

AREA

let's get started

Plan

Introduction

Fonctionnalités & Services

Outils d'organisation

Organisation & Stack

Conclusion

27/11/24 · Défense 1

Introduction

Le projet AREA(Action-REAction) vise à développer une plateforme similaire à IFTTT et Zapier. Cette plateforme permettra aux utilisateurs d'accéder à des services en s'y connectant. Ils pourront ensuite exécuter des actions automatisées lorsqu'une condition est remplie. Par exemple, envoyer un mail à l'utilisateur lorsqu'il ajoute une tâche dans son google agenda.

Fonctionnalités

- authentication: classique(mail et mot de passe), avancée(proposition d'un service tier tel que google);
- enregistrement utilisateurs/inscription(via un formulaire et l'utilisation de services tiers comme google);
- Autorisation pour éviter que les simples utilisateurs accèdent à la section administration.
- gestion du profil utilisateur: l'utilisateur peut voir et modifier ses informations, il peut aussi supprimer son compte;
- services que proposent la plateforme: à choisir(07 services);
- Action + triggers + réaction à chaque service.
- Compilation via DOCKERFILE

Services

- Gmail API: Lire, envoyer et organiser les emails gmail.
- Dropbox Developers: Synchroniser et partager des fichiers.
- Google Calendar API: Créer, modifier ou lire des événements.
- Trello Developer API: Ajouter des cartes, mettre à jour des listes.
- Spotify API: Interagir avec des données musicales.
- Calendarific API: Surveiller les jours fériés et événements mondiaux.
- Google Drive API: Télécharger, supprimer et partager des fichiers.

Organisation

Quels sont les outils d'organisation dont nous disposons?

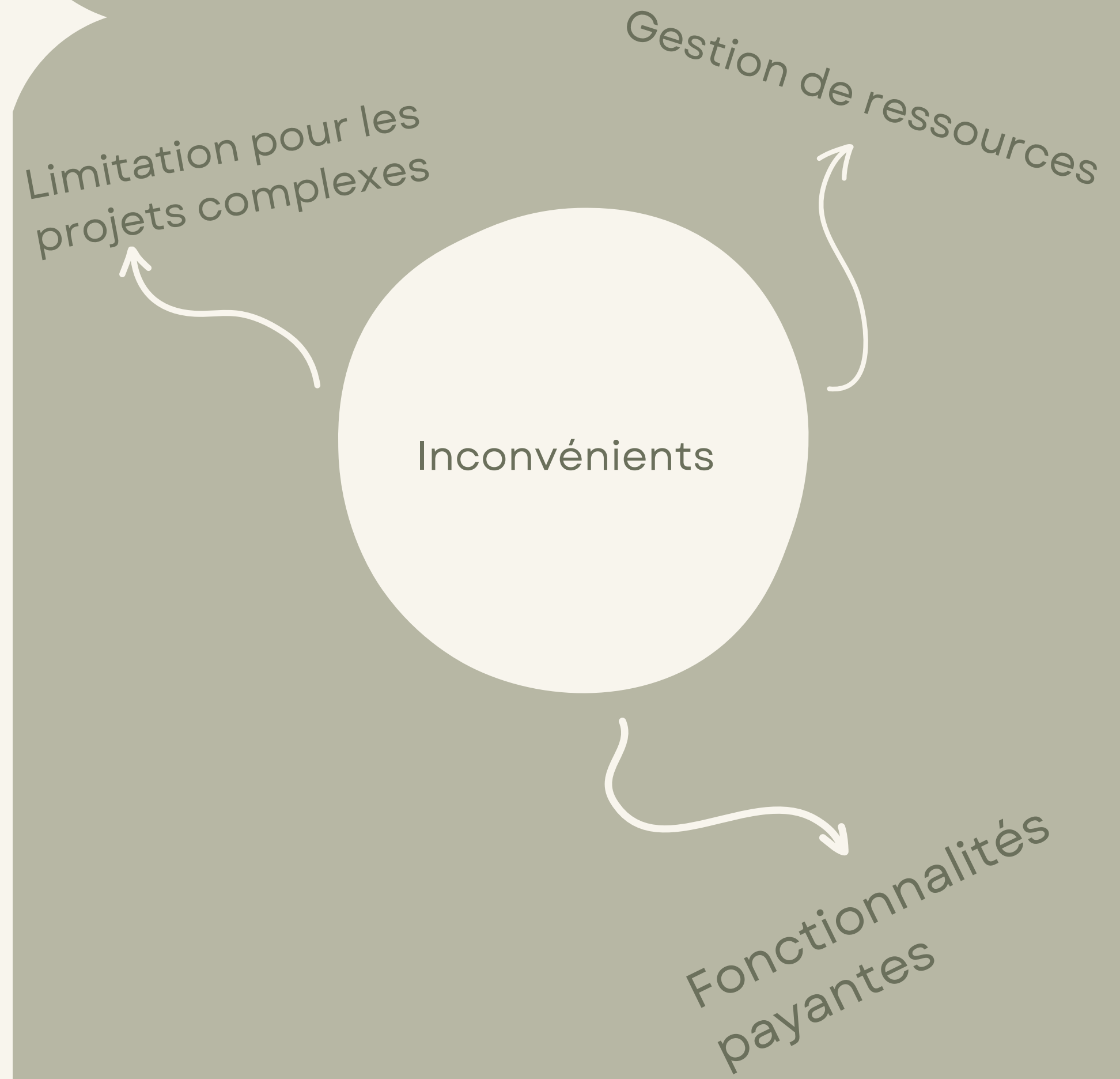
Outils:

- 1- Trello
- 2- Notion
- 3- Asana

Trello

Avantages

- Simplicité et ergonomie
- Visualisation claire
- Collaboration
- Flexibilité
- Accessibilité mobile



Noïion

Avantages

- Interface intuitive et design épuré
- Polyvalence
- Collaboration en temps réel
- Flexibilité
- Intégrations puissantes

Inconvénients

- Courbe d'apprentissage
- Performances
- Mode hors ligne limité
- Coût

Asana

Avantages

- Interface intuitive
- Collaboration efficace
- Flexibilité
- Suivi des tâches
- Automatisation

Inconvénients

- Courbe d'apprentissage
- Coût
- Dépendance à Internet
- Notifications excessives

Organisation

The image displays four Kanban boards, each representing a different team or role in a project. The boards are organized into columns and rows, with tasks listed as cards. Each card includes a title, a description, and a due date. The boards are: 1. Hodavia et Jessica (frontend web), 2. Samuel et Julia (Docker et Mobile), 3. Yann-Méric et Roy (backend), and 4. Samuel et Julia (Database). The boards are set against a background of a forest scene.

Hodavia et Jessica (frontend web)	Samuel et Julia (Docker et Mobile)	Yann-Méric et Roy (backend)	Samuel et Julia (Database)
Page d'accueil	Création du fichier du docker-compose.yml	Coder une fonction pour supprimer les informations de l'utilisateur!(DELETE FUNCTION)	Configuration des différents utilisateurs du site web
Page d'inscription et connexion (Sign Up / Login)	Définition du service pour le serveur d'application pour fonctionner sur le port 8080.	Dec 2 - Dec 8	Dec 2 - Dec 6
Dashboard principal	Définition du Service pour le client mobile pour la construction et le déploiement.	Décider des apis à utiliser pour les services!	Développement des scripts SQL pour initialiser, insérer des données, etc
Choose your process (Voir la maquette).	Définition du service pour le client web pour fonctionner sur le port 8081.	Dec 1 - Dec 2	Dec 6 - Dec 9
Bibliothèque d'automatisations (Automation Library)	Établir les dépendances entre les services en utilisant depends_on.	Décider des actions et réactions à implémenter!	+ Add a card
Page de profil utilisateur (User Profile)	Tester l'intégrité des images en lançant docker-compose up.	Dec 2 - Dec 2	
Page des paramètres (Settings)	Développer l'interface utilisateur du client mobile en suivant les directives de conception de la partie web	Faire la maquette du repo au niveau du back-end.	
Page d'aide et support (Help & Support)	Intégrer la fonctionnalité de configuration de l'emplacement du serveur d'application dans	Dec 2 - Dec 2	
Page à propos (About)		Écrire la documentation du back-end	
+ Add a card	+ Add a card	Finaliser la documentation canva/cahier des charges!	
		Dec 2 - Dec 2	
		Choix et insertion de la stack dans le canva	
		Dec 2 - Dec 2	
		+ Add a card	

Stack

Front-End: React.js

- **Composant réutilisable:** facilite la création d'éléments récurrents dans notre application.
- **Virtual DOM:** rends l'interface rapide et créative
- **Écosystème riche:** compatible avec des systèmes comme Axios(pour la requête HTTPS)

Siack

Back-End: Express.js & Node.js

- **Simplicité:** Express est minimaliste, vous permettant de créer une structure adaptée à votre projet.
- **Évolutivité:** Node.js est asynchrone et performant, idéal pour gérer plusieurs utilisateurs simultanément.
- **Écosystème NPM:** Accès à une grande variété de modules pour étendre les fonctionnalités.

Siack

DATABASE: MySql

- **Fiabilité:** Idéal pour les projets nécessitant des transactions complexes ou des relations entre les tables.
- **Outils compatibles:** Fonctionne bien avec des ORM comme Sequelize pour simplifier les requêtes.



Merci! ✨