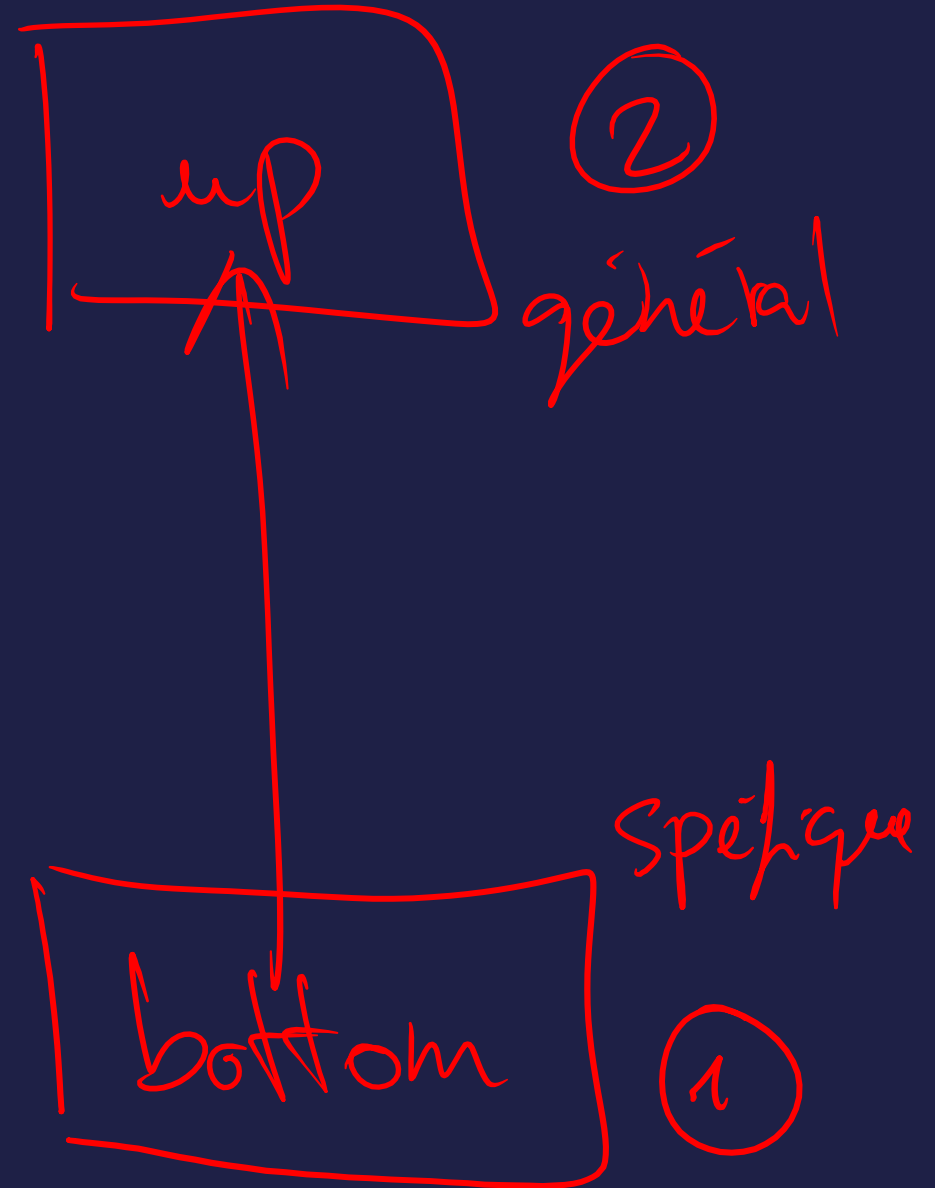
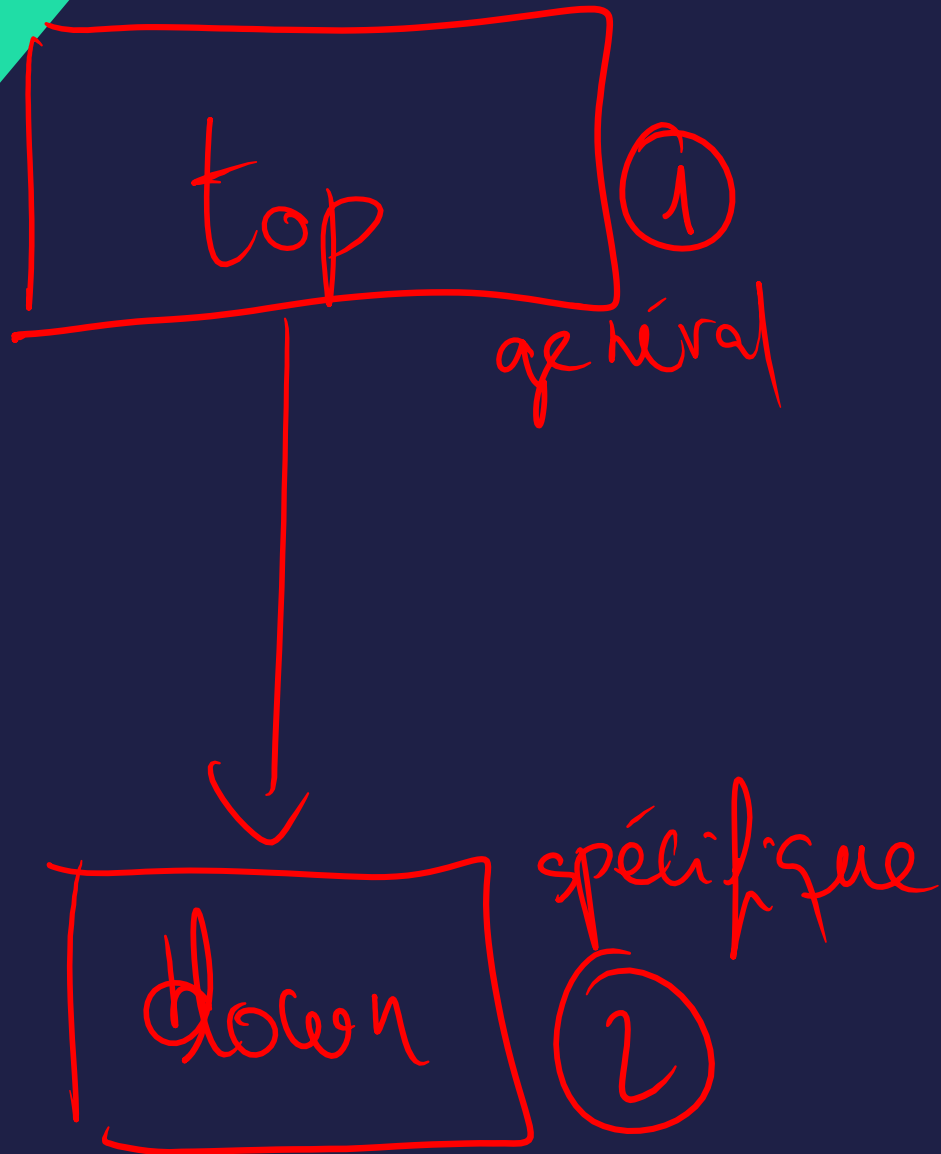


