Table des matières

Liste des tableaux

Liste des figures

1. Introduction

Le monde de l'informatique évolue beaucoup et très rapidement. Mais des tendances se dessinent et l'on peut déjà apercevoir de quoi pourra être fait le monde de demain. Les villes intelligentes, et par extension tout un tas d'objets connectés, sont en train de prendre place tout autour de nous. Des villes meilleures, qui nous promettent un avenir plus efficace, plus écologique et plus démocratique.

Mais pour répondre à ces nouveaux besoins, il faut de nouvelles technologies et des personnes s'attèlent en ce moment même à proposer les solutions de demain. D'autres cherchent à imposer leur méthode, espérant devenir la nouvelle référence pour tel ou tel domaine.

La fondation IOTA pourrait être l'un ou l'autre. Elle cherche non seulement à révolutionner le monde des objets connectés – l'*Internet of Things* (IoT) – mais aussi celui des cryptomonnaies. Le plus intéressant, c'est que cela semble fonctionner. Il ne se passe pas une semaine sans qu'un nouveau partenariat ne soit annoncé avec une grande entreprise : Volkswagen, Bosch ou même Microsoft pour n'en citer que quelques-unes.[1]

Pourquoi ? IOTA, leur cryptomonnaie, n'est pas basée sur la *blockchain.* Il n'y a pas de mineurs, pas de commissions et pas de baisse de performance lorsqu'elle est très sollicitée. Leur but ? Faire d'IOTA le « *backbone of IoT* ».

Il est vrai que tous ces objets auront besoin non seulement de communiquer et de se transmettre des informations, mais aussi de s'échanger de la valeur. Il faudrait donc que cela soit fait plus facilement : sans toutes les contraintes du système bancaire actuel.

1.1 Motivations

Ce travail est l'aboutissement de ma formation à la HEG. Pour moi, il était important de trouver une thématique qui soit à la fois intéressante, nouvelle et inédite.

Partant de là, étudier la thématique des *smart cities* à l'aide d'une cryptomonnaie complétement à contrepied de ce qui se fait actuellement me semble parfait.

[1] ARNOLD, Jake. IOTA Partner and Affiliations. *Untangled World* [en ligne]. 4 novembre 2017.

En effet, j'ai déjà travaillé sur la *blockchain,* et je suis convaincu qu'il s'agit d'une technologie révolutionnaire qui fera partie intégrante du monde de l'informatique de ces dix prochaines années. Mais cette dernière possède des défauts et ce n'est certainement pas une mauvaise idée que d'essayer de chercher de bonnes alternatives.

L'idée derrière ce travail est d'expérimenter ces alternatives, de découvrir de nouvelles technologies prometteuses tout en les contextualisant et en y apportant une critique constructive. D'où l'envie de combiner ce travail de recherche avec ce prototype.

> *« By using prototypes as a way to provoke and challenge the existing practice, it enables us to build a more coherent picture of the many implications – technical and social – in the development of specific design proposals and solutions. The point is not to underplay the importance of the final outcome, but to highlight the ability to understand broader aspects of the existing situation and the desires of the involved stakeholders towards developing a 'smarter' city, through the development of prototypes and by approaching the underlying design process as design-oriented research. »* [2]

1.2 Problématiques

À la fin de ce document, il sera possible de répondre à ces différentes questions :

- Est-il possible de procéder à un échange automatisé et décentralisé de ressources dans le monde de l'IoT ?
- Est-ce faisable avec IOTA ?
- De quoi sera fait le futur suite à ces recherches ?

1.3 Cadre

Le prototype sera développé sous la forme d'une application, réplicable sur différentes machines. Cette dernière permettra aux machines de vendre ou d'acheter des données entre elles, de manière automatisée et décentralisée.

Un cas d'utilisation imaginable pourrait être une application de mobilité, type application CFF ou TPG, qui permettrait la vente de données personnelles, comme la position, à un serveur. Le revenu servirait à acheter des tickets de transport à ce même serveur.

Ci-après sont exposés les différents concepts que j'ai abordé durant mes recherches. Cela commence par une contextualisation des grands domaines à étudier, suivi d'un chapitre dédié à la gestion des ressources et d'un autre dédié à la gestion économique.

Par la suite, j'aborderai le fonctionnement du prototype, les technologiques qu'il utilise et les points qu'il m'a permis d'éclairer. Je finirai par la conclusion, qui me permettra de répondre aux problématiques.

[2] KORSGAARD, Henrik et al. Prototyping a Smart City. [en ligne]. 2013.

2. Contexte

Le but de cette rubrique est de poser les différents concepts en lien avec ce travail. Définis et synthétisés, ils sont le fruit de mes recherches en vue du développement du prototype.

L'aspect technologique sera abordé plus loin dans le document, ci-dessous, je me focalise sur l'aspect conceptuel.

2.1 Internet of Things

2.1.1 Définition

L'IoT est un réseau de dispositifs physiques, intégrant de l'électronique, des capteurs, et une connectivité permettant à ces éléments de se connecter et d'échanger des données. Ainsi, une intégration plus directe du monde physique dans les systèmes informatiques est possible, ce qui se traduit par des améliorations d'efficacité, des avantages économiques et une réduction des efforts humains.[3]

L'idée étant de pouvoir connecter des objets qui ne sont pas, à la base, des ordinateurs, dans le but de les faire interagir entre eux ou avec des ordinateurs pour améliorer le traitement de l'information.

Un objet connecté basique ne ferait que des actions simples : capter une information, puis la transmettre à un ordinateur qui peut la traiter. Parfois, l'objet peut lui-même commencer à analyser l'information.

> *« Moi, le vélo en libre-service, étant donné que je suis immobile depuis plus de deux jours, ne dois-je pas être vérifié ? »*[4]

Un objet plus avancé pourrait prendre des décisions basées sur ce qu'il peut déduire de ses analyses.

> *« Moi, le chariot automatique, on me demande, au même moment, d'amener le produit x dans la zone A et le produit y dans la zone B. Étant donné l'endroit où je me trouve, je commence par le produit y dans la zone B. »*[4]

Finalement, un objet très avancé serait un objet qui embarque ou qui serait connecté à une intelligence artificielle. Il pourrait alors « comprendre » sa raison d'être et faire des analyses encore plus poussées.

> *« Moi, la palette, j'ai été déplacée dix fois de suite. Alors, sans avoir été programmée pour cela, je me donne pour objectif d'être contrôlée (support, houssage) et alerte les opérationnels concernés. Si on me rapporte une altération, je préviens les autres palettes et leur fais bénéficier de mon retour d'expérience. »*[4]

[3] Internet of things. *Wikipedia* [en ligne]. 29 juin 2018.
[4] GAUTIER, Philippe et al. *L'Internet des objets : Internet, mais en mieux*. 2011.

2.1.2 Fonctionnement

De manière générale, chaque objet connecté est muni d'un ou de plusieurs capteurs. Ceux-ci, petits et économes, transmettent des informations tirées de leur environnement direct à un actionneur. Ces informations peuvent être brutes ou en partie déjà analysées par l'objet.

L'actionneur peut prendre la forme d'un processeur intégré à l'objet, d'un serveur ou d'un smartphone auquel le capteur pourra accéder à distance. C'est ce dernier qui récoltera les données et transmettra des ordres au capteur.[5]

2.2 Smart cites

2.2.1 Définition

Une *smart city* est une ville utilisant différents types de capteurs de collecte de données afin de fournir des informations qui sont utilisées pour gérer efficacement les ressources.

Cela comprend les données recueillies auprès des citoyens, les dispositifs et les biens qui sont traités et analysés pour surveiller et gérer la circulation et les systèmes de transport, les réseaux d'approvisionnement en eau, l'application de la loi et bien d'autres services communautaires.

Cette ville utilisera donc justement des objets connectés afin d'optimiser l'efficacité des opérations et des services de la ville, mais aussi les technologies de l'information et de la communication (ICT) pour se connecter aux citoyens.

Une *smart city* permettra aux fonctionnaires municipaux d'interagir directement avec la communauté et l'infrastructure de la ville et de surveiller ce qui se passe dans la ville et la façon dont la ville évolue.[6]

2.2.2 Fonctionnement

Bien qu'au départ le mot *smart* signifiait « une ville plus informatisée, plus connectée », aujourd'hui on entend cette appellation à la fois pour le côté de l'innovation technologique, mais également pour une ville plus intelligente au sens propre, plus fonctionnelle, et de ce fait plus durable et plus agréable à vivre, conséquences directes de cette digitalisation.[7]

[5] ROCHAS, Audrey. *SmartLife : vivre avec les objets connectés*. 2016.
[6] Smart city. *Wikipedia* [en ligne]. 28 juin 2018.
[7] MEOLA, Andrew. How smart cities & IoT will change our communities. *Business Insider* [en ligne]. 20 décembre 2016.

On a tendance à identifier six grands axes pour le développement et la classification des s*mart cites*[8] :

- Smart Economy
    - Digitalisation des produits et des services, économie de partage
- Smart Mobility
    - Transports (publics ou privés) optimisés, écologiques et fluides
- Smart Environement
    - Développement d'énergies renouvelables, optimisation des ressources
- Smart People
    - Égalité des chances (travail, formation, personnes à mobilité réduite etc.)
- Smart Living
    - Qualité de vie, sécurité, cohésion sociale
- Smart Governance
    - Digitalisation de l'administration, démarches participatives, open data

> *« We believe a city to be smart when investments in human and social capital and traditional (transport) and modern (ICT) communication infrastructure fuel sustainable economic growth and a high quality of life, with a wise management of natural resources, through participatory governance. »*[8]

Ce qui ressort systématiquement lorsque l'on étudie la question des *smart cites*, c'est l'importance que prendra le consensus entre les différents acteurs pour la gestion de la communauté. On voit naitre une foule de nouvelles démarches participatives et d'actions de concertations de la part du gouvernement.

C'est toute la ville qui se met d'accord sur la manière dont il faut organiser la vie quotidienne[9] :

- Le gouvernement
- Les citoyens
- Les entreprises publiques et privées
- Les organisations internationales
- Les universités et les centres de recherche

[8] CARAGLIU, Andrea et al. Smart Cities in Europe. *Journal of Urban Technology*. 1 avril 2011.

[9] ARCORACI, Krystal. *Développer le concept de Smart Canton de Genève* [en ligne]. 2017.

2.3 Systèmes distribués

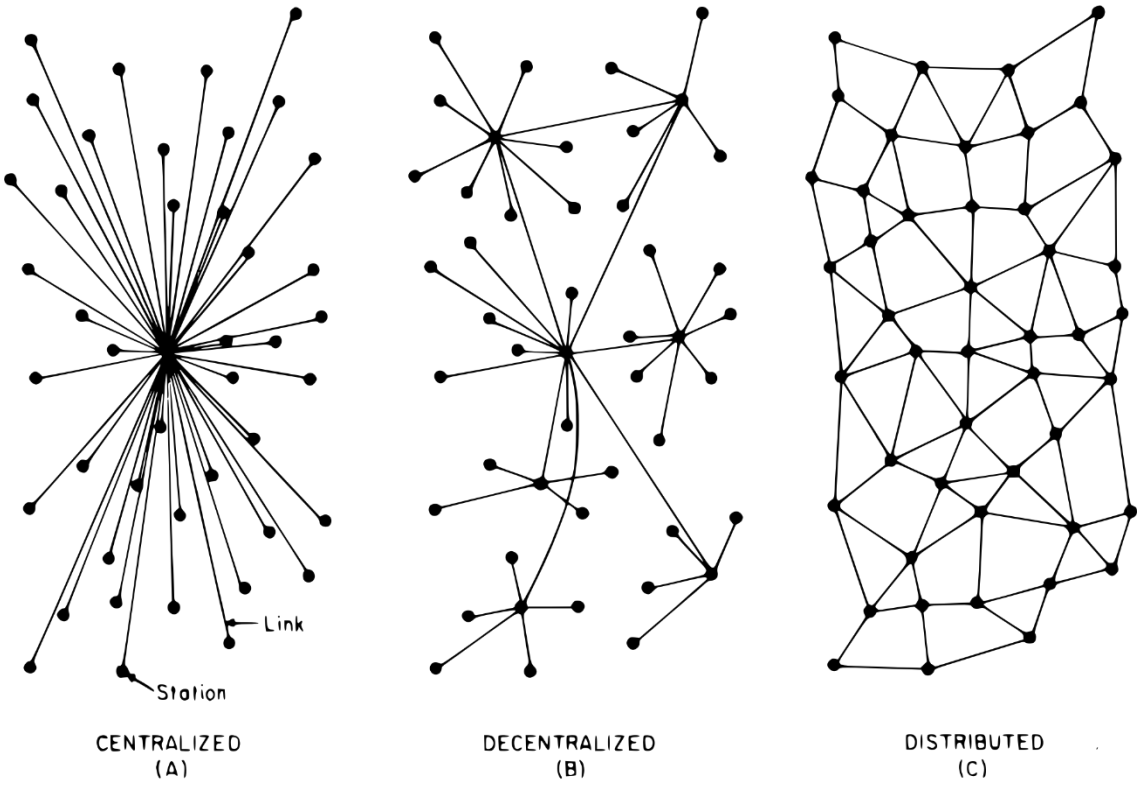
2.3.1 Définition

Ces dernières années ont vu émerger un fort intérêt et un fort investissement pour des solutions distribués ou décentralisés. Le concept de ce genre de système est d'éviter de stocker toutes les ressources au même endroit ou sur la même machine.[10]

Au contraire, les composants et les ressources sont situés sur différents ordinateurs répartis sur le réseau, les *nœuds,* qui communiquent et coordonnent leurs actions en se transmettant des messages.[11]

Néanmoins, les termes « distribué » et « décentralisé » ne sont pas équivalents. Un système décentralisé possède quand même des nœuds centraux, tandis que dans un système distribué chaque machine est l'égale des autres.[12]

Figure 1 : Vue d'ensemble des trois types d'architectures[13]



[10] SONM. Decentralized Computing on Ethereum Smart - Contracts is Gaining Momentum. *SONM* [en ligne]. 26 avril 2017.
[11] Decentralized computing. *Wikipedia* [en ligne]. 17 mai 2018.
[12] GOYAL, Saurabh. Centralized vs Decentralized vs Distributed. *Delta Exchange* [en ligne]. 1 juillet 2015.
[13] BARAN, Paul. *Centralized Decentralized and Distributed Networks* [en ligne]. 1964.

2.3.1.1 Exemples

Voici quelques exemples de systèmes centralisés, décentralisés et distribués :

Tableau 1 : Exemples de systèmes centralisés, décentralisés et distribués

Système	*Centralisé*	*Décentralisé*	*Distribué*
Réseaux	Wi-Fi domestique	Internet, Réseau cellulaire	-
Base de données	SQL Server	-	Cassandra, MongoDB
Fichiers	SMB	World Wide Web	IPFS, BitTorrent
Messagerie	Forums internet	Email, Matrix	Ricochet
Registres	Syslog	-	Blockchain, Tangle

2.3.2 Propriétés

Mon travail se focalisant sur les systèmes distribués, je vais uniquement aborder les propriétés et les limites de ces derniers, mais la plupart s'appliquent également aux systèmes décentralisés.

> *« A distributed system is a collection of independent computers that appears to its users as a single coherent system. »*[14]

Différentes raisons peuvent pousser à implémenter une solution distribuée. Parfois, c'est la nature même d'une application qui impose l'utilisation d'un réseau distribué, lorsque par exemple des données sont produites à un endroit mais requises à un autre endroit.[15]

D'un autre côté, il existe de nombreux cas dans lesquels l'utilisation d'un seul ordinateur serait possible en principe, mais où l'utilisation d'un système distribué est bénéfique pour, entre autres, les quelques raisons détaillées si dessous.[15]

2.3.2.1 Performance

Souvent, il peut être plus rentable d'obtenir le niveau de performance désiré en utilisant un *cluster* de plusieurs ordinateurs bas de gamme, en comparaison avec un seul ordinateur haut de gamme.

