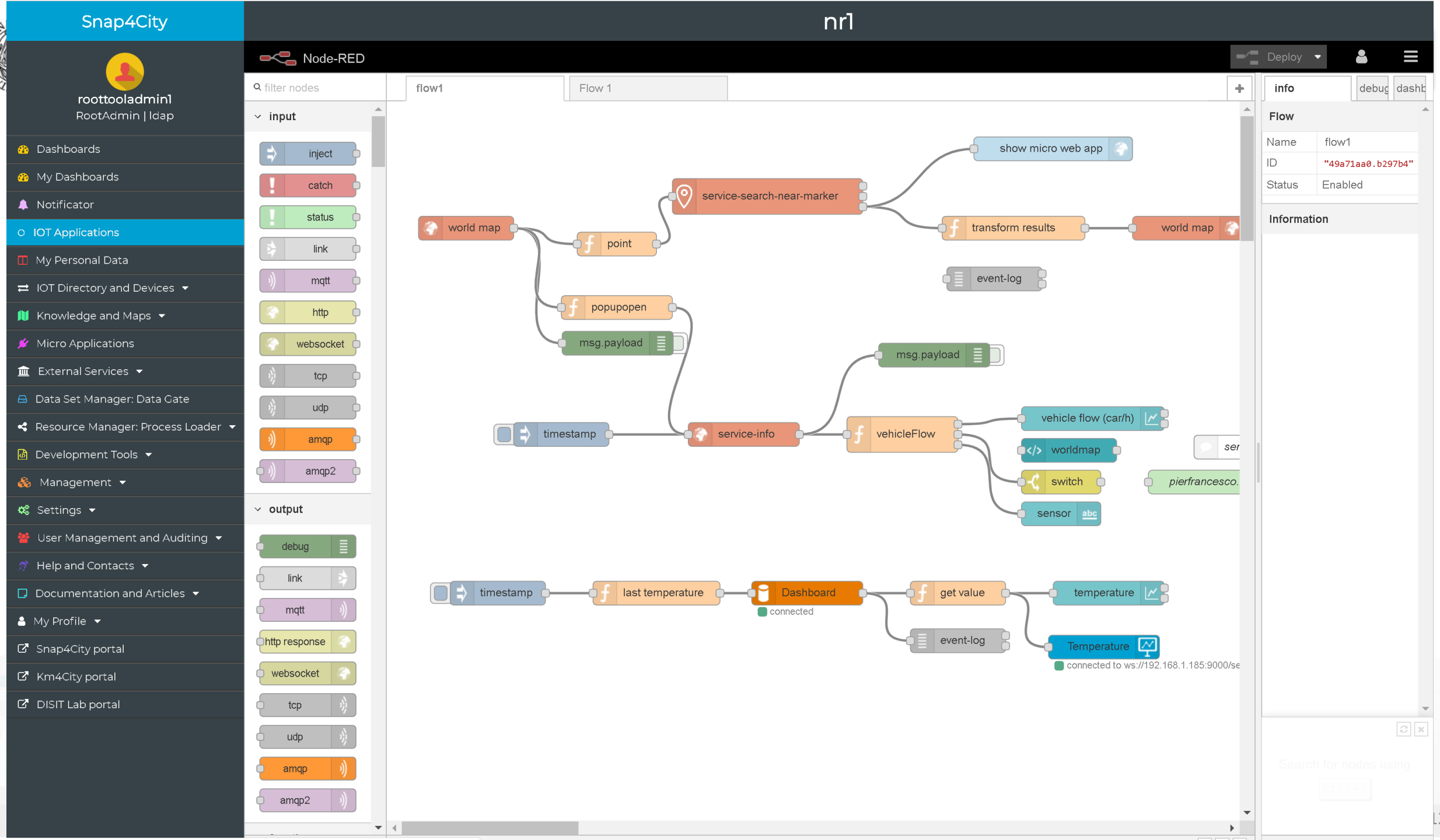




HOW TO learn creating IOT Applications

- Follow the IOT Application development tutorial from Snap4City
  - 3° Part
  - https://www.snap4city.org/577
  - https://www.snap4city.org/download/video/course2020/iot/

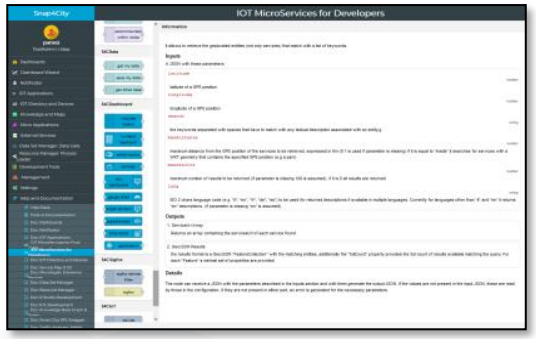



Snap4City
nr1
roottooladmin1
RootAdmin | ldap
Dashboards
My Dashboards
Notificator
IOT Applications
My Personal Data
IOT Directory and Devices
Knowledge and Maps
Micro Applications
External Services
Data Set Manager: Data Gate
Resource Manager: Process Loader
Development Tools
Management
Settings
User Management and Auditing
Help and Contacts
Documentation and Articles
My Profile
Snap4City portal
Km4City portal
DISIT Lab portal
Node-RED
Deploy
filter nodes
input
inject
catch
status
link
mqtt
http
websocket
tcp
udp
amqp
amqp2
output
debug
link
mqtt
http response
websocket
tcp
udp
amqp
amqp2
flow1
Flow 1
world map
point
service-search-near-marker
show micro web app
transform results
world map
event-log
popupopen
msg.payload
msg.payload
timestamp
service-info
vehicleFlow
vehicle flow (car/h)
worldmap
switch
sensor
timestamp
last temperature
Dashboard
connected
get value
temperature
event-log
Temperature
connected to ws://192.168.1.185:9000/se
info
debug
dashb
Flow
Name
flow1
ID
"49a71aa0.b297b4"
Status
Enabled
Information

IOT Applications Development

IOT Discovering

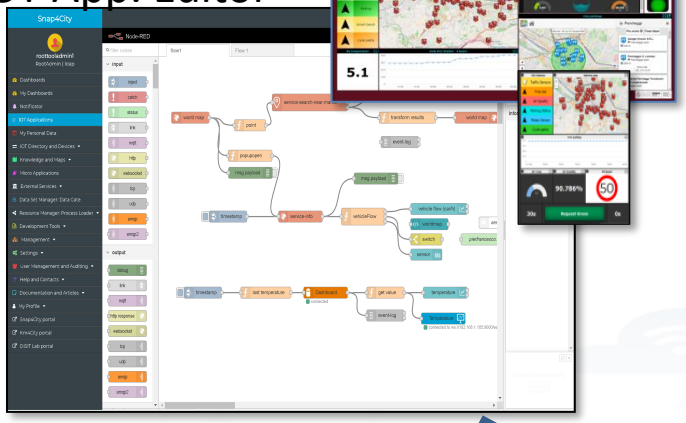
MicroServices collections



My IOT Applications



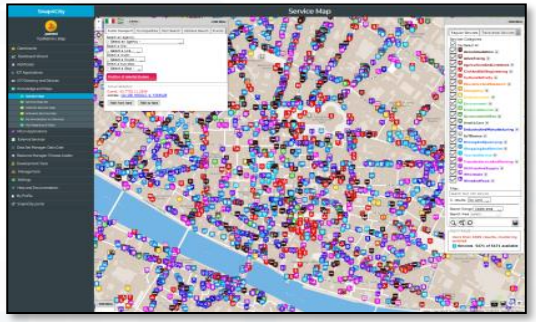
IOT App. Editor



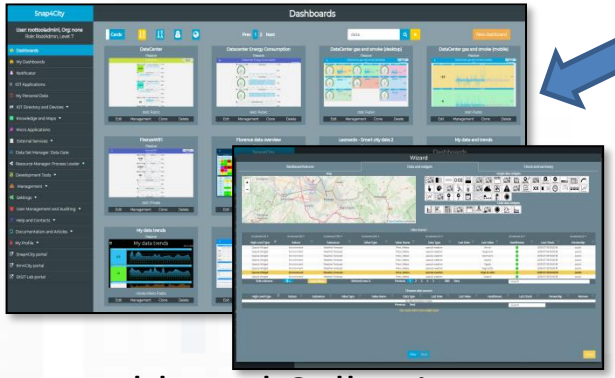


Generating IOT App With Dashboard





ServiceMap Discovery



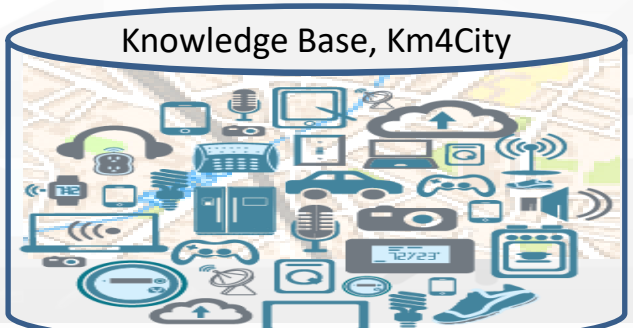
Dashboard Collection, Editor and Wizard

Sharing/saving reusing IOT App



Resource Manager

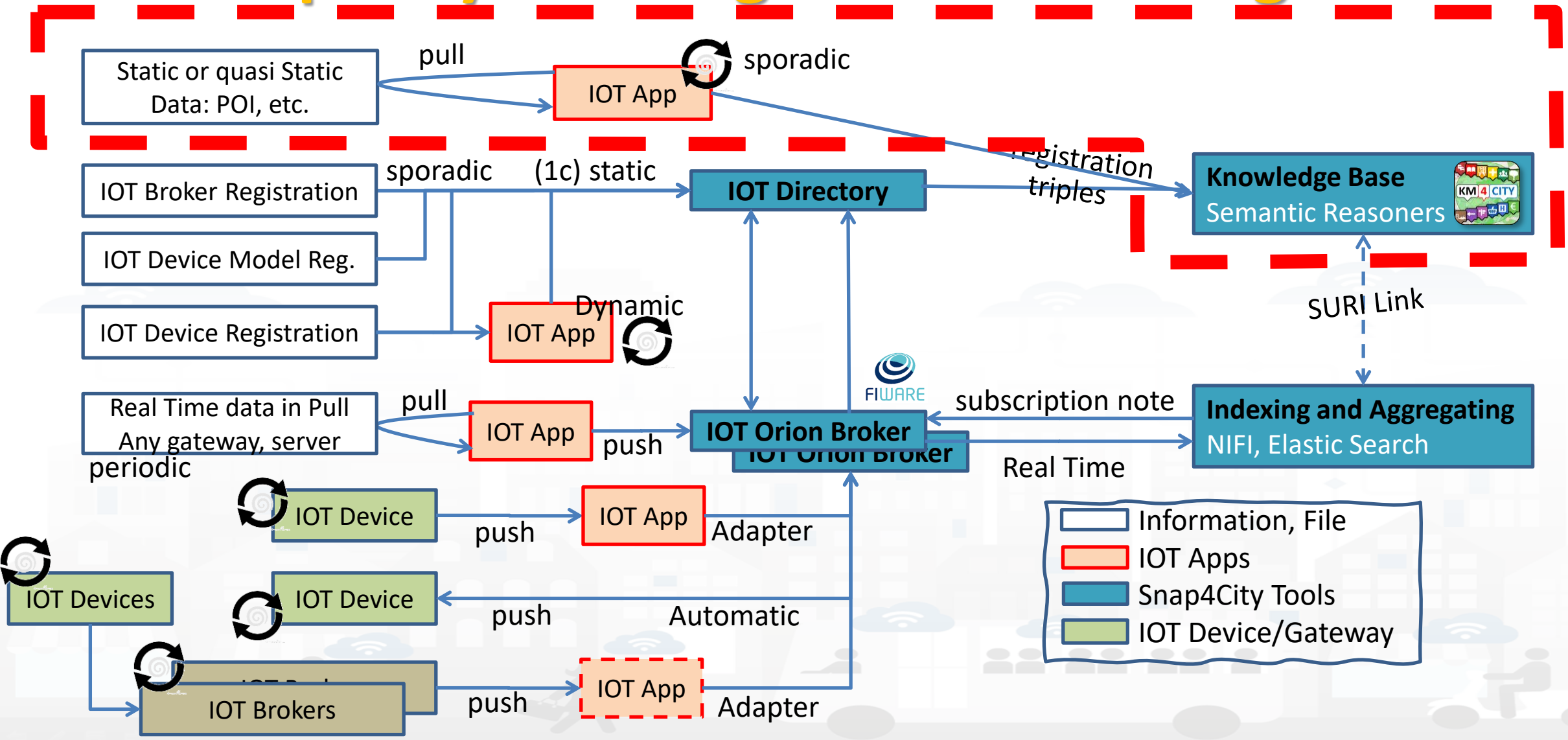




Static Data Ingestion via IOT Apps, API, and Files



Snap4city Data Ingestion Flow Diagram



Regularizing Static data via IOT App

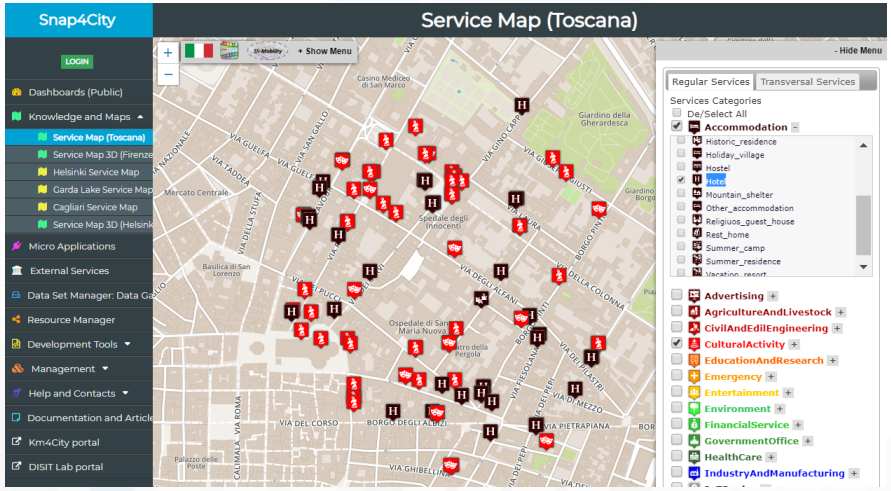
Pre-requirements:

- Knowledge Base main categories (Service Map or)

Register Static Data on Snap4City KB:

1. ‘Regularize’ your data putting them in csv file (possible fields available here: https://www.snap4city.org/589)
2. Create your **IoT App** starting from the sample available here:
   - Description: https://www.snap4city.org/596
   - Source code: https://www.snap4city.org/drupal/sites/default/files/files/iotapp-poi-ingestion.txt
3. Upload your csv containing N rows, each row is a Point of Interest

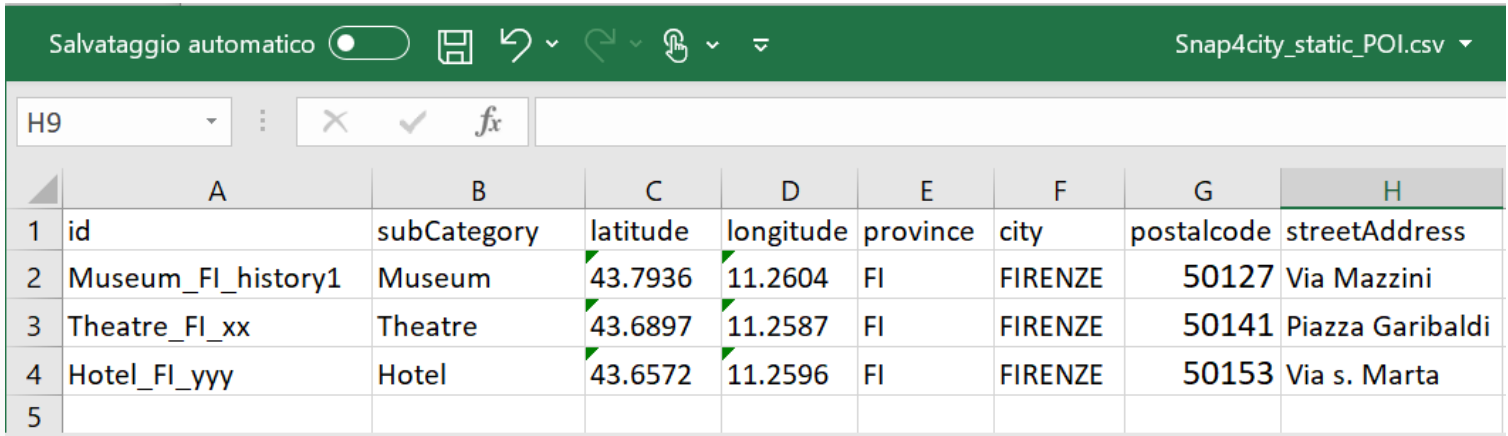
Snap4City Home > Knowledge and Maps > Services Menu (right)



Static data

Regularization

1. Create your csv (with 33 fields in the header). Here after some of them (M = to be filled Mandatory):
   - id: Unique POI identifier (**R**ecommended, if not present is created starting form the file name)
   - subCategory (**M**): KM4City category (ex: "Museum")
   - latitude (**M**): POI latitude (ex: "43.7936")
   - longitude (**M**): POI longitude (ex: "11.2604")
   - province: (ex.: "FI")
   - nameITA/nameENG: POI name
   - city: (ex.: "FIRENZE")
   - postalcode: (ex.: "50100")
   - streetAddress: (ex.: "via di Santa Marta")
   - civicNumber: (ex.: "3")
   - descriptionShortENG : (ex: "Exhibition... ")
   - url: (ex: "https://www.snap4city.org")
   - … others…here: https://www.snap4city.org/589



	A	B	C	D	E	F	G	H
1	id	subCategory	latitude	longitude	province	city	postalcode	streetAddress
2	Museum_FI_history1	Museum	43.7936	11.2604	FI	FIRENZE	50127	Via Mazzini
3	Theatre_FI_xx	Theatre	43.6897	11.2587	FI	FIRENZE	50141	Piazza Garibaldi
4	Hotel_FI_yyy	Hotel	43.6572	11.2596	FI	FIRENZE	50153	Via s. Marta
5								

Snap4city_static_POI.csv

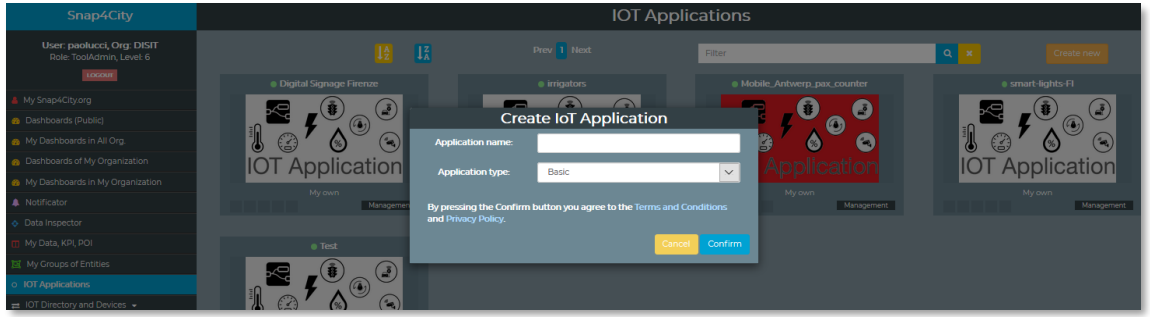
```
id;subCategory;latitude;longitude;province;city;postalcode;streetAddress;civicNumber;nameITA;nameENG;description;url;OTHERS…
Museum_FI_history1;Museum;43.7936;11.2604;FI;FIRENZE;50127;Via Mazzini;34;Museo storia contemporanea;Museum of Contemporary History;
Theatre_FI_xx;Theatre;43.6897;11.2587;FI;FIRENZE;50141;Piazza Garibaldi;4;Teatro Comunale;Municipal Theatre;Funded by …;https://www.
Hotel_FI_yyy;Hotel;43.6572;11.2596;FI;FIRENZE;50153;Via s. Marta;13;Hotel Nuovo;New Hotel;offerte …;https://www.hotel_XXX.com;
```

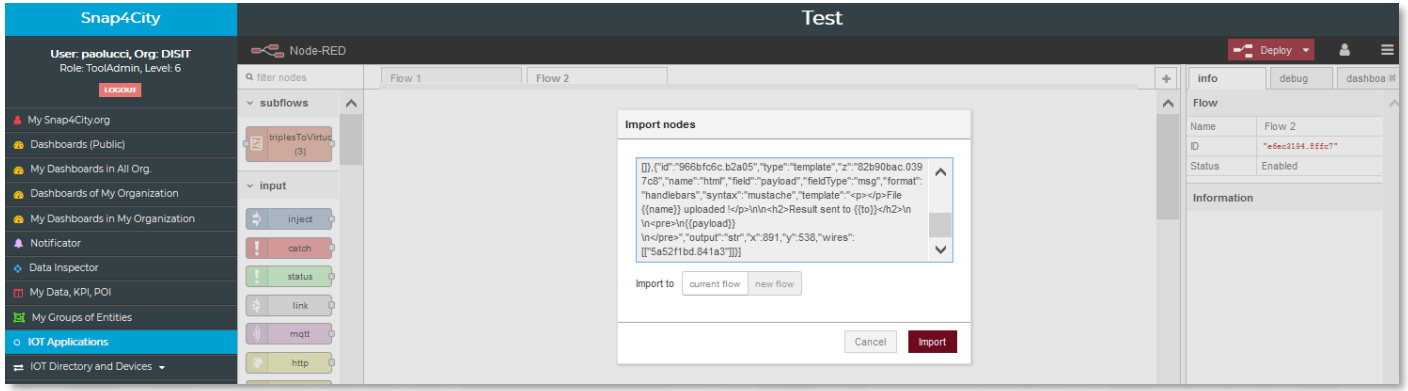
Static data

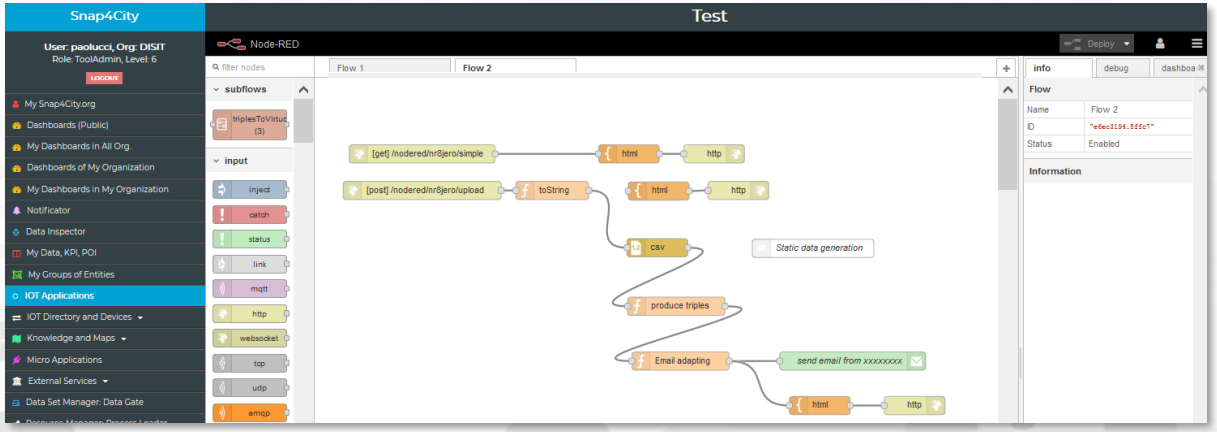
IoTApp

Create your IoTApp:

- Login > IoT Application > 'Create New' button
- Wait some minutes
- Copy the Source code available here: https://www.snap4city.org/drupal/sites/default/files/files/iotapp-poi-ingestion.txt
- Import the code and save ('Deploy' button)





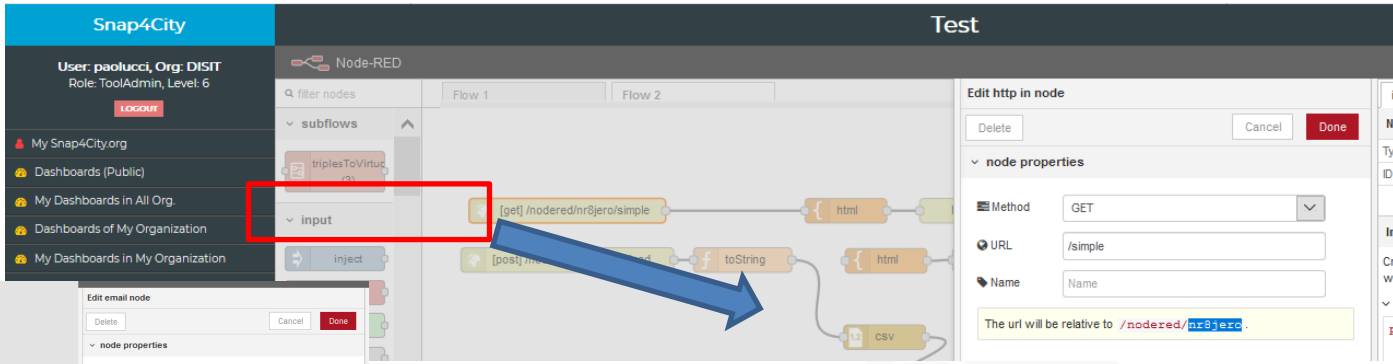


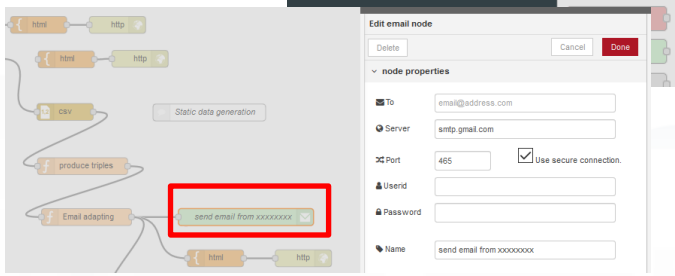
Static data

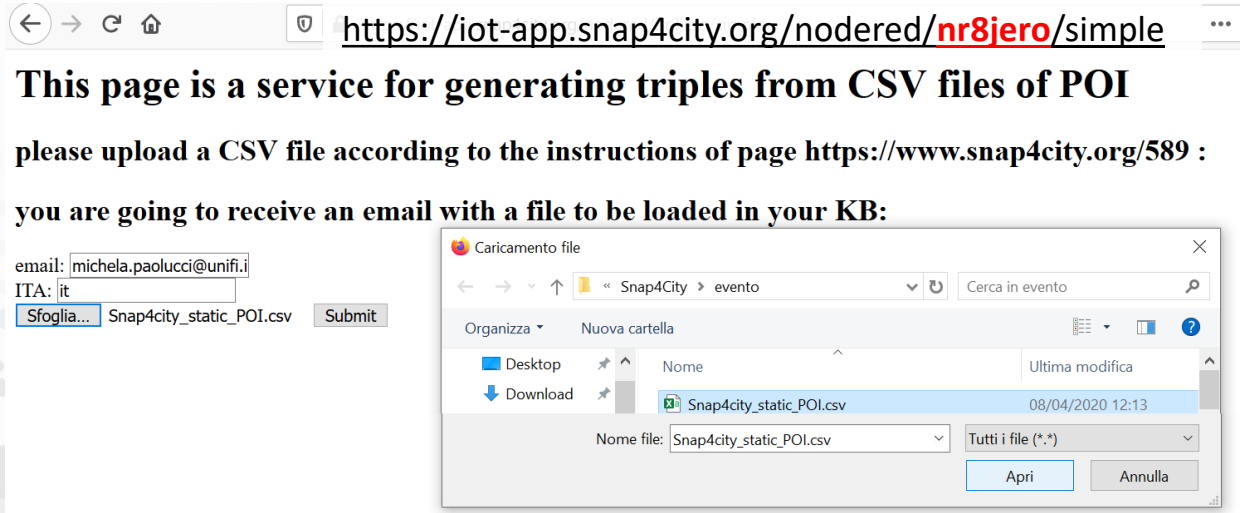
Upload your csv

Modify your IoTApp:

- Open the ‘get simple block’ and serach for your App id (ex: ‘**nr8jero**’)
- Put your mail and credentials on the ‘send email form ..’ block
- Go to the page ‘ https://iot-app.snap4city.org/nodered/**nr8jero**/simple ’ and upload you csv file
- Add your e-mail credential





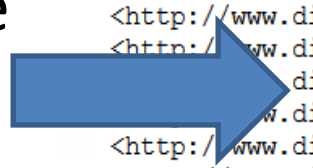


IoTApp (2)

Modify your IoTApp:

- Send the file you receive to us, or copy in a file the triples available in the web page and send them us
- Reference people:

  snap4city@disit.org



https://iot-app.snap4city.org/nodered/nr8jero/upload

File Snap4city_static_POI.csv uploaded !

Result sent to michela.paolucci@unifi.it

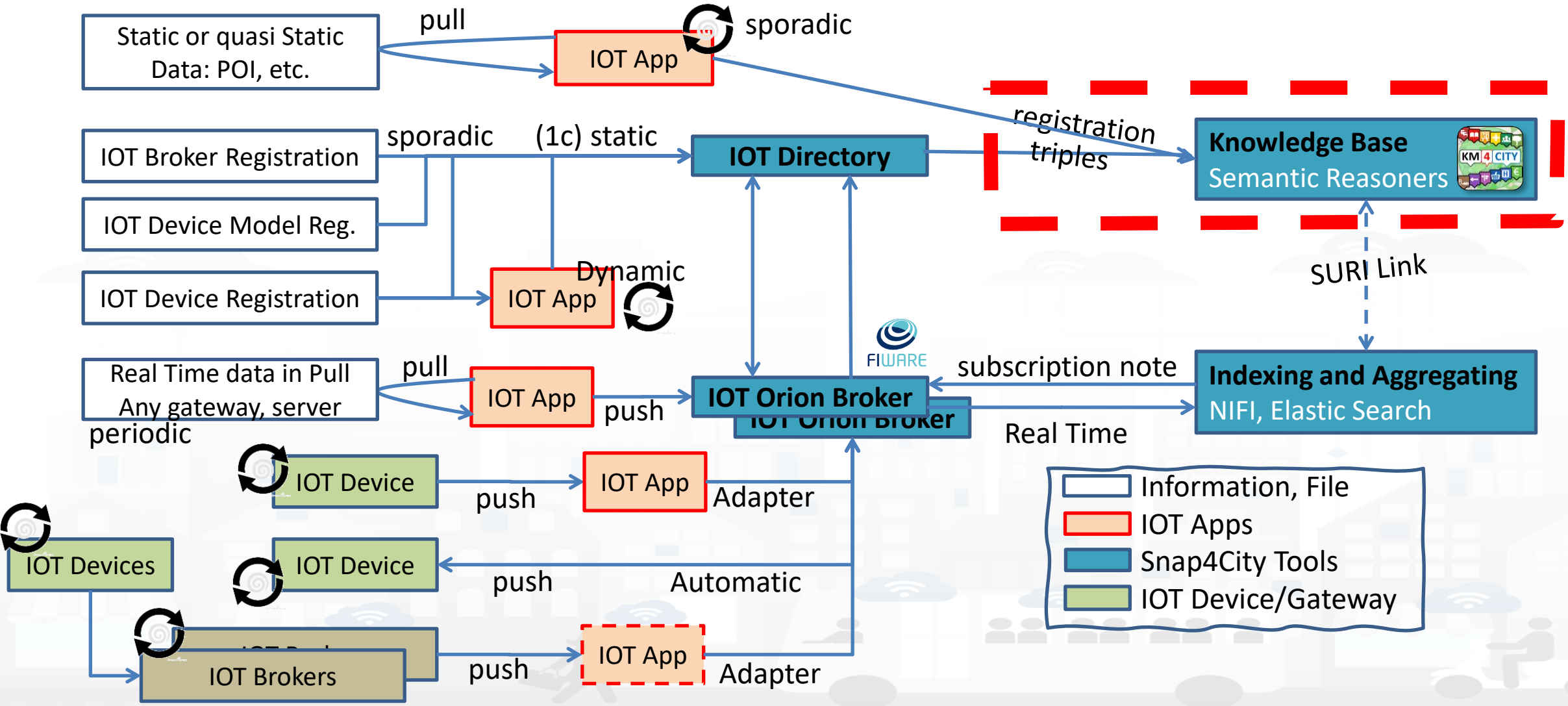
```
<http://www.disit.org/km4city/resource/Museum_FI_history1> a <http://www.disit.org/km4city/schema#Museum> .
<http://www.disit.org/km4city/resource/Museum_FI_history1> <http://purl.org/dc/terms/identifier> "Museum_FI_histor
<http://www.disit.org/km4city/resource/Museum_FI_history1> <http://schema.org/name> "Museo storia contemporanea"@i
<http://www.disit.org/km4city/resource/Museum_FI_history1> <http://schema.org/name> "Museum of Contemporary Histor
<http://www.disit.org/km4city/resource/Museum_FI_history1> <http://www.w3.org/2003/01/geo/wgs84_pos#lat> "43.7936"
<http://www.disit.org/km4city/resource/Museum_FI_history1> <http://www.w3.org/2003/01/geo/wgs84_pos#long> "11.2604
<http://www.disit.org/km4city/resource/Museum_FI_history1> <http://www.w3.org/2003/01/geo/wgs84_pos#geometry> "POI
<http://www.disit.org/km4city/resource/Museum_FI_history1> <http://www.disit.org/km4city/schema#shortDescription>
<http://www.disit.org/km4city/resource/Museum_FI_history1> <http://schema.org/url> "https://www.museo_XXX.com" .
<http://www.disit.org/km4city/resource/Museum_FI_history1> <http://schema.org/streetAddress> "Via Mazzini" .
<http://www.disit.org/km4city/resource/Museum_FI_history1> <http://www.disit.org/km4city/schema#houseNumber> "34"
<http://www.disit.org/km4city/resource/Museum_FI_history1> <http://schema.org/postalCode> "50127" .
<http://www.disit.org/km4city/resource/Museum_FI_history1> <http://schema.org/addressLocality> "FIRENZE" .
<http://www.disit.org/km4city/resource/Museum_FI_history1> <http://schema.org/addressRegion> "FI" .

<http://www.disit.org/km4city/resource/Theatre_FI_xx> a <http://www.disit.org/km4city/schema#Theatre> .
<http://www.disit.org/km4city/resource/Theatre_FI_xx> <http://purl.org/dc/terms/identifier> "Theatre_FI_xx" .
<http://www.disit.org/km4city/resource/Theatre_FI_xx> <http://schema.org/name> "Teatro Comunale"@it .
<http://www.disit.org/km4city/resource/Theatre_FI_xx> <http://schema.org/name> "Municipal Theatre"@en .
<http://www.disit.org/km4city/resource/Theatre_FI_xx> <http://www.w3.org/2003/01/geo/wgs84_pos#lat> "43.6897"^^<ht
<http://www.disit.org/km4city/resource/Theatre_FI_xx> <http://www.w3.org/2003/01/geo/wgs84_pos#long> "11.2587"^^<h
<http://www.disit.org/km4city/resource/Theatre_FI_xx> <http://www.w3.org/2003/01/geo/wgs84_pos#geometry> "POINT(11
<http://www.disit.org/km4city/resource/Theatre_FI_xx> <http://www.disit.org/km4city/schema#shortDescription> "Fund
<http://www.disit.org/km4city/resource/Theatre_FI_xx> <http://schema.org/url> "https://www.teatro_XXX.com" .
<http://www.disit.org/km4city/resource/Theatre_FI_xx> <http://schema.org/streetAddress> "Piazza Garibaldi" .
<http://www.disit.org/km4city/resource/Theatre_FI_xx> <http://www.disit.org/km4city/schema#houseNumber> "4" .
<http://www.disit.org/km4city/resource/Theatre_FI_xx> <http://schema.org/postalCode> "50141" .
<http://www.disit.org/km4city/resource/Theatre_FI_xx> <http://schema.org/addressLocality> "FIRENZE" .
<http://www.disit.org/km4city/resource/Theatre_FI_xx> <http://schema.org/addressRegion> "FI" .
```

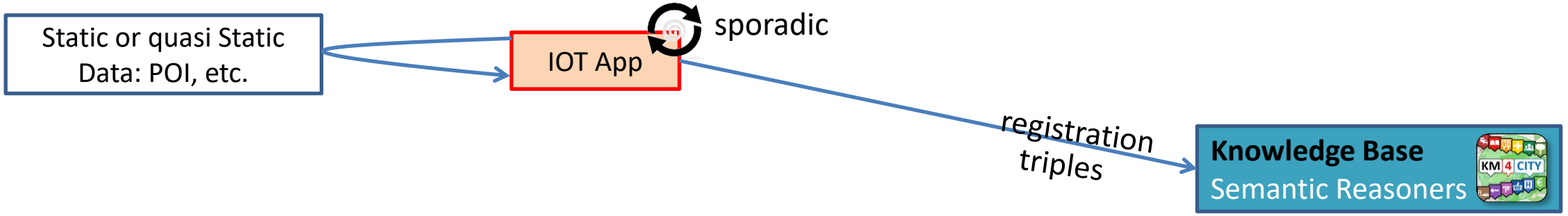
How to load triples into the KB
(Admin and on-premise)



Snap4city Data Ingestion Flow Diagram



Loading Static Data into KB



- The loading is performed via IOT App, with a specific IOT App flow exploiting a MicroService/API of ServiceMap, KB
- Moreover for massing loading of Triples, also files of triples can be used posting them on a shared folder of the ServiceMap, KB
  - This is used for batch loading of triples

Load Triples in KB

Pre-requirements:

- Available only for your IOT App on premise and for administrators

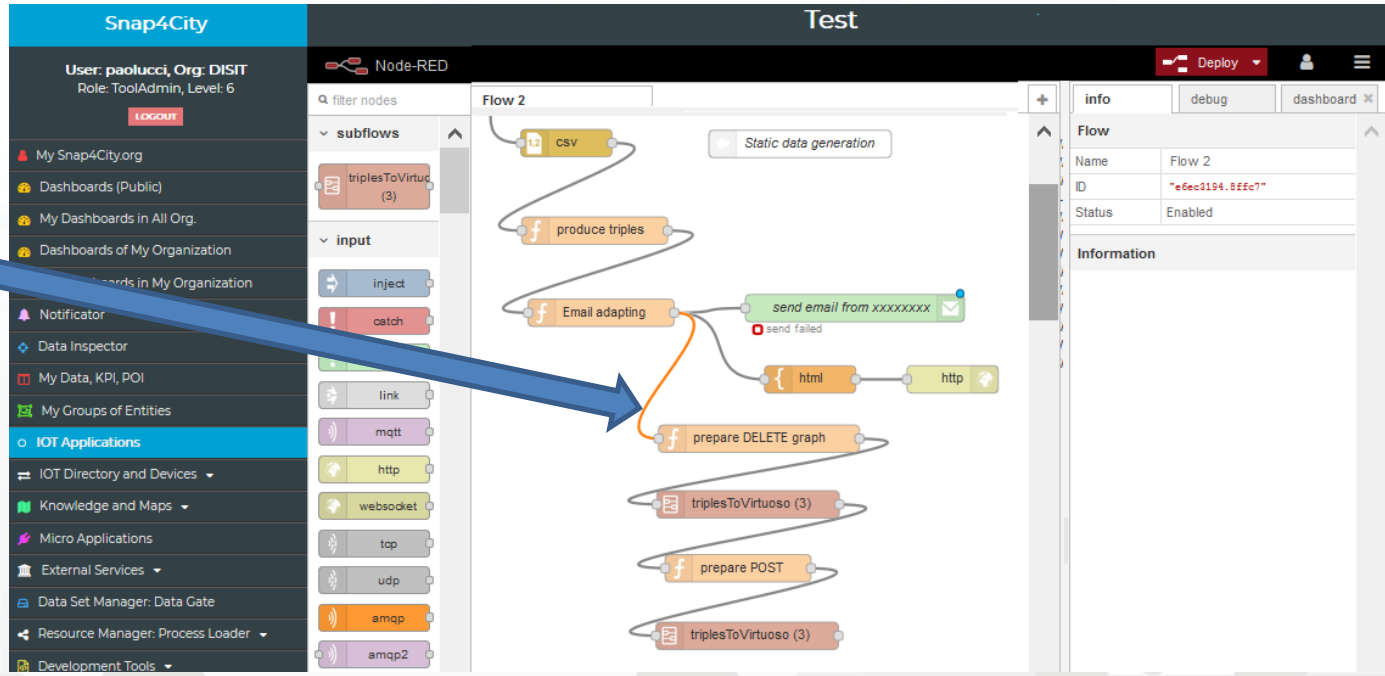
Load Static Data on Snap4City KB:

- Register Static Data on Snap4City KB (point 1 and 2 on the right)
- Connect the additional blocks present in the IOT App and save (‘Deploy’ button)
- Upload your csv

https://www.snap4city.org/596

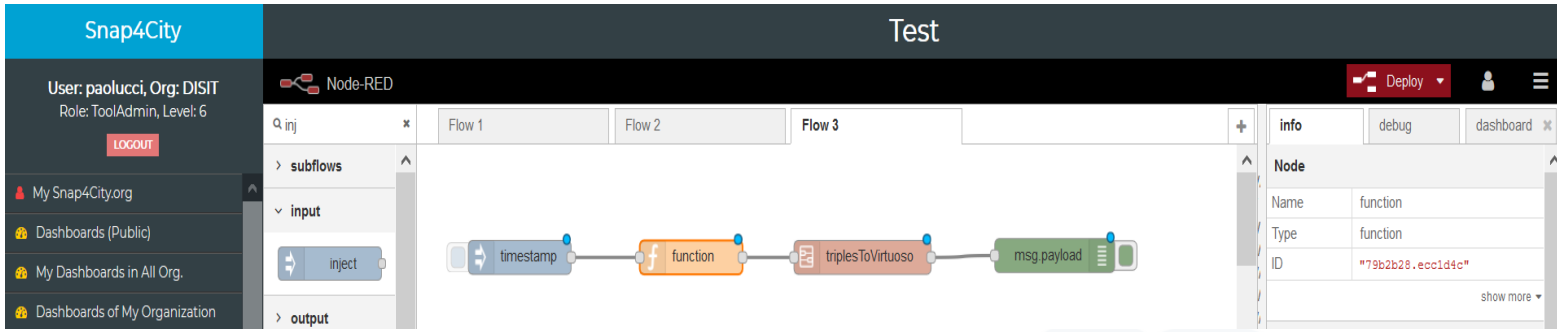
Register Static Data on Snap4City KB:

1. ‘Regularize’ your data (csv)
2. Create your IoTApp
3. Upload your csv



Load Triples in KB

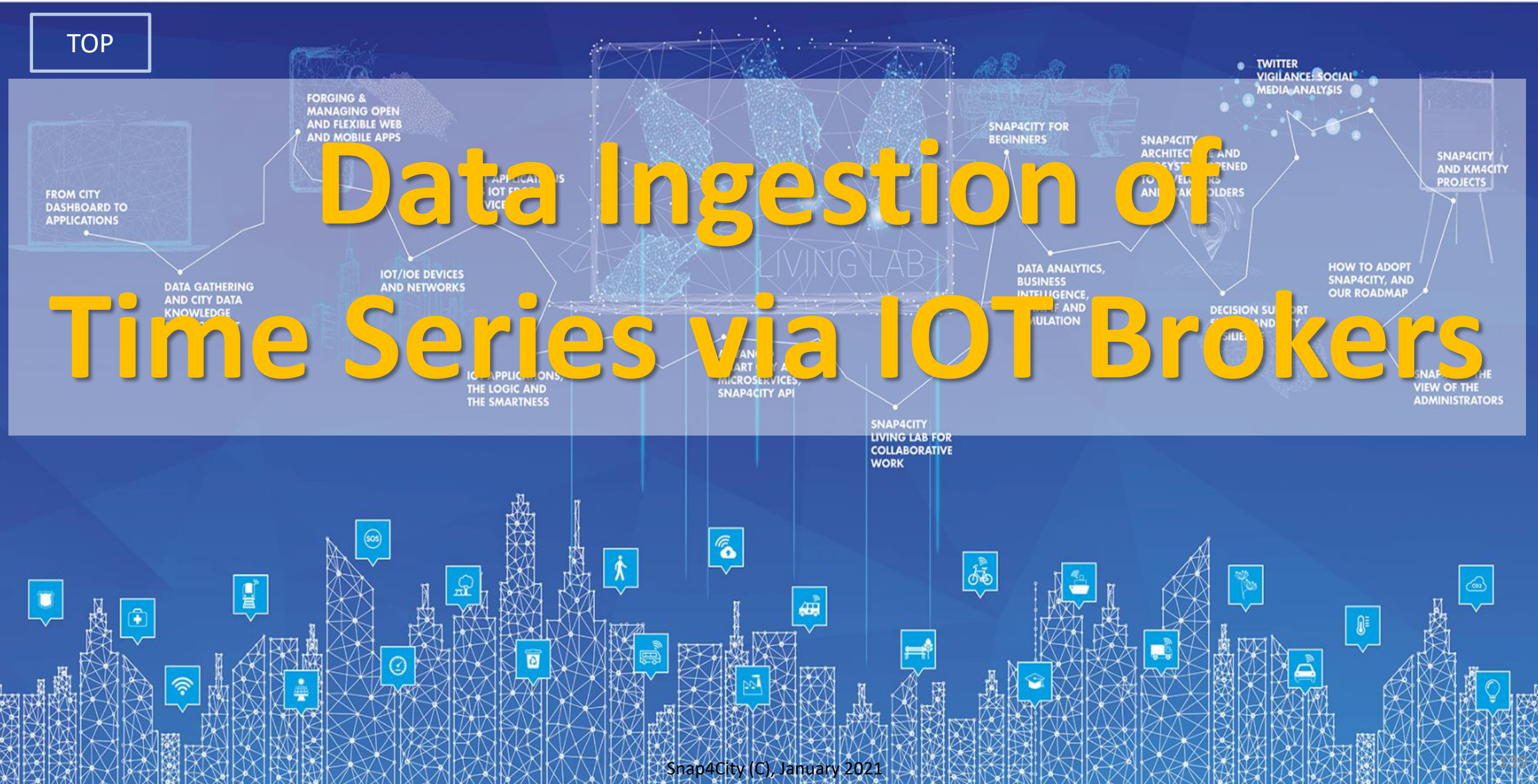
- Additional Option Only for ONPREMISE Snap4City versions and for administrators
- In case you need to upload ONLY on the Knowledge Base a set of data containing static data and dynamic data all together:
  - Example: Cultural Events or weather predictions
- Use a predefined IoT App available here https://www.snap4city.org/594 (with the copy and past method, you can create your Mobile App)
- Add:
  - Inject block
  - Function block
  - Debug block
- The function block must contain …
- Now you can see the triples to be updated in the Knowledge Base in the Debug Tab



The input message to be provided to *triplesToVirtuoso* has to have the following properties:

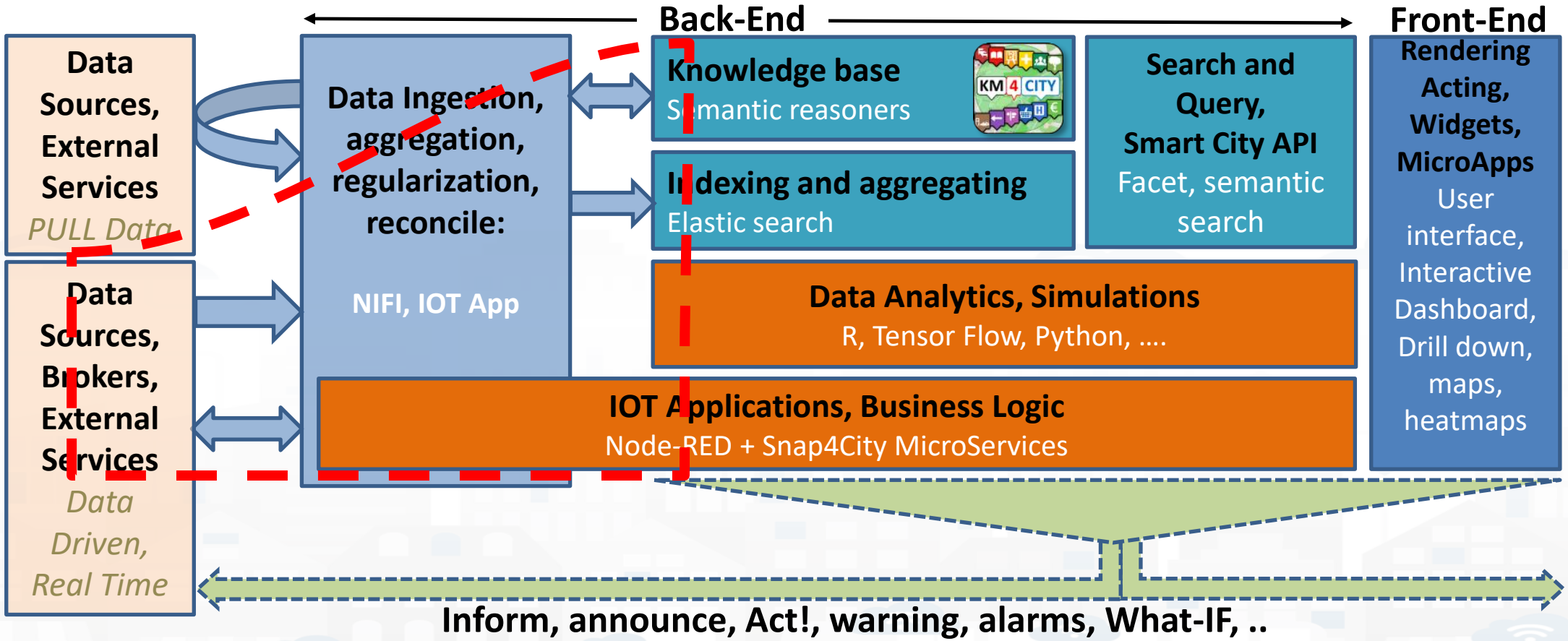
msg.user="dba" // the user of virtuoso to be used to access to virtuoso
msg.passw="dba" //the password
msg.method="POST" //the http method: GET, POST, DELETE, PUT
msg.url="http://virtuoso-kb:8890/sparql-graph-crud-auth?graph-uri=urn:graph:POIs"
msg.triples= "..."

TOP

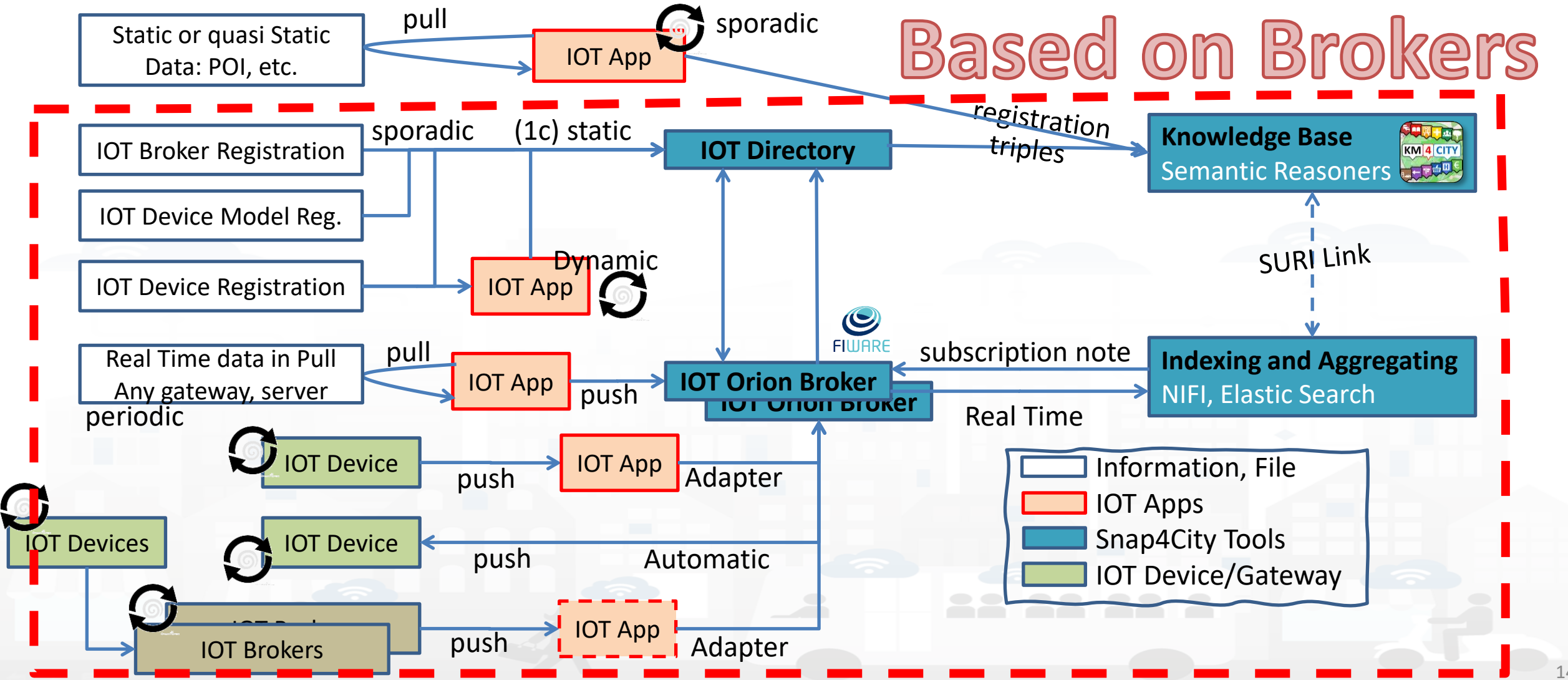


Data Ingestion of Time Series via IOT Brokers

Snap4City Architecture vs Data Ingestion

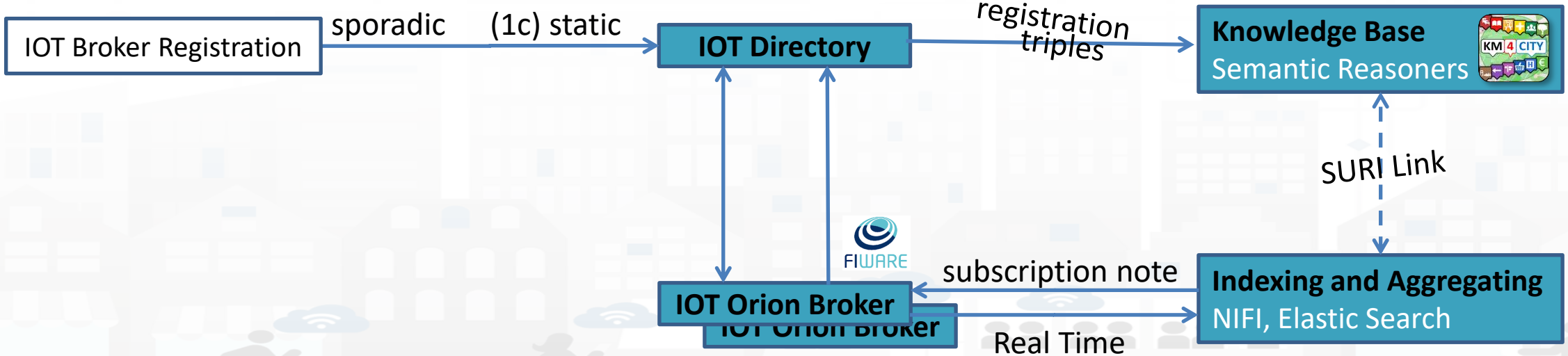


Snap4city Data Ingestion Flow Diagram



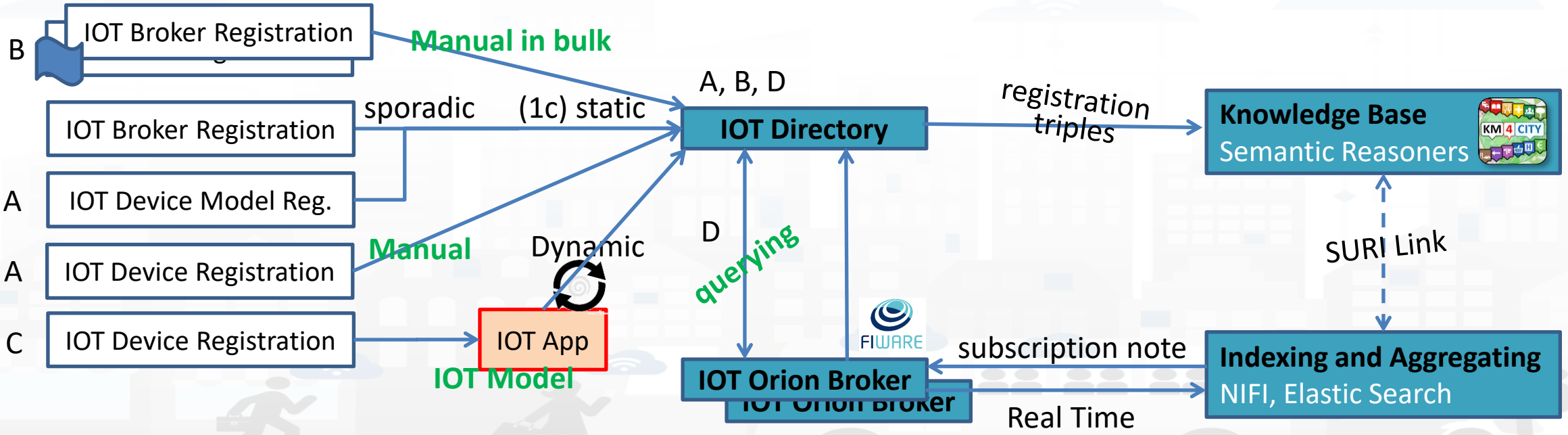
The process

- **IOT Network concepts**, See Course 2020 Part 3
  - IOT Devices, IOT vs Dashboards, etc.
- **IOT Broker registration** if not performed, See Course 2020 Part 3
  - IOT Brokers can be Internal or External (managed by third party)



The process

- **IOT Device registration** on IOT Directory: **concept**
  - the IOT Directory performs the registration on KB and NIFI for automated Data Shadow storage
- **HOW TO: register IOT Device(s)** on IOT Directory:
  A. Single IOT Device **Manually from Zero** or by **IOT Device Model** (via user interface IOT Dir)
  B. in **Bulk** by uploading a File with IOT Devices via the IOT Directory user interface + CSV file
  C. Automatically via an **IOT App using special Block/MicroService having** an IOT Device Model
  D. Automatically by activating a **querying the IOT Broker** from the IOT Directory **(for External)**



IOT Directory



IOT Network Manager vs Final User

Network of IOT Brokers

IOT Broker

External

Internal

IOT Directory

IOT Network Manager

My IOT Device

Register

Registering

Discovering

Knowledge Base, Km4City

Knowledge and Storage Data from the Field and City

Browsing

ServiceMap Knowledge Base

IOT Application

Final user Manager

Discovering

Prepare

Deep Manage

Exploit

Dashboard Wizard

Main Features of the Snap4City IOT Directory:

- **IOT Directory is a technology of Snap4City ONLY**
- **Register IOT Brokers**
  - Different kind of Brokers, different kinds of authentications and protocols
  - Registered IOT Orion Brokers can be queried for collecting their managed devices (typically for External IOT Brokers), so that those IOT Devices are registered
  - IOT Brokers/Gateways are registered on NIFI to send messages into the Data Shadow, automatically
- **Register IOT Devices**: singularly or at groups (in Bulk)
  - Registration can be custom or based on IOT Device Model
  - IOT Edge are registered as IOT Devices as well
  - Registered IOT Devices are saved into local DataBase and into the Knowledge Base
- **Provide support for security aspects**:
  - Generation of Certificates, Keys, etc., according to the model
  - Collection of keys when IOT devices are on some IOT Gateway or Second Level IOT Broker.
- **Manage Ownership and Delegation** for
  - IOT Brokers, IOT Devices, IOT Device Values also called Sensors/actuators, IOT Device Models

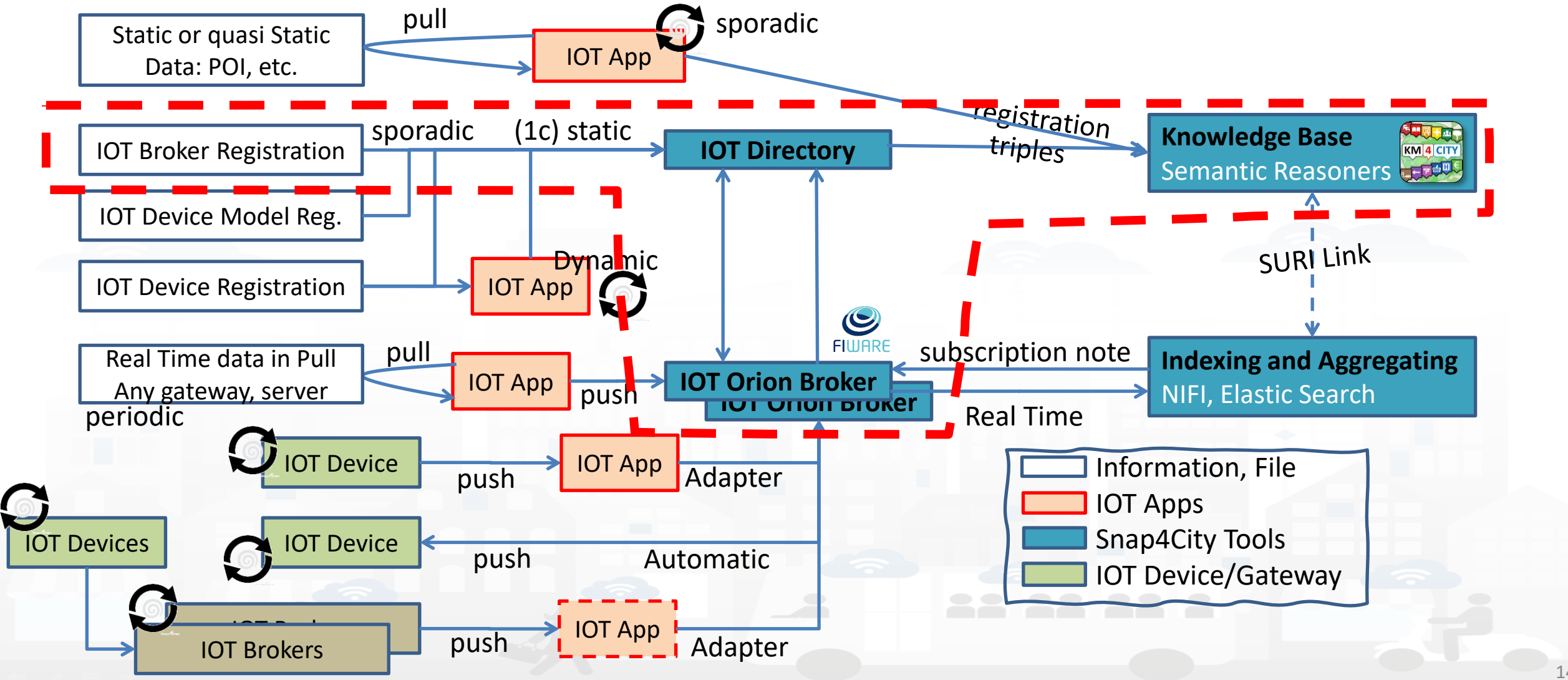
IOT Directory Features vs Users Roles

Entities	what	By using IOT Directory and:	Manager	AreaManager	ToolAdmin/ RootAdmin	IOT App microservices
IOT Sensor/Actuator	Browse, use	Several Tools	X	X	X	Yes
	Delegate	API, ..	X	X	X	
	Discovery	KB, API, ..	X	X	X	Yes
IOT Devices	Browse, use	Several Tools	X	X	X	Yes (use)
	Create, change, delete	API, ..	X	X	X	
	Register in Bulk	API, ..		X	X	
	Delegate, Change Owner	API, ..	X	X	X	Yes
	Discovery	KB, API, ..	X	X	X	Yes
IOT Device Model	Browse, Use		X	X	X	
	Create, change, delete			X	X	
	delegate, change ownership			X	X	
IOT Broker	Browse, use		use	Browse, use	X	Yes (use)
	Register/change/Delete				X	
	Delegate				X	
	Periodic Update				X	

IOT Broker Registration



Snap4city Data Ingestion Flow Diagram



IOT Brokers

Snap4City

User: roottooladmin1, Org: DISIT
Role: RootAdmin, Level: 7
LOGOUT

- My Snap4City.org
- Dashboards
- My Dashboards in All Org.
- Dashboards of My Organization
- My Dashboards in My Organization
- Notificator
- Data Inspector
- My Data, KPI, POI
- My Groups of Entities
- IOT Applications
- IOT Directory and Devices
  - My IOT Devices
  - IOT Sensors and Actuators
  - IOT Devices
  - IOT Devices Management
  - IOT Brokers
  - IOT Device Models
  - IOT Devices Bulk Registration
  - IOT Broker Periodic Update setting
  - IOT Orion Broker Mapping Rules
- Knowledge and Maps
- Micro Applications
- External Services
- Data Set Manager: Data Gate
- Synoptics
- Resource Manager: Process Loader
- Development Tools
- Management

IOT Brokers

New IOT Broker

Show 10 entries

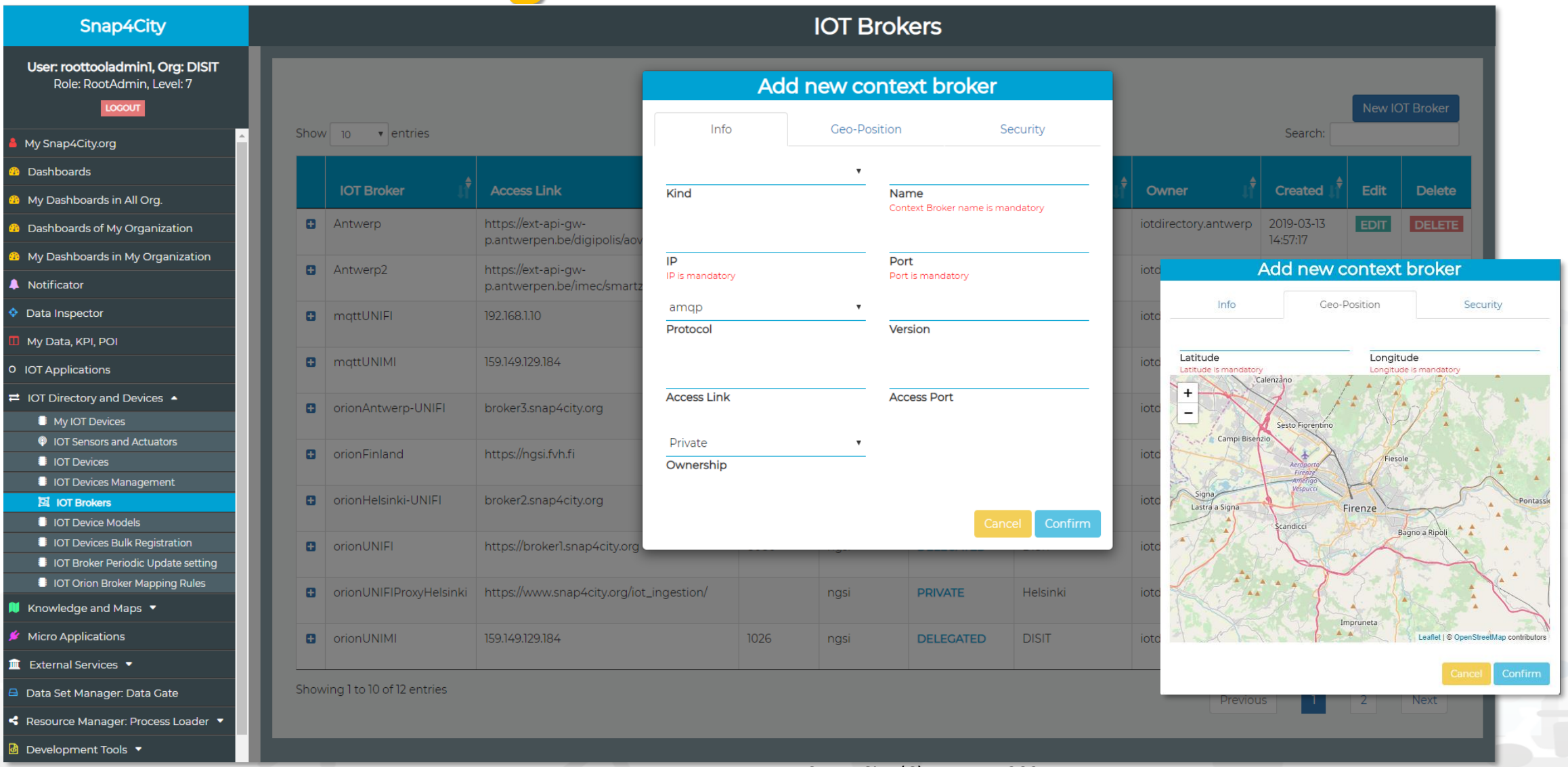
Search:

	IOT Broker	Access Link	Access Port	Protocol	Ownership	Organization	Owner	Created	Edit	Delete
+	Antwerp	https://ext-api-gw-p.antwerpen.be/digipolis/aovmma/v1/entities		ngsi	DELEGATED	Antwerp	iotdirectory.antwerp	2019-03-13 14:57:17	EDIT	DELETE
+	Antwerp2	https://ext-api-gw-p.antwerpen.be/imec/smartzone/v1		ngsi	DELEGATED	Antwerp	iotdirectory.antwerp	2019-01-01 00:00:00	EDIT	DELETE
+	iotobsf-smartbed	192.168.1.47	8443	ngsi	DELEGATED	SmartBed	angelo.difino	2019-11-29 15:31:51	EDIT	DELETE
+	mqttUNIFI	192.168.1.10	1883	mqtt	DELEGATED	DISIT	iotdirectory.disit	2018-02-07 15:14:39	EDIT	DELETE
+	mqttUNIMI	159.149.129.184	1884	mqtt	DELEGATED	DISIT	iotdirectory.disit	2018-04-30 16:49:05	EDIT	DELETE
−	orionAntwerp-UNIFI	broker3.snap4city.org	8080	ngsi	PUBLIC	Antwerp	iotdirectory.antwerp	2019-06-03 14:25:16	EDIT	DELETE
	IP: 192.168.1.12					Port: 1026				
	Latitude: 43.76666					Longitude: 11.26242				
	Login:					Password:				
	SHA:									
+	orionFinland	https://ngsi.fvh.fi		ngsi	PUBLIC	Helsinki	iotdirectory.helsinki	2018-11-21 16:05:24	EDIT	DELETE
+	orionFirenze-UNIFI	192.168.1.17	8443	ngsi	PRIVATE	Firenze	iotdirectory.firenze	2019-10-28 10:01:53	EDIT	DELETE
+	orionHelsinki-UNIFI	broker2.snap4city.org	8080	ngsi	PUBLIC	Helsinki	iotdirectory.helsinki	2019-06-03 10:26:34	EDIT	DELETE
+	orionLonatoDelGarda-UNIFI	192.168.1.46	8443	ngsi	DELEGATED	LonatoDelGarda	iotdirectory.ldg	2019-10-28 10:04:05	EDIT	DELETE

Showing 1 to 10 of 17 entries

Previous 1 2 Next

Register a New IOT Broker



Snap4 technology is broker Agnostic

- **IOT Brokers**
  - are associated with an Organization
    - Each **Organization** has a **Knowledge Base** of reference (KB, ServiceMap)
    - Each **KB** may host multiple **Organizations** and addresses multiple **Geographic areas**
  - can be compliant with
    - **NGSI version**: V1, V2-1, V2, etc…
      - with Snap4City Security or regular NGSI FiWare
    - other protocols as well such as: MQTT, COAP, AMQP, etc.
  - can
    - expose different **authentication methods**: K1/K2, Certificate, etc.
    - be accessible from IOT Devices and IOT App **in Cloud only**
    - be accessible **from Internet to post data from outside, etc.**

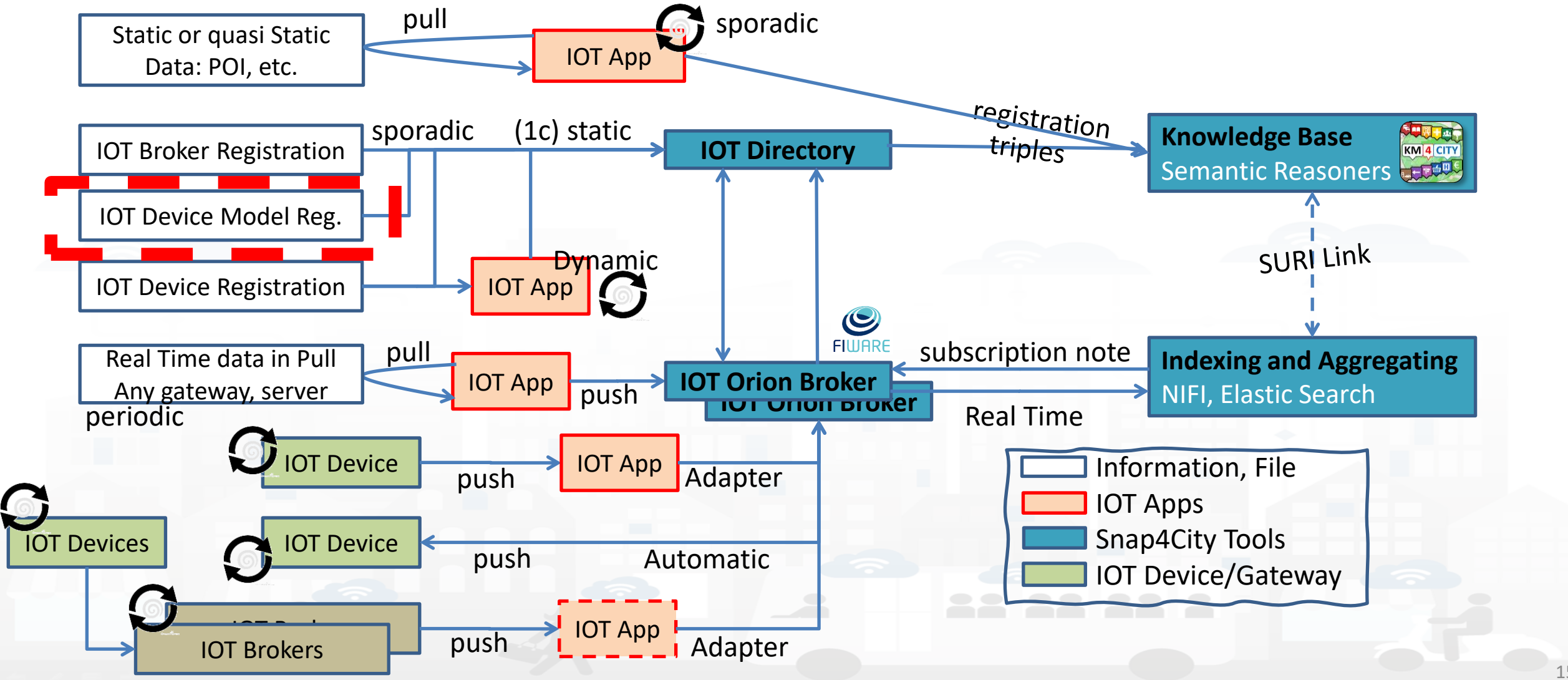
IOT Orion Broker Network: V1 and V2

- **IOT Broker can be Internal (on Snap4City Cloud)**
  - Registration of IOT Devices can be performed by the IOT Directory
  - Authentication is automatic, K1 and K2 are not needed since the security is performed via Access Token, M2M secure communication, on the basis of IOT App ownership
  - The **NIFI Cluster** automatically subscribes to all the entities on the Broker, to post data into the Data Shadow enriched with data of the KB
- **IOT Broker can be External (managed by third party)**
  - Registration of IOT Devices is managed by third parties
  - The registered IOT Devices can be collected and queried from the IOT Directory as well
  - The NIFI Cluster **may** automatically subscribes to all the entities on the Broker, to post data into the Data Shadow enriched with data of the KB
- **IOT Brokers can be networked**
  - Services, Service paths: for managing the IOT Broker network
  - Multi-tenant: more than one user/org on the same IOT Broker

IOT Device Model

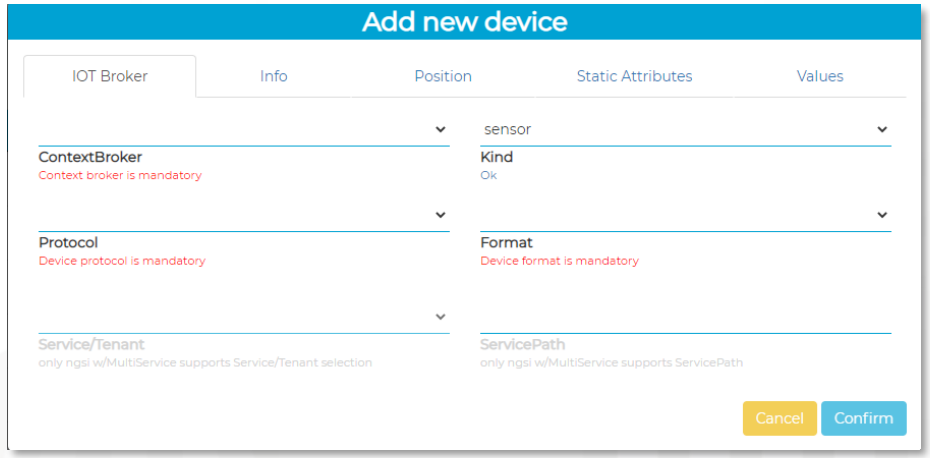


Snap4city Data Ingestion Flow Diagram

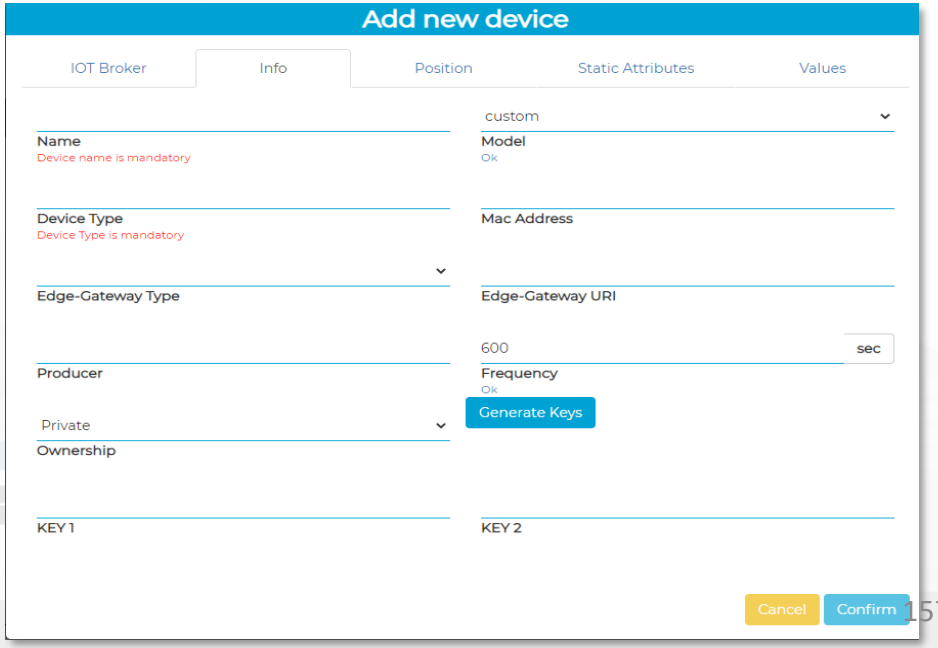


IOT Device Data Model (1)

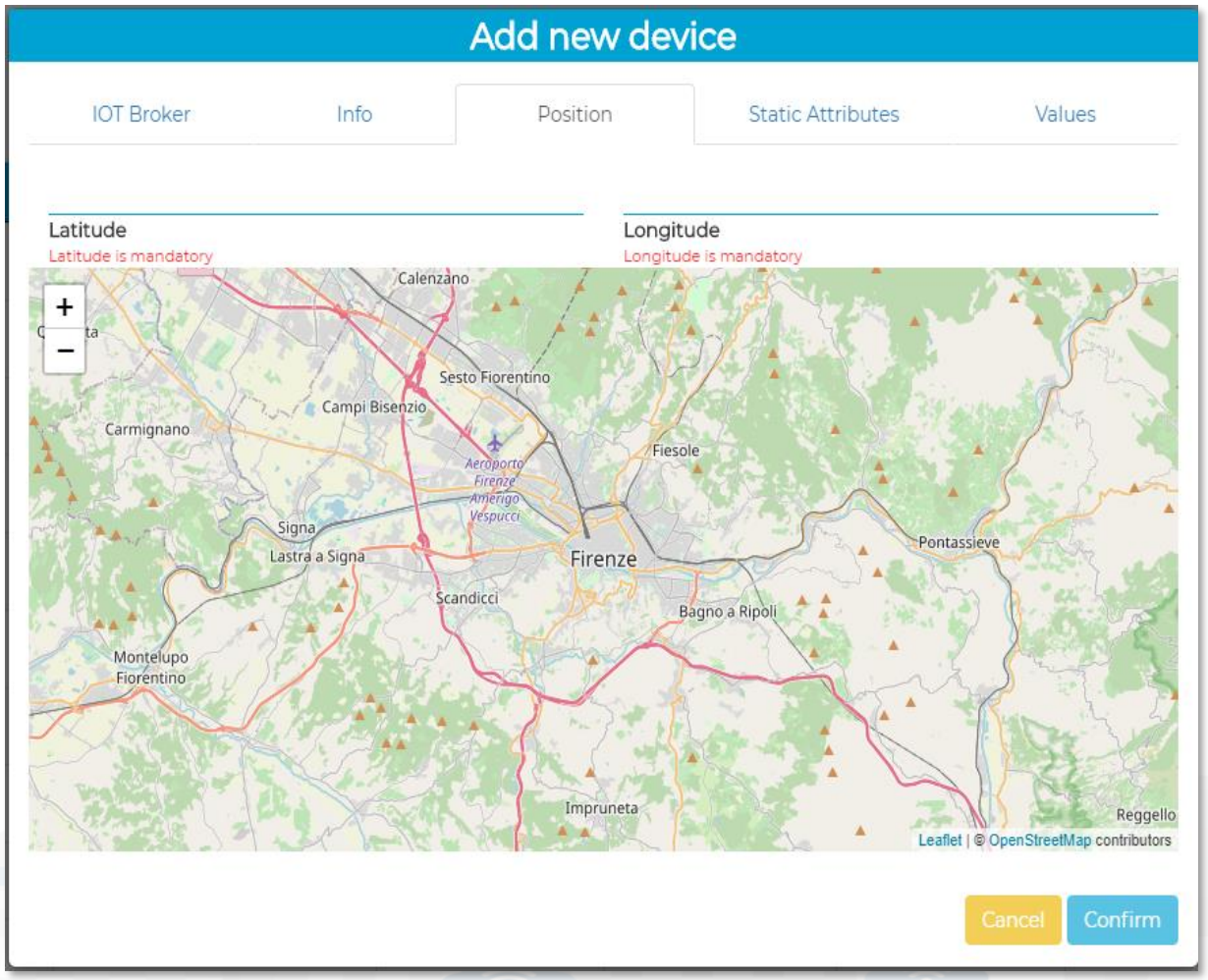
- IOT Broker
  - Name of the Brokers: among those registered
  - Protocol: NGSI, AMQP, MQTT, etc..
  - Format: CSV, JSON, XML.
  - Service/Tenant:…….
  - ServicePath:……

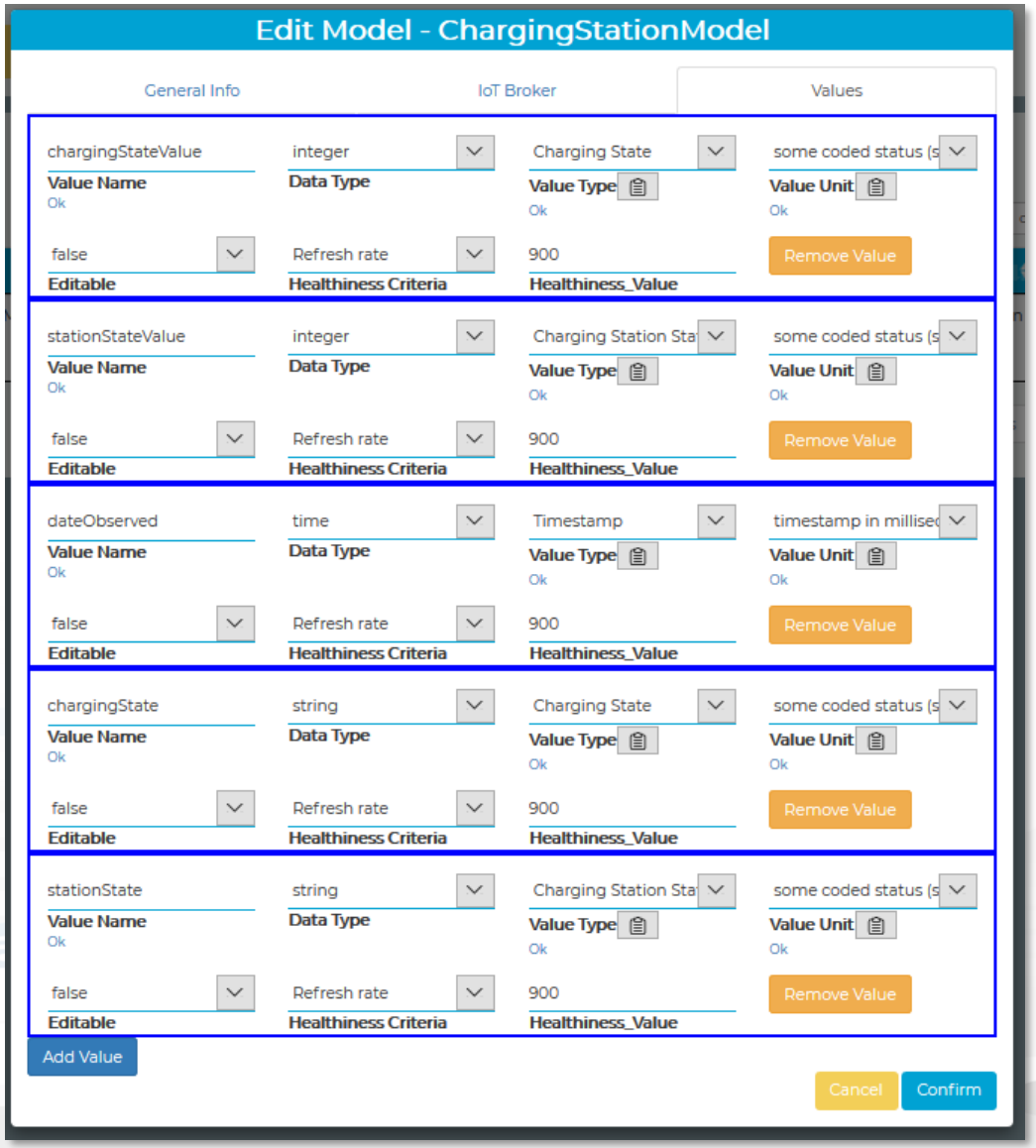


- Info
  - Name (Identifier)
  - Model: Custom or Model ID
  - DeviceType: ..a string..
  - MAC address: …optional…
  - Edge-GW: Raspberry, Android, …
  - Edge-GW: URI
  - Producer
  - Owner
  - Freq: ….. Sec
  - Keys: K1, K2



IOT Device Data Model (2)





IOT Device Data Model (3): Attributes

Where	IOT Device Model	IOT Device	A Temporal Instance
IOT Broker	**Broker: OrionUNIFI**		
IOT Broker	**Protocol: NGSI**		
Info	**ID: string**	ID: "park45"	**park45**
Position	**GPS: lat, long**	GSP Position: 43.12, 11.34	GSP Position: 44.12, 11.12
Static attribute	**Description: string**	Description: "parking massaia"	
Static attribute	**Location: string**	Location: "Via Massaia"	
Static attribute	**Civic Number: string**	Civic Number: 3	
Static attribute	**MaxCapacity: number, cars**	MaxCapacity: 456	
Values	**dateObserved: Timestamp**		23-12-2019T20:13:12…
Values	**FreeSlots: Integer, #**		345
Values	**Humidity: float, %**		25,5
Values	**Temperature: float, celsius**		34

Model meaning

- **ID:** is the unique identifier for reconnecting Temporal Instances with registered IOT Devices
- **Static Attributes:**
  - Are typically associated with instances of the IOT Device.
    E.g.:, You have a set of parking areas, each of them is located in a specific street, and has its one name, etc.
  - Different kinds of attributes can be set for each SubNature. Their definition has to be prepared into the Knowledge Base ☺ for automated indexing.
- **Values**: they are time varying variables (temporal values/instances)
  - They change over time, the time stamp of the time serie is conventionally «dateObserved» in Snap4City
  - In new *SensorMobile* HLT, also GPS can be changing over time as in the MyKPI

Using the IOT Device Model notes!!!

