



pSensei

10ème dan de la productivité



**v1:**

Notre première idée était d'avoir un assistant de travail qui récompense en fonction de la concentration de l'utilisateur.

On utilise un arduino qui detecte la présence de l'utilisateur pour distribuer une récompense en fonction du temps de travail effectué. La récompense peut être un café, une clope, un cookie etc, les possibilités sont infinies.

**v2:**

Notre deuxième idée était de créer une sorte de focus app (inspi forest/zen mode op) qui permet de gérer/calculer son temps de travail. Et nous voulions ensuite pouvoir display des stats en fonction du temps passé sur telle ou telle tâche.



**v1:**

Notre première idée était d'avoir un assistant de travail qui récompense en fonction de la concentration de l'utilisateur.

On utilise un arduino qui detecte la présence de l'utilisateur pour distribuer une récompense en fonction du temps de travail effectué. La récompense peut être un café, une clope, un cookie etc, les possibilités sont infinies.

**v2:**

Notre deuxième idée était de créer une sorte de focus app (inspi forest/zen mode op) qui permet de gérer/calculer son temps de travail. Et nous voulions ensuite pouvoir display des stats en fonction du temps passé sur telle ou telle tâche.



Ce genre d'application existant déjà nous avons décidé de nous concentrer sur le point commun de ces deux idées : **l'affichage des stats**. Créer quelque chose à la fois utile mais aussi ludique dans la façon dont ces stats et explications sont affichées pour l'utilisateur. (inspi temps d'écran ios)



pSensei est un assistant de conseils de productivité  
basé sur ton activité en temps réel.



Le programme **pSensei** récolte et analyse votre **activité** en temps réel sur votre ordinateur. Il agrège ensuite ces données sur un **tableau de bord** pour donner à l'utilisateur une vue d'ensemble.

Une **formule** basée sur ces statistiques dont seul pSensei en a le secret est ensuite calculée. Cela donne une valeur centrale qui permet à l'utilisateur de connaître son niveau de productivité par rapport à son **objectif**.

Enfin des **conseils** sont donnés par rapports à certains chiffres clés que pSensei juge **révélateur**.



## Gontran

26 ans

Developpeur

Paris

Optimisateur infini

Travail souvent en  
télé-travail

### Objectifs

-connaître son activité sur son ordianteur

-améliorer sa productivité

-obtenir des conseils basé sur son utilisation

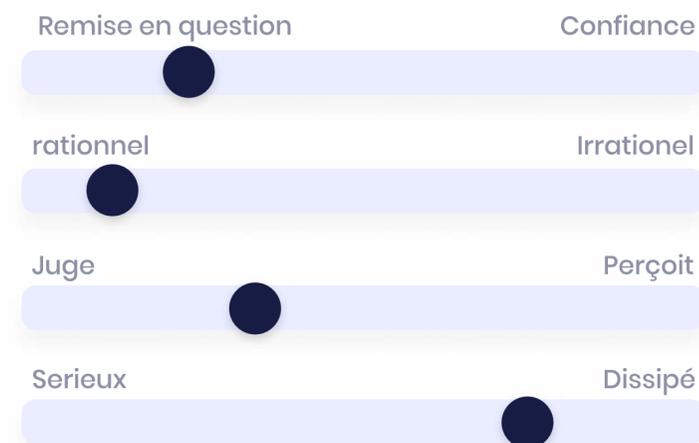
### Frustrations

-a le sentiment de perdre du temps

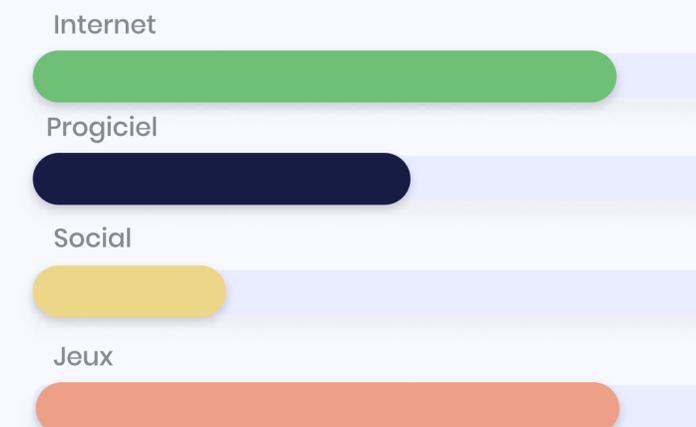
-n'a pas de recul sur son activité globale

