



- **1. Généralités**

- **SIG Web**

- » Accès aux fonctionnalités offertes par un SIG à travers le web
    - **Production d'information**
    - **Communication**
    - Acquisition des données
    - Maintenance
    - Gestion des données
    - Intégration des données
  - » Exploitation d'un ou plusieurs **serveurs** avec différents outils
    - Base de données (ex: PostgreSQL + PostGIS)
    - Serveur web (ex: Apache)
    - Serveur de données spatiales (ex: GeoServer)
  - » Exploitation d'un ou plusieurs logiciels **clients** communiquant avec le/les serveur(s) à travers le **web**
    - Navigateur (Chrome, Firefox, Edge, ...)
    - Logiciel SIG (QGIS, ArcGIS, ...)

} Cartes, tableaux et graphiques interactifs

- **2. Programmation web**

- **Langages**

- » Nécessite la manipulation de plusieurs langages

- Interprétés par le **serveur**

- **SQL**

- Langage de manipulation de la base de données

- **PHP** et/ou **Python** et/ou ...

- Langage orienté objet du serveur web

- Interprétés par le **navigateur (client)**

- **HTML**

- Langage de description du contenu « statique » d'une page web

- **CSS**

- Langage de description du style d'un document HTML

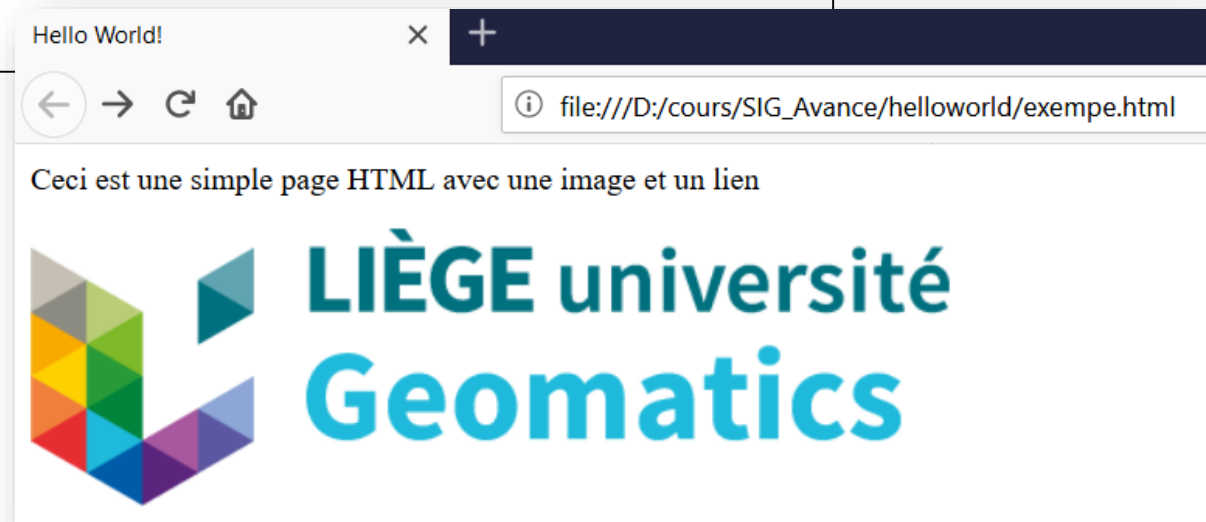
- **Javascript**

- Langage orienté objet permettant l'exécution d'opérations dynamiques par le navigateur