Architecture

Michael
X  NATIS



Compétence demandée :
Comprendre l'architecture générale du
backend



1. Le but de PHP
2. Les serveurs
3. L'environnement PHP

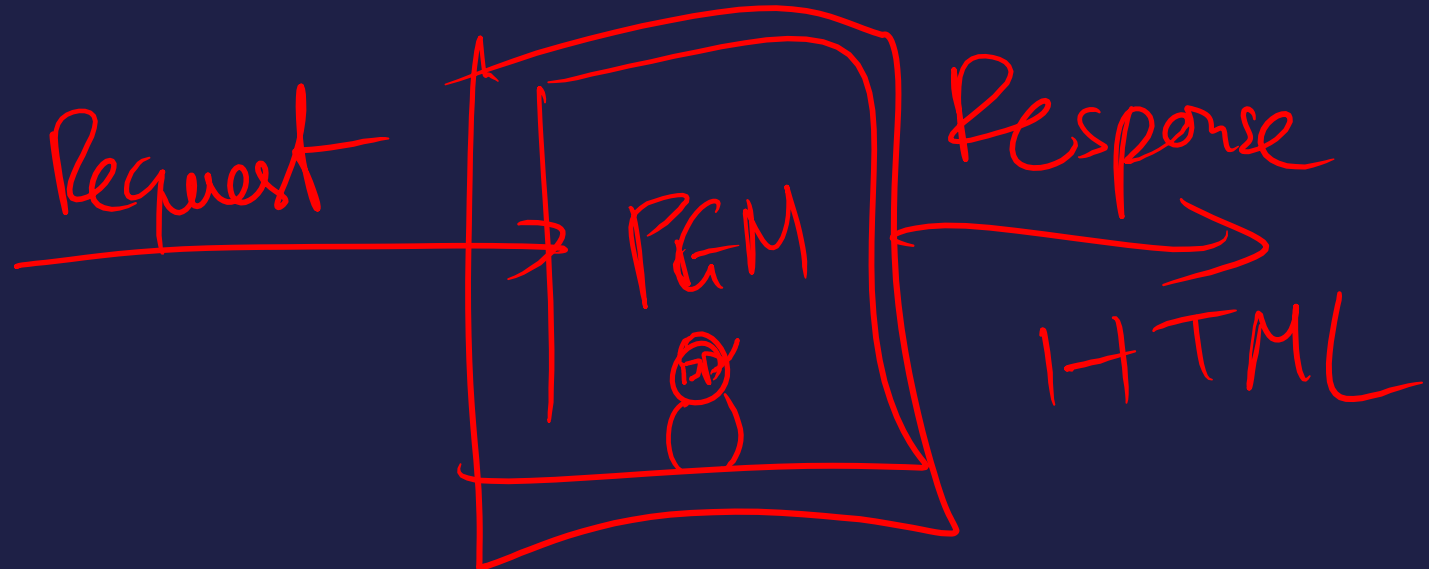
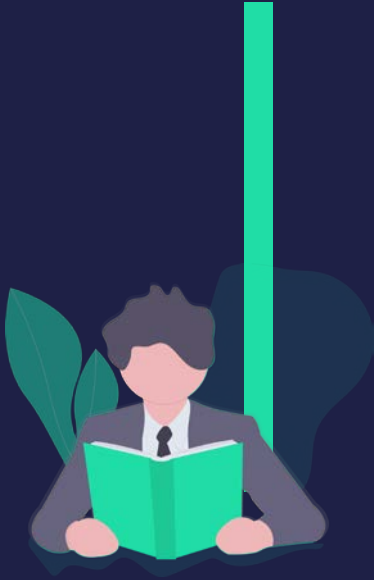
1. Le but de PHP

