# Setting up a Spatial Data Infrastructure (SDI) with Open Source Software using OSGeoLive

Astrid Emde, Angelos Tzotzos, Nicolas Roelandt

FOSS4G 2019 Bucharest

# SDI Workshop

- **What is a Spatial Data Infrastructure (SDI)**

- **Components of an SDI**

- **Data**

- **Services & OGC Standards**

- **Clients**

- **Metadata**

- **Get to know OSGeo Software**

# Workshop with OSGeoLive

- **OSGeoLive 13.0 (August 2019)**



- **OSGeoLive http://live.osgeo.org**

- **Download Data & Presentation**

  **http://trac.osgeo.org/osgeolive/wiki/Live_GIS_Workshop_Install**

# What is a SDI?

# What is an SDI?

A spatial data infrastructure (SDI) is a data infrastructure implementing a framework of geographic data, metadata, users and tools that are interactively connected in order to use spatial data in an efficient and flexible way. Another definition is "the technology, policies, standards, human resources, and related activities necessary to acquire, process, distribute, use, maintain, and preserve spatial data".

Source: Wikipedia
https://en.wikipedia.org/wiki/Spatial_data_infrastructure

# SDI

- **Data infrastructure that provides geographic data and metadata**

- **Data network to exchange data**

- **Data producer and data user are connected via a physical network f.e internet / intranet**

- **it is accessible for several users**

- **Users can use different tools for different processes**

- **It helps you to make data accessible, maintainable and findable throughout your organization**
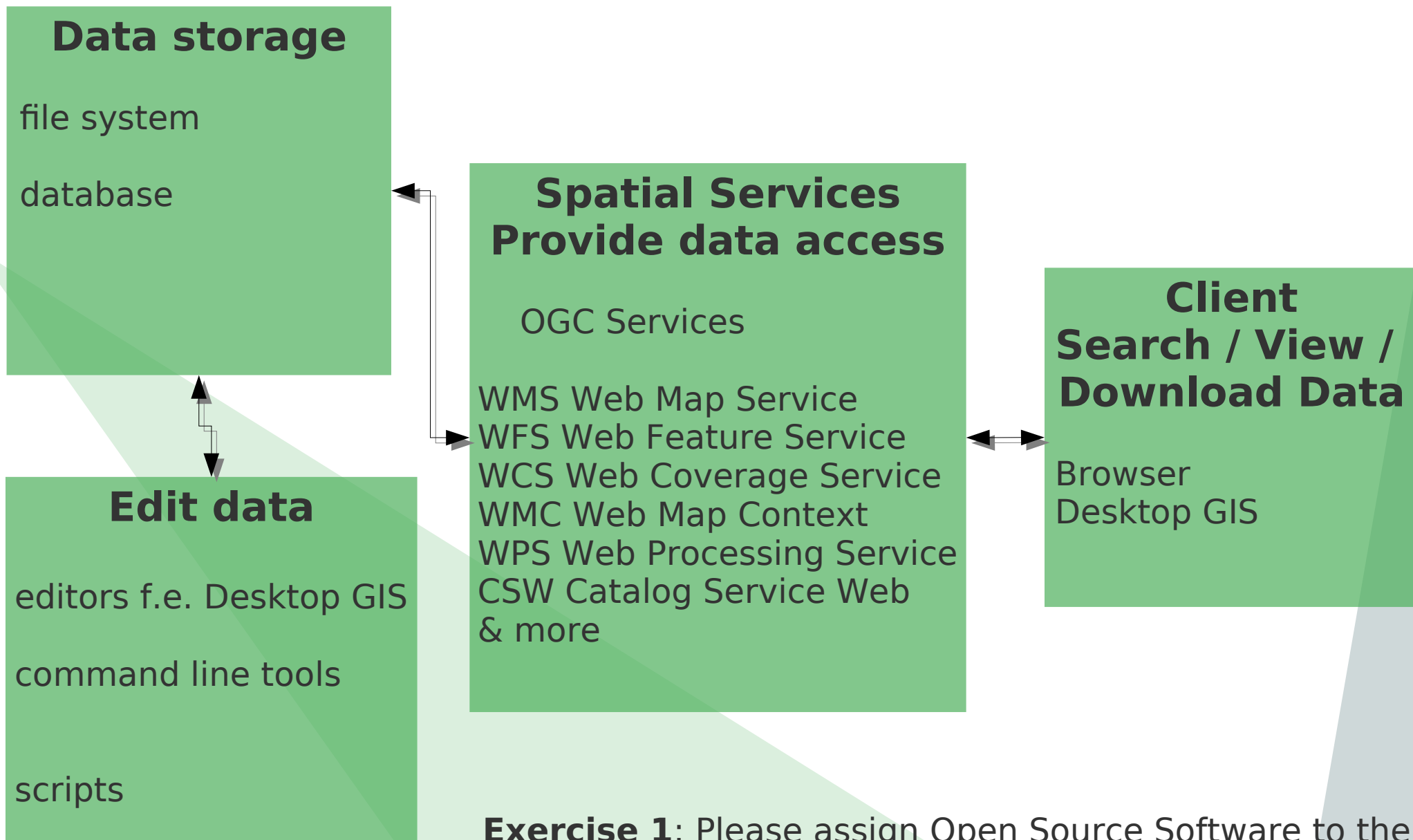
- **Increases efficiency and flexibility**

