





AP

2-1 Présentation du modèle PHP

1- Rappel de l'architecture applicative globale et objectif de cette étape
2- Organisation du dossier modele dans le projet
3- Diagramme des classes

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 en PHP et le test du **modèle objet côté serveur**.

Ce modèle objet se compose de 4 classes métiers, de 2 classes techniques et d'un fichier contenant les paramètres de l'application :

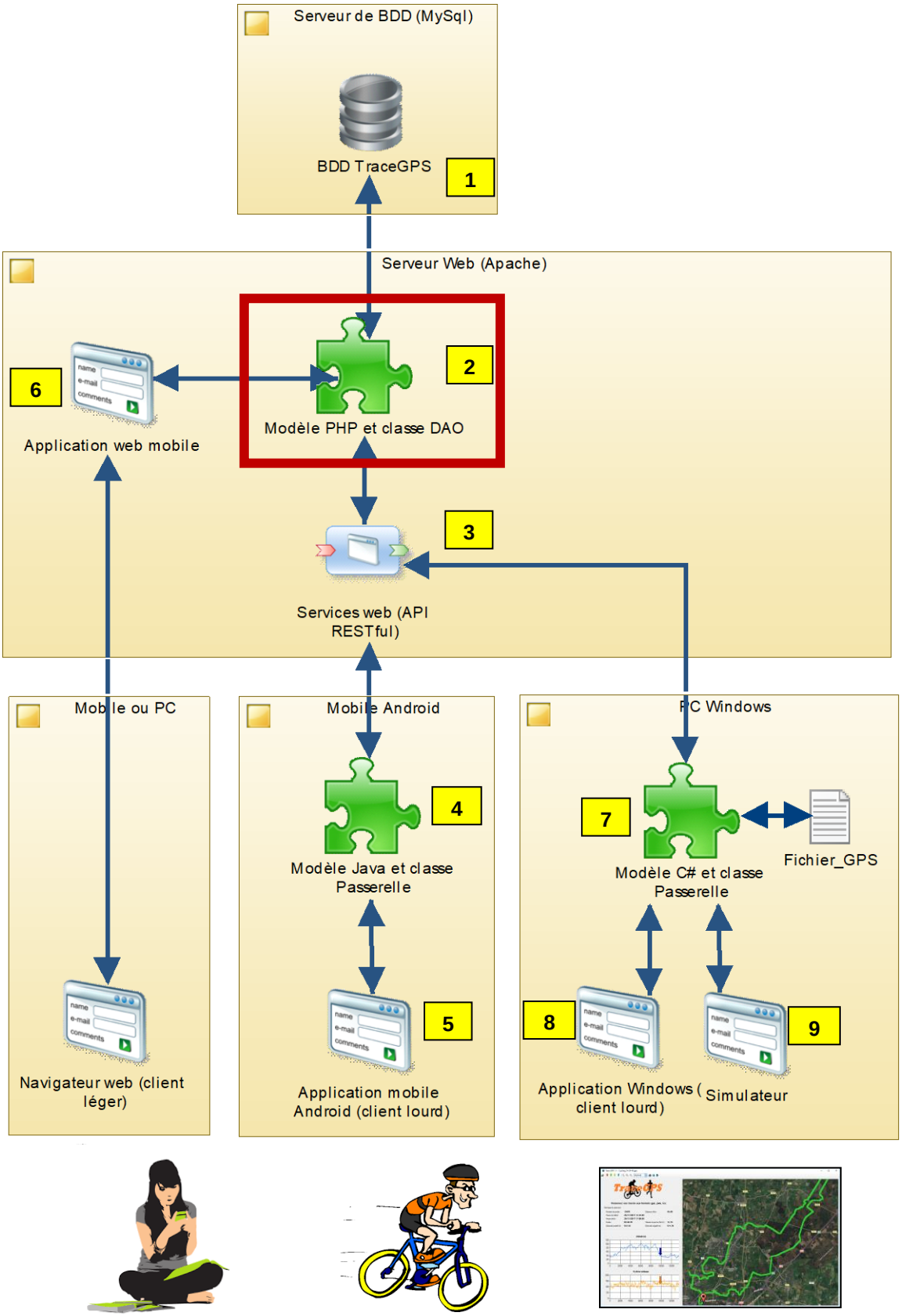
Les 4 **classes métiers** :

- **Point** : représente un point géographique quelconque
- **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

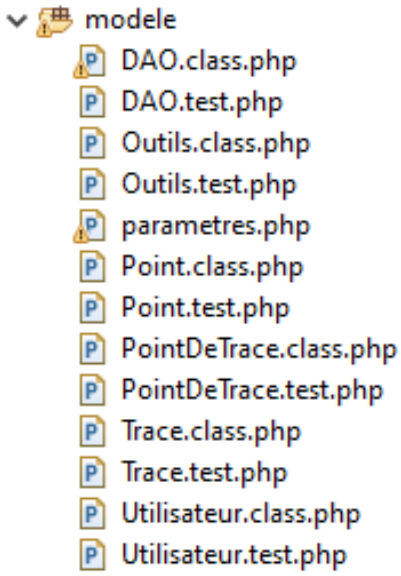
Les 2 **classes techniques** :

- **DAO** (Data Access Object) : classe gérant tous les accès à la base de données (Select, Insert, Update et Delete) ; c'est la seule classe exécutant des requêtes SQL
- **Outils** : boîte à outils offrant différents services sous forme de méthodes statiques (contrôle de données, remise en forme de données, création de mot de passe aléatoire, envoi de mail, ..)

Le fichier **parametres.php** : il contient les paramètres de l'application, notamment les paramètres de connexion au SGBD et à la base de données.


Serveur de BDD (MySql)
BDD TraceGPS
1
Serveur Web (Apache)
2
Modèle PHP et classe DAO
6
Application web mobile
3
Services web (API RESTful)
Mobile ou PC
Navigateur web (client léger)
Mobile Android
4
Modèle Java et classe Passerelle
5
Application mobile Android (client lourd)
PC Windows
7
Modèle C# et classe Passerelle
Fichier_GPS
8
Application Windows (client lourd)
9
Simulateur

2- Organisation du dossier modele dans le projet



A chaque **classe XXXX** correspondent 2 fichiers :

- le fichier **XXXX.class.php** contient le code de la classe (attributs privés, accesseurs publics get et set, méthodes publiques)
- le fichier **XXXX.test.php** contient le code des tests fonctionnels de la classe ; cette page web est destinée à être appelée directement dans un navigateur web.

Comme les langages C# et Java, le langage PHP dispose d'un outil de gestion de tests unitaires nommé **PHPUnit**.

Nous ne l'utiliserons pas ici (nous referons des tests unitaires bientôt en Java avec JUnit).
A la place, nous effectuerons des **tests visuels** avec une page web qui sera appelée avec un navigateur web.

3- Diagramme des classes PHP