DELIVERABLE

Project Acronym:	CitySDK
Grant Agreement number:	297220
Project Title:	Smart City Service Development Kit and its Application Pilots

D4.2 – Mobility, Lead and Replication Pilots reports, Smart Mobility SDK Components

Revision: 1.0

Authors:

Job Spierings (Waag Society, Amsterdam)

Bert Spaan (Waag Society, Amsterdam)

İlker Yilmam (IBB Istanbul)

George Koukas (Gnosis, Lamia)

Marcello Zini (Provincia di Roma)

Pekka Koponen (Forum Virium, Helsinki)

Adrian Slatcher (MDDA Manchester)

Julian Tait (Future Everything, Manchester)

Project co-funded by the European Commission within the ICT Policy Support Programme

Dissemination Level

P	Public	**X**
C	Confidential, only for members of the consortium and the Commission Services	

Revision history and statement of originality

Revision history

Revision	Date	Author	Organization	Description
0.1	20/12/2013	Job Spierings	Waag Society	
0.5	23/12/2013	Job Spierings	Waag Society	
0.6	13/01/2014	Pekka Koponen	Forum Virium	
0.7	03/02/2014	Julian Tait	Future Everything	
1.0	03/02/2014	Job Spierings	Waag Society	

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Table of Contents

1. Introduction

1.1. The CitySDK Mobility API

The City Service Development Kit (CitySDK) mobility API is a system which collects open data and makes these data easily applicable by unified, real-time distribution. CitySDK helps with freeing data and offers essential tools to develop digital services in the City. It ensures quick and easy adaptation to the ever-increasing technical capabilities, as well as societal needs.

The CitySDK mobility API makes data applicable in five steps.
It:

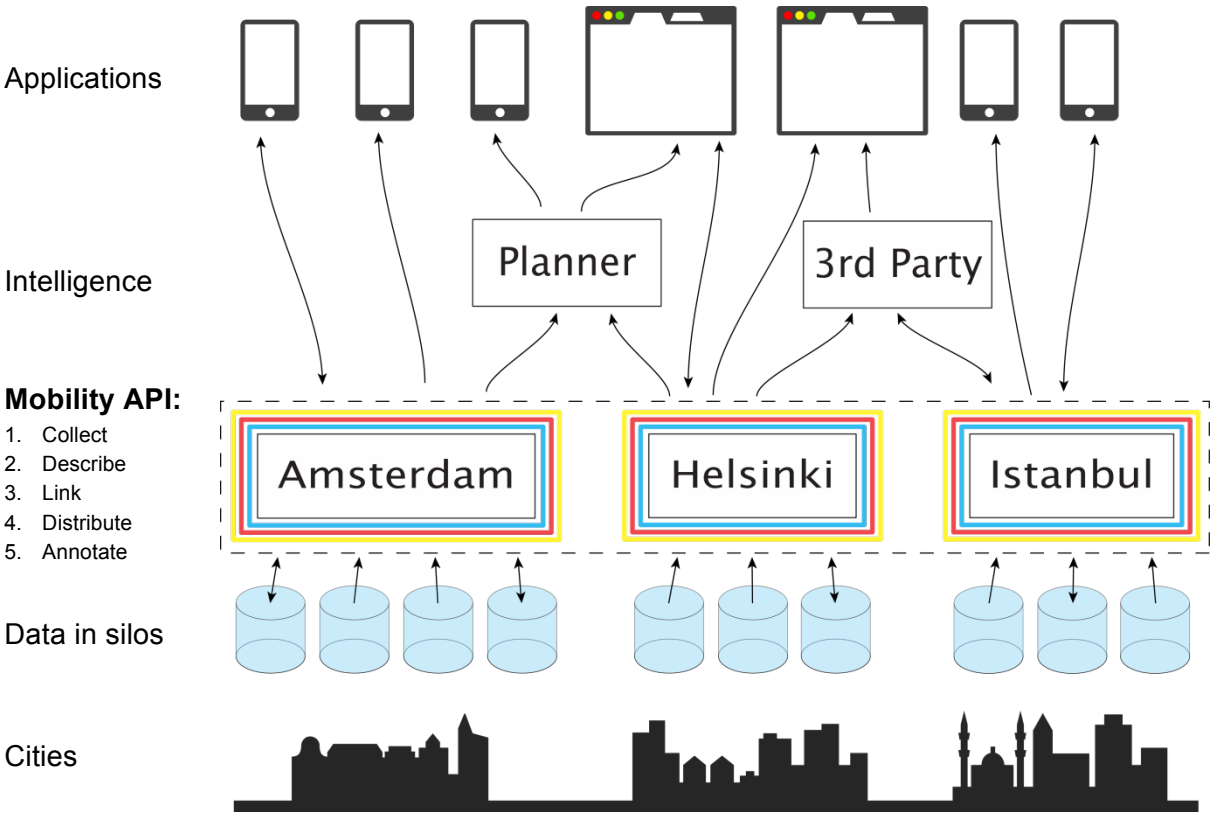
1. Collects data or web services with data owners.
2. Describes the data.
3. Links the data to relevant reference files (Cadastre, Open Street Map).
4. Distributes data via a unified web service (API).
5. Allows users to annotate data and/or objects (open 311).

Regardless of format, refresh rate or granularity, open data are available and applicable on demand for businesses, researchers and software developers. Thanks to the open design, the CitySDK mobility API can be applied to all national and international systems and is interoperable with European standards.

CitySDK is a perfect fit with the EU regulatory framework for the provision of open data. These regulations will be mandatory in 2015 for governments (see Directive 2003/98/EC).

Step-by-step implementation
Implementation is simple and always starts at the existing systems of the municipality. Thanks to phased implementation it allows users and providers to gradually gain experience with the system and open data in general. The mobility API is stable, robust and scalable, and creates a link with major reference databases from cadastre, open street map and others.

1.2. Architecture & key figures



1.2.1. Key figures CitySDK mobility API

- Five Star Linked Open Data
- 4 endpoints (FI, GR, NL, TR, UK), 1 discovery service
- More than 60 datasets, over 60 mln data points
- EU Partners: Amsterdam, Barcelona, Helsinki, Istanbul, Lamia (Gr), Lisbon, Manchester, Rome
- Dutch partners: Amsterdam DIVV/EZ, Amsterdam Smart City, Amsterdam Economic Board, Civity BV, Glimworm BV

1.2.2. Technical figures

- Unified REST API, data from different sources available per object
- Standardised interface in 6 cities
- JSON, RDF API in Ruby + Sinatra
- PostgreSQL/PostGIS database

1.2.3. Links

- EU project website: http://www.citysdk.eu
- Mobility API website: http://mobility.citysdk.eu
- Realtime overview available data in Amsterdam endpoint http://cms.citysdk.waag.org
- Demo applications: http://citysdk.waag.org/apps
- Discovery service: http://cat.citysdk.eu/endpoints

1.3. Replication process

A first beta version of the API was presented during Future Everything, Manchester, March 2013. A final stable version was presented to developers on 28 May 2013 in Amsterdam.

Replication partners then started research into technical requirements to implement the API and possibilities regarding replication of one of the available pilot applications. IBB Istanbul was the first one to have API and pilot application live, closely followed by the city of Lamia.

1.3.1. WP4 data and replications:

Underline – Local API enpoint implemented

(●) – Data in testing phase, expected to be live before M26

	Amsterdam	Netherlands	Helsinki	Istanbul	Lamia	Manchester	Rome
OpenStreet Map	●	●	●	●	●	●	●
Public Transport	●	●	●	●	●	●	(●)
Real-Time traffic flow	●	●					(●)
Population Statistics	●	●					
Cadastre	●	●	(●)				
Parking	●			●			
Electrical Charging	●	●					
POI	●	●	●	●	(●)	●	●

2. Lead Pilot: Amsterdam

2.1. Implementation activities

2.1.1. Implementation of the Mobility API

Focus was to get the API up and running and to get commitment from both the city and developers and SME's.

Getting access to new datasets was key for both purposes. But enabling the release of open data and working on use cases in mobility often means dealing with live operational systems for traffic management and traffic research. These systems, often run by specialist 3rd parties, require setting up extra, secure one way connections to open the data. Technically speaking this is important but not terribly complicated. But on an organizational level this creates a myriad of new issues. From engaging into new agreements with suppliers to needing a new position in the organization to manage the open data stream.

Currently the endpoint in Amsterdam offers access to over 60 datasets. To engage developers and SME to use the platform the addition of rich reference datasets has been very important. Especially the availability of (complex) Cadastre data for the whole country has been key to the increased uptake the API has at this point.

2.1.2. Mobility API development and improvement

2.1.3. Demonstration applications

The following demo apps have been made in order to demonstrate how the CitySDK platform works and to make its interoperability visible:

- Open Data Globe: This visualisation shows the dynamics of European cities on the basis of available real-time mobility data. (Not for smartphones or tablets.)
    - http://citysdk.waag.org/visualisation/
- Buildings: This map, based on BAG data, shows every building in the Netherlands, colour-coded by its year of construction.
    - http://citysdk.waag.org/buildings/
- A demonstration of queries and results in CitySDK: map viewer
    - http://citysdk.waag.org/map

- Now: This super-simple web app shows the public transport departure times in your immediate environment (real-time if available).
    - http://citysdk.waag.org/now

Third parties (SME's and one code fellow from Code for Europe) made the following apps:

- Tree Spotter - Amsterdam (iOS, Android)
    - https://play.google.com/store/apps/details?id=nl.twocoolmonkeys.opendata.bomenspotter.amsterdam&hl=en
    - https://itunes.apple.com/us/app/bomenspotter-utrecht-open/id657413587?mt=8
- Citymarker - read and write data (GitHub)
    - https://github.com/2coolmonkeys/citymarker
- Traze – game for visitors, tourists in Amsterdam.
    - video: www.vimeo.com/74889947
    - www.trazeapp.com/
- "Penalty Radar": shows speeds on Dutch highways, and calculates the fines that could have been collected by police.
    - http://boeteradar.2coolmonkeys.nl

2.1.4. Libraries for developers

A Ruby API Gem was developed, see https://rubygems.org/gems/citysdk.

2.1.5. Available data sets

UGC: User Generated Content
Source: Unless otherwise noted: http://www.amsterdamopendata.nl

category:	mobility.verkeer
description:	**Traffic Intensity, traffic flow, predicted traffic flow, traffic speed, vehicle category for main roads in the Netherlands ("A" and "N")**
source:	http://www.ndw.nu/pagina/nl/4/databank/31/actuele_verkeersgegevens/

category:	natural.trees
description:	**Trees in Nijmegen, Den Haag, Amsterdam, ROtterdam**
source:	Gemeenten

category:	administrative.dbpedia

description: DBpedia offers LOD access to extract structured information from Wikipedia
source: http://wiki.dbpedia.org/

category: environment.meldappdev
description: messages from meldapp (UGC)
source: 2CoolMonkeys

category: environment.energie_labels_amsterdam
description: Energy labels for houses in Amsterdam

category: mobility.roadmarkers
description: Hectometer markers on Dutch roads. Includes all A and N roads.
source: http://geoserv.nl/bag/Overig/NationaalWegenBestand.zip

category: commercial.energieverbruik
description: Energy use for natural gas and electricity
source: http://www.liander.nl/liander/campagne/open_data.htm

category: commercial.leegstandsdata
description: Vacant buildings (leegstandsdata) in Amsterdam, collected by Abram de Boer, phd candiate at TU Delft. (UGC)
source: http://www.linkedin.com/in/abramarchitect

category: security.p2000bag
description: p2000 alerts by emergency services, linked to official buildings/addresses
source: p2000

category: civic.service_requests
description: Service requests in the Amsterdam region. (UGC)
maintainer: citysdk@waag.org

category: civic.service_requests
description: Service requests city of Helsinki (UGC)
source: http://dev.hel.fi/apis/issuereporting

category: administrative.borders
description: Adminstrative Borders the Netherlands

source: Bron: © 2012, Centraal Bureau voor de Statistiek / Kadaster, Zwolle, 2012

category: tourism.hotels
description: Hotels in metro region of Amsterdam.
organization: Hotelloods Amsterdam
source: www.amsterdamopendata.nl/

category: tourism.food
description: Restaurants in the Amsterdam metro area.
source: ATCB/Amsterdam Marketing

category: environment.waste
description: Waste bins in central Amsterdam.

category: mobility.parking
description: General parking locations in the Amsterdam City Centre

category: mobility.parking
description: Parking locations in Crentral Amsterdam reserved for disabled.

category: mobility.parking
description: Parking reserved for disabled, specific vehicles. Central Amsterdam

category: mobility.parking
description: Locations of electric vehicle charge points in Central Amsterdam.

category: mobility.parking
description: Reserved parking for specific stakeholders, in Central Amsterdam.

