



UNIVERSITÀ
DEGLI STUDI
FIRENZE

DINFO
DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE

DISIT
DISTRIBUTED SYSTEMS
AND INTERNET
TECHNOLOGIES LAB



MINISTERO DELL'ISTRUZIONE DELL'UNIVERSITÀ E DELLA RICERCA



Smart City API: Sii-Mobility, Km4City

<http://www.sii-mobility.org>

Pierfrancesco Bellini, pierfrancesco.bellini@unifi.it



<http://www.Km4City.org>

Questions to be sent at km4city@disit.org



Agenda



- 9:00-10:30 <http://www.disit.org/6993>
 - Sii-Mobility Overview
 - General Model
 - Engaging Users via Apps
 - Overview of development tools for Apps
 - How to Legally work with SDK
 - Planned Additional Modules for the Apps
 - Info and Documents
- 10:30-11:30
 - ServiceMap usage <http://www.disit.org/6994>
 - Smart City API <http://www.disit.org/6995>
- 11:00-12:30 <http://www.disit.org/6992>
 - Sii-Mobility Mobile App Kit on GitHub
 - How to develop new module
- 12:30-13:30: lunch
- 13:30-17:00:
 - Workshop on development: esercitazioni



Smart City API



Basics

- REST APIs
 - Mainly available via HTTP GET method
 - Parameters provided in the query string of the URL
 - Production & Test
 - Production Example:
 - <http://servicemap.disit.org/WebAppGrafo/api/v1?selection=...&format=...>
 - Test Example:
 - <http://www.disit.org/ServiceMap/api/v1?selection=...&format=...>
 - The **format** parameter can be json or html (default json)
 - json → machine readable, CORS enabled
 - html → human readable, embeddable in iframes (generated by ServiceMap UI)
 - Currently no API key or authentication needed

Current available APIs (1/3)

Description	API	Save on ServiceMap	Embed	Where?
Service search near GPS position	Y	Y	Y	Prod & Test
Service search within a GPS area	Y	Y	Y	Prod & Test
Service search within a WKT described area	Y	N	Y	Prod & Test
Service search within a stored WKT described area	Y	Y	Y	Prod & Test
Service search by municipality	Y	Y	Y	Prod & Test
Service search by query id	Y		Y	Prod & Test
Full text search	Y	Y	Y	Prod & Test
Event search	Y	Not yet	Not yet	Prod & Test
Address and geometry search by GPS	Y	Not yet	Not yet	Prod & Test

Currently available APIs (2/3)

Service info	API	Save on ServiceMap	Embed	Where?
Generic Service	Y	Y	Y	Prod & Test
Event	Y	Y	Y	Prod & Test
Parking service	Y	Y	Y	Prod & Test
Traffic sensor	Y	Y	Y	Prod & Test
Weather Forecast	Y	Y	N	Prod & Test
Bus station	Y	Y	Y	Prod & Test
Fuel Station	Y	Y	Y	Test
First aid	Y	Y	Y	Test
Smart waste container	Y	Y	Y	Test
Smart bench	Y	Y	Y	Test
Smart irrigator	Y	Y	Y	Test
Energy meter	Not yet	Not yet	Not yet	-
Recharge station	Not yet	Not yet	Not yet	-
Smart street light	Not yet	Not yet	Not yet	-
Air quality monitoring station	Not yet	Not yet	Not yet	-

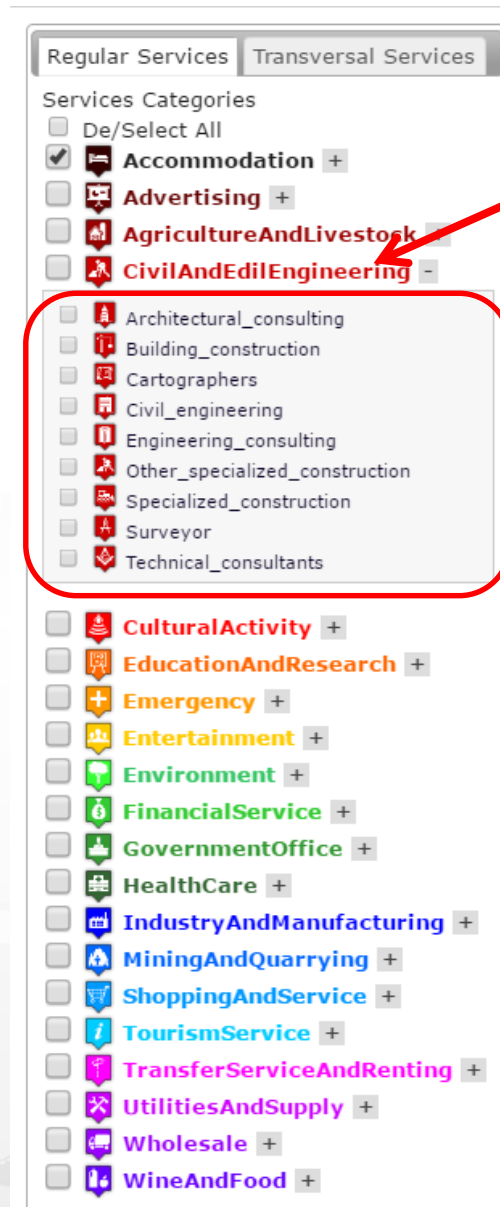
Currently available APIs (3/3)

Service info	API	Save on ServiceMap	Embed	Where
(Bus) Agency list	Y	N	N	Prod & Test
(Bus) Lines list	Y	N	N	Prod & Test
(Bus) Routes list	Y	Not yet	Not yet	Prod & Test
(Bus) Stop list	Y	Not yet	Not yet	Prod & Test
Search (Bus) Routes in a geographic area	Y	Not yet	Not yet	Prod & Test
Estimated Bus position	Y	Not yet	Y	Prod & Test
Rating and comment API	Y	N	N	Prod & Test
Service Photo API	Y	N	N	Prod & Test
Last contributions API	Y	N	N	Prod & Test
Recommender API	Y	N	N	Prod & Test
Image caching API	Y	N	Y	Prod & Test
Shortest path finder API	Not yet	Not yet	Not yet	-

