

# Stratégie de sécurité d'une application



# Sommaire

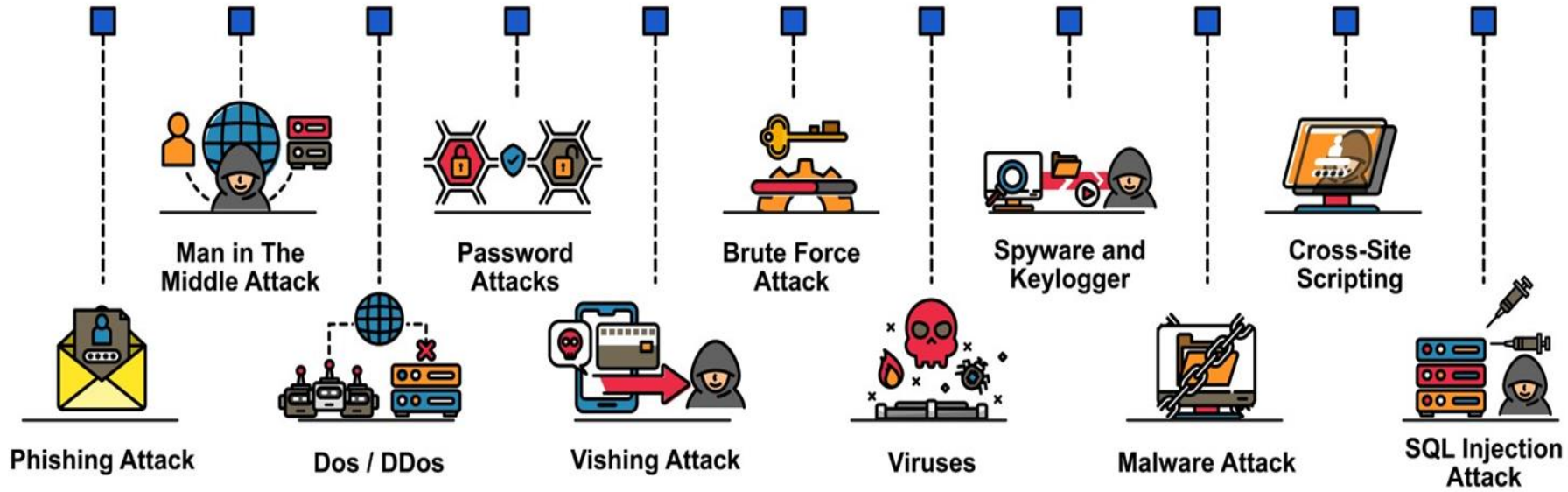
- .1. Pourquoi la API sécurité ?**
- .2. Types d'attaques**
- .3. Rappel des règles d'hygiène**
- .4. Stratégies de sécurité Web**
- .5. Authentification et Autorisation**
- .6. Recommandations pour les mots de passe**
- .7. HTTPS Protocol**
- .8. Sessions et Tokens**
- .9. API Stateful et API Stateless**

