Interoperability + Diversity == Growth

WIRELESS BATTLE OF THE MESH v8
MARIBOR

CINECA

OpenWISP

CONFINE
CONNECTED

Overcoming silos

What is **NetJSON**?

data interchange format
designed for networking software

based on JSON
JavaScript Object Notation
(RFC7159)

Dynamic routing protocols

(eg: olsrd, batman-adv)

Network databases

(A.K.A. node-db)

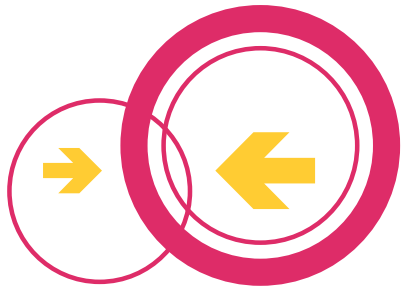
Monitoring tools

Firmwares / OS

(eg: OpenWRT, Raspbian)

What kind of networks?

Community networks!



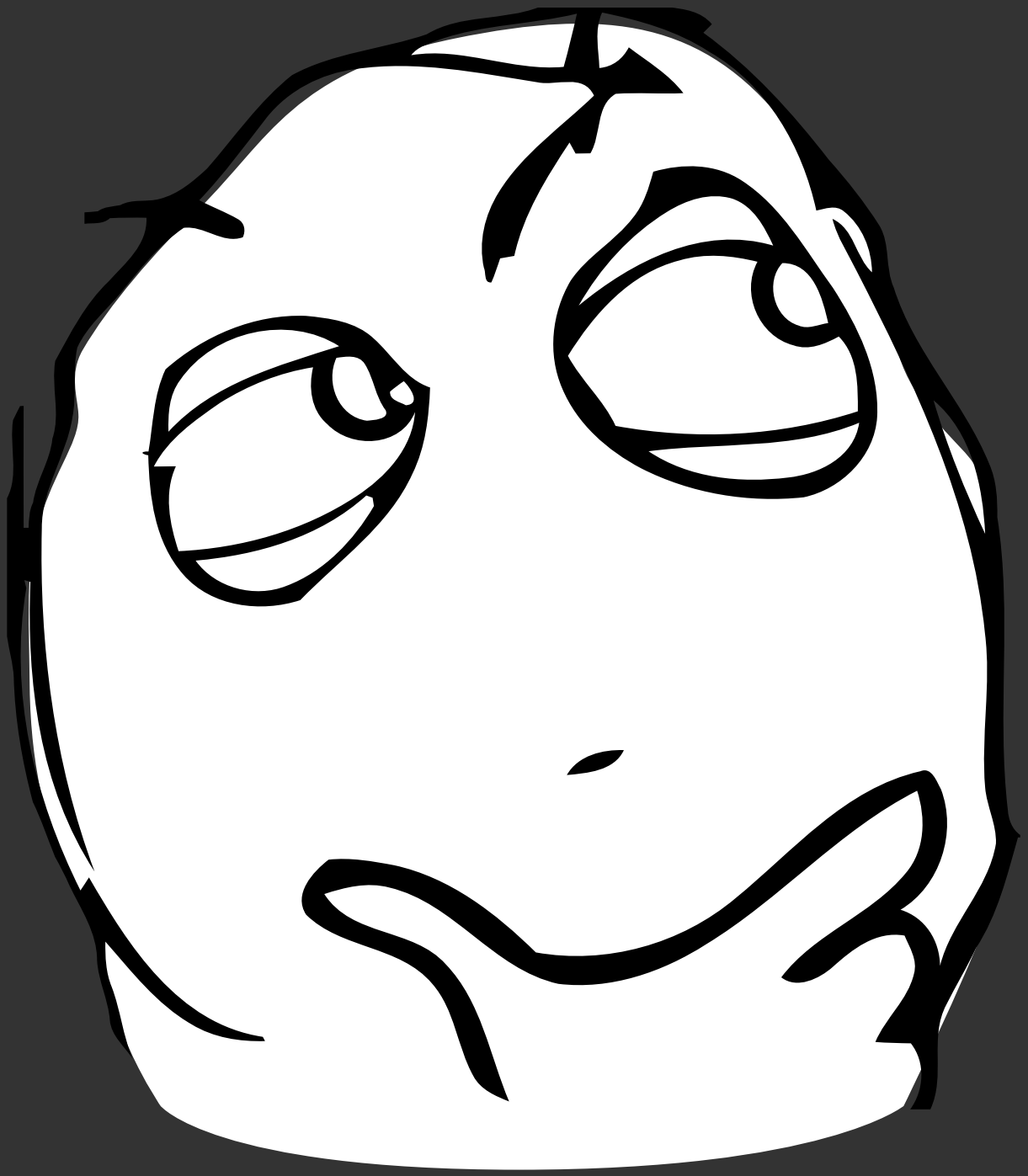
guifi·net

Also...

Municipal wifi

Research (eg: confine)

ISP (innovative ones)



WTF are you talking about?

Show us some examples!

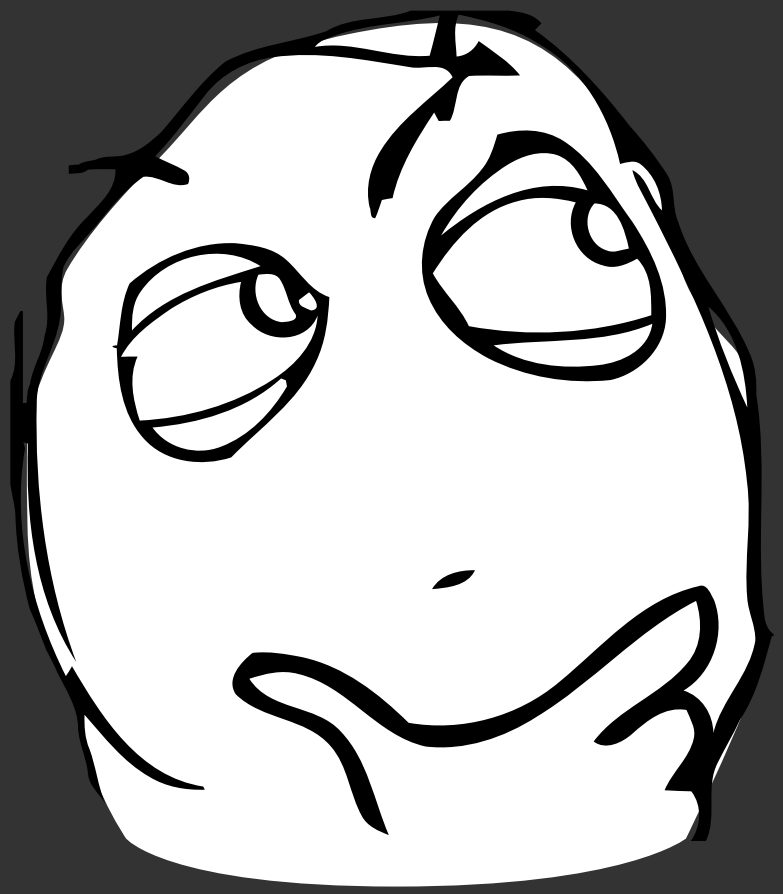
“NetworkGraph”

https://github.com/interop-dev/json-for-networks/blob/master/examples/network-graph.json

“DeviceConfiguration”

https://github.com/interop-dev/json-for-networks/blob/master/examples/device-configuration.json

Where did it came from?



A bit of history: 2013-2014

Working with GeoJSON quite a bit...

