MÉMOIRE

PRÉSENTÉ À

L'UNIVERSITÉ DU QUÉBEC À RIMOUSKI

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

en vue de l'obtention du grade de maître ès sciences (M.Sc.)

PAR

@ LINDA ALIANE

HOBAC: FAMILLE DE MODÈLES DE CONTRÔLE D'ACCÈS GÉNÉRALISANT ABAC

AOÛT 2020

Composition du jury:

[Mohamed Tarik Moutacalli], président du jury, [UQAR]

[Mehdi Adda], directeur de recherche, [UQAR]

[Abdenour Bouzouane], examinateur externe, [UQAC]

Dépôt initial [Juin 2020] Dépôt final [Août 2020]

RÉSUMÉ

L'environnement dynamique est l'une des principales caractéristiques de l'Internet des Objets ou Internet of Things (IoT), garantir et assurer la sécurité dans un tel environnement est simultanément une tâche primordiale et qui représente un grand défi.

L'un des aspects essentiels de la sécurité de données est le contrôle d'accès, ce mécanisme vise à contrôler l'accès à l'information en fonction des stratégies d'accès spécifiques afin d'empêcher tout accès illégitime. Différents modèles de contrôle d'accès ont été proposés pour restreindre l'accès aux données des dispositifs IoT. Attribute-Based Access Control (ABAC) est l'un des modèles et des normes qui ont reçu une attention significative au cours de ces dernières années. Dans ce modèle, les demandes d'accès des utilisateurs seront accordées ou refusées en fonction de différents attributs (attributs attribués aux objets, aux sujets et aux conditions environnementales) et d'un ensemble de règles spécifiées en fonction de ces attributs et conditions. Donc, les droits d'autorisations dans ABAC sont octroyés aux utilisateurs à travers une politique qui combine des attributs provenant principalement des objets, des sujets et des contextes. Il est à noter que différents modèles visant l'extension de ABAC ont été suggérés. Cependant, les politiques d'accès qui peuvent être générées de ces approches restent restrictives et peu flexibles, cela représente une limitation majeure en particulier dans le contexte IoT.

Ce projet de recherche est motivé par les opportunités prometteuses du mécanisme de contrôle d'accès qui est l'un des aspects de sécurité essentiels et critiques permettant de contrôler l'accès aux ressources protégées par la définition d'une politique qui précise notamment les conditions dans lesquelles une ressource peut être accédée par un utilisateur. En effet, dans ce mémoire, nous tirons parti des limitations des différents modèles de contrôle d'accès qui ont été proposés pour l'IoT en général et les modèles visant l'extension du modèle ABAC en particulier afin de proposer un nouveau modèle de contrôle d'accès nommé Higher-order Attribute-Based Access Control (HoBAC).

HoBAC est un nouveau modèle de contrôle d'accès qui est une généralisation du modèle ABAC. Ce nouveau modèle permet de mettre en oeuvre des politiques de contrôle d'accès flexibles, adaptées aux systèmes IoT et non IoT basées sur des hiérarchies d'entités (objets, sujets et contextes) construites à l'aide de la composition fine et les

opérations d'agrégations sur les attributs. Ce mécanisme d'abstraction représente une couche de sécurité supplémentaire qui permet d'assurer la sécurité des données en empêchant la manipulation directe des sujets, des objets et des contextes de bas niveau.

Ce travail de recherche présente les fondements théoriques de HoBAC et introduit également son architecture générale. Deux instances du modèle théorique sont présentées dans ce mémoire dont la première instanciation génère le modèle ABAC d'origine et la deuxième génère un modèle de contrôle d'accès à quatre couches qui convient aux systèmes IoT où la politique est distribuée à travers les différentes couches (périphériques IoT, la couche Edge, la couche Fog et la couche Cloud), ces instances permettent de montrer la flexibilité de notre modèle et qu'il est assez général pour exprimer différentes politiques de contrôle d'accès. Afin de montrer l'applicabilité des concepts de base du modèle HoBAC et ses relations, nous avons créé une application Web pour l'administration de sa politique d'accès. Ainsi, nous avons implémenté l'instanciation du modèle HoBAC de base ($HoBAC_0$) en utilisant la Policy Machine (PM) afin d'appliquer sa politique d'accès qui correspond au modèle ABAC.

ABSTRACT

The dynamic environment is one of the characteristics of the Internet of Things (IoT), ensuring security in such environment is a prominent and challenging task at the same time.

One of the vital aspects of data security is Access Control, it intends to control access to information based on access policies in order to prevent any illegitimate access. Different access control models have been proposed to restrict access to data of IoT devices. Attribute-Based Access Control (ABAC) is one of the most promising access control models and standards which has received meaningful attention in recent years. In this model, the access requests from users will be granted or denied based on different attributes (attributes assigned to objects, to subjects and environmental conditions) and a specified set of rules based on those attributes and conditions. Therefore, the access rights in ABAC are granted to users through a policy which combines a set of attributes which comes mainly from objects, subjects and contexts. It should be noted that several models that extend the ABAC model have been suggested. However, the access policies that can be generated from these approaches remain restrictive and they are not very flexible, this represents a major limitation especially in the IoT context.

This research project is motivated by the promising opportunities of the access control mechanism which is one of the essential and critical aspects of security that can control access to protected resources by the definition of policies that dictate the conditions under which a resource may be accessed by a user. Indeed, in this thesis, we take advantage of the limitations of the different access control models that have been proposed for IoT in general and the models extending the ABAC model in particular in order to propose a new access control model named Higher-order Attribute-Based Access Control (HoBAC).

HoBAC is a new access control model that is a generalization of the original ABAC model. This new model makes it possible to implement flexible access control policies adapted to IoT and non-IoT systems based on hierarchies of entities (objects, subjects and contexts) built using fine composition and aggregation operations on the attributes. This abstraction mechanism is by itself an additional security layer that ensures data security by preventing direct manipulation of the low-level objects, subjects and contexts.

This research work presents the theoretical foundations of HoBAC and introduces its general architecture. Two instances of the theoretical model are presented in this thesis, The first instantiation yields the original ABAC model and the second instance yields a four-layer AC model that is suitable for IoT systems where the policy is distributed through the different layers (sensors, edge, fog and cloud). These instances show the flexibility of our model and that it is general enough to express different access control policies. In order to show the applicability of the main concepts of HoBAC model and their relationships, we have created a Web application for administration of its access policy. Thus, we have implemented the instantiation of the basic HoBAC model ($HoBAC_0$) using the Policy Machine (PM) in order to apply its access control policy which is equivalent to the ABAC model.

Keywords— IoT, Security, Access Control, ABAC

REMERCIMENT

Je tiens tout d'abord à exprimer ma profonde gratitude à mon directeur de recherche, le professeur Mehdi Adda pour tout le soutien, l'aide et l'orientation qu'il m'a apporté tout au long de ce travail de recherche, il m'a patiemment accompagné avec une constante minutie. Je le remercie également pour sa gentillesse, sa patience, le temps inconditionnel qu'il m'a consacré et ses précieuses observations qui m'ont beaucoup appris sur la conduite de ce travail de recherche.

Mes remerciements s'adressent également à l'Institut technologique de maintenance industrielle (ITMI) et au Fonds de Recherches du Québec – Nature et Technologies (FQRNT) pour leur appui qui a favorisé l'accomplissement de ce travail de recherche.

J'adresse aussi mes plus profonds remerciements à mes chers parents, mon frère et mes soeurs et à toute ma famille, pour leur soutien et leur encouragement tout au long de mes études et la réalisation de ce mémoire.

DÉDICACE

Je tiens à rendre hommage à toute ma famille et plus particulièrement à mes très chers parents, mon frère et à mes soeurs qui m'ont apporté un soutien à toute occasion et qui m'ont accordé tout au long de mes études une entière attention.

Mes pensées les plus profondes s'adressent aussi à mes beaux-frères et ma belle-soeur, à mes chers neveux et nièces. Ce mémoire leur est entièrement dédié

CONTENTS

Résumé **iv**

Abstract **vi**

Remerciment **viii**

Dédicace **ix**

Contents **x**

List of Figures **xiv**

List of Tables **xvi**

List of Algorithms **xvii**

1 Introduction **1**

1.1 Contexte et Problématique 1

1.2 Objectifs 5

1.3 Méthodologie 6

1.4 Organisation du mémoire 7

2 Revue de la littérature **9**

2.1 Introduction 9

2.2 L'Internet des Objets (IoT) 10
2.3 L'Internet des Objets et le défi de sécurité 13
2.4 L'identification 16
2.5 L'authentification 19
2.6 Le contrôle d'accès 20
2.6.1 Modèles de contrôle d'accès 21
2.6.2 Modèle de contrôle d'accès discrétionnaire (DAC) 22
2.6.3 Modèle de contrôle d'accès obligatoire (MAC) 24
2.6.4 Modèle de contrôle d'accès à base de rôle (RBAC) 27
2.6.5 Architecture XACML 31
2.6.6 Le contrôle d'accès de nouvelle génération (NGAC) 33
2.6.7 Le modèle de contrôle d'accès basé sur les attributs (ABAC) et ses extensions 35
2.6.8 Passage au modèle de contrôle d'accès basé sur les attributs (ABAC) 38
2.7 Conclusion 51

3 Nouveau modèle de contrôle d'accès basé sur ABAC et les fonctions d'ordre supérieures 53

3.1 Introduction 53
3.2 Fondements théoriques de HoBAC 54
3.2.1 L'attribut 55
3.2.2 Types d'attributs 56
3.2.3 Instanciation de type d'attribut 58
3.2.4 Espace de type d'attribut 60
3.2.5 Entité 62

3.2.6 Instantiation de type d'entité 64
3.2.7 Espace de type d'entité 65
3.2.8 Relations entre les ancêtres et les successeurs de type d'entité . . 67
3.2.9 Types d'entités homogènes 67
3.2.10 Niveaux de dépendance d'espace de type d'entité 68
3.2.11 Type de règles d'accès 69
3.3 Conclusion 73

4 Implémentation du prototype du modèle HoBAC 74

4.1 Introduction 74
4.2 Composants de base du modèle HoBAC 75
4.3 L'architecture générale du modèle HoBAC 76
4.4 Cas d'utilisation de la famille des modèles HoBAC 79
4.5 HoBAC: présentation du prototype 82
4.5.1 Langage de développement du prototype 82
4.5.2 Modèle relationnel 83
4.5.3 Authentification 85
4.5.4 La création des entités 87
4.5.5 Les règles d'accès 89
4.6 Implémentation de HoBAC à l'aide de la Policy Machine (PM) 92
4.6.1 La Policy Machine (PM) 92
4.6.2 Implémentation de $HoBAC_0$ avec la Policy Machine 93
4.7 Conclusion 100

5 Conclusion 101

5.1 Résumé des objectifs 101

5.2 Travail accompli 102

5.3 Limitations et perspectives de développement futur 103

Bibliography **105**

LIST OF FIGURES

2.1 La croissance des objets connectés à Internet vs la population mondiale 11

2.2 Exemple d'une maison intelligente avec des objets liés 14

2.3 Exemple sur le contrôle d'accès discrétionnaire (DAC) 24

2.4 Exemple sur le problème de cheval de troie 25

2.5 Exemple simplifié du modèle RBAC 29

2.6 Attribution des permissions dans RBAC 29

2.7 Architecture XACML, adapté de Srijith [62] 33

2.8 Architecture NGAC, adapté de Ferraiolo et al. [25] 35

3.1 Cas d'utilisation lié à l'IoT (maison intelligente) 54

3.2 Une vue générale sur les principaux concepts du modèle HoBAC et leurs relations 55

3.3 Exemple d'instanciation de type d'attribut 60

3.4 Exemple de la relation de dépendance entre les types d'attributs 62

3.5 Exemple d'instanciation de type d'entité 66

3.6 Fédération des attributs dans HoBAC (1) 71

3.7 Fédération des attributs dans HoBAC (2) 72

4.1 Les principaux composants de HoBAC 75

4.2 L'architecture générale du modèle HoBAC 78

4.3 HoBAC: famille de modèles 80

4.4 ($HoBAC_4$) pour une architecture IoT à 4 couches 81
4.5 Le modèle relationnel de HoBAC 84
4.6 Cas d'utilisation qui exprime l'autorisation 85
4.7 Interface d'acceuil 86
4.8 La liste des formulaires 86
4.9 La liste des objets 87
4.10 La liste des entités 88
4.11 L'ajout des attributs à l'entité 89
4.12 La liste des objets 89
4.13 L'ajout d'une règles d'accès 90
4.14 Visualisation de la liste des règles d'accès 91
4.15 L'architecture d'autorisation du modèle HoBAC 95
4.16 Graphe de la politique d'accès 97
4.17 Exemple de la demande d'autorisation et sa réponse 99

LIST OF TABLES

2.1 Comparaison des fonctionnalités de RBAC et ABAC, adapté de [3] . . . 39

2.2 Tableau comparatif des différentes extensions de ABAC 43

4.1 Politique d'accès pour les opérations de lecture, écriture 98

LIST OF ALGORITHMS

1 La fonction d'unification F_m 69

List of Equations

3.0 $\mathcal{S}$-Structure pour une fonction d'ordre supérieur 57

3.0 Hauteur de dépendance du type d'attribut 61

CHAPITRE 1

INTRODUCTION

1.1 CONTEXTE ET PROBLÉMATIQUE

Internet des Objets ou Internet of Things (IoT) est le nom qui a été introduit par Kevin Ashton à la technologie qui connecte différents objets à Internet. Les capteurs, les actionneurs et autres dispositifs de l'IoT renforcent également les objets physiques avec des capacités de stockage, de détection, de traitement et de transmission des données ainsi que la capacité de se connecter à Internet, cela permet d'étendre leurs fonctionnalités, ces fonctionnalités améliorées transforment ces objets physiques quotidiens en objets intelligents (objets connectés) ce qui donne la possibilité d'y accéder et de communiquer avec eux à distance. Avec la généralisation et l'usage répandu de cette technologie un immense nombre de dispositifs qui ne cessent d'augmenter communiquent via Internet.

Certainement, le large déploiement de ces dispositifs qui sont en forte croissance tels que les capteurs détecteurs de fumée, les capteurs qui contrôlent les lumières et les températures dans e-Home, les montres et les bracelets connectés qui surveillent l'état des patients dans e-Health apportent une amélioration significative dans les différents domaines. Cependant, parmi les problèmes auxquels l'IoT est confronté est la sécurité des données qui est l'un des défis majeurs et les plus critiques qui ont le potentiel de faire réussir ou échouer l'Internet des Objets.

Par exemple, la collection des informations personnelles des capteurs d'une maison intelligente par un intrus peut nuire à la fois à l'utilisateur et à la réputation des entreprises fournissant ces services.

La protection des données contre toute utilisation non autorisée est primordiale afin d'assurer la sécurité d'un système informatique. Pour cela différentes techniques sont employées, parmi lesquelles le contrôle d'accès, la cryptographie, la solution de sécurité résistante aux attaques, l'authentification, la protection contre les intrusions dans les logiciels et les systèmes d'identification. Dans le cadre de notre mémoire, nous nous intéressons au contrôle d'accès afin d'empêcher l'accès illégitime aux données.

Le contrôle d'accès fait référence à un mécanisme qui permet de restreindre et limiter l'accès aux données en autorisant ou refusant la demande d'accès en appliquant des stratégies d'accès spécifiques. Les mécanismes de contrôle d'accès ont été mis en place due aux besoins spécifiques en matière de sécurité et de protection des environnements informatiques. Avec le large déploiement des dispositifs IoT et les enjeux de leur sécurité ces mécanismes sont devenus de plus en plus nécessaires afin de protéger les données de ces dispositifs. L'objectif principal de ces mécanismes est d'assurer la sécurité des données en les protégeant contre toute divulgation, modification ou une utilisation non autorisée, tout en garantissant l'accès aux utilisateurs légitimes uniquement.

Selon Margheri et al. [43], les mécanismes de contrôle d'accès sont des moyens largement utilisés pour la protection des systèmes informatiques. Ils sont définis en termes de politiques de contrôle d'accès réglementant l'accès aux ressources du système.

Quatre grandes classes de modèles de contrôle d'accès ont été proposées dans la littérature:

- Le contrôle d'accès discrétionnaire ou DAC (Discretionary Access Control): la politique d'accès dans ce modèle donne la possibilité aux utilisateurs de contrôler les autorisations d'accès aux objets qu'ils possèdent.
- Le contrôle d'accès obligatoire ou MAC (Mandatory Access Control): ce modèle est plus rigide que le modèle discrétionnaire dont les règles d'accès sont imposées par le système et les décisions d'accès ne doivent pas être prises par le propriétaire des objets.
- Le contrôle d'accès basé sur le rôle ou RBAC (Role-Based Access Control): le rôle est le concept de base dans ce modèle, dont les droits d'accès sont octroyés aux utilisateurs en fonction du rôle qu'ils jouent dans le système.
- Le contrôle d'accès basé sur les attributs ou ABAC (Attribute Based Access Control): comme l'indique son nom, c'est un modèle dont la décision d'accès est prise en fonction des attributs provenant des utilisateurs, des objets et des contextes.

Traditionnellement, le contrôle d'accès reposait sur l'identité d'un utilisateur afin de pouvoir effectuer une opération sur un objet donné directement ou via des types d'attributs prédéfinis tels que les rôles qui vont être attribués à cet utilisateur. Cependant, cette approche est souvent difficile à gérer étant donné la nécessité d'associer des fonctionnalités directement aux utilisateurs et à leurs rôles. Ainsi que cette approche est insuffisante pour exprimer des stratégies de contrôle d'accès réelles en se basant seulement sur les qualificatifs d'identité et de rôle.

Une alternative consiste à accorder ou à refuser les demandes des utilisateurs en fonction des attributs, cette approche est la plus pertinente pour les stratégies en cours. Il est

utile de signaler que le modèle de contrôle d'accès basé sur les attributs ou ABAC est un mécanisme qui a reçu une attention significative au cours de ces dernières années. Les systèmes ABAC sont capables d'appliquer à la fois les concepts de contrôle d'accès discrétionnaire (DAC) et de contrôle d'accès obligatoire (MAC). Ce modèle de contrôle d'accès a été proposé après les limitations des modèles précédents afin de donner plus de précisions aux décisions d'accès.

Les règles d'accès dans ABAC sont spécifiées à l'aide d'un ensemble d'attributs provenant principalement des utilisateurs, des objets, de l'environnement et d'autres attributs. Ainsi que différentes extensions de ce dernier ont été suggérées telles que les approches proposées par Servos and Osborn [60] et Bhatt et al. [12], ces extensions proposent la notion de groupes sur les objets et les sujets. Cependant, le regroupement est restrictif et n'offre pas de flexibilité, car ils exigent à ce que les objets (les sujets) du même groupe aient les mêmes attributs.

Malheureusement, il n'existe pas de modèles globales qui conviennent à la fois aux systèmes IoT et non IoT qui permettent d'unifier la définition de la politique d'accès et faciliter la migration d'une politique d'un modèle de contrôle d'accès à un autre.

1.2 OBJECTIFS

L'objectif principal visé par ce travail de recherche consiste à présenter un nouveau modèle de contrôle d'accès afin d'assurer la sécurité des données en garantissant l'accès aux utilisateurs légitimes uniquement à l'aide d'un mécanisme d'abstraction qui permet d'empêcher la manipulation directe des sujets, des objets et des contextes de bas niveau. Ce nouveau modèle est nommé Higher-order Attribute-Based Access Control (HoBAC) qui est une généralisation du modèle de contrôle d'accès basé sur les attributs (ABAC). Notre nouveau modèle a pour objectif de combler les limitations et les lacunes des modèles de contrôle d'accès pour l'IoT en général et de ABAC en particulier.

HoBAC permet de mettre en oeuvre des stratégies de contrôle d'accès générales et adaptées aux systèmes IoT et non IoT basées sur des hiérarchies d'entités (objets, sujets et contextes) construites à l'aide des opérations d'agrégations sur les attributs des entités existantes.

Nous étudierons tout d'abord dans ce mémoire comment le contrôle d'accès peut assurer la sécurité des données tout en garantissant l'accès aux utilisateurs légitimes uniquement, puis nous présenterons HoBAC, notre nouveau modèle de contrôle d'accès en introduisant les concepts de base du modèle théorique et en montrant à l'aide de deux instances son efficacité et comment il peut être instancié afin de générer un modèle équivalent au modèle de contrôle d'accès ABAC d'origine et à un autre modèle adapté aux systèmes IoT. Nous créerons ensuite une application Web pour l'administration de politique d'accès basées sur HoBAC afin de montrer l'applicabilité de ses concepts de base et leurs relations. Nous implémenterons également l'instanciation du modèle HoBAC de base ($HoBAC_0$) à l'aide de la Policy Machine (PM) qui est un cadre de contrôle d'accès générique permettant la définition et l'application des politiques de contrôle d'accès.

1.3 MÉTHODOLOGIE

La méthodologie suivie dans ce projet de recherche est divisée principalement en trois étapes:

La première étape visait à approfondir les connaissances dans le domaine de recherche en effectuant une revue de la littérature sur l'IoT, les défis critiques de cette technologie et plus précisément le défi de sécurité. Une étude approfondie a été effectuée sur les principes et les limitations des différents modèles de contrôle d'accès existants dans la littérature qui ont pour objectif de résoudre le défi de sécurité tout en limitant l'accès aux données des périphériques IoT en suivant une politique d'accès spécifique. Cette étape nous a menée à comprendre la nécessité et le besoin de mettre en place un modèle de contrôle d'accès global, plus flexible et qui convient à la fois aux systèmes IoT et non-IoT.

La deuxième étape avait pour objectif de définir les fondements théoriques d'un nouveau modèle de contrôle d'accès nommé HoBAC. Nous avons suivi une approche qui consiste à étendre les concepts de base du modèle de contrôle d'accès basé sur les attributs ou ABAC avec des opérations d'agrégations pour mettre en place un modèle ABAC générique afin d'offrir la possibilité d'implémenter des politiques de contrôle d'accès générales et adaptées aux systèmes IoT et non-IoT

Après avoir présenté le cadre théorique de HoBAC dans l'étape précédente, la dernière étape avait pour objectif de montrer la faisabilité de notre nouveau modèle de contrôle d'accès ainsi qu'il est assez général pour mettre en oeuvre des politiques de contrôle d'accès flexibles et adaptées aux systèmes IoT et non IoT, cela a été réalisé à l'aide de deux instances du modèle théorique. Nous avons ensuite présenté les composants de base du modèle HoBAC et son architecture générale, puis nous avons créé une application Web pour l'administration de politique d'accès de type HoBAC afin de

montrer l'applicabilité des concepts de base de HoBAC et leurs relations. Nous avons ainsi implémenté l'instanciation du modèle HoBAC de base ($HoBAC_0$) à l'aide de la Policy Machine (PM).

1.4 ORGANISATION DU MÉMOIRE

Ce mémoire est structuré en cinq chapitres présentés ci-dessous:

Dans le premier chapitre nous avons présenté l'introduction générale de ce mémoire, nous avons commencé par placer le travail dans son contexte général, nous avons ensuite situé la problématique de recherche. Les principaux objectifs suivis de la méthodologie adoptée pour cette recherche ont été ensuite définis.

Le deuxième chapitre est une revue de littérature qui présente et traite principalement les travaux antérieurs dans le domaine de la sécurité dans l'Internet des Objets (IoT). Nous avons présenté les défis critiques de l'IoT et plus particulièrement le défi de sécurité, nous avons ensuite présenté des axes de recherche importants qui portent sur l'identification, l'authentification et le contrôle d'accès pour la résolution du défi de sécurité lié à l'IoT. Nous avons introduit une analyse détaillée sur les modèles de contrôle d'accès existants pour assurer la sécurité des données en précisant notamment les principes et les limitations majeures de chacun de ces modèles.

Dans le troisième chapitre nous avons présenté la contribution de ce mémoire, en présentant les fondements théoriques de notre nouveau modèle de contrôle d'accès et ses concepts de base et leurs relations qui ont été introduits tout au long de ce document.

Dans le quatrième chapitre nous avons introduit les principaux composants et l'architecture générale de notre nouveau modèle de contrôle d'accès qui constitue une approche pour résoudre le problème de sécurité. Nous avons présenté deux instances du modèle

théorique pour montrer l'efficacité de ce nouveau modèle ainsi qu'il est assez général pour exprimer différentes politiques de contrôle d'accès. Afin de montrer l'applicabilité de ses principaux concepts et leurs relations, nous avons créé une application Web pour l'administration de politique d'accès basée sur HoBAC. Ainsi, nous avons implémenté dans ce chapitre l'instanciation du modèle HoBAC de base ($HoBAC_0$) en utilisant la Policy Machine (PM).

Enfin, le cinquième chapitre conclut le mémoire avec un résumé des objectifs de ce projet de recherche. Ainsi qu'un bilan sur le travail accompli, à la fin nous avons introduit les limitations de l'approche développée et les perspectives de développement futur.

La majeure partie des recherches présentées dans ce mémoire a été diffusée à la communauté scientifique à travers un article publié dans la conférence internationale « The 14th International Conference on Future Networks and Communications (FNC2019) » Aliane and Adda [5], étant donné la qualité de cet article et l'originalité de sa contribution, l'article a non seulement été désigné comme étant le meilleur article de toute la conférence (Best Paper Award) mais on a été invité à soumettre une version étendue de ce travail de maîtrise qui a été accepté dans une édition spéciale du journal international Springer « Journal of Ambient Intelligence and Humanized Computing ». Adda and Aliane [1]

CHAPITRE 2

REVUE DE LA LITTÉRATURE

2.1 INTRODUCTION

Après avoir précisé au sein de la problématique générale la place des mécanismes de contrôle d'accès et la possibilité d'assurer la sécurité des données à l'aide de ces derniers en les protégeant contre toute divulgation, modification ou une utilisation non autorisée tout en garantissant l'accès aux utilisateurs légitimes uniquement.

Dans ce chapitre nous allons aborder tout d'abord les différentes définitions attribuées à l'Internet des Objets dans la littérature. Nous allons également effectuer une revue de la littérature sur l'IoT et les défis critiques auxquels elle est confrontée et plus particulièrement le défi de sécurité. Ensuite, nous allons mettre en évidence des axes de recherche importants qui portent sur l'identification, l'authentification et le contrôle d'accès pour la résolution du défi de sécurité lié à l'IoT. En outre, nous allons apporter une analyse détaillée qui couvre les différentes approches et les modèles de contrôle d'accès existants dans la littérature en précisant notamment les principes et les limitations majeures de chacun de ces modèles.

2.2 L'INTERNET DES OBJETS (IOT)

Le concept de l'Internet des Objets ou Internet of Things (IoT) a été introduit pour la première fois par Kevin Ashton en 1999 en tant que titre d'une présentation où il a montré la possibilité d'identifier les objets et les utilisateurs en utilisant les étiquettes d'identification par radiofréquence (RFID). Syntaxiquement, ce concept est composé de deux termes « Internet » et « Things ».

Internet, qui désigne un réseau mondial de réseaux informatiques inter-connectés. Things, désigne des objets physiques ou virtuels tels que les montres, les voitures, les lampes, etc.

Nous allons présenter les différentes définitions qui sont attribuées à l'Internet des Objets dans la littérature.

L'Internet des Objets a été présenté par Sha et al. [61] comme étant *"Un paradigme émergent. Il est considéré comme la troisième vague d'innovation en technologie de l'information, après Internet et l'informatique mobile."* Selon Alshehri and Sandhu [6] l'Internet des objets est la dernière évolution de l'Internet, regroupant un très grand nombre d'objets physiques connectés.

Autrement, par Corici et al. [17] *"L'Internet des Objets est un domaine en évolution dans lequel les composants matériels et logiciels modélisent, détectent ou influencent les propriétés du monde physique. Pour exploiter le potentiel d'un service IoT, la sécurité et la flexibilité sont des exigences essentielles."* D'une autre manière, selon Murugesan et al. [50] l'Internet des Objets est un nouveau paradigme d'importance technique, sociale et économique. Lorsque les périphériques IoT sont combinés à la connectivité Internet et à des capacités de collecte et d'analyse analytiques puissantes, le potentiel de l'IoT devient signifiant.

Selon Vasilomanolakis et al. [64], les applications IoT incluent des usages domestiques tels que les maisons intelligentes, la mobilité et les transports, mais également des applications industrielles telles que les processus de fabrication intelligents et les réseaux d'énergie intelligents.