- Once performed the IOT Device Model, a number of IOT Devices can be produce **using the model as a Template**

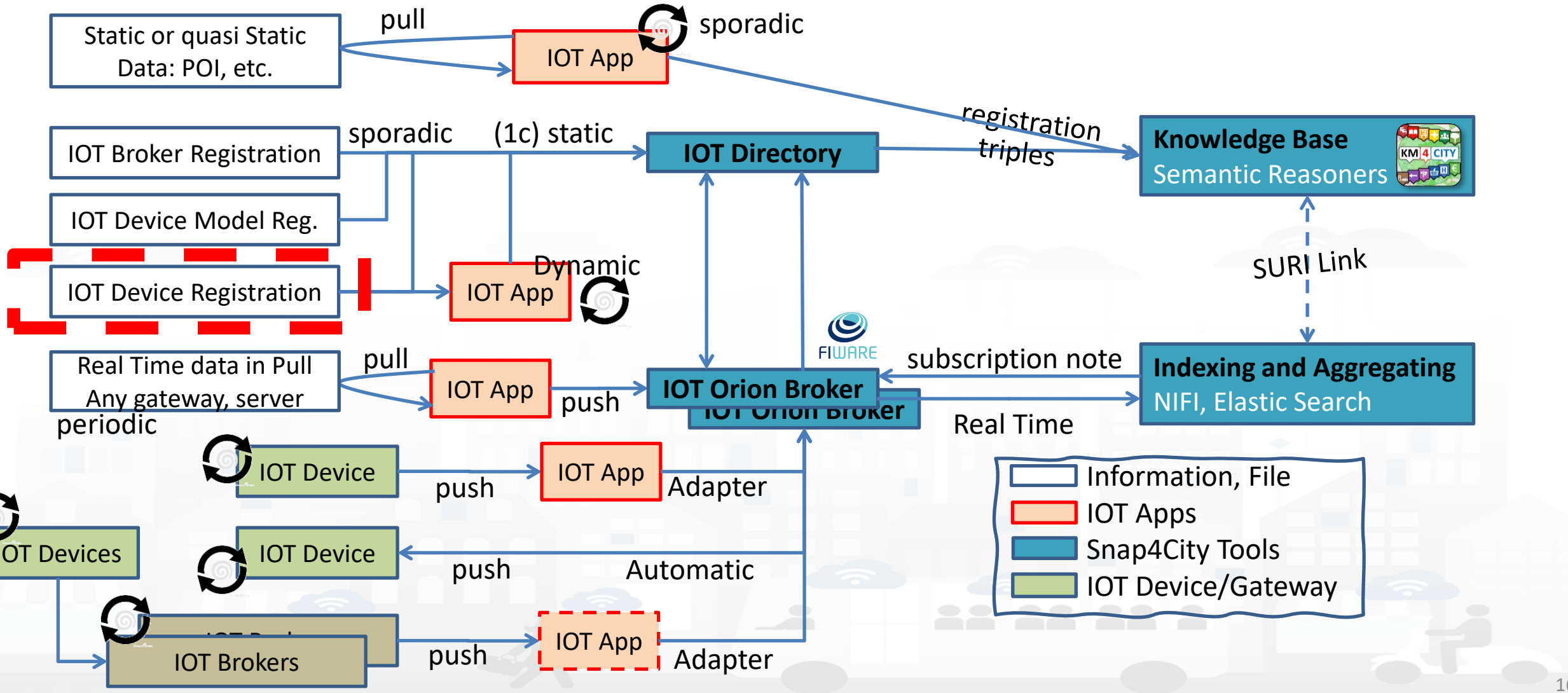
  - **NOTE:** the produced IOT Devices are not going to change if the IOT Device Model is modified.
  - *Your biscuit is not changing if the template is modified after the printout*



IOT Device Registration many possibilities



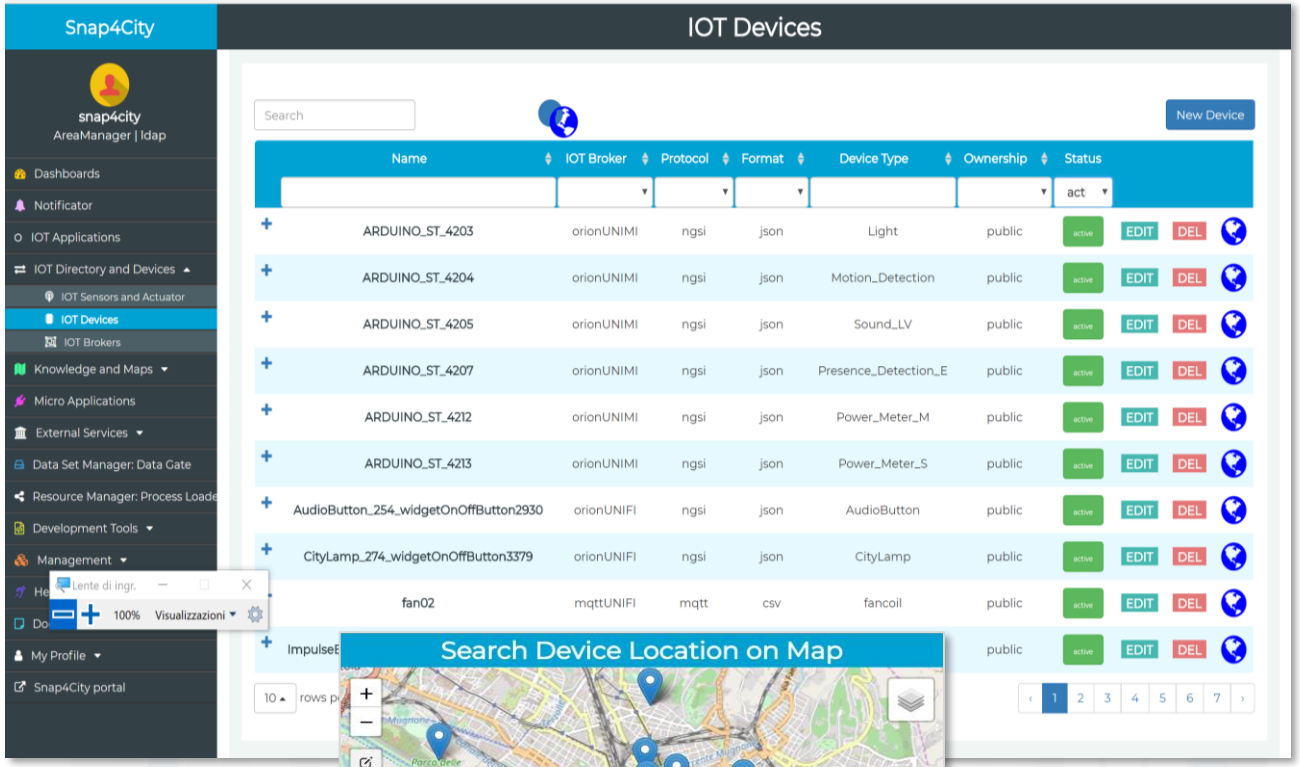
Snap4city Data Ingestion Flow Diagram

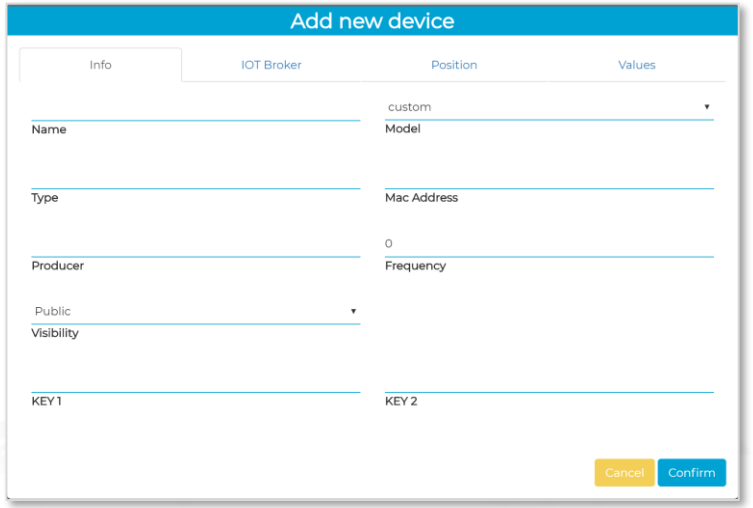


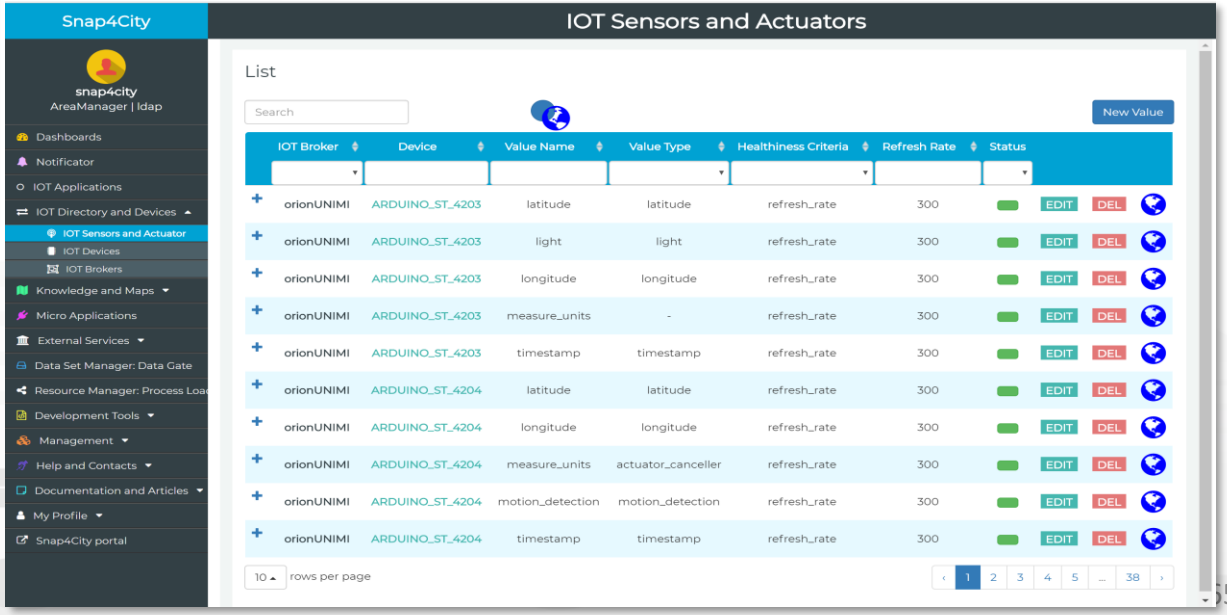
Activities for IOT Device Registration

- **Registration of**
  - an IOT Device Manually from Zero or from an IOT Device Model
  - a set of IOT Devices with the same IOT Device Model from IOT App
  - a set of IOT Devices in BULK loading a CSV (with or without a reference IOT Model)
- The IOT Device registration implies the **automated production of the Digital Twin Device into the Knowledge Base**, which implies:
  - Activation of the Storage “DataShadow” for historical data access
  - Activation of all the relationships
  - Activation of Discovery mechanisms via IOT Directory, KB
  - Activation of Dashboard Wizard (after a few minutes), and Data Inspector
  - Etc.

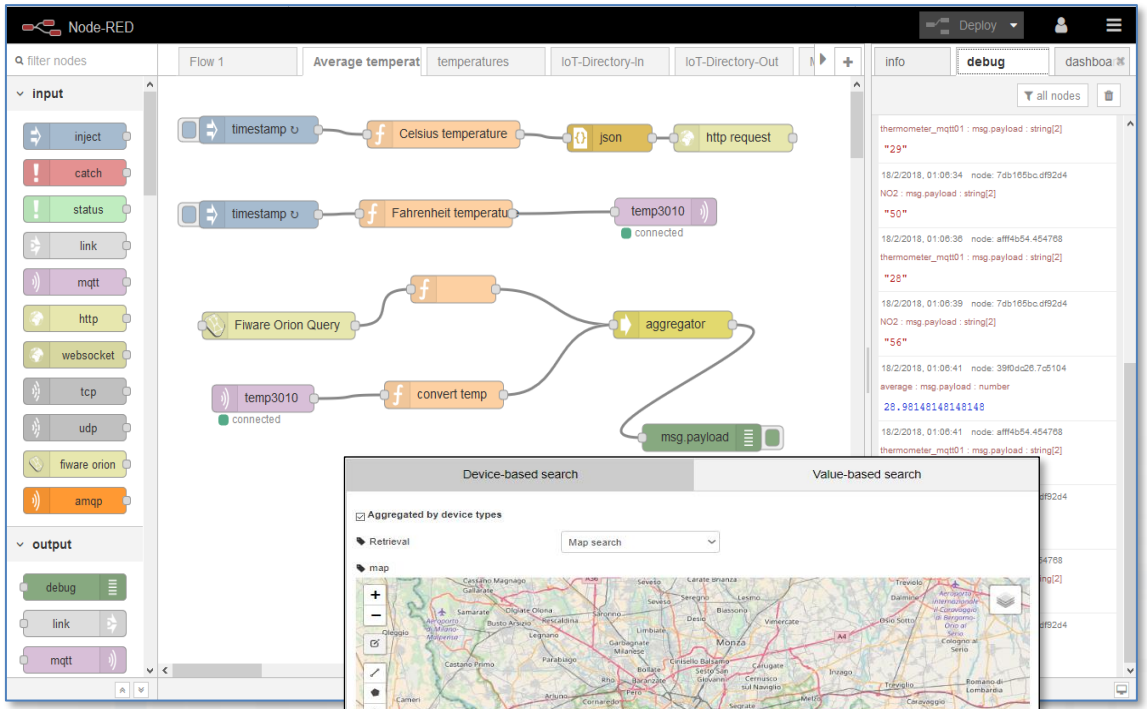
IOT Devices/Sensors

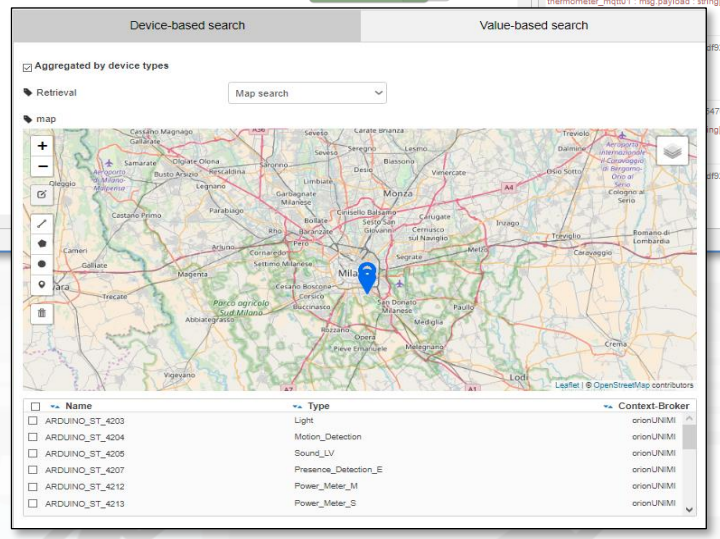


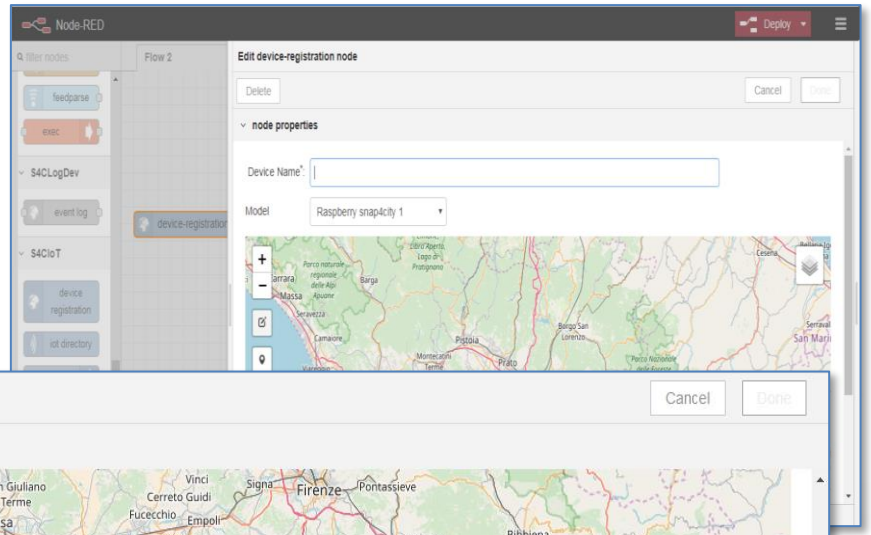


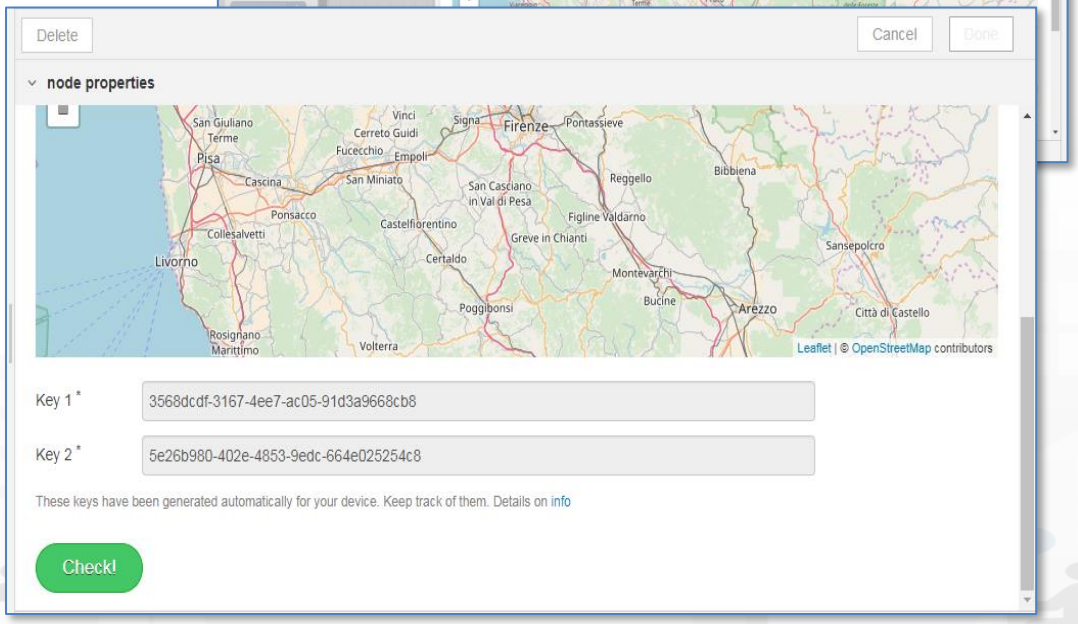


IOT Discovery on IOT Application Development





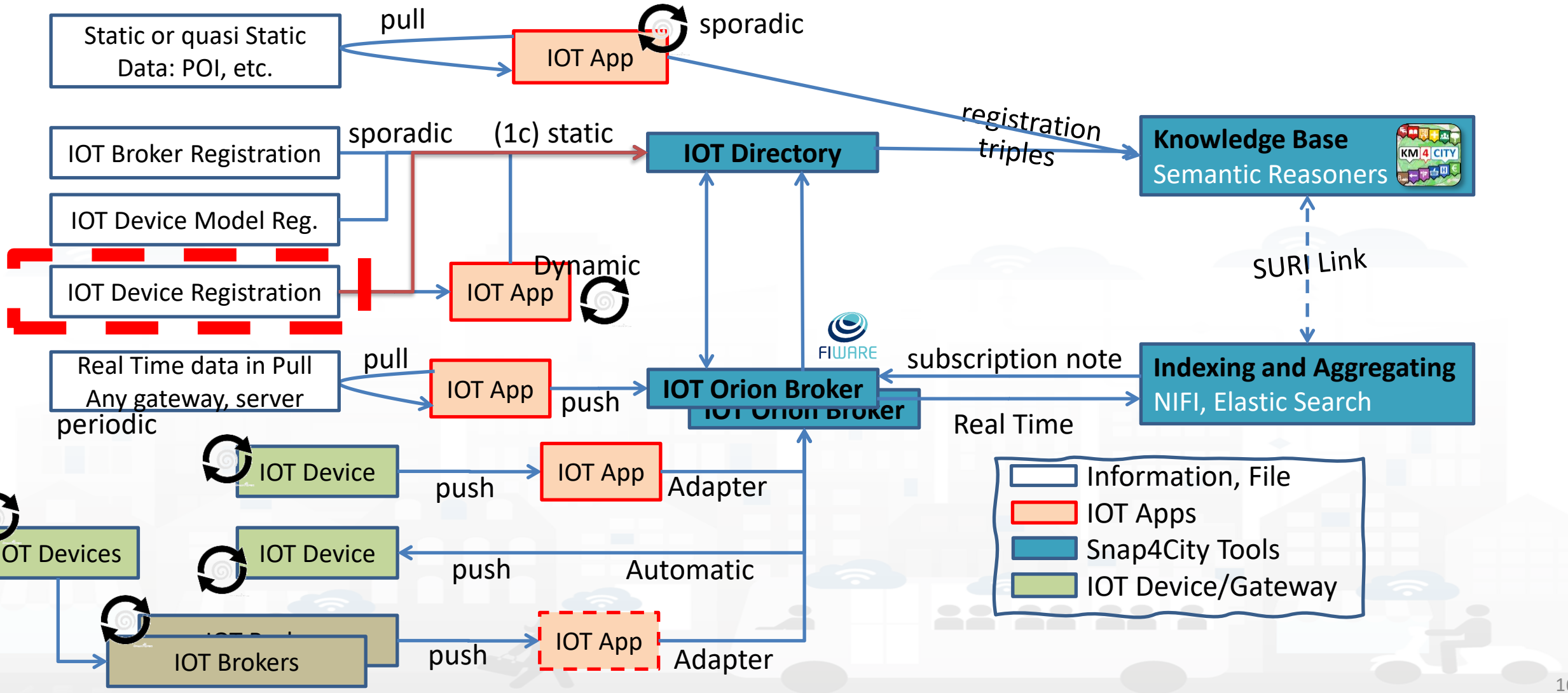




IOT Device Registration from IOT Model

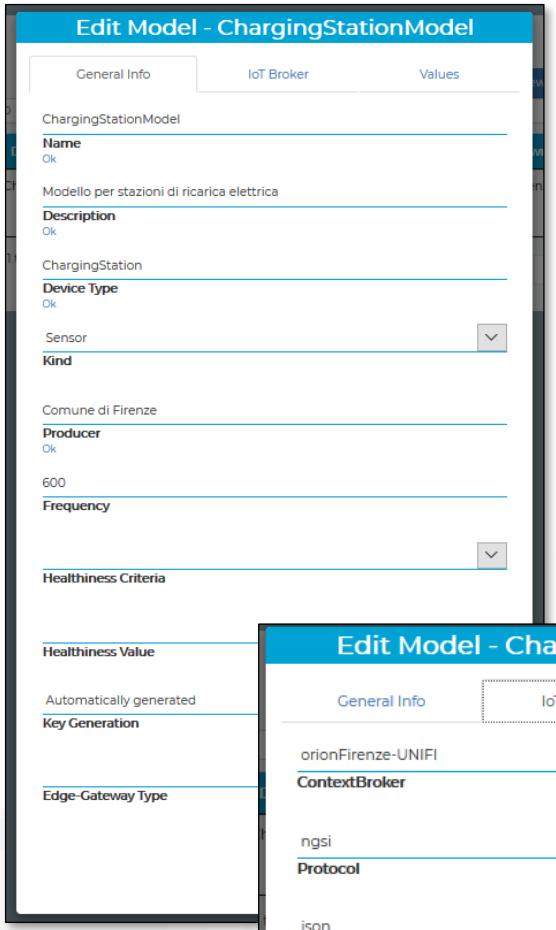


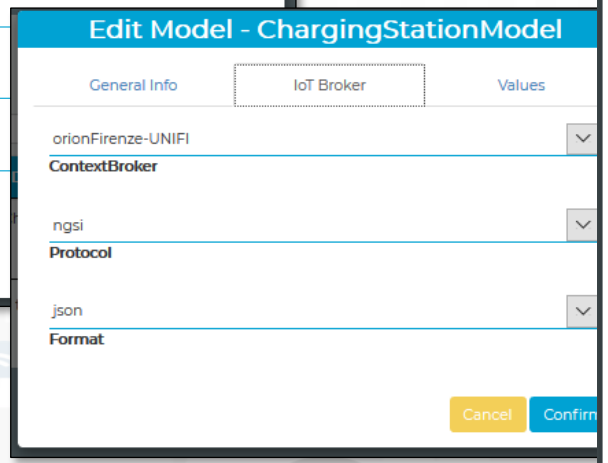
Snap4city Data Ingestion Flow Diagram



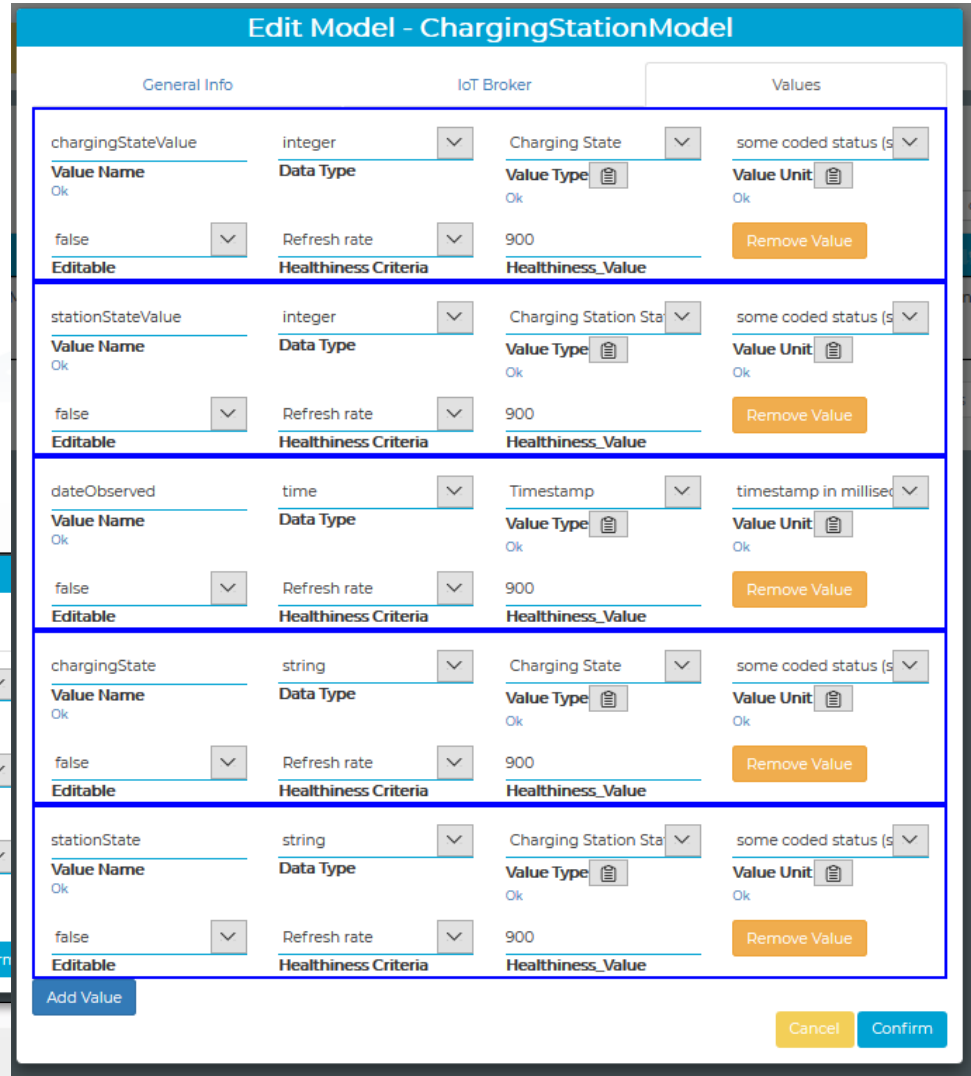
Many IoT Devices? IOT Device Model!!!

- **Prerequirements**: only for AreaManager users
- If you have a set of sensors with the same features,
  - you can create a model and then a set of instances (IoT Devices) in compliance with the model (not time consuming and avoiding errors)
- IoT Directory and Devices > IoT Device Models > ‘New Model’ button

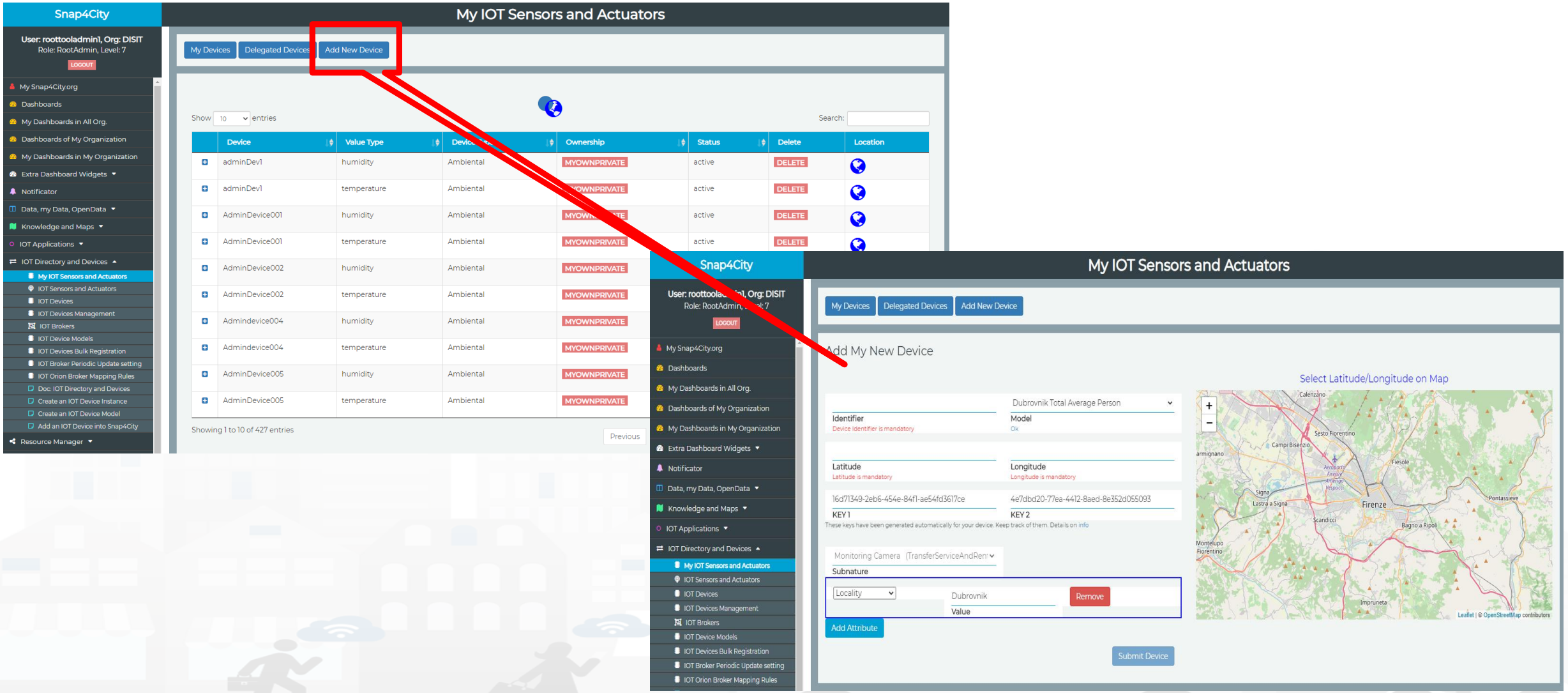




Example: ChargingStationModel



Add IOT/IOE Devices, exploiting an IOT Device Model



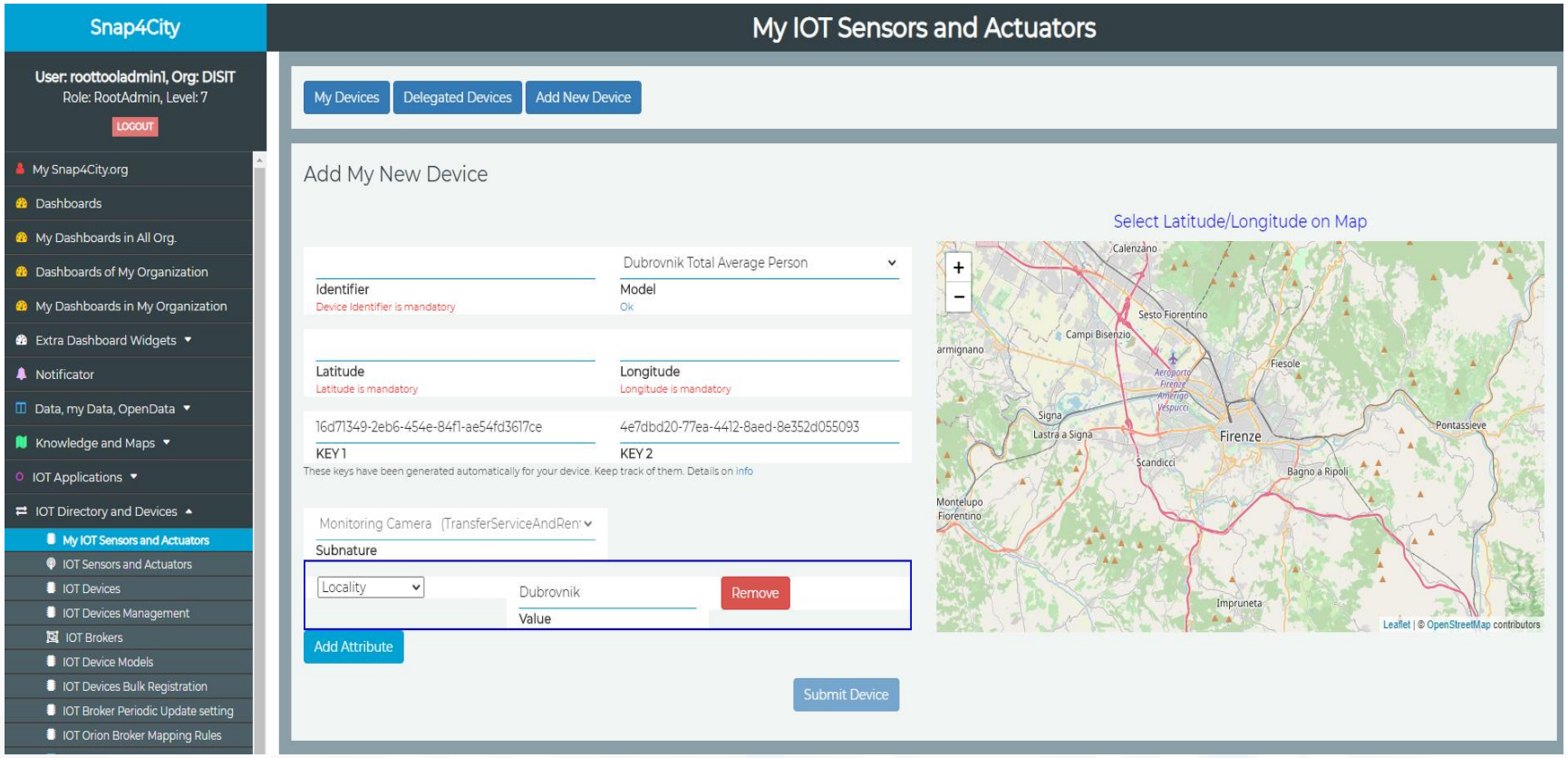
Add IOT/IOE Devices, exploiting an IOT Device Model

Just Buy an IOT Device and register: SigFOX, MQTT, NGSI (FiWare), …

- Attach them by
  - Models
- A range of protocols, formats, approaches

Create your own devices:

- Arduino,
- Raspberry,
- Android,
- LoraWAN + Arduino,
- etc.

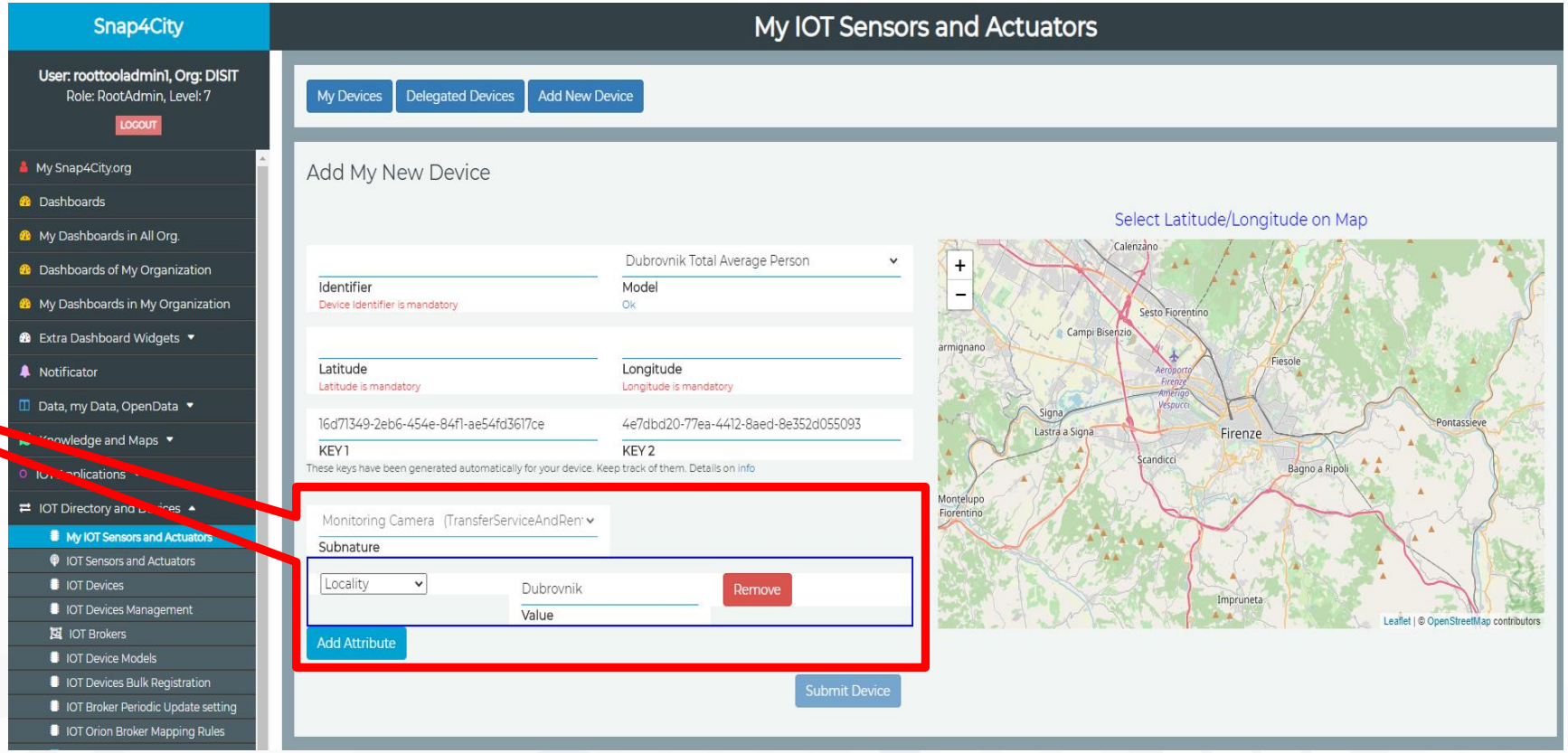


Secure Communication: HTTPS, TLS (K1, K2), Certificates

Add IOT/IOE Devices, exploiting an IOT Device Model

Addition of Static Attributes of the IOT Device

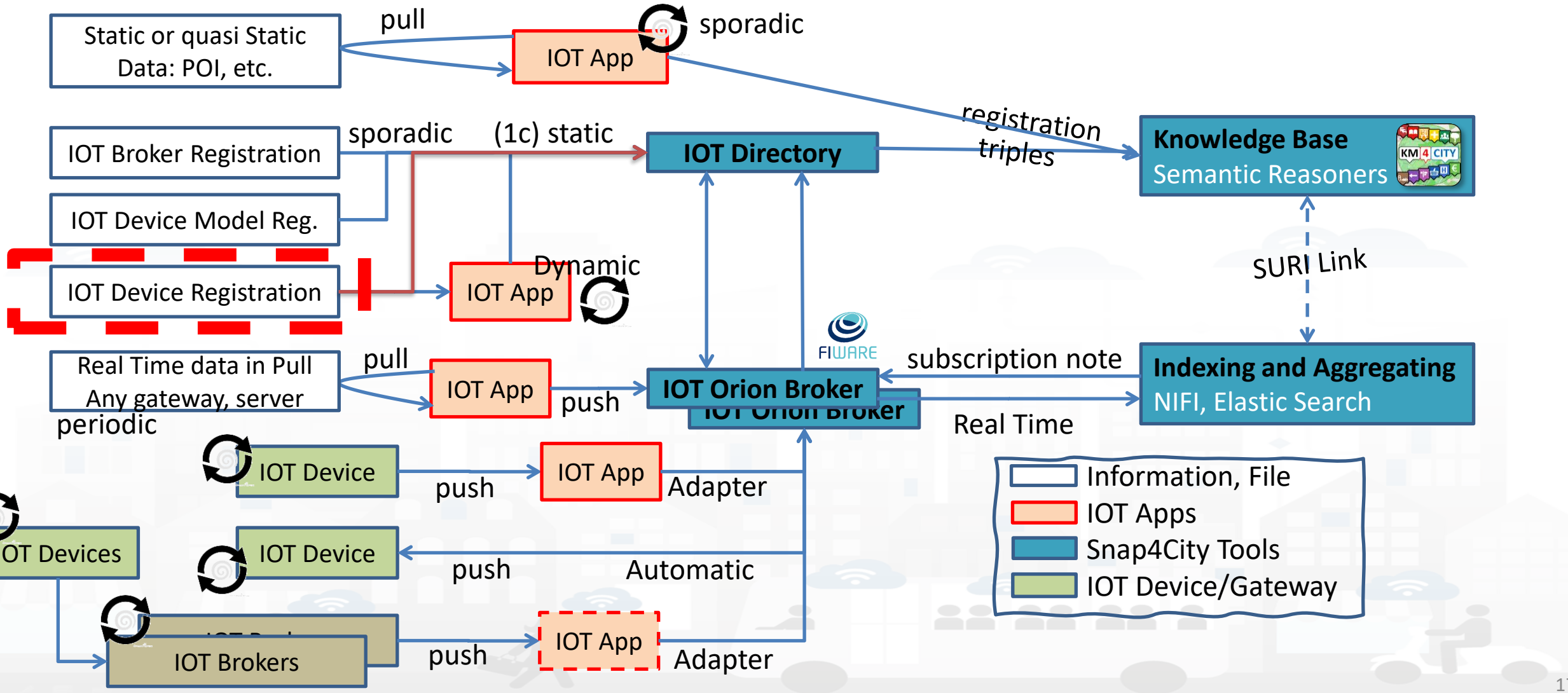
Only if you enabled from model



IOT Device Registration in Bulk from CSV File



Snap4city Data Ingestion Flow Diagram



From CSV → register IoT Devices in BULK

- Create a CSV from the CSV Model provided
- The columns must respect the CSV Model (every field present in the Model)
- One row of the CSV is one new IOT Device
- You have to create to create two keys (called k1, k2) that are necessary to read and write access to the device. They must be different each other.
- Each group of devices, that has the same IoT Model (data set), could/should have the same K1, K2. In this way, it is easier to read or write all the IOT Devices of the same set at the same time.
- These keys are in the UUID v4 format and can be generated online on this website: https://www.uuidgenerator.net

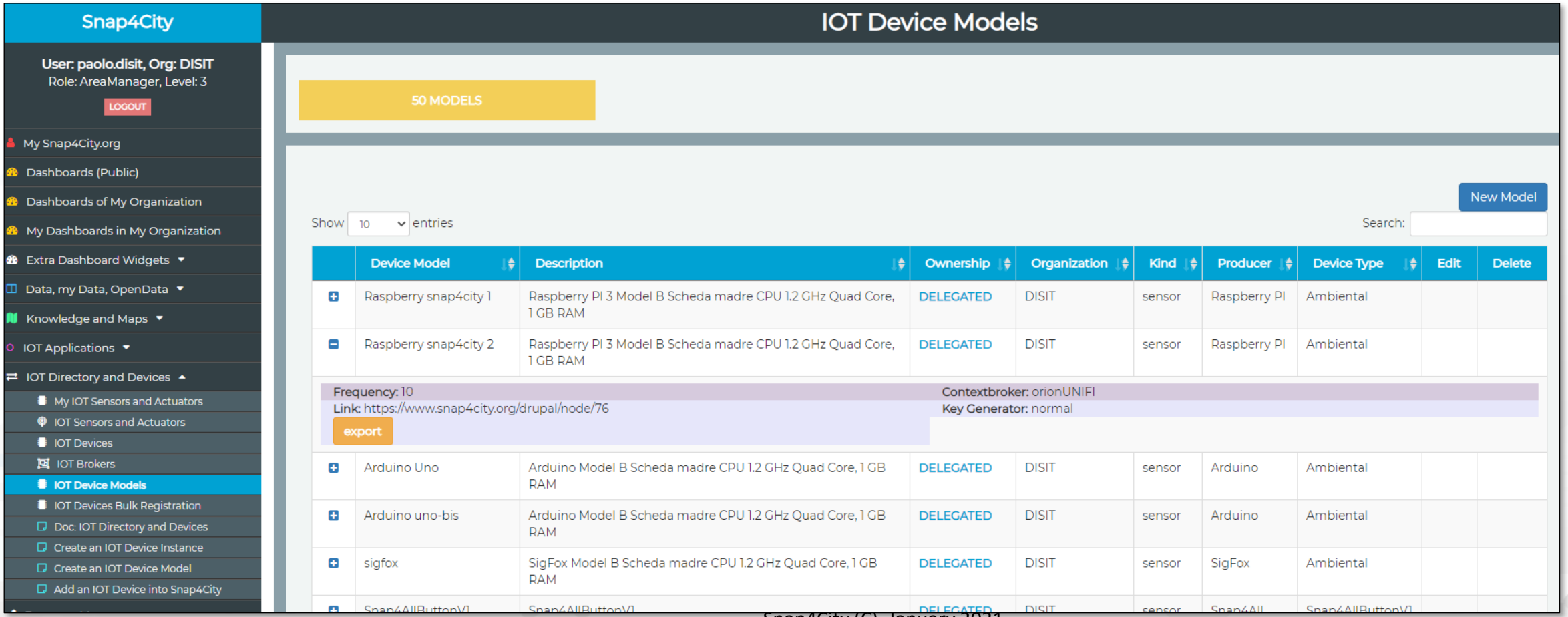
Available example: https://www.snap4city.org/592

File model in https://www.snap4city.org/289

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	name	devicetype	mac	frequency	kind	protocol	format	producer	lat	long	valuename	data_type	value_type	editable	value_unit	health_criteria	healthiness_value	K1	k2
2	eCharging_16ZP22T2AA1S000003	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.222	1.125.338	chargingState	string	charging_state	false	-	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
3	eCharging_16ZP22T2AA1S000003	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.222	1.125.338	chargingStateValue	integer	charging_state	false	#	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
4	eCharging_16ZP22T2AA1S000003	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.222	1.125.338	stationState	string	charging_station_state	false	-	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
5	eCharging_16ZP22T2AA1S000003	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.222	1.125.338	stationStateValue	integer	charging_station_state	false	#	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
6	eCharging_16ZP22T2AA1S000003	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.222	1.125.338	dateObserved	time	timestamp	false	s	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
7	eCharging_15EP22T2AA1S000051	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.483	1.125.993	chargingState	string	charging_state	false	-	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
8	eCharging_15EP22T2AA1S000051	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.483	1.125.993	chargingStateValue	integer	charging_state	false	#	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
9	eCharging_15EP22T2AA1S000051	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.483	1.125.993	stationState	string	charging_station_state	false	-	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
10	eCharging_15EP22T2AA1S000051	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.483	1.125.993	stationStateValue	integer	charging_station_state	false	#	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
11	eCharging_15EP22T2AA1S000051	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.483	1.125.993	dateObserved	time	timestamp	false	s	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38

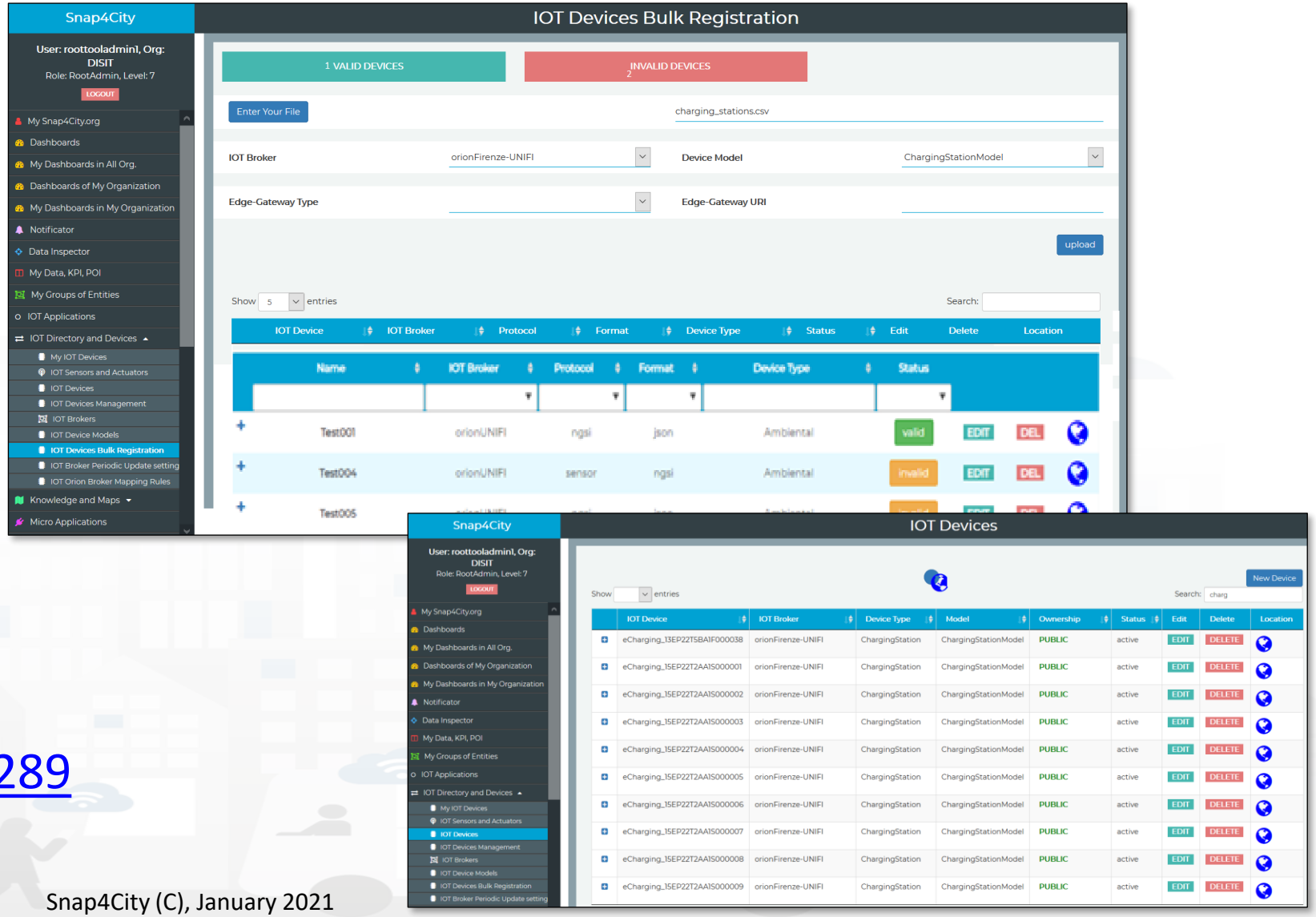
Get a CSV template of your device for filling it

- download a csv template conformant to the IOT Device Model,
  - Go on the IOT Device Model, click on (+) and press the EXPORT button



Register IoT Devices in BULK

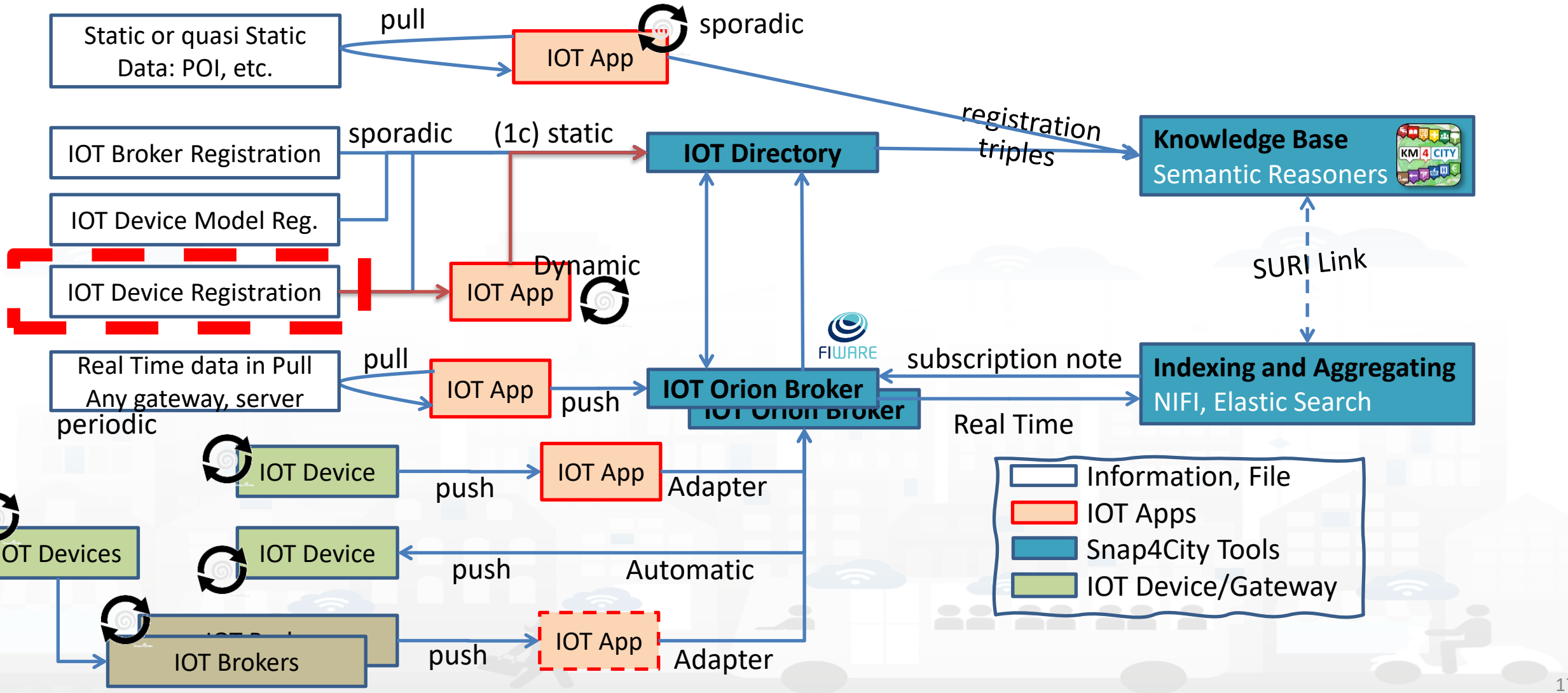
- IoT Directory and Devices > IoT Devices Bulk Registration
- Select: Model, Broker
- Upload the CSV file
- Wait
- Verify the presence of your Devices in:
  - IoT Directory and Devices > IoT Devices
- https://www.snap4city.org/289



IOT Device Registration from IOT App (automation)

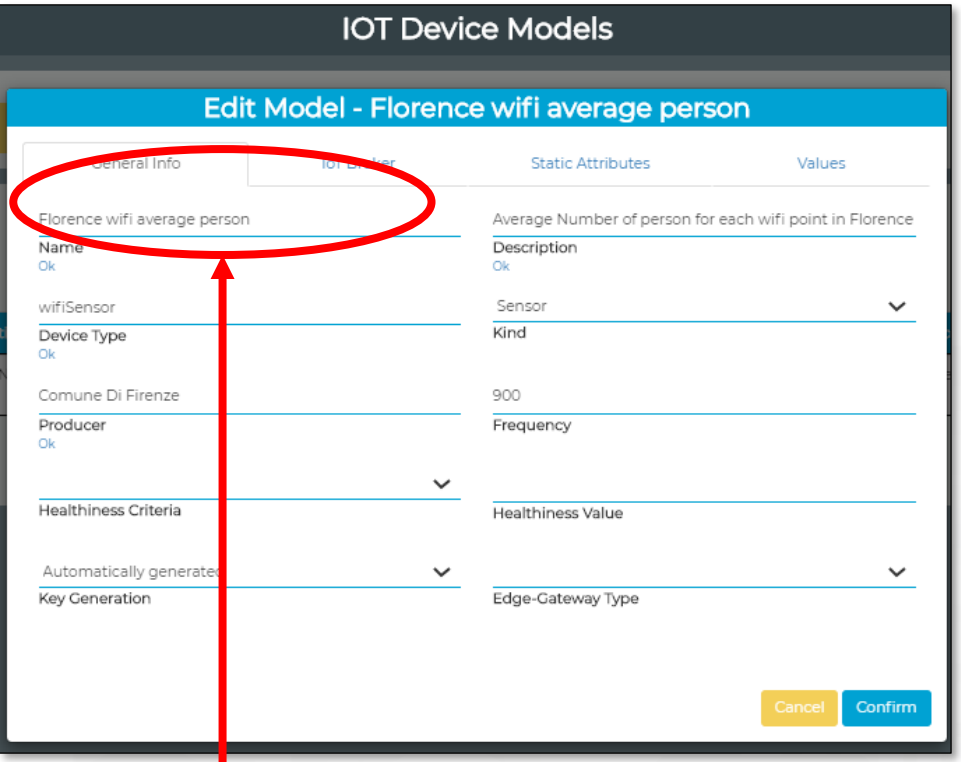


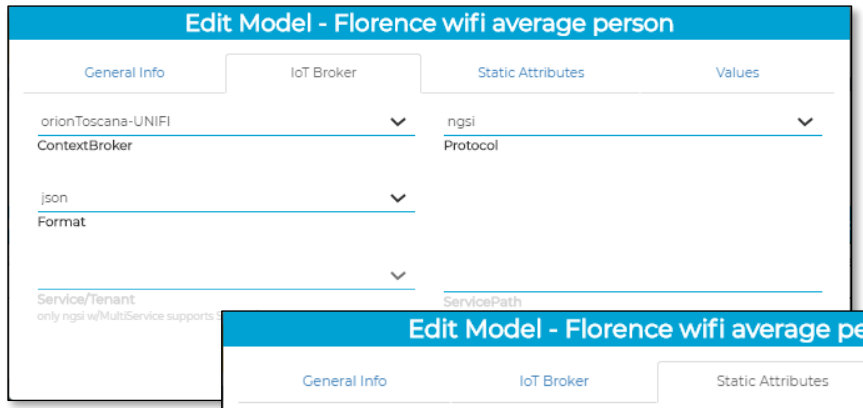
Snap4city Data Ingestion Flow Diagram

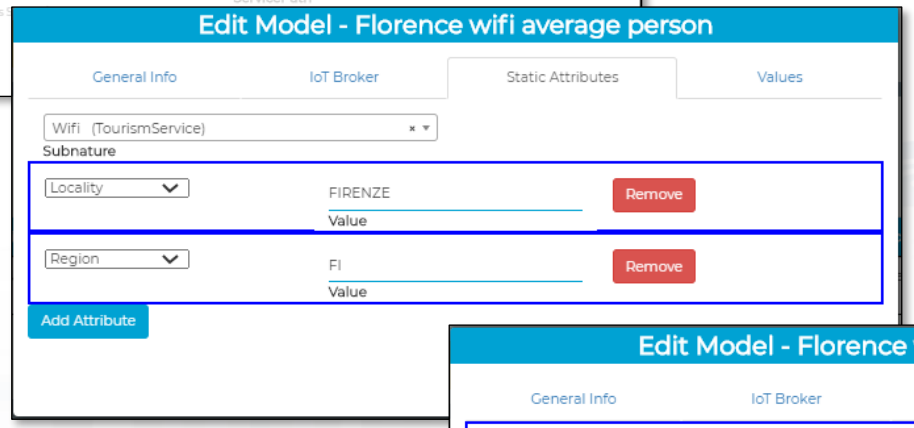


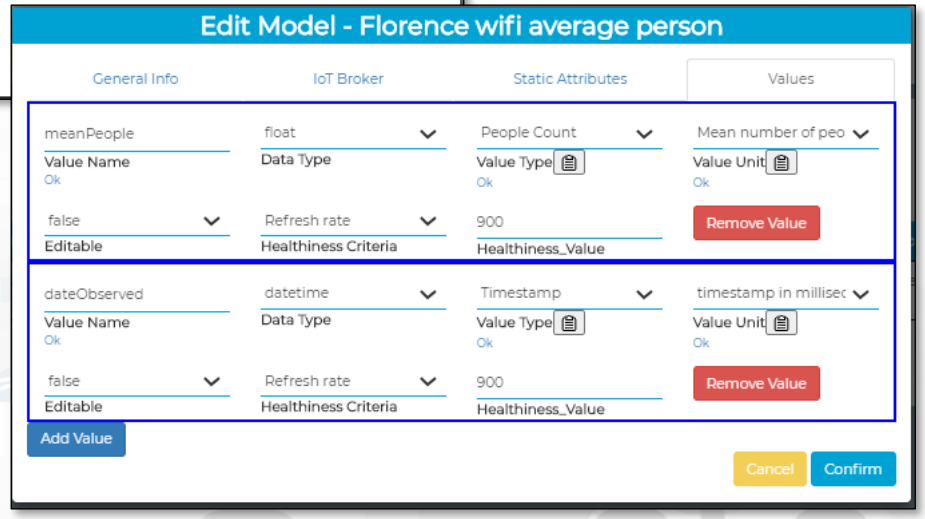
Dataset Creation

1) Model creation



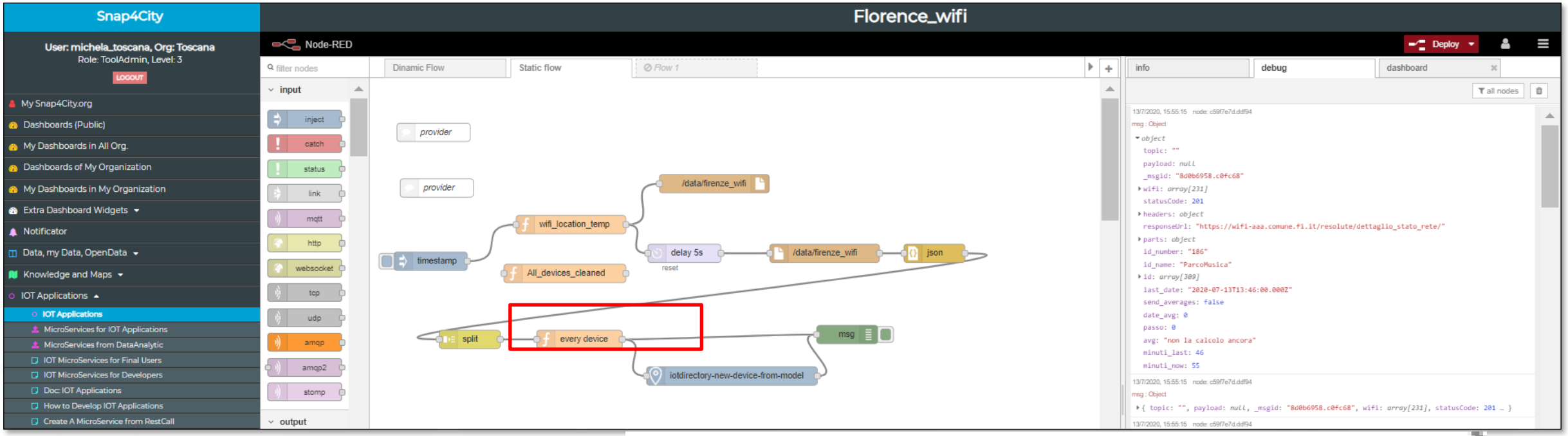






Model name: Florence wifi average person

IoTApp static flow

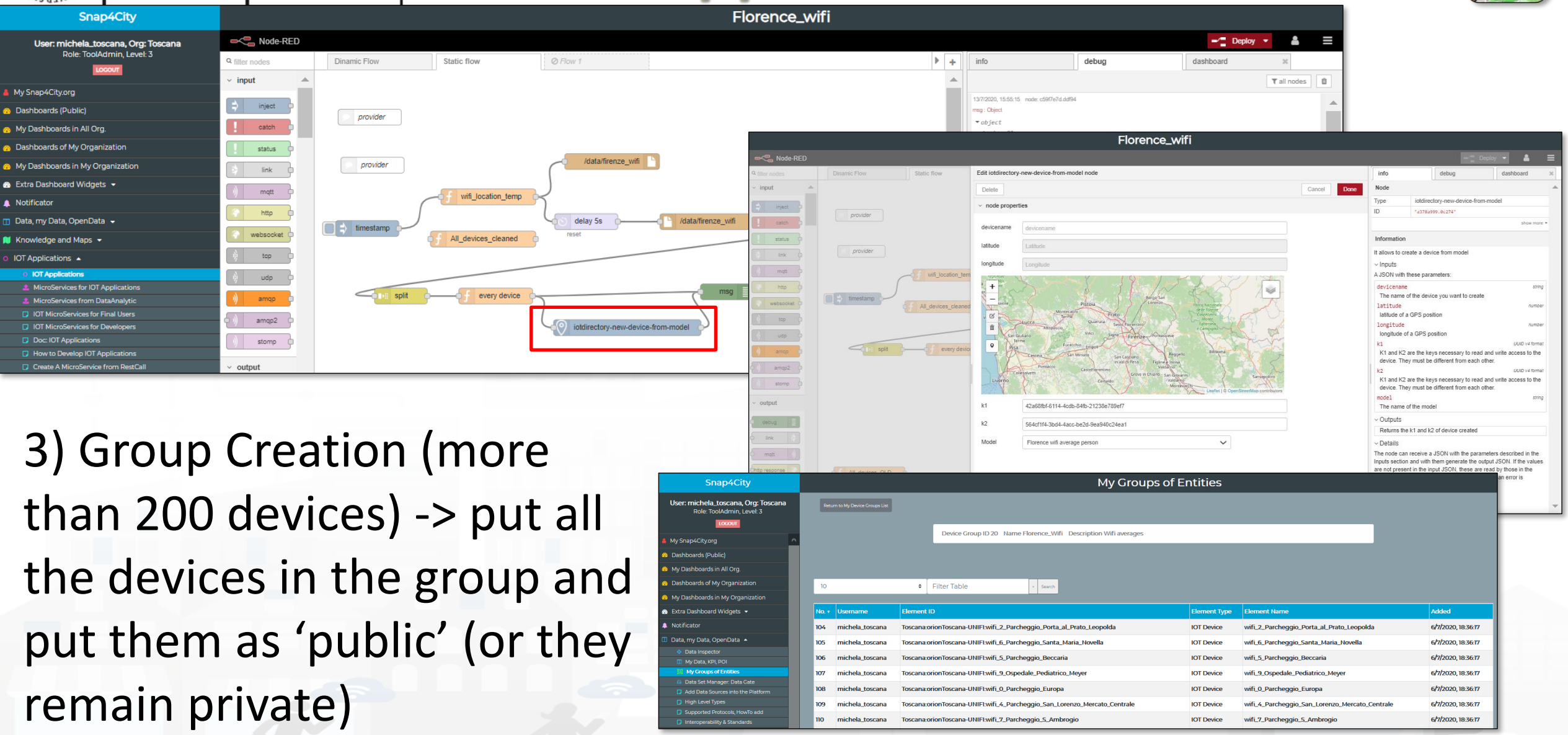


2) IoT Devices Creation from IOT APP



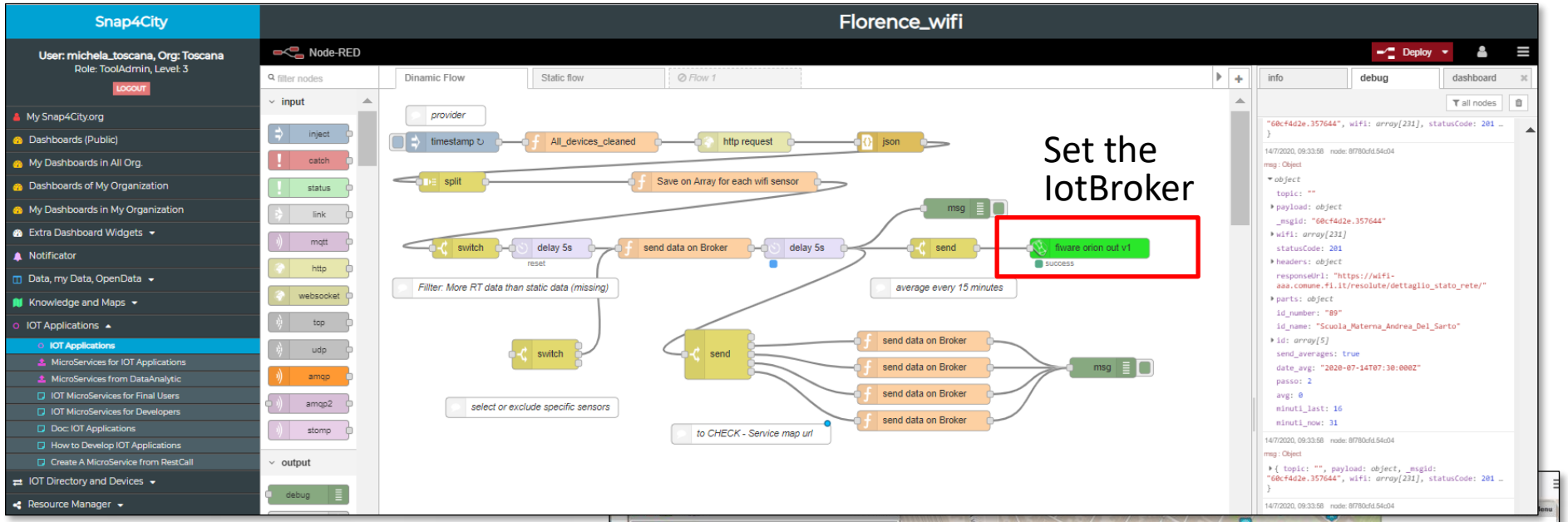
BLOCK: ‘IoTDirectory-new-device-from-model’
Model name: Florence wifi average person

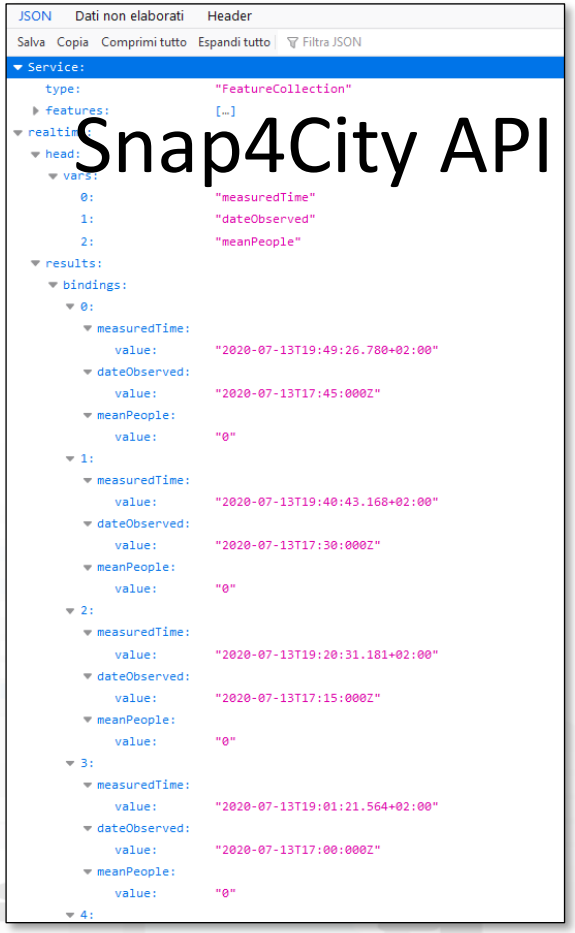
IoTApp static flow

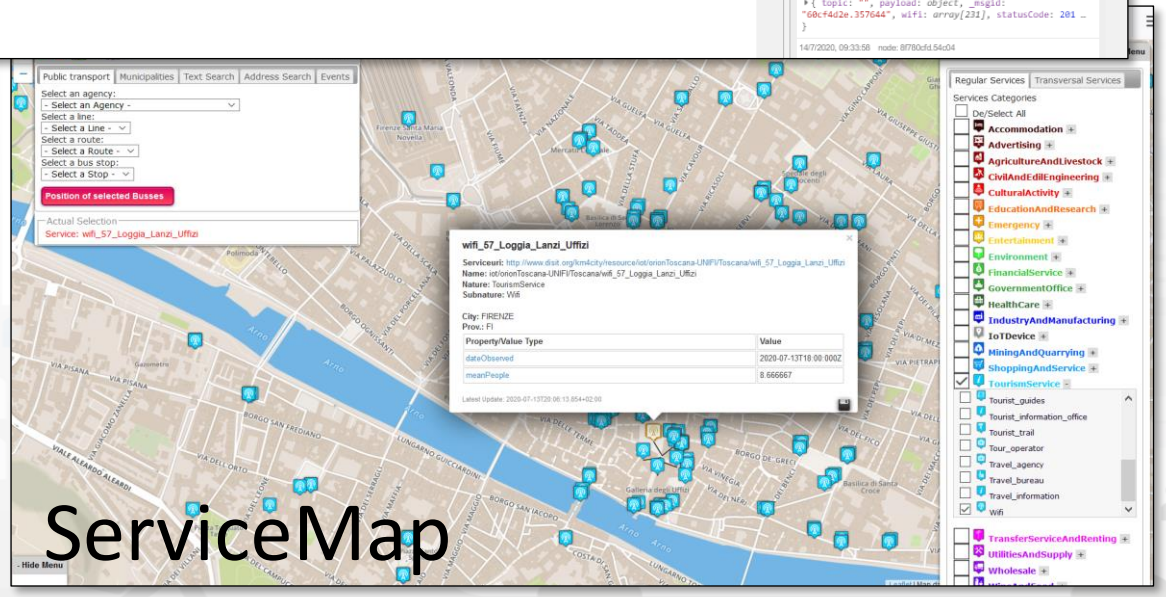


3) Group Creation (more than 200 devices) -> put all the devices in the group and put them as ‘public’ (or they remain private)

IoTApp Dynamic Flow





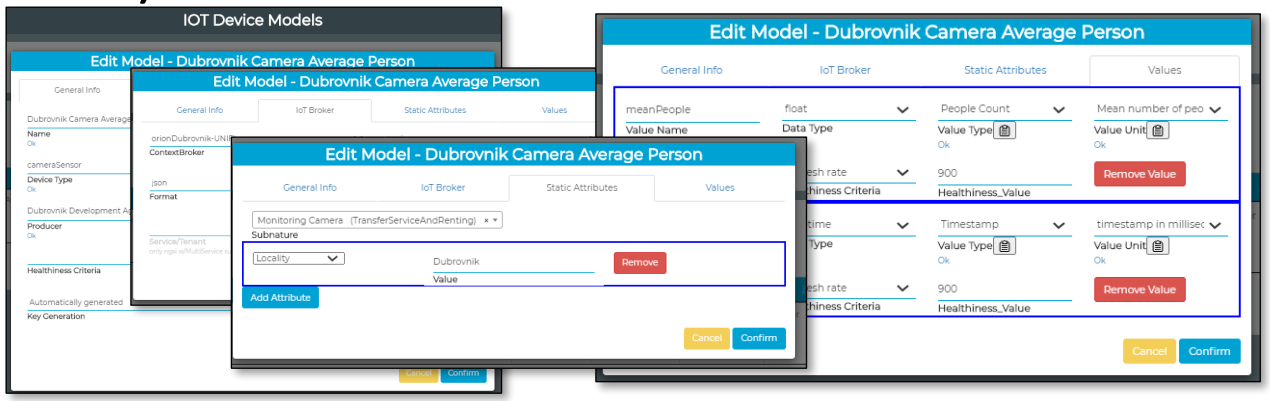


4) Send RT data to the IoTDevices

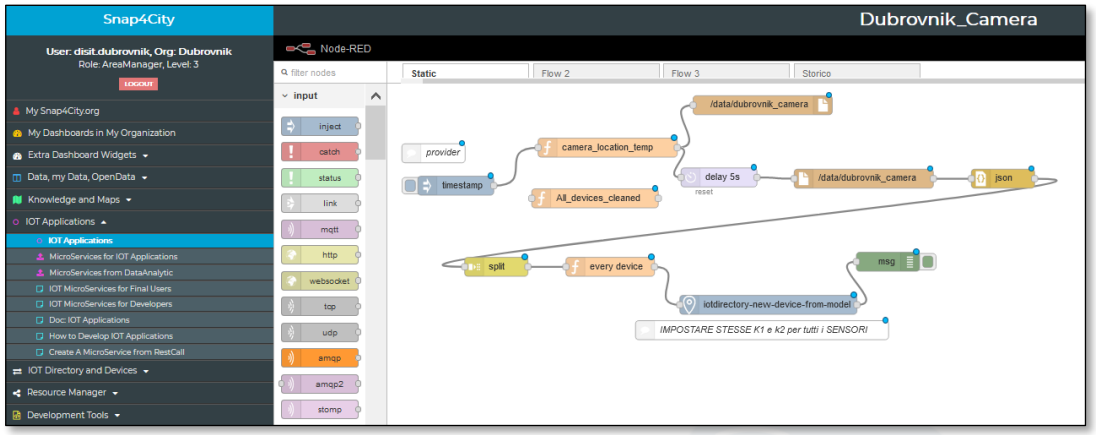
5) Verify RT Data via Snap4City API or via ServiceMap

Work in Progress...

