2-5 Travail collaboratif avec la forge logicielle GitHub

1- Le travail collaboratif
2- Les logiciels de gestion de versions
3- Les forges logicielles
4- Les 4 documents d'apprentissage pour l'AP
5- Schéma récapitulatif

1- Le travail collaboratif

Les développeurs travaillent en général en équipe sur un projet commun dont le code est stocké sur un "**dépôt distant**" ou "**remote repository**".

Ils doivent tous disposer du code du projet sur leur poste de travail, dans un "**dépôt local**" ou "**local repository**".

Quand un développeur a modifié (et testé positivement) un élément de code, il doit pouvoir l'intégrer au code commun et le rendre accessible à ses collaborateurs.

Il arrive parfois que la modification d'un élément entraine des dysfonctionnements sur d'autres éléments : il y a alors une "**régression de code**". Dans ce cas, il est important de pouvoir retourner aux versions précédentes. L'équipe de développement doit donc pouvoir retrouver l'historique complet de l'évolution du projet.

Il arrive également que 2 développeurs modifient le même fichier source, ce qui crée un conflit. L'équipe doit également disposer de systèmes d'aide au règlement de ces conflits.

2- Les logiciels de gestion de versions

Les logiciels de **gestion de version** (ou de "**versionning**") permettent à une équipe de partager les codes sources de leur projet et offrent de nombreuses fonctionnalités, dont :

- la mémorisation de toutes les modifications effectuées
- la résolution des conflits

Celui que nous utiliserons en AP s'appelle **Git** (Global Information Tracker).

Pour fonctionner avec le logiciel de gestion de versions Git, l'EDI Eclipse dispose du plugin **EGit**.

Nous aurions pu installer Git sur un serveur local (et donc protégé de l'extérieur) mais la solution choisie en AP consiste à utiliser la forge logicielle publique **GitHub** qui repose sur Git.

3- Les forges logicielles

L'objectif d'une **forge logicielle** est de permettre à plusieurs développeurs de participer ensemble au développement d'un ou plusieurs logiciels, le plus souvent à travers le réseau Internet.

Les outils offerts par une forge logicielle sont principalement :

- un ou plusieurs système(s) de gestion des versions (SVN, Git, Mercurial, ...)
- un gestionnaire de listes de discussion (et/ou de forums)
- un outil de suivi des tickets d'incidents
- un gestionnaire de documentation (souvent sur le principe du wiki)
- un gestionnaire de tâches
- un outil de traduction en ligne

Nous utiliserons la plateforme **GitHub** qui utilise **Git** comme logiciel de gestion de versions (comme son nom l'indique).

