Securing industrial internet of things architectures through Blockchain

Valentin Vallois

Université Paris Cité

École doctorale EDITE ED 130

Laboratoire Centre Borelli ENS

Sécurisation des architectures d'objets connectés industriels à base de Blockchain

Securing industrial internet of things architectures through Blockchain

Par Valentin VALLOIS

Thèse de doctorat d'informatique

Dirigée par Ahmed MEHAOUA

Présentée et soutenue publiquement le 12/12/2022

Devant un jury composé de :

Lyes KHOUKHI, Professeur des universités, ENSICAEN, Ecole Nationale Supérieure d'Ingénieurs de Caen. Rapporteur

Raouf BOUTABA, Professeur des universités, Cheriton School of Computer Science, Université de Waterloo (Canada). Rapporteur

Ken CHEN, Professeur des universités, Laboratoire de Traitement et Transport de l'Information, Université Paris Nord. Examinateur

Ahmed SERHROUCHNI, Professeur, Laboratoire Traitement et Communication de l'Information (LTCI), Telecom Paris. Examinateur

Selma BOUMERDASSI, Maîtresse de Conférences (HDR), Centre d'études et de recherche en informatique et communication, CNAM. Examinatrice

Fouad GUENANE, Docteur ingénieur chez Capgemini. Membre invité

Ahmed MEHAOUA, Professeur des universités, Centre Borelli, Université Paris Cité. Directeur de thèse

Osman SALEM, Maître de Conférences (HDR), Centre Borelli, Université Paris Cité. Co-encadrant de thèse

Résumé en français du mémoire de thèse

Sécurisation des architectures d’objets connectés industriels à base de Blockchain

Introduction

Au cours des 3 années écoulées depuis le début de ce programme de doctorat, nous avons assisté à une évolution des connaissances et de la sensibilisation à l'Internet des objets et à la blockchain. Ces nouvelles technologies étaient autrefois l'objet de rêves, promettant de révolutionner l'informatique. Si IoT a apporté des avantages et des cas d'utilisation évidents et est à l'avant-garde de nombreuses innovations, blockchain n'a pas réussi à percer dans notre vie quotidienne. Avec ses nombreux inconvénients, les entreprises et organisations hésitent à l'intégrer dans leurs systèmes d'information. Pourtant, la mise en œuvre de ces technologies reste marginale en raison de leur complexité et de l'impact qu'elles ont sur les systèmes existants. Les organisations doivent changer leur mode de fonctionnement pour s'adapter à ces nouveaux composants. Les start-ups n'ont pas les mêmes difficultés que les grandes entreprises, il est plus facile de partir de zéro que d'intégrer ces nouvelles technologies dans les systèmes existants. Les entreprises doivent donc disposer d'outils de soutien et de conseil. Elles doivent d'abord prendre conscience des enjeux et des défis, puis déterminer si ces nouvelles solutions répondent à leurs besoins actuels.

Stratégie de recherche

Cette thèse porte sur la cybersécurité pour l'Internet des objets industriels. La criticité du domaine d'application et son ouverture aux technologies de l'Internet justifient la nécessité de cette recherche. Les attaques (Stuxnet, WannaCry, Shamon, etc.) qui ont été menées sur les ICS justifient l'investissement. Les méthodes et solutions de sécurité traditionnelles ne peuvent pas être intégrées telles quelles dans un environnement industriel ouvert à l'Internet. Les aspects opérationnels des ICS ne sont pas efficacement pris en compte par ces solutions classiques. Ainsi, l'intégration de protocoles cryptographiques traditionnels ne couvre pas les

besoins de l'IIoT, simplement en raison des problèmes de latence dans les différentes opérations de validation, notamment pour les certificats. Ces mêmes certificats supportent les services de sécurité requis.

Par conséquent, la définition de plusieurs exigences de sécurité s'inspire de méthodes existantes dans les domaines de l'informatique mais adaptées aux contraintes de la nouvelle génération de systèmes de contrôle industriel (IIoT).

Cette adaptation conserve les mêmes problèmes de sécurité connus dans les systèmes informatiques, mais qui exposent les systèmes industriels à des problèmes de sécurité plus coûteux, d'où l'objectif de la recherche proposée : définir et concevoir de nouvelles méthodes et mécanismes structurant ces architectures IIoT tout en préservant les aspects opérationnels mentionnés ci-dessus.

La définition d'un modèle de sécurité intégrant divers mécanismes de sécurité tels que l'authentification et l'autorisation ainsi que des mécanismes basés sur la cryptographie pour une plateforme d'objets hétérogènes dans un système IIoT représente l'axe central de cette recherche. La plateforme ainsi que les protocoles à définir doivent offrir la capacité d'interfacer des composants de différents environnements (Cloud, applications mobiles, environnements embarqués, capteurs avec des contraintes de consommation d'énergie) et la capacité de monter en charge de manière transparente par rapport aux exigences de sécurité et aux contraintes opérationnelles. La nature décentralisée de la Blockchain et des services de sécurité associés constitue une solution potentielle pour prendre en charge des services de sécurité IIoT à grande échelle. Il existe en effet une activité scientifique très importante avec des innovations pertinentes pour couvrir différents besoins de sécurité.

Plan du mémoire

Les dispositifs IoT, comme tout composant informatique, ont besoin de sécurité pour être intégrés dans un système informatique, en raison de leurs caractéristiques : diversité, ouverture, vivacité et faible capacité de calcul. L'intégration ne se fait pas sans effort et requiert une attention particulière de la part des organisations. Des améliorations des processus et des technologies sont nécessaires pour faciliter la démocratisation des systèmes IoT. Parmi les domaines à améliorer figurent les processus d'administration qui doivent être modifiés pour répondre aux nouveaux besoins des dispositifs IoT à l'intérieur d'un SCI. Une

gestion améliorée offre davantage de services de sécurité, et de nombreuses questions se posent autour de la surveillance et de la gestion des identités.

Le cloud computing est à la pointe de l'innovation pour les systèmes d'information. Le cloud offre des solutions natives pour gérer des systèmes complexes et fournit une gamme de services pour répondre aux besoins et aux défis de l'IoT industriel. Cependant, il est facile de se perdre devant les capacités des fournisseurs de services. Ce projet de recherche porte sur la spécification et la validation des exigences de sécurité dans une architecture Internet d'objets industriels. L'objectif est de définir et d'identifier les méthodes de sécurité intégrées par défaut dans la conception des architectures IIoT décentralisées (security by design). Ce travail doit prendre en compte les différentes technologies émergentes (Cloud, Fog, Mist) et les contraintes opérationnelles des SCI basés sur l'IIoT.

Nous avons consulté la littérature académique, en compilant les architectures de référence ainsi que les architectures présentes chez les fournisseurs de services. L'objectif est de déterminer l'architecture de haut niveau qui fournit les mesures de sécurité nécessaires pour l'IoT afin d'établir une base de recommandation aux clients potentiels d'une société de conseil. Nous avons dû résoudre les problèmes suivants :

- Comment comparer les architectures de référence, et quelle méthode de comparaison utiliser ?
- Quels critères de comparaison ont été utilisés, et comment ont-ils été choisis de manière pertinente ?

Proposition de mise en œuvre de la blockchain comme solution de sécurité pour un parc d'équipements industriels IoT. Suite à nos travaux et à l'état de l'art, nous avons conclu que la gestion des identités était un enjeu fort, notamment en prévision du nombre futur d'appareils connectés dans les années à venir. Nous devons concevoir une solution qui répond aux défis actuels et futurs d'un tel système.

- La blockchain est-elle une solution, et en quoi répond-elle aux besoins des entreprises ?
- Comment construire un système offrant ces nouveaux services de sécurité dans un système d'information existant ?

Comparaison des architectures de référence de l'internet des objets

La sécurité dès la conception doit être prise en compte tant au niveau de l'architecture que du déploiement du système. Il existe de nombreux cadres adaptés à différents contextes et technologies pour intégrer les systèmes IoT dans les systèmes d'information. L'IoT peut apporter de nombreuses vulnérabilités inconnues à un système d'information. Pour garantir une intégration conforme aux meilleures pratiques, les industries doivent évaluer la meilleure architecture pour leur système IoT. Dans ce chapitre, nous avons examiné 10 architectures de référence pour les systèmes IoT en comparant leurs capacités à répondre aux exigences de sécurité. L'un des services de sécurité les plus importants est la disponibilité, car les dispositifs IoT peuvent avoir des tâches critiques ou doivent rendre compte à un centre de commande en temps réel. Pour examiner et comparer les architectures de référence, nous avons utilisé la méthode de décision multicritères : Analytic Hierarchical Process (AHP), qui nous permet d'évaluer chaque capacité et de les comparer pour élire l'architecture la plus adaptée à un cas d'utilisation. En proposant un cadre d'atterrissage robuste pour les dispositifs IoT, nous assurons un périmètre de sécurité au niveau de l'architecture.

Dans cette étude, nous avons comparé 10 architectures, 5 définies comme des cadres et 5 définies comme des plateformes. Grâce à cette comparaison, nous avons également évalué les capacités des méthodes AHP en tant que modèles de classement multicritères. Cette comparaison a été faite dans le but de trouver la meilleure architecture qui couvre les besoins de sécurité pour la disponibilité.

Le résultat des méthodes AHP, fig. 19, montre que les architectures qui couvrent la plupart des besoins sont IBM, IoT-A, Azure et IIRA. La plupart des cadres du consortium et du groupe de recherche obtiennent un score élevé car ils ont un niveau d'abstraction plus élevé et couvrent un plus large spectre de recommandations, bien qu'ils n'approfondissent pas l'aspect technique de la mise en œuvre. L'architecture de WSO2 est la moins complète dans sa description et ne couvre donc pas toutes les exigences abordées par l'UIT. En revanche, les architectures des plates-formes présentent une approche plus terre à terre, car elles peuvent

étayer l'architecture par leur technologie réelle. Ainsi, elles montrent leurs capacités réelles à se conformer aux exigences pour assurer la disponibilité.

L'étude n'a pas pris en compte les critères suivants : la complexité de la mise en œuvre et de la gestion, le coût, l'évolutivité et la performance. Ces critères étaient hors du champ de cette comparaison, mais ils doivent être pris en compte afin de rendre la comparaison des architectures de plate-forme plus complète. L'utilisation de la méthode AHP s'est avérée fructueuse. Nous avons comparé facilement l'architecture et la méthode peut être étendue pour inclure d'autres critères sans complexifier le processus.

Gestion des identités et des accès basée sur la blockchain dans les systèmes IoT industriels.

Dans ce chapitre, nous allons nous concentrer sur deux des aspects de la gestion du système IoT : le premier est la gestion des identités et des accès (IAM) et le second est la prévention des attaques man-in-the-middle pendant les mises à jour du firmware. La croissance de l'IoT est inévitable ; la plupart des estimations font état d'environ 5,8 milliards d'appareils en 2020 [5], les appareils IoT seront congestionnés en raison du volume même des appareils, car nous devons identifier chaque appareil individuellement. Une infrastructure distribuée et complexe rend difficile une gestion efficace, mais [147] affirme que les systèmes de gestion centralisés sont trop coûteux pour les grands réseaux. Ainsi, ces dernières années, en raison des multiples avancées en matière de technologie distribuée (cloud computing, blockchain) et de l'ouverture de l'écosystème industriel avec l'émergence du paradigme de l'entreprise plateforme, les systèmes distribués sont au centre des préoccupations des entreprises lors de l'évolution ou de la mise en œuvre de nouveaux systèmes. Par exemple, des initiatives telles que le protocole ActivityPub pour les réseaux sociaux prouvent l'utilité des systèmes fédérés qui mettent en œuvre un IAM distribué [148] ou l'utilisation de la blockchain pour propager de manière sécurisée la mise à jour des firmwares.

L'organisation du chapitre restant est la suivante . Dans la section 2, nous présenterons un aperçu de l'IAM, de la mise à jour du firmware et des technologies blockchain. Dans la section 3, nous explorerons différentes approches pour les systèmes distribués basés sur la blockchain. Ensuite, dans la section 4, nous proposons une solution pour valider l'intégrité du firmware en utilisant la technologie blockchain dans un système IoT distribué.

Dans notre proposition, nous n'avons pas abordé la confidentialité à l'intérieur d'une transaction. Par exemple, un message peut être crypté avant d'être ajouté à la blockchain et

seul un participant au système d'information pourra le décrypter. Le cryptage n'est pas une implémentation triviale et prend en compte de multiples paramètres tels que l'échange de clés, le cryptage symétrique ou asymétrique, le stockage des clés... La fonction IAM pourrait être réalisée directement dans un smart contract, mais toutes les blockchains n'ont pas les capacités d'exécuter des smart contracts complexes ; nous avons choisi de proposer une solution agnostique. Une blockchain comme Bitcoin a des règles strictes pour son contrat intelligent natif, seul un ensemble limité de fonctions est disponible, tandis que les blockchains Ethereum ou Hyperledger offrent un langage de programmation complet de Turing. L'utilisation du contrat intelligent augmente la sécurité du système, garantit que l'exécution de la fonction IAM sera directement enregistrée sur la blockchain et réduit le nombre de composants nécessaires pour un cadre IAM. Nous avons présenté une mise en œuvre de la blockchain pour un système IAM distribué ainsi que les avantages et les inconvénients d'une telle technologie. La blockchain est un outil utile dans les scénarios où plusieurs actionnaires doivent rendre des comptes les uns aux autres, mais elle n'est pas une solution miracle. Il s'agit d'une réponse complexe à des besoins spécifiques. Notre solution utilise la blockchain comme un bus de messages pour transmettre les instructions IAM en toute sécurité dans plusieurs environnements.

Mise à jour sécurisée du firmware IoT grâce à la technologie de la blockchain

La solution proposée est un système de blockchain visant à fournir un environnement sécurisé pour la mise à jour du firmware des dispositifs IoT. Bien que les canaux de communication officiels soient fiables et sécurisés, à mesure que l'expansion des dispositifs IoT déployés augmente, l'attrait de l'injection de code malveillant le long de la chaîne d'approvisionnement augmente, par exemple en introduisant une porte dérobée dans une bibliothèque open-source utilisée dans un firmware ou en aval en usurpant ou en interceptant la mise à jour. Notre approche consiste à prévenir et à détecter les attaques Man-in-the-Middle pendant le transfert de la mise à jour. Par conséquent, notre système se décline en deux fonctions : La prévention en utilisant la cryptographie asymétrique et la blockchain. La détection en analysant le processus de comportement de la mise à jour et en détectant une potentielle mise à jour de

firmware trafiquée. Dans cette section, nous utilisons la nomenclature SUIT [159] établie par un groupe de travail IETF.

Au cours de notre enquête sur l'utilisation de la technologie blockchain pour les systèmes IIoT, nous avons démontré les avantages que la blockchain peut apporter à de tels environnements contraints. La blockchain offre l'immuabilité des données stockées sur la chaîne, ce qui permet l'horodatage, la disponibilité et l'intégrité des informations. Nous avons proposé une solution tirant parti de ces caractéristiques pour gérer l'identité et l'accès, où les dispositifs IoT prouvent leur identité en s'adressant à la blockchain et où les administrateurs stockent les politiques d'identité et d'accès sur la blockchain. Nous avons ensuite proposé une solution pour délivrer de manière sécurisée les mises à jour de firmware en utilisant la cryptographie inhérente à la technologie blockchain et les algorithmes d'apprentissage automatique pour détecter les attaques man-in-the-middle pendant le transfert.

Conclusion

Nous avons exploré une nouvelle piste de réflexion sur les possibilités de la blockchain pour améliorer la sécurité des IoT industriels. La solution proposée est un système de blockchain pour fournir un environnement sécurisé pour la mise à jour du firmware des dispositifs IoT. Bien que les canaux de communication formels soient fiables et sécurisés, à mesure que l'expansion des dispositifs IoT déployés augmente, l'attrait de l'injection de code malveillant le long de la chaîne d'approvisionnement augmente, par exemple en introduisant une porte dérobée dans une bibliothèque open-source utilisée dans un firmware ou en aval en usurpant ou en interceptant la mise à jour. Notre approche consiste à prévenir et à détecter les attaques de type "Man-in-the-Middle". Ainsi, notre système a deux fonctions : la prévention en utilisant la cryptographie asymétrique et la blockchain. La détection en analysant le processus de comportement de la mise à jour et en détectant une potentielle mise à jour de firmware corrompue.

La technologie blockchain offre de nouvelles possibilités pour sécuriser les systèmes d'information. Cependant, les contraintes sont des bloqueurs majeurs pour une démocratisation. La technologie blockchain est adaptée aux systèmes avec plusieurs acteurs où l'information doit être distribuée. Ainsi, les utilisations de la blockchain, bien que révolutionnaires dans ses concepts, ses applications dans le monde industriel restent des niches.

Titre : **Sécurisation des architectures d'objets connectés industriels à base de Blockchain**

Résumé :
Cela fait dix ans que la technologie blockchain a été créée. Cet amalgame de cryptographie et d'application peer to peer apporte de nombreuses innovations et services de sécurité au-delà des services financiers aux systèmes d'information ordinaires et offre de nouveaux cas d'utilisation pour les applications distribuées dans le contexte industriel. Pendant ce temps, l'IoT est devenu proéminent dans l'industrie comme la future révolution industrielle apportant de nouvelles applications mais ouvrant la voie à des vulnérabilités de sécurité.
Au cours de cette thèse, nous avons exploré les principaux problèmes auxquels est confronté l'Internet des objets. Nous avons étudié comment les fournisseurs de plates-formes IIOT abordent ces défis en comparant les mesures qu'ils ont mises en œuvre avec les recommandations de l'UIT en utilisant le processus analytique hiérarchique (AHP). Cette étude nous a permis d'identifier les domaines d'amélioration et les cas d'utilisation de la blockchain. La gestion des identités est un problème récurrent dans la littérature IIoT, nous proposons une approche de gestion des identités pour les systèmes distribués assistés par blockchain afin de garantir l'unicité des identités et l'intégrité de l'annuaire. Sur la base de ce travail, nous avons développé un système de distribution et de validation des mises à jour de micrologiciel sécurisé par blockchain et l'algorithme de machine learning Locality sensitive hashing (LSH).

Mots clefs :
IIoT, Blockchain, Sécurité

Title : **Securing industrial internet of things architectures through Blockchain**

Abstract :
It's been ten years since blockchain technology was created. This amalgam of cryptography and peer-to-peer application brings many innovations and securities services beyond financial services to regular information systems and offers new use cases for distributed applications in industrial context. Meanwhile, IoT became prominent in the industry as the future industrial revolution, bringing new applications but paving the way for security vulnerabilities.
During this thesis, we explored the main issues facing the Internet of Things. We studied how IIoT platform providers address these challenges by comparing the measures they have implemented with the ITU recommendations using the Analytic Hierarchical Process (AHP). This study allowed us to identify areas of improvement and use cases for the blockchain. Identity management is a recurring problem in the IIoT literature, and we propose an identity management approach for distributed systems assisted by blockchain to guarantee the uniqueness of identities and the integrity of the directory. From this work, we have developed a blockchain-secured firmware update distribution and validation system using the machine learning algorithm Locality Sensitive Hashing (LSH).

Keywords :
IIoT, Blockchain, Security

Acknowledgements

Je souhaite remercier mon directeur de thèse *Pr. Ahmed MEHAOUA*, qui a toujours été la pour me donner des conseil avisé et me soutenir dans les moment difficile que nous avons tous traversé ces dernières années. C'est grâce à lui qui depuis les cours de réseaux en licence que je suis intéressé au domaine de la sécurité et des réseaux.
Dr Osman SALEM pour son rôle de co-encadrant et sa pédagogie inoubliable, qui m'a transmis tant de connaissance.
Je remercie aussi *Dr Fouad GUENANE* pour m'avoir accompagné depuis le début de ma thèse Cifre et d'avoir eu confiance en moi. Je n'aurai jamais commencé et fini cette thèse sans lui.
J'exprime tous mes remerciements à l'ensemble des membres de mon jury qui ce sont rendu disponible malgré des contraintes de temps : *Pr. Lyes KHOUKHI, Pr. Raouf BOUTABA, Pr. Ken CHEN, Pr Ahmed SERHROUCHNI, Dr. Selma BOUMERDASSI.*

Je suis heureux d'avoir pu travailler avec les équipes de Beamap et de Sopra Steria pendant ma thèse. Notamment *Thierry MENDES* avec qui j'ai beaucoup échangé et brainstormé pendant nos pause du midi et *Eglantine GRANIER* qui m'a rendu de grand services et une fabuleuse partenaire de travail. Et enfin mes deux partenaires de stage avec qui tout a commencé *Ahmed YAKDHANE* et *Anis HAMZAOUI*.

Je tient à remercier ma mère *Véronique GENET* pour son soutient tout le long et pour toute les fois où elle me questionnait sur l'avancement de ma thèse. Et a mon frère *Basile VALLOIS* qui lui ne posait pas de question. Je remercie aussi mon amie *Marie LAROCHE* pour son support pendant le sprint final, ce n'était pas facile.

Table of contents

Table des illustrations

Introduction

During the 3 years since the start of this PhD program, we have seen an evolution of knowledge and awareness about the Internet of Things and blockchain. These new technologies were once the stuff of dreams, promising to revolutionize IT. While IoT has brought obvious benefits and use cases and is at the forefront of many innovations, blockchain has not been able to break through into our daily lives. With its many drawbacks, companies and organizations are hesitant to integrate it into their information systems. However, the implementation of these technologies remains marginal due to their complexity and the impact they have on legacy systems. Organizations need to change their operating mode to accommodate these new components. Start-ups do not have the same difficulties that big companies do; it is easier to start from scratch than to integrate these new technologies into existing systems. Organizations must therefore be provided with support and consulting tools. First, they must be aware of the stakes and challenges and then determine whether these new solutions meet their current needs.

Industrial Control Systems (ICS) are monitoring and control networks and systems designed to support industrial processes. These systems are used to monitor and control a wide range of processes and operations, such as gas and electricity distribution, water treatment, oil refining or rail transportation. Supervisory Control and Data Acquisition (SCADA) systems are the main subgroup of ICS. In recent years, ICS have undergone a significant transformation from isolated proprietary systems to open architectures and standard technologies that are highly interconnected with other corporate networks and the Internet. Today, ICS products are mainly based on standard computer systems, integrated in different devices such as routers or cable modems, and often use commonly available software. All this has led to cost reductions and ease of use and has enabled remote control and monitoring.

The new generation of control systems (Industry 4.0) aims to interconnect all the components of an industrial infrastructure such as machines, physical elements in all types of industries by making them connected objects (Cyber-Physical Systems) in order to provide more maintenance services, predictability, diagnostics and operational efficiency.

In order to achieve this goal, multiple technologies such as Cloud Computing, Fog, Mist, and Digital Twins have been incorporated into new architectures of control systems based on industrial connected objects. These new technologies allow any element in the system to become an actor in the system and to generate data flows that cross the different layers of the communication architecture.

The issue of cybersecurity in ICS has been the subject of several research and standardization efforts. This is due to the criticality of the fields of application of these systems and to the nature of the technologies used, which do not integrate security mechanisms. The main challenge in securing ICS has been to integrate security services in the protocols, systems, and software of ICS without impacting the main operational constraints such as high performance, reliability, and real-time. This new generation designed around the IIoT technology inherits the same issues but with a greater complexity due to the decentralization of control and orchestration of objects and the high distribution of systems. This is also amplified by the opening to Internet technologies and the new data flows generated by all this transformation.

Securing a system means ensuring the different security principles: integrity, confidentiality, availability, and non-repudiation of system data. In order to provide these different security principles, it will be a matter of providing and implementing platforms that are components of IIoT systems that incorporate authentication, authorization, cryptography, and access control mechanisms. Technologies such as Blockchain appear to provide services that approach the objective of this research. The constraints of operations in IIoT and the constraints induced by Blockchain remain to be studied. The methods to be defined and the mechanisms to be developed must consider the heterogeneity of the technologies used, the communication protocols, and the performances of the different entities.