Service search near GPS position

URL	http://servicemap.disit.org/WebAppGrafo/api/v1/
it allows to retrieve the set of services that are near a given GPS position. The services can be filtered as belonging to specific categories (e.g. Accomodation, Hotel, Restaurant etc), or having specific words in any textual field. It can also be used to find services that have a WKT spatial description that contains a specific GPS position.	
Parameters:	
selection	"<latitude>;<longitude>" of the GPS position
categories	the list of categories of the services to be retrieved, if omitted all kinds of services are returned. It can contain macro categories or categories, if a macro category is specified all categories in the macro category are used. The complete list of categories and macro categories can be retrieved on servicemap.disit.org .
text	words in this parameter are used to retrieve services that contain all these words in any textual description associated with the service.
maxDists	maximum distance from the GPS position of the services to be retrieved, expressed in Km (0.1 is used if parameter is missing) if it is equal to "inside" it searches for services with a WKT geometry that contains the specified GPS position (e.g a park)
maxResults	maximum number of results to be returned (if parameter is missing 100 is assumed), if it is 0 all results are returned.
lang	ISO 2 chars language code (e.g. "it", "en", "fr", "de") to be used for returned descriptions if available in multiple languages. Currently for languages other than "it" and "en" it returns "en" descriptions. (if parameter is missing "en" is assumed)
geometry	true/false, if true it returns a "hasGeometry" property for each service stating if the service has a complex WKT geometries (linestring, polygon) associated with it (if parameter is missing "false" is assumed)
uid	optional user identifier (see later)
format	html or json





20 Service Macro Class

Service Class

HTML Example

- Search for all Accommodations, bus stops, sensor sites or car parks within 200m from 43.7756;11.2490

[http://servicemap.disit.org/WebAppGrafo/api/v1/?](http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=43.7756;11.2490&categories=Accommodation;BusStop;SensorSite;Car_park&maxResults=0&maxDists=0.2&lang=it&format=html)

[selection=43.7756;11.2490&](#)

[categories=Accommodation;BusStop;SensorSite;Car_park&](#)

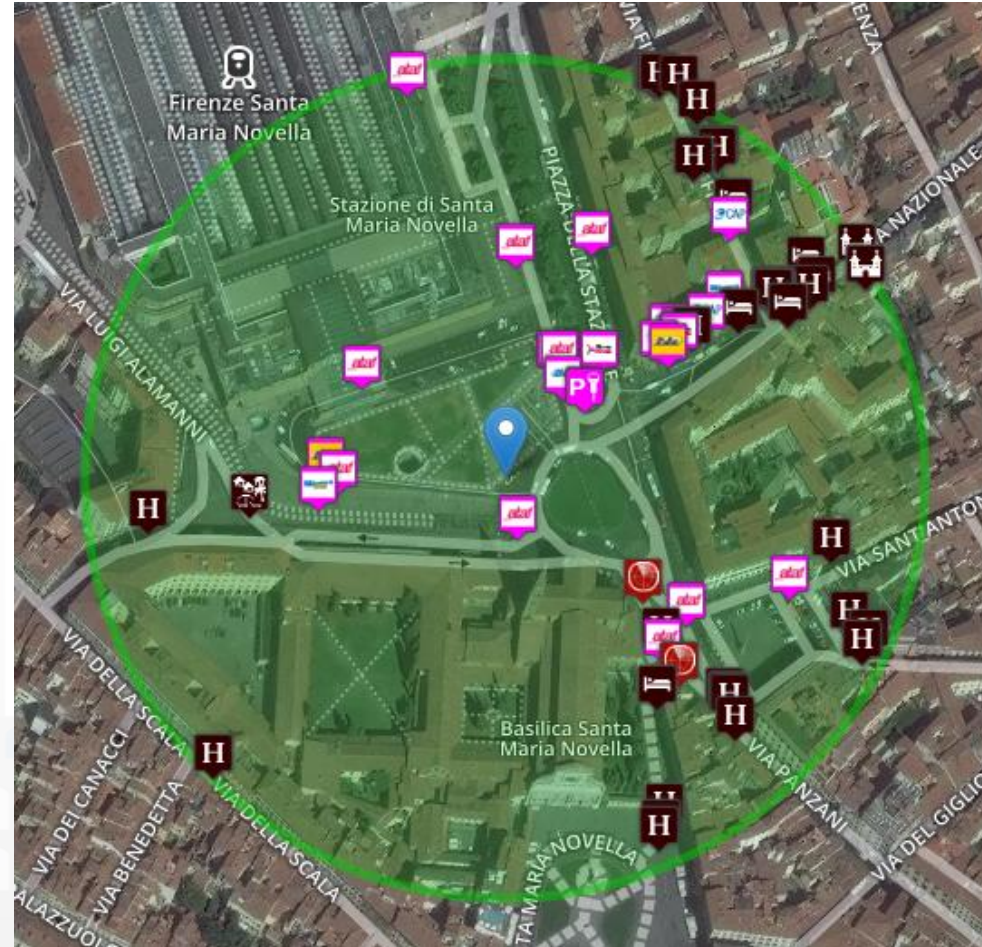
[maxResults=0&](#)

[maxDists=0.2&](#)

[lang=it&](#)

[format=html](#)

Result



JSON Example

- Search for all Accommodations, bus stops, sensor sites or car parks within 200m from 43.7756;11.2490

[http://servicemap.disit.org/WebAppGrafo/api/v1/?](http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=43.7756;11.2490&categories=Accommodation;BusStop;SensorSite;Car_park&maxResults=0&maxDists=0.2&lang=it&format=json)

[selection=43.7756;11.2490&](#)

[categories=Accommodation;BusStop;SensorSite;Car_park&](#)

[maxResults=0&](#)

[maxDists=0.2&](#)

[lang=it&](#)

[format=json](#)