Une page statique : une page qui est renvoyée au client telle qu'elle a été stockée sur le serveur

ce n'est pas immobile

Une page dynamique: une page qui est
générée à la volée par le serveur

Une page dynamique: une page qui est générée à la volée par le serveur



2. Les serveurs

Un serveur

SR

Un serveur

Per

Une machine ou un logiciel fournissant
des services à d'autres machines ou
logiciels, par exemple 24h/24

honey me

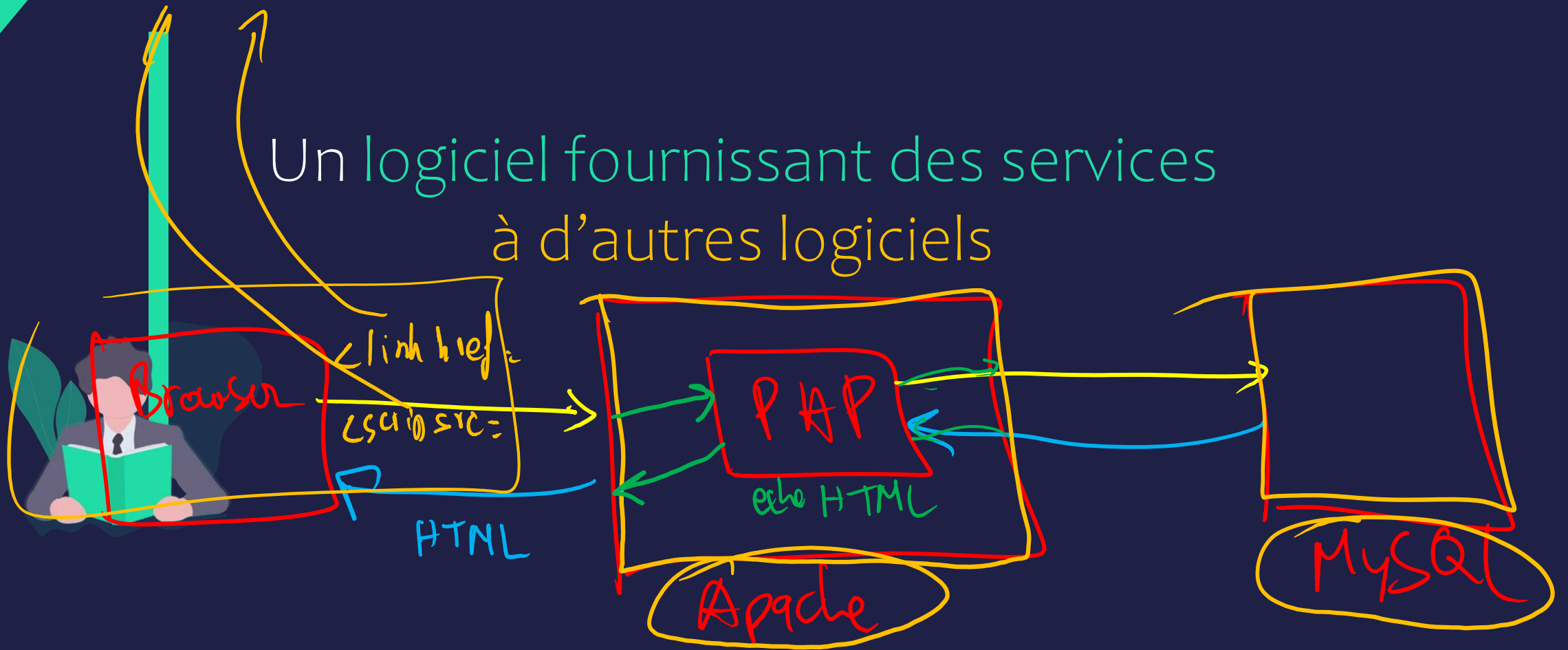
```
graph TD; SR --> machine; Per --> logiciel; honey_me[honey me] --> services
```

