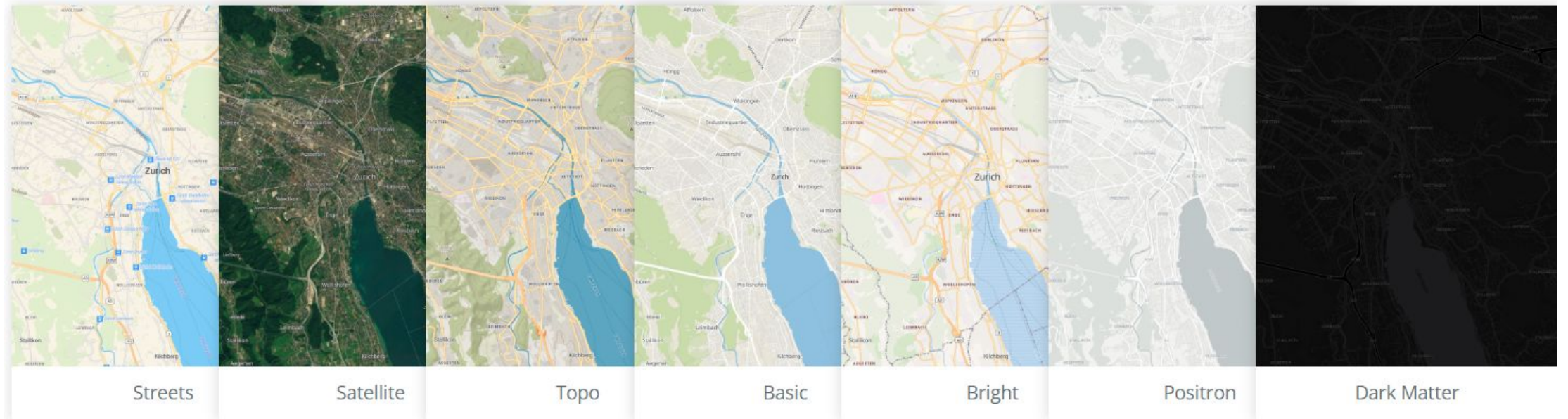
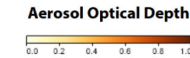
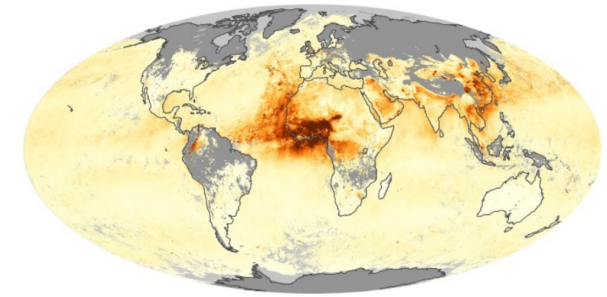


GeoContainers- OGC GeoPackage



Day 9 – OGC Winter School

By Vamsi Y, Senior Research Scientist, GISEHub, IIT Bombay

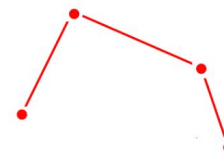
Why this presentation?

- Good time to survey the field of generic formats
- Great new options for sharing geodata
- GeoJSON
- GeoPackage: great balance of simplicity and performance
- Understand the Context Better

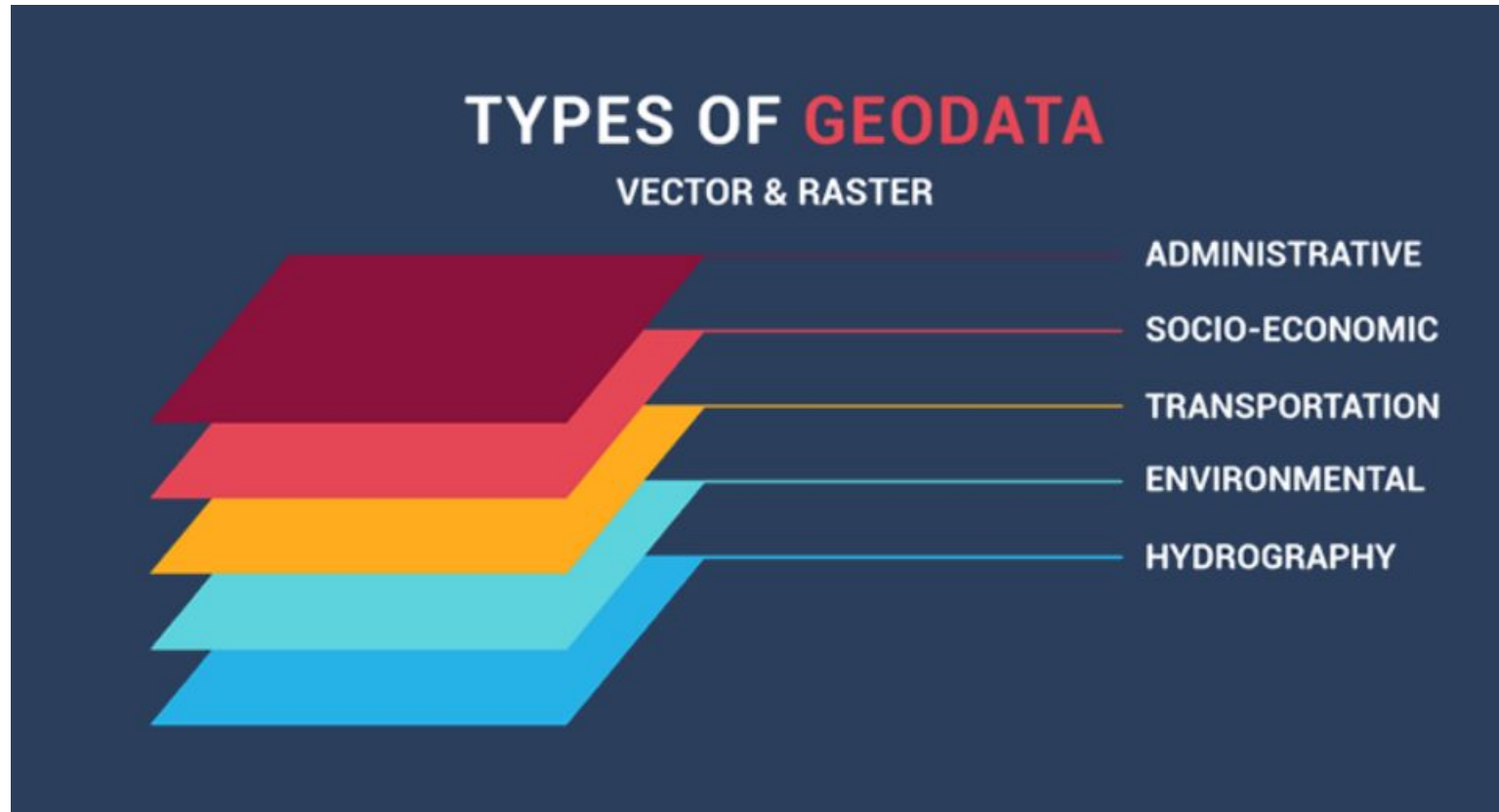
GEO Data

Types of Data

- Vector
- Raster
- GeoDatabases
- WebFiles
- Multi Temporal



Themes of Data



Team Activity -1

- <https://datacarpentry.org/organization-geospatial/aio/index.html>

GeoJSON

Geo+JSON

GeoJSON

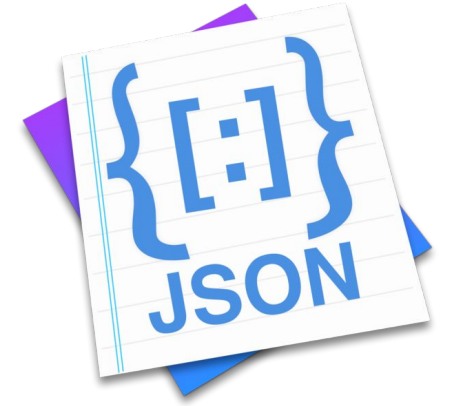
- GeoJSON is a subset of JSON (JavaScript object notation). It was developed around 2008 by a group of enthusiastic GIS developers. The core idea is to provide a specification for encoding geospatial data while remaining decodable by any JSON decoder.

JSON

- What is a JSON file
- JSON is a lightweight format for storing and transporting data
- JSON is often used when data is sent from server to Webpage
- JSON is “self-describing” and easy to understand

JSON syntax Rules

- Data is name/value pairs
- Separated by colons
- Data is separated by commas
- Curly brackets hold objects
- Square brackets hold Arrays



JSON Example [1]



```
{  
  "Song" : {  
    "title": "Out of Control ",  
    "band": "Rolling Stones",  
    "players": [  
      "Mick Jagger",  
      "Keith Richards",  
      "Charlie Watts",  
      "Ron Wood"  
    ]  
  }  
}
```



Why Use JSON ?

