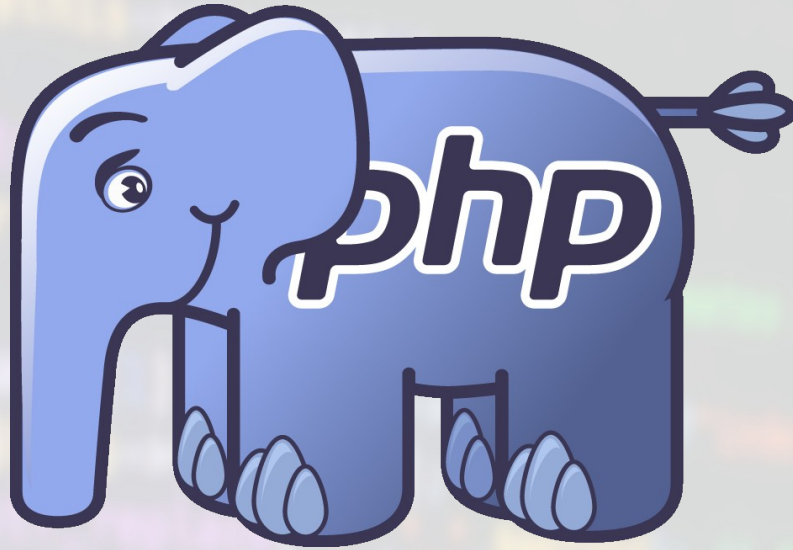


Protocole HTTP, cookies, sessions, authentication

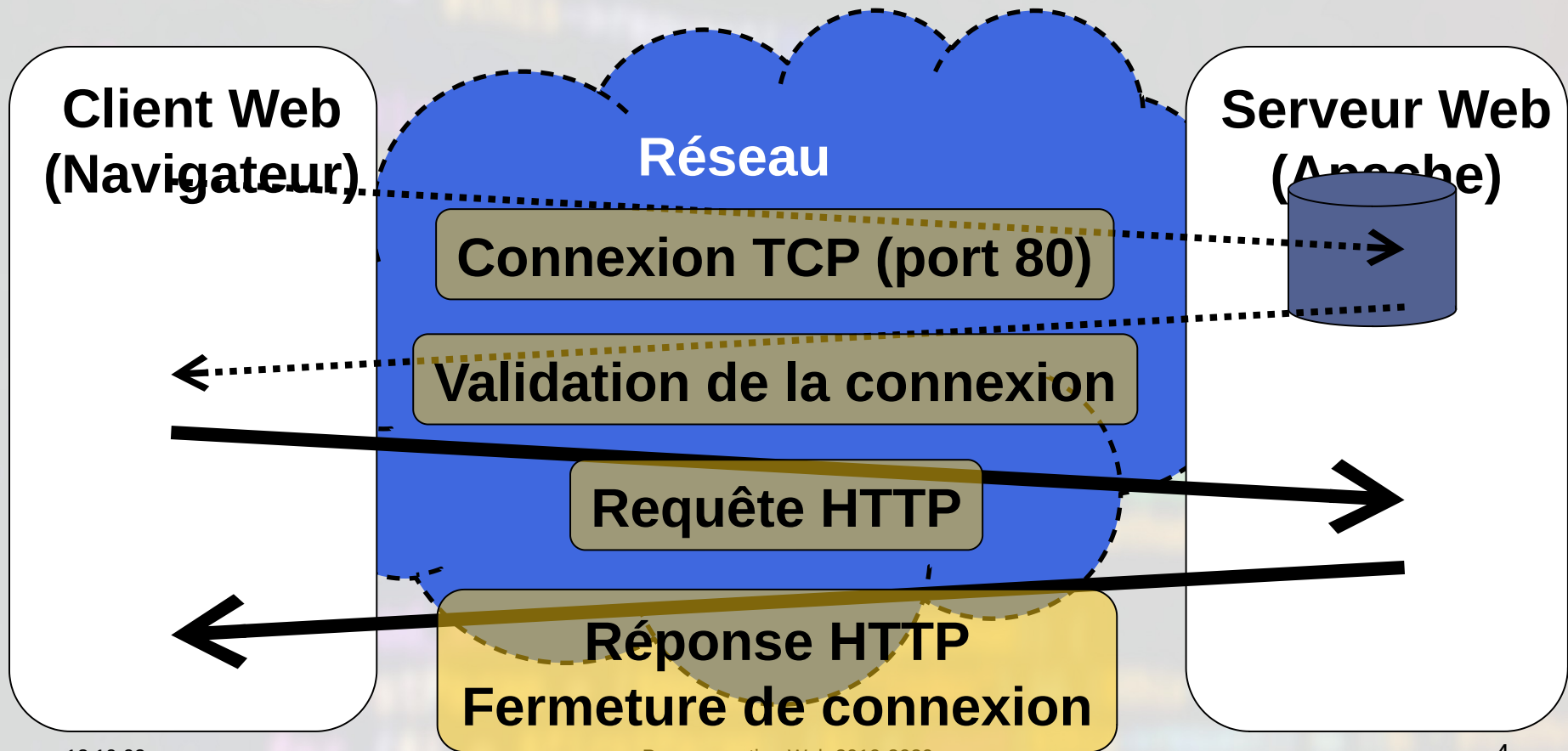


INTRODUCTION

Introduction

- HTTP : HyperText Transfer Protocol
- HTTP : Protocole du Web
- Protocole d'échange entre client et serveur
- Protocole orienté ligne de caractères
- Dans notre cas :
 - Étude du protocole pour la culture
 - Compréhension des cookies
 - Compréhension des sessions
 - Utilisation avancée de PHP (côté serveur)
 - Utilisation d'AJAX

Utilisation de HTTP/1.0



Remarques importantes

- Le **client ouvre** la connexion
- Le **serveur ferme** la connexion



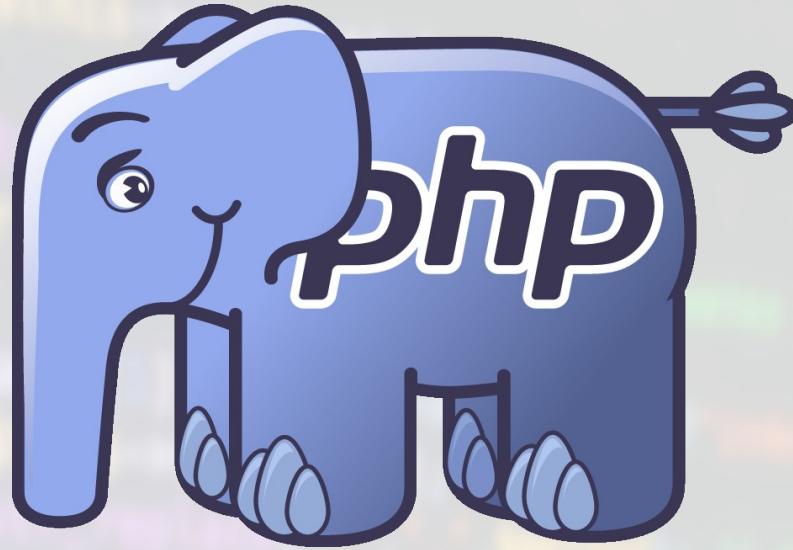
- 1 transaction = 1 ressource transférée (v 1.0)
- **Protocole sans état**
 - Aucune information gardée entre deux transactions
 - Le serveur "oublie" le client après chaque transaction



- **Problème pour la gestion d'une session**

Autres remarques importantes

- Le client demande des ressources
- Le serveur répond aux demandes des clients :
 - Délivre la ressource demandée si possible
 - Informe de la raison de non remise
- Requêtes émises en clair
- Réponses émises en clair



STRUCTURE DES MESSAGES

Structure d'un message HTTP

Orienté lignes de caractères :

= retour à la ligne

Requête ou réponse :

Requête ou état

[Entête: valeur]

[Entête: valeur]

[...]

