YYT-C3001 Management of environmental data and information

Lecture 6: Technical Standards for Spatial Data

Aalto University
School of Engineering

Jussi Nikander

24.9.2020

Contents of this lecture

OGC data interfaces (WFS, WMS, WMTS, etc)

Textual representation formats for vector data (GML, KML, GeoJSON)

Raster data representation formats (TIFF, PNG)

Learning goals for this lecture

Understand what is the difference between standards for data transfer and standards for data representation

Know the basics of OGC data interface standards

Know the basics of GML, KML, GeoJSON and understand their differences

Know how raster data can be stored

Web Spatial Data Transfer Standards

Aalto University
School of Engineering

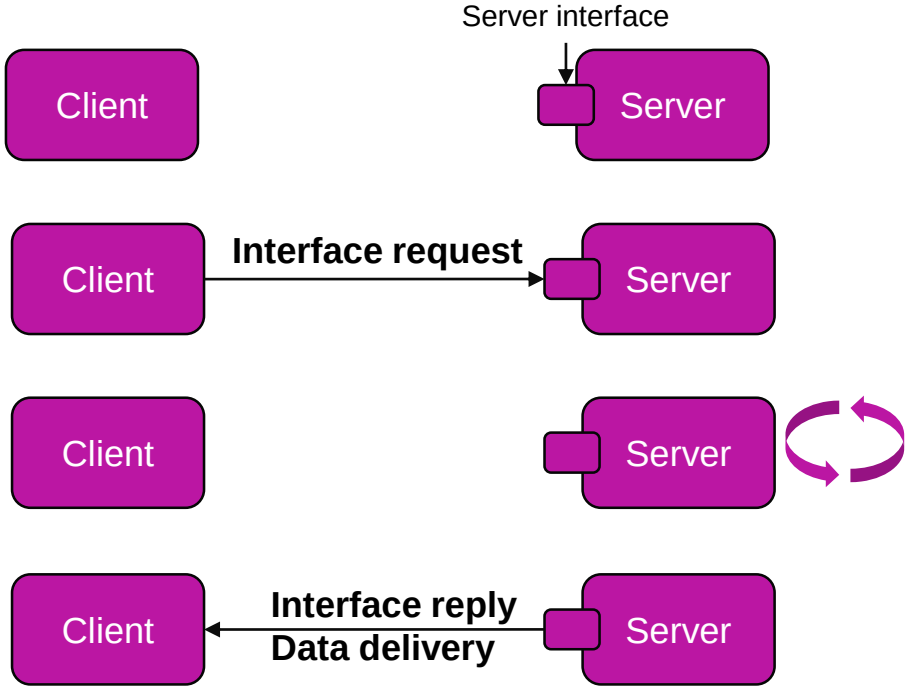
Spatial data transfer over the web

Web server has a public interface, ready to serve client data requests. The client needs to know the address of this interface to access it

Client contacts the interface to request data.

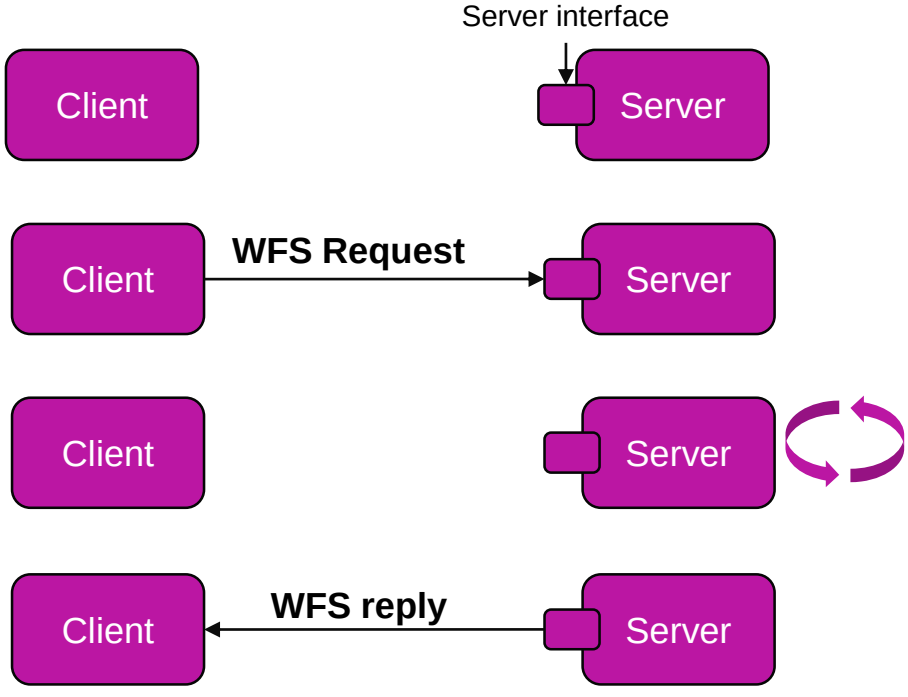
Server processes the request

Server delivers the requested data to the client. The client may continue to request new data upon demand.



