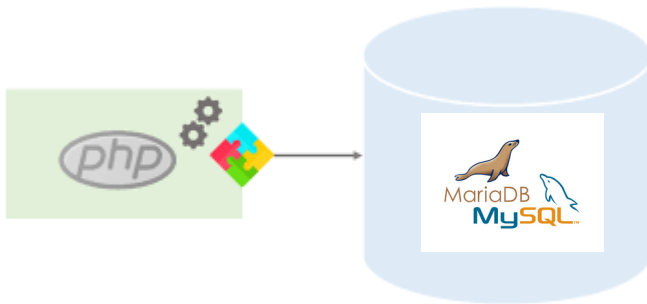




# *PHP & MariaDB / MySQL*



# Présentation PHP avec BDD

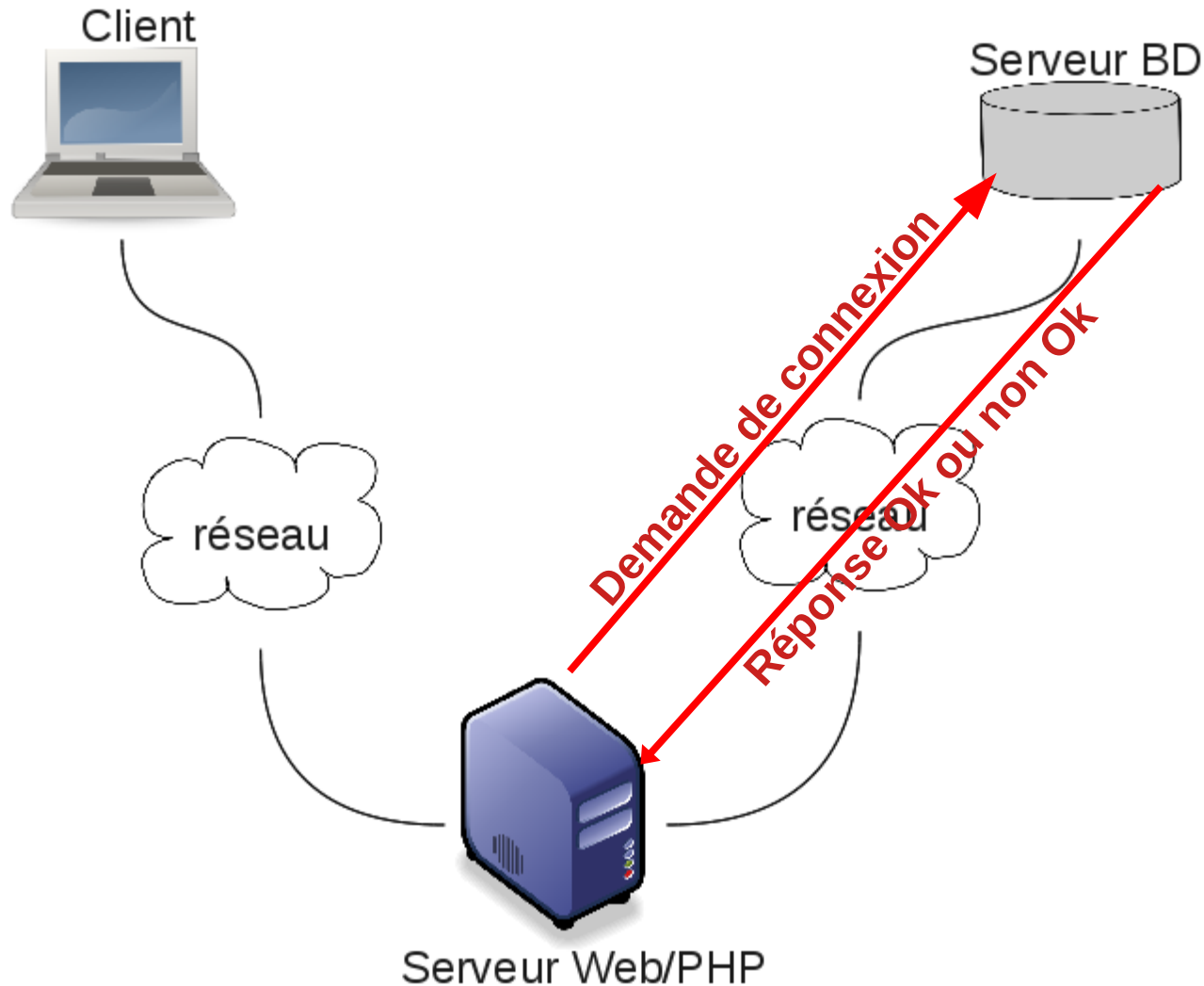
- Principes
- PHP Data Objects (PDO)
- Conclusion

