CitySDK Participation

Nodeshot

A crowdsourcing platform designed for communities that will change our future

City Service Development Kit

CINECA

Who am I?

Federico Capoano

Web Designer & Web Developer

Ninux hacktivist

@nemesisdesign on twitter and github

Nodeshot developer

OpenWISP developer

What is CitySDK?

Development toolkit for the city services

City Service Development Kit

CINECA

Goals of CitySDK

1. Helping cities to open their data

Goals of CitySDK

2. Giving developers the tools they need to build apps

Goals of CitySDK

3. Reuse apps and services across European cities

Goals of CitySDK

4. Create business opportunities for SMEs (Small & Medium Enterprises)

Ok but in practice what do you do?

I know you geeks want to skip buzzwords and get to the technical stuff

Municipalities + SMEs

First cities partner up with SMEs

Municipalities: data

Municipalities put effort in retrieving data to release with an open license

Commons API

SMEs develop **common APIs** based on existing standards to use the data

Open Data + API = devs

Data and **APIs** are given to developers

Result: reusable services and apps

European cities can then replicate an useful & innovative service with low effort and costs

City Service Development Kit

CINECA

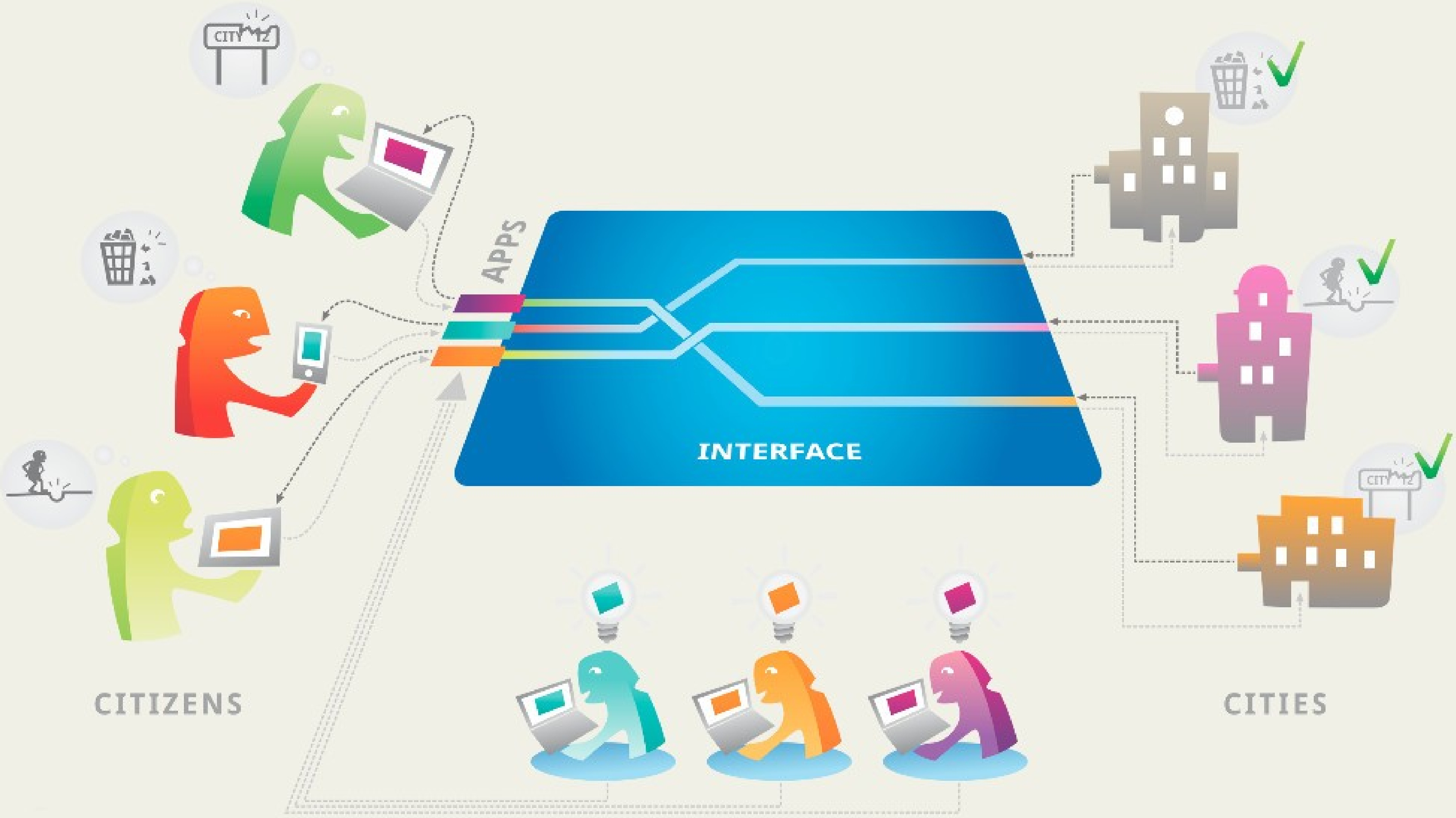
Main work packages

WP3: smart participation

WP4: smart mobility

WP5: smart tourism

Participation of citizens



What is the relation?

How did I end up working on Nodeshot for this EU project?