Un serveur

Un logiciel fournissant des services
à d'autres logiciels

Un serveur

Un logiciel fournissant des services
à d'autres logiciels



Un serveur

Un serveur est localisé par :

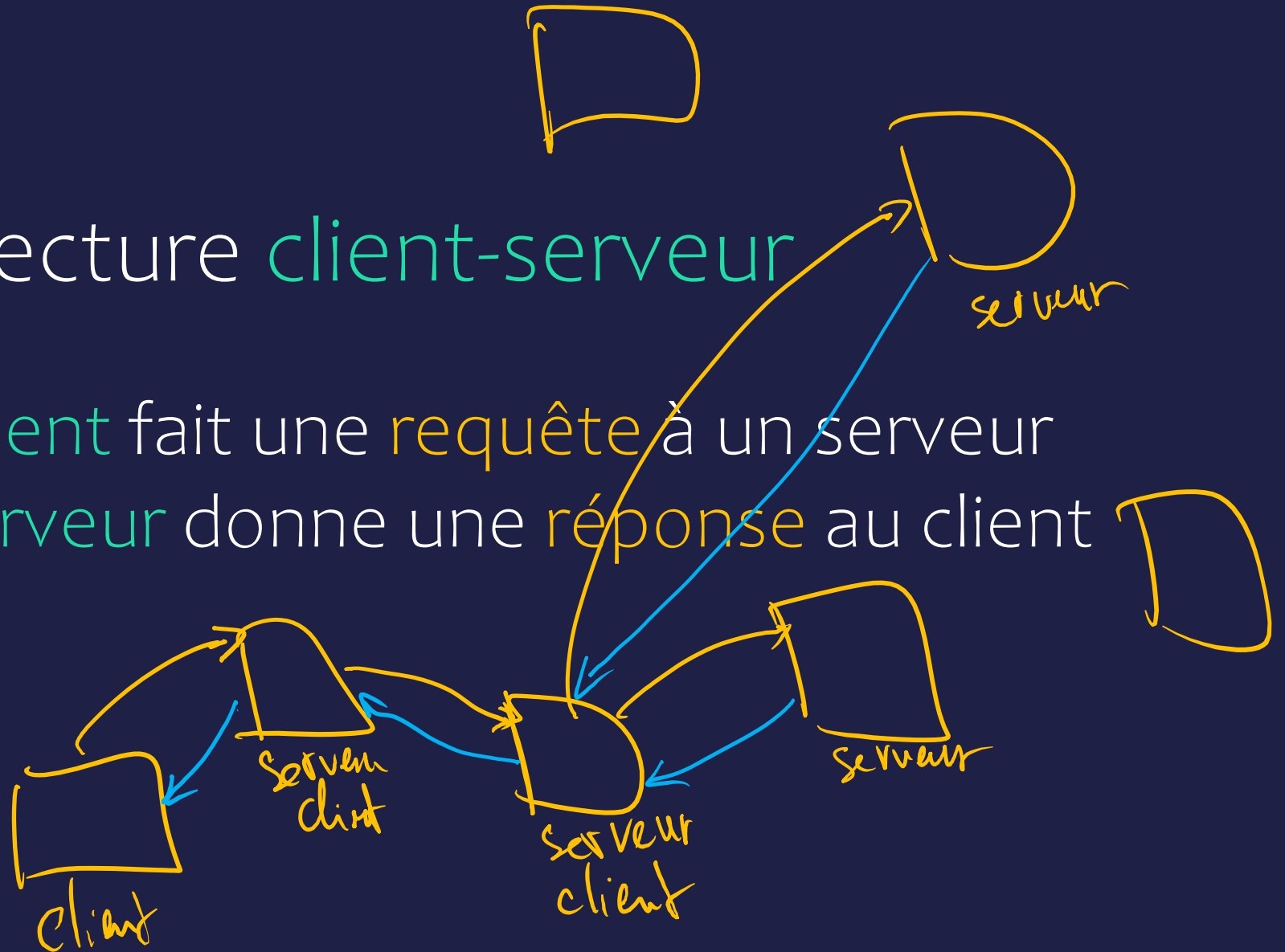
- la machine (identifiée par une adresse IP ou un hostname)
- le port (identifié par un numéro)

