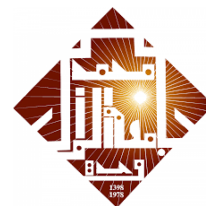


Université Mohammed Premier
École Nationale des Sciences Appliquées
Oujda



Mémoire de Projet de Fin d'Année
Filière : Ingénierie Data Science & Cloud Computing
Soutenu le : 08/07/2023

Création automatisée d'une DAO sur la blockchain
Ganache et la mise en place de l'infrastructure
sous-jacente sur AWS

Réalisé par :

HAOUZI Basma
SIDIBE Mamadou

Membres de jury :

M. KOULALI Mohammed Amine
M. AZGHIOU Kamal

Encadré par :

M. AZGHIOU Kamal

Année Universitaire 2022 - 2023

Remerciements

Nous tenons à exprimer nos sincères remerciements à toutes les personnes qui ont joué un rôle essentiel dans la réussite de ce projet de fin d'année. En particulier, nous souhaitons remercier notre encadrant, Monsieur Azghiou Kamal, pour son précieux encadrement, son soutien constant et ses conseils instructifs tout au long de l'élaboration de ce travail. Nos remerciements les plus sincères vont également au membre du jury, notamment Monsieur **M. KOULALI** Mohammed Amine, pour l'intérêt qu'il a porté à notre travail et pour avoir accepté de l'examiner. Nous souhaitons également exprimer notre gratitude envers toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet.

Résumé

Dans ce travail, nous proposons une solution basée sur la blockchain Ganache (un simulateur de blockchain Ethereum, à des fins de démonstration) pour la création d'organisations autonomes décentralisées (DAO) dans le contexte de gestion de données. Notre approche consiste à générer un fichier JSON à partir des données fournies par les utilisateurs, qui sera utilisé par AWS (Amazon Web Services) pour créer des machines virtuelles (instances) en utilisant des services tels que Amazon Lightsail. Une fois que le fichier JSON est généré, il contient les informations nécessaires pour créer les instances virtuelles souhaitées, telles que le nom de l'instance, la région, le type d'instance, etc. Ce fichier est ensuite utilisé dans un script d'automatisation pour créer les instances de manière programmée. L'utilisation d'AWS Lightsail présente de nombreux avantages, notamment la facilité de déploiement et de gestion des instances, la scalabilité, la disponibilité des ressources et l'intégration avec d'autres services cloud d'AWS. Lightsail permet également de lier une adresse IP publique aux instances, ce qui facilite l'accès aux sites web ou aux applications déployées. En résumé, en utilisant AWS Lightsail et en automatisant la création d'instances à partir du fichier JSON généré, nous pouvons rapidement déployer des machines virtuelles pour héberger des sites web WordPress dans le cadre de notre DAO. Cela permet de fournir une infrastructure solide et évolutive pour faciliter la rencontre entre les fournisseurs de données et les fournisseurs de services, tout en garantissant la sécurité et la protection des données grâce aux rôles IAM et aux politiques de sécurité appropriées.

Abstract

In this work, we propose a solution based on the Ganache blockchain (an Ethereum blockchain simulator for demonstration purposes) for the creation of decentralized autonomous organizations (DAOs) in the context of data management. Our approach involves generating a JSON file from user-provided data, which will be used by AWS (Amazon Web Services) to create virtual machines (instances) using services like Amazon Lightsail. Once the JSON file is generated, it contains the necessary information to create the desired virtual instances, such as instance name, region, instance type, etc. This file is then used in an automation script to programmatically create the instances. The use of AWS Lightsail offers several advantages, including ease of instance deployment and management, scalability, resource availability, and integration with other AWS cloud services. Lightsail also allows linking a public IP address to instances, facilitating access to deployed websites or applications. In summary, by utilizing AWS Lightsail and automating the creation of instances from the generated JSON file, we can quickly deploy virtual machines to host WordPress websites within our DAO. This provides a robust and scalable infrastructure to facilitate the interaction between data providers and service providers while ensuring security and data protection through IAM roles and appropriate security policies.

Table des matières

Liste des figures	6
Liste des tableaux	7
Introduction	8
1 Terminologie et concepts	10
1.1 Blockchain : principe clé et concept de base . . .	10
1.2 Définition d'un DAO , ses fonctionnalités et avantages	12
1.2.1 DAO (Decentralized Autonomous Organization)	12
1.2.2 Les fonctionnalités d'un DAO	12
1.2.3 Les avantages des DAO	13
2 Conception de l'interface web avec WordPress	15
2.1 Description des fonctionnalités de l'interface . .	15
2.2 Explication des choix des outils utilisés	17
2.3 Architecture et structure de l'interface	19
3 Développement du DAO sur Ganache	22
3.1 Présentation de Ganache	22
3.2 Explication du langage Solidity et Truffle	24
3.2.1 Solidity	24
3.2.2 Turffle	25

3.3	Détails sur les smart contracts développés	27
3.4	Utilisation de Truffle	28
3.4.1	Déploiement des smart contracts et création du dao	28
3.4.2	Création d'une proposition, vote et exécution	32
3.5	Résultats et évaluation	35
3.5.1	Tests et validation du DAO	35
4	Analyse des services Cloud	36
4.1	Étude comparative entre les différents fournisseurs cloud	36
4.1.1	Méthodologie :	36
4.1.2	Résultats préliminaires :	37
4.2	Les services cloud : AWS	38
4.2.1	Les Services proposés par AWS Lightsail :	41
4.3	Intégration avec le cloud AWS	42
4.3.1	Configuration des machines virtuelles	42
4.3.2	Création de notre instance Lightsail :	42
4.3.3	Installation de WordPress :	43
4.3.4	Migration des données :	44
4.3.5	Configuration de la base de données :	45
4.3.6	Génération du fichier JSON :	46
4.3.7	Création un script d'automatisation :	46
4.3.8	Configuration des informations d'identification :	46
4.3.9	Analysez le fichier JSON :	47
4.3.10	Utilisation de l'API Lightsail :	47
4.3.11	création d'un compartiment "dataprovers3" dans s3	47
4.3.12	Stocker le file json dans s3 :	48
4.3.13	Creation user dataprovider et sa permission :	48

4.3.14	Creation user serviceprovider et sa permis-	
	sion :	49
4.4	avantage/contrainte du cloud	50
4.4.1	Avantages de l'utilisation du cloud dans le	
	DAO :	50
4.4.2	Contraintes de l'utilisation du cloud dans	
	le DAO :	51
	Conclusion et perspectives	52
	Bibliographie	54

Liste des figures

1.1	Architecture et Fonctionnalité blockchain	14
2.1	Page d'accueil de l'interface	17
2.2	Structure du fichier JSON pour la création du DAO	21
3.1	Interface de Ganache	24
3.2	Fichier de config Turffle	26
3.3	code de déploiement des smart contracts	30
3.4	Exemple de création de DAO	32
3.5	Transaction sur la blockchain	35
4.1	adresse IP Public	43
4.2	service lightsail	44
4.3	Base de données Lightsail	46
4.4	compartiment dataproviders3 dans S3	48
4.5	stockage des données JSON	48
4.6	Permissions user dataprovider	48
4.7	Permissions user serviceprovider	49

Liste des tableaux

4.1	Tableau des différences entre Amazon EC2 et Amazon Lightsail	40
-----	--	----

Introduction

Les infrastructures IoT génèrent actuellement une grande quantité de données à exploiter dans le monde. Les grandes entreprises rivalisent pour fournir des dispositifs intelligents de nouvelle génération, offrant un mode de vie amélioré aux clients et une surveillance en temps réel des opérations commerciales aux entreprises. Quel que soit le système IoT utilisé, une infrastructure sous-jacente génère toujours un ensemble de données exploitable.

Dans ce contexte, notre travail propose une solution qui accorde plus de droits aux propriétaires des données. L'objectif principal est de permettre au propriétaire de décider de l'exploitation des données, en lui garantissant un retour sur investissement optimal. Pour prévenir toute utilisation non autorisée des données, nous proposons une solution basée sur le chiffrement des données à l'aide du calcul multipartite sécurisé. La blockchain est utilisée pour assurer la traçabilité des données grâce à ses registres immuables, ce qui est particulièrement utile en cas de litiges.

Pour préserver les droits des utilisateurs de données, notre architecture propose une structure organisationnelle basée sur le paradigme des organisations décentralisées et autonomes (DAO). Un DAO est créé lorsque le propriétaire des données et un fournisseur de valeur conviennent de collaborer. D'autres fournisseurs de valeur peuvent rejoindre un DAO existant en redistribuant les avantages et en mettant à jour leur version. Ce processus se répète

jusqu'à ce que le DAO atteigne un état d'équilibre, par exemple lorsque le rendement devient moins attrayant pour de nouveaux participants.

Le projet que nous présentons a pour objectif de créer un système de DAO (Decentralized Autonomous Organization) entre un fournisseur de données et un fournisseur de services afin d'apporter de la valeur aux données. Dans un monde de plus en plus numérique et connecté, la quantité de données générées est exponentielle. Cependant, de nombreux fournisseurs de données ne parviennent pas à exploiter pleinement le potentiel de ces informations précieuses.

La mise en place d'un système de DAO permettrait de faciliter l'échange et la valorisation des données. Dans ce contexte, nous prévoyons de développer un site Internet doté d'une interface de communication conviviale. Cette interface permettra aux utilisateurs de données de contacter les utilisateurs de services et de négocier les conditions d'utilisation des données. Ainsi, les fournisseurs de données auront la possibilité de monétiser leurs données en les rendant disponibles à des tiers intéressés.

Chapitre 1

Terminologie et concepts

1.1 Blockchain : principe clé et concept de base

Blockchain : technologie de stockage et de transmission d'informations à coût minime, sécurisée, transparente, et fonctionnant sans organe central de contrôle. Par extension, une blockchain (littéralement une « chaîne de blocs ») désigne une base de données sécurisée et décentralisée, répliquée sur un très grand nombre de noeuds, et contenant un ensemble de transactions dont chacun peut vérifier la validité. Une blockchain peut donc être assimilée à un grand livre comptable transparent, pseudonyme et infalsifiable.

Clé privée : clé permettant à l'utilisateur d'une blockchain d'initier une transaction en signant cryptographiquement son message.

Clé publique : clé servant d'adresse sur une blockchain. Connue de tous, elle permet à un émetteur de désigner un destinataire.

