وزارة التعليم العالي و البحث العلمي

BADJI MOKHTAR UNIVERSITY ANNABA

UNIVERSITE BADJI MOKHTAR ANNABA

جامعة باجي مختار – عنابة

Faculté des Sciences de l'Ingéniorat

Département d'Informatique

THÈSE

Présentée en vue de l'obtention du diplôme de
Doctorat 3ème cycle
Intitulée

La sécurité dans les grilles vs le Cloud Computing

Filière: Informatique

Spécialité: Réseaux et Sécurité Informatique

Par

Namane Sarra

DEVANT Le JURY

Directeur	Madame Ghoualmi Nassira	Pr. Université d'Annaba
Président	Madame Merouani Hayet Farida	Pr. Université d'Annaba
Examinateur	Monsieur Derdour Makhlouf	MCA. Université de Tebessa
Examinateur	Monsieur Abdelkrim Bouramoul	MCA. Université de Constantine 2
Examinateur	Madame Nawel Arrar Remita	MCA. Université d'Annaba

Année 2018

Dédicace

Je dédie ce travail

- A la mémoire de mon père
- A mon adorable maman
- A mon époux
- A ma petite princesse : Nadine
- A ma famille

Remerciement

« Dans la vie, les hommes sont tributaires les uns des autres.
Il y a donc toujours quelqu'un à maudire ou à remercier »
Madeleine Ferron, Extrait de « Le chemin des dames ».

Je remercie d'abord la grâce de dieu, le clément et miséricordieux pour m'avoir guidée, éclairée sur la bonne voie du savoir et de la recherche scientifique afin de pouvoir mener à terme ce travail modeste.

Je tiens en premier à remercier ma directrice de thèse Madame *Nassira Ghoualmi*, professeur à l'université d'Annaba pour son soutien moral, scientifique et administratif qu'elle m'a apportée durant ces quatre années de thèse. Toujours disponible, elle a pris le temps de m'écouter, de répondre à mes nombreuses questions. Elle m'a aussi conseillée, guidée dans mon travail de recherche. Je la remercie également pour tous les moments agréables que nous avons partagés.

J'adresse également mes très sincères remerciements à Madame *Hayet Farida Merouani*, professeur à l'université d'Annaba pour avoir accepté de présider mon jury.

Je remercie chaleureusement Monsieur *Makhlouf Derdour*, maitre de conférences à l'université de Tebessa, Monsieur *Abdelkrim Bouramoul*, maitre de conférences à l'université de Constantine, madame *Nawel Arrar Remita*, maitre de conférences à l'université d'Annaba, de m'avoir fait l'honneur d'accepter d'examiner ma thèse.

Tout cela n'aurait pas été fait sans l'aide de Monsieur *Mustafa Kaiiali*, maitre de conférences à l'université Queen's Belfast, Royaume-Unis. De mon humble point de vue de doctorante, je peux dire aujourd'hui que je lui dois beaucoup, aussi bien pour m'avoir encadrée que pour une chose

Remerciement

essentielle qu'il m'a transmise, l'attitude à tenir face aux obstacles rencontrés tout au long du parcours de recherche.

Je tiens également à exprimer ma gratitude à tous les membres du laboratoire ERIC, plus particulièrement à Madame *Nouria Harbi*, maitre de conférences à l'université Lumière Lyon2, qui m'a donnée goût à la recherche lors de mon stage de courte durée. Qu'elle reçoive toute ma considération.

Je remercie également mon amie *Marwa Ahmim* pour tous ses conseils et sa disponibilité.

Je tiens à remercier également Monsieur *Raouf Toumi* pour son suivi et ses conseils sans oublier les aides de Monsieur *Djamel Znakhra*.

Je remercie mon adorable *Maman*, tante *Nina*, mon amie *Roumaissa*, tante *Jahida* et Dr *Abdelaziz Khati* pour la lecture de ce manuscrit.

Je remercie ma famille, à commencer par *mon mari* qui m'a toujours soutenue et supportée durant ces années de recherche, sans oublier le soutien moral et physique de *ma mère*, mes *beaux parents* et *Besma* que je remercie vivement. Je remercie également ma petite princesse *Nadine* qui est venue au monde à temps pour m'encourager et illuminer ma vie.

Enfin, mes remerciements vont à mes amis et toute personne ayant participé de près ou de loin à la réalisation de ce travail.

الملخص

يتناول موضوع اطروحتنا مشكلة الامن التي تعد واحدة من المشاكل العويصة في المحيطات الدينامكية و الموزعة مثل : الحوسبة الشبكية و الحوسبة السّحابية . ركزنا اوّلا على مسألة الأمن الخاص بهما بشكل عام حيث درسنا جميع الأعمال المقترحة في الادبيات. سمحت لنا هذه الدراسة بأن نعرف ان اشكالية التحكم في الدخول تمثل المشكلة الأكثر استهدافا من الأبحات الحالية لأن أهميتها ملحوظة مقارنة مع باقي الإشكاليات و ذلك في كل من محيط الحوسبة الشبكية و الحوسبة السّحابية. هذا ما ادى بنا الى تركيز كل اعمالنا حول هذه المشكلة. في المساهمة الاولى، قدّمنا نموذجا متوازي للتحكم في الدخول الى محيط الحوسبة الشبكية مكون من عدّة مجالات، سمحت هذه المساهمة بتسهيل التعاون بين مختلف المجالات الإدارية علما انّ كل مجال لديه سياسته الأمنية الخاصة. في المساهمة الثانية، قمنا بإقتراح بيان موزون للسّماح بالدخول الى الحوسبة الشبكية (WGAG) الذي يمثل آلية لتخزين و ادارة سياسات الأمن . في المساهمة الثالثة قمنا بتقديم بيان موزون للسماح بالدخول الى الحوسبة الشبكية على أساس الإنجازات (Action-WGAG) تعتبر هذه الآلية اكثر تعبيرا مقارنة مع الآليات السابقة لأنّها قدّمت فكرة العمل المنجز التي كانت غائبة في الآليات الأخرى. بالإضافة الى ذلك كانت هذه المساهمة مرافقة بمحلل سياسات الأمن الذي يقوم بتصفح ملفات الأمن و تحويلها الى جدول الأمن الممثل لها .في النهاية، في مساهمتنا ألأخيرة أخذنا بعين الاعتبار عملية التحكم في الدّخول الى الحوسبة السحابية، اقترحنا نموذجا للتّحكم في الدّخول الى مستويات مختلفة من المعلومات المستضافة في محيط السحاب، لقد ضمّ هذا الاقتراح كيفية تخزين سياسات الامن اضافة الى عملية التحكم في الدّخول نفسها.

كلمات البحث: الحوسبة الشبكية، الحوسبة السحابية، الامن، التحكم في الدخول، سياسات الامن، المصادقة، السلامة، الخصوصية.

Abstract

The subject of our thesis deals with the security issue, which is widely considered in dynamic and distributed environments such as grids and Cloud Computing. We first focused generally on the security issue in grids as in Cloud Computing. We found that access control is the security issue most targeted by current research. That's what pushed us to focus our work on this issue. First, we presented a parallel access control model in a cross-domain grid computing environment. Hence, this contribution allowed the collaboration between several administrative domains knowing that each domain has its own security policy. In the following contribution, we proposed a Weighted Grid Authorization Graph (WGAG) that represents a mechanism for storing and managing security policies. In the third contribution, we presented an Action Weighted Grid Authorization Graph (Action-WGAG). In fact, this mechanism is more expressive because it introduced the notion of action that was absent in the other mechanisms. In addition, it was accompanied by a Security Policy Parser (SP-Parser) which browses the security policy files and returns as a result a security table that represents them. Finally, in our last contribution, we targeted the process of access control in Cloud Computing. Thus, we proposed an access control model to different levels of data hosted in a Cloud environment. This model has managed the storage of security rules as well as the access control process itself.

Keywords: Grid Computing, Cloud Computing, Security, Access Control, Security Policies, Authentication, Integrity, Confidentiality.

Résumé

Le sujet de notre thèse traite l'issue de sécurité, qui est largement considérée dans des environnements dynamiques et distribués tels que les grilles et le Cloud Computing. Nous avons commencé par faire un état de l'art détaillé et général sur la sécurité de ces deux technologies. Nous avons trouvé que le contrôle d'accès représente l'issue de sécurité la plus visée par les recherches actuelles. C'est ce qui nous a poussés à concentrer notre travail sur cette issue. D'abord, nous avons présenté un modèle de contrôle d'accès parallèle dans un environnement de grille inter-domaines. Cette contribution a pu garantir la collaboration entre plusieurs domaines administratifs sachant que chaque domaine a sa propre politique de sécurité. Dans la contribution suivante, nous avons proposé un graphe pondéré d'autorisation dans les grilles (WGAG) qui représente un mécanisme de stockage et de gestion des politiques de sécurité. Dans notre troisieme contribution, nous avons présenté un graphe pondéré basé sur l'action pour l'autorisation dans les grilles (Action-WGAG). Ce mécanisme est plus expressif car il a introduit la notion d'action qui était absente dans les autres mécanismes. En outre, il a été accompagné par un parseur de politiques de sécurité (SP-Parser) qui parcourt les fichiers de ces dernières et donne comme résultat une table de sécurité qui les représente. Enfin, dans notre dernière contribution, nous nous sommes intéressés au processus de contrôle d'accès dans le Cloud Computing. Nous avons proposé un modèle de contrôle d'accès à différents niveaux de données hébergées dans un environnement de Cloud. Ce modèle a géré le stockage des règles de sécurité ainsi que le processus de contrôle d'accès lui-même.

Mots clés : Grilles, Cloud Computing, Sécurité, Contrôle d'accès, Politiques de sécurité**,** Authentification, Intégrité, Confidentialité**.**

Table des matières

Table des matières

Table des matières

Table des matières

Table des matières

Table des matières

Liste des Figures

Liste des Figures

Liste des tableaux

Introduction générale

Introduction générale

1. Le cadre scientifique

Ces dernières années, les besoins des utilisateurs en termes de puissance de calcul et de capacité de stockage ont changé, c'est ce qui a permis à l'informatique distribuée de progresser considérablement. En outre, la popularité d'internet ainsi que la disponibilité d'ordinateurs puissants et des technologies de réseau à haute vitesse, ont changé la façon dont les ordinateurs sont utilisés. Les grilles de calcul ont vu le jour au milieu des années 1990 [Sadashiv N. and Dilip Kumar S.M., 2011], similaire à internet, les technologies de grilles initiales ont été développées principalement dans les universités et les laboratoires de recherche pour résoudre des problèmes de recherche uniques et pour permettre la collaboration entre différents chercheurs à travers le monde. Le but principal de cette technologie était de permettre aux utilisateurs de profiter à distance d'une puissance de calcul inutilisée dans d'autres centres de calcul lorsque le centre local est occupé. Initialement, cette technologie ne faisait référence qu'à une seule grille et avait une audience plutôt limitée. Cependant, après des années de développement, la grille a pris de l'ampleur et est devenue un moyen efficace pour le partage coordonné des ressources et la résolution de problèmes dans des organisations virtuelles dynamiques et multi-institutionnelles. D'autre part, le Cloud Computing est une sorte de modèle informatique qui a vu le jour à la fin de l'année 2007 [Sadashiv N. and Dilip Kumar S.M., 2011]. Ce modèle permet d'accéder à un réseau partagé de ressources informatiques configurables (réseaux, serveurs, stockages, applications et services), qui peuvent être rapidement provisionnées et libérées avec un minimum d'effort de gestion [Mell P. and Grance T., 2011]. Le partage des ressources dans les grilles se fait entre plusieurs domaines administratifs où chacun a sa propre politique de sécurité. D'un autre coté, le Cloud Computing est un paradigme qui permet de fournir à l'utilisateur tout ce dont il a besoin sous la forme d'un service. Ce dernier peut être provisionné par un ou plusieurs fournisseurs ayant chacun sa propre politique de sécurité. Enfin, l'émergence des grilles et du Cloud Computing a introduit de nouveaux concepts liés à leur sécurité. Cette dernière est une question cruciale car en plus de leur nature dynamique, ce n'est pas uniquement les données qui doivent être sécurisées mais également les ressources, les calculs et les services doivent l'être pour éviter les accès inadéquats d'où la nécessité d'un mécanisme de contrôle d'accès rapide et efficace.

2. La problématique

Dans les grilles comme dans le Cloud, contrôler l'accès aux ressources ou aux services est une tâche difficile car ces deux environnements sont caractérisés par le grand nombre d'utilisateurs qu'ils doivent gérer ce qui donne un grand nombre de requêtes d'accès à traiter. Par conséquent, la problématique principale consiste à essayer d'assurer un contrôle d'accès efficace et rapide dans ces deux environnements de nature dynamique. Garantir cette efficacité peut être réalisée en améliorant l'une des deux phases du processus de contrôle d'accès. La première qui prend en charge le mécanisme de stockage et de gestion des politiques de sécurité. La seconde gère la manière de répondre aux requêtes d'accès. Si les politiques de sécurité soumises par l'administrateur sont stockées, représentées et gérées d'une manière efficace, cela pourra améliorer le processus entier. Plusieurs mécanismes de représentation et de gestion de politiques de sécurité ont été proposés dans un environnement de grille. La première sous-problématique qui doit être résolue consiste à vérifier si les améliorations apportées à ces mécanismes sont optimales, c'est-à-dire aucune autre amélioration ne pourra être rajoutée pour avoir de meilleurs résultats. La seconde sous-problématique, consiste à vérifier si ces mécanismes peuvent être utilisés pour représenter des politiques de sécurité de systèmes réels, c'est-à-dire est ce que ces mécanismes tiennent compte de tous les éléments nécessaires pour représenter une politique de sécurité ? D'un autre côté, les grilles sont connues par leur architecture multi-domaines. L'une des sous-problématiques qui devra être traitée est : comment assurer un contrôle d'accès gérant la collaboration entre ces domaines sachant que chaque domaine a ses propres politiques de sécurité et les caractéristiques d'un utilisateur diffèrent d'un domaine à un autre ? Enfin, le Cloud Computing est un environnement qui est orienté entreprise, c'est une technologie qui permet d'avoir accès à tout ce dont on a besoin sous la forme d'un service. L'accès à ces services est également géré par des politiques de sécurité. Aucun mécanisme de représentation et de gestion de politiques n'a été proposé dans un environnement de Cloud. L'une des sous-problématiques qui reste à prendre en considération, consiste à trouver un mécanisme efficace pour la représentation et la gestion des politiques de sécurité. Ce mécanisme doit améliorer le processus de contrôle d'accès en entier.