# Communication with the Clients

**Clients**

**centralized Data storage**
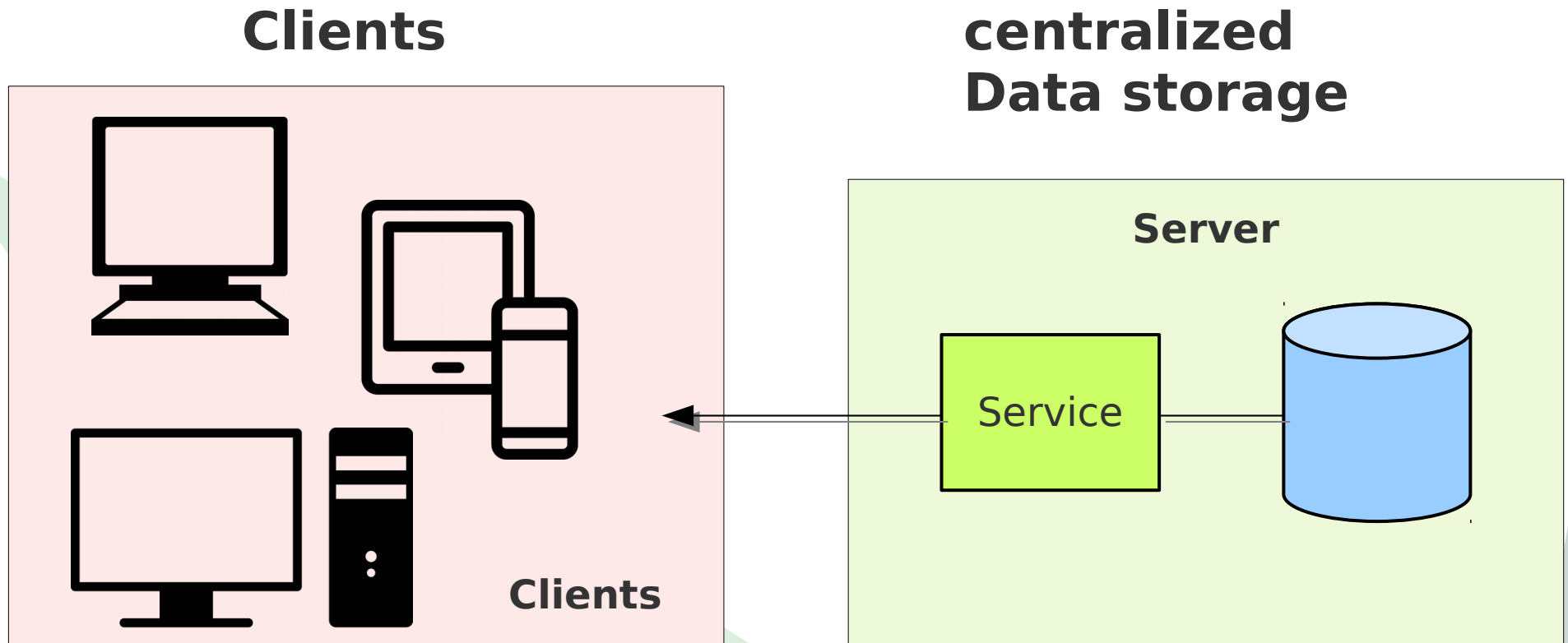


From Geodatendienste im Internet (3. Auflage, KSt. GDI-DE)
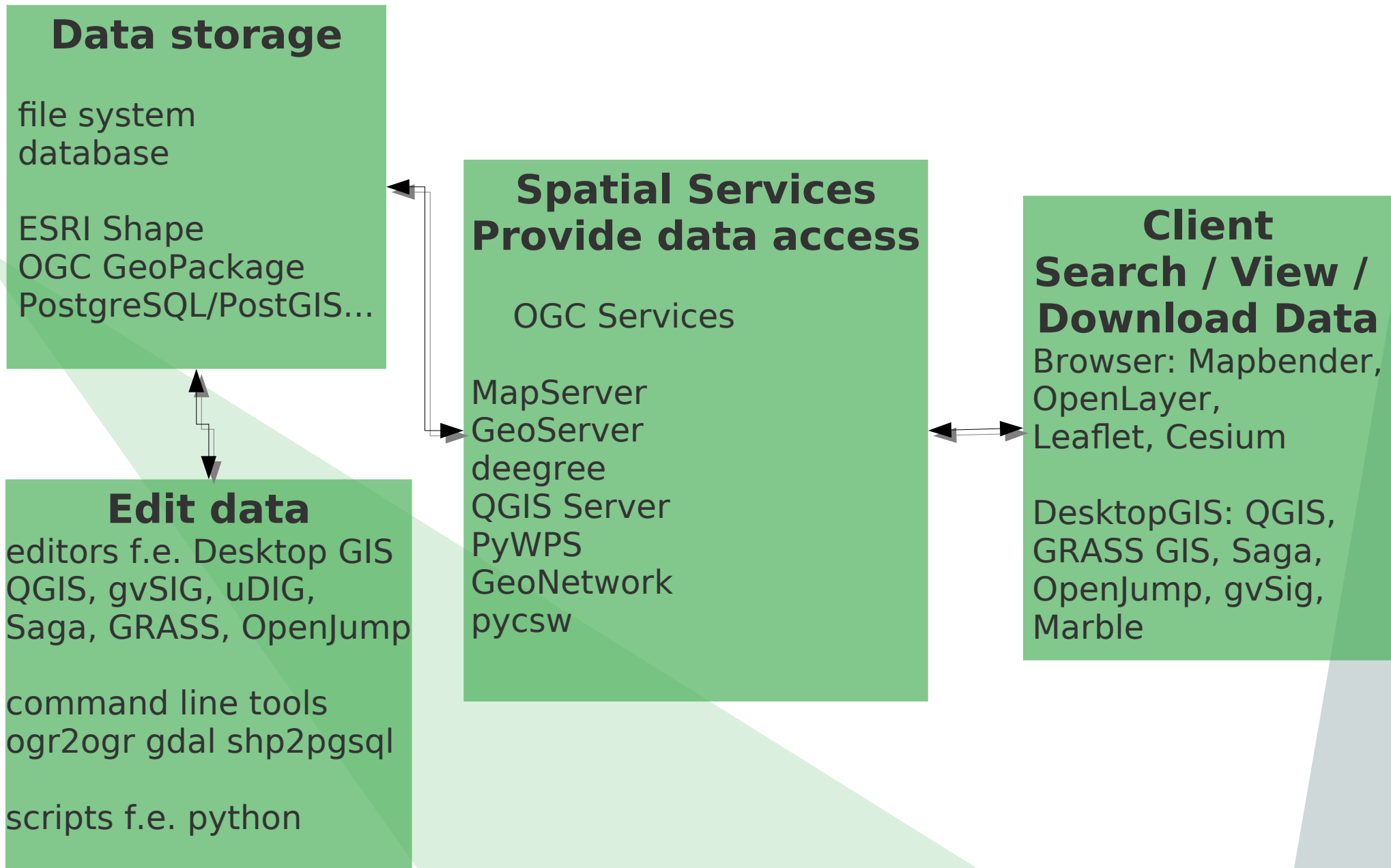
http://www.geoportal.de/SharedDocs/Downloads/DE/GDI-DE/Flyer-Broschueren/Leitfaden-Geodienste-im%20Internet.pdf?__blob=publicationFile