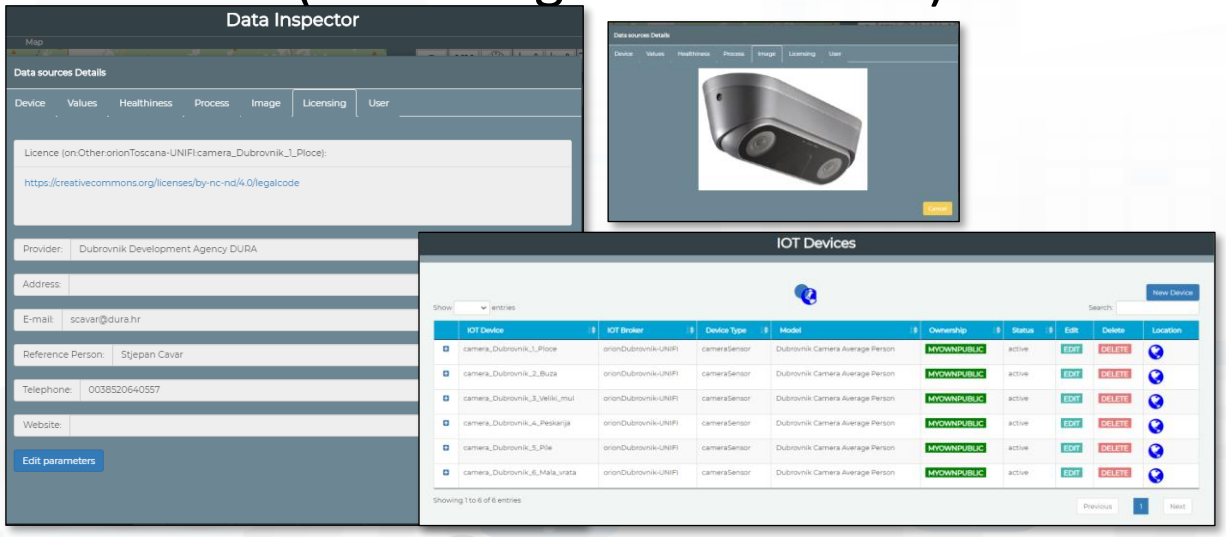
1) IoTModel



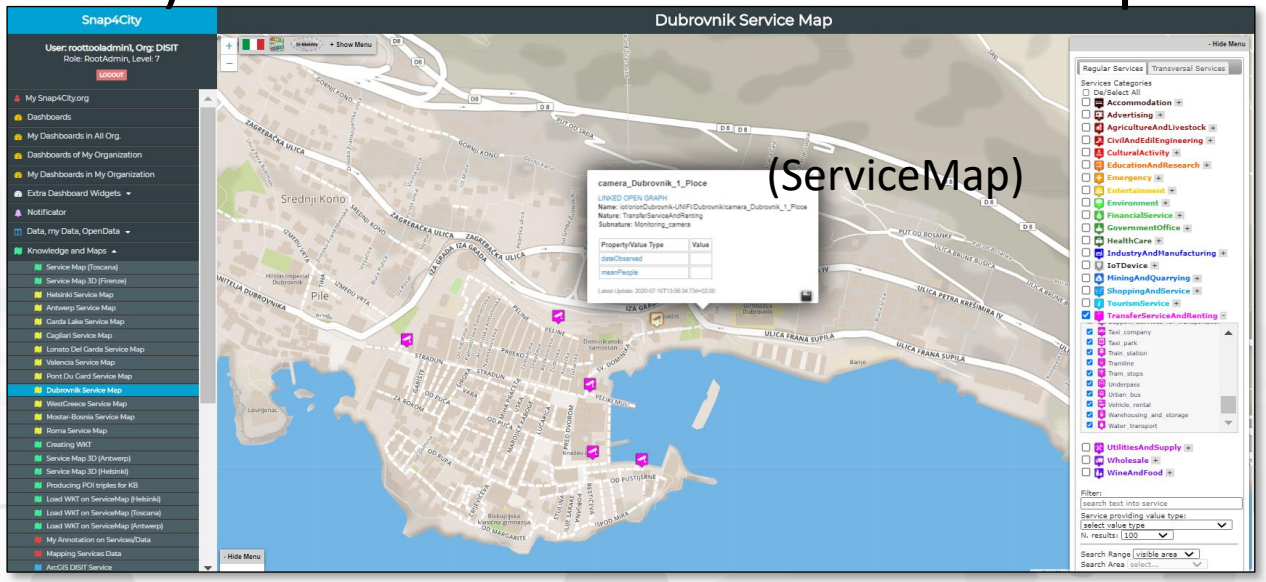
2) Static Flow to create IoTDevices



3) Add the license and Make Public the IoTDevices (according to the license)



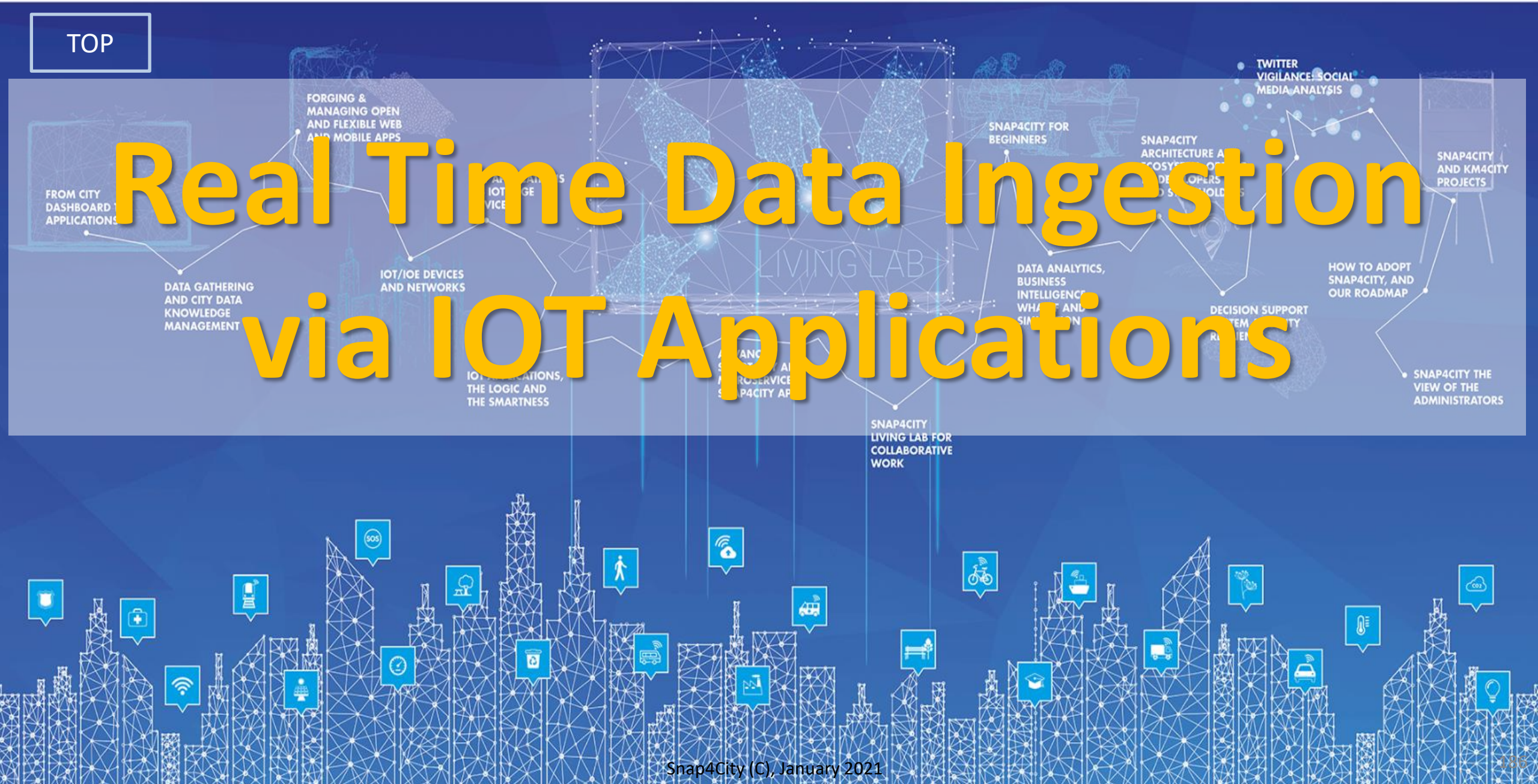
3) Search for the Cameras on Map (ServiceMap)



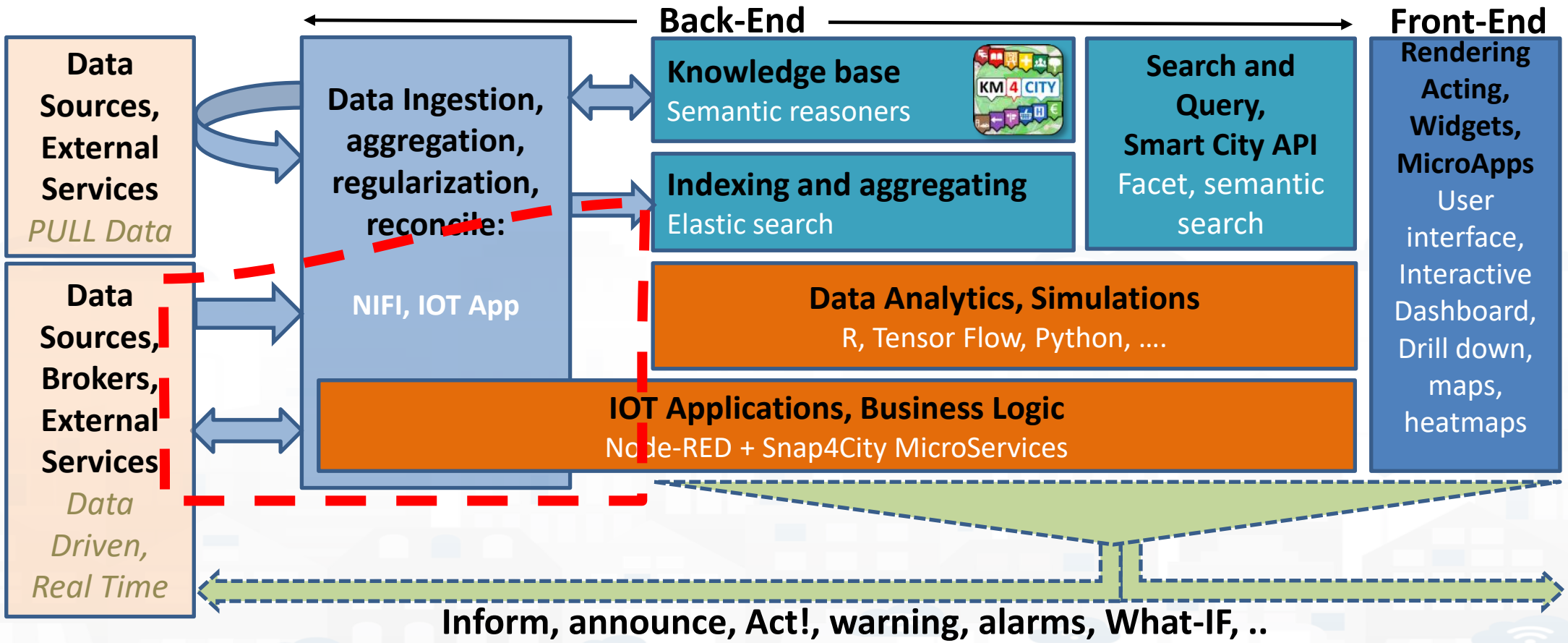
5) **Working on** Dynamic Flow to save Average #people every 15 minutes for each IoTDevice

TOP

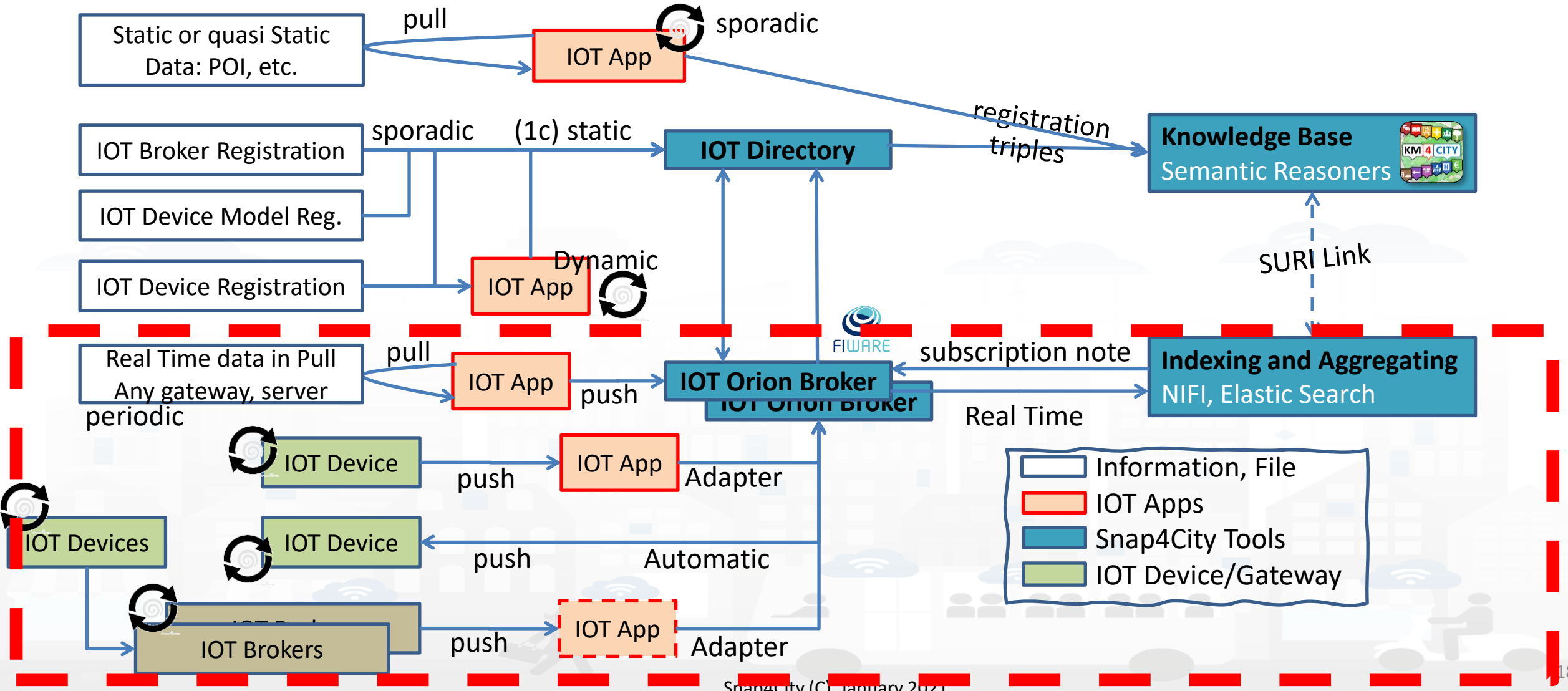
Real Time Data Ingestion via IOT Applications



Snap4City Architecture vs Data Ingestion



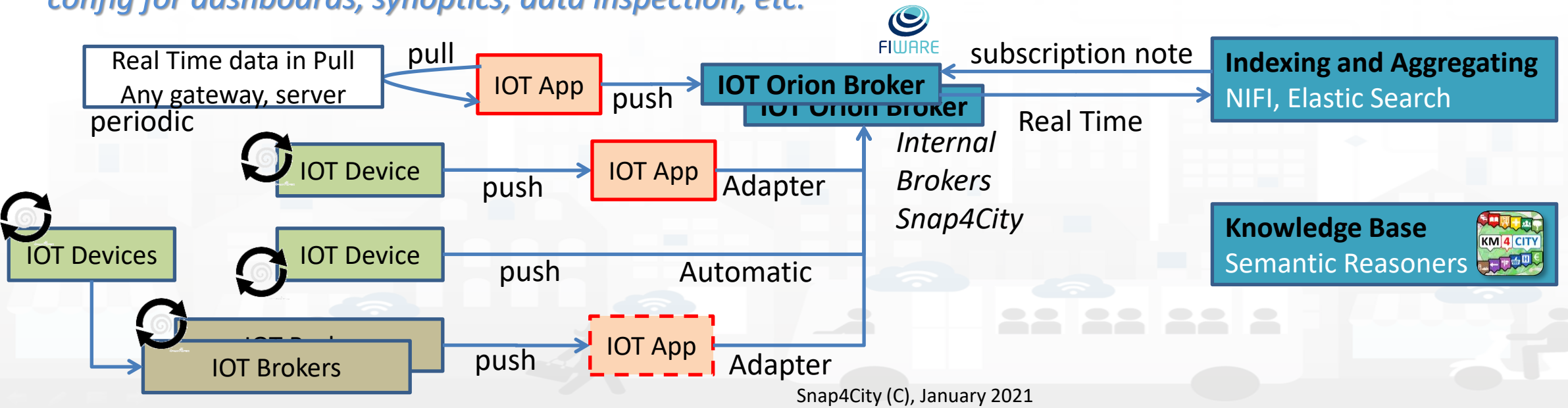
Snap4city Data Ingestion Flow Diagram



A range of Possibilities

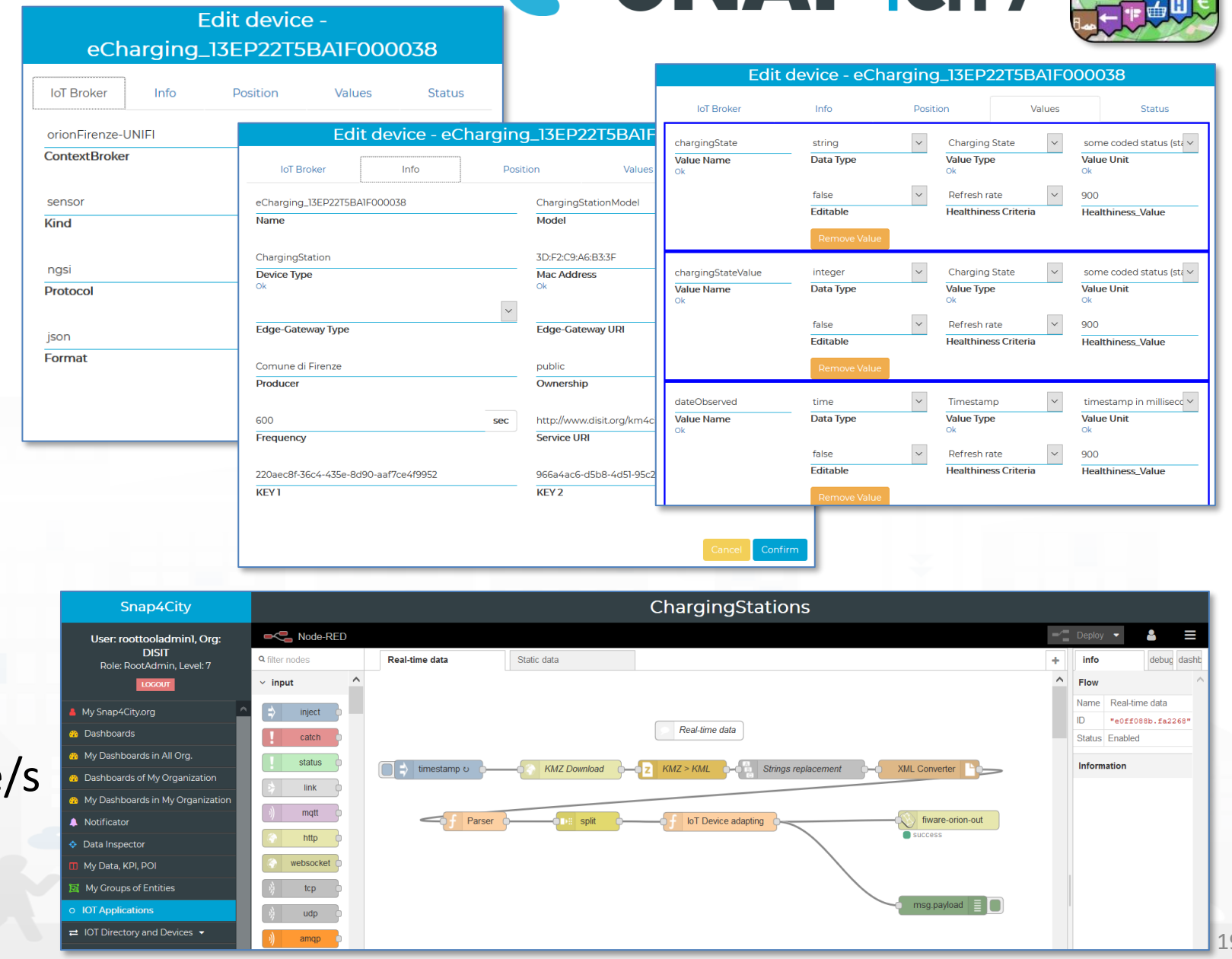
- Any Data can be ingested from any kind of service via an IOT App by using any kind of protocol
- IOT Devices NGSI, MQTT, … can push data into *Internal Brokers Snap4City*
- IOT Devices with other protocols can connect with IOT App for adapting their data to the data of the *Internal Brokers Snap4City*
- IOT Devices can be connected to any IOT Broker of any protocol and the IOT Broker can talk with *Internal Brokers Snap4City*, with or without adapter

THUS any data passing on our Internal IOT Brokers is automatically indexed into Elastic Search and Knowledge Base creating the full support for query and data shadow, usable immediatly without any config for dashboards, synoptics, data inspection, etc.



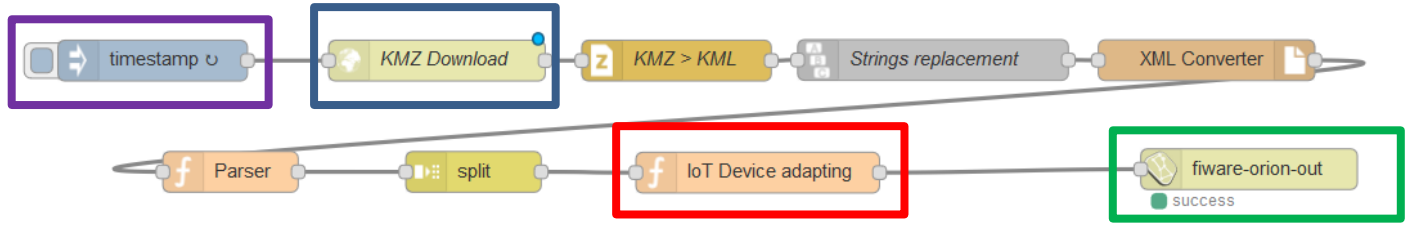
Dynamic Data

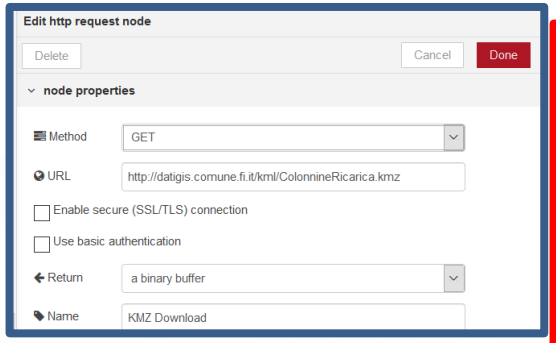
- **Hyp**: IOT Device created
- Create an IOT App to send the dynamic data to your Device/s (in push or Pull)
- The IOT App is based on the data you have:
  - Data ingestion
  - Data transform to a json referring to the IOT Device/s
  - Choose the correct broker



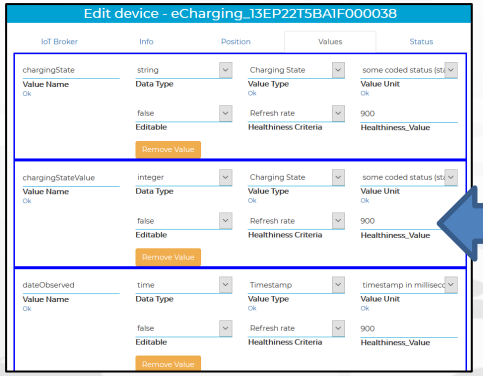
Dynamic Data

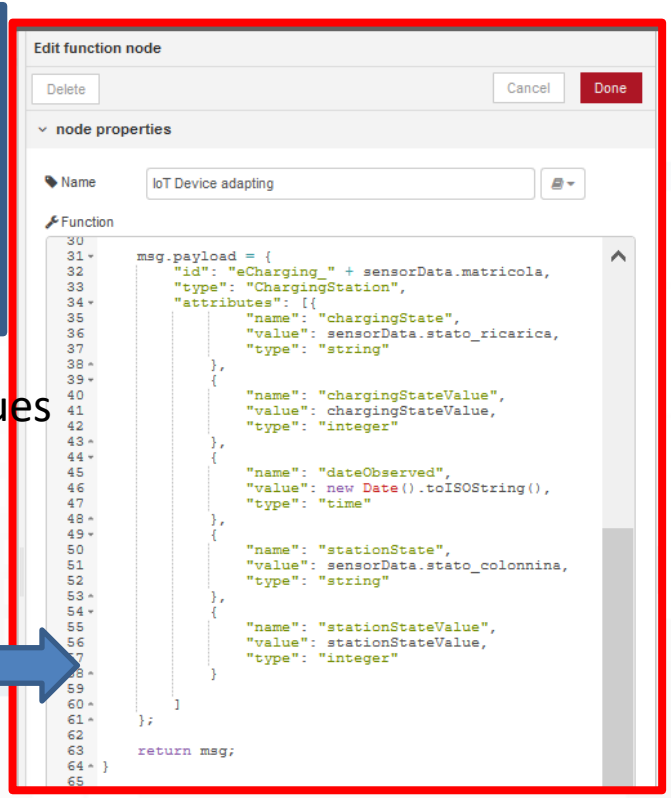
- Charging Station in Florence (Open Data)
  1. Use an **inject** block to chose the frequency of update
  2. Use **http** block to download data
  3. Use a set of blocks to convert data from xml to json
  4. Use a **function** block to create a json referring to your Model and Devices (ChargingStationModel)
  5. Use a **fiware** Orion block to send the dynamic data:
     - selecting the broker
     - Put your K1, K2

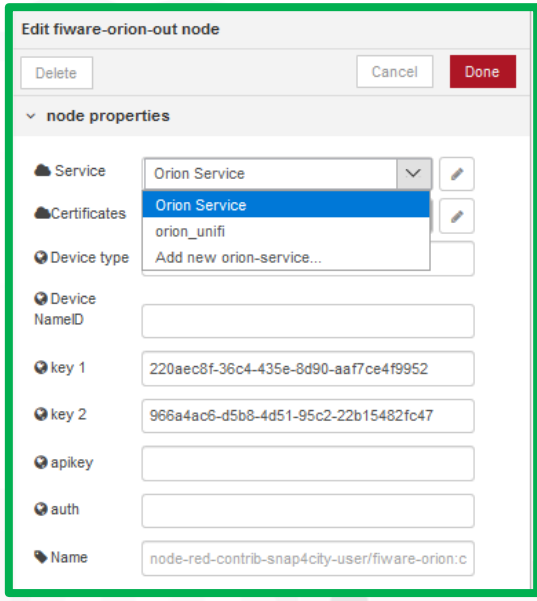




ChargingStationModel: values







Further readings

- 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

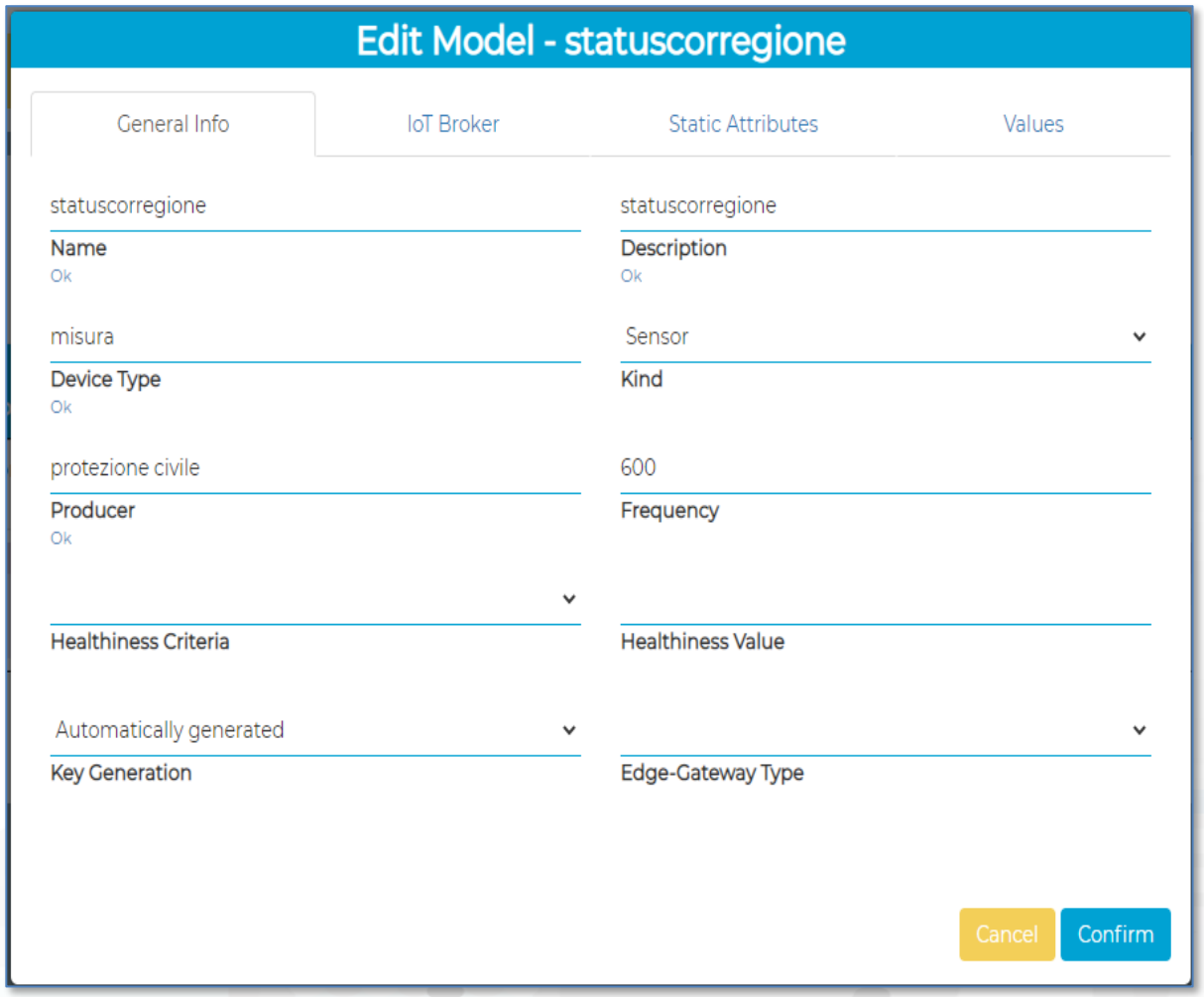
Further readings

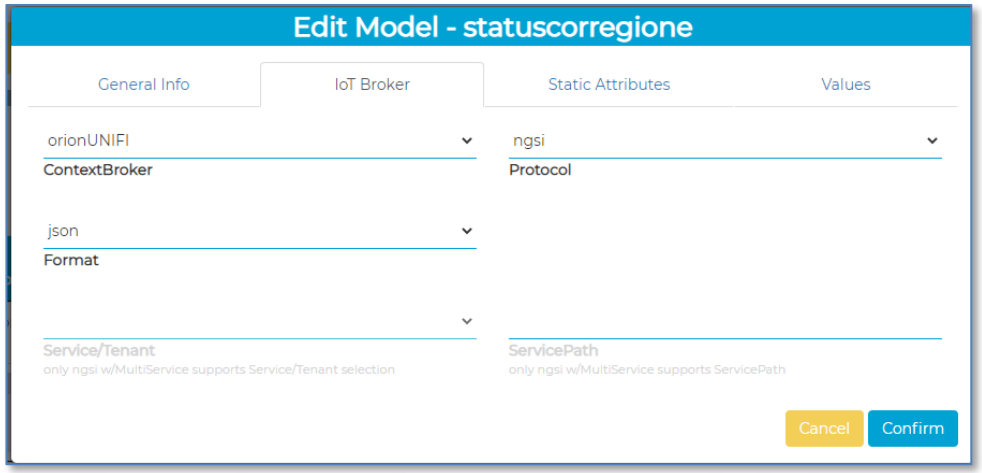
- HOW TO: add a device to the Snap4City Platform
- HOW TO: add data sources to the Snap4City Platform
- HOW TO: add IOT Device data source from external broker to the platform.
- TC9.13: How to upload a local file into your IOT Application
- TC9.2. Managing heterogeneous File Ingestion, protocols, formats via IOT applications, and open standards
- TC2.25. Registering external MicroService calling RestCall services, using it on IOT applications

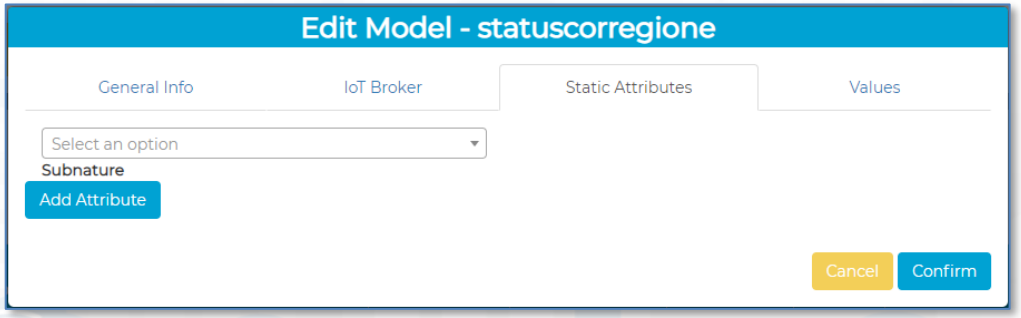
A Complete Example for Time Series: IOT Device Model + IOT Data Ingestion

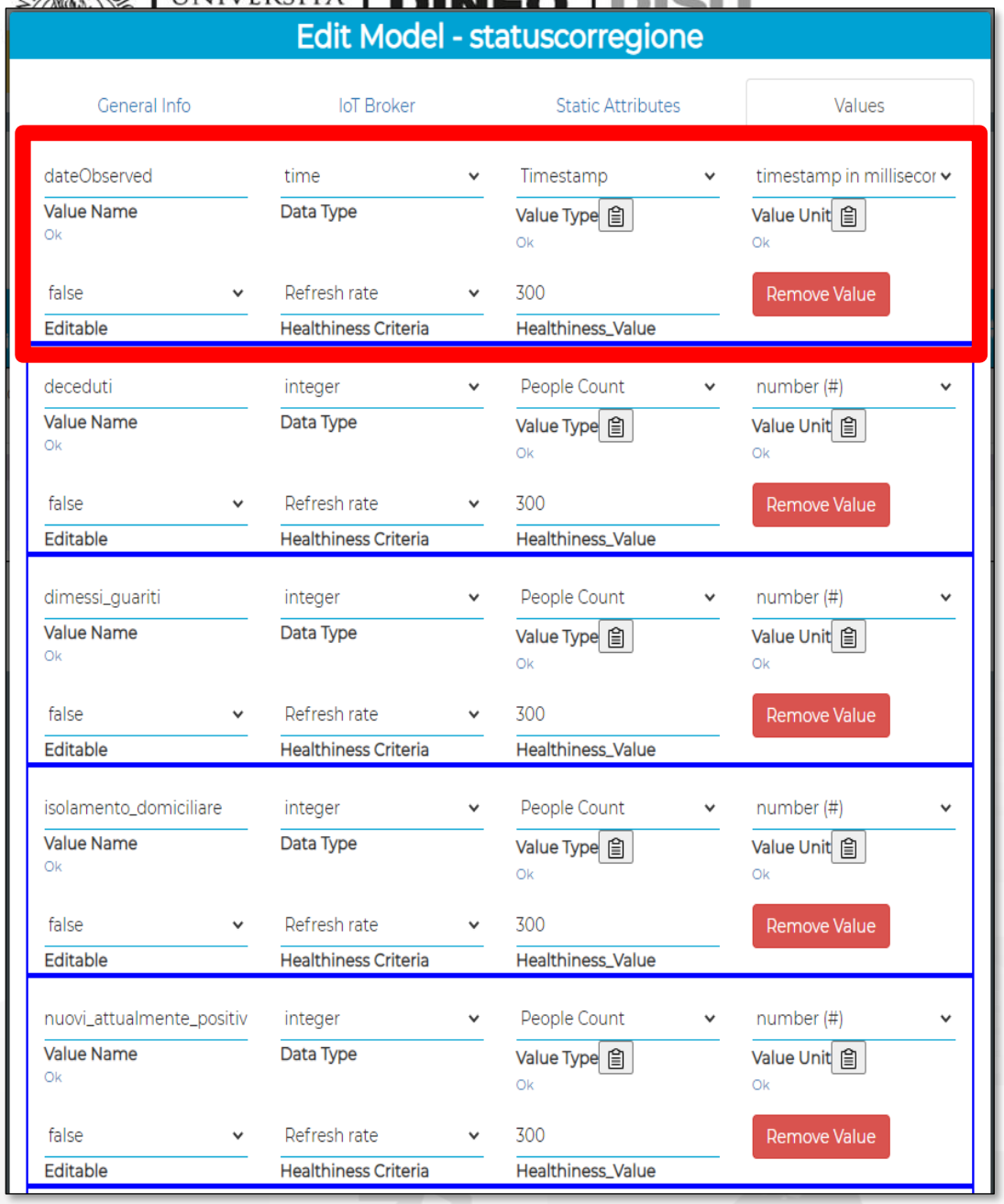


I have created an IOT Device Model as:



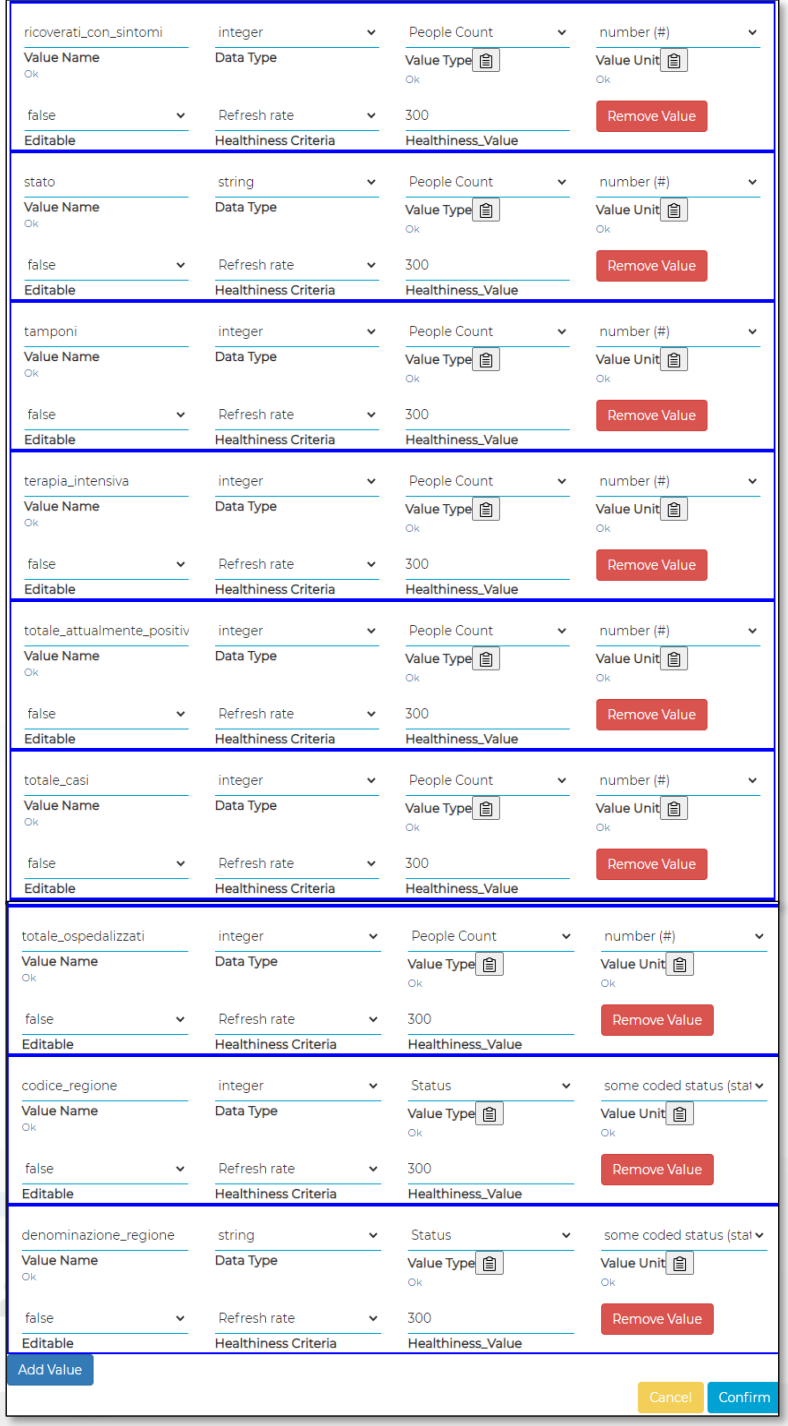




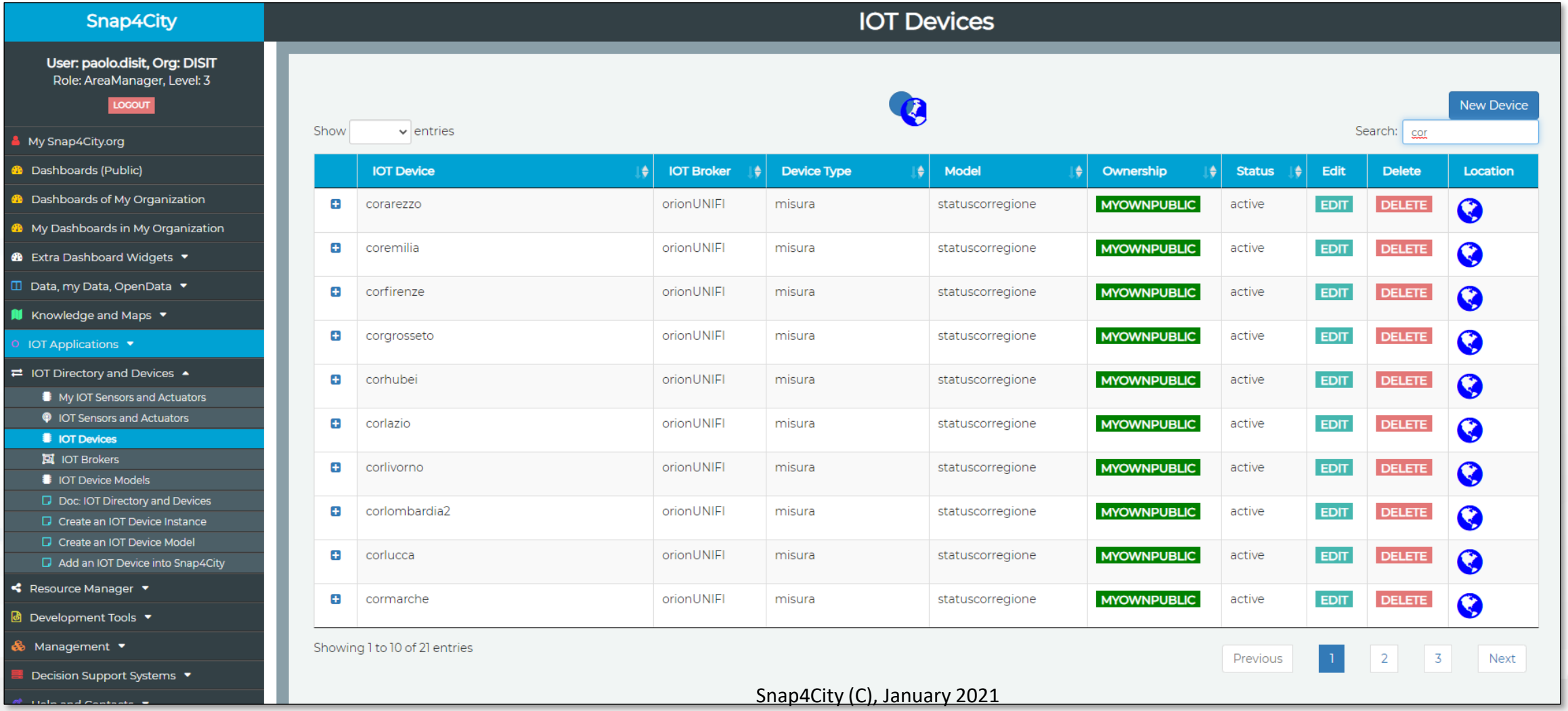


For Time Series

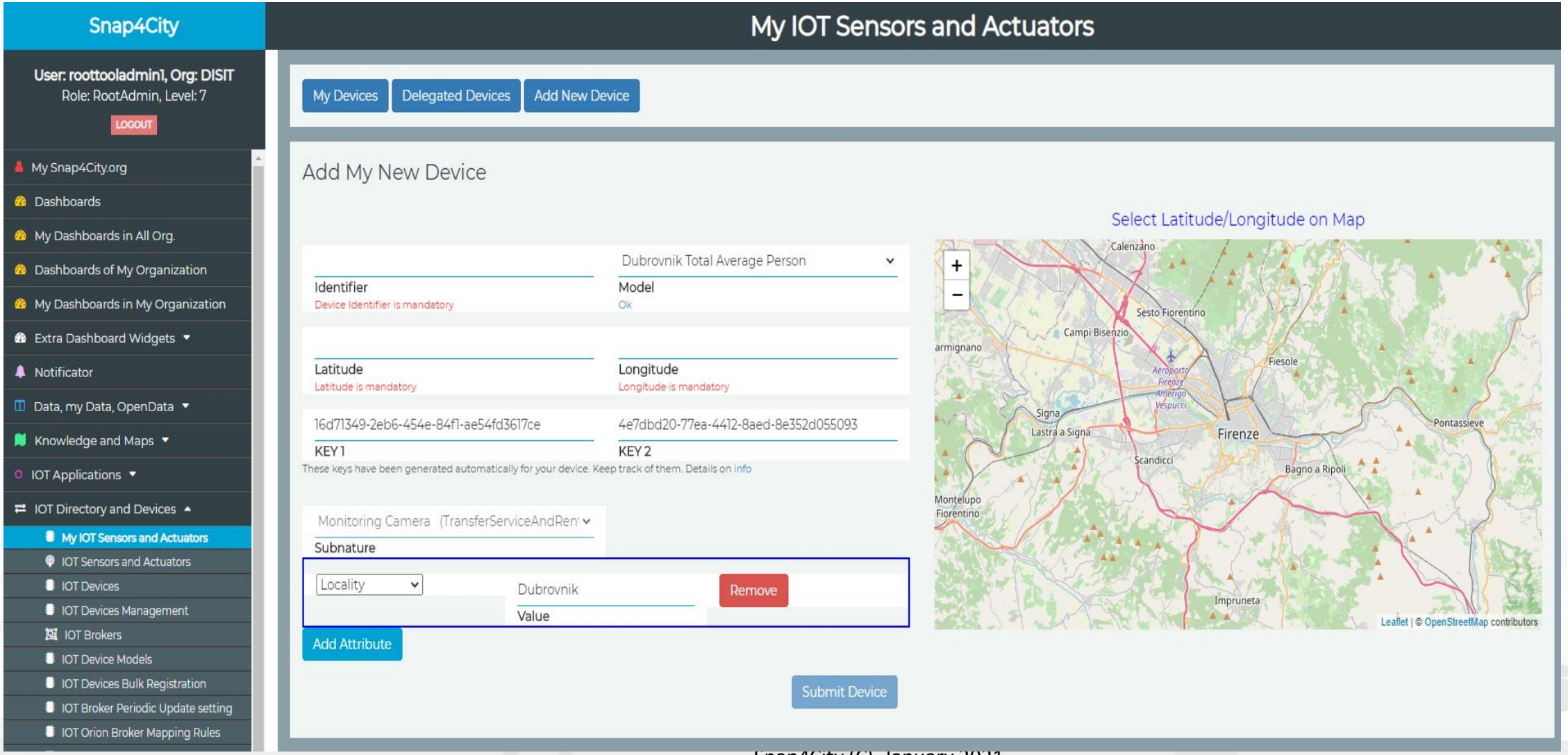
- ValueName: dateObserved
- DataType: time
- ValueType: timestamp
- ValueUnit: timestamp in millisecond
- E.g.: ISO string of the date-time



From IOT Model I have created some instances: the IOT Devices



They have been created by «Add new Device»



IOT Device from IOT Model by Providing:

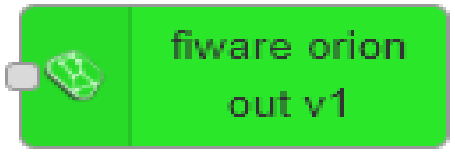
- **NAME (it has to be unique)**
- Select the IOT Model: «**statuscorregione**»
  - Thus the K1, K2 appears since the model is associated to an Orion Broker that needs to have them, the tool generate them for you but you can impose if you like
  - See in previous slide the ID name of the IOT Broker used
- **Lat** and **Lon**, GPS coordinates you can:
  - pick on the map
  - Write the coordinates manually and see the pin on map

Once created the IOT Device you may send data on it

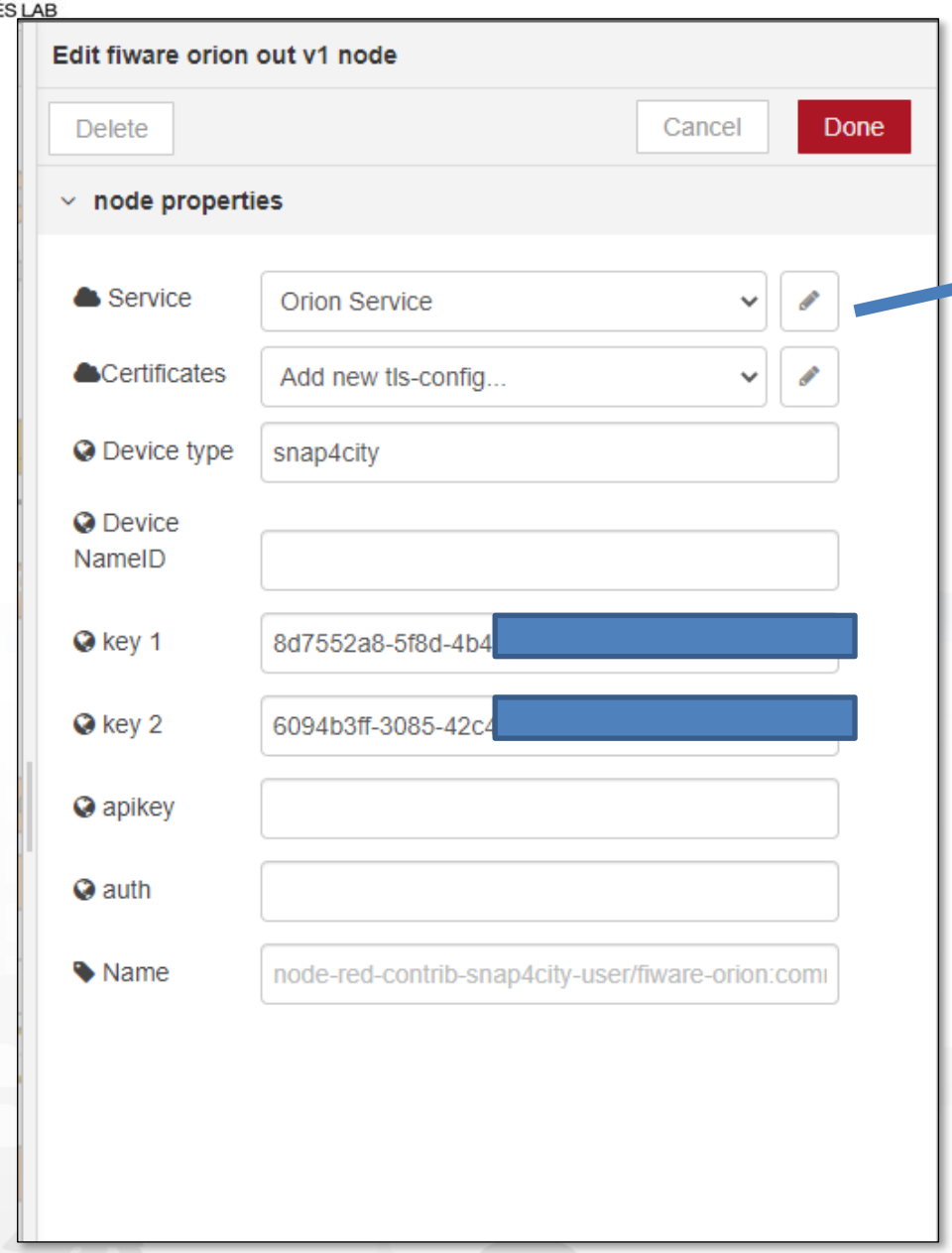


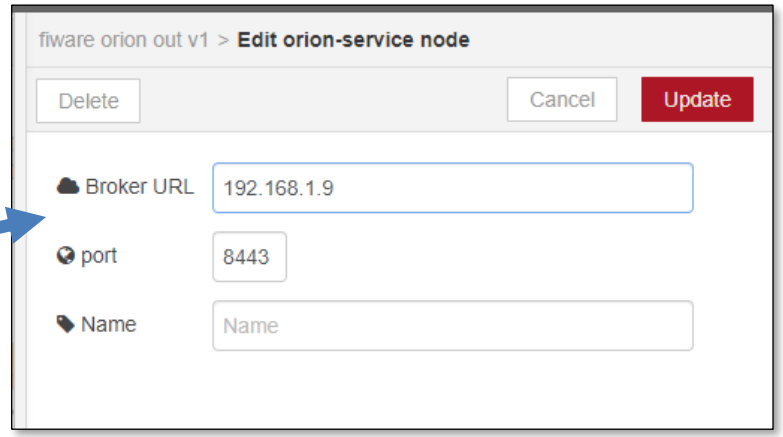
- You may create an IOT App, where:
  - Function: is preparing the JSON package
  - Block «Fi-Ware Orion OUT V1» is sending the data to the Orion Broker. Namely: «OrionUNIFI»
  - Please note that several version of IOT ORION Brokers and protocols exists:
    - So that you have to know which protocols you need to use for your broker

Settings ?



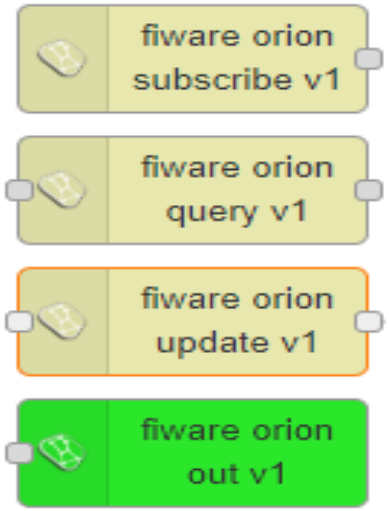
- Certificates are automatically loaded at the first authentication
- Done!!c



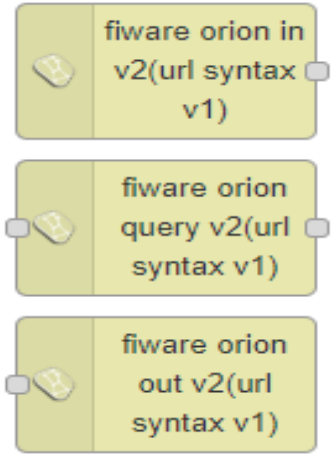


- IP if the Broker is in cloud
- Symbolic address of IOT Broker can be taken from IOT Directory

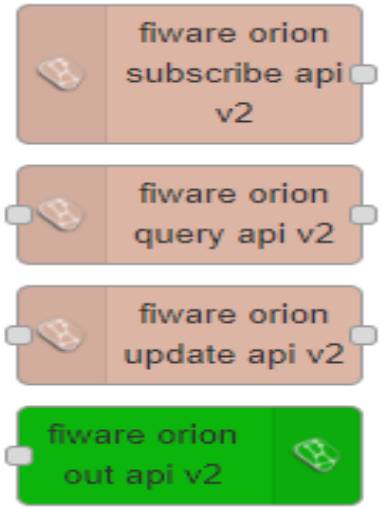
NGSI versions



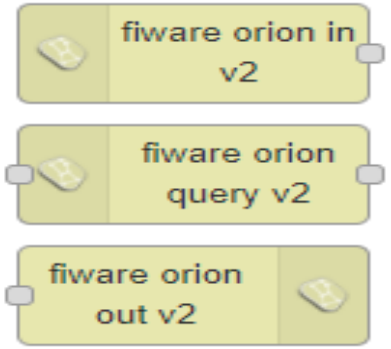
- Orion Broker of V1 with **NGSI syntax of V1** + Secure Filter of Snap4city



- Orion Broker of V2 with NGSI syntax of V1



- Orion Broker of V2 with **NGSI syntax of V2**

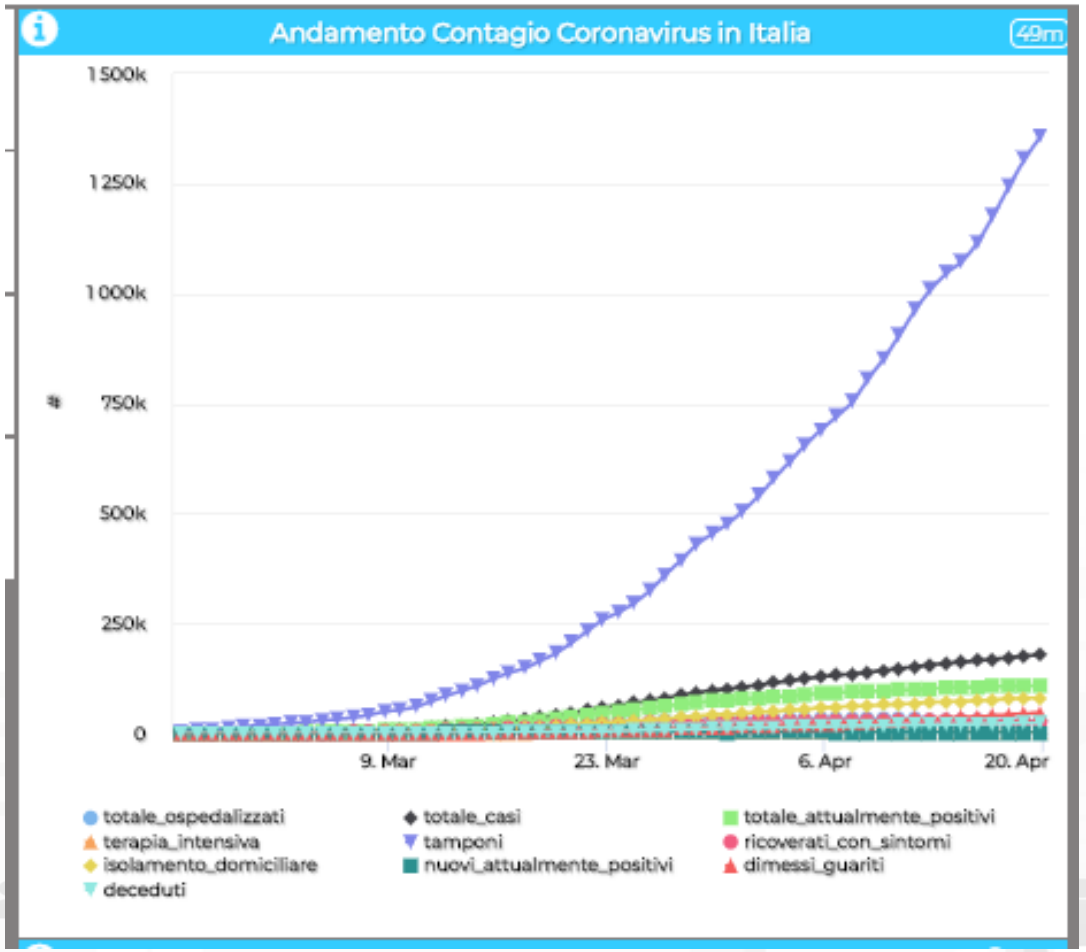


- A Json from the IOT App
- **ID**: The Name of the IOT Device: «corveneto»
- **Type** as that defince in the IOT Device when you created
- The **Time stamp**: “dateObserved” to have a time series data
  - “str” is a string with the date and time in standard ISO, such as ,
  - “2020-08-04T04:00:00+02:00”,
  - “2020-08-03T00:00:00.000Z”
- And the **vector** of “**attributes**”

```
msg = { payload : {
 "id": "corveneto",
 "type": "misura",
 "attributes":[
{ "name": "dateObserved",  "value": str,  "type": "time" },
{ "name": "stato",  "value": “active”,  "type": "string" },
{ "name": "ricoverati_con_sintomi",  "value": 12,  "type": "integer" },
{ "name": "terapia_intensiva",  "value": 34,  "type": "integer" },
{ "name": "totale_ospedalizzati", "value": 34,  "type": "integer" },
{ "name": "isolamento_domiciliare",  "value": 334,  "type": "integer" },
{ "name": "totale_attualmente_positivi",  "value": 12,  "type": "integer" },
{ "name": "nuovi_attualmente_positivi", "value": 33,  "type": "integer" },
{ "name": "dimessi_guariti",  "value": 22222,  "type": "integer" },
{ "name": "deceduti",  "value": 2,  "type": "integer" },
{ "name": "totale_casi",  "value": 2222,  "type": "integer" },
{ "name": "tamponi",  "value": 222222344,  "type": "integer" }
]
}
}
return msg;
```

Multi Series Widget coming from the same IOT Device

- Over on the serie label to highlight
- Click on the serie label to on/ok
- Over on the graph to see the values



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

Andamenti Nazionali e Regionali infezione COVID-19

Sulla base dei dati della protezione civile, elaborazioni DISITLab

Sun 19 Apr 19:21:39

per evidenziare gli andamenti di vostro interesse: eliminare le curve che non interessano selezionandole in legenda.
Alcuni dati in passato non sono pervenuti alla protezione civile

486	nuovi_att_positivi
108257	Attualmente_positivi
23660	deceduti
47055	dimessi_guariti

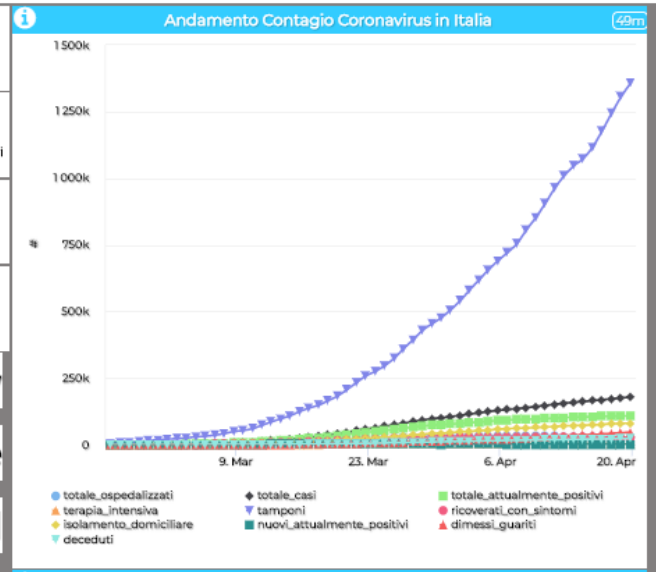
LOG view

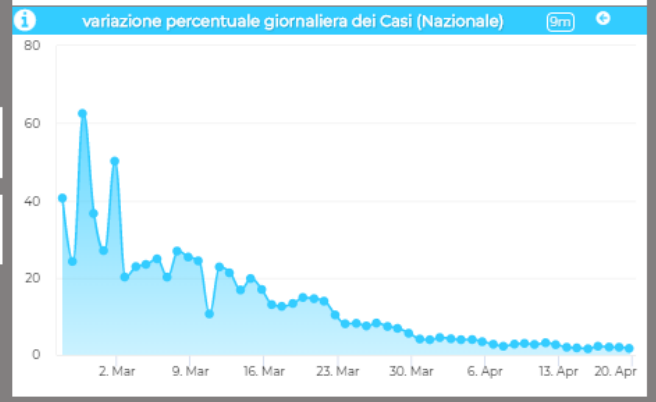
Mobile

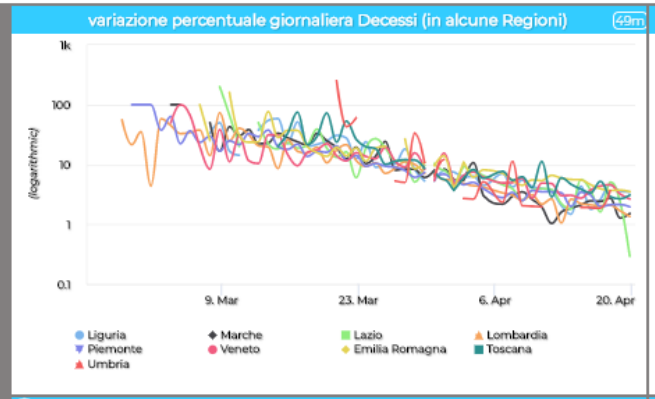
Staked

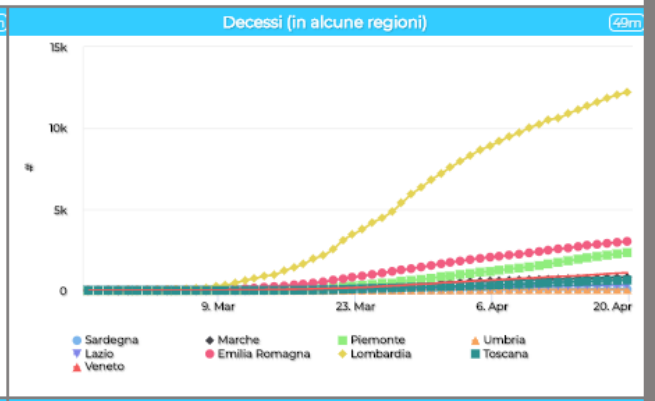
Regione Toscana

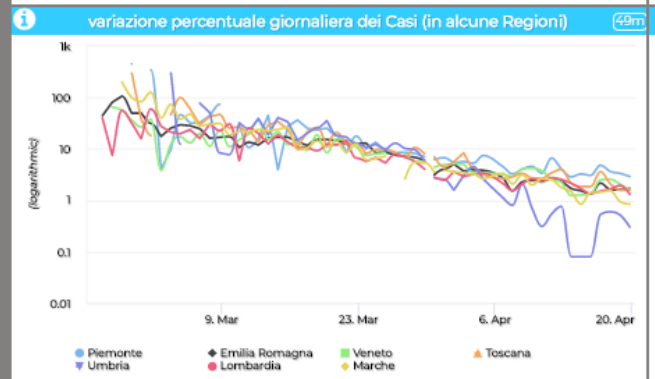
Regione LOG

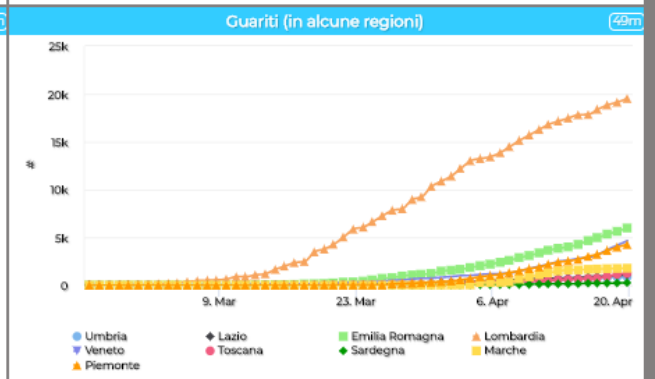


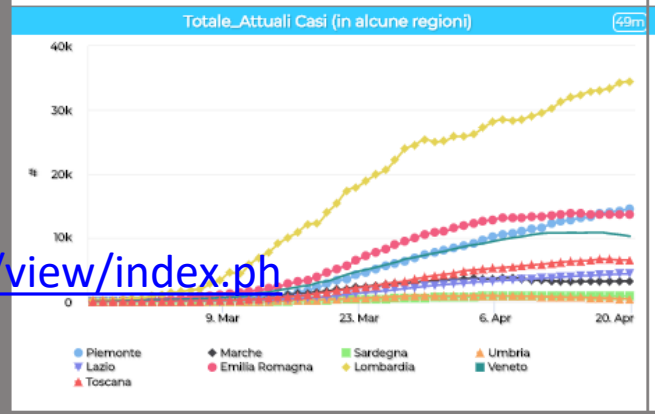


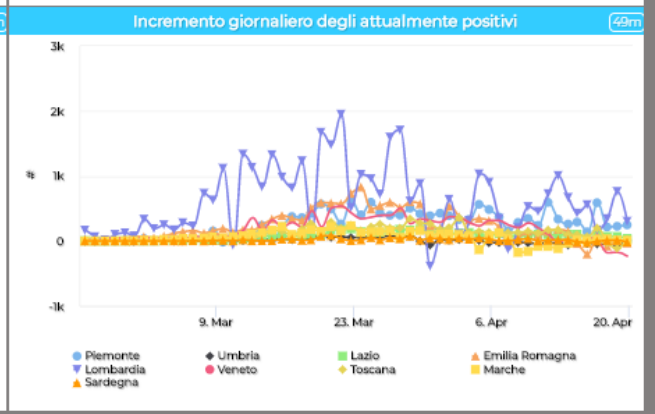










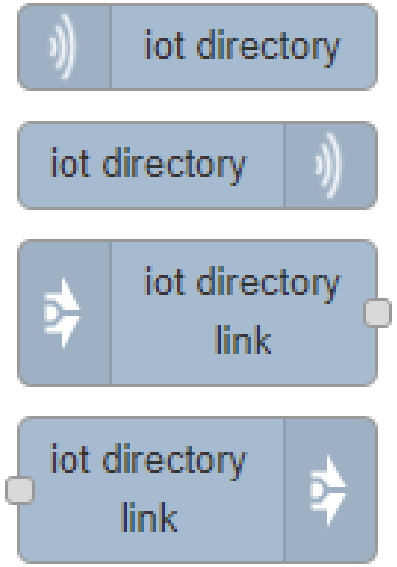


https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjU2OQ==

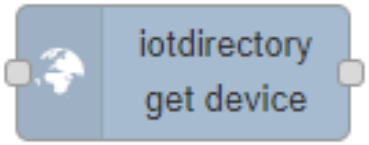
JSON for Authentication as well

- msg.auth= {
- "k1": "1ef0e5e8-yyyy-xxxx-9462-0aa4cfcf5e19",
- "k2": "b2b34425-yyyy-xxxx-818d-2d6cac2314a6",
- "apikey":"apikey",
- "basicAuth": "basicAuthKey"
- };