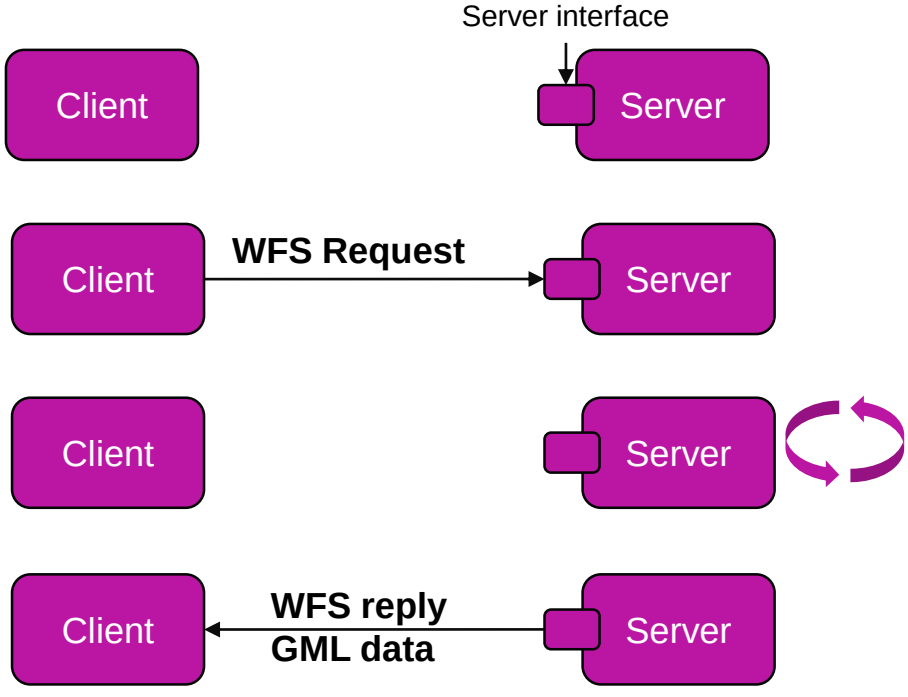
Data interface

- **A location that clients can use to request data**
  - Typically follows a widely-known standard that client software know how to use
- **OGC Web Interfaces are widely used**
  - Web Feature Service (WFS)
  - Web Map Service (WMS)
  - Web Map Tile Service (WMTS)
  - Etc.



Data delivery

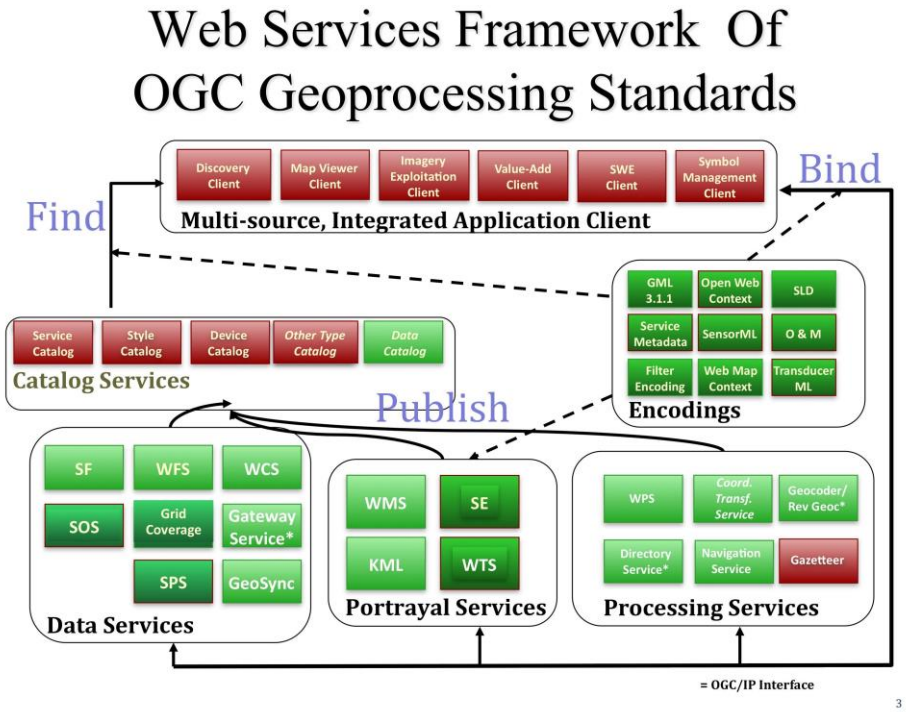
- **The data provided by an interface also follows a widely-known standard**
  - This allows the client to interpret and use the delivered data
- **OGC GML, Google KML, GeoJSON**
- **More domain-specific languages such as CityGML**



OGC Data Interfaces

The Open Geospatial Consortium (OGC)

- **Standards organization for geospatial data**
- **Develops and maintains many standards related to the use of geospatial data**
- **Official standardization done in cooperation with standardization organizations**
  - E.g. ISO



Aalto University
School of Engineering

Image: https://live-archive.osgeo.org/4.0/standards/standards.html

OGC Web Interfaces

- **An important set of OGC standards are the OGC spatial data web interfaces**
  - Web Feature Service
  - Web Map (Tile) Service
  - Web Coverage Service
  - Web Processing service

- **Common to all standards**
  - Works over HTTP
  - Each interface contains a getCapabilities –function that can be used to find out what the service contains

