





Powered by FIWARE















SMARTCITY

EXPO WORLD CONGRESS

Visit Snap4City in Hall 1

Exploiting Snap4City API, and Web/Mobile Applications SDK



Sept. 2023, Course, Part 7

https://www.snap4city.org/944

https://www.snap4city.org/577

SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES

















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

https://www.Snap4City.org





https://www.snap4city.org/944

https://www.snap4city.org/577

Paolo Nesi, paolo.nesi@unifi.it

https://www.Km4City.org

https://www.disit.org













Work with US: Open 3 Grants for PHD Course

- https://www.snap4city.org/581
- 2 Grants from late 2023-2027 on:
 - Study of AI tools for the automatic generation of scenarios for the resolution of critical conditions in complex geolocated structures.
 - Deadline: August 7, 2023, h 13:00 CET
- 1 Grant from late 2023-2027 on:
 - Study of artificial intelligence techniques integrated with visual analytics and business intelligence tools at support of Public Administrations decisions.
 - Deadline: August 11, 2023, h 12:00 CET





FREE TRIAL



















SMART SOLUTIONS AND DECISION SUPPORT SYSTEMS



DASHBOARDS - VISUAL ANALYTICS - SYNOPTICS - DIGITAL TWIN - GRAPHICAL WIDGETS - ANALYTICS - GUI CUSTOM STYLES VISUAL PROGRAMMING

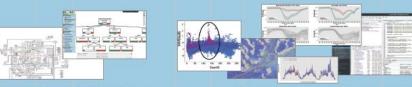


DASHBOARDS, WIDGETS TEMPLATES

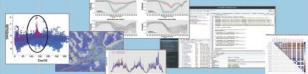
PREDICTION - ANOMALY DETECTION - CLUSTERING - ROUTING - SENTIMENT NLP - TRAFFIC FLOW PEOPLE FLOWS - SDG - 15 MIN CITY INDEX - KPI - HEATMAPS - ORIGIN DESTINATION - ETC..

API - MICROSERVICES - GIS - BPM VIDEO - REPORTS - MAPS - 3D ...





EXPERT SYSTEM, KNOWLEDGE BASE SEMANTIC REASONING **SMART DATA MODEL IOT DEVICE MODELS, STORAGE**



BIG DATA ANALYTICS, ARTIFICIAL INTELLIGENCE EXPLAINABLE AI, MACHINE LEARNING OPERATIVE RESEARCH, STATISTICS



VISUAL PROGRAMMING, ADAPTERS DATA FLOWS, WORKFLOWS PARALLEL DISTRIBUTED PROCESSING **DATA DRIVEN**



Smart Parking

Smart Light

Smart Waste

Smart Energy

Social Media Analysis









https://www.snap4city.org/577



On Line Training Material (free of charge)

https://www.snap4city.org/944

	1					1		
	1st part	2nd part	3rd part	4th part	5th part	6th part	7th part	8th
what	Overview	Dashboards	IOT App, IOT Network	Data Analytics	Data Ingestion processes	System and Deploy Install	Smart City API: Web & Mob. App	Design and Develo Smart Solutions
PDF 2022	C DADAM TO THE BANK OF THE BAN	CHAMEN S	CENANAGOR SEA	CENAMOR STATE OF THE PROPERTY	C SMADA m Part has DAAP	CEMANDON ESTABLISHED TO THE PROPERTY OF THE PR	CSMADdon Service to State Stat	CENANTORY OF THE SAME OF THE S
Interactive (2022) with video and animations	C'SHAPAon Service to 3 SOAT	CENAMON STATE OF THE PARTY OF T	COLANGE STATE OF THE STATE OF T	CERANDON DE SOAR DE SO	CHARMEN STATE OF THE STATE OF T	C'SHAPAGY STORY IS SOUTH TO SHAPAGY STORY IN SHAPAGY STORY IS SOUTH TO SHAPAGY STORY IN SHAPAGY STORY IS SOUTH TO SHAPAGY STORY IN SHA	CENANACIO DE LOS DEL LOS DE LOS DEL LOS DELLOS DEL LOS DELLOS DEL LOS DELLOS DEL LOS DELLOS DELLO	CENANDO DE SOLATION DE SOLATIO
Videol	You Tube	You	You		You		You	You
Video2	You Tube	You	You		You	V _{OI}	You	You
Video3	You Tube	You	You		You	You	You	You
Video4	You Tuhe	You Tube	You	nor	pur	You	none	none









Note on Training Material

- Course 2023: https://www.snap4city.org/944
 - Introductionary course to Snap4City technology
- Course https://www.snap4city.org/577
 - Full training course with much more details on mechanisms and a wider set of cases/solutions of the Snap4City Technology
- Documentation includes a deeper round of details
 - Snap4City Platform Overview:
 - https://www.snap4city.org/drupal/sites/default/files/files/Snap4City-PlatformOverview.pdf
 - Development Life Cycle:
 - https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf
 - Client Side Business Logic:
 - https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf
- On line cases and documentation:
 - https://www.snap4city.org/108
 - https://www.snap4city.org/78
 - https://www.snap4city.org/426





















Snap4City Platform

Technical Overview

From: DINFO dept of University of Florence, with its

DISIT Lab, Https://www.disit.org with its Snap4City solution

Snap4City:

- Web page: <u>Https://www.snap4city.org</u>
- https://twitter.com/snap4city
- https://www.facebook.com/snap4city

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

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



Tech Overview

https://www.snap4city.o rg/drupal/sites/default/f iles/files/Snap4City-PlatformOverview.pdf













Development Life-Cycle

https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle-v1-1.pdf

From Snap4City:

- We suggest you to read the TECHNICAL OVERVIEW:
 - https://www.snap4city.org/download/video/Snap4City-
- https://www.snap4city.org
- https://www.snap4industrv.org
- https://twitter.com/snap4city
- https://www.facebook.com/snap4city
- https://www.youtube.com/channel/UC3tAO09EbNba8f2-u4vandg

Coordinator: Paolo Nesi, Paolo.nesi@unifi.it

DISIT Lab, https://www.disit.org DINFO dept of University of Florence, Via S. Marta 3, 50139, Firenze, Italy Phone: +39-335-5668674









Development

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







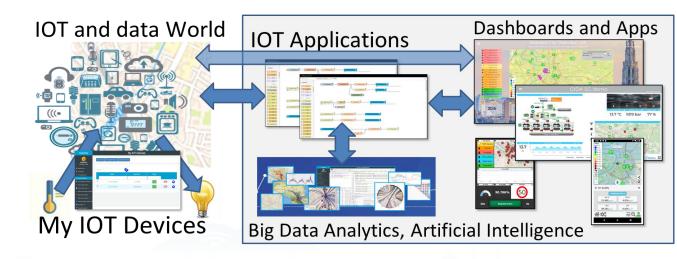






T TECHNOLOGIES LAB ITECHNOLOGIES LAB ITECHNOLOGIES LAB ITECHNOLOGIES LAB ITECHNOLOGIES LAB ITECHNOLOGIES LAB

- 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.









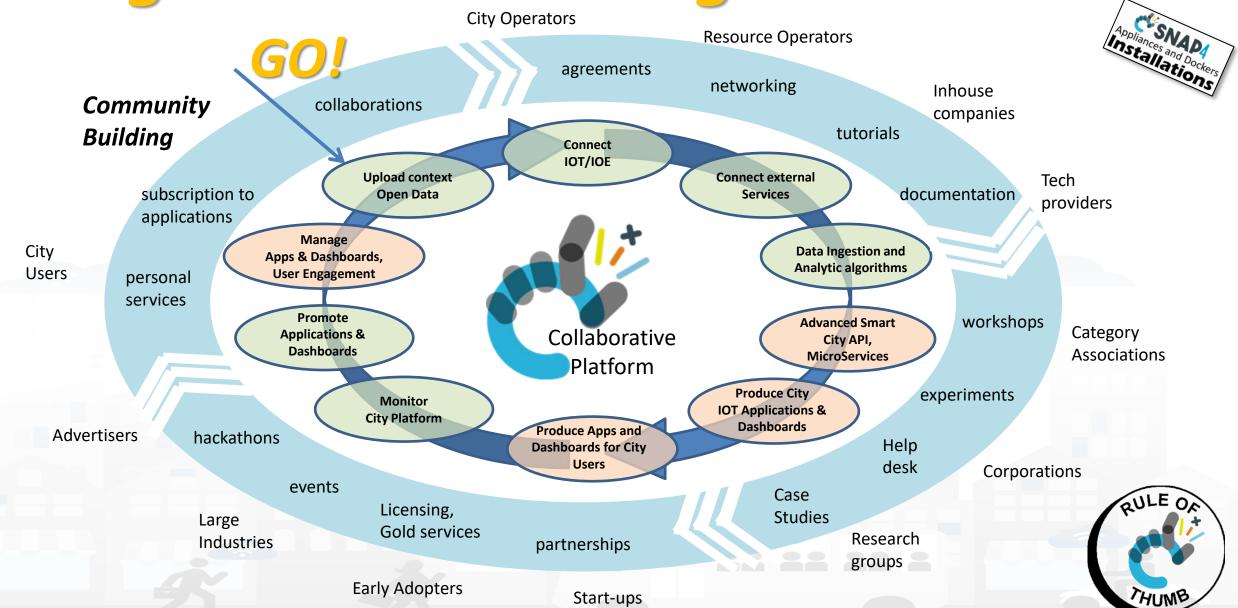


Agenda of Part 7

- Smart City API: Internal and External
- Concepts and tools for using Knowlege Base, ServiceMap, API
- Federated Knowledge Bases and Smart City APIs
- Advanced Smart City API
- Access to Protected data
- Forging and managing: Mobile and Web Apps, MicroApplications
- Web and Mobile App Development Kit
- Training Material

Living Lab Accelerating





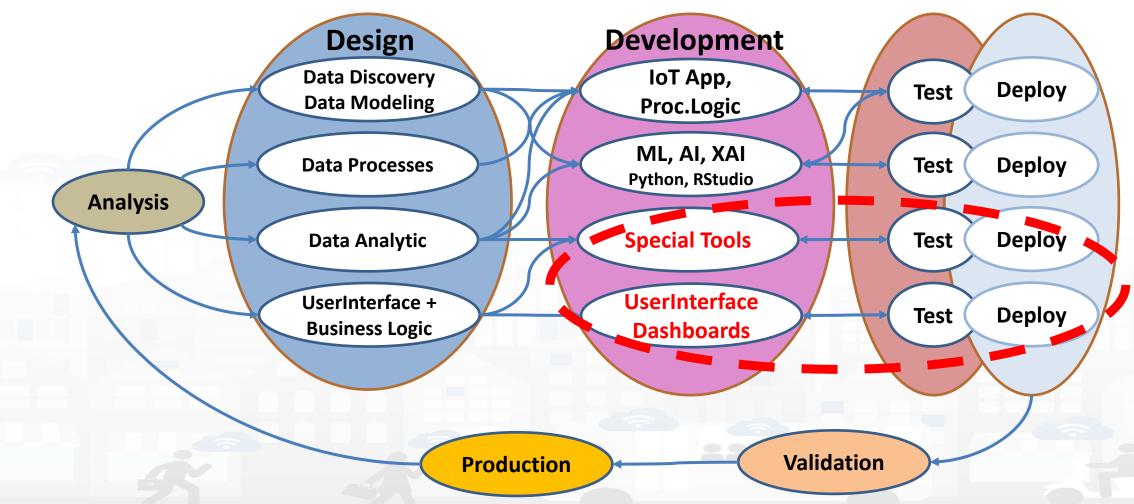








Development Life Cycle Smart Solutions



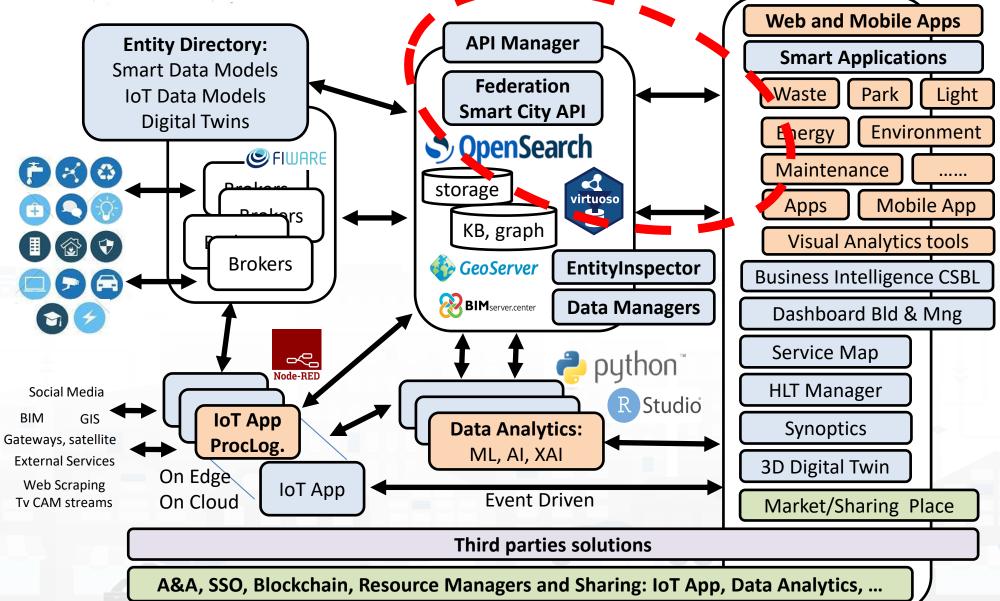






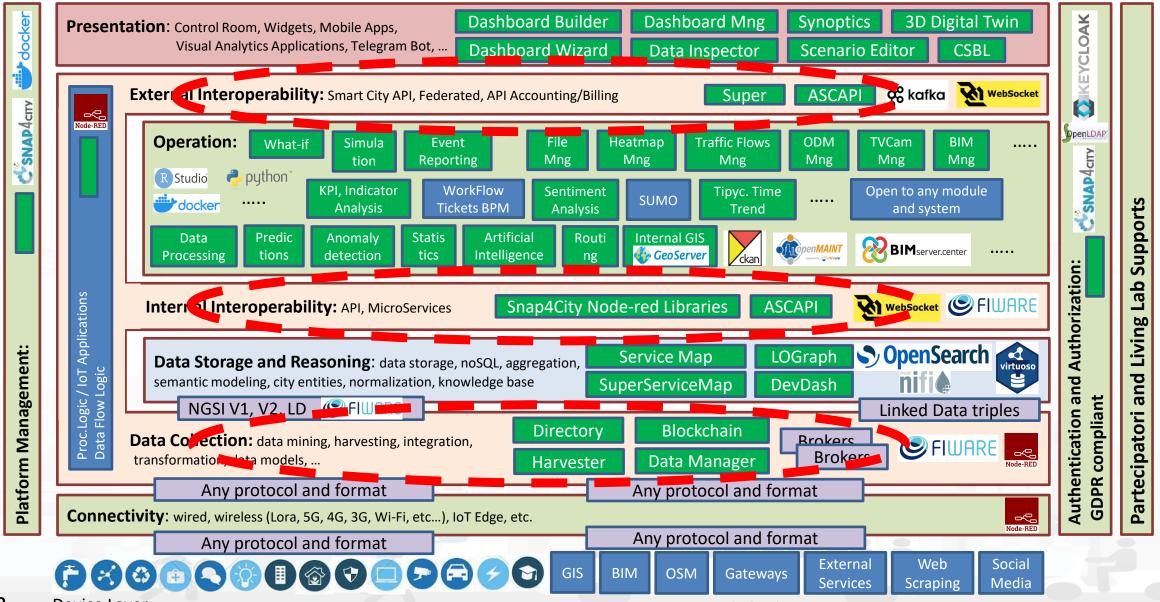








legenda









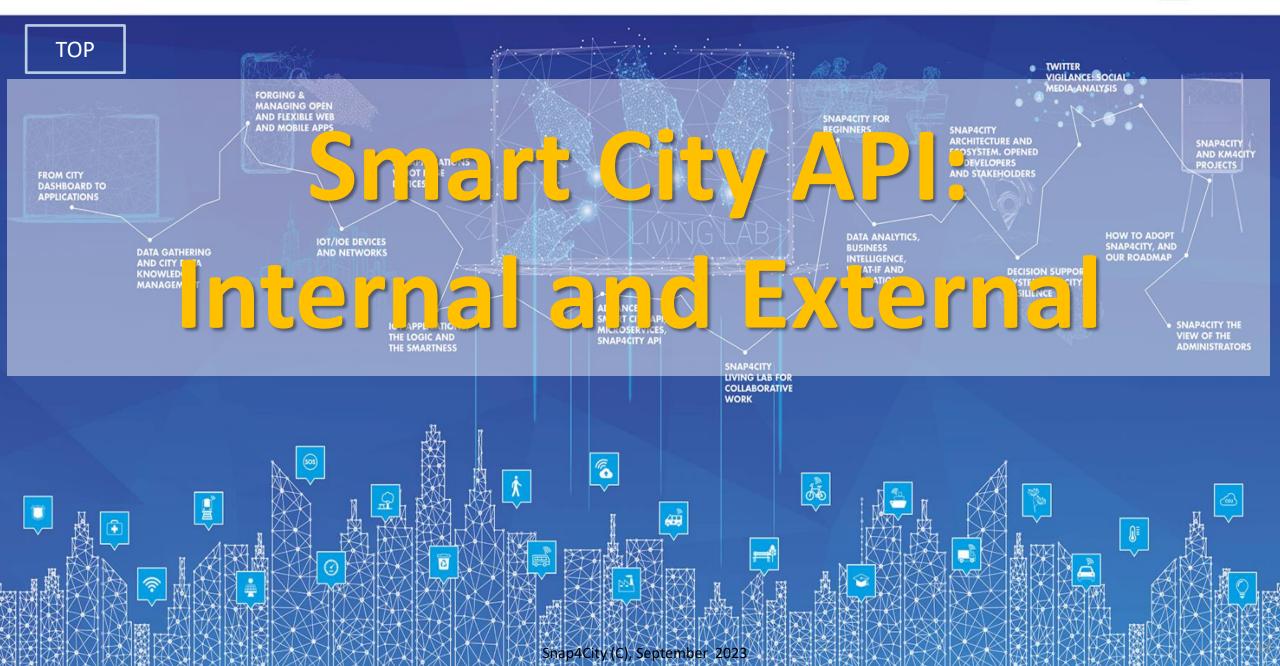




- Smart Applications can be easily developed exploiting the cloud infrastructure by producing only:
 - Processing Logic / IoT App with almost no coding activities
 - Data Analytics in Python or Rstudio
 - Dashboards with almost no coding activities.
- > Orange parts of the previous figure slide are those usually developed,
 - all the rest, is part of the provided microservices and infrastructure.
- Third party applications can dialog with the solutions via
 - Smart City API, Swagger: https://www.km4city.org/swagger/external/
 - Brokers/IoT Brokers, for example for NGSI Orion Broker:
 https://www.km4city.org/swagger/external/?urls.primaryName=Orion%20Broker%20K1-K2%20Authentication%20API
 - Processing Logic / IoT App any protocols: https://www.snap4city.org/65 They can also expose some specific API, custom made

SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES





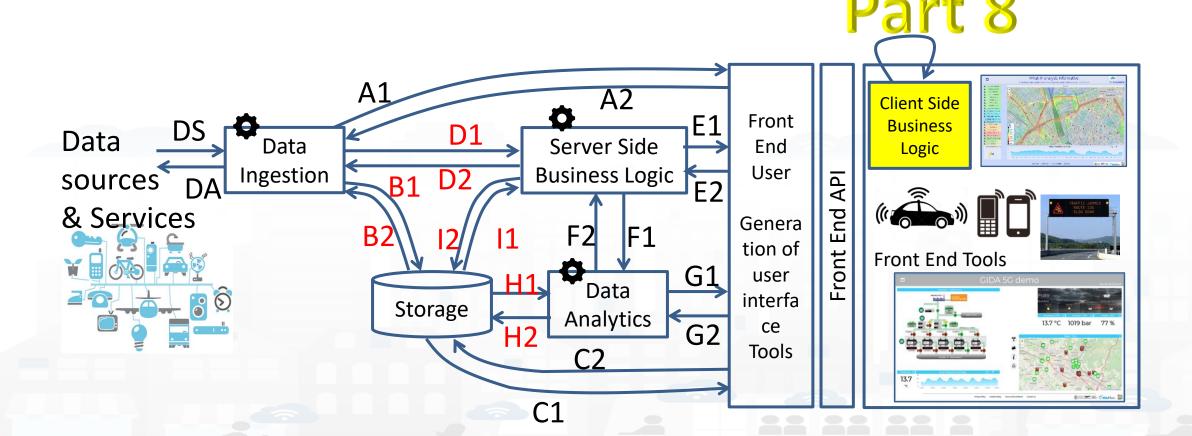








Internal API



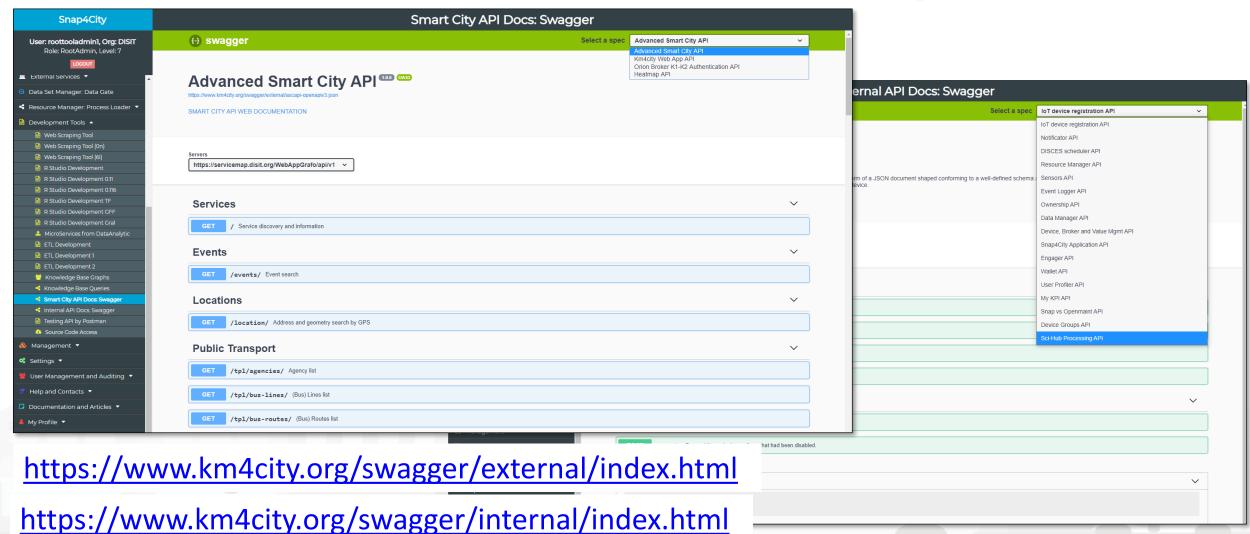








Internal and External Smart City API



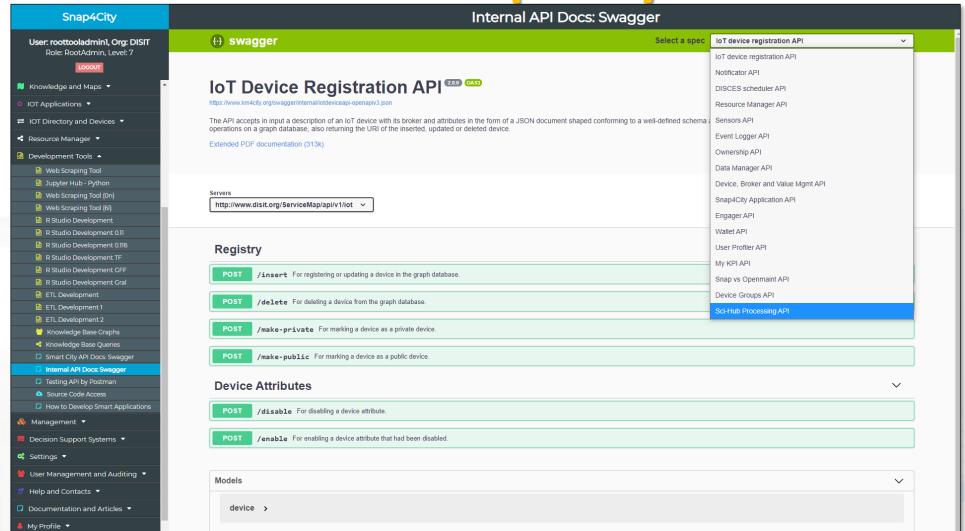








Internal Snap4City API



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









Internal Smart City API

- IOT devices / Entities and tools API:
 - IoT device /entity registration API
 - API of the IoT / Entity Directory
 - Sensors API
 - API of the IoT / Entity Directory
 - Device, Broker and Value Mgmt API
 - API of the IoT / Entity Directory
- Mobile App management
 - User Profiler API
 - To manage the user profile for the Engager on Mobile Apps
 - Engager API
 - From the Engager to prepare engagements to the Mobile Apps
 - Wallet API
 - From the Engager to Wallet o the users of Mobile Apps and in general
 - Snap4City Application API











Internal Smart City AP

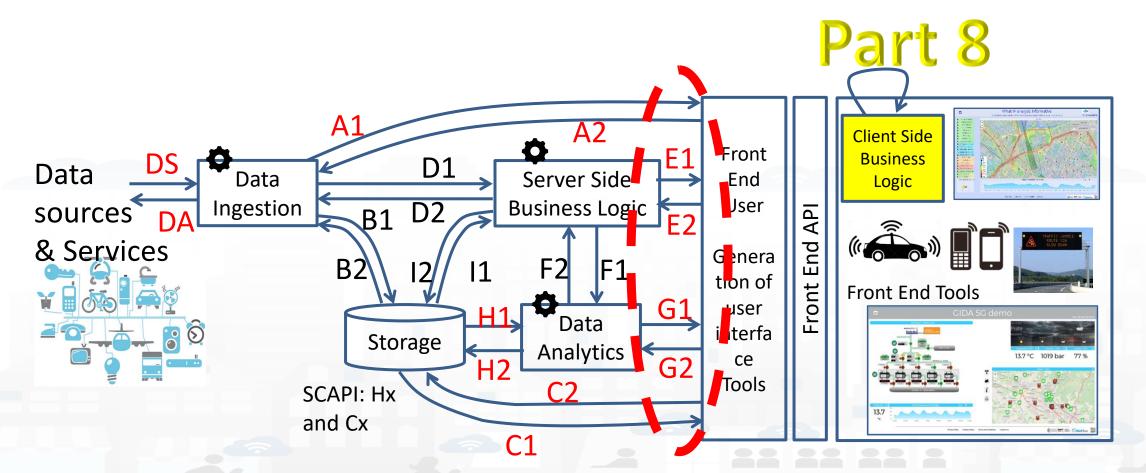
- Resources and entities (Partially usable also as External API)
 - Snap4City Application API
 - To manage IOT Apps, Proc.Logic
 - My KPI API
 - To manage MyKPI, MyPOI, POI, etc.
 - **Data Manager API**
 - Resource Manager API
 - To manage resources on the market place
 - Ownership API
 - To manage ownerships and delegations
 - Device Groups API
 - To manage ownerships and delegations
- **Notificator API**
- **DISCES** scheduler API (deprecated)
- **Event Logger API (partially deprecated)**
- **Snap vs OpenMaint API**
 - Integration with the workflow management and ticketing
- **SCI-HUB Processing API**
 - To activate data download and heatmap production from Copernicus satellite services







Snap4City External API











Snap4City API families

- C1-C2 / H1-H2: ASCAPI, Advanced Smart City API + Data Managers
 - Mainly to access at data, pose query, etc.
- A1-A2: broker connections
 - Mainly access to send data, get data here is without storage !!!!
- E1-E2: Server Side Business Logic, Proc.Logic/IoT App, Node-red
 - Envent Driven data INPUT/OUTPUT, via WS
- G1-G2: Dedicated Connections with Data Analytics, Custom communications
- Internal API:
 - All the others

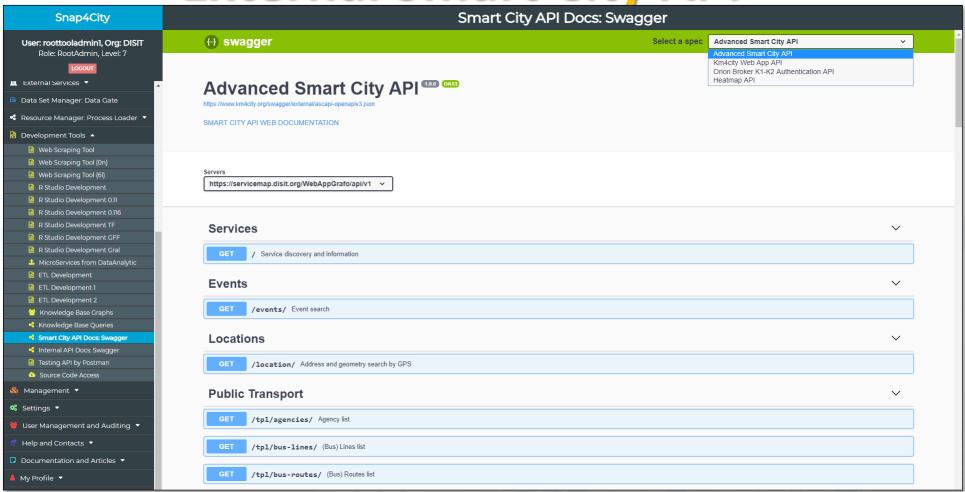








External Smart City API



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







External Smart City API

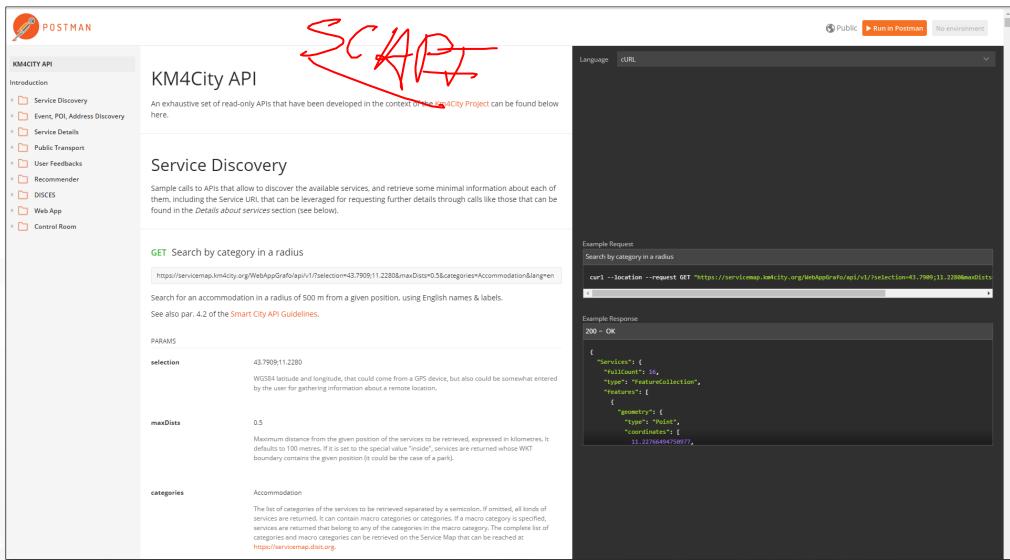
- Advanced Smart City API, normal or Super
 - To access the Service Map resources and query
- Orion Broker K1-K2 Authentication, etc., SSO
 - To communicate with IOT Orion Brokers exploiting the Secure Filter of Snap4City.
- Heatmap
 - To save and access to HeatMaps of the Heatmap server
- Other: ODM, Traffic Flow, etc.
- Km4city Web App API
 - To exploit MicroApplications created as tools for Dashboards, totem, web Apps, etc.