Cryptomonnaie : monnaie électronique, échangeable en pair-à-pair (c'est-à-dire sans intermédiaire), se basant sur des principes cryptographiques et des mécanismes d'incitation économique pour la validation des transactions et la génération de la monnaie elle-même.

Ethereum [1] : plateforme décentralisée, fondée sur la blockchain éponyme, qui vise l'avènement d'un web décentralisé. La blockchain d'Ethereum fonctionne avec la cryptomonnaie ether. Contrairement à la blockchain du bitcoin, focalisée sur l'aspect monétaire, la blockchain d'Ethereum a vocation à accueillir des programmes très divers, appelées dApps, qui fonctionnent avec des smart contracts. On parle d'écosystème Ethereum pour désigner l'ensemble des acteurs qui travaillent sur cette plateforme.

Ether : cryptomonnaie d'Ethereum, considérée non comme une monnaie mais comme un 'carburant' pour faire fonctionner les smart contracts sur cette blockchain.[1]

Metamask : est une extension de navigateur qui fonctionne comme un portefeuille de cryptomonnaie et un outil d'interaction avec les applications décentralisées. Il permet aux utilisateurs de se connecter à des blockchains, de gérer leurs comptes, d'effectuer des transactions et de signer des contrats intelligents, le tout directement à partir de leur navigateur web.

Solidity : langage de programmation utilisé pour développer des smart contracts, en particulier sur Ethereum.[2]

Smart contract : programme autonome qui, une fois démarré, exécute automatiquement des conditions inscrites en amont dans la blockchain, sans nécessiter d'intervention humaine. Il fonctionne comme toute instruction conditionnelle de type « if – then » (si telle condition est vérifiée, alors telle conséquence s'exécute). Pour déclencher son exécution, un smart contract se connecte à une base de données jugée fiable, via l'intermédiaire d'un oracle (un service qui fait le lien entre le smart contract et le monde réel).

Token : actif numérique personnalisé par son auteur, émis et échangeable sur une blockchain, et possédant les caractéristiques d'une cryptomonnaie : infalsifiabilité, unicité, enregistrement des échanges dans un registre immuable, sécurité des échanges, etc. En particulier, un token est transférable (et non duplicable) entre deux parties sur Internet, sans nécessiter l'accord d'un tiers. [3]

1.2 Définition d'un DAO , ses fonctionnalités et avantages

1.2.1 DAO (Decentralized Autonomous Organization)

organisation décentralisée dont les règles de gouvernance sont automatisées et inscrites de façon immuable et transparente dans une blockchain ,c'est une forme d'organisation autonome et décentralisée qui fonctionne sur la base de contrats intelligents exécutés par une blockchain. Un DAO est conçu pour permettre la prise de décisions collectives et l'exécution automatisée d'actions prédéterminées, sans nécessiter d'intervention humaine.

1.2.2 Les fonctionnalités d'un DAO

Les fonctionnalités d'un DAO peuvent varier en fonction de sa conception spécifique, mais voici quelques exemples courants :

- Gouvernance démocratique : Les membres d'un DAO peuvent participer aux processus décisionnels en votant sur des propositions. Ces décisions peuvent concerner des questions stratégiques, des modifications des règles internes du DAO ou l'allocation des ressources.
- Transparence et auditabilité : La technologie de la blockchain offre une transparence totale des transactions et des activités au sein du DAO. Toutes les actions, les décisions et

les échanges de valeur sont enregistrés de manière immuable et vérifiable.

- Automatisation des processus : Les DAO sont conçus pour exécuter automatiquement des actions prédéfinies en fonction de conditions spécifiques. Par exemple, un DAO peut déclencher des paiements ou allouer des ressources une fois qu’une condition contractuelle est remplie.

1.2.3 Les avantages des DAO

Les avantages des DAO comprennent :

- Décentralisation : Les DAO éliminent le besoin d’une autorité centrale ou d’intermédiaires pour prendre des décisions et exécuter des actions. Cela permet une plus grande autonomie et une réduction des coûts liés à la gestion traditionnelle des organisations.
- Transparence et confiance : La nature transparente de la blockchain offre une visibilité totale des activités du DAO, renforçant ainsi la confiance entre les membres.
- Participation ouverte : Les DAO permettent à quiconque détient les jetons ou respecte les critères d’adhésion d’y participer. Cela favorise l’inclusion et l’ouverture des organisations à des contributions diverses.
- Flexibilité et évolutivité : Les DAO sont conçus pour être adaptables et évolutifs, permettant aux organisations de s’ajuster aux changements (ajout d’un nouveau membre) et de répondre aux besoins changeants de leurs membres.

En résumé, un DAO est une organisation autonome et décentralisée qui fonctionne sur la base de contrats intelligents exécutés par une blockchain. Il offre des fonctionnalités telles que la gouvernance démocratique, la participation active des membres, la

répartition des bénéfices, la transparence, l'automatisation des processus, ainsi que des avantages tels que la décentralisation, la confiance et la résistance à la censure. Voici ci-dessous un schéma explicatif représentant l'architecture et les fonctionnalités de la blockchain :

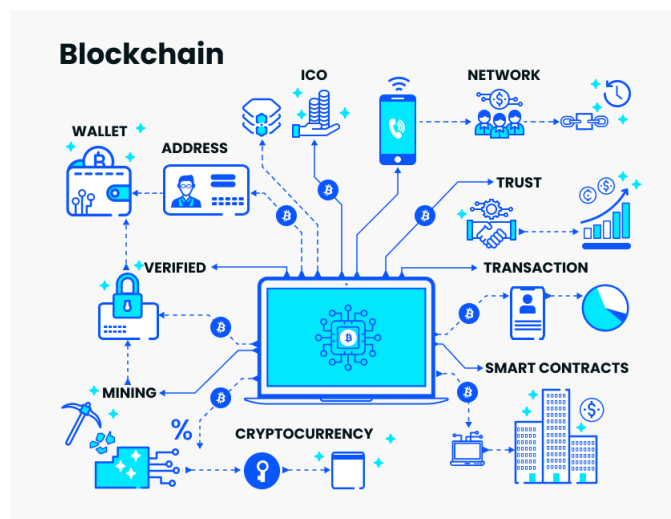


FIGURE 1.1 – Architecture et Fonctionnalité blockchain

Chapitre 2

Conception de l'interface web avec WordPress

2.1 Description des fonctionnalités de l'interface

L'interface que nous avons créée offre plusieurs fonctionnalités essentielles pour faciliter les interactions entre les utilisateurs, à la fois fournisseurs de données et fournisseurs de services. Voici une description détaillée de ces fonctionnalités :

Publication de postes : Les utilisateurs fournisseurs de données peuvent publier des postes pour présenter les données qu'ils sont prêts à partager. De même, les utilisateurs fournisseurs de services peuvent publier des postes pour indiquer les services qu'ils offrent. Cette fonctionnalité permet aux deux parties de présenter leurs offres et de faciliter la recherche et la découverte de collaborations potentielles.

Recherche de postes : Les utilisateurs peuvent effectuer des recherches sur l'interface pour trouver des postes pertinents en fonction de leurs besoins. Les fournisseurs de données peuvent rechercher des postes de services qui correspondent à leurs besoins spécifiques, et vice versa. Cela permet aux utilisateurs de trouver les offres qui les intéressent le plus et

de se connecter avec les personnes appropriées.

Fonction de contact : Une fois qu'un utilisateur a trouvé un poste qui l'intéresse, il peut utiliser la fonction de contact pour entrer en communication avec la personne concernée. Cette fonctionnalité permet d'initier des discussions, d'échanger des informations supplémentaires et d'explorer les possibilités de collaboration. Elle facilite la mise en relation directe entre les utilisateurs et favorise les échanges constructifs.

Chat en direct L'interface propose une fonction de chat en direct qui permet aux utilisateurs de communiquer instantanément entre eux. Cette fonctionnalité de messagerie instantanée offre un moyen pratique de discuter en temps réel, de poser des questions, de partager des informations et de clarifier les détails importants. Elle facilite la communication et renforce la collaboration entre les utilisateurs.

Formulaire de création de DAO : L'interface comprend également un formulaire dédié à la création d'un DAO. Les utilisateurs intéressés par la création d'un DAO peuvent remplir ce formulaire en fournissant les informations requises. Ces informations seront utilisées pour configurer et personnaliser le DAO selon les besoins spécifiques de l'utilisateur. Cette fonctionnalité simplifie le processus de création du DAO en guidant les utilisateurs à travers les étapes nécessaires.

En résumé, l'interface que nous avons développée offre des fonctionnalités clés telles que la publication de postes, la recherche de postes, la fonction de contact, le chat en direct et un formulaire dédié à la création de DAO. Ces fonctionnalités per-

mettent aux utilisateurs de trouver des opportunités de collaboration, d'entrer en communication, d'échanger des informations et de faciliter le processus de création d'un DAO.