- Since the JSON format is text only, it can be easily sent to and from the server and used as a data format by any programming language
- Easy to understand
- JavaScript has a built in function to convert a string, written in JSON format ,in to native JavaScript objects:
 - `Json.Parse()`
- So if you receive data from a server ,in JSON format, you can use it like any other JavaScript object.

JSON _Example [2]

```
Let text = '{"students": [{"firstName": " Raghav" ,  
  "lastName": "Chadda" } ,'+  
  '{"firstName": "Manish" , "lastName": "Ganguly" } ,'+  
  '{"firstName": "Snighdha" , "lastName": "More" } ] }';
```

```
Let obj=JSON.parse(text)
```

JSON_Usage

```
/Function to get a short url with Google API
function shortURL (longURL) {
let url = {
  "longDynamicLink":
  "https://example.page.link/?link=http://www.example.com/&apn=com.example.android&ibi=com.example.
ios",
  "suffix": {
    "option": "UNGUESSABLE"
  }
}
$.ajax({
  url:
  'https://firebasedynamiclinks.googleapis.com/v1/shortLinks?&key=AIzaSyCM8UYPSDicKI0Y1dCBG5TMCpB6x
B9iotc',
  type: 'POST',
  contentType: 'application/json; charset=utf-8',
  data:JSON.stringify(url),
  contentType:"application/json",
  dataType:"json",
  success: function(response) {
let result = JSON.parse(response);
console.log(result);
}
});
}
```

GeoJSON Example [1]

- GeoJSON is a format for encoding a variety of geographic data structures.

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {},
      "geometry": {
        "coordinates": [
          78.47911152539268,
          17.39602012607405
        ],
        "type": "Point"
      }
    }
  ]
}
```

GeoJSON Example [2]

```
{  
  "type": "Feature",  
  "geometry": {  
    "type": "Point",  
    "coordinates": [17.445304796868236, 78.34968618220879]  
  },  
  "properties": {  
    "name": "Data Sciences And Analytics Centre,IIITH"  
  }  
}
```

GeoJSON Example [3]

```
{
  "type": "Feature",
  "properties": {
    "id": "aabbaa",
    "name": "Ark"
  },
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [76.078122854233, 30.96094250679], [41.624997854233, 34.47656750679], [52.874997854233, 33.07031750679],
        [63.421872854233, 40.10156750679], [76.078122854233, 30.96094250679] ]
      ]
    },
    "crs": {
      "type": "name",
      "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" }
    }
  }
}
```


GeoJSON

- GeoJSON supports the following geometry types : Point ,LineString,Polygon,MultiPoint,MultiLineString and MultiPolygon.
- Geometric objects with additional properties are Feature objects.
- Sets of features are contained by Feature Collection objects

Why GeoJSON?

- only well-defined generic JSON model for geodata
 - a lot like GML Simple Features
 - Not an OGC standard, but strong OGC member interest
 - lots of uptake
-
- OGC context :First tried out as a lightweight exchange format in Testbed 10 geosynchronization experiment
 - <http://www.geojson.org/geojson-spec.html>

Team Activity -2

Please Try out play around

<https://geojson.io/>

References :

- GeoJSON
 - <http://www.geojson.org/geojson-spec.html>

The Road to GeoPackage



GeoPackage Brief History

- January 2012 – First discussions
- October 2012 – GeoPackage SWG Charter published
- February 2014 – GeoPackage 1.0 adopted
- April 2014 – GeoPackage 1.0.1 adopted
 - Focus on *features*
- August 2015 – GeoPackage 1.1 adopted
 - Focus on *tiles*
- June 2017 – GeoPackage 1.2 *adopted*
 - Focus on *extensions*
 - Geo Package 1.3 is the latest [OGC Testbed-16: GeoPackage Engineering Report \(opengeospatial.org\)](https://www.opengeospatial.org/standards/ogc-testbed-16-geo-package-engineering-report)

GeoPackage



GeoPackage

[Read the Standard](#)

[Find Software](#)

[Follow GeoPackage](#)

Find me on GitHub

An Open Format for Geospatial Information

GeoPackage is the modern alternative to formats like SDTS and Shapefile. Its SQLite-based format efficiently stores and transfers geographic vector features and image tiles.

GeoPackage is the modern alternative to formats like SDTS and Shapefile. At its core, GeoPackage is simply a SQLite database schema. If you know SQLite, you are close to knowing GeoPackage. Install Spatialite – the premiere spatial extension to SQLite – and you get all the performance of a spatial database along with the convenience of a file-based data set that can be emailed, shared on a USB drive or burned to a DVD.

GeoPackage was carefully designed this way to facilitate widespread adoption and use of a single simple file format by both commercial and open-source software applications – on enterprise production platforms as well as mobile hand-held devices. GeoPackage is a standard from the Open Geospatial Consortium. It was designed and prototyped following a multi-year, open process of requirements testing and public input. It is designed for extension. So if you need more than the core GeoPackage feature set, join OGC's open process to standardize community-tested enhancements.

[View the Webinar](#) given recently by [Directions Magazine](#) in collaboration with OGC!



Official Standards Information

For all official, normative information on the GeoPackage standard, including PDF format download, see the [OGC standards program GeoPackage page](#).

 [Implementations](#)

 [Issue Tracker](#)

What is a GeoPackage

- GeoPackage was first developed and adopted by Open Geospatial Consortium (OGC) in 2014 , making it the official alternative for Shapefile. It is a subset of SQLite, which in turn is a lightweight SQL implementation designed for stand-alone databases. Similar to GeoJSON, this makes GeoPackage highly compatible by design, and accessible by non-GIS software as well.

What is a GeoPackage ..

- Open format for geospatial information
 - Vector geospatial features
 - Raster tile matrix sets (pyramids) of imagery and raster maps at various scales
 - Extensions
- SQLite database schema
 - Table definitions
 - Integrity assertions
 - Format limitations
 - Content constraints

Why GeoPackage?



Resource Intensive....Shapefiles, as a format, are dinosaurs
Evolving needs/Nature of usage...web services don't work
without internet access
internet sucks power from a mobile
device

Overview

- A **GeoPackage** is a platform-independent SQLite database file
- A **GeoPackage** may contain
 - Data in any geographic or projected CRS using any Datum
 - Vector Feature User Data Tables
 - GP BLOB Geometry Binary Format containing WKB Geometries
 - Linear 2D Geometries with optional elevation and measure values
 - Tile Matrix Pyramid User Data Tables
 - PNG and JPEG Tiles
 - Zoom times two (adjacent zoom level pixel sizes)
- First open source implementation
 - <https://bitbucket.org/luciad/libgpkg>
 - A SQLite 3 extension that provides a minimal implementation
 - distributed under the [Apache Software License](#) version 2.0

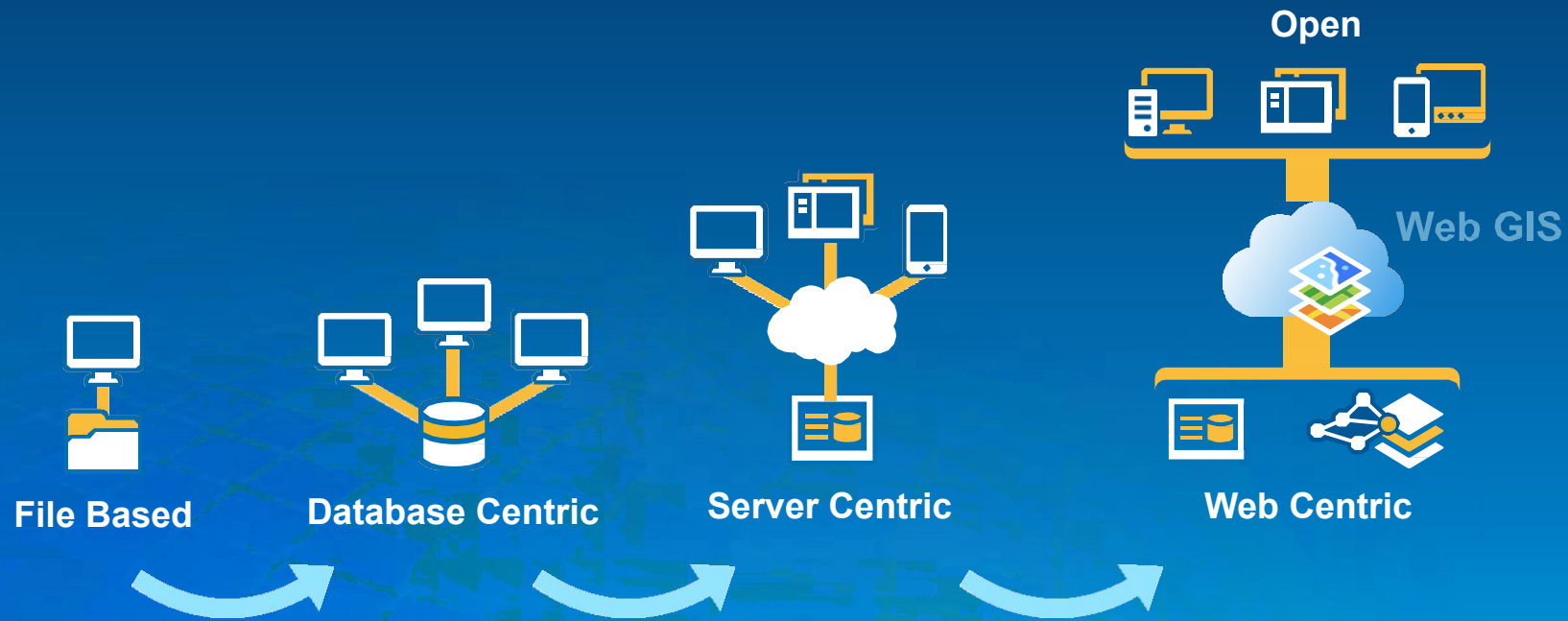
GitHub Hosting

- <https://github.com/opengeospatial/geopackage>
- We're the first SWG to do as much as possible on GitHub
 - Encourages public input
 - Provides more modern tools
 - Portal, Wiki, and email list still used to conduct SWG business and to protect participants' intellectual property
- Issue Tracker
 - <https://github.com/opengeospatial/geopackage/issues>
- <http://geopackage.org> hosted by GH Pages

Vector Tables

- Similar to other relational databases
- Built on existing standards
 - OGC Simple Features
 - Well-known Binary (WKB)
- Not subject to many of Shapefile limitations

GeoPackage Useable in Multiple Implementation Patterns



Emerging Uses

- Mobile applications
- Data dissemination
- Not the solution for everything
 - For simpler vector applications, GeoJSON may be plenty
 - In some enterprises an SQLite-based solution may be inappropriate

GeoPackage Implementation Examples

- GDAL
- Luciad Lightspeed, Mobile, and Fusion
- QGIS
- Esri ArcGIS (10.2.2 for vector, 10.3 for tiles)
- NGA
- SpatiaLite (4.2.0)
- GeoServer
- Compusult
- GeoTools
- US Army ERDC
- Envitia MapLink
- Terrago (GeoPDF)

Benefits

- Supports sharing of raster and vector geospatial information in a single container
- Supports direct use
 - Avoid intermediate format translations (extract, transform, load)
 - SQLite is a true relational database with built-in indexing
- Platform independent, supporting multiple computing environments (hardware and operating system)
 - Local storage reduces power requirements and supports disconnected/intermittent/limited connectivity
 - Conserves storage space by allowing multiple applications to access the same data store

GeoPackage Tables for Features

[Redacted]

- defines spatial reference systems

[Redacted]

- identifies user data tables

[Redacted]

- identifies feature geometries

[Redacted]

- contain feature data

SQLite tables in a sample file[†]

World.gpkg

- Master Table (1)
 - sqlite_master
- Tables (14)
 - gpkg_contents
 - gpkg_data_columns
 - gpkg_extensions
 - gpkg_geometry_columns
 - gpkg_metadata
 - gpkg_metadata_reference
 - gpkg_spatial_ref_sys
 - gpkg_tile_matrix_metadata
 - rtree_world_shape
 - rtree_world_shape_node
 - rtree_world_shape_parent
 - rtree_world_shape_rowid
 - sqlite_sequence
 - world**
- Views (0)
- Indexes (5)
 - sqlite_autoindex_gpkg_contents_1
 - sqlite_autoindex_gpkg_data_columns_1
 - sqlite_autoindex_gpkg_extensions_1
 - sqlite_autoindex_gpkg_geometry_columns_1
 - sqlite_autoindex_gpkg_tile_matrix_metadata_1
- Triggers (6)
 - rtree_world_shape_delete
 - rtree_world_shape_insert
 - rtree_world_shape_update1
 - rtree_world_shape_update2
 - rtree_world_shape_update3
 - rtree_world_shape_update4

TABLE gpkg_contents

rowid	table_name	data_type	identifier	description	last_change	min_x	min_y	max_x	max_y	srs_id
1	world	features	world		2013-09-20...	-180	-89.900...	180	83.533339	1

TABLE gpkg_geometry_colur

rowid	table_name	column_name	geometry_type	srs_id	z	m
1	world	shape	GeometryCollecti...	1	2	2

TABLE gpkg_spatial_ref_sys

srs_name	srs_id	organization	organization_coord...	definition	description
Undefined Cartesian	-1	NONE	-1	undefined	
Undefined Geographic	0	NONE	0	undefined	
EPSG:4326	1	EPSG	4326	GEOGCS["WGS_1984...	

TABLE world

COUNT...	CAPITAL	POP_19...	POP_G...	POP_M...	POP_FEM	POP_0...	POP_15...	POP_65...	MALE...	MALE...	65PLUS	FE
Afghani...	Kabul	15513267	5.2	7962397	7550870	7146575	7791404	575288	3610630	4014894	336873	3535
Albania	Tirane	1626315	1.8	835294	791021	1626315	0	0	835294	0	0	7910
Algeria	Algiers	22600957	2.5	11425492	11175465	9946100	11758841	893159	5087856	5894449	441519	4858

GeoPackage Tables for Tiles

- defines spatial reference systems

- identifies user data tables

- tile pyramid envelope

- describes tile zoom levels

- contain tile pyramids

SQLite command line example

- GeoPackage> .tables