Postman





https://documenter.getpostman.com/view/4177198/km4city-api/RW83QsX5?version=latest







- Advanced Smart City API which can be confined into a single Smart City installation or Federated as well as for **Super Service Map**
 - https://www.km4city.org/swagger/external/index.html
- **Federated Multiple Snap4City** Knowledge Bases. This allows the creation of mobile applications that may move from multiple cities and area accessing data and making queries transparently. This solution is presently in place among the Knowledge Bases of: Antwerp/Helsinki, Tuscany/Firenze, Sardegna, etc. The resulting Service is called Super Service Map and it is integrated in the Smart City API. For example, via:
 - https://www.disit.org/superservicemap/api/v1
- **Federated Open Data Portals** via DataGate/CKAN that presently presents now more than 13800 data sets linked for the cities of Helsinki and Antwerp.
 - https://datagate.snap4city.org/organization
 - Federation, Harvesting interface is: https://datagate.snap4city.org/harvest
- WFS service of Snap4City on top of Federated Smart City API or simple Smart City API of a single ServiceMap (smart City installation). This solution permits to GIS applications and platforms (such as ArcGIS OnLine ESRI, ArcGIS Enterprise ESRI, ArcGIS Map/pro Desktop, QGIS, GeoServer, etc.) to access at Snap4City data. For Example, via:
 - https://www.disit.org/superservicemap/api/v1/wfs
 - https://www.disit.org/superservicemap/api/v1/wfs?service=WFS&request=GetCapabilities&version=2.0.0
- WMS service of Snap4City for publishing maps and heatmaps, provided by an installed GeoServer third party open source tool. For example, via:
 - https://wmsserver.snap4city.org/geoserver/Snap4City/wms
 - https://www.km4city.org/swagger/external/index.html?urls.primaryName=Heatmap%20API



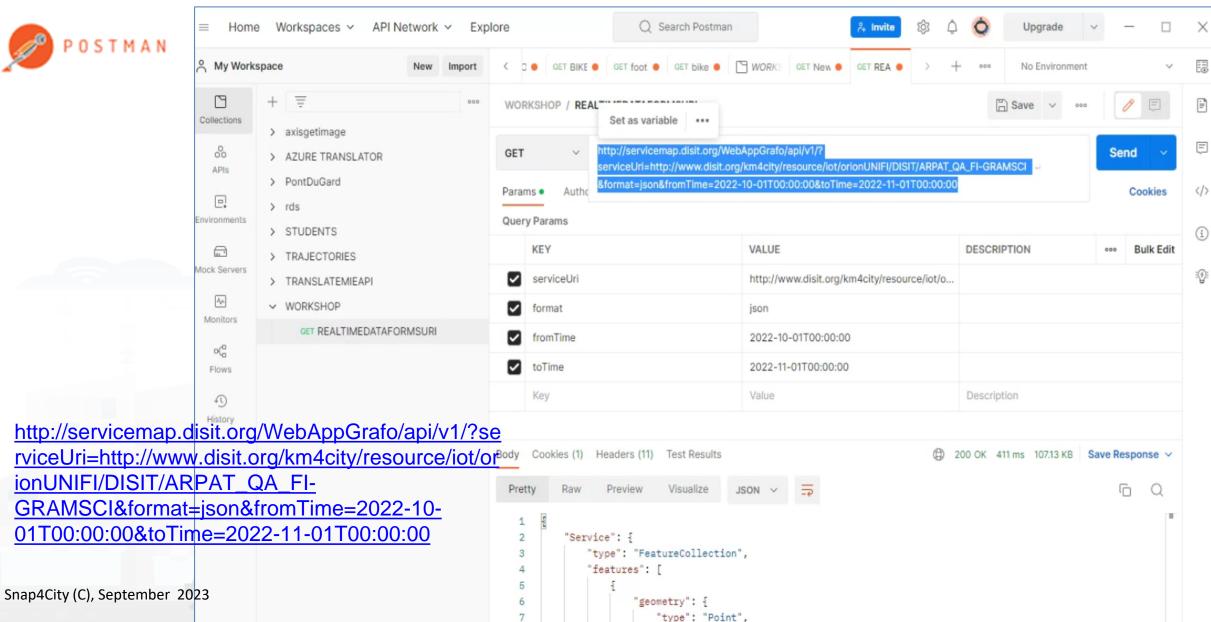




Test the API







SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES







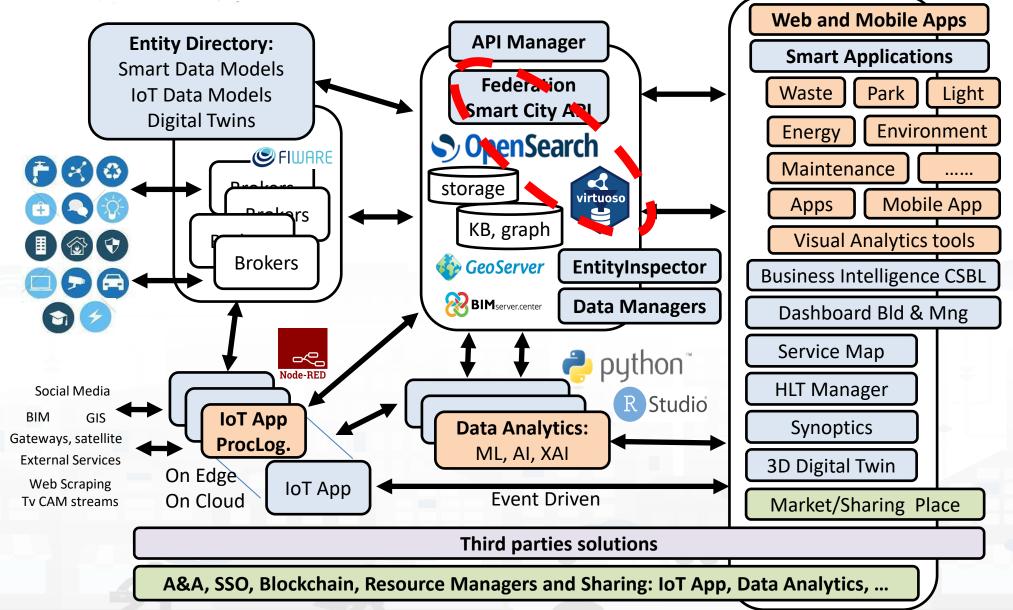


DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB DISTRIBUTED DATA INTELLIGENCE AND TECHNOLOGIES LAB

Tech Arch







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

Ontology Documentation:

http://www.disit.org/6506

http://www.disit.org/6507

http://www.disit.org/5606

http://www.disit.org/6461







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







Big Data Tools

LOD and

reasoners

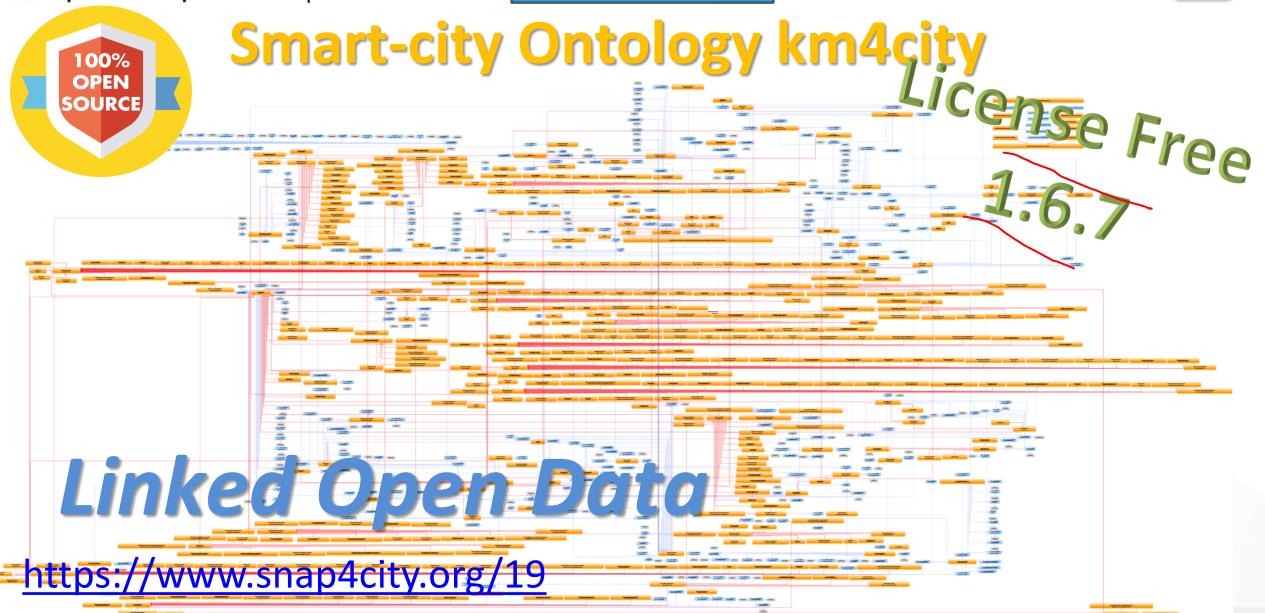








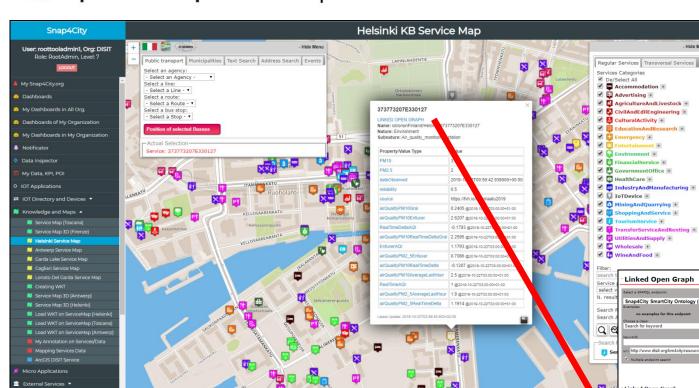








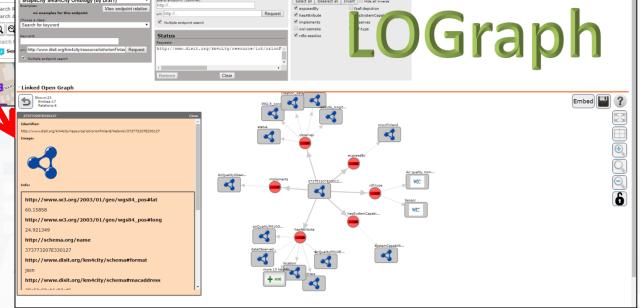




Views of the Knowledge Base

Knowledge Base
Semantic Reasoners

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

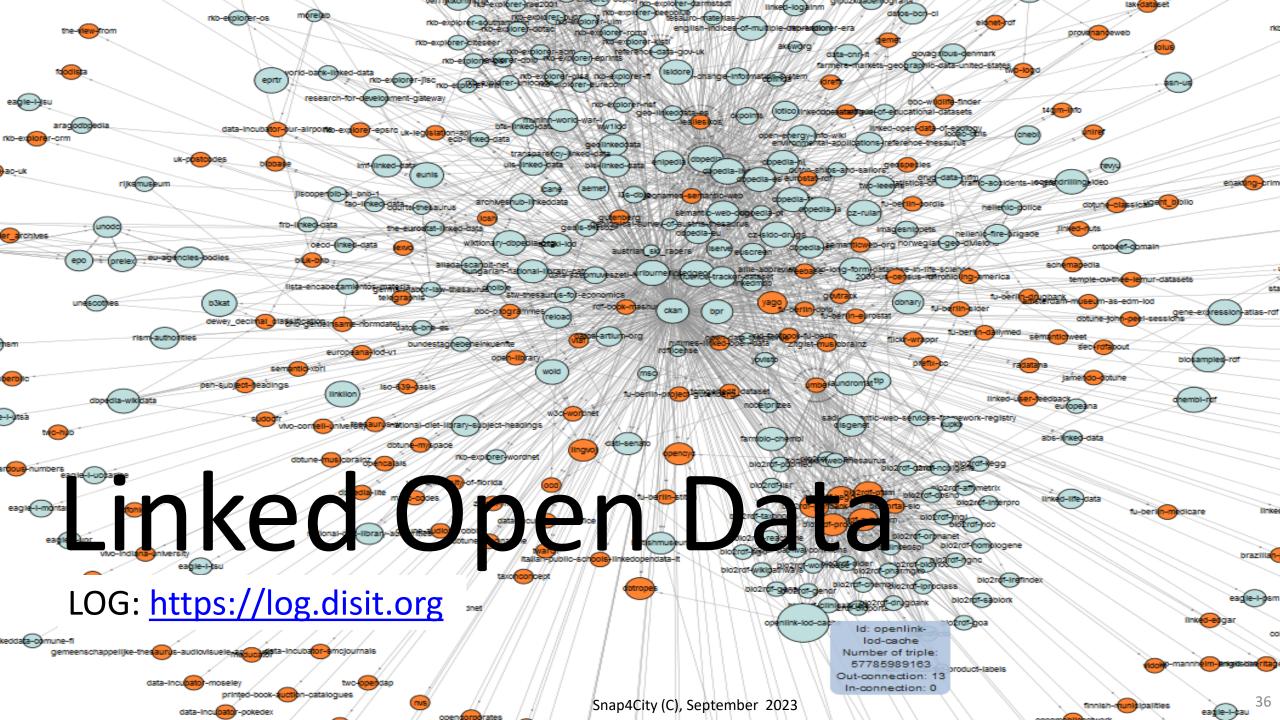






Linked Open Graph LOG: https://log.disit.org

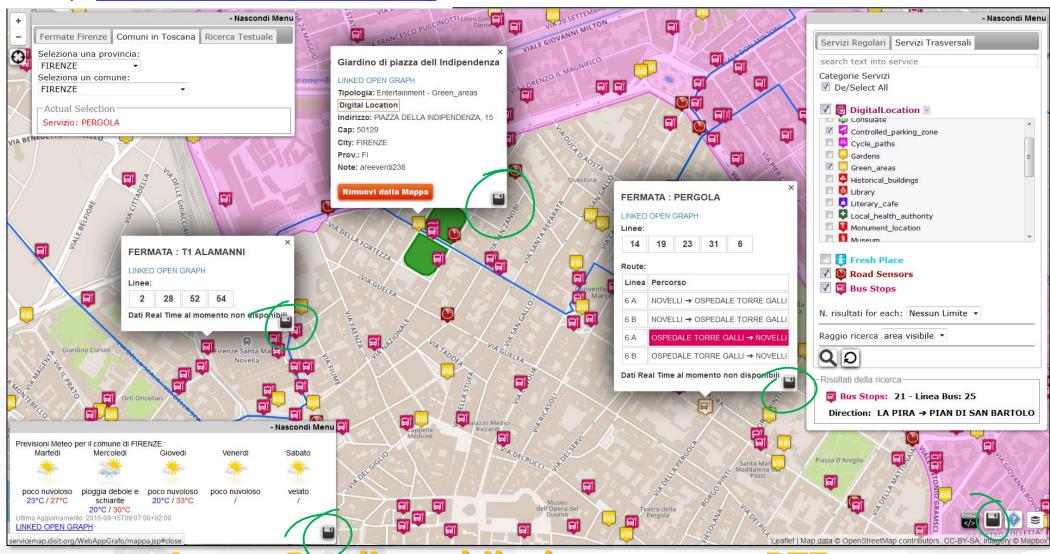
Mobility (by DISIT)	Your data	Type of relations	
nples: • VIA GIACOMO MATTEOTTI • Bagno a ripoli	sparql endpoint: (optional) http://	Select all Deselect all Invert Hide all inverse	
• Florence	uri: http:// Request	 ✓ belongTo ✓ coincideWith ✓ contains ✓ depiction 	
ose a class: arch for keyword	Troquot Troquot	ends forming	
		□ has ☑ hasAccess	
word:	Status	 ✓ hasExternalAccess ✓ hasMunicipality ✓ hasProvince ✓ hasRule 	
	Requests:	■ nasprovince ■ naskule ■ hasStreetNumber ■ inMunicipalityOf	
http:// Request	http://www.disit.dinfo.unifi.it/SiiMobility/MUSE _	☑ isIn ☑ isPartOf	
		✓ isPartOfProvince ✓ isPartOfRegion	
	1 <u> </u>	✓ placedIn □ sameAs	
	Remove	✓ seeAlso □ starts ↓	
nked Open Graph			_
PISTOIA	type		-]
has Province is Part Of Region	isIn type	has External Acce type	ĺ
type	coincideWith	SEO_SALVATORE hasAccess RT048017017682A	
FIRENZE	hasStreetNumbeR,T04801702380TO	SEO_SALVATORE hasAccess RT048017017682A	
hasMunicipality	isPartOf	placedIn	
ownerAuthority is Part Of Provinc	contains	04801723494ES type	
managing Authori FIREN	type	museo ferragamo	
	RT04801708991EETQ4801724784EET,048017247	2	ne roa
managingAuthori	🕒 🙀 🕰 🕰	managingAuthori	



Some structures from Km4City model



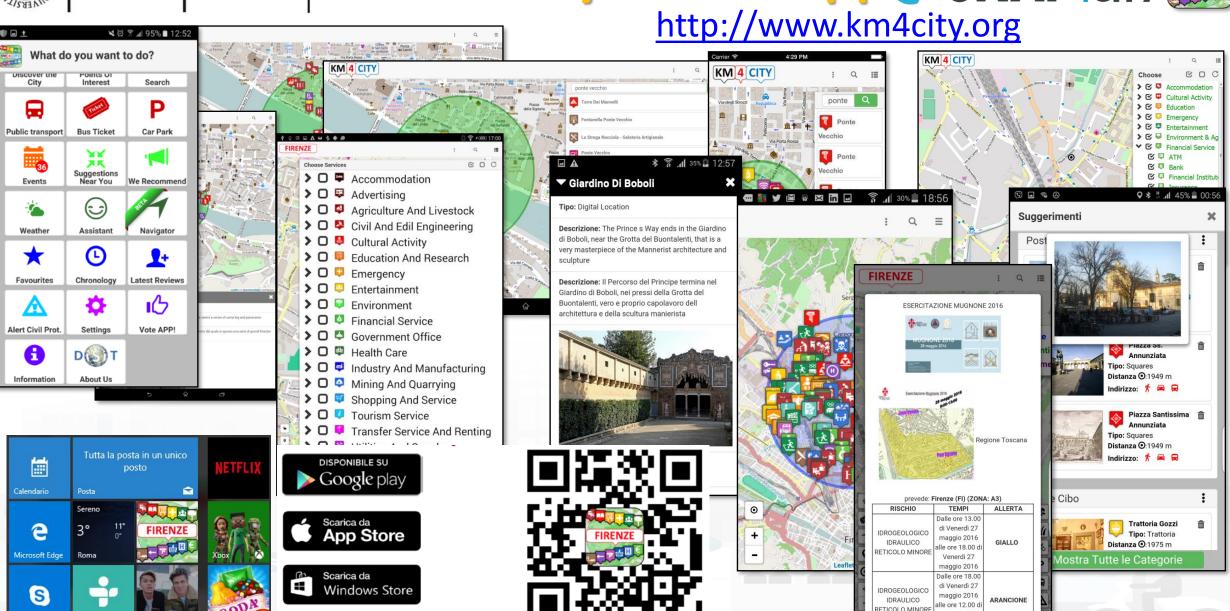
ServiceMap: https://servicemap.km4city.org





DISIT MACITY Obile App ST SNAP4city MADINTERNET TECHNOLOGIES LA MACITY TECHNOLOGIES LA MACI





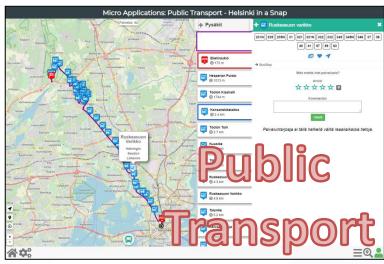


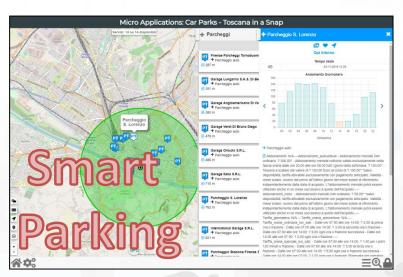
DISIT DISTRIBUTED SYSTEMS VICTOAPPLICATIONS CSNAP4CITY KM 4 CITY AND INTERNET TECHNOLOGIES LAB VICTOAPPLICATIONS



















Advanced Devevelopment Kit features

- Exploiting Km4City Advanced Smart City API
 - Open Source: GitHub
 - Multiplatform: exploiting Apache Cordova Framework
 - Active since 2015
 - Adopted by a community of several Projects, Cities and SME
- Respecting user privacy:
 - Anonymous usage vs Authenticated usage (OAuth, email, ...)
- Modular & Dynamic:
 - Loading new modules from the WEB, and/or creating App by modular approach
- Personalization and Profiling:
 - Personalized menu, proposed POI for search
- Reaching City Users:
 - Alerting and notifications by location, by user behaviour





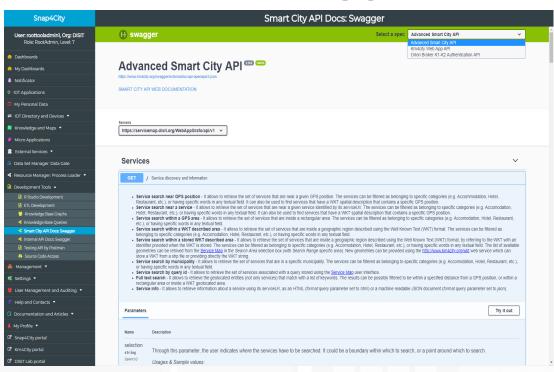




Advanced SmartCity API

- Search data: by text, near, along, etc.
 - Resolving text to GPS and formal city nodes model
- Empowering city users: contributions, suggestions, forum discussions, etc.
- Events: Entertainment, critical and mobility
- Public and Private Mobility & Transport, and predictions
- POIs, Cultural and Touristic info
- Health services and predictions
- Environmental information, heatmaps; values
- **Profiled Suggestions to City Users**
- Traffic flow reconstruction
- Personal Assistant: PAVAL
- User Engagement: goal experiences, and assessment
- Sharing knowledge among cities → see Knowledge base Management

Swagger



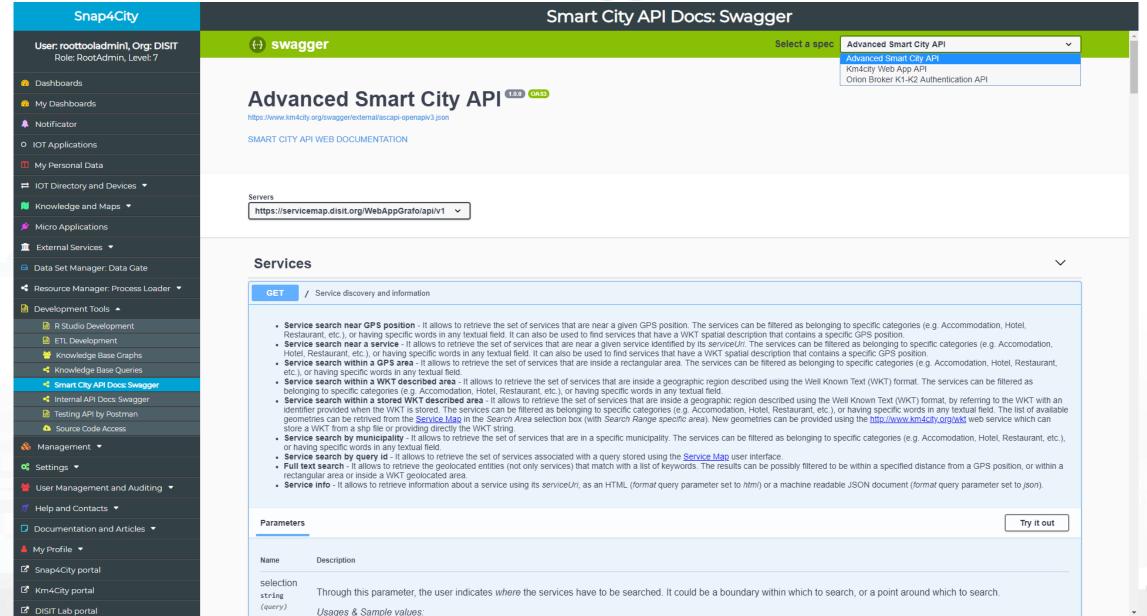














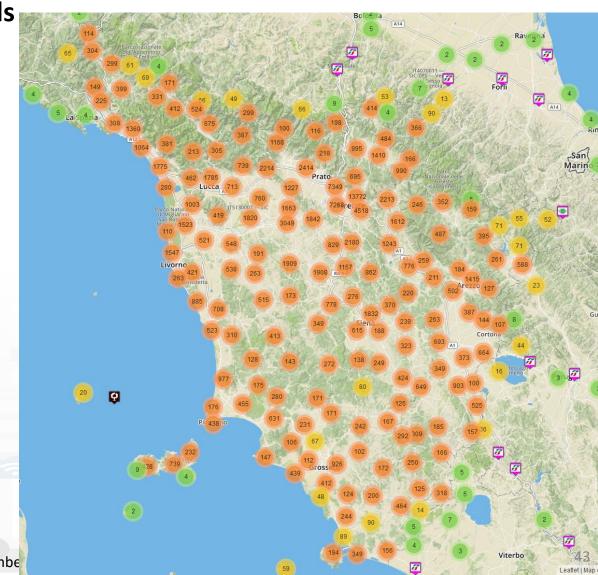






Thematic Data Domain Tuscany

- Street and geoinformation of the territory and details for routing, navigation, ...
- GeoResolution, Environmental data
- Mobility and Transport: public and private, public transport, parking status, fuel stations prices, traffic sensors, etc.
- **Culture and Tourism**: POI, churches, museum, schools, university, theatres, events in Florence
- **Environmental**: pollution real time, weather forecast, etc.
 - Environmental data geo resolution
- Social Media: twitter data
- **Health**: hospital, pharmacies, status of the first aid triage in major hospitals, ...
- Alarms: civil protection alerts, hot areas, ...







