Web-mapping

Functions. Structure. Tools.

Advantages of web mapping

- Easily accessing geospatial data from everywhere
- Interact with geospatial data
- Great for continuously changing data (easy to update)
- Great for community data, easy to share
- Easy to deploy on many different platforms

Disadvantage / limitations

Web mapping user can be limited by:

- Internet access
- Distant services, servers : no long-term waranty, sudden modifications of functionning or content

Exploitation of spatial personnal data for marketing purposes (Google, and all other geo-services): *Location Based Services* … and *Advertisement.*

Short lifetime (*trashable maps)*

Example

Google Maps Engine

Evolved web application: wide range of functionnalities (only advanced geo-processing functions are missing)

https://www.google.com/maps/d/

Functionnalities

- **Navigation**
  - Zoom, panning
  - Localisation (by adress, place, postal code, country...)
  - Display at a defined scale (for instance 1:100000)
- **Layer visualization**
  - Compose the map
  - Manage layers order and transparency
  - Choose map background: aerial, satellites, topographic
  - Show labels and popups (hover, click)
  - Change symbology
- **Export : print or save (image/vecteur)**

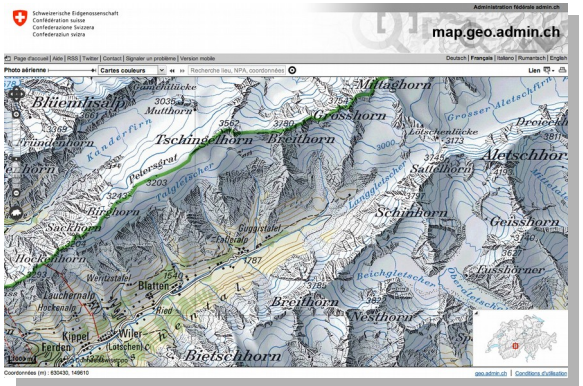
Functionnalities

- **Queries**
  - Information tool (select and display the attributes)
  - Distance or area measurement
  - Spatial queries
  - Extraction de données (géographiques ou attributs)
- **Temporal animations**
- **Dynamic calculation**
  - Road itineraries
  - Aggregations
  - Geo-processing (*Web GIS*)

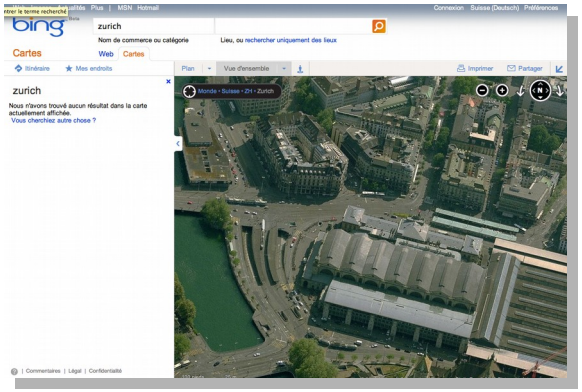
Functionnalities

- **Real time display of data**
  - webcam, GPS, weather information, data from sensors
- **Advanced web application**
  - User identification
  - Personnal profile can be saved (own maps, configuration, personnal data,...)
  - Save and share bookmarks (hyperlinks)
  - Add and save new spatial features: point, line, polygon
  - Links with collaborative websites and social networks

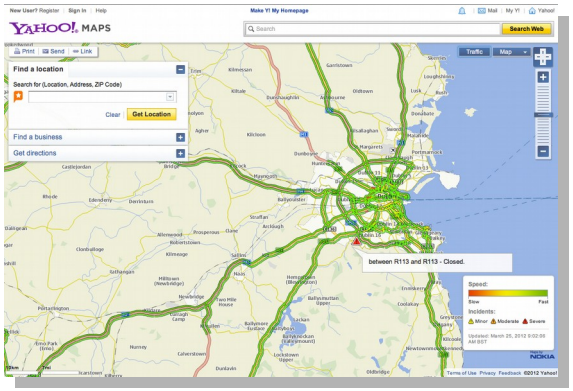
Other examples...



