

MapProxy

The caching proxy for web map services

Oliver Tonnhofer <olt@omniscale.de>

Omniscale

From Oldenburg in Germany

We focus on:

- OpenSource WebGIS, Server development
- OpenStreetMap WMS
- MapProxy development and support

Content

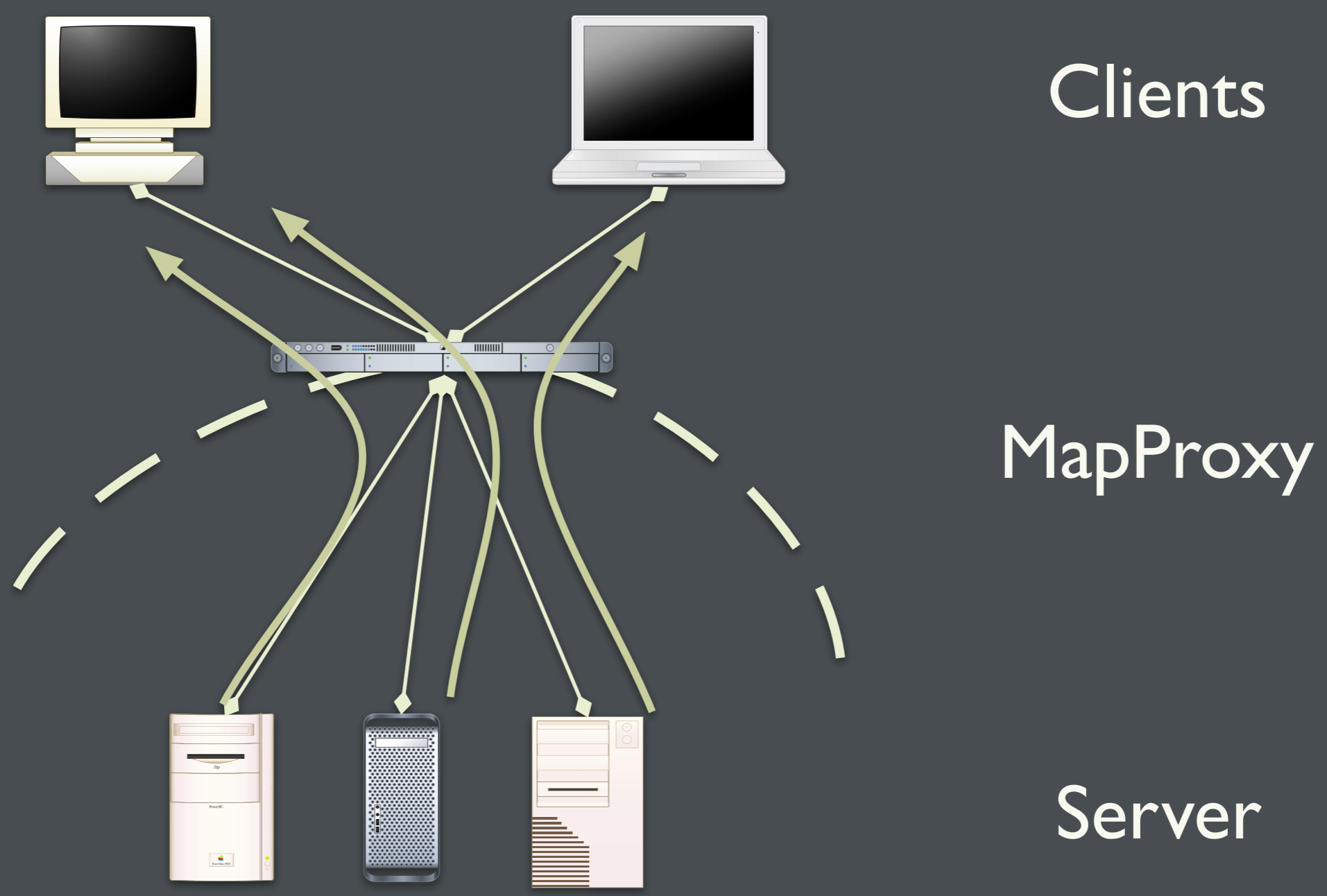
What is it?

About the project

Examples

News

What is it?



Clients

MapProxy

Server

Cache

Improve WMS Performance

Tiled access

Change

Protocols/versions

Image formats

Projection

Enhance

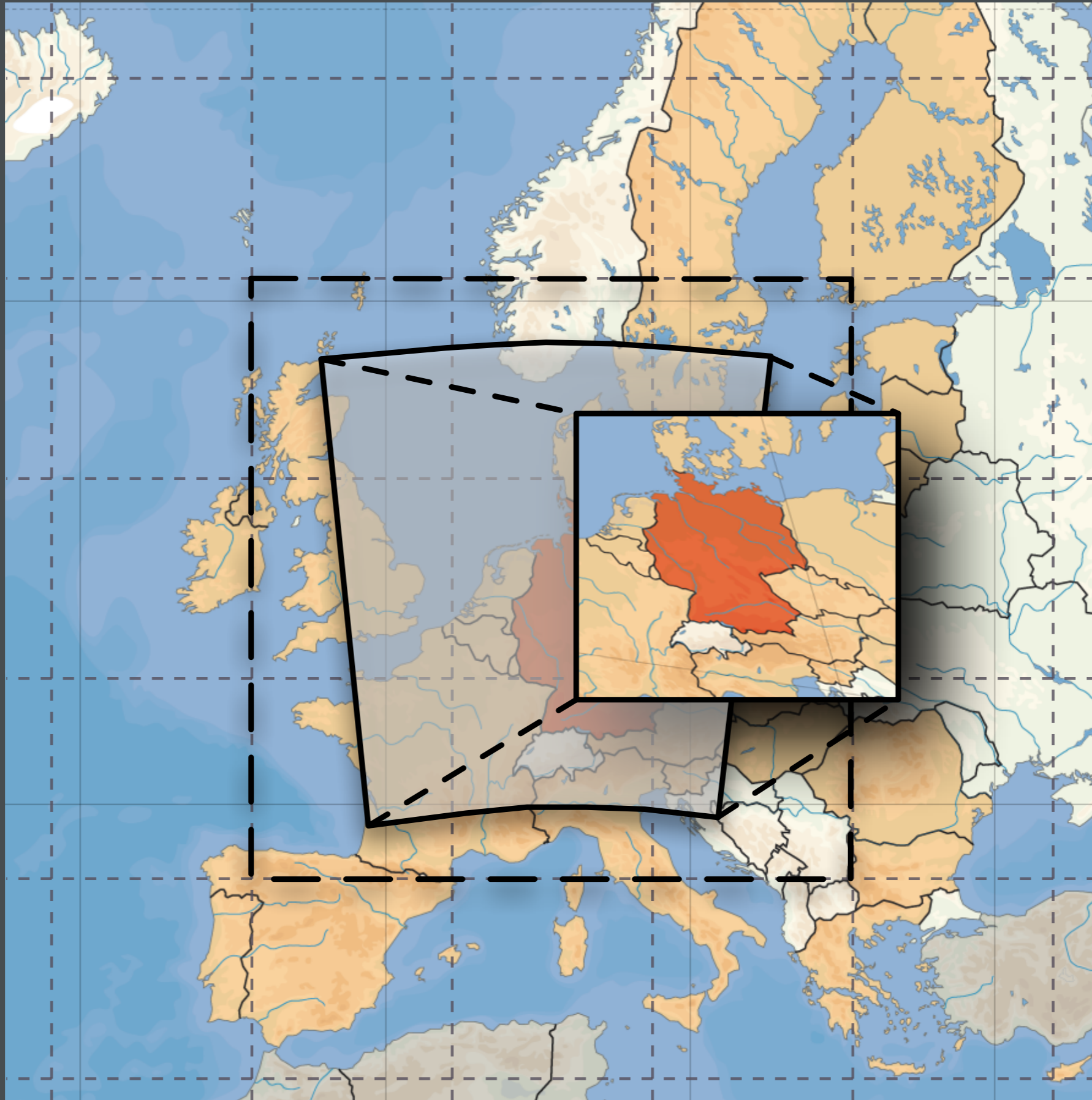
Watermarks

Attribution lines

Combine Layer/Server



Map: Copyright David Liuzzo



Map: Copyright David Liuzzo

Sources

WMS
1.0.0/1.1.1/1.3.0

WMS-C

TMS

Custom Tiles
(Google Maps/Bing/etc.)

Services

WMS
1.0.0/1.1.1/1.3.0

WMS-C

TMS

KML
Super Overlays

About the project

History

Started late 2008

In production since mid 2009

OpenSource since early 2010

Releases

0.8.4 current

0.9.0 upcoming release

Written in	Python
Runs on	Linux/Unix Windows
License	Affero GPL v3
Tested	lots of unit/system tests
Deployment	HTTP or FastCGI

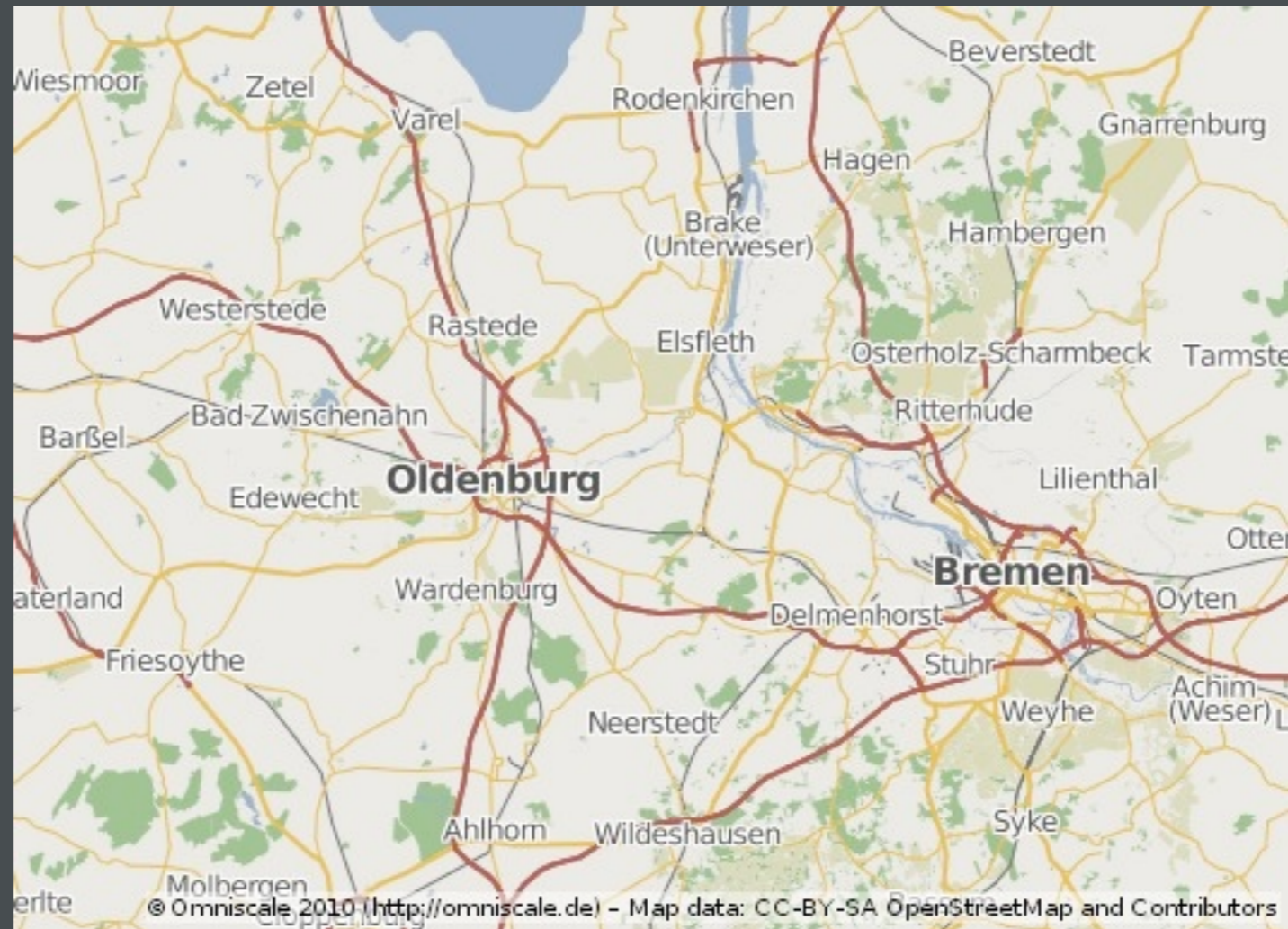
Distributed SCM	✓
Issue tracker	✓
Mailing list	✓
Blog	✓