[14] TANENBAUM, Andrew S. et al. *Distributed systems: principles and paradigms*. 2007.
[15] Distributed computing. *Wikipedia* [en ligne]. 27 juin 2018.

Dans une architecture de type client à serveur centralisé, plus il y a de clients, moins le système sera performant. À l'inverse, dans une architecture distribuée, plus il y a de composants du système et plus le système sera performant (on le dit *scalable*).

2.3.2.2 Fiabilité et redondance

Un système distribué est plus résilient qu'un système non distribué, car il n'a pas de point de défaillance unique (*single point of failure*). En effet, il est extrêmement redondant et ne sera pas affecté par la perte de l'un de ses composants.

Par ailleurs, un système distribué qui doit stocker des données peut très facilement rejeter des données qui seraient corrompues et est peu affecté par ce problème.

2.3.2.3 Indépendance

Un système distribué n'a pas de nœud central, par conséquent, il ne peut pas appartenir ou être contrôlé par une quelconque autorité. Le système est indépendant de tout contrôle, si ce n'est celui exercé par consensus de la part de ses composants.

2.3.2.4 Collaboration

La collaboration des différents nœuds qui forment le système amène à une répartition des charges (coûts, infrastructures, électricité, puissance de calcul…) qui auraient toutes dû être assumées par un seul acteur dans le cas d'un système centralisé.

2.3.3 Limites

Les solutions distribuées ont cependant un coût et de nouveaux problèmes à résoudre. Ces derniers sont essentiellement dus à l'éloignement géographique des machines, mais aussi à leur nombre important.[16]

2.3.3.1 Réplication et conflits

Dans le cas d'un registre ou d'une base de données qui est nécessaire au bon fonctionnement du système, ceux-ci doivent être répliqués sur tous les nœuds. Par la nature même d'un système distribué, ce processus prendra forcément un certain temps et sera extrêmement sensible à une éventuelle saturation du réseau.

Par conséquent, il y a un risque de données obsolètes et de conflits qui doivent pouvoir être gérés par le système.

[16] TEOTIA, Siddharth et al. How does centralized and decentralized computing differ? *Quora* [en ligne]. 16 juin 2016.

2.3.3.2 Consensus

Si le système est ouvert et qu'un nœud malveillant ou défaillant ne respecte pas l'homogénéité ou la cohérence du système, il doit pouvoir être écarté par le système.

C'est pareil s'il faut prendre une décision à un moment ou un autre, le système doit être capable d'arriver lui-même à établir un consensus.

2.3.3.3 Mises à jour

Si le ou les programmes exécutés doivent être mis à jour, il faut prendre en compte le fait que la mise à jour ne pourra pas être faite simultanément sur toutes les machines.

Les différentes versions doivent donc pouvoir être compatibles entre elles, du moins un certain temps.

2.3.3.4 Gestion du temps

Beaucoup d'opérations nécessitent un timing ou la connaissance d'une heure précise (rien que pour les *timestamps* par exemple). Il faut donc que tout le système soit synchronisé et utilise la même horloge pour éviter toutes aberrations dans la gestion de l'heure.

Par ailleurs, le système se reposant sur un réseau, il y aura forcément un certain temps de latence à chaque communication entre les nœuds. Celui-ci doit être pris en compte et géré par le système.

3. Gestion des ressources

La gestion des ressources pose des enjeux de taille, que ce soit au niveau de la gouvernance ou de la collecte des données. Ci-après, j'approfondis ces deux domaines, dans le but de comprendre le fonctionnement profond de l'IoT et des *smart cities* et ainsi pouvoir répondre plus précisément aux problématiques.

3.1 Collecte des données

Nous avons vu que l'IoT, et par extension les *smart cites*, basent leur fonctionnement sur des capteurs. Ces derniers récoltent et transmettent toutes sortes de données pour permettre leur analyse.

L'énorme quantité de données et d'informations générés par ces objets connectés et leurs capteurs dépassent la capacité d'analyse humaine et entrent dans ce que l'on appelle le *big data*.

3.1.1 Big Data

Le *big data*, c'est toutes ces données qui permettent de mieux analyser les comportements et l'environnement, mais dont la quantité et l'hétérogénéité est trop importante.

Partant de là, le défi de la collecte des données réside dans la capacité à pouvoir analyser ces données efficacement et à moindre coût.

L'exploitation de ces données doit forcément passer par de nouvelles méthodes de rationalisation telles que l'agrégation de données[17], mais aussi de nouvelles méthodes d'analyse de données comme le *datamining*.[18]

C'est grâce au *big data* que certaines entreprises développent un monde plus intelligent, où la captation et l'analyse des données va permettre d'optimiser les processus, de mieux gérer les appareils et donc de gagner du temps en faisant des économies d'énergie et de coût.[19]

C'est un cercle vertueux, l'analyse des *big data* permet de dégager des tendances, d'identifier des besoins pour finalement proposer de nouvelles applications et services encore plus performants.

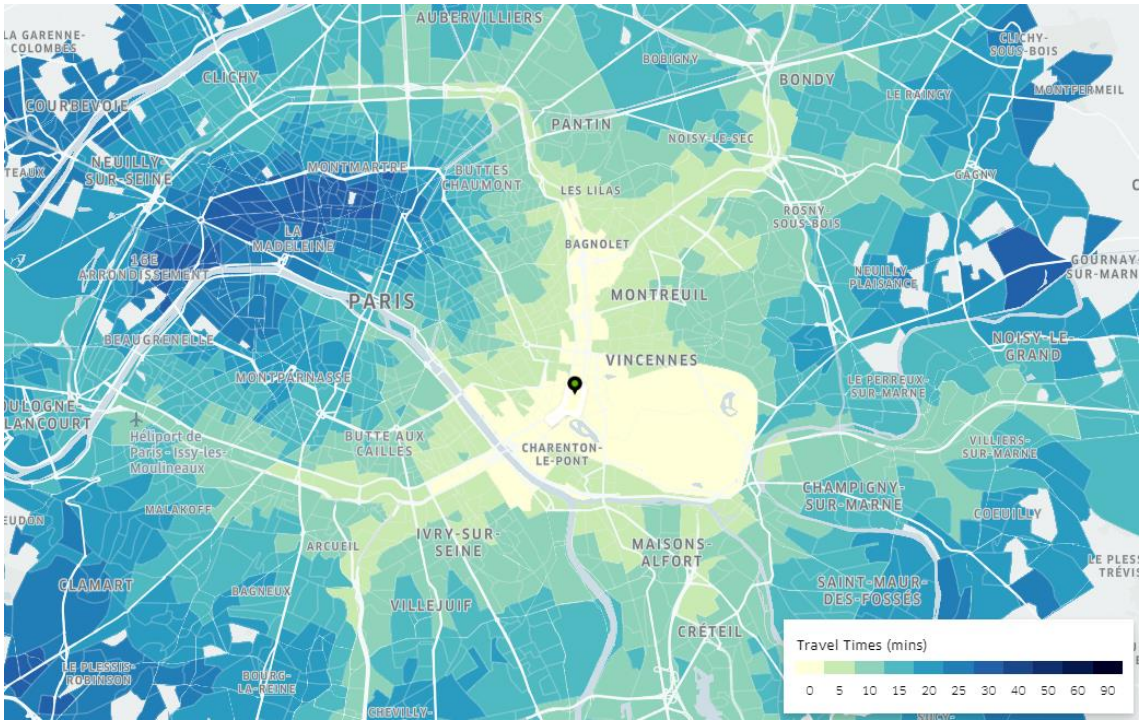
[17] MATTA, Natalie. *Vers une gestion décentralisée des données des réseaux de capteurs dans le contexte des smart grids* [en ligne]. 20 mars 2014.
[18] FERNÁNDEZ-ARES, A. et al. Studying real traffic and mobility scenarios for a Smart City using a new monitoring and tracking system. *Future Generation Computer Systems*. 1 novembre 2017.
[19] ROCHAS, Audrey. *SmartLife : vivre avec les objets connectés*. 2016.

3.1.2 Exemple

Un exemple pratique de ces *big data* en phase d'amélioration continue est le cas d'*Uber Movement*. En effet, les véhicules Uber sont tous équipés d'un GPS et d'une horloge, il est donc possible de récupérer les données de ces courses, de les anonymiser et de les agréger afin de mettre à disposition cette énorme quantité de données, brutes ou en partie analysés, aux citoyens et autorités de planification de la ville.[20]

De cette manière, Uber peut mieux connaître la ville, planifier au mieux ses déplacements, tout en partageant ces données aux autorités. Ces dernières pourront ensuite les utiliser pour résoudre des problèmes dans le but d'améliorer la fluidité du trafic, et par conséquent… la ponctualité des courses d'Uber.

Figure 2 : Exemple de cartographie *Uber Movement*[21]



Temps moyen d'une course *Uber* au départ de Porte Dorée, à Paris.
Sur cette image, on remarque facilement l'efficacité des voies rapides et du périphérique.

[20] UBER. Introducing UBER Movement | UBER. *YouTube* [en ligne]. 9 janvier 2017.
[21] UBER TECHNOLOGIES INC. Uber Movement: Let's find smarter ways forward. [en ligne].

3.1.3 Hétérogénéité

Une autre difficulté à laquelle fait face l'IoT, c'est l'hétérogénéité, que ce soit au niveau fonctionnel (différentes alimentations, connectiques, capteurs ou actionneurs), ou au niveau technologique (différents systèmes d'exploitation, interfaces, technologies propriétaires ou format des données).[22]

Toutes ces incompatibilités limitent les interactions naturelles entre les appareils IoT et leurs terminaux.

Pour réduire l'hétérogénéité fonctionnelle, des couches logicielles supplémentaires peuvent être développées, celles-ci devant en priorité utiliser des formats ouverts et interopérables.

Quant à l'hétérogénéité technique, elle peut être réduite en conduisant l'effort de standardisation entamé par les différent organismes (IEEE, W3C, …) et en utilisant des architectures faiblement couplées et fortement cohésives telle que de SOA.[23]

3.1.4 Sécurité et vie privée

La sécurité et le respect de la vie privée est un grand défi dans le monde de l'IoT. Les objets connectés, de par leur nature, peuvent être très sensible aux attaques.[22]

Ces derniers étant directement impliqués dans le monde physique, ils sont donc physiquement accessibles et pourraient être trafiqués manuellement. Par ailleurs, cette présence dans le monde réel peut rendre certains objets particulièrement dangereux, que ce soit des voitures autonomes ou même des capteurs destinés à faire fonctionner une centrale nucléaire.

En outre, les capacités de calcul encore relativement faibles de ces objets les rendent un peu plus vulnérables aux attaques à distance, certains ne pouvant pas utiliser d'algorithmes de chiffrement forts, et d'autres faisant face au risque de se faire intercepter leurs clés de chiffrement par des attaques de type *man in the middle*.

Finalement, la forte utilisation du sans-fil rend ces objets vulnérables à l'écoute non autorisée et aux attaques par déni de service à l'aide de brouilleurs.

[22] BILLET, Benjamin. *Système de gestion de flux pour l'Internet des objets intelligents* [en ligne]. 19 mars 2015.
[23] Architecture orientée services. *Wikipédia* [en ligne]. 31 mai 2018.

En ce qui concerne la vie privée, les risques de dérives sont grands. La plupart des objets connectés d'aujourd'hui fonctionnent en récoltant des données sur des individus. Certaines de ces données peuvent être sensibles. Il est alors très important de les identifier afin de les stocker sur des serveurs séparés et de manière sécurisée.

Par ailleurs, rien n'empêche certains objets de récolter des données sur des individus qui n'utilisent pas le service proposé par l'objet et qui n'ont pas accepté cette récolte. Or, il est aujourd'hui possible, en croisant les informations de divers objets et de divers capteurs, d'établir des profils très complets sur ces individus.

Il est alors primordial qu'un cadre légal soit établit pour empêcher ce genre de pratique. L'arrivée du RGPD est parfaite pour cela, car il garantit que l'on ne puisse pas stocker d'informations sur les individus qui n'ont pas explicitement donné leur accord, en plus de donner le droit de regard, de rectification et de suppression aux utilisateurs de ces services.[24]

3.2 Gouvernance

Il est clair que la réussite d'une *smart city* passe par une bonne gouvernance.

Aujourd'hui, une *smart city* est la résultante d'initiatives commerciales, le plus souvent menées par des start-ups, d'initiatives étatiques et d'initiatives citoyennes, en général participatives. Mais on constate qu'il y a rarement une coordination entre ces différentes initiatives.[25]

Pourtant, pour que ces projets restent cohérents, il y a besoin d'une coordination de toutes ces nouveautés, et actuellement c'est l'état qui cherche à se positionner sur ce sujet.

Mais une *smart city* devrait être une ville construite autour et pour ses citoyens, dans le but d'améliorer la qualité de vie. Elle ne peut découler d'un plan directeur qui planifierait tout de bout en bout, sans interroger les principaux intéressés.

Ce sont ces mêmes citoyens, que ce soit par des démarches commerciales ou bénévoles, qui vont tâcher de rendre une ville plus *smart*.[26]

[24] *Règlement général sur la protection des données* [en ligne]. 4 mai 2016.
[25] EECKHOUT, Laetitia Van. « Le concept de "Smart City" n'est plus opérant ». *Le Monde* [en ligne]. 25 avril 2018.
[26] MAGNE, Aleksi. Vers une approche de smart cities participatives. *Les Echos* [en ligne]. 12 juin 2017.

C'est un changement de paradigme, de la même manière que le citoyen ou le consommateur a gagné en pouvoir depuis l'arrivée d'internet, celui-ci participe activement à la formation de sa ville, c'est l'acteur qui est au centre.

3.2.1 Communs

On voit justement réapparaitre la notion de communs, notion presque disparue depuis la révolution industrielle. On pourrait regrouper sous cette appellation les ressources partagées, les communautés de toutes sortes et les projets collaboratifs, trois aspects qui construisent le capital social de notre société.[27]

Il y a ces plateformes participatives accessibles par tous (Github, Wikipedia), ces technologies qui fonctionnent par la mise en commun de ressources (BitTorrent, Blockchain), ces initiatives qui tendent à remettre le citoyen au cœur de la gestion de la vie courante, parfois citoyennes (Monnaie Léman), parfois étatiques (Genève Lab).

Cette évolution commence même à se faire une place dans le monde de l'entreprise, certaines choisissant de mettre en place des ressources communes, et d'autres, une organisation horizontale libre de toute hiérarchie.[28]

L'essayiste J. Rifkin – un personnage plutôt controversé[29] – théorise que l'économie de partage est gentiment en train d'éclipser l'économie de l'échange, et que ce sont des citoyens à la fois producteurs et consommateurs qui peupleront le monde de demain.[30]

Je pense en effet, sans tout le pathos qu'il y met, que c'est la direction que prend la société. Le gouvernement va successivement contrôler, puis fédérer, et après simplement coordonner ces initiatives, à chaque fois en perdant un peu plus de pouvoir au profit de ces citoyens.

Il finira par avoir un rôle d'agrégateur, ou simplement d'intermédiaire, entre tous ces projets et tous ces communs. Les décisions, elles, finiront par être prises ailleurs, dans un système plus fiable, plus juste, plus *smart*, possible grâce aux registres distribués et aux nouvelles technologies.[31]

---

[27] Communs. *Wikipédia* [en ligne]. 27 juin 2018.
[28] SLADE, Samantha. The future is in business as commons. *TEDx Geneva* [en ligne]. 22 mars 2018.
[29] BAQUIAST, Jean-Paul. Jeremy Rifkin. La nouvelle société du coût marginal zéro. *Club de Mediapart* [en ligne]. 7 octobre 2014.
[30] RIFKIN, Jeremy et al. *La nouvelle société du coût marginal zéro : L'internet des objets, l'émergence des communaux collaboratifs et l'éclipse du capitalisme*. 2014.
[31] VERDON, Antoine. Pourquoi la blockchain augure la fin de l'Etat-nation. *Le Temps* [en ligne]. 11 septembre 2016.