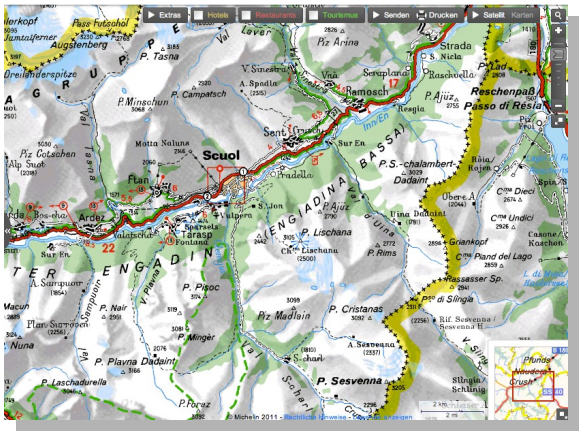
http://map.geo.admin.ch/



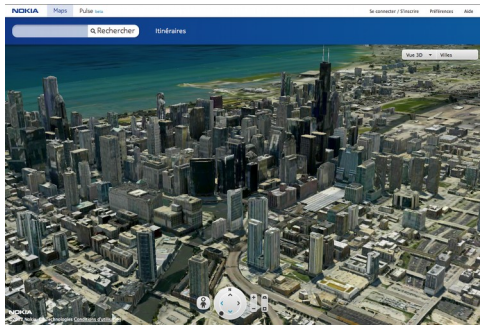
http://binged.it/H0tUsH



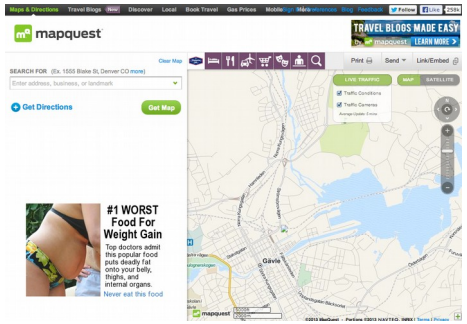
http://maps.yahoo.com



http://fr.viamichelin.ch/



http://here.com



http://www.mapquest.com/

Types of services and maps

Distinguish:

- **Maps** services / servers
- **Data** services / servers
- **Processing** services / servers

Growing GIS expertise

- **Static** map (predifined picture)
- **Interactive** maps
- **Dynamic** maps

How does it work?

Client – serveur infrastructure

- **Client** = web navigator

  For instance: Internet Explorer, Firefox, Chrome, Safari, Edge ...

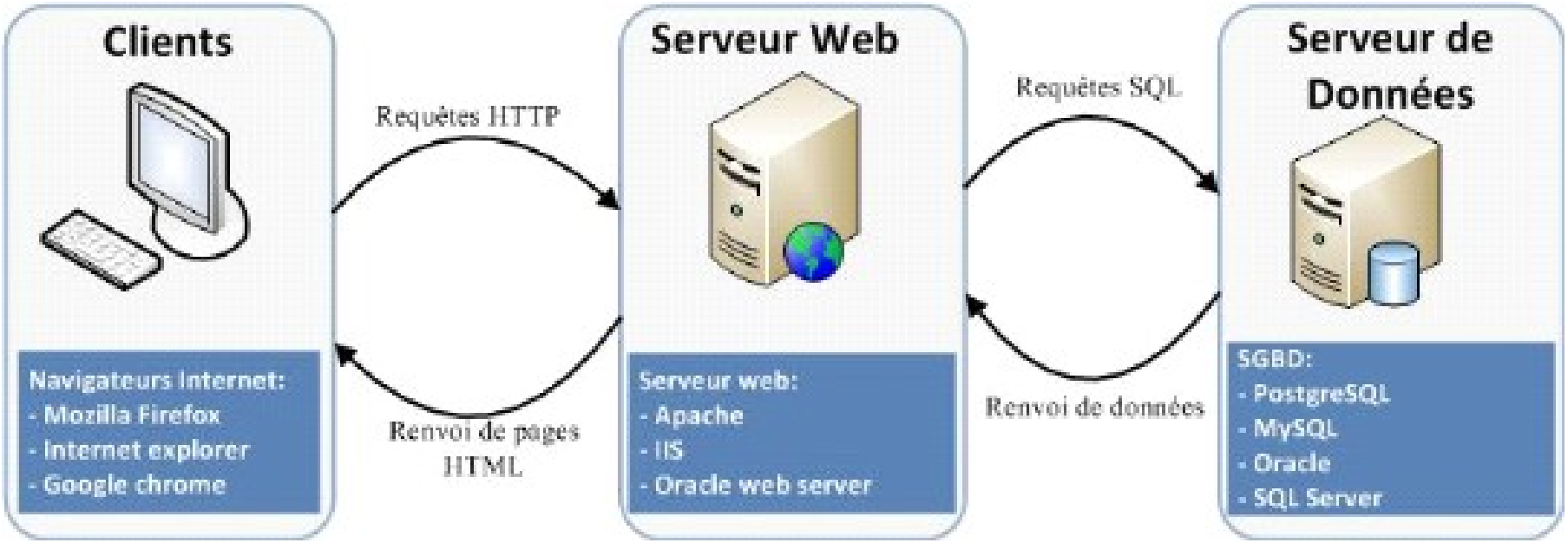
- **Server** = programm sending information according to HTTP protocol (Hyper Text Transfert Protocol).

