

AJAX

Asynchronous JavaScript And XML

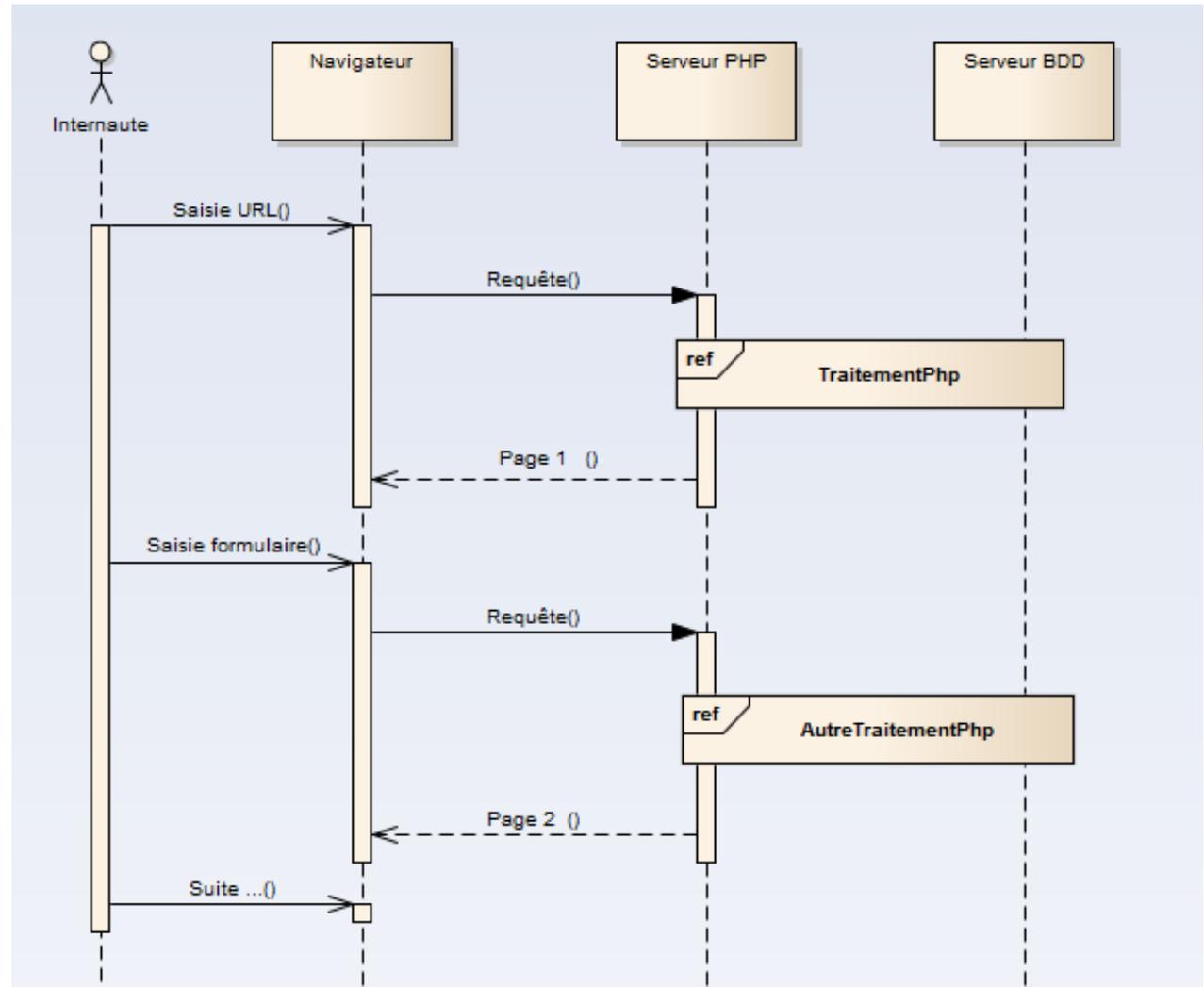
Objectifs:

- Obtenir des informations d'un serveur sans recharger la page html
- Intégrer dynamiquement ces informations à la page html (avec du JavaScript)

Sans AJAX ...