3.2.2 Paradoxe

Ce que j'ai trouvé étrange en faisant mes recherches, c'est une sorte de double utilité à l'IoT qui me parait contradictoire. D'un côté, il peut servir à fonder une *smart city* plus équitable et plus participative.

De l'autre, il se résume aujourd'hui principalement à des entreprises qui cherchent à optimiser au maximum leur rendements, ou même à exploiter les données personnelles de leurs clients.[32]

On l'utilise principalement pour vendre, et avec tout ce dont sont capables les algorithmes et les intelligences artificielles aujourd'hui, si l'on peut contrôler les choix d'une personne en lui montrant une publicité adaptée, on peut probablement contrôler bien d'autres aspects de sa vie.

C'est là pour moi le paradoxe qui existe entre ces deux faces d'une même pièce, l'IoT commercial et la *smart city* participative.

[32] ROCHAS, Audrey. *SmartLife : vivre avec les objets connectés*. 2016.

4. Gestion économique

La gestion économique est un domaine à part dans un monde automatisé et décentralisé rempli de cryptomonnaies. Ci-après, j'explique le fonctionnement, les avantages et les limites des registres distribués que j'ai étudiés, soit la *blockchain*, mais surtout, le Tangle.

4.1 Registres distribués

Un registre distribué (*distributed ledger*) est l'utilisation et le stockage d'un registre – une liste d'enregistrements ajoutés les uns après les autres – implémenté dans une architecture distribuée, sous forme de réseau *peer-to-peer*.[33]

Dans le but de garantir l'intégrité (pas de modification ultérieure) et la cohérence (pas d'ajouts incohérents) des enregistrements, des algorithmes de consensus font partie intégrante du système.

Les registres distribués sont principalement connus de par leur utilisation par les cryptomonnaies, mais peuvent aussi servir à stocker des données ou à exécuter des petits programmes informatiques, les *smart contracts*.

Bien que la gestion d'une monnaie, le stockage de données ou l'exécution de programmes sont possibles sans l'intervention de systèmes distribués, ceux-ci sont de plus en plus utilisés car les propriétés fondamentales d'un registre distribué sont très intéressantes et ont un certain potentiel émancipateur.

Un registre distribué peut être implémenté de différentes manières, mais la plus reconnue et la plus rependue reste la *blockchain*.

4.2 Cryptomonnaies

Les cryptomonnaies sont des monnaies alternatives qui sont utilisables au travers d'un registre distribué. Ces dernières se basent sur les principes de la cryptographie et intègrent l'utilisateur dans le processus de règlement des transactions et parfois dans le processus d'émission.[34]

Le réseau étant décentralisé, la monnaie est indépendante de toute autorité, elle est auto-régulée. Les échanges se font d'une adresse à une autre. Ils sont instantanés et sans intermédiaire.

[33] Distributed ledger. *Wikipedia* [en ligne]. 18 juin 2018.
[34] Cryptomonnaie. *Wikipédia* [en ligne]. 25 juin 2018.

Les unités monétaires, les *tokens*, n'existent pas. C'est le registre distribué, en faisant office de grand livre de compte, qui retrace toutes les entrés et les sorties, toutes les transactions. En remontant le registre dans l'ordre antéchronologique, on peut connaître par de simples additions et soustractions le solde d'une adresse donnée.

Lorsque l'on utilise un portefeuille, un *wallet*, on détient une simple paire de clés publique / privée standards, couramment utilisés en cryptographie asymétrique. Ces clés permettent au détenteur d'être authentifié : si une transaction est signée par sa clé privée, c'est que la transaction vient de lui, ceci étant vérifiable par le reste du réseau grâce à sa clé publique, qui fait aussi office d'adresse.

Chaque nouvelle transaction ajoutée au registre référence directement ou indirectement les précédentes. Toute tentative de modification d'anciennes transactions est donc immédiatement repérée car elle rompt la cohérence dans le registre et cette nouvelle version sera rejeté par le réseau. Les transactions sont donc irréversibles.

4.2.1 Sécurité et consensus

Par conséquent, s'il on veut tenter de falsifier la monnaie, il faut falsifier ce registre distribué. Or les algorithmes de consensus s'assurent que l'ajout de transactions au registre ne peut être fait que sous certaines conditions (travail nécessaire, actifs bloqués, vérifications), ce qui limite grandement la capacité à émettre beaucoup de transactions en peu de temps.

Même en admettant qu'un individu malveillant arrive, par chance, à ajouter des transactions considérées comme valides, cela lui aura pris tellement de temps et tellement de ressources que son registre se verra forcément dépassé en taille par le vrai registre, les autres nœuds préférant le registre le plus long / récent.

Le seul moyen possible pour un attaquant de réussir à compromettre le registre serait de contrôler plus de 50% des nœuds, ce qui est réputé très difficile et improbable, mais pas impossible.

4.2.2 Cryptoactifs

Les cryptomonnaies ont avant tout été conçues pour être utilisables dans des échanges monétaires sans tiers, en se passant du système bancaire actuel.

Mais ces dernières n'ont souvent pas cours légal et l'absence d'autorité de régulation provoque de fortes variations de leurs valeurs, au point qu'elles sont qualifiées d'extrêmement volatiles et de non fiables. Ces propriétés limitent donc grandement leur adoption pour les échanges courants.

Certaines banques[35] et autorités préfèrent même parler de cryptoactifs, réduisant ces monnaies à de simples actifs volatils sur lesquels il est possible de spéculer. Ce nouveau « marché » des « cryptoactifs » est dérisoire quand on pense que le but initial était de partir à contrepied du système bancaire.

4.2.3 Transparence et anonymat

Le registre étant accessible librement sur Internet, il est possible de connaître toutes les transactions qui ont eu lieu depuis la création de la monnaie, leurs sources, leurs destinataires et leurs montants.

Par conséquent, le solde de tous les utilisateurs de la monnaie est connu et publiquement accessible.

En revanche, les utilisateurs sont identifiés par leur clé cryptographique publique, leur adresse. On ne peut pas faire de lien entre l'adresse et l'identité réelle, les utilisateurs sont donc presque anonymes.

Presque, car les plateformes d'échanges qui permettent d'acheter des cryptomonnaies et les autorités imposent souvent la présentation d'une pièce d'identité pour procéder à un achat, afin de tracer les fonds et de s'assurer qu'ils ne financent pas d'activités criminelles ou servent à blanchir de l'argent.

4.2.4 Confiance

Comme dit précédemment, des algorithmes de consensus tournent sur ces systèmes distribués pour permettre de valider les enregistrements ou transactions.

Ces algorithmes qui mêlent cryptographie et apports d'une preuve de travail garantissent que les transactions précédentes ne peuvent être modifiés et que les nouvelles transactions ne puissent pas contredire celles qui existent déjà, soit pour une cryptomonnaie que l'on ne puisse ni créer de l'agent fictif, ni dépenser deux fois la même somme.

Cette garantie de confiance permet de se passer des tiers habituels lors de transactions, comme des banques ou des notaires, et c'est là que se trouve le vrai potentiel des registres distribués.[36]

[35] L'émergence du bitcoin et autres crypto-actifs : enjeux, risques et perspectives. *Banque de France* [en ligne]. 5 mars 2018.

[36] QUEISSER DE STOCKALPER, Derek. La blockchain comme contrat social de la 4e révolution industrielle. *Le Temps* [en ligne]. 27 novembre 2017.

Mais malgré tout, pour certaines personnes, le problème de la confiance à simplement dérivé de ces tiers au développeur qui a écrit le code, à qui il faut toujours faire confiance, d'où la grande importance d'avoir un code source qui soit open-source.

4.2.5 Limite

Mais selon moi, la plus grande limite des cryptomonnaies et des registres distribués est le fait que l'apport de confiance et l'absence de la nécessité d'un tiers ne fonctionne que pour le transfert monétaire, mais ne change rien au problème dans sa globalité.

Par exemple, quand j'utilise une cryptomonnaie pour acheter quelque chose, le transfert sera fait de manière complétement sûre et sécurisée, mais rien n'empêche le destinataire de simplement partir avec l'argent sans me livrer la chose due.

Certes, le même problème existe dans le monde réel et des lois sont là pour éviter cela, mais si l'on voulait réellement devenir indépendant de tout tiers, il faudrait établir une sorte de « confiance de bout en bout ». C'est ce que les *smart contracts* permettent de faire et c'est à mon sens un sujet à développer.

4.3 Blockchain

La *blockchain* est l'implémentation la plus répandue et la plus populaire d'un registre distribué qui gère une cryptomonnaie. Les deux plus grandes, le *Bitcoin* et *Ethereum*, qui capitalisent aujourd'hui à elles seules près de 60% du marché, sont basées sur la *blockchain*.[37]

4.3.1 Fonctionnement

Son fonctionnement repose sur des blocs, qui sont ajoutés les uns après les autres pour former une chaine, chaque bloc référençant le précédant. Un bloc contient un nombre fini de transactions, et pour être valide, il doit être miné.

L'algorithme de consensus de la plupart des *blockchains* est basé sur la *proof of work*, la preuve de travail. Les nœuds sont des mineurs, qui cherchent tous en même temps à résoudre un problème cryptographique donc la difficulté est régulièrement ajustée pour durer un temps définit (10 minutes pour le Bitcoin, 15 secondes pour Ethereum).[38]

[37] Global Charts | CoinMarketCap. [en ligne].
[38] BLOCKGEEKS. Basic Primer: Blockchain Consensus Protocol. *Blockgeeks* [en ligne]. 2018.

Le nœud qui réussit à trouver la réponse au problème peut donc proposer sa solution au réseau, et une fois celle-ci validée par les autres nœuds, le bloc est ajouté à la chaine. Le mineur touche une récompense pour son travail : c'est de cette manière que la cryptomonnaie est émise.

Si deux mineurs trouvent la solution au problème au même moment, il y un *fork* et deux chaines coexistent. La « vraie » *blockchain* sera celle qui sera adopté par le plus de monde, soit la plus longue. En effet, il est très improbable que deux mineurs rajoutent, en même temps et encore une fois, un bloc de plus.

Pour éviter tout problème du genre, on recommande d'attendre la validation de quelques blocs pour confirmer un transfert.

4.3.2 Limites

La *blockchain* est une invention qui a profondément révolutionné la manière d'appréhender les systèmes décentralisés. Pour la première fois, deux entités ont pu établir une relation de confiance sans aucun intermédiaire ni registre centralisé mais simplement en faisant confiance à un code source commun basé entièrement sur la cryptographie.

Mais, dans sa forme actuelle, elle peut poser problème.

4.3.2.1 Proof of work

De par son fonctionnement même, une *blockchain* basée sur un système de *proof of work*, comme celle du Bitcoin, est contre-productive : Des millions d'ordinateurs, partout dans le monde, perdent du temps et de l'énergie à essayer de trouver par *bruteforce* la solution à un problème cryptographique… encore et encore !

Aujourd'hui, en juillet 2018 le Bitcoin gaspille 71.12 TWh / an. Si c'était un pays, il serait le 41e plus gros consommateur de la planète, juste derrière le Chili et l'Autriche et devant la République tchèque et la Suisse. Ethereum, lui, serait 71e.[39]

Néanmoins, il est important de noter que ce problème est de plus en plus pris au sérieux : Ethereum par exemple, va progressivement abandonner le système de *proof of work* pour un système de *proof of stake*, moins énergivore.

[39] Bitcoin Energy Consumption Index. *Digiconomist* [en ligne]. 2018.

4.3.2.2 Mineurs

Le rôle des mineurs a beaucoup trop dérivé. D'abord sensés fournir les vérifications au réseau en échange d'une contrepartie, l'aspect de participation a presque totalement laissé sa place à la motivation pécunière tellement le gain potentiel est énorme, et les seuls mineurs encore actifs du réseau sont devenus des énormes *datacenters*.

La baisse du nombre de *tokens* créés à chaque bloc, couplée à l'augmentation massive de trafic sur le réseau a fait tellement augmenter le coût des commissions que l'utilisation d'une monnaie comme le Bitcoin pour de petits paiements est totalement absurde. Ceci va même empirer lorsque ces commissions deviendront la seule source de revenus des mineurs, une fois tous les *tokens* mis en circulation.

4.3.2.3 Non-scalabilité

La scalabilité est la capacité d'un produit informatique à faire face à une forte demande sans voir ses performances diminuer.

Etant donné que le nombre de blocs émis est régulé dans le temps et que la taille des blocs est limitée, le nombre de transactions est limité. Une cryptomonnaie basée *blockchain* n'est pas scalable, car elle ne peut pas s'adapter à une forte montée en charge et cela va limiter le nombre de transactions par seconde.[40]

Pire, lorsqu'un réseau est très sollicité, comme c'est le cas du Bitcoin actuellement, les personnes qui souhaitent voir leurs transactions validées rapidement vont offrir une plus grande commission, faisant encore monter le prix des commissions par simple effet d'offre et de demande.

4.3.2.4 Cryptographie pré-quantique

Un dernier problème, bien que moins actuel et qui ne concerne que certaines monnaies (Bitcoin et Ethereum notamment) mais qui pourrait poser problème à long terme, c'est que leurs algorithmes ne sont pas *quantum-proof*. On suppose qu'un jour des ordinateurs quantiques seraient capables de casser certaines cryptographies asymétriques en retrouvant des clés privées à partir des clés publiques.

La méthode pour y parvenir est même déjà connue, mais n'est pas réalisable avec des ordinateurs classiques.

Par ailleurs, des ordinateurs quantiques pourraient très facilement enchainer les *proof of work* et attaquer le réseau sans avoir besoin de contrôler plus de 50% des nœuds.

[40] MASSET, Arnaud. Le bitcoin est-il bientôt prêt à concurrencer PayPal et Visa? *Le Temps* [en ligne]. 18 décembre 2017.

4.4 IOTA

IOTA est une cryptomonnaie développée par l'IOTA Foundation, une organisation à but non lucratif basée en Allemagne. Son marché étant orienté pour une utilisation dans le domaine de l'IoT, elle est très différente des cryptomonnaies habituelles. En effet, si une bonne partie des cryptomonnaies sont basés sur la *blockchain*, ce n'est pas le cas d'IOTA. Ses créateurs ont complétement innové en basant sa technologie sur le concept de *graphe orienté acyclique*, qu'ils ont surnommé le « Tangle ».[41]

4.4.1 Le Tangle

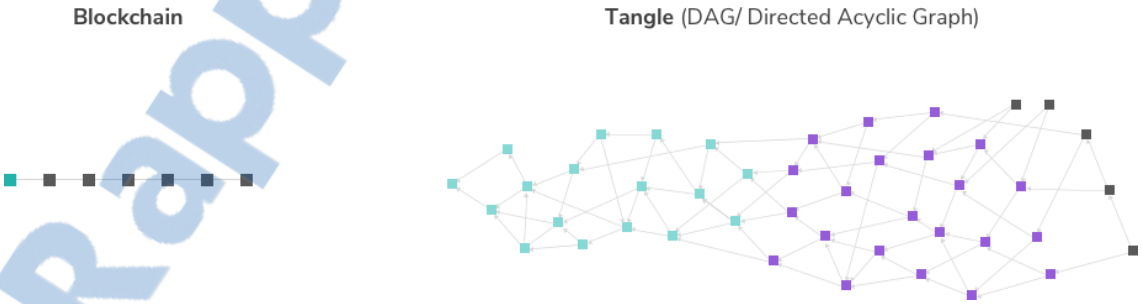
Le Tangle fonctionne sur le modèle d'un graphe orienté acyclique. Chaque transaction est elle-même un « bloc », et le registre n'est pas linéaire comme le serait une *blockchain*.

Par *graphe* on entend une structure composée d'objets qui sont liés les uns aux autres (dans le Tangle, ce lien est une référence), *orienté* veut dire que les références se font toutes dans la même direction et *acyclique* signifie qu'il n'y a pas de boucles dans un tel graphe.[42]

Pour qu'une transaction soit valide dans le Tangle, il faut qu'elle référence deux autres transactions aléatoires et effectue une petite *proof of work* (de l'ordre de quelques secondes, simplement pour éviter les spammeurs).