-n'arrive pas à s'allouer des bons temps de pause

### Personnalité



### Utilisation ordinateur

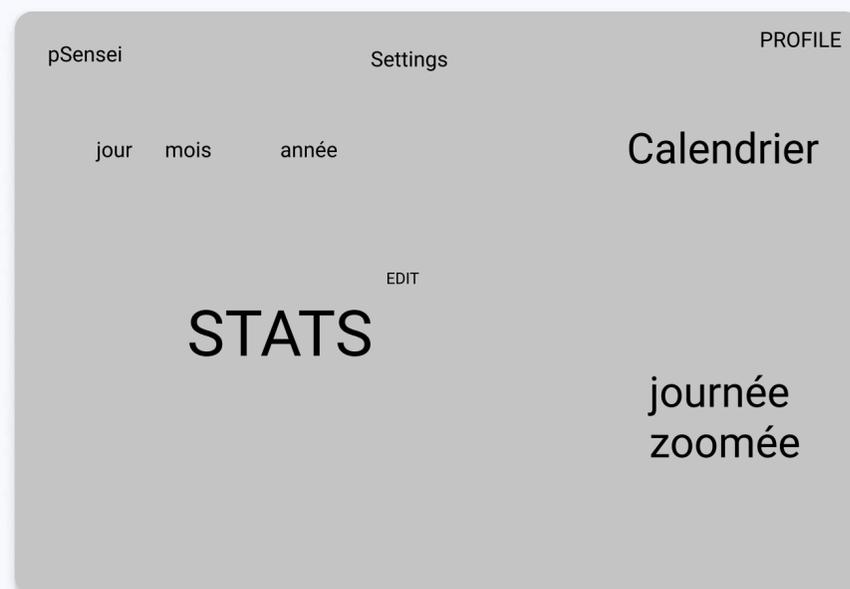




- un tableau de bord qui concentre les données brutes en carte de données pertinentes et facilement appréhendables au premier coup d'oeil.
- Des graphiques et des diagrammes pour comprendre plus facilement les chiffres non parlants.
- Des conseils sur un ton ironique afin de ne pas décourager ou culpabiliser l'utilisateur mais plutôt l'encourager.
- Une formule calculée à partir des différentes statistiques agrégées pour arriver à se situer par rapport à son objectif défini de productivité.
- Un "zanshin mode" qui est un seuil de productivité optimale dans lequel le logiciel change complètement d'apparence afin de motiver l'utilisateur à obtenir cet environnement.
- Un récap de la journée, du mois et de l'année afin d'avoir du recul et un conseil général de pSensei.
- Un calendrier du mois afin d'avoir une vue d'ensemble de ses performances et ses "séries de productivité"



## Premier brouillon d'organisation de l'information



Permet de définir rapidement les besoins en tant que designer et/ou développeur.

Permet de se rendre compte au premier coup d'oeil du layout à adopter pour une tel organisation.

