DELIVERABLE

Project Acronym: CitySDK

Grant Agreement number: 297220

Project Title: Smart City Service Development Kit and its Application Pilots

D3.2 – Participation, Lead and Replication Pilots reports, Smart Participation SDK Components

Revision: Final

Authors:

Hanna Niemi-Hugaerts (Forum Virium Helsinki)

Jaakko Rajaniemi (City of Helsinki)

Katalin Gallyas (City of Amsterdam)

Alex Serret Garrigues (Adjuntament de Barcelona)

Valter Ferreira (City of Lisbon)

Marcello Zini (Provincia di Roma)

George Koukas (Municipality of Lamia)

Adrian Slatcher (MDDA)

Dissemination Level

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

Revision history and statement of originality

Revision history

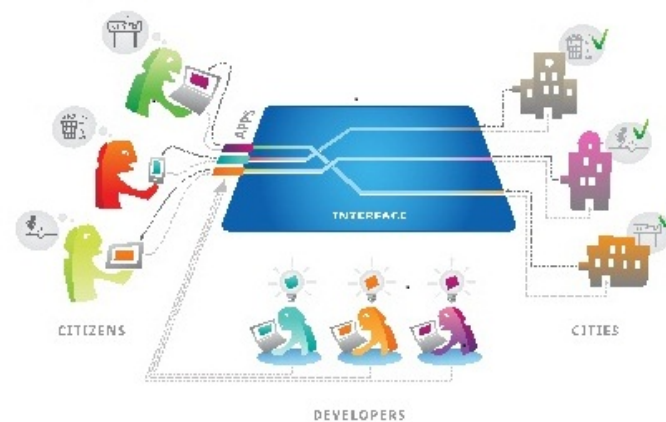| Revision	Date	Author	Organization	Description
Final	30.12.2013	Hanna Niemi-Hugaerts	Forum Virium Helsinki	All replication pilots included.

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This deliverable contains original unpublished work except where clearly indicated otherwise.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

Introduction

The purpose of the Smart Participation work package is to create an open interface that acts as an issue-reporting channel between the citizens and the civil servants. The work is based on the Open311 technology, which is a standardized protocol for location-based collaborative issue tracking. Further work on the interface was needed though, as it didn't fulfill all needs of the Lead Pilot and Replication Pilots.

The goal of this work package is to a) provide cities with a specification for issue reporting interface, b) support developers building apps that make citizen feedback easier and c) allow citizens to give feedback via commonly used virtual platforms or applications that are not necessarily owned or maintained by the city. To make the work manageable Smart Participation focuses mainly on location related issue reports, leaving topics like participatory budgeting out of scope. Having said that, the replication pilots of Rome and Manchester have taken slightly widened scope on utilising the API for their local use cases.

Within the Smart Participation domain interfaces have been and will be built between the feedback systems and other platforms to enable a direct flow of citizen feedback to relevant recipients in the City Hall. For the citizens, different pilots give an opportunity not only to give feedback but also to follow city's process of handling the given feedback thru commonly used digital services.

The CitySDK Issue reporting API

Requirements for Smart Participation

The requirements for the Smart Participation interface were collected both from existing services and through collaboration with active partners in Smart Participation. This resulted in the Smart Participation requirements document.[1] Partners then collaborated on prioritizing the suggested extensions to interface specification using a collaborative document. [2]