Automatiquement, une transaction sera réputée confirmée du moment qu'elle se sera faite une place dans le graphe : elle sera référencée par des transactions elles-mêmes référencées par d'autres transactions, etc.

Figure 3 : Comparaison visuelle entre la Blockchain et le Tangle[43]



[41] IOTA FOUNDATION. Documentation. *iota.org* [en ligne]. 2018.
[42] Graphe orienté acyclique. *Wikipédia* [en ligne]. 2 mai 2018.
[43] IOTA FOUNDATION. What is IOTA? [en ligne]. 2018.

4.4.2 Fonctionnement

L'accès au portefeuille se fait à l'aide d'un *seed* : une chaine de 81 caractères. Cette chaîne étant suffisamment complexe, elle sert à la fois d'identifiant et de mot de passe. Cette simple chaine, qu'il faut bien évidement garder secrète, permet de générer une infinité de paires de clés publique / privée de manière incrémentale.[44]

C'est à l'adresse, identique à la clé publique, qu'est attribué un solde de iotas. C'est cette partie qui est visible sur le réseau. Un *seed* peut donc posséder plusieurs adresses, car celles-ci sont « jetables ».

En effet, dans IOTA, les adresses ne sont utilisables qu'une seule fois. L'algorithme cryptographique que le système utilise est basé sur un système de *one-time signature*. Bien qu'il soit possible d'utiliser une adresse plusieurs fois, c'est très fortement déconseillé car un individu malveillant pourrait croiser ces signatures et retrouver la clé privée.

On peut voir ces adresses comme des tirelires : on peut y mettre des fonds autant de fois que l'on en a envie, mais pour les utiliser il faut « casser » la tirelire et elle n'est plus utilisable de manière sécurisée. S'il l'on ne souhaite dépenser qu'une partie des iotas disponibles sur une adresse, le solde est expédié sur une nouvelle adresse dérivée du *seed* et ce transfert fait partie intégrante de la transaction.

Par ailleurs, pour pouvoir participer au registre distribué qu'est IOTA, il faut utiliser un nœud. Mais l'installation d'un nœud est relativement compliquée et les ressources nécessaires pour le faire tourner sont très élevés, trop pour des objets connectés.

Par conséquent, il est possible d'utiliser soit un *full node*, soit un *light node*.

Le *full node* stocke une réplique du Tangle et communique avec d'autre nœuds, c'est donc grâce à lui qu'il est possible d'envoyer des transactions ou de s'informer sur des transactions existantes.

Le *light node* est une méthode qui permet de se connecter directement à un *full node*, et de participer au réseau en envoyant des commandes à l'aide de l'API. C'est ce que fait l'application officielle. Néanmoins, même en cas de ressources limités, le *seed* ne sort jamais du *light node* et toutes les signatures sont faites localement. C'est pour émettre et recevoir qu'il y a besoin d'un *full node*.

[44] IOTA FOUNDATION. Documentation. *iota.org* [en ligne]. 2018.

4.4.3 Avantages

Dans IOTA, il n'y a plus de mineurs et le *proof of work* est réduit au strict minimum. C'est en effectuant des transactions qu'on valide les autres transactions. De cette manière, chaque utilisateur du réseau participe activement à l'achèvement du consensus. Tous les utilisateurs étant égaux, il n'y a pas de conflits d'intérêts comme on peut le voir dans une *blockchain* classique (mineurs contre utilisateurs).[44]

Par ailleurs, le Tangle est *scalable*, étant donné que la validation des transactions se fait en parallèle. Chaque transaction en validant deux autres, plus il y aura de transactions, plus d'autres transactions seront validées. Par conséquent, la rapidité à laquelle les transactions seront confirmées augmentera en fonction du nombre de transactions. Le Tangle est fait pour s'agrandir.

Mais l'argument principal de la fondation IOTA, c'est bien évidemment l'absence complète de commissions. Puisque chaque utilisateur participe activement à la validation des transactions et à l'achèvement du consensus, aucune commission n'est nécessaire pour inciter les utilisateurs à fournir un travail. Le travail doit être fournit pour pouvoir émettre des transactions.

Cette absence de commission, la rapidité à laquelle les transactions sont validées et son adaptabilité sont justement les raisons qui démarquent IOTA des autres cryptomonnaies et qui font d'elle, justement, la monnaie parfaite pour l'IoT et tous les systèmes nécessitants un nombre important de microtransactions.

Finalement, et même si ce n'est pas une propriété intrinsèque du Tangle, les algorithmes utilisés par IOTA sont *quantum-proof*. Ils n'ont rien à craindre de l'arrivée potentielle des ordinateurs quantiques.[45]

4.4.4 Limites

4.4.4.1 Coordinateur

Le principal problème d'IOTA, c'est que la monnaie est activement en développement. Un des avantages cités précédemment est que plus la monnaie sera utilisée, plus elle sera sécurisée. Mais son utilisation relativement marginale fait que, pour l'heure, il n'y a pas assez d'utilisateurs sur le réseau pour assurer sa sécurité.

[45] FEENSTRA, Evan. How is the tangle from IOTA quantum resistant? *Quora* [en ligne]. 29 novembre 2017.

La fondation IOTA a donc été contrainte d'y installer ce qu'ils appellent un coordinateur. Il s'agit d'un nœud du réseau qui effectue des transactions de 0 iotas : les *milestones*. Toutes les transactions référencées directement ou indirectement par un *milestone* sont véritablement confirmées.[46]

Le rôle du coordinateur est de faire avancer le Tangle dans la bonne direction, celui-ci étant encore potentiellement sensible aux attaques. La fondation assure que le Tangle peut très bien fonctionner sans coordinateur si le réseau est suffisamment utilisé – la proportion d'utilisateurs « honnêtes » dépassant fortement la proportion d'attaquants potentiels – et devrait le retirer du réseau courant 2018.

Néanmoins, le réseau reste aujourd'hui fortement centralisé et soumis à la fondation IOTA à cause de ce coordinateur.

4.4.4.2 Adresses jetables

Un autre problème concerne la gestion des adresses. Les adresses ne peuvent être utilisées qu'une seule fois pour envoyer des fonds.

Donc, à chaque fois que l'on souhaite envoyer des fonds, il faut demander au destinataire son adresse actuelle. Il n'est pas non plus possible de poster une adresse de réception indéfiniment sur un site web : il faudra le mettre à jour à chaque fois que l'on change d'adresse.

Tout ceci implique que lors d'un transfert de fonds *machine to machine*, un protocole doit encadrer cet échange et doit faire en sorte que la source demande l'adresse de réception au destinataire.

4.4.4.3 Trinaire

La fondation IOTA a pris le pari téméraire de baser entièrement son architecture sur un système « trinaire équilibré » ou *balanced ternary*[47] et non pas binaire. Son fonctionnement, l'encodage des adresses, des *seeds*, des messages, tout est encodé sous la forme de *trytes*, des groupes de 3 *trits*.

Un *tryte*, c'est donc 3 x 3, soit 27 états différents. Pour cette représentation, ce sont les lettres de l'alphabet et le chiffre 9 qui sont utilisés, soit 27 caractères différents. En soi, un entier ou un message encodé en *trytes* prend donc beaucoup moins de place que le même entier ou le même message encodé en binaire.

[46] GAL, Alon. The Tangle: an illustrated introduction. *IOTA* [en ligne]. 28 février 2018.
[47] Balanced ternary. *Wikipedia* [en ligne]. 21 mai 2018.

Mais pourquoi ce choix ? La fondation défend que le trinaire est plus optimisé que le binaire et possède différents avantages. Par exemple, celui-ci est capable de gérer les nombres négatifs sans avoir besoin de les signer ou bien il peut arrondir les nombres réels simplement et nativement.[48] Mais l'argument principal reste la rapidité et l'optimisation de la mémoire par rapport au binaire.[49]

Le problème, c'est que faire tourner du trinaire sur des processeurs binaires est loin d'être optimal, et les implémentations actuelles d'IOTA doivent donc faire le travail de traduction.[50] La fondation a bien des plans pour des puces trinaires qui fonctionneraient sur des objets connectés, mais c'est risqué.[51]

Premièrement, durant la guerre froide, l'URSS a mené des expérimentations dans ce sens qui se sont avérées non-concluantes.[48] Deuxièmement, si le développement de composants électroniques trinaires par la fondation est fructueux, le fait de mettre dans les mains de la même société la gestion de la monnaie et la vente du matériel nécessaire la mettrait dans une dangereuse position de monopole.

4.4.4.4 Algorithme cryptographique

Par ailleurs, pour optimiser ce système, la fondation a choisi de développer son propre algorithme cryptographique en trinaire[52], même si la plupart des cryptographes du monde diront que c'est une très mauvaise idée. C'est une règle d'or : *« Don't roll your own crypto ».*

> *« Anyone, from the most clueless amateur to the best cryptographer, can create an algorithm that he himself can't break. It's not even hard. What is hard is creating an algorithm that no one else can break, even after years of analysis. »*[53]

[48] BUNTINE, Andrew. The Balanced Ternary Machines of Soviet Russia. *The Practical Dev* [en ligne]. 21 novembre 2016.
[49] SØNSTEBØ, David. When is Ethereum going to run in to serious scaling issues? *reddit* [en ligne].
[50] JOHNSON, Nick. Why I find Iota deeply alarming. *Hacker Noon* [en ligne]. 26 septembre 2017.
[51] What is JINN project? *Iota Stack Exchange* [en ligne].
[52] IVANCHEGLO, Sergey. Drawing the line for "don't roll own crypto". *Come-from-Beyond* [en ligne]. 2 mars 2018.
[53] SCHNEIER, Bruce. Memo to the Amateur Cipher Designer. [en ligne]. 15 octobre 1998.

Sans surprise, celui-ci a vite été cassé par une équipe du MIT, qui y a découvert une collision.[54] La fondation a répondu principalement de deux manières ; que la résistance à une collision n'est pas nécessaire dans IOTA et que les chances que ça arrive sont négligeables.[55] Entre temps, ils ont dû changer d'algorithme et celui-ci est « traduit » en binaire.[56]

L'un des cofondateurs argumente même que cette vulnérabilité était volontaire et a été placée là en tant que protection anti copie.[57] Bien qu'il s'agisse selon moi d'une excuse complétement bidon, c'est très dérangeant : IOTA a fait le choix de rendre son code *open source*, avec tout ce que ça implique. Mais rendre le code public et y placer ce genre de piège est une attitude complétement hostile et malhonnête envers la communauté open source.[58]

4.4.4.5 Fondation IOTA

Suite à ces évènements, le ton est monté d'un cran entre certains développeurs clés derrière la fondation IOTA (dont le fondateur) et des développeurs ou cryptographes reconnus, ces derniers s'étant rangés du côté de l'équipe du MIT. Des attaques personnelles ont été échangés par e-mail[59] et même des menaces de poursuites judiciaires.[60]

Tout le drame autour de cette vulnérabilité est assez inhabituel dans le monde du logiciel open source, dans lequel les développeurs tendent à collaborer avec les chercheurs plutôt que de les menacer.

Les développeurs d'IOTA estiment être des visionnaires qui se font attaquer par des gens qui ne comprennent rien au futur, tout en cherchant à toujours justifier tous leurs choix plutôt que d'accepter les critiques.[61] D'un autre côté les « anti-IOTA » se moquent régulièrement de ce que fait la fondation et postent régulièrement des provocations.[62]

---

[54] HEILMAN, Ethan et al. IOTA Vulnerability Report: Cryptanalysis of the Curl Hash Function Enabling Practical Signature Forgery Attacks on the IOTA Cryptocurrency. [en ligne]. 7 juin 2018.
[55] WEIS, Steve. Tweet de @sweis. *Twitter* [en ligne]. 22 février 2018.
[56] HOP, Eric. Debunking the 'IOTA Vulnerability Report'. *IOTA Demystified* [en ligne]. 25 février 2018.
[57] IVANCHEGLO, Sergey. Come_from_Beyond comments on Integrity questions. [en ligne]. 1 janvier 2018.
[58] JOHNSON, Nick. Why I find Iota deeply alarming. *Hacker Noon* [en ligne]. 26 septembre 2017.
[59] LIMO. Full Emails of Ethan Heilman and the Digital Currency Initiative with the IOTA Team leaked. *The Tangler* [en ligne]. 24 février 2018.
[60] IVANCHEGLO, Sergey. Tweet de @c___f___b. *Twitter* [en ligne]. 26 février 2018.
[61] SØNSTEBØ, David. The Transparency Compendium. *IOTA* [en ligne]. 15 juin 2017.
[62] ANDRE□. Tweet de @puellavulnerata. *Twitter* [en ligne]. 2 juillet 2018.

4.4.5 MAM

Le *Masked Authenticated Messaging* est une fonctionnalité d'IOTA qui permet de transmettre des données au travers du registre distribué de façon cryptée et sécurisée, tout en permettant de valider l'identité de l'émetteur de ces données.[63]

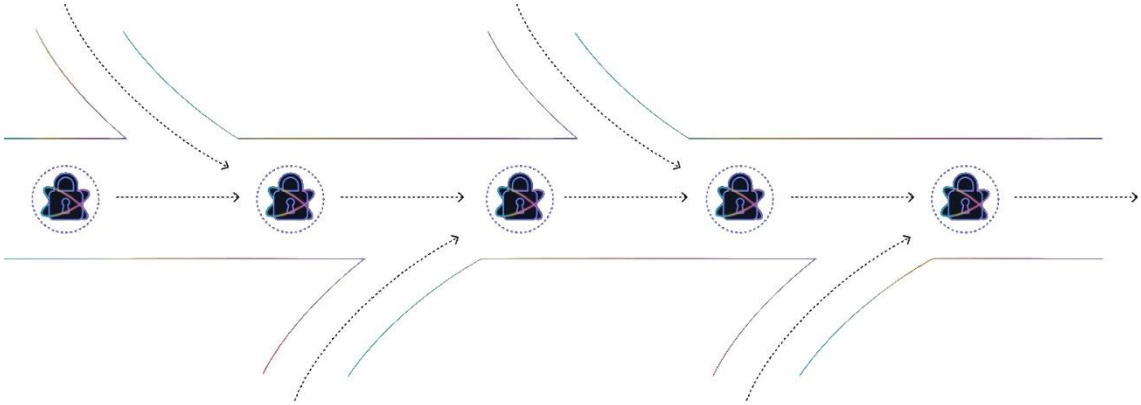
Il est principalement utilisé pour transmettre des informations textuelles, bien que rien n'empêche la transmission d'autres types de données. Il n'y a pas de limite de taille, mais un trop long message risquant de saturer le réseau, il sera rejeté par les nœuds.[64]

MAM est basé sur un système *Publish / Subscribe*. Un *stream* (ou *channel*) est associé à chaque *seed* et des lecteurs peuvent s'y abonner.

Pour pouvoir déchiffrer les messages, il faut utiliser le *root*, une chaine de *trytes* dérivée du *seed*. Chaque message posté sur le réseau est identifié par une adresse, qui correspond à un hash du *root*.

Chaque message contient aussi le *root* du message suivant. À partir d'un *root* donné, on peut donc suivre le flux du message courant jusqu'au plus récent, mais il n'est pas possible de voir un message antérieur au message courant, à moins que le root précédent ait été donné.

Figure 4 : Représentation d'un flux MAM [64]



Sur cette image on visualise bien le fait qu'on ne peut remonter la chaine que dans un sens.

Il y a trois modes de confidentialité : *Public, Private* et *Restricted*.

Le mode *Restricted* permet l'ajout d'une clé cryptographique supplémentaire, en plus du *root*. Cette clé est librement ajoutée par l'auteur et celui-ci peut la modifier à sa guise. Ci-après se trouve un tableau comparatif des différents modes de confidentialité.

