Mémoire d'ingénieur

Implémentation d'un service de liste de confiance globale basé sur la blockchain

Yoann Raucoules

Année 2016–2017

Stage de fin d'études réalisé dans l'entreprise ARHS Spikeseed

en vue de l'obtention du diplôme d'ingénieur de TELECOM Nancy

Maître de stage : Vincent Bouckaert

Encadrant universitaire : Olivier Festor

Déclaration sur l'honneur de non-plagiat

Je soussigné(e),

Nom, prénom : Raucoules, Yoann

Élève-ingénieur(e) régulièrement inscrit(e) en 3e année à TELECOM Nancy

Numéro de carte de l'étudiant(e) : 1205028998

Année universitaire : 2016–2017

Auteur(e) du document, mémoire, rapport ou code informatique intitulé :

Implémentation d'un service de liste de confiance globale basé sur la blockchain

Par la présente, je déclare m'être informé(e) sur les différentes formes de plagiat existantes et sur les techniques et normes de citation et référence.

Je déclare en outre que le travail rendu est un travail original, issu de ma réflexion personnelle, et qu'il a été rédigé entièrement par mes soins. J'affirme n'avoir ni contrefait, ni falsifié, ni copié tout ou partie de l'œuvre d'autrui, en particulier texte ou code informatique, dans le but de me l'accaparer.

Je certifie donc que toutes formulations, idées, recherches, raisonnements, analyses, programmes, schémas ou autre créations, figurant dans le document et empruntés à un tiers, sont clairement signalés comme tels, selon les usages en vigueur.

Je suis conscient(e) que le fait de ne pas citer une source ou de ne pas la citer clairement et complètement est constitutif de plagiat, que le plagiat est considéré comme une faute grave au sein de l'Université, et qu'en cas de manquement aux règles en la matière, j'encourrais des poursuites non seulement devant la commission de discipline de l'établissement mais également devant les tribunaux de la République Française.

Fait à Luxembourg, le 21 août 2017

Signature :

Mémoire d'ingénieur

Implémentation d'un service de liste de confiance globale basé sur la blockchain

Yoann Raucoules

Année 2016–2017

Stage de fin d'études réalisé dans l'entreprise ARHS Spikeseed

en vue de l'obtention du diplôme d'ingénieur de TELECOM Nancy

Yoann Raucoules
6, rue du général Frère
57070, METZ
+33 (0)6 77 48 04 38
yoann.raucoules@telecomnancy.eu

TELECOM Nancy
193 avenue Paul Muller,
CS 90172, VILLERS-LÈS-NANCY
+33 (0)3 83 68 26 00
contact@telecomnancy.eu

ARHS Spikeseed
2B, rue Nicolas Bové
1253, LUXEMBOURG
+352 26 11 02 1

Maître de stage : Vincent Bouckaert

Encadrant universitaire : Olivier Festor

Remerciements

"Night gathers, and now my watch begins.
It shall not end until my death.

I shall take no wife, hold no lands, father no children.
I shall wear no crowns and win no glory.
I shall live and die at my post.

I am the sword in the darkness.
I am the watcher on the walls.
I am the shield that guards the realms of men.

I pledge my life and honor to the Night's Watch,
for this night and all the nights to come."

– The Night's Watch oath

Avant-propos

Ce mémoire résulte d'un stage de fin d'études qui s'est déroulé du 3 avril 2017 au 30 septembre 2017 au sein de l'entreprise Arηs Spikeseed située au Luxembourg. Ce stage vient clôturer et valider la formation d'ingénieur du numérique de l'école TELECOM Nancy que j'ai débuté en septembre 2014. Cette formation qui s'est étendue sur une période de trois ans m'a permis d'acquérir de nombreuses compétences dans les domaines de l'informatique, des mathématiques, du management, de la gestion de projet, de la communication, de l'économie, du droit et des langues. J'ai choisi de me spécialiser en Ingénierie Logicielle au cours du cursus de par ma passion pour la programmation et l'architecture logicielle depuis que j'ai découvert l'informatique lors de mon stage de découverte professionnelle réalisé en classe de troisième.

Au cours de ce stage de fin d'études, j'ai eu le plaisir de travailler sur une technologie à laquelle je m'intéresse depuis deux ans, la blockchain. Dans le cadre d'un projet proposé par la Commission Européenne, nommé FutureTrust, j'ai pu concevoir et implémenter un service de trust list global basé sur la blockchain. Mes tâches ont été de me familiariser avec les principes de la blockchain et les concepts de cryptographie appliquée afin de les mettre en application dans le projet, d'effectuer une analyse des solutions de blockchain existantes afin de réaliser des choix d'implémentation, de concevoir l'architecture du service de liste de confiance globale, d'implémenter la solution conçue et de documenter tous les aspects techniques et fonctionnels de la solution implémentée.

Dans ce mémoire est présenté le résultat du stage de fin d'études et est mis en avant l'utilisation de la blockchain dans le cadre d'un projet de confiance numérique d'échelle mondiale. L'intérêt de ce document est dans un premier temps d'expliquer les tâches réalisées au cours du stage et dans un second temps de montrer qu'il est possible d'élargir le champ d'application de la technologie blockchain et des différents aspects qui la composent.

Table des matières

1 Introduction

La technologie blockchain s'est popularisée ces dernières années grâce à l'expansion de la crypto-monnaie [1] Bitcoin [2] [10] à travers le monde. En effet, cette technologie a bouleversé aussi bien le monde de l'informatique que le monde de la finance. L'investissement autour de la blockchain a mené à un engouement général pour ce concept. Le Bitcoin a réussi à remettre en cause des acteurs majeurs de notre société tels que les banques ou les géants du Web, en sécurisant des échanges d'actifs sans organe central de contrôle. La révolution qu'il a engendré amène aujourd'hui les gouvernements et autres organisations publiques à réfléchir sur la régulation de la technologie et des crypto-monnaies naissantes. Depuis son lancement en 2009, la blockchain n'a cessé d'évoluer et d'étendre son champ d'application. Bien qu'à l'origine elle a été conçue pour le transfert de crypto-monnaie, les avantages qu'elle apporte permettent d'imaginer de multiples cas d'utilisation qui dépassent son cadre initial d'échanges d'actifs. À l'heure où l'ubérisation [3] de notre société est en marche, la technologie blockchain amène une approche nouvelle qui permet de se détacher de toute organe central ou tierce partie. La blockchain ira-t-elle jusqu'à ubériser [4] Uber [5] ?

1.1 Présentation de la technologie blockchain

Une blockchain est basée sur l'échange d'actifs numériques, réalisé grâce à des transactions signées, et agit comme un registre publique distribué où toutes les transactions y sont répertoriées. Elle repose sur des principes de cryptographie afin d'assurer l'intégrité de ces transactions et sur un protocole décentralisé, dit *peer-to-peer*, qui permet à la blockchain d'avoir une disponibilité maximale et d'établir un consensus entre les participants du réseau afin de protéger contre les falsifications. La Figure 1.1 représente le processus d'émission et de validation d'une transaction sur la blockchain.

1. La crypto-monnaie aussi appelée monnaie cryptographique est une monnaie électronique basé sur les principes de la cryptographie.
2. Bitcoin est une crypto-monnaie et un système de paiement pair-à-pair.
3. L'ubérisation est un phénomène économique désignant l'utilisation de services permettant aux professionnels et aux clients de se mettre en contact direct grâce à l'utilisation des nouvelles technologies.
4. Ubériser est le verbe issu du substantif ubériser.
5. Uber est l'entreprise qui déclencha ce qu'on appelle l'ubérisation.

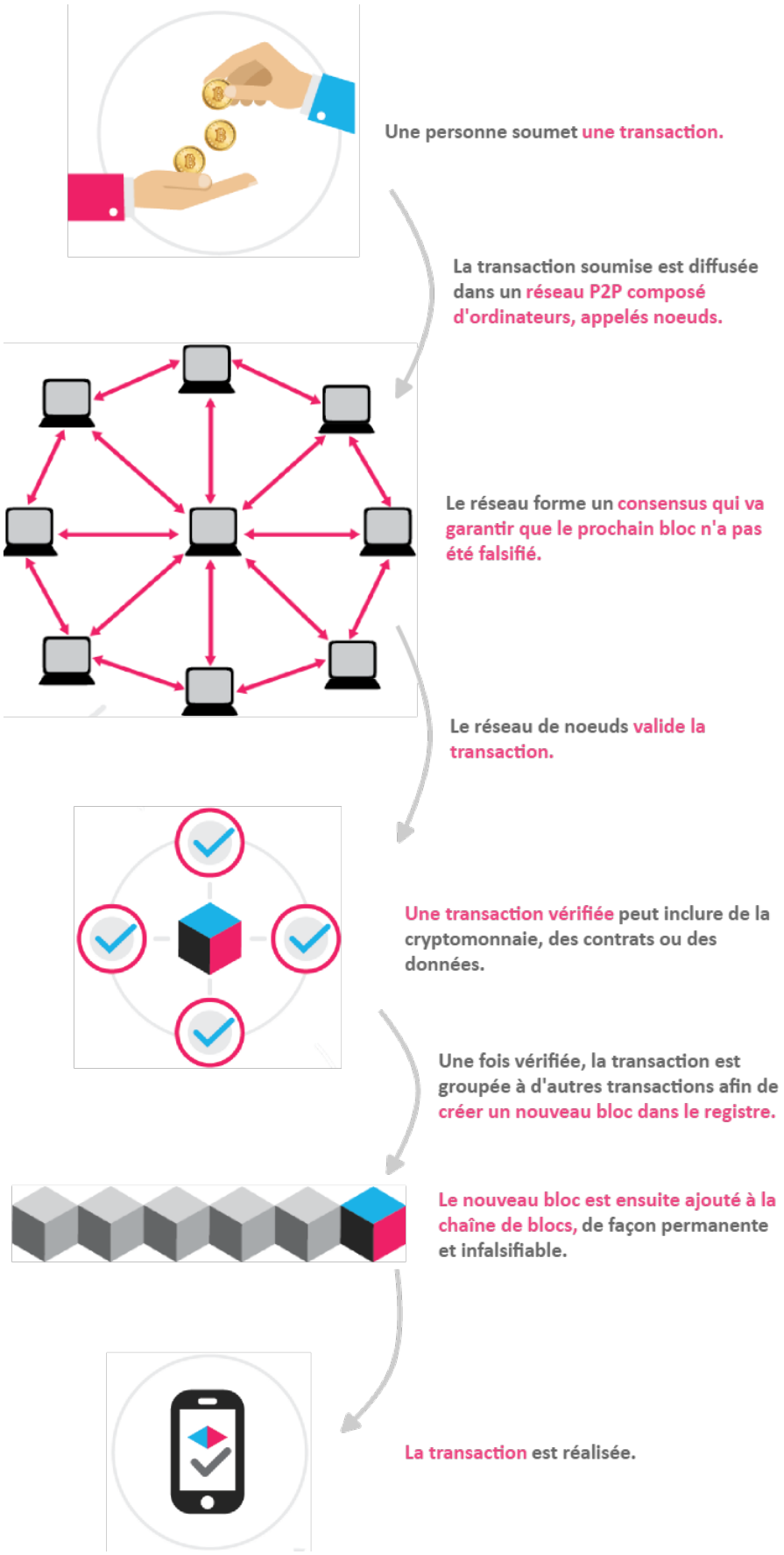

FIGURE 1.1 – Processus de création est de validation d'une transaction sur la blockchain [4]

Les blocs de transaction sont agencés dans un ordre linéaire, c'est-à-dire comme une chaîne, et contiennent une référence au bloc précédent, ainsi qu'un enregistrement des transactions. La preuve de travail est le traitement nécessaire pour générer un nouveau bloc basé sur les nouvelles transactions diffusées sur le réseau. Les blocs de transaction sont créés par un processus appelé le minage [6], qui est conçu pour être coûteux en temps et en énergie ainsi qu'être complexe à réaliser, et s'appuie sur un consensus pour ajuster la difficulté de créer de nouveaux blocs. Le minage est aussi un moyen de sécuriser le réseau en créant, vérifiant, publiant et propageant les blocs dans la blockchain.

1.1.1 Avantages de la blockchain

Si cette technologie connaît un tel succès c'est parce qu'elle apporte de nombreux avantages :

- la décentralisation, qui signifie que son architecture ne repose pas sur une entité centrale et permet d'enregistrer des données dans un réseau distribué ;
- la transparence, puisque l'état des données conservées est consultable publiquement par tout le monde ;
- l'autonomie, puisqu'elle est basée sur un consensus dans lequel chaque partie prenante peut transférer des données de manière sécurisée et autonome ;
- l'immutabilité, en effet toute transaction est persistée définitivement et donc ne peut être effacée ;
- l'anonymat, car toute personne est anonyme dans le sens où elle n'est pas désignée par son identité mais uniquement par une clé publique [7].

1.1.2 Inconvénients de la blockchain

Bien que la blockchain apporte de nombreux avantages, elle comporte aussi des inconvénients :

- la performance, en effet cette technologie sera toujours plus lente qu'une base de données centralisée puisqu'elle nécessite pour chaque transaction une vérification de signature, une validation pour le consensus et la redondance des informations ;
- la consommation énergétique, puisque la validation de blocs reposent sur la résolution d'un puzzle cryptographique nécessitant une grande puissance de calcul ;
- le coût, dans le cas où il est nécessaire d'effectuer un grand nombre de transactions coûteuses ;
- la confidentialité, puisque toute information enregistrée dans la blockchain est publique, il est fortement déconseillé d'y stocker des informations confidentielles ou personnelles, même si elles sont chiffrées.

1.1.3 Introduction aux Smart Contracts

En 1994, Nick Szabo, chercheur juridique et cryptographe, s'est rendu compte que le registre décentralisé pouvait être utilisé pour des smart contracts (contrats intelligents), autrement appelés contrats auto-exécutés, contrats blockchain ou contrats numériques. Les contrats peuvent être convertis en code informatique, stockés et répliqués sur le système et supervisés sur le réseau

6. mining en anglais.

7. Une clé publique est un encodage rendu public dans le cadre d'un échange d'informations utilisant le principe de la cryptographie asymétrique.

qui exécutent la blockchain. Les smart contracts permettent d'échanger de l'argent, des biens, des parts ou n'importe quel actif de manière transparente, sans conflit et en évitant tout service intermédiaire. La Figure 1.2 représente l'établissement d'un smart-contract entre deux parties, sans service intermédiaire.

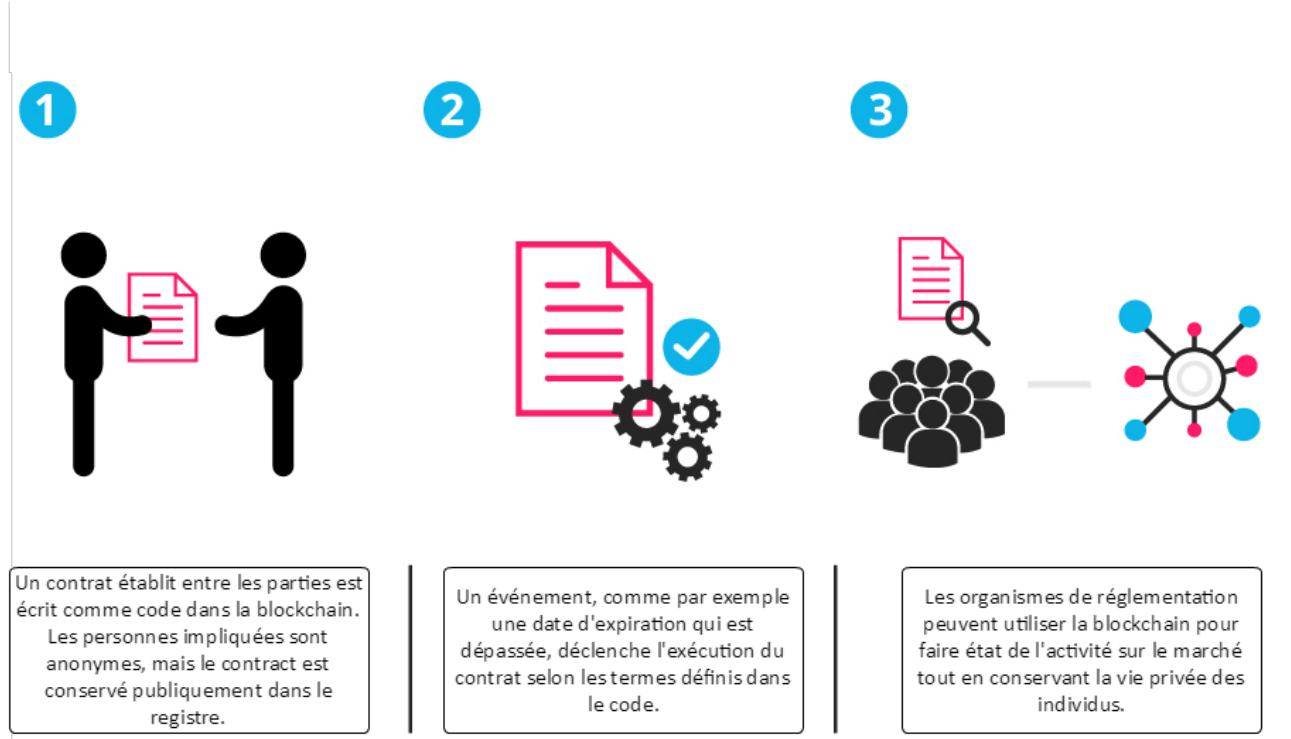

FIGURE 1.2 – Établissement d'un smart-contract [3]

Les smart-contracts apportent de nombreux avantages :

- l'autonomie, qui signifie que l'accord engendré par le contrat n'a pas besoin d'être confirmé par un intermédiaire tierce comme un notaire ou un avocat ; Cela empêche aussi toute manipulation par un tiers sur l'un des parties signataires du contrat puisque l"exécution du contrat est géré automatiquement par le réseau, ce qui empêche tout individu biaisé d'avoir une influence sur le contrat
- la confiance, puisque les documents sont chiffrés dans le registre distribué, il n'y a aucun moyen qu'ils soient perdus ;
- la sauvegarde, toutes les informations sont stockées dans la blockchain et sont répliquées à travers de multiples nœuds du réseau qui les maintiennent ;
- la sécurité, les concepts de cryptographie utilisés vous permettre de conserver vos contrats en sécurité, ils ne peuvent être modifiés ou usurpés ;
- la vitesse, les smart-contracts permettent d'accélérer les procédures, puisque toutes les tâches sont automatisées, réduisant ainsi les heures de travail sur le contrat ;
- des économies, en effet se détacher des intermédiaires permet de réduire considérablement les charges administratives engendrées par la plupart des contrats classiques ;
- la précision, puisque le contrat est automatisé et exécuté par une machine, si l'on admet que le contrat a été codé correctement, alors tout erreur pouvant être introduite lors d'une rédaction manuelle ne peut être présente dans un smart-contract.