What's GeoJSON

Geospatial data interchange format

geojson.org

```
{
    "type": "Feature",
    "geometry": {
        "type": "Point",
        "coordinates": [125.6, 10.1]
    },
    "properties": {
        "name": "Dinagat Islands"
    }
}
```

GeoJSON allows different GIS Libraries to interoperate

GeoJSON > GEOS

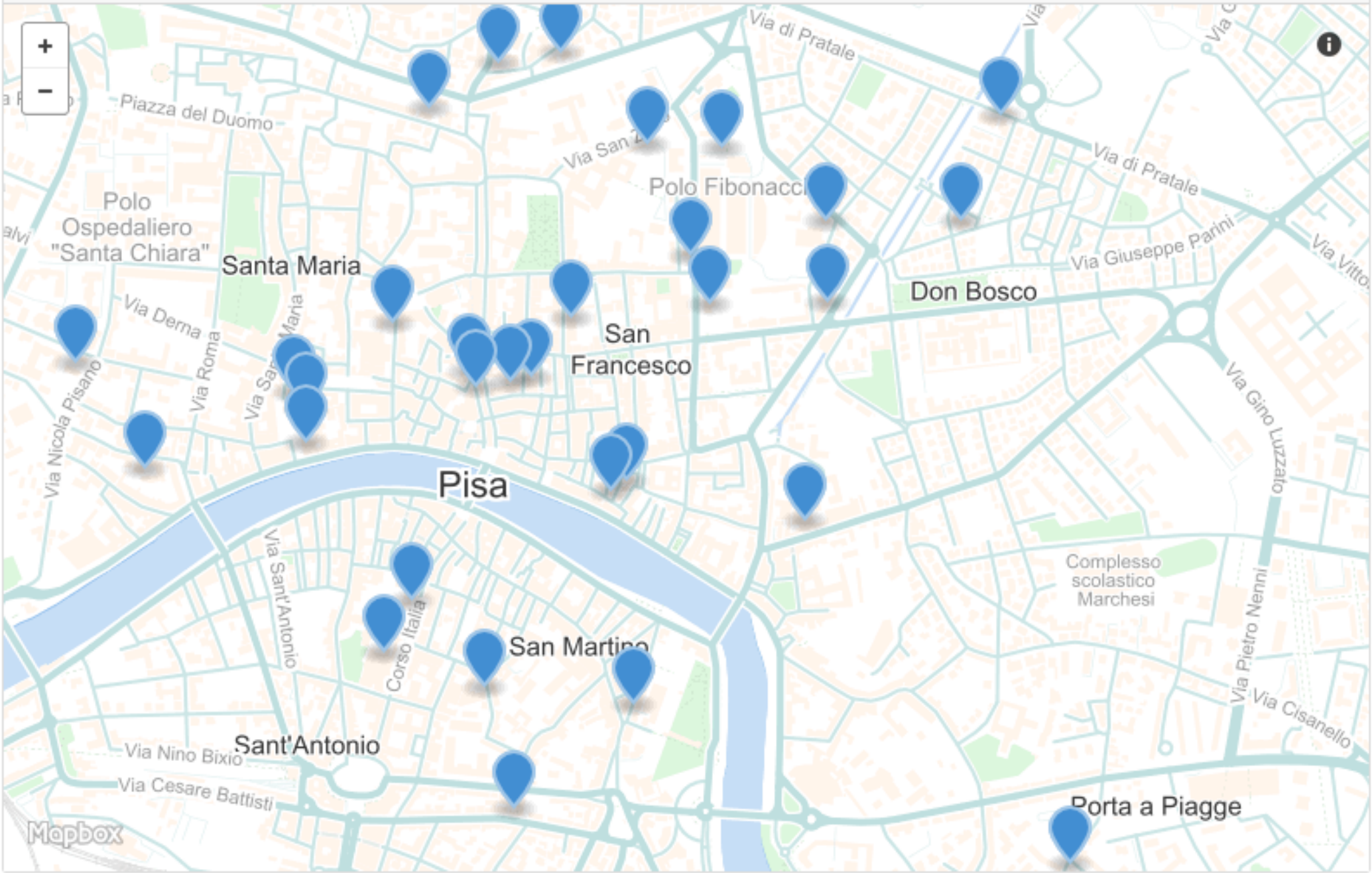
```
>>> from django.contrib.gis.geos import GEOSGeometry
>>> geojson = '{"type":"Point","coordinates":
[125.6,10.1]}'
>>> GEOSGeometry(geojson)
<Point object at 0x3b66c70>
```

GeoJSON facilitates creating

Maps for webpages

nodi-pisa-test

gistfile1.json



Leaflet.js

```
geojsonFeature = {
    "type": "Feature",
    "geometry": {type":"Point",coordinates": [125.6,
10.1]},
    "properties": { name": "Dinagat Islands" }
}

// create a web map with one line of code!

L.geoJson(geojsonFeature).addTo(map);
```

A bit of history: 2014

GsoC 2014: netengine

http://github.com/ninuxorg/netengine

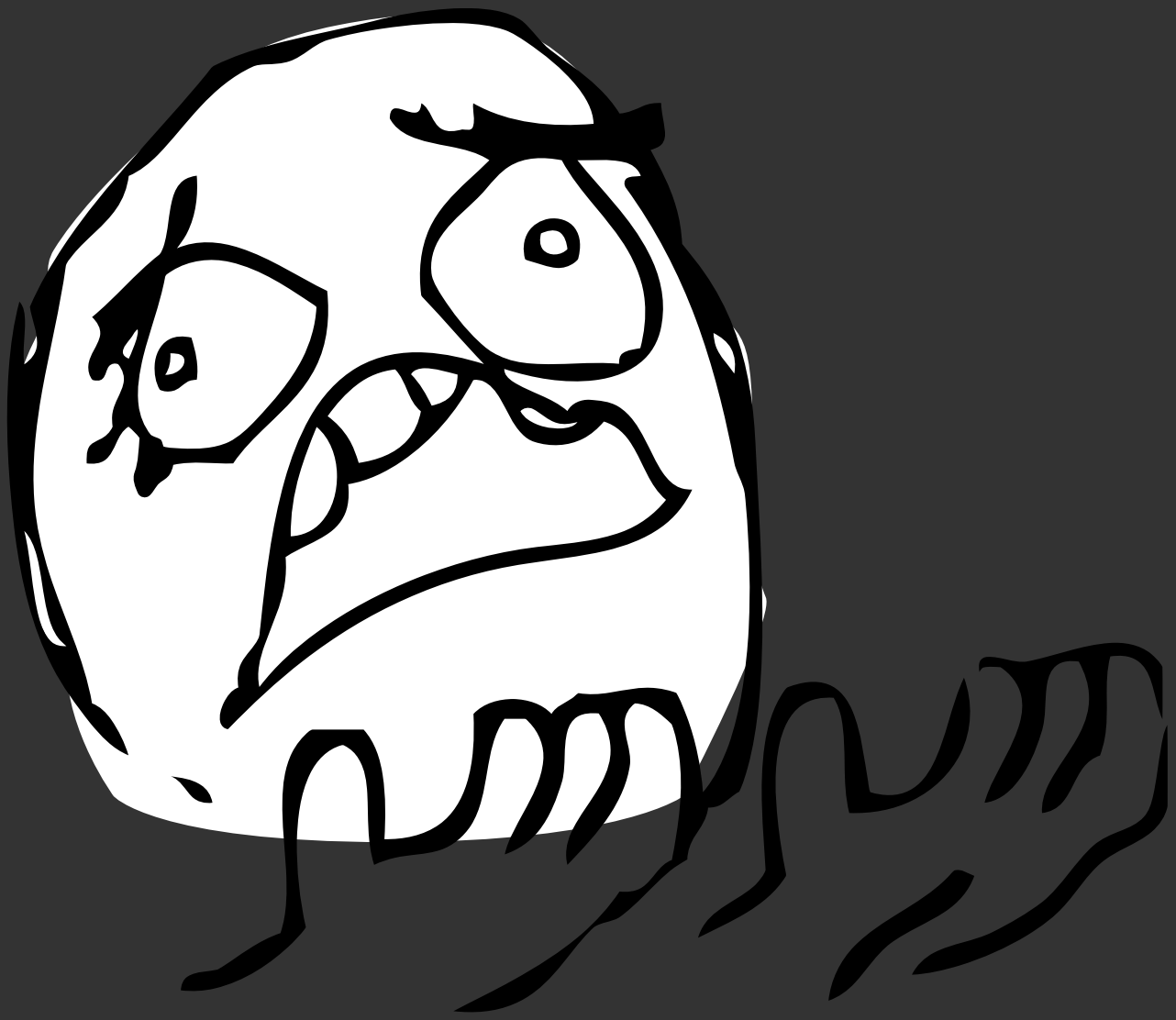
We need some standard JSON!