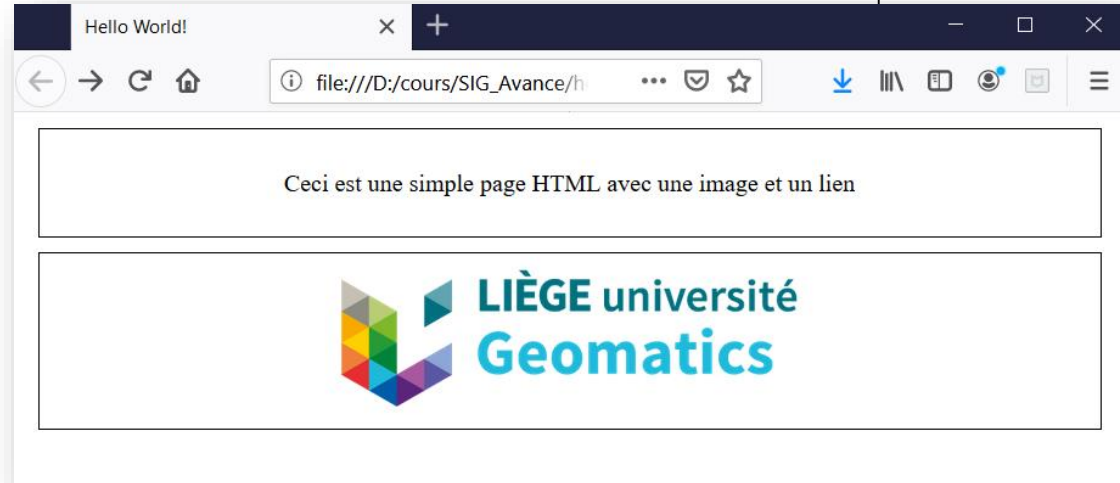
- Note: javascript peut aussi être utilisé comme langage serveur via le serveur web **NodeJS**

```
<html>
  <head>
    <title>
      Hello World!
    </title>
  </head>
  <body>
    <p>
      Ceci est une simple page HTML avec une image et un lien
    </p>
    <a href="http://geomatics.ulg.ac.be">
      
    </a>
  </body>
</html>
```

*Exemple de page HTML*

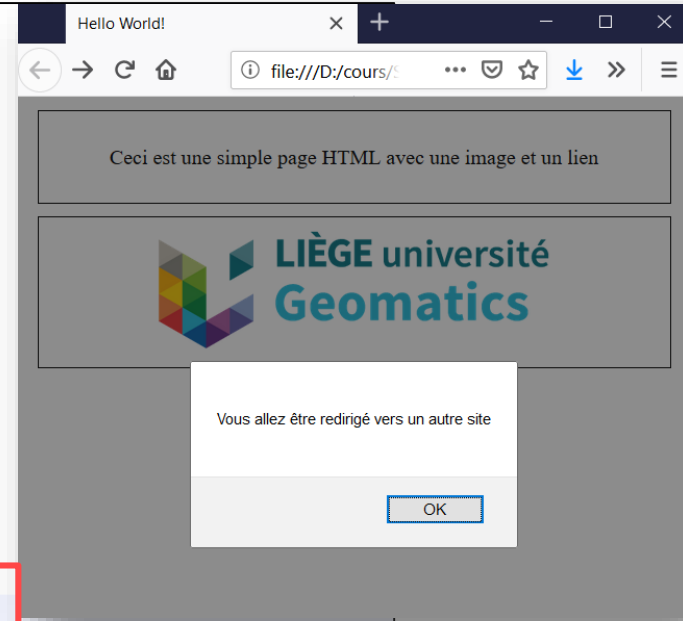


```
<html>
  <head>
    <title>
      Hello World!
    </title>
    <style type="text/css">
      .mon-style{
        text-align:center;
        border-style: solid;
        border-width: 1px;
        margin: 10px;
        padding:10px
      }
      .image{
        width: 300px;
      }
    </style>
  </head>
  <body>
    <div class="mon-style">
      <p>
        Ceci est une simple page HTML avec une image et un lien
      </p>
    </div>
    <div class="mon-style">
      <a href="http://geomatics.ulg.ac.be">
        </img>
      </a>
    </div>
  </body>
</html>
```



*Exemple de page HTML + CSS*

```
<html>
  <head>
    <title>
      Hello World!
    </title>
    <style type="text/css">
      .mon-style{
        text-align:center;
        border-style: solid;
        border-width: 1px;
        margin: 10px;
        padding:10px
      }
      .image{
        width: 300px;
      }
    </style>
    <script>
      function message() {
        alert("Vous allez être redirigé vers un autre site");
      }
    </script>
  </head>
  <body>
    <div class="mon-style">
      <p>
        Ceci est une simple page HTML avec une image et un lien
      </p>
    </div>
    <div class="mon-style">
      <a onClick="message()" href="http://geomatics.ulg.ac.be">
        </img>
      </a>
    </div>
  </body>
</html>
```



