ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ETTECHNOLOGIES

N° d'ordre : 3319

THESE

En vue de l'obtention du : **DOCTORAT**

Structure de Recherche : Laboratoire de Recherche en Informatique et Télécommunications
Discipline : Sciences de l'Ingénieur
Spécialité : Informatique et Télécommunications

Présentée et soutenue le 23/07/2020 par :

Fatima SIFOU

Contributions to Access Control for Cloud-Enabled Internet of Things

Devant le jury :

Moulay Driss RAHMAN	**PES**	Faculté des Sciences, Université Mohammed V, Rabat	**Président**
Ahmed HAMMOUCH	**PES**	ENSET, Université Mohammed V, Rabat	**Directeur de Thèse**
Soumia ZITI	**PH**	Faculté des Sciences, Université Mohammed V, Rabat	**Rapporteur/Examinateur**
Mounir AIT KERRUM	**PH**	ENCG, Université Ibn Tofail, Kénitra	**Rapporteur/Examinateur**
Abdelali LASFAR	**PH**	EST, Université Mohammed V, Rabat	**Rapporteur/Examinateur**
Mohamed LAZAAR	**PH**	ENSIAS, Université Mohammed V, Rabat	**Examinateur**

Année Universitaire : 2019/2020

Dedication

I dedicate this work to my family and friends. A special dedication goes to my greatest mom, Fatima, and to the memory of my father, Ali, who brought me to the life, raised me, and believed in my ability to accomplish my PhD degree. I would like to give special gratitude to my siblings Ibrahim, Mohammed, Aisha, Said, Khadija, and Zahra who supported and encouraged me to successfully complete my study. I also want to thank my sister in law Zhour and my two brothers in law. A special feeling of gratitude to my lovely nieces, Youssra, Lilia, Hajar, Ghizlane, Soukaina and Amal, to my nephews Ali Sifou, Ali, Youssef and Ayoub. I would like to express sincere thanks to my friend and colleague Mbarek, to my best friends Amina, Adra and Samira.

Acknowledgments

Most of the important things in the world have been accomplished by people who have kept on trying when there seemed to be no hope at all.

Dale Carnegie

This thesis is the end of almost four years journey to obtain my Ph.D. degree in Computer Science. The research work of this Ph.D thesis has been carried out in Laboratory for Research in Computer Sciences and Telecommunications, Mohammed V University, Faculty of Science, Morocco, during the years 2017-2020.

First and foremost, I thank God the most gracious and most merciful for blessing me with guidance, strength, and patience to complete this work.

Special thanks to my thesis supervisor, Prof. Ahmed HAMMOUCH for his selfless support and encouragement I received throughout my Ph.D study. Your guidance, patience, advice and feedback led to the successful completion of this work.

I would like to thank Prof. Moulay Driss RAHMANI, Professor in the Faculty of Sciences in Rabat, for accepting to preside the jury of my thesis defense and for reporting on my thesis.

I would like to thank Prof. Mounir AIT KERROUM, Professor in the National School of Business and Management, Ibn Tofail University in Kenitra, for reporting on my thesis and for examining my thesis defense.

I would like to thank Prof. Abdelali LASFAR, Professor in the Higher School of Technology in Sale, for reporting on my thesis and for examining my thesis defense.

I would like to thank Prof. Soumia ZITI, Professor in the Faculty of Sciences in Rabat, for reporting on my thesis and for examining my thesis defense.

I would like to thank Prof. Mohamed LAZAAR, Professor in the National School of Computer Science and Systems Analysis in Rabat, for examining my thesis defense.

I am extremely grateful to Prof. Achraf Benba, Professor in the Higher Normal School for Technical Education in Rabat, for his patience for correcting this thesis and critical comments on every chapter.

I would like to express my sincere gratitude towards my family for the encouragement, which helped me in completion of my Ph.D degree. The endless forms of support I received from my father Ali (May Allah grant his soul Jannat Alfirdaouss) and my mother Fatima since my childhood are unimaginable, immeasurable, and incomprehensible.

Abstract

The Internet of Things (IoT) paradigm promises to make any things/objects part of both industries and humans every-day life. This sensor network is opening valuable opportunities for a large number of novel applications that promise to improve the quality of humans' live. The present thesis investigates the integration of cloud and IoT, called Cloud-enabled IoT to tackle the main challenges facing IoT systems, such as device's resource scarcity, energy storage capacities, memory resource and processors' computational capacities. Initially, this thesis explores the security and privacy challenges encountered in adopting cloud computing and IoT. Thus, a comprehensive taxonomy of vulnerabilities and threats in distributed systems, especially IoT and cloud, are highlighted, setting the context of the work, and investigating the state-of-the-art.

The main goal of this PhD thesis is to design and implement an efficient access control system tailored to the IoT application domain. In order to achieve this, we study and compare the most commonly used access control models in distributed environments. We shall use Attributes-Based Access Control (ABAC) as a secure, flexible and scalable security control mechanism so that it meets the required properties of cloud-enabled IoT. The second original contribution suggests OM-AM reference to design the appropriate access control for cloud-enabled IoT with respect to technical constraints. The effectiveness of the proposed solution is evaluated by means of real XACML-based policies. Our third contribution deals with policy conflict detection and resolution in ABAC. In this respect, we opt for clustering approach, especially the fuzzy c-means (FCM) algorithm, to detect and resolve anomalies. We prove the correctness of this method in terms of computational complexity.

Keywords: Internet of Things, Cloud Computing, Access Control Models, ABAC Model, XACML Language, Fuzzy C-Means Algorithm.

Résumé

L'Internet des objets (IoT) joue un rôle très important dans plusieurs domaines. Ainsi, elle est utilisée dans le domaine logistique pour la gestion des stocks et des acheminements. Aussi dans le domaine environnemental pour la surveillance des paramètres météorologiques. Sans oublier le domaine médical pour le suivi des paramètres vitaux. Ainsi, cette technologie permet d'interconnecter plusieurs objets, de collecter plusieurs types d'informations, de transmettre les données en utilisant des réseaux sans fil et de traiter certaines données pour l'aide à la prise de décision. Malgré les gains offerts par cette technologie, elle rencontre deux problèmes majeurs qui freinent son utilisation sur une échelle plus large. Le premier consiste au stockage des données volumineuses sur des objets de capacité de mémoire limitées, le second dépend de la consommation importante de l'énergie stockée sur des batteries de charge limitée.

Le Cloud Computing se présente comme la solution idéale pour externaliser le stockage et le traitement des données volumineuses. Cependant, la sécurité des données critiques et des paramètres confidentiels est citée comme l'obstacle principal dans le processus de l'intégration des technologies du Cloud Computing et de l'IoT. Ainsi, un mécanisme de contrôle d'accès fort et puissant au niveau des ressources du Cloud est nécessaire pour protéger le patrimoine informatique des entreprises des attaques malveillantes et des abus de confiance.

Dans ce contexte, nous proposons d'abord un nouveau Framework basé sur le modèle ABAC pour assurer l'accès sécurisé aux ressources déployées dans le Cloud. Ensuite, pour vérifier et contrôler les requêtes d'accès, nous utilisons un nouvel algorithme qui permet de détecter et réduire les redondances et les conflits dans une politique de sécurité. Cet algorithme est basé sur la méthode Fuzzy C-Means (FCM) pour la classification des règles afin de faciliter l'évaluation de la politique de sécurité. Enfin, nous identifions les principales voies futures de recherches sur ce sujet pour améliorer la sécurité et la qualité des services de cette nouvelle architecture.

Mots-clés: L'Internet des Objets, Informatique en Nuage, les Modèles du Contrôle d'Accès, le Modèle ABAC, Langage XACML, L'algorithme Fuzzy C-Means.

Résumé détaillé

L'Internet des Objets (IoT) se définit comme un réseau mondial de services interconnectés et d'objets intelligents de toutes natures destinés à soutenir les humains dans les activités de la vie quotidienne grâce à leurs capacités de détection, de calcul et de communication. Leurs aptitudes à observer le monde physique et à fournir des informations pour la prise de décision, seront partie intégrante de l'architecture de l'Internet du futur.

L'IoT comprend une grande diversité de dispositifs intégrants capteurs, qui observent et mesurent l'état du monde réel, et actuateurs qui agissent sur lui. Il relie la vie réelle au monde virtuel. L'IoT doit s'intégrer dans un système plus global qu'est l'écosystème digital. Cet écosystème se définit comme un ensemble formé par une communauté de services en interrelation avec son environnement. L'utilisateur est au centre de cet écosystème et doit être en mesure d'accéder à tous les services. Cependant, le nombre d'appareils connectés augmente au fur et à mesure, passant de milliards à bientôt centaines de milliards. Si l'on considère que chaque dispositif IoT comprend un ou plusieurs micro-services, le nombre croissant d'appareils présents autour de l'utilisateur, les rend difficiles à contrôler et à gérer.

L'utilisateur d'aujourd'hui bouge et change d'environnement, change d'appareil, souhaite une continuité de service sans faille et une qualité de service (QoS) de bout en bout. L'écosystème digital doit être au service de l'utilisateur. Le contrôle de la QoS est crucial pour la prise de décision. L'application IdO doit donc être contrôlée de bout en bout de manière à assurer une continuité de service.

En plus de ce qui a été mentionné ci-dessus, l'IoT, elle rencontre deux problèmes majeurs qui freinent son utilisation sur une échelle plus large. Le premier consiste au stockage des données volumineuses sur des objets de capacité de mémoire limitées, le second dépend de la consommation importante de l'énergie stockée sur des batteries de charge limitée.

Le Cloud Computing se présente comme la solution idéale pour externaliser le stockage et le traitement des données volumineuses. Cette technologie fournit la mise en disponibilité des ressources informatiques en tant que services plutôt que produits. Dans ce contexte, les ressources et les logiciels partagés sont offerts aux appareils numériques sous forme d'utilitaires à travers un réseau Internet. Les entreprises peuvent entamer leurs activités en cloud dès qu'elles décident d'utiliser ce dernier. Elles se débarrassent, de cette façon, du besoin d'installation locale d'applications dans chaque ordinateur au sein de l'organisme. Ainsi, elles sont déléguées au fournisseur des services cloud toutes les tâches laborieuses de mise à jour et de mise à niveau possibles.

Comme toutes les innovations technologiques majeures, et malgré tous ses avantages, le cloud computing présente un nouvel ensemble de défis qui doivent être résolus. L'une de ces principales préoccupations est la sécurité. Tel qu'il a été mentionné avant, le cloud computing est le point culminant de bon nombre de technologies indépendantes. Par conséquent, le cloud hérite des problèmes de sécurité qui existent dans ces technologies. En outre, de nouveaux problèmes de sécurité découlant du cloud computing, qui lui sont propres et qui n'ont jamais été observés auparavant, apparaissent. Pour nous, cela crée une étape intéressante pour concentrer nos efforts en recherche dessus, tout en nous permettant d'aborder la question d'un point de vue différent.

La sécurité des données critiques et des paramètres confidentiels est citée comme l'obstacle principal dans le processus de l'intégration des technologies du Cloud Computing et de l'IoT. Ainsi, un mécanisme de contrôle d'accès fort et puissant au niveau des ressources du Cloud est nécessaire pour protéger le patrimoine informatique des entreprises des attaques malveillantes et des abus de confiance.

Premièrement, nous avons proposé un nouveau Framework basé sur le modèle du contrôle d'accès ABAC pour assurer l'accès sécurisé aux ressources déployées dans le Cloud. Le choix de ce modèle a été fait après une étude comparative et analytique de tous les modèles du contrôle d'accès existant. Le modèle ABAC est un mécanisme de contrôle flexible et évolutif afin qu'il réponde aux exigences de l'IoT et le Cloud Computing.

Deuxièmement, nous avons utilisé le référentiel OM-AM pour la conception d'un nouveau mécanisme de contrôle d'accès adapté aux applications spécifiques dans le contexte des besoins de l'environnement CEIoT. En effet, en ce qui concerne les mécanismes d'identification, d'authentification et d'autorisation, nous avons opté pour le langage XACML pour garantir une politique de sécurité robuste sur le modèle ABAC.

Troisièmement, nous avons implémenté la politique de sécurité ABAC en utilisant le langage XACML qui permet d''exprimer les règles utilisées pour le contrôle d'accès. Il est basé sur le langage de balisage XML qui décrit la syntaxe des règles et des politiques d'accès. Dans un environnement distribué et dynamique comme le CEIoT, il est difficile d'implémenter une solution XACML vu sa complexité en terme de la création et la gestion d'une politique XACML. Et pour réduire cette complexité, nous avons proposé d'utiliser l'éditeur SPT du langage XACML qui est une interface graphique gratuite qui aide à concevoir, tester et vérifier les politiques de sécurité.

Ensuite, pour vérifier et contrôler les requêtes d'accès, nous avons utilisé un nouvel algorithme qui permet de détecter et réduire les redondances et les conflits dans une politique de sécurité. Cet algorithme est basé sur la méthode Fuzzy C-Means (FCM) pour la classification des règles afin de faciliter l'évaluation de la politique de sécurité. Enfin, nous identifions les principales voies futures de recherches sur ce sujet pour améliorer la sécurité et la qualité des services de cette nouvelle architecture.

List of Publication

Published Papers

1. **Sifou, F.**, Kartit, and A., Hammouch, A. "Different access control mechanisms for data security in cloud computing." Proceedings of the 2017 International Conference on Cloud and Big Data Computing, ICCBDC 2017, London, United Kingdom, ACM 2017, pages 40-44, September 2017.

2. **Sifou, F.**, Hammouch, A., and Kartit, A. "Ensuring security in cloud computing using access control: A survey." Proceedings of the Mediterranean Symposium on Smart City Applications, SCAMS 2017, Lecture Notes in Networks and Systems, Vol 37. Springer, March 2018.

3. Marwan, M., Alshahwan, F**., Sifou, F.**, Kartit, A., and Ouahmane, H. "Improving the security of cloud-based medical image storage." Engineering Letters, Vol. 27, pages 175-193, February 2019.

4. **Sifou, F.**, Marwan, M., and Hammouch, A. "Applying OM-AM Reference to an ABAC Model for Securing Cloud-Enabled Internet of Things." 2018 3rd International Conference on System Reliability and Safety (ICSRS), Barcelona, Spain, IEEE, pages 86-91, April 2019.

5. **Sifou, F.**, Alshahwan, F., Marwan, M., Hammoud, A., and Hammouch, A. "Implementing policy rules in attributes based access control with XACML within a cloud-enabled IoT environment." Journal of Communications, Vol. 15, No. 1, pages 107-112, 2020.

Accepted Papers

1. **Sifou, F.**, Douzi, S., Marwan, M., Hammoud, A., Ait Temghart, A., and Hammouch, A. "Efficient ABAC Rules Evaluation through Fuzzy C-Means Algorithm in Cloud-Enabled IoT." International Journal of Machine Learning and Computing, 2020.
2. Marwan, M., Ait Temghart, A., **Sifou, F.,** and AlShahwan, F. "A Cloud Solution for Securing Medical Image Storage", JOURNAL OF INFORMATION AND ORGANIZATIONAL SCIENCES, VOL. 44, NO. 1, 2020.
3. Marwan, M., **Sifou, F.,** AlShahwan, F. and Ait Temghart, A. "An Efficient Privacy Solution for Electronic Health Records in Cloud Computing." International Journal of High Performance Systems Architecture.
4. Marwan, M., Ait Temghart, A., **Sifou, F.**, and AlShahwan. "BLOCKCHAIN APPROACH FOR A SECURE CLOUD-ENABLED IOT IN SMART HEALTHCARE", Journal of Mobile Multimedia.

Table of Contents

List of Tables

List of Figures

Introduction

In the last few years, Internet of Things (IoT) has become one of the most important and rapidly growing technologies in telecommunication field. The IoT enables the interconnection between the physical objects, people, vehicles, buildings, and devices (sensors, actuators). These sensors are responsible to collect data from the surrounding environment. In this model internet is considering the main core of the communication between these things [1]. The most important value of IoT is the ability to efficiently incorporate and coordinate new devices in the network. This new paradigm has a high impact in different areas such as industry, healthcare, smart buildings, and businesses so forth. However, IoT devices face several challenges: energy consumption, security and limited processing capacity. In this respect, Cloud Computing has recently adopted to address IoT constraints.

Cloud Computing is a new paradigm that has been appeared recently as an alternative solution to outsource both storage and computing to an external provider, thereby mitigating the problems connected to traditional datacentres. Thus, Cloud Computing delivers different types of services including servers, storage, networking, and software over the internet. More importantly, the storage and data processing are carried out remotely rather than on local device. Hence, this paradigm offers many benefits for users and enterprises like reducing costs, increasing flexibility, and easy handling. In fact, users only pay for what they use. Despite these advantages, Cloud Computing still faces some issues. Security remains the major issue that needs to be addressed in the Cloud Computing environment.

Specifically, Access Control has an important role in the area of cloud security issues [2]. This mechanism is essential to limit and control user's access to the system and its resources in order to grant access only to authorized users.

Correspondingly, authentication and authorization are the key component of any Access control system. Basically, we design a security policy to allow, deny or restrict access cloud resources. The research community has proposed several access control models in both industrial and academia fields.

Motivation

Internets of Things and Cloud Computing have emerged in recent few decades as an important technology to collect and process data. This concept has attracted the attention of governments, companies, and academia. Nevertheless, connecting smart devices to Internet raises the security issues compared with traditional distributed systems. In this thesis, we propose security measures to protect confidential data, which are collected by IoT devices and stored or processed in Cloud Computing.

Problem Statement

As discussed above, the integration of IoT and Cloud Computing poses many security concerns. The literature survey shows that existing access control models do not satisfy the constraints of IoT and Cloud, especially in terms of flexibility and dynamicity. For this reason, we propose a new approach to secure Cloud-Enabled IoT (CEIoT). In this thesis, we developed an Attributes Based Access Control (ABAC) mechanism and demonstrated its applicability in a dynamic environment, CEIoT, to overcome the security risks that occur when integrating IoT onto Cloud Computing. In fact, the ABAC is an efficient mechanism to define and implement robust security policy for the case of critical assets. That is why 70% of all businesses use this access control as the dominant component to prevent unauthorized use and access to secret data [3] due to its

flexible, dynamic and heterogeneous properties. Practically, we rely on XACML standard to implement ABAC in the proposed architecture.

Summary of Contributions

This thesis aims to explore the advantages of the integration of IoT onto Cloud platform, as well as its limitations. We will propose solutions to address the security challenges in distributed systems, in particular, CEIoT. The main contributions of this research are as follows:

1. An overview of different access control models for data security in cloud computing environment. We will compare these mechanisms based on some criteria like dynamicity, flexibility, reliability and so on. This would undoubtedly help choose the appropriate model that satisfies cloud constraints.
2. In the context of CEIoT, we adopted the reference OM-AM (Objective, Model, Architecture, and Mechanism) to design an efficient access control to reinforce the security in this distributed system. Mainly, the proposal is meant to enable fine-grained and flexible solution to remote cloud resources.
3. Access control is one of the main elements to address security issues in CEIoT environment. One of the significant components of any access control model is access policies, they are used to build the criteria to permit or deny any access request. There are several access policy languages to implement policy rules in ABAC model. We will propose eXtensible Access Control Markup Language (XACML) as the most efficient and appropriate security component for the CEIoT system as it is flexible and powerful policy language.
4. Within a large distributed system as CEIoT implementing these policies have increased the number of rules conflicts. Therefore, we propose a novel approach based on a fuzzy c-means algorithm to reduce rules space and resolve conflict rules in the CEIoT environment.

Organization of the Dissertation

The rest of this thesis is organized as follows:

Chapter 1 builds preliminary concepts which form a foundation for this thesis including cloud computing and its essential characteristics, services models and deployment models, access control models.

Chapter 2 presents a reference OM-AM to an ABAC model for CEIoT system and discusses the proposed OM-AM enhancement to CEIoT.

Chapter 3 establishes an implementation of policy rules in ABAC model with XACML language within CEIoT.

Chapter 4 introduces a novel approach to reduce rules space and resolve conflict using fuzzy c-means algorithm.

Conclusion of this thesis and future work.

CHAPTER 1 : Access Control Mechanisms for Cloud Computing

''I don't need a hard disk in my computer if i can get to the server faster…carrying around these non-connected computers is byzantine by comparison.''

Steve Jobs

This chapter presents an overview of different access control models and basic concepts of Cloud Computing. Firstly, we discuss the motivation and scope behind the current chapter (section1); and secondly we establish an overview of Cloud Computing (section 2). Then, we provide taxonomy of existing access control mechanisms (section 3). Finally, a comparative study is presented in (section 4) and section 5 summarizes the chapter.

1. Motivation and Scope

Many IT companies have recently adopted an advanced emerging technology namely cloud computing providing cost-efficient computing resources with high flexibility, availability and scalability. However, security and privacy are the main issues in cloud computing implementation. For that reason, access control models are used to address security threats and vulnerabilities. In fact, there exist a variety of access control models, including DAC (Discretionary Access Control), MAC (Mandatory Access Control), RBAC (Role Base Access Control), ORBAC (Organizational Role Based Access Control) and ABAC (Attributes Based Access Control). That is why it is important to analyze and compare different existing access control models. To achieve this goal, we determine well defined criteria deemed necessary in the system based on the current cloud computing requirements. Accordingly, we choose the appropriate model for the cloud environment.

The rest of this chapter is as follows: Section 2.2 presents an overview of cloud computing including its definition, service models, deployments models, benefits and security challenges. The different access controls mechanisms as well as their flaws and virtues are discussed in the Section 2.3 and Section 2.4. A comparative study based on a set of predefined criteria is provided in Section 2.5. The Section 2.6 concludes this Section.

2. Overview of Cloud Computing

Cloud Computing is essentially an Internet-based computing model for sharing resources like infrastructures, software, applications, and business processes. The Cloud itself defines that data is saved somewhere where we can access it using a computer device and internet together rather than stored on local servers. This has led to overcoming the physical barrier present in isolated systems. Cloud computing as an emerging computing paradigm aims to share on demand storage, computation and services transparently among different users through Internet [4]. Therefore, many IT organizations adopt Cloud as an alternative solution to manage their data with good cost, efficiency, elasticity, scalability and delivers online business. It is being used by large industries like Google, IBM, Microsoft, and Amazon. However, it also poses serious limitations that include security and privacy.

Understanding the main characteristics of Cloud services, its benefits and limitations, is crucial for cloud actors to make adequate decisions and get full advantages of this technology. This is the purpose of this section that surveys the main aspects of Cloud system [5]. There are several definitions for Cloud Computing. The most popular one is provided by National Institute of Standards and Technology (NIST) : '' Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications,

and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction '' [6]. According to the NIST, the Cloud Computing paradigm is composed of five essential characteristics, three service models, and four deployment models.

2.1 Essential Characteristics

The NIST group presented some basic characteristics of Cloud Computing that distinguish it from traditional computing models as discussed below [7][8]:

On-Demand Self-Service

This characteristic enables consumer to use services and resources automatically and on demand. Since accessing on demand services, users can customize their services without requiring any human interaction. The services that are required by the users can be provided to them when needed which saves time and reduce the overall cost.

- **Broad Network Access**

The broad network access characteristic gives users the opportunity to access pervasively cloud services and resources through standard mechanisms that enable them to use heterogeneous platforms i.e they can access cloud resources from a wide range of devices such as laptops, tablets, smart phones. Therefore, users can access these services from any location any time provided that there is adequate IP networking.

- **Resource Pooling**

The resource pooling characteristic allows several users to utilize services using a multi-agent model with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. As the services are offered in a cloud environment, users are not aware of the exact location of the services and resources they are using but may be able to specify location at a

higher level of abstraction (e.g., country, state, or datacentre). For instance, a user saving data on Drop box is not aware of the location of the storage server. Examples of resources include storage, processing, memory, and network bandwidth.

- **Rapid Elasticity**

The rapid elasticity characteristic makes the Cloud Computing architecture more flexible and scalable. This elasticity allows organizations to scale their services rapidly based on their needs and increases service capacity during busy periods, and reduces capacity during customers' off-peak periods, enabling cloud consumers to minimize costs while meeting their service quality expectations.

- **Measured Service**

This characteristic automatically controls and optimizes the use of infrastructure including (storage, processing, bandwidth and active user accounts) by leveraging metering capabilities. This allows users to utilize the services when needed while reducing the cost of the service or resource.

2.2 Cloud Service Models

Cloud Computing is offered in three different service models which each satisfy a unique set of business requirement. These models are the main components of Cloud Computing. The three most commonly used service models are briefly described as follows [9][10]:

- **Infrastructure as a Service (IaaS)**

In this model, basic services such as storage, servers, virtual machine, networking equipment, datacentre and so forth, are provided over the network. The customer would typically install, control, and maintain its own operating system and software application.

- **Platform as a Service (PaaS)**

In this model Cloud providers offer infrastructure, operating systems and development tools as a service to the clients. So, consumers use these cloud resources to deploy their application using programming languages offered by the vendor. In this model, hardware, virtualization hypervisor, operating systems and application are basically managed and maintained by the cloud provider. Meanwhile, clients are only responsible for administration of their specific applications. In this regard, various industry solutions are available: Google App Engine, Force.com, Amazon Web Services Elastic Beanstalk, and the Microsoft Windows Azure platform.

- **Software as a Service (SaaS)**

The vendor provides needed applications to the consumer. Furthermore, the cloud providers typically ensure the control and maintain of hardware, software applications, and operating system. So, consumers can only manage permissions and authorizations to their sensitive data. In this respect, various industry solutions such as government Office Productivity as a Service (OPaaS), Microsoft Office 365, Google Apps, Salesforce.com and Oracle Applications Cloud [11] are available.

2.3 Cloud Deployment Models

Many IT companies use some deployment models to implement the services models discussed above. There are four commonly used deployment models. Each of these models has different characteristics and implications for the customers [12].

- **Public Cloud**

This model is the most widely used one for the Cloud Computing service. In this type services are shared among several organizations. Also, resources are owned and managed by a service provider. In addition, services are

designed to dramatically reduce IT costs. In fact, the customers are billed according to their actual use of cloud services. This model has a variety of inherent security risks that need to be considered when moving to cloud.

- **Private Cloud**

This model is used exclusively by a single organization where resources and services are only offered to their own employees or customers. In such an approach, multi-tenancy environment is used by the same organization so as to guarantee a great degree of security. However, it is considered the most expensive option of the cloud model. Consequently, cloud resources are basically owned or managed by a specific organization.

- **Community Cloud**

Generally speaking, in this model, resources and services are managed and maintained by a consortium of organization who have shared interests in contrast to public cloud in which users do not have shared common interests. It is a particular model of private cloud in which resources are devoted to specific users from two or more organizations.

- **Hybrid Cloud**

A hybrid cloud is a combination of the private and public cloud models. The primary objective of this concept is to take advantage of each model involved in this solution. The organizations can distribute part of their services on public and the other part on private cloud infrastructure.
The cloud deployment models discussed above have advantages as well as disadvantages. These pros and cons are mentioned in the Table 1 below.

Table 1: Pros and cons of each cloud deployment model

Cloud Deployment Models	Pros	Cons
Public Cloud	- Low Cost - Increased Efficiency - Ease to use	- Wrong Provider - Reduced Control - Perceived Weaker Security
Private Cloud	- Security - Performance - Control and Flexibility	- Additional Maintenance - Higher Costs
Community Cloud	- Lower Cost - Ability to share and collaborate	- Slow Adoption to date
Hybrid Cloud	- Cost Effective - Scalability and Flexibility - Balance of convenience and security	- Infrastructure Dependency

The Figure 1 below described the Cloud Computing definition framework according to NIST.

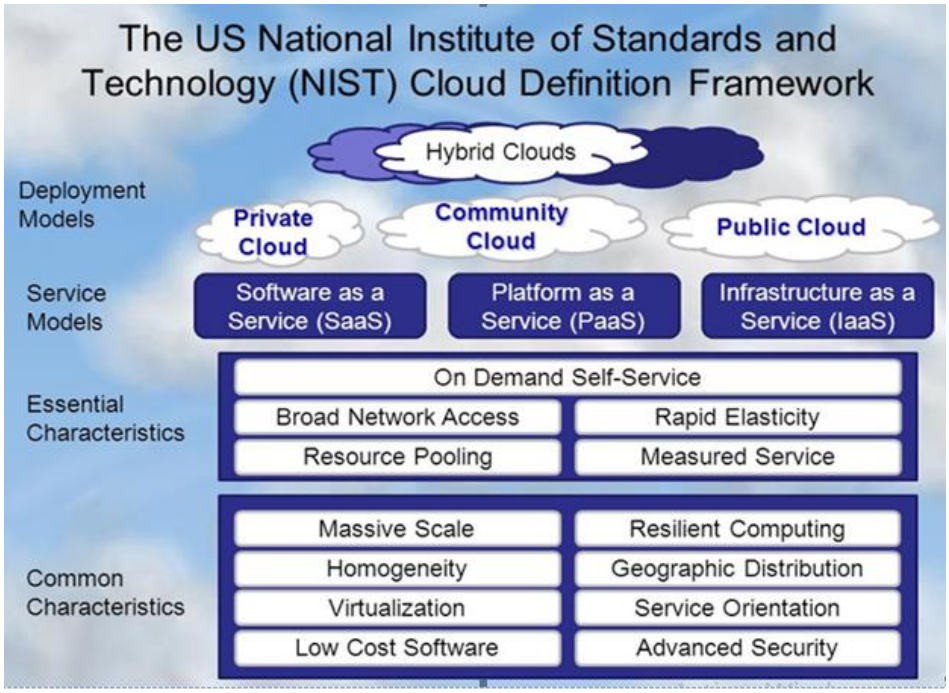


Figure 1: Cloud computing framework

2.4 Benefits of Cloud Computing

Organizations need to move their applications to cloud in order to exploit the architecture models that Cloud Computing offers. As illustrated in Figure 2, there exist a wide range of benefits of cloud adoption [13]:

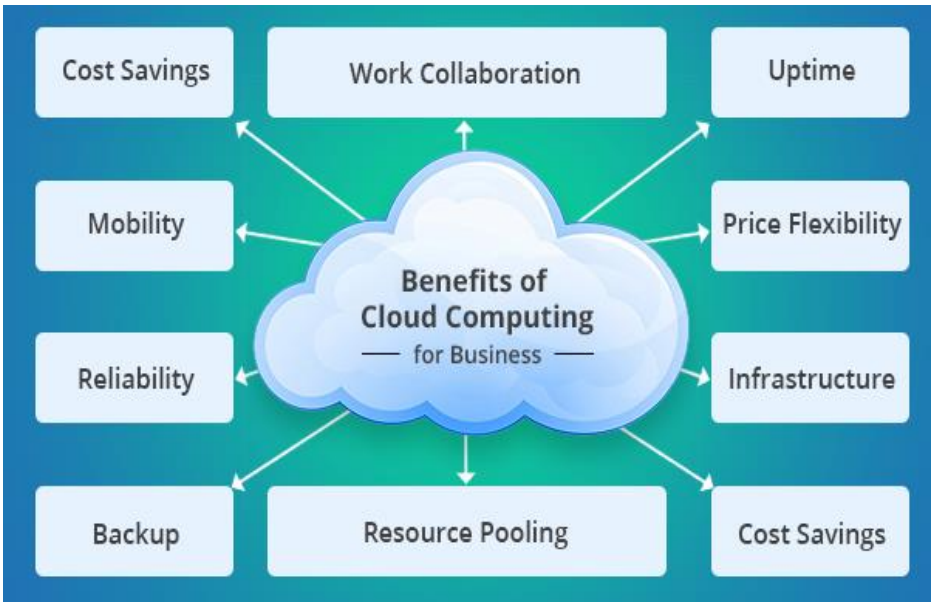


Figure 2: Major advantages of cloud services

- **Cost Savings**

Using cloud services may reduce the cost of managing and maintaining any IT system. The billing model is pay as per usage, rather than purchasing the infrastructure. Unlike traditional computing, initial expense and recurring expenses are much lower.