CitySDK Smart Participation interface specification

The specification of the interface for CitySDK Smart Participation Work Package WP3 can be found in the document D3.1 Participation Pilot Application and it's SDK components. The interface is specified based on commonly used GeoReporting version 2, which is better known as Open311 specification [http://open311.org/]. The interface is designed in such a way that any GeoReporting version 2 compatible client is able to use the interface. In addition to the standard version, new fields were added to API methods.
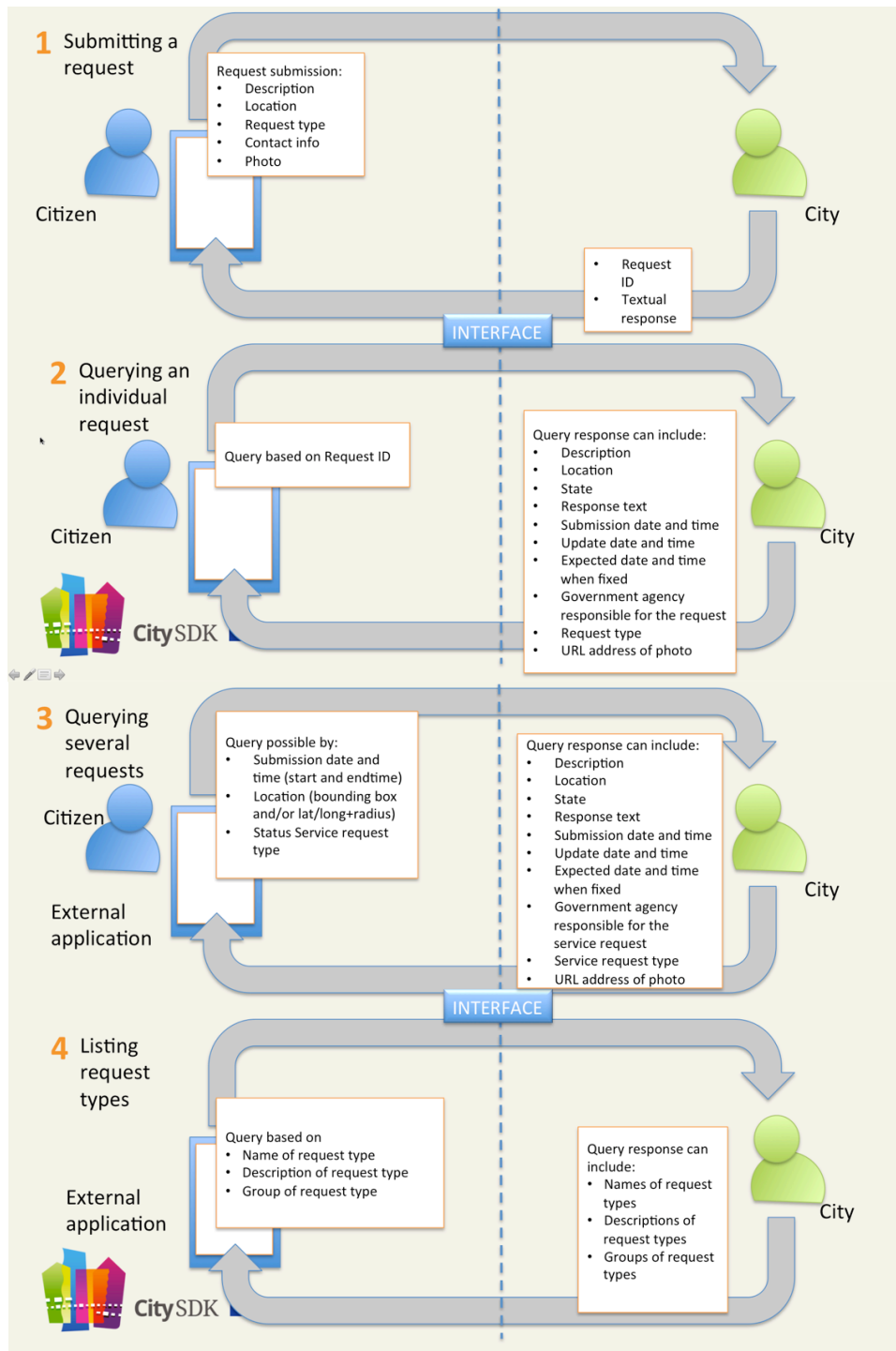
Illustrations on the next page show how and what information travels thru the interface from the citizen to the city and vice versa.

---

[1] https://docs.google.com/a/forumvirium.fi/spreadsheet/ccc?key=0AqvnokBCNLe5dGE5ak5yd21tajNHaTRVSU0tZ2RROHc#gid=0

[2] https://docs.google.com/a/forumvirium.fi/spreadsheet/ccc?key=0AqvnokBCNLe5dFJfNXNKUjlkdHE2S2pJTzNPZ1M5eGc#gid=0

1 Submitting a request

Citizen

Request submission:
- Description
- Location
- Request type
- Contact info
- Photo

- Request ID
- Textual response

City

INTERFACE

2 Querying an individual request

Citizen

CitySDK

Query based on Request ID

Query response can include:
- Description
- Location
- State
- Response text
- Submission date and time
- Update date and time
- Expected date and time when fixed
- Government agency responsible for the request
- Request type
- URL address of photo

City

3 Querying several requests

Citizen

External application

Query possible by:
- Submission date and time (start and endtime)
- Location (bounding box and/or lat/long+radius)
- Status Service request type

Query response can include:
- Description
- Location
- State
- Response text
- Submission date and time
- Update date and time
- Expected date and time when fixed
- Government agency responsible for the service request
- Service request type
- URL address of photo

City

INTERFACE

4 Listing request types

External application

CitySDK

Query based on
- Name of request type
- Description of request type
- Group of request type

Query response can include:
- Names of request types
- Descriptions of request types
- Groups of request types

City

Replication process

During the period from M12-M18 participation services for replication pilots was completed, and the WP3 specification work was finalized and documented in the first deliverable (*D3.1 Participation Pilot Application and its SDK components*).

The replication pilots had various options for setting up the interface to be utilised in their pilots. If the city chose not to implement the interface directly to their back-end system, they could either use an internal proxy or external message broker, like in the case of Amsterdam and Barcelona. It was also possible to utilise an existing solution by a service provider to offer the standard interface (Mark-a-spot, See-click-fix), but none of the replication pilots chose to do so.