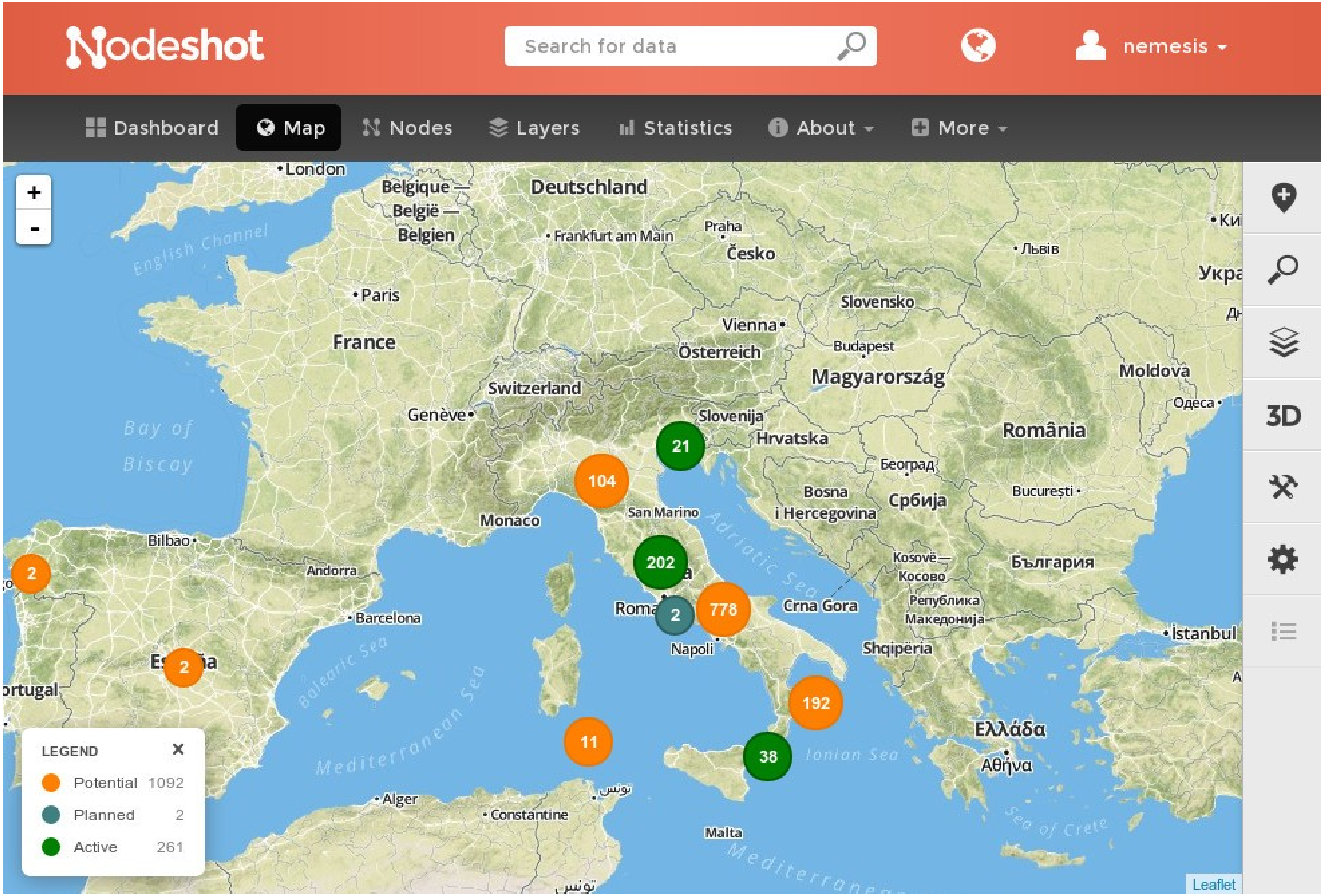
Votaspot Pilot

PROVINCIA
DI ROMA

City Service Development Kit

CINECA

Enter Nodeshot



What is nodeshot?

Modular web application for crowdsourcing georeferenced data

Aims to be flexible and exstensible

Achievements?

What has been achieved so far?

Generic

Core objects aims to be generic

Generic: geography

Node: any geometry is ok

PostGIS geometry

Generic: extensible

Customize nodes by using PostgreSQL HSTORE

(django-hstore)

https://github.com/djangonauts/django-hstore

Generic: layers

Nodes can be grouped in Layers

Interoperable

We need these kind of applications to speak to one another

Interoperable

We need these kind of applications to speak to one another

Interoperability module

Layers can be flagged as “external”

Interoperability module

An external layer can be configured to use a “synchronizer”

Interoperability module

Synchronizers can follow different approaches:

Interoperability module

1. Pull data periodically

Interoperability module

2. Push data on add, edit, delete

Interoperability module

3. mix of the two: pull periodically, push changes back

Interoperability module

4. on the fly: take data via HTTP and translate it to formats used by Nodeshot (GeoJSON and JSON)