## Brouillons du layout UI



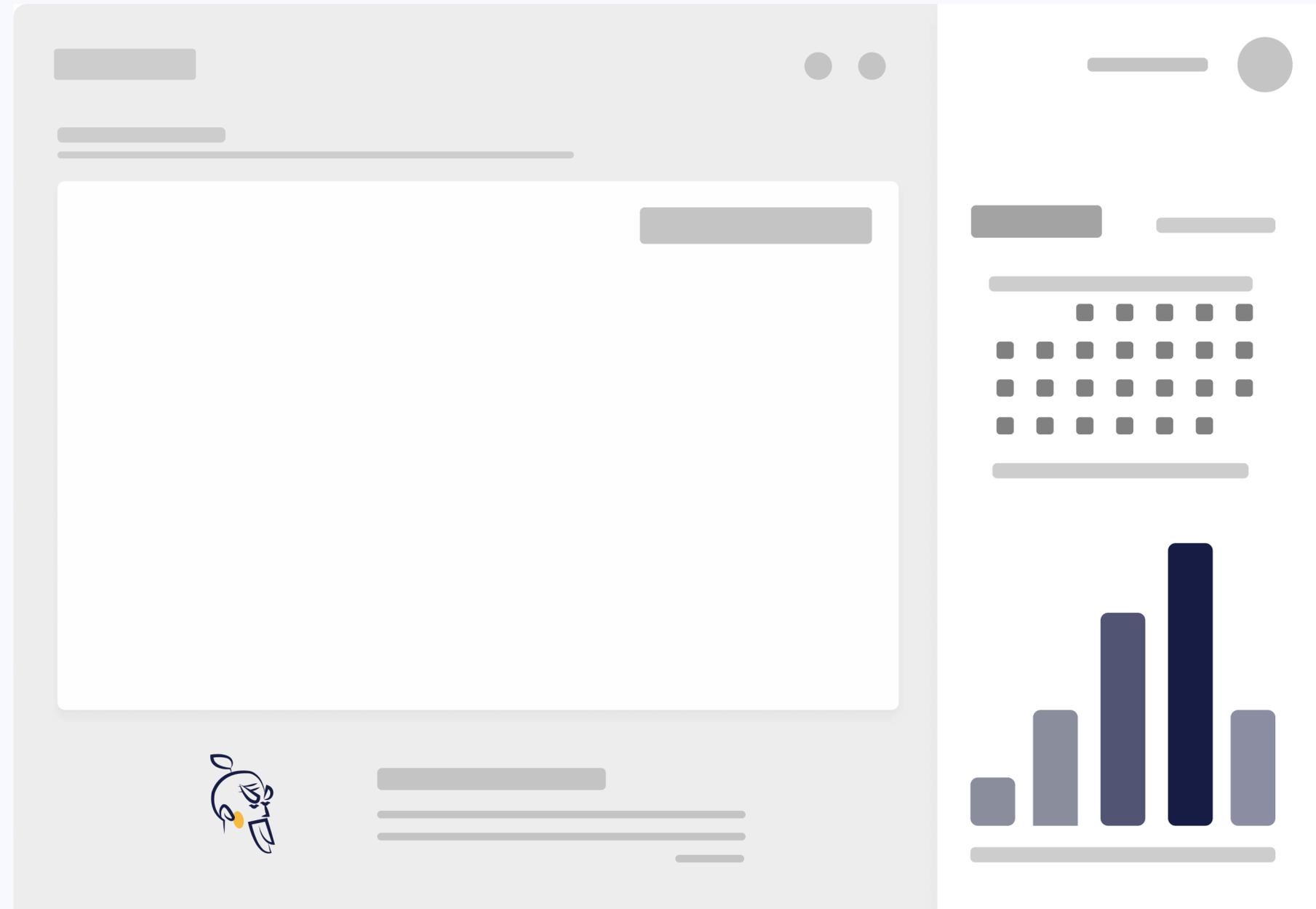
Premier itération lors de la définition de notre layout global, cette démarche nous a permis des réflexions plus simple par la suite pour les choix UI / UX de notre app.



Layout intermediaire, resultant de la mutation du premier layout conçu, ici on s'approche plus du layout de notre produit final, il remplit déjà nos conditions et nos besoins.