Example-driven introduction

Getting started

```
% paster create --template \  
  mapproxy_conf foss4g_example  
% cd foss4g_example  
% paster serve etc/develop.ini --reload
```

<http://localhost:8080/service?...>



Example 1

Enhance existing WMS server

BlueMarble
by NASA

Satellite
imagery

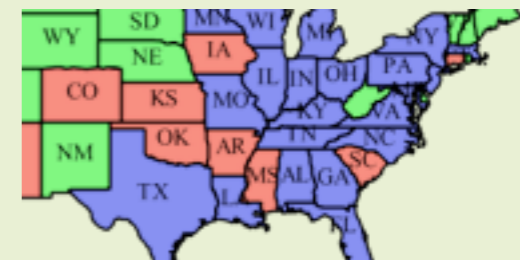
Background



US
population
by demo.opengeo.org

Vector data

Overlay



sources

sources:

us_pop_wms:

type: wms

req:

url: <http://demo.opengeo.org/geoserver/wms?>

layers: 'topp:states'

transparent: true

format: image/png

bluemarble_wms:

type: wms

supported_srs: ['EPSG:4326']

req:

url: <http://neowms.sci.gsfc.nasa.gov/wms/wms?>

layers: BlueMarbleNG

caches & layers

caches:

us_pop_cache:

sources: [bluemarble_wms, us_pop_wms]

grids: [GLOBAL_GEODETTIC]

layers:

world:

title: BlueMarble + US Population

sources: [us_pop_cache]

services

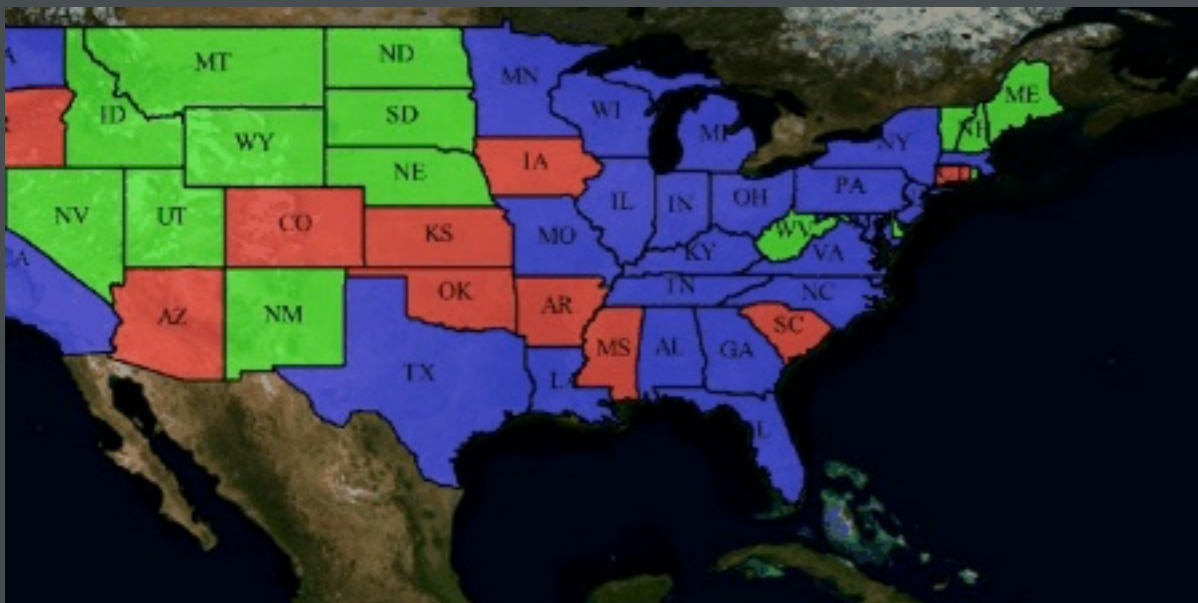
```
services:  
  tms:  
  kml:  
  wms:  
    srs: ['EPSG:900913', 'EPSG:4326']  
    image_formats: [image/png, image/jpeg]  
  md:  
    title: MapProxy WMS Demo  
    abstract: FOSS4G Demo  
    contact:  
      person: Oliver Tonnhofer  
      organization: Omniscale
```

WMS

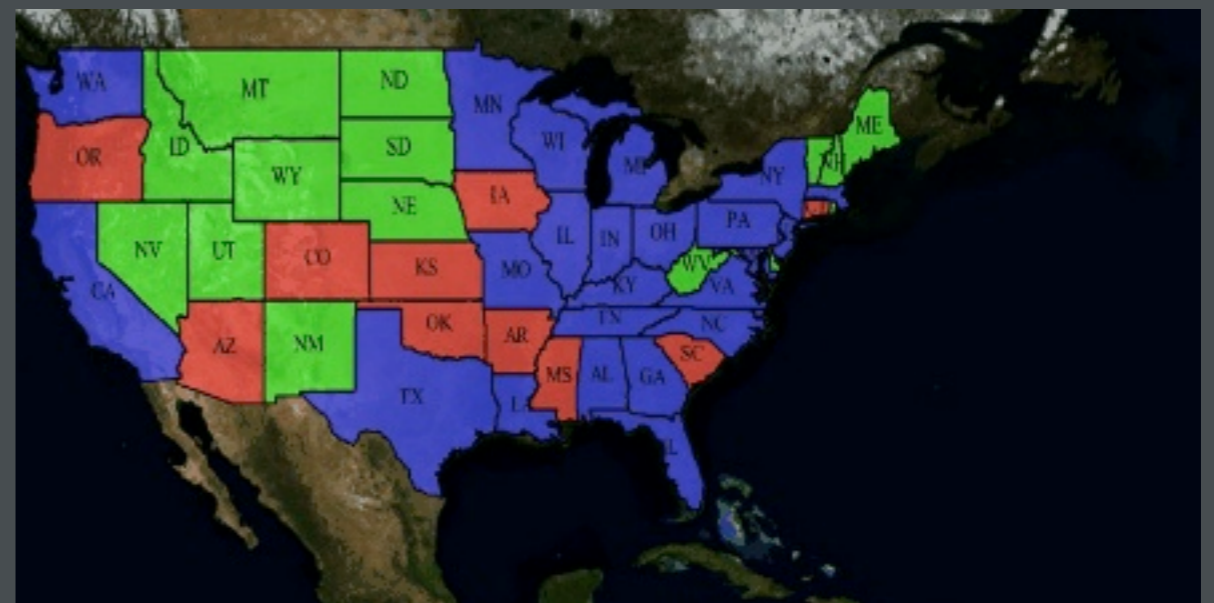
GetCapabilities

<http://localhost:8080/service?request=GetCapabilities&service=WMS>

GetMap



EPSG:4326



EPSG:900913

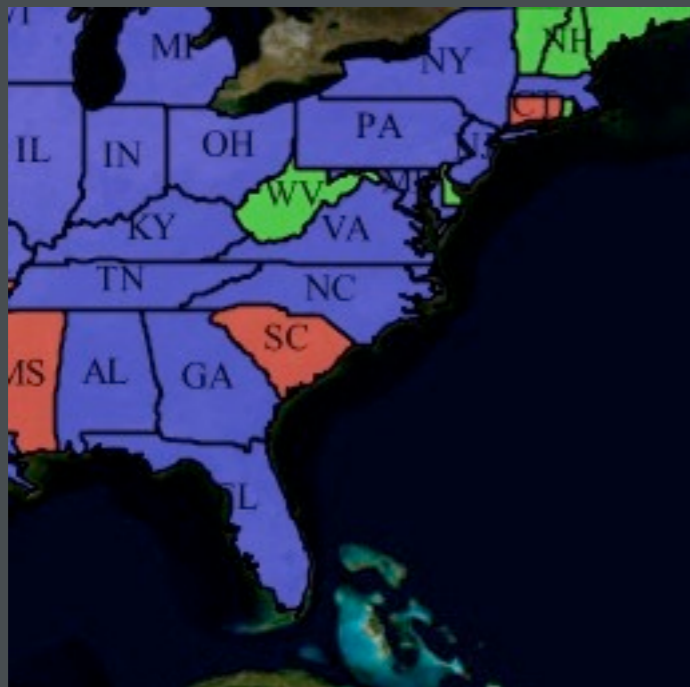
TMS

http://localhost:8080/tms/1.0.0/world_EPSG4326

```
<?xml version="1.0" encoding="UTF-8" ?>
<TileMap version="1.0.0">
  <Title>BlueMarble + Country Borders</Title>
  <Abstract></Abstract>
  <SRS>EPSG:4326</SRS>
  <BoundingBox minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0" />
  <Origin x="-180.0" y="-90.0" />
  <TileFormat width="256" height="256" mime-type="image/jpeg" extension="jpeg" />
  <TileSets profile="global-geodetic">
    <TileSet href="http://localhost:8080/tms/1.0.0/world_EPSG4326/0"
      units-per-pixel="0.703125" order="0" />
    <TileSet href="http://localhost:8080/tms/1.0.0/world_EPSG4326/1"
      units-per-pixel="0.3515625" order="1" />
    ...
  </TileSets>
</TileMap>
```