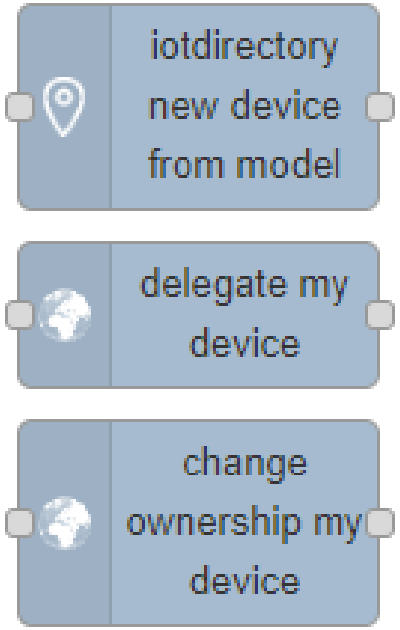
You may use other functions from IOT Directory



- IOT Discovery in an area
- Query on IOT Directory

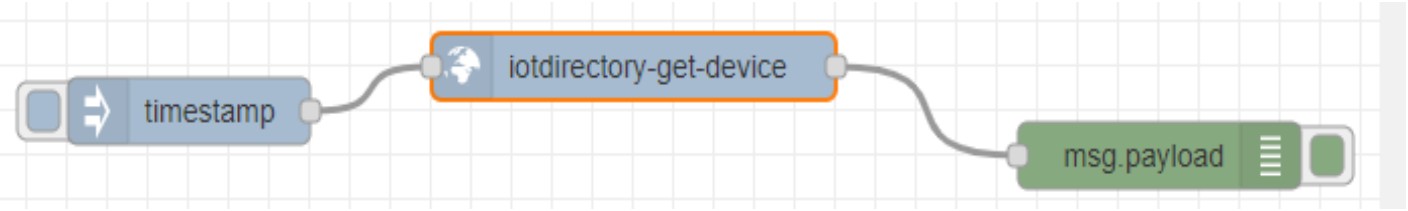


- Get IOT Device Info



- Registering an IOT Device from model
- Delegate an IOT Device
- Change Ownerhip of an IOT Device

Get IOT Device Info

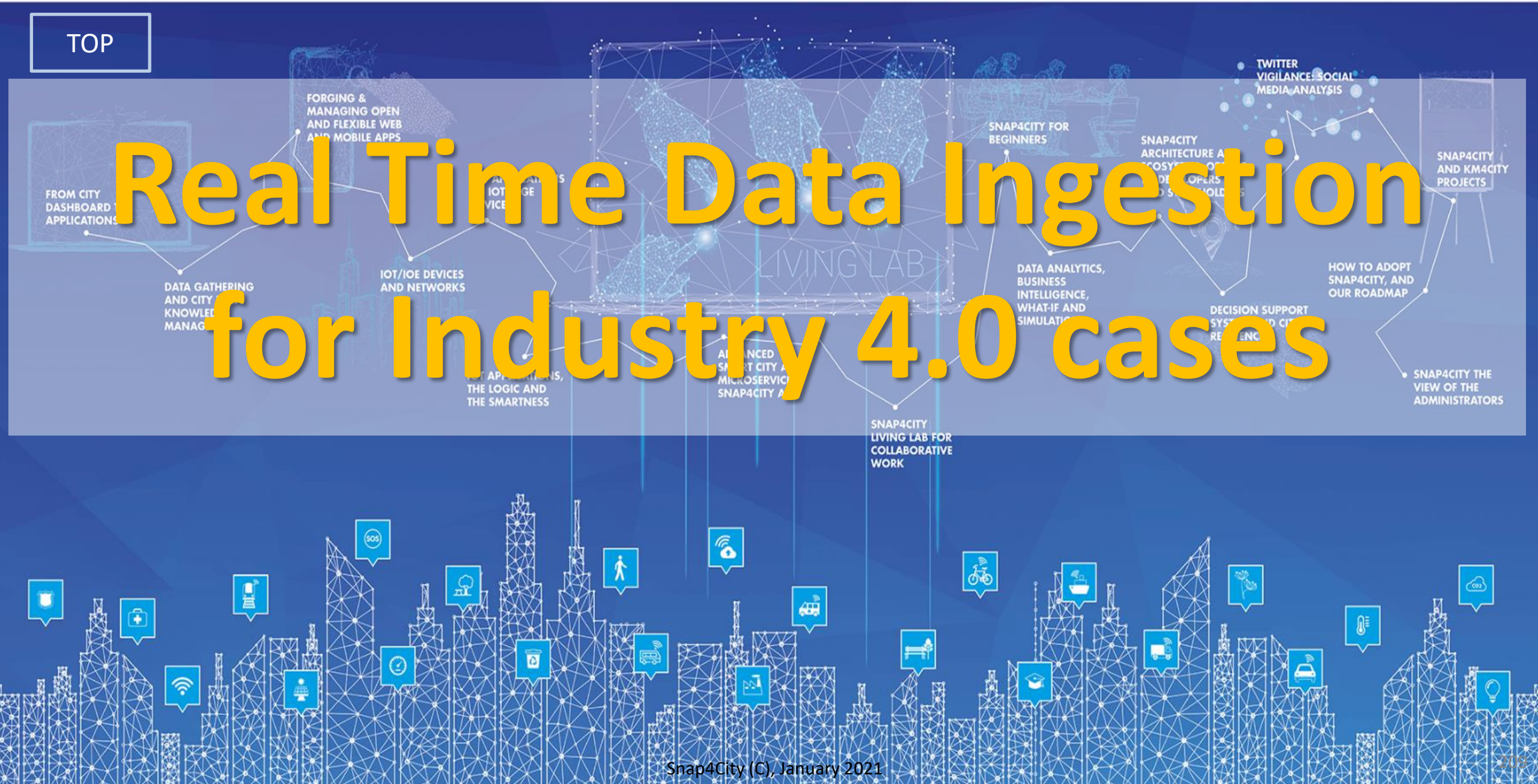


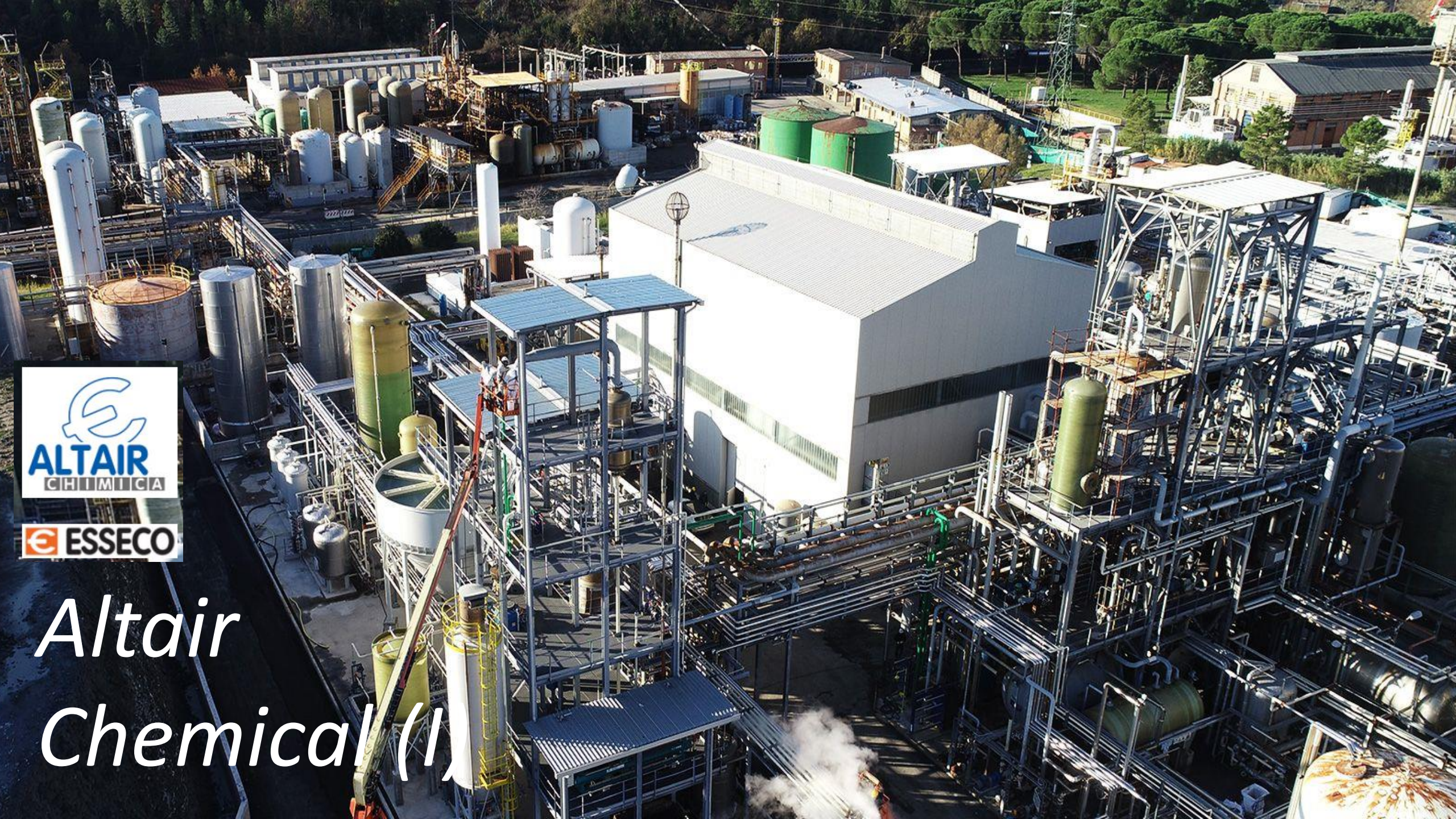
- You can create smart IOT Applications that on t he basis of the list of IOT Devices would request all what you need to load data into **YOUR OWN** IOT Devices including:
  - **Service URI**
  - **K1, K2**
  - Authentication

msg.payload : Object
object
status: "ok"
content: object
uri: "http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/corarezzo"
devicetype: "misura"
kind: "sensor"
status1: "active"
macaddress: ""
model: "statuscorregione"
producer: "protezione civile"
longitude: "11.88228"
latitude: "43.46642"
protocol: "ngsi"
format: "json"
visibility: "public"
frequency: "600"
created: "2020-03-21 18:34:32"
privatekey: ""
certificate: ""
organization: "DISIT"
accesslink: "https://broker1.snap4city.org"
accessport: "8080"
sha: "C61E32DBFAE7F14C0810177F2D2300843C41C550"
subnature: null
static_attributes: null
k1: "bf739214-f6b4-45fe-85f7-97cd09fe8e57"
k2: "c19e0b6f-8f98-4130-b135-e7a1dfae9273"
error_msg: ""

TOP

Real Time Data Ingestion for Industry 4.0 cases





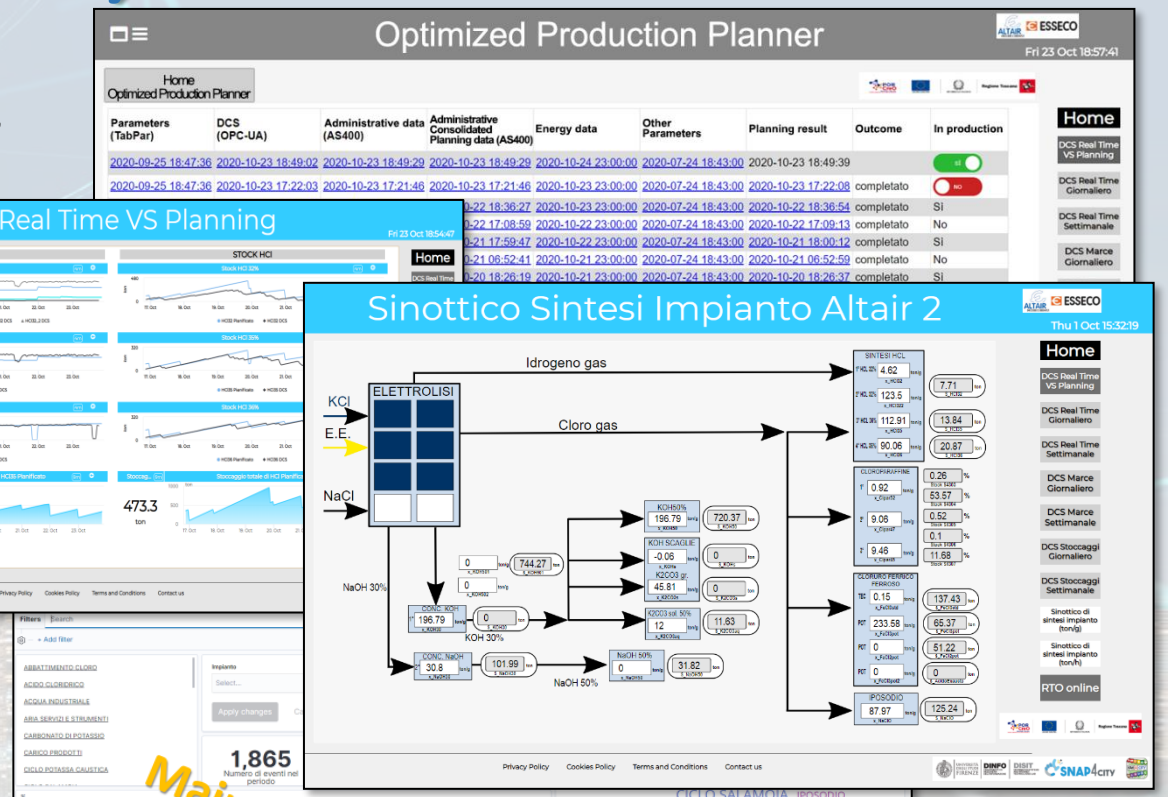
Altair Chemical (I)

Snap4Altair Decision Support supervision and control, Industry 4.0

reference

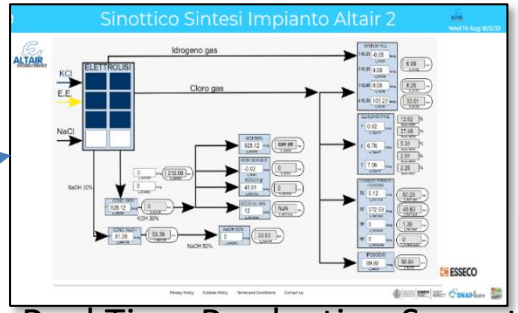
- **Multiple Domain Data**
  - Distributed Control System: energy, flows, storage, chemical data, settings, ..
  - Cost of energy
  - Orders
  - Production Parameters
  - Maintenance data
- **Multiple Levels & Decision Makers**
- **Historical and Real Time data**
  - Billions of Data
  - Optimized planning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







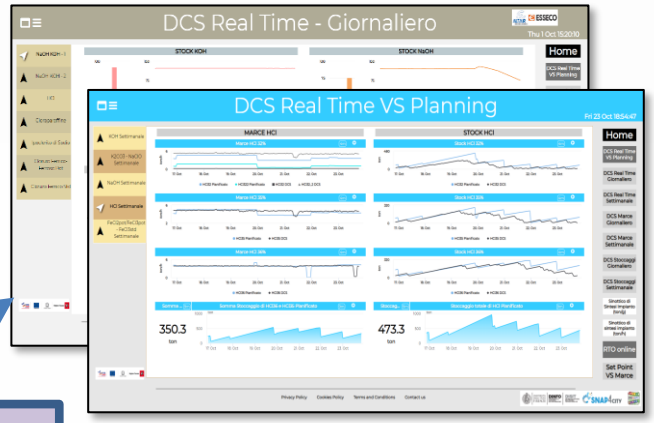
Plant Status



Real Time Production Synoptic

Production Plan

Production vs Planning



Data Storage

Decision Support

Data Ingestion



AS400

Orders

Energy Service

Transportation

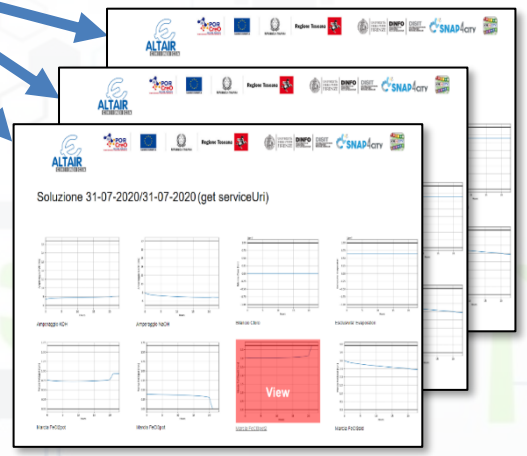
Other Costs

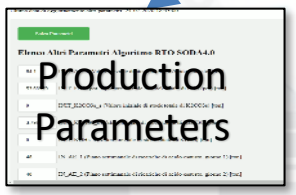
Production Plant Management

Optimized Production Planner

Possible Plan

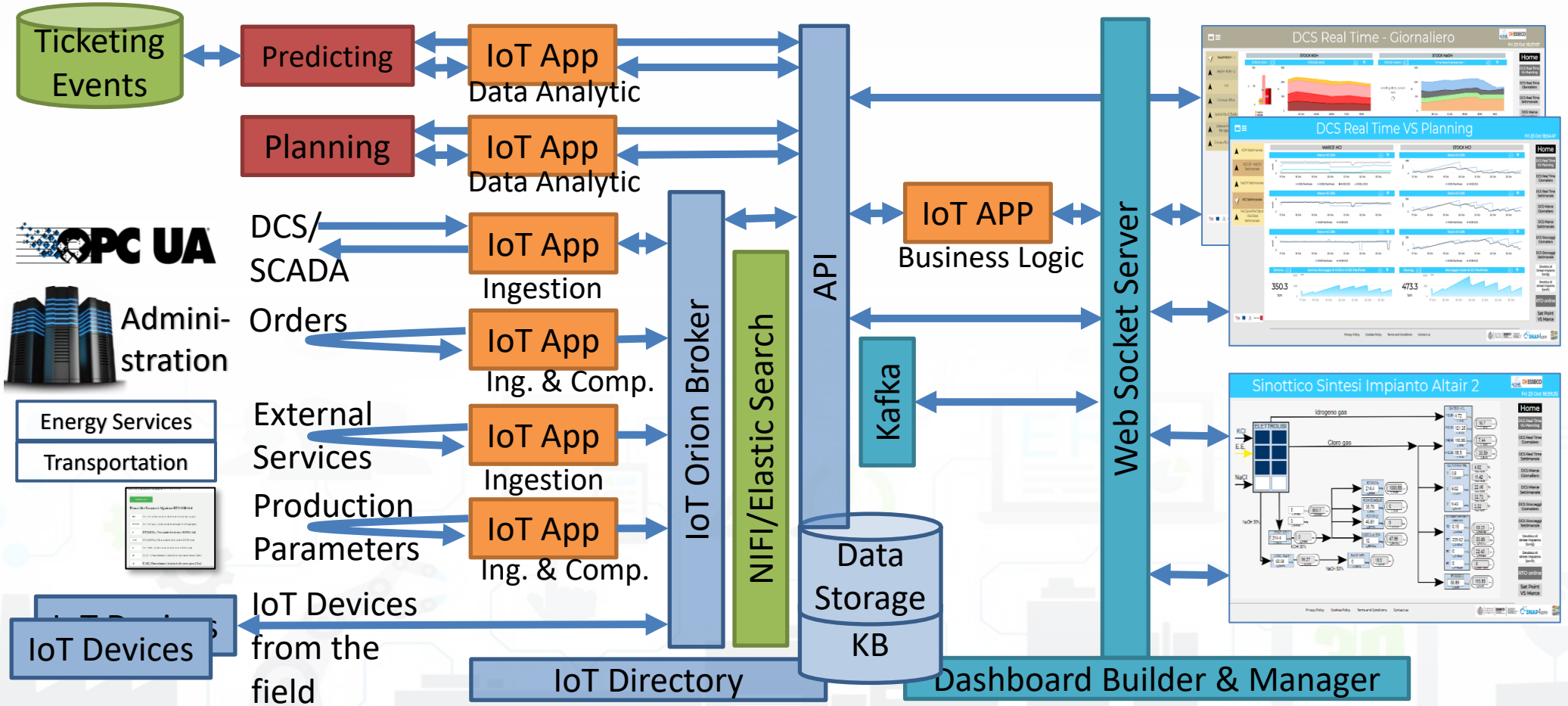
Possible Plans

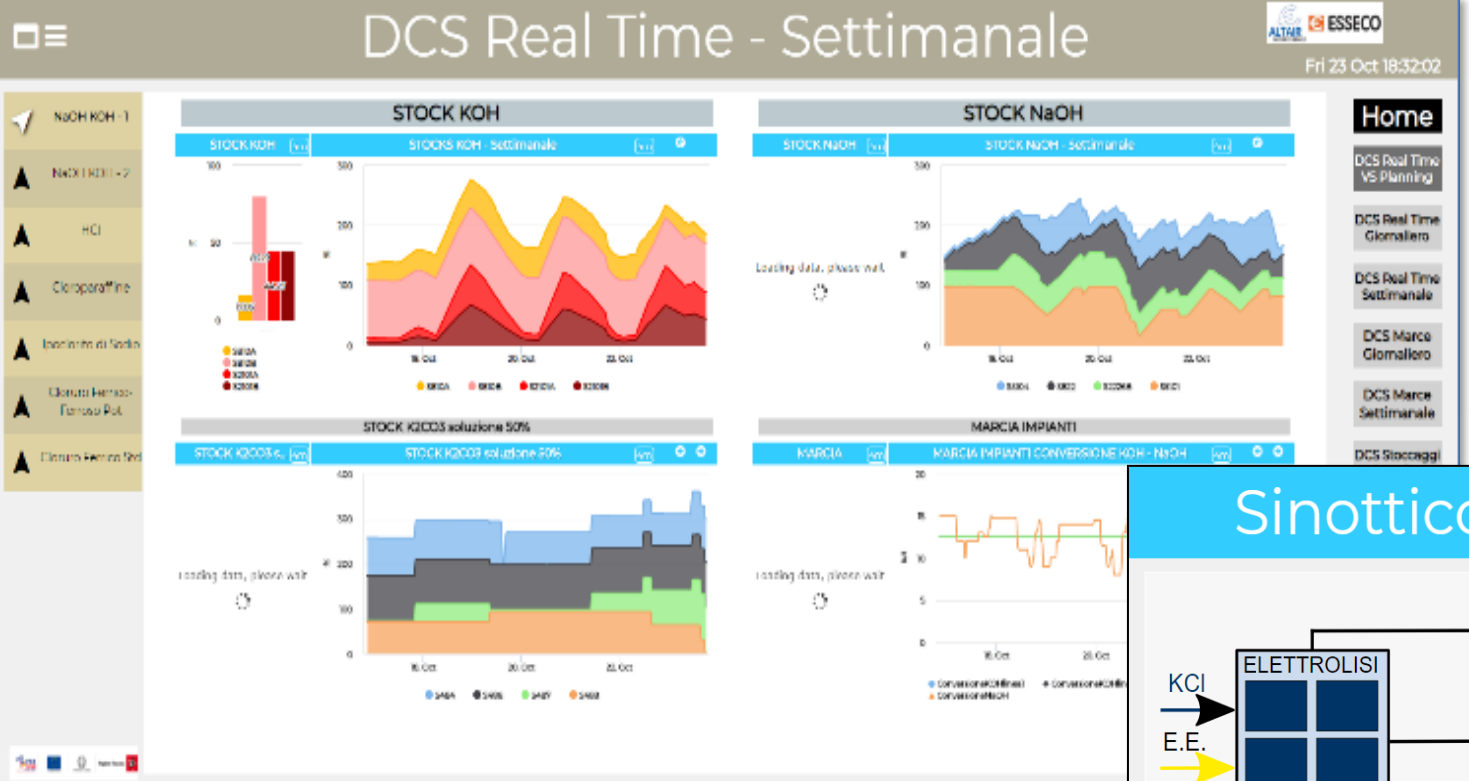


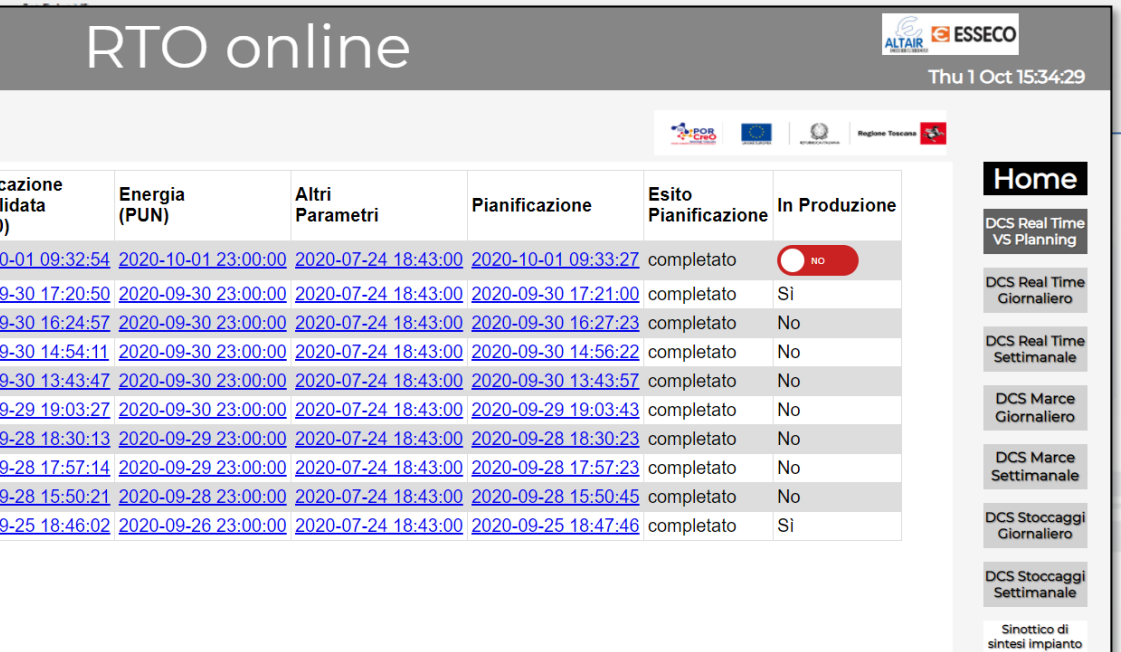




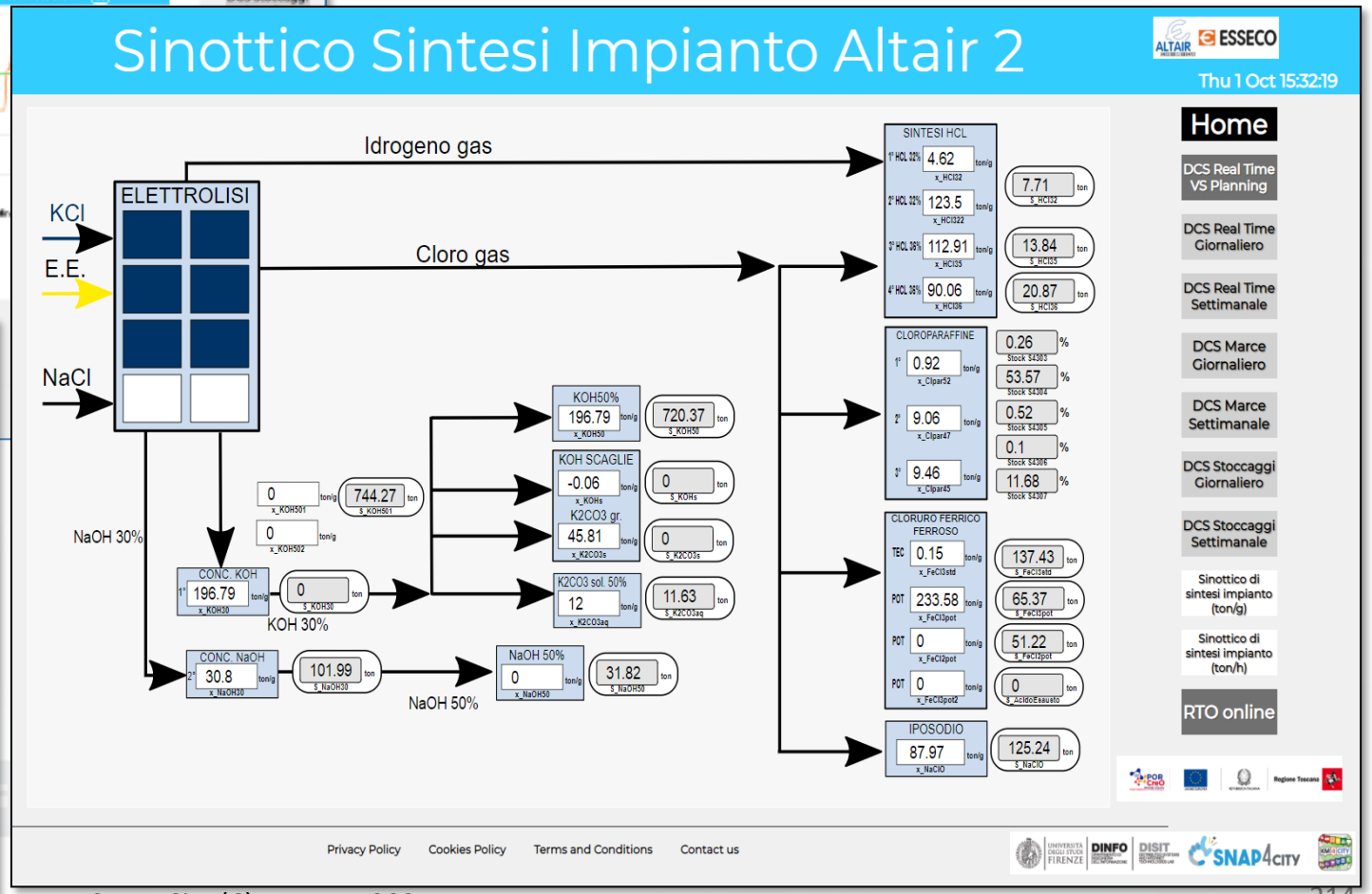
Plant Management


Ticketing Events
Predicting
IoT App
Data Analytic
Planning
IoT App
Data Analytic
OPC UA
DCS/ SCADA
IoT App
Ingestion
Admini-stration
Orders
IoT App
Ing. & Comp.
Energy Services
Transportation
External Services
IoT App
Ingestion
Production Parameters
IoT App
Ing. & Comp.
IoT Devices
IoT Devices from the field
IoT Orion Broker
NIFI/Elastic Search
API
IoT APP
Business Logic
Kafka
Data Storage
KB
IoT Directory
Web Socket Server
Dashboard Builder & Manager
DCS Real Time - Giornaliero
DCS Real Time VS Planning
Sinottico Sintesi Impianto Altair 2





cazione lidata)	Energia (PUN)	Altri Parametri	Pianificazione	Esito Pianificazione	In Produzione
0-01 09:32:54	2020-10-01 23:00:00	2020-07-24 18:43:00	2020-10-01 09:33:27	completato	NO
9-30 17:20:50	2020-09-30 23:00:00	2020-07-24 18:43:00	2020-09-30 17:21:00	completato	Sì
9-30 16:24:57	2020-09-30 23:00:00	2020-07-24 18:43:00	2020-09-30 16:27:23	completato	No
9-30 14:54:11	2020-09-30 23:00:00	2020-07-24 18:43:00	2020-09-30 14:56:22	completato	No
9-30 13:43:47	2020-09-30 23:00:00	2020-07-24 18:43:00	2020-09-30 13:43:57	completato	No
9-29 19:03:27	2020-09-30 23:00:00	2020-07-24 18:43:00	2020-09-29 19:03:43	completato	No
9-28 18:30:13	2020-09-29 23:00:00	2020-07-24 18:43:00	2020-09-28 18:30:23	completato	No
9-28 17:57:14	2020-09-29 23:00:00	2020-07-24 18:43:00	2020-09-28 17:57:23	completato	No
9-28 15:50:21	2020-09-28 23:00:00	2020-07-24 18:43:00	2020-09-28 15:50:45	completato	No
9-25 18:46:02	2020-09-26 23:00:00	2020-07-24 18:43:00	2020-09-25 18:47:46	completato	Sì



Optimized Production Planner

ESSECO ALTAIR

Fri 23 Oct 18:57:41

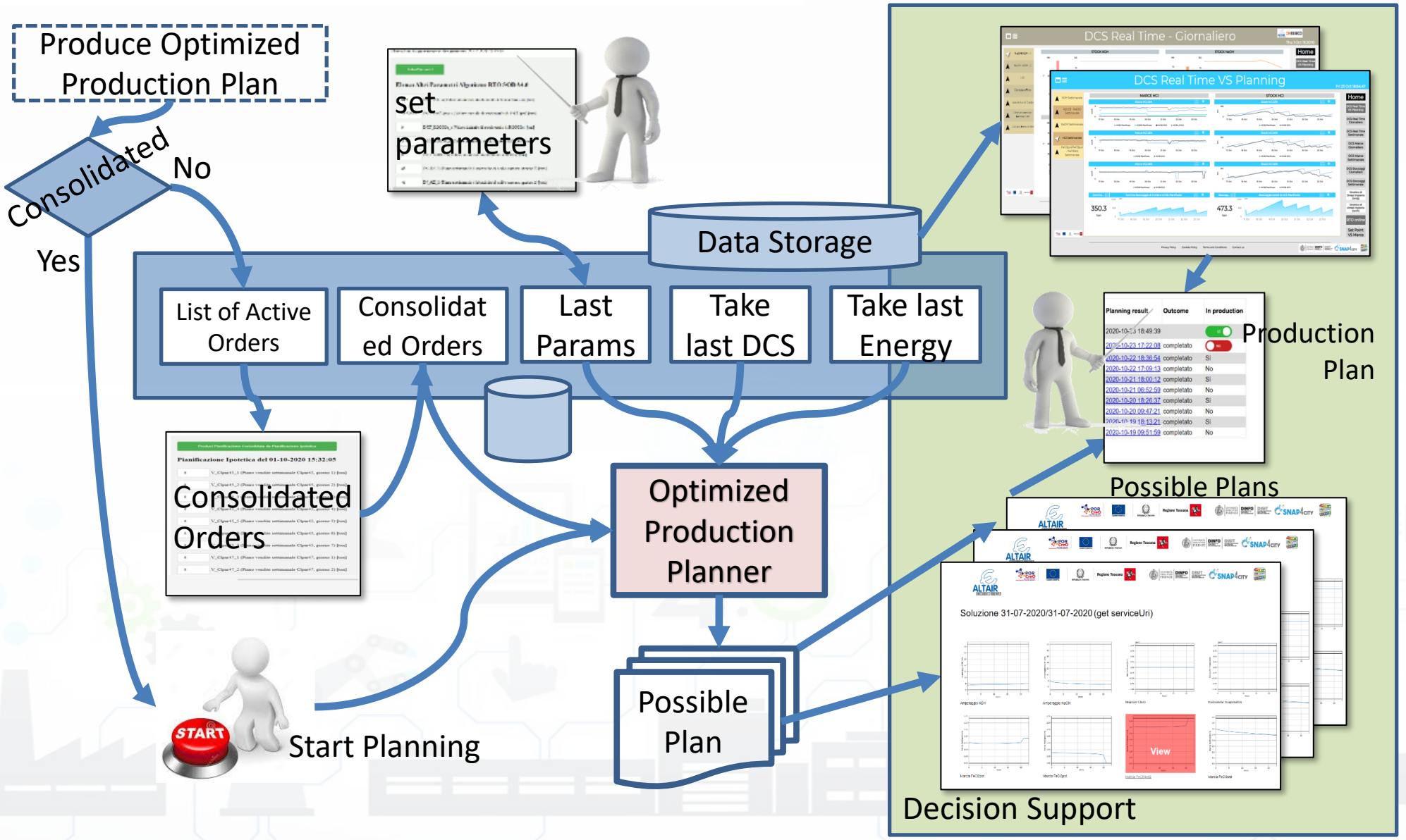
Home
Optimized Production Planner

Parameters (TabPar)	DCS (OPC-UA)	Administrative data (AS400)	Administrative Consolidated Planning data (AS400)	Energy data	Other Parameters	Planning result	Outcome	In production
2020-09-25 18:47:36	2020-10-23 18:49:02	2020-10-23 18:49:29	2020-10-23 18:49:29	2020-10-24 23:00:00	2020-07-24 18:43:00	2020-10-23 18:49:39		SÌ
2020-09-25 18:47:36	2020-10-23 17:22:03	2020-10-23 17:21:46	2020-10-23 17:21:46	2020-10-23 23:00:00	2020-07-24 18:43:00	2020-10-23 17:22:08	completato	NO
2020-09-25 18:47:36	2020-10-22 18:36:02	2020-10-22 18:36:27	2020-10-22 18:36:27	2020-10-23 23:00:00	2020-07-24 18:43:00	2020-10-22 18:36:54	completato	Sì
2020-09-25 18:47:36	2020-10-22 17:09:02	2020-10-22 17:08:59	2020-10-22 17:08:59	2020-10-22 23:00:00	2020-07-24 18:43:00	2020-10-22 17:09:13	completato	No
2020-09-25 18:47:36	2020-10-21 18:00:02	2020-10-21 17:59:47	2020-10-21 17:59:47	2020-10-22 23:00:00	2020-07-24 18:43:00	2020-10-21 18:00:12	completato	Sì
2020-09-25 18:47:36	2020-10-21 06:52:02	2020-10-21 06:52:41	2020-10-21 06:52:41	2020-10-21 23:00:00	2020-07-24 18:43:00	2020-10-21 06:52:59	completato	No
2020-09-25 18:47:36	2020-10-20 18:26:02	2020-10-20 18:26:19	2020-10-20 18:26:19	2020-10-21 23:00:00	2020-07-24 18:43:00	2020-10-20 18:26:37	completato	Sì
2020-09-25 18:47:36	2020-10-20 09:47:03	2020-10-20 09:47:05	2020-10-20 09:47:05	2020-10-20 23:00:00	2020-07-24 18:43:00	2020-10-20 09:47:21	completato	No
2020-09-25 18:47:36	2020-10-19 18:13:02	2020-10-19 18:13:09	2020-10-19 18:13:09	2020-10-20 23:00:00	2020-07-24 18:43:00	2020-10-19 18:13:21	completato	Sì
2020-09-25 18:47:36	2020-10-19 09:51:02	2020-10-19 09:51:08	2020-10-19 09:51:08	2020-10-19 23:00:00	2020-07-24 18:43:00	2020-10-19 09:51:59	completato	No

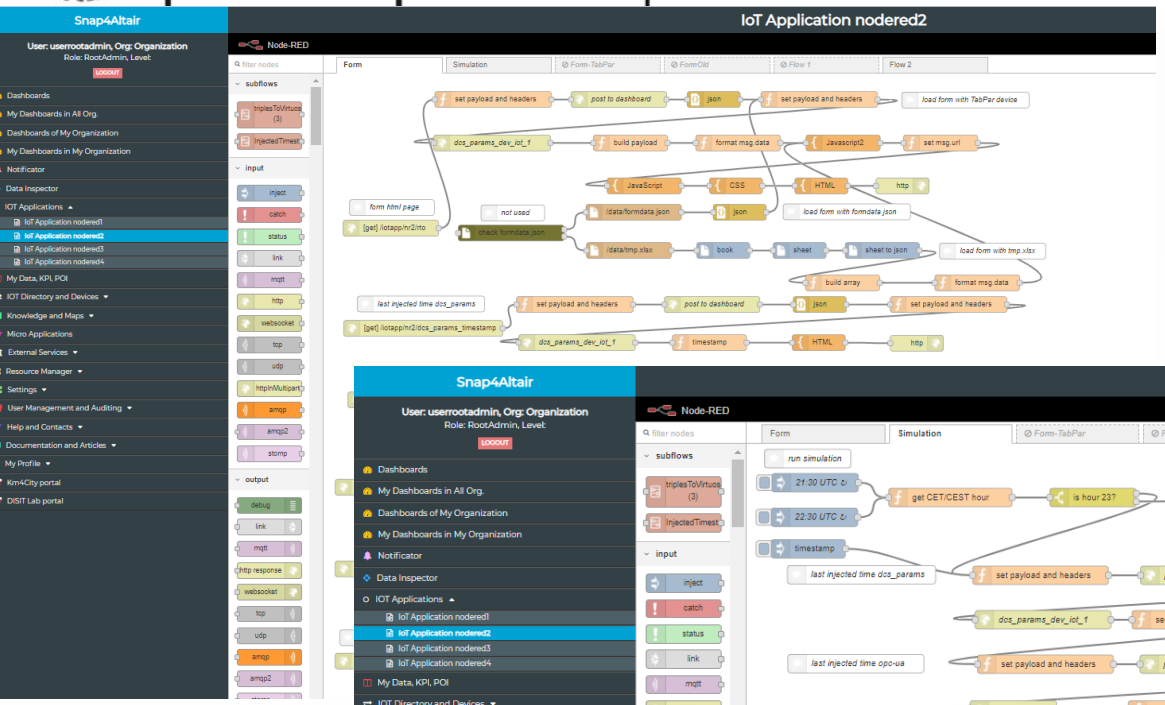
<< 1 2 3 4 5 6 7 8 9 10 11 12 13 14 >>

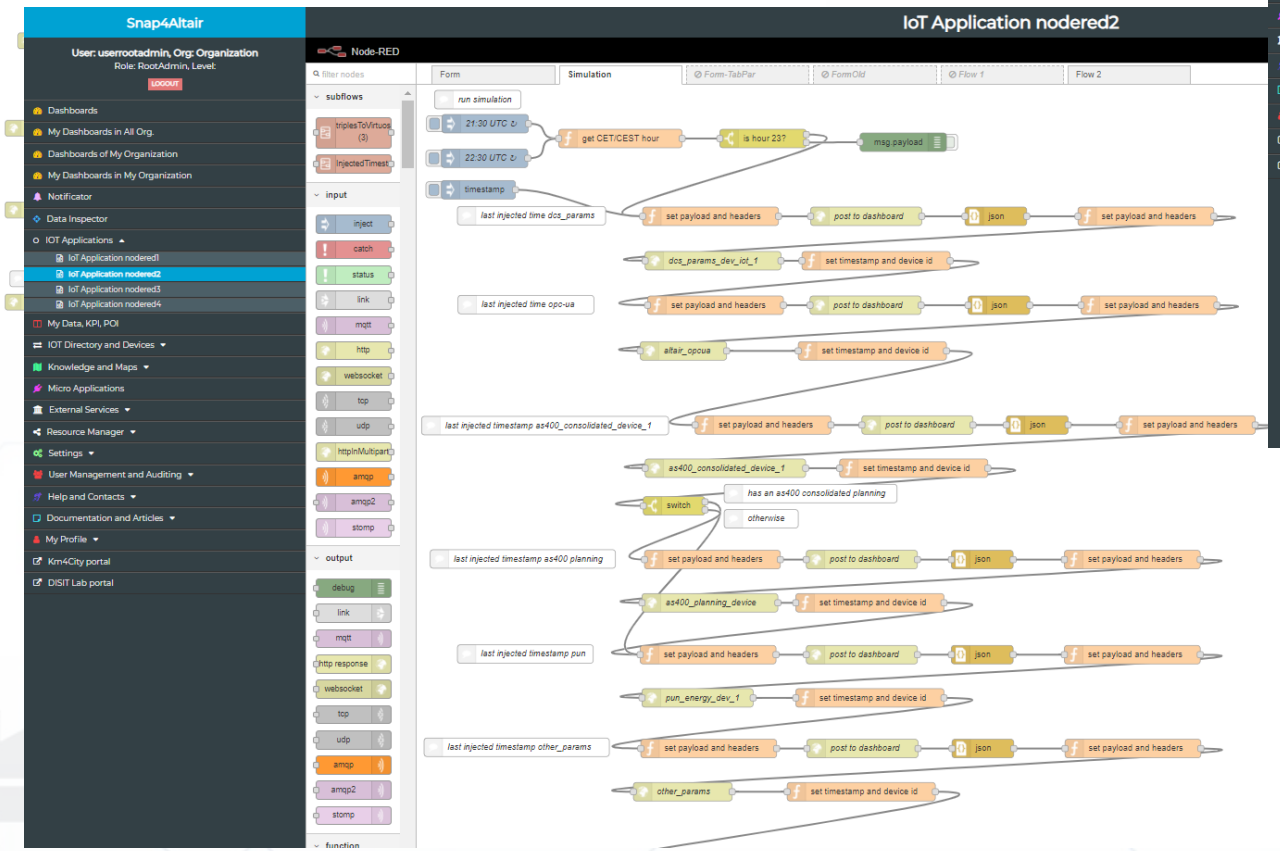
Home
DCS Real Time VS Planning
DCS Real Time Giornaliero
DCS Real Time Settimanale
DCS Marce Giornaliero
DCS Marce Settimanale
DCS Stoccaggi Giornaliero
DCS Stoccaggi Settimanale

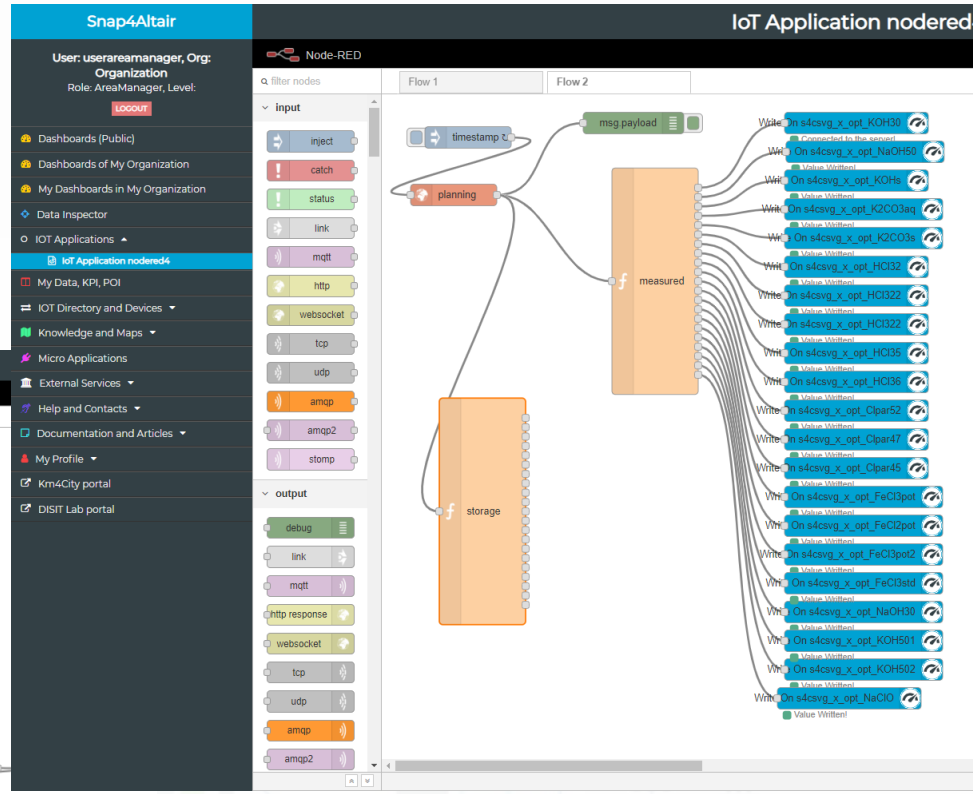
Business Logic



Some Flows

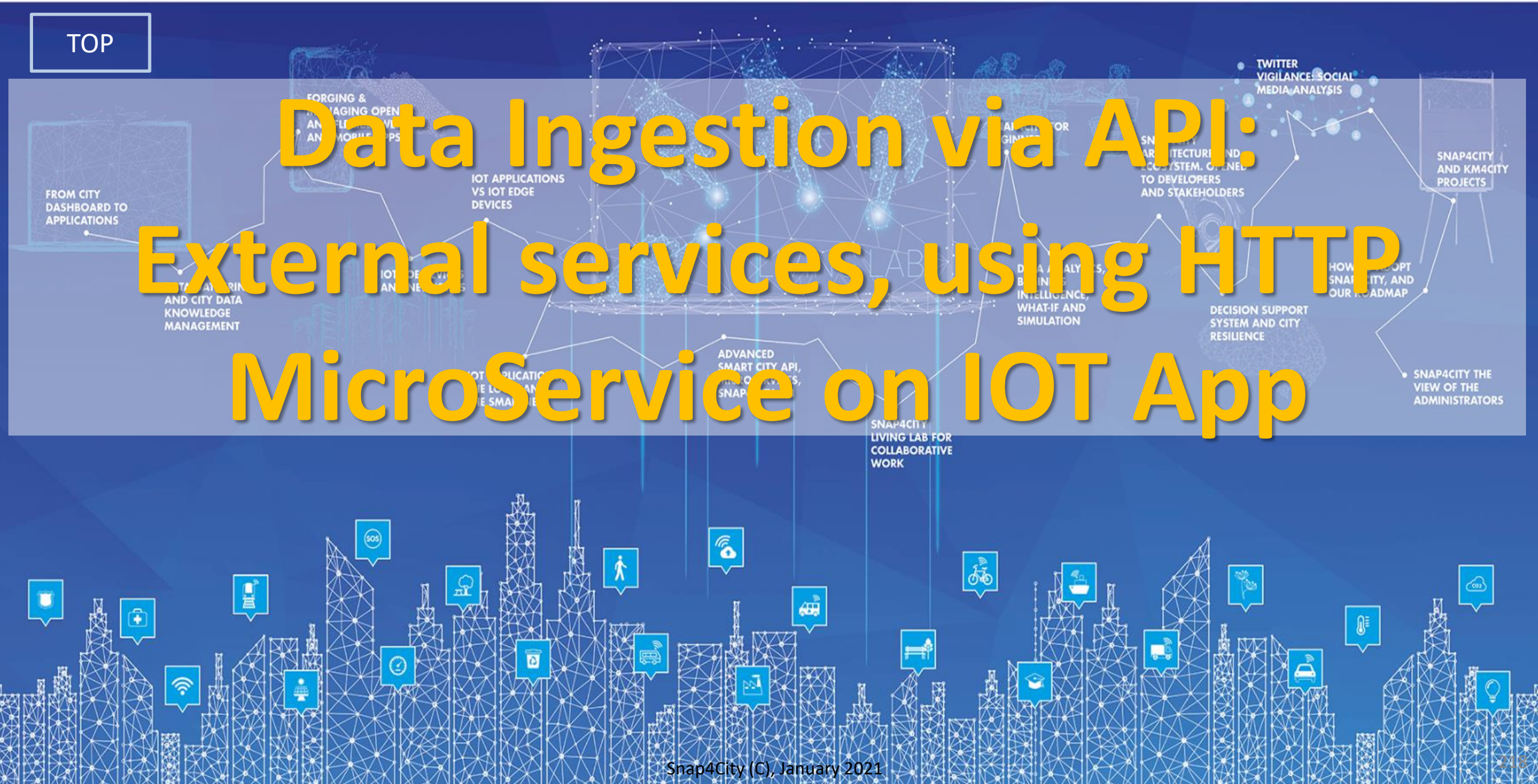

Snap4Altair
IoT Application nodered2
Node-RED


Snap4Altair
IoT Application nodered2
Node-RED


Snap4Altair
IoT Application nodered
Node-RED

TOP

Data Ingestion via API: External services, using HTTP MicroService on IOT App



General solution, bring data from API to Dashboards

- You can use the MicroServices HTTP in get and post to act on REST Call

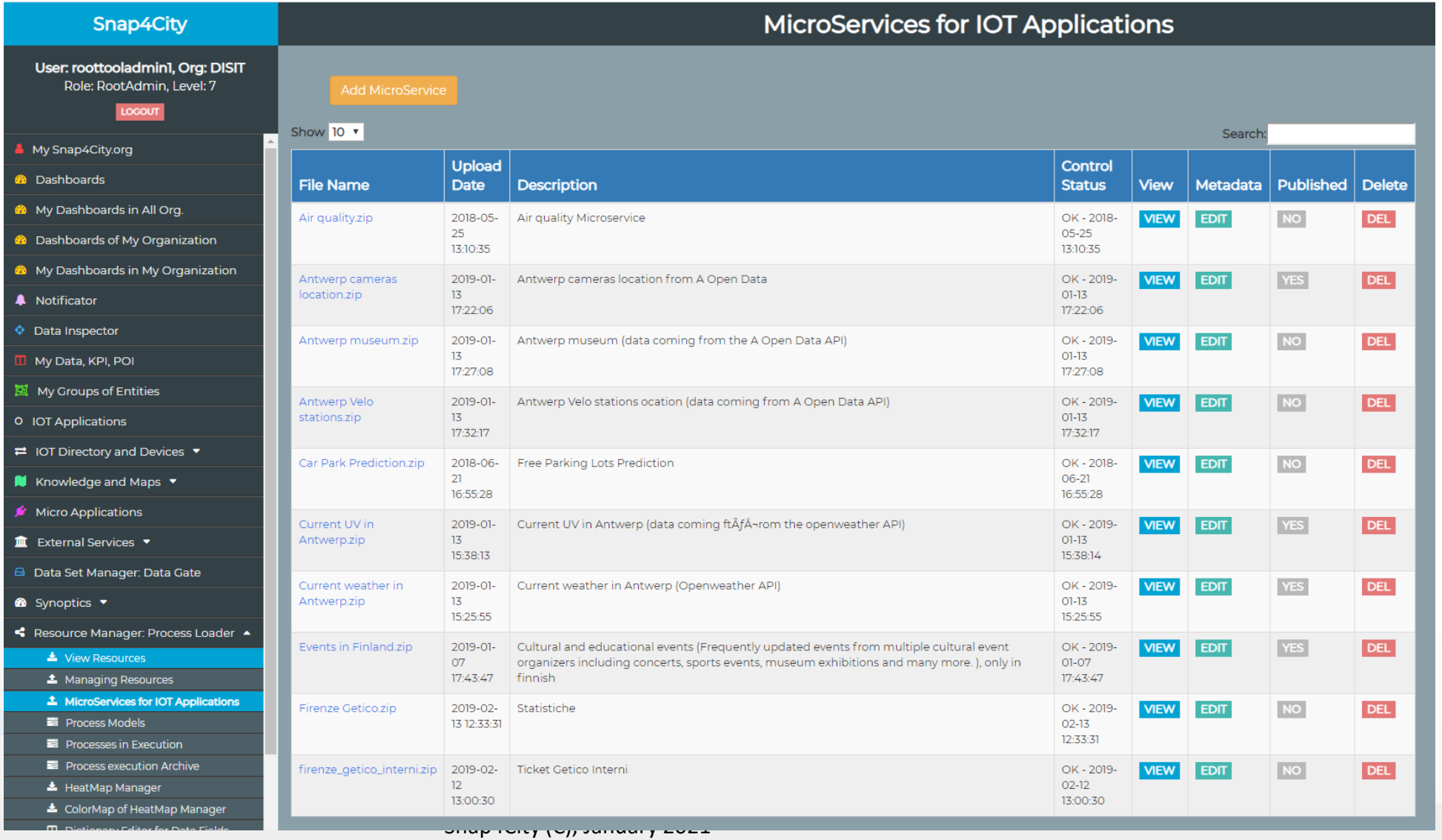
- OR

- You can **save/consolidate your rest API** transforming it in a MicroService usable for many colleagues into IOT Applications:
  - TC2.25- Registering external MicroService calling RestCall services, using it on IOT applications https://www.snap4city.org/129

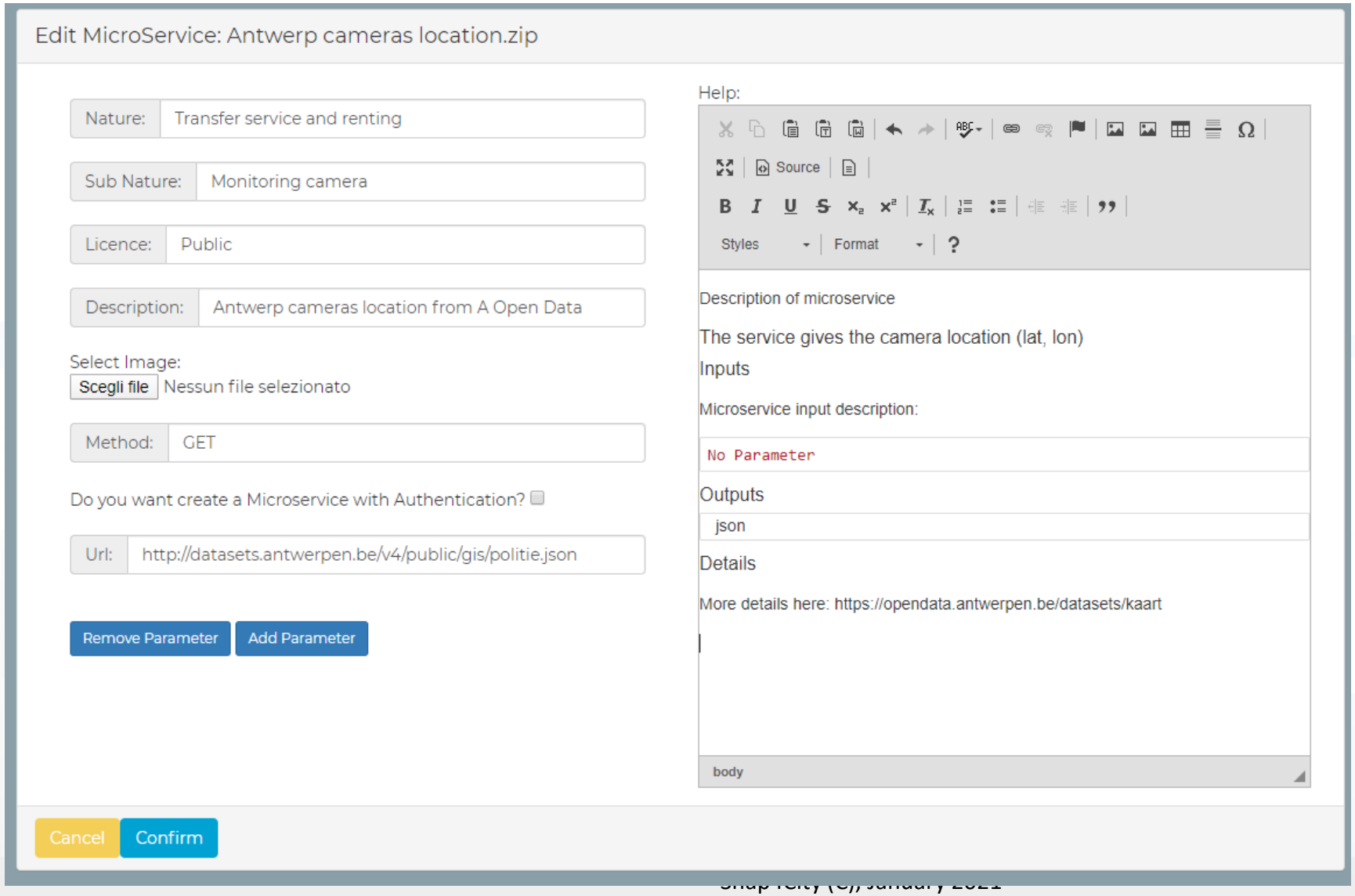
IF your REST API is going to use credentials as username and password, we suggest you to save them into MyPersonalData of Snap4City
  - so that the code will not provide clear credentials and you can update from user interface on your personal data profile.
  - The IOT App can retrieve the Username and Password at the moment in which they are used with the security shield of Snap4City

External REST Call API vs MicroServices

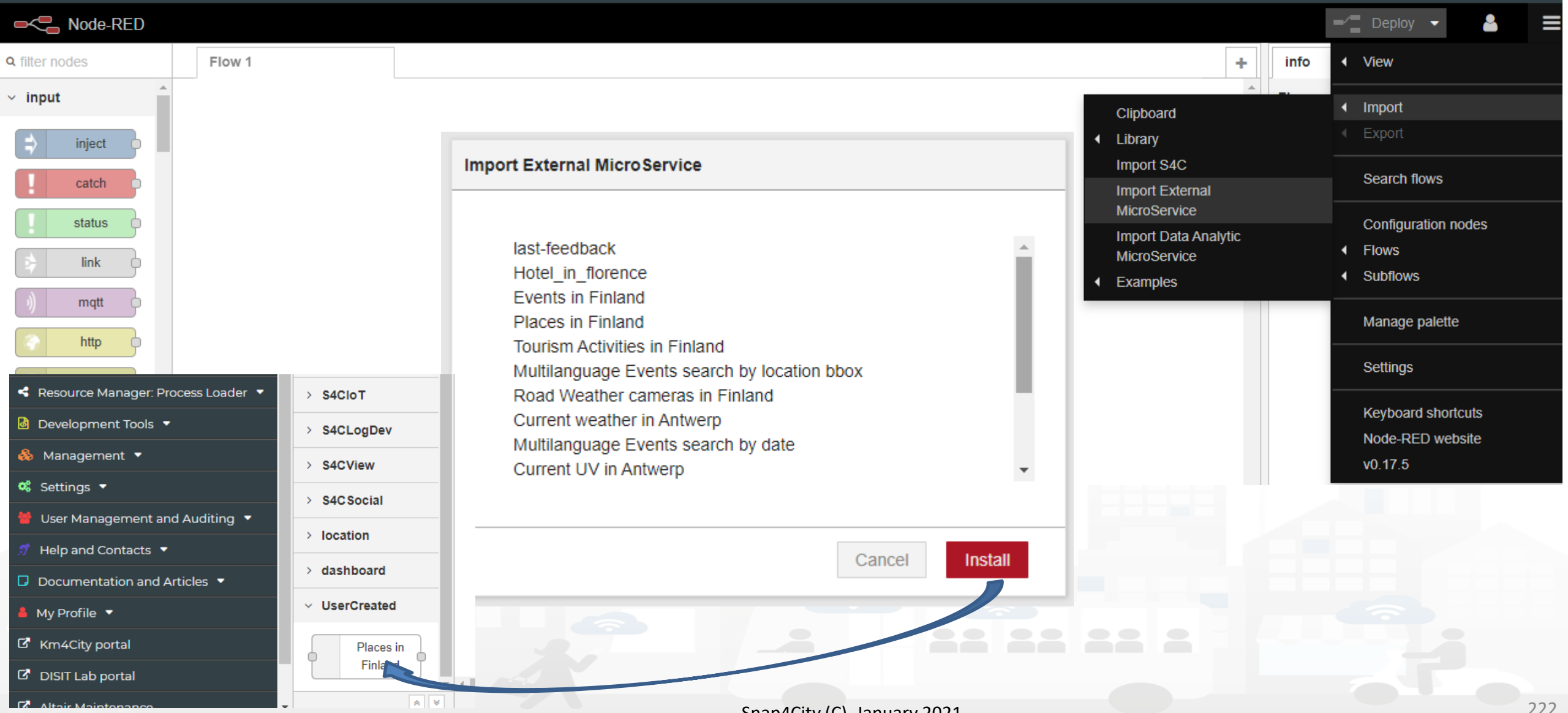
- Each Rest Call API can be automaticaly transformed into e MicroService for the IOT Applications



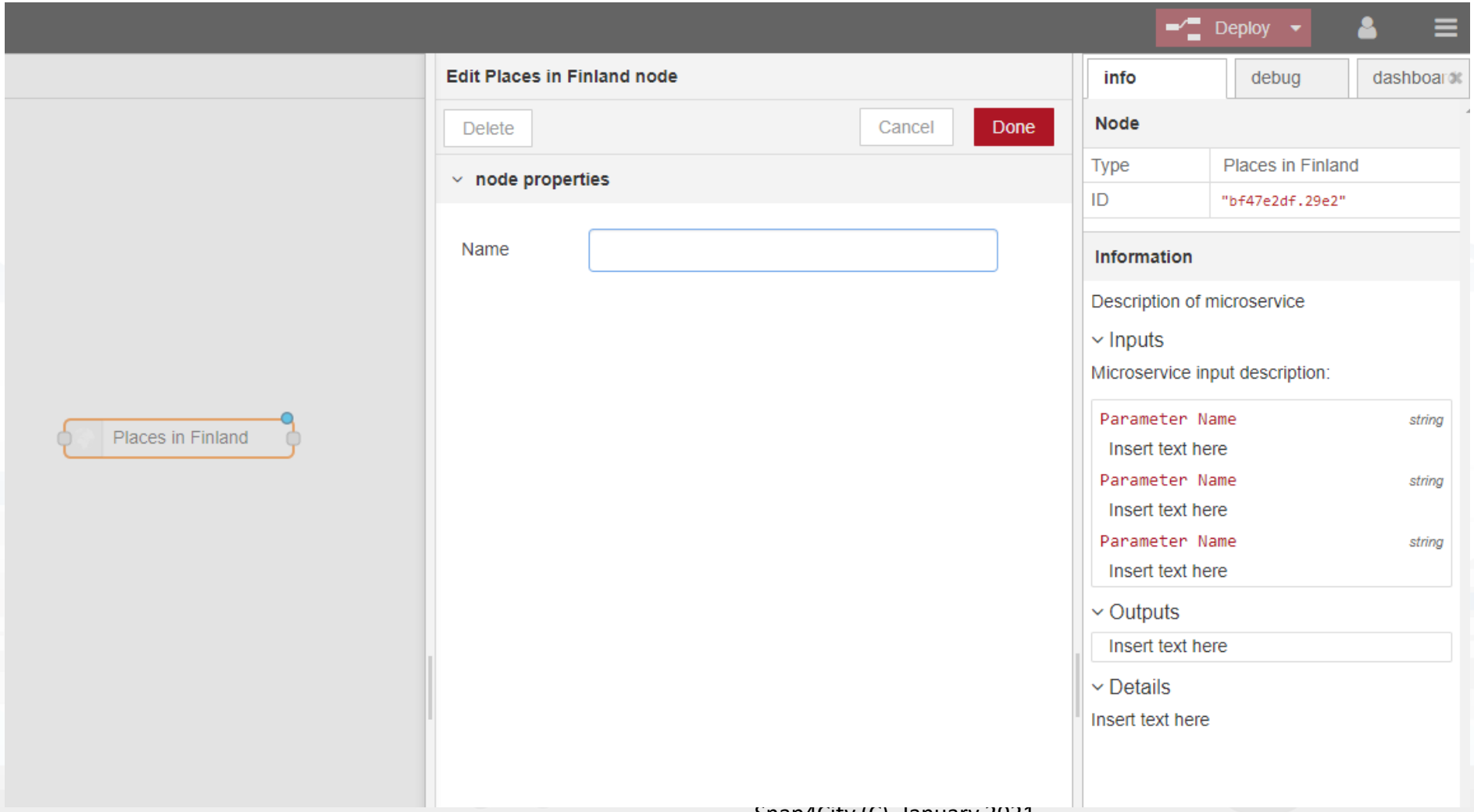
Edit MicroService Call: call and help editing



a

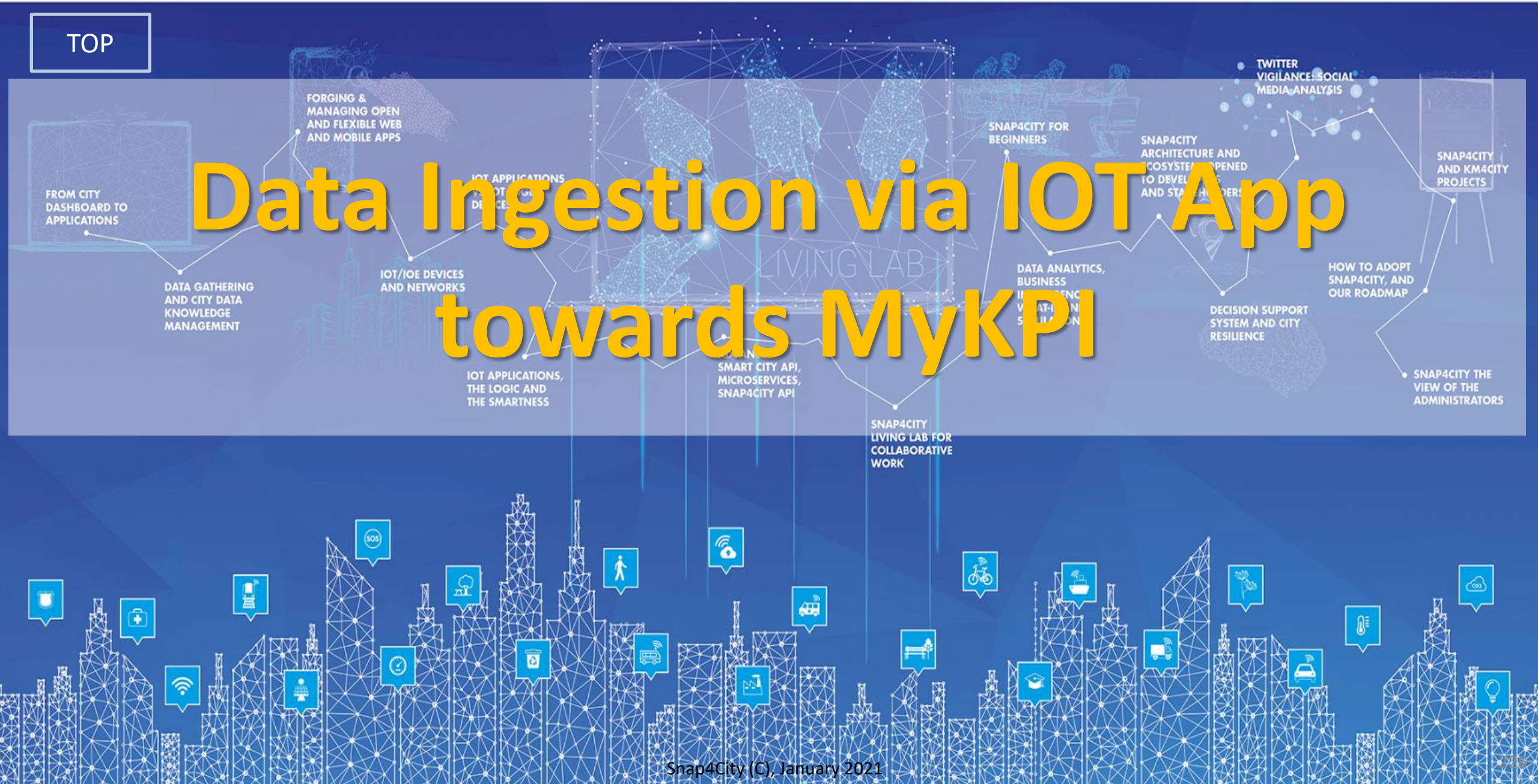

Node-RED
Deploy
filter nodes
Flow 1
info
input
inject
catch
status
link
mqtt
http
View
Import
Export
Search flows
Configuration nodes
Flows
Subflows
Manage palette
Settings
Keyboard shortcuts
Node-RED website
v0.17.5
Clipboard
Library
Import S4C
Import External MicroService
Import Data Analytic MicroService
Examples
Import External MicroService
last-feedback
Hotel_in_florence
Events in Finland
Places in Finland
Tourism Activities in Finland
Multilanguage Events search by location bbox
Road Weather cameras in Finland
Current weather in Antwerp
Multilanguage Events search by date
Current UV in Antwerp
Cancel
Install
Resource Manager: Process Loader
Development Tools
Management
Settings
User Management and Auditing
Help and Contacts
Documentation and Articles
My Profile
Km4City portal
DISIT Lab portal
S4CIoT
S4CLogDev
S4CView
S4CSocial
location
dashboard
UserCreated
Places in Finland