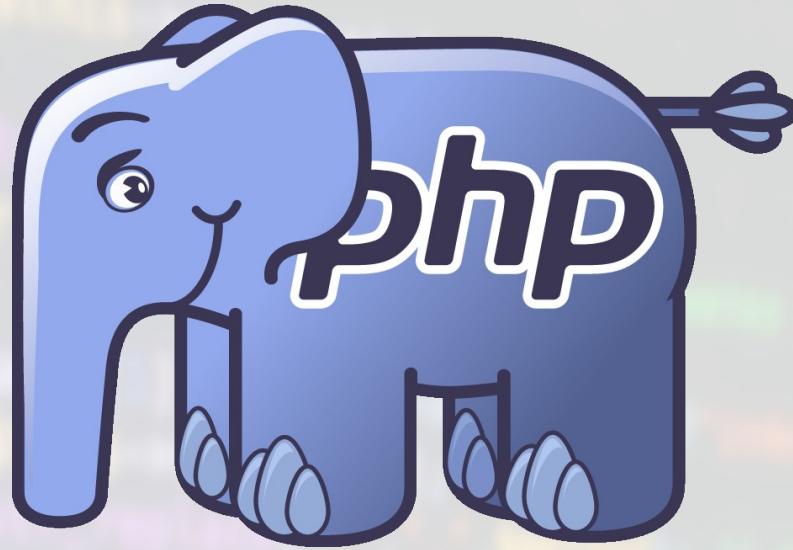
Ligne blanche

[Corps de message]



HTTP 1.0: 16 en-têtes

HTTP 1.1: 46 en-têtes



REQUÊTES HTTP

Structure d'une Requête HTTP 1.0

Une requête HTTP est un ensemble de lignes envoyé au serveur par le navigateur. Elle comprend :

- **Une ligne de requête**: comprend trois éléments devant être séparés par un espace
- **Les champs d'en-tête de la requête**: un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la requête et/ou le client (Navigateur, système d'exploitation, ...).
- **Le corps de la requête**: un ensemble de lignes optionnelles devant être séparées des lignes précédentes par une ligne vide.

Requête HTTP 1.0

GET|POST|HEAD *chemin* HTTP/1.0

User-Agent: *agent*

[Entête: valeur]

[Entête: valeur]

[...]

Ligne blanche

[Données]

Méthodes de requête HTTP

GET

demande des informations et la ressource désignée

HEAD

demande des informations concernant la ressource

POST

envoi de données (formulaire vers le serveur) et demande la ressource désignée

PUT

enregistrement du corps de la requête à l'URL indiquée

DELETE

suppression de la ressource désignée par l'URL

Exemple de requête HTTP

GET / HTTP/1.0

Ligne blanche

POST /page1.php HTTP/1.0

User-Agent: Mozilla/5.0 (Windows NT 10.0; ...)