## Data storage

file system

database

## Spatial Services
## Provide data access

OGC Services

WMS Web Map Service
WFS Web Feature Service
WCS Web Coverage Service
WMC Web Map Context
WPS Web Processing Service
CSW Catalog Service Web
& more

## Client
## Search / View /
## Download Data

Browser
Desktop GIS

## Edit data

editors f.e. Desktop GIS

command line tools

scripts

**Exercise 1**: Please assign Open Source Software to the components of a SDI. See https://live.osgeo.org

## Data storage

file system
database

ESRI Shape
OGC GeoPackage
PostgreSQL/PostGIS...

## Spatial Services
## Provide data access

OGC Services

MapServer
GeoServer
deegree
QGIS Server
PyWPS
GeoNetwork
pycsw

## Client
## Search / View /
## Download Data

Browser: Mapbender,
OpenLayer,
Leaflet, Cesium

DesktopGIS: QGIS,
GRASS GIS, Saga,
OpenJump, gvSig,
Marble

## Edit data

editors f.e. Desktop GIS
QGIS, gvSIG, uDIG,
Saga, GRASS, OpenJump

command line tools
ogr2ogr gdal shp2pgsql

scripts f.e. python

# Data - Status

- Users work with different data

- Different formats

- Different tools to view and edit

- Data can be spread and copied

- Data can have different versions that are in use

# Data

**Goals**

- Central data storage

- Editing does not have to take place always in the central data storage, but define how to exchange

- Easy import & export

- Flexible visualisation with different tools

- Data storage with defined access

- Data storage with authentification/authorisation and multi user access

- Data history

# Data

**Goals**

- **First Goal: centralized data storage**

# Database
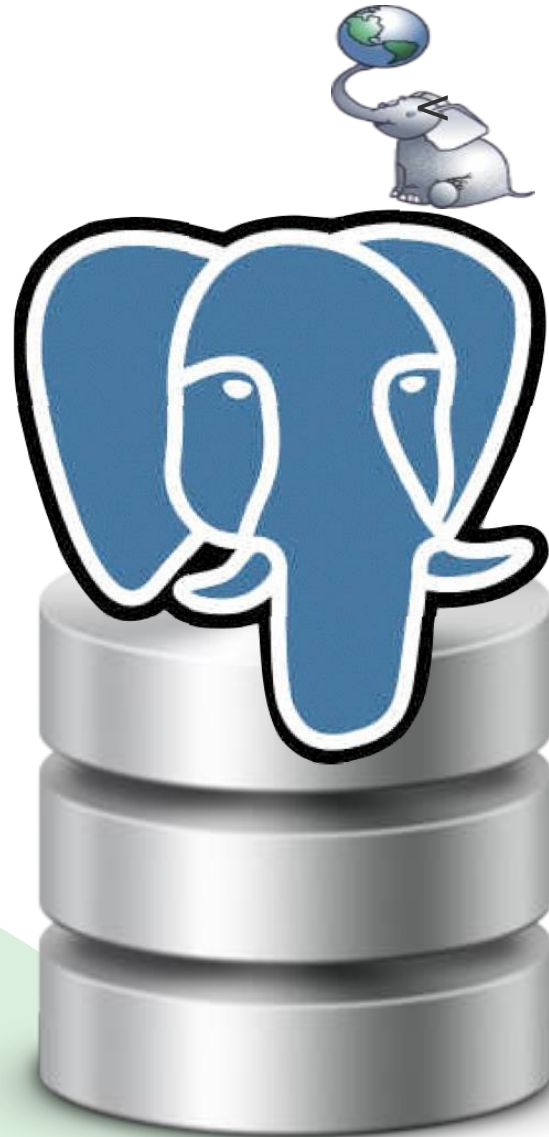
# PostgreSQL & PostGIS

# PostgreSQL/PostGIS

- **Supported by several other programs and programming languages**

- **Fast, powerful, reliable, robust, easy to maintain**

- **PostGIS is an extension for PostgreSQL**

- **Let PostGIS do the work - not your Desktop GIS**

- **Follows standard - OGC Simple Feature Specification for SQL and ISO SQL/MM Specification**