Usage of the MicroService from IOT App

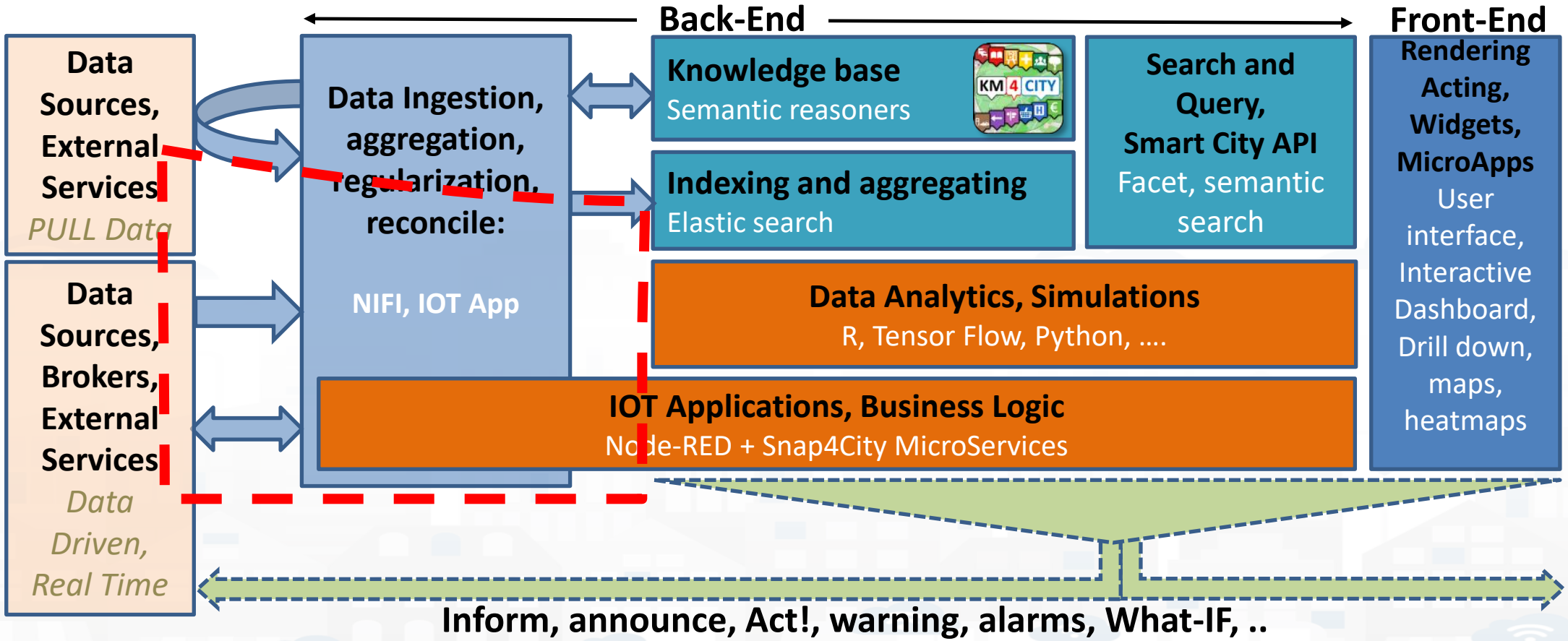


TOP

Data Ingestion via IOT App towards MyKPI



Snap4City Architecture vs Data Ingestion

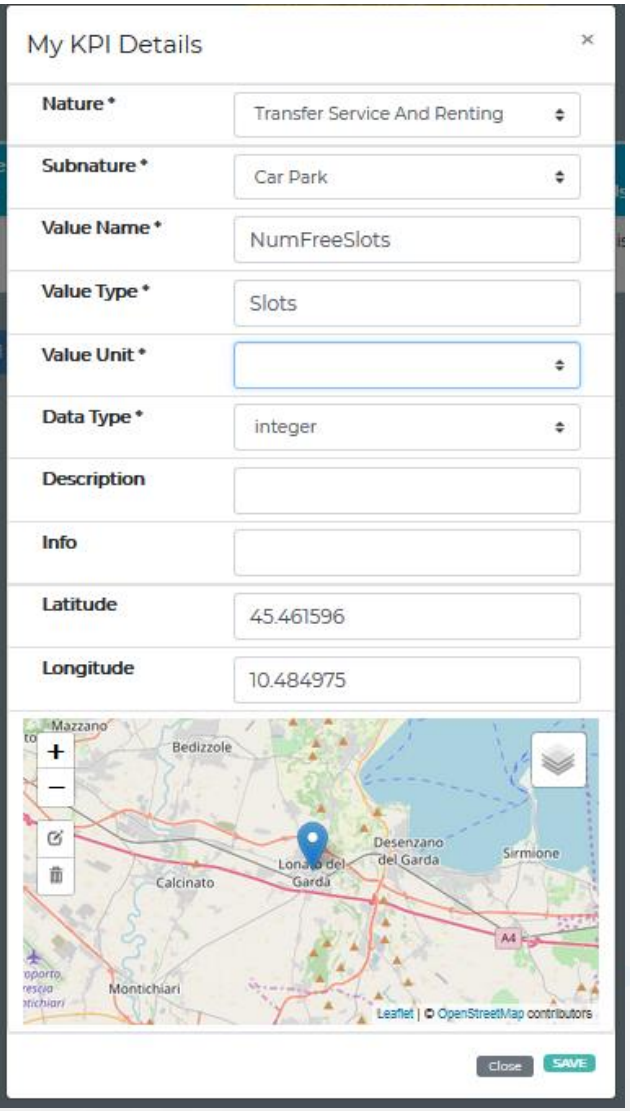


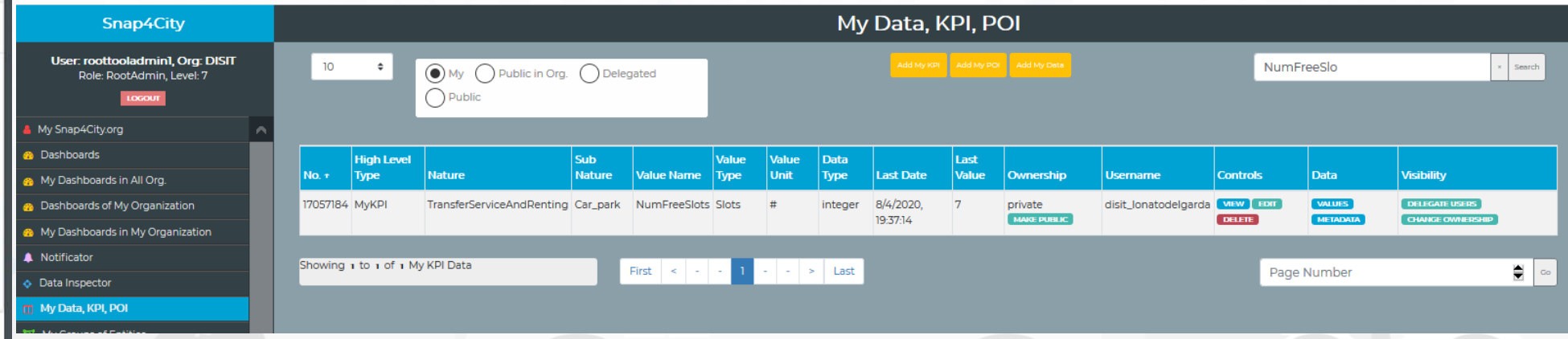
MyKPI

- **MyKPI can**
  - be time series + metadata
  - be POI with full metadata decryption, MyPOI
  - be passed in ownership to other users,
  - be delegated in access to other users
  - model daily trajectories from: Mobile Phone Apps, CANBUS data and GPS location from mobiles, PAX Counter Mobile, mobile IOT Devices, etc.
  - be saved and retrieved from IOT Apps
  - create events at their changes towards IOT Apps
  - be saved into: MySQL and/or Elastic Search

Create your MyKPIs

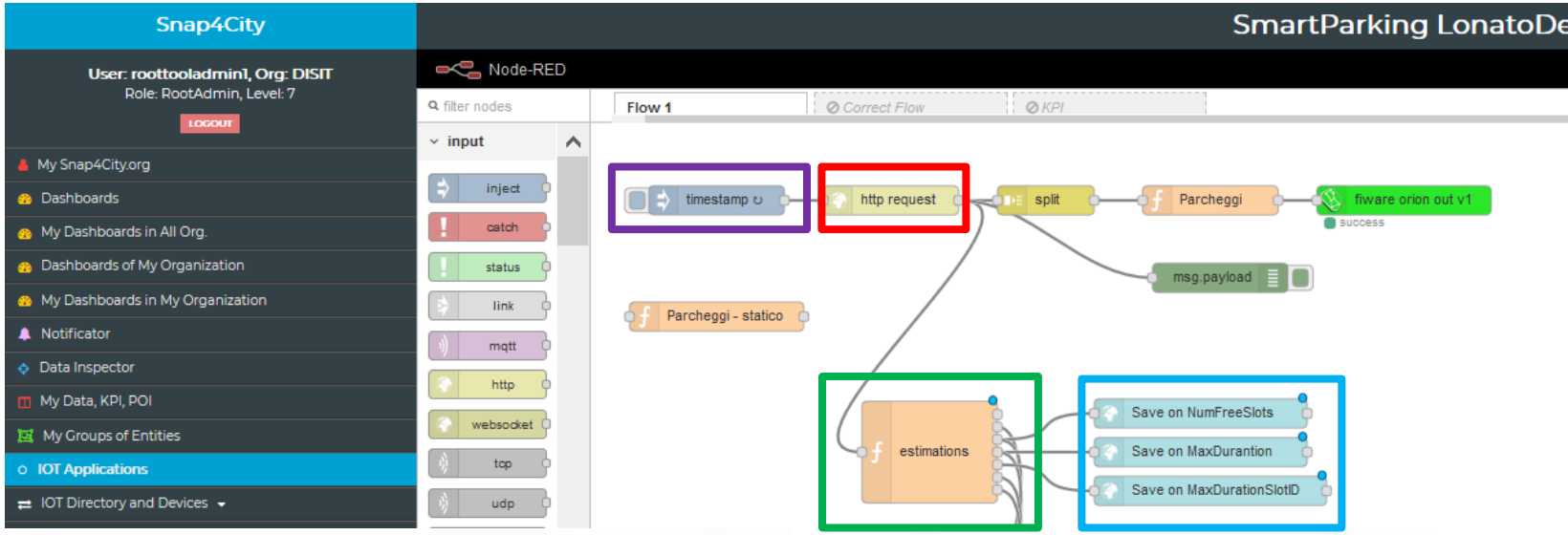
- My Data, KPI, POI > ‘Add My KPI’ button
- Verify the KPI existence in My Data, KPI, POI
- Create your IoT App sending data to your KPIs
- Example: Lonato Car Park:
  - NumFreeSlots
  - MaxDuration
  - MaxDurationSlotId

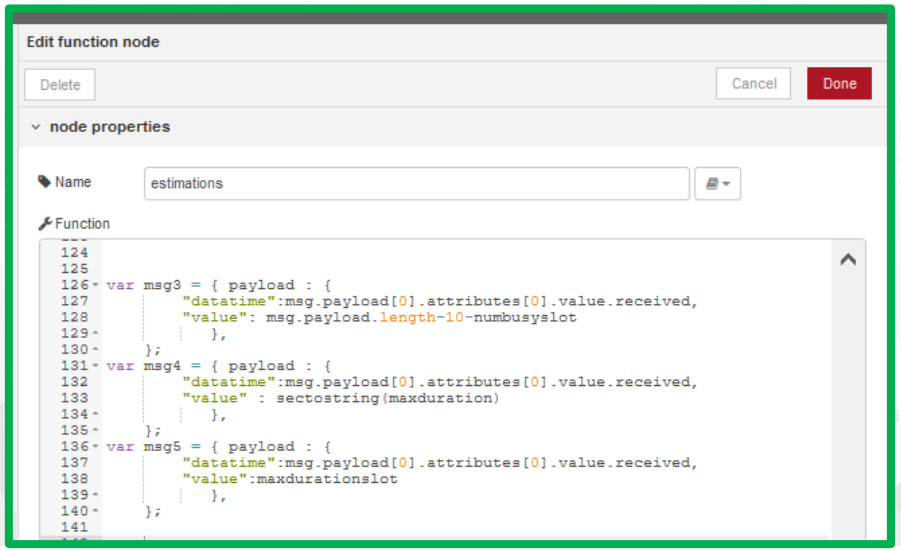


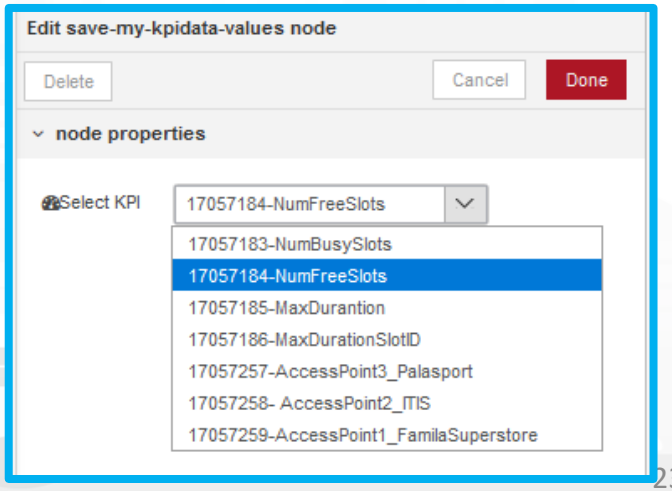


IoT App sending data to your KPIs

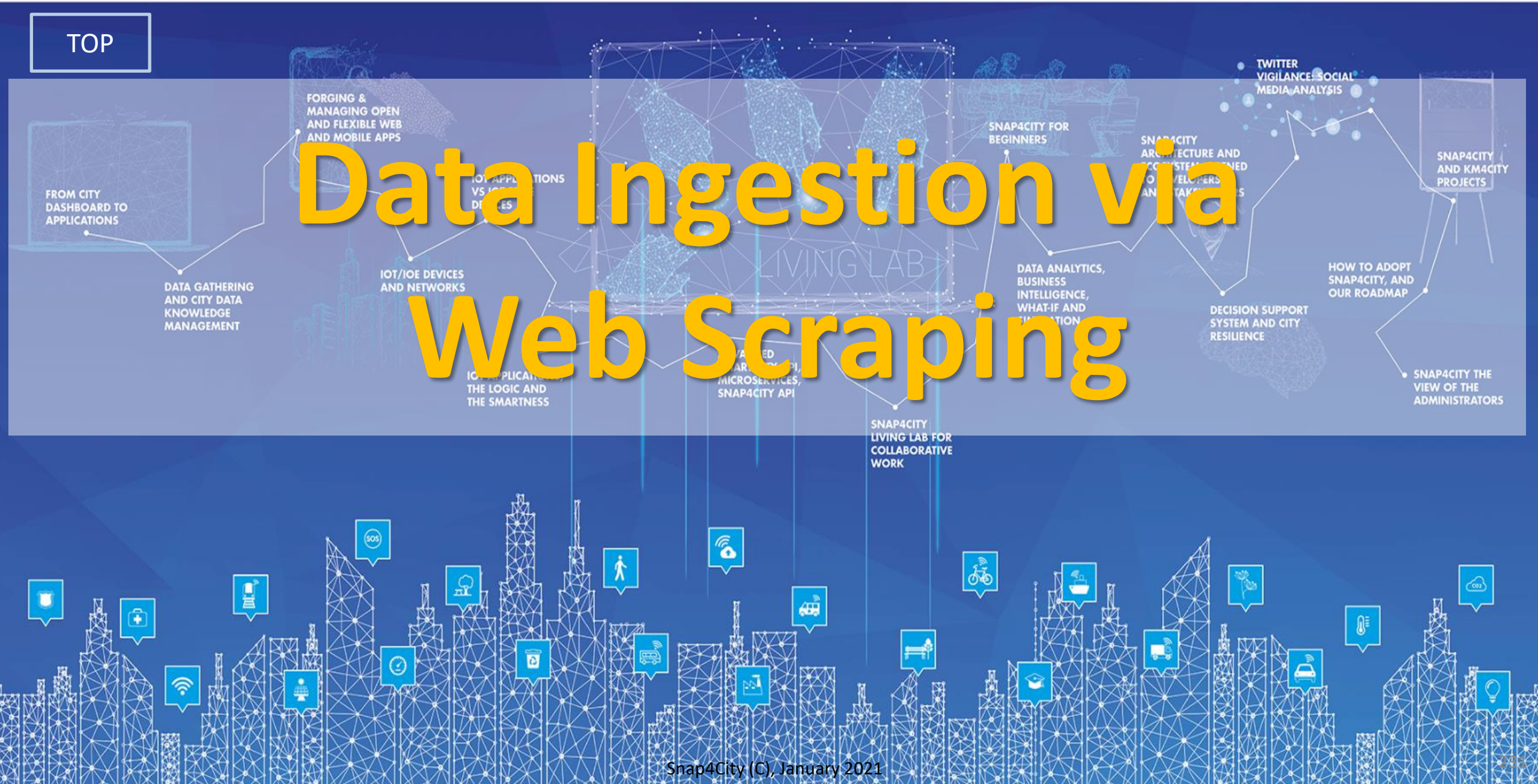
- Create your IoT App (ex: ‘SmartParking LonatoDelGarda’)
  1. Use an **inject** block to choose the frequency of update
  2. Use **http** block to download data (e.g.)
  3. Use a **function** of blocks to convert data in a specific json sending data to each KPI
  4. Use a **save-my-kpidata-values** block





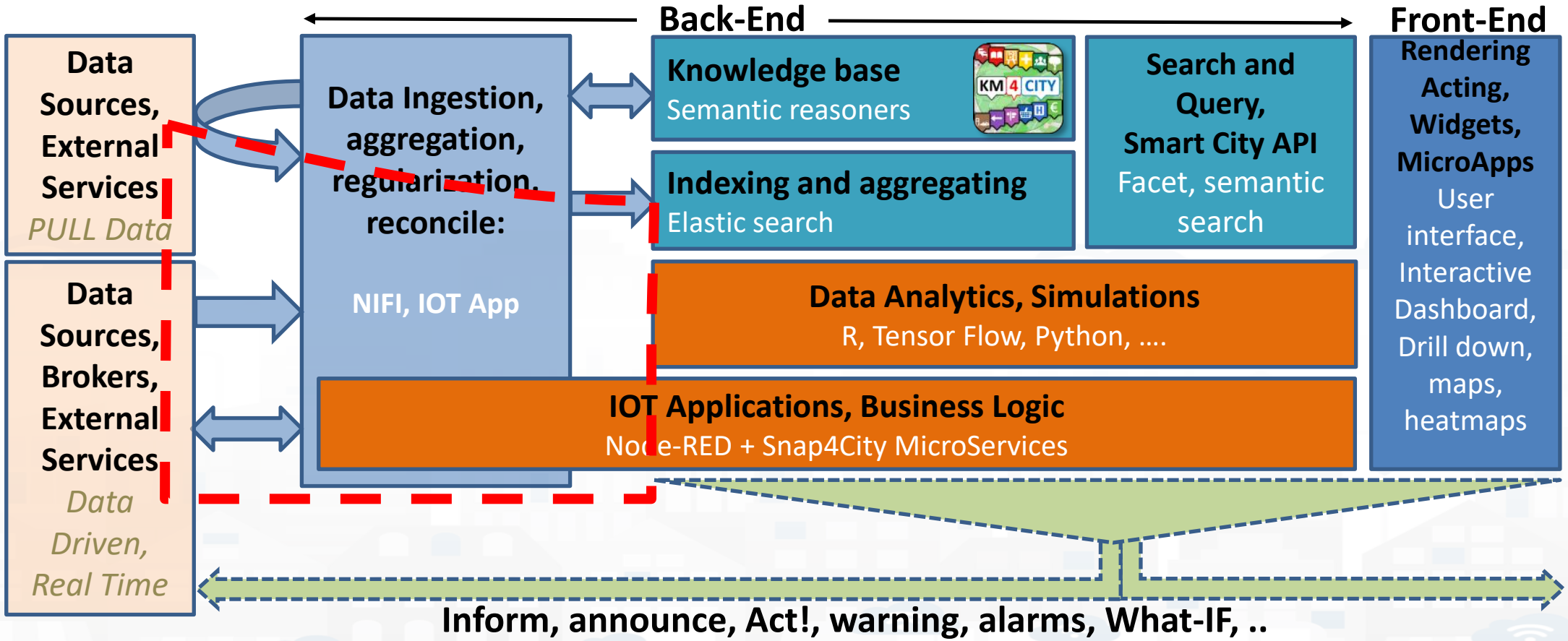


TOP



Data Ingestion via Web Scraping

Snap4City Architecture vs Data Ingestion

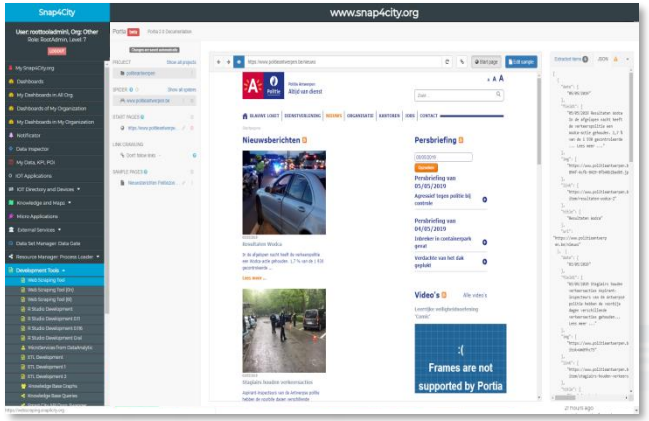


Web Scraping from Portia

My Scraping process



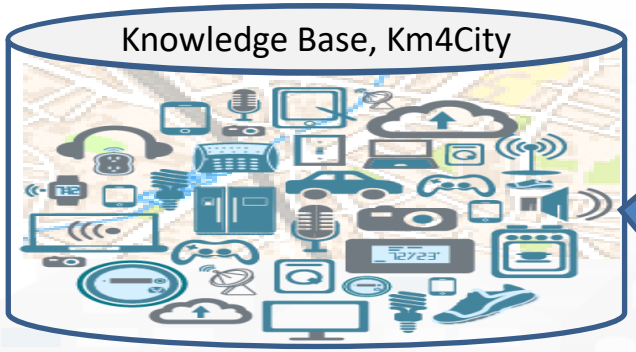
Web Scraper PORTIA

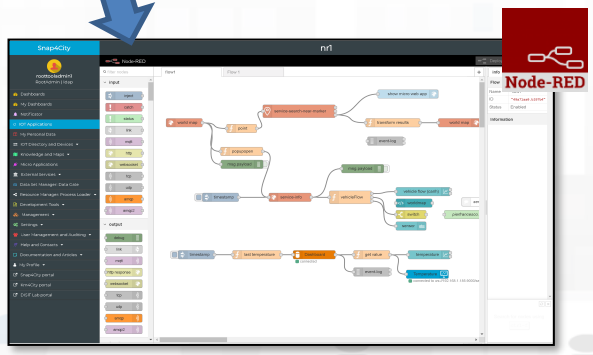


Generating WEB Scraping



Knowledge Base, Km4City





IOT App. Editor

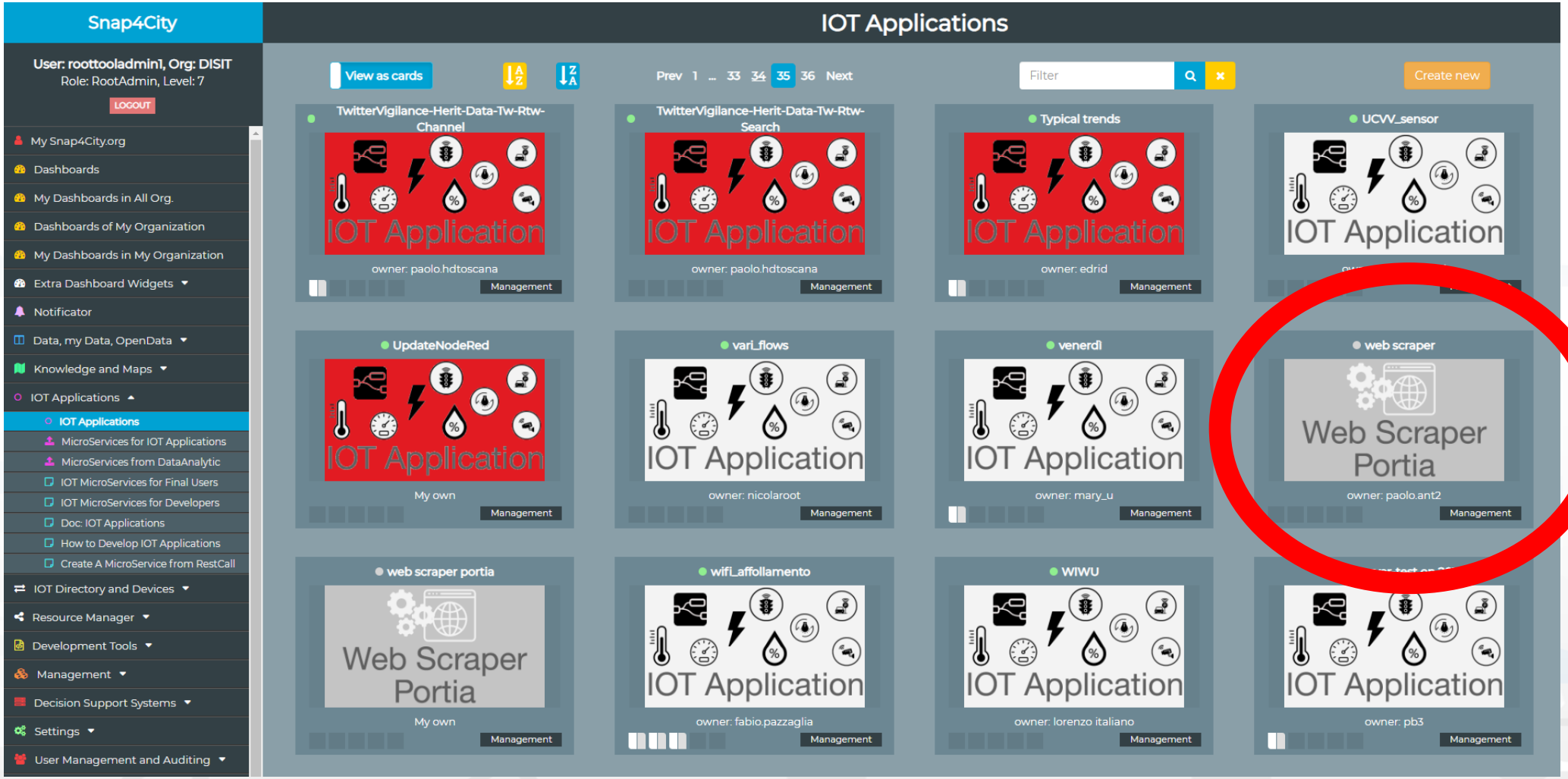
Sharing/saving reusing Scraping



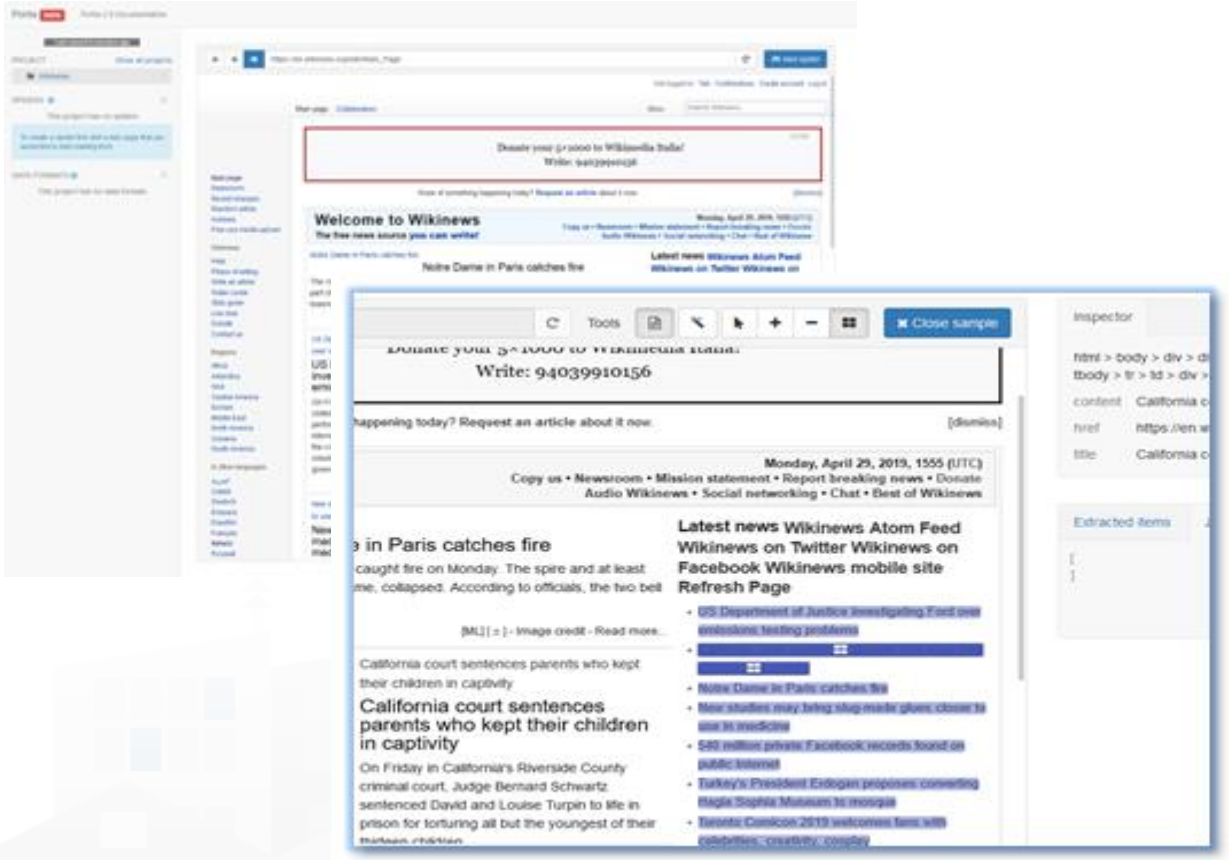
Resource Manager



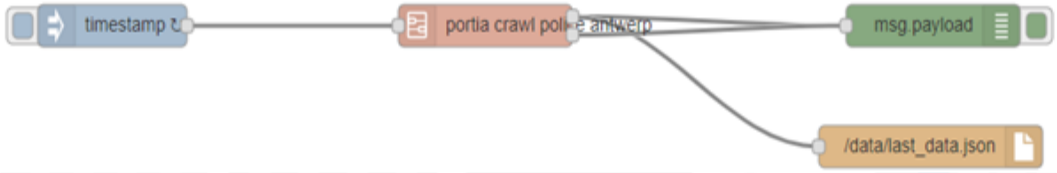
See them in the list of IOT Apps



Web scraping

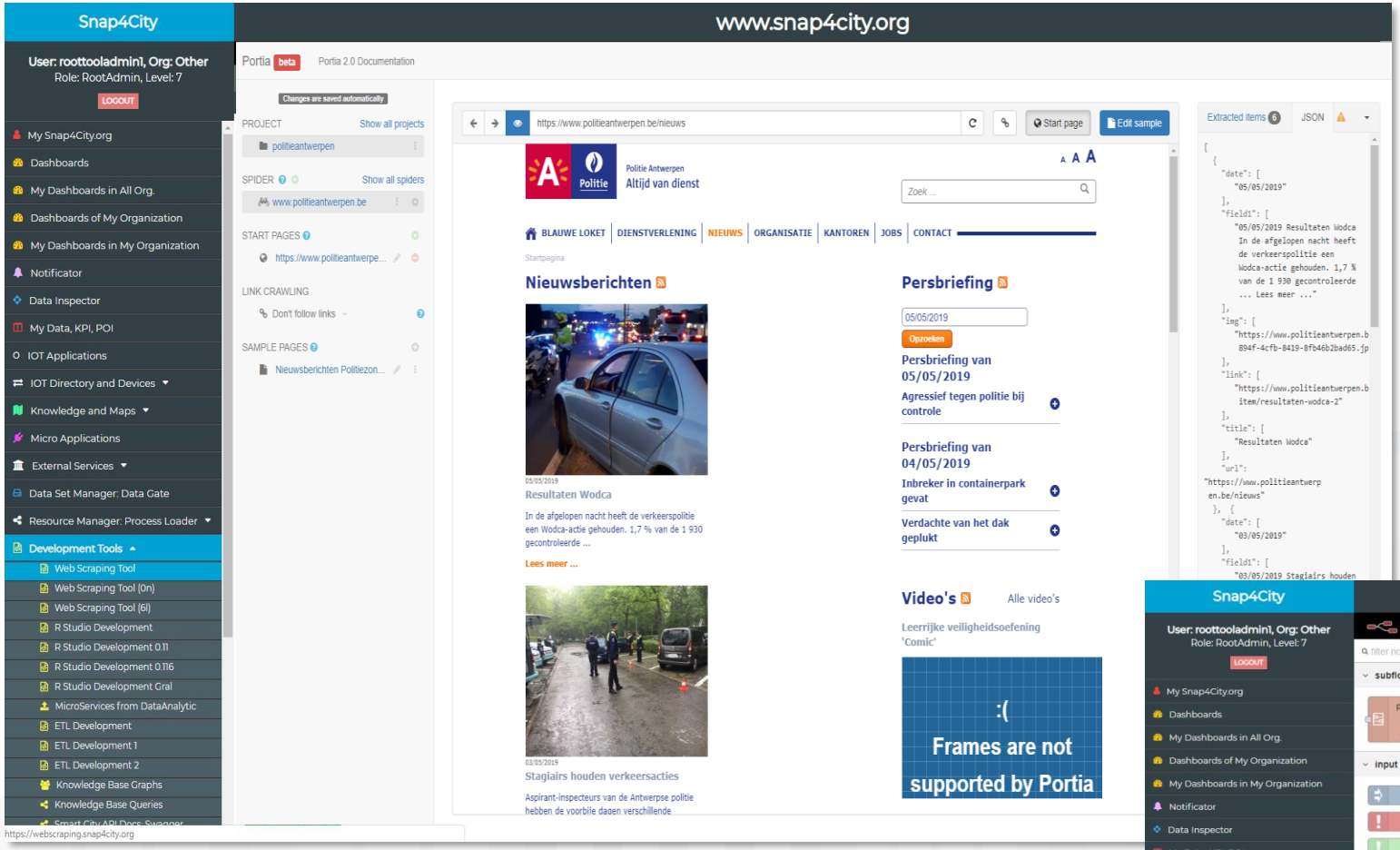


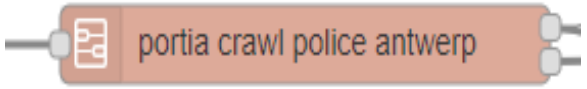


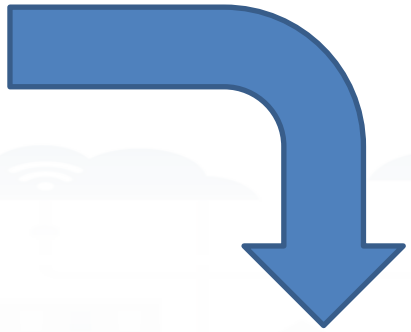


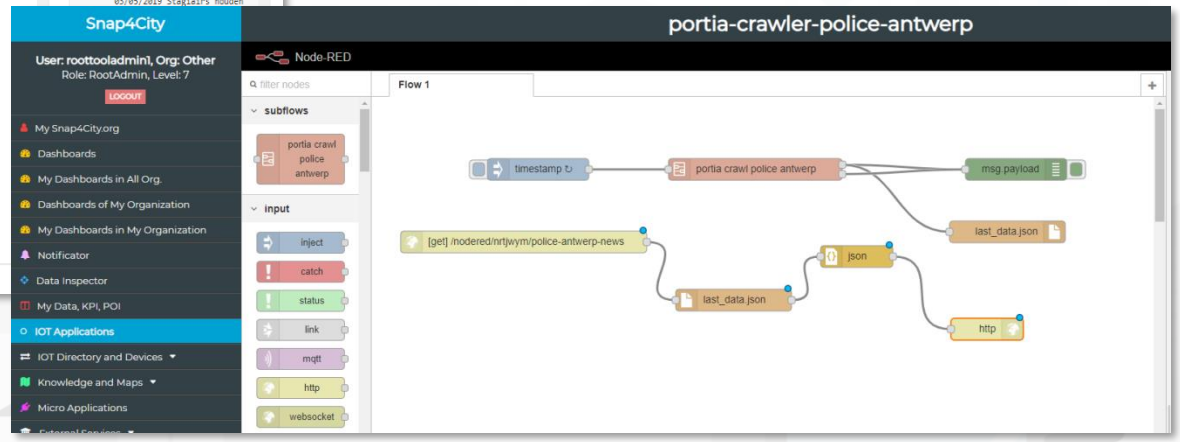
- [TC9.16 – Web Scraping to get data from web pages]

Web Scraping

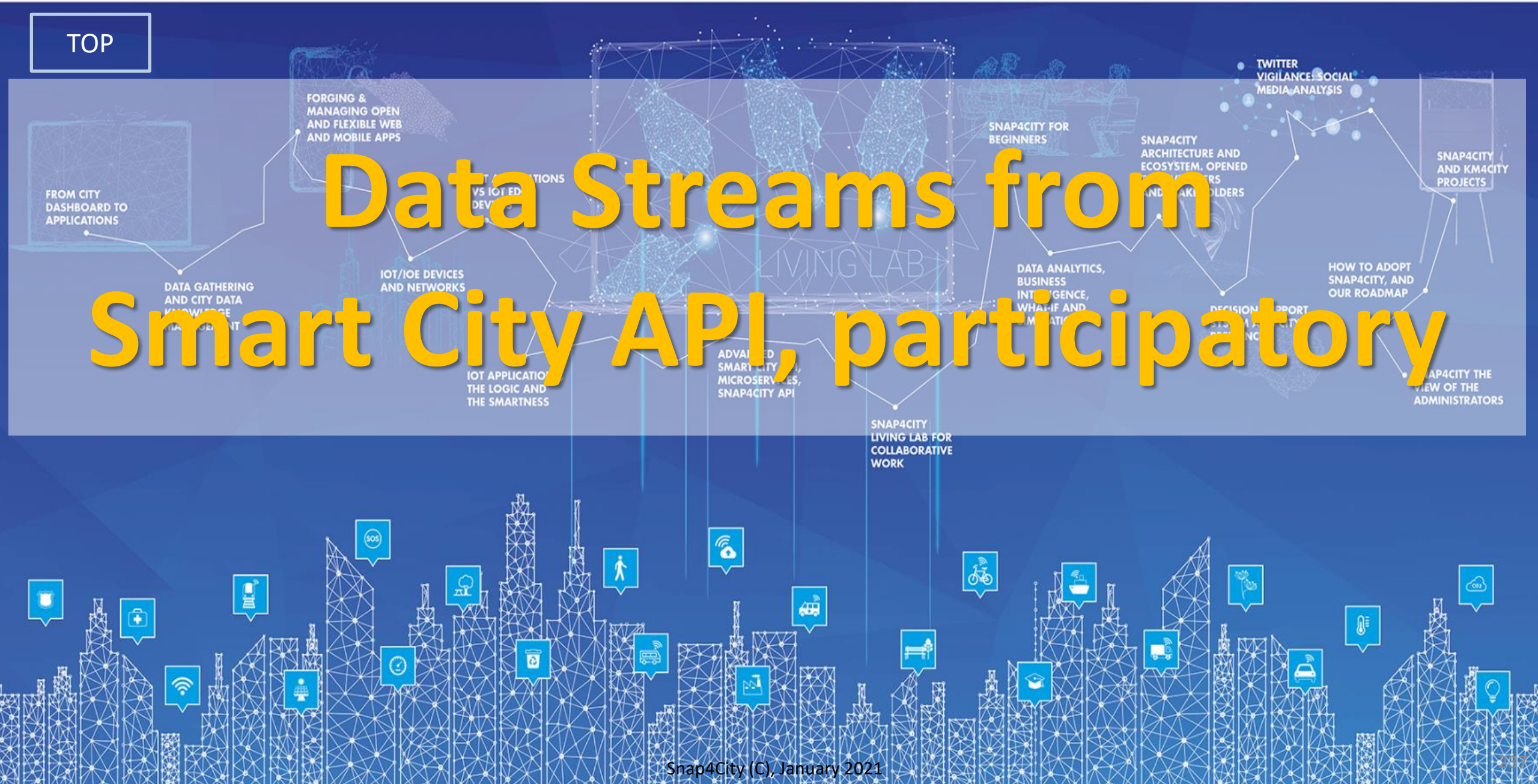








TOP



Data Streams from Smart City API, participatory

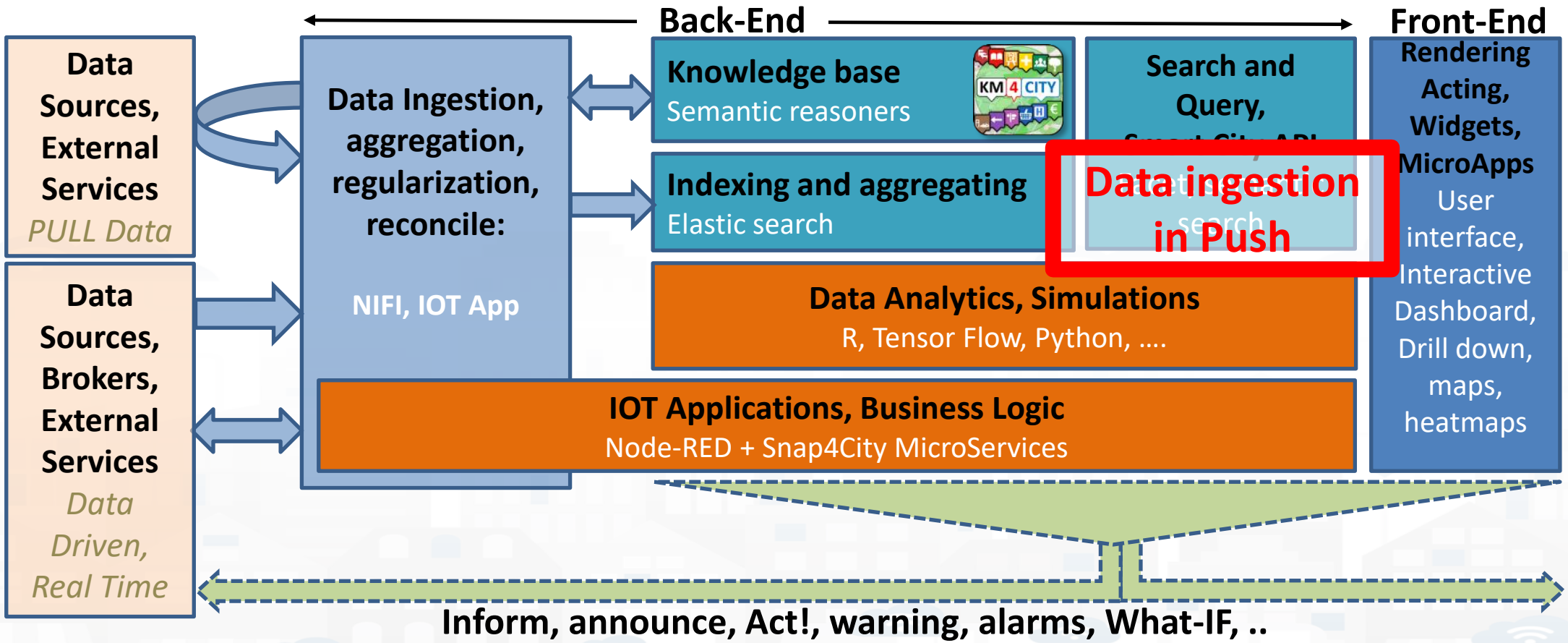
- We intend in this cases the data that can be posted on the infrastructure **by using the Smart City API**, such as:
  - MyPersonalData, MyKPI, data from IOT App on mobile, etc.
  - please see slides of Day 7
- Next section discusses those that are **automatically collected from Mobile Phone** and sent to the infrastructure on cloud via the Smart City API. For example:
  - Clicks on App
  - Post of images on POI
  - Post of comments on POI
  - Post of raking on POI
  - Questionnaires, and reactions
  - Eventual scores collected on the basis of the actions performed by the users.
  - Trajectories taken from the mobile phone positions and clicks
  - OBD2 data and positions
  - Etc.

TOP

Data Streams from Mobile Devices



Snap4City Architecture vs Data Ingestion

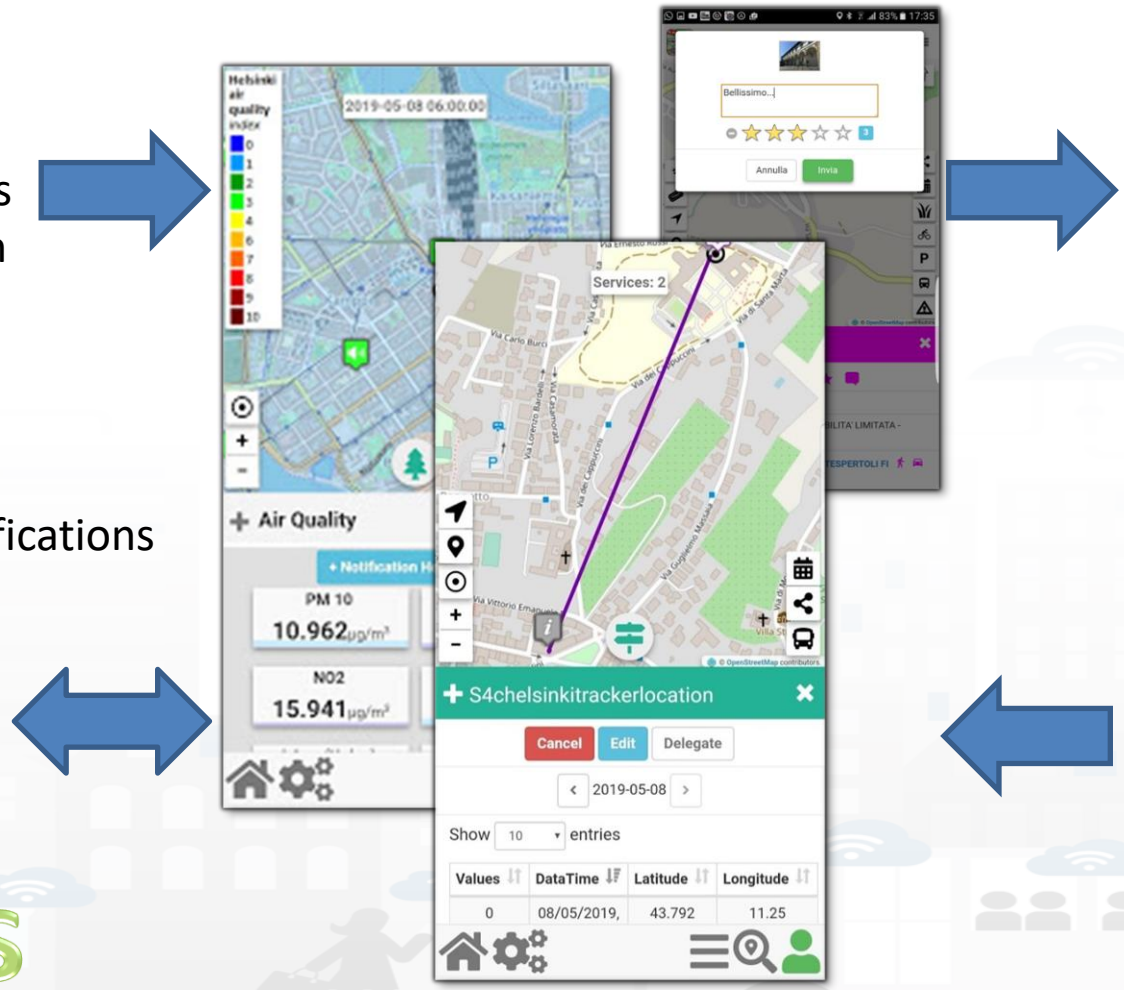


The App is a Bidirectional Device

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

Produced information

- Accepted ?
- Performed ?
- …



Derived information

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

Produced information

- Suggestions
- Engagements
- Notifications
- …

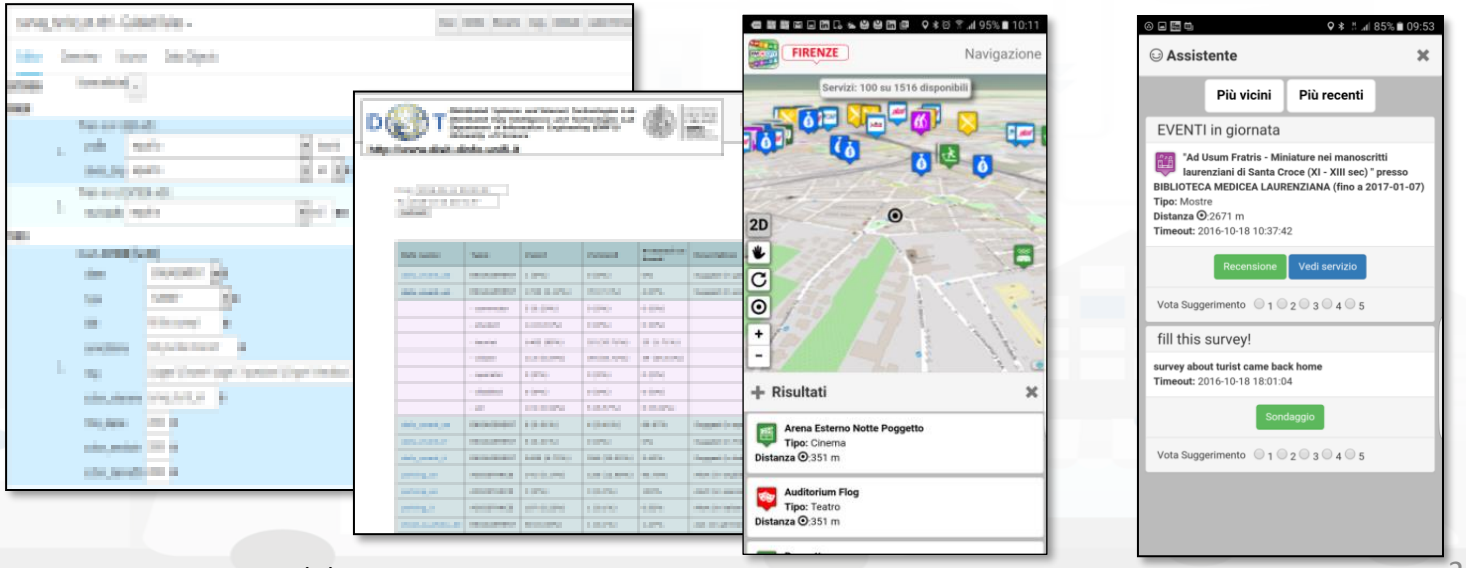
Users

System

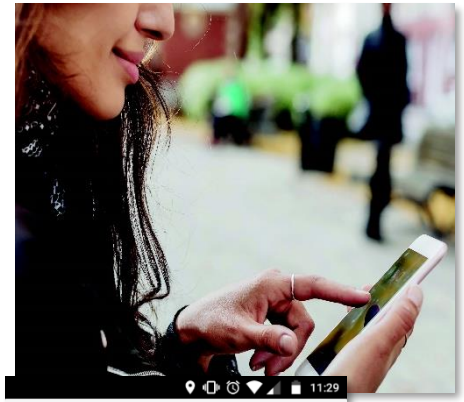
Profiled Engagements to City Users

- The users are profiled to learn habits:
  - Personal POI, paths, Mobility habits
- Information and engagements sent to the users are programmed according to the context and user behavior to:
  - Stimulate virtuous habits
  - More sustainable habits
  - More healthy habits, etc.
  - Get feedbacks
  - Provide bonus and prices, …..
  - Send alerts, ….

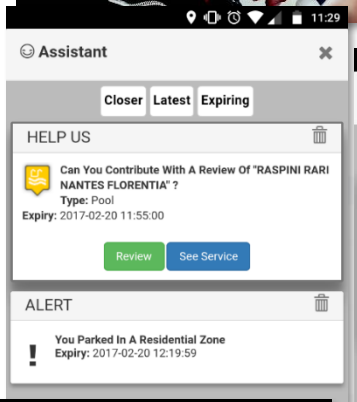


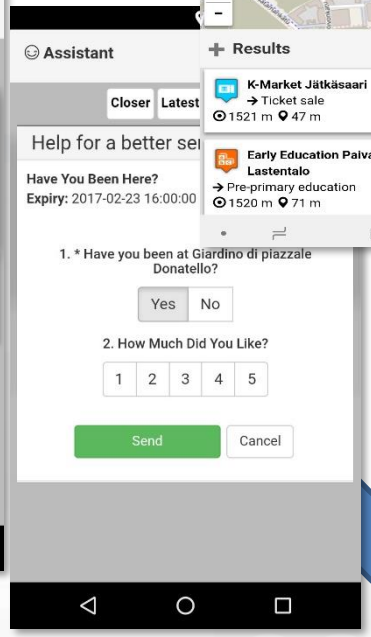


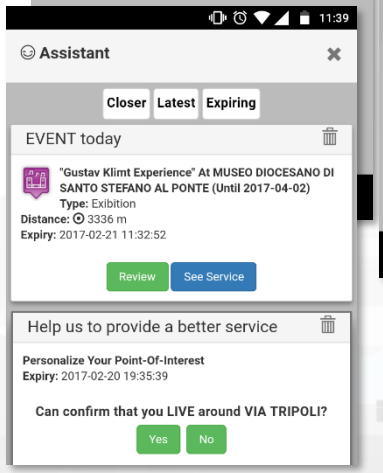
Users' Engagement









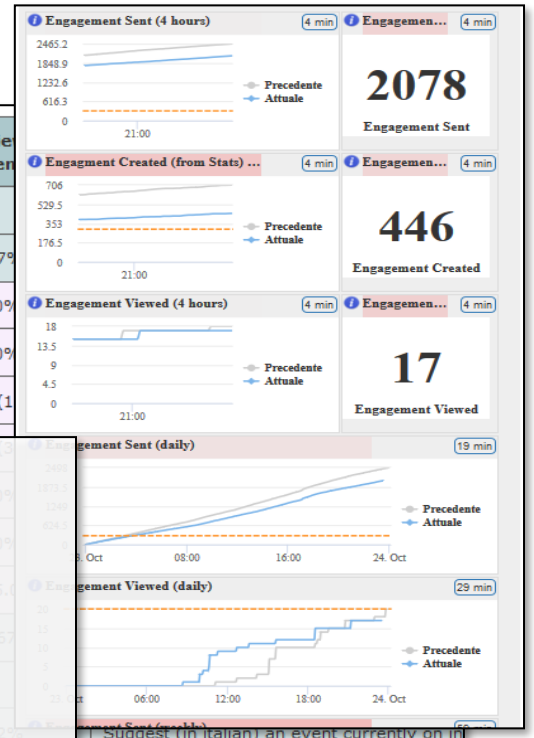


User context

Rules

City context

Rule name	Type	#sent	#viewed	#vie / #sen
daily_event_de	ENGAGEMENT	1 (0%)	0 (0%)	0%
daily_event_en	ENGAGEMENT	1720 (2.12%)	70 (7.1%)	4.07%
	- commuter	5 (0.29%)	0 (0%)	0 (0%
	- student	14 (0.81%)	0 (0%)	0 (0%
	- tourist	1462 (85%)	25 (35.71%)	25 (1



Inform

- Air Quality forecast is not very nice
- You have parked out of your residential parking zone
- The Road cleaning is this night
- The waste in S.Andreas Road is full

Engage

- Provide a comment, a score, etc.

Stimulate / recommend

- Events in the city, services you may be interested, etc..

Provide Bonus, rewards if needed

- you get a bonus since you parked here
- We suggest: leave the car out of the city, this bonus can be used to by a bus ticket

Suggest (in italian) an event currently on in

Alert (in english) if the user parked in a res

Alert (in spanish) if the user parked in a res

Alert (in italian) if the user parked in a resid

Ask (in german) a contribution for a nearby

MyKPI: Tracking of Devices and Mobiles

- Real Time Trajectories for
  - Mobile Phone
  - Moving IOT Devices
  - OBU, Vehicular Kits
  - Multiple tracks
  - Day by day
- Micro Application



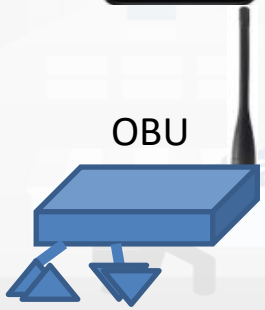
Mobile
PAX Counter



Apps

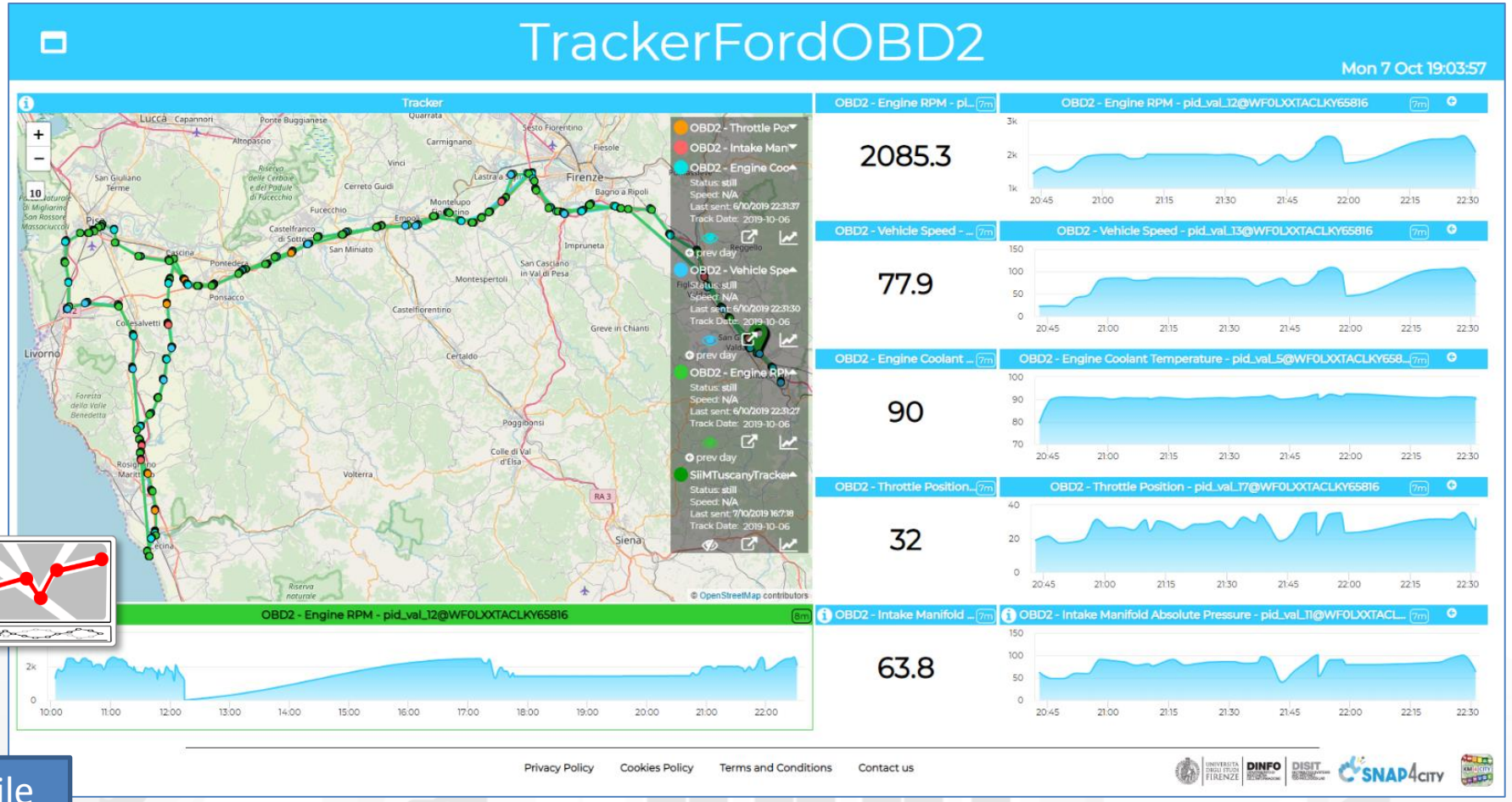


OBD2



OBU

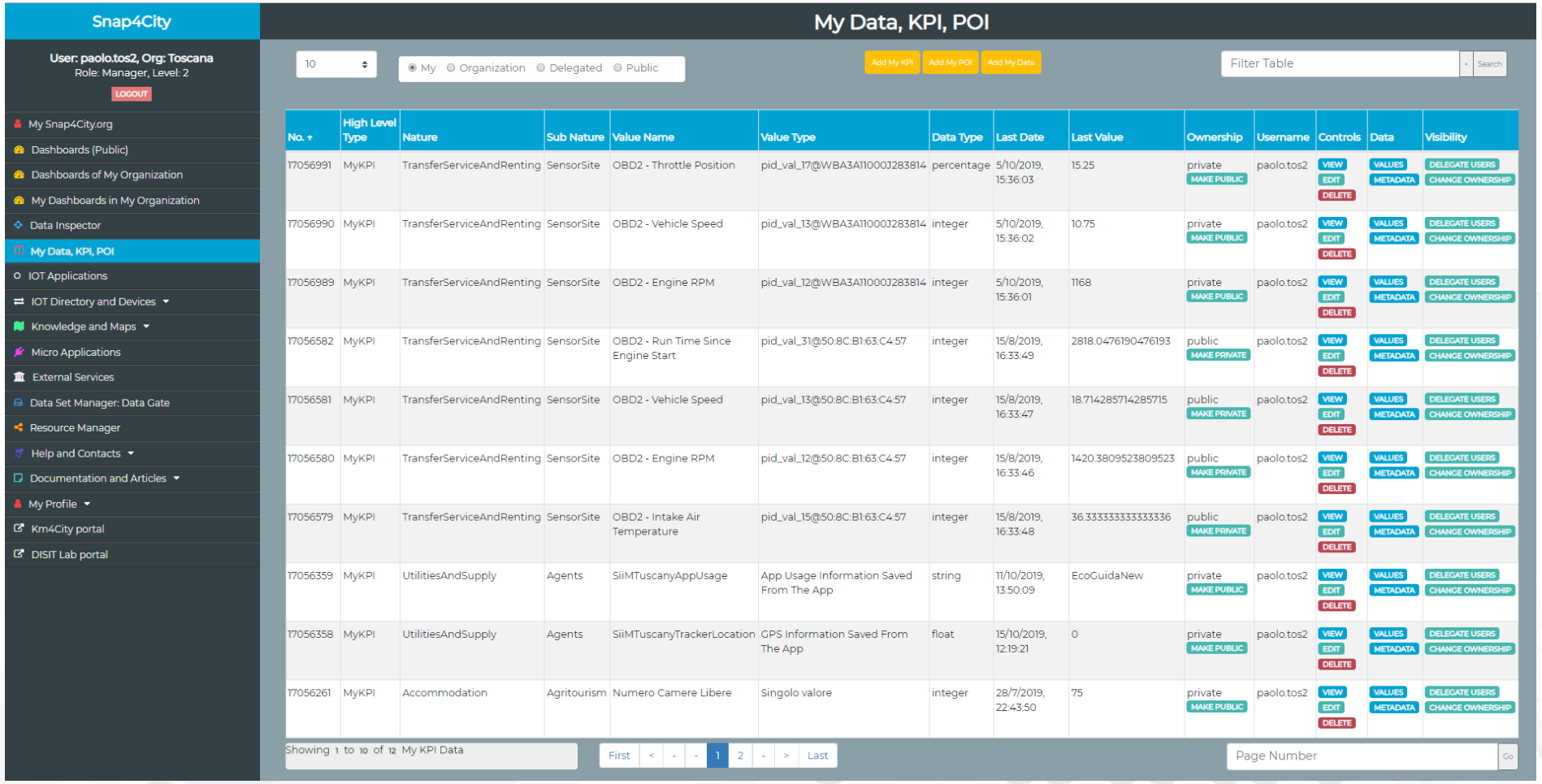
Mobile sensors



Load and change data via Smart City API

- **Data regarding Mobile Phones:**
  - Clicks on App
  - Post of images on POI
  - Post of comments on POI
  - Post of raking on POI
  - Questionnaires, and reactions
  - Trajectories taken from the mobile phone positions and clicks
  - OBD2 data and positions
  - etc.
- They are **automatically collected** and can be inspected by the user via special tools as presented in the following.

My KPI data view and manipulation



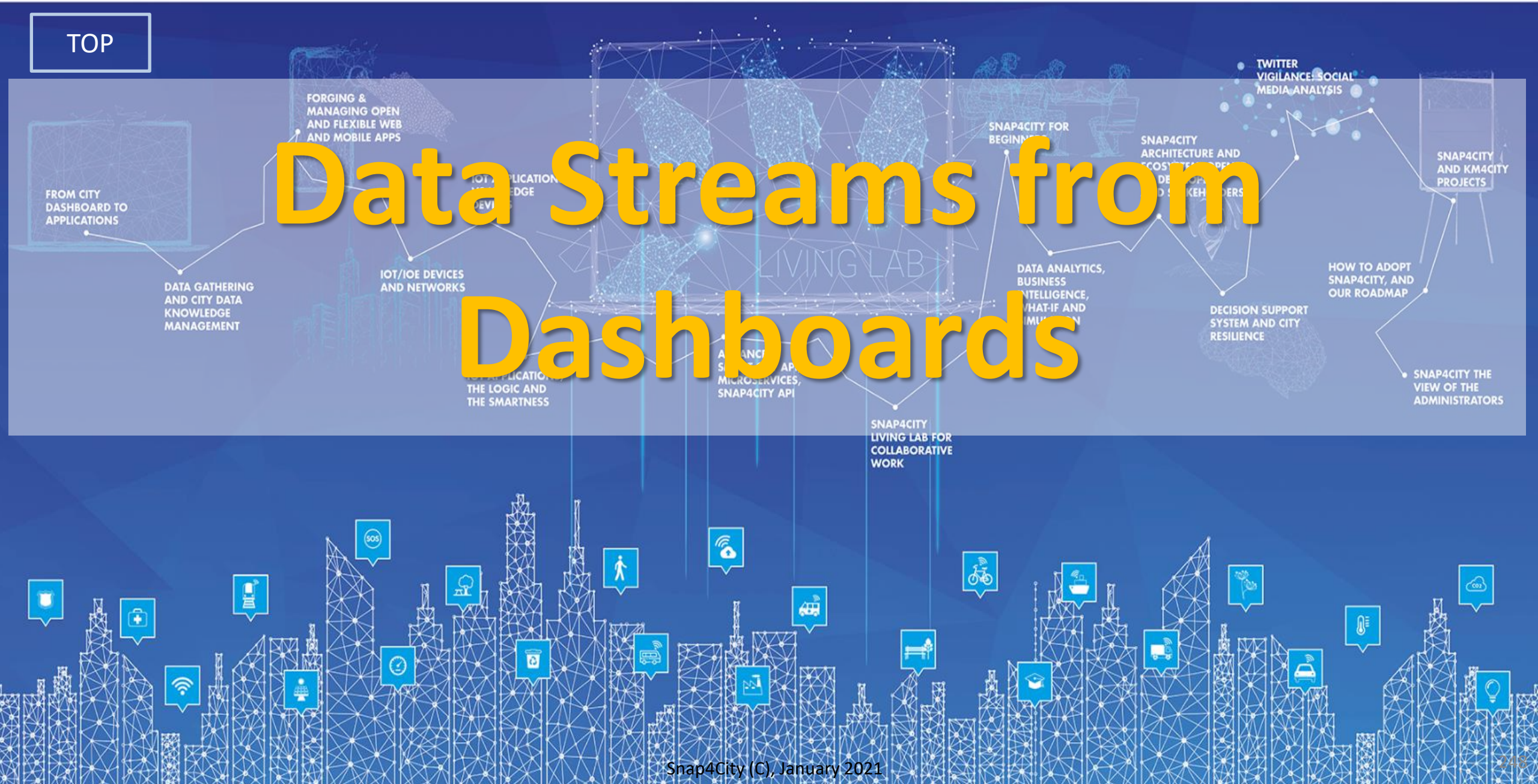
https://www.snap4city.org/mypersonaldata/

Further reading on MyKPI

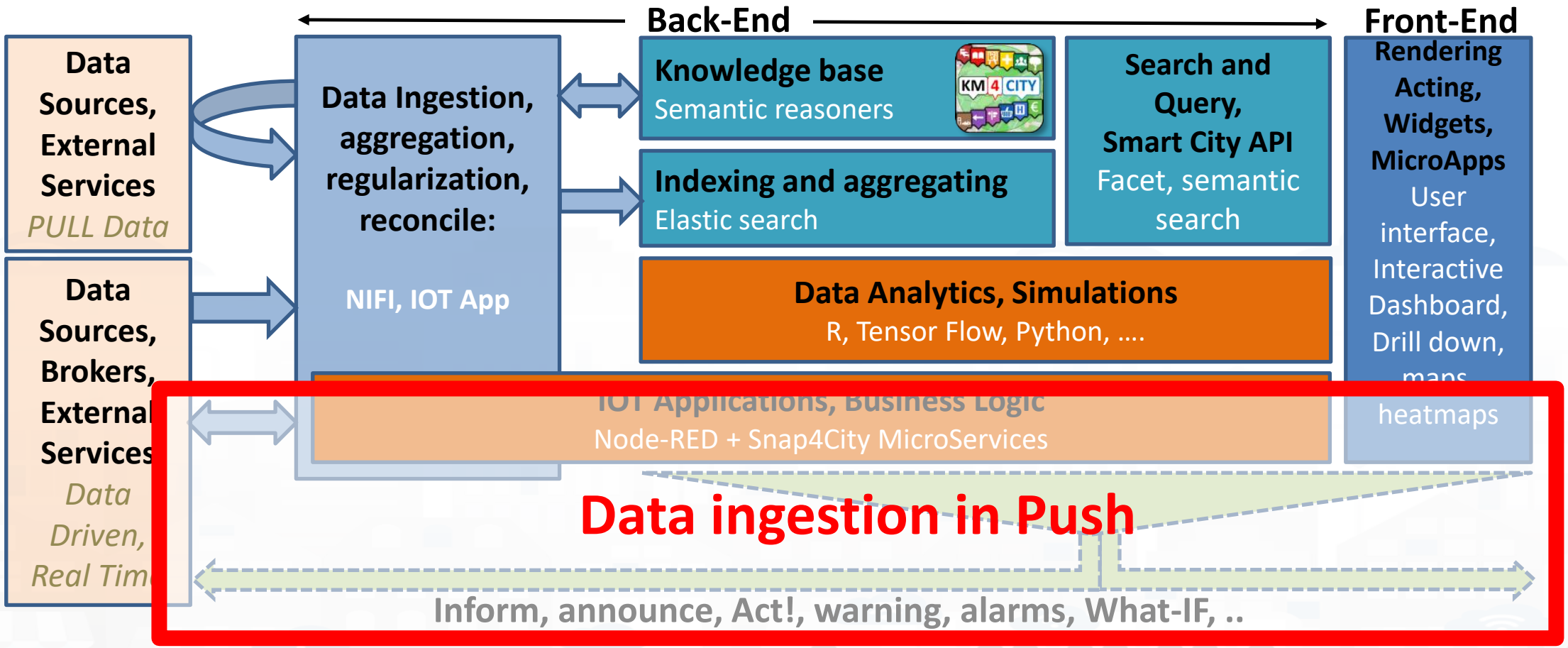
- TC1.17. Object tracking, widget tracker, personal tracking/trajectories, moving sensors
- TC 2.35- How manage My KPI with Dashboard
- Solution: using PAX Counters, monitoring museum and events
- HOW TO: define privacy rules for personal data, produced by the end-users own device
- TC9.13: How to upload a local file into the platform (IOT Application and MyKPI)

TOP

Data Streams from Dashboards



Snap4City Architecture vs Data Ingestion



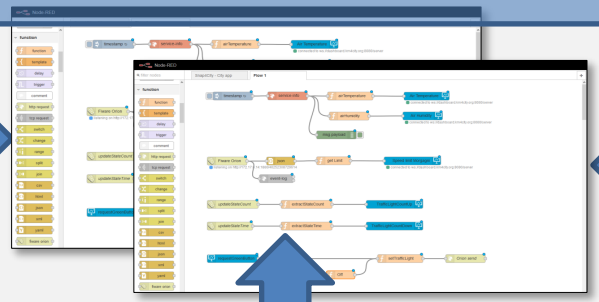
Snap4City: Builder of Sentient Cities Solutions

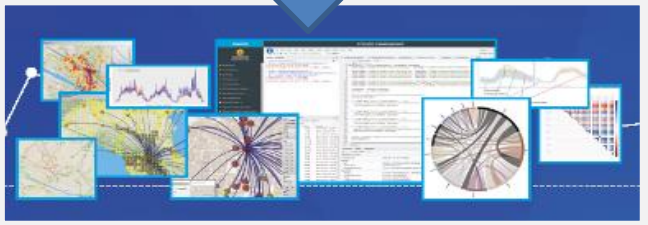
Dashboards with data driven IOT Applications enforcing intelligence

IOT and data World



IOT Applications



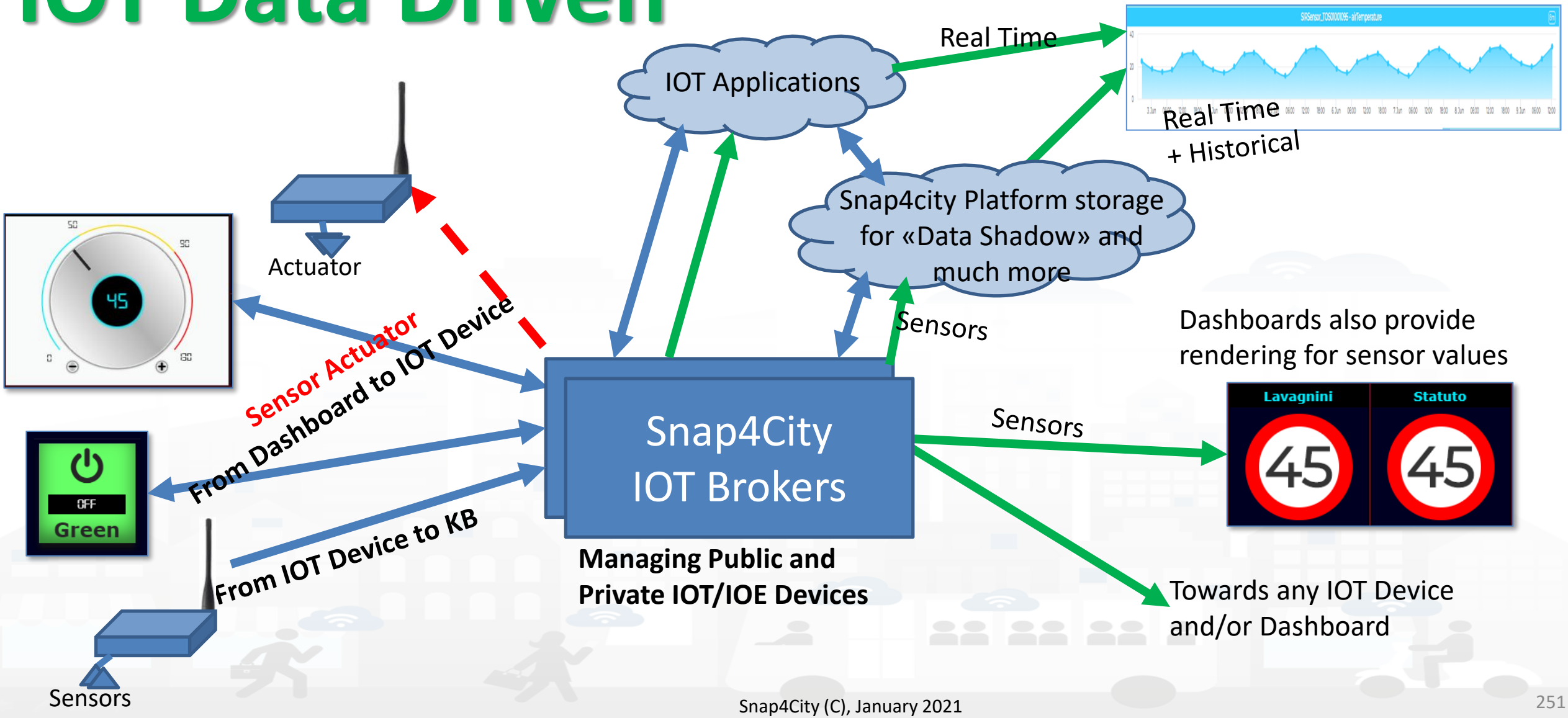


Dashboards and Apps

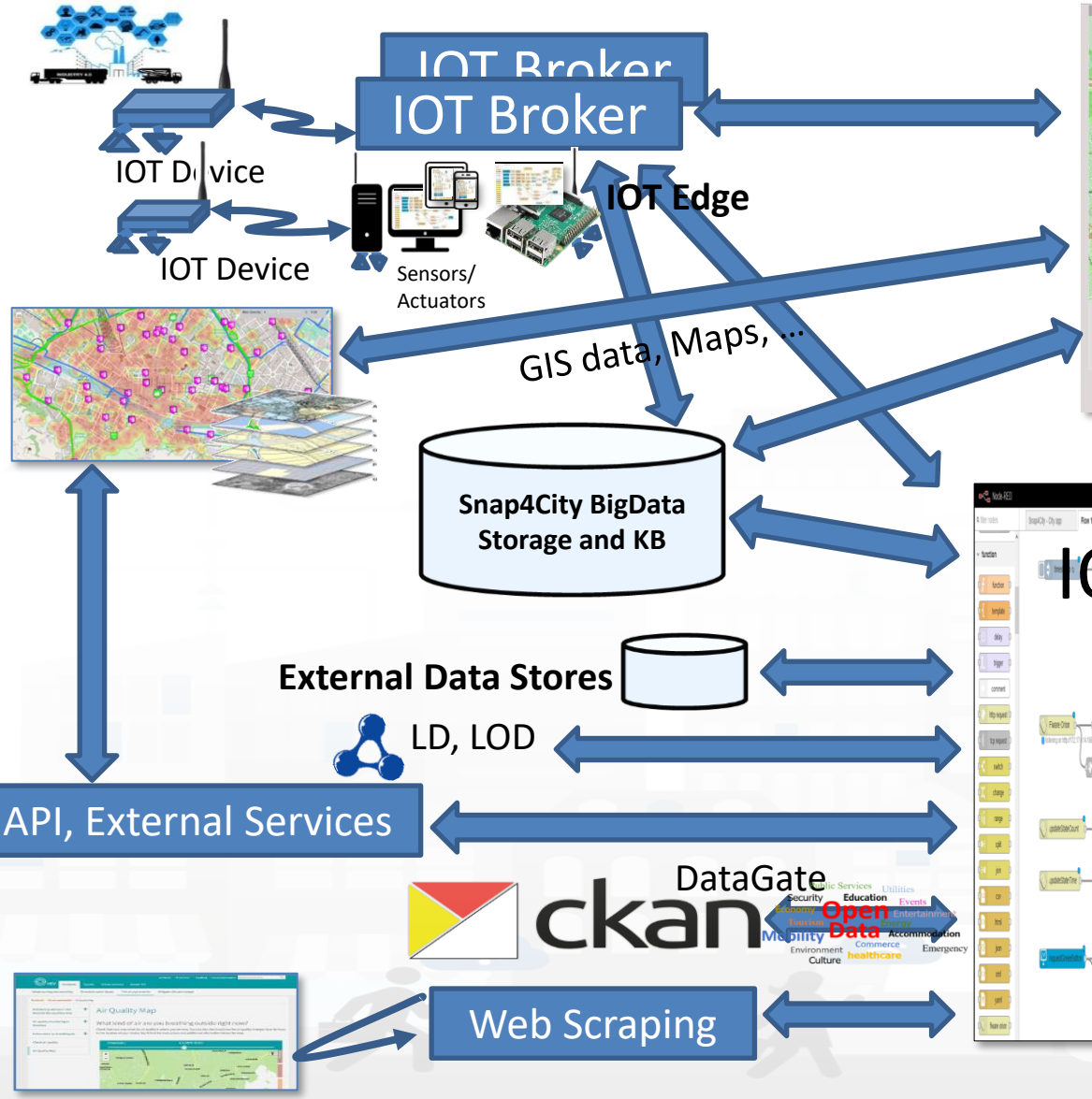


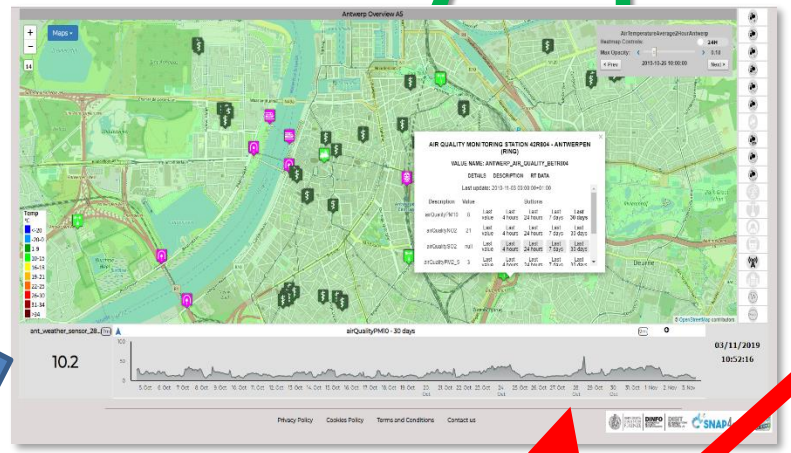
Big Data Analytics, Artificial Intelligence