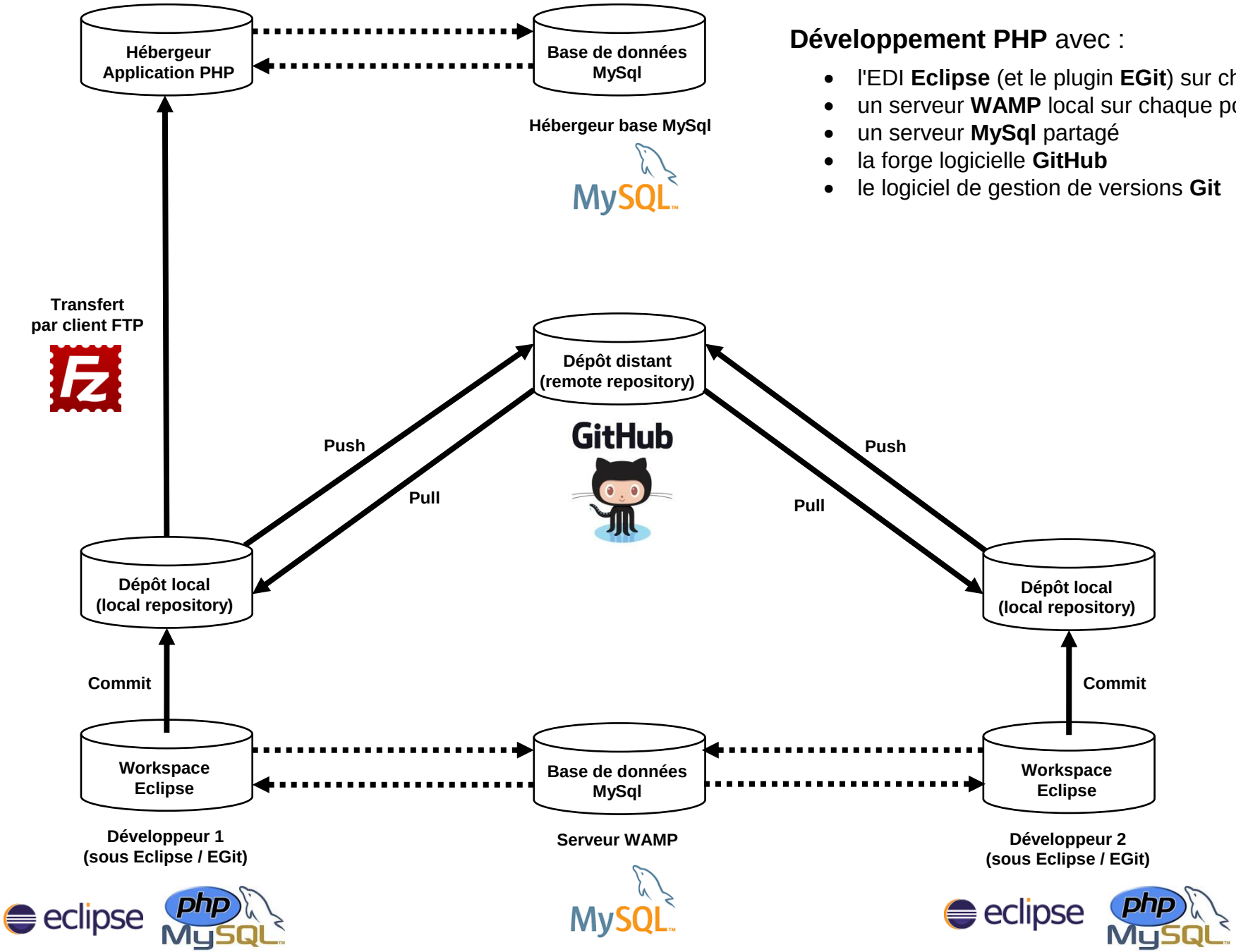
Avec un compte gratuit, les codes déposés sur GitHub sont publics et peuvent être consultés par n'importe qui. Mais seuls les collaborateurs autorisés pourront modifier le code. GitHub propose bien sûr des comptes payants permettant d'empêcher les personnes non autorisées de consulter un projet.

4- Les 4 documents d'apprentissage pour l'AP

- Le document ***2-5-1 - Inscription à la forge logicielle GitHub*** indique comment se créer un compte sur la plateforme GitHub (concerne aussi bien le chef de projet que ses collaborateurs)
- Le document ***2-5-2 - Initialisation d'un projet partagé avec Eclipse-EGit*** explique comment le chef de projet initialise le projet sur son poste de travail et sur la plateforme GitHub et comment il désigne les collaborateurs autorisés à participer au projet
- Le document ***2-5-3 - Participation à un projet partagé avec Eclipse-EGit*** explique comment les collaborateurs autorisés par le chef de projet vont créer une copie du projet sur leur poste de travail à partir du dépôt distant
- Le document ***2-5-4 - Opérations courantes avec Eclipse-EGit*** explique comment les développeurs utilisent GitHub au quotidien, et notamment lors des conflits sur les fichiers

5- Schéma récapitulatif (page suivante)

- 2 développeurs utilisent l'EDI **Eclipse** avec le plugin **EGit**. Leur projet est stocké sur leur **workspace local**.
- Ils disposent tous les 2 d'un serveur WAMP local et d'une base de données locale MySql (leurs bases de données peuvent donc avoir des contenus différents).
- S'ils le souhaitent, ils peuvent effectuer leurs tests avec une base de données partagée sur un serveur WAMP commun.
- Quand un développeur a terminé et testé la modification d'un élément, il valide cette modification en effectuant un **Commit**, puis un **Push** (pousser) qui copie la nouvelle version vers le dépôt central (donc sur GitHub).
- Quand un développeur veut récupérer les modifications effectuées par d'autres développeurs, il effectue un **Pull** (tirer).
- Quand une version du projet peut être déployée sur un hébergeur Internet, on utilise un logiciel **client FTP** tel que **FileZilla**.
- La version hébergée du site doit bien sûr accéder à la base de données hébergée sur Internet.



Développement PHP avec :

- l'EDI **Eclipse** (et le plugin **EGit**) sur chaque poste
- un serveur **WAMP** local sur chaque poste
- un serveur **MySql** partagé
- la forge logicielle **GitHub**
- le logiciel de gestion de versions **Git**