- **Provides many spatial functions**

- **Control access to your data**

# Exercise 2: Create Spatial Database in PostgreSQL

- **Open Database Client pgAdmin III**

- **Create database: context menu on database → new database → name foss4g**

- **Load postgis Extension**

  - **Context menu on database foss4g → new object → new extension → name: postgis**

# Data

**Goals**

- First Goal: centralized data storage

- **Second Goal: Easy import & export of data**

- **Third Goal: Flexible visualisation with different tools**

# Exercise 3: Load Natural Earth data (Shapes) in Desktop GIS QGIS

- **/home/user/data/natural_earth2/ ne_10m_admin_1_states_provinces_shp.shp**

- **Provinces of Romania**

- **Filter show only admin = 'Romania'**

- **Label with column [name]**

# Data import from QGIS to PostgreSQL

- **You can import Shape to PostgreSQL via**

    - **QGIS DB Manager or**

    - **shp2pgsql**

    - **ogr2ogr**

    - **python**

    - **...**

# Create a new PostgreSQL Connection in QGIS
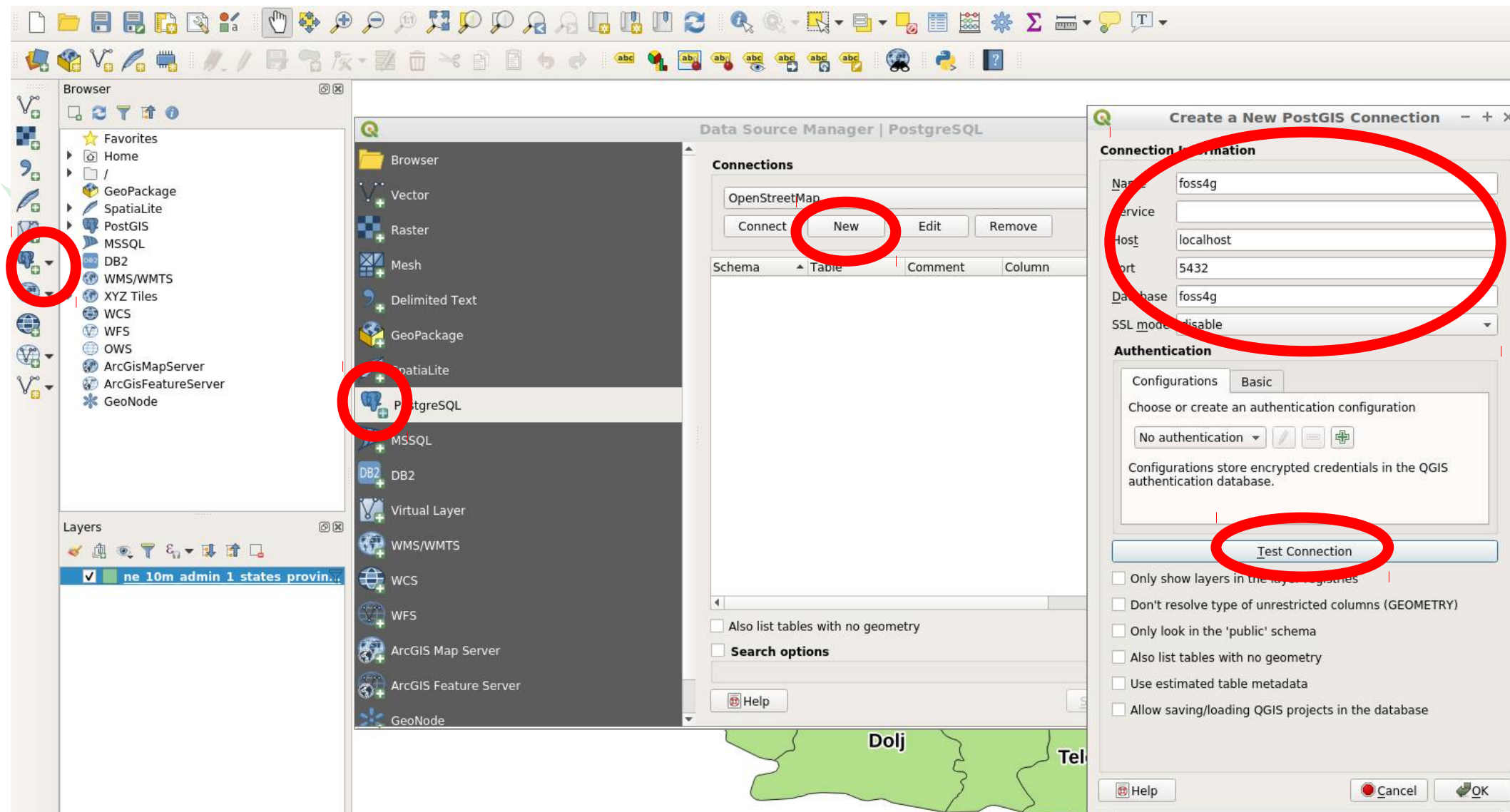
# QGIS DB Manager

- **Easy Import / Export**

- **Supports many data formats**

- **Visualize your data**

- **Show & edit table structure**

- **Create index**

- **Vacuum / Analyze**

# QGIS DB Manager Import

# Exercise 4: Import & Export

- **Import provinces from Natural Earth data (only Romania)**

- **Import populated places from Natural Earth data (only Romania)**

- **Add the table from your database to your QGIS project (drag & drop)**

- **Export: Save populated places as geopackage via DB Manager or via QGIS save as...**