Access to Point of Interest information, POI

- POI: point of interest
- type: macro and subcategories
- Position: GPS, address, telephone, fax, email, URL, ...
- Description: textual, multilingual, with images, ...
- Link to dbPedia, Linked Open Data
- Links to other services
- Real time data if any: sensors data, timeline, events, prices, opening time, rules of access, status of services, status of queue, etc..
- See transversal services on ServiceMap
 - Regular and in test platform







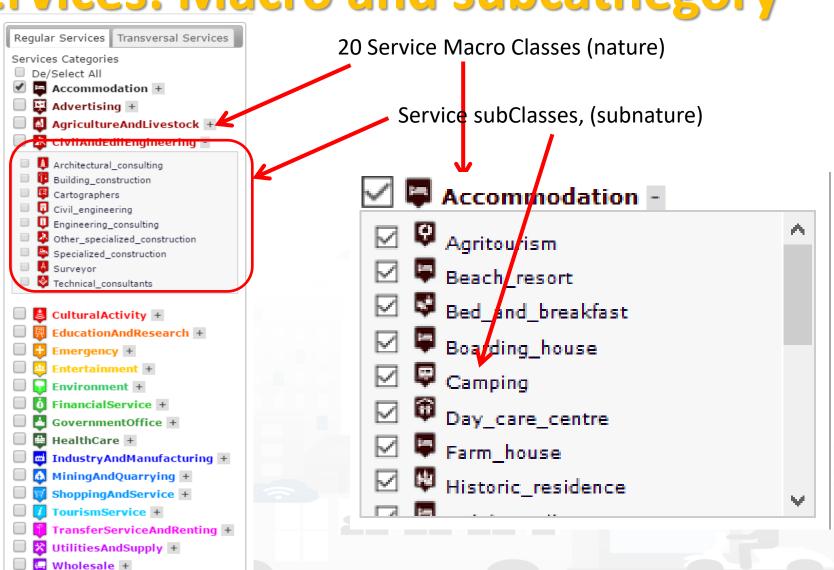




Concepts of Services: Macro and subcathegory

₩ineAndFood +

A SKOS area into the Km4Clty Ontology and Knowledge base for modeling POI and any element on map



mber 2023

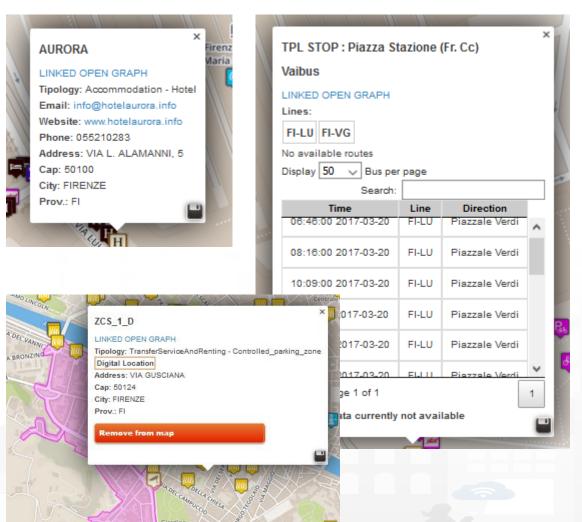


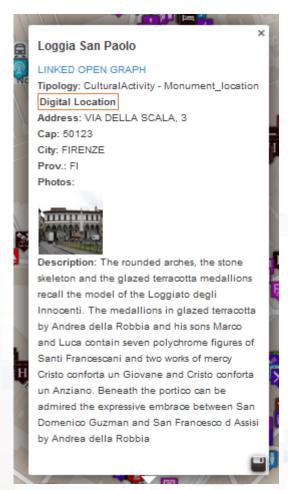


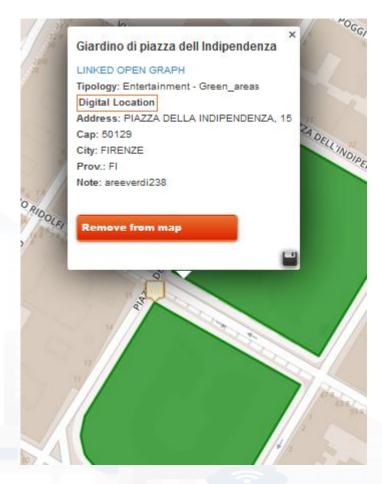




Service Information: different kinds of services















General Text Search Features

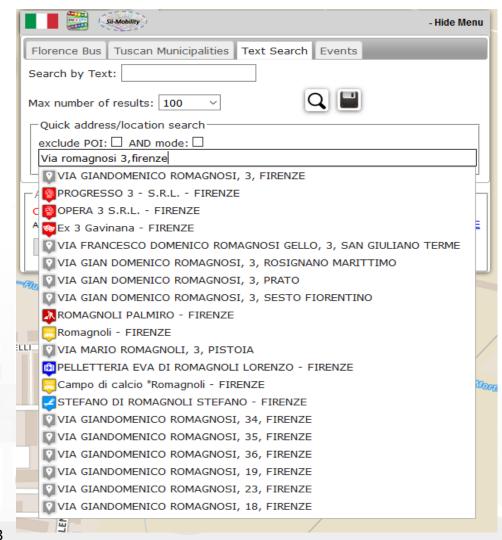
Search by text for POIs via:

- Full text: description, title, macro and category name
- Filtering by macro-cat and subcategory
- Filtering on distance and geometric shape

Search by text with assisted suggestion to get:

Streets and civic numbers, or POI, locations

Geo resolution, from point to street; from civic to GPS, etc.











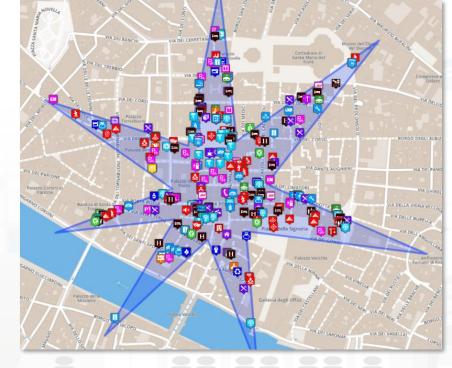
Around a point or POI



Search by Shape (WKT) or Distance

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

Inside an area



Along a polyline













Empowering City Users

- Allow city users to
 - provide comments, images and scores associated with a certain Service (or place, via GPS), discussions on forums, etc.
 - Get list of last contributions of the same kind provided by other users
 - Save favorites
 - Share trajectories,
 - Save and Manage their own data, IOT data, etc.
- Contributions can be:
 - used as feedbacks
 - moderated by a back-office personnel
 - **—** ...
- Connection with powerful servers based on 311 standard it also possible



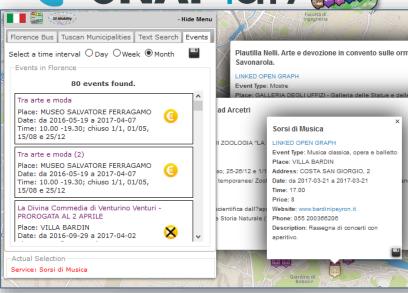




Access to Event information

- Getting Traffic Events: ESB, etc.
- Getting Critical Events: CAP standard
- Getting Police events
- Getting Entertainment Events in the city
 - Theater, museum, show, sport, etc.
 - Getting Event details
 - Event kind, and thus ordering
 - in the day, week, and month
 - Location, and thus ordering, or selecting events per area, per residence
 - General information
 - Opening and cost (if any)
 - Etc.











Maas

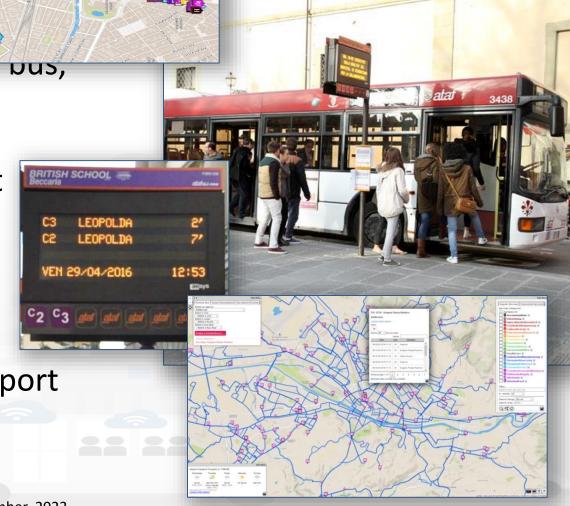




Supporting City Users in using Public Mobility

Public Transportation, PT

- Getting tickets
- Getting bus stops, lines, and time in bus, train and tramline (GTFS, ETL, ...)
- Getting Tunnel and Ferry Status
- Searching Services along a Pub. Transport line or closer to a stop
- Searching the closest bus stops
- searching for BUS stops via name
- real time delays of busses
- Modal/multimodal routing for Pub. Transport
- Tracking fleets, trajectories, etc.
- Get connected drive data

















Supporting City Users using Private Mobility

Private Transport

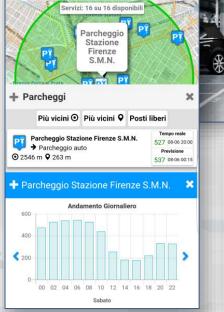
- Parking status (DATEX II, ...)
- Saving car park
- Getting closer parking
- OBD2 data from your engine or fleet
- Getting parking forecast: short and long term
- Getting closer free space on parking
- Getting fuel stations location and fuel product prices
- Getting bike sharing rack status
- Searching Services along a path or closer to a point or Service as Hotel, Restaurants, square, etc.
- Getting closer cycling paths
- Recharging stations: location and status
- Getting traffic information
- Heatmap where is safer to bike















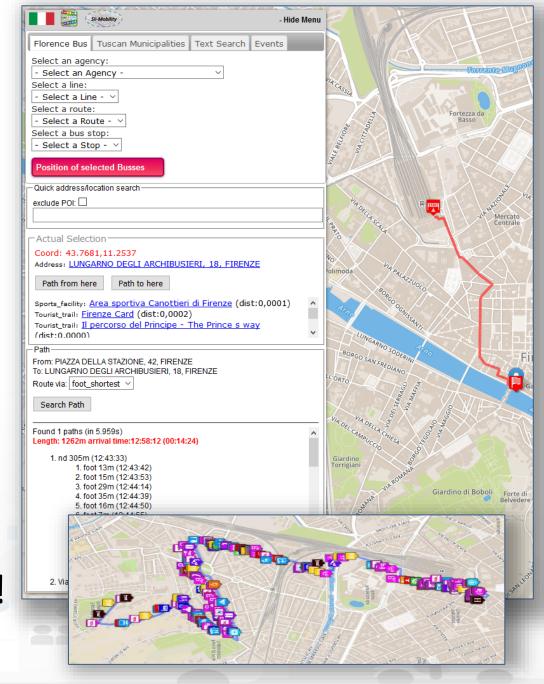
Private Mobility: routing and navigation paths

To get the path from two points/POIs:

- Shortest for pedestrian
- Quietest for pedestrian
- Shortest for private vehicles
- Multimodal with Public Transportation
- Constrained routing

Search for POIs along the identified Path!

http://www.disit.org/ServiceMap



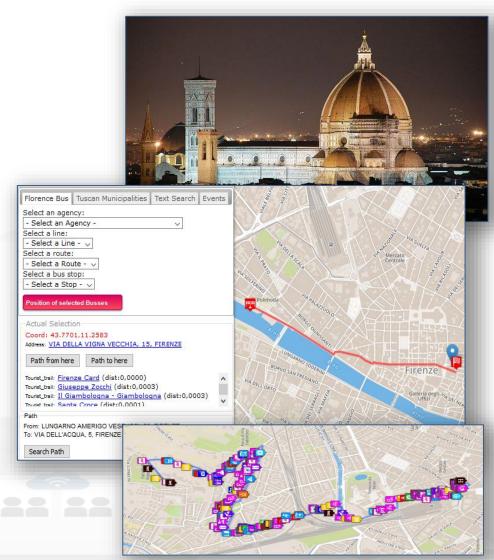






New Experience to access at Cultural and Touristic info

- Getting location and description of Point of Interests, POIs: culture and tourism first
 - Location, images, phone, URL, etc.
 - Get image, video, audio, ...
- Search for POIs in areas and closer
- Get routing to reach location or POI by walking downtown
 - searching Services along the path
- Search for location, full text assisted
- Leave a score, take a picture, etc.





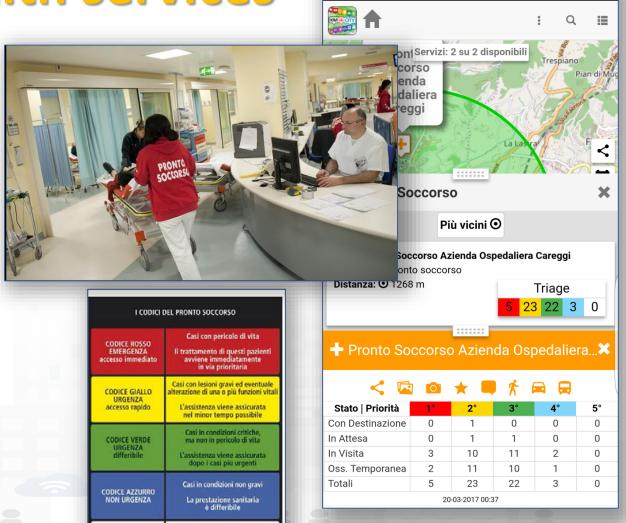






New way to access at health services

- Searching for pharmacies and hospitals
- Getting the closest hospital first aid locations and status
- Getting real time updated information about the first aid status of major hospitals (triage)



Casi con problematiche risolvibili dal medico curante, dalla guardia medica

o da ambulatori specialistici Tempi di attesa molto lungh

CODICE BIANCO





Codice Allerta Meteo

Access at Environmental information

Getting weather forecast for the next hours and days

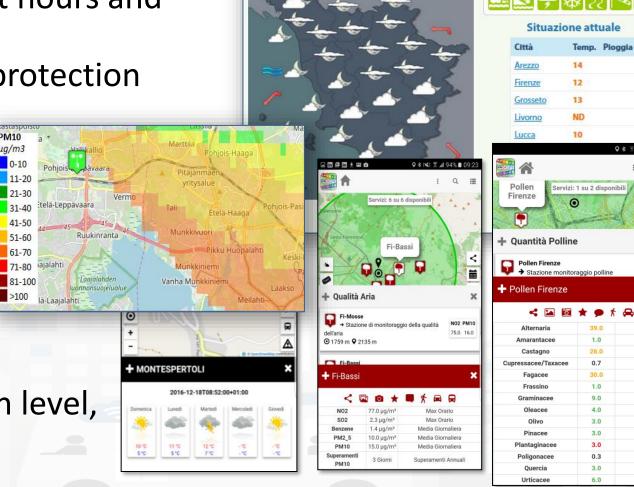
Getting alert information from Civil protection

Getting air quality status

 Getting Air quality via heatmaps, heatmap animation

- Computing Air quality indexes
- Computing Air quality predictions
- Getting pollination status
- getting actual weather status: temperature, humidity, pressure, rain level,

· etc.





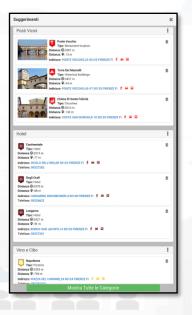


Profiled Suggestions to City Users

Personalized suggestions

- The server provide suggestions in the user context (location and time) arranged in a number of categories
 - Culture, mobility, food and drink, etc.
 - Alerts: civil protection, city council, twitter data, etc.
- The city user may reject some of them, thus the suggestion engine learns about preferred topics and category









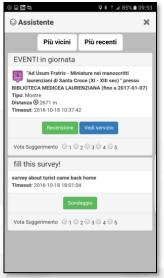


Profiled Engagements to City Users

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















Developing Web and Mobile Apps, MicroApps,...

Mobile Apps



Web App HTML5, MicroApplications



Embed into Web pages





Developer

Advanced Smart City API



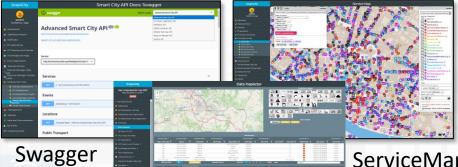
Mobile Application
Monitoring
Administrator



Knowledge Base,

Snap/Km4City Open Source development tool kit





ServiceMap

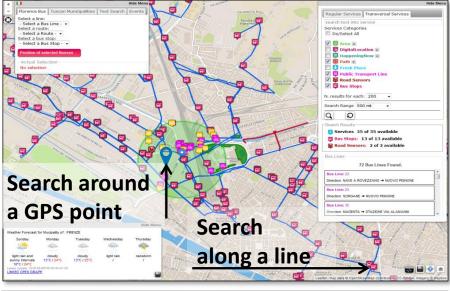
DataInspector



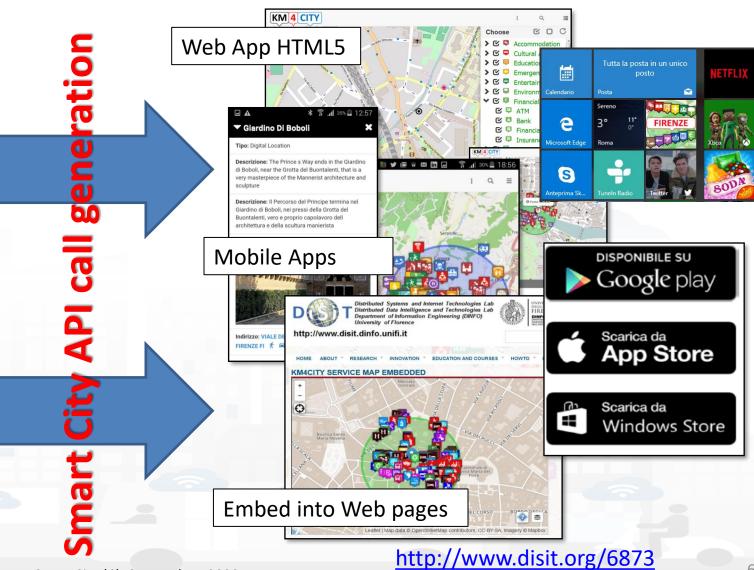




DINFO DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB SERVITA STUDI NATIONE DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB SERVITA STUDI DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB CSNAP4city KM/4 city SERVITA CITY KM/4 city AND INTERNET TECHNOLOGIES LAB (knowledge & Map tool)











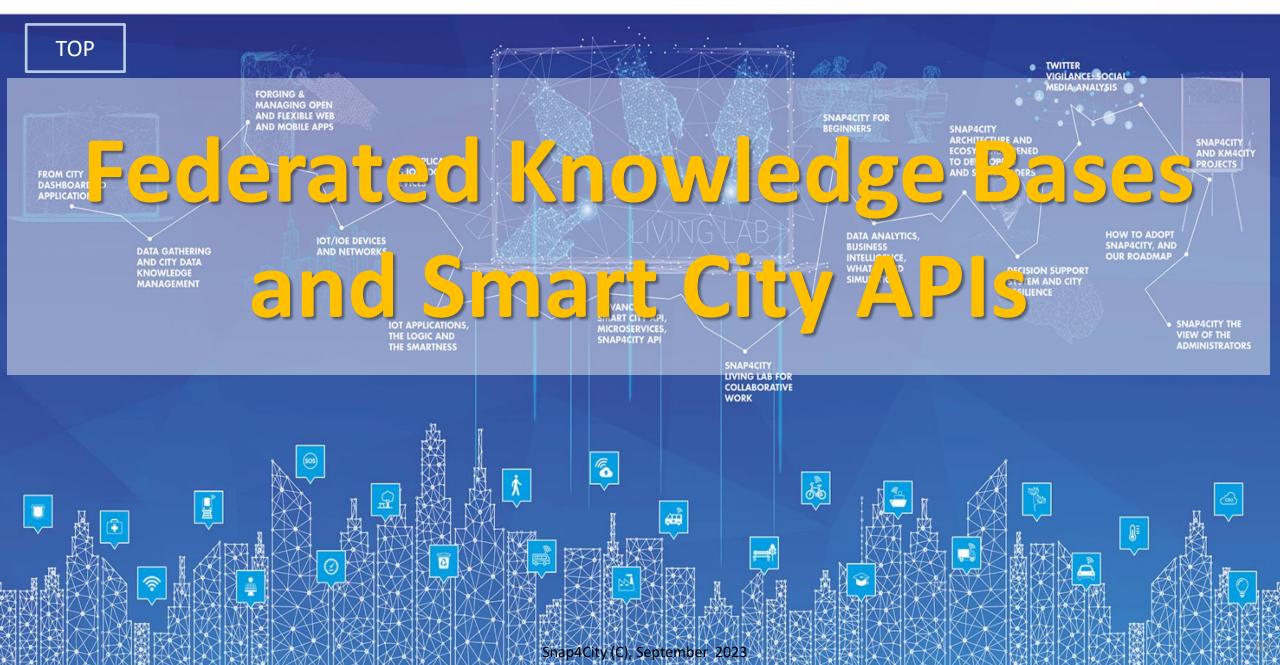
Smart City API



- based on Km4City engine on the back
- documented: https://www.snap4city.org/404
- ServiceMap tool to generate visually calls to exploit the Smart City API in web and mobile applications
- Documentation and examples:
 - 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
- The Alternatives:
 - just Dashboards directly exploiting data on graphics and/or
 - IOT Applications via Node-RED exploiting MicroServices also using the Smart City APIs

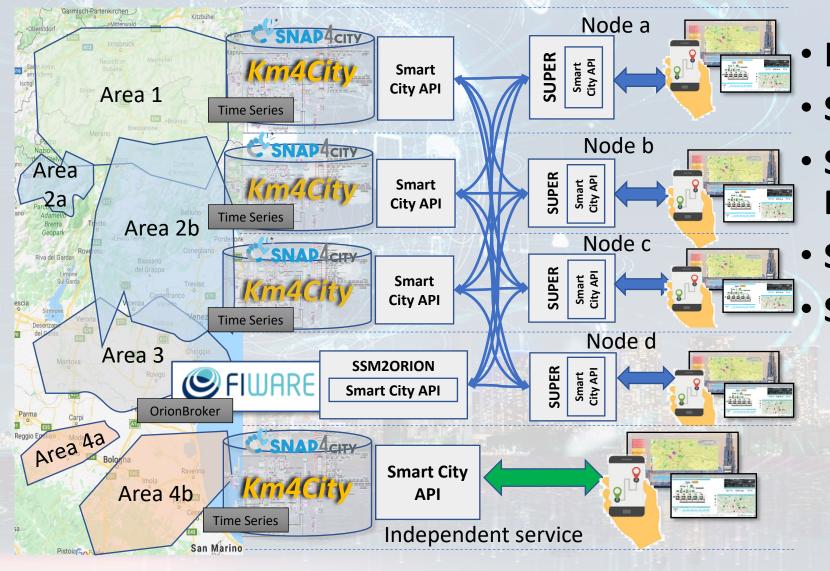
SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES





Federation of Smart City Services





- Km4City Semantic Reasoner
- ServiceMap interoperability
- Seamless for multiple Mobile Apps
- Smart City API
- Super:
 - distributed access and sharing services
 - Each city control its own data
 - Final user can pass from one city / area to another in seamless manner: without changing the mobile Apps

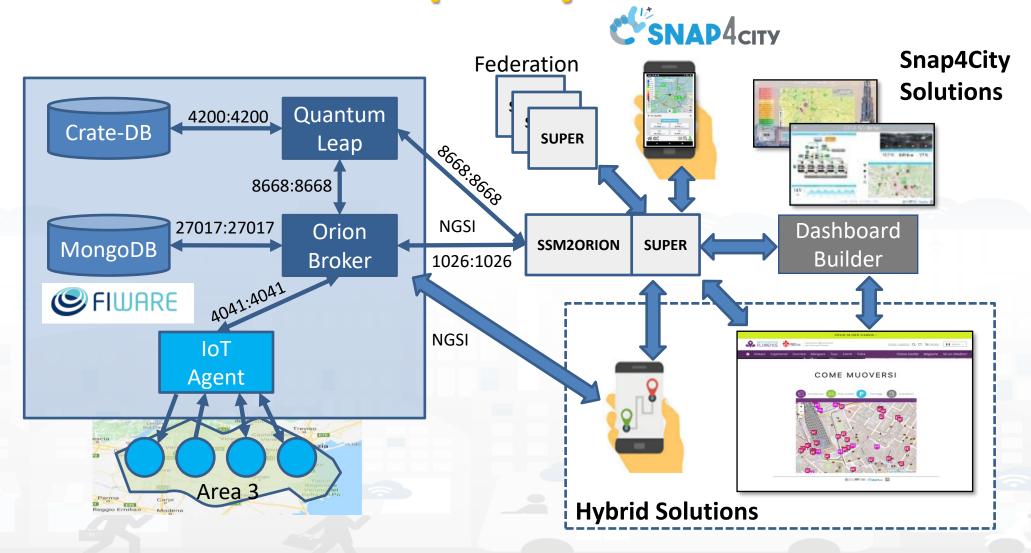








Federation of Snap4City vs ORION Broker











- 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

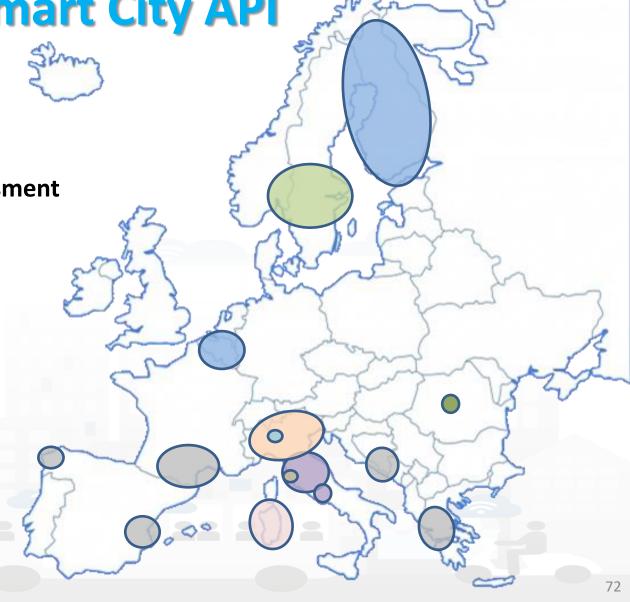
Multiple Knowledge base See performance assessment

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
- CKAN via harvesting



SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES













Development

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









Development Life-Cycle

https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle-v1-1.pdf

From Snap4City:

- We suggest you to read the TECHNICAL OVERVIEW:
 - https://www.snap4city.org/download/video/Snap4City-
- https://www.snap4city.org

- https://www.snap4industrv.org
- https://twitter.com/snap4city
- https://www.facebook.com/snap4city
- https://www.youtube.com/channel/UC3tAO09EbNba8f2-u4vandg

Coordinator: Paolo Nesi, Paolo.nesi@unifi.it

DISIT Lab, https://www.disit.org DINFO dept of University of Florence, Via S. Marta 3, 50139, Firenze, Italy Phone: +39-335-5668674



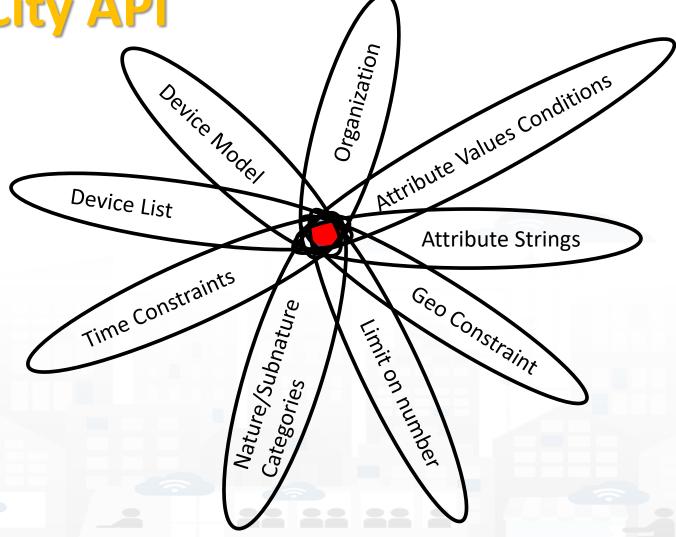






Selection on Smart City API

- Combining different filters for selecting entities from Smart City APIs
- Be care: filtering too much may lead to empty set ☺







How to Get the «Query» used in More Options (2a)

- REST CALL by category → JSON (Options in RED), they are REST ASCAPI calls
 - Requesting a category, so that to see all Services of the same category (subNature)
 - http://svealand.snap4city.org/ServiceMap/api/v1/?selection=59.581458578537955;16.71183586120606;59.62
 http://svealand.snap4city.org/ServiceMap/api/v1/?selection=59.581458578537955;16.71183586120606;59.62
 http://svealand.snap4city.org/ServiceMap/api/v1/?selection=59.581458578537955;16.71183586120606;59.62
 - Please note that in the MoreOption dashboard the GPS area is neglected
 - https://servicemap.disit.org/WebAppGrafo/api/v1/?selection=43.64471;11.005751;43.89471;11.505751&cate gories=Green areas&maxResults=200&format=json
 - Please note that in the MoreOption dashboard the GPS area is neglected
 - Custom PINS note: "selection" coordinates are used for collecting attributes in custom PINS. Other options such as "maxDists" cannot be used in custom PIN. All parameters can be used in other cases.
 - Different KB links are identified by their ASCAPI links: svealand.snap4city.org, servicemap.disit.org,
 - Requests to SuperServiceMap for the network of Federated KBs by using /api/....
 Without prefixed KB to obtain merged results from more KBs. For example as:
 - /api/v1/?categories=Air_quality_monitoring_station&format=json
 - Please note that the direct links to the superservicemap can be of the form:
 - https://www.disit.org/superservicemap/api/v1/?





