



## 3-1 Présentation des services web

- 1- Rappel de l'architecture applicative globale et objectif de cette étape
- 2- Rappel de l'architecture orientée services
- 3- Les 2 types de services : Extraction de données et Mise à jour de la base
- 4- Inventaire des services web (fournis et restant à développer)
- 5- Importation des fichiers fournis
- 6- Organisation des tests

### 1- Rappel de l'architecture applicative globale et objectif de cette étape

La page suivante présente l'architecture générale de l'application **TraceGPS**.

Cette étape porte sur le **développement et le test des services web** qui seront appelés par l'application Android pour accéder à la base de données au moyen d'une connexion Internet.

Ces services web forment une API web de type REST.

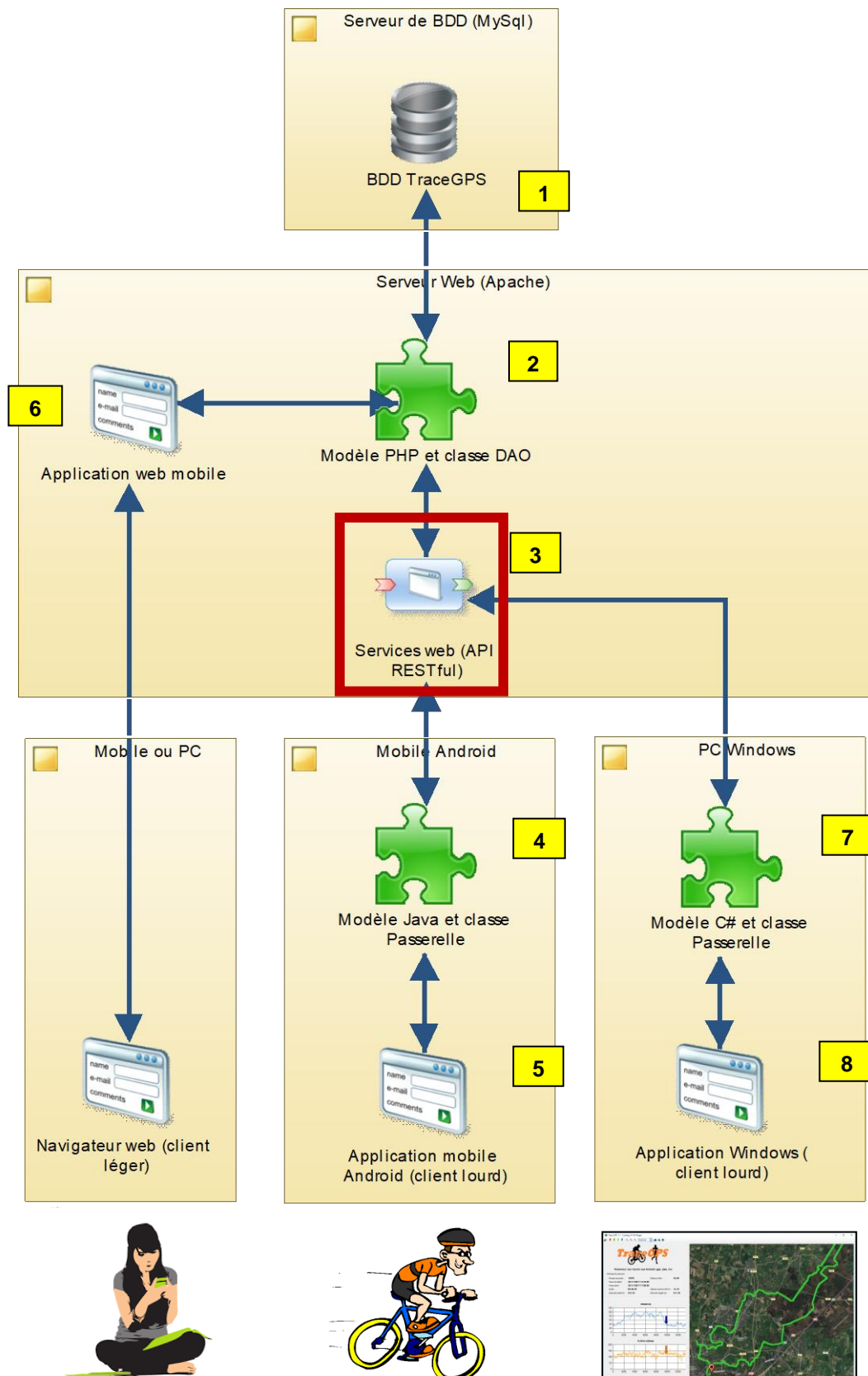
Les services web accèdent à la base de données en utilisant le **modèle** créé à l'étape précédente, qui comprend :