# Data

**Goals**

- First Goal: centralized data storage

- Second Goal: Easy import & export of data

- Third Goal: Flexible visualisation with different tools

- **Fourth Goal: Defined access to data**

# PostgreSQL roles

- **PostgreSQL has roles**
    - **groups**
    - **User with login**
- **You can give access to databases, schema, tables, views ... to roles**
- **Write or read access**

# Create roles in PostgreSQL

- **Create role, create login role**
- **Give read access to table places**
- **Give write access to table countries**

# Exercise 5: Control data access

- **Create a role workshop_read and workshop_writer**

- **Create a login role robert with a password and add to workshop_reader**

- **Create a new login role wilma and add wilma to the workshop_writer role**

- **Grant read access to table places to your new role workshop_reader**

- **Grant write access to table countries to your new role workshop_writer**

- **Try to access and edit via QGIS**

# Exercise 5: Control data access Solution

**CREATE ROLE workshop_reader;**

**CREATE ROLE workshop_writer;**

**CREATE ROLE robert WITH LOGIN PASSWORD 'foss4g';**

**GRANT workshop_reader TO robert;**

**CREATE ROLE wilma WITH LOGIN PASSWORD 'foss4g';**

**GRANT workshop_writer TO wilma;**

# Exercise 5: Control data access Solution

GRANT SELECT ON places to workshop_reader;

-- change to user robert

Select * from places;


GRANT ALL ON countries to workshop_writer;

GRANT USAGE ON SEQUENCE countries_gid_seq TO workshop_writer;


-- change to user wilma

Select * from countries;

Update countries set name = 'TEST' WHERE name = 'Romania';

# Data storage

file system

database

# Spatial Services
# Provide data access

OGC Services

WMS Web Map Service
WFS Web Feature Service
WCS Web Coverage Service
WMC Web Map Context
WPS Web Processing Service
CSW Catalog Service Web
& more

# Client
# Search / View /
# Download Data

Browser
Desktop GIS

# Edit data

editors f.e. Desktop GIS

command line tools

scripts
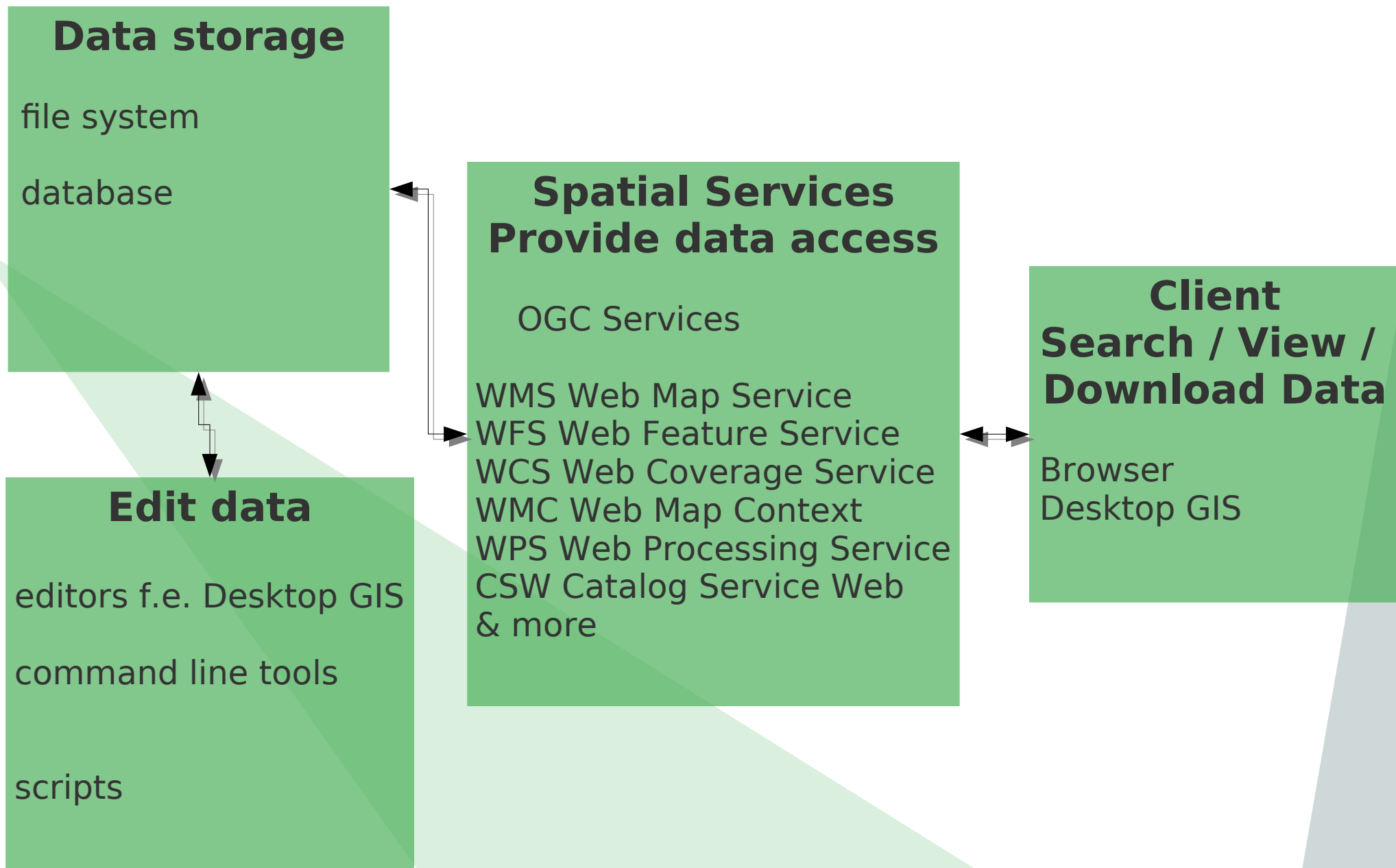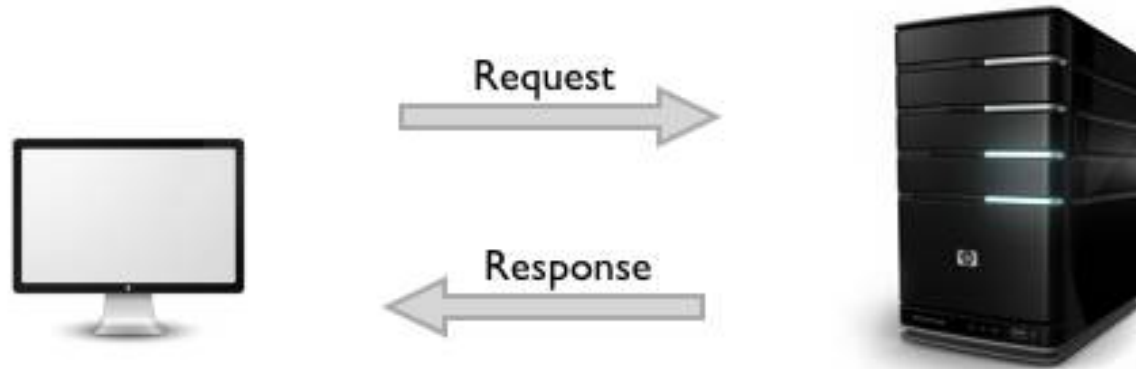
# Services