- **Mobility**

The large volume of data that we have today can be stored, maintained and managed remotely over a network thanks to cloud service provider. These data are accessible from anywhere in the world with an Internet connection, even if something happens to your work computer.

- **Flexibility**

This is an extremely important characteristic. With enterprises having to adapt, even more rapidly, to changing business conditions, speed to deliver is critical. Cloud computing stresses on getting applications to market very quickly, by using the most appropriate building blocks necessary for deployment.

- **Work Collaboration**

Cloud applications enhance collaboration by authorizing diverse groups of people virtually meet and exchange information with the help of shared storage. Such capability helps in improving customer service and product development and also reducing the marketing.

- **Reliability**

It means that users are able to completely forget about the other issues of their data being secure and available in the cloud without them having to be at risk of personal loss.

- **Resource Pooling**

The sharing of computing capabilities, leads to increased resource utilization rates. This means you need fewer resources and thus save costs. This animation shows how that works (using virtualization).

- **Backup**

Since all the data is stored in the cloud, backing it up and restoring the same is relatively much easier than storing the same on a physical device. Furthermore, most cloud service providers are usually competent enough to handle recovery of information. Hence, this makes the entire process of backup and recovery much simpler than other traditional methods of data storage [14].

2.5 Security Challenges in Cloud

Notwithstanding the above mentioned benefits, the migration to Cloud Computing brings several challenges especially security concerns. This is mainly due to two factors. Firstly, clients do not have full control over their data since they are generally stored on remote servers. Secondly, Cloud Computing is built upon different technologies that have a variety of inherent security risks. In this context, Figure 3 summarizes the essential root cause of security problems in cloud environment [15].

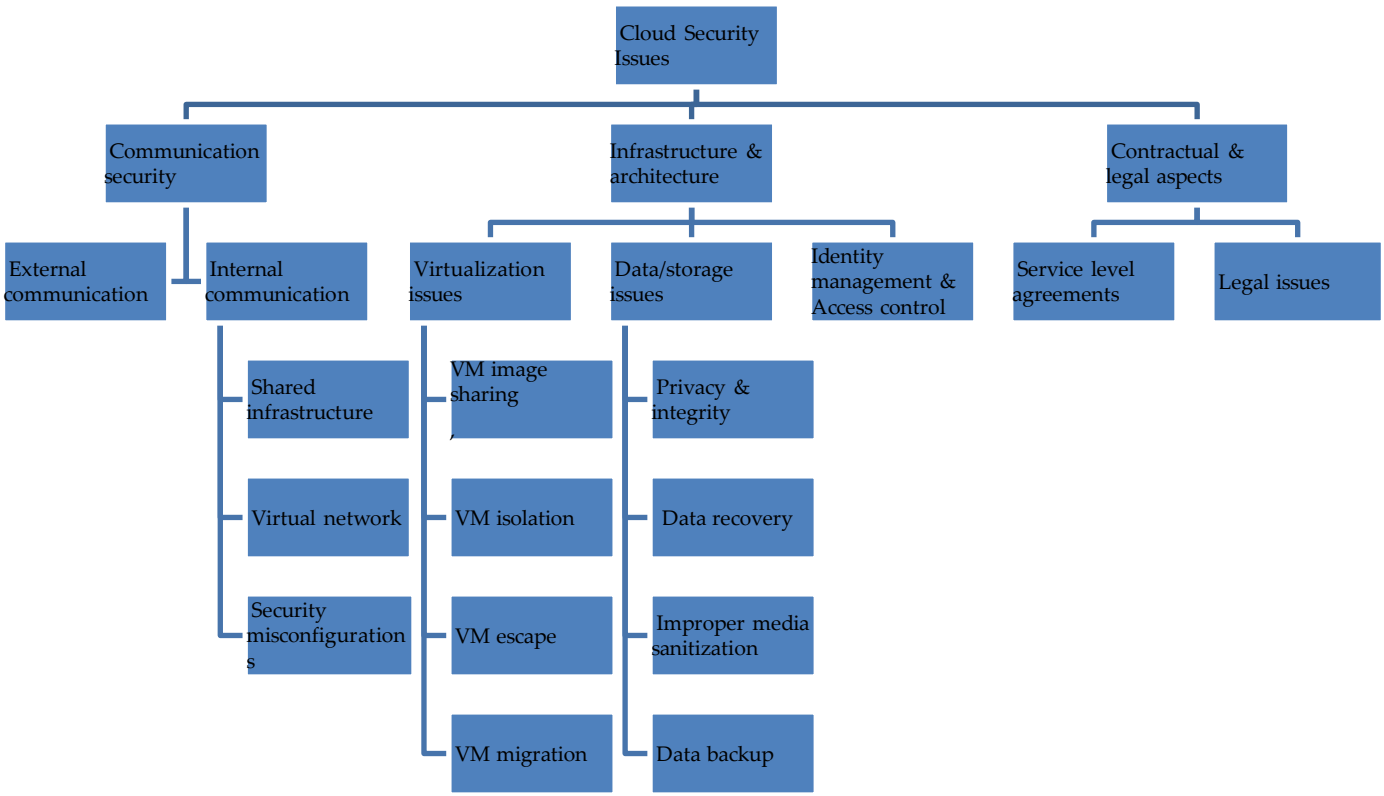


Figure 3: Cloud security issues

a) Communication Security

The security challenges faced by the cloud in case of external communication are similar to security issue of any conventional IT communication. In this context, there are many challenges including denial-of-service, man-in-the-middle, eavesdropping, IP-spoofing based flooding, and masquerading. The solutions to these challenges are classical like Secure Socket Layer (SSL), cryptographic algorithms, intrusion detection and prevention systems and so on. However, internal communication challenges are due to virtual network, shared communication infrastructure and security misconfigurations [16].

b) Infrastructure and architecture

- **Virtual Machine**

Virtualization is a new technology that allows the use of same physical resources by different clients in order to reduce cost and to improve performance and scalability. With this technique, multiple virtual machines can be mapped to the same physical resources to enable resource pooling in a multi-tenant environment. Despite its remarkable potential to provide cost effective, virtualization brings additional security concerns and threats related to virtualization technology [17]: VM image sharing, VM isolation, VM escape, Hypervisor issues and VM migration.

- **Data and Storage**

As outlined above, cloud computing architecture is a multi-domain environment in which each domain can use different security policies, privacy and trust requirements. Also, business organization is physically separated from their data hosted in the cloud provider at an un-known location. Consequently, they have less control over them. Indeed, ensuring confidentiality, integrity and availability of data in the cloud computing

environment is a complicated task due to various issues [18]: Data recovery vulnerability, improper media sanitization and Data backup.

- **Web API**

Cloud based on software as a service delivery model relies on web service application to meet customer's needs. In fact, most of cloud computing services are accessed via web-based user agents. Also, web applications are the important feature and characteristic of the cloud computing, which allow ubiquitous access to remote services. Beside its advantages, web application and Application Programming Interface (API) suffer from costly breaches due to risks outlined by Open Web Application Security Project (0WSSAP). It can be summarized as follows [19]:

- ✓ Injection (SQL, OS and LDAP)
- ✓ Broken authentication and session management
- ✓ Cross-Site Scripting (XSS)
- ✓ Insecure direct object references
- ✓ Security misconfiguration and sensitive data exposure
- ✓ Missing function level access control
- ✓ Cross-Site Request Forgery (CSRF)
- ✓ Using known vulnerable components
- ✓ Invalidated redirects and forwards

- **Access Control**

In cloud model, a third party delivers applications at any time and from any location through the Internet. Hence, access control systems are necessary to deny, restrict or allow client access to cloud applications. In an access control

decision, various attributes come into play, especially subjects, objects, action and condition [20]. In the same line, it is necessary to monitor all login attempts to enforce security measures during cloud utilization. To accomplish this, there exist several multiple access control schemes [21].

c) Contractual and Legal Aspects

In addition to technical issues, cloud platforms are spread across servers located in different countries around the world, which brings compliance and legal problems. Furthermore, current acts do not yet cover these issues related to this new paradigm. The majority of cloud computing relies on Service Level Agreements (SLA) to clearly set expectations for service between the consumer and the cloud provider. However, to monitor and evaluate this agreement is still facing complications such as billing system and quality of service (QoS). Also, to implement cloud computing brings organizational change issues related to new procedures, policies, and workflow [22].

2.6 Security and Privacy Requirements

In such a sharing environment as public cloud, the protection of business data requires strong security measures. Since generated data are often an efficient decision means to support and improve service quality, they should remain unchanged and safe. In this respect, we deeply analyse different parameters involved in data protection. Accordingly, we provide a comprehensive taxonomy based on a broad range of existing literature sources [23]. The main basics are shown in Figure 4.

Figure 4: Basic security and privacy requirements

a) Confidentiality

In practice, data storage systems provide tremendous benefits to business professionals. However, strong security measures are required in order to prevent the disclosure of sensitive data when using these applications to process business data. Outsourcing the storage and data processing are yet another benefit being offered by the cloud computing because it does not require the clients to deploy local solutions and applications. This implies, however, that consumers need to protect their digital data against untrusted cloud providers and other malicious users. For the aforementioned reasons, it is mandatory to take preventive actions for safeguarding data confidentiality before using cloud services [24].

b) Integrity

Data Integrity protects the useful information from tempering by the cybercriminals, during the communication not only between cloud storage

and organization storage but also between different cloud storage services. Data integrity is essential to make correct decisions. In this regard, it is of the utmost importance to guarantee the accuracy and consistency of the sensitive data during transmission, storage and processing. In order to properly ensure security, both encryption techniques and data processing operations must be performed without affecting the quality of business information. Obviously, methods used for integrity protection need to be both lossless and reversible. Moreover, integrity checking in cloud becomes more confusing as data are saved on remote servers.

c) Availability

It has been universally recognized that cloud computing is an efficient way to guarantee timely availability of required business applications. In this case, security measures are designed to ensure a balance between security and usability. Additionally, it is important to guarantee that clients are able to access their data in both normal and disastrous situations. Unfortunately, a wide range of factors can influence the availability of cloud services such as Denial of Service (DoS), Distributed Denial of Service (DDoS), network failure, etc.

d) Data ownership

Data ownership is a legally complex issue. It is the process that typically establishes a connection between data stored in cloud and their owner. In this case, the ownerships should have legal rights, complete control and unrestricted access to their data. Hence, the stored data is often characterized as the owner property. Watermarking methods are used to determine the legitimate owner of digital data [25].

e) Authentication

Since a public cloud is meant to serve a multitude of clients and organizations, it is necessary to implement authentication mechanisms in order to identify the end-user of cloud services. In such a scenario, the authentication server is used to check and validate the users' identity through various authentication mechanisms, including digital signatures, biometric authentication systems, Zero-Knowledge Authentication Protocol (ZKAP), Single Sign-On (SSO), etc.

f) Anonymization

Adopting Cloud Computing in the industry domain has increased the security and privacy issues. Indeed, unauthorized users should not reveal any data stored in cloud. Cloud services are required to not disclose the identity of their users. This is achieved by using data anonymization techniques to comply with data protection laws. In particular, the identity of an end-user is considered as sensitive data that should be confidential and secret. Among several existing solutions, pseudonym-based secure authentication and K-anonymity algorithms are used to maintain data privacy when using cloud services [26].

g) Auditability

In addition to the aforementioned security mechanisms, auditing is another prominent concept that provides proof for cloud resource usage in the platform. More precisely, it is necessary to formally keep a log for all actions and activities executed by cloud customers. For instance, giving details of who has accessed each resource and what actions are performed during a given period of time. The auditing process helps to ensure that users are accountable and also helps to detect unauthorized attempts to access the resources [27].

h) Unlinkability

Unlinkability is achieved if the user can make multiple actions without linked them together unless necessary. However security techniques protect data, they cannot prevent an attacker from establishing a relation between different sensitive data and events. For example, a malicious user relies greatly on events occurring during data process to get useful information in a particular context. In this respect, it is necessary to minimise risks associated with the misuse of the privacy-relevant data. Besides, we need to prohibit or restrict profiling spanning across contexts and potentially violating the purpose limitations related to the data. To deal with this problem, a distributed privacy-preserving data aggregation protocol has become largely used to avoid link ability of two or more items of interest [28].

As mentioned above, Access Control is still a big issue in cloud environment. In fact, cloud is a distributed and dynamic system that requires a sophisticated mechanism to limit unauthorized access to confidential data. Moreover, classical Access Control models are not designed to meet cloud constraints such as flexibility, scalability, fine grained access, etc. For this reason, this chapter aims to explore different existing models in order to choose the most appropriate one for Cloud Computing.

3. Access Control Models

The primary objective of any Access control is to prevent unauthorized use of confidential data. Currently, there exist various models to validate clients requests, as well as making access decision [29][30] [31].

3.1 Mandatory Access Control

This model is mainly based on the security level like Top Secret (TS), Secret(S), Confidential(C) and Unclassified (U). So, each object and subject present in cloud environment assigned some security level. The latter helps to identify the current access state of the object. In this model, the system administrator is only responsible for managing and defining an access control policy that cannot be modified by the subjects. It is the most model used in a system as far as confidentiality is concerned. The concept of this model is illustrated on Figure 5 below.

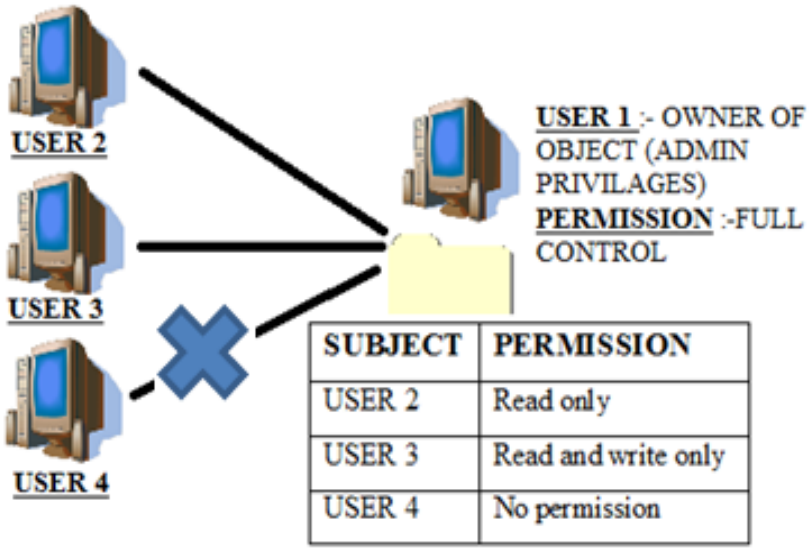


Figure 5: Mandatory access control model

3.2 Discretionary Access Control

The Owner of objects has the authority for controlling access to their objects based on user entity. Furthermore, users can create the permissions to access their data and share them with other subjects. It is the most used model in a commercial operating system such Unix and Windows-based platforms

thanks to its flexibility compared to other models [32]. The model is illustrated in Figure 6 below.

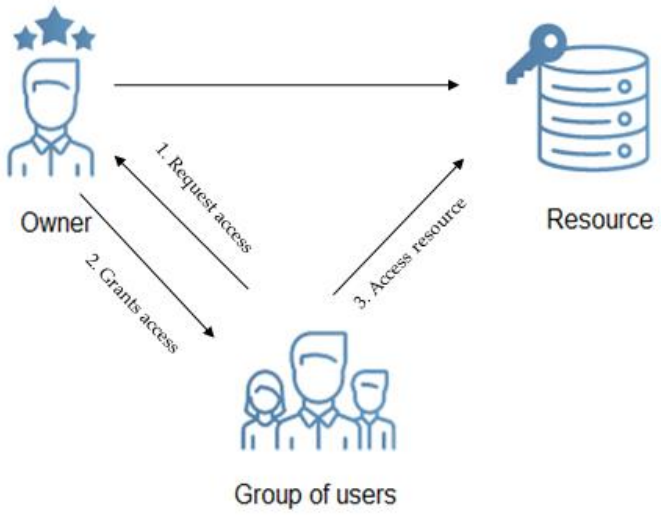


Figure 6: Discretionary access control model

3.3 Role-Based Access Control

RBAC is the prominent mechanism for controlling access within the organization. In this model rights of access are given to roles which are assigned to users. Each user can have more than one role. The latter can be a set of objects and actions associated with the user. RBAC has several administrative policies centralized, hierarchical, co-operative, ownership or decentralized [33]. The model is presented in Figure 7 below.

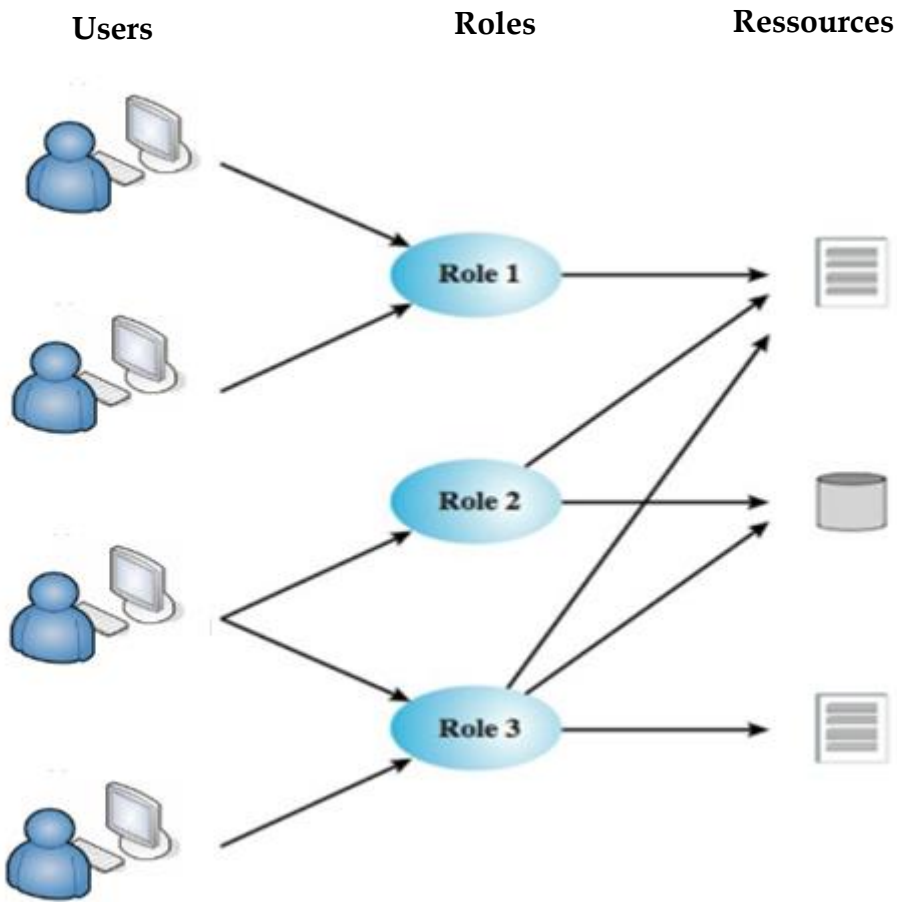


Figure 7: Role-based access control model

3.4 Organization Based-Access Control

In this model, access to cloud resources depends mostly on user's roles within an organization. Following this, various parameters have an impact on making decision, parameters such as subject, action, object, view, and context. This solution seeks to ensure fine-gained access control. For example, in an emergency case, an exceptional role can be affected to a doctor to deal with this situation [34].

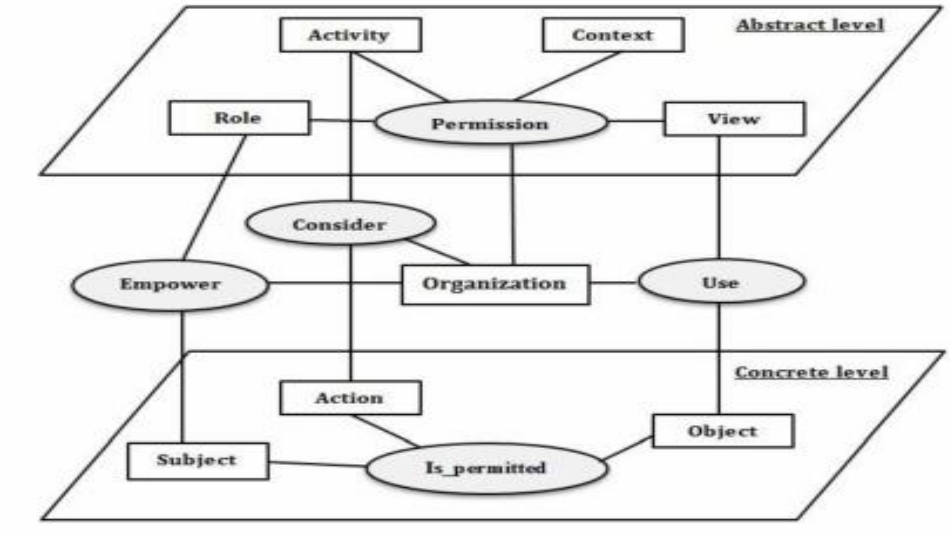


Figure 8: Organizational-based access control model

3.5 Attributes Based-Access Control

ABAC is a logical access control model based on the subject, object, operations and environment attributes to control access to resources by evaluating these attributes against rules. It supports both mandatory and discretionary access control needs. This concept solves the problem of assigning privileges. Figure 9 illustrated the basic core capabilities of basic ABAC solutions to evaluate attributes and enforce rules or a relationship between those attributes [35][36].

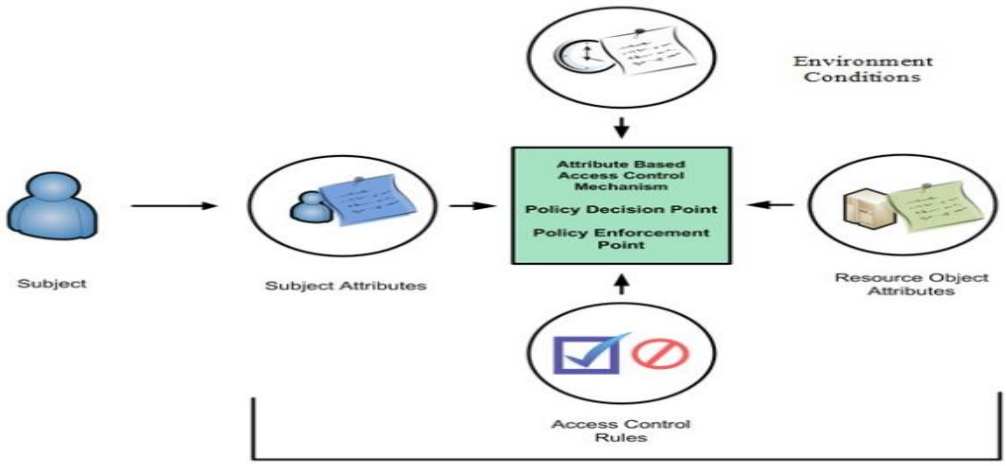


Figure 9: Attributes-based access control model

Table 2: Advantages and disadvantages of various access control models

Models	Advantages	Disadvantages
MAC	- Provides strong security - Centralized security policies	- Needs a high system management - It is costly to Implement - Vulnerable to attacks
DAC	- Makes access control more flexible - Used in environment that does not require a high level of protection	- Passing privileges between users - Attacked easily by Trojans - Security policies can be change by inserting malicious program
RBAC	- The security policy is very simple to manage - The assignment of the roles based on the least privilege minimizes the damage of information by intruders	- It is difficult to implement it in a dynamic and distributed environment - Impossible to change access rights of the user without changing the roles of that use
ORBAC	- Requires more scalability - It easily manages changes in organization	- It has high complexity - It has high computational costs
ABAC	- More flexible in a dynamic and distributed environment - Simpler to implement - It supports for the global agreement	- It requires a long running time - Difficult to manage

4. Comparative Study

This section aims at comparing different existing access control models. To this end, some criteria should be defined [37].

1.1 The Selection Criteria

In this study we propose various criteria to identify the weaknesses and strengths of existing access control models. Next we choose the appropriate

model for Cloud Computing environment.

- **Dynamicity**

The overall goal of security policy in cloud computing is to create an environment that dynamic, sharable, flexible, and simple.

- **Flexibility**

Access control model needs to be more flexible to support changes and synchronizes with the access control decisions.

- **Reliability**

To deal with security issues confronted in open and distributed system, access control models must have a high level of security.

- **Global Management**

The permission is dynamically assigned in an open and distributed environment. So, we need the access control models that support global agreement.

- **Security Policy Implementation**

The performance of access control models has become complex in a large and open system due to the sharing of resources this can consequently yield some problems related to integrity or loss of information.

- **Computational costs**

To implement access control mechanism in a large and distributed system storage space, flexibility and scalability are required. This involves a very high cost.

- **Easy in Administration**

One of the problems in a distributed system is how to handle access to resources; in this environment managing access is more complex.

Furthermore, it is difficult to protect centralized data.

- **Support Scalability**

In Cloud Computing environment which serves a large number of users, access control models must be more scalable to adapt with several organizations and support global agreement.

- **Fine-grained Access**

Access control system grants to user a set of permissions So as to allow flexibility in accessing their records. In addition, it is a security requirement to protect sensitive information from unauthorized access.

1.2 Findings

The main contribution of this study is to compare different models according to predefined criteria, as illustrated in Table 3 bellow. For this reason, various criteria are defined to choose the appropriate access control model for cloud.

Table 3: Comparison of various access control models

Models / Criteria	MAC	DAC	RBAC	OrBAC	ABAC
Dynamicity	-	-	-	+	++
Flexibility	-	-	-	+	+++
Reliability	+	++	+	+	++
Easy in administration	+	+	+	-	--
Security policy implementation	+	+	++	--	--
Global management	-	-	-	+	++
Support scalability	-	-	-	+	++
Computational costs	-	-	+	-	--
Fine-grained access	+	+	+	++	++

In consequence, we opt for the ABAC model to ensure data security in the cloud environment. In fact, this mechanism is more secure, flexible and scalable than other models. In addition it provides a hierarchical structure to facilitate access control.

5. Summary

This chapter presented and evaluated the state-of-art access control models proposed for Cloud Computing. Access control is a key component to protect data that is permanently stored, and rely on manually configured access control policies. Various access control solutions are proposed in literature to reduce the security and privacy concerns for unauthorized access to data generated, shared and processed in cloud. Based on our analysis and evaluation of these models, ABAC has proven to be an appropriate mechanism for large-scale and open distributed environments. We have adopted this security mechanism because of its flexibility and scalability within a dynamic and open environment such as the cloud.

CHAPTER 2 : A Proposed Access Control Model in Cloud-Enabled Internet of Things

"Digital technology, pervasively, is getting embedded in every place: everything, every person, every walk of life is being fundamentally shaped by digital technology – it is happening in our homes, our work, our places of entertainment. It's amazing to think of a world as a computer. I think that's the right metaphor for us as we go forward."

Satya Nadella

As discussed in the previous Chapter, Cloud Computing has several security risks which lead to the integration of Cloud Computing and Internet of Things (IoT) to deal with these security challenges. The current chapter presents a proposed access control model in Cloud-Enabled Internet of Things (CEIoT). Firstly, we discuss the motivation and scope behind the current chapter (section 1); and secondly we will introduce a background and basic concepts of the Internet of Things (section 2). Then, we propose a solution for the security risks in CEIoT (section 3). Finally, a summary for this chapter is presented in (section 4).

1. Motivation and Scope

Security and privacy are the main challenges in the internet of things (IoT) environment. When devices are interconnected and connect to the internet to form a CEIoT, unauthorized alteration, access to confidential data, or even denial of service frequently occur. To address those vulnerabilities, it is necessary to implement an access control model. Previous research shows that OM-AM is appropriate for architecture handling security problems in a dynamic environment such as IoT. This framework comprises four layers, i.e. objective, model, architecture and mechanism. Our contribution consists in applying OM-AM to an Attribute- based access control (ABAC) model to

protect IoT device 'data in cloud resources.

The rest of the chapter is organized as follows. Section 3.2 introduces a background and basic concepts of the Internet of Things. Section 3.3 discusses the fundamental of our proposed model. Finally, the Section 3.4 concludes this chapter.

2. Background and Basic Concepts

This section is an overview of the IoT paradigm, includes definition of IoT, application domains, IoT technologies, IoT architecture and IoT characteristics.

2.1 Definition

The term "Internet of Things (IoT)" was used fisrt time in 1999 by Kevin Ashton the pioneer of British technology. He used the term to explain a system in which the Internet and the real world unite with the help of a ubiquitous network of data sensors. Of course, the use of this term has grown somewhat beyond the original intention, and today it means many things to many people. But to get back to the root of it all, we should also consider the Internet itself to understand the full context of the IoT. In general, ATM's might be treated as the first IoT items which gone online as back as 1974 [38]. The Internet of Things (IoT) refers to the use of intelligently connected devices and systems to leverage data gathered by embedded sensors and actuators in machines and other physical objects. IoT has spread rapidly in recent years and this convergence will unleash a new dimension of services that improve the quality of life of consumers and productivity of entreprises. For consumers, the IoT has the potential to deliver solutions that dramatically

improve energy efficiency, security, health, education and many other aspects of daily life. For enterprises, IoT can be pillar solutions that improve decision-making and productivity in various application domains which will be discussed in details in this section. Figure 10 illustrated the IoT paradigm.