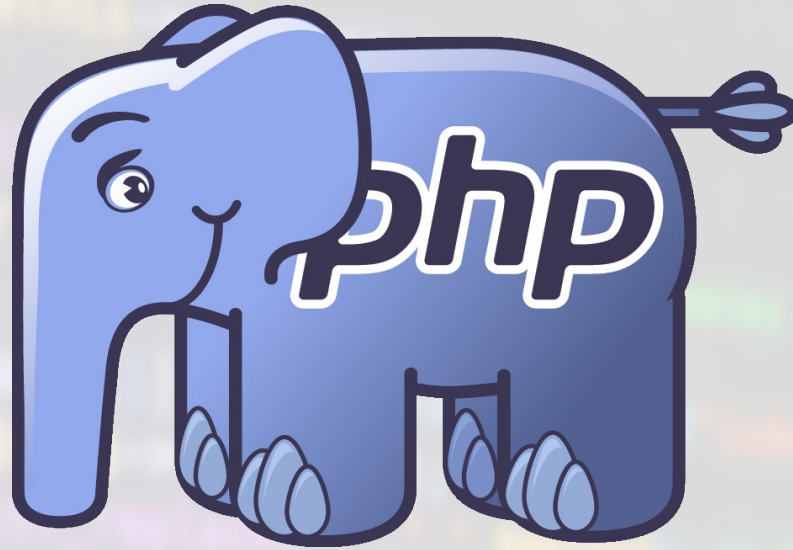
Gecko/20100101 Firefox/60.0

Content-Type: application/x-www-form-urlencoded

Content-Length: 32

Ligne blanche

prenom=SpongeBob&nom=SquarePants



RÉPONSES HTTP

Réponse HTTP

Une réponse HTTP est un ensemble de lignes envoyées au navigateur par le serveur. Elle comprend :

- **Une ligne de statut**: c'est une ligne précisant la version du protocole utilisé et l'état du traitement de la requête à l'aide d'un code et d'un texte explicatif.
- **Les champs d'en-tête** de la réponse: il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la réponse et/ou le serveur.
- **Le corps de la réponse**: il contient le document demandé

Réponse HTTP

HTTP/version code phrase

Content-Type: *type_mime*

[Entête: valeur]

[Entête: valeur]

[...]

Ligne blanche

[Données]

**Obligatoire si
corps présent**

Le client doit être informé de la
nature de la ressource :

HTML

Image JPEG

Doc PDF

...

Exemple de réponse HTTP

HTTP/1.x 200 OK

Date: Mon, 25 Apr 2005 04:25:17 GMT

Server: Apache/2.0.46 (Red Hat)

X-Powered-By: PHP/4.3.2

Content-Type: text/html

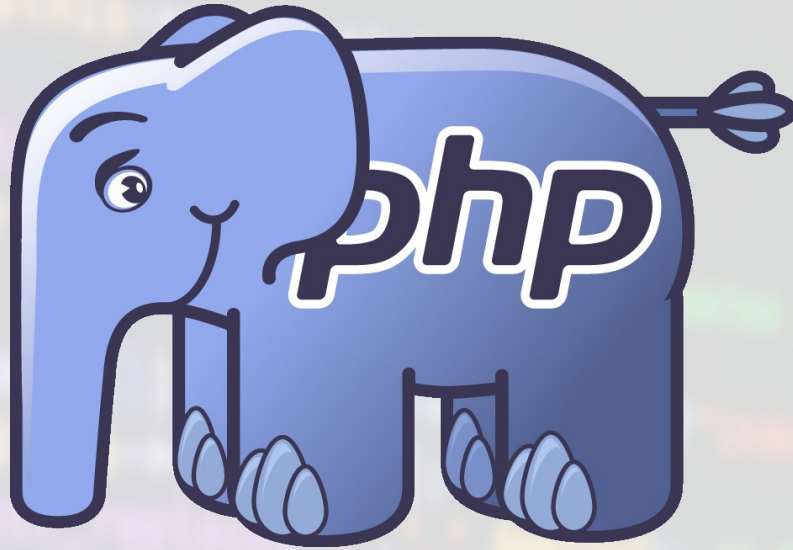
Ligne blanche

<html>

<head><title> ...

