université de BORDEAUX

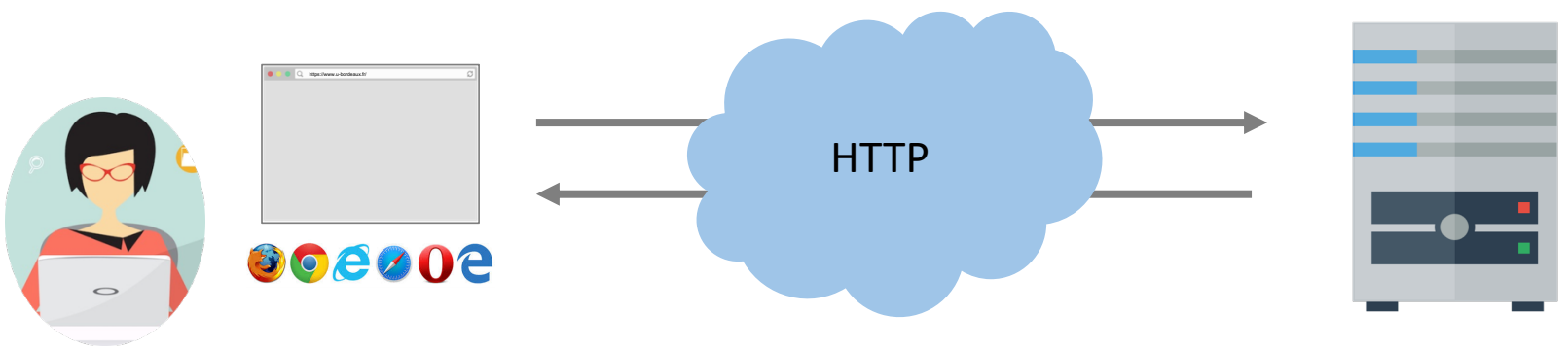
Bloc 1

Développement

université de BORDEAUX

Front et Back

Le développement d'un site web contient deux parties :



Front
Ce qui s'exécute
dans le navigateur

Back
Ce qui s'exécute
sur le serveur

Front – HTML, CSS, JavaScript

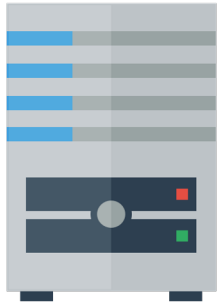
- **Affichage graphique**
  - Adapté à l'écran (*responsive*)
  - Accessibilité (handicap)
- **Interaction utilisateur**
  - Réactivité (asynchronisme)
  - Rapidité (*progressive apps*)
- **Communication serveur**
  - Envoi des requêtes (gestion des erreurs)
  - Récupération des réponses

Nombreux *framework* de développement

(jQuery, ReactJS, Angular, VueJS, etc.)