# Principes

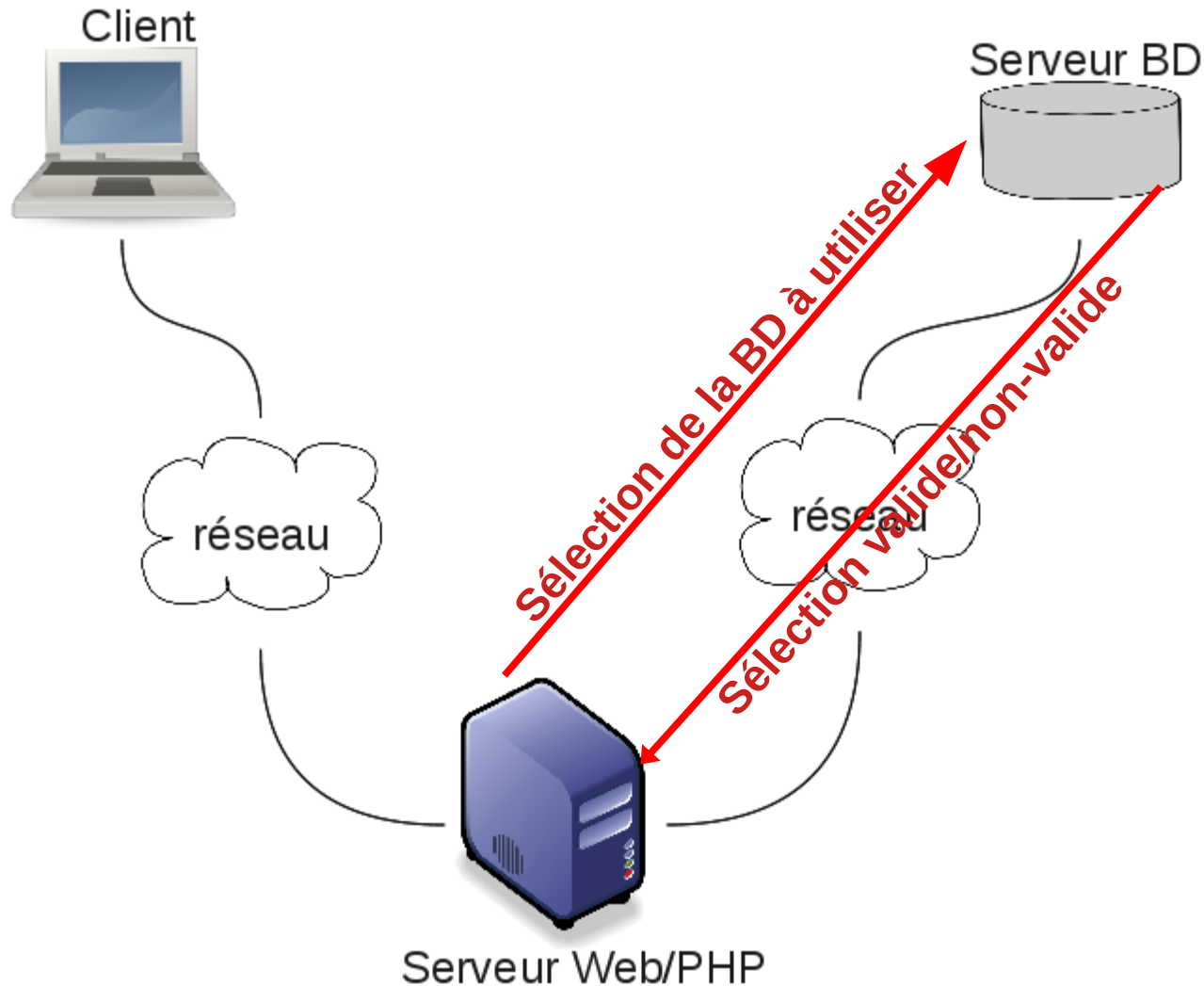
4 phases :