TMS

http://localhost:8080/tms/world_EPSG4326/3/4/5.jpeg



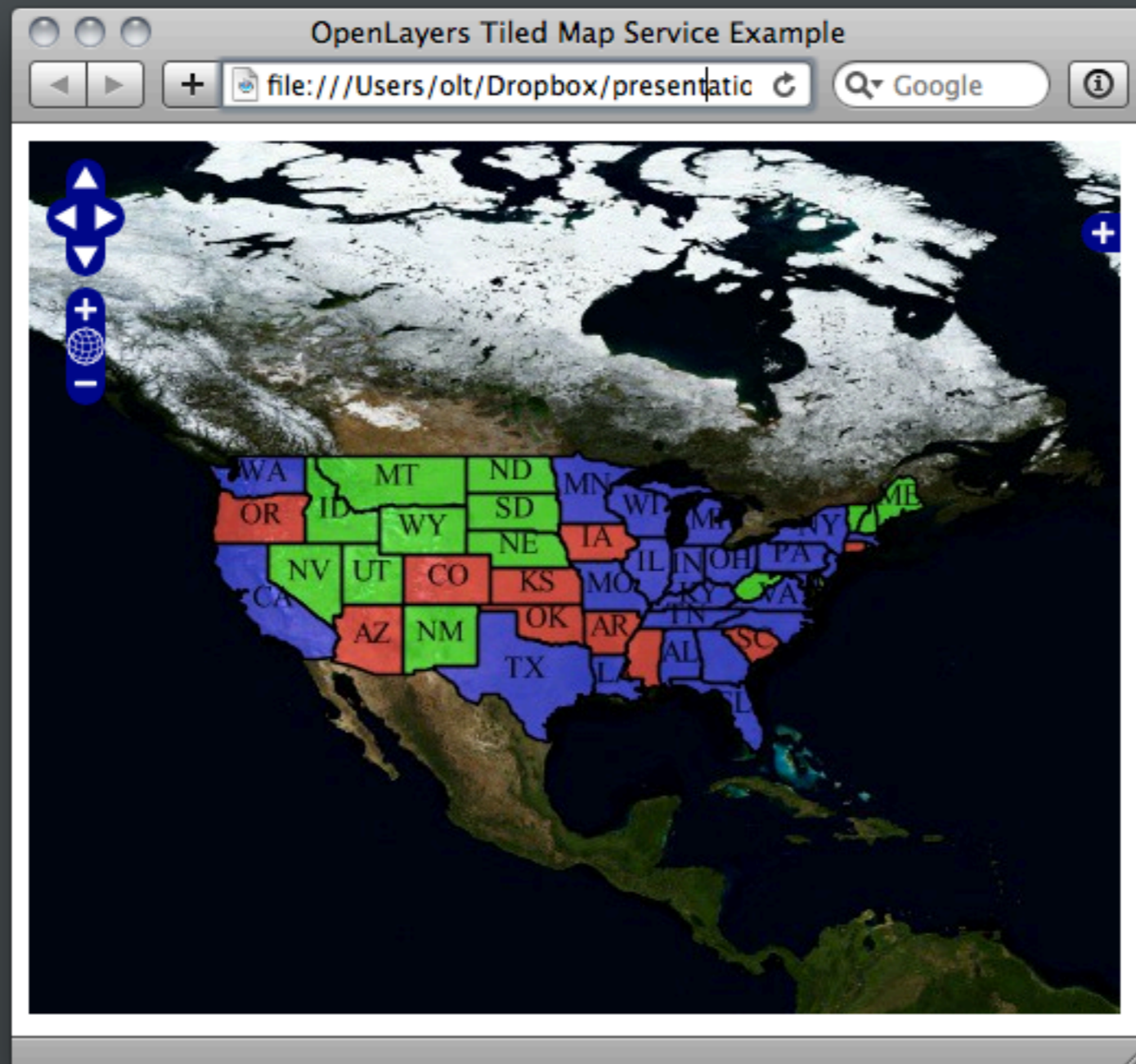
HTTP cache control

```
Etag: 5c4a1f9bc1b95b283a512357ff42fc  
Cache-control: max-age=259200 public  
...
```

HTTP/1.1 304 Not Modified

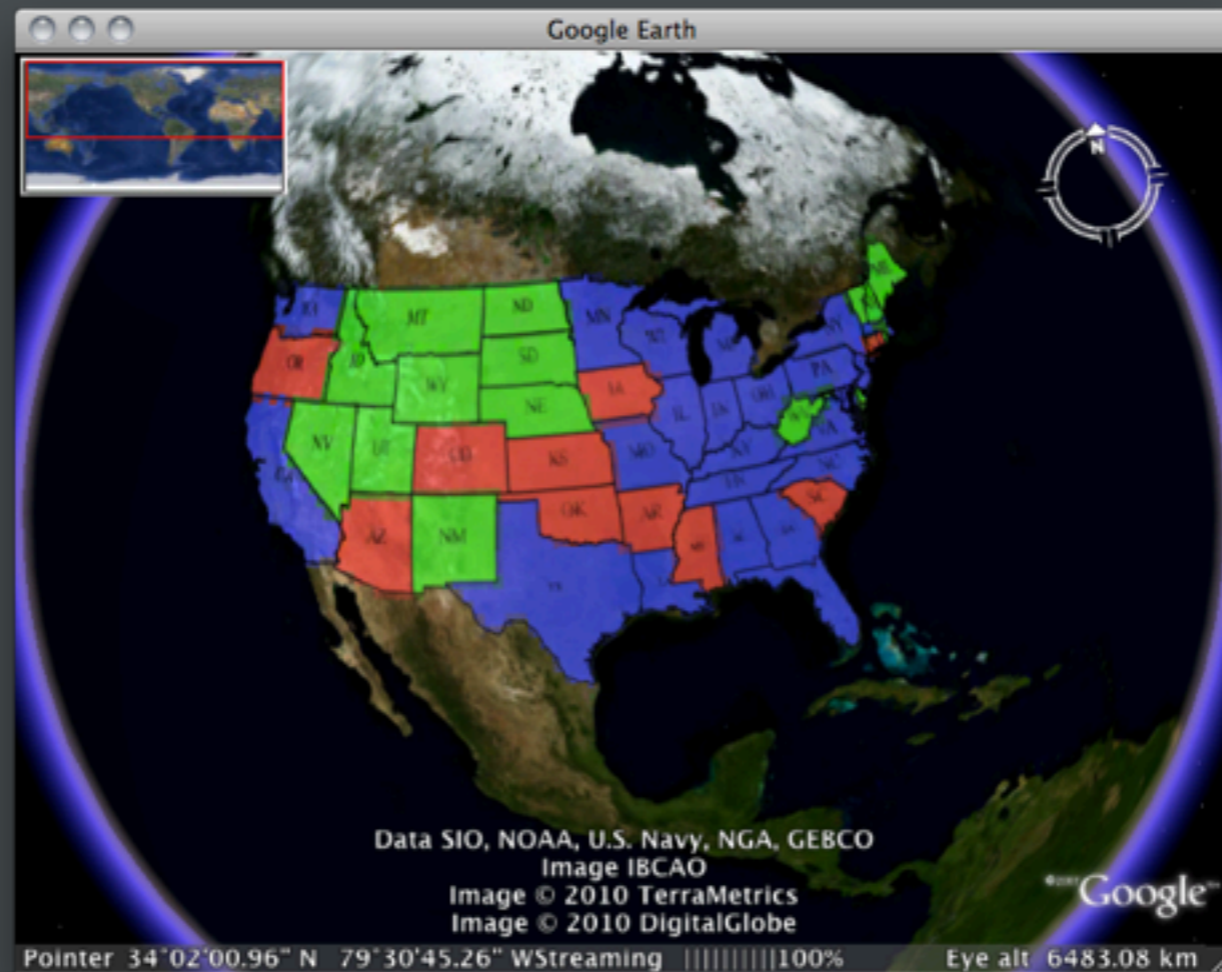
TMS

http://localhost:8080/tms/world_EPSG4326/3/4/5.jpeg



KML

http://localhost:8080/kml/world_EPSG4326/0/0/0.kml



Spherical mercator?

Google Maps, Bing, OSM, etc.

```
caches:
```

```
  us_pop_cache:
```

```
    sources: [bluemarble_wms, us_pop_wms]
```

```
    grids: [GLOBAL_GEODETTIC, GLOBAL_MERCATOR]
```

Reprojects BlueMarble from EPSG:4326

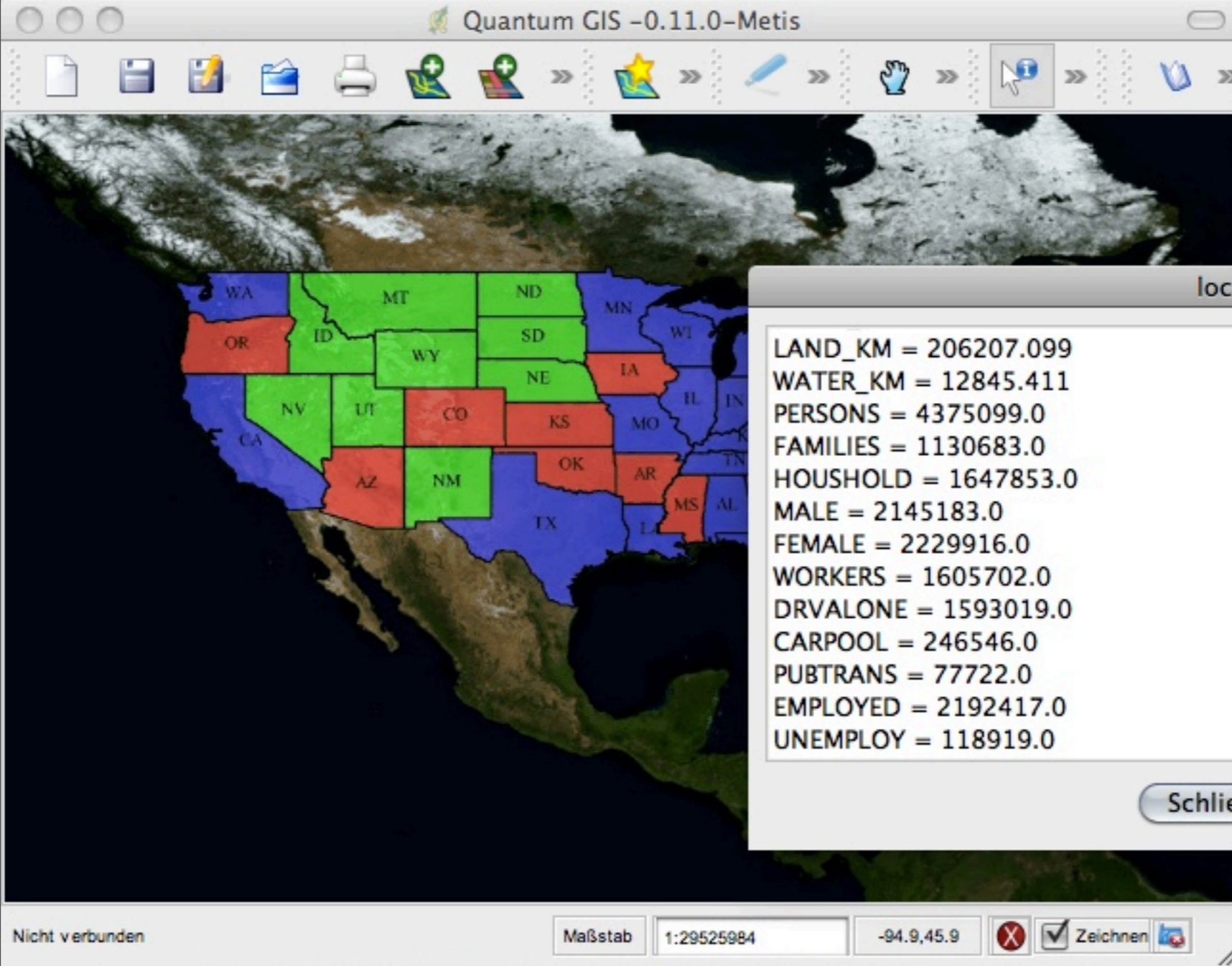
adds tms/1.0.0/world_EPSG900913

Featureinfo?

```
sources:  
  us_pop_wms:  
    type: wms  
    wms_opts:  
      featureinfo: true  
    req:  
      url: http://demo.opengeo.org/geoserver/wms?  
      layers: 'topp:states'  
      transparent: true  
      format: image/png
```

world layer is now queryable

GetFeatureInfo



The screenshot shows the Quantum GIS -0.11.0-Metis interface. The main map displays the United States with state boundaries and labels. A popup window titled 'local' is open, displaying the following data for a selected feature:

LAND_KM	=	206207.099
WATER_KM	=	12845.411
PERSONS	=	4375099.0
FAMILIES	=	1130683.0
HOUSHOLD	=	1647853.0
MALE	=	2145183.0
FEMALE	=	2229916.0
WORKERS	=	1605702.0
DRVALONE	=	1593019.0
CARPPOOL	=	246546.0
PUBTRANS	=	77722.0
EMPLOYED	=	2192417.0
UNEMPLOY	=	118919.0

The popup window has a 'Schließen' button at the bottom right. The bottom status bar shows 'Nicht verbunden', 'Maßstab 1:29525984', and coordinates '-94.9,45.9'. There are also icons for 'Zeichnen' and a red 'X' icon.