How to Get the «Query» used in More Options (2b)

- REST CALL by ServiceURI → JSON (ServiceURI in RED), they are ASCAPI calls
 - Requesting single Service
 - https://servicemap.disit.org/WebAppGrafo/api/v1/?serviceUri=http://www.disit.org/km4city/resource/ARPAT _QA_FI-BOBOLI&format=json
 - https://servicemap.disit.org/WebAppGrafo/api/v1/?serviceUri=http://www.disit.org/km4city/resource/ARPAT
 QA FI-MOSSE SV&format=json
 - Different KBs links are identified by their ASCAPI links: svealand.snap4city.org, servicemap.disit.org,
 - Requesting all IoT Devices that have been produced by the same Model
 - https://www.disit.org/superservicemap/api/v1?selection=59.36535064975547;13.457822799682619;59.39031474260852;13.566999435424806&model=SmartLightCapelon&format=json
 - Please note that in this case the call is performed on the superservicemap, you can change to go directly on the right KB
 - You can specific both category and model to be more precise and focused.
 - https://www.disit.org/superservicemap/api/v1/?selection=36.8092847020594;12.216796875000002;42.71473218539458
 ;32.03613281250001&categories=Travel_information&format=json&fullCount=false&maxResults=500&model=DOMESTIC
 MOVEMENTS2013-2018 1620304406
 - In this case, we have a double filtering for model and for categories, plus other constraints
 - Please note that in the MoreOption dashboard the GPS area is neglected





How to Get the «Query» used in More Options (2c)

- Requesting get data single device (view on map, if format **HTML** and not JSON)
 - Request to see the single device:
 - https://svealand.snap4city.org/ServiceMap/api/v1/?serviceUri=http://www.disit .org/km4city/resource/iot/orionCAPELON-UNIFI/CAPELON/5C0272FFFE894AF7&format=json&fromTime=3-day
 - With ServerURI: http://www.disit.org/km4city/resource/iot/orionCAPELON-UNIFI/CAPELON/5C0272FFFE894AF7
 - From KB: https://svealand.snap4city.org







https://svealand.snap4city.org/ServiceMap/api/v1/?serviceUri=http://www.disit.org/km4city/resource/iot/orionCAPE LON-UNIFI/CAPELON/5C0272FFFE894AF7&format=json&fromTime=3-day

```
{ "Service":
{"type": "FeatureCollection",
"features": [
       "geometry": { "type": "Point", "coordinates": [ 13.46701, 59.37458 ] },
       "type": "Feature",
       "properties": { "serviceUri": "http://www.disit.org/km4city/resource/iot/orionCAPELON-UNIFI/CAPELON/5C0272FFFE894AF7",
         "serviceType": "Environment_Smart_street_light",
         "name": "5C0272FFFE894AF7",
         "typeLabel": "Smart street light",
         "protocol": "ngsi",
        "format": "ison",
         "model": "SmartLightCapelon2",
         "producer": "Capelon",
         "macaddress": "",
        "brokerName": "orionCAPELON-UNIFI",
         "ownership": "public",
         "organization": "CAPELON",
         "description": "",
         "website": "",
         "maintenanceUrl": "",
         "maxCapacity": "",
         "minCapacity": "",
         "isMobile": "",
         "nature": "Environment",
```





Query by value



Queries can be complex by geo-area, by cathegory, by IoT Device Model, a list of ServiceURI (all the same kind), with filters by value on specific Variables (numeric, and textual in AND), QUERY:

- https://www.snap4city.org/superservicemap/api/v1/iot-search/?selection=43.77;11.2&maxDists=700.2&model=CarPark
- https://www.snap4city.org/superservicemap/api/v1/iot-search/?selection=42.014990;10.217347;43.7768;11.2515&model=metrotrafficsensor&valueFilters=vehicleFlow>0.5;vehicleFlow
 300
- https://www.snap4city.org/superservicemap/api/v1/iot-search/?selection=43.77;11.2&maxDists=200.2&model=metrotrafficsensor&valueFilters=vehicleFlow>10;vehicleFlow<400&service Uri=http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO1;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO10;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO11;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO13;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO15;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO15;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO17;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO18;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO19;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO19;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO20;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO21;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO23;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO23;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO24;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO25;http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/METRO26





How to Get the «Query» used in More Options (3)

- ServiceMap (specific KB) and Query service
 - The Query performed is saved and can be recalled with a QueryID, valid for that specific KB, and not accessible via SuperServiceMap / Federated KB
 - The QueryID is communicated via email
 - Specific REST Call with HTML is also provided to change the Query in server associated with the QueryID received
- Query ID (only Read and Read/Write of the query)
 - https://servicemap.disit.org/WebAppGrafo/api/v1/?queryId=1c8111893d40a2bb0 7a2078ffe299ced&format=json
 - Cannot be used for Custom PINs.
 - Cannot be used to get data via ServiceMap since the Query ID is KB based





Special Commands in «Query» of More Options (4)

- Commands for Special Tool:
 - Traffic Flow tool: https://firenzetraffic.km4city.org/trafficRTDetails/roads/read.php
 - Scenario tool: /scenario/
 - Whatif tool: /whatif/
- Heatmaps, see Data Analytic part of the training for the several versions which can be used:
 - https://wmsserver.snap4city.org/geoserver/Snap4City/wms?service=WMS&layers=PM2 5Average24HourFlorence
 - https://wmsserver.snap4city.org/geoserver/Snap4City/wms?service=WMS&layers=denseNO2 Firenze IDW
 - WMSServer that is a GeoServer may be different for different installations of Snap4City









Time Series Data Access

- Time Series are attached to Devices which are identified by ServiceURI
- To Access at the Time Series (also called real time data) you can:
 - 1. From IoT App use the block «service info dev» In this case, you automatically access to your private and delegated data. You do not need to perform the authentication since it is performed directly from the microservice IoT App context, both on cloud and on edge
 - 2. From Python/Rstudio, Web and Mobile App, you can call Smart City API, see in this section and in Part 7 of the course.
 - 3. Retrieve data from IoT App and pass them to Python/Rstudio as presented in other sections. This approach is viable for small amount of data, such as some thousands. For larger amount of data or to be more efficient we suggest to use case (2) which is a direct access to the Smart City API.







IoT Search API (Search Entities)

- These API allows to find «IoT devices» matching a specific query on the dynamic data
 - On the last values
 - On a temporal range
- For example:
 - Find all weather sensors with a last temperature value greater than 35
 - Find all weather sensors that last week had a temperature greater than 38
- The API over a temporal range can return the list of matching devices or a list of dynamic data records matching the query
 - the list of times when temperature was greater than 38
- These APIs for performance reason query only the OpenSearch index and have some limitations





IOTSearch APIs



- The base url is https://www.snap4city.org/superservicemap/api/v1/iot-search/?...
- Query params are similar to other apis:
 - selection=...
 - A GPS point <latitude>;<longitude>
 - A GPS rectangular area <lat1>;<long1>;<lat2>;<long2>
 - A service uri (uses its lat;lon position)
 - More complex geographic filters are not supported
 - maxDists=..
 - Maximum distance in km from the GPS point (default 0.1 km)
 - categories=...
 - A list of categories as nature and subnature separated by ";"
 - model=...
 - Search for devices of a specific model (only one allowed)
 - serviceUri=<suri1>;<suri2>...
 - Filters on specific service uris
 - text=...
 - Filters using keywords and phrases to be searched in any string value of the device (phrases are delimited by "...")
 - valueFilters=<cond1>;<cond2>;...
 - A set of conditions in AND on specific values
 - <value name>:<string> (ex. Status:Active)
 - -- <value name>(=|<|<=|>>|<number> (ex. temperature>38)





IOTSearch APIs



Other query params:

- fromResult=...
 - Start from a result at a given position (starting from 0)
- maxResults=...
 - Maximum number of results returned (default 100)
- values=...
 - Report data of only specific fields, separated by «;» (e.g. temperature; humidity) if omitted all fields are reported
- sortOnValues=...
 - Allow to sort results on a specific field, if omitted are sorted by distance from the GPS point, adding «:asc» or «:desc» sort direction can be provided (e.g. temperature:asc)
- notHealthy=true
 - Reports only sensors that are now considered unhealthy as have not provided data in the expected next time
- For more details see the documentation:

https://www.km4city.org/swagger/external/#/IOT_Search/get_iot_search





IoTSearch Example

- Find all weather sensors with last values of temperature in [25,30] and humidity greater than 50
- GET https://www.snap4city.org/superservicemap/api/v1/iot-search/?
 categories=Weather_sensor&
 valueFilters=temperature>=25;temperature<=30;humidity>50
- Only GeoJSON response
- https://www.snap4city.org/superservicemap/api/v1/iotsearch/?categories=Weather sensor&valueFilters=temperature>=25;temperature<=3 0;humidity>50











```
snap4city.org/superservicemap/a X
                   snap4city.org/superservicemap/api/v1/iot-search/?categories=Weather_sensor&valueFilters=temperature...
                                                  Snap4 26 Calendar 26 Translate 🌨 Google Scholar Cita... D DISIT D DISIT
                            M Gmail

✓ Snap4City

{
    "features": [
             "geometry": {
                 "coordinates": [
                     10.5224,
                     43.794
                 "type": "Point"
            "properties": {
                 "date_time": "2023-07-27T08:30:00.000Z",
                 "deviceModel": "SirSensors",
                "deviceName": "SIRSensor TOS11000512",
                 "expected_next_date_time": "2023-07-27T08:45Z",
                "nature": "Environment",
                 "organization": "DISIT",
                "serviceUri": "http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/SIRSensor TOS11000512",
                 "subnature": "Weather sensor",
                 "values": {
                     "dateObserved": "2023-07-27T08:30:00.000Z",
                     "humidity": 57,
                     "rainDelta15": 0,
                     "temperature": 28.6,
                     "windDirection": 346,
                     "windGust": 3.4,
                     "windSpeed": 1.8
            "type": "Feature"
        },
             "geometry": {
                 "coordinates". [
```



DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

Results



```
"fullCount": 49.
"type": "FeatureCollection"
"features": [ {
        "type": "Feature"
        "geometry": { "type": "Point", "coordinates": [10.5224,43.794] },
        "properties": {
            "date time": "2023-07-26T07:00:00.000Z",
            "deviceModel": "SirSensors",
            "deviceName": "SIRSensor TOS11000512",
            "expected next date time": "2023-07-26T07:15Z",
            "nature": "Environment",
            "organization": "DISIT",
            "serviceUri": "http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/SIRSensor_TOS11000512",
            "subnature": "Weather sensor",
            "values": {
                "dateObserved": "2023-07-26T07:00:00.000Z",
                "humidity": 59,
                "rainDelta15": 0,
                "temperature": 27,
                "windDirection": 340,
                "windGust": 2.2,
                "windSpeed": 1.3
```





Results pagination

- Results can be paginated using
 - fromResult=... (first result to be returned starting with 0)
 - maxResults=... (page size)
- The total results is reported in the fullCount field of the results.

- To get page p (1..n) of results (with page size 50)
 - ...?...&fromResult=(p-1)*50&maxResults=50





IoTSearch over time Example

- Find all instants where a weather sensor has temperature in [25,30] and humidity greater than 50 in the last 7 days
- GET https://www.snap4city.org/superservicemap/api/v1/iot-search/time-range/?
 fromTime=7-day&
 categories=Weather_sensor&
 valueFilters=temperature>=25;temperature<=30;humidity>50
- Only GeoJSON response
- https://www.snap4city.org/superservicemap/api/v1/iot-search/timerange/?fromTime=7day&categories=Weather sensor&valueFilters=temperature%3E=25;temperature%3 C=30;humidity%3E50





Results



```
"fullCount": 122339,
"type": "FeatureCollection"
"features": [ {
        "type": "Feature",
        "geometry": { "coordinates": [ 9.23375, 45.47745 ], "type": "Point" },
        "properties": {
            "deviceModel": "Arduino Uno",
            "deviceName": "station01",
            "nature": "Environment",
            "organization": "DISIT",
            "serviceUri": "http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/station01",
            "subnature": "Weather sensor",
            "values": {
                "date time": "2023-07-26T08:44:56.791Z",
                "expected_next_date_time": "2023-07-26T08:45:06.791Z",
                "humidity": 86,
                "temperature": 30
```





IoTSearch over time Example

- Find all weather sensor with at least one time with temperature in [25,30] and humidity greater than 50 in the last 7 days
- GET https://www.snap4city.org/superservicemap/api/v1/iot-search/time-range/?

```
fromTime=7-day&
```

aggregate=true&

categories=Weather_sensor&

valueFilters=temperature>=25;temperature<=30;humidity>50

Only GeoJSON response





Results



```
"sumOtherDocs": 10771,
"type": "FeatureCollection"
"features": [
        "geometry": { "coordinates": [9.23375, 45.47745], "type": "Point"},
        "properties": {
            "aggregationCount": 44870,
            "deviceModel": "Arduino Uno",
            "deviceName": "station03",
            "nature": "Environment",
            "organization": "DISIT",
            "serviceUri": "http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/station03",
            "subnature": "Weather sensor"
        "type": "Feature"
```

- Returns the first 100 devices matching the request, «sumOtherDocs» is the number of matching records that are left out in the aggregation process, meaning that there are more than 100 device, adding maxResults=200 in the request the sumOtherDocs become 0 meaning that all matching data is aggregated in the results.
- No pagination is possible for aggregated results.







IoTSearch Limitations

- Report only limited static data of matching devices (model, nature, subnature, geographic position)
- Limited geographic queries (no polygon or line search)
- In case of private data, the owner and the delegated users can access the data but if the owner or the delegations change these applies only to newely submitted data (old data can be accessed only by old owner or old delegations)





Legenda on REST Call 1/2

- the **black continuous line** (push) will be used to send some data on the platform broker with a REST call which has to be Authenticated and Authorized according to the OpenId Connect as explained later, and would be in the form of:
 - https://<platformdomain>:8443/orionbrokerfilter/v1/updateContext
 - Or in the form for non TSL protected interaction:
 - http://iot-app.snap4city.org:80/orion-broker/v1/updateContext?elementid=ELEMENTID&k1=K1&k2=K2
- the **black dashed line** (pull) will be used to request some data from the platform by using a REST call to smart city API (Authenticated and Authorized according to the OpenId Connect as explained later), in the forms:
 - via regular Smart city API by category, etc.
 - http://svealand.snap4city.org/ServiceMap/api/v1/?selection=59.581458578537955;16.71183586120606;59.62875017053684;16.
 875171661376957&categories=Street light&maxResults=100&format=json
 - Via Super
 - https://www.disit.org/superservicemap/api/v1/?.....
 - Via Super by values
 - https://www.snap4city.org/superservicemap/api/v1/iot-search/?selection=43.77;11.2&maxDists=700.2&model=CarPark
 - https://www.snap4city.org/superservicemap/api/v1/iot-search/?selection=42.014990;10.217347;43.7768;11.2515&model=metrotrafficsensor&valueFilters=vehicleFlow>0.5;vehicleFlow<300





Legenda on REST Call 2/2

- the **red dashed line** (push) will be used to send some data from the platform (from an Orion broker) to some stable IP client or other machine for machine-to-machine communication
 - As a first step the client has to subscribe to some entity on the Orion Broker passing its IP where the broker will have to send the data in push
 - The POST will be in the form of <u>/v1/subscribeContext p</u>assing as parameters: elementid (the device ID, and K1, K2) or TSL approach
 - curl -X POST "https://broker1.snap4city.org:8080/v1/subscribeContext?elementid=mypersonaldatatester-device&k1=4e0924a8-fdd6-49cf-8d4a-f49cb5710d8b&k2=240567da-64a4-43b3-8ac9-1265178f3cbe" -H "accept:application/json" -H "Content-Type: application/json" -d "{\"entities\":[{\"type\":\"Ambiental\",\"isPattern\":false,\"id\":\"mypersonaldatatester-device\"}],\"attributes\":[\"temperature\"],\"reference\":\"http://prova/\",\"duration\":\"P1M\",\"notifyConditions\":[{\"type\":\"ONCHANGE\",\"condValues\":\"temperature\"}],\"throttling\":\"PT10S\"}"
 - Then the broker will send the messages to the subscribed client
 - it could be possible to have this kind of push also by using Kafka and/or WebSocket, but this is possible with simple and direct exposed API to all Snap4City platforms.
- The external APIs of Snap4City are documented in Swagger
 - https://www.km4city.org/swagger/external/index.html

SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES













TOP

Why and How to use Delegations to READ/WRITE







Delegations Rules

- Each entity in Snap4City is created as private and can be only visible/editable only for its owner, the so-called Creator.
 - The Owner/creator can pass the ownership to some other user of the same Organization.
 - Each entity in Snap4City provide a specific graphic user interface to change the ownership for: IoT Device Models, Dashboards, IoT Apps, etc.
- Once the Ownerships is passed, the former Owner/creator loses the possibility of editing and view the entity.
- The Owner of an IoT Device/Entity Instance can provide grant rights
- to other users.
 - READ_ACCESSS: means to be capable to read data messages of a device/Entity Instance even if you are not the Owner. This grant allows you to create devices/entities which can read from one or several users.
 - READ_WRITE: means to be capable to send new messages on that device/Entity, and also to read the
 data provided. This grant allows you to create devices/entities which can receive messages / data
 from one or several users.
 - MODIFY: means to have the right to modify the device/entity structure. This grant is quite strong and should be carefully used and in general the Owner should be conceptually the only one authorized to change the device / entity structure.





Example of Delegation Patterns

- A user A creates a device, and post data for it, and it is interested to communicate the data to many users at which the user A provide READ ACCESS.
 - To this end, A has to know the UserName of the platform to create the delegations.
- A user **A creates a device to receive messages notifications** from many users. A sort of mailbox for receiving some event notifications.
 - To this end, user A provide READ_WRITE grant to each of them. They are going to write their messages on the same mailbox, with the hope to avoid them to send messages at the same time stamp.
 - A can read the message notification and can overwrite them to confirm their reception.
- If all the IoT Devices/EntityInstances produce by a given Device Model are for instance userprofiles of some application, they can be searched and listed by all users having at least the READ ACCESS to those devices.
 - The platform provides a Search block in IoT App / Proc.Logic, as well as Smart City API as query by model.
 - In both cases, the user performing the query will receive back only the device he/she created and those that have been delegated to him.









TOP

Develop: Provide Data and Access to Protected Data





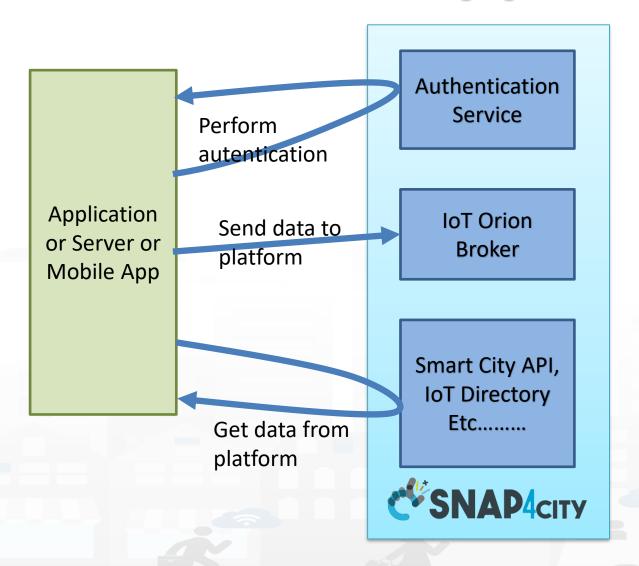








Approaches



• First

authenticate

Second

use the services to provide and/or get data











KEYCLOAK

Open Source Identity and Access
Management
Add authentication to applications and secure services with minimum effort.

https://www.keycloak.org/



OpenID®







Two Possible Approaches for Authentication

Authentication Code Flow Protocol (confidential application)

 For Web Application with Server Side functionalities or native applications, including services towards mobile applications

Single Page App

 For Web Application without Server Side functionalities

Implicit Flow Protocol referred to as Direct Grant with username/password (public applications)

- Less secure: It's not recommended to use this flow unless you absolutely need to
- For Front-End Web Application that do not have Server Side functionalities.
- JavaScript can do only this kind of applications









Authentication Flow Protocol

(confidential application, Web Server Application)

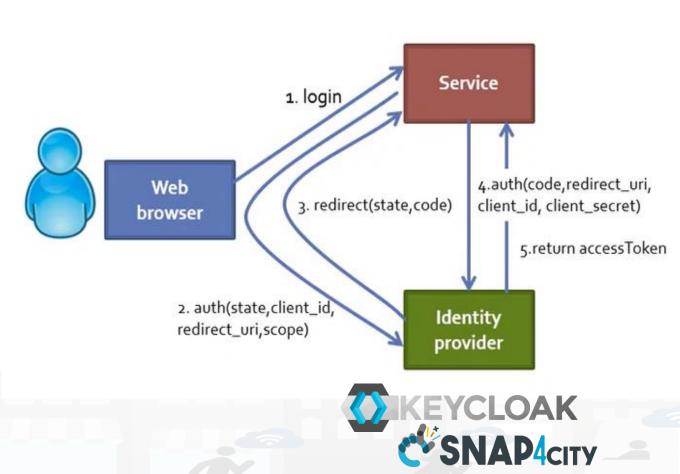
Step1 The Web Server Application provides a way to securely store information, and provide service to your users via HTML pages

- In particular: client_id and client_secret are secured on the WSA
 - They have to be requested to the snap4City platform organization
 - Snap4City has to know the redirect uri of your Application Server to complete the round and provide back the information
- This approach is valid for application servers which provide html pages to your users, while this information is saved into the Applications Server which also interact with the Snap4City Platform
 - It in practice the same approch used by the Dashboard manager to provice access to the dashboard at the users.





Authentication Flow Protocol, step 2



Step2

- the user login is redirected to the identity provider
- 2) Given the client_id of the application and the client_secret (in the diagram called code)
- 5) The Service get the AccessToken
- Then the Service can pose any API rest call to get data for the User







- The access token is in the format of JSON Web Tokens
- *

- https://jwt.io/
- The access token was:
- eyJhbGciOiJSUzI1NilsInR5cClgOiAiSIdUliwia2lkliA6ICJOZVBpSFRvREtibWZzbl9hREtETGpGTHFKQXluTXNNWjZjS1IMeGRoS29zIn0.eyJqdGkiOilyZGQ xYmVkZC1jODlmLTRjY2QtODM3MS1mN2Y2OWY5OTU2YjliLCJleHAiOjE2NzE1NTMxMjgslm5iZil6MCwiaWF0ljoxNjcxNTUxNjl4LCJpc3MiOiJodHRw czovL3d3dy5zbmFwNGNpdHkub3JnL2F1dGgvcmVhbG1zL21hc3RlcilsImF1ZCl6Imp1cHl0ZXJodWltcG9udGR1Z2FyZClsInN1Yil6ImQzZmMyNml3LW Q1MTktNGJmYy04OGExLWU1OWMwNDRmNjcxNClsInR5cCl6lkJlYXJlcilsImF6cCl6Imp1cHl0ZXJodWltcG9udGR1Z2FyZClsImF1dGhfdGltZSl6MCwic 2Vzc2lvbl9zdGF0ZSl6ljl0ODJiZTNiLTBkYTUtNDNkZS04MzMwLTBiMzJmNjQ0ZmlyZSlsImFjcil6ljEiLCJhbGxvd2VkLW9yaWdpbnMiOltdLCJyZWFsbV9h Y2Nlc3MiOnsicm9sZXMiOlsiQXJlYU1hbmFnZXliLCJ1bWFfYXV0aG9yaXphdGlvbiJdfSwicmVzb3VyY2VfYWNjZXNzljp7ImFjY291bnQiOnsicm9sZXMiO lsibWFuYWdlLWFjY291bnQiLCJtYW5hZ2UtYWNjb3VudC1saW5rcylsInZpZXctcHJvZmlsZSJdfX0sInJvbGVzljpblkFyZWFNYW5hZ2VyliwidW1hX2F1d Ghvcml6YXRpb24iLCJvZmZsaW5lX2FjY2VzcyJdLCJuYW1lljoic3VybmFtZSlsInByZWZlcnJlZF91c2VybmFtZSl6InRlc3R3YXN0ZSlsImZhbWlseV9uYW1llj oic3VybmFtZSlsImVtYWlsljoidGVzdHdhc3RlQGdtYWlsLmNvbSJ9.RE7whLSrXRpf3uXFV32rVb90YHY4GW0g087OS_k-p79Q84twdQswu-8OaAT0bV1RKep0qpRKZpWAsBWcHwrWEeDDNadUbv6n-GmUT0qfZRTpRzn2N8JfpqHGal2sC4-ThstKxgH99fkl6e_9ubZOz4G9zWHQRrlHTcEmReYfazOnutdmgSX0F-ErM8eO9vRPmUmWBn5y7ZUm8re7CH6UPZNb15P4dXUnfR_zZK3gw8tyUyXdkHSSYTZrtj3fFbVjq3zxzV1Do2al-BpqS7quiCyHMG0qlnYriWZKSARUyzjulL1QqBoSS6 xTe25wyizvwZ1BwHoeak40oRc0lqAgw











Encoded PASTE A TOKEN HERE

"eyJhbGci0iJSUzI1NiIsInR5cCIg0iAiSldUIi wia2lkIiA6ICJ0ZVBpSFRvREtibWZzb19hREtET GpGTHFKQX1uTXNNWjZjS11MeGRoS29zIn0.eyJq dGkiOiIyZGQxYmVkZC1jODlmLTRjY2QtODM3MS1 mN2Y2OWY5OTU2YjIiLCJleHAiOjE2NzE1NTMxMj gsIm5iZiI6MCwiaWF0IjoxNjcxNTUxNjI4LCJpc 3MiOiJodHRwczovL3d3dy5zbmFwNGNpdHkub3Jn L2F1dGgvcmVhbG1zL21hc3RlciIsImF1ZCI6Imp 1cH10ZXJodWItcG9udGR1Z2FyZCIsInN1YiI6Im QzZmMyNmI3LWQ1MTktNGJmYy040GExLWU10WMwN DRmNjcxNCIsInR5cCI6IkJ1YXJ1ciIsImF6cCI6 Imp1cHl0ZXJodWItcG9udGR1Z2FyZCIsImF1dGh fdGltZSI6MCwic2Vzc2lvbl9zdGF0ZSI6IjI00D JiZTNiLTBkYTUtNDNkZS04MzMwLTBiMzJmNjQ0Z mIyZSIsImFjciI6IjEiLCJhbGxvd2VkLW9yaWdp bnMiOltdLCJyZWFsbV9hY2Nlc3MiOnsicm9sZXM iOlsiQXJlYU1hbmFnZXIiLCJ1bWFfYXV0aG9yaX phdGlvbiJdfSwicmVzb3VyY2VfYWNjZXNzIjp7I mFjY291bnQiOnsicm9sZXMiOlsibWFuYWdlLWFj Y291bnQiLCJtYW5hZ2UtYWNjb3VudC1saW5rcyI sInZpZXctcHJvZmlsZSJdfX0sInJvbGVzIipbIk

Decoded EDIT THE PAYLOAD AND SECRET

```
HEADER: ALGORITHM & TOKEN TYPE
    "alg": "RS256",
    "typ": "JWT",
    "kid": "NePiHToDKbmfsn_aDKDLjFLqJAynMsMZ6cKYLxdhKos"
PAYLOAD: DATA
    "jti": "2dd1bedd-c89f-4ccd-8371-f7f69f9956b2",
    "exp": 1671553128,
    "nbf": 0,
    "iat": 1671551628,
    "iss":
  'https://www.snap4city.org/auth/realms/master",
    "aud": "jupyterhub-pontdugard",
    "sub": "d3fc26b7-d519-4bfc-88a1-e59c044f6714",
    "typ": "Bearer",
    "azp": "jupyterhub-pontdugard",
    "auth_time": 0,
    "session_state": "2482be3b-0da5-43de-8330-
 0b32f644fb2e",
    "acr": "1",
    "allowed-origins": [],
    "realm_access": {
      "roles": [
        "AreaManager",
```



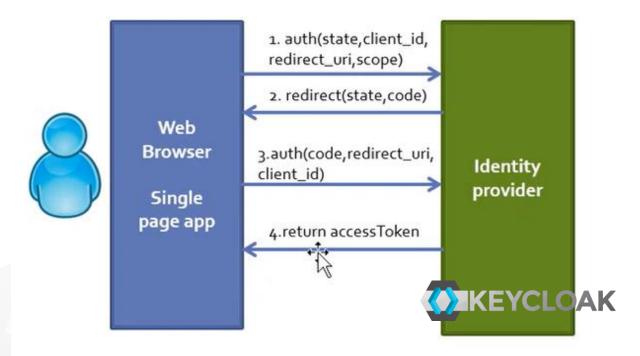






Single Page App

- The Single Page has no secure way to store information on client side
- All the secret information is maintained on the Identity Provider side



- The Single Page has to bring the login on the Identify Provider, which redirect on the applications
- The process follow the above presented approach
- Given the client_id of the application, the users can get the accessToken to make requests.

