

RAPPELS

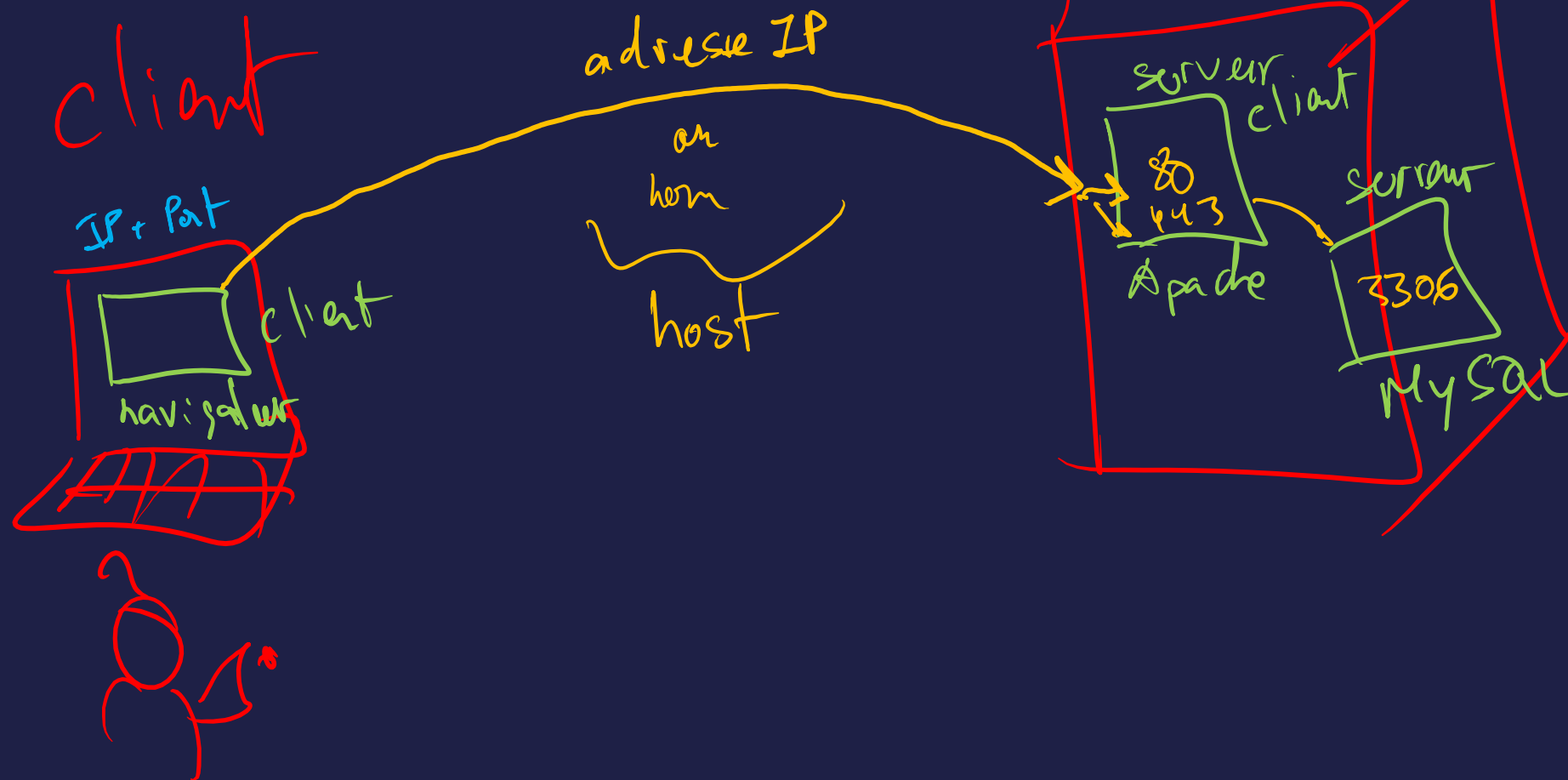
Michael
X  NATIS



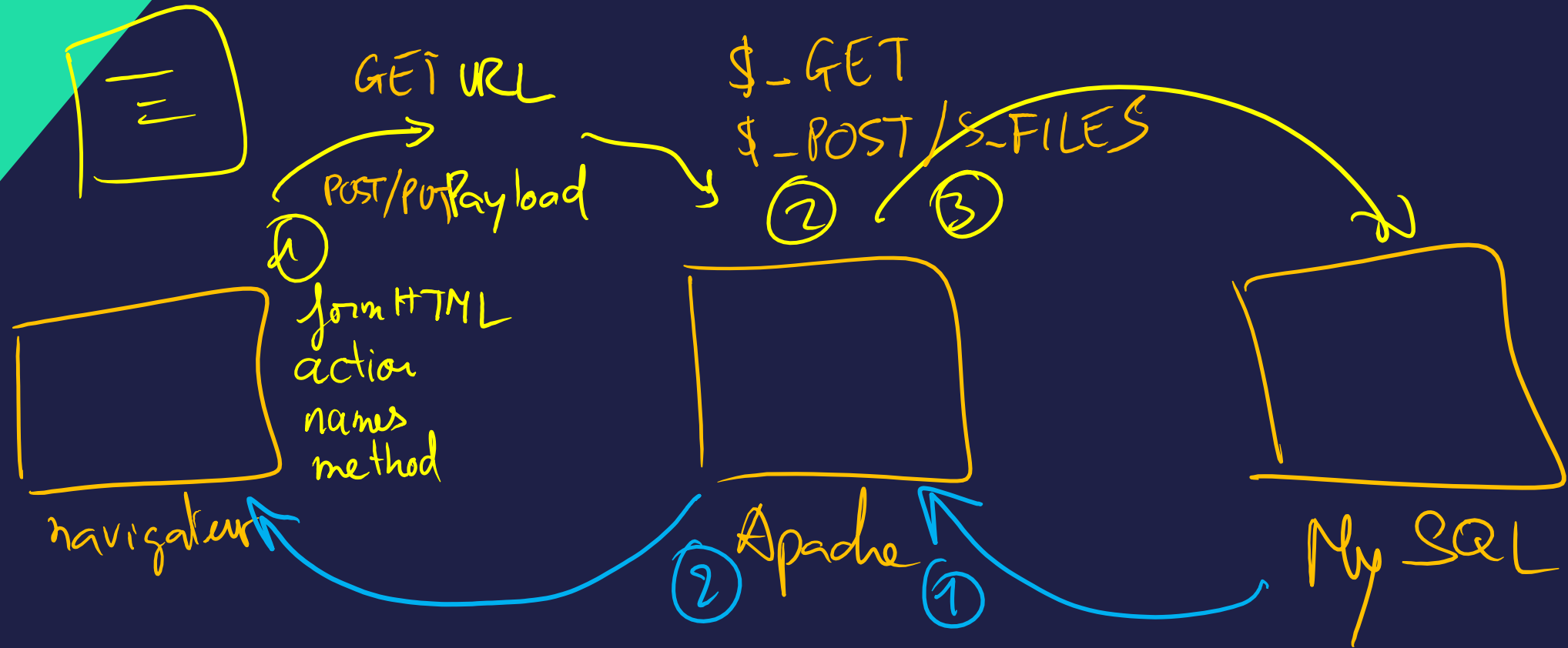
Compétence demandée :
Maîtriser les concepts GET, POST, Cookies,
Session et PDO

1. GET et POST
2. Cookies et Session
3. Mysqli et PDO

Une histoire de ports



L'aller et le retour



Les méthodes de requêtes GET et POST

LE TRANSPORT BACKEND-FRONTEND



Chrome

Je cherche la page
de google pour 'vélo'

HTML + JavaScript + CSS
EXECUTION DE LA REPONSE



REPONSE HTML, CSS, JS + CODE



Apache

Je dessine
ma page

PHP, Python, Node.js
EXUCUTION DU PGM

LE TRANSPORT BACKEND-FRONTEND



Chrome

Je cherche la page
de google pour 'vélo'

HTML + JavaScript + CSS
EXECUTION DE LA REPONSE



REPONSE HTML, CSS, JS
+ EN-TETES (CODE ...)