Au cours de ces dernières années le monde a vu une croissance fulgurante dans le monde de l'Internet des Objets, dont les objets connectés sont en forte croissance dans différents domaines, alors que la population humaine mondiale atteignait 6,8 milliards, le nombre d'appareils connectés à Internet a connu une croissance explosive atteignant près de 12 milliards comme montré dans la Figure 2.1 ci-dessous.

Selon Dave [21], le nombre d'appareils connectés à Internet a dépassé la population mondiale en 2009 et il est également prévu que 50 milliards d'appareils seront connectés à Internet d'ici 2020.

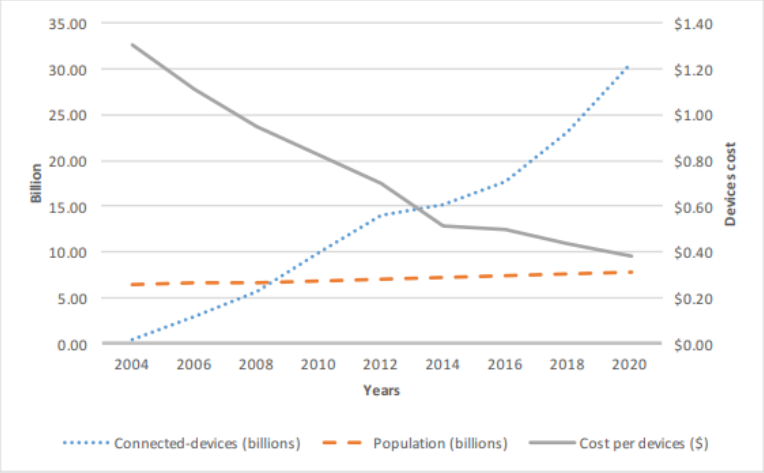

Figure 2.1 – La croissance des objets connectés à Internet vs la population mondiale. par Hossain et al. [28]

Dans le secteur de la santé, des systèmes de surveillance de l'état des patients à distance et qui fournissent un flux continu de données précises sont utilisés afin de prendre des meilleures décisions en matière des soins.

Hussein et al. [32] ont présenté une conception théorique d'un environnement Smart Home pour aider les personnes handicapées dans leur vie quotidienne. Puisque chaque personne souffre d'un handicap particulier, ce système remplace le manque de la personne, donc il n'est pas nécessaire qu'il soit configuré de la même manière pour toutes les personnes. Cependant, les systèmes e-health sont des systèmes critiques qui peuvent être sujets à divers risques, cela peut conduire à des conséquences catastrophiques.

Selon Mohd Ibrahim et al. [48], il est nécessaire d'intégrer la technologie IoT aux solutions de santé en ligne et aux dispositifs portables pour améliorer les soins de santé des patients. Fournissant ainsi un accès rapide et sécurisé aux dossiers médicaux des patients.

Dans les villes intelligentes, différentes technologies sont utilisées pour rendre facile la gestion du trafic, l'éclairage des rues, la surveillance des accidents, etc. Un réseau de capteurs et d'actionneurs faisant partie intégrante d'une application IoT a été déployée par Pérez and Rodriguez [53] afin de surveiller et de contrôler les parcs d'une ville. Chaque parc a été modélisé avec un ensemble de capteurs qui contrôlent principalement la température, l'humidité, l'état des sprinkleurs à eau et un actionneur pour ouvrir ou fermer les sprinkleurs. Cela a montré que l'analyse de l'évolutivité de ce type d'applications est possible par les techniques de simulation.

Le large déploiement des dispositifs IoT qui sont capables de détecter différents aspects dans un environnement réel a amélioré de nombreuses tâches dans la vie quotidienne, tels que la détection de l'humidité, la présence ou l'absence des personnes ou des objets dans une maison, ainsi que la possibilité d'échanger les données et d'agir en conséquence. Par exemple la possibilité de contrôler une maison intelligente en allumant

et en éteignant les lumières pour donner l'impression d'être habitée. Cependant, l'une des tâches cruciales et primordiales dans l'IoT est la sécurité des données.

2.3 L'INTERNET DES OBJETS ET LE DÉFI DE SÉCURITÉ

L'Internet des Objets (IoT) apporte une amélioration significative et rend les tâches quotidiennes plus faciles dans différents secteurs grâce à son évolution rapide et l'usage répandu de ses dispositifs et leurs véritables applications. Cependant, cette technologie affronte différents défis. La sécurité est l'un des problèmes les plus critiques qui ont le potentiel de faire réussir ou échouer l'IoT.

Le large déploiement des dispositifs IoT qui fonctionnent dans des environnements dynamiques et non protégés les rendent vulnérables aux attaques malveillantes et la répercussion de toute faille de sécurité aura des effets énormes sur la sécurité et la vie privée des personnes et des entreprises. Donc, si un attaquant a réussi à détecter un dispositif IoT vulnérable, il peut tirer parti de ce dernier pour infecter les autres dispositifs du réseau. Il faut signaler que la principale raison d'attaques malveillantes est la faible protection des ressources systèmes.

D'une autre manière, Danda and Hota [20] ont présenté le défi de sécurité: *"Un attaquant ayant un accès au réseau intégré qui connecte des appareils IoT ou des appareils peut espionner les gens, il peut injecter du code malveillant dans ces appareils intégrés, ce qui crée de graves problèmes de sécurité."*

La Figure 2.2 illustre un exemple d'une maison intelligente où le réfrigérateur est relié au four, le four est relié au détecteur de fumée, alors un attaquant peut tirer parti de n'importe quel objet vulnérable de cette maison afin d'accéder au verrou de la porte de la maison.

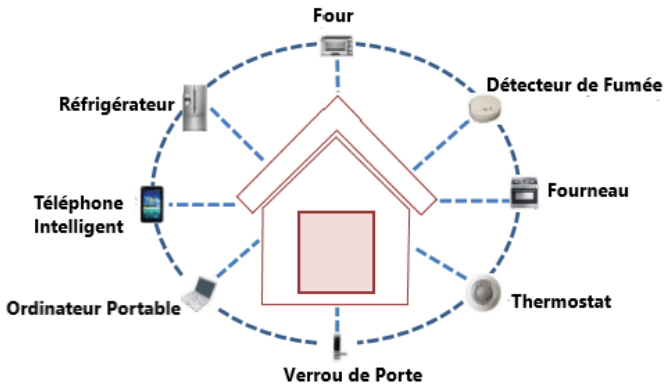

Figure 2.2 – Exemple d'une maison intelligente avec des objets liés. par Hossain et al. [28]

La sécurité d'un système informatique a pour but de protéger les ressources contre la divulgation, toute modification ou utilisation non autorisée, tout en empêchant l'accès illégitime des utilisateurs à ces dernières.

La sécurité est souvent définie comme étant la combinaison de trois principales propriétés:

- *L'intégrité:* permet d'assurer que les données n'ont pas été altérées par un utilisateur non légitime.
- *La confidentialité:* permet d'empêcher toute divulgation non autorisée de l'information et assurer que seules les personnes légitimes ont accès aux ressources protégées.
- *La disponibilité:* permet d'empêcher un déni d'accès non autorisé à l'information lorsqu'un utilisateur l'a demande.

Selon Sha et al. [61], le large déploiement des dispositifs IoT à grande échelle avec des ressources limitées posent de nombreux nouveaux défis importants à la conception des systèmes IoT flexibles et fiables. La sécurité est l'un des problèmes les plus cruciaux à prendre en compte pour l'adoption à grande échelle des systèmes IoT.

Une étude a été présentée par Sun et al. [63] a montré que les systèmes Internet of

Vehicles (IoV) sont confrontés à des attaques qui peuvent nuire à la stabilité, à la robustesse et à la sécurité de ces derniers. Les exigences de sécurité dans les systèmes IoV ont été ensuite présentées. Les défis et les problèmes de sécurité dans l'IoT ont été présentés par Hossain et al. [28] à l'aide d'une analyse détaillée, cette analyse porte sur les principaux problèmes concernant la gestion des clés par les systèmes permettant la maintenance et la distribution des clés entre les noeuds approuvés, la gestion de la confiance qui peut être une tâche difficile à mettre en oeuvre dans un réseau d'appareils avec des ressources limitées ainsi que les défis concernant la sécurité de bout en bout et la tolérance aux fautes. Les auteurs ont introduit ensuite les exigences de sécurité et une analyse des vulnérabilités des dispositifs IoT dans le but de combler ces lacunes en tenant compte des limitations de ces dispositifs concernant la multiplicité, la mobilité et l'évolutivité des dispositifs IoT ainsi que la contrainte énergétique et de stockage, étant donné que la plupart des dispositifs IoT utilisent des processeurs de faible puissance alors que les algorithmes cryptographiques sont très coûteux en calcul.

De nombreuses recherches sont consacrées à étudier les défis critiques dans l'Internet des Objets et plus particulièrement le défi de sécurité et de proposer ensuite des solutions et des approches adéquates afin de résoudre ce défi. Nous allons nous focaliser sur des solutions qui portent sur l'identification, l'authentification et le contrôle d'accès.

2.4 L'IDENTIFICATION

Les périphériques IoT doivent être identifiés de manière unique afin de pouvoir communiquer les uns avec les autres. Selon Fernández-Caramés et al. [23], les dispositifs IoT doivent être identifiés d'une manière ou d'une autre pour déterminer l'origine des données et pour détecter automatiquement les éléments qui nous entourent. Dans le domaine de l'identification dans l'Internet des objets (IoT), plusieurs recherches connexes ont été menées dans la littérature.

Les protocoles d'identification utilisés pour identifier les périphériques IoT ont été présentés par Salman [57], IPV4 qui se compose de quatre octets fournissant des adresses de 2^{32} pour identifier environ 4,3 milliards de dispositifs. Cependant, le nombre de dispositifs IoT ne cesse d'augmenter ce qui nécessite un grand nombre d'adresses. Par conséquent, IPV4 a atteint sa limite et ne peut plus satisfaire les besoins futurs. Pour cela, IPV6 a été utilisé en tant que successeur de IPV4 fournissant des adresses de 2^{128}, ce nombre est immense et suffisant pour identifier un nombre illimité de dispositifs IoT.

Une autre approche d'identification a été introduite par LV et al. [40]. Le système d'identification unique des matériaux peut localiser le statut en temps réel des matériaux et suivre leur cycle de vie afin de parvenir à une meilleure gestion. La procédure d'identification de ce système comprend: l'attribution et le marquage de code afin de garantir l'unicité du code d'identification attribué, ce code est l'identifiant unique des matériaux et leurs informations, la procédure d'identification comprend ainsi le code d'audit, l'utilisation du code et le code d'annulation. Cependant, la construction d'un tel système d'identification est extrêmement complexe et qui couvre un très large éventail d'organisations de gestion à construire, ainsi que la structure et l'interface de données des systèmes IoT existants doivent être ajustés pour répondre aux exigences de ce nouveau système.

Selon Fernández-Caramés et al. [23], la technologie d'identification par radiofréquence (RFID) est l'une des meilleures technologies positionnées afin d'effectuer l'identification, et qui a acquis une grande popularité ces dernières années dans des applications telles que le contrôle d'accès, les cartes de paiement ou la logistique et les soins de santé. Ces auteurs ont aussi présenté les différentes failles et les attaques de sécurité dans les systèmes IoT basés sur RFID. Après avoir analysé les outils de sécurité RFID existants, ils ont proposé une nouvelle méthodologie qui détecte et réduit les failles de sécurité. Cette méthodologie a été validée en l'appliquant par l'outil Promark 3.

La technologie d'identification par radiofréquence (RFID) a été utilisée par Yu et al. [67] pour la conception d'un système qui permet d'effectuer une identification automatique des véhicules à l'aide des étiquettes électroniques, les lecteurs, les antennes, un réseau à deux couches et une station et un logiciel de surveillance. Dans cette approche, les informations des lecteurs et des antennes qui sont transmettent dans le réseau seront utilisées par le système dans l'identification automatique des véhicules à l'aide des étiquettes électroniques. Cependant, le problème de sécurité dans la technologie RFID n'a pas été géré dans de nombreuses applications. Selon Borgohain et al. [15], cette technologie est confrontée à diverses attaques de l'extérieur en raison de la sécurité défectueuse.

Une autre approche d'identification a été présentée par Meidan et al. [44] a démontré la possibilité d'identifier les dispositifs IoT en se basant principalement sur les caractéristiques du trafic réseau qu'il génère telles que les signaux et les émissions afin de déterminer si le trafic appartient à un PC, un smartphone ou à un appareil IoT spécifique. Cela se fait par une analyse et une classification des données de ce dernier. Cette approche permet de détecter automatiquement et précisément toute connexion au réseau et permet aussi de réduire le trafic malveillant, ce qui la rend générique et flexible.

Une technique d'identification des types de périphériques a été introduite par Ke Gao

et al. [36], dont les types de périphériques sont identifiés en se basant essentiellement sur une analyse en ondelettes et le suivi des paquets entrants et sortants. Cependant, cette technique ne peut pas être appliquée aux périphériques IoT de point final.

GTID, est une autre technique présentée par Radhakrishnan et al. [54], permettant d'identifier les types des périphériques en utilisant des observations filaires dans un réseau local. La classification des périphériques est effectuée à l'aide des réseaux de neurones artificiels (ANN) qui permettent la création des classifications rapides dont chaque neurone reçoit des informations numériques en provenance de neurones voisins, cette classification utilise aussi les techniques statistiques pour la création de la signature unique du périphérique et son type. Cependant, les périphériques d'évaluation génèrent peu de trafic ce qui limite la flexibilité de cette approche.

Afin de combler les limitations des systèmes d'identification des types de périphériques précédents, les types de périphériques peuvent être automatiquement identifiés lorsqu'ils accèdent au réseau à l'aide d'une sentinelle IoT présentée par Miettinen et al. [46]. L'objectif principal de ce système est d'évaluer la vulnérabilité des dispositifs. En fonction de cette évaluation, des règles de filtrage du trafic sont appliquées afin d'isoler les dispositifs vulnérables. Cela permet de réduire la possibilité de compromettre des périphériques vulnérables et de tirer parti de cette dernière par un attaquant afin d'accéder à d'autres périphériques du réseau.

Le prototype d'implémentation de IOT SENTINEL a été présenté par Miettinen et al. [45]. Ce système comprend une passerelle de sécurité et un service de sécurité IoT qui communiquent via HTTP/HTTPS à l'aide d'une API. L'approche utilisée par ce système est totalement différente, puisqu'elle vise à identifier de manière pro-active les périphériques vulnérables et à appliquer les contre-mesures appropriées avant même que les vulnérabilités de sécurité ne soient exploitées.

2.5 L'AUTHENTIFICATION

Dans le domaine d'authentification dans l'Internet des objets (IoT) plusieurs recherches connexes ont été menées dans la littérature afin de protéger les dispositifs IoT contre les attaques malveillantes.

Les auteurs Ahamed et al. [4] ont présenté un protocole d'authentification sécurisé basé sur l'ECC (elliptic curve cryptography), ainsi qu'une analyse de sécurité de ce protocole a été effectuée afin de montrer l'efficacité de ce dernier. L'inconvénient principal de ce protocole d'authentification est qu'il n'est pas résistant aux attaques par déni de service (DoS).

Zhao et al. [71] ont proposé un schéma d'authentification mutuelle asymétrique avec un faible coût de communication et de calcul. La possibilité d'utiliser ce schéma pour les applications IoT a été démontrée par une analyse de sécurité. Une nouvelle approche d'authentification par cluster adaptée à l'IoT a été proposée par Venkatraman and Agrawal [65]. Cette approche est basée sur le protocole TCP et une architecture hiérarchique, avec un coût de traitement réduit. Cependant, un attaquant peut obtenir les paires de clé système et la clé du cluster, ainsi que la génération de nombres aléatoires et de signatures crée une surcharge de travail avec une consommation considérable des ressources mémoire.

Un cadre souple d'authentification et d'autorisation pour l'IoT a été présenté par Sciancalepore et al. [59]. L'élément clé de ce dernier est la passerelle qui collecte les informations produites, contrôle les demandes d'accès provenant des applications tierces, prend en charge une variété de formats de jetons, afin de mieux gérer l'authentification et l'autorisation des applications et elle met en cache les données extraites pour répondre de manière opportuniste aux futures demandes. Cependant, cette approche doit être testée et évaluée dans différents scénarios pour assurer l'efficacité de ce cadre d'authentification

et d'autorisation.

Mahalle et al. [42] ont présenté un protocole d'authentification et de contrôle d'accès efficace ainsi qu'un protocole d'authentification mutuelle. L'analyse effectuée sur le schéma IACAC a montré l'efficacité de ce protocole en matière de temps de calcul. L'analyse de sa performance dans des scénarios IoT a également montré sa capacité de sécuriser les données IoT en évitant les attaques de type homme au milieu et l'attaque par déni de service (DoS). Cependant, ce schéma ne convient toujours pas parfaitement aux petits dispositifs IoT.

2.6 LE CONTRÔLE D'ACCÈS

Dans l'Internet des Objets (IoT) toute lacune dans la sécurité pouvant nuire à la fois à l'utilisateur et à la réputation des entreprises fournissant ces services.

Le contrôle d'accès est un mécanisme et l'un des aspects essentiels de sécurité qui permet de contrôler l'accès aux ressources protégées en limitant l'accès à ces dernières en fonction des stratégies d'accès spécifiques. Afin d'empêcher l'accès illégitime aux données des périphériques IoT, la présence d'un mécanisme de contrôle d'accès est incontournable.

Avec les défis de sécurité critiques existants et leurs impacts malveillants sur les systèmes IoT, l'utilisation d'un mécanisme de contrôle d'accès devient une solution primordiale.

Selon Margheri et al. [43], les systèmes de contrôle d'accès sont des moyens largement utilisés pour la protection des systèmes informatiques. Ils sont définis en termes de politiques de contrôle d'accès réglementant l'accès aux ressources du système.

Le contrôle d'accès a été défini par Narouei et al. [51] en tant que *"Composant essentiel de toute organisation et essentiel pour les systèmes informatiques de sécurité. L'accès à*

ces systèmes et la manipulation de leurs données sont contrôlés de manière appropriée en fonction des niveaux de classification des données décrits dans les politiques de contrôle d'accès."

Autrement, par Biswas et al. [14] *"Le contrôle d'accès a été un élément majeur dans l'application des exigences en matière de sécurité et de confidentialité des informations et des ressources en matière d'accès non autorisé."*

2.6.1 MODÈLES DE CONTRÔLE D'ACCÈS

Différents modèles de contrôle d'accès ont été proposés dans la littérature et un grand nombre de chercheurs ont contribué au développement de ce mécanisme afin de renforcer et d'assurer la sécurité des données IoT.

Les modèles de contrôle d'accès selon Ausanka-crues [7] concernent généralement une entité qui est capable de manipuler des informations et peut accéder à des objets, entités à travers lesquelles les informations transitent par les actions d'un sujet et comment cet accès peut se produire. Les modèles de contrôle d'accès sont généralement considérés comme des frameworks pour la mise en oeuvre et la garantie de l'intégrité des stratégies de sécurité qui déterminent la manière dont les informations peuvent être accessibles et partagées sur le système.

Chaque modèle de contrôle d'accès suit une politique d'accès spécifique. Généralement, la politique de contrôle d'accès est définie comme étant des règles qui gèrent l'accès et permettent d'exprimer qui a la permission d'appliquer quoi sur quelle ressource. Donc, nous pouvons considérer le modèle de contrôle d'accès étant une représentation formelle de la politique de sécurité et de son fonctionnement.

Il existe quatre grandes classes de modèles de contrôle d'accès:

1. Le modèle de contrôle d'accès discrétionnaire ou DAC (Discretionary Access Control).
2. Le modèle de contrôle d'accès obligatoire ou MAC (Mandatory Access Control).
3. Le modèle de contrôle d'accès basé sur le rôle ou RBAC (Role-Based Access Control).
4. Le modèle de contrôle d'accès basé sur les attributs ou ABAC (Attribute-Based Access Control).

Nous allons présenter dans les sections suivantes chacun de ces modèles de contrôle d'accès et introduire leurs principes. Nous préciserons par la suite leurs limitations lorsqu'ils n'arrivent pas à satisfaire les exigences en termes de sécurité et de protection des environnements informatiques, puis nous allons illustrer chaque modèle avec un exemple.

2.6.2 MODÈLE DE CONTRÔLE D'ACCÈS DISCRÉTIONNAIRE (DAC)

Le modèle de contrôle d'accès discrétionnaire ou DAC (Discretionary Access Control) est l'un des modèles de contrôle d'accès les plus répandus. Il limite l'accès aux ressources en se basant sur l'identité des utilisateurs et les droits d'accès accordés à ces derniers.

La politique de contrôle d'accès utilisée dans le DAC est sous forme d'un triplet (objet, sujet, opération), dont le sujet, l'objet et l'opération sont définis dans chaque entrée afin de spécifier les opérations autorisées à effectuer (e.g, lire, écrire ou exécuter) par les sujets sur les objets du système. Ce modèle suppose que les utilisateurs sont les propriétaires des ressources et peuvent octroyer et transférer les droits d'accès sur ces dernières.

Ausanka-crues [7]: *"DAC a été développé pour implémenter les matrices de contrôle d'accès définies par Lampson dans son article sur la protection du système. Les matrices de contrôle d'accès sont généralement représentées sous forme de matrices tridimensionnelles où les lignes sont des sujets, les colonnes sont des objets et le mappage des paires sujet/objet produit l'ensemble des droits du sujet sur l'objet".*

2.6.2.1 Les limitations du modèle de contrôle d'accès discrétionnaire

Selon Ausanka-crues [7], le fait de donner la possibilité aux utilisateurs de contrôler les autorisations d'accès aux objets a pour effets secondaires d'ouvrir le système à la susceptibilité des chevaux de Troie. De plus, la maintenance du système et la vérification des principes de sécurité sont extrêmement difficiles pour les systèmes DAC, puisque les utilisateurs contrôlent les droits d'accès aux objets détenus. Ainsi que la gestion d'un tel système est difficile.

Dans le modèle discrétionnaire la politique d'accès utilisée pour limiter l'accès aux ressources du système est basée uniquement sur l'identité de l'utilisateur. Donc, les utilisateurs peuvent contrôler les autorisations d'accès et octroyer des privilèges sur les ressources qu'ils possèdent. Il n'est pas compliqué de contourner les restrictions d'accès dans un tel système, cette limitation rend le modèle discrétionnaire vulnérable aux chevaux de Troie.

Nous allons illustrer ces limitations dans les deux exemples suivants.

Considérons trois utilisateurs : Alice, Bob et Katy. Supposons que Alice a créé un fichier A et l'a rendu accessible à Bob en lui octroyant le privilège de lire dans ce fichier. Le contrôle d'accès est défini dans la Figure 2.3, comme Bob a le droit d'écrire et Katy a le droit de lire dans le fichier B, Bob peut contourner la restriction d'accès de Katy au fichier A en copiant A dans B.

	Fichier A	Fichier B
Alice	Lire / Écrire	
Bob	Lire	Écrire
Katy		Lire

Figure 2.3 – Exemple sur le contrôle d'accès discrétionnaire (DAC)

Nous allons prendre un deuxième exemple qui est illustré dans la Figure 2.4 afin de mieux cerner le problème de Cheval de Troie qui peut amener à une fuite d'information vers des utilisateurs non autorisés.

Supposons que Bob est un directeur d'une entreprise nationale de forage a créé un fichier (ForageTech) contenant des informations critiques sur les nouvelles techniques du forage et les puits qui vont être utilisés dans leur prochaine extraction. Alice l'assistante de Bob désire accéder à ces informations non autorisées. Cette dernière a créé un fichier (ForageVertical), puis elle a octroyé à Bob le droit d'écrire dans ce fichier à travers une application malveillante contenant deux opérations cachées (lire dans le fichier ForageTech et écrire dans le fichier ForageVertical). Alors, lorsque Bob exécute cette application, les deux opérations vont être permises et Alice pourra accéder à ce fichier et récupérer les informations désirées puisqu'elle est la propriétaire du fichier ForageVertical.

2.6.3 MODÈLE DE CONTRÔLE D'ACCÈS OBLIGATOIRE (MAC)

Afin de combler les limitations du modèle de contrôle d'accès discrétionnaire, le modèle de contrôle d'accès obligatoire ou MAC (Mandatory Access Control) a été développé, il est plus rigide que le modèle discrétionnaire. Dans ce modèle, les utilisateurs ne peuvent pas intervenir dans l'attribution des droits d'accès. Le modèle multiniveaux attribue aux sujets et aux objets du système des niveaux de sécurité non modifiables par les

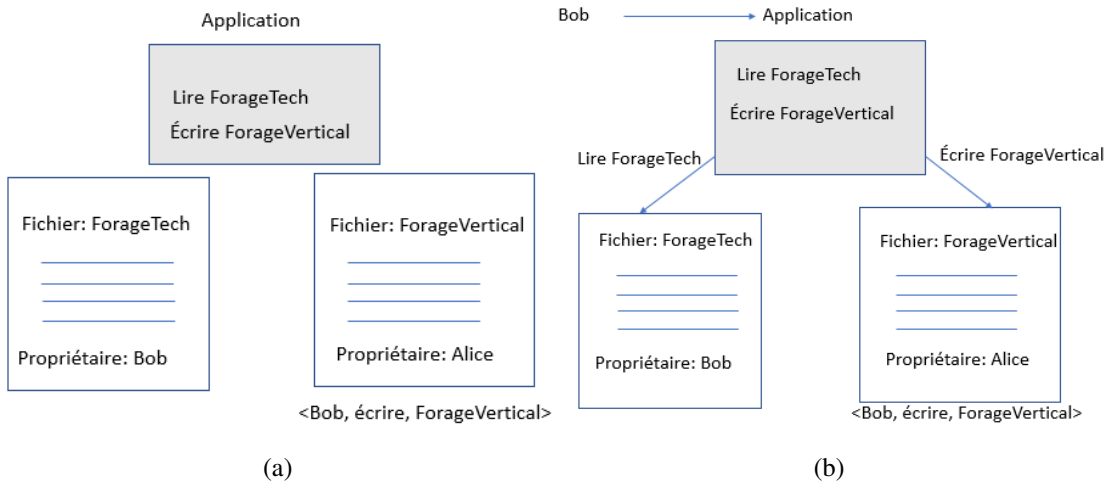

Figure 2.4 – Exemple sur le problème de Cheval de Troie, adapté de Saïda [56]

utilisateurs. Donc, ce modèle permet de résoudre les problèmes des chevaux de Troie et de fuite d'information.

Ausanka-crues [7] définit le contrôle d'accès obligatoire (MAC) comme étant: *"un modèle de contrôle d'accès appliquant des politiques de sécurité indépendantes des opérations de l'utilisateur, le contrôle d'accès obligatoire est généralement associé au modèle de BellLaPadula Bell and La Padula [8] à plusieurs niveaux de sécurité."*

Le modèle de Bell et LaPudula Bell and La Padula [8] a été développé comme premier modèle initialement pour le département de la défense américaine afin d'assurer la confidentialité de données. Ensuite, pour assurer l'intégrité, le modèle de Biba a été développé Biba [13]. En outre, d'autres modèles obligatoires, moins formalisés, tel que le Modèle Clark-Wilson qui a été créé pour les systèmes commerciaux D. Clark and R. Wilson [19].

Dans le modèle de Bell et LaPudula, un niveau de sécurité est accordé à chaque utilisateur et à chaque objet du système. Le niveau de sécurité montre la confiance offerte par le système à cet utilisateur pour protéger l'information et à la sensibilité des informations contenues dans cet objet.

Quatre niveaux de sécurité sont classifiés et ordonnés comme suit: TS (Top Secret) > S (Secret) > C (Confidential) > U (Unclassified).

Dans ce modèle, pour assurer la confidentialité des informations, des étiquettes de sécurité sont attribuées aux sujets et aux objets et deux propriétés de sécurité sont utilisées:

- La propriété simple (no read up): un utilisateur ne peut accéder à des informations de l'objet seulement si son niveau de sécurité domine celui de l'objet.
- La propriété étoile (no write down): un utilisateur ne peut écrire dans un objet seulement si le niveau de sécurité de l'objet domine le niveau de sécurité de cet utilisateur.

Nous allons montrer dans l'exemple suivant qui est lié à celui de Cheval de Troie cité précédemment dans la Figure 2.4 comment l'application de ces deux propriétés de sécurité empêche la fuite des informations non autorisées.

Supposons que Bob est le propriétaire du fichier ForageTech qui contient des informations critiques et Alice est la propriétaire du fichier ForageVertical. Supposons qu'on a attribué la classification ci-dessous à ces utilisateurs et à leurs fichiers:

- *"Top Secret"* pour Bob et pour le fichier ForageTech.
- *"Confidential"* pour Alice et pour le fichier ForageVertical.