3. Le but du travail et les contributions réalisées

Le but principal de notre thèse consiste à étudier la sécurité des grilles et celle du Cloud Computing. Nous avons commencé par analyser les différents travaux qui ont été proposés

pour résoudre cette problématique dans les deux domaines, nous avons trouvé que l'issue du contrôle d'accès est la plus visée par les chercheurs, c'est pourquoi nous avons concentré notre travail sur le processus du contrôle d'accès. Si on arrive à proposer un modèle de contrôle d'accès efficace et rapide pour chacun des environnements, nous pouvons dire que nous avons pu assurer la sécurité. L'efficacité du processus de contrôle d'accès peut être garantie de deux manières. La première essaye d'assurer un processus de réponse à une requête d'accès rapide et efficace en supposant que les politiques de sécurité sont sous la forme avec laquelle elles ont été soumises par l'administrateur. En se basant sur cette issue, nous avons proposé notre première contribution qui permet d'assurer un contrôle d'accès parallèle dans un environnement de grille inter-domaines. Cette contribution permet la collaboration entre plusieurs domaines administratifs sachant que chacun a sa propre politique de sécurité. En ce qui concerne la deuxième façon, elle consiste à trouver le mécanisme le plus efficace pour représenter les politiques de sécurité. Cette représentation doit permettre la vérification d'une politique de sécurité en prenant un temps minimum et doit donner un résultat juste. Pour répondre à ces besoins, nous avons commencé par analyser les différents mécanismes existants pour la représentation des politiques de sécurité, nous avons trouvé que cette issue a été prise en compte uniquement dans les grilles où le meilleur mécanisme proposé était le graphe d'autorisation dans les grilles (GAG). Ce dernier a permis d'éliminer la redondance dans la vérification des politiques de sécurité, mais il y a certaines règles de sécurité qui peuvent ne pas être vérifiées dès le début et GAG les vérifie quand même. C'est ce qui nous a poussés à proposer notre deuxième contribution qui représente un graphe pondéré d'autorisation dans les grilles (WGAG). Ensuite, nous avons pu constater que le WGAG ne peut pas être appliqué à toutes les politiques de sécurité d'un système réel car il ne contient pas tous les éléments nécessaires. Ceci, nous a permis de proposer notre troisième contribution qui présente un graphe pondéré basé sur l'action pour l'autorisation dans les grilles (Action-WGAG). Enfin, dans notre dernière contribution, nous nous sommes intéressés au processus du contrôle d'accès dans le Cloud. Nous avons proposé un modèle de contrôle d'accès à différents niveaux de données hébergées dans un environnement de Cloud, ce modèle a géré le stockage des règles de sécurité ainsi que le processus du contrôle d'accès lui-même.

4. La structure de la thèse

Notre thèse est composée de cinq chapitres. Dans le premier chapitre, nous introduisons toutes les notions relatives aux deux technologies : grilles et Cloud Computing. Nous commençons par leurs définitions, leurs architectures, leurs avantages et inconvénients ainsi que toutes informations nécessaires pour comprendre au mieux le fonctionnement de ces environnements. Dans le deuxième chapitre, nous présenterons un état de l'art détaillé sur la sécurité des grilles. Nous avons attribué les travaux qui existent à différentes classes selon l'issue de sécurité traitée, à savoir: l'authentification, le contrôle d'accès, l'intégrité, la confidentialité et issues multiples. Dans le troisième chapitre, un état de l'art détaillé sur la sécurité du Cloud Computing a été présenté. Nous avons suivi la même classification utilisée dans le deuxième chapitre. Dans le quatrième chapitre, une étude comparative entre les travaux présentés dans chaque classe de chaque environnement a été faite suivie par une comparaison entre les différentes techniques proposées pour résoudre une issue particulière dans les grilles et celles présentées pour résoudre cette même issue dans le Cloud. Dans le dernier chapitre, nous avons présenté nos quatre contributions. Les trois premières concernaient le contrôle d'accès dans les grilles. Enfin, notre dernière contribution propose un modèle de contrôle d'accès dans un environnement de Cloud.

Chapitre I : Les grilles versus le Cloud Computing

1.1. Introduction

Dans ce premier chapitre, nous introduisons tout ce qui concerne les grilles informatiques et le Cloud Computing. Dans la première partie, nous commencerons par les grilles, leur définition leur histoire, leur architecture, leurs types et topologies, leurs domaines d'application, leurs avantages et inconvénients. Ensuite, nous analyserons les besoins de sécurité liés à cette technologie. Dans la deuxième partie, nous prendrons en considération le Cloud Computing. Nous commencerons par présenter les différentes définitions données pour cette technologie. Ensuite, nous résumerons l'histoire du Cloud, en tenant compte des événements remarquables qui lui ont apporté des changements. Puis, nous passerons aux avantages et inconvénients du Cloud, les différentes architectures proposées ainsi que les modèles de services et de déploiement. Ensuite, nous ferons une comparaison avec les grilles, et pour finir, nous analyserons les différents besoins de sécurité qui ont un impact sur l'utilisation du Cloud et son émergence.

1.2. Les grilles informatiques

1.2.1. Définition des grilles

Plusieurs définitions concernant les grilles ont été données, En 2002, lors de la conférence planet du Grid Computing en Californie, le professeur Buyya a défini la grille comme un type de système parallèle et distribué permettant : le partage, la sélection et l'agrégation de ressources autonomes géographiquement distribuées. Ces opérations sont faites de manière dynamique au moment de l'exécution en fonction de la disponibilité des ressources, leurs capacités, leurs performances ainsi que du coût et de la qualité de service exigée par les utilisateurs [Buyya R. et Al., 2011]. Daniel Minoli a défini les grilles comme un environnement informatique réparti et virtuel. Un tel environnement vise à permettre la sélection, le partage et l'agrégation dynamiques des ressources autonomes (géographiquement) distribuées en fonction de la disponibilité, de la capacité, de la performance et du coût de ces ressources informatiques et simultanément en fonction des spécificités d'une organisation [Minoli D., 2005]. Mais, Ian Foster qui est l'un des partisans de la technologie des grilles les a définies comme un système qui coordonne les ressources qui ne sont pas soumises à un contrôle centralisé (distribuées), en utilisant des protocoles et des interfaces standards, ouverts et polyvalents afin de fournir des qualités de service non

triviales. Ian Foster n'a pas uniquement défini les grilles mais il a également expliqué les différents points de sa définition comme suit [Foster I. and Kesselman C., 2002] :

- **Coordonne des ressources distribuées** : les ressources et les utilisateurs appartiennent à différents domaines administratifs, par exemple un utilisateur dans une entreprise alpha appartenant au service achat veut utiliser une ressource appartenant au service comptabilité.
- **Utilisation de protocoles et d'interfaces standards, ouverts et à usage général** : une grille est basée sur des protocoles et interfaces standards qui gèrent des problématiques diverses telles que : l'authentification, le contrôle d'accès, la découverte de ressource, …
- **Pour fournir des qualités de service non triviales** : une grille permet d'utiliser ses ressources de manière coordonnée afin de pouvoir offrir différentes qualités de service.

1.2.2. Histoire et origine des grilles

La grille informatique n'est pas une nouvelle technologie qui a été développée du néant, mais il s'agit plutôt de l'assemblage de différentes technologies existantes telles que : l'informatique en grappes, les technologies pair-à-pair (P2P) et les services web [Chakrabarti A., 2007]. Le terme « Grid », a été inventé dans les années 1990 pour représenter une infrastructure contenant des ressources partagées, hétérogènes, distribuées, externalisées et coordonnées afin de répondre aux besoins de calcul intensif. L'idée de création de grille est née dans le but d'utiliser les cycles de processeurs inutilisés dans des calculs complexes nécessitant un large nombre de ressources. Le concept de la grille est analogue à un réseau d'énergie électrique (réseau électrique) dans lequel les groupes électrogènes sont distribués mais les utilisateurs peuvent accéder à l'énergie électrique sans se préoccuper de la source d'énergie et de sa gestion opérationnelle. En ce qui concerne l'histoire, les grilles sont passées par plusieurs étapes, leur évolution était liée à certains mécanismes qui ont permis de donner plusieurs versions (la figure 1.1). En 1970, le docteur Richard Crandall a été le premier à avoir exprimé le terme « grille de calcul » grâce à son programme parallèle distribué « Zilla ». Ce dernier représente un ensemble de machines chainées entre elles afin de réaliser des calculs mathématiques complexes [Crandall R., 2018]. Dans les années 1987, Smarr L. et Catlett C.E. ont inventé le terme méta-calcul (metacomputing) pour décrire le concept d'un

environnement de calcul cohérent et connecté [IRMA, 2018]. En 1992, Smarr L. et Catett C.E. ont proposé un méta-système défini comme un réseau d'observation qui incrémentera les ressources intelligentes et de calcul applicables à une fonction [Smarr L. and Catllet C.E., 1992]. En 1993, Andrew et Al. ont proposé un projet pour la réalisation d'un ordinateur virtuel national nommé « Legion » [Globus, 2018]. A la fin de l'année 1994, le projet I-WAY (Information Wide Area Year) a été créé par Rick Stevens, qui est directeur de la division mathématique et informatique au laboratoire national d'Argonne et Tom DeFanti, qui est directeur du laboratoire de virtualisation électronique à l'université d'Illinois à Chicago [Dugénie P., 2006]. Après la réalisation de la grille I-WAY, une petite équipe dirigée par Ian Foster a proposé de nouveaux protocoles qui ont permis aux utilisateurs d'I-WAY d'exécuter des applications sur des ordinateurs à travers le pays. Cette expérience a permis de pousser la DARPA (Defense Advanced Research Projects Agency) à financer le projet. Ce qui a permis à la première version de la boite à outils Globus (GT1) d'apparaitre en 1997 et d'être déployée sur plus de 80 sites dans le monde entier [Globus, 2016]. En 1997 également, le projet UNICORE a été réalisé en Allemagne afin de permettre l'accès aux ressources de calcul à haute performance (HPC) de différentes villes allemandes par les utilisateurs de manière transparente, sécurisée et intuitive sur internet [Erwin D.W. and Snelling D.F, 2001]. En 2002, la norme OGSA (Open Grid Service Architecture) qui est un modèle d'architecture orienté service a été présentée lors d'une conférence à Toronto par le GGF (Grid Global Forum). Cette norme a pour but de virtualiser les ressources et de les restituer sous forme de services pour pouvoir les réunir et les diviser selon le besoin [OASIS, 2016][b]. OGSA représente une évolution naturelle de la boite à outils Globus 2 (GT2) car cette norme réévalue et résume davantage les éléments clés déjà existants dans la boite à outils Globus pour pouvoir les appliquer à n'importe quel niveau afin de virtualiser les ressources de la VO (Virtual Organization) [Foster I. et Al., 2002]. A la fin de l'année 2002, une implémentation open source et soi-disant complète d'OGSA a été réalisée pour donner la boite à outils Globus 3 (GT3) [Foster I. et Al., 2002]. En 2002 également, des membres d'OASIS (Organization for the Advancement of Structured Information Standards) [OASIS, 2017] ont proposé SAML (Security Assertion Markup Language) version 1[OASIS, 2016][c] .Ce dernier est un standard informatique basé sur XML définissant un protocole permettant l'échange des informations de sécurité [OASIS, 2016][d]. En 2003, des membres d'OASIS ont également présenté XACML (eXtensible Access Control Markup Language) version 1[OASIS, 2017][b] qui représente un langage de rédaction de politique de sécurité ainsi qu'un cadre de contrôle d'accès utilisant

des messages de demande et de réponse [OASIS, 2016][e]. En 2003 également, le GGF (Global Grid Forum) a proposé l'infrastructure de services de grille ouverte OGSI (Open Grid Service Infrastructure) qui permet de fournir une couche infrastructure à OGSA. Le but principal de cette couche était d'étendre les services web pour s'adapter aux ressources de la grille [Tuecke et Al., 2003]. Plusieurs versions de la boite à outils Globus version 3 ont été présentées jusqu'à la proposition de la dernière version (GT 3.2) qui prend en charge les corrections de bugs importants ainsi que des améliorations sur les fonctionnalités existantes (GridFtp, CAS, GSI, GRAM, …) [DTIC, 2017]. En 2004, des membres d'OASIS ont annoncé leur planification pour définir WSRF (Web Service Resource Framework) qui représente un ensemble de spécifications interopérables et modulaires permettant la modélisation et l'accès aux ressources dynamiques en utilisant des services web [Globus, 2017]. En avril 2005, la Globus alliance [Globus, 2016][b] a développé la boite à outils Globus 4 (GT4) qui représente l'implémentation des spécifications WSRF. En outre, les langages SAML et XACML ont été utilisés dans GT4 pour assurer le processus d'autorisation [Foster I., 2005]. En 2010, l'un des membres de l'équipe de développement de la boite à outils Globus a annoncé la disponibilité de Globus 5 (GT5). Dans cette version, des mises à jour (correction de bugs et nouvelles fonctionnalités) ont été faites sur les composants existants dans GT4 (GridFtp, RLS, MyProxy, GSI-Open SSH), mais certains composants de GT4 ont été remplacés par de nouveaux logiciels dans GT5 [Gtnews, 2017]. Dans la période entre juillet 2010 et février 2013, plusieurs versions de GT5 ont été proposées jusqu'à la dernière version GT5 2.4 [Gtnews, 2017]. En novembre 2014, la boite à outils Globus 6 a été réalisée. Son objectif principal était de réduire la complexité du développement ainsi que la complexité du test [Gtnews, 2017].