Back – PHP, C#, Java, JavaScript, Ruby, …

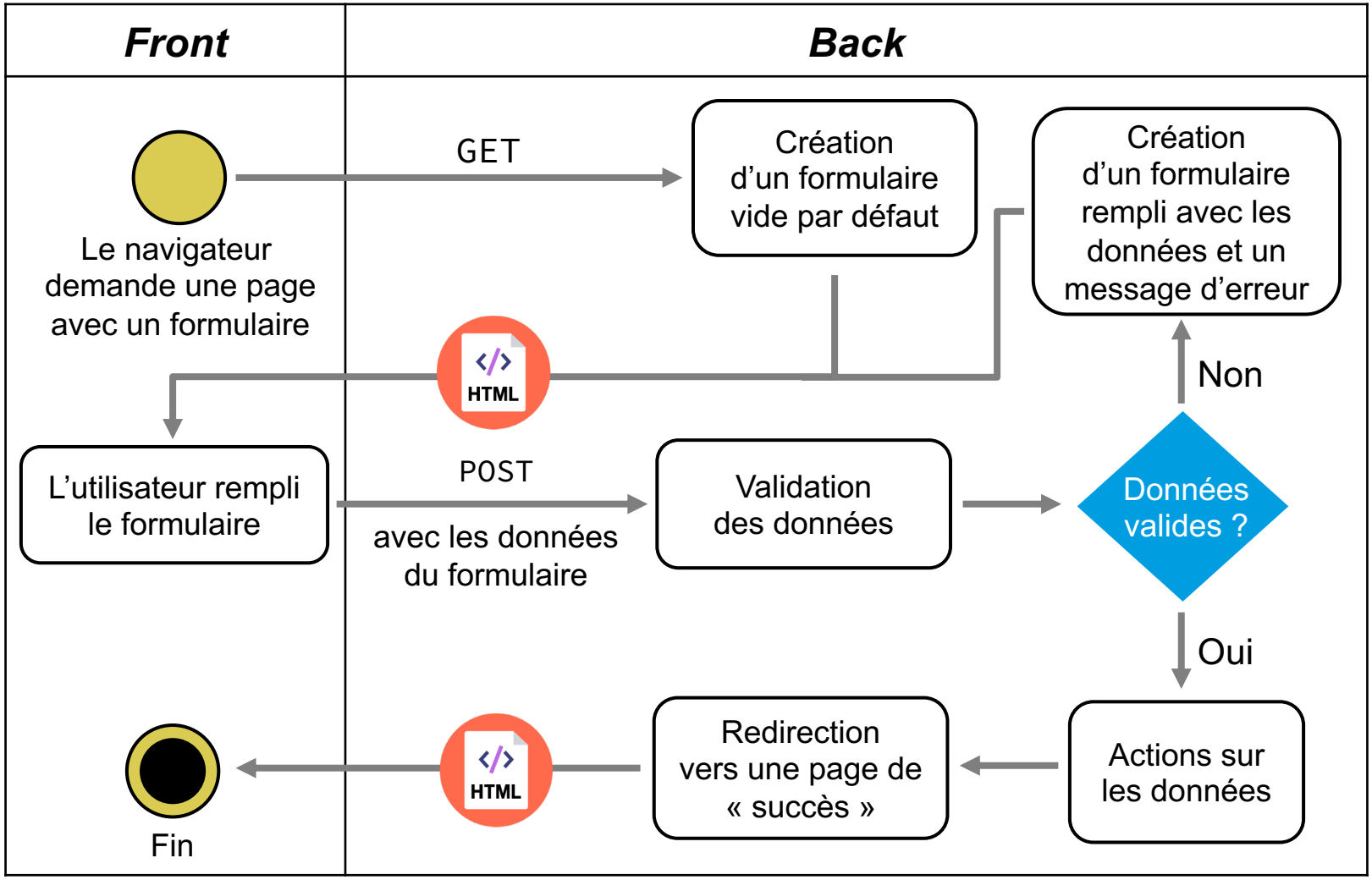
- **Support de l'application**
  - Services métier (recherche, calculs…)
  - Données (base de données)
- **Interaction avec le *Front***
  - Accès aux ressources
  - Lien avec l'utilisateur
- **Sécurité**
  - Authentification / confidentialité
  - Attaques / bots
- **Performance**
  - Nombre d'utilisateurs simultanés
  - Latence (temps entre la requête et la réponse)



Nombreux *framework* proposés

(Symfony, Rails, ASP, JSP, ExpressJS, etc.)

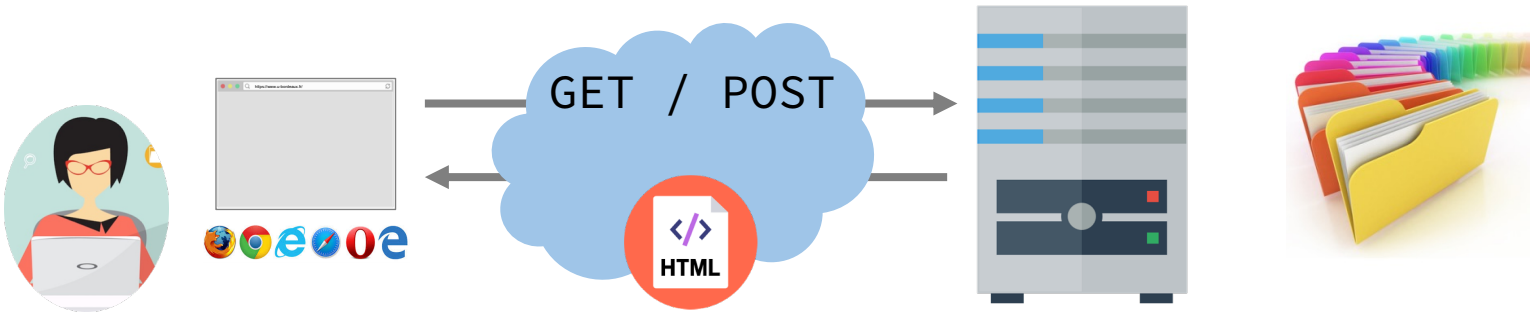
Exemple – Formulaire



Front I Back – Pages Statiques

→ **Page statique** : **toujours le même contenu** quels que soient l'utilisateur et ses interactions