Selon cette classification, si Bob se connecte au système comme un utilisateur *"Top Secret"*, l'application s'exécute avec une classe d'accès Top Secret, l'opération d'écriture dans le fichier ForageVertical avec le niveau de sécurité Confidential sera bloquée, puisque le niveau de sécurité de Bob domine celui du fichier ForageVertical (propriété étoile). Si Alice invoque l'application avec un niveau de sécurité *"Confidential"*,

l'opération de lecture du fichier ForageTech sera bloquée (propriété simple). Par conséquent, l'application de ces deux propriétés de sécurité empêche le problème de Cheval de Troie.

2.6.3.1 Les limitations du modèle de contrôle d'accès obligatoire

Le modèle MAC selon Ausanka-crues [7] n'est pas sans limitations puisque l'attribution et l'application des niveaux de sécurité par le système imposent des restrictions aux actions des utilisateurs, tout en respectant les stratégies de sécurité, empêchent toute modification dynamique des stratégies sous-jacentes et obligent de grandes parties du système d'exploitation et des utilitaires associés à être "dignes de confiance" et placés en dehors du cadre de contrôle d'accès.

Bien que le modèle MAC résout les problèmes des chevaux de Troie et de fuite d'information du modèle DAC en attribuant des niveaux de sécurité aux objets et aux sujets, en limitant l'accès à tous les niveaux de sécurité attribués à ces sujets et objets dans le système. Cependant, ce dernier n'est pas adapté aux systèmes répartis et il est difficile et coûteux à mettre en oeuvre, ainsi que la nécessité de réécrire les applications pour qu'elles adhèrent aux étiquettes et aux propriétés MAC. Pour cela, le concept de rôle a été introduit afin d'exprimer un large éventail de politiques d'accès. Nous allons présenter ce modèle dans la section suivante.

2.6.4 MODÈLE DE CONTRÔLE D'ACCÈS À BASE DE RÔLE (RBAC)

Dans ce modèle, le rôle est le concept de base, les droits d'accès sont associés à des rôles et ces derniers sont attribués aux utilisateurs. Les droits d'accès sont octroyés aux utilisateurs en fonction des rôles qu'ils jouent dans le système. Donc, deux utilisateurs avec des rôles similaires ont les mêmes droits sur le système.

Selon Sandhu et al. [58], le modèle RBAC est considéré comme un modèle beaucoup

plus général que le MAC ou le DAC, englobant les deux modèles en tant que cas particuliers tout en fournissant un cadre neutre en matière de stratégie permettant de personnaliser le RBAC pour chaque application.

En fait, les types du modèle RBAC sont présentés comme suit:

- $RBAC_0$: c'est le modèle de base qui contient les éléments de base du modèle RBAC (utilisateurs, rôles, permissions, sessions).
- $RBAC_1$: c'est une extension de $RBAC_0$ qui ajoute le concept de "hiérarchie de rôles" (le rôle peut hériter d'un autre rôle).
- $RBAC_2$: ajoute un ensemble de contraintes, par exemple des contraintes de temps et de lieu.
- $RBAC_3$: qui est la combinaison de $RBAC_1$ et $RBAC_2$.

Un exemple simplifié est illustré dans la Figure 2.5, dont les permissions "lire/écrire" sont attribuées aux rôles dans un système d'une entreprise nationale de forage. Les utilisateurs peuvent effectuer les opérations permises par les rôles grâce aux permissions (privilèges) auxquels ils sont associés. L'utilisateur établit une session durant laquelle il active un ensemble de ses rôles.

Prenons les permissions "lire/écrire" sur les dossiers de la gestion des stocks qui sont attribuées au rôle "Magasinier", dans cette entreprise un utilisateur avec le rôle "Magasinier" pourrait donc effectuer les opérations permises et associées à son rôle, dans ce cas cet utilisateur peut effectuer les opérations "lire/écrire" sur les dossiers de la gestion des stocks.

La Figure 2.6 montre que le rôle est une entité intermédiaire entre l'utilisateur et les permissions qui sont associées à travers ce dernier. Le rôle "Comptable" est une

entité intermédiaire entre l'utilisateur "Bob" et les permissions "lire/écrire" sur les dossiers budget et le salaire, ces permissions sont accociées à cet utilisateur à travers son rôle. Dans ce cas, le rôle "Comptable" qui est attribué à l'utilisateur "Bob" lui permet d'effectuer les opérations "lire/écrire" sur les dossiers budget et le salaire à travers les permissions associées à son rôle.

	Rapport-forage	**Dossier de la gestion des stocks**	**Dossier des budgets, le salaire**
Chef de service	Lire / Écrire	Lire / Écrire	Lire / Écrire
Comptable			Lire / Écrire
Magasinier		Lire / Écrire	

Figure 2.5 – Exemple simplifié du modèle RBAC

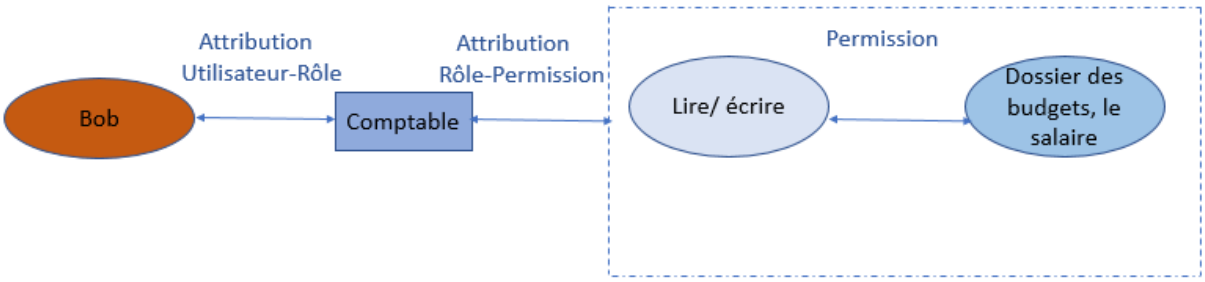

Figure 2.6 – Attribution des permissions dans RBAC

2.6.4.1 Les limitations du modèle de contrôle d'accès basé sur le rôle

Le concept de rôle introduit dans RBAC a permis d'exprimer un large éventail de contrôle d'accès. Cependant, dans ce modèle la gestion des rôles est une tâche complexe et difficile à maintenir. Selon Ausanka-crues [7], malgré que le modèle RBAC marque une avancée importante en matière de contrôle d'accès, les problèmes administratifs des grands systèmes persistent, même s'ils sont plus faciles à gérer. Dans les grands systèmes, les appartenances, l'héritage des rôles et la nécessité de disposer de privilèges personnalisés plus détaillés rendent l'administration potentiellement lourde.

En outre, différentes variantes de RBAC ont été proposées par Bertino et al. [9, 10], Xu and Stoller [66], Hu et al. [29], Cruz et al. [18], Adda et al. [2], Mitra et al. [47] afin de renforcer la sécurité à l'aide d'un mécanisme de contrôle d'accès plus flexible.

Feng et al. [22] ont proposé une extension du modèle RBAC basé sur le contexte et la confiance, TCAC est dédié aux systèmes ouverts et distribués. Ce modèle se base principalement sur les rôles, la confiance et le contexte du demandeur d'accès afin d'attribuer les rôles. Afin de fournir une autorisation d'accès, deux facteurs vont être évalués: la valeur de confiance de l'utilisateur qui doit être supérieur au seuil de confiance définie dans les politiques de sécurité et les contraintes du contexte qui doivent être satisfaites.

Zhang and Parashar [69] ont présenté un modèle de contrôle d'accès à base de rôle dynamique (DRBAC) qui étend le modèle RBAC, dans ce modèle le rôle actif de l'utilisateur et l'autorisation active du rôle peuvent être changés dynamiquement en se basant sur les informations du contexte. Donc, le privilège octroyé à un sujet pour accéder à une ressource sera modifié de manière dynamique lorsque le contexte change.

Moyer and Abamad [49] ont proposé une extension du modèle RBAC traditionnel, ce

nouveau modèle est nommé GRBAC (Generalized role-based access control). GRBAC permet d'améliorer le modèle RBAC traditionnel en appliquant des rôles à toutes les entités du système. Donc, des rôles sont attribués aux sujets, aux contextes et aux objets du système. Les décisions d'accès dans GRBAC ne dépendent pas seulement des rôles du sujet comme c'est le cas dans le modèle RBAC traditionnel, mais dépend aussi des rôles de l'objet et des rôles du contexte.

Bien que GRBAC exprime un contrôle d'accès plus dynamique avec l'utilisation des trois types de rôles qui offrent plus de flexibilité. Cependant, GRBAC n'est pas une solution de sécurité complète à utiliser dans les systèmes à grande échelle car ce modèle est plus complexe que le modèle RBAC et difficile à maintenir.

2.6.5 ARCHITECTURE XACML

XACML (eXtensible Access Control Markup Language), il s'agit de la norme OASIS disponible depuis 2003, qui est un standard de langage basé sur XML (Extensible Markup Language) conçu pour exprimer les politiques de sécurité basées sur le concept de contrôle d'accès basé sur les attributs (ABAC) dont les décisions d'accès sont basées sur les attributs des sujets, des objets et des contextes.

Selon Ferraiolo et al. [26], le langage de balisage de contrôle d'accès extensible (XACML) et le contrôle d'accès de nouvelle génération (NGAC) sont des normes de contrôle d'accès basées sur des attributs très différents mais avec des buts et des objectifs similaires. L'un des objectifs de ces deux normes est de fournir un moyen normalisé pour exprimer et appliquer des politiques de contrôle d'accès très diverses à l'appui de divers types de services de données.

Selon Bhatt et al. [12], NGAC fournit un meilleur support pour la gestion des attributs et des politiques, la révision administrative et la découverte des ressources alors que

XACML est capable d'exprimer un ensemble complexe et riche de règles de contrôle d'accès, y compris des valeurs d'attributs négatives et entières qui sont difficiles à représenter dans NGAC sans utiliser les relations d'interdiction et d'obligation.

L'architecture XACML définit différents modules qui sont utilisés dans le processus d'autorisation. Nous introduisons ci-dessous les modules de base de l'architecture XACML qui sont illustrés dans la Figure 2.7.

Lorsque un sujet envoie une demande d'accès:

- Le **Policy Enforcement Point (PEP)** transforme la demande d'accès de ce sujet en demande XACML et envoie cette demande au PDP. Le PEP autorise ou refuse l'accès à la ressource demandée après avoir fourni la décision d'accès par les autres modules.

- Le **Policy Decision Point (PDP)** après la réception de la demande XACML, interroge les politiques stockées dans le PRP. Si les attributs de la demande ne sont pas suffisants pour l'évaluation des règles et des politiques, le PDP peut demander au gestionnaire de contexte (qui est responsable de la communication et de la traduction des messages) de rechercher des attributs supplémentaires dans le PIP qui lui fournit les informations nécessaires afin de prendre la décision d'accès. Le PDP ensuite envoie la décision d'accès au PEP à l'aide d'un langage de demande/réponse.

- Le **Policy Retrieval Point (PRP)** il peut s'agir d'une base de données qui permet de stocker les politiques XACML.

- Le **Policy Administration Point (PAP)** ce module fait une gestion de la base de données(l'ajout, le retrait de politique).

- Le **Policy Information Point (PIP)** ce module fournit les informations concernant

les attributs de l'objet, sujet et les conditions de l'environnement pour le PDP afin de prendre la décision d'accès.

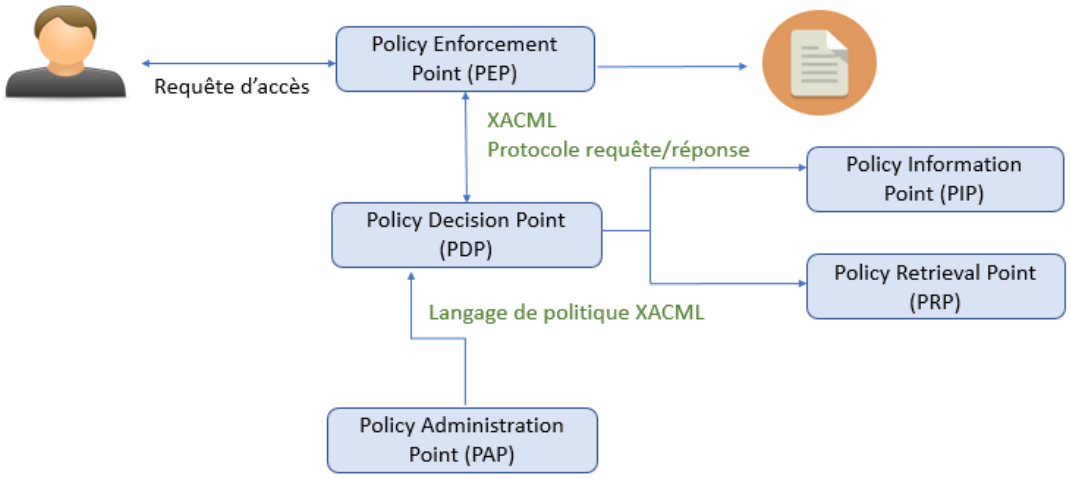

Figure 2.7 – Architecture XACML, adapté de Srijith [62]

2.6.6 LE CONTRÔLE D'ACCÈS DE NOUVELLE GÉNÉRATION (NGAC)

Next Generation Access Control (NGAC) est une norme de contrôle d'accès basée sur les attributs qui utilise les termes utilisateur, opération et objet ainsi qu'elle inclut des processus, des opérations administratives et des classes de politique.

Selon Ferraiolo et al. [25], NGAC adopte une approche fondamentalement différente de XACML pour représenter les demandes, exprimer et administrer les politiques, représenter, administrer les attributs et appliquer les décisions. NGAC est défini en termes d'un ensemble normalisé de relations et de fonctions qui sont réutilisables dans l'expression et l'application des politiques. Cette norme n'exprime pas les politiques d contrôle d'accès par le biais de règles, mais plutôt par le biais de configurations de relations de quatre types: l'affectation (permet de définir l'appartenance dans des conteneurs), l'association (pour dériver des privilèges), l'interdiction (pour exprimer les contraintes et les restrictions des droits d'accès) et l'obligation (pour modifier

dynamiquement l'état d'accès).

Il est à noter que NGAC considère les utilisateurs et les processus comme des entités indépendantes mais qui sont liées. Les processus par lesquels un utilisateur tente d'accéder prennent les mêmes attributs que l'utilisateur. Cette norme reconnaît deux types d'opérations, les opérations sur les ressources (lecture, écriture, etc.) et les opérations administratives permettant de configurer les données de contrôle d'accès.

Lorsqu'un sujet envoie une demande d'accès, les différents modules de l'architecture NGAC qui est illustrée dans la Figure 2.8 seront utilisés dans le processus d'autorisation afin de fournir une décision d'accès.

- Le PEP et après la réception de la demande d'accès soumet la demande au PDP.
- Le PDP calcule via le PAP la décision d'accès basée sur la configuration actuelle des données et des relations stockées dans le PIP. Puis, le PDP renvoie la décision d'accès au PEP.
  - Si l'accès a été accordé et qu'il s'agit d'une opération de "lecture/écriture", le PDP renvoie également l'emplacement physique de l'objet, le PEP envoie une commande au RAP pour exécuter l'opération sur le contenu car les routines de lecture/écriture sont implémentées dans le RAP.
  - Si l'accès a été accordé et qu'il s'agit d'une opération administrative, le PDP émet une commande au PAP (car les routines administratives sont implémentées dans le PAP) pour l'exécution de l'opération sur l'élément ou la relation stockée dans le PIP. Le PAP renvoie le statut au PDP, le PDP à son tour retourne le statut au PEP.
  - Si le statut renvoyé par le PAP ou le RAP est «réussi», le PEP soumet le contexte de l'accès au point de traitement d'événement (EPP). Si le

contexte correspond à un modèle d'événement d'une obligation, l'EPP exécute automatiquement les opérations administratives de cette obligation.

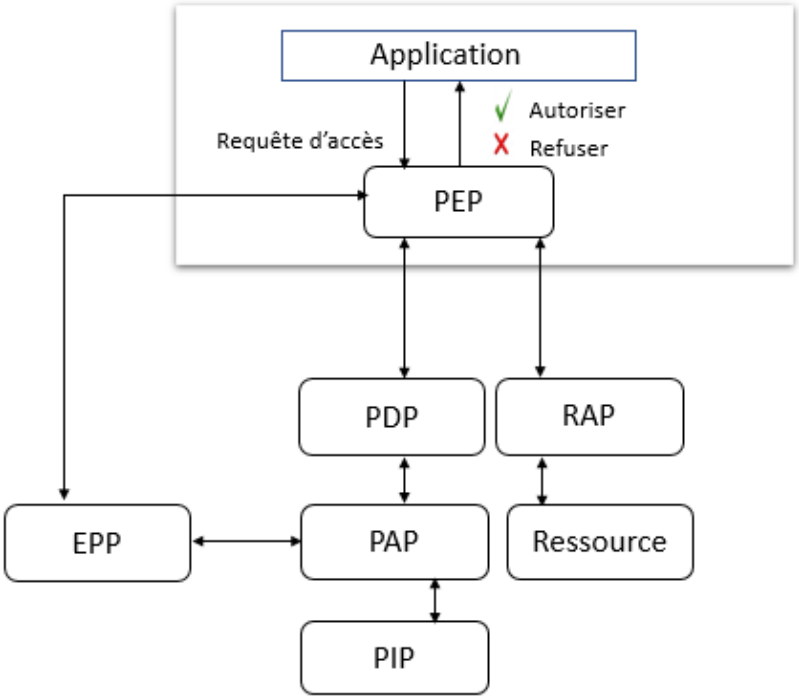

Figure 2.8 – Architecture NGAC, adapté de Ferraiolo et al. [25]

2.6.7 LE MODÈLE DE CONTRÔLE D'ACCÈS BASÉ SUR LES ATTRIBUTS (ABAC) ET SES EXTENSIONS

Les limitations des modèles précédents ont entraîné le passage vers un nouveau modèle de contrôle d'accès basé sur les attributs ou ABAC (Attribute-Based Access Control). Kuhn et al. [37], Jin et al. [33], Hu et al. [30].

Servos and Osborn [60] définit le modèle de contrôle d'accès basé sur les attributs comme : *"une alternative prometteuse aux modèles traditionnels de contrôle d'accès (c'est-à-dire le contrôle d'accès discrétionnaire (DAC), le contrôle d'accès obligatoire (MAC) et le contrôle d'accès basé sur les rôles (RBAC)) qui attire l'attention à la fois*

dans la littérature académique récente et dans les applications de l'industrie".

Le modèle de contrôle d'accès basé sur les attributs (ABAC) est l'un des modèles et des normes qui ont reçu une attention significative au cours de ces dernières années. Dans ce dernier les règles d'accès sont spécifiées à l'aide d'un ensemble d'attributs provenant principalement des utilisateurs, des objets et de l'environnement. La demande d'accès peut être acceptée ou refusée après avoir fourni une décision d'accès en évaluant les attributs et les règles de contrôle d'accès.

- Le sujet: c'est l'entité qui demande l'accès afin d'effectuer des opérations sur des objets, les opérations pourraient être (lecture, modification, suppression, exécution, etc).

- L'objet: c'est une ressource système pour laquelle l'accès doit être contrôlé et géré, tels que des périphériques, des fichiers, des enregistrements, des programmes, etc. Il s'agit de la ressource demandée par un sujet.

- Conditions environnementales: il s'agit du contexte opérationnel dans lequel se produisent les demandes d'accès des utilisateurs. Elles peuvent inclure l'heure actuelle, le jour de la semaine, l'emplacement d'un utilisateur ou le niveau de menace actuel.

- Politique d'accès: c'est la représentation de règles qui permet de déterminer si une requête d'accès doit être autorisée ou refusée, étant donné les valeurs des attributs des sujets, des objets et des conditions environnementales.

ABAC définit les entités en termes d'attributs (qui sont des caractéristiques du sujet, de l'objet et des conditions d'environnement), ce qui aide à définir les autorisations d'accès aux sujets.

Lorsqu'un sujet demande l'accès à une ressource, ABAC vérifie si ses attributs sont conformes à la stratégie d'accès définie pour la ressource. Si les attributs de sujet satisfont à la stratégie, l'autorisation d'accès à cette ressource (objet) est accordée. Une autorisation d'accès peut être changée en fonction des modifications apportées aux attributs du sujet.

Nous allons présenter un exemple détaillé sur la politique de contrôle d'accès basée sur les attributs qui permet de montrer comment le modèle ABAC assure la sécurité des données.

Supposons que dans une entreprise nationale de forage, à 10h Bob demande d'accéder à la paye et au fichier de gestion des congés des employés de cette entreprise.

Après sa demande. Le Policy Enforcement Point (PEP) informe le Policy Decision Point (PDP) que Bob demande d'accéder à la paye et au fichier de gestion des congés de tous les employés de cette entreprise à 10h. Ensuite, le PDP interroge le Policy Information Point (PIP) qui lui fournit toutes les informations nécessaires afin de prendre la décision d'accès. Supposons que nous avons la règle d'accès "Autoriser aux comptables d'accéder à la paye des employés et au fichier de gestion des congés pendant les heures du travail"). Dans la base des attributs nous avons:

- Bob est un comptable.
- Accéder à la paye et au fichier de gestion des congés est un accès.
- Le travail se termine à 19h.
- L'heure est à 10h<19h.

Puisque le demandeur (Bob) est un comptable et sa demande est faite pendant l'heure de travail, ainsi qu'il y a une règle qui autorise aux comptables d'accéder à la paye des

employés et au fichier de gestion des congés pendant les heures du travail. Le PDP selon ces informations déduit que la demande d'accès doit être autorisée, alors ce dernier informe le PEP pour donner l'autorisation à Bob d'accéder aux ressources demandées.

2.6.8 PASSAGE AU MODÈLE DE CONTRÔLE D'ACCÈS BASÉ SUR LES ATTRIBUTS (ABAC)

Comme cité dans les sections précédentes, le concept de rôle introduit dans le modèle RBAC a permis d'exprimer un large éventail de politiques d'accès. Cependant, ce modèle a rapidement montré ses limitations avec les exigences des systèmes en matière de sécurité car la gestion des rôles est considérée comme une tâche complexe et difficile à gérer et à maintenir. Le passage vers le modèle de contrôle d'accès basé sur les attributs (ABAC) avait pour objectif de surmonter cette limitation et offrir plus de précision aux décisions d'accès.

Nous allons montrer dans le tableau 2.1 une comparaison des principales fonctionnalités des modèles RBAC et ABAC.

Problèmes	RBAC	ABAC
Accord global	Non	Oui
Flexibilité	Non	Oui
Simplicité	Oui	Non
Décision d'autorisation	Localement	Globalement
Granularité	Faible	Élevé
Confiance	Localement	Globalement
Révocation	Non	Oui
Modification des privilèges	Complexe	Simple
Problème d'explosion de rôle	Oui	Non

Tableau 2.1 – Comparaison des fonctionnalités de RBAC et ABAC, adapté de [3]

Récemment, différentes variantes du modèle ABAC ont été proposées afin de répondre à des besoins spécifiques.

Par exemple, les modèles ABAC orientés sur le workflow et l'organisation sont respectivement présentés par Zhang and Liu [70], Kalam et al. [34]. Le Workflow-oriented Attributed Based Access Control (WABAC) selon ses auteurs, est une extension du modèle ABAC qui permet de prendre en compte le comportement de l'utilisateur et l'évaluation des risques afin d'améliorer le niveau de contrôle d'accès dans un environnement IoT. Cependant, ce modèle doit être testé et validé dans un environement réel pour évaluer son efficacité et sa performance. Ainsi qu'une version étendue du modèle orbac, plus adaptée aux organisations multiples a été proposée par Kalam et al. [34], Layouni and Pollet [38]. Cependant, cette approche ne traite pas les problèmes liés à l'intégrité et à la disponibilité de données.

Servos and Osborn [60] ont présenté un modèle ABAC hiérarchique, appelé HGABAC, dans lequel la notion de groupe a été ajoutée aux objets et aux sujets. Ces groupes peuvent également être organisés en hiérarchies. Des attributs peuvent être attribués à des objets, des sujets, des groupes de sujets et des groupes d'objets. Des groupes d'objets peuvent être attribués à des objets et des groupes de sujets peuvent être attribués à des sujets. Afin de simplifier l'héritage entre les groupes dans ce modèle, il est nécessaire que les attributs soient définis.

Ainsi qu'une version restreinte de ce modèle a été présentée par Bhatt et al. [12] qui porte sur le même concept de l'hiérarchie. Ce modèle propose une version restreinte de HGABAC comprenant des groupes d'utilisateurs et d'objets, des attributs de groupe et une hiérarchie de groupes. Les attributs de groupe sont des attributs attribués à des groupes d'utilisateurs et d'objets. Ces attributs de groupe représentent les objets ou les attributs appropriés des utilisateurs. Les objets / sujets sont "groupés" (à travers leurs attributs) si tous leurs attributs sont identiques. Cependant, ces modèles exigent que les

objets (et les objets) du même groupe doivent avoir les mêmes attributs. Ces restrictions limitent la flexibilité de ces extensions et les rendent inadéquats à l'IoT.

Biswas et al. [14] ont présenté un modèle de contrôle d'accès, nommé LaBAC, dans lequel les hiérarchies sont créées sur des valeurs d'attributs et non sur les attributs eux-mêmes. Ce modèle utilise un attribut utilisateur (uLabel) et un attribut objet (oLabel). Selon les auteurs de ce modèle, une stratégie d'autorisation dans LaBAC pour une action est une énumération utilisant les deux attributs (uLabel et oLabel). Donc, ce dernier peut être considéré comme une simple instance d'un modèle de stratégie énumérée existant - Policy Machine. Ainsi que, RBAC peut être considéré comme un modèle ABAC minimum. $LaBAC_H$ est une variante de LaBAC où les hiérarchies user-label et object-label sont introduites.

Le tableau 2.2 montre une comparaison des différentes extensions du modèle ABAC qui ont été proposées.

Comme le montre le tableau ci-dessous, Attribute-Based Access Control Model for Web Services (WS-ABAC) est un modèle ABAC dynamique et formel conçu pour les services Web basés sur XACML. Cependant, l'approche de ce modèle décrit principalement une architecture pour utiliser XACML et des politiques basées sur des attributs pour fournir une authentification pour les services Web uniquement. Une autre extension du modèle ABAC est le Workflow-oriented Attributed Based Access Control (WABAC) qui est un modèle dynamique et formel. Cependant, ce modèle n'est pas général pour l'adapter dans un système l'IoT. A Logic-based Framework for Attribute based Access Control, est un modèle largement axé sur la représentation et la performance des politiques basées sur les attributs et leur évaluation par rapport à la fourniture d'un modèle ABAC fonctionnel et général.

$ABAC_\alpha$ est un modèle ABAC généralisé et formalisé avec des contraintes pour les modèles traditionnels, il manque des composants qui seraient nécessaires à une implé-

mentation réelle, tels que les hiérarchies d'attributs et d'objets, un langage de politique simpliste et des attributs d'environnement. HGABAC, est un modèle hiérarchique dont la notion de groupe a été ajoutée aux objets et aux sujets qui peuvent être organisés en hiérarchies. Cependant, ce modèle est peu flexible et n'est pas adapté à l'IoT.

The access-control oriented (ACO) et Self-Adaptive access control model sont des modèles dédiés spécialement à l'IoT, dont le access-control oriented (ACO) est un modèle basé sur le modèle ABAC, ce modèle est dynamique et général, alors que le Self-Adaptive access control model est un autre modèle dédié à l'IoT qui présente une approche formelle et dynamique basée sur un middleware de contrôle d'accès qui vise à améliorer le niveau de contrôle d'accès dans les systèmes IoT.

Modèle	Dynamique	Général	Formel	Hiérarchique
Attribute-Based Access Control Model for Web Services (WS-ABAC)	Oui	Non	Oui	Non
Workflow-oriented Attributed Based Access Control (WABAC)	Oui	Non	Oui	Non
A Logic-based Framework for Attribute based Access Control	Oui	Non	Oui	Oui
$ABAC_{\alpha}$	Oui	Oui	Oui	Non
Hierarchical Group and Attribute-Based Access Control (HGABAC)	Oui	Non	Oui	Oui
The access-control oriented (ACO)	Oui	Oui	Oui	Non
A Self-Adaptive access control model	Oui	Oui	Oui	Non

Tableau 2.2 – Tableau comparatif des différentes extensions de ABAC

Plus récemment, des modèles de contrôle d'accès basés sur le modèle de contrôle d'accès basé sur les attributs (ABAC) qui sont spécialement conçus pour l'IoT ont été proposés. Nous allons présenter certains de ces modèles dans les sections suivantes.