For the replication pilots the Lead pilot team provided support during the project workshops and via online support activities, such as telcos and e-mails. Face-to-face visits were organized with the cities of Barcelona, Manchester and Lisbon, in order to assist with some concrete issues regarding the technical implementations and other emerging issues. The Lead Pilot team also developed materials and documentation for the replication cities in order to help them on replicating the CitySDK Issue reporting API in their home city. In the end, interface implementation testing was provided by the Lead Pilot team for those cities that already reached that point.

Pilots

In the WP 3 seven cities have implemented or will soon finalize the implementation of the CitySDK Issue Reporting API.

At M24 of the project, 4 cities are done with their implementations and three will soon be finalized. There are currently 10 apps utilizing these APIs and one of them is already utilizing both the Lamia and Helsinki endpoints. The WP Partners are aiming to increase this number in the last 6 months of the project.

Pilot

The pilot was done together Sanomat and their local news site Metro.fi. Metro has been running their Pitäiskö fiksata (fix my street) - service for some years. In the pilot the service was connected to city's endpoint. Now when Metro users report their issues to Metro site, the issue can be submitted to City's issue report management system. Metro service also shows city's response text and status to the feedback .
Internally, city's issue report management system was also integrated with Public Works department system. This allows forwarding the issue reports from the issue report management system to Public Works department case management system (Aspa).
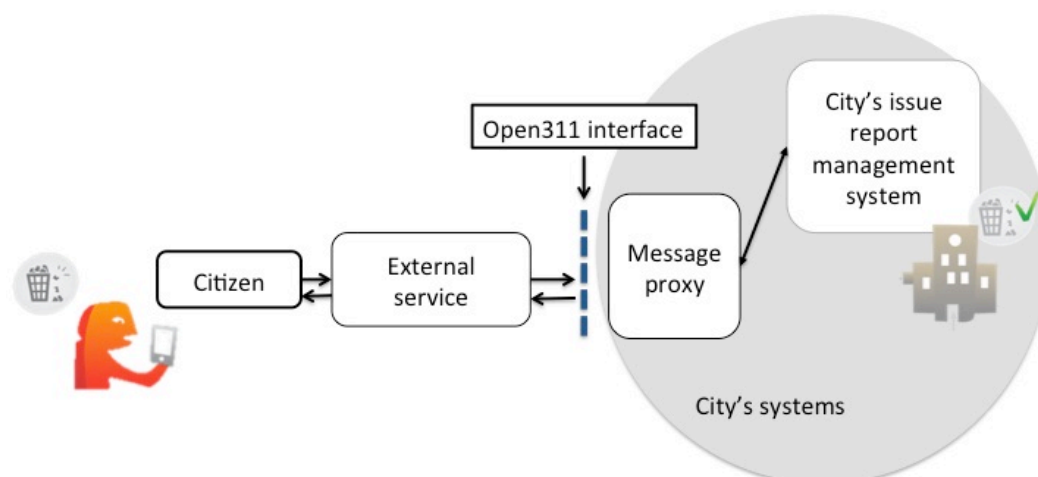
Pre-pilot Situation

City of Helsinki was developing a new issue report management system. Before the new system each individual departments handled feedback differently. There was no interface into any issue report management systems.

Endpoint and Back-end system

http://dev.hel.fi/apis/issuereporting

City's API is implemented as an independent web service (Message proxy) which connects to city's main issue report handling system called the issue report management system. Both message proxy and main issue report management systems are J2EE implementations.

9

Helsinki Fact file

How long it took to implement the API?	Approximately 3 man-months for the API and 1 man-month for internal integration
Costs for implementation	50k euros for API and 15k euros for integration
Services and apps created using the API	The pilot app was Metro's Pitäiskö fiksata (fix my street) -service. Other apps developed: • local news site Sanom.at (https://sanom.at/) • Korjaa kaupunki (http://korjaakaupunki.fi/) • Georeporter iPhone application (https://itunes.apple.com/us/app/georeporter/id487304759) Later also Android app is available Demo apps like Map viewer (http://dev.hel.fi/open311-test/mapviewer.html)
Code Libraries or other resources	There are several resources which can be used due to the compliancy with Open311. List of resource: http://wiki.open311.org/GeoReport_v2/Resources City of Helsinki used code library Three and also GeoReporter application for testing the API. Also some contributions were made to those libraries. • Metro used Three library in their service.
Data sources used in the pilot	It is possible to use city's service map data source to link issue reports to service map points of interest. Link to the data source: http://www.hel.fi/palvelukarttaws/rest/ver2_en.html For example, if an issue is related to children's playground, it can be linked to the corresponding service map point. It is also possible to use city's open street address data or API to include address information together with coordinate information into the issue report. Link to the data source: http://dev.hel.fi/apis/geocoder
Business models evolved	Korjaa kaupunki has started a membership fee based foundation for funding the service. Metro is based on ad sales.