États des réponses HTTP

- Codes à 3 chiffres + phrase
- 1er chiffre : classe de réponse
 - **1xx** : Information (HTTP 1.1)
 - **2xx** : Succès
 - **200 OK**
 - **3xx** : Redirection
 - **304 NOT MODIFIED**
 - **4xx** : Erreur client
 - **403 FORBIDDEN**
 - **404 NOT FOUND**
 - **5xx** : Erreur serveur
 - **500 INTERNAL ERROR**

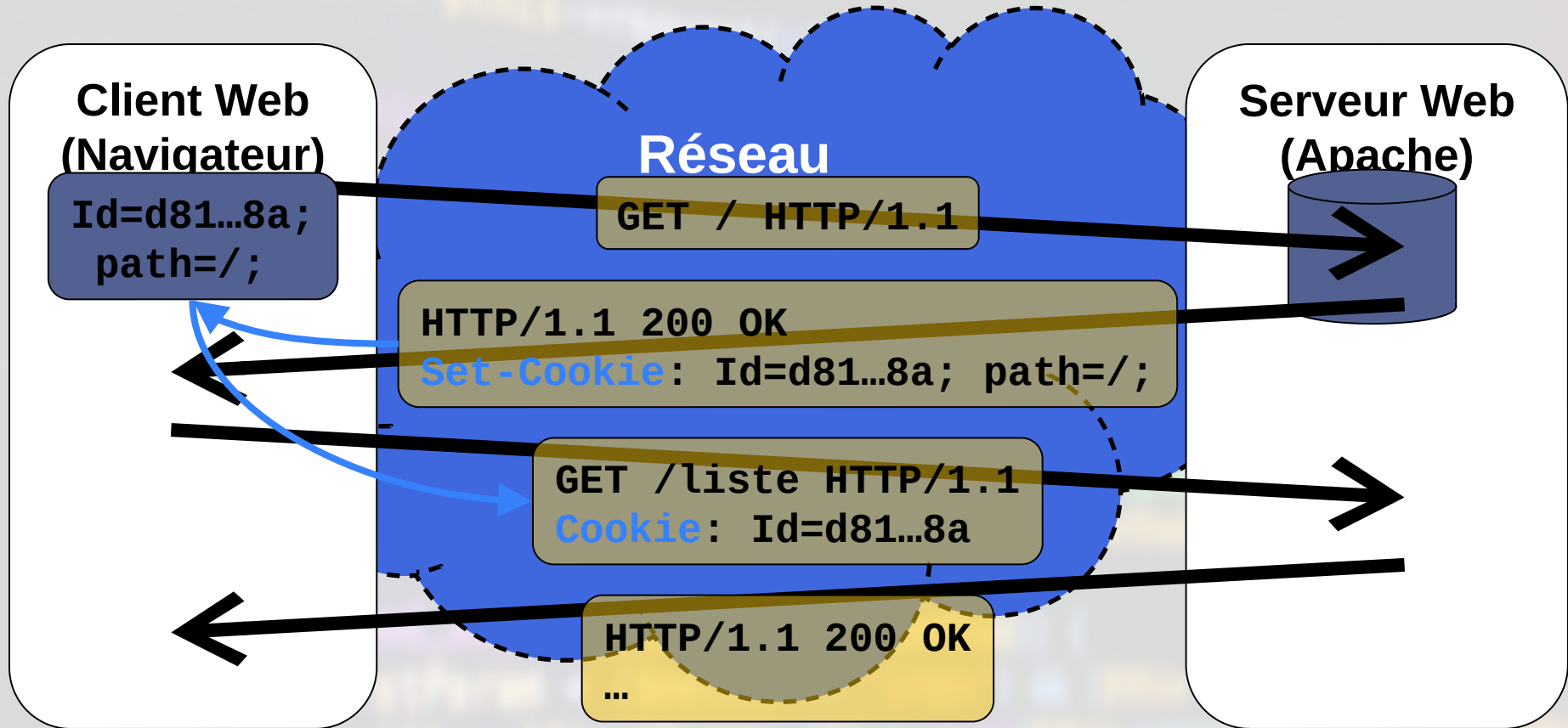