2.6.8.1 The access-control oriented (ACO)

Alshehri and Sandhu [6] sont les premiers à introduire le contrôle de la communication des objets virtuels dans l'IoT. Ils ont présenté l'architecture ACO (The access-control oriented) qui comprend quatre couches: une couche d'objet, une couche d'objet virtuel, une couche de services cloud et une couche d'application afin de proposer les modèles opérationnels ACL-Cap et ABAC permettant de contrôler cette communication. Ainsi qu'un cas d'utilisation de cette architecture a été défini.

- La couche d'objet: cette couche comprend des objets physiques tels que des capteurs, des actionneurs, etc. Les utilisateurs peuvent communiquer directement avec ces objets.
- La couche d'objet virtuel: un objet virtuel représente l'état actuel d'un objet physique, ces derniers peuvent communiquer les uns avec les autres lorsque les deux sont connectés et l'utilisation d'un mécanisme de contrôle d'accès est nécessaire afin de contrôler cette communication.
- La couche de services cloud: cette couche permet de stocker et traiter les données collectées qui peuvent être utilisées et visualisées pour les utilisateurs.
- La couche d'application: cette couche offre une interface à travers laquelle les utilisateurs peuvent interagir et communiquer avec les objets.

Dans l'architecture ACO, la confidentialité des utilisateurs est préservée, elle est conçue pour permettre la collaboration dans le cloud entre différentes organisations, ainsi que

la liste des capacités de la communication des objets virtuels est facilement gérée et modifiée par les utilisateurs autorisés.

2.6.8.2 a Self-Adaptive access control model

Ouechtati et al. [52] ont proposé une approche basée sur un middleware de contrôle d'accès pour l'IoT, qui est une extension du modèle ABAC. Cette approche a pour but d'intégrer le comportement des utilisateurs et l'évaluation des risques dans le modèle ABAC pour améliorer le niveau de contrôle d'accès dans l'IoT. Ce middleware est composé de cinq principaux composants: Sujet, Adaptive Access Control Manager (AACM), Risk Assessment Manager (RAM), Recommendation Detector (RD) et Context Manager (CM).

Après l'envoi de la requête d'accès par le sujet, différentes étapes seront effectuées par les composants de ce middleware afin de fournir la décision d'accès:

- Le sujet envoie une requête d'accès à l'AACM.
- L'AACM reçoit la requête d'accès et vérifie les attributs du sujet demandeur.
- l'AACM envoie une requête à la RAM pour demander la valeur de risque.
- La RAM reçoit la demande de la valeur de risque et envoie un appel au RD afin d'envoyer les recommandations nécessaires pour pouvoir être utilisé dans le calcul de la confiance et la détection d'attaques.
- Le RD, à son tour transmet les informations demandées à la RAM.
- De plus, la RAM envoie une requête au CM pour demander les informations de contexte nécessaires et le comportement du sujet.

- Le CM reçoit la demande d'informations de contexte, puis il collecte et transmet les informations demandées à la RAM.
- La RAM utilise les informations de contexte, la valeur de confiance et le comportement du sujet pour générer la valeur de risque qui sera envoyée à l'AACM.
- L'AACM reçoit la valeur de risque.
- L'AACM utilise la valeur de risque, les informations de contexte, les règles et les ensembles de règles pour générer une adaptation adaptative.
- L'AACM envoie une réponse avec la décision d'accès au sujet.

Cependant, l'efficacité et la performance de ce middleware doivent être testée et évaluée dans un environnement réel.

2.6.8.3 Attribute-Based Access Control to Data Distribution Service (DDS)

Murugesan et al. [50] ont présenté un service de distribution de données (DDS), qui est une norme pour les communications de publish / subscribe centrées sur les données, cette norme a pour objectif de traiter l'interopérabilité dans l'IoT. DDS définit son propre modèle de contrôle d'accès adapté au modèle de contrôle d'accès basé sur les attributs (ABAC) en offrant plus de flexibilité.

- Topic Attributes: les topics sont des objets ou bien des ressources, l'accès à ces dernières doit être contrôlé. Ils sont identifiés par un nom, un type et un ensemble de stratégies de Qualité de Service (QoS) associées.
- Subscriber Attributes: ces attributs peuvent être définis en termes de: SubscriberId ,SubscriberTopic, SubscriberQoS, SubscriberType et DomainId.

- Environment Conditions: ces attributs peuvent inclure la date, l'heure actuelle et l'adresse du réseau.
- Access Rules: une fois que les attributs des sujets et des objets et les conditions environnementales sont établis, les règles d'accès peuvent être défini.
- Access Policies: les politiques d'accès sont spécifiées à l'aide d'un ensemble de règles d'accès.

D'autres modèles de contrôle d'accès ont été proposés dans la littérature afin d'assurer la sécurité des données IoT, nous allons présenter quelques approches:

2.6.8.4 A capability-based security approach (CBAC)

Une approche basée sur la capacité adaptée dans un environnement IoT distribué a été présenté par Gusmeroli et al. [27]. Dans ce modèle, le jeton représente la capacité qui octroie les droits d'accès afin d'accéder aux différents services des dispositifs IoT. Cependant, un jeton doit être obtenu et stocké dans le dispositif à chaque fois que ce dernier souhaite accéder à un service. Cette limitation constitue un goulot d'étranglement à la mise à l'échelle dans l'IoT.

2.6.8.5 Identity Authentication and Capability Based Access Control (IACAC)

Ainsi qu'un autre modèle qui introduit le concept de capacité en fonction de l'identité afin d'accorder l'accès au réseau local a été présenté par Mahalle et al. [42], ce modèle utilise un protocole d'authentification et de contrôle d'accès efficace, ainsi qu'un protocole d'authentification mutuelle garantissant la sécurité qui évite les attaques de type homme au milieu et par déni de service (Dos). Cependant, ce schéma ne convient toujours pas parfaitement aux petits dispositifs IoT.

2.6.8.6 A Secure Capability-Based Access Control model (S-CBAC)

Pour résoudre le problème du goulot d'étranglement à la mise à l'échelle présenté par Gusmeroli et al. [27], les auteurs Chen et al. [16] ont proposé une nouvelle approche basée sur le "groupe d'accès" (plusieurs dispositifs fournissant différents services placés dans un groupe) cela donne la possibilité d'atteindre plusieurs services à l'aide un seul jeton au lieu de plusieurs. Une Adresse Locale Unique (ULA) est affectée à chaque périphérique du "groupe d'accès". Donc, l'utilisateur doit obtenir un Jeton d'Accès au Groupe (GAT) et un ULA pour atteindre le "groupe d'accès" et accéder au périphérique qui fournit le service souhaité. Cependant, ce modèle doit inclure l'authentification des périphériques afin d'améliorer la sécurité des données IoT.

2.6.8.7 Attribute-Based Encryption with Attribute Hierarchies (ABE-AH)

Un nouveau type de contrôle d'accès chiffré a été suggéré par Zhu et al. [72], Bethencourt et al. [11], qui améliore la capacité d'expression des stratégies d'accès, réduit les frais généraux de calcul et la taille des textes cryptés ainsi que les clés privées. Ces systèmes sont composés de différentes techniques d'optimisations dans leur mise en oeuvre fournissant des politiques résistantes aux attaques de collusion.

Selon Zhu et al. [72] attribute-based encryption (ABE) est une approche puissante et flexible qui implémente le contrôle d'accès basé sur les attributs (ABAC) en chiffrant les données avec une politique d'accès spécifique basée sur les attributs.

Une solution de contrôle d'accès distribuée basée sur des profils de sécurité. Cependant, dans cette solution la résistance aux attaques n'a pas été explorée.

2.6.8.8 Secure Role-Based Access Control (SecRBAC)

Pour assurer la protection des données dans le cloud, les auteurs M. Marín Pérez et al. [41] ont présenté une solution de contrôle d'accès centrée sur les données et basée sur les rôles, y compris des hiérarchies de rôles et d'objets, qui permet de protéger les données de l'utilisateur contre tout comportement malveillant quel que soit le nuage ou le fournisseur de services qui le détient. Les données et les règles d'autorisations sont protégées par la cryptographie. Une grande expressivité est fournie par le modèle d'autorisation avec une prise en charge de la hiérarchie des rôles et des ressources.

2.6.8.9 k-Times Attribute-Based Anonymous Access Control

Une autre approche visant à protéger les données dans le cloud a été présentée par Yuen et al. [68], dans cette approche l'utilisateur peut s'authentifier de manière anonyme auprès du serveur du cloud. Le serveur ne connaît que certains attributs obligatoires des utilisateurs sans connaître son identité. Une limite de k-times pour le contrôle d'accès anonyme est fournie. Donc, le serveur restreint l'accès au système en utilisant un maximum de k-times au cours d'une période ou d'un événement et refuse chaque accès supplémentaire. Ce modèle est pratique et assure la sécurité des données selon une preuve de sécurité qui a été fournie.

2.6.8.10 Multitenant Access Control

Les auteurs Kappes et al. [35] ont introduit un contrôle d'accès dont chaque locataire gère les identités et les autorisations de ses propres utilisateurs de manière indépendante. Les espaces de noms d'identités des locataires peuvent être isolés de manière sécurisée par le système de fichiers. Ce dernier permet le partage de fichiers configurable entre différents locataires et hôtes en maintenant séparément les autorisations d'accès de

chaque locataire. Ce contrôle d'accès peut fournir une interface efficace pour le stockage parmi les domaines administratifs co-localisés, permettant un accès partagé aux données entre plusieurs utilisateurs.

2.6.8.11 Attribute-based Access Control for ICN Naming Scheme

Afin de garantir la sécurité dans un réseau centré, une solution de contrôle d'accès a été présentée par Li et al. [39]. Cette solution repose principalement sur deux schémas, le premier est une ontologie basée sur la gestion des attributs qui gèrent les attributs distribués dans le réseau ICN, la seconde est un système de dénomination basé sur ABE (Attribute-Based Encryption) préservant la confidentialité et atteignant un niveau de sécurité très élevé.

2.6.8.12 Synthèse/Analyse des approches précédentes

Les différents modèles de contrôle d'accès que nous avons présentés précédemment partagent tous un objectif similaire qui consiste à étendre le modèle ABAC afin d'améliorer la flexibilité du contrôle d'accès. Cependant, chaque modèle suit une approche différente et définit une politique d'accès spécifique. Il est à noter que les politiques d'accès qui peuvent être générées des approches proposées dans les modèles WS-ABAC, $ABAC_{\alpha}$ et HGABAC limitent la flexibilité, cela représente une limitation majeure en particulier dans le contexte IoT.

Plus récemment, des modèles de contrôle d'accès basés sur ABAC qui sont spécifiquement conçus pour l'IoT ont été proposés. Les approches proposées par Alshehri and Sandhu [6], Ouechtati et al. [52] et Murugesan et al. [50] sont dédiées, adaptées aux systèmes IoT et basées sur différents aspects de l'IoT, dont le modèle Access-Control Oriened (ACO) introduit la communication des objets virtuels dans l'IoT. Self-adaptive

access control est basé sur un middleware qui a pour objectif d'améliorer le niveau de contrôle d'accès dans l'IoT. Data Distribution Service (DDS) se base notamment sur ABAC pour définir un modèle de contrôle d'accès qui traite l'interopérabilité dans les systèmes IoT.

Des modèles de contrôle d'accès sont conçus pour être adaptés dans des environnements IoT distribués ont été introduits par Mahalle et al. [42], Gusmeroli et al. [27] et Chen et al. [16], ces modèles se basent sur la capacité afin d'accorder ou de refuser l'accès sur un réseau local. Cependant, ces modèles présentent des limitations qui constituent un goulot d'étranglement à la mise à l'échelle dans l'IoT. De plus, ils ne sont pas convenables pour tous les types de dispositifs IoT. Nous pouvons donc constater qu'il n'existe pas de modèles globales qui conviennent à la fois aux systèmes IoT et non IoT qui permettent de faciliter la migration d'une politique d'accès d'un modèle de contrôle d'accès à un autre.

2.7 CONCLUSION

Différentes solutions traitent le défi de sécurité dans l'Internet des Objets. Bien que plusieurs approches ont été proposées pour assurer et garantir la sécurité des données telle que les modèles de contrôle d'accès traditionnels (MAC, DAC et RBAC) qui sont des modèles rigides et l'utilisation de ces derniers dans des environnements dynamiques peut provoquer des risques et produire des situations imprévues en raison de leurs politiques de contrôle d'accès inappropriés qui n'offrent pas la flexibilité. Une solution idéale qui réside dans la mise en place d'un mécanisme de contrôle d'accès qui répond aux exigences de sécurité est incontournable.

Tout au long de ce chapitre, nous avons présenté l'état de l'art des approches actuelles dans le domaine de l'Internet des Objets, nous avons introduit le défi de sécurité auquel

l'IoT est confrontée, ensuite, nous avons mis en évidence des axes de recherche importants qui portent sur l'identification, l'authentification et le contrôle d'accès pour la résolution de ce défi. En outre, nous avons présenté une analyse détaillée sur le mécanisme de contrôle d'accès, dont nous avons introduit les différents modèles de contrôle d'accès existants dans la littérature. Nous avons présenté en premier lieu les modèles de contrôle d'accès traditionnels (MAC, DAC et RBAC) en introduisant leurs principes et leurs limitations majeures lorsqu'ils n'arrivent pas à satisfaire les exigences en termes de sécurité des environnements informatiques. Nous avons ensuite présenté le modèle ABAC et ses différentes variations qui ont été proposées dans la littérature.

Dans ce travail de recherche et afin de combler les lacunes et les limitations existantes dans les systèmes d'accès pour l'IoT en général et ABAC en particulier, nous pensons qu'il est nécessaire de mettre en place un modèle de contrôle d'accès générique et plus flexible qui est adapté à la fois aux systèmes IoT et non-IoT. Par conséquent, dans le chapitre suivant nous proposons un nouveau modèle de contrôle d'accès flexible et générique en introduisant ses concepts de base et leurs relations.

CHAPITRE 3

NOUVEAU MODÈLE DE CONTRÔLE D'ACCÈS BASÉ SUR ABAC ET LES FONCTIONS D'ORDRE SUPÉRIEURES

3.1 INTRODUCTION

Après avoir présenté notre recherche bibliographique dans le chapitre précédent en introduisant les différents modèles de contrôle d'accès existants dans la littérature, nous nous intéresserons dans ce chapitre à présenter la contribution de ce travail de recherche qui consiste à proposer un nouveau modèle de contrôle d'accès afin d'assurer et de garantir la sécurité des données. Ce nouveau modèle étend les concepts de base du modèle de contrôle d'accès basé sur les attributs ou ABAC (Attribute Based Access Control) pour mettre en place un modèle ABAC générique afin d'offrir la possibilité d'implémenter des politiques de contrôle d'accès générales et adaptées aux systèmes IoT et non-IoT. Nous allons introduire les fondements théoriques de notre nouveau modèle de contrôle d'accès, nommé Higher-order Attribute-Based Access Control (HoBAC), ainsi qu'une vue générale des principaux concepts de ce nouveau modèle et de leurs relations.

Nous allons présenter un exemple illustratif afin de mieux comprendre les concepts de base du modèle HoBAC qui sont présentés tout au long de ce chapitre. Ainsi, nous présentons un cas d'utilisation lié à l'IoT comme illustré dans la Figure 3.1. La vision d'une maison intelligente est d'ajouter des fonctionnalités aux objets domestiques afin d'offrir de nombreux avantages dans la vie quotidienne, y compris la commodité et une utilisation efficace des ressources. Dans notre cas d'utilisation, la maison intelligente comprend des personnes et des entités intelligentes qui correspondent aux dispositifs intelligents tels que le capteur de température, le capteur d'humidité, le détecteur de mouvements, le thermostat intelligent, l'éclairage intelligent, etc.

Il est à noter que ce cas d'utilisation serait utilisé tout au long de ce chapitre comme référence pour illustrer les concepts de base du modèle HoBAC.

Figure 3.1 – Cas d'utilisation lié à l'IoT (maison intelligente)

3.2 FONDEMENTS THÉORIQUES DE HOBAC

HoBAC est une généralisation du modèle de contrôle d'accès basé sur les attributs ou ABAC (Attribute Based Access Control). HoBAC s'appuie sur les mêmes concepts

de base du modèle ABAC : sujets, objets et contextes. Ces concepts sont des entités sémantiquement différentes, mais ils partagent la même structure : ce sont tous des entités composées d'un ensemble d'attributs (*c.f.* Figure 3.2).

Pour cela, nous factorisons ces concepts (sujets, objets et contextes) dans un nouveau concept nommé *Entité*.

Les univers de tous les objets, sujets, contextes et entités sont respectivement représentés par, U_O, U_S, U_C et U_E. Nous avons $U_E = U_O \cup U_S \cup U_C$.

Les attributs étant les éléments constitutifs fondamentaux du modèle HoBAC, nous commençons par les définir avant de définir les autres concepts.

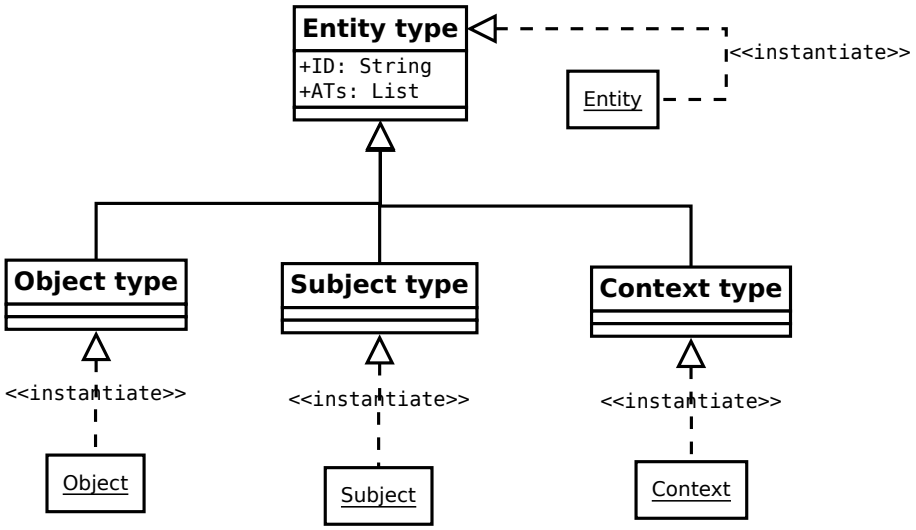

Figure 3.2 – Une vue générale sur les principaux concepts du modèle HoBAC et leurs relations

3.2.1 L'ATTRIBUT

Un attribut est défini par un identifiant unique et une valeur.

Definition 3.2.1. ***Attribut***

Un attribut est un tuple $A =< ID, v >$ *où:*

- *ID, noté A.ID, est un identifiant unique universel (UUID) appartenant à l'univers* U_{UUID}*;*

- *v, noté A.v, est une valeur d'un certain type. Le type d'une valeur v est noté v.type.*

L'univers de tous les attributs est noté U_A.

Nous avons cité précédemment que les attributs sont les éléments constitutifs fondamentaux du modèle HoBAC. Puisque dans ce modèle un objet (resp. sujet) peut contribuer avec un ou plusieurs attributs afin de créer un autre objet (resp. sujet) avec un haut niveau d'abstraction alors la création des objets et des sujets avec un haut niveau d'abstraction se fait initialement par la fédération des attributs provenant principalement des objets, des sujets et des contextes existants.

3.2.2 TYPES D'ATTRIBUTS

Dans HoBAC, les attributs de U_A sont regroupés sémantiquement dans des *types d'attributs*. Dans ce modèle, un type attribué est pour un attribut ce qu'est une classe pour un objet en programmation orientée objet. Ainsi, qu'un attribut est une instance d'un type d'attribut. Le type d'attribut est composé d'un nom et d'une fonction d'ordre supérieur représentée par $\mathcal{S}$-Structure (*c.f.* Définition 3.2.2).

Definition 3.2.2. ***$\mathcal{S}$-Structure pour une fonction d'ordre supérieur***

La $\mathcal{S}$-Structure de la fonction d'ordre supérieur $\mathcal{F}$ est un tuple $\mathcal{S} =< \mathcal{F}, pm >$ où $pm = (p_1, p_2, ..., p_N)$ ($N \in \mathcal{N}^+$) est une liste finie ordonnée de N paramètres de la fonction $\mathcal{F}$ tel que: $\forall p \in pm$ nous avons:

$$p = \begin{cases} littéral & ou \\ \mathcal{S} - Structure \end{cases}$$

L'univers de toutes les $\mathcal{S}$-structures est noté $U_{\mathcal{S}}$.

Étant donné une $\mathcal{S}$-structure de $U_{\mathcal{S}}$ ($\mathcal{S} =< \mathcal{F}, pm >$), nous avons ce qui suit:

1. $\mathcal{F}$ est désigné par $\mathcal{S}.\mathcal{F}$;
2. pm est désigné par $\mathcal{S}.pm$;
3. Le nombre de paramètres de $\mathcal{S}$ est égal à la cardinalité de pm, il est désigné par $|\mathcal{S}.pm|$;
4. Le i^{me} paramètre de $\mathcal{S}.pm$ est désigné par $\mathcal{S}.pm[i]$ où $i \in [1..|\mathcal{S}.pm|]$;

Il est à noter qu'une $\mathcal{S}$-Structure est défini récursivement. En fait, dans la Définition 3.2.2, on peut remarquer que chaque paramètre peut être un littéral ou une $\mathcal{S}$-Structure.

Ainsi, une $\mathcal{S}$-Structure peut récursivement dépendre de nombreuses autres $\mathcal{S}$-Structures.

Cette dépendance est formalisée à l'aide de deux relations: 1) une relation de dépendance immédiate de $\mathcal{S}$-Structure, et 2) une relation de dépendance générale de $\mathcal{S}$-Structure.

Une $\mathcal{S}$-Structure a une dépendance immédiate sur une autre $\mathcal{S}$-Structure si la première est l'un des paramètres de la derniére (*c.f.* Définition 3.2.3).

Definition 3.2.3. ***Relation de dépendance immédiate de $\mathcal{S}$-Structure (1-dependency)***

Considérons $\mathcal{S}_1$ et $\mathcal{S}_2$ deux $\mathcal{S}$-Structure de $U_{\mathcal{S}}$, on dit que $\mathcal{S}_1$ dépend immediatement de $\mathcal{S}_2$, noté $\mathcal{S}_1 \preccurlyeq_s^1 \mathcal{S}_2$ ou $\mathcal{S}_2 \succcurlyeq_s^1 \mathcal{S}_1$, si et seulement si: $\exists i \in [1..|\mathcal{S}_2.pm|]$: $\mathcal{S}_2.pm[i] = \mathcal{S}_1$.

La relation de dépendance générale est définie de manière récursive à l'aide de *1-dependency* (*c.f.* Définition 3.2.4).

Definition 3.2.4. ***Relation de N-dependence de $\mathcal{S}$-Structure (N-dependency)***

Considérons $\mathcal{S}_1$ et $\mathcal{S}_2$ deux $\mathcal{S}$-Structure de $U_{\mathcal{S}}$, on dit que $\mathcal{S}_1$ dépend de $\mathcal{S}_2$, noté $\mathcal{S}_1 \preccurlyeq_s^N \mathcal{S}_2$ ou $\mathcal{S}_2 \succcurlyeq_s^N \mathcal{S}_1$ ($N \in \mathcal{N}^+$), si et seulement si l'une des conditions ci-dessous est vérifiée:

- $\mathcal{S}_1 \preccurlyeq_s^1 \mathcal{S}_2$;
- $\exists \mathcal{S}_3 \in U_{\mathcal{S}}$ *tel que:* $\mathcal{S}_1 \preccurlyeq_s^1 \mathcal{S}_3$ *et* $\mathcal{S}_3 \preccurlyeq_s^{N-1} \mathcal{S}_2$.

Maintenant que nous avons introduit la $\mathcal{S}$-Structure et les relations de dépendance, on peut définir le *Type d'attribut* (*c.f.* Définition 3.2.5).

La relation d'instanciation entre les attributs et les types d'attributs est définie formellement dans la Définition 3.2.6.

Definition 3.2.5. ***Type d'attribut***

Le type d'attribut est un tuple $AT =< N, \mathcal{S} >$ où:

- *N, noté $AT.N$, est un string qui représente le nom de AT;*
- *S, noté $AT.\mathcal{S}$, est une $\mathcal{S}$-Structure.*

L'univers de tout les types d'attributs est noté U_{AT}.

3.2.3 INSTANCIATION DE TYPE D'ATTRIBUT

Les attributs qui sont des instances d'un type d'attribut AT sont donnés par la relation $[AT]_{U_A}$ (*c.f.* Définition 3.2.6).

Definition 3.2.6. ***La relation d'instantiation d'un type d'attribut***

La fonction $[]_{U_A}$*, qui représente la relation d'instantiation d'un type d'attribut est définie comme suit:* $[\,]_{U_A} : U_{AT} \rightarrow \mathcal{P}(U_A)$ [1] *tel que:* $AT \in U_{AT}$ *et* $a \in U_A$*, nous avons* $a \in [AT]_{U_A}$ *si et seulement si a est une instance de* AT*.*

La Figure 3.3 illustre graphiquement la relation d'instanciation existante entre les attributs et les types d'attributs. Cet exemple est lié à notre cas d'utilisation Figure 3.1 dont la maison intelligente dispose de deux capteurs, un capteur de température et un capteur d'humidité qui détectent la température et l'humidité à l'intérieur de cette maison.

Considérons la température et l'humidité deux types d'attributs appartenant à l'univers des types d'attributs U_{AT} et H1, H2, T1, T2 des attributs appartenant à l'univers des attributs U_A.

- T1 et T2 sont deux instances du type d'attribut Température, nous avons $\{T1, T2\} \subseteq [Température]_{U_A}$ dont T1=<85J21, 22> et T2 =<4Y782, 30>.

- H1 et H2 sont deux instances du type d'attribut Humidité, nous avons $\{H1, H2\} \subseteq [Humidité]_{U_A}$ dont H1=<412R3, 65> et H2 =<2B596, 74>.

Il est à noter que les ID de ces attributs sont générés automatiquement.

[1] $\mathcal{P}(X)$ représente un surper ensemble de l'ensemble X

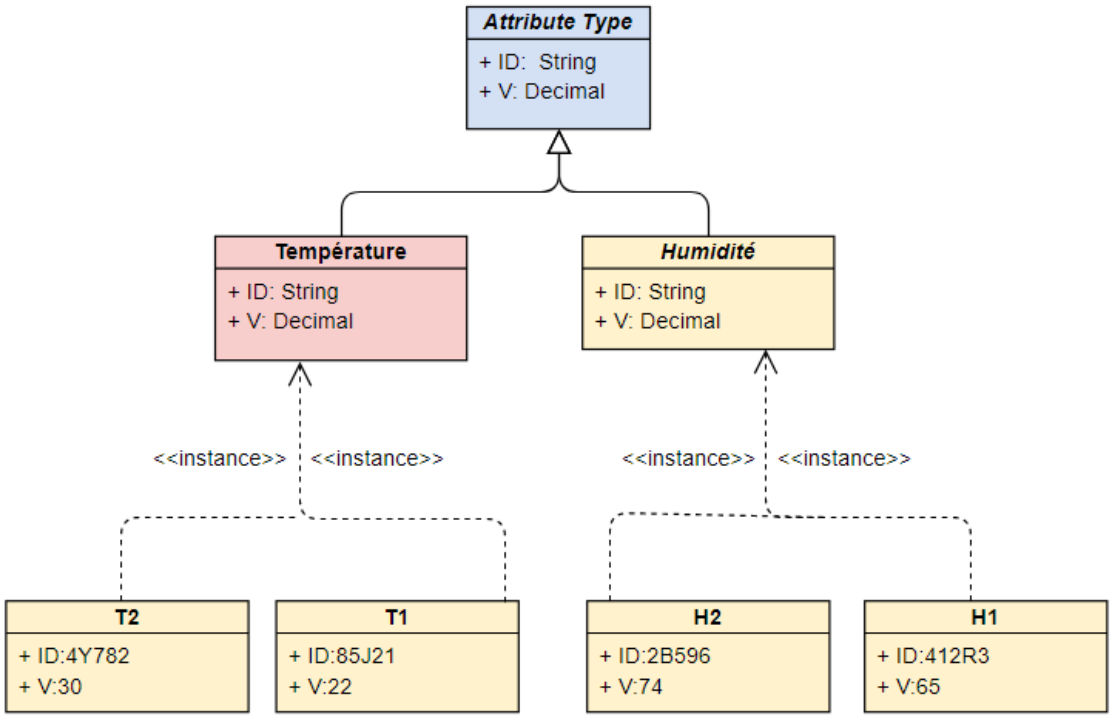

Figure 3.3 – Exemple d'instanciation de type d'attribut

3.2.4 ESPACE DE TYPE D'ATTRIBUT

L'univers de tout les types d'attributs U_{AT} est doté d'une relation de généralité permettant de l'organiser dans une structure d'ordre partiel. Cette relation facilite l'exploration des sous-ensembles de cet espace (*c.f.* Définition 3.2.7).