```
gpkg_contents          gpkg_tile_matrix_metadata
gpkg_data_columns      rtree_world_shape
gpkg_extensions        rtree_world_shape_node
gpkg_geometry_columns  rtree_world_shape_parent
gpkg_metadata          rtree_world_shape_rowid
gpkg_metadata_reference world
gpkg_spatial_ref_sys
```

- GeoPackage> .schema world

```
CREATE TABLE "world" ( "COUNTRY" TEXT, "CAPITAL" TEXT, "POP_1994" INT, "POP_GRW_RT" DOUBLE,
"POP_MALE" INT, "POP_FEM" INT, "POP_0_14" INT, "POP_15_64" INT, "POP_65PLUS" INT, "MALE_0_14" INT,
"MALE_15_64" INT, "65PLUS" INT, "FEM_0_14" INT, "FEM_15_64" INT, "FEM_65PLUS" INT, "POP_URBAN" INT,
"POP_RURAL" INT, "URB_MALE" INT, "URB_FEM" INT, "RUR_MALE" INT, "RUR_FEM" INT, "ARABLE_PCT" DOUBLE,
"LITERACY" DOUBLE, "RATE" DOUBLE, "RATE0" DOUBLE, "GROWTH" DOUBLE, "CONTINENT" TEXT, "_id" INTEGER
PRIMARY KEY AUTOINCREMENT , "shape" GeometryCollection);
```

- GeoPackage> select country,capital from world limit 3;