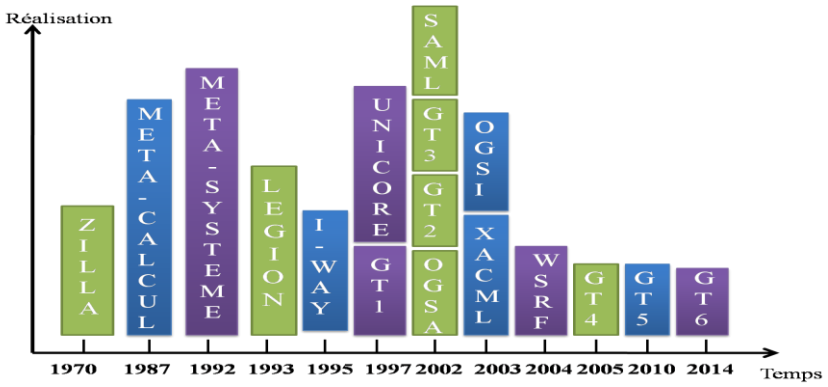

Figure 1. 1 L'histoire des grilles

1.2.3. Domaines d'utilisation des grilles

1.2.3.1. Les sciences de la vie

Les grilles présentent un grand potentiel pour devenir une infrastructure standard dans le domaine des sciences de la vie qui nécessitent souvent un calcul de haute performance et un traitement de données important dépassant la capacité de calcul d'une seule institution [Akihiko K., 2006]. L'un des meilleurs exemples de calcul de haute performance dans la branche des sciences de la vie est le sous-projet WISDOM (Wide In Silico Docking On Malaria) du projet EGEE (Enabling Grids for E-sciencE) [Wang L. et Al., 2009]. Il a réalisé plus de 46 millions de simulations d'amarrage en utilisant 1700 ordinateurs distribués dans 15 pays en environ 6 semaines.

1.2.3.2. L'analyse financière

Le besoin de puissance de calcul dans le secteur des services financiers a longtemps été un problème critique. La concurrence accrue, l'environnement réglementaire changeant et les pressions économiques ont amené de nombreuses institutions financières à constater que leurs ressources informatiques existantes ne suffisent plus à répondre à la demande [Brighthub, 2016]. Ces exigences comprennent la réalisation de tâches complexes telles que l'analyse de marché, la tarification et la gestion des risques, qui sont essentielles pour que les institutions financières dépassent leurs concurrents et prennent les meilleures décisions possibles dans les plus brefs délais. Enfin, les grilles fournissent des services de systèmes avancés offrant toutes les solutions compétitives afin d'atteindre le caractère unique que requiert le secteur de l'analyse financière [Joshy J. and Craig F., 2003].

1.2.3.3. La collaboration scientifique

La recherche scientifique dans différents domaines génère d'énormes quantités de données qui nécessitent une grande puissance d'analyse et de calcul ne pouvant pas être procurée par une seule organisation. Les grilles fournissent le principe d'organisation virtuelle qui permet à différentes universités (organisations) de s'engager dans des activités de collaboration scientifique. Les chercheurs appartenant à ces organisations, doivent partager des données, des processeurs et des instruments matériels tels que les télescopes et des équipements de test avancé. La plupart de ces ressources sont dispersées sur une vaste zone géographique [Julius M. and Elyjoy M., 2017].

1.2.3.4. L'ingénierie et le design

La pression concurrentielle dans le secteur des affaires et de l'industrie, pousse les ingénieurs et les concepteurs à essayer d'utiliser des mécanismes de saisie et d'analyse de données plus complexes afin de donner des réponses plus rapides aux exigences du marché. Les grilles offrent un large éventail de fonctionnalités qui répondent aux types nécessaires d'analyse et de modélisation tels que : l'analyse des données en temps réel, les études paramétriques pour la vérification des différents aspects des systèmes, les expériences de modélisation pour la création de nouveaux designs et les activités de simulation pour la vérification de l'exactitude et la précision des modèles existants [Joshy J. and Craig F., 2003].

1.2.3.5. Les jeux collaboratifs

La planète terre comporte environ 430 millions de joueurs de différents types, il y'a ceux qui préfèrent les jeux d'arcades, d'autres qui préfèrent les consoles et enfin il existe les joueurs de poches assis dans l'avion qui tentent désespérément d'atteindre le niveau suivant avant que le capitaine demande que tous les appareils électroniques soient éteints pour l'atterrissage [Foster I. and Kesselman C., 2002]. Un jour, tous ces joueurs voudront se retrouver en ligne pour faire preuve d'intelligence et de réflexe dans des milliers de sessions simultanées. Servir des millions de joueurs dans des tournois demandant une haute performance et une faible latence nécessite une infrastructure distribuée qui peut être largement déployée en tirant parti du matériel de base en utilisant autant de composants libres que possible [Foster I. and Kesselman C., 2002]. Les grilles sont capables de prendre en charge de tels environnements virtuels pour permettre des jeux collaboratifs [Joshy J. and Craig F., 2003]. En mai 2002, la grille « Butterfly » qui est basée sur la boite à outils Globus, a été introduite à l'industrie du jeu vidéo dans l'Electronic Entertainment Expo à Los Angeles [Foster I. and Kesselman C., 2002].

1.2.3.6. Le gouvernement

Les grilles de gouvernement se concentrent sur la fourniture d'un accès coordonné à des quantités massives de données détenues par les différentes agences d'un gouvernement. Cela permet un accès plus rapide pour résoudre les problèmes critiques, tels que les situations d'urgence, et d'autres activités normales. Ces environnements clés permettent une prise de décision plus efficace avec un temps d'exécution minimal. Les grilles de gouvernement permettent également la création des organisations virtuelles qui incluent plusieurs participants appartenant à plusieurs agences gouvernementales, c'est ce qui offre la possibilité

d'analyser des données en temps réel et de fournir des solutions rapides à des problèmes spécifiques [Joshy J. and Craig F., 2003].

1.2.4. Avantages et inconvénients des grilles

Ces dernières années, les besoins des organisations ont changés car elles se retrouvent dans l'obligation de traiter, stocker et analyser d'énormes quantités de données. Les grilles permettent de répondre à ces besoins mais comme tout système, elles ont des avantages et des inconvénients. Tout d'abord, l'utilisation des grilles au sein d'une organisation permet d'augmenter la capacité de calcul et cela est dû au grand nombre de ressources partagées. De cette manière l'organisation n'a plus besoin de faire des efforts énormes afin de pouvoir se procurer une telle puissance. D'autre part, le matériel informatique (processeur, espace de stockage, …) sera utilisé d'une manière plus efficace (profiter du décalage horaire pour utiliser les ressources pendant leur temps libre). Les grilles permettent aussi, l'utilisation de plusieurs ressources appartenant à plusieurs institutions pour résoudre un problème unique, ce qui donne la rapidité du traitement. En outre, l'appartenance des ressources à plusieurs institutions donne le principe d'externalisation de calcul ce qui permet aux organisations de profiter d'une puissance de calcul exceptionnelle tout en évitant les contraintes d'endroit, d'entretien du matériel (climatisation, gestion d'énergie, …) dans un centre de recherche. Enfin, les grilles fournissent également la flexibilité nécessaire pour répondre à des demandes d'urgences imprévues en louant des ressources externes pour des périodes requises au lieu de les posséder.

Evidemment, il ne peut pas y avoir que des avantages, il existe aussi des inconvénients qui peuvent empêcher le choix d'utilisation des grilles. Tout d'abord, l'un des plus importants inconvénients de la grille concerne les processus et leurs résultats. Plus précisément, les résultats de tous les processus sont envoyés en premier lieu à tous les nœuds de la grille, puis évalués en collaboration. Donc, avant l'évaluation finale il est impossible de définir un résultat final [IGI Global, 2018]. C'est un problème lorsqu'il s'agit de projets sensibles au temps. En outre, le second inconvénient classé comme important est que les grilles reposent en premier lieu sur le principe de gestion des données dispersées pour pouvoir améliorer les performances, en second lieu elles reposent sur la connectivité qui peut engendrer des erreurs de manière inattendue . Enfin, les grilles demandent une infrastructure avancée (des serveurs, des connexions rapides, …) afin de maximiser le potentiel de cette infrastructure. Cela

demande l'utilisation d'un ensemble d'outils de qualité, des logiciels ainsi que des techniciens pour gérer la grille. En d'autres termes, cette technologie est coûteuse [Chakrabarti A., 2007].

1.2.5. Architecture d'une grille

Dans [Foster I. and Kesselman C., 2002], les auteurs ont présenté une architecture de grille qui identifie les composants fondamentaux du système. Ils ont spécifié le but et la fonction de ces composants en indiquant comment ces derniers interagissent les uns avec les autres. Leur but n'était pas de fournir une énumération complète de tous les composants requis mais d'identifier les exigences pour les classes générales de composants. Le résultat est une structure architecturale extensible et ouverte (Figure 1.2) au sein de laquelle peuvent être placées des solutions aux exigences clés. L'architecture de la grille proposée est basée sur les principes du modèle du sablier, le col étroit du sablier définit un petit ensemble d'abstractions de protocoles centraux (ex : TCP et http) sur lesquels de nombreux comportements de hauts niveaux différents peuvent être posés (le sommet du sablier) et qui peuvent être eux même posés sur de nombreuses différentes technologies sous-jacentes (la base du sablier) [Foster I. and Kesselman C., 2002]. Dans l'architecture proposée, le col du sablier est constitué des protocoles de ressources et de connectivité qui facilitent le partage de ressources individuelles [Foster I. and Kesselman C., 2002]. Les protocoles de ces couches sont conçus pour être implémentés au dessus d'un large éventail de types de ressources définis dans la couche fabrique et peuvent à leur tour être utilisées pour construire une large gamme de services globaux appartenant à la couche collective nommée ainsi car elle implique l'utilisation coordonnée (collective) de plusieurs ressources [Foster I. and Kesselman C., 2002]. Dans ce qui suit nous allons citer les différentes couches et expliquer leur fonctionnement.

1.2.5.1. La couche fabrique

Cette couche représente la plus basse couche du modèle, elle fournit les ressources qui seront partagées via la grille .Ces ressources peuvent être d'origine physique (des processeurs, des disques, des capteurs, des réseaux, …) ou logiques (un système de fichier distribué, une ferme de calcul, une base de données, …) [Foster I. and Kesselman C., 2002]. La couche fabrique fournit un mécanisme de gestion et de contrôle de ressource, c'est-à-dire lorsqu'une ressource est demandée par une couche supérieure, les composants de la couche fabrique se chargent d'implémenter les opérations spécifiques au partage de cette ressource (caractéristique du matériel, pourcentage d'occupation).

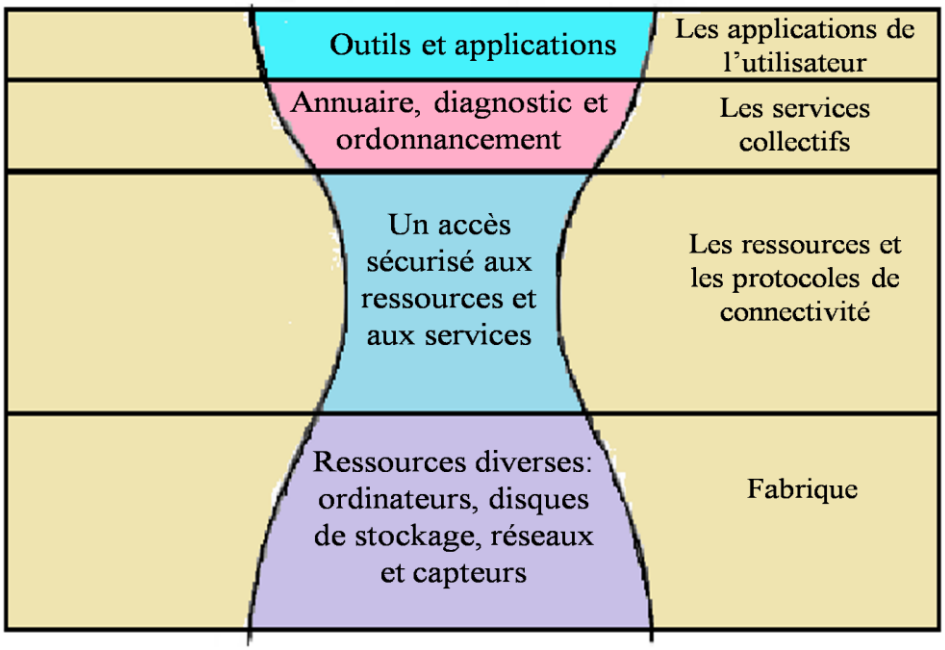

Figure 1. 2 L'architecture des grilles [Foster I. and Kesselman C., 2002]

1.2.5.2. La couche connectivité

Cette couche implémente les principaux protocoles de communication incluant le transport, le routage et le nommage en réutilisant principalement ceux d'internet (IP, TCP, DNS), ceci permet le transfert de données à travers les ressources de la couche fabrique. La couche connectivité se charge également des services de sécurité tels que : la délégation, l'authentification unique (SSO : Single Sign On) … [Foster I. and Kesselman C., 2002]

1.2.5.3. La couche ressource

Cette couche concerne les ressources que d'un point de vue individuel. Grace aux deux couches précédentes (fabrique et connectivité), elle collecte les informations concernant les ressources (structure et état, configuration, politique de sécurité et charge) en utilisant des protocoles d'information. D'autre part, elle fournit des protocoles de gestion afin de négocier l'accès à une ressource partagée tout en assurant que l'utilisation de la ressource respecte bien la politique de cette dernière soumise par le propriétaire [Foster I. and Kesselman C., 2002].