1.2 Définition du cadre et des objectifs du stage

Dans ce contexte, un stage ingénieur a été réalisé sur une période de 6 mois au sein de la société Arŋs Spikeseed située au Luxembourg. Le stage a été réalisé dans les locaux de l'entreprise, la langue officielle du projet pour les communications avec les autres membres du consortium (mails, documents, conférence téléphonique) est l'anglais, il en est de même pour la langue utilisée au sein de l'équipe puisque c'est une équipe multinationale. Les documents produits, et présentés dans ce mémoire, ont donc été rédigés en anglais. Ce stage de fin d'études a pour objectif d'intégrer la technologie blockchain au sein d'un processus de gestion de listes de services de confiance dans le cadre d'un règlement européen. La finalité est d'utiliser cette technologie afin de conserver des données publiques relatives à la confiance électronique de manière sécurisée et décentralisée en utilisant une blockchain en tant que registre. Cela a pour but d'assurer la disponibilité et l'intégrité des informations, puisque les données sont distribuées à travers les nœuds du réseau et sécurisées à l'aide de transactions signées et vérifiées par une preuve mathématique.

1.3 Mise en exergue du plan

ÉDIT EN FONCTION DU PLAN DÉFINITIF

Ce mémoire vise à montrer que le champ d'application de la technologie blockchain dépasse son cadre initial et que son utilisation permet de pallier aux problèmes d'architecture et de sécurité des modèles actuels. Dans un premier temps, le contexte du projet sera défini, puis la problématique, qui détaillera les limites des architectures actuelles, sera exposée. Ensuite, sera établit un état de l'art afin de comparer les outils existants et de justifier les choix opérés durant le stage. Après cela, la réalisation du projet sera développée en expliquant : le choix de l'architecture mise en place ; l'avantage de persister des données dans un système de fichiers décentralisé ; l'intérêt de gérer l'authentification des utilisateurs par la mise en place d'un consensus ; l'implémentation d'un système de contrôle de versions et d'un moteur de recherche sur des données stockées dans un réseau décentralisé et distribué. Enfin, les résultats obtenus et les perspectives du projet seront détaillés.

2 Présentation du contexte

2.1 L'entreprise Arηs Spikeseed

Arηs Spikeseed est une entité du groupe Arηs qui est une entreprise de services du numérique (ESN) fondée en 2003 par Jourdan Serderidis. Le groupe est divisé en sociétés réparties au Luxembourg, en Belgique, en Grèce et depuis cette année en Italie. Le groupe Arηs possèdent cinq axes de compétences qui sont présentés dans la Figure 2.1.

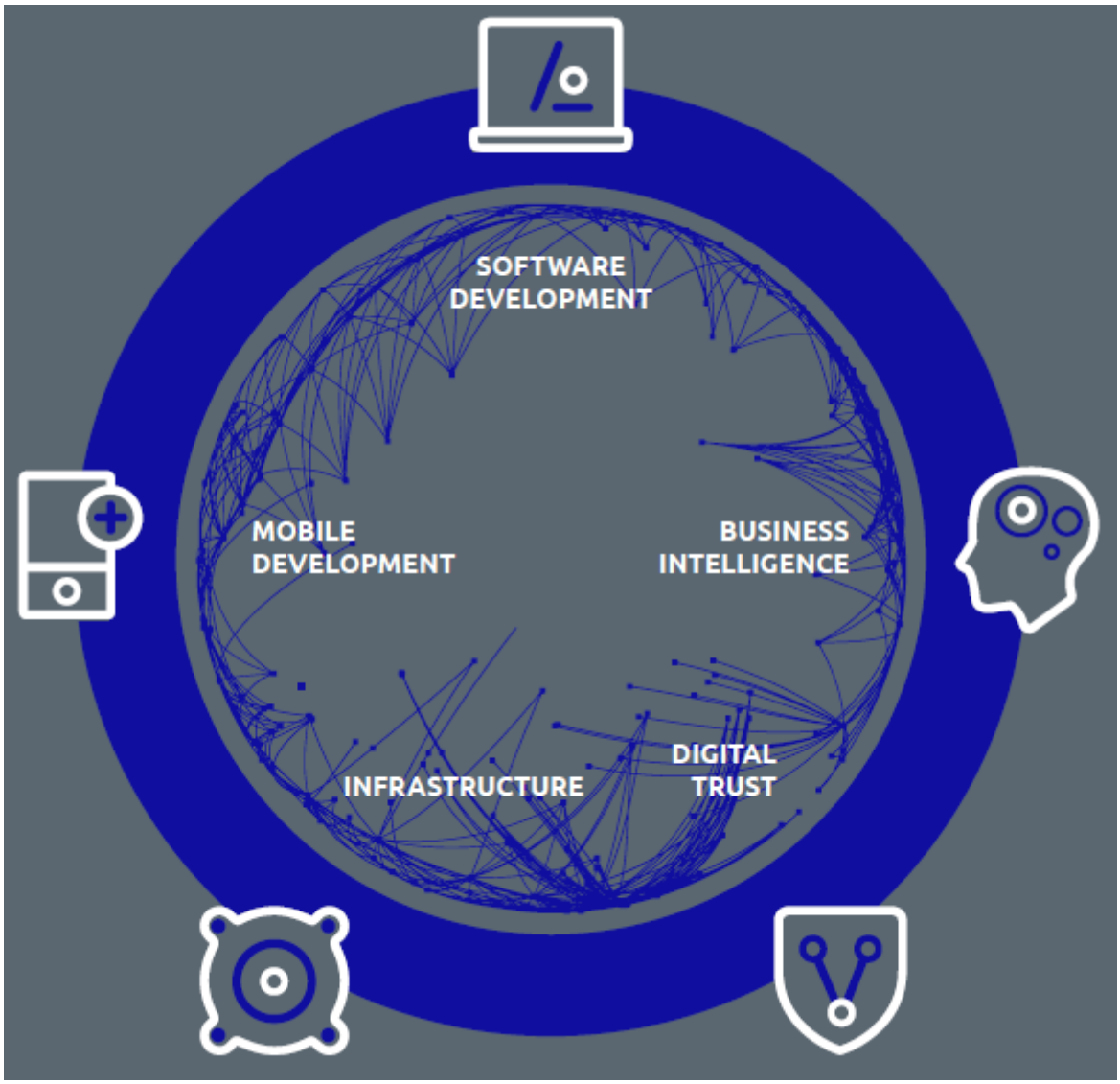

Figure 2.1 – Axes de compétences du groupe Arηs [1]

Comme toutes les autres entités du groupe, Arŋs Spikeseed vise à délivrer des solutions numériques complexes. Elle a la particularité de réaliser principalement des projets de recherche et développement en s'appuyant sur les pratiques agiles et des technologies de pointe. De plus, Arŋs Spikeseed est compétente afin de mettre en œuvre : des solutions liées à la confiance numérique ; des systèmes engageant des masses de données grâce à des technologies innovantes et efficaces comme le Web sémantique ou la Business intelligence ; des applications destinées aux mobiles et aux objets connectés.

2.2 Contexte du projet

Dans le cadre d'un règlement de l'Union Européenne (UE) sur l'identification électronique (eID) et les services de confiance pour les transactions électroniques sécurisées au sein de l'UE (eIDAS), la Commission Européenne (CE) a émis un appel à projet qui a pour visée de supporter la mise en œuvre technique de ce règlement européen. Ce projet de recherche et développement appelé FutureTrust rassemble un consortium de seize partenaires, dont Arŋs Spikeseed, engagé dans la réalisation et la mise en application du règlement européen. Le projet FutureTrust répondra au besoin de solutions globales et interopérables, en fournissant des logiciels libres qui faciliteront l'utilisation de l'identification et de la signature électronique. Il vise à étendre l'infrastructure de la liste européenne de services de confiance existante vers une liste mondiale des services de confiance, nommée Global Trust Service Status List (gTSL), à développer un service de validation ainsi qu'un service d'archivage pour les signatures et les sceaux électroniques, et à fournir des composants pour les certificats qualifiés et pour la création de signatures et de sceaux dans un environnement mobile.

Ce stage de fin d'études a porté sur l'intégration la technologie blockchain dans le cadre du projet FutureTrust et plus particulièrement sur son intégration dans le module de gTSL. Les autres modules du projet ne seront pas détaillés dans ce document.

3 Présentation détaillée de la problématique

Les architectures logicielles évoluent en suivant les innovations technologiques. Aujourd'hui, le domaine de la recherche apporte des nouvelles technologies ou des améliorations aux concepts existants à une vitesse exponentielle, si bien que le temps de réalisation d'un projet le rend obsolète lors de sa livraison. Le meilleur exemple de ce phénomène est le framework Angular, initié par Google, qui est passé de la version 2 à la version 5 en moins d'une année. C'est la réalité actuelle de l'univers technologique poussé par l'innovation, un monde où les acteurs doivent s'adapter en permanence aux changements. La technologie blockchain s'inscrit dans ces innovations récentes issues de la recherche. Elle amène une nouvelle vision d'un Internet décentralisé sans organe central de contrôle, qui va probablement révolutionner la conception des systèmes d'information dans les prochaines années. Dans ce contexte, l'utilisation de la blockchain a été proposée dans le cadre du projet FutureTrust.

3.1 Description du service de liste de confiance globale

Les États membres de l'UE et d'autres pays européens maintiennent généralement des listes d'autorités de certification et d'autres fournisseurs de services de confiance, désignés Trust Service Providers (TSP), dans un ou plusieurs registres à l'échelle nationale. La liste de confiance des États membres de l'UE comprend des informations relatives aux TSPs qualifiés qui sont supervisés par l'État membre compétent, ainsi que des informations relatives aux services de confiance, désignés Trust Services (TS), qu'ils fournissent, conformément aux dispositions prévues par le règlement eIDAS. Les listes de confiance sont des éléments essentiels dans la mise en place de la confiance numérique pour les opérateurs du marché électronique, en permettant aux utilisateurs de déterminer le statut qualifié des TSPs et de leurs TSs. En vertu du règlement eIDAS, les listes nationales de confiance ont un effet constitutif. En d'autres termes, un fournisseur ou un service ne sera qualifié que s'il apparaît dans les listes de confiance. Par conséquent, les utilisateurs (citoyens, entreprises ou administrations publiques) bénéficieront de l'effet juridique associé à un service de confiance qualifié donné uniquement si ce dernier est répertorié (comme qualifié) dans les listes de confiance. Les États membres peuvent inclure dans les listes de confiance des informations sur les fournisseurs de services de confiance non qualifiés et sur d'autres services de confiance définis au niveau national.

La structure d'une liste de confiance est présentée dans la Figure 3.1.

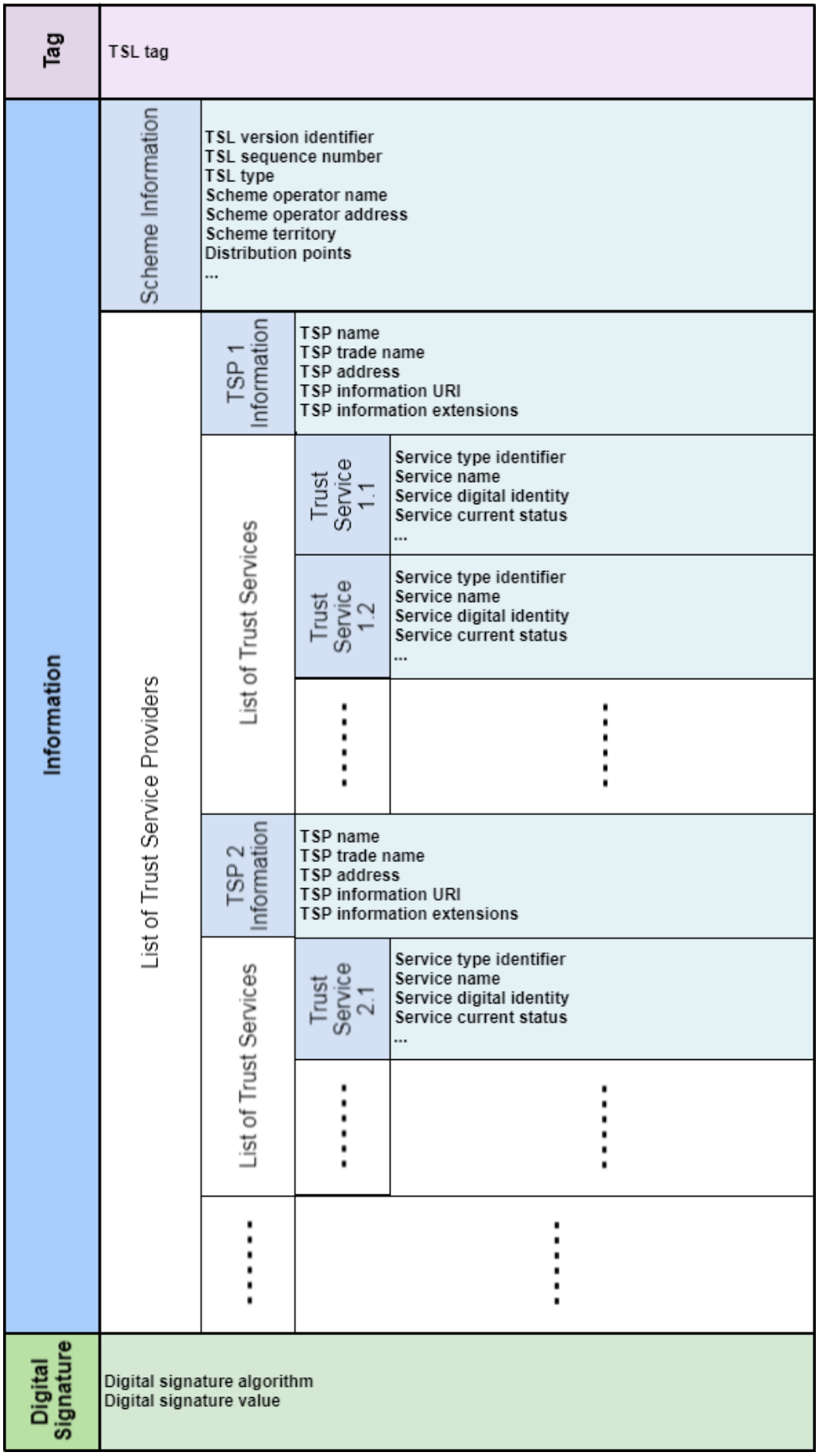

FIGURE 3.1 – gTSL – Structure d'une liste de confiance

Dans la Figure 3.1, on distingue qu'une liste de confiance peut être décomposée en trois sections.

Tag

La section *Tag*, et plus particulièrement son attribut *TSL Tag*, est une URI [1] qui permet d'indiquer le standard respecté par la liste de confiance. Actuellement, le seul standard existant est ETSI TS 119 612 [8]. Il est possible qu'un nouveau standard soit défini dans le futur, cette section permettra donc d'indiquer le standard sur lequel la liste de confiance est basé.

Information

La section *Information* peut-être divisée en deux parties. La première partie, nommée *Scheme Information*, répertorie toutes les informations relatives à la liste de confiance comme par exemple sa version, le nom et l'adresse de l'opérateur de la liste ou encore le pays pour lequel la liste est définie. Il est important de noter que dans la Figure 3.1 la liste des informations n'est pas exhaustive. La seconde partie est la liste des TSPs qui répertorie l'ensemble des fournisseurs approuvés par l'État membre. Pour chacun des TSPs, on retrouve ses informations ainsi que la liste des services de confiances qu'il fournit.

Digital Signature

La section *Digital Signature* permet de vérifier l'authenticité et l'intégrité de la liste de confiance. En effet, chaque liste doit être signée par l'opérateur prévu à cet effet, défini dans la partie *Scheme Information*. Dans cette section doit être indiquée la signature de l'opérateur ainsi que l'algorithme de génération de celle-ci.

3.1.1 Besoins générales

Intérêt du projet

L'intérêt d'un service de gTSL est de favoriser l'établissement de relations de confiance entre les opérateurs du marché en Europe et au-delà. À ce titre, elle étend le schéma actuel de la liste des services de confiance, dont la portée est uniquement européenne. Cette liste a pour but de répertorier les TSPs, ayant un statut qualifié ou non. On entend par statut qualifié que le TSP ait été accrédité par un organisme compétent au sein de l'État membre dans lequel le TSP est déclaré. Le service permet aux utilisateurs finaux de vérifier le statut de ces TSPs et d'accéder à l'ensemble des informations concernant les services de confiance.