Example 2

Improve WMS performance

OpenStreetMap WMS
with CPU/IO intensive
rendering

- Large dataset
- Lots of details
- Rendering with anti-aliasing



sources

```
sources:  
  osm_wms:  
    type: wms  
    req:  
      url: http://localhost:81/mapnik?  
      layers: osm
```


caches

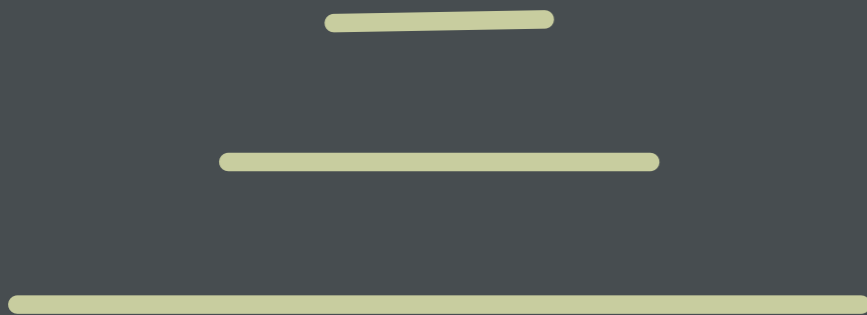
```
caches:  
  osm_cache:  
    sources: [osm_wms]  
    grids: [osm_germany]  
    link_single_color_images: true
```

Caching

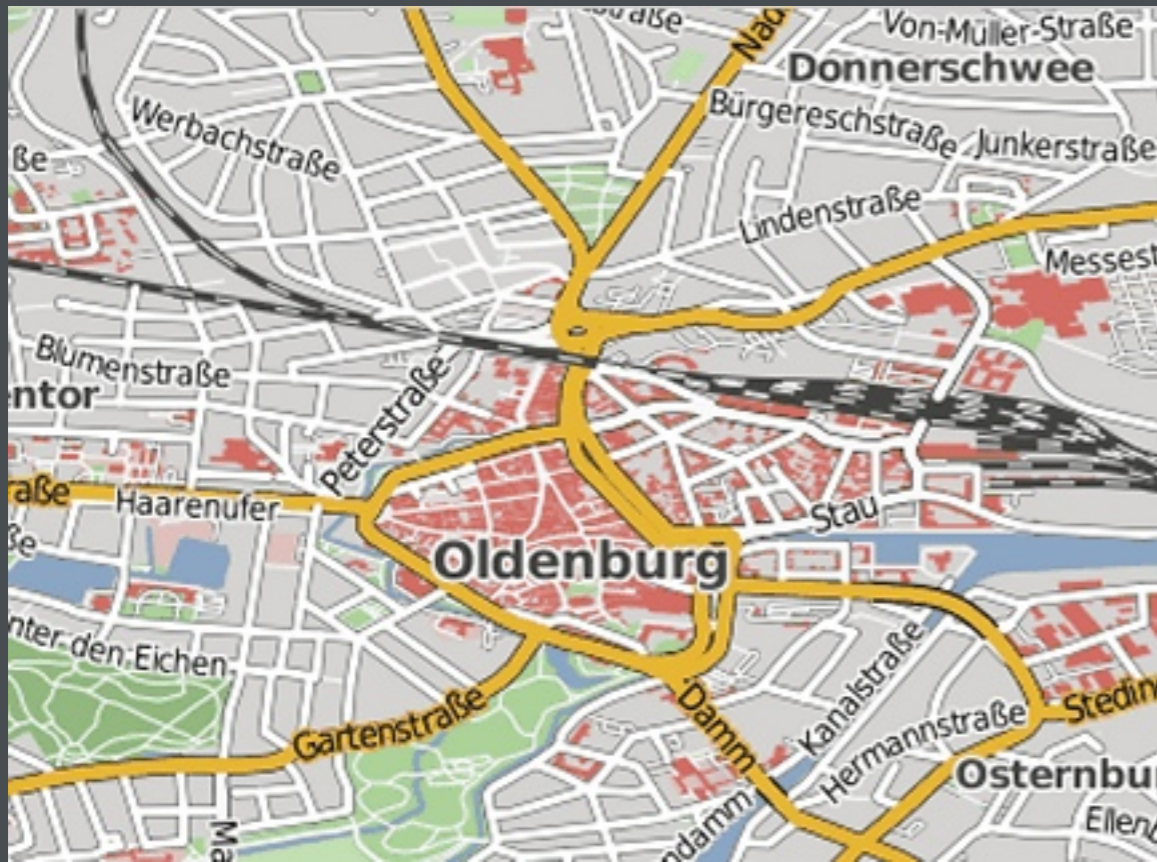
MapProxy allows free-zooming

but

caches at fixed resolutions



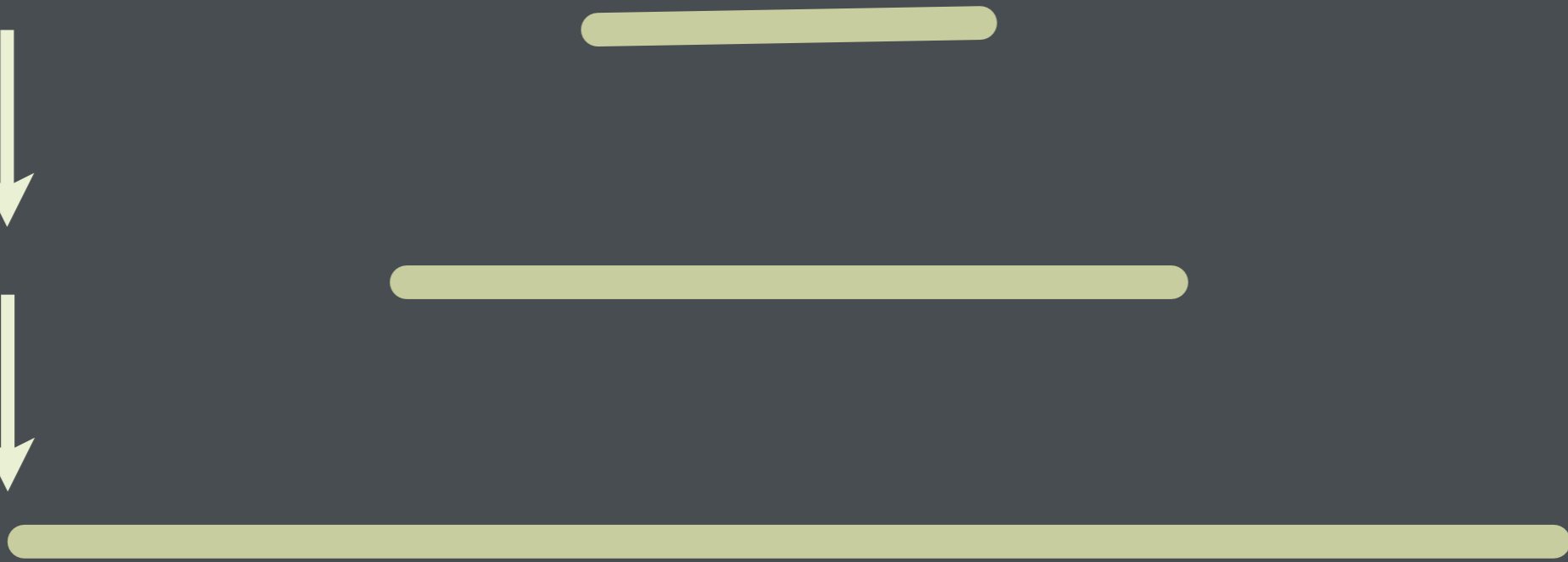
Vector scaling



x2



x2





Vector scaling



$$\sqrt{2}$$



grid

grids:

osm_germany:

srs: 'EPSG:25832'

bbox: [-3.82, 46.18, 24.91, 55.23]

bbox_srs: 'EPSG:4326'

res_factor: sqrt2

More grid options

<code>min_resolution</code>	<code>max_resolution</code>	<code>num_levels</code>
<code>tile_size</code>	<code>res</code>	<code>res_factor</code>

Cache pre-generation

```
% mapproxy-seed -f etc/mapproxy.yaml \  
etc/seed.yaml -c 8
```

Cache pre-generation

```
% mapproxy-seed -f etc/mapproxy.yaml \  
etc/seed.yaml -c 8
```



Multiprocessing

seed.yaml

```
views:  
  germany:  
    bbox: [7.36, 46.33, 14.28, 56.09]  
    bbox_srs: 'EPSG:4326'  
    level: [0, 14]  
    srs: ['EPSG:900913', 'EPSG:4326']  
  
seeds:  
  osm:  
    views: ['germany']  
    remove_before:  
      month: 1  
      days: 3
```



Colorado



not-Colorado

seed.yaml

Any OGR source (Shapefile, PostGIS, etc.)

```
germany:  
  ogr_datasource: 'shps/world_boundaries_m.shp'  
  ogr_where: 'CNTRY_NAME = "Germany" '  
  ogr_srs: 'EPSG:900913'  
  level: [0, 14]  
  srs: ['EPSG:900913', 'EPSG:4326']
```

WKT polygons <http://mapproxy.org/static/polygons/>

```
germany:  
  polygons: 'polygons/GM.txt'  
  polygons_srs: EPSG:900913  
  level: [0, 14]  
  srs: ['EPSG:900913', 'EPSG:4326']
```