COOKIES

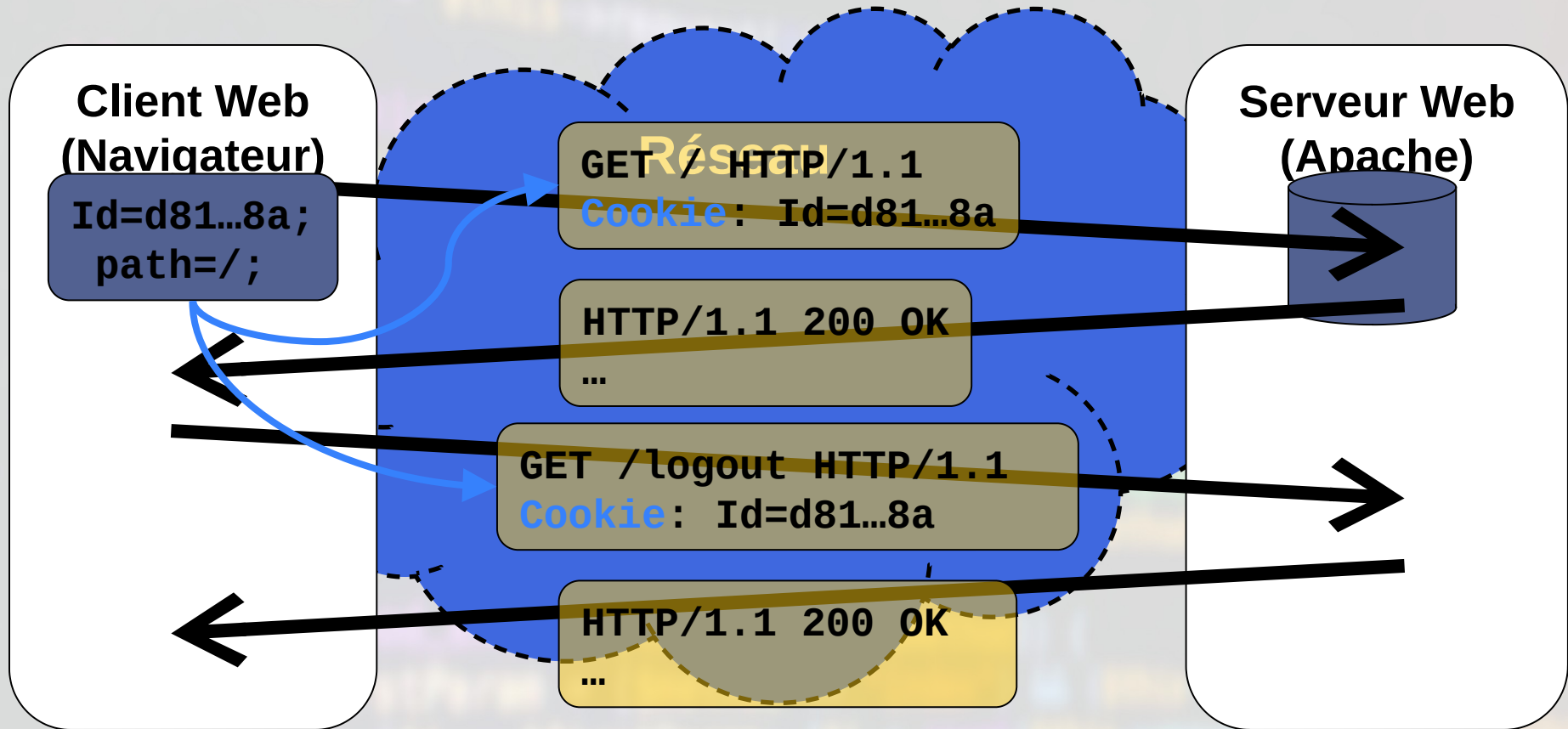
Cookies

- But :
 - Éviter que le serveur « oublie le client »
 - Maintenir un « mode connecté » (= session)
 - Rendre transparent un échange client / serveur
 - Exemple e-commerce : ajouter des articles au panier
- Serveur :
Set-Cookie: *var=val*; *expires=date; path=chemin; domain=domaine*
- Client :
Cookie: *var=val*;

