GIS.lab

news from development of technology for rapid deployment of complete geospatial infrastructure with supercow capabilities

Ivan Minčík (imincik)

FOSS4G Europe 2015, Como, Italy

The Cow

3 years old, **30 litres** of milk for **20 kg** of GRASS daily

Introduction

The Problem

We always need **more than one app** for our **GIS work flow**

The Problem

Deployment and maintenance of **complex system** is **time consuming** even when things are going flawlessly

The Problem

which usually they are **not** (:

The Idea

Is there any chance to have a system which is working
out-of-box ?

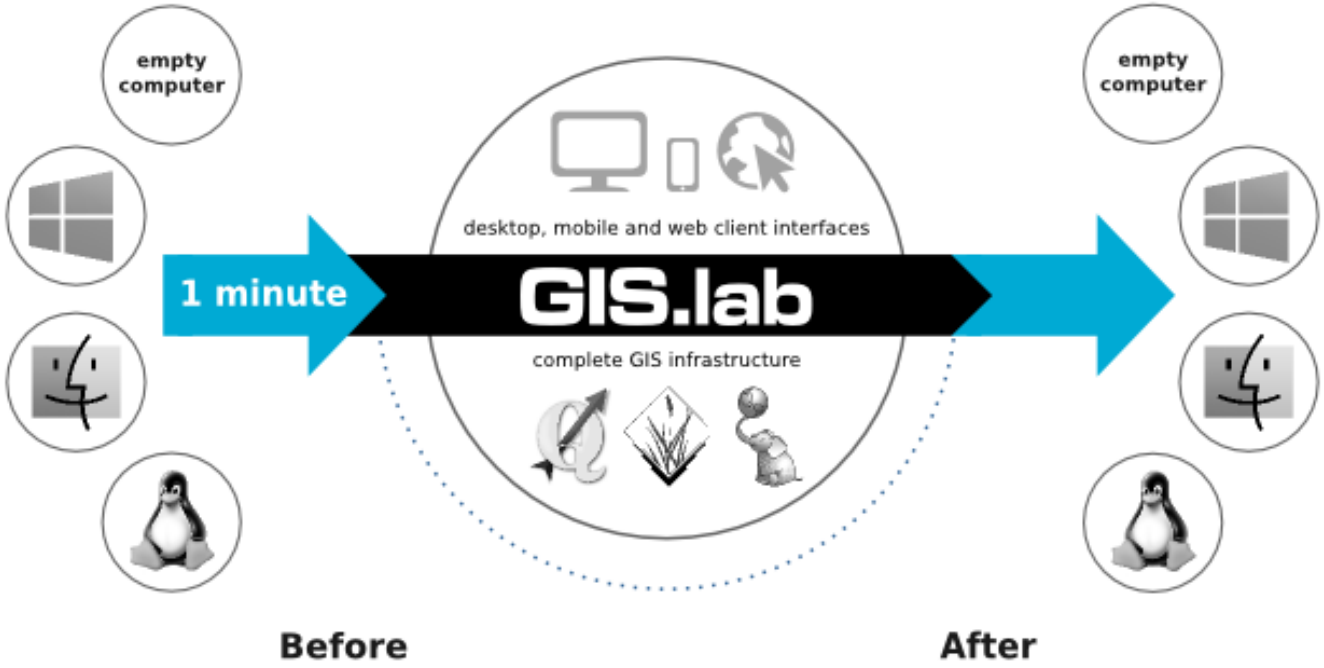
The Super Cow

Instantly in production, **no feeding**

What is GIS.lab ?

What is GIS.lab ?

Free technology which can **instantly** turn any computer network in to the **fully equipped geospatial cluster**



What is GIS.lab ?

and **back** again

Key Features

- only **single one machine** needs to be installed per network
- fully **automatic** installation
- client machines are working **out-of-box**
- contains everything **from data storage** to **mobile client interface**

Key Features

- **instant** deployment

Key Features

- **instant** deployment
- **central management**

Key Features

- **instant** deployment
- **central management**
- **desktop, web and mobile** client interfaces