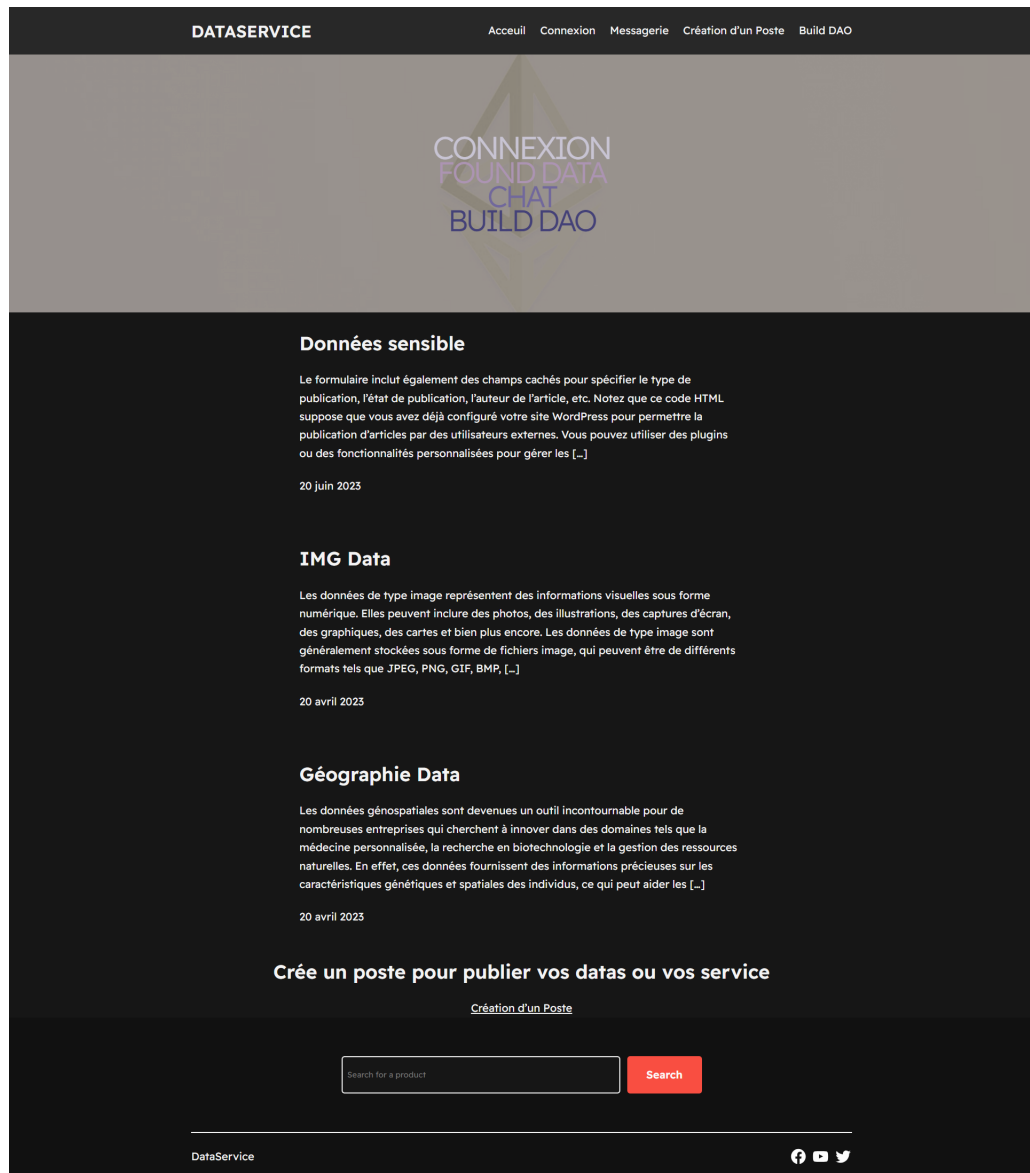


FIGURE 2.1 – Page d'accueil de l'interface

2.2 Explication des choix des outils utilisés

Pour la connexion du portefeuille électronique de cryptomonnaie, nous avons opté pour l'utilisation du plugin "**crypto**" créé

par **ODude**. Ce plugin offre des fonctionnalités avancées pour la gestion des transactions de cryptomonnaie, permettant aux utilisateurs de connecter leur portefeuille électronique et d'effectuer des paiements en toute sécurité. Grâce à sa compatibilité avec différentes cryptomonnaies, ce plugin offre une flexibilité accrue pour les transactions financières au sein de notre application.

En ce qui concerne la fonctionnalité de chat, nous avons choisi d'utiliser le plugin **Front End PM** développé par **Shamim Hassan**. Ce plugin offre une interface conviviale et intuitive pour permettre aux utilisateurs de communiquer en temps réel. Il prend en charge les fonctionnalités de messagerie privée et de groupe, ce qui permet aux utilisateurs d'interagir facilement et de partager des informations importantes. Grâce à sa modularité et à sa personnalisation avancée, ce plugin s'intègre parfaitement à notre interface et répond à nos besoins en matière de communication entre les utilisateurs.

Pour la gestion des fonctionnalités de connexion, d'inscription, de modification de profil et de création de postes, nous avons utilisé le plugin **Profile Builder** développé par **Cozmoslabs**. Ce plugin offre une solution complète pour la gestion des profils utilisateurs, permettant aux utilisateurs de s'inscrire, de se connecter, de personnaliser leur profil et de créer des postes. Grâce à ses fonctionnalités avancées de personnalisation et de contrôle d'accès, ce plugin répond à nos besoins spécifiques en termes de gestion des utilisateurs et des informations associées à leurs profils.

Le code pour la création du formulaire de demande d'information pour la création du dao à été fait en html/css pour une meilleur liberté de personnalisation.

En conclusion, nous avons fait le choix d'utiliser le plugin **"crypto"** pour la connexion du portefeuille électronique de cryptomonnaie, le plugin **Front End PM** pour la fonctionnalité de chat, et le plugin **Profile Builder** pour la gestion des fonctionnalités de connexion, d'inscription, de modification de profil et de création de postes. Ces outils ont été sélectionnés en fonction de leur compatibilité avec nos besoins fonctionnels spécifiques et de leur facilité d'intégration à notre interface. Ils offrent des fonctionnalités avancées et une flexibilité permettant de fournir une expérience utilisateur optimale au sein de notre application.

2.3 Architecture et structure de l'interface

L'architecture et la structure de notre interface sont les suivantes :

- **Page d'accueil** : C'est la page principale de notre application où les utilisateurs peuvent consulter les postes publiés par les fournisseurs de données et les fournisseurs de services. Cette page présente les différentes offres disponibles et permet aux utilisateurs de parcourir les informations et de trouver les collaborations qui les intéressent.
- **Page de connexion** : Cette page permet aux utilisateurs de se connecter à leur compte sur notre application. Ils peuvent utiliser leurs identifiants et mots de passe pour accéder à leur compte. De plus, ils ont la possibilité de connecter leur portefeuille électronique en utilisant l'extension Metamask. Si un utilisateur n'a pas encore de compte, il peut en créer un en remplissant un formulaire d'inscription. De plus, cette page permet également aux utilisateurs de modifier leur pro-

fil existant.

- **Page de messagerie** : Cette page offre une interface conviviale pour la gestion des messages entre les utilisateurs. Les utilisateurs peuvent envoyer des messages privés, discuter en temps réel et créer des annonces pour des groupes de personnes spécifiques. Cette fonctionnalité favorise la communication efficace et facilite les échanges d'informations entre les utilisateurs.
- **Page de création de poste** : Sur cette page, les utilisateurs peuvent remplir un formulaire pour créer un nouveau poste. Le formulaire permet de saisir toutes les informations pertinentes sur le poste, telles que la description, les compétences requises, les délais, etc. Une fois le formulaire soumis, le poste sera examiné et validé par un administrateur avant d'être publié sur le site. Cette étape de validation garantit la qualité des postes publiés.
- **Page de création de DAO** : Cette page contient un formulaire créé en HTML et CSS qui permet aux utilisateurs de saisir les informations nécessaires à la création d'un DAO. Les utilisateurs peuvent fournir des détails tels que le nom du DAO, le pourcentage de vote pour exécuter une proposition, etc. Une fois que le formulaire est rempli, les données sont envoyées à un fichier PHP qui les stocke automatiquement dans un fichier JSON pour que l'administrateur puisse récupérer les informations et procéder à la création du DAO en validant les informations. L'utilisateur sera ensuite informé par e-mail que le DAO a été créé avec succès.

Cette architecture et cette structure permettent aux utilisateurs de naviguer facilement à travers les différentes fonction-

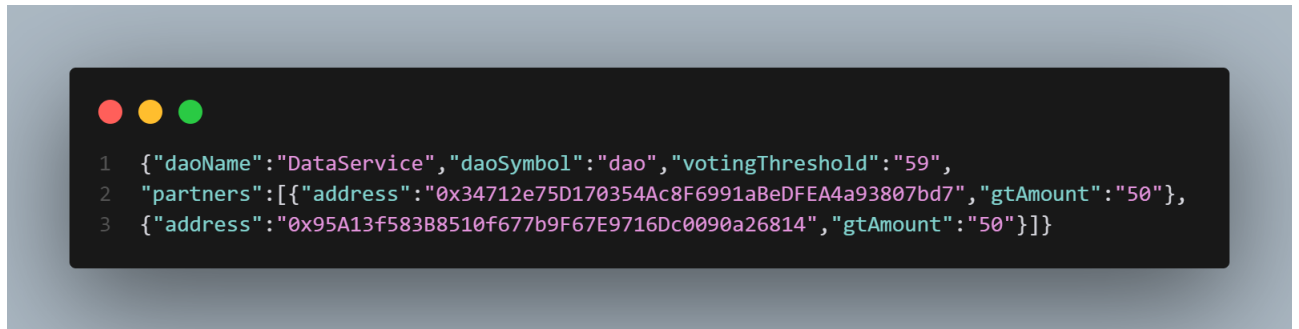


FIGURE 2.2 – Structure du fichier JSON pour la création du DAO

nalités de notre interface. De la consultation des postes à la création de DAO, les utilisateurs peuvent bénéficier d’une expérience fluide et intuitive, favorisant ainsi les collaborations et les échanges au sein de notre plateforme.

Chapitre 3

Développement du DAO sur Ganache

3.1 Présentation de l'environnement de développement Ganache

Ganache est un environnement de développement personnel et convivial pour Ethereum. Il fournit un réseau blockchain local privé qui nous permet de développer, tester et déployer des contrats intelligents sans avoir à nous connecter à un réseau public Ethereum. Voici une présentation de l'environnement de développement Ganache [\[4\]](#) :

Ganache nous offre les fonctionnalités suivantes :

- **Génération de comptes Ethereum** : Ganache génère automatiquement un certain nombre de comptes Ethereum pré-approvisionnés avec des ethers (la cryptomonnaie native d'Ethereum) pour faciliter le développement et les tests. Ces comptes nous permettent de simuler des transactions et de tester nos contrats intelligents sans dépenser de vrais ethers.
- **Réseau blockchain local** : Ganache crée un réseau blockchain local privé sur notre machine. Ce réseau fonctionne de manière similaire à la blockchain Ethereum publique, mais il est isolé et ne nécessite pas de connexion Internet. Cela

nous permet de développer et de tester nos contrats intelligents en toute sécurité, sans craindre les erreurs ou les coûts associés aux interactions sur un réseau public.