category: cultural.events
description: Cultural event calendar.
source: http://dev.artsholland.com

category: administrative.houseboats
description: Houseboats from BAG, Dutch national building and address register.
source: BAG, Kadaster http://nlextract.nl

category: administrative.buildings

description: **Buildings from BAG, Dutch national building and address register.**

source: BAG, Kadaster http://nlextract.nl

category: administrative.caravans

description: **Semi-permanent caravan locations from BAG, Dutch national building and address register.**

source: BAG, Kadaster http://nlextract.nl

category: administrative.vbo

description: **Addresses with real-estate functional units, contained within a building: apartments, shops, houses, industrial spaces, etc.Data from BAG, Dutch national building and address register.**

source: BAG, Kadaster http://nlextract.nl

category: administrative.agriculture

description: **Basic Registration of Land Plots (BRP) is a national registry of Service Regulations (DR) of the Ministry of Economic Affairs. DR registers the use of crop land by agricultural entrepreneurs and the topography of the land. Per plot the BRP contains information about user, title, crop, surface modification date, manure number and geographical location.**

source: www.pdok.nl www.nationaalgeoregister.nl/ geodata.nationaalgeoregister.nl/

category: administrative.statistics

description: **Statistical Data per administrative region in the Netherlands.**

source: © 2012, Centraal Bureau voor de Statistiek / Kadaster, Zwolle, 2012

category: mobility.parking

description: **Bicycle parking locations in Amsterdam.**

category: mobility.parking

description: **Selected parking garages in Amsterdam**

category: mobility.parking

description: **Parking rates in Amsterdam.**

category: mobility.parking

description: **Parking rates Amsterdam, exceptions**

category: mobility.parking
description: **Location of parking ticket machines in Amsterdam.**

category: mobility.taxi
description: **Location of taxi queues**

category: mobility.traffic_flow
description: **Real-time traffic speed measurements on selected main roads in Amsterdam.**
source: http://www.trafficlink-online.nl/

category: mobility.public_transport
description: **Layer representing gtfs public transport information.**
source: Netherlands: OpenOV/GOVI import through gtfs.ovapi.nl
Manchester: http://store.datagm.org.uk/sets/tfgm/tfgmgtfs.zip
Tampere: http://files.itsfactory.fi/google_transit.zip
Helsinki: http://developer.reittiopas.fi/pages/en/other-apis.php

category: tourism.waitingtime
description: **Layer for waiting time data Van Gogh Museum**
source: HVA

category: tourism.poi
description: **Points of interest in Lisbon, PT.**
source: http://citysdk.ist.utl.pt/index.html

category: environment.risks
description: **Location of fireworks depots.**
source: risicokaart.nl

category: environment.risks
description: **Locations lpg depots and gas stations with LPG.**
source: risicokaart.nl

category: mobility.public_transport

description: **NS railway stations with live information on departures.**
source: http://www.ns.nl/api/api

category: administrative.statistics
description: **Statistics from Dienst Onderzoek en Statistiek, Amsterdam**
source: http://www.os.amsterdam.nl

category: mobility.charging_points
description: **Charging points for electric vehickles and -- bikes.**
source: oplaadpalen.nl

category: geography/base
description: **Base geograpy layer.**
source: Data from OpenstreetMap; openstreetmap.org © OpenStreetMap contributors

category: administrative.postcodes
description: **Postcodes NL, 4 digit.**
source: BAG, Kadaster http://nlextract.nl

category: administrative.postcodes
description: **Postcodes NL, 5-digit**
source: BAG, Kadaster http://nlextract.nl

category: administrative.postcodes
description: **Postcodes NL**
source: BAG, Kadaster http://nlextract.nl

category: natural.weather
description: **Chance of rain, on the city-quarter level.**
source: http://gratisweerdata.buienradar.nl/

category: cultural.heritage
description: **Rijksmonumenten: official Monuments**
source: http://api.rijksmonumenten.info/

category: mobility.comments
description: **anonimized crowd sourced data from social travel app (ReisRadar, UGC).**

source: reisradar.intelligence

category: natural.water
description: Water levels in the Netherlands.
source www.rijkswaterstaat.nl

category: natural.water
description: Water temperature in the Netherlands
source: Rijkswaterstaat

category: environment.measurements
description: **Smart Citizen Kits in Amsterdam see: http://www.smartcitizen.me and: http://www.kickstarter.com/projects/acrobotic/the-smart-citizen-kit-crowdsourced-environmental-m**
source: Waag Society

category: mobility.cycling
description: Old Fillarikanava data from 2009-2011 Helsinki
source: fillarikanava.fi

2.1.6. Support to Replication Pilots

Between March and December 2013 extensive support has been given to replication cities to support installation of API and replication apps. Local developers were supported via email and skype and in addition developers from Waag Society made site visits to Manchester (3) and Istanbul (2).

2.2. Dissemination and engagement activities

2.3. Impact

2.3.1. Outcomes from opening data

- New datasets have become available due to successful implementation and visualization of previous datasets.
- As a result of opening up Cadastre data various business are developing applications that enrich their data with Cadastre data.
- Dept. of Transport and Infrastructure in Amsterdam is starting a new program called Open Data FWD, which aims to use open data as well as user generated data to tackle specific

use cases in the city. Open Data FWD is joined by Utrecht, Den Haag and Rotterdam (G4) and the program will run for 4 Months with a total value of over €75k. Data will be offered via CitySDK API's. Use cases include:

- Research regarding Traffic Flow of bikes in the city with user generated gps locations instead of classic traffic research done by tallying traffic.
- Abandoned bicycles: involve crowd, SME's into new solutions for a costly issue in main Dutch cities (€4.2 mln a year on identifying, removing and storing abandoned bicycles).
- Bicycle parking: parking space for bicycles is lacking.
- Car parking: 50k kilometers a *day* are lost by drivers looking for a place to park their car. How can we lower this amount?

2.3.2. Stakeholders involvement in open innovation and service co-design

2.3.3. Project communication

To tell the CitySDK story, we needed an audience. We had some names, addresses due to hackathons and events organised earlier such as Apps for Amsterdam, Mobiles for Good Challenge and the like. But now our story was more abstract and complex.
We decided to divide the audience in the following target groups:

- Developers: which encompasses everybody with an interest in data or applications with above average knowledge of the workings of these. So it includes people like front-end developers, designers etc.
- Policy makers: everybody with an active interest in social innovation trough technical change. This can be an innovation officer with a city council, journalists, individuals etc.
- Key organisations: owners of specific data we needed, SME's, companies, departments that we wanted to make sure were aware of CitySDK. But also researchers and Linked Open Data specialists.

To reach these target groups we decided to set up events that targeted them specifically.

- @Waag Open Space: developers where invited to organise their own event at the Waag, e.g. the developer group Appsterdam @Waag Open Space, or Quantified Self @waag open space. etc. In total XX number of Waag Open Spaces were organised for specific groups of hard & software developers.
- Big Open Beautiful: During these bigger events focus was on what data can do for society, what it can mean in the future and how these data looks. Trough showing the visualisations

we were able to tell about the problems of making the visualisations and the need for an SDK at the city level.

- Key Organizations are contacted 1:1. This includes organisations as the OPen State Foundations, various universities, governmental departments and traffic companies and startups like TomTom. Most often we went to their offices to give a presentation.

We started with these activities early on - before there was anything to show. It was very helpful to have a visualisation early on (for Picnic 2012). The visualisation did nothing more than show the geolocation of 30 static datasets of Amsterdam. But it was good training for our team and helpful visual material for early dissemination.

What's the pitch?

Selling a solution for distribution is not easy. Most people, and especially high level decision makers care exclusively about the experience of end-users: 'what is the direct result for our citizens?' Although this is understandable for a number of reasons, real results are hardly ever achieved directly and therefore this approach inevitably leads to numerous ad hoc solutions, apps, widgets and websites that show one or two datasets with only limited possibilities for re-use, if at all.
An additional threshold was the amount of prior knowledge of the audience. It became quickly apparent that terms like interoperable, API, SDK and others needed not only to be explained but to interpreted by us in the right context in every discussion about CitySDK.

When the CitySDK platform went live, our visualisations were working and we got positive feedback from numerous developers we were able to be more and more specific about the qualities and benefits of CitySDK. This resulted in a 4 page brochure for the general public and a website for developers.

Results

After being live for 5 Months, 5 SME's are developing with CitySDK in the Netherlands. This means parties that are actively developing with CitySDK API that we know of. In addition we are discussing use of CitySDK for open data distribution with the 5 major cities. The CitySDK visualisations such as http://citysdk.waag.org/buildings/ had considerable impact on social media, are a good showcase of the calue of open data and general visibility of the project.

2.3.4. Engagement of other cities

The 4 main cities in The Netherlands (Amsterdam, Den Haag, Utrecht, Rotterdam) are either experimenting with or implementing CitySDK API's as part of their open data policy. Numerous

parties have expressed specific interest in using and promoting CitySDK standards to ensure interoperability.

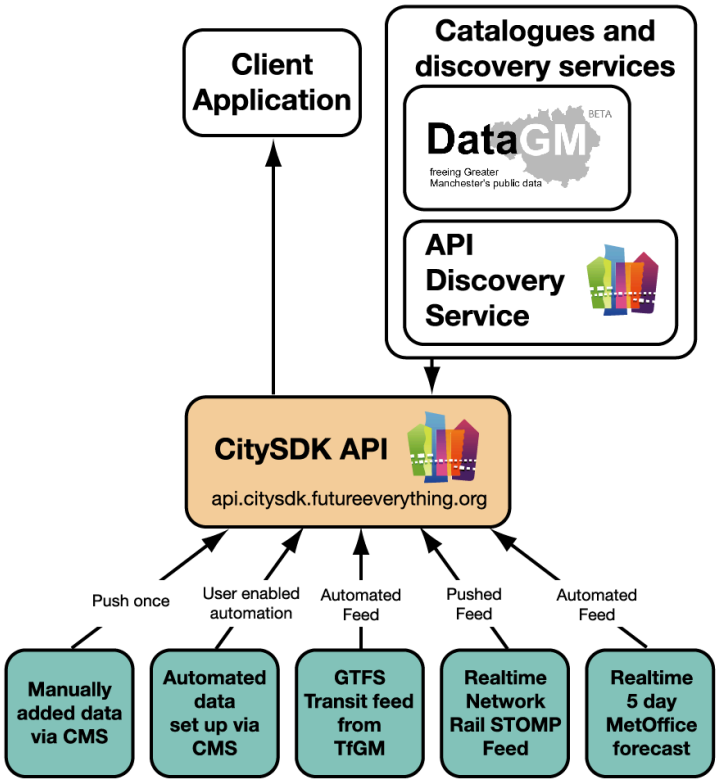
Contact info: http://waag.org/nl/project/smart-citysdk
End-point url: http://api.citysdk.waag.org

3. Replication Pilot: Manchester

3.1. Implementation activities

3.1.1. Implementation of the Mobility API

The implementation of the CitySDK API includes the sub domains of api.citysdk.futureeverything.org for the endpoint, dev.citysdk.futureeverything.org for information about the CitySDK API and cms.citysdk.futureeverything.org for an enhanced CMS functionality. The CitySDK endpoint is also accessible through DataGM – The Greater Manchester Datastore. The CitySDK components were set up on an Ubuntu 12.04 LTS server with SSL configured.



Manchester has implemented the API so that it can easily accommodate a number of data types. It was recognised at an early stage that tools would be needed that could generically parse CSV data into the City SDK API. Much of the static transport related data in the Greater Manchester region takes this form. The CMS has been augmented with a number of tools to enable this to be done easily. Along with the ability to import and parse CSV files manually, functionality has been built to enable automation of the process so that time intervals can be set by the data holder for the data to be automatically uploaded. The GTFS feed that is supplied by Transport for Greater Manchester is automatically uploaded to the API. The pushed Network Rail STOMP feed includes the real time

location of trains has been integrated but there is still need to improve the performance of this. Weather data is supplied to the CitySDK API via the API calling against the MetOffice JSON API. The endpoint is catalogued on the DataGM www.datagm.org.uk and is discoverable through the CitySDK API Discovery Service.

3.1.2. Data sets

Data sets currently available:

- GTFS Schedules and stopping points for all buses and trams within the Greater Manchester region supplied by TfGM
- OSM (Open Street Map) Data Layer
- Met Office 3 hourly weather report
- National Rail real time arrival and departure information

Datasets currently under development:

- Rail stations and Metrolink stop locations
- Bus routes – Map files
- Variable message signs positional information
- Traffic signal locations
- Traffic camera locations
- Road accidents in Greater Manchester
- Cycle locker locations
- Cycle routes
- Real time environmental sensor data

Datasets currently being sourced for the API:

- Real time arrival and departure information from Manchester Airport

3.1.3. Applications and services

A number of prototype applications were created out of the Routes to the Future Innovation Challenge that was run in March 2013. These were:

- City Navigator – Manchester http://dev.hsl.fi/fe2013
- Locus – Personal topographic travel mapping
- Go Board – Community mapping and transport http://198.199.68.108:9053/
- Pac-MANchester – Arcade version of public transport routes https://foxdogstudios.com/pac-manchester.html
- Manchester Realtime Scheduler
- Manchester Crowd Buses

- Who Runs That Bus? – Visualising the operators of Manchester's Buses http://hacks.bjwebb.co.uk/whorunsthatbus/
- BusBox – Timetable subscription service
- Mapchester – Point to Point browser based routing http://mapchester.azurewebsites.net.
- RutoApp – Multimodal trip planning
- Bus Whereabouts – Bus routing http://gentle-spire-6151.herokuapp.com/
- Bustle – Mass participation travel and feedback
- Travel Hub – Multi-modal travel application
- Busify – Augmented reality bus finder

City Navigator – Manchester, Go Board and Locus were given additional funding to bring their application to early stage prototype stage. This process is nearing completion.
For more detail on the applications created please see attached appendix.

Applications developed through release of data enabled by the CitySDK programme.