*Exemple de page HTML + CSS + Javascript*

**– Serveur web**

- » Permet une communication avec un client (navigateur ou autre) à travers le protocole **HTTP** (ou HTTPS)
- » **Principe:**
  - Le **client** envoie une **requête HTTP** vers le serveur comprenant notamment:
    - L'adresse du serveur (nom de domaine ou adresse IP) et le port de communication (généralement 80)
    - Des paramètres
      - Exemple: nom d'utilisateur et mot de passe pour se connecter à une page web privée
  - Le serveur répond en envoyant une **page HTML**
    - La page est construite dynamiquement par le serveur
    - Le navigateur interprète son contenu statique (html + css) ou dynamique (html + css + javascript)
- » Le serveur web communique généralement avec une **base de données** à travers le protocole **TCP/IP**
  - Transmission de requêtes SQL et réception des résultats

- » Un serveur web peut être programmé via plusieurs **langages** et **outils**
  - Exemples
    - **PHP** + **Apache**
    - **Java** + **Tomcat**
    - **Javascript** + **NodeJS**
    - **Python** + **Zope**
- » Il n'est pas rare qu'un SIG exploite **plusieurs serveurs web** différents
  - Exemple:
    - PHP pour la gestion des pages web et Java pour la gestion des cartes via le protocole WMS (voir plus loin dans le cours)
  - Ces serveurs web peuvent être hébergés sur la même machine et communiquer avec des ports différents (ex: 80 et 8080)



```
<?php
$dbname="database";
$user="postgres";
$password="postgres";
$host="localhost";
$port="5432";

try
{
    $bdd = new PDO("pgsql:host=$host; port=$port;dbname=$dbname", $user, $password,
    array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
}
catch (Exception $e)
{
    die('Erreur : ' . $e->getMessage());
}

$query="SELECT id, measure FROM MyTable";

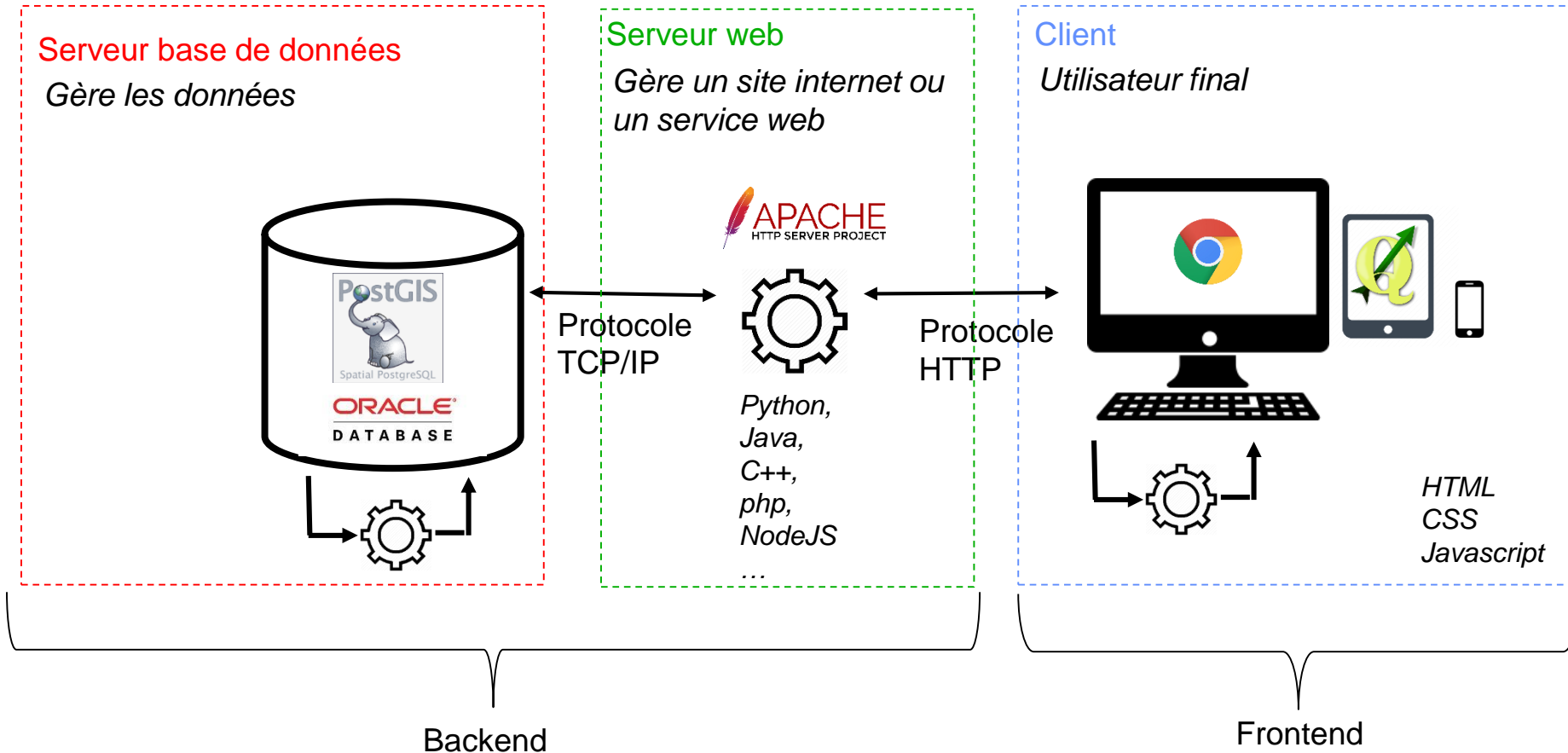
$array_result=array();

$req=$bdd->query($query) or die();

while($result=$req->fetch(PDO::FETCH_BOTH)) {
    $measure=$result['measure'];
    $id=$result['id'];
    $array_result["$id"]=$measure;
}
?>
```