Les 4 classes métiers :

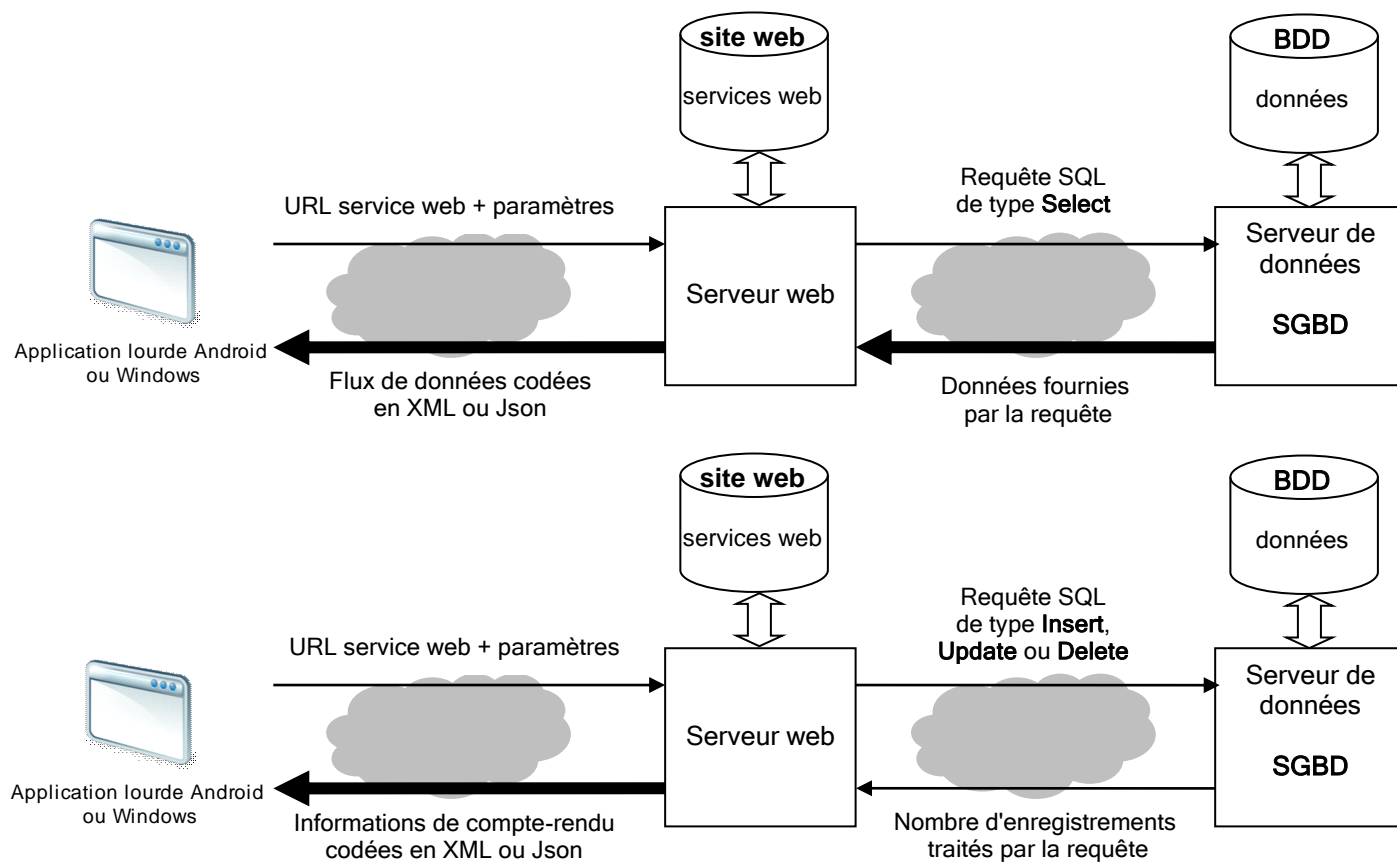
- **Point** : représente un point géographique
- **PointDeTrace** : (hérite de **Point**) ; représente un point de passage lors d'un parcours
- **Trace** : représente une trace (ou un parcours) réalisé par un utilisateur
- **Utilisateur** : représente un utilisateur