# Pourquoi la API sécurité ?



# Types d'attaques

## CYBER SECURITY ATTACKS

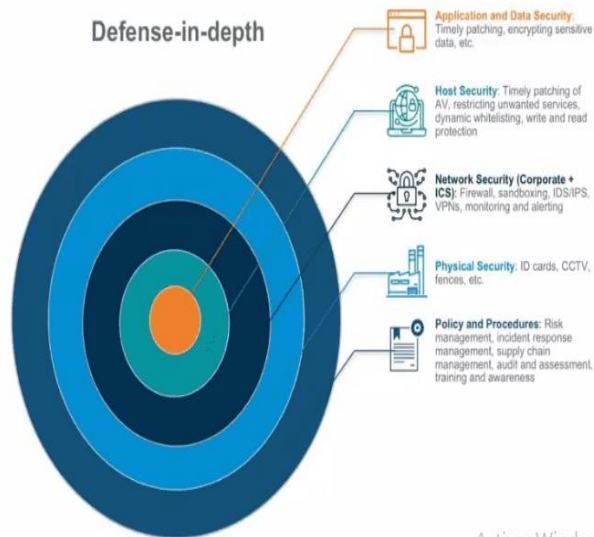


# Rappel des règles d'hygiène

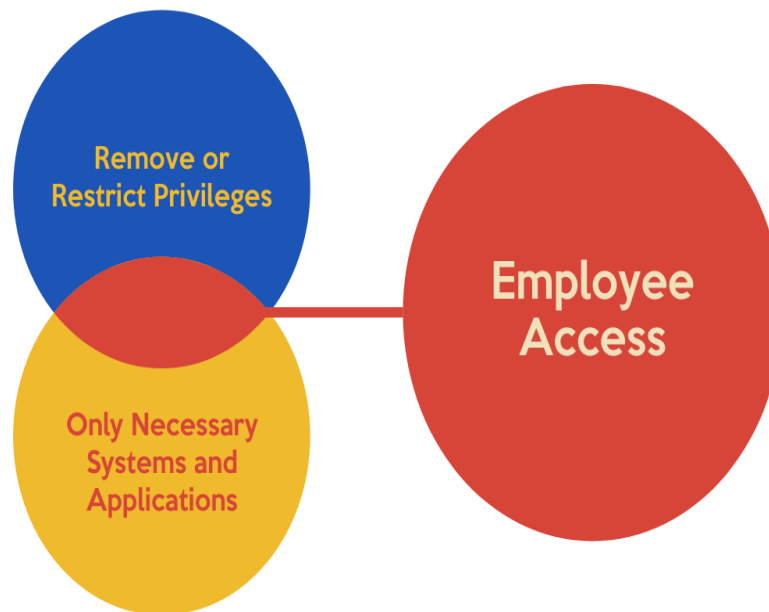
ICS/SCADA

Cybersécurité des systèmes industriels

## Défense en profondeur



## Principle of Least Privilege



# Rappel des règles d'hygiène

## RGPD

### PASSER À L'ACTION

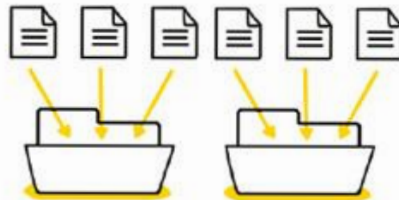
en 4 étapes

1



Constituez un registre  
de vos traitements de données

2



Faites le tri  
dans vos données

3



Respectez les droits  
des personnes

4



Sécurisez  
vos données

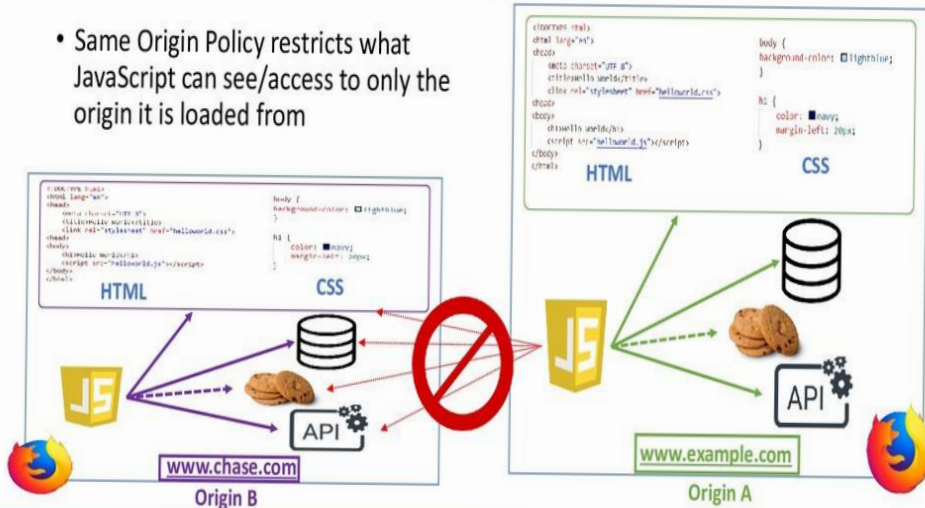


# Stratégies de sécurité Web

## Same Origin Policy (SOP)

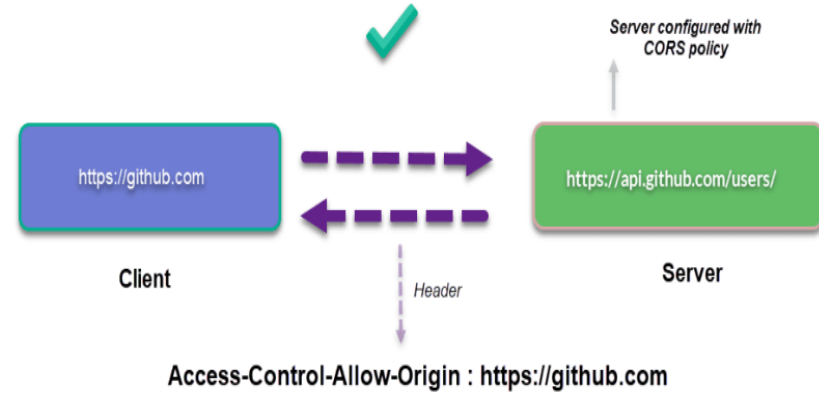
Without the SOP, the web would be a very, very dangerous place

- Same Origin Policy restricts what JavaScript can see/access to only the origin it is loaded from