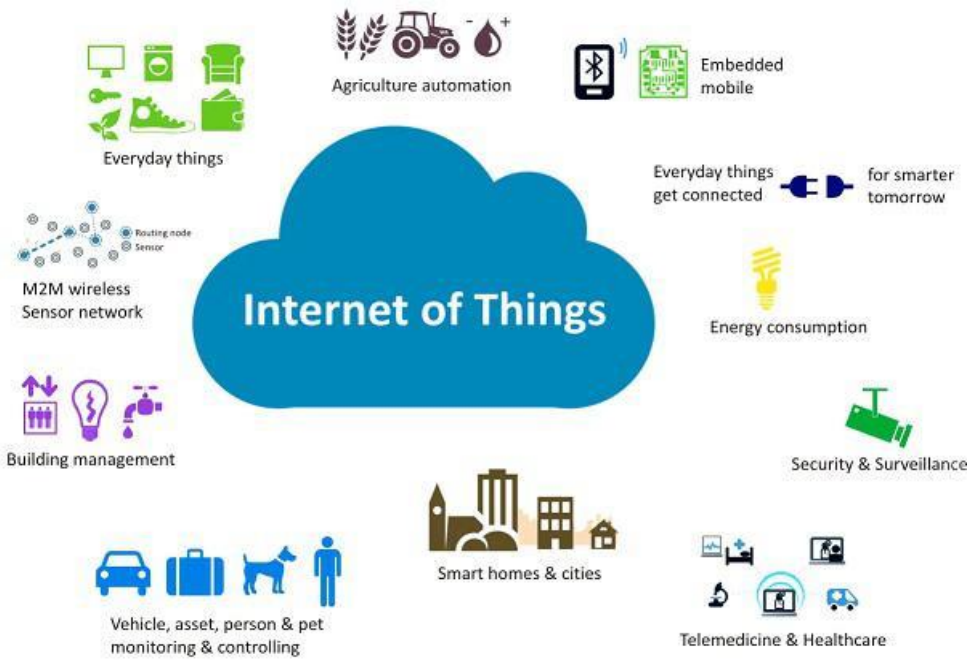


Figure 10: Internet of Things (IoT)

2.2 Application Domains

Potential applications of the IoT are numerous and diverse, permeating into practically all areas of every-day life of individuals, enterprises, and society as a whole. Below are some of the real world IoT applications.

- **Smart Homes**

Smart Homes is the most trending feature of IoT. People want their homes to be converted to smart homes in order to lead a more comfortable and convenient life [16]. For instance, refrigerators with LCD screen telling what's inside, food that's about to expire, ingredients you need to buy and with all

the information available on a Smartphone app. Washing machines allowing you to monitor the laundry remotely, and. Kitchen ranges with interface to a Smartphone app allowing remotely adjustable temperature control and monitoring the oven's self-cleaning feature[39]

- **Smart Cities**

From smart homes the IoT applications now extend to smart cities. This concept is used to describe the better use of public resources. In this context, the IoT provides several benefits in the management and optimization of public services, such as transport and parking, lighting, surveillance and maintenance of public areas, preservation of cultural heritage, and garbage collection. Furthermore, the availability of different types of data collected by IoT devices can be used to enhance the awareness of people about the status of their city and stimulate the active participation of the citizens in the management of public administration [40].

- **Wearable Gadgets**

There is heavy demand of wearable IoT devices in the market. These wearable IoT devices have sensors and software installed in them that collect valuable information about the user and processing generates useful insights for the user. These devices are mainly for health, fitness and entertainment purposes. The main advantages of these gadgets are small size, highly efficient and low power.

- **Smart Agriculture**

Green Houses: Control micro-climate conditions to maximize the production of fruits and vegetables and its quality, **Compost:** Control of humidity and temperature levels in alfalfa, hay, straw, etc. to prevent fungus and other microbial contaminants, **Animal Farming/Tracking:** Location and identification of animals grazing in open pastures or location in big stables,

Study of ventilation and air quality in farms and detection of harmful gases from excrements, **Offspring Care:** Control of growing conditions of the offspring in animal farms to ensure its survival and health, **field Monitoring:** Reducing spoilage and crop waste with better monitoring, accurate on-going data obtaining, and management of the agriculture fields, including better control of fertilizing, electricity and watering [41].

- **Healthcare**

Patients Surveillance: Monitoring of conditions of patients inside hospitals and in old people's home, **Medical Fridges:** Control of conditions inside freezers storing vaccines, medicines and organic elements, **Fall Detection:** Assistance for elderly or disabled people living independent, **Dental:** Bluetooth connected toothbrush with Smartphone app analyses the brushing uses and gives information on the brushing habits on the Smartphone for private information or for showing statistics to the dentist, **Physical Activity Monitoring:** Wireless sensors placed across the mattress sensing small motions, like breathing and heart rate and large motions caused by tossing and turning during sleep, providing data available through an app on the Smartphone [41].

- **Smart Industry**

Explosive and Hazardous Gases: Detection of gas levels and leakages in industrial environments, surroundings of chemical factories and inside mines, Monitoring of toxic gas and oxygen levels inside chemical plants to ensure workers and goods safety, Monitoring of water, oil and gas levels in storage tanks and Cisterns, **Maintenance and repair:** Early predictions on equipment malfunctions and service maintenance can be automatically scheduled ahead of an actual part failure by installing sensors inside equipment to monitor and send reports.

- **Smart Environment**

Air Pollution monitoring: Control of CO2 emissions of factories, pollution emitted by cars and toxic gases generated in farms, **Forest Fire Detection:** Monitoring of combustion gases and pre-emptive fire conditions to define alert zones, **Weather monitoring:** weather conditions monitoring such as humidity, temperature, pressure, wind speed and rain, Earthquake Early Detection, **Water Quality:** Study of water suitability in rivers and the sea for eligibility in drinkable use**, River Floods:** Monitoring of water level variations in rivers, dams and reservoirs during rainy days**, Protecting wildlife:** Tracking collars utilizing GPS/GSM modules to locate and track wild animals and communicate their coordinates via SMS.

- **Smart Energy**

Smart Grid: Energy consumption monitoring and management, **Wind Turbines/ Power house:** Monitoring and analysing the flow of energy from wind turbines & power house, and two-way communication with consumers' smart meters to analyse consumption patterns, **Power Supply Controllers:** Controller for AC-DC power supplies that determines required energy, and improve energy efficiency with less energy waste for power supplies related to computers, telecommunications, and consumer electronics applications, **Photovoltaic Installations**: Monitoring and optimization of performance in solar energy plants[41].

- **Smart Vehicles**

Connected cars are equipped with Internet access and can share their access with others, just like connecting to a wireless network in a home or office. More vehicles are starting to come equipped with this functionality, so be prepared to see more apps included in future cars. The connected car is considered as the best way to minimize accidents such that a pilot can operate

the car remotely to minimize car accidents and reduce human errors. These driverless cars can provide functions more than just safety such as they can save valuable time, and reduce the stress of driving etc. Some studies reveal that by 2040, driverless cars will account for up to 75 % of cars on the road worldwide.

The IoT application area is very diverse and it serves different users. These different user categories have different driving needs. The most popular IoT application domains are depicted in Figure 11 below:

Figure 11: Application domains of IoT

2.3 IoT Technologies and Protocols

In IoT applications, it is necessary to convey the data collected by the devices or sources to the internet. Proving connectivity and coverage is a daunting task for IoT applications. The users or vendors always wanted to collect data

and analyse it for further processing for better enhancement of their devices. It is mandatory to properly advent new technologies for communicating and processing the data. Rather than going to other network, a better privileged network especially for its own applications would be a good choice. Nowadays, the industries are actively involved developing wired or wireless communication channels or protocols. But the cost and infrastructure development plays a vital role in developing technology for IoT. The technology embedded in IoT is shown in Figure 12. A glimpse of each of these items is discussed next [42].

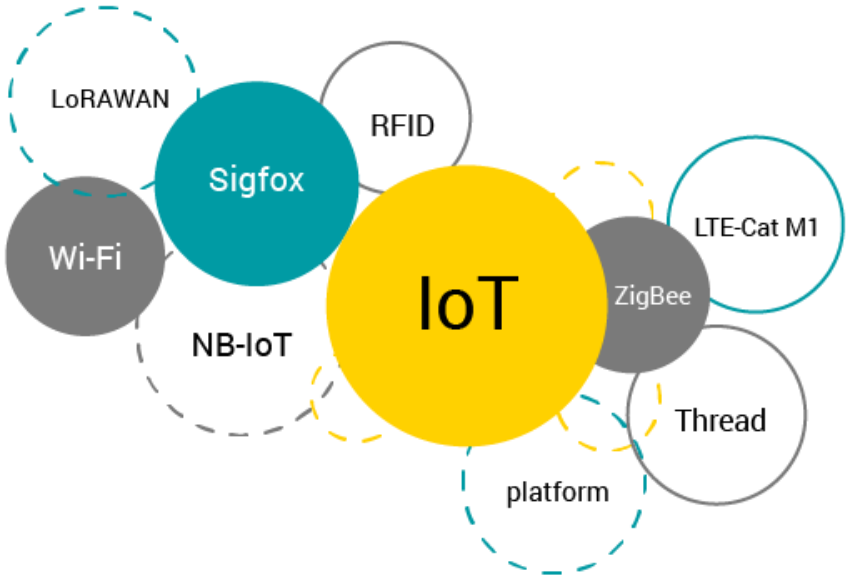


Figure 12: IoT technologies

2.4 IoT Technology

The IoT vision can be supported by a variety of exciting technologies for different kinds of applications. This section is dedicated to presenting and compiling the most appropriate IoT technology. In the following, the compiled technologies are presented [43]:

- **RFID (Radio frequency identification)**

RFID is an automatic technology and aids machines or computers to identify objects, record metadata or control individual target through radio waves [44]. It uses tags (transmitters/responders) and readers (transmitters/receivers). The tag is a microchip connected with an antenna, which can be attached to an object as the identifier of the object. The RFID reader communicates with the RFID tag using radio waves.

- **Bluetooth**

As a well-established short-range connectivity technology, Bluetooth is considered to be the key solution particularly for the future of the wearable electronics market such as wireless headphones or geolocation sensors, especially given its widespread integration with smartphones. Designed with cost-effectiveness and reduced power consumption in mind, the Bluetooth Low-Energy (BLE) protocol requires very little power from the device. Yet, this comes with a compromise: when transferring frequently higher amounts of data, BLE may not be the most effective solution [45].

- **Wi-Fi**

Wi-Fi connectivity is one of the most popular IoT communication protocols, often an obvious choice for many developers, especially given the availability of Wi-Fi within the home environment within LANs. There is a wide existing infrastructure as well as offering fast data transfer and the ability to handle high quantities of data. Currently, the most common Wi-Fi standard used in homes and many businesses is 802.11n, which offers range of hundreds of megabit per second, which is fine for file transfers but may be too power-consuming for many IoT applications [46].

- **ZigBee**

ZigBee is similar to Bluetooth and is majorly used in industrial settings. It has

some significant advantages in complex systems offering low-power operation, high security, robustness and high and is well positioned to take advantage of wireless control and sensor networks in IoT applications. The latest version of ZigBee is the recently launched 3.0, which is essentially the unification of the various ZigBee wireless standards into a single standard [47][48].

- **Z-Wave**

It is a low energy Radio Frequency (RF) technology for sub-GHz communications. It is a mesh networking protocol, ofen adopted for home automation, security systems, and lighting controls. Z-Wave employs a simpler protocol than some other alternatives, which allow faster and simpler development. It also supports full mesh networking without requiring a coordinator node and is highly scalable. It operates on 900MHz with 9.6/40/100 kbit/s data rates [49].

- **Thread**

Designed specifically for smart home products, Thread employs IPv6 connectivity to enable connected devices to communicate between one another, access services in the cloud, or interact with the user via Thread mobile applications. The critics of Thread have pointed out that given the saturation of the market, yet another wireless communication protocol leads to further fragmentation within the IoT technology stack [50].

- **SigFox**

The concept behind Sigfox is to provide an effective connectivity solution for low-power M2M applications requiring low levels of data transfer for which the WiFi range is too short, and cellular range is too expensive and too power-hungry. Sigfox employs UNB, a technology that enables it to handle low data-

transfer speeds of 10 to 1,000 bits per second. Consuming up to 100 times less energy compared to cellular communication solutions, it delivers a typical stand-by time of 20 years for a 2.5Ah battery. Offering a robust, energy-efficient and scalable network able to support communication between thousands of thousands of battery-operated devices across areas of several square kilometres, Sigfox proves suitable for various M2M applications, including smart street lighting, intelligent meters, patient monitors, security devices, and environmental sensors. Sigfox is currently employed in a growing number of IoT technology solutions such as AVSystem's Coiote IoT Data Orchestration, to name only one [50].

- **LoRaWAN**

LoRaWAN is one of popular IoT Technology, targets wide-area network (WAN) applications. The LoRaWAN design to provide low-power WANs with features specifically needed to support low-cost mobile secure communication in IoT, smart city, and industrial applications. Specifically meets requirements for low-power consumption and supports large networks with millions and millions of devices, data rates range from 0.3 kbps to 50 kbps.

2.5 IoT Protocols

IoT is a very large scale network consisting of heterogeneous constrained devices and smart objects. Such constrained network imposes a significant impact on designing of different protocols. These protocols are supposed to offer efficient and scalable communications and allow developing and deploying applications/services adopted for a variety of environments. Some of these protocols are shown in the Figure 13 and discussed below.

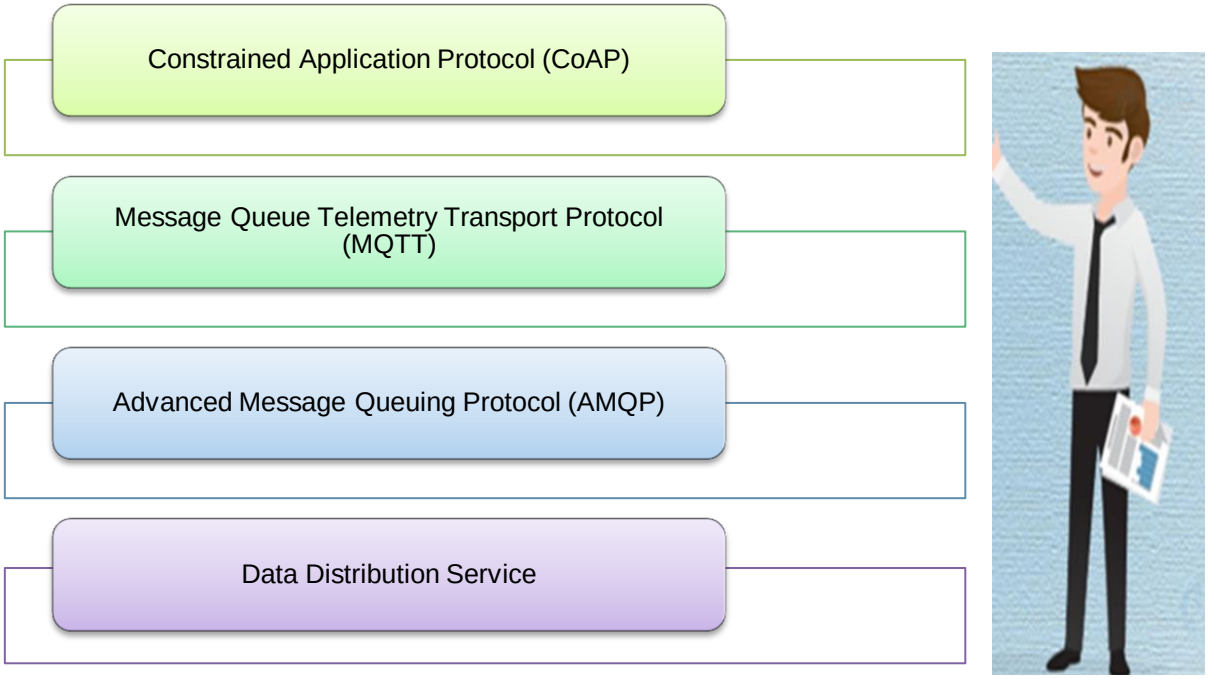


Figure 13: IoT protocols

- **Constrained Application Protocol (CoAP)**

CoAP is an internet utility protocol mainly developed for constrained smart gadgets. It is designed to enable simple, constrained devices to join IoT through constrained networks having low bandwidth availability. This protocol is primarily used for machine-to-machine (M2M) communication and is particularly designed for IoT systems that are based on HTTP protocols. It uses the protocol-UDP for implementation of lightweight data. It also uses restful architecture, which is just like the HTTP protocol. It is also used inside the mobiles and the other social communities that are basic programs. CoAP helps in getting rid of ambiguity through HTTP get, put up, delete and placed strategies [48].

- **Message Queue Telemetry Transport Protocol (MQTT)**

MQTT IoT is a message protocol developed in 1999 by Arlen Nipper (Arcom) and Andy Stanford-Clark (IBM.) This is mostly used for monitoring from a

remote area in IoT. Its main purpose is obtaining data from so many electrical devices and conveys them to the IT communications or infrastructure. A hub-and-spoke architecture is fundamentally ordinary for MQTT IoT Protocol. It works on top of the TCP to offer easy and dependable streams of information. This protocol is made of three core components or mechanisms: **Subscriber**, **Publisher**, and **Broker**. The **publisher** generates the information and transmits the facts to **subscribers** through the dealer. The **dealer** ensures security by checking and rechecking the authorization of the subscribers and the publishers [48][49].

- **Advanced Message Queuing Protocol (AMQP)**

AMQP was evolved by John O'Hara at JP Morgan Chase in London. It is an application layer protocol for message oriented and designed middleware environments [50]. The AMQP IoT messaging protocols got the approval as an international standard. The processing chain of AMQP IoT Protocol consists of 3 necessary components, and those are Exchange, Message Queue and Binding. The **Exchange** part receives messages from publisher primarily based programs and putting them in the **Message Queues**. And the **Message Queues** stores messages until they may completely process via the client software. The role of the **Binding** part is stating the connection between the Exchange component and the Message Queue component [51].

- **Data Distribution Service (DDS)**

DDS is a standard for high-performance, expandable and real-time machine-to-machine communication. It is developed and designed by Object Management Group (OMG). Thanks to this protocol, data can be transferred both in the low-footprint devices and with the cloud platforms. The DDS includes two significant layers. Those are the Data-Centric Publish-Subscribe (DCPS) works by delivering information to the subscribers and the Data-Local Reconstruction Layer (DLRL) which provides an interface to DCPS

functionalities, permitting the sharing of distributed data amongst IoT enabled objects [51].

2.6 Architecture and Characteristics

a) IoT Architecture

Different IoT architecture has been proposed in scientific literature. According to the research we reviewed, there are different opinions regarding the number of layers in IoT. The basic model is a three-layer architecture, which consists of Perception, Network, and Application layer. The authors in [50] proposed five layer architecture. In the same context, IoT World Forum (IWF) architecture committee released an IoT reference model in October 2014 [41]. All these models help the industry to accelerate IoT deployments. However, the most popular IoT architecture model is the one with five layers which consists of Perception, Network, Processing, Application and Business layer. They are briefly discussed and depicted in Figure 14 below.

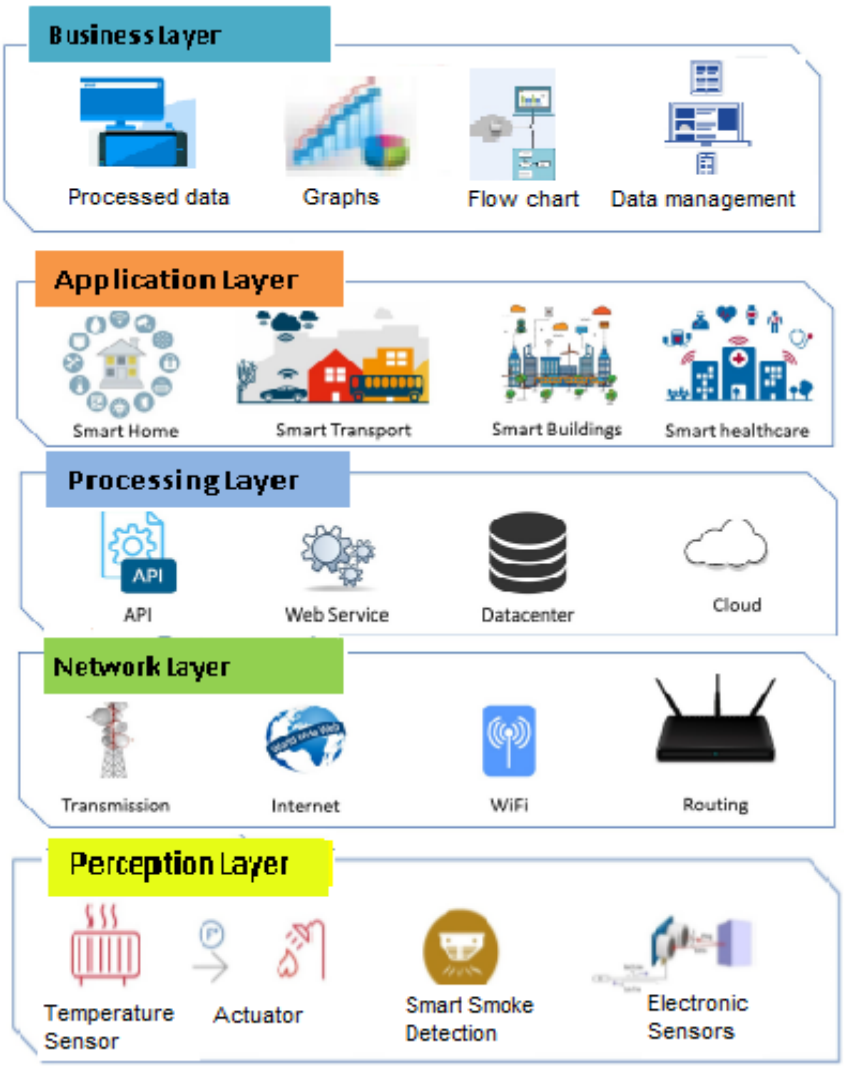


Figure 14: IoT archiecture layers

- **Perception Layer**

It is also known as sensors layer, it represents the physical sensors of the IoT that aim to collect and process information. This layer includes sensors and actuators to perform different functionalities such as querying location, temperature, weight, motion, vibration, acceleration, humidity, etc. Standardized plug-and-play mechanisms need to be used by the perception layer to configure heterogeneous objects [52]. The perception layer digitizes and transfers data to the Object Abstraction layer through secure channels. The big data created by the IoT are initiated at this layer [53][54].

- **Network Layer**

The network layer transfers data produced by the perception layer to the processing layer through secure channels. Data can be transferred through various technologies such as RFID, 3G, GSM, UMTS, Wi-Fi, Bluetooth Low Energy, infrared, ZigBee, and so forth [53]. Furthermore, other functions like cloud computing and data management processes are handled at this layer [54].

- **Processing Layer**

The processing layer is also known as a middleware layer. It collects the information that is sent from a network layer. Then, it performs processing onto the collected information. In addition, it has the responsibility to eliminate extra information that has no meaning and extracts the useful information. However, it also removes the problem of big data in IoT. In big data, a large amount of information is received which can affect performance of IoT. There are numerous attacks that can affect the processing layer and disturb the performance of IoT such as Exhaustion and Malwares [55].

- **Application Layer**

The application layer provides the services requested by customers. For instance, the application layer can provide temperature and air humidity measurements to the customer who asks for that data. The importance of this layer for the IoT is that it has the ability to provide high-quality smart services to meet customers 'needs. Moreover, it covers numerous vertical markets such as smart home, smart building, transportation, industrial automation and smart healthcare [55].

- **Business Layer**

It manages the overall IoT system activities and services. The responsibilities

of this layer are to build a business model, graphs, flowcharts, etc. based on the received data from the Application layer. It is also supposed to design, analyse, implement, evaluate, monitor, and develop IoT system related elements. The Business Layer makes it possible to support decision-making processes based on Big Data analysis. In addition, monitoring and management of the underlying four layers is achieved at this layer. Moreover, this layer compares the output of each layer with the expected output to enhance services and maintain users 'privacy [56].

b) IoT Characteristics

The most essential IoT characteristics are described below [57][58].

- **IoT Intelligence**

IoT is a combination of hardware and software along with complex algorithms and computations. The capabilities of IoT are enhanced due to its intelligence which enables them to respond and act according to the situation. The interaction between different devices is only because of its intelligence.

- **IoT Connectivity**

Connectivity in IoT enables it to connect various everyday use objects. This contributes to overall intelligence of IoT network. This also makes way for new market opportunities by creating network of smart objects and applications. Moreover, the network will be more accessible and compatible.

- **IoT Dynamic Nature**

The IoT devices capture data from its surrounding environment. This is done by the dynamic changes that take place around these devices. The state of IoT devices change dynamically like connected or disconnected and also due to temperature, location and speed. Also, it can change due to person, place or time.

- **IoT Enormity**

In the near future, the number of devices connected to the network for communication will be much larger than it is today. Also, it will become much more complex to manage and handle data from these devices. A statistics suggest that more than 5 million new devices are connected every day and the number is only going to increase.

- **IoT Sensing**

Sensors are an important component in IoT without which the changes in the environment cannot be detected and measured. These sensors interact with the environment to detect and collect data. The information that is sensed by the sensor is basically the input from the environment that can provide some valuable information.

- **IoT Diversity**

Diversity or heterogeneity is one of the main characteristics of IoT. The IoT devices have different hardware platforms and network and they are able to communicate with other devices through different networks. The IoT network is able to support connectivity between distinct networks. The core requirements for this diversity are scalability, modularity, extensibility, and interoperability.

- **IoT Security**

Currently there is some security and privacy issues with IoT network which with more development in this field will be vanished. It is very important to secure data while it is being transferred between devices.

2.7 IoT Security Issues

Security and privacy requirements for the IoT systems are more rigorous than those of the typical IoT-based infrastructures. IoT systems have many additional security requirements, such as device localization [59], which can also contribute to ensure the security and privacy of the systems. The functionalities of each level of the IoT systems are different, which means each level has different security and privacy requirements. Therefore, the requirements for each level are analysed and discussed individually.

a) Data Level

- **Confidentiality**

Collection and storage of data must comply with legal and ethical privacy regulations. To prevent breaches of data, adequate measures must be adopted to ensure the confidentiality of the data associated with clients. The importance of such measures cannot be overemphasized, as the data stolen by cyber criminals could be sold in illegal markets, causing the clients to suffer from not only privacy violation but also possible financial and reputational damages [60].

- **Integrity**

For IoT systems, the purpose of the data integrity requirement is to ensure that the data arriving at the intended destination have not been compromised in any way during the wireless transmission [61]. Attackers could gain access to and modify patient data by taking advantage of the broadcast characteristic of the wireless network, and which could lead to severe implications in life-threatening cases. To guarantee that the data have not been compromised, the capacity to detect potential unauthorized distortions or manipulations of the data is critical. Therefore, appropriate mechanisms of data integrity must be

implemented to prevent alteration of transferred data by malicious attacks. Moreover, the integrity of the data stored in the cloud servers also needs to be ensured, which means the data cannot be tempered with.

- **Availability**

Services and data must be accessible when they are required to the relevant users. Such services and data, provided by the cloud servers and devices, will become inaccessible if DoS attacks occur. Therefore, to accommodate the possibility of availability loss, applications must be always-on to ensure data availability to the users and companies [61].

b) Sensor Level

The security and privacy in the sensor level faces the most challenges of the 3-tier IoT system, due to the limited computational capability and power constraint of the IoT devices and sensors [62]. The current trend in sensor level security research is to put most of the computations in the personal server level instead, and the security measures in the sensor level are required to be light-weight and less communication overheads.

- **Tamper-Proof Hardware**

IoT devices, especially ambient sensors, can be stolen physically, which leads to security information being exposed to attackers. Furthermore, the stolen devices can be reprogrammed by attacker sander deployed to the system, listening to communications without being noticed [63]. Therefore, physical theft of IoT devices is a severe security threat and must be addressed in the IoT systems. IoT devices in systems should at least have tamper resistant integrated circuits, preventing codes loaded on the devices being read by third parties once being deployed.

- **Localization**

The former sensor localization is typically designed to identify whether the sensors are located in the desired body positions. Such on-body sensor position identification is of vital importance for applications such as activity recognition [64]. The later sensor localization, also known as Location of Things (LoT), is designed to locate the sensor in a room or to locate the patient wearing the sensor in a building. In IoT systems, IoT devices could move in and out of the network coverage very frequently. Therefore, a real time intrusion detection measure is required if the network allows its sensors to leave and re-join irregularly [65].

- **Forward and Backward Compatibility**

This is also a key requirement in real-time business applications where faulty IoT sensors are replaced promptly with new ones. Forward compatibility is characterized by the fact that future messages cannot be read by IoT sensors if their transmission occurs after the sensors have left the network. Conversely, in backward compatibility, messages that have been transmitted earlier cannot be read by a sensor which just entered the network [66].Compatibility issues can potentially be solved by implementing OTA programming for the distribution of the newest software update promptly.

3. Proposed Solution

From their emergence, the two concepts of Cloud Computing and IoT have evolved separately. For many years, they have seen independent evolution in their hardware and software aspects. In its evolution, IoT faces many problems among them storage capacity, energy efficiency, computational

capabilities [67]. To overcome these problems, scientists found that Cloud Computing could help in this context. That is why we think about how to combine the two concepts. The integration of Cloud Computing and IoT has brought several benefits for each of them. By visiting other network technologies, the Cloud Computing paradigm seems to be an answer in regards to its characteristics. It could fill some gaps of the IoT which has limitations with regard to the storage, networking and computing capabilities of different connected objects. IoT is also limited as far as energy, scalability, interoperability, flexibility, reliability, efficiency and availability are concerned [68] [69]. On the one hand, as the cloud Computing has virtually unlimited resources and capabilities, the IoT could be interested in this potential which can help in the compensation of its technological constraints such as processing, storage, and energy [70]. For instance, Cloud Computing can help to effectively implement many IoT applications. Some of them can be found among the solutions for IoT service management and composition, applications exploiting the produced data from devices. Otherwise, IoT can give to the Cloud Computing the chance to deal with real-world objects in a more dynamic and distributed manner for delivering new attractive services and applications in some real life scenarios. Hence, the need to merge the Cloud and IoT technologies emerge. This new concept is described in Figure 15 below.

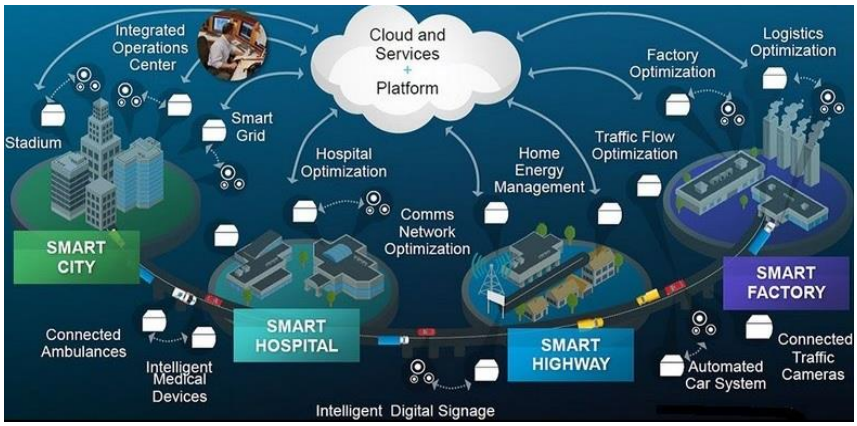


Figure 15: The integration of Coud and IoT [71]

These two concepts are different from each other though their features are often complementary as depicted in Table 4. Different studies have been proposed in the integration of Cloud and IoT because of this complementary relationship [71] [72].