IOT Data Driven

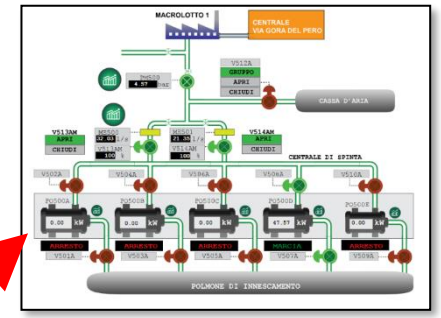


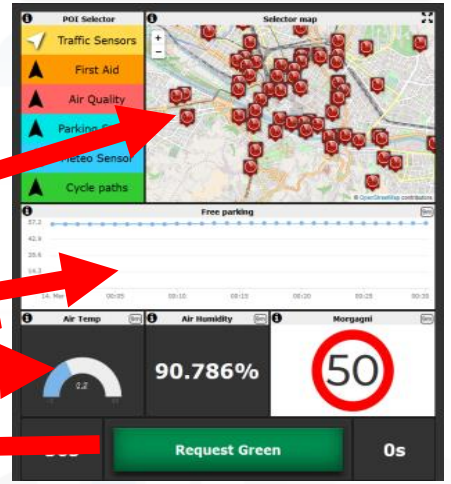
What the Dashboards can directly exploit



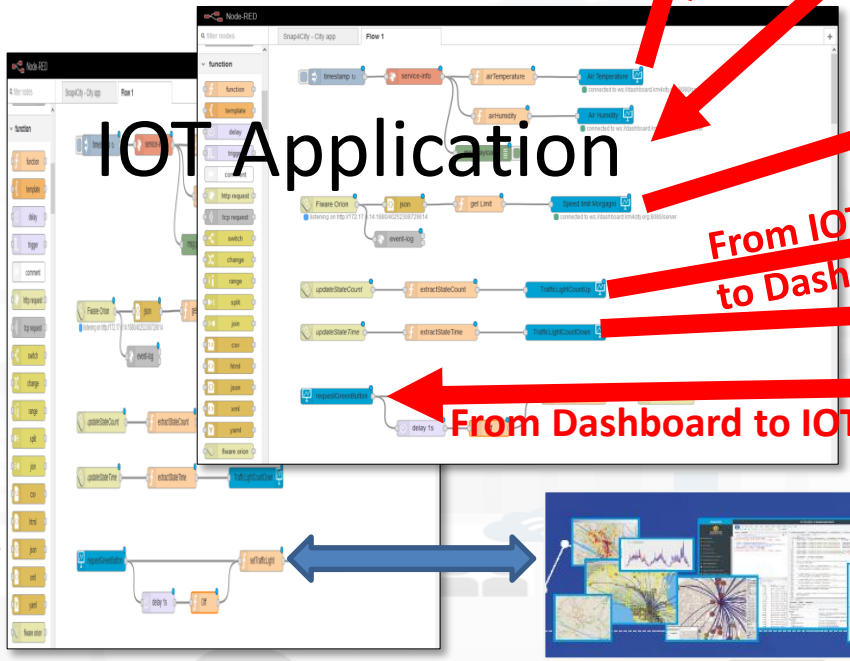


Dashboards





IOT Application



From IOT App to Dashboard

From Dashboard to IOT App

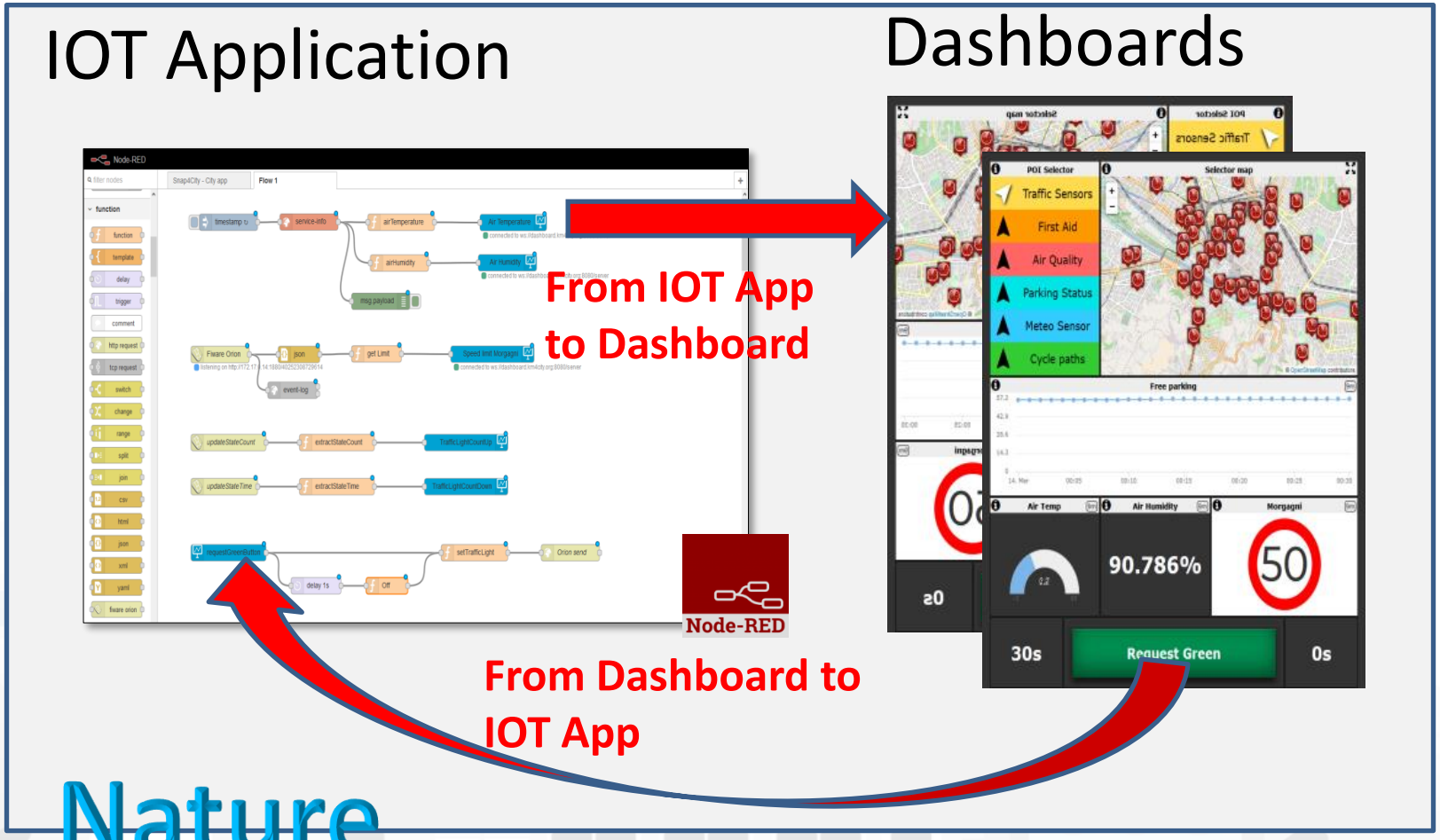


Big Data Analytics,
Artificial Intelligence
Access to all data

HLT: Sensors-Actuators

High Level Types

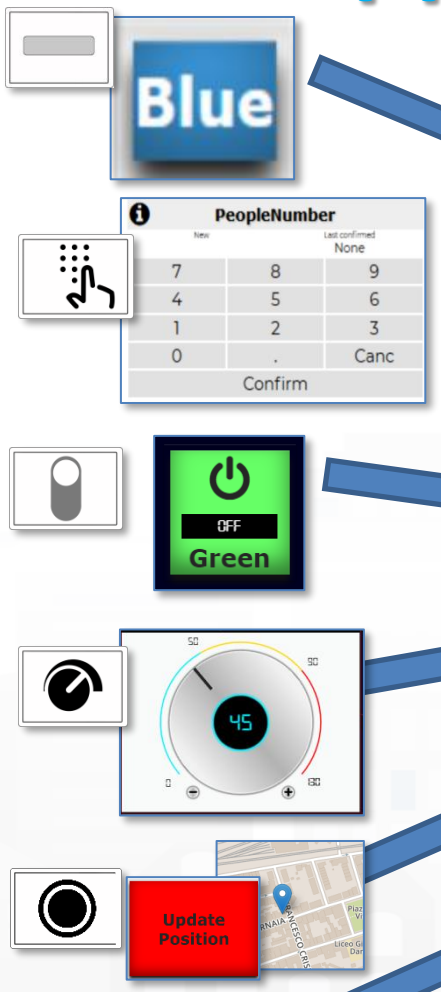
- **Complex Event**
- **Dashboard-IOT App**
- **External Service**
- **Heatmap**
- **KPI** (Key Performance Indicator)
- **MicroApplication**
- My Personal Data
- MyKPI
- MyPOI
- **POI** (Point of Interest)
- **Sensor**
- **Sensor Actuator**
- **Special Widget**
- Wfs (GIS)



Dashboard-IOT App

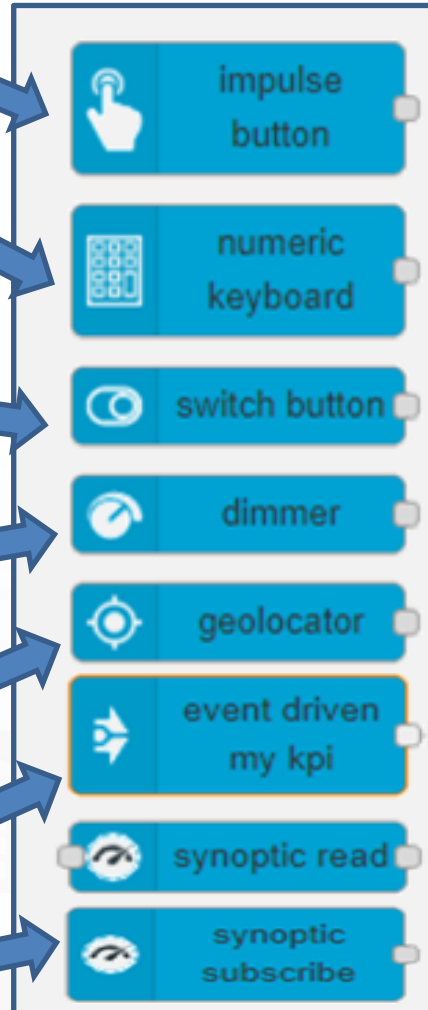
From Dashboard to IOT App

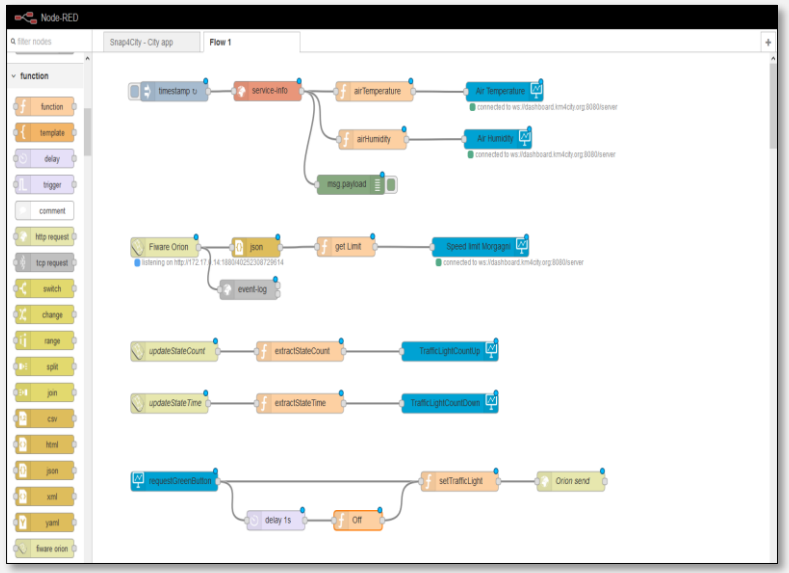




MyKPI variable

Synoptics



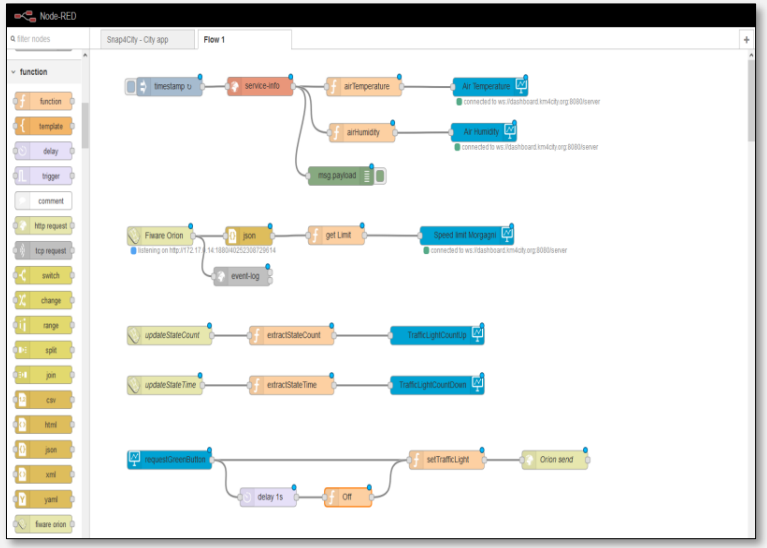


IOT Application

Nature

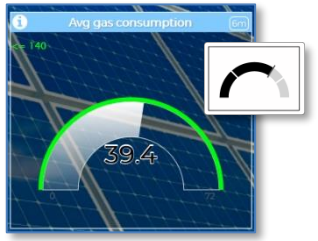
Dashboard-IOT App

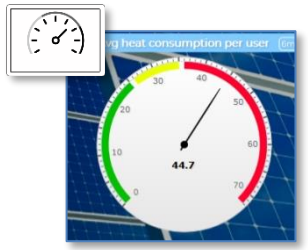
From IOT App to Dashboard



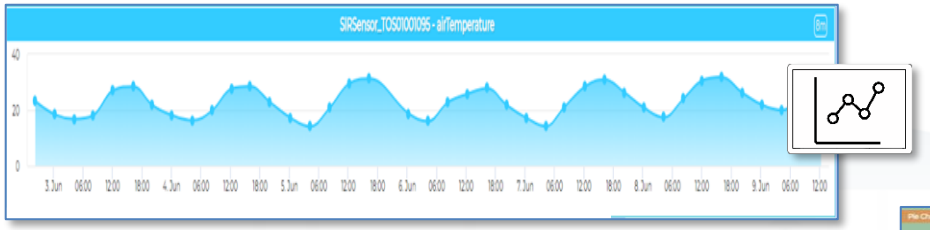
IOT Application

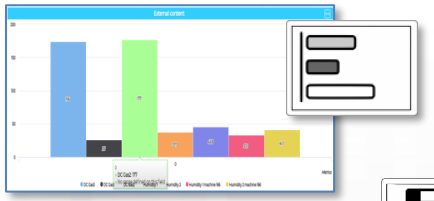
- gauge chart
- single content
- speedometer
- horizontal single bar
- vertical single bar
- web content
- time trend
- bar series
- radar series
- pie chart
- curved line series
- table content
- synoptic write

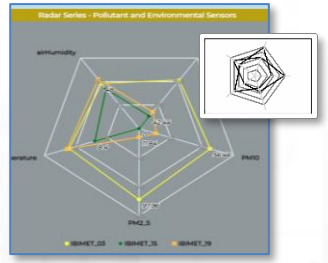


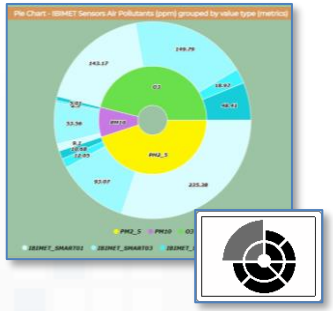


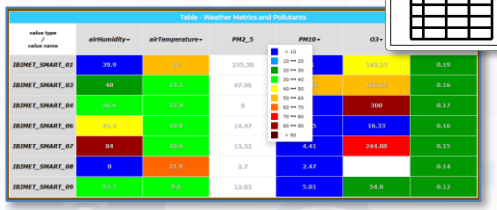


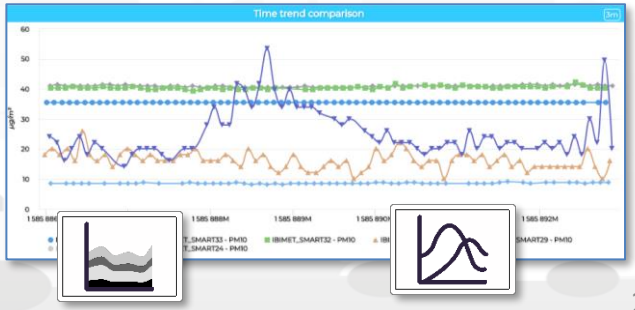


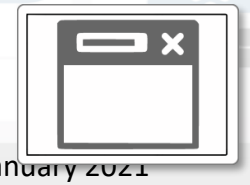












Dynamic (+++)

Widgets ICONS	Widget Name, Description	IOT App	Dashboard-IOT App	KPI (metric)	MyPersonal Data	MyData	My KPI	Sensor
XX	Single Content	X (cs)	X (DD)	X	X	X	X	X
50	Speed Limit			X				X
	Speedometer	X (cs)	X (DD)	X	X	X	X	X
	Gauge	X (cs)	X (DD)	X	X	X	X	X
	Single Bar, V/H	X	X (DD)	X				
	Single and Multiple Bars, stacked or not	X (cs)	X (DD)	X	X	X	X	X
	MultiSeries, shaded, staked and non staked	X (cs)	X	X	X	X	X	X
	Time Trend (single)	X	X (DD)	X	X	X	X	X
	Time trend Compare			X			X	X
	SpiderNet, radar, Kiviat	X (cs)	X (DD)	X	X	X	X	X
	Pie, Donut, 2 layers Donut	X (cs)	X (DD)	X	X	X	X	X
	Table	X (cs)	X (DD)	X	X	X	X	X

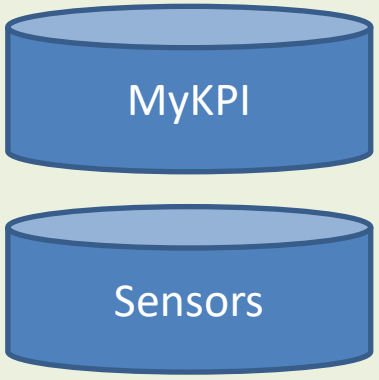
DD: Data Driven

Legenda

- **IOT APP column in previous table:**
  - **X**: means that from the IOT App you can send a new value or array to the widget directly, without the need to have is stored into Sensor or MYKPI variable, etc.
  - **CS, widget supports Change Source**, in the sense that: from the IOT App is possible to send a command to the Widget to change the data source. E.g., selecting sources among: Sensors (service URI), MyKPI (ID), any value produced on the IOT App directly. (cs) recent additions
- **Dashboard IOT App column in previous table**:
  - **X:** there is a MicroService / node on IOT App to act on those widgets on dashboard. The data are visualized.
  - **DD, widget is Data Driven**, in the sense that new data in push can be sent and the widget is updated in real time on web page without web page realoading

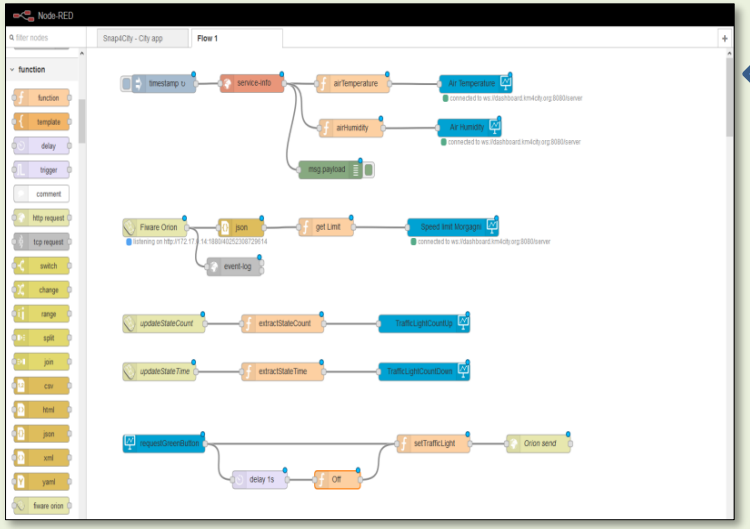
TC4.9: New Support Widgets for Bars, Barseries, Trend, and Series, on Dashboards and IOT Applications (partially obsolete)

From-To Custom Widgets / Synoptics to Storage in WS



MyKPI

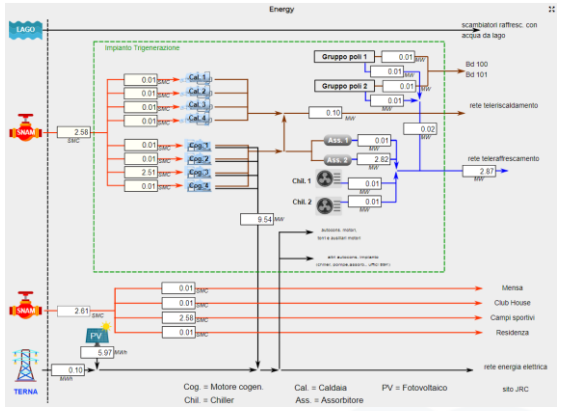
Sensor

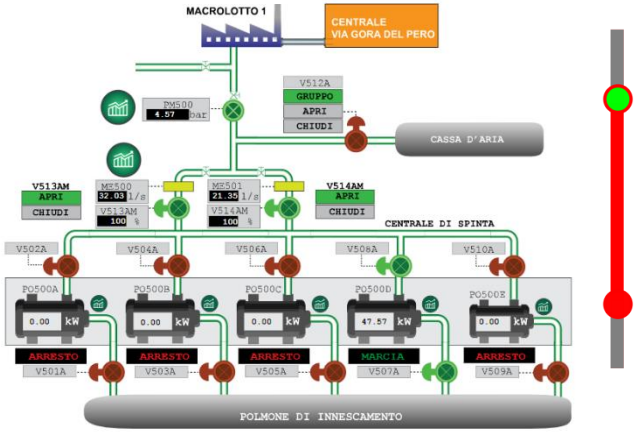


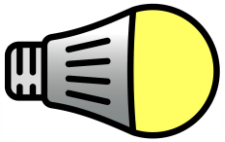
[ ]

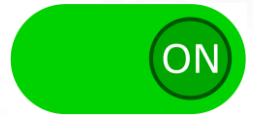
New Shared Variables

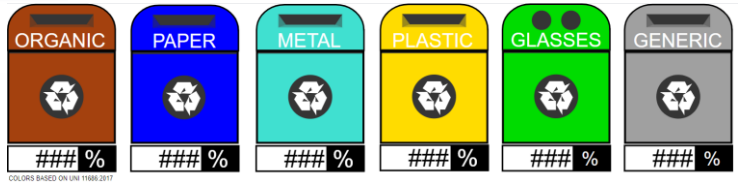
Constant Values

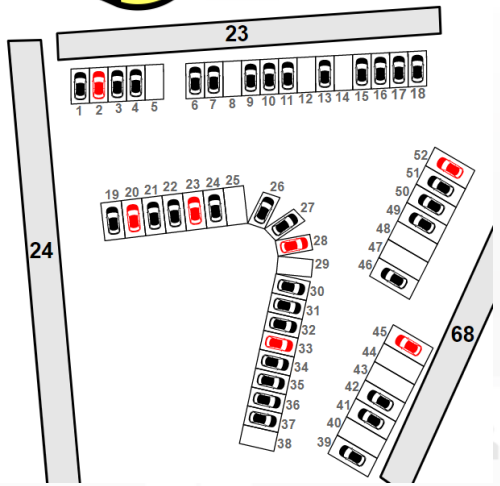






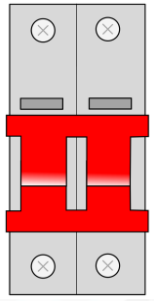


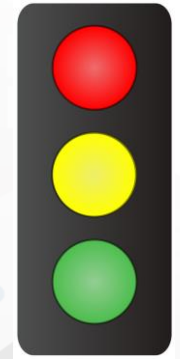












Web Socket Secure

TOP

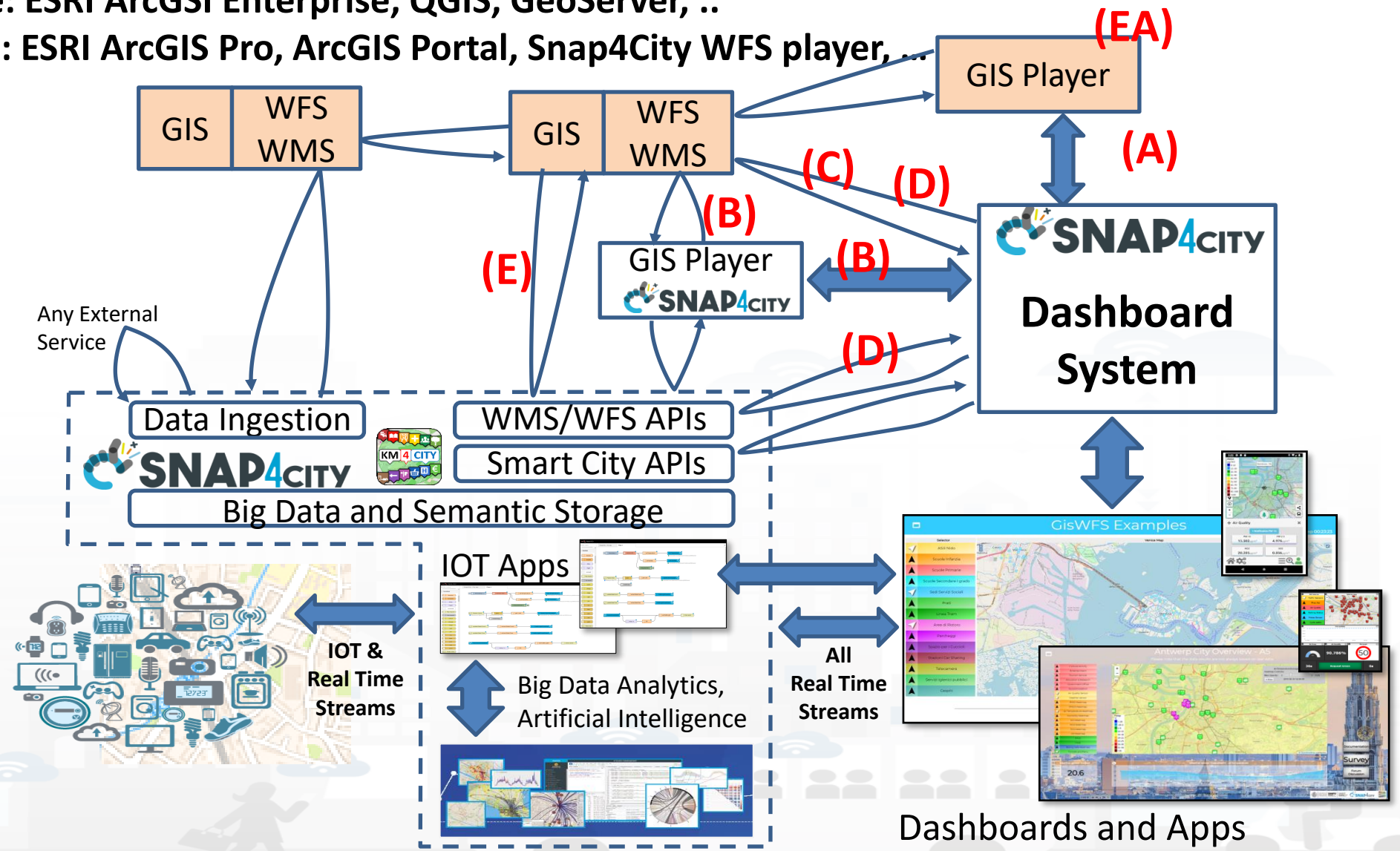
GIS Data Import and Export: WFS and WMS



https://www.snap4city.org/drupal/node/368

GIS vs Sna4City

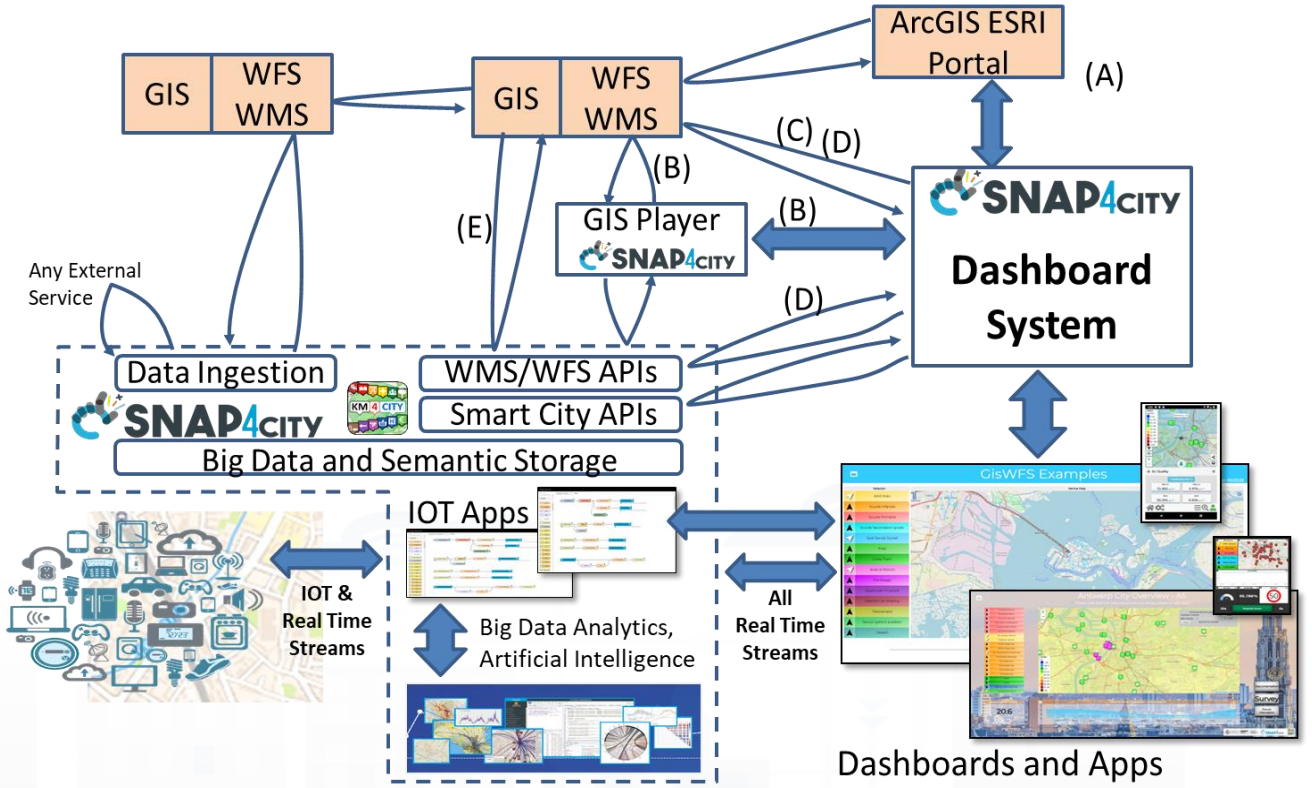
GIS Server can be: ESRI ArcGSI Enterprise, QGIS, GeoServer, ..
GIS Player can be: ESRI ArcGIS Pro, ArcGIS Portal, Snap4City WFS player, ...



- **GIS:**
  - Geographic Information System
- **WMS:**
  - Web Map Service
- **WFS:**
  - Web Feature Services

Interoperability ESRI

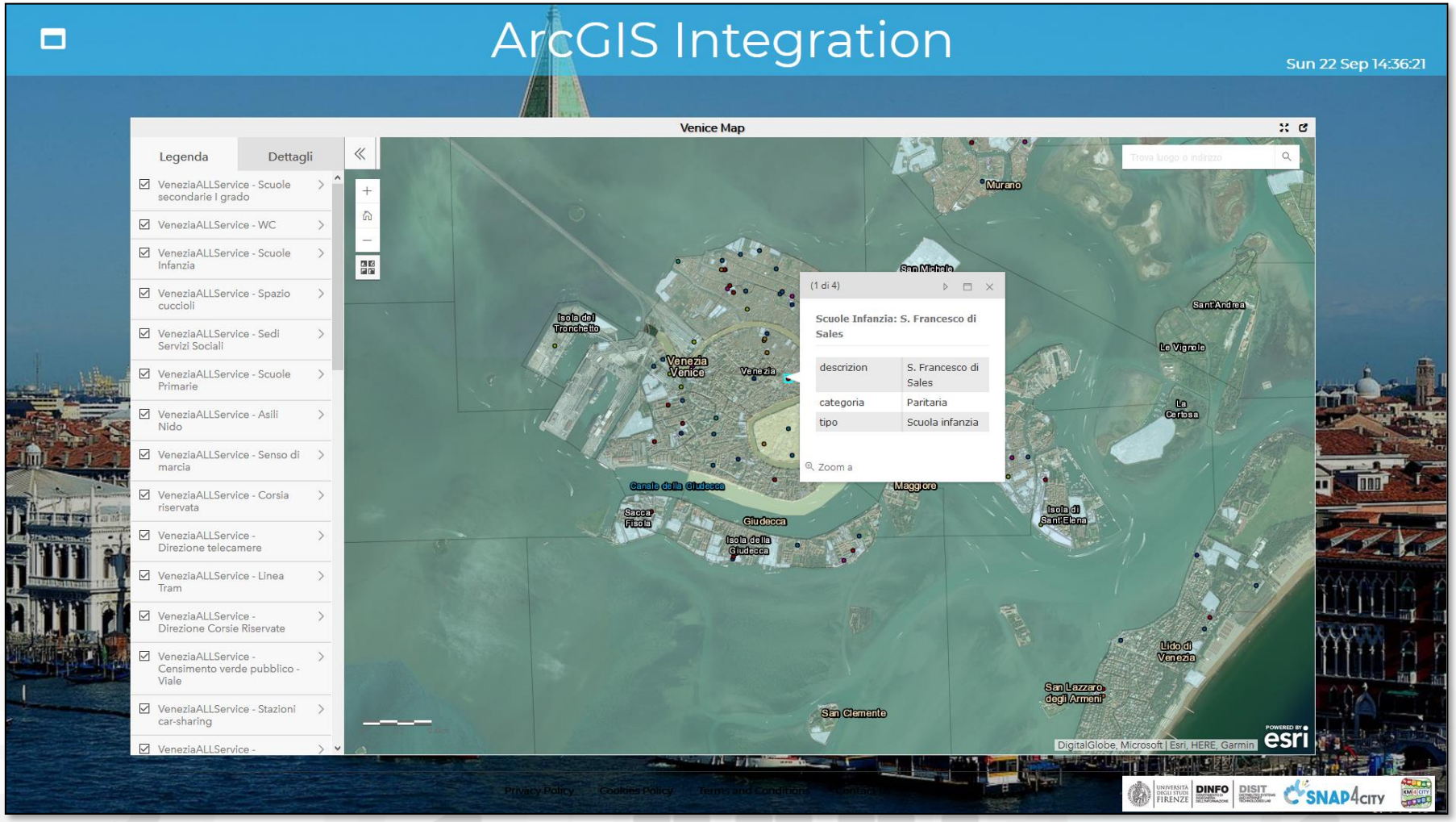
- **Snap4City is interoperable with**
  - ESRI ArcGIS Enterprise, Portal, Pro/MAP, ...
  - other GIS tools supporting WFS, WMS, GeoJSON, GML
- **Snap4City is interoperable since:**
  - **Provides** info/data in WFS, WMS
  - **Exploits** data/info from WFS, WMS
  - **Import** data/info from WFS/WMS



- The Snap4City platform can be installed on premise using **Snap4City Appliance** https://www.snap4city.org/drupal/node/471
  - **StartSNAP4CITYVM** includes the Dashboard Builder that is capable to work with WFS WMS protocols for the integration with GIS platforms as ESRI ArcGIS, QGIS, directly or using **Snap4City GIS player**.
  - **KBSSMVM** includes the Smart City API and WFS API which can be used to data harvest from any GIS servers and GIS desktop tool

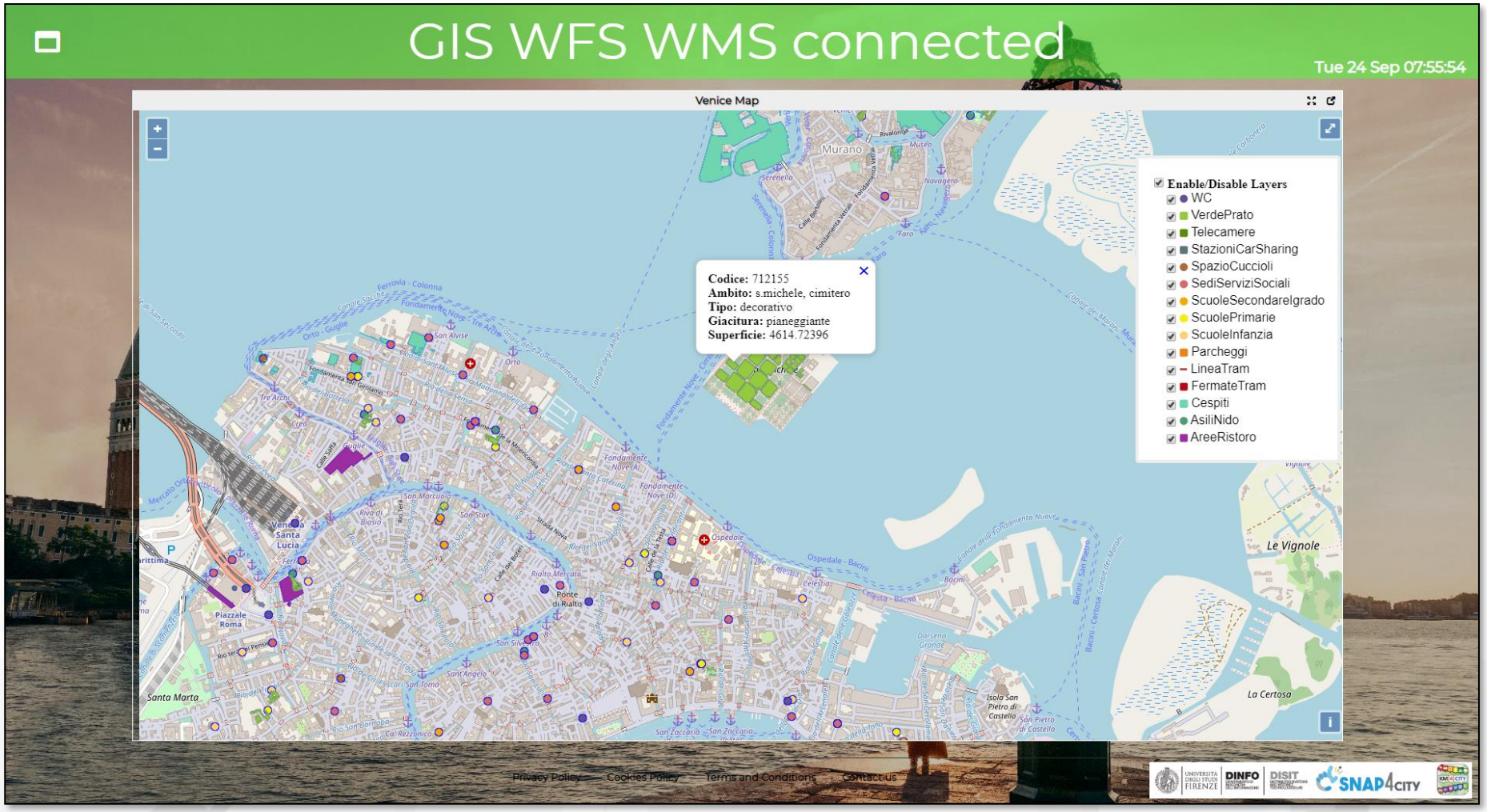
(A) ArcGIS ESRI as External Service

- DISIT Lab has ESRI ArcGIS Enterprise 10.6 installed
- ArcGIS Portal accesses to ArcGIS Enterprise server
  - https://arcgis.km4city.org/arcgisportal/apps/Embed/index.html?webmap=5774fa9d0f7f4a2fb2d7cd52dffcd8c9&extent=12.3144,45.4276,12.3461,45.4419&home=true&zoom=true&previewImage=false&scale=true&search=true&searchextent=true&details=true&legendlayers=true&active_panel=details&basemap_gallery=true&disable_scroll=true&theme=light
- Snap4City Dashboard uses as External Service: ArcGIS ESRI Portal
- https://main.snap4city.org/view/index.php?iddasboard=OTkw



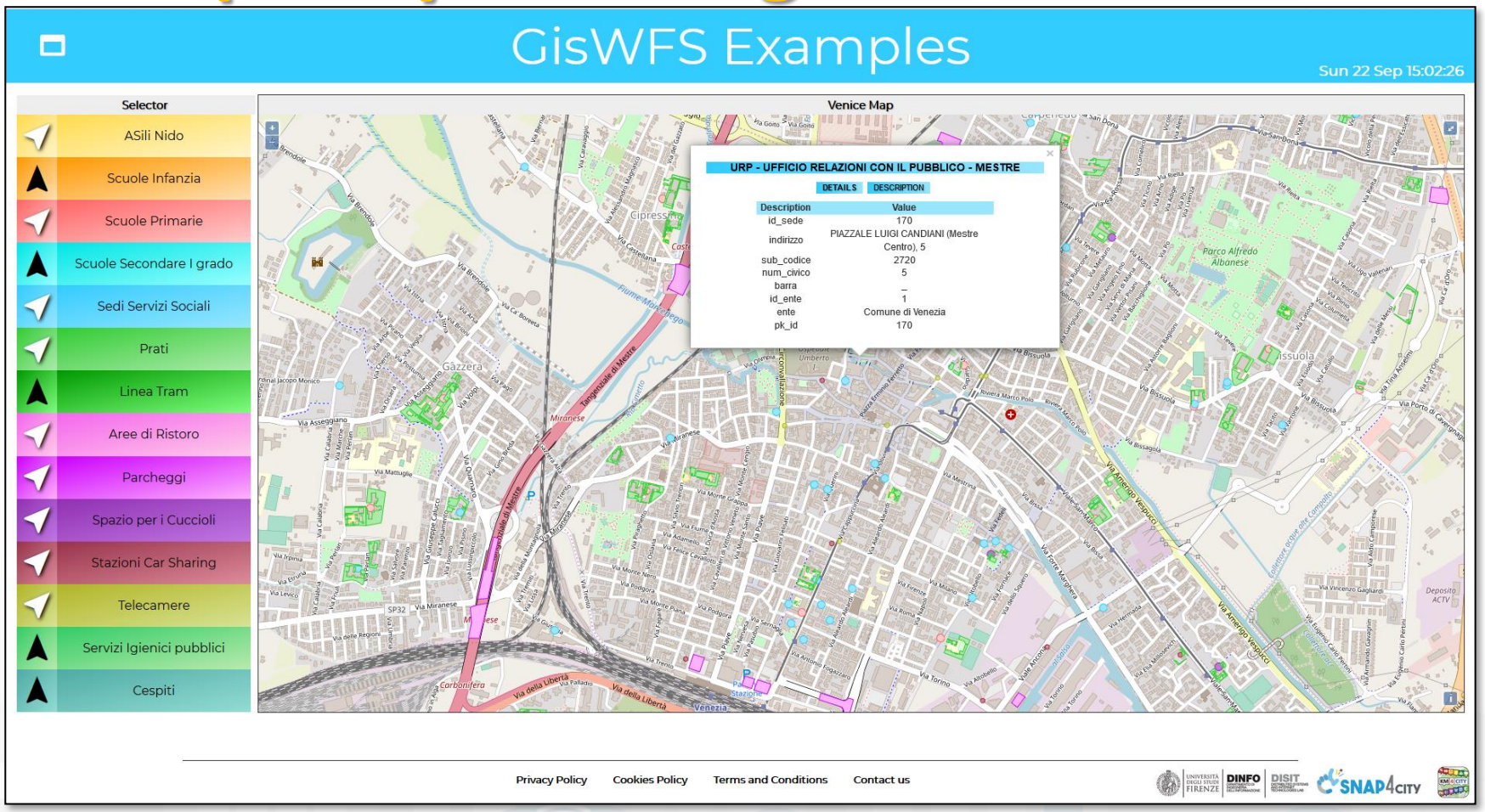
(B) GIS data on Dashboard via Snap4City GIS Player

- DISIT Lab has ESRI ArcGIS Enterprise 10.6 installed
- Snap4City has its WFS Player https://main.snap4city.org/widgets/venezia/index.php
- Snap4City Dashboard uses as External Service: Snap4City GIS viewer via WFS/WMS: https://main.snap4city.org/view/index.php?iddasboard=MTIxNg==



(C) Dash with Snap4City GIS widget and Selector

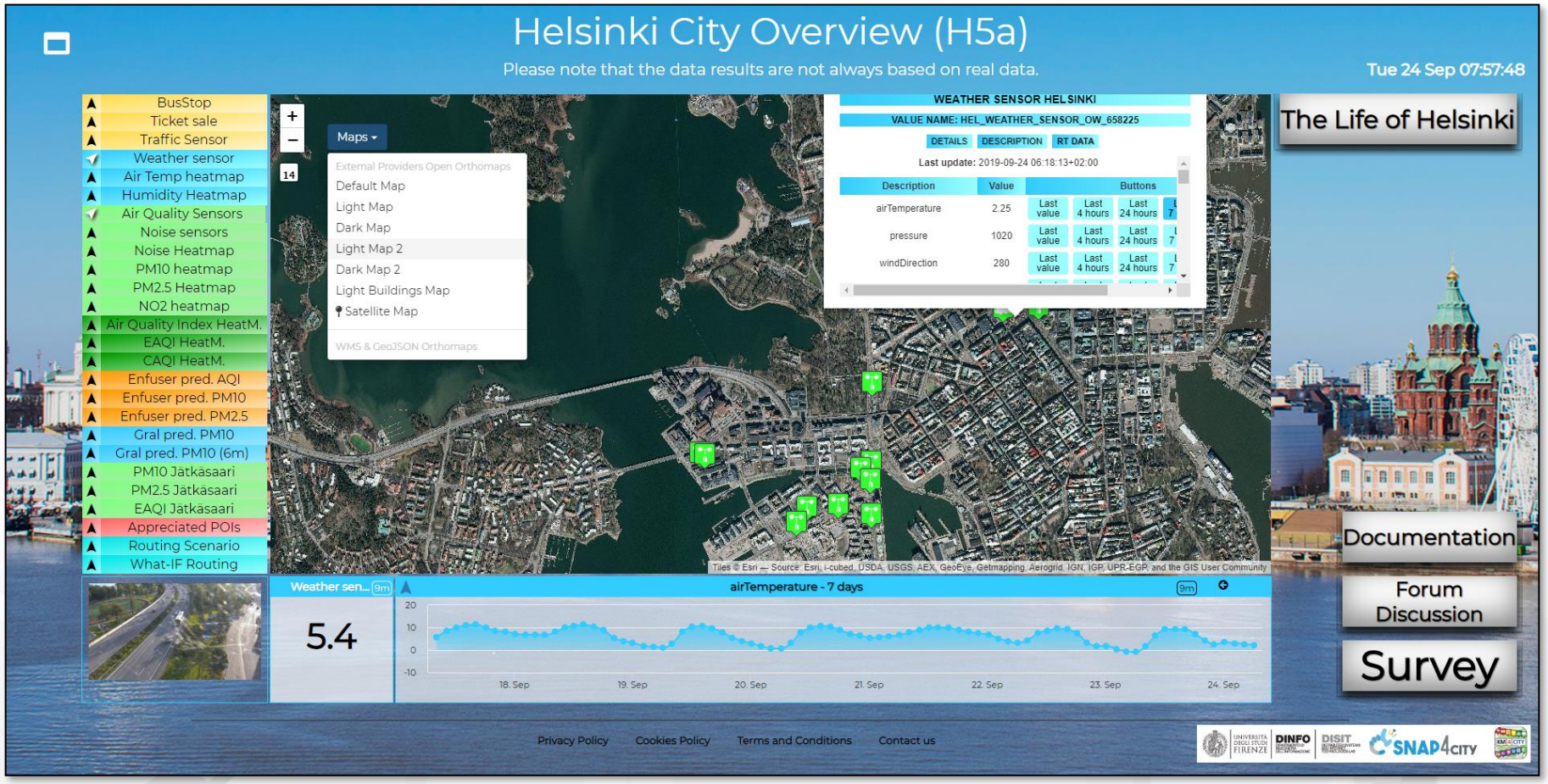
- DISIT Lab has ESRI ArcGIS Enterprise 10.6 installed
- Snap4City has its WFS / WMS widget / Player
- Snap4City Dashboard shows WFS/WMS data via Special GIS Widget Map:
  - https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTQwMw==
- Snap4City can use **Selector** to select WFS / WMS sources to be shown from ESRI ArcGIS (as well as from any other WFS service) on Widget map



The Snap4City Widget Map allows to **mixt WFS GIS sources with Smart City API**
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTM5NA==

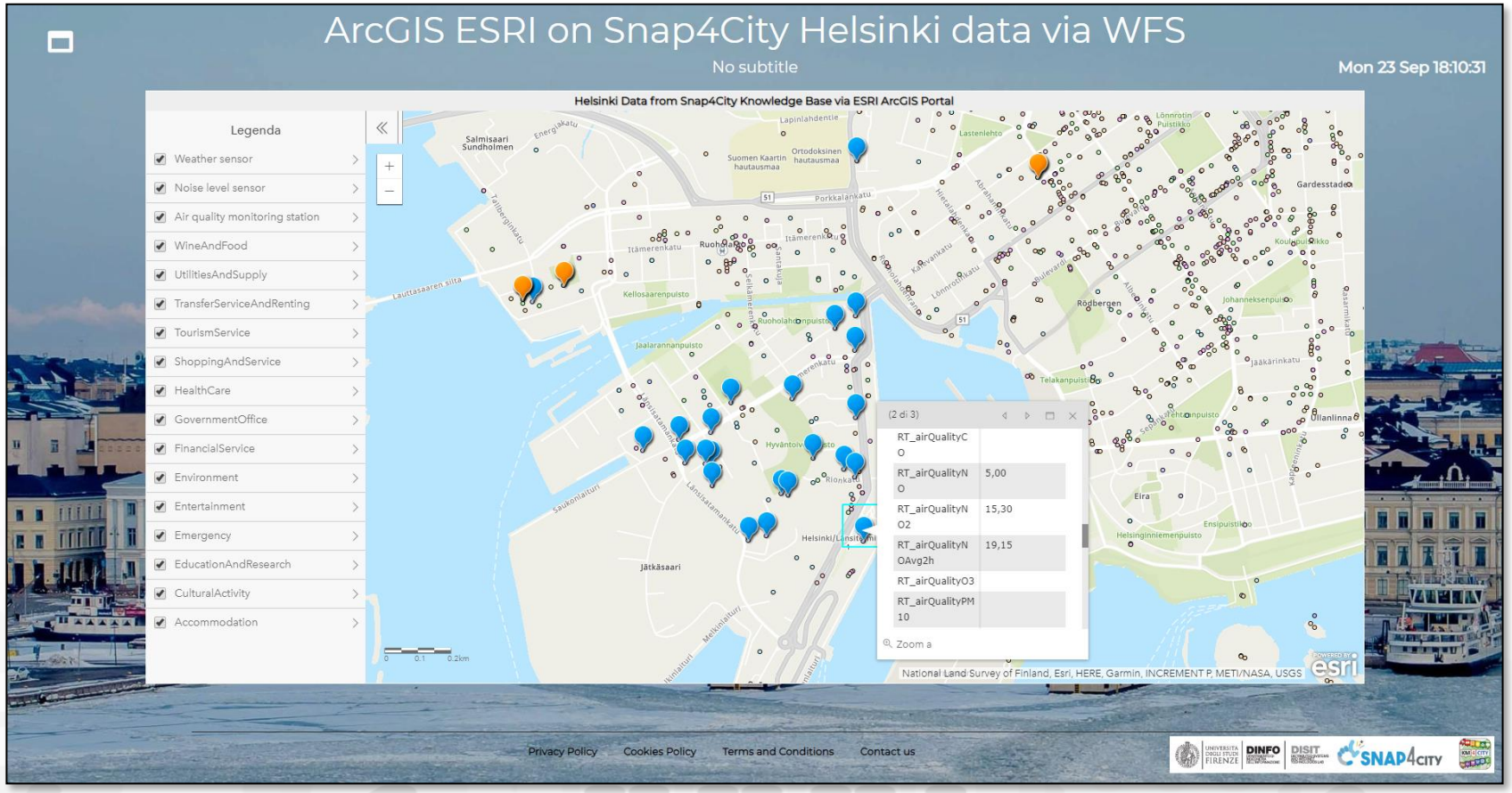
(D) Dashboard with Orthomaps and shapes layers based on WMS, GeoJSON

- DISIT Lab has ESRI ArcGIS Enterprise 10.6 installed, and GeoServer
- Snap4City main MultidataMap Widget can load WMS background images and shapes, the Orthomaps
- Also Heatmaps calibrated are provided from GeoServer using WMS protocol
- Maps in this case are directly **taken from ESRI Server, free of charge**
- https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTQwNg==#



(E) Snap4City via WFS on top of Smart City API provide data to ESRI ArcGIS Enterprise or ArcGIS Pro, and thus the data become accessible on ArcGIS ESRI Portal

- Snap4City provides a WFS service on top of SmartCity API. Thus providing data to any GIS/WFS client, including ESRI ArcGIS Enterprise, ArcGIS Pro, QGIS, etc.
- In the example, our ingested Helsinki Data have been harvested from ESRI ArcGIS via WFS.
- Once ingested on ESRI ArcGIS can be visualized, by using ArcGIS Portal
  - https://arcgis.km4city.org/arcgisportal/apps/Embed/index.html?webmap=fbcec7055a3446e29fc1e9c55cb53d79&extent=24.8335,60.1465,25.013,60.1965&zoom=true&previewImage=false&scale=true&legendlayers=true&disable_scroll=false&theme=light
- Snap4City Dashboard can show ESRI ArcGIS Portal **(A)** as External Services in a dashboard.
  - https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjIwNg==



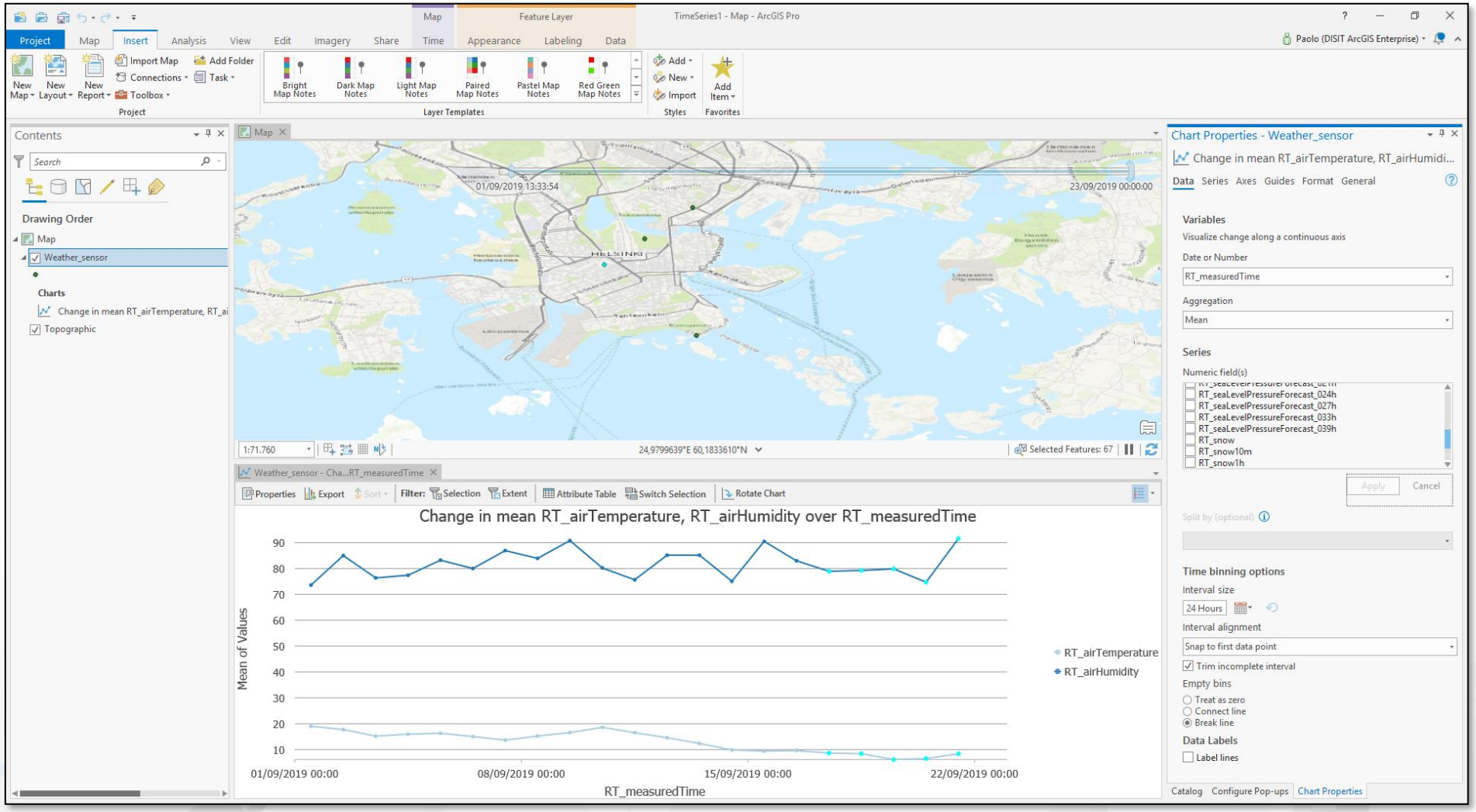
(EA) ArcGIS ESRI Pro as GIS / WFS play

To test you need to have installed ArcGIS pro on your pc and connect with Snap4City WFS/WMF server or with ArcGIS server which has done the same connection with our server WFS/WMS

On PC:

(i) Get data via WFS connected to ESRI ArcGIS Enterprise or other sources

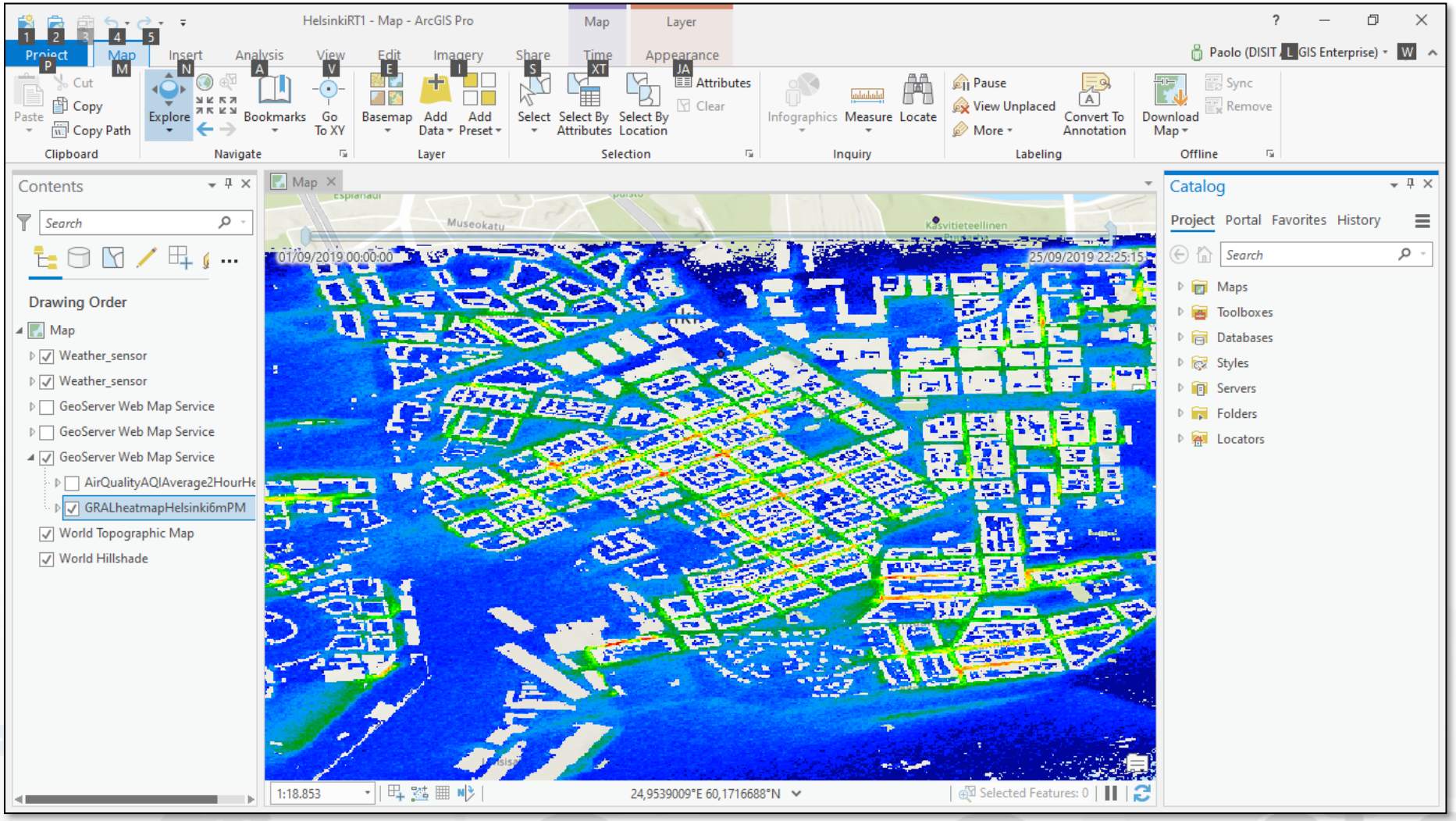
(ii) Download data via WFS … from…



(EA) ArcGIS ESRI Pro as GIS / WMS play

To test it you need to have installed ArcGIS pro on your pc and connect with our WFS/WMF server or with ArcGIS server which has done the same connection with our server WFS/WMS

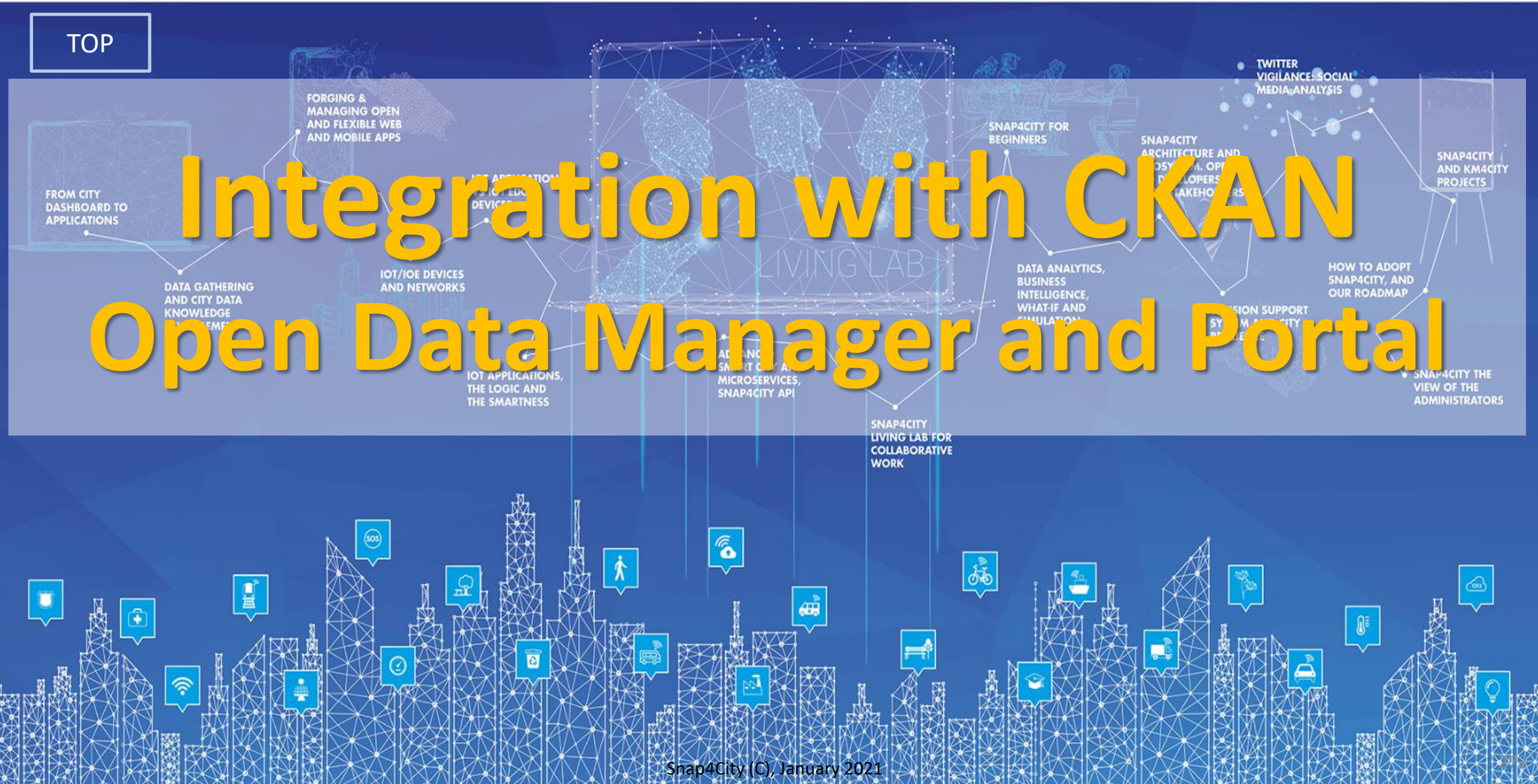
Heatmap taken from Snap4City GeoServer via WMS protocol



Snap4City vs GIS, WFS/WMS

- GIS data:
  - Ingested via WFS/WMS protocols, and then managed as the other data. Data ingestion from GIS server can be performed via ETL processes, or directly from Dashboards
  - Shown on Dashboards via third party GIS tools as External Services
  - Shown on Dashboards using Special GIS Widget Map which directly access to GIS data via WFS/WMS
  - Heatmaps and Maps are distributed via a GeoServer
- Snap4City can interact with ArcGIS Real Time Events via MQTT protocol as well
- **Snap4City vs GIS solutions and connections**

TOP



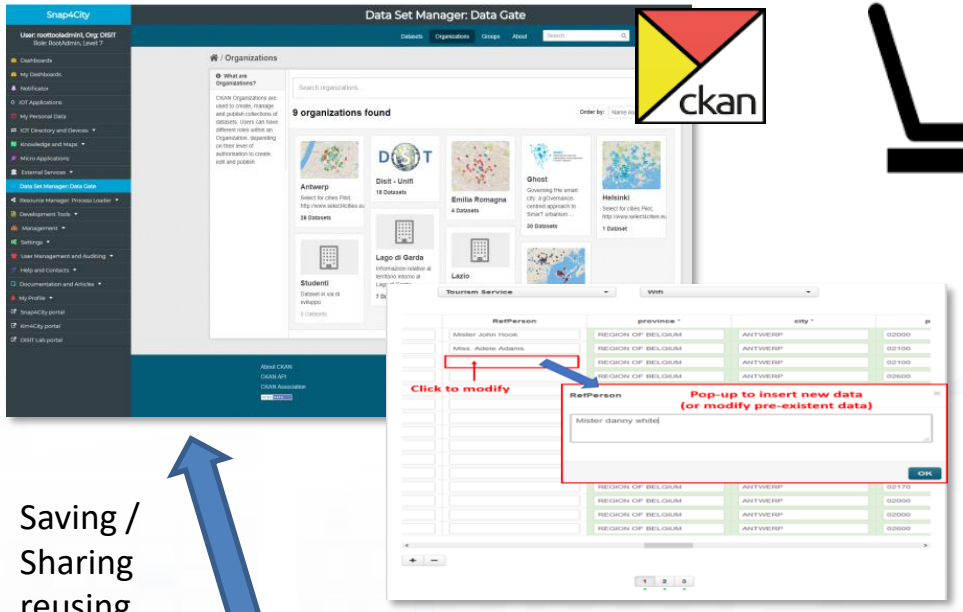
Integration with CKAN Open Data Manager and Portal

Data Ingestion via Data Gate



Knowledge and Storage Data from the Field and City

Data Gate





Data Set:

- Search
- Loading
- Download
- Share
- Publish
- Also automated

Saving / Sharing reusing

Federated Network

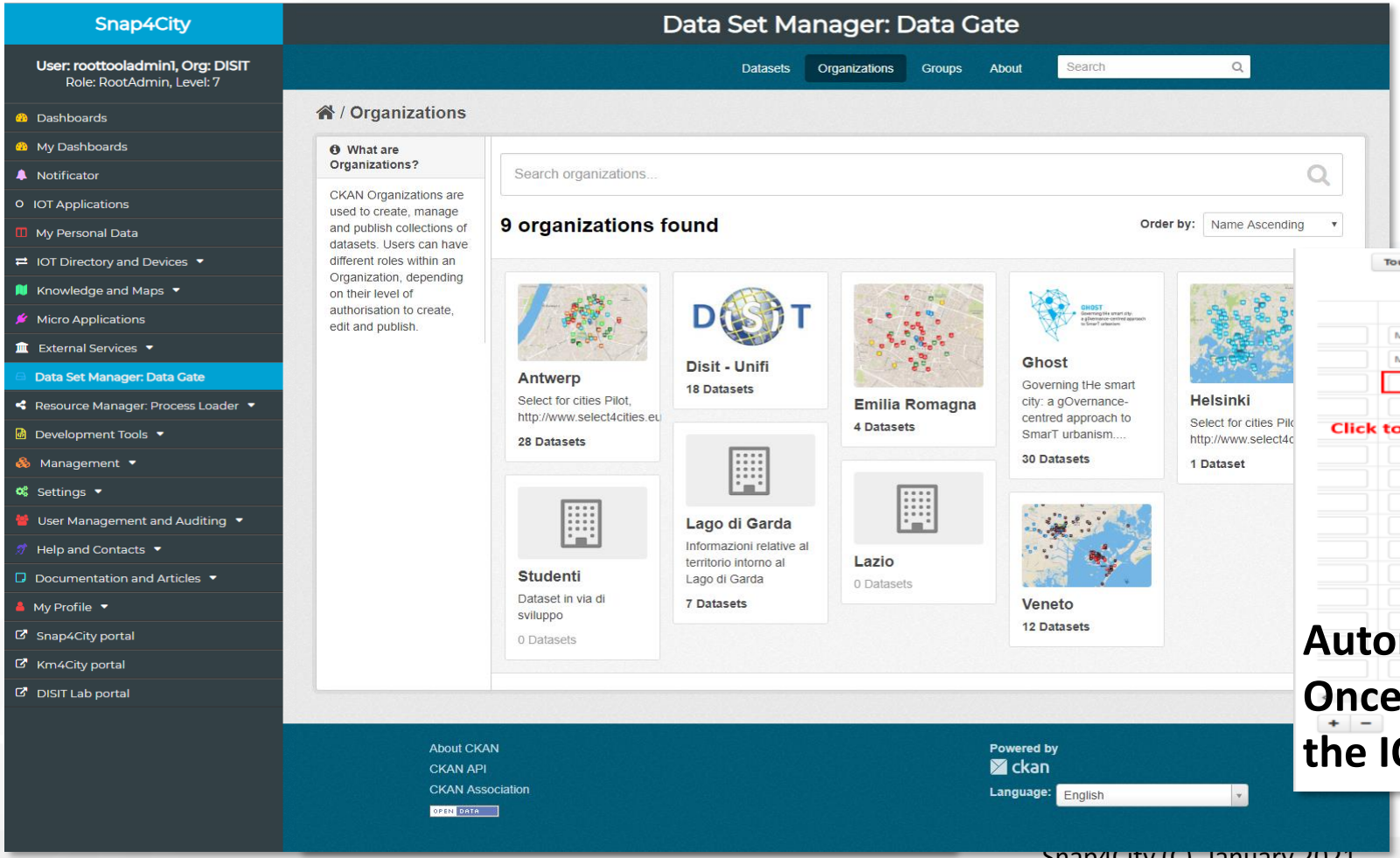
Data Set: sharing, Harvesting, Loading/ Downloading

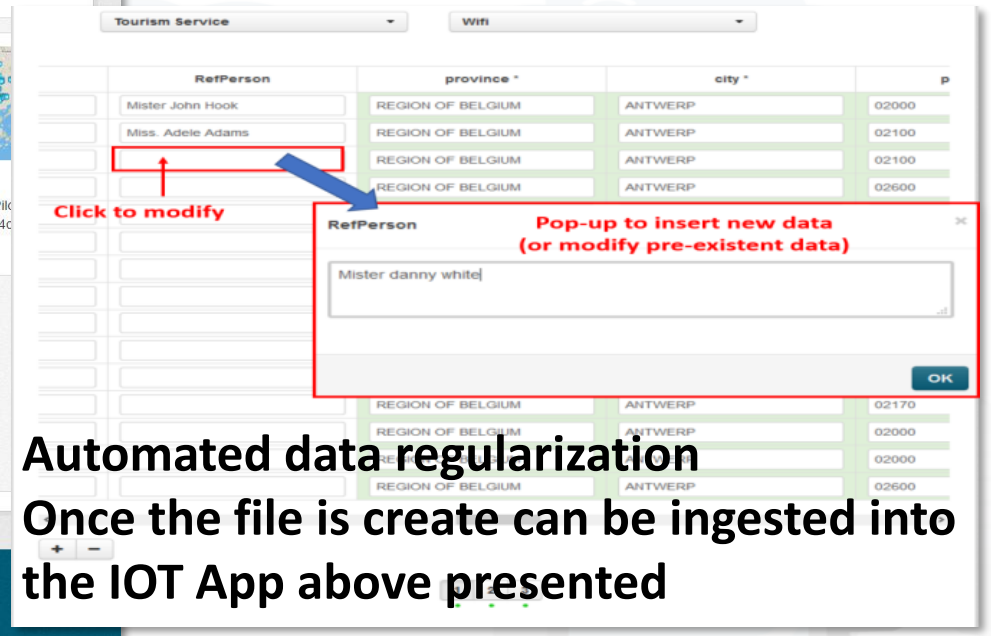
https://datagate.snap4city.org/ssologin_handler

Integrated DataGate/CKAN
Static open data ingestion

Federated Crawling
Federated Distribution

Data Set:
- Search
- Loading
- Download
- Share
- Publish
- Also automated





Automated data regularization
Once the file is create can be ingested into the IOT App above presented

https://datagate.snap4city.org/ssologin_handler

Further readings on Datagate

- Data Set Manager: Data Gate / CKAN federated
- HOW TO: add a device to the Snap4City Platform
- HOW TO: add data sources to the Snap4City Platform
- TC6.1- Managing DataSets via DataGate: ingest, search, download, upload, annotate, share
- TC9.3- Managing data sets with IOT Applications, and exploiting DataGate
- TC6.2- Search on DataGate for Data Sets
- US6. Developing and using processes for data transformation
-

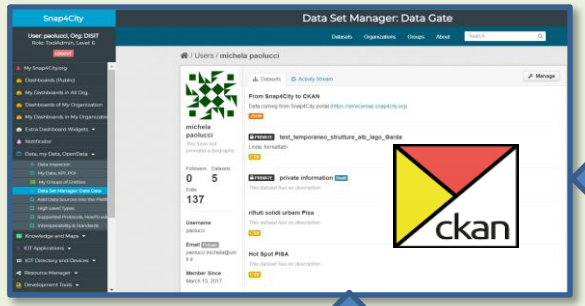
Integration with CKAN: automated data set (i) ingestion and (ii) production via IOT App