[63] ABMUSHI. IOTA: MAM Eloquently Explained. *ABmushi* [en ligne]. 24 février 2018.
[64] HANDY, Paul. Introducing Masked Authenticated Messaging. *IOTA* [en ligne]. 4 novembre 2017.

Tableau 2 : Comparaison des trois modes de confidentialité de MAM

	Public	*Private*	*Restricted*
Adresse	= root	= hash (root)	= hash (root)
Déchiffrement avec	Adresse ou root	Root	SideKey
Accès avec	Adresse ou root	Root	Root et SideKey

5. Prototype

Le développement du prototype s'est déroulé en deux parties. Tout d'abord, pour pouvoir interagir avec le Tangle, il fallait avoir un nœud pour se connecter au réseau, j'ai donc installé un *full node*.

Par la suite, j'ai développé l'application à proprement parler. Elle permet l'achat et la vente de données via le Tangle, en payant en iotas et en transmettant les données avec MAM. L'initiative de la vente peut venir de l'acheteur ou du vendeur. Tout le reste se passe automatiquement.

L'idée derrière l'application était de pouvoir réutiliser la même version pour tous les types de machines. Elle est donc basée sur un environnement et sur des librairies réutilisables sur différents systèmes d'exploitation.

5.1 Application

5.1.1 Processus de vente

Il y a deux manières d'effectuer une transaction : lorsque l'initiative vient de l'acheteur (« je veux t'acheter quelque-chose ») et lorsque l'initiative vient du vendeur (« je veux te vendre quelque-chose »).

Lors des échanges préliminaires, les informations à propos de la vente sont représentées par un JSON transmis dans le corps de la requête. Ces informations sont, entre autres, le type de données que l'on veut acheter, la quantité ou l'adresse sur laquelle on veut recevoir la réponse. Dans les diagrammes ci-dessous, ce JSON est représenté par le tableau qui accompagne les messages.

Les différentes instances de l'application peuvent savoir s'il elles ont un processus de vente ou d'achat qui est ouvert grâce à une petite base de données stockée localement.

Pour différencier les ventes entre elles et pour s'assurer de l'identité de l'expéditeur suite à la réception d'un transfert de iotas entrant, le vendeur génère une adresse de réception différente pour chaque vente, qu'il transmet à l'acheteur via le JSON.

On peut donc savoir facilement si c'est bien le bon expéditeur qui a payé lorsque les fonds arrivent sur une adresse en particulier.

5.1.2 Modules

Chaque machine exécute une instance de l'application, elle-même constituée de plusieurs modules. Les modules *main*, *entryPoint, rightsManager* et *sensor* sont les seuls qui tournent en permanence. Les modules *buyer, seller* et *mam* sont exécutés sur demande.

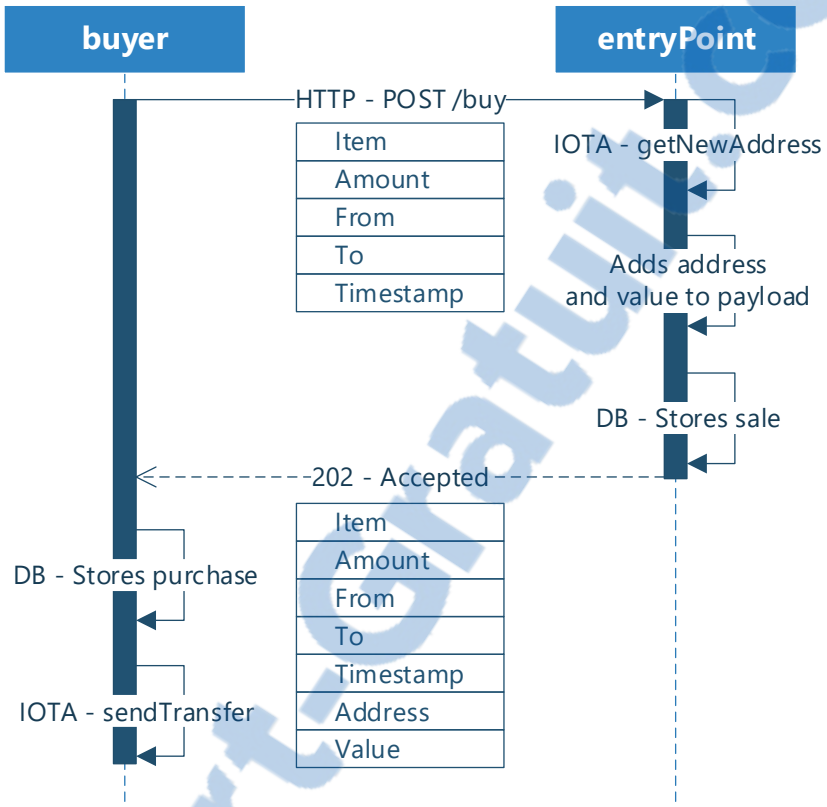
Tableau 3 : Aperçu des modules du prototype

Module	*Exécution*	*Rôle*
Main	Au démarrage	C'est le module qu'il faut lancer pour démarrer l'application. Il sert d'orchestrateur pour tous les autres modules. Il permet aussi de garder en mémoire les variables globales.
EntryPoint	Permanent	C'est le point d'entrée des requêtes HTTP POST. Il accepte : • Les requêtes de vente • Les requêtes d'achat • Les requêtes d'ajouts de *streams*
RightsManager	Toutes les 5 secondes	C'est un script qui est exécuté à intervalles réguliers pour chercher les payements entrants. Si un payement entrant est confirmé, un *stream* de données est créé et le *root* de celui-ci est expédié à l'adresse renseignée par l'acheteur.
Sensor	Toutes les minutes	C'est un ou plusieurs scripts exécutés à intervalles réguliers. Chacun étant relié à un hypothétique capteur, ils vont peupler les streams de leurs informations.
Buyer	Manuellement	Il permet de lancer une demande d'achat
Seller	Manuellement	Il permet de lancer une demande de vente
Mam	Manuellement	Il permet de lire les messages des *streams* abonnés

5.1.3 Initiative de l'acheteur

Lorsque l'initiative vient de l'instance de l'acheteur (*buyer*), celui-ci contacte l'*entryPoint* de l'instance du vendeur pour lui demander une adresse de versement et un montant.

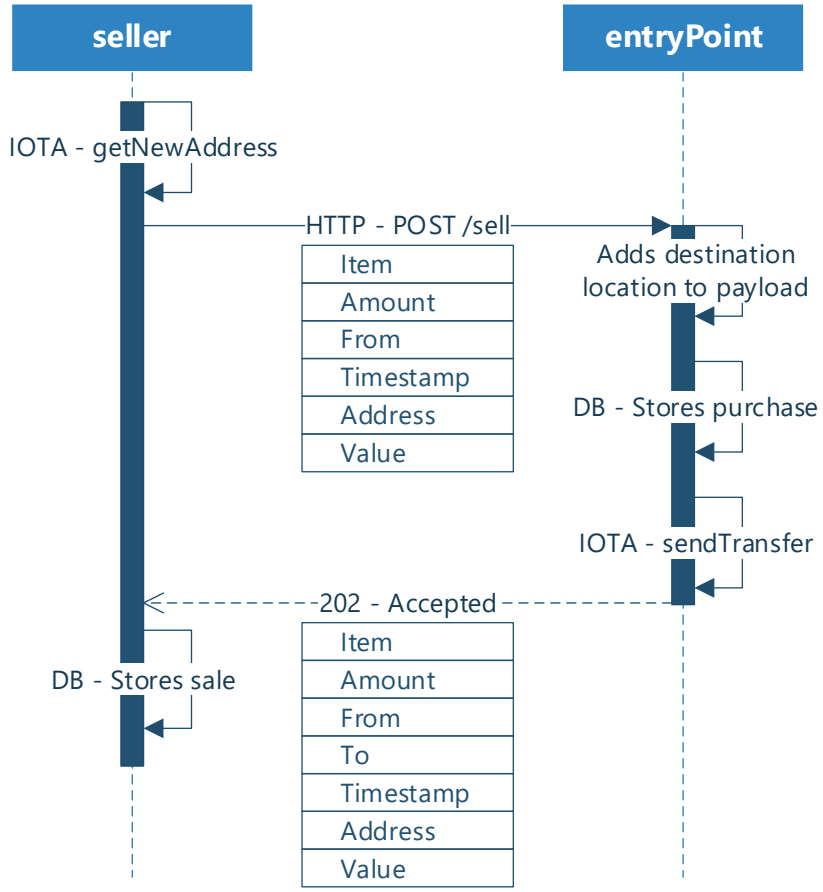
Figure 5 : Diagramme de séquence (initiative de l'acheteur)



5.1.4 Initiative du vendeur

Lorsque l'initiative vient de l'instance du vendeur (*seller*), celui-ci contacte l'*entryPoint* de l'instance de l'acheteur pour l'informer de son souhait et pour récupérer l'adresse de destination (*To*).

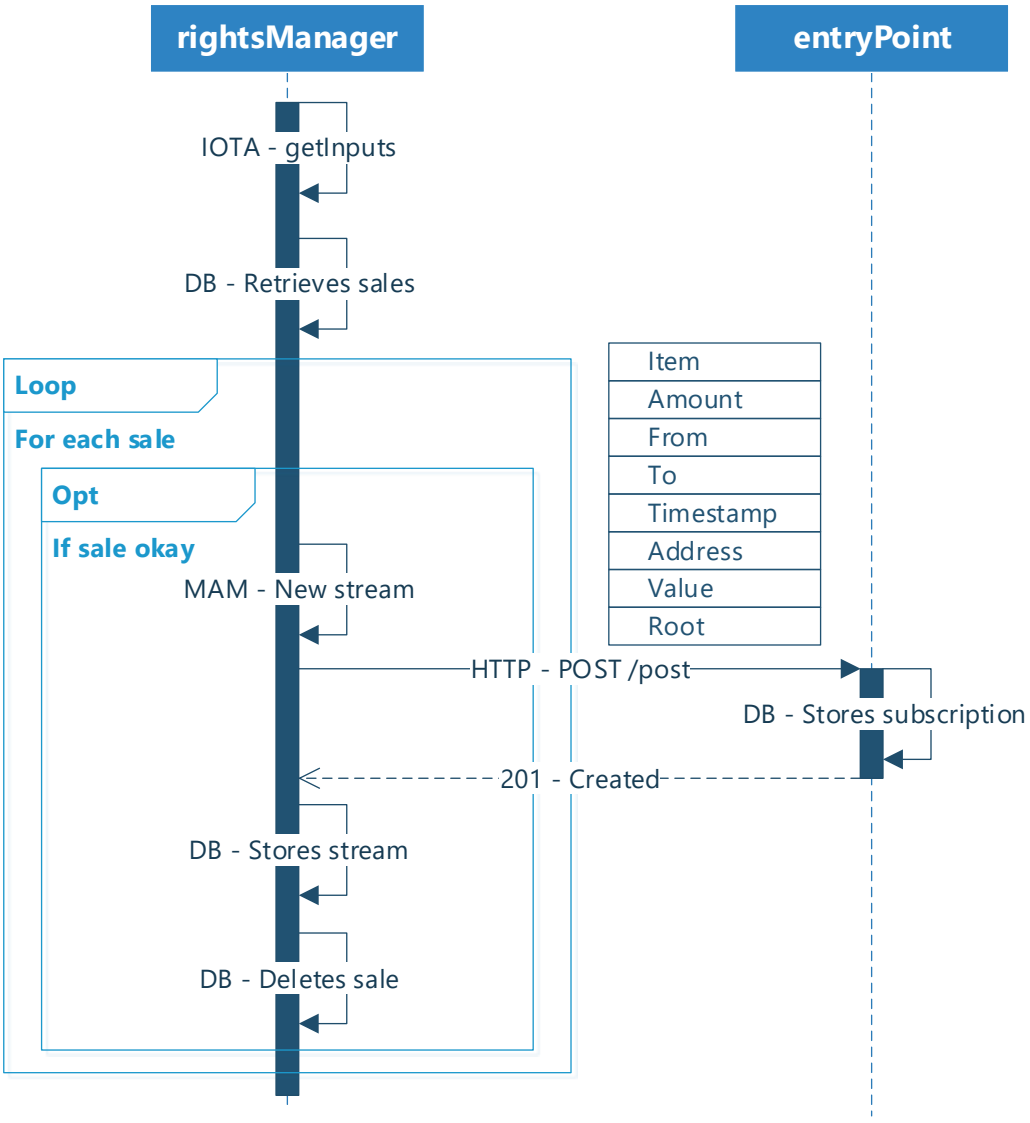
Figure 6 : Diagramme de séquence (initiative du vendeur)



5.1.5 Validation des payements

Régulièrement, dans chaque instance de l'application, le module *RightsManager* vérifie si un payement entrant correspond à une vente (en l'identifiant par l'adresse de réception). Si c'est le cas, il crée un nouveau *stream* de données et il envoie à l'adresse renseignée par l'acheteur le *root* qui permet d'y accéder. Pour finir, il clôt la vente.

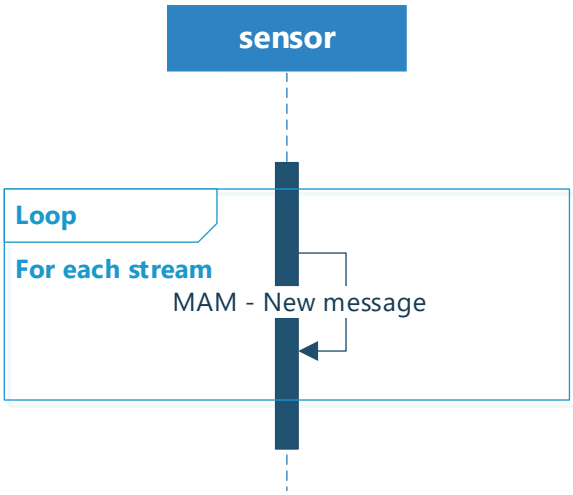
Figure 7 : Diagramme de séquence (validation des payements)



5.1.6 Capteur

Un ou plusieurs capteurs sont présents dans le cas où la machine est un objet connecté qui souhaiterait transmettre les données qu'elle recueille. Ce module envoie régulièrement un message avec de nouvelles données sur tous les *streams* ouverts en cours.

Figure 8 : Diagramme de séquence (capteur)



5.1.7 Modèle de données

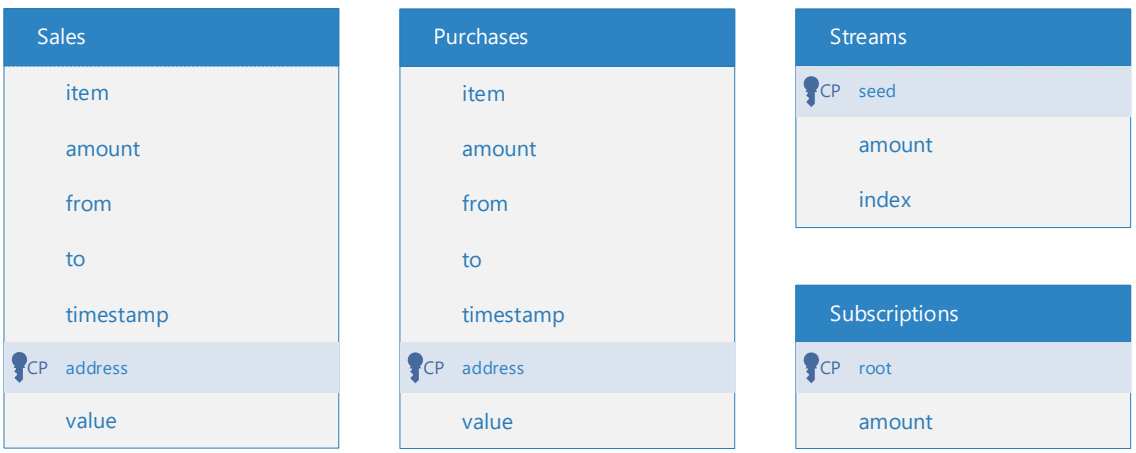
Une base de données SQLite est présente dans l'application pour permettre le stockage temporaire des données.