```
Afghanistan|Kabul
```

```
Albania|Tirane
```

```
Algeria|Algiers
```

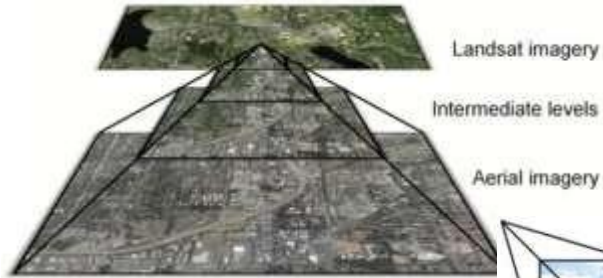
GeoPackage

<http://geopackage.org>

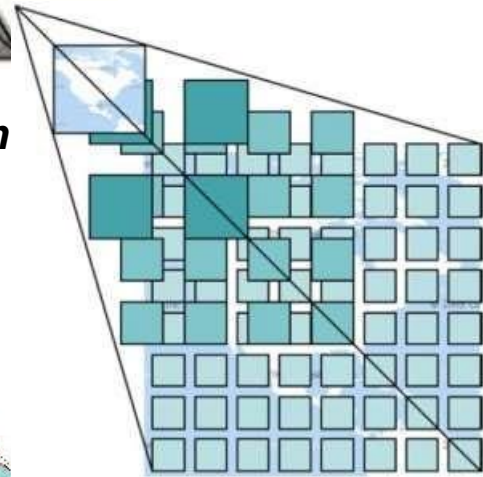


GeoPackage:

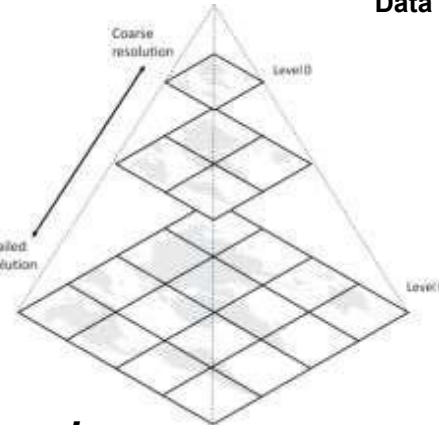
Raster Maps, Images and Feature Data in One File



Imagery
Low – High Resolution



Tile Pyramids
24 zoom levels

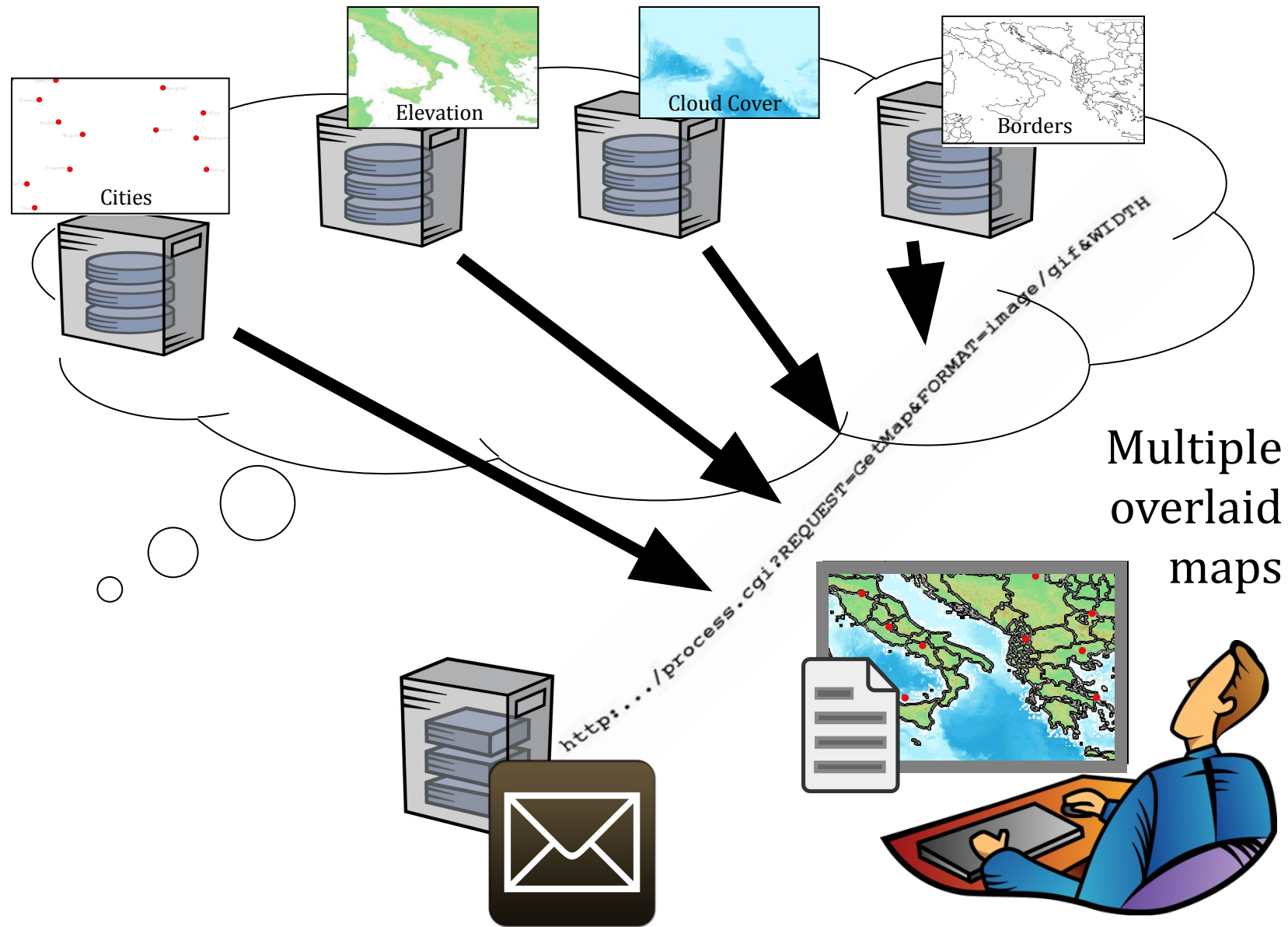


Single File Sqlite Database

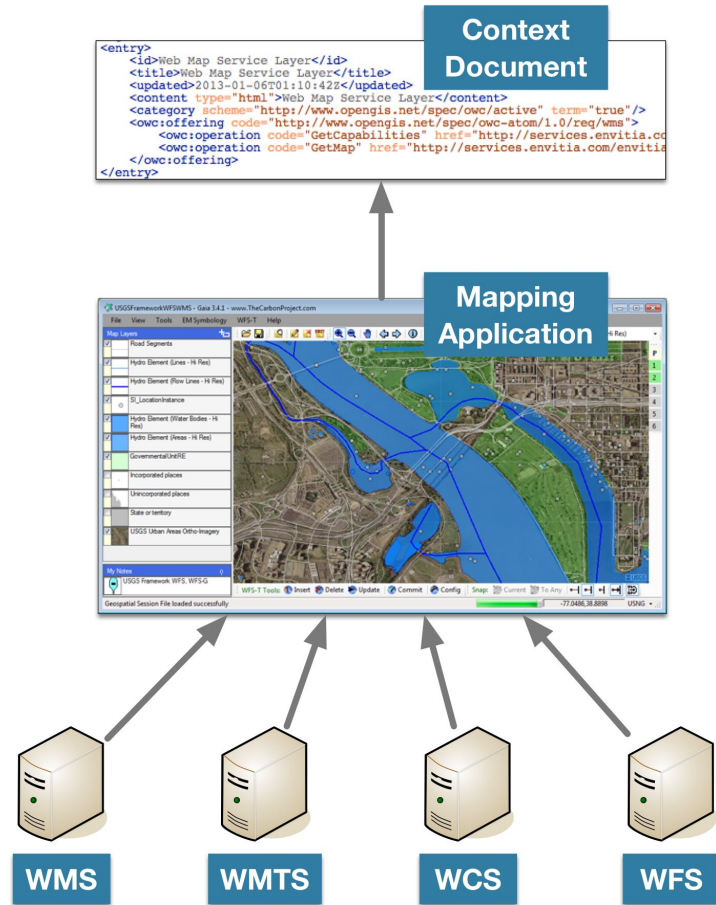
containing all data for direct-use on mobile platforms & handheld devices

OGC Web Context

Sharing a map



Anatomy of a Context document



- A context document extends XML format (or JSON).
- OGC <offerings>
- Offerings are data sources. They may be either services (and particularly OGC services like WMS, WFS, WMTS and WCS) or file-based data sets, like GML, KML and PDF.
- File-based data sets may be specified by reference with a URL, or included inline.
- OGC service offerings contain a GetCapabilities request and a request to get data, such as GetMap, GetTiles, GetFeature, etc.
- Offerings do not have to be spatial, so clients implementing Context should handle non-spatial offerings in a way other than mapping them.

Example Offering - WMS

```
<entry>
```

```
  <owc:offering code="http://www.opengis.net/spec/owc-atom/1.0/req/wms">
    <owc:operation code="GetCapabilities" method="GET" type="application/xml"
href="http://ows.genesi-dec.eu/geoserver/385d7d71-650a-414b-b8c7-739e2c0b5e76/wms?
SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities"/>
    <owc:operation code="GetMap" method="GET" type="image/png"
href="http://ows.genesi-dec.eu/geoserver/385d7d71-650a-414b-b8c7-739e2c0b5e76/wms?
SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&
SRS=EPSG:4326&BBOX=-2,45,8,55&WIDTH=500&HEIGHT=500&
LAYERS=385d7d71-650a-414b-b8c7-739e2c0b5e76:MER_RR__2PRLRA20120406_
102429_000026213113_00238_52838_0211.N1.tif&FORMAT=image/png&
BGCOLOR=0xffffffff&TRANSPARENT=TRUE&
EXCEPTIONS=application/vnd.ogc.se_xml"/>
  </owc:offering>
```

