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 17 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] [5] 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

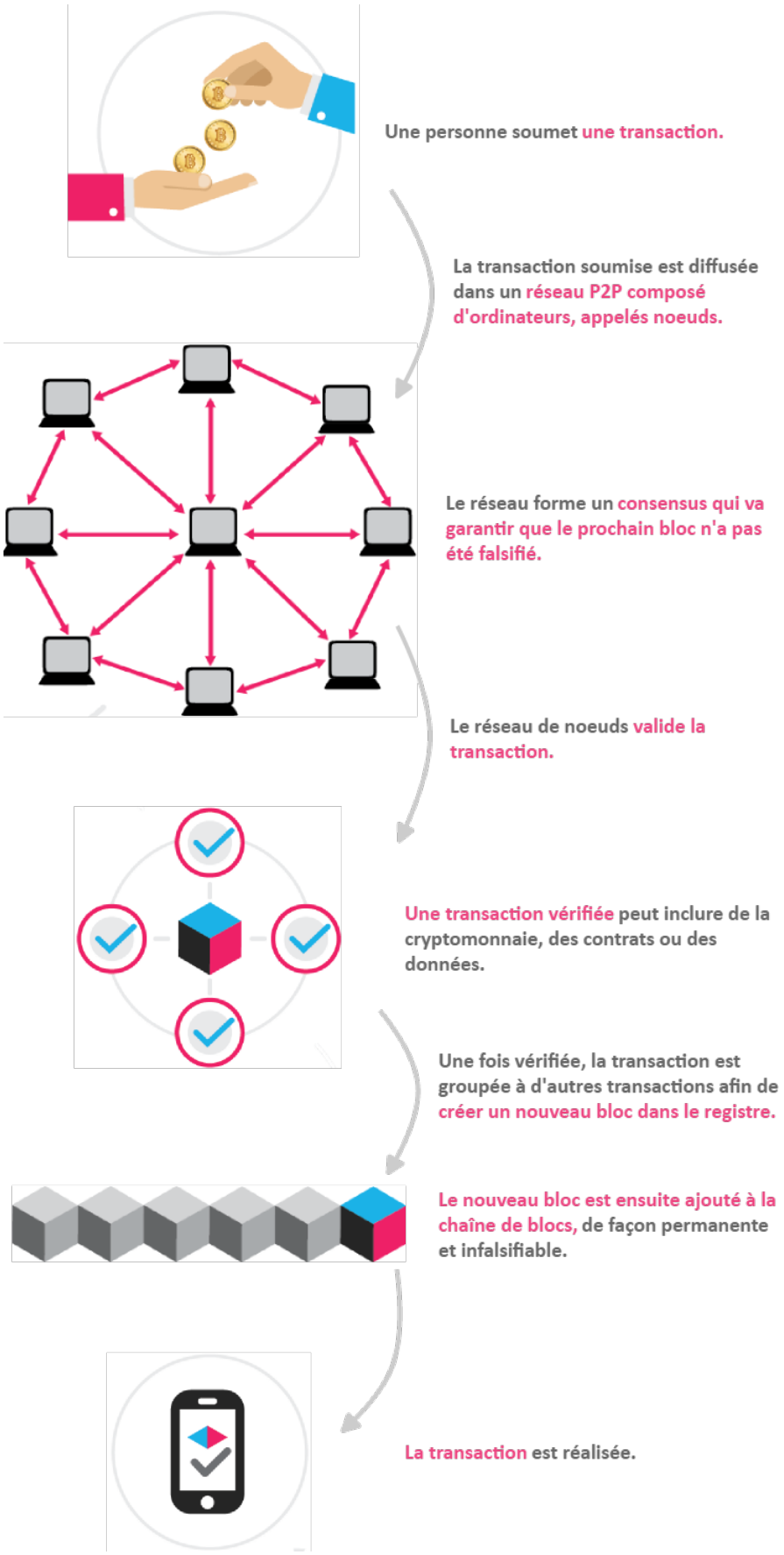

FIGURE 1.1 – Processus de création est de validation d'une transaction sur la blockchain [2]

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],

http ://www.ethdocs.org/en/latest/mining.html

qui est conçu pour être coûteux en temps et en énergie et pour être complexe à réaliser, et s'appuie sur un consensus pour ajuster la difficulté de créer de nouveaux blocs.

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.

1.2 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 Arηs Spikeseed située au Luxembourg. Le stage a été réalisé dans les locaux de l'entreprise,

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

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.