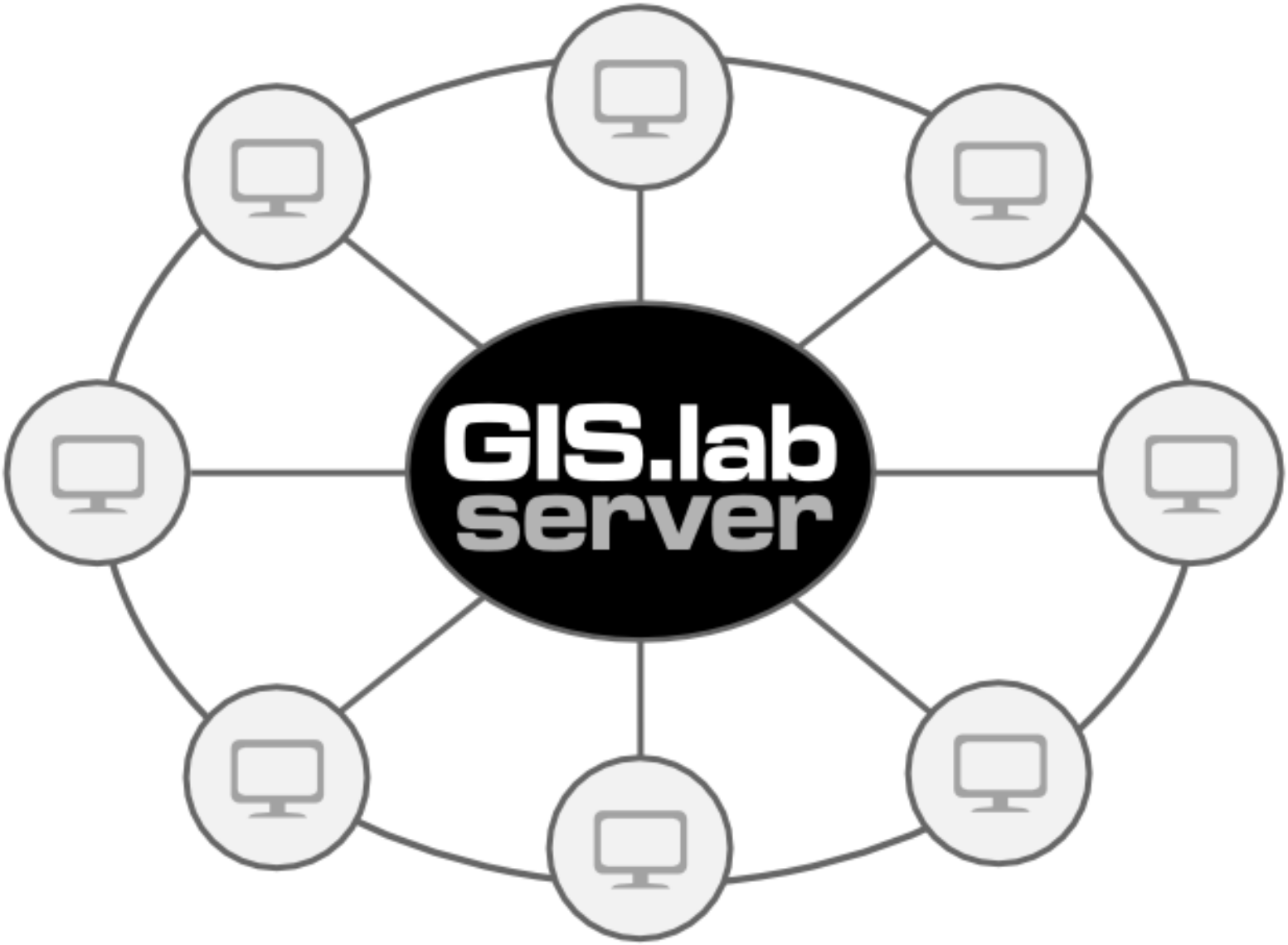
Key Features

- **instant** deployment
- **central management**
- **desktop, web and mobile** client interfaces
- automatic **clustering** and **computing power sharing**

Key Features

- **instant** deployment
- **central management**
- **desktop, web and mobile** client interfaces
- automatic **clustering** and **computing power sharing**
- **no dependencies**

GIS.lab Cluster Architecture



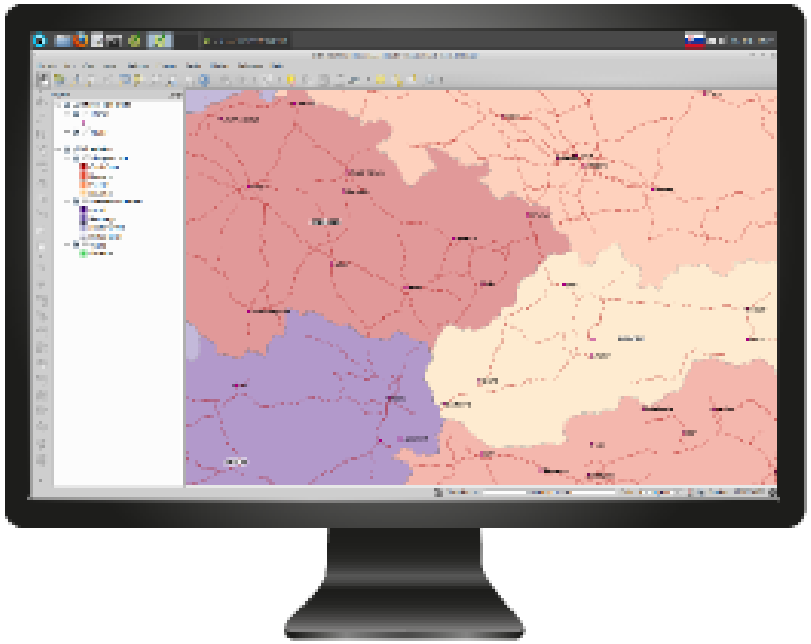
GIS.lab Server (Master)

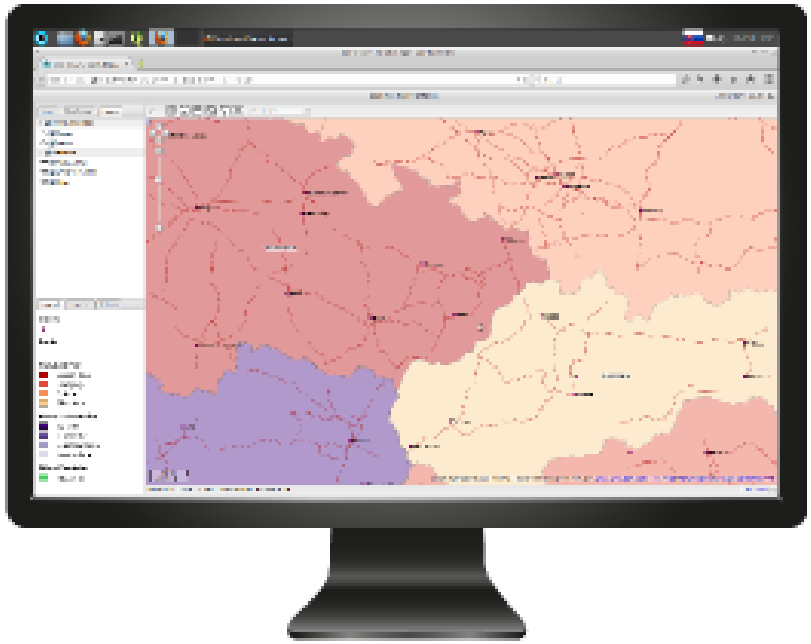
- **cluster** orchestration
- **data storage** and **sharing**
- **load balancing**

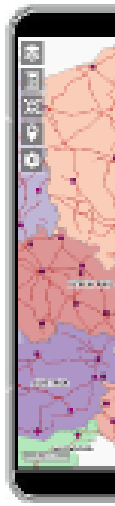
GIS.lab Clients

- initialized **from server**
- **user interfaces** for data processing, analysis and collaboration
- **computing power** for cluster

Desktop, Web and Mobile Client Interfaces







So, what's new ?

mostly **everything**

Deployment

Automatic Installation

- **no requirements on target machine** except SSH
- idempotent **modules, templates**
- **cloud providers** AWS, GCE, Digital Ocean, Azure ...