## Layout définitif





## Typo

Titre - H1

Titre - H3

Titre - H5

Corps

Corps medium

## Couleurs

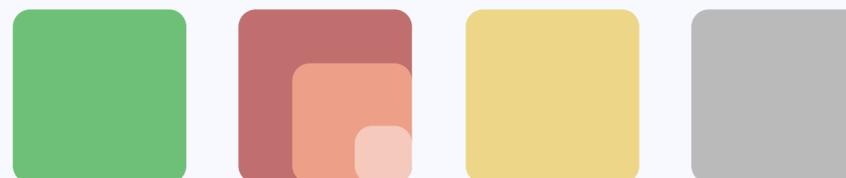
Principale + gradation



## Boutons



Valeurs



Text







pSensei

75%  
zanshin mode

Profil

## Mon Dashboard

Journée
Mois
Année

**Productivité**  
selon le type de logiciel actif

● Productivité+ ● Gaming+ ● Médias+

**Intérêt/productivité**  
En fonction des soft

20 054  
Nombre de clics

**Optimisation**  
de votre productivité par rapport aux autres utilisateurs

**Nuage de mots**  
mots représentant ton activité

**Temps de réaction aux notifications**  
Différence de temps entre réception et ouverture de la notification

**APM (actions par minute)**

**Objectif de productivité**  
11 minutes restantes

+3

**Pertinence**  
onglet/type de travail

-35%

26/11

**Tu as travaillé efficacement 5h25 hier !**

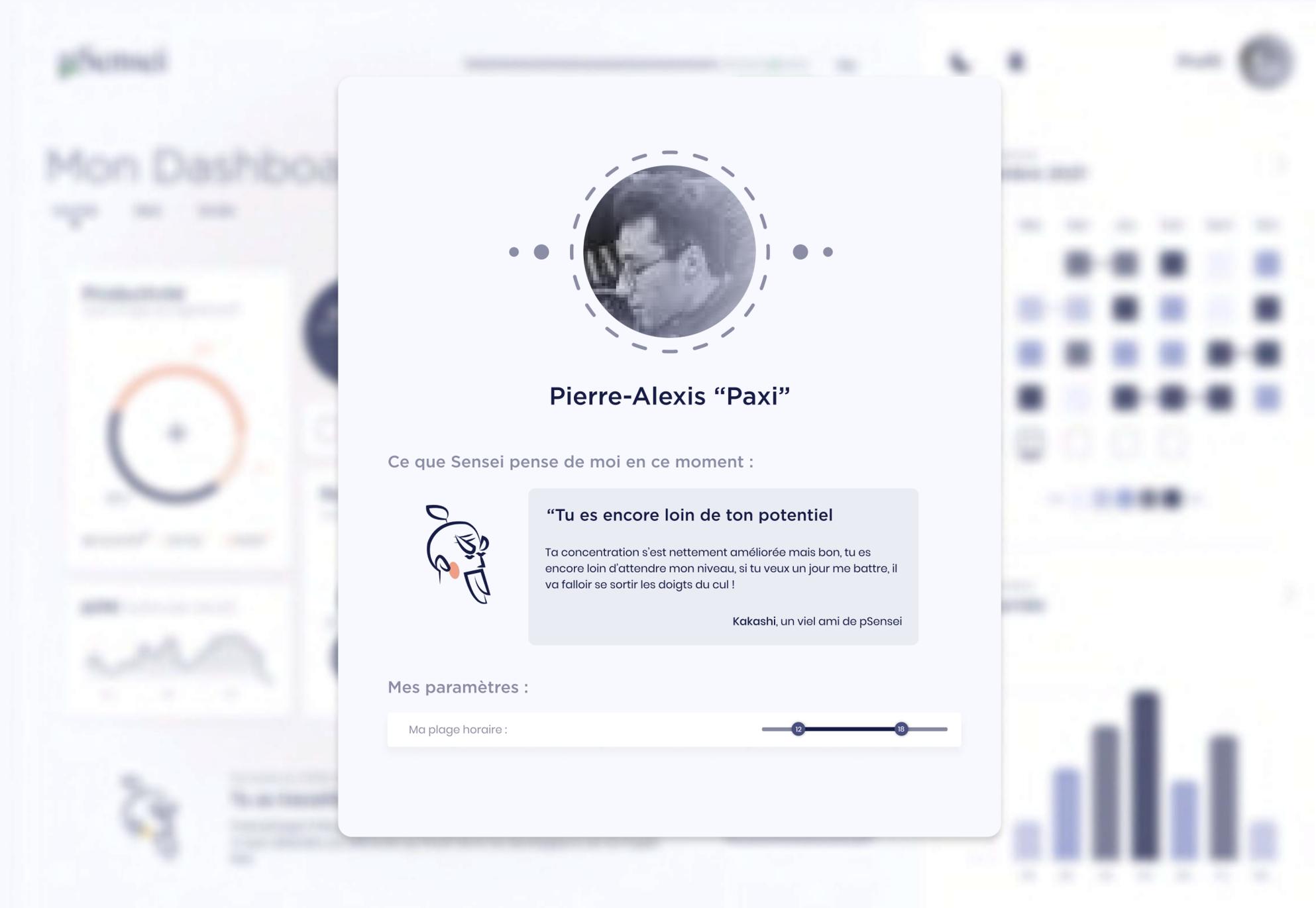
C'est presque 3 fois plus qu'avant hier ! Continue sur cette voie si tu veux un jour atteindre une efficacité qui ferait frémir les développeurs de CD Projekt Red.

Recap journée

Activité globale  
**Septembre 2021**

Loss  More

Concentration  
**Ma journée**





## :Stack dev:



### :FrontEnd:

#### React

- Design basé sur des composants.
- Single page App.
- Soft logiciel, et non web app.

#### Redux

- Data partagé au 4 coins de l'app.
- Aggregation de data venant de plusieurs composants.
- Utilisation des reducers et dispatchers.

#### Electron

- Windows, MacOS, Linux app, besoin d'être run en local.
- Besoin de permission pour read toutes sortes de data lors de l'utilisation de l'ordi

#### gsap / css-transition-group

- Interaction léchée pour s'inscrire dans cette nouvelle génération de webapp native comme discord, ou figma.



### :BackEnd:

#### Firestore

- Scalable & Simple
- Profile et settings de profil save dans un Firestore
- Pas besoin de DB, on ne souhaite pas stocker la data analysé, donc elle sera en local.

#### Blackbone

- ??.
- Memory reading pour pouvoir aller recuperere les differentes actions réalisé par notre utilisateur.

#### Python app

- En local, au launch, notre app reliera blackbone a une architecture de script python qui s'occuperont de compute les données reçus.
- Envoi des donnée qui calculé sous forme de JSON à l'app electron, React rerender et gère ce que le user voit à chaque update des données.



**MERCI !**