# Provide Data via OGC Services

**Goals**

- **Provide Data in the web - Intra- or Internet**

- **Provide data via standards that many tools support**

- **Provide INSPIRE conform Services in Europe**

- **OGC WMS Web Map Service – show data and get information (advantage: styling is already defined)**

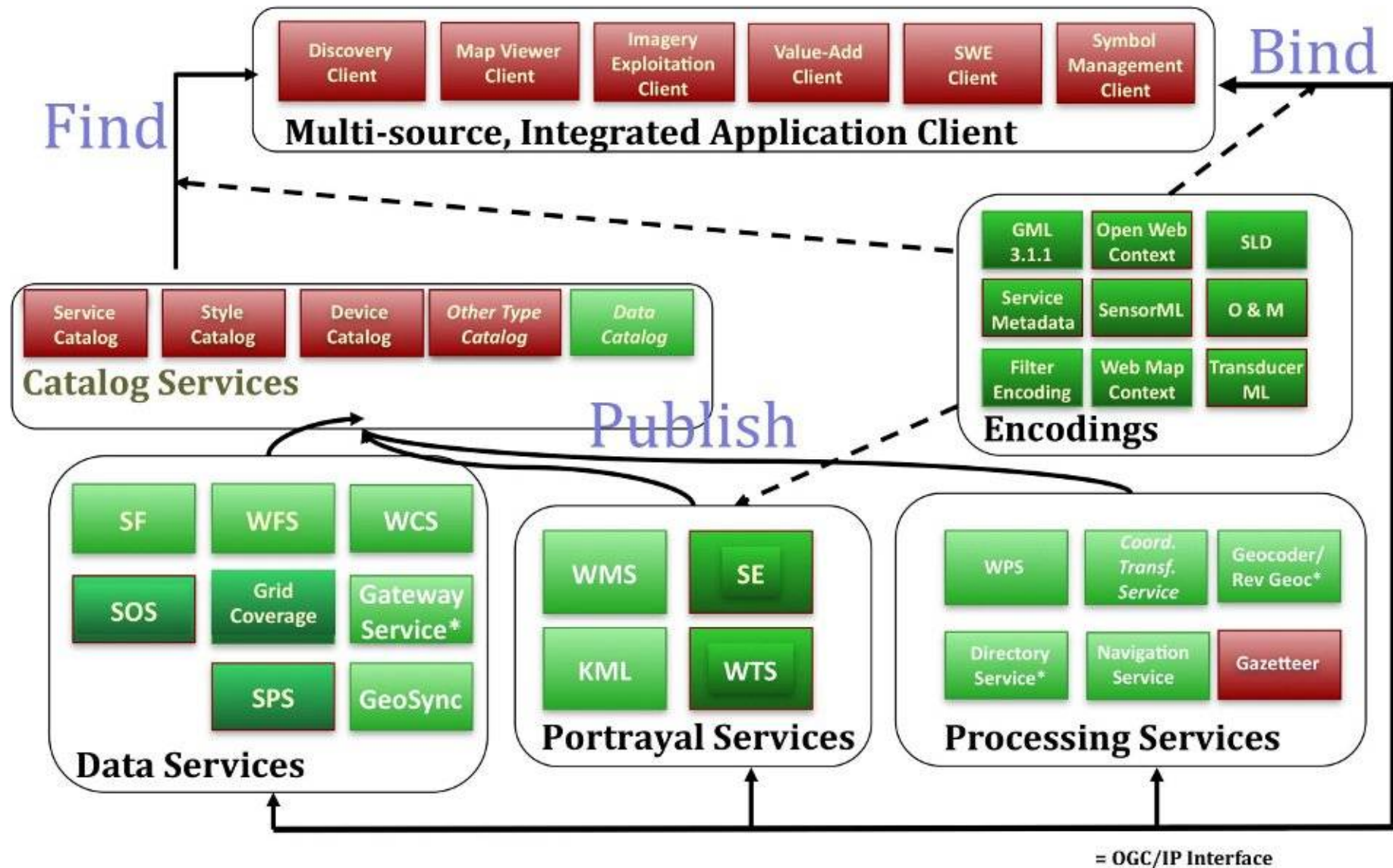- **OGC WFS Web Feature Service  - download service, provide data, edit data**