October 2014

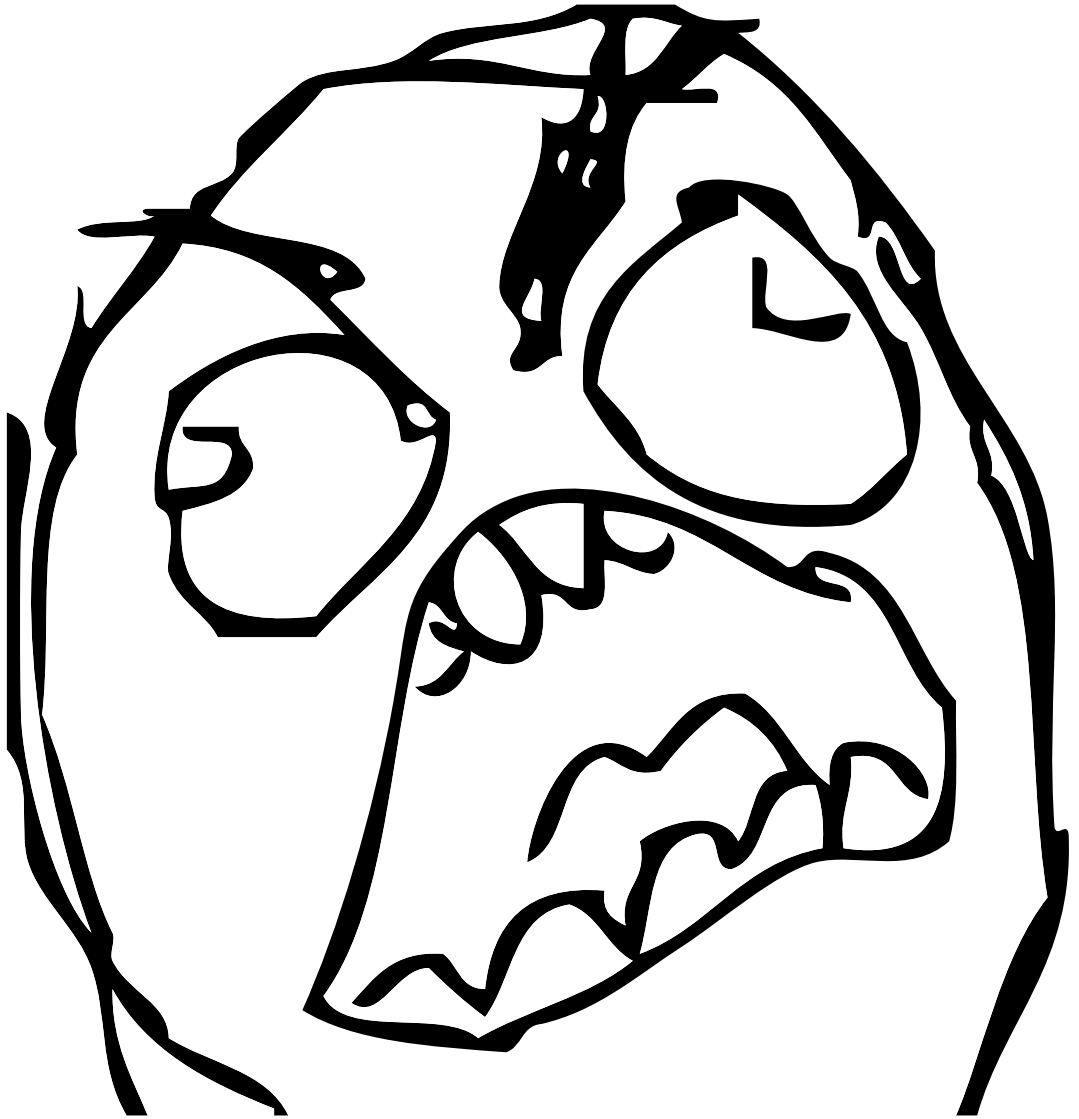
started working on the

first drafts of NetJSON

But... why?

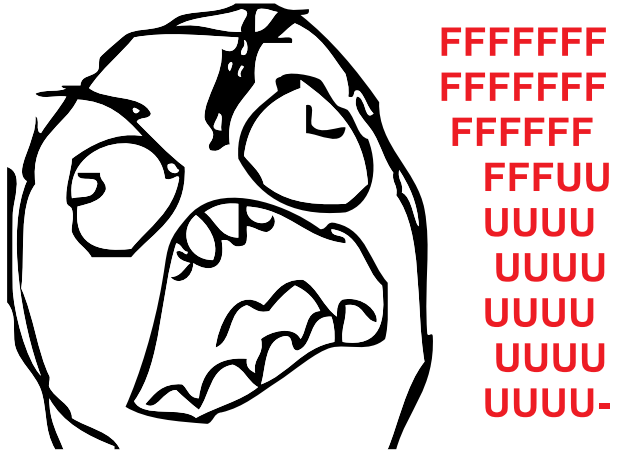


ever tried to develop software for **heterogeneous** networks?

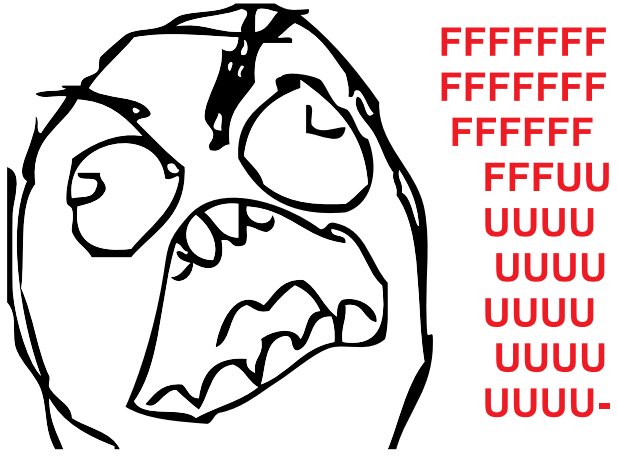


FFFFFFF
FFFFFFF
FFFFFF
FFFUU
UUUU
UUUU
UUUU
UUUU
UUUU-

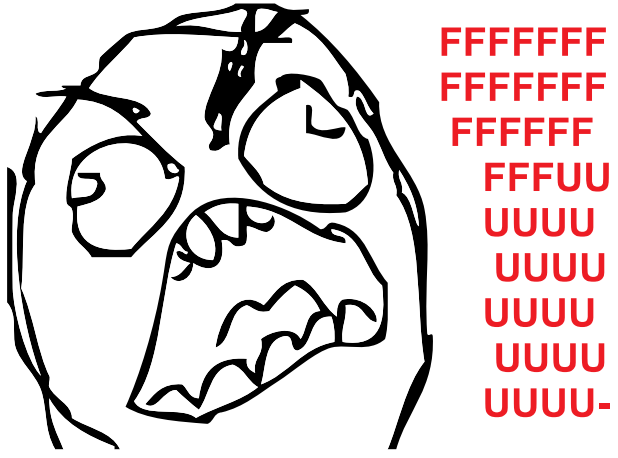
Vendors don't care about interoperability



FOSS projects
seem too busy to care either



No standard way to extract and parse data



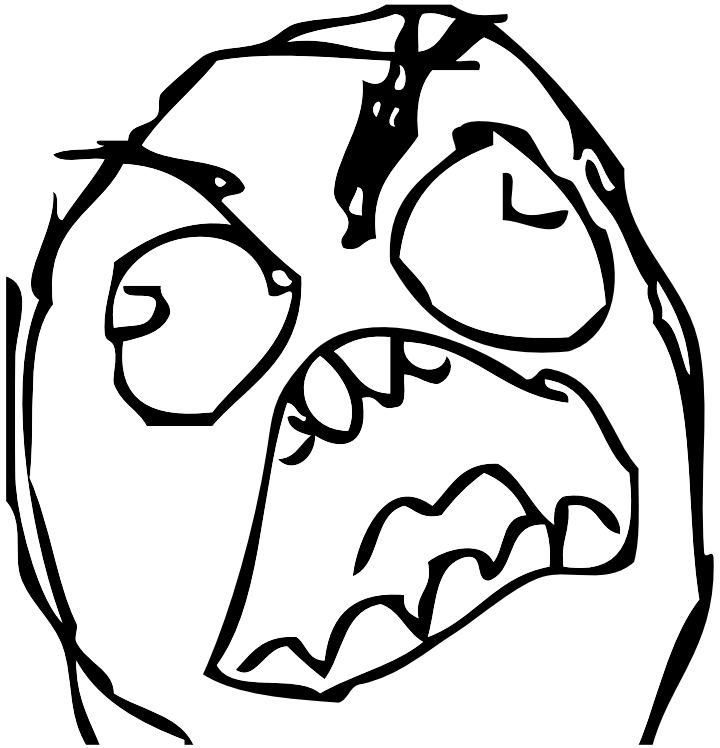
SILOS

+

VENDOR LOCK-IN

=

VERY SLOW INNOVATION



FFFFFFF
FFFFFFF
FFFFFF
FFFUU
UUUU
UUUU
UUUU
UUUU
UUUU-

We can do better than this.

