

## C36 Développer dans le cadre d'une architecture client-serveur

### Le candidat pourra associer ou non comme compétences satellites C35 ou C38

Le candidat est invité à rechercher une illustration, tirée de son activité, de chaque aspect technique de la compétence correspondant présenté dans cette fiche. Cette illustration pourra être mentionnée dans le dossier, et sera présentée lors de l'épreuve.

La présente fiche se présente sous la forme d'une énumération, elle n'est ni un format ni un plan pour l'épreuve ou le dossier d'activité.

Les termes en gras appartiennent au vocabulaire technique de référence de la compétence, ils doivent donc être naturellement employés par le candidat lors de l'épreuve

**Architecture Client/Serveur** : Il y a coopération entre des processus logiciels : l'un appelé client (demandeur) et l'autre serveur (traite la demande et fournit une réponse) qui communiquent au moyen de **requêtes** et de **réponses**

**Serveur** : Tout processus qui reçoit une requête (**demande de service**) venant d'un client, via un réseau, traite cette demande et renvoie la réponse au client.

**Middleware** : Communication entre applications mettant en œuvre des formats de données différents (Exemple : Créer une connexion ODBC et l'appeler dans le code php pour pouvoir communiquer avec la base de données). La mise en œuvre d'un middleware n'est pas nécessaire si le format des données est **natif** au sein de l'environnement de développement.

**Schéma architecture Client/Serveur** : Ce schéma permet d'explicitier les rôles des différents composants logiciels (données et/ou traitements) de l'architecture client-serveur mise en œuvre dans une application

#### 1. Client serveur web

HTML : Afficher des pages **statiques** : les pages seront toujours les mêmes. **Feuilles de style** pour la présentation.

PHP : S'exécute côté serveur. Code défini entre les balises `<?php` et `?>`.

Avoir un site web **dynamique** : application Web qui dialogue avec une base de données. Selon la demande de l'utilisateur et le contenu de la base de données, les pages affichées ne seront pas toujours les mêmes.

Requête de sélection : Rend un seul enregistrement ou un jeu d'enregistrements à parcourir dans le php, qui prépare alors dynamiquement la page à envoyer au client.

Enregistrement de données dans la base : **Formulaire** de saisie que le client renseignera. Vérifications puis requête action INSERT. `$_GET["champ"]` ou `$_POST["champ"]` permet de récupérer les valeurs saisies dans "champ".

Javascript : S'exécute du côté client. Pour limiter le trafic réseau et optimiser le temps de réponse. Permet les vérifications de saisie ou des actions qui ne nécessitent pas l'accès aux données.

#### 2. Client serveur de base de données

Une application cliente accède aux données stockées sur un **serveur de base de données**.

La mise en œuvre de **déclencheurs** sur certaines tables de la base permet de déporter certains traitements sur les données (requêtes) consécutifs à des opérations d'ajout, modification ou suppression.

Les déclencheurs sont automatiques : ils sont exécutés immédiatement après n'importe quelle manipulation des données, que ce soit une action manuelle directe sur une table, ou une action programmée dans une application.

Les déclencheurs permettent de mettre en place des restrictions sur des valeurs de champs plus complexes que celles prévues par la clause check constraints. En particulier, contrairement à la clause check constraints, les déclencheurs peuvent référencer plusieurs colonnes de diverses tables.

#### 3. Enrichissements possibles

Web :

- CSS
- XHTML

Bases de données :

- Réplication
- Répartition des données
- Synchronisation
- Procédures stockées
- Tâches planifiées, sauvegarde
- Etc.