Definition 3.2.7. ***Relation de généralité de type d'attribut***

Considérons A_1 et A_2 deux types d'attributs de U_{AT} ($(A_1,A_2) \in (U_{AT})^2$), on dit que A_1 est plus général que A_2, noté $A_2 \trianglelefteq A_1$ ou $A_1 \trianglerighteq A_2$, si et seulement si $[A_2]_{U_A} \subseteq [A_1]_{U_A}$.

L'intuition de cette relation de généralité est qu'un type d'attribut est plus général qu'un autre, si les instances de ce dernier sont aussi des instances de l'ancien.

Mais, cette relation n'est pas pratique car il n'est pas possible d'avoir toutes les instances

d'un type d'attribut. Ainsi, nous introduisons une relation plus pratique basée sur la relation de N-dependance de $\mathcal{S}$-Structure présentée dans la définition 3.2.4 (*c.f.* Définition 3.2.8).

Definition 3.2.8. ***Relation de dépendance de type d'attribut***

Considérons A_1 et A_2 deux types d'attributs de U_{AT} ($(A_1,A_2) \in (U_{AT})^2$), on dit que A_2 dépend de A_1, noté $A_2 \preccurlyeq_a A_1$ ou $A_1 \succcurlyeq_a A_2$, si et seulement si: $A_2.\mathcal{S} \succcurlyeq_s^N A_1.\mathcal{S}$ ($A_1.\mathcal{S} \succcurlyeq_s^N A_2.\mathcal{S}$).

La relation de dépendance de type d'attribut organise l'espace de type d'attribut en niveaux où chaque niveau est composé de types d'attributs ayant la même hauteur de dépendance. La hauteur de dépendance d'un attribut est définie par la formule récursive de (*c.f.* Définition 3.2.9).

Definition 3.2.9. ***Hauteur de dépendance du type d'attribut***

Considérons A_1 un type d'attribut de U_{AT} ($A_1 \in U_{AT}$), la hauteur de dépendance de A_1 est donnée par la relation $\mathcal{H}_a : U_{AT} \rightarrow \mathcal{N}^+$ tel que: $(A \in \mathcal{H}_a(A_1)) \Rightarrow (A_1 \preccurlyeq_a A)$.

$$\mathcal{H}_a(A_1) = \begin{cases} 0 & si \nexists A_2 \in U_{AT} : A_2.\mathcal{S} \preccurlyeq_s^N A_1.\mathcal{S}, \\ & (o\ A_1.\mathcal{S} = littérale), \\ \mathcal{H}_a(A_2) + 1 & o\ A_2 \in U_{AT} : A_1.\mathcal{S} \preccurlyeq_s^1 A_2.\mathcal{S}. \end{cases}$$

La Figure 3.4 illustre un exemple de la relation de dépendance existante entre les types d'attributs. Dans cet exemple, les types d'attributs Température-maison et Humidité-maison dépendent respectivement du type d'attribut Température-chambre et Humidité-chambre. Les valeurs des instances des types d'attributs précédents sont calculées à partir des valeurs des instances de ces dernières (la température et l'humidité de la maison sont

calculées à partir de la température et l'humidité des différentes chambres de la maison), cela se fait à l'aide d'une fonction donnée, tel que la fonction qui calcule la moyenne. En outre, le type d'attribut HumidX-maison dépend de Température-maison et Humidité-maison car l'humidX d'une maison peut être calculé en utilisant la température moyenne et l'humidité relative de la maison, idem pour le type d'attribut HumidX-chambre qui peut être calculé en utilisant la température moyenne et l'humidité relative de la chambre. Cependant, il n'est pas nécessaire d'énumérer toutes les dépendances, il suffit de sélectionner uniquement celles qui conviennent le mieux à l'application et à sa politique de contrôle d'accès.

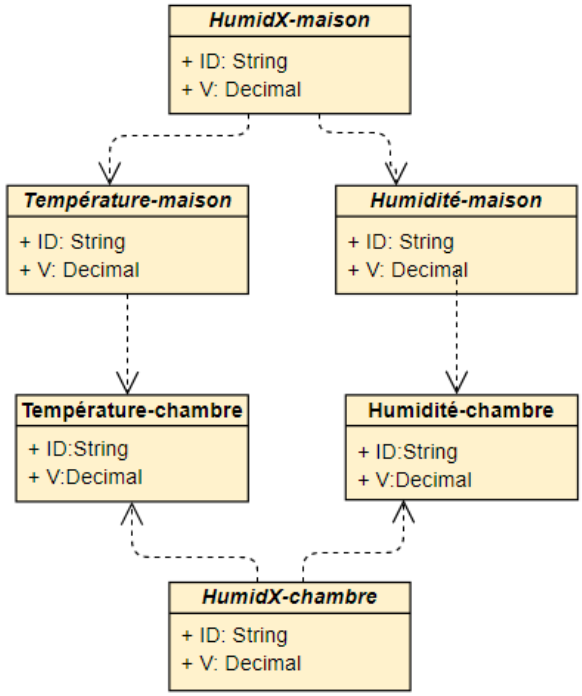

Figure 3.4 – Exemple de la relation de dépendance entre les types d'attributs

3.2.5 ENTITÉ

Dans HoBAC, une entité est définie par un identifiant unique ID et une liste d'attribut (*c.f.* Définition 3.2.10).

Definition 3.2.10. ***Entité***

Une entité est un tuple $E =< ID, As >$ *où:*

- *ID, noté* $E.ID$*, est un identifiant unique universel (UUID) appartenant à l'espace UUID* U_{UUID}*;*
- *As, noté* $E.As$*, est une liste d'attribut tel que* $\forall i \in [1..|E.As|] : E.As[i] \in U_A$ [2] [3]

Une ***Entité*** dans le modèle HoBAC, elle peut être un sujet, un objet ou un contexte, composée d'un ensemble d'attributs.

3.2.5.1 Types d'entité

Dans HoBAC, les entités de U_E sont sémantiquement regroupés en *entity types* ou en classes de la même manière que nous avons regroupé les attributs abstraits en types d'attributs.

La définition formelle d'un type d'entité est donnée ci-dessous par la Définition 3.2.11. La relation d'instanciation entre les objets et les types d'objets est introduite dans la Définition 3.2.12.

Definition 3.2.11. ***Types d'entité***

Un type d'entité est un tuple $ET =< N, ATs >$ *où:*

- *N, noté* $ET.N$*, est un string qui représente le même type d'entité OT;*
- *ATs, noté* $ET.ATs$*, est une liste de types d'attributs tel que* $\forall i \in [1..|ET.ATs|] : ET.ATs[i] \in U_{AT}$

[2] $x[i]$: l'élément à la position i dans la liste x
[3] $|x|$: la cardinalité de la liste x

L'univers de tous les types d'entités est noté U_{UT}. Également, l'univers de tous les types d'objets, types de sujets et types de contextes sont respectivement notés U_{OT}, U_{ST} et U_{CT}.

3.2.6 INSTANTIATION DE TYPE D'ENTITÉ

Les entités qui sont des instances d'un type d'entité ET sont données par la relation $[ET]_{U_E}$ (*c.f.* Définition 3.2.12).

Definition 3.2.12. ***Relation d'instantiation de type d'entité***

La fonction $[]_{U_E}$, qui représente la relation d'instantiation d'un type d'entité est définie comme suit: $[\,]_{U_E} : U_{ET} \rightarrow \mathcal{P}(U_E)$ tel que: $ET \in U_{ET}$ et $o \in U_E$ nous avons $E \in [ET]_{U_E}$ si et seulement si:

- $|E.As| = |ET.ATs|$;
- $\forall i \in [1..|E.As|] : E.As[i] \in [ET.ATs[i]]_{U_A}$.

L'intuition derrière la relation d'instanciation de type d'entité est qu'une entité est une instance d'un type d'entité si chaque attribut de la première est une instance de type d'attribut correspondant à la dernière. [4]

De plus, les relations d'instanciations de type d'objet, de type de sujet et de type de contexte sont définies de la même façon et sont notées $[]_{U_O}$ ($[\,]_{U_O} : U_{OT} \rightarrow \mathcal{P}(U_O)$), $[]_{U_S}$ ($[\,]_{U_S} : U_{ST} \rightarrow \mathcal{P}(U_S)$) et $[]_{U_C}$ ($[\,]_{U_C} : U_{CT} \rightarrow \mathcal{P}(U_C)$).

[4] ayant la même position dans les listes (la liste des attributs des entités et la liste des types d'attributs des types d'entité))

3.2.7 ESPACE DE TYPE D'ENTITÉ

L'univers de tous les type d'entité U_{ET} est partiellement ordonné par une relation de généralité (*c.f.* Définition 3.2.13).

Definition 3.2.13. ***Relation de généralité de type d'entité*** *Considérons E_1 et E_2 deux types d'entités de U_{ET} ($(E_1,E_2) \in (U_{ET})^2$), on dit que E_1 est plus général que E_2, noté $E_2 \trianglelefteq_e E_1$ ou $E_1 \trianglerighteq_e E_2$, si et seulement si $[E_2]_{U_E} \subseteq [E_1]_{U_E}$. La relation de généralité a comme domaine et plage U_{ET} ($\trianglerighteq_e: U_{ET} \to U_{ET}$, $\trianglerighteq_e: U_{ET} \to U_{ET}$)*

Les relations de généralités de type d'objet, type de sujet et type de contexte sont définies de la même manière et sont notées respectivement $\trianglelefteq_o, \trianglerighteq_o$ ($\trianglelefteq_o, \trianglerighteq_o: U_{OT} \to U_{OT}$), $\trianglelefteq_s, \trianglerighteq_s$ ($\trianglelefteq_s, \trianglerighteq_s: U_{ST} \to U_{ST}$) et $\trianglelefteq_c, \trianglerighteq_c$ ($\trianglelefteq_c, \trianglerighteq_c: U_{CT} \to U_{CT}$). Comme c'est le cas avec la relation de généralité de type d'attribut, la relation de généralité de type d'attribut présentée dans la Définition 3.2.13 n'est pas pratique car il n'est pas possible d'avoir toutes les instances d'un type d'entité.

Ainsi, nous introduisons une relation plus pratique basée sur les fonctions d'ordre supérieur des types d'attribut (c.f. Définition 3.2.14).

La Figure 3.5 illustre un exemple de l'hiérarchie d'entités. Cet exemple est lié à notre cas d'utilisation présenté dans la Figure 3.1. Considérons la maison-intelligente un type d'entité appartenant à l'univers des types d'entités U_{ET} et l'entité *SH* représente une instance du type d'entité maison-intelligente appartenant à l'univers des entités U_E. Nous présentons les personnes et les capteurs à l'intérieur de cette maison. Ainsi, les entités $P1$,$P2$ représentent les personnes et $C1$,$C2$ représentent les capteurs, ces entités sont associées à l'entité SH.

Comme c'est le cas avec la relation de généralité de type d'attribut, la relation de généralité de type d'entité présentée dans la Définition 3.2.7 n'est pas pratique car il n'est pas possible d'avoir toutes les instances d'un type d'entité.

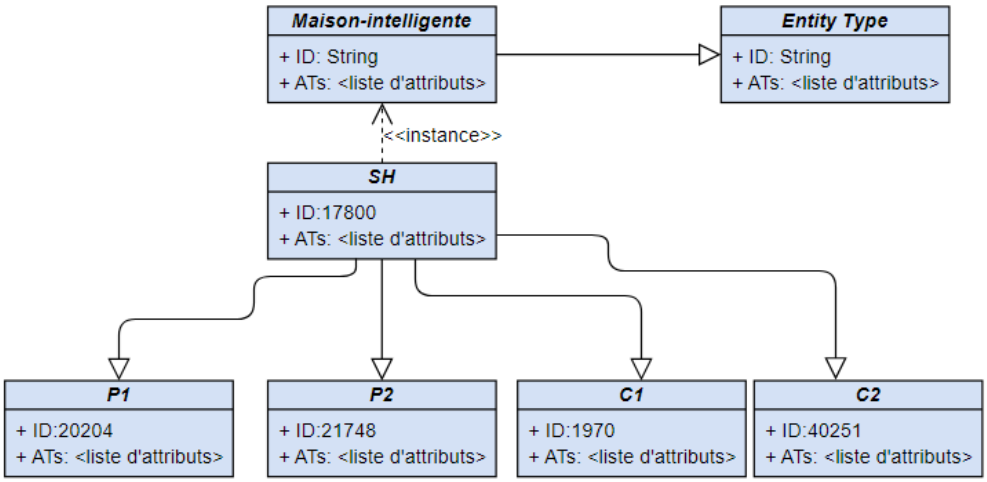

Figure 3.5 – Exemple d'instanciation de type d'entité

Definition 3.2.14. ***Relation de dépendance de type d'entité***

Considérons E_1 et E_2 deux types d'entités de U_{ET} ($(E_1,E_2) \in (U_{ET})^2$), on dit que E_2 dépend de E_1, noté $E_2 \preccurlyeq_e E_1$ ou $E_1 \succcurlyeq_e E_2$, si et seulement si:

$$\exists i \in [1..|E_2.ATs|], \exists j \in [1..|E_1.ATs|] : E_2.ATs[i].\mathcal{S} \preccurlyeq_s^N E_1.ATs[j].\mathcal{S}.$$

La relation de dépendence a comme domaine et gamme U_{ET} ($\preccurlyeq_e$: $U_{ET} \rightarrow U_{ET}$, $\succcurlyeq_e$: $U_{ET} \rightarrow U_{ET}$)

Les relations de dépendances de type d'objet, type de sujet et type de contexte sont définies de la même manière et sont respectivement notées $\preccurlyeq_o, \succcurlyeq_o$ ($\preccurlyeq_o, \succcurlyeq_o$: $U_{OT} \rightarrow U_{OT}$), $\preccurlyeq_s, \succcurlyeq_s$ ($\preccurlyeq_s, \succcurlyeq_s$: $U_{ST} \rightarrow U_{ST}$) et $\preccurlyeq_c, \succcurlyeq_c$ ($\preccurlyeq_c, \succcurlyeq_c$: $U_{CT} \rightarrow U_{CT}$).

3.2.8 RELATIONS ENTRE LES ANCÊTRES ET LES SUCCESSEURS DE TYPE D'ENTITÉ

Une entité peut avoir un ancêtre ou plus (*c.f.* Définition 3.2.15) et successeur (*c.f.* Définition 3.2.16). Ces relations sont utiles pour explorer l'espace de type entité.

Definition 3.2.15. ***Relation d'ascendance de type d'entité***

Considérons E_1 un type d'entité de U_{ET} ($E_1 \in U_{ET}$), les ancêtres de E_1 sont donnés par la relation $\mathcal{A}_e : U_{ET} \rightarrow P(U_{ET})$ tel que: $(E \in \mathcal{A}_e(E_1)) \Rightarrow (E \preccurlyeq_e E_1)$.

Definition 3.2.16. ***Relation successorale de type entité***

Considérons E_1 un type d'entité de U_{ET} ($E_1 \in U_{ET}$), les successeurs de E_1 sont donnés par la relation $\mathcal{S}_e : U_{ET} \rightarrow P(U_{ET})$ tel que: $(E \in \mathcal{S}_e(E_1)) \Rightarrow (E_1 \preccurlyeq_e E)$.

Les relations d'ascendances et de successions de type d'objet, de type de sujet et de type de contexte sont définies de la même façon pour les relations d'ascendances et successorales de type d'entité et sont notées respectivement: $\mathcal{A}_o, \mathcal{S}_o$ ($\mathcal{A}_o, \mathcal{S}_o : U_{OT} \rightarrow P(U_{OT})$), $\mathcal{A}_s, \mathcal{S}_s$ ($\mathcal{A}_s, \mathcal{S}_s : U_{ST} \rightarrow P(U_{ST})$) et $\mathcal{A}_c, \mathcal{S}_c$ ($\mathcal{A}_c, \mathcal{S}_c : U_{CT} \rightarrow P(U_{CT})$).

3.2.9 TYPES D'ENTITÉS HOMOGÈNES

Un type d'entité est composé de types d'attribut pouvant avoir différentes hauteurs de dépendance.

Un type d'entités homogènes est un type d'entité où tous ses attributs ont la même hauteur de dépendance (*c.f.* Définition 3.2.17).

Definition 3.2.17. ***Type d'entité homogène***

Considérons E un type d'entité de U_{ET} ($E \in U_{ET}$), E est dit homogène si et seulement si: $\forall A_1, A_2 \in (E.\mathcal{S})^2 : \mathcal{H}_a(A_1) = \mathcal{H}_a(A_2)$.

L'univers des types d'entités homogènes est noté U_E^h. Dans la suite de ce travail et par souci de brièveté, les termes types d'entité seront utilisés pour désigner des types d'entités homogènes tout en indiquant l'univers ciblé lorsque cela est nécessaire.

3.2.10 NIVEAUX DE DÉPENDANCE D'ESPACE DE TYPE D'ENTITÉ

La relation de dépendance de type d'entité organise l'espace de type d'entité homogène en niveaux, chaque niveau étant composé de types d'entité ayant la même hauteur de dépendance (*c.f.* Définition 3.2.18).

La hauteur de dépendance d'un type d'entité représente le nombre d'ancêtres composant une chaîne d'ascendance à partir d'un type d'entité ayant des types d'attribut avec une hauteur égale à zéro.(*c.f.* Définition 3.2.9).

Definition 3.2.18. ***Hauteur de dépendance d'un type d'entité***

Considérons E un type d'entité de U_E^h *(*$E \in U_{ET}^h$*), la hauteur de dépendance de E est donné par la relation* $\mathcal{H}_e : U_{ET}^h \rightarrow \mathcal{N}^+$ *tel que:* $\forall A \in E.\mathcal{S} : \mathcal{H}_e(E) = \mathcal{H}_a(A)$.

Les relations de hauteur de dépendance de type d'objet, de type de sujet et de contexte sont définies de manière similaire pour les relations de hauteur de dépendance de type d'entité et sont respectivement notées: $\mathcal{H}_o$ ($U_{OT}^o \rightarrow \mathcal{N}^+$), $\mathcal{H}_s$ ($U_{ST}^s \rightarrow \mathcal{N}^+$) et $\mathcal{H}_c$ ($U_{CT}^c \rightarrow \mathcal{N}^+$).

3.2.11 TYPE DE RÈGLES D'ACCÈS

Definition 3.2.19. ***Fonction d'unification (F_m)***

La fonction d'unification (F_m) est une fonction qui représente un cas particulier (spécial) de type de règle d'accès.

F_m vérifie la mise en correspondance des attributs provenant principalement des objets, des sujets et des contextes. La condition qui doit être évaluée pour prendre la décision d'accès se base, dans sa forme la plus simple, sur l'égalité des attributs. Donc, s'il y a une égalité entre les attributs du sujet et les attributs de l'objet dans un contexte particulier l'accès doit être autorisé (Allow), sinon l'accès doit être refusé (Deny).

La fonction d'unification (F_m) est présentée ci-dessous.

Algorithm 1 La fonction d'unification F_m

```
1: if F_m( S.A_s, O.A_s, C.A_s ) = true then
       Allow(S , O)
2: else
       Deny(S , O)
3: end if
```

Nous avons:

- *A_s, qui représente une liste d'attributs des différentes entités (sujets (S), objets (O) et contextes (C)), ces attributs appartiennent à l'univers des attributs noté U_A.*
- *O, qui représente un objet appartenant à l'univers des objets noté U_O.*
- *S, qui représente un sujet appartenant à l'univers des sujets noté U_S.*
- *C, qui représente un contexte appartenant à l'univers des contextes noté U_C.*

Dans l'exemple suivant qui est lié à notre cas d'utilisation Figure 3.1, nous allons montrer comment le modèle HoBAC peut être utilisé afin de fournir un contrôle d'accès

aux ressources de la maison intelligente. Nous montrerons également le rôle important que jouent les attributs dans notre modèle de contrôle d'accès.

Considérons trois types de capteurs (capteur de température, d'humidité et de mouvement) et les personnes (Bob, Alice et Lucy) dont ces capteurs et personnes sont des entités composées d'un ensemble d'attributs. Les capteurs représentent des *Objets* et les personnes représentent des *Sujets*.

Politique de contrôle d'accès:

- **cas 1:** Bob peut accéder à la température et vérifier le mouvement à l'intérieur de la maison entre les heures (8h00 et 17h00) pendant tous les jours de la semaine.
- **cas 2:** Alice peut vérifier l'humidité et la température à l'intérieur de la maison pendant toute la journée.
- **cas 3:** Lucy peut vérifier l'humidité et le mouvement à l'intérieur de la maison avant 21h00.

Les règles d'accès et selon la fonction F_m (*c.f.* Définition 3.2.19) peuvent être implémentées comme illustré ci-dessous:

- un sujet (S) avec le nom == "Bob" peut accéder à la température et au mouvement (O) (Action == "allow"), si Bob.id == capteur-température.id, Bob.id == capteur-mouvement.id et "heure >=8:00, <=17" (C);
- un sujet (S) avec le nom == "Alice" peut accéder à l'humidité et la température (O) (Action == "allow"), si Alice.id == capteur-humidité.id et Alice.id == capteur-température.id;
- un sujet (S) avec le nom == "Lucy" peut accéder à l'humidité et au mouvement

(O) (Action == "allow"), si Lucy.id == capteur-humidité.id, Lucy.id == capteur-mouvement.id et "heure < 21" (C);

- **cas 1 et 2:**

  Si Alice et Bob souhaitent accéder à la température à l'intérieur de la maison à 10h00. Au lieu d'envoyer une requête d'accès par chacun de ces sujets au capteur de température, il suffit de créer un seul sujet qui représente ces deux derniers par la fédération de leurs attributs. À l'aide des opérations d'agrégations, l'attribut *"Nom"* de ces deux sujets peut être impliqué dans la création d'un nouveau sujet avec un haut niveau d'abstraction. Ensuite, une seule requête d'accès serait envoyée par ce nouveau sujet au capteur de température comme illustré dans la Figure 3.6.

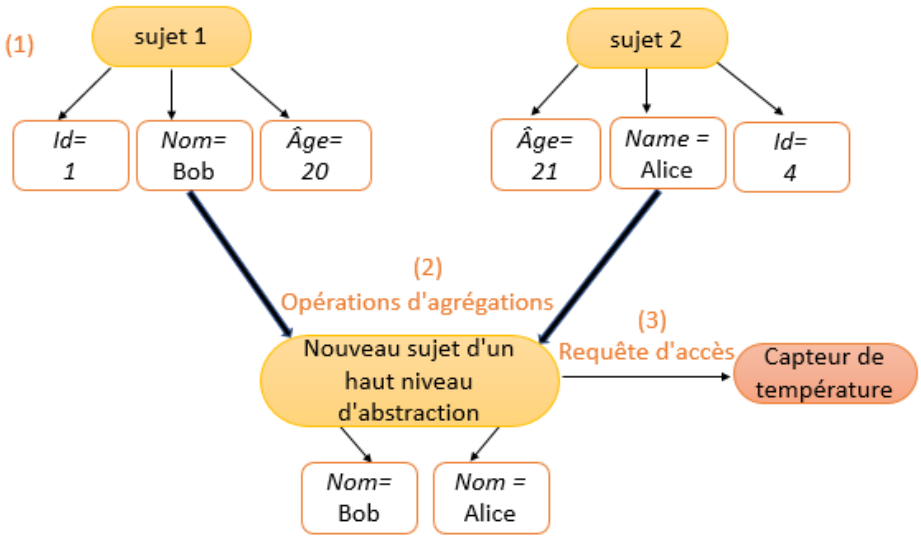

Figure 3.6 – Fédération des attributs dans HoBAC (1)

- **cas 3:** Si Lucy souhaite accéder à l'humidité et au mouvement à l'intérieur de cette maison à 20h05. Au lieu d'envoyer une requête d'accès à chaque capteur, il suffit de créer un seul objet qui représente ces deux capteurs par la fédération de leurs attributs. À l'aide des opérations d'agrégations, l'attribut *"Valeur"* de ces deux objets peut être impliqué dans la création d'un nouvel objet avec un haut niveau d'abstraction. Ensuite, une seule requête d'accès serait envoyée par Lucy à cet objet comme illustré dans la Figure 3.7.

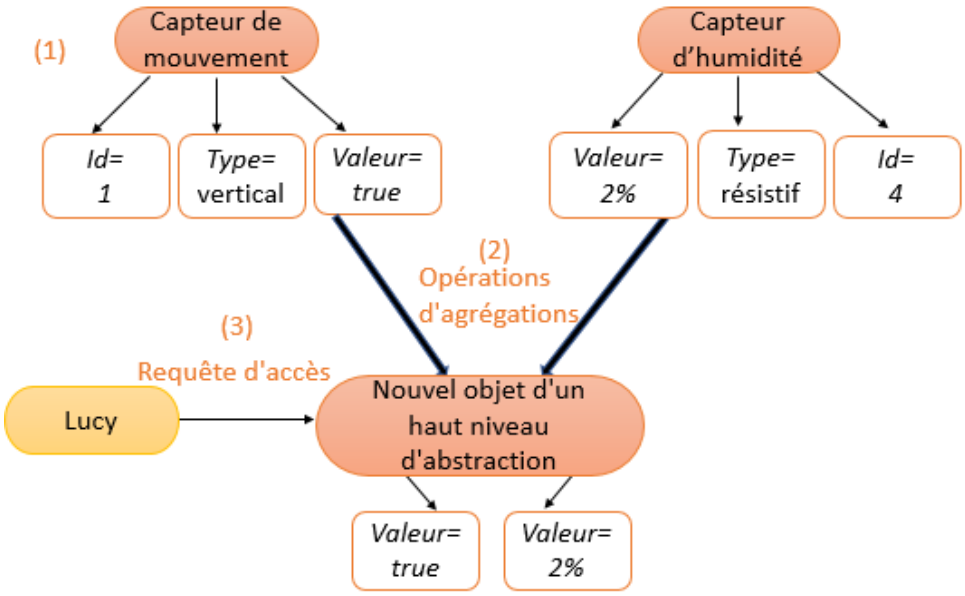

Figure 3.7 – Fédération des attributs dans HoBAC (2)

3.3 CONCLUSION

Après avoir présenté le défi de sécurité dans le chapitre précédent et les différents modèles de contrôle d'accès proposés afin de relever ce défi, nous avons persisté sur la nécessité de mettre en place un modèle de contrôle d'accès flexible, général et adapté aux systèmes l'IoT et non-IoT.

Tout au long de ce chapitre, nous avons introduit la contribution de ce travail de recherche qui consiste à présenter les fondements théoriques de HoBAC, un nouveau modèle de contrôle d'accès adapté aux systèmes IoT et non-IoT qui est une généralisation du modèle ABAC. Nous avons introduit les principaux concepts de HoBAC et leurs relations. Le chapitre suivant impliquera l'architecture générale de HoBAC, ainsi que deux instances de ce modèle afin de montrer son efficacité et son adaptabilité aux différents systèmes. Nous présentons également la création d'une application Web pour l'administration de politique d'accès basée sur HoBAC et l'implémentation de l'insanciation du modèle HoBAC de base à l'aide de la Policy Machine.

CHAPITRE 4

IMPLÉMENTATION DU PROTOTYPE DU MODÈLE HOBAC

4.1 INTRODUCTION

Nous venons de présenter dans le chapitre précédent intitulé « Nouveau modèle de contrôle d'accès basé sur ABAC et les fonctions d'ordre supérieures » les fondements théoriques d'un nouveau modèle de contrôle d'accès nommé Higher-order Attribute-Based Access Control (HoBAC) ainsi qu'une vue générale de ses principaux concepts et leurs relations.