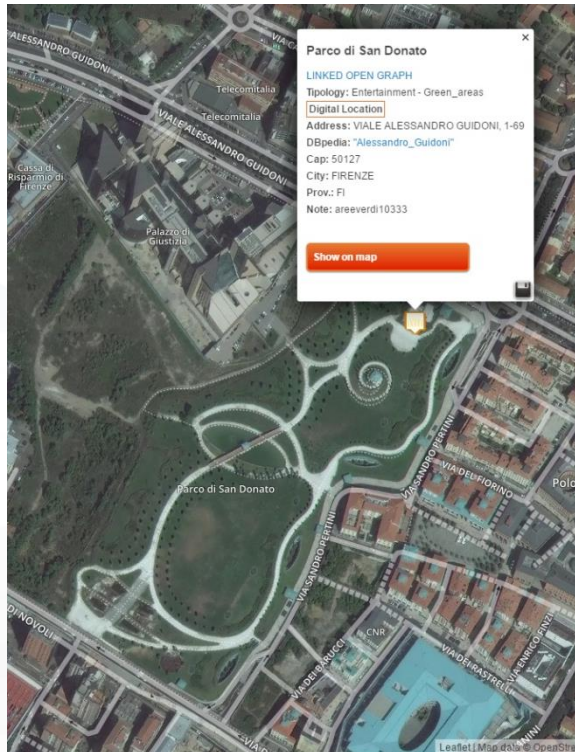
```
{
  "BusStops": {
    "fullCount": 26,
    "type": "FeatureCollection",
    "features": [{
      "geometry": {
        "type": "Point",
        "coordinates": [11.249078, 43.775326]
      },
      "type": "Feature",
      "properties": {
        "name": "Stazione Abside S.M.N.",
        "typeLabel": "Fermata",
        "tipo": "fermata",
        "serviceType": "TransferServiceAndRenting_BusStop",
        "busLines": "13 - 36 - 37",
        "serviceUri":
"http://www.disit.org/km4city/resource/Bus_ataflinea_Stop_FM0328_5",
        "agency": "Ataf&Linea",
        "agencyUri":
"http://www.disit.org/km4city/resource/Bus_ataflinea_Agency_172",
        "photoThumbs": []
      },
      "id": 1
    }, ... ]
  },
  "SensorSites": {
    "fullCount": 3,
    "type": "FeatureCollection",
    "features": [{
      "geometry": {
        "type": "Point",
        "coordinates": [11.24982, 43.77505]
      },
      "type": "Feature",
      "properties": {
        "name": "FI055ZTL00101",
        "tipo": "sensore",
        "typeLabel": "Sensore",
        "serviceType": "TransferServiceAndRenting_SensorSite",
        "serviceUri":
"http://www.disit.org/km4city/resource/FI055ZTL00101",
        "photoThumbs": []
      },
      "id": 1
    }, ... ]
  },
  "Services": {
    "fullCount": 84,
    "type": "FeatureCollection",
    "features": [{
      "geometry": {
        "type": "Point",
        "coordinates": [11.249473, 43.775867]
      },
      "type": "Feature",
      "properties": {
        "name": "Parcheggio Stazione Firenze S.M.N.",
        "tipo": "Parcheggio_auto",
        "typeLabel": "Parcheggio auto",
        "serviceType": "TransferServiceAndRenting_Car_park",
        "serviceUri":
"http://www.disit.org/km4city/resource/CarParkStazioneFirenzeS.M.N.",
        "multimedia": ""
      },
      "id": 1
    }, ... ]
  }
}
```

Example

- Any Entertainment service whose geometry contains a GPS position

<http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=43.7754;11.2494&categories=Entertainment&maxDists=inside&geometry=true&format=html>

format=html



format=json

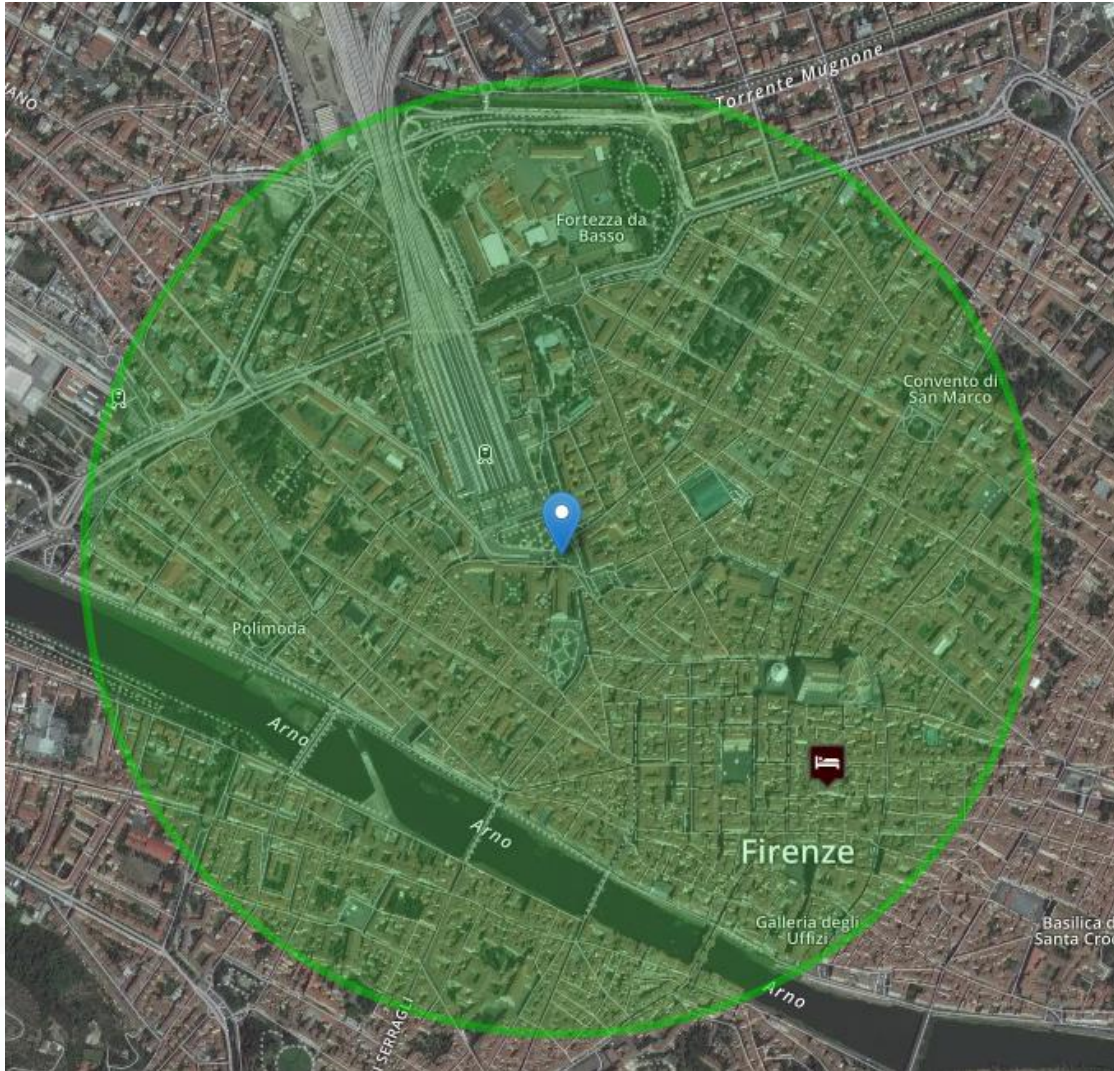
```
{
  "Services": {
    "fullCount": 3,
    "type": "FeatureCollection",
    "features": [{
      "geometry": {
        "type": "Point",
        "coordinates": [11.228672, 43.79453]
      },
      "type": "Feature",
      "properties": {
        "name": "Parco di San Donato",
        "tipo": "Green_areas",
        "typeLabel": "Green areas",
        "serviceType": "Entertainment_Green_areas",
        "hasGeometry": true,
        "serviceUri": "http://www.disit.org/km4city/resource/b92d4592bfb340d4ca7ea881b3334f61",
        "multimedia": ""
      },
      "id": 1
    }]
  }
}
```