Cookies, principe des échange



Cookies, principe des échanges (suite)



Cookies en PHP

- Placer un cookie sur le client :

```
bool setcookie(    string name  
                  [, string value  
                  [, int    expire  
                  [, string path  
                  [, string domain  
                  [, bool    secure
```

Domaine de validité, disponibilité :

Cookie uniquement par HTTP ?

true ☐ uniquement HTTP

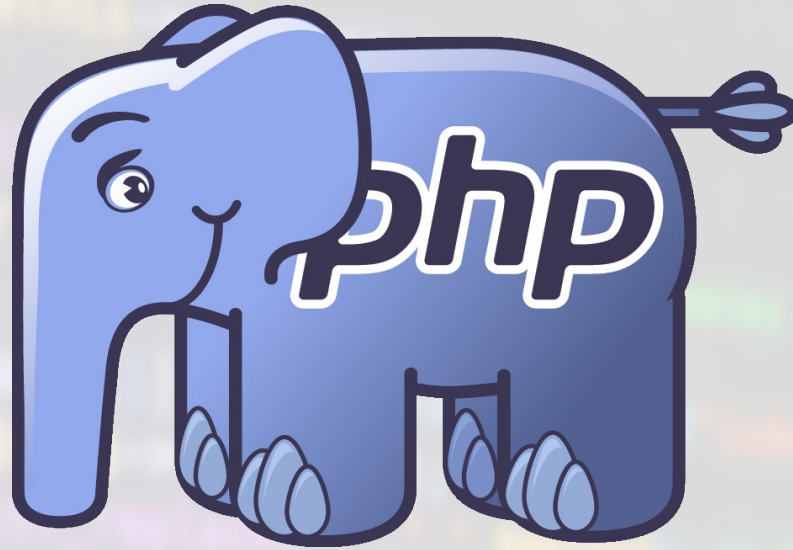
false ☐ défaut, HTTP, JavaScript, ...

Cookies en PHP

- Vérifier la présence d'un cookie :
 - tableau associatif superglobal **`$_COOKIE`**
 - ex : cookie 'passage' `$_COOKIE['passage']`
- Remarques utiles :
 - Le **cookie** doit être **placé** avant **echo** (en-tête HTTP)
 - Le **cookie** placé avec **setcookie()** n'est **accessible** qu'**au prochain chargement** de page :
En-tête HTTP Set-Cookie: suivi de l'en-tête HTTP Cookie: à la demande de page suivante
 - **Effacer un cookie** déjà placé :
le remplacer avec les **mêmes paramètres** mais avec une **valeur vide** ou **false**

Cookies en PHP

```
<?php
    setcookie('user_id', '1234');
    setcookie('user_pref', 'dark_theme', time()+3600*24, '/', '', true, true);
?>
<!DOCTYPE html>
<html>
    <head> ..... </head>
    <body>
        <h1>Titre principal</h1>
        <?php
            ?>
            <p>Un paragraphe</p>
        </body>
</html>
```