- Tramchester – Tram planner for Manchester utilising GTFS
- Augmentation of the Nokia Garmin product line with real time car park data

Visualisations

- OSM GTFS transport node comparison developed by Tuukka Hastrup (FI) converted to use in Manchester http://jsfiddle.net/eldog/n5mT5/embedded/result/

3.1.4. Feedback on the Mobility API

The installation of the API provided some challenges as there were some elements of the code base that were hard coded to the Amsterdam instance. During the Manchester implementation the hard coded elements were removed. The CMS features were initially found to be unstable and more work is being done to increase stability and functionality in order to improve the user experience of importing data. These improvements are being added to the CitySDK code base.

3.2. Dissemination and engagement activities

3.2.1. Engagement activities

Engagement activities have been targeted at both the supply and demand side of data.
On the demand side, CitySDK has been regularly featured at monthly Open Data Manchester meetings since project inception. In March 2013 an open workshop focusing on the CitySDK Mobility API was run as a prelude to Routes to the Future – An Innovation Challenge with the

express aim of encouraging developers to use the CitySDK mobility API to create applications and services. Routes to the Future was run in association with the Open Data Institute and Transport for Greater Manchester (TfGM). 76 developers attended with 12 CitySDK partner observers and over 20 applications created. In February 2014 there will be a Transportation Special event held with Open Data Manchester that will heavily feature the Manchester implementation of the Mobility API.
On the supply side a number of events have taken place where the API has been featured. A presentation was given to the Real Time Information Group – a consortium of regional transit agencies and service providers on the 17th January 2013. There has been ongoing engagement with TfGM throughout the lifespan of CitySDK. It has also been necessary to engage other city partners such as Manchester City Council, Visit Manchester, Marketing Manchester and Manchester Airport. There is an ongoing project to engage First Group to create an experiential travel application for their rail services.

3.2.2. Project dissemination

FutureEverything has taken part in a number of dissemination activities focusing on the CitySDK mobility API. Apart from the activities mentioned in the 4.2.1 Engagement section, a number of other dissemination activities have taken place. CitySDK has prominently featured on the FutureEverything website and in email marketing that FutureEverything regularly conducts. It as has also featured on the Open Data Manchester website and disseminated to the local open data community that contains 250 members. The Mobility API was highlighted at the FutureEverything conference in March 2013 and will also feature in March 2014. Presentations have been given to the Government of the Isle of Man and BCS Isle of Man on 23rd August 2013. The API was showcased by Waag and FutureEverything at a transport industry event at Manchester Town Hall called Making Data Work – Cities & Transport - How to reduce costs and enhance public services on the 18th June 2013, a Smart City event for the City of Namur on the 19th August 2013 and Afsnit I in Hørsholm, Denmark on the 29th November 2013.

Impact

3.2.3. Outcomes from opening data

FutureEverything has been advocating and delivering open data programmes through its Open Data Cities Laboratory since early 2009. The CitySDK programme has enabled a more focused conversation around open data with the regional transit agency. This has lead to a number of developments regarding transport open data within Greater Manchester. In preparation for the hackathon event, Routes to the Future – An Innovation Challenge TfGM started a process of engagement with the local open data community posting on community message boards asking for input into the types of data that they should be releasing and in what format. This lead to several

meetings where community members inputted into how certain real time data would be made available. In preparation for Routes to the Future, TfGM made available schedules and stop locations in GTFS format, bus route map files, real-time journey down road times, real time car park capacities and real time positional information for city centre shuttle services. Open Data was also used as part of TfGM's successful £32 million infrastructure bid to central government.
The data that was made available through CitySDK activity is now used in a number of services and applications including Nokia's Garmin product line.

3.2.4. Stakeholders involvement in open innovation and service co-design

Open Innovation and co-design have been at the core of the engagement with TfGM. It was essential that the activity we were undertaking had to be relevant and help meet the challenges that TfGM needed to address. Assurances had to be given that Routes to the Future had to create new ideas that could then be supported to create useable services rather than proof of concepts. The design of Routes to the Future created a mechanism that enabled developers to speak directly to TfGM staff and to understand their needs. A fund was made available so that the best ideas from Routes to the Future could be developed along with TfGM. Although this didn't work as smoothly as we hoped – due to a number of significant staff changes within TfGM. Three applications were selected for additional support.

Project communication

3.2.5. Engagement of other cities

Engagement with other cities has taken through activities during FutureEverything and other Smart City events. Through Routes to the Future, interest was shown by Newcastle City Council and their Tyne and Wear Passenger Transport Executive representatives. The CitySDK project was presented to the City of Namur and through working with the City of Eindhoven there is an active conversation between Eindhoven and Waag Society to create a CitySDK mobility API for Eindhoven.

3.3. Facts

Contact info: Julian Tait – julian@futureeverything.org, +44 7802 851 394
End-point url:

- api.citysdk.futureeverything.org
- cms.citysdk.futureeverything.org
- dev.citysdk.futureeverything.org

How long it took to implement the API?

Implementation of the API: approx 7 days

Implementation of API resources and components: approx 3WPM

Services and apps created using the API

The following services are still in development and are slated for completion in mid-March.

GO Board http://198.199.68.108:9053/

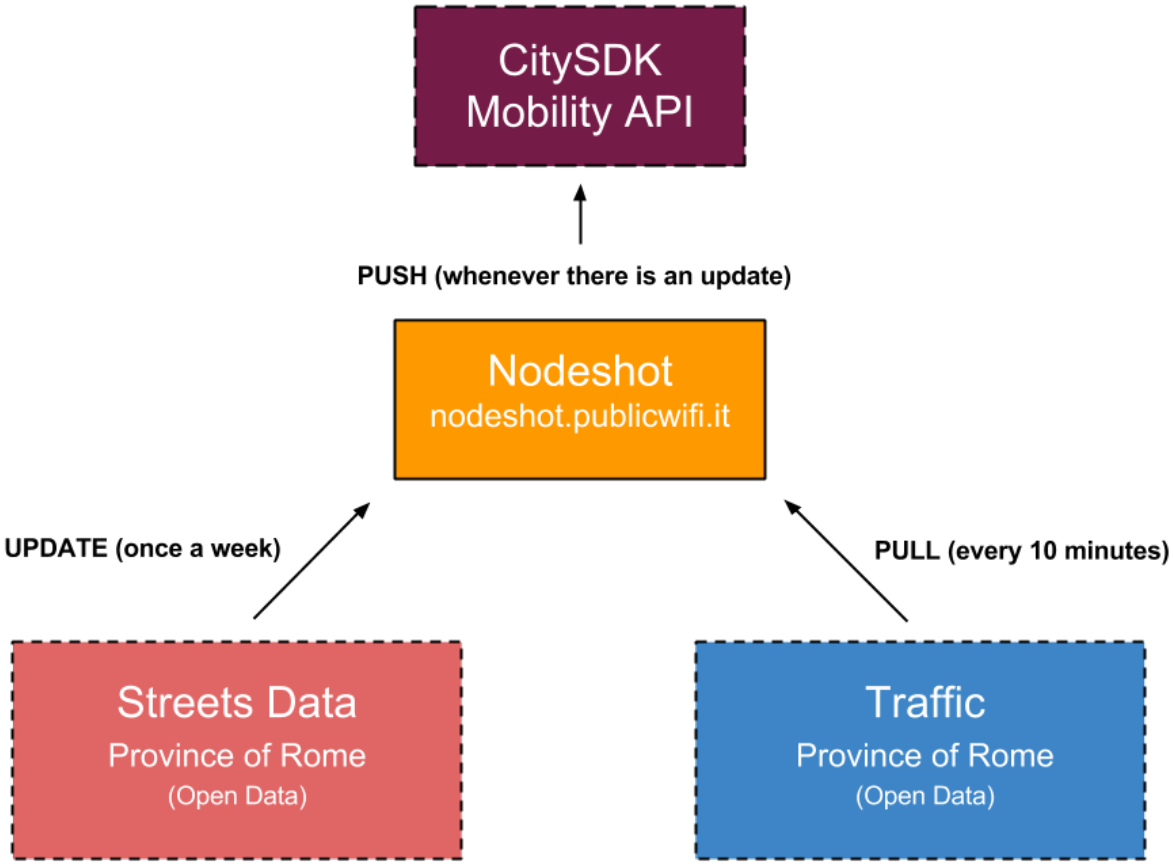
City Navigator Manchester http://dev.hsl.fi/fe2013

Code Libraries or other resources created: https://github.com/foxdog-studios/citysdk-deploy

4. Replication Pilot: Rome

4.1. Implementation activities

4.1.1. Implementation of the Mobility API



We have a central application which continuously refreshes the data in the CitySDK Mobility API.

This application, which we are using also for the others WPs, takes data from different sources and is in charge of sending only relevant updates to the CitySDK API.

4.1.2. Data sets

Data sets currently available:

- main streets of the Province of Rome
- traffic flow measurements

Data sets soon available:

- administrative borders of the Province of Rome

- public transport routes and info from the municipality of Rome

4.1.3. Applications and services

We implemented the map viewer provided by Waag and we are currently populating it: http://dev.citysdk-mobility.provincia.roma.it/map
We are also about to launch the "Now" application provided by waag: http://dev.citysdk.waag.org/now/

4.1.4. Feedback on the Mobility API

So far the feedback we can provide concerns the installation of the ruby server side applications; we think that making the installation procedure simpler would help getting other cities involved in CitySDK.

4.2. Dissemination and engagement activities

4.2.1. Engagement activities

The developer community has been engaged in formal events and informal sessions adapting the content to the interest of the target groups. Key points for getting their interest were:

- describing CitySDK development and strategy towards open innovation processes
- discussing the connection between CitySDK development and possible involvement in PA
- suggesting that knowing CitySDK development will facilitate job mobility within EU

4.2.2. Project dissemination

- Informal sessions have been held with the aim to involve the ninux.org community into CitySDK;
- 14.3.2012: Workshop "The Open Road. Opendata e Opensource: job opportunities and technical solutions" - thematic event, held in Rome, dedicated to developers, students and jobseekers;
- 17.4.2012: Rome Kick off event for Public Administration - event targeted to administrators and workers involved in PA;
- 5.2012: FORUM PA 2012 - Launch of the project targeted to LPAs across Italy, held in Rome, dedicated to administrators, PA workers and students;
- 7.6.2012: Workshop with Peter Corbett of Code for America on "Apps for Democracy and how to develop effective Apps Challenges" - held in Rome and dedicated to administrators, PA workers and students;
- 12.10.2012: Public presentation of Opendata CKAN portal of Provincia di Roma and Appcontest, held in Rome, dedicated to citizens, academic audience and policy makers;
- 18/19.1.2013: "Wherecamp Open meeting on geolocalization and digital maps'" - developer-oriented planned and carried out in Porta Futuro, Rome, by Provincia di Roma, Lynx and Caspur;

- 23.1.2013: “International Open Data Day” – Event aimed to support actions for the Agency for digital Italy; dissemination of CitySdk concept / participation pilot by Lynx, planned by Provincia di Roma, Lynx and Caspur/Cineca;
- 22.4.2013: “Smart Cities and apps: driving the change in Public Bodies” – Provincia di Roma issued an app contest related to its OpenData portal to engage developers in existing datasets and publish new ones. The event, planned also with Lynx and Caspur, focused on Smart Cities from the CitySdk perspective;
- 21.10.2013: "The 'Free ItaliaWiFi' federation conference meeting" - decision board meeting of 25+ italian PA representative, introduction to CitySDK as a "bridge" towards European top level good practices, held in Rome, hosted by Provincia di Roma

4.3. Impact

4.3.1. Outcomes from opening data

Open data introduction somehow represents:

- a push for institutional investments towards the creation of an additional layer of IT-driven interaction with citizens
- getting the local community feeling somehow progressively closer to the Institution
- enhanced cooperation among internal departments of Provincia di Roma in projects where sharing data is required
- stronger cooperation strategies within the “Rome Team” as a consequence of the actions taken to improve the quality of data and the creation of new services
- new cooperation strategies with the different levels of administration forming Italian LPAs infrastructure

4.3.2. Stakeholders involvement in open innovation and service co-design

The development has been influenced by the datasets available when the project started and by those potentially available during the project lifetime – and beyond. The efforts made by creating existing data and acquiring new ones came, in turn, from requests made by end-users.

4.3.3. Project communication

We noticed interest shown regarding CitySDK and Smart Cities related informative web pages published by Provincia di Roma, Caspur/Cineca and Lynx, from citizens, developers PA decision makers and operators in the field of communication. The idea (and the hope) is that working on Open data driven European framework may also help making the local infrastructure more efficient when delivering services to citizens.

We are plannining, to make communication more effective and able to reach all target groups, to schedule communication events when our replication pilot applications will be nearly ready for launch, to show the added value given by the CitySdk framework

4.3.4. Engagement of other cities

Although not exploited by the mobility work package, we’ve been able to expand CitySDK to other cities by importing free wifi hotspot data into CitySDK. This first step may make it easier to involve other cities in other domains.