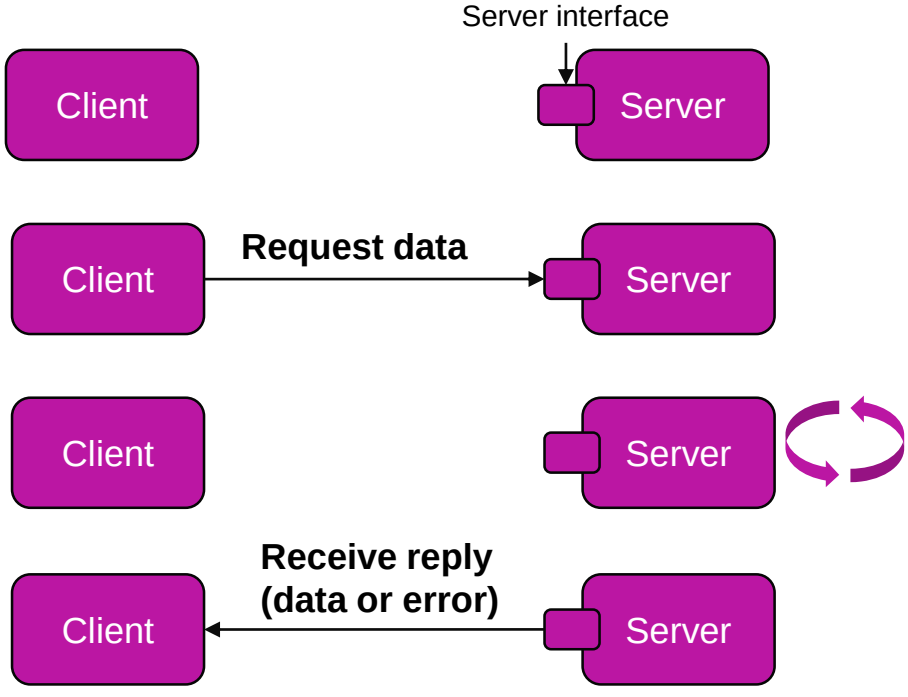
Open datasets

Standard	Connection URL
WMS	http://avaa.tdata.fi/geoserver/paituli/gwc/service/wms?
WMTS	http://avaa.tdata.fi/geoserver/paituli/gwc/service/wmts?
WFS	http://avaa.tdata.fi/geoserver/paituli/wfs?
WCS	http://avaa.tdata.fi/geoserver/paituli/wcs?

http://avaa.tdata.fi/geoserver/paituli/gwc/service/wms?request=getCapabilities gives a machine-readable list of the Paituli wms service data layers

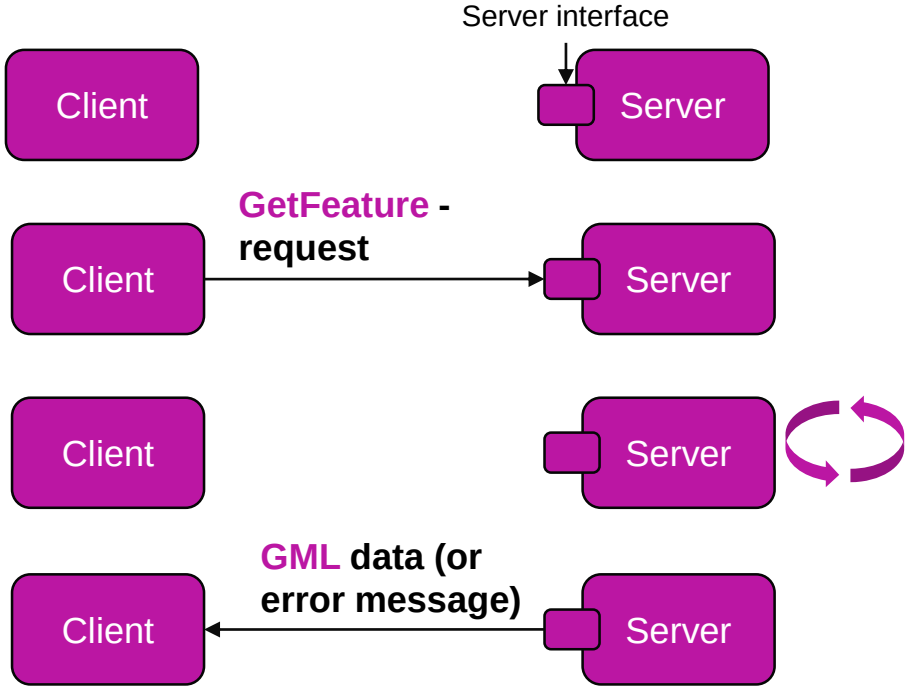
OGC Web Interfaces

- **The goal of all interfaces is similar: to provide a service where a user can request data and receive it in a well-defined format**
- **Some services also provide capability of processing (analyzing) the data**



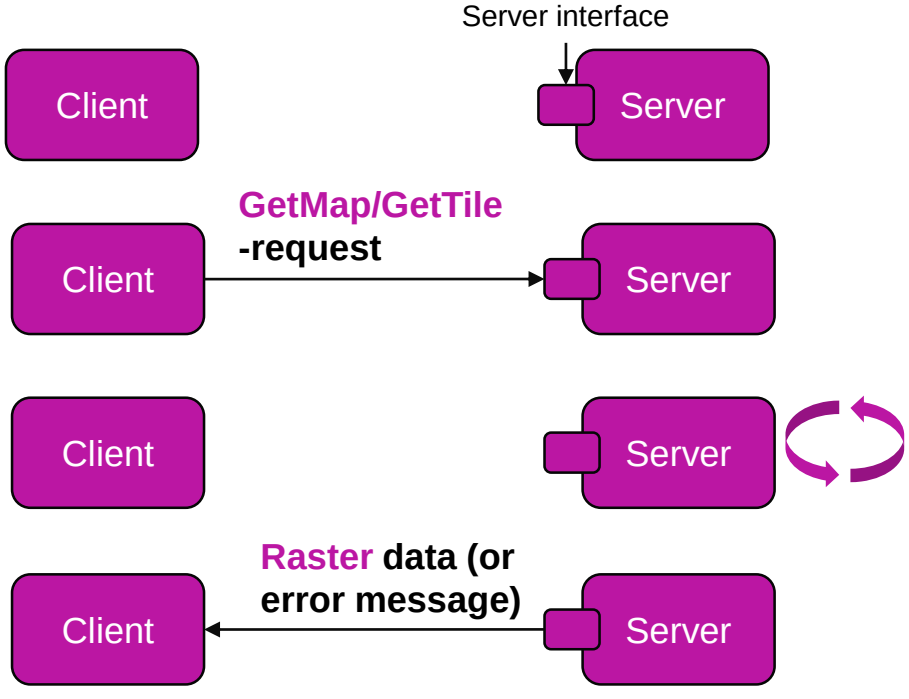
Web Feature Service

- **Web Feature Service (WFS) is an interface for requesting distinct features**
  - That is, vector data
- **The standard also contains functionality for data management**
  - Create, delete, update
  - Obviously, these are typically not enabled in open data services
- **Feature requests are are stateless, management requires transactions**