- 1) Connexion au serveur de base de données
- 2) Sélection de la base de données
- 3) Requêtes
- 4) Libération des ressources de la requête

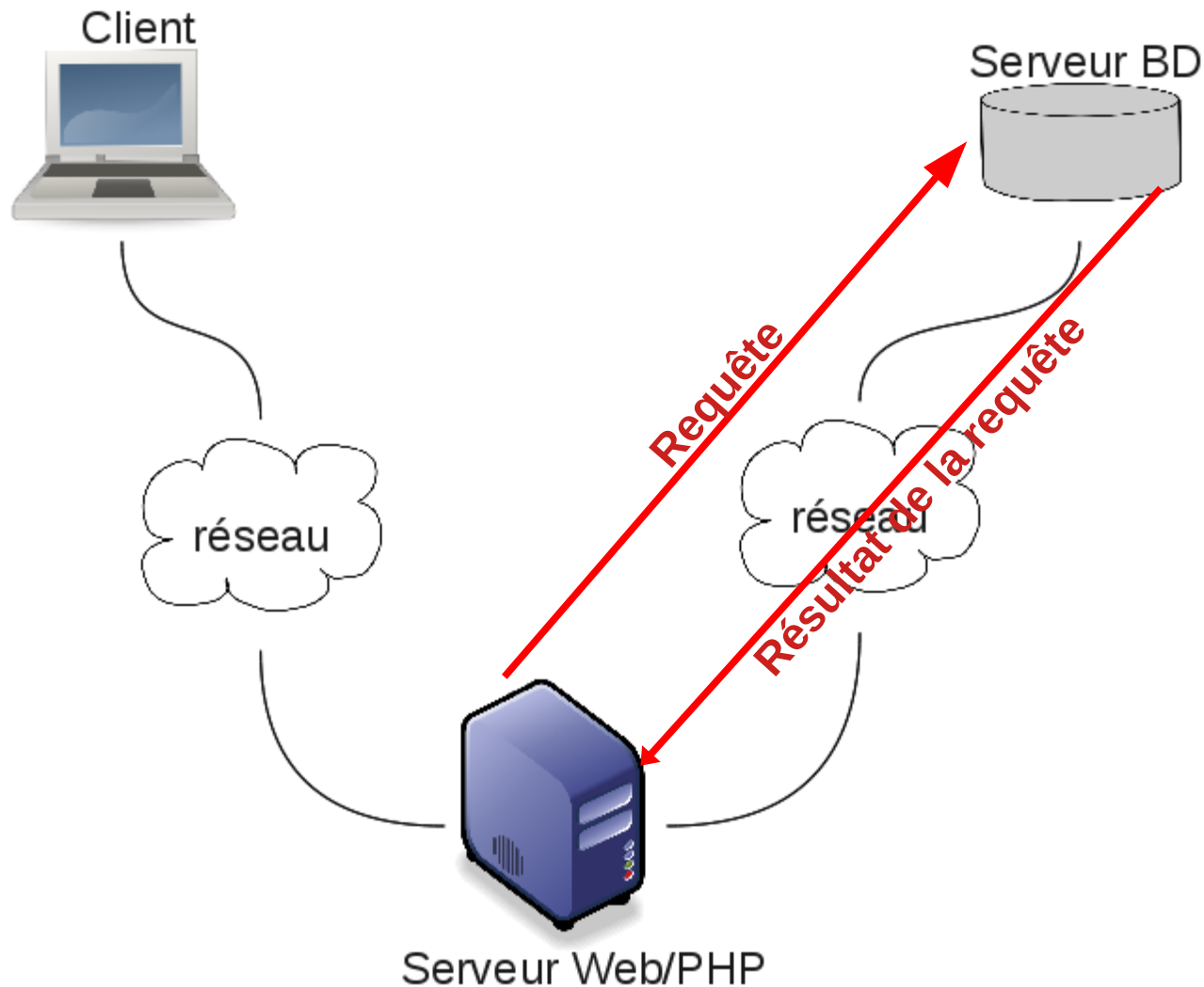
# Principe : Connexion au serveur



# Principe : Sélection de la Bdd



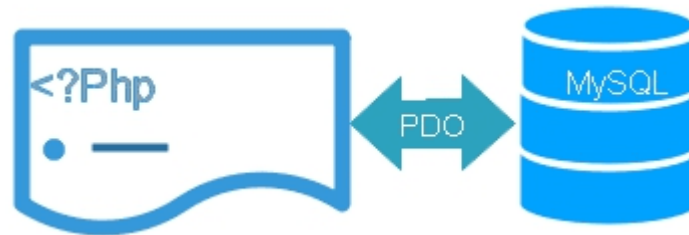
# Principe : Requêtes



# Extension PDO



- PHP Data Objects, API introduit avec PHP 5.1



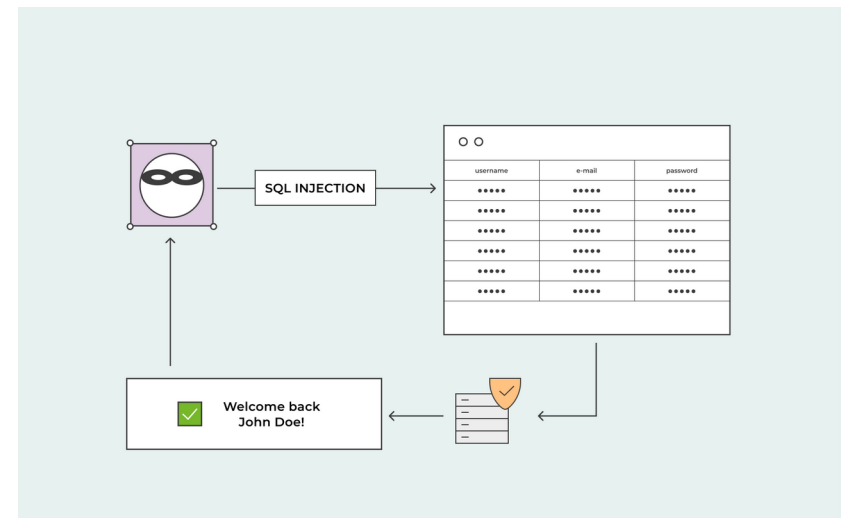
- Avantage majeur de proposer une API qui permet d'interroger plusieurs moteurs de bases de données différents, sans avoir à modifier le code PHP.



# Plus de sécurité avec PDO



- PDO facilite l'utilisation des instructions préparées, essentiel pour protéger l'application contre les injections SQL.
- En utilisant des instructions préparées, les données sont séparées des instructions SQL.
- Empêche les attaquants d'injecter du code SQL malveillant.



<https://www.one.com/fr/securite-de-site-web/injection-sql-definition>



# Les bonnes pratiques avec PDO

Définition des constantes nécessaires à la connexion dans un fichier **config.inc.php** :

```
<?php
// config.inc.php
define("SERVEUR_BDD", "adresseServeurSQL");
define("LOGIN", "login");
define("MOT_DE_PASSE", "motDePasse");
define("NOM_DE_LA_BASE", "nomDeLaBase");
```

# Connexion à la base de données

Le traitement d'erreur est réalisé par des exceptions.