# Web Services Framework Of OGC Geoprocessing Standards



**Find**

**Bind**

**Publish**

**Multi-source, Integrated Application Client**
- Discovery Client
- Map Viewer Client
- Imagery Exploitation Client
- Value-Add Client
- SWE Client
- Symbol Management Client

**Catalog Services**
- Service Catalog
- Style Catalog
- Device Catalog
- Other Type Catalog
- Data Catalog

**Encodings**
- GML 3.1.1
- Open Web Context
- SLD
- Service Metadata
- SensorML
- O & M
- Filter Encoding
- Web Map Context
- Transducer ML

**Data Services**
- SF
- WFS
- WCS
- SOS
- Grid Coverage
- Gateway Service*
- SPS
- GeoSync

**Portrayal Services**
- WMS
- SE
- KML
- WTS

**Processing Services**
- WPS
- Coord. Transf. Service
- Geocoder/ Rev Geoc*
- Directory Service*
- Navigation Service
- Gazetteer

= OGC/IP Interface

3

**https://live.osgeo.org/archive/10.5/en/standards/standards.html**

# OSGeo Software with OGC WMS Support

- **MapServer**
- **GeoServer**
- **QGIS Server**
- **deegree**
- **MapProxy**



**OGC WMS – map service, provide maps as raster, information as html, plain text, GML**

# OSGeo Software with OGC WFS Support

- **MapServer**
- **GeoServer**
- **QGIS Server**
- **deegree**
- **MapProxy**

**OGC WFS – feature service, data access f.e. via GML**

# Provide Data via OGC Services

**Goals**

- **First Goal: Provide Data in the web - Intra- or Internet**

- **Provide data via standards that many tools support**

- **OGC WMS Web Map Service – show data and get information (styling is already defined)**

# WMS example with QGIS Server

**Publish your data via QGIS Server as WMS**

**Menu → Project → Project properties → OWS Server**

- **Name and title, extent, layer access**

- **(make sure you saved the password for your database access)**

- **http://localhost/cgi-bin/qgis_mapserv.fcgi? map=/home/user/ service_wms.qgz&SERVICE=WMS&VERSION= 1.3.0&REQUEST=GetCapabilities**

# WMS Capabilities Document

# Exercise 6: WMS example with QGIS Server

- **Publish your data via QGIS Server as WMS**

- **Create a WMS with two layers countries and places – style them nicely with labeling**

- **Save your project at: /home/user/service_wms.qgs**

- **Load your WMS in an empty QGIS project**

# Exercise 7: Load external Services to QGIS

- **http://localhost/cgi-bin/qgis_mapserv.fcgi?map=/usr/local/share/qgis/QGIS-NaturalEarth-Example.qgz&SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities**

- **http://osm-demo.wheregroup.com/service?REQUEST=GetCapabilities&Service=WMS&Version=1.3.0**

- **Romania http://geo-spatial.org/geoserver/ows?service=wms&version=1.3.0&REQUEST=getCapabilities**

- **Find a WMS Server for you needs**

    **https://inspire-geoportal.ec.europa.eu/**

# How to spread your service in the web?

- **Refer to your GetCapabilities-Urls**

- **Add your GetCapabilities & Metadata in a Metadata catalog**

- **Offer your WMS in WebGIS Client in a ready to use application f.e. OpenLayers, Leaflet, Mapbender, GeoMoose, MapStore, QGIS Map Client**