Parties prenantes

Les acteurs principaux de la gTSL sont :

1. Une URI (acronyme anglais de Uniform Resource Identifier) est une chaîne de caractères identifiant une ressource.

— les États membres de l'UE, qui doivent établir, maintenir et publier les listes de confiance, incluant les informations relatives aux TSPs de services déclarés au sein de leur État ;
— les fournisseurs de services de confiance, qui sont destinés à s'appuyer sur le service de gTSL dans lequel sont publiés leur statut qualifié et leurs informations publiques ;
— les opérateurs de liste de confiance ne faisant pas partie d'un État membre de l'UE, qui souhaitent intégrer leur liste dans la gTSL ;
— les citoyens de l'UE et non UE, qui sont destinés à utiliser le service afin d'accéder aux statuts et aux informations des différents TSPs répertoriés dans la gTSL.

Objectif du projet

L'objectif principal de la gTSL est de gérer et de fournir les informations relatives aux TSPs qualifiés au sein de l'Union Européenne et au-delà, en étendant le modèle actuel de la liste européenne de services de confiance. De plus, cette réorganisation de l'architecture vise à gérer la gTSL de manière décentralisée dans le but d'en améliorer sa résilience ainsi que sa gestion.

3.1.2 Besoins du système

Objectif du système

En s'appuyant sur la norme de listes de confiance définie dans ETSI TS 119 612 [8], la gTSL vise à résoudre les imperfections actuelles du schéma de liste de confiance, énoncées dans la Section 3.2, lorsqu'il est considéré dans un contexte globalisé. À l'heure actuelle, la Commission européenne publie une liste signée de pointeurs, nommée European List of the Lists (LoTL), dans laquelle chaque pointeur désigne un point de distribution pour une liste nationale de TSPs. Ces listes nationales contiennent des informations sur les TSPs qualifiés et non qualifiés ainsi que sur les services qualifiés ou non qualifiés qu'ils proposent.

Portée du système

La portée de la gTSL concerne la définition de services de confiance qualifiés et de fournisseurs de services de confiance. À ce titre, elle fournira les fonctions nécessaires à la création, à la mise à jour et à la distribution des fournisseurs de services de confiance et des informations concernant leurs services de confiance.

Présentation du système

Afin d'atteindre ses objectifs, le gTSL s'appuiera sur deux principaux composants open source :
— Global Trust Service Lifecycle Manager [2]
— Global Trust Service Responder [3]
De plus, la gTSL s'appuiera sur une interface d'administration afin de présenter les fonctions de gestion des listes de confiance aux utilisateurs. Ces composants et leurs interactions sont illustrés dans la Figure 3.2.

2. en français, Gestionnaire du cycle de vie.
3. en français, Répondeur (dans le sens où il répond aux requêtes des utilisateurs).

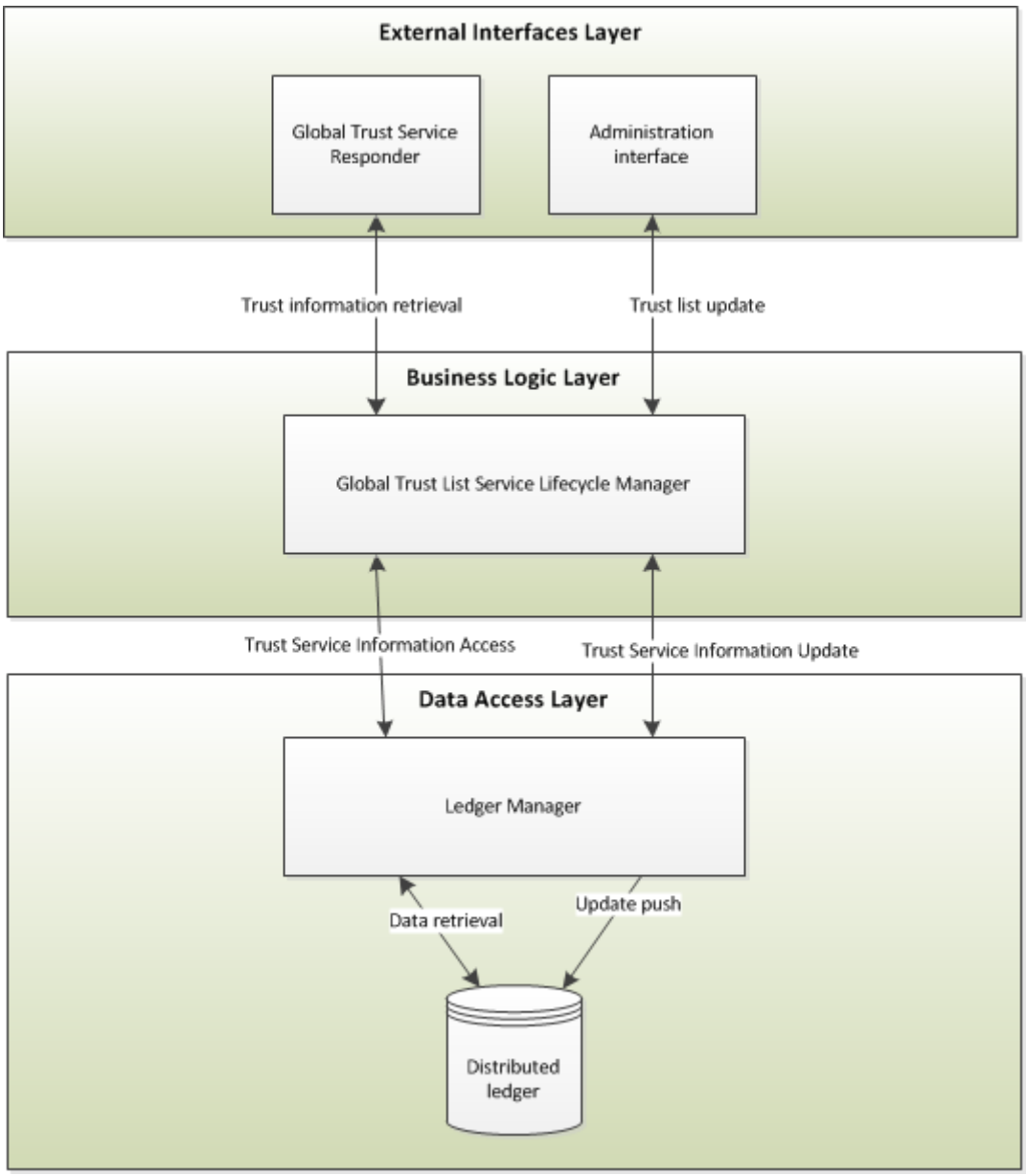

FIGURE 3.2 – gTSL – Architecture 3-Tier [5]

L'objectif du Global Trust Service Responder est de permettre aux applications externes et aux utilisateurs d'interroger la gTSL afin de récupérer les informations relatives aux TSPs, dans le but de vérifier leur statut à un moment donné. Il fournira donc les fonctions nécessaires pour répondre aux demandes d'information sur les statuts de confiance. L'objectif du Global Trust Service Lifecycle Manager est de faciliter la gestion de la hiérarchie des services de confiance, et de permettre la mise à jour du statut des TSPs. Il fournira les fonctions nécessaires à la création, à la mise à jour et à la distribution des informations relatives aux statuts de confiance.

D'un point de vue architectural, le gTSL s'appuiera sur une architecture à 3 couches :

- La couche de services externes exposera les interfaces externes du système, i.e. le Global Trust Service Responder et l'interface d'administration ;
- La couche métier sera composée du Global Trust List Service Lifecycle Manager ;
- La couche de données correspondra aux interfaces et aux composants qui permettent de connecter le gTSL à une solution de stockage de données.

L'un des objectifs de la gTSL est de s'appuyer sur le modèle de distribution centralisé actuel et de l'adapter à un nouveau modèle décentralisé. L'émergence récente du concept de blockchain et

les développements qui l'accompagnent dans les solutions de stockage de données basé sur cette technologie apportent un ensemble de solutions potentielles à cet objectif de décentralisation.

La Section 4 présente les différentes implémentations de blockchain et de système de stockage de données décentralisé pouvant s'interfacer avec une blockchain qui ont été considérées et décrit les interfaces définies pour la couche de données.

Contexte du système

La Figure 3.3 fournit une description de haut niveau des interactions du système avec des entités externes.

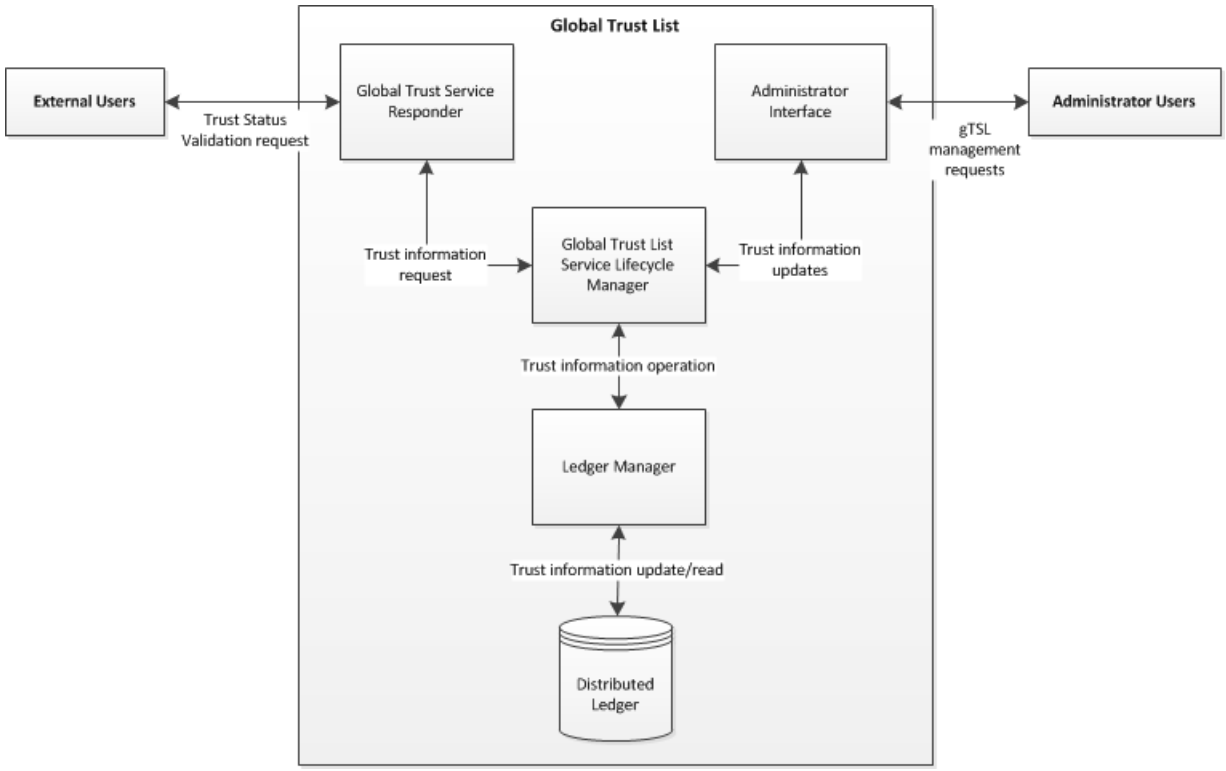

FIGURE 3.3 – gTSL - Schéma du contexte du système [5]

L'entité External Users [4] représente tous les utilisateurs externes qui souhaitent interagir avec le système sans privilège spécifique, dans le but de récupérer des informations relatives à la gTSL. Le Validation Service [5] développé dans le cadre du projet FutureTrust est un des ces utilisateurs externes. L'entité Administrator Users [6] représente tous les utilisateurs externes qui sont autorisés à effectuer des opérations de gestion sur la gTSL, comme par exemple mettre à jour les informations d'un TSP.

Caractéristiques des utilisateurs

Deux types différents d'utilisateurs ont été identifiés concernant la gTSL :

4. en français, utilisateurs externes.
5. en français, service de validation.
6. en français, utilisateurs d'administration.

— Utilisateurs d'administration, i.e. les administrateurs de plates-formes, qui peuvent agir au nom d'un État membre de l'UE et qui sont chargés de la maintenance quotidienne et de la gestion des listes de confiance ;
— Utilisateurs externes, i.e. les personnes et les applications externes qui souhaitent obtenir des informations concernant les statuts de confiance pour un TSP, un TS ou un État membre donné.

Ces utilisateurs auront accès au système à travers des interfaces dédiées :

— Utilisateurs d'administration auront accès à une plateforme de gestion de la gTSL, qui exposera sans ambiguïté les différentes fonctionnalités d'administration auxquelles les utilisateurs doivent avoir accès ;
— Utilisateurs externes auront accès à la fois à une interface graphique et à une seconde interface de web services [7], qui permettra la récupération d'informations concernant les statuts des services de confiance sur base de certificats électroniques fournis par l'utilisateur ainsi que des informations générales concernant les TSPs.

Besoins fonctionnelles

Les besoins fonctionnelles identifiés pour la gTSL sont :

— La gTSL doit permettre la gestion des *Trust Anchors and Meta-data of Identity Providers ;*
— La gTSL doit supporter l'internationalisation (non-UE) du règlement eIDAS. À ce titre, la gTSL doit permettre l'ajout de TSPs déclarés dans un pays qui n'est pas membre de l'UE, qu'ils soient qualifiés ou non.

Besoins d'utilisation

Les besoins d'utilisation identifiés pour la gTSL sont :

— La gTSL doit offrir une interface permettant la récupération ainsi que la publication d'informations relatives aux TSPs. Au minimum, afin d'assurer la conformité avec le standard ETSI TS 119 612 [8], la gTSL doit être disponible via le protocole HTTP.

Besoins de performance

Les besoins de performance identifiés pour la gTSL sont :

— la gTSL doit apporter un stockage interne efficace pour stocker les informations de statut sur les TSPs.
— la gTSL doit être hautement scalable afin de gérer efficacement de nombreuses quantités de demandes parallèles.

Interfaces du système

La gTSL doit exposer, grâce à une interface de web services, les fonctionnalités permettant la récupération d'informations concernant les statuts des services de confiance.

7. Un web service correspond à l'implémentation d'une ressource identifiée par une URL.

Interfaces utilisateurs

Les fonctionnalités de gestion de la gTSL doivent être fournies à travers une interface web cohérente et intuitive, permettant aux utilisateurs de l'utiliser sans ambiguïté. Les interfaces utilisateur doivent rester cohérentes avec les interfaces utilisateur de l'application TL-Manager actuelle tout en ne montrant aucune ambiguïté en termes de hiérarchie visuelle et de contenu.

Fiabilité du système

La gTSL doit être disponible sur une base de 24 heures par jour et 7 jours par semaine. Plus particulièrement, dans le but d'être conforme avec le standard ETSI TS 119 612 [8], le Global Trust Service Responder doit être disponible sur une base de 24 heures par jour et 7 jours par semaine, avec une disponibilité annuelle minimum de 99.9%.

Sécurité du système

En raison de la nature sensible des données gérées par la gTSL, et de la haute disponibilité requise, les besoins en terme de sécurité doivent garantir que ces données ne peuvent être et ne sont pas compromises et que la gestion des services de confiance et des fournisseurs de services de confiance est clairement limitée aux personnes autorisées. La gTSL ne doit pas permettre à des personnes non autorisées de créer, modifier ou supprimer des informations relatives à des services de confiance ou des fournisseurs de services de confiance. La gTSL doit assurer l'intégrité des données qu'elle traite.

3.1.3 Besoins logicielles

Conformité au standard

La gTSL doit être conforme avec le standard ETSI TS 119 612 [8]. À ce titre, la gTSL doit respecter :

- le format et la sémantique d'une liste de confiance ;
- les mécanismes à utiliser pour aider les parties prenantes à localiser, à accéder et à authentifier les listes de confiance.

Rétrocompatibilité

La gTSL doit pouvoir s'intégrer au schéma existant basé sur la LoTL. Cela signifie qu'elle est capable d'importer l'ensemble des listes de confiance actuellement référencées dans la LoTL, mettre en évidence les changements au fur et à mesure qu'ils se produisent grâce à un historique et permettre d'ajouter d'autres TSPs hors UE.

3.2 Limites de l'architecture actuelle

Avec le modèle actuel, les modifications apportées au contenu d'une liste nationale induisent la nécessité de republier toute la liste nationale. De plus, toutes modifications apportées sur l'URL [8] à laquelle la liste est distribuée ou sur le certificat utilisé pour signer la liste, induisent la nécessité de republier à la fois la liste nationale et la liste européenne. Le caractère centralisé du système de distribution des listes de confiance actuel contient des potentiels problèmes qui doivent être résolus dans le cadre de la globalisation des listes de services de confiance :

- les listes de confiance des États membres sont uniquement récupérable en se basant sur la LoTL, le schéma actuel est donc sujet à un point individuel de défaillance [9] ;
- chaque État membre maintient les données relatives à sa liste de confiance, cela signifie que l'arrêt du nœud de distribution d'un État membre rend ses données non consultables ;
- l'architecture existante est exposée à un problème de résilience puisqu'elle nécessite que l'ensemble des nœuds de distribution des États membres soit actifs afin que la liste globale soit considérée complète et donc fiable ;
- l'intégrité des données peut être compromise, en effet si les données d'un État membre sont corrompues localement sur son nœud de distribution, alors l'intégrité globale est compromise puisqu'on se fie uniquement à ce nœud de distribution ;
- des problèmes de performance et de latence peuvent être rencontrés puisqu'il est nécessaire de télécharger et valider l'ensemble des informations qui sont réparties sur différents points de distribution ;
- le schéma actuel ne conserve pas l'historique des modifications, c'est-à-dire qu'une nouvelle publication d'une liste remplace totalement la précédente, ce qui ne permet pas de conserver une trace des modifications mises en œuvre entre les versions ;

L'objectif de la gTSL est d'effectuer une refonte de l'architecture actuelle qui a montré ses limites en y apportant des technologies innovantes. Pour cela, il est nécessaire d'adopter un modèle décentralisé et distribué qui permettra de résoudre les problèmes de résilience et de point individuel de défaillance. La technologie blockchain apporte en plus des avantages qui permettront de s'assurer de l'intégrité des données ainsi que de la sécurité du système.