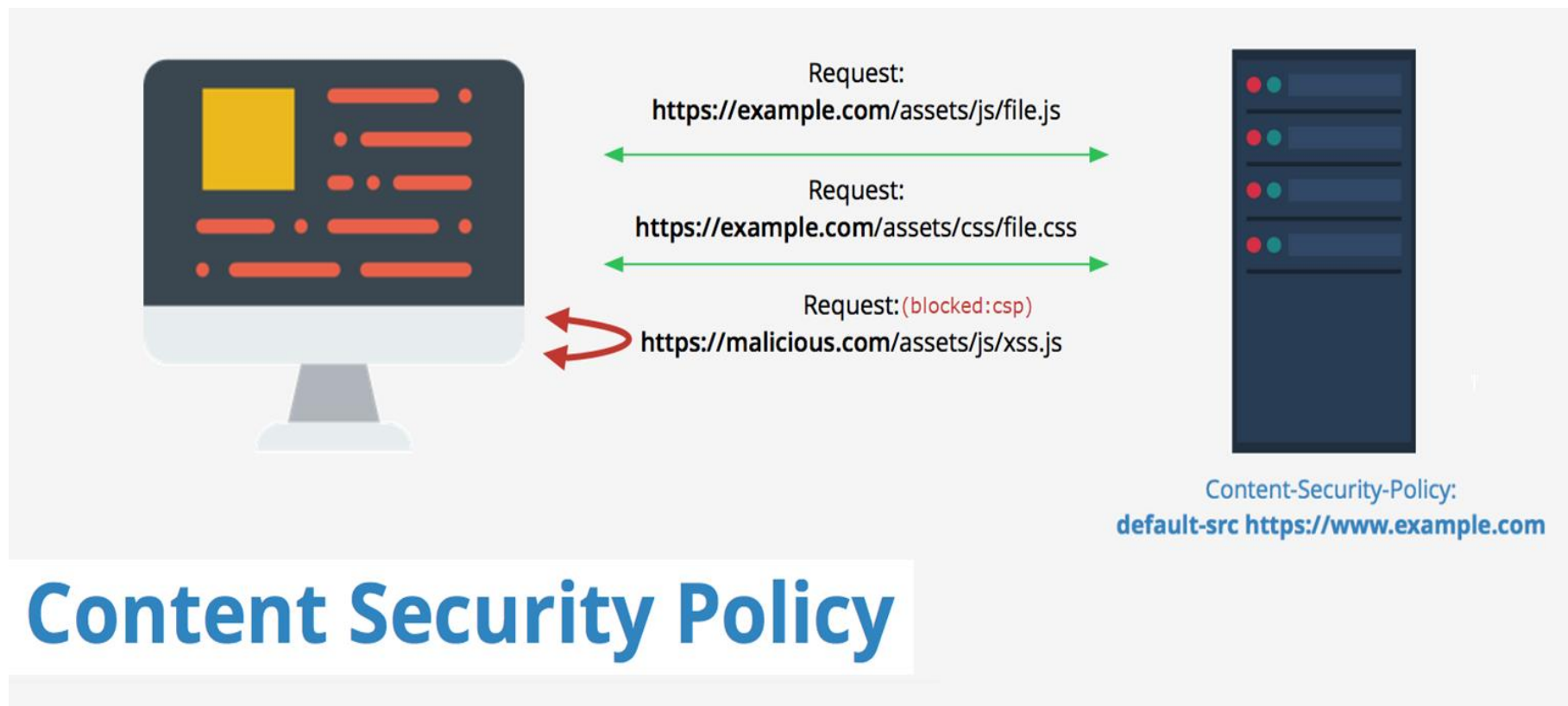


14

## How CORS Works ?



# Stratégies de sécurité Web



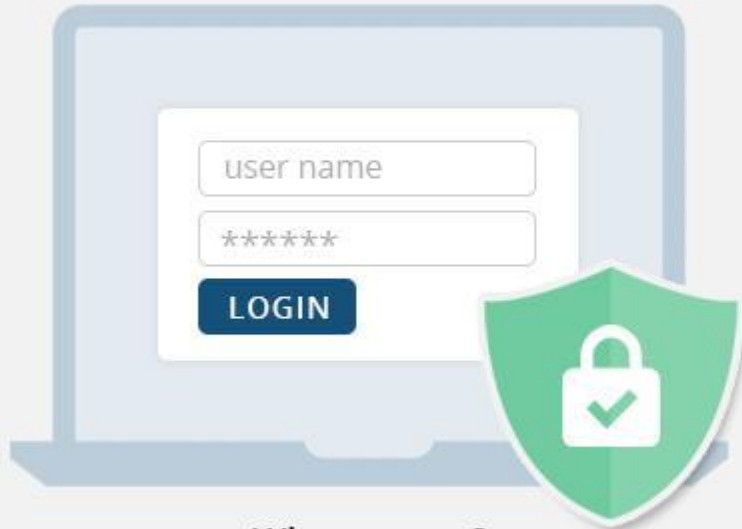


# Maîtrise de l'intégrité des ressources



# Authentication et Autorisation

## Authentication



**Who are you?**

Validate a system is accessing by the right person

## Authorization



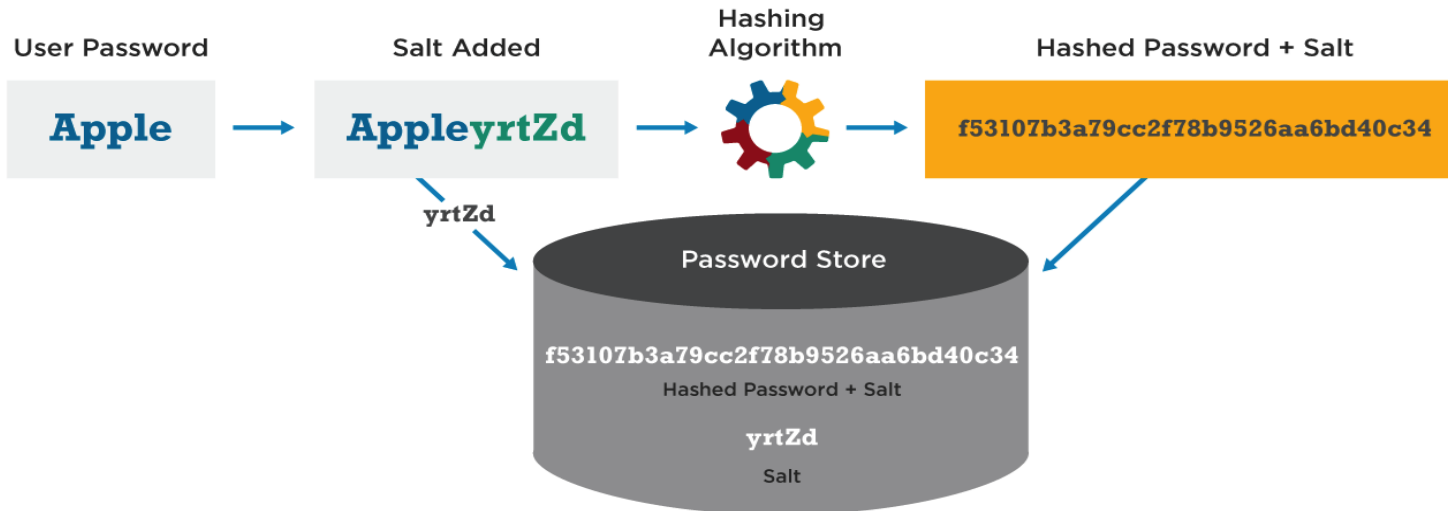
**Are you allowed to do that?**

Check users' permissions to access data

