

# Mémoire d'ingénieur

---

Implémentation d'une liste mondiale des services de  
confiance basée 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 3<sup>e</sup> 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 mondiale des services de confiance 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 27 juillet 2017**

**Signature :**



# Mémoire d'ingénieur

---

## Implémentation d'une liste mondiale des services de confiance basée 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](mailto: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](mailto: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 Arns 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, 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 trust list global, 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

<b>Remerciements</b>	<b>v</b>
<b>Avant-propos</b>	<b>vii</b>
<b>Table des matières</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Présentation de la technologie blockchain . . . . .	1
1.2 Définition du cadre et des objectifs du stage . . . . .	1
1.3 Mise en exergue du plan . . . . .	2
<b>2 Présentation du contexte</b>	<b>3</b>
2.1 L'entreprise Arns SpikeSeed . . . . .	3
2.2 Le contexte du projet . . . . .	3
<b>3 Présentation détaillée de la problématique</b>	<b>4</b>
3.1 Description du service de trust list global . . . . .	4
3.1.1 Exigences générales . . . . .	4
3.1.2 System Requirements . . . . .	5
3.1.3 Software Requirements . . . . .	9
3.2 Limites de l'architecture actuelle . . . . .	10
3.3 Objectifs et gestion de projet . . . . .	10
<b>4 État de l'art des solutions envisagées</b>	<b>11</b>
<b>5 Analyse du problème et solution élaborée</b>	<b>12</b>
<b>6 Réalisation, présentation et validation de la solution proposée</b>	<b>13</b>
<b>7 Résultats obtenus &amp; Perspectives</b>	<b>14</b>

<b>8 Conclusion</b>	<b>15</b>
<b>9 Exemples Listings</b>	<b>17</b>
<b>10 Autre chapitre</b>	<b>21</b>
10.1 Autre section . . . . .	21
10.1.1 Première sous-section . . . . .	21
<b>11 Conclusion</b>	<b>23</b>
<b>Bibliographie / Webographie</b>	<b>25</b>
<b>Liste des illustrations</b>	<b>27</b>
<b>Liste des tableaux</b>	<b>29</b>
<b>Listings</b>	<b>31</b>
<b>Glossaire</b>	<b>33</b>
<b>Annexes</b>	<b>36</b>
<b>A Première Annexe</b>	<b>37</b>
<b>B Seconde Annexe</b>	<b>39</b>
<b>Résumé</b>	<b>41</b>
<b>Abstract</b>	<b>41</b>

# 1 Introduction

La technologie blockchain s'est popularisée ces dernières années grâce à l'expansion de la crypto-monnaie Bitcoin [1] 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.

## 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. 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'immuabilité, 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.

## 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 Arqs 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. 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 de 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 établi 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 Arns Spikeseed**

Arns Spikeseed est une entité du groupe Arns 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. Comme toutes les autres entités du groupe, Arns 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, Arns 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 Le 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 du règlement européen. Ce projet de recherche et développement appelé FutureTrust rassemble un consortium de seize partenaires, dont Arns 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. À l'heure où l'ubérisation (déf en footnote) de notre société est en marche, la technologie blockchain apporte une approche nouvelle qui permet de se détacher de toute organe central ou tierce partie. Dans ce contexte, l'utilisation de la blockchain a été proposée dans le cadre du projet FutureTrust.

### 3.1 Description du service de trust list global

- Détail du sujet
- (design document 4. Requirements)
- (schéma page 15 ETSI 119 612 : Structure d'une TSL)

#### 3.1.1 Exigences générales

##### L'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 fournisseurs de services de confiance, nommés Trust Service Providers (TSP), 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.