Seed strategy

Normal

level per level, top to bottom, left to right

Seed strategy

Normal

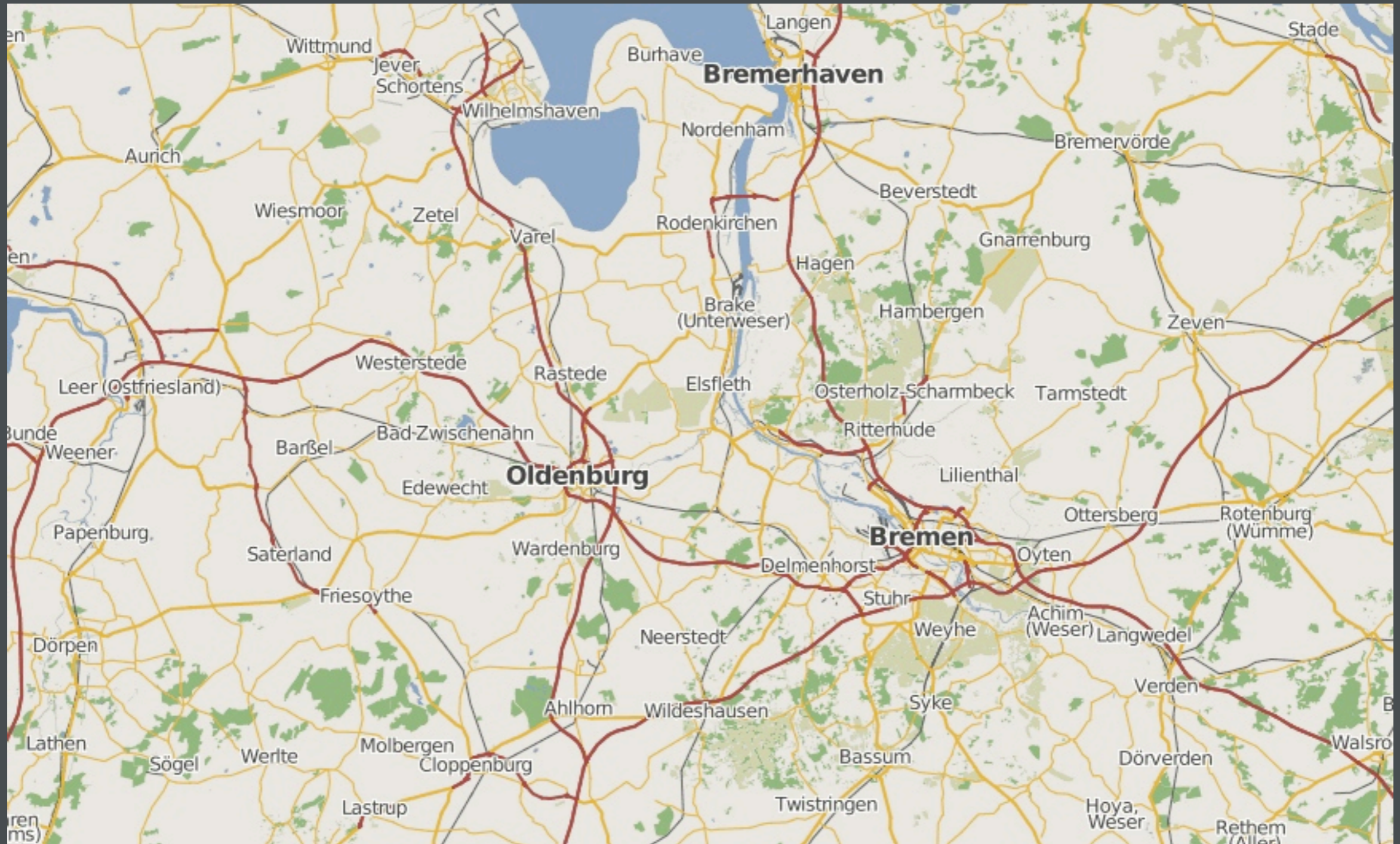
level per level, top to bottom, left to right

mapproxy-seed

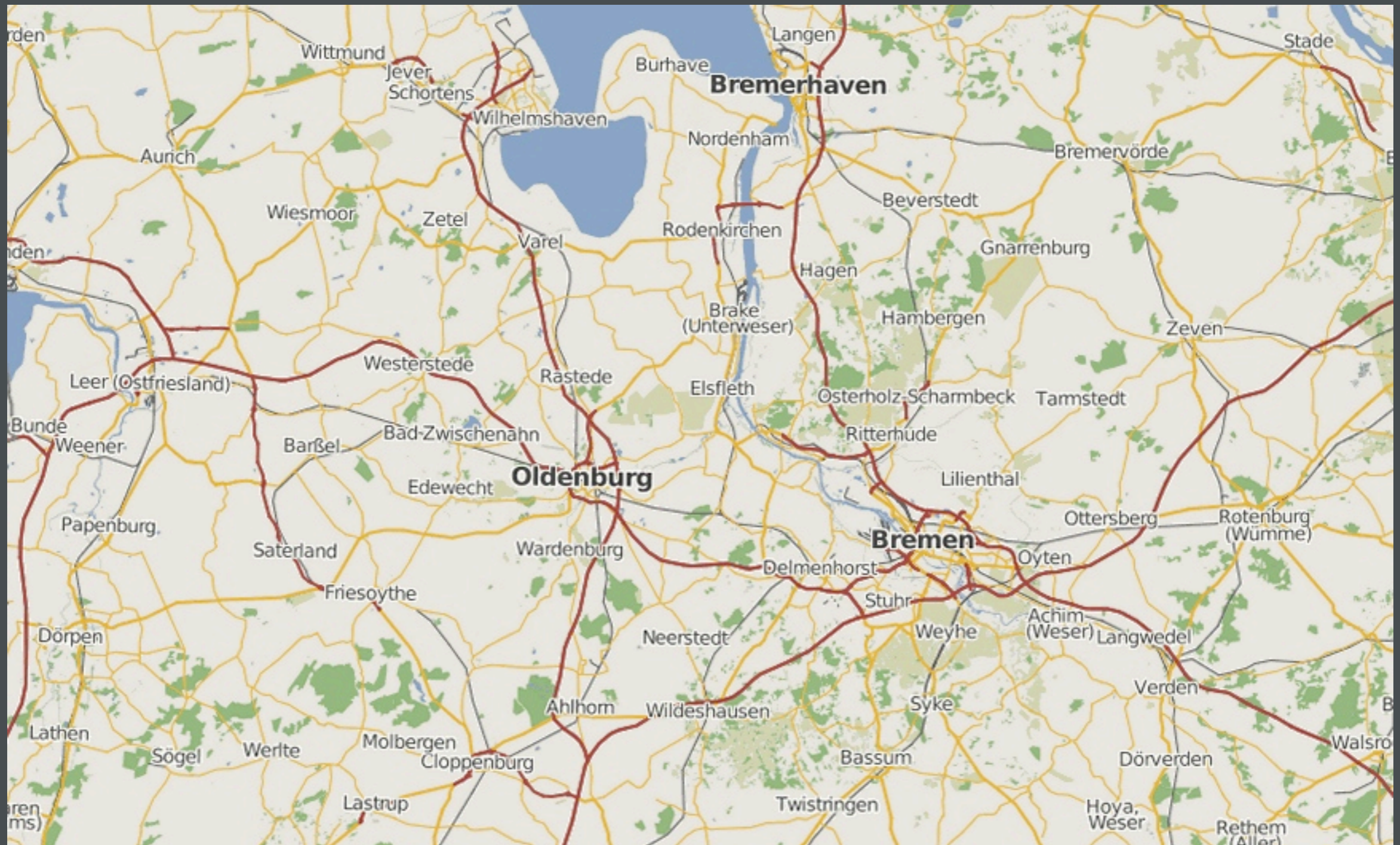
recursive depth-first traversal

works with the OS/DB cache

PNG8 870x530 = 50ms



Reprojected PNG8 870x530 = 130ms



Single Tile < Ims



News

Python Image Library

Really fast C-based imaging library

only Issues: slow PNG performance
no transparency PNG8

Python Image Library

Really fast C-based imaging library

Now fixed! 10x faster – same as libjpeg
to be included in next PIL release **full transparency for PNG8**

<http://bitbucket.org/olt/pil-117-fastpng>

Jython

Python implementation for JVM

no support for C extensions

Jython

Python implementation for JVM

no support for C extensions

Jython wrapper with

Java ImageIO/JAI

GeoTools

Jython

Python implementation for JVM

no support for C extensions

Jython wrapper with
Java ImageIO/JAI
GeoTools

MapProxy as .war
alpha release



NEW

Thank you.

Oliver Tonnhofer <olt@omniscale.de>

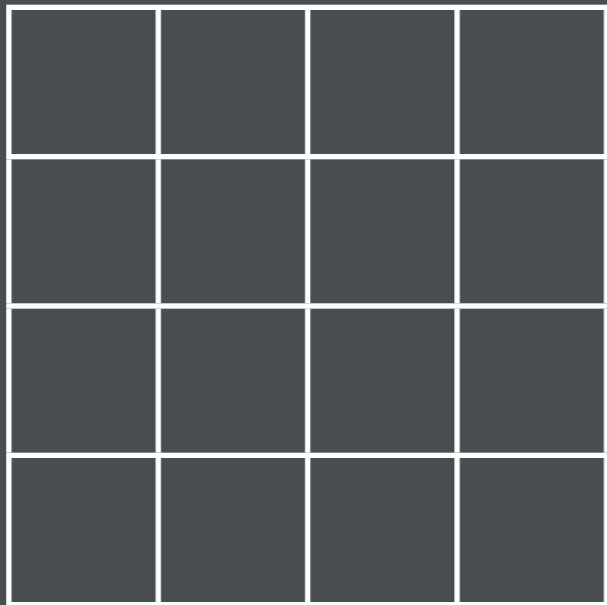
Visit <http://mapproxy.org>

Join [http://lists.osgeo.org/mailman/
listinfo/mapproxy](http://lists.osgeo.org/mailman/listinfo/mapproxy)