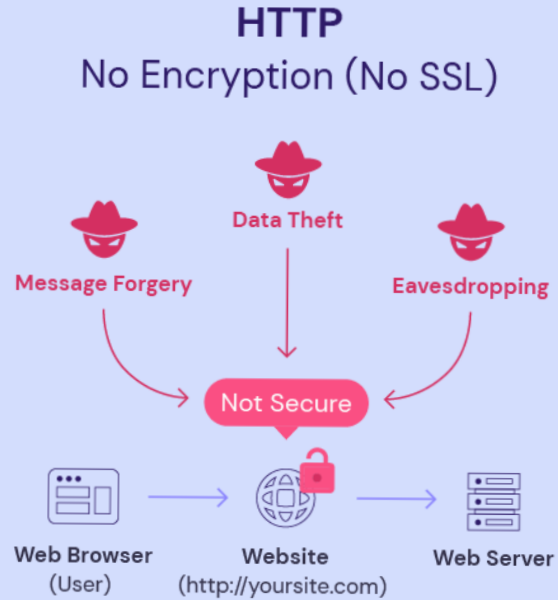
# Recommandations pour les mots de passe



## Password Hash Salting



# HTTPS Protocol



Data, such as a user password and a user ID, is **visible** to anyone.

VS



Data, such as a user password and a user ID, is **encrypted** to anyone

# Sessions et Tokens

## Session

The user sends login request