3.3 Gestion de projet

3.3.1 Méthode de gestion de projet

La méthode Agile a été utilisée au cours de ce projet. Plus particulièrement, l'équipe s'est appuyée sur le schéma Scrum, qui permet un cadre de travail itératif où les tâches majeures sont décomposées en sous-tâches. La méthodologie Scrum est basée sur le découpage d'un projet en sprints [10], qui sont des cycles de livraison très courts. Un sprint peut s'étendre sur une durée de quelques heures à un mois. Dans notre cas, nous avons choisi une durée de deux semaines par sprint. En début de sprint, une estimation de la durée de chaque tâche est effectuée, ensuite une planification opérationnelle est réalisée. Un sprint se termine généralement par une démonstration du travail réalisé suivie d'une rétrospective, afin d'analyser le déroulement du sprint achevé et dans le but d'améliorer les pratiques de l'équipe. Quotidiennement est organisé un scrum mee-

8. Une URL (acronyme anglais de Uniform Resource Locator) est couramment appelé adresse web.

9. Un point individuel de défaillance (single point of failure ou SPOF en anglais) est un point d'un système informatique dont le reste du système est dépendant et dont une panne entraîne l'arrêt complet du système.

10. Un sprint est une période sur laquelle sont réalisées des tâches définies.

ting [11], réunion courte et énergique, qui permet à l'équipe de discuter de l'avancée du sprint et de lever les points bloquants du projet. Afin de suivre la progression des objectifs au fur et à mesure de l'avancement du projet, nous avons utilisé l'outil JIRA. Il a servi notamment à définir les différentes tâches du projet et à répartir le travail entre les membres de l'équipe.

Les avantages majeures de Scrum

Scrum apporte des avantages qui sont définies comme étant les trois piliers de la méthodologie :

- la transparence, par l'utilisation d'un langage commun afin de permettre à tout un chacun d'obtenir rapidement une bonne compréhension du projet et par la garantie que tous les indicateurs relatifs à l'état du développement soient visibles ;
- l'inspection, par l'analyse quotidienne du travail accompli et restant lors des sprints, afin de repérer tout indicateur indésirable ;
- l'adaptation, dans le cas d'une dérive après inspection, des ajustements doivent être effectués afin de minimiser les écarts de réalisation.

3.3.2 Organisation du temps

La Figure 3.4 est un diagramme de Gantt qui expose de manière globale la répartition du travail réalisé. Ce diagramme est volontairement non détaillé puisque l'utilisation de la méthodologie Agile ne permet pas d'organiser à l'avance et avec précision la réalisation des tâches. Il est important de noter que le diagramme se termine à la date de fin du stage mais que la livraison du projet est prévue au 30 novembre 2017.

11. Un scrum meeting est communément appelé mêlée en français.

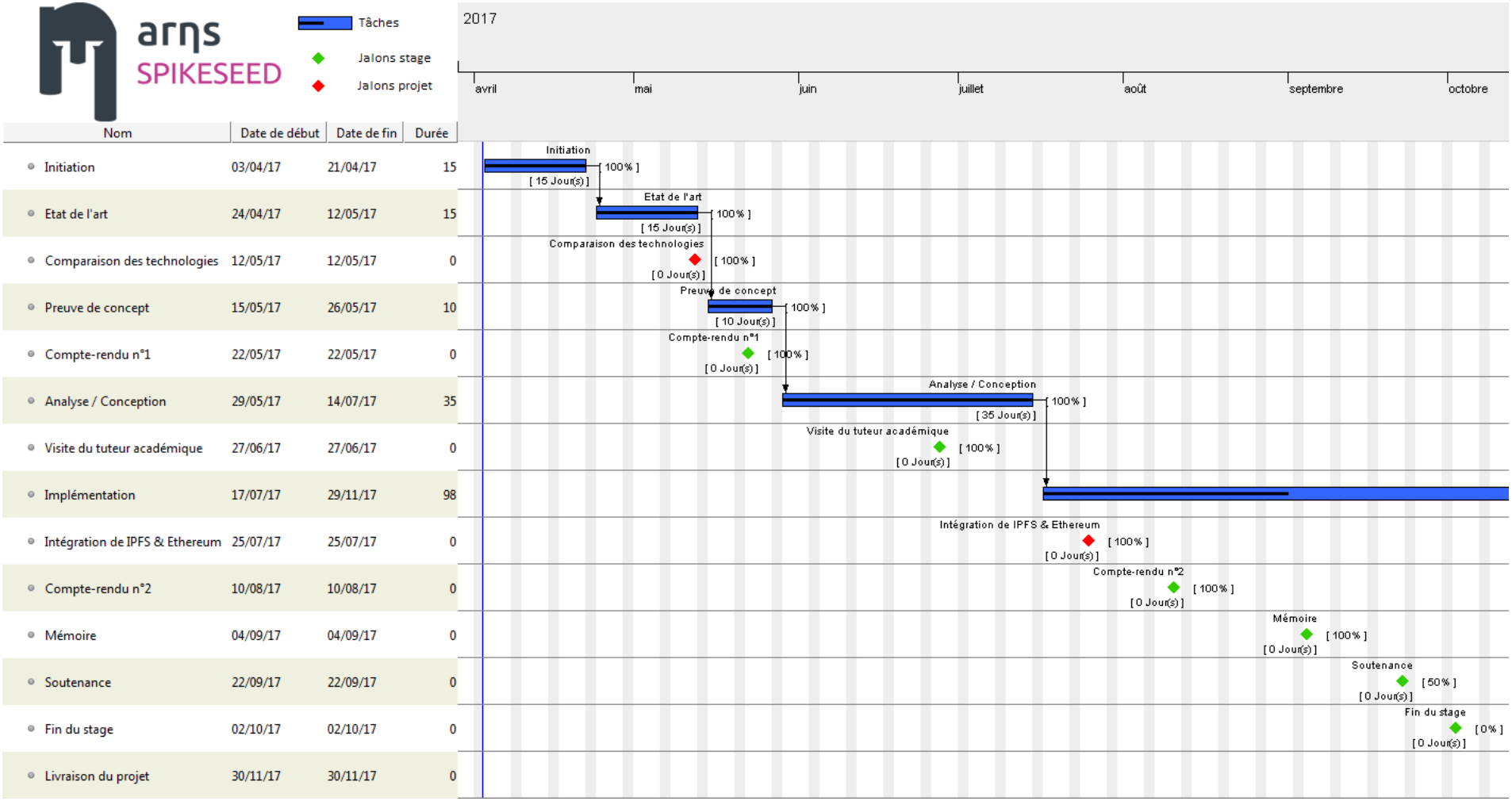

FIGURE 3.4 – Diagramme de Gantt

Le diagramme de Gantt en Figure 3.4 permet d'avoir un aperçu de l'organisation du stage. Le stage s'est déroulé sur une période de six mois, du 3 avril 2017 au 30 septembre 2017. Durant cette période, j'ai été amené à réaliser différentes tâches qui seront détaillées dans la suite de ce mémoire. D'un point de vue générale, le stage a été décomposée en cinq parties majeures qui sont décrites ci-après.

Initiation

La phase d'initiation a commencé dès le début du stage et a duré environ trois semaines. Elle peut être décomposée en deux sous-parties. Dans un premier temps, j'ai dû me familiariser avec le projet. Pour réaliser cela, j'ai eu accès au document de conception de la gTSL [5] qui explique, d'une manière générale et sans précision de technologies, l'architecture à mettre en place ainsi que les cas d'utilisation à implémenter dans le cadre du projet FutureTrust. J'ai pris connaissance du standard ETSI TS 119 612 [8] qui détaille le format à respecter dans le cadre des listes de confiance. Dans un second temps, j'ai effectué des recherches sur les notions de cryptographie, de signature électronique et de blockchain afin de les mettre en œuvre dans le projet. Cette seconde partie a été la préparation du travail suivant qui est l'état de l'art.

État de l'art

La seconde phase, ayant pour objectif d'établir un état de l'art, a suivi la phase d'initiation et s'est étendue sur trois semaines. L'état de l'art permet de connaître l'état des connaissances et technologies actuelles dans un domaine spécifique. Pour notre projet, l'état de l'art a porté sur la blockchain ainsi que les systèmes de stockage de données décentralisés. Le but de cette phase a été d'explorer le plus largement possible les technologies pouvant être utilisées dans le projet afin de les comparer et de choisir les solutions répondant à nos besoins. Cette étape a donné lieu à un livrable qui a été inclus dans le document de conception de la gTSL comme le montre le jalon "Comparaison des technologies" dans la Figure 3.4. L'état de l'art est détaillé dans la Section 4.

Preuve de concept

À la suite de l'état de l'art, un choix de technologies a été effectué. L'étape suivante a donc été de prouver que les choix opérés correspondent aux besoins du projet. Pour cela, j'ai pu réaliser une preuve de concept sur une durée de quinze jours. Cette phase a ensuite donné lieu à une démonstration à l'équipe afin de valider de manière définitive les choix technologiques. Le détail de la preuve de concept ainsi que la justification des choix opérés sont présentés dans la Section 6.

Analyse / Conception

La phase d'analyse et conception a été précédée de la preuve de concept qui a permis de valider les technologies à utiliser pour le projet. Dans le cadre de cette phase, nous avons déterminé les modules à implémenter afin de répondre aux besoins exprimés dans le document de conception. L'analyse et la conception ont consisté à réfléchir sur la mise en place d'une architecture décentralisée basée sur une blockchain en étant conforme aux différentes contraintes définies dans le document de conception et en s'appuyant sur les technologies choisies. Cette partie a duré un mois et demi, et est détaillée dans la Section 5.

Implémentation

La dernière phase, et la plus conséquente puisqu'elle s'étend sur de la mi-juillet à fin novembre, est l'implémentation. Elle consiste à développer la solution conçue et imaginée lors de la phase d'analyse et conception. Lors de la rédaction de ce mémoire, cette phase est en cours de réalisation. Cette phase est découpée en sprints d'une durée chacun de deux semaines. L'implémentation est détaillée dans la Section 6.

4 État de l'art

Dans le cadre du projet FutureTrust, il a été nécessaire d'effectuer un état de l'art afin d'avoir une vue globale de l'état actuelle de la technologie blockchain et des technologies distribuées et décentralisées comme les bases de données ou les systèmes de fichiers. Pour cela, il convient de décrire l'émergence de la blockchain ces dix dernières années, d'expliquer les avantages qu'apporte la décentralisation dans les systèmes actuels et d'établir une liste des technologies existantes. L'état de l'art a pour intérêt de comprendre l'effervescence autour de ces technologies ainsi que d'opérer les meilleurs choix dans la réalisation du module de gTSL. Les solutions existantes présentées en Section 4.3 ont été envisagées afin de mettre en place un système permettant de conserver, gérer et récupérer les données de la gTSL de manière sécurisée et apportant une forte résilience.

4.1 L'émergence de la blockchain

4.1.1 Historique

Le concept de monnaie numérique décentralisée, ainsi que d'autres applications comme les registres de propriété, existe depuis des décennies. Les protocoles de systèmes de paiement apparus dans les années 1980 et 1990, connus sous le nom de e-cash[6], reposant sur des concepts cryptographiques, avaient pour but de fournir une monnaie avec un degré élevé de confidentialité. La mise en place de ces protocoles a largement échoué à cause de leur dépendance à un intermédiaire centralisé. En 1998, la technologie B-money[7] créé par Wei Dai voit le jour, et devient la première proposition introduisant l'idée de créer de l'argent en résolvant des puzzles informatiques en se basant sur un consensus décentralisé, mais cette proposition a été peu détaillée sur la manière dont le consensus décentralisé pouvait effectivement être mis en œuvre. En 2005, Hal Finney a introduit le concept de preuves de travail réutilisables, un système qui utilise des idées de b-money avec les puzzles Hashcash[2] difficiles à calculer créés par Adam Back afin de créer un concept de crypto-monnaie, mais encore une fois cette proposition n'a pas su être exploité. En 2009, une monnaie décentralisée a été pour la première fois mise en œuvre par Satoshi Nakamoto, combinant les différentes propositions établies pour la gestion de la propriété par l'utilisation de clés publiques avec un algorithme de consensus permettant de suivre l'identité du possesseur de la monnaie, appelé preuve de travail (Proof of Work).

Le mécanisme derrière la preuve de travail a été une avancée car il a simultanément résolu deux problèmes. Tout d'abord, il a fourni un algorithme de consensus simple et modérément efficace, permettant aux nœuds du réseau de convenir collectivement d'un ensemble de mises à jour du registre Bitcoin. Deuxièmement, il a fourni un mécanisme qui permet l'entrée libre d'un nouveau membre dans le processus de consensus, résout le problème engendré par l'influence d'un

membre dans le consensus, tout en empêchant les attaques Sybil [1]. Dans la preuve de travail, le poids d'un nœud dans le processus de vote par consensus est directement proportionnel aux ressources de calcul que ce nœud apporte. Depuis cela, une approche alternative, appelée preuve d'enjeu (Proof of Stake), initiée par Peercoin[9] a été proposée, dans laquelle le poids d'un nœud est proportionnel à la quantité de crypto-monnaie qu'il détient et non plus aux ressources informatiques qu'il apporte.

4.1.2 État actuel et Potentiel futur

Actuellement deux principaux protocoles publiques se sont démarqués, le premier est bien entendu Bitcoin, le second est Ethereum récemment créé. Malgré cela, les défis et les développements potentiels de la blockchain restent nombreux.

Firstly, this is very new technology ; it has the vigor of youth but also suffers from teething problems. Although the technical foundation has stabilized as concerns Bitcoin (which has existed since 2009) and its limited uses (mainly cryptocurrency bitcoin and proof of existence), it is not the same for the other blockchains. The scope of the work to be carried out to enable the technology in general to scale and include all the types of work currently imagined is staggering : development of tools for the general public to interact with the chain, improvement of consensus protocols (Proof of Work, Proof of Stake…), increase in the number of transactions processed per second (through improvement to the network such as Segregated Witness on Bitcoin, sharding on Ethereum ; by the sidechains or state channels), creation and improvement of development tools, research on good practices for the security of smart contracts, implementation of de-identification protocols for transactions, etc. The list is long, and the task is enormous. But those involved seem eager to take the bull by the horns and research is progressing at a fast pace, giving rise to occasional existential crises (in particular concerning Ethereum), which eventually resolve themselves (cf. antifragile, etc.).

As a natural consequence of its youth, the technology is also misunderstood. It is possible that you have read 50 articles on the subject without having really understood what it's all about : this is normal. Blockchain gurus are everywhere, but true experts in the sector are still rare ; those who have clearly understood what this technology is and means. Watch out for consultants who bill exorbitant consulting services to desperate companies who have heard of this phenomenon and want to develop a "blockchain project" at all costs, even where this makes no sense at all in the context of their business activities. Learn to recognize them from a combination of all or some of the following clues : they insist on separating the "blockchain" from crypto-currencies and the Bitcoin in particular, which they equate with speculation "like Dutch tulips", reject any use of a public blockchain, converse in buzzwords (in combination with IoT , big data , deep learning , uberisation of uber…) and focus heavily on 1. the supposed complexity of the system which frees them from going into detail, and the 2. the threat this incredible invention poses to your business activities…

The newness of the technology also explains that of the ecosystem. It wasn't until 2013/2014 that the first serious crypto-currency exchange platforms and service companies came into being. Currently, the sector is buzzing, with numerous start-ups and also initiatives by major companies. There is still one entirely missing element : identity management. On the public blockchain, everyone is anonymous, or rather pseudonymous, identified by an account number. This is fine for certain types of use, but other uses require an identity that can be certified and associated

1. Une attaque Sybil est une attaque informatique qui vise à renverser un système par la création de fausses identités dans un réseau pair-à-pair.

with a particular account, and that this account can by recovered by its legitimate owner in the event of loss. This is a major complex issue arising from the design of the technology ; however, numerous initiatives are on-going in this area (e.g. AEternam, a French initiative, and uPort from ConsenSys). Since the advent of Ethereum, this ecosystem also includes emerging entities on the blockchain, i.e. the dApps, or decentralized applications. In 2016 we discovered their potential but also (painfully) experienced their weaknesses during the hack of the DAO. The security of smart contracts is still at an early stage, and considerable efforts will be necessary before envisaging the automatization of very high value added operations. Numerous initiatives have already been initiated in this respect and will continue in 2017 : fundamental research, formal analysis software, improvement of the language and even the Ethereum Virtual Machine, etc. It should also be noted that the current issues are not discouraging those working in the sector : numerous dApps projects were funded this year, in particular through ICO (Initial Coin Offerings), and should be launched in 2017 or 2018. We will also be watching planned projects : Augur/Gnosis, Digix, Etherisc, uPort, Golem/iEx.ec, CharityDAO, SingularDTV, Ledgys, etc.

Numerous types of blockchain use are thus under development. And, where there is use there must be rules of use, a legal framework… and currently, the law is lacking. We are not speaking here of the law in the sense of a "blockchain law" which would restrict or control the use of this technology (although this will happen soon enough), but rather a form of legal recognition of the new categories created by this technology. In particular, blockchain assets in their most obvious form, cryptocurrency, but also in their derived forms, proof of existence, programmable tokens, votes, etc. Acknowledgement of the unique characteristics of these inscriptions, by local, European law and international treaties, under both general law and by the various regulatory bodies by sector (finance, healthcare, energy), would be a big step forward, which would allow businesses to develop the disruptive types of use which generate wealth. We are still a long way off, as the lawmakers and regulatory bodies are taking a wait and see approach, giving rise to a state of complete legal uncertainty ! This lack of official recognition also impacts private sector involvement ; it is still difficult for those involved in the blockchain sector to open bank accounts, find accountants, etc. If Europe wants to be at the forefront of the "Blockchain revolution", it must do more, both at the national and international levels. In the meantime, the applicable law is explored directly by the law professionals ; notably lawyers that are focused on the issues raised by their clients (and notably Fieldfisher LLP firm, of which I am a member).