- **Interface utilisateur conviviale** : Ganache est doté d'une interface utilisateur conviviale qui affiche des informations détaillées sur les blocs, les transactions, les comptes et les événements de notre réseau local. Nous pouvons facilement suivre l'état de notre blockchain, examiner les transactions effectuées et vérifier l'état de nos contrats intelligents.
- **Débogage et inspection des transactions** : Ganache facilite le processus de débogage de nos contrats intelligents en fournissant des outils d'inspection des transactions. Nous pouvons examiner les détails des transactions, y compris les entrées, les sorties et les événements générés, pour comprendre le comportement de nos contrats et résoudre les problèmes éventuels.
- **Contrôle total de la blockchain** : Ganache nous donne un contrôle total sur notre environnement de développement. Nous pouvons facilement démarrer, arrêter et réinitialiser notre réseau local, ainsi que gérer les comptes, les blocs et les transactions. Cela nous permet d'expérimenter différentes configurations et de simuler divers scénarios pour tester la robustesse de nos contrats intelligents.

En résumé, Ganache est un environnement de développement puissant et convivial pour Ethereum. Il nous permet de créer un réseau blockchain local privé, de générer des comptes Ethereum pré-approvisionnés, de déboguer et d'inspecter des transactions, et de contrôler pleinement notre environnement de développement. Ganache est un outil essentiel pour les développeurs Ethe-

reum, leur permettant de développer et de tester efficacement des applications décentralisées basées sur la blockchain Ethereum.

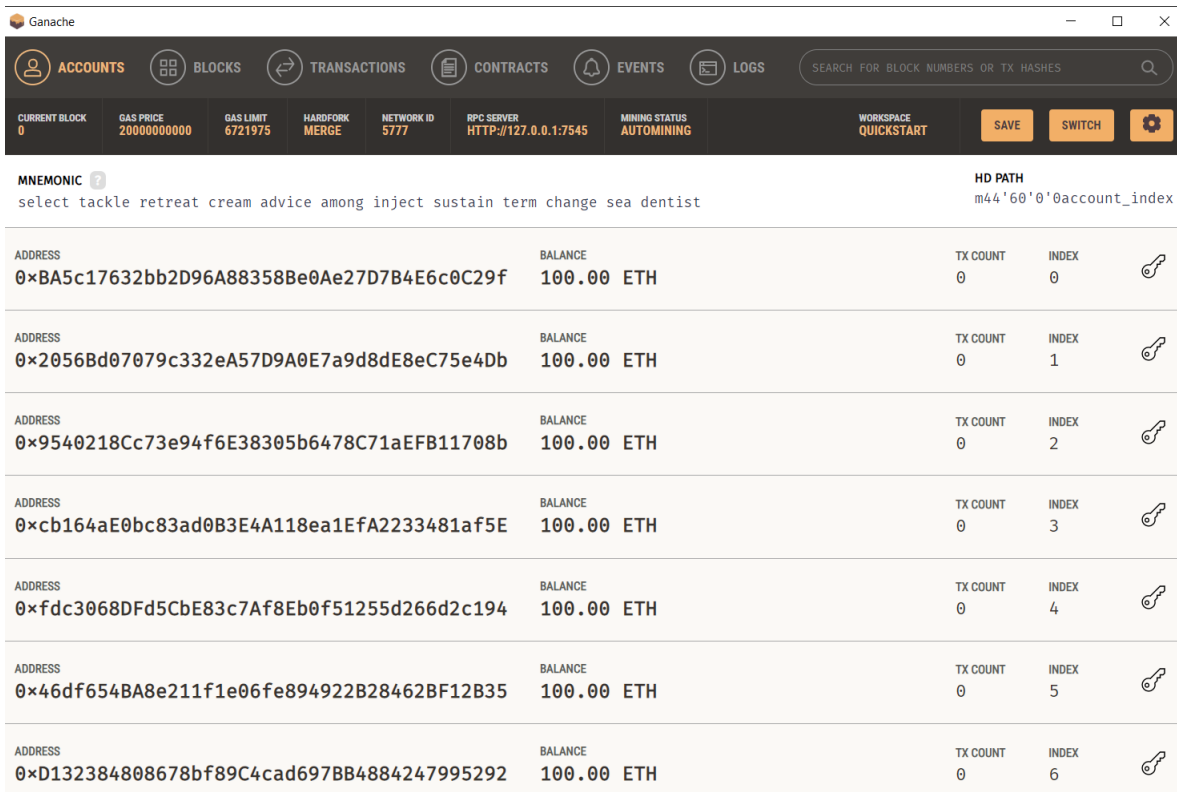


FIGURE 3.1 – Interface de Ganache

3.2 Explication du langage Solidity utilisé pour la création des smart contracts et de Truffle utilisé pour le déploiement

3.2.1 Solidity

Solidity est un langage de programmation utilisé spécifiquement pour créer des contrats intelligents sur la plateforme Ethereum. Il est conçu pour définir les règles et les fonctionnalités des contrats de manière claire et structurée. Voici les points clés de Solidity pour notre projet :

- **Syntaxe similaire à JavaScript** : Solidity utilise une syn-

taxe similaire à JavaScript, ce qui facilite la compréhension et l'apprentissage pour les développeurs familiers avec ce langage.

- **Types de données spécifiques à Ethereum** : Solidity offre des types de données adaptés à la blockchain Ethereum, ce qui permet de manipuler et de stocker les informations nécessaires dans nos contrats intelligents.
- **Interopérabilité avec d'autres contrats intelligents** : Solidity permet une communication et un appel de fonctions entre différents contrats intelligents, ce qui favorise l'interopérabilité et l'intégration entre les parties impliquées.
- **Sécurité et vérification du contrat** : Solidity intègre des mécanismes de sécurité tels que la gestion des exceptions et la vérification des opérations, ce qui aide à créer des contrats intelligents robustes et à éviter les erreurs potentielles.

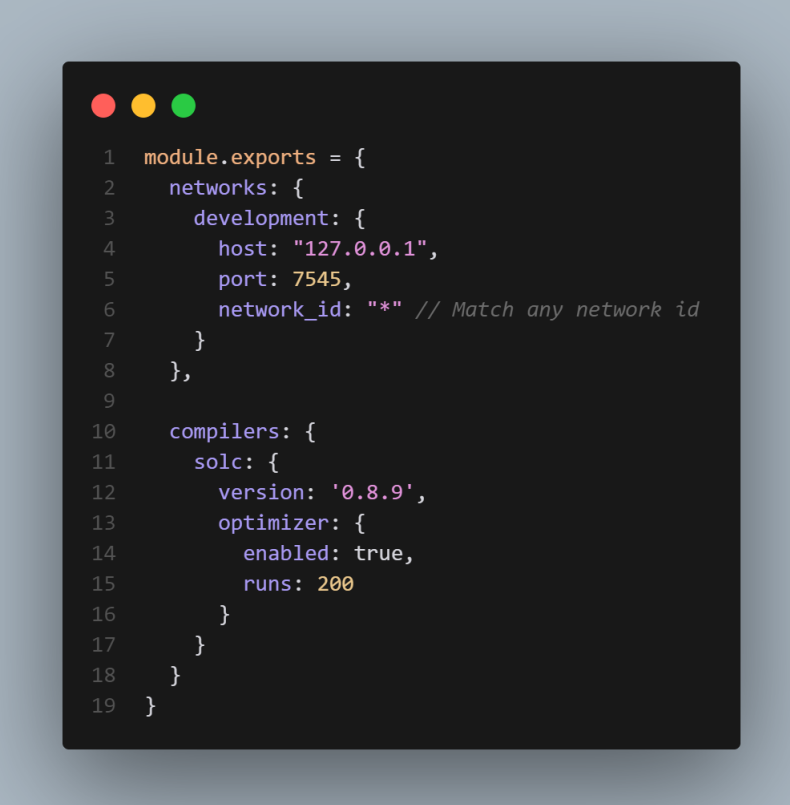
En utilisant Solidity, nous pourrions développer des contrats intelligents qui répondent aux besoins de notre projet, en définissant les règles et les fonctionnalités nécessaires pour établir des interactions sécurisées et fiables entre les fournisseurs de données et les fournisseurs de services.

3.2.2 Turfuffle

Truffle est un outil de développement et de déploiement de contrats intelligents sur la blockchain Ethereum. Il simplifie le processus de création, de compilation, de test et de déploiement de contrats intelligents. Truffle offre également des fonctionnalités avancées telles que la gestion des migrations de contrats et la génération automatique d'interfaces JavaScript pour faciliter l'interaction avec les contrats.

Avec Truffle, nous pouvons configurer notre projet pour déployer nos contrats intelligents sur Ganache en spécifiant les détails de notre réseau Ganache dans le fichier de configuration Truffle. Une fois notre contrat prêt à être déployé, nous utilisons la commande Truffle appropriée pour le déployer sur Ganache. Truffle se charge alors de gérer les détails du déploiement, tels que la création de transactions de déploiement et la mise à jour des adresses des contrats déployés.

En utilisant Truffle et Ganache conjointement, nous développons et déployons nos contrats intelligents de manière efficace et sécurisée sur un réseau Ethereum local pour les tester et les valider avant de les déployer sur un réseau Ethereum réel.



```
1  module.exports = {
2    networks: {
3      development: {
4        host: "127.0.0.1",
5        port: 7545,
6        network_id: "*" // Match any network id
7      }
8    },
9
10   compilers: {
11     solc: {
12       version: '0.8.9',
13       optimizer: {
14         enabled: true,
15         runs: 200
16       }
17     }
18   }
19 }
```

FIGURE 3.2 – Fichier de config Turffle

3.3 Détails sur les smart contracts développés

Nous avons utilisé ce projet 5 smart contrat que nous avons déployer sur la blockchain local ethereum créer avec ganache :

- **Token** : Ce contrat représente une unité de valeur numérique sur la blockchain Ethereum. Il peut être comparé à une sorte de "jeton" virtuel qui peut être transféré entre les utilisateurs. Le contrat **Token** définit les règles pour la création et le transfert de ces jetons, ainsi que la gestion des soldes de chaque utilisateur.
- **Timelock** : Ce contrat agit comme une sorte de coffre-fort avec une fonctionnalité de verrouillage temporel. Il permet de retarder l'exécution d'une action spécifique pendant un certain laps de temps prédéfini. Cela peut être utile pour garantir que certaines actions ne peuvent être effectuées qu'après un délai donné, permettant ainsi aux utilisateurs d'avoir le temps de réagir ou de discuter avant que l'action ne soit réalisée.
- **Governance** : Ce contrat gère la gouvernance d'une organisation décentralisée (DAO) en permettant aux détenteurs de tokens de voter sur des propositions spécifiques. Les propositions peuvent inclure des changements dans les règles de l'organisation, des modifications du contrat ou d'autres décisions importantes. Le contrat **Governance** détermine le quorum requis (pourcentage minimal de participation requis pour qu'une proposition soit valide), la durée du vote et d'autres règles spécifiques à la gouvernance de l'organisation.
- **Treasury** : Ce contrat représente la trésorerie d'une organisation décentralisée. Il agit comme un coffre-fort où les fonds

de l'organisation sont stockés et gérés. Le contrat **Treasury** permet de recevoir des fonds, de les stocker et de les utiliser pour des transactions financières au sein de l'organisation, telles que le paiement de fournisseurs ou de services.