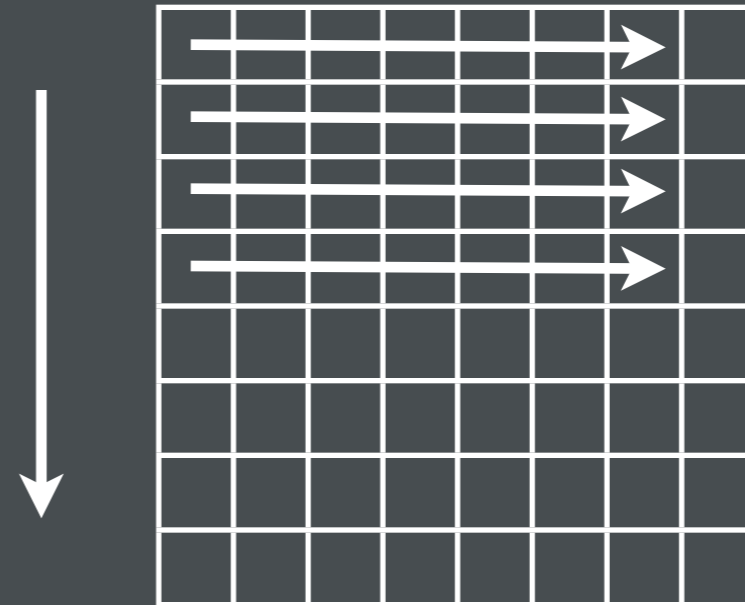
Read <http://blog.mapproxy.org>

Backup tiles below

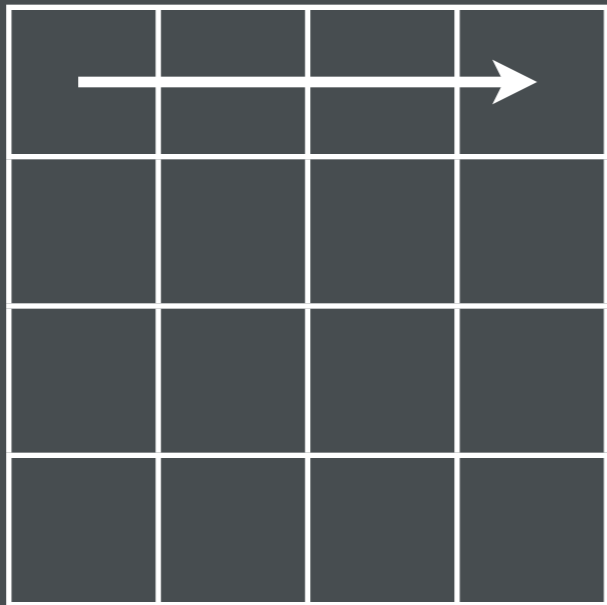
Level 3



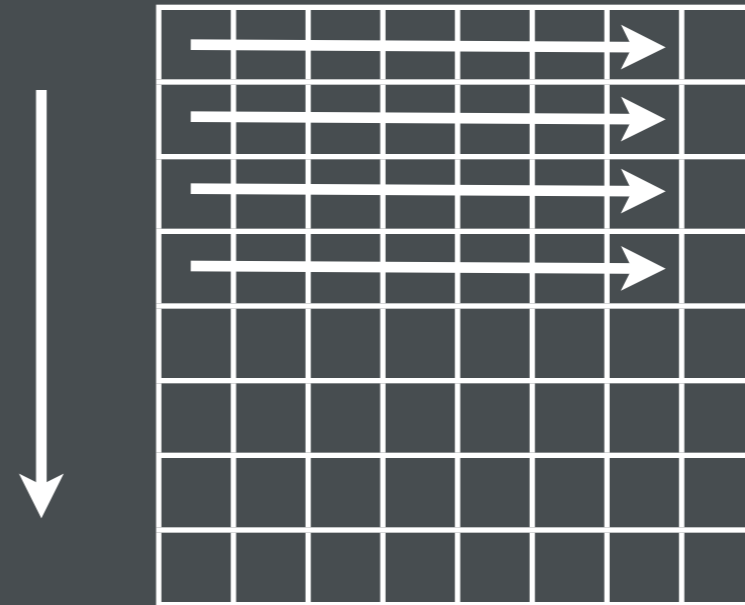
Level 4



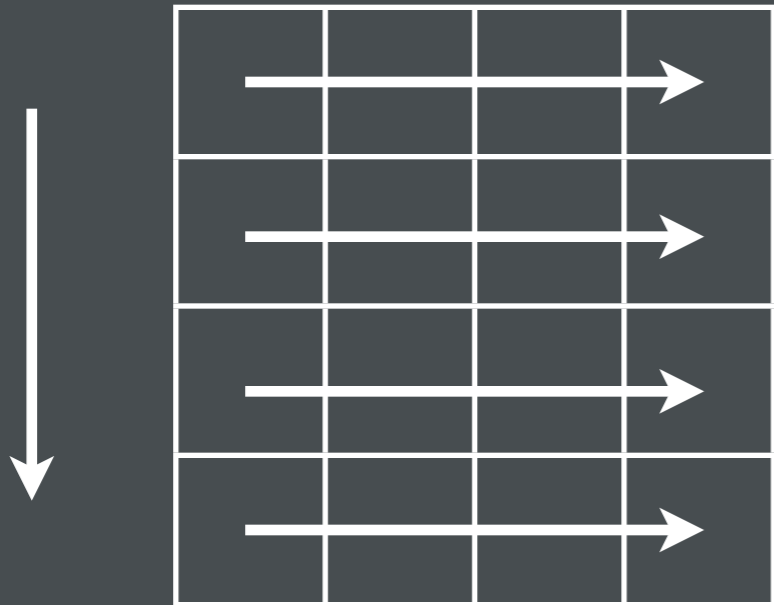
Level 3



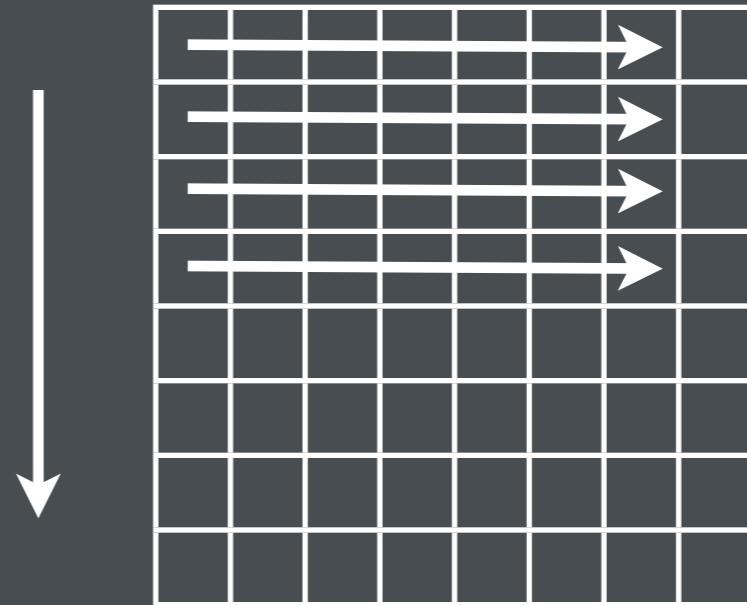
Level 4



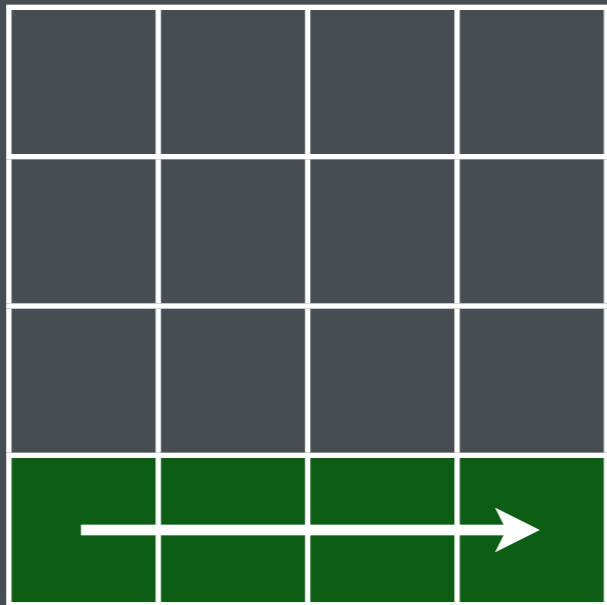
Level 3



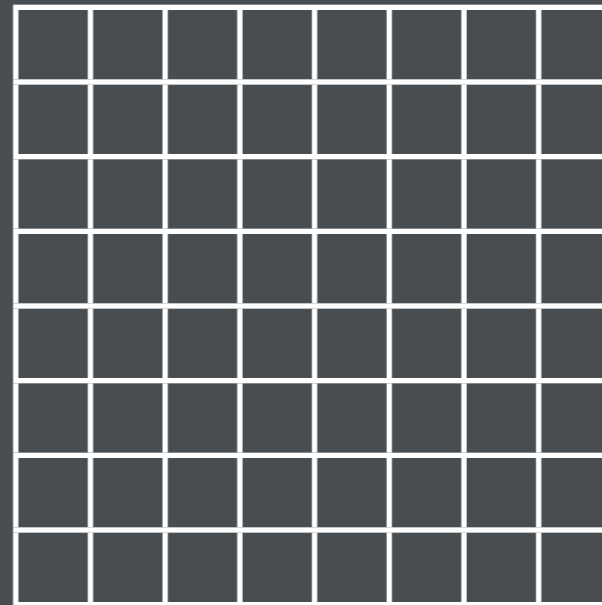
Level 4



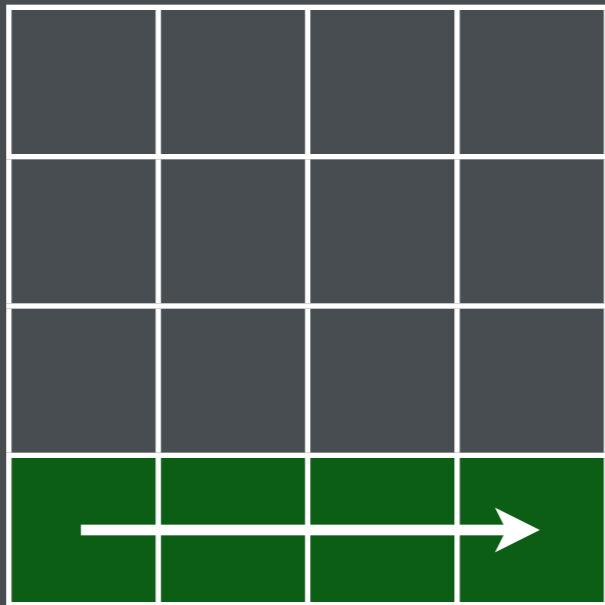
Level 3



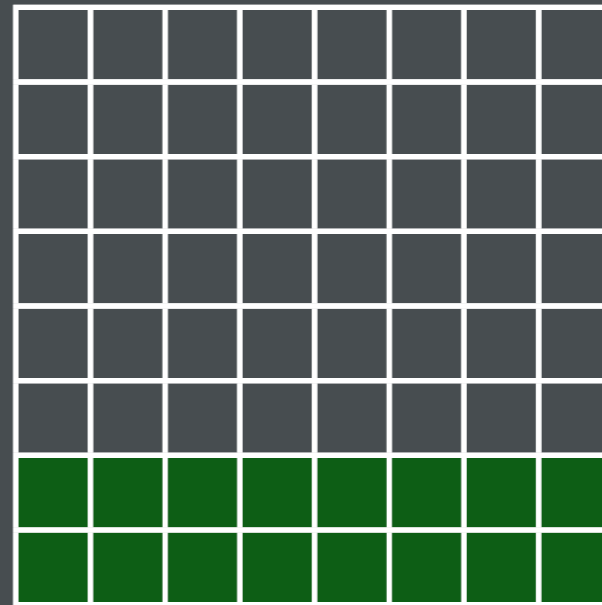
Level 4



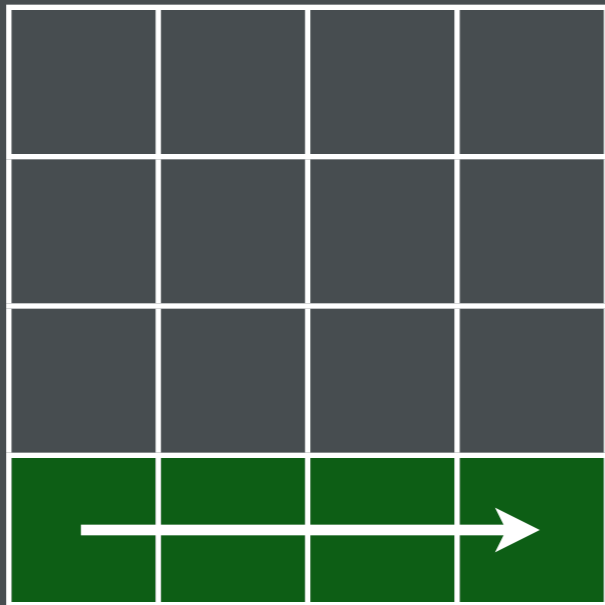
Level 3



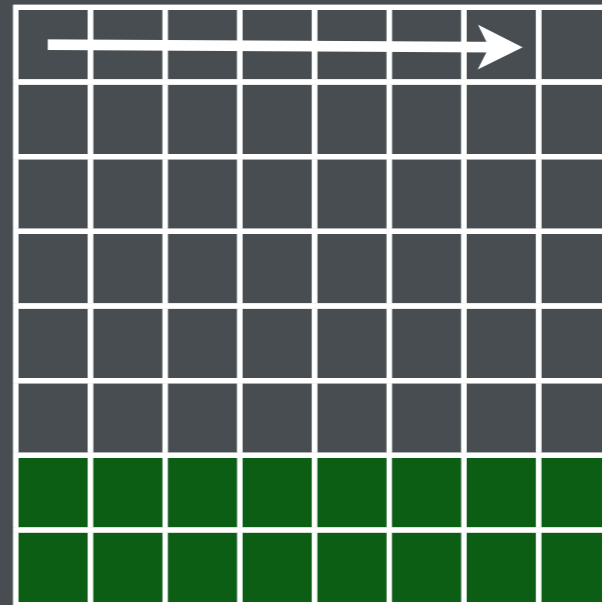
Level 4



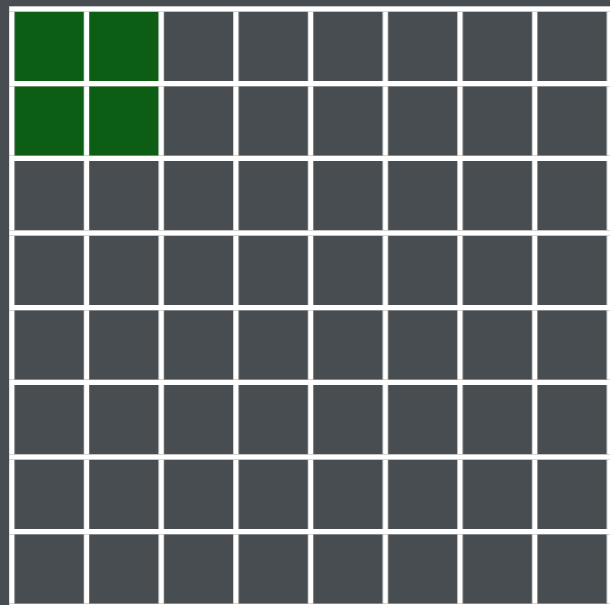
Level 3