1.2.5.4. La couche collective

Cette couche fournit des protocoles et services qui prennent en charge les interactions entre une collection de ressources, elle est responsable de l'ordonnancement et la co-allocation de ces dernières dans le cas ou un utilisateur demande plusieurs ressources en même temps. En outre, cette couche joue le rôle d'annuaire pour stocker les différentes ressources ainsi que

leurs caractéristiques (fournies par la couche ressource). Ces annuaires permettent au courtier (RB : Resource broker) de faire le choix le plus adapté lors de l'exécution d'une tâche particulière [Foster I. and Kesselman C., 2002].

1.2.5.5. La couche application

Cette couche représente la couche supérieure de l'architecture proposée par Ian Foster et Kesselman C. Elle contient les applications de la grille qui sont construites en utilisant les services définis dans les couches inférieures, ce genre d'application peut directement accéder à la ressource ou bien en utilisant une API (Application Provider Interface) [Foster I. and Kesselman C., 2002].

1.2.6. Les différents types de grilles informatiques

Les grilles informatiques peuvent être partagées en trois types (figure 1.3) et cela en prenant en considération le service qu'elles offrent.

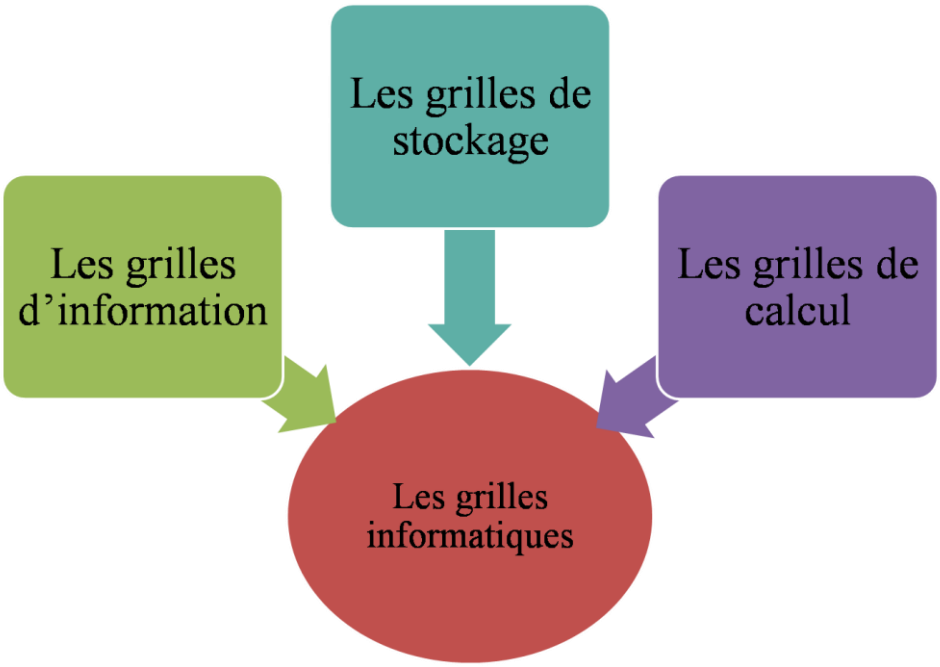

Figure 1. 3 Les différents types de grilles

1.2.6.1. Les grilles d'information

Ce type de grille partage des informations entre plusieurs consommateurs et applications. Elle traite l'information comme une ressource qui sera partagée via la grille [Acqnotes, 2018]. Le web représente l'un des exemples les plus remarquables des grilles d'information. Le problème avec ce type de grille est que l'information partagée peut ne pas être fiable car sa

source est inconnue. Un autre exemple d'une grille d'information est la grille d'information globale (GIG : Global Information Grid) qui a été crée par le département de la défense des Etats Unis (DoD : Department of Defense). Cette grille permet la transmission d'informations dans le monde entier, d'autre part du personnel doit être disponible pour recueillir, traiter, sauvegarder et gérer ces informations [iphc, 2017].

1.2.6.2. Les grilles de stockage

Ce type de grille donne la possibilité de stockage externalisé des données ce qui permet un partage de différents types de données (scientifique, musique, vidéo,…). Le cas le plus représentatif de ce type de grille est le réseau pair à pair qui permet le partage de fichier et le flux multimédia.

1.2.6.3. Les grilles de calcul

La grille de calcul est une infrastructure informatique destinée à mettre à la disposition des utilisateurs des ressources pour réaliser un calcul distribué et stocker des données [in2p3, 2018]. Elle permet l'accès d'une manière transparente et facile à ces ressources distribuées, hétérogènes et appartenant souvent à différents domaines administratifs [Wlcg, 2018]. Un des exemples de grilles de calcul est représenté par le WLCG (Worldwide LHC Computing Grid) qui est un projet de collaboration global de plus de 170 centres de calcul dans 42 pays liant des infrastructures de grilles nationales et internationales [Resinfo, 2018]. Il permet de fournir la capacité de stockage et de traitement des données produites par le grand collisionneur de Hardrons (LHC : Large Hadron Collider). Il s'agit du plus grand accélérateur de particules au monde, construit par le CERN (organisation européenne pour la recherche nucléaire) [Ortiz A, 2009].

- ***Les approches utilisées par une grille de calcul***

La manière dont est fait le calcul sur une grille permet d'avoir les trois approches (figure 1.4) suivantes :

- **Virtual supercomputing**

Ce type consiste à construire un supercalculateur virtuel à l'échelle d'internet, la boite à outils Globus [Globus, 2018] utilise cette approche pour le calcul.

❖ **Internet computing**

Ce type consiste à utiliser les machines en attente sur internet. Cette approche est facile à mettre en œuvre et c'est la plus répandue. Elle permet de combiner la puissance de calcul de machines complètement différentes sur un même projet [Noureddine R., 2010].

❖ **Le metacomputing**

Ce type consiste à acheter du service de calcul sur internet (CPU+ logiciel). Ce principe est utilisé lorsqu'un client a besoin d'une puissance de calcul pendant une période bien déterminée, il n'a qu'à acheter le service sur internet [Noureddine R., 2010].

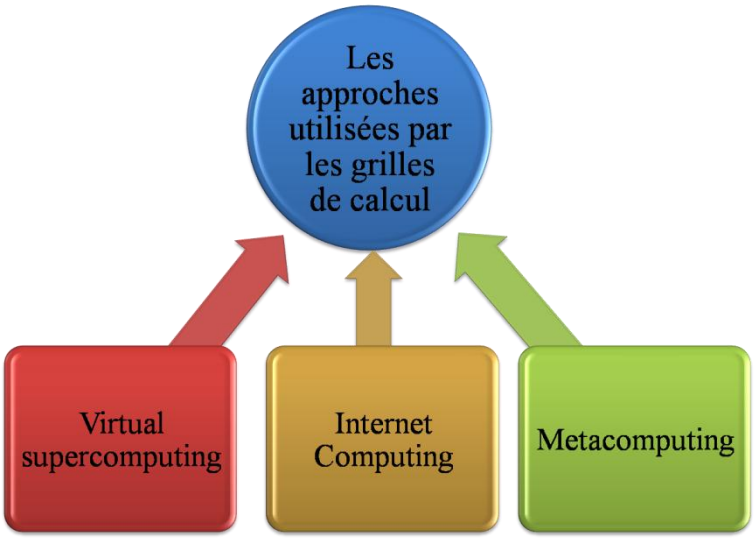

Figure 1. 4 Les approches utilisées par une grille de calcul

- ***Fonctionnement d'une grille de calcul***

La grille de calcul prend en charge l'exécution des jobs soumis par les utilisateurs. Cette exécution fait intervenir plusieurs composants (figure 1.5). Dans ce qui suit nous allons citer ces composants et expliquer le rôle de chacun. En suite, l'interaction entre ces composants sera présentée pour bien éclaircir le fonctionnement d'une grille de calcul.

Les utilisateurs d'une grille de calcul soumettent leurs jobs en utilisant le système de gestion de jobs (WMS : Workload Management System) qui s'occupe de la gestion, de l'ordonnancement et de la distribution des ressources. Il se compose des éléments suivants :

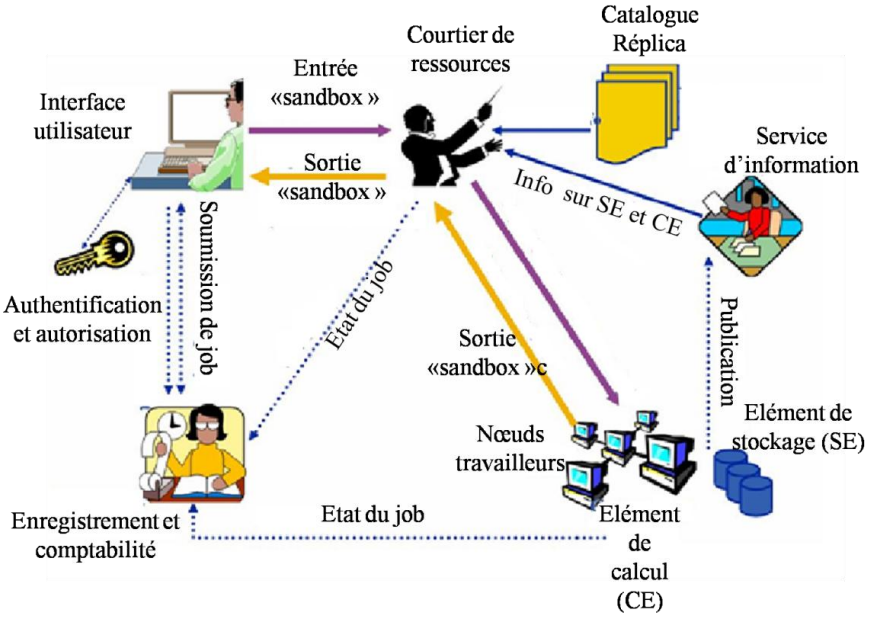

Figure 1. 5 Fonctionnement d'une grille de calcul [EGEE-II, 2014]

- ✓ **L'interface utilisateur (UI : User Interface)** : c'est l'interface qui permet à l'utilisateur d'accéder à la grille de calcul afin de soumettre son job.
- ✓ **Le service d'information (IS : Information Service)** : c'est un système qui fournit les informations concernant les ressources. Ces dernières peuvent être de deux types soit de calcul (CE :Computing Element) ou bien de stockage (SE : Storage Element).
- ✓ **Le courtier de ressource (RB : Resource Broker) :** c'est une machine qui se charge de la recherche des éléments de calcul appropriés disponibles sur la grille pour l'exécution des jobs.
- ✓ **L'élément de calcul (CE : Computing Element) :** cet élément permet l'accès unifié à des ressources de calcul, il prend en charge les jobs qui lui sont attribués tout en gérant une liste de jobs à soumettre (job queue).
- ✓ **Les éléments de stockage (SE : Storage Element) :** ces éléments gèrent le stockage des données.
- ✓ **Les nœuds travailleurs (WN : Worker Nodes) :** c'est un groupe de machines qui prend en charge l'exécution des jobs ou le stockage des données transmises par les SE.
- ✓ **Bac à sable d'entrée (Input Sandbox) :** est une boite qui permet à l'utilisateur de transmettre ses fichiers d'entrée.

- ✓ **Bac à sable de sortie (Output Sandbox) :** est une boite qui permet au courtier de ressources (RB) de récupérer les fichiers de sortie.

L'interface utilisateur (UI) permet à l'utilisateur de transmettre le job à exécuter au courtier de ressource (RB). Ce dernier consulte le service d'information (IS) afin de trouver les éléments de calculs (CE) pouvant prendre en charge l'exécution du job soumis. Les fichiers d'entrée soumis par l'utilisateur dans l'Input Sandbox ainsi que le job sont envoyés à l'élément de calcul qui prend en charge le job dans la queue des jobs. Ensuite, l'élément de calcul envoie le job sur un ou plusieurs nœuds travailleurs (WN) disponibles. Dès la fin de l'exécution du job, le courtier de ressource (RB) est averti pour récupérer les fichiers de sortie dans l'Output Sandbox. Les résultats sont ensuite transférés à l'utilisateur via l'UI. Il faut noter que l'utilisateur peut à tout moment avoir des informations sur l'état de son job en utilisant le service d'enregistrement et de comptabilité qui conserve une trace de l'exécution [EGEE-II, 2014].