Automatic Installation


```
$ ansible-playbook
  --inventory=gislab.inventory
  --private-key=~/.ssh/id_rsa
  system/gislab.yml
```

Virtual Machine - Development and Testing


```
$ vagrant up

Bringing machine 'gislab_vagrant' up with 'virtualbox'
    provider...
==> gislab_vagrant: Importing base box
    'precise-canonical'...
==> gislab_vagrant: Running provisioner: install
    (ansible)...
```

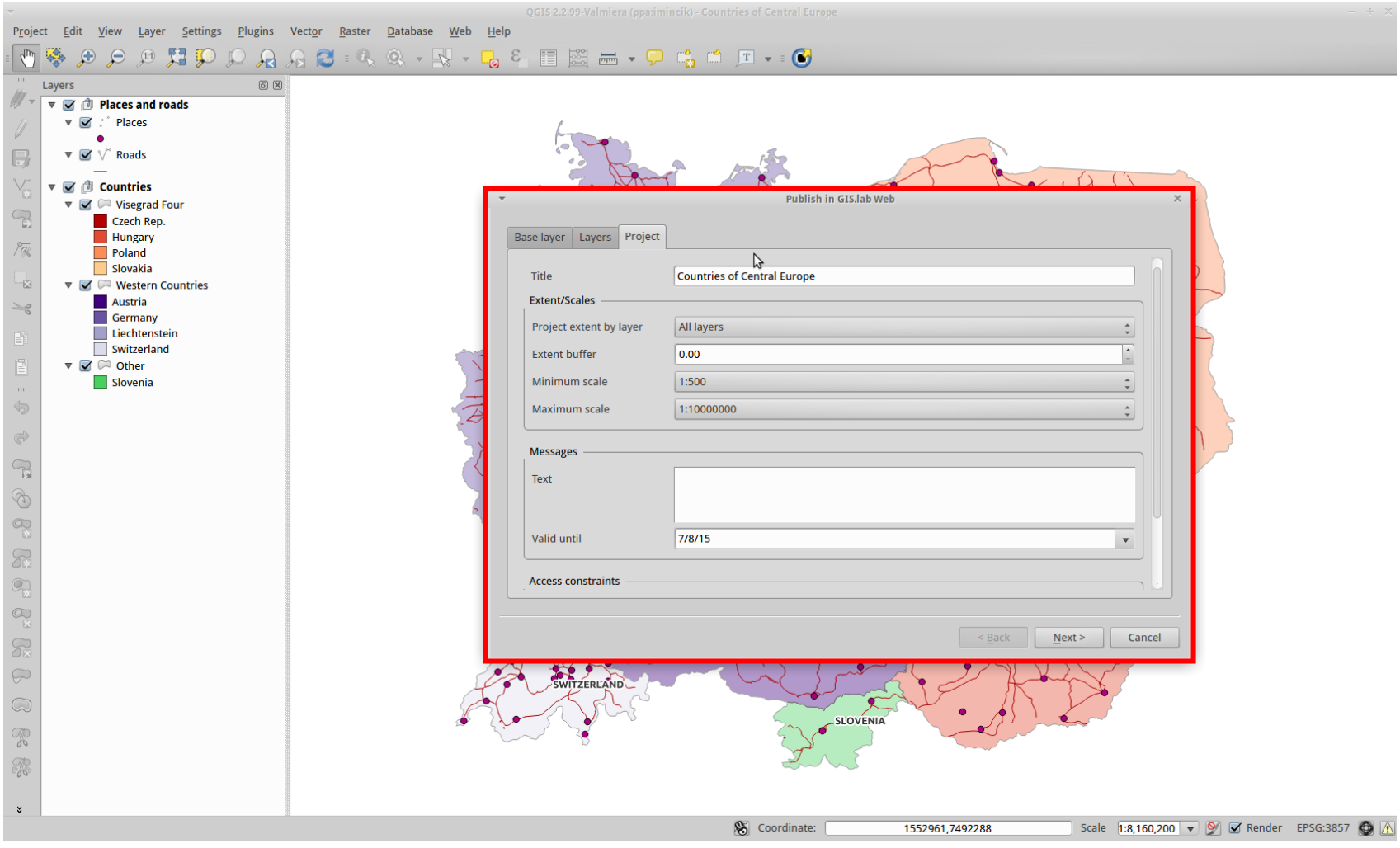
GIS.lab Unit - End User Deployment

- Intel Haswell, **16 GB RAM**, **SSD**, tested with **20 clients**
- **portable**, **pocket size** (11 x 11 x 4 cm)
- **plug-and-play**
- **automatic** host **network adaptation**