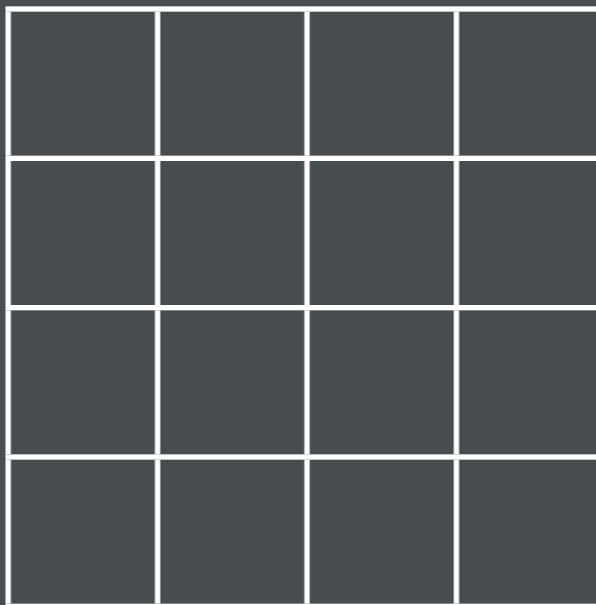
Level 4



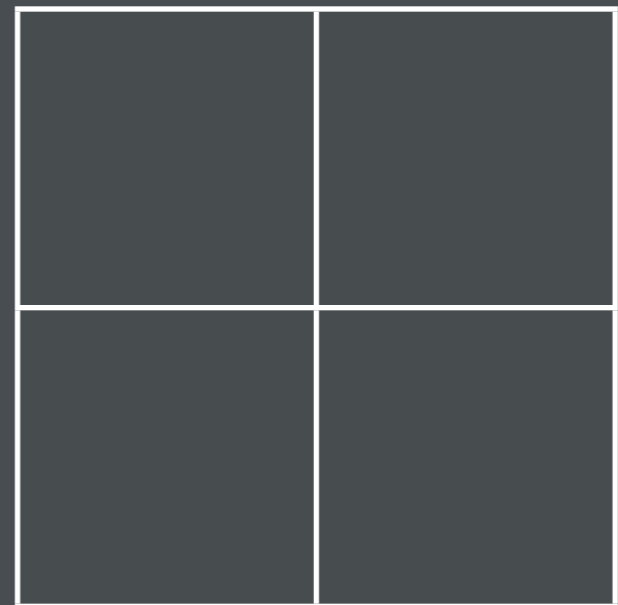
Level 4



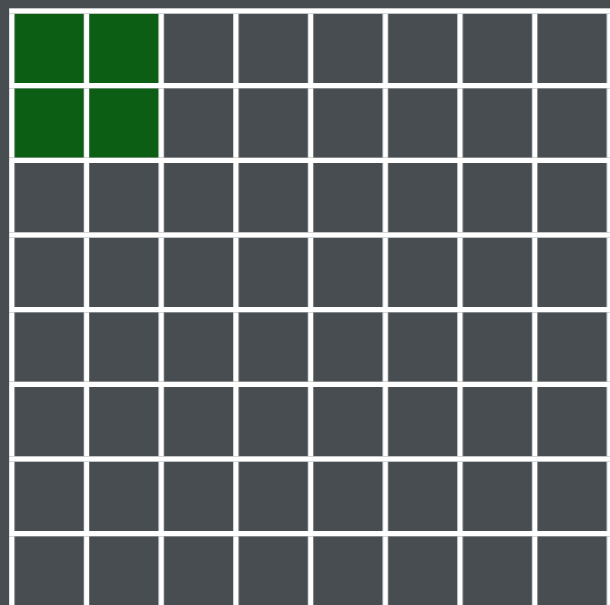
Level 3



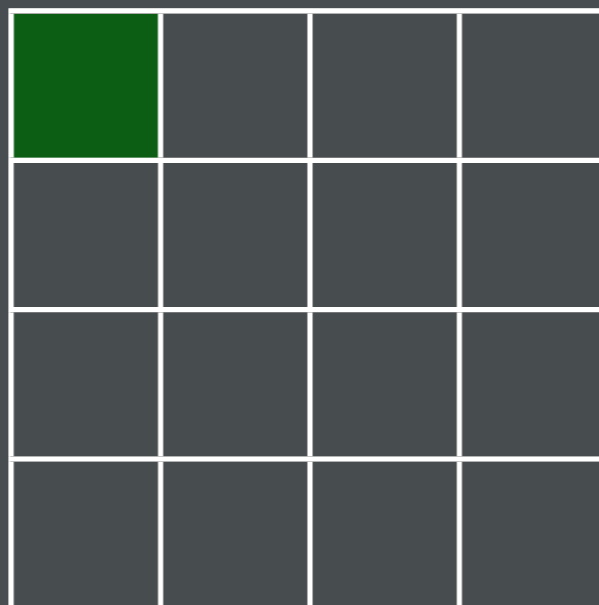
Level 2



Level 4



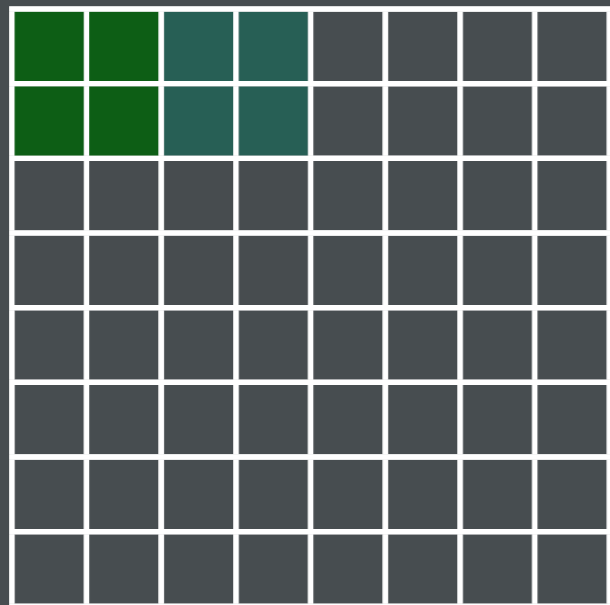
Level 3



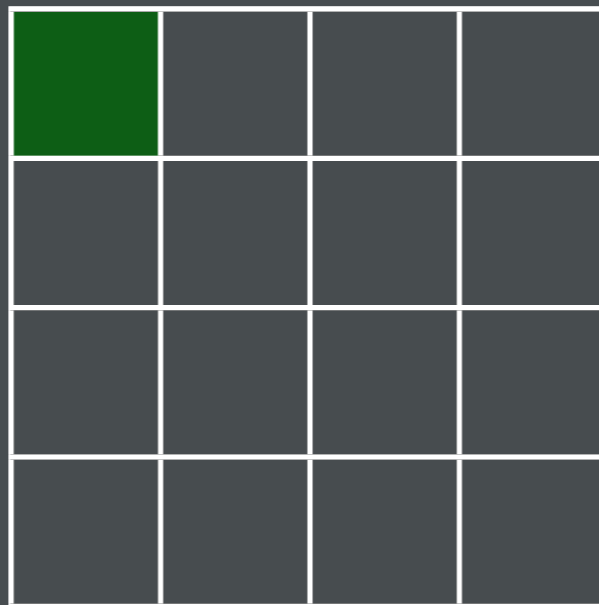
Level 2



Level 4



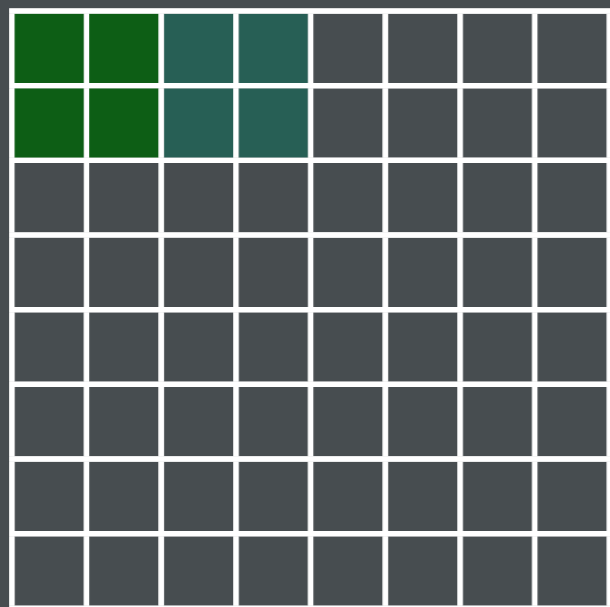
Level 3



Level 2



Level 4



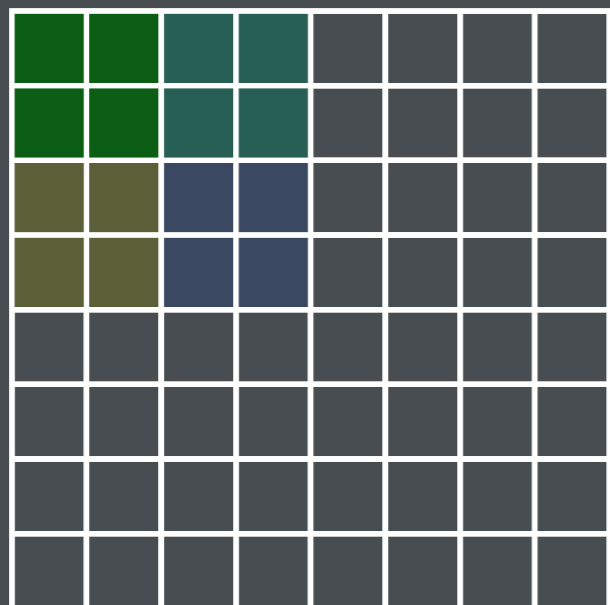
Level 3



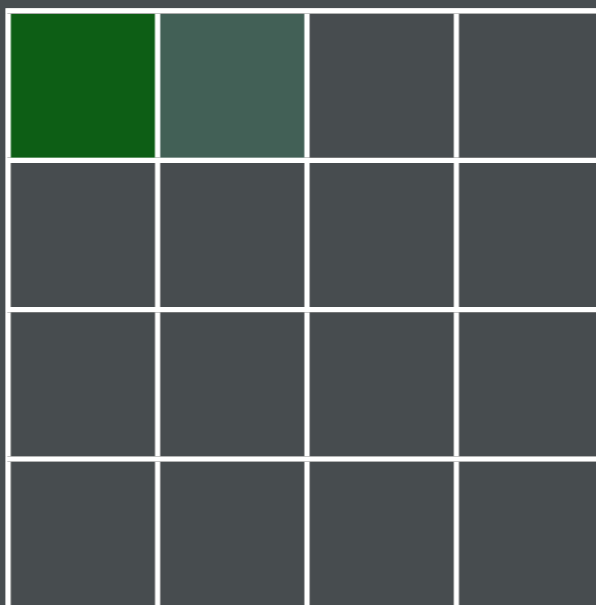
Level 2



Level 4



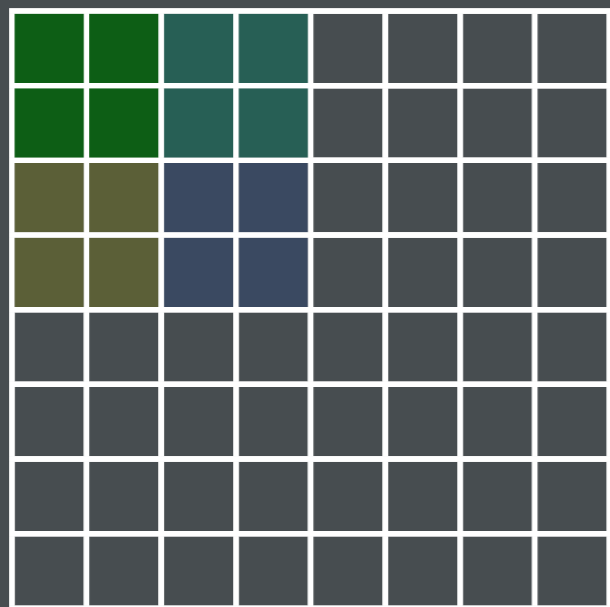
Level 3



Level 2



Level 4



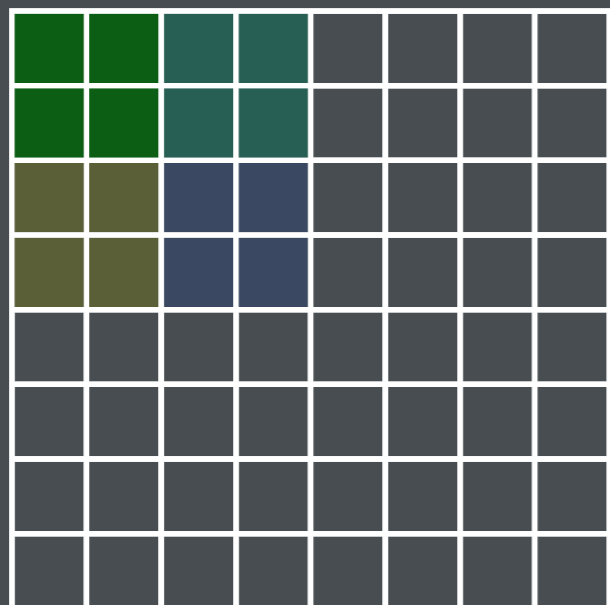
Level 3



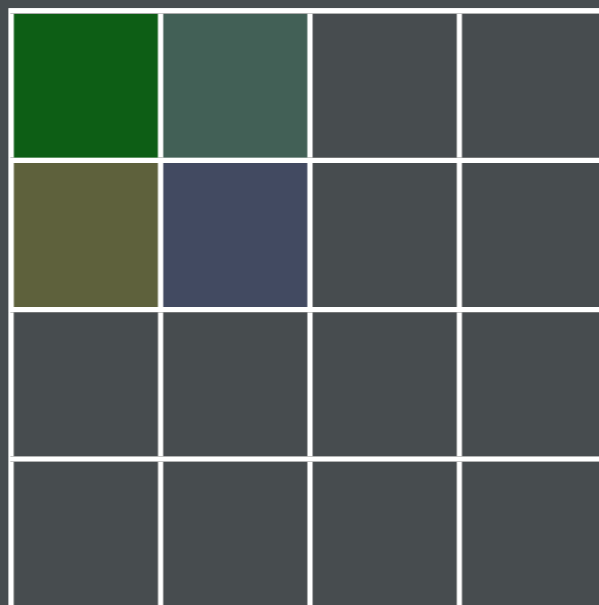