*Exemple de connexion à une base de données et requête SQL dans un script php*

– Architecture 3 tiers



**– Service web**

- » Selon ISO 19119, un **service** est une fonctionnalité fournie par une entité à travers une interface.
- » Un **service Web** est une application qui accepte et traite les requêtes d'une autre application transmises à travers le réseau Internet.
  - Ex: communication entre plusieurs serveurs web
- » Un service Web est aussi décrit comme un **échange de messages** entre le service et ses clients (machines)
  - Un service Web **n'est pas** un site Web ni un portail, qui eux sont directement exploités par des humains et fournissent toujours une visualisation de l'information (textuelle ou image).
- » Les services Web reposent sur des **protocoles Web standardisés** pour envoyer et recevoir de l'information via Internet.
  - Toutes les requêtes qui sont envoyées à un service Web sont généralement formatées
    - en **JSON** (JavaScript Object Notation)
    - ou **XML** (langage à balises à la base du HTML)

- **3. Web Mapping**

- **Généralités**

- » Un des rôles de l'interface client du SIG web est l'affichage d'information (**cartes interactives**, tableaux, ...) construite à partir des **données spatiales** de la BD
    - » Exemple
      - Données = polygones avec attribut sur la densité de population
      - Information = carte choroplèthe de densité de population
    - » Il existe **2 stratégies** pour construire une **carte interactive** sur le web interprétable par un navigateur (→ symbolisation, classification, généralisation, etc)
      - Construction par le client à partir d'un format de données standardisé transmis par le serveur
        - **GeoJSON**, GML, WKT, ...
      - Construction par le serveur et transmission d'une image géoréférencée au client en suivant un protocole standard
        - « Web Map Service » (**WMS**)