- **Migration** : Ce contrat de migration est utilisé pour déployer et configurer les autres contrats intelligents dans l'environnement Ethereum. Il spécifie les détails de déploiement tels que les paramètres des contrats et les rôles des utilisateurs. Le contrat **Migration** est utilisé dans le cadre du processus de déploiement initial ou de mise à jour des contrats intelligents.

En résumé, ces cinq smart contracts travaillent ensemble pour créer une organisation décentralisée à partir des informations du fichier JSON avec son propre **token**, un système de **gouvernance** permettant aux détenteurs de tokens de voter sur les décisions importantes, une **trésorerie** pour gérer les fonds de l'organisation, et un mécanisme de **verrouillage temporel** pour retarder certaines actions. Ces contrats permettent de mettre en place une structure solide et sécurisée pour la gestion et le fonctionnement d'une organisation basée sur la blockchain Ethereum.

3.4 Utilisation de Truffle pour le déploiement et la gestion du DAO

3.4.1 Déploiement des smart contracts et création du dao

Le code que nous avons fourni effectue un total de 5 déploiements de contrats. Voici une explication détaillée de chaque déploiement :

1. Déploiement du contrat *Token* : Le contrat Token est déployé en utilisant la ligne de code :

```
await deployer.deploy(Token, name, symbol, supply);
```

Cela crée une nouvelle instance du contrat Token avec les paramètres spécifiés (nom, symbole, approvisionnement) et le déploie sur la blockchain.
2. Déploiement du contrat *Timelock* : Le contrat Timelock est déployé en utilisant la ligne de code :

```
await deployer.deploy(Timelock, minDelay, [proposer], [executor]);
```

Cela crée une nouvelle instance du contrat Timelock avec le délai minimum spécifié et les adresses des proposants et des exécutants. Le déploiement de ce contrat permet de gérer les délais et les autorisations d'exécution des propositions dans le système.
3. Déploiement du contrat *Governance* : Le contrat Governance est déployé en utilisant la ligne de code :

```
await deployer.deploy(Governance, token.address, timelock.address, quorum, votingDelay, votingPeriod);
```

Cela crée une nouvelle instance du contrat Governance en spécifiant l'adresse du contrat Token, l'adresse du contrat Timelock, le quorum nécessaire pour approuver les propositions, le délai de vote et la période de vote. Le contrat Governance gère la gouvernance et les décisions dans le système.
4. Déploiement du contrat *Treasury* : Le contrat Treasury est déployé en utilisant la ligne de code :

```
await deployer.deploy(Treasury, executor, value: funds );
```

Cela crée une nouvelle instance du contrat Treasury avec l'adresse de l'exécutant spécifiée et la valeur des fonds spécifiée. Le contrat Treasury est responsable de la gestion des fonds dans le système.

5. Attribution des rôles et affichage des informations du DAO :
Les rôles de proposant et d'exécutant sont attribués au contrat Governance en utilisant les lignes de code :

```
await timelock.grantRole(proposerRole,  
governance.address, from: executor );  
et await timelock.grantRole(executorRole,  
governance.address, from: executor );.
```

Enfin, les informations sur les adresses des contrats Token, Timelock, Governance et Treasury sont affichées dans la console à l'aide des lignes de code :



FIGURE 3.3 – code de déploiement des smart contracts

Les étapes suivies par le code pour réaliser la création du DAO sont :

- Récupère les comptes "executor" et "proposer" à l'aide

de `web3.eth.getAccounts()`.

- Importer les valeurs pour le nom, le symbole les adresses des votants à partir du fichier JSON.
- Définit l’approvisionnement du token.
- Déploie le contrat du token avec les valeurs fournies.
- Transfère une quantité égale de tokens à chaque votant.
- Déploie le contrat Timelock avec une durée minimale de retard.
- Déploie le contrat Governance avec le token, le Timelock et d’autres paramètres.
- Déploie le contrat Treasury avec l’adresse de l’exécuteur et une valeur en ETH.
- Transfère la propriété du Treasury au contrat Timelock.
- Attribue les rôles de proposant et d’exécutant au contrat Governance en utilisant le contrat Timelock.
- Affiche des informations sur le DAO, telles que le nom, le symbole, le nombre de votants et les propriétés de chaque votant.

En résumé, ce code déploie un DAO avec un token, un contrat Timelock, un contrat Governance et un contrat Treasury. Il collecte les adresses des votants, transfère des tokens à chaque votant, et configure les rôles et la propriété du Treasury. Enfin, il affiche des informations sur le DAO créé.

```
DAO Information:
Name: DApp University
Symbol: DAPPU
Number of Voters: 2
Voter 1 Address: 0x691C00f0AA4A3E20D44FED9fac2943204a2fd6Bc
Voter 1 Percentage of Ownership: 50.00%
Voter 2 Address: 0x8e560729f4A6Fc6fCAA4d240fAf48b2728C21B2B
Voter 2 Percentage of Ownership: 50.00%
> Saving migration to chain.
> Saving artifacts
-----
> Total cost:      50.033235538823775584 ETH

Summary
=====
> Total deployments: 5
> Final cost:      50.034079808573775584 ETH
```

FIGURE 3.4 – Exemple de création de DAO

3.4.2 Création d'une proposition, vote et exécution

Le script que nous avons, déploie des contrats, crée une proposition de gouvernance pour libérer des fonds du trésor, effectue des votes sur la proposition, met la proposition en file d'attente et l'exécute. Il vérifie également les fonds disponibles dans le trésor avant et après l'exécution de la proposition.

Le code source fourni est un script JavaScript qui effectue différentes actions liées à une proposition de gouvernance décentralisée. Voici ce que le script fait étape par étape :

1. Importation des contrats "Token", "Timelock", "Governance" et "Treasury" :
 - (a) Les contrats nécessaires pour l'exécution de l'écosystème de gouvernance décentralisée sont importés. Cela permet au script d'interagir avec ces contrats.
2. Récupération des comptes Ethereum :
 - (a) Les adresses des comptes Ethereum sont récupérées à partir de Web3. Ces adresses correspondent aux différents acteurs impliqués dans le processus de gouvernance.

3. Délégation des jetons de vote aux électeurs :
 - (a) Le contrat "Token" est déployé et les électeurs (voter1 à voter5) délèguent leurs jetons de vote à eux-mêmes. Cela permet de déterminer qui peut participer au processus de vote.
4. Vérification des fonds dans le trésor :
 - (a) Le contrat "Treasury" est récupéré et la variable "isReleased" est utilisée pour vérifier si des fonds ont été débloqués dans le trésor.
 - (b) La variable "funds" récupère le solde du trésor en Ether (ETH) et l'affiche.
5. Proposition d'une action de libération des fonds du trésor via le contrat de gouvernance :
 - (a) Le contrat "Governance" est déployé.
 - (b) La fonction "releaseFunds()" du contrat "Treasury" est encodée en utilisant la méthode "encodeABI()" pour obtenir sa représentation en bytes.
 - (c) Une proposition est créée en utilisant la méthode "propose()" du contrat "Governance". La proposition inclut l'adresse du trésor, la valeur à envoyer (0), la fonction encodée, ainsi qu'une description de la proposition.
6. Affichage de l'ID de la proposition créée :
 - (a) L'ID de la proposition créée est récupéré à partir des logs de la transaction et affiché.
7. Vérification de l'état actuel de la proposition (en attente) :
 - (a) L'état actuel de la proposition est récupéré en utilisant la méthode "state()" du contrat "Governance" et affiché. À ce stade, la proposition est en attente.

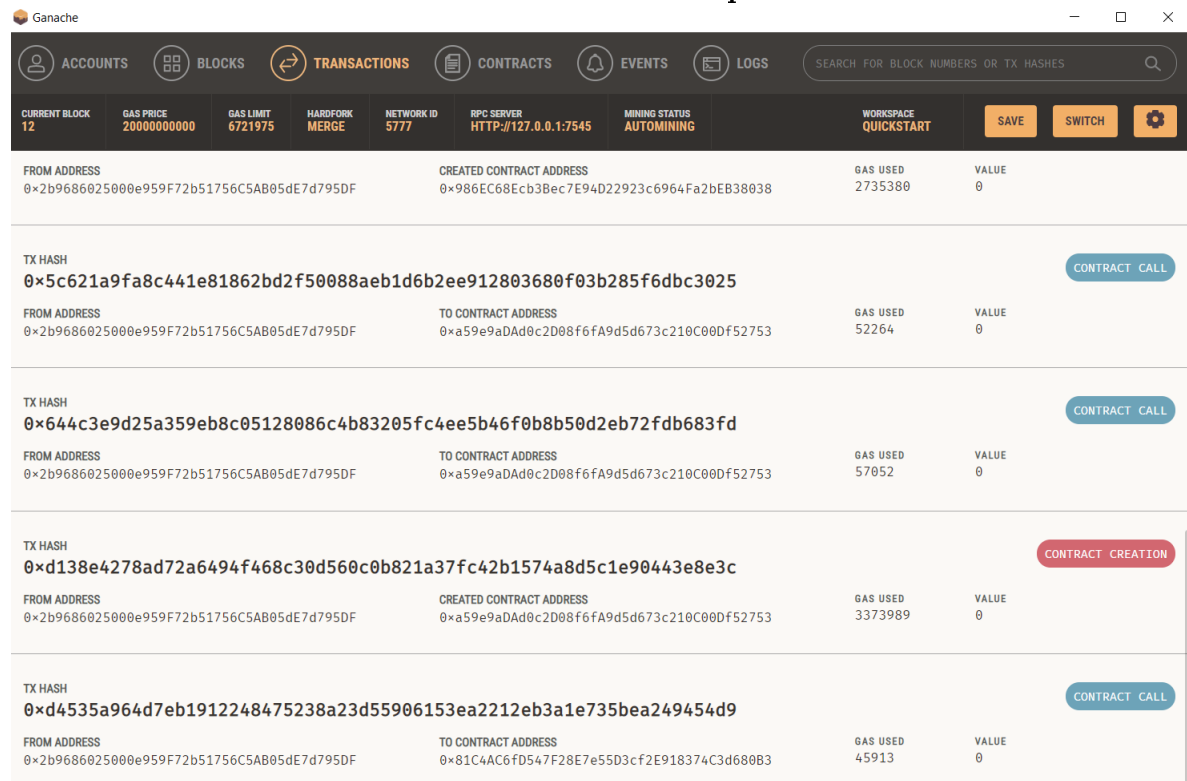
8. Capture de l'instantané de la proposition (numéro de bloc de création) :
 - (a) L'instantané de la proposition est capturé en utilisant la méthode "proposalSnapshot()" du contrat "Governance" et affiché. Cela représente le numéro de bloc auquel la proposition a été créée.
9. Vérification du délai de la proposition (numéro de bloc d'échéance) :
 - (a) Le délai de la proposition est récupéré en utilisant la méthode "proposalDeadline()" du contrat "Governance" et affiché. Cela représente le numéro de bloc à partir duquel la proposition expire.
10. Vote sur la proposition par plusieurs votants :
 - (a) Plusieurs votants expriment leur vote sur la proposition en utilisant la méthode "castVote()" du contrat "Governance". Chaque votant peut voter pour (1), contre (0) ou s'abstenir (2).
 - (b) L'état de la proposition est à nouveau récupéré et affiché. À ce stade, la proposition est active, car des votes ont été exprimés.