Cette pratique permet d'avoir un code davantage portable et plus facile à maintenir.

```
<?php
require_once __DIR__.'./chemin/vers/config.inc.php'; // selon le chemin du fichier

try {
    $dsn = 'mysql:host=' . SERVEUR_BDD . ';dbname=' . NOM_DE_LA_BASE;
    $pdo = new PDO($dsn, LOGIN, MOT_DE_PASSE);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $pdo->exec("set names utf8");
    ....
}
catch (PDOException $e) {
    print "La connexion a échoué :". $e->getMessage() . "</br>";
    die();
}
```

# Requêtes sans paramètres

## Syntaxe

// execution de la requête

```
$requete = $pdo ->query("la requête SQL à effectuer ;") ;
```

## Exemple

```
$requete = $pdo ->query("select * from adherents ;") ;
```

# Requêtes préparées : 3 étapes

- Préparation de la requête

```
$requete = $pdo->prepare("la requête SQL à effectuer contenant des  
items :item1, :item2 ;");
```

- Association des valeurs aux items

```
$requete->bindParam(":item1", $donnee1);  
$requete->bindParam(":item2", $donnee2);
```

- Exécution de la requête

- ```
$requete->execute();
```

# Requêtes préparées : Exemple

```
try {  
    ...    // connexion à la Bdd  
  
    $nom = "Jean Dupont"; // ou obtenu depuis le formulaire HTML  
    $email = "jeandupont@example.com";  
  
    $requete = $pdo->prepare("INSERT INTO utilisateurs (nom,email)  
                             VALUES (:nom, :email)");  
  
    $requete->bindParam(':nom', $nom);  
    $requete->bindParam(':email', $email);  
  
    $requete->execute();  
  
    echo "Utilisateurs insérés avec succès.";  
} catch (PDOException $e) {  
    print( "Erreur d'exécution :" . $e->getMessage() . "</br>" );  
    die();  
}
```

# Requêtes de sélection : 3 étapes

- Exécution de la requête

```
$requete = $pdo->query("select champ1,champ2 from ...where ...");
```

ou

```
$requete = $pdo->prepare("select champ1,champ2 from ...where ...");  
$requete->bindParam()  
$requete->execute() ;
```

- Exploitation des résultats

```
while ($ligne=$requete->fetch() ){  
    // $ligne contient la ligne courante des résultats de la requête  
    // on accède aux champs du select via $ligne['nomDuChamp'];  
}
```

- Libération de la ressource

```
$requete->closeCursor() ;
```

# Requêtes de sélection : Exemple

```
try {  
    ...    // connexion à la Bdd  
  
    $requete = $pdo->query("SELECT nom, prenom FROM utilisateurs");  
  
    while($ligne=$requete->fetch())  
    {  
        echo "Nom : {$ligne['nom']} <br>";  
        echo "Prénom : {$ligne['prenom']} <br>";  
    }  
    $requete->closeCursor();  
  
} catch (PDOException $e) {  
    print( "Erreur d'exécution :" . $e->getMessage() . "</br>" );  
    die();  
}
```

# Requêtes de sélection : Exemple

```
try {  
    ... // connexion à la Bdd  
    $age = 15;  
    $requete = $pdo->prepare("SELECT nom, prenom FROM utilisateurs WHERE age=:a");  
  
    $requete->bindParam(":a", $age);  
    $requete->execute();  
  
    while($ligne=$requete->fetch())  
    {  
        echo "Nom : {$ligne['nom']} <br>";  
        echo "Prénom : {$ligne['prenom']} <br>";  
    }  
    $requete->closeCursor();  
  
} catch (PDOException $e) {  
    print( "Erreur d'exécution :" . $e->getMessage() . "</br>" );  
    die();  
}
```



# Conclusion

- Documentation complète sur PDO :  
<http://www.php.net/manual/fr/book.pdo.php>
- Liste des méthodes applicables aux objets PDO :  
<http://www.php.net/manual/fr/class.pdostatement.php>