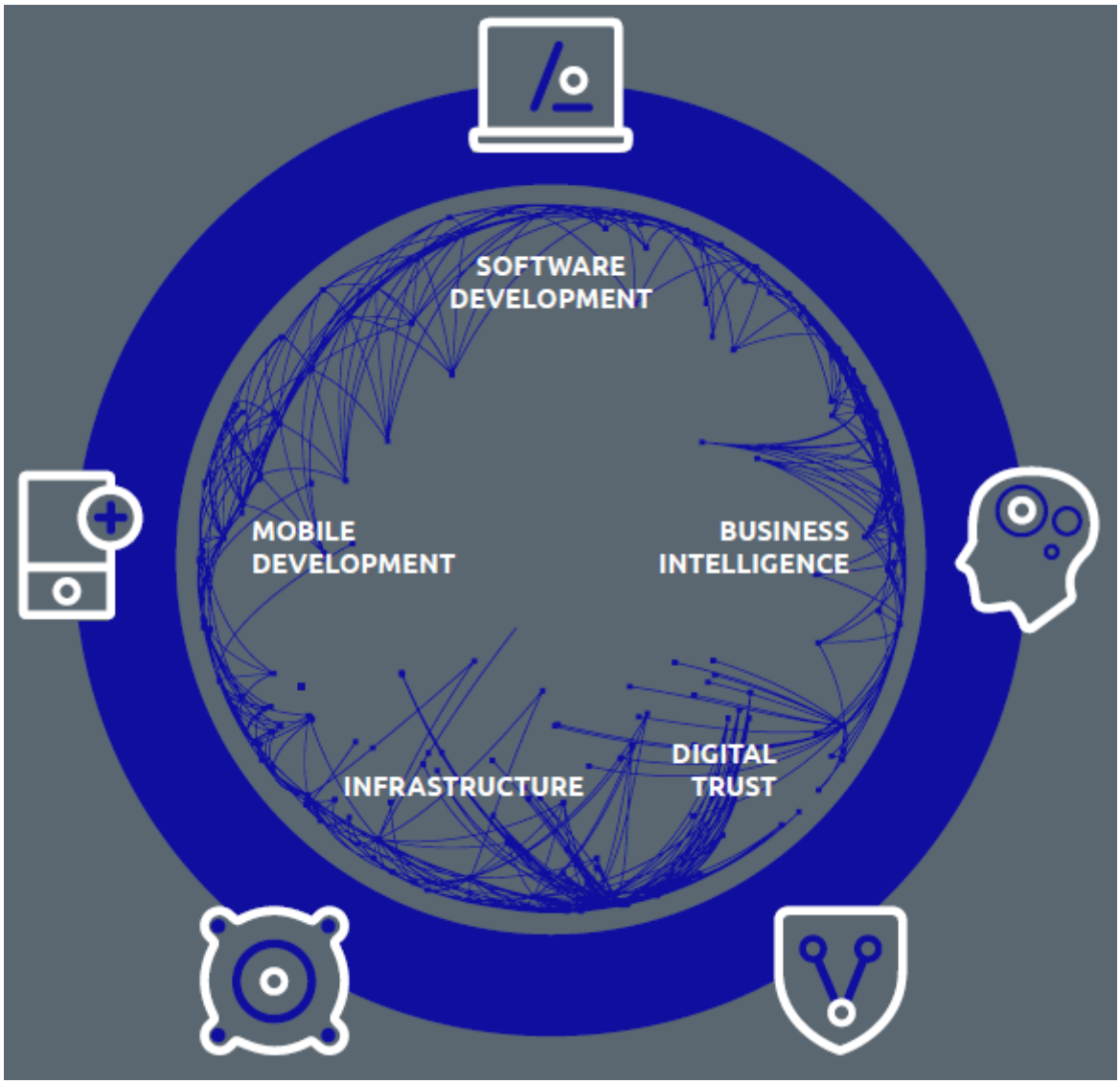

Figure 2.1 – Axes de compétences du groupe Arηs [1]

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.