Example

- **Accommodation within 1Km with “casa di dante” in a textual description**

<http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=43.7754;11.2494&maxDists=1&categories=Accommodation&text=casa%20di%20dante&lang=it&format=json>

format=html

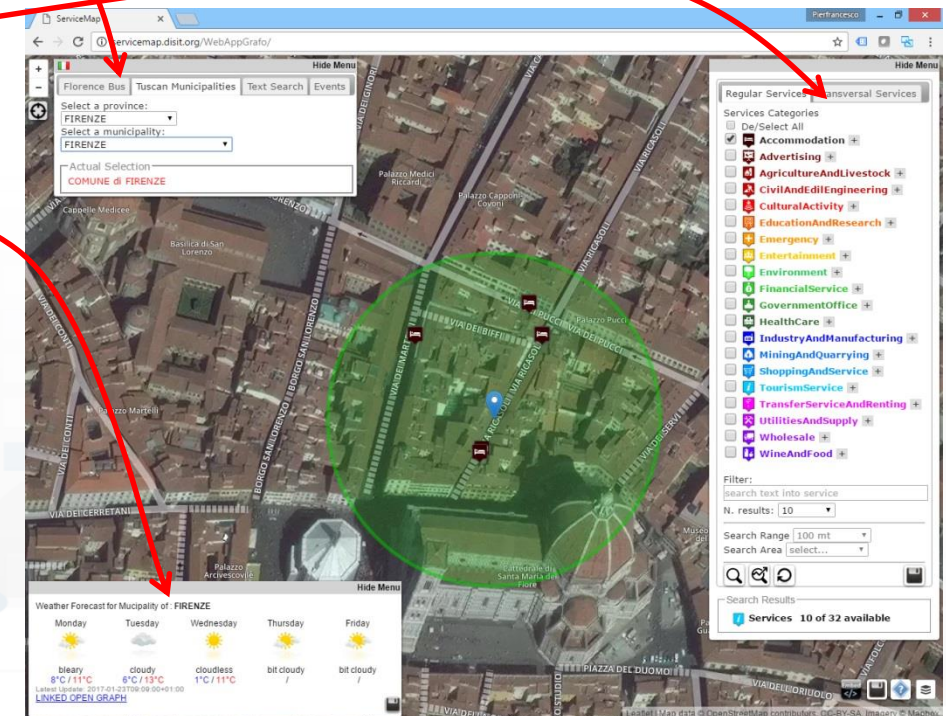


format=json

```
{
  "Services": {
    "fullCount": 2,
    "type": "FeatureCollection",
    "features": [{
      "geometry": {
        "type": "Point",
        "coordinates": [11.256365, 43.771023]
      },
      "type": "Feature",
      "properties": {
        "name": "CASA_DI_DANTE",
        "tipo": "Affittacamere",
        "typeLabel": "Affittacamere",
        "serviceType": "Accommodation_Boarding_house",
        "serviceUri":
          "http://www.disit.org/km4city/resource/c1cd4b12fabce2d9b3a1527fd5a7be79",
        "multimedia": ""
      },
      "id": 1
    }, ...]
  }
}
```

Note on format=html

- The following additional parameters are available:
 - map* : type of map to be used can be *satellite* (default), *streets*, *grayscale*.
 - controls* : apperance of control panels can be *hidden* or *false*, *collapsed*
 - info*: apperance of info panel can be *hidden* or *false*, *collapsed*



Note on uid parameter

- All requests can have an optional unique user identifier that is used by the recommender API to see which requests were made by the user in the past.
- Currently the *uid* is generated as a SHA256 hash of the device uuid generated by cordova device plugin
(see <https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-device/>)

Service search within a GPS area

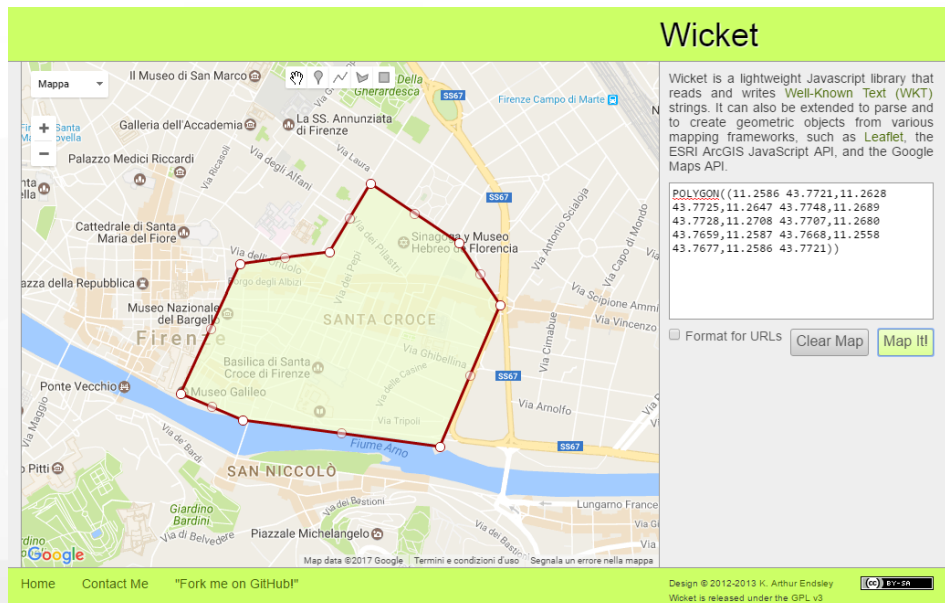
- Search within a rectangular area
- The same as the previous API but *selection* parameter is like:
 - "<lat1>;<long1>;<lat2>;<long2>"



- http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=43.7741;11.2453;43.7768;11.2515&categories=Accommodation;BusStop;SensorSite;Car_park&maxResults=10&lang=it&format=json

Service search within a WKT described area

- Search within a geographic area described using **Well-Known Text (WKT)** format (https://en.wikipedia.org/wiki/Well-known_text) with WGS84 coordinate system.
- to write a WKT string <https://arthur-e.github.io/Wicket/sandbox-gmaps3.html> can be used



Wicket

Wicket is a lightweight Javascript library that reads and writes Well-Known Text (WKT) strings. It can also be extended to parse and to create geometric objects from various mapping frameworks, such as Leaflet, the ESRI ArcGIS JavaScript API, and the Google Maps API.

```
POLYGON((11.2586 43.7721,11.2628 43.7725,11.2647 43.7748,11.2689 43.7728,11.2708 43.7707,11.2680 43.7659,11.2587 43.7668,11.2558 43.7677,11.2586 43.7721))
```

☐ Format for URLs

Home Contact Me "Fork me on GitHub!"