Following the recent October event, the “Free ItaliaWiFi federation conference meeting”, the 23+ italian LPAs have been shown the potential of CitySDK with the double aim to get:

- access to hotspots data to be brought into CitySDK;
- participating administrations to explore the advantages of CitySDK driven application development

Outcomes (as of today):

- formal access to the hotspot data of 14 cities/administrations
- 13 datasets already in CitySDK, more to be added
- dissemination of CitySDK to the 3 levels of LPAs (Region, Province, Municipality) on the whole italian territory

4.4. Facts

Contact info:

Marcello Zini, Provincia di Roma, m.zini@provincia.roma.it, +39 06 6766 7318
Federico Capoano, Cineca, f.capoano@cineca.it, +39 3664253802

End-point url: http://api.citysdk-mobility.provincia.roma.it/

Services and apps created using the API

Map viewer: http://dev.citysdk-mobility.provincia.roma.it/map

5. Replication Pilot: Lamia

5.1. Implementation activities

5.1.1. Implementation of the Mobility API

We tried to implement the CitySDK mobility API locally. Therefore followed the instructions that WAAG provided us for the mobility API platform. Actually these instructions were a document file with a few notes about installation procedure for the mobility API Platform and an email with server requirements.

We employed aLinux Server, Ubuntu 12.04 LTS 64-bit Server, with latest updates for the operating system. This server has a static IP address and opened ports 80 (http), 22 (ssh) and 443 (https). Also we installed the following software and general we followed the procedure:

- Install Postgresqldatabase v.9.2
- Install PostGIS
- Install Postgres extensions
- Install a program that converts OpenStreetMap data to postGIS-enabled PostgreSQL (osm2pgsql)
- Install Ruby, passenger and nginx (open source reverse proxy server for HTTP, HTTPS, SMTP, POP3, and IMAP protocols)
- Install Memcached
- Download API source code from Waag GitHub and run API Installation Procedure
- Download and install OSM planet files
- Install the 'server' part of the downloaded API to the following directory structure as per Capistrano:
    - /var/www/citysdk
    - /var/www/citysdk/shared
    - /var/www/citysdk/releases
    - /var/www/citysdk/current
- The /shared directory has some more entries than the default:
    - /shared/daemons -- we run some data importer daemons from here
    - /shared/periodic -- periodically run importer/updater scripts
    - /shared/importers -- all 'other' importers (gtfs)
- Download and Import OSM layer file for Greek territoryto our local postgres database
- Add DNS Records to assign the local server IP to following subdomains:
    - cms.citysdk.lamia-city.gr
    - api.citysdk.lamia-city.gr
    - dev.citysdk.lamia-city.gr

- services.citysdk.lamia-city.gr

- Convert our mobility data to GTFS format
- Managed to import some of our GTFS converted data (bus stops and some of bus lines)and municipality of Lamia bounds (shape file). These data can be viewed at:
  - http://dev.citysdk.lamia-city.gr/map#http://api.citysdk.lamia-city.gr/nodes?layer=gtfs
  - http://dev.citysdk.lamia-city.gr/map#http://api.citysdk.lamia-city.gr/nodes?layer=admr

5.1.2. Data sets

Lamia Pilot publishes the 3 categories of datasets:

- Data sets that have been exposed before CitySDK API implementation:
  - Urban Bus Stops
  - Urban Bus Lines
  - Urban Bus Arrivals
  - Urban Bus Timetable
  - Interurban Bus Stops
  - Interurban Bus Lines
  - Interurban Bus Arrivals
  - Interurban Bus Timetable
  - Urban Trip Planner
  - Car Parking
  - Taxi Stands
  - Motorcycle parking and bicycle routes
  - Ticket Machines
- Data sets that have been exposed after CitySDK API implementation:
  - Urban Bus Stops
  - Some of Urban Bus Lines
  - Lamia municipal bounds
- Data sets that we will aim to expose via CitySDK API:
  - The whole set of Urban Bus Lines
  - Interurban Bus Stops
  - Interurban Bus Lines
  - Car Parking
  - Taxi Stands
  - Motorcycle parking and bicycle routes
  - Ticket Machines

- ❖ Show real time traffic data with low, medium, high congestion (information from sensors)

5.1.3. Applications and services

NOW:

An application that consumes local mobility municipal endpoints was developed with many functionalities. The main goal of the application is to test data that have exposed through municipal web services and to give developers a challenge to develop their own applications. Still, it is fully functional taking advantage the use of all data.

User interface's language depends on device's settings. Currently supports Greek and English.

Main functionalities:

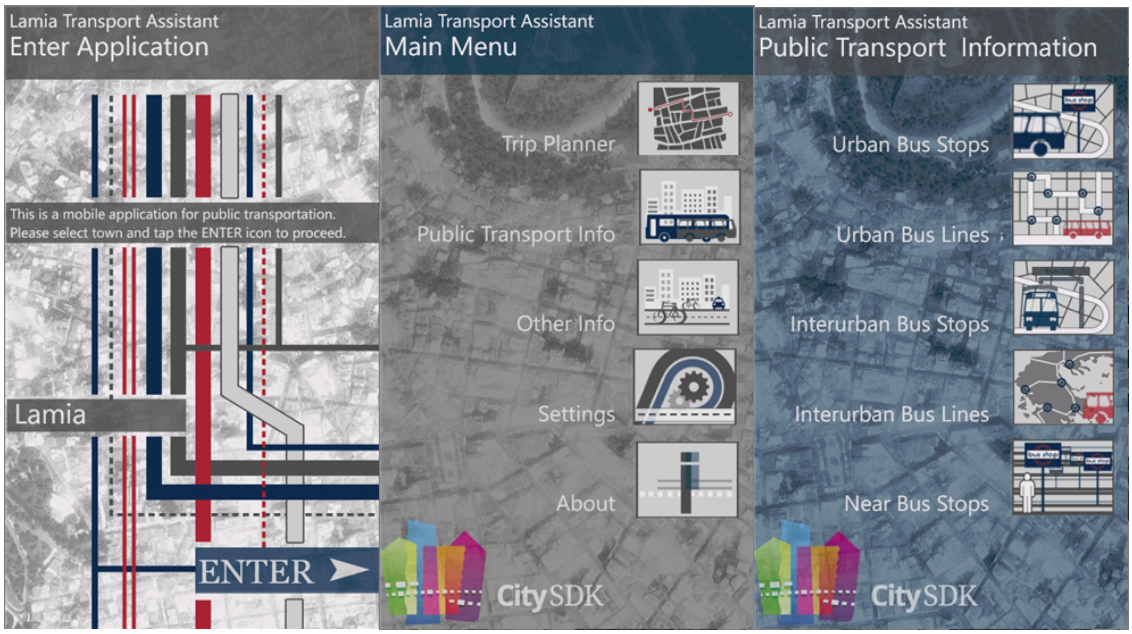
- Information for public transport
- Search and view public transport stops points on list or map
- Search and view public transport lines on list
- "Near Bus Stops" close from user's current location or any location on the map
- Browse map feature using Google maps V2 for Android app and MS Bing maps for Windows Phone app
- Schedule/timetable departures-arrivals lines
- Search and view lines passing through one public transport stop on list or map
- Search and view a line or route on a list
- Search and display line or routes on the map
- Information about taxi, parking cars, bicycles, automatic ticket vendors on list or map
- Nearest public transport stops from taxi, parking, etc.
- Search/View on the map of the route between two points (from-to: stop, user's current position, user's point choice on map) using Google navigation for car or on foot, for Android app.
- Trip planner designer between two points (from-to: stop, user's current position, user's point choice on map).

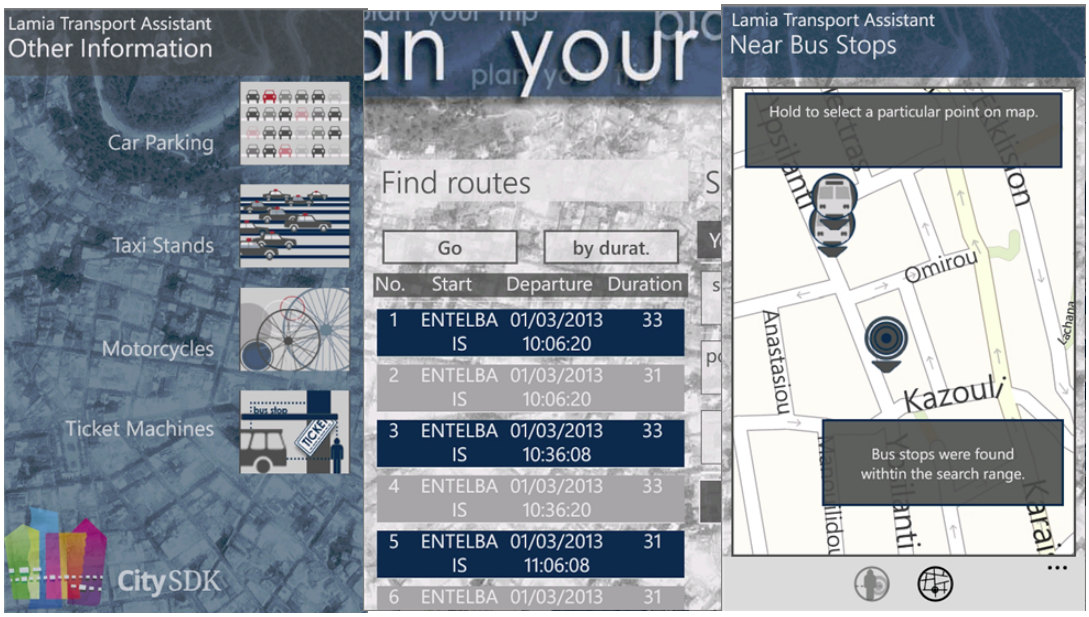
FUTURE:

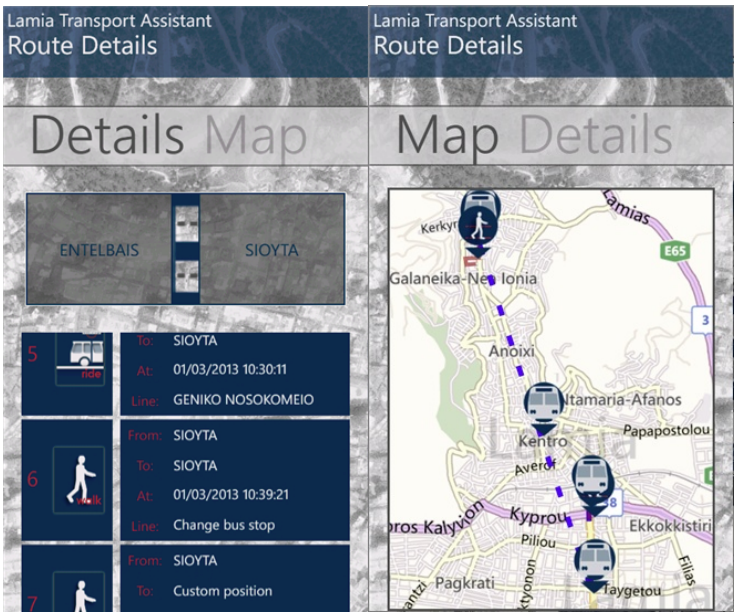
When the local CitySDK mobility API will be ready, our aim is to connect the application that described above to this API endpoint. First we have to test the API. Furthermore, endpoints

consumed by the API will be Amsterdam, Istanbul, Manchester, Helsinki, Rome and of course Lamia. So this application could be the same for all the above cities.

Below you can see some screenshots from the existing application:







Applications can be download from below links (Over 2100 downloads in 2 OS platforms):

- Windows Phone: http://www.windowsphone.com/en-us/store/app/transportassistant/c35d605f-abac-4c99-90df-dd71b2e2b567
- Google Play: https://play.google.com/store/apps/details?id=com.smarts.mobility&hl=en

5.1.4. Feedback on the Mobility API

Improvements for making Mobility API more useful:

- For Municipalities :
  - Collect and expose new datasets in a wider region
  - Intermunicipal approach to cover more Municipalities of the region
  - Analysis of a new business model for co-exploitation of the information with SMEs
- For Developers :
  - New datasets to be collected and exposed (e.g. real time data showing the congestion)
  - Central rewarding mechanism
  - Engagement activities
- For Users:
  - Gamification techniques
  - Crowd sourcing approach
  - Rewarding system
  - Selective redesign for ease of access to use of critical information and services
  - More mobile's OS platforms to be supported

5.2. Dissemination and engagement activities

5.2.1. Engagement activities

The Activities we carried out to engage developer communities and end-users into open innovation processes, are presented in the following table:

Table 7.2.1a Engagement activities of developers

Actual Dates	Location	Event	Main target groups – Participants
5-9-12	Lamia's 5th High School	Annual meeting of High School's teachers (with specialty in Computer Science)	Academic audience (70).Teachers and Professors with specialty in Computer Science to transfer knowledge to young students
19-9-12	University of Central Greece	1st Open Data Workshop for Academic audience.	
24-4-13		Meeting for the project's presentation and dissemination	
12-9-12	Technological Education Institution of Lamia		
24-10-12			Software engineers (10)Engineer's association advisory board.
18–26.5.13	PanHellenic Exhibition of Greece	Special Hall for the promotion of CitySDK activities	Wide Public (35.000 visitors)
3 – 6.6.13	City's events center	Pan-European week of environment	Ecologists, environmental groups, (Over 800 participants)

Our efforts for engagement were mostly to developers of our city. These are High School teachers with specialty in computer science and academic teachers and students from the local University of Central Greece and the Technological Education institution of Lamia.
Moreover a PanHellenic Exhibition is held every year in the city of Lamia. This exhibition usually has wide public participation, more than 35.000 visitors. This year there was a special kiosk for the promotion of CitySDK project.

Finally, few days ago a conference of the Hellenic Society for Computational Biology and Bioinformatics was organized in University of Central Greece, at the city of Lamia. We had a general

presentation for CitySDK project at an academic audience, but at the same time we tried to engage students into open innovation processes.