Web Map Service and Web Map Tile Service

- **Web Map Service (WMS) is an interface for requesting maps**
  - That is, raster data
- **Web Map Tile Service (WMTS) is an interface for requesting map tiles**
  - Tiles are small, regular-sized map pieces (typically rectangles)



Web Coverage Service, Web Processing Service, and others

- **Web Coverage Service (WCS) is an interface that handles coverages that contain spatio-temporal data**
  - Handled as special type of feature, where data values change according to location
  - Can also contain complex data semantics
- **Web Coverage Processing Service (WCPS) defines a programming language used to manipulate coverages from client-side**
- **Web Processing Service (WPS) is an interface for requesting spatial processing (analysis) over the web**
  - Allows to use remote processing for manipulation of GIS data
- **Standardization makes it easier to implement processing services and using them**
  - Everyone shares the same language

Other interfaces (examples)

- **OGC standards are widely used because**
  - They have significant support in software
  - Are open and free to use for anyone
- **ESRI has its own ArcGIS-based web interfaces**
  - They also have support for OGC, but recommend people use Arc-specific things
- **Google has their own Map API for Google Maps**
- **In Earth Sciences OPeNDAP is often used**
  - Meteorological data, other earth science data
  - Related to the NetCDF raster data format

Classroom exercise: data delivered

I'll divide you into breakout rooms

Each breakout room has a different OGC web interface

Your task is to try and find out in what data format(s) the interface supports

Let's use 15 minutes for this

https://jamboard.google.com/d/1n4uf26PD5wblb_XPywyYbpEPkUp9xMtFRS4Fiqxaq7w/edit?usp=sharing

Spatial data representation standards: text-based vector representation

Aalto University
School of Engineering

Vector data languages

- **Vector data delivery from service to client requires a method for representing the data transferred**
  - The method should be such that it can be used to easily create arbitrary sets of data elements
  - The method should also follow well-established data exchange standards
  - The method should allow the recipient to start interpreting the data immediately when they start receiving it

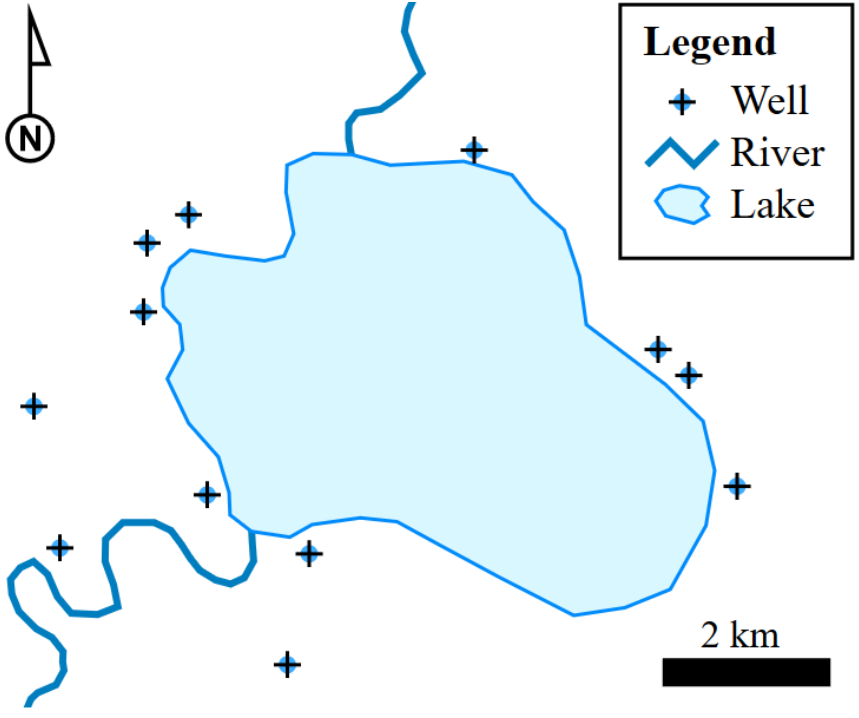


Image: Wikipedia

Vector data languages

- **Text-based data delivery is very common over the internet**
  - eXtensible Markup Language (XML)
  - JavaScript Object Notation (JSON)
  - Etc.
- **Vector data delivery over internet typically follows these approaches**
  - GML, GeoJSON, etc.