Design © 2012-2013 K. Arthur Endley
Wicket is released under the GPL v3

```
POLYGON((11.2586
43.7721,11.2628 43.7725,11.2647
43.7748,11.2689 43.7728,11.2708
43.7707,11.2680 43.7659,11.2587
43.7668,11.2558 43.7677,11.2586
43.7721))
```

Service search within a WKT described area

- The same as the previous APIs but *selection* parameter is like:
 - "wkt:<WKT string>"
- Example:
 - [http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=wkt:POLYGON\(\(11.2586%2043.7721,11.2628%2043.7725,11.2647%2043.7748,11.2689%2043.7728,11.2708%2043.7707,11.2680%2043.7659,11.2587%2043.7668,11.2558%2043.7677,11.2586%2043.7721\)\)&format=html](http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=wkt:POLYGON((11.2586%2043.7721,11.2628%2043.7725,11.2647%2043.7748,11.2689%2043.7728,11.2708%2043.7707,11.2680%2043.7659,11.2587%2043.7668,11.2558%2043.7677,11.2586%2043.7721))&format=html)

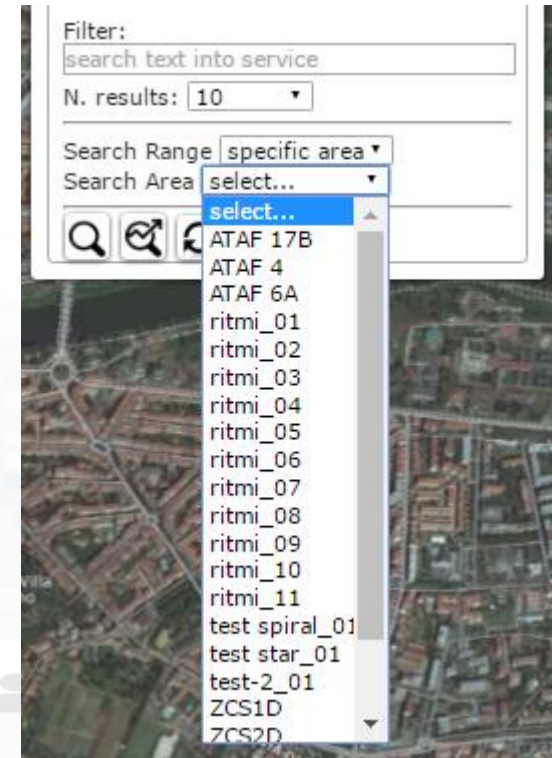


Service search within a stored WKT described area

- The same as previous API but the WKT geometry was uploaded using <http://www.km4city.org/wkt>
- The geometry can be used using its «label» in `selection=geo:<geometry label>`
- The labels available can be found on servicemap
- Example:

<http://servicemap.disit.org/WebAppGrafo/api/v1/?>

`selection=geo:test%20star_01&maxResults=0&format=html`



Service search by municipality

- Search for services of a specific municipality
- Similar to previous API but selection is the name of the municipality like FIRENZE, EMPOLI, PISA possibly with prefix "COMUNE di ".

- Example

<http://servicemap.disit.org/WebAppGrafo/api/v1/?selection=COMUNE%20di%20FIRENZE&categories=Entertainment&maxResults=100&lang=it&format=html>

Service search by query id

- it allows to retrieve the set of services associated with a query stored using the servicemap user interface and received by email.
- Parameters:
 - *queryId*: identifier of the query stored on servicemap
 - *lang*: language code (e.g. “it”, “en”, “fr”, “de”)
 - *geometry*: true/false, if true it returns a “hasGeometry” property for each service stating if the service has a complex WKT geometries (linestring, polygon) associated with it (if parameter is missing “false” is assumed)
 - *uid*: optional user identifier
 - *format*: json or html
- Example:
 - <http://servicemap.disit.org/WebAppGrafo/api/v1/?queryId=e02db54355fea40808300473c3537ff&format=json&lang=it>

Full text search

- it allows to retrieve the geolocated entities (not only services) that match with a list of keywords. The results can be possibly filtered to be within a specified distance from a GPS position, or within a rectangular area or inside a WKT geolocated area.
- Parameters:
 - *search*: the keywords separated with spaces that have to match with any textual description associated with an entity.
 - *selection*: optional the geographic area where to search (GPS point, GPS rectangle, WKT)
 - *maxDists*: optional maximum distance (used when selection is a GPS point)
 - *and maxResults, geometry, uid, format*

<http://servicemap.disit.org/WebAppGrafo/api/v1/?search=via%20nave&maxResults=10&lang=en&format=json>

```
{
  "fullCount": 558,
  "type": "FeatureCollection",
  "features": [{
    "geometry": {
      "type": "Point",
      "coordinates": [11.315443, 43.756367]
    },
    "type": "Feature",
    "properties": {
      "serviceUri":
"http://www.disit.org/km4city/resource/e96076db6e4e2b8b43fb660579eb4de8",
      "name": "PICCIOLI DANIELE",
      "tipo": "servizio",
      "photoThumbs": [],
      "multimedia": "",
      "civic": "",
      "serviceType": "CulturalActivity_Theatre",
      "typeLabel": "Theatre"
    }
  },
```

```
    "id": 1
  }, ... {
    "geometry": {
      "type": "Point",
      "coordinates": [10.898357, 43.729973]
    },
    "type": "Feature",
    "properties": {
      "serviceUri": "http://www.disit.org/km4city/resource/RT04801406596TO",
      "name": "VIA NAVE DI VITIANA",
      "tipo": "servizio",
      "photoThumbs": [],
      "multimedia": "",
      "civic": "1",
      "serviceType": "",
      "typeLabel": "Road"
    },
    "id": 8
  }, ... ]
```

Event search

- it allows to retrieve the geolocated events in a given temporal range (day, week or month). The results can be possibly filtered to be within a specified distance from a GPS position, or within a rectangular area or inside a WKT described geographic area.
- Parameters:
 - *range*: time range for the events to be retrieved, it can be 'day' for the events of the day of the request, 'week' for the events in the next 7 days or 'month' for the events in the next 30 days (if omitted 'day' is assumed).
 - *selection*: optional the geographic area where to search (GPS point, GPS rectangle, WKT)
 - *maxDists*: optional maximum distance (used when selection is a GPS point)
 - *and maxResults, uid, format (only json by now)*