Partners involved
City of Helsinki worked together with local companies Affecto and Keypro and made the API and required internal integration work. Sanoma Metro.fi developed Pitäiskö fiksata (fix my street) -service.

Contact info
Jaakko Rajaniemi, City of Helsinki, jaakko.rajaniemi@hel.fi, +358 40 5165931

Pilot

The pilot consists of creating an Open311-endpoint and a general broker to connect to different case management systems of Dutch municipalities (including the transformation of Open311-messages to the appropriate formats that can be handled by the case management system. Amsterdam worked together with GovUnited, a cooperation of Dutch municipalities. During the pilot we created an general Open311 endpoint, a broker and several, different connections to backend systems of municipalities.
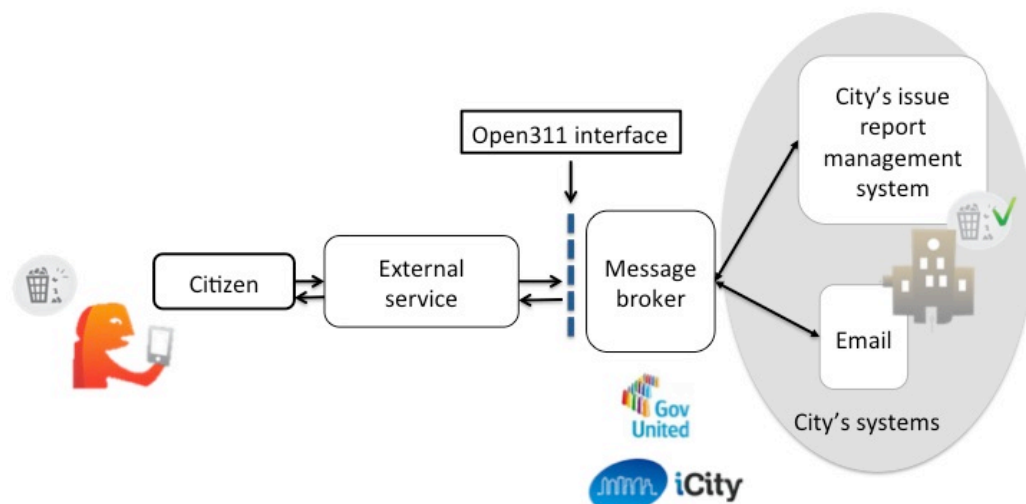
Pre-pilot Situation

Amsterdam has a (outdated) case management system for handling requests and a custom made presentation on a map. There is no API. There is one app (Opgeruimd) for iPhone but that app is not connected to the case management system en doesn't use an API.

Endpoint and Back-end system:

http://acc.dev.civity.nl/api/open311/services.xml?jurisdiction_id=0363&api_key=y8EQ4EUkTQSIZl1R62H5777g (Acceptance server (public API-key))

Several different case managementsystems: we connect to any case management system of a municipality, because we translate Open311-messages to the right format.

11

Amsterdam Fact File

How long it took to implement the API?	The API itself took 4-6 man weeks.
Costs for implementation	All activities (backend connections, app-development, API modifications, testing, different municipalities, etc.) cost between 40-60.000 euro.
Services and apps created using the API	Working Open311 service (acceptance and production server), Live documentation (Swagger). Twoo apps connected: Buurtvitaal (www.buurtvitaal.nl) and www.verbeterdebuurt.nl. At the end of 2013 four municipalities; in 2014 more will follow. There is a reusable, standards based platform available for all cities with full Open311 implementation.
Code Libraries or other resources	In progress
Data sources used in the pilot	-
Business models evolved	In progress

Partners involved
City of Amsterdam (www.amsterdam.nl)
GovUnited (www.govunited.nl)
Civity (www.civity.nl)

Contact info
Katalin Gallyas, City of Amsterdam,Department of Economic Affairs
Gallyas@ez.amsterdam.nl
+31(0)6-83209166

Smart Participation Barcelona

Pilot

The pilot for the city of Barcelona has deviated from the traditional connection of an API to a service into connecting to the system using the icity platform (http://icityproject.com/). Icity is an EU project that Barcelona is coordinating. This will not only give CitySDK Issue reporting API the local sustainability needed but it also introduces the Open311 standard to a platform that will connect to all of the cities backend proprietary systems.

Pre-pilot Situation

Barcelona has had in place an incident reporting system for the last 8 years. The IRIS system is very mature and continuously improving. It boasts hundreds of categories and all work orders though stored in the city councils repository are automatically routed to its corresponding contracted service.