```
</entry>
```

Example Offering - WFS

```
<entry>
```

```
  <owc:offering code="http://www.opengis.net/spec/owc-atom/1.0/req/wfs">
    <owc:operation method="GET" code="GetCapabilities"
href="http://services.interactive-instruments.de/xsprojects/ows9-tds/services/ltds/wfs?
SERVICE=WFS&VERSION=1.0.0&REQUEST=GetCapabilities"
type="text/xml"/>
    <owc:operation method="GET" code="GetFeature"
href="http://services.interactive-instruments.de/xsprojects/ows9-tds/services/ltds/wfs?
SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature&
NAMESPACES=xmlns(tds,http:%2F%2Fmetadata.dod.mil%2Fmdr%2Fns%2FGSIP%2F3.0%2Ftds%2F3.0)
&OUTPUTFORMAT=text/xml;%20subtype=gml/3.2.1&
BBOX=-90,-180,90,180&TYPENAME=gml:AbstractFeature"/>
  </owc:offering>
```

```
</entry>
```

Example Context Document

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:owc="http://www.opengis.net/owc/1.0">
  <title>Sample Map Showing WMS and WFS</title>
  <id>http://www.example.com/ogcservices/map#dasfe3dsf</id>
  <author><name>Bob Smith</name></author>
  <updated>2020-12-03T12:00:30Z</updated>
<entry>

  <owc:offering code="http://www.opengis.net/spec/owc-atom/1.0/req/wms">
    <owc:operation code="GetCapabilities" method="GET" type="application/xml"
      href="http://ows.genesi-dec.eu/geoserver/385d7d71-650a-414b-b8c7-739e2c0b5e76/wms?
        SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities"/>
    <owc:operation code="GetMap" method="GET" type="image/png"
      href="http://ows.genesi-dec.eu/..."/>
  </owc:offering>

</entry>

<entry>

  <owc:offering code="http://www.opengis.net/spec/owc-atom/1.0/req/wfs">
    <owc:operation method="GET" code="GetCapabilities"
      href="http://services.interactive-instruments.de/xsprojects/ows9-tds/services/ltids/wfs?
        SERVICE=WFS&VERSION=1.0.0&REQUEST=GetCapabilities"
      type="text/xml"/>
    <owc:operation method="GET" code="GetFeature"
      href="http://services.interactive-instruments.de/xsprojects/ows9-tds/services/ltids/wfs..."/>
  </owc:offering>
</entry>
</feed>
```

Example WCS Context Document

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:georss="http://www.georss.org/georss"
      xmlns:owc="http://www.opengis.net/owc/1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xml:lang="en">
  <category scheme="http://www.opengis.net/spec/owc/specReference" term="http://www.opengis.net/spec/owc/1.0/req/atom" label="This file is compliant with version 1.0 of OGC Context"/>
  <id>http://www.opengis.net/owc/1.0/examples/AGU/wcs</id>
  <title>Context Example for AGU :: Web Coverage Service</title>
  <author><name>John Doe</name></author>
  <updated>2020-12-06T00:00:00Z</updated>
  <entry>
    <id>http://eos.ga.gov.au/thredds/wcs/LANDSAT/2008/10/LS7_ETM_NBAR_P54_GANBAR01-002_115_079_20081021_BX.nc</id>
    <title>Australian Reflectance Grid 25 (ARG25)</title>
    <updated>2020-08-06T00:00:00Z</updated>
    <dc:publisher>Geoscience Australia</dc:publisher>
    <author><name>John Doe</name></author>
    <dc:creator>Raj Singh with OxygenXML</dc:creator>
    <dc:source>http://eos.ga.gov.au/thredds/wcs/</dc:source>
    <summary>Australian Reflectance Grid 25</summary>
    <georss:where>
      <gml:Envelope srsName="EPSG:4326" srsDimension="2">
        <gml:lowerCorner>111.842625 -28.410875</gml:lowerCorner>
        <gml:upperCorner>114.357125 -26.434375000000003</gml:upperCorner>
      </gml:Envelope>
    </georss:where>
    <owc:offering code="http://www.opengis.net/spec/owc-atom/1.0/req/wcs">
      <owc:operation code="GetCapabilities" method="GET" type="text/xml"
href="http://eos.ga.gov.au/thredds/wcs/LANDSAT/2008/10/LS7_ETM_NBAR_P54_GANBAR01-002_115_079_20081021_BX.nc?service=WCS&version=2.0&request=GetCapabilities"/>
      <owc:operation code="GetCoverage" method="GET" type="image/jp2"
href="http://eos.ga.gov.au/thredds/wcs/LANDSAT/2008/10/LS7_ETM_NBAR_P54_GANBAR01-002_115_079_20081021_BX.nc?service=WCS&version=2.0&request=GetCoverage&coverageid=Band1&format=image/JPEG2000&BoundingBox=18.111.842625,-28.410875,114.357125,-26.434375000000003"/>
    </owc:offering>
  </entry>
</feed>
```

SQLite command line example

```
$ gpkg World.gpkg
```

- GeoPackage> .tables

```
gpkg_contents          gpkg_tile_matrix_metadata
gpkg_data_columns      rtree_world_shape
gpkg_extensions        rtree_world_shape_node
gpkg_geometry_columns  rtree_world_shape_parent
gpkg_metadata          rtree_world_shape_rowid
gpkg_metadata_reference world
gpkg_spatial_ref_sys
```

- GeoPackage> .schema world

```
CREATE TABLE "world" ( "COUNTRY" TEXT, "CAPITAL" TEXT, "POP_1994" INT, "POP_GRW_RT" DOUBLE, "POP_MALE" INT,
"POP_FEM" INT, "POP_0_14" INT, "POP_15_64" INT, "POP_65PLUS" INT, "MALE_0_14" INT, "MALE_15_64" INT, "65PLUS"
INT, "FEM_0_14" INT, "FEM_15_64" INT, "FEM_65PLUS" INT, "POP_URBAN" INT, "POP_RURAL" INT, "URB_MALE" INT,
"URB_FEM" INT, "RUR_MALE" INT, "RUR_FEM" INT, "ARABLE_PCT" DOUBLE, "LITERACY" DOUBLE, "RATE" DOUBLE, "RATE0"
DOUBLE, "GROWTH" DOUBLE, "CONTINENT" TEXT, "_id" INTEGER PRIMARY KEY AUTOINCREMENT , "shape"
GeometryCollection);
```

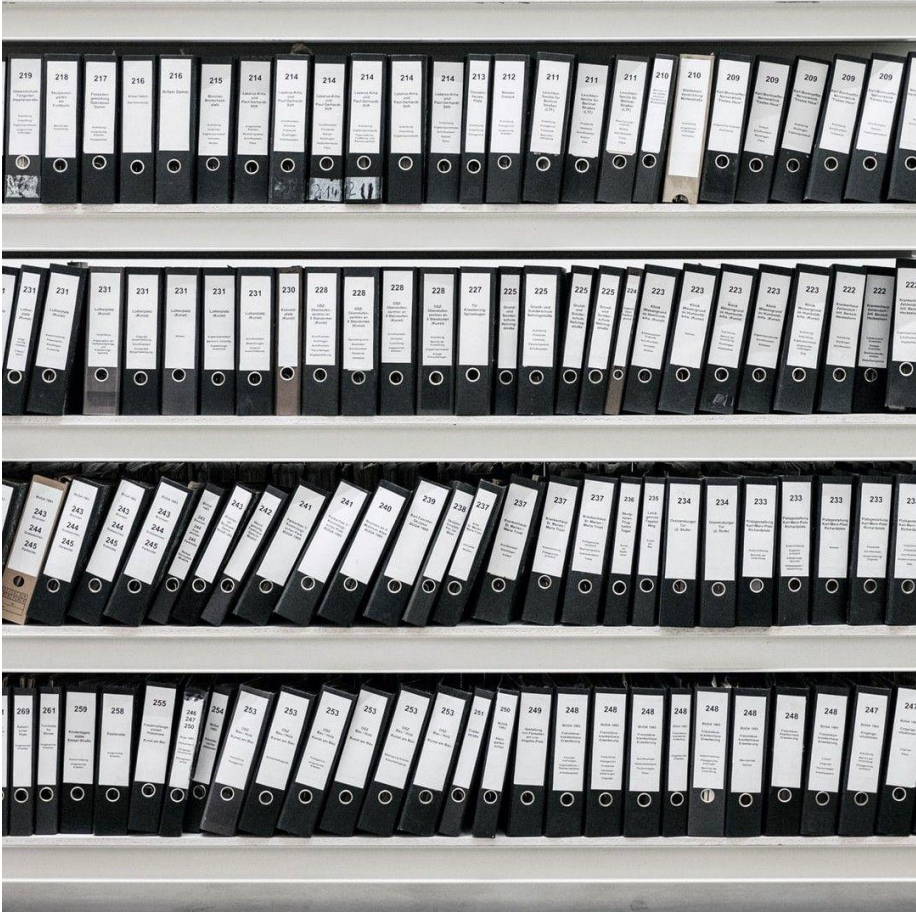
- GeoPackage> select country,capital from world limit 3;