# Provide Services with Leaflet

- **Open the leaflet demo**

- **Add things to it**

    - **Bounding box**

    - **WMS SERVICE**

# Exercise 8: Leaflet

- **Copy paste leaflet files from /var/www → desktop**

- **Fix libraries URLs**

- **Add code before </script>**

```
// define rectangle geographical bounds

var bounds = [[43.7,20.0], [48.3,29.7]];

// create an orange rectangle
L.rectangle(bounds, {color: "#ff7800",
weight: 1}).addTo(map);

// zoom the map to the rectangle bounds

map.fitBounds(bounds);
```
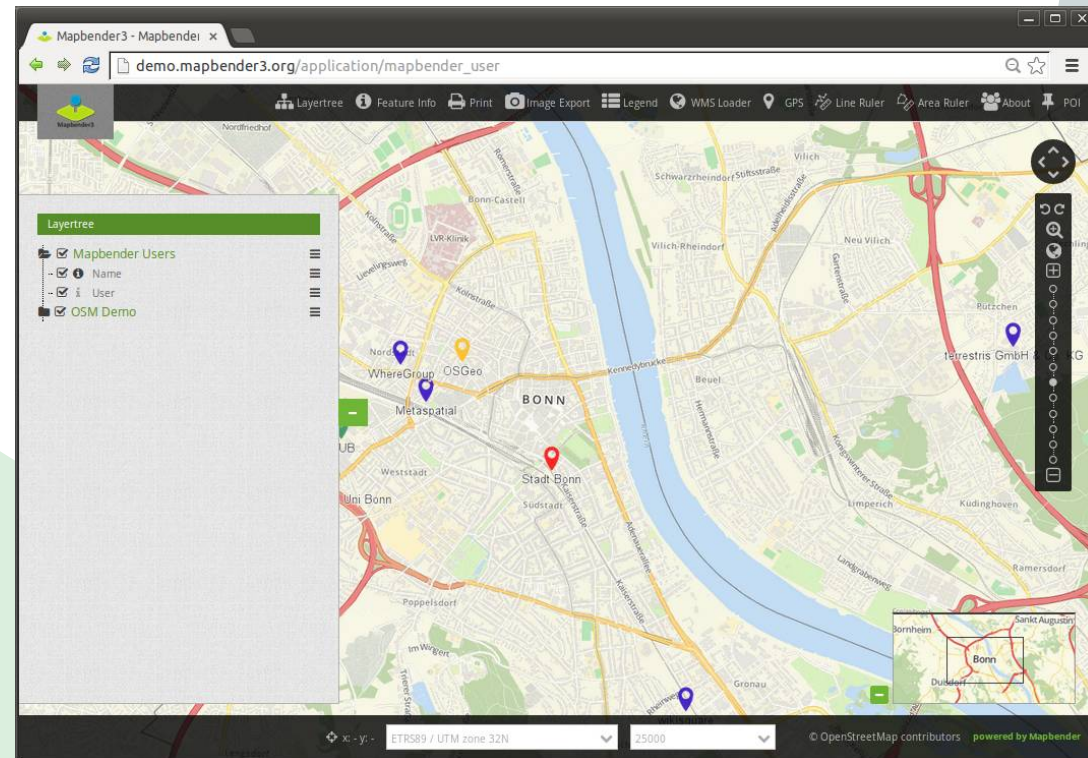
# Exercise 9: Leaflet

- **Add WMS service**

```
var countriesAndBoundaries =
L.tileLayer.wms('http://geo-spatial.org/geo
server/ows?', {

layers: 'omc:ro_admin_gr_colegii_sen'

}).addTo(map);
```



FOSS4G is here this year!

Leaflet

# Provide Services in a Geoportal f.e. Mapbender

- **http://localhost/mapbender/**

- **Login via: root / root**

- **Provide Applications for different needs**

# Publish WMS in Mapbender

- **Menu → New DataSource**
- **Load GetCapabilities Url**

# Exercise 9: Load WMS in Mapbender

- **Load your QGIS WMS and some other WMS in Mapbender**

# Exercise 10: Create a new Application in Mapbender

- **Your application should start with the extent of Bucharest**

- **Copy Application mapbender_user and rename it to FOSS4G**

- **Switch to Layout → map-Element and set SRS EPSG:4326**

- **Modify MAX EXTENT to Romania lower left 20.1139 43.5602 – upper right 29.8267 48.3647**

- **Modify Start Extent Bucharest lower left 26.0985 44.4329 upper right 26.1064 44.4399**

# Exercise 11: Add Services to your application

- **Your application should show your QGIS WMS and some other external WMS**

- **Go to Tab Layerset**

- **Add WMS via +**

# Metadata Catalog to find your data

**Goals**

- **Your user should find your data**
- **Provide a catalog for you data**

# Provide Metadata for your data and Services

- **Metadata should be provides & has to be up-to-date**

- **OGC Catalogue Service Web (CSW)**

- **GeoNetwork, GeoNode, pycsw, Metador**

- **f.e Germany: GDI-DE & 16 Catalogues for provinces**

- **f.e. Greek http://geodata.gov.gr/**

- **INSPIRE**

- **https://inspire.ec.europa.eu/INSPIRE-in-your-Country/DE**

- **https://inspire.ec.europa.eu/INSPIRE-in-your-Country/RO**

# Exercise 11: Search in the Romanian Metadata Catalog for a Service

- **https://inspire.ec.europa.eu/INSPIRE-in-your-Country/RO**

# Exercise 11: Add your new QGIS WMS to GeoNode or GeoNetwork

- **Open GeoNode**

- **Login: admin/admin**

- **Add a new Metadata entry for you WMS**

- **Follow the quickstart**

- **https://live.osgeo.org/en/quickstart/geonode_quickstart.html**

# Wrap things up

- **Central data storage in a database offers lot of advantages**

- **Data Sharing via Services is easy and supported by many programs**

- **Offering ready-to-use applications covers the needs of many not advanced users**

- **Metadata helps you to find the data you need**

- **Metadata should be up-to-date**

- **A SDI is a profit for all involved parties**