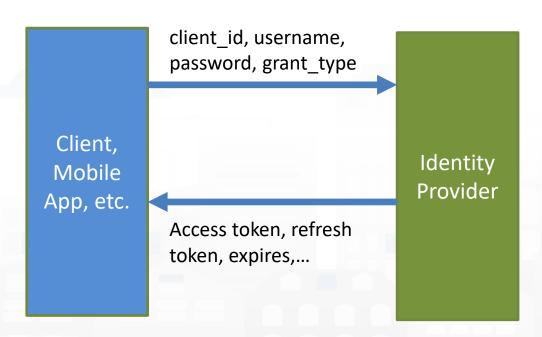




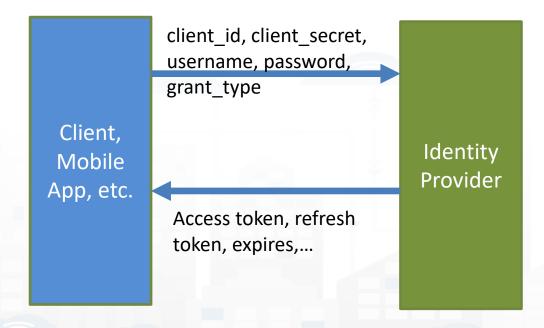


Two Simpler Cases

Public Applications



Implicit Flow Protocol





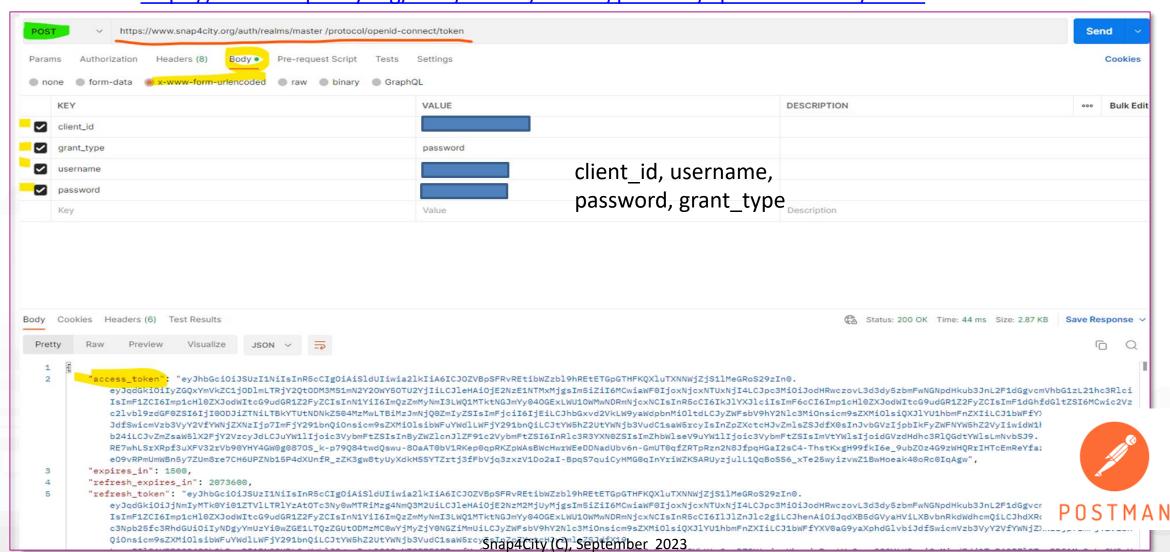






Public Applications

https://www.snap4city.org/auth/realms/master/protocol/openid-connect/token



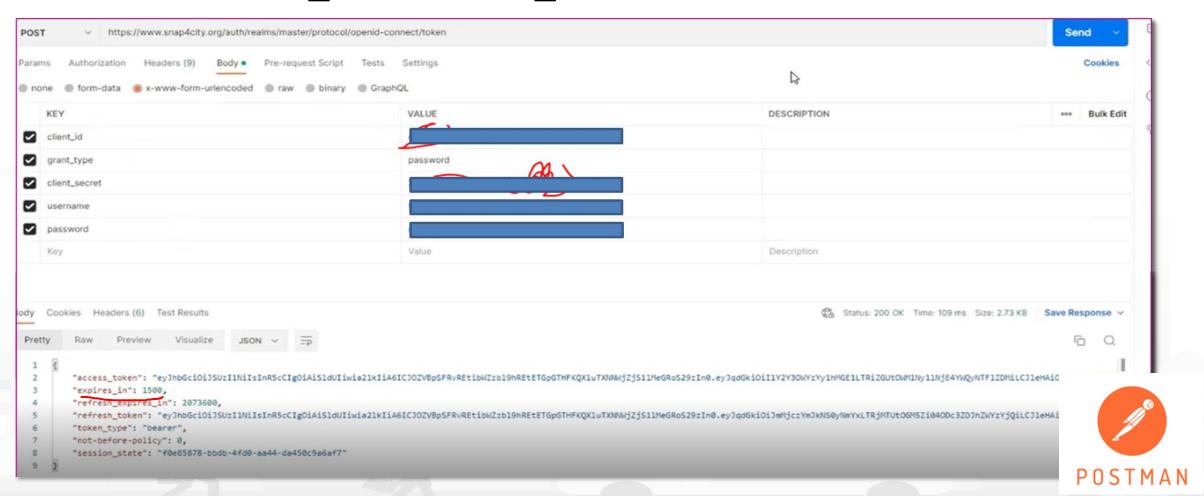






Implicit Flow Protocol

For some client_id the client_secret are needed







Access Token & Refresh Token

- Access tokens have typically of short duration
 - Once the access token is expired,
 - The refresh token can be used to request another fresh access token and this can be done at the endpoint
- https://www.snap4city.org/auth/realms/master/protocol/openidconnect/token/
- With the parameters reported in the next slide



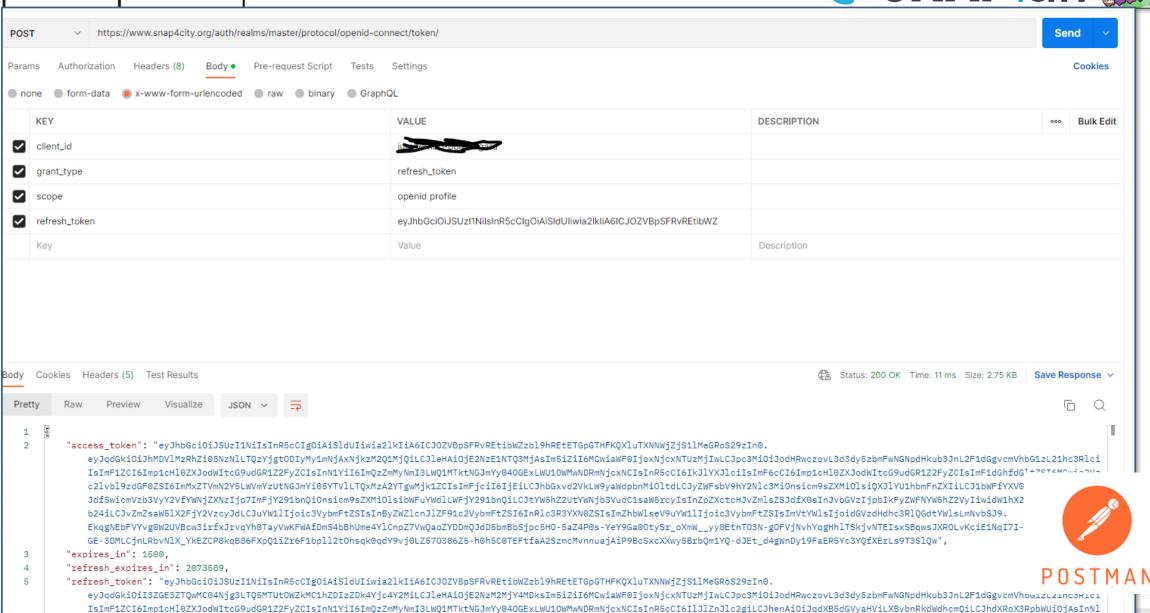
UNIVERSITÀ DEGLI STUDI FIRENZE

DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE



Officeries OFTWOOD of bull-world | Lucivand bearing to the Part of the Part of





c3Npb25fc3RhdGUi0iJjMWU1ZjdmOS1lZmM1LTRiZmIt0WE1ZS00MTMwNmE4MDI5NWQiLCJyZWFsbV9hY2Nlc3MiOnsicm9sZXMi0lsiQXJlYU1hbmFnZXIiLCJ1bWFfYXV0aG9yaXphdGlvbiJdfSwicmVzb3VyY2VfYWNjZXNzIjp7ImFjY291bn









TOP

Develop: How Cloud Containers may Access to Protected Data (example of Python)







Private Device Data Retrieval

- We'll use the cloud installation of jupyterhub
- https://www.snap4city.org/650



Not All The Device in Snap4City are public...

for some you'll need an access token to the private **IoT Device** of that authenticated user ¶ so let's get the username and password

```
### in the config.py file that i've created are stored the user and password for the snap4city authentication
# snap4cityauth = dict(
# user = 'user name of snap4city',
# psw = 'the password of the user',
# clid= '<client id depending on the App kind>' has to be obtained from Snap4City organization by sending an email to snap4city@disit.org.
# )
import config
utente = config.snap4cityauth['user']
password = config.snap4cityauth['psw']
client_id = config.snap4cityauth['clid']
```







Private Device Data Retrieval

next let's get the auth token 1

```
2]: import requests
     import json
     url = "https://www.snap4city.org/auth/realms/master/protocol/openid-connect/token/"
     data = {"client id": client id, "grant type": "password", "username": utente, "password": password}
     r=requests.post(url, data)
     print(r.status code, r.reason)
     responseToken=json.loads(r.text)
     refreshToken = responseToken["refresh token"]
     print("access_token : {}... expires in {}s, token_type: {}".format(responseToken['access_token'][:20],responseToken['expires_in'],responseToken['token_type'] ))
     #to update the token using the refresh token
     url = "https://www.snap4city.org/auth/realms/master/protocol/openid-connect/token/"
     data = {"client id": client id, "grant type": "refresh token", "scope": "openid profile", "refresh token": refreshToken}
     r=requests.post(url, data)
     print("updating token using the refresh token ",r.status_code, r.reason)
     responseToken=json.loads(r.text)
     200 OK
     access token : eyJhbGciOiJSUzI1NiIs... expires in 1500s, token type: bearer
     updating token using the refresh token 200 OK
```







Private Device Data Retrieval

so now you can access the private iot device data...

```
auth_token=responseToken['access_token']
hed = {'Authorization': 'Bearer ' + auth_token}

url = "https://www.snap4city.org/superservicemap/api/v1?serviceUri=http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/118907.682_485819.390-Plastic&accessToke

response = requests.get(url, headers=hed)
if response.status_code == 200: # ok
    print(json.loads(response.text))

{'Service': {'features': [{'geometry': {'coordinates': [4.857379, 52.359085], 'type': 'Point'}, 'properties': {'address': '', 'avgStars': 0, 'brokerName': 'orionUNIFI', 'cap': '', 'city': '', 'civic': '', 'comments': [], 'description': 'Plastic', 'email': '', 'fax': '', 'format': 'json', 'frequencySec': '600', 'isMobile': '', 'li
```

SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES





Basic version for MaaS mobile App: missing payments: a small version is on GITHUB, a wider version will be Published soon













Developing Web and Mobile Apps, MicroApps,...

Mobile Apps



Web App HTML5, MicroApplications



Embed into Web pages





Developer

Advanced Smart City API



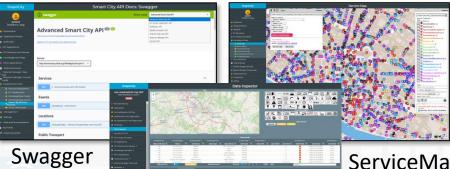
Mobile Application Monitoring Administrator



Knowledge Base,







ServiceMap







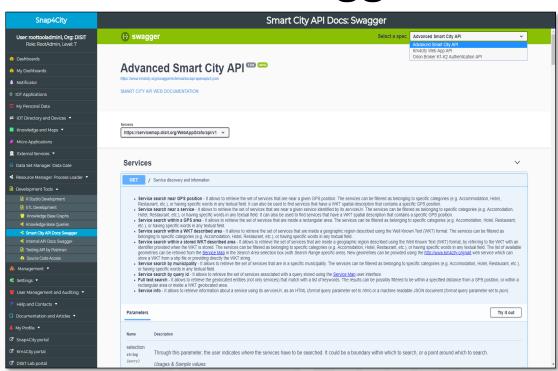




Advanced SmartCity API

- Search data: by text, near, along, etc.
 - Resolving text to GPS and formal city nodes model
- Empowering city users: contributions, suggestions, forum discussions, etc.
- Events: Entertainment, critical and mobility
- Public and Private Mobility & Transport, and predictions
- POIs, Cultural and Touristic info
- Health services and predictions
- Environmental information, heatmaps; values
- **Profiled Suggestions to City Users**
- Traffic flow reconstruction
- Personal Assistant: PAVAL
- User Engagement: goal experiences, and assessment
- Sharing knowledge among cities → see Knowledge base Management

Swagger









DINFO DIPARTIMENTO DI DISTRIBUTE SYSTEMS AND DISTRIBUTE SYSTEMS AND

- Firenze Dove Cosa, Km4City:
 - Android: https://play.google.com/store/apps/details?id=org.disit.siiMobile&hl=en&gl=US
 - Apple iOS: https://apps.apple.com/it/app/firenze-dove-cosa-km4city/id1028356115
- Toscana dove cosa, Km4City
 - Android: https://play.google.com/store/apps/details?id=org.disit.toscana&hl=en&gl=US
 - Apple iOS: https://apps.apple.com/it/app/toscana-dove-cosa-km4city/id1064554200
- **Antwerp in a Snap**
 - Android: https://play.google.com/store/apps/details?id=org.disit.snap4city.mobileApp.antwerp&hl=en&gl=US
 - Apple iOS: https://apps.apple.com/it/app/antwerp-in-a-snap/id1467737363
- Helsinki in a Snap
 - Android: https://play.google.com/store/apps/details?id=org.disit.snap4city.mobileApp.helsinki&hl=en&gl=US
 - Apple iOS: https://apps.apple.com/it/app/helsinki-in-a-snap/id1466970280
- **Tuscany in a Snap**
 - Android: https://play.google.com/store/apps/details?id=org.disit.snap4city.mobileApp.tuscany&hl=en&gl=US
 - Apple iOS: https://apps.apple.com/us/app/toscana-in-a-snap/id1471094480
- **Snap4Pisa: Pisa in a Snap:**
 - Android: https://snap4city.aedit.it/drupal//sites/default/files/snap.apk
- **Weee Life Mobile App:**
 - Android: https://play.google.com/store/apps/details?id=org.disit.lifeweee&hl=en&gl=US
 - Apple iOS: https://apps.apple.com/it/app/life-weee/id1470224854
- **Comune Facile Android:**
 - Android: https://www.e015.regione.lombardia.it/site/app-detail?id=131
 - Comune Facile IOS: https://www.e015.regione.lombardia.it/site/app-detail?id=130
- POS by EPSON: https://www.e015.regione.lombardia.it/site/app-detail?id=58
- SnapBot: https://www.snap4city.org/684





















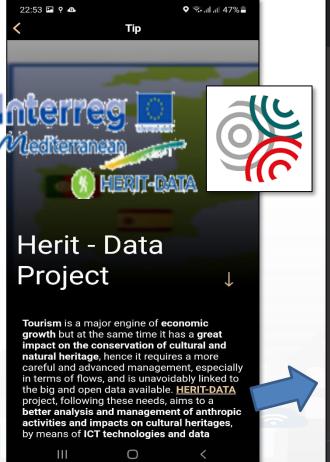


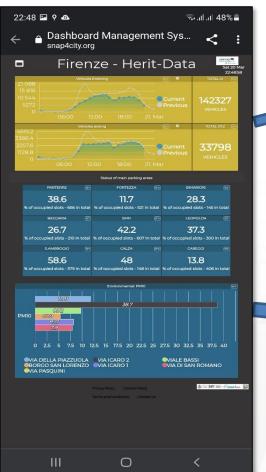
22:49 🖾 🕈 👁



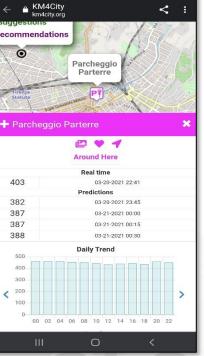
Florence Dashboard Smart Shoard Smart Smar

https://www.snap4city.org/dashboardSmartC ity/view/index.php?iddasboard=MzAwNA==

















TOP

Web and Mobile App with Open Development Kit





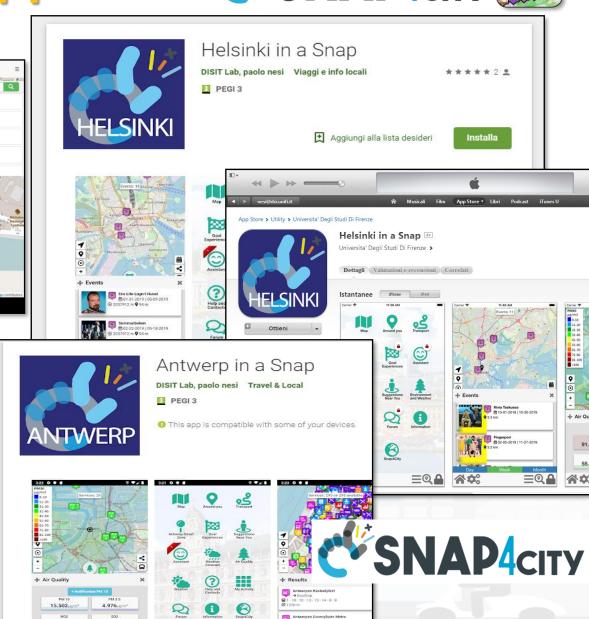
UNIVERSITÀ **DEGLI STUDI** FIRENZE

Mobile Apps















8 0°

≡@ ♣

A 00

Km4City APP, features

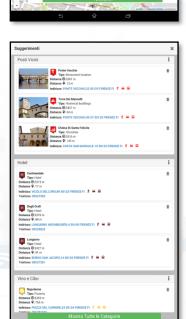


- **5 languages:** IT, EN, SP, DE, FR
- **Profiles** city users: citizens, commuter, student, tourist, operator, etc.
- Profiled Menu per POI
 - adaptive
- Main Menu: dynamic, and personalized
- Search Text
- Search per POI
 - Near to you, near to a point, a line, ...
- Other search
 - Close to you, events green areas, public transport, tickets, Cycling, parking, ...
 - Etc.
- POI
 - Preferred, Social icon
 - Ranking, Comments, Images









➤ □ ■ Accommodation

Cultural Activity
 □ Education And Research

➤ □ ■ Agriculture And Livestock

> 🖸 💆 Civil And Edil Engineering

➤ □ ■ Advertising

➤ □ □ Emergency
➤ □ □ Entertainment

➤ □ ■ Environment
➤ □ ■ Financial Service

> O P Health Care

➤ □ □ Government Office

➤ □ □ Mining And Quarrying➤ □ □ Shopping And Service

D □ Tourism Service
 D □ Transfer Service And Renting
 D □ Utilities And Supply

> D 📮 Industry And Manufacturing









Km4City APP



- **Smart Parking, in Tuscany**
- **Smart First Aid in Tuscany**
- **Smart Public Transportation** in Tuscany
- **Smart Fuel pricing in Tuscany**
- **Bike Sharing in Pisa**
- Weather condition in Tuscany
- **Environmental data**
- Pollution and Pollination in **Tuscany**
- **Traffic Sensors in Tuscany**
- **Smart Routing in Tuscany**
- **Smart Transportation in Florence**
 - Events, traffic, ...
- **Entertainment Events in Florence**

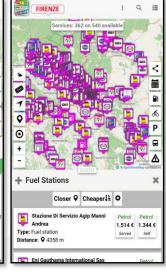


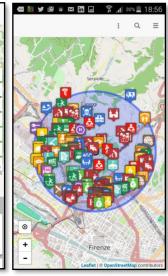












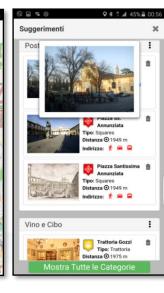
Search Stops By Agency

⊘CAP

BluBus

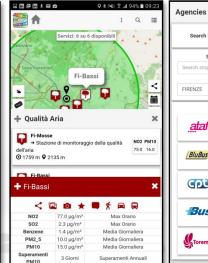
CPU

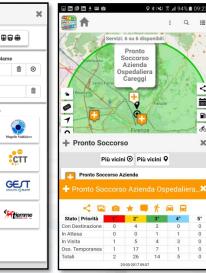
Bus











Km4City APP, features 3/3 CSNAP4city

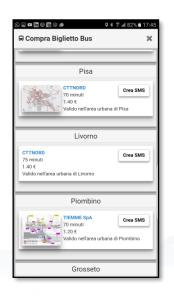




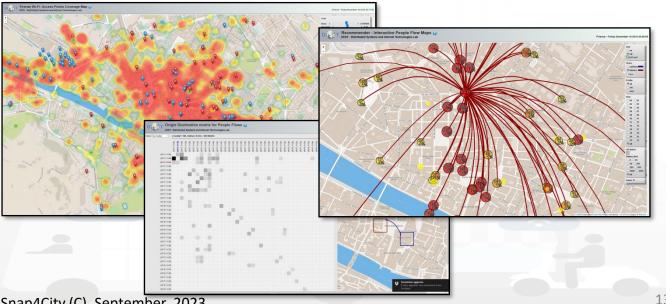
- **Navigation 3D**
- Ticketing for busses
- App used are tool for city assessment
 - Wi-Fi status
 - iBeacon status
 - User behavior analysis
 - GPS movements kinds
 - OD matrix
 - International flows

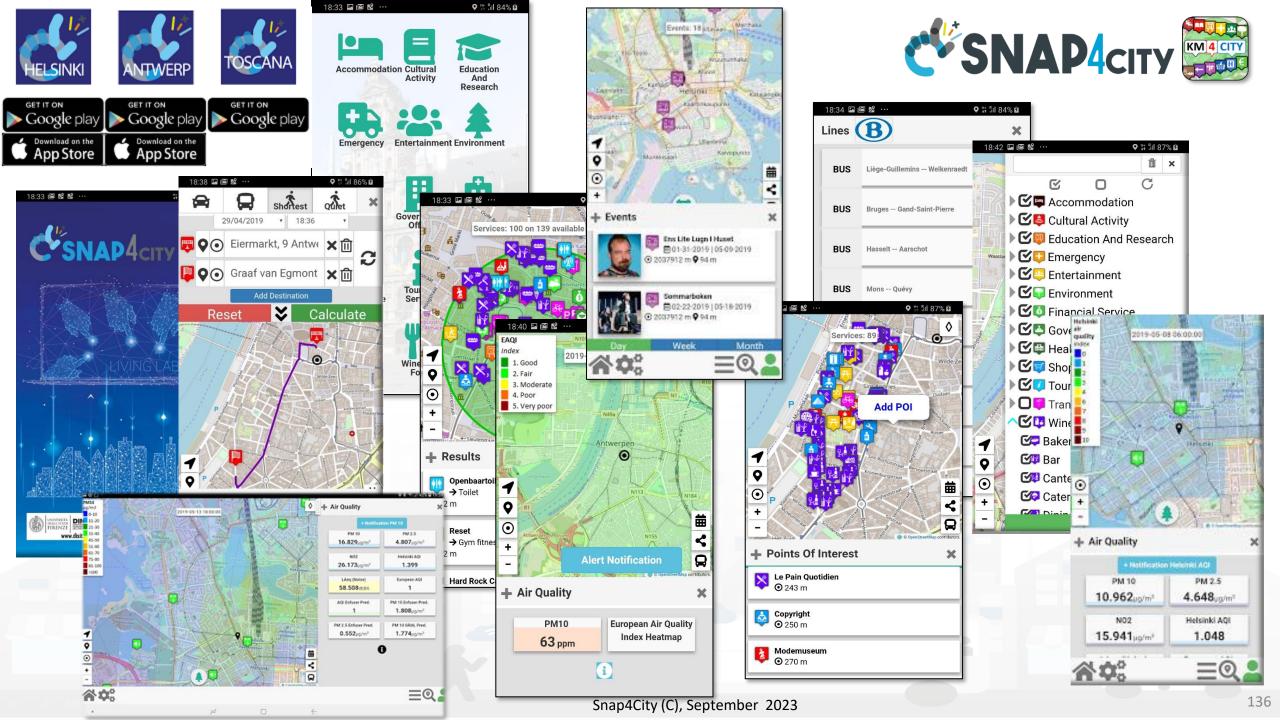




















Mobile App Features



- **Discovery** POI/services
- Search: POI, streets, suggestions
- Mobility and transport: Pub/priv, routing, car position, time table, park, sharing, tickets, etc.
- Environment and Weather: values, sensors, heatmaps, notifications
- Assistant, Forum, Developer Assistant
- Goal Experiences (Engagement)
- Personal data, activities, POI, tracking, IOT App, Dashboards, etc.
- Events: entertainment, critical
- Sharing position and trajectories with friends
- Monitoring city and personal Dashboards
- Personalized for Operators and Developers full control of their applications on cloud









MicroApplications











TOP

Understanding how City Users are using the City Services











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





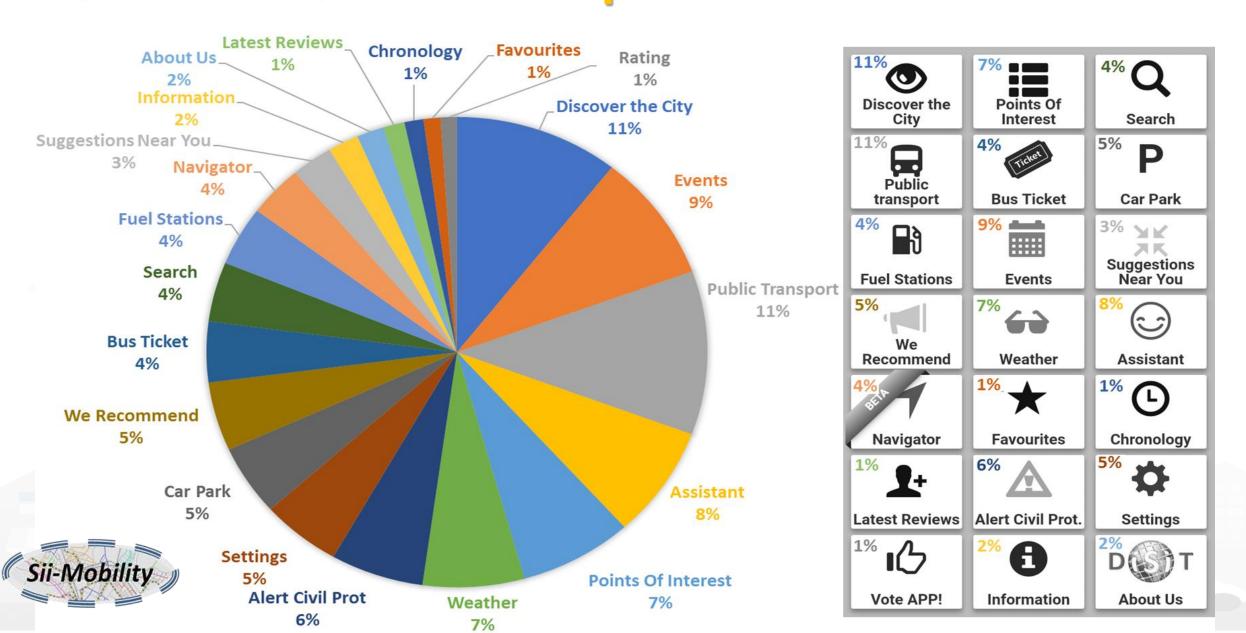




DISIT DISTRIBUTED SYSTEMS USERS' preferences CSNAP4CITY KM 4 CITY AND INTERNET TECHNOLOGIES LAB USERS' Preferences