Les 2 classes techniques :

- **DAO** (Data Access Object) : regroupe tous les accès à la base de données
- **Outils** : regroupe des fonctions statiques utilisées couramment



## 2- Rappel de l'architecture orientée services



Les services web développés dans cette partie permettront à l'application mobile Android et à l'application Windows de consulter et de mettre à jour la base de données **tracegps**.

Les données retournées seront au format **XML** ou **Json**, le choix du langage se faisant à l'aide d'un paramètre **lang** figurant dans l'URL de chaque service.

### 3- Les 2 types de services : Extraction de données et Mise à jour de la base

Les services que nous allons développer sont déduits des cas d'utilisation présentés dans le dossier des spécifications de l'application **TraceGPS**.

Ils peuvent être classés en 2 catégories :

- des services chargés d'extraire des données et de les retourner ; ils font appel à une ou plusieurs requêtes SQL de type **Select** et ne modifient pas le contenu de la base de données ; leur nom commence souvent par **Get** :
  - **Connecter**
  - **GetTousLesUtilisateurs**
  - **GetLesUtilisateursQueJautorise**
  - **GetLesUtilisateursQuiMautorisent**
  - **GetUnParcoursEtSesPoints**
  - **GetLesParcoursDunUtilisateur**
- des services chargés de mettre à jour la base de données ; ils font appel à une ou plusieurs requêtes SQL de type **Insert**, **Update** ou **Delete** :
  - **CreerUnUtilisateur**
  - **ChangerDeMdp**
  - **SupprimerUnUtilisateur**
  - **DemanderMdp**
  - **DemanderUneAutorisation**
  - **ValiderDemandeAutorisation**
  - **RetirerUneAutorisation**
  - **EnvoyerPosition**
  - **DemarrerEnregistrementParcours**
  - **ArreterEnregistrementParcours**
  - **SupprimerUnParcours**

## 4- Inventaire des services web (fournis et restant à développer)

### 4-1 Services web fournis :

#### Gestion des utilisateurs

Connecter  
GetTousLesUtilisateurs  
CreerUnUtilisateur  
ChangerDeMdp  
SupprimerUnUtilisateur

#### Gestion des autorisations

ValiderDemandeAutorisation

### 4-2 Services web restant à développer :

#### Gestion des utilisateurs

DemanderMdp

#### Gestion des autorisations

GetLesUtilisateursQueJautorise  
GetLesUtilisateursQuiMautorisent  
DemanderUneAutorisation  
RetirerUneAutorisation

#### Gestion des points de traces

EnvoyerPosition

#### Gestion des traces

GetUnParcoursEtSesPoints  
GetLesParcoursDunUtilisateur  
SupprimerUnParcours  
DemarrerEnregistrementParcours  
ArreterEnregistrementParcours

## 5- Importation des fichiers fournis

Ouvrez **Eclipse** avec le workspace PHP utilisé pour le projet **tracegps** développé en PPE.