We can achieve

interoperability

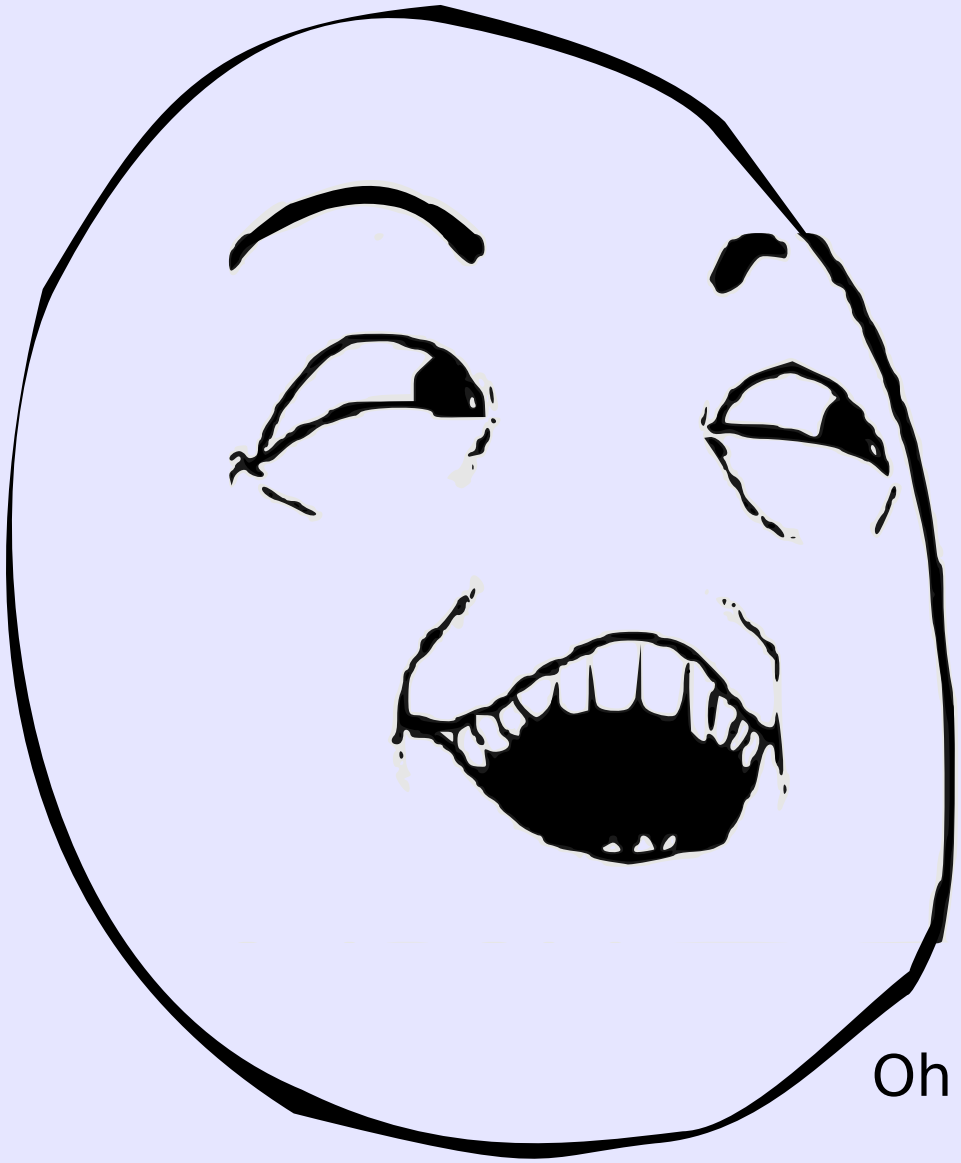
We can create an

ecosystem

We can foster

growth

AHAH, growth … but how?

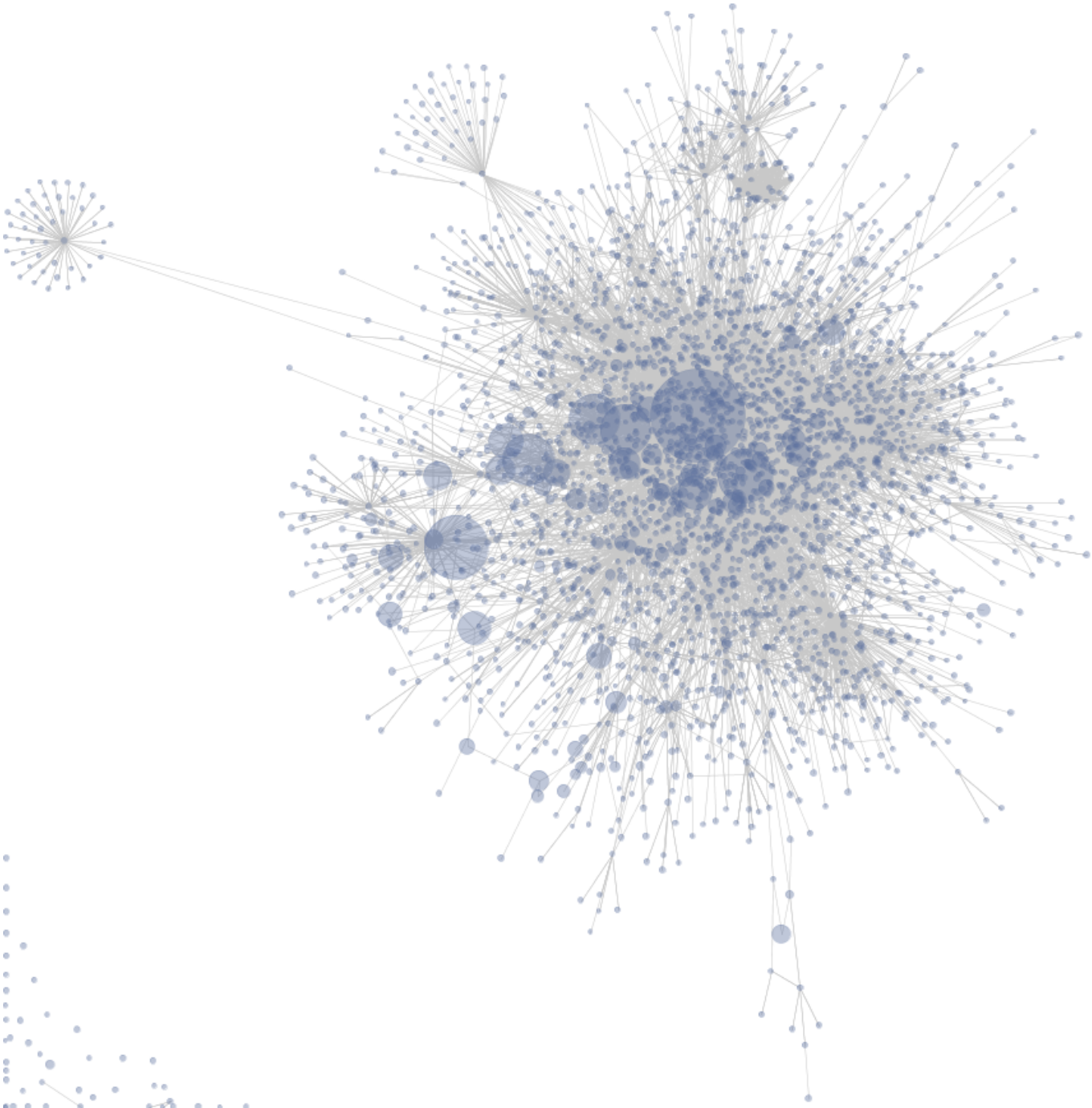


Oh yeah...