Architecture client-serveur

1. Un client fait une requête à un serveur
2. Le serveur donne une réponse au client

Architecture client-serveur

1. Un client fait une requête à un serveur
2. Le serveur donne une réponse au client



Cinématiques à connaître

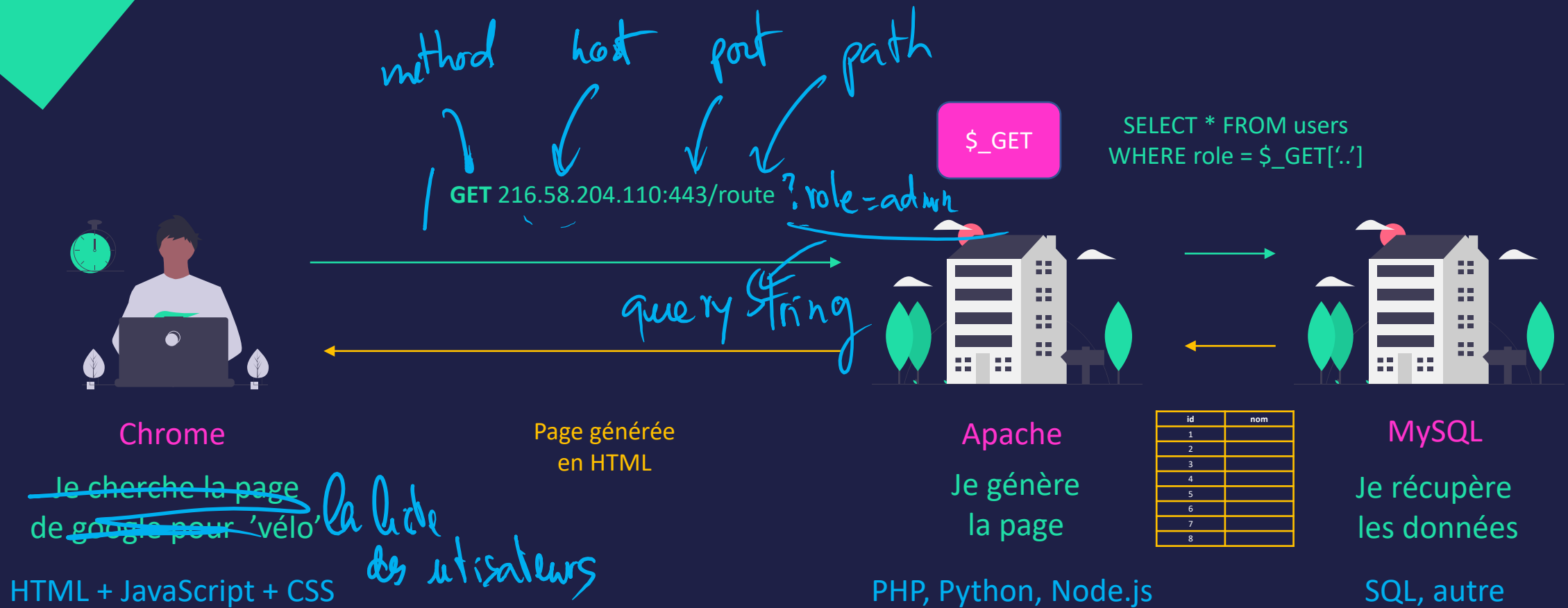
1. Affichage (*lecture*) de données de la DB
2. Enregistrement (*écriture*) de données dans la DB