Client Interfaces

Client Interfaces Architecture

Publishing to Web and Mobile

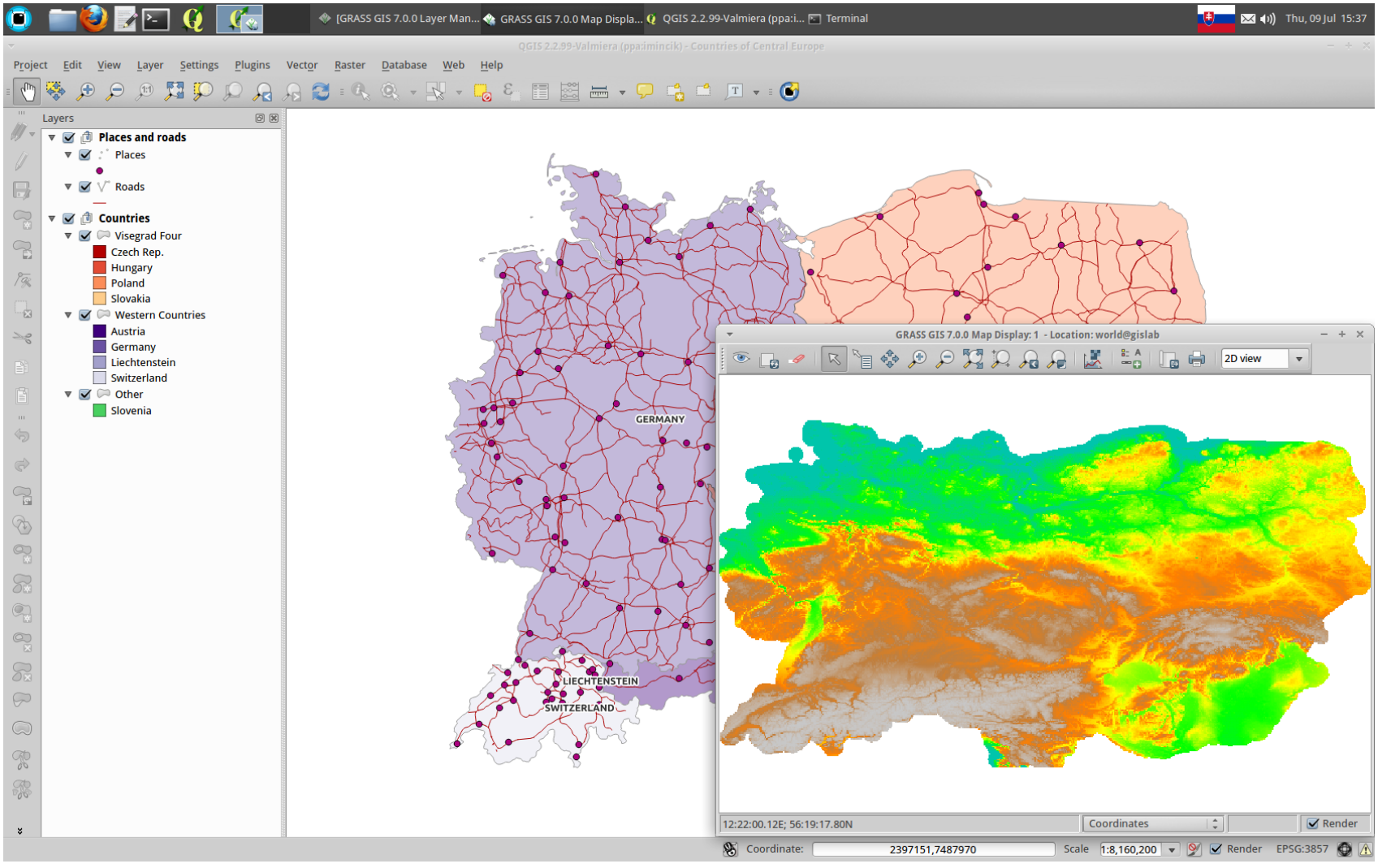


Desktop

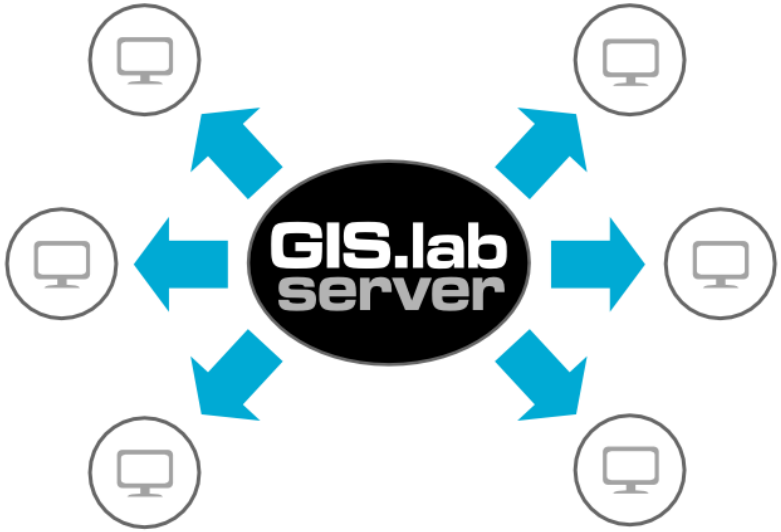
Desktop Interface

- **traditional**, **customized**, **low resources** environment
- **office** and **geospatial** software
- combination of **desktop performance** with **web accessibility**