*EU Member States and other European nations generally maintain lists of CAs and other TSPs in one or more nationally maintained registers.*

*EU Member States trusted list as defined in Regulation (EU) No 910/2014 include information related to the qualified trust service providers which are supervised by the competent Member State, together with information related to the qualified trust services provided by them, in accordance with the relevant provisions laid down in the Regulation.*



*Trusted lists are essential elements in building trust among electronic market operators by allowing users to determine the qualified status and the status history of trust service providers and their services. Under eIDAS Regulation, national trusted lists have a constitutive effect. In other words, a provider/service will be qualified only if it appears in the trusted lists. Consequently, the users (citizens, businesses or public administrations) will benefit from the legal effect associated with a given qualified trust service only if the latter is listed (as qualified) in the trusted lists.*

*The trusted lists of Member States include, as a minimum, information specified in Articles 1 and 2 of Commission Implementing Decision (EU) 2015/1505 profiling technical specifications defined in ETSI TS 119 612 v2.1.1.*

*Member States may include in the trusted lists information on non-qualified trust service providers and on other nationally defined trust services.*

*To allow access to the trusted lists of all Member States in an easy and trustworthy manner, the European Commission publishes a central list with links to the locations where the trusted lists are published as notified by Member States to the EC. This central list, called the List Of Trusted Lists (LOTL), is available as a signed or sealed XML machine-processable form (hereafter the LOTL) at the following URL :*

## **Parties prenantes**

Les acteurs principaux de la gTSL sont :

- 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 fournir les informations relatives aux fournisseurs de services de confiance 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 la but d’en améliorer sa résilience ainsi que sa gestion.

### **3.1.2 System Requirements**

#### **System Purpose**

Building upon the Trusted Lists standard defined in (ETSI TS 119 612, 2016), the gTSL aims at solving the current shortcomings of the Trusted List scheme when considered in a globalized context. Currently, the European Commission publishes a signed list of pointers (the European

List of the Lists - LoTL), each of which being a distribution point for a national list of trust service providers (TSP). Such national lists contain information on qualified and optionally non-qualified trust services providers and on the qualified and non-qualified trust services they provide. Under this scheme, changes performed in the content of a national list induce the need to re-publish the whole national list, while changes performed either on the URL at which the list is distributed, or either on the certificate used to sign it, induce the need to re-publish both the national list and the European list. The centralized nature of the Trusted Lists distribution scheme brings a number of potential issues that could emerge across all prospective stakeholders : -Member States TSLs being retrievable only based on the LoTL, the current scheme is subject to a single point of failure ; -Performance/latency issues can be encountered due to the need to download and validate information from various locations ; -The TSL requires all member states distributing nodes to be up and ; -There are no means for creating and maintaining application-specific TLs, as they are necessary for the e-Apostille application for example. Furthermore, the current TSL scheme suffers from a lack of history : amendments made to a TL are not recorded. As such, a new version of a TL completely overrides the previous one, making it possible to lose track of the changes implemented in between versions.

## **System Scope**

The scope of the gTSL is related to the definition of qualified trust services and trust service providers. As such, it will provide the functions necessary for the creation, update and distribution of trust service provider and trust service information.

## **System Overview**

In order to achieve its purposes, the gTSL will rely on two main open source components : a Global Trust Service Lifecycle Manager and a Global Trust Service Responder. Additionally, the gTSL will rely on an administration interface for daily management functionalities (e.g. trust information updates, certificates updates, etc.). These components and the interactions they have with the system's users are illustrated in Figure 3.1.

The purpose of the Global Trust Service Responder will be to allow external applications and users to query the gTSL for information related to the TSPs, in order to verify their status at a given point in time. It will therefore provide the functions necessary for answering to trust status information requests. The purpose of the Global Trust Service Lifecycle Manager is to facilitate the handling of the hierarchy of trust services, and to allow updating the status of each TSP over time. As such, it will provide the functions necessary for the creation, update and distribution of trust status information.

From an architectural perspective, the gTSL will rely on a 3-layer architecture : - The external services layer will expose the system's external interfaces, i.e. the Global Trust Service Responder and the administration interface ; - The business layer will be composed of the Global Trust List Service Lifecycle Manager ; - The data layer will correspond to the interfaces and components that allow connecting the gTSL to a data storage solution. As introduced in Section 4.2.1, one of the goals of the gTSL is to part from the current centralised distribution scheme and to adopt a decentralised approach. The recent emergence of block-chain concepts and the accompanying developments in block-chain based data storage solutions bring a set of potential solutions for this decentralisation goal. Section 5.4.2 introduces the different block-chain and block-chain data storage implementations that were considered, and describes the interfaces defined for the system