Some special events took place in the city of Lamia for the same scope, for engagement of developers and end-users. These were 2 athletic events had wide public participation, as we can see from the following table. During these events, two applications were developed by a team of developers, supported by GNOSIS, to help the organizing committee, runners, volunteers and audience that combined tourism and mobility data.

Table 7.2.1b Special events for developers engagement

Actual Dates	Location	Event	Main target groups – Participants
11-5-13	1st Lamia Night Run	CitySDK support for athletic event	Wide Public (Over 1200 participants. Over 25000 visitors of special web site)
22-9-13	7st Hercules Marathon Skyrunning		

5.2.2. Project dissemination

Dissemination activities are being presented in tables 7.2.1.b, 7.2.1a, 7.3.2., 7.3.3. and 7.3.4.

5.3. Impact

5.3.1. Outcomes from opening data

The impacts from the process of opening data in city of Lamia, that were already achieved, are:

- For Municipality:
    - Improvement of local governance through regeneration and more efficient and sustainable uses of the facilities and resources located in the city
    - Enhance communication and interactions between city's administrations, developers, businesses and citizens
    - Develop new public assets, including a city information infrastructure
    - Provide more effective links between economic, social and environmental planning within the city
    - Captured and shared experience and expertise in these areas from across the EU and worldwide

- For Developers:
    - Datasets that have never been collected and exposed, now have been done
    - Incentive for creating digital content and online enhancing services
    - Creation of an open digital framework for cooperation with city
- For Users:
    - Multichannel (web, mobile devices) access to Lamia's digital services through a single point
    - New more attractive and useful online services and apps
    - Improved quality of information for citizens
    - Ease of access to use of critical information and services
    - Provision of complex sets of information in a friendly way

5.3.2. Stakeholders involvement in open innovation and service co-design

Furthermore, we have done two (2) events for stakeholders' involvement in open innovation, as shown in the following table:

Table 7.3.2 Stakeholders involvement in open innovation and service co-design

Actual Dates	Location	Event	Main target groups – Participants
10-10-12	Mercantile Chamber of Fthiotida	Meeting for the project's presentation and dissemination	Commercial public (40).Representatives of tourism businesses and tour operators
11-3-13	Municipality of Lamia's Boardroom		

5.3.3. Project communication

Table 7.3.3 Project communication

Actual Dates	Location	Event	Main target groups – Participants
15-2-13	Lamia's Conference Centre	Official launch of Lamia's new webpage	Representatives of press, media, research institutions, commercial organizations, scientific bodies, academic
15-4-13		1st Wide event for the project's presentation	

		and result's dissemination as they have become available.	institutions, Town Council and wide public (120)

5.3.4. Engagement of other cities

Finally, we have participated to a Hall meeting of regional association of municipalities in Sterea Hellas and to TIDE project Workshop that took place at Frauhofer Institute, Stuttgart.

Table 7.3.4 Engagement of other cities

Actual Dates	Location	Event	Main target groups – Participants
15-11-12	Hall meeting of regional association of municipalities in Sterea Hellas	Meeting of regional association of Municipalities in Central Greece. Short presentation	Mayors of Municipalities in Region of Sterea Hellas (15)
12–13.11.13	Stutrgart Frauhofer institute	TIDE project Workshop	Participants from Wien, Madrid, Reading, Kalw, etc.

5.4. Facts

Contact info:

- Polizos Zois, municipality of Lamia, p.zois@lamia-city.gr, +302231351098
- Koukas George, Gnosis Computers, koukas@gnosis.gr, +302231020000

End-point url

Local Current endpoints:

- http://mobility-soapws.lamia.gr:8180/flashws_new/flashws.asmx
- http://mobility-restws.lamia.gr/mobilityservice/api/interurban

CitySDK mobility API endpoints:

- http://dev.citysdk.lamia-city.gr
- https://cms.citysdk.lamia-city.gr
- https://services.citysdk.lamia-city.gr
- https://api.citysdk.lamia-city.gr

How long it took to implement the API?

Municipality of Lamia 3 MM
Gnosis Computers 6 MM

Municipality of Lamia 15.000 Euros (without dissemination costs)
Gnosis Computers 25.800 Euros (without dissemination costs)

Services and apps created using the API

- Windows Phone: http://www.windowsphone.com/en-us/store/app/transportassistant/c35d605f-abac-4c99-90df-dd71b2e2b567
- Google Play: https://play.google.com/store/apps/details?id=com.smarts.mobility&hl=en

6. Replication Pilot: Istanbul

6.1. Implementation activities

6.1.1. Implementation of the Mobility API

Istanbul has configured a server as a CitySDK Mobility API server in August 2011. Setting up server process has been completed in 2 days. After Configuration Istanbul Metropolitan Municipality prepared open data sets in the domain of mobility in 1 month and imported to the API.
Istanbul's endpoint owns three domain names related to the project. These are:

- http://apicitysdk.ibb.gov.tr
- https://cmscitysdk.ibb.gov.tr
- http://devcitysdk.ibb.gov.tr

6.1.2. Data sets

Istanbul Endpoint has open data sets related with the mobility. All data sets provided by Istanbul Metropolitan Municipality.
admr layer has the all administrative boundaries in Istanbul. All data has been matched with the OSM tags like admn_level=3, admn_level=4 and admn_level=5. This data sets includes a city boundary, 39 district and 783 neighbourhoods.
gtfs layer is the main public transportation layer. It has 535 Routes, 9770 Stops, 117480 Trips and nearly 42000 Average trips per date for Bus, Railroads and Waterway.
istanbulkart layer describes the place where public transport ticket can be bought. There are 112 istanbulkart point in that layer.
osm layer stores the all open street map data for İstanbul. Main role of this data is proving pedestrian ways for routing algorithms.
parkingareas layer describes the car parking areas in Istanbul. This layer has 1485 parking area polygons in it.
station layer describes the bus depots for intercity buses and there are 2 station in that data set.
taxi layer describes the taxi stands in Istanbul. There are 271 taxi stand points in that data set.

6.1.3. Applications and services

A Hackathon, named Hackathonist, was organized by TAGES in cooperation with Istanbul Metropolitan Municipality on 1-3 November 2013 in Istanbul. The applications have been developed during Hackathonist listed below;

Travelist

- Searches for POIs
- Route Planning (A to *)
- Android and iOS

Smart Citizen

- Smart Participation
- Asks for confirmation (nearest 60 people)

CitySDK Android Library

Traffic+

- Sends/Receives voice messages for traffic information and, generates traffic data with gps positions.

City+

- Augmented Reality
- Searches for POIs
- Route Planning

6.1.4. Feedback on the Mobility API

Based on Istanbul Endpoint configuration and implementation experiences, CitySDK needs more cities. With the focus of mobility API the main requirements is a routing algorithm. Because of defining CitySDK ecosystem, Istanbul's studies show that CitySDK is interoperable. Enriching that ecosystem with more powerful tools will create more comfort developers and cities.
Also the success stories of current pilots will be a good example for new cities. New cities will make CitySDK more useful.

6.2. Dissemination and engagement activities

6.2.1. Engagement activities

The various activities are being carried out in Istanbul both by TAGES and Istanbul Metropolitan Municipality (IMM), Turkish partners of the project, to engage the developer communities and end-users into open innovation processes. This happens through hackathons, developer meet ups and citizen workshops both in Turkey and several EU countries, as well as through dissemination via related events and publications.

The first developer engagement event of Istanbul City was CitySDK Istanbul Hackathon named "Hackathonist" organized in Istanbul on Nov 1-3, 2013. The details of Hackathonist are given in the sub-sections below.

TAGES and IMM also carried out the engagement activities in the several events where they participated in Turkey and also in the EU countries. The details of these events and the activities are given in 7.2.2 Project Dissemination Section.

.

6.2.1.1 HACKATHONIST, Nov 1-3, 2013, Istanbul

6.2.1.1.1 Backround and Preparation Phase

HACKATHONIST was an award winning developer engagement event as an activity of CitySDK Project, organized by TAGES (www.tages.biz) in cooperation with Istanbul Metropolitan Municipality (www.ibb.gov.tr) and technology sponsorship of Turkcell Developers of The Future (https://gelecegiyazanlar.turkcell.com.tr) during the DevFest organized by the Google Developer Group Istanbul. The aim of the event was to introduce CitySDK project and Istanbul Mobility APIs developed in the project to the developer community and to organize a competition to be developed creative and useful smart city applications by the developers. HACKATHONIST is also the 1st Hackathon Event with Istanbul Mobility Open Data.

The interest of the people was really high during the registration period of the event and also afterwards. Due to limited participation, the registration applications of the participants were evaluated by a questionnaire and among 324 applications in total, **55 of them with 13 teams** were selected. The required information and the announcements of the event were given through the various channels such as Tumblr page (http://hackathonist.tumblr.com/), Facebook (https://www.facebook.com/hackathonist?fref=ts), Twitter (https://twitter.com/search?q=%23hackathonist&src=typd&f=realtime), Eventbrite (https://hackathonist.eventbrite.com/) and also the official web pages and social media pages of the organizers and sponsors. The following dissemination and promotional materials have been also designed and created to be used during the Hackathonist: Roll-ups, T-shirts, stickers, promotion cards and documents and participation certificates.

6.2.1.1.2 Data sets made available

Istanbul Mobility API which was developed as a result of CitySDK Project had been available for the Hackathonist. Mobility API includes Istanbul City's transportation and the POIs data. The mobility API is available in **http://devcitysdk.ibb.gov.tr/**. Besides the Istanbul Mobility APIs, TURKCELL Location Based Data and APIs were also available through the following links:

http://partnerportal.turkcell.com.tr/spgw/services/AuthenticationPort?wsdl,

http://partnerportal.turkcell.com.tr/spgw/view/LbsOpaqueService.wsdl

6.2.1.1.3 Award Categories

1. Best Application Created on the CitySDK API, 1000 TL + Travel Support for 1 person to Manchester 2014 Hackathon
2. Most Innovative Use of Data, 3 T40 Turkcell Smart Phones
3. "Developers of the Future" special award, 1 Turkcell Tablet
4. "Entrepreneurship Factory" special award, Participation to 2-days Entrepreneurship Training Programme and its book.

6.2.1.1.4 Organization and Results

Hackathonist was held in Istanbul Technical University at User Experience & Telecom on the Job Training and Test Center on November 1-3, 2013. The kick-off was held on November 1st, 2013 where CitySDK project, APIs and Istanbul transport data were presented and the participants found a networking opportunity to build teams. Hackathonist started on November 2nd 2013 where it took 31 hours and ended on November 3rd, 2013 in the afternoon with the presentations of the teams. Among 55 selected participants, 40 of them with 10 teams were participated with 3 participations on-line. The jury was composed of Fatih Yentürk, İBB, Sezer Yeşiltaş, Turkcell, Bert Spaan, Waag Society, Erman Turan, Entrepreneurship Factory, Bülent Erbaş, Koç Sistem. The main selection criteria was to develop a unique and interesting, close to the product, creative, visual and use friendly smart city application with a good them cooperation and attractive presentation. After jury's hard decisions 5 teams were awarded in 4 categories.

1. Best Application Created on the CitySDK API, Travelist by Team 2 (Cemile Diler Özdemir, Çağatay Koç, Göktürk Ramazanoğlu, İlhan Adıyaman, Altuğ Bayram, Cihan Turkay)

   The winner of best application created on the CiytSDK API won 1000 TL and travel support for 1 person to Futureeverything Festival in Manchester on March 26th – April 1st 2013.

2. Most Innovative Use of Data, Smart Citizen by Team 8 (Onur Sarıkaya, Tahsin Demir, Rıza Selçuk Saydam, Hasan Devinç, Emir Karşıyakalı, Onat Yiğit Mercan, Hasan Sarp Somer)

   The winner of Most Innovative Use of Data won 3 Turkcell T40 SmartPhones
3. "Developers of the Future" special award, CitySDK Android Library by Mikail Oral

   The winner of "Developers of the Future" won a Turkcell Tablet
4. "Entrepreneurship Factory" special award, City+ by Team 5 (Yavuz İrfanoğlu, Seçkin Tokcan) and Traffic+ by Team 1 (Mert Saraç, Turaç Demir, Sabri Şahin Can)

   The winners of "Entrepreneurship Factory" won participation to 2-days Entrepreneurship Training Programme and its book.

The next Hackathonist was planned on May 2014 with better and exciting expectations from the Istanbul City developer community.