Cinématiques à connaître

1. Affichage (*lecture*) de données de la DB
2. Enregistrement (*écriture*) de données dans la DB



1. Cinématique de lecture



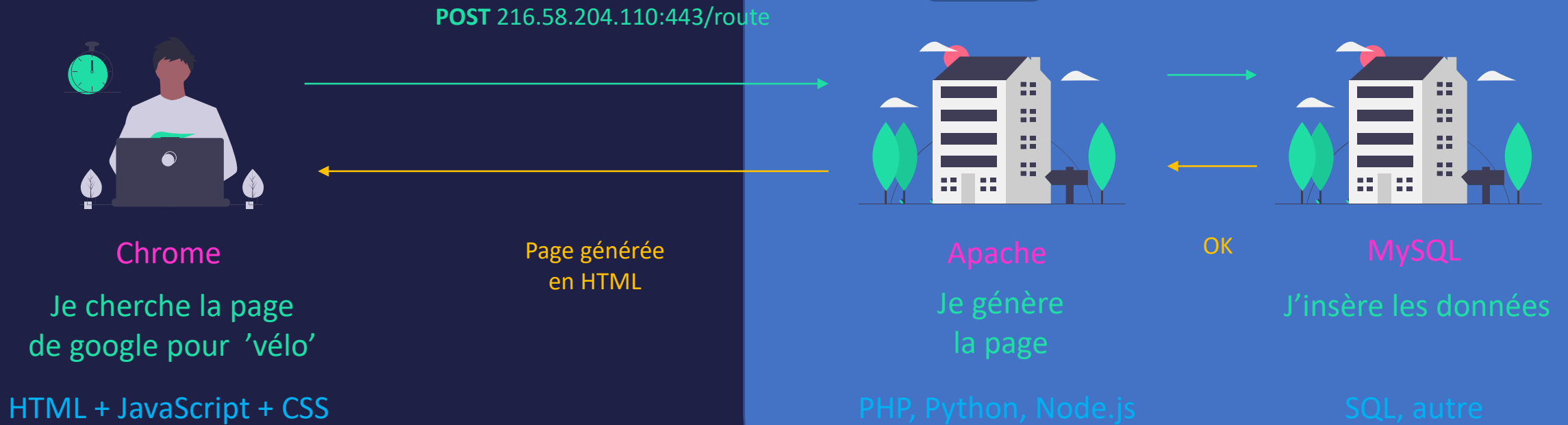
1. Cinématique de lecture



2. Cinématique d'écriture



2. Cinématique d'écriture



3. Le stockage de données (réutilisables)

Cookie

\$_COOKIE
setcookie

Session

\$_SESSION
session_start

DB

PDO
Mysqli
ORM (simpleORM,
Eloquent, Doctrine
...)



Chrome

Je cherche le site
super-site.com

HTML + JavaScript + CSS



Page HTML généré



Apache

Je génère
l'HTML

PHP



Cookie

`$_COOKIE`
setcookie

TEMPORAIRE !