  For instance: Apache, IIS, Google Web Server, ...

- **Principle**: the client sends a query to the server; the server sends back the requested information (a picture, some text...).

How does it work?

Client – serveur infrastructure



On side

A web navigator can process data in different formats :

- **HTML**
- **CSS**
- **Pictures** (PNG, GIF, JPEG)
- Vectors (SVG, but not for older navigators)
- 3D (WebGL, sometimes)
- … and other, through plugins: Flash, PDF, Java…

Web navigators also interpret **JavaScript** language. This language allow manipulating HTML, CSS, SVG (etc) data.

How to learn web langages

Every website is an example

Show sourcecode (Ctrl+U)

Have a look on every page element (Firefox, Chrome)

Official sources, among many others

HTML : www.w3schools.com/html

CSS : www.w3schools.com/css

Javascript: http://www.w3schools.com/js/

On side

Display a map in a web navigator:

- As a (PNG, JPEG, GIF)
- As a group of (SVG)
- As a Flash graphic (with plugin)
- As a WebGL media
- As HTML5 (in a <canvas> element, with JavaScript)

And use JavaScript to add interactivity.

How to build his own web mapping application?

Use a JavaScript library, with an API to use it easily :

- Google Maps API, OpenLayers, Leaflet, etc.

Prepare the data according to the selected library. Some formats don't use geographic coordinates (Flash, SVG), others use only one coordinate reference system: WGS 84.

Current data formats:

Raster data: XYZ tiles ; WMS (Web Mapping Service) ; WMTS (Web Mapping Tile Service)

Vector data (geometry and attributes): KML / KMZ ; GeoJSON ; WFS (Web Feature Service)

KML format

OGC standard, based on XML, then easily readable by a humanbeing

Contents geometries, attributes, and styles

Can also content links to raster files (pictures)

Can be used for 3D objects

Easy to create (Google Earth and Sketchup, GIS, par un script)

Disadvantages:

Heavy files (better with compressed KMZ files)

All options are not supported by all programms

System take time to read (XML > client's memory)

KML format: example

```
<?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>-122.082,37.422,0</coordinates>
    </Point>
  </Placemark>
</kml>
```

GeoJSON format

JSON extension (JavaScript Object Notation), widely used.

A GeoJSON includes JavaScript objects and variables. It is therfore very fast to be read by the client, which itself uses Javascript.

Still readable by humanbeings.

Quite eay to create (GIS, PostGIS, ogr2ogr ; very easy with a script)

Disadvantages:

Need additionnal libraries for some API like GoogleMaps.

No validation of data structure: GeoJSONs can have errors.

GeoJSON format: example

```
{ "type": "FeatureCollection",
  "features": [
    { "type": "Feature",
        "geometry": {
        "type": "Point",
            "coordinates": [-122.082,37.422]
        },
      "properties": {
          "description": "a simple point",
          "temperature": 10.5,
      }
    }, { ... another Feature ... }, ...
  ]
}
```

Basic web map example: HTML structure

With **Leaflet** library, tile background furnished by **OpenStreetMap** and one **GeoJSON** vector feature.

```
<html>
  <head>
    <link rel="stylesheet" href="leaflet.css" />
    <script src="leaflet.js"></script>
    <script src="script.js"></script>
  </head>
  <body onload="main()">
    <h1 id="title">GeoJSON Example</h1>
    <div id="map" style="background-color: #666; width: 800px; height: 600px;"></div>
  </body>
</html>
```