Exemple de site avec formulaire html

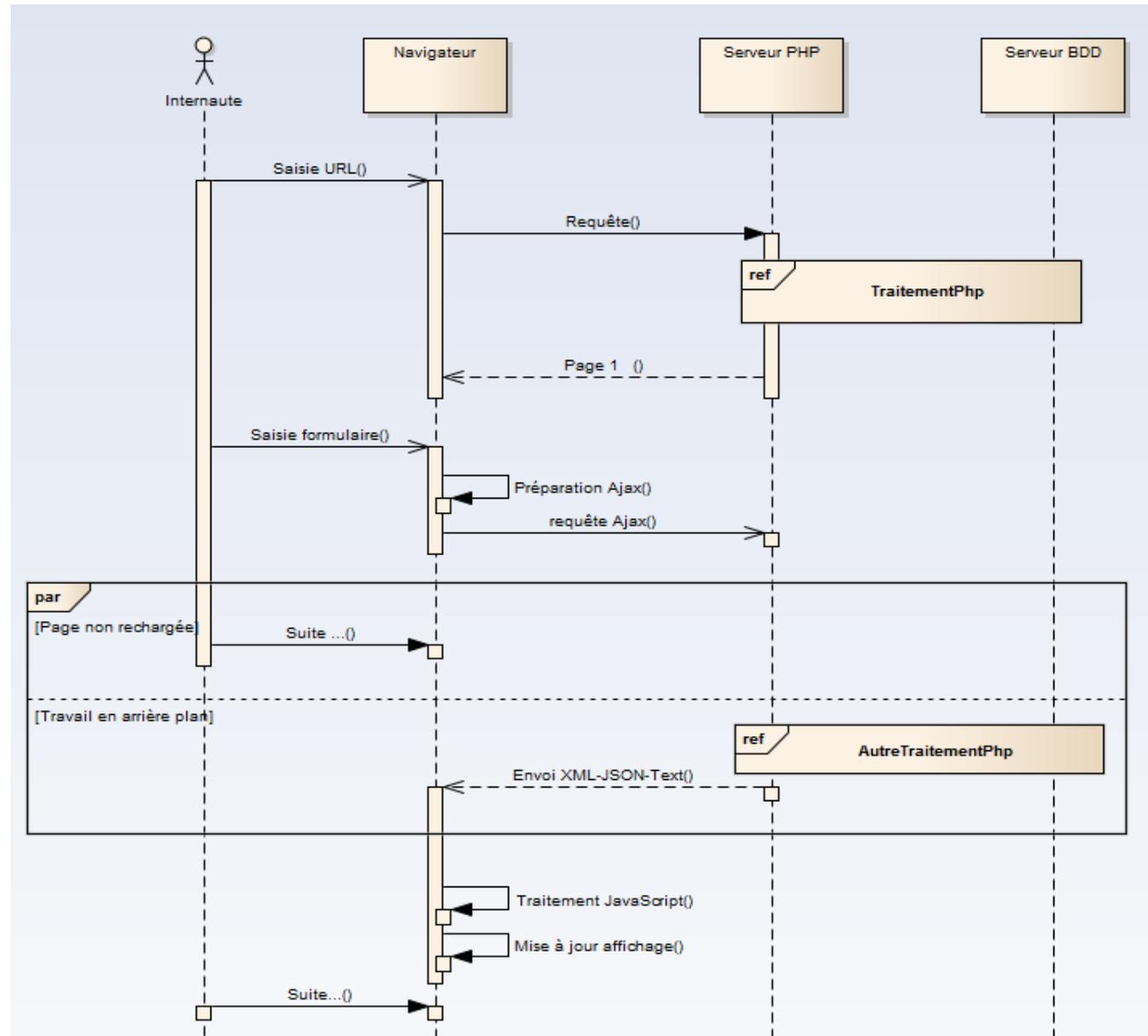
Chaque requête implique une mise à jour par rechargement de page.



Avec AJAX ...

Exemple de site avec
formulaire html

A chaque
requête, les
données
renvoyées par le
serveur arrivent
en arrière plan
déclenchent
l'exécution d'un
JavaScript



AJAX

- Objet JavaScript : XMLHttpRequest

```
xhr = new XMLHttpRequest (); // Création de l'objet Ajax

// Préparation de l'échange Ajax :
// 1-On indique la fonction JavaScript à exécuter lorsque des données seront reçues :
xhr.onload = travail; // 'travail' est le nom de la fonction.

//2- On indique le nom de la page qui fournit les données :
xhr.open ("GET", "toto.php", true); // Page 'toto.php' à contacter.

// 3-On envoie de la requête Ajax :
xhr.send ();
```

Types de pages à contacter :

Script Serveur (php, asp,...),
Fichiers textes : txt, xml, ...

AJAX

- Objet : **XMLHttpRequest** : Méthode « **open** » :

```
xhr.open ("GET", "toto.php?maVar=3", true);
```

Méthode de transmission:

GET (exemple ci-dessus) ou POST.

Pour les requête POST, il faut créer un « header » semblable à celui d'un formulaire :

```
var form = new FormData();  
form.append('maVar', 3);  
xhr.send(form);
```

Mode Asynchrone (true)
ou Synchron (false)

Le mode synchrone peut « bloquer » le navigateur puisqu'on doit attendre la réponse du serveur pour continuer le script.

AJAX

La question du Cross-domain

- Restrictions de l'objet : XMLHttpRequest :

La page appelée par Ajax doit être sur le même serveur que les pages du site.

Sinon, le SERVEUR qui héberge la page appelée par Ajax doit ajouter au *header* de la page :

```
header ("Access-Control-Allow-Origin: *");
```

Le « * » autorise les requêtes de TOUS les clients.
On peut aussi n'autoriser que certaines plages d'@ip