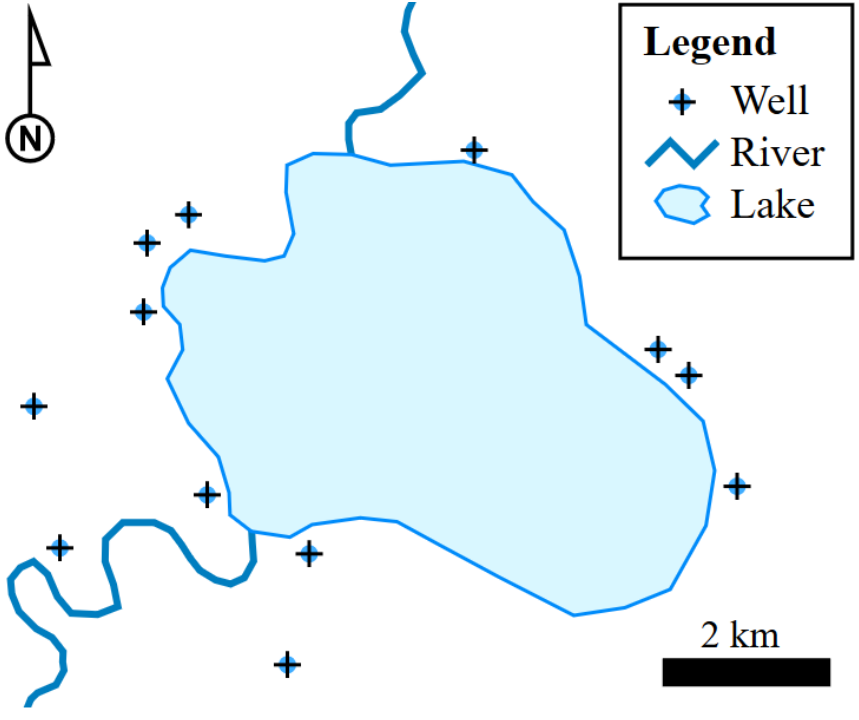
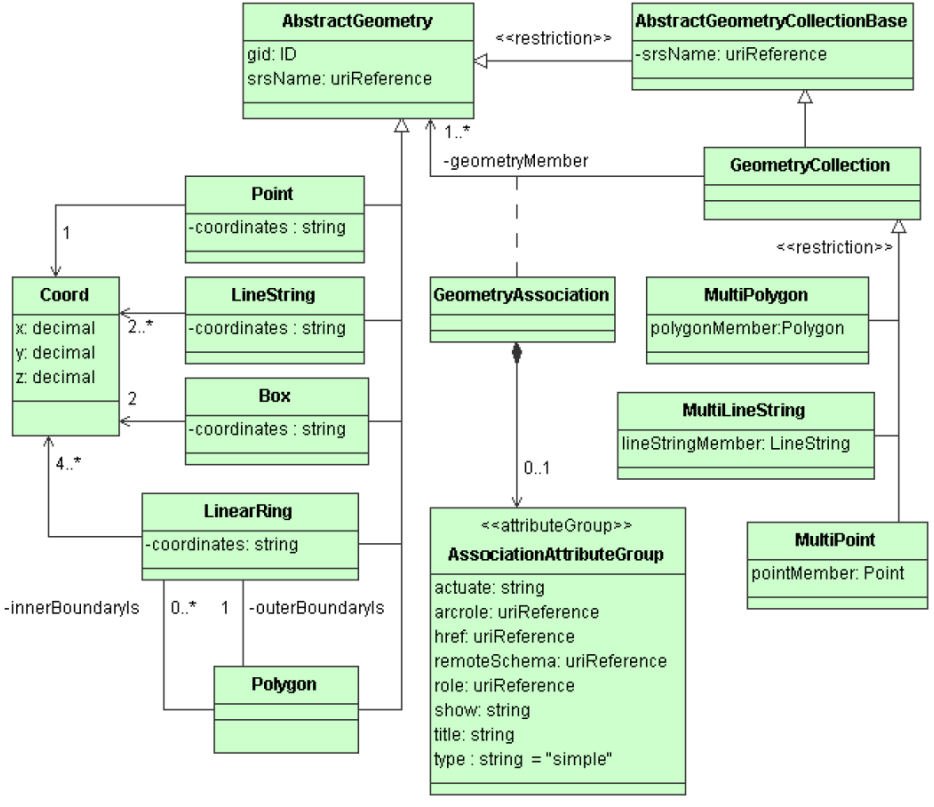


Image: Wikipedia

Geography Markup Language

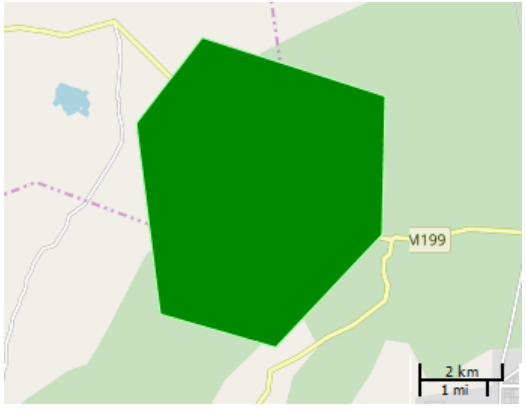
- **XML grammar for representing geographical features, defined by OGC**
- **Supports rich, hierarchical set of different data types**
- **A GML dataset contains**
  - Metadata
  - Spatial reference
  - A set of features / coverages defined using e.g. Simple Features definitions



GML Geometry schema in UML notation (ver. 2.1.2)