Les tables *Sales* et *Purchases* sont identiques, mais correspondent respectivement aux ventes et aux achats d'une machine.

La table *Streams* stocke les *streams* MAM qui sont activement utilisés par les capteurs.

La table *Subscriptions* stocke quant à elle les *streams* auxquels la machine est abonnée.

Figure 9 : Modèle UML



5.2 Choix et technologies utilisées

Puisqu'il n'est pas possible de tout distribuer, de tout intégrer à IOTA, il fallait utiliser différentes technologiques externes. J'ai dû faire certains choix, que ce soit pour l'utilisation de certaines technologies ou au moment de l'utilisation de celles-ci. Ce chapitre relate ces aspects du développement de mon prototype.

5.2.1 Full Node

Sur une machine virtuelle Debian, j'ai installé IRI, la *IOTA Reference Implementation*, qui permet de faire tourner un *full node* sans interface graphique.[65] Mais dans le réseau d'IOTA, il n'y a pas de découverte automatique des pairs. Autrement dit, il faut aller soi-même chercher des *neighbors*, des nœuds voisins, pour s'appareiller manuellement.

Je suis donc allé sur le Discord officiel de la communauté IOTA et j'ai demandé à trouver des *neighbors*. Après un partage d'IPs et un accord mutuel sur le protocole à employer (TCP ou UDP), j'avais quelques voisins avec lesquels mon nœud communiquait.

Néanmoins, mon nœud avait toujours du mal à se synchroniser avec le réseau. Je pense que c'est les voisins qui posaient problème, ces derniers étant aussi des « nouveaux », j'imagine qu'ils n'étaient pas complétement intégrés au réseau.

D'autant plus que les connections statiques posent problème : beaucoup de mes voisins ne répondaient plus au bout de quelques jours et c'était difficile de trouver une machine stable et bien synchronisée au réseau avec laquelle je pouvais me connecter.

J'ai donc utilisé Nelson, un outil développé par la communauté pour la découverte automatique de pairs.[66] Avec mes quelques voisins statiques complétés par les voisins dynamiques de Nelson, c'était mieux et mon nœud commençait enfin à se synchroniser.

Mais au bout de quelques jours, même si le téléchargement du Tangle avait un peu avancé, je remarquais que l'opération risquait de mettre beaucoup de temps : environ une semaine pour une synchronisation complète.

Après quelques recherches, j'ai découvert un tutoriel signé *IOTA Partners* qui permet de télécharger un *snapshot* de la base de données actuelle du réseau dans le but d'éviter toute cette attente pour se synchroniser.[67] En effet, elle pesait près de 60 GB.

Finalement, mon *full node* était synchronisé et opérationnel.

[65] IOTA FOUNDATION. *IOTA Reference Implementation* [en ligne]. 24 février 2018.
[66] SEMKO, Roman et al. *CarrIOTA Nelson, allows auto-discovery (P2P) for IOTA-Fullnodes* [en ligne]. 3 février 2018.
[67] IOTA Full Node Copy-Paste Installation Guide. [en ligne]. 2018.

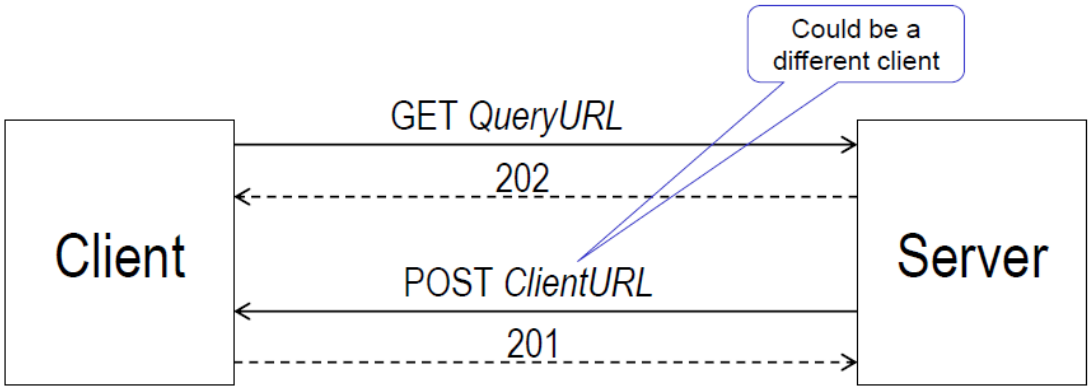
5.2.2 Request / Acknowledge / Callback

Le premier choix que j'ai dû faire concerne le protocole à utiliser en dehors des transactions dans le cade d'IOTA.

Au vu de la nature asynchrone des requêtes, et du fait que les transactions peuvent prendre plusieurs minutes pour être confirmés, il fallait utiliser un pattern qui permettait une certaine attente entre ces requêtes.

C'est tout naturellement que j'ai utilisé le pattern *Request / Acknowledge / Callback*. Le concept est simple, la machine source envoie une requête dans laquelle il précise l'adresse sur laquelle il souhaite voir arriver la réponse. Une fois le traitement terminé du côté du destinataire, une requête est lancée pour transmettre la réponse à la machine source.

Figure 10 : Fonctionnement du Request / Acknowledge / Callback[68]



Le prototype utilise un POST et non pas un GET pour la première requête, car il faut transmettre les informations dans le corps de la requête et non pas dans les headers.

Son seul défaut est d'imposer l'utilisation d'un serveur Web des deux côtés, ce qui n'est pas un problème dans le cadre d'un échange M2M, d'autant plus que la même copie de l'application tourne aussi des deux côtés.

[68] DUGERDIL, Philippe. *Cours de Génie Logiciel*. 9 mars 2018.

5.2.3 HTTP

L'implémentation du *Request* / *Acknowledge* / *Callback* se fait dans le cadre d'un échange HTTP. L'utilisation de ce protocole étant presque obligatoire aujourd'hui au vu de son universalité, ce n'était pas non plus un problème.

Pour pouvoir transmettre les données, c'était déjà tout trouvé : des appels HTTP POST qui transmettent les données directement dans le corps de la requête au format JSON. De plus, c'est ce qui était déjà utilisé pour les appels via l'API d'IOTA mais aussi dans beaucoup d'autres API.

Le point auquel il faut faire attention en utilisant des API HTTP reste toujours l'obligation de s'accorder sur un protocole commun.

5.2.4 Node.js

Pour ce qui est de l'environnement, j'ai assez naturellement choisi Node.js, un environnement d'exécution libre en JavaScript.

Son principal atout et de pouvoir exécuter des scripts côté serveur, mais son architecture événementielle lui permet de gérer les entrées et sorties de manière asynchrone. Il possède aussi l'avantage d'avoir un système intégré de gestion de *packages* (NPM) et une communauté très active.[69]

Grâce à la possibilité d'exécuter des traitements asynchrones, cet environnement est très apprécié dans le monde des cryptomonnaies et de nombreuses librairies sont mises à dispositions en packages Node.js par les organismes qui gèrent ces cryptomonnaies.

Les raisons de mon choix sont multiples, le support d'exécutions asynchrones était déterminant, mais j'ai aussi trouvé plus simple de rester sur du JavaScript, car c'est dans ce langage que sont écrites les librairies les plus abouties qui sont mises en ligne par la fondation IOTA.

Par ailleurs, l'environnement de Node.js est très léger et peut tourner sur tous les systèmes d'exploitation courants, y compris sur ceux basés sur des processeurs ARM, très utilisés pour les objets connectés.

Finalement, c'est aussi un environnement qui permet de monter des serveurs web très simplement et nativement, mais aussi des scripts et des évènements récurrents, tout trois utilisés dans l'application.

[69] MAKINA CORPUS. Introduction à Node.js. [en ligne]. 29 janvier 2014.

5.2.5 MAM

L'utilisation de MAM était elle aussi très naturelle : c'est la solution que la fondation IOTA cherche à pousser pour l'envoi de messages et elle a été développé spécialement pour la publication de données de capteurs. Par conséquent, j'en ai profité pour le tester également.

De plus, MAM permet d'authentifier la source d'un message grâce à tous les avantages du registre distribué.

J'ai choisi d'utiliser le mode de confidentialité *private*. En effet, celui-ci a l'avantage d'être quand même sécurisé, mais moins restrictif que le *restricted* qui impose l'utilisation de la *side key*. La *side key* n'est pas un problème en soi, mais le fait que l'auteur puisse changer cette clé quand bon lui semble est à mon sens une trop grosse permission qui peut rompre la confiance dans l'échange.

5.3 Limites

5.3.1 Confirmation des transactions

Nous avons vu que pour émettre une transaction dans IOTA, celle-ci devait sélectionner aléatoirement deux autres transactions. De cette manière, elle se fait une place dans le graphe et la transaction est confirmée si, suffisamment sélectionnée par d'autres transactions, elle est référencée directement ou indirectement par un *milestone*.

Mais j'ai dû faire face à un gros problème : les transactions que j'envoyais ne se faisaient que rarement référencer, et plus le temps passait, moins l'algorithme de sélection avait de chance de sélectionner ma transaction.

IOTA possède deux mécanismes pour pallier à cela : le *reattach* et le *promote*. Le *reattach* sert à émettre une nouvelle fois la transaction, comme si c'était une nouvelle. N'importe qui peut le faire, puisqu'on ne change pas la signature. C'est très utile si la transaction qui avait été émise à la base se retrouve trop ancienne pour être référencée.

Le *promote* quant à lui sert simplement à émettre des transactions à 0 iotas qui vont référencer la transaction que l'on souhaite voir confirmée. L'idée est de spammer le réseau de transactions qui vont référencer la bonne : il y aura alors plus de chances d'être référencé indirectement par le coordinateur. Là aussi, n'importe qui peut le faire.

Pour en revenir au problème, deux fois sur trois, ma transaction n'était toujours pas confirmée au bout de dix minutes. J'ai même pu remarquer que la probabilité que ma transaction se fasse référencer était corrélée au nombre d'utilisateurs qui utilisent le nœud auquel j'étais connecté. Lorsque j'utilisais mon nœud, qui était peu sollicité, je devais systématiquement faire des *promotes* et des *reattach*s et ça pouvait facilement prendre jusqu'à 40 minutes.

J'ai donc dû me résoudre à utiliser un nœud public. Même si c'était un peu mieux, je me suis fait à l'idée qu'il était impossible de partir du principe qu'une transaction serait confirmée sans avoir besoin de faire de *promotes* ou de *reattachs*.

Par conséquent, j'ai implémenté dans l'application une fonction qui spamme automatiquement le réseau de *promotes* suite à l'émission d'une de mes transactions, dans le but qu'elle puisse être confirmée en moins de 10 minutes.

Le problème, c'est que ça pouvait planter, car il y a une chance non négligeable que la transaction que j'émette référence des transactions invalides, destinées à ne jamais être confirmées. Ici, le *reattach* était nécessaire et je devais me résoudre à le faire à la main, en prenant soin de redémarrer mon script.

Finalement, j'ai découvert un *pool* de nœuds qui automatise ces actions. C'était finalement bien plus simple et bien plus pratique, et c'est finalement la solution vers laquelle je me suis dirigé.[70]

Quoi qu'il en soit, je constate que, dans IOTA, on ne peut pas partir du principe qu'une transaction va être validée sans, peut-être, devoir la rattacher. Mais cette opération est très contraignante et prend forcément plus de temps.

5.3.2 Double spending

En testant le prototype, j'ai découvert que lorsque l'on a émis un transfert qui n'a pas encore été confirmé, les iotas sont toujours considérés comme propriété de l'adresse d'expédition.

Pour rappel, si lors d'un transfert tous les iotas ne sont pas dépensés, le solde est censé être expédié à une nouvelle adresse du *seed*, ceci afin d'éviter d'utiliser deux fois la même adresse et de risquer les failles de sécurité du mécanisme de *one-time signature*.

[70] CarrIOTA Field. [en ligne].

Le problème est que si j'émet une transaction, que celle si n'est pas encore confirmée et que je tente d'en émettre une autre, à la fois le *wallet* et l'API vont tenter de faire partir les fonds de la même adresse : conséquence, l'une des deux sera forcément rejetée par le réseau car assimilée à du *double spending*.

Et c'est là un problème très grave : on ne peut pas émettre plus d'une transaction à la fois. Sachant que ça peut prendre entre 5 et 40 minutes pour qu'une transaction soit confirmée, on ne peut pas envoyer de transaction plus d'une fois toutes les 5 à 40 minutes !

Les seuls moyens de contourner ce problème sont, soit d'utiliser plusieurs *seeds* différents, soit de réussir à récupérer l'adresse sur laquelle le solde de la première transaction va atterrir, puis de tenter d'émettre depuis cette adresse en générant soi-même les trytes nécessaires à une telle transaction. En effet, l'API ne permet pas une telle opération. Du coup, cette manipulation s'avère compliquée et part du constat pas forcément réaliste que la première transaction sera validée avant la deuxième.

5.4 Analyse SWOT

Suite au développement de mon prototype, j'ai identifié ci-après les forces, faiblesses, opportunités et menaces relatives au développement d'un concept d'échange automatisé et décentralisé de ressources en utilisant les technologies précitées.

Tableau 4 : Analyse SWOT

	Positif	**Négatif**
Interne	**Forces :** • Échange automatisé • Résilient • Indépendant de tout tiers • Polyvalent • Instanciable • Scalable • Absence de commissions	**Faiblesses :** • Confirmation des transactions aléatoire • Confiance nécessaire en dehors du payement • Choix d'une architecture non adaptée au matériel actuel (trinaire) • Système pas entièrement distribué à l'heure actuelle (coordinateur)
Externe	**Opportunités :** • Adaptable • Support technologique adapté aux façons de faire actuelles (librairies node.js, API, …)	**Menaces :** • Vulnérabilités dans les technologies utilisés (IOTA, MAM) • Attitude de la fondation IOTA

6. Conclusion

6.1 Bilan

Les systèmes distribués, c'est une solution à beaucoup de problèmes. Ils vont s'imposer. Ils se sont même déjà fait une grande place dans le monde de l'informatique et je pense que la fin des systèmes centralisés approche à grand pas.

En ce qui concerne IOTA, le bilan est mitigé. Bien que chaque nouvelle solution amène avec elle de nombreux problèmes, le Tangle est un sérieux concurrent de la *blockchain* et il a énormément de potentiel. Rien que sur le plan de la scalabilité, la faible consommation et l'absence de commission font d'elle une très bonne monnaie pour les objets connectés, mais aussi une très bonne cryptomonnaie tout court.

Mais d'un autre côté, la rigidité de l'équipe derrière la monnaie, certains de ses choix de designs (trinaire, algorithme de *hashage*…) et la difficulté de confirmer des transactions proprement – que ce soit avec ou sans coordinateur – font qu'elle n'est pas utilisable à l'heure actuelle, que ce soit pour des humains ou pour des objets.

Certes la technologie est encore en phase *beta*, mais ces différents problèmes risquent de sérieusement limiter son développement, au risque de se faire dépasser par d'autres alternatives et / ou de se faire lâcher par ses investisseurs.

Hedera hashgraph est justement l'une de ces alternatives. Elle est d'ailleurs si récente qu'elle n'existait pas encore lorsque j'ai commencé ce travail. Mais il serait très intéressant de voir ce qu'elle propose et de la comparer à IOTA.

> *« Est-il possible de procéder à un échange automatisé et décentralisé de ressources dans le monde de l'IoT ? »*

Oui. Ethereum et ses *smart contracts* le permettent déjà, mais avec beaucoup de contraintes. Cependant, une éventuelle diversification de leur part dans l'IoT est entièrement possible, car c'est un nouveau marché qui va prendre beaucoup d'ampleur. En outre, le concept derrière le Tangle, le *graphe orienté acyclique* a lui aussi beaucoup de potentiel grâce à ses nombreux avantages applicables à l'IoT.