3.5 Résultats et évaluation

3.5.1 Tests et validation du DAO

Chaque transaction pour la création de la DAO le virement des fonds de la tresaury l'exécution des proposition est exécuter à la fin d'un vote avec succès et sans erreur.

Transaction lier à l'exécution des contrats pour la création du DAO :



TX HASH	FROM ADDRESS	CREATED CONTRACT ADDRESS	GAS USED	VALUE	ACTION
0x5c621a9fa8c441e81862bd2f50088aeb1d6b2ee912803680f03b285f6dbc3025	0x2b9686025000e959f72b51756c5AB05dE7d795DF	0x986EC68Ec3Bec7E94D22923c6964Fa2bEB38038	2735380	0	CONTRACT CALL
0x644c3e9d25a359eb8c05128086c4b83205fc4ee5b46f0b8b50d2eb72fdb683fd	0x2b9686025000e959f72b51756c5AB05dE7d795DF	0xa59e9aDAd0c2D08f6fA9d5d673c210C00Df52753	52264	0	CONTRACT CALL
0xd138e4278ad72a6494f468c30d560c0b821a37fc42b1574a8d5c1e90443e8e3c	0x2b9686025000e959f72b51756c5AB05dE7d795DF	0xa59e9aDAd0c2D08f6fA9d5d673c210C00Df52753	57052	0	CONTRACT CALL
0xd4535a964d7eb1912248475238a23d55906153ea2212eb3a1e735bea249454d9	0x2b9686025000e959f72b51756c5AB05dE7d795DF	0xa59e9aDAd0c2D08f6fA9d5d673c210C00Df52753	3373989	0	CONTRACT CREATION
0x81C4AC6fD547F28E7e55D3cf2E918374C3d680B3	0x2b9686025000e959f72b51756c5AB05dE7d795DF	0x81C4AC6fD547F28E7e55D3cf2E918374C3d680B3	45913	0	CONTRACT CALL

FIGURE 3.5 – Transaction sur la blockchain

Chapitre 4

Analyse des services Cloud

4.1 Étude comparative entre les différents fournisseurs cloud

Dans cette étude comparative, nous analysons différents fournisseurs de services cloud pour évaluer leur pertinence et leurs capacités pour l'hébergement d'un site WordPress. Nous nous concentrons sur les performances, les fonctionnalités et l'intégration des API pour prendre en compte les besoins spécifiques liés à notre site WordPress. Parmi les fournisseurs examinés, nous avons choisi AWS (Amazon Web Services) en raison de ses API AWS étendues et de l'offre d'AWS Lightsail qui correspond à nos besoins.

4.1.1 Méthodologie :

Nous avons sélectionné plusieurs fournisseurs de services cloud populaires, dont AWS, Google Cloud Platform (GCP) et Microsoft Azure, pour les comparer sur plusieurs critères clés. Ces critères incluent les performances des machines virtuelles, la disponibilité des ressources, la scalabilité, la facilité d'utilisation, les fonctionnalités spécifiques à WordPress, et l'intégration des API pour faciliter l'automatisation et la gestion du site.

4.1.2 Résultats préliminaires :

Les résultats préliminaires de notre étude comparative mettent en évidence les avantages d’AWS dans le contexte de l’hébergement d’un site WordPress. Voici quelques exemples des raisons pour lesquelles AWS peut être préféré par rapport à d’autres fournisseurs :

Étendue des services : AWS offre une vaste gamme de services cloud qui peuvent être facilement intégrés à l’hébergement WordPress. Par exemple, Amazon S3 peut être utilisé pour le stockage des fichiers multimédias, Amazon RDS pour la gestion de la base de données, et Amazon CloudFront pour la mise en cache et la distribution de contenu.

Évolutivité : AWS propose des solutions hautement évolutives, ce qui signifie que votre site WordPress peut facilement s’adapter à la croissance du trafic et aux besoins en ressources. Vous pouvez ajuster la capacité des instances AWS en fonction de la demande, assurant ainsi une performance optimale pour les visiteurs du site.

Facilité d’utilisation : AWS Lightsail, une offre d’AWS, simplifie grandement le déploiement et la gestion d’un site WordPress. Il offre des plans préconfigurés avec des ressources équilibrées pour le fonctionnement fluide de WordPress. De plus, Lightsail propose une interface utilisateur conviviale pour faciliter la configuration et la gestion du site.

Intégration des API : Les API AWS sont puissantes et permettent d’automatiser diverses tâches liées à l’hébergement WordPress, telles que la création et la gestion d’instances, la gestion des sauvegardes, et bien d’autres. Cette intégration facilite le déploiement, la gestion et la mise à l’échelle de votre site.

À ce stade, les résultats préliminaires suggèrent qu’AWS avec

AWS Lightsail est une option solide pour l'hébergement d'un site WordPress, grâce à ses performances, à la facilité d'utilisation de ses services et à son intégration étendue des API. Cependant, une analyse plus approfondie et une comparaison détaillée avec d'autres fournisseurs cloud sont nécessaires pour une évaluation complète.

4.2 Les services cloud : AWS

Amazon S3 : Amazon S3 (Simple Storage Service) est un service de stockage d'objets offert par Amazon Web Services (AWS). Il fournit une solution de stockage hautement évolutive, sécurisée et durable pour stocker et récupérer vos données dans le cloud.^[5]

Voici ce que S3 permet de faire dans le cloud :

- **Stockage d'objets :** Amazon S3 vous permet de stocker et de gérer des objets de n'importe quelle taille, allant de quelques octets à plusieurs téraoctets. Les objets sont stockés dans des seaux (buckets) et organisés en utilisant des clés uniques.
- **Durabilité des données :** Amazon S3 réplique automatiquement vos objets dans différentes installations physiques, ce qui garantit une haute durabilité de vos données. Le taux de durabilité annoncé est de 99,999999999
- **Disponibilité élevée :** S3 est conçu pour fournir une disponibilité élevée des données. Il garantit une disponibilité de 99,99
- **Sécurité des données :** Amazon S3 offre des fonctionnalités de sécurité avancées pour protéger vos données, telles que le contrôle d'accès basé sur les stratégies, le chiffrement des données en transit et au repos, ainsi que la gestion des clés de chiffrement.

- **Gestion du cycle de vie des données :** S3 vous permet de configurer des règles de cycle de vie pour gérer automatiquement vos données. Vous pouvez définir des actions pour déplacer, archiver ou supprimer des objets en fonction de leur âge ou d'autres critères.
- **Partage des données :** Vous pouvez partager vos objets stockés dans S3 avec d'autres utilisateurs en leur accordant des autorisations appropriées. Vous pouvez également utiliser des URL de pré-signature pour permettre un accès temporaire aux objets sans avoir à partager vos informations d'identification.
- **Intégration avec d'autres services AWS :** S3 s'intègre étroitement avec d'autres services AWS tels que Lambda, DynamoDB, Glacier, Redshift, et bien d'autres encore, vous permettant de construire des applications et des architectures cloud puissantes et évolutives.

Lors de la mise en œuvre de cette solution cloud, nous avons utilisé plusieurs services cloud pour répondre aux besoins spécifiques. Voici une analyse des services cloud que nous avons utilisés :

AWS Lightsail : Nous avons utilisé AWS Lightsail pour héberger notre site WordPress. Lightsail est un service d'hébergement simplifié qui fournit des machines virtuelles préconfigurées avec des ressources matérielles dédiées. Il nous a permis de déployer rapidement et facilement notre site WordPress sans avoir à gérer l'infrastructure sous-jacente.

Service de machines virtuelles : Nous avons créé des machines virtuelles pour chaque fournisseur de données et fournisseur de services à l'aide de notre script. Les machines virtuelles ont fourni un environnement isolé pour exécuter les services de

chaque fournisseur de manière sécurisée. Cela a permis de prévenir les accès non autorisés et de protéger les données sensibles.

Quelles sont les différences entre Amazon EC2 et Amazon Lightsail ?