1.2.7. Les topologies des grilles

Dans cette partie, les grilles informatiques sont classées d'un point de vue topologique par ordre croissant d'étendue géographique et de complexité (type de ressources, type de réseaux d'interconnexion et domaine de sécurité), ce qui donne : les intra-grilles, les extra-grilles et les inter-grilles (figure 1.6).

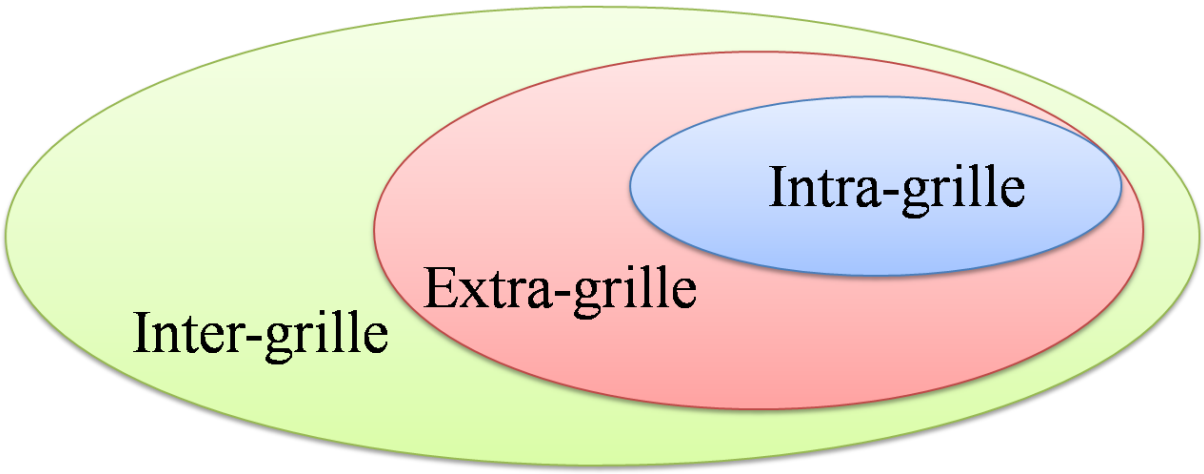

Figure 1. 6 Les topologies des grilles

1.2.7.1. Intra-grille (En analogie avec Intranet)

Ce type de grille est le plus simple, il est souvent déployé au sein d'une organisation unique, qui utilise un réseau d'interconnexion performant et un ensemble statique et homogène de

ressources afin d'augmenter la puissance de calcul. Cette grille prend en charge un seul domaine de sécurité et elle est maitrisée par les administrateurs de l'organisation.

1.2.7.2. Extra-grille (En analogie avec Extranet)

Cette topologie est basée sur l'agrégation de plusieurs intra-grilles. Ce qui permet d'unir les grilles de différentes organisations en utilisant un réseau d'interconnexion hétérogène haut et bas débit (LAN/WAN). En outre, cette topologie est caractérisée par plusieurs domaines de sécurité distincts et un ensemble de ressources dynamiques.

1.2.7.3. Inter-grille (En analogie avec Internet)

Une inter-grille consiste à unir les grilles de multiples organisations utilisant un réseau d'interconnexion très hétérogène haut et bas débit (LAN/WAN). Ce qui donne une topologie caractérisée par un ensemble de domaines distincts avec différentes politiques de sécurité et un ensemble de ressource très dynamique.

1.2.8. Les différents projets de grilles informatiques et les Middlewares

Dans cette partie, nous allons commencer par définir le middleware (intergiciel) qui est souvent confondu avec les projets de grilles proposés. Ensuite, nous donnerons les différents middlewares qui ont été présentés suivis par les plus importants projets de grilles.

1.2.8.1. Les middlewares

Le middleware (intergiciel) est une couche logicielle qui doit être installée au cœur de la grille de calcul entre la couche fabrique et la couche application. Elle permet l'utilisation des ressources de différents types (calcul ou stockage) en assurant une communication transparente pour l'utilisateur. Le middleware doit être en mesure d'identifier et d'utiliser efficacement les ressources disponibles en fournissant des mécanismes de réservation et d'allocation. Il gère également la sécurité, l'accès et l'échange d'information. L'évolution des grilles a donné naissance au principe d'organisation virtuelle, qui consiste à unir plusieurs institutions ou individus, partageant leurs ressources entre plusieurs domaines administratifs afin de résoudre un problème particulier. Ces organisations virtuelles sont gérées par la couche middleware. Le middleware prend également en charge la sécurité de la grille. Le processus d'authentification, le contrôle d'accès ainsi que les politiques de sécurité forment un composant important du middleware. Enfin le middleware représente le cœur de la grille, il se charge de la réservation et l'allocation des ressources, l'ordonnancement et l'exécution de

tâches, suit l'activité du système et prend en charge les pannes qui peuvent survenir. Plusieurs middlewares ont été proposés, dans ce qui suit on va prendre en considération les plus connus.

- **Globus**

La boite à outils open source Globus est une technologie de base fondamentale pour les grilles. Elle permet aux utilisateurs de partager en toute sécurité la puissance informatique, les bases de données et d'autres outils en ligne à travers les frontières corporatives, institutionnelles et géographiques sans sacrifier l'autonomie locale. Cette boite comprend des services logiciels et des bibliothèques pour la surveillance, la découverte et la gestion des ressources, ainsi que la sécurité et la gestion des fichiers. En plus d'être un élément central des projets scientifiques et d'ingénierie qui totalisent près d'un demi-milliard de dollars à l'échelle internationale, la boite à outils Globus est un substrat sur lequel les principales sociétés IT construisent des produits « grille » commerciaux importants [Globus, 2018]. Plusieurs versions de Globus ont été proposées, la dernière version complète et disponible est la version 5 (GT5). Dans ce qui suit nous allons expliquer les différents composants de cette boite à outils qui sont illustrés dans la (figure 1.7).

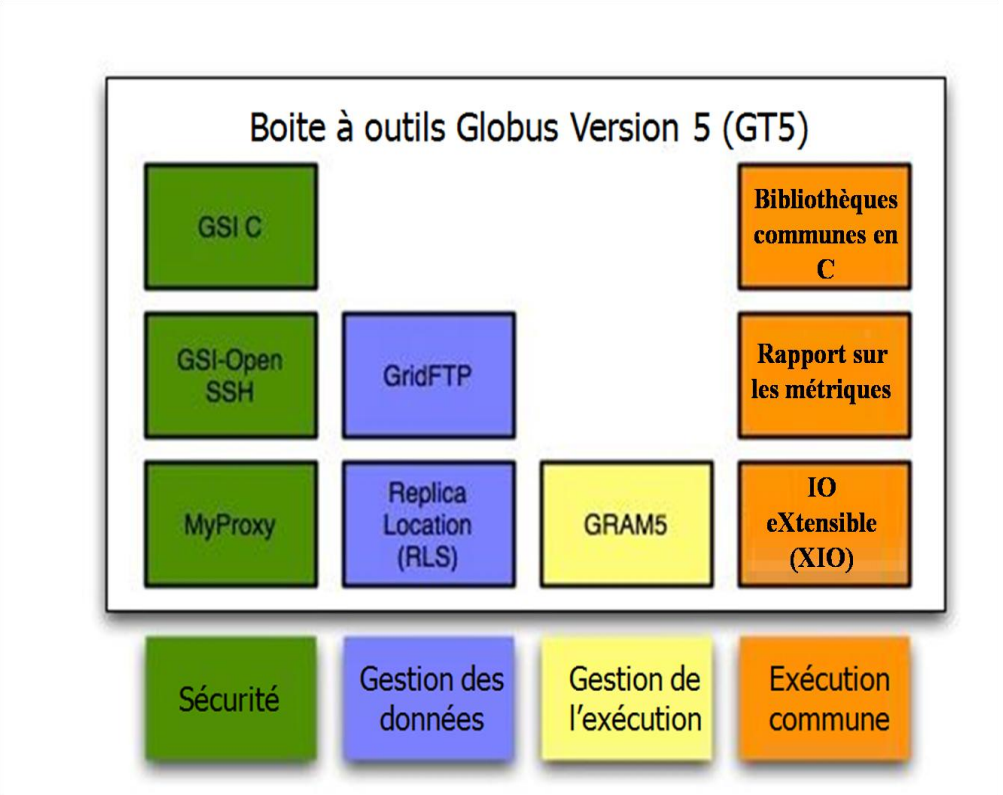

Figure 1. 7 Architecture de la boite à outils Globus 5 [Vachhani M.K. and Atkotiya K.H., 2012]

Pour gérer la sécurité, la boite à outils Globus version 5 offre les éléments suivants [Vachhani M.K. and Atkotiya K.H., 2012] :

- **L'infrastructure de sécurité de la grille en c (GSI-C) :** ce composant offre des API et des outils pour l'authentification, l'autorisation et la gestion de certificats. L'API d'authentification est basée sur l'infrastructure à clés publique (PKI) utilisant les certificats X.509 et TLS. Ce composant prend également en charge le processus de délégation en utilisant les certificats proxy X.509. Pour le contrôle d'accès, un couple d'API est utilisé, la première consiste en une API d'autorisation générique qui permet un contrôle d'accès basé sur une chaine de certificats X.509 de l'utilisateur. La seconde API fournit une liste de contrôle d'accès simple qui mappe les entités distantes autorisées à des noms d'utilisateurs locaux. GSI utilise aussi une cryptographie à clé publique comme base pour ses fonctionnalités [Article GT5].
- **Myproxy :** est un logiciel open source qui gère la sécurité des certificats X.509 de l'infrastructure PKI [article GT5].
- **GSI-Open-SSH :** est une version modifiée d'Open-SSH en ajoutant du support au certificat proxy X.509 pour l'authentification et la délégation fournissant un accès distant avec une connexion unique ainsi que le transfert de fichier.

Pour gérer les données, la boite à outils Globus version 5 offre les éléments suivants [Vachhani M.K. and Atkotiya K.H., 2012]:

- **GridFTP :** qui est une version de FTP spécifique pour les grilles, elle permet le transfert de données volumineux en assurant la sécurité de ces dernières.
- **Replica location (RLS) :** Ce service est un outil qui permet de garder une ou plusieurs copies des fichiers dans un environnement de grilles. C'est un élément du service de gestion des données, un simple registre qui permet de garder trace des copies sur le système de stockage physique [Globus, 2016][c].

Pour gérer l'exécution, la boite à outils Globus version 5 offre les éléments suivants [Vachhani M.K. and Atkotiya K.H., 2012] :

- **Globus Resource Allocation Manager (GRAM 5) :** permet d'assurer l'initiation, la surveillance, la gestion, l'ordonnancement et / ou la coordination des calculs à distance. Afin de résoudre des problèmes tels que le transfert de données, la délégation des informations d'identification du proxy ainsi que la surveillance et la gestion des jobs, le

serveur GRAM est déployé avec des serveurs de délégation et de transfert de fichiers fiables (RFT : Reliable File Transfer). GRAM dépend généralement d'un mécanisme local pour démarrer et contrôler les tâches.

Pour permettre aux services web d'être indépendants des plateformes, la boite à outils Globus version 5 offre les éléments d'exécution communs suivants :

- **XIO** : est une librairie d'entrée/sortie extensible écrite en langage C pour la boite à outils Globus. Elle fournit une seule API (ouvrir, fermer, lire, écrire) qui supporte plusieurs protocoles tels que : TCP, UDP, http, …

- **Bibliothèques communes en C**: permettent la création de l'infrastructure de la grille. Ces librairies fournissent une couche abstraite pour les types de données, les appels système et les structures de données utilisées entre la boite à outils et les applications qui l'utilisent.

- **gLite**

Il représente un ensemble intégré de composants conçus pour permettre le partage de ressources. En d'autres termes, c'est un intergiciel qui peut être utilisé pour construire une grille [grid-deployment, 2017]. Le middleware gLite a été proposé par le projet EGEE (Enabling Grid for E-sciences in Europe) puis développé par le projet EMI (European Middleware Initiative). En plus du code développé dans le projet, la distribution gLite a rassemblé les contributions de nombreux autres projets. Le modèle de distribution consistait à construire différents services (types de nœuds) à partir de ces composants, puis à faciliter l'installation et la configuration sur les plates-formes choisies [grid-deployment, 2017].

- **Condor G**

Condor est un outil pour exploiter la capacité des stations de travail inactives pour les tâches de calcul. Condor est bien adapté aux études de paramètres et au calcul à haut débit, où les tâches n'ont généralement pas besoin de communiquer entre elles. Nous pouvons classer Condor comme un système spécialisé de gestion de la charge des jobs nécessitant beaucoup de calculs. Comme d'autres systèmes de traitement par lots complets, Condor fournit un mécanisme de mise en file d'attente des jobs, une politique de planification, un schéma de priorités et prend en charge la surveillance des ressources et la gestion de ces dernières. Lors de la réception de jobs en série ou en parallèle de l'utilisateur, le système Condor les place

dans une file d'attente, choisit quand et où exécuter les jobs en fonction d'une politique, surveille attentivement leur progression et informe finalement l'utilisateur [Joshy J. and Craig F., 2003]. En outre, Condor peut être utilisé pour gérer un cluster de nœuds de calcul dédiés. Il est adapté pour exploiter efficacement la puissance du processeur à partir de postes de travail inactifs. Condor dispose de mécanismes pour faire correspondre les demandes de ressources (jobs) avec les offres de ressources (machines). Alors que les outils logiciels Condor s'attachent à exploiter la puissance des ressources opportunistes et dédiées, Condor-G est un système logiciel dérivé qui exploite les logiciels de Condor et de Globus en mettant l'accent sur les services de gestion des tâches pour les applications de grilles. Il s'agit d'une combinaison de protocoles de gestion des ressources inter-domaines de Globus (GRAM, Index Services) avec les méthodes de gestion des ressources intra-domaines de Condor [Joshy J. and Craig F., 2003].