Table 4: .Complementary features of Cloud and IoT

Features	Cloud Computing	Internet of Things (IoT)
Displacement	Centralised	Pervasive
Processing Capability	Virtually unlimited	Limited
Storage	Unlimited storage	Limited or no storage
Connectivity	Internet of delivering service	Internet is point of convergence
Components	Virtual resources	Real world things
Security &Privacy	Meet requirements	Easy to attack
Big Data	Way to manage data	Source of data

3.1 Proposed Architecture

One particularly important challenge faced by the Cloud-Enabled Internet of Things (CEIoT) approach is related to the extensive heterogeneity of devices, platforms, operating systems. In this respect, we add novel component that acts as a gateway to address this issue. Consequently, the proposed architecture, as illustrated in Figure 16, is composed of three layers namely Cloud, Gateway, Things.

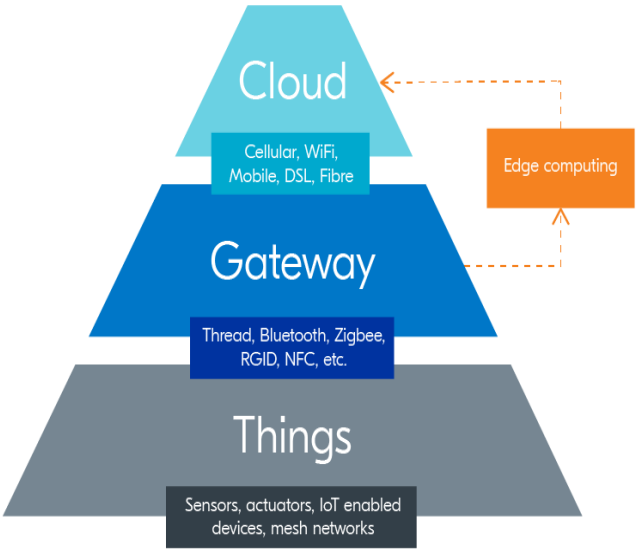


Figure 16: The core element of the proposed IoT architecture [73]

Functionally, the gateway sends data collected by sensors to Cloud Computing. Subsequently, client can access to these data through an appropriate API. In summary, our proposed framework has four components, i.e, Sensors, Gateway, Cloud, and Clients. A general overview of this architecture is presented in Figure 17 below.

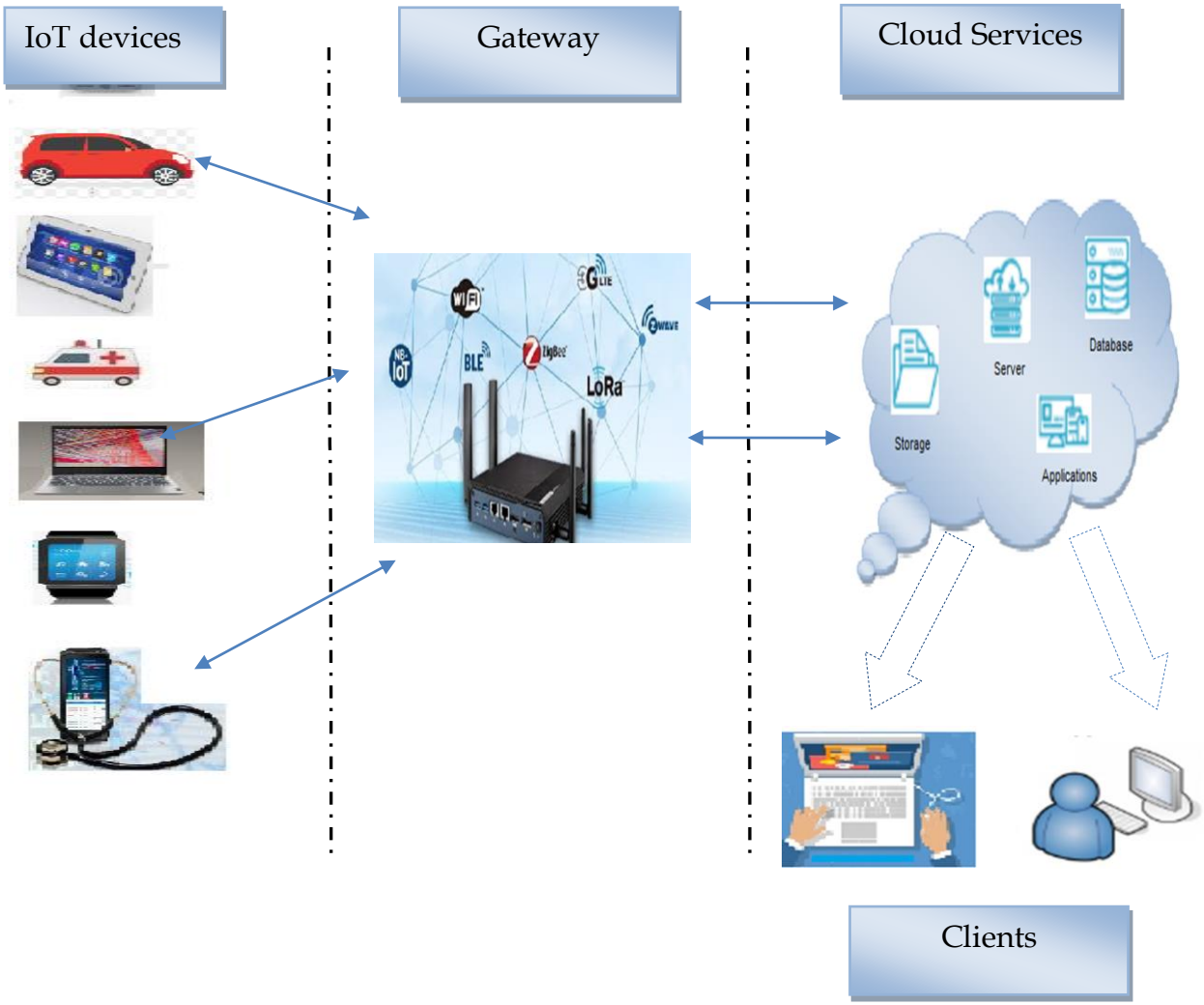


Figure 17: Proposed framework for Cloud-Enabled IoT

- **Sensors:** In an IoT system more than 20 billion devices are connected with each other and to external devices too, and all of which use sensors [73]. Sensors play an important role in our everyday life. Its main function is to gather data and detect change from the surroundings [74]. It located to the bottom layer. This layer includes connectivity to next layer like the getaway. There are many types of sensors in the market depending to IoT applications, some of them cited below:

  - ✓ Temperature Sensor

- ✓ Smoke Sensor
- ✓ Motion Sensor
- ✓ Humidity Sensor
- ✓ Pressure Sensor
- ✓ Gas Sensor etc.

- **Gateway:** It is an intermediate element in an IoT system to bridge the communication gap between IoT sensors and devices out in the field and the applications situated in the cloud [75][76]. In this regard, it is responsible for managing IoT network device compatibility, IoT security, configuration management of IoT devices, and device to IoT cloud connectivity. By systematically connecting the field and the cloud, IoT gateway devices offer local processing and storage solutions, as well as the ability to autonomously control field devices based on data input by sensors [77].

- **Cloud Service:** IoT systems frequently involve the use of Cloud Computing platforms. This latter offers the potential to use large amounts of resources, both in terms of the storage of data and also in the ability to bring flexible and scalable processing resources to the analysis of data [78]. IoT systems are likely to require the use of a variety of processing software and the adaptability of cloud services is likely to be required in order to deal with new requirements, firmware or system updates and offer new capabilities over time [79].

- **Clients:** Users can access to sensor information via a web interface to cloud services that provides processing power, data analytics, and storage capabilities. Where they can view statistics and graphs of the data collected from the IoT devices. For instance, in the healthcare field patients can access to their medical records and the doctors can also

access to their patients' data. In addition, doctors can follow up with their patients remotely.

3.2 Proposed Access Control Model for CEIoT

In this subsection, we discuss the motivation behind our proposed approach. We provide an overview of some basic concepts that we need in our proposal. Principally focusing on the OM-AM and an ABAC mechanism in a way designed to protect an IoT devices' access data as hosted by cloud resources.

a) OM-AM Reference

Despite of the concept CEIoT offers outstanding advantages, there are many challenges that hinder its successful implementation. In general, there exist several references to modulate privacy requirements and access control in a distributed system. We opt for OM-AM reference that contains four layers namely Objective, Model, Architecture, and Mechanism [80]. Actually, the objective and model (OM) layers articulate what the security objectives are and what should be achieved, while the architecture and mechanism (AM) layers address how to meet these requirements. Figure 18 shows an OM-AM reference diagram. The OM-AM reference resembles an OSI 7 layer network protocol stack. Like Open Systems Interconnection 7 layers, each OM-AM reference layer's mapping to adjacent layers is many-to-many. In other words, security policy can be formalized with many access control models as they can support different security policies. Moreover, an access control model, in its turn, can be supported by multiple architectures, while specific architecture can support multiple models and do not necessarily comply with the top-down waterfall-style software engineering process [81]. We argue our choice to adopt the OM-AM framework for analysis because it allows us [82]:

- ✓ To define in perspicuous way the boundaries and the

relationship between each phase in the authorization process, because each phase matches a specific layer.

- ✓ To discuss each phase independently from the other and as an example, discuss the security requirements separately from the mechanism required for its implementation.
- ✓ To compare in a vertical way different access control policies that encapsulate the same security policy or different architectures that implement the same access control model and different mechanisms that enforce the access control architecture
- ✓ To compare in a horizontal way within one layer for example between different access controls models/mechanisms.
- ✓ To design each layer separately, for example, design mechanisms that are able to enforce multiple policies [83]. This latter aspect is particularly crucial because it will give great flexibility and scalability to the whole access control system. In fact, if a tool in one layer is tied to a specific component in another layer, changing in the policy would require changing the whole access control system. Let us discuss in details each layer.

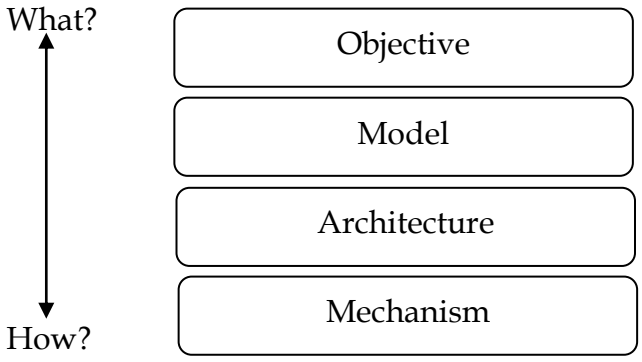


Figure 18: OM-AM reference [81]

- **Objective Layer**

The main function of this layer consists in defining an access control policy that defines the (high-level) rules according to which access control must be regulated.

- **Model Layer**

The role of this layer is to bridge the gap between high- level policies and low-level mechanisms by defining means of how authorization rules should be applied to protect resources. Actually, an authorization model is formalism for representing in a clear way the security policy [84]. In our proposed approach we adopt Attributes-Based Access Control as a model.

- **Architecture Layer**

The architecture typically includes components responsible for authentication, authorization and policy enforcement.

- **Mechanism Layer**

It defines the protocols and software implementations to enforce policies and define how access requests are evaluated against those policies [84].

b) Proposed Access Control Model

Many access control models have been proposed in the literature to address security issues in IoT. But the most recent and relevant ones are proposed by Sandhu [81][83]. In our proposed solution we adopt ABAC as an efficient access control model.

- **ABAC Model**

ABAC model makes an authorization decision based on attribute values required to allow access to a resource according to rules created through the

security administrator. Attributes are the set of properties that may be associated with a given entity. An entity may be a subject, resource or environment that is considered related to the interaction between a user and an application. Therefore, attributes may be subject attributes (subject's identity, role, age, zip code, IP address, memberships, citizenships, etc.), may be resource attributes (resource's identity, location, size, value, etc.), or may be environmental attributes (time of day, date, system state, etc.) [85]. ABAC is composed of two aspects: the policy model (sometimes referred as PBAC: Policy Based Access Control) and architecture model which applies the policy. Policy rules specify conditions under which access is granted or denied [86].

- ✓ **Attributes:** are characteristics of the subject, object, or environment conditions. Attributes contain information given by a name-value pair.
- ✓ **Subject:** is a human user or NPE, such as a device that issues access requests to perform operations on objects. Subjects are assigned one or more attributes. For the purpose of this document, assume that subject and user are synonymous.
- ✓ **Object or Resource:** is a system resource for which access is managed by the ABAC system, such as devices, files, records, tables, processes, programs, networks, or domains containing or receiving information. It can be the resource or requested entity, as well as anything upon which an operation may be performed by a subject including data, applications, services, devices, and networks.
- ✓ **Operation:** is the execution of a function at the request of a subject upon an object. Operations include read, write, edit, delete, copy, execute, and modify.

- ✓ **Policy:** is the representation of rules or relationships that makes it possible to determine if a requested access should be allowed, given the values of the attributes of the subject, object, and possibly environment conditions.

- ✓ **Environment conditions:** operational or situational context in which access requests occur. Environment conditions are detectable environmental characteristics. Environment characteristics are independent of subject or object, and may include the current time, day of the week, location of a user, or the current threat level.

- **AC-CEIoT Objective**

Robust security and privacy measures in CEIoT are the major factors that will help the wide adoption and continued success to IT services. In this respect, we need to define the access control policy that determines the (high-level) rules according to which access control must be regulated [87][82]. This layer provides required functions and expressions to conduct a circumstantial study of the system and its environment, as well as drawing the perimeter and the scope to be targeted. Moreover, we analyse the risk assessment and identify of the security objectives of the proposed solution. Concretely, a whole arsenal of methodologies, methods and tools are used in step, including Risk Assessment Methods, ISO/IEC 27002 and 27005. OTACE EBIOS, MAHARI, CRAMM, OWASP [82].

- **AC-CEIoT Model**

Access control models are the key components to achieve the objective of security and privacy. The purpose of a model is to formalize the satisfaction of these objectives. In this context, this layer comes to bridge the gap between high level policies and low-level mechanisms. It is a way for defining how authorization rules should be applied to protect cloud resources. An

authorization model is often a mathematical formalism for representing in a clear and unambiguous way the security policy. It helps to abstract it (i.e., reduce its complexity) and to facilitate its understanding. It can be used to verify that the policy is complete and consistent. Popular authorization model include discretionary model DAC, mandatory model MAC, RBAC [81], and usage control UCON [83] presented by some authors. We adopt the Attributes- Based Access Control (ABAC) for our proposed model thanks to its flexibility and scalability within a distributed system like CEIoT. In general, access control requires both authentication and authorization techniques, however, we scope our work here to the authorization models in a specific instance of IoT, the CEIoT. We do not consider how to implement ABAC solution systems using these key components. This is left to the architecture and mechanism layers.

- **AC-CEIoT Architecture**

This layer presents the entities, the workflow and interactions between them (centralized or decentralized). Given this set of entities, several authorization sequences can be defined for example: Push, Pull or Agent sequence. The most popular authorization architecture is published by an ISO standard for the access control framework ISO/IEC 10181-3 [82][87] that defines the main features of the reference monitor. According to that standard, a reference monitor consists of two basic components: an access enforcement facility (AEF) or a policy enforcement point (PEP) and an access decision facility (ADF) or a policy decision point (PDP). Every request made by a subject is intercepted by the AEF/PEP and then forwarded to the ADF/PDP for an access decision evaluation. The ADF/PDP may reply either yes/grant or no/deny depending on the security policy, while the AEF/PEP enforces this decision appropriately. Tools in this layer could be in form of protocols or framework such as Oauth protocol and XACML standard. In our proposed framework we chose XACML as language to evaluate the policies mentioned above.

- **AC-CEIoT Mechanism**

This layer defines the low level (hardware and software) functions to enforce

policies and define how access requests are evaluated against those policies. Actually, configuring access control policies is a non-trivial and highly critical process, and it should be subject to periodic review and verification to ensure that security policies are correctly expressed and implemented. Proposed verification methods include formally testable policy specification, detection of anomalies or connecting rules via segmentation technique, and analysis tools that enable policy administrators to evaluate policy interpretations. A plethora of tools belongs to this layer such as: ACLs, Routers, Encryption, Audit logs, IDS, Antivirus software, Firewalls, Smart cards, Dial-up call-back systems, alarms and alerts among others [87].

4. Summary

The highly dynamic nature of Cloud-Enabled IoT poses access control challenges rendering standard access control policies, models, and mechanisms unsuitable for to CEIoT scenario. In this chapter, we suggested the OM-AM reference to manage security and privacy concerns in CEIoT environments. In fact, strong identification, authentication and authorization mechanisms are necessary in distributed and dynamic systems. This chapter presented a novel access control framework adapted to specific applications in the context of CEIoT environments needs. We adopted the OM-AM concept to ABAC model as a way to protect IoT device 'data stored in cloud. We also relied for this novel access control framework XACML language to ensure a powerful security policy regarding ABAC model.

CHAPTER 3: Implementing Policy Rules in Attributes Based Access Control with XACML within a Cloud-Enabled IoT Environment

This chapter presents an implementation of our proposed solution in the chapter 3, as we mentioned above there are many languages policy to build access control policies into a distributed environment like CEIoT. For this purpose, we chose eXtensible Access Control Markup Language (XACML) in order to make access decisions. This chapter is structured as follows. Firstly, we discuss the motivation and scope behind the current chapter (section 1); and secondly, a detailed overview of Cloud–Enabled Internet of Things (CEIoT) is discussed (section 2). Then, we introduce an access control within CEIoT (section 3). The implementation of the proposed policy based on ABAC model and an XACML editor is presented in (section 4) with simulations and results. Finally, we end this chapter with some concluding remarks (section 5).

1. Motivation and Scope

The Internet of Things (IoT) extends internet connectivity to a wide range of smart devices. However, battery autonomy, computational capability and storage capacity are major technology challenges that hinder increased implementation and adoption. Although the integration of the Internet of Things (IoT) with Cloud Computing is considered as a highly promising solution in overcoming these bottlenecks, it raises security concerns, especially access control. Recently, a variety of access control models have been developed to help protect confidential information and restrict access to

sensitive data [88]. IoT is flexible and scalable, the consensus is that the Attribute Based Access Control (ABAC) is the most appropriate model in a dynamic environment. In the context of IoT, the ABAC model has the ability to enforce data privacy and ensure a secure connection between IoT devices and cloud providers. One of the core components of the ABAC model is access policies, these are used to deny or allow user' requests. To achieve that, an access policy language is required to implement policy rules in ABAC model [89]. In this chapter, we propose a novel approach to implement the access policies in order to prevent all unauthorized access to remote resources. These policies can be implemented using an access policy language. In this regards, many access policy languages are proposed. Our proposed method is based on Extensible Access Control Markup Language (XACML) as being the most efficient and appropriate and appropriate technique within CEIoT due to its compatibility with heterogonous platforms.

2. Cloud-Enabled Internet of Things

Cloud Computing and IoT are two different technologies. Each has its advantages and limitations. IoT is characterized by widely-distributed devices with limited processing capabilities and data storage; these devices can be connected, utilized remotely. Obviously, the connection to Cloud resources would address some IoT limitations, since these two technologies are complementary [90].

This has led to a natural merging which can partially resolve most of the issues faced by IoT. Hence, CEIoT is changing the current and future environment of internetworking services [91].

Suppose that Cloud' characteristics are A, IoT characteristics are B, and Cloud Enabled characteristics are C, we can write this formula:

$$A \cap B = \emptyset \quad \text{and} \quad A \cup B = C$$

2.1 CEIoT Benefits

The Cloud can also benefit from IoT by developing its limits with real world objects in a more dynamic and distributed way, and providing new services for billions of devices in different real life scenario [90] [91]. Likewise, CEIoT has advantages, and has several benefits:

- **Economic benefits:** computing and storage resources paid for only as used reducing costs and potentially improve earnings.
- **Safe:** critical and important data stored in the Cloud can be accessed by any device or machine and remotely recover them.
- **Efficient growth:** the capacity of resources needed in the Cloud is used.
- **Elastic and scalable:** resources used can be increased and reduced according to the business demand.
- **Avoiding downtime or delay:** in case of a node failing, its load is taken up by the second Cloud node, so the business site is never down.
- **Disaster recovery:** big business can afford additional IT resources for disaster management.
- **Environmental compatibility:** cloud conserves energy and provides efficient technical solutions to the business.

Given those benefits, CEIoT had been applied in several domains as discussed below [92][93][94].

2.2 CEIoT Application's Domain

The Cloud-Enabled IoT paradigm has introduced a number of applications and smart services, which have affected end users 'daily lives. A brief discussion of certain applications which have been improved by the CEIoT concept is discussed and illustrated in Figure 19 below.

- **Healthcare:** CEIoT improves the quality and effectiveness of service, bringing high value for patients with chronic conditions. It implements smart features into medical devices and related software systems, keeping the domain innovative. The CEIoT provides many services in the healthcare field.

- **Logistics:** CEIoT can help mainstream logistics systems to deal automatically with complexity and changes. It provides new interesting scenarios and allows the easy and automated management of flows of goods between the point of origin and the point of consumption, in order to meet specific requirements expressed in terms of time, cost or means of transport [95]. CEIoT do a remarkable job in terms of reducing core management complexities which are problematic for many companies across sectors.

- **Energy:** cloud computing and the IoT can work together effectively to provide consumers with smart management of energy consumption. CEIoT solutions have come up with a practical processing formula to compensate for the increasing sophistication of the energy distribution networks and achieve real-time visibility of the consumption process (e.g. smart meters, smart appliances, smart lighting, remote infrastructure maintenance, renewable energy resources, smart grid asset monitoring) [96].

- **Home and Buildings:** CEIoT applications transform our everyday objects into information appliances by connecting them to internet so as to remotely monitor their behaviour (e.g. electrical power distribution, consumption of water, gas emissions, safety systems, lighting, heating and air conditioning).

- **Mobility and Transportation:** several existing challenges in this field have been solved thanks to CEIoT applications, which bring business benefits, such as increasing road safety, reducing road congestion, managing traffic and parking, performing warranty analysis and recommending car maintenance or repairs [97].

- **Media:** the CEIoT is an incredible opportunity for Telco'-s operational to reduce costs by applying IoT technology for software defined networking and network function virtualization.

Figure 19: CEIoT application domains

2.3 CEIoT Issues

As CEIoT provides several benefits and promotes the improvement of many applications, the merging of those paradigms results in many issues, challenges and research problems. These challenges are discussed in detail in our paper [98].

- **Heterogeneity:** by merging Cloud and IoT, a large number of heterogeneous devices, platforms, services and operating systems can be synchronized used for new applications.
- **Big data:** CEIoT produces a huge amount of data, which then requires particular attention to transfer, storage, access and processing.
- **Large scale:** achieving the required computational capability and storage capacity is becoming difficult in new applications, and the IoT devices have to deal with connectivity issues and latency dynamics.
- **Performance:** CEIoT applications initiate specific performance and Quality of Service (QoS) at different levels including communication, computation and storage aspects.
- **Monitoring:** CEIoT applications are impacted by volume, variety and velocity characteristics of IoT.

Privacy and security: because of their distributed architecture, CEIoT systems are exposed to several potential attacks and have common vulnerabilities. Another important issue that has not yet been resolved is how to provide an appropriate authorization rules and policies while ensuring that only authorized users have access to the sensitive data. This challenge is called

access control and it will be discussed in more details in the following section. Figure 20 depicted the CEIoT issues discussed above.

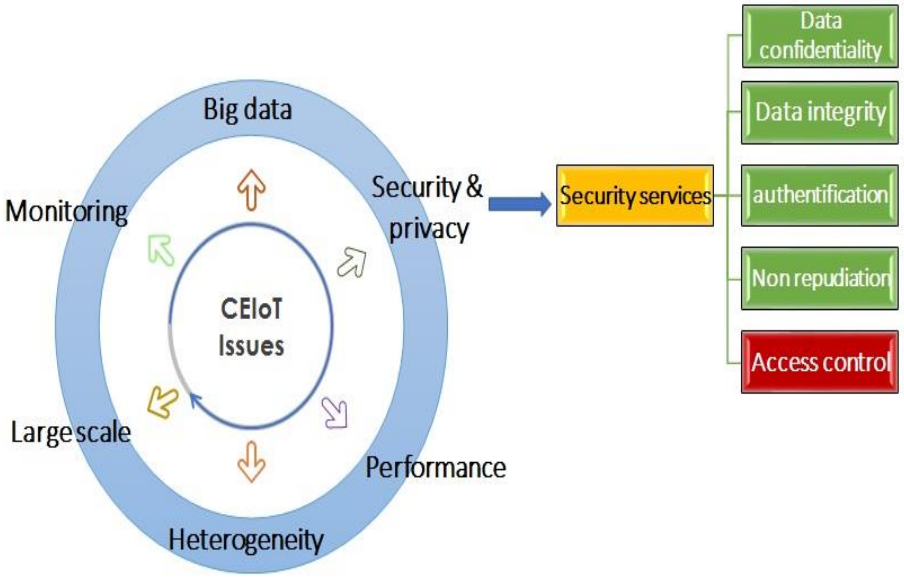


Figure 20: CEIoT issues

3. Access Control in Cloud-Enabled Internet of Things

Access Control is an important component to take into account in terms of security and privacy within a dynamic environment such Cloud-Enabled IoT. It makes sure that only trusted users can access the resource. Thus, it governs the actions that maybe taken on objects. In the context of CEIoT, the Cloud provider will offer access to resources such as file, record, data and so forth, to ensure that only the correct subject (IoT devices or device's owner) can access its services. The access control system consists of five elements (e.g. Subject, Object, Action, Permission, and Environment). These elements are described below.

- ✓ **Subject:** the entity that requests data access (e.g. Users or group of users).
- ✓ **Object:** the entity can be accessed (e.g. File, Computer, Database…).
- ✓ **Permission:** defines the type of access that is granted to a subject or group of subject for an object.
- ✓ **Action:** the operation to be performed on the object (e.g. Write, Delete, Update…).
- ✓ **Environment:** the context in which the action will be performed on an object (e.g. time, location…).

Authentication and Authorization are the main aspects in access control. Authentication verifies who will be given access to the resource, while Authorization defines the rights and privileges of the user identified and what actions they are authorized to undertake [99]. In this context, we proposed a new framework based in ABAC model and OM-AM reference. According to the research, ABAC also named Policy Based Access Control is the most appropriate model in a dynamic environment due to its flexibility, scalability and many other characteristics. Making a decision in favour of the ABAC model is based on evaluating policy rules against user 'attributes. So, these policies can be implemented using an access policy language eXtensible Access Control Markup Language (XACML) as an example. The following subsection will offer more details about XACML and the purpose behind choosing this language more than others.

3.1 Design a Security Policy Using XACML

Extensible Access Control Markup Language (XACML) is a standard language widely adopted in the field of information security especially access control. It uses Extensible Markup Language (XML) as the internal format. XACML has several advantages, some of which are: extensibility, flexibility, reusability, scalability and a policy referencing mechanism. Besides this, it provides a high level of granularity and the adaptability necessary to express and enforce security policy [100][101]. Practically, XACML includes three parts as shown in Figure 21: The request/response language expresses queries about whether a particular access should be allowed (requests) and describes answers to those queries (responses). The reference architecture proposes a standard for deployment of necessary software modules within an infrastructure to allow efficient enforcement of policies.

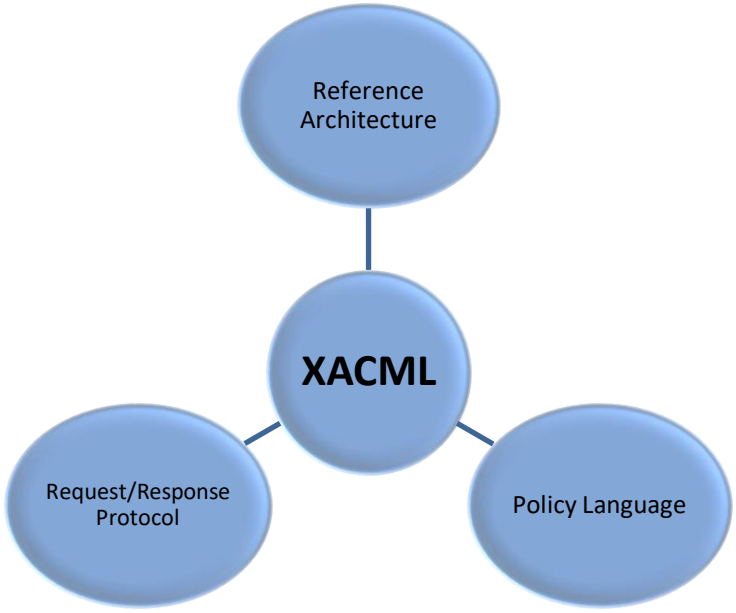


Figure 21: The core components of XACML language

XACML architecture has four logical functions as depicted in the Figure 22. [102].

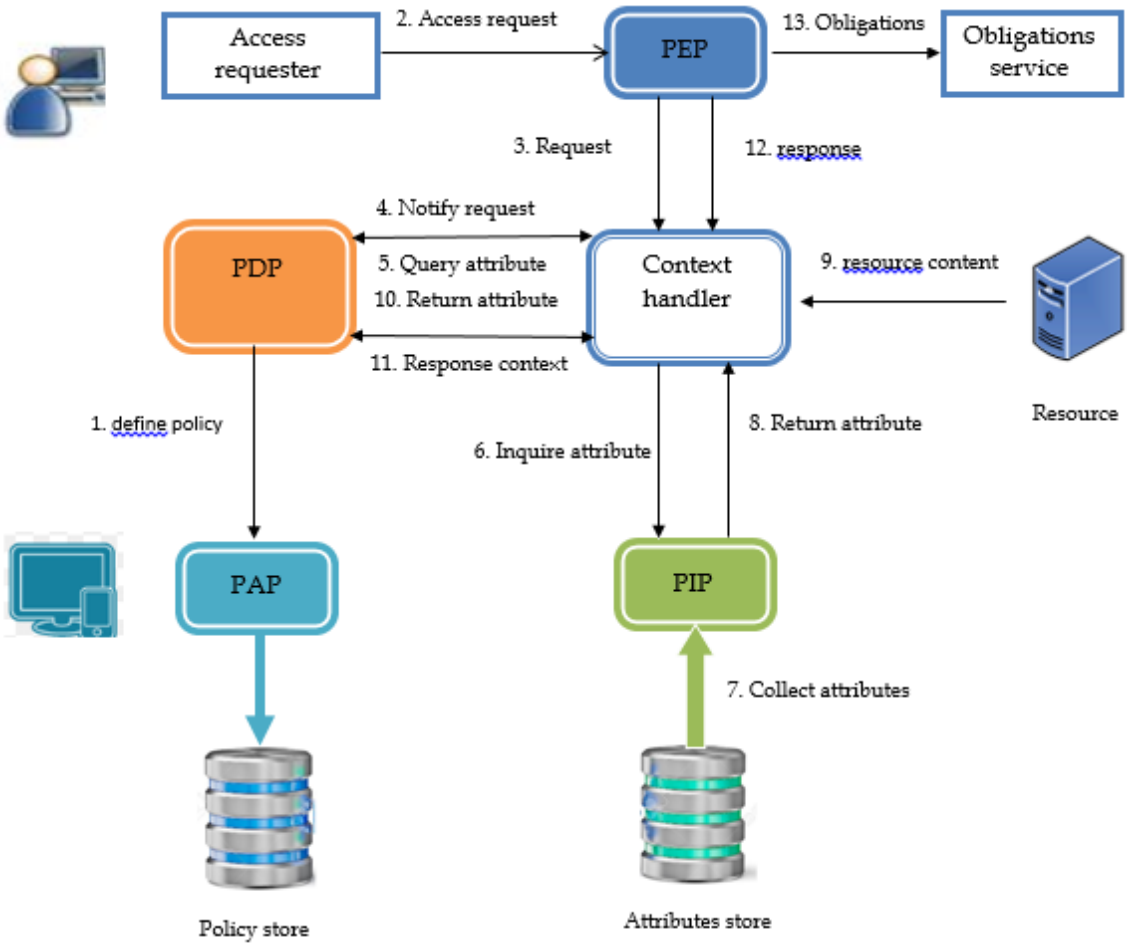


Figure 22: XACML architecture

- **The Policy Decision Point (PDP)**

The PDP receives an XACML request, fetches the applicable policy(s) from the policy administration point, retrieves the attribute values from the policy enforcement point, evaluates the request against the applicable access control policies and returns an authorization decision to the PEP [103].