Easily import/export/deploy **device configurations**

Understand & visualize
network topology

ANY ROUTING PROTOCOL



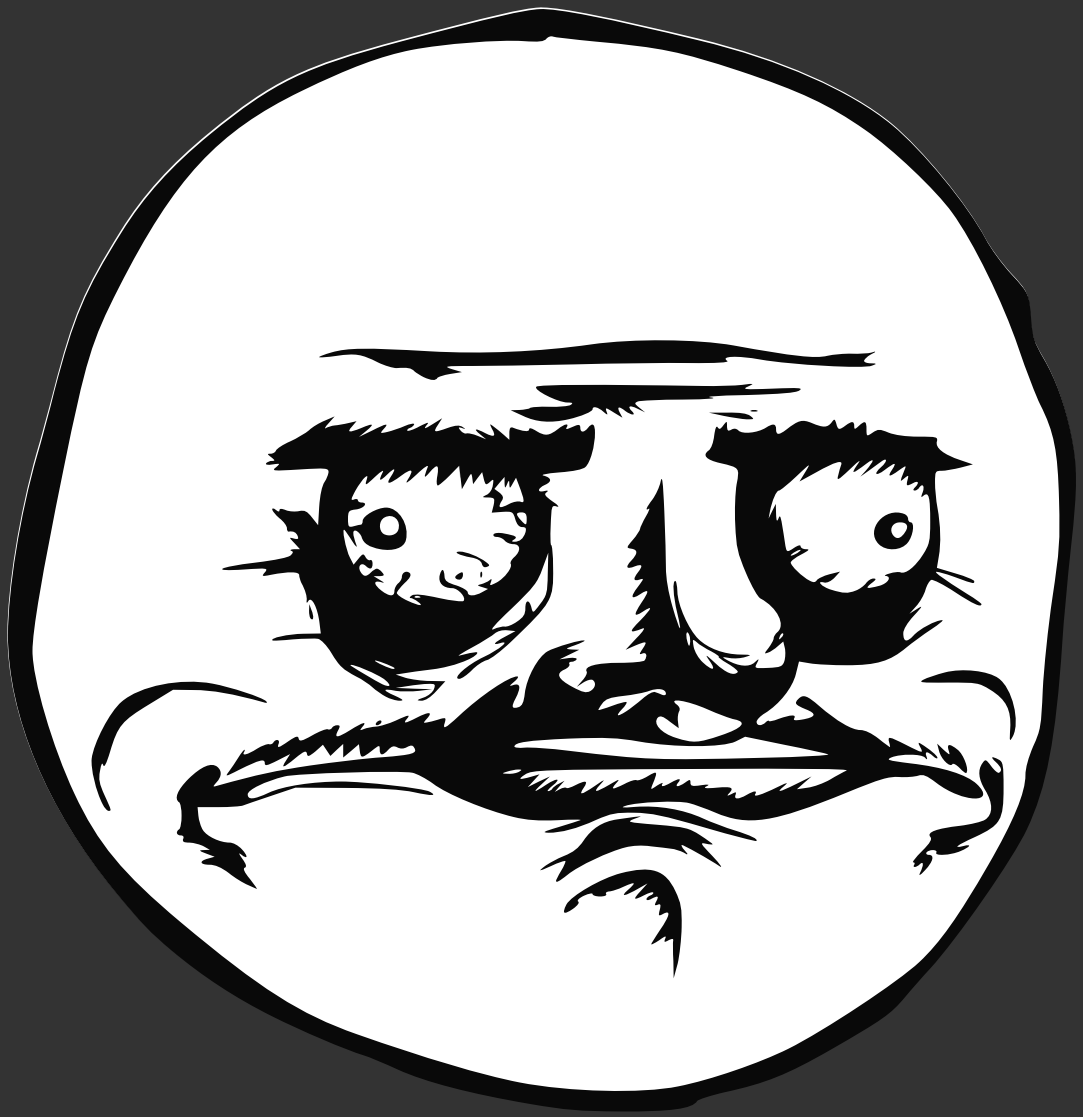
Small libraries developed by
different communities

written in
different languages

can **interoperate**

Develop the new cool thing...

And anybody can start using it straightaway!



ME GUSTA

Current implementations?

OLSR Network Framework

NetJSON info plugin

netdiff

“calculate difference of network topologies”

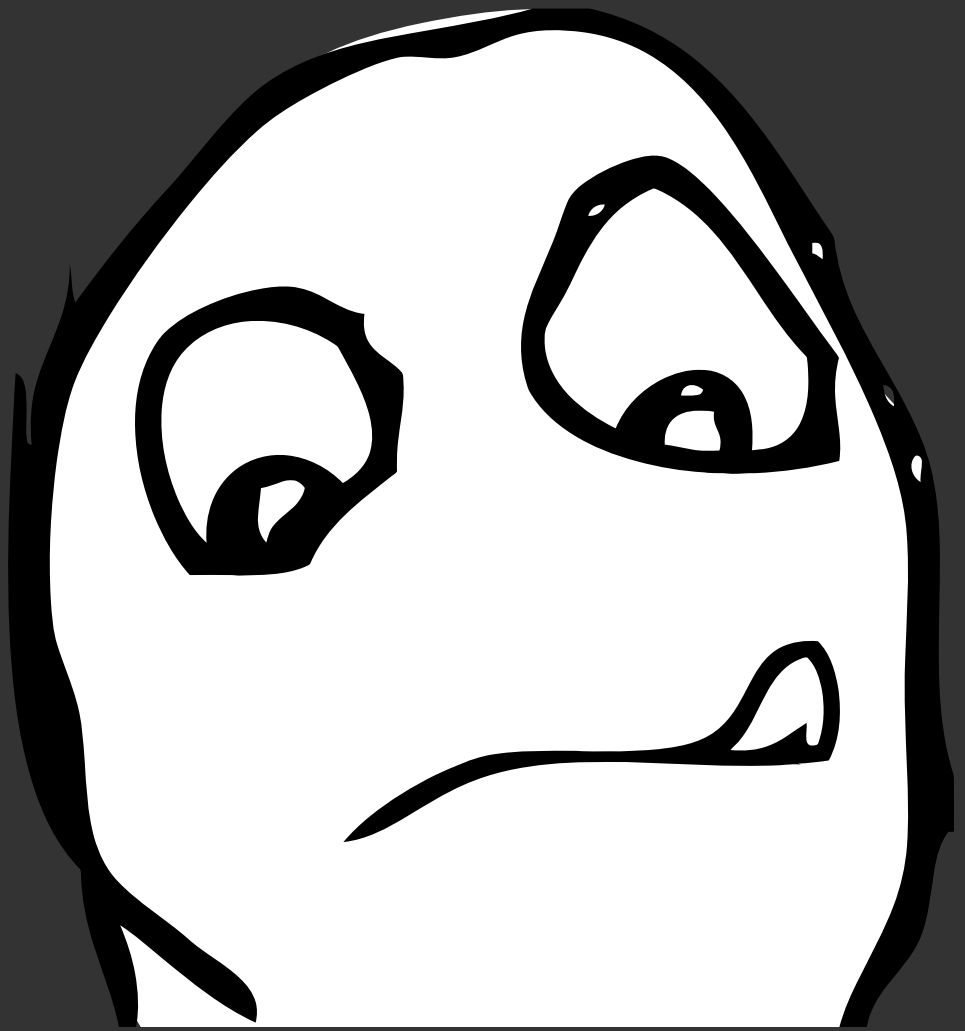
github.com/ninuxorg/netdiff

netengine-utils

“utilities for parsing output of common shell utilities like ifconfig and iwconfig”

github.com/ninuxorg/netengine

We are working on it...



Roadmap

1. Moaar implementations

nodeshot

“crowdmapping for community networks”

github.com/ninuxorg/nodeshot

Javascript d3 library

visualize “NetworkGraph” objects
(topology of any routing protocol)