Endpoint and Back-end system:

End point TBC, back-end Oracle Databases.
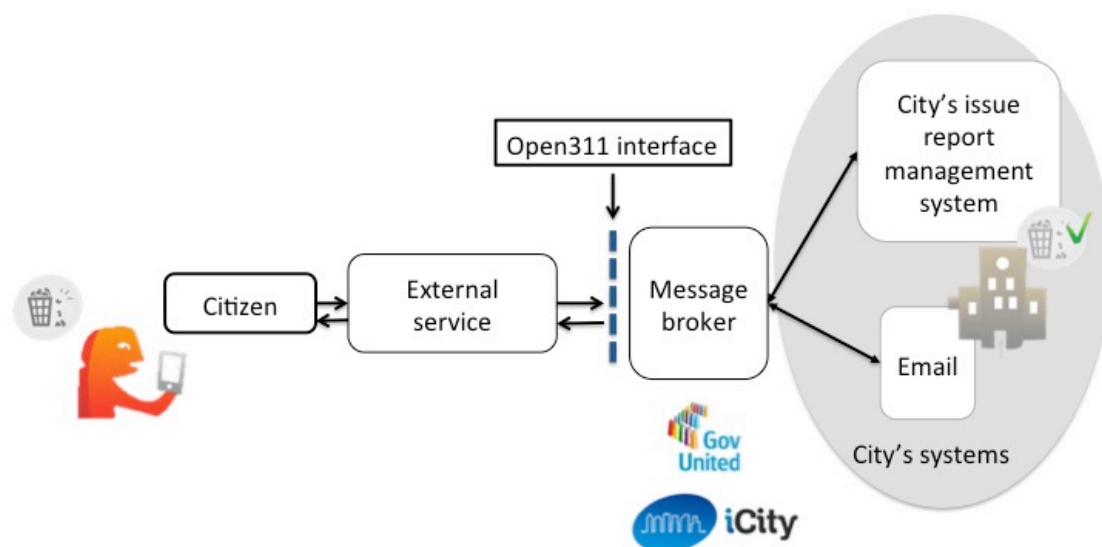
Barcelona Fact file

How long it took to implement the API?	Still implementing – Man weeks so far 11
Costs for implementation	10,000 +
Services and apps created using the API	0
Code Libraries or other resources	-
Data sources used in the pilot	-
Business models evolved	-

Partners involved

Ajuntament de Barcelona
www.bcn.cat
Abertis Telecom
www.abertistelecom.com

Contact info

Alex Serret, IMI City Council, aserretg@bcn.cat, 620851771

Smart Participation Lamia

Pilot

With the new implementation of CitySDK Issue Reporting API, it allows external applications to send issue reports to city's issue report management system. In pilot Android and Windows phone apps were developed which use the API. These apps are also cross city as they can be used to send reports to the issue report management system of City of Helsinki.
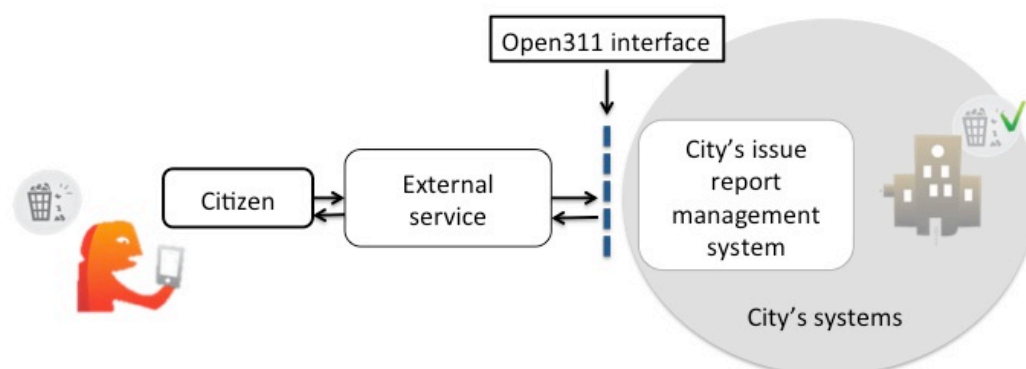
Pre-pilot Situation

Municipality of Lamia offers various services including possibility to submit service requests from its web site (http://services.lamia-city.gr). Before the start of the project there was already a web service running that was firstly consumed by an app that was being used only by municipality's staff and later the app was available for everyone. Web service was exposed through SOAP technology and can still be reached at:
http://84.205.233.188/workflowSVC/Workflow.svc

Endpoint and Back-end system:
https://participation.citysdk.lamia-city.gr/rest/open311/v1

Municipality's back-end system is based on a client-server windows application that accesses SQL SERVER database.