Fonctionnalité/Aspect	Amazon EC2	Amazon Lightsail
Modèle de tarification	Paieement à l'utilisation	Plans mensuels fixes
Capacité de calcul	Évolutif	Tailles d'instance fixes
Performance réseau	Réseau haute performance	Performances réseau limitées
Auto Scaling	Prise en charge de l'Auto Scaling	Pas de prise en charge de l'Auto Scaling
Équilibrage de charge	Prise en charge de l'équilibrage de charge	Pas de prise en charge de l'équilibrage de charge
Bases de données gérées	Prise en charge des bases de données gérées	Prise en charge limitée des bases de données
Gestion DNS	Intégré à Route 53	Gestion DNS limitée
Sauvegarde	Sauvegardes manuelles	Instantanés automatiques
Surveillance	Intégré à CloudWatch	Capacités de surveillance limitées
IP élastique	Prise en charge des IP élastiques	Adresse IP statique incluse
Firewall	Groupe de sécurité	Capacités de pare-feu limitées

TAB 4.1 – Tableau des différences entre Amazon EC2 et Amazon Lightsail

4.2.1 Les Services proposés par AWS Lightsail :

Amazon Lightsail fournit des ressources cloud faciles à utiliser pour rendre opérationnels nos applications ou sites web en quelques clics. Lightsail offre des services simplifiés, tels que les instances, les conteneurs, les bases de données, le stockage et plus encore. Avec Lightsail, nous pouvons facilement lancer des sites ou applications web à l'aide de modèles préconfigurés, comme WordPress, Prestashop ou LAMP. Nous pouvons utiliser les fonctions de Lightsail pour héberger du contenu statique, connecter notre contenu à notre public à travers le monde ou rendre opérationnel notre serveur Windows Business, et ce en toute simplicité. La console Lightsail nous guide tout au long du processus de configuration et, dans bien des cas, les composants sont déjà configurés.

Instances : Lightsail offre des serveurs virtuels (instances) faciles à configurer et optimisés grâce à la puissance et à la fiabilité d'AWS. Nous pouvons non seulement lancer notre site/application web ou notre projet en quelques minutes, mais aussi gérer notre instance depuis l'API ou la console intuitive Lightsail.

Pendant que nous créons notre instance, Lightsail nous permet de cliquer pour lancer un système d'exploitation (OS) simple, une application préconfigurée ou une pile de développement, par exemple WordPress, Windows, Plesk, LAMP, Nginx et plus encore. Chaque instance de Lightsail est dotée d'un pare-feu intégré qui nous permet d'autoriser ou de restreindre le trafic vers nos instances en fonction de l'adresse IP source, du port et du protocole.

Bases de données gérées : Lightsail fournit un plan de bases de données MySQL ou PostgreSQL entièrement configurées qui

inclut l'allocation de capacité de mémoire, de traitement, de stockage et de transfert. Avec les bases de données gérées Lightsail, nous pouvons facilement mettre à l'échelle nos bases de données indépendamment de nos serveurs virtuels, améliorer la disponibilité des applications ou exécuter des bases de données autonomes dans le cloud. Nous pouvons également déployer des applications à plusieurs niveaux entièrement au sein de Lightsail, en créant plusieurs instances connectées à une base de données gérée centrale, ainsi qu'un équilibreur de charge qui achemine le trafic vers les instances.

4.3 Intégration avec le cloud AWS

4.3.1 Configuration et déploiement des machines virtuelles pour le DAO

Nous avons choisi AWS Lightsail comme solution d'hébergement pour notre site WordPress en raison de plusieurs raisons. Lightsail offre une interface conviviale et simplifiée qui facilite le déploiement d'un site WordPress. De plus, il offre une tarification transparente avec des forfaits mensuels fixes, ce qui nous permet de contrôler nos coûts. Nous avons également été attiré par l'intégration transparente de Lightsail avec les autres services AWS, nous offrant ainsi la possibilité d'étendre notre infrastructure en fonction de nos besoins futurs.

4.3.2 Création de notre instance Lightsail :

Une fois que nous avons opté pour Lightsail, nous avons créé une instance Lightsail pour héberger notre site WordPress. Lors de la création de l'instance, nous avons pu choisir la taille de l'instance en fonction des ressources nécessaires pour notre site.

De plus, nous avons configuré le nom de l'instance, le plan de paiement, les options de mise en réseau, y compris l'assignation d'une adresse IP publique, et nous avons ajouté les clés SSH nécessaires pour la connexion sécurisée à notre instance. Notre site web est lié avec une adresse IP public : `http://15.236.142.118/`

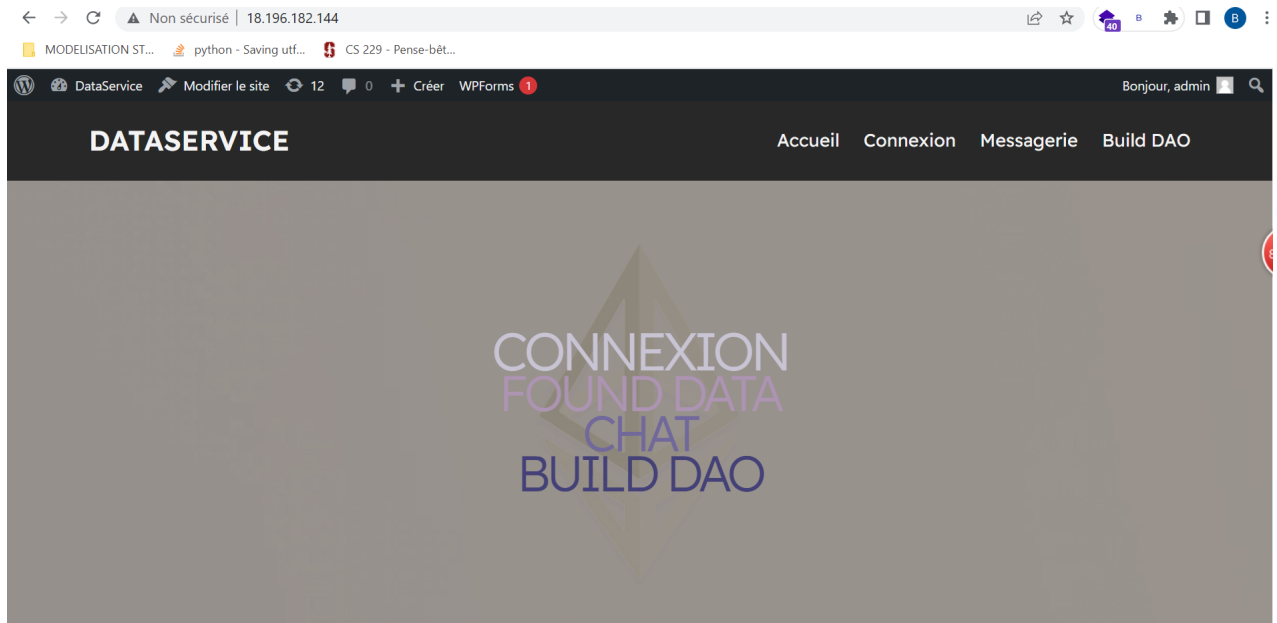


FIGURE 4.1 – adresse IP Public

4.3.3 Installation de WordPress :

Après avoir créé notre instance Lightsail, nous avons utilisé SSH pour nous connecter à notre instance. Ensuite, nous avons installé WordPress en suivant les instructions fournies par Lightsail. Cela comprenait le téléchargement et la configuration des fichiers WordPress, la création d'une base de données pour stocker nos données et la configuration des paramètres de base tels que le titre du site, l'adresse e-mail de l'administrateur, etc.



FIGURE 4.2 – service lightsail

4.3.4 Migration des données :

- **Préparation de l'environnement local :** Avant de migrer notre site WordPress vers AWS Lightsail, nous devons nous assurer que notre site fonctionne correctement dans notre environnement local. Cela inclut la vérification de la compatibilité de notre site avec la version de WordPress prise en charge par Lightsail, ainsi que la sauvegarde de toutes les données et fichiers importants de notre site.
- **Exportation des données depuis l'environnement local :** Dans notre environnement local, nous utilisons l'outil d'exportation intégré de WordPress pour exporter toutes les données de notre site, y compris les articles, les pages, les paramètres, les médias, etc. L'exportation se fait généralement sous la forme d'un fichier avec le plugin de WordPress "all-in-One WP Migration".
- **Importation des données dans Lightsail :** Une fois que nous avons exporté les données depuis notre environnement local, nous pouvons les importer dans notre instance Lightsail. Nous accédons à l'interface d'administration de notre

site WordPress sur Lightsail et utilisons l'outil d'importation intégré pour importer le fichier XML contenant nos données. L'outil effectuera la conversion et l'importation des données dans la base de données MySQL de notre instance Lightsail.

- **Configuration des paramètres du site :** Après l'importation des données, nous devons configurer les paramètres spécifiques à notre site, tels que les permaliens, les plugins, les thèmes, etc. Assurons-nous également de mettre à jour les informations de connexion à la base de données pour refléter les informations d'identification de notre base de données Lightsail.

4.3.5 Configuration de la base de données :

Une fois que notre instance Lightsail est créée, nous devons configurer une base de données pour notre site WordPress. Lightsail offre une option intégrée pour créer une base de données MySQL compatible avec WordPress. Lors de la configuration de la base de données, assurons-nous de noter les informations d'identification, telles que le nom d'utilisateur et le mot de passe, car nous en aurons besoin pour la migration des données.

Avec AWS Lightsail nous avons facilement pu lier le site web avec une base de données MySQL



FIGURE 4.3 – Base de données Lightsail

4.3.6 Génération du fichier JSON :

Lorsqu'un utilisateur soumet le formulaire, les données saisies sont traitées par WordPress. À ce stade, nous avons pu utiliser des extensions ou des scripts personnalisés pour générer un fichier JSON à partir des données du formulaire. Ce fichier JSON contient toutes les informations collectées, structurées dans un format lisible par les machines.

4.3.7 Création un script d'automatisation :

Utilisation d'un langage de programmation comme Python pour écrire un script qui lit le fichier JSON et utilise les bibliothèques AWS SDK (Software Development Kit) pour interagir avec les services AWS. Dans ce cas, nous pouvons utiliser la bibliothèque Boto3, qui est l'API Python officielle pour AWS.

4.3.8 Configuration des informations d'identification :

Pour que notre script puisse accéder à AWS Lightsail, nous devons configurer les informations d'identification. Nous pouvons le faire en utilisant les clés d'accès IAM (Identity and Access Management) d'un utilisateur avec les autorisations appropriées pour créer des instances Lightsail.

4.3.9 Analysez le fichier JSON :

Utilisation du script pour analyser le fichier JSON et extraire les informations nécessaires, telles que le nom de l'instance, la région, le plan de tarification, etc.

4.3.10 Utilisation de l'API Lightsail :

Une fois que nous avons extrait les informations du fichier JSON, utilisons l'API Boto3 pour créer une instance Lightsail en utilisant les paramètres correspondants. Nous pouvons spécifier des options supplémentaires telles que les clés SSH, les scripts de démarrage, etc.