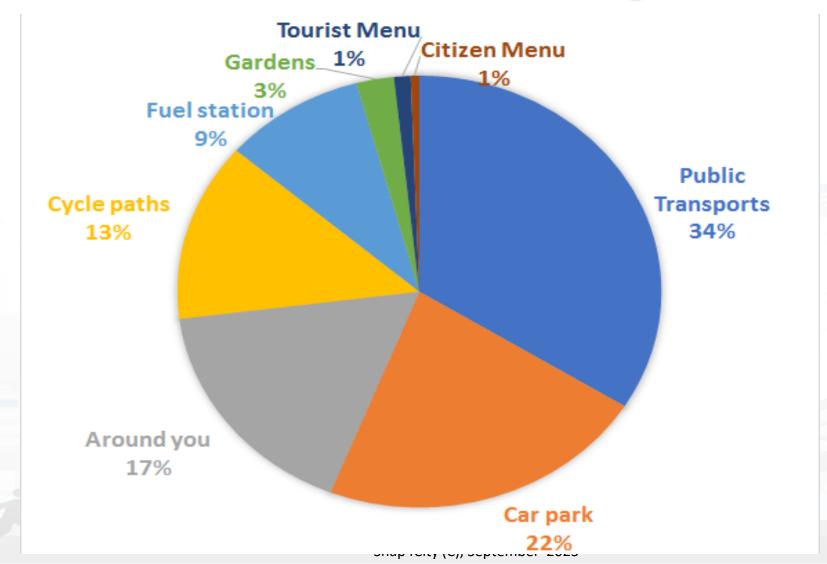








Preferred Users' Cathegories

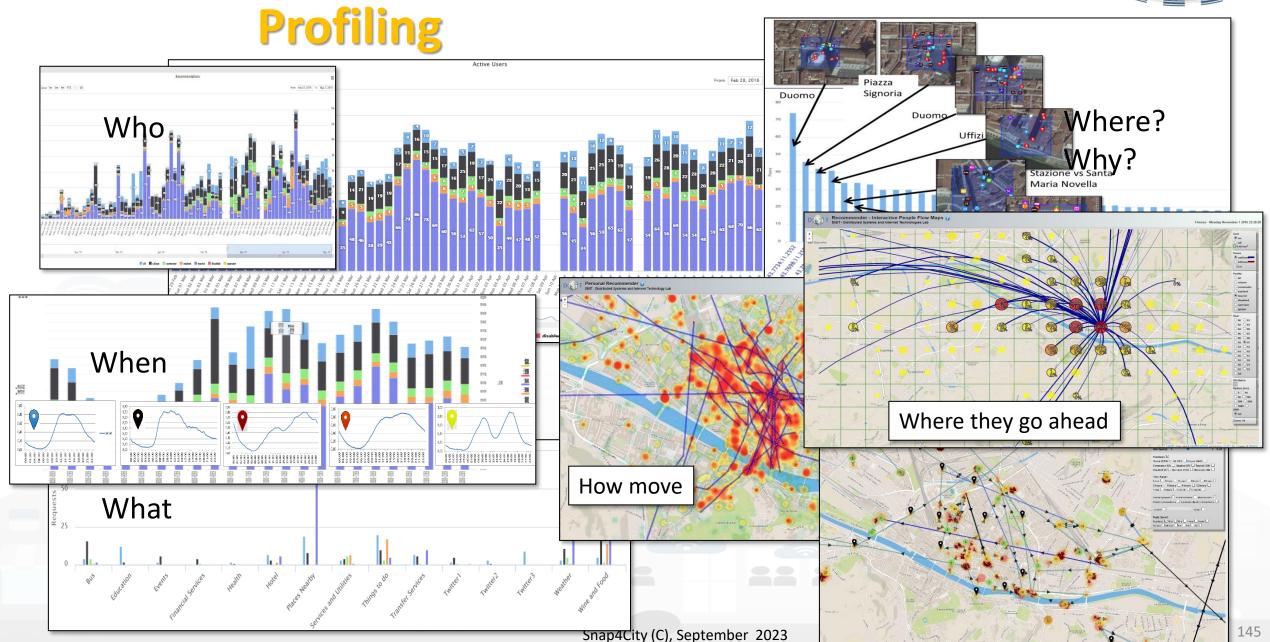






User Behavior Analyser for Collective





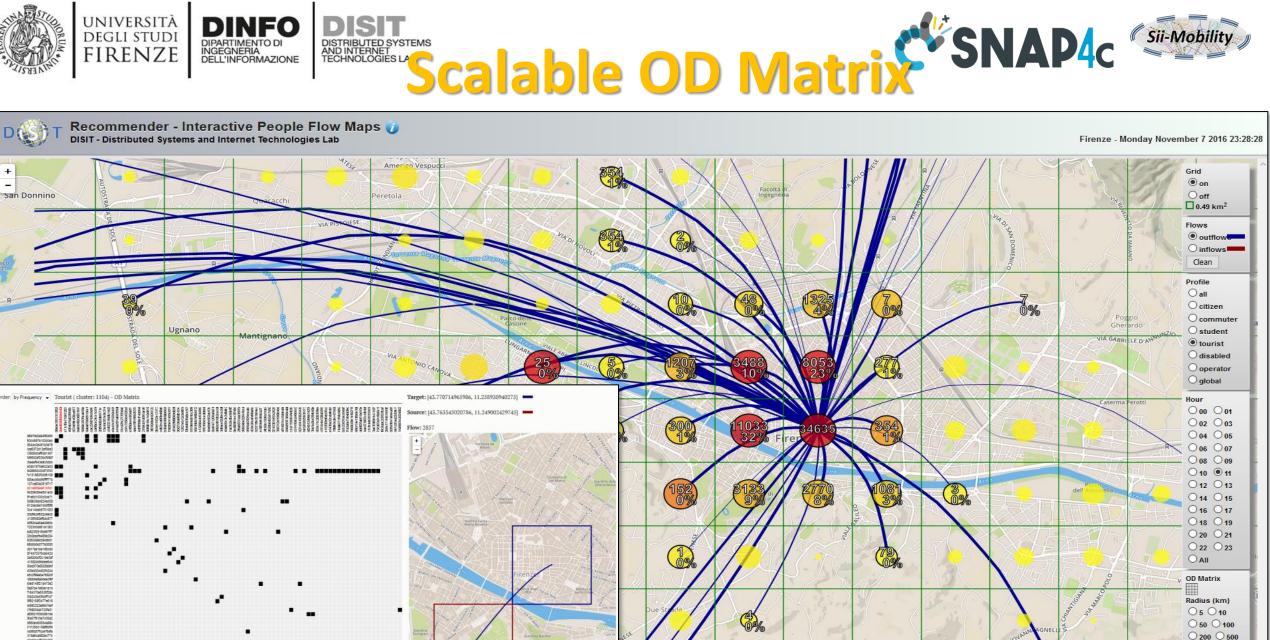






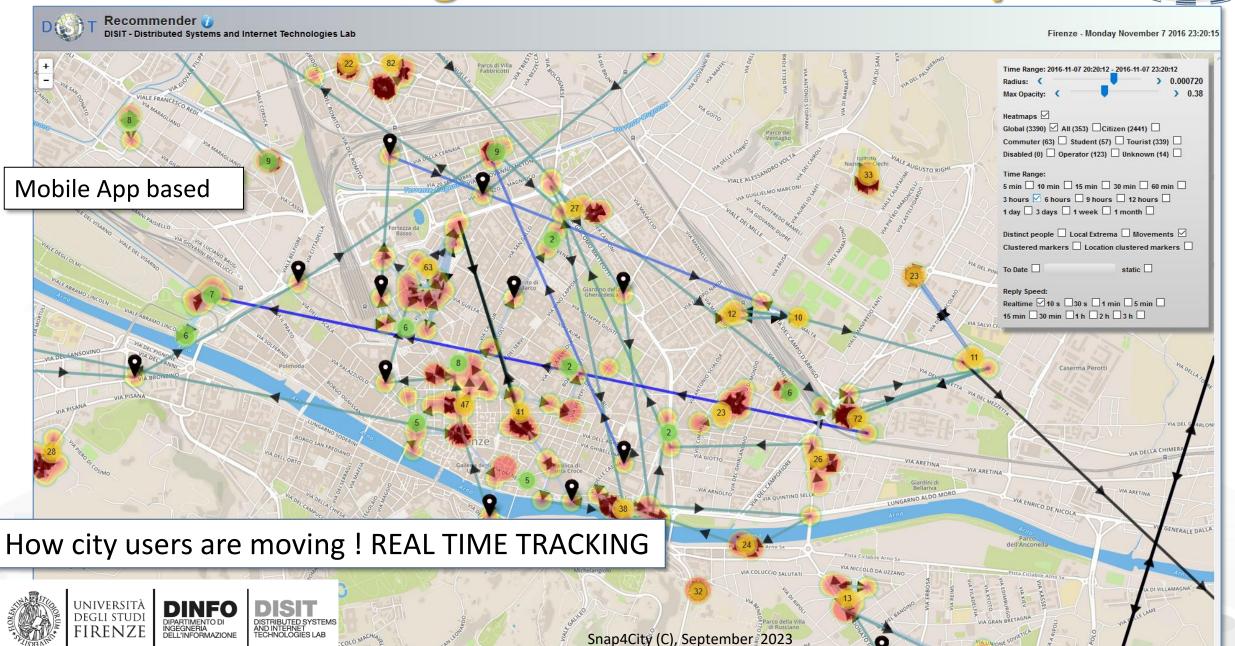


O 1000 O All Zoom: 14



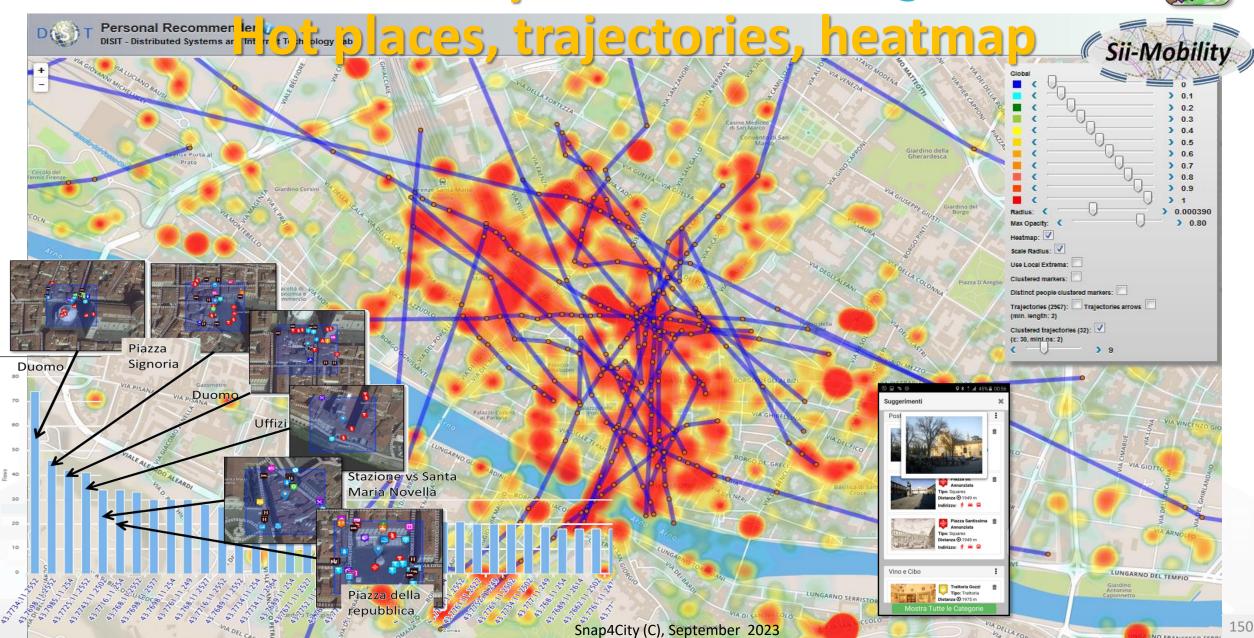
Real Time Traking: User Behaviour Analysis





User Behaviour Analyser









Understanding City User Behaviour

- Mobile Applications can send data via Advanced Smart City API to collect data about the city usage by the city users via a signed consent
 - See Mobile and Web App: Toscana in a Snap, Helsinki in a Snap, Antwerp in a Snap.
- City User behavior analysis includes production of:
 - suggestions, trajectories, hot places/heatmaps, etc.
 - origin destination matrices
 - data for the city user engagement
 - Etc.











https://www.snap4city.org/drupal/node/489









TOP

Engaging City Users Towards Virtuous Participated Attitude





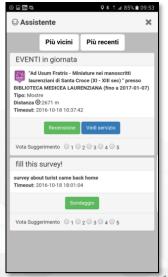


Profiled Engagements to City Users

- 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,















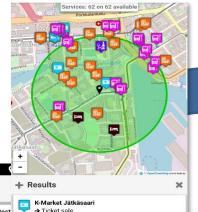
1 Engagement Sent (4 hours)

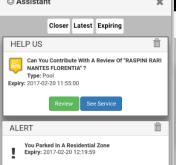


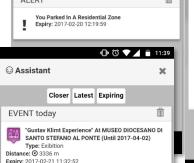
4 min DEngagemen... 4 min



Users' Engagement





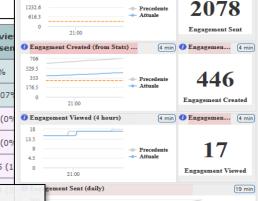


Help us to provide a better service	Î
Personalize Your Point-Of-Interest Expiry: 2017-02-20 19:35:39	
Can confirm that you LIVE around VIA TRIF	OLI?

⊖ Assistant



Rule name Type #sent #viewed #se 1 (0%) daily event de **ENGAGEMENT** 0 (0%) 0% 1720 (2.12%) 4.07 **ENGAGEMENT** 70 (7.1%) daily event en 5 (0.29%) 0 (0%) commuter 14 (0.81%) 0 (0%) 0 (09 student 1462 (85%) 25 (35.71%) tourist 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



Rules

City

context





Engaging City Users

- Mobile Applications can use Advanced Smart City API to collect data about the city usage by the city users via a signed consent
- It can be used for sending engagements to them such as to:
 - Inform
 - You have parked out of your residential parking zone
 - The Road cleaning is this night
 - The waste in S.Andreas Road is full
 - Engage
 - Please Provide a comment, a score, etc.
 - Stimulate / recommend
 - Events in the city, services you may be interested, etc..
 - Provide Bonus
 - Since you have parked here you can get 1 Bonus
 - We suggest you to leave the car out of the city, this bonus can be used to buy a bus ticket





users

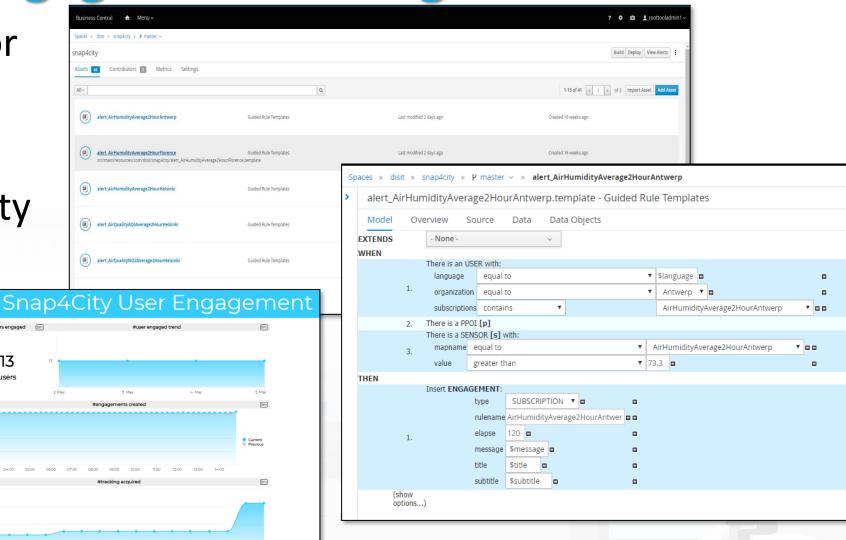


Engagement Manager

- Definition of Rules for campaigns
- Monitoring and follow-up for each City

Segmented for user

kind and interest



Sii smart. Sii-Mobility!

In palio per te

Carnet multicorsa Cap e voucher per:

Scarico

Dal 15 aprile al 1: trasporto pubblico Scarica l'app "Tos guadagna punti vi autobus e vinci tar



Dal 15 aprile al 15 luglio scegliere il trasporto pubblico ti premia! Scarica l'app "Toscana dove, cosa", quadagna punti viaggiando in autobus e vinci tanti fantastici premi! Per maggiori informazioni visita il sito info.sii-mobility@org



















In palio per te

Carnet multicorsa Cpt e

PISA Ci Prendiamo cura del tuo benessere

FARMACIE COMUNALI

voucher per:

Campaing on Sustainable Mobility

Snap4City (C), Septem

Sii smart. Sil-Mobility! Scarica, viaggia, vinci!



Dal 15 aprile al 15 luglio scegliere il trasporto pubblico ti premia! Scarica l'app "Toscana dove, cosa", guadagna punti viaggiando in autobus e vinci tanti fantastici premi. Per maggiori informazioni visita il sito info.sii-mobility.org



























Rules for Rewards

ASSISTANCE

- If public transport is detected after bus line suggestion on trajectory usually made on private transport → 10points
 - Why don't you take the bus line 4 in Piazza Marconi to reach your workplace? You save money, you respect the environment and you will be stress free for not worry about parking!
- Once a day, if public transport is detected after suggestion on an alternative bus line availability
 →3points
 - Why don't you take the bus line 4 that stop just 50 meters far from you? You save money, you respect the environment and you will be stress free for the traffic jam!
- If public transport is detected for at least 30(?)
 minutes a day → 1point

ENGAGEMENT

- Survey on commuter and their preferred way of mobility → 1point
 - How many minutes you usually commute to go to work? How do you rate the service?
- Feedback on public transport → 1point
 - Which current public transport are you using? Are the service in line with your expectation?
- Comments/Photo/Rate or survey on POI (public transport) → 1point
- Survey on use of the App after N days or for tourist coming home → 1point
- Feedback on PPOI or mobility → 1point



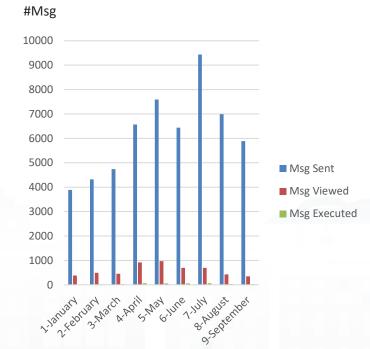


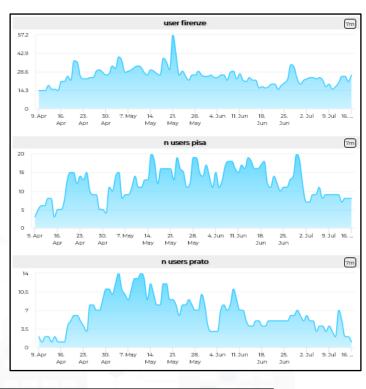


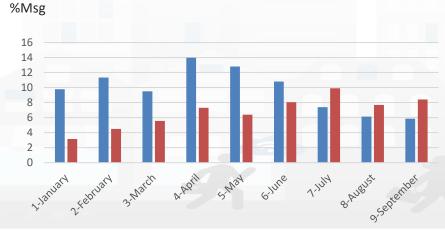


Validation of user Engagement

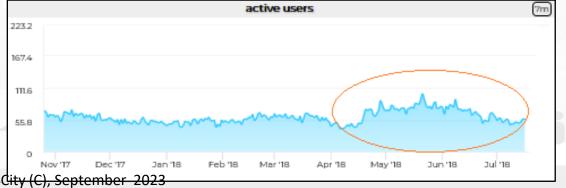
Months	Mcg Sont	Mcg Viewed	Msg Executed
IVIOITLIIS	ivisg sent	ivisg vieweu	IVISG EXECUTED
1-January	3888	380	12
2-February	4319	489	22
3-March	4739	450	25
4-April	6567	918	67
5-May	7594	972	61
6-June	6437	695	55
7-July	9432	697	69
8-August	6988	429	73
9-September	5885	345	49
Total	55849	5375	433











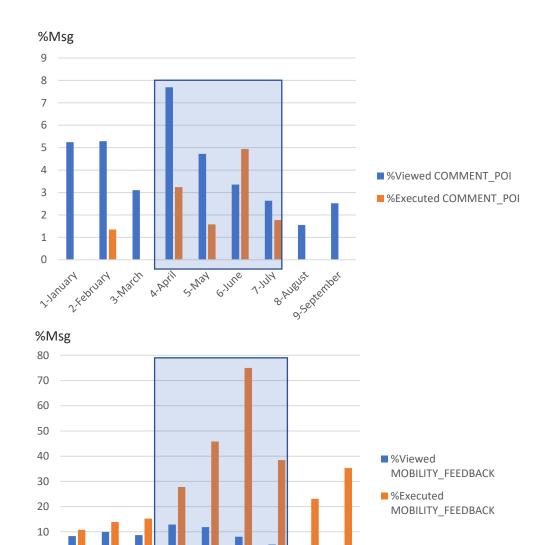
User Behaviour Analysis

VALIDATION

- During the PILOT new rules has been added (30 on a total of 80) and mostly all of them are still online
- COMMENT_POI: requires more user interaction and not very contextualized (POI proximity) → higher rate of sent, lower rate on execution
- MOBILITY_FEEDBACK: requires less user iteration and very contextualized (user in MOBILITY) → normal rate of sent, high rate on execution

	Msg Sent	Msg Viewed	Msg Executed
COMMENT_POI	21632	804	15
MOBILITY_FEEDBACK	5378	371	94



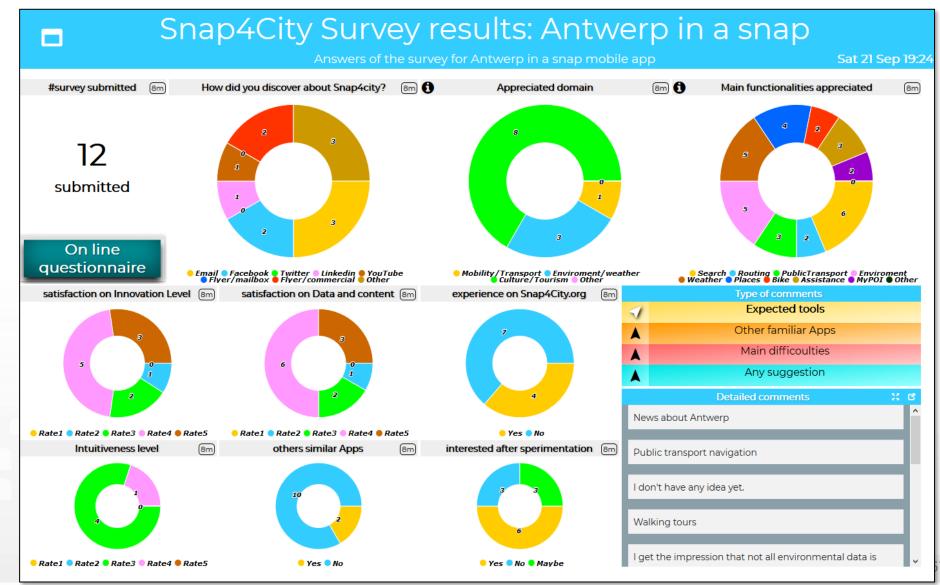






https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc2OQ==

Dashboard
created to monitor
in real time the
answers to the
survey provided
on the Mobile
App directly by the
Engagement tool







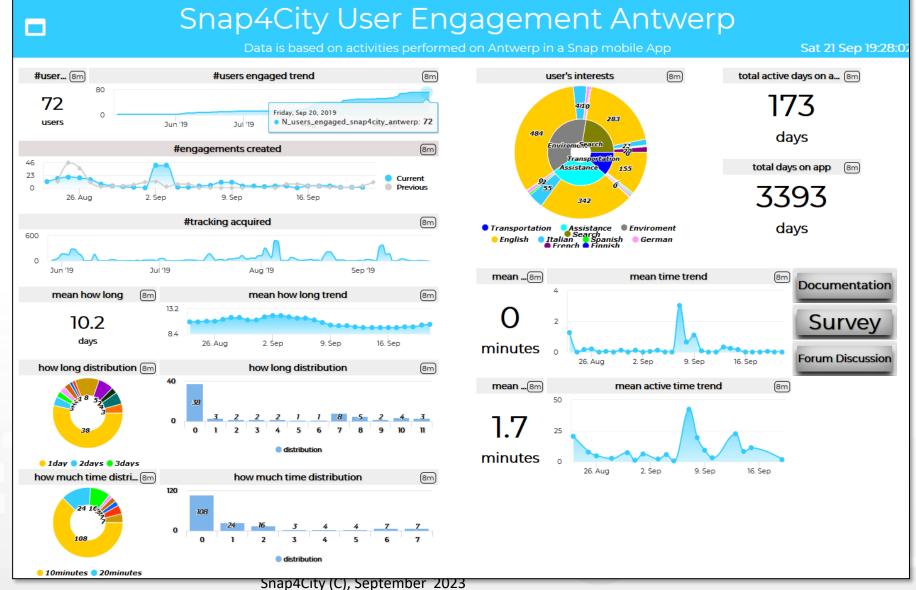




https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc1OQ==

Dashboard monitoring the Mobile App:

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











TOP

Connected Drive





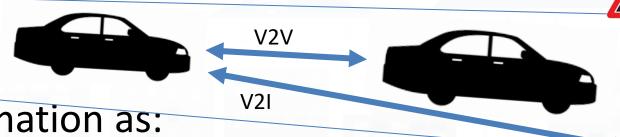


Main Concept of Connected Drive

 Different kinds of communications may arrive on the vehicles on board devices



Mobile Phones can be a possible facilities



- Geolocated Real time Information as:
 - Alerting, dynamic digital signage (may not present physically on the road)
 - Supporting autonomous driving vehicles



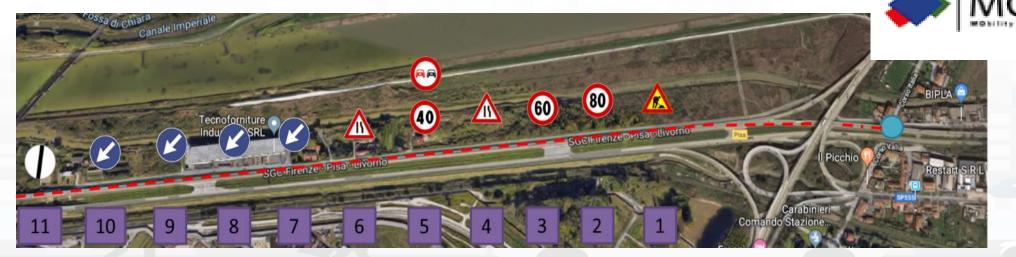






Experimentation on «Toscana Dove Cosa»

- Mobile App supporting connected Drive V2I connections:
 - https://play.google.com/store/apps/details?id=org.disit.toscana&hl=it
 - https://apps.apple.com/it/app/toscana-where-what-km4city/id1064554200
 - For the MOSAIC project and pilot in Tuscany
- The mobile App has a Navigator which includes now the acquisition of connected drive messages



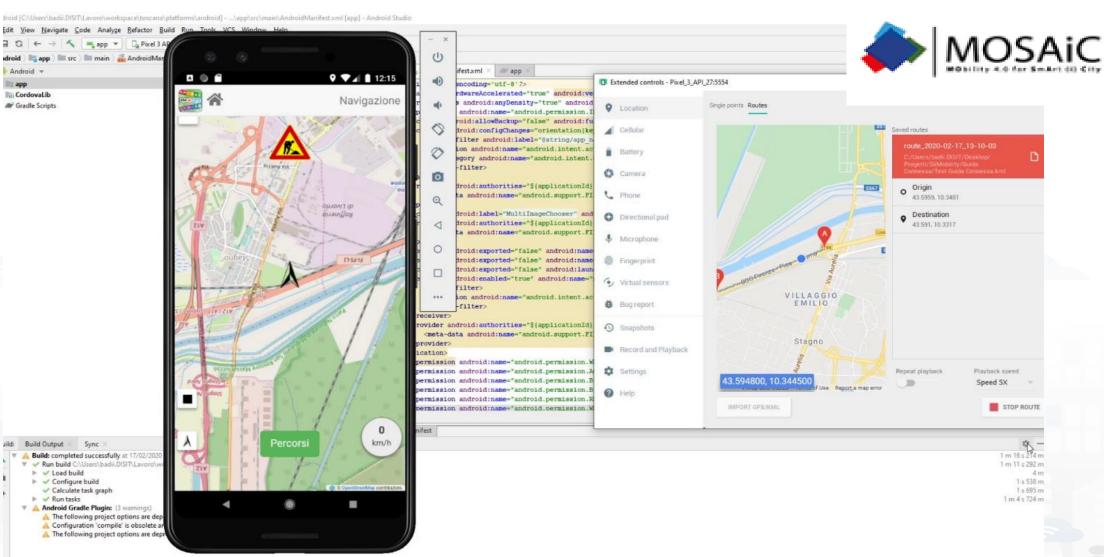




DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

Scenario





SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES





SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES











- Apache Cordova is a set of JavaScript APIs that enable the devices to the application developer
 to access native features of the device such as the camera or accelerometer, storage, network,
 gps
- Combined with a user interface framework such as Dojo Mobile or jQuery Mobile or Sencha Touch, allows the development of smartphone applications using only HTML, CSS and JavaScript.
- When using the Cordova API, an application can be built without any native code (Java, Objective-C, C# etc.). The **web technologies** used are **hosted in the same application** at the local level (usually not on a remote http server).
- These JavaScript API are consistent and valid for the different platforms of mobile devices, in this way the application built on the Web standard, should be portable with a minimum of changes.