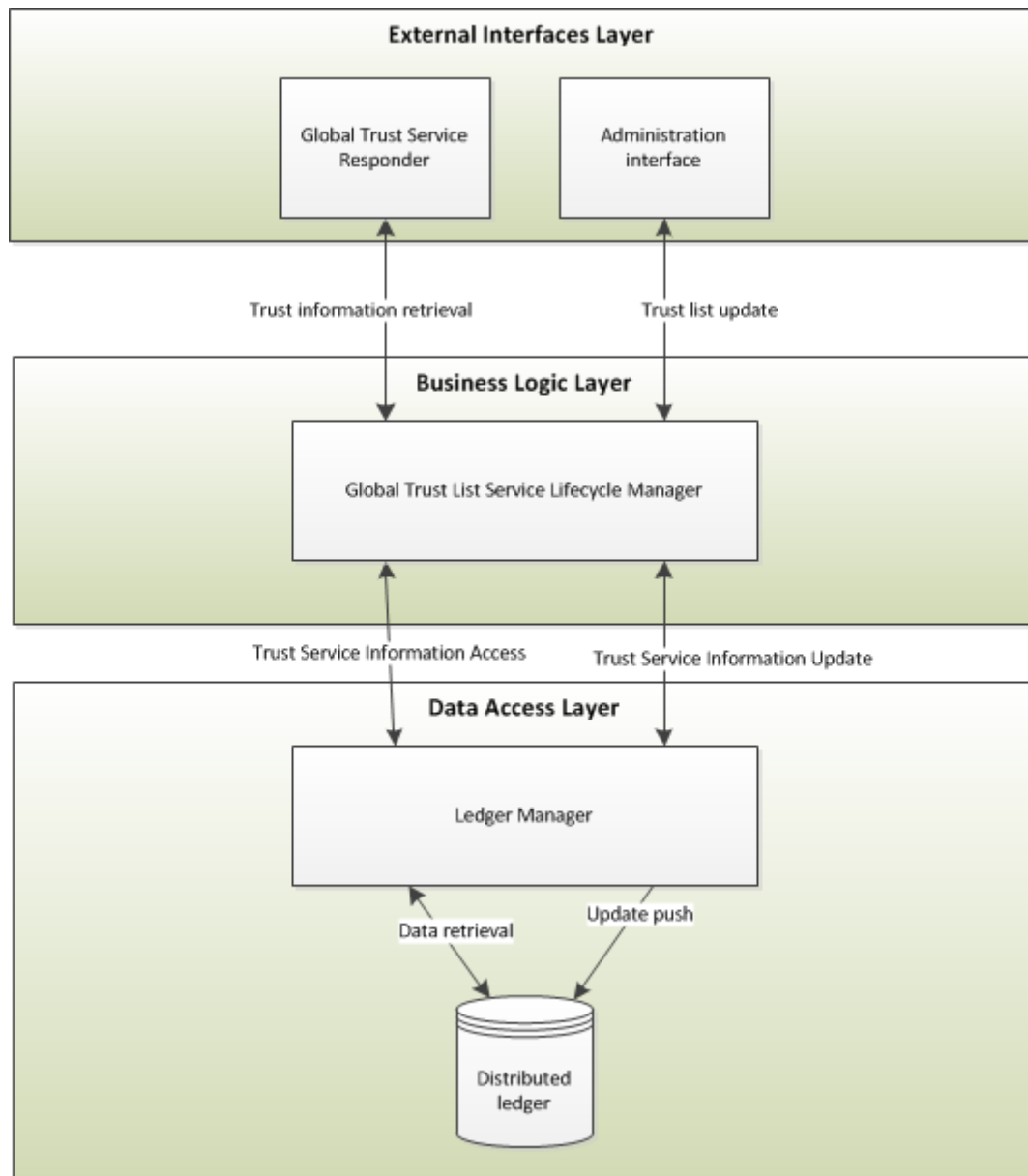


FIGURE 3.1 – gTSL – 3-Tier Architecture

on this layer.

## System Context

Figure 3.2 provides a high-level description of the system's interactions with external entities.

The External Users entity represents all external users that are seeking interactions with the system without specific privileges. The Comprehensive Validation Service developed as part of the Future Trust project is an example of such users. The Administrator Users entity represents all external users that are allowed to perform management operations on the gTSL, such as updates on Trust Service Providers' information.