Lamia Fact File

How long it took to implement the API?	Municipality of Lamia 3 Man-Months Gnosis Computers 6 Man-Months
Costs for implementation	Municipality of Lamia 15.000 Euros (without dissemination costs) Gnosis Computers 25.800 Euros (without dissemination costs)
Services and apps created using the API	Apart from the implementation of the API that is hosted on municipality's infrastructure there are two apps developed that support the API, one for Android devices and one for Windows Phone : Google Play: https://play.google.com/store/apps/details?id=com.smarts.pta&hl=en Windows Phone: http://www.windowsphone.com/en-us/store/app/mts/d941bf31-f5b6-4bb1-bf06-a27b1987bbea Apps currently use Lamia's and Helsinki's endpoints.
Code Libraries or other resources	
Data sources used in the pilot	The only datasource used was municipality's database that supported the existing issue reporting system
Business models evolved	Models that based on motivation techniques: • Rewarding • Gamification • Crowd sourcing

Partners involved
Municipality of Lamia(www.lamia.gr)
GNOSIS Computers www.gnosis.com.gr

Contact info
Ioannis Sirakis, Municipality of Lamia, info@lamia-city.gr, +302231351018

George Koukas, Gnosis Computers, koukas@gnosis.gr, +302231020000

Pilot

We developed a webservice to comply with the CitySDK Issue reporting API, disseminating it through several events, such as OpenDataMakers and Blackberry Hackathon-Setúbal.
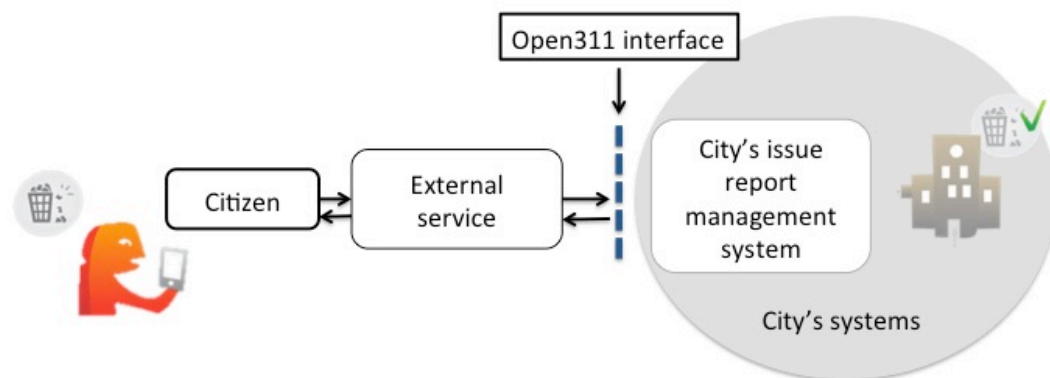
Pre-pilot Situation

Web application that accepts requests from outside Lisbon city hall (http://lxi.cm-lisboa.pt/lxi/?application=NaMinhaRua). Internal web application to treat requests and reply citizen (http://gopi.cm-lisboa.net).

Endpoint

http://open311.cm-lisboa.pt/GeoReport/discovery/discovery.xml

Back-end system:

GOPI http://gopi.cm-lisboa.net. Only internal access.

Lisbon Fact File

How long it took to implement the API?	2 Months
Costs for implementation	Estimated value around 43000 (internal)
Services and apps created using the API	IST is currently developing a multi-device application for Android. It enables users to access their issue reports from multiple devices and not only the one where they were submitted. The application is comprised of a website (beta version at https://tagus.inesc-id.pt/participation/) and an Android app. The app allows issues to be submitted and followed. It reports issue IDs to the website, which glues all devices together. All devices from the same user learn about the submitted issues from the website, enabling the use to consult the issues he submitted on any device. Users may also follow their submitted issues using the website. Both the website and android app are tentatively called FixEurope. They enable transparent access to several cities Open311 interfaces.
Code Libraries or other resources	-
Data sources used in the pilot	-
Business models evolved	-

Partners involved
City of Lisbon
ISA

Contact info
Mario Fernandes, Lisbon Municipality, mario.fernandes@cm-lisboa.pt
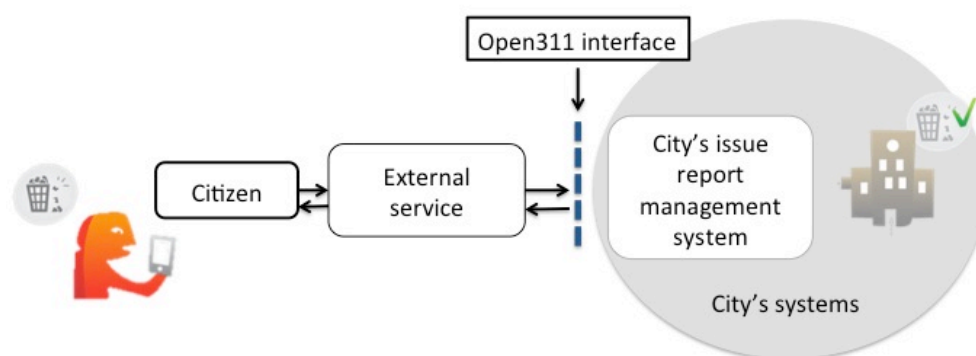
Pilot

Manchester implements a business pilot to enable businesses to better access council services and be put through to the correct department and to pilot functionality that may be later added via a business CRM. The pilot has been chosen to fill a gap in current provision via the website by providing a one-stop shop. It is also the aim to link the two pilots - Participation and Mobility - in Manchester through incorporating opendata sets relevant to business through this mechanism.