2. Moaar feedback

3. Integrate feedback

4. Freeze specification

5. JSON Schema

6. Validator

7. RFC

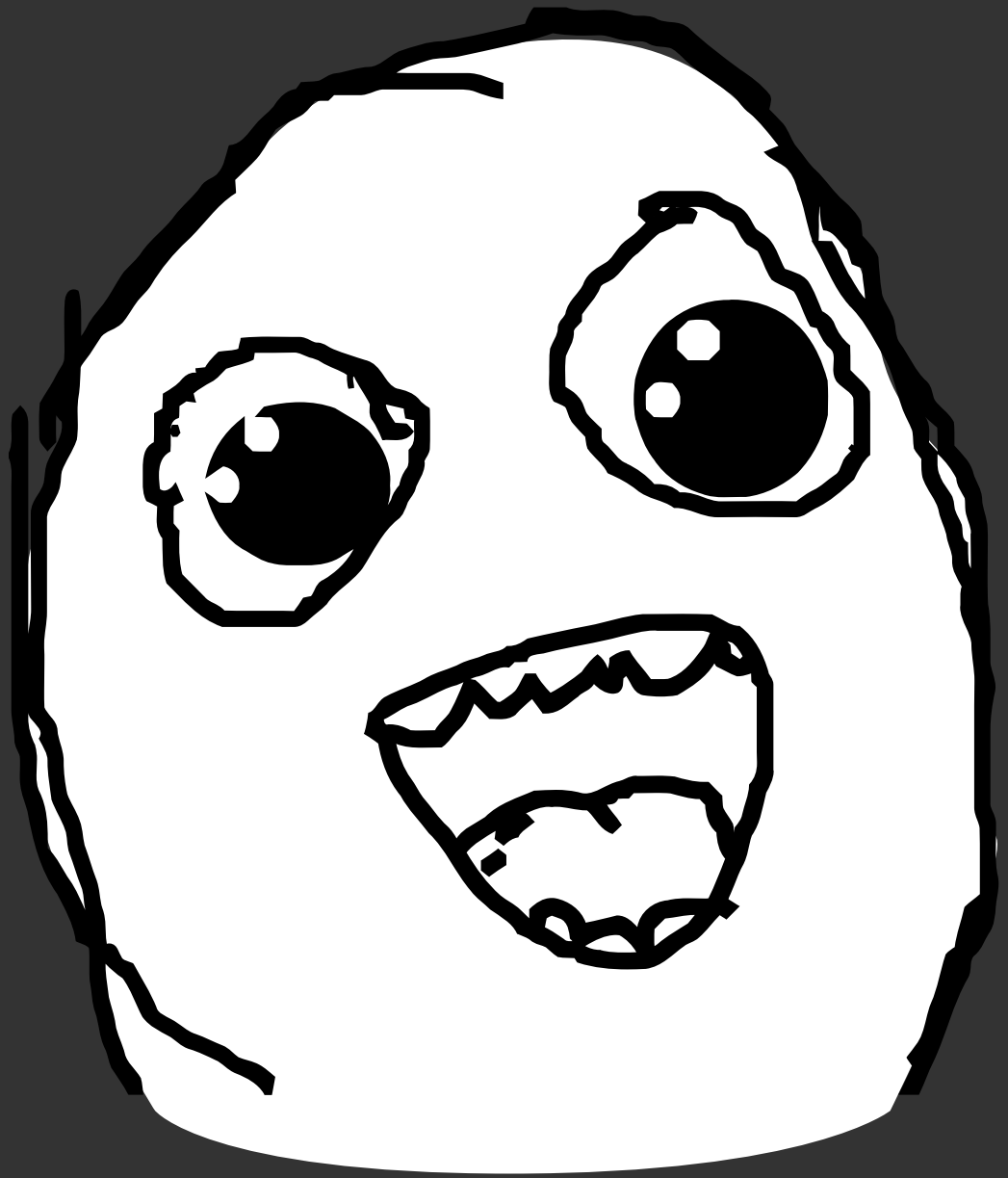
Want to help out?

1. read the spec

(10-15 mins max)

2. implement netjson

3. send feedback



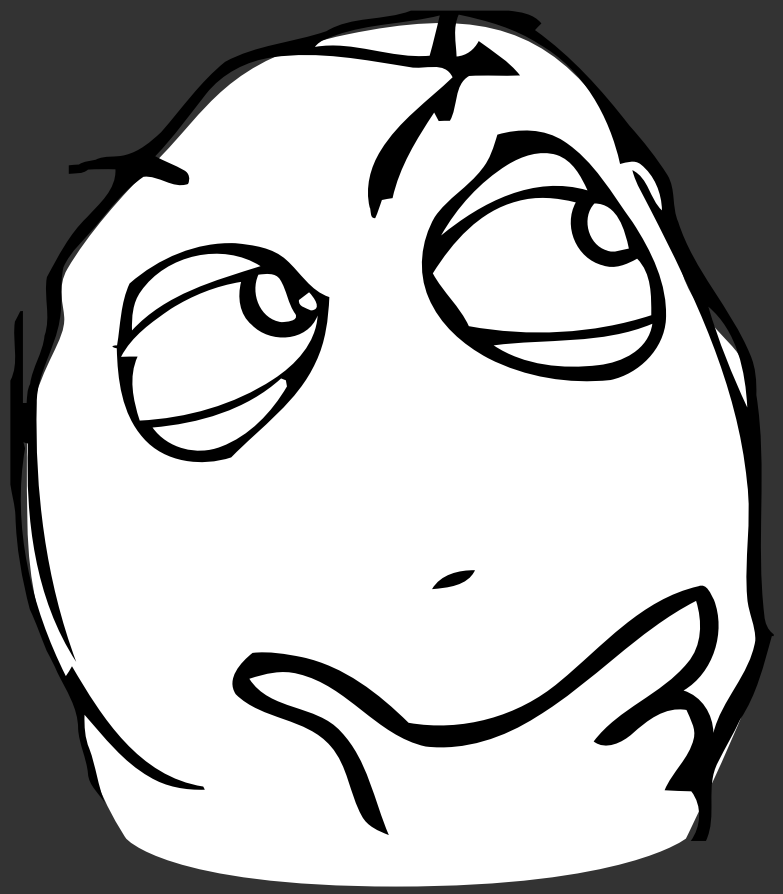
Find more about NetJSON

netjson.org

github.com/interop-dev/json-for-networks

mailing list **interop-dev**

Why so much interest in interoperability?



A bit of history: 2011

Common **NodeDB** effort

Didn't go well

Different communities have different needs.

Fast forward

Do we have what we hoped for?

imho: NO

Nikolas is not proud of us

Reasons?

Slow progress

Reasons?

Fragmentation and duplication of efforts

Reasons?

Very different software

Not interoperable


WHAT IF PEOPLE ARE NOT CONTRIBUTING TO MY PROJECT
BECAUSE IS JUST TOO FUCKING COMPLEX?
imgflip.com

experience has taught me

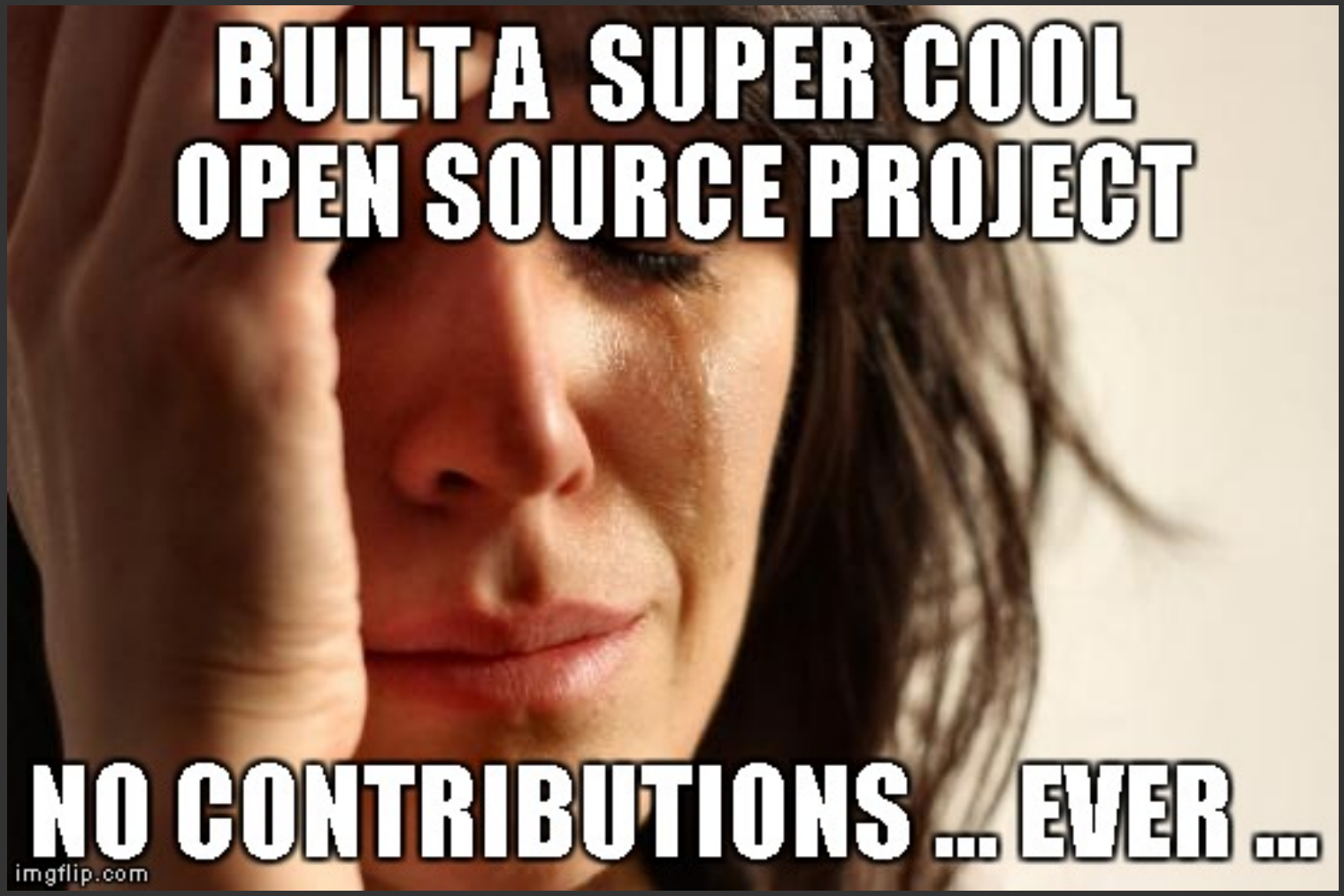
Developers interested in a **specific** feature are **discouraged** by **complexity**

Nor they will be able

to contribute without putting up

a huge (**and unlikely**) amount of effort

How to remediate?



netjson

Could be a step in the **right direction**

But **it will take time**

“Now is better than never”

What can we do?