6.2.2. Project dissemination

Istanbul Metropolitan Municipality and TAGES disseminate the CitySDK project, CitySDK toolkit and Mobility API in the several events where they participate in Turkey and also in the EU countries. Through these events, other municipalities in Turkey and developer communities got a chance to collaborating for smart city projects. The project is also disseminated in the local events such as Hackathonist (see 7.2.1.1) organized by both these partners. Besides participating to the events, the articles and news on CitySDK are prepared and shared with the network of TAGES and IMM through newsletters, journals, newspapers and e-mailing as another dissemination activity. The main tools and materials used in the Dissemination activities are as following:

- CitySDK and Mobility API ppt presentations
- CitySDK Roll-up
- Hackathonist Roll-up, sticker, t-shirt, promotion cards and participation certificates

Some of the events where CitySDK has been disseminated by TAGES and IMM are also listed in the table below:

Date	Location	Event	No of people
November 7, 2012	Istanbul, Turkey	VI. Istanbul Informatics Congress, Smart Cities	Around 200
January 26-27, 2013	Istanbul, Turkey	DJCamp2013	Around 100
March 8-10, 2013	Istanbul, Turkey	Hack the Hackaton Event	Around 50
March 12, 2013	Istanbul, Turkey	IBM Smarter Solutions Summit in Istanbul	Around 150
March 19-24, 2013	Manchester, UK	FutureEverything Festival	Around 600
March 28- 29, 2013	Istanbul, Turkey	1st Smart Municipal Summit	Around 150
June 14-15, 2013	Istanbul, Turkey	Android Developers' Day	Around 700
Sep 24-26, 2013	Istanbul, Turkey	ICT Summit Now 2013	Around 4921
Sep 2-7, 2013	London, UK	Campus Party Europe	Around 10000
Sep 27-29, 2013	Istanbul, Turkey	Hack the Hackathon	Around 60
Nov 1-3, 2013	Istanbul, Turkey	Hackathonist	Around 60
Nov 6-8, 2013	Vilnius, Lithuania	ICT 2013	Around 4827
Nov 13, 2013	Istanbul, Turkey	7th Istanbul Information Congress – Smart Cities	Around 100
Nov 11-13, 2013	Ankara, Turkey	Congress of Geographical Information Systems	Around 100
Nov 22-23, 2013	Istanbul, Turkey	International Smart Infrastructure and Geographic Information Systems Congress	Around 750
Nov 28-29, 2013	Ankara, Turkey	30th National ICT General Assembly	Around 500
Dec 19-22, 2013	Istanbul, Turkey	Kodathon	Around 60

The links of some of the articles on CitySDK published are given below:

- Istanbul becomes a pilot European smart city, March 23, 2012 http://www.tages.biz/index.php?module=news&page=readmore&news_id=416,
- Smart City SDK Meeting in Istanbul, Oct 29, 2012 http://www.tages.biz/index.php?module=news&page=readmore&news_id=422
- Istanbul has to be Smart, May 5, 2012 http://www.bthaber.com/istanbul-%E2%80%9Cakillanmak%E2%80%9D-zorunda/
- Open Mind for Smart City, Feb 12, 2013 http://www.tages.biz/index.php?module=news&page=readmore&news_id=426
- Smart Ideas Rushed in Manchester, Apr 17, 2013 http://www.tages.biz/index.php?module=news&page=readmore&news_id=430

- CitySDK APIs in Android Developer Days, Jun 28, 2013
  http://www.tages.biz/index.php?module=news&page=readmore&news_id=435
- Join HACKATHONIST for Smart Istanbul Apps!, Jun 28, 2013
  http://www.tages.biz/index.php?module=news&page=readmore&news_id=436
- Public Authorities' Baby Steps on Open Data, Oct 21, 2013
  http://www.tages.biz/index.php?module=news&page=readmore&news_id=441
- Develop with Istanbul Transportation Data, Oct 21, 2013
  http://www.tages.biz/index.php?module=news&page=readmore&news_id=442
- Catch the Research and Innovation Funding Opportunities, Dec 26, 2013
  http://tages.biz/index.php?module=news&page=readmore&news_id=446
- Apps with Istanbul Transport Open Data Awarded, Dec 30, 2013
  http://tages.biz/index.php?module=news&page=readmore&news_id=445

6.3. Impact

6.3.1. Outcomes from opening data

Open Data is restricted by the governmental laws in Turkey. With the special allowance just only for this project, IMM has published open data very first time. Private business and developer communities were interested in that issue. For the point of citizen, open data brings more qualified services and variety together. Especially in Hackathonist event, with the open data many applications were created. The most important outcome is the increase of awareness on open data in Turkey by TAGES and Istanbul Metropolitan Municipality. Recently, an open data session was organized within the 30th National ICT General Assembly which was held on Nov 27-28, 2013 in Ankara, Turkey and the issues on open data were discussed with the people from government and private institutions. CitySDK Project and Hackathonist were also told during the session.Stakeholders involvement in open innovation and service co-design.

Two main requirements of developing an application are data and developer. Developers cannot find data easily. When data is collected, the second tough subject is methodology of using that data. CitySDK has solutions for all that steps. Except publishing open data by local authorities, developers can create layers on CitySDK. All the data can be used by nodes and there are built-in spatial queries. All of them make CitySDK developer friendly.

6.3.2. Project communication

Istanbul Metropolitan Municipalities official twitter accounts tweets CitySDK news. Also IMM published news from web pages.

https://twitter.com/ibb_ABveIFM

http://www.ibb.gov.tr/sites/avrupa-birligi/abfonlari/Sayfalar/ABProjeler.aspx

http://www.ibb.gov.tr/tr-TR/Pages/Haber.aspx?NewsID=19642#.UqrFOuLlO24

http://www.ibb.gov.tr/tr-TR/Pages/Haber.aspx?NewsID=21521#.UqrFPeLlO24

http://www.gdgistanbul.com/2013/10/devfest-istanbulda-hackathonist-heyecan.html
http://hackathonist.tumblr.com/
http://ekonomi.haberturk.com/makro-ekonomi/haber/662030-turkiye-abye-odediginden-fazla-fon-alacak
http://cadde.milliyet.com.tr/2013/01/29/YazarDetay/1661608/akilli-sehir-olur-muyuz-

6.4. Facts

Contact info:

İlker YILMAM, Istanbul Metropolitan Municipality
ilker.yilmam@ibb.gov.tr
+90 212 455 25 29

End-point url

- http://apicitysdk.ibb.gov.tr

How long it took to implement the API?

- Man-weeks = 1.18
- Costs for implementation (€) = 0 €

Services and apps created using the API

- Travelist
- Smart Citizen
- City+
- Trafik+

Code Libraries or other resources created

- CitySDK Android Library - https://github.com/mikailoral/citysdklib

7. Replication Pilot: Helsinki

7.1. Implementation activities

7.1.1. Implementation of the Mobility API

We installed the server software needed to implement CitySDK Mobility API endpoint in Helsinki in October 2013. Installation was quite straightforward and Waag Society was supporting the process online. First dataset provided through the CitySDK API, besides the Open Street Map data, was Helsinki Region public transport data in GTFS format.

Helsinki has already quite well established ecosystem around public transport open data. Regional authority HSL/HRT has opened APIs five years ago and there are dozens of journey planner apps for all the major mobile OS platforms. CitySDK adds value on this by bringing harmonized API with several European cities. Developers benefit also from the Open Street Map compliant way of searching and retrieving information.

We are now in the phase of testing the API and endpoint. Next steps are defining a roadmap about which datasets should be included and setting up the API documentation and promotion for the developers, including the domain name.

7.1.2. Data sets

Data sets currently available:
- Open Street Map data
- Public Transport, schedules, routes and bus stops, in GTFS format
- administrative boundaries (still some issues with this)
- Open311 requests are available from Amsterdam endpoint, will be moved also to Helsinki later

Next steps:
There are over 1.000 datasets listed in the Helsinki Region Infoshare catalog (www.hri.fi) and some of them relatively easily available for CitySDK Mobility API. Cadastre data is one example of location based data which can be linked through CitySDK API and is one of the first to be implemented 2014.

We have been evaluating Mobility API as a way to provide also tourism related events in addition to API developed in WP5, but that proved to be problematic because the time dimension is missing in

the search. The POIs where events take place can of course be added as a layer, but events are usually searched within a time frame.

7.1.3. Applications and services

CityNavigator

Helsinki Region Transport Authority, HRT / HSL, has initiated an open source project to create multi modal mobile journey planner, CityNavigator. CitySDK Mobility API supports this development project reducing the effort needed to transfer CityNavigator to other cities. CityNavigator is now working in Helsinki, Manchester, Amsterdam (testing phase), Tampere and Oulu.

This mobile web application is a 100% open public transport journey planner and navigation application for on-the-go use using HTML5 and incorporating Open Data and Open Source components such as the OpenTripPlanner API for journey planning and the CitySDK Mobility API for point-of-interest search. The application works on most mobile platforms including iPhone and Android, and it's designed to be straightforward to deploy in other cities as well. Transport for Greater Manchester and Tampere have joined HSL in the Navigator effort, which is made possible by building on datasets with potentially global coverage including OpenStreetMap and timetables in the GTFS format.

See: http://www.europecommons.org/app/hsl-navigator

Now

We have tested the “Now” application of WAAG lightly in Helsinki and it seems to be working also in Helsinki.

HRT Code Fellow Tuukka Hastrup has also made some tests with the CitySDK API related to routing and addresses:

http://tuukka.kapsi.fi/streets/

http://tuukka.kapsi.fi/entrances/

7.1.4. Feedback on the Mobility API

To get more cities onboard we need clear story about the continuity of the development of the Mobility API after the project. We need to define also common next steps in populating the API with new data sets. To achieve harmonization goals of CitySDK work we should agree on common standards for the data layers. One good example of the low hanging fruits is all parking related data.

SIRI and Datex II are also widely used and gaining momentum and thus should be supported in the future when applicable.

7.2. Dissemination and engagement activities

7.2.1. Engagement activities

Forum Virium Helsinki organized together with Helsinki Region Transport Authority HRT, Ministry of Transport and Communications, Finnish Transport Agency and the National Broadcasting Company YLE a transport hackathon - Dev4Transport - on October 2013. The event gathered together some 50 developers, authorities and other open data enthusiastics to ideate and code new solutions and services utilizing open transport data. CitySDK APIs were introduced to the participants and they were being provided guidance throughout the 2-day event in utilizing the APIs. As a result the participants showcased 6 new services and ideas of which all were also submitted to the Apps4Europe-contest that took place in November-December 2013.

Helsinki has a developer portal Hel<3Dev at dev.hel.fi. Helsinki Loves Developers brings together a wide range of activities through which the City of Helsinki collaborates with developers. The CitySDK Mobility-related data and API's are also presented in the website and all CitySDK-related events and news to developers are promoted through the website. Helsinki Loves Developers activity was started by CitySDK project in co-operation between City of Helsinki IT department and Forum Virium Helsinki.

7.2.2. Project dissemination

Dissemination activities are described in detail in the Work Package 6 report.
Project partners in Helsinki - Forum Virium Helsinki, City of Helsinki and Sanoma (biggest media company in Finland) - all used their existing channels to disseminate progress of the project (Twitter, Facebook and web sites, news letters, informing journalists).

7.3. Impact

Helsinki is implementing all 3 CitySDK domains and the lead pilot effort is put on the Participation domain (WP3). Thus the impact CitySDK has had in general to the open data scene and developer ecosystem in Helsinki has been described in the other WP reports.
Mobility API is one of the CitySDK APIs developers can utilise and because public transport APIs have been mostly open in Helsinki already before the project, the impact is seen in the

harmonisation of the transport APIs between the cities. Which of course also strengthens the general story of the benefits of opening data.

CitySDK concept has been received very well amongst the Finnish cities and it is included in the so called SixPack strategy in Finland, where 6 largest cities in Finland will invest together in opening up data and APIs during the next seven years. Harmonisation and sharing the results are built-in requirements in the strategy, so that work complements the work of CitySDK project nicely. EU (ITI) and Finnish state have decided to fund two thirds of the budget, which is roughly 10 million EUR a year during the next seven years. That money is not used solely on harmonizing APIs, but opening up the cities in wide variety of ways, eg. also through social programs.

7.4. Facts

Contact info:

Pekka Koponen, Forum Virium Helsinki, pekka.koponen@forumvirium.fi, +358 40 5017114
Susanna Ollila, Forum Virium Helsinki, susanna.ollila@forumvirium.fi, +358 40 6722515

End-point url

- frontpage http://144.76.172.136/
- API http://144.76.172.136:8080/
- map http://144.76.172.136/map
- CMS https://144.76.172.136:8443/

Naming of the URL is still an open issue. It will be done in early 2014 when testing phase is over and API will be promoted to the developers.

How long it took to implement the API?

Basic installation took 3 days. When importing new datasets time needed varies depending on the data set (ongoing process).

Cost of the implementation was 0 € for the work force. We co-operated with the Commons4EU project and Code4Europe code fellow working for HRT installed the endpoint. We had to set up a dedicated server which costs about 100 EUR/month at the moment. However, most of the work related to create a working, well supported API with the right data sets, is not technical. We are having constant negotiations with the data owners and other parties involved.

Services and apps created using the API

CityNavigator (HSL navigator)

8. Smart Mobility SDK Components

8.1. The SDK components

The CitySDK MobilityAPI is a linked data distribution platform. Developed by Waag Society, the distribution API is a component of the CitySDK toolkit. This toolkit supports the development of open and interoperable interfaces for open data and city services in eight European cities (Amsterdam, Helsinki, Manchester, Lisbon, Istanbul, Lamia, Rome and Barcelona). The CitySDK MobilityAPI enables the distribution and linking of open data sets and city services.

Benefits

1. CitySDK MobilityAPI is a one-stop-shop for developers. With standardized interfaces, developers can build better apps and services for end users and governments.
2. Moveable code – developers can use the same interface for open data, in Amsterdam, Helsinki and Istanbul alike.
3. The API enables the Read/Write City: per layer, data can be added to objects the city, using URIs.
4. Data exchange between citizens and the city will become open, more efficient and more transparent data.
5. The API enables innovation for businesses, media and citizens, and manages the constant technological change: adding new datasets and services is made easy.
6. The API is released as open source on GitHub, and can easily be implemented and amended to local needs.