<http://servicemap.disit.org/WebAppGrafo/api/v1/events/?range=day&format=json>

```
{
  "Event": {
    "type": "FeatureCollection",
    "features": [{
      "geometry": { "type": "Point", "coordinates": [11.251058,
43.769848]},
      "type": "Feature",
      "properties": {
        "serviceUri":
"http://www.disit.org/km4city/resource/Event_18794_973b96efaf3f99f1b70
af19cda4e3bf4",
        "name": "Tra arte e moda",
        "tipo": "event",
        "place": "MUSEO SALVATORE FERRAGAMO ",
        "startDate": "2016-05-19",
        "startTime": "10.00 -19.30; chiuso 1/1, 01/05, 15/08 e 25/12",
        "endDate": "2017-04-07",
```

```
      "freeEvent": "NO",
      "address": "PIAZZA DI SANTA TRINITA",
      "civic": "2",
      "categoryIT": "Mostre",
      "price": "6 (incluso museo/including museum)",
      "phone": "055 3562466",
      "descriptionIT": "La mostra riflette il complesso rapporto fra arte e
moda prendendo spunto dalla storia di Salvatore Ferragamo che si ispirò
alle avanguardie artistiche del '900 per realizzare le sue creazioni. ",
      "website": "www.ferragamomuseo.com/museo",
      "serviceType": "Event"
    },
    "id": 1
  }, ... ]
}
```

Address and geometry search by GPS

- it allows to retrieve the complete address (municipality, street and civic number) given the GPS position. It may also provide a list of services or public transport lines intersecting with the provided GPS position.
- Parameters:
 - *position*: “<lat>;<long>” with GPS position
 - *intersectGeom*: true or false (assumed false if missing), if true it reports all the services and public transportation lines that have a geometry intersecting with the provided GPS position.
 - *uid*: optional user identifier
 - *format*: only json

Without geometry intersection

<http://servicemap.disit.org/WebAppGrafo/api/v1/location/?position=43.7741;11.2505&format=json>

```
{
  "address": "VIA PANZANI",
  "municipality": "FIRENZE",
  "number": "17\A",
  "addressUri":
    "http://www.disit.org/km4city/resource/RT048017023351CV",
  "municipalityUri":
    "http://www.disit.org/km4city/resource/048017"
}
```

With geometry intersection

<http://servicemap.disit.org/WebAppGrafo/api/v1/location/?position=43.7741;11.2505&intersectGeom=true&format=json>

```
{
  ... Same as the other plus ...,
  "intersect": [{
    "distance": 1.2392468323025842E-4,
    "name": "Firenze Card",
    "class": "http://www.disit.org/km4city/schema#Tourist_trail",
    "type": "LineString",
    "uri":
      "http://www.disit.org/km4city/resource/2a93692aa1eb7d680d9b4e0da668b408"
  }, {
    "distance": 3.1448272583131523E-4,
    "routeType": "Bus",
    "direction": "Salviatino",
    "name": "11",
    "agency": "Ataf&Linea",
    "class": "http://vocab.gtfs.org/terms#Route",
    "type": "LineString",
    "uri": "http://www.disit.org/km4city/resource/Bus_ataflinea_Trip_1923_3337883"
  }, ... ]
}
```

Service info

- it allows to retrieve information about a service using its serviceUri. It can return an html representation (format="html") or a machine readable representation (format="json")
- Parameters:
 - *serviceUri*: the serviceUri of the service
 - *lang*: language code (e.g. "it", "en", "fr", "de")
 - *realtime*: true or false (if omitted true is implied) indicates if the last value of the time varying properties should be provided in the result or not.
 - *uid*: optional user identifier
 - *format*: html or json
- The properties returned depend on the type of service