## User Characteristics

Two different types of users have been identified for the gTSL : - Administrator users, i.e. platform administrators, which may act on behalf of an EU-Member State, in charge of the system's daily maintenance and management, e.g. publishing and updating trust status information ; - Ex-

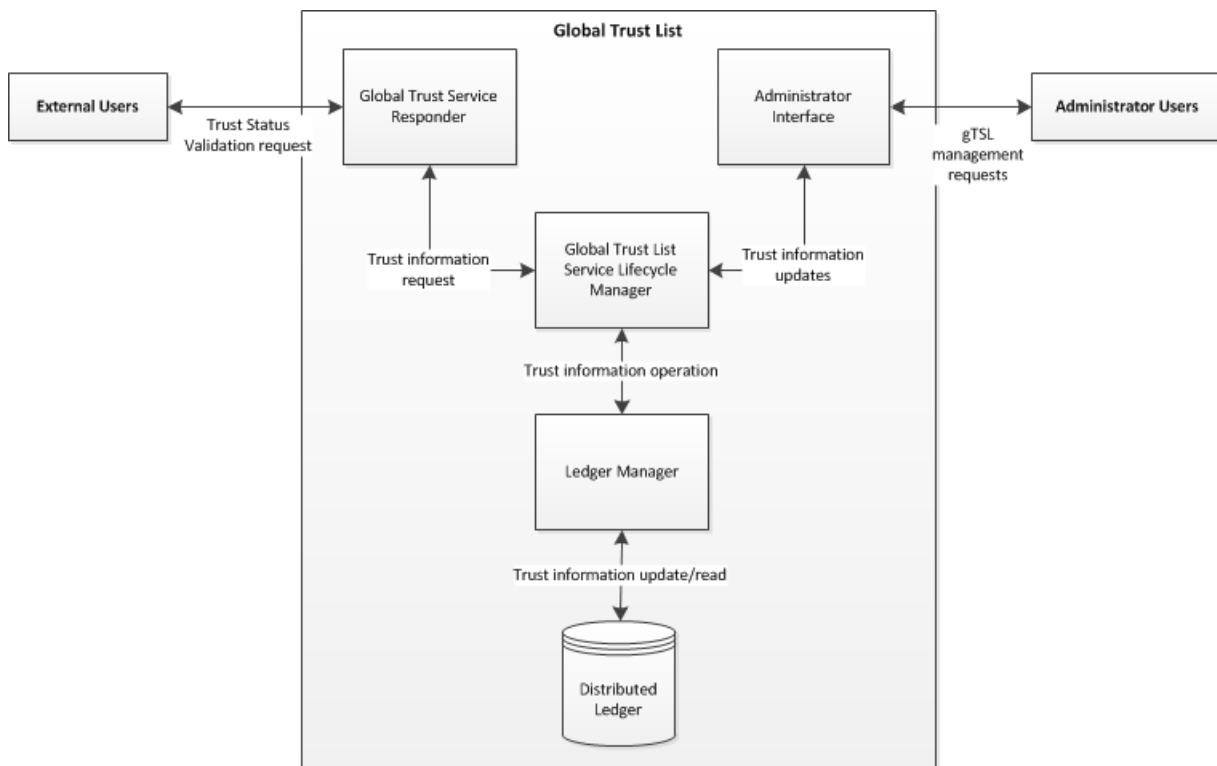


FIGURE 3.2 – gTSL System Context Diagram

ternal users, i.e. persons and remote applications willing to obtain trust status information for a given TSP, Member State, etc. These users will be offered access to the system through specific interfaces : - Administrator users will have access to a platform management interface, which will unambiguously expose the various administration functionalities to which the users must have access to; - External users will have access to both a graphical interface and a Web-Service interface, enabling trust status information retrieval based on provided electronic certificates as well as on general information regarding TSPs.

## System Functions and Functional Requirements

The following section provides the list of functional requirements identified for the gTSL. R1. The gTSL MUST allow the management of Trust Anchors and Meta-data of Identity Providers R2. The gTSL MUST support the international (non-EU) aspects of the eIDAS regulation As such, the gTSL must allow the inclusion of trust service providers from non-EU countries, whether qualified (i.e. when a reciprocity agreement is in place regarding qualified trust services) or not.