- **The Policy Enforcement Point (PEP)**

The PEP receives an access request, extracts the attributes in the request, generates an XACML request and sends it to the PDP for evaluation. It also

makes sure that all the obligations with an authorization decision are executed. An obligation is an action that should be performed together with the enforcement of an authorization decision, and is specified in an access control policy [103].

- **The Policy Administration Point (PAP)**

The PAP creates an XACML access control policy(s) and stores it in a policy database server. It addition to this, it sets a restriction in order to prevent unauthorized access to the access control policies. Beside this, it conducts a regular check in order to maintain the uniqueness of policy identifiers. [104]

- **The Context Handler**

The Context Handler converts request and responses between native formats and the XACML canonical representation and coordinates, with the PIPs, gathering of the required attributes values.

- **The Policy Information Point (PIP)**

The PIP is a component that acts as a directory server that stores the attribute values and makes it available to the PDP. Attributes values are the data that describe the characteristics of a subject, resource, action and environment. Examples of attributes include: the subject's name, ID and login ID, the time of the day, the name of the resource, and what action (read, write, execute) is required [104].

The XACML policy language is composed of several hierarchical objects depicted in Figure 23 [105][106]. And the most three core structural elements are PolicySet, Policy, and Rule.

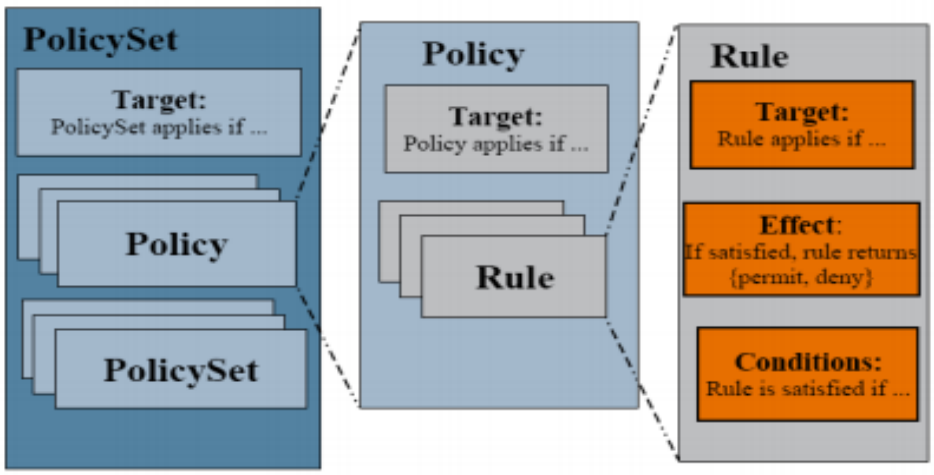


Figure 23: XACML policy language

- **PolicySet:** It is a container that can hold other Policies or PolicySets, as well as references to policies found in remote locations.
- **Policy:** It represents a single access control policy, expressed through a set of Rules.
- **Rule:** Rule within a policy is evaluated as to whether it should grant access to a resource**.**

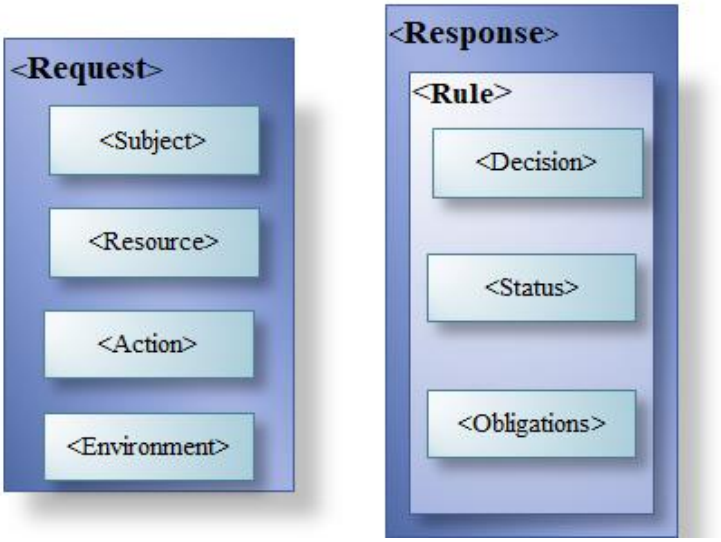


Figure 24: XACML request/response protocol

Even Though XACML was introduced a long time ago, but, there is still lack of interest of the organizations in using XACML for authorization. Many researchers in academia and industry do not have a clear understanding on the importance and advantages of XACML. It is also hard to implement a XACML solution due to some complexity in defining and managing XACML policies. However, there is an emerging motivation for XACML-based authorization systems in the current market. So, it would be economically beneficial if there is a better tooling component available such that it reduces the existing complexity in dealing with XACML [107]. The following are some of the existing XACML editors:

3.2 Exciting XACML Editors

a) WSO2 Identity Server XACML Editor

WSO2 Identity Server (WSO2 SI) is an identity and access management server that facilitates security, while connecting and managing multiple identities across different applications. It enables enterprise architects and developers to

improve customer experience through a secure single sign-on environment. In the context of XACML, WSO2 SI Management Console provides functionality to write and evaluate XACML Policies. The XACML support of the server provides two methods of creating XACML Policies [107]. One of the nice features of this UI is that you can plug any attribute value sources and select those attribute values when creating the policy; rather than filling text boxes by your own. By default, WSO2 registry resources, Roles of the underline user store and some pre-defined actions are the attribute value sources for resource, subject and action attributes respectively.

However, this editor has some deficiencies as it is difficult to get an overview over the policies without reading the generated XACML and you cannot create complex policies using it.

- **Writing a XACML Policy using a Policy Template**

Identity server gives some predefined commonly used XACML Policy template samples where users can use and edit according to their use case.

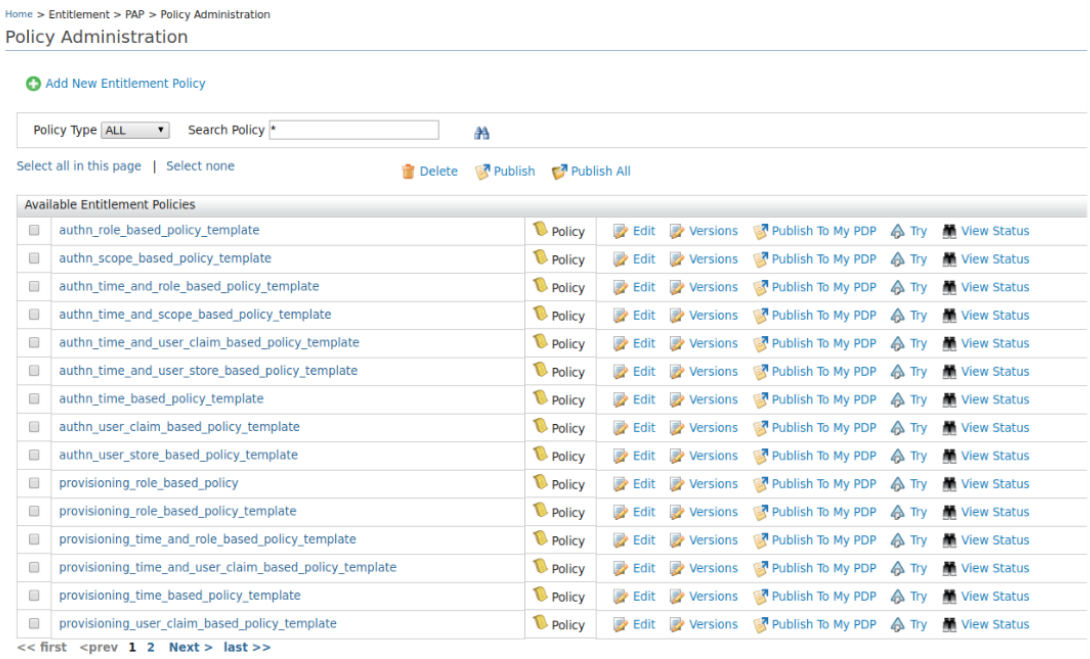


Figure 25: List of available policy templates in WSO2 IS [107]

- **Creating a XACML policy using UI configurations**

A set of UI editors are available in the management console of Identity server that provides functionality to create XACML policies. Following is the Policy Administration panel of WSO2 Management Console.

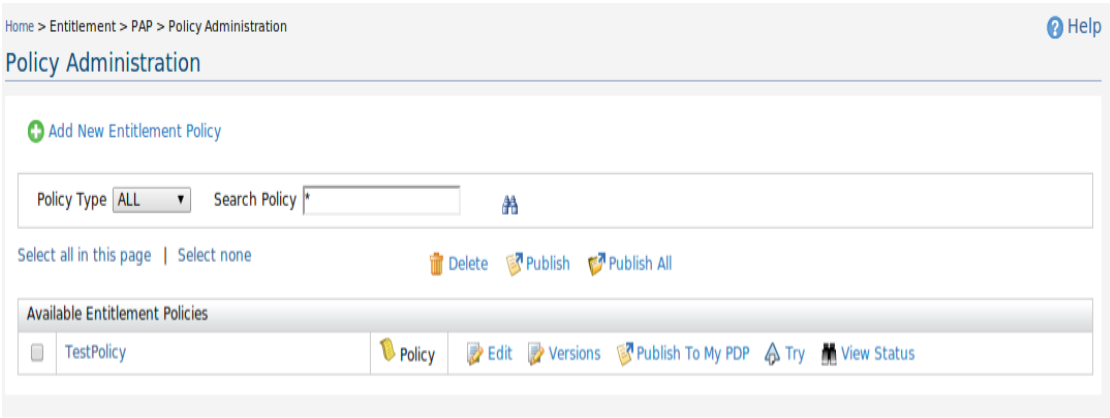


Figure 26: Policy administration panel in WSO2 IS [107]

There are six ways of writing an XACML 3.0 policy:

- ✓ Simple Policy Editor
- ✓ Basic Policy Editor
- ✓ Standard Policy Editor
- ✓ Policy Set Editor
- ✓ Import Existing Policy
- ✓ Write Policy in XML

b) UMU- XACML-Editor

UMU-XACML-Editor is an open source policy editor for XACML, it is written in Java. This editor is developed by the University of Murcia (UMU) [108]. It manages a DOM tree with XACML nodes, and provides a user interface with sensible default values or choices for each type of DOM nodes in the XACML

document. The editor supports folding down elements within a given policy in order to view parts of the DOM tree. It is actually intended to cover XACML 2.0 standard policies as defined by the OASIS eXtensible Access Control Markup Language (XACML). In this editor a new policy set or a policy can be created by right clicking the policy document option in the policy window. Then thereby a policy, descriptions, rule, condition, obligation, rule combining algorithm, combining parameters and so other options can be added. The created policy appears as a tree structure. The user can fill the respective values of the policy elements created by the right side window of the editor panel. Following is a snapshot of a simple policy using UMU XACML Editor [108][109].

This editor has similar deficiencies as the WSO2 Identity Server editor since the details of each XACML element is shown in a separate window, which means that it is not possible for a policy-maker to get an overview over how a given policy works without reading the generated XACML [109].

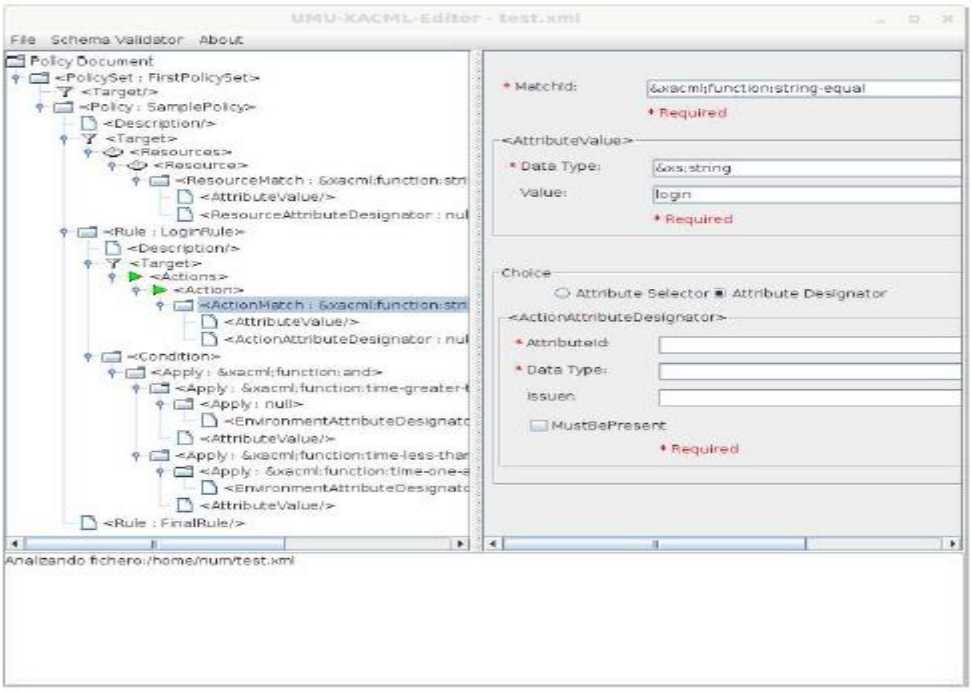


Figure 27: Simple policy using UMU XACML Editor [109]

To evaluate the policy there is an option in the menu bar called validator schema, once you click on this option, a window appears and it contains a validation button for accepting the policy and the schema as illustrated on Figure 28 below.

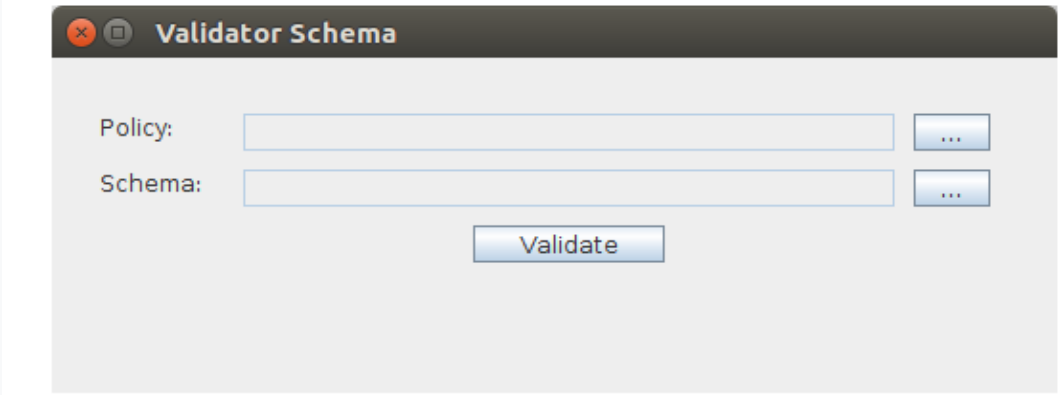


Figure 28: Evaluating XACML policy in UMU-XACML-Editor [108]

c) Axiomatic Policy Server's XACML Policy Editor

The XACML Policy Editor is an easy-to-use rich with web-based graphical user interface for access control policy creation and editing. It is designed to facilitate this transition by giving policy authors the power to represent policies graphically and policy administrators the means to verify. The editor offers an enhanced attribute dictionary with the notion of namespaces. This also has the support for REST and JSON for a fast, easy, and simple integration with a wide range of applications from mobile to single page to enterprise. Figure 29 depicted a snapshot of the graphical user interface of Axiomatic XACML policy generation panel [109][110].

Figure 29: Axiomatics policy server's XACML policy editor [108]

d) Security Policy Tool (SPT)

Security Policy Tool is a Free Graphic User Interface (GUI) based XACML Editor that helps to design, test, and verify security policies to prevent access control leakage. It can be used as an access control solution empowering organizations to enhance access control security of any IoT and Cloud systems. In this context, it allows the user to verify the XACML rules in terms of security before operations, test the policies to help the user prevent access control leaks as a result of these errors and save time and cost in development of XACML policies. In this regard, it automatically converts any access control model into XACML policies. Thanks to this tool we can ensure that the access control policies are fully safe before implementing them into an Access Control System as shown in Figure 28 [111].

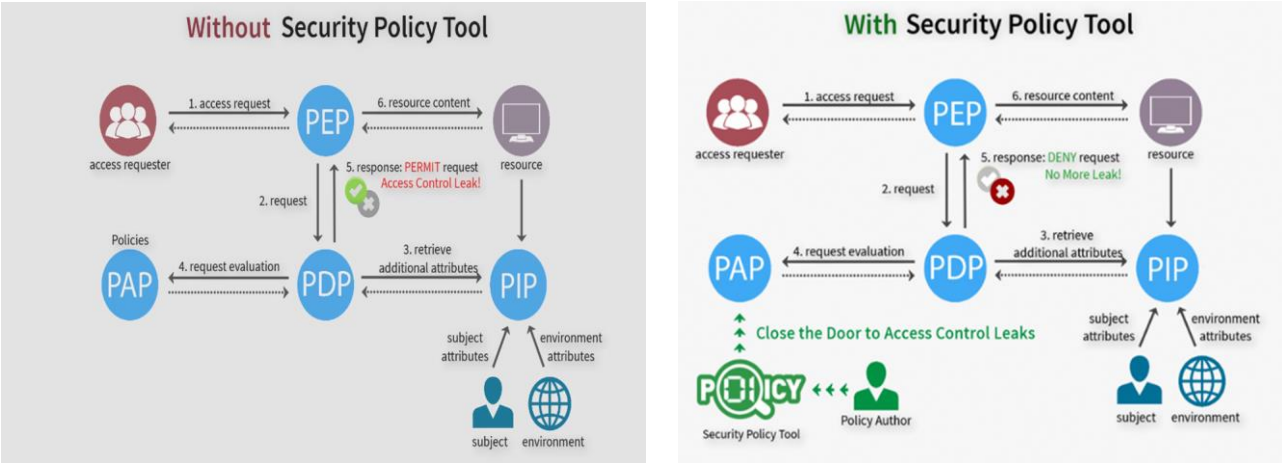


Figure 30 : Preventing access control leakage using SPT [112]

- **SPT Key Benefits**

Security Policy Tool helps the organizations to keep the access control policies free from vulnerabilities while also saving time and cost in the process. Here a few SPT benefits:

- ✓ **User-Friendly:** Enabling efficient policy composition and analysis.
- ✓ **Leak Prevention:** Comprehensive testing and verification enables you to identify hidden errors that could otherwise lead to access control leaks (e.g., data breach).
- ✓ **Automation:** Generate (> 99%) policy testing coverage, unachievable with manual testing approaches
- ✓ **Save Cost:** Intuitive design reduces the time and cost required for policy development and modification.

- **SPT Key Functions**

The policies written in XACML give the information and direction needed for the system to deliver an access decision (e.g., Deny, Permit) when a

request is made. These policies containing unknown errors or logical flaws can enable malicious individuals/groups to exploit these unsecured organizations. The SPT enables powerful AC testing and analysing functions such that the policy authors can validate and fix the faulty, unintended, misconfigured policies. Here the ways to close the door to access control leaks:

- ✓ **Security Model and Policy Editing:** SPT has user-friendly GUI (Graphic User Interface) for you to conveniently compose/edit AC Attributes (Subject, Resource, Action, and Environment), Conditions, Rules, Policies, and algorithms. AC model templates, e.g., ABAC (Attributed-based Access Control), MLS (Multilevel Security), and Workflows, are provided for systematically policy editing, modification, and updating. Policy inheritance allows the policy authors to effectively compose and manage the policies for a large hierarchical organization [112].
- ✓ **Policy Testing:** SPT presents numerous and rich functions for comprehensive policy tests to verify the AC policies against security requirements in a systematic environment. Giving the security requirements, one or a set of policies can be tested for policy leakage discovery [113]:
    - Merged Policy Verification
    - Combined Policy Verification
    - Merged Policy Separation of Duty
    - Combined Policy Separation of Duty

These tests support all rule combination algorithms such as First Applicable. The Combinatorial Test Suite feature can automatically

generate a substantial amount of test cases enabling rigorous policy model verification efficiently. Manual, human testing does not feasibly allow full-coverage testing; however, with Security Policy Tool full-coverage can be achieved with just a few clicks of your mouse [114].

- ✓ **Policy Analysing and Verification:** SPT enables a policy author to analyse the rules and policies of their AC authentication consequences in responses to the various AC requests. Potential security vulnerabilities in the policies could be detected in order to prevent AC flaws before these policies are deployed. AC Separation of Duty/Conflicts of Interest can be identified by analysing the testing results. From the analysis, the policy author can fix the problematic policies with new tests and analysis till the intended AC security goal is achieved [113] [114].
- ✓ **XACML Converter and Editor:** PT can automatically input and convert XACML 2.0/3.0 documents into AC models in SPT data format. It further has an XACML 3.0 editor to reduce the mistakes caused by policy manually editing. It is able to automatically convert the SPT data into XACML 3.0 policy format and output it for portability [115].

4. Implementing the Proposed Policy Based on ABAC Model

In this section, we discuss the tools we used to implement our proposed approach and the simulation results of this approach. As presented in our previous work [98], the ABAC model is the most appropriate model within a dynamic environment such as CEIoT due to its flexibility and scalability. In case of access requests, the ABAC engine's decision is based on users,

resource, environment attributes and set of policies. To implement the ABAC rules we chose XACML as a language to evaluate requests for resources according to rules defined in policies, because it is one of most popular methods for managing data access [115].

a) XACML Editor Used (SPT)/ Used Tool

As outlined above, there exist various XACML editors. The simplest one is SPT, so, we chose it to implement the proposed security policy. As discussed in section (4.3.2.4), in the SPT editor, the access permissions are granted to a request using a set of policies. In our implementation, we used this tool to evaluate, test and validate the access control policies, in order to ensure data privacy in a system. Besides, it is an efficient access control for testing and verifying security policies. More importantly, it provides an interface to design XACML access control policies more efficiently, as illustrated in Figure 30 above.

4.1 Simulation and Results

This subsection is intended for simulation and results. We simulate a University Library as a case study. Test Case (Hospital Test Case): In our implementation, we treat scenario, a Doctor, Manager, and Patient as subjects who request access to Patient's Medical Records, Personal Information, and Private Notes as objects in order to execute the View an Add actions. These actions are limited by an environment and condition attribute in our study. We suppose time as an attribute, allowing the subject access to resources for a limited, predefined time under an emergency condition. Implementing policy rules in a security policy tool requires four steps as depicted in Figure 31 below.

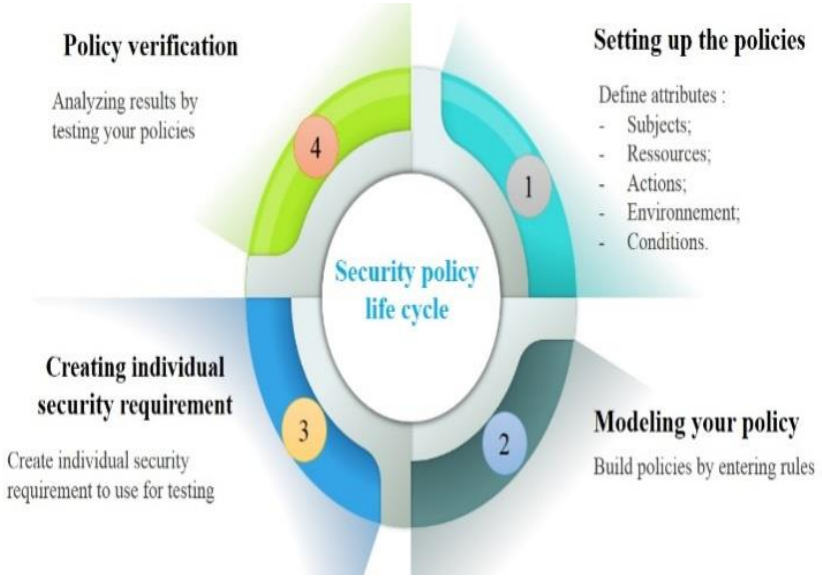


Figure 31: Security policy life cycle

4.2 Setting the policies

The Hospital Test case contains three policies (Doctor Policy, Manager Policy, and Patient Policy). Each policy includes several attributes values. The Doctor Policy attributes are: Dentist, Gynecologist, and Psychologist. The Manager Policy has one attribute which is Hospital Manager. The Patient Policy has two attributes which are PatientA and PatientB. After defining different attributes and attributes values the setting up of these policies is shown as below in Figure 32.

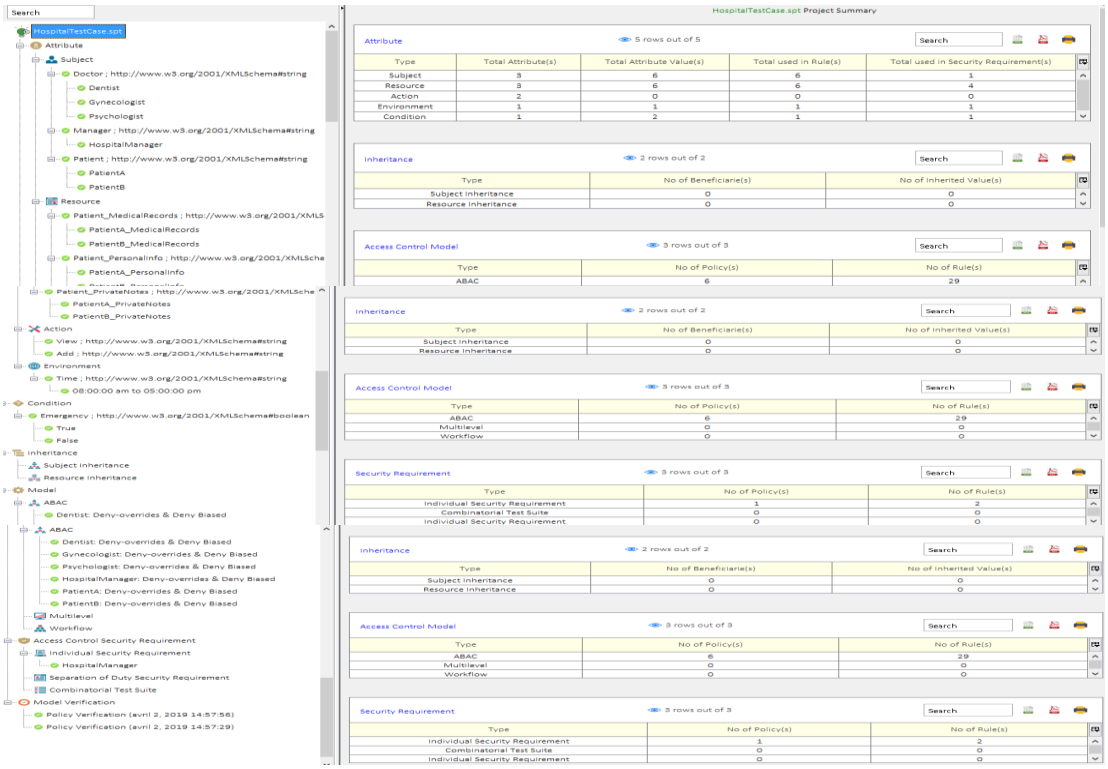


Figure 32: TestCase interface

4.3 Modeling the policies

After the setting up of the policies, we can model them by entering the rules. Each policy has various rules as illustrated in the list below.