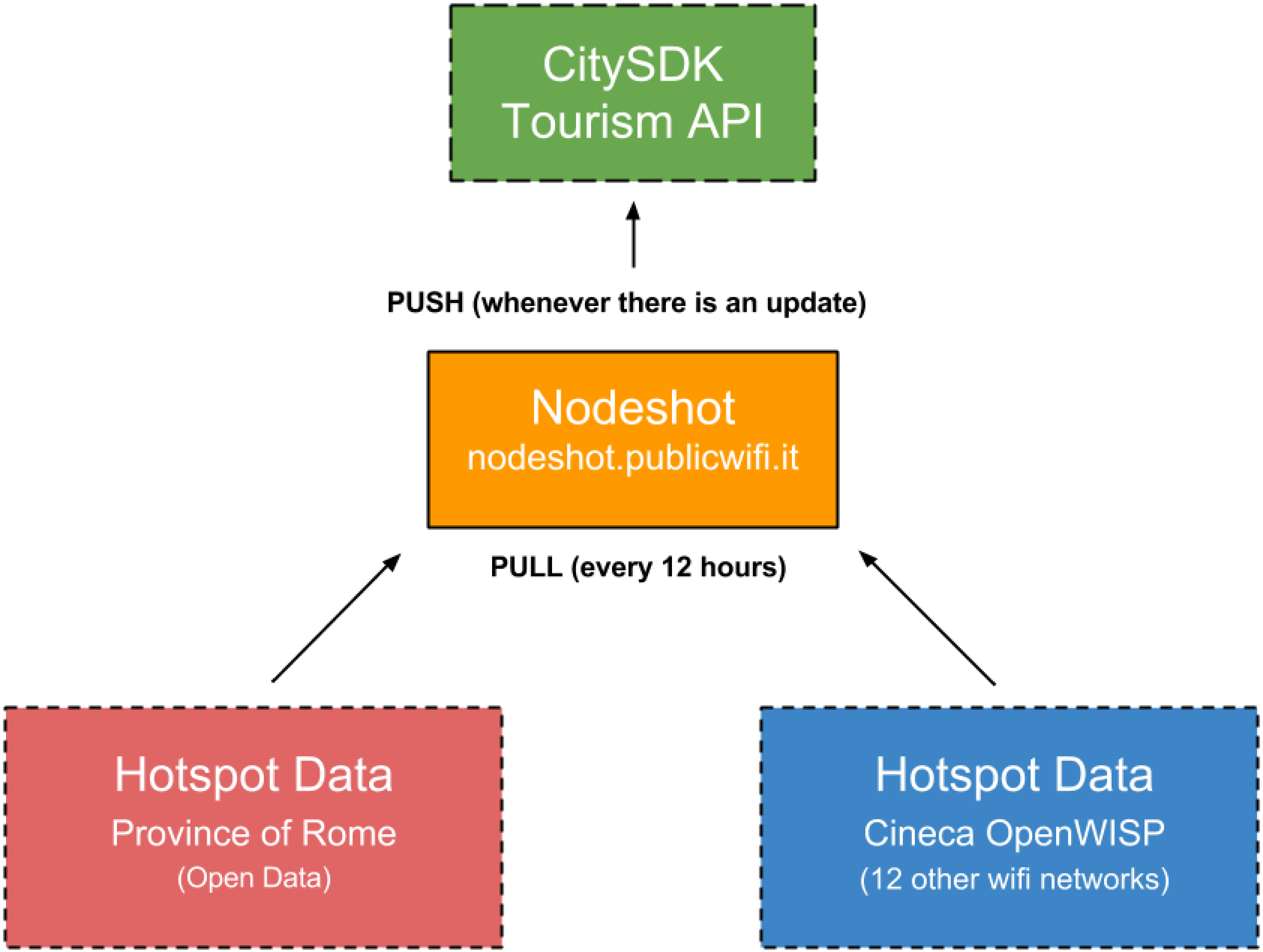
Interoperability module

5. extensible: write your own synchronizers

Interoperability module

Pratical use cases?

CitySDK was a really good chance to use and improve it!



CitySDK Tourism API

PUSH (whenever there is an update)

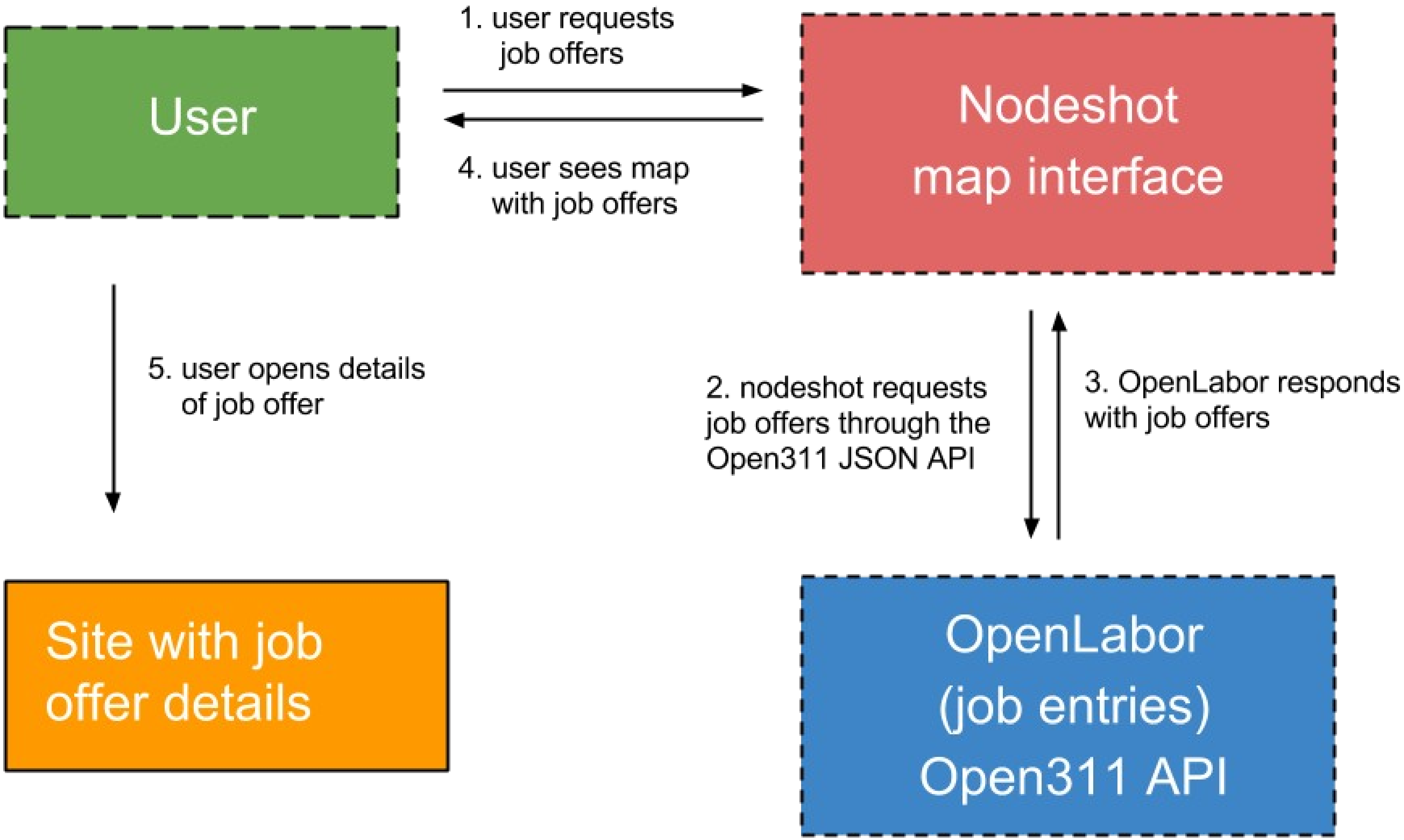
Nodeshot
nodeshot.publicwifi.it

PULL (every 12 hours)

Hotspot Data
Province of Rome
(Open Data)