## Usability Requirements

The following section provides the list of usability requirements identified for the Global Trust Status List Service. R3. The gTSL MUST offer an interface allowing the retrieval as well as the submission of TSP information At minimal, to ensure compliance with (ETSI TS 119 612, 2016), the gTSL shall be available through HTTP/1.1, as defined in (RFC 2616, 1999).

## **Performance Requirements**

R4. The gTSL MUST include an efficient internal storage for storing status information on TSPs.  
R5. The gTSL MUST be highly scalable in order to handle large amounts of parallel requests efficiently.

## **System Interfaces**

R6. The gTSL SHALL expose, through a Web Services interface, the functionalities enabling the retrieval of trust service status information.

## **User Interfaces**

R7. The gTSL management features should be provided through an intuitive and coherent web interface, enabling its users to operate it unambiguously. As such, the user interfaces should remain consistent with the user interfaces of the current TL-Manager application<sup>1</sup> while showing no ambiguity in terms of visual hierarchy and content. R8. The gTSL web interfaces should enable users to work in a non-blocking manner. More specifically, the functionalities provided through this web interface should be asynchronous and should ensure a high responsiveness.

## **System Reliability**

R9. The gTSL SHALL be available on a 24 hours a day, 7 days a week basis More specifically, in order to comply with (ETSI TS 119 612, 2016), the gTSL Responder must be available on a 24 hours a day and 7 days a week basis, with an availability percentage of minimum 99.9% over one year periods. 4.2.9 System Security Due to the sensitive nature of the data managed by the gTSL, and to the high availability requirements, the security requirements must ensure that this data is not compromised and that the management of trust services and trust service providers is clearly restricted to authorised persons only. R10. The gTSL SHALL not allow unauthorised users to create, edit or delete trust service information or trust service provider information. R11. The gTSL SHALL ensure the integrity of the data it handles. 1 See <https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Ma>

### **3.1.3 Software Requirements**

R12. The Global Trust Status List Service MUST be highly scalable in order to handle large amounts of parallel requests. R13. The Global Trust Status List Service SHOULD not impose specific requirements for the runtime environment and SHOULD be deployable on any Open Source JEE Application Server.

## **Standard Compliance**

R14. The gTSL SHALL comply with standard (ETSI TS 119 612, 2016) As such, the gTSL shall comply with : - The format and semantics of a TL ; - The mechanisms to be used to support relying parties locating, accessing and authenticating TLs. R15. The gTSL SHALL be able to integrate with the existing LotL-based scheme This means that it is able to import the set of TLs referenced in the EU-LotL, highlight the changes as they occur and allow to add additional non-EU TSPs. R16. The

gTSL SHALL support the management of additional application-specific TLs This is important for supporting specific applications, such as the e-Apostille pilot for example.

## 3.2 Limites de l'architecture actuelle

Détailler l'architecture existante : "La finalité du stage est de d'effectuer une refonte de l'architecture actuelle qui a montré ses limites en y apportant des technologies innovantes."

décentralisation, pour éviter le single point of failure -> actuellement LOTL est le single point of failure

distribué, afin de maintenir les données dans l'ensemble du réseau -> actuellement chaque Member State maintient ses propres de données ce qui veut dire que si le noeud d'un Member State tombe, ses données ne sont plus disponibles

sécurité, intégrité des données + consensus -> actuellement, une partie des données est stockée dans une BD, l'autre partie sont des fichiers XML répartis sur plusieurs endpoints, difficile de vérifier l'intégrité des données de tous les endpoints en effet si un Member State est corrompu on ne peut pas le savoir

résilience, réseau publique avec +1M de noeuds jamais down -> actuellement l'architecture client-serveur repose entièrement sur la LOTL (qui agit comme un point central, donc à éviter) + chaque noeud est indépendant et se gère seul mais pb de résilience "globale" si un noeud est down

Enoncer clairement la problématique en une phrase : "Décentraliser le modèle actuel afin de résoudre les pbs de résilience, sécurité... grâce à un réseau décentralisé et distribué"

## 3.3 Objectifs et gestion de projet

- Objectifs - Gantt - Scrum description

## 4 État de l'art des solutions envisagées

Enoncer le principal désavantages de la blockchain qui est son coût (cf. Ethereum & IPFS integration).