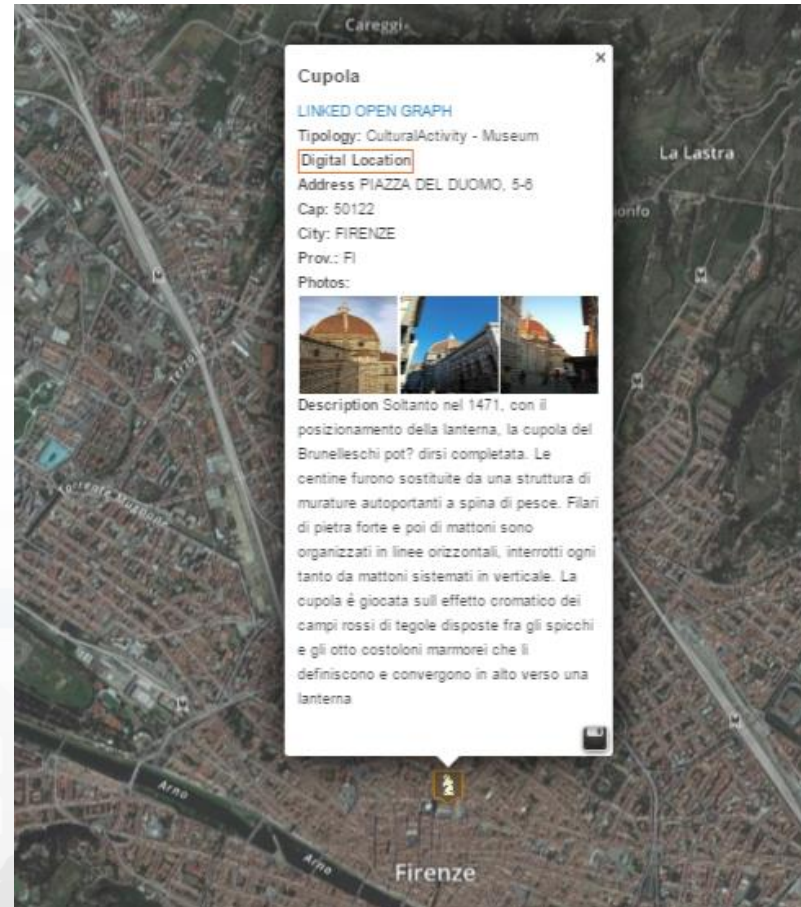
Example General Service

<http://servicemap.disit.org/WebAppGrafo/api/v1/?serviceUri=http://www.disit.org/km4city/resource/45e458d0c9e5bf53e34e7e2fed73a6b4>

```
{
  "Service": {
    "type": "FeatureCollection",
    "features": [
      {
        "geometry": {"type": "Point", "coordinates": [11.256423, 43.773254]},
        "type": "Feature",
        "properties": {
          "name": "Cupola",
          "typeLabel": "Museum",
          "serviceType": "CulturalActivity_Museum",
          "phone": "",
          "fax": "",
          "website": "",
          "province": "FI",
          "city": "FIRENZE",
          "cap": "50122",
          "email": "",
          "linkDBpedia": [],
          "note": "",
          "description": "Soltanto nel 1471, ...",
          "description2": "The Brunelleschi s Cupola ...",
          "multimedia":
"http://www.florenceheritage.it/mobileApp/immagini/aebTornabuoni/35.jpg",
```

```
          "serviceUri":
"http://www.disit.org/km4city/resource/45e458d0c9e5bf53e34e7e2fed73a6b4",
          "address": "PIAZZA DEL DUOMO",
          "civic": "5-6",
          "wktGeometry": "POINT (11.256423 43.773254)",
          "photos": ["http://servicemap.disit.org/WebAppGrafo/api/v1/photo/file-
1558826286406815456.jpg", "http://servicemap.disit.org/WebAppGrafo/api/v1/photo/file-
538829262336420198.jpg"],
          "photoThumbs":
["http://servicemap.disit.org/WebAppGrafo/api/v1/photo/thumbs/file-
1558826286406815456.jpg",
"http://servicemap.disit.org/WebAppGrafo/api/v1/photo/thumbs/file-
538829262336420198.jpg"],
          "photoOrigins":
["http://servicemap.disit.org/WebAppGrafo/api/v1/photo/originals/file-
1558826286406815456.jpg",
"http://servicemap.disit.org/WebAppGrafo/api/v1/photo/originals/file-
538829262336420198.jpg"],
          "avgStars": 5.0,
          "starsCount": 2,
          "comments": []
        },
        "id": 1
      }
    ]
  }
}
```

Example General Service Html



Example Parking

```
{
  "Service": {
    "type": "FeatureCollection",
    "features": [
      {
        "geometry": {
          "type": "Point",
          "coordinates": [11.24947, 43.77587]
        },
        "type": "Feature",
        "properties": {
          "name": "Garage La Stazione Spa",
          "typeLabel": "Car park",
          "serviceType": "TransferServiceAndRenting_Car_park",
          "phone": "055284784",
          "fax": "", "website": "",
          "province": "FI",
          "city": "FIRENZE",
          "cap": "50123",
          "email": "",
          "linkDBpedia": [],
          "note": "", "description": "", "description2": "",
          "multimedia": "",
          "serviceUri": "http://www.disit.org/km4city/resource/RT04801702315PO",
          "address": "PIAZZA DELLA STAZIONE",
          "civic": "3A",
          "wktGeometry": "",
          "photos": [],
          "photoThumbs": [],
```

```
        "photoOrig": [],
        "avgStars": 0.0,
        "starsCount": 0,
        "comments": []
      },
      "id": 1
    ]
  },
  "realtime": {
    "head": {
      "parkingArea": ["Garage La Stazione Spa"],
      "vars": ["capacity", "freeParkingLots", "occupiedParkingLots", "occupancy",
        "updating"]
    },
    "results": {
      "bindings": [{
        "capacity": {"value": "617"},
        "freeParkingLots": {"value": "322"},
        "occupiedParkingLots": {"value": "579"},
        "occupancy": {"value": "0.0"},
        "status": {"value": "enoughSpacesAvailable"},
        "updating": {"value": "2017-01-18T14:25:00+01:00"}
      }]
    }
  }
}
```

Example Fuel station

http://www.disit.org/ServiceMap/api/v1/?serviceUri=http://www.disit.org/km4city/resource/Fuel_station_01a234db6235dd55448a5044d9d26a52

```
{
  "Service": {
    "type": "FeatureCollection",
    "features": [
      {
        "geometry": {
          "type": "Point",
          "coordinates": [11.279211, 43.78041]
        },
        "type": "Feature",
        "properties": {
          "serviceUri":
            "http://www.disit.org/km4city/resource/Fuel_station_01a234db6235dd55448a5044d9d26a52",
          "serviceType": "TransferServiceAndRenting_Fuel_station",
          "name": "PINI E SETTESOLDI SNC",
          "typeLabel": "Fuel station",
          "province": "FI",
          "city": "FIRENZE",
          "cap": "50131",
          "address": "VIALE DEI MILLE",
          "brand": "AgipEni",
          ...
        }
      }
    ]
  },
}
```

```
"realtime": {
  "head": {"vars": ["measuredTime", "fuel", "price", "currency", "self"]},
  "results": {
    "bindings": [{
      "measuredTime": {"value": "2017-01-13 16:01:52"},
      "fuel": {"value": "Benzina"},
      "price": {"value": "1.579"}, "currency": {"value": "EUR"},
      "self": {"value": "true"}
    }, {
      "measuredTime": {"value": "2017-01-13 16:01:52"},
      "fuel": {"value": "Blue Diesel"},
      "price": {"value": "1.539"}, "currency": {"value": "EUR"},
      "self": {"value": "true"}
    }, {
      "measuredTime": {"value": "2017-01-13 16:01:52"},
      "fuel": {"value": "Blue Super"},
      "price": {"value": "1.729"}, "currency": {"value": "EUR"},
      "self": {"value": "true"}
    }, {
      "measuredTime": {"value": "2017-01-13 16:01:52"},
      "fuel": {"value": "Gasolio"},
      "price": {"value": "1.439"}, "currency": {"value": "EUR"},
      "self": {"value": "true"}
    }
  ]
}
```

Many other

- **Bus stops** with timetable and next passings
- **Traffic sensors** with current traffic status
- **Weather forecasts** for municipality
- **First aids** with number of users per status and color code
- **Air quality stations** with information on air quality
- **Smart waste containers** with waste level
- **Smart benches** with number of seatings and passings, temperature, air quality, etc.
- **Smart irrigators** with information on soil and leaves
- **Smart street lights**
- **Energy meters**
- **etc.**

Image Caching API

- provides a cache for the given image url, it downloads the image, scales it to the thumbnail or medium size depending on the size requested and save it for future requests.
- Parameters:
 - imageUrl: url to the image
 - size: the size of the image to be produced (can be «thumb» or «medium» or a number between 1 and 2000 pixels)
- It provides the scaled image produced in the same format as the original to be fit in a square of *size x size*, if the url is not an image it redirects to the original url.