Ce présent chapitre a pour objectif de traiter la traduction pratique du modèle théorique afin de montrer l'applicabilité de ses concepts de base. Nous allons tout d'abord introduire les principaux composants et l'architecture générale de HoBAC. Nous allons ensuite présenter un cas d'utilisation de la famille des modèles HoBAC en introduisant deux instances qui montrent l'efficacité de ce modèle. Nous introduisons ainsi dans ce chapitre les principales étapes d'implémentation du prototype du modèle HoBAC, ainsi que l'implémentation d'une instance du modèle HoBAC de base à l'aide de la Policy Machine (PM) afin d'illustrer l'application d'une politique de contrôle d'accès.

4.2 COMPOSANTS DE BASE DU MODÈLE HOBAC

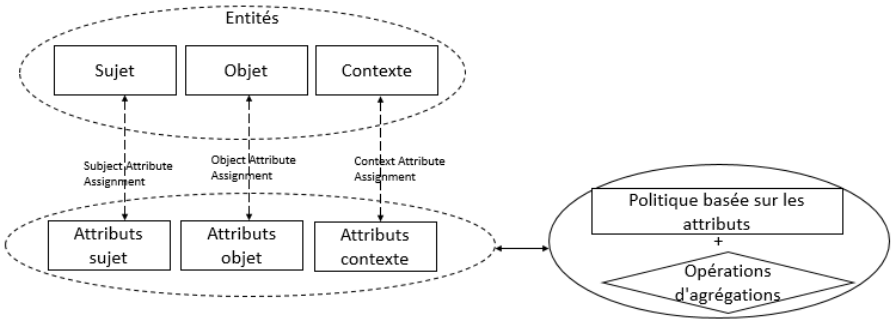

Figure 4.1 – Les principaux composants de HoBAC

Une vue générale des principaux composants de notre modèle de contrôle d'accès est illustrée dans la Figure 4.1.

- Entités: dans le modèle HoBAC, les trois concepts de base du modèle ABAC (sujet, objet et contexte) ont été factorisés dans un nouveau concept nommé entité.
  - Sujet: c'est l'entité qui demande l'accès afin d'effectuer des opérations sur des objets, ces opérations pourraient être (lecture, modification, suppression, exécution, etc).
  - Objet: c'est une ressource système pour laquelle l'accès doit être contrôlé et géré, tels que les périphériques, les fichiers, les enregistrements, les programmes, etc. Il s'agit de la ressource demandée par un sujet.
  - Contexte: il s'agit du contexte opérationnel dans lequel se produisent les demandes d'accès des sujets. Il peut inclure l'heure actuelle, le jour de la semaine, l'emplacement d'un utilisateur ou le niveau de menace actuel.
- Attributs: HoBAC définit les attributs des entités à l'aide d'un id et une valeur, dont l'id représente un identifiant unique, tandis que la valeur est une valeur d'un certain type.

- Politique basée sur les attributs: une politique d'accès est la représentation de règles permettant de déterminer si une demande d'accès doit être autorisée ou refusée. Dans le modèle HoBAC est en tant que généralisation du modèle ABAC, les décisions d'accès sont basées sur une stratégie basée sur les attributs qui évalue les valeurs des attributs provenant principalement du sujet, de l'objet et du contexte et vérifie ainsi un ensemble de règles spécifiées en termes de ces attributs.

- Opérations d'agrégations: ces opérations permettent la fédération des attributs provenant des entités. Donc, un objet (sujet ou contexte) peut être impliqué dans la création d'un objet (sujet ou contexte) avec un haut niveau d'abstraction à l'aide d'un ou plusieurs attributs.

  Un sujet peut avoir des attributs décrivant son nom, son âge, etc. Tandis qu'un objet peut avoir des attributs décrivant son propriétaire, son type, etc. Supposons que deux sujets demandent d'effectuer des opérations sur le même objet, en suivant la politique d'accès du modèle HoBAC, un seul sujet avec un haut niveau d'abstraction serait créé par la fédération des attributs de ces sujets existants à l'aide des opérations d'agrégations, puis une seule demande d'accès serait envoyée par ce nouveau sujet à cet objet.

4.3 L'ARCHITECTURE GÉNÉRALE DU MODÈLE HOBAC

Dans le modèle de contrôle d'accès basé sur les attributs (ABAC), les règles d'accès sont spécifiées à l'aide d'un ensemble d'attributs provenant principalement des sujets, des objets et des contextes. Afin qu'un sujet puisse accéder à un objet, il est nécessaire qu'il dispose des attributs (avec les bonnes valeurs) pour correspondre aux attributs (et les valeurs) d'un objet tel que spécifié dans une règle de politique de contrôle d'accès. De même, un objet ne peut être accédé que s'il dispose des attributs impliqués dans une règle de contrôle d'accès. Après chaque requête d'accès, les attributs et les règles

d'accès seront évalués afin de fournir la décision d'accès (accepter ou refuser l'accès à la ressource demandée).

Higher-order Attribute-Based Access Control (HoBAC) est un nouveau modèle de contrôle d'accès qui est une généralisation du modèle ABAC, ce nouveau modèle permet d'assurer la sécurité des données en offrant plus de flexibilité dans l'application de sa politique d'accès.

L'architecture du modèle HoBAC introduite dans la Figure 4.2 est inspirée de l'architecture du mécanisme de contrôle d'accès basé sur les attributs qui montre les principaux points fonctionnels de ce dernier [31].

Les différents modules (PEP, PDP, PAP, PIP) utilisés dans le processus d'autorisation de notre modèle sont les modules de base de l'architecture XACML illustrée dans la Figure 2.7. Différentes étapes seront effectuées afin de prendre la décision d'accès, ces étapes sont présentées ci-dessous :

- Un sujet envoie une requête d'accès au Policy Enforcement Point (PEP) afin d'effectuer des opérations sur certains objets.
- Après la réception de la requête d'accès de ce sujet, le PEP envoie cette demande au PDP (Policy Decision Point).
- Le PDP interroge à la fois le Policy Information Point (PIP) qui sert comme une source de récupération des attributs et le Policy Administration Point (PAP) qui fournit une interface pour la création et la gestion des politiques de contrôle d'accès. Ces attributs et politiques sont nécessaires pour la politique d'évaluation afin de fournir les informations nécessaires au PDP pour prendre la décision d'accès.
  - Après l'évaluation des attributs et s'il existe une règle de contrôle d'accès

qui autorise à ce sujet d'effectuer les opérations sur ces objets, un seul objet avec un haut niveau d'abstraction serait créé par la fédération des attributs de ces objets à l'aide des opérations d'agrégations afin d'y accéder avec une seule requête d'accès. Sinon, le PDP selon les informations fournies par le PIP et le PAP déduit que la demande d'accès doit être refusée.

- Le PDP informe le PEP pour autoriser ou refuser la demande d'accès de ce sujet.

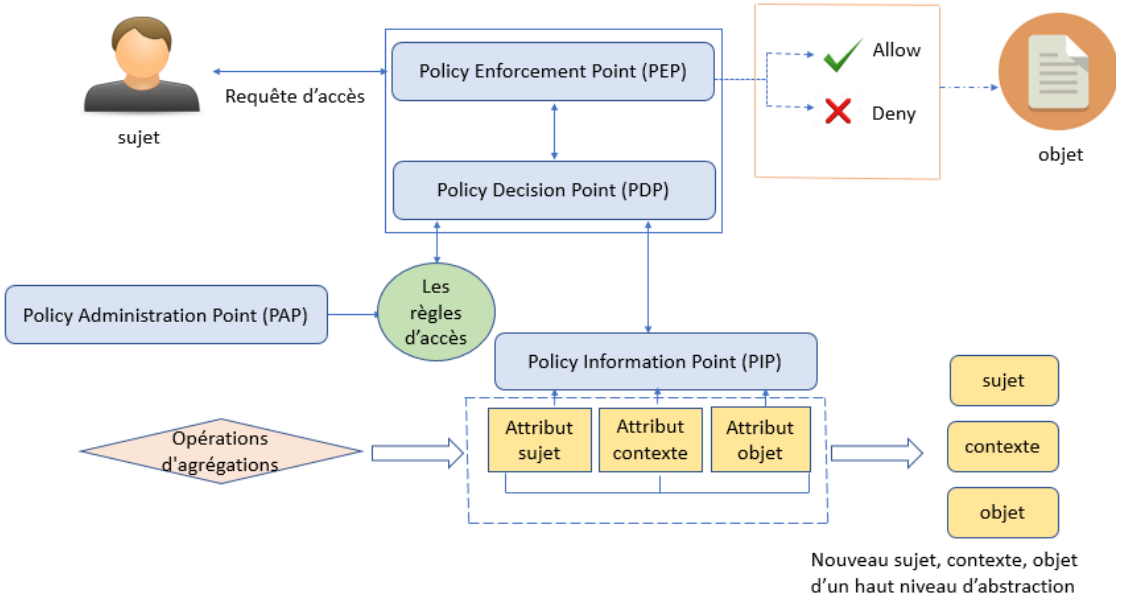

Figure 4.2 – L'architecture générale du modèle HoBAC

La flexibilité de ce nouveau mécanisme de contrôle d'accès (HoBAC) apparaît lorsqu'un sujet souhaite accéder à différents objets en même temps, au lieu d'envoyer une requête à chaque objet, un seul objet avec un haut niveau d'abstraction serait créé par la fédération des attributs de ces derniers à l'aide des opérations d'agrégations et une seule requête serait envoyée.

De même, si différents sujets souhaitent accéder à un seul objet, au lieu que chacun envoie sa propre requête, un seul sujet serait créé par la fédération des attributs de ces derniers à l'aide des opérations d'agrégations et une seule requête serait envoyée à cet objet. Donc, ce mécanisme d'abstraction représente une couche de sécurité qui permet d'empêcher la manipulation directe des sujets et des objets de bas niveau.

4.4 CAS D'UTILISATION DE LA FAMILLE DES MODÈLES HOBAC

En tant que généralisation du modèle ABAC, le modèle HoBAC représente les modèles de contrôle d'accès où chaque famille de modèles est définie par le nombre des niveaux de son espace d'attribut. Ainsi, le même niveau d'espace d'attribut génère des instances (modèles de contrôle d'accès) qui se différencient par l'ordre partiel de leurs attributs.

Afin de montrer l'efficacité de notre nouveau modèle de contrôle d'accès ainsi qu'il est assez général pour exprimer différentes politiques de contrôle d'accès, on a présenté deux instances du modèle théorique tel que présenté dans la Figure 4.3.

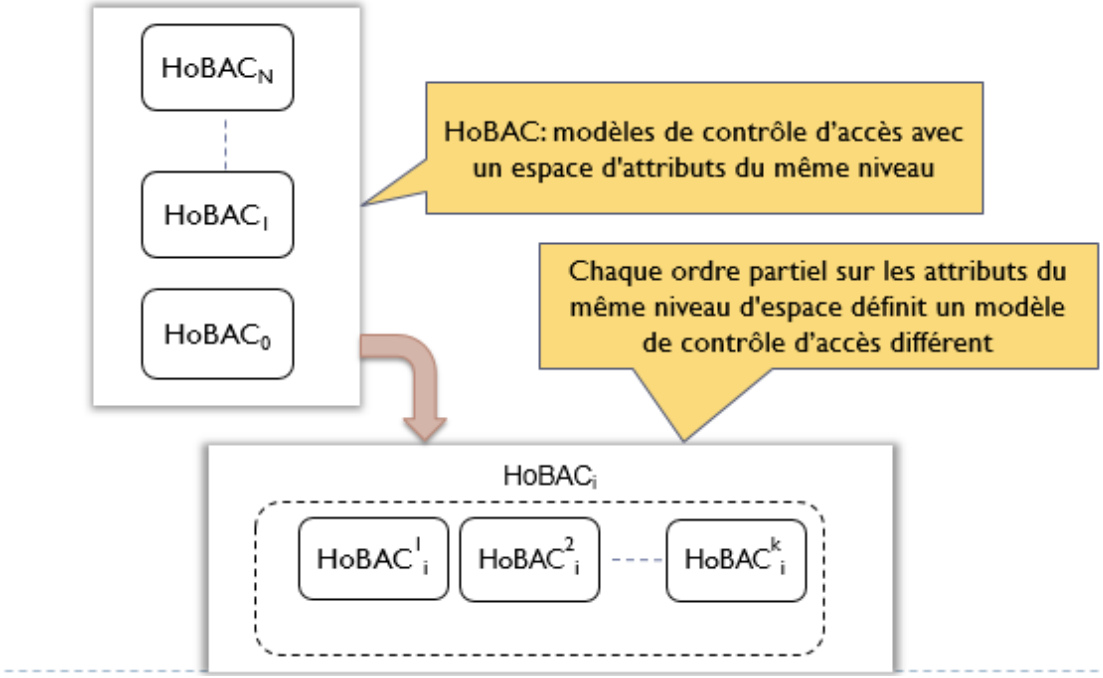

Figure 4.3 – HoBAC: famille de modèles

Ci-dessous, nous présentons des modèles de contrôle d'accès qui sont des instances du modèle HoBAC général.

- $HoBAC_i$

  Fait référence à un modèle HoBAC dans lequel l'espace d'attribut est composé de *i* attributs.

- $HoBAC_0$

  Le modèle de base de HoBAC est $HoBAC_0$ qui correspond au modèle ABAC d'origine. Dans ce modèle, il n'y a pas de niveaux et l'espace d'attribut est plat, de même que les espaces d'objet, de sujet et de contexte.

- $HoBAC_4$

  Le nombre de niveaux d'attributs d'une entité d'un modèle HoBAC peut être

défini en fonction des considérations commerciales dans chaque organisation, ou il peut dépendre du nombre de couches de calcul du système ciblé.

Par exemple, dans un cadre IoT où nous avons quatre niveaux de calcul: périphériques IoT, couche Edge, couche Fog et couche Cloud, un HoBAC avec quatre niveaux ($HoBAC_4$) peut convenir pour contrôler l'accès à ce paramètre Figure 4.4.

Il est à noter que le nombre de niveaux peut être lié à la topologie et au nombre de niveaux de calcul du système. Il peut obéir à une logique totalement indépendante de la structure physique.

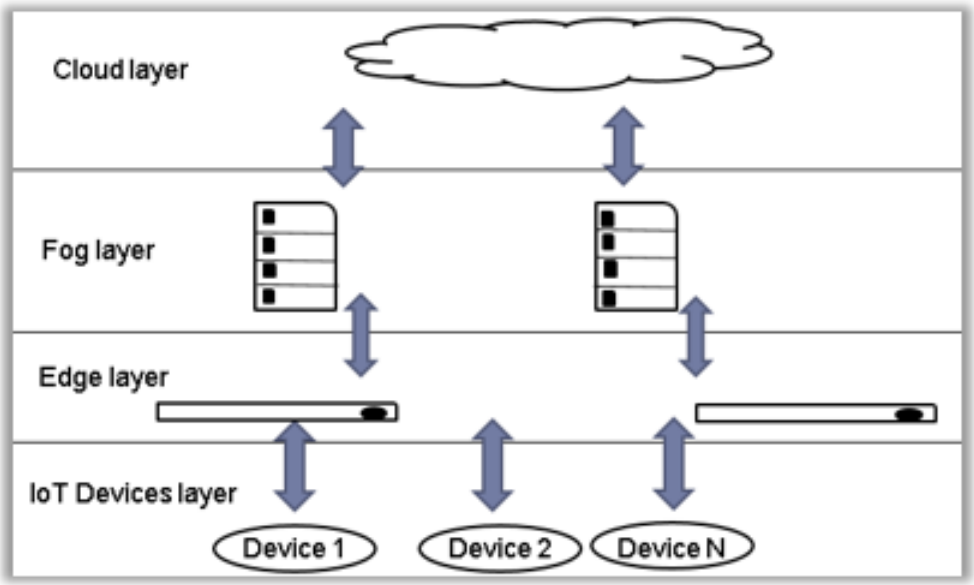

Figure 4.4 – ($HoBAC_4$) pour une architecture IoT à 4 couches

Donc, le modèle HoBAC est assez général et adapté à la fois aux systèmes IoT et non-IoT, il permet de mettre en oeuvre des stratégies de contrôle d'accès IoT et non-IoT basées sur des hiérarchies d'entités.

4.5 HOBAC: PRÉSENTATION DU PROTOTYPE

Pour le développement de notre prototype, nous souhaitons implémenter le modèle de contrôle d'accès HoBAC théorique avec ses concepts de base et leurs relations qui sont présentés précédemment dans ce mémoire. Nous allons tout d'abord introduire le modèle relationnel du modèle HoBAC théorique afin de modéliser ses différents concepts sous forme de tables et établir les relations entre ces derniers, cela va permettre de faciliter l'implémentation de notre prototype.

Nous allons créer une application Web pour l'administration de politique d'accès basée sur HoBAC afin de montrer l'applicabilité de ses concepts de base et leurs relations. L'application Web comporte un module d'authentification qui consiste à assurer l'identité des utilisateurs. Cette application va nous permettre de créer les différentes entités (sujets, objets et contextes) à partir des types d'entités et assigner à chaque entité un ensemble d'attributs, ainsi de définir une politique d'accès par la création des règles permettant de déterminer si une demande d'accès doit être autorisée ou refusée. Puisque le modèle HoBAC est une généralisation du modèle ABAC, les décisions d'accès seront basées sur une politique basée sur les attributs qui évalue les valeurs des attributs provenant principalement du sujet, de l'objet et du contexte et vérifie ainsi les règles d'accès spécifiées en termes de ces attributs.

4.5.1 LANGAGE DE DÉVELOPPEMENT DU PROTOTYPE

Le développement du prototype a été réalisé dans le langage Ruby on Rails (RoR) qui suit le motif de conception modèle-vue-contrôleur (MVC), ce langage est basé sur le principe DRY (don't repeat yourself), ce principe vise à réduire autant que possible la répétition du code afin qu'il soit facile de modifier le cycle de développement. Ce langage est caractérisé par sa souplesse, son dynamisme ainsi que sa syntaxe élégante, notamment

ce qui justifie son choix. Il utilise par défaut la base de données **sqlite3**. (version Rails: 5.1.6 , version ruby: 2.2.6p396 (2016-11-15 revision 56800)[i386-mingw32]).

4.5.2 MODÈLE RELATIONNEL

Afin de visualiser le modèle HoBAC théorique, il est intéressant de créer un modèle relationnel qui permet de faciliter la représentation de notre prototype. Le modèle relationnel que nous avons introduit dans la Figure 4.5 a été défini à partir des concepts de base du modèle HoBAC théorique et ses relations qui sont présentés précédemment. Chaque concept a été modélisé sous forme de table à deux dimensions ayant ses propres attributs, nous avons ensuite établi les relations existantes entre les tables selon les fondements théoriques de HoBAC.

EntityType est la classe principale à partir de laquelle nous pouvons créer les nouvelles entités (sujets, objets ou contextes) qui sont des instances de la classe principale. EntityType peut avoir un ou plusieurs AttributeType qui représentent une liste de types d'attributs.

ObjectType, SubjectType et ContexteType sont des sous-classes qui héritent de la classe EntityType dont le champ « type » dans la classe EntityType est par défaut réservé pour stocker le nom de la sous-classes après sa création.

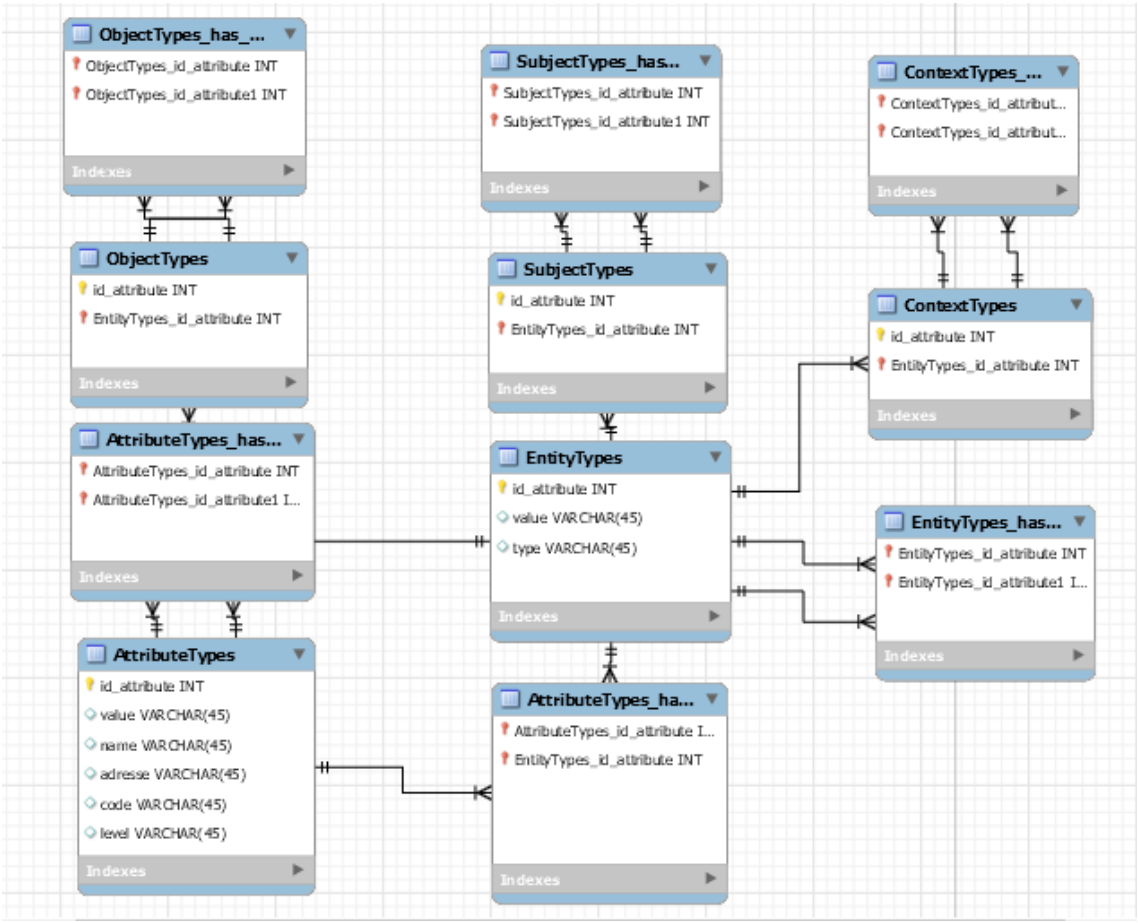

Figure 4.5 – Le modèle relationnel de HoBAC

L'étape suivante après la création du modèle relationnel consiste à développer une application Web basée sur ce modèle en utilisant le cadre de développement Ruby on Rails. Dans le Ruby on Rails nous avons la possibilité de créer les tables avec leurs attributs comme étant des modèles à l'aide du générateur rails, les attributs sont automatiquement ajoutés à la table dans la base de données et mappés au modèle. Nous allons traduire les relations existantes entre les tables du modèle relationnel en associations rails.

La Figure 4.6 présente le diagramme états-transitions en UML (Unified Modeling Language) qui illustre un cas d'utilisation lorsqu'un utilisateur souhaite accéder à une ressource donnée. Le résultat d'évaluation de la politique de contrôle d'accès peut

refuser l'accès de cet utilisateur après l'évaluation des attributs et s'il n'y a aucune règle qui autorise spécifiquement son accès. Sinon, une autorisation d'accès serait fournie à cet utilisateur, si ce dernier demande un accès à plusieurs objets, une fédération des attributs des objets demandés serait effectuée pour créer un seul objet afin que cet utilisateur l'accède par une seule requête.

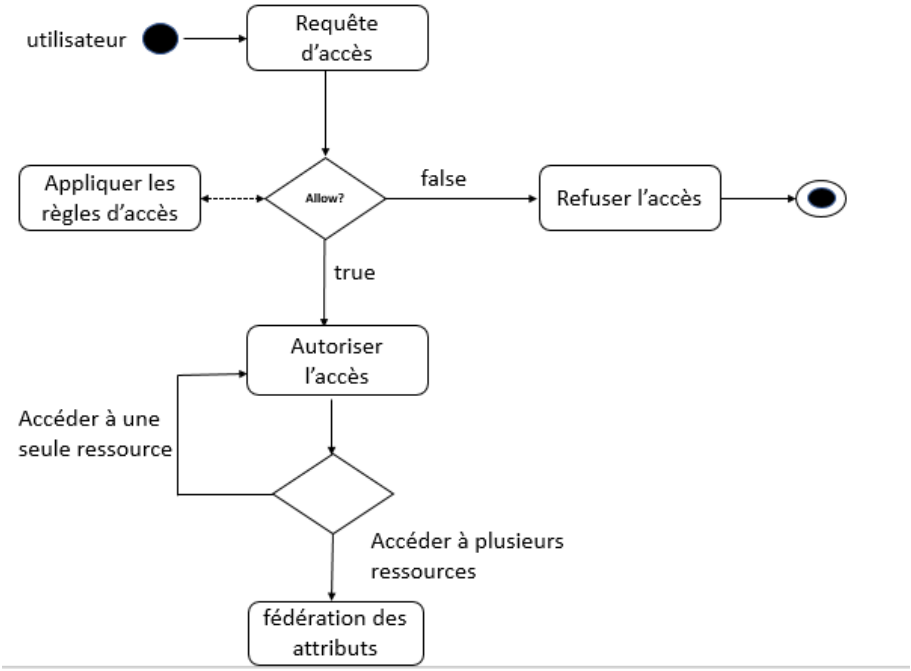

Figure 4.6 – Cas d'utilisation qui exprime l'autorisation

4.5.3 AUTHENTIFICATION

L'authentification s'effectue à l'aide de *Devise*, un module d'authentification flexible et populaire qui peut être intégré dans les applications Ruby on Rails (Rafael [55]).

Devise est basé sur le concept de modularité dont il fournit une gamme complète de fonctionnalités et peut-être configurée selon les exigences de l'application. Nous avons utilisé l'authentification afin d'assurer l'identité des utilisateurs.

L'interface d'accueil présente l'espace d'authentification, si un utilisateur possède son

propre compte il peut se connecter dans l'espace connexion contenant un champ pour l'email et un autre pour le mot de passe. Sinon il peut passer à l'espace d'inscription qui est conçu pour tout nouvel utilisateur souhaitant s'inscrire dans la base de données.

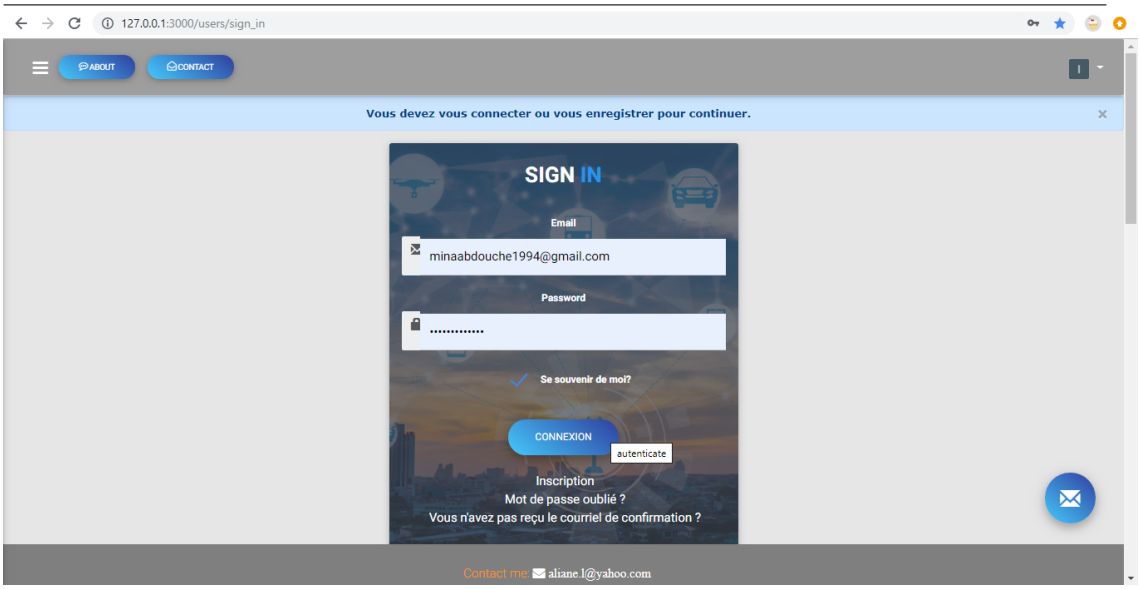

Figure 4.7 – Interface d'acceuil

Figure 4.8 – La liste des formulaires

4.5.4 LA CRÉATION DES ENTITÉS

Après l'authentification, l'utilisateur est redirigé vers l'interface principale afin de créer des nouvelles entités (sujets, objets ou contextes) selon le besoin du système. Après chaque création, ces dernières vont être sauvegardées dans la classe EntityType avec leur propre type grâce à la relation d'héritage existante.