Desktop Interface

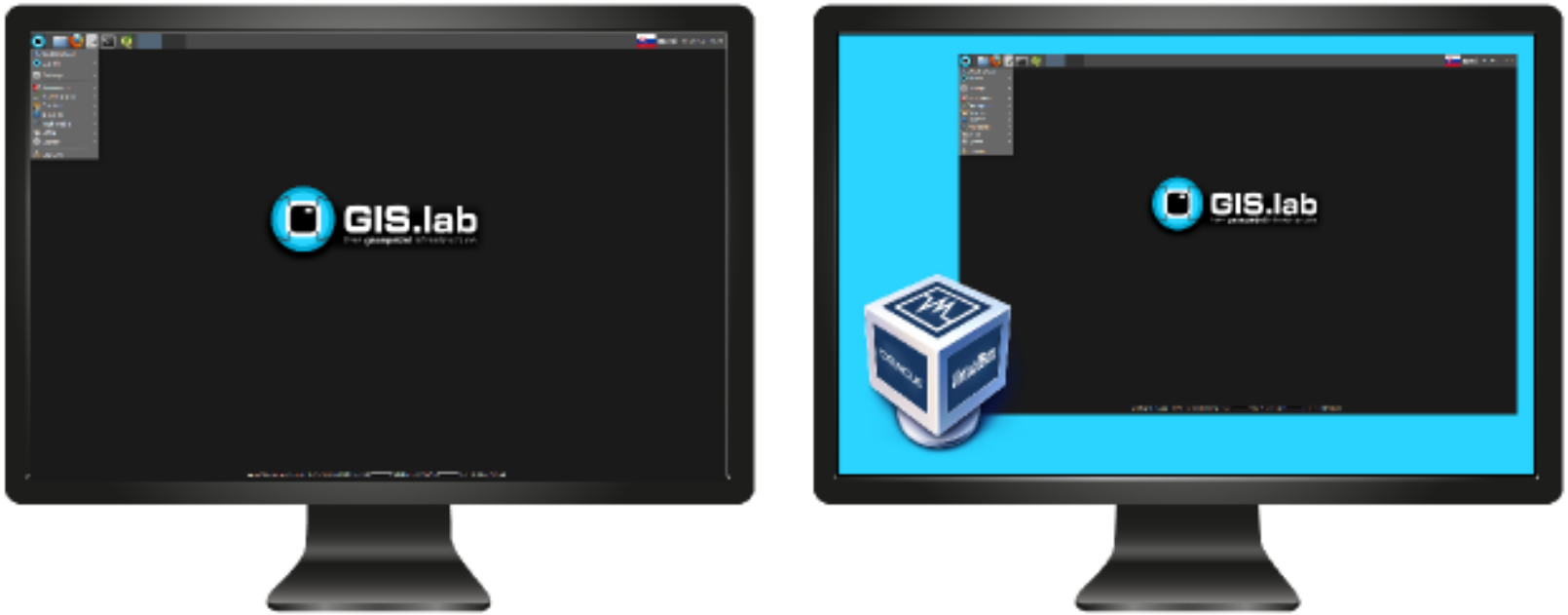


Machines Initialization



- **initialized** from **GIS.lab network** (PXE, HTTP)
- always **clean system**, **maintenance-free**
- **no HDD** required
- using **full hardware potential** - opposite to thin client

Physical or Virtual Mode



- **physical mode**: best performance, original OS is temporary lost
- **virtual mode**: any OS, original OS and GIS.lab are available

Customization

```
$ gislab-client-shell -i    # enter client env

$ apt-get install gedit     # install Gedit
$ exit                      # exit client env

$ gislab-client-image       # deploy updated client image
```

- **well known** tools
- **rollback**

Booster File System

Test writing of 2 GB file

HDD

```
$ dd if=/dev/zero of=/tmp/test.f bs=1M count=2048
  2147483648 bytes (2,1 GB) copied, 24,8055 s, 86,6 MB/s
```

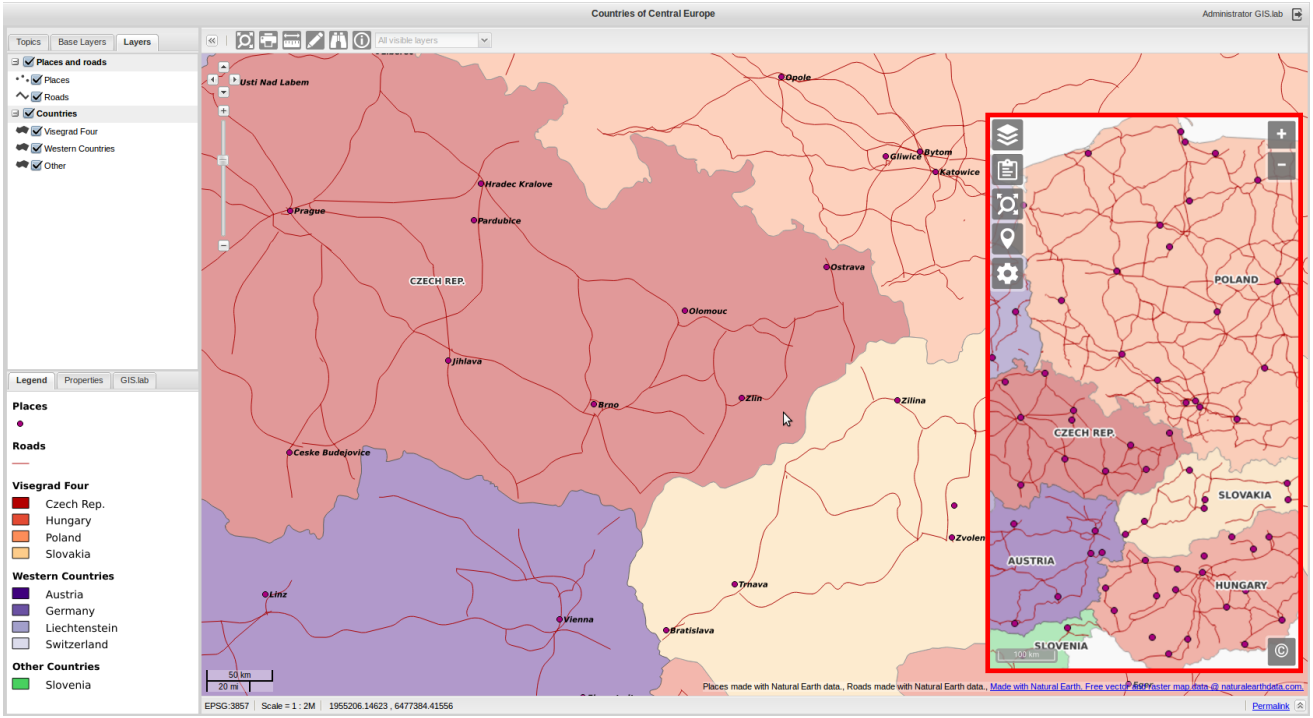
Booster

```
$ dd if=/dev/zero of=~/Booster/test.f bs=1M count=2048
  2147483648 bytes (2,1 GB) copied, 0,582147 s, 3,7 GB/s
```