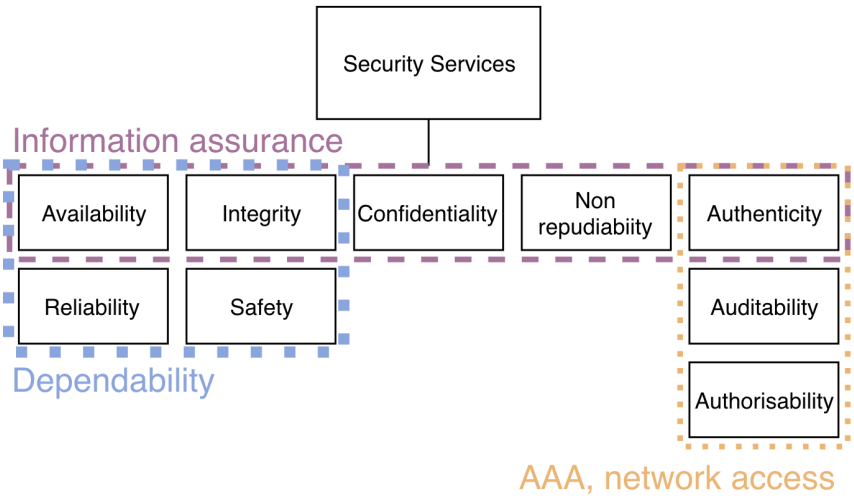

Fig. 1: Security services requirement for IoT

Security services relating to Information Assurance

Also known as the five pillars of information security, Information assurance [1] (Authenticity, Availability, Confidentiality, Integrity, Non-repudiability) extend the CIA (Confidentiality, Integrity, Availability) principle. The CIA triad is considered obsolete to characterize a system [2], therefore the principles of information assurance have been introduced.

Confidentiality allows users to exchange information securely and ensure the message isn't readable by anyone, making it only readable for authorized users. The goal is to ensure the privacy of the message transmitted, for this purpose several encryption algorithms should be used. Proper confidentiality brings trust between entities and prevents data leaks. In the IIoT context, confidentiality is essential when sensors collect sensitive data (personal data or production critical data).

Integrity is the protection against any intentional or unintentional alteration. Criminals will attempt to modify the data to conceal illegal activities or to send incorrect information to industrial systems that would prevent the plant from operating properly. But natural phenomena can also alter data; electromagnetic interference or packet loss can make transmission incomplete. Simple methods such as checksum and digital signature are used to ensure the integrity of messages without requiring large computations.

Availability is the assurance that information or components are accessible. Data availability is important for industrial systems; in a ubiquitous plant, to maintain optimized production, production data must be accessible for all control devices. In the same way, machines need to

communicate without interruption; if a stop message cannot reach a machine, disastrous events can occur. To prevent these scenarios, there are different solutions depending on the architecture. The network topology can be a mesh network or by making every route redundant. Firewall and anti-DDoS technologies can be deployed to protect the most vulnerable devices or applications.

Authenticity is the property validating the identity of actors (devices, users, applications) in an IIoT system. By using an authentication mechanism, entities can safely exchange messages and ensure that only the authorized and authenticated can send and read messages. It also uses mechanisms such as digital signatures to guarantee the identity of the sender.

Non-repudiability is the security property where an entity cannot deny sending or creating a message. This property ensures traceability of data in the system. Proper non-repudiation mechanism allows the detection of man-in-the-middle attacks and track back the source of incorrect messages.

Security service relating to Access Control

Access control (AAA) are three security principles working conjointly: Authenticity, auditability, and authorizability. AAA defines access control protocols extended to information systems [3].

Authenticity, known as authentication, is the property that validates the identities of entities. There are multiple processes to verify an identity, each depending on the infrastructure implementation or use cases. Once their identity is accepted by the process, the entities can interact with the system.

Auditability is the principle that every action, identification, or communication is recorded for traceability. Constant monitoring allows real-time security operations to be carried out while keeping logs allows forensic analysis after an incident.

Authorizability is identity control, the device, application, or user has its credentials controlled before giving it access to a resource or service. Depending on the context and the identity, the access can vary. For example, a web developer might have only access to the development environment.

Security service relating to Dependability

Dependability is a security service that characterizes the reliability of a service. In this discipline, there are four services: **availability** and **integrity** are the same security services from information assurance, but they extend to components of a system instead of only data. The availability of a service or device, as opposed to the availability of data, is important for systems that need to cooperate with each other. The integrity property is the guarantee that a system component is uncompromised and will function as intended.

Reliability in an industrial system is the rate of failure caused by incidents, malfunctions, bugs, etc. A 100% reliable system is a chimera; incidents will always occur and machines will break. Thus, building redundant systems by eliminating single points of failure (SPOF) through the use of techniques such as duplication of network routes or adoption of new topologies such as mesh networks can achieve a more reliable system. Designing the industrial system to be antifragile [4] or fault-tolerant will improve its reliability.

Safety is a property that characterizes the impact of a failure. It is the security of the peripheral elements of the system and the implementation of mitigation processes to minimize human, material, and financial risks. IIoT devices are an integral part of factory safety, and their ubiquity allows the implementation of contextual rules controlling machines that, for example, would prevent the operation of a mechanical arm if a human being is within its area of operation.

Research strategy

This thesis focuses on cybersecurity for the Internet of Industrial Objects. The criticality of the field of application and its openness to Internet technologies justifies the need for this research. The attacks (Stuxnet, WannaCry, Shamon, etc.) that have been carried out on ICS justify the investment. Traditional security methods and solutions cannot be integrated as such in an industrial environment open to the Internet. The operational aspects of ICS are not effectively considered by these classical solutions. Thus, the integration of traditional cryptographic protocols does not cover the requirements of the IIoT, merely due to latency issues in the various validation operations, particularly for certificates. These same certificates support the required security services.

Consequently, the definition of several security requirements is inspired by existing methods in the fields of information technology but adapted to the constraints of the new generation of industrial control systems (IIoT).

This adaptation keeps the same security problems known in IT systems, but which expose industrial systems to more expensive security problems, hence the objective of the proposed research: to define and design new methods and mechanisms structuring these IIoT architectures while preserving the operational aspects mentioned above.

The definition of a security model integrating various security mechanisms such as authentication and authorization as well as cryptography-based mechanisms for a platform of heterogeneous objects in a IIoT system represents the central axis of this research. The platform as well as the protocols to be defined must provide the ability to interface components from different environments (Cloud, mobile applications, embedded environments, sensors with energy consumption constraints) and the ability to scale up in a transparent manner with respect to security requirements and operational constraints. The decentralized nature of the Blockchain and associated security services is a potential solution for supporting large-scale IIoT security services. There is indeed a very large scientific activity with relevant innovations to cover different security needs.

Thesis outline

IoT devices, like any IT component, need security to be integrated in an IT system, due to their characteristics: diversity, openness, liveliness, and low compute capacity. Integration is not effortless and requires special attention from organizations. Improvements in processes and technology are necessary to facilitate the democratization of IoT systems. Among the areas for improvement are administration processes that need to be changed to accommodate the new needs of IoT devices inside an ICS. Improved management offers more security services, and there are numerous issues around monitoring and identity management.

Cloud computing is at the forefront of innovation for information systems. The cloud offers native solutions to manage complex systems and provides a range of services to meet the needs and challenges of industrial IoT. However, it is easy to get lost in front of the capabilities of the service providers. This research project deals with the specification and validation of security requirements in an Internet architecture of industrial objects. The objective is to define and identify security methods integrated by default in the design of

decentralized IIoT architectures (security by design). This work must consider the various emerging technologies (Cloud, Fog, Mist) and the operational constraints of ICS based on IIoT.

We consulted the academic literature, compiling reference architectures as well as architectures present in service providers. The objective is to determine the high-level architecture that provides the necessary security measures for IoT in order to establish a basis for recommendation to potential clients of a consulting firm. We had to solve the following problems:

- How to compare the reference architectures, and which comparison method to use?
- Which comparison criteria were used, and how were they chosen in a relevant way?

Implementation proposal of blockchain as a security solution for an IoT industrial equipment park. Following our work and the state of the art, we concluded that identity management was a strong challenge, especially in anticipation of the future number of devices connected in the coming years. We need to design a solution that meets the present and future challenges of such a system.

- Is the blockchain a solution, and how does it meet the needs of companies?
- How to build a system offering these new security services in an existing information system?

State of the Art

Industrial IoT

Introduction

In the last decade, the rise of the Internet of Things (IoT) has connected many smart devices to the Internet [5]. The integration of IoT devices has initiated the fourth industrial revolution [6]. The concept of IoT isn't new for industries; they have already used sensors linked to Supervisory Control and Data Acquisition (SCADA) systems for monitoring production chains and Cyber-Physical Systems (CPS). However, SCADA systems are subject to technical limitations [7], as they use proprietary protocols that make it difficult to integrate with modern technology. Maintaining these legacy systems is expensive, and replacing them requires a significant investment. CPS systems are designed for specific use cases, making them expensive and complex to upgrade. On the other hand, IoT devices are off-the-shelf solutions that are cheap, use well-known protocols, and operate on the internet or Ethernet. They are also easily replaced. However, these devices often lack good security measures because they have constrained resources, computational power, or energy. In addition, the use of different protocols in IoT systems makes them lack standardization. As a result, IoT devices are usually the weakest links in IoT systems and require special precautions when integrating them into an information system.

There are several ways to describe the Internet of Things (IoT) and the Industrial Internet of Things (IIoT). The NIST released a publication [8] showing a convergence of the definitions present in the academic literature of the Cyber-Physical Systems and the Internet of Things (fig. 1). The Industrial Internet of Things is at the point of convergence between these two definitions. Serpanos and Wolf [9] argue that IIoT systems require more consideration of safety and continuous operation when designing these systems, as a potential failure can result in significant financial loss or a life-threatening situation.

IoT systems are used in a variety of domains, including Industry 4.0, logistics, retail, smart grids, and agriculture. IIoT systems are a subset of IoT systems that are used in these domains. IoT devices are the bridge between the physical and digital worlds. The major distinction between an IIoT device and an IoT device is whether the device is part of a product used by a consumer. For example, a geolocalized electric scooter is an IoT device, but an autonomous robot in a warehouse [10] is an IIoT device. IIoT devices in an industrial system enable ubiquitous monitoring, which is essential for logistics, smart grids, and agriculture.

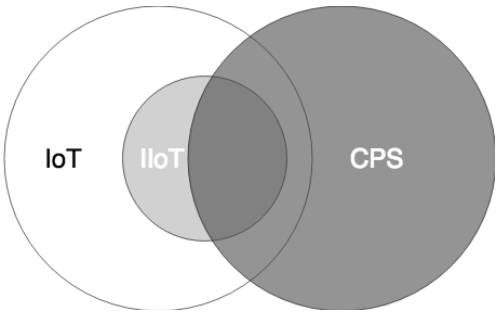

Fig. 2: IoT, IIoT and CPS in Venn diagram, inspired from [11]

Security is a major concern for any system. Spathoulas and Katsikas [12] argue that securing the IIoT involves a trade-off between security and availability. A data leak may not damage a machine, but if a crucial sensor detecting an anomaly does not transmit the data, it can result in significant financial loss or even a life-threatening situation. The heterogeneity of IIoT devices increases the complexity of designing a cohesive system. Integration with existing equipment is particularly problematic, especially if the equipment is highly customized to meet production needs, as it may not be compatible with recent network protocols or lack up-to-date security mechanisms.

This state of the art is organized as follows: first, we introduce the layered reference architecture of an IIoT system and the paradigm of the platform enterprise. In a second part, we describe the security services into three categories: information assurance, dependability, and access control. We then present the cyber-security threats for the Industrial Internet of Things system for each layer of the reference architecture.

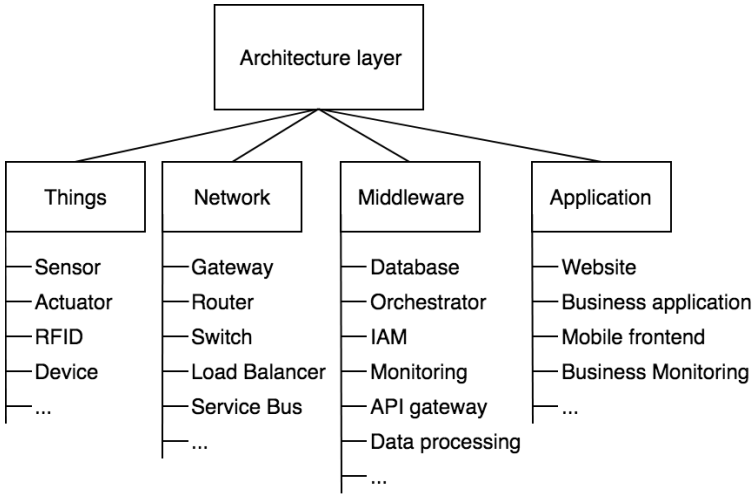

Fig. 3: IoT Reference Architecture

Architecture model

Security risks can be avoided by following a proven architecture model. Conceptually, IoT infrastructures follow a layered architecture. Depending on the level of granularity, the models include between three and five layers [13]–[15]. (A) : The three-layer model [16]–[24] , (B) : the four-layer model [22], [25]–[30], (C) : and the five-layer model [31]–[34]. are the most common patterns. We use the three-layer model to define the types of layers as follows: Object or Perception, Network, and Application. Figure 4 shows a comparison of the different architectures and the corresponding layers."

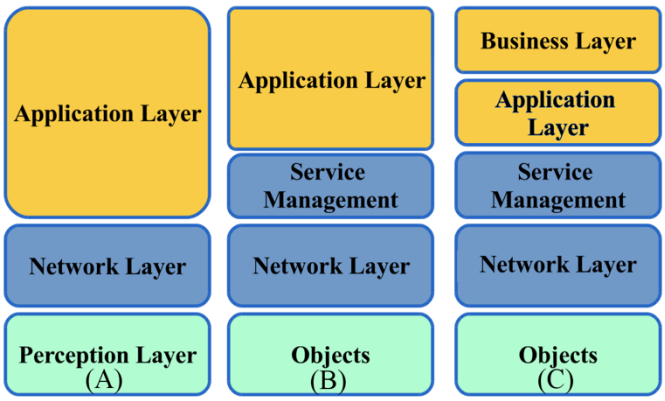

Fig. 4: The layered architectures based on [35]

The Object layer, first layer, is the symbolization of the logical space where the IoT devices are located. Communication with the upper layer is usually done through a gateway centralizing the IoTs. It is common on all the architectures and it's a layer of the Object type. The Network layer, second layer, is a transition layer between the Object and the Application

types layers. It includes all the network infrastructure and connects the IoTs to the other components of the system. This layer is also common through all the architectures and it's a layer of the Network type, but some of its components can be different as it might not include the network management, subsequently relayed to the upper layers. The Service Management layer, third layer, contains the administration and orchestration tools used to manage the system. It includes network management, application management, API management, and so on. This layer is mostly composed of middleware solutions ensuring the wellbeing of the system and it's a layer of the Network type. The Application layer, the fourth layer, is where data storage and processing are located. Moreover, it includes the many applications needed by the system, such as web services, API, etc. It's through this layer that end users interact with devices or the system. This layer is of the Application type. The Business layer, the fifth and last one, also defined as the administration layer, is the level where data collected on the system are processed and visualized. The management of the system, not the IoT, is found in this layer. It's of the Application type.

According to [36], a (B) architecture is the most suitable model for an abstract representation of an IoT architecture. Therefore, we decide to match the corresponding layer with the proposed model by the ITU [37]. This architecture, in fig. 5, is defined by: The Device layer, that is composed of 2 elements: the device and the gateway. The device can be an actuator, in which case its role is to perform an action in response to a received command. The device can act as a sensor whose purpose is to collect information about its environment and transmit it to the upper layers. Connections to networks can be direct or indirect through a gateway, for example. A device may be able to connect ad-hoc for scenarios requiring scalability and rapid deployment. Finally, a device can have the ability to put it into standby and wake up in order to save energy. The gateway must be connected to many devices for this reason it must be compatible with a maximum of communication protocol whether it is wired (ethernet, internet) or wireless (e.g., Zigbee, Wifi, 2/3G, Bluetooth...). Its role may be to translate communication protocols to allow exchanges between devices or between devices and services situated in the upper layers of the architecture. The Network layer represents 2 principles: Network capabilities, including connection control, access control, resource control, authentication, authorization and traceability (AAA). Transport capacities including connectivity with IoT services, data transport, and IoT control. The Service Support and Application Support layer includes data processing applications, data storage, and specific applications depending on the use cases. The services present in this layer are intended to be

used by IoTs, users, and other services present in this layer, they are mainly known as middleware services. The Application layer includes applications for end users, but also for data visualization, control applications. This architecture includes 2 verticals called capabilities. The Management Capabilities are management of the system, covering devices management, network management, traffic management, and specific application management. The Security Capabilities follow the pillar of security for the generic aspect, confidentiality, integrity, authentication, availability, and non-repudiation. Specific security capability must be applied in certain scenarios like mobile payment.

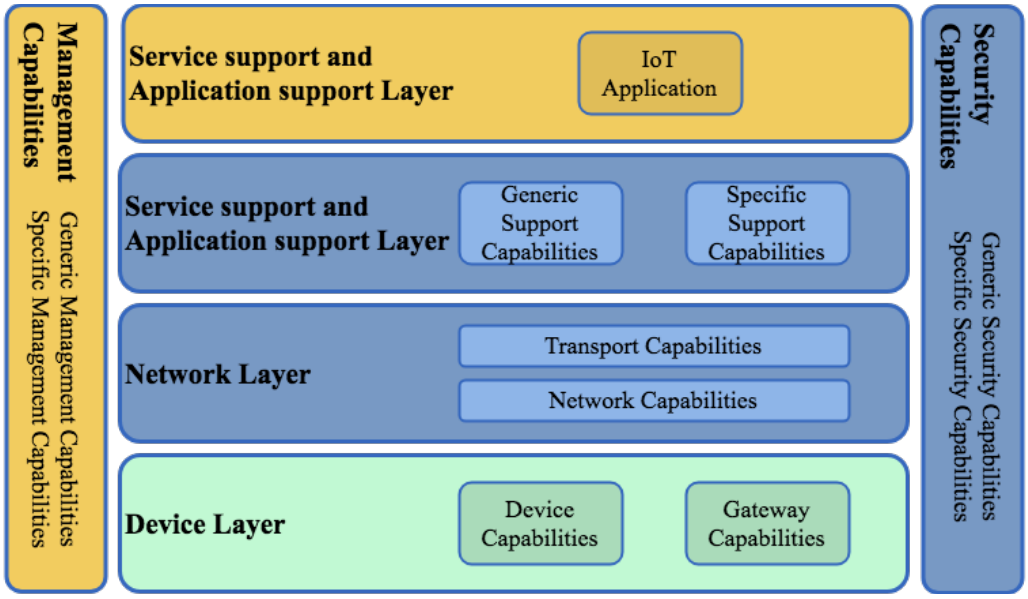

Fig. 5: ITU Reference architecture

Challenges

The interconnectivity of people, devices, and organizations in today's digital world opens up a whole new field of vulnerabilities and access points where cybercriminals can operate. Today, the landscape of risk for organizations is a combination of real and potential threats that come from unexpected and unforeseen actors with unpredictable consequences. In this post-economic crisis world, companies are evolving rapidly. New product launches, mergers, acquisitions, market expansion, and the introduction of new technologies are all on the rise: these changes invariably have a complicated impact on the strength and scale of an organization's cybersecurity and its ability to keep pace. Therefore, in this thesis, we will cover the current cybersecurity challenges of industrial information systems.

Physical limitations

Physical limitations are the physical capabilities of a computer system, they can be characterized by computing power, available memory, battery life, environment, distance latency, etc. An IoT system is built around these constraints, and each function will consume these physical capabilities. In addition to addressing use cases, adding security increases the consumption of its resources, needing more computational power that consumes more energy. We can distinguish IIoT devices into 2 categories: power-supplied devices and battery-powered devices. In the first case, the device has fewer restrictions and allows the constructor to design it with more computational power and capabilities. In the second category, the device is designed to be energy efficient. The lifespan for this kind of device can reach many years in most extreme cases. Battery-powered devices are generally used in places that are difficult to access. The common behavior for a sensor is to wake up, power on, acquire data, send it, and go back into sleep mode. Such processes need to use as little battery life as possible, and as stated earlier, adding any form of security (encryption, signature...) will decrease the device longevity. The evolution of the communication protocols improves energy consumption and allows the use of simple cryptographic algorithms. Nowadays, Elliptic Curves are proven to be more efficient than RSA [38] for the same key size. Data encryption for IoT is improving over the years, but the key exchange protocols are still a challenge [39]. IoT devices are constrained by their storage capacity, which requires them to transfer data to a data processing center. The devices must be able to send reliably over the network. This is a problem if they are not designed with sufficient means of communication, or in the case of a wireless network with enough power to reach the receiving antenna. Designing IoT systems according to these technical constraints is a matter of compromising functionality, cost, and security.

Heterogeneity

In the manufacturing sector, there are many highly specific components and machines built by a vast array of companies. Some are stand-alone and some are part of bundle solutions, in either case, they might be designed with features that make them difficult to integrate into an industrial information system. IoT devices aren't exempt from these shortcomings. There are many standards for communication between devices: RFID (e.g., ISO 18000 6c EPC class 1 Gen2), NFC, IEEE 802.11 (WLAN), IEEE 802.15.4 (ZigBee), IEEE 802.15.1 (Bluetooth),

Multihop Wireless Sensor/Mesh Networks, IETF Low power Wireless Personal Area Networks (6LoWPAN), Machine to Machine (M2M), IP, IPv6, etc [26]. This multitude of standards challenges IoT device designers, adding compatibility to all of them increases complexity, cost, and energy consumption for the IoT. While an Arduino board might include most of these protocols, it is several orders of magnitude more powerful than a simple sensor. Examining further than the network layer, in the application layer, the devices may use proprietary protocols or they are only compatible with a restricted set of appliances or applications. For example, the data collected are formatted according to specific criteria, which makes them easily readable by some data processing software and not by others. A high level of heterogeneity widens the attack surface for the whole system [40]. Keeping track of all vulnerabilities of each chosen solution is the challenge hindering the deployment of Industrial Internet of Things devices. Interoperability is the key factor for the adoption of IoT by industrials.

Authentication and Identity

Identity management is the combination of assigning identity to each device or user of the systems, defining their permission and validating their identity through the authentication process. Identity is essential for managing a potentially large number of components, potentially billions. Authentication is a critical security service for protecting against attacks such as identity theft or unauthorized access to services or data. Identity management becomes challenging when the number of identified entities increases over time. Moreover, the assignment of identities is difficult in environments where actors can come and go unexpectedly. Administrators must ensure that each actor is unique and there is no usurpation. The assignment of device identities requires an Identity Provider (IdP) that will create, maintain, and manage the identity. The IdP is a piece of software connected to the company information system and the Access Control management. Each IoT device may have a unique identifier assigned depending on its context of operation. The use of certificates provides a way to provide identities, but its management is complicated for distributed systems, as Won and others discussed in their research [41], and we see research and effort on building new methods to identify IoT devices [42], [43]. Identity management remains manageable as long as it remains in a specific environment, such as a factory, farm, or vehicle, but in an open environment like a smart city where enthusiastic weather stations can freely participate in the systems, or in a distributed system like a supply chain with multiple actors, the administration

is a challenge. An attacker who manages to gain access to the Identity and Access Management can compromise all security services of the system, thereby compromising data integrity, breaking confidentiality, and so on.

Identity is nothing without authentication. Means of controlling identities, such as certificates or multi-factor authentication, are needed. Authentication is divided into two categories, for users and for devices [44]. Authentication by username/password is the most commonly used method, but it is difficult to implement in some contexts, for example, when directly connecting to a sensor, the latter may not have the technical capabilities (connection to an external authentication server, internal software...) to perform the user's authentication. In addition, in a supply chain case, a device can communicate with several information systems of different companies and must be able to be identified in each. Access control and identity management is the first step to ensure accountability on your system, especially for industrial IoT systems.

Access control is the control of the permission for each identity. Depending on their identity, a panel of actions or resources are available to them. The definition of permissions is a challenge in itself in information systems, traditionally an administrator needs to keep track of every permission attributed to every user and must ensure that when a user leaves, his or her permissions are removed. The IoT context increases the complexity of this management as they are more dynamic systems, in constant evolution where devices come and go on the network. One of the challenges is also to anticipate new cases, where a device would need authorizations not initially planned, which poses the problem of automatically assigning permissions. Whatever the permission management mechanism used, whether it is an access control dictionary (DAC), access defined by their roles (RBAC) or attributes (ABAC), they operate on the principle of access control list which requires a prior definition of the permissions [2]. The propagation of this information is a challenge in distributed networks where synchronization may have latency and each component needs to adjust their behavior in accordance with the new rules. Devices might need to function in isolation without access to a central control unit, so they require other means of authentication. In addition, the access control center must be particularly well-protected, as once it is compromised, it would open the door to all types of attacks that would then be undetectable.