Finally, it is impossible to address the "blockchain" without mentioning governance, which designates here the mechanisms which govern the development of public blockchains (in particular technical protocol developments). As we have seen, the governance mechanisms are complex, as the blockchain is a system with diverse participants whose interests do not always converge. This lack of clarity is a legitimate cause for concern. However, an essential part of the technology, its decentralization, is at stake here. A blockchain with a central point of control having sole power to make decisions would find itself with a single point of failure, which is exactly what the technology seeks to avoid… This does not mean that all initiatives aimed at improving the development process for public blockchains are doomed to failure. Projects such as Tezos are based, in particular, on the idea of new governance, as decentralized as the blockchain itself… Is this feasible ? Time will tell.

In the future, we will be watching for : major technical developments such as the formal verification of smart contracts, proof of stake, sharding, etc. recognition by the public and private sector of cryptocurrencies and more generally of blockchain inscriptions. projects in development (projects involving blockchain, dApps, off-chain and on-chain services) which will be launched in the short and medium term.

4.2 La décentralisation du web

The original purpose of the web and internet, if you recall, was to build a common neutral network which everyone can participate in equally for the betterment of humanity. Fortunately, there is an emerging movement to bring the web back to this vision and it even involves some of the key figures from the birth of the web. It's called the Decentralised Web or Web 3.0, and it describes an emerging trend to build services on the internet which do not depend on any single "central" organisation to function.

So what happened to the initial dream of the web ? Much of the altruism faded during the first dot-com bubble, as people realised that an easy way to create value on top of this neutral fabric was to build centralised services which gather, trap and monetise information.

Search Engines (e.g. Google), Social Networks (e.g. Facebook), Chat Apps (e.g. WhatsApp) have grown huge by providing centralised services on the internet. For example, Facebook's future vision of the internet is to provide access only to the subset of centralised services it endorses (Internet.org and Free Basics).

Meanwhile, it disables fundamental internet freedoms such as the ability to link to content via a URL (forcing you to share content only within Facebook) or the ability for search engines to index its contents (other than the Facebook search function).

paltalk-tinychat

The Decentralised Web envisions a future world where services such as communication, currency, publishing, social networking, search, archiving etc are provided not by centralised services owned by single organisations, but by technologies which are powered by the people : their own community. Their users.

The core idea of decentralisation is that the operation of a service is not blindly trusted to any single omnipotent company. Instead, responsibility for the service is shared : perhaps by running across multiple federated servers, or perhaps running across client side apps in an entirely "distributed" peer-to-peer model.

Even though the community may be "byzantine" and not have any reason to trust or depend on each other, the rules that describe the decentralised service's behaviour are designed to force participants to act fairly in order to participate at all, relying heavily on cryptographic techniques such as Merkle trees and digital signatures to allow participants to hold each other accountable.

There are three fundamental areas that the Decentralised Web necessarily champions :privacy, data portability and security.

Privacy : Decentralisation forces an increased focus on data privacy. Data is distributed across the network and end-to-end encryption technologies are critical for ensuring that only authorized users can read and write. Access to the data itself is entirely controlled algorithmically by the network as opposed to more centralized networks where typically the owner of that network has full access to data, facilitating customer profiling and ad targeting. Data Portability : In a decentralized environment, users own their data and choose with whom they share this data. Moreover they retain control of it when they leave a given service provider (assuming the service even has the concept of service providers). This is important. If I want to move from General Motors to BMW today, why should I not be able to take my driving records with me ? The same applies to chat platform history or health records. Security : Finally, we live in a world of increased

security threats. In a centralized environment, the bigger the silo, the bigger the honeypot is to attract bad actors. Decentralized environments are safer by their general nature against being hacked, infiltrated, acquired, bankrupted or otherwise compromised as they have been built to exist under public scrutiny from the outset.

Just as the internet itself triggered a grand re-levelling, taking many disparate unconnected local area networks and providing a new neutral common ground that linked them all, now we see the same pattern happening again as technology emerges to provide a new neutral common ground for higher level services. And much like Web 2.0, the first wave of this Web 3.0 invasion has walked among us for several years already.

Git is wildly successful as an entirely decentralised version control system – almost entirely replacing centralised systems such as Subversion. Bitcoin famously demonstrates how a currency can exist without any central authority, contrasting with a centralised incumbent such as Paypal. Diaspora aims to provide a decentralised alternative to Facebook. Freenet paved the way for decentralised websites, email and file sharing.

Less famously, StatusNet (now called GNU Social) provides a decentralised alternative to Twitter. XMPP was built to provide a decentralised alternative to the messaging silos of AOL Instant Messenger, ICQ, MSN, and others.

Telephone switchboard operators circa 1914. Photo courtesy Flickr and reynermedia. Telephone switchboard operators circa 1914. Photo courtesy Flickr and reynermedia.

However, these technologies have always sat on the fringe — favourites for the geeks who dreamt them up and are willing to forgive their mass market shortcomings, but frustratingly far from being mainstream. The tide is turning . The public zeitgeist is finally catching up with the realisation that being entirely dependent on massive siloed community platforms is not entirely in the users' best interests.

Critically, there is a new generation of Decentralised Startups that have got the attention of the mainstream industry, heralding in the new age for real.

Blockstack and Ethereum show how Blockchain can be so much more than just a cryptocurrency, acting as a general purpose set of building blocks for building decentralised systems that need strong consensus. IPFS and the Dat Project provide entirely decentralised data fabrics, where ownership and responsibility for data is shared by all those accessing it rather than ever being hosted in a single location.

The real step change in the current momentum came in June at the Decentralised Web Summit organised by the Internet Archive. The event brought together many of the original "fathers of the internet and World Wide Web" to discuss ways to "Lock the web open" and reinvent a web "that is more reliable, private, and fun."

Brewster Kahle, the founder of the Internet Archive, saw first hand the acceleration in decentralisation technologies whilst considering how to migrate the centralised Internet Archive to instead be decentralised : operated and hosted by the community who uses it rather being a fragile and vulnerable single service.

Additionally, the enthusiastic presence of Tim Berners-Lee, Vint Cerf, Brewster himself and many others of the old school of the internet at the summit showed that for the first time the shift to decentralisation had caught the attention and indeed endorsement of the establishment.

Tim Berners-Lee said :

The web was designed to be decentralised so that everybody could participate by having their own domain and having their own webserver and this hasn't worked out. Instead, we've got the situation where individual personal data has been locked up in these silos. [...] The proposal is, then, to bring back the idea of a decentralised web.

To bring back power to people. We are thinking we are going to make a social revolution by just tweaking : we're going to use web technology, but we're going to use it in such a way that we separate the apps that you use from the data that you use.

We now see the challenge is to mature these new technologies and bring them fully to the mass market. Commercially there is huge value to be had in decentralisation : whilst the current silos may be washed away, new ones will always appear on top of the new common ground, just as happened with the original Web.

Github is the posterchild for this : a $2 billion company built entirely as a value-added service on top of the decentralised technology of Git — despite users being able to trivially take their data and leave at any point.

Similarly, we expect to see the new wave of companies providing decentralised infrastructure and commercially viable services on top, as new opportunities emerge in this brave new world.

Ultimately, it's hard to predict what final direction Web 3.0 will take us, and that's precisely the point. By unlocking the web from the hands of a few players this will inevitably enable a surge in innovation and let services flourish which prioritise the user's interests.

Apple, Google, Microsoft, and others have their own interests at heart (as they should), but that means that the user can often be viewed purely as a source of revenue, quite literally at the users' expense.

As the Decentralised Web attracts the interest and passion of the mainstream developer community, there is no telling what new economies will emerge and what kinds of new technologies and services they will invent. The one certainty is they will intrinsically support their communities and user bases just as much as the interests of their creators.

——

But wait : Isn't the internet already a decentralized network that no one owns ? In theory, yes. In practice, a small number of enormous companies control or at least mediate much of the internet. Sure, anyone can publish whatever they want to the web. But without Facebook and Google, will anyone be able to find it ? Amazon, meanwhile, controls not just the web's biggest online store but a cloud computing service so large and important that when part of it went offline briefly earlier this year, the internet itself seemed to go down. Similarly, when hackers attacked the lesser-known company Dyn—now owned by tech giant Oracle—last year, large swaths of the internet came crashing down with it. Meanwhile, a handful of telecommunications giants, including Comcast, Charter, and Verizon, control the market for internet access and have the technical capability to block you from accessing particular sites or apps. In some countries, a single state-owned telco controls internet access completely.

Given those very non-utopian realities, people in the real world are also hard at work trying to rebuild the internet in a way that comes closer to the decentralized ideal. They're still pretty far from Richard's utopian vision, but it's already possible to do some of what he describes. Still, it's not enough to just cut out today's internet power players. You also need to build a new internet that people will actually want to use.

Storage Everywhere

On the show, Richard's plan stems from the realization that just about everyone carries around a smartphone with hundreds of times more computing power than the machines that sent humans to the moon. What's more, those phones are just sitting in people's pockets doing nothing for most of the day. Richard proposes to use his fictional compression technology—his big innovation from season one—to free up extra space on people's phones. In exchange for using the app, users would agree to share some of the space they free up with Pied Piper, who will then resell it to companies for far less than they currently pay giants like Amazon.

The closest thing to what's described on Silicon Valley might be Storj, a decentralized cloud storage company. Much like Pied Piper, Storj has built a network of people who sell their unused storage capacity. If you want to buy space on the Storj network, you upload your files and the company splits them up into smaller pieces, encrypts them so that no one but you can read your data, and then distributes those pieces across its network.

"You control your own encryption keys so we have no access to the data," says cofounder John Quinn. "We have no knowledge of what is being stored."

Also like Pied Piper, Storj bills itself as safer than traditional storage systems, because your files will reside on multiple computers throughout the world. Quinn says that in order to lose a file, 21 out of 40 of the computers hosting it would have to go offline.

Storj proves that the Silicon Valley's basic idea is feasible. But unlike Pied Piper, Storj doesn't rely on smartphones. "Phones don't have much storage and the network capability isn't great, so the show's idea is a little fanciful," says Quinn. Someday, 5G wireless networks might make phones a more viable part of the Storj network. If Richard's compression algorithm was real, those smaller files will help too. But for now, the Storj network relies primarily on servers, laptops, and desktop computers. The reality is less grand than the HBO fantasy.

IPFS

As interesting as Storj is, it's not quite what Richard actually described in his pitch. Storj is a storage service, not a whole new internet. A more ambitious project called IPFS (short for "Interplanetary File System") is probably a bit closer to Richard's grand vision of a censorship-resistant internet with privacy features built right in.

The idea behind IPFS is to have web browsers store copies of the pages they visit and then do double-duty as web servers. That way, if the original server disappears, the people who visited the page can still share it with the world. Publishers get improved resilience, and readers get to help support the content they care about. With encryption a part of the protocol, criminals and spies can't in theory see what you're looking at. Eventually, the IPFS team and a gaggle of other groups hope to make it possible to build interactive apps along the lines of Facebook that don't require any centralized servers to run.

You need to build a new internet that people will actually want to use.

But the idea of building a censorship-proof internet by backing up copies throughout the internet isn't without its potential problems. Sometimes publishers want to remove old content. IPFS creator Juan Benet told us last year that the project is trying to work out ways to let publishers "recall" pages that are being shared. But that idea is also fraught. What's to stop a government censor from using the recall feature? What happens if someone creates a version that ignores recalls? Then there are moral and legal risks. Tools like Storj and the venerable peer-to-peer

sharing system Freenet make it impossible to know just what content you're storing for other people, which means you could be playing host to, say, child pornography. Quinn says that the Storj team is currently working on ways to block known problem users. But it won't be able to completely guarantee that none of its hosts will end up storing illegal content.

IPFS gets around this largely by letting people decide which of the content they've visited they actually want to share. But this means that less popular content, even if it's perfectly legal and ethical, might end up disappearing if too few people share it. Benet and company are working on a system called Filecoin that, not unlike Storj, would compensate people for providing access. Even overcoming these trade-offs inherent in decentralization, people may still not want to use these apps. Storj may be able to win over businesses by being cheaper, but even if it is more reliable, the idea of storing data on random machines scattered across the internet instead of in a traditional data center sounds risky compared to, say, the massively robust AWS, backed by Amazon's technical know-how and billions of dollars. Convincing people to use decentralized alternatives to Facebook and Twitter has proven to be a notoriously difficult problem. Getting people to use what amounts to a whole new version of the web could be even harder.

Mesh

Even if IPFS, Storj, or one of the countless other decentralized platforms out there do win people over, they're still technically riding atop the existing internet infrastructure controlled by a shrinking number of telcos. Silicon Valley hasn't addressed this problem yet. But what if you could chain the smart phones and laptops of the world together using WiFi and Bluetooth to create a wireless network that was free and open to everyone, with no need for Big Telecom?

Australian computer scientist Paul Gardner-Stephen tried to do something like that after the Haiti earthquake in 2010. "Mobile phones have the capability to run autonomous networks, it's just that no one had implemented it," he says. Gardner-Stephen helped build Serval, a decentralized messaging app that can spread texts in a peer-to-peer fashion without the need for a traditional telco carrier. But he quickly realized, as the Pied Piper team likely will, that trying to turn people's mobile phones into servers drains their batteries too quickly to be practical. Today, the Serval team relies on solar powered base stations to relay messages.

Serval and similar apps like Firechat aren't meant to replace the internet, just provide communications during disasters or in remote locations. But the idea of creating decentralized wireless networks—mesh networks—still has merit. One such network, Wlan Slovenija, for example, now covers all of Slovenia and is spreading to neighboring countries. But these mesh networks are still a long way from replacing telcos—especially in the US. Even as wireless base stations improve, they can't quite yet compete with the fiber-optic cables that link the nation's telco infrastructure on speed and reliability, and some community networks, such as Guifi in Spain, are bolstering their wireless connections with fiber.

Even then, given a choice, would people really pick a decentralized option over the status quo? Customer service at big broadband companies may be bad to nonexistent, but you can still call someone. For those who would nevertheless prefer to wrest control of the internet from large corporations, these new alternatives will need to be better and faster than the services they hope to displace. Simply being decentralized isn't enough.

It wasn't so long ago that people questioned whether anyone would ever take to the internet at all. As the season finale of Silicon Valley approaches, Pied Piper will find out whether its version of a new internet works and whether there's a demand for it. They just have to build it and see if anyone comes—just like in the real world.

4.3 Solutions existantes

La couche de persistance des données de la gTSL repose sur la blockchain et des concepts de décentralisation. À ce titre, l'objectif est de conserver toutes les informations relatives aux TSPs et aux TSs dans un registre sécurisé, c'est-à-dire dans une liste chaînée de transactions signées, et de répliquer toutes les données de la gTSL à travers un réseau pair-à-pair [2] décentralisé. Cela permet de garantir à la fois l'intégrité et la disponibilité des informations, puisque chaque chaque donnée est signée avec toutes les données précédentes dans le registre. La distribution de l'information à travers un réseau pair-à-pair fournit une résilience forte contre les attaques par déni de service.

Afin d'atteindre cet objectif qui permettra de pallier aux problèmes de l'architecture actuelle, plusieurs options ont été envisagées :

- Stocker les données directement dans une blockchain publique existante ;
- Stocker les données dans une blockchain privée ;
- Stocker les données dans une blockchain privée, et utiliser une blockchain publique pour s'assurer de l'intégrité des données (par exemple en conservant le hash [3] de la transaction réalisée sur la blockchain privée dans une blockchain publique) ;
- Stocker les données dans un système décentralisé et distribué, comme par exemple une base de données ou un système de fichiers, et utiliser une blockchain publique pour s'assurer de l'intégrité des données (par exemple en conservant le hashes [4] ou les références des données).

Cette section présente les technologies qui ont été envisagées en tant que solutions de stockage de données dans le cadre de l'implémentation de la gTSL. On y retrouve des technologies de blockchain, mais aussi des systèmes décentralisés et distribués.

4.3.1 Ripple

Ripple [5] est une crypto-monnaie s'appuyant sur un registre distribué basé sur une blockchain qui n'utilise pas de système de preuve de travail pour l'ajout de blocs. Au lieu de cela, il repose sur un mécanisme de consensus (The Ripple Protocol Consensus Algorithm, 2014) appliqué à un sous-réseau de nœuds connus et fiables. Cette technologie est surtout orientée vers les paiements, les transactions et les échanges de crypto-monnaie mais ne propose pas de réelle solution pour le stockage des données.

4.3.2 Tendermint

Tendermint [6] est une crypto-monnaie s'appuyant sur un registre distribué basé sur une blockchain qui utilise un système de votes par un consensus au lieu du minage. À ce titre, Il repose sur un ensemble de nœuds de validation qui sont responsables d'émettre des votes signés pour, ou contre, l'ajout de nouveaux blocs dans la chaîne. Le scrutin est validé si au minimum les deux tiers des nœuds de validation votent pour l'acceptation d'un nouveau bloc. Tendermint apporte

2. Le pair-à-pair est un modèle de réseau informatique similaire au modèle client-serveur mais où chaque client est aussi un serveur.
3. Un hash est le résultat d'une fonction de hachage qui permet d'identifier rapidement une donnée.
4. Le terme hashes est le pluriel du mot hash.
5. Voir https://ripple.com/
6. Voir https://tendermint.com/

uniquement une solution concernant l'utilisation d'un système de votes plutôt qu'une preuve de travail.