- **super fast** file system in RAM
- ideal for **temporary files**

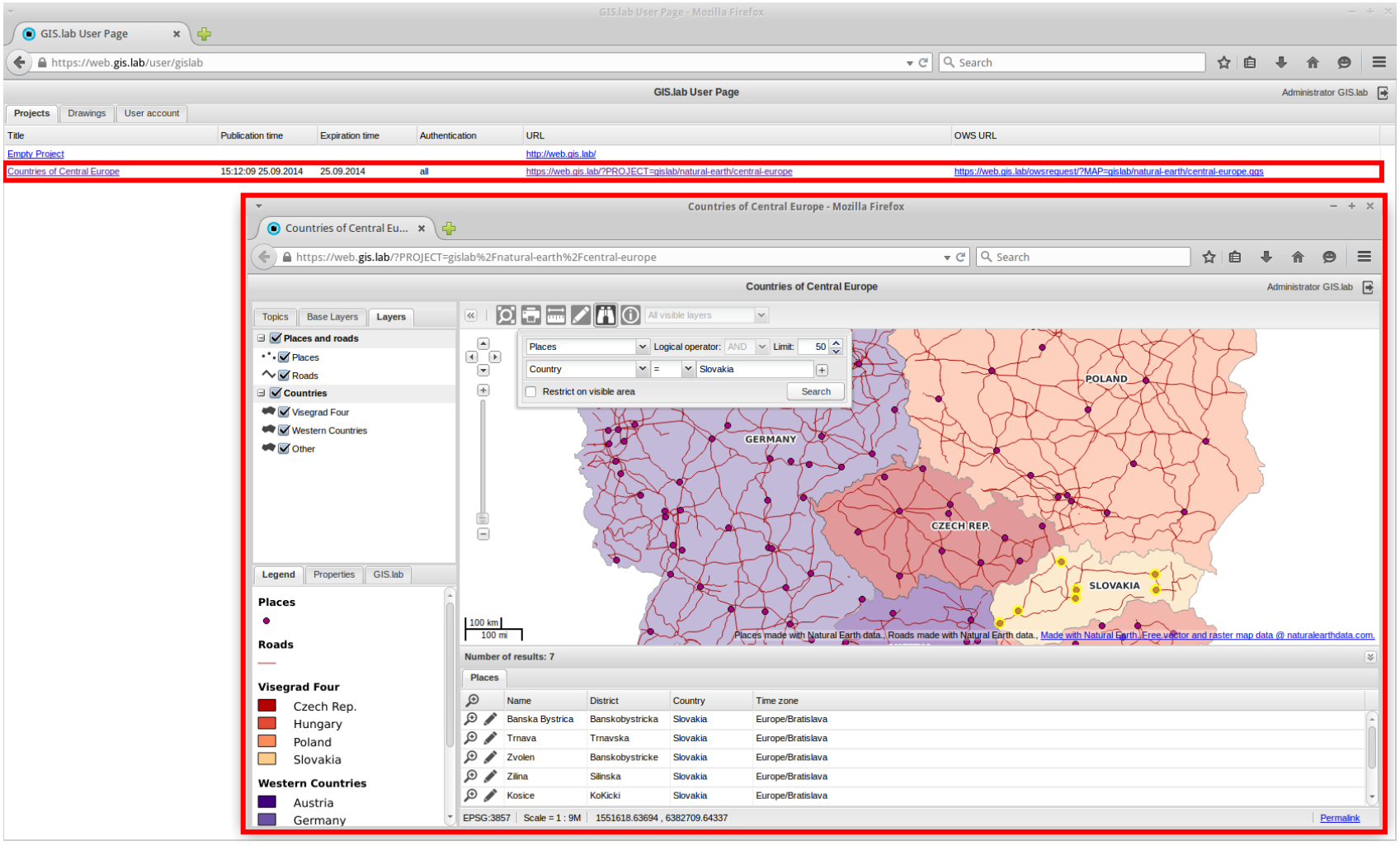
Web and Mobile

Web and Mobile Interface

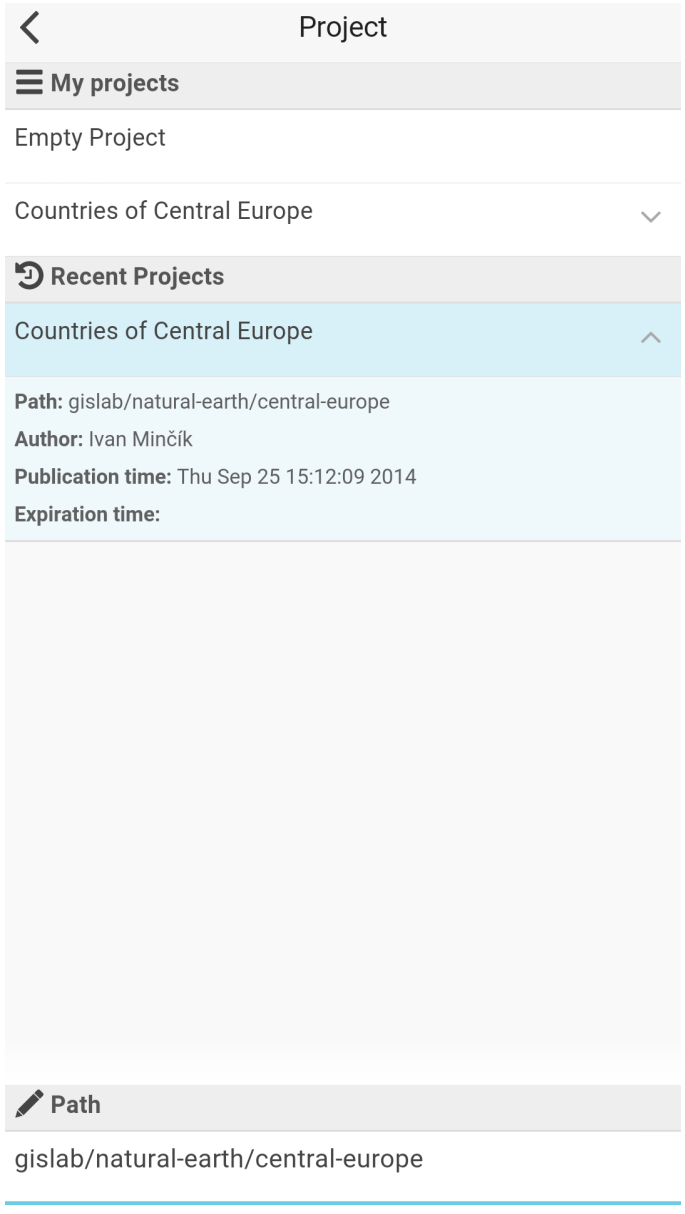


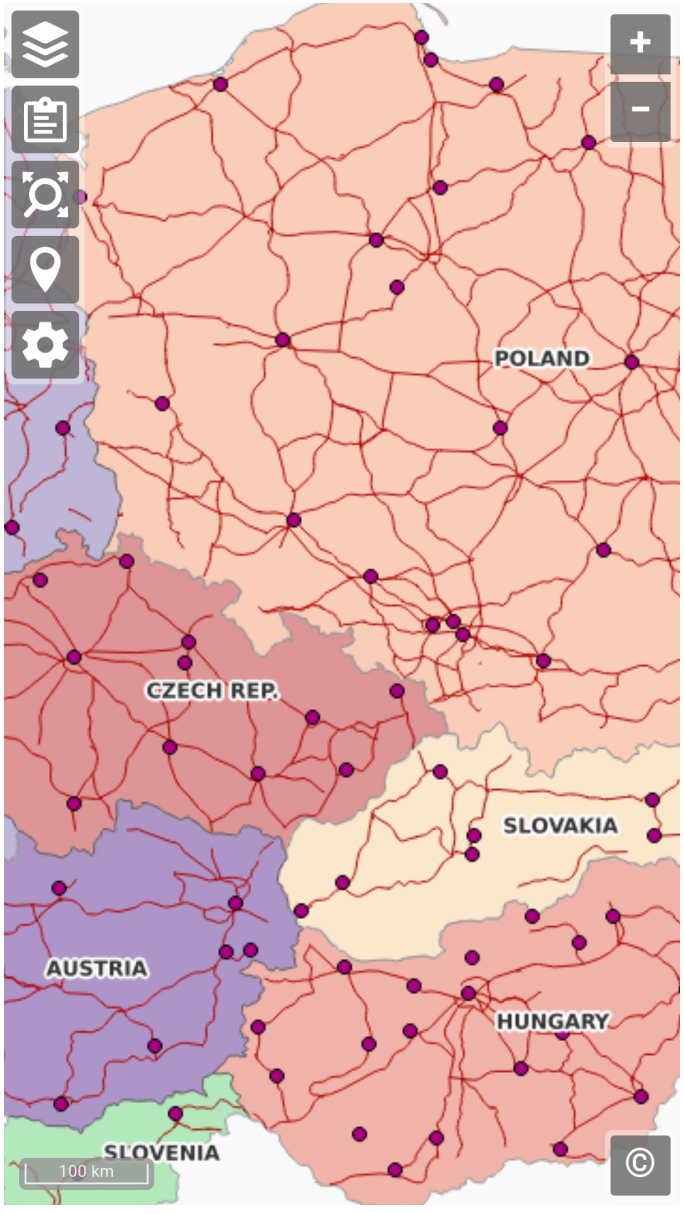
- **themes**, **base** and **overlay** layers
- advanced **search** forms
- **print** outputs
- vector features **drawing** and **sharing**

Web Interface



Mobile Interface





Cluster

Automatic Cluster Orchestration

- **server** and **client machines**
- **decentralized** cluster **membership** and **failure detection** system based on GOSSIP protocol

Basic Information About Machines

```
$ gislab-cluster members # format text or json

server.gis.lab  192.168.15.5:7946
                alive  role=server

c50             192.168.15.50:7946
                alive
                role=client,worker=yes,session-active=user1

c51             192.168.15.51:7946
                left
                role=client,worker=yes
```

Events and Queries

Syntax

```
$ gislab-cluster event <EVENT-NAME>
$ gislab-cluster query <QUERY-NAME>
```

Reboot event

```
$ gislab-cluster event reboot
```

Parallel Commands Execution

Detection of running (alive) client machines

```
$ MACHINES="$(gislab-cluster members
  -status=alive
  -tag role=client ...
)"
```

Parallel installation of Gedit package

```
$ parallel-ssh -H "$MACHINES"

  sudo DEBIAN_FRONTEND=noninteractive
  apt-get install -y --no-install-recommends gedit

  ...
  [1] 23:02:57 [SUCCESS] c51
  [1] 23:02:57 [SUCCESS] c51
  ...
```