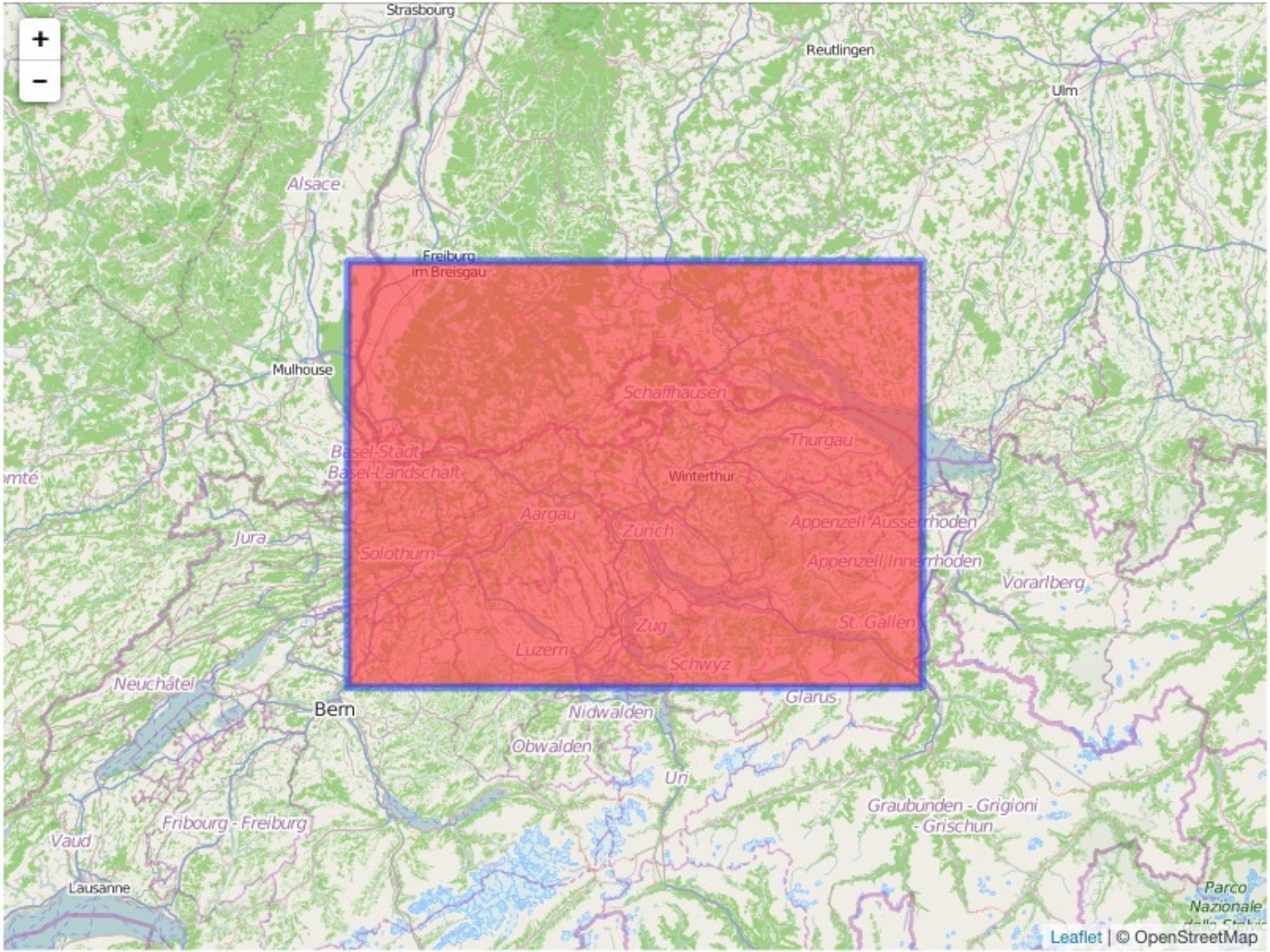
Basic web map example: JavaScript part

```
function main() {
  map = L.map("map").setView([8.5, 47.5], 8)
  L.tileLayer("http://{s}.tile.osm.org/{z}/{x}/{y}.png", { attribution: "&copy OpenStreetMap" }).addTo(map);

  var rectangle = {
    "type": "FeatureCollection",
      "features": [{
        "type": "Feature", "geometry": {
          "type": "Polygon",
            "coordinates": [[ [7.5, 47.0], [9.5, 47.0], [9.5, 48.0], [7.5, 48.0], [7.5, 47.0] ]]
      }, "properties": { "geocode": 1 } }]
  };
  L.geoJson(rectangle.features, {
    style: {fillColor: "#f33", fillOpacity: 0.6}
  }).addTo(map);
}
```

< GeoJSON

GeoJSON Example



Questions ?