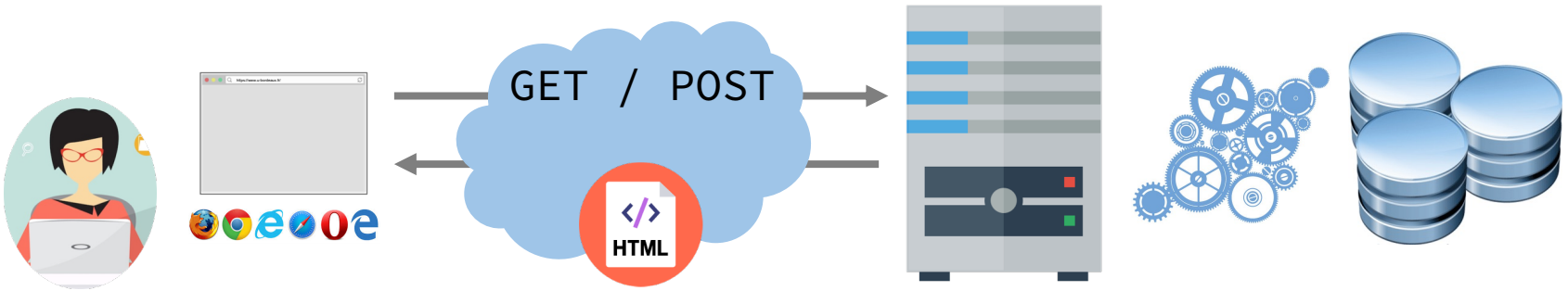
1. Le serveur possède toutes les pages statiques
2. Le navigateur demande la page qu'il veut lire (`GET`)
3. Depuis cette page, il demande d'autres pages (balises `<a>`, code JavaScript)



Front I Back – Pages Dynamiques - Moteur

→ **Page dynamique : contenu qui change** en fonction de l'utilisateur et de ses interactions

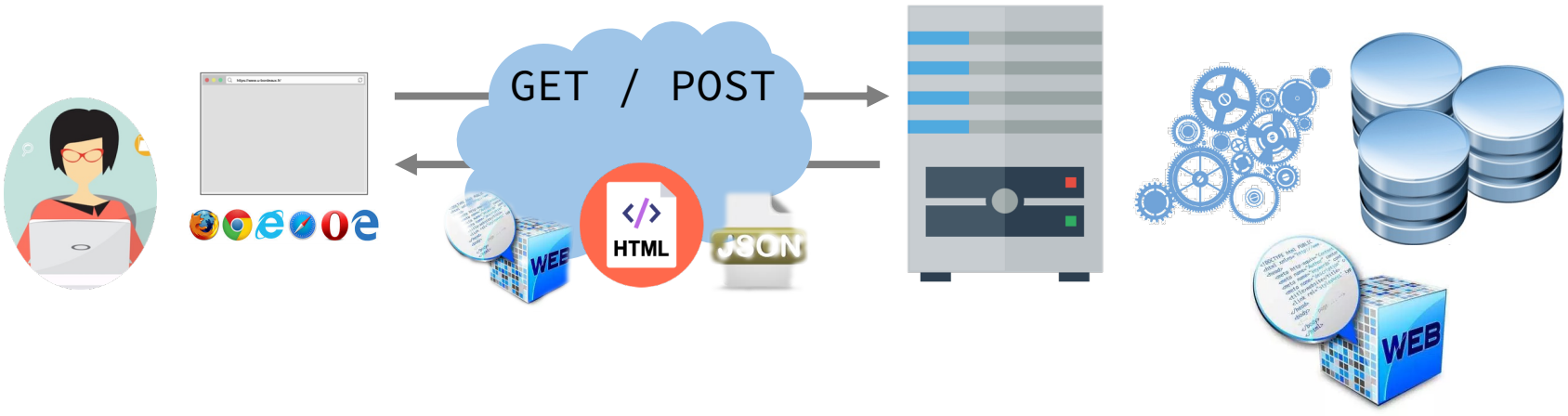
1. Le serveur dispose d'un moteur capable de générer les pages
2. Ce moteur est souvent couplé avec une base de données
3. L'utilisateur demande une page en précisant des paramètres
4. Le serveur génère la page HTML correspondante



Front I Back – Application Web

→ **Application web : exécutée sur le navigateur**, interagit avec le serveur

1. Le serveur dispose de l'application web (HTML, CSS, JavaScript)
2. L'utilisateur récupère l'application (GET)
3. L'application s'exécute dans le navigateur (JavaScript)
4. L'application interagit avec le serveur qui renvoi des ressources (JSON, HTML, images, etc.)



Résumé

- Développement bicéphale (*Front* / *Back*)
  - Objectifs complémentaires
  - Compétences différentes
- Équilibrage des traitements
  - Client lourd / Serveur léger
  - Client léger / Serveur lourd
- Tendance
  - *WebApp* (client lourd) / Service Web
  - Application *Smartphone* (client lourd) / Service Web
- Évolutions fortes
  - *Front* : *framework* MVC dans le navigateur
  - *Back* : fortement répartis (*cloud*) et très dynamique