Session

`$_SESSION`
session_start

DB



Chrome

Je cherche le site
super-site.com

HTML + JavaScript + CSS



Page HTML généré



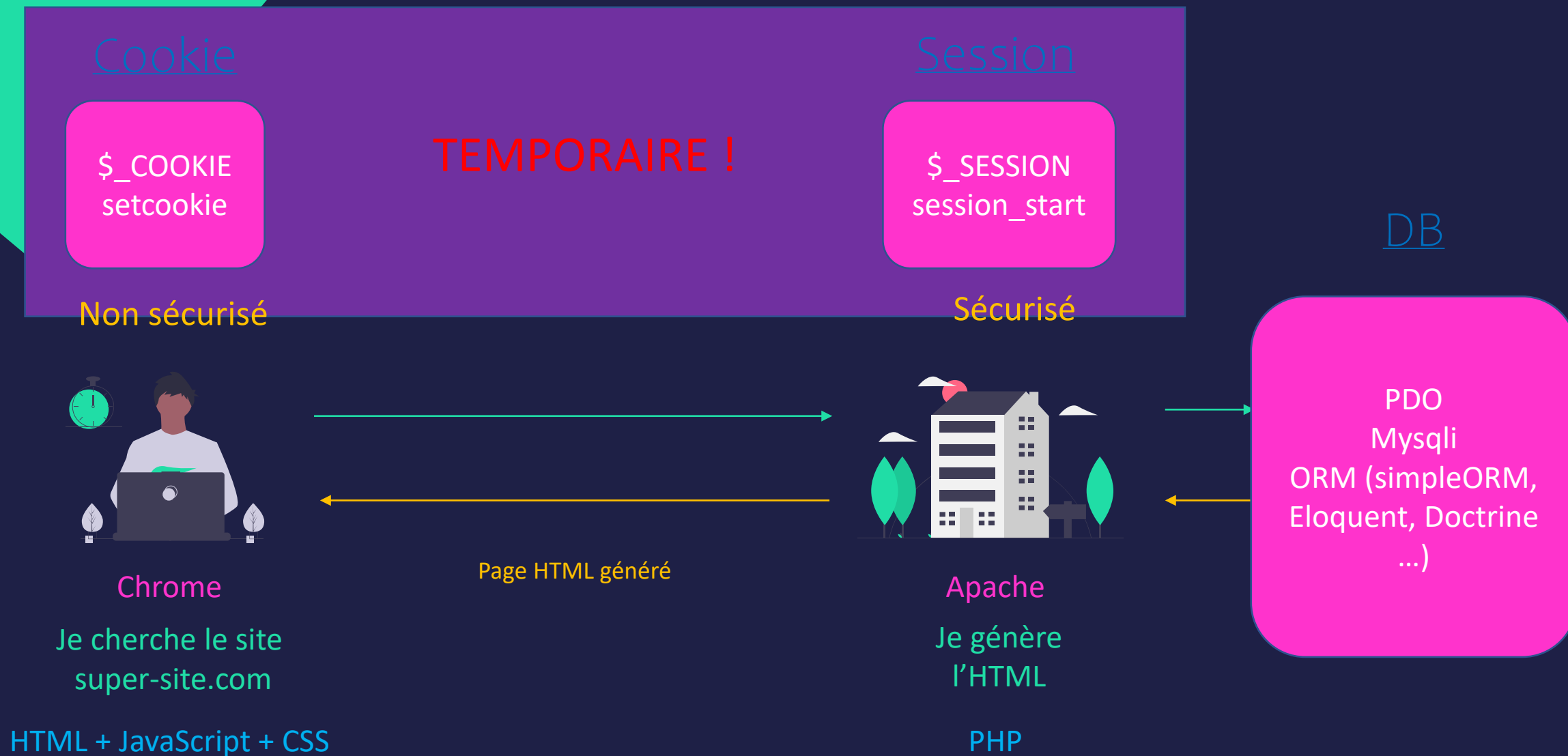
Apache

Je génère
l'HTML

PHP



PDO
Mysqli
ORM (simpleORM,
Eloquent, Doctrine
...)



COOKIE

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

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

Une **clé** est un **identifiant** pour retrouver la valeur stockée plus tard

Le **système de cookie** ne peut stocker que des chaînes de caractères (string)

Pour un **type complexe** :

! Pour le stocker, **json_encode**

! Pour le lire, **json_decode**

SESSION

```
session_start();
```

```
$_SESSION['cle'] = 'valeur';
```

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

```
session_destroy();
```

Une **clé** est un **identifiant** pour retrouver la valeur stockée plus tard

Le **système de session** peut **stocker** tout type de données

DB #1

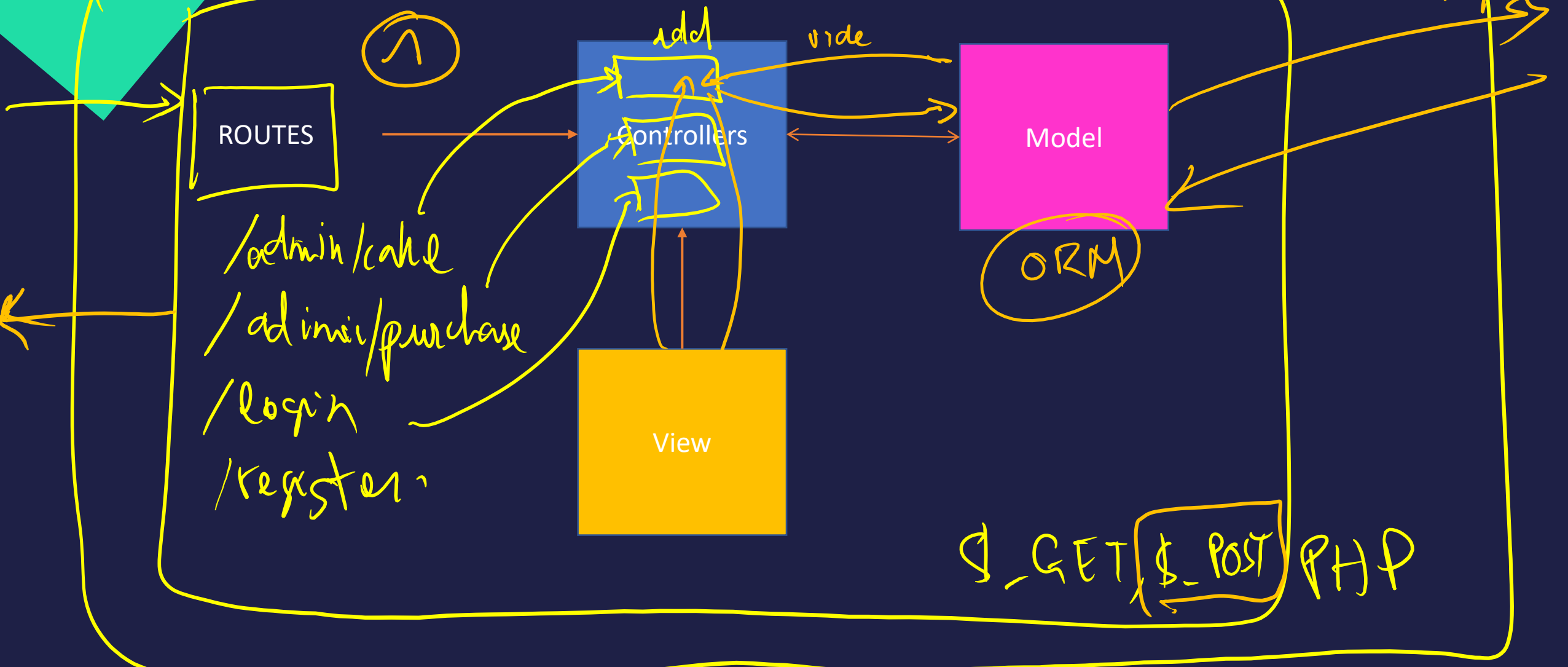
<https://github.com/noetix/Simple-ORM>

```
<?php  
  
include __DIR__.'/SimpleOrm.class.php';  
  
$conn = new mysqli('localhost', 'chuibho', 'suhimauche');  
SimpleOrm::useConnection($conn, 'bazedonais');
```