Pre-pilot Situation

There are many legacy systems. Recent redevelopment of website, and the implementation of a city CRM, for residents, using MicrosoftCRM. Individual online requests have seen an "email" system replaced with online forms which when integrated with the CRM can provide feedback on specific queries. 20% cut in local government grants has led to pressures on resources which has delayed the Manchester pilot as we identifiy key priorities for implementation that can reduce demand on call centre and provide better quality information.

Endpoint and Back-end system:

End-point TBC, Back-end Jadu Galaxies - Ask About Business website is hosted and uses a CMS developed by Jadu who run the city's

Manchester Fact File

How long it took to implement the API?	To be confirmed, but by end of February 2013. Likely to be a "scrum" development including 3 people x 6 weeks = 18 person weeks for implementation and pilot testing.
Costs for implementation	TBC
Services and apps created using the API	TBC
Code Libraries or other resources	TBC
Data sources used in the pilot	TBC: Likely to include State of the City data; Library locations; Car Park locations
Business models evolved	We are taking advantage of a Manchester Libraries project to develop a business support service across 10 local authorities across Greater Manchester and a data synchronisation project funded by TSB in the UK, which will synchronise dataset release.

Partners involved

Manchester City Council Regeneration Team
(http://www.manchester.gov.uk/business)
Manchester Libraries Business Service http://www.askaboutbusiness.org/;
Manchester Digital (www.manchesterdigital.com) (trade association);
FutureEverything (www.futureeverything.org)

Contact info

Adrian Slatcher / Manchester City Council / a.slatcher@manchesterdda.com
+44(0)1612344029

OpenLabor

Pilot
Openlabor is a set of API that allows for searching, submitting and commenting reports about jobs and training opportunities in the Province of Rome.

Pre-pilot Situation
Jobs and vocational training data
OpenData on jobs and vocational training existed in the CKAN repository of the Province of Rome, and a web search form was available (only for jobs), but there were no apps or APIs that allows to comment and submit new report, nor to cross jobs with training
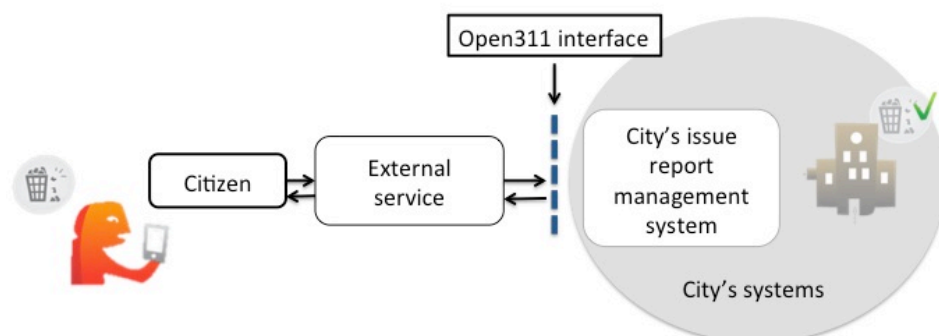
Endpoint and Back-end system:
http://openlabor.lynxlab.com/api/v1/doc/openlaborAPIspecification.html

Openlabor retrieves XML files from http://www.opendata.provincia.roma.it/ CKAN repository daily, then it stores them in a MySQL database.
We make use of Italian standard professional taxonomy (ISTAT CP2011, http://cp2011.istat.it/), imported in our DB
We also use Python NLTK library (http://nltk.org/) to extract linguistic features from texts.
We use MultiWordNet ontology (http://multiwordnet.fbk.eu/english/home.php) to transparetly translate English names of jobs (if submitted by non-italian citizens) into the corresponding Italian names used in the ISTAT taxonomy.
Some Php classes have been built to implement the CitySDK APIs.