Availability and reliability

In the manufacturing world, some security characteristics are more important than others. Availability and resilience are among the main requirements for industrial sites [45]. Indeed, availability guarantees accessibility to and between services and devices. Resilience is the implementation of countermeasures or redundancy to ensure the continuous operation of the system. However, IoT devices are not known for their durability. By their simplicity they tend to be fragile and easily malfunction. These devices are very sensitive to cyber-attacks as well as physical attacks, their low computing capabilities do not give them the ability to defend themselves effectively. They are primarily designed for their functionality (e.g., a sensor must collect the data and transmit it) and security is often overlooked or considered an afterthought.

The unavailability of a device or data may lead to serious human, material or financial damage. A message that does not reach its destination may prevent the safety measurement from being activated, for example a temperature sensor that does not transmit an overheating in a foundry. The origin of an unavailability can be multiple, ranging from a DDoS attack to a physical interference. Detecting the cause of such incidents is a real challenge for today's industrial systems. A failure is inevitable, and therefore it is necessary to set up resilience mechanisms. It encompasses different concepts such as dependability, fault tolerance, robustness [46], Laprie in [47] defines resilience as “The persistence of dependability when facing changes.” The resilience of the components ensures the security and transmission of information even in hostile environments. However, the implementation of redundancy does not come without financial cost and in addition to complicating the management of the industrial IoT system. The management of the IoT devices also presents its own challenges. This starts with the design of the architecture, the choice of management software, deployment patterns, and operational recovery processes. Each of these steps needs to be carefully planned and tested, which goes against recent trends in the development cycle of technology solutions.

Maintainability

The maintenance of IoT devices poses several challenges, even though they are inexpensive and can be easily replaced. Some are sensitive to their environment, and subtle changes can render them inoperative. First, physical access: devices are often positioned [48]–[50] in places that may be difficult to access, geographically dispersed, or simply inaccessible. In

such cases, physical maintenance needs to be as little as possible, and software maintenance needs to be done remotely. In addition, updating software in a distributed environment is a challenge. Forcing the system to keep entities at the obsolete security level causes a high security risk, especially when the devices are too simple and cannot be updated. An update of a device can change the behavior of certain features, making it incompatible with the rest of the system. An additional issue is the responsibility for an unpatched vulnerability and determining who is accountable for updating. Industrial systems are complex, and the digital transformation allows more interactions between manufacturers through exposed APIs or remote access. Therefore, an infected system in company A could infect the system in company B. In any case, ensuring the right version is a complex task that requires remote access and awareness of the devices connected to your network.

To achieve maintainability, visibility of the devices fleet is crucial, yet difficult [51]. Thus, equipment for discovering devices to map the network is needed. The IoT devices must be selected to meet this requirement. The discovery of the devices connected to the system is particularly complex when the components can come and go, and due to the liveliness of the IoT environment. Future predictions [52] advance that industrial IoT systems will be self-aware and self-maintained by monitoring their health and detecting anomalies. This allows them to reconfigure their interactions and operations to accommodate a defective machine. Edge computing and artificial intelligence are the enabling technologies for self-maintenance.

Integrity

IoT devices operate in highly hostile environments. However, the information they send can be vital. An interception and modification of a message can cause damage just as serious as the activity of a malicious component. The functioning of an IoT system depends on the integrity of the data as well as the integrity of its components. Aman & al state two challenges of data integrity [53]: first, at the network level, protecting data integrity costs a high amount of energy and has many overheads for calculation. Secondly, most protection methods assume that the device isn't physically attackable and storing secret keys on the device is safe. Messages are transmitted through two types of environments: the Internet or the company network. The difference between these two environments is trust. The Internet is said to be trustless, while an enterprise network is trustful. On the Internet, data needs to be encrypted with VPN, IPSec or TLS during transit for keeping confidentiality, and IoT devices often lack the computing capacity to encrypt using strong cryptographic parameters. To verify the

integrity, mechanisms such as MAC, checksum algorithms, PKI, or even blockchain [54] can be used.

Integrity also includes the integrity of devices, software, and operating systems. Hardware can be compromised and then become under the control of a hacker, who will exploit the device's permissions to force harmful behavior, such as during the attack against DNS Dyn where an army of IoT devices such as IP cameras, printers, and other connected consumers performed a DDoS attack impacting the global Internet [55]. No antivirus or anti-malware is installed on machines yet they are hacked [56]. The implementation of countermeasures must be a coordinated action between the IoT equipment supplier and the manufacturer. The lifecycle management of IoT devices poses many challenges. First, there is the supply chain problem; it is necessary to ensure the device is not compromised during its fabrication, transport, and installation [57]. At the end of their life, the devices must be destroyed safely; for example, a server not properly erased will leak its data to the new purchaser, or a small device will keep its network information or store its credentials that will serve as intelligence for a future attacker.

Observability

Observability [58] extends the notion of monitoring a system in the form of three pillars [59]: traceability, logging, and metrics. Traceability is the ability to know an action or message's information through metadata, network observation, or direct communication of components. Logging is the ability to save action records in a format processable by big data platforms. Logging, coupled with traceability, enables forensic analysis. Finally, metrics are the data collected by monitoring. The metrics depend on the service provided by the system or devices and must be chosen carefully. Observability allows you to view an information system, react to an attack, prevent an attack, and consolidate security measures. A mature observable system can respond to any request required to verify the status of its components. The major benefit of observability is from a business perspective; it provides information on the capacity of its systems and thus enables decisions such as production adjustments or changes in safety rules to be taken.

Two things are monitored: the health of the systems and the production environment. The first is the monitoring of connected devices, network status, application traffic, applications, and users in order to detect malfunctions and suspicious activities. In the second case, the

monitoring is at a macro level of the system and takes information measured by IoT sensors, giving, for example, the production status of a smart factory.

For the industrial sector, the implementation of this type of infrastructure is complicated mainly due to the heterogeneity of IoT and the hostile environment in which IoT equipment operates. There may be discrepancies between the data collected by the multiple sensors, the information collected may be incomplete or corrupt [46] once it reaches the user. Thus, to produce quality data, it must come from multiple sources of information, which implies the production of a large volume of data requiring high computing power to be processed efficiently.

CyberAttacks

Every system is vulnerable and industrial systems aren't exempt from weakness. Attackers don't break defense but exploit these weaknesses. Nonetheless, security measures are to be deployed to reduce the potential attack surface. To help architects designing Industrial Internet of Things systems, we propose a taxonomy of attacks on IIoT systems through a layer approach. Industrials have specific requirements that differ from consumer needs [60], [61]. Consumer IoT devices are more sensitive to privacy issues because they collect personal data on the end user. Reliability and availability will depend on the applications, but most of the time they are not critical for the consumer. Meanwhile for IIoT systems, the concern will not be on privacy but on the confidentiality of the data, and reliability and availability are essential for most use cases.

We categorized the attack into four categories based on the affected component level in the reference architecture.

Threats on the Thing layer

- **Physical access**: Data or programs of an IoT device can be compromised if an attacker can physically connect to it. Critical systems should reduce the number of physical I/O to reduce voluntary or involuntary attacks like the Stuxnet case [62]. Furthermore, devices can be damaged by malicious people to malfunction, causing the sensor to send distorted data and the actuator to not follow instructions correctly.
- **Compromised hardware**: Companies trust their suppliers, but they can be the Achilles heel of their security. Even with major investment, the security level of all the supply chains needs to be identical for all the actors [63], [64]. In 2018, Bloomberg

published a story about compromised hardware during the supply chain [57], although the article is strongly criticized for its veracity, the scenario was plausible enough to cause a drop of 40% in the value of Super Micro [65]. This attack causes a breach in the confidentiality of the data collected by the IoT devices. A compromised device can be used to perform simple instructions that will not hinder its normal functioning, but when a massive number of devices take joint action, such as a DDoS attack, the impact can be severe for the victim. In recent years, the number of DDoS attacks from IoT and the volume of the attack have reached their all-time highs [66].

- **Spoofing**: This attack allows the attacker to masquerade its identity by forging messages or falsifying data. The attacker will infiltrate the system by posing as a legitimate IoT device [67] and then carry out a man-in-the-middle attack.
- **Jamming**: By overwhelming the radio waves in a wireless network, the attacker can block or delay the communication between the devices and the system by sending noise signals [68]. It can also be used to isolate part of the network in order to blind the system and allow the attacker to act without being monitored because the sensors will be unable to transmit the wrongdoings.
- **Sleep deprivation**: This attack is aimed at devices running on battery. When an IoT device isn't working, it will be in a sleep state for energy saving. The attacker will send messages that will wake up the device, consuming the battery [69]–[71], usually by pretending to be a node of the network and transmitting mundane network protocol messages that will pass for legitimate communication between two nodes.
- **DoS attack**: As the IoT devices have constrained resources, they are vulnerable to most DoS attacks [72]. This kind of attack can affect different components such as the network bandwidth, CPU time, or memory. IoT devices that are accessible through the internet are particularly sensitive to DoS attacks.

Threats on the Network layer

- **Sybil attack**: The attacker will create multiple identities that will send false information to the system [73]. The goal can be to change the network topology, such as network routes or sending false information to the system. For example, an array of fake sensors can be created to send wrong data about the weather to a smart city or a smart farm. This attack is known to affect mainly distributed and Peer-to-Peer networks where the nodes function as autonomous organizations and nodes are

expected to connect and disconnect frequently [74]. By separating a node or a part of the network, the attacker can divert the network traffic to a sinkhole, consequently interrupting the communication [75], [76].

- **Privileged access**: The attacker either gets the credentials, through malicious means, or exploits weaknesses to gain privileged access to a network component (router, gateway, switch...) allowing them to control it [77]. The configuration of the component can be exploited by an attacker to set up an Advanced Persistent Threat (APT) by collecting metadata and information about the network topology.
- **DoS attack**: Denial of service attacks in the Network layer affect the whole IIoT system. The network is flooded with useless traffic, or vital network components (router, switch, gateways, load balancer) have their resources depleted [78]. The goal is to slow down or block the communication on the network.
- **Man-in-the-middle** (Eavesdropping, traffic analysis): The attacker infiltrates the network to intercept, create, or modify the network communication between nodes [79]. End-to-end encryption is a current challenge for IIoT systems [30], IIoT devices may not be capable of encrypting data with strong algorithms, allowing the data to be easily captured and decrypted by a packet inspector, allowing an attacker to breach confidentiality and to analyze network behavior [80].

Threats on the Middleware layer

- **Privileged access**: The middleware layer is particularly sensitive to privileged access. The components of this layer are the backbone of the whole IoT system and rely on a centralized IAM system. Once a hacker manages to infiltrate the IAM systems, the attacker will be able to tamper with the data, control access, and change security policies [81]. In cloud-based implementations of IIoT platforms, access control is delegated to the cloud services. Poor management of these services and account permissions put the whole system at risk [82] especially in a platform environment where actors come from different companies.
- **Virtualization-based attack**: In this layer, components are either hosted on hardware or virtualized. In the second case, an attacker can exploit a less secure virtual machine to do an escape attack and take control of another virtual machine hosted on the same server [83]. If the Hypervisor is compromised, the system will leak data, or the hacker can stop virtual machines or take control and spread a virus.

- **Database corruption**: Databases can be modified even though the attacker doesn't have credentials. For example, the SQL injection attack exploits insecure fields in applications to send SQL commands to the database, allowing the attacker to delete, insert, or modify the data [84]. An attacker that manages to poison the database will create wrong behavior for the IIoT systems by manipulating the data or altering the visualization for the Application layer.

Threats on the Application layer

- **Code execution**: Through the front-end of most business applications, code can be injected into the back-end in the Middleware layer. The goal is to gain control of the application or the server, and in the worst case, the attack can escalate to the whole system. Malware, such as ransomware, can have a major impact on an information system and the source of financial and reputational loss.
- **Man-in-the-middle attack**: Web applications can be masqueraded to steal information about the user or their credentials. There are multiple ways to do phishing, by sending an email with a malicious link or by poisoning the DNS server to redirect traffic to the phishing application [85]. The attacker installs traffic listeners to collect data on the user and application behavior. The listener can be found on the user's devices or on the host of the application. The potential risk is a breach of confidentiality for the business [80].
- **DoS attack**: A denial of service attack on the application layer affects the users. They might lose the ability to access or control the system, leaving the attacker free to take any malicious actions.

Blockchain

Introduction

A blockchain is a distributed system defined by its architecture and consensus algorithm. It originated from Satoshi Nakamoto's original white paper [86] in 2009, this proposal aimed to resolve double spending in a digital transaction system. The original blockchain, Bitcoin, has succeeded in creating an ecosystem that enables secure, censor-resistant financial transactions, without a trusted third party and with high availability. Since then, many

evolutions of the Bitcoin blockchain have been created, each one responding to specific needs or constraints related to its area of application. There are two main axes of evolution, the first being to improve confidentiality for private use (private blockchain) and the second to improve throughput and computing capacity for public use (public blockchain).

The interest in blockchain technology was twofold, at first it was mostly cryptographic enthusiasts also called crypto anarchists. This movement was born by the end of the 80s, through the cypherpunk mailing list. The term was coined by Timothy C. May in the crypto anarchist manifesto [87], manifesto which was behind the design philosophy of Bitcoin. Bitcoin empowers the user; they have full control over their data (cryptocurrency) without the need of a third party and no entity can block their interaction with the blockchain. It took a few years for Bitcoin to reach a value that it could be exchanged for goods and services; the first exchange was for 2 pizzas for 10,000 bitcoins [88]. Since then, the environment has evolved and blockchain technology has become an industry and a major part of fintech. While the public blockchain shows a lot of progress in recent years, private blockchain still struggles to penetrate the market. The benefit of private blockchain technology holds a lot of expectation but there are few implementations outside of experimentation. One of the main reasons is the improper use of the blockchain in use cases where a traditional solution will be more efficient. A second reason is the lack of global knowledge and misinformation around the technology, inducing a Fear Of Missing Out (FOMO) from CXO and managers, that leads them to put blockchain in non-compatible projects.

Architecture

A blockchain is a data structure composed of packets or blocks of data, which are linked together as a chain where each block references the previous one. This chain is a ledger of transactions that is distributed in a decentralized network. This network is composed of servers (nodes), each one keeping a copy of the blockchain. The nodes are maintained by individuals or companies creating a diverse and complex ecosystem.

There are many blockchain systems, each one proposing different features and capabilities. It becomes difficult to keep up with the evolution of technology, almost all blockchains are open source and are easily forked to implement into a new blockchain. Most of them have their own implementation and their own restrictions on read-write permissions. But overall, they

follow all the precepts of Nakamoto. According to him, a blockchain must be public, readable by everyone, and everyone can interact with it.

We distinguish two types of blockchain, public and private:

- A public blockchain allows all users to read and write to the chain, anyone can host a copy of the chain. It is immutable and highly decentralized and relies on a crypto currency to encourage users to maintain the nodes. The more nodes there are in the network, the more resistant it is.
- A private blockchain allows only authorized users to read and write to the chain, in accordance with governance rules only certain nodes have a copy of the blockchain. These permissions are set by a governance entity (a company for example). Private chains are more centralized and need fewer nodes than a public blockchain, a low number of nodes reduces latency for information propagation and allows for better throughput of transactions than public blockchains.

Consensus

A consensus is when all components of a system agree on a state, whether it is information, a rule, or a decision. The problem of consensus in a distributed system is illustrated by the problem of Byzantine generals [89], where generals have to agree to attack a city but they cannot trust each other and their messenger networks. Blockchain technology proposes a new solution to this problem with less communication between nodes. There are four algorithms commonly used to reach consensus in a distributed system [90] related to the creation of blocks.

- Proof of Work (PoW) [86]: The node that approves a block is called a miner, this one must solve a cryptographic puzzle that validates the construction of a block. This puzzle is in the form of a hash calculation. A calculation that depends on the algorithm defined in the parameters of the blockchain (e.g. SHA256, Scrypt, Dagger-Hashimoto...).
- Proof of Stake (PoS) [91], [92]: The node validating a block is called a validator, it is chosen following a pseudo-random selection. This selection can be influenced by the amount of cryptocurrency the node holds, for example, the richer the node is, the more likely it is to be chosen.

- Practical Byzantine Fault Tolerance (PBFT) [93]: A node broadcasts a block to the whole network, each node of the network will in turn perform validation on this block, if it is validated by the majority it will then be added in the blockchain. This algorithm is mostly used in a private blockchain containing few nodes.
- Delegated Proof of Stake (DPoS) [94]: Variant of PoS where only certain nodes (delegates) can participate in the draw. These delegates are elected by the other nodes. This system implies a strong centralization of the blockchain and is therefore not widely used for public blockchains.

By its nature, a blockchain can have several competing branches that are the result of a simultaneous publication of blocks due to the probabilistic properties of the creation of a block or the speed of the information propagation through the network. Then comes the choice of which branch to continue. This choice is made by a majority vote, depending on which branch is the most popular on the network. The branch selection is part of the consensus rule, each blockchain has its own rule. For example, in the bitcoin blockchain, a node will conserve the longest branch.

The consensus principle is so central in public blockchains that its philosophy influences the entire ecosystem. As most blockchains are open source, the majority decides which version of the protocol will be applied in the network, which software will be used. The strength of numbers also implies that the minority is forced to bend to the majority or to continue on its own version of the blockchain, thus creating a fork [95]. There are two major examples when a community divided and created a fork of a blockchain. First, after the DAO attack [96], the majority of Ethereum users wanted to roll back the blockchain before the attack, while a minority was against it because rolling back didn't follow the principle of blockchain technology, thus two blockchains coexist with the same history. The first one is the Ethereum followed by the majority, and the Ethereum Classic followed by the minority. The second example is with Bitcoin, some miners wanted to change the maximum size of a block to increase their profit, they started to use their influence in social media to promote the idea. Then, they announced the incoming fork of Bitcoin to create BitcoinCash. In a way, when a blockchain forks to create a sub-blockchain, there is always a more popular blockchain followed by a majority, hence reaching a consensus on the most usable blockchain.

Use cases

The blockchain is a very versatile tool with many potential uses. These can be classified into three main categories: the transfer of assets, ledger and traceability, and smart contracts.

Transfer of asset

The first historical use case of blockchain technology is the transfer of money between two parties without the intervention of a third party. Bitcoin is the best example of this use case. Bitcoins can be sent to someone and this information will remain recorded on the blockchain, with the blockchain serving as proof of the authenticity of the transaction. This makes it unnecessary to have a central control authority (such as a bank or credit card company) that ensures the transfer. Most public blockchains are used for the transfer of funds through their associated cryptocurrency. The blockchain can also be used for the transfer of property rights. The transparency of this register provides the integrity and non-repudiation of a transaction, and it is particularly useful for tracing the origin of a good. It is within this framework that Sweden and the Republic of Georgia use blockchain technology for the management of property registers. In March 2017, Sweden completed its second set of tests with startup Chroma way [97]. The Republic of Georgia works with Bitfury and has more than 100,000 pieces of land registered on its blockchain [98].

Ledger and traceability

This use case is very popular with private companies because a blockchain is considered immutable, making it a perfect medium for storing information. And since it is distributed, everyone has access to the same information. Alibaba and PWC use a private blockchain to combat illegal food [99]. IBM and Walmart use it for pork traceability in China [100]. In both cases, new information is added at each step of the supply chain. Project Everledger uses a blockchain to track diamonds [101] throughout the world. With more than a million diamonds registered on its platform, it helps to fight against blood diamonds. A collaboration between Sacem, Ascap, and PRS has set up a shared database to record the International Standard Musical Work Code (ISMWC), which are identification codes for musical works [102]. This shared database avoids registering a work under two different codes in two different countries.

Smart contract

In some blockchains, a user can store code that will be executed later when a transaction calls it. These codes are called smart contracts, and they follow the adage popular among technology enthusiasts, "The code is law," where instructions are executed exactly as they are programmed, blaming the programmer rather than the system for any errors. A smart contract is waiting for a trigger, often in the form of a transaction on the blockchain, to be executed. They can therefore be executed remotely by any member of the network and will remain accessible as long as the blockchain exists, offering greater resilience than a single-server-dependent web service. This is particularly useful in areas such as the Internet of Things (IoT) where we can imagine that a smart contract is used to give instructions to a connected object [103]. Smart contracts can also be used as a contract, where, for example, two parties must fulfill certain obligations for the smart contract to be executed. Another example is a smart contract that deals with the distribution of the shares of different musicians at the time of the sale of a piece of music. This distribution is coded into the contract and, since it is on the blockchain, no one can modify it.

Smart contracts are very powerful tools, but if the code is not written carefully, the consequences can be significant. In mid-2016, the Ethereum blockchain hosted a smart contract for a Decentralized Autonomous Organization (DAO). A hacker found a flaw in the code and managed to steal 3.6 million ethers (the currency of the Ethereum blockchain). The result of this attack triggered a hard fork in the blockchain. On one branch, the blockchain went backwards to cancel this attack. On the other branch, the hacker still has the stolen ethers at his disposal. This new blockchain, which refused the rollback because it was judged to be against the principles of immutability of the blockchain, is called Ethereum Classic [96], [104].

Challenges

As previously mentioned, blockchain technology was designed to solve specific problems, but it is not without drawbacks. The technology was built to be extremely resilient in a hostile environment, but this comes at the cost of performance.

In Table 1, we show a comparison of characteristics between common public blockchains. We observed that the volume of information a blockchain can process varies greatly between different blockchains. Bitcoin has the fewest blocks per hour, but it compensates by having more transactions per block. Transaction throughput was one of the first challenges that needed to be overcome in the early days of blockchain technology, and it was the primary differentiator between different blockchain platforms. Nowadays, blockchains try to differentiate from each other by offering additional features such as privacy, smart contract capabilities, and security. There are two characteristics that are particularly important for the integrity of a blockchain: the number of nodes and the hashrate. These factors determine the robustness of the blockchain against 51% attacks [105]–[107], Control of the blockchain can be achieved if you control either the majority of nodes or the majority of mining power (hashrate).

The main challenges that blockchain technology needs to overcome are scalability, either for transaction throughput or for the size of the blockchain. Another challenge is the validation time, which is the time between when a transaction is sent and when its position is sufficiently consolidated on the blockchain. Privacy is also complex, as increasing it can reduce the traceability of information. For example, while Bitcoin is completely open and readable, Monero hides the sender, recipient, and amount transferred. Finally, the computational cost of Proof of Work blockchains is high enough to raise environmental concerns, and alternative consensus mechanisms are being heavily researched, but they provide less integrity than Proof of Work.

Features Name	Total number of block	Number of nodes	Hash algorithm	Consensus	Block Per hour[3]	Transaction per block	Blockchain total size	Hashrate
Bitcoin BTC	482 331	8843	SHA-256d	POW	6	1630	153.85 Go	5.355 Ehash/s
Ethereum ETH	4 212 361	23168	Dagger Hashimoto	POW	144	100	120.21 Go	85.438 Thash/s
Dash DASH	727 681	4800[3]	X11	POW & POS	23	15	3.80 Go	25.056 Thash/s
Monero XMR	1 386 519	1960	CryptoNight	POW	28	12	24.80 Go	219.671 Mhash/s
Dogecoin DOGE	1 860 757	556[3]	Scrypt	POW	58	11	22.61 Go	12.987 Thash/s
Zcash ZEC	174 669	1389	Equihash	POW	24	10	6.27 Go	294.658 Mhash/s

Table 1: Public Blockchain Comparison (as of August 28, 2017)

Scalability

Scalability is the ability of a system to adapt to changes in size, regardless of the network level or the amount of data to be processed or stored. It is one of the major challenges facing blockchain technology, according to [90]. There is a correlation between the transaction processing rate and the security that a blockchain can provide [108].

If transactions are confirmed too quickly, the system becomes susceptible to double-spend attacks. But if confirmation is too slow, the system loses its appeal compared to traditional systems. We can see that Bitcoin has the lowest throughput compared to other blockchains, yet it remains the most widely used, as evidenced by its higher number of transactions per hour (see Table 1). As a result, researchers are working on developing new proof-of-work or consensus algorithms to allow blockchains to flourish.

The size of the blockchain is also a concern. It is an ever-growing database that requires a significant amount of storage space. The cost of maintaining this system is directly impacted by the need to increase the free space on each node. We can see a significant difference in the sizes of different blockchains, especially among younger blockchains that have more modern and innovative architectures, which result in smaller sizes. However, we can see that Dash has a much smaller size than its competitors. The size of a blockchain depends on the data structure of its blocks, i.e., how the information is stored on disk.