Un **ORM** (Object Relational Mapping) est un composant qui permet d'ajouter une **couche d'abstraction** sur la gestion d'une base de données relationnelle en **retranscrivant automatiquement les lignes en objets** et vice-versa

4. Architecture MVC

Architecture MVC



Couper son code en 3 parties

M, V et C

MVC :
Model
View
Controller

MVC :
Model = Entités

MVC :

Model = Entités

View = Code pour le visuel

MVC :

Model = Entités

View = Code pour le visuel

Controller = Le reste

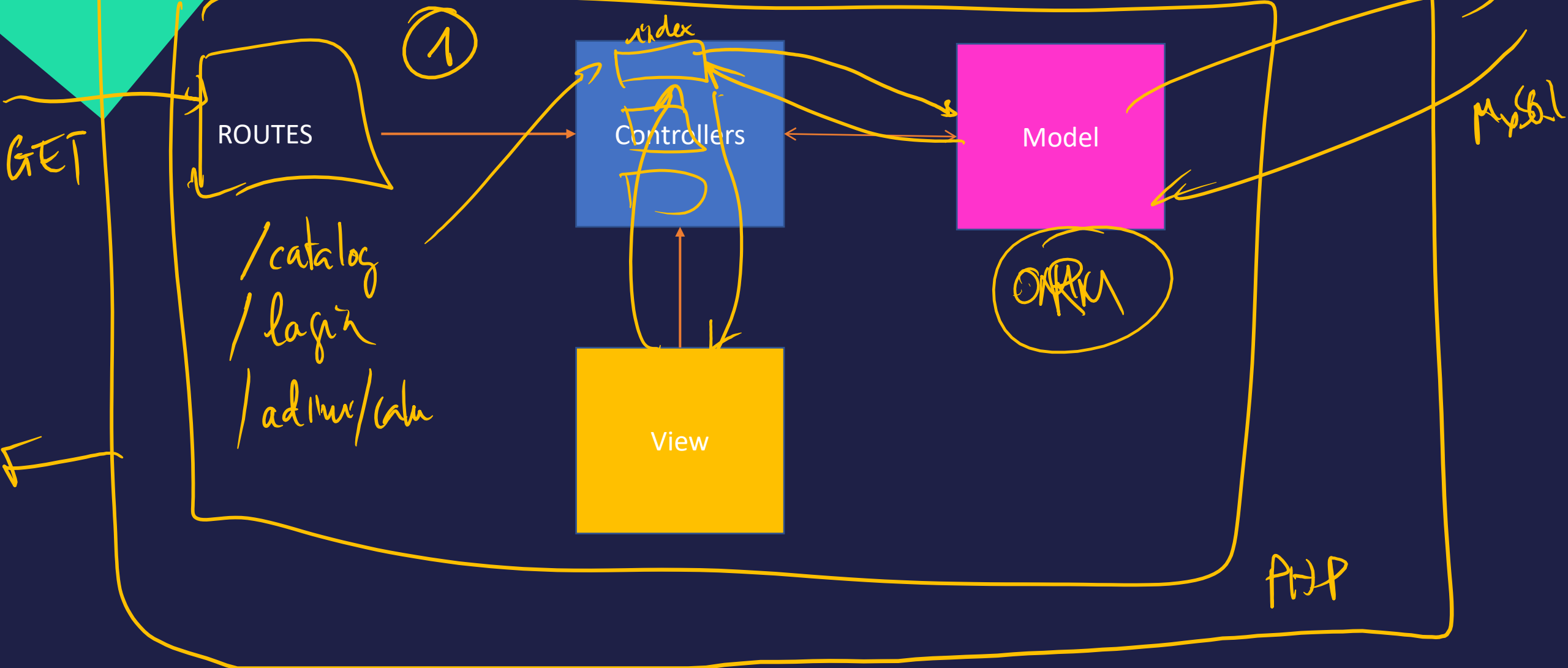
MVC :

Model = Entités

View = Code pour le visuel

Controller = Le reste

Architecture MVC



Architecture MVC

