

# Web Temps Réel

Amin NAIRI — ESGI

# Pré-requis

- Système d'exploitation Linux ou OS X
- Navigateur Web
- Éditeur de texte
- Connaissances en HTML
- Connaissances en CSS
- Connaissances en JavaScript
- Connaissances en Node.js
- Connaissances en Docker
- Connaissances en Docker Compose

# Qu'est-ce que HTTP ?

- Un protocole de communication
- Repose sur le protocole TCP
- Permet d'échanger des données
- Un client envoie une requête
- Un serveur envoie une réponse

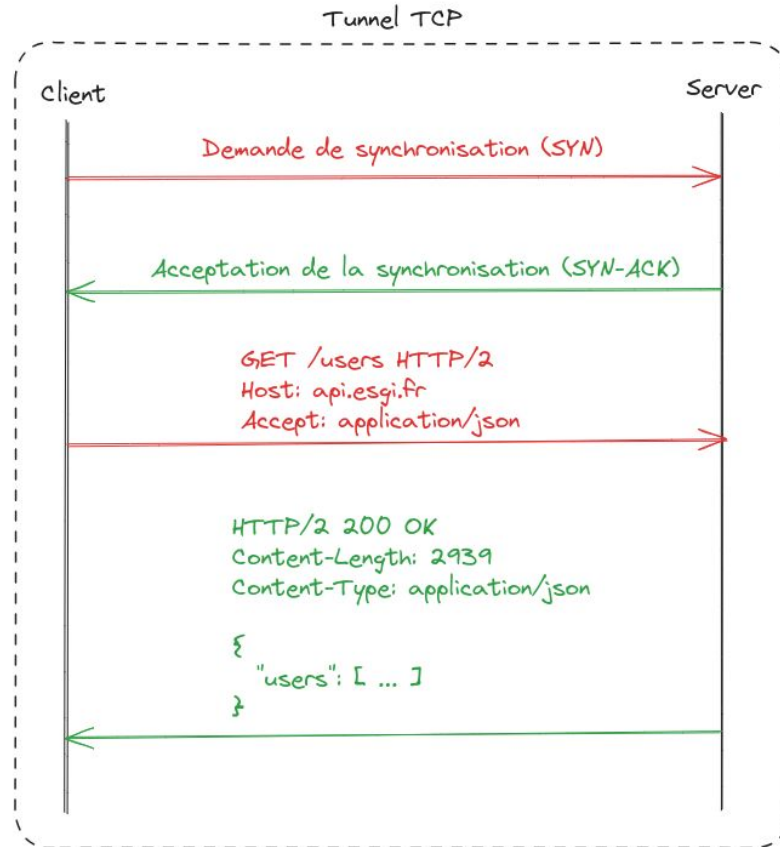


# Pourquoi utiliser le protocole HTTP ?

- Protocole simple
- Au cœur des communications
- Utilisé par le navigateur
- Utilisé par les applications natives
- Utilisé par les applications systèmes



# Comment fonctionne le protocole HTTP ?



# Comment fonctionne une communication HTTP ?

## REQUÊTE

```
POST /chat HTTP/2
Content-Type: application/json
Accept: application/json
Authorization: Bearer 12cf3hg1fc3hg1fc23

{
  "type": "private",
  "message": "Hello, world!"
}
```

## RÉPONSE

```
OK 200 HTTP/2
Content-Type: application/json
Content-Length: 1234

{
  "success": true
}
```

# Quelles sont les limites à l'utilisation du protocole HTTP ?

- Gourmand en ressource
- Plusieurs connexions à un même serveur
- Parfois impossible à paralléliser
- Pas sécurisé par défaut
- Pas de reconnexion automatique (TCP)



# Qu'est-ce que le Web Temps Réel ?

- Réaction à un événement instantané
- Moyen de communication
- Solution pour applications réactive
- Look & Feel Android & iOS