> *« Est-ce faisable avec IOTA ? »*

Pas encore. Techniquement, oui c'est possible. Le prototype que j'ai développé est fonctionnel. Mais il y a beaucoup de contraintes et l'avenir de la technologie est très incertain, à un éventuel investisseur, je lui dirai d'attendre.

6.2 Futur

« De quoi sera fait le futur suite à ces recherches ? »

Je pense qu'on peut dire sans trop se tromper que les objets connectés et les *smart cites*, c'est le futur. C'est même le présent. Les plus grandes villes d'Europe sont déjà sur la voie pour devenir *smart* : Paris, Londres, mais aussi de plus petites comme Amsterdam, Barcelone ou Genève (et même Carouge !).[71]

C'est sûr qu'il s'agisse d'un grand changement, et toutes ces nouvelles technologies font peur à certaines personnes. Et on peut comprendre ces peurs. Que faire le jour où je me fais refuser une assurance à cause de données récoltés à mon insu ? Qui aura le pouvoir de contrôler les objets critiques, comme les feux de signalisation ou les capteurs des centrales nucléaires ? Ou simplement comment trouver un nouveau travail à toutes ces personnes dont les emplois ont été remplacés dans ce grand processus d'automatisation ?

De mon point de vue, on ne peut pas faire marche arrière. L'attitude de certains acteurs visant à décrédibiliser les technologies sous le couvert d'argument rétrogrades n'est pas soutenable. Une technologie ne doit pas être jugée à la lumière de ses implémentations, mais des améliorations qu'elle permet.

Et je pense que la *smart city* en est un exemple parfait, elle nous donne les bonnes armes pour s'assurer de construire l'avenir sereinement. La *smart governance* en particulier, me donne beaucoup d'espoir.

Mais d'un autre côté, je prends peur quand je vois le taux de participation aux dernières votations. Je me pose la question, est-ce que les citoyens en ont vraiment quelque chose à faire ? Vont-ils vraiment s'impliquer, eux qui ont déjà du mal à répondre à une question par oui ou par non ?

Peut-être préfèrent-ils simplement agir, faire des choses pour prendre leur avenir en main, plutôt que de répondre à ces questions de façon passive. Ou peut-être que ce qui ne va pas, c'est finalement juste le cadre dans lequel ces questions sont écrites et la manière dont elles sont posées...

[71] ARCORACI, Krystal. *Développer le concept de Smart Canton de Genève* [en ligne]. 2017.

Bibliographie

ABMUSHI. IOTA: MAM Eloquently Explained. *ABmushi* [en ligne]. 24 février 2018. [Consulté le 28 juin 2018]. Disponible à l'adresse : https://medium.com/@abmushi/iota-mam-eloquently-explained-d7505863b413.

ANDRE□. Tweet de @puellavulnerata. *Twitter* [en ligne]. 2 juillet 2018. [Consulté le 5 juillet 2018]. Disponible à l'adresse : https://twitter.com/puellavulnerata/status/1013906986717581313.

ARCORACI, Krystal. *Développer le concept de Smart Canton de Genève* [en ligne]. 2017. Haute école de gestion de Genève. Travail de Bachelor. [Consulté le 15 avril 2018]. Disponible à l'adresse : http://doc.rero.ch/record/306617?ln=fr.

ARNOLD, Jake. IOTA Partner and Affiliations. *Untangled World* [en ligne]. 4 novembre 2017. [Consulté le 9 juillet 2018]. Disponible à l'adresse : http://untangled.world/iota-partner-and-affiliations/.

BAQUIAST, Jean-Paul. Jeremy Rifkin. La nouvelle société du coût marginal zéro. *Club de Mediapart* [en ligne]. 7 octobre 2014. [Consulté le 4 juillet 2018]. Disponible à l'adresse : https://blogs.mediapart.fr/jean-paul-baquiast/blog/071014/jeremy-rifkin-la-nouvelle-societe-du-cout-marginal-zero-linternet-des-objets-lemergence-des.

BARAN, Paul. *Centralized Decentralized and Distributed Networks* [en ligne]. 1964. [Consulté le 6 juillet 2018]. Disponible à l'adresse : https://openclipart.org/detail/277506/Centralized-Decentralized-and-Distributed-Networks.

BILLET, Benjamin. *Système de gestion de flux pour l'Internet des objets intelligents* [en ligne]. 19 mars 2015. Université de Versailles-Saint Quentin en Yvelines. phdthesis. [Consulté le 3 juillet 2018]. Disponible à l'adresse : https://tel.archives-ouvertes.fr/tel-01166047/document.

BLOCKGEEKS. Basic Primer: Blockchain Consensus Protocol. *Blockgeeks* [en ligne]. 2018. [Consulté le 5 juillet 2018]. Disponible à l'adresse : https://blockgeeks.com/guides/blockchain-consensus/.

BUNTINE, Andrew. The Balanced Ternary Machines of Soviet Russia. *The Practical Dev* [en ligne]. 21 novembre 2016. [Consulté le 28 juin 2018]. Disponible à l'adresse : https://dev.to/buntine/the-balanced-ternary-machines-of-soviet-russia.

CARAGLIU, Andrea et al. Smart Cities in Europe. *Journal of Urban Technology*. 1 avril 2011. Vol. 18, n° 2, p. 65-82. DOI 10.1080/10630732.2011.601117.

DUGERDIL, Philippe. *Cours de Génie Logiciel*. 9 mars 2018.

EECKHOUT, Laetitia Van. « Le concept de "Smart City" n'est plus opérant ». *Le Monde* [en ligne]. 25 avril 2018. [Consulté le 4 juillet 2018]. Disponible à l'adresse : https://www.lemonde.fr/smart-cities/article/2018/04/25/le-concept-de-smart-city-n-est-plus-operant_5290389_4811534.html.

FEENSTRA, Evan. How is the tangle from IOTA quantum resistant? *Quora* [en ligne]. 29 novembre 2017. [Consulté le 5 juillet 2018]. Disponible à l'adresse : https://www.quora.com/How-is-the-tangle-from-IOTA-quantum-resistant.

FERNÁNDEZ-ARES, A. et al. Studying real traffic and mobility scenarios for a Smart City using a new monitoring and tracking system. *Future Generation Computer Systems*. 1 novembre 2017. Vol. 76, p. 163-179. DOI 10.1016/j.future.2016.11.021.

GAL, Alon. The Tangle: an illustrated introduction. *IOTA* [en ligne]. 28 février 2018. [Consulté le 28 juin 2018]. Disponible à l'adresse : https://blog.iota.org/the-tangle-an-illustrated-introduction-79f537b0a455.

GAUTIER, Philippe et GONZALEZ, Laurent. *L'Internet des objets : Internet, mais en mieux*. 2011. Paris, FR : AFNOR. ISBN 978-2-12-465316-4.

GOYAL, Saurabh. Centralized vs Decentralized vs Distributed. *Delta Exchange* [en ligne]. 1 juillet 2015. [Consulté le 4 juillet 2018]. Disponible à l'adresse : https://medium.com/delta-exchange/centralized-vs-decentralized-vs-distributed-41d92d463868.

HANDY, Paul. Introducing Masked Authenticated Messaging. *IOTA* [en ligne]. 4 novembre 2017. [Consulté le 7 juillet 2018]. Disponible à l'adresse : https://blog.iota.org/introducing-masked-authenticated-messaging-e55c1822d50e.

HEILMAN, Ethan et al. IOTA Vulnerability Report: Cryptanalysis of the Curl Hash Function Enabling Practical Signature Forgery Attacks on the IOTA Cryptocurrency. [en ligne]. 7 juin 2018. [Consulté le 6 juillet 2018]. Disponible à l'adresse : https://github.com/mit-dci/tangled-curl/blob/master/vuln-iota.md.

HOP, Eric. Debunking the 'IOTA Vulnerability Report'. *IOTA Demystified* [en ligne]. 25 février 2018. [Consulté le 5 juillet 2018]. Disponible à l'adresse : https://medium.com/iota-demystified/debunking-the-iota-vulnerability-report-c40fb07a6ae8.

IOTA FOUNDATION. Documentation. *iota.org* [en ligne]. 2018. [Consulté le 14 avril 2018]. Disponible à l'adresse : https://docs.iota.org/.

IOTA FOUNDATION. *IOTA Reference Implementation* [en ligne]. 24 février 2018. Java. IOTA. [Consulté le 7 juillet 2018]. Disponible à l'adresse : https://github.com/iotaledger/iri.

IOTA FOUNDATION. What is IOTA? [en ligne]. 2018. [Consulté le 8 juillet 2018]. Disponible à l'adresse : https://www.iota.org/get-started/what-is-iota.

IVANCHEGLO, Sergey. Come_from_Beyond comments on Integrity questions. [en ligne]. 1 janvier 2018. [Consulté le 7 juillet 2018]. Disponible à l'adresse : https://web.archive.org/web/20180101231015/https://www.reddit.com/r/Iota/comments/6yzm9g/integrity_question_for_come_from_beyond_sergey/dmsxaa5/.

IVANCHEGLO, Sergey. Drawing the line for "don't roll own crypto". *Come-from-Beyond* [en ligne]. 2 mars 2018. [Consulté le 5 juillet 2018]. Disponible à l'adresse : https://medium.com/@comefrombeyond/drawing-the-line-for-dont-roll-own-crypto-5d01c8525ca5.

IVANCHEGLO, Sergey. Tweet de @c___f___b. *Twitter* [en ligne]. 26 février 2018. [Consulté le 9 juillet 2018]. Disponible à l'adresse : https://twitter.com/c___f___b/status/968107528142770177.

JOHNSON, Nick. Why I find Iota deeply alarming. *Hacker Noon* [en ligne]. 26 septembre 2017. [Consulté le 28 juin 2018]. Disponible à l'adresse : https://hackernoon.com/why-i-find-iota-deeply-alarming-934f1908194b.

KORSGAARD, Henrik et BRYNSKOV, Martin. Prototyping a Smart City. [en ligne]. 2013. Aarhus Universitet. [Consulté le 15 avril 2018]. Disponible à l'adresse : http://pure.au.dk/portal/en/persons/martin-brynskov(b75c46ba-21ca-4810-8424-fcaac7a229b1)/publications/prototyping-a-smart-city(5b4855b9-7643-41d6-88f7-0a2e21443223).html.

LIMO. Full Emails of Ethan Heilman and the Digital Currency Initiative with the IOTA Team leaked. *The Tangler* [en ligne]. 24 février 2018. [Consulté le 9 juillet 2018]. Disponible à l'adresse : http://www.tangleblog.com/2018/02/24/full-emails-ethan-heilman-digital-currency-initiative-iota-team-leaked/.

MAGNE, Aleksi. Vers une approche de smart cities participatives. *Les Echos* [en ligne]. 12 juin 2017. [Consulté le 4 juillet 2018]. Disponible à l'adresse : https://www.lesechos.fr/idees-debats/cercle/cercle-170943-vers-une-approche-de-smart-cities-participatives-2093740.php#Xtor=AD-6000.

MAKINA CORPUS. Introduction à Node.js. [en ligne]. 29 janvier 2014. [Consulté le 8 juillet 2018]. Disponible à l'adresse : https://makina-corpus.com/blog/metier/2014/introduction-a-nodejs.

MASSET, Arnaud. Le bitcoin est-il bientôt prêt à concurrencer PayPal et Visa? *Le Temps* [en ligne]. 18 décembre 2017. [Consulté le 5 juillet 2018]. Disponible à l'adresse : https://www.letemps.ch/economie/bitcoin-estil-bientot-pret-concurrencer-paypal-visa.

MATTA, Natalie. *Vers une gestion décentralisée des données des réseaux de capteurs dans le contexte des smart grids* [en ligne]. 20 mars 2014. Université de technologie de Troyes. [Consulté le 3 juillet 2018]. Disponible à l'adresse : http://www.theses.fr/2014TROY0010.

MEOLA, Andrew. How smart cities & IoT will change our communities. *Business Insider* [en ligne]. 20 décembre 2016. [Consulté le 15 avril 2018]. Disponible à l'adresse : http://www.businessinsider.fr/us/internet-of-things-smart-cities-2016-10/.

QUEISSER DE STOCKALPER, Derek. La blockchain comme contrat social de la 4e révolution industrielle. *Le Temps* [en ligne]. 27 novembre 2017. [Consulté le 6 juillet 2018]. Disponible à l'adresse : https://www.letemps.ch/opinions/blockchain-contrat-social-4e-revolution-industrielle.

RIFKIN, Jeremy et al. *La nouvelle société du coût marginal zéro : L'internet des objets, l'émergence des communaux collaboratifs et l'éclipse du capitalisme*. 2014. Traduction française. Paris, FR : Les Liens qui libèrent. ISBN 979-10-209-0141-5.

ROCHAS, Audrey. *SmartLife : vivre avec les objets connectés*. 2016. Amilly, FR : Éditions Médicilline. ISBN 978-2-915220-84-1.

SCHNEIER, Bruce. Memo to the Amateur Cipher Designer. [en ligne]. 15 octobre 1998. [Consulté le 6 juillet 2018]. Disponible à l'adresse : https://www.schneier.com/crypto-gram/archives/1998/1015.html#cipherdesign.

SEMKO, Roman et SEMKO, Vitaly. *CarrIOTA Nelson, allows auto-discovery (P2P) for IOTA-Fullnodes* [en ligne]. 3 février 2018. JavaScript. SemkoDev. [Consulté le 7 juillet 2018]. Disponible à l'adresse : https://github.com/SemkoDev/nelson.cli.

SLADE, Samantha. The future is in business as commons. *TEDx Geneva* [en ligne]. 22 mars 2018. Genève, CH : . Disponible à l'adresse : https://www.youtube.com/watch?v=1qkhWa9XoFo.

SONM. Decentralized Computing on Ethereum Smart - Contracts is Gaining Momentum. *SONM* [en ligne]. 26 avril 2017. [Consulté le 2 juillet 2018]. Disponible à l'adresse : https://blog.sonm.io/decentralized-computing-is-gaining-momentum-c921f75e2a5c.

SØNSTEBØ, David. The Transparency Compendium. *IOTA* [en ligne]. 15 juin 2017. [Consulté le 28 juin 2018]. Disponible à l'adresse : https://blog.iota.org/the-transparency-compendium-26aa5bb8e260.

SØNSTEBØ, David. When is Ethereum going to run in to serious scaling issues? *reddit* [en ligne]. [Consulté le 8 juillet 2018]. Disponible à l'adresse : https://www.reddit.com/r/ethereum/comments/696iln/when_is_ethereum_going_to_run_in_to_serious/.

TANENBAUM, Andrew S. et STEEN, Maarten van. *Distributed systems: principles and paradigms*. 2007. 2nd ed. Upper Saddle RIiver, NJ : Pearson Prentice Hall. ISBN 978-0-13-239227-3. QA76.9.D5 T36 2007

TEOTIA, Siddharth et al. How does centralized and decentralized computing differ? *Quora* [en ligne]. 16 juin 2016. [Consulté le 2 juillet 2018]. Disponible à l'adresse : https://www.quora.com/How-does-centralized-and-decentralized-computing-differ.

UBER. Introducing UBER Movement | UBER. *YouTube* [en ligne]. 9 janvier 2017. [Consulté le 3 juillet 2018]. Disponible à l'adresse : https://www.youtube.com/watch?time_continue=5&v=bszvEIMVsIc.

UBER TECHNOLOGIES INC. Uber Movement: Let's find smarter ways forward. [en ligne]. [Consulté le 3 juillet 2018]. Disponible à l'adresse : https://movement.uber.com/?lang=en-US.

VERDON, Antoine. Pourquoi la blockchain augure la fin de l'Etat-nation. *Le Temps* [en ligne]. 11 septembre 2016. [Consulté le 6 juillet 2018]. Disponible à l'adresse : https://www.letemps.ch/economie/blockchain-augure-fin-letatnation.