Level 2



Level 4



Level 3



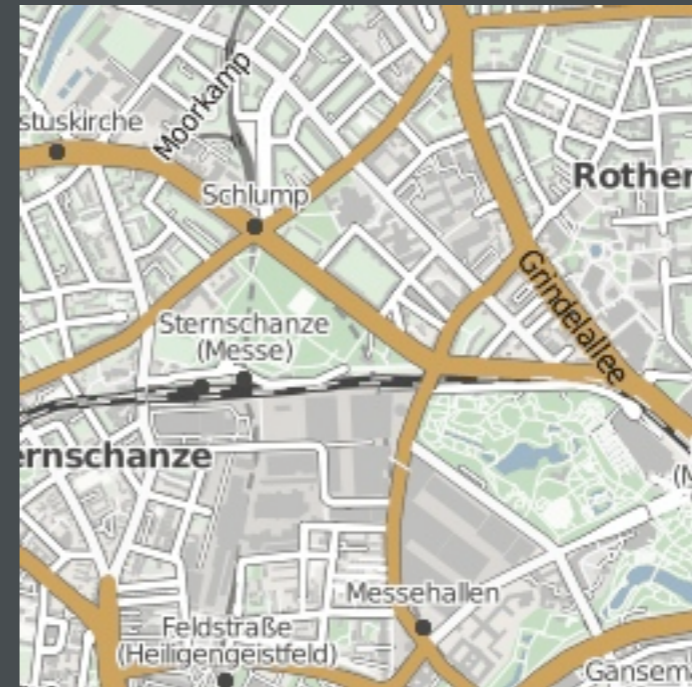
Level 2



Level 5



Level 14

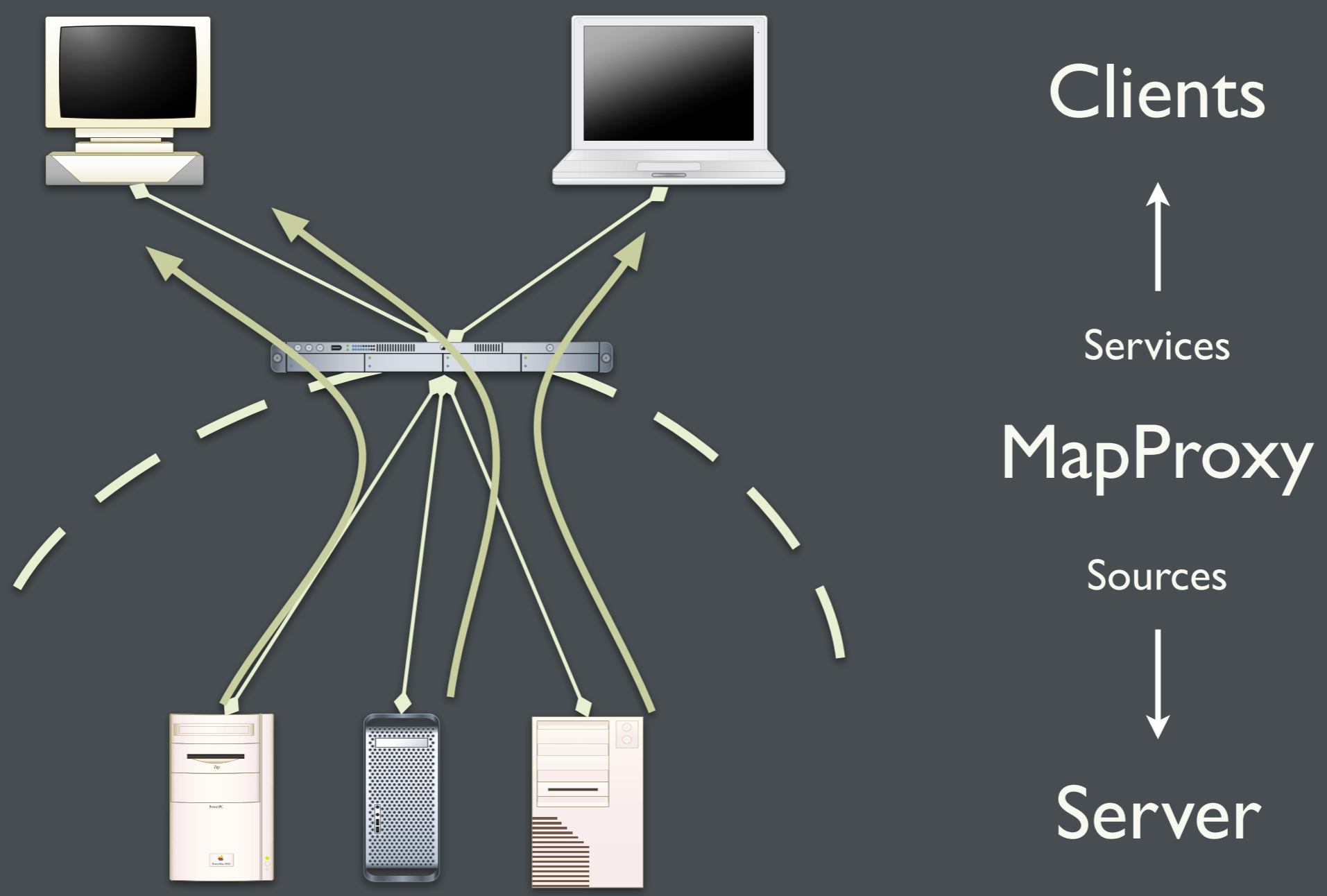


Seed with avg. 150 tiles/second
on a Quad Core (i7@2.6GHz)

Deployment

	Develop	Production
Protocol	HTTP	FastCGI or HTTP
Server	embedded	Apache, nginx, lighttpd, varnish, squid, ...

with auto
reloading



Clients



Services

MapProxy

Sources



Server

reprojection

merge layers

on request

before caching

watermarks

on-the-fly

fast PNG encoding

free zooming

advanced seed strategy

custom color quantizer improved vector support

“empty tiles” handling

KML

TMS

WMS

seed polygon areas

WMS-C

WKT

shapefiles