Stronger With Each Client Machine

OWS load balancing

```
$ while true; do
    curl "http://ms.gis.lab:90/cgi-bin/qgis_mapserv?
      SERVICE=WMS&REQUEST=GetCapabilities"
  done
```

ms.gis.lab

	Queue			Session rate			Sessions					Bytes	
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out
Frontend				0	61	-	0	1	1 000	2 313		506 547	9 434 727
server	0	0	25	0	13		0	1	25	463	463	101 397	1 888 577
192.168.19.50	0	0	25	0	12		0	1	100	463	463	101 397	1 888 577
192.168.19.51	0	0	25	0	12		0	1	100	463	463	101 397	1 888 577
192.168.19.52	0	0	25	0	12		0	1	100	462	462	101 178	1 884 498
192.168.19.53	0	0	25	0	12		0	1	100	462	462	101 178	1 884 498
Backend	0	0		0	61		0	1	150	2 313	2 313	506 547	9 434 727

Other

Integration Test Suite

```
$ vagrant provision --provision-with test
...
TASK: [basic-server-configuration-test | Test if ordinary
    test user account exists in PostgreSQL]
...
TASK: [service-dns-test | Test 'gis.lab' DNS records are
    resolved]
...
TASK: [service-mapserver-test | Test WMS GetCapabilies
    request with example GIS.lab project]
...
TASK: [service-mapserver-test | Test WMS GetMap request
    with example GIS.lab project]
...
```

Where to Use ?

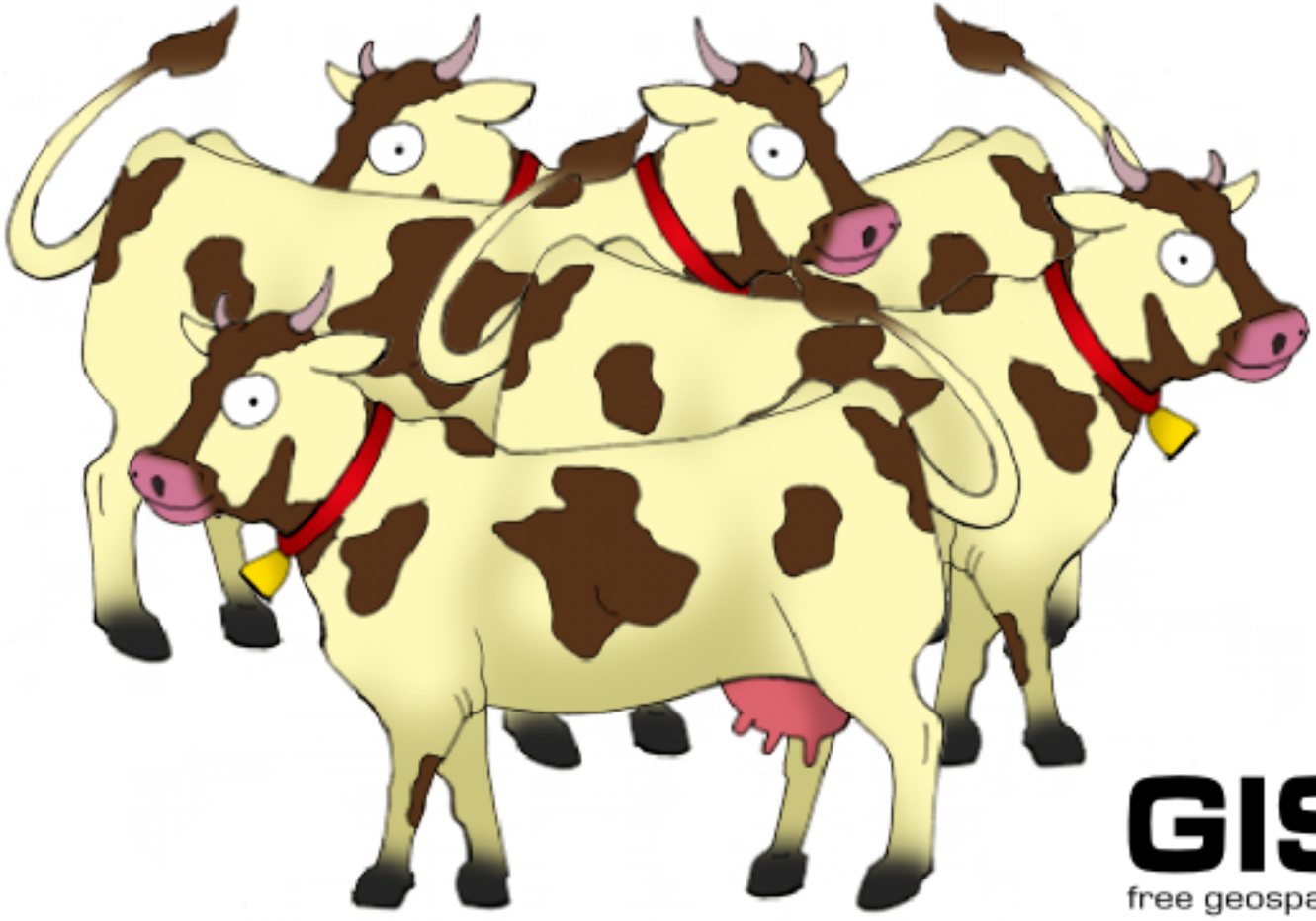
- **schools**: central management, maintenance-free clients
- **small projects**: affordable, complete solution
- **poor countries**: low system requirements, maintenance-free clients
- **crisis management**: portable, instant deployment, no dependencies

Future Plans

- **web administration** interface
- integration of **WPS** services
- better **GRASS** integration
- **web client** rewrite with **OL 3**
- update to **Ubuntu 16.04** and **systemd**

Summary

Spreading Free Software To the World



GIS.lab
free geospatial infrastructure

http://web.gislab.io
wiki:Quick-Start
gis.lab@lists.osgeo.org