- **Legion**

Legion un projet de middleware initié par l'Université de Virginie. Il représente un logiciel de méta-systèmes à base d'objets pour les applications de grille. L'objectif du projet Legion est de promouvoir la conception du software des systèmes distribués en fournissant des représentations d'objets standards pour les processeurs, les systèmes de données, les systèmes de fichiers, etc. Les applications de Legion sont développées en fonction de ces objets standards. Des groupes d'utilisateurs peuvent construire un espace de travail virtuel partagé pour collaborer dans la recherche et échanger des informations [Globus, 2018].

- **UNICORE**

Le projet UNICORE [Erwin D.W. and Snelling D.F, 2001] a été financé par le ministère allemand de l'éducation et de la recherche pour concevoir : une interface utilisateur graphique (GUI) uniforme et facile à accéder, une architecture ouverte basée sur le concept de job abstrait, une architecture de sécurité cohérente, une interface minimale avec les procédures administratives locales ainsi que l'exploitation des technologies existantes et émergentes y compris le web et le langage java [Joshy J. and Craig F., 2003]. UNICOREpro a été produit au sein de l'UNICORE, afin de pouvoir fournir une interface uniforme pour la préparation des jobs et la soumission sécurisée de ces derniers, similaire à un portail. Cela permet aux utilisateurs de créer un flux de travail pour l'exécution des jobs et contrôler les comportements d'exécution. UNICOREpro est un projet open source développé en utilisant le langage Java, il offre des fonctionnalités d'autorisation, de gestion de jobs, de transfert de données et une

interface de traitement par lots. Un projet appelé GRIP (GRid Interoperability Project) a été lancé en 2002 pour réaliser l'interopérabilité entre UNICORE et Globus. La grille EUROGRID est basée sur le système UNICORE développé et utilisé par les principaux centres High Performance Computing (HPC) allemands [Joshy J. and Craig F., 2003].

- **Advanced Resource Connector (ARC)**

Est une solution logicielle open source qui a été introduite par NorduGrid. Ce middleware offre une interface commune pour la soumission de jobs. Il permet donc de créer des infrastructures de grille de taille et de complexité variables. Depuis la première version (mai 2002), le middleware ARC est déployé et utilisé dans des environnements de production. L'accent a été mis sur son évolutivité, sa stabilité, sa fiabilité et sa performance [Kónya B. ,2004].

1.2.8.2. Les différents projets de grilles réalisés

Plusieurs projets de grilles ont été réalisés, la plupart utilisent un des middlewares cités précédemment, dans ce qui suit nous allons présenter les projets les plus importants.

- **Le projet EGEE**

EGEE (Enabeling Grids for E-science in Europe) est un programme qui est passé par plusieurs phases, il a commencé en 2004 jusqu'à la fin de 2010. Ce projet a pu unir 250 centres de ressources de 48 pays offrant 50000 CPU et plusieurs péta-bytes de stockage. Cette infrastructure est utilisée généralement par 5000 utilisateurs formant 200 organisations virtuelles et exécutant exactement prêt de 140000 jobs par jour. Cette grille européenne utilise le middleware gLite pour gérer son infrastructure, elle a été crée au départ pour la physique nucléaire puis elle a été ouverte à toutes les communautés scientifiques (EGEE II) [Wang L et Al., 2009].

- **Le projet DEISA**

Le DEISA (Distributed European Infrastructure for Supercomputing Applications) est un projet de l'union de superordinateurs européens qui a été lancé de 2004 à 2008. Il a pu unir 11 centres de super-calcul de 7 pays européens. Depuis mai 2008, le consortium a continué à développer l'infrastructure via le projet DEISA II jusqu'à 2011 [Gentzsch, W. et Al., 2011]. Les activités et les services relatifs à la mise en œuvre, au fonctionnement et aux technologies des applications sont poursuivis et améliorés, car ils sont indispensables au soutien efficace des sciences de calcul dans le domaine des HPC. L'infrastructure résultante est incomparable

dans le monde entier dans son hétérogénéité et sa complexité, permettant l'exploitation d'une puissante grille de superordinateurs construite sur les services nationaux, facilitant la capacité de l'Europe à entreprendre des recherches scientifiques à l'échelle mondiale [Gentzsch, W. et Al., 2011].

- **Le projet Grid '5000**

Est une grille française qui fournit une infrastructure matérielle et logicielle interconnectant à très haut débit une dizaine de grappes de PC oŭ chacune comprend 500 unités de calcul, ce qui donne le total 5000 d'où vient le code du projet Grid'5000 [Inria, 2017]. Le premier objectif de cette grille était de donner aux informaticiens les moyens expérimentaux pour mener à bien les recherches dans le domaine des grilles [Inria, 2017].

- **Le projet TeraGrid**

Est un projet qui a été financé par la fondation nationale de la science aux Etats Unis. C'est une infrastructure de découverte scientifique ouverte. Onze sites partenaires ont été utilisés pour créer une ressource de calcul intégrée et persistante. TeraGrid sert plus de 4500 chercheurs de plus de 300 collèges, universités et instituts de recherche aux Etats Unis [Internet 2, 2017]. Selon une déclaration de la NSF (National Science Foundation), TeraGrid a été utilisée pour réaliser des tâches complexes telles que : la modélisation sismique et la compréhension de l'énergie sombre [Computerworld, 2016].

- **Le projet NorduGrid**

Le but du projet NorduGrid est de créer l'infrastructure de calcul de grille dans les pays nordiques, en utilisant les outils logiciels fournis par les projets de recherche et de développement des grilles dans le monde entier. Au cours de l'évaluation de tels outils par ce projet, il s'est avéré que rien ne répondait aux exigences fixées par les participants au projet. Par conséquent, il a été décidé de développer un ensemble d'outils originaux et innovants afin d'atteindre l'objectif de permettre de garantir des services de grille au niveau de la production dans les pays nordiques [Nordugrid, 2018]. Les participants au projet comprennent des universités et des centres de recherche au Danemark, en Suède, en Finlande et en Norvège. La phase active du projet a débuté en mai 2001 avec la participation de l'Institut Niels Bohr (Danemark), des Universités Lund et Uppsala (Suède), de l'Université d'Oslo (Norvège) et de l'Institut de physique d'Helsinki (Finlande). Les centres nationaux de superordinateurs en Norvège (Parallab) et en Suède (NSC, HPC2N) sont entrés dans le projet en installant leurs ressources avec le middleware ARC.

1.2.9. Les besoins de sécurité dans les grilles

La création d'une grille implique le partage de ressources de différentes natures d'une manière coordonnée. En outre, une grille permet la collaboration et le partage d'unité de calcul et de stockage afin de garantir les besoins qu'une seule organisation ne peut offrir. Les ressources partagées ainsi que les utilisateurs de la grille appartiennent souvent à différents domaines administratifs et sont géographiquement distribués ce qui rend la gestion de la sécurité d'un tel système difficile. Ceci est dû au fait que les utilisateurs de la grille peuvent avoir des intérêts contradictoires et souhaiteraient donc avoir l'assurance que leurs transactions soient à l'abri des yeux des autres utilisateurs. Les problèmes de sécurité liés à la grille sont vastes, c'est ce qui nous a poussé à se poser plusieurs questions : Comment garder les informations personnelles privées ? Comment contrôler l'accès aux différentes ressources ? Comment assurer qu'un utilisateur donné ne bloque pas toutes les ressources de la grille? Donc l'issue de sécurité joue un rôle très important dans le domaine des grilles.

1.3. Le Cloud Computing (Informatique en nuage)

1.3.1. Définition et caractéristiques

L'institut national des standards et de la technologie (NIST) a défini le Cloud Computing (informatique en nuage) comme un modèle permettant d'accéder à un réseau partagé de ressources informatiques configurables (réseaux, serveurs, stockages, applications et services), qui peuvent être rapidement provisionnées et libérées avec un minimum d'effort de gestion [Mell P. and Grance T., 2011]. Selon la définition donnée par le NIST, le Cloud Computing est caractérisé par les cinq points suivants :

- **Accès aux services par l'utilisateur à la demande**

Le cloud permet aux consommateurs de se procurer les ressources d'une manière simple et flexible sans avoir besoin d'une interaction humaine avec chaque fournisseur de service. La puissance de calcul ainsi que la capacité de stockage sont adaptées automatiquement aux besoins des consommateurs [Mell P. and Grance T., 2011].

- **Accès réseau large bande**

Les services sont disponibles sur le réseau et sont accessibles via des mécanismes standards qui favorisent l'utilisation par des plateformes « client » hétérogènes minces ou épaisse (ex :

téléphone mobile, tablette, ordinateur portable ou poste de travail) [Mell P. and Grance T., 2011].

- **Mise en commun des ressources (pooling)**

Les ressources du fournisseur sont mises en commun entre plusieurs consommateurs en utilisant un modèle multi-tenants (locations multiples) avec plusieurs ressources physiques et virtuelles attribuées dynamiquement et réaffectées en fonction de la demande des utilisateurs. Généralement, le consommateur n'a aucune information sur l'emplacement exacte des ressources fournies mais peut spécifier l'emplacement à un niveau d'abstraction supérieur (pays, état ou centre de données) [Mell P. and Grance T., 2011].

- **Redimensionnement rapide (élasticité)**

Les services peuvent être provisionnés et libérés de manière élastique c'est-à-dire les services disponibles semblent souvent illimités aux consommateurs et peuvent être appropriés en n'importe quelle quantité à tout moment [Mell P. and Grance T., 2011].

- **Facturation à l'usage**

Les systèmes Cloud contrôlent et optimisent automatiquement l'utilisation des ressources en exploitant une capacité de mesure à un niveau d'abstraction approprié au type de service (par exemple : stockage, traitement, bande passante et comptes d'utilisateurs actifs). L'utilisation des ressources peut être surveillée, contrôlée et signalée, ce qui garantit la transparence tant pour le fournisseur que pour le consommateur du service utilisé. Ceci permet aux consommateurs de payer uniquement ce qu'ils consomment [Mell P. and Grance T., 2011].

1.3.2. Histoire du Cloud Computing

Le Cloud Computing a évolué à travers un certain nombre de phases incluant les grilles et l'informatique utilitaire. Il a emprunté de nombreux concepts de grilles, cependant les ressources informatiques peuvent être allouées de façon dynamique. De plus, le Cloud peut être développé avec des environnements non-grille comme une architecture web exécutant des applications traditionnelles ou les applications web2. La base du Cloud est l'informatique utilitaire, c'est une nouvelle génération d'informatique après le mainframe, l'ordinateur personnel, l'informatique client serveur et le web. Comme tout contexte relevant autant de l'économie que de la technologie, il est difficile de dire avec précision quand a été inventé le

Cloud, nous allons dans cette partie essayer en premier lieu de connaître l'origine de cette technologie, en second lieu on essayera de citer les projets de Cloud les plus important avec leurs dates (figure 1.8). Certains attribuent le concept du Cloud au scientifique John McCarty qui a proposé en 1960 l'idée que les calculs soient livrés comme un service public [Computerweekly, 2018]. Dans la fin des années 1990, le terme Cloud a été utilisé pour représenter l'espace de calcul entre le fournisseur et l'utilisateur [eci, 2018]. En 1997, le professeur Ramnath Chellapa de l'université Emory a défini le Cloud comme un nouveau paradigme informatique ou les limites de ce dernier seront déterminées par la logique économique plutôt que par les limites techniques [eci, 2018]. L'un des premiers repères du Cloud Computing était l'arrivée de salforce.com [salesforce, 2018] en 1999, pionnier du concept de fourniture d'applications d'entreprise via un simple site web [salesforce, 2018][b]. L'entreprise de services a ouvert la voie aux entreprises de logiciels spécialisées et traditionnelles pour la diffusion d'applications sur internet. Ensuite, Amazon a développé des services web en 2002 fournissant une suite de services basés sur le Cloud incluant le calcul, le stockage et l'intelligence humaine via Amazon Micanical Turk [mturk, 2018]. En 2005, l'intégration Castle Eze a construit et déployé la première plateforme hébergée du Cloud pour un grand fond de couverture basé à New York [eci, 2018][**b**]. En 2006, Amazon a lancé sa plateforme du Cloud qui regroupe plusieurs services incluant l'Elastic Compute Cloud (EC2) qui permet aux organisations de louer des ordinateurs pour exécuter leurs applications [journaldunet, 2018]. En avril 2008, Google lance son engin d'application pour 2000 développeurs en tant qu'outil pour exécuter des applications web sur l'infrastructure de Google. Les applications devraient être écrites en Python et étaient limitées à 500 Mo de stockage, à 200 millions de mégacycles de CPU et à 10 GO de bande passante par jour [medium, 2018]. En 2010, Microsoft commercialise son premier Cloud nommé Azure, il s'agissait dans un premier temps d'un Cloud de plateforme qui n'a cessé d'évoluer jusqu'à ce qu'il soit complété d'un Cloud d'infrastructure en juin 2012 [lebigdata, 2018]. Ce n'est pas que l'histoire du Cloud s'arrête mais dans cette partie nous avons cité ce qui a été présenté pour la première fois, il faut noter que la plupart des Cloud cités au dessus ont évolué avec le temps et sont arrivés à des versions plus sophistiquées en terme de nombre d'utilisateurs ou de mécanismes et technique utilisés ou bien en terme d'étendue géographique.