4.3.11 création d'un compartiment "dataproviders3" dans s3

La création d'un compartiment (bucket) dans Amazon S3 offre plusieurs avantages et cas d'utilisation dans le cloud. Parmi eux :

- **Stockage et archivage de données** : S3 nous permet de stocker et d'archiver de grandes quantités de données de manière durable et hautement disponible. En créant un compartiment dans S3, nous pouvons organiser et structurer nos données en fonction de nos besoins spécifiques.
- **Partage de fichiers et de données** : Les compartiments S3 peuvent être utilisés pour partager des fichiers et des données avec d'autres utilisateurs ou des systèmes externes. Nous pouvons définir des autorisations granulaires pour contrôler qui peut accéder aux fichiers et comment ils peuvent interagir avec eux.

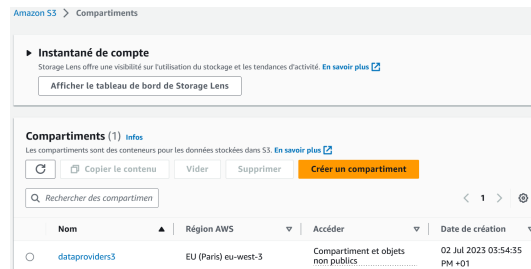


FIGURE 4.4 – compartiment dataproviders3 dans S3

4.3.12 Stocker le file json dans s3 :

```
aws s3 cp data.json s3 ://dataproviders3/donnee.json
```

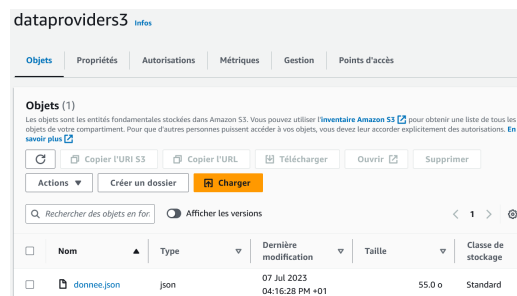


FIGURE 4.5 – stockage des données JSON

4.3.13 Creation user dataprovider et sa permission :

Nom de la politique	Type	Attaché via
allow_awscli	Gérées par le client	Directement
Data_Provider_Policy	Gérées par le client	Directement
IAMUserChangePassword	Gérées par AWS	Directement
instance_lightsail	Gérées par le client	Directement
lightsail_FULL	Gérées par le client	Directement

FIGURE 4.6 – Permissions user dataprovider

4.3.14 Creation user serviceprovider et sa permission :

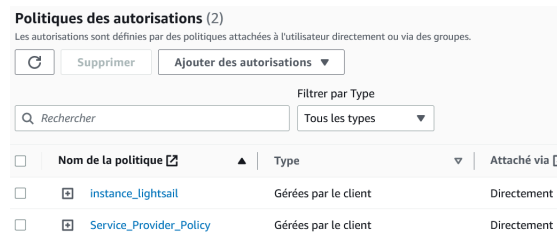


FIGURE 4.7 – Permissions user serviceprovider

Nous avons créé deux utilisateurs avec des autorisations différentes dans le but de restreindre les actions que chaque utilisateur peut effectuer sur Amazon S3.

L'utilisateur "dataproducer" dispose de permissions étendues, lui permettant d'effectuer des opérations telles que l'ajout, la modification et la suppression de fichiers dans le compartiment S3. Cette autorisation est appropriée si nous souhaitons donner à ce fournisseur de données le contrôle total sur les fichiers stockés dans S3, leur permettant de gérer les données de manière complète.

D'un autre côté, l'utilisateur "serviceprovider" a des permissions limitées. Ils sont autorisés uniquement à visualiser les fichiers stockés dans le compartiment S3, sans possibilité de les modifier ou de les supprimer. Cette restriction est mise en place pour garantir que le fournisseur de services ne peut pas accidentellement supprimer ou modifier des fichiers importants dans S3, préservant ainsi l'intégrité des données.

En séparant les autorisations de cette manière, nous créons une stratégie de sécurité plus fine pour gérer les accès aux ressources. Cela permet de limiter les risques potentiels en contrôlant strictement les actions que chaque utilisateur peut entreprendre. Cela peut également faciliter la gestion des autorisations, car nous pouvons accorder des privilèges spécifiques à chaque utilisateur

en fonction de leurs responsabilités et besoins.

4.4 Explication des avantages et des contraintes de l'utilisation du cloud

4.4.1 Avantages de l'utilisation du cloud dans le DAO :

- Évolutivité et flexibilité : Le cloud offre une grande échelle et une flexibilité pour répondre aux besoins changeants de notre DAO. Nous pouvez facilement augmenter ou réduire les ressources informatiques en fonction de la demande, ce qui nous permet de faire évoluer notre DAO sans contraintes matérielles.
- Disponibilité élevée : Les fournisseurs de cloud proposent des architectures hautement disponibles qui garantissent la disponibilité continue de nos services. Notre DAO peut fonctionner 24h/24 et 7j/7, assurant une accessibilité constante aux utilisateurs et aux fournisseurs de données.
- Sauvegarde et récupération des données : Les services de cloud offrent des fonctionnalités intégrées de sauvegarde et de récupération des données. Nous pouvons facilement mettre en place des stratégies de sauvegarde automatisées pour protéger les données critiques de notre DAO, minimisant ainsi les risques de perte de données.
- Sécurité renforcée : Les fournisseurs de cloud mettent en place des mesures de sécurité avancées pour protéger nos ressources et données. Ils offrent des pare-feu, des contrôles d'accès, des options de chiffrement et des outils de surveillance pour renforcer la sécurité de notre DAO.
- Coûts optimisés : L'utilisation du cloud permet de réduire les coûts d'infrastructure. Nous payons uniquement pour les

ressources que nous consommez, ce qui nous permet d'ajuster les dépenses en fonction des besoins réels de notre DAO. De plus, nous évitons les investissements initiaux élevés associés à l'achat de matériel informatique.

4.4.2 Contraintes de l'utilisation du cloud dans le DAO :

- Dépendance au fournisseur de cloud : L'utilisation du cloud implique une dépendance à l'égard du fournisseur de services cloud choisi. Nous devons prendre en compte les politiques de tarification, les conditions de service et les éventuelles limitations techniques imposées par le fournisseur.
- Risques de sécurité : Bien que les fournisseurs de cloud mettent en place des mesures de sécurité robustes, il existe toujours des risques potentiels tels que les violations de données ou les attaques de sécurité. Il est important de mettre en œuvre des pratiques de sécurité appropriées et de surveiller en permanence la sécurité de notre DAO.
- Connexion Internet nécessaire : L'utilisation du cloud nécessite une connexion Internet stable et rapide. Si notre connexion Internet est interrompue, cela peut entraîner une interruption des services et de la communication au sein de notre DAO.
- Gestion des coûts : Bien que l'utilisation du cloud permette d'optimiser les coûts, il est important de surveiller et de gérer les dépenses cloud. Une mauvaise gestion des ressources peut entraîner des coûts élevés et inutiles.
- Conformité réglementaire : Si notre DAO manipule des données sensibles ou est soumis à des réglementations spécifiques, nous devons nous assurer que le fournisseur de cloud choisi respecte les normes de conformité nécessaires.

Conclusion et perspectives Dans le cadre des travaux futurs, il serait intéressant d'explorer et d'implémenter des outils d'automatisation pour simplifier et améliorer la gestion de l'infrastructure du système, tels que l'utilisation de technologies telles que l'orchestration de conteneurs (par exemple, Kubernetes) ou les outils de gestion de configuration (par exemple, Ansible) pour automatiser les tâches de déploiement, de mise à l'échelle et de gestion des ressources. L'objectif serait de réduire les efforts manuels nécessaires et d'optimiser les performances de l'infrastructure. Une autre piste de développement future consisterait à explorer les possibilités de transactions entre les réseaux blockchain privés tels que Hyperledger Fabric ou Ganache et l'infrastructure cloud, en exploitant les avantages de la blockchain pour garantir la sécurité et la traçabilité des transactions, tout en bénéficiant des capacités évolutives et flexibles offertes par l'infrastructure cloud. Il serait important d'explorer les protocoles et les mécanismes nécessaires pour faciliter ces transactions et d'assurer une intégration fluide entre les différents environnements. Enfin, il serait essentiel de continuer à rechercher des perspectives d'amélioration et d'optimisation du système, notamment en identifiant et en résolvant les goulots d'étranglement, en optimisant les performances, en améliorant l'expérience utilisateur, en ajoutant des fonctionnalités supplémentaires en fonction des besoins des utilisateurs, et en explorant de nouvelles technologies émergentes. L'objectif global serait de faire évoluer et de continuer à améliorer le système pour qu'il réponde de manière optimale aux exigences changeantes du marché et aux attentes des utilisateurs.

En conclusion, cette étude préliminaire nous a permis d'approfondir notre compréhension des DAO et de leurs applications potentielles. Nous avons identifié les outils et les technologies né-

cessaires pour créer un DAO, et nous avons discuté des fonctionnalités clés de l'interface utilisateur. Bien que des défis et des limitations subsistent, le potentiel des DAO pour transformer divers secteurs de l'économie est prometteur. Nous espérons que cette étude préliminaire servira de base solide pour des travaux futurs et des développements plus avancés dans le domaine des DAO.

Bibliographie

- [1] V. Buterin, “Ethereum : A next-generation smart contract and decentralized application platform,” in *Proceedings of the Bitcoin Workshop*. The Cryptocurrency Research Group, 2014, pp. 3–16. [Online]. Available : https://blockchainworkshop.org/2014/papers/bitcoin2014_submission_9.pdf
- [2] Ethereum Foundation, “Solidity : A contract-oriented programming language,” <https://soliditylang.org/>, 2023, accessed : July 14, 2023.
- [3] S. Nakamoto, “Bitcoin : A peer-to-peer electronic cash system,” *Bitcoin.org*, 2008. [Online]. Available : <https://bitcoin.org/bitcoin.pdf>
- [4] Truffle Suite, “Ganache,” 2023, accessed : July 14, 2023.
- [5] “Amazon web services,” <https://aws.amazon.com/>, Amazon, 2023, accessed : July 14, 2023.