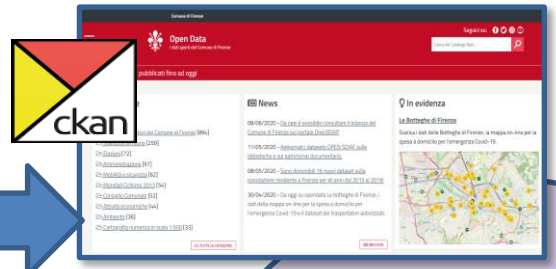
Snap4City vs CKAN

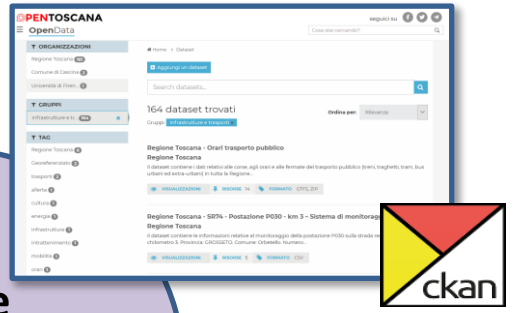
Snap4City Portal and Integrated tools

Datagate



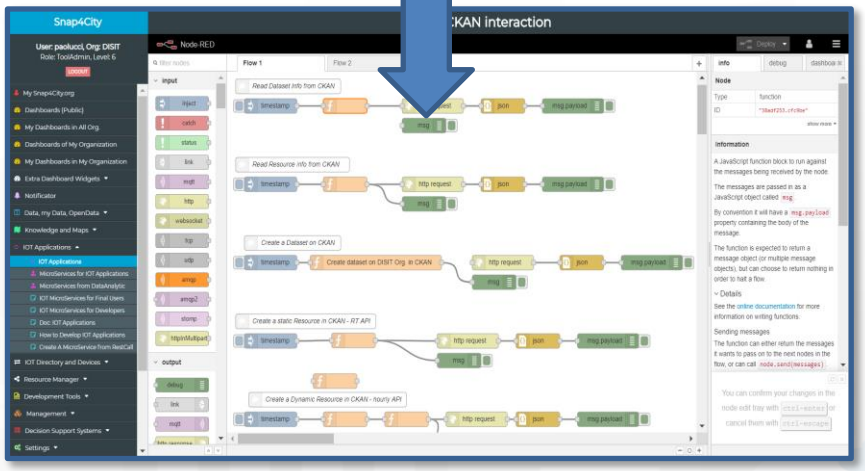
Harvesting and Publishing

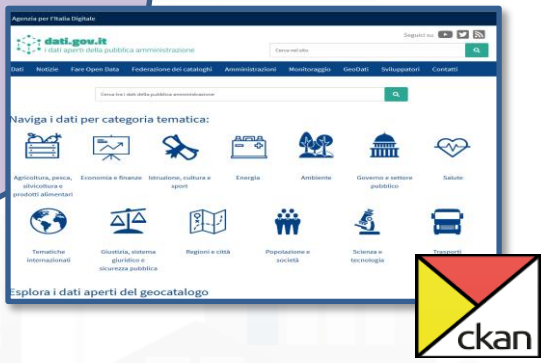




Open or Private External CKAN Data Portals

Advanced Snap4City APIs and Micro Services

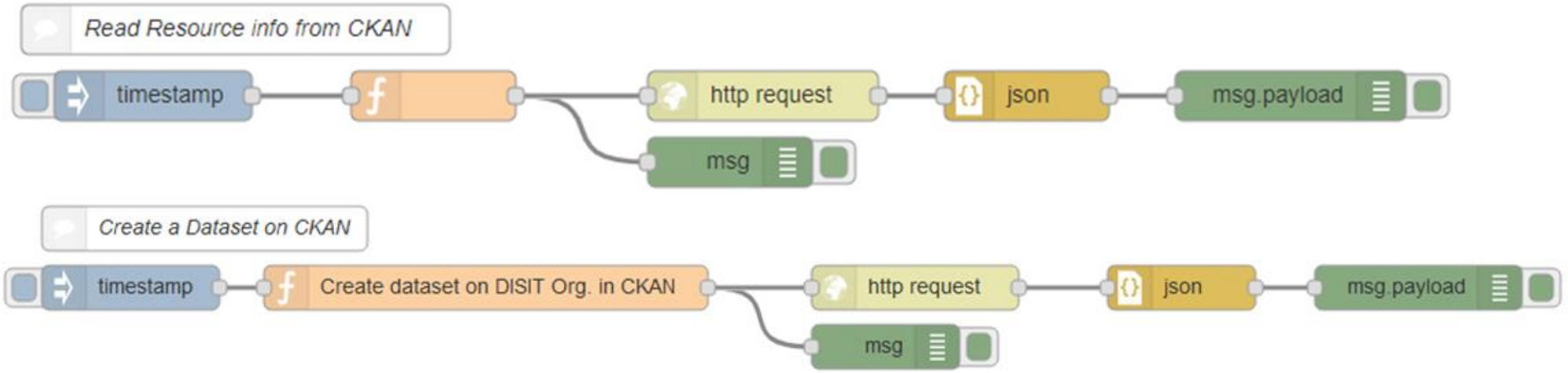




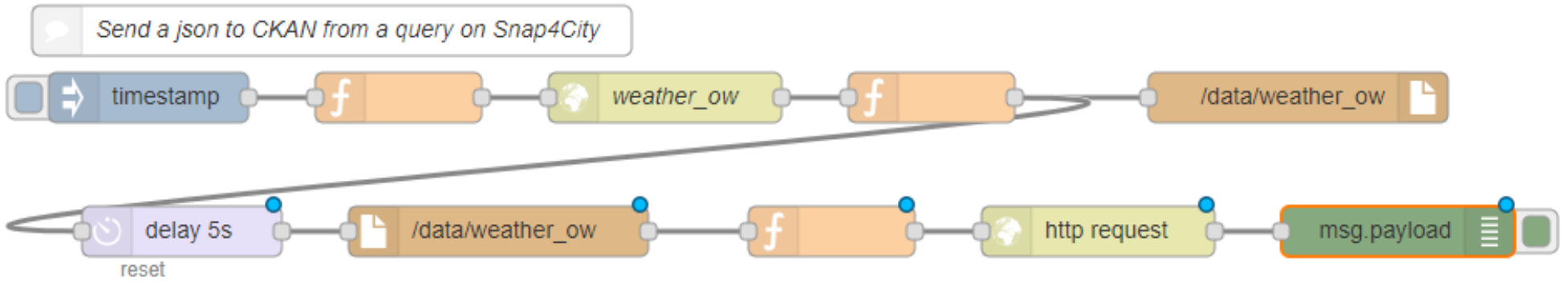
Automatize:

- Import data from CKAN to Snap4City
- Upload Public Data from Snap4City to CKAN
- Data Harvesting
- Dashboards and Mobile/Web Apps creation

Some IOT App segements



Almost all the calls to CKAN are quite similar

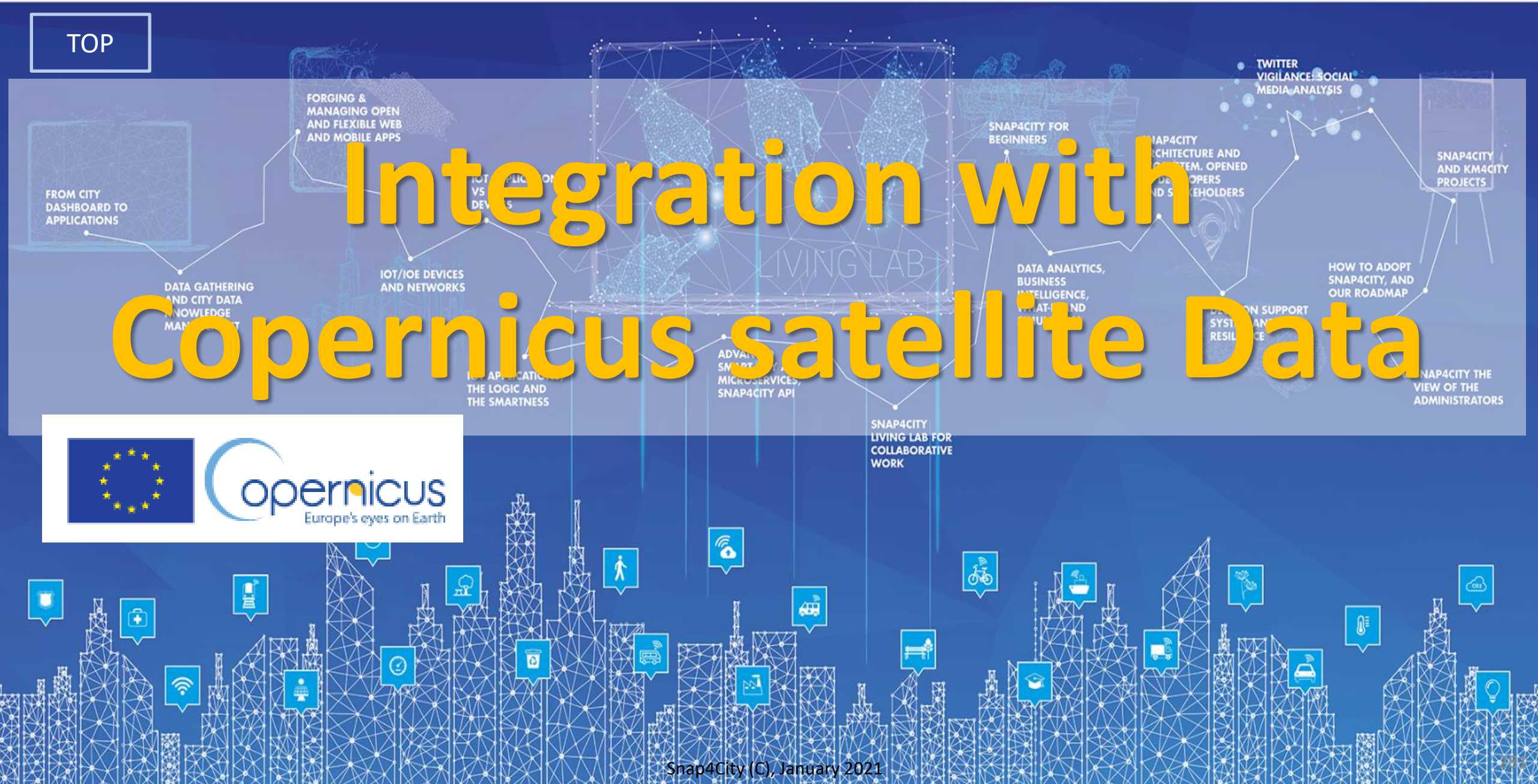


Read more on

- **TC9.17 – CKAN vs Snap4City Integration and Interaction**
  - automating the *Read of a Dataset Info from CKAN*
  - automating the *Read of a Resource info from CKAN*
  - automating the *Creation of a Dataset on CKAN*
  - automating the *Creation of a static Resource in CKAN*
  - automating the *Creation of a dynamic Resource in CKAN*
  - automating the *Sending of a json to CKAN from a query to Snap4City to perform any other action on the Smart City*
- **Data Set Manager: Data Gate / CKAN federated**

TOP

Integration with Copernicus satellite Data

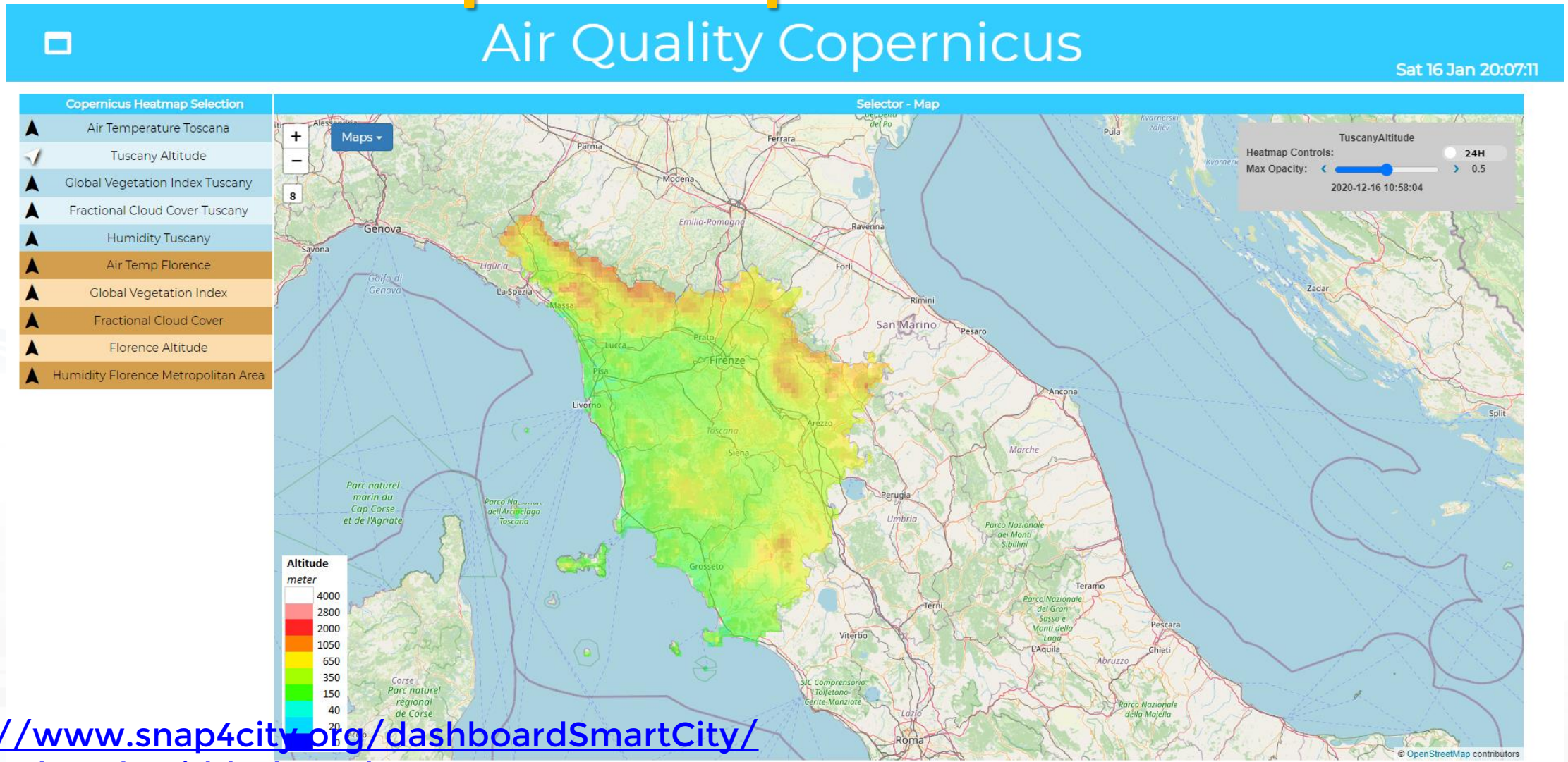


Copernicus



https://www.copernicus.eu/en/copernicus-satellite-data-access

Example of Copernicus Data

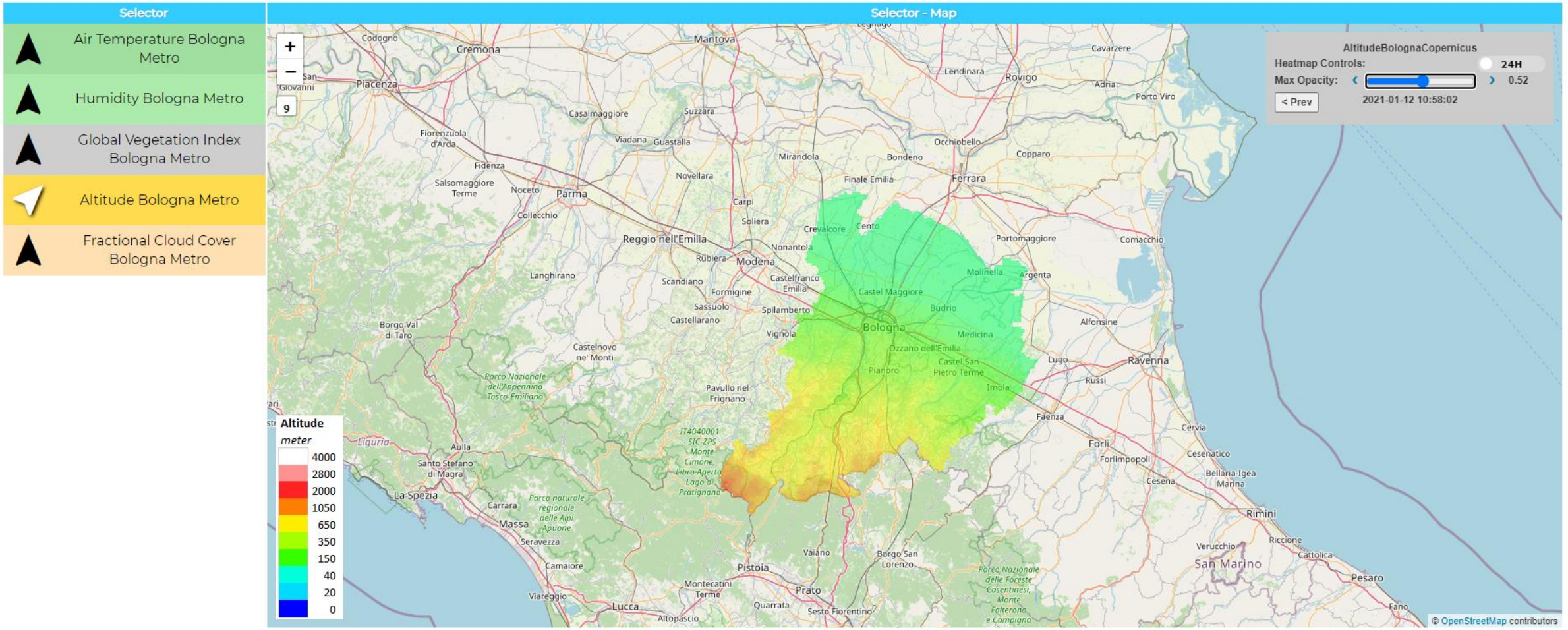


https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MzAwNQ==

Bologna Metro

Bologna Metropolian Area Copernicus Data

Sat 16 Jan 20:08:03



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

TOP



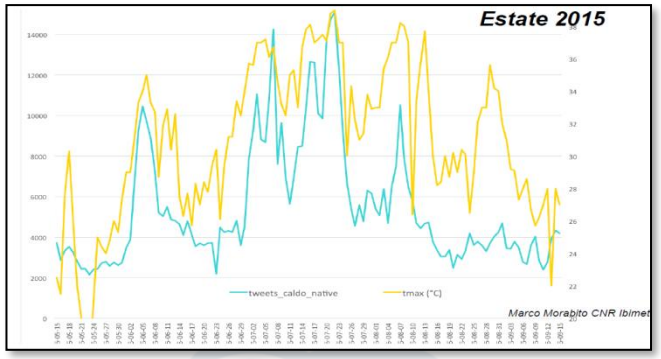
Social Media data collection and exploitation

Social Media Data

- Data from Social Media can be ingested in different manners:
- Twitter Data can be:
  - Collected via Twitter Vigilance programming keywords.
  - Colelcted in real time via IOT Applications
  - Once collected the Metrics (number of Tweets, reTweets, NLP, Sentiment Analysis, etc. can be:
    - Shown, exploited in IOT Applications
    - Exploited for early warning
- FaceBook data:
  - Can be generated by using IOT App
- Telegram data:
  - Can be generated by using IOT App

Prediction/Assessment

- Football game results as related to the volume of Tweets
- Number of votes on political elections, via sentiment analysis, SA
- Size and inception of contagious diseases
- marketability of consumer goods
- public health seasonal flu
- box-office revenues for movies
- places to be visited, most visited
- number of people in locations like airports
- audience of TV programmes, political TV shows
- weather forecast information
- Appreciation of services



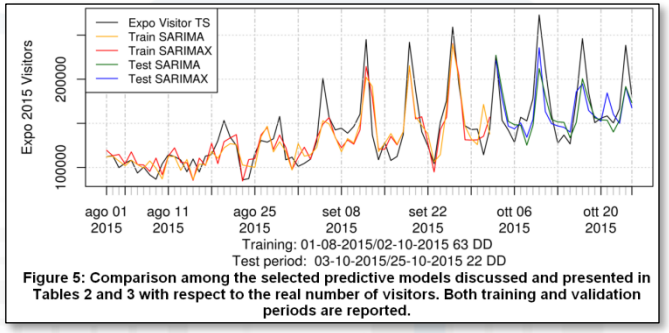
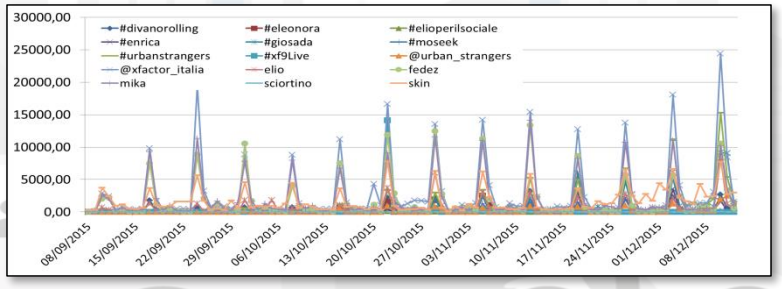
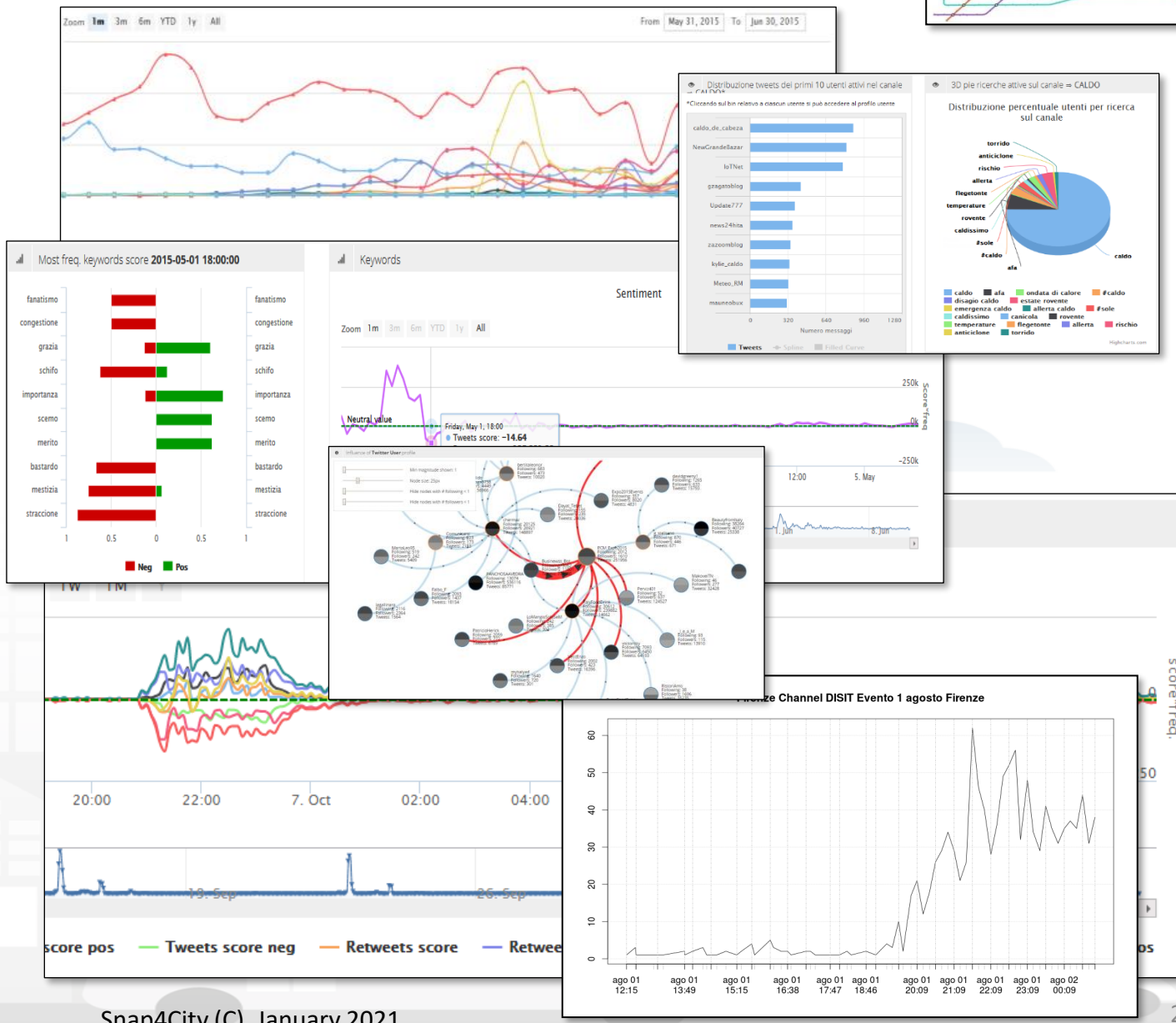


Figure 5: Comparison among the selected predictive models discussed and presented in Tables 2 and 3 with respect to the real number of visitors. Both training and validation periods are reported.



Twitter Vigilance

- http://www.disit.org/tv
- http://www.disit.org/rttv
- Citizens as sensors to
  - Assess sentiment on services, events, …
  - Response of consumers wrt, …
  - Early detection of critical conditions
  - Information channel
  - Opinion leaders
  - Communities
  - Formation
  - Predicting volume of visitors for tuning the services



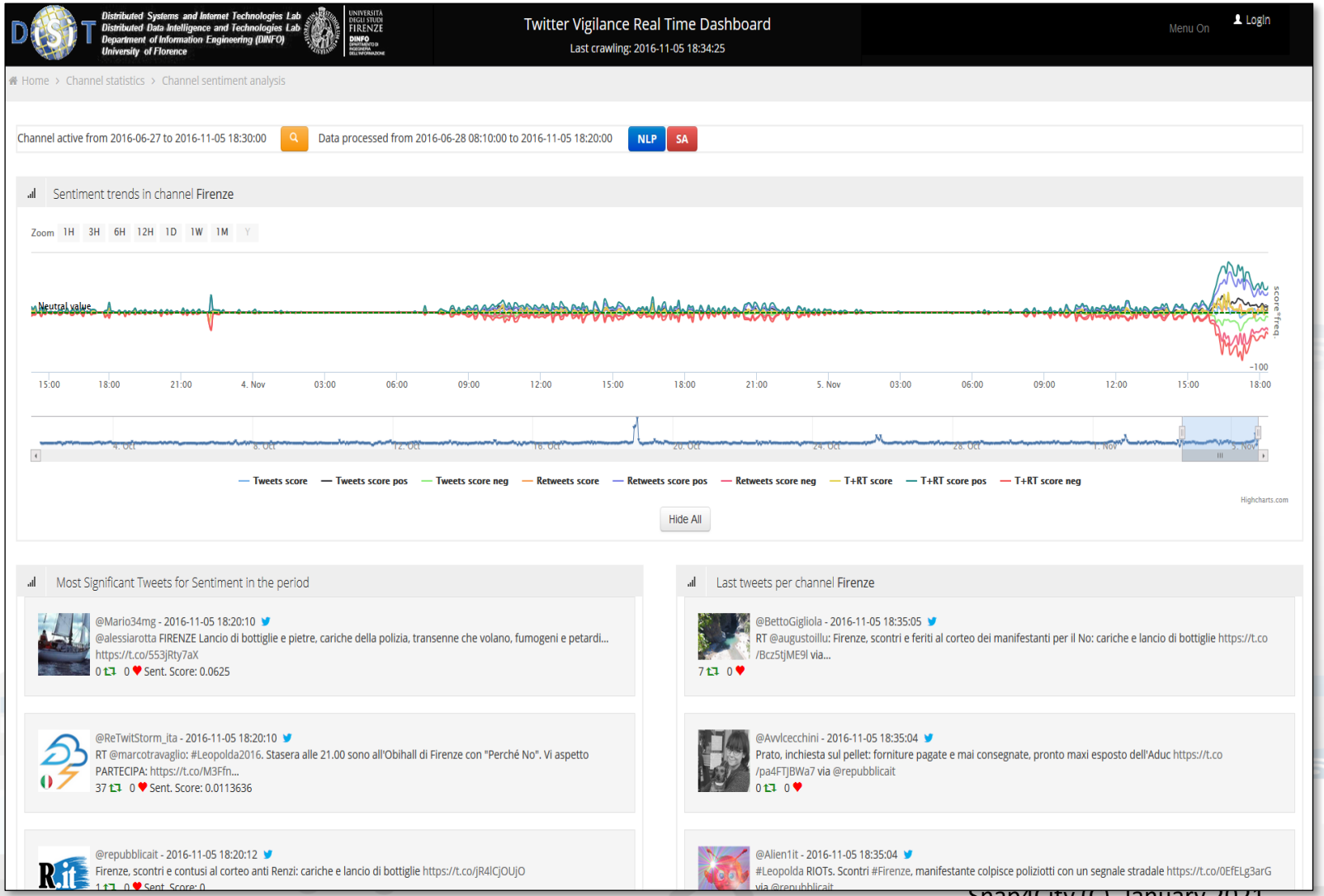
Twitter Vigilance

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

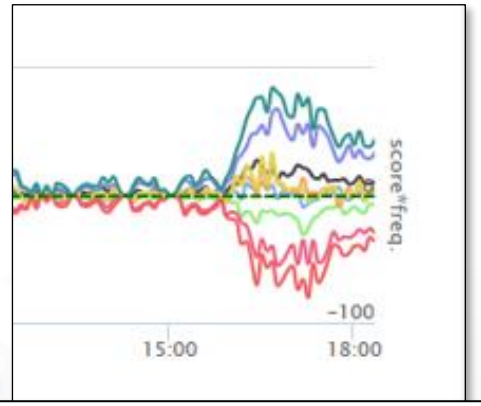
Twitter Vigilance

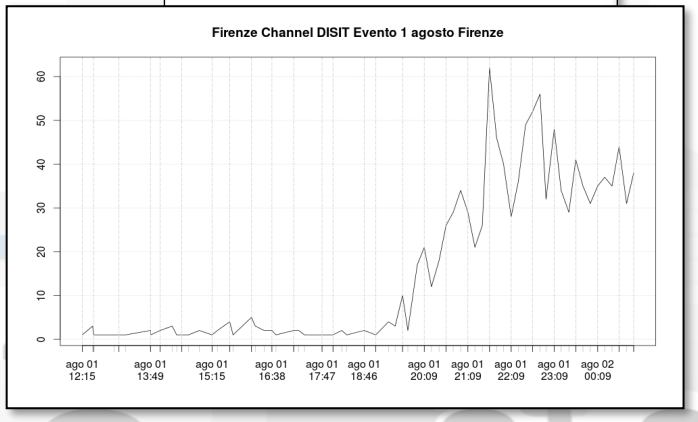
resolute

Twitter Vigilance RT: sentiment analysis

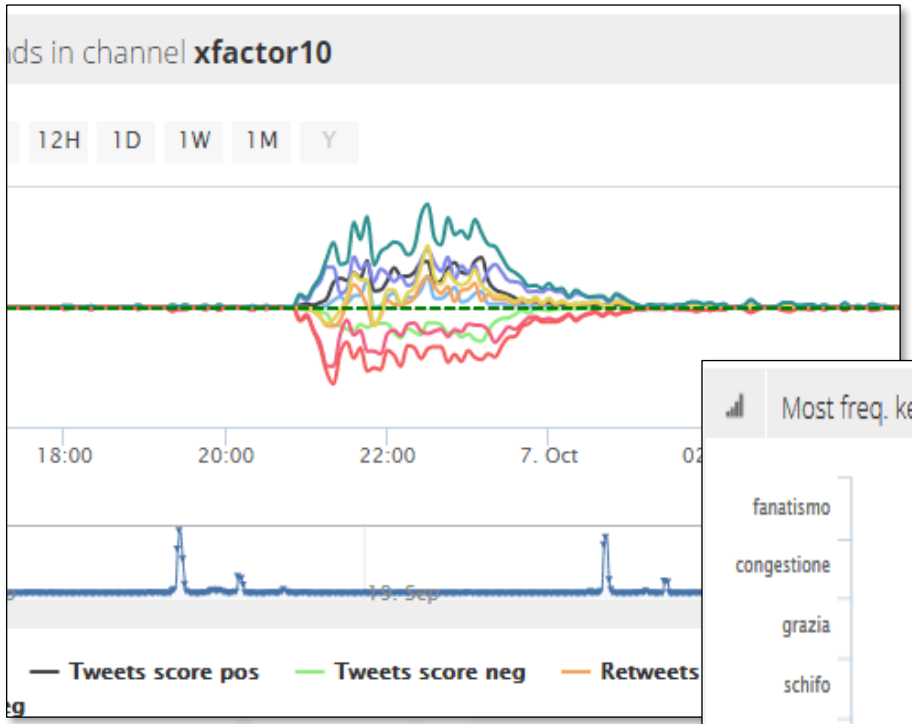


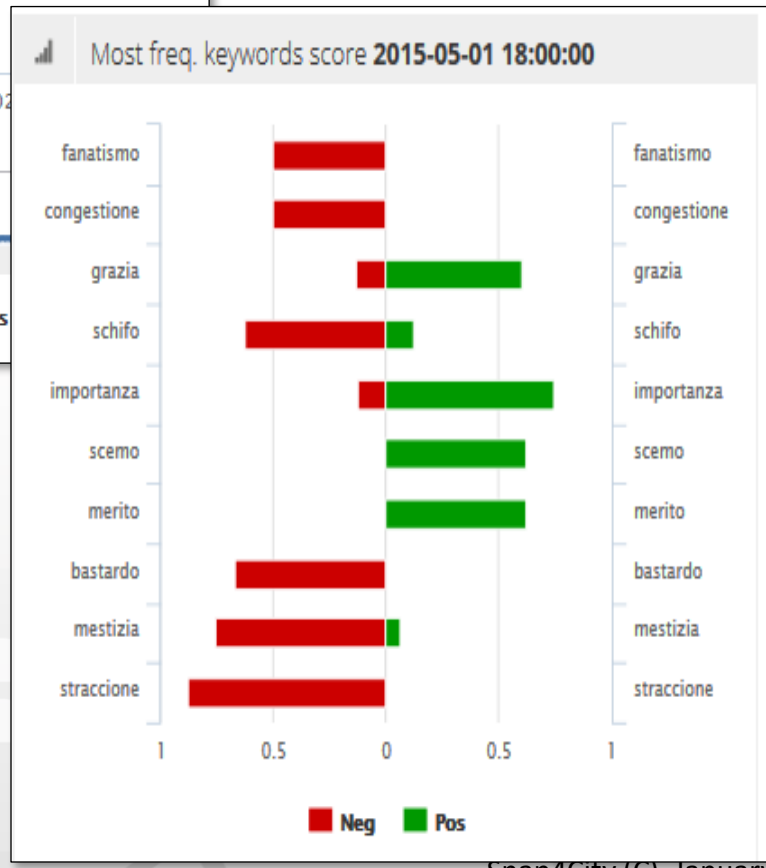
Real time
Early Warning

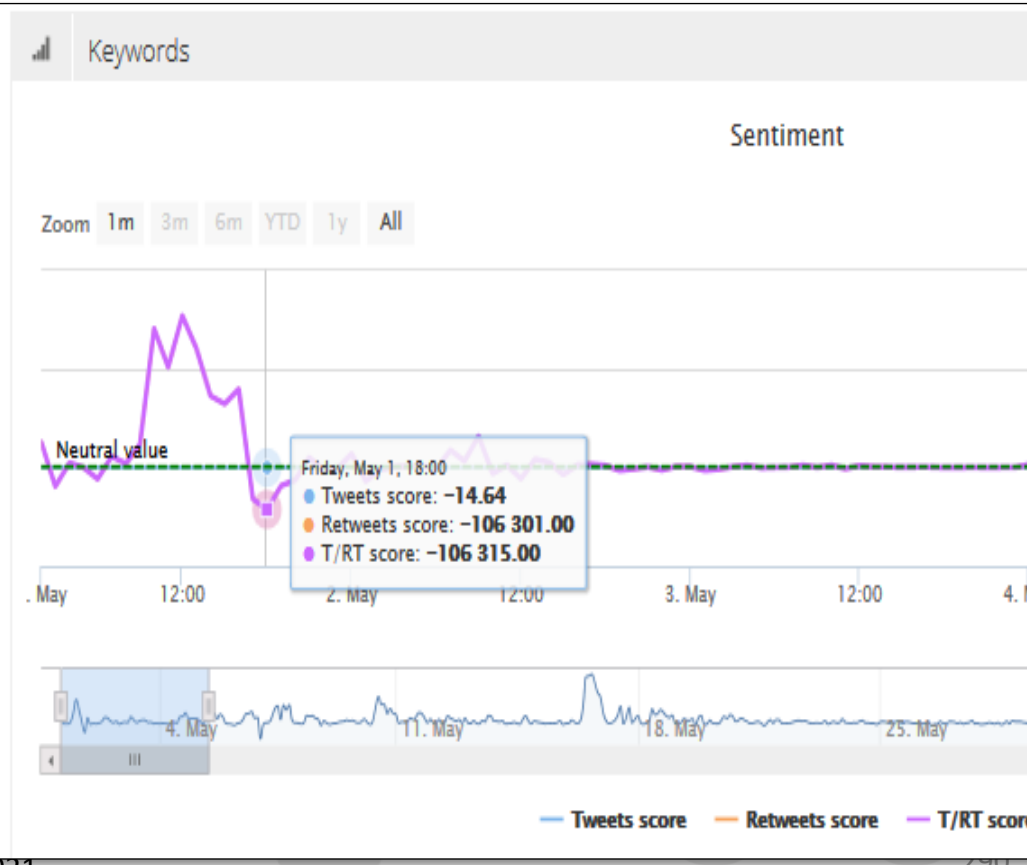




Sentiment Analysis

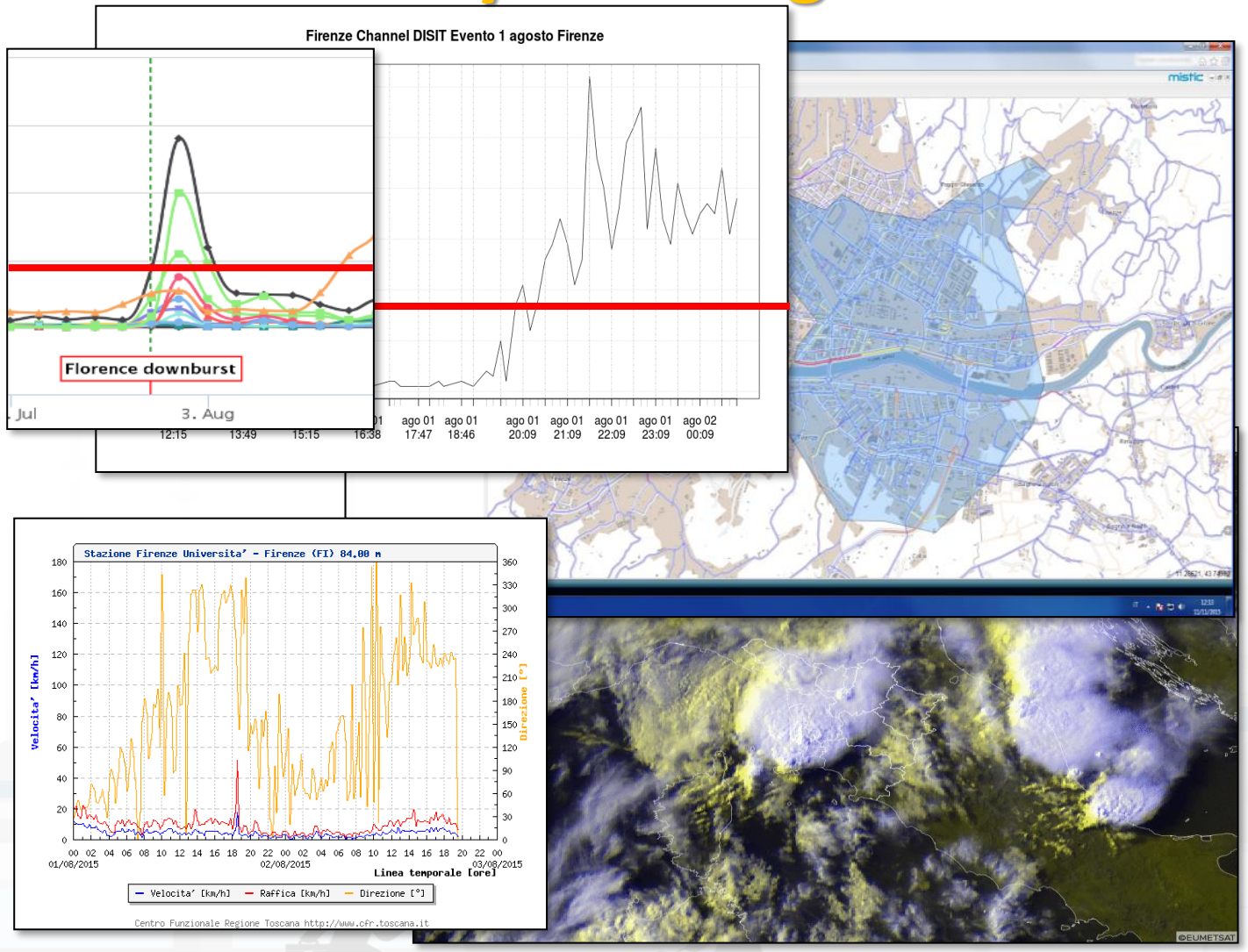




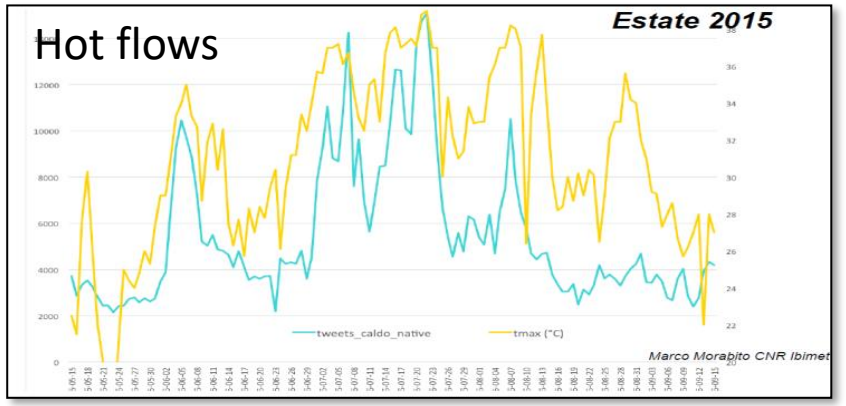


Twitter Vigilance

Early Warning



Predictive models



Attendance at long lasting events: EXPO2015

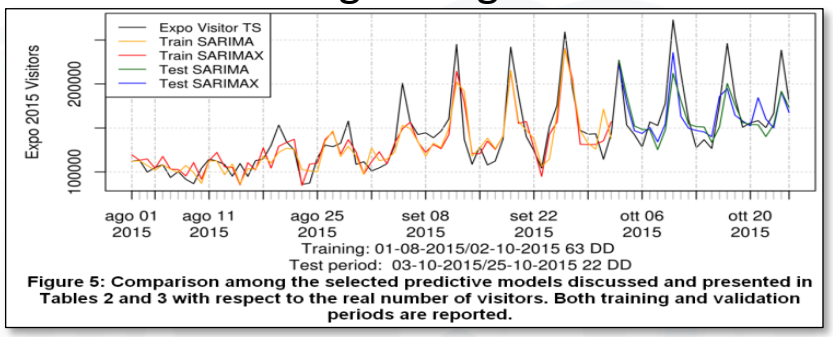
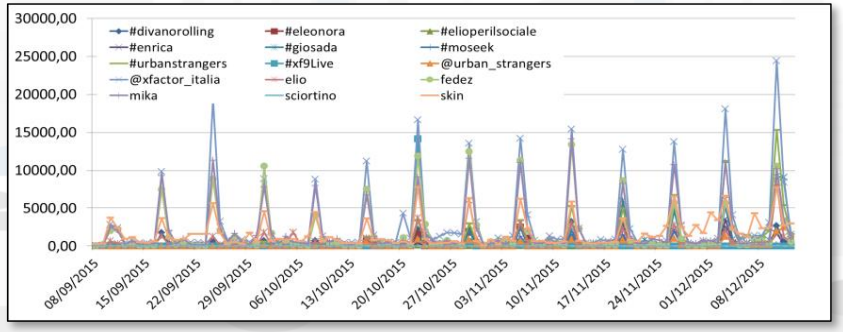


Figure 5: Comparison among the selected predictive models discussed and presented in Tables 2 and 3 with respect to the real number of visitors. Both training and validation periods are reported.

Attendance at recurrent events: TV, footbal

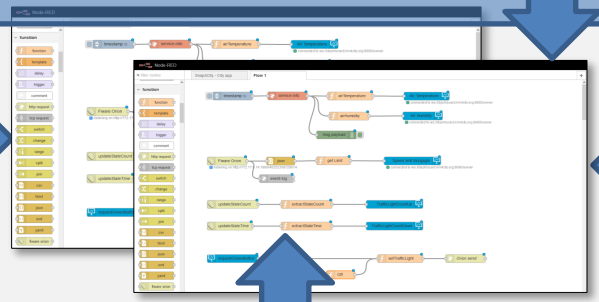


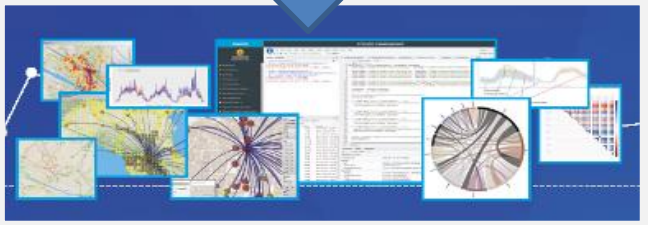
Twitter Vigilance

IOT and data World

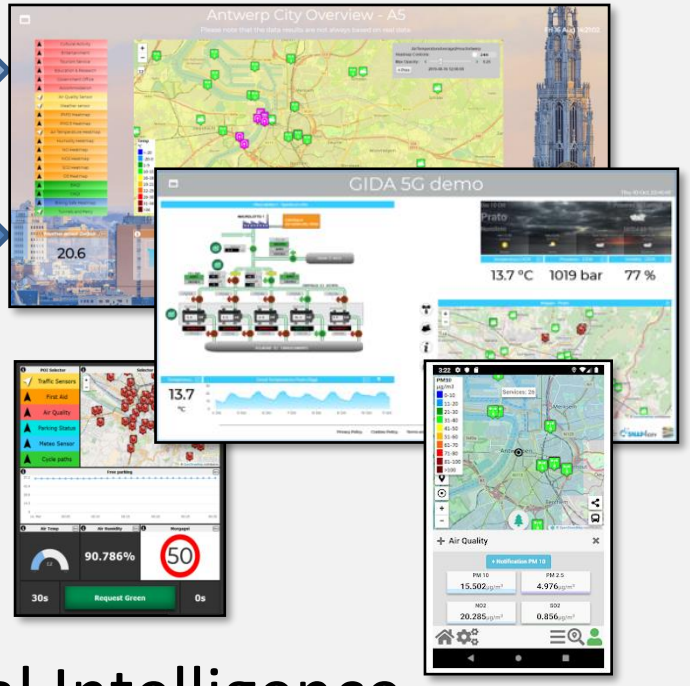


IOT Applications





Dashboards and Apps



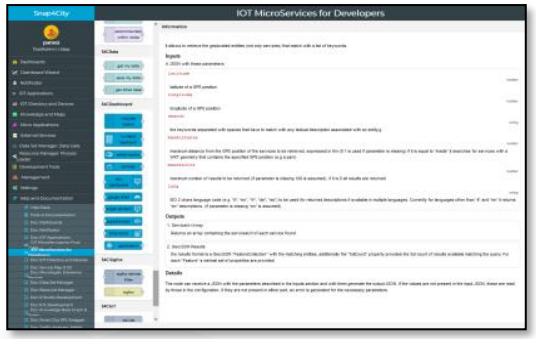
Big Data Analytics, Artificial Intelligence

Paolo

IOT Applications Development

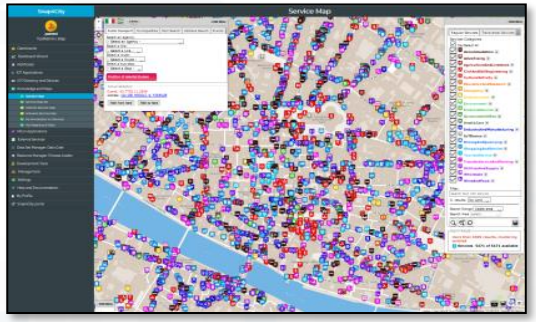
IOT Discovering

MicroServices collections

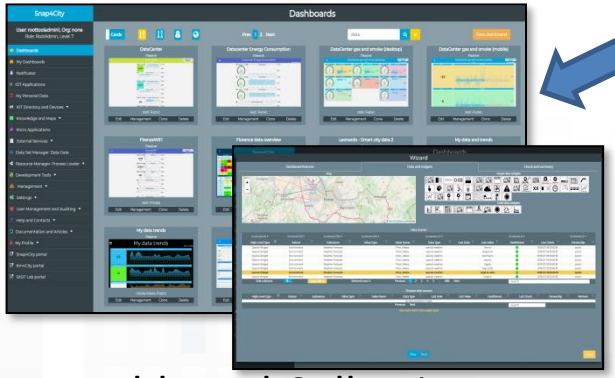


My IOT Applications



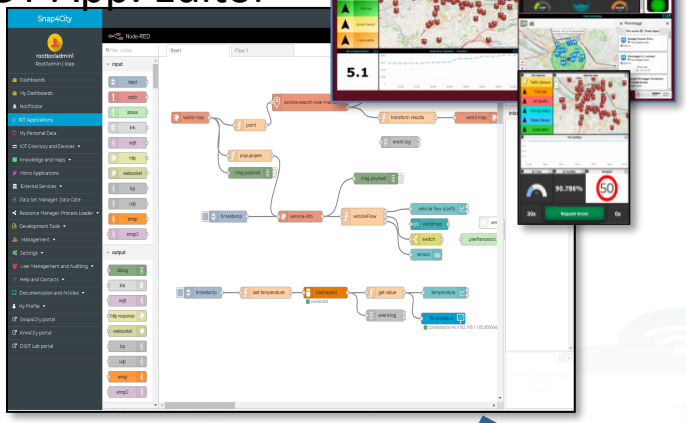


ServiceMap Discovery



Dashboard Collection, Editor and Wizard

IOT App. Editor





Generating IOT App With Dashboard

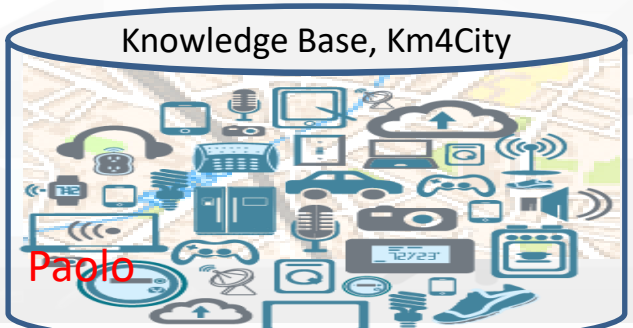


Sharing/saving reusing IOT App



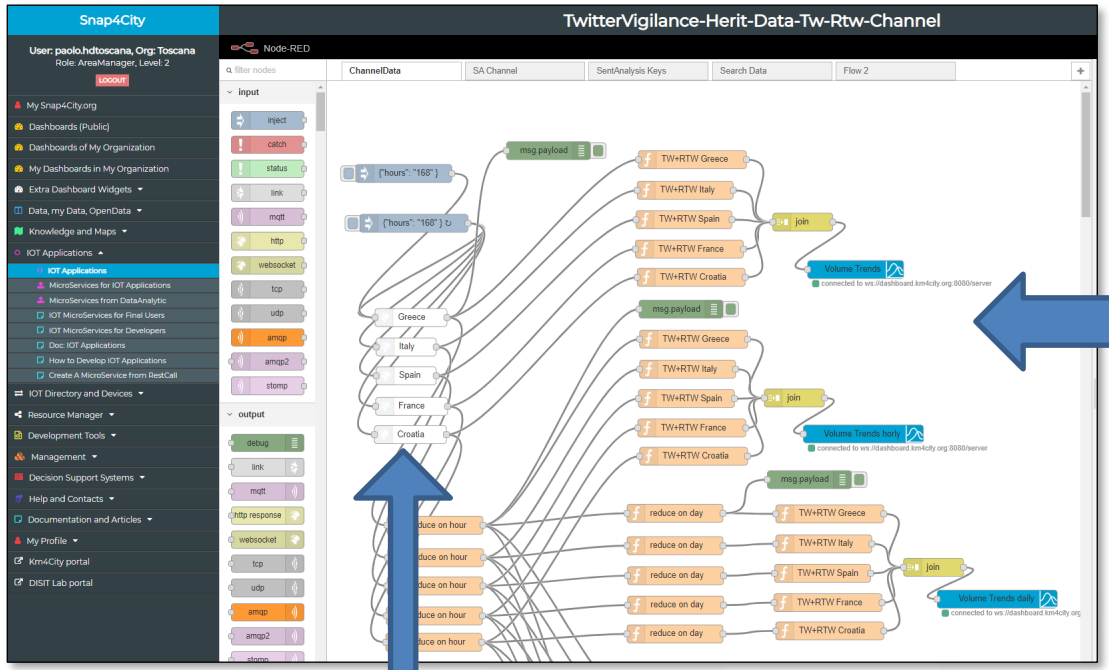
Resource Manager

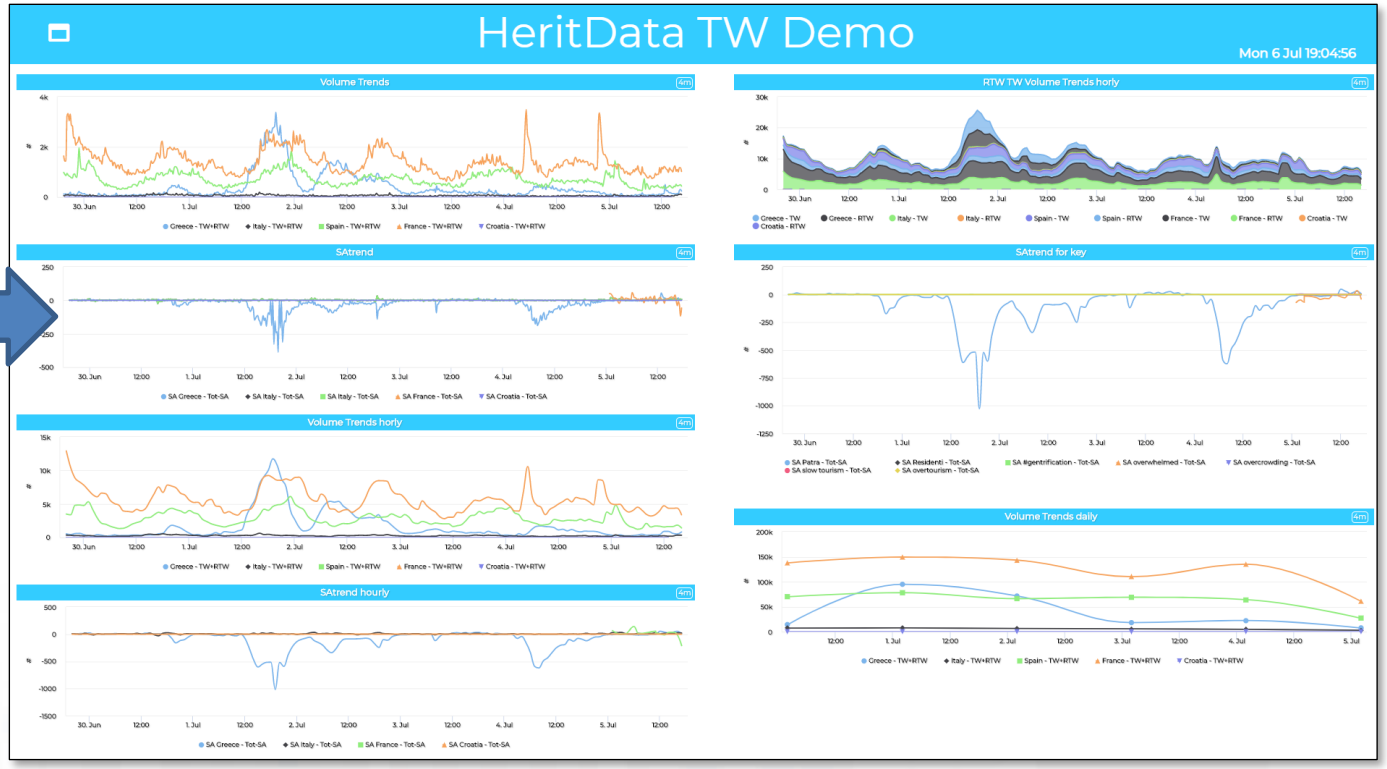


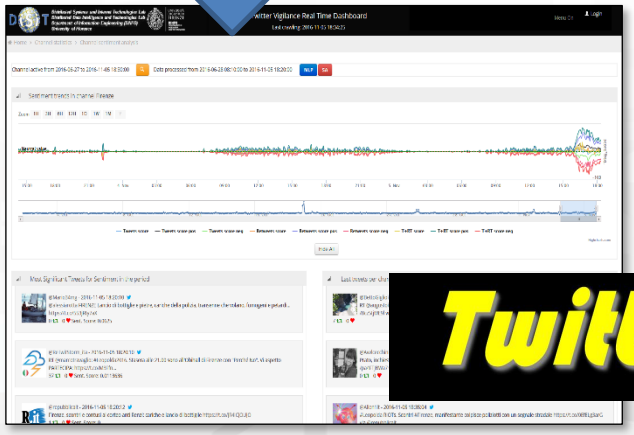


Paolo

Overview (paolo)







Twitter Vigilance

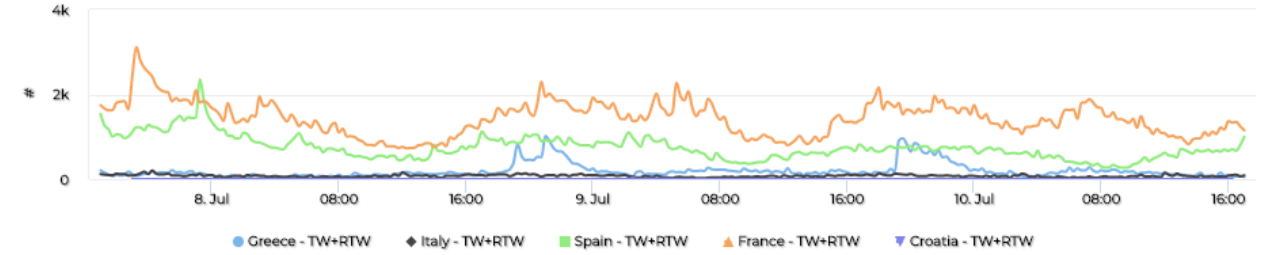
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=Mjc3NQ==

Paolo

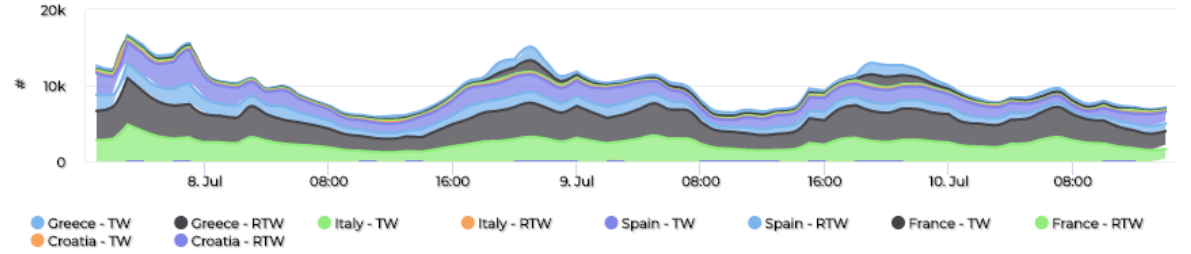
HeritData TW Demo

Tue 14 Jul 16:54:43

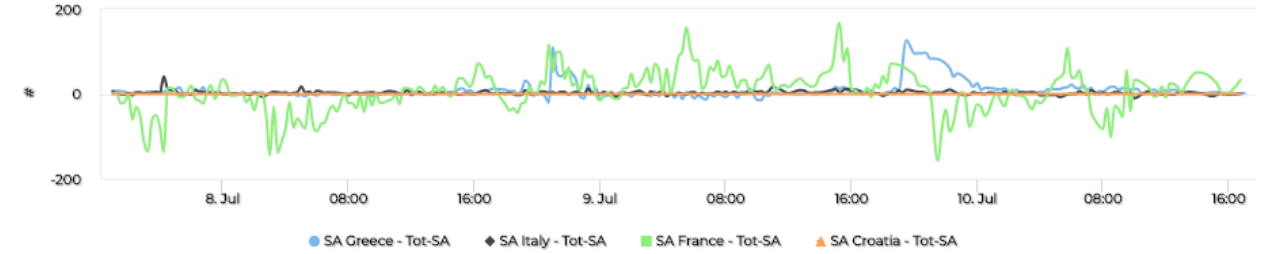
Volume Trends



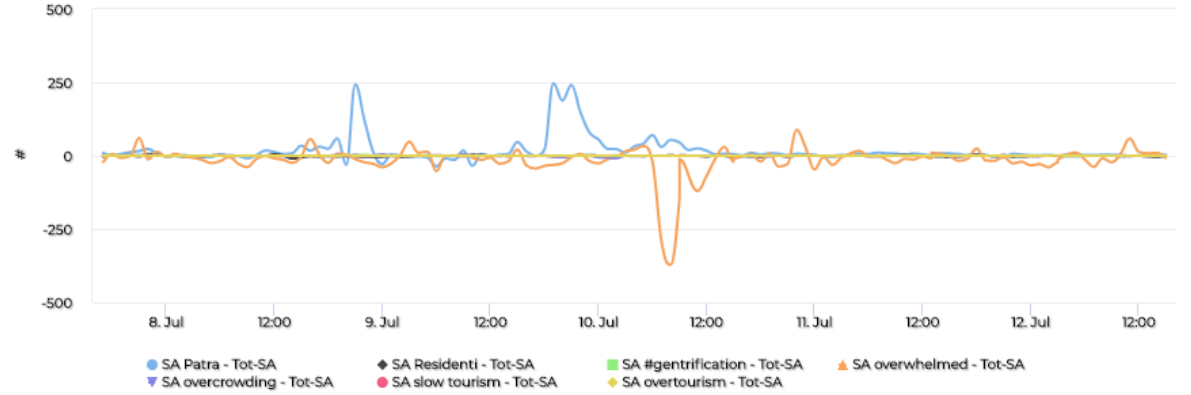
RTW TW Volume Trends horly



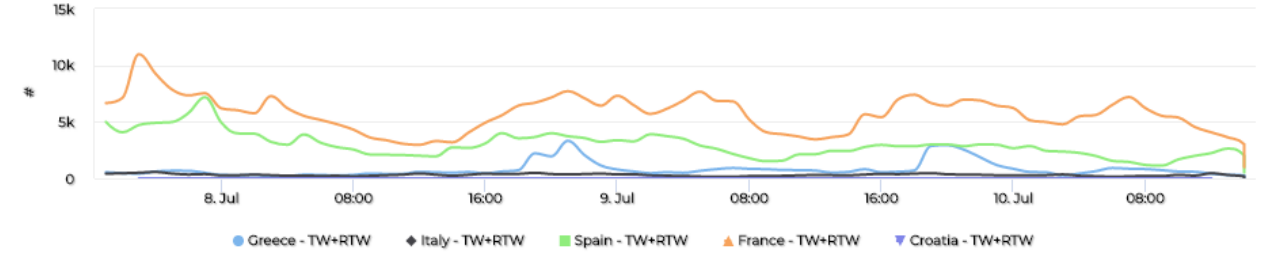
SAtrend



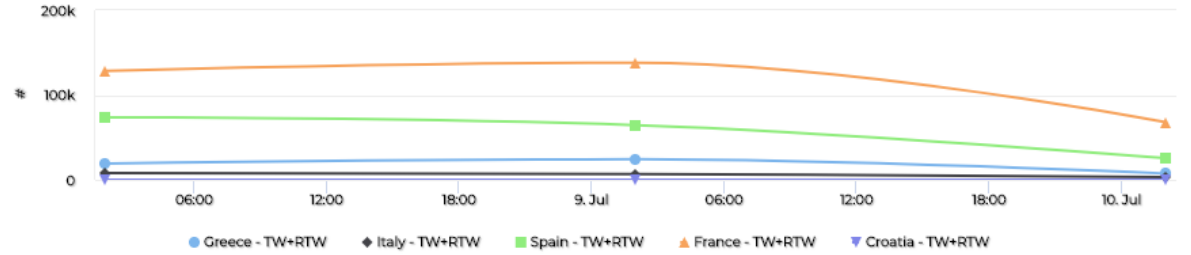
SAtrend for key



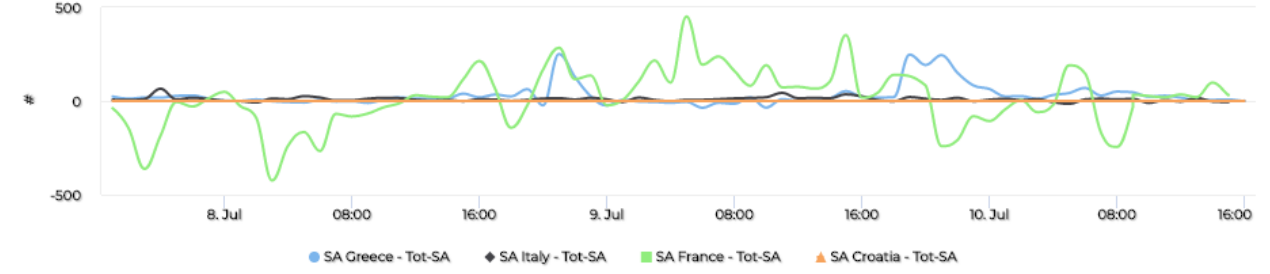
Volume Trends horly

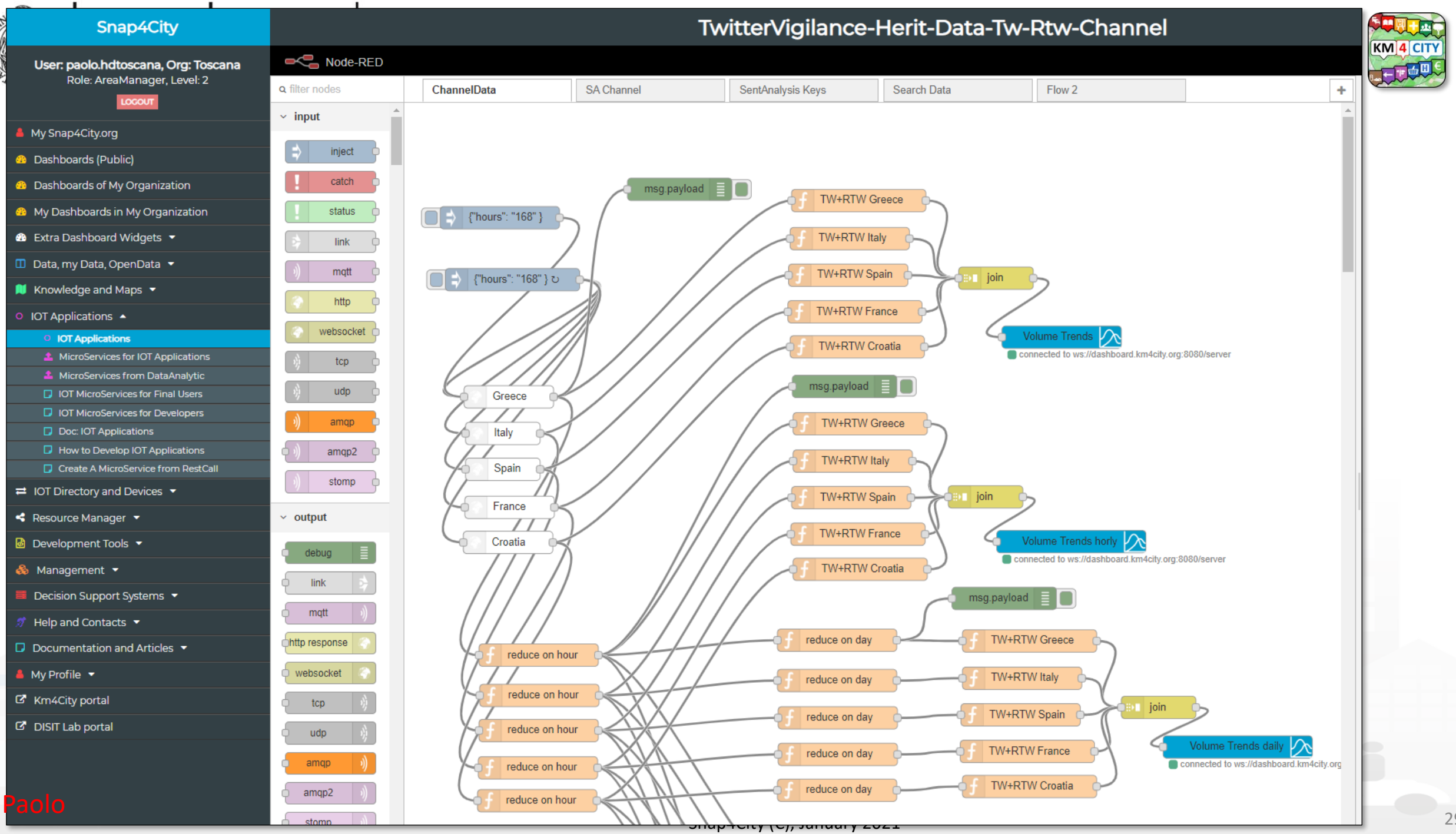


Volume Trends daily



SAtrend hourly

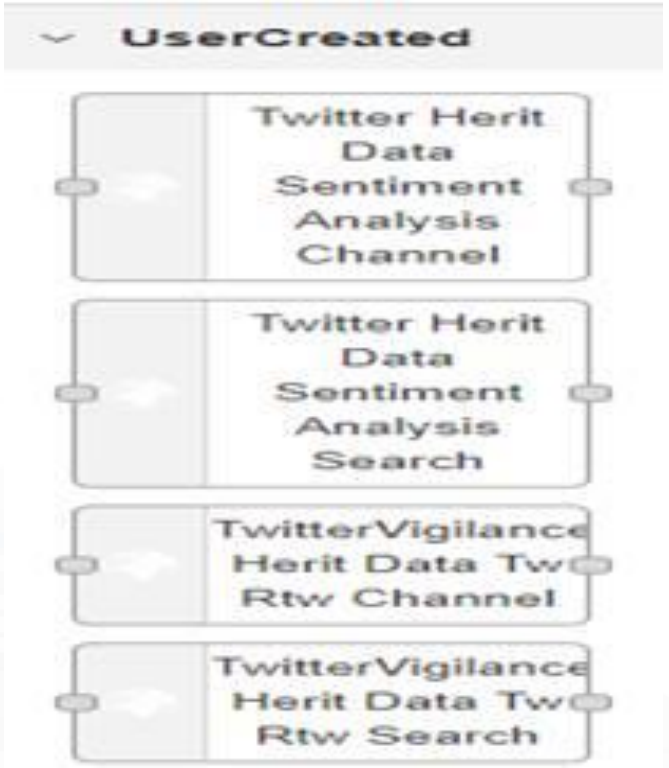



Snap4City
User: paolo.hdtoscana, Org: Toscana
Role: AreaManager, Level: 2
LOGOUT
My Snap4City.org
Dashboards (Public)
Dashboards of My Organization
My Dashboards in My Organization
Extra Dashboard Widgets
Data, my Data, OpenData
Knowledge and Maps
IOT Applications
IOT Applications
MicroServices for IOT Applications
MicroServices from DataAnalytic
IOT MicroServices for Final Users
IOT MicroServices for Developers
Doc: IOT Applications
How to Develop IOT Applications
Create A MicroService from RestCall
IOT Directory and Devices
Resource Manager
Development Tools
Management
Decision Support Systems
Help and Contacts
Documentation and Articles
My Profile
Km4City portal
DISIT Lab portal
TwitterVigilance-Herit-Data-Tw-Rtw-Channel
Node-RED
filter nodes
input
inject
catch
status
link
mqtt
http
websocket
tcp
udp
amqp
amqp2
stomp
output
debug
link
mqtt
http response
websocket
tcp
udp
amqp
amqp2
stomp
ChannelData
SA Channel
SentAnalysis Keys
Search Data
Flow 2
{"hours": "168" }
{"hours": "168" }
msg.payload
Greece
Italy
Spain
France
Croatia
TW+RTW Greece
TW+RTW Italy
TW+RTW Spain
TW+RTW France
TW+RTW Croatia
join
Volume Trends
connected to ws://dashboard.km4city.org:8080/server
msg.payload
TW+RTW Greece
TW+RTW Italy
TW+RTW Spain
TW+RTW France
TW+RTW Croatia
join
Volume Trends horly
connected to ws://dashboard.km4city.org:8080/server
msg.payload
reduce on hour
reduce on hour
reduce on hour
reduce on hour
reduce on hour
reduce on day
reduce on day
reduce on day
reduce on day
reduce on day
TW+RTW Greece
TW+RTW Italy
TW+RTW Spain
TW+RTW France
TW+RTW Croatia
join
Volume Trends daily
connected to ws://dashboard.km4city.org
KM 4 CITY

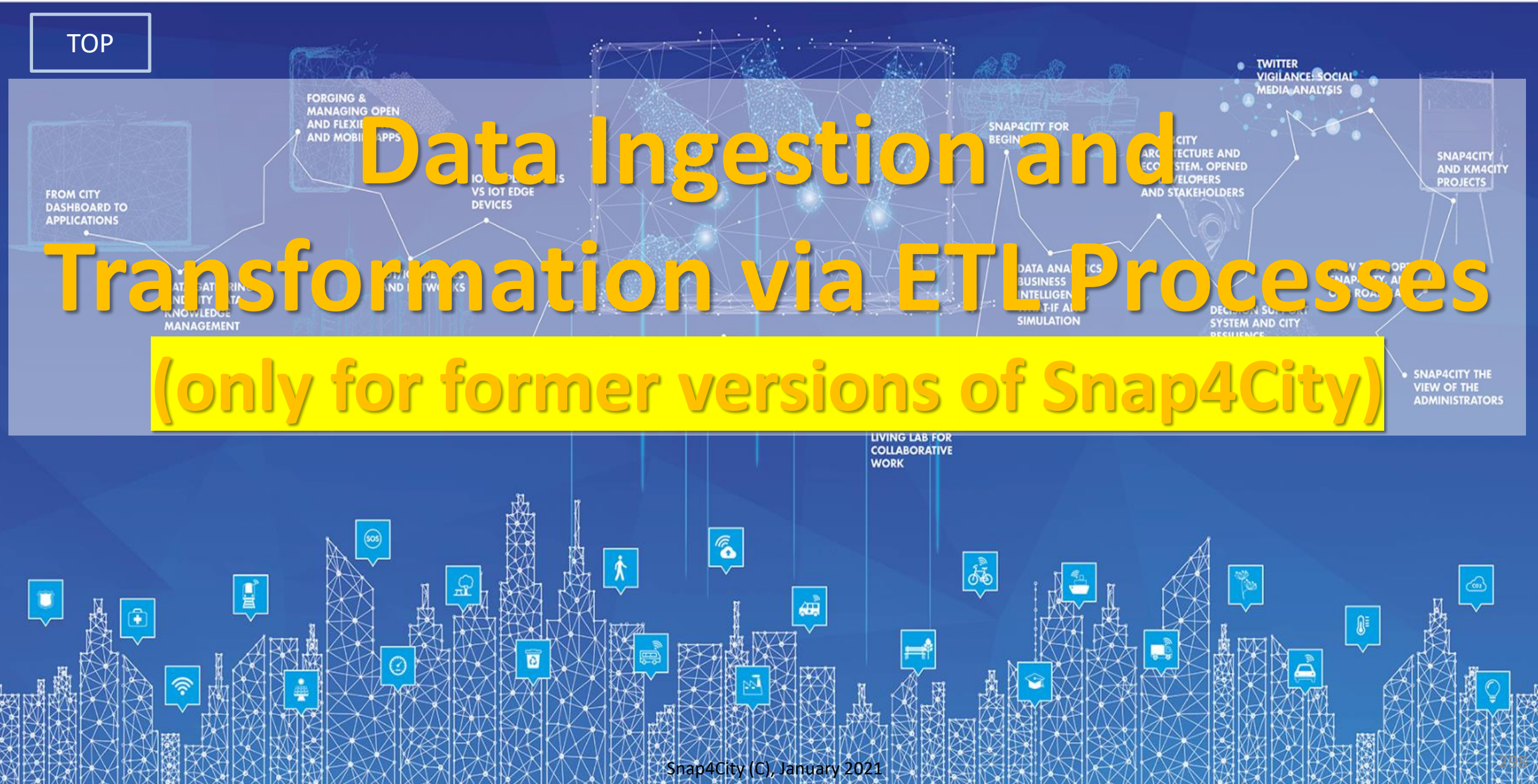
Paolo

Further reading on Social Media

- TC2.22- Exploiting Twitter Vigilance as External Service, in Dashboard, and as RestCall as MicroService in IOT applications
- TC2.21- IOT Applications with Social Media Actions, and cultural scenarious
- External Services
- TC2.21- IOT Applications with Social Media Actions, and cultural scenarious