```
Afghanistan|Kabul
```

```
Albania|Tirane
```

```
Algeria|Algiers
```

The Dilemma



Shape Files ?

- The Shapefile in fact consists of several files: in addition to one file with the actual geometry data, another file for defining the coordinate reference system is needed, as well as a file for defining the attributes and a file to index the geometries. This makes operating Shapefiles slightly clunky and confusing. However, Shapefile has been around for so long that any GIS software supports handling it.
- Internally, Shapefile uses Well-known binary (WKB) for encoding the geometries. This is a compact format which is based on tabular thinking, i.e. the row and column number of a value is significant. A minor nuisance is the limitation of the attribute field names to 10 characters and poor Unicode support, so some abbreviations and forcing to ASCII may have to be used.

Context

- Going Beyond the Shape File , [Need For it](#)
- Why OGC standard_GeoPackage
- Pros and Cons
- Uses case for a GeoPackage

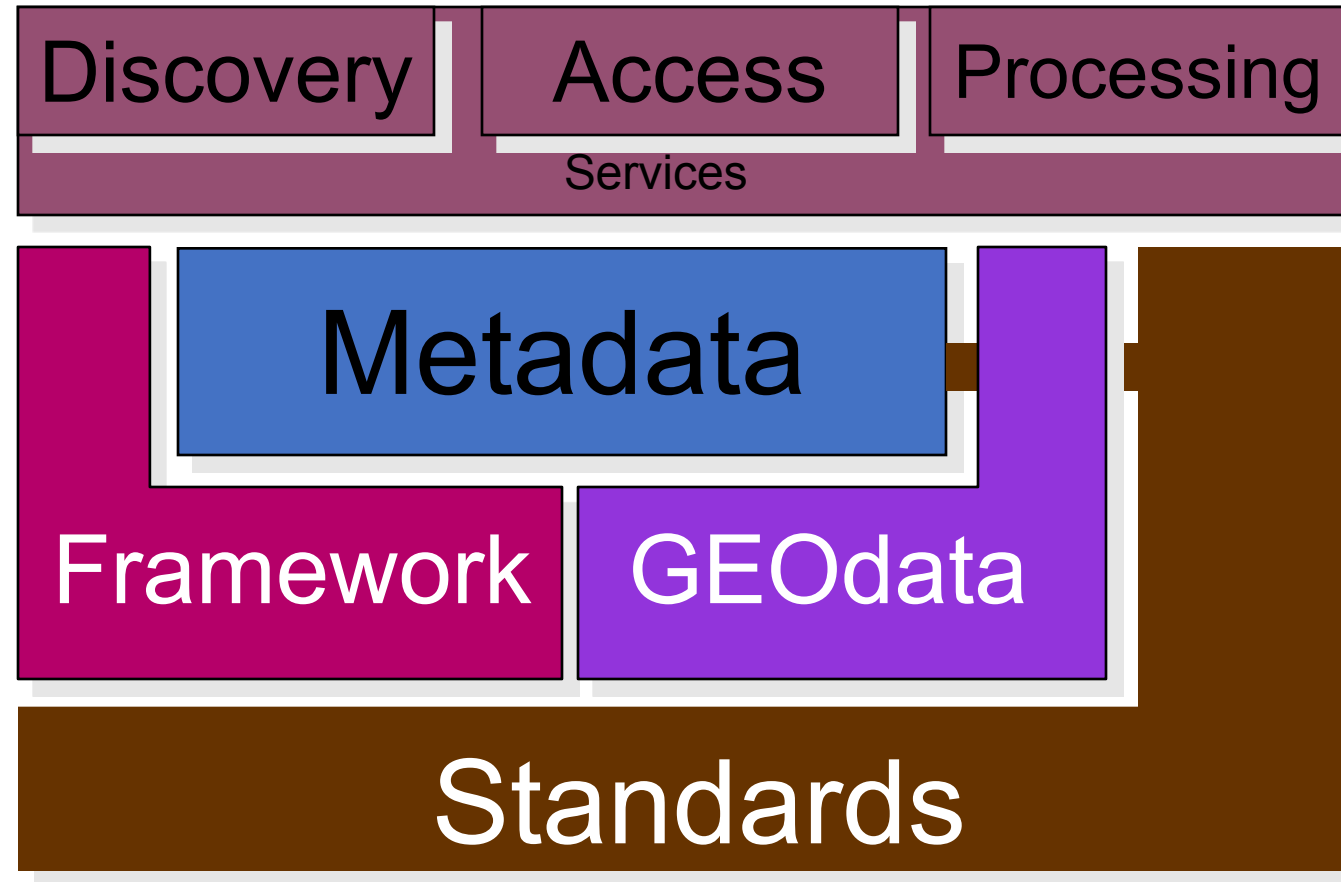
Making Good Extensions

- May be developed by the SWG or brought in from outside
- Must have:
 - Clear Use Case
 - Sound technical approach
 - Must be compatible with clients that just implement the core
 - Commitment from critical mass of vendors (usually 3) to implement