GML example

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:FeatureCollection xmlns:wfs="http://www.opengis.net/wfs" xmlns:ows="http://www.opengis.net/ows" xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
 xmlns:geonode="http://worldmap.harvard.edu/" xsi:schemaLocation="http://worldmap.harvard.edu/ http://worldmap.harvard.edu/geoserver/wfs?\
 service=WFS&version=1.0.0&request=DescribeFeatureType&typeName=geonode%3Amypolygon_px6 http://www.opengis.net/wfs
 http://worldmap.harvard.edu/geoserver/schemas/wfs/1.1.0/wfs.xsd" timeStamp="2018-10-23T10:25:49.082Z" numberOfFeatures="0">
 <gml:boundedBy>
  <gml:Envelope srsName="http://www.opengis.net/gml/srs/epsg.xml#4326" srsDimension="2">
   <gml:lowerCorner>79.3906402477365 11.627857397681</gml:lowerCorner>
   <gml:upperCorner>79.4476318248771 11.6971217989284</gml:upperCorner>
  </gml:Envelope>
 </gml:boundedBy>
 <gml:featureMembers>
  <geonode:mypolygon_px6 gml:id="mypolygon_px6.1">
  <gml:boundedBy>
   <gml:Envelope srsName="http://www.opengis.net/gml/srs/epsg.xml#4326" srsDimension="2">
    <gml:lowerCorner>79.39064024773653 11.627857397680973</gml:lowerCorner>
    <gml:upperCorner>79.44763182487713 11.697121798928404</gml:upperCorner>
   </gml:Envelope>
  </gml:boundedBy>
  <geonode:the_geom>
   <gml:Polygon srsName="http://www.opengis.net/gml/srs/epsg.xml#4326" srsDimension="2">
    <gml:exterior>
     <gml:LinearRing srsDimension="2">
      <gml:posList>79.39064024773653 11.677958136567844 79.40574644890594 11.697121798928404 79.44763182487713 11.684009962648828 79.44694517936921 11.65307701972681
                   79.42256926384478 11.627857397680973 79.39613341179829 11.635255390466018 79.39064024773653 11.677958136567844</gml:posList>
     </gml:LinearRing>
    </gml:exterior>
   </gml:Polygon>
  </geonode:the_geom>
  <geonode:Name>Boundary1</geonode:Name>
  <geonode:Description>This is sample boundary created for testing purpose<BR></geonode:Description>
  <geonode:Start_Date>2012-04-15T18:30:00Z</geonode:Start_Date>
  <geonode:End_Date>2012-04-15T18:30:00Z</geonode:End_Date>
  <geonode:String_Value_1>Surya</geonode:String_Value_1>
  </geonode:mypolygon_px6>
 </gml:featureMembers>
</wfs:FeatureCollection>
```



Source: http://worldmap.harvard.edu/data/geonode:mypolygon_px6

GML Application Schemas

- **There are plenty of more specialized spatial data description languages based on GML**
  - These are made with GML application schemas, which describe the domain of a GML document
  - GML itself is defined using an XML schema, which describes the structure of an XML document

- **CityGML is a language for describing 3D models of cities and landscapes**
- **INSPIRE defines a number of GML application schemas**
- **SensorML is a language for describing sensors and measurements**
- **InfraGML describes civil engineering infrastructure**
- **Etc.**

Keyhole Markup Language

- **A language for representing geographic data on Earth browsers**
- **Used on Google Earth and provided significant support by Google**
- **KML is focused on visualization, GML on spatial element properties**

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Simple placemark</name>
    <description>Attached to the ground. Intelligently places itself
       at the height of the underlying terrain.</description>
    <Point>
      <coordinates>-122.0822035425683,
                37.42228990140251,0</coordinates>
    </Point>
  </Placemark>
</kml>
```

Note: No SRS, bounding box, nor other such things. Just what is needed to visualize and annotate graphics.

GeoJSON

- **A way to transfer OGC Simple Features data based on the JavaScript Object Notation**
- **SRS is WGS84, and representation uses decimals**
- **Data is in key:value -pairs**
- **To compare, GML is very expressive, but GML data tends to be complex and long**
  - GML standards document is 300+ pages
  - GeoJSON specification is 28

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

- GeoJSON is technically much simpler to interpret than GML
- GML can be combined with other XML-based languages

Spatial representation vs visualization classroom exercise?

- **Now, remember that GML is a language for representing spatial data, and KML is a language for visualizing spatial features**
- **Answer in Presemo**
  - Which standard would include the following (GML/KML/Both/Neither)

1. **Line feature width**
2. **TM35FIN data**
3. **Polygon**
4. **Place name**
5. **Property: soil type**
6. **Feature color**

You have 10 minutes (we'll continue at 15.35)

https://presemo.aalto.fi/medi2

Spatial representation vs visualization classroom exercise?

- **Now, remember that GML is a language for representing spatial data, and KML is a language for visualizing spatial features**
- **Answer in Presemo**
  - Which standard would include the following (GML/KML/Both/Neither)

1. **Line feature width KML**
2. **TM35FIN data GML**
3. **Polygon Both**
4. **Place name Both**
5. **Property: soil type GML**
6. **Feature color KML**

Spatial data representation standards: vectors in files and databases

Aalto University
School of Engineering

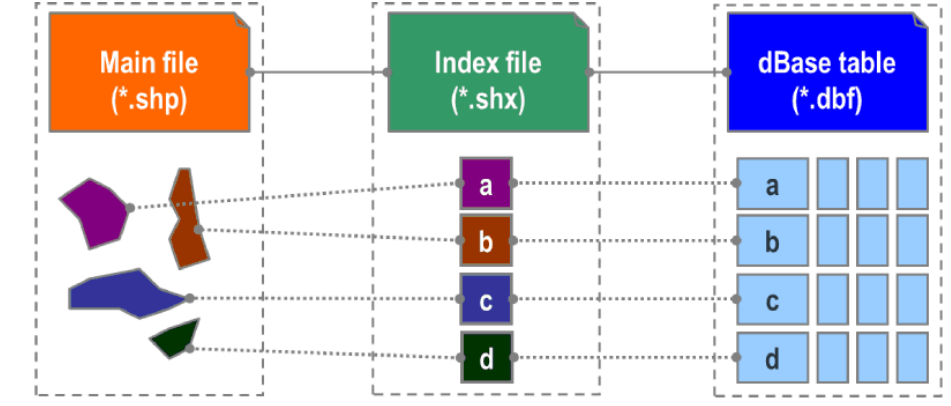
Vector data languages and files

- **GML, KML, and GeoJSON are designed for data transfer and human readability**
- **Human readability means the data requires a lot of storage space**
- **Design for data transfer means the data is cumbersome to modify after creation**
- **Other data storage formats are more useful, if you’re storing the data locally**

- **For local storage there are two basic approaches**
- **Files**
  - Should be accessed by one user at a time, typically read completely to memory when used in a GIS software
- **Databases**
  - More complex, support concurrent access, typically parts can be read to memory

Shapefile

- **For a long time was the de-facto standard for spatial data download services**
- **Developed and maintained by ESRI**
- **One shapefile dataset actually consists of several files**
- **Stores vector data as points, lines, and polygons**
  - Only one type per file
- **File size restricted to 2GB**