The server authorizes the login, sends a session to the database, and returns a cookie containing the session ID to the user



The user sends new request (with a cookie)



The server looks up in the database for the ID Found in the cookie, if the ID is found it sends the requested pages to the user

## Token

The user sends login request

The server authorizes the login and sends a token to the user

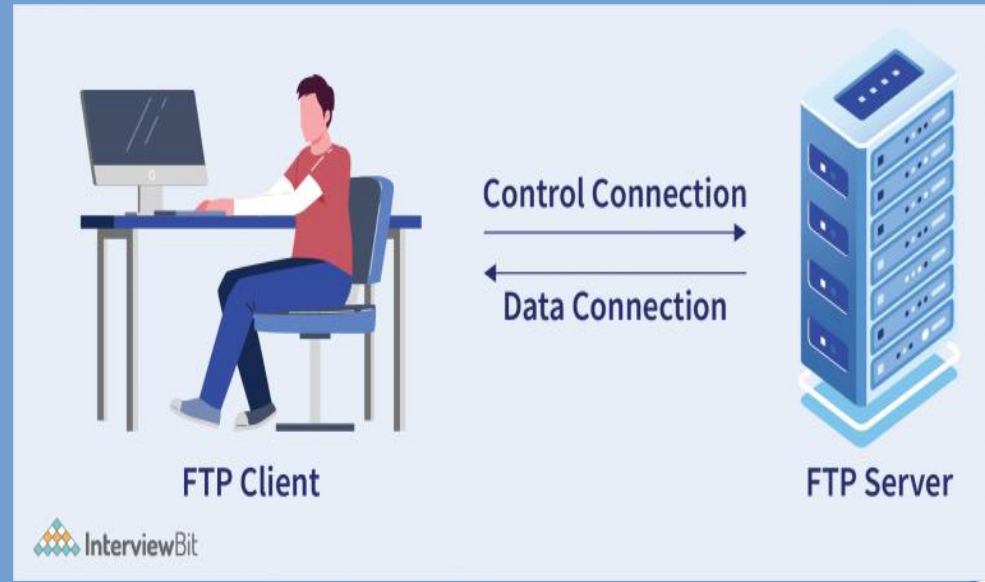


The user sends new request (with a token)