Features

1. Open API, open source
2. Unified REST API, data from different sources available on a per-object basis
3. No access keys for reading
4. Write access for data owners and app developers
5. CMS for data owners for easy integration of new datasets
6. Interactive linking of data sets
7. Map viewer
8. Standardized interface in 8 cities
9. Ruby API gem

Adding your own data

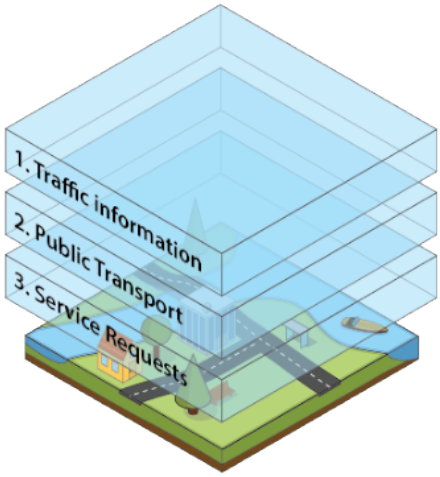
If you are a data owner, the API offers a user-friendly CMS. It makes it easy to upload and (automatically) update your static and realtime data sets.

Concept

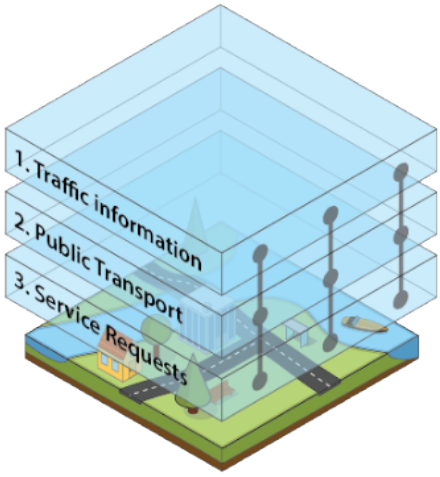
The concept of CitySDK MobilityAPI is best explained using Tim Berners-Lee's Five Star Linked Open Data model.

On the right, you see a city. Like more and more cities all over Europe, this city is opening its data to the public, and making it available through data catalog portals like CKAN. Moreover, it is also working on APIs and services to facilitate the communication with citizens (i.e. Open311).

Although many different open datasets are available through this city's data catalog, it's not always easy for developers to use the data. Data is offered in different file formats, has unclear update policies and incomplete metadata. When the data is accessible under an open license and offered in a structured and non-propriatory format, the data is rated with three stars.

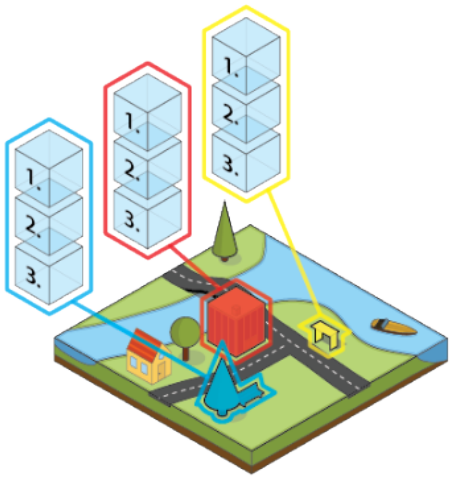


Moreover, it can be difficult to find out that different datasets have data about the same objects. According to the five star model, URIs should be used to uniquely identify objects and links should be created between those objects to define relations; it is not unusual that data about one object is contained in multiple datasets.

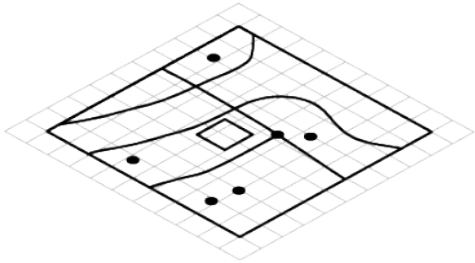


The city on the right has decided to use CitySDK to give all individual objects - such as buildings, public parks and bus stops (or train stations, bridges, museums and parking ga rages) - a unique identifier. This URI can then be used to identify those objects across multiple datasets.

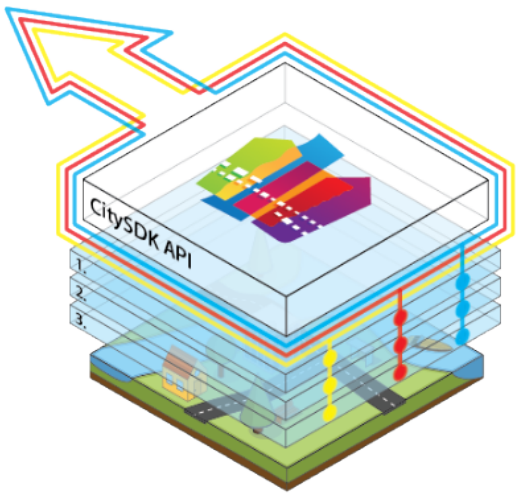
All objects with an URI have a geographic location, and are called a node within CitySDK. To those nodes, per layer, key-value data can be added. For example: the URI of a bus stop can be used to access bus schedules, but also in Open311 service requests, and with the URI of a road, data about planned roadworks and traffic information can be accessed.



Many of the datasets in open data catalog contain data about objects that exist in the real world. This is why CitySDK uses the OpenStreetMap database as a geospatial base layer. Via CitySDK, all nodes, ways and relations from OSM can be used to attach data from other open datasets to. GTFS data will be attached to OSM bus stops and train stations, tourist information will be attached to OSM museums and theatres an d planned roadwork data to OSM roads.



CitySDK is a linked open data distribution platform - for static and real-time data - and connects existing open datasets, data catalogs and APIs. CitySDK provides one easy to use REST API with both JSON and Turtle/RDF output. With this API, datasets that were previously difficult to access and use can be accessed in a single unified way, on an per-object basis.



8.1.1. Software

All code and documentation can be found at: https://github.com/waagsociety/citysdk.

CitySDK Mobility links datasets through addressable, real-world objects. The organization of the information is extremely simple; there are nodes and information on these nodes. A node can be anything; a landmark, a bus-stop, a stretch of road, a town or a public transport line. All nodes are a node, but there are multiple types of nodes which are treated specially: these are route, region, ptstop and ptline. A route is a series of nodes in a particular order. A region is a node that represents a formal administrative region. Since the API also deals with open government data, it is convenient to have these nodes available as a shortcut. This allows for easy attachment of data on towns, suburbs and neighborhoods. We define the country as level 0, the provinces as 1, the municipalities as level 3 (level 2 is for larger regions that are not provinces, like Stadsregio Amsterdam), quarters ('wijken') are level 4, neigborhoods are level 5. Amsterdam also defines an even smaller area, level 6. We are assessing whether to bring these levels in line with the levels as used in OSM, so this may change. ptstop and ptline nodes represent public transport stops and lines.

There is a further entity, the layer. Layers are the means through which the data is organized, and through which data is addressed and updated. A layer has an ownerwho will be responsible for the data on that layer. A layer is also the organizational unit through which applications will write to the API. The app reads and processes information from any layer, but writes only to its own layer(s). The base geography data lives in the OpenStreetMap layer (name: osm); this basically represents the OSM nodes, ways and relations. The regions live in the admr layer, public transport is on the gtfs layer. Nodes are uniquely addressable through their cdk_id. Although thecdk_id of a node is not meant to represent data or information, conventions that we implement, and are handy to remember when exploring the API. For example:gtfs.line.gvb.25-1 represents southbound GVB tram line 25,admr.nl.amsterdam represents the municipality of Amsterdam and admr.nl.amsterdam_stadsdeel_centrum represents the city center of Amsterdam.

8.1.2. GET Interface

Read access to nodes

GET /nodes Returns a list of nodes, as specified through url parameters (see below). You will want to limit the selection on geography, tag or with layer specifications. Without layer spec, only the node information itself is returned.

GET /routes Returns a list of defined 'routes', see above with 'nodes'; you'll want to limit the

selection.

GET /regions	Returns a list of defined administrative regions, see above with 'nodes'; you'll want to limit the selection, usually.
GET /ptstops	Returns a list of defined public transport stops.
GET /ptlines	Returns a list of public transport lines.

URL parameters

There are several ways to specify and narrow down the extend of your request. You can specify the geographic area of interest (other than searching 'within' an area, as described below), limit your request to data and nodes on specific layers, and do searches on name or on specific data in specified layers. These options are passed through url parameters. Parameters can be combined in a single request, where this makes sense.

?per_page=<num>	Limits the number of returned nodes to <num>. Defaults to 10. The maximum value is capped in the backend, currently to 1000 nodes. Multiple requests may be necessary to get all the data for a request.
?page=<num>	Requests page <num>.
?name=<string>	Most nodes have a name. This parameter allows you to do a sub-string search on that name. The search is case-insensitive and returns nodes whith the specified <string> anywhere in the name.
?layer=<name>	Limits the request to nodes that have data on any or all of the specified layers. <name> is a single layer or a comma-separated (AND) or pipe-separated (OR) list of layer names. Wildcards can be used after layer name separators. Examples are ?layer=admr,cbs;?layer=divv.*|gtfs. Logic cannot be mixed, it is either all ANDs or all ORs.
?<layer>::<key>[=<val>]	Returns nodes with specified key-value pairs on specified layers. When =<val> is ommitted, returns nodes with any value present for that key. For example: ?osm::tourism=museum returns all nodes with museums on the osm layer, where ?osm::tourismreturns any osm node with the tag 'tourism'. You can specify multiple key-value pair filters in a single GET request. By default, only nodes are returned which satisfy **all** specified key-value pair filters (AND), but you can change this behaviour by setting URL parameter ?data_op=or. Then, all nodes are returned which satisfy **any** of the specified filters. For example, ?osm::tourism=museum|zoo&osm::amenity=theatre&data_op=or will return all OSM nodes that are either a museum, a zoo or a theatre.
?lat=<num>&lon=<num	This returns nodes within the specified radius (in meters) around lat/lon.

>[&radius=<num>]	Without the radius parameter the request returns the<per_page> geographically closest matches to lat/lon.
?bbox=[<t>,<l>,<b>,<r>]	Match nodes within the given bounding box. Order of the coordinates is top, left, bottom, right.
?geom	Returns the geometry values of the nodes. No parameter value. When not present the geometry is not returned; this can save considerable bandwidth when node geometries are not needed.
?count	Returns the total of matched nodes in the results. Default is off, as counting can have a considerable impact on performance.

The following URL parameters are used to filter routes by the nodes they consist of:

?starts_in=<cdk_id>	Returns routes starting in a specific node.
?ends_in=<cdk_id>	Returns routes ending in a specific node.
?contains=<cdk_id>[,<cdk_id>]*	Returns routes containing a specific set of nodes, in the correct order. The contains parameter takes at least one cdk_id, but you can also specify more nodes in a comma-separated list.

Access to elements within a geographic boundary

Apart from limiting your selection to bounding box or circle radius parameters in the url (see below), more ususally you'll want to limit the selection to a particular (administrative) region. The node your are specifying as a boundary can be any node, really, but will only make sense in the case of a node with definite area.

GET /<cdk_id>/nodes	All nodes geographically intersecting with node <cdk_id>.
GET /<cdk_id>/routes	All routes.
GET /<cdk_id>/regions	All regions.
GET /<cdk_id>/ptstops	All public transport stops.
GET /<cdk_id>/ptlines	All public transport lines.

Interface to individual nodes

GET /<cdk_id>	Returns data of the node specified. Useful in combination with layer url

	parameter to return the node together with data on specified layers. Otherwise, only the node's cdk_id, layer, name and, optionally, geometry is returned.
GET /<cdk_id>/<layer>	Shortcut to access node data on a specific layer. Differs from the layer url parameter in that this only returns the node data, not the node itself.
GET /<cdk_id>/<layer>/<key>	Access specific piece of data on a specific layer, see examples.

Layers

GET /layers	Returns a list of the layers currently defined; a general Overview of data available in the endpoint. Layers can be searched for by name and by category: GET /layers?name=divv.* or GET /layers?category=mobility
GET /layer/<name>	Returns information on the specified layer.

Commands

Information on nodes can be static, or dynamic. Also, some information is not stored as such, but can be derived from the data that is. To provide an interface to these kinds of data, some types of nodes can be queried through the **select** command. The general form of this command is /<cdk_id>/select/<command> Currently the following commands are defined:

On nodes:

GET /<cdk_id>/select/regions	will list the administrative region hierarchy starting at this node.
GET /<cdk_id>/select/routes	will list routes containing this node.
GET /<cdk_id>/select/routes_start	will list routes starting in this node.
GET /<cdk_id>/select/routes_end	will list routes ending in this node.

On routes:

GET /<cdk_id>/select/nodes	will list the nodes that make up a route.
GET /<cdk_id>/select/routes	will list the routes which nodes intersect with the nodes of route with /<cdk_id>.
GET /<cdk_id>/select/start_end	will list the start and end node of a route.

On ptstops:

GET /<cdk_id>/select/ptlines	will list the public transport lines that frequent a stop

GET /<cdk_id>/select/schedule	will list the schedule for the coming week for all lines that stop at this stop.
GET /<cdk_id>/select/now	will, for eacht line at this stop, give you the (real-time) departure times for the coming hour.