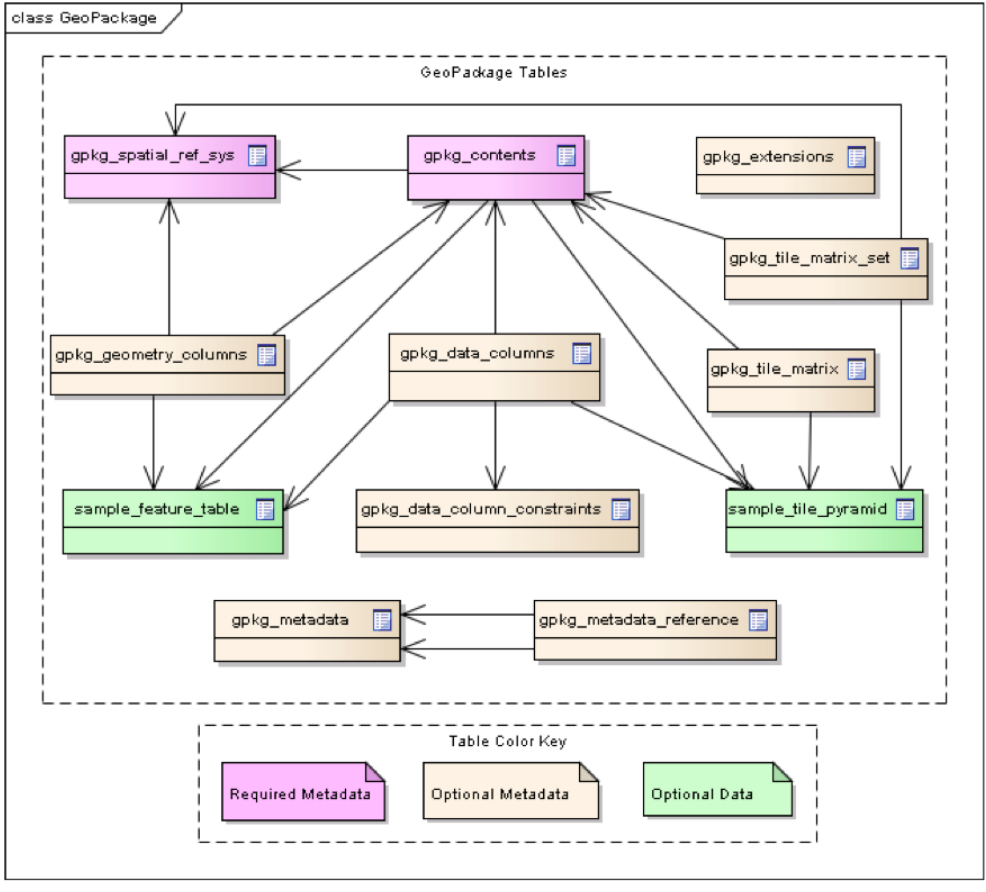
geom	id	shp_len	type	surface	width	lanes	name
	101	4507.4	2	asphalt	85.3	4	I95
	102	3491.1	1	concrete	45.1	2	Route 4
	103	2321.8	3	asphalt	75.9	4	Pinewood
	104	682.9	5	gravel	35.2	2	Ridge
	105	1279.1	4	asphalt	60.3	4	Main
	...	...	...	...	...	...	...

Predefined fields | custom fields

Image: https://transportgeography.org/?page_id=7636

GeoPackage

- **An open standard for geospatial data storage based on the SQLite database**
- **Increasingly used in spatial data download services**
  - Not tied to any specific commercial actor
- **Can also hold raster data**
- **Fewer limitations and often more responsive than Shapefile**

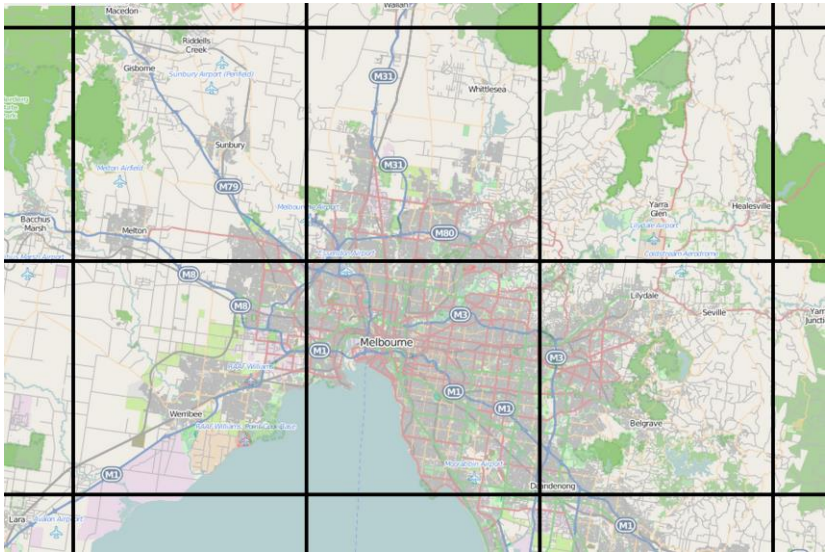


Spatial data representation standards: raster data

Aalto University
School of Engineering

Raster data

- **Raster data storage is typically based on image formats**
  - The format needs to be lossless (compression does not remove pixel data)
- **Most common image formats for spatial data are GeoTIFF and PNG**
- **Tiling is often important**



- **Tiled image is divided into smaller images that are handled separately**
  - Saves resources compared to one large image
  - Individual image locations provided by offset from origin

GeoTIFF

- **Metadata standard for the TIFF image format**
- **Attaches spatial metadata to a TIFF image, making it possible for GIS software to interpret the image as map**
- **Can be in the image file, or as a separate world file**
- **Unlimited number of data bands (raster layers)**