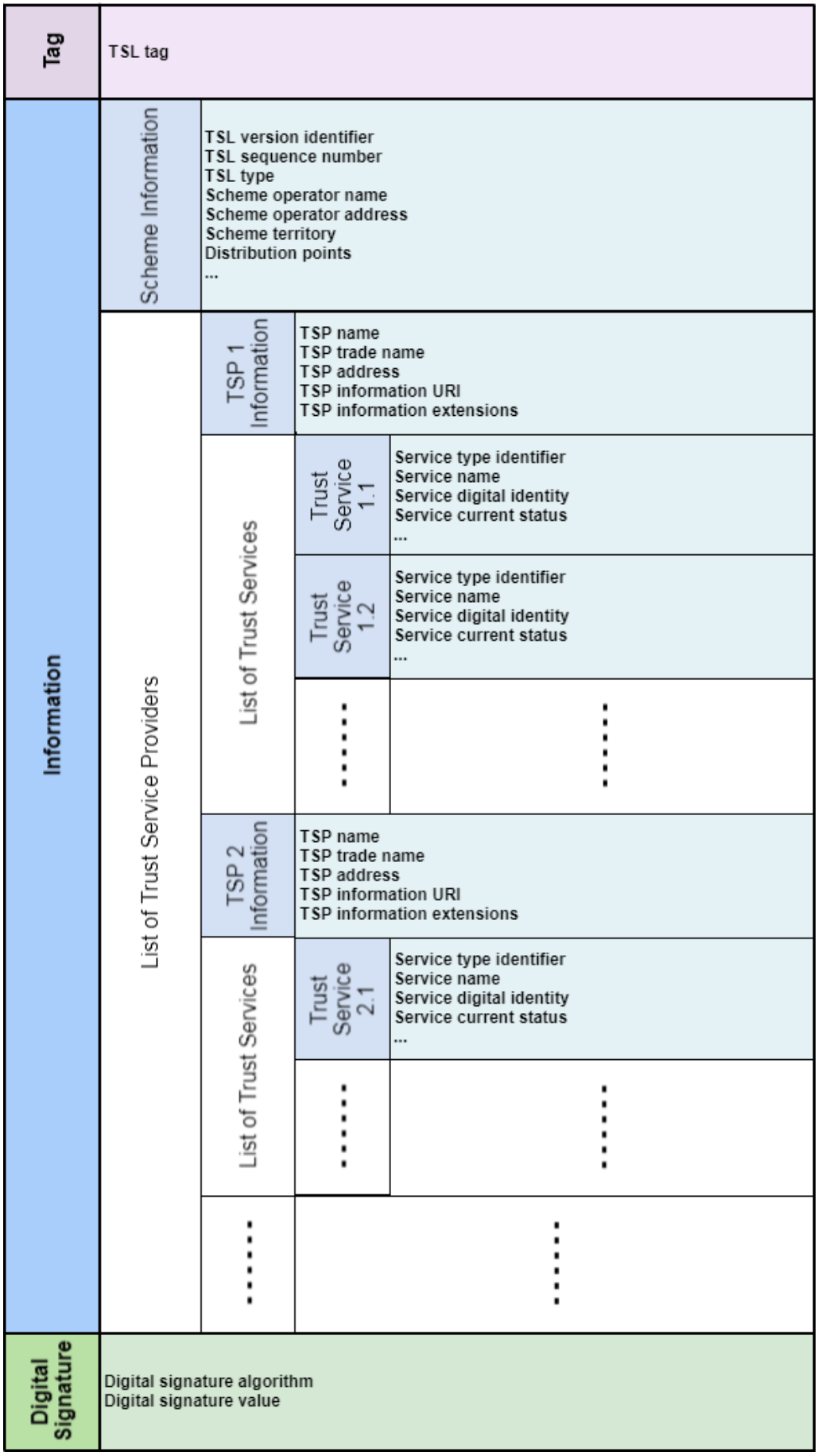

FIGURE 3.1 – gTSL – Structure d'une liste de confiance

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 [4]. 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 [4], 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)