Problem with multimedia property

- The *multimedia* property provided by some APIs often contains a URL to an external multimedia file that is no more available. Since the Image Caching APIs was created because the original images were too large for mobiles, now using this cache is currently the only way to retrieve these images. Unfortunately the cache was realized only for images and thus pdf files and audio files are no more available.
- Example:
 - <http://servicemap.disit.org/WebAppGrafo/api/v1/imgcache?imageUrl=http://www.florenceheritage.it/mobileApp/immagini/zocchi/148.jpg&size=thumb>

Recommender API

- The API suggests a set of services near a user, the services are grouped depending on the user profile and it can suggest services on the basis of past user behavior (svd=true) or only by position.
- Example: <http://screcommender.km4city.org/SmartCityRecommender/?action=recommend&user=3043b85d23d6f4879e1765c2c2e431cbc71d393065af06b03486ba4a04642b5b&profile=student&language=en&latitude=43.7727&longitude=11.2532&distance=1&version=1>

Public transport API

- Note: in the API urls and parameter names it is often referred to «bus», it should be intended any public transport not only buses.
- **Agency list (only json)**
 - provide a list of the public transport agencies available
- **(Bus) Lines list**
 - provide a list of the public transport lines available for a given agency
- **(Bus) Routes list**
 - provide a list of the public transport routes available for a given line or passing by a specific stop
- **(Bus) Stop list**
 - provide a list of the public transport stops available for a given route.
- **Search (Bus) Routes in a geographic area**
 - provides a list of the public transport routes that have a stop in a specified geographic area
- **Estimated Bus position**
 - provides the estimated current position of buses (currently only for ATAF&Linea)

Feedback API

- Rating and comment API
 - accepts a star rating (1-5) and/or a comment on a specific service. Comments are not automatically associated with the service, a moderator has to validate the comment provided.
- Service Photo API
 - accepts in POST as a multipart form the serviceUri, the user id and a photo in jpeg or png format. The photo provided is not automatically associated with the serviceUri a moderator will check it and decide.
- Last contributions API
 - Reports a list of the last ratings, comments and photos provided by users

Shortest Path Finder API

- Under development
- will allow to find a path between two geographic positions or services using car, public transport, feet or bicycle



Linked Data & SPARQL access

- The resources are also accessible using the W3C Linked Data protocol
 - using "Accept: application/rdf+xml" in the HTTP header to retrieve a RDF/XML representation of the resource
- Data is accessible via SPARQL endpoint (virtuoso 7.2)
 - <http://servicemap.disit.org/WebAppGrafo/sparql>
 - <http://log.disit.org> can be used to navigate in the knowledge graph
 - http://log.disit.org/sparql_query_frontend/ can be used to play with SPARQL queries

- **Documentation Smart City API, version 1, January 2017**
 - <http://www.disit.org/6991> (document from Sii-Mobility)
- **App Kit development page:**
 - <http://www.disit.org/6977> (slides ready, and video to appear)
- **Ontology and Km4City Tools:**
 - [Http://www.km4city.org](http://www.km4city.org)
 - <http://www.disit.org/6506> Ontology and documentation
- **Sii-Mobility is Open Source on GitHub as DISIT lab:**
 - <https://github.com/disit>
 - <https://github.com/disit/siiMobilityAppKit> (mobile App kit)
- **Data Ingestion processes and tools, tutorial:**
 - <http://www.sii-mobility.org/index.php/documentazione/slide-e-altro>
 - <http://www.disit.org/6690>
- **Deliverables of Sii-Mobility:**
 - <http://www.sii-mobility.org/index.php/documentazione/deliverable>
- **FAQ of Sii-Mobility:**
 - <http://www.sii-mobility.org/index.php/il-progetto/faq>
- **Promotional Kit for «Toscana dove cosa App»**
 - <http://www.sii-mobility.org/index.php/documentazione/kit-promozionale-app-toscana>

Open Source

Km4City, Sii-Mobility, RESOLUTE, REPLICATE: smart city big data open source tools. Km4City is a knowledge base and a research line of DISIT lab mainly developed before the start of Sii-Mobility, RESOLUTE, REPLICATE projects. While it has been mainly improved by them. Those projects are complementar each other and almost all of them use and contribute the Km4City research line. [ServiceMap smart city knowledge base tool](#): smart city service tool (mainly developed for Sii-Mobility project) for accessing to km4city knowledge base, for service browsing and query, for [Smart City API](#) for mobile and for mobile development tool, <http://www.disit.org/km4city>



- [Km4City ontology model and files](#) are accessible from <http://www.disit.org/km4city> improved with the support of projects as Sii-mobility, REPLICATE and RESOLUTE



- [SCE, Smart City/Cloud Engine front end interface](#), SCE is part of [DISCES](#) a Distributed SCE Scheduler Tools (SCE: Smart City/Cloud Engine), a DISIT tool for smart environments. It is currently in use in [Km4City](#) tools, and in [ICARO Cloud project and service, see CLOUD page. Developed for ICARO, and then improved for Sii-Mobility, and used in many other projects](#)



- [SCE, Smart City/Cloud Engine backend](#), SCE is part of [DISCES](#) a Distributed SCE Scheduler Tools (SCE: Smart City/Cloud Engine), a DISIT tool for smart environments. It is currently in use in [Km4City](#) tools, and in [ICARO Cloud project and service, see CLOUD page. Developed for ICARO, and then improved for Sii-Mobility, and used in many other projects](#)



- [DIM-RIM](#): Data Ingestion Manager and RDF Indexing Manager, [WEB page on DISIT lab with user manuals, DIM and RIM](#) area used in [Km4City](#) and tools, [Sii-Mobility](#) smart city national SCN project, [RESOLUTE H2020](#)



- [Dashboard Builder and executor](#): a tool for creating dashboard for smart city. See Km4City example of dashboard as reported in <http://www.km4city.org/controlroomtools.html> for the documentation see <http://www.disit.org/6935> which is manual with examples regarding widgets. Developed for REPLICATE Project, and used in others as Sii-Mobility, RESOLUTE.



- [Sii-Mobility Dev Kit Mobile AppKm4city](#): Open Source version of the Sii-Mobility mobile and web app, open modular (the full version is operative and accessible on all stores as "Firenze dove cosa", or "Toscana dove cosa", you can install from <http://www.km4city.org/app>). Developed for Sii-Mobility, adopted for the training and development meeting of the 24 January 2017, and as a basis for the Hackathon of 7-8 April 2017 in Florence.

Acknowledgement

- Thanks to the European Commission for founding. All slides reporting logo of **RESOLUTE H2020** are representing tools and research founded by European Commission for the RESOLUTE project. **RESOLUTE** has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 653460).
- Thanks to the European Commission for founding. All slides reporting logo of **REPLICATE H2020** are representing tools and research founded by European Commission for the REPLICATE project. **REPLICATE** has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 691735).
- Thanks to the MIUR for co-founding and to the University of Florence and companies involved. All slides reporting logo of **Sii-Mobility** are representing tools and research founded by MIUR for the Sii-Mobility SCN MIUR project.
- **Km4City** is an open technology exploited by those projects and line of research of DISIT Lab. Some of the innovative solutions and research issues developed into the above mentioned projects are also compliant and contributing to the Km4City approach and thus are contributing to the open Km4City model of DISIT lab.