Validation

The validation process for a block depends on the implementation, but in the majority of public blockchains, it is based on the proof of work used by Bitcoin.

When a user sends a transaction over the network, it is added to a list of unconfirmed transactions. A miner then retrieves transactions from this list and starts solving a cryptographic puzzle to create a new block. Before doing so, the miner checks that the transactions are valid. Once the block is mined, it is sent over the network and propagated to other nodes.

A transaction is considered confirmed when a certain number of blocks have been added to the blockchain after it. This means that the block is part of the consensus blockchain and is considered valid. The number of blocks needed to guarantee the confirmation of a transaction depends on the characteristics of the blockchain. For example, in Bitcoin, it is estimated that it

takes six blocks, or about an hour, for a transaction to be confirmed. This validation time depends on the block rate and the probability of a fork occurring on the blockchain.

Privacy

In a public blockchain, the contents of a transaction are readable by everyone to guarantee the integrity of the ledger and to verify the information present during the validation stage of a transaction, or to verify that the sender has the necessary amount of currency to send the transaction.

However, this poses a major problem in a professional context, where certain information should not be published in clear text on the internet, such as personal information or confidential cryptocurrency transfer amounts. In the current implementation of Bitcoin, there is no way to hide this information [109]. Hiding information goes against the concept of a blockchain, but professionals are pushing for developments in this direction. Zerocoin and Zcash implement a method to hide the value of a transaction as well as the identities of the sender and receiver, but this security remains marginal [110]. Monero only hides the identities of the parties involved, not the amounts, which makes the blockchain vulnerable, as shown in [111], [112].

Computation cost

The cost of computation is a major problem with blockchain technology, because most systems use proof of work [105] which is very expensive. In Bitcoin, this proof of work is a hash calculation that requires a lot of computing power and energy. This is partly why there is a reward for mining work. The mining power is so great that the power consumption is estimated to be equivalent to that of Ireland [113]. The cost of mining has an impact on the environment and the value of Bitcoin.

Hash calculations are not the only way to do proof of work. Innovative methods have been implemented to make it more economical or better suited to the system architecture (e.g. [114]).

A private blockchain has no need to use proof of work. Proof of stake or practical Byzantine fault tolerance methods are more suitable because the validator nodes are under the governance of the chain owner. Private blockchains are more efficient and economical than public blockchains.

Vulnerabilities

There are many potential attack vectors on a blockchain system. The most well-studied is the double spending attack, where an attacker tries to spend money more than once. This means that it is possible to make two purchases of the same value but only be charged once. The second most well-known attack is the 51% attack or consensus attack [95]: An attacker controls the majority of the network, either by computing power or number of nodes. He can then create his own branch and force it to be accepted as a legitimate chain by continuing to add blocks to it. By controlling the blockchain, the attacker can create transactions that allow him to do double spending.

The blockchain is a peer-to-peer distributed network where nodes come and go. Each time a node reconnects, it connects to other nodes according to its address book, taken from its connection history. But the first time a node arrives on the network, it asks a DNS server for a list of IP addresses to connect to.

An attacker can then infect the DNS server so that it sends malicious nodes, or set up nodes under his control to perform a man-in-the-middle attack. The victim is then isolated from the network. The attacker then controls the traffic to the victim. By leaving a controlled connection between the victim node and the network, the attack becomes difficult to detect [115].

These attacks for network control are known as Sybil attacks or Eclipse attacks [115], [116]. In both cases, the attacker creates nodes to isolate the victim's node. The default configuration of a node limits the number of possible connections to other nodes. By isolating some nodes, the attacker can increase his percentage of control over the network [106].

A node can also lie when sending a transaction to miners. By sending an erroneous transaction and preventing the miner from realizing it, the miner will waste time creating a block that can never be validated, making them lose money through useless PoW calculations [117].

Another way to isolate a node is to change the BGP routing tables [118], [119]. The traffic can then be either slowed down or diverted. In both cases, the hacker only needs a few autonomous systems.

The list of unconfirmed transactions can be overwhelmed by small transactions. This attack is particularly costly because the attacker has to pay for the transaction fees. Regular transactions will thus take longer to be added to a block [120].

But there are DDoS attacks, not dependent on blockchain protocols, directly targeting the resources of the server hosting the node. We are referring to classic DDoS attacks whether they are ICMP flood or TCP syn. Miners in a pool are often under this kind of attack by a competing pool seeking to recover a percentage of the network computing power. Currency exchange sites are also regularly under DDoS attack in order to manipulate currency rates or to attract users to a competing site [121].

Bitcoin: Protocol Description

To have a better comprehension of blockchain technology, we will use Bitcoin as an example to introduce the fundamental concept of blockchain. We employ Bitcoin when talking about the blockchain and bitcoin (BTC) when talking about the cryptocurrency/token associated with it. A bitcoin is divided into a smaller unit called the satoshi, giving the correspondence: 1 BTC = 100,000,000 satoshi.

Bitcoin is a network of nodes, each of which holds the blockchain. Tokens are used to incentivize users to keep the network running smoothly. The value of this token is influenced by supply and demand. New bitcoins are minted when a miner creates a block. This implies that only a limited number of bitcoins can be in circulation. This incentive is very important for the system, as it forces users to remain honest if they want to continue to earn bitcoins. Information about who and how much Bitcoin is held is stored in the blockchain. Anyone can consult it to check its contents. Content that is replicated on every node in the network. Bitcoin operates in a non-trusted environment, so the veracity of the information is crucial. A constant verification is performed by the nodes to ensure the health of the network.

A transaction is the data structure containing information about a bitcoin transfer between two parties. This data will pass through the network until a miner adds it to a block. A block is the data that is saved in the blockchain, and it contains one or more transactions. Miners are the ones creating the block by solving a computational puzzle called a proof of work.

Transaction

Bitcoin was created to exchange tokens having a financial value. [86]. This currency transfer is done by publishing information about the transaction over the network. A transaction is a data structure that contains all the information necessary to send bitcoins. The sender creates a signed transaction with their private key. In the Bitcoin system, there is no notion of an account, but the owner of a private key controls the associated transactions, so we can say that there is an account for each private key. A transaction will be broadcast over the network. The sender wants the visibility of their transaction to be as high as possible so that it can be processed quickly. A transaction ending in the blockchain is considered confirmed.

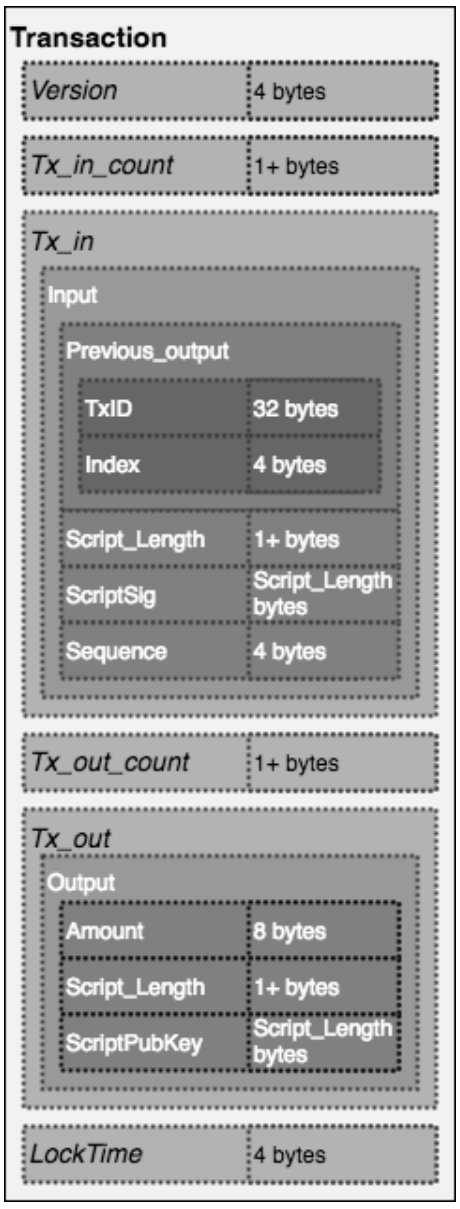

Fig. 6: Diagram of the data structure of a transaction

Block

The database update is done in a batch format to facilitate synchronization between nodes. This packet, containing all the updates at a given moment, is the simplest form of information transfer for a distributed network. In Bitcoin, these packets are called blocks. A block is created by a miner. This particular user of the network has the task of solving a cryptographic puzzle that will verify the integrity of the block thus created. The creator of a block receives a Bitcoin reward. This creates competition between the miners to be the first to solve the puzzle. The system throughput is one block every 10 minutes on average. This means that it takes 10 minutes for a miner to find the solution to the puzzle. Miners may have a different view of the available transactions to be added in a block, due to the information propagation time and the high volatility of the network. The transactions are stored in a mempool of available transactions. This list is saved in a node's RAM, so it is unique to each node. If the node is switched off, it loses this information, and it will rebuild it by asking its neighbors to transfer their list to it. A freshly mined block is sent on the network and will be broadcast if it is considered valid. A transaction present in the blockchain is considered confirmed according to its maturity. The maturity is the number of blocks after the one containing the transaction. The Bitcoin documentation recommends waiting 6 blocks, or one hour, for an almost guaranteed confirmation. The durability of a transaction increases exponentially with maturity.

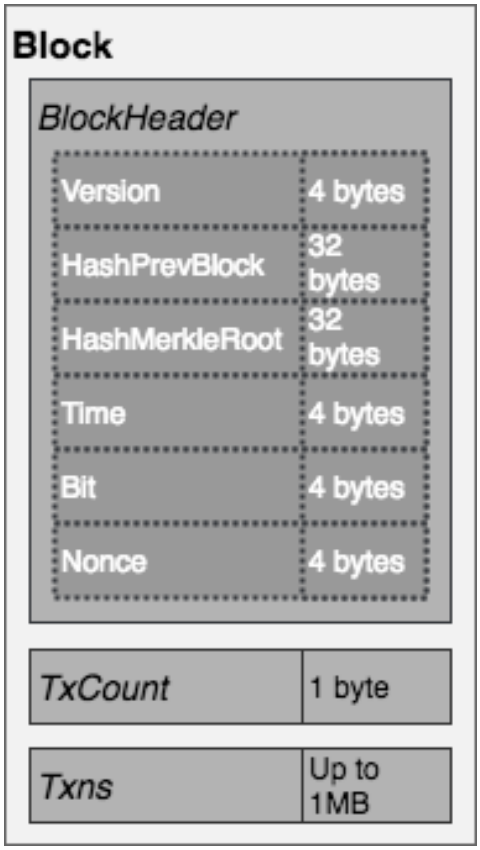

Fig. 7: Diagram of the data structure of a block

Blockchain

The blocks (fig. 7) are linked together by a field in their data structure, this field contains the hash of the previous block, its parent. As its name indicates, a blockchain is made of several blocks chained together. This way, information is stored chronologically, like in a ledger or accounting book. Therefore, the transactions have an order, and knowing the total value associated with a key is simply done by looking in the blockchain in reverse order.

The blockchain is organized in a tree structure, from the root (the first block) to the end of the longest branch. The root is called the genesis block, and it is the only block without a predecessor. We can then distinguish four different types of blocks. A normal block, being part of the blockchain, is valid, and its parents are known and present in the blockchain. Invalidated blocks that are rejected by the nodes during their validation are destroyed. Stale blocks are blocks that are not part of the largest branch, they are valid and have a parent in the blockchain, but they often come from a fork where they were in the shortest branch. The last type is the orphan blocks, they are valid blocks, but their parents are not yet known. Most of the time, they appear when their parents have not yet had time to propagate in the network.

When a miner creates a block, their parents are often the end of the blockchain. But they can create a block whose parent is not sufficiently broadcast on the network, it doesn't mean that this block will be invalidated. Let's take the example where the end of the blockchain is A, a miner creates a block B whose parent is A. Immediately after, they find the solution that allows them to create block C, whose parent is B. Some nodes may receive block C before receiving block B. For them, the transactions in block C will not be considered confirmed until B is received.

Blockchain forks

Forks are common occurrences in a blockchain environment. A fork is an event when two miners find a block at the same time, these two blocks having the same parent. The problem is that only the longest branch is the correct database. So the fate of the blockchain will be decided by the next block. The choice of a miner between two potential parents depends on their vision at time T of the blockchain present in their node or that of their neighbors. While the mining process is taking place, the election of the longest chain will be chosen by the majority of the network. This election is influenced by the dissemination of information and the choices made by the miners. The blocks present in the losing branch are the stale blocks. They will still be saved on the disks by the nodes, but they will not be part of the blockchain. They are saved because they are still in a potential branch, and if a miner with enough computing power catches up with the longest branch, they can change the state of the blockchain. Then, all the transactions present in the old branch will have to be reprocessed if they are not in the new branch. A transaction is never confirmed, but its probability of being in the blockchain is exponential after each new block [86]. The branch detachment process will be explained later in this work.

Creation of a transaction

Creating a transaction means that the sender already has an output that they can prove they own. This output can be a coinbase or an output from a transaction. Unlike the banking system, a Bitcoin user does not have an account; they only have private keys linked to public keys. This means that exchanges are only between keys and not between individuals. These actions are performed by the functions of the wallet software. It is possible to create a transaction without the help of third-party software, but it turns out that this requires some

effort [122]. A wallet also offers convenience to the user. It saves information about the user's outputs locally, shows the total amount of bitcoin linked to several keys.

First, the wallet retrieves information about the outputs to be spent. Their cumulative value must be greater than the amount to be sent. The sender adds their output TxID and the index in the input, followed by their ScriptSig for each output. Inputs can come from several private keys. It then adds the outputs of the transaction. These outputs can be for different destinations, each one is associated with a public key. This key is used to indicate who will be able to use the output. Note that if the sender has an amount of input greater than the output, the output will be used. They will have to create an output to reassign this difference. This output is called the Change output. Otherwise, this difference will go to the miner as a transaction tax. In both cases, the user will have to put a tax in their transaction. It is used to incentivize the miners to process the transaction and put it in a block. The miner receives all the taxes present in a block. Miners tend to choose the transactions with the highest taxes. The higher the tax, the faster the transaction will be processed. Currently, the average tax is 420 satoshi per byte, the average transaction size is 226 bytes, which makes a tax of 94,920 satoshi (or 0.00094920 bitcoins) per transaction [123].

Validation of a transaction

A transaction must be sent over the network to be processed. Each node will broadcast the transaction to its neighbors after checking its validity. A transaction must meet several criteria described below. These criteria are the consensus rules for adding data to the blockchain. The majority will dictate the rules that the minorities will have to follow. Currently, the rules are those of Bitcoin Core because it is the most widely used implementation. But a new set of rules can be pushed by users using another implementation. If they become the majority, their rules will become the consensus rules.

We categorize the process in four steps. We consider that we are referring to an honest node; a malicious node will seek to propagate transactions that do not respect the rules and therefore will be invalid.

Version verification

The node will check if the version number of the transaction is standard and implemented in the node.

Currently (06/2017), the standard number is 2; any other number will be considered non-standard and therefore rejected by nodes using the rules defined by the Bitcoin Core developers.

Simple verification

There are several checks at this stage, which are inexpensive for the node to perform. They require little computing power because they are mainly database accesses. Base which is very optimized and small in size.

First of all, the node checks that there is one or more inputs in the transaction.

Then for each input:

- Look if the input is not already referenced in the UTxO. The Chainstate database is consulted.
- Check if the input is not already saved in the blockchain. The index will be consulted.
- If the input is a coinbase, the node will look if it is mature enough. There must be a difference of at least 100 blocks between the height of the input block and the height of the last block.
- Look to see if the amount is not negative or overflowed, no more than 21 million bitcoin.

Then the node will verify that the total inputs are greater than the total outputs. Then it will check the tax, which must neither be negative nor overflowing.

Signature verification

This verification is the most expensive in terms of computation. The node will execute the scripts. At first, it will always run the ScriptSig to stack the data on the stack. Then the ScriptPubKey. These scripts are present in the inputs of the transaction, and the node will have to retrieve the ScriptPubKey by consulting the Chainstate database.

P2PKH

This script compares the public key with the hash of the public key present in the ScriptSig. If they are identical, it will then check that the signature matches the public key. The script returns TRUE if everything was executed correctly and FALSE if there is an error.

P2SH

The script compares the hash of the redeemScript present in the ScriptSig and the ScriptPubKey. If they match, the redeemScript will be executed.

LockTime verification

The node will look to see if the time between the LockTime and the current time is sufficient. Bitcoin Core considers a block valid for up to 2 hours in the future.

Bloom Filter

A bloom filter is a probabilistic data structure that offers high compression for a high false positive rate [109].

A bloom filter allows a node to filter transactions they receive. Transactions are filtered by ID, output scripts, inputs, input scripts, and transaction fees. A node sends this filter to its peers, who will decide whether to send or not send transactions that match the filter criteria. Every node containing a wallet uses a bloom filter because it simplifies the update process.

Comparison of Internet of Things Reference Architectures

Introduction

The advent of smart objects in our society is leading us to an ultra-connected world. IoTs (Internet of Things) are devices that communicate through internet protocols, and their uses are multiple. They allow for example to record data on their environment as well as to act on it. IoTs have evolved over the years from fixed sensors connected via copper wires to mobile sensors communicating wirelessly over long-distance networks. Therefore, the protection of these systems is key to their sustainability. IoTs are sensitive to attacks, e.g. Stuxnet, or they can be the source of a vector of attack, as shown by the Mirai botnet. Thus, IoT projects require great attention at their design stage, as they are particularly vulnerable systems. Security by design must be considered both at the architectural level and in the deployment of the system. Numerous frameworks exist that are adapted for different contexts and technologies to integrate IoT systems in Information Systems. IoT can bring many unknown vulnerabilities to an Information system. To ensure an integration following the best practices, industries need to assess the best architecture for their IoT system. In this chapter, we reviewed 10 reference architectures for IoT systems by comparing their capacities to fulfill security requirements. One of the most important security services needed is availability, as IoT devices can have mission critical tasks or need to report to a command center in real time. To review and compare the reference architectures, we used the multi-criteria decision method: Analytic Hierarchical Process (AHP), which allows us to evaluate each capability and compare them to elect the architecture most suited for a use case. By proposing a robust landing framework for IoT devices, we ensure a security perimeter at the architectural level.

Platform Enterprise

Nowadays, companies' IT infrastructure can be summarized into a five-function model (Fig. 8) based on data platform reference architectures [123], [124]. This model is composed of the following functions:

The **Collect** function, the data is collected by the multiple sensors of the systems, also known as the data acquisition phase. The type of data varies from environmental data (temperature, hydrometry...) to messages (logs, error messages...). The data are transferred from the Things layer through the Network layer.

The **Store** function, the data is stored inside a database, HDD, or Hadoop cluster. After pre- and post-processing, it is retrievable for the other layers, and its access is controlled by an Identity and Access Management service. The data can be stored locally close to the machine, centralized in a cloud environment, or distributed between the different industrial sites.

The **Analyze** function, the data is processed through algorithms by compute units to transform and make it readable by software or humans. The data is used to optimize the system, predict behavior or operation, and detect anomalies.

The **Visualize** function, the data is aggregated into applications to monitor the systems. Decisions can be taken based on metrics or Key Performance Indicators (KPIs). This process can be automated by algorithms or AI or by human intervention.

The **Execute** function, once interpreted, the data is used to give instructions to the system, which can be destined for middleware tools or machines. The instructions can be to optimize production or to carry out certain actions remotely.

This data pipeline is fundamental to every IIoT (Industrial Internet of Things) system. The objective is to change the status of the system based on the information collected. This proposed model can be applied to every domain of the industrial sector. It is derived from our previous research [125] and academic literature [126], [127], We concluded that the data and business perspective is often absent from proposed IIoT reference architectures. We present the different industrial domains (Fig. 6), namely Industry 4.0, Logistics, Retail, Smart Grid, Agriculture, and Healthcare, where the context needs a ubiquitous view of the environment and remote control. These use cases imply that a significant amount of data is collected and processed in datacenters, edge, fog, or cloud computing, in the so-called middleware layer of a traditional IoT system architecture. As systems become more autonomous, the need for

human intervention will lessen, and we will see an increase in productivity. Before, IT systems were considered a necessary expense, but nowadays, thanks to innovations in infrastructure, automation, and security, IT systems are providing new use cases and creating value for businesses. Those innovations transform businesses into platforms of exchange, where cooperation between companies will increase productivity, reduce maintenance costs, and raise their observability of the ecosystem. This new paradigm of companies is called a platform enterprise [128] supported by the concept of IT as a platform [129], [130]. This concept is embodied in the IoT platform cloud solutions (Google IoT, Azure IoT hub, AWS IoT...) that provide services centralizing all the needs and applications. These platforms at the heart of the middleware layer are supported by management and security tools. Operating an IT system as a platform requires culture and technology. Approaches such as Agile Devops and SRE (Site Reliability Engineering) change the management of IT systems, while innovative companies (Netflix, Uber, AirBnB) are natively agile in culture, they also demonstrate it in their technology approach to innovation. Netflix can afford to run multiple instances of chaos monkeys [131] in their infrastructure, meanwhile an industrial might be reluctant because their IT systems are not resilient against insider attackers. A machine shutting down on an assembly line will have a larger impact on the physical world rather than a virtual component (e.g. router, server). Industrials need to change their approach to designing information systems and take into consideration the benefits of IoT complemented with cloud computing.

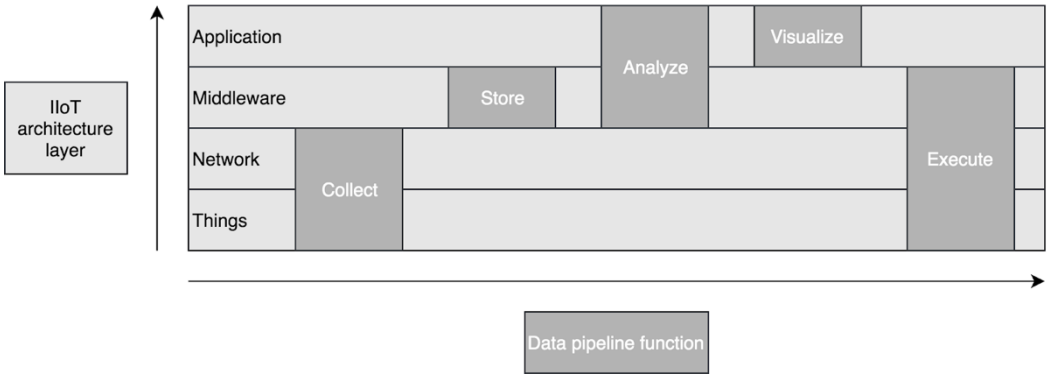

Fig. 8: Mapping of data functions on IIoT architecture.

	Collect	Store	Analyze	Visualize	Execute
Agriculture	Data about the fields or the cattle : temperature, luminosity, hydrometry, GPS coordinate ...	The data are stored in the production site locally for remote place or in the cloud	Data are used to make prediction and view the state of the fields or cattle ...	Application that allow the producer to review his farm and take decision based of the data. The data can be accessible to other adjacent producer for enabling cooperration	The producer can remotely control the farm, for example feeding the cattle or watering the fields
Logistic	Data about the goods and the delivery fleets : GPS coordinate, fuel consumption, traceability in the supply chain ...	The data can be stored distributed or centralized in a cloud	Data are used to estimate the time to arrival, optimize the supply route, anticipate shortage in a supply chain ...	Companies can visualize their goods in transit. Delivery compagnies manage their fleets	Reroute the supply chain ...
Healthcare	Data about the patients : Body temperature, heart rate, blood pressure ...	The data can be stored locally in the hospital or an certified datacenter	Data can be used to predict patient condition, to find the best treatment, to detect anomaly or health issues ...	Doctors can remotely monitor the patients, adjust the treatment, exchange the data between experts	Doctors can remotely control heath equipment
Industry 4.0	Data about the production : rate of production, maintenance state of the machines, sensors ...	The data can be stored distributed between multiple factory or companies or centralized in a cloud	Data can be used to predict maintenance, to optimize the supply chain, to adapt the production ...	Employe can remotely monitor the the factory plant, adjust the volume of production	Machines adapt their behaviors based on the data or order given by humans
Retail	Data about the product : Temperature in storage area, position of the goods, position ...	The data can be stored locally in each store and recentralized in a datacenter	The data are used to optimize the storage, predict sell ...	Employe have an insight of their store, manager can see their stock and they can follow and detect misplaced product	Temperature in the storage area can be adjusted remotely ...
Smart Grid	Data about the energy grid : The maintenance state of the grid, the consumption of the client, intensity ...	The data can be stored distributed or centralized in a cloud	The data are used to optimize the power flow, predict maintenance ...	Employe see the incident in the power grid, redirect the power flow ...	The smart grid reroute the power based of the data and incident, add new section (plug and play) ...