- **DoctorPolicy**

(Subject = Any Value & Doctor = Dentist, View&Add, PatientA_MedicalRecords, Time= 08:00:00 am to 05:00:00 pm) →Permit

(Subject = Any Value & Doctor = Dentist, Any Action, PatientB_MedicalRecords, Any Value) →Deny

(Subject = Any Value & Doctor = Dentist, Any Action, PatientA_PersonalInfo&PatientB_PersonalInfo, Any Value) →Deny

(Subject = Any Value & Doctor = Dentist, Any Action, PatientA_PersonalInfo, Time= 08:00:00 am to 05:00:00 pm, Emergency=True) →Permit

(Subject = Any Value & Doctor = Dentist, Any Action, PatientA_PrivateNotes, Any Value) →Permit

(Subject = Any Value & Doctor = Dentist, Any Action, PatientB_PrivateNotes, Any Value) →Deny

(Subject = Any Value & Doctor = Gynecologist, View&Add, PatientB_MedicalRecords, Time= 08:00:00 am to 05:00:00 pm) Permit

(Subject = Any Value & Doctor = Gynecologist, Any Action, PatientA_MedicalRecords, Any Value) →Deny

(Subject = Any Value & Doctor = Gynecologist, Any Action, PatientB_PersonalInfo&PatientA_PersonalInfo, Any Value) →Deny

(Subject = Any Value & Doctor = Gynecologist, Any Action, PatientB_PersonalInfo, Time= 08:00:00 am to 05:00:00 pm, Emergency=True) →Permit

(Subject = Any Value & Doctor = Gynecologist, Any Action, PatienB_PrivateNotes, Any Value) →Permit

(Subject = Any Value & Doctor = Gynecologist, Any Action, PatientA_PrivateNotes, Any Value) →Deny

(Subject = Any Value & Doctor = Psychologist, View&Add, PatientA_MedicalRecords&PatientB_MedicalRecords, Time= 08:00:00 am to 05:00:00 pm) →Permit

(Subject = Any Value & Doctor = Psychologist, View, PatientA_PersonalInfo&PatientB_PersonalInfo, Time= 08:00:00 am to 05:00:00 pm) →Permit

(Subject = Any Value & Doctor = Psychologist, Any Action, PatientA_PrivateNotes&PatientB_PrivateNotes, Time= 08:00:00 am to 05:00:00 pm) →Permit

- **ManagerPolicy**

(Subject = Any Value &Manager = HospitalManager, Any Action, PatientA_MedicalRecords&PatientB_MedicalRecords) Deny

(Subject = Any Value & Manager = HospitalManager, View, PatientA_PersonalInfo&PatientB_PersonalInfo, Time= 08:00:00 am to 05:00:00 pm) →Permit

(Subject = Any Value & Manager = HospitalManager, Any Action, PatientA_PrivateNotes&PatientB_PrivateNotes) →Deny

- **PatientPolicy**

(Subject = Any Value & Patient = PatientA, View, PatienA_MedicalRecords, Any Value) →Permit

(Subject = Any Value &Patient = PatientA, Add, PatientA_MedicalRecords) →Deny

(Subject = Any Value & Patient = PatientA, View&Add, PatientA_PersonalInfo, Any Value) →Permit

(Subject = Any Value & Patient = PatientA, Any Action, PatientA_PrivateNotes, Any Value) →Deny

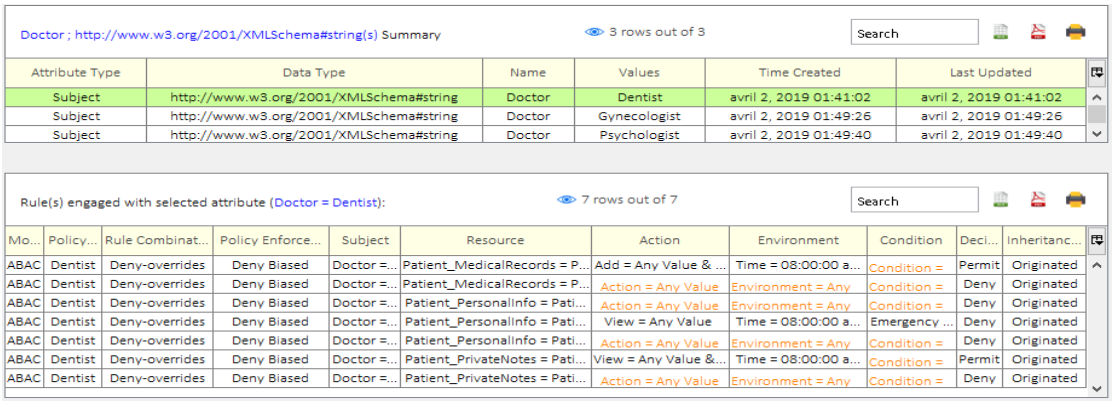
(Subject = Any Value & Patient = PatientB, View, PatientB_MedicalRecords, Any Value) →Permit

(Subject = Any Value &Patient = PatientB, Add, PatientB_MedicalRecords) →Deny

(Subject = Any Value & Patient = PatientB, View&Add, PatientB_PersonalInfo, Any Value) →Permit

(Subject = Any Value & Patient = PatientB, Any Action, PatientB_PrivateNotes, Any Value) →Deny

The modelling of the Doctor Policy, Manager Policy, and Patient Policy after entering their rules is depicted in Figure 33, Figure 34, and Figure 35.



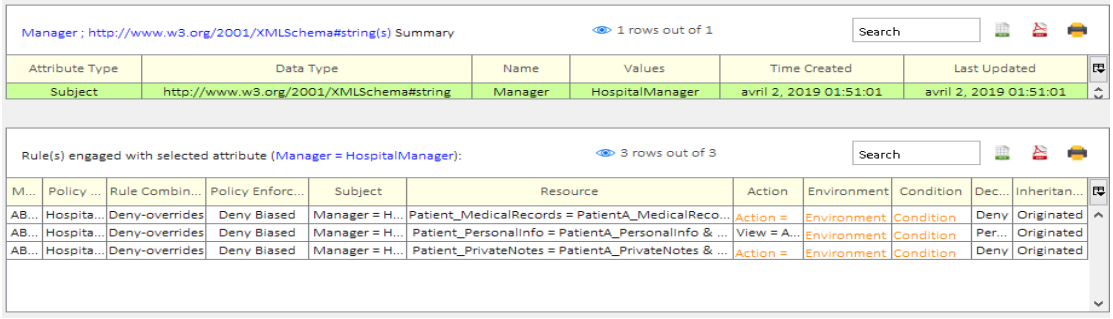
Doctor ; http://www.w3.org/2001/XMLSchema#string(s) Summary — 3 rows out of 3 — Search

Attribute Type	Data Type	Name	Values	Time Created	Last Updated
Subject	http://www.w3.org/2001/XMLSchema#string	Doctor	Dentist	avril 2, 2019 01:41:02	avril 2, 2019 01:41:02
Subject	http://www.w3.org/2001/XMLSchema#string	Doctor	Gynecologist	avril 2, 2019 01:49:26	avril 2, 2019 01:49:26
Subject	http://www.w3.org/2001/XMLSchema#string	Doctor	Psychologist	avril 2, 2019 01:49:40	avril 2, 2019 01:49:40

Rule(s) engaged with selected attribute (Doctor = Dentist): — 7 rows out of 7 — Search

Mo...	Policy...	Rule Combinat...	Policy Enforce...	Subject	Resource	Action	Environment	Condition	Deci...	Inheritanc...
ABAC	Dentist	Deny-overrides	Deny Biased	Doctor =...	Patient_MedicalRecords = P...	Add = Any Value & ...	Time = 08:00:00 a...	Condition =	Permit	Originated
ABAC	Dentist	Deny-overrides	Deny Biased	Doctor =...	Patient_MedicalRecords = P...	Action = Any Value	Environment = Any	Condition =	Deny	Originated
ABAC	Dentist	Deny-overrides	Deny Biased	Doctor =...	Patient_PersonalInfo = Pati...	Action = Any Value	Environment = Any	Condition =	Deny	Originated
ABAC	Dentist	Deny-overrides	Deny Biased	Doctor =...	Patient_PersonalInfo = Pati...	View = Any Value	Time = 08:00:00 a...	Emergency ...	Deny	Originated
ABAC	Dentist	Deny-overrides	Deny Biased	Doctor =...	Patient_PersonalInfo = Pati...	Action = Any Value	Environment = Any	Condition =	Deny	Originated
ABAC	Dentist	Deny-overrides	Deny Biased	Doctor =...	Patient_PrivateNotes = Pati...	View = Any Value &...	Time = 08:00:00 a...	Condition =	Permit	Originated
ABAC	Dentist	Deny-overrides	Deny Biased	Doctor =...	Patient_PrivateNotes = Pati...	Action = Any Value	Environment = Any	Condition =	Deny	Originated

Figure 33: DoctorPolicy



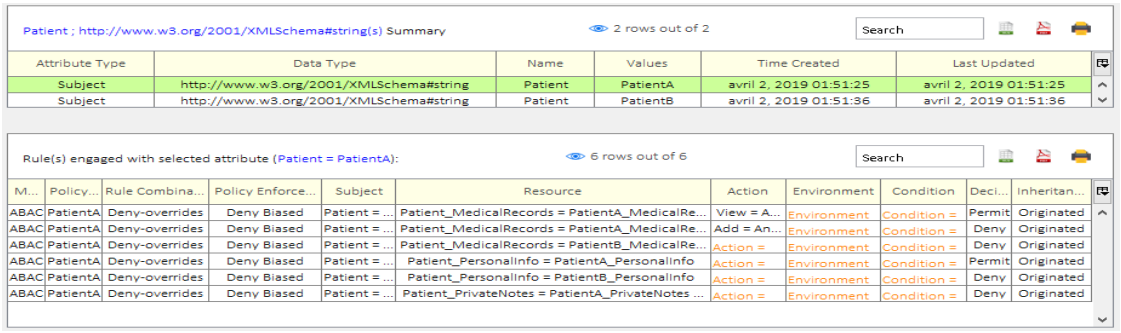
Manager ; http://www.w3.org/2001/XMLSchema#string(s) Summary — 1 rows out of 1 — Search

Attribute Type	Data Type	Name	Values	Time Created	Last Updated
Subject	http://www.w3.org/2001/XMLSchema#string	Manager	HospitalManager	avril 2, 2019 01:51:01	avril 2, 2019 01:51:01

Rule(s) engaged with selected attribute (Manager = HospitalManager): — 3 rows out of 3 — Search

M...	Policy ...	Rule Combin...	Policy Enforc...	Subject	Resource	Action	Environment	Condition	Dec...	Inheritan...
AB...	Hospita...	Deny-overrides	Deny Biased	Manager = H...	Patient_MedicalRecords = PatientA_MedicalReco...	Action =	Environment	Condition	Deny	Originated
AB...	Hospita...	Deny-overrides	Deny Biased	Manager = H...	Patient_PersonalInfo = PatientA_PersonalInfo & ...	View = A...	Environment	Condition	Per...	Originated
AB...	Hospita...	Deny-overrides	Deny Biased	Manager = H...	Patient_PrivateNotes = PatientA_PrivateNotes & ...	Action =	Environment	Condition	Deny	Originated

Figure 34: ManagerPolicy



Patient ; http://www.w3.org/2001/XMLSchema#string(s) Summary — 2 rows out of 2 — Search

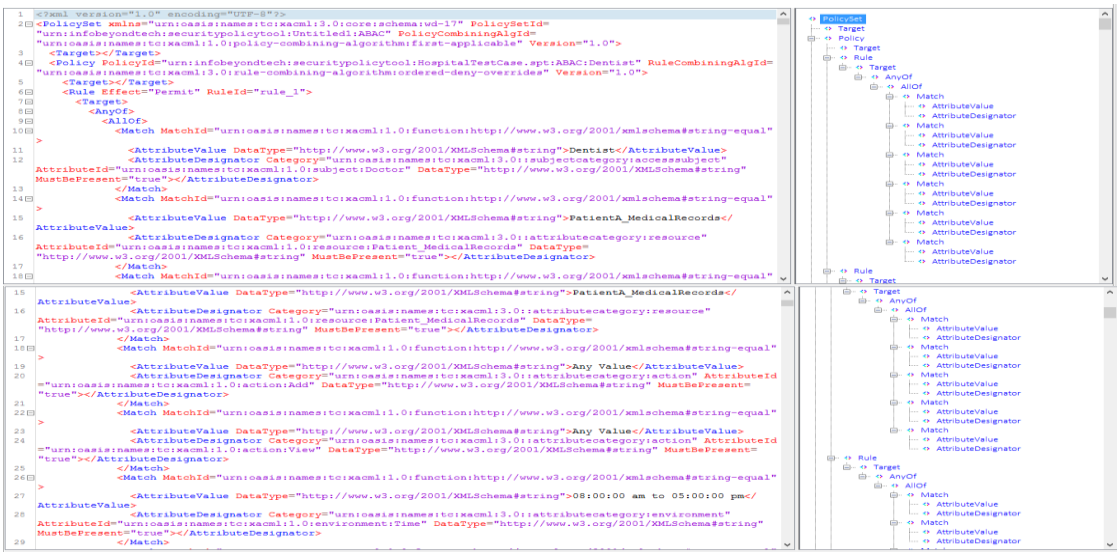
Attribute Type	Data Type	Name	Values	Time Created	Last Updated
Subject	http://www.w3.org/2001/XMLSchema#string	Patient	PatientA	avril 2, 2019 01:51:25	avril 2, 2019 01:51:25
Subject	http://www.w3.org/2001/XMLSchema#string	Patient	PatientB	avril 2, 2019 01:51:36	avril 2, 2019 01:51:36

Rule(s) engaged with selected attribute (Patient = PatientA): — 6 rows out of 6 — Search

M...	Policy...	Rule Combina...	Policy Enforce...	Subject	Resource	Action	Environment	Condition	Deci...	Inheritan...
ABAC	PatientA	Deny-overrides	Deny Biased	Patient = ...	Patient_MedicalRecords = PatientA_MedicalRe...	View = A...	Environment	Condition =	Permit	Originated
ABAC	PatientA	Deny-overrides	Deny Biased	Patient = ...	Patient_MedicalRecords = PatientA_MedicalRe...	Add = An...	Environment	Condition =	Deny	Originated
ABAC	PatientA	Deny-overrides	Deny Biased	Patient = ...	Patient_MedicalRecords = PatientB_MedicalRe...	Action =	Environment	Condition =	Deny	Originated
ABAC	PatientA	Deny-overrides	Deny Biased	Patient = ...	Patient_PersonalInfo = PatientA_PersonalInfo	Action =	Environment	Condition =	Permit	Originated
ABAC	PatientA	Deny-overrides	Deny Biased	Patient = ...	Patient_PersonalInfo = PatientB_PersonalInfo	Action =	Environment	Condition =	Deny	Originated
ABAC	PatientA	Deny-overrides	Deny Biased	Patient = ...	Patient_PrivateNotes = PatientA_PrivateNotes ...	Action =	Environment	Condition =	Deny	Originated

Figure 35: PatientPolicy

Due to the difficulty of using XACML, we rely on the Security Policy Tool Editor to implement the proposed policy rules. Subsequently, we use this tool to convert these policies into XACML language. As an illustration, Figure 36 presents XACML policy for Doctor with Dentist attribute.



```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17" PolicySetId=
"urn:infobeyondtech:securitypolicytool:Untitled1:ABAC" PolicyCombiningAlgId=
"urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:first-applicable" Version="1.0">
  <Target></Target>
  <Policy PolicyId="urn:infobeyondtech:securitypolicytool:HospitalTestCase.apt:ABAC:Dentist" RuleCombiningAlgId=
"urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:ordered-deny-overrides" Version="1.0">
    <Target></Target>
    <Rule Effect="Permit" RuleId="rule_1">
      <Target>
        <AnyOf>
          <AllOf>
            <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:http://www.w3.org/2001/xmlschema#string-equal"
>
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Dentist</AttributeValue>
              <AttributeDesignator Category="urn:oasis:names:tc:xacml:3.0::subjectcategory:accesssubject"
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:Doctor" DataType="http://www.w3.org/2001/XMLSchema#string"
MustBePresent="true"></AttributeDesignator>
            </Match>
            <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:http://www.w3.org/2001/xmlschema#string-equal"
>
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">PatientA_MedicalRecords</
AttributeValue>
              <AttributeDesignator Category="urn:oasis:names:tc:xacml:3.0::attributecategory:resource"
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Patient_MedicalRecords" DataType=
"http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"></AttributeDesignator>
            </Match>
            <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:http://www.w3.org/2001/xmlschema#string-equal"
```

Figure 36: Policy rules converted on XACML language

4.4 Creating individual security requirement

This phase is meant for testing the rules. So, it is necessary to create an individual security policy for this purpose. The Figure 37 below is shown the security requirement of the Manager Policy after entering the following rule.

(Manager=HospitalManager)&(Action=AnyValue)&(Environment=AnyValue)&(Patient_PrivateNotes=PatientA_PrivateNotes&PatinetB_PivateNotes) →decision=Deny

(Manager=HospitalManager)&(Action=View)&(Environment=AnyValue)&(Condition=Emergency)&(Patient_MedicalRecords=PatientA_MedicalRecords&PatinetB_MedicalRecords) →decision=Permit

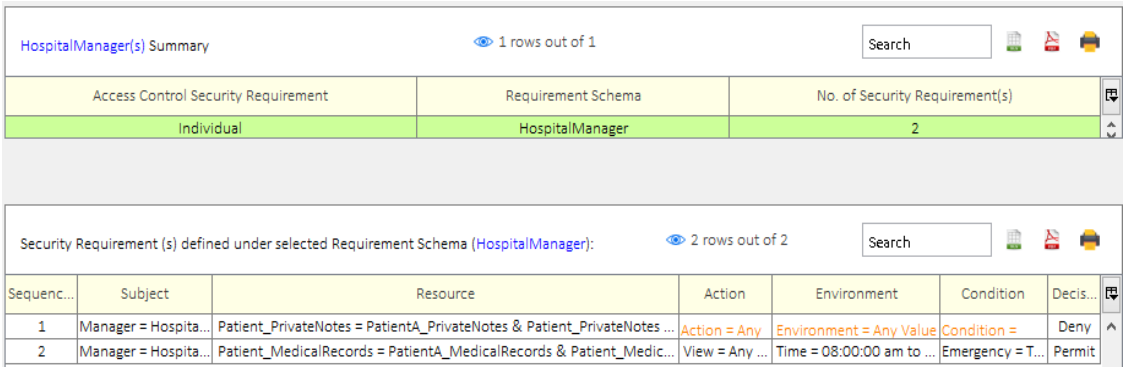


Figure 37: Security requirement for HospitalManager

4.5 Verification of the policies

In this phase the SPT can test one or multiple policies against specific security requirements. A security requirement represents an access request (e.g., a HospitalManager requests any action to a Patient_MedicalRecords at any time and in case of emergency) as shown in the Figure 38. For instance, we evaluate access control decision for a Manager who tries to get access Patient_MedicalRecords.

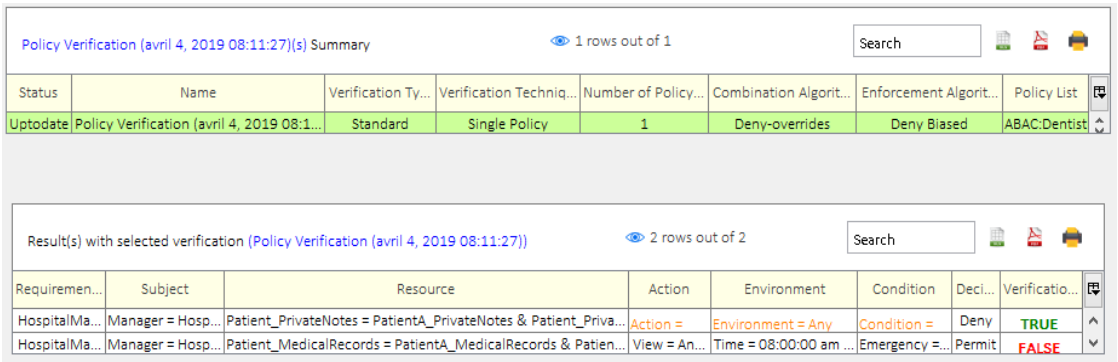


Figure 38: Verification of HospitalManager within Manager Policy

5. Summary

The integration of Cloud and IoT has emerged as an innovative solution to address the scarcity of IoT resources. In fact, Cloud provides large scale storage systems and powerful tools. However, despite the benefits of this model, it still faces several issues regarding security and privacy. We suggested an efficient XACML basic access control to protect data collected from IoT devices. We use SPT editor to easily design security policies. The simulation results prove that our proposed solution can prevent unauthorized access to confidential data. To illustrate the importance of the proposed methodology, we designed and implemented a policy that ensures the management of the hospital in a secure environment.

CHAPTER 4 : Efficient ABAC Rules Evaluation through Fuzzy C-Means Algorithm in Cloud-Enabled IoT

The current chapter presented a novel approach based on a fuzzy c-means algorithm to detect and resolve redundancy and conflicts in ABAC policy. This chapter is structured as follows. Firstly, we discuss the motivation and scope behind the current chapter (section 1); and secondly, basic concepts of the used methods in our proposed approach (section 2). Then, we introduce details around the problem statement (section 3). The proposed approach is outlined in section (section 4). Finally, we end the chapter with some concluding remarks (section 5).

1. Motivation and Scope

Distributed environments such as cloud computing and the Internet of Things (IoT) have played a critical role in data storage and management. Granted, they have increased the space for storage of data. They have also decreased the chances of loss of data, especially where there happens to be a breakdown of physical computer components. Thus, the ease of retrieval of data stored in such systems is easy. However, distributed environments have also created a major challenge in Access Control. The owner of information becomes less aware of other parties who might gain access and how they could manipulate it to their advantage. In mitigation, different access control mechanisms have been suggested in line with easing the challenge of Access Control. The Attributes-Based Access Control (ABAC) model is the most appropriate one in an open distributed computing environment because of its flexibility and scalability. ABAC allows attributes of subject, object, and environment to take

part in authorization decision. These attributes are used by a policy to describe specific access rules. Moreover, the Authorization rules can be defined to control access to sensitive and private data stored in cloud computing. Due to the huge number of users and resources the ABAC policy contains a large number of rules which leads to an explosion of these rules. Additionally, ABAC policy rules are often complex making them prone to conflicts. Consequently, the idea of an optimized approach to reduce the rules space and resolving the rules conflict using Fuzzy c-means algorithm (FCM) is wide and varied in the application of cyber security.

The remainder of this chapter is organized as follows. A detailed literature review about the basic concepts is given in section 5.2. Section 5.3 is devoted to the problem statement and the definition of the methods used in our proposed approach. Our suggested solution and discussion are presented in section 5.4. Finally, a conclusion is provided in section 5.5.

2. Basic Concepts

ABAC is a general access control model where decision making is based on subject attributes (e.g., user, application...), object attributes (e.g., data structure, web service…), and environment attributes (e.g., time, current location…). It is more expressive and fine-grained, because it might consider any combination of subject, object and environment attributes. These attributes are evaluated against the rules associated to each object to grant or deny access [116]. Broadly speaking, XACML is a very good way to implement authorization policies in ABAC model.

2.1 Algebraic Approach

In most cases, the Fine-Grained Integration Algebra (FIA) is used to describe XACML policies [117][118][119]. It is an algebra that can be used to handle the merging of two or more policies.

a) Basic Definitions

A policy can be considered as a function mapping each request to a value in {Y,N,NA}, where:

- Y denotes the set of attributes that evaluate to Permit
- N denotes the set of attributes that evaluate to Deny
- NA denotes the set of attributes that evaluate to Not Applicable

And $\boldsymbol{R_Y^P}$, $\boldsymbol{R_N^P}$, $\boldsymbol{R_{NA}^P}$ denote the sets of requests permitted, denied and not applicable by the policy P respectively.

- $\boldsymbol{R_\Sigma = R_Y^P \cup R_N^P \cup R_{NA}^P}$
- $\boldsymbol{\emptyset = R_Y^P \cap R_N^P}$
- $\boldsymbol{\emptyset = R_Y^P \cap R_{NA}^P}$
- $\boldsymbol{\emptyset = R_N^P \cap R_{NA}^P}$

Or, Σ denotes the vocabulary containing all possible attribute names and their corresponding domain and R_Σ is the set of all possible requests.

- Policy P is defined as the triple $\{ \mathbf{R_Y^P}, \mathbf{R_N^P}, \mathbf{R_{NA}^P} \}$.
- R_{NA} can always be mentioned as $R_{NA} = R_\Sigma \setminus (R_Y \cup R_N)$

b) Basic Operators

- **Constants**

- P_Y is a policy constant that permits everything. Thus $P_Y \equiv <R_\Sigma; \emptyset; \emptyset >$
- P_N is a policy constant that denies everything. Thus $P_N \equiv <\emptyset; R_\Sigma; \emptyset >$

- **Addition (+):** It combines two policies into a single one:

- A Permit Policy: $P_Y = <R_\Sigma; \emptyset; \emptyset >$, this policy permits everything.
- A Deny Policy: $P_N = <\emptyset; R_\Sigma; \emptyset >$, this policy denies everything.

$$P_3 = P_1 + P_2 \Leftrightarrow \begin{cases} \boldsymbol{R_Y^{P3}} = \boldsymbol{R_Y^{P1}} \cup \boldsymbol{R_Y^{P2}} \\ \boldsymbol{R_N^{P3}} = \left(\boldsymbol{R_N^{P1}} \setminus \boldsymbol{R_Y^{P2}}\right) \cup \left(\boldsymbol{R_N^{P2}} \setminus \boldsymbol{R_Y^{P1}}\right) \end{cases}$$

That means P_3 is permitted for all requests that are permitted by either P_1 or P_2, requests that are denied by one policy and are not permitted by the other are denied.

- **Intersection (&):** Given two policies P_1 and P_2, the intersection operator returns a policy P_3 which is applicable to all requests having the same decisions from P_1 and P_2. More precisely:

$$P_3 = P_1 \,\&\, P_2 \Leftrightarrow \begin{cases} \boldsymbol{R_Y^{P3}} = \boldsymbol{R_Y^{P1}} \cap \boldsymbol{R_Y^{P2}} \\ \boldsymbol{R_N^{P3}} = \boldsymbol{R_N^{P1}} \cap \boldsymbol{R_N^{P2}} \end{cases}$$

The integrated policy makes a decision only when the two policies agree. The intersection operator is useful in situations in which consensus is required for making a permit or deny decision.

The addition and intersection are two binary operators. They can be viewed as a function that maps a pair of values {Y,N,NA} to one value. Table 5 introduces this view of addition and intersection operators. Binary operator is represented using a matrix that illustrates the effect of integration for a given request r. The first column of each matrix denotes the effect of P_1 with respect to r and the first row denotes the effect of P_2 with respect to r.

Table 5: Policy combination matrix of operators +, &

$P_1 + P_2$

P_2 \ P_1	Y	N	NA
Y	Y	Y	Y
N	Y	N	N
NA	Y	N	NA

P1 & P2

P_2 \ P_1	Y	N	NA
Y	Y	NA	NA
N	NA	N	NA
NA	NA	NA	NA

- **Negation (¬):** Given a policy P_1, $\neg P_1$ returns a policy P_2, the negation inverts all decisions of a policy:

$$P_2 = \neg P_1 \Leftrightarrow \begin{cases} \boldsymbol{R_Y^{P2} = R_N^{P1}} \\ \boldsymbol{R_N^{P2} = R_Y^{P1}} \end{cases}$$

The negation operator does not affect those requests that are not applicable to the policy.

- **Domain Projection (Π_{dc}):** The Domain Projection (Π_{dc}) restricts a policy to the set of requests defined by a specific domain. This restriction is named domain constraint (dc). A domain constraint dc takes the following form:

$\{(a_1, range_1), (a_2, range_2),..., (a_n, range_n)\}$[1], where $a_1, a_2, ..., a_k$ are attribute names, and rangei($1 \leq i \leq k$) are sets of values from the vocabulary Σ. Given a request $r = \{(a_{r1}, v_{r1}), \cdots, (a_{rm}, v_{rm})\}$. We say r satisfies dc if the following condition holds: for each $(a_{rj}, v_{rj}) \in r$ ($1 \leq j \leq m$), if there exists $(a_{rj}, range_i) \in dc$, then $v_{rj} \in range_i$.

The semantics of $\Pi_{dc}(P)$ is given by:

$$P_1 = \Pi_{dc}(P_2) \Leftrightarrow \begin{cases} \boldsymbol{R_Y^{P1} = \{ r | r \in R_Y^{P2} \text{ and } r \text{ satisfies dc} \}} \\ \boldsymbol{R_N^{P1} = \{ r | r \in R_N^{P2} \text{ and } r \text{ satisfies dc} \}} \end{cases}$$

2.2 Geometric Approach

In general, ABAC policy contains a huge number of rules and the policies distributed management. Accordingly, it is too complex and hard to deploy and manage an ABAC model for an efficient access control solution. Therefore, ABAC policies may contain several anomalies [120][121], such as conflicts and redundancies.

a) Formal Definitions of the Considered Anomalies

An anomaly is defined as a situation that does not conform to a well-defined notion of normal behaviour [122] [123]. More specifically, in a security policy P, an anomaly may exist only if several rules of P intercept a same query to give access decisions (permit, deny). In our contribution, we have considered two types of anomalies. Remove the Redundancies in order to reduce rules space and Conflicts [124][125].

Let consider a unique set of values (Vatt1; Vatt2;…; Vattn) for the n-tuple (att1; att2;…; attn) where (Vatt1 ; Vatt2;…; Vattn) is called the access domain of the rule ri and it noted ADri. Therefore ri can be expressed in the form ri = Xact (ADri).

Or Xact is an access decision of a rule, where X is the decision (Permit or Deny), and act is a set of access actions (e.g read, write, delete …).

- **Definition 1: Redundancy**

Redundancy occurs in a policy P, when P contains useless (or redundant) rule. For example, we say that two rules are redundant if removing one of them does not change the behaviour of the policy P.

Consider two rules ri = Xa(ADri) and rj = Yb(ADrj). ri and rj are redundant iff :

$$\begin{cases} \mathbf{ADri} \subseteq \mathbf{ADrj} \\ \mathbf{X} = \mathbf{Y} \\ \mathbf{a} \subseteq \mathbf{b} \end{cases}$$

This means that any decision taken by ri on any request is directly taken by r_j. Therefore, r_i is useless and hence can be removed from the policy. As redundancy may affect the performance of a policy as well as slow down the system, because verifying if an access request respects a policy depends on the size of the policy. For this reason, we consider redundancy as anomaly [121].

- **Definition 2: Conflict**

A conflict occurs in a policy P, when P contains two or more rules that can match the same profile with different effect (Permit, Deny) for the same request.
Consider two rules ri = Xa(ADri) and rj = Yb(ADrj). ri and rj are conflicting iff :

$$\begin{cases} \mathbf{ADri} \cap \mathbf{ADrj} \neq \emptyset \\ \mathbf{X} \neq \mathbf{X} \\ \mathbf{a} \cap \mathbf{b} \neq \emptyset \end{cases}$$

This means when an access request matches the common access domain of r_i and r_j ($AD_{ri} \cap AD_{rj}$), we have contradictory decisions (from $X \neq Y$) on common actions (from $a \cap b \neq \emptyset$).

2.3 Formalization of ABAC Policy

The ABAC model defines permissions based on predefined attributes. These attributes consist of any characteristics that should be taken into account for

the authorization decisions. The attributes are associated to three different entities: Subject (i.e. the user or the process that takes action on a resource), Resource (i.e. the entity that is acted upon by a subject) and Environment (i.e. the operational, technical and the context in which the information access occurs) [126]. Therefore, the attributes are Subject attributes, Resource attributes and Environment attributes. Basic ABAC scenario is presented in Figure 39 below [127].