Rome Fact File 1

How long it took to implement the API?	36 man-weeks
Costs for implementation	40320 euros (staff costs)
Services and apps created using the API	Openlabor Roma https://play.google.com/store/apps/details?id=it.lynx.openlabor&hl=en This Android app allows also to set and automatic daily search and to share the data with friends A web app interface is also available at the URL http://openlabor.lynxlab.com/ The web app allows to locate on a map the job/course selected and to contact the provider
Code Libraries or other resources	OpenLabor https://github.com/lynx-lab/openlabor/tree/master/modules
Data sources used in the pilot	Openlabor: Jobs http://co.provincia.roma.it/ido/XmlOpenData/Preselezioni.xml Training: http://85.18.173.117/Dip3OpenData/CorsiFormazioneNonFinanziata.xml
Business models evolved	The business models for openlabor API and apps are several:- integrating the system within the public services for job and training search support and guidance - licencing the use of the APIs to private search engines - providing free apps with advertising from verified companies - giving PA access to a dashboard with real time statistics to help the redesign vocational training plans and active labour policies

Partners involved
Provincia di Roma
http://www.provincia.roma.it/
Lynx s.r.l.
http://www.lynxlab.com
Cineca
http://www.cineca.it/

Contact info
Marcello Zini, Provincia di Roma, m.zini@provincia.roma.it, +390667667318
Stefano Penge, Lynx s.r.l., steve@lynxlab.com, Tel +39 06 97658937
Federico Capoano, Cineca, f.capoano@cineca.it, +39 366425380

Smart Participation Rome 2

VoteSpot (proposed name)

Pilot

VoteSpot is a web application based on Nodeshot, an open source platform that aims to easen up the development of community-led georeferenced data. It comprises several components of which the most important for this pilot are:
- A web interface for end-users
- Open311 compatible RESTful API for developers
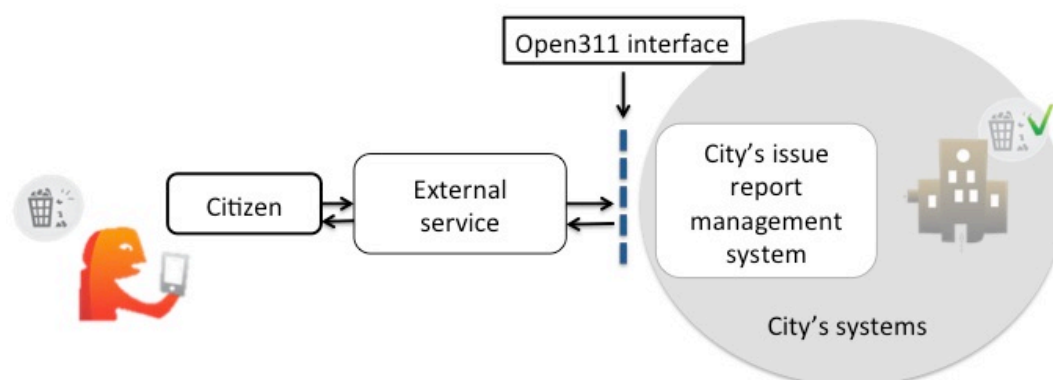- Administration interface
- Statistical analytics

Pre-pilot Situation

Provincia di Roma created a large free wifi network and publishes the location of the hotspots on the CKAN portal, but currently a system to efficiently gather, classify and analyze the requests for new access points is not available

Endpoint and Back-end system

Nodeshot (alpha): https://nodeshot.publicwifi.it/api/v1/docs/ (open311 API to be finished by the next CitySDK workshop in February)

Nodeshot is a web application built with the django framework which provides many extensible features for crowdsourcing of georeferenced data.

Rome 2 Fact File

How long it took to implement the API?	Not finished yet
Costs for implementation	Not finished yet
Services and apps created using the API	Pilot not launched yet
Code Libraries or other resources	Nodeshot https://github.com/nemesisdesign/nodeshot
Data sources used in the pilot	Wifi hotspots of different cities: https://nodeshot.publicwifi.it/api/v1/layers/
Business models evolved	The business models for VoteSpot are: - integration with the existing institutional web portals of Provincia di Roma as the front end to deliver a better service to citizens willing to use the city wifi network - integration with the existing backoffice workflow of Provincia di Roma to speed up the process of managing the locations of the wifi hotspot to understand the needs of citizens for new hotpots and take prompt action

Partners involved

Provincia di Roma
http://www.provincia.roma.it/
Lynx s.r.l.
http://www.lynxlab.com
Cineca
http://www.cineca.it/

Contact info

Marcello Zini, Provincia di Roma, m.zini@provincia.roma.it, +390667667318

Stefano Penge, Lynx s.r.l., steve@lynxlab.com, Tel +39 06 97658937

Federico Capoano, Cineca, f.capoano@cineca.it, +39 366425380

Conclusion

Smart Participation SDK component, the interface, was specified through the collaborative requirement definition process in late 2012. The Lead Pilot used this interface and now the Smart Participation concept has been expanded to Amsterdam, Lamia, Lisbon and Rome. Barcelona and Manchester will soon follow with the launch of their APIs.

These partner cities have enabled (or will soon do so) the two-way issue-reporting channel for their citizens. All of them had an opportunity to develop the interface according to their special needs but mainly they stuck to the jointly specified API. The next step is to engage developers in the partner cities to boost the development of innovative participation apps that will save not only time of the users but increase efficiency in the city side as well.