Hotspot Data
Cineca OpenWISP
(12 other wifi networks)

Check out map of data: http://citysdk.inroma.roma.it/wp5-pilot/



User

1. user requests job offers

Nodeshot map interface

4. user sees map with job offers

5. user opens details of job offer

2. nodeshot requests job offers through the Open311 JSON API

3. OpenLabor responds with job offers

Site with job offer details

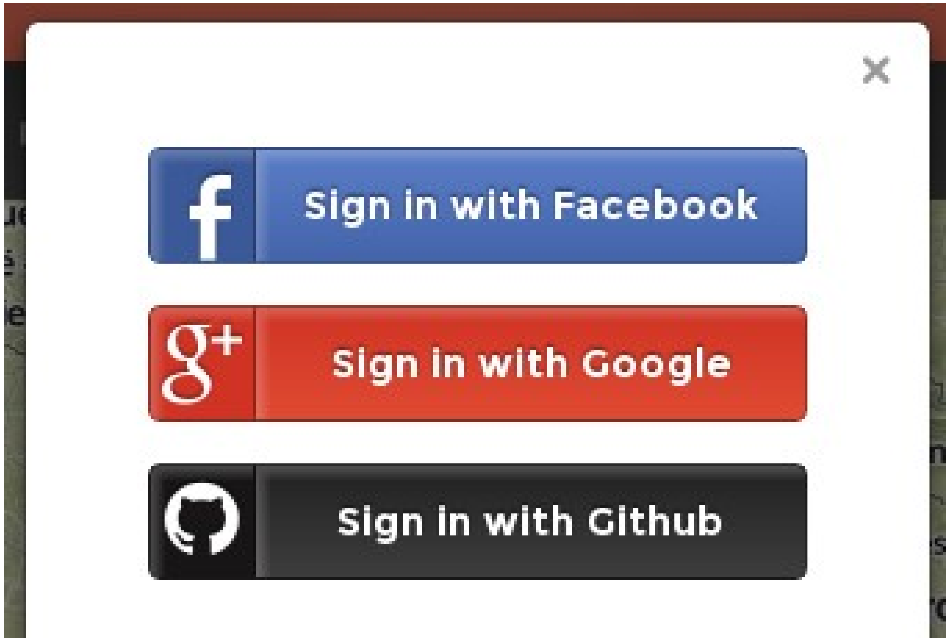
OpenLabor (job entries) Open311 API

Love Your Community

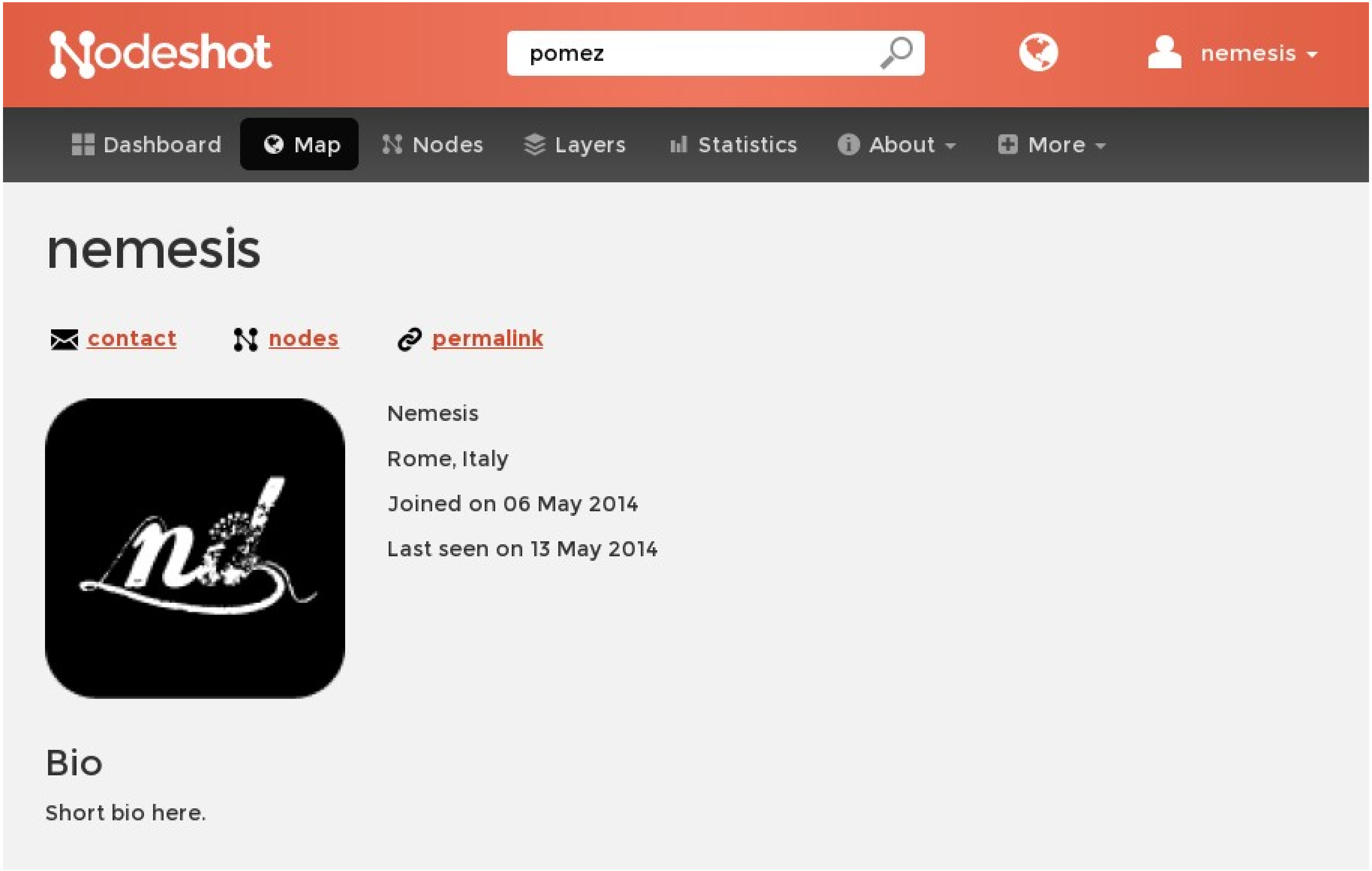
People need to communicate

And easily see what is happening

Social Login



Profiles



nemesis

contact nodes permalink

Nemesis

Rome, Italy

Joined on 06 May 2014

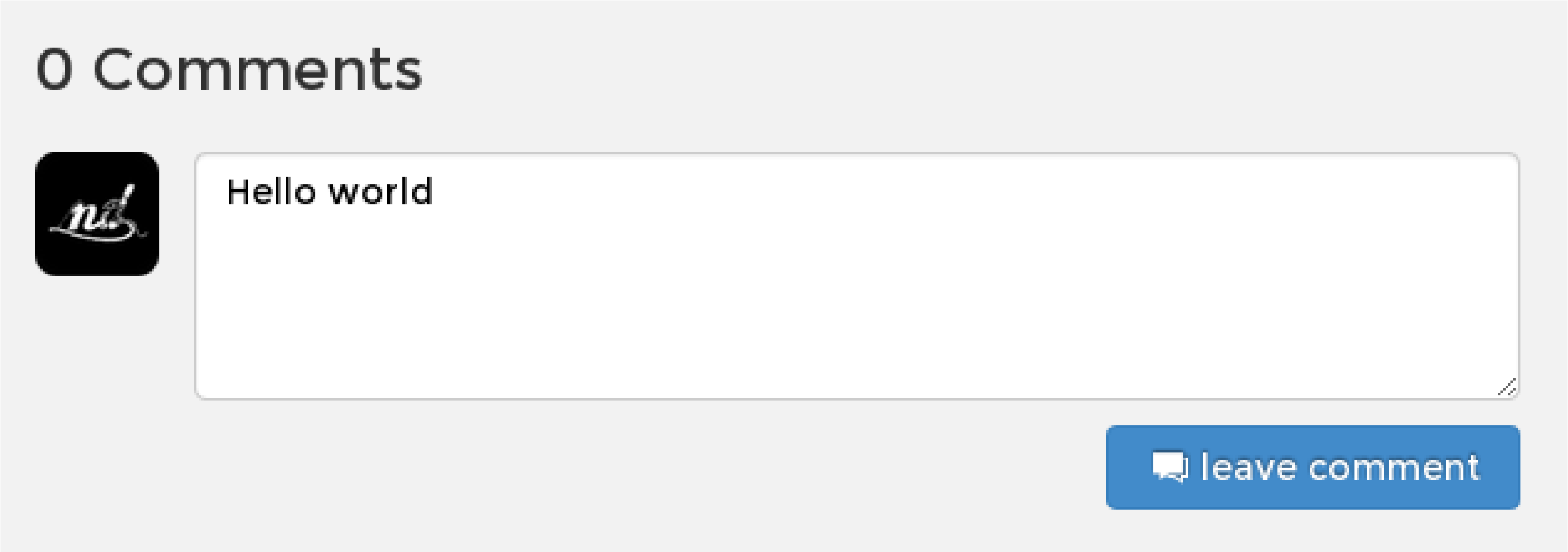
Last seen on 13 May 2014

Bio

Short bio here.