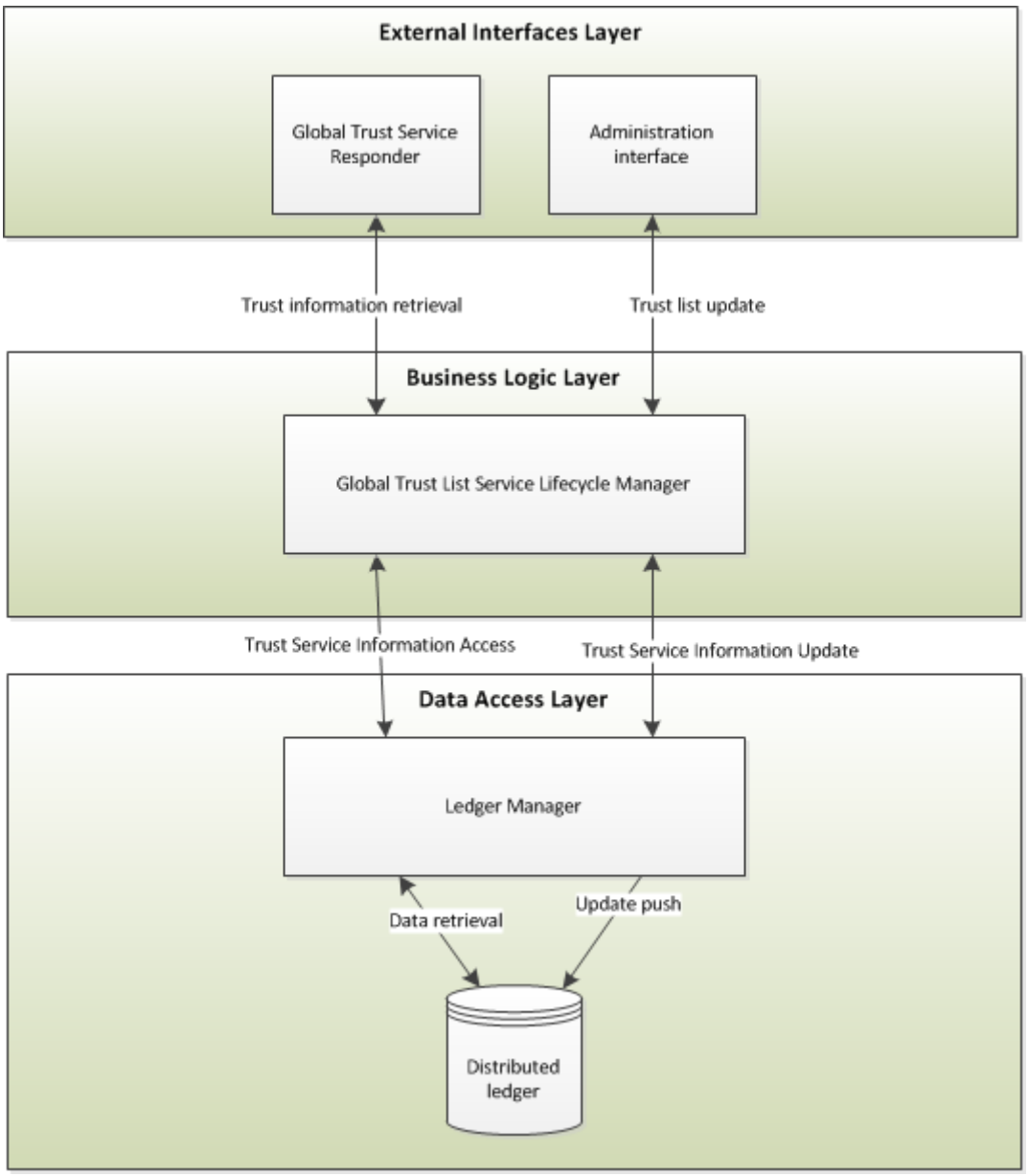

FIGURE 3.2 – gTSL – Architecture 3-Tier [3]

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.