Créez un dossier nommé **api** dans votre projet avec la commande **File / New / Folder**.

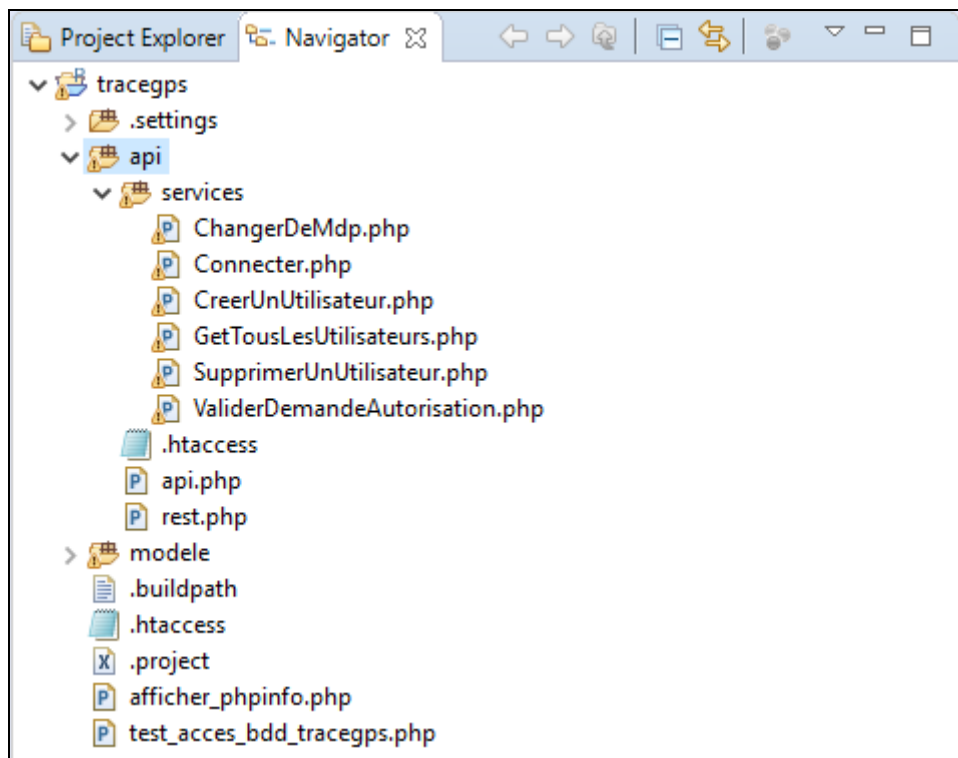
Dans ce dossier **api**, importez les 3 fichiers fournis (**api.php**, **rest.php** et **.htaccess**) avec la commande **File / Import... / General / File System**.

Dans le dossier **api**, créez un sous-dossier nommé **services** avec la commande **File / New / Folder**.

Dans ce dossier **services**, importez les 6 services fournis avec la commande **File / Import... / General / File System**.

La vue **Project Explorer** ne montre pas le fichier **.htaccess**, il faut utiliser la vue **Navigator**. Si cette vue n'est pas affichée, utilisez la commande **Window / Show View / Other... / General / Navigator**.

La structure du projet doit ressembler à :



## 6- Organisation des tests

Pour tester un service web, il suffit de l'appeler dans un navigateur web, en ajoutant les paramètres nécessaires dans l'URL.

Chaque service web sera testé avec toutes les situations possibles (paramètres incomplets ou incorrects, paramètres corrects, action demandée impossible à réaliser, ...).

L'ensemble des situations (ou exceptions) de chaque cas d'utilisation est présentée dans le dossier des spécifications de l'application **TraceGPS**.

Pour permettre de réaliser des tests complets et rigoureux, une fiche de test a été créée pour chaque service web.