WEIS, Steve. Tweet de @sweis. *Twitter* [en ligne]. 22 février 2018. [Consulté le 5 juillet 2018]. Disponible à l'adresse : https://twitter.com/sweis/status/966728742163791872.

Architecture orientée services. *Wikipédia* [en ligne]. 31 mai 2018. [Consulté le 3 juillet 2018]. Disponible à l'adresse : https://fr.wikipedia.org/w/index.php?title=Architecture_orient%C3%A9e_services&oldid=149093930.

Balanced ternary. *Wikipedia* [en ligne]. 21 mai 2018. [Consulté le 6 juillet 2018]. Disponible à l'adresse : https://en.wikipedia.org/w/index.php?title=Balanced_ternary&oldid=842218688.

Bitcoin Energy Consumption Index. *Digiconomist* [en ligne]. 2018. [Consulté le 6 juillet 2018]. Disponible à l'adresse : https://digiconomist.net/bitcoin-energy-consumption.

CarrIOTA Field. [en ligne]. [Consulté le 9 juillet 2018]. Disponible à l'adresse : https://field.carriota.com/.

Communs. *Wikipédia* [en ligne]. 27 juin 2018. [Consulté le 4 juillet 2018]. Disponible à l'adresse : https://fr.wikipedia.org/w/index.php?title=Communs&oldid=149876217.

Cryptomonnaie. *Wikipédia* [en ligne]. 25 juin 2018. [Consulté le 5 juillet 2018]. Disponible à l'adresse : https://fr.wikipedia.org/w/index.php?title=Cryptomonnaie&oldid=149835947.

Decentralized computing. *Wikipedia* [en ligne]. 17 mai 2018. [Consulté le 2 juillet 2018]. Disponible à l'adresse : https://en.wikipedia.org/w/index.php?title=Decentralized_computing&oldid=841738580.

Distributed computing. *Wikipedia* [en ligne]. 27 juin 2018. [Consulté le 2 juillet 2018]. Disponible à l'adresse : https://en.wikipedia.org/w/index.php?title=Distributed_computing&oldid=847802825.

Distributed ledger. *Wikipedia* [en ligne]. 18 juin 2018. [Consulté le 2 juillet 2018]. Disponible à l'adresse : https://en.wikipedia.org/w/index.php?title=Distributed_ledger&oldid=846351924.

Global Charts | CoinMarketCap. [en ligne]. [Consulté le 6 juillet 2018]. Disponible à l'adresse : https://coinmarketcap.com/charts/.

Graphe orienté acyclique. *Wikipédia* [en ligne]. 2 mai 2018. [Consulté le 6 juillet 2018]. Disponible à l'adresse : https://fr.wikipedia.org/w/index.php?title=Graphe_orient%C3%A9_acyclique&oldid=148071352.

Internet of things. *Wikipedia* [en ligne]. 29 juin 2018. [Consulté le 1 juillet 2018]. Disponible à l'adresse : https://en.wikipedia.org/w/index.php?title=Internet_of_things&oldid=848125659.

IOTA Full Node Copy-Paste Installation Guide. [en ligne]. 2018. [Consulté le 28 juin 2018]. Disponible à l'adresse : http://iota.partners/.

L'émergence du bitcoin et autres crypto-actifs : enjeux, risques et perspectives. *Banque de France* [en ligne]. 5 mars 2018. [Consulté le 6 juillet 2018]. Disponible à l'adresse : https://publications.banque-france.fr/lemergence-du-bitcoin-et-autres-crypto-actifs-enjeux-risques-et-perspectives.

Règlement général sur la protection des données [en ligne]. 4 mai 2016. [Consulté le 6 juillet 2018]. Disponible à l'adresse : http://data.europa.eu/eli/reg/2016/679/oj/fra.

Smart city. *Wikipedia* [en ligne]. 28 juin 2018. [Consulté le 1 juillet 2018]. Disponible à l'adresse : https://en.wikipedia.org/w/index.php?title=Smart_city&oldid=847846731.

What is JINN project? *Iota Stack Exchange* [en ligne]. [Consulté le 8 juillet 2018]. Disponible à l'adresse : https://iota.stackexchange.com/questions/344/what-is-jinn-project.

Annexes

Le projet entier est disponible sur GitHub, à l'adresse : https://github.com/riccardoricc/tb-prototype.

Annexe 1 : main.js

```
// Required libraries
const IOTA = require('iota.lib.js');
const SQLite3 = require('sqlite3');

// Required app components
const entryPoint = require('./entryPoint');
const rightsManager = require('./rightsManager');
const sensor = require('./sensor');
global.mam = require('./mam');
global.buyer = require('./buyer');
global.seller = require('./seller');

// Global APIs
global.iota = new IOTA({ provider: 'https://field.carriota.com/' });
global.db = new SQLite3.Database('log.db');

// Global variables
global.myPort = ***;
global.myName = '***' + myPort;
global.mySeed = '***';
global.minWeightMagnitude = 14;
global.depth = 3;

// Init app
exports.init = function () {
    iota.api.getNewAddress(mySeed, { 'returnAll': true }, function (error,
addresses) {
        if (error) {
            console.error(error);
        } else {
            global.addressIndex = addresses.length - 1;
            entryPoint.run();
            rightsManager.run();
            sensor.run();
        }
    });
}
```

Annexe 2 : entryPoint.js

```
// Required libraries
const express = require('express');
const bodyParser = require('body-parser');

// Init libraries
const app = express();
app.use(bodyParser.json());

// GET Infos
app.get('/info', function (request, response) {
    iota.api.getNodeInfo(function (error, infos) {
        if (error) {
            response.writeHead(500, { 'Content-Type': 'application/json' });
            response.write(error.toString());
        } else {
            response.writeHead(200, { 'Content-Type': 'application/json' });
            response.write(JSON.stringify(infos));
        }
        response.end();
    });
})

// POST Sale
app.post('/buy', function (request, response) {
    let thisIndex = addressIndex;
    addressIndex++;
    iota.api.getNewAddress(mySeed, { 'index': thisIndex }, function (error,
address) {
        if (error) {
            response.writeHead(500, { 'Content-Type': 'application/json' });
            response.write(error.toString());
        } else {
            // Attach new address to tangle
            iota.api.sendTransfer(mySeed, depth, minWeightMagnitude, [{
'address': address, 'value': 0 }], (error) => {if(error)
console.error(error)});
            let sale = request.body;
            sale.address = address;
            sale.value = sale.amount;
            console.log('Received sale:');
            console.log(sale);
            console.log('Saving...');
            db.run('INSERT INTO Sales VALUES (?,?,?,?,?,?,?);', sale.item,
sale.amount, sale.from, sale.to, sale.timestamp, sale.address, sale.value);
            console.log('Replying...');
            response.writeHead(202, { 'Content-Type': 'application/json' });
```

```
            response.write(JSON.stringify(sale));
            console.log('---------------------------------');
        }
        response.end();
    });
});

// POST Purchase
app.post('/sell', function (request, response) {
    let purchase = request.body;
    purchase.to = myName + '/post';
    console.log('Received purchase:');
    console.log(purchase);
    console.log('Saving...');
    db.run('INSERT INTO Purchases VALUES (?,?,?,?,?,?,?);', purchase.item,
purchase.amount, purchase.from, purchase.to, purchase.timestamp,
purchase.address, purchase.value);
    console.log('Replying...');
    response.writeHead(202, { 'Content-Type': 'application/json' });
    response.write(JSON.stringify(purchase));
    console.log(`Sending ${purchase.value} to ${purchase.address}...`);
    iota.api.sendTransfer(mySeed, depth, minWeightMagnitude, [{ 'address':
purchase.address, 'value': parseInt(purchase.value) }], function (error,
transaction) {
        if (error) {
            console.error(error);
        } else {
            let transactionHash = transaction[0].hash;
            console.log('Transaction sent:', transactionHash);
            console.log('---------------------------------');
            //console.log('Sending promotes...');
            //buyer.doPromotes(transactionHash, transactionHash);
        }
    });
    response.end();
});

// POST Stream
app.post('/post', function (request, response) {
    var subscription = request.body;
    console.log('Received subscription:');
    console.log(subscription);
    console.log('Saving...');
    db.run('INSERT INTO Subscriptions VALUES (?,?);', subscription.root,
subscription.amount);
    console.log('---------------------------------');
    response.writeHead(201);
    response.end();
});
```

```
// Run this component
exports.run = function () {
    app.listen(myPort);
    console.log('--------------------------------');
    console.log('Entry point running...');
    console.log('Index:', addressIndex);
    console.log('Listening on:', myName)
    console.log('--------------------------------');
}
```

Annexe 3 : rightsManager.js

```
// Required libraries
const Mam = require('mam.client.js');
const TimerJob = require('timerjobs').TimerJobs;
const request = require('request-promise');

// Periodic function
function manageRights(done) {
    // Get all inputs (balances) of the main seed ==> result
    iota.api.getInputs(mySeed, function (error, result) {
        if (error) {
            console.error(error);
            done();
        } else {
            // Get all pending sales stored
            db.all('SELECT * FROM Sales;', async function (error, sales) {
                if (error) {
                    console.error(error);
                    done();
                } else {
                    // For each pending sale
                    for (let sale of sales) {
                        // If an existing input corresponds to the sale
                        if (result.inputs.find(i => i.address == sale.address
&& i.balance == sale.value)) {
                            await sendAccess(sale);
                            console.log('--------------------------------');
                        }
                    }
                }
            });
            done();
        }
    });
}

function sendAccess(sale) {
    // Printing sale infos
    console.log('Sale confirmed:');
    console.log(sale);
    // Creating new stream with new random seed associated
    console.log('Creating new stream...');
    let mamState = Mam.init(iota);
    mamState = Mam.changeMode(mamState, 'private');
    // Sending stream to buyer
    console.log('Sending stream\'s root to buyer...');
    sale.root = Mam.getRoot(mamState)
```

```
    return request({
        url: sale.to,
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        json: sale,
        resolveWithFullResponse: true
    }).then(function (response) {
        console.log(`Status: ${response.statusCode} -
${response.statusMessage}`);
        console.log('Closing sale...');
        db.run('INSERT INTO Streams VALUES (?, ?, ?);', mamState.seed,
sale.amount, 0);
        db.run('DELETE FROM Sales WHERE "address" = ?;', sale.address);
    }).catch(function (response) {
        if (response.statusCode) {
            console.log(`Status: ${response.statusCode} -
${response.response.statusMessage}`);
            console.error(`Error ${response.statusCode}, keeping sale
stored...`);
        } else {
            console.error(response.error);
            console.error('No response, keeping sale stored...');
        }
    });
}

// Run this component
exports.run = function () {
    const timer = new TimerJob({ autoStart: true, interval: 5000, immediate:
true, ignoreErrors: true, infinite: true }, manageRights);
    console.log('Rights manager running...');
    console.log('--------------------------------');
}
```

Annexe 4 : sensor.js

```
// Required Libraries
const Mam = require('mam.client.js');
const TimerJob = require('timerjobs').TimerJobs;
const crypto = require('crypto');

// Periodic function
function publishData (done) {
    data = {
        item: 'something',
        value: crypto.randomBytes(1)[0],
        timestamp: Math.floor(Date.now() / 1000)
    }
    trytes = iota.utils.toTrytes(JSON.stringify(data));
    db.each('SELECT * FROM Streams;', function (error, stream) {
        if (error) {
            console.error(error);
            done();
        } else {
            let mamState = Mam.init(iota, stream.seed);
            mamState = Mam.changeMode(mamState, 'private');
            mamState.channel.start = stream.index;
            let message = Mam.create(mamState, trytes);
            if (stream.index < stream.amount) {
                db.run('UPDATE Streams SET "index" = ? WHERE "seed" = ?;',
stream.index+1, stream.seed);
            } else {
                db.run('DELETE FROM Streams WHERE "seed" = ?;', stream.seed);
            }
            Mam.attach(message.payload, message.address).then((result) => {
                console.log('Data published');
                console.log('Hash:', result[0].hash);
                console.log('Address:', result[0].address);
                console.log('Root:', message.root);
                console.log('------------------------------');
            });
        }
    });
    done();
}
// Run this component
exports.run = function () {
    const timer = new TimerJob({ autoStart: true, interval: 60000, immediate:
true, ignoreErrors: true, infinite: true }, publishData);
    console.log('Sensor running...');
    console.log('------------------------------');
}
```

Annexe 5 : buyer.js

```
const request = require('request');

exports.run = function () {

    const targetName = 'http://localhost:82';

    const purchase = {
        item: 'sometype',
        amount: 10,
        from: targetName + '/info',
        to: myName + '/post',
        timestamp: Math.floor(Date.now() / 1000)
    }

    request({
        url: targetName + '/buy',
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        json: purchase,
        resolveWithFullResponse: true
    }, function (error, response, body) {
        if (error) {
            console.error(error);
            console.error('No response, canceling...');
        } else {
            let purchase = body;
            console.log('-------------------------------');
            console.log('Request purchase:', purchase.from);
            console.log(`Status: ${response.statusCode} -
${response.statusMessage}`);
            if (response.statusCode < 200 || response.statusCode >= 300) {
                console.error(`Error ${response.statusCode}, canceling...`);
            } else {
                console.log('Purchase:');
                console.log(purchase);
                console.log('Saving...');
                db.run('INSERT INTO Purchases VALUES (?,?,?,?,?,?,?);',
purchase.item, purchase.amount, purchase.from, purchase.to,
purchase.timestamp, purchase.address, purchase.value);
                console.log('-------------------------------');
                console.log(`Sending ${purchase.value} to
${purchase.address}...`);
                iota.api.sendTransfer(mySeed, depth, minWeightMagnitude, [{
'address': purchase.address, 'value': parseInt(purchase.value) }], function
(error, transaction) {
                    if (error) {
```

```
                        console.error(error);
                    } else {
                        let transactionHash = transaction[0].hash;
                        console.log('Transaction sent:', transactionHash);
                        console.log('---------------------------------');
                        //console.log('Sending promotes...');
                        //doPromotes(transactionHash, transactionHash);
                    }
                });
            }
        }
    });

}
```

Annexe 6 : seller.js

```
const request = require('request');

exports.run = function () {

    const targetName = 'http://localhost:82';

    iota.api.getNewAddress(mySeed, { 'index': addressIndex }, function
(error, address) {
        const sale = {
            item: 'sometype',
            amount: 10,
            from: myName + '/info',
            timestamp: Math.floor(Date.now() / 1000),
            address: address,
            value: 10
        }

        request({
            url: targetName + '/sell',
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            json: sale,
            resolveWithFullResponse: true
        }, function (error, response, body) {
            if (error) {
                console.error(error);
                console.error('No response, canceling...');
            } else {
                let sale = body;
                console.log('--------------------------------');
                console.log('Request sale:', sale.to);
                console.log(`Status: ${response.statusCode} -
${response.statusMessage}`);
                if (response.statusCode < 200 && response.statusCode >= 300)
{
                    console.error(`Error ${response.statusCode},
canceling...`);
                } else {
                    console.log('Sale:');
                    console.log(sale);
                    console.log('Saving...');
                    db.run('INSERT INTO Sales VALUES (?,?,?,?,?,?,?);',
sale.item, sale.amount, sale.from, sale.to, sale.timestamp, sale.address,
sale.value);
                    console.log('--------------------------------');
                }
```

```
            }
        });
    });

}
```

Annexe 7 : mam.js

```
const Mam = require('mam.client.js');

exports.run = function () {
    let mamState = Mam.init(iota);
    mamState = Mam.changeMode(mamState, 'private');

    db.each('SELECT * FROM Subscriptions;', function (error, subscription) {
        Mam.fetch(subscription.root, 'private', null).then(function (stream)
{
            console.log('--------------------------------');
            console.log('Reading subscriptions...')
            console.log('--------------------------------');
            console.log('Subscription:')
            console.log(subscription);
            console.log('Values:')
            for (let message of stream.messages) {
                console.log(JSON.parse(iota.utils.fromTrytes(message)));
            }
            console.log('--------------------------------');
        });
    });
}
```