# Pourquoi faire du Web Temps Réel ?

- Ajouter de la communication synchrone
- Notification instantanées
- Mises à jour en temps réel
- Tout cas d'usage qui nécessite un retour synchrone à l'utilisateur
- Optimiser les ressources



# Comment faire du Web Temps Réel ?

- Avec du Polling
- Avec du Long Polling
- Avec des Server-Sent Events
- Avec le protocole WebSocket
- Avec un serveur Web
- Avec des librairies Node.js
- Avec l'API Web WebSocket
- Avec la librairie Node.js ws
- Avec la librairie Node.js socket.io



# Dans quels domaines est utilisé le temps réel ?

- Base de données
  - Rethink DB
  - Sapphire DB
  - Firebase Realtime Database
- Jeux-Vidéos
  - World of Warcraft
  - League of Legends
  - Half-Life 3
- Monitoring applicatif
  - Datadog
  - Firebase Crashlytics
  - Sentry
- Social
  - Discord
  - Slack
  - Teams

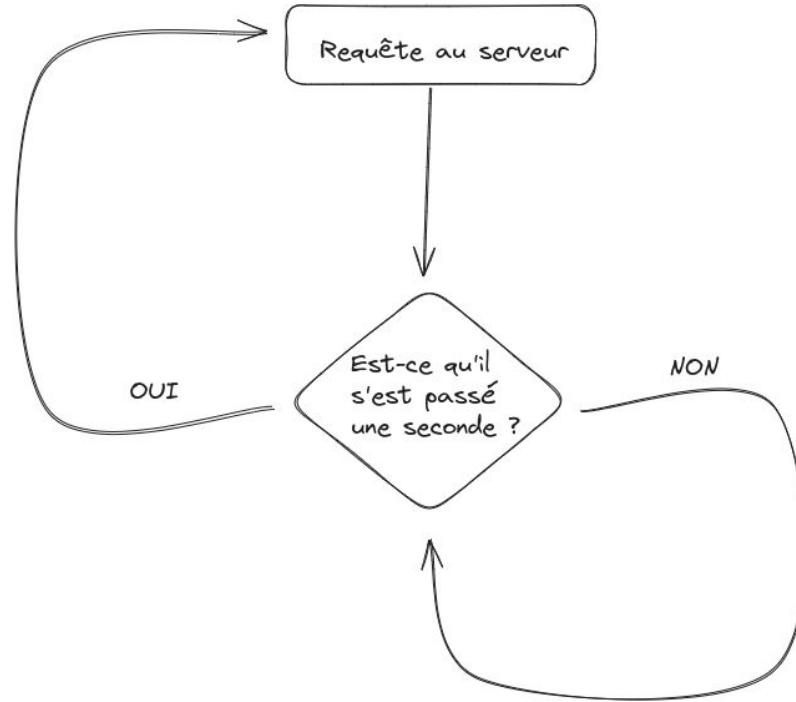


# Qu'est-ce que le Polling ?

- Stratégie de communication
- Pseudo-temps réel
- Repose sur le protocole HTTP
- Repose sur des intervalles fixes
- Décorrélacion envoi/réception de données



# Comment fonctionne le Polling ?



# Comment faire du Polling ?

```
const EACH_SECONDS = 1000
const usersEndpoint = "https://jsonplaceholder.typicode.com/users"

let users = []

const fetchAndRefreshUsers = () => {
  fetch(usersEndpoint).then(response => {
    return response.json()
  }).then(newUsers => {
    users = newUsers
  })
}

setInterval(fetchAndRefreshUsers, EACH_SECONDS)
```

# Exercice #1

- Créer un serveur qui envoie une liste d'utilisateurs
- La liste d'utilisateurs doit être mise à jour toutes les 5 secondes
- Créer un client qui récupère la liste d'utilisateur et l'affiche
- La liste doit être rafraîchie toutes les secondes
- Un utilisateur a un identifiant et une adresse email