Extract key features in libraries

Small

Reusable

Standalone

Well documented

Should focus on **one** problem

(or very few related problems)

And solve it well

Aka: “the Unix philosophy”

Should **encourage contributions**

by clearly explaining

“how to contribute”


WHAT ARE THE
FRIGGING BENEFITS?

“Simplicity is beautiful”

One problem is...

easier to get right

A small library is...

easier to maintain

A small library is...

easier to document

A simple library is...

easier to use and integrate

A simple library is...

more likely to receive contributions

A standalone library can...

Be used by a wider range of ppl

Such a library will result in...

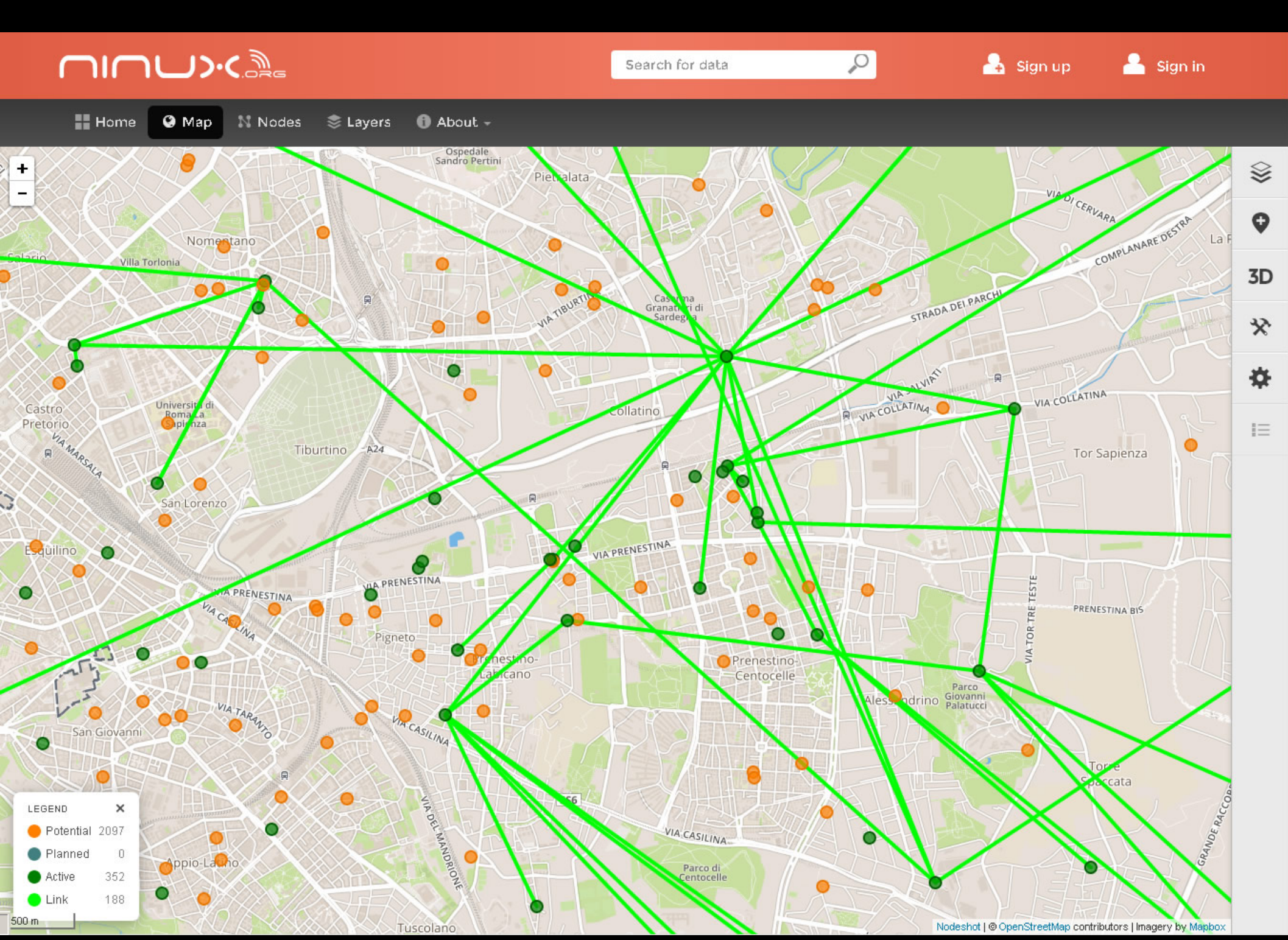
Better and longlived software

Such a library will also...

Attract interest to your main project

Real world examples

nodeshot.org


Search for data
Sign up
Sign in
Home
Map
Nodes
Layers
About
Ospedale Sandro Pertini
Pietralata
Nomentano
Villa Torlonia
Castro Pretorio
Università di Roma La Sapienza
Tiburtino
A24
San Lorenzo
Esquilino
Collatino
Tor Sapienza
Pigneto
Prenestino-Labicano
Prenestino-Centocelle
Parco Giovanni Palatucci
Parco di Centocelle
San Giovanni
Appio-Latino
Tuscolano
VIA PRENESTINA
VIA CASILINA
VIA TARANTO
VIA DEL MANDRIONE
VIA TIBURTINA
VIA COLLATINA
VIA DI CERVARA
COMPLANARE DESTRA
STRADA DEI PARCHI
VIA ALVIATI
VIA TOR TRE TESTE
PRENESTINA BIS
VIA MARSALA
LEGEND
Potential 2097
Planned 0
Active 352
Link 188
500 m
Nodeshot | © OpenStreetMap contributors | Imagery by Mapbox

What was wrong with it?

- Too many features
- Hard to contribute
- Modules not standalone

Then we started extracting and simplifying

netdiff

“calculate difference of network topologies”