Aalto University
School of Engineering

```
Driver: GTiff/GeoTIFF
Files: clc2012_fi20m.tif
       clc2012_fi20m.tif.ovr
       clc2012_fi20m.tif.aux.xml
Size is 35989, 61978
Coordinate System is:
PROJCS["EUREF_FIN_TM35FIN",
    GEOGCS["GCS_EUREF_FIN",
        DATUM["European_Terrestrial_Reference_System_1989",
            SPHEROID["GRS_1980",6378137,298.257222101,
                AUTHORITY["EPSG","7019"]],
            AUTHORITY["EPSG","6258"]],
        PRIMEM["Greenwich",0],
        UNIT["degree",0.0174532925199433]],
    PROJECTION["Transverse_Mercator"],
    PARAMETER["latitude_of_origin",0],
    PARAMETER["central_meridian",27],
    PARAMETER["scale_factor",0.9996],
    PARAMETER["false_easting",500000],
    PARAMETER["false_northing",0],
    UNIT["metre",1,
        AUTHORITY["EPSG","9001"]]]
Origin = (19999.999999999708962,7836760.000000000000000)
Pixel Size = (20.000000000000000,-20.000000000000000)
Metadata:
  AREA_OR_POINT=Area
Image Structure Metadata:
  COMPRESSION=LZW
  INTERLEAVE=BAND
Corner Coordinates:
Upper Left  (   20000.000, 7836760.000) ( 14d13'57.64"E, 70d10'55.87"N)
Lower Left  (   20000.000, 6597200.000) ( 18d34'19.38"E, 59d14'24.11"N)
Upper Right (  739780.000, 7836760.000) ( 33d27'12.66"E, 70d31'16.80"N)
Lower Right (  739780.000, 6597200.000) ( 31d13'48.20"E, 59d26'40.96"N)
Center      (  379890.000, 7216980.000) ( 24d26'49.89"E, 65d 3'17.00"N)
Band 1 Block=128x128 Type=Byte, ColorInterp=Palette
  Min=1.000 Max=48.000
```

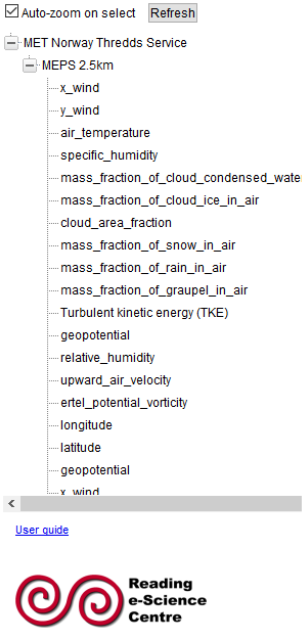
PNG

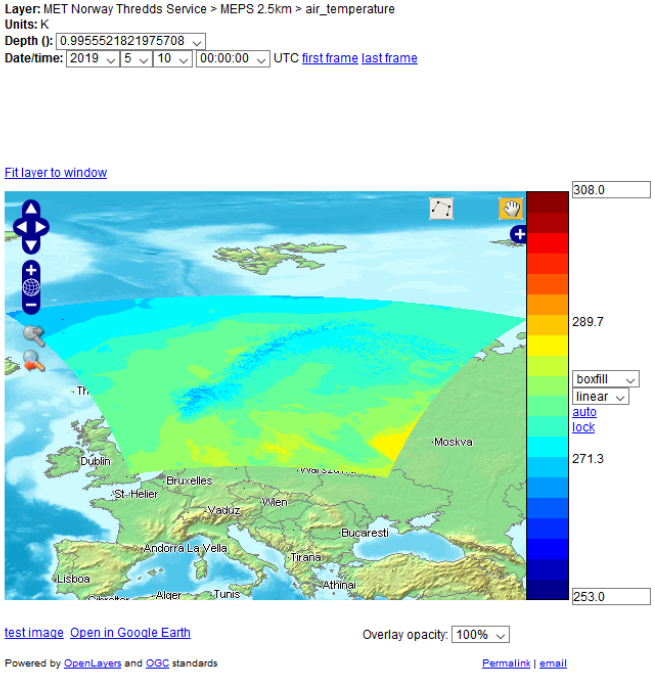
- **Lossless graphics format originally developed to replace GIF**
- **Used to render e.g. web maps**
- **Does not contain spatial metadata**
  - Metadata needs to be stored separately
- **Can have only limited number of data bands**
- **Used for raster data storage in GeoPackage**

https://maps.onyourmap.com/oym?f=m&ft=png_std_256&x=4&y=13&z=12&key=FO1349G5NGDTJ52H913SFRK63950
(from fonecta.fi/kartat)

NetCDF

- **Raster data is basically multidimensional (typically 2d) array data**
- **Network Common Data Form (NetCDF) is a format for array-oriented scientific data**
- **Used in earth sciences, together with the OpeNDAP protocol**
- **Typically handled as arrays of numbers instead of images**
- **Can be visualized as a raster layer on a GIS software**





Data: thredds.met.no weather data; air temperature layer shown on map

Aalto University
School of Engineering

Interface request and data download comparison

- **Request typically delivers the data inside a specific area**
  - GML, KML, GeoJSON, or similar
  - Data is not permanently stored at the client
- **Data download typically covers a whole dataset**
  - Shapefile, GeoPackage, etc.
  - Data is permanently stored at the client
- **Data stream vs. file/database**

- **Streamed data is designed to be read from beginning to the end, and then discarded while the data is used in-memory**
- **Files and databases are designed for the data to be read and modified**

Data streaming vs permanent data storage

- **Reliability of the data**
  - When data is on a server, you're dependent on the server being available
  - When the data is local, you can modify it (also by accident)
- **Use of internet**
  - Local data can be used even when you're not on-line
- **Data management and updates**
  - Server managers manages the dataset
    - User doesn't need to worry about updates
  - When the data is on your own computer, you need to update it manually

For the next time…

Do the third exercise round

Submit your learning diary for peer assessment

The next learning session will be on Thursday, October 1st