Table 2: Application of our model and examples for each industrial sector

IoT Architectures

We compare two types of IoT architectures: frameworks and platforms. The first ones are architectures from industrial consortiums, research groups, and laboratory research work. These architectures are abstract in their design and tend to focus on the design of a smart factory rather than an Information system integrating an IoT system. Thus, their approach is interesting when starting from scratch (building a new factory) rather than modifying an existing system. Their broad architectural view allows to englobe often forgotten or peripheral components of a system (human interaction, physical incident...). They are very well documented and the result of several years of research. The second type is the architectures provided by the IoT platform providers. These architectures are based on technological solutions proposed by the service provider. Although they offer a concrete answer, they tend to lock their clients into their ecosystems, which makes it more difficult to implement future evolutions. Their propositions are more grounded in reality, often presented with real use cases.

IOT Frameworks

The framework architectures selected for this study are the most common architectures discussed in the literature. We chose the Reference Architectural Model Industrie 4.0 (RAMI 4.0), Industrial Internet Reference Architecture (IIRA), Internet of Thing Architecture (IoT-A), the reference architecture from WSO2, and the reference architecture from Cisco. The first three are from either consortiums or research groups formed by public and private researchers. WS02 and Cisco, on the other hand, are from private companies presenting their vision for these new technologies. In the following section, we will present you with the key features of each framework architecture.

RAMI 4.0 [132]

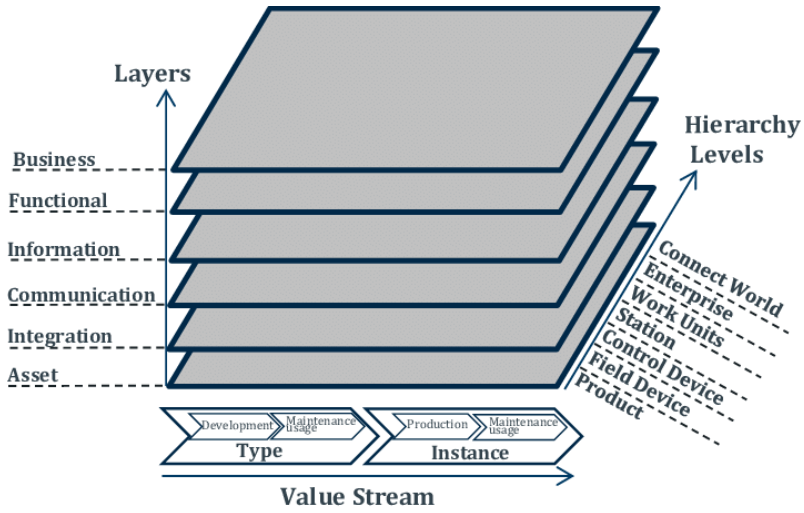

Fig. 9: Reference Architectural Model Industrie 4.0 (RAMI 4.0) [132]

The reference architecture RAMI 4.0 (Reference Architectural Model Industrie 4.0), in Fig. 9, is the result of the Industrie 4.0 initiative in Germany. It is composed of three axes: the hierarchy axis, the product life cycle axis, and the architecture axis. The first axis symbolizes the hierarchy of machines, the interconnection between the components of the smart factory, and the integration of the machines in the factory. The product life cycle axis represents the stages of development of the manufactured product from design to maintenance. The third axis is the architecture, a representation in the form of a layer of the information system, ranging from the connected object to the business applications. RAMI 4.0 is an end-to-end

design for a smart factory and the manufactured product, considering not only the supply chains but also the maintenance of the product.

IIRA [134]

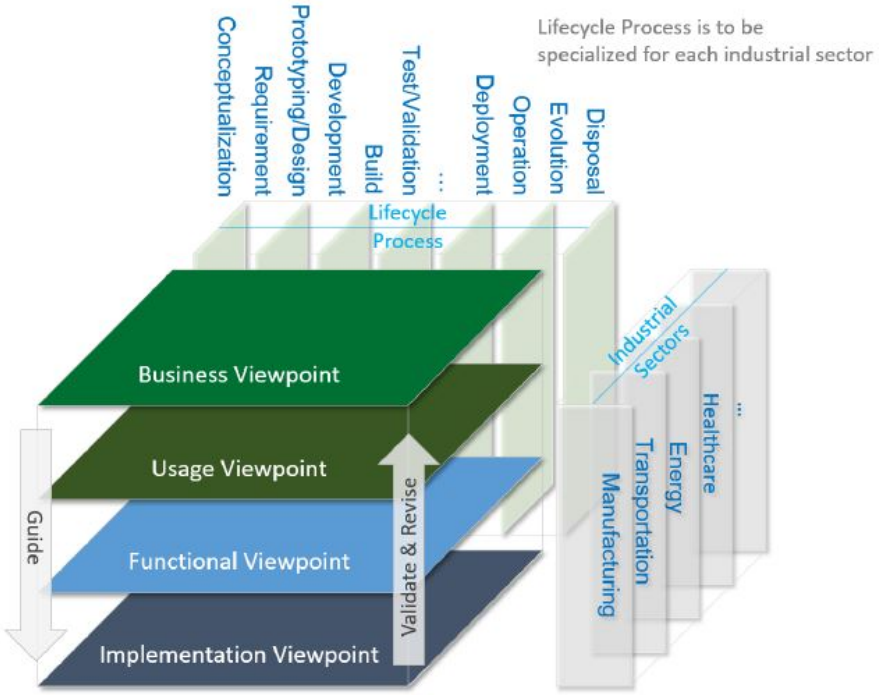

Fig. 10: Industrial Internet Reference Architecture (IIRA)

The IIRA (Industrial Internet Reference Architecture) was designed by the Industrial Internet Consortium, an American group aimed at standardizing and promoting the evolution of the industry towards the IT world. This architecture is composed of three axes: functional domains, system characteristics, and cross-cutting functions. The functional domains are divided by representing the system architecture, symbolizing, among other things, the interactions between devices, applications, and business, etc. System characteristics are the services that the system must cover, for example, scalability or confidentiality. Finally, cross-cutting functions are the functionalities that must be provided throughout the scope of the information system. IIRA is designed around infrastructure management in a highly industrial context.

IoT-A [135]

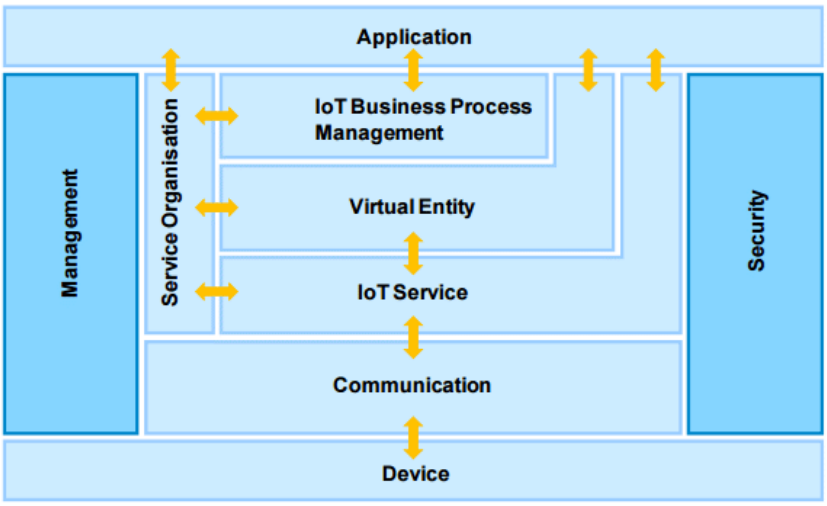

Fig. 11: Internet of Thing Architecture (IoT-A)

As a result of a European initiative, a research group was set up to design a reference architecture to facilitate interaction between European industry players. IoT-A (Internet of Things Architecture) is a layered architecture with two verticals. It presents the services needed to connect devices to business applications. It emphasizes the importance of middleware for the management and proper functioning of the infrastructure. IoT-A enables high interoperability in device and service communications.

WSO2 [136]

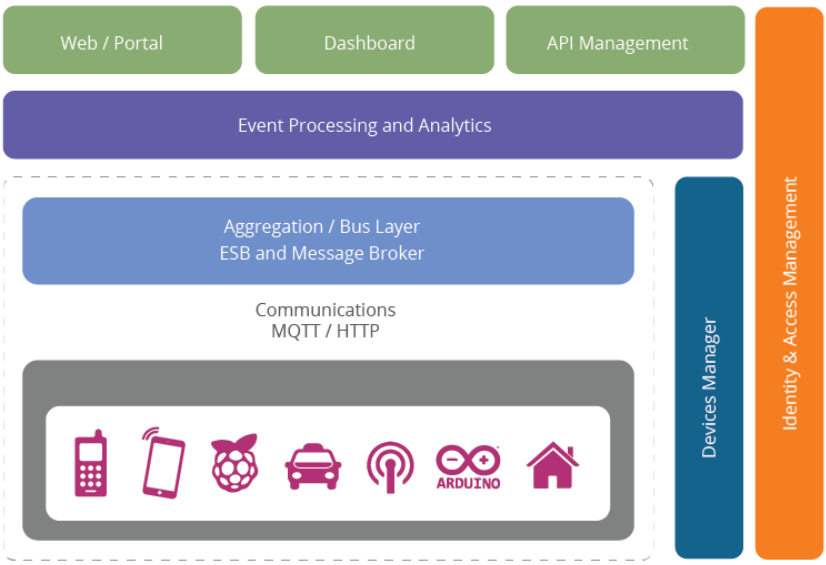

Fig. 12: WSO2 Reference architecture for IoT

WSO2 offers a reference architecture to support integration between systems and devices. This architecture is modeled in five layers: Device, Communication, Aggregation/Bus, Event Processing and Analytics, and Client/External Communication. There are two transverse verticals, the first one is for the management of connected objects, and the second one is for the identity and access control to the system. This architecture focuses on data collection and analytics.

Cisco [137]

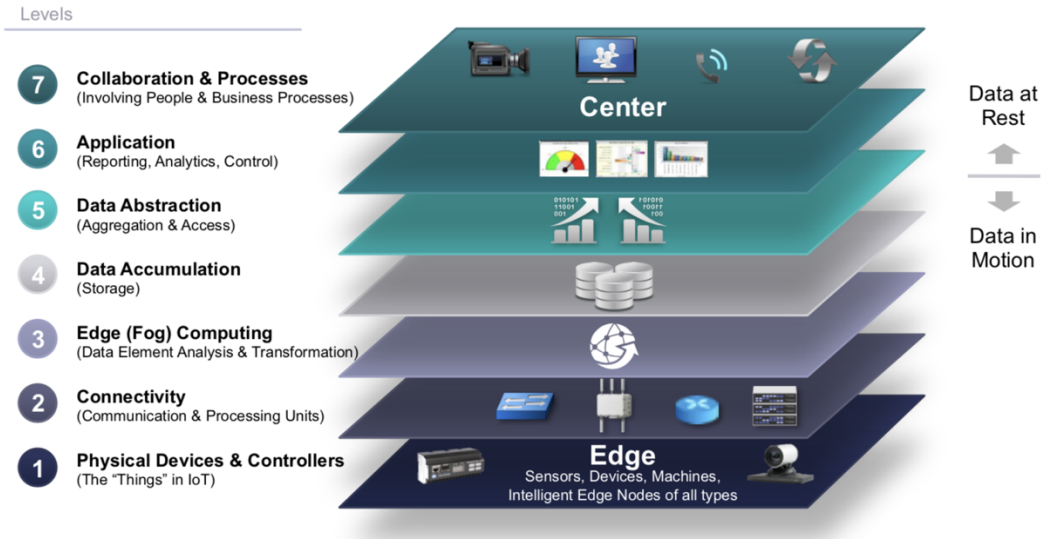

Fig. 13: Cisco Reference architecture for IoT

The reference architecture proposed by Cisco provides a definition and description of the elements necessary for the operation of an IoT system. The model includes seven layers: Physical Device, Connectivity, Edge Computing, Data Accumulation, Data Abstraction, Application, Collaboration & Processes. This architecture carefully describes the importance of data throughout its life cycle, from its collection to its interpretation by a human or machine. However, this architecture remains vague on the technical implementation.

IOT Platforms

Platforms architectures are examples of implementation using the services proposed by the corresponding platforms. The main advantage of these architectures is the availability of documentation on the services and solutions proposed by the service providers, helping industries in the planning of their information systems. As stated earlier, these architectures

focus mainly on higher layers, middleware, and application, while introducing tools for managing the system. The selection of the architecture was based on the availability of the documentation and their references in academic literature.

SiteWhere [138]

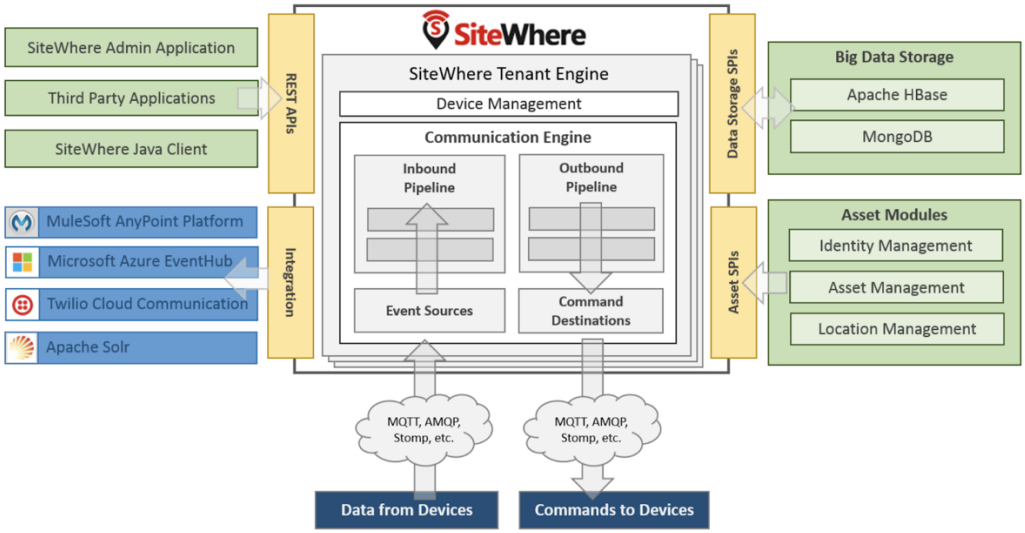

Fig. 14: SiteWhere Reference architecture for IoT

SiteWhere is an open-source IoT management platform that can be deployed on-premise or in a cloud infrastructure. The reference architecture accompanying this solution is centered around a central application that interconnects objects and services (storage, management, interface to third-party solutions, business applications). The architecture is based on the principles of SOA (Service-Oriented Architecture). The SiteWhere application is the gateway between the device and the information system. It ingests the data from the device and redirects it to databases or other applications. The application can also be configured to command devices.

AWS IoT [139]

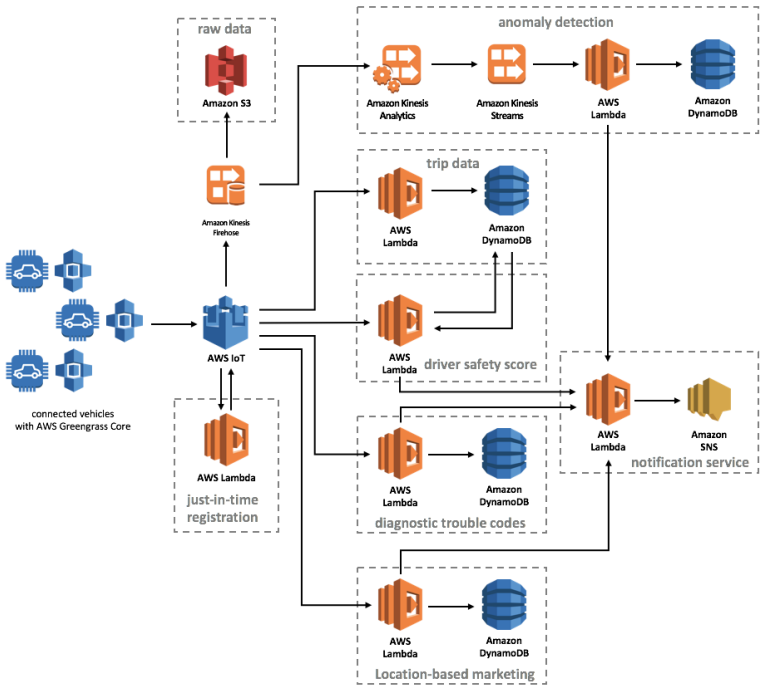

Fig. 15: Reference architecture for IoT

The AWS IoT reference architecture, here for connected vehicles, describes the essential components for processing information from connected objects as well as the tools or services to be deployed to manage IoT such as identity management, access control, and security policies. The documentation presents the best practices for developing an IoT system but does not cover communication between devices. Overall, it remains focused on middleware and application, the upper layers of an IoT system.

IBM Watson IoT [140]

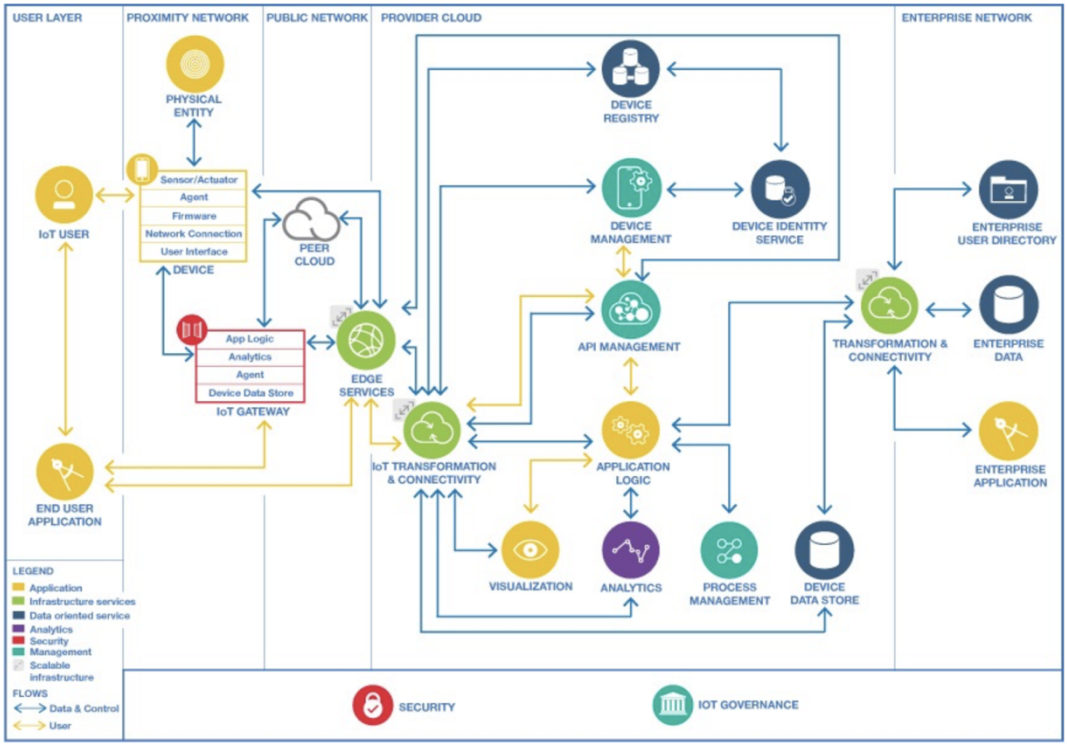

Fig. 16: IBM Reference architecture for IoT

The model proposed by IBM is based on IIRA and shows the integration of objects connected with their Watson IoT platform. The architecture delimits services by their position in networks: User, Proximity (Devices), Public, Provider Cloud, Enterprise. The services offered by IBM are mainly of the PaaS or SaaS type, and their interactions are shown on the diagram of their architecture. Their design includes the essential security services for these systems. IBM stands out for its data processing capabilities that integrate artificial intelligence.

Azure IoT Hub [141]

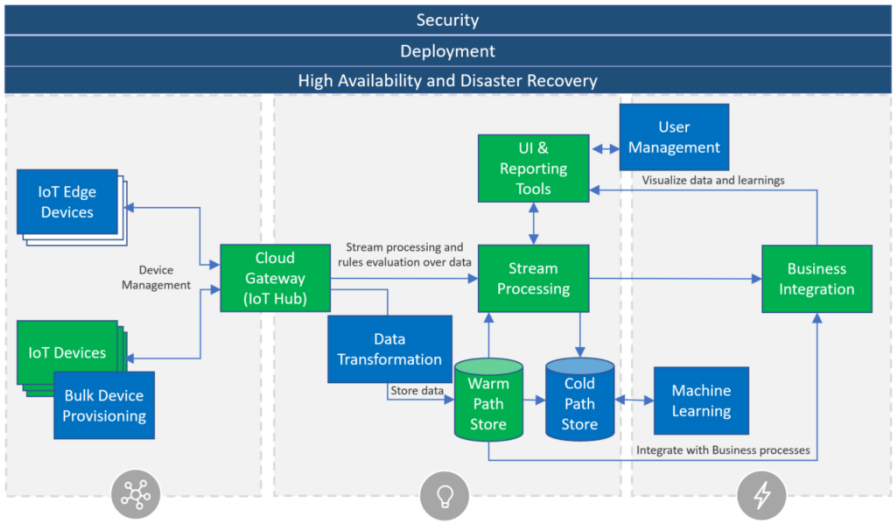

Fig. 17: Azure Reference architecture for IoT

Microsoft Azure recommends an IoT platform architecture based on microservices and serverless concepts. This allows each sub-service to be sized according to the load. The architecture is divided into three parts: Things, where the connected devices connected to the system via a gateway are located, Insight, representing the logical area for data processing and storage as well as monitoring, and Actions, the integration of business applications into the system. Three transversal functionalities are described: security, which must be end-to-end, logging and monitoring (Deployment), which is present on each element, and high availability and disaster recovery, which must be provided on the system components to ensure service continuity.

Google IoT [142]

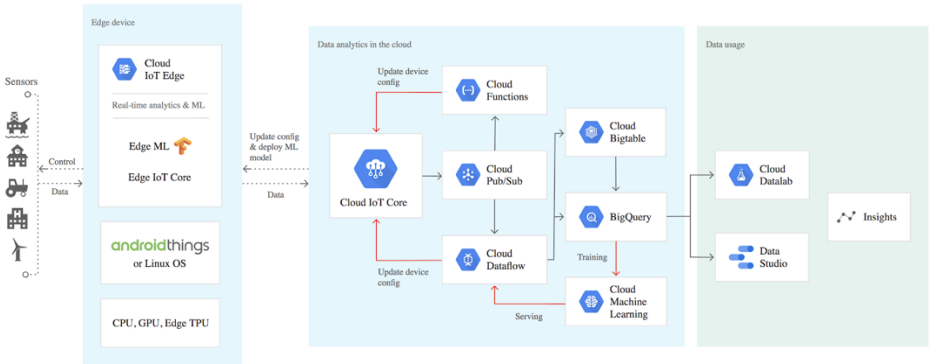

Fig. 18: Google Reference architecture for IoT

The architecture proposed by Google is oriented towards processing large amounts of data arriving in real time. It is composed of three layers: Devices, Data Analytics, and Data Usage, corresponding respectively to data acquisition, processing, and finally, visualization and decision making. The architecture considers access control for both devices and connected objects. Data processing can be done in the cloud or on the edge.

Contribution

The goal of our contribution is to validate a method of comparison between multiple solutions. While a straight comparison between two elements with quantitative metrics is facilitated by numerous methods, qualitative comparison is more subjective and depends on the reviewer's appreciation. In our line of work, we use questionnaires to evaluate the needs and requirements of our customers. This method needs to be customized for each client and takes a lot of effort. The proposed method offers the advantage of being agnostic of the context while accurately choosing from different alternatives with only qualitative data. In the following section, we will present and implement this method by comparing the reference architectures on their capability to deliver the availability security services. To compare the architectures, we choose the Analytic Hierarchy Process (AHP) [143] because it allows us to compare the alternatives through qualitative pairwise evaluation. [144] states that the AHP method publications outnumber the other multi-criteria methods in the span of 15 years, reinforcing the choice of this solution.