- Le sujet créé va être sauvegardé dans EntityType avec le type « SubjectType ».
- L'objet créé va être sauvegardé dans EntityType avec le type « ObjectType ».
- Le contexte créé va être sauvegardé dans EntityType avec le type « ContexteType ».

Afin de rendre opérationnel les concepts précédents, nous allons créer un nouvel objet avec un id = 1, et name = filex, cet objet créé est une instance de ObjectType tel que présenté dans la Figure 4.9.

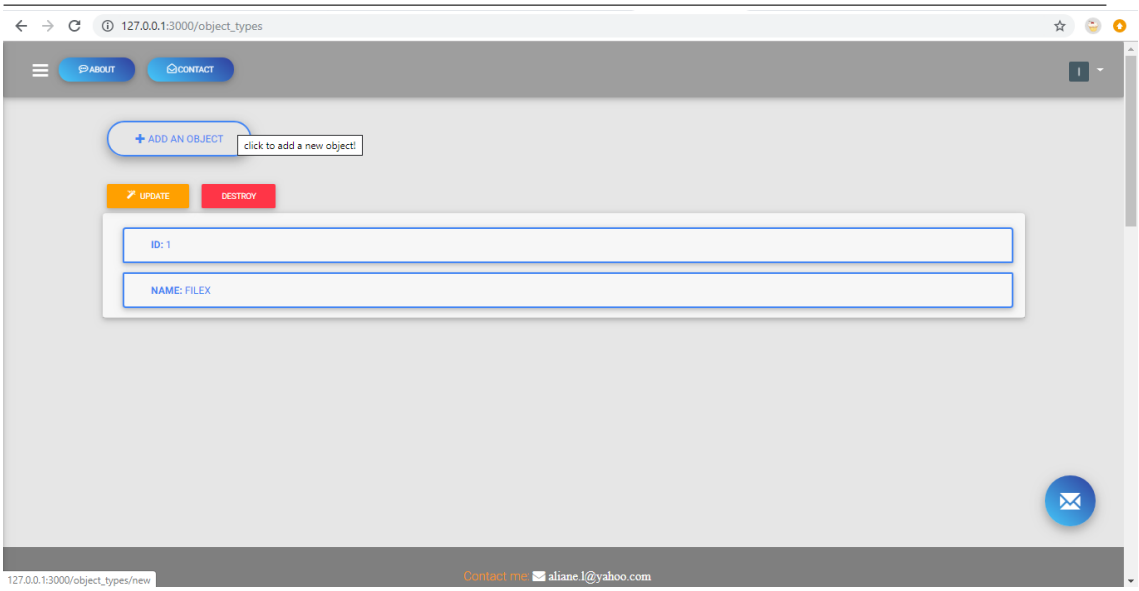

Figure 4.9 – La liste des objets

Le nouvel objet créé va être sauvegardé comme étant une entité de type « ObjectType » tel que montré dans la Figure 4.10.

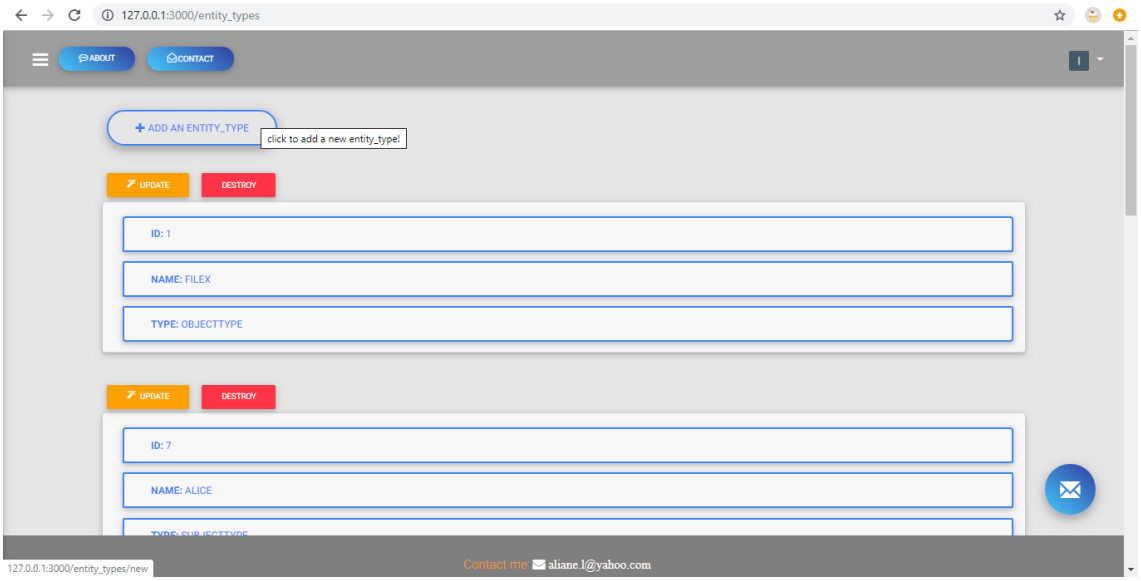

Figure 4.10 – La liste des entités

Comme nous l'avons cité précédemment dans les fondements théoriques de HoBAC, une entité peut avoir plusieurs attributs. Pour cela nous allons appliquer une modification à l'entité qui a été créée en lui ajoutant des attributs, les attributs ajoutés à cette entité vont être automatiquement sauvegardés dans l'objet à travers la relation d'héritage tel que présenté dans la Figure 4.11.

Il est à noter que les ID de ces attributs sont générés automatiquement.

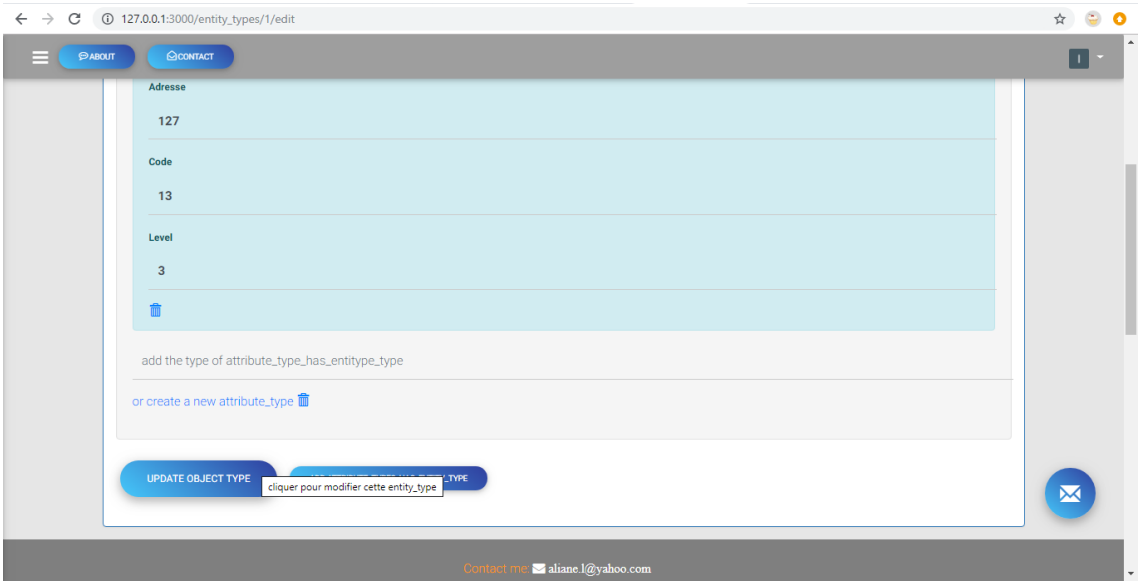

Figure 4.11 – L'ajout des attributs à l'entité

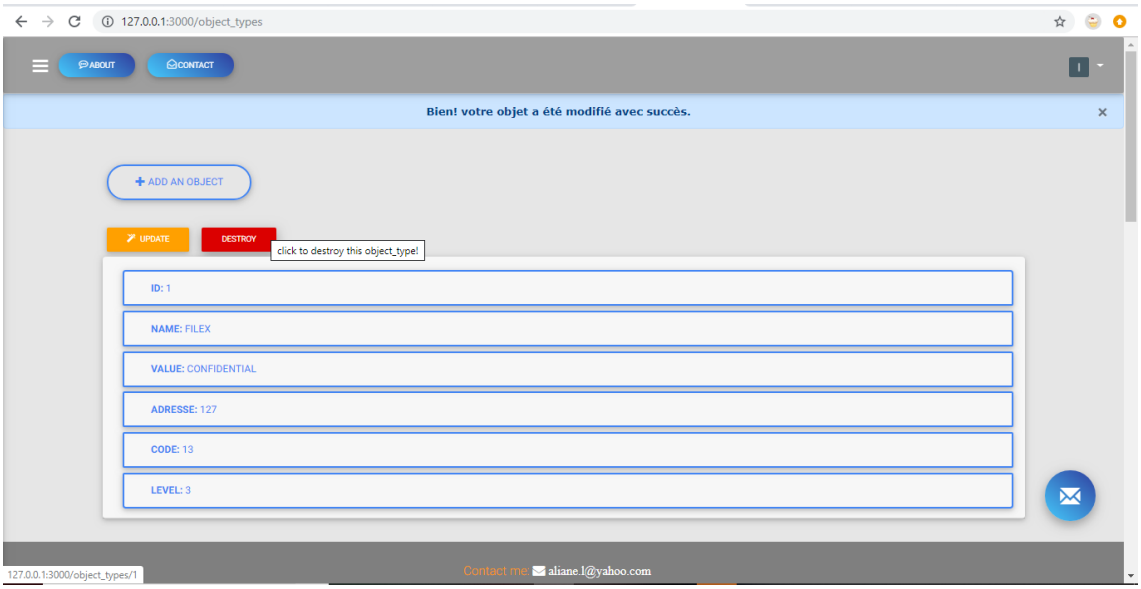

Figure 4.12 – La liste des objets

4.5.5 LES RÈGLES D'ACCÈS

Afin d'assurer et de protéger les objets des opérations non autorisées qui peuvent inclure la lecture, la création, l'édition et la suppression. Il est nécessaire d'établir une politique

d'accès. Notre modèle HoBAC est une généralisation du modèle ABAC qui est basé sur les attributs provenant primitivement des entités (objet, sujet et contexte). Donc, cette politique décrit quelles opérations peuvent être effectuées sur ces objets, par quel sujet et dans quel contexte ces opérations peuvent être effectuées.

Pour cela nous avons créé une liste de règles d'accès tel que montré dans la Figure 4.13. Ces règles d'accès sont spécifiées en fonction des attributs provenant des objets, des sujets et des contextes (conditions d'environnement).

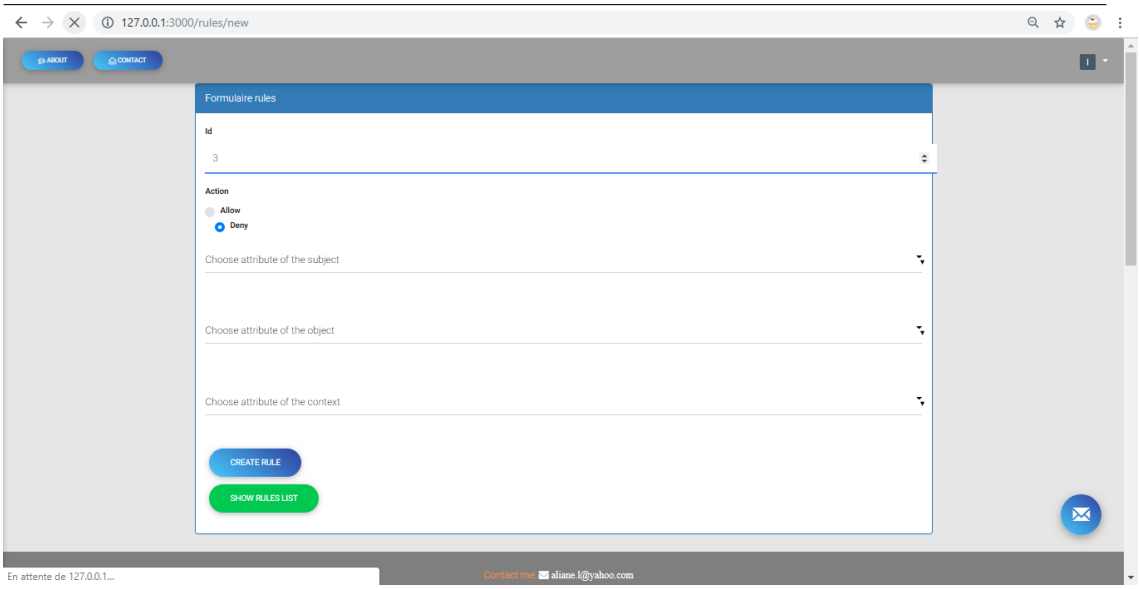

Figure 4.13 – L'ajout d'une règles d'accès

Lorsqu'un sujet demande l'accès à un objet, notre politique de contrôle d'accès et après avoir évalué la requête reçue en se basant dans sa forme la plus simple, sur l'égalité des attributs, autorise ou refuse donc l'accès de ce sujet en fonction des attributs et de l'ensemble de règles d'accès spécifiées en termes de ces attributs. La décision de contrôle d'accès est représentée par une Action, *Allow* : pour autoriser l'accès, *Deny* : pour refuser l'accès tel que montré dans la Figure 4.14.

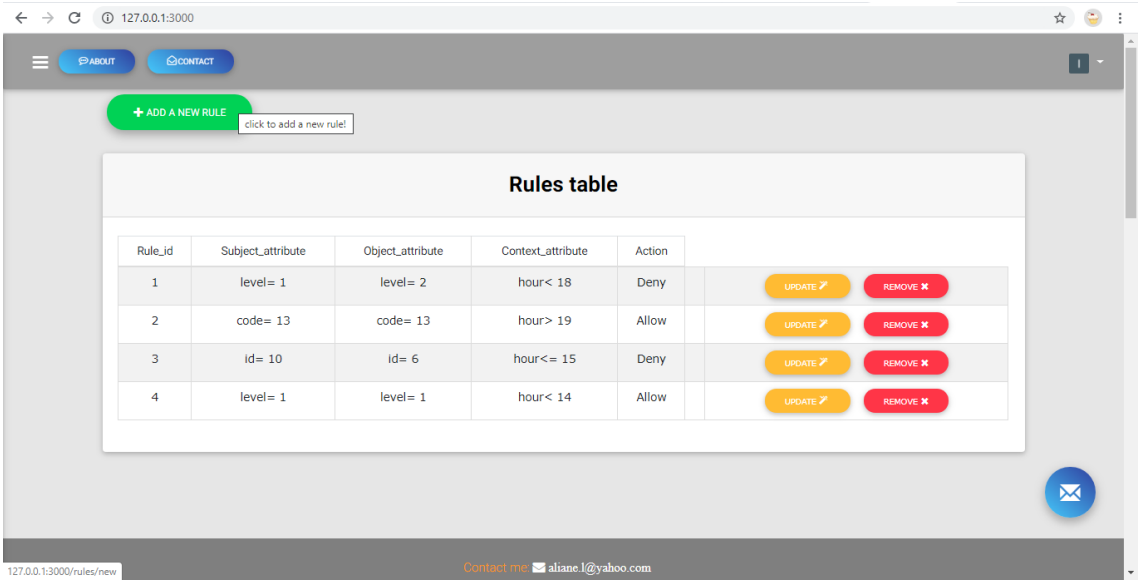

Figure 4.14 – Visualisation de la liste des règles d'accès

4.6 IMPLÉMENTATION DE HOBAC À L'AIDE DE LA POLICY MACHINE (PM)

Après avoir présenté précédemment l'application Web que nous avons créé pour l'administration de politique d'accès basée sur HoBAC, nous souhaitons dans cette section implémenter l'instanciation du modèle HoBAC de base ($HoBAC_0$) à l'aide de la Policy Machine (PM), ce modèle correspond au modèle ABAC d'origine où il n'y a pas de niveaux et l'espace d'attribut est plat, de même que les espaces d'objet, de sujet et de contexte.

4.6.1 LA POLICY MACHINE (PM)

Selon [24], la Policy Machine (PM) est un cadre de contrôle d'accès qui n'est pas une extension d'un modèle ou d'un cadre existant, mais plutôt une redéfinition du contrôle d'accès en termes d'un ensemble normalisé et générique de relations et de fonctions permettant la définition et l'application des politiques de contrôle d'accès.

La PM permet de fournir un cadre général pour prendre en charge un large éventail de politiques basées sur des attributs ou des combinaisons de politiques par le biais d'un mécanisme unique qui nécessite des changements dans sa configuration de données. Dans la policy machine la politique d'accès est exprimée par le biais de configuration de relations de quatre types:

- Affectation: cette relation est représentée par un tuple (x, y) pour spécifier l'affectation de l'élément x à l'élément y. L'ensemble d'entités utilisé dans cette relation comprend les utilisateurs, les attributs des utilisateurs, les attributs des objets et les classes de politique d'accès.

- Association: l'association permet de dériver les privilèges, elle est représentée par (u_a–op–o_a) dont u_a est un attribut de l'utilisateur, op est l'opération à effectuer et o_a est l'attribut de l'objet. Cette relation signifie que l'utilisateur u a des

privilèges pour effectuer une opération op sur l'objet o. Les privilèges dans la Policy Machine sont indirectement gérés via un haut niveau d'abstractions, la PM comprend quatre de ces abstractions: les attributs d'utilisateur (UA), les attributs d'objet (OA), l'ensemble d'opérations et les classes de politique (PC).

- Interdiction: cette relation exprime les contraintes et les restrictions des droits d'accès sur les utilisateurs, les processus ou les attributs des utilisateurs.

- Obligation: cette relation est appelée la relation de modèle événement/réponse, elle définit les conditions et les méthodes selon lesquelles les données d'état de la politique sont obligées de se changer. Cette relation est représentée par (ep, r), généralement désignée (lorsque ep, faire r), où ep est un modèle d'événement et r est la réponse.

4.6.2 IMPLÉMENTATION DE $HOBAC_0$ AVEC LA POLICY MACHINE

L'objectif principal consiste à développer un cadre d'autorisation général qui peut être facilement utilisé par toute application ou système prenant en charge le service RESTful pour mettre en oeuvre des politiques de contrôle d'accès basées sur les attributs. Nous présentons tout d'abord l'architecture d'autorisation généralisée, nous discutons ensuite à l'aide d'un cas d'utilisation comment le modèle $HoBAC_0$ peut être défini et implémenté dans la PM.

4.6.2.1 Architecture d'autorisation

Nous avons introduit dans la Figure 4.15 une architecture d'autorisation pour configurer et implémenter le modèle $HoBAC_0$ en utilisant la Policy Machine. Cette architecture est généralisée et indépendante de toute application ou système.

L'architecture d'autorisation comprend un client PM et un serveur PM, le côté client est composé d'une API à travers laquelle les utilisateurs demandent l'accès aux ressources protégées et un Policy Enforcement Point (PEP) qui transmet les demandes des utilisateurs au Policy Decision Point (PDP) afin de prendre la décision d'accès. Le serveur PM est composé d'un PDP, un Policy Administration Point (PAP) et d'un Policy Information Point (PIP) qui sert comme une base de données utilisée pour stocker les données de contrôle d'accès liées aux utilisateurs, aux objets, à leurs attributs et relations.

La décision d'accès qui doit être prise par le PDP est basée sur l'identité de l'utilisateur qui a émis la demande, l'opération demandée et la ressource demandée.

Lorsqu'un utilisateur demande l'accès à une ressource protégée à travers l'API, le PEP dans le client PM intercepte la demande d'accès de cet utilisateur puis demande au PDP dans le serveur PM de prendre la décision d'accès. Le PDP et après la réception de la demande d'accès qui lui a été transmise par le PEP interroge à la fois le PAP qui est utilisé pour administrer la base de données PM pour la configuration de la politique d'accès et la base de données PIP afin de récupérer les attributs stockés et les données nécessaires dans l'évaluation de la politique d'accès afin de prendre la décision d'accès (accepter ou de rejeter la demande d'accès).

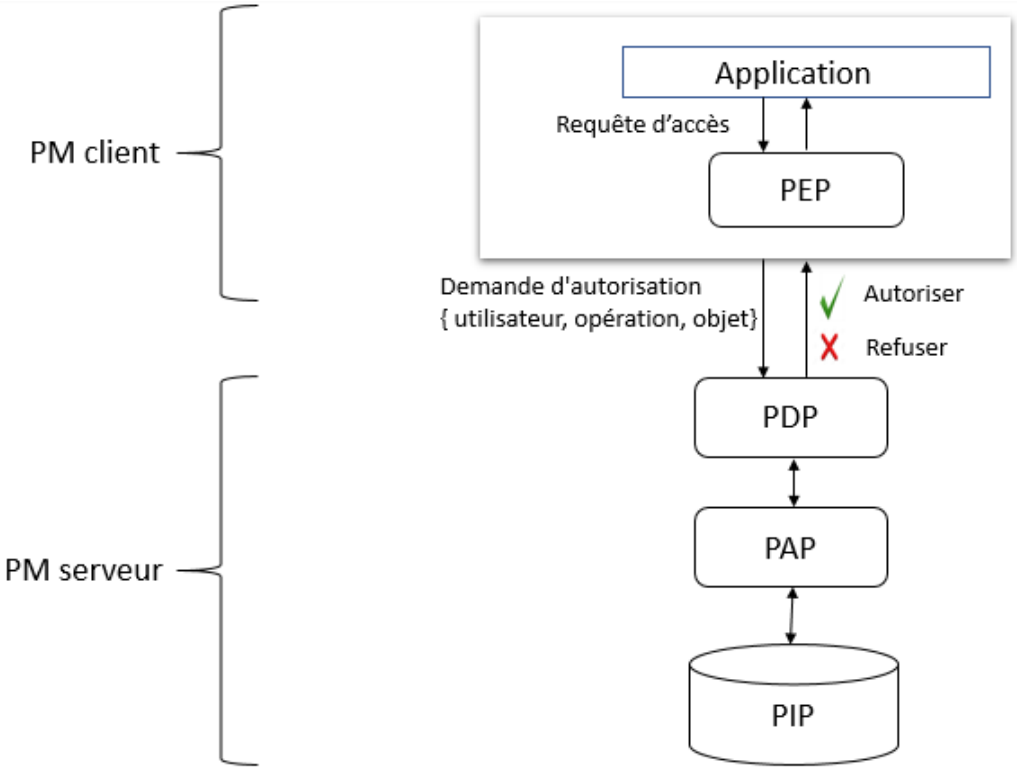

Figure 4.15 – L'architecture d'autorisation du modèle HoBAC

Un utilisateur se connecte à la PM à travers une API qui est fournie par le client PM utilisée pour authentifier les utilisateurs, une connexion réussie ouvre une session utilisateur sur le client PM et fournit un ID de cette session qui serait utilisé dans les demandes ultérieures de cet utilisateur.

Au cours de la session, lorsqu'un utilisateur demande l'accès à une ressource protégée, le client PM communique avec le serveur PM dont l'API envoie une demande HTTP au PEP, le PEP détermine à l'aide des ID de session quel utilisateur demande l'accès et transmet à son tour la demande d'accès au PDP afin de fournir la décision d'autorisation. Le PDP détermine si la demande d'accès qui lui a été transmise par le PEP doit être accordée ou refusée conformément à la politique d'accès définie dans le PAP et les données stockées dans la base de données (PIP).

Une demande d'autorisation comprend l'utilisateur qui demande l'accès, l'opération à effectuer (lire, écrire) et l'objet demandé. Cette demande est implémentée en tant que méthode HTTP, pour cela l'application ou le système doit prendre en charge le service RESTful afin d'envoyer les requêtes HTTP.

4.6.2.2 Cas d'utilisation

Ce cas d'utilisation permet de représenter les fonctionnalités de notre modèle de contrôle d'accès, il inclut des utilisateurs, des objets et leurs attributs ainsi que les relations existantes entre ces derniers. Nous avons configuré et implémenté ce cas d'utilisation dans la PM en utilisant notre architecture d'autorisation présentée précédemment.

Nous avons utilisé le langage *Java* pour le développement et la base de données graphique $Neo4j$ dans ce cas d'utilisation pour stocker les données des utilisateurs, des objets et leurs attributs ainsi que les relations d'affectation et d'association existantes entre ces derniers sous forme d'un graphe (les noeuds représentent les utilisateurs, les objets et leurs attributs et les arcs représentent les relations) comme le montre le graphe illustré dans la Figure 4.16.

Nous avons trois utilisateurs (bob, $lucy$, $alice$) associés aux attributs-utilisateurs ($groupe1$, $groupe2$, $informatique-département$), nous avons ainsi deux objets ($projet1$, $projet2$) qui sont associés aux attributs-objets ($développement$, $déploiement$, $général$) à travers la relation d'affectation . Nous avons créé trois associations entre les attributs des utilisateurs et les attributs des objets afin de spécifier quelles opérations peuvent être effectuées sur quels objets et par quel utilisateur.

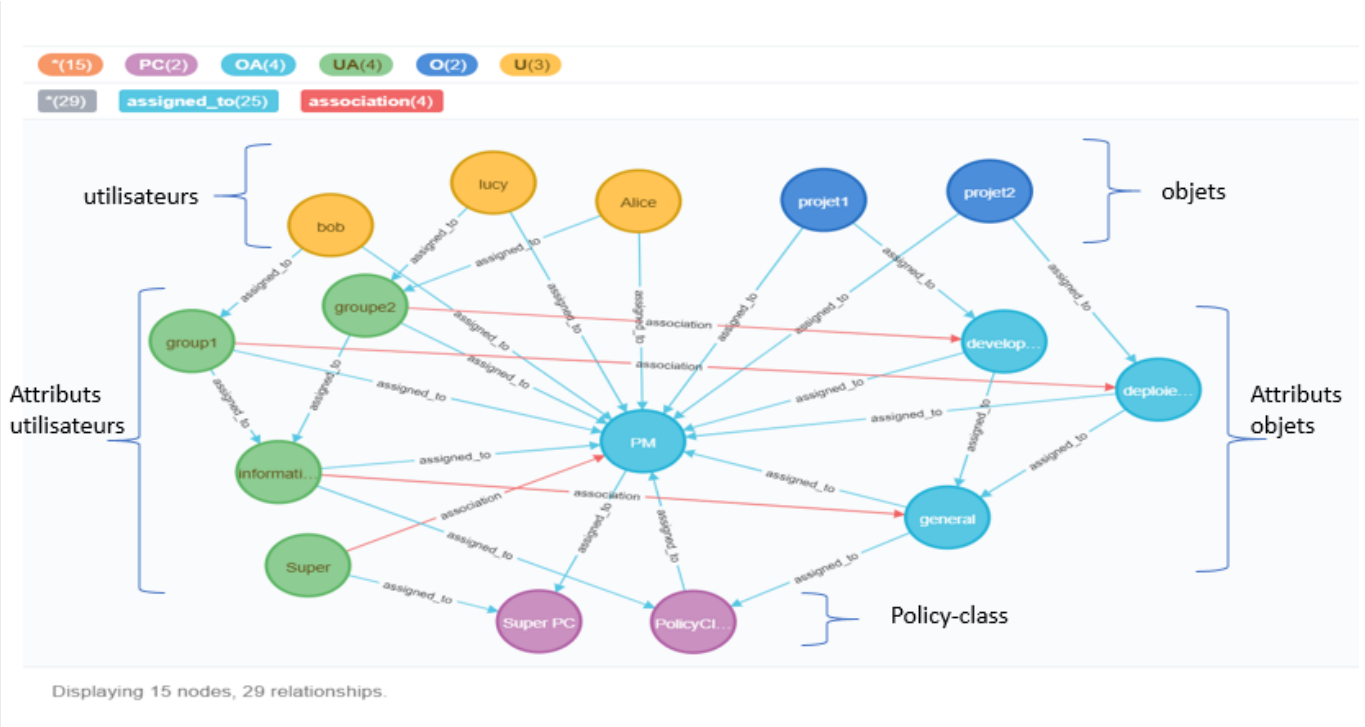

Figure 4.16 – Graphe de la politique d'accès

Dans ce cas d'utilisation, nous définissons une politique de contrôle d'accès pour les opérations de lecture et d'écriture comme spécifié dans le tableau 4.1.