On ptlines:

GET /<cdk_id>/select/ptstops	will list the public transport stops that make up a pt line.
GET /<cdk_id>/select/schedule	will list the schedule for today; adding ?day=<n>will list schedule for the <n> days from now.

8.1.3. PUT/POST/DELETE Interface

Authentication

To use the CitySDK Mobility Write API, you need a valid user account. All Write API requests require authentication by means of a session key, a random string that provides temporary, secure access to the Write API. To start a session and request a session key, you need to do the following call:

GET /get_session ?e=<email> &p=<password>	Request a session key using a valid <email> and <password>combination.

Important: the /get_session call requires a **HTTPS connection** for secure transfer of user account details. The API response will be of the following form, with the session key in the resultsarray:

```
{   "status": "success",   "results": ["session_key"] }
```

All the following API calls on this page expect this session key in the X-Auth HTTP header.
Session keys are valid for **one minute** only, but each request to the Write API will extend the validity with another minute. After one minute of inactivity, your session will time out and you will need to request a new session key to do new Write API requests.
When done, you should release your session by calling GET /release_session, again with your current session key in the X-Auth header. If you don't, you will not be able to acquire a new session util the current one has timed out.

GET /release_session	Release current session.

A Ruby utility library to illustrate and manage these steps is available here.

Adding Data

PUT /<cdk_id>/<layer> — Add data to layer <layer> of node <cdk_id>.

This call expects a JSON body in the following form:

```
{ "modalities": ["rail"], "data" : { "naam_lang": "Amsterdam Centraal", "code": "ASD" } }
```

The data field is a one-dimensional, unnested JSON object (e.g. a list of key-value pairs).
If data on layer <layer> already exists on node <cdk_id> the key-value pairs will be merged with the existing data, overwriting duplicate keys with data from the current API call.
If you want to completely replace all node data instead of merging old data with new data, you will have to use the delete data API call first.

You can supply a modalities array containing the types of transport modalities that are valid for the data you are adding.

Deleting Data

DELETE /<cdk_id>/<layer> — Delete the layer data of layer <layer> of node <cdk_id>.

By default, the node is kept when removing data. Afterwards, the <cdk_id> and the associated geometry still exist. If you completely want to delete the data as well as the node itself, you can add ?delete_node=true. This will only delete the node if it is a node that was created by the layer <layer>, of course. Also, the node will not be deleted if other layers have added data to this node.

Bulk API: writing/updating multiple nodes and node data at once

PUT /nodes/<layer> — Write and/or update multiple nodes on layer <layer>.

You can only write data or create nodes in one layer at a time. Of course, you can only modify nodes on layers you own. You can create, modify and delete layers in theCitySDK Mobility CMS.

Input

The Bulk API expects JSON in the following form:

```
{ "create": { "params": { "create_type": "create", "srid": 4326 } }, "nodes": [ { "id": "ASD", "name": "Amsterdam Centraal", "modalities": ["rail"], "geom" : { "type": "Point", "coordinates" : [ 4.9002776, 52.378887 ] }, "data" : { "naam_lang": "Amsterdam Centraal", "code": "ASD" } }, { "cdk_id":
```

```
"n46419880",      "modalities": ["rail"],            "data" : {       "naam_lang": "Amsterdam Centraal",
"code": "ASD"      }          }   ] }
```

The create object contains parameters, the nodes array contains the nodes you want to create, update or add data to.

Create parameters

- srid: define the SRID of the GeoJSON geometries for all new nodes to be created.
- modalities: defines the node-level modalitites. Really only relevant for route nodes.
- create_type: sets the way the Bulk API handles existing and new nodes. Possible values are update, routes and create. Default is update. The differences between the create types are explained in the next paragraph.

Nodes

The bulk API can add data to existing nodes and create new nodes and add data to those new nodes. Each node in the nodes array require a data field, a one-dimensional, unnested JSON object, e.g. a list of key-value pairs. For example:

```
{   "key1": "value1",   "key2": "value2",   "key3": "value3", }
```

Nodes without a data field are skipped.

You can supply a modalities array containing the types of transport modalities that are valid for the node; these are associated with the layer data, not with the node itself (route nodes can have modalities in themselves).

The create_type parameter defines three modes of handling nodes in the nodesarray.

Add data to or update existing nodes

```
{"create_type": "update"}
```

This is the default mode, used if you don't specify create_type or if you set"create_type": "update". The update mode only only adds data - on the specified layer - to already existing nodes. Each node in the nodes array should specify the cdk_id of the node the data should be added to. If data on your layer already exist for this node, this data will first be removed. If you want to append data to existing nodes, you can - for now - only do this by using the add data call for each node separately.

You do not need to provide a geometry field and nodes without a cdk_id field are skipped. The data will be added to existing nodes with an existing geometry.

Create routes and add data

```
{"create_type": "routes"}
```

This mode expects a cdk_ids field which contains an ordered list of cdk_ids through which the new route should be created. You can also supply an identifier and a name for the new route in the id and name fields. The API will create a new route with a cdk_id based on the layer and identifier and will add the key-value data from the data object to the new route. The bulk API will return a list of newly created identifier-cdk_id combinations which you can use to link your own systems with CitySDK.
If nodes with a cdk_id field are encountered instead, the bulk API will handle those nodes the same way as it would in the default update mode described above.
Nodes with an id field are disgarded.

Create new nodes and add data

```
{"create_type": "create"}
```

All new nodes you want to create need to have an id field. This identifier must be unique for your layer. You can also supply a name for the new node in the name field. Furthermore, you *must* supply a valid GeoJSON geometry in the geom field.
The Bulk API will create a new node with the name and geometry you supplied, and will add the key-value data from the data object. The API will generate a newcdk_id based on the layer and identifier. The bulk API will return a list of newly created id-cdk_id combinations which you can use to link your own systems with CitySDK.
In this mode, the bulk API will handle nodes with a cdk_id or cdk_ids field as it would in update or routes modes.

Output

Bulk API output looks like this:

```
{ "status": "success", "create": { "results": { "updated": [], "created": [ { "id": "ASD", "cdk_id": "<layer>.asd" } ], "totals": [ "updated": 0, "created": 1 ] } } }
```

The create.results.created object contains a list of id-cdk_id combinations which you can use to link your own systems with CitySDK.

Modalities

CitySDK distinguishes the following modes of transport. You can use any combination of these when creating new routes or when adding new data to existing nodes or routes.

Modality	Description
tram	Tram, Streetcar, Light rail
subway	Subway, Metro

Modality	Description
rail	Rail
bus	Bus
ferry	Ferry
cable_car	Cable car
gondola	Gondola, Suspended cable car
funicular	Funicular
airplane	Airplane
foot	Foot, walking
bicycle	Bicycle
moped	Light motorbike, moped
motorbike	Motorbike
car	Car
truck	Truck
horse	Horse

8.1.4. Match API

The Match API enables you to link objects your own data with existing nodes in CitySDK Mobility. The API will match objects on name, location and key-value data and return the cdk_id of the CitySDK nodes it finds. You need a valid CitySDK user account to use the Match API.

POST /util/match — Match objects with existing CitySDK nodes.

The following use cases will illustrate why you need the Match API and how it can be used:

1. **Train stations**: suppose you have data about train stations (e.g. departure times) in a certain country. Most of those stations will be probably already exist in CitySDK on the OSM base layer. You can read more about how OSM tags train stations on the OSM wiki. The Match API can be used to find those train stations and link them to the corresponding objects in your dataset, based on name, location and meta-data.
2. **Museum opening times**: let's say you keep an updated list of the opening times of all the museus in your home town. Again, probably many of the museums in your dataset will be readily available in CitySDK, since OSM has a dedicatedmuseum tag. If you provide your museum dataset in the JSON structure explained in detail below, the Match API will try to find museums that exist in both your dataset and CitySDK.

3. **Real-time traffic information**: the previous use cases showed how the Match API can match objects from your datasets to single CitySDK nodes. Just finding single nodes will not suffice when you have, for example, traffic jam data for stretches of highways. OSM uses the highway tag to distinguish different types of roads and highways. The Match API can also find paths through connected OSM ways if they closely match a given linestring.
Important: matching is a database-intensive task and requests to the Match API can take some time to return. Requests can time out when you are trying to match too many nodes at once. If time-out problems occur, please try to match your data in batches.

Input

The Match API expects JSON in the following form:

```
{ "match": { "params": { "radius": 350, "debug": true, "srid": 4326, "geometry_type": "point", "layers": { "osm": { "railway": "station" } } }, "known": { "ASD": "n46419880" } }, "nodes": [ { "id": "ASD", "name": "Amsterdam Centraal", "modalities": ["rail"], "geom" : { "type": "Point", "coordinates" : [ 4.9002776, 52.378887 ] }, "data" : { "naam_lang": "Amsterdam Centraal", "code": "ASD" } } ] }
```

Match parameters

- radius: the distance from the node geometry in meters within which the Match API should search.
- srid: defines the SRID of the GeoJSON geometries for all nodes to be matched.
- modalities: defines the node-level modalitites. Really only relevant for route nodes.
- geometry_type: sets the geometry type of the nodes in the nodes array. Must be either point or line.
- layers: specifies, per layer, filters on key/value pairs of acceptable nodes. Matched nodes must satisfy any of the given filters, not all.
- debug: the Match API returns useful debug information when set to true.

Known matches

You can turn off matching for certain nodes if you've already (manually) linked your data to existing CitySDK nodes. This can be useful if you want to (partially) update your dataset but want to keep existing links, or if you want to manually set thecdk_id for objects you know the Match API returns the wrong results. The known object is a key/value hash, where keys are ids from your data and the values are **existing** cdk_ids.

Matching

Point matching

```
{"geometry_type": "point"}
```

The Match API will search all applicable nodes within radius meters of the node geometry and sort them by name similarity and distance, using amatch's pair distance metric and PostGIS's ST_Distance function.

Line matching

```
{"geometry_type": "line"}
```

The Match API can do more than just find matches between point geometries; the Match API can also try to match given linestrings to the OSM node/way graph and return sets of CitySDK nodes from which consequently CitySDK routes can be created. OSM (and thus CitySDK) contains networks of connected nodes and ways, defined by different OSM tags. You can use OSM's highway tag to match a linestring to an existing stretch of road, highway or bike path, but you can the same for rail networksor waterways.

More information about the OSM graph can be found here:

- http://wiki.openstreetmap.org/wiki/Routing
- http://wiki.openstreetmap.org/wiki/OSM_tags_for_routing
- https://www.gaia-gis.it/fossil/spatialite-tools/wiki?name=graphs-intro

For example, let's consider a GeoJSON file which contains a linestring annotated with real-time traffic speed in Amsterdam. When you can view this file on a map byconverting it to KML and opening it (i.e. in Google Earth), you can see that the linestring lies close lies close to a road called the 'Muiderstraatweg'. This Muiderstraatweg is available both in OpenStreetMap and in CitySDK. When set to line matching, the Match API will try to find a connected network close to any given linestring, using the provided OSM key/tag filters. Through this network, the Match API will use Dijkstra's algorithm to find a path in the OSM network starting close to the start of the linestring and ending close to the end of the linestring.

The following Match API input is a fragment from the divv.traffic layer importer data:

```
{ "match": { "params": { "radius": 125, "srid": 4326, "debug": true,
"geometry_type": "line", "ignore_oneway": false, "layers": { "osm": { "highway": [
"unclassified", "road", "motorway", "motorway_link", "trunk", "trunk_link",
"primary", "primary_link", "secondary", "secondary_link", "tertiary", "tertiary_link"
] } } } }, "nodes": [ { "id": "TrajectSensor_Route130_R", "data": {
"location": "TrajectSensor_Route117_R", "velocity": 48, "length": 700 }, "geom":
```

{"type":"LineString","coordinates":[[4.95344,52.33958],[4.95532,52.33916],[4.95684,52.33883],[4.95897,52.33840],[4.95982,52.33836],[4.96037,52.33825],[4.96195,52.33789],[4.96353,52.33755],[4.96622,52.33714],[4.96797,52.33662],[4.97038,52.33593],[4.97322,52.33510],[4.97585,52.33431],[4.97628,52.33422],[4.97675,52.33415],[4.97682,52.33411],[4.97704,52.33403],[4.97715,52.333886],[4.97741,52.333751],[4.97793,52.33356],[4.97960,52.33328],[4.98043,52.33315],[4.98169,52.33307],[4.98345,52.33302],[4.98399,52.33313],[4.98430,52.33330]]} }] }

You can set the ignore_oneway to true if you ant the line matcher to ignore the OSM oneway tag. The Match API will return a set of cdk_ids in the correct order, which is used to create a new CitySDK route:

- Route with divv.traffic data
- Nodes this route consists of

Polygon matching

Polygon matching is not yet supported.

Output

Important: the Match API will of course not always be able to match all the objects in your dataset with an object in CitySDK; sometimes the Match API will find no object at all, and sometimes it will find an incorrect match.

You can use the Bulk API to create missing nodes, but please only proceed with creating new nodes when you're absolutely sure the objects you need do not exist in CitySDK. You should always thouroughly check Match API results, try different matching parameters and try to look for objects manually as well. The linking of different dataset through existing CitySDK/OSM nodes is the main advantage of using the CitySDK API. The Match API can help data owners with this linking, but manual checks and manual linking will be inevitable. You can use the known matches object to manually turn off matching for certain nodes.

The Match API returns a modified version of the submitted nodes array which you can send directly to the Bulk API.