- What if you were given the option to choose your GIS file format freely, is there one format which is technically superior to the others?

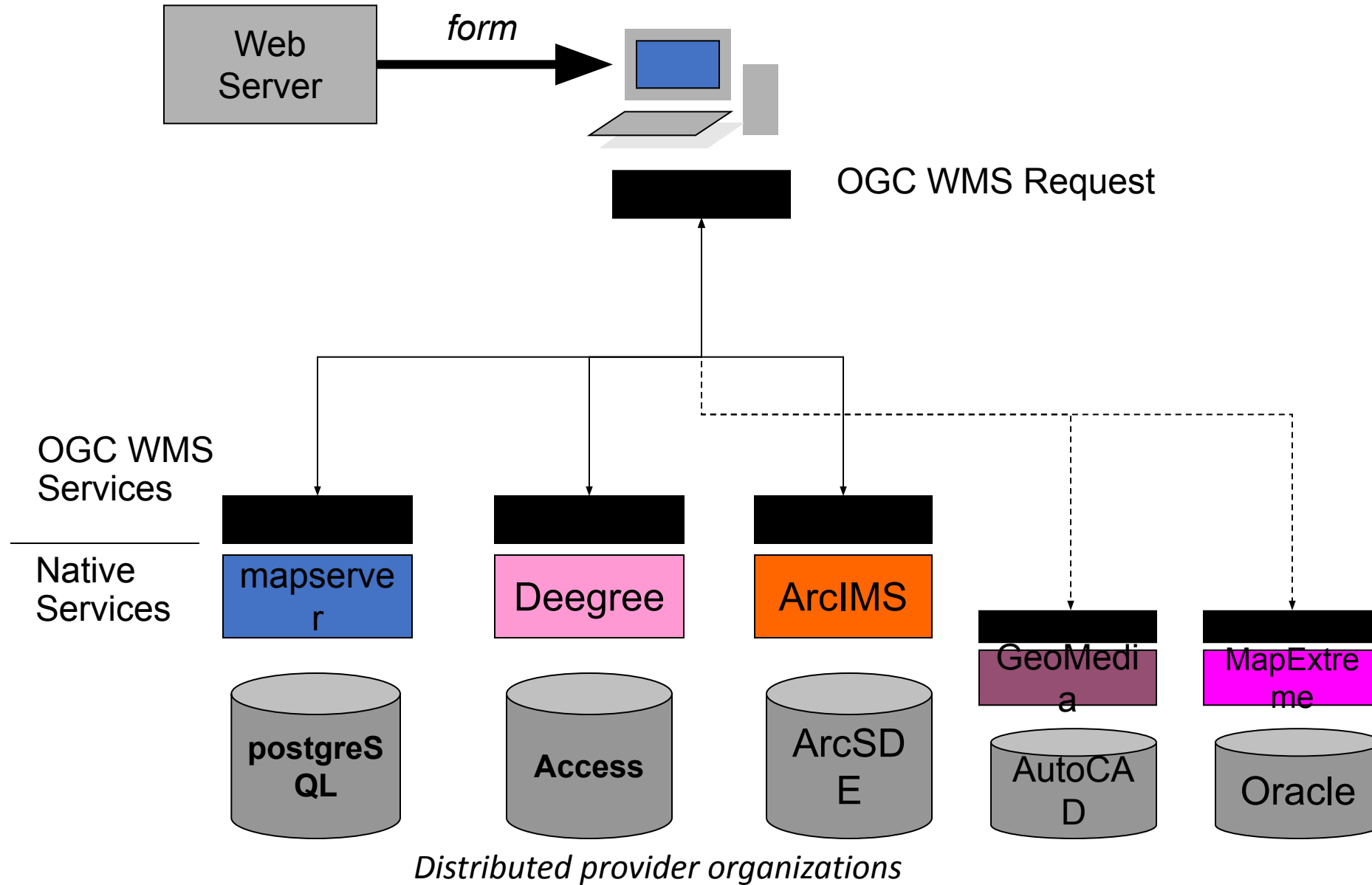
Shapefile vs. GeoJSON vs. GeoPackage

Standardization makes SDI work
Standards touch every SDI activity



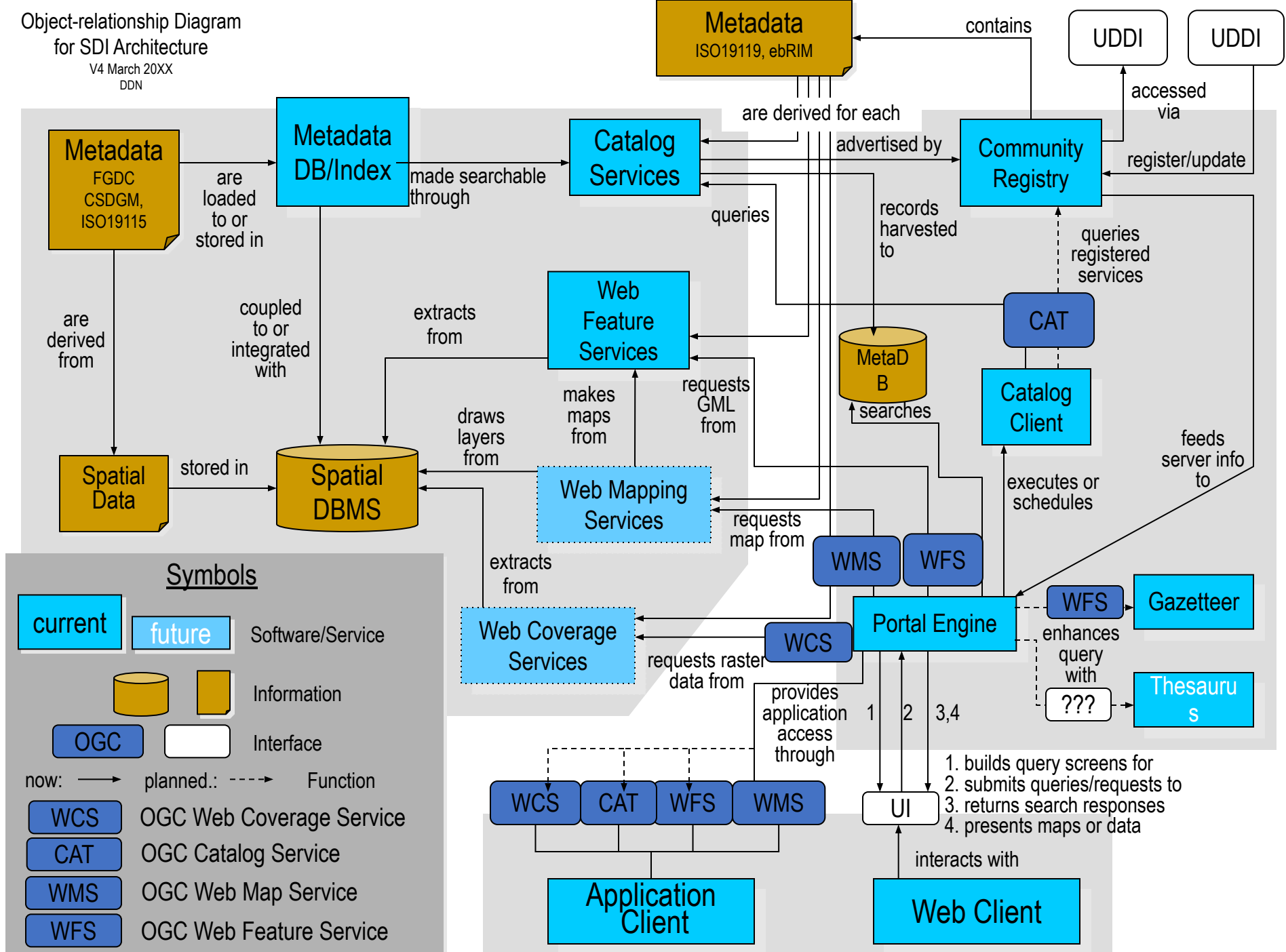
Standards include specifications, formal standards, and documented practices

Interoperable data access



Object-relationship Diagram
for SDI Architecture

V4 March 20XX
DDN



Thoughts...Reflection...Discussion

- Geospatial data in Shapefile, GeoJSON and Geopackage formats. Your choice of file format is likely based on the Use case, tools you have and the formats you are used to. But what if you were given the option to choose freely? Is there one file format that is technically superior to the others?
- Understand the context better - Security ,Performance Considerations.

- Try it out

Team Activity -3

1. [GML to GeoPackage Converter Online - MyGeodata Cloud](#)

Check the geopackage using QGIS

2. Take a look inside the actual SQL Lite database (.gpkg file) using DB Browser or any database browser tool
3. Display the geopackage feature class in QGIS
4. Each of team can do their own geopackage and exchange the same

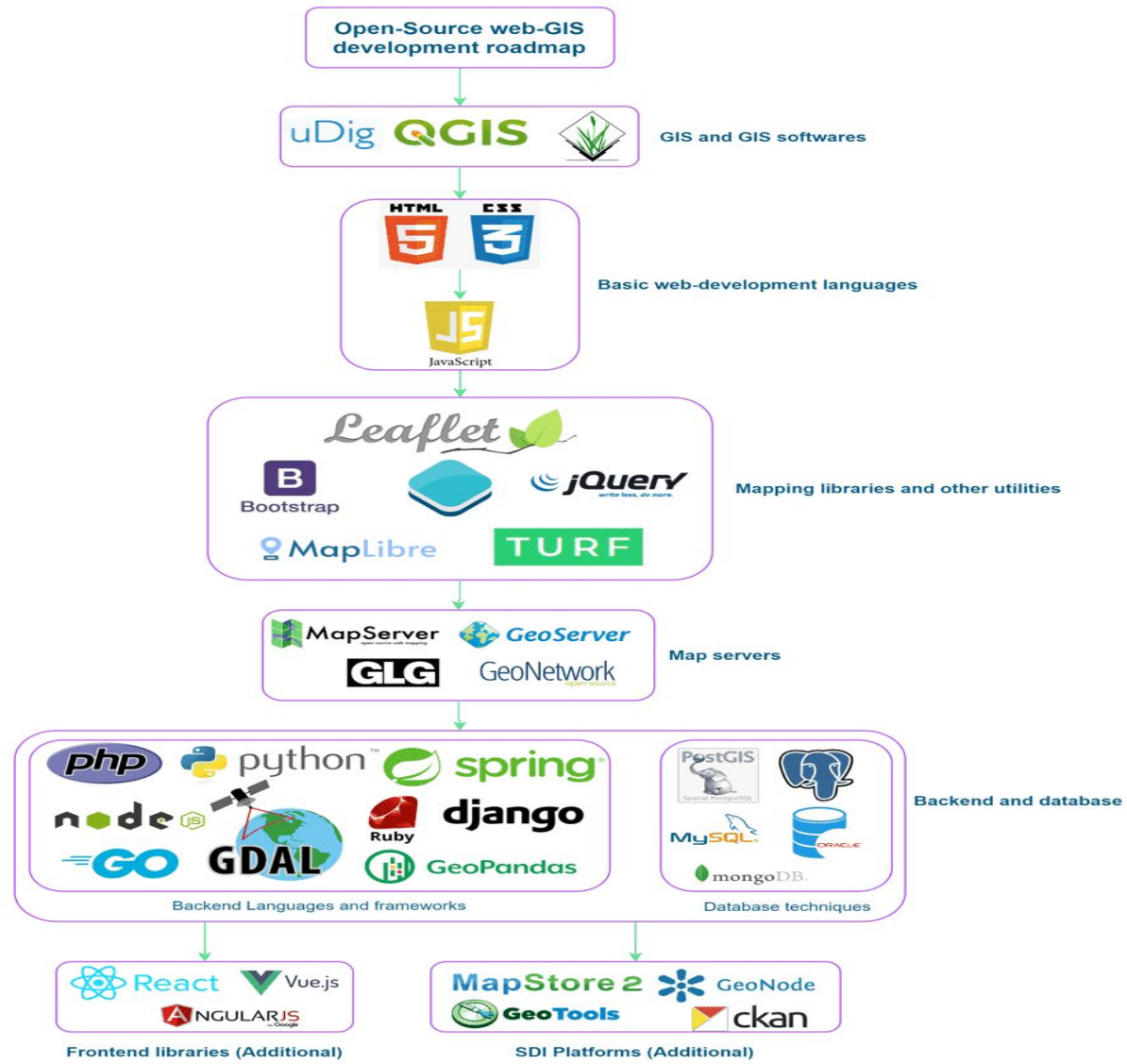