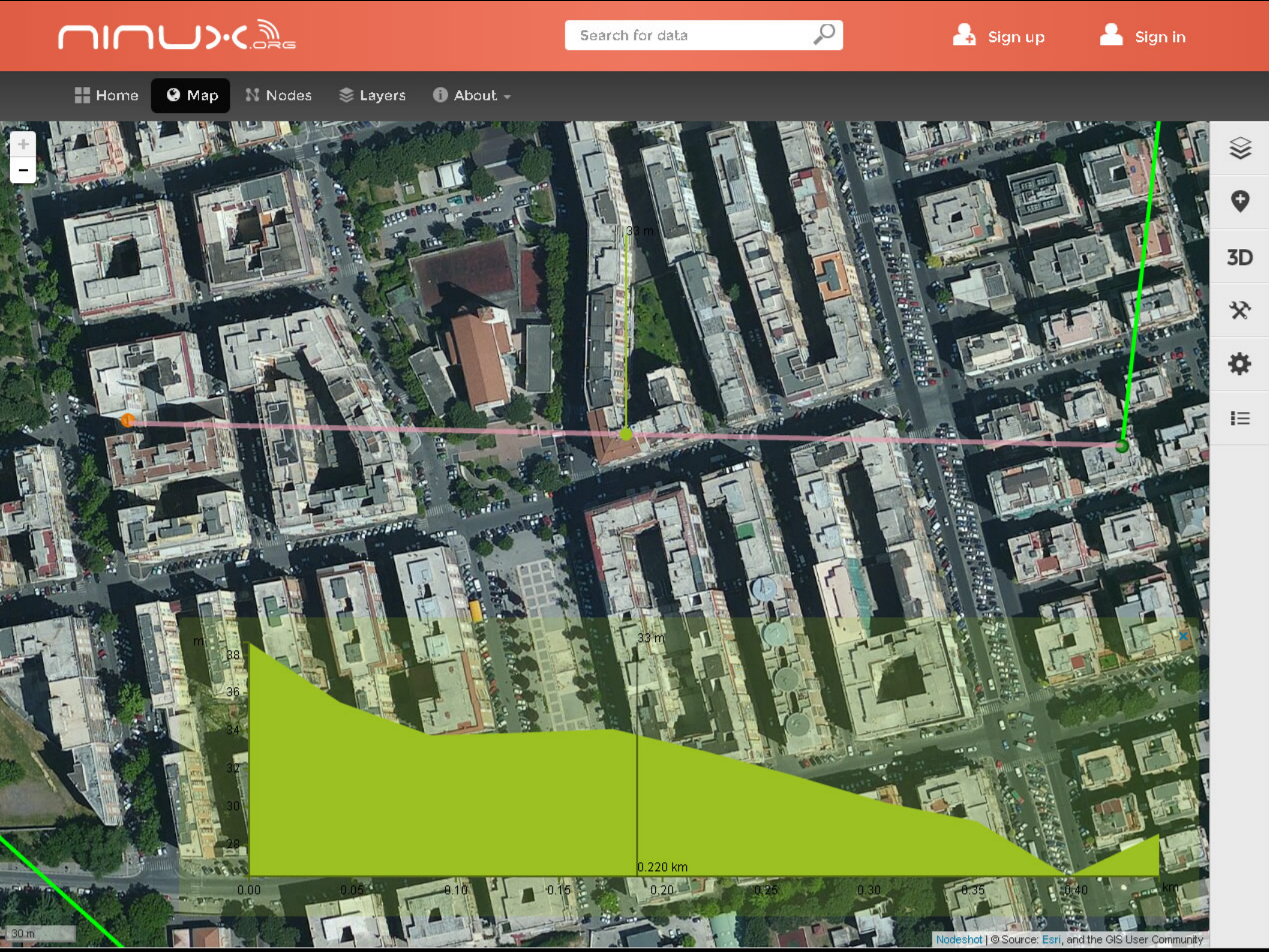
github.com/ninuxorg/netdiff

quick netdiff demo

python-geojson-elevation

proxy to Google Elevation API
which returns GeoJSON

github.com/ninuxorg/python-geojson-elevation


ninux.org
Search for data
Sign up
Sign in
Home
Map
Nodes
Layers
About
3D
33 m
m
38
36
34
32
30
28
33 m
0.220 km
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
km
30 m
Nodeshot | © Source: Esri, and the GIS User Community

django-rest-framework-gis

Geographic add-ons for Django Rest Framework

In short: GeoJSON restful API (read/write)

github.com/djangonauts/django-rest-framework-gis

django-hstore

PostgreSQL HStore support for Django

github.com/djangonauts/django-hstore

Django 1.8 ships HStoreField

which is a simplified field based on django-hstore

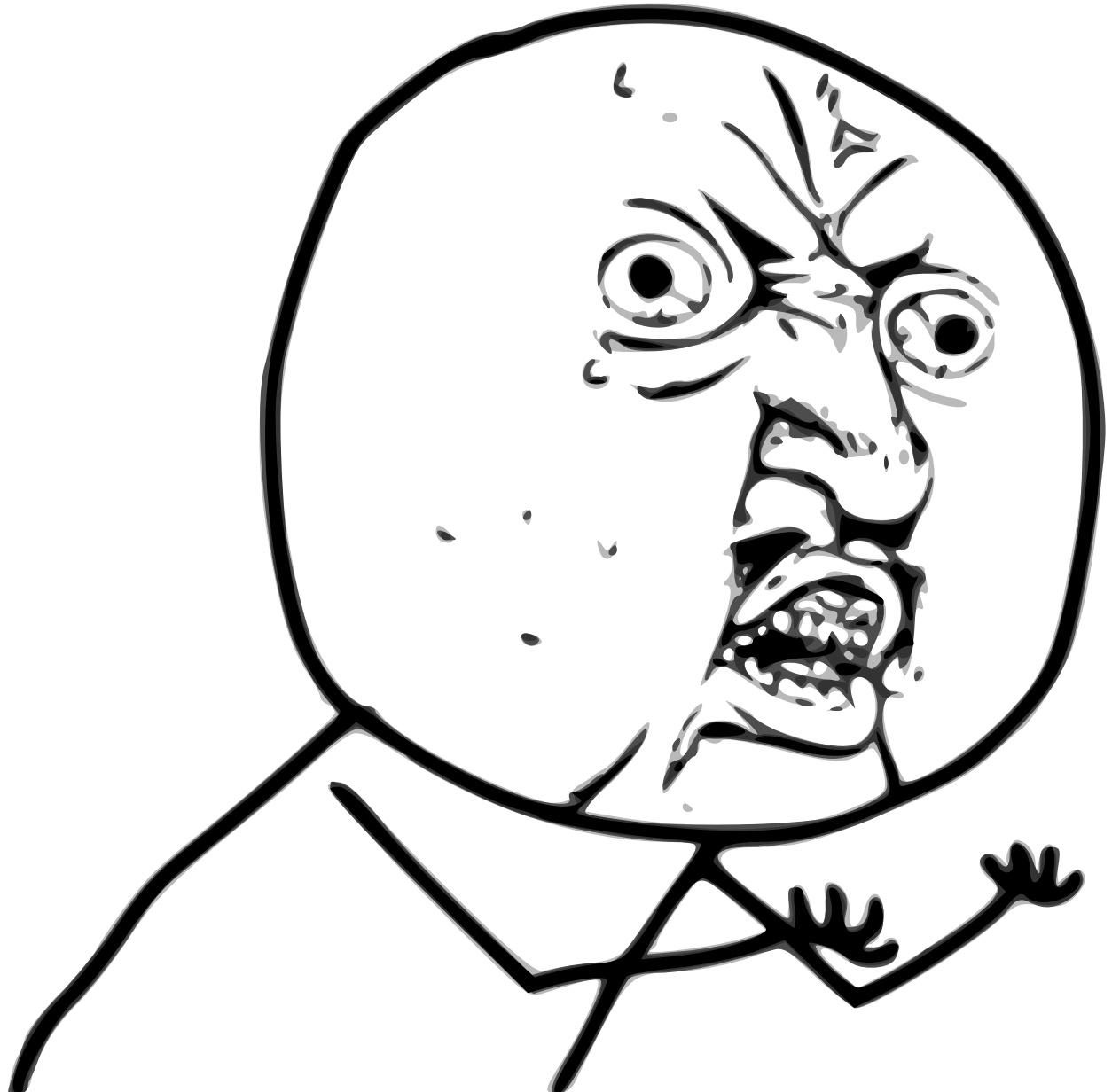
ref: postgres.mjtamlyn.co.uk/launch.html

A few more libraries:

django-rest-framework-hstore

netengine

Why U no provide some data?



Ok, ok, let's talk about data

nodeshot

vs

extracted features (summed)

contributions & downloads

Pull request (last 6 months)

Nodeshot: **13**

Libraries: **44**

3x

PyPi downloads (last month)

Nodeshot: **324**

Libraries: **25199**

78x

Github clones (last month)

Nodeshot: **95**

Libraries: **1392**

14x

Github unique clones (last mo)

Nodeshot: **14**

Libraries: **471**

33x

Github page views (last month)

Nodeshot: 1100

Libraries: 2385

2x

Github unique visitors (last mo)

Nodeshot: **128**

Libraries: **635**

5x

usage metrics of **libraries**
(summed up together)
have an average

22x increase

compared to **nodeshot**

Abraham approves

now **immagine...**

if all those features were stuffed into **nodeshot...**

would we still get the same amount of overall **usage metrics**?

I don't think so.

Why usage metrics matter?

usage metrics help you
to understand
If you are on the **right path**

Let's sum up

What you SHOULD NOT be doing

Do not stuff too many **features**

in a single **huge** project

Do not mix logic of

core features

with web framework code

(eg: **django**, **flask**, **web2py**)

Do not wait to achieve perfection before releasing

What you SHOULD be doing

Extract key features

In standalone, small, reusable libraries

Contribute to existing projects when possible

#writethedocs

In english!

provide a way

to get in touch

release early

release often

find ways for others

to know about these libraries

listen to feedback from occasional contributors

Diversity is HEALTHY

different communities

build

different solutions

according to their needs

but...

the low level implementations can be **shared** across projects

We are **already**
doing that with
Linux and **OpenWRT**

Let's do more of it

Interoperability

+ Collaboration

+ Diversity

= Growth

Let's thrive together

Thank you

@nemesisdesign

twitter & github