Comments

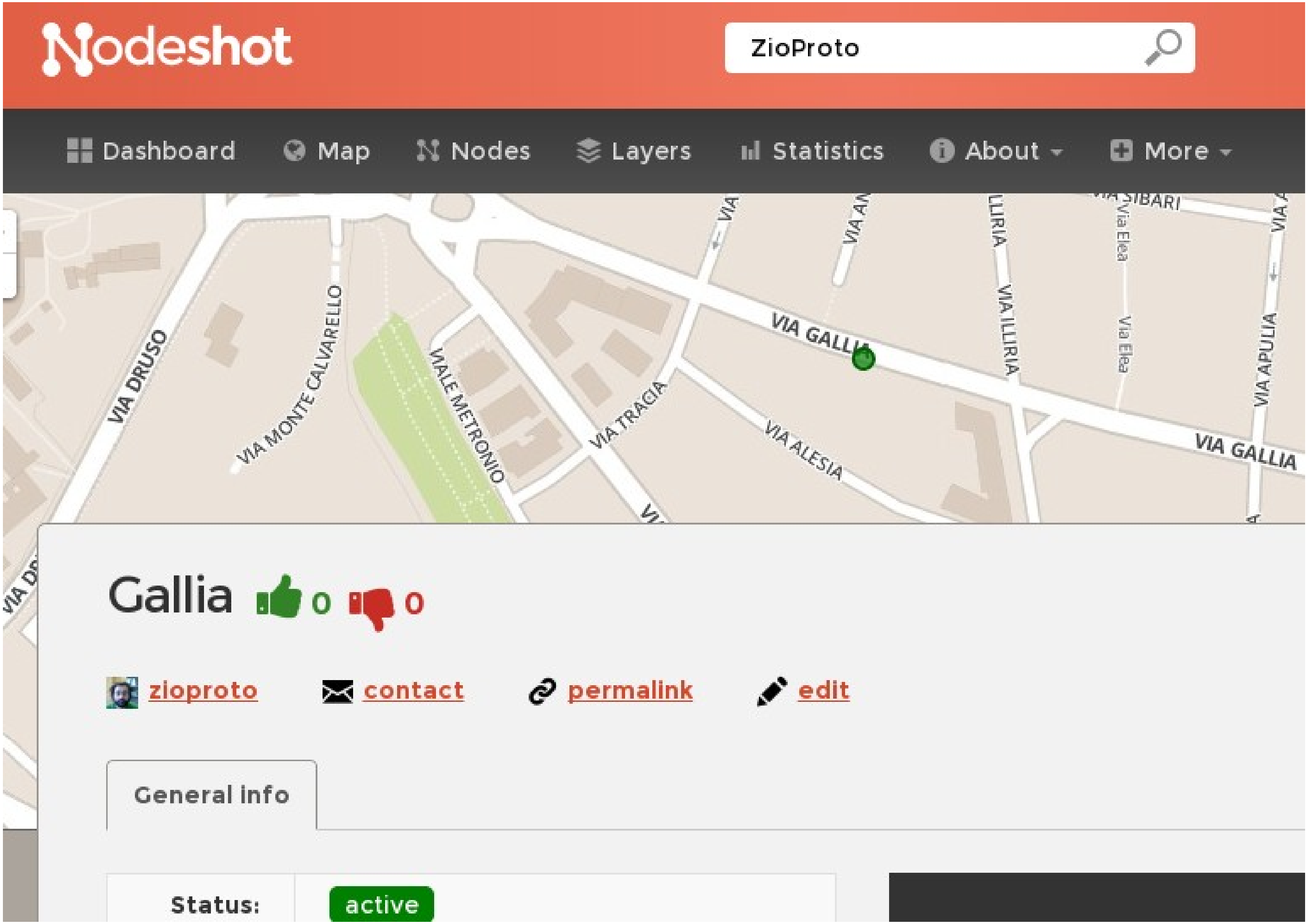
Leave comments on nodes



0 Comments

Hello world

Like or dislike



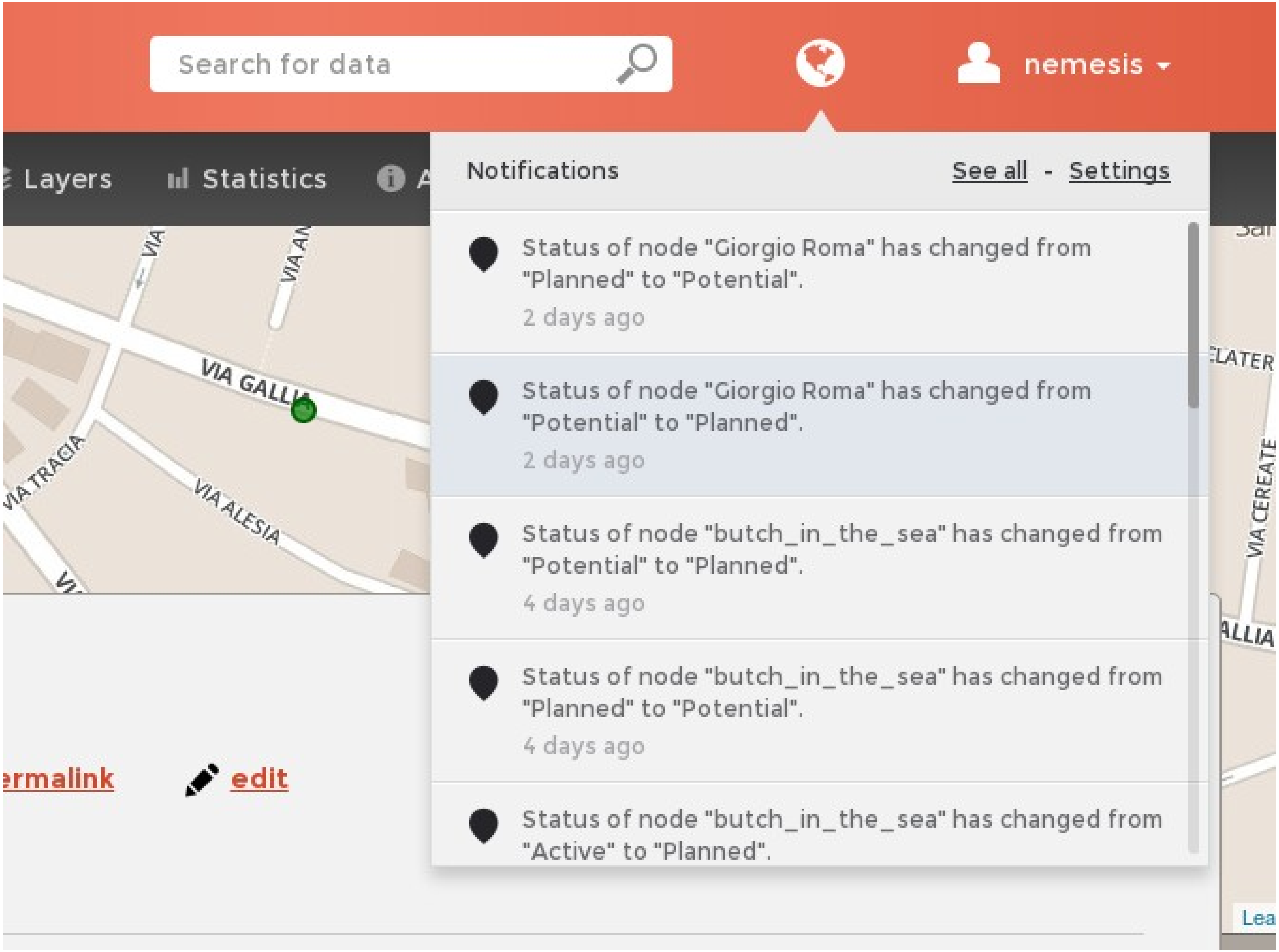
Gallia 👍 0 👎 0

zioproto ✉ contact permalink edit

General info

Status: active

Notifications



Notifications See all - Settings

Status of node "Giorgio Roma" has changed from "Planned" to "Potential".
2 days ago

Status of node "Giorgio Roma" has changed from "Potential" to "Planned".
2 days ago