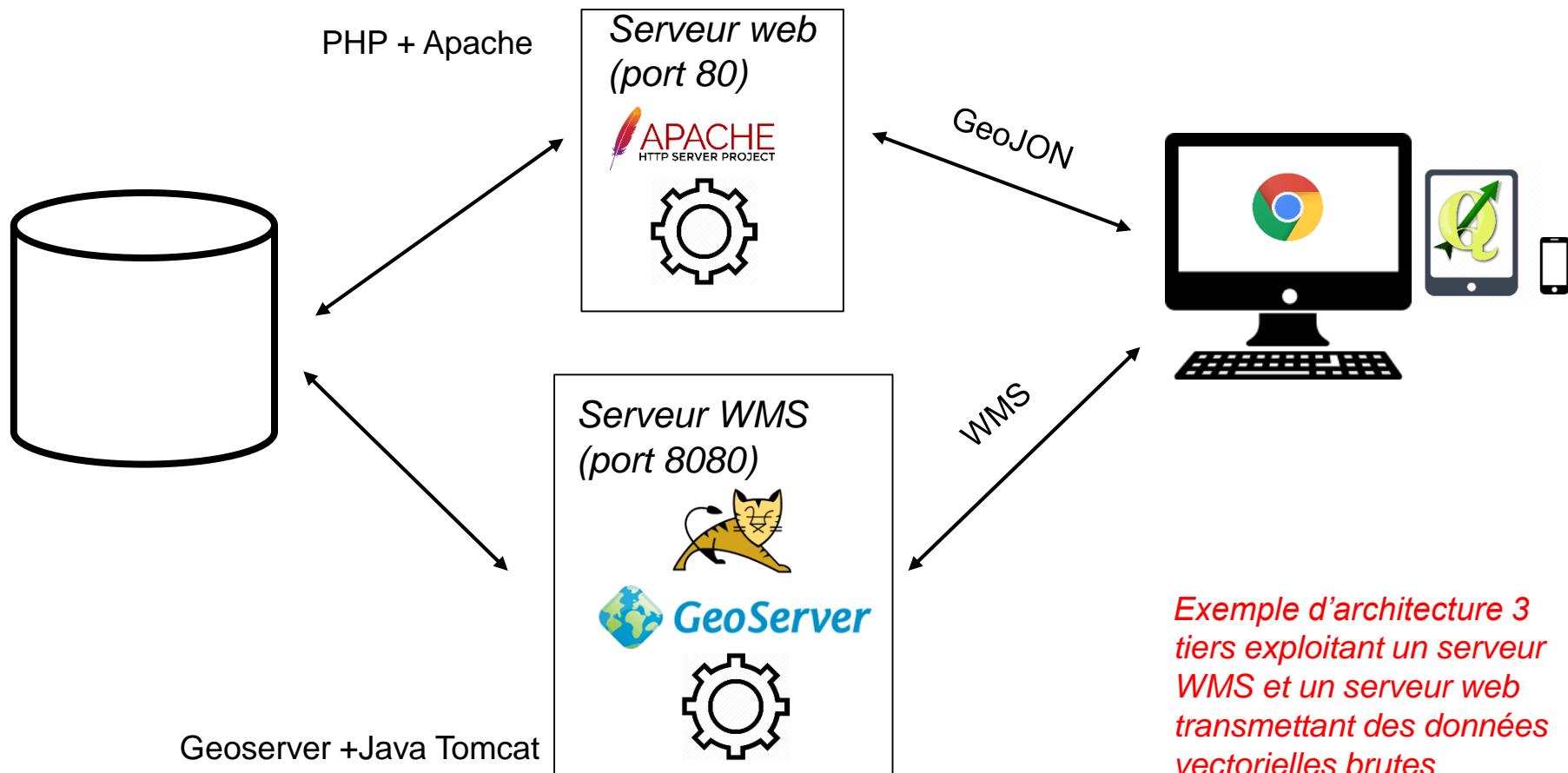
## – Transmission des données vectorielles brutes au client

- » Les données sont manipulées par le client pour en faire une carte dynamique
  - Le format de **données** (ex: GeoJSON) transmis par le serveur (HTTP) doit être **interprété par le code javascript**
  - Exploitation de bibliothèques Javascript capables de construire une carte à partir des données vectorielles (OpenLayers, Leaflet, ...)
- » **Avantages:**
  - Opérations de navigation cartographique très **fluides** puisque tout est géré par le client
  - D'**autres traitements** peuvent être appliqués par le client sur les données spatiales en exploitant d'autres bibliothèques spatiales (ex: calcul d'un « buffer » avec la bibliothèque Turf.js) ou non-spatiales (chart.js pour la construction de graphiques)
  - **Allègement** de la charge de travail du **serveur**
- » **Inconvénients:**
  - **Données vecteur** uniquement
  - **Fonctionne mal** pour les cartes impliquant **beaucoup de données** (plusieurs milliers d'entités)
  - Protocole d'échange pas complètement standard → **manque d'interopérabilité**