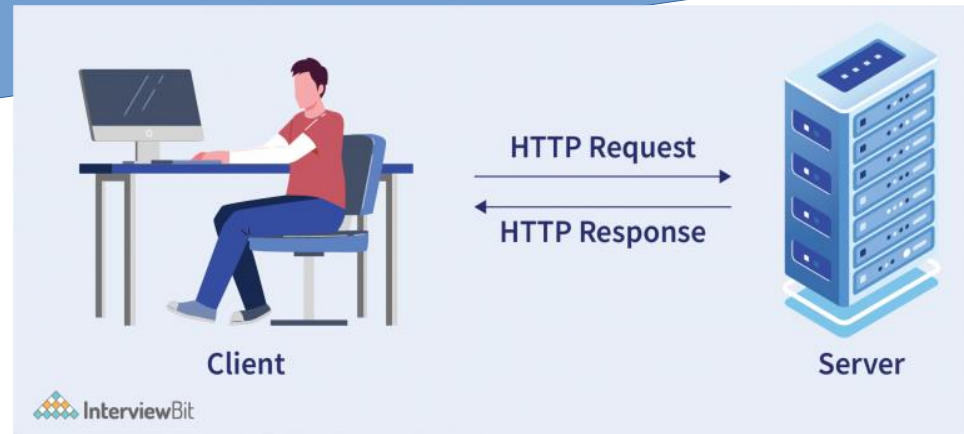


The server checks the token is valid and sends the requested pages to the user

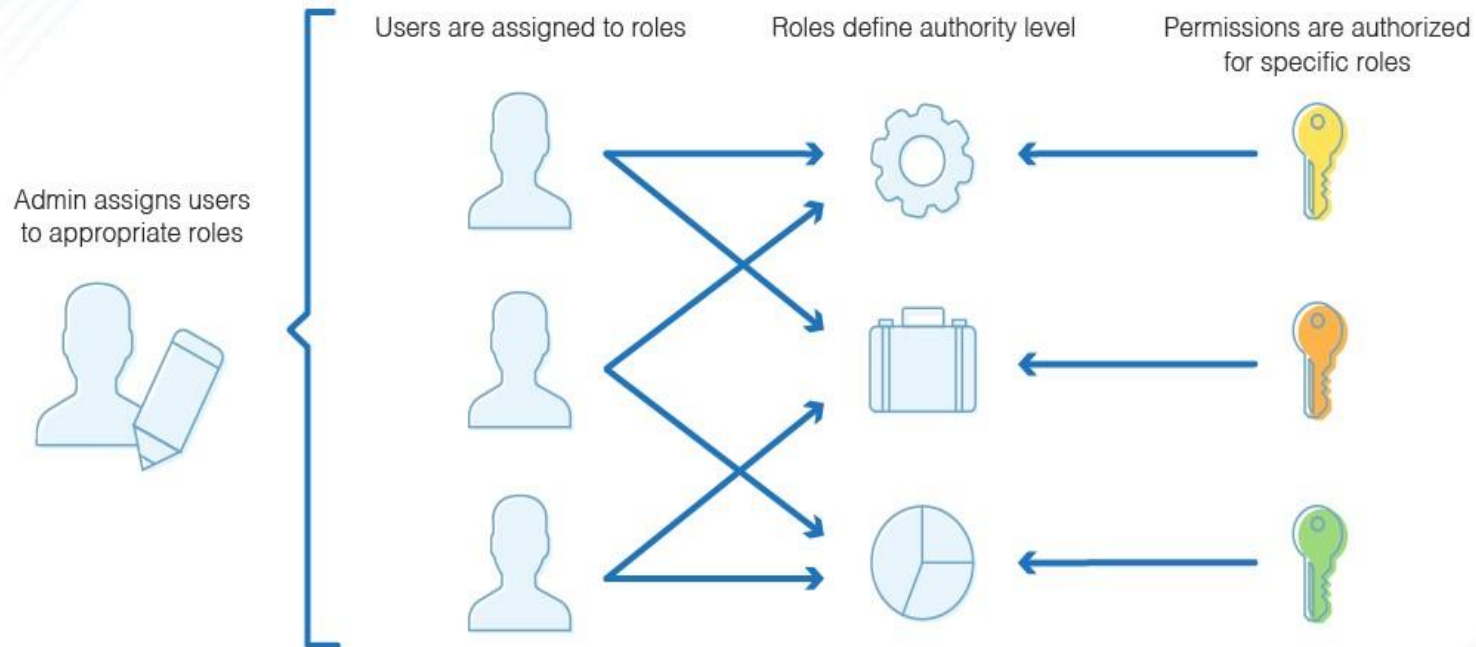
# API Stateful



# API Stateless

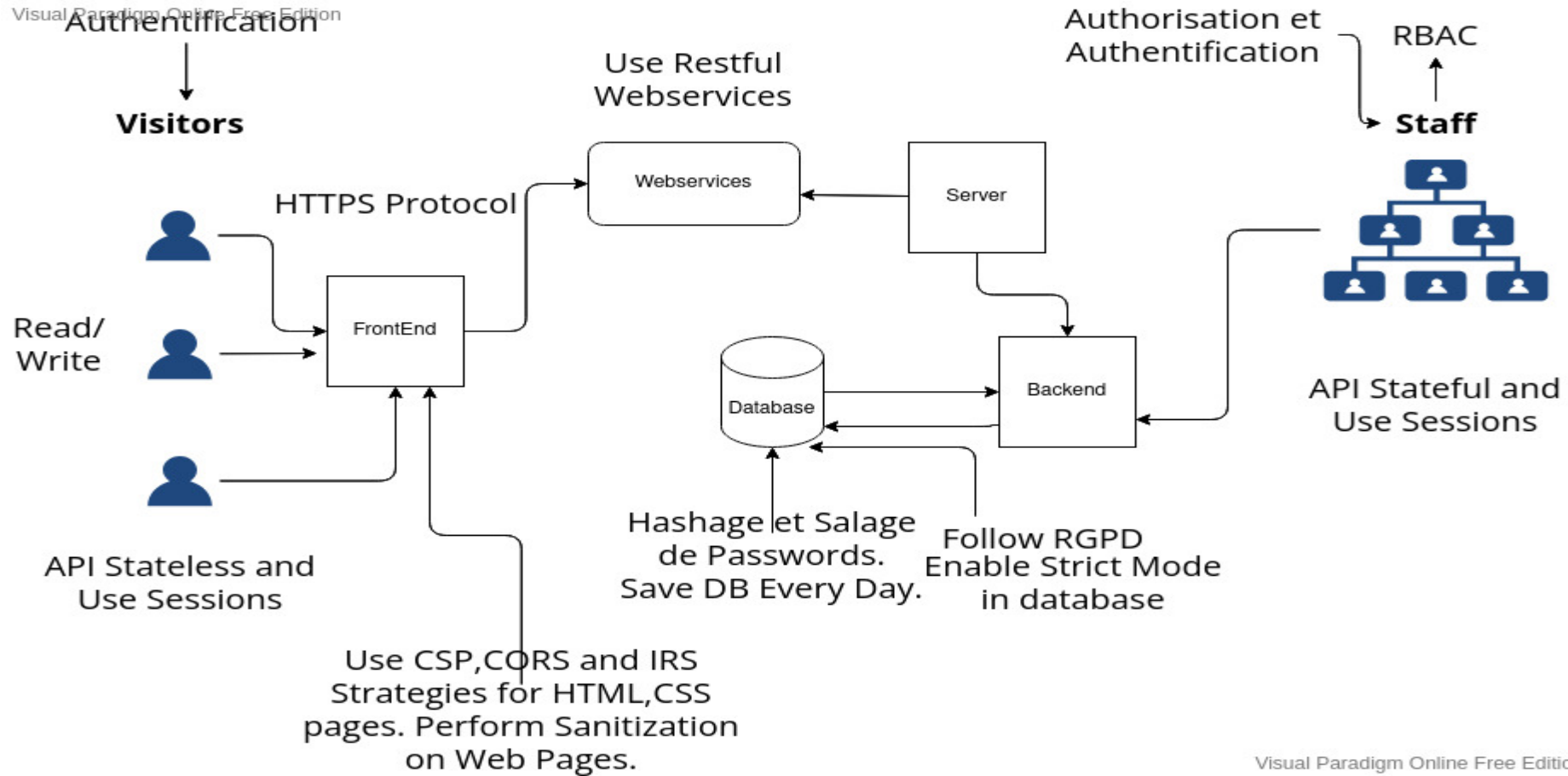


# Role-Based Access Control





# Proposition de stratégie de sécurité





*Thank you.*