AHP is a method for multi-criterion decision used for comparing solutions based on a set of criteria. This method is well suited for fuzzy comparison where the assessment of a criterion is left to the auditor. First, we need to set a hierarchy symbolizing all the components. At the top, we define the goal, here it's to guarantee the availability of the system. Next, we set the criteria or the elements defining the goal, here it's functional requirements. And at the bottom, there are the alternatives, which are the different reference architectures. The second step is to classify the criteria pairwise, in order to determine the most important one. The classification is done by grading a pair following the scale in table 3.

Intensity of Importance	Definition	Explanation
1	Equal Importance	Two activities contribute equally to the objective
2	Weak or slight	
3	Moderate importance	Experience and judgement slightly favor one activity over another
4	Moderate plus	
5	Strong importance	Experience and judgement strongly favor one activity over another
6	Strong plus	
7	Very strong or demonstrated importance	An activity is favored very strongly over another; its dominance demonstrated in practice
8	Very, very strong	
9	Extreme importance	The evidence favoring one activity over another is of the highest possible order of affirmation
Reciprocals of above	If activity i has one of the above non-zero numbers assigned to it when compared with activity j, then j has the reciprocal value when compared with i	A reasonable assumption

Table 3: The fundamental scale of absolute numbers from [143]

The third step is to classify the alternatives for each criterion, this is also a pairwise classification. Each comparison determines if the two options are equal or which one has the most varying degree of importance. Thus, by the end of this process, we can determine the reference architecture which addresses with the most consideration the availability of the entire system. We used an online implementation of AHP for this analysis [145].

Functional requirements

We have taken up the functional requirements presented by ITU [60]. In this document, the requirements are defined in subcategories according to their function. The non-functional requirements (Interoperable, Scalable, Reliable, High Availability, Adaptability, Manageability) aren't considered in our study because they include too many concepts behind them. Therefore, we choose to focus on the 48 remaining requirements. We established a selection of 12 requirements addressing the availability problematic:

	Criterion	Definition
1	Remote monitoring	Notification of the status of things and its changes is a necessity for anticipating maintenance and detecting down time
2	Plug and play functionality	A new device or service should be able to connect to the system seamlessly, for example if a service goes down a replacement need to be integrated immediately
3	Mobility of device, user and service	Things, users or services should be able to access the system wherever they are
4	Communications between devices	The communication needs to be reliable between devices. It can be direct through gateway or through the intermediary of the upper layer
5	High reliability and security on human body devices	IoT device worn directly on the human body can carry critical mission, e.g. in E-Health scenarios. In those cases, a discontinuity in the system would cause a life threating situation
6	Self-configuring, self-healing, self-protecting	The system needs to be able to react itself to ensure its well-being. hence, by the combination of constant monitoring, redundancy and automatization, reaching a high availability can be achieved
7	Location based interaction between IoT actor	The system should enable flexible, user-customized and autonomic services based on the location information and related context of things
8	Programmable interface (or API)	To ensure better availability to services, the implementation of programmable interfaces ease the communication and allows to easily lead to the right resources
9	Global time-stamping synchronization	The global synchronization is crucial for time sensitive work by extent for the availability
10	Highly available and reliable service	Services in the upper layer of the architecture need to be behind mechanism insuring high availability and reliability
11	Discovery services	A new device or service need to be able to communicate easily with the rest of the system. Reliable discovery service needs to be deployed as to lead request to the available resources
12	Standardized naming and addressing	Standardized naming and addressing will facilitate the integration of new service and device

Table 4: Criteria

The pairwise comparison gives us the following matrix in table 3.

	1	2	3	4	5	6	7	8	9	10	11	12	*Priorities*
1	1	0.20	1.00	0.33	0.20	1.00	3.00	7.00	0.33	0.20	5.00	9.00	0.06
2	5.00	1	5.00	0.33	1.00	1.00	5.00	7.00	5.00	3.00	5.00	5.00	0.15
3	1.00	0.20	1	0.33	0.50	0.33	5.00	7.00	3.00	0.50	1.00	3.00	0.059
4	3.00	3.00	3.00	1	5.00	5.00	7.00	9.00	7.00	3.00	7.00	7.00	0.25
5	5.00	1.00	2.00	0.20	1	1.00	5.00	7.00	5.00	1.00	7.00	7.00	0.123
6	1.00	1.00	3.00	0.20	1.00	1	7.00	9.00	3.00	1.00	7.00	7.00	0.107
7	0.33	0.20	0.20	0.14	0.20	0.14	1	1.00	0.33	0.20	0.20	0.33	0.016
8	0.14	0.14	0.14	0.11	0.14	0.11	1.00	1	0.14	0.14	1.00	3.00	0.016
9	3.00	0.20	0.33	0.14	0.20	0.33	3.00	7.00	1	0.14	5.00	5.00	0.055
10	5.00	0.33	2.00	0.33	1.00	1.00	5.00	7.00	7.00	1	5.00	7.00	0.123
11	0.20	0.20	1.00	0.14	0.14	0.14	5.00	1.00	0.20	0.20	1	2.00	0.026
12	0.11	0.20	0.33	0.14	0.14	0.14	3.00	0.33	0.20	0.14	0.50	1	0.017

Table 5: Requirement matrix

IOT Architectures Ranking

Following the classification of the requirements we perform the one for the reference architecture. First, we need to determine how an architecture meets a requirement. If the documentation hints, mentions, or presents a solution to the requirement then we determine that it is covered. We make a distinction for requirements that include multiple criteria, like “Self-configuring, self-healing, self-protecting”, which can be partially covered, hence it will be noted as such.

Coverage	None	Partial	Full
None	1	1/3	1/5
Partial	3	1	1/3
Full	5	3	1

Table 6: Coverage rating matrix

We then do a pairwise comparison for each requirement. Using the rating presented in table 4, the ranking previously done set the priorities (weight) for each requirement. The final ranking of the alternative is based on the conjunction of the priority for each requirement and the architectures' evaluations.

RAMI 4.0	IIRA	IoT-A	WSO2	Cisco	SiteWhere	AWS IoT	IBM Watson IoT	Azure IoT Hub	Google IoT
10.0%	11.3%	12.6%	5.0%	8.9%	8.5%	9.6%	14.3%	10.5%	9.3%

Fig. 19: Final evaluation of the alternatives

Results Analysis

In this study, we compared 10 architectures, 5 defined as frameworks and 5 defined as platforms. Through this comparison, we also evaluated the capabilities of the AHP methods as multi-criterion ranking models. This comparison was made with the aim to find the best architecture that covers the security needs for availability.

The result of the AHP methods, fig. 19, show us the architectures that cover most of the requirements are IBM, IoT-A, Azure, and IIRA. Most of the frameworks from consortium and research group attain a high score as they have a higher level of abstraction and cover a larger spectrum of recommendations, although they don't delve deep into the technical aspect of the implementation. The WSO2 architecture is the least complete in its description and thus doesn't cover all the requirements addressed by ITU. Meanwhile, the platforms' architectures present a more down-to-earth approach, as they can back up the architecture with their actual technology. Hence, they show their real capabilities to conform to the requirements for assuring the availability.

The study didn't take into consideration the following criteria: complexity of implementation and management, cost, scalability, and performance. These criteria were out of the scope of this comparison, but they must be taken into account in order to make the comparison of the platform architectures more complete. The utilization of the AHP method has proven to be successful. We compared with ease the architecture and the method can be extended to include other criteria without complexifying the process.

Conclusion

In our study, we explored a method to compare architectures based on qualitative data. We used the Analytic Hierarchy Process (AHP) to compare reference architectures, either frameworks or platforms, that responded to the requirements of ITU for assuring availability. We conclude that the IBM platform has the best approach for the availability of an IoT system. Although we consider the second-best solution, which is IoT-A for an agnostic approach, we demonstrated the utility of a multi-criteria decision algorithm and how this method can be used in a consulting context for our clients. The result of our study highlighted the most important requirements to achieve in designing availability security services. We plan to integrate and adapt this method into our process during consulting missions.

Blockchain-based Identity and Access Management in Industrial IoT Systems

Introduction

There are three major innovations in the past years: IA (machine learning, deep learning...), big data (data clustering, data analytics...), and distributed systems (IoT, blockchain, micro-services...). IoT systems are at the convergence of at least two of them, big data, and distributed systems, while machine learning can also be a part of IoT, but its use cases are marginal or underexploited but are a vast sector of innovation. IoT systems are complex; they use a mix of legacy and new technology, and keeping a coherent level of security is a real challenge. The majority of IoT devices are considered not secure [146]. Managing such systems is a feat on its own and is the result of years of careful upgrades, transforming them into Frankenstein monsters of information systems. Especially critical systems, such as power grids, e-health, water delivery, or dangerous environment factories, need to be resilient. Components need to communicate without interruptions, their integrity needs to be guaranteed, and all the devices and components are monitored. Those are the major challenges for industrial IoT systems.

In this chapter, we will focus on two of the aspects of managing the IoT system: first is the Identity and Access Management (IAM) and second is preventing man-in-the-middle attacks during firmware updates. The growth of the IoT is inevitable; most estimates are around 5.8 billion devices in 2020 [5], IoT devices will reach congestion due to the sheer volume of devices as we need to identify each device individually. Distributed and complex infrastructure make it difficult for effective management, but [147] state that centralized management systems are too expensive for large networks. Thus, in recent years, due to multiple advancements in distributed technology (cloud computing, blockchain) and industrial ecosystem opening up with the emergence of the platform enterprise paradigm, distributed

systems are the main focus of businesses during the evolution or implementation of new systems. For example, initiatives like the protocol ActivityPub for social networks prove the usability of federated systems that implement a distributed IAM [148] or using blockchain to propagate securely firmware update.

The organization of the remaining chapter is as follows. In section 2, we will present an overview of IAM, firmware update, and blockchain technologies. In section 3, we will explore different approaches for blockchain-based distributed systems. Then, in section 4, we propose a solution to validate the integrity of firmware using blockchain technology in a distributed IoT system.

IAM, Firmware Integrity and Blockchain

Identity and Access Management

Identity and Access Management (IAM) is the association of identity management and access control, accomplishing two main goals. The attribution and orchestration of digital identity of users (admin, operator, developer...), devices (sensors, RFID chips, heavy machinery...), services (web services, applications, databases...) or resources (data, computing power...) and authentication and authorization of these identities. One of the leading challenges for IIoT is the management of the ever-growing number of IoT devices. IAM needs to function "at scale" and in an open ecosystem. IAM is a necessity for securing machine-to-machine communication. IIoT devices need to be uniquely identifiable to establish trust, prevent spoofing, and data corruption. One of the components of IAM is the permissions configuration, each actor must have a set of actions depending on their identities. There are several methods to define an access control, it can be RBAC (Role-Based Access Control) or ABAC (Attribute-Based Access Control) [149], [150]. AA role is a set of actions based on tasks; a network administrator gains access to network resources but can't access development resources. The roles are predefined, and each identity is assigned to them. In Attribute-based access control, the permissions are defined by attributes of an identity, which can be location, features, credentials... For example, a sensor in factory A will have different permissions than the same type of sensor in factory B.

As IoT devices have a short lifespan, the IAM lifecycle needs to be executed more frequently (Provisioning, Authentication, Authorization, Permissions, Self-service, De-provisioning). The provisioning is particularly important in IIoT scenarios, assigning a unique identity to each device requires them to have unique features to differentiate them. These unique features can come from the deployed environment, models, location, function... Then, this new identity needs to be shared to every relevant actor and registered inside their information system. However, in an open, platform-based environment, information can come from different sources with different rules, making it unrealistic to trust each input-output without the presence of a central authority. On the internet, trust comes from the certificate system where certificate authorities deliver identities to websites.

Firmware integrity

One of the key challenges in modern information systems is supervising many devices. As we stated previously, IAM is crucial and the first step of a managed industrial environment, but one of the major pain points in any system is patch management.

IoT devices are often the least secure part of information systems as they are new technologies and tend to be used in experimental usage. Among the classical security flaws any digital component can have, administrators need to be vigilant with IoT devices to secure firmware updates. If the distribution and installation isn't secured, it can lead to: reverse engineering, firmware modification, privilege escalation, unauthorized firmware, or unauthorized device.

Contrary to software patching and updating, IoT devices are difficult, unlike consumer IoT devices where users can easily update through apps. Industrial IoT devices need to be manually updated by downloading the firmware from the manufacturer's website or repository and deploying it in the environment. As sometimes the device can't be accessed physically, most updates are remote, and administrators must ensure that the correct firmware has been installed and that the update was correctly installed. This process can be a source of insecure behavior and sensitive to man-in-the-middle attacks (MitM). This type of attack covers multiple methods; the attacker can usurp the repository or intercept the download to replace the firmware with a compromised one.

Firmware update is critical as innovation pushes insecure IoT devices into the real world, and a faulty or compromised update can lead to life-threatening situations [151]. Nowadays, manufacturers secure their distribution process by following these five conditions [153]:

- **Request for update**: The IoT device receives an external firmware update request from an authorized entity that manages the firmware update operation.
- **Authorized flash driver**: The entity managing the firmware update process is the same as the entity issuing the update.
- **Authentic firmware**: The firmware should be checked to ensure that it has not been altered and that it is compatible.
- **Authorized parts**: Analysis devices and tools guarantee the security of the firmware update process.
- **Rollback mechanism**: An appropriate rollback mechanism must be in place in the event of a failure. This may include an update failure or the detection of malicious or compromised firmware.

Blockchain

Blockchains, in simple terms, are a distributed database. The data is stored inside a block, each block refers to a previously published block through a cryptographic hash of its content. Thus, creating an oriented graph, also called a chain. A block is composed of multiple transactions, which are a data structure containing at least a timestamp and a cryptographic signature from the uploader. Information is replicated in all the nodes of the blockchain using a peer-to-peer protocol.

Blockchain offers many security services [151]. In the following section, we will present the major benefits and constraints of a blockchain system and their impact on an industrial IoT system.

First and foremost, blockchain was created to solve the double-spending problem in a distributed environment. Information, data, or a digital resource can't be duplicated or replicated. For example, in an exploitation system, mutual exclusion (mutex) synchronization mechanisms prevent double utilization of a resource. In the Bitcoin blockchain, the resource is the currency, each amount of currency is owned by an entity, and nobody can claim the ownership of someone else's bitcoins. When someone transfers some of their bitcoin, they lose the ownership of that amount of bitcoin.

Data integrity is the second major benefit of a blockchain system. An effective timestamping mechanism is by design to ensure proper sequencing of the block. The robustness of the data integrity is secured by the consensus algorithm chosen by the implementers of the blockchain. The strength of the consensus is based on an opposed competition in which the actors put investment at stake, whether it is proof of work or proof of stake in a public blockchain. In a private blockchain, trust is based on a contractual agreement.

By being distributed without a trusted third party, the blockchain guarantees the availability of its content and offers censorship-proof capabilities. Interacting with the blockchain only requires a network connection to one of its nodes. This node can be hosted inside a private network or accessed through the internet. In a public blockchain environment, to completely censor an actor, the majority of the participants need to block its participation in the blockchain by not relaying its transactions or blocks, making it incapable of interacting with the blockchain. Attacks have been theorized and some implemented [115], [118], [155], but countermeasures are quickly deployed into the blockchain software. In most cases, the attacked participant can reroute its transaction to a node that will accept its data. Blockchains are particularly suited for adversary environments where actors don't trust each other. This untrust environment guarantees data integrity. If one of the nodes modifies the blockchain, every other node will instantly notice it and reject the modification if it isn't compliant with the consensus rules.

Blockchain has major drawbacks that hinder its adoption by businesses and industries, such as data processing latency, security depending on the number of nodes, and the "append only" approach to data storage. Blockchain use cases must consider these drawbacks and look forward to future evolution. As research progresses in this field, some of these issues will be resolved, notably concerning latency in private blockchains [153], [156].

Blockchains process data slowly, even the fastest ones are slower than traditional centralized systems [157] because the speed of data processing is correlated to security. A public blockchain needs to be slow, the information needs to propagate through the distributed network to synchronize between the nodes. Proof of work consensus strongly secures the chain of blocks. Due to the security constraint, storing data on chain can't be used for real-time use cases. The data is considered to be saved and validated on the blockchain as soon as a sufficient number of blocks are created after their insertion. In the possibility of two concurrent blocks (fork) being created at the same time, the information stored in them is in a state of non-confirmation as the system needs to elect the correct branch. This phenomenon is

called a reorganization, although it is a natural event part of the public blockchain, it nevertheless changes the state of the blockchain. A transaction that is only present in the losing branch must be reintegrated in a future block. In the bitcoin blockchain, a transaction is considered confirmed if there are 6 blocks created after the transaction is inserted in the blockchain, with an average creation time of 10 minutes per block. This means that data is considered to be safely added in the bitcoin blockchain after 60 minutes [158].

A blockchain isn't a traditional CRUD (Create, Read, Update, and Delete) database. Information is only appended, and deletion isn't possible as it will go against the consensus mechanism. The size of the blockchain increases continuously as it is used. IoT devices collect a significant amount of data, and only adding these to the blockchain will saturate the storage of the nodes. There is a more efficient way to use the blockchain by using off-chain functionality, such as a database where data is time-stamped using a merkle tree or any other data structure then stored in the blockchain, saving storage space. For example, backing up only the root of the merkle tree will ensure the integrity of all associated data.

In conclusion, blockchains are suitable for virtually exchanging resources, for timestamping information and for distributed databases. Their security differs between blockchain implementation, private blockchains are easier to set up, manage, and run but the same level of integrity as a standard database and the same level of availability as a standard distributed database. They are more centralized than public databases but have shorter latency. Meanwhile, public blockchains have their own constraints, they have an operational expense as any transaction uploaded in the blockchain needs to pay a small fee in the associated cryptocurrency but they are more decentralized as more actors/nodes are keeping the blockchain and watching its integrity. Private blockchains have better throughput and better latency but are more centralized. Public blockchains are more secure and more decentralized but are slower and more expensive.

Designing use cases for IoT systems using blockchain needs to take into consideration these benefits and constraints. Blockchain isn't the solution to every problem and most of the time it's less effective than standard centralized software.

Related Works

We see a growing interest in the application of blockchain technology in convergence with IoT systems, especially at the dawn of the Industrial Revolution 4.0. [159], [160]. In this

section, we review related publications on the application of blockchain for replacing distributed functions focusing on IAM solutions [161].

Wang et al [162] present an IAM implementation on the Ethereum blockchain where the functions are done through a smart contract. The smart contract manages the identity and the access control directly on chain without intermediary. The access control mechanism used in this contribution is Attribute Based Access Control (ABAC). IoT devices or gateways host a light peer that manages the communication between the system and the blockchain.

An implementation of a distributed Software Define Network (SDN) for IoT is presented by Yazdinejad et al [163]. The authors combine public and private blockchain (Ethereum) in a cluster architecture to configure the routing between multiple IoT sub networks. The public blockchain contains the registry of all the SDN domains and is stored in each cluster head. The role of a cluster head or SDN controller is to be responsible for the activation of the IoT devices. The private blockchain is placed between the IoT device and the SDN controller of a sub network and is used as an authentication and access control. Blockchains securely deliver messages to the SDN controller and users.

Zhang et al [164] propose an implementation of multiple smart contracts providing an on-chain access control to any other smart contract. First, the contract judge evaluates the behavior of the entity that connects to the system and applies a sanction if the behavior is malicious. The second, the Register Contract records information on access control and accessible smart contracts. This implementation makes it possible to secure the monitoring of access management directly on the blockchain.

Dhakal et al [165] built a solution to the integrity problem of firmware update and especially delta updates. The metadata, firmware, and checksum are stored in a private blockchain allowing the device to confront their download version against the version stored on the blockchain.

Lee and Lee [166] propose a solution where a device query a blockchain to determine is its firmware version is up to date, if not the blockchain server transfer a metadata file containing a list of peers which the device can connect to download the update. If the device has the last firmware version, then it will check its integrity against version stored in the blockchain.

Distributed Identity and Access Management

Modern Industrial Internet of Things infrastructures are deployed on multiple locations inside different networks. The IoT devices need to reach databases or services securely. In the era of industry 4.0, a smart factory will be in interaction with diverse actors creating a complex ecosystem where interoperability is a key factor. A distributed IAM framework will solve the problems of consistency between multiple information systems. The blockchain solves the problem of interoperability between system and components [161] and keeps a log of all the operations.

Architecture

The proposed IAM system is composed of an IAM controller, a server hosting the IAM solution, and the blockchain node. We designed this framework to be blockchain agnostic; the blockchain can be either a public or a private one. Public blockchain is an option for scenarios that require a high level of integrity and traceability between multiple actors, but the instructions sent to the IAM systems will have a higher latency. The use of a private blockchain will allow faster instruction throughput and greater customization, but data integrity will be based on trust between participants. In our approach, we decided to choose the Hyperledger private blockchain because it offers more control, smart contract capabilities, and better governance of the blockchain members, which are important criteria for industrial IoT scenarios. Hyperledger by default uses a variant of byzantine fault-tolerant (BFT) consensus where the validating node creating a new block is chosen probabilistically in the network. When a node publishes a new block, it is broadcasted to the network, and each IAM controller reads the new transactions and applies the new instructions. A validation node may be hosted by any participant of the private blockchain. For example, in an industrial context with a production chain involving several smart factories belonging to several companies, each company hosts an IAM controller and a validation node. This ensures the decentralization of the blockchain, considering that if a majority of the validation nodes are controlled by a single entity, the integrity of the blockchain is at risk.

In our system, the blockchain acts as the message bus (inspired from [162]) for the IAM infrastructure and as a log for IAM instructions. The blockchain ensures the traceability of all the IAM functions and guarantees the state synchronicity for each IAM controller. Hyperledger uses a traditional key-value database to store the state of the blockchain. This database allows external applications to consult the data faster than by going through all the blocks. A translation layer is necessary to translate the data from the database to the IAM solution. The identity register is built based on the state of the blockchain and can be recreated by scanning the blockchain from the origin block. Sharing a common identity registry securely opens the industrial network to additional authenticated companies.

Each IAM controller has a unique identity. A transaction is either addressed to every controller to set a global configuration for example or a specified number of controllers, allowing to restrict the actions of an identity to specific subdomains. Instructions sent through the blockchain have their integrity and availability and non-repudiability guaranteed by the intrinsic functionality of blockchain technology.

In our proposition, the IAM controller performs 5 actions (tab 1), based on [161]: Provisioning (Creation of an identity), Update (Modification of an identity), Revocation (Deletion of an identity), Lookup (Verification of the presence of an identity), Evaluate (Authentication of an identity). Action orders are issued through the blockchain and are executed by the IAM controller. Each of these actions is traceable on the blockchain, where the identity of the issuer is stored.

Scenario

In this section, we will present how various scenarios unfold, underpinning the interaction between the components of the proposed blockchain-based IAM framework.

Provisioning a new identity

An administrator collects the unique identifier of a new IoT device before deploying it. Based on this information and the context of interaction, permissions, and fields of operation are associated with the device. Then, the administrator sends a new transaction on the blockchain containing the instructions to the IAM controller. The transaction is signed by the administrator, and every blockchain node will check the integrity of the data. In the case where the IoT device is stationary, the transaction will contain a field specifying the network

domain that will host the device, meaning only the IAM controller will save the identity in their database, but every blockchain node will have a record of the identity.

Device request access to a resource

A device needs to access a database; it will communicate directly with the IAM controller through standard communication protocols. First, the IAM controller will authenticate the device and check the permission associated with its identity. Once the device is authorized to access the database, it collects the needed information.

Deployment of a new IAM controller

When an IAM controller is deployed in a new domain, two things need to be initiated: the blockchain node and identity registry. First, the blockchain node will download the blockchain from the rest of the network, and this process will recreate the identity registry. The IAM controller is configured with a unique identifier linked to the domain it belongs to. This parameter is used to determine the segmentation in the network and create virtual subdomains.