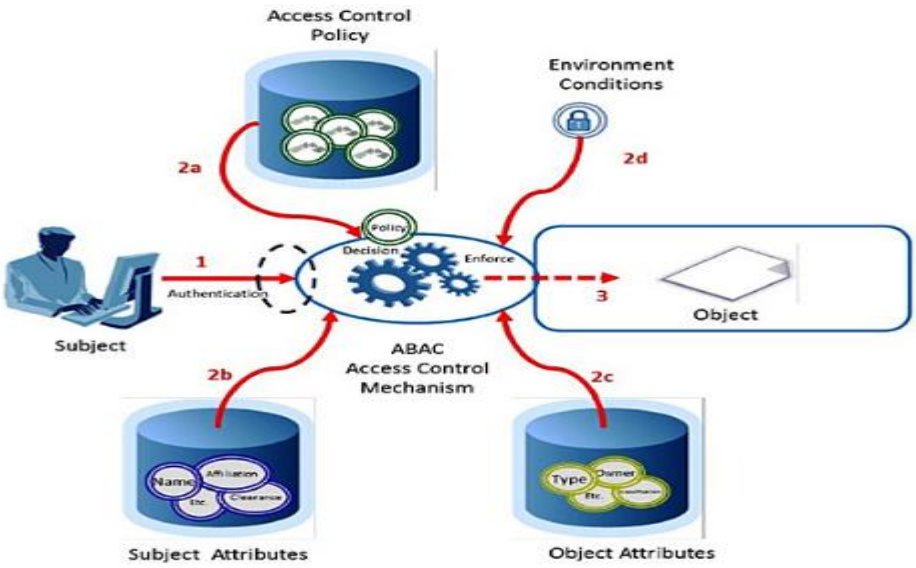


Figure 39: Basic ABAC scenario [127]

The formalization of ABAC model is as follows:

- **Attributes definition**
  - ✓ S: Set of Subjects
  - ✓ O: Set of Objects
  - ✓ E: Set of Environments
  - ✓ SA_n is a subject attribute with $1 \leq n \leq N$
  - ✓ OA_m is an object attribute with $1 \leq m \leq M$
  - ✓ EA_k is an environment attribute with $1 \leq k \leq K$

The attributes assignment relationship of the entities subject, object, environment are attr(s), attr(o), attr(e) respectively, or:

$attr(s) \subseteq SA_1 \times SA_2 \times \dots SA_N$

$attr(o) \subseteq OA_1 \times OA_2 \times \dots OA_M$

$attr(e) \subseteq EA_1 \times EA_2 \times \dots EA_K$

- **Policy and Rule definition**

The ABAC policy is a set of rules which deny or permit a subject **s** to access an object **o** in an environment **e**. It can formalize as follows: $P= \{R_1, R_2, \dots R_n\}$ with $R_1, R_2, \dots R_n$ are the rules. The policy rule is a set of attributes that can be used to evaluate the access request and then make a decision. The rule is presented as follows:

$R\{ attr(s) \subseteq SA_1 \times SA_2 \times \dots SA_N, attr(o) \subseteq OA_1 \times OA_2 \times \dots OA_M , attr(e) \subseteq EA_1 \times EA_2 \times \dots EA_K, action\} \rightarrow$Deny or Permit.

- ✓ Rule example:

R{Job ⊆ (Nurse, Doctor, Manager), File ⊆ (Medical-Records, Personal-Info, Documentation), Time ⊆ (8:00 am – 4:00 pm, 6:00 pm – 12:00 am), Action ⊆ (read, write, delete)}→Deny or Permit

3. Problem Statement

Access control mechanisms are destined to allow activities of authorized users to access resources in a given system. In our pervious chapters and in the current chapter as well, we consider ABAC as an efficient access control model within a distributed system such as CEIoT in terms of access control granularity, flexibility and decision efficiency. However, it still has some

shortcomings, as in large and distributed environment with complex ABAC policies, deploying and managing an ABAC model to ensure authorization management might become too complex and hard to manage [128]. An ABAC policy in distributed applications may be aggregated from multiple parties and can be managed by more than one administrator. Therefore, it may contain several anomalies such as redundancies and conflicts (see definitions 1 and 2 in subsection 5.2.2.1) which lead to complexity and availability issues. Since redundancies are here users also experience a problem in having a proper understanding of all the rules as they need to spend a great deal of time to capture insights on their meanings as well as where they can be applied. . Another challenge is that its policies are normally complex and are, therefore, prone to conflicts. Where conflicts arise, there is normally a challenge in determining the specific rules that would need to be followed in a given context [129][130]. Hence, detecting and resolving such anomalies is essential to ensure that an ABAC policy conforms to desired correctness properties [131]. The proposed approach aims to solve two problems which are: rules redundancy and rules conflicts.

3.1 Conflict Detection and Resolution

Anomaly detection and resolution is motivated by the fact that errors in the policy definition may compromise the system security. Conflicts, if not handled properly, may lead to inappropriate decisions. As a result, conflicts may lead to safety problems by allowing unauthorized accesses, and availability problems by denying authorized accesses [132]. As for redundancies, their detection and resolution is motivated by the fact that they affect the performance of the policy execution, since the response time of a policy to an access request depends on the number of rules to be parsed in the policy [133]. Figure 40 depicts the anomalies detection and resolution in ABAC policy.

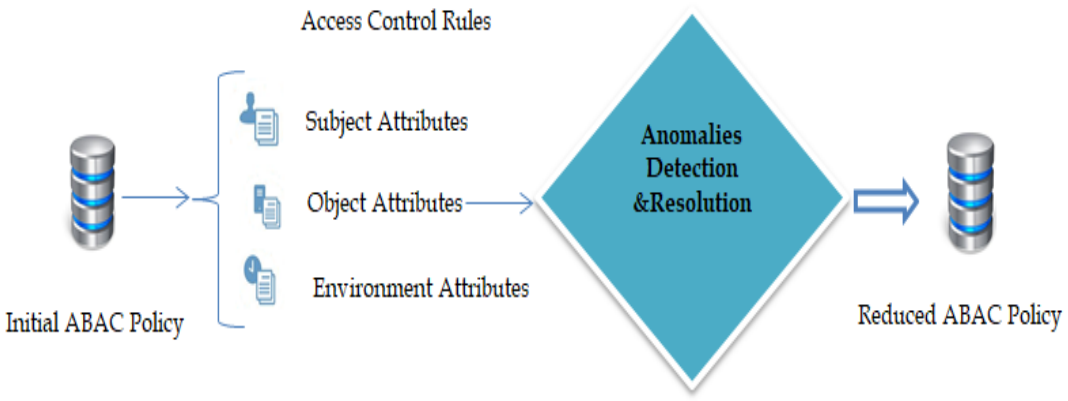


Figure 40: Anomalies detection and resolution in ABAC policy

a) Rules Redundancy

A rule is a set of attributes taken into account in decision making to access a resource. Within a dynamic and distributed environment like CEIoT, the number of users is increased which implicitly increases the number of attributes and thereafter the number of rules. To reduce rules space we propose removing redundant attributes among these rules [134]. Therefore, we use a function to calculate the membership degree of each attribute and then we remove the attributes that have a membership degree less than a given threshold.

b) Rules Conflict

A policy contains multiple rules each of which can lead to different access control decisions. Therefore, a conflict occurs in a policy when one rule permits a request and another denies that same request. In this regard, two rules may contain common attributes with different access decisions (Permit, Deny). Such errors can cause serious risks involving both access to an unauthorized user and denial of access to a person who needs it [135].

Let consider R_1 and R_2 two rules with the following description:

R_1 {Job $\subseteq$ (Nurse, Doctor, Manager), File $\subseteq$ (Medical-Records, Personal-Info, Documentation),

Time ⊆ (8:00 am – 4:00 pm, 6:00 pm – 12:00 am), Action ⊆ (read, write)}→Permit

R_2 {Job ⊆ (Nurse, Doctor), File ⊆ (Medical-Records), Time ⊆ (8:00 am – 4:00 pm), Action ⊆ (read)}→ Deny

For example, the rule R_1 permits a Nurse to read and write in Medical-Records at 8:00 am - 4:00 pm and from 6:00 pm to12:00 pm. However, the rule R_2 denies the same subject (e.g. Nurse) to read in Medical-Records at 8:00 am - 4:00 pm. If the request is as follows: a Nurse requesting access to Medical-Records at 8:00 am - 4:00 pm in order to read it. In this situation the two rules R_1 and R_2 are match to the request, but their decisions are different. This is called rules conflict.

3.2 An Overview of Used Methods

As outlined above we opt for ABAC to ensure access control in CEIoT environment. Despite the benefits of this solution, to implement a large policy would lead to several technical challenges such as rule conflict and redundancy. In this respect, we suggest a novel approach to address the issue of computation costs in policy evaluation. In the following an overview of the used methods in our proposed solution.

a) Clustering

Clustering is a mechanism that allows regrouping objects into a cluster depending on the relationship or similarity between them. The goal of this technique is to get similar objects within the same cluster. In addition, the objects in a cluster are similar between them and dissimilar to objects in other cluster [136]. Basically, there are two types of clustering hard and soft clustering. In the first type, one piece of data is affected exactly to one cluster. In contrast, the second category uses fuzzy sets theory to affect one piece of data to more than one cluster with a probability between 0 and 1, as shown in Figure 41 [137][138].

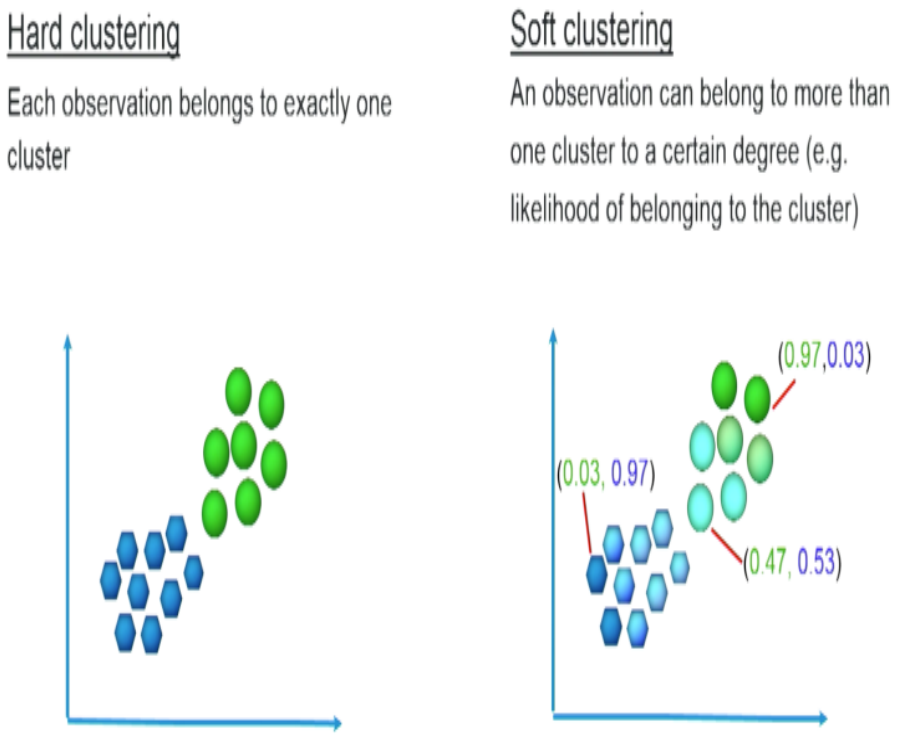


Figure 41: Hard clustering vs soft clustering[138]

These types of clustering belong to different theories. The hard clustering belongs to the classical set theory and the soft clustering belongs to fuzzy set theory. In the following there is a comparison between these two theories.

b) Fuzzy Set Theory vs Classical Set Theory

In real world, there exists much fuzzy information and data (imprecise data, uncertain data, inexact or probabilistic data). However, systems based upon classical set theory are unable to answer many requests. They are incapable to cope with unreliable and incomplete information. Fuzzy set theory is an extension of classical set theory where elements have degree of membership. It can be described as a set for which elements do not have the simple property of either being in the set or out of it [139]. An element can be partially in the set also. So, the membership function in the case of fuzzy is logic is not a simple set consisting of 1 and 0, it represents the degree of truth.

The Figure 42 below represents a diagram comparing classical set and fuzzy set [140].

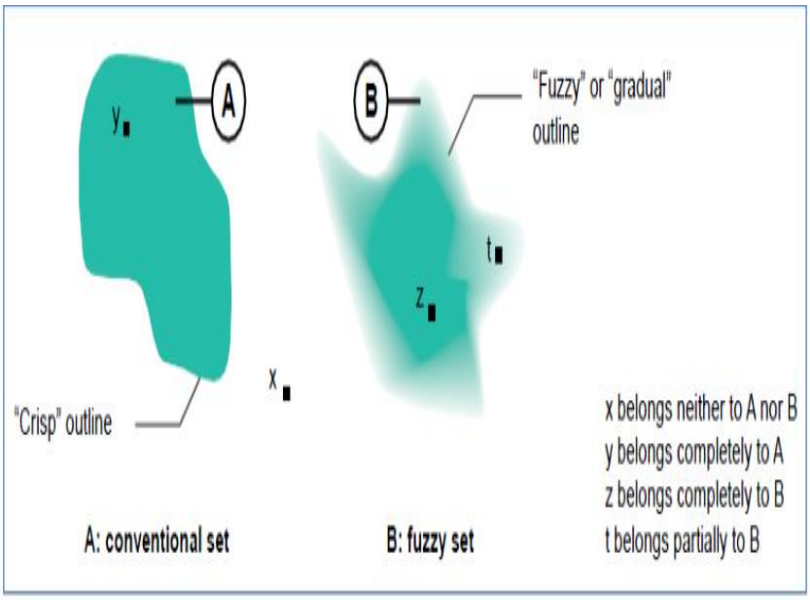


Figure 42: Diagram comparing classical set and fuzzy set [141]

In classical logic, the arguments are of the form:

$$\begin{cases} if\ p\ then\ q \\ p\ true\ then\ q\ true \end{cases}$$

In fuzzy logic, fuzzy reasoning is based on fuzzy rules that are expressed in natural language using linguistic variables. All the rules of a fuzzy system are called the decision matrix. A fuzzy rule has the form: If $x \in A$ and $y \in B$ then $z \in C$, with A, B and C fuzzy sets. The notion of belonging to a set is very important in set theory, it mentions either an element is part of a set or not.

- **Membership Function**

Let consider X a set and A is a fuzzy subset of X and μA the membership function characterizing A. $\mu A: X \rightarrow [0, 1]$. μA(x) is called the membership degree of x in A.

Let $X=\{x_1,x_2,\ldots x_n\} \subseteq IR_n$ is a data set and c is the number of clusters or $2 \leq c < n$, μ_{ij} is the degree of membership of x_i in the j^{th} cluster.

In practice, k-means algorithm is widely used to perform hard clustering while Fuzzy c-means is used in soft clustering.

c) K-means Algorithm

K-means (KM) algorithm is one of the simplest and popular unsupervised machine learning algorithms. The goal of this algorithm is to classify a given data set through a certain number of disjoint clusters (assume k clusters) fixed a priori. Its main idea is to define k center, one for each cluster and then take each point belonging to a given data set and associate it to the nearest center. K-Means is reported fast, robust and simple to understand and implement, it also gives comparatively good results if clusters in datasets are distinct or well separated [142]. Algorithm 1 presents the pseudo code of k-means algorithm.

Algorithm 1: k-means

Input: $X = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^D$ (N×D input data set)

Output: $C=\{c_1, c_2, \ldots, c_k\} \in \mathbb{R}^D$ (K cluster centers)

Select a random subset C of X as the initial set of cluster centers

While termination criterion is not met do

  For (i = 1; i ≤ N; i = i +1) do

    Assign xi to the nearest cluster;

      $m[i] = \underset{k\in\{1,2,\ldots,K\}}{\operatorname{argmin}} ||\, x_i - c_k ||^2$

  End

  Recalculate the cluster centers;

    For (k = 1; k ≤ K; k = k + 1) do

      Cluster S_k contains the set of points x_i that are nearest to the center c_k;

        $Sk = \{xi \mid m[i] = k\}$;

      Calculate the new center c_k as the mean of the points that belong to S_k ;

        $\mathbf{c_k} = \frac{1}{|\mathbf{S_k}|} \sum_{\mathbf{x_i} \in \mathbf{S_k}} \mathbf{x_i}$

    End

End

Return $< c_1, c_2, \ldots, c_k >$

d) Fuzzy c-means Algorithm

Fuzzy c-means (FCM) is an extension of k-means algorithm by combining it with fuzzy logic. It is a method of clustering that normally allows for one piece of data to belong to more than one cluster with a degree of membership between 0 and 1. As in fuzzy logic, rather than belonging completely to just one cluster. It is introduced by Lotfi Zadeh in 1965 as a way of dealing with imprecision and uncertainty [143][144]. The Fuzzy c-means algorithm is an iterative algorithm in which the desired number of clusters c and the initial clustering seeds has to be pre-defined. In the algorithm, when the objects are assigned to clusters they are also assigned a membership function that indicates the membership degree to that cluster, and thus, compared with the k-means algorithm, FCM algorithm improves partition performance and reveals the classification of objects more reasonable [145]. The parameter m is a fuzziness parameter selection. Its value is greater than 1. ZHOU Kaile et al. proposed an algorithm to calculate the value of m and the median value is m=2 [146][147]. The pseudo code of FCM algorithm is illustrated in Algorithm 2 below.

Algorithm 2: Fuzzy c-means

Input: $U^{(0)}=[\mu_{ij}]$: matrix, m=2
Output: $C^{(k)}$, μ_{ij}
Begin
  While (|| $U^{(k+1)} - U^{(K)}$ || < ε) do
    If Step= k **then**
      //calculate centers vectors $C^{(k)}=[C_j]$ with $U^{(k)}$
      For (i=1;i< =N;i++)
        For (j=1;j<=2;j++)
          c[j]= (μ[i][j]+ μ[i][j] ^{m} * x)/ (μ+ μ ^{m})
        End
      End
    End If
    For (k=1;k< =c;k++)
      For (j=1;j<=2;j++)
        μ[i][j] = 1/ ([||(x[i]-c[j])||/||(x[i]-c[k])||]$^{2/m-1}$
      End
    End
  End
End

The matrix $U^{(0)}$ values are random and they are between 0 and 1. The sum of each row is equal to 1. Let consider the matrix below as example.

$$U^{(0)} = \begin{array}{c} \\ att_1 \\ att_2 \\ \dots \\ att_n \end{array} \begin{array}{c} \begin{array}{cc} C_1 & C_2 \end{array} \\ \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ .. & .. \\ 0.1 & 0.9 \end{bmatrix} \end{array} \quad \text{Or,} \quad U^{(0)} = \begin{array}{c} \\ att_1 \\ att_2 \\ att_3 \\ \dots \\ att_n \end{array} \begin{array}{c} \begin{array}{cc} C_1 & C2 \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ .. & .. \\ 0 & 1 \end{bmatrix} \end{array}$$

In these matrixes the columns represent the clusters C_1 and C_2, the rows represent the data. For example, the value 0.8 is the membership's degree of the data att1 in the cluster $C_1$and the value 0.2 is the membership's degree of the data att1 in the cluster C_2.

e) Comparative study of k-means and Fuzzy c-means

In spite of these advantages, k-means algorithm has some disadvantages. For example, if there are two highly overlapping data then k-means will not be able to resolve that. The algorithm fails for non-linear data set [148]. Moreover, the KM with multiple starts was extremely superior to the FCM in terms of computing time in all datasets analysed comparatively good results if clusters in datasets are distinct or well separated. The KM algorithm is based on hard clustering that means one piece of data can belong to one and only one cluster. To address this issue we suggested the usage of Fuzzy c-means since it is more suitable for a security policy with large number of rules. In fact, FCM offers the possibility to perform a soft clustering by assigning each data to more than one cluster. The Figure 43 below illustrates the main difference between k-means and Fuzzy c-means algorithms [149].

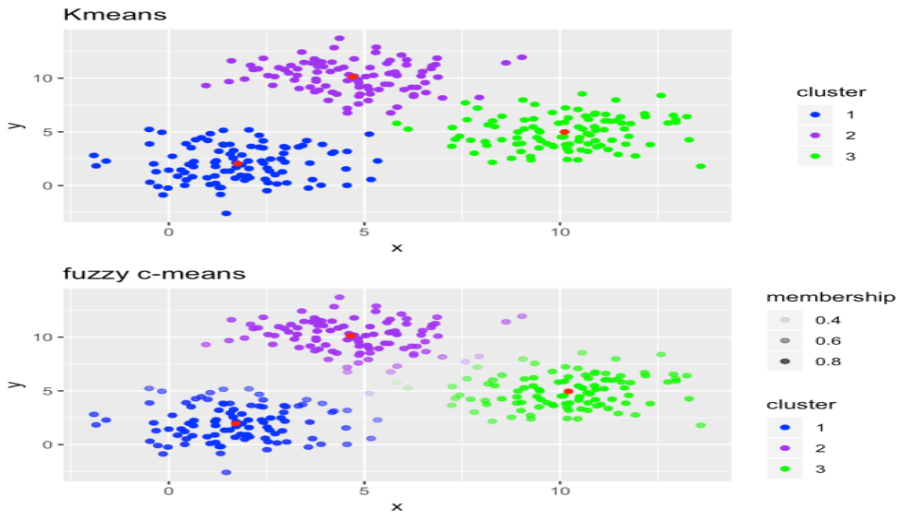


Figure 43: Difference between k-means and c-means algorithms [150]

4. The Proposed Approach

This section provides full details of the proposed solution. The steps of this approach are given below:

- **Step 1: Extraction of Rules Attributes**

We extract attributes from existing rules in order to obtain a data set of features and their values.

- **Step 2: Clustering of Data Set**

We apply FCM algorithm to the obtained data set in the step 1 in order to split it into two clusters C_1 and C_2. C_1 contains attributes appear in the rules that have Permit effect and C_2 contains attributes appear in the rules that have Deny effect. As result of applying FCM on our data set, a degree of membership μ_{ij} is evaluated for each attribute for both clusters C_1 and C_2. Each attribute has two memberships, one in the cluster C_1 and the other in the cluster C_2.

- **Step 3: Reducing Rules Space**

We consider a threshold m for the memberships, as in equation (1).

$$m = \frac{\sum j = 0{,}1 \sum_{i=1}^{n} \mu ij}{2n} \qquad (1)$$

μ_{ij} is the degree of membership of x_i , i=1…n, (x_i : i^{th} attribute in the data set) in the cluster j, where j= {C_1, C_2} and n is the number of attributes.

Consequently, we remove the rules which the sum of their attributes' membership is less than the threshold m.

- **Step 4: Calculation of Rule's Significance**

We calculate the significance of each rule R_i using sum of the attributes' membership in this rule as in formula (2).

$$S(R_i) = \sum_{i=1}^{n} \mu ij \qquad (2)$$

We use this significance to compare the pertinence of each rule. Also, we can use it to reduce rules space by removing the rules which have the significance less than a threshold predefined.

- **Step 5: Conflict Resolution**

When the conflict occurs between two rules, we resolve it with comparing their significance (step 4), we consider the decision of the rule that has the high significance. In the case that the rules have similar significance we use the sum of the attributes' weight $w_i^{(j)}$. This letter is calculated as follow:

We calculate the weight $w_i^{(j)}$ of each attribute in each cluster j using the formula (3).

$$w_i^{(j)} = \frac{\mu ij}{\sum_{k=1}^{n} \mu kj} \qquad (3)$$

μ_{kj} is the membership of the attributes appears in the cluster j. Note that each attribute has two weights, one in the cluster C_1 and the other in the cluster C_2.

The sum of the attributes' weight is calculated as in the formula (4) below.

$$w(R_k^{(j)}) = \sum_{i=1}^{n} w_i^{(j)} \qquad (4)$$

$w_i^{(j)}$ is defined in the equation (3)

k: number of the rules

Therefore, we consider the decision of the rule which has a high $w(R_k^{(j)})$. In the following the pseudo code of the algorithm of our proposed solution.

Algorithm 3: Reduced rules space and resolved rules conflict

```
Input: P= {R1, R2,…,Rn}
Output: reduced rules space, resolved rules conflict
Begin
  //apply fuzzy C-means algorithm to P in such way to
  //get two clusters C1 and C2
  //C1: rules with permit decision
  //C2: rules with deny decision
  //reduce rules space

  For (i=0; i<=n;i++)
      For (j=0; j<=1;j++)
          m += μ[i][j]/2*n
          If μij (Ri)<m then
              P-{Ri}
          End if
          S (R[i])+= μ[i][j]
      If S(R[i]) > S(R[j]) then Decision ←R[i]
      Else if (S R[i]) < S(R[j]) then Decision ← R[j]
      Else
       for (k=1;k<=n;k++)
          S+= μ[k][j]
          w(R[i]) = μ[i][j]/S
            If w(R[i])>w(R[j])) then Decision ←R[i]
                  Else decision ← R[j]
             End If
        End
      End if
    End

  End
End
```

5. Summary

In this chapter, we have proposed an optimized approach that detects and resolves redundancy and conflict in ABAC policy. Since the policy contains several rules, we classified rules into two clusters. Then the proposed approach is applied in each cluster to calculate the membership of each attribute in order to remove the attributes with membership less than a threshold, implicitly removing rules that contain those attributes. The ABAC model is deemed to be the most appropriate in an open computing distributed environment. It is highly flexible and scalable, therefore, it can be applied in various forums to create optimal outcomes. It also increases the ease of making authorization decisions, which is important in terms of aiding the ability of the user to navigate through it. The Fuzzy c-means algorithm seeks to make the application of the ABAC easy and, hence, help to maintain the security of sensitive information while at the same time, reducing the rules explosion and complexity of policies.

Conclusion and Future Work

Conclusion

This thesis proposes frameworks and solutions to promote effective usage of Internet of Things (IoT) in both industries and humans every-day life. Unfortunately, IoT suffers from constraints of devices' size, energy, power, and computational capability. Besides, IoT deployments are characterized by large heterogeneity in terms of adopted technologies and communication protocols, introducing scalability and interoperability issues. In order to address the aforementioned issues, we suggest the integration of IoT and Cloud Computing, called also cloud-enabled IoT (CEIoT), to improve the function of the IoT and offer a unified service delivery platform for the IoT applications. The primary objective of this approach is the design and development of a flexible and distributed IoT system. Such architecture is needed to allow IoT devices to fully utilize the enormous computational power and services provided by the clouds. It should be based on the idea of bringing processing, security and data qualification closer to the actual data sources, in order to face scalability, reliability and performance issues as well as to perform an assessment of the managed information. From the presented scenario, several issues naturally emerge for handling the complexity of this concept. Of course, this high level of heterogeneity, along with the wide scale of IoT systems, is expected to magnify the security threats of the existing technologies. Moreover, traditional security countermeasures cannot be directly applied to CEIoT because of the flexible and dynamic nature of this new paradigm and the high number of interconnected devices raises scalability issues. At the same time, to reach full acceptance by users and companies, it is mandatory to define valid security, privacy and trust models suitable for the cloud-enabled IoT.

From the analysis of the state of the art in the field of the security of cloud computing and internet of things, many questions arose about how to prevent unauthorized access to a typical cloud-enabled IoT system, which comprises heterogeneous technologies and different communication protocols. This

thesis addressed some of these questions. In particular:

- We provided an extensive analytical and comparative study of the access control solutions. Accordingly, it is more effective to adopt ABAC model in which access control decision are performed in a distributed way in order to manage the scalable CEIoT architecture, where not only users, but also "things" could be authorized to interact with the system.

- We relied on OM-AM reference to design a novel access control framework tailored specific applications in the context of CEIoT environments needs. In fact, as regards identification, authentication and authorization mechanism, we opted for XACML language to ensure a robust security policy regarding ABAC model.

- The access decisions are managed by a rules engine, on the basis of entity customizable policies, written in XACML. One significant problem with XACML based policy is that it is more complex and difficult to manage. It's obvious that graphical tools can help users to easily design and implement security policies. To cope with such issues, we used security policy tool (SPT) to facilitate the elaboration of ABAC policy.

- Addressing identity issue requires reformulating the ABAC policy using XACML, however it generates an excessive overhead. Clearly, in this case it is not suitable for IoT. In light of this fact, classification is considered as an efficient approach to improve policy evaluation. Ideally, we use Fuzzy c-means to divide all rules into distinct cluster. The proposal undoubtedly has a significant capability to reduce time required to make access decision.

Future Work

The future work considers the following enhancements to the policy specification, enforcement and adjustment mechanisms:

- We have studied a complex case that involves hospital management system. Future extensions include testing in a larger and more complex setting, possibly characterised by the presence of a plurality of users and multiple conditions.
- The same remarks also apply to the security policy, which should be tested in a real IoT scenario, including more data sources and different application domains and, consequently, more complex policies.
- At the moment, on-going work consists of the application and testing of deep learning security policy conflict detection.
- Another future research direction is to propose a framework based deep learning approaches to extract and analyse attributes from natural language access control policies, and automatically identify the core properties needed to define the attributes contained in these policies.

Bibliography

[1] Asma. H. A. "Access Control Models for Cloud-Enabled Internet of Things." Ph.D Thesis, University of Texas at San Antonio, March 2018.

[2] Xinwen. Z., Yingjiu. L., and Divya. N. "An Attribute-Based Access Matrix Model." In Proceeding of the ACM Symposium on Applied Computing (SAC '05), pages 359-363, 2005.

[3] Malik. A. A., Hussain. B., and Kazmi, S. O. H." Access Control Model for Data Stored on Cloud Computing." International Journal of Innovation and Scientific Research, Vol. 20 No. 1, pages 233-241, January 2016.

[4] Amit. H., and Karuna. P. J."A Semantic Approach to Cloud Security and Compliance," IEEE 8th International Conference on Cloud Computing, pp. 1081-1084, 2015.

[5] Preeti. S., and Amit. A. "Cloud computing data storage security framework relating to data integrity, privacy and trust," In the 1st International Conference on Next Generation Computing Technologies (NGCT), pp. 115-118, IEEE, January 2016.

[6] Peter. M., and Tim. G. "The NIST definition of cloud computing." NIST special publication, 800-145, September 2011, 2011.

[7] Ahmed E. Youssef. "Exploring Cloud Computing Services and Applications." Journal of Emerging Trends in Computing and Information Sciences, Vol. 3, No. 6, July 2012.

[8] Bokefode. J. D, and Apte. S. S. "Analysis of DAC MAC RBAC Access Control based Models for Security." International Journal of Computer Applications, (0975 – 8887), Vol. 104, October 2014.

[9] Sarada. W., Lakshmi. B. P., and Padmalatha. V. "A Comparative study on cloud models and services." International Journal of Advanced Research in Computer Engineering and Technology (IJARCET), Vol. 3, No.10, October 2014.

[10] Salman. I., Miss. L. M. K., Nor. B. A., Babak. D., Ainuddin .W.A. W., and Suleman . K. "Service delivery models of cloud computing: security issues and

open challenges." Security and Communication Networks, pp. 4726–4750, August 2016.

[11] William. V., James. B., and Rajkumar. B. "Introduction to cloud computing. In Cloud Computing: Principles and Paradigms." (ed.) by R. Buyya, J. Broberg, A.Goscinski, Pp. 1–41, Wiley, 2011.

[12] Fang. L., Jin. T., Jian. M., Robert. B. B., John. V. M., Mark L. B., and Dawn M. L. "NIST cloud computing reference architecture." NIST, special publication, pp. 500-292, 2011.

[13] Asharul. I., Mohammed. I., Khalid. M., and Hashim. A-K. "Cloud: The Global Transformation," In International Conference on Cloud & Ubiquitous Computing & Emerging Technologies, pp. 58-62, IEEE, 2013.

[14] Ahmed. S. and Maria. S. "Cloud computing: paradigms and technologies." In Xhafa F., Bessis N. (eds) Inter-cooperative Collective Intelligence: Techniques and Applications. Studies in Computational Intelligence, Vol. 495, pp. 39-67, Springer, 2014.

[15] Farid. S., Adnan. S., and Amna. I. "Cloud computing security and privacy: an empirical study." Lecture Notes in Computer Science, , Vol. 10272, pp. 534-549, Springer, December 2017.

[16] Ravi. P. K., Raj. P. H., and Jelciana. P. "Exploring security issues and solutions in cloud computing services – a survey." Cybernetics and Information Technologies, pp. 3-31, July 2017.