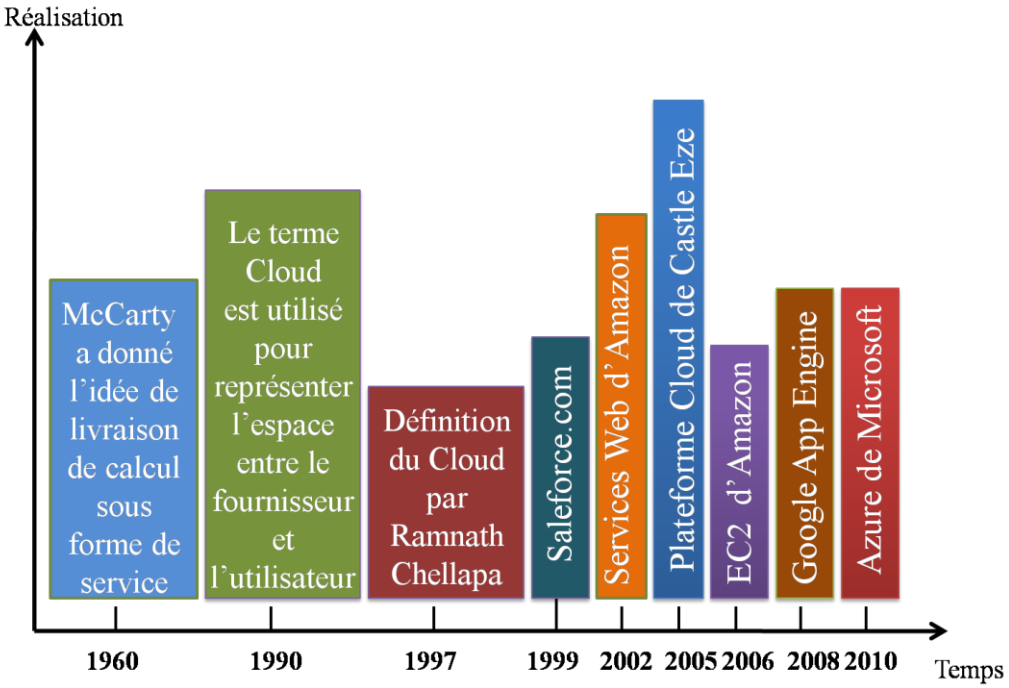

Figure 1. 8 Histoire du Cloud Computing

1.3.3. Avantages et obstacles du Cloud

Le Cloud présente de nombreux avantages. D'abord, ses ressources peuvent être augmentées et réduites à la demande et payées sur une base d'utilisation mesurée. Cette capacité offre d'énormes avantages aux clients car pour implémenter une nouvelle application ils n'ont pas besoin d'investir d'importantes dépenses telles que l'achat de nouveaux équipements et la prise en charge des licences. En outre, les clients ne sont pas obligés de maintenir des systèmes informatiques c'est à dire prendre en charge leur maintenance et leur mise à niveau. L'utilisation des services du Cloud offre la possibilité d'avoir une sécurité améliorée car tenant compte d'une étude qui a été faite en 2010 [salesforce, 2017], les entreprises perdent plus de 263 ordinateurs personnels par an contenant des informations confidentielles ce qui implique de sérieuses conséquences, tandis que l'utilisation du Cloud rend la perte d'un ordinateur personnel moins importante. Enfin, l'un des avantages les plus importants du Cloud est que le client peut accéder n'importe où et à tout moment à ses données ou aux applications et à l'infrastructure du Cloud. Les principaux avantages du paradigme du Cloud peuvent être résumés par : sa flexibilité, son potentiel de réduction des coûts, la disponibilité de très grandes quantités de stockage centralisé des données, les moyens de déploiement rapide des ressources informatiques et son évolutivité.

Bien sûr, le Cloud Computing présente également certains inconvénients. Tout d'abord, comme avec tout système physique, le calcul dans le Cloud doit fonctionner dans les limites physiques. Il offre la possibilité de fournir des quantités massives de puissance de calcul et de stockage, mais ces quantités ne sont pas infinies. Par conséquent, les utilisateurs peuvent devoir adapter leurs applications à un ensemble de catégories d'utilisation des ressources définies par le fournisseur de Cloud. En outre, les services Cloud sont souvent distants donc ils peuvent souffrir de problèmes de latence et de bande passante associés à toute application distante. Second, les services du Cloud hébergés desservent plusieurs clients, divers problèmes liés à plusieurs clients partageant le même morceau de matériel peuvent survenir. Par exemple, si l'application d'un utilisateur compromet le système, elle peut également compromettre les applications d'autres utilisateurs partageant le même système. En plus, le fait d'avoir des données accessibles à des tiers (comme un fournisseur de services Cloud) peut présenter des problèmes de sécurité, de conformité et de réglementation. D'abord, les données transférées dans le Cloud ne sont pas forcément présentes sur le territoire national ce qui empêche le client de savoir le lieu exacte de ses données. En outre, le client n'a aucun accès physique à ces données qui peuvent être très importants. En second lieu, la sécurité des données reste un point très important à gérer. Lorsqu'une entreprise héberge ses données confidentielles dans le Cloud, elle doit s'assurer que ce fournisseur utilise les bonnes méthodes pour assurer la sécurité, c'est à dire est ce que les données sont hébergées sur un seul disque ou bien partitionnées ? Est ce que le fournisseur fait des tests pour vérifier la sécurité du Cloud et est ce que ces tests sont faits de manière régulière ? Est ce que les locaux sont inaccessibles aux personnes mal intentionnées? En plus de l'issue de sécurité, le client doit penser à la pérennité du service c'est à dire est ce que ce fournisseur de service va durer dans le temps car le changement du Cloud prend beaucoup de temps ce qui peut engendrer des problèmes importants dans les entreprises. Enfin, le Cloud repose sur le transfert des données donc l'utilisation d'une bonne connexion internet est indispensable pour pouvoir passer vers une solution Cloud.

1.3.4. Les modèles de services du Cloud

L'architecture du Cloud présentée par le NIST [Mell P. and Grance T., 2011] comprend les trois modèles de services suivants (figure 1.9):

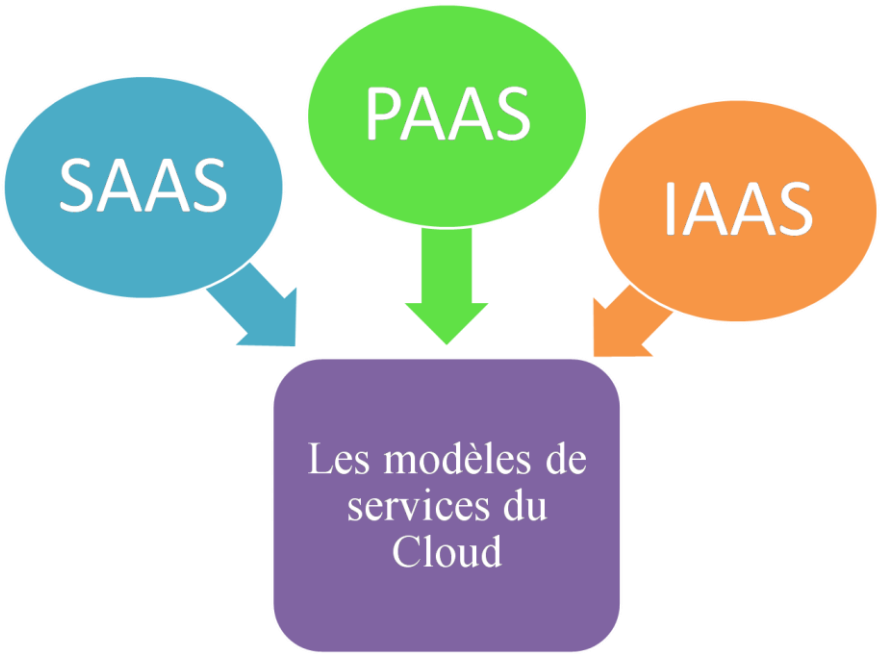

Figure 1. 9 Les modèles de services du Cloud

1.3.4.1. Le logiciel comme service (SAAS : Software As A Service)

Les services offerts au consommateur consistent à utiliser les applications du fournisseur fonctionnant sur une infrastructure du Cloud. Les applications sont accessibles à partir de divers dispositifs clients par le biais d'une interface « client » légère, telle qu'un navigateur web ou une interface de programme. Le consommateur ne peut ni gérer ni contrôler l'infrastructure Cloud sous-jacente, y compris le réseau, les serveurs, les systèmes d'exploitation, le stockage ni même les applications individuelles, à l'exception possible des paramètres de configuration des applications spécifiques à l'utilisateur.

1.3.4.2. La plateforme comme service (PAAS : Platform As A Service)

Les services offerts au consommateur consistent à déployer sur l'infrastructure du Cloud des applications créées ou acquises par le consommateur. Ces applications sont créées à l'aide de langages de programmation, de bibliothèques, de services et d'outils pris en charge par le fournisseur. Le consommateur ne peut ni gérer ni contrôler l'infrastructure Cloud sous-jacente, y compris le réseau, les serveurs, les systèmes d'exploitation ou le stockage, mais il contrôle les applications déployées et éventuellement les paramètres de configuration de l'environnement d'hébergement des applications.

1.3.4.3. L'infrastructure comme service (IAAS : Infrastructure As A Service)

Les services offerts au consommateur consistent à fournir le traitement, le stockage, les réseaux et d'autres ressources informatiques fondamentales où le consommateur est capable de déployer et d'exécuter un logiciel arbitraire, qui peut inclure des systèmes d'exploitation et des applications. Le consommateur ne peut ni gérer ni contrôler l'infrastructure Cloud sous-jacente mais contrôle les systèmes d'exploitation, le stockage et les applications déployées; et éventuellement un contrôle limité des composants réseau sélectionnés (par exemple, les pare-feu hôtes).

1.3.5. Les modèles de déploiement du Cloud

L'architecture du Cloud présentée par le NIST [Mell P. and Grance T., 2011] comprend les quatre modèles de déploiement suivants (figure 1.10):

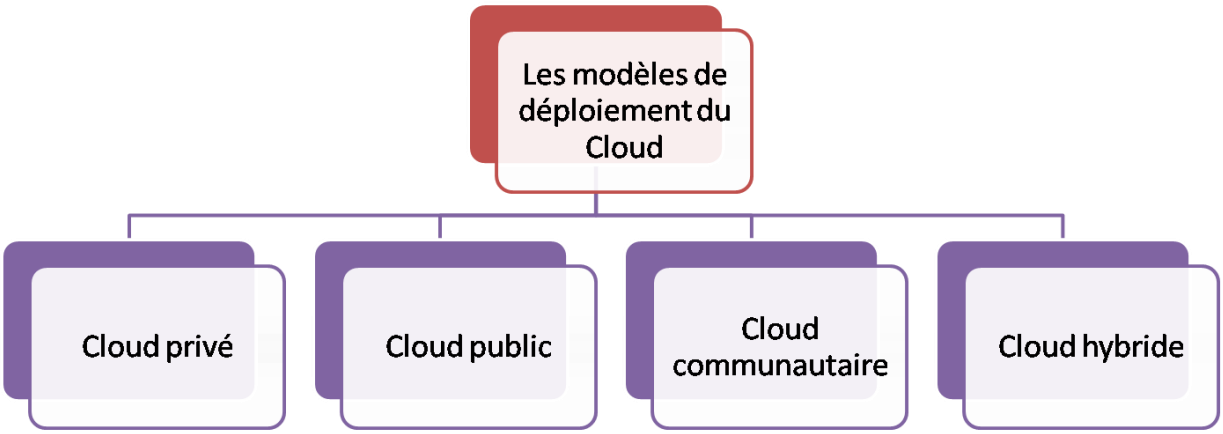

Figure 1. 10 Les modèles de déploiement du Cloud

1.3.5.1. Cloud privé

L'infrastructure du Cloud est fournie pour une utilisation exclusive par une seule organisation comprenant plusieurs consommateurs (par exemple, des unités commerciales). Il peut être détenu, géré et exploité par l'organisation, une tierce partie ou une combinaison d'entre eux.

1.3.5.2. Cloud public

L'infrastructure du Cloud est provisionnée pour une utilisation ouverte au grand public. Il peut être détenu, géré et exploité par un organisme commercial, universitaire ou gouvernemental, ou une combinaison d'entre eux. Il existe dans les locaux du fournisseur du Cloud.

1.3.5.3. Cloud communautaire

L'infrastructure de cloud est mise en service pour une utilisation exclusive par une communauté spécifique de consommateurs provenant d'organisations ayant des préoccupations communes (par exemple : la mission, les exigences de sécurité, les règles et les considérations de conformité). Il peut être détenu, géré et exploité par un ou plusieurs organismes de la communauté, une tierce partie ou une combinaison d'entre eux.