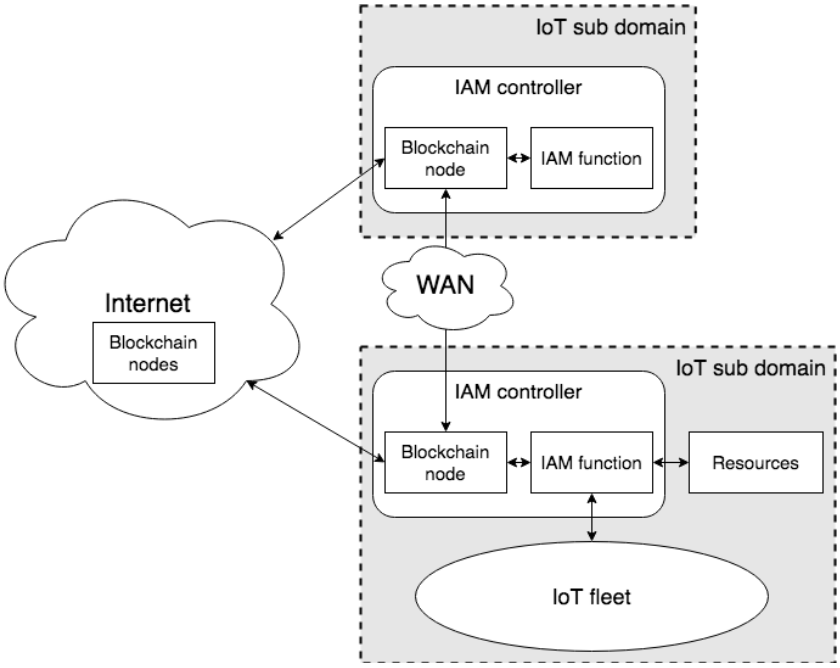

Fig. 20: Diagram of the IAM infrastructure

Conclusion

In our proposition, we didn't address the confidentiality inside a transaction. For example, a message may be encrypted before being added to the blockchain and only a participant in the information system will decrypt it. Encryption isn't a trivial implementation and takes into consideration multiple parameters such as the keys exchange, symmetric or asymmetric encryption, key storage… The IAM function could be done directly inside a smart contract, but not all the blockchains have the capabilities to execute complex smart contracts; we choose to propose an agnostic solution. A blockchain like Bitcoin has strict rules for its native smart contract, only a limited set of functions are available, while the Ethereum or Hyperledger blockchains offer Turing complete programming language. Using the smart contract increases the security of the system, ensures the execution of the IAM function will be directly saved on the blockchain, and reduces the number of components needed for an IAM framework. We presented an implementation of blockchain for a distributed IAM system and the benefits and drawbacks of such technology. A blockchain is a useful tool in scenarios with multiple shareholders that need to keep accountable to each other and isn't a miracle solution. It's a complex answer to specific needs. Our solution uses the blockchain as a message bus to transmit IAM instructions securely across multiple environments.

Secure IoT Firmware Update Through Blockchain Technology

Introduction

The proposed solution is a blockchain system aimed at providing a secure environment for IoT device firmware update. Although official communication channels are reliable and secure, as the expansion of deployed IoT devices increases, the appeal of injecting malicious code along the supply chain increases, for example, by introducing a backdoor into an open-source library used in a firmware or downstream by spoofing or intercepting the update. Our approach is to prevent and detect Man-in-the-Middle attacks during the transfer of the update. Therefore, our system is declined into two functions: Prevention by using asymmetric

cryptography and blockchain. Detection by analyzing the update behavior process and detecting a potential tampered firmware update. In this section, we use the nomenclature SUIT [159] established by an IETF working group.

Architecture

In order to solve the problem, we propose an architecture where the blockchain is the keystone. The roles of the blockchain are to verify the identity of the update issuer, the integrity of the update, and to validate and log the result of the update process. Therefore, our proposed system can be defined by 3 main components:

- **The Manufacturer**, issues and hosts the firmware update.
- **The Smart Contract**, verifies the Manufacturer identity and the integrity of the update. Notifies the IoT devices and validates the successful conduct of the update.
- **The IoT Device**, receives notification of a new update, downloads the new firmware, and sends back metadata on the update process to the Smart Contract.

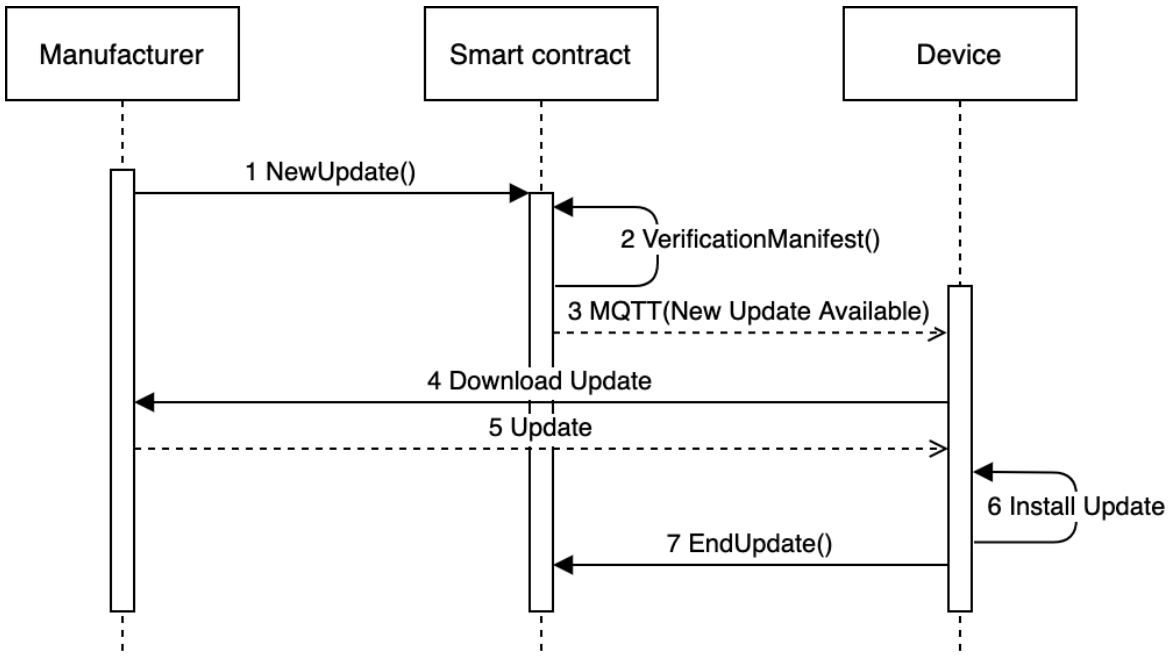

Fig. 21: Firmware update process

Attack prevention

To prevent man in the middle attacks, we use the inherent functionalities of the blockchain to log the update process. The manufacturer owns a private key which is used to sign its interaction with the blockchain. With this signature, we keep a trace of every action from the

manufacturer and ensure its identity. In the initial phase of the deployment of a new firmware update, the manufacturer needs to create a manifest containing all the metadata concerning this update. The metadata are [160]:

- Version ID, an identifier of the version format of the manifest.
- Monotonic Sequence Number, an ever-increasing number determining the firmware version.
- Class ID, a set of information determining the targets of the update such as model name or number, hardware revision, runtime library version, bootloader version, ROM revision, silicon batch number.
- Payload Indicator, a URI where devices can download the update.
- Payload Format, the file format of the update.
- Payload Digest, a hash of the new firmware.
- Size, the size of the update.

The update process is as follows: the manufacturer first sends the manifest containing the information about the new firmware update to the smart contract host on the blockchain. The smart contract will verify the manifest by checking the identity of the sender and will check the cohesiveness of the data stored in the manifest such as the Monotonic Sequence Number is higher than the one in the previous manifest. If the manifest is valid, the smart contract sends it in a message aimed at the device with matching Class ID. Once a device receives the manifest from the smart contract, it will access the URI present in the payload indicator and download the update. When the download is completed, the device executes quick verification by comparing the size of the file, the extension, and the hash of the new downloaded file with the information inside the manifest. The device also collects data on the file transfer such as the number of IP packets, the average number of TTL. The collected network data and the result of the verification are hashed and sent back to the smart contract. The smart contract will compare the hash up to the last 10 updates. If no anomaly is detected, the update is recorded on the blockchain.

Attack Detection

To detect a man in the middle attack during the download of the update, we decided to use the method Local Sensitivity Hashing (LSH) which allows us to compare a set of hashed data to previously recorded hashed data and evaluate a similarity factor. The LSH algorithm is used in machine learning applications to find nearest neighbors while reducing the data to easy to compare hash values. This hash has the property of representing a set of data points and keeping their values. Which means that multiple vectors of data points with close values will

generate similar or close hashes. As opposed to traditional hashing methods where hashes vary greatly even with close values.

In our design, the device will calculate LSH from the data collected from the update process. The information collected is file size, number of IP packets, and the average number of TTL. The device sends the hash to the smart contract. The smart contract will calculate the similarity between the received hash and the 10 last hashes for this device's Class ID. If MinHash similarity is found, then the download wasn't victim of a man in the middle attack. The hash ends up being logged in the blockchain to serve as the next comparison element for future MinHash similarity.

Experimentation

For our proposition, we need a dataset of firmware to perform our tests. We use the data from Costin & al [168] which gives us a large set of firmware, their download URLs and their sizes. We base our model on the assumption that the device will update its whole system hence, we extracted from the dataset only firmwares in zip and tar.gz file formats as they are used in the cases of complete update contrary to differential updates which use patch files. We reduced the result by filtering by manufacturer based on the domain name in the URL and select the 30 most occurring manufacturers giving us a sample size of 40,415 firmwares. From the reduced dataset, we extracted a sequence of firmware updates for a device model. Those sequences will be the foundation of our experimentation as history of past updates.

	Filename	URL	Filesize
2	v26g0256.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.6/v26g0256.zip	3240257
3	v26ga256.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.6/v26ga256.zip	3244054
4	v26gb256.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.6/v26gb256.zip	3226091
5	v26gc256.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.6/v26gc256.zip	3230015
6	v26gi256.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.6/v26gi256.zip	3248027
7	v26gk256.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.6/v26gk256.zip	3240283
8	v26gy256.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.6/v26gy256.zip	3245865
9	v26gz256.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.6/v26gz256.zip	3242345
10	v26g0257.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.7/v26g0257.zip	3295721
11	v26ga257.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.7/v26ga257.zip	3303954
12	v26gb257.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.7/v26gb257.zip	3282271
13	v26gc257.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.7/v26gc257.zip	3285756
14	v26gi257.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.7/v26gi257.zip	3303225
15	v26gk257.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.7/v26gk257.zip	3295438
16	v26gn257.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.7/v26gn257.zip	3281880
17	v26gy257.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.7/v26gy257.zip	3301249
18	v26gz257.zip	ftp.draytek.com/Vigor2600Gplus/Firmware/2.5.7/v26gz257.zip	3297596

Fig. 22: Extract of the Firmware dataset

We used the LSH implementation Floky [167] to execute our software. First, we tested similarity on the dataset by querying samples from the dataset. We found that an update will

find multiple similarities with previous sets of data. Due to time constraints, additional testing could not be conducted.

Conclusion

During our investigation of the use of blockchain technology for IIoT systems, we demonstrated the benefits blockchain can bring to such constrained environments. Blockchain offers immutability of data stored on the chain which enables timestamping, availability, and integrity of information. We proposed a solution taking advantage of these characteristics for managing identity and access where IoT devices prove their identity by addressing the blockchain and administrators store identity and access policies on the blockchain. Then we proposed a solution for securely delivering firmware updates by using the inherent cryptography of blockchain technology and machine learning algorithms to detect man-in-the-middle attacks during the transfer.

Conclusion

The industrial world is vast and demanding, and the IT systems that compose it must meet its needs. IoT devices form the bridge between the physical and virtual worlds, providing the intelligence necessary for the smooth running of industries. Whether it's through the monitoring of a production line, the piloting of public infrastructure, or E-health, IoT devices improve the collection of information and automate processes. IoT is not just a passing fad; it's a set of paradigms and technologies that already exist but are being pushed to the extreme. Devices are getting smaller and smaller but collecting more and more data, complicating the management of such systems.

We conducted an evaluation of architectures for the IoT based on the ITU architecture requirements. Initially, we observed whether an architecture met these requirements and its coverage rate. But this approach proved to be flawed by not taking into account the specific requirements of the use cases. Telemedicine does not have the same requirements as a solution for industry. We had to revise our approach and classify the 48 requirements as either vital or non-vital in relation to use cases.

The identification and classification of the security vulnerabilities of industrial IoT systems was done after a bibliographic work including more than fifty scientific publications. The objective was to establish a consensus on the state of scientific knowledge. The result of this work is compiled in a scientific article.

We present our results according to 2 axes:

- Logical layers
- Security services

The layer approach continues the work done in the previous contribution on architectures and allows to cover the scope of an industrial information system. Each layer is associated with the different devices or software that can be affected by cyber attacks.

The security services have been chosen by the essential needs of IoT systems and the industrial environment.

We directed our research towards the uses of blockchain to secure Identity and Authentication Manager (IAM) operations in an industrial IoT environment.

The IAM system we propose combines an IAM controller, a server hosting the IAM solution and a blockchain. We have designed this framework to be blockchain agnostic, where the blockchain can be public or private. The public blockchain is an option for scenarios that require a high level of integrity and traceability between multiple actors, but the instructions sent to the IAM systems will have a higher latency.

We explored a new avenue of thinking about the possibilities of blockchain to improve the security of Industrial IoTs. The proposed solution is a blockchain system to provide a secure environment for updating the firmware of IoT devices. Although formal communication channels are reliable and secure, as the expansion of deployed IoT devices increases, the attractiveness of injecting malicious code along the supply chain increases, e.g. by introducing a backdoor into an open-source library used in a firmware or downstream by spoofing or intercepting the update. Our approach is to prevent and detect Man-in-the-Middle attacks. Thus, our system has two functions: prevention by using asymmetric cryptography and blockchain. Detection by analyzing the update behavior process and detecting a potential corrupted firmware update.

Blockchain technology offers new possibilities to secure information systems. However, the constraints are major blockers for a democratization. The blockchain technology is suitable for systems with several actors where the information has to be distributed. Therefore the uses of blockchain, although revolutionary in its concepts, its applications in the industrial world remain niche.

Bibliography

[1] C. E. Landwehr, “Computer security,” *IJIS*, vol. 1, no. 1, pp. 3–13, Aug. 2001, doi: 10.1007/s102070100003.

[2] C. Maple, “Security and privacy in the internet of things,” *Journal of Cyber Policy*, vol. 2, no. 2, pp. 155–184, 2017, doi: https://doi.org/10.1080/23738871.2017.1366536.

[3] C. Metz, “AAA protocols: authentication, authorization, and accounting for the Internet,” *IEEE Internet Computing*, vol. 3, no. 6, pp. 75–79, 1999.

[4] A. Abid, M. T. Khemakhem, S. Marzouk, M. B. Jemaa, T. Monteil, and K. Drira, “Toward Antifragile Cloud Computing Infrastructures,” *Procedia Computer Science*, vol. 32, pp. 850–855, 2014, doi: 10.1016/j.procs.2014.05.501.

[5] Gartner, “Gartner Says 5.8 Billion Enterprise and Automotive IoT Endpoints Will Be in Use in 2020,” *Gartner*, Aug. 29, 2019. https://www.gartner.com/en/newsroom/press-releases/2019-08-29-gartner-says-5-8-billion-enterprise-and-automotive-io (accessed Oct. 28, 2019).

[6] K. Schwab, *The fourth industrial revolution*. Currency, 2017.

[7] C.-W. Ten, C.-C. Liu, and G. Manimaran, “Vulnerability assessment of cybersecurity for SCADA systems,” *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1836–1846, 2008.

[8] C. Greer, M. Burns, D. Wollman, and E. Griffor, “Cyber-physical systems and internet of things,” National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 1900-202, Mar. 2019. doi: 10.6028/NIST.SP.1900-202.

[9] D. Serpanos and M. Wolf, “Industrial Internet of Things,” in *Internet-of-Things (IoT) Systems: Architectures, Algorithms, Methodologies*, D. Serpanos and M. Wolf, Eds. Cham: Springer International Publishing, 2018, pp. 37–54. doi: 10.1007/978-3-319-69715-4_5.

[10] “Amazon Robotics,” *Amazon Robotics*. https://www.amazonrobotics.com/#/ (accessed Mar. 20, 2019).

[11] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, “Industrial Internet of Things: Challenges, Opportunities, and Directions,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018, doi: 10.1109/TII.2018.2852491.

[12] G. Spathoulas and S. Katsikas, “Towards a Secure Industrial Internet of Things,” in *Security and Privacy Trends in the Industrial Internet of Things*, C. Alcaraz, Ed. Cham: Springer International Publishing, 2019, pp. 29–45. doi: 10.1007/978-3-030-12330-7_2.

[13] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[14] P. P. Ray, “A survey on Internet of Things architectures,” *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, Jul. 2018, doi: 10.1016/j.jksuci.2016.10.003.

[15] C. Li and B. Palanisamy, “Privacy in Internet of Things: from Principles to Technologies,” *IEEE Internet of Things Journal*, Jul. 2018, Accessed: Aug. 13, 2018. [Online]. Available: http://d-scholarship.pitt.edu/35103/

[16] H. Rahimi, A. Zibaeenejad, and A. A. Safavi, “A Novel IoT Architecture based on 5G-IoT and Next Generation Technologies,” *arXiv preprint arXiv:1807.03065*, 2018.
[17] Miao Wu, Ting-Jie Lu, Fei-Yang Ling, Jing Sun, and Hui-Ying Du, “Research on the architecture of Internet of Things,” in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, Chengdu, China, Aug. 2010, pp. V5-484-V5-487. doi: 10.1109/ICACTE.2010.5579493.
[18] R. Duan, X. Chen, and T. Xing, “A QoS Architecture for IOT,” in *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, Dalian, China, Oct. 2011, pp. 717–720. doi: 10.1109/iThings/CPSCom.2011.125.
[19] Z. Yang, Y. Yue, Y. Yang, Y. Peng, X. Wang, and W. Liu, “Study and application on the architecture and key technologies for IOT,” in *2011 International Conference on Multimedia Technology*, Jul. 2011, pp. 747–751. doi: 10.1109/ICMT.2011.6002149.
[20] D. P. Abreu, K. Velasquez, M. Curado, and E. Monteiro, “A resilient Internet of Things architecture for smart cities,” *Ann. Telecommun.*, vol. 72, no. 1–2, pp. 19–30, Feb. 2017, doi: 10.1007/s12243-016-0530-y.
[21] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, “Security of the Internet of Things: perspectives and challenges,” *Wireless Networks*, vol. 20, no. 8, pp. 2481–2501, Nov. 2014, doi: 10.1007/s11276-014-0761-7.
[22] S. K. Datta, C. Bonnet, and N. Nikaein, “An IoT gateway centric architecture to provide novel M2M services,” in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, 2014, pp. 514–519.
[23] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, “Internet of things (IoT) security: Current status, challenges and prospective measures,” in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, London, United Kingdom, Dec. 2015, pp. 336–341. doi: 10.1109/ICITST.2015.7412116.
[24] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions,” *arXiv:1207.0203 [cs]*, Jul. 2012, Accessed: Sep. 12, 2018. [Online]. Available: http://arxiv.org/abs/1207.0203
[25] H. Suo, J. Wan, C. Zou, and J. Liu, “Security in the internet of things: a review,” in *Computer Science and Electronics Engineering (ICCSEE), 2012 international conference on*, 2012, vol. 3, pp. 648–651.
[26] L. Da Xu, W. He, and S. Li, “Internet of things in industries: A survey,” *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.
[27] A. F. A. Rahman, M. Daud, and M. Z. Mohamad, “Securing Sensor to Cloud Ecosystem using Internet of Things (IoT) Security Framework,” in *Proceedings of the International Conference on Internet of things and Cloud Computing - ICC ’16*, Cambridge, United Kingdom, 2016, pp. 1–5. doi: 10.1145/2896387.2906198.
[28] G. Fortino, A. Guerrieri, W. Russo, and C. Savaglio, “Integration of agent-based and Cloud Computing for the smart objects-oriented IoT,” in *Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Hsinchu, Taiwan, May 2014, pp. 493–498. doi: 10.1109/CSCWD.2014.6846894.
[29] Qian Xiaocong and Zhang Jidong, “Study on the structure of ‘Internet of Things(IOT)’ business operation support platform,” in *2010 IEEE 12th International Conference on Communication Technology*, Nanjing, China, Nov. 2010, pp. 1068–1071. doi: 10.1109/ICCT.2010.5688537.
[30] I. Yaqoob *et al.*, “Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges,” *IEEE wireless communications*, vol. 24, no. 3, pp. 10–16, 2017.

[31] L. Tan and N. Wang, “Future internet: The internet of things,” in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, 2010, vol. 5, pp. V5-376.
[32] D. Bandyopadhyay and J. Sen, “Internet of Things: Applications and Challenges in Technology and Standardization,” *Wireless Personal Communications*, vol. 58, no. 1, pp. 49–69, May 2011.
[33] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, “Future internet: the internet of things architecture, possible applications and key challenges,” in *Frontiers of Information Technology (FIT), 2012 10th International Conference on*, 2012, pp. 257–260.
[34] C.-L. Zhong, Z. Zhu, and R.-G. Huang, “Study on the IOT architecture and gateway technology,” in *Distributed Computing and Applications for Business Engineering and Science (DCABES), 2015 14th International Symposium on*, 2015, pp. 196–199.
[35] M. Ashi and T. Rees, “System security versus system reliability-similarities and differences”.
[36] J. Guth, U. Breitenbücher, M. Falkenthal, F. Leymann, and L. Reinfurt, “Comparison of IoT platform architectures: A field study based on a reference architecture,” in *2016 Cloudification of the Internet of Things (CIoT)*, Nov. 2016, pp. 1–6. doi: 10.1109/CIOT.2016.7872918.
[37] “Overview of the Internet of things.” ITU-T, Jun. 2012.
[38] N. Jansma and B. Arrendondo, “Performance Comparison of Elliptic Curve and RSA Digital Signatures,” p. 20, 2004.
[39] E. Vasilomanolakis, J. Daubert, M. Luthra, V. Gazis, A. Wiesmaier, and P. Kikiras, “On the security and privacy of internet of things architectures and systems,” in *2015 International Workshop on Secure Internet of Things (SIoT)*, 2015, pp. 49–57.
[40] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in Internet of Things: The road ahead,” *Computer networks*, vol. 76, pp. 146–164, 2015.
[41] J. Won, A. Singla, E. Bertino, and G. Bollella, “Decentralized Public Key Infrastructure for Internet-of-Things,” in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, Oct. 2018, pp. 907–913. doi: 10.1109/MILCOM.2018.8599710.
[42] M. Eldefrawy, N. Pereira, and M. Gidlund, “Key Distribution Protocol for Industrial Internet of Things without Implicit Certificates,” *IEEE Internet of Things Journal*, 2018, Accessed: Aug. 27, 2018. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:miun:diva-34278
[43] M. Ma, D. He, N. Kumar, K.-K. R. Choo, and J. Chen, “Certificateless searchable public key encryption scheme for industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 759–767, 2018.
[44] A. K. Das, S. Zeadally, and D. He, “Taxonomy and analysis of security protocols for Internet of Things,” *Future Generation Computer Systems*, vol. 89, pp. 110–125, Dec. 2018, doi: 10.1016/j.future.2018.06.027.
[45] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, “Security and privacy challenges in industrial internet of things,” in *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*, 2015, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7167238/
[46] D. Ratasich, F. Khalid, F. Geissler, R. Grosu, M. Shafique, and E. Bartocci, “A Roadmap Towards Resilient Internet of Things for Cyber-Physical Systems,” *IEEE Access*, pp. 1–1, 2019, doi: 10.1109/ACCESS.2019.2891969.
[47] J.-C. Laprie, “From dependability to resilience,” in *38th IEEE/IFIP Int. Conf. On dependable systems and networks*, 2008, pp. G8–G9.