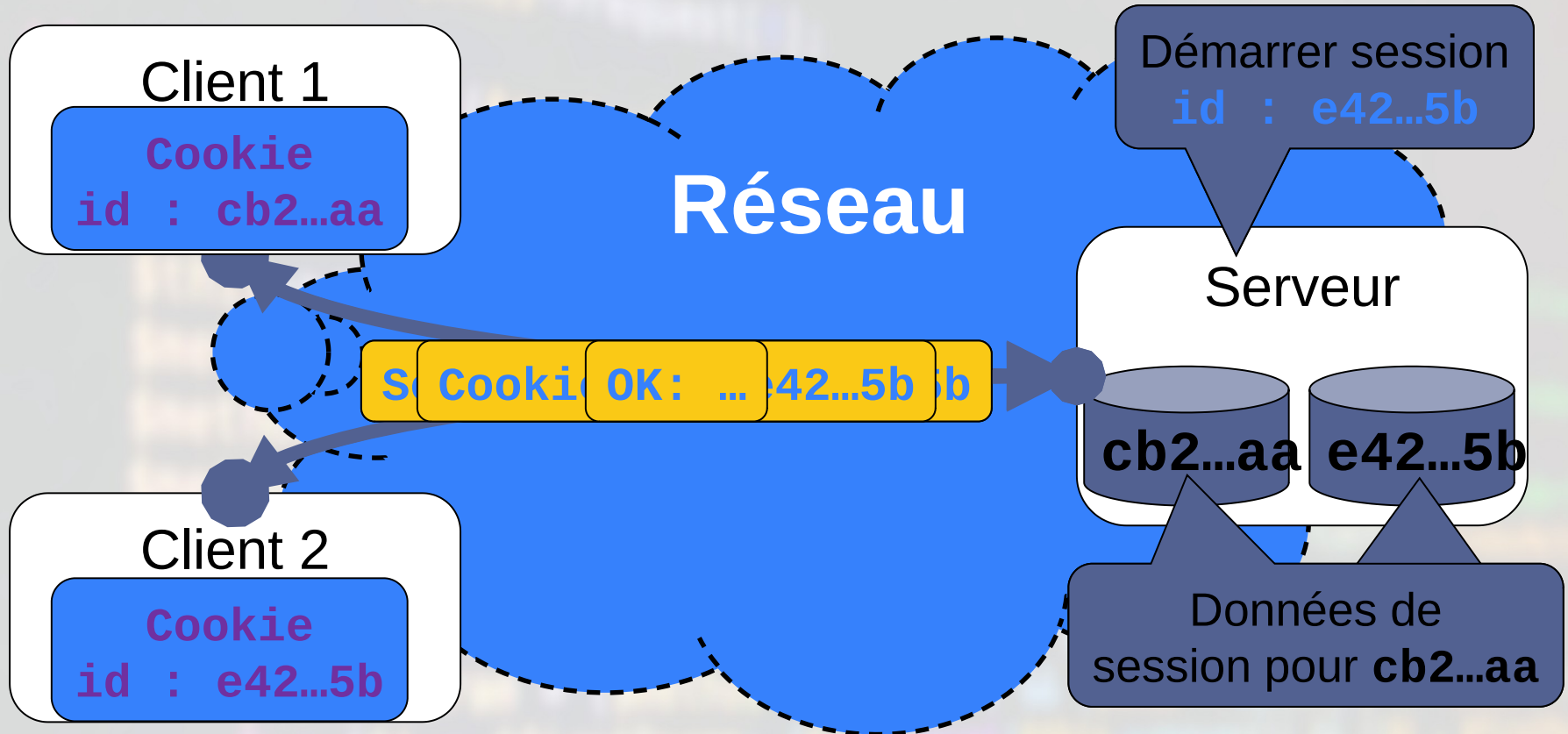


SESSIONS

Sessions

- Stockage sur le serveur de données associées à un client particulier
- Nécessite une **identification unique pertinente et persistante des clients**
 - **Identifiant de session** (~~MD5 128bits~~ / ~~SHA-1 160bits~~ / SHA-2 512bits)
 - Persiste par **paramètre d'URL** ou **cookie**
- **Évite l'échange permanent de données** (en dehors de l'identifiant)
- Nécessite la **linéarisation des variables** pour leur stockage (fichier, BD, personnalisé)
- Simule un **mode connecté**

Sessions



Sessions

- Créer ou restaurer une session
 - `bool session_start (void)`
- Manipuler les données de session
 - Tableau associatif superglobal `$_SESSION`
- Lire ou définir l'identifiant de session
 - `string session_id ([string id])`
- Détruire toutes les variables d'une session
 - `void session_unset (void)`
- Détruire une session
 - `bool session_destroy (void)`