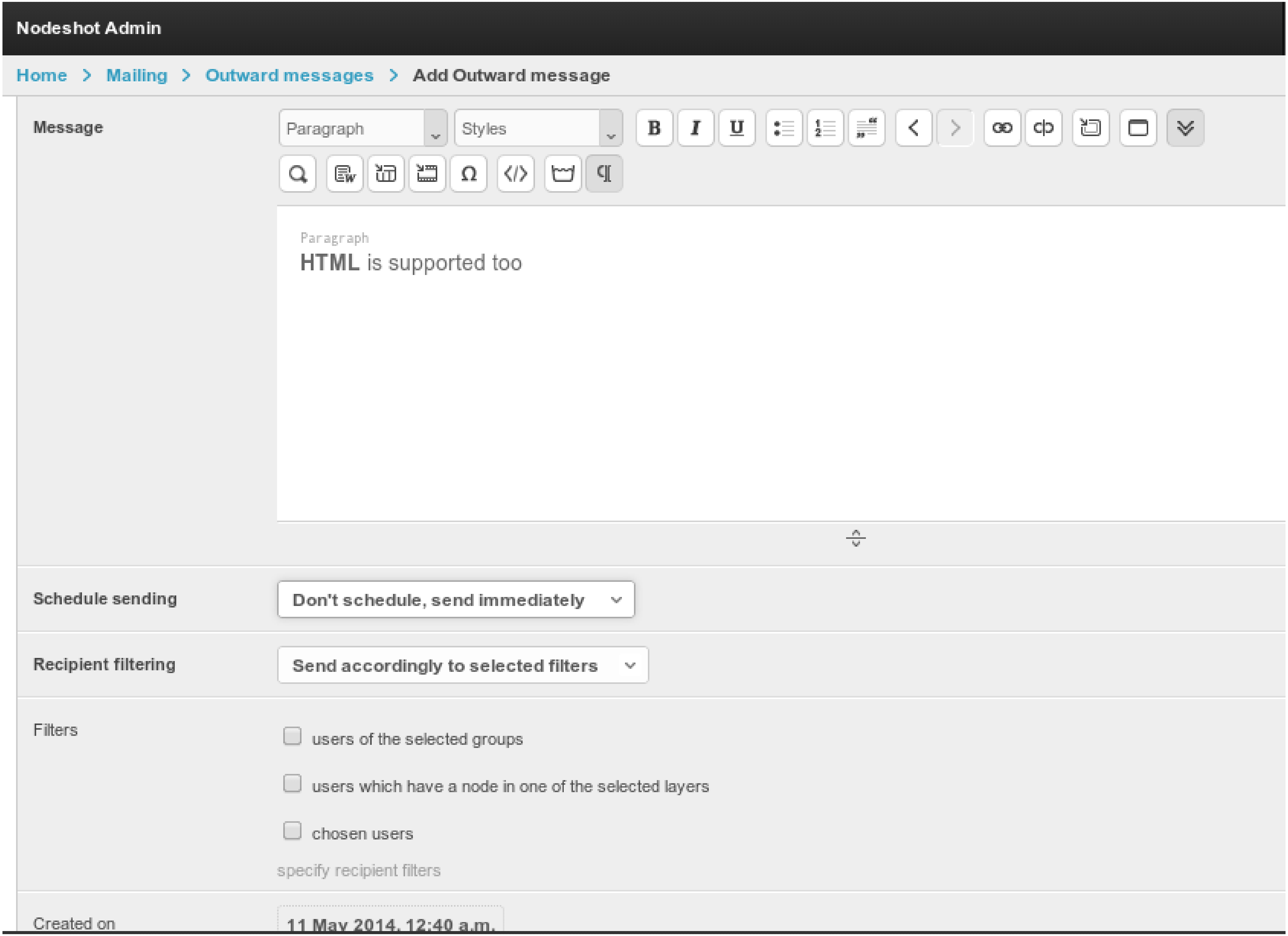
Status of node "butch_in_the_sea" has changed from "Potential" to "Planned".
4 days ago

Status of node "butch_in_the_sea" has changed from "Planned" to "Potential".
4 days ago

Status of node "butch_in_the_sea" has changed from "Active" to "Planned".

Mailing app

Send mails to members of your community without hacking scripts



Nodeshot Admin

Home › Mailing › Outward messages › Add Outward message

Message

Paragraph

HTML is supported too

Schedule sending: Don't schedule, send immediately

Recipient filtering: Send accordingly to selected filters

Filters

□ users of the selected groups

□ users which have a node in one of the selected layers

□ chosen users

specify recipient filters

Created on: 11 May 2014, 12:40 a.m.

Contact app

Users can get in touch with...

- Other nodes
- Other users
- Layer admins

Access Levels

In some cases we need to hide sensitive data.

Default:

public, registered, community, trusted

Based on existing standards

Let's try to not reinvent the wheel

Based on existing standards

GeoJSON API

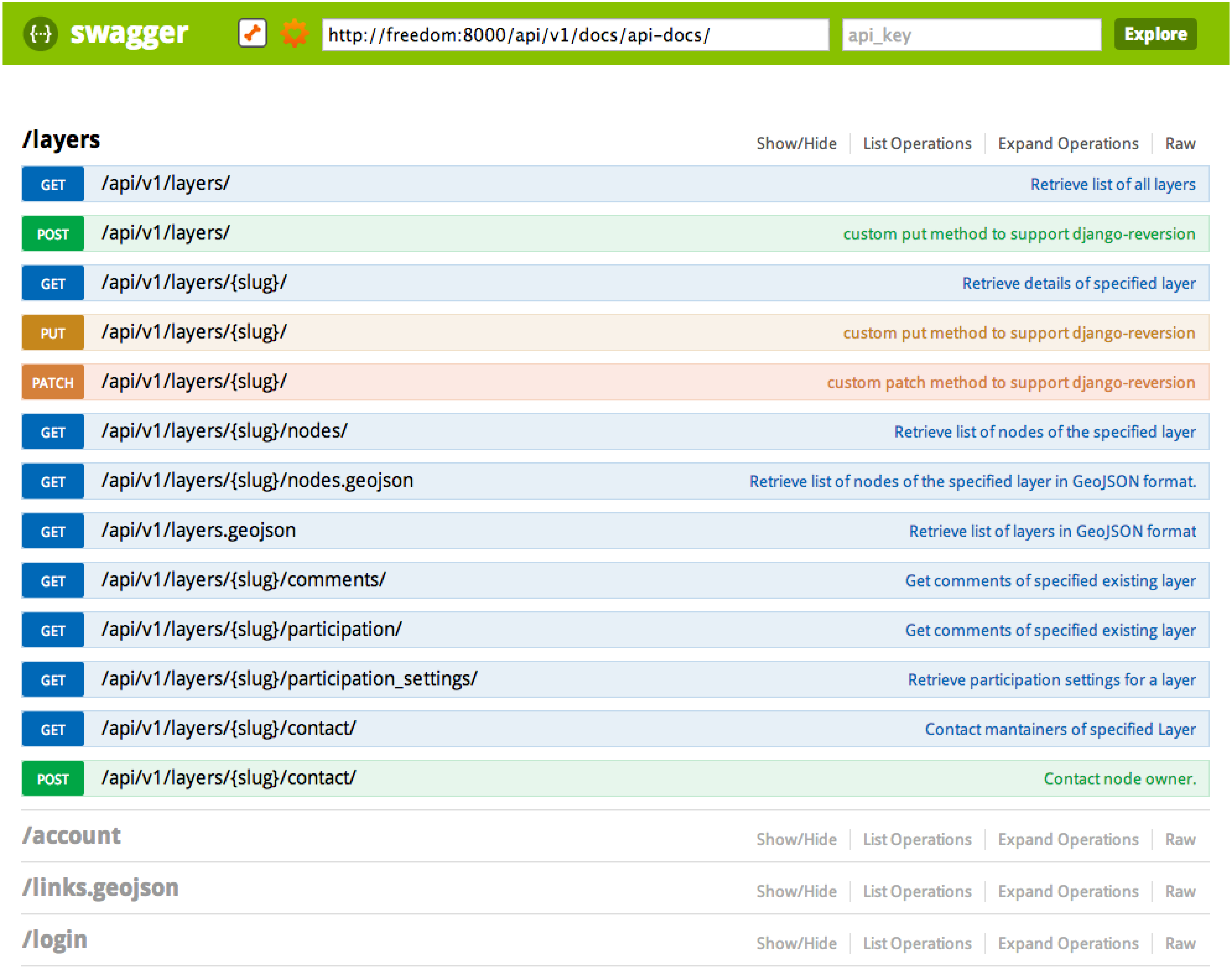
Based on existing standards

Open311 API

(Issue Reporting)

Based on existing standards

REST Swagger API docs



http://freedom:8000/api/v1/docs/api-docs/

/layers

GET	/api/v1/layers/	Retrieve list of all layers
POST	/api/v1/layers/	custom put method to support django-reversion
GET	/api/v1/layers/{slug}/	Retrieve details of specified layer
PUT	/api/v1/layers/{slug}/	custom put method to support django-reversion
PATCH	/api/v1/layers/{slug}/	custom patch method to support django-reversion
GET	/api/v1/layers/{slug}/nodes/	Retrieve list of nodes of the specified layer
GET	/api/v1/layers/{slug}/nodes.geojson	Retrieve list of nodes of the specified layer in GeoJSON format.
GET	/api/v1/layers.geojson	Retrieve list of layers in GeoJSON format
GET	/api/v1/layers/{slug}/comments/	Get comments of specified existing layer
GET	/api/v1/layers/{slug}/participation/	Get comments of specified existing layer
GET	/api/v1/layers/{slug}/participation_settings/	Retrieve participation settings for a layer
GET	/api/v1/layers/{slug}/contact/	Contact mantainers of specified Layer
POST	/api/v1/layers/{slug}/contact/	Contact node owner.

/account

/links.geojson

/login

Modular

Each module should do one things and do it well (in theory)

Modular

It should be possible to override default settings

Modular

It should be possible to alter behaviour of the software to adapt it to specific needs

Modular

Interesting modules can be separated into separate projects (eg: Netengine)

Not only wifi stuff

Not designed specifically for wifi.

Vacant spaces, City Issues, ecc.

Any geographic data!

Sustainable business model

Funding from:

Votaspot Pilot (CitySDK)

Existing Cineca WI-FI Networks

CityServiceDevelopmentKit

CINECA

Interesting future

OpenWISP Geographic Monitoring replacement

Interesting future

OpenWISP Manager replacement

Interesting future

Sensor data visualization

Standing on the shoulders of giants

Built on stable, modern, open source technologies

Python

Python 2.6 & 2.7

Python 3 coming in the future

Django 1.6

Django is used also by:

Instagram

Pininterst

Disqus

Celery (background queue)

Celery is used also by:

Mozilla

Instagram

AdRoll

RESTful API

JSON HTTP RESTful API

Supports GeoJSON

Django-rest-framework

Used by: **Mozilla, EventBrite**

Performant web server stack

Default deployment:

Nginx (web server)
UWSGI (application server)
Tornado (websockets)

Redis

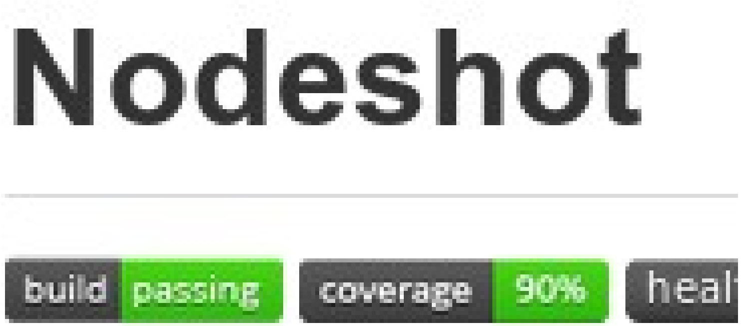
Redis for caching & queue processing

Redis is used by:

Twitter
Github
StackOverflow
Many more

Unit tested

90% test coverage



Nodeshot

build passing coverage 90% heal

User testing planned

User testing for Votaspot pilot
(June / July 2014)

City Service Development Kit

CINECA

User testing planned

Continous testing, improvement and delplooyment on

test.map.ninux.org

City Service Development Kit

CINECA

Auto Install & Deploy

Automatic install & deploy procedure for Ubuntu 12 LTS

Links

Repo:
https://github.com/ninuxorg/nodeshot

Docs:
http://nodeshot.readthedocs.org/en/latest/

Mailing list:
http://ml.ninux.org/mailman/listinfo/nodeshot

Any Question?

Thanks!

This huge work was also possible thanks to:

PROVINCIA
DI ROMA

City Service Development Kit

CINECA