[48] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, “Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges,” *IEEE Access*, 2017.
[49] K. A. Kumar and K. Ramudu, “Precision Agriculture using Internet of Things and Wireless sensor Networks,” vol. 07, no. 03, p. 5, 2019.
[50] M. Lee, J. Hwang, and H. Yoe, “Agricultural production system based on IoT,” in *2013 IEEE 16th International Conference on Computational Science and Engineering*, 2013, pp. 833–837.
[51] C. Adaros Boye, P. Kearney, and M. Josephs, “Cyber-Risks in the Industrial Internet of Things (IIoT): Towards a Method for Continuous Assessment,” in *Information Security*, 2018, pp. 502–519.
[52] J. Lee, H.-A. Kao, and S. Yang, “Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment,” *Procedia CIRP*, vol. 16, pp. 3–8, Jan. 2014, doi: 10.1016/j.procir.2014.02.001.
[53] M. N. Aman, B. Sikdar, K. C. Chua, and A. Ali, “Low Power Data Integrity in IoT Systems,” *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3102–3113, Aug. 2018, doi: 10.1109/JIOT.2018.2833206.
[54] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, “Blockchain based data integrity service framework for IoT data,” in *2017 IEEE International Conference on Web Services (ICWS)*, 2017, pp. 468–475.
[55] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IoT: Mirai and other botnets,” *Computer*, vol. 50, no. 7, pp. 80–84, 2017.
[56] R. M. Lee, M. J. Assante, and T. Conway, “German steel mill cyber attack,” *Industrial Control Systems*, vol. 30, p. 62, 2014.
[57] J. Robertson and M. Riley, “China Used a Tiny Chip in a Hack That Infiltrated U.S. Companies,” *Bloomberg.com*, Oct. 04, 2018. Accessed: Jun. 24, 2019. [Online]. Available: https://www.bloomberg.com/news/features/2018-10-04/the-big-hack-how-china-used-a-tiny-chip-to-infiltrate-america-s-top-companies
[58] “Guide Achieving Observability.” HoneyComb.io, Jul. 2018. Accessed: Oct. 04, 2019. [Online]. Available: https://www.honeycomb.io/wp-content/uploads/2018/07/Honeycomb-Guide-Achieving-Observability-v1.pdf
[59] C. Sridharan, *Distributed Systems Observability: A Guide to Building Robust Systems*. O’Reilly Media, 2018.
[60] ITU-T, “SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS : Common requirements of the Internet of things.” ITU-T, Jun. 2014.
[61] “CYBER; Cyber Security for Consumer Internet of Things.” ETSI, Feb. 2019.
[62] N. Falliere, L. O. Murchu, and E. Chien, “W32. stuxnet dossier,” *White paper, Symantec Corp., Security Response*, vol. 5, no. 6, p. 29, 2011.
[63] B. Turnbull, “Cyber-resilient Supply Chains: Mission Assurance in the Future Operating Environment,” *Army*, p. 41, 2018.
[64] T. Omitola and G. Wills, “Towards Mapping the Security Challenges of the Internet of Things (IoT) Supply Chain,” *Procedia Computer Science*, vol. 126, pp. 441–450, 2018.
[65] “The Strangest Technology Story Of The Year | Bruceb News,” *Bruceb news*, Jan. 09, 2019. https://www.brucebnews.com/2019/01/bloomberg-and-chinese-spies-the-strangest-technology-story-of-the-year/ (accessed Jun. 24, 2019).
[66] Akamai, “State of the Internet / Security Q4 2016,” 2016. https://blogs.akamai.com/2017/02/state-of-the-internet-security-q4-2016.html (accessed Apr. 18, 2017).

[67] A. S. Sastry, S. Sulthana, and S. Vagdevi, “Security threats in wireless sensor networks in each layer,” *International Journal of Advanced Networking and Applications*, vol. 4, no. 4, p. 1657, 2013.
[68] E. Shi and A. Perrig, “Designing secure sensor networks,” *IEEE Wireless Communications*, vol. 11, no. 6, pp. 38–43, 2004.
[69] T. Bhattasali, R. Chaki, and S. Sanyal, “Sleep Deprivation Attack Detection in Wireless Sensor Network,” *International Journal of Computer Applications*, vol. 975, p. 8887.
[70] M. Brownfield, Y. Gupta, and N. Davis, “Wireless sensor network denial of sleep attack,” in *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, 2005, pp. 356–364.
[71] A. Gallais, T.-H. Hedli, V. Loscri, and N. Mitton, “Denial-of-Sleep Attacks against IoT Networks,” presented at the CoDIT 2019 - 6th International Conference on Control, Decision and Information Technologies, Paris, France, Apr. 2019, p. 6.
[72] D. R. Raymond and S. F. Midkiff, “Denial-of-service in wireless sensor networks: Attacks and defenses,” *IEEE Pervasive Computing*, no. 1, pp. 74–81, 2008.
[73] J. R. Douceur, “The Sybil Attack,” in *Peer-to-Peer Systems*, Mar. 2002, pp. 251–260. doi: 10.1007/3-540-45748-8_24.
[74] J. Newsome, E. Shi, D. Song, and A. Perrig, “The sybil attack in sensor networks: analysis & defenses,” in *Third international symposium on information processing in sensor networks, 2004. IPSN 2004*, 2004, pp. 259–268.
[75] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos, “Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things,” in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 606–611.
[76] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Packet leashes: A defense against wormhole attacks in wireless ad hoc networks,” in *Proceedings of INFOCOM*, 2003, vol. 2003.
[77] M. J. Covington and R. Carskadden, “Threat implications of the internet of things,” in *2013 5th International Conference on Cyber Conflict (CYCON 2013)*, 2013, pp. 1–12.
[78] J. Mirkovic and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.
[79] M. Conti, N. Dragoni, and V. Lesyk, “A Survey of Man In The Middle Attacks,” *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2027–2051, thirdquarter 2016, doi: 10.1109/COMST.2016.2548426.
[80] Q. Xu, P. Ren, H. Song, and Q. Du, “Security enhancement for IoT communications exposed to eavesdroppers with uncertain locations,” *IEEE Access*, vol. 4, pp. 2840–2853, 2016.
[81] A. Arsenault and S. Farrell, “Securely available credentials-requirements,” IETF, RFC 3157, 2001.
[82] K. S. Tep, B. Martini, R. Hunt, and K.-K. R. Choo, “A taxonomy of cloud attack consequences and mitigation strategies: The Role of Access Control and Privileged Access Management,” in *2015 IEEE Trustcom/BigDataSE/ISPA*, 2015, vol. 1, pp. 1073–1080.
[83] K. Hashizume, D. G. Rosado, E. Fernández-Medina, and E. B. Fernandez, “An analysis of security issues for cloud computing,” *J Internet Serv Appl*, vol. 4, no. 1, p. 5, Feb. 2013, doi: 10.1186/1869-0238-4-5.
[84] W. G. Halfond, J. Viegas, and A. Orso, “A classification of SQL-injection attacks and countermeasures,” in *Proceedings of the IEEE International Symposium on Secure Software Engineering*, 2006, vol. 1, pp. 13–15.

[85] S. Gupta, A. Singhal, and A. Kapoor, “A literature survey on social engineering attacks: Phishing attack,” in *2016 international conference on computing, communication and automation (ICCCA)*, 2016, pp. 537–540.
[86] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System.” 2009.
[87] T. C. May, “The crypto anarchist manifesto [Electronic mailing list message],” *activism. net/cypherpunk/crypto-anarchy. html*, 1992.
[88] “Pizza for bitcoins?” https://bitcointalk.org/index.php?topic=137.0 (accessed Sep. 16, 2020).
[89] L. Lamport, R. Shostak, and M. Pease, “The Byzantine generals problem,” *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.
[90] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, “Where Is Current Research on Blockchain Technology?—A Systematic Review,” *PLoS ONE*, vol. 11, no. 10, pp. 1–27, Oct. 2016, doi: 10.1371/journal.pone.0163477.
[91] S. King and S. Nadal, “Ppcoin: Peer-to-peer crypto-currency with proof-of-stake,” *self-published paper, August*, vol. 19, 2012, [Online]. Available: http://peerco.in/assets/paper/peercoin-paper.pdf
[92] “Nxt - The Blockchain Application Platform.” https://nxtplatform.org/ (accessed Aug. 28, 2017).
[93] M. Castro, B. Liskov, and others, “Practical Byzantine fault tolerance,” in *OSDI*, 1999, vol. 99, pp. 173–186. Accessed: Apr. 05, 2017. [Online]. Available: https://www.usenix.org/events/osdi99/full_papers/castro/castro_html/castro.html
[94] D. Larimer, “Delegated proof-of-stake (dpos),” *Bitshare whitepaper*, 2014.
[95] I. Eyal and E. G. Sirer, “Majority Is Not Enough: Bitcoin Mining Is Vulnerable,” in *Financial Cryptography and Data Security*, Mar. 2014, pp. 436–454. doi: 10.1007/978-3-662-45472-5_28.
[96] “Understanding The DAO Attack,” *CoinDesk*, Jun. 25, 2016. http://www.coindesk.com/understanding-dao-hack-journalists/ (accessed Jun. 14, 2017).
[97] “Sweden tests blockchain technology for land registry,” *Reuters*, Jun. 16, 2016. [Online]. Available: http://www.reuters.com/article/us-sweden-blockchain-idUSKCN0Z22KV
[98] L. Shin, “The First Government To Secure Land Titles On The Bitcoin Blockchain Expands Project,” *Forbes*. http://www.forbes.com/sites/laurashin/2017/02/07/the-first-government-to-secure-land-titles-on-the-bitcoin-blockchain-expands-project/ (accessed Jun. 14, 2017).
[99] S. Thomsen, “Alibaba wants to use blockchain to prevent counterfeit Australian food products in China,” *Business Insider Australia*, Mar. 24, 2017. https://www.businessinsider.com.au/alibaba-wants-to-use-blockchain-to-prevent-counterfeit-australian-food-products-in-china-2017-3 (accessed Mar. 28, 2017).
[100] “Wal-Mart Tackles Food Safety With Trial of Blockchain,” *Bloomberg.com*, Nov. 18, 2016. Accessed: Mar. 28, 2017. [Online]. Available: https://www.bloomberg.com/news/articles/2016-11-18/wal-mart-tackles-food-safety-with-test-of-blockchain-technology
[101] “Everledger Plans Blockchain Database to Combat Art Fraud,” *CoinDesk*, May 02, 2016. http://www.coindesk.com/everledger-announces-partnership-vastari-combat-art-fraud/ (accessed Jun. 14, 2017).
[102] “Blockchain: SACEM, ASCAP and PRS FOR MUSIC join forces to improve the...” https://societe.sacem.fr/en/news/authors-rights/blockchain-sacem-ascap-and-prs-for-music-join-forces-to-improve-the-identification-of-the-works (accessed Jun. 14, 2017).

[103] K. Christidis and M. Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things,” *IEEE Access*, vol. 4, pp. 2292–2303, 2016, doi: 10.1109/ACCESS.2016.2566339.
[104] “Inside the Bold Attempt to Reverse a $55 Million Digital Heist,” *Bloomberg.com*. Accessed: Jun. 14, 2017. [Online]. Available: https://www.bloomberg.com/features/2017-the-ether-thief/
[105] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, “On the Security and Performance of Proof of Work Blockchains,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA, 2016, pp. 3–16. doi: 10.1145/2976749.2978341.
[106] B. Johnson, A. Laszka, J. Grossklags, M. Vasek, and T. Moore, “Game-Theoretic Analysis of DDoS Attacks Against Bitcoin Mining Pools,” in *Financial Cryptography and Data Security*, Mar. 2014, pp. 72–86. doi: 10.1007/978-3-662-44774-1_6.
[107] A. L. Calvez, “How to destroy bitcoins?,” *Antoine Le Calvez*, Nov. 16, 2015. https://medium.com/@alcio/how-to-destroy-bitcoins-255bb6f2142e (accessed Mar. 06, 2017).
[108] A. Kiayias and G. Panagiotakos, “Speed-Security Tradeoffs in Blockchain Protocols,” 2015, Accessed: Mar. 07, 2017. [Online]. Available: http://www.research.ed.ac.uk/portal/en/publications/speedsecurity-tradeoffs-in-blockchain-protocols(141c78de-df5e-4afe-ab82-1cf81039e7dc).html
[109] A. Gervais, S. Capkun, G. O. Karame, and D. Gruber, “On the Privacy Provisions of Bloom Filters in Lightweight Bitcoin Clients,” in *Proceedings of the 30th Annual Computer Security Applications Conference*, New York, NY, USA, 2014, pp. 326–335. doi: 10.1145/2664243.2664267.
[110] I. Miers, C. Garman, M. Green, and A. D. Rubin, “Zerocoin: Anonymous distributed e-cash from bitcoin,” in *Security and Privacy (SP), 2013 IEEE Symposium on*, 2013, pp. 397–411. Accessed: Apr. 05, 2017. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/6547123/
[111] A. Miller, M. Möser, K. Lee, and A. Narayanan, “An Empirical Analysis of Linkability in the Monero Blockchain,” *arXiv preprint arXiv:1704.04299*, 2017, [Online]. Available: https://arxiv.org/abs/1704.04299
[112] A. Kumar, C. Fischer, S. Tople, and P. Saxena, “A Traceability Analysis of Monero’s Blockchain.,” *IACR Cryptology ePrint Archive*, vol. 2017, p. 338, 2017.
[113] K. J. O’Dwyer and D. Malone, “Bitcoin mining and its energy footprint,” 2014, Accessed: Mar. 28, 2017. [Online]. Available: http://digital-library.theiet.org/content/conferences/10.1049/cp.2014.0699
[114] M. Vukolić, “The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication,” in *Open Problems in Network Security*, Oct. 2015, pp. 112–125. doi: 10.1007/978-3-319-39028-4_9.
[115] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, “Eclipse Attacks on Bitcoin’s Peer-to-peer Network,” in *Proceedings of the 24th USENIX Conference on Security Symposium*, Berkeley, CA, USA, 2015, pp. 129–144. Accessed: Mar. 06, 2017. [Online]. Available: http://dl.acm.org/citation.cfm?id=2831143.2831152
[116] “Weaknesses - Bitcoin Wiki.” https://en.bitcoin.it/wiki/Weaknesses#Sybil_attack (accessed Mar. 03, 2017).
[117] M. Andrychowicz, S. Dziembowski, D. Malinowski, and \Lukasz Mazurek, “On the malleability of bitcoin transactions,” in *International Conference on Financial Cryptography and Data Security*, 2015, pp. 1–18. Accessed: Apr. 05, 2017. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-662-48051-9_1

[118] P. L. and J. S. Unit Dell SecureWorks Counter Threat, “BGP Hijacking for Cryptocurrency Profit.” https://www.secureworks.com/research/bgp-hijacking-for-cryptocurrency-profit (accessed Apr. 25, 2017).
[119] M. Apostolaki, A. Zohar, and L. Vanbever, “Hijacking Bitcoin: Routing Attacks on Cryptocurrencies”, Accessed: Apr. 25, 2017. [Online]. Available: https://pdfs.semanticscholar.org/0cc7/2d3f08071fdb2c956d32e4ec98fc4250c1ff.pdf
[120] K. Baqer, D. Y. Huang, D. McCoy, and N. Weaver, “Stressing Out: Bitcoin ‘Stress Testing,’” in *Financial Cryptography and Data Security*, Feb. 2016, pp. 3–18. doi: 10.1007/978-3-662-53357-4_1.
[121] C. Decker and R. Wattenhofer, “Bitcoin transaction malleability and MtGox,” in *European Symposium on Research in Computer Security*, 2014, pp. 313–326. Accessed: Apr. 05, 2017. [Online]. Available: http://link.springer.com/10.1007/978-3-319-11212-1_18
[122] “Bitcoins the hard way: Using the raw Bitcoin protocol.” http://www.righto.com/2014/02/bitcoins-hard-way-using-raw-bitcoin.html (accessed Mar. 20, 2017).
[123] P. Pääkkönen and D. Pakkala, “Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems,” *Big Data Research*, vol. 2, no. 4, pp. 166–186, Dec. 2015, doi: 10.1016/j.bdr.2015.01.001.
[124] R. Kune, P. K. Konugurthi, A. Agarwal, R. R. Chillarige, and R. Buyya, “The anatomy of big data computing,” *Software: Practice and Experience*, vol. 46, no. 1, pp. 79–105, 2016.
[125] V. Vallois, F. Guenane, and A. Mehaoua, “Reference Architectures for Security-by-Design IoT: Comparative Study,” in *2019 Fifth Conference on Mobile and Secure Services (MobiSecServ)*, Mar. 2019, pp. 1–6. doi: 10.1109/MOBISECSERV.2019.8686650.
[126] H. P. Breivold, “A Survey and Analysis of Reference Architectures for the Internet-of-things,” p. 7, 2017.
[127] B. Di Martino, M. Rak, M. Ficco, A. Esposito, S. A. Maisto, and S. Nacchia, “Internet of things reference architectures, security and interoperability: A survey,” *Internet of Things*, vol. 1–2, pp. 99–112, Sep. 2018, doi: 10.1016/j.iot.2018.08.008.
[128] P. C. Evans and A. Gawer, “The rise of the platform enterprise: a global survey,” 2016.
[129] C. M. Felipe, D. E. Leidner, J. L. Roldán, and A. L. Leal-Rodríguez, “Impact of IS Capabilities on Firm Performance: The Roles of Organizational Agility and Industry Technology Intensity,” *Decision Sciences*, 2019.
[130] V. Sambamurthy, A. Bharadwaj, and V. Grover, “Shaping agility through digital options: Reconceptualizing the role of information technology in contemporary firms,” *MIS quarterly*, pp. 237–263, 2003.
[131] A. Basiri *et al.*, “Chaos engineering,” *IEEE Software*, vol. 33, no. 3, pp. 35–41, 2016.
[132] P. Adolphs, “RAMI 4.0 - An architectural Model for Industrie 4.0,” Jun. 2015. https://www.omg.org/news/meetings/tc/berlin-15/special-events/mfg-presentations/adolphs.pdf (accessed Sep. 03, 2018).
[133] K. Hell *et al.*, “Demands on Virtual Representation of Physical Industrie 4.0 Components,” Nov. 2016.
[134] M. Buchheit, “IIRA Reference Architecture,” p. 58.
[135] J. De Loof *et al.*, “Internet of Things–Architecture IoT-A Deliverable D1. 5–Final architectural reference model for the IoT v3. 0”.
[136] P. Fremantle, “A Reference Architecture for the Internet of Things.” WSO2, 2015. doi: 10.13140/rg.2.2.20158.89922.
[137] “The Internet of Things Reference Model - Whitepaper.” Cisco, Jun. 2014.

[138] “SiteWhere Documentation | System Architecture.” http://documentation.sitewhere.io/architecture.html (accessed Sep. 12, 2018).
[139] “Architecture Overview - AWS Connected Vehicle Solution.” https://docs.aws.amazon.com/solutions/latest/connected-vehicle-solution/architecture.html (accessed Sep. 05, 2018).
[140] “Internet of Things reference architecture - IBM Cloud Garage Method.” https://www.ibm.com/cloud/garage/architectures/iotArchitecture/reference-architecture/ (accessed Sep. 12, 2018).
[141] “Microsoft Azure IoT Reference Architecture V2.0.” 2018.
[142] “Architecture: Real-Time Stream Processing for IoT | Architectures,” *Google Cloud*. https://cloud.google.com/solutions/architecture/real-time-stream-processing-iot (accessed Sep. 05, 2018).
[143] T. L. Saaty, “Decision making with the analytic hierarchy process,” *International journal of services sciences*, vol. 1, no. 1, pp. 83–98, 2008.
[144] J. Wallenius, J. S. Dyer, P. C. Fishburn, R. E. Steuer, S. Zionts, and K. Deb, “Multiple Criteria Decision Making, Multiattribute Utility Theory: Recent Accomplishments and What Lies Ahead,” *Management Science*, vol. 54, no. 7, pp. 1336–1349, Jul. 2008, doi: 10.1287/mnsc.1070.0838.
[145] K. D. Goepel, “Online Software Tool for the Analytic Hierarchy Process – BPMSG.” https://bpmsg.com/ahp-software-2/ (accessed Oct. 29, 2018).
[146] M. Ahmad, T. Younis, M. A. Habib, R. Ashraf, and S. H. Ahmed, “A Review of Current Security Issues in Internet of Things,” in *Recent Trends and Advances in Wireless and IoT-enabled Networks*, M. A. Jan, F. Khan, and M. Alam, Eds. Cham: Springer International Publishing, 2019, pp. 11–23. doi: 10.1007/978-3-319-99966-1_2.
[147] A. Ouaddah, H. Mousannif, A. A. Elkalam, and A. A. Ouahman, “Access control in the Internet of Things: Big challenges and new opportunities,” *Computer Networks*, vol. 112, pp. 237–262, 2017.
[148] S. Göndör and A. Küpper, “The Current State of Interoperability in Decentralized Online Social Networking Services,” in *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2017, pp. 852–857.
[149] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas, “Attribute-based access control,” *Computer*, vol. 48, no. 2, pp. 85–88, 2015.
[150] V. Franqueira and R. Wieringa, “Role-based access control in retrospect,” *Computer*, vol. 45, no. 6, pp. 81–88, 2012.
[151] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, “Security Services Using Blockchains: A State of the Art Survey,” *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018, doi: 10.1109/COMST.2018.2863956.
[152] M. Khera, “Think Like a Hacker: Insights on the Latest Attack Vectors (and Security Controls) for Medical Device Applications,” *J Diabetes Sci Technol*, vol. 11, no. 2, pp. 207–212, Mar. 2017, doi: 10.1177/1932296816677576.
[153] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman, “Performance Analysis of a Hyperledger Fabric Blockchain Framework: Throughput, Latency and Scalability,” in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 536–540.
[154] M. Bettayeb, Q. Nasir, and M. A. Talib, “Firmware update attacks and security for IoT devices: Survey,” in *Proceedings of the ArabWIC 6th Annual International Conference Research Track*, 2019, pp. 1–6.
[155] K. Wüst and A. Gervais, “Ethereum Eclipse Attacks,” ETH Zurich, 2016. doi: 10.3929/ethz-a-010724205.

[156] S. Pongnumkul, C. Siripanpornchana, and S. Thajchayapong, “Performance analysis of private blockchain platforms in varying workloads,” in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, 2017, pp. 1–6.
[157] R. Yasaweerasinghelage, M. Staples, and I. Weber, “Predicting Latency of Blockchain-Based Systems Using Architectural Modelling and Simulation,” in *Software Architecture (ICSA), 2017 IEEE International Conference on*, 2017, pp. 253–256.
[158] A. M. Antonopoulos, *Mastering Bitcoin: unlocking digital cryptocurrencies*. O’Reilly Media, Inc., 2014. [Online]. Available: https://books.google.fr/books?hl=en&lr=&id=IXmrBQAAQBAJ&oi=fnd&pg=PR4&dq=mastering+bitcoin&ots=9B7WjqOlQS&sig=Pie-RXsXM3qQz50_hC5ReXkC4CQ
[159] T. Alladi, V. Chamola, R. M. Parizi, and K.-K. R. Choo, “Blockchain Applications for Industry 4.0 and Industrial IoT: A Review,” *IEEE Access*, pp. 1–1, 2019, doi: 10.1109/ACCESS.2019.2956748.
[160] M. Kamran, H. U. Khan, W. Nisar, M. Farooq, and S.-U. Rehman, “Blockchain and Internet of Things: A bibliometric study,” *Computers & Electrical Engineering*, vol. 81, p. 106525, Jan. 2020, doi: 10.1016/j.compeleceng.2019.106525.
[161] M. Nuss, A. Puchta, and M. Kunz, “Towards Blockchain-Based Identity and Access Management for Internet of Things in Enterprises,” in *Trust, Privacy and Security in Digital Business*, vol. 11033, S. Furnell, H. Mouratidis, and G. Pernul, Eds. Cham: Springer International Publishing, 2018, pp. 167–181. doi: 10.1007/978-3-319-98385-1_12.
[162] P. Wang, Y. Yue, W. Sun, and J. Liu, “An Attribute-Based Distributed Access Control for Blockchain-enabled IoT,” in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2019, pp. 1–6. doi: 10.1109/WiMOB.2019.8923232.
[163] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, Q. Zhang, and K.-K. R. Choo, “An Energy-efficient SDN Controller Architecture for IoT Networks with Blockchain-based Security,” *IEEE Transactions on Services Computing*, pp. 1–1, 2020, doi: 10.1109/TSC.2020.2966970.
[164] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, “Smart Contract-Based Access Control for the Internet of Things,” *arXiv:1802.04410 [cs]*, Feb. 2018, Accessed: Feb. 21, 2018. [Online]. Available: http://arxiv.org/abs/1802.04410
[165] S. Dhakal, F. Jaafar, and P. Zavarsky, “Private blockchain network for IoT device firmware integrity verification and update,” in *2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE)*, 2019, pp. 164–170.
[166] B. Lee and J.-H. Lee, “Blockchain-based secure firmware update for embedded devices in an Internet of Things environment,” *The Journal of Supercomputing*, vol. 73, no. 3, pp. 1152–1167, 2017.
[167] “ritchie46/lsh-rs,” *GitHub*. https://github.com/ritchie46/lsh-rs (accessed Dec. 17, 2021).
[168] A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti, “A large-scale analysis of the security of embedded firmwares,” in *23rd ${$USENIX$}$ Security Symposium (${$USENIX$}$ Security 14)*, 2014, pp. 95–110.