[17] Zahir. T. "Security and Privacy in Cloud Computing," IEEE Cloud Computing, Vol. 1, No. 1, pp. 54-57, May 2014.

[18] Abdullah. A., Harkeerat. B., and Sajjan. S. "Towards a Stakeholder-Oriented Taxonomical Approach for Secure Cloud Computing," IEEE Sixth International Conference on Cloud Computing, pp. 958-959, July 2013.

[19] Buket. Y., Alptekin., and Öznur. Ö. "Research issues for privacy and security of electronic health services." Future Generation Computer Systems, Vol. 68, pp. 1-13, March 2017.

[20] Gururaj. R., Mohsin. I., and Farrukh. A. K. "A comprehensive survey on security in cloud computing." In Procedia Computer Science, Elsevier, Vol. 110, pp. 465–472, December 2017.

[21] Harikrishna. B., Kiran. S., Murali .G., and Reddy. R.P.K. “Security issues in service model of cloud computing environment.” In Procedia Computer Science, Vol. 87, pp. 246-251, April 2016.

[22] Mbarek. M., Feda. A., Fatima. S., Ali. K., and Hassan. O. “Improving the security of cloud-based medical image storage”. Engineering Letters, pp.175-193, February 2019.

[23] Assad. A., and Samee. U. K. “E-Health cloud: privacy concerns and mitigation strategies.” In Gkoulalas-Divanis, A., Loukides, G. (eds.) Medical Data Privacy Handbook, pp. 389–421. Springer, 2015.

[24] Noura. H. A-N., Asma. A., Nader. M., and Jameela. A-J. “E-Health cloud implementation issues and efforts.” In the International Conference on industrial Engineering and Operations Management, pp. 1–10, March 2015.

[25] Shashikala. S., and Kavitha. V. “A survey on security issues in service delivery models of cloud computing.” Journal of Network and Computer Applications, pp. 1–11, January 2011

[26] Mbarek. M., Ali. K., and Hassan. O. “Towards Optimizing the Usability of Homomorphic Encryption in Cloud-Based Medical Image Processing.” In International Symposium on Ubiquitous Networking, May 2017.

[27] Randolph. C. B., and Paul. D. C." Privacy, Confidentiality: and Electronic Medical Records." Journal of the American Medical Informatics Association, Vol.3, No. 2, April 1996.

[28] ALI. G." Security and Privacy of Sensitive Data in Cloud Computing." At KTH School of Computer Science and Communication, Doctoral Thesis, April 2016.

[29] Abhishek. M., Suyel. N., and Samir. N. “Taxonomy and Classification of Access Control Models for Cloud Environments.” Springer, 2014.

[30] Punithasurya. K., and Jeba. P. S. “Analysis of Different Access Control Mechanism in Cloud.” International Journal of Applied Information Systems (IJAIS). Vol .4, No.2, September 2012.

[31] Rajanikanth. A., and Lakshmi. M. "A Survey on Access Control Models in Cloud Computing." In: Satapathy S., Govardhan A., Raju K., Mandal J. (eds) Emerging ICT for Bridging the Future. In the 49th Annual Convention of the Computer Society of India (CSI), Vol.1, Springer, January 2015.

[32] Chen. D., Huang. X., and Ren. X. “Access Control of Cloud Service Based on UCON.” In: Jaatun M.G., Zhao G., Rong C. (eds) Cloud Computing. CloudCom 2009. Lecture Notes in Computer Science, Vol. 5931, pp. 559–564, Springer, 2009.

[33] Anant. V. N, and Soumya. K. G. “A Theoretical Study on Access Control Model in Federated Systems.” In: Martínez Pérez G., Thampi S.M., Ko R., Shu L. (eds) Recent Trends in Computer Networks and Distributed Systems Security. SNDS. Communications in Computer and Information Science, Vol. 420. pp 310-321, Springer, 2014.

[34] Dongwan. S., Hakan. A., William. C., and Kwanjoong. K. “Toward role-based provisioning and access control for infrastructure as a service (IaaS).” J. Internet Serv. Appl, Vol. 2, No . 3, pp. 243–255, December 2011.

[35] ByungRae. C., JaeHyun. S., and JongWon. K. “Design of Attribute-Based Access Control in Cloud Computing Environment”. In: Kim K., Ahn S. (eds). In Proceedings of the International Conference on IT Convergence and Security 2011. Lecture Notes in Electrical Engineering, Vol. 120, Pp. 41-50, Springer, December 2011.

[36] Vincent. C. H., David. F., Rick. K., Adam. S., Kenneth. S., Robert. M., and Karen. S. "Guide to Attribute Based Access Control (ABAC) Definition and Considerations." NIST special publication, January 2014.

[37] Fatima. S., Ahmed. H., and Ali. K. “Ensuring Security in Cloud Computing Using Access Control: A Survey.” In: Ben Ahmed M., Boudhir A. (eds) Innovations in Smart Cities and Applications. SCAMS. In Proceedings of the Mediterranean Symposium on Smart City Applications. Lecture Notes in Networks and Systems, Vol. 37, pp. 255-264, Springer, March 2018.

[38] Kartik. U., Sumit. S. D., and Sanyog. R. “IoT: Pillars and Technology.” Indian Journal of Science and Technology, Vol.47, December 2016.

[39] Keyur. K. P., and Sunil. M. P. “Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges.” International Journal of Engineering Science and Computing, Vol.6, No. 5, May 2016.

[40] Internet of Things(IoT): https://www.techsparks.co.in/thesis-topics-in-internet-of-things/

[41] Hany F. A., Robert. J., and Gary. B. W. “Internet of Things: State-of-the-art, Challenges, Applications, and Open Issues.” International Journal of Intelligent Computing Research (IJICR), Vol. 9, No. 3, September 2018.

[42] Balaji. S., Karan. N., and Santhakumar. R. “IoT Technology, Applications and Challenges: A Contemporary Survey.” In Wireless Personal Communications, April 2019.

[43] Anna. T. , Panagiotis. S., and Thomas. D. L. “Network Protocols, Schemes, and Mechanisms for Internet of Things (IoT): Features, Open Challenges, and Trends.” In Wireless Communications and Mobile Computing, Vol. 2018, 24 pages, September 2018.

[44] Xiaolin. J., Quanyuan. F., Taihua. F., and Quanshui. L. “RFID technology and its applications in Internet of Things (IoT).” In the 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), pp. 1282-1285, April 2012.

[45] What technologies are used in IoT—technology behind Internet of Things: https://www.avsystem.com/blog/iot-technology/

[46] IoT technology & Protocols: https://data-flair.training/blogs/iot-technology/

[47] IoT technology & Protocols: https://data-flair.training/blogs/iot-technology/

[48] IoT Protocols and their Architecture: https://www.elprocus.com/iot-protocols-and-its-architectures/

[49] Standard IoT Protocols: https://www.ubuntupit.com/top-15-standard-iot-protocols-that-you-must-know-about/

[50] Ala. A. F., Mohsen. G., Mehdi. M., Mohammed. A., & Moussa. A." Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", IEEE Communication Surveys & Tutorials, Vol. 17, No. 4, pp. 2347-2376, June 2015.

[51] Sobin, C.C. “A Survey on Architecture, Protocols and Challenges in IoT.” In Wireless Pers Commun , Vol. 112, No. 3, pp. 1383–1429, January 2020.

[52] Asma. H. A., and Ravi. S. “Access Control Models for Cloud- Enabled Internet of Things: A Proposed Architecture and Research Agenda. “ IEEE 2nd International Conference on Collaboration and Internet Computing, pp. 530-538, November 2016.

[53] Pallavi. S., & Smruti. R. S. “Internet of Things: Architectures, Protocols, and Applications.” Journal of Electrical and Computer Engineering, January 2017.

[54] Mohammad. A, Imran. K, Aymen Abdullah. A and Eui-Nam. H. "Cloud of Things: Integrating Internet of Things and Cloud Computing and the Issues Involved", In the 11th International Bhurban Conference on Applied Sciences & Technology, pp. 414-419, January 2014.

[55] Muhammad. B, Rana A. R, Bilal. K and Byung-Seo. K. “IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey.” Open Access Journal of Sensors, Vol.8, August 2018.

[56] Glenn. D., Mike. E., Tim. H., Gopal. I., Heather. K., Eric. L., Bob. M., Bob. M., Peter. N., and Alex. T. “Cloud Customer Architecture for IoT.” In Cloud Standards Customer Council, 2016.

[57] Miroslav. B., Xavier. B., Karel. F., and Bestoun. S.A “A Comprehensive View on Quality Characteristics of the IoT Solutions.” In: José R., Van Laerhoven K., Rodrigues H. (eds), In the 3rd EAI International Conference on IoT in Urban Space. Urb-IoT 2018. EAI/Springer Innovations in Communication and Computing. Springer, October 2018.

[58] From Nordic: "IoT Wireless Architecture": https://www.semiconductorstore.com/blog/2019/From-Nordic-IoT-Wireless-Architecture/3865/

[59] Yingnan. S., Frank. P.-W. L., and Benny. L. “Security and Privacy for the Internet of Medical Things Enabled Healthcare Systems: A Survey.” In IEEE Access, Vol. 7, pp. 183339-183355, December 2019.

[60] Top 10 Biggest IoT Security Issues: https://www.intellectsoft.net/blog/biggest-iot-security-issues/

[61] Internet of Things Security: Challenges and Best Practices: https://www.apriorit.com/dev-blog/513-iot-security

[62] Ado. A. A. A., Olga. K. N., Chafiq. T., Ousmane. T., Kolyang., Alidou. M., & Abdelhak. M. G. “Enabling privacy and security in Cloud of Things: Architecture, applications, security & privacy challenges.” Journal of Applied Computing and Informatics, November 2019.

[63] Arsalan.M., and Niraj. K. J. “A comprehensive study of security of internet-of-things,” IEEE Transactions on Emerging Topics in Computing, vol. 5, no. 4, pp. 586–602, December 2017.

[64] Ikram. U. D., Mohsen. G., Byung. S. k., Suhaidi. H., and Muhammad. K. K. “Trust management techniques for the internet of things: A survey,” IEEE Access, vol. 7, pp. 29 763–29 787, November 2018.

[65] Nomusa. N. D., and Kevin. J. “The use, benefits and challenges of using the internet of things (iot) in retail businesses: A literature review,” In the International Conference on Advances in Computing and Communication Engineering (ICACCE). IEEE, pp. 430–436, November 2016.

[66] Alexandra. S., Titus-Constain. B., Carmen. G., and Sorin. Z. “Securing the iot gateway based on the hardware implementation of a multi pattern search algorithm,” In the International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP), pp. 1001–1006, IEEE, May 2017.

[67] Yaqoob, I.; Ahmed, E.; Hashem, I.A.T.; Ahmed, A.I.A.; Gani, A.; Imran, M.; Guizani, M. “Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges.” IEEE Wirel. Communcations, Vol. 24, No. 3, pp. 10-16, June 2017,

[68] Jing. Q., Vasilako. A. V., Wan. J., Lu. J., and Qiu. D. “Security of the Internet of things: Perspectives and challenges.” In Wireless Networks, Vol. 20, No. 8, pp. 2481–2501, 2014.

[69] Sicari, S.; Rizzardi, A.; Grieco, L.A.; Coen-Porisini, A. “Security, privacy and trust in Internet of Things: The road ahead.” In Computer Networks, Vol. 76, pp. 146–164, January 2015.

[70] Noboru. k., and Ken. S. “Ubiquitous ID: Standards for ubiquitous computing and the Internet of Things.” IEEE Pervasive Computing., Vol. 9, No. 4, pp. 98–101, December 2010.

[71] Wei. W., Peng. X., and Laurence. T. Y. ". “Secure data collection, storage and access in cloud-assisted iot.” IEEE Cloud Computing, Vol. 5, No. 4, pp. 77–88, January 2018.

[72] Muhammad. B. M., Md.Abul. K. A., and Athanasios. V. "Secure data sharing and searching at the edge of cloud-assisted internet of things," IEEE Cloud Computing, Vol. 4, No. 1, pp. 34–42, February 2017.

[73] IoT System / Sensors and Actuators: https://bridgera.com/IoT-system-sensors-actuators/

[74] Crosby, G.V.; Vafa, F. "Wireless sensor networks and LTE-A network convergence." In Proceedings of the IEEE 38th Conference on Local Computer Networks (LCN), pp. 731–734, October 2013.

[75] Securing the IoT Gateway: https://www.citrix.com/blogs/2015/07/24/securing-the-iot-gateway/

[76] Sinan. D., and Oguzhan. U. "Secure gateway for network layer safety in IoT systems," In the 26th Signal Processing and Communications Applications Conference (SIU), pp. 1-4, May 2018.

[77] Cao. Q., Abdelzahe. T., Stankovic. J., and He. T. "The liteos operating system: Towards unix-like abstractions for wireless sensor networks." In Proceedings of the International Conference on Information Processing in Sensor Networks, 2008 (IPSN'08), pp. 233–244, April 2008.

[78] Alessio. B., Walter. de. D., Valerio. P., and Antonio. P. "On the Integration of Cloud Computing and Internet of Things," In the International Conference on Future Internet of Things and Cloud, Vol. 56, pp. 23-30, 2014.

[79] Jiang. R., and Sun. A., "Architecture Design of the Internet of Things Based on Cloud Computing," 2015 Seventh International Conference on Measuring Technology and Mechatronics Automation, pp. 206-209, June 2015.

[80] Zhiyoung. Z., Ln. Y., Qingpi. P., and Jianfeng. M. "Research on Usage Control Model with Delegation Characteristics Based on OM-AM Methodology," IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007), pp. 238-243, September 2007.

[81] Jaehong. P., and Ravi. S. "Towards an Engineering Framework for Usage Control and Digital Rights Management." October 2011.

[82] Aafaf. O., Hajar. M., Anas. A. E., and Abdellah .A. O. "Access control in The Internet of Things: Big challenges and new opportunitie." Journal of Computer Networks, Vol. 12, pp. 237-262, November 2016.

[83] Ravi. S. “Engineering Authority and Trust in Cyberspace: The OM-AM and RBAC Way.” In Proceedings of the fifth ACM workshop on Role-based access control, pp. 111-119, July 2000.

[84] Fatima. S., Mbarek. M., and Ahmed. H “Applying OM-AM Reference to an ABAC Model for Securing Cloud-Enabled Internet of Things.” In the 3rd International Conference on System Reliability and Safety (ICSRS), pp. 86-91, November 2018.

[85] Hai-bo. S, and Fan. H. “An Attribute-Based Access Control Model for Web Services.” In proceedings of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06), December 2006.

[86] Djebari. N., Hachem. S., Hassina. N., Djamil. A., and Kada. B. B. "ABAC Conceptual Graph Model for Composite Web Services," IEEE 5th International Congress on Information Science and Technology (CiSt), , pp. 36-41, October 2018.

[87] Aafaf. O., Anas. A. E., and Abdellah .A. O. “FairAccess: a new Blockchain-based access control framework for the Internet of Things.” In Security and Communication Networks, February 2017.

[88] M. Díaz, C. Martín, and B. Rubio, “State-of-the-art, challenges, and open issues in the integration of internet of things and cloud computing,” Journal of Network and Computer Applications, Elsevier, Vol. 67, pp. 99-117, May 2016.

[89] Attribute-based access control: https://www.axiomatics.com/attribute-based-access-control/

[90] Vandna. D., and Sandeep. D., "Fog Computing: A Review on Integration of Cloud Computing and Internet of Things." IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), pp. 1-6, February 2018.

[91] Shaik. M. B., Jaya. A. L., and Thirumala. A. R. “A study on cloud based Internet of Things: CloudIoT,” In Global Conference on Communication Technologies (GCCT), pp. 60-65, April 2015.

[92] Mbarek. M., Ali. K., and Hassan. O. “A framework to secure medical image storage in cloud computing environment,” Journal of Electronic Commerce in Organizations, Vol. 16, No. 1, pp. 1-16, January 2018.

[93] Mbarek. M., Ali. K., and Hassan. O. "Secure cloud-based medical image storage using secret share scheme," In the International Conference on Multimedia Computing and Systems (ICMCS), pp. 366–371, October 2016.

[94] Mbarek. M., Ali. K., and Hassan. O. "Applying secure multi-party computation to improve collaboration in healthcare cloud." In Third International Conference on Systems of Collaboration (SysCo), pp. 1-6, November 2016.

[95] Hany F. A., Ahmed. A., Abdulrahman. A., Rober J. W. Gray B. W. "Integration of cloud computing with internet of things: challenges and open issues," IEEE International Conference on Internet of Things (iThings), pp. 670-675, June 2017.

[96] Wu. H., Guoqiang. Y., and Xu. L. D. "Developing vehicular data cloud services in the IoT environment." IEEE Transactions on Industrial Informatics, vol. 10, No.2, pp. 1587-1595, May 2014.

[97] Javeria. S., Karl. R., and Seng W. L. "A risk aware development and deployment methodology for cloud enabled Internet-of-Things." IEEE 4th World Forum on Internet of Things (WF-IoT), pp. 433-438, February 2018.

[98] Fatima. S., Mbarek. M., and Ahmed. H "Applying OM-AM Reference to an ABAC Model for Securing Cloud-Enabled Internet of Things." In the 3rd International Conference on System Reliability and Safety (ICSRS), November 2018.

[99] Jatinder. S., Thomas. P., Jean. B., Hajoon. K., and David. E. "Twenty security considerations for cloud-supported Internet of Things." IEEE, Internet of Things Journal, Vol.3, No. 3, pp. 269-284, June 2016.

[100] Abdl-Aziz, A. A., and Kannan. A. "A comprehensive presentation to XACML." In the Third International Conference on Computational Intelligence and Information Technology (CIIT), pp. 155-161, October 2013.

[101] Understanding XACML combining: https://www.axiomatics.com/blog/understanding-xacml-combining-algorithms/

[102] Yared. K., Jan H. P. E., and H.s. Venter. "Proposing a secure XACML architecture ensuring privacy and trust." January 2005.

[103] Bo. L., Nan. Z., Kun. G., and Kai. C. "An XACML Policy Generating Method Based on Policy View." In Third International Conference on Pervasive Computing and Applications, pp. 295-301, October 2008.

[104] Fidel. P. D., Amrutha. C. V., Diego. S. T., and José. M. S. C. "Modeling XACML Security Policies Using Graph Databases." In IT Professional, Vol. 19, No. 6, pp. 52-57, December 2017.

[105] https://www.axiomatics.com/100-pure-xacml/

[106] Cataldo. B., and Antonio. L." Modern Standard-based Access Control in Network Services: XACML in action." IJCSNS International Journal of Computer and Network Security, Vol.8 No.12, December 2008.

[107] XACML Editors Today and How They Should Be?: https://medium.com/@chathurangijks/xacml-editors-today-and-how-they-should-be-2ee941f56c9f

[108] http://umu-xacmleditor.sourceforge.net/

[109] Henrik. N., Nils. U. M., and Terje. G." A Scratch-based Graphical Policy Editor for XACML." In the 1st International Conference on Information Systems Security and Privacy, February 2015

[110] AXIOMATICS POLICY SERVER 6.2 Enterprise-wide authorization for applications

[111] What is Security Policy Tool?: https://securitypolicytool.com/help

[112] Fatima. S., Fedaa. A., Adra. H., Mbarek. M., and Ahmed. H. "Implementing Policy Rules in Attributes Based Access Control with XACML within Cloud-Enabled IoT Environment." Journal of Communications, Vol. 15,, pp. 107-112, January 2020.

[113] Security Policy Tool User Manual, Info Beyond Technology LLC, August 2017.

[114] Security Policy Tool: 5 Ways to Close The Door to Access Control Leaks: https://securitypolicytool.com/security-policy-tool-5-ways-to-close-the-door-to-access-control-leaks

[115] SECURITY POLICY TOOL: https://securitypolicytool.com/product#S2

[116] Dijiang. H., and Huijiun. W. "Mobile Cloud Security: Attribute-Based Access Control." Mobile Cloud Computing, pp. 181-211, September 2017.

[117] Prathima. R., Dan. L., Elisa. B., Ninghui. L., and Jorge. L. “An Algebra for Fine-Grained Integration of XACML Policies." In the 14th ACM symposium on Access Control Models and Technologies, pages 63–72, June 2009.

[118] Florian. H. "Conflict Detection and Resolution of XACML Policies." Master Thesis at the University of Applied Sciences Rapperswil, July 2010.

[119] Christian. H., and Anca. M. “Computing ε-free nfa from regular expressions in O(nlog2(n)) time.” Technical report, Institut f¨ur Informatik, Universit¨at Stuttgart, pp. 257-277, 1998.

[120] Maryem. H., Meryeme. A., Yahya. B., Ahmed. K., and Mohammed. E. “Clustering-based Approach for Anomaly Detection in XACML Policies." In the 14th International Joint Conference on e-Business and Telecommunications, Vol.4, pp. 548-553, December 2017.

[121] Maryem. H., Ahmed. K., Yahya. B., and Mohammed. E. “Formal Approach to Detect and Resolve Anomalies while Clustering ABAC Policies.” In the EAI Endorsed Transactions on Security and Safety, December 2018.

[122] Mohamed. Y., Ahmed. Z., and Mostafa. H. "Conflicts between Geoxacml Access Control Policies in Geographic Information Systems." International Journal on Computer Science and Information Systems, Vol. 9, No. 1, pp. 16-29,

[123] Mafruz. Z. A., David. T., and Kate. S-M. "A New Approach of Eliminating Redundant Association Rules." In Lecture Notes in Computer Science, August 2004.

[124] Mohan. A., and Blough. D.M. “An attribute based authorization policy framework with dynamic conflict resolution.” In the 9th Symposium on Identity and Trust on the Internet, pp. 37-50, April 2010.

[125] Varun. C., Arindam.B., and Vipin. K. “Anomaly detection for discrete sequences: A survey.” IEEE Transactions on Knowledge and Data Engineering, Vol. 24, No. 5, pp.823-839, May 2012.

[126] Samarati. P., and de Vimercati. S. C. “Access control: Policies, models, and mechanisms. In International School on Foundations of Security Analysis and Design, pp. 137-196, Springer, September 2000.

[127] Attribute-Based Access Control (ABAC): https://www.axiomatics.com/attribute-based-access-control/

[128] Maryem. H., Yahya. B., Bernd. F., and Mohammed. E. "ABAC rule reduction via similarity computation," In International Conference on Networked Systems, pp. 86-100, May 2017.

[129] Michel. S-M., and Amy. P. F.. "A verified algorithm for detecting conflicts in XACML access control rules," In the 5th ACM SIGPLAN Conference on Certified Pograms and Proofs, ACM, pp. 166–175, January 2016.

[130] Apurva. M, Blough. D. M., Kurc. T., Post. A., and Saltz. J. "Detection of conflicts and inconsistencies in taxonomy-based authorization policies," In International Conference on Bioinformatics and Biomedicine, IEEE, pp. 590-594, November 2011.

[131] Maryem. H., Yahya. B., Ahmed. K., and Mohammed. E. "Access Domain-Based Approach for Anomaly Detection and Resolution in XACML Policies." In International Conference on Innovations in Bio-Inspired Computing and Applications, pp. 298-308, Springer, December 2017.

[132] Jiang. L., Hongqi. Z., Xiangdong. D., and Yigong. W. "A static ABAC policy conflict resolution algorithm," In the 4th International Conference on Multimedia Information Networking and Security, pp. 83–86, IEEE, 2012.

[133] Aijuan. Z., Cheng. J., Yu. B., and Xin. L. "Conflict analysis and detection based on model checking for spatial access control policy. Tsinghua Science and Technology, pp.478-488, October 2017.

[134] Hongxin. H., Gail-Joon. A., and Ketan. K. "Anomaly Discovery and Resolution in Web Access Control Policies." In 16th ACM Symposium on Access Control Models and Technologies, June, 2011.

[135] Cheng-chun. S., Erica Y. Y., and Alvaro E.A. "Detecting Conflicts in ABAC Policies with Rule-reduction and Binary-search Techniques." In International Symposium on Policy for Distributed Systems and Networks, IEEE , August 2009.

[136] Zeshui. X., and Junjie. W. "Intuitionistic fuzzy C-means clustering algorithms." Journal of Systems Engineering and Electronics Vol. 21, No. 4, pp.580–59, August 2010.

[137] Introduction to Fuzzy Logic and its Application to Text Summarization: https://becominghuman.ai/introduction-to-fuzzy-logic-and-its-application-to-text-summarization-dee71da67bbe

[138] https://www.google.co.uk/search?q=hard+and+soft+clustering&client=ms-android-samsung&source=lnms&tbm=isch&sa=X&ved=2ahUKEwj15fHT66TpAhVJSxoKHUmRBk8Q_AUoAXoECBIQAw

[139] Samira. D., Ibtissam. B., and Bouabid. El-O. "Hybrid approach for Intrusion detection using fuzzy association rules plus anomaly and misuse detection," Journal of Machine Learning and Computing, Vol. 8, No. 5, October 2018.

[140] Fuzzy C-Means Clustering: https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/cmeans.html

[141] https://www.google.co.uk/search?q=Diagram+comparing+the+difference+between+classical+set+and+fuzzy+&tbm=isch&ved=2ahUKEwiI49TX8KTpAhUJwoUKHXuqAIEQ2-cCegQIABAA

[142] Z. Cebeci, and F.Yildiz "Comparison of k-means and fuzzy c-means algorithms on different cluster structures", Journal of Agricultural Informatics, Vol. 6, No. 3, October 2015.

[143] Fernando. L. ""Fuzzy c-means algorithm,": http://www.fernandolobo.info/dm/slides/fuzzy-c-means.pdf

[144] Velmurugan. T. "Performance based analysis between k-Means and Fuzzy C-Means clustering algorithms for connection oriented telecommunication data." Journal of Applied Soft Computing Vol. 19, pp. 134-146, June 2014.

[145] Z. Xu and J. Wu, "Intuitionistic fuzzy c-means clustering algorithms," Journal of Systems Engineering and Electronics, Vol. 21, No. 4, August 2010.

[146] R. Ghogge, "Brain tumour detection using k-means and fuzzy c-means clustering algorithm," Journal of Science, Engineering and Technology Research, Vol. 3, No. 7, July 2014.

[147] KaiLe. W., Chao. F, and ShanLin. Y. "Fuzziness parameter selection in fuzzy c-means: The perspective of cluster validation," Science China Information Sciences, Springer, Vol. 57, pp. 1–8, September 2014.

[148] Salam Al-A., Sebastian. M., Agnieszka. M., and Kesra. N. "A Comparison of K-Means And Fuzzy C-Means Clustering Methods For A Sample Of Gulf Cooperation Council Stock Markets." Folia Oeconomica Stetinensia, Vol 4. No 3, June 2015.

[149] Shedthi. B. S., Shetty. S., and Siddappa. M. "Implementation and comparison of K-means and fuzzy C-means algorithms for agricultural data," In the International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 105-108, March 2017.

[150] https://www.google.com/search?q=Difference+between+hard+and+soft+clustering+schema&source=lnms&tbm=isch&sa=X&ved=0ahUKEwig9orPxfjiAhVhx4UKHQ8jCw8Q_AUIECgB&biw=1366&bih=657#imgrc=2z0gCdL44OTIKM:

ABSTRACT

The Internet of Things (IoT) paradigm promises to make any things/objects part of both industries and humans every-day life. This sensor network is opening valuable opportunities for a large number of novel applications that promise to improve the quality of humans' live. The present thesis investigates the integration of cloud and IoT, called Cloud-enabled IoT to tackle the main challenges facing IoT systems, such as device's resource scarcity, energy storage capacities, memory resource and processors' computational capacities. Initially, this thesis explores the security and privacy challenges encountered in adopting cloud computing and IoT. Thus, a comprehensive taxonomy of vulnerabilities and threats in distributed systems, especially IoT and cloud, are highlighted, setting the context of the work, and investigating the state-of-the-art.

The main goal of this PhD thesis is to design and implement an efficient access control system tailored to the IoT application domain. In order to achieve this, we study and compare the most commonly used access control models in distributed environments. We shall use Attributes-Based Access Control (ABAC) as a secure, flexible and scalable security control mechanism so that it meets the required properties of cloud-enabled IoT. The second original contribution suggests OM-AM reference to design the appropriate access control for cloud-enabled IoT with respect to technical constraints. The effectiveness of the proposed solution is evaluated by means of real XACML-based policies. Our third contribution deals with policy conflict detection and resolution in ABAC. In this respect, we opt for clustering approach, especially the fuzzy c-means (FCM) algorithm, to detect and resolve anomalies. We prove the correctness of this method in terms of computational complexity.

Keywords: *Internet of Things, Cloud Computing, Access Control Models, ABAC Model, XACML Language, Fuzzy C-Means Algorithm.*

RESUME

L'Internet des objets (IoT) joue un rôle très important dans plusieurs domaines. Ainsi, elle est utilisée dans le domaine logistique pour la gestion des stocks et des acheminements. Aussi dans le domaine environnemental pour la surveillance des paramètres météorologiques. Sans oublier le domaine médical pour le suivi des paramètres vitaux. Ainsi, cette technologie permet d'interconnecter plusieurs objets, de collecter plusieurs types d'informations, de transmettre les données en utilisant des réseaux sans fil et de traiter certaines données pour l'aide à la prise de décision. Malgré les gains offerts par cette technologie, elle rencontre deux problèmes majeurs qui freinent son utilisation sur une échelle plus large. Le premier consiste au stockage des données volumineuses sur des capteurs de capacité de mémoire limitées, le second dépend de la consommation importante de l'énergie stockée sur des batteries de charge limitée.

Le Cloud Computing se présente comme la solution idéale pour externaliser le stockage et le traitement des données volumineuses. Cependant, la sécurité des données critiques et des paramètres confidentiels est citée comme l'obstacle principal dans le processus de l'intégration des technologies du Cloud Computing et de l'IoT. Ainsi, un mécanisme de contrôle d'accès fort et puissant au niveau des ressources du Cloud est nécessaire pour protéger le patrimoine informatique des entreprises des attaques malveillantes et des abus de confiance.

Dans ce contexte, nous proposons d'abord un nouveau Framework basé sur le modèle ABAC pour assurer l'accès sécurisé aux ressources déployées dans le Cloud. Ensuite, pour vérifier et contrôler les requêtes d'accès, nous utilisons un nouvel algorithme qui permet de détecter et réduire les redondances et les conflits dans une politique de sécurité. Cet algorithme est basé sur la méthode Fuzzy C-Means (FCM) pour la classification des règles afin de faciliter l'évaluation de la politique de sécurité. Enfin, nous identifions les principales voies futures de recherches sur ce sujet pour améliorer la sécurité et la qualité des services de cette nouvelle architecture.

Mots-clés: *L'Internet des Objets, Informatique en Nuage, les Modèles du Contrôle d'Accès, le Modèle ABAC,* Langage XACML, L'algorithme Fuzzy C-Means.

✉ Faculté des Sciences, avenue Ibn Battouta, BP. 1014 RP, Rabat –Maroc
☎ 00212(0)5 37 77 18 76 🖷 00212(0)5 37 77 42 61 ; http://www.fsr.Um5.ac.ma