LOCATION



Apache

Je dessine
ma page

PHP, Python, Node.js
EXUCUTION DU PGM

Une requête est composée de :

1. Une **méthode** (**GET**, **POST**, PUT, PATCH, DELETE, HEAD)
2. Des **en-têtes** (Authorization, X-headers)
3. Un **corps**/contenu (payload)

Avant/Aujourd'hui : on utilise le **POST** lorsqu'on ne veut pas que les données de la requête soit visibles dans l'URL

Aujourd'hui/Futur : on utilise le **POST** pour exprimer **l'intention de créer** de l'information

Avant/Aujourd'hui : on utilise le **POST** lorsqu'on ne veut pas que les données de la requête soit visibles dans l'URL

Aujourd'hui/Futur : on utilise le **POST** pour exprimer **l'intention de créer** de l'information



GET : Les données sont à l'intérieur de l'URL

Frontend :

```
GET localhost/exos/exemple.php?cle=valeur
```

Backend PHP :

```
$boite = $_GET['cle'];
```

POST : Les données sont à l'intérieur du **payload**

Frontend :

```
POST localhost/exos/exemple.php
```

Backend PHP :

```
$boite = $_POST['cle'];
```

Pour **savoir** si une **clé est présente** dans le dictionnaire

Backend PHP (la fonction retourne un boolean):

```
isset($_POST['cle'])
```


Un **cookie**, c'est quoi ?

Enregistrer temporairement des
informations sur le visiteur
COOKIE et **SESSION**

LE TRANSPORT BACKEND-FRONTEND



Chrome

Je cherche la page
de google pour 'vélo'

HTML + JavaScript + CSS
EXECUTION DE LA REPONSE



REPONSE HTML, CSS, JS
+ EN-TETES (CODE...)



Apache

Je dessine
ma page

PHP, Python, Node.js
EXUCUTION DU PGM

Backend PHP pour demander au frontend de créer un magic cookie :

```
setcookie('cle', 'valeur');
```

Backend PHP pour lire le magic cookie :

```
$boite = $_COOKIE['cle'];
```


LE TRANSPORT BACKEND-FRONTEND



Chrome

Je cherche la page
de google pour 'vélo'

HTML + JavaScript + CSS
EXECUTION DE LA REPONSE



REPONSE HTML, CSS, JS
+ EN-TETES (CODE...)



Apache

Je dessine
ma page

PHP, Python, Node.js
EXUCUTION DU PGM

Backend PHP pour ajouter une clé à la session :

```
session_start();  
$_SESSION['cle'] = 'valeur';
```

Backend PHP pour récupérer la valeur d'une clé :

```
session_start();  
$boite = $_SESSION['cle'];
```

Backend PHP pour détruire la session :

```
session_destroy();
```

Le **cookie** est stocké du côté du **client** (émetteur de la demande).

La **session** est stocké sur le **serveur**.

Le **cookie** est stocké du côté du **c**lient
(émetteur de la demande).

La **session** est stocké sur le **s**erveur.



La connexion à une base de données MySQL Mysqli et PDO

Mysqli

```
$connection = mysqli_connect('challenge.xonatis.academy', 'wf3_621', 'tropbien', 'wf3_621');  
$resource = mysqli_query($connection, "SELECT * FROM exo1_categories");  
while ($ligne = mysqli_fetch_assoc($resource))  
{  
  
}
```

PDO

```
$pdo = new PDO('mysql:host=challenge.xonatis.academy;dbname=wf3_621', 'wf3_621', 'tropbien');  
$statement = $pdo->query("SELECT * FROM exo1_categories");  
while ($ligne = $statement->fetch())  
{  
  
}
```

Mysqli ou PDO ?

Mysqli est une interface pour MySQL construite pour PHP

PDO est une couche d'abstraction pour l'accès aux bases de données