## – Transmission d'un « Web Map Service » (WMS) au client

- » Les données sont manipulées par le serveur pour les transformer en une **image géoréférencée** représentant une **carte** (+ légende)
- » Information (carte) transmise par le serveur comme un **service web standardisé**: « Web Map Service » (**WMS**) via le protocole **HTTP**
- » Affichage de la couche WMS dans le navigateur grâce à une **bibliothèque javascript compatible** (OpenLayers, Leaflet, ...)
- » **Avantages:**
  - **Fonctionne bien** même lorsque la carte implique le traitement de **beaucoup de données** (une simple image est transmise au client)
  - **Interopérabilité**: les couches WMS peuvent être affichées dans n'importe quel outil client compatible, notamment SIG-logiciel (QGIS, ...)
  - Fonctionne pour les données **vecteur** et **raster**
- » **Inconvénients:**
  - Opérations cartographiques **manquent de fluidité** (échanges d'information avec le serveur à chaque opération)
  - **Pas** d'opérations possibles sur les **données brutes** par le client
  - Plus de **travail** pour le **serveur**
- » **Note**: WMS est un service web spatial parmi d'autres (WCS, WFS, WPS, ...)

- » **Outils** pour pour l'impémentation d'un serveur WMS
  - GeoServer (java)
  - MapServer (C++)

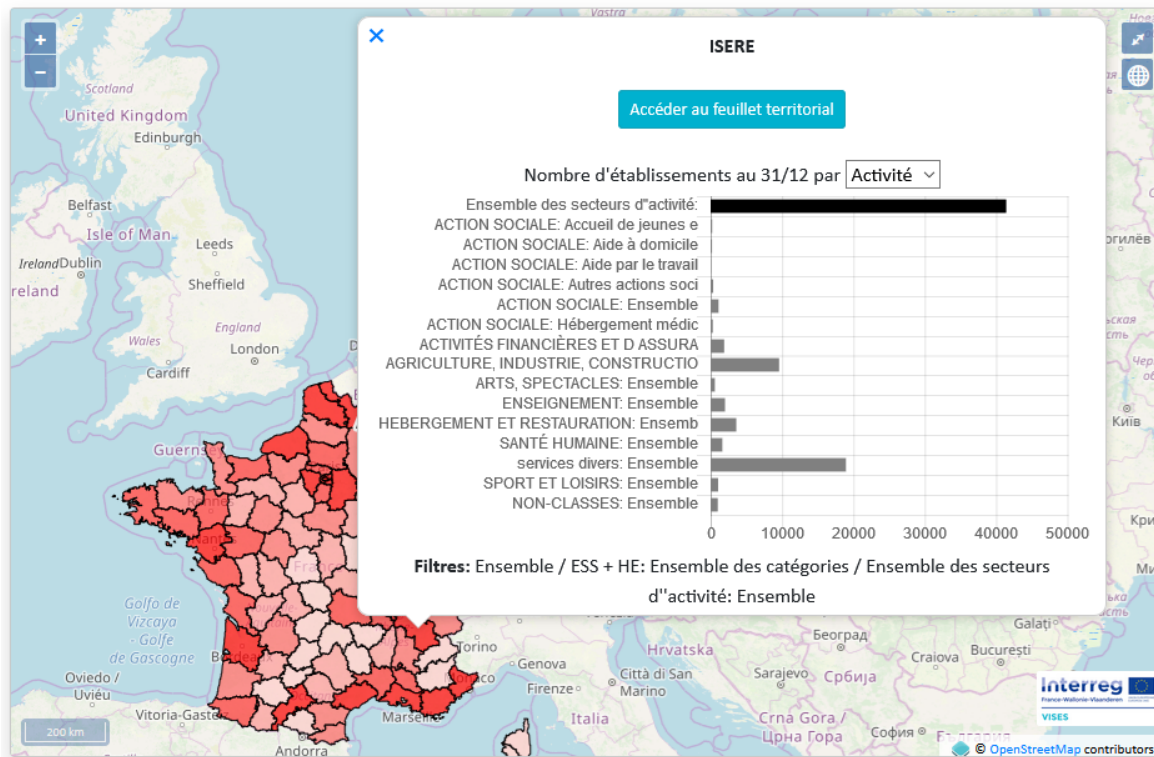


*Exemple d'architecture 3 tiers exploitant un serveur WMS et un serveur web transmettant des données vectorielles brutes*



• 4. Démonstration SIG Web (<http://racines.projetvisesproject.eu>)

Année:  Territoire:  Jeux de données:  Départements français:



Cliquez sur la carte ou utilisez les menus ci-dessous pour plus d'information

Indicateur:

Taille:

Famille:

Activité:

Classification cartographique:

