



Rapport d'alternance

Projet global de l'entreprise Smartb

Réalisé par
Basile SAVOURET

Sous la direction de

Adrien POUPARD - Référent
Fabien MICHEL - Enseignant tuteur

Pour l'obtention de la LICENCE APIDAE

Année universitaire 2020 - 2021

Remerciements

A Adrien POUPARD pour son accompagnement et son soutien précieux tout au long de mon stage puis de mon alternance. A Fabien MICHEL mon tuteur à l'IUT pour sa présence et ses conseils. Et à ma professeur Justine DELEBARRE pour ses conseils et sa disponibilité.

Sommaire

Remerciements	1
Sommaire	2
Tables de figures	3
Glossaire	4
Introduction	6
1. Présentation de l'entreprise	7
1. 1. Objectifs de l'entreprise	7
1. 2. Projets en développement	9
1.3 Organisation interne de l'entreprise	10
2. Le projet global de Smartb	11
2.1 Analyse de l'existant	11
2.2 Stratégie de l'entreprise	13
3. Cahier des charges	15
3.1 Présentation du sujet et analyse du contexte	15
3.2 Analyse des besoins fonctionnels	16
3.3 Analyse des besoins non-fonctionnels	18
3.3.1 Spécification techniques	18
3.3.2 Contraintes ergonomiques	18
4. Rapport technique	19
4.1 Conception	19
4.1.1 Archetypes-ui	19
4.1.2 I2	22
4.1.3 X2	22
4.2 Réalisation	23
4.2.1 Archetypes-ui	28
4.2.3 X2	31
5. Résultats	35
6. Gestion du projet	37
6.1 Démarche personnelle	37
6.2 Planification des tâches	38
6.3 Bilan critique par rapport au cahier des charges	38
7. Bilan de mi-parcours	40
8. Bilan de fin de parcours	41
Bibliographie	42

Tables de figures

Figure 1: Les objectifs de développement durable selon l'ONU	8
Figure 2: Le tableau des tâches sur Jira	11
Figure 3: Explication tirée du site https://www.ixo.world/	12
Figure 4: Explication tiré du site https://proofofimpact.com/	13
Figure 5: Schéma de l'environnement d'outils de Smartb	14
Figure 6: Diagramme de classe des entités de la plateforme	18
Figure 7: Un morceaux du design system sur figma	21
Figure 8: Exemple de props déclaré en typescript avec sa doc	22
Figure 9: Exemple d'un tableau généré à partir de la doc typescript	22
Figure 10: Première partie d'un composant react	25
Figure 11: Seconde partie d'un composant react (1)	26
Figure 12: Seconde partie d'un composant react (2)	27
Figure 13: Dernière partie d'un composant react	28
Figure 14: Schéma du cycle de vie d'un composant react	29
Figure 15: Exemple d'un type de props complète	30
Figure 16: Décomposition des props d'un composant	30
Figure 17: Exemple d'un composant générique	31
Figure 18: en haut le composant en react, en bas le design sur figma	31
Figure 19: module d'authentification sur storybook	32
Figure 20: L'interface de X2	33
Figure 21: La page de détails d'une session sur X2	34
Figure 22: Récupération du code d'intégration sur X2	34
Figure 23: Interface de l'intégration continue de gitlab	37

Glossaire

Eco-blockchain: Selon Smartb (<https://smartb.city/>) une éco-blockchain est une blockchain* à l'impact énergétique réduit et avec un respect pour la vie privée des utilisateurs.

Blockchain: Une blockchain est une technologie de stockage et de transmission de données stockées de manière répartie dans tous les appareils l'utilisant.

API: Acronyme anglais signifiant Application Programming Interface, c'est un ensemble normalisé de classes, de méthodes, de fonctions et de constantes qui sert de façade par laquelle un logiciel offre des services.

Git: Git est un outil de gestion de code en ligne.

React: React est un framework permettant de créer facilement des composants d'une interface en javascript.

Composants d'interface: Ce sont des fragments de code web pouvant être réutilisés à plusieurs endroits. Grâce à eux on peut créer plus simplement des applications en empilant simplement des composants les uns sur les autres.

Swagger: Une librairie Java permettant d'ajouter automatiquement une interface de documentation à une API

Intégration continue: L'intégration continue est un ensemble de pratiques utilisées en génie logiciel consistant à vérifier à chaque modification de code source que le résultat des modifications ne produit pas de régressions dans l'application développée.

Script de migration: Lorsque les données sauvegarder de l'application doivent être modifiées pour qu'une mise à jour de la plateforme fonctionne, on fait un script de migration de ces données. C'est un morceau de code qui va modifier les données sans les supprimer pour qu'elles puissent convenir à la mise à jour.

Livraison des derniers km: C'est la dernière étape d'une livraison, celle où l'on va apporter le colis directement chez le particulier. C'est la partie la plus courte en distance.

La mobilité active: C'est une mobilité dite écologique où la personne doit être active pour permettre son déplacement (marche à pied, vélo, etc).

Preuve d'impact: Ce sont des preuves d'un impact bénéfique sur l'environnement ou la société, générées à partir de différents outils. Elles ont plus ou moins de crédibilité et permettent à une entreprise de prouver ses actions bénéfiques.

Design system: Un Design System s'apparente à une bibliothèque de composants, visuels et principes au code réutilisable. Ce kit évolutif offre un référentiel UX et UI pour les designers et les développeurs de produits et services digitaux.

Props: Ce sont les propriétés d'un composant JSX. C'est le langage de balisage de react. Le JSX est très proche du HTML donc ses propriétés le sont aussi.

Introduction

Le réchauffement climatique, l'extinction de nombreuses espèces végétales et animales, la famine dans le monde sont tous des enjeux géo-politiques qu'il est de plus en plus urgent de résoudre. Pour trouver des solutions à toutes ces crises grandissantes, des États et des entreprises rassemblent des fonds pour financer les acteurs majeurs dans la mise en place de ces solutions. Afin d'obtenir ces financements, les acteurs doivent apporter des preuves de l'impact* positif qu'ils ont eu face à ces enjeux. C'est dans ce but que Smartb offre ses services permettant de créer des preuves d'impact.

Après un stage très enrichissant au sein de SmartB avec de nombreux projets concentrés sur les problèmes environnementaux du vingt et unième siècle. Il m'a été proposé une alternance pour y continuer mes études et participer au développement du projet de l'entreprise.

Ce projet se caractérise par la création d'un ensemble d'outils permettant à une personne, une structure, une ville ou même un État de stocker et valider ses preuves d'impact bénéfique pour l'environnement et/ou la société.

Ce rapport est un bilan de mon alternance et de tous les projets que j'ai abordés, pour des raisons de cohérence et de fluidité certaines parties de mon rendu de mi-parcours ne seront plus présentes ici.

La première partie est la présentation de l'entreprise. Puis nous verrons le projet global de l'entreprise en présentant un à un les projets auxquels j'ai participé, permettant à Smartb d'améliorer les solutions qu'elle propose. Ensuite il y aura le cahier des charges de Colisactiv qui permettra de résumer et de présenter l'une des solutions apportée par Smartb. Dans une suite logique, nous retrouverons le rapport technique du projet global de Smartb, son utilité dans la solution Colisactiv et son amélioration après celle-ci. Enfin viendront les résultats obtenus, le bilan sur le projet de Smartb et son avenir.

1. PRÉSENTATION DE L'ENTREPRISE

L'entreprise SmartB est une jeune start-up de deux ans. Elle est localisée dans le centre-ville de Montpellier et est composée d'une équipe d'une quinzaine de personnes, toutes spécialisées dans des domaines différents. SmartB est une entreprise visant à développer la ville intelligente au profit d'un impact économique, social et environnemental bénéfique. L'entreprise développe des outils basés sur des technologies innovantes avec, comme exemple le plus concret, leur éco-blockchain*. Cette partie a pour objectif de présenter plus en détails les objectifs et les activités de l'entreprise.

1. 1. OBJECTIFS DE L'ENTREPRISE

SmartB affiche ses objectifs: elle prévoit de se développer internationalement et de créer une nouvelle économie basée sur l'échange de données.

SmartB apporte une évolution technologique pour pouvoir appliquer et atteindre les objectifs de développement durable proclamés par l'ONU que voici (voir figure 1):



Figure 1: Les objectifs de développement durable selon l'ONU

Les cent quatre vingt treize pays membres de l'ONU ont décidé ces objectifs en 2015 (voir figure 1).

Cette évolution technologique se traduit par les outils que SmartB propose. Ces outils ont pour but de devenir open-source et de permettre leur utilisation de manière internationale pour des projets en tout genre. Ils introduisent une nouvelle économie permettant une rémunération en échange de preuves d'impacts bénéfiques sur le développement durable d'une ville ou d'un État. Cette rémunération est basée sur un budget d'impact de sources publiques et privées.

Pour atteindre ces objectifs, il convient d'évoquer et de bien différencier les outils, et les projets développés par Smartb.

Les outils sont:

- Leur blockchain qui apporte une confiance entre les acteurs d'un projet car elle permet de stocker et partager des preuves d'impacts bénéfiques.
- Un portefeuille de données et d'impacts qui permet à n'importe quel citoyen d'y accéder grâce à son identité dématérialisée.
- Des interfaces et des API permettant de visualiser et d'accéder aux objets stockés dans la blockchain
- Une librairie de composants d'interfaces* propre à l'identité visuelle de Smartb utilisée pour développer rapidement les interfaces utilisateurs de ses projets.

Les projets de Smartb sont des applications permettant de faire une démonstration de l'utilisation de ces outils.

Les applications ont pour but de mesurer l'impact bénéfique produit sur un objectif et de générer une preuve, puis la blockchain permet de transmettre cette preuve d'impact pour assurer la rémunération de son auteur.

1. 2. PROJETS EN DÉVELOPPEMENT

Smartb est une jeune entreprise qui essaye de se faire une place et une renommée dans la gestion de preuves d'impact sur l'environnement. Son but premier est donc de travailler sur des projets lui apportant des partenariats durables dans son domaine d'expertise.

Le projet le plus important dans ce cadre est Colisactiv. C'est un projet territorial dirigé par l'entreprise partenaire Sonergia visant à développer l'utilisation de mobilité écologique tel que le vélo pour toutes les livraisons dites "des derniers kilomètres". Smartb a été chargée en tant que prestataire de développer la plateforme web permettant de gérer, organiser et payer les différents acteurs de ce projet. Grâce à ses outils, Smartb est capable de fournir des preuves concrètes aux livreurs optant pour une mobilité plus respectueuse de l'environnement.

Ensuite il y a les projets développés pour améliorer la vitesse et la qualité des solutions mises en place par Smartb:

En premier il y a **Archetypes-ui**, une librairie de composants d'interface permettant d'intégrer rapidement l'identité visuelle de Smartb dans ses projets. Cette librairie est en développement depuis plus d'un an et atteint bientôt sa première version stable.

En second il y a **X2**, une application web permettant la visualisation d'une section de la blockchain et des transactions qui ont été effectuées dedans. Très utile pour vérifier les données qui transitent dans un projet tel que ColisActiv.

Et enfin il y a **I2**, un système d'authentification pouvant être facilement intégré à une application. Il permet d'unifier l'authentification d'utilisateur sur nos projets et la blockchain.

1.3 ORGANISATION INTERNE DE L'ENTREPRISE

L'organisation interne de Smartb mêle habilement coopération, confiance et autonomie. Les horaires de travail sont décidés par les employés et le télétravail est optionnel. La productivité est essentiellement basée sur la coopération. Étant dans une équipe réduite, il est important de pouvoir compter sur tout le monde pour avancer avec régularité sur les projets en cours.

Afin de veiller à ce que cette coopération perdure, une réunion de tous les membres de l'entreprise est organisée tous les lundis matin pour faire un point sur l'avancée des différentes tâches de chaque domaine de l'entreprise. Une réunion de l'équipe technique a lieu quotidiennement pour permettre de clarifier et d'organiser au mieux le travail à réaliser dans la journée.

Les différentes tâches sont notées et suivies grâce au gestionnaire de tâches Jira (voir figure 2).

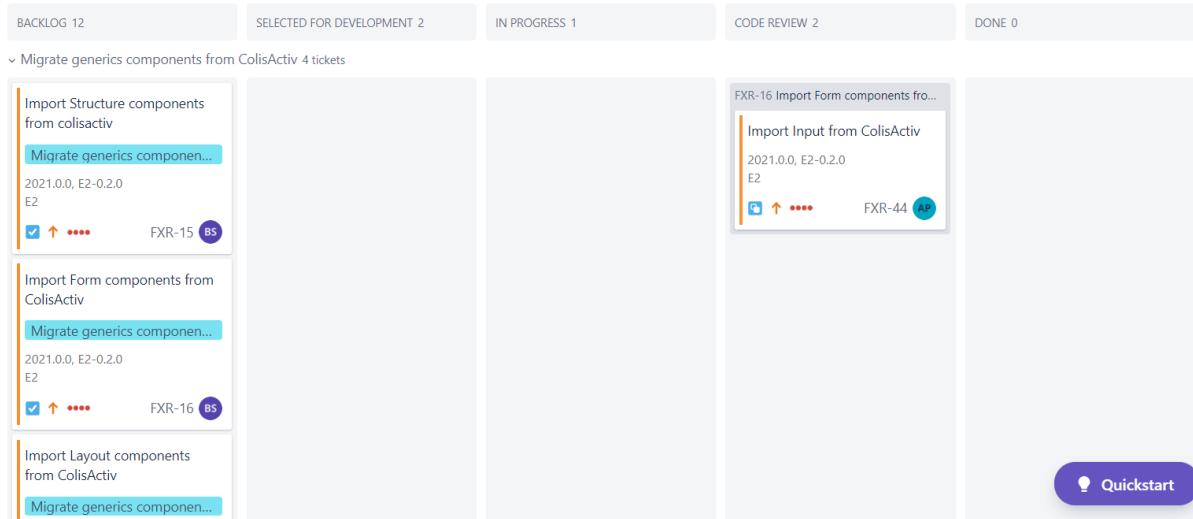


Figure 2: Le tableau des tâches sur Jira

Toutes les réunions de l'entreprise sont effectuées en visio-conférence et une application de chat en temps réel permet aux membres de l'équipe de communiquer à tout moment. Pour la gestion du code des applications créées, Git* est utilisé pour sauvegarder le code sur les serveurs de Smartb grâce à Gitlab.

2. LE PROJET GLOBAL DE SMARTB

Nous venons de voir dans quel contexte Smartb se fait une place dans la gestion de l'impact, en optant pour des technologies novatrices et écologiques. Nous allons maintenant voir ses concurrents et étudier la position et la stratégie de Smartb dans le domaine. Nous pourrons ensuite mieux comprendre les directions prises au niveau de la recherche et du développement de l'entreprise.

2.1 ANALYSE DE L'EXISTANT

Le principal concurrent de Smartb dans ce domaine est [Ixo](#) qui propose un service similaire de déclarations d'impacts vérifiables au travers d'une blockchain (voir figure 3). A la différence que Ixo travaille davantage avec les associations et sur une blockchain privée alors que Smartb se spécialise dans les villes intelligentes et travaille sur une blockchain publique.

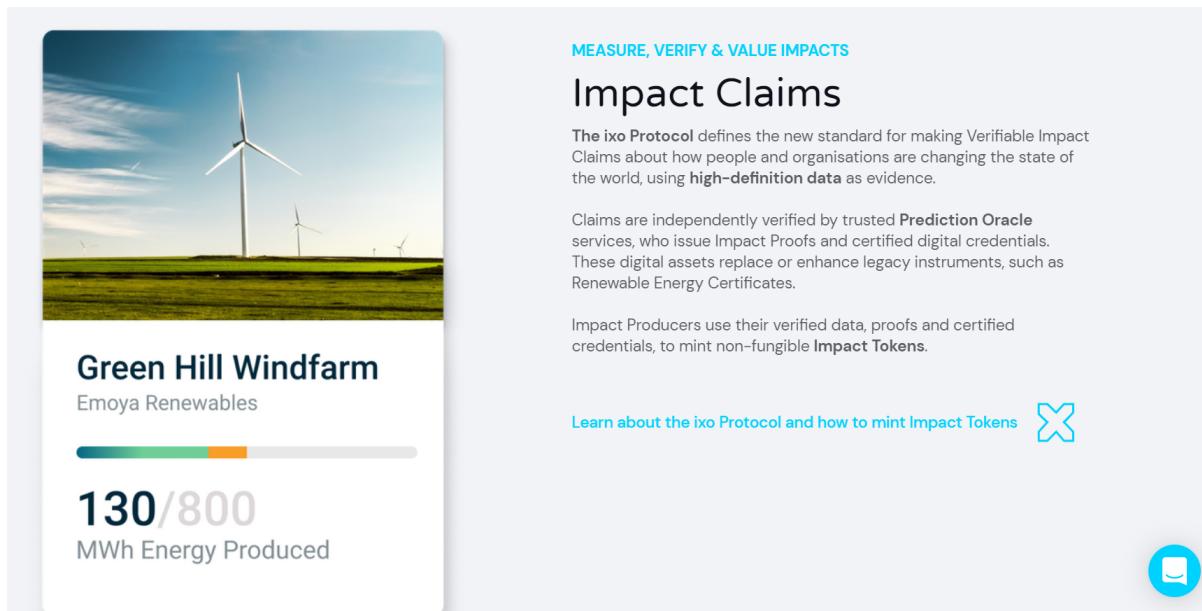


Figure 3: Explication tirée du site <https://www.ixo.world/>

Proof of impact peut aussi être considéré comme un concurrent direct. Leurs plateformes permettent aux entreprises de générer des preuves d'impacts écologiques qui confortent le consommateur ou l'investisseur dans son engagement pour l'entreprise (voir figure 4). De son côté la solution de Smartb intervient davantage comme une monnaie d'échange: l'impact sur la

plateforme peut avoir une valeur et l'organisme l'ayant généré pour obtenir une rémunération en retour.



The screenshot shows the homepage of Proof of Impact. At the top, there's a dark banner with the text "PROOF OF IMPACT" and "MAKING EVERY DATA POINT COUNT". Below this, a large blue header reads "Making Every Data Point Count" with a "Learn More" button. The main background image is a photograph of Earth from space. At the bottom of the page, there's a section with text about the platform's purpose and a heading "Impact-as-a-Service (IaaS): Collect, Verify and Analyze Real time Impact Data". To the right of this text is a small circular icon with a white letter 'Q' and a red notification dot.

Proof of Impact is a data intelligence platform which makes impact data accessible to maximize return on your investments.

Impact-as-a-Service (IaaS): Collect, Verify and Analyze Real time Impact Data

Proof of Impact will help you transform your static impact data reporting into a cost-effective, real-time, dynamic digital solution. This solution will change your company's consumer perception, uplift the people in your organization, and make your company more attractive to next generation investors.

Figure 4: Explication tiré du site <https://proofofimpact.com/>

Ensuite il y a [Inuk](#) qui est une plateforme qui permet de faire des crédits carbone sur une blockchain. Cette plateforme est spécialisée dans l'économie du carbone et non dans la gestion d'impact comme Smartb.

Enfin il y a d'autres organisations qui apparaissent plutôt comme des concurrents de certaines des solutions que vend Smartb aux entreprises plutôt que du projet global. Comme par exemple [Get The Data](#) qui fournit de l'analyse des données d'impact.

2.2 STRATÉGIE DE L'ENTREPRISE

Pour se positionner par rapport à ses concurrents, Smartb a décidé de créer un environnement lui facilitant la création rapide de solutions de qualités pour ses clients. Cet environnement se compose d'une série de plateformes, de bibliothèques et protocoles qui sont présentés dans la figure 5.

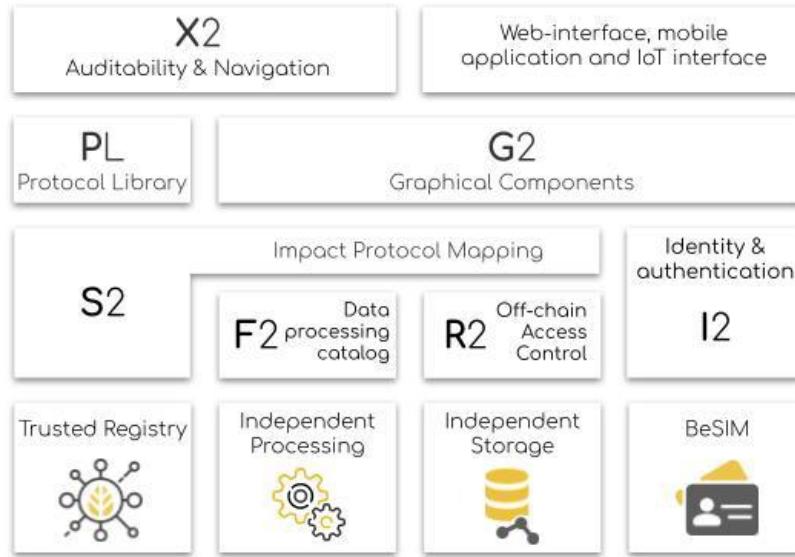


Figure 5: Schéma de l'environnement d'outils de Smartb

Nous verrons uniquement les projets de cet environnement sur lesquels j'ai travaillé et dont j'ai suffisamment de connaissances pour en expliquer les principes fondamentaux.

Tout d'abord **G2** aussi appelé **Archetypes-ui**, est la librairie de composants d'interface basée sur l'identité visuelle de l'entreprise.

Le projet est en développement depuis plus d'un an mais sa concrétisation est récente. Il est avant tout utile pour les développeurs, pour faciliter l'intégration et l'uniformisation du design d'une application en développement mais aussi pour accélérer l'intégration d'autres outils tels que **I2**.

L'identité visuelle de Smartb est d'abord décidée et modélisée dans [Figma](#). Figma est un outil graphique permettant à des designers de concevoir des maquettes d'applications. Nous utilisons cette outil pour réaliser le design system* de Smartb qui est ensuite reproduit sous forme de composant réutilisable dans Archetypes-ui.

Basé sur le projet open-source Keycloak, **I2** est une sorte de plug-in qui va prendre en charge l'authentification à la plateforme et sa sécurisation.

Débutée récemment, **X2** est l'application permettant de visualiser, de traduire et de rassembler des données d'impacts stockées sur la blockchain. X2 n'étant pas attachée à un projet, elle peut être déployée pour chaque projet

dont Smartb est le prestataire. Cela permet aux utilisateurs du projet d'accéder aux données qu'il a générées.

Cet environnement permet à Smartb d'améliorer la qualité et la fiabilité des solutions qu'elle apporte comme avec Colisactiv dont nous allons voir le cahier des charges. Au début du développement du projet une partie de ces outils n'existaient pas encore c'est pourquoi nous verrons par la suite, dans le rapport technique, comment ils ont été intégrés ou démarrés dans Colisactiv.

3. CAHIER DES CHARGES

Le cahier des charges concerne le projet Colisactiv.

C'est un projet dédié au développement de la livraison des derniers km* par mobilité active.

Il n'a pas été écrit de cahier des charges pour celui-ci.

3.1 PRÉSENTATION DU SUJET ET ANALYSE DU CONTEXTE

ColisActiv c'est une expérimentation à échelle réelle d'un mode de développement durable de la livraison de colis sur les derniers kilomètres par la mobilité active*.

Le Programme ColisActiv propose de favoriser les modes de livraison actifs, marche et vélo, pour la livraison de colis, en apportant des financements extérieurs pour rendre économiquement concurrentiels ces modes de livraison.

Smartb intervient dans ce projet comme prestataire pour construire la plateforme nécessaire à son déroulement. Cette plateforme doit faire office d'outil de gestion du projet qui sera utilisé majoritairement par ColisActiv.

Le projet Colisactiv est lauréat de l'appel à programme CEE 2019. Le dispositif des certificats d'économie d'énergie (CEE) a été créé par la loi n° 2005-781 du 13 juillet 2005 de Programme fixant les Orientations de la Politique Énergétique (loi POPE). Ce dispositif repose sur une obligation triennale de réalisation d'économies d'énergie en CEE (1 CEE = 1 kWh cumac d'énergie finale). Les CEE sont attribués, sous certaines conditions, par les services du ministère chargé de l'énergie, aux acteurs éligibles réalisant des opérations d'économies d'énergie. Ces actions peuvent être menées dans tous les secteurs d'activité, dont le transport. C'est pourquoi ce projet rentre dans le cadre du CEE.

Colisactiv est en partie porté par Sonergia qui est une entreprise partenaire de Smartb. Elle a été choisie pour ce projet grâce à ses outils et son expertise dans le domaine de la gestion de mobilité active.

C'est un projet innovant qui répond à un besoin très particulier pour lequel il n'y a pas d'équivalent ou de concurrent.

3.2 ANALYSE DES BESOINS FONCTIONNELS

Comme indiqué précédemment la plateforme Colisactiv est un outil de gestion du projet, à ce titre elle va devoir permettre:

- De subventionner des livraisons effectuées à pied ou à vélo.
- D'augmenter, en impliquant territoires et opérateurs de livraison, le flux de colis livrés par mode actif pour en réduire le coût et le rendre ainsi économiquement avantageux dans la majorité des zones urbaines face à la livraison thermique.
- De valider puis déployer un mode de subventionnement de la livraison par mobilité active et notamment à vélo pour permettre aux collectivités d'agir au profit de son développement.
- Un report modal, en incitant les principales entreprises de logistique à basculer progressivement la livraison des derniers kilomètres d'un véhicule thermique vers un mode actif.
- De certifier une livraison de « colis livré en mode actif ».
- De gérer administrativement les acteurs du projet.

Concrètement l'application doit pouvoir récupérer les données de livraisons des transporteurs et certifier qu'elles sont issues d'une livraison en mode actif. Une fois ces livraisons récupérées les regroupe et leur lie une prime à destination des transporteurs et des opérateurs ayant contribué à ces livraisons. Cette prime doit les inciter à continuer d'utiliser ce mode de livraison et permettre la transition écologique dans ce domaine.

Pour mieux comprendre les acteurs du projet et leurs interactions avec la plateforme, voici un diagramme de classe simplifié des entités de la plateforme (voir figure 6).

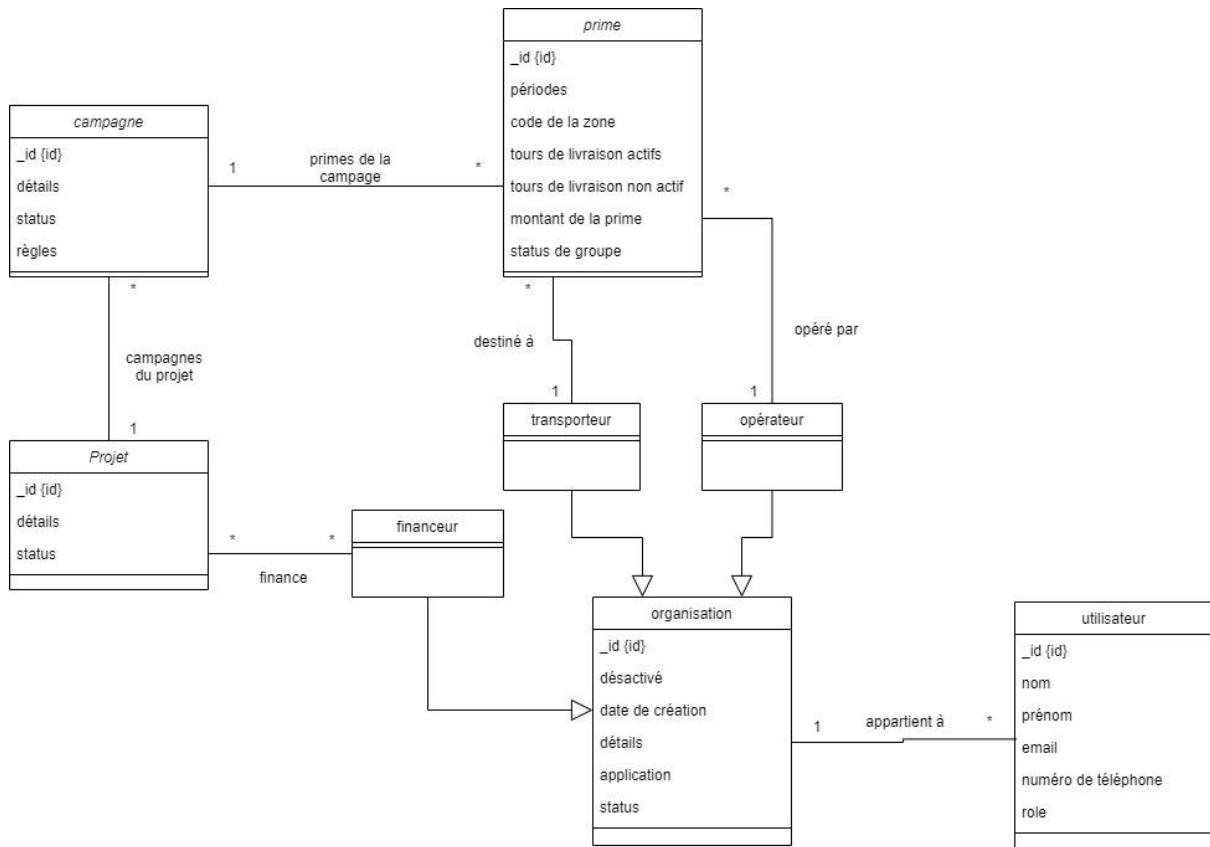


Figure 6: Diagramme de classe des entités de la plateforme

Ce diagramme montre que les utilisateurs de l'application font partie d'une organisation de la plateforme. Les organisations sont de trois types différents, soit :

- Les opérateurs et transporteurs qui s'occupent des livraisons
- les financeurs qui financent les projets de livraisons.

Les financeurs peuvent être des communes ou des départements investissant dans la transition écologique au travers de Colisactiv. Chaque projet de livraison aura plusieurs campagnes se déroulant sur une période et une zone donnée. Dans ces campagnes il est créé des primes à destination des transporteurs pour financer les livraisons faites en mobilité active.

3.3 ANALYSE DES BESOINS NON-FONCTIONNELS

Il y a peu de besoins non-fonctionnels spécifiques. L'application n'est pas destinée à avoir un nombre important d'utilisateurs, ni à être utilisée de manière très régulière.

3.3.1 SPÉCIFICATION TECHNIQUES

La seule spécification technique demandée par Colisactiv est d'utiliser [Docusign](#) pour gérer les signatures de documents administratifs dans la plateforme.

3.3.2 CONTRAINTES ERGONOMIQUES

L'application est conçue pour une utilisation web sur un ordinateur. Il n'y aura pas d'utilisation sur un téléphone donc pas de responsivité à la taille de l'écran à intégrer dans l'interface. L'application est réservée à un usage professionnel, ce qui demande moins d'exigences dans l'expérience utilisateur.

4. RAPPORT TECHNIQUE

Dans chaque partie de ce rapport technique nous verrons chacun des outils avec leur application entre eux et dans le projet Colisactiv. Archetypes-ui sera plus développé que les autres car c'est le projet sur lequel j'ai le plus travaillé depuis la sortie de Colisactiv. Je suis quasiment le seul développeur de l'équipe à travailler sur ce projet. En conséquence, je dois choisir les technologies que j'utilise dans la librairie.

Ces derniers mois, dans l'équipe, je me suis spécialisé pour faire les interfaces des applications. C'est pourquoi je vais aborder seulement la réalisation des projets front-end de smartb. De plus, j'apprécie de plus en plus de réaliser des designs sur Figma ce qui fait que j'ai aussi participé à la création du design system de Smartb dans Figma.

4.1 CONCEPTION

La conception de ces outils a beaucoup évolué , en particulier à la suite de la sortie de Colisactiv. Le développement de la plateforme Colisactiv a engendré certains problèmes que nous verrons juste après ce qui nous a poussé à créer ou améliorer ces outils pour résoudre les dysfonctionnements.

4.1.1 ARCHETYPES-UI

Ce projet a démarré il y a un an et demi quand j'étais en stage chez Smartb. Développé en react sur du Typescript pour pouvoir être réutilisé, nous nous sommes d'abord inspiré et basé sur la librairie open source [Material-ui](#) pour développer les composants. Pour les documenter nous utilisons [Storybook](#) et pour compiler la librairie nous utilisons [Rollup](#). Rollup nous permet de séparer le typescript et de regrouper le javascript dans un seul module ce qui diminue la taille de la librairie .

La librairie a été utilisée dès le début du développement Colisactiv. Dans sa version initiale, Archetypes-ui ne contenait qu'une poignée de composants avec peu de rapport entre eux. De plus, le design dans Colisactiv a changé plusieurs fois ce qui a impliqué beaucoup de travail dans l'interface. C'est pourquoi il nous a fallu un design system (voir figure 7) pour pouvoir donner à

un designer les maquettes des composants déjà existant. Le design system nous évite de changer totalement le design des composants entre chaque application comme ce que nous avions dû faire pour Colisactiv.

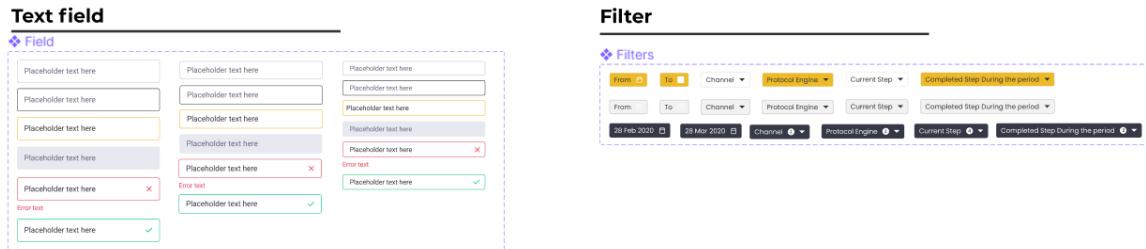


Figure 7: Un morceaux du design system sur figma

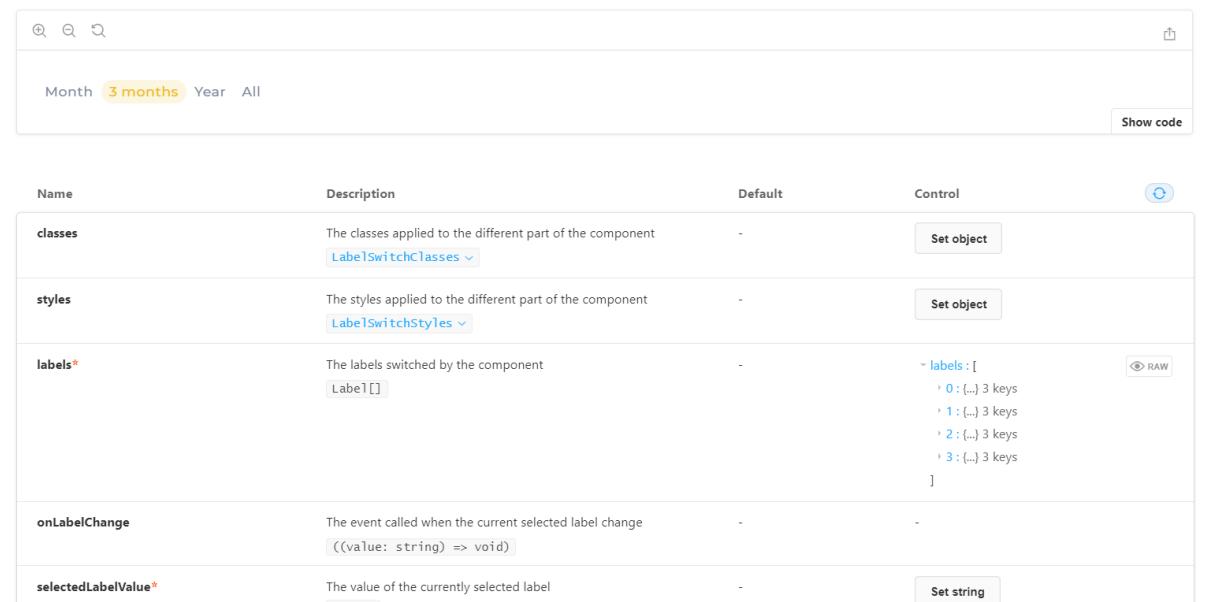
Storybook que nous utilisons pour la documentation est en constante amélioration. Après la sortie de Colisactiv, une mise à jour nous a permis de transformer la documentation typescript (que nous utilisons pour documenter les props* de nos composants) en un tableau affichable dans l'interface (voir figure 8 et 9).

```

export interface LabelSwitchBasicProps extends BasicProps {
    /**
     * The labels switched by the component
     */
    labels: Label[]
    /**
     * The event called when the current selected label change
     */
    onLabelChange?: (value: string) => void
    /**
     * The value of the currently selected label
     */
    selectedLabelValue: string
    /**
     * The classes applied to the different part of the component
     */
    classes?: LabelSwitchClasses
    /**
     * The styles applied to the different part of the component
     */
    styles?: LabelSwitchStyles
}

```

Figure 8: Exemple de props déclaré en typescript avec sa doc
LabelSwitch



The screenshot shows the Storybook interface for the **LabelSwitch** component. At the top, there are three navigation buttons (Home, Back, Forward) and a search bar. Below that is a filter bar with options: Month, 3 months (highlighted in yellow), Year, All, and a "Show code" button.

The main area displays a table of properties:

Name	Description	Default	Control
classes	The classes applied to the different part of the component LabelSwitchClasses	-	Set object
styles	The styles applied to the different part of the component LabelSwitchStyles	-	Set object
labels*	The labels switched by the component Label[]	-	labels : [↳ 0 : (...) 3 keys ↳ 1 : (...) 3 keys ↳ 2 : (...) 3 keys ↳ 3 : (...) 3 keys]
onLabelChange	The event called when the current selected label change ((value: string) => void)	-	-
selectedLabelValue*	The value of the currently selected label	-	Set string

Figure 9: Exemple d'un tableau généré à partir de la doc typescript

Ce système fonctionne à peu près pareil que la génération d'une javadoc ce qui nous fait gagner du temps. De plus, nous avons passé beaucoup de

temps sur Colisactiv à créer des composants qui n'existaient pas dans la librairie mais que nous avons migrés dedans par la suite.

4.1.2 I2

I2 est un projet dérivé du module d'authentification de Colisactiv. Créé à partir de Keycloak nous avons décidé après coup de le séparer dans un projet à part, pour pouvoir le réutiliser ensuite. Mon travail par rapport à ça a été de créer un composant dans archetypes-ui permettant d'ajouter facilement le module I2 dans une application react. En dehors de la partie front-end de cet outil je n'ai quasiment pas travaillé dessus, je n'aborderai que brièvement sa réalisation.

4.1.3 X2

X2 est né récemment d'un besoin logique. Quand nous fournissons une application à un client, les données d'impact de l'application seront stockées sur notre blockchain. Il nous fallait logiquement une solution pour permettre au client de visualiser les données d'impact qu'il génère sur la blockchain.

Cette application est construite en majeure partie grâce à archetypes-ui ce qui permet de tester la librairie dans une application en production. Elle n'a pas de besoin spécifique de performance ou de référencement, nous sommes donc partie sur une application react rendu entièrement côté client. Plutôt que de choisir des technologies de rendu côté serveur comme Next.js qui m'aurait demandé un temps d'apprentissage supplémentaire.

X2 effectue des échanges avec une API connecté à la blockchain pour pouvoir récupérer ses données et les traduire. Elle est codée en kotlin avec le framework spring comme la majeure partie de nos API. J'ai travaillé très rapidement dessus mais le plus gros de mon travail a été sur le front-end de l'application.

X2 avait besoin de graphiques pour permettre de mieux analyser certaines données j'ai donc choisi dans un premier temps d'utiliser la librairie open source Chart.js pour effectuer les graphiques dans l'application react.

4.2 RÉALISATION

Nous ne verrons pas l2 car je n'ai que peu participé à sa réalisation. La partie la plus longue sera Archetypes-ui. C'est le seul projet sur lequel je travaille depuis mon stage en mars 2020 et sur lequel on peut observer mon évolution tout au long de mon apprentissage.

Pour une meilleure compréhension je vais tout d'abord décomposer ce qu'est un composant et comment il est formé.

Nous travaillons sur une architecture react dite "moderne", c'est-à-dire que nous faisons des composants react à base de fonctions et que nous faisons évoluer l'état du composant grâce à des "hooks".

Dans notre architecture chaque fichier (en .tsx pour utiliser react avec Typescript) est un composant.

Dans un fichier on a plusieurs parties pour faire fonctionner un composant:

- La première en haut du fichier regroupe les imports des objets du composant, les types créés pour le composant et le type des props* du composant (les props sont les objets utilisés par le composant pour calculer, organiser et afficher des données) (voir figure 10).

```
//Tous les imports des objets utilisés dans le composant
✓ import React, { useCallback, useMemo, useState } from 'react'
  import { city as cityGeo } from 'plateform-geozone-domain'
  import { SelectorDialog, GeoZoneWithTerritory } from './SelectorDialog'
  import { MultiSelectBox } from '../MultiSelect'
  import { Option } from '../Select'
  import { GenericGroup } from '../TransferList'

//Les types que l'on utilise dans le composant
export interface Territory
  extends cityGeo.colisactiv.platform.geozone.model.Territory {}
export interface GeoZone
  extends cityGeo.colisactiv.platform.geozone.model.GeoZone {}

//Les props typées du composant
✓ interface TerritorySelectorDialogProps {
  title: string
  territories: Territory[]
  //ici le point d'intérogation indique que la prop est optionnelle
  selectedZones?: GeoZone[]
  onValidate: (zones: Map<string, GeoZoneWithTerritory>) => void
  readonly?: boolean
  className?: string
  error?: boolean
  errorMessage?: string
}
```

Figure 10: Première partie d'un composant react

- Ensuite on a la déclaration de la fonction et son contenu dans lequel nous allons déclarer les hooks, les objets et les fonctions que va utiliser le composant (voir figure 11 et 12).

```
//déclaration et export du composant fonctionnel
export const TerritorySelectorDialog = (props: TerritorySelectorDialogProps) => {
    //ici on déstructure l'objet props pour une meilleure compréhension
    const {
        title,
        territories,
        className,
        readonly,
        onValidate,
        selectedZones,
        error,
        errorMessage
    } = props
    //useState est hook de react permettant de stocker
    //et de modifier un état local: le boolean open
    const [open, setOpen] = useState(false)

    //useCallback est un hook de react permettant de garder en mémoire
    //tout au long des mutations de l'objet la fonction à l'intérieur de lui
    const handleOpen = useCallback(() => setOpen(true), [])
    const handleClose = useCallback(() => setOpen(false), [])

    const [loading, setLoading] = useState(false)

    const onChangeLoading = useCallback(
        (Loading: boolean) => setLoading(Loading),
        []
    )
}
```

Figure 11: Seconde partie d'un composant react (1)

```
//ici on utilise le hook de react "useMemo" qui permet de garder
//en mémoire le résultat de la fonction à l'intérieur de lui
//et d'éviter de réexécuter le calcul à chaque re-render du composant
const selectedTerritories: {
  values: string[]
  options: Option[]
} = useMemo(() => {
  if (!zones.selectedZonesMap) return { values: [], options: [] }
  const values: string[] = []
  const options: Option[] = []
  zones.selectedZonesMap.forEach((zone) => {
    if (!values.includes(zone.territoryId)) {
      values.push(zone.territoryId)
      options.push({ label: zone.territoryName, value: zone.territoryId })
    }
  })
  return { values: values, options: options }
}, [zones.selectedZonesMap])
```

Figure 12: Seconde partie d'un composant react (2)

- Ensuite la dernière partie est la partie du “return” de la fonction dans laquelle on retourne un bloc de jsx (qui est le langage de templating de react) qui va être compilé en html et css grâce aux données et aux fonctions du composant (voir figure 13).

```
//on retourne le block jsx du composant
return (
  // le block a forcément un parent qui une balise div dans ce cas
  <div>
    /*
      Dans le jsx on peut inclure d'autre composant react
      ici par exemple on inclut le composant MultiSelectBox
    */
    <MultiSelectBox
      // ici on donne les props au composant
      noMenu
      values={selectedTerritories.values}
      options={selectedTerritories.options}
      id='territory-Selector-input'
      label={title}
      className={className}
      onClick={handleOpen}
      error={error}
      errorMessage={errorMessage}
      readonly={readonly}
    />
    <SelectorDialog
      zonesGroups={zones.groups}
      selectedZones={zones.selectedZonesMap}
      title={title}
    />
)
```

Figure 13: Dernière partie d'un composant react

Comme vu plus haut dans le corps du composant nous utilisons de plus en plus les hooks de react qui nous permettent d'optimiser les performances et de segmenter notre code. Une amélioration significative est la manière dont je structure mes composants. Les composants react ont un cycle de vie plutôt simple dans le résumé (voir figure 14).

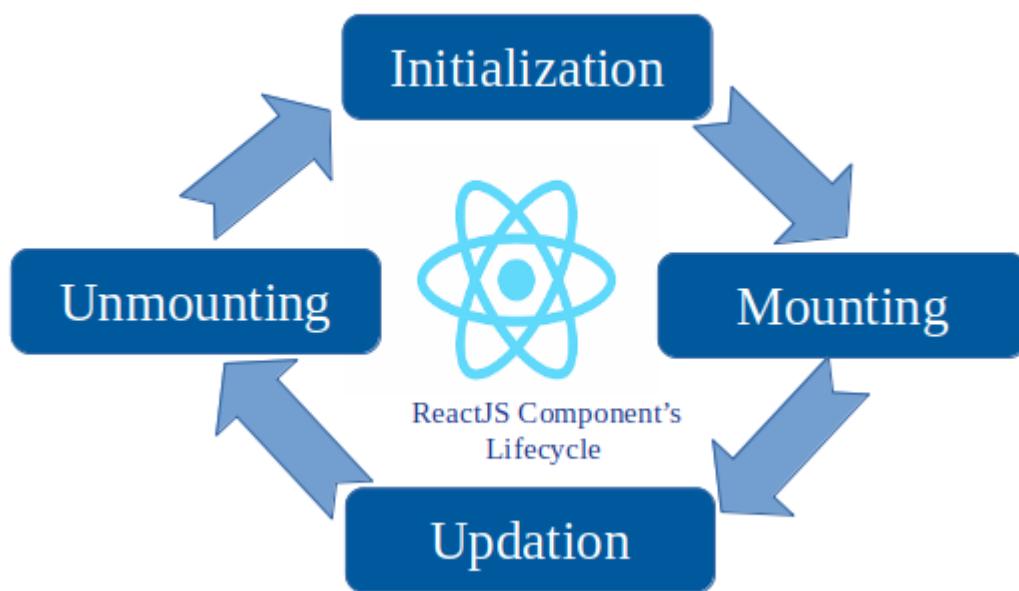


Figure 14: Schéma du cycle de vie d'un composant react

Un composant qui n'est pas affiché dans l'interface est “unmounted” il est “mount” la première fois qu'il apparaît dans l'interface. Une fois affiché si les props qui lui sont données ou des states en local sont modifiés il s'update. C'est à dire que react va recalculer son rendu graphique (ce qu'on appelle un re-render). Donc si des données sont calculées inutilement à chaque re-render ou que des props sont modifiées de manière excessive, beaucoup de calculs inutiles vont être effectués. C'est là qu'interviennent les hooks, ils permettent notamment de:

- garder en mémoire une fonction pour éviter de faire re-render des composants par sa faute.
- utiliser un “state” local c'est-à-dire un objet local capable de faire re-render le composant lors de ses mutations.
- garder le résultat d'une fonction en mémoire au travers des re-render.
- n'exécuter une fonction seulement si certains objets sont modifiés.

4.2.1 ARCHETYPES-UI

Les composants créés dans archetypes-ui doivent être le plus générique possible pour pouvoir être réutilisable et modifiable le plus facilement possible. C'est un point sur lequel je me suis beaucoup amélioré depuis le début du projet.

Chaque composant est basé sur un ou plusieurs composants de material-ui je fais bien attention de donner des méthodes pour changer le style de chaque sous-composant et de transmettre toute les props du composant racine pour garder l'aspect modulable de mon composant. A mes début j'omettais de transmettre les props et de faire en sorte que le composant soit complètement customisable. Pour transmettre les props aujourd'hui je les merge à celles du composant racine de base. Par exemple sur le composant "Link" je merge mes props à celles du composant "Link" de material-ui sur lequel je me base (voir figure 15).

```
export type LinkProps<T extends object = {}> = MergeMuiElementProps< // on crée le type des props du composant
  MuiLinkProps, // ici on lui donne les props du composant material-ui
  LinkBasicProps<T> // et on merge les props de mon composant
>
```

Figure 15: Exemple d'un type de props complète

Ensuite dans le composant on récupère les props par défaut que l'on va repasser directement dans le composant racine (voir figure 16). On donne plusieurs "class" pour s'assurer que chaque partie du composant est stylisable (voir figure 17).

```
const {
  href,
  className,
  component,
  componentProps,
  disabled,
  ...other // cette variable contient les props par défaut
} = props // on déstructure les props
const [linkStyle, linkColor] = useStyle()
```

Figure 16: Décomposition des props d'un composant

Rapport du projet global de l'entreprise Smartb

Basile Savouret

```
return (
  <MuiLink
    ref={ref} // référence de l'objet permettant d'accéder à l'élément html généré
    href={!disabled ? href : undefined}
    className={clsx(
      className, // la classe passé dans les props
      'AruiLink-root', // la classe statique par défaut du composant
      classes.link, // la classe contenant les styles pour accorder le composant au design system
      disabled && classes.disabled
    )}
    variant='body2'
    color='inherit'
    {...other} // on passe toutes les autres props du composant pour ne pas limiter l'utilisateur
  />
)
```

Figure 17: Exemple d'un composant générique

En ce qui concerne la documentation, nous lions directement dans l'interface générée par storybook, le design avec le composant (voir figure 18).

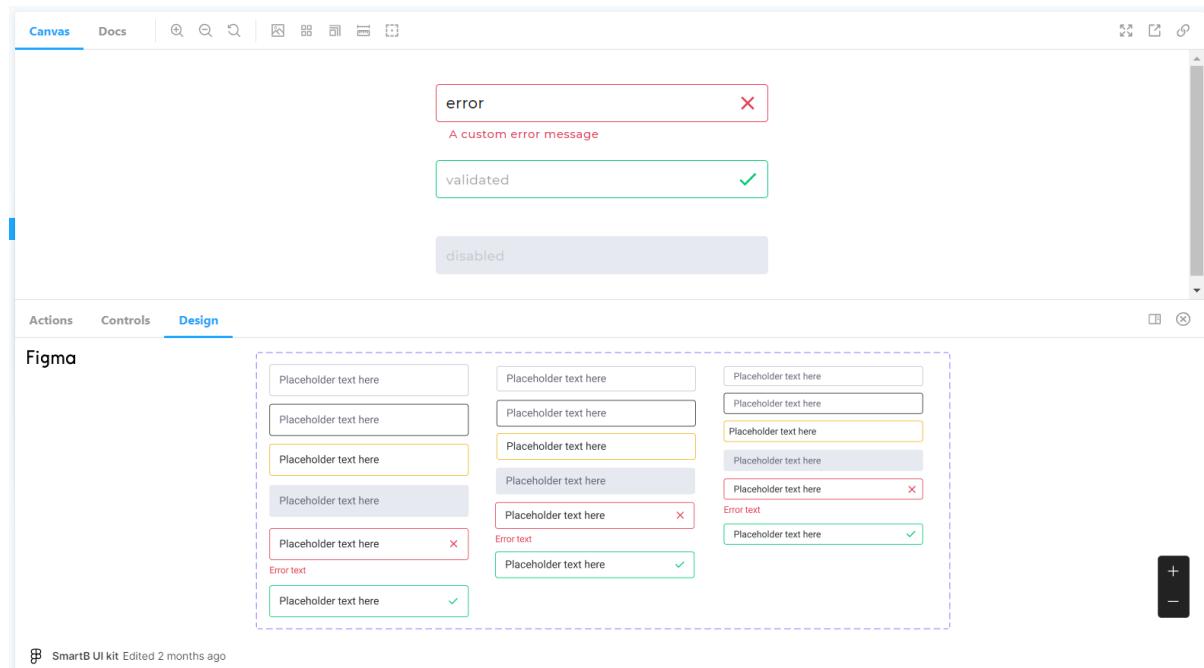


Figure 18: en haut le composant en react, en bas le design sur figma

Tous les composants que j'avais créés avant et qui ne sont pas liés au design system sont obsolètes. Le plus important est de refaire le moins de travail possible donc de pouvoir créer une maquette la plus proche possible des composants déjà existants. C'est aussi pour apporter le point de vue d'un développeur que je participe à l'amélioration du design system.

Archetypes-ui contient des composants mais également des providers. Un provider c'est un module que l'on va ajouter à l'application, il implique toujours un context provider que l'on ajoute à la racine du projet react. Par exemple redux (la librairie pour gérer un store dans react) a un provider. J'ai notamment créé un provider pour se connecter à l2 et donc ajouter facilement le module d'authentification à une application react.

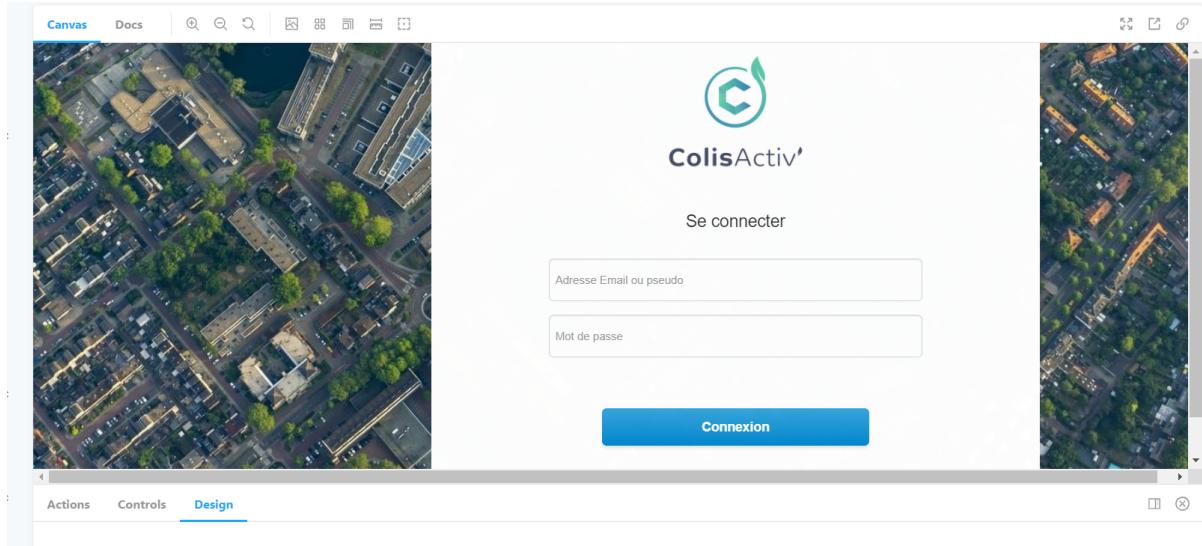


Figure 19: module d'authentification sur storybook

4.2.3 X2

X2 reprend les technologies de bases que nous utilisions pour Colisactiv c'est-à-dire: le framework react en typescript, redux pour la gestion du store, material-ui et archetypes-ui pour l'interface. Ce qu'il y a de particulier c'est qu'elle est construite entièrement autour de archetypes-ui, j'essaye donc de ne faire aucun composant basique et d'utiliser le plus possible la librairie.

De plus, par rapport à Colisactiv j'ai rajouté un package nommé "utils" qui permet de rassembler toutes les fonctions utilitaires que je crée. Ce package a pour but d'être transformé en librairie à terme ou incorporé dans archetypes-ui. En prévision de ça, il ne dépend pas de l'application et reste indépendant.

X2 reprend toutes les bonnes pratiques que je mets en place. Une grosse utilisation de librairies et de modules externes pour limiter un travail spécifique à l'application. Une utilisation constante des hooks de react pour

garder de bonnes performances dans l'application tout au long du développement. Une décomposition assidue des composants react pour réduire leurs tailles et faciliter leurs réécritures.

L'organisation de l'interface reprend de manière générale l'organisation de Colisactiv (voir figure 20).

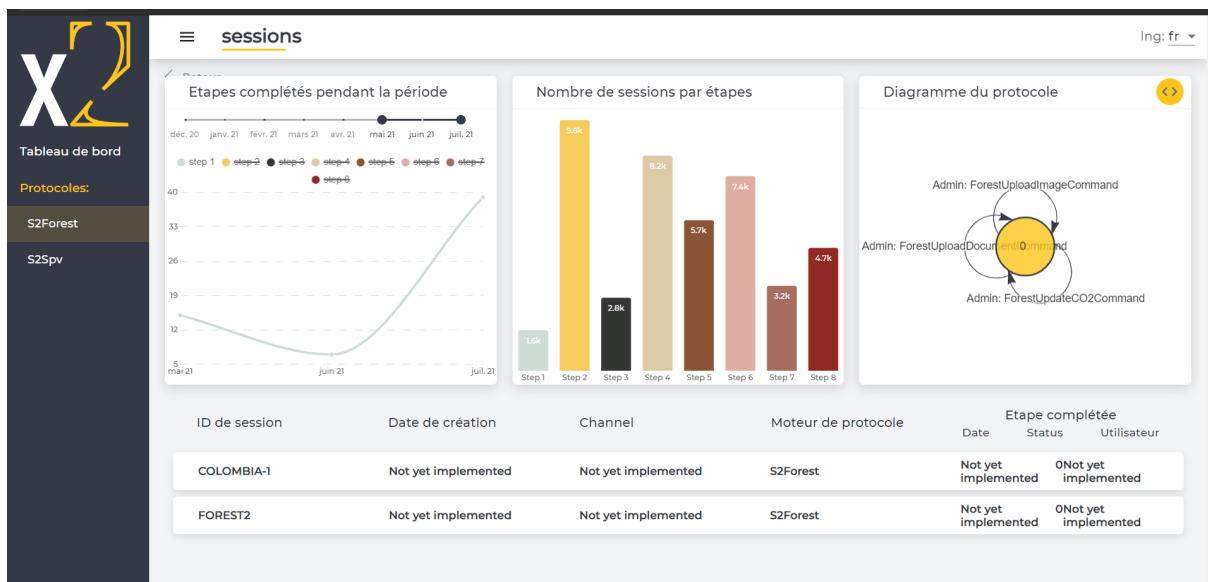


Figure 20: L'interface de X2

une partie de la blockchain que l'on voit contient une liste de SSM (la liste dans la navbar, voir figure 20): c'est une machine à état, un modèle de données dans la blockchain. chaque SSM à une liste de sessions (le tableau en dessous des graph, voir figure 20). Lorsque l'on clique sur une session on arrive à une page qui la détaille et qui nous permet de visualiser la liste des transactions effectuées dessus (voir figure 21).

The screenshot shows the 'Détails' (Details) page of a session on X2. The interface is divided into four main sections:

- Diagramme du protocole:** A circular state transition diagram with three states: 'Admin: ForestUploadImageCommand', 'Admin: ForestUploadDocument0Command' (highlighted in yellow), and 'Admin: ForestUpdateCO2Command'. Transitions exist between all pairs of states.
- Informations sur la session:**
 - ID de session: COLOMBIA-1
 - Date de création: Not yet implemented
 - Etape courante: 0
 - Dernière action effectuée: Admin: ForestUploadDocum...
 - Channel: Not yet implemented
 - Version du moteur de protocole: Not yet implemented
 - Moteur de protocole: S2Forest
- Historique des transactions:**
 - 12/02/2021: Admin: ForestUploadDocumentCommand (status: en cours)
 - 12/02/2021: Initialization (status: terminé)
- Détails de la transaction:**
 - ID de la transaction: 7826d8ca9be276c60490660...
 - Date de la transaction: Not yet implemented
 - Utilisateur: Not yet implemented
 - De: 0
 - à: 0
 - Clé publique: Not yet implemented

Figure 21: La page de détails d'une session sur X2

La séparation en blocs de l'interface permet de créer des fragments pouvant être intégrés dans d'autres sites web (voir figure 22).

This screenshot illustrates the 'Récupération du code d'intégration' (Integration code recovery) feature on X2. It shows the same four panels as Figure 21, but with specific areas highlighted for integration:

- Protocol Diagram Panel:** The 'Diagramme du protocole' section contains an 'Appréciation:' (Preview) box. Inside the preview, the protocol diagram is shown with a yellow highlight around the 'Admin: ForestUploadDocument0Command' state. Below the preview is a code block for embedding the diagram:

```
<iframe
width="300"
height="300"
src="https://axesspoc.smartb.city/embed/S2Forest/diagram"
title="X2 embed content"
frameborder="0"
allowfullscreen>
</iframe>
```
- Session Information Panel:** The 'Informations sur la session' section also contains an 'Appréciation:' box. Inside, it shows the session details with a yellow highlight around the 'COLOMBIA-1' ID.

Figure 22: Récupération du code d'intégration sur X2

Le code d'intégration passe par une route secondaire dans l'application. Par exemple pour les informations de la session accessibles à cette route:

Rapport du projet global de l'entreprise Smartb
Basile Savouret

“/S2Forest/COLOMBIA-1/details” il est possible d’en obtenir une intégration à cette route: “/embed/S2Forest/COLOMBIA-1/informations”.

5. RÉSULTATS

Le projet le plus abouti est actuellement archetypes-ui qui a connu grand bon après la sortie de Colisactiv. Il est déployé et souvent mis à jour à cette url: <https://docs.smartb.city/e2/>. Et nous allons bientôt sortir une version 1.0, c'est-à-dire que nous estimons être proche d'une version stable qui n'aura pas besoin de modification majeure rapidement.

Nous testons la librairie dans les projets que nous utilisons. Avec la sortie de la version 1.0 il faudra sans doute faire des versions instables uniquement pour les tests des nouvelles fonctionnalités. Le design system sur figma est aussi assez peu testé.

Colisactiv est en utilisation depuis plusieurs mois maintenant. Nous continuons d'y apporter des corrections suivant les retours du client et des utilisateurs. Récemment le client à demandé de nouvelles fonctionnalités sur la plateforme donc nous travaillons un peu plus dessus. Nous avons constaté qu'il faudrait effectuer une migration vers la nouvelle version d'archetypes-ui pour réduire la redondance et corriger des bug mineurs. Mais cela demande beaucoup de temps et nous ne pouvons actuellement pas nous permettre de le consacrer.

X2 est toujours en développement, nous espérons pouvoir bientôt en déployer une application pour Colisactiv. Il reste encore du travail à faire dans l'analyse des données car elles viennent de la blockchain et ne sont pas facilement manipulables.

Chaque projet est toujours testé bien entendu par l'intégration continue de gitlab qui nous aide énormément pour maintenir toutes nos applications (voir figure 23).

Rapport du projet global de l'entreprise Smartb
Basile Savouret

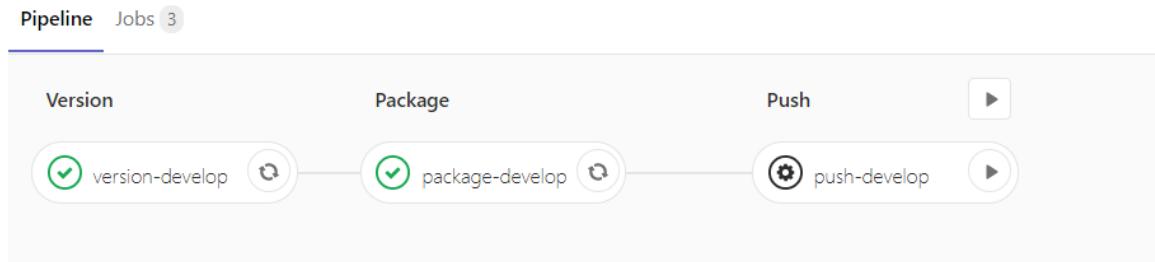


Figure 23: Interface de l'intégration continue de gitlab

6. GESTION DU PROJET

La gestion du projet a été un travail important à fournir. Étant dans une équipe peu nombreuse, une autonomie et une polyvalence était attendue de la part de chaque membre de l'équipe.

6.1 DÉMARCHE PERSONNELLE

L'équipe technique de l'entreprise est, je trouve, très équilibrée. Nous avons chacun trouvé notre place dans le développement du projet et, même si nous étions capables d'effectuer toutes les tâches du développement, nous avions un domaine de prédilection. Le mien fut le développement de l'application web en react et de l'interface utilisateur. Nous avons toujours su collaborer dans l'équipe surtout grâce à la visio-conférence matinale qui nous a permis de mettre chaque jour au clair le travail de chacun. L'entente au sein de l'équipe est restée tout au long du projet très bonne, ce qui a favorisé notre productivité.

Une grande autonomie et possibilité d'initiative dans le travail était accordée ce qui m'a beaucoup plu durant le développement du projet. Chaque proposition d'un membre de l'équipe est retenue et donne lieu à des réunions pour échanger et choisir les directions à prendre. Le plus intéressant fut d'avoir le soutien et l'enseignement d'un tuteur avec plusieurs années d'expérience dans le domaine. J'ai eu beaucoup de place pour exprimer mon point vue et avoir des retours très instructifs quand il n'était pas bon.

Je me suis aussi grandement amélioré pour trouver des réponses par moi-même. Je me suis amélioré dans la construction de solutions face à des problèmes: en m'inspirant de code trouvé sur des forums comme Stack-overflow, en prenant exemple de réponses données par les créateurs de librairies "open-source" que nous utilisons qui sont souvent très exemplaires, ou bien à force de persévérance et de réflexion. J'ai pu avoir le plaisir de proposer et de voir ces solutions approuvées par le reste de l'équipe.

Au cours de ce projet j'ai géré mon travail de manière de plus en plus rigoureuse pour faciliter la coopération et la compréhension autour du travail que je fournis.

D'un point de vue personnel, je trouve cette organisation très agréable et efficace. Je m'épanouis beaucoup au travers de cette gestion d'équipe souple.

6.2 PLANIFICATION DES TÂCHES

Les tâches étaient en grande partie découpées, administrées et assignées par ma collègue qui était la “product owner” du projet. C'était elle qui avait le plus d'interactions avec le client. Elle le représentait au sein de l'équipe et écrivait les tâches par rapport à ses demandes tout au long du développement du projet. Les tâches étaient suivies au travers de Jira qui nous a permis de les ranger par état d'avancement et définir le travail qu'il restait à faire sur chacune d'elles. Les tâches n'étaient pas toutes prédéfinies au début du projet mais étaient ajoutées au fur et à mesure que la vision du client évoluait. Ce choix de développement agile nous a parfois freiné quand la vision du client changeait et qu'il fallait refaire d'anciennes parties de l'application.

Au début du projet à chaque tâches étaient liées des “user stories” qui retracait ce que devait être le comportement que devrait avoir un utilisateur par rapport à la tâche à faire. Cette décision a très vite été abandonnée car elle demande beaucoup de temps et que lier un prototype du designer à chaque tâche à réaliser était beaucoup plus clair.

6.3 BILAN CRITIQUE PAR RAPPORT AU CAHIER DES CHARGES

En ce qui concerne Colisactiv les retours du client jusqu'à maintenant sont très bons. L'application est proche de ce qu'il voulait grâce à un développement Agile qui lui a permis de pouvoir réfléchir à sa demande et à sa vision du projet tout au long du développement. De plus, cette plateforme nous a aussi été très utile. J'ai pu pour ma part migrer des composants depuis Colisactiv dans Archetypes-ui ce qui m'a fait gagner beaucoup de temps.

La gestion de ce projet était un petit peu approximative au début mais de mieux en mieux à la fin. L'équipe technique de Smartb est jeune et c'était le premier gros projet sur lequel nous avons travaillé. Nous avons de ce fait appris à travailler ensemble et monté en compétence en tant qu'équipe.

Nous en sommes sorties plus mûres (professionnellement parlant). Surtout pour ma part, il y a beaucoup d'erreurs faites sur Colisactiv que je ne reproduis pas sur les autres applications. Dans X2 j'ai mis en place, pour moi et les autres développeurs qui pourraient venir travailler sur le projet, des règles de développement. Je m'impose une rigueur pour éviter d'avoir, comme sur Colisactiv, certains endroits avec un code très dense, peu décomposé et que l'on ne peut malheureusement pas corriger après coup, faute de temps. J'essaye de prévenir les problèmes avant qu'ils ne surviennent.

Nous retenons aussi le positif de Colisactiv. C'est-à-dire que les initiatives prises sur Colisactiv qui s'avèrent très utiles nous les reprenons et les généralisons. Comme avec le système d'authentification I2 que nous avons transformé en un projet générique.

7. BILAN DE MI-PARCOURS

Colisativ, avec pour objectif d'aider la transition écologique des modes de livraisons, devait avoir un outil pour gérer les acteurs du projet. Un outil de gestion devant être capable de récupérer et d'approuver les données des livraisons et de calculer en fonction les primes que doivent recevoir les transporteurs. Smartb a su construire et livrer cet outil dans les temps. La plateforme finale a quelques soucis mineurs, il reste des bugs qui sont découverts petit à petit et l'application web a besoin d'encore un peu d'amélioration pour être au maximum de ses performances mais les attentes du client sont atteintes. La plateforme finale est actuellement utilisée et suivant l'évolution du projet, Colisactiv pourrait recontacter Smartb pour continuer de la faire évoluer. Leur vision n'est toujours pas fixée dans le sens où le projet vient tout juste de commencer et que des besoins apparaîtront sûrement tout au long de son déroulement.

Ce fut un projet très enrichissant à concevoir et à développer. J'ai beaucoup appris tout au long de celui-ci notamment grâce à Adrien mon tuteur qui s'est révélé être un très bon pédagogue. J'ai beaucoup aimé et beaucoup appris en travaillant dans une petite équipe où la cohésion était très importante pour avancer, car nos tâches étaient interdépendantes. Au travers de ce projet j'ai aussi pu me concentrer et m'améliorer dans ce que je préfère faire, c'est-à-dire concevoir et réaliser des interfaces. J'ai progressé aussi dans des outils comme Figma permettant de prototyper des applications. C'est un travail créatif que j'ai particulièrement aimé. J'ai pu corriger de mauvaises habitudes que j'avais dans mon code.

Et depuis je prends plaisir à écrire le code le plus propre et intelligent possible, et ainsi faciliter l'utilisation de mon travail par mes collègues. Je cherche davantage de solutions déjà existantes plutôt que de sacrifier mon temps pour les créer. J'essaye de communiquer plus sur ce que je réalise avec mon équipe afin de rester cohérent avec la vision de chacun. Je pense que Smartb est un très bon cadre pour être apprenti, cohérent avec la licence professionnelle APIDAE, et je suis content d'évoluer en son sein.

8. BILAN DE FIN DE PARCOURS

J'ai laissé mon bilan de mi-parcours car il n'a pas beaucoup changé par rapport à aujourd'hui, à l'époque cela faisait déjà un an que j'étais dans l'entreprise. J'avais un bon recul par rapport à tout ce que j'avais appris. Je n'aurais pas grand chose à ajouter mais juste des choses à approuver.

Colisactiv est toujours en phase d'amélioration et nous continuons de faire des prestations pour la faire évoluer. Comme je l'ai dit, c'est le premier projet professionnel que j'ai fait du début jusqu'à la fin. Il m'a été très enrichissant et très utile pour les autres projets qui ont suivi. Smartb développe ses outils et son expertise dans un creux avec des négociations pour concrétiser des prestations. Le plus dur étant, pour une start-up, de trouver les clients ayant besoin de cette expertise. Ce qui fait que l'avenir reste incertain, pour ma part je ferais de mon mieux pour l'aider à proposer à chaque fois de meilleures solutions.

Aujourd'hui, je suis très content du parcours que j'ai suivi, la licence plus l'alternance au sein d'une équipe agréable et prévenante m'a conforté dans mon choix professionnel. Je suis serein pour entrer dans le monde du travail et je suis impatient de voir ce que me réserve mon futur professionnel.

Bibliographie

Smart B | Blockchain for smart cities - La solution des Smart City pour améliorer la qualité des services. (2019, novembre 13). Consulté le 20 mars 2021, à l'adresse <https://smartb.city>. [1]

Code du travail - Article L3261-3-1 | Legifrance. (s. d.). Consulté le 20 mars 2021, à l'adresse <https://www.legifrance.gouv.fr/affichCodeArticle.do?idArticle=LEGIARTI000037985601&cidTexte=LEGITEXT000006072050&dateTexte=20181231>. [2]

Storybook: UI component explorer for frontend developers. (s. d.). Consulté le 20 mars 2021, à l'adresse <https://storybook.js.org/> . [3]

React – Une bibliothèque JavaScript pour créer des interfaces utilisateurs. (s. d.). Consulté le 20 mars 2021, à l'adresse <https://fr.reactjs.org/> . [4]

ColisActiv' - Valoriser la livraison verte pour un environnement sain. (2020, 3 décembre). Consulté le 24 mars 2021, à l'adresse <https://colisactiv.city/>. [5]

The first single application for the entire DevOps lifecycle. (s. d.). Consulté le 20 mars 2021, à l'adresse <https://about.gitlab.com/> . [6]

npm | build amazing things. (s. d.). Consulté le 20 mars 2021, à l'adresse <https://www.npmjs.com/> . [7]

Home. (s. d.). Consulté le 8 mai 2020, à l'adresse <https://yarnpkg.com/>. [8]

Redux - A predictable state container for JavaScript apps. | Redux. (s. d.). Consulté le 20 mars 2021, à l'adresse <https://redux.js.org/>. [9]

Wikipedia contributors. (2021, 16 mars). Interface de programmation. Consulté le 20 mars 2021, à l'adresse https://fr.wikipedia.org/wiki/Interface_de_programmation [10]

Atlassian. (s. d.). Jira | Logiciel de suivi des tickets et des projets. Consulté le 20 mars 2021, à l'adresse <https://www.atlassian.com/fr/software/jira> [11]

API Documentation & Design Tools for Teams | Swagger. (s. d.). Consulté le 21 mars 2021, à l'adresse <https://swagger.io/> [12]

Wikipedia contributors. (2021a, janvier 28). Intégration continue. Consulté le 21 mars 2021, à l'adresse https://fr.wikipedia.org/wiki/Int%C3%A9gration_continue#:~:text=L'int%C3%A9gration%20continue%20est%20un,r%C3%A9gression%20dans%20l'application%20d%C3%A9velopp%C3%A9e [13]

Stack Overflow - Where Developers Learn, Share, & Build Careers. (s. d.). Consulté le 21 mars 2021, à l'adresse <https://stackoverflow.com/> [14]

DocuSign : Envoyez, Signez, Approuvez électroniquement. (s. d.). Consulté le 24 mars 2021, à l'adresse <https://www.docusign.fr/> [15]

Dispositif des Certificats d'économies d'énergie. (s. d.). Consulté le 24 mars 2021, à l'adresse <https://www.ecologie.gouv.fr/dispositif-des-certificats-deconomies-dennergie?fbclid=IwAR0ug1QwOibCIDYIC2w2hTZ5l6SBHXM89q0NJsQom1v-IA11PfJP-CnmOsU> [16]

ixo The Internet of Impact. (s. d.). Consulté le 26 août 2021, à l'adresse <https://www.ixo.world/> [17]

Proof of Impact – Make every data point count. (s. d.). Consulté le 26 août 2021, à l'adresse <https://proofofimpact.com/> [18]

Inuk - Compensez votre empreinte carbone de façon transparente. (s. d.). Consulté le 26 août 2021, à l'adresse <https://www.inuk.co/> [19]

Figma : the collaborative interface design tool. (s. d.). Consulté le 27 août 2021, à l'adresse <https://www.figma.com/> [20]

U. (s. d.). Qu'est-ce qu'un Design System ? Définition Design System UX. Consulté le 27 août 2021, à l'adresse <https://www.usabilis.com/design-system/> [21]

Résumé en français

After completing Colisactiv by creating an adequate web platform for this ecological project. Smartb improves and develops its tools and expertise. It creates a desAprès avoir mené à son terme Colisactiv en réalisant une plateforme web adéquate pour ce projet à but écologique. Smartb améliore et développe ses outils et son expertise. Elle crée un design system basé sur son identité visuelle lui permettant d'améliorer considérable archetypes-ui, sa librairie de composants. Smartb utilisant une blockchain publique avec une gestion de données complexe il lui fallait un moyen de permettre à ses clients de visualiser et de comprendre ces données. X2 à alors été désigné et développé pour remplir ce rôle. Pour réaliser la plateforme de Colisactiv et ses nombreux outils, Smartb a utilisé une grande variété de services dont:

- Une intelligence artificielle codée en python. Elle permet de pouvoir valider les livraisons effectuées avec une mobilité active.
- Une base de données de type MongoDB pour stocker ces livraisons.
- Une API REST codée en kotlin avec le framework spring.
- Une application en typescript avec le framework react.
- Une librairie de composant react documentée grâce à Storybook.
- une suite de composant basique material grâce à Material-ui.

Mot clés: transition écologique, preuves d'impact, mobilité active, API, Spring, React, livraisons, plateforme web, blockchain, Material-ui, Storybook.

Résumé en anglais

ign system based on its visual identity allowing it to considerably improve archetypes-ui, its component library. Smartb using a public blockchain with complex data management, it needed a way to allow its customers to visualize and understand this data. X2 was then designated and developed to fulfill this role. To create the Colisactiv platform and its many tools, Smartb used a wide variety of services including:

- An Artificial intelligence coded in python. It makes it possible to validate deliveries made with active mobility.
- A MongoDB type database to store these deliveries.
- A REST API coded in kotlin with the spring framework.

- A typescript web application with the react framework.
- A react component library documented thanks to Storybook.
- a basic material component suite using Material-ui.

Keywords: ecological transition, proofs of impact, active mobility, API, Spring, React, deliveries, web platform, blockchain, Material-ui, Storybook.

Signature de l'apprenti:



Signature du référent:

Lu et approuvé

30/08/2021