Mustache JS

- The library is **independent** from specific framework but there are plugins for the integration with jQuery, Dojo, and YUI.
- Possibility to work with javascript objects and then exploit the communication of data in JSON format from a REST call via AJAX.
- The **templates** for Mustache may be assigned or loaded as a string to a variable and the placeholder are identified by two braces, for example: {{miopplaceholder}}.
- One of the most interesting of the library feature is support in enumerable values
- Documentation and downloads are available on the official website: http://mustache.github.io









Mustache JS

JSON -

```
var data = {
    risultato: true,
    titolo: Città italiane,
    descrizione: Lista delle città italiane,
    citta: [
        {nome: Milano, sigla: MI},
        {nome: Roma, sigla: RM}
    ]
};
```









Mustache JS

risultato: true,
titolo: Città italiane,
descrizione: Lista delle città italiane,
citta: [
{nome: Milano, sigla: MI},
{nome: Roma, sigla: RM}
};

var data = {

Template + JSON + Mustache

Città italiane

Lista delle città italiane

- Milano (MI)
- Roma (RM)







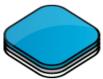
- OpenLayers is an open source JavaScript library for displaying map data in web browsers and can be used with a hybrid application developed with Cordova
- In the **early versions** of the app, the map was managed by **Leaflet.js** library. This was replaced because it didn't support the rotation, which is required to insert navigation functions within the app
- In addition, OpenLayers 3.0 builds the map and objects added to it
 with a canvas renderer, which is very efficient when objects are numerous and small
 as the markers displayed for each search done with the app
- Documentation and downloads are available on the official website: http://openlayers.org



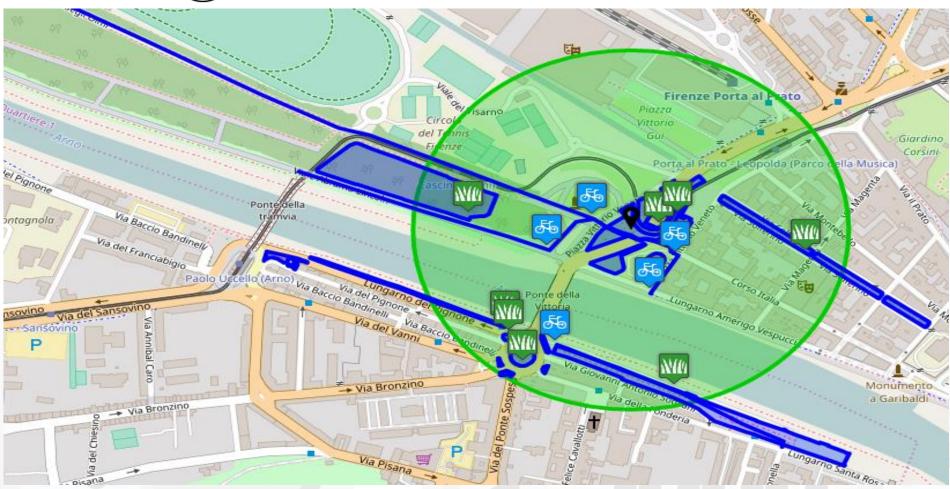








OpenLayers 3.0





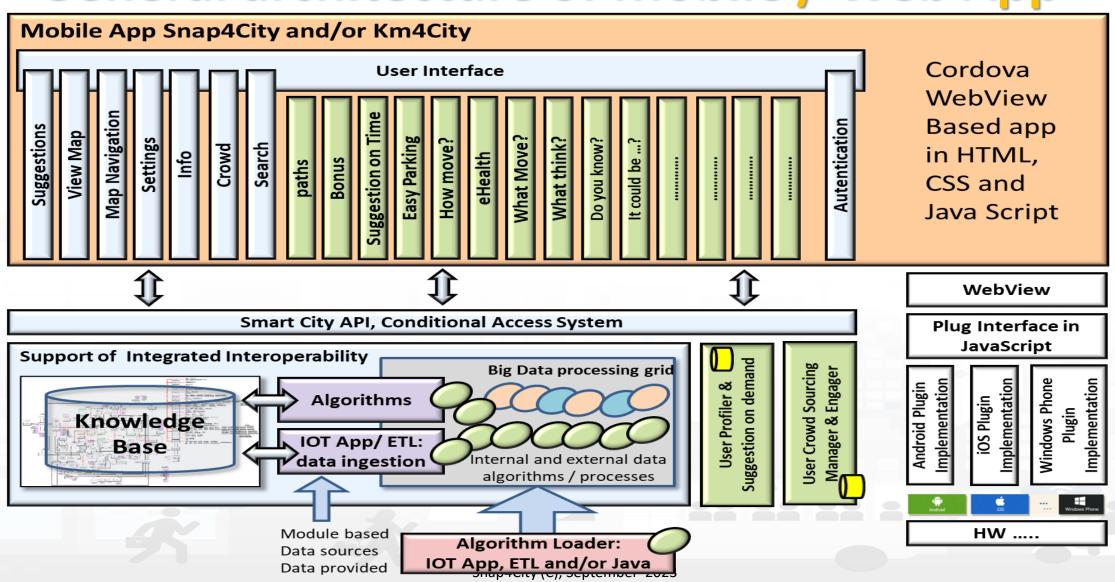








General architecture of Mobile / Web App







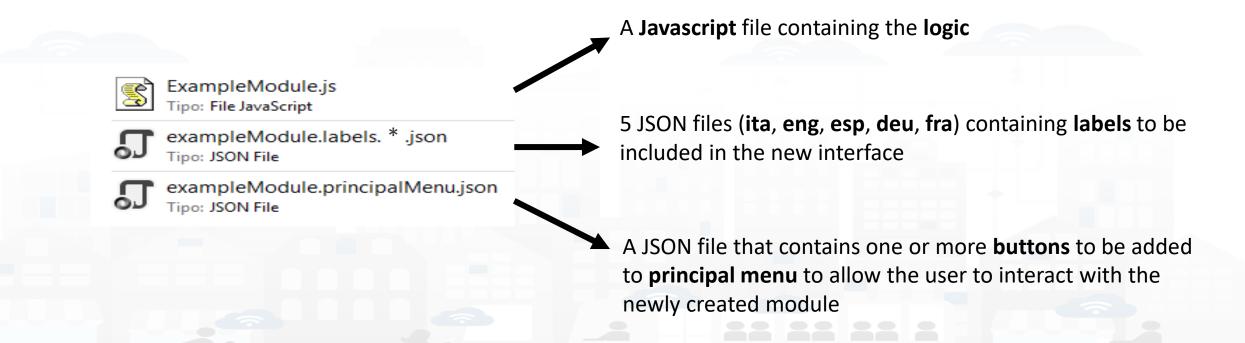
In the slides following there is an **example** of how to **add a module** to the app.

The goal of this example is to create a **new module** that in addition to viewing the list of car parks as is already the case for the button named "Parking" will **show directly** the **number of free parking lots** for each car park found





• Files required for creating a new module are as follows

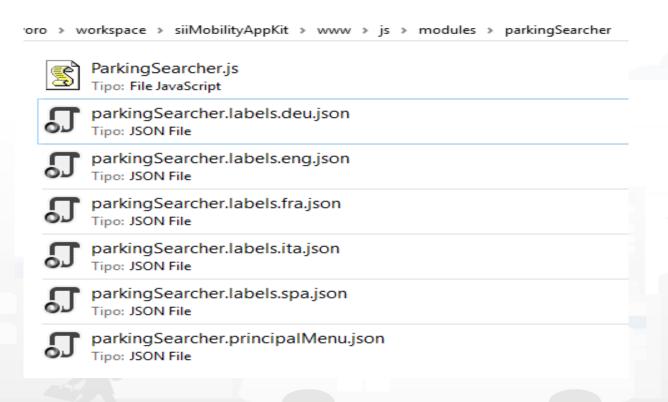








Copy these files to a new folder that will have the name of the new module (i.e.,
 ParkingSearcher): the names of the files copied have to be changed to get the module name as
 a prefix







• Field descriptions for creating buttons in the main menu

```
"iconId": "",
"iconClass": "icon ion-android-bus",
"iconFontSize": "41px",
"iconColor": "#CC0000",
"imgSrc": "img/ticketmenu.png",
"imgHeight": "37px",
"text": "P",
"textFontSize": "38px",
"textColor": "#CC0000",
"captionId": "principalMenuParkingSearcher",
"captionTextId": "moduleParkingSearcher",
"step": true,
"stepId": "eventsBadge",
"ribbon": true,
"ribbonId": "",
"ribbonStyle": "background: #336633;background: linear-gradient(#33FF33 0%, #336633 100%);
"ribbonText": "Beta",
"removed": false,
"index": 0
```

This field contains the **callback** for the new module.

The present callbacks should be left, because they serves to close the main menu and to center the map on the GPS





• Field descriptions for creating buttons in the main menu

```
"iconClass": "icon ion-android-bus"
"iconFontSize": "41px",
"iconColor": "#CC0000"
"imgSrc": "img/ticketmenu.png",
"imgHeight": "37px",
"textFontSize": "38px",
captionid": "principalMenuParkingSearcher",
"captionTextId": "moduleParkingSearcher",
"step": true,
"stepId": "eventsBadge",
"ribbon": true,
"ribbonId": "",
"ribbonStyle": "background: #336633;background: linear-gradient(#33FF33 0%, #336633 100%);
"ribbonText": "Beta",
"removed": false,
"index": 0
```

These blocks of fields are **mutually exclusive**. Allow you to choose the icon that will identify the button that you are creating. This icon can be chosen as an **image**, a **text**, a **glyphicon** (Bootstrap) or **ionicons** (ionicons.com).

N.B. Field **iconId** can be useful if you plan to edit the selected icon **dynamically**





• Field descriptions for creating buttons in the main menu

```
"iconId": "",
"iconClass": "icon ion-android-bus",
"iconFontSize": "41px",
"iconColor": "#CC0000"
"imgSrc": "img/ticketmenu.png",
"imgHeight": "37px",
"textFontSize": "38px",
captionid: "principalMenuParkingSearch
"captionTextId": "moduleParkingSearcher
"step": true,
"stepId": "eventsBadge",
"ribbon": true,
"ribbonId": "",
"ribbonStyle": "background: #336633;background: linear-gra
                                                      ent(#33FF33 0%,
"ribbonText": "Beta",
"removed": false,
"index": 0
                            Trasporto Pubblico
                                                Biglietti Bus
                                                                  Parcheggi
```

These blocks of fields are **mutually exclusive**. Allow you to choose the icon that will identify the button that you are creating. This icon can be chosen as an **image**, a **text**, a **glyphicon** (Bootstrap) or **ionicons** (ionicons.com).

N.B. Field **iconId** can be useful if you plan to edit the selected icon **dynamically**





Field descriptions for creating buttons in the main menu

```
"iconId": "",
"iconClass": "icon ion-android-bus",
"iconFontSize": "41px",
"iconColor": "#CC0000",
"imgSrc": "img/ticketmenu.png",
"imgHeight": "37px",
"text": "P",
"textFontSize": "38px",
"textColor": "#CC0000",
"captionId": "principalMenuParkingSearcher",
"captionTextId": "moduleParkingSearcher",
"step": true,
"stepId": "eventsBadge",
"ribbon": true,
"ribbonId": "",
"ribbonStyle": "background: #336633;background: linear-gradient(#33FF33 0%, #336633 100%);
"ribbonText": "Beta",
"removed": false,
"index": 0
```

captionId serves to indicate the **container tag** of the text that is located at the bottom of each button.

captionTextId indicates the name of the field in labels.*.json whose value is the text to be inserted in the previous container.





Field descriptions for creating buttons in the main menu

```
"iconId": "",
"iconClass": "icon ion-android-bus",
"iconFontSize": "41px",
"iconColor": "#CC0000",
"imgSrc": "img/ticketmenu.png",
"imgHeight": "37px",
"text": "P",
"textFontSize": "38px",
"textColor": "#CC0000",
"captionId": "principalMenuParkingSearcher",
"captionTextId": "moduleParkingSearcher",
"step": true,
"stepId": "eventsBadge",
'ribbon": true,
"ribbonId": "",
"ribbonStyle": "background: #336633;background: linear-gradient(#33FF33 0%, #336633 100%);
"ribbonText": "Beta",
"removed": false,
"index": 0
```

These blocks of fields are used to show the user **badges containing information** related to the button on which are located





• Field descriptions for creating buttons in the main menu

```
"iconId": "",
"iconClass": "icon ion-android-bus",
"iconFontSize": "41px",
"iconColor": "#CC0000",
"imgSrc": "img/ticketmenu.png",
"imgHeight": "37px",
"text": "P",
"textFontSize": "38px",
"textColor": "#CC0000",
"captionId": "principalMenuParkingSearcher",
"captionTextId": "moduleParkingSearcher"
"step": true,
"stepId": "eventsBadge",
'ribbon": true,
"ribbonId": "",
"ribbonStyle": "background: #336633;background: linear-gradient(#33FF33 0%, #336633 100%)
"ribbonText": "Beta",
"removed": false,
"index": 0
```

These blocks of fields are used to show the user **badges containing information** related to the button on which are located





• Field descriptions for creating buttons in the main menu

```
"iconId": "",
"iconClass": "icon ion-android-bus",
"iconFontSize": "41px",
"iconColor": "#CC0000",
"imgSrc": "img/ticketmenu.png",
"imgHeight": "37px",
"text": "P",
"textFontSize": "38px",
"textColor": "#CC0000",
"captionId": "principalMenuParkingSearcher",
"captionTextId": "moduleParkingSearcher",
"step": true,
"stepId": "eventsBadge",
"ribbon": true,
"ribbonId": "",
"ribbonStyle": "background: #336633;background: linear-gradient(#33FF33 0%, #336633 100%);
"ribbonText": "Beta",
"removed": false,
"index": 0
```

removed field is useful to allow the removal and the insertion of the buttons in the main menu by the user.

index field is useful for rendering the buttons in the order chosen by the user.







• Field descriptions for creating buttons in the main menu

```
What do you want to do?
                                                                                                                               What do you want to do?
"iconId": "",
"iconClass": "icon ion-android-bus",
                                                                                                                                                 Points Of
Interest
"iconFontSize": "41px",
"iconColor": "#CC0000",
                                                                                                                                                D
"imgSrc": "img/ticketmenu.png",
                                                                                                                 D T
"imgHeight": "37px",
"text": "P",
                                                                                               Events
                                                                                                        Settings
"textFontSize": "38px",
"textColor": "#CC0000",
"captionId": "principalMenuParkingSearcher",
"captionTextId": "moduleParkingSearcher",
"step": true,
"stepId": "eventsBadge",
"ribbon": true,
"ribbonId": "",
"ribbonStyle": "background: #336622 Jackground: linear-gradient(#33FF33 0%, #336633 100%);
"ribbonText": "Beta",
"removed": false,
"index": 0
```





- Loading new buttons modules within the main menu, takes place by comparing the captionId field.
- If the menu already has a button with the **same captionId**, the first is **replaced** with the **new one**.
- To remove a button from the main menu (field removed hides it) add a delete field with value equal to true.







First version of the button

```
What do you want to do?
"callback": "PrincipalMenu.hide(); MapManager.centerMapOnGps();",
"iconId": "",
                                                                                                                                  Punti di
                                                                                                                                             Q
"iconClass": ""
"iconFontSize": "",
                                                                                                                     Scopri la Città
                                                                                                                                             Ricerca
"iconColor": ""
                                                                                                                                              繭
                                                                                                                      Trasporto
Pubblico
"imgSrc": ""
"imgHeight": "",
                                                                                                                                 Parcheggi
                                                                                                                                              Eventi
"text": "LP",
                                                                                                                                              A
"textFontSize": "38px",
"textColor": "#CC0000",
                                                                                                                                 Impostazioni
                                                                                                                                            Informazioni
"captionId": "principalMenuParkingSearcher",
                                                                                                                      D
"captionTextId": "moduleParkingSearcher",
"step": "",
                                                                                                                                            remote poi
"stepId": "",
"ribbon": true,
"ribbonId": ""
"ribbonStyle": "background: #CC0000; background: linear-gradient(#FF6600 0%, #CC0000 100%);".
"ribbonText": "NEW",
                                                                                                                                      Label missing
"removed": false,
"index": 0
```

parkingSearcher.principalMenu.json





Labels of ParkingSearcher

• Description of label.*.json files

```
label.ita.json

{
    "principalMenu": {
        "moduleParkingSearcher": "Lista Parcheggi"
    }
}

label.eng.json

{
    "principalMenu": {
        "moduleParkingSearcher": "Car Park List"
    }
}

label.deu.json

{
    "principalMenu": {
        "moduleParkingSearcher": "Parkplatz Liste"
    }
}
```

```
label.fra.json

{
    "principalMenu": {
        "moduleParkingSearcher": "Liste parkings"
}

label.esp.json

{
    "principalMenu": {
        "moduleParkingSearcher": "Lista de Aparcamiento"
}
}
```

Three important things to check:

- Languages shall be indicated by 3 characters: ita, deu, esp, fra, eng
- The label for the button must be contained within the object "principalMenu"
- The name of the field inside "principalMenu" must be the same of "captionTextId" seen before





Labels of ParkingSearcher

• Description of label.*.json files

```
label.ita.json
                                                                          'principalMenu": {
                                                                           "moduleParkingSearcher" "Lista Parcheggi'
  "iconId": "",
                                                                                  label.eng.json
  "iconClass": ""
  "iconFontSize": ""
                                                                          'principalMenu": {
  "iconColor": "",
                                                                           "moduleParkingSearcher": "Car Park List"
  "imgSrc": "",
  "imgHeight": "",
  "text": "LP".
                                                                                  label.deu.json
  "textFontSize": "38px".
  "textColor": "#CC0000",
                                                                          "principalMenu": {
  "captionId": "principalMenuParkingSearcher"
                                                                           "captionTextId": "moduleParkingSearcher",
  "step": "",
  "stepId": "",
                                                                                   label.fra.json
  "ribbon": true,
  "ribbonId": "".
                                                                          'principalMenu": {
  "ribbonStyle": "background: #CC0000;background: linear-gradient(#
                                                                            'moduleParkingSearcher': "Liste parkings
  "ribbonText": "NEW",
  "removed": false,
  "index": 0
                                                                                  label.esp.json
                                                                          "principalMenu": {
parkingSearcher.principalMenu.json
                                                                           "moduleParkingSearcher": "Lista de Aparcamiento"
```

\$(captionId).html(
labels.principalMenu[
captionTextId]);







Labels of ParkingSearcher

• Description of label.*.json files

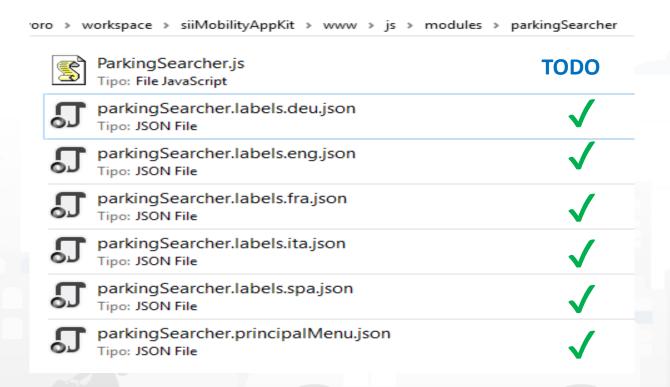








 It is seen as fill most of the files in the folder of new module ParkingSearcher that is developed in this presentation









ParkingSearcher Module Functions

Functions contained in ParkingSearcher.js

```
show: function () {
    application.resetInterface();
    MapManager.showMenuReduceMap("#" + ParkingSearcher.idMenu);
    $("#" + ParkingSearcher.idMenu + "Collapse").hide();
    ParkingSearcher.open = true;
    InfoManager.addingMenuToManage(ParkingSearcher.varName);
    application.addingMenuToCheck(ParkingSearcher.varName);
    application.setBackButtonListener();
},

hide: function () {
    $("#" + ParkingSearcher.idMenu).css({ 'z-index': '1001' });
    MapManager.reduceMenuShowMap("#" + ParkingSearcher.idMenu);
    InfoManager.removingMenuToManage(ParkingSearcher.varName);
    application.removingMenuToCheck(ParkingSearcher.varName);
    ParkingSearcher.open = false;
},
```

Closes any previously opened menu, shrinks the map to display the menu, hides the button to reduce the menu, since it will open already reduced.

Recording to other variables to get notifications when:

- users press the back button
- users change the device orientation
- must be **closed the menu** opened by this module







ParkingSearcher Module Functions

Functions contained in ParkingSearcher.js

```
show: function () {
    application.resetInterface();
    MapManager.showMenuReduceMap("#" + ParkingSearcher.idMenu);
    $("#" + ParkingSearcher.idMenu + "Collapse").hide();
    ParkingSearcher.open = true;
    InfoManager.addingMenuToManage(ParkingSearcher.varName);
    application.addingMenuToCheck(ParkingSearcher.varName);
    application.setBackButtonListener();
},

hide: function () {
    $("#" + ParkingSearcher.idMenu).css({ 'z-index': '1001' });
    MapManager.reduceMenuShowMap("#" + ParkingSearcher.idMenu);
    InfoManager.removingMenuToManage(ParkingSearcher.varName);
    application.removingMenuToCheck(ParkingSearcher.varName);
    ParkingSearcher.open = false;
},
```

Does the **opposite functions** to those performed by the **function show**, also reset the z-indexof the menu







Functions contained in ParkingSearcher.js

```
checkForBackButton: function () {
    if (ParkingSearcher.open) {
        ParkingSearcher.hide();
refreshMenuPosition: function () {
    if (ParkingSearcher.open) {
        MapManager.showMenuReduceMap("#" + ParkingSearcher.idMenu);
        Utility.checkAxisToDrag("#" + ParkingSearcher.idMenu);
        if (ParkingSearcher.expanded) {
            ParkingSearcher.expandBusRoutesMenu();
closeAll: function () {
    if (ParkingSearcher.open) {
        ParkingSearcher.hide();
```

These are the **callbacks** called to **notify** the occurrence of an event among those described previously (see show function) and for which we recorded the module

- users press the back button
- users change the device orientation
- must be closed the menu opened by this module





Functions contained in ParkingSearcher.js

- Checks if there is the element that will contain the html code created through the use of Mustache library.
- It is generated the html code with **template ParkingMenu.mst.html** and **JSON ParkingSearcher.results** and added to the element container.
- Finally, the feature that allows the users to widen the menu by dragging the handler is added to it





Functions contained in ParkingSearcher.js

- Checks if there is the element that will contain the html code created through the use of Mustache library.
- It is generated the html code with **template ParkingMenu.mst.html** and **JSON ParkingSearcher.results** and added to the element container.
- Finally, the feature that allows the users to widen the menu by dragging the handler is added to it





Functions contained in ParkingSearcher.js

```
successQuery: function (response) {
    ParkingSearcher.results = responseObject["Results"];
    ParkingSearcher.refreshMenu();
    ParkingSearcher.show();
    MapManager.addGeoJSONLayer(responseObject);
    ParkingSearcher.resetSearch();
},

errorQuery: function(error) {
    navigator.notification.alert(
        Globalization.alerts.servicesServerError.message,
        function () { },
        Globalization.alerts.servicesServerError.title);
},
```

These are the callbacks that should be called once the **JSON**, containing the **data to be displayed** to the user, is created. The **success callback**:

- will locally save the response
- will create the menu
- will show it.

If the menu will contain **elements** that it is possible to **show on the map** they will be added to the map by last function





• Before adding the logic of the new module, we create the template to be filled with the correct JSON.

```
div id="parkingMenuHeader" class="panel panel-default" style="position: absolute;right: 0px;left: 0px;border-radius: 0px;"
   <div id="parkingMenuExpandHandler" class="grippyContainer grippyContainer-horizontal" style="text-align: center;">
       <div class="grippy grippy-horizontal"></div>
   <div class="panel-heading" style="padding: 0px 10px;height: 52px; border: none;">
       <a class="pull-right" onclick="ParkingSearcher.hide();">
           <i class="glyphicon glyphicon-remove"</pre>
              style="float: right; padding-left: 8px; color: #777; line-height: 52px;"></i></i>
       <a id="parkingMenuExpand" class="pull-left" onclick="ParkingSearcher.expandParkingSearcher();">
           <i class="glyphicon glyphicon-plus" style="padding-right: 8px; color: #777; line-height: 52px;"></i>
       <a id="parkingMenuCollapse" class="pull-left" onclick="ParkingSearcher.collapseParkingSearcher();">
           <i class="glyphicon glyphicon-minus" style="padding-right: 8px; color: #777; line-height: 52px;"></i></i>
       <b id="parkingMenuHeaderTitle" style="line-height: 52px;color: #333;">
               $("#parkingMenuHeaderTitle").html(
                   Globalization.labels.parkingMenu.title)
           </script>
div id="parkingMenuInner" class="commonHalfMenuInner">
```

This default template will simply show a menu with a header and body empty. Must have the same name as the string entered as the third parameter in the call

```
ViewManager.render (
ParkingSearcher.results,
"#" + ParkingSearcher.idMenu,
"ParkingMenu");
```

ParkingMenu.mst.html





• Before adding the logic of the new module, we create the template to be filled with the correct JSON.

This template will be saved in the folder called ***templates***.

To add a title to the header we should add this item to all files labels.*. Ison

```
"principalMenu": {
    "moduleParkingSearcher": "Lista Parcheggi"
}.
"parkingMenu": {
    "title": "Parcheggi"
}
```

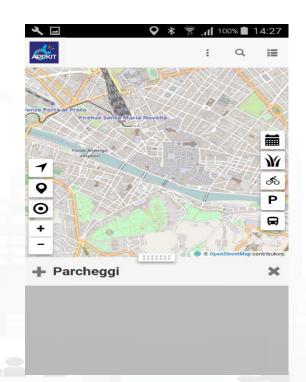
templates/ParkingMenu.mst.html





• Before adding the logic of the new module, we create the template to be filled with the correct JSON.

```
div id="parkingMenuHeader" class="panel panel-default" style="position: absolute;right: 0px;left: 0px;border-radius: 0px;">
   <div id="parkingMenuExpandHandler" class="grippyContainer grippyContainer-horizontal" style="text-align: center;">
       <div class="grippy grippy-horizontal"></div>
   <div class="panel-heading" style="padding: 0px 10px;height: 52px; border: none;">
       <a class="pull-right" onclick="ParkingSearcher.hide();">
           <i class="glyphicon glyphicon-remove"</pre>
              style="float: right; padding-left: 8px; color: #777; line-height: 52px;"></i></i>
       <a id="parkingMenuExpand" class="pull-left" onclick="ParkingSearcher.expandParkingSearcher();">
           <i class="glyphicon glyphicon-plus" style="padding-right: 8px; color: #777; line-height: 52px;"></i>
       <a id="parkingMenuCollapse" class="pull-left" onclick="ParkingSearcher.collapseParkingSearcher();">
           <i class="glyphicon glyphicon-minus" style="padding-right: 8px; color: #777; line-height: 52px;"></i></i>
       <b id="parkingMenuHeaderTitle" style="line-height: 52px;color: #333;">
               $("#parkingMenuHeaderTitle").html(
                   Globalization.labels.parkingMenu.title)
           </script>
div id="parkingMenuInner" class="commonHalfMenuInner">
```



templates/ParkingMenu.mst.html





Create ParkingSearcher Module

The goal of this example is to create a **new module** that in addition to viewing the list of car parks as is already the case for the button named "Car Park" will **show directly** the **number of free parking lots** for each car park found

In ParkingSearcher.js must be made the logic that **retrieves data** from API describer in previous presentations and creates the **JSON** to fill the **template** and generate the new menu





• The following API returns **the list of parking** that are located at a maximum distance of 300 meters from the location sent. The list is limited to 100 items.

```
http://www.disit.org/ServiceMap/api/v1/?
selection=43.7778;11.2481&
categories=Car_park&
maxResults=100&
maxDists=0.3&
format=json&
lang=it&
geometry=true
```







• The returned data are not sufficient to create the final JSON, because these data are lacking on the realtime information

```
▼ object {1}
   ▼ Services {3}
         fullCount : 5
         type : FeatureCollection
         features [5
             ▼ geometry {2}
                   type : Point
                ▶ coordinates [2]
                type : Feature
                properties
                   name : Garage La Stazione Spa
                   tipo : Parcheggio auto
                   typeLabel: Parcheggio auto
                   serviceType: TransferServiceAndRenting Car park
                   hasGeometry:  false
                   serviceUri: http://www.disit.org/km4citv/resource/RT04801702315P0
                   multimedia : value
```

There are data from all car parks nearby, but there are few properties that are received





• The following API which returns all information relating to a single service

http://www.disit.org/ServiceMap/api/v1/?
serviceUri=http://www.disit.org/km4city/resource/RT04801702315PO&
format=json&
lang=it











 The returned data are not sufficient to create the final JSON, because these data are relative to only one car park



There are data from **one car parks nearby**, but there are many properties that are received





The idea is to call the first API that returns the complete
list of nearby car park, and for each car park in the list call
the second API that returns detailed information with the
number of free parking lots





• The first API can be call in the app with the following functions

```
search: function(){
   var parkingQuery = QueryManager.createCategoriesQuery(['Car_park'], SearchManager.searchCenter, "user");
   APIClient.executeQuery(parkingQuery,ParkingSearcher.searchInformationForEachFeature,ParkingSearcher.errorQuery);
},
```

http://www.disit.org/ServiceMap/api/v1/?
selection=43.7778;11.2481&
categories=Car_park&
maxResults=100&
maxDists=0.3&
format=json&
lang=it&
geometry=true

The **first function** creates the string that contains the **parameters** from "?" to the end.