Other Exercises

- Create/Extend a Geopackage
- Use it in your applications
- Any non database geo-files could be a immediate and straight forward/good use case
- Lets say ,Any thing that is not in PostGIS

Time Permitting – Try this

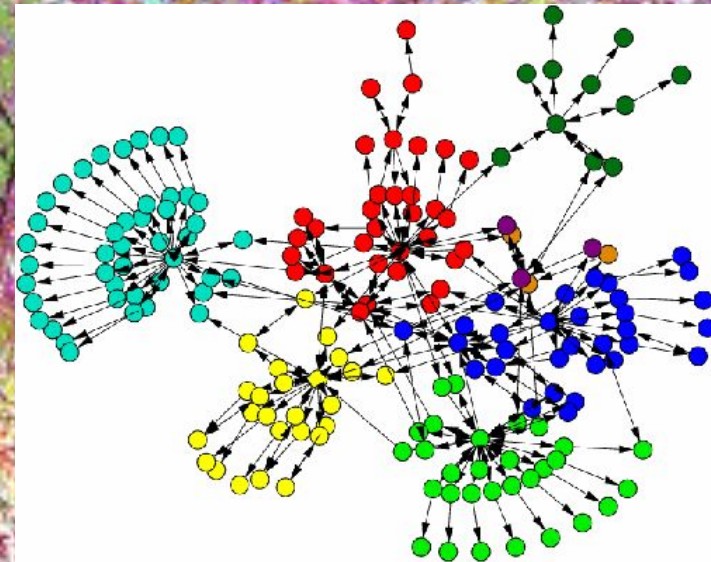
- Imagery in to Geo package...challenges
- <https://scihub.copernicus.eu/dhus/#/home>

The Ecosystem

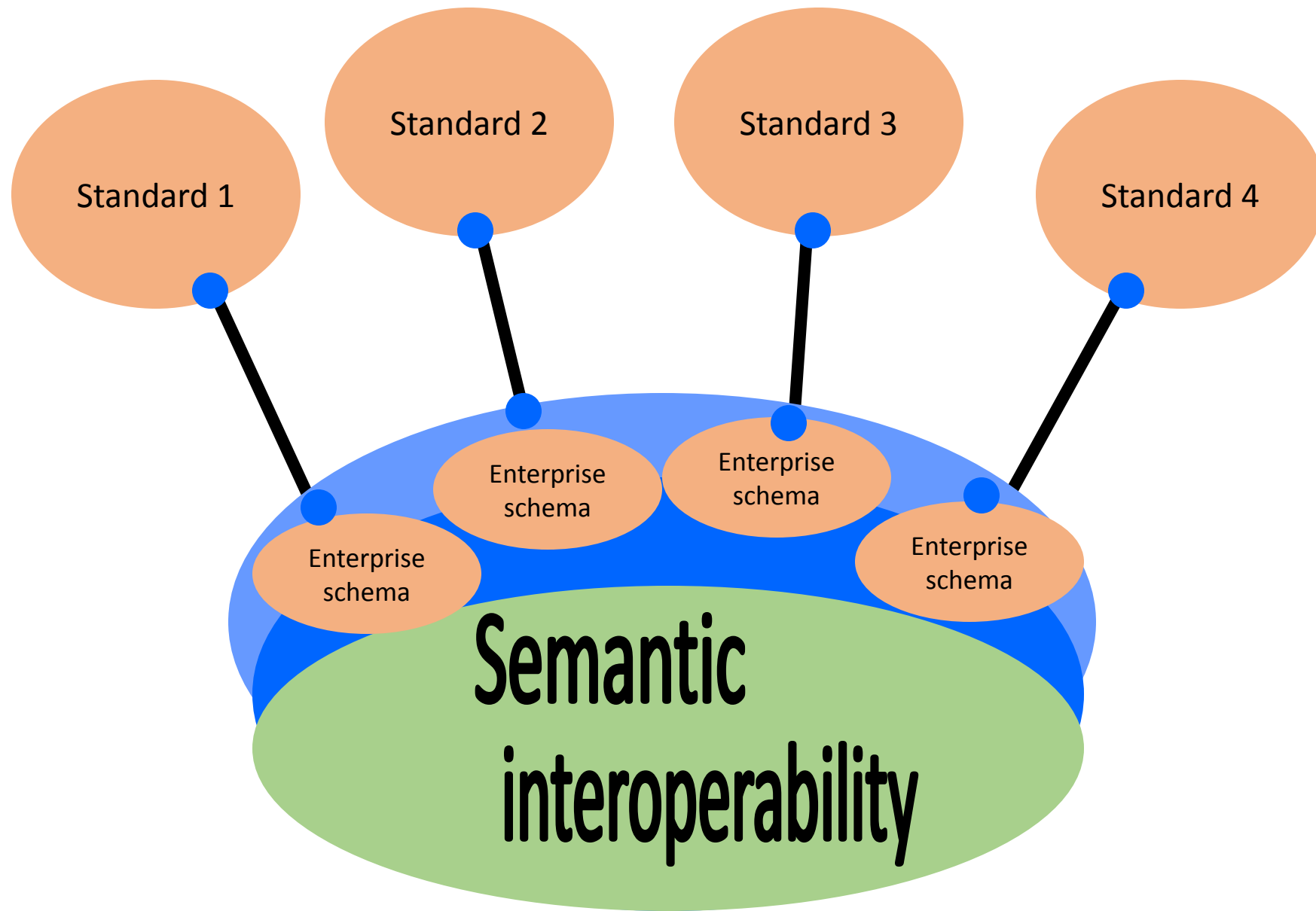


Copyright:
Tek Kshetri and Emmanuel Jolaiya

Patterns from different sectors *emerge*.....Standards



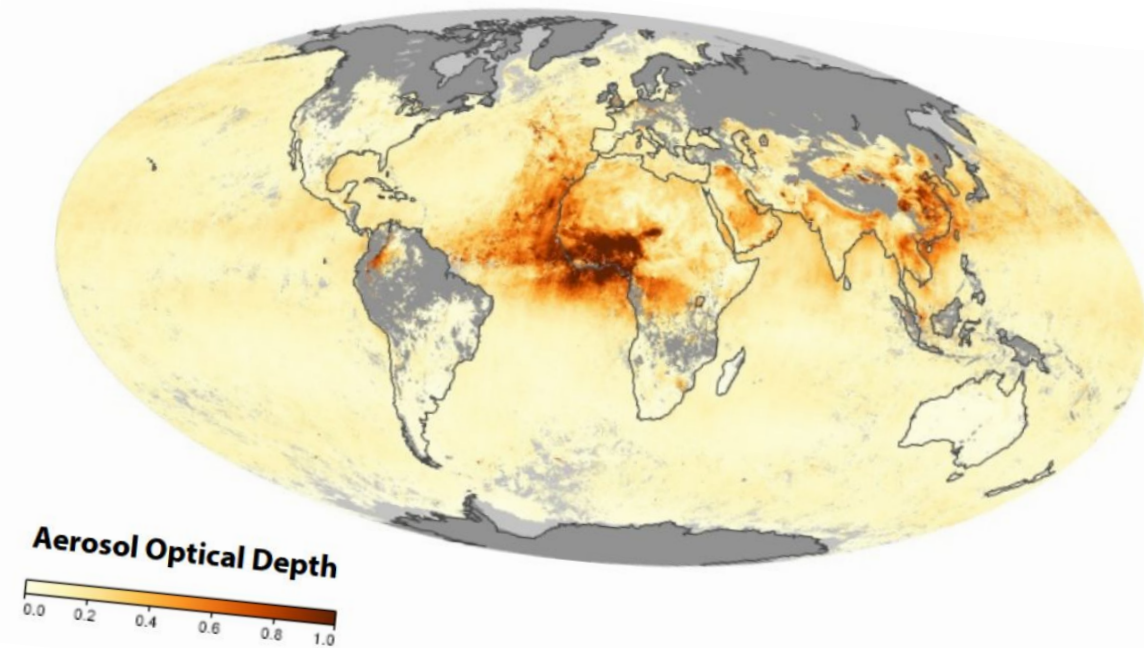
Is a developmental approach required?



Partnerships are the glue...Adoption of Standards

- Proper governance of the community is essential through a variety of roles and responsibilities
- National government or NGOs and Technical working groups should partner with other levels of government and sectors to promote Multi coordination
- The government or a foundation may be able to fund agencies with “seed” funding to further existing efforts toward common goals
- Partnerships extend local capabilities in technology, skills, logistics, and data

Our earth is tilted - 23.5 degrees relative to the axis of rotation around the sun...Similarly our understanding of the uses cases and hence standards evolve too



Look Forward to Continued Emergence....

Thank you