REUNION DU 06/04/2024

Contraintes fichier :

* Question de dimensionnement ?
* Combien de serveurs ?
* Prendre en compte le temps de chargement des fichiers 🡪 Fenêtre d’anticipation de l’artiste

Protocole :

* Gestion d’erreurs (ex : Safe points ?)
* Communication App/serveurs 🡪 Apache ? Node.js ?

Contraintes interface :

* Adaptation Android/IOS
* Authentification pour le chef d’orchestre 🡪 hiérarchie (chef d’orchestre/ Performer/ Public) ?
* Si on passe par une page web : Imposer un navigateur pour tout le monde !!

Contraintes synchronisation :

* On estime qu’un délai < 300 ms suffit. 🡪 (<https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API>) => ce document explique la Audio API que nous pouvons utiliser dans notre site : *“Timing is controlled with high precision and low latency, allowing developers to write code that responds accurately to events”* Il sera donc possible d’avoir un temps acceptable de latence lorsqu’on lancera le fichier audio.
* Tous les téléphones doivent être sur la même horaire GPS
* Pour synchroniser les performer 🡪 indication (ex : décompte « 3, 2, 1… »)

Contrainte lecture :

* Comment faire de la lecture en background sur un site (voir sites de radio) ? 🡪 utiliser des « service workers » qui permettent l’execution de tâche en background sur un thread différent du script js (voir documentation) : https://developer.mozilla.org/en-US/docs/Web/Progressive\_web\_apps/Guides/Offline\_and\_background\_operation
* Comment lancer un fichier audio à un temps donné ? 🡪 la fonction classique setTimeout() qui exécute après un temps donné : (https://developer.mozilla.org/en-US/docs/Web/API/setTimeout)
* De quoi se composent les fichiers audios ? (Question à poser à l’artiste)