The **second function** adds the URL of the API and makes the call. When the data has been received calls the error or success callback.









The second API can be call in the app with the following functions

```
searchInformationForEachFeature(response) {
   for (var category in response) {
       if (response[category].features.length != 0) {
           ParkingSearcher.responseLength = response[category].features.length;
           ParkingSearcher.temporaryResponse = {
               "Results": {
                   "features": [],
                   "fullCount": ParkingSearcher.responseLength,
                    "type": "FeatureCollection",
           Loading.showAutoSearchLoading():
           for (var i = 0; i < response[category].features.length; i++)</pre>
               var serviceQuery = QueryManager.createServiceQuery(response[category].features[i].properties.serviceUri, "app");
               APIClient.executeQueryWithoutAlert(serviceQuery,
                   ParkingSearcher.mergeResults,
                   ParkingSearcher.decrementAndCheckRetrieved);
           SearchManager.startAutoSearch(ParkingSearcher.varName);
```

For each car park listed is called the API that returns details.

If there is **no car park** in the list is called a function which doubles the radius of the search area until at least one car park is in the list or the radius is greater than 200 km









• The number of free parking lots is copied **from realtime object in the properties** to make writing the template easier. Is also added as a property a string that identifies the **text color** based on the number of free parking lots

```
mergeResults: function (response)
    for (var category in response) {
       if (response[category].features != null) {
           if (response[category].features.length != 0) {
               if (response.realtime != null) {
                   if (response.realtime.results != null) {
                       if (response.realtime.results.bindings[0] != null) {
                           if (response realtime results hindings[Al freeDarkingLots | null) [
                               response[category].features[0].properties.freeParkingLots = response.realtime.results.bindings[0].freeParkingLots.value;
                               if (response[category].features[0].properties.freeParkingLots > 20) {
                                   response[category].features[0].properties.freeParkingLotsColor = "green";
                               } else if (response[category].features[0].properties.freeParkingLots > 0) {
                                   response[category].features[0].properties.freeParkingLotsColor = "orange";
                                   response[category].features[0].properties.freeParkingLotsColor = "red";
               ParkingSearcher.temporaryResponse["Results"].features.push(response[category].features[0]);
   ParkingSearcher.decrementAndCheckRetrieved();
decrementAndCheckRetrieved: function(){
                                                                         This function controls how many
   ParkingSearcher.responseLength--;
                                                                         calls have already returned the details
    if (ParkingSearcher.responseLength == 0) {
       ParkingSearcher.successQuery(ParkingSearcher.temporaryResponse)
       Loading.hideAutoSearchLoading();
                                                                         or returned error.
```









```
successQuery: function (response) {
   var responseObject = response;
   if (SearchManager.typeOfSearchCenter == "selectedServiceMarker") {
       MapManager.searchOnSelectedServiceMarker = true;
   for (var i = 0; i < responseObject["Results"].features.length; i++) {</pre>
       responseObject["Results"].features[i].id = i;
       Utility.enrichService(responseObject["Results"].features[i], i);
   if (responseObject["Results"].features[0].properties.distanceFromSearchCenter != null) {
       responseObject["Results"].features.sort(function (a, b) {
            return a.properties.distanceFromSearchCenter - b.properties.distanceFromSearchCenter
       });
   } else {
       responseObject["Results"].features.sort(function (a, b) {
            return a.properties.distanceFromGPS - b.properties.distanceFromGPS
       });
   ParkingSearcher.results = responseObject["Results"];
   ParkingSearcher.refreshMenu();
   ParkingSearcher.show();
   MapManager.addGeoJSONLayer(responseObject);
   ParkingSearcher.resetSearch();
```

This is the **function** that receives the **end**JSON and shows it to the user, by
creating the marker on the map and
populating the list through the
template.

The JSON is enriched with additional information such as distance from GPS or from a manual search and list is sorted according to these values.







• This is the **final template** that allows you to show the user a list of car parks in its vicinity with an **indication of the number of free parking lots**





ParkingSearcher in main menu

• Final version of the button with call to module logic

```
"callback": "PrincipalMenu.hide(); MapManager.centerMapOnGps()
                                                                 SearchManager.search('ParkingSearcher');
"iconId": "",
"iconClass": ""
"iconFontSize": "",
"iconColor": "",
"imgSrc": ""
"imgHeight": ""
"text": "LP".
"textFontSize": "38px",
"textColor": "#CC0000",
"captionId": "principalMenuParkingSearcher",
"captionTextId": "moduleParkingSearcher".
"step": "",
"stepId": "",
"ribbon": true,
"ribbonId": "",
"ribbonStyle": "background: #CC0000; background: linear-gradient(#FF6600 0%, #CC0000 100%);",
"ribbonText": "NEW",
"removed": false,
"index": 0
```

The search function of the variable SearchManager asks the user where want search (GPS, Manual or Last Service) and then call the search function of the variable which is passed as string

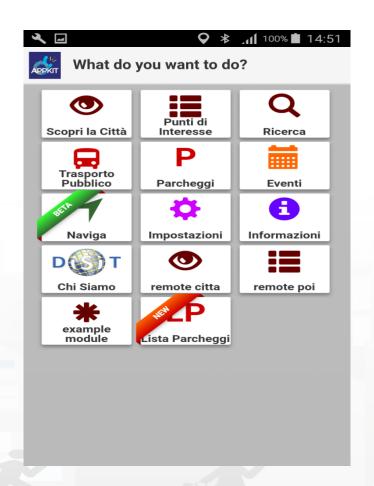
parkingSearcher.principalMenu.json







ParkingSearcher Module Finished













For the creation of the app through the modules it is necessary to compile it with grunt before doing it with Cordova



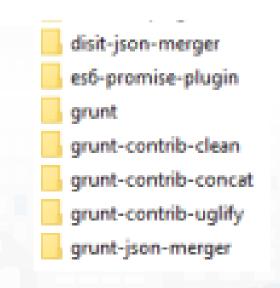
https://gruntjs.com





Gruntfile

Inside the root folder of the Sii Mobility App Kit there is a gruntfile.js which will merge the js and json files as described inside it. Inside the root there is also a node_modules directory within which there must necessarily be these folders containing the plugins useful for merging files.













Gruntfile

To compile the files, the grunt command must be launched on the open terminal in the project root If everything works properly the screen that should appear is as follows. If some package is missing it can be installed with the npm i packagename command. If disit-json-merger is missing download it from https://github.com/disit/siiMobilityAppKit/tree/master/node_modules/disit-

json-merger

```
nning "concat:dist" (concat) task
Running "concat:allTogether" (concat) task
tunning "clean:0" (clean) task
  1 path cleaned.
unning "disit-json-merger:singleTemplate" (disit-json-merger) task
ile "www/js/build/singleTemplate.json" created.
unning "json-merger:ita" (json-merger) task
ile "www/js/build/labels.ita.json" created.
dunning "json-merger:eng" (json-merger) task
ile "www/js/build/labels.eng.json" created.
Running "json-merger:deu" (json-merger) task
ile "www/js/build/labels.deu.json" created.
Running "json-merger:esp" (json-merger) task
File "www/|s/build/labels.esp.|son created.
Running "json-merger:fra" (json-merger) task
ile "www/js/build/labels.fra.json" created.
```





Gruntfile

If the grunt command was successful as in the image of the previous slide then the command Cordova build android can be launched and if all goes well you will have a screen like the following one





Further readings

- TC5.16. Exploiting Smart City API for developing Mobile and Web Apps
- TC5.15. Snap4City Smart City API Collection and overview, real time
- <u>TC5.17. Search on Services via Smart City API: MicroApplication,</u>
 <u>Exploiting Micro Applications in HTML5 based on Advanced Smart City API</u>
- TC5.18. Snap4City API are documented in Swagger, and tested in Postman
- TC5.19. Using ServiceMap as a Tools for Developing web and mobile apps and micro applications

SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT SNAP4INDUSTRY















What is missing here and you can find in the former course https://www.snap4city.org/577

- Data Streams from partecipatory, Mobile App
- Data streams from Mobile vehicles
 and smart phones Devices
- Data Ingestion via Web Scraping
- Data stream from TV Cameras, TV Cam Manager
- Social Media interoperability

- Another Complete Example
- BlockChain models and devices in Snap4City (new feature)
- **Orion Broker:**
 - Services/SrvPath and Multitenant
- External and Internal Brokers,
 - External Broker harvesting
- Managing Node-RED on edge from cloud
- More on: Security of Snap4City Stack from device to dashboards
- VM based installation of Snap4City
- ETL: Penthao Kettle interoperability

https://www.snap4city.org/577

You Tube

Video4

On Line Training Material (free of charge)



https://www.snap4city.org/944

none

	1st part	2nd part	3rd part	4th part	5th part	6th part	7th part	8th
what	Overview	Dashboards	IOT App, IOT Network	Data Analytics	Data Ingestion processes	System and Deploy Install	Smart City API: Web & Mob. App	Design and Develop Smart Solutions
PDF 2022	CONADION STATE OF STA	CSNAMAN S	CERTAIN AGE CONTRACTOR OF THE STATE OF THE	CINANON STATE OF THE PROPERTY	C SABOAM STATE OF THE SABO	COMADAIN STATE OF STA	C SMADA CO	CENADACE CONTROL SALE
Interactive (2022) with video and animations	C SHAP4OT SOME SOME SOME SOME SOME SOME SOME SOM	CONAMOR STATE OF THE PARTY OF T	C SHANAOTE William to a Shall be of the shall	C'BRANAGE STATE OF ST	C'SHAMOTE PROPERTY OF SHAME AND SHAM	C SHAP for Section to SHAP for the Shape of	C'SMAPACE COMMITTEE COMMIT	COSMANATOR STATE OF S
Videol	You Tube	You Tube	You		You	Tubo	You	You
Video2	You Tube	You	You	-	You	You	You	You
Video3	You	You	You	(TTT)	You	You	You	You Tube

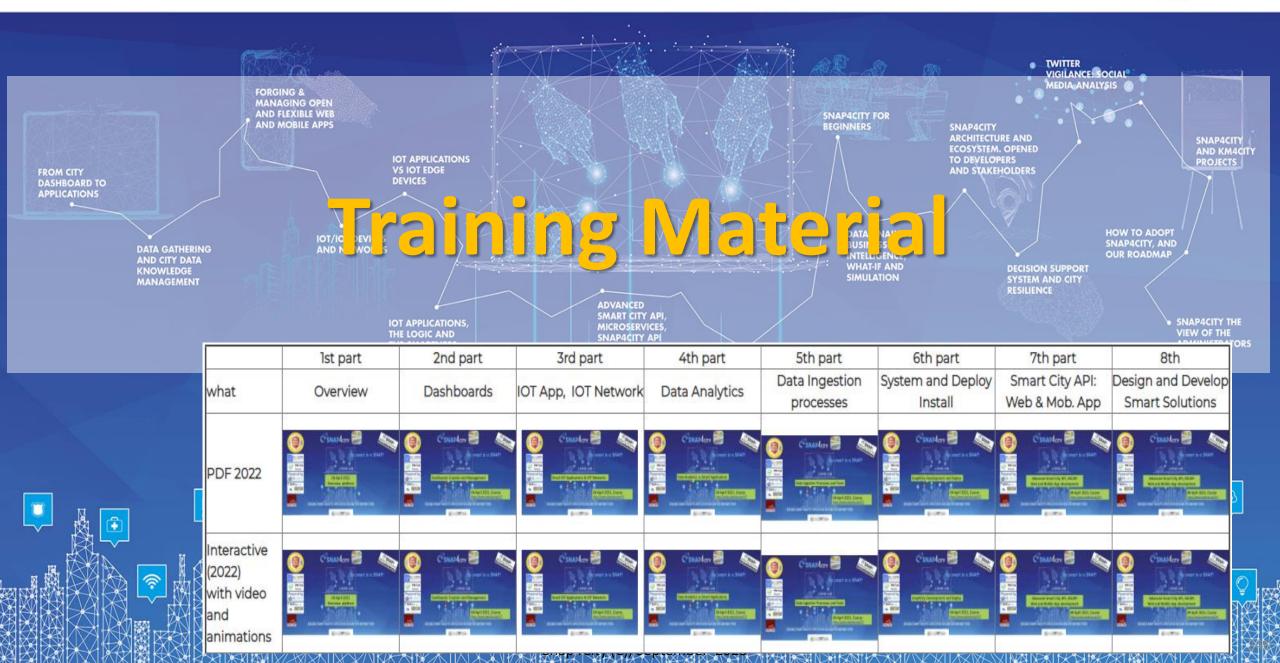
none

none

SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT SNAP4INDUSTRY











Note on Training Material

- Course 2023: https://www.snap4city.org/944
 - Introductionary course to Snap4City technology
- Course https://www.snap4city.org/577
 - Full training course with much more details on mechanisms and a wider set of cases/solutions of the Snap4City Technology
- Documentation includes a deeper round of details
 - Snap4City Platform Overview:
 - https://www.snap4city.org/drupal/sites/default/files/files/Snap4City-PlatformOverview.pdf
 - Development Life Cycle:
 - https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf
 - Client Side Business Logic:
 - https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf
- On line cases and documentation:
 - https://www.snap4city.org/108
 - https://www.snap4city.org/78
 - https://www.snap4city.org/426

Snap4City Snap4City

Switch To New Layout (Beta)

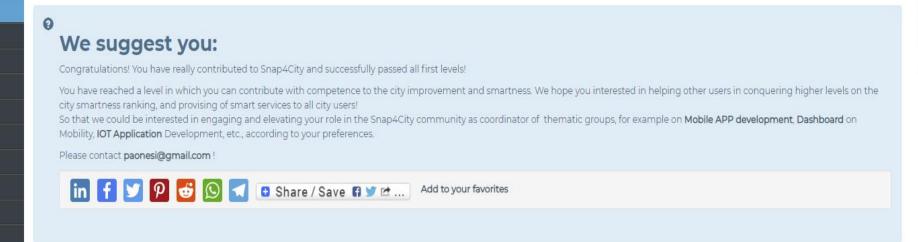
User: paolo.disit, Org: DISIT Role: AreaManager, Level: 3



- My Snap4City.org
- Tour Again
- www.snap4solutions.org
- Dashboards (Public)
- Dashboards of My Organization
- My Dashboards in My Organization
- My Data Dashboard Dev Kibana
- Data Management, HLT ▼
- Knowledge and Maps ▼
- Processing Logics / IOT App
- Resource Manager 🔻
- Development Tools ▼
- Management *
- Decision Support Systems
- Deploy and Installation ▼
- Help and Contacts -
- Documentation and Articles
- My Profile ▼
- Km4City portal
- DISIT Lab portal

Home / Tutorials and Videos / Welcome: how to start using Snap4City for beginners

Welcome: how to start using Snap4City for beginners





















0:00

Dashboards



Living Lab



API

Smart City API



Smart City

Ontology

Home How and Why To Use it ▼

DISIT

Developer

Groups

Operativo

Updates on Tools

Tutorials and Videos ▼

Training on Tools

and Platform

www.km4city.org

Sii-Mobility

Organization

Powered by

Tools ▼

Username: paolo.disit

Search

Search -Any-

Training Course Snap4City -2023 Edition new drupaladmin

Snap4City Newsletter of April 2023 new roottooladmin1







SCIENCE CLOUD

C'SNAP4city on





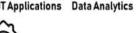








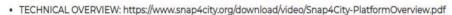












- Development Life Cycle: https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf
- Client-Side Business Logic Widget Manual: https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf
- Booklet Data Analytics, Snap4Solutions: https://www.snap4city.org/download/video/DPL_SNAP4SOLU.pdf

Please start a fully guided training cases:

Articles

- 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.

Dashboards (Public)

www.snap4solutions.org

Dashboards of My Organization

My Dashboards in My Organization

My Data Dashboard Dev Kibana

Extra Dashboard Widgets

Data Management, HLT

Knowledge and Maps

Processing Logics / IOT App

Entity Directory and Devices

Decision Support Systems

Deploy and Installation

Documentation and Articles

UNIVERSITÀ DINFO

Home How and Why To Use it ▼ Tools ▼

Tutorials and Videos ▼

HOW ARE YOU GOING TO BUILD THE FUTURE?

Snap4City: a framework for rapid implementation of Decision Support Systems and Smart Applications.



Home / Snap4City: Smart aNalytic APp builder for sentient Cities and IOT

Snap4City: Smart aNalytic APp builder for sentient Cities and IOT

You can't delete this newsletter because it has not been sent to all its subscribers.



Training on Tools

and Platform

Username: paolo.disit

Search





What People say Mobile Apps











API



Smart City

Ontology



Work with Us





Powered by www.km4city.org



Organization INDUSTRY 4.0 Groups

Living Lab Smart City API



₩ (

DISIT

"







IOT Devices IOT Applications Data Analytics Dashboards





- Development Life Cycle: https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf
- Client-Side Business Logic Widget Manual: https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf
- Realist Data Applytics Span (Salutions: https://www.span/city.org/download/video/DDI_SNAD/SOLUIndf

Developer

- Operativo
- I Indatas an

2022 booklets

Snap4City





https://www.snap4city.org/download/video/DPL_SN AP4CITY 2022-v02.pdf Snap4Industry





https://www.snap4city.org/download/video/DPL SNAP4INDUSTRY 2022-v03.pdf





Data Analytics



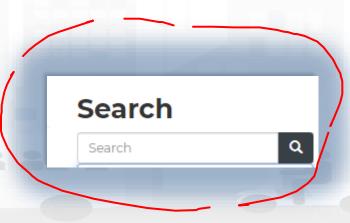


https://www.snap4city.o rg/download/video/DPL SNAP4SOLU.pdf





- Free Registration on Snap4City.org
 - Please select DISIT ORG to be sure to access at the examples
 - Most of the cities / tenant are private and they do not left much visible
- What you get is probably the 10% of what is on the platform ©
- Training: https://www.snap4city.org/577
- Scenarious: https://www.snap4city.org/4
- Publications: https://www.snap4city.org/426
- WEB pages: https://www.snap4city.org/78
- SEARCH on the right side























Snap4City Platform

Technical Overview

From: DINFO dept of University of Florence, with its

DISIT Lab, Https://www.disit.org with its Snap4City solution

Snap4City:

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

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

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



Tech. Overview

• https://www.snap4city. org/drupal/sites/default /files/files/Snap4City-PlatformOverview.pdf

















Development Life-Cycle

https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle-v1-1.pdf

From Snap4City:

- We suggest you to read the TECHNICAL OVERVIEW:
 - https://www.snap4city.org/download/video/Snap4City-
- https://www.snap4citv.org
- https://www.snap4industrv.org
- https://twitter.com/snap4city
- https://www.facebook.com/snap4city
- https://www.youtube.com/channel/UC3tAO09EbNba8f2-u4vandg

Coordinator: Paolo Nesi, Paolo.nesi@unifi.it

DISIT Lab, https://www.disit.org DINFO dept of University of Florence, Via S. Marta 3, 50139, Firenze, Italy Phone: +39-335-5668674







Development

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











Client Side Business Logic









Client-Side Business Logic Widget Manual

From Snap4City:

- We suggest you read https://www.snap4city.org/download/video/Snap4Tech- Development-Life-Cycle.pdf
- We suggest you read the TECHNICAL OVERVIEW
 - https://www.snap4city.org/download/video/Snap4City-

Coordinator: Paolo Nesi, Paolo.nesi@unifi.it

DISIT Lab, https://www.disit.org DINFO dept of University of Florence, Via S. Marta 3, 50139, Firenze, Italy







https://www.snap4city.org/d ownload/video/ClientSideBusi nessLogic-WidgetManual.pdf













SMART CITIES AND SMART INDUSTRY

Snap4City: FIWARE powered smart app builder for sentient cities







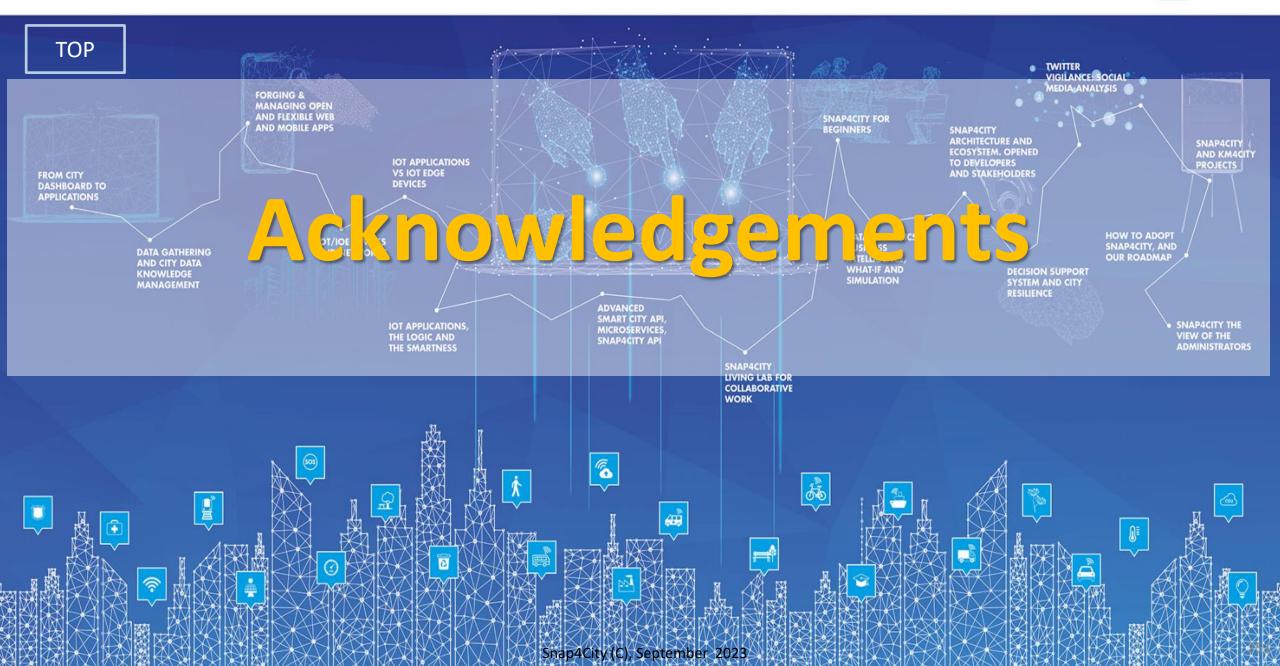
Commercial Overview SIWARE

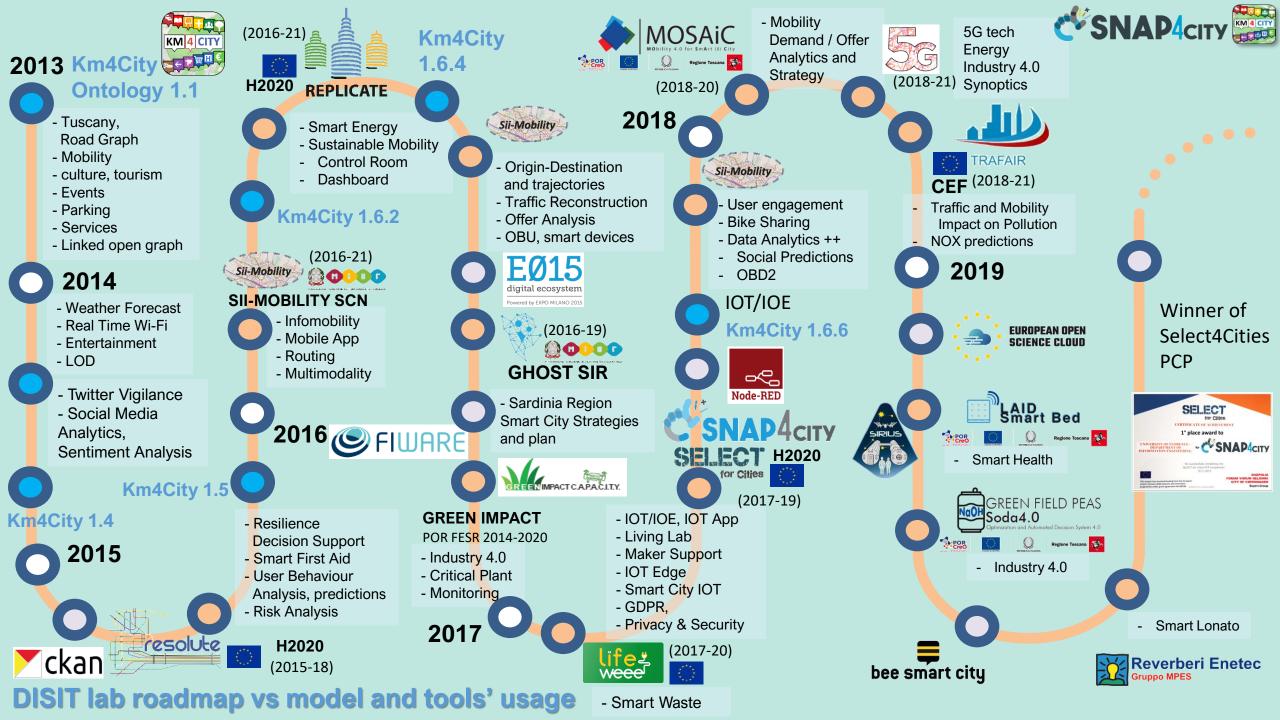


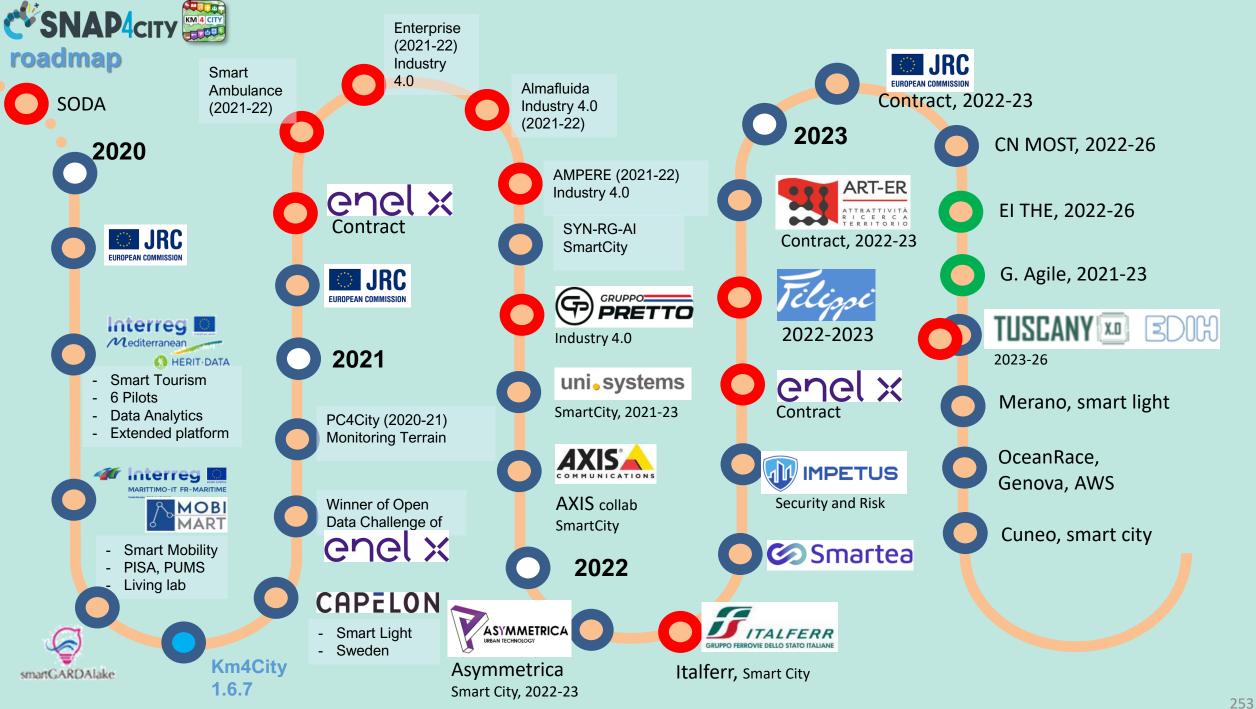
- https://fiwarefoundation.medium.com/snap4 city-fiware-powered-smart-appbuilder-for-sentient-citiesacfe24df49d5
- https://www.snap4city.org/drup al/sites/default/files/files/FF Im pactStories Snap4City.pdf

SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES









TOP













Office: +39-055-2758-515 / 517

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