1.3.5.4. Cloud hybride

L'infrastructure cloud est une composition de deux ou plusieurs infrastructures cloud distinctes (privées, communautaires ou publiques) qui restent des entités uniques, mais sont liées par une technologie standardisée ou propriétaire qui permet la portabilité des données et des applications (par exemple : l'éclatement du Cloud entre Clouds). Une organisation peut implémenter un ou plusieurs modèles de déploiement différents, le choix sera selon le modèle qui fournit la meilleure solution. Par exemple, une application critique dont la conformité ou d'autres spécifications de sécurité peuvent nécessiter un modèle de Cloud hybride ou privé. À l'inverse, une application générale qui peut être nécessaire pour un projet temporaire peut être parfaitement adaptée à un Cloud public.

1.3.6. L'architecture du Cloud

Il existe plusieurs versions qui définissent l'architecture du Cloud ; Dans ce qui suit nous allons voir et expliquer deux modèles d'architecture : le premier a été présenté par [Foster I et Al., 2008] où les auteurs ont essayé de proposer un ensemble de couches en se basant sur l'architecture des grilles proposée dans la section 1.2.5. Le second modèle a été présenté par [Buyya R. et Al., 2013].

1.3.6.1. L'architecture proposée par Ian Foster et Al., 2008

Dans [Foster I et Al., 2008], les auteurs ont défini une architecture à quatre couches (figure 1.11) en se basant sur celle qui a été proposée pour les grilles.

- **La couche fabrique**

Cette couche se compose des ressources matérielles brutes, telles que : les unités de calcul de base, les disques de stockage et les bandes passantes réseau. Similaire aux grilles, dans cette couche la plupart des ressources sont hétérogènes. Par exemple, dans un data-center du Cloud, les machines physiques sous-jacentes peuvent être des PC, des stations de travail ou des superordinateurs [Liu X. et Al., 2012].

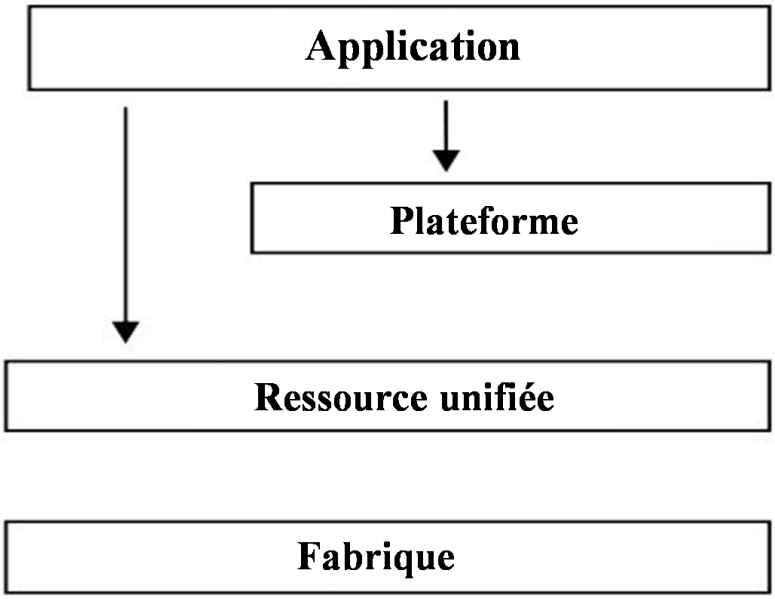

Figure 1. 11 Architecture du Cloud [Foster I et Al., 2008]

- **La couche ressource unifiée**

Cette couche est constituée de ressources hétérogènes qui se présentent généralement sous la forme de ressources virtuelles. Dans cette couche, les machines physiques sous-jacentes ont été abstraites / encapsulées généralement par des outils de virtualisation afin qu'elles puissent être exposées aux couches supérieures et aux utilisateurs finaux comme des ressources intégrées [Liu X. et Al., 2012].

- **La couche plateforme**

Cette couche consiste en un ensemble d'outils de gestion des ressources et de services du middleware au-dessus des ressources unifiées. La couche plateforme peut fournir une plateforme de développement et / ou de déploiement, par exemple, un environnement d'hébergement web, un service de modélisation de flux de travail et un service de planification et ainsi de suite [Liu X. et Al., 2012].

- **La couche application**

Cette couche se compose des applications utilisateur qui peuvent être n'importe quel type d'applications telles que : les applications de flux de travail dans le Cloud, les outils de réseaux sociaux et les sites web de commerce électronique [Liu X. et Al., 2012].

1.3.6.2. L'architecture proposée par Buyya R. et Al., 2013

Dans [Buyya R. et Al., 2013], les auteurs ont proposé une architecture de quatre couches (figure 1.12). Dans cette architecture le middleware est divisé en deux parties, le cœur du middleware et le middleware niveau-utilisateur. La couche la plus basse représente l'infrastructure du Cloud, qui peut être de nature hétérogène car une grande variété de ressources telle que : les clusters et les réseaux de pc, peuvent être utilisés pour la construire. De plus, les systèmes de base de données ainsi que les services de stockage peuvent faire partie de cette infrastructure. La couche suivante représente le cœur du middleware qui a pour objectif de gérer l'infrastructure du Cloud en fournissant un environnement d'exécution d'application et une meilleure utilisation des ressources. Cette couche prend également en charge les technologies de virtualisation qui sont utilisées pour garantir la personnalisation de l'environnement d'exécution, l'isolation des applications et la qualité de service [Buyya R. et Al., 2013]. La virtualisation matérielle est généralement utilisée à ce niveau. Elle prend en charge le groupement des ressources et expose l'infrastructure distribuée sous la forme d'une collection de machines virtuelles. En utilisant la technologie des machines virtuelles, il est possible de partitionner finement les ressources matérielles et de virtualiser des périphériques spécifiques, répondant ainsi aux exigences des utilisateurs et des applications [Buyya R. et Al., 2013]. La combinaison des plateformes d'hébergement du Cloud et les ressources sont généralement classés comme IAAS. Les solutions IaaS sont adaptées à la conception de l'infrastructure système mais fournissent des services limités pour la création d'applications. Ce service est fourni par des outils d'environnement de programmation Cloud, qui forment une nouvelle couche pour offrir aux utilisateurs une plateforme de développement pour les applications. La gamme d'outils comprend des interfaces web, des outils de ligne de commande et des frameworks. Dans ce scénario, les utilisateurs développent leurs applications spécifiquement pour le Cloud en utilisant l'API exposée au niveau du middleware niveau-utilisateur. Pour cette raison, cette approche est également connue sous le nom de Plateforme comme service (PaaS) car le service offert à l'utilisateur est une plateforme de développement plutôt qu'une infrastructure. Les solutions PaaS incluent également l'infrastructure, qui est fournie dans le cadre des services fournis. La couche supérieure de la figure de référence contient les services fournis au niveau de l'application. Ceux-ci sont principalement appelés logiciel comme service (SaaS). Dans la plupart des cas, il s'agit d'applications web qui s'appuient sur le Cloud pour fournir des services aux utilisateurs finaux. La puissance du Cloud fournie par les solutions IaaS et PaaS permet aux éditeurs de

logiciels indépendants de fournir leurs services applicatifs via internet. Les autres applications appartenant à cette couche sont celles qui exploitent fortement internet pour leurs fonctionnalités de base qui reposent sur le Cloud afin de supporter un plus grand nombre d'utilisateurs; C'est le cas des portails de jeux et, en général, des sites de réseaux sociaux [Buyya R. et Al., 2013].

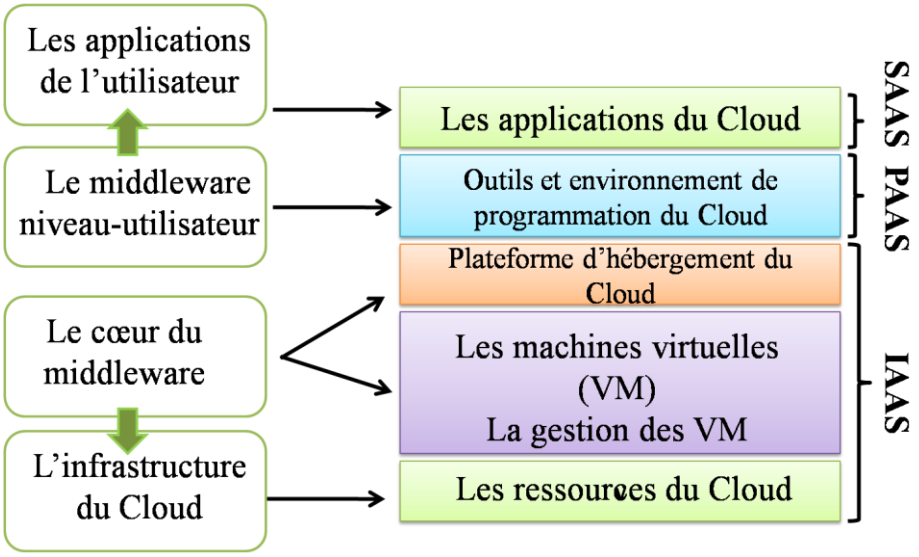

Figure 1. 12 Architecture du Cloud [Buyya R. et Al., 2013]

1.3.6.3. L'architecture proposée par le NIST

L'architecture de référence du Cloud Computing présentée par le NIST [Hogan M et Al., 2011] a défini cinq acteurs participants au fonctionnement de cette architecture à savoir : Le consommateur du Cloud, le fournisseur du Cloud, l'auditeur du Cloud, le courtier du Cloud et le transporteur du Cloud. Dans ce qui suit nous allons définir chaque acteur et expliquer les interactions possibles entre tous ces acteurs.

- **Le consommateur du Cloud**

Est l'acteur final pour lequel le service du Cloud a été créé. Un consommateur du Cloud représente une personne ou une organisation qui entretient une relation commerciale avec le fournisseur du Cloud et utilise le service créé. Cet acteur parcourt le catalogue de service du fournisseur du Cloud, demande le service voulu, établie des contrats de services avec le fournisseur, utilise le service et enfin il paye ce qu'il a consommé [Hogan M et Al., 2011].

- **Le fournisseur du Cloud**

Cet acteur peut être une personne, une organisation ou une entité chargée de mettre un service à la disposition des consommateurs [Hogan M et Al., 2011]. Le fournisseur du Cloud se charge des activités suivantes :

- Construit les services de type logiciel, plateforme ou infrastructure demandés.
- Gère l'infrastructure technique requise pour fournir les services demandés.
- Fournit les services.
- S'occupe de la sécurité de ces services.

- **L'auditeur du Cloud**

Est une partie qui peut effectuer une évaluation indépendante : des services du Cloud, des opérations du système d'information, des performances et de la sécurité du Cloud [Hogan M et Al., 2011].

- **Le courtier du Cloud**

Est une entité qui gère l'utilisation, la performance et la délivrance des services du Cloud. Cette entité négocie également les relations entre le fournisseur du service et le consommateur du Cloud. On fait appel au courtier lorsque les services sont difficiles à gérer par le consommateur, au lieu de les demander au fournisseur du Cloud, le consommateur les demande au courtier [Hogan M et Al., 2011].

- **Le transporteur du Cloud**

C'est un intermédiaire qui fournit la connectivité et le transport des services depuis le fournisseur vers le consommateur du Cloud. Le transporteur fournit l'accès au consommateur via les réseaux, la télécommunication et des dispositifs d'accès [Hogan M et Al., 2011].

1.3.7. Les différents projets de Cloud

Dans cette partie nous allons présenter les projets de Cloud les plus importants.

1.3.7.1. Google App Engine

Google App Engine est un service qui permet d'héberger des applications web dans l'infrastructure Google existante. Il permet également aux développeurs de concevoir, développer et déployer des applications Java et Python dans des environnements Java, Go et Python sans être obligé de maintenir des serveurs. En outre, cet outil offre un support pour une multitude de langages de programmation. Le service prend également en charge les

environnements standards et flexibles. Les utilisateurs ont juste besoin de choisir un environnement, sélectionner un langage et programmer [ikbooks, 2017].

1.3.7.2. Microsoft Azure

Microsoft propose une plateforme cloud au nom de Window Azure. Il s'agit d'un environnement de développement, d'hébergement et de gestion qui offre une capacité de calcul et de stockage à la demande au niveau de l'entreprise. Pour utiliser les fonctionnalités du Cloud Azure, vous devez utiliser l'API Azure. Windows Azure est hébergé dans le centre de données Microsoft et fournit des systèmes d'exploitation, des outils de développement pour créer des applications web pouvant également avoir une interface avec des périphériques locaux. Ces applications peuvent être développées à l'aide de l'environnement de développement Visual Studio et du framework .NET. Il prend également en charge plusieurs protocoles internet, notamment HTTP, REST, SOAP et le standard XML. Ses différents composants de support sont les suivants [ikbooks, 2017]:

- **SQL Azure** : offre des fonctionnalités Microsoft SQL Server pour les applications basées sur le Cloud afin de stocker des données structurées, semi-structurées et non structurées.
- **Windows Azure Marketplace** : est un marché en ligne permettant aux développeurs d'applications d'acheter et de vendre du code, des composants, des formations, des modèles de services et de nombreuses autres fonctionnalités nécessaires au développement d'applications Windows Azure.
- **Les services Windows Azure** : aident à la collaboration au-delà des frontières organisationnelles en maintenant la sécurité dans les domaines avec simplicité. Il fournit des fonctionnalités d'authentification et de contrôle d'accès à l'aide d'une infrastructure puissante, sécurisée et normalisée.
- **Windows Azure HPC Scheduler** : fournit des modules et des fonctionnalités permettant de lancer et de gérer des applications de calcul haute performance (HPC) dans un service Windows Azure.

1.