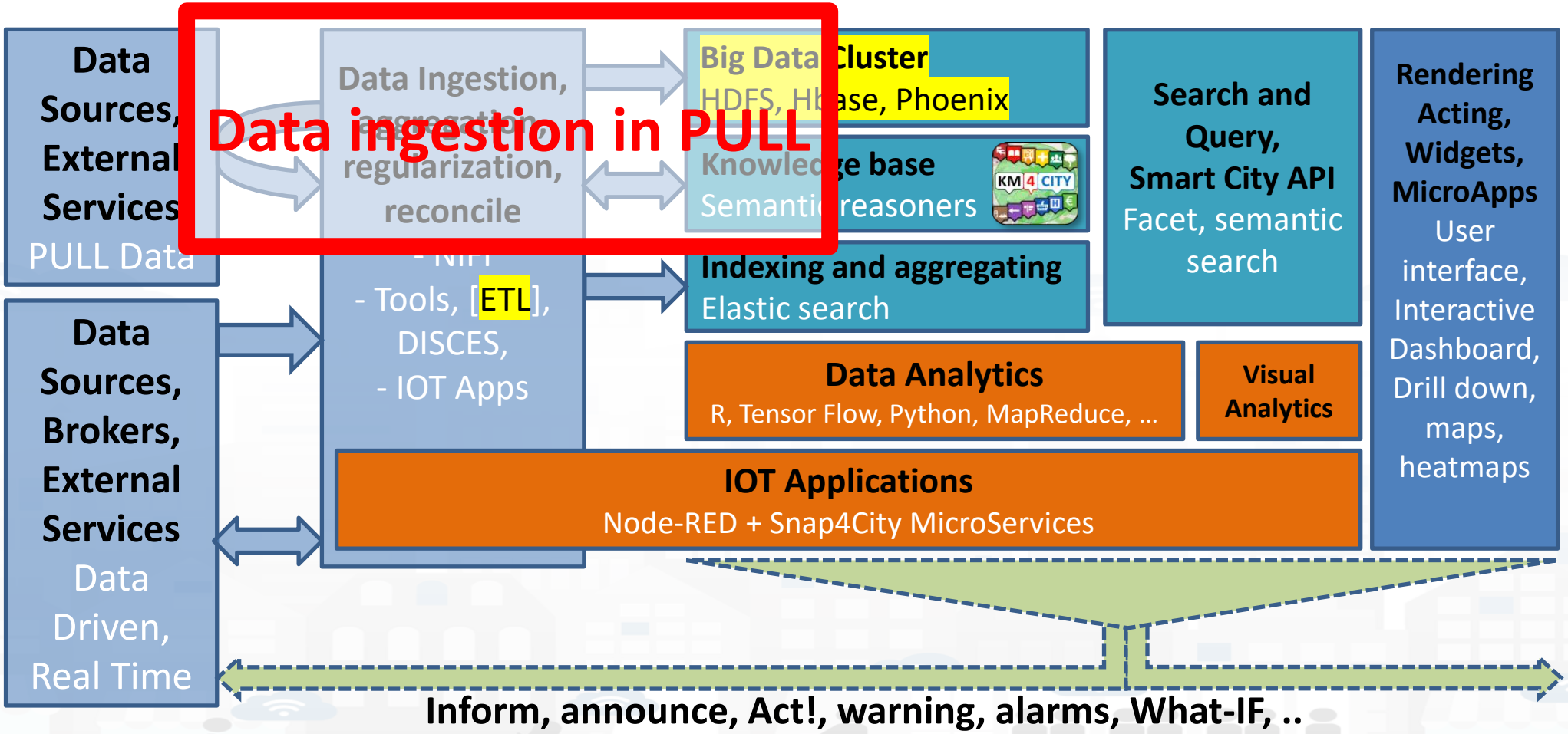
TOP



Data Ingestion and Transformation via ETL Processes

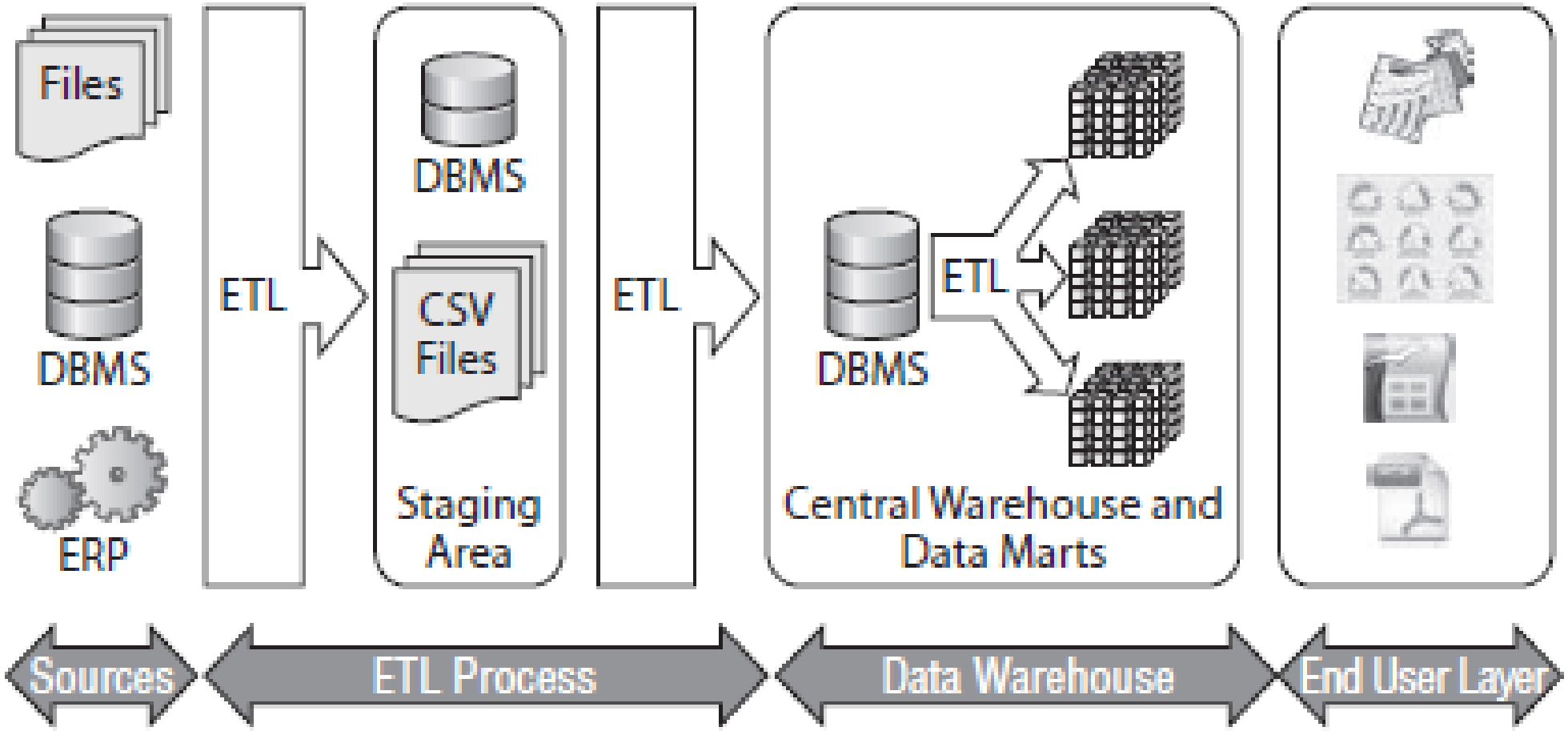
(only for former versions of Snap4City)

Snap4City Architecture vs Data Ingestion



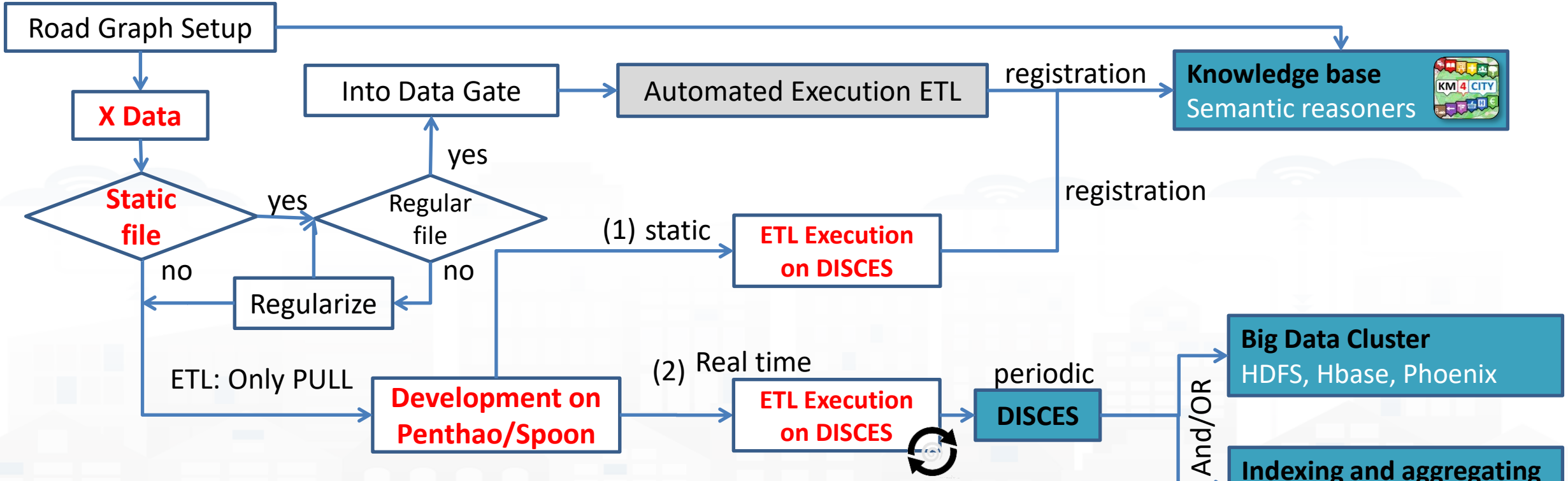
In Yellow alternative & legacy solutions

Classic Data warehouse Architecture



Alternative & legacy Snap4City V1

Data Ingestion Flow Guideline, thumb rules



Alternative & legacy Snap4City V1

Developers of ETL, Data Manager

kettle™ pentaho data integration

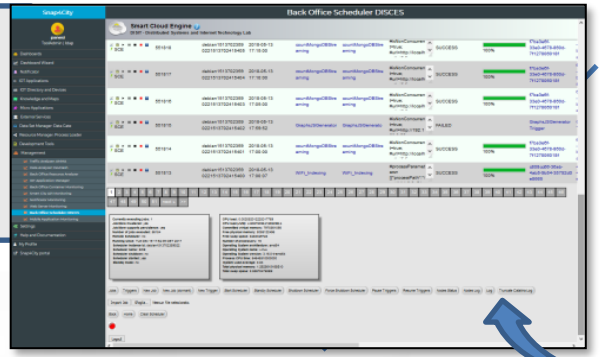
Distributed Back Office

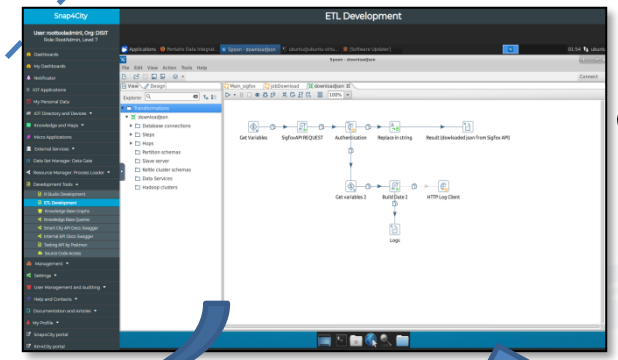
DISCES scheduler production

ETL Development Environment

Data sources

ETL Process



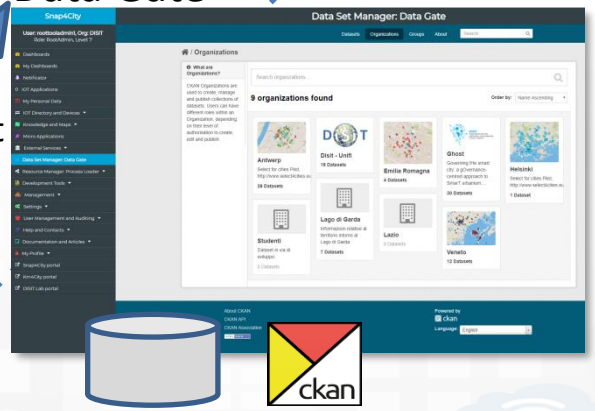


Knowledge Base, Km4City



Knowledge and Storage Data from the Field and City

Data Gate



Data Set Saving / Sharing

Data Set Loading/ Downloading

Saving / Sharing reusing



Resource Manager

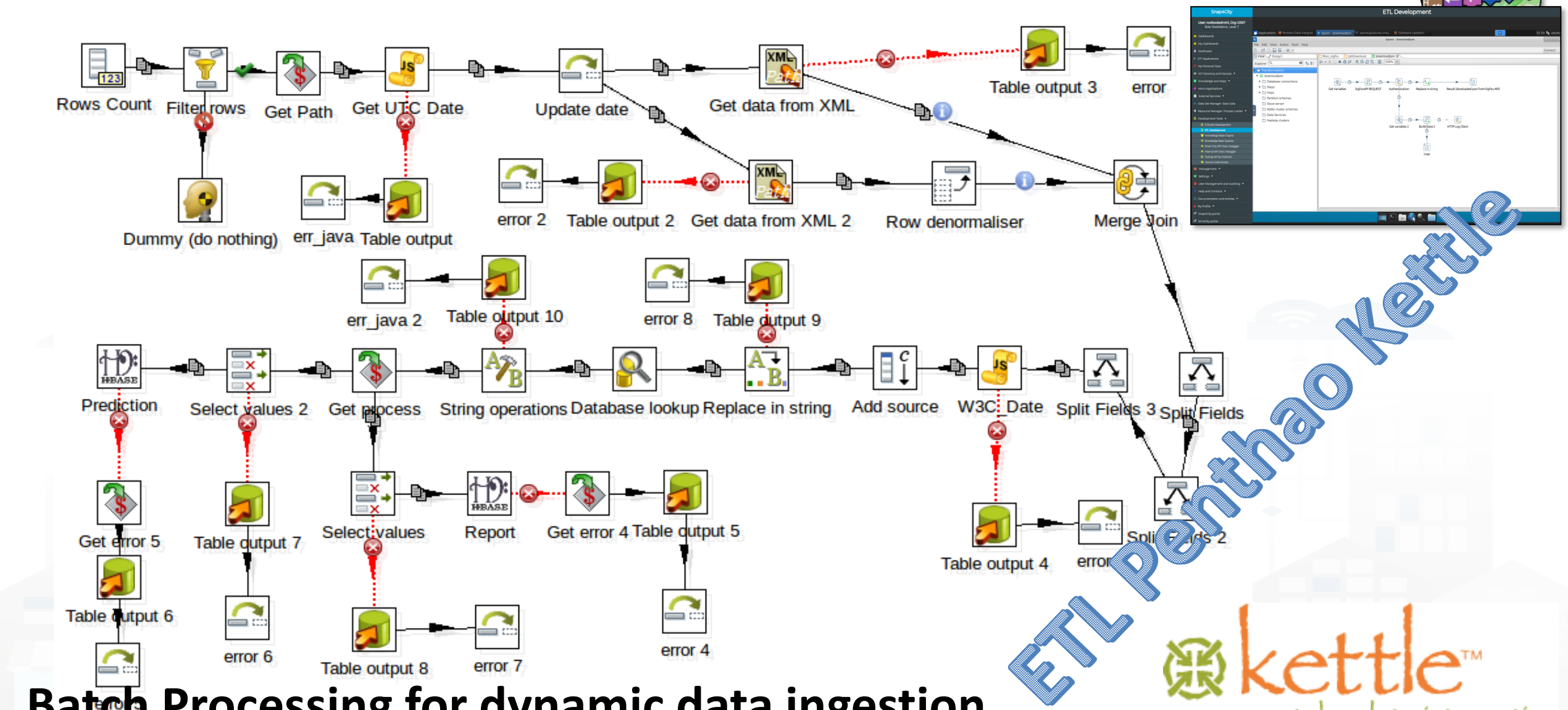
Schedule Execute

Load data or prepare for data ingestion

Alternative & legacy Snap4City V1

Integrated ETL development

SNAP4city



ETL Penthao Kettle

kettle™ pentaho data integration

Batch Processing for dynamic data ingestion

Alternative & legacy Snap4City V1

ETL Processes

The three phases are:

- **Extracting** data from outside sources (**Ingestion** phase).
- **Transforming** data to fit operational needs which may include improvements of quality levels (**Data Quality Improvement** phase).
- **Loading** data into the end target (database, operational data store, data warehouse, data mart, etc.). So the data can be translated in **RDF triples using a specific ontology (Static/periodic datasets) or on NoSQL Databases (Dynamic datasets)**

Phase I: Data Ingestion

- **Purpose is to store data in HBase**

(Big Data NoSQL database).

- **Acquisition of wide range of OD/PD**: open and private data, static, quasi static and/or dynamic real time data.
- **Static and semi-static data** include: points of interests, geo-referenced services, maps, accidents statistics, etc.
  - files in several formats (SHP, KML, CVS, ZIP, XML, JSON, etc.)
- **Dynamic data** mainly data coming from sensors
  - parking, weather conditions, pollution measures, bus position, …
  - using Web Services
- Using **Pentaho - Kettle** for data integration (Open Source tool)
  - using specific **ETL** Kettle transformation processes (one or more for each data source)

Phase II: Data Quality Improvement

- **Purpose: add more information as possible and normalize data from ingestion**
- **Problems kinds:**
  - Inconsistencies, incompleteness, typos, lack of standards, multiple standards, …
- **Problems on:**
  - Place-name code
  - Street names (e.g., dividing names from numbers, normalize when possible)
  - Dates and Time: normalizing
  - Telephone numbers: normalizing
  - Web links and emails: normalizing

Phase III: Data mapping

- **Purpose is to translate data from QI in RDF triples**
- **We use triples to do inference on data.**

- Using **Karma Data Integration tool, a mapping model** from SQL to RDF on the basis of the ontology was created.
  - Data to be mapped first temporary passed from HBase to MySQL and then mapped using Karma (in batch mode)

- The mapped data in triples have to be uploaded (and indexed) to the **RDF Store** (Virtuoso).
- Triples are composed by a subject, a predicate and an object.

Pentaho Data Integration (Kettle)

Main strengths:

- Collect data from a **variety of sources** (extraction);
- Move and modify data (transport and transform) while cleansing, denormalizing, aggregating and enriching it in the process;
- Frequently (daily) store data (loading) in the final target destination, usually a **large** dimensionally modeled **database** (or **data warehouse**).

- **Spoon:** graphically oriented end-user tool to model the **flow of data** from input through transformation to output (**transformation**)
- **Pan** is a **command line tool** that executes transformations modeled with Spoon
- **Chef**: a graphically oriented **end-user tool** used to model **jobs** (transformations, FTP downloads etc. placed in a flow of control)
- **Kitchen** is a **command line tool** to execute jobs created with Chef.

Type of Steps in Spoon

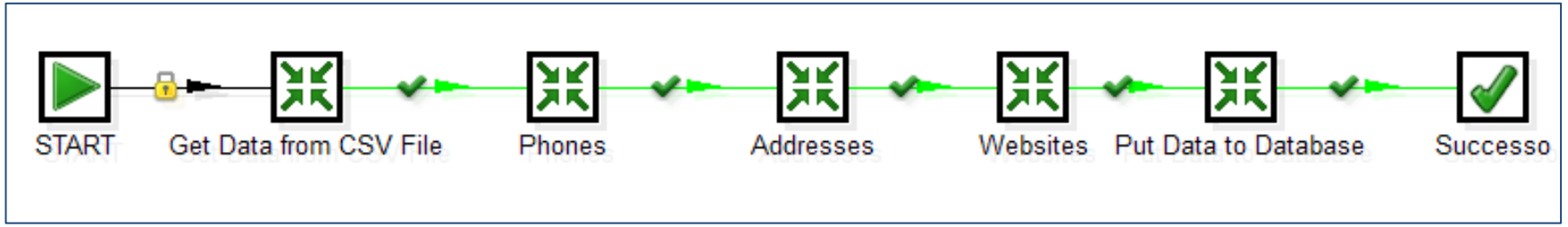
Three different kinds of steps:

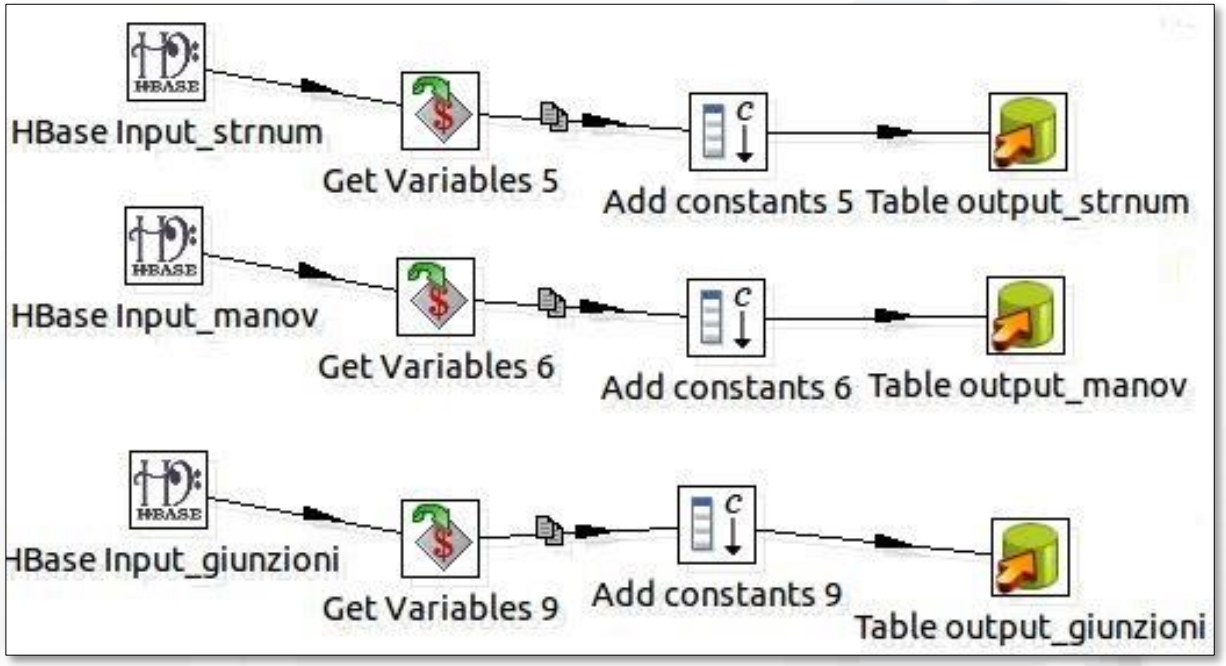
- **Input:** process some kind of 'raw' resource (file, database query or system variables) and create an output stream of records from it.
- **Output:** (the reverse of input steps): accept records, and store them in some external resource (file, database table, etc.).
- **Transforming:** process input streams and perform particular actions on them (adding new fields/new records); these actions produce one or more output streams.

Type of Transformations in Spoon

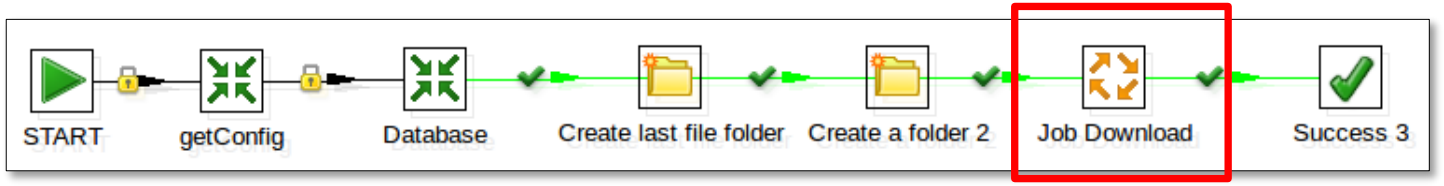
- Input
  - Access Input
  - CSV file input
  - Data Grid
  - De-serialize from file
  - Email messages input
  - ESRI Shapefile Reader
  - Excel Input
  - Fixed file input
  - Generate random credit
  - Generate random value
  - Generate Rows
  - Get data from XML
  - Get File Names
  - Get Files Rows Count
  - Get SubFolder names
  - ...
  - Salesforce Input
  - SAP Input
  - Table input
  - Text file input
  - XBase input
  - Yaml Input
- Scripting
  - Execute row SQL script
  - Execute SQL script
  - Formula
  - Modified Java Script Value
  - Regex Evaluation
  - User Defined Java Class
  - User Defined Java Expression
- Transform
  - Add a checksum
  - Add constants
  - Add sequence
  - Add value fields changing
  - Add XML
  - Calculator
  - Closure Generator
  - Example plugin
  - Number range
  - Replace in string
  - Row denormaliser
  - Row flattener
  - Row Normaliser
  - Select values
  - Sort rows
  - Split field to rows
  - Split Fields
  - String operations
  - Strings cut
  - Unique rows
  - Unique rows (HashSet)
  - Value Mapper
  - XSL Transformation
- Lookup
  - Call DB Procedure
  - Check if a column exists
  - Check if file is locked
  - Check if webservice is avail
  - Database join
  - Database lookup
  - Dynamic SQL row
  - File exists
  - Fuzzy match
  - HTTP client
  - HTTP Post
  - Stream lookup
  - Table exists
  - Web services lookup
- Utility
  - Change file encoding
  - Clone row
  - Delay row
  - Execute a process
  - If field value is null
  - Mail
  - Metadata structure of str
  - Null if...
  - Process files
  - Run SSH commands
  - Send message to Syslog
  - Write to log
- Output
  - Access Output
  - Delete
  - Excel Output
  - Insert / Update
  - Json output
  - LDAP Output
  - Palo Cells Output
  - Palo Dimension Output
  - Properties Output
  - RSS Output
  - Salesforce Delete
  - Salesforce Insert
  - Salesforce Update
  - Salesforce Upsert
  - Serialize to file
  - SQL File Output
  - Synchronize after merge
  - Table output
  - Text file output
  - Update
  - XML Output

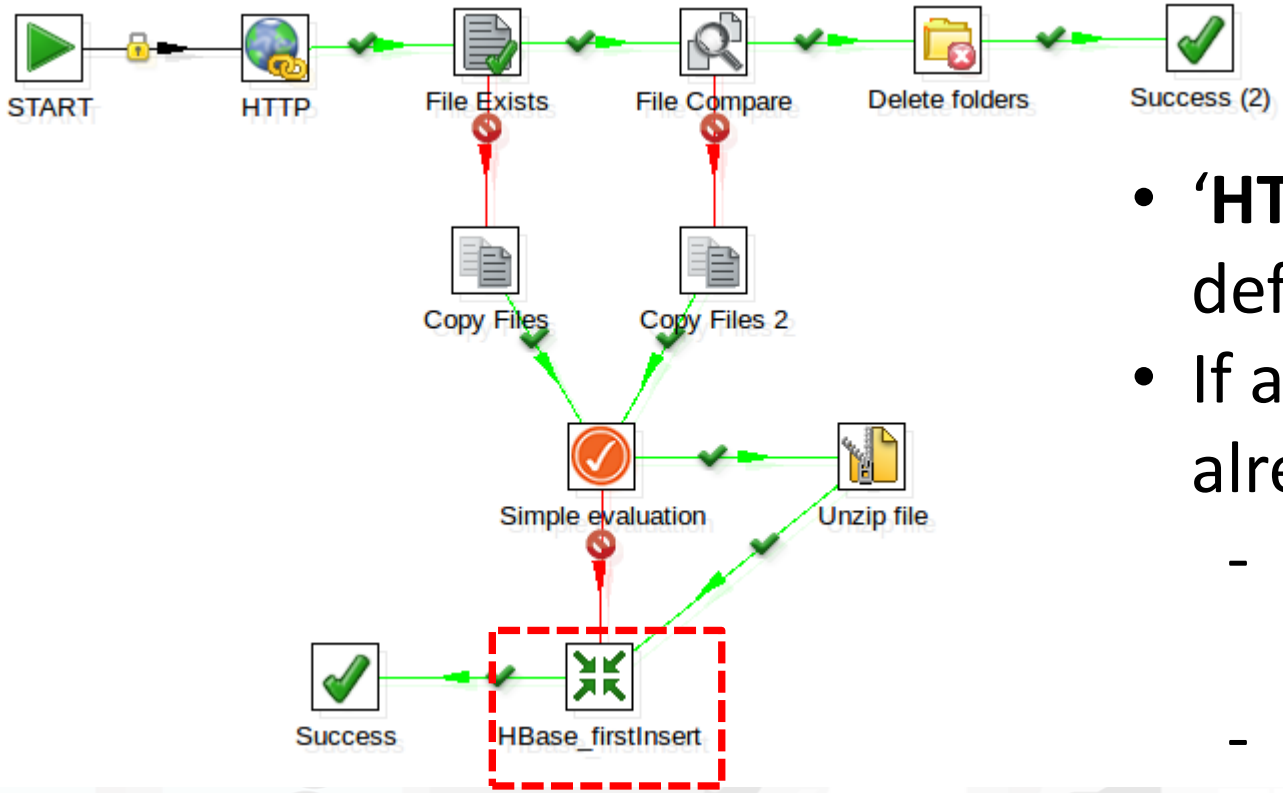
Sequential vs parallel





Job Download

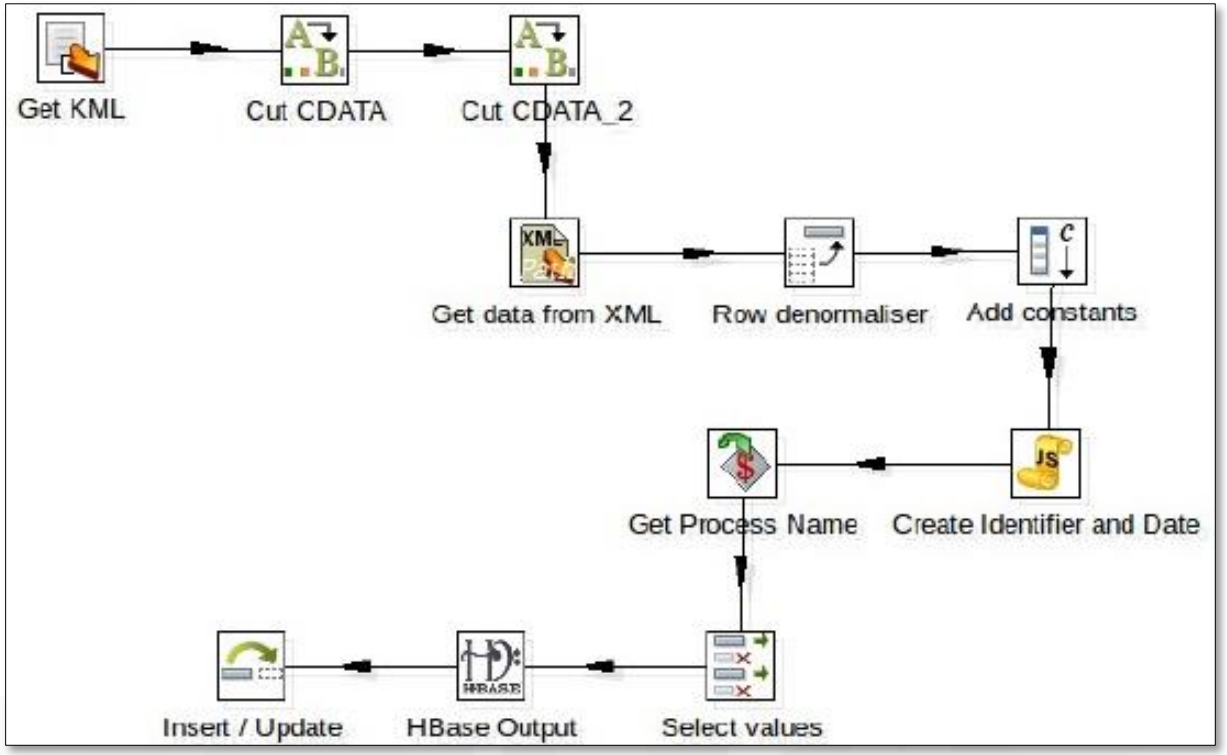




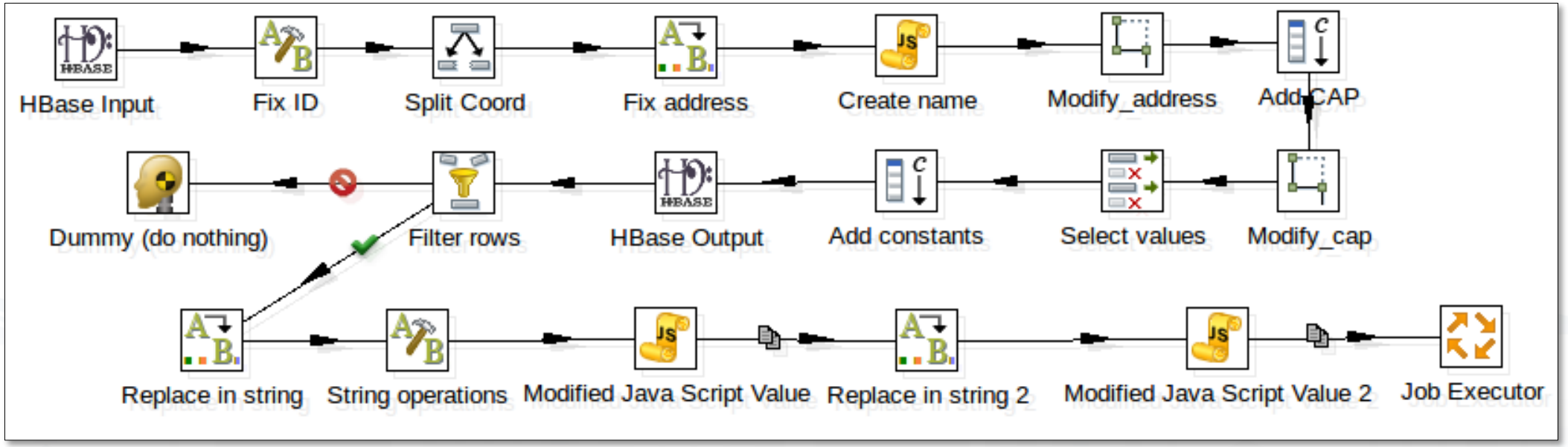
- '**HTTP**' step downloads the dataset defined in *PARAM* (URL) field.
- If a file with the same name already exists, we compare them.
  - If they are the same, we delete the folders created before.
  - Otherwise we unzip the file and '**HBase_firstInsert**' is called.

HBase_firstInsert

- '**Get KML**' loads the file just downloaded.
- '**Get data from XML**' and '**Row denormaliser**' extract fields from source file.
- In the JS step, we create an identifier (it will be use as key in HBase).
- '**HBase Output**' saves the information in a HBase table.
- '**Insert/Update**' updates the last ingestion date in MySQL table *process_manager2.*

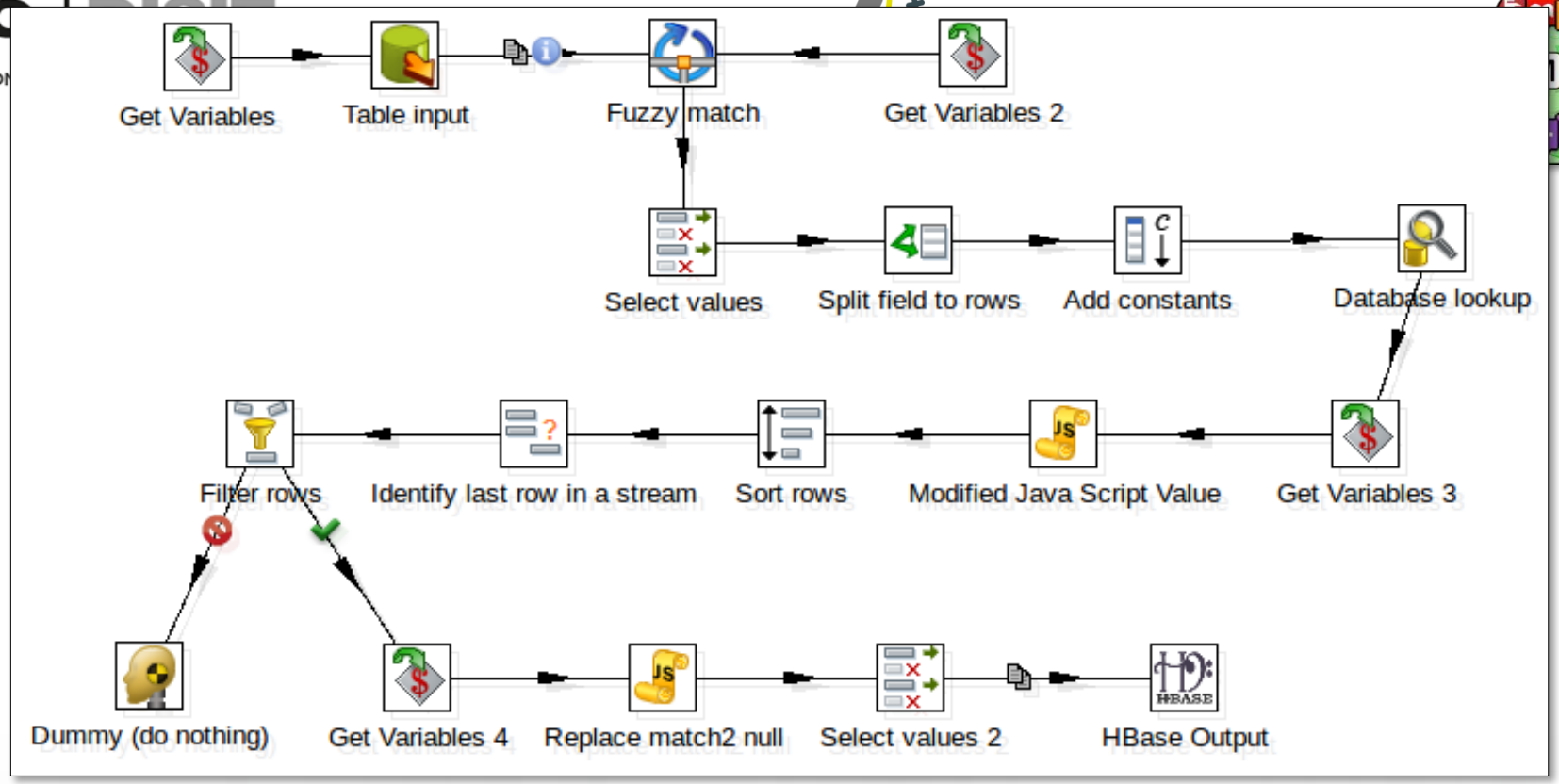


Quality Improvement



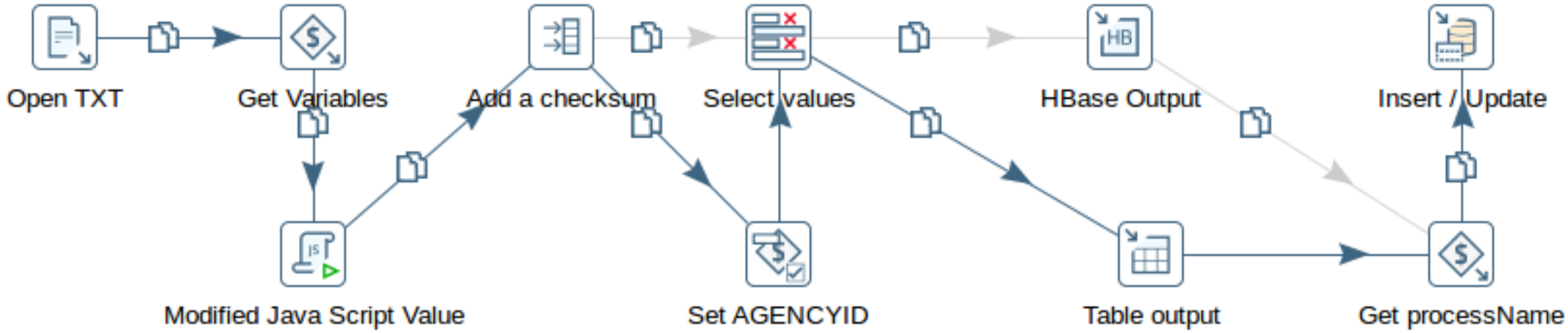
- '**HBase Input**' gets back data saved at the ingestion end.
- '**Fix address**' is used to correct typing error (e.g. Giovambattista instead of Giovanbattista) or to simplify search of right toponym code.
- '**Modify_***' transformations normalize address, CAP, website, e-mail, phone number, …
- '**Add constants**' adds two fields (address_syn, codice_toponimo) which we will use in the next job.
- '**HBase Output**' saves in a new HBase table the quality improvement result.
- For the rows which have an un-empty *streetAddress*, the steps below extract a word from *streetAddress* which we will use to find the right toponym code.

Job Executor



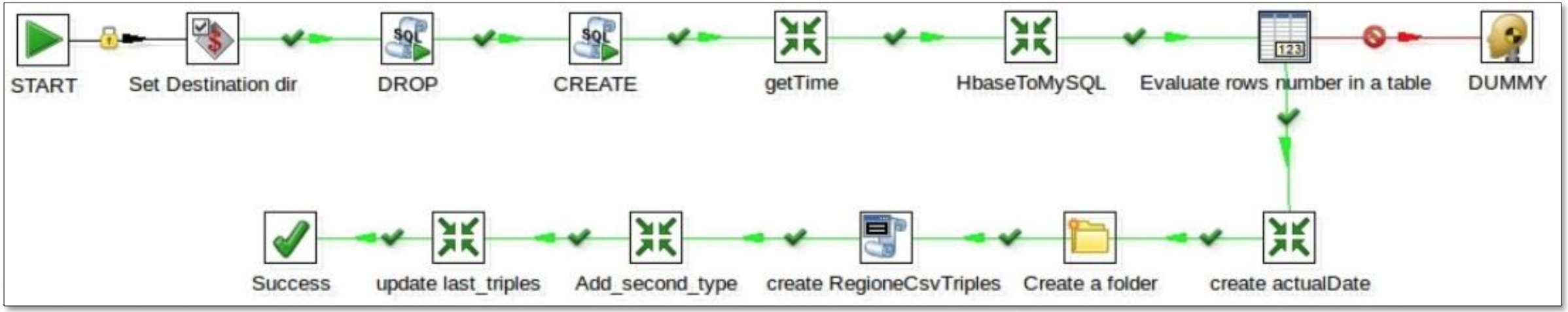
- To get toponym codes we use *tbl_toponimo_bis* MySQL table. It contains road names, toponym codes, town codes, …. Each road has his center coordinates (lat, long).
- '**Table input**' does the query on that table using the word created before.
- '**Fuzzy match**' calculates similarity (from 0.2 up to 1) between the query result and the address acquired during ingestion phase.
- Following steps calculate distance from ingestion coordinates and query ones and select the closest.
- '**HBase Output**' fills *address_syn* and *codice_toponimo* in QI HBase table.

QI Problems



- The method used for toponym code extraction sometimes produces wrong results.
- Given a word, might be impossible to determine right toponym code (e.g., query using '*Brunelleschi*' returns '*Via dei Brunelleschi*', '*Piazza Brunelleschi*').
- Summarize roads with their centers and calculate distances could not return right results (if the point of interest is far from his road center, it could be closer to another road center).

Triplification



- Since we use Karma to generate models, we have to move data from HBase to MySQL.
- '**DROP**' and '**CREATE**' respectively deletes MySQL table if already exists and creates a new one.
- '**getTime**' returns last triple generation timestamp for a specific process (using MySQL *process_manager2* table).
- '**HBaseToMySQL**' moves data from HBase to MySQL only if timestamp just calculated is older than the date of last ingestion (it prevents to generate triples based on the same data).
- '**create RegioneCSVTriples**' calls Karma script to generate triples based on the model.
- '**update last_triples**' updates last triple generation timestamp in *process_manager2*.

Scheduling Real Time Ingestions

- For **Real Time data** (car parks, road sensors, etc.), the ingestion and triple generation processes should be performed periodically (no for **static data**).

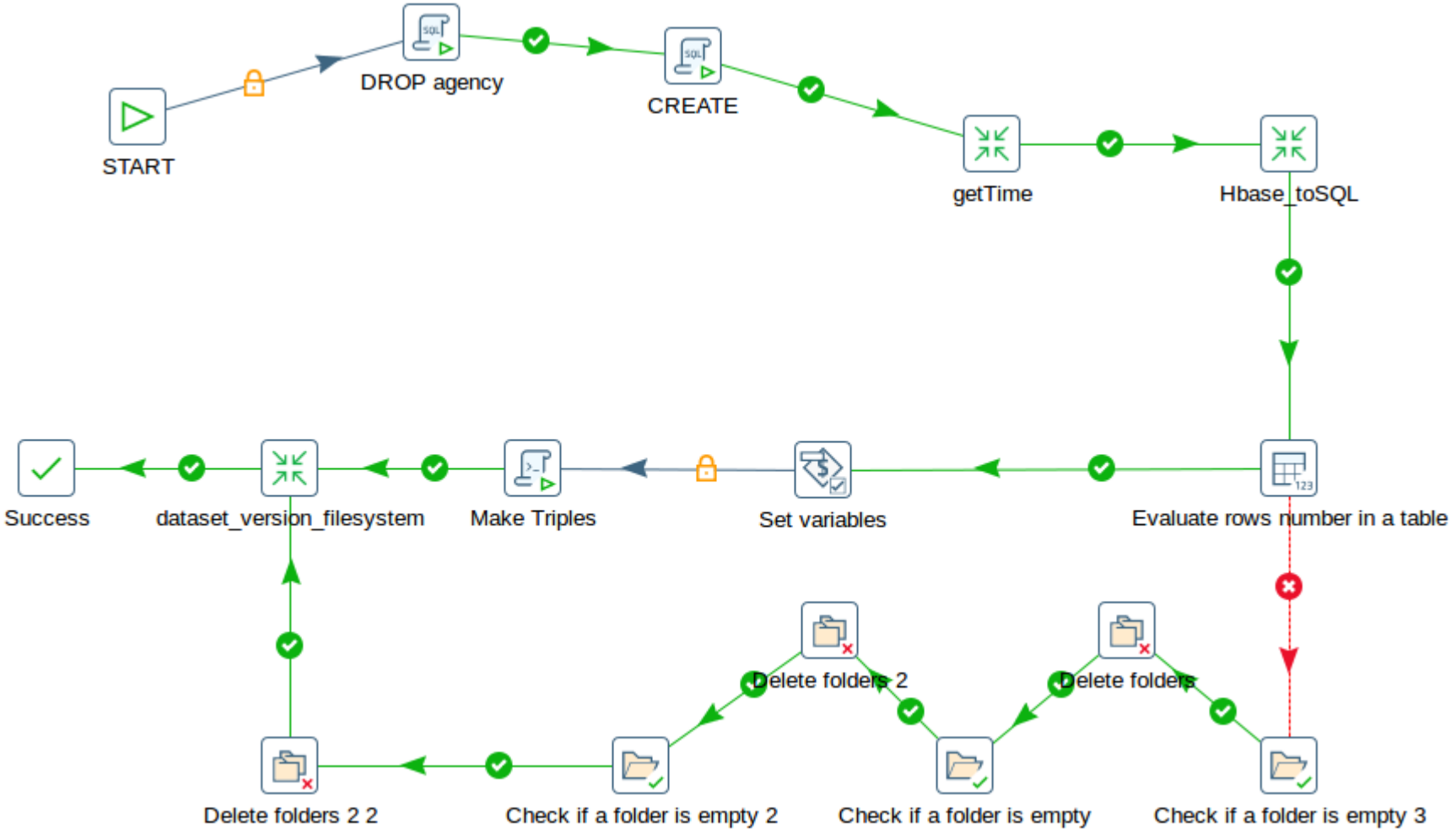
- A scheduler is used to manage the periodic execution of ingestion and triple generation processes;
  - this tool throws the processes with a predefined interval determined in phase of configuration.

Scheduler: DISCES

Smart Cloud Engine
DISIT - Distributed Systems and Internet Technology Lab

SCHEDULER NAME	ID	FIRE INSTANCE ID	DATE	JOB NAME	JOB GROUP	JOB DATA	STATUS	PROGRESS	TRIGGER NAME	TRIGGER GROUP	PREV FIRE TIME	NEXT FIRE
SCE	183652	hadoopnodet0715265676204441526567620531	2018-05-17 19:52:04	sensore_Demidoff_periferia	sensori_traffico_veicolare	#processParameters=[{"processPath":"/u	SUCCESS	100%	sensore_Demidoff_periferia_trigger	sensori_traffico_veicolare_trigger	2018-05-17 19:52:00	2018-05-17
SCE	183651	hadoopnodet0415265553660711526555366283	2018-05-17 19:50:02	Sensori_MeteoSIR_Phoenix	Sensori_Phoenix	#processParameters=[{"processPath":"//	FAILED		Sensori_MeteoSIR_Phoenix_trigger	Sensori_Phoenix	2018-05-17 19:50:00	2018-05-17
SCE	183650	hadoopnodet0715265676204441526567620530	2018-05-17 19:50:32	CarPark_MIIC_160_Phoenix	CarPark_Phoenix	#processParameters=[{"processPath":"/h	SUCCESS	100%	CarPark_MIIC_160_Phoenix_trigger	CarPark_Phoenix_trigger	2018-05-17 19:50:00	2018-05-17
SCE	183649	hadoopnodet04c1526555081135152655550811132	2018-05-17 19:49:19	Bike_Siena_Phoenix	Bike_Phoenix	#processParameters=[{"processPath":"/h	SUCCESS	100%	Bike_Siena_Phoenix_trigger	Bike_Phoenix_trigger	2018-05-17 19:49:00	2018-05-17
SCE	183648	hadoopnodet0415265553660711526555366281	2018-05-17 19:49:38	smart_waste_FI_phoenix	smart_waste_phoenix	#processParameters=[{"processPath":"/h	SUCCESS	100%	8b095cf2-d5b3-4c93-890a-b6544d950542	137efde2-25b6-4b7c-95ef-1aaff2f8ea51	2018-05-17 19:49:00	2018-05-17
SCE	183647	hadoopnodet0715265676204441526567620528	2018-05-17 19:49:36	Bike_Pisa_Phoenix	Bike_Phoenix	#processParameters=[{"processPath":"/h	SUCCESS	100%	Bike_Pisa_Phoenix_trigger	Bike_Phoenix_trigger	2018-05-17 19:49:00	2018-05-17
SCE	183646	hadoopnodet0715265676204441526567620527	2018-05-17 19:49:05	Sensore_Senese_periferia	sensori_traffico_veicolare	#processParameters=[{"processPath":"/u	SUCCESS	100%	Sensore_Senese_periferia_trigger	sensori_traffico_veicolare_trigger	2018-05-17 19:49:00	2018-05-17
SCE	183645	hadoopnodet04c1526555081135152655550811131	2018-05-17 19:48:06	Sensore_Senese_Centro	sensori_traffico_veicolare	#processParameters=[{"processPath":"/u	SUCCESS	100%	Sensore_Senese_Centro_trigger	sensori_traffico_veicolare_trigger	2018-05-17 19:48:00	2018-05-17
SCE	183644	hadoopnodet0415265553660711526555366280	2018-05-17 19:49:38	charging_stations_FI_phoenix	charging_stations_phoenix	#processParameters=[{"processPath":"/h	SUCCESS	100%	charging_stations_FI_phoenix_trigger	charging_stations_phoenix_trigger	2018-05-17 19:47:00	2018-05-17
SCE	183643	hadoopnodet0715265676204441526567620526	2018-05-17 19:48:10	weather_sensor_SantaMarta_FI_phoenix	weather_sensor_phoenix	#processParameters=[{"processPath":"/h	SUCCESS	100%	weather_sensor_SantaMarta_FI_phoenix_trigger	weather_sensor_phoenix_trigger	2018-05-17 19:46:00	2018-05-17
SCE	183642	hadoopnodet0415265553660711526555366278	2018-05-17 19:46:58	CarPark_SienaParcheggi_Phoenix	CarPark_Phoenix	#processParameters=[{"processPath":"/h	SUCCESS	100%	CarPark_SienaParcheggi_Phoenix_trigger	CarPark_Phoenix_trigger	2018-05-17 19:45:00	2018-05-17
SCE	183641	hadoopnodet0715265676204441526567620520	2018-05-17 19:47:58	CarPark_MIIC_164_Phoenix	CarPark_Phoenix	#processParameters=[{"processPath":"/h	SUCCESS	100%	CarPark_MIIC_164_Phoenix_trigger	CarPark_Phoenix_trigger	2018-05-17 19:45:00	2018-05-17

Triplification Overview



Triplification Overview

java -cp target/karma-offline-0.0.1-SNAPSHOT-shaded.jar **edu.isi.karma.rdf.OfflineRdfGenerator**
--sourcetype DB
--modelfilepath "${MODELPATH}/${processName}_**agency.ttl**"
--outputfile ${TRIPLESDESTDIRECT}/**agency.n3**
--dbtype **MySQL**
--hostname ${IPADDRESSMASTER}
--username ${USERNAMEMYSQL}
--password ${PSWMYSQL}
--portnumber ${PORTMYSQL}
--dbname ${DATABASEMYSQL}
--tablename ${processName}_agency

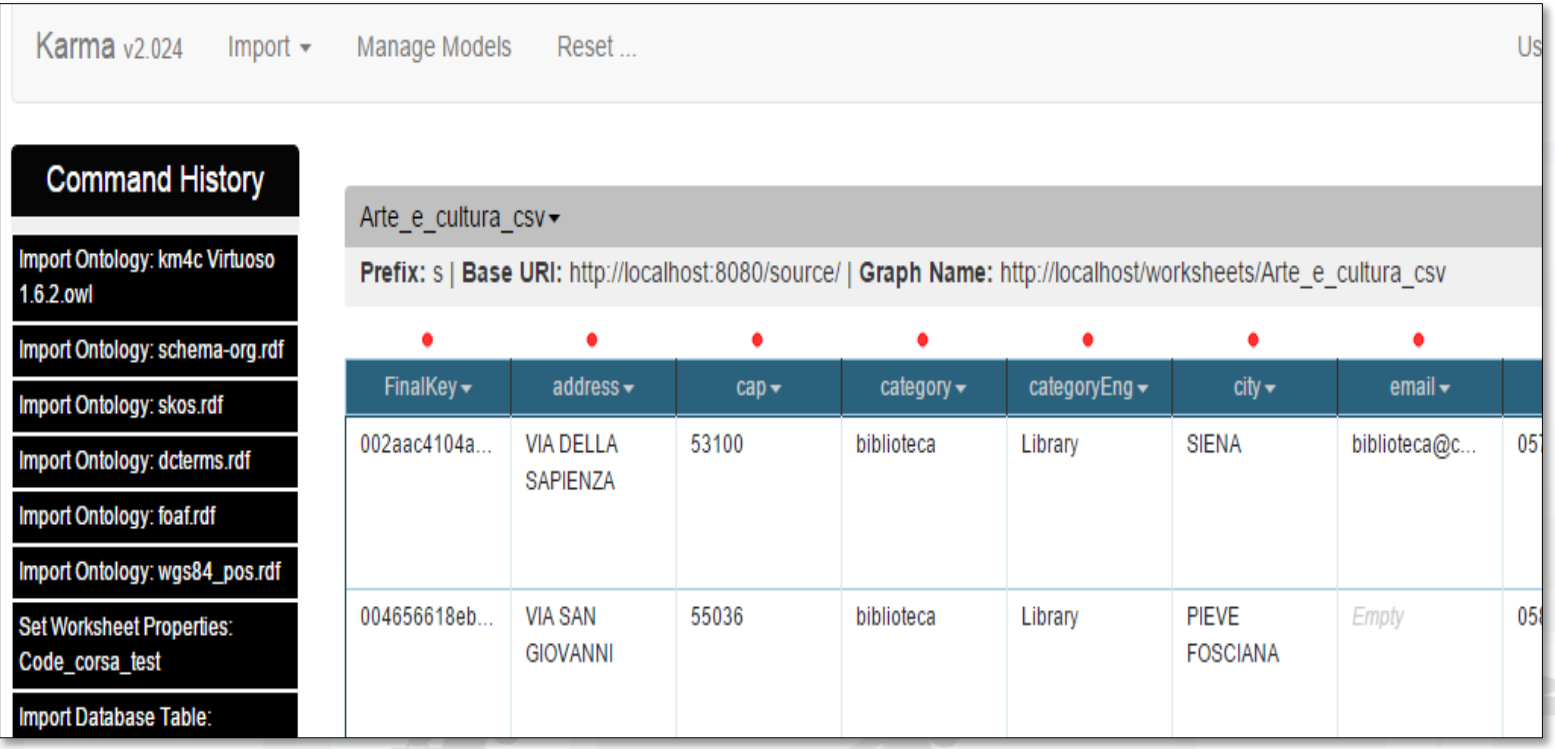
Karma

An introductory guide for generating RDF triples from relational data in minutes

Karma (http://usc-isi-i2.github.io/karma/) is a data integration tool developed at the Center of Knowledge Graphs of the Information Science Institute at the University of Southern California. Here at DISIT, we exploit Karma for triplifying relational data.

Karma

- Karma is a mapping model based on ontology (**km4city**) from MySQL tables to RDF.
- Triples are uploaded to **Virtuoso**, an RDF Store.
- It can import MySQL tables but no HBase ones.



Here are the steps to have your triples in your pocket in minutes:

1. Get one of the ready-to-use DISIT VMs
2. Launch the Karma server
3. Build your own model:
    1. Load vocabularies
    2. Load relational tables
    3. (Optional) Load R2RML models
    4. Define mappings
4. Export your model
5. Launch the command-line tool
6. Enjoy!

Get one of the ready-to-use DISIT VMs

The DISIT Lab makes available through its Drupal portal a set of ready-to-use virtual machines specifically oriented to data integration. Here is how you can get one of them:

1. Connect to http://www.disit.org/drupal/?q=node/6690 and scroll down to the section "MACCHINA VIRTUALE, VMSDETL, GIA' PRONTA"
2. Get the "Versione del 2017/2018 0.8 con Phoenix" at http://www.disit.org/vmsdetl/VMSDETL-2017-v0-8.rar, or the "Versione del 2017/2018 0.8 con Phoenix per Virtualbox" at http://www.disit.org/vmsdetl/VMSDETL-2017-v0-8-ovf.rar, unless you have a good reason for picking a different one
3. Wait for the download to complete, and extract the archive
4. Launch the VM player of your choice
5. Open the VM
6. Run it
7. https://www.snap4city.org/download/video/ETL_and_Console_of_the_Virtual_Machine_-_User_Manual.pdf
8. https://www.snap4city.org/download/video/Snap4city_VM_Quick_guide.pdf

Karma

Launch the Karma server

Do the following to run the Karma server:

1. Open a shell
2. Move to ~/programs/Web-Karma-master/karma-web
3. Run mvn -Djetty.port=9999 jetty:run
4. Wait while the Jetty server comes up
5. Connect to localhost:9999 where you will find the Web application for building your model

Karma

Build the model

For that Karma could produce the RDF triples for you, it is required that you instruct it about how relational data should be mapped to semantic data. Documents that describe such a mapping are called R2RML models. Models are built operating a dedicated Karma Web application, and they are exported as ttl files.

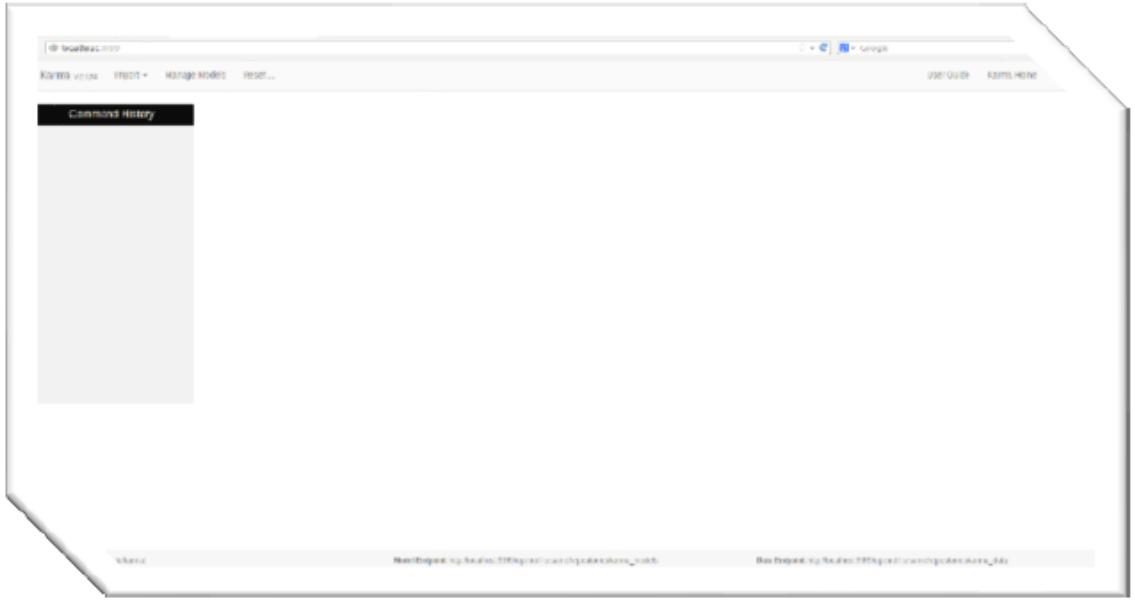
Steps:

1. Load Vocabularies
2. Load Relational Tables
3. Load R2RML Models
4. Define Mappings

Karma

Load Vocabularies

Have you connected to http://localhost:9999? Are you displaying something similar to this? Right, you are ready to load your vocabularies.



Identify classes and properties that you wish to appear in RDF triples that will be the result of the whole process. Identify vocabularies where such classes and properties are defined. Load them.

Below here is how you load a vocabulary:

1. Hit `Import`, at the top left corner of the Web page
2. Hit `From File`
3. Select the vocabulary file (it can be an OWL, RDF/XML, or TTL file)
4. Leave `OWL Ontology` selected, and hit `Next`
5. Indicate the correct file encoding if the proposed one is not, and hit `Import`

You should now see your newly imported vocabulary displayed in the Command History (left column).

Load Relational Tables

Identify tables in your RDB where source data can be found. Load them in your model.

Below here is how you load a table:

1. Hit `Import`, at the top left corner of the Web page
2. Hit `Database Table`. The `Import Database Table` dialog should open.
3. Fill in the form with authentication data and RDB name, and hit `OK`
4. A table listing should appear below the form
5. Put the mouse pointer over the table of your interest
6. Buttons `Import` and `Preview` should appear at the right of the table name. Hit `Import`.
7. Confirmation message "Table imported in the workspace!" should appear. Hit `OK`.
8. Repeat steps 5 – 8 for each table where source data are found
9. Hit `Close` at the bottom right corner of the `Import Database Table` dialog to dismiss it

Karma

Load R2RML Models

If you already have built and exported a model in the past, and you now just need to make a modification over it, you can start loading and applying your existing model, instead of rebuilding it from scratch.

Below here is how you apply an existing model:

1. Identify the table to which the model has to be applied, and hit the triangle that is displayed next to the table name
2. Select `Apply R2RML Model`, and then `From File`
3. Select the ttl file that contains your model, and hit `Open`
4. Done. Classes and links should appear in the workspace.



Karma

Define Mappings

Below here is how you specify that a column of an RDB table maps to a data property of a semantic class, and how to specify that a column contains an identifier that can be used for building the URIs of instances of the semantic class:

1. Below RDB table name, identify the blue box that contains the column name written in white
2. Hit the white triangle that you can see next to the column name
3. Hit `Set Semantic Type`
4. Pick the checkbox at left of *`property of Class`*
5. Hit the `Edit` button at right of *`property of Class`*
6. Select the semantic class from the `All Classes` list. Use the `Class` textbox for filtering.
7. Select the property from the `All Properties` list. Use the `Property` textbox for filtering.
8. If the column is a key, pick the `Mark as key for the class` checkbox
9. You can map the column to a *typed* literal, filling the textbox below the label `Literal type`
10. When you are done, hit `Save`. Repeat the procedure for each of the columns to be mapped.

Karma

Define Mappings

Below here is how you specify instead that a foreign key of a table in a relational database corresponds to an object property of a semantic class. Scenario: a relational table stops, that corresponds to a semantic class Stop, has a column agency_id where the unique identifier of the agency that manages the stop can be found. Each value in agency_id corresponds to one and only one value in a column, let's say *id*, that can be found in the relational table agencies. Table agencies corresponds to the semantic class Agency. We wish resources of class Stop to be linked each to the appropriate resource of class Agency, through the property gtfs:agency. For such a purpose, we will do the following:

1. Load relational table stops to workspace
2. Map data properties, linking columns in table stops, to the class Stop, through appropriate properties, as outlined above
3. Identify the grey box with rounded angles that has the name of the class Stop written within. It should locate in the workspace. Identify the black triangle that should locate near the right margin of the box. Click it.
4. Select `Add Outgoing Link`. A popup window should open.
5. Type `gtfs:agency` in the box labelled `Property`, and `gtfs:Agency` in the box labelled `To Class`
6. Click `Save` in the bottom right corner of the popup window to dismiss it.
7. A new grey box, related to class Agency, will appear in the workspace, linked to the grey box related to class Stop through a link labelled agency for brevity
8. Map column agency_id as a data property of class Agency, also specifying that it is a unique identifier, as described in the above paragraph
9. Repeat for all foreign keys to be mapped, then go to next step (*Export your model*).

Karma
Export Model

Once you have defined all needed mappings, you have to export your model to a ttl file, so that you can provide it as a parameter to the command-line Karma tool that performs the triplification. Here is how you can export your model:

1. Identify the RDB table whose model you wish to export
2. Hit the black triangle at the right of the table name
3. Select Publish, and then Model
4. A popup should appear at the top right corner of the window, saying “R2RML Model published”
5. Hit Manage Models, in the menu bar at the top of the page
6. A listing should appear of all models that you have exported in the current session
7. Identify the row corresponding to the last model exported, based on the File Name (the name of the RDB table) and the Publish Time.
8. Cut the URL that you can find in the rightmost column of the prospect, and open it in a new tab
9. Save As… the page that you have opened at step 8.
10. Done. The file that you have saved at step 9 is your ready-to-use R2RML model.

Karma

Launch the command-line tool

Once you have exported your R2RML model as a ttl file, you are ready to perform the triplification:

1. Open a shell
2. Move to `/home/ubuntu/programs/Web-Karma-master/karma-offline`
3. Launch the following as a single line command, customizing parameter values in bold:

```
mvn exec:java
-Dexec.mainClass="edu.isi.karma.rdf.OfflineRdfGenerator"
-Dexec.args=" --sourcetype DB --modelfilepath /path/to/model.ttl
--outputfile /path/to/output_triples_file.n3 --dbtype MySQL
-hostname mysql_srv_hostname_or_ip_address --username mysql_user
--password mysql_pwd --portnumber 3306 --dbname mysql_dbname
--tablename mysql_table_name -Dexec.classpathScope=compile
```

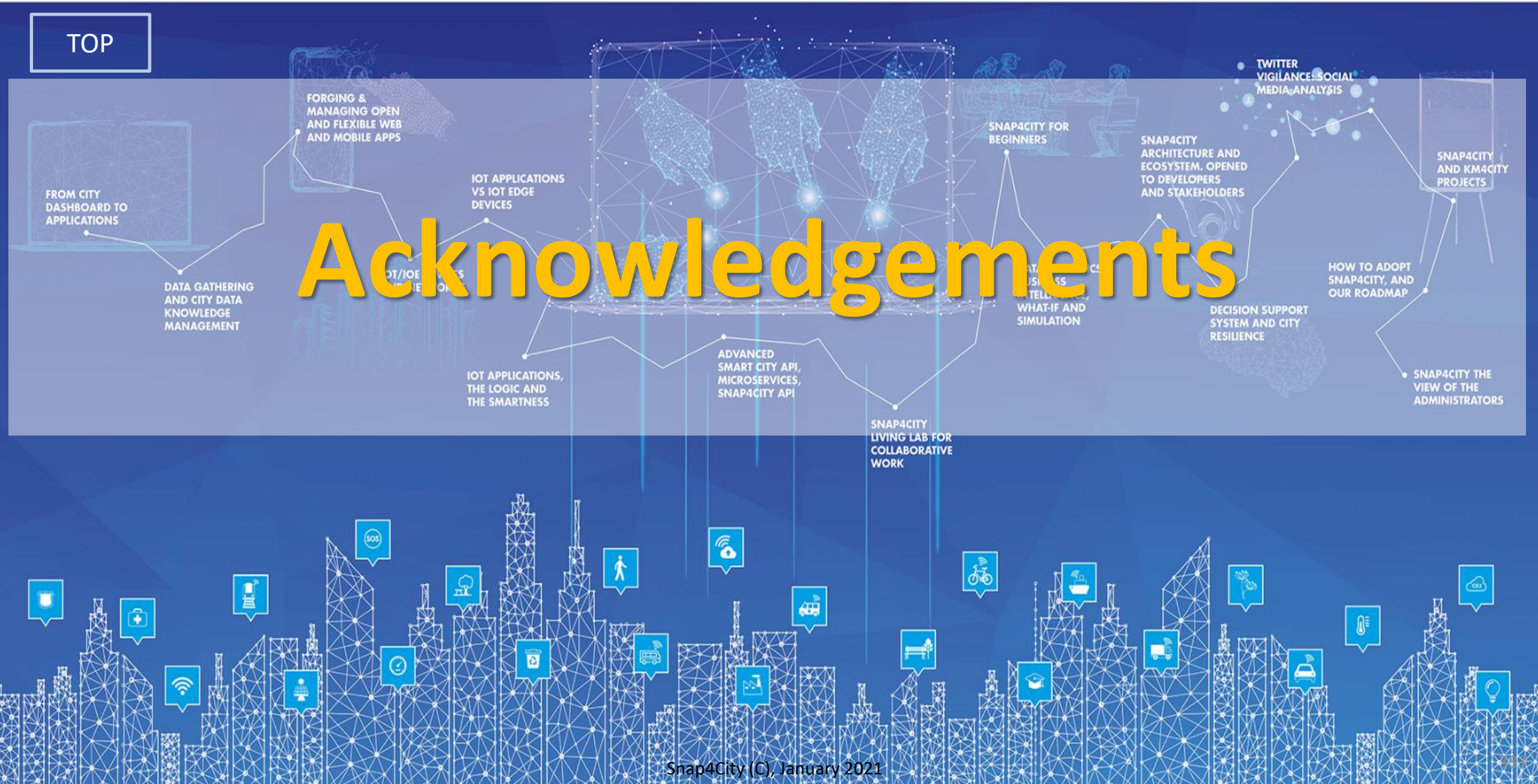
ETL SDK Virtual machine

- https://www.snap4city.org/download/video/ETL_and_Console_of_the_Virtual_Machine_-_User_Manual.pdf
- https://www.snap4city.org/download/video/Snap4city_VM_Quick_guide.pdf
- videos: https://www.snap4city.org/drupal/node/139
- example on Github: https://github.com/disit/smart-city-etl
- https://www.snap4city.org/download/video/Snap4City-ETL-VM.rar
- “Versione del 2017/2018 0.8 con Phoenix” at http://www.disit.org/vmsdetl/VMSDETL-2017-v0-8.rar, or the “Versione del 2017/2018 0.8 con Phoenix per Virtualbox” at http://www.disit.org/vmsdetl/VMSDETL-2017-v0-8-ovf.rar, unless you have a good reason for picking a different one

Further readings on ETL

- ETL Development: https://www.snap4city.org/drupal/node/24
- US6. Developing and using processes for data transformation
- HOW TO: add a device to the Snap4City Platform
- HOW TO: add data sources to the Snap4City Platform
- TC6.3. Creating ETL processes for automated data ingestion and data transformation
- TC6.4- Managing ETL processes via Resource Manager, upload, execute, monitor
- TC6.5- Managing Heterogeneous File Ingestion via ETL processes
- TC6.6- Producing data-sets in Bundle via ETL
- TC6.8- ETL processes for data transformation, and exploiting MicroServices/API/RestCall
- TC6.9- ETL processes for multiprotocol and format data ingestion, see on GITHUB for library
- TC6.10- ETL Applications using multiple protocols, and formats for files and to calling services using REST and WS
- TC6.11- Add a new ETL coping with a new Protocol
- ETL processes for massive Data Ingestion and Transformation

TOP



Acknowledgements

DISIT lab roadmap vs model and tools' usage

2013
Km4City Ontology 1.1

- Tuscany, Road Graph
- Mobility
- culture, tourism
- Events
- Parking
- Services
- Linked open graph

2014
- Weather Forecast
- Real Time Wi-Fi
- Entertainment
- LOD

- Twitter Vigilance
- Social Media Analytics, Sentiment Analysis

Km4City 1.4

2015
Km4City 1.5

- Resilience Decision Support
- Smart First Aid
- User Behaviour Analysis, predictions
- Risk Analysis

ckan, resolute

H2020 (2015-18)

2016
(2016-21) H2020 REPLICATE

- Smart Energy
- Sustainable Mobility
- Control Room
- Dashboard

Km4City 1.6.2

Sii-Mobility (2016-21) MIUR

SII-MOBILITY SCN

- Infomobility
- Mobile App
- Routing
- Multimodality

FIWARE

2017
Km4City 1.6.4

Sii-Mobility

- Origin-Destination and trajectories
- Traffic Reconstruction
- Offer Analysis
- OBU, smart devices

E0l5 digital ecosystem (Powered by EXPO MILANO 2015)

(2016-19) MIUR

GHOST SIR

- Sardinia Region Smart City Strategies and plan

GREEN IMPACT C.A.P.A.C.I.T.Y.

GREEN IMPACT
POR FESR 2014-2020

- Industry 4.0
- Critical Plant
- Monitoring

- IOT/IOE, IOT App
- Living Lab
- Maker Support
- IOT Edge
- Smart City IOT
- GDPR,
- Privacy & Security

life weee (2017-20)

- Smart Waste

2018
MOSAiC MObility 4.0 for SmArt (i) City (2018-20)

- Mobility Demand / Offer Analytics and Strategy

Sii-Mobility

- User engagement
- Bike Sharing
- Data Analytics ++
- Social Predictions
- OBD2

IOT/IOE

Km4City 1.6.6

Node-RED

SNAP4CITY

SELECT for Cities H2020 (2017-19)

2019
5G (2018-21)

- 5G tech
- Energy
- Industry 4.0
- Synoptics

TRAFAIR

CEF (2018-21)

- Traffic and Mobility Impact on Pollution
- NOX predictions

EUROPEAN OPEN SCIENCE CLOUD

SIRIUS

LAID Smart Bed

- Smart Health

GREEN FIELD PEAS Soda4.0 (NaOH) Optimization and Automated Decision System 4.0

- Industry 4.0

bee smart city

- Smart Lonato

Reverberi Enetec Gruppo MPES

- PCP Award



2020
…2022

2021
- CAPELON
- Sweden

smartGARDAlake

Interreg MARITTIMO-IT FR-MARITIME MOBI MART

- Smart Mobility
- PISA, PUMS
- Living lab

Interreg Mediterranean HERIT-DATA

- Smart Tourism
- 6 Pilots
- Data Analytics
- Extended platform

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

Main running instances

- Sii-Mobility → mobility and transport, sustainability
- REPLICATE → ICT, smart City Control room, Energy, IOT
- RESOLUTE → Resilience, ICT, Big Data
- GHOST → Strategies, smart city
- TRAFAIR → Environment & transport
- MOSAIC → mobility and transport
- WEEE Life → Smart waste, environment
- Smart Garda Lake → Castelnuovo del Garda
- 5G → Industry 4.0 vs SmartCity
- Green Impact → Industry 4.0, Chemical Plant
- SmartBed (laid → smart health
- Green Field Peas (soda) → Industry 4.0, Chemical plant
- MobiMart and PISA Agreement → data aggregation, Living Lab
- Lonato del Garda → smart parking, environment
- Herit Data → tourism, culture and management
- ISPRA JRC → site management and services
- Capelon (Sweden) → smart light solutions

Acknowledgements

- Thanks to the European Commission for founding. All slides reporting logo of **Snap4City** https://www.snap4city.org **of Select4Cities H2020** are representing tools and research founded by European Commission for the **Select4Cities** project. **Select4Cities** has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation Programme (grant agreement n° 688196)
- **TRAFAIR is a CEF** project. All slides reporting logo of TRAFAIR project are representing tools and research founded by the EC on CEF programme http://trafair.eu/
- Thanks to the European Commission for founding. All slides reporting logo of **REPLICATE H2020** are representing tools and research founded by European Commission for the REPLICATE project. **REPLICATE** has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation Programme (grant agreement n° 691735).
- Thanks to the European Commission for founding. All slides reporting logo of **RESOLUTE H2020** are representing tools and research founded by European Commission for the RESOLUTE project. **RESOLUTE** has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation Programme (grant agreement n° 653460).
- Thanks to the MIUR for co-founding and to the University of Florence and companies involved. All slides reporting logo of **Sii-Mobility** are representing tools and research founded by MIUR for the Sii-Mobility SCN MIUR project.
- **Km4City** *is an open technology and research line of DISIT Lab exploited by a number of projects. Some of the innovative solutions and research issues developed into projects are also compliant and contributing to the Km4City approach and thus are released as open sources and are interoperable, scalable, modular, standard compliant, etc.*

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

DISIT thanks to

Herit Data: Tourism and Mng. https://herit-data.interreg-med.eu/
Snap4City: IOT/IOE smart city www.snap4city.org
Trafair: CEF project with several Cities http://trafair.eu/
Mosaic: Mobility and transport model
Km4City: http://www.km4city.org
REPLICATE H2020, SCC1, EC flagship
http://replicate-project.eu/
Sii-Mobility SCN MIUR: http://www.sii-mobility.org
Feedback: retail and GDO Big Data analytics
5G with 3G-Wind, Open Fiber, Estra
Coll@bora Social Innovation, MIUR:
http://www.disit.org/5479
RESOLUTE H2020, EC:
http://www.resolute-eu.org
TRACE-IT, RAISSS, TESYSRAIL, …
Mobile Emergency:
http://www.disit.org/5404

KM4CITY
SNAP4CITY
REPLICATE
resolute
Sii-Mobility
TRAFAIR
GREEN IMPACT C.A.P.A.C.I.T.Y.
LAID Smart Bed
GREEN FIELD PEAS Soda4.0
5G
MOSAiC
Coll@bora
TESYS RAIL
feedback
Trace-IT
RAISSS
Smart COMMUNITIES Tech Italian Technology Cluster for Smart Communities
Regione Toscana
MIUR
Horizon 2020 European Union Funding for Research & Innovation

Further readings

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

TOP





Be smart in a SNAP!





CONTACT

DISIT Lab, DINFO: Department of Information Engineering
Università degli Studi di Firenze - School of Engineering

Via S. Marta, 3 - 50139 Firenze, ITALY
https://www.disit.org

www.snap4city.org

Email: snap4city@disit.org

Office: +39-055-2758-515 / 517
Cell: +39-335-566-86-74
Fax.: +39-055-2758570