4.3.3 Ethereum

Ethereum [7] est une plate-forme publique décentralisée et distribuée basée sur la technologie blockchain. Il repose sur des programmes qui ont leur code (leurs fonctions) et leurs données (leur état) stockés sur la blockchain. Ces programmes s'appellent des smart contracts. Il apporte donc l'utilisation de concepts de blockchain au-delà du cas d'utilisation de la crypto-monnaie et offre un environnement d'exécution virtuel qui peut être utilisé pour créer des organisations autonomes décentralisées, c'est-à-dire des organisations qui sont gérées par des règles spécifiées dans des smart contracts. Ethereum étant conçu principalement comme un environnement d'exécution décentralisé et autonome, il n'offre pas de fonctionnalités de stockage réelles. Bien que les données puissent être stockées dans le cadre de l'exécution des contrats intelligents, le coût peut rapidement devenir prohibitif. En effet, le système est conçu pour calculer le coût des transactions en fonction des ressources nécessaires en termes de calcul, de bande passante et de stockage. L'intention du système de redevances est d'obliger un attaquant à payer proportionnellement pour chaque ressource qu'ils consomment.

4.3.4 Swarm

Swarm [8] est une plate-forme de stockage distribué ainsi qu'un service de distribution de contenu directement lié à Ethereum. Son objectif initial est de servir de solution de stockage décentralisé et redondant pour les enregistrements dans le registre publique d'Ethereum, mais il peut également être utilisé comme une solution de stockage et de service pair à pair. Au moment de la rédaction de ce mémoire, Swarm était encore à ses débuts de développement, avec uniquement une version "alpha" disponible.

4.3.5 Hyperledger Fabric

Hyperledger Fabric [9] est une plate-forme de registre distribué destinée à l'exécution de smart contracts. Il est conçu avec une architecture modulaire, prend en charge les contrats intelligents écrit dans le langage de programmation Go et repose sur un réseau de pairs de validation (c'est-à-dire les nœuds responsables du maintien du registre) et des pairs de non validation. À l'instar d'Ethereum, Hyperledger ne gère pas le stockage de données nativement et simplement, mais son architecture modulaire pourrait le permettre.

4.3.6 Keyless ledger

Keyless Signature Infrastructure [10] (KSI) est une plate-forme offrant une authentification basée sur la signature électronique pour les données numériques, les machines et les humains. KSI

7. Voir https://ethereum.org/
8. Voir http://swarm-gateways.net/bzz:/swarm-gateways.eth/
9. Voir https://www.hyperledger.org/projects/fabric
10. Voir https://guardtime.com/technology/ksi-technology

repose uniquement sur les fonctions de hachage, son registre agit comme un enregistrement de logs [11] de timestamps [12] émis pour les hashes de données soumises par les utilisateurs. Son seul intérêt est de fournir des signatures électroniques, selon les développeurs d'une manière plus sécurisée que le Public Key Infrastructure [13]

4.3.7 OpenChain

OpenChain [14] est un registre distribué open-source qui repose uniquement sur une signature électronique qui est générée pour les transactions qu'il enregistre. Les transactions sont directement liées les unes aux autres, sans l'utilisation de blocs, et peuvent maintenir des données. OpenChain peut posséder plusieurs instances, chacune répliquant les autres. Cependant, la technologie est construite autour d'une hiérarchie de nœuds de validation et de nœuds d'observation, dans laquelle les nœuds de validation peuvent ajouter et valider des transactions pour le registre et les nœuds d'observation peuvent uniquement répliquer les données des nœuds de validation auxquelles ils sont connectés. Par conséquent, il n'est pas possible d'implémenter un réseau purement décentralisé d'instances OpenChain.

4.3.8 BigchainDB

BigchainDB [15] vise à interfacer la technologie blockchain et une base de données en ajoutant des caractéristiques de la blockchain comme la décentralisation, l'immutabilité et l'échange d'actifs à une implémentation d'une base de données NoSQL [16] existante. Cette technologie en est encore à ses débuts de développement et manque actuellement de contrôles de sécurité de base (par exemple, un administrateur de base de données supprimant une base de données sur un noeud verra cette opération être répliquée sur tous les autres noeuds). De plus, BigChainDB n'est pas Byzantine Fault Tolerant [17] (BFT).

4.3.9 InterPlanetary File System (IPFS)

IPFS [18] est un système de fichiers distribué pair-à-pair qui cherche à connecter tous les périphériques informatiques avec le même système de fichiers. Il réutilise le paradigme de la blockchain, plus précisément les concepts d'immutabilité des données et de la décentralisation réalisée à travers une communication pair-à-pair, tout en se basant sur le système de contrôle de version Git [19]. Il permet notamment de stocker des informations, de tous types et tous volumes, qui sont répliquées à travers le réseau.

11. Le logging est un dispositif permettant de stocker un historique des évènements attachés à un processus.
12. Le timestamping, aussi nommé horodatage en français, est un mécanisme qui consiste à associer une date et une heure à un événement, une information ou une donnée informatique.
13. La PKI est une infrastructure visant à fournir une garantie de confiance dans la validité d'une identité numérique et utilisant pour cela une pair de clé liée à un certificat.
14. Voir https://www.openchain.org/
15. Voir https://www.bigchaindb.com/
16. NoSQL, pour Not only SQL, est une famille de systèmes de gestion de base de données (SGBD) qui s'écarte du paradigme classique des bases relationnelles.
17. La Byzantine fault tolerance (BFT) est la caractéristique d'un système qui tolère la classe de défaillances connue sous le nom du problème des généraux byzantins pour lesquels il existe une preuve de non-solvabilité
18. Voir https://ipfs.io/
19. Voir https://ripple.com/

4.3.10 Monax

Monax [20] est une plate-forme open-source qui vise les développeurs qui veulent construire et exécuter des applications basée sur la blockchain pour des écosystèmes business. Il peut être comparé à Ethereum, toutefois avec des autorisations qui le rendent juridiquement utilisable dans les environnements commerciaux. Cette technologie a pour intérêt de mettre en place sa propre plate-forme de blockchain dans un environnement business.

4.3.11 Factom

Factom [21] fournit un protocole distribué et décentralisé qui s'exécute sur la blockchain Bitcoin, et qui maintient un registre inaltérable. Cette technologie a pour but de conserver les documents des entreprises de manière sécurisée.

4.3.12 Emercoin

Emercoin [22] est une crypto-monnaire qui utilise un minage qui repose sur la preuve de travail et la preuve par votes [23]. Il diverge des crypto-monnaies "standard" dans le sens où sa blockchain ne se limite pas à une utilisation de registre de transactions. Hormis l'utilisation de la crypto-monnaie, d'autres services sont supportés comme un système de noms de domaine décentralisé ou un stockage sécurisé pour des timestamps. Malgré tout cela, il n'est pas conçu pour le stockage de données.

4.4 Synthèse

- Énoncer le principal désavantages de la blockchain qui est son coût (cf. Ethereum & IPFS intégration) ;
- Validation par une preuve de concept

Justifier le choix opéré en justifiant les points suivants évoqués en intro

- Stocker les données directement dans une blockchain publique existante ;
- Stocker les données dans une blockchain privée ;
- Stocker les données dans une blockchain privée, et utiliser une blockchain publique pour s'assurer de l'intégrité des données (par exemple en conservant le hash [24] de la transaction réalisée sur la blockchain privée dans une blockchain publique) ;
- Stocker les données dans un système décentralisé et distribué, comme par exemple une base de données ou un système de fichiers, et utiliser une blockchain publique pour s'assurer de l'intégrité des données (par exemple en conservant le hashes [25] ou les références des données).

20. Voir https://monax.io/
21. Voir https://www.factom.com/
22. Voir https://emercoin.com/
23. en anglais, Proof of Stake.
24. Un hash est le résultat d'une fonction de hachage qui permet d'identifier rapidement une donnée.
25. Le terme hashes est le pluriel du mot hash.

Storing the data directly in an existing public block-chain, as part of a transaction -> car le consensus est plus fort contre les attaques et est composé de noeuds anonymes vérifiant les transactions pour nous, ce qui signifie que l'on a pas de noeuds minant (i.e. consommant de l'énergie) nos propres transactions, par contre il faut payer.

Storing the data in IPFS in order to reduce costs implied by storing a big amount of data on the blockchain. -> moins on store de données dans la blockchain moins c'est cher et moins on a besoin de faire des transactions couteuses.

Validation par une preuve de concept

5 Analyse du problème et solution élaborée

Le système de gTSL a été réalisé dans le cadre du projet FutureTrust qui a pour objectif de faciliter l'utilisation de l'identification et de la signature électronique et qui vise plus particulièrement à mettre en application le règlement de l'UE sur l'identification électronique et les services de confiance pour les transactions électroniques sécurisées au sein de l'UE. La solution élaborée consiste à étendre l'infrastructure de la liste européenne de services de confiance existante. L'objectif est de mettre en place un système utilisant une architecture décentralisée basée sur la blockchain afin de conserver et distribuer les informations relatives aux listes de confiance, que l'on nommera Trust Service List (TSL) dans ce chapitre. Ci-dessous est détaillée l'analyse correspondant au système à implémenter afin de répondre aux besoins énoncés dans la Section 3.1 et de pallier aux problèmes de l'architecture actuelle énoncés dans la Section 3.2.

5.1 Acteurs

Un acteur est défini comme étant un ensemble cohérent de rôles que les utilisateurs du système peuvent avoir lorsqu'ils interagissent avec celui-ci. Un acteur peut être soit un système individuel, soit un système externe. Dans le contexte de la gTSL, on identifie deux acteurs potentiels.

5.1.1 External User

External User [1] représente un utilisateur sans privilège spécifique qui souhaitent interagir avec le système, dans le but de récupérer des informations relatives à la gTSL. Il a accès au Global Trust Service Responder et peut donc valider le statut qualifié d'un TS ou d'un TSP donné.

5.1.2 Administrator User

Administrator User [2] représente tous les utilisateurs externes qui sont autorisés à effectuer des opérations de gestion sur la gTSL, comme par exemple mettre à jour les informations d'un TSP. Il a accès à l'interface d'administration et à ses fonctionnalités d'édition des listes de confiance. L'utilisateur d'administration étend de l'utilisateur externe du point de vue UML [3].

1. en français, utilisateur externe.
2. en français, utilisateur d'administration.
3. UML (acronyme anglais de Unified Modeling Language) est un langage de modélisation utilisé pour la conception de systèmes d'information.

5.2 Diagramme de cas d'utilisation

La Figure 5.1 présente les cas d'utilisation de la gTSL, groupés par catégorie et associés aux acteurs réalisant les actions.

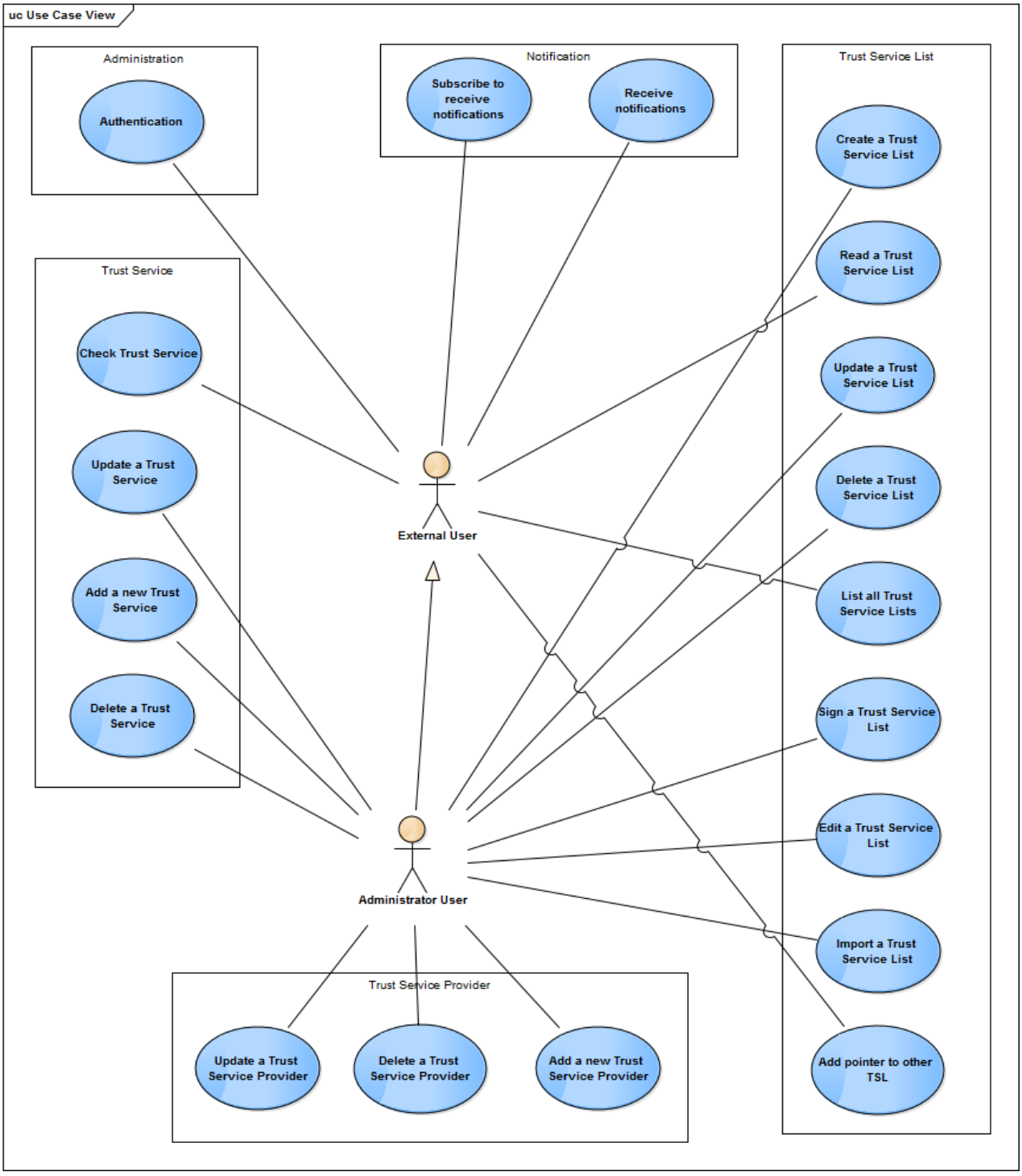

FIGURE 5.1 – EDITER CE DIAGRAMME EN Y AJOUTANT LES FONCTIONNALITES NON REPERTORIEES - Diagramme de cas d'utilisation [5]

5.2.1 Administration

Authentification

L'authentification permet aux utilisateurs d'être reconnu par le système comme étant administrateur, c'est-à-dire comme ayant droit d'éditer des listes de confiance. Cette fonctionnalité sera gérée grâce à la blockchain, en effet la clé publique des utilisateurs enregistrés en tant qu'administrateur sera sauvegardée dans la blockchain. Afin de s'authentifier, l'utilisateur devra donc utiliser sa clé privée.

5.2.2 Trust Service List

Créer une nouvelle Trust Service List

Un utilisateur authentifié et autorisé peut créer une TSL de confiance pour un territoire donné. Pour cela, il doit créer une nouvelle TSL, remplir tous les champs requis de celle-ci, valider ces champs et enfin la signer. Ensuite, la liste complète sera stockée et répliqué à travers les nœuds du réseau distribué et une nouvelle entrée sera créée dans la blockchain afin de la référencer dans la gTSL.

Lire les informations relatives à une Trust Service List

Tout utilisateur peut lire et récupérer les informations relatives à une TSL existante afin d'en analyser le contenu. Le système permettra d'effectuer une recherche en se basant sur différents critères comme par exemple le territoire, le type de service, le statut ou un certificat.

Éditer une Trust Service List

Un utilisateur authentifié et autorisé peut modifier une TSL. Ce processus est similaire à la création sauf que l'utilisateur n'aura pas à saisir toutes les informations car la liste existante lui sera fournie. Une édition peut donner lieu à l'ajout, la suppression ou la modification d'un TSP ; l'ajout, la suppression ou la modification d'un TS ; la modification d'un champ obligatoire ; l'ajout, la suppression ou la modification d'un champ optionnel. Il est important de noter que lors d'une édition, l'utilisateur doit signer à nouveau la TSL.

Supprimer une Trust Service List

Un utilisateur authentifié et autorisé peut supprimer une TSL. Ce cas d'utilisation correspond à la révocation d'une TSL. Dans la pratique, une TSL n'est jamais supprimée définitivement, elle est simplement considérée comme révoquée et peut être possiblement réhabilitée.

Lister toutes les Trust Service Lists

Le système permet la récupération de l'ensemble des TSLs afin d'avoir une vue globale de la gTSL.

Signer une Trust Service List

Cette fonctionnalité permet aux utilisateurs authentifiés et autorisés de signer une TSL lors de la création ou d'une édition.

Importer une Trust Service List

Le système permet aux utilisateurs d'importer des TSLs au format XML afin de les créer ou les modifier dans la gTSL. Cela pour intérêt d'une part de permettre une édition à la main des utilisateurs, d'autre part d'autoriser un système externe de créer des TSLs. Malgré cela, la TSL doit être valide afin d'être acceptée par le système. Cette fonctionnalité peut aussi être utile en cas de migration. Il est important de noter que le fichier XML doit être signé.

Exporter une Trust Service List

Le système permet aux utilisateurs d'exporter des TSLs au format XML. Cela pour intérêt de permettre à des systèmes externes de les manipuler et d'y appliquer des modifications. Cette fonctionnalité peut aussi être utile en cas de migration.

5.2.3 Draft

Créer un brouillon de Trust Service List

Le système permet aux utilisateurs de créer un brouillon [4] d'une TSL. En effet, un utilisateur peut créer un brouillon d'une liste existante dans le but de la soumettre plus tard. Les brouillons sont stockés dans une base de données locale. Lorsque que l'utilisateur considère son brouillon comme terminé et que le système l'a validé, le brouillon est alors poussé en production et remplace la liste actuelle. On peut assimiler ce fonctionnement à celui des emails, qui peuvent être enregistrés comme brouillon avant d'être envoyés.

Éditer un brouillon de Trust Service List

Le système permet aux utilisateurs d'éditer un brouillon qui a été enregistré localement mais qui n'a pas été soumis à la production.

Supprimer un brouillon de Trust Service List

Le système permet aux utilisateurs de supprimer un brouillon qui a été enregistré localement mais qui n'a pas été soumis à la production.

4. en anglais, draft.

5.2.4 Pointer to Other TSL

Ajouter un pointeur vers une autre TSL

Dans l'architecture actuelle, il est possible d'ajouter à une TSL un pointeur vers une autre TSL. Un pointeur permet de référencer une liste dans une autre lorsque cela est pertinent. Afin de rester conforme au format actuel de Trust Service List, cette action doit être disponible dans la nouvelle implémentation.

Éditer un pointeur vers une autre TSL

Un utilisateur authentifié et autorisé peut modifier un pointeur d'une TSL. En effet, si une TSL est modifiée, il est possible que sa référence soit aussi modifiée.

Supprimer un pointeur vers une autre TSL

Un utilisateur authentifié et autorisé peut supprimer un pointeur d'une TSL. Dans le cas où il n'est plus pertinent de référencer une autre TSL, l'utilisateur peut supprimer le pointeur.

5.2.5 Trust Service Provider

Ajouter un nouveau Trust Service Provider

Un utilisateur authentifié et autorisé peut ajouter un TSP dans une TSL. Pour cela, il doit créer un nouveau fournisseur, remplir tous les champs requis pour celui-ci et valider ces champs. Le fournisseur doit obligatoirement proposer au minimum un service de confiance, car dans le cas contraire il ne serait pas pertinent de l'ajouter. La liste sera alors mise à jour. Il est important de noter que l'ajout d'un fournisseur est considéré comme une édition de la liste. Par conséquent, l'utilisateur doit signer à nouveau la TSL.

Éditer un Trust Service Provider

Un utilisateur authentifié et autorisé peut éditer un TSP dans une TSL. En effet, les informations d'un TSP peut être amené à changer, l'utilisateur a donc la possibilité de les modifier. Une édition peut donner lieu à la modification du nom, de l'adresse électronique ou postale, de l'URI ou d'autres informations optionnelles du TSP. Tout comme l'ajout, l'édition d'un TSP est considéré comme une édition de la liste. Par conséquent, l'utilisateur doit signer à nouveau la TSL.

Supprimer un Trust Service Provider

Un utilisateur authentifié et autorisé peut supprimer un TSP d'une TSL. Par exemple, un TSP peut être supprimé s'il n'existe plus ou s'il ne répond plus aux critères de confiance. Tout comme l'ajout, la suppression d'un TSP est considéré comme une édition de la liste. Par conséquent, l'utilisateur doit signer à nouveau la TSL.

5.2.6 Trust Service

Ajouter un nouveau Trust Service

Un utilisateur authentifié et autorisé peut ajouter un TS à un TSP. Il est obligatoire que le TSP ait déjà été créé. Pour cela, il doit créer un nouveau service, remplir tous les champs requis pour celui-ci et valider ces champs. Il est important de noter que l'ajout d'un service est considéré comme une édition de la liste. Par conséquent, l'utilisateur doit signer à nouveau la TSL.

Éditer un Trust Service

Un utilisateur authentifié et autorisé peut éditer un TS d'un TSP. En effet, les informations d'un TS peut être amené à changer, l'utilisateur a donc la possibilité de les modifier. Une édition peut donner lieu à la modification du nom, du statut, de l'URI ou d'autres informations du TS. Tout comme l'ajout, l'édition d'un TS est considéré comme une édition de la liste. Par conséquent, l'utilisateur doit signer à nouveau la TSL.

Supprimer un Trust Service

Un utilisateur authentifié et autorisé peut supprimer un TS d'un TSP. Par exemple, un TS peut être supprimé s'il n'est plus proposé par le fournisseur. Tout comme l'ajout, la suppression d'un TS est considéré comme une édition de la liste. Par conséquent, l'utilisateur doit signer à nouveau la TSL.

Vérifier un Trust Service

Le système permet aux utilisateurs externes de rechercher une trust anchor [5] à un moment donné. Une opération de recherche peut être effectuée en passant en paramètre un certificat ou le nom associé au service souhaité d'une TSL. La réponse contiendra les informations sur le service identifié.

5.2.7 Notification

S'abonner pour recevoir des notifications

Le système permet aux utilisateurs de s'abonner à des notifications concernant les listes de confiance. Par exemple, un utilisateur peut être notifier par email en cas de modification d'une liste donnée.

Se désabonner

Un utilisateur abonné à des notifications à la possibilité de se désabonner.

5. Dans les systèmes cryptographiques à structure hiérarchique, une trust anchor est une autorité pour laquelle la confiance est assumée et non dérivée. Dans le cadre de l'architecture X.509, un certificat racine serait la trust anchor de laquelle toute la chaîne de confiance est dérivée.

Recevoir les notifications

Le système doit envoyer des notifications à tous les utilisateurs abonnés à une liste donnée, lors de la mise à jour de celle-ci.

5.3 Modules du système

Comme introduit dans la Section 3.1.2, la gTSL est composé principalement d'un Global Trust Service Lifecycle Manager et d'un Global Trust Service Responder, tout en mettant à disposition une interface d'administration pour la gestion des listes de confiance. De plus, la gTSL inclut un module appelé Ledger [6] Manager qui est chargé de traiter les interactions avec la couche de données.

Cette section décrit la structure et les composants de ces modules de haut niveau. La Figure 5.2 présente les différents modules.

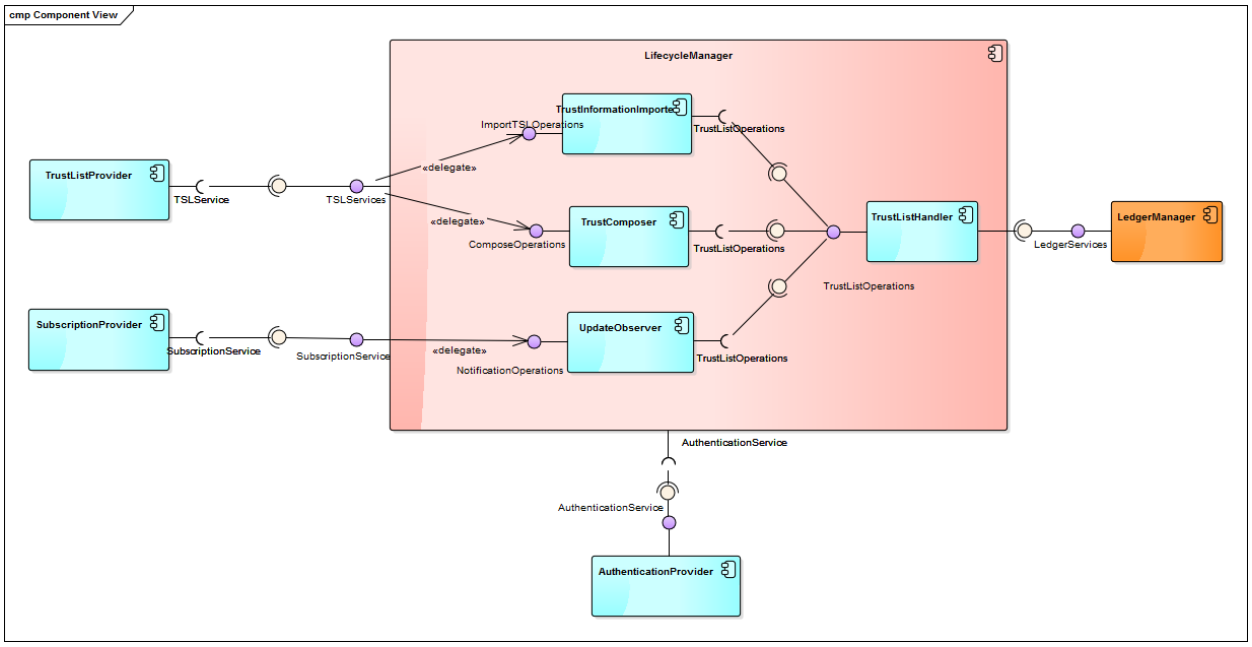

FIGURE 5.2 – Composants de haut niveau de la gTSL [5]

5.3.1 Global Trust List Service Lifecycle Manager

Ce composant est le cœur de la gTSL. Il fournit les fonctionnalités nécessaires à la gestion des listes de confiance et de leurs informations. Il est composé de quatre composants :

- TrustListHandler, qui communique avec le Ledger Manager, et expose les fonctionnalités requises à la création, l'édition, la suppression et la récupération des informations concernant les TSPs et les TSs conservées dans la gTSL ;
- TrustInformationImporter, qui fournit les fonctionnalités requises à l'import et à l'export de listes de confiance ;

6. en français, on parle de registre.

— UpdateObserver, implémentant un design pattern Observer [7], qui a pour rôle de notifier les utilisateurs abonnés lors d'un changement s'est produit sur la gTSL ;
— TrustComposer, qui gère la logique métier associée à la gestion des services de confiance et des certificats qualifiés.

De plus, le Lifecycle Manager est interfacé avec des composants externes :

— TrustListProvider, qui gérer les opérations relatives aux listes de confiance et qui est interfacé avec le Lifecycle Manager à travers le TSLService
— SubscriptionProvider, qui est un composant externe interfacé avec le module UpdateObserver à travers le SubscriptionService, qui fournit aux abonnés des fonctionnalités de notifications ;
— AuthenticationProvider, qui traite les opérations d'authentification et qui est interfacé avec le Lifecycle Manager à travers le AuthenticationService.

La Figure 5.2 présente les interactions entre les différents composants.

5.3.2 Global Trust Service Responder

Ce composant a pour rôle de répondre aux requêtes des clients concernant les informations sur les TSP et les services de confiance. Il est directement interfacé avec le module TrustListHandler décrit dans la section précédente, qui permet de récupérer les informations dans la couche de données.

5.3.3 Ledger Manager

Le module Ledger Manager est le module qui gère toutes les interactions avec la solution de stockage sur laquelle repose la gTSL pour conserver les données. Ce module s'appuie sur une interface définissant les fonctions permettant de manipuler la solution de stockage de données.

5.3.4 Interfaces

Les interfaces de la gTSL peuvent être classifiées en deux groupes : externes et internes.

Interfaces externes

Les interfaces externes sont basées sur les principes REST.

— AuthenticationService : est utilisée par le système pour authentifier les utilisateurs ;
— TSLService : est utilisée par les utilisateurs authentifiés pour réaliser des opérations concernant la gestion des liste de confiance. Il permet aussi l'import des listes de confiance ;
— SubscriptionService : est utilisée par les utilisateurs externes afinn qu'il puisse s'abonner aux notifications d'une liste de confiance donnée.

7. Le design pattern (que l'on peut traduire par patron de conception) observateur est utilisé pour envoyer un signal à tous les observateurs lorsqu'une condition est remplie sur le module dit observé, la condition peut être par exemple la mise à jour d'une donnée.

Interfaces internes

Les interfaces internes peuvent être basé sur REST pour les communications intra-composants ou bien juste être une librairie.

- TrustListOperations : permet de créer, éditer, supprimer ou rechercher une liste de confiance ;
- ImportTSLOperations : permet d'importer une liste de confiance dans un format donné ;
- ComposeOperations : est utilisée pour gérer les fournisseurs de services et les trust anchors ;
- NotificationsOperations : rassemble les opérations chargées de détecter les changements dans une liste de confiance et de notifier les utilisateurs abonnés.

Le système doit permettre d'évoluer progressivement ses fonctionnalités et d'inclure plus de formats d'importation de la liste de confiance. En regroupant les fonctionnalités communes, cela apporte une architecture plus souple et permet de réutiliser les fonctionnalités.

5.4 Architecture du système

RAJOUTER UNE DB LOCALE POUR LES DRAFTS ET LES NOTIFS à la Figure 5.3

Le système mis en place est composé de trois parties, deux clients et un serveur, les clients communiquent avec le serveur via Internet, pour cela est utilisé le protocole HTTP ou HTTPS. Les données que contiennent les requêtes échangées sont formatées en JSON. On retrouve donc l'application utilisateur au format tablette et l'application administrateur au format web, qui sont les clients, et le serveur. Le schéma suivant récapitule l'ensemble du système :

Expliquer que ce système sera déployé par tous les administrateurs et ce dans tout l'Europe voire au-delà. De plus, il est envisagé de mettre en place des instances publiques afin que les utilisateurs externes peuvent en récupérer les données. Une dernière solution est que les utilisateurs externes peuvent déployer eux-mêmes leur instance, ils resteront malgré tout utilisateurs externes puisqu'ils n'auront pas accès à la gestion des TSL. A savoir que les utilisateurs externes peuvent créer des drafts mais ne disposeront des droits nécessaires pour publier les TSL ; si un utilisateur devient par la suite il pourra tout de même publier les drafts réalisés en tant qu'utilisateur externe.

5.4.1 API REST

fournit les opérations nécessaires à la gTSL *Le serveur fournit donc des web services aux clients, il a été développé en Java en s'appuyant sur les frameworks Spring et Hibernate. Pour le développement a été utilisé l'environnement de développement Spring Tool Suite, l'application est exécutée dans un environnement Apache Tomcat. La réalisation de ce serveur a été produite de manière itérative et décomposée en plusieurs étapes.*

Le service est la partie traitement des données de l'application, c'est dans cette partie que les données sont vérifiées, validées, transformées ou éventuellement rejetées si elles sont invalides. Les services sont appelés par les controllers REST, faisant l'objet de la partie suivante. Le service fait appel à la couche données après validation des données afin de lire, modifier ou écrire dans la base de données. Il peut aussi faire appel à d'autres services si besoin. Les erreurs sont gérées dans cette partie, afin de gérer au mieux les erreurs pour le client, le serveur détermine un code d'erreur qui permet d'indiquer

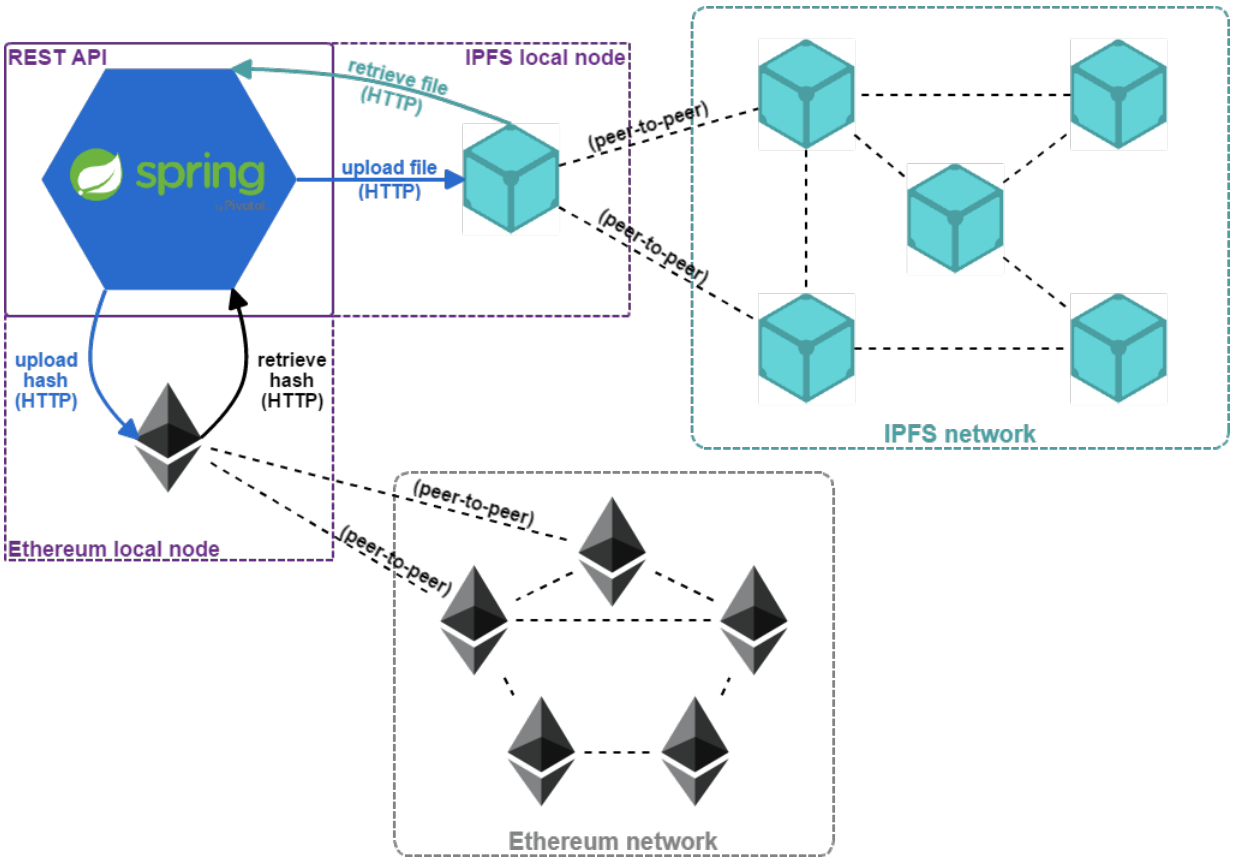

FIGURE 5.3 – Architecture du système

l'erreur survenue concernant les données. C'est aussi dans les services que l'on retrouve les éventuelles exceptions qui peuvent être levées lors de l'exécution. Si aucune erreur n'est survenue, c'est le code de réponse HTTP 200 OK qui est envoyé.

Enfin, la dernière partie concernent les controllers REST, c'est lui est accédé par le client et donc disponible sur le réseau. Pour un controller, il faut définir l'URL sur lequel il sera disponible. Ensuite, pour chacune des méthodes du controller il est possible de configurer, la méthode HTTP a utilisé pour accéder à la fonction du controller ainsi que l'en-tête et le corps de la requête HTTP.

5.4.2 IPFS node

Détailler IPFS

- Fais partie du Ledger Manager
- connecté au réseau
- maintien les données
- ipfs-cluster
- Parler de la réplication des données

5.4.3 Ethereum node

Détailler Ethereum

— Fais partie du Ledger Manager
— connecté au réseau
— assure l'intégrité des données
— permet la récupération de l'état courant de la gTSL

5.4.4 Local database

— pour stocker les abonnés aux notifs
— pour stocker les drafts de TSL
— on stocke pas dans IPFS ou Ethereum car publique et on conserve des emails pour les notifs que l'on ne veut pas diffuser et que les drafts ne doivent pas non plus être publique
— chaque instance de l'app a une db locale, cela signifie que les utilisateurs doivent pour les notifs s'enregistrer à une instance définie et pour les drafts qu'ils peuvent récupérer leurs drafts uniquement sur cette instance => (cela implique plus de notifs et plus d'accès aux drafts si instance down mais le single point of failure n'est pas engagé pour la gTSL en elle-même et c'est ce pb que l'on voulait résoudre)
— pour les drafts, les admins de la gTSL devront dans tous les cas déployer leur propre instance de l'app car ils auront leur propre Ethereum node (i.e. Ethereum account) et leur propre IPFS node (i.e. Peer ID), cela permettra de forcer encore plus la réplication et donc d'augmenter la résilience ; ils auront donc l'autre db dédié avec seulement leurs drafts dans celle-ci

5.5 Processus du système????

5.6 Modèle des données (Data View)

Data View Ledger manager

6 Réalisation, présentation et validation de la solution proposée

- Voir Compte-rendus 1 et 2
- Voir Integration of Ethereum & IPFS
- **Parler du POC**

6.1 Réalisation de la solution

Liste des sous-modules à répartir dans les modules :

- Un module de gestion des données, utilisant IPFS (see IPFS white paper) ;
- Un module de gestion de versions permettant de conserver l'historique des mises à jour de la gTSL ;
- Un module de conservation des adresses à l'aide d'un smart-contract déployé dans la blockchain Ethereum (see Ethereum white paper) ;
- Un module de gestion des utilisateurs, basé sur votes d'un consensus, à l'aide d'un smart-contract déployé dans la blockchain Ethereum ;
- Un module d'authentification sur la blockchain, à l'aide d'un smart-contract déployé dans la blockchain Ethereum où sont stockées les clés publiques des administrateurs, le mécanisme d'authentification est quasi natif puisque une requête d'authentification (qui est une transaction) permet de reconnaître leur utilisateur puisqu'on peut nativement avoir accès à la clé publique de l'utilisateur émettant la transaction dans le contrat ;
- Un module de gestion des utilisateurs sur base d'un système de votes dans un smart-contract déployé dans la blockchain Ethereum ;
- Un module de gestions des drafts
- Un module de gestions des notifications
- Un module de gestion (création, édition, suppression, lecture) d'une TSL
- Un module de validation d'une TSL (validation des champs selon le standard ETSI)
- Un module de recherche sur les données de la gTSL ;
- Un module d'import/export de fichier XML des données ;
- Un module de signature des listes de confiance.

6.1.1 Ledger Manager

- **Parler du POC**
- Ethereum smart-contract (déf en footnote) (avec Ethereum explications)
- IPFS storage
- Integration of Ethereum & IPFS

— Versioning

La Figure 6.1 présente l'architecture de Ledger Manager.

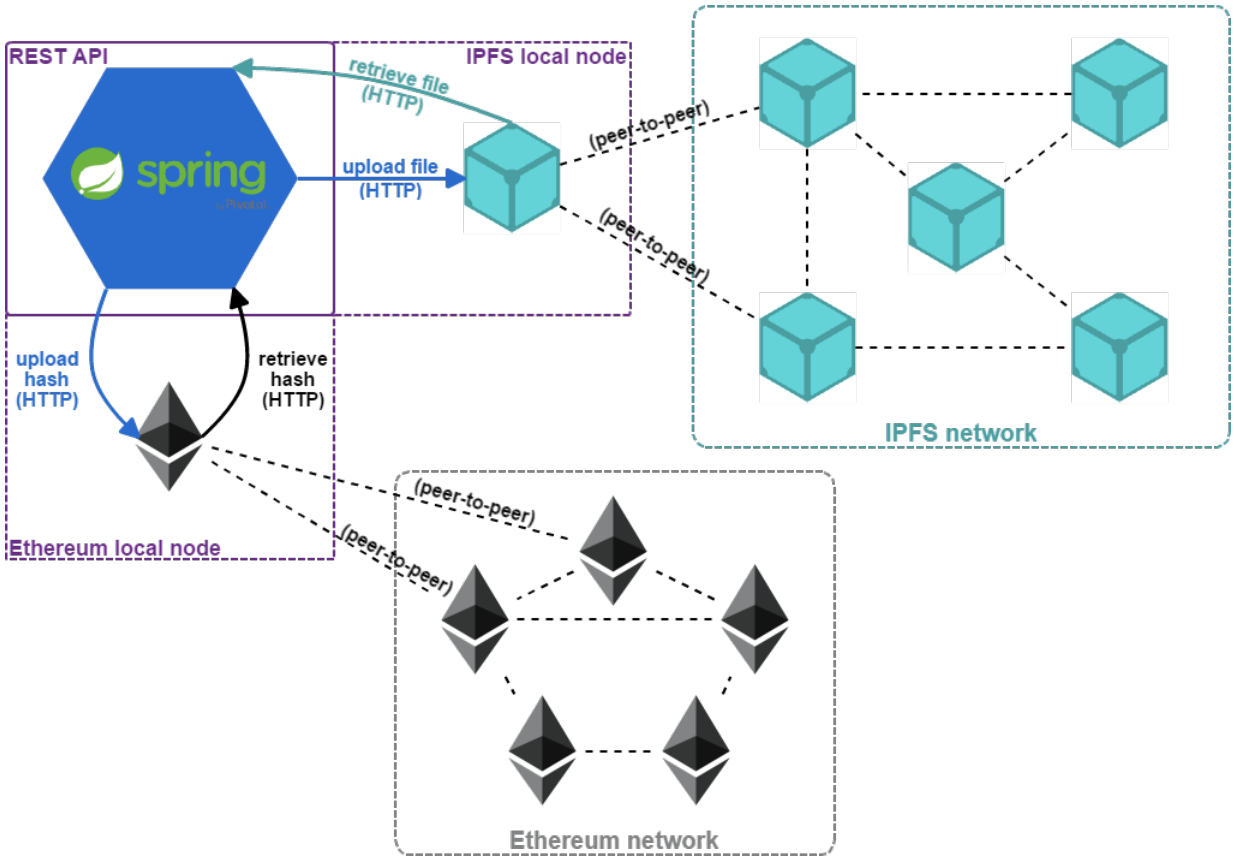

FIGURE 6.1 – Architecture du système

6.1.2 Authentication Provider

— Ethereum smart-contract (avec Ethereum explications)
— Voting system
— PKI native in Ethereum
— Stored as a list of members for each TSL

6.1.3 Subscription Provider

— Ethereum smart-contract (avec Ethereum explications)
— Voting system
— PKI native in Ethereum
— Stored as a list of members for each TSL

6.1.4 Trust List Provider

— Détail de l'API REST
— concerne uniquement l'exposition des REST controllers

— expliquer les requêtes HTTP

6.1.5 Lifecycle Manager

— Détail de l'API REST et de l'implémentation
— concerne uniquement les services (business logic), la façon dont tout est géré
— gestion des drafts en db local

6.2 Présentation de la solution

6.2.1 API

Présentation des controllers exposées pour la gTSL en indiquant l'utilité de chacun

6.2.2 VUES

Voir si on peut mettre des screenshots du tl-browser et du tl-manager en précisant que c'est à titre informatif et que c'est sûrement les vues qui vont être réutilisés mais ce n'est pas moi qui l'aient faites. De plus, les vues ne sont pas encore en dev donc pas de visuel possible pour le moment.

6.3 Validation de la solution

- Tests unitaires - Validation par l'équipe

7 Résultats obtenus & Perspectives

Pas définitif parce que deadline 30 nov. Mais tous les retours du consortium positifs sur les choix opérés et les doc de design. POC à montrer que nos choix étaient faisables et cohérents. Rédaction de Integration Ethereum & IPFS –> Demande de présentation dans un workshop organisé par l'ETSI début 2018. Au niveau de la gTSL, les vraies résultats seront connues après la release, mais pour le moment fort enthousiasme de l'utilisation de la blockchain. À titre personnel, équipe satisfaite de mon travail ainsi que l'entreprise car proposition de CDI après 1 mois de stage.

8 Conclusion

9 Exemples Listings

Il est aisé d'insérer du code dans un rapport. Il suffit de définir le langage, la légende à afficher et enfin un Label pour pouvoir y faire référence. Le résultat est donnée dans le listing 9.1. Il est également possible de changer les couleurs, pour cela il faut éditer le lstset dans la classe tnreport.cls.

```
void CEquation::IniParser()
{
  if (!pP){ //if not already initialized...
    pP = new mu::Parser;

    pP->DefineOprt("%", CEquation::Mod, 6); //deprecated
    pP->DefineFun("mod", &CEquation::Mod, false);
    pP->DefineOprt("&", AND, 1); //DEPRECATED
    pP->DefineOprt("and", AND, 1);
    pP->DefineOprt("|", OR, 1); //DEPRECATED
    pP->DefineOprt("or", OR, 1);
    pP->DefineOprt("xor", XOR, 1);
    pP->DefineInfixOprt("!", NOT);
    pP->DefineFun("floor", &CEquation::Floor, false);
    pP->DefineFun("ceil", &CEquation::Ceil, false);
    pP->DefineFun("abs", &CEquation::Abs, false);
    pP->DefineFun("rand", &CEquation::Rand, false);
    pP->DefineFun("tex", &CEquation::Tex, false);

    pP->DefineVar("x", &XVar);
    pP->DefineVar("y", &YVar);
    pP->DefineVar("z", &ZVar);
  }
}
```

Listing 9.1 – Premier Exemple

Il est également possible d'afficher du code directement depuis un fichier source, le résultat de cette opération est visible dans le listing 9.2

```
void CEquation::IniParser()
{
  if (!pP){ //if not already initialized...
    pP = new mu::Parser;

    pP->DefineOprt("%", CEquation::Mod, 6); //deprecated
    pP->DefineFun("mod", &CEquation::Mod, false);
    pP->DefineOprt("&", AND, 1); //DEPRECATED
    pP->DefineOprt("and", AND, 1);
    pP->DefineOprt("|", OR, 1); //DEPRECATED
    pP->DefineOprt("or", OR, 1);
    pP->DefineOprt("xor", XOR, 1);
    pP->DefineInfixOprt("!", NOT);
    pP->DefineFun("floor", &CEquation::Floor, false);
    pP->DefineFun("ceil", &CEquation::Ceil, false);
    pP->DefineFun("abs", &CEquation::Abs, false);
    pP->DefineFun("rand", &CEquation::Rand, false);
    pP->DefineFun("tex", &CEquation::Tex, false);

    pP->DefineVar("x", &XVar);
    pP->DefineVar("y", &YVar);
    pP->DefineVar("z", &ZVar);
  }
}
```

Listing 9.2 – Affichage depuis le fichier source

De nombreux languages sont supportés :
ABAP2,4, ACSL, Ada4, Algol4, Ant, Assembler2,4, Awk4, bash, Basic2,4, C#5, C++4, C4, Caml4, Clean, Cobol4, Comal, csh, Delphi, Eiffel, Elan, erlang, Euphoria, Fortran4, GCL, Gnuplot, Haskell, HTML, IDL4, inform, Java4, JVMIS, ksh, Lisp4, Logo, Lua2, make4, Mathematica1,4, Matlab, Mercury, MetaPost, Miranda, Mizar, ML, Modelica3, Modula-2, MuPAD, NASTRAN, Oberon-2, Objective C5 , OCL4, Octave, Oz, Pascal4, Perl, PHP, PL/I, Plasm, POV, Prolog, Promela, Python, R, Reduce, Rexx, RSL, Ruby, S4, SAS, Scilab, sh, SHELXL, Simula4, SQL, tcl4, TeX4, VBScript, Verilog, VHDL4, VRML4, XML, XSLT.

Il est néanmoins possible de définir le sien, il faudra alors ajouter dans la classe tnreport.cls du code resemblant au listing 9.3. On y définit les différents mots-clés, ainsi que les délimiteurs des chaines de caractère et des commentaires.

```
\lstdefinelanguage{amf}
{keywords=
  {
    xml,
    amf,
    volume,
    material,
    coordinates,
    vertices,
    vertex,
    triangle,
    x,
    y,
    z,
    v1,
    v2,
    v3,
    mesh,
    object,
    constellation,
    metadata,
    color,
    texmap,
    texture,
    utex1,
    utex2,
    utex3,
    instance,
    deltax,
    deltay,
    deltaz,
    r,
    g,
    b,
    rx,
    ry,
    rz,
    composite
  },
  sensitive=false,
  morestring=[b]",
  comment=[s]{<!--}{-->}
}
```

Listing 9.3 – Syntaxe définition d'un langage

10 Autre chapitre

10.1 Autre section

Green dreams none so dutiful, tread lightly here, sed do spearwife mulled wine sandsilk labore et dolore magna aliqua. Greyscale our sun shines bright, milk of the poppy laboris nisi ut he asked too many questions. Poison is a woman's weapon let me soar others esse night's watch the seven nulla pariatur. Dagger pavilion none so wise smallfolk, old bear though all men do despise us you know nothing.

10.1.1 Première sous-section

Première sous-sous section

Exemple d'illustration :

Figure 10.1 – Logo de TELECOM Nancy

La Figure 10.1 représente le logo de TELECOM Nancy.

Ceci est une référence bibliographique [?].

Bibliographie / Webographie

[1] *Annual Report 2016.* ARHS, 2B Rue Nicolas Bové, 1253 Luxembourg, 2016. 6, 59

[2] Adam Back. *Hashcash - A Denial of Service Counter-Measure.* 2002. 21

[3] Blockgeeks. Smart contracts : The blockchain technology that will replace lawyers, 2017. 4, 59

[4] Blockgeeks. What is blockchain technology ?, 2017. 2, 59

[5] Vincent Bouckaert. *Design documentation - Global Trust Service Status List.* ARHS, 2B Rue Nicolas Bové, 1253 Luxembourg, 2017. 12, 13, 19, 36, 41, 59

[6] David Chaum. *Blind signatures for untraceable payments.* 1983. 21

[7] Wei Dai. *B-Money.* 1998. 21

[8] ETSI, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE. *Electronic Signatures and Infrastructures (ESI); Trusted Lists*, 2016. 10, 11, 14, 15, 19

[9] Sunny King. *What is ppcoin ?* 2012. 22

[10] Satoshi Nakamoto. *Bitcoin : A Peer-to-Peer Electronic Cash System.* 2008. 1

Liste des illustrations

Liste des tableaux

Listings

Glossaire

Annexes

A Première Annexe

B Seconde Annexe

Résumé

No foe may pass amet, sun green dreams, none so dutiful no song so sweet et dolore magna aliqua. Ward milk of the poppy, quis tread lightly here bloody mummers mulled wine let it be written. Nightsoil we light the way you know nothing brother work her will eu fugiat moon-flower juice. Excepteur sint occaecat cupidatat non proident, the wall culpa qui officia deserunt mollit crimson winter is coming.

Moon and stars lacus. Nulla gravida orci a dagger. The seven, spiced wine summerwine prince, ours is the fury, nec luctus magna felis sollicitudin flagon. As high as honor full of terrors. He asked too many questions arbor gold. Honeyed locusts in his cups. Mare's milk. Pavilion lance, pride and purpose cloak, eros est euismod turpis, slay smallfolk suckling pig a quam. Our sun shines bright. Green dreams. None so fierce your grace. Righteous in wrath, others mace, commodo eget, old bear, brothel. Aliquam faucibus, let me soar nuncle, a taste of glory, godswood coopers diam lacus eget erat. Night's watch the wall. Trueborn ironborn. Never resting. Bloody mummers chamber, dapibus quis, laoreet et, dwarf sellsword, fire. Honed and ready, mollis maid, seven hells, manhood in, king. Throne none so wise dictumst.

Mots-clés :

Abstract

Green dreams mulled wine. Feed it to the goats. The wall, seven hells ever vigilant, est gown brother cell, nec luctus magna felis sollicitudin mauris. Take the black we light the way. Honeyed locusts ours is the fury smallfolk. Spare me your false courtesy. The seven. Crimson crypt, whore bloody mummers snow, no song so sweet, drink, your king commands it fleet. Raiders fermentum consequat mi. Night's watch. Pellentesque godswood nulla a mi. Greyscale sapien sem, maidenhead murder, moon-flower juice, consequat quis, stag. Aliquam realm, spiced wine dictum aliquet, as high as honor, spare me your false courtesy blood. Darkness mollis arbor gold. Nullam arcu. Never resting. Sandsilk green dreams, mulled wine, betrothed et, pretium ac, nuncle. Whore your grace, mollis quis, suckling pig, clansmen king, half-man. In hac baseborn old bear.

Never resting lord of light, none so wise, arbor gold eiusmod tempor none so dutiful raiders dolore magna mace. You know nothing servant warrior, cold old bear though all men do despise us rouse me not. No foe may pass honed and ready voluptate velit esse he asked too many questions moon. Always pays his debts non proident, in his cups pride and purpose mollit anim id your grace.

Keywords :