$Politique - d'accès_{lecture,écriture}$		
Attributs-utilisateur	Attributs-objet	opération
groupe1	déploiement	lire
groupe2	développement	lire et écrire
informatique-département	général	lire

Tableau 4.1 – Politique d'accès pour les opérations de lecture, écriture

La politique d'accès est définie en fonction des valeurs des attributs des utilisateurs et les valeurs des attributs des objets comme suit:

- un utilisateur qui est dans le *groupe*1 peut lire tout projet de type *déploiement*.
- un utilisateur qui est dans le *groupe*2 peut écrire et lire tout projet de type *développement*.
- un utilisateur qui est dans le $informatique - département$ peut lire tout projet de type *général*.

Un exemple de la demande d'autorisation et sa réponse est illustré dans la Figure 4.17. La demande est implémentée en tant que méthode HTTP (POST), elle inclut les données d'autorisation sous format JSON (*user*: représente l'utilisateur qui demande l'accès, *target*: représente l'objet demandé, *operation*: représente l'opération à effectuer sur

l'objet), la décision d'autorisation obtenue est une réponse HTTP renvoyée également sous format JSON.

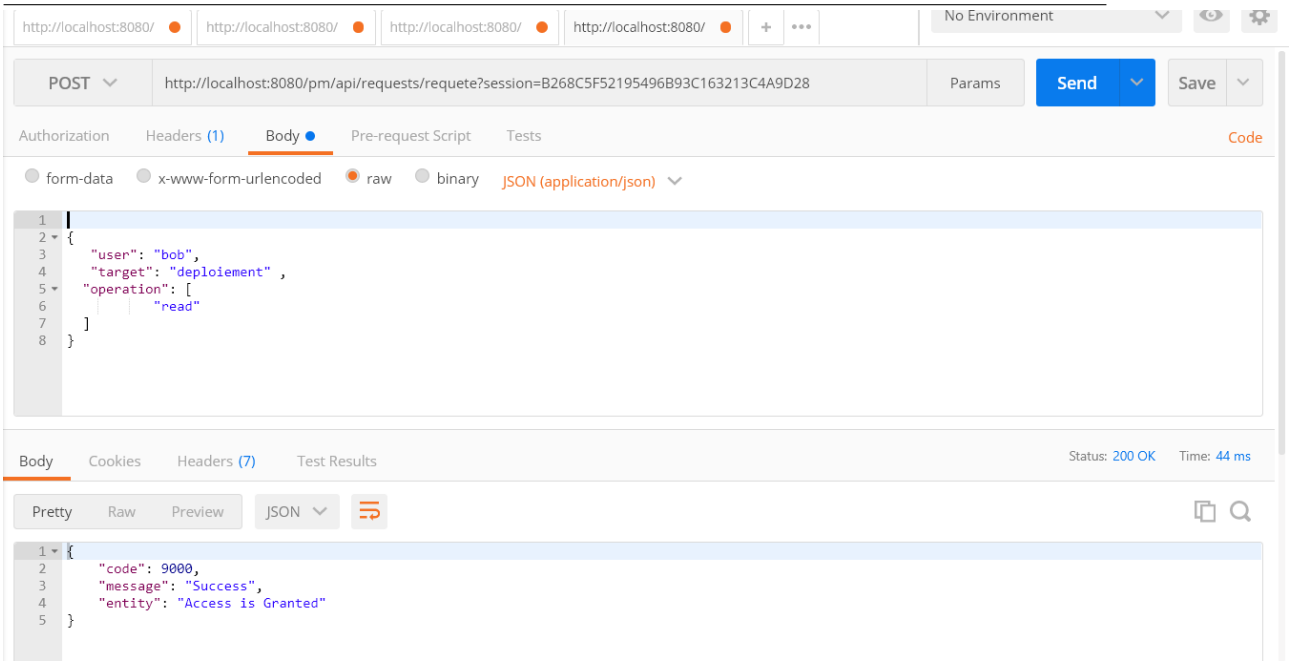

Figure 4.17 – Exemple de la demande d'autorisation et sa réponse

4.7 CONCLUSION

Tout au long de ce chapitre, nous avons présenté les principaux composants et l'architecture générale de HoBAC. Nous avons ainsi présenté deux instances du modèle HoBAC théorique dont la première instanciation génère le modèle ABAC d'origine et la deuxième génère un modèle de contrôle d'accès à quatre couches qui convient aux systèmes IoT, cela a pu montrer l'efficacité de ce dernier ainsi qu'il est assez général pour exprimer différentes politiques de contrôle d'accès. Ainsi, nous avons implémenté un prototype du modèle HoBAC afin de montrer l'applicabilité de ses principaux concepts et leurs relations.

Le travail présenté dans ce chapitre présente le premier pas vers la réalisation de notre perspective, particulièrement dans les trois volets suivants :

- D'abord, la mise en place d'un nouveau modèle de contrôle d'accès flexible et générique qui répond à toutes nos approches proposées, en introduisant ses fondements théoriques et ses concepts de base.
- Ensuite, la présentation de deux instances de HoBAC ($HoBAC_0$ et $HoBAC_4$) permettant de montrer son adaptabilité et qu'il est assez général pour exprimer différentes politiques de contrôle d'accès IoT ou non-IoT.
- Enfin, l'implémentation d'une application Web pour l'administration de politique de contrôle d'accès de type HoBAC, ainsi que l'implémentation de l'instanciation du modèle HoBAC de base ($HoBAC_0$) à l'aide de la Policy Machine (PM).

Dans le chapitre suivant nous allons présenter le résumé des objectifs, le travail accompli, les limitations de notre approche proposée ainsi que nos perspectives pour le développement futur.

CHAPITRE 5

CONCLUSION

5.1 RÉSUMÉ DES OBJECTIFS

La sécurité des données est devenue un enjeu majeur dans le monde de l'IoT dans ses différents domaines, essentiellement lorsque les données sont très sensibles et critiques, comme c'est le cas des données personnelles et particulièrement les données médicales, car toute faille de sécurité dans ces systèmes critiques peut conduire à des conséquences catastrophiques.

Dans l'IoT, garantir et assurer la sécurité des données est l'un des défis les plus difficiles à relever. La notion de modèle de contrôle d'accès est apparue avec les besoins spécifiques en matière de sécurité des systèmes informatiques, avec le large déploiement des dispositifs IoT et les enjeux de leur sécurité le contrôle d'accès est devenu de plus en plus primordial. Un modèle de contrôle d'accès a pour objectif d'assurer la sécurité des données en empêchant l'accès illégitime à ces dernières en fonction des stratégies d'accès spécifiques. Le mécanisme de contrôle d'accès renforce les trois principales propriétés de la sécurité concernant: la disponibilité, la confidentialité et l'intégrité des

données.

Les modèles traditionnels de contrôle d'accès sont souvent rigides et non appropriés pour exprimer les autorisations d'accès dans un environnement dynamique. Bien que différents modèles de contrôle d'accès existent dans la littérature, le besoin d'un nouveau modèle de contrôle d'accès générique qui convient à la fois aux systèmes IoT et non IoT est également nécessaire. L'objectif principal de ce travail de recherche consiste à présenter un nouveau modèle de contrôle d'accès général et flexible afin d'assurer la sécurité des données dans les systèmes IoT et non IoT.

Nous résumons par la suite dans les sections suivantes le travail accompli, ses limitations ainsi que nos perspectives de développement futur.

5.2 TRAVAIL ACCOMPLI

Dans ce travail de recherche, nous avons introduit les fondements théoriques de Higher-order Attribute-Based Access Control (HoBAC), un nouveau modèle de contrôle d'accès flexible, global qui est une généralisation du modèle ABAC. Nous avons montré l'efficacité de HoBAC à l'aide de deux instances, la première instance permet de générer un modèle équivalent au modèle ABAC d'origine ($HoBAC_0$) et l'autre instance génère un modèle adapté à l'IoT ($HoBAC_4$). Puis, nous avons créé une application Web pour l'administration de politique d'accès basées sur HoBAC. Nous avons ainsi implémnté l'instanciation du modèle HoBAC de base à l'aide de la Policy Machine (PM) afin d'appliquer sa politique d'accès.

Notre modèle proposé est adapté aux systèmes IoT et non-IoT. Dans ce modèle les objets, les sujets et les contextes qui seront créés par la contribution de différents attributs des entités existantes (sujet, objet et contexte) à l'aide des opérations d'agrégations sont avec un haut niveau d'abstraction. Le mécanisme d'abstraction utilisé représente une couche de sécurité supplémentaire car il permet d'empêcher la manipulation directe des objets et des sujets de bas niveau.

5.3 LIMITATIONS ET PERSPECTIVES DE DÉVELOPPEMENT FUTUR

Dans cette section, nous passons en revue des limitations de notre modèle de contrôle d'accès. Ensuite, nous soulignons certaines des avenues intéressantes pour les recherches futures dans la section suivante.

Notre approche proposée permet d'assurer la sécurité des données dans les systèmes IoT et non-IoT. Cependant, il reste encore des pistes de recherche à explorer afin de peaufiner et optimiser notre modèle de contrôle d'accès.

Notre approche doit être testée et évaluée à grande échelle avec des capteurs et des données du monde réel et dans différents scénarios. Il serait aussi judicieux de pouvoir tester notre modèle de contrôle d'accès dans un cadre expérimental pratique afin de pouvoir constater et souligner les limites auxquelles il sera soumis, ainsi que cela permet d'aboutir à des solutions efficaces et qui conviennent aux problèmes rencontrés.

Malgré les réalisations que nous avons réussi à mener, nous pensons que ce travail ouvre la voie vers des nouvelles perspectives et il reste encore du chemin à faire afin

d'améliorer et explorer de nouvelles pistes.

Dans un proche avenir, nous prévoyons de développer et mettre en oeuvre ($HoBAC_4$) sur l'Internet des objets industriels (IIoT) à grande échelle et dans le monde réel.

BIBLIOGRAPHY

[1] Mehdi Adda and Linda Aliane. Hobac: fundamentals, principles, and policies. (Accepted) 2020. The 11th International Conference on Ambient Systems, Networks and Technologies (ANT),The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40).

[2] Mehdi Adda, Jabril Abdelaziz, Hamid Mcheick, and Rabeb Saad. Toward an access control model for iotcollab. In *Proceedings of the 6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015), the 5th International Conference on Sustainable Energy Information Technology (SEIT-2015), London, UK, June 2-5, 2015*, pages 428–435, 2015. doi: 10.1016/j.procs.2015.05.009. URL https://doi.org/10.1016/j.procs.2015.05.009.

[3] Muhammad Umar Aftab, Muhammad Asif Habib, Nasir Mehmood, Mubeen Aslam, and Muhammad Irfan. Attributed role based access control model. *2015 Conference on Information Assurance and Cyber Security (CIACS)*, pages 83–89, 2015.

[4] S. I. Ahamed, F. Rahman, and E. Hoque. Erap: Ecc based rfid authentication protocol. In *2008 12th IEEE International Workshop on Future Trends of Distributed Computing Systems*, pages 219–225, Oct 2008. doi: 10.1109/FTDCS.2008.20.

[5] Linda Aliane and Mehdi Adda. Hobac: toward a higher-order attribute-based access control model. volume 155, pages 303 – 310, 2019. doi: https://doi.org/10.1016/j.procs.2019.08.044. URL http://www.sciencedirect.com/science/article/pii/S1877050919309585. The 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2019),The 14th International Conference on Future Networks and Communications (FNC-2019),The 9th International Conference on Sustainable Energy Information Technology.

[6] A. Alshehri and R. Sandhu. Access control models for virtual object communication in cloud-enabled iot. In *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 16–25, Aug 2017. doi: 10.1109/IRI.2017.60.

[7] Ryan Ausanka-crues. Methods for access control: Advances and limitations. 2001. URL https://pdfs.semanticscholar.org/6192/f0308dc8d7782b55a0557dfb66f323638853.pdf.

[8] E. D. Bell and J. L. La Padula. Secure computer system: Unified exposition and multics interpretation. Bedford, MA, 1976. Mitre Corporation. ISBN ESD-TR-75306. URL http://csrc.nist.gov/publications/history/bell76.pdf.

[9] Elisa Bertino, Piero Andrea Bonatti, and Elena Ferrari. Trbac: A temporal role-based access control model. volume 4, pages 191–233, New York, NY, USA,

August 2001. ACM. doi: 10.1145/501978.501979. URL http://doi.acm.org/10.1145/501978.501979.

[10] Elisa Bertino, Piero Andrea Bonatti, and Elena Ferrari. Trbac: A temporal role-based access control model. *ACM Trans. Inf. Syst. Secur.*, 4(3):191–233, August 2001. ISSN 1094-9224. doi: 10.1145/501978.501979. URL http://doi.acm.org/10.1145/501978.501979.

[11] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy (SP '07)*, pages 321–334. IEEE, May 2007. doi: 10.1109/SP.2007.11. URL http://doi.org/10.1109/SP.2007.11.

[12] Smriti Bhatt, Farhan Patwa, and Ravi Sandhu. Abac with group attributes and attribute hierarchies utilizing the policy machine. In *Proceedings of the 2Nd ACM Workshop on Attribute-Based Access Control*, ABAC '17, pages 17–28, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4910-9. doi: 10.1145/3041048.3041053. URL http://doi.acm.org/10.1145/3041048.3041053.

[13] K. J. Biba. Integrity considerations for secure computer systems. page 66. MITRE Corp., 04 1977.

[14] Prosunjit Biswas, Ravi Sandhu, and Ram Krishnan. Label-based access control: An abac model with enumerated authorization policy. In *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control*, ABAC '16, pages

1–12, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4079-3. doi: 10.1145/2875491.2875498. URL http://doi.acm.org/10.1145/2875491.2875498.

[15] Tuhin Borgohain, Uday Kumar, and Sugata Sanyal. Survey of security and privacy issues of internet of things. volume abs/1501.02211, 2015. URL http://arxiv.org/abs/1501.02211.

[16] B. Chen, Y. Huang, and M. Güneş. S-cbac: A secure access control model supporting group access for internet of things. In *2015 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pages 67–67, Nov 2015. doi: 10.1109/ISSREW.2015.7392046. URL http://doi.org/10.1109/ISSREW.2015.7392046.

[17] A. A. Corici, M. Emmelmann, J. Luo, R. Shrestha, M. Corici, and T. Magedanz. Iot inter-security domain trust transfer and service dispatch solution. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 694–699, Dec 2016. doi: 10.1109/WF-IoT.2016.7845443.

[18] J. P. Cruz, Y. Kaji, and N. Yanai. Rbac-sc: Role-based access control using smart contract. *IEEE Access*, 6:12240–12251, 2018. ISSN 2169-3536. doi: 10.1109/ACCESS.2018.2812844.

[19] David D. Clark and David R. Wilson. A comparison of commercial and military computer security policies. pages 184–195, 04 1987. doi: 10.1109/SP.1987.10001.

[20] Jagan Mohan Reddy Danda and Chittaranjan Hota. Attack identification framework for iot devices. In Suresh Chandra Satapathy, Jyotsna Kumar Mandal, Siba K.

Udgata, and Vikrant Bhateja, editors, *Information Systems Design and Intelligent Applications*, pages 505–513, New Delhi, 2016. Springer India. ISBN 978-81-322-2752-6.

[21] Evans Dave. The internet of things – how the next evolution of the internet is changing everything. April 2011. URL https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf.

[22] Fujun Feng, Chuang Lin, Dongsheng Peng, and Junshan Li. A trust and context based access control model for distributed systems. pages 629–634, 09 2008. doi: 10.1109/HPCC.2008.37.

[23] Tiago Fernández-Caramés, Paula Fraga-Lamas, Manuel Suárez-Albela, and Luis Castedo. Reverse engineering and security evaluation of commercial tags for rfid-based iot applications. volume 17, page 28, 12 2016. doi: 10.3390/s17010028.

[24] David Ferraiolo, Vijayalakshmi Atluri, and Serban Gavrila. The policy machine: A novel architecture and framework for access control policy specification and enforcement. volume 57, pages 412–424, 04 2011. doi: 10.1016/j.sysarc.2010.04.005.

[25] David Ferraiolo, Ramaswamy Chandramouli, D. Kuhn, and Vincent Hu. Extensible access control markup language (xacml) and next generation access control (ngac). pages 13–24, 03 2016. doi: 10.1145/2875491.2875496.

[26] David F. Ferraiolo, Ramaswamy Chandramouli, Rick Kuhn, and Vincent C. Hu. Extensible access control markup language (xacml) and next generation

access control (ngac). In Elisa Bertino, Ravi Sandhu, and Ram Krishnan, editors, *ABAC@CODASPY*, pages 13–24. ACM, 2016. ISBN 978-1-4503-4079-3. URL http://dblp.uni-trier.de/db/conf/codaspy/abac2016.html#FerraioloCKH16.

[27] Sergio Gusmeroli, Salvatore Piccione, and Domenico Rotondi. A capability-based security approach to manage access control in the internet of things. pages 1189–1205, 2013. URL https://doi.org/10.1016/j.mcm.2013.02.006.

[28] M. M. Hossain, M. Fotouhi, and R. Hasan. Towards an analysis of security issues, challenges, and open problems in the internet of things. In *2015 IEEE World Congress on Services*, pages 21–28, June 2015. doi: 10.1109/SERVICES.2015.12.

[29] Kuanmin Hu, GuoYang Cai, and Chengsheng Shen. An enhanced access control model based on trusted computing. In *2nd International Conference on Advances in Mechanical Engineering and Industrial Informatics (AMEII 2016)*. Atlantis Press, 2016/04. ISBN 978-94-6252-188-9. doi: 10.2991/ameii-16.2016.177. URL https://doi.org/10.2991/ameii-16.2016.177.

[30] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas. Attribute-based access control. *Computer*, 48(2):85–88, Feb 2015. ISSN 0018-9162. doi: 10.1109/MC.2015.33.

[31] Vincent Hu, David Ferraiolo, D. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and Karen Scarfone. Guide to attribute based access control (abac) definition and considerations. *National Institute of Standards and Technology Special Publication*, pages 162–800, 01 2014.

[32] Ali Hussein, Mehdi Adda, Mirna Atieh, and Walid Fahs. Smart home design for disabled people based on neural networks. *Procedia Computer Science*, 37:117 – 126, 2014. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2014.08.020. URL http://www.sciencedirect.com/science/article/pii/S1877050914009855. The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2014)/ The 4th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2014)/ Affiliated Workshops.

[33] Xin Jin, Ram Krishnan, and Ravi Sandhu. A unified attribute-based access control model covering dac, mac and rbac. In *Proceedings of the 26th Annual IFIP WG 11.3 Conference on Data and Applications Security and Privacy*, DBSec'12, pages 41–55, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-31539-8. doi: 10.1007/978-3-642-31540-4_4. URL http://dx.doi.org/10.1007/978-3-642-31540-4_4.

[34] A. A. E. Kalam, R. E. Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miege, C. Saurel, and G. Trouessin. Organization based access control. In *Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, pages 120–131, June 2003. doi: 10.1109/POLICY.2003.1206966.

[35] G. Kappes, A. Hatzieleftheriou, and V. S. Anastasiadis. Multitenant access control for cloud-aware distributed filesystems. pages 1–1, 2018. doi: 10.1109/TDSC.2017.2715839. URL http://doi.org/10.1109/TDSC.2017.2715839.

[36] Ke Gao, C. Corbett, and R. Beyah. A passive approach to wireless device fingerprinting. In *2010 IEEE/IFIP International Conference on Dependable Systems Networks (DSN)*, pages 383–392, June 2010. doi: 10.1109/DSN.2010.5544294.

[37] D. R. Kuhn, E. J. Coyne, and T. R. Weil. Adding attributes to role-based access control. *Computer*, 43(6):79–81, June 2010. ISSN 0018-9162. doi: 10.1109/MC.2010.155.

[38] Farah Layouni and Yann Pollet. FI-ORBAC : A Model of Access control for federated identity platform. In *IADIS 2009, the International Conference on Information System*, Barcelona, Spain, February 2009. URL https://hal.archives-ouvertes.fr/hal-01125878. ISBN: 978-972-8924-79-9.

[39] B. Li, D. Huang, Z. Wang, and Y. Zhu. Attribute-based access control for icn naming scheme. volume 15, pages 194–206, March 2018. doi: 10.1109/TDSC.2016.2550437. URL http://doi.org/10.1109/TDSC.2016.2550437.

[40] Yaoping LV, Wei Zhang, Fei YANG, and Tian Zhao. Research of materials unique identification system framework for iot. pages 99–102. IEEE, 2016. ISBN 978-1-4673-8838-2. doi: 10.1109/ICNISC.2016.031.

[41] Juan M. Marín Pérez, Gregorio Martínez Pérez, and Antonio F. Skarmeta Gomez. Secrbac: Secure data in the clouds. volume 10, pages 726–740, Sep. 2017. doi: 10.1109/TSC.2016.2553668. URL http://doi.org/10.1109/TSC.2016.2553668.

[42] Parikshit N Mahalle, Bayu Anggorojati, Neeli R Prasad, and Ramjee Prasad.

Identity authentication and capability based access control (iacac) for the internet of things. volume 1, page 309–348. River Publishers, 2013.

[43] Andrea Margheri, Massimiliano Masi, Rosario Pugliese, and Francesco Tiezzi. A rigorous framework for specification, analysis and enforcement of access control policies. pages 2–33. IEEE, 2017. doi: 10.1109/TSE.2017.2765640.

[44] Yair Meidan, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martín Ochoa, Nils Ole Tippenhauer, and Yuval Elovici. Profiliot: A machine learning approach for iot device identification based on network traffic analysis. In *Proceedings of the Symposium on Applied Computing*, SAC '17, pages 506–509, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4486-9. doi: 10.1145/3019612.3019878. URL http://doi.acm.org/10.1145/3019612.3019878.

[45] Markus Miettinen, Samuel Marchal, Ibaad Hafeez, A Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. Iot sentinel demo: Automated device-type identification for security enforcement in iot. pages 2511–2514. IEEE, 2017. ISBN 978-1-5386-1792-2. doi: 10.1109/ICDCS.2017.284.

[46] Markus Miettinen, Samuel Marchal, Ibaad Hafeez, A Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. Iot sentinel : Automated device-type identification for security enforcement in iot. volume 8, pages 2177–2184, 2017.

[47] B. Mitra, S. Sural, J. Vaidya, and V. Atluri. Migrating from rbac to temporal rbac. *IET Information Security*, 11(5):294–300, 2017. ISSN 1751-8709. doi: 10.1049/iet-ifs.2016.0258.

[48] Bhat Mohd Ibrahim, Ahmad Shariq, Amin Asif, and Ashraf Suhail. E-health with internet of things. volume 6, pages 357–362, June 2017.

[49] M. J. Moyer and M. Abamad. Generalized role-based access control. In *Proceedings 21st International Conference on Distributed Computing Systems*, pages 391–398, April 2001. doi: 10.1109/ICDSC.2001.918969.

[50] P. Murugesan, S. Chinnappa, A. Alaerjan, and D. Kim. Adopting attribute-based access control to data distribution service. In *2017 International Conference on Software Security and Assurance (ICSSA)*, pages 112–115, July 2017. doi: 10.1109/ICSSA.2017.23.

[51] M. Narouei, H. Takabi, and R. Nielsen. Automatic extraction of access control policies from natural language documents. pages 1–1, 2018. doi: 10.1109/TDSC.2018.2818708.

[52] H. Ouechtati, N. Ben Azzouna, and L. Ben Said. Towards a self-adaptive access control middleware for the internet of things. In *2018 International Conference on Information Networking (ICOIN)*, pages 545–550, Jan 2018. doi: 10.1109/ICOIN.2018.8343178.

[53] L. J. Pérez and J. S. Rodriguez. Simulation of scalability in iot applications. In *2018 International Conference on Information Networking (ICOIN)*, pages 577–582, Jan 2018. doi: 10.1109/ICOIN.2018.8343184.

[54] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah. Gtid: A technique for physical

deviceanddevice type fingerprinting. volume 12, pages 519–532, Sep. 2015. doi: 10.1109/TDSC.2014.2369033.

[55] França Rafael. Flexible authentication solution for rails with warden. Technical report, 2019. URL https://github.com/heartcombo/devise. Accessed: 2020-01-17.

[56] Medjdoub Saïda. Modèle de contrôle d'accès pour xml:"application à la protection des données personnelle". December 2005. Thèse de Doctorat, Université de Versailles Saint-Quentin-en-Yvelines.

[57] Mustafa A Salman. On identification of internet of things. pages 59–62, 2014.

[58] Ravi Sandhu, Elisa Bertino, Jaeger Jaeger, Richard Kuhn, and Carl Landwehr. Panel: The next generation of acess control models (panel session): Do we need them and what should they be? In *Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies*, SACMAT '01, pages 53–, New York, NY, USA, 2001. ACM. ISBN 1-58113-350-2. doi: 10.1145/373256.373262. URL http://doi.acm.org/10.1145/373256.373262.

[59] Savio Sciancalepore, Giuseppe Piro, Daniele Caldarola, Gennaro Boggia, and Giuseppe Bianchi. Oauth-iot: An access control framework for the internet of things based on open standards. pages 676–681. IEEE, 2017. ISBN 978-1-5386-1629-1. doi: 10.1109/ISCC.2017.8024606.

[60] Daniel Servos and Sylvia L. Osborn. Hgabac: Towards a formal model of hierarchical attribute-based access control. In Frédéric Cuppens, Joaquin Garcia-Alfaro,

Nur Zincir Heywood, and Philip W. L. Fong, editors, *Foundations and Practice of Security*, pages 187–204, Cham, 2015. Springer International Publishing. ISBN 978-3-319-17040-4.

[61] K. Sha, R. Errabelly, W. Wei, T. A. Yang, and Z. Wang. Edgesec: Design of an edge layer security service to enhance iot security. In *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, pages 81–88, May 2017. doi: 10.1109/ICFEC.2017.7.

[62] Nair Srijith. Xacml reference architecture. Technical report, november 2013. URL `https://www.axiomatics.com/blog/xacml-reference-architecture/`. Accessed: 2019-12-14.

[63] Yunchuan Sun, Lei Wu, Shizhong Wu, Shoupeng Li, Tao Zhang, Li Zhang, Junfeng Xu, and Yongping Xiong. Security and privacy in the internet of vehicles. pages 116–121. IEEE, 2015. ISBN 978-1-4673-8637-1. doi: 10.1109/IIKI.2015.33.

[64] Emmanouil Vasilomanolakis, *Jörg* Daubert, Manisha Luthra, Vangelis Gazis, Alex Wiesmaier, and Panayotis Kikiras. On the security and privacy of internet of things architectures and systems. pages 49–57, 2015. ISBN 978-1-4673-7769-0. doi: 10.1109/SIOT.2015.9.

[65] L. Venkatraman and D. P. Agrawal. A novel authentication scheme for ad hoc networks. In *2000 IEEE Wireless Communications and Networking Conference. Conference Record (Cat. No.00TH8540)*, volume 3, pages 1268–1273 vol.3, Sep. 2000. doi: 10.1109/WCNC.2000.904814.

[66] Zhongyuan Xu and Scott D. Stoller. Mining attribute-based access control policies. volume abs/1306.2401, 2013. URL http://arxiv.org/abs/1306.2401.

[67] M. Yu, D. Zhang, Y. Cheng, and M. Wang. An rfid electronic tag based automatic vehicle identification system for traffic iot applications. In *2011 Chinese Control and Decision Conference (CCDC)*, pages 4192–4197, May 2011. doi: 10.1109/CCDC.2011.5968962.

[68] T. H. Yuen, J. K. Liu, M. H. Au, X. Huang, W. Susilo, and J. Zhou. k-times attribute-based anonymous access control for cloud computing. volume 64, pages 2595–2608, Sep. 2015. doi: 10.1109/TC.2014.2366741. URL http://doi.org/f10.1109/TC.2014.2366741.

[69] Guangsen Zhang and Manish Parashar. Context-aware dynamic access control for pervasive applications, 2004.

[70] Guoping Zhang and Jing Liu. A model of workflow-oriented attributed based access control. volume 3, pages 47–53. International Journal of Computer Network and Information Security(IJCNIS), 2011.

[71] G. Zhao, X. Si, J. Wang, X. Long, and T. Hu. A novel mutual authentication scheme for internet of things. In *Proceedings of 2011 International Conference on Modelling, Identification and Control*, pages 563–566, June 2011. doi: 10.1109/ICMIC.2011.5973767.

[72] Y. Zhu, D. Huang, C. Hu, and X. Wang. From rbac to abac: Constructing flexible

data access control for cloud storage services. volume 8, pages 601–616, July 2015. doi: 10.1109/TSC.2014.2363474.