Etat de l'art : Comparison document + Design document

## **5 Analyse du problème et solution élaborée**

Document de design de la gTSL (copy/paste) (design document 5. Software Architecture)



## **6 Réalisation, présentation et validation de la solution proposée**

Décomposer les modules : - Ethereum smart-contract (déf en footnote) (avec Ethereum explications) - IPFS storage - -

## **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.

```
1 void CEquation::IniParser()
2 {
3     if (!pP){ //if not already initialized ...
4         pP = new mu::Parser;
5
6         pP->DefineOpert("%", CEquation::Mod, 6); //deprecated
7         pP->DefineFun("mod", &CEquation::Mod, false);
8         pP->DefineOpert("&", AND, 1); //DEPRECATED
9         pP->DefineOpert("and", AND, 1);
10        pP->DefineOpert("|", OR, 1); //DEPRECATED
11        pP->DefineOpert("or", OR, 1);
12        pP->DefineOpert("xor", XOR, 1);
13        pP->DefineInfixOpert("!", NOT);
14        pP->DefineFun("floor", &CEquation::Floor, false);
15        pP->DefineFun("ceil", &CEquation::Ceil, false);
16        pP->DefineFun("abs", &CEquation::Abs, false);
17        pP->DefineFun("rand", &CEquation::Rand, false);
18        pP->DefineFun("tex", &CEquation::Tex, false);
19
20        pP->DefineVar("x", &XVar);
21        pP->DefineVar("y", &YVar);
22        pP->DefineVar("z", &ZVar);
23    }
24 }
```

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

```
1 void CEquation::IniParser()
2 {
3     if (!pP){ // if not already initialized ...
4         pP = new mu::Parser;
5
6         pP->DefineOpert("%", CEquation::Mod, 6); // deprecated
7         pP->DefineFun("mod", &CEquation::Mod, false);
8         pP->DefineOpert("&", AND, 1); //DEPRECATED
9         pP->DefineOpert("and", AND, 1);
10        pP->DefineOpert("|", OR, 1); //DEPRECATED
11        pP->DefineOpert("or", OR, 1);
12        pP->DefineOpert("xor", XOR, 1);
13        pP->DefineInfixOpert("!", NOT);
14        pP->DefineFun("floor", &CEquation::Floor, false);
15        pP->DefineFun("ceil", &CEquation::Ceil, false);
16        pP->DefineFun("abs", &CEquation::Abs, false);
17        pP->DefineFun("rand", &CEquation::Rand, false);
18        pP->DefineFun("tex", &CEquation::Tex, false);
19
20        pP->DefineVar("x", &XVar);
21        pP->DefineVar("y", &YVar);
22        pP->DefineVar("z", &ZVar);
23    }
24 }
```

Listing 9.2 – Affichage depuis le fichier source

De nombreux langages 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 ressemblant au listing 9.3. On y définit les différents mots-clés, ainsi que les délimiteurs des chaînes de caractère et des commentaires.

```
1 \lstdefinlanguage{amf}
2 {keywords=
3   {
4     xml,
5     amf,
6     volume,
7     material,
8     coordinates,
9     vertices,
10    vertex,
11    triangle,
12    x,
13    y,
14    z,
15    v1,
16    v2,
17    v3,
18    mesh,
19    object,
20    constellation,
21    metadata,
22    color,
23    texmap,
24    texture,
25    utex1,
26    utex2,
27    utex3,
28    instance,
29    deltax,
30    deltay,
31    deltaz,
32    r,
33    g,
34    b,
35    rx,
36    ry,
37    rz,
38    composite
39  },
40  sensitive=false,
41  morestring=[b]",
42  comment=[s]{<!--}{-->}
43 }
```

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 [?].



## **11 Conclusion**



## Bibliographie / Webographie

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



# Liste des illustrations

3.1	gTSL – 3-Tier Architecture . . . . .	7
3.2	gTSL System Context Diagram . . . . .	8
10.1	Logo de TELECOM Nancy . . . . .	21





## Liste des tableaux



# Listings

9.1	Premier Exemple . . . . .	17
9.2	Affichage depuis le fichier source . . . . .	18
9.3	Syntaxe définition d'un langage . . . . .	19



## **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, maiden-head 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 euismod 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 :**