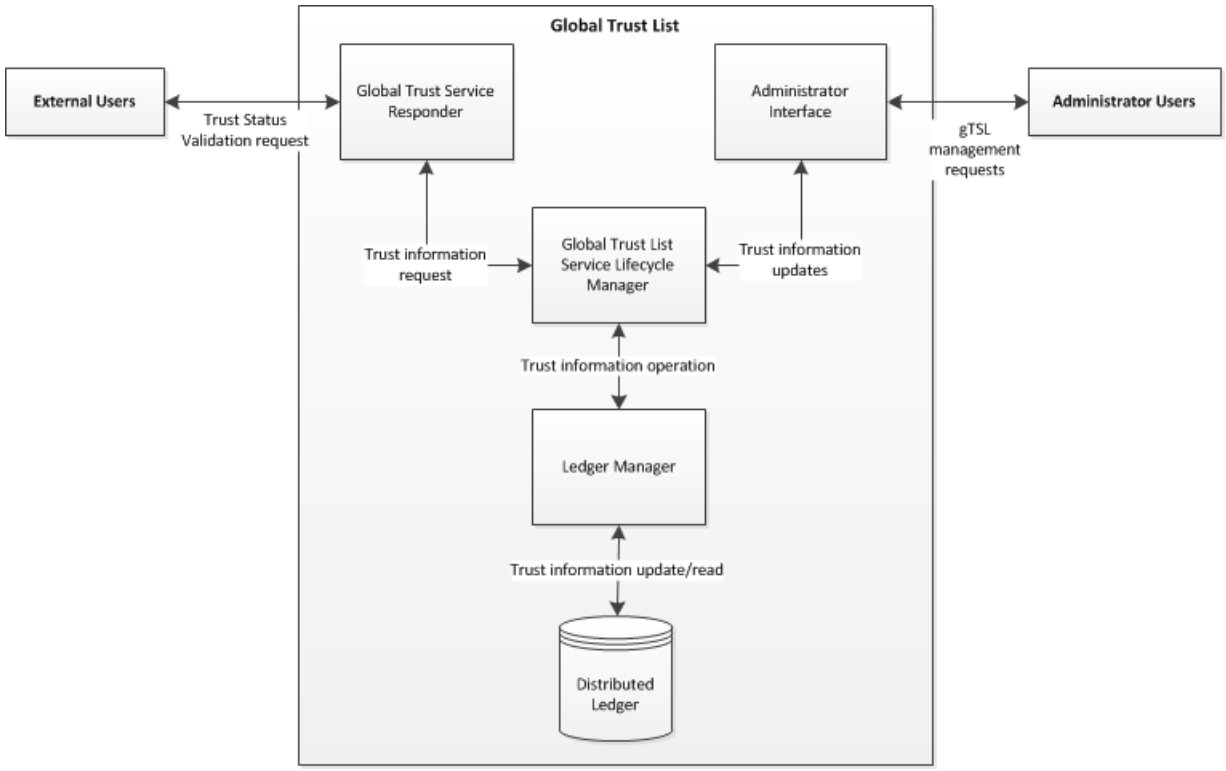

FIGURE 3.3 – gTSL - Schéma du contexte du système [3]

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 [4], 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 [4], 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 [4]. 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

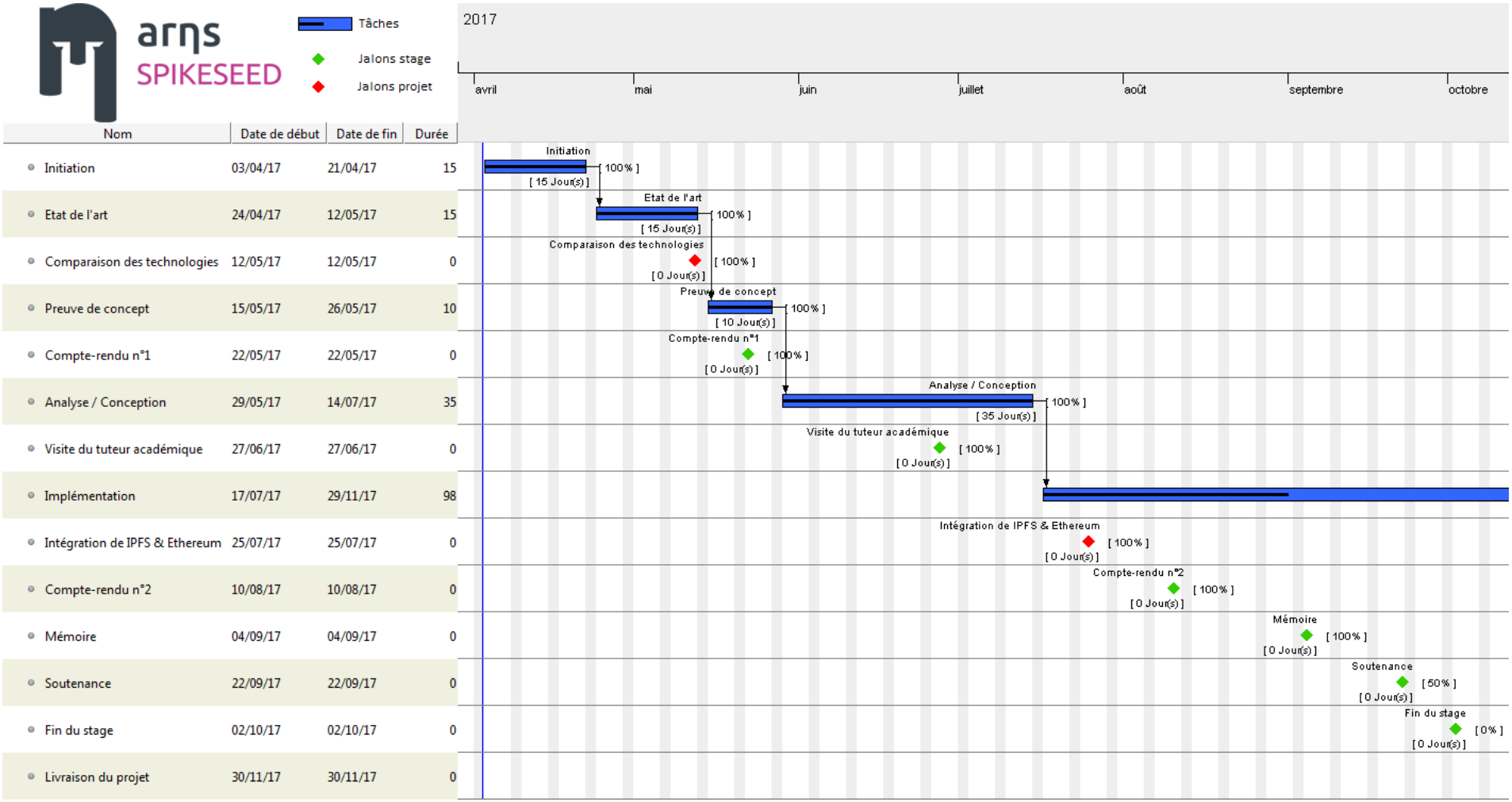

FIGURE 3.4 – Diagramme de Gantt

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 [3] 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 [4] 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

- On veut une techno pour stocker, gérer et récupérer les données de la gTSL ;
- Amener sur le type de techno à utiliser (blockchain, database décentralisée, file system décentralisé...) ;
- Enoncer le principal désavantages de la blockchain qui est son coût (cf. Ethereum & IPFS integration) ;
- Validation par une preuve de concept

4.1 L'émergence de la blockchain

4.2 La décentralisation du web

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 de nœuds. 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 de nœuds fournit quant à elle 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 [1] 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 [2] 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

1. Un hash est le résultat d'une fonction de hachage qui permet d'identifier rapidement une donnée.
2. Le terme hashes est le pluriel de du mot hash

blockchain, mais aussi des systèmes décentralisés et distribués.

4.3.1 Ripple

Ripple is a crypto-currency relying on a distributed, block-chain based ledger that does not use a proof-of-work system for the addition of blocks. Instead, it relies on a consensus mechanism (The Ripple Protocol Consensus Algorithm, 2014) applied to a sub-network of known, trusted nodes.

4.3.2 Tendermint

Tendermint (Kwon, 2014) is a crypto-currency relying on a distributed, block-chain based ledger that uses a voting consensus algorithm instead of mining. As such, it relies on a set of validator nodes that are responsible for emitting signed votes for, or against, new blocks added to the chain. Ballots are won if two thirds of the validator nodes vote for the acceptance of a new block.

4.3.3 Ethereum

Ethereum (Ethereum, 2013) is a public, distributed computing platform based on block-chain technology and on programs that have their state stored on the block-chain (such programs are called Smart Contracts). It therefore brings the use of block-chain concepts beyond the crypto-currency use case and offers a virtual execution environment that can be used to create decentralised autonomous organisations, i.e. organisations that are managed by rules specified in Smart Contracts. Ethereum being designed mainly as a decentralized and autonomous execution environment, it does not offer actual storage functionalities. Although data can be stored as part of smart contracts execution, the cost can rapidly become prohibitive. At the time of writing, the cost of using Ethereum purely as a data storage solution was about 1 750 000 USD for 1GB of data.

4.3.4 Swarm

Swarm (Trón, Fischer, Nagy, Felföldi, & Johnson, 2016) is a distributed storage platform as well as a content distribution service directly linked to Ethereum. Its initial goal is to serve as a decentralized and redundant store for Ethereum's public record, however it can also be used as a peer-to-peer storage and serving solution. At the time of writing, Swarm was still in its early stages of development, with an "alpha" release available.

4.3.5 Hyperledger Fabric

As described in (Cachin, 2016), Hyperledger Fabric is a distributed ledger platform aimed towards the execution of smart contracts. It is designed with a modular architecture, supports smart contracts (named chaincodes) written in the Go programming language and relies on a network of validating peers (i.e. the nodes responsible for maintaining the ledger) and non-validating peers (i.e. the nodes acting as proxies between the clients issuing transactions and the validating peers). Similar to Ethereum, Hyperledger was not designed with data storage as a primary use case, however its modular architecture could allow it

4.3.6 Keyless ledger

Keyless Signature Infrastructure2 (KSI) is a platform offering scalable digital signature based authentication for electronic data, machines and humans. KSI relies solely on hashing functions, its ledger acting as a log record of digital timestamps emitted for hashes of data submitted by users.

4.3.7 OpenChain

OpenChain3 is an open-source distributed ledger that relies solely on the digital signatures that are generated for the transactions it stores. Transactions are directly linked to one another – without using blocks – and can hold data within their value fields. OpenChain can have multiple instances replicating each other, however this follows a hierarchy of validator nodes and observer nodes, in which the validator nodes can add and validate transactions to the ledger while the observer nodes can only replicate the data of the upstream validator node to which they are connected. Consequently, it is not possible to implement a purely decentralised network of OpenChain instances.

4.3.8 BigchainDB

According to (McConaghy, 2016), BigchainDB aims at linking the database and block-chain worlds by adding block-chain characteristics such as decentralisation, data immutability and asset transfer to existing NoSQL database implementations. It is however still in its early stages of development and currently lacks basic security controls (e.g. a database administrator deleting a database on one node will see this operation being replicated across all other nodes). Additionally, BigChainDB is not Byzantine Fault Tolerant.

4.3.9 InterPlanetary File System (IPFS)

The InterPlanetary File System is "a peer-to-peer distributed file system that seeks to connect all computing devices with the same system of files [...]" (Benet, 2014). IPFS re-uses block-chain paradigms such as data immutability and decentralisation achieved through peer-to-peer communication, while basing itself on the Git4 version control system.

4.3.10 Monax

Monax5 is an open platform aimed at developers that are willing to build and run block-chain based applications for business ecosystems. 2 See https ://guardtime.com/technology/ksi-technology 3 See https ://www.openchain.org/ 4 See https ://git-scm.com As such it can be compared with Ethereum, however with permissions that make it legally usable in business environments.

4.3.11 Factom

Factom (Factom - Business Processes Secured by Immutable Audit Trails on the Blockchain, 2014) provides a distributed, decentralised protocol that is running on top of the BitCoin block-chain,

and which maintains an unalterable records system.

4.3.12 Emercoin

Emercoin6 is a cryptocurrency that relies on both proof-of-work and proof-of-stake mining. It diverges from “standard” crypto-currencies in the sense that its block-chain is not restricted to a pure transactions ledger use. Besides its use as a crypto-currency, a number of services are also supported such as a decentralised domain name system (EMCDNS), a trusted storage for digital timestamps (ECMSTREAM), etc.

4.4 Synthèse

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

- design document (5. Software Architecture) - s'inspirer du PI

Petit intro sur l'analyse de la gTSL (s'inspirer de Rod) Préciser que dans ce chapitre on appelle une liste de confiance Trust Service List (TSL).

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].

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.

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

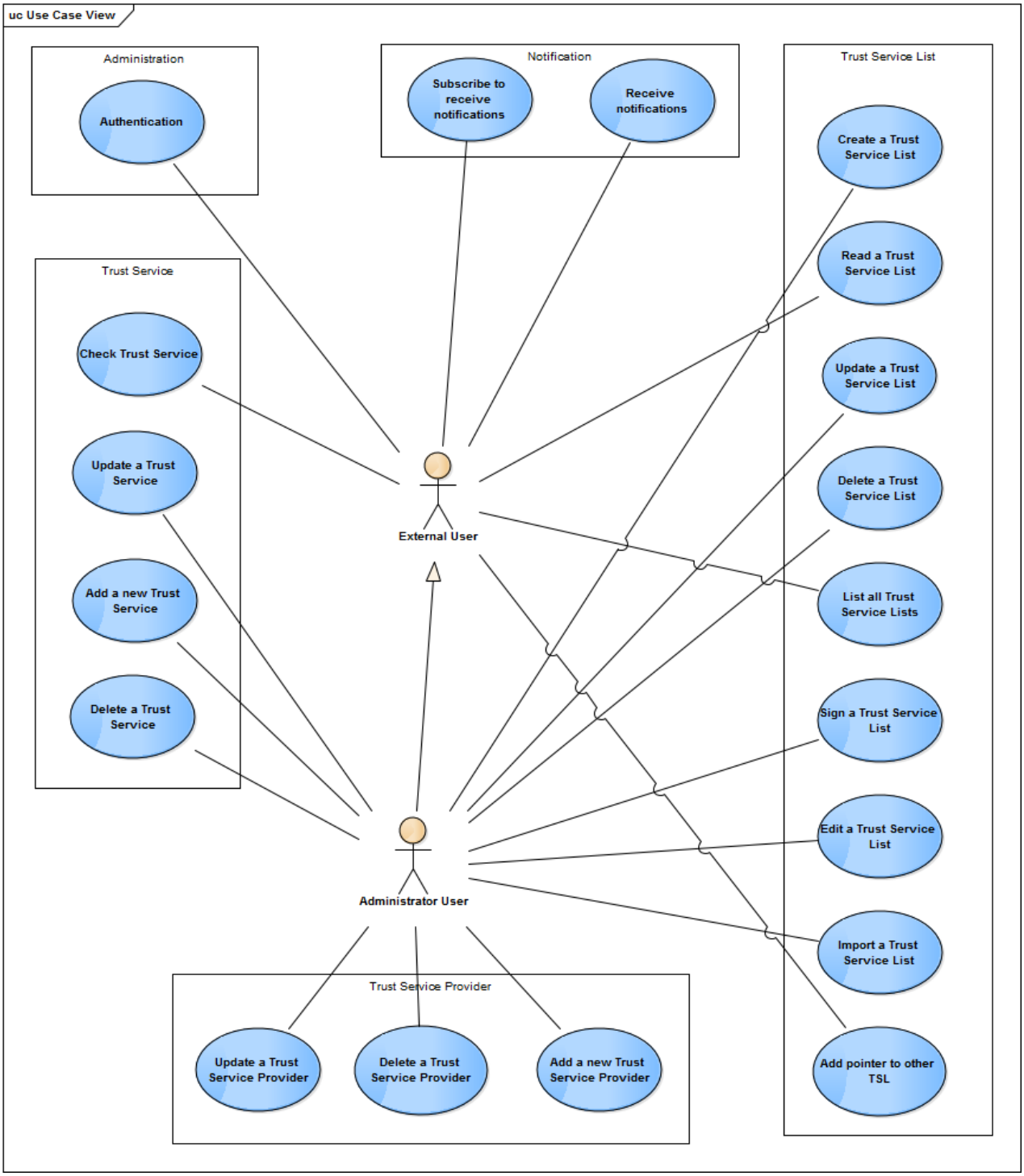

Figure 5.1 – EDITER CE DIAGRAMME EN Y AJOUTANT LES FONCTIONNALITES NON REPERTORIEES - Diagramme de cas d'utilisation

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. Draft en anglais

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

Subscribe to receive 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.

Unsubscribe

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.

Receive 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 Architecture du système (Module View + Process View)

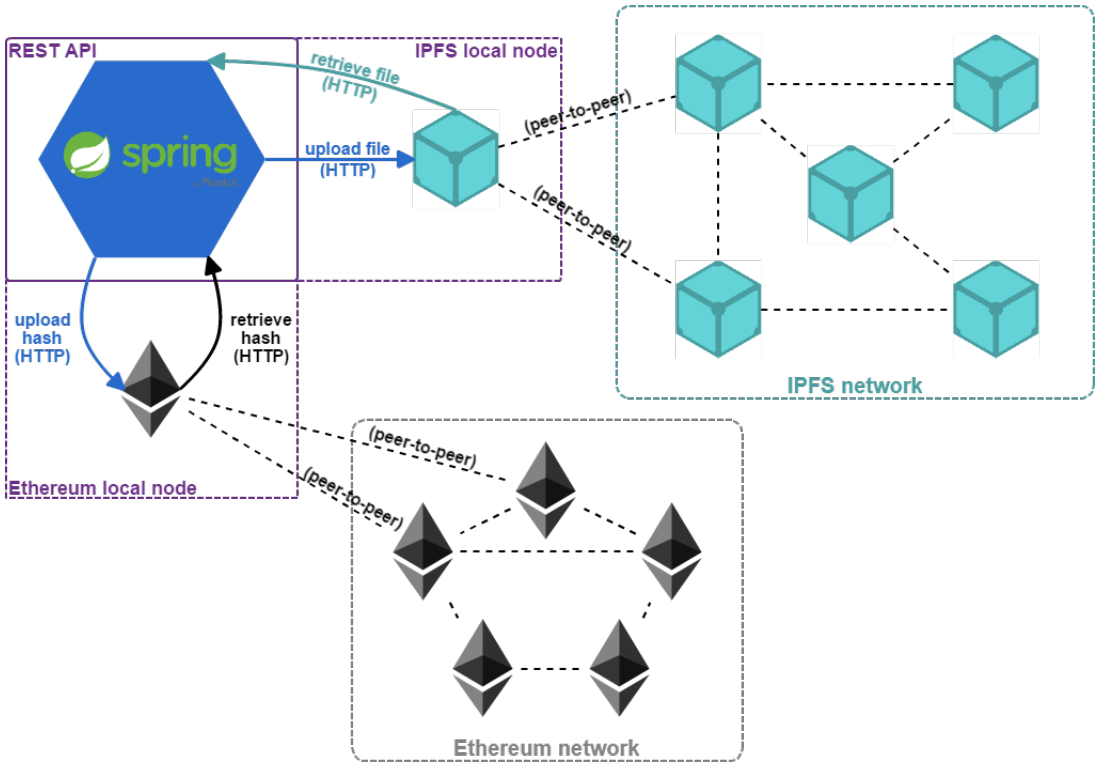

FIGURE 5.2 – Architecture envisagée

5.4 Gestion des données (Data View)

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

— Détail de l'API REST et de l'implémentation
— Ethereum smart-contract (déf en footnote) (avec Ethereum explications)
— IPFS storage
— Integration of Ethereum & IPFS

Décomposer les modules :
— Un module de gestion des données, utilisant IPFS (see IPFS white paper) ;
— 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 de versions permettant de conserver l'historique des mises à jour de la gTSL ;
— 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.

6.2 Présentation de la solution

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

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. 5, 43

[2] Blockgeeks. What is blockchain technology ?, 2017. 2, 43

[3] Vincent Bouckaert. *Design documentation - Global Trust Service Status List.* ARHS, 2B Rue Nicolas Bové, 1253 Luxembourg, 2017. 11, 12, 18, 43

[4] ETSI, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE. *Electronic Signatures and Infrastructures (ESI); Trusted Lists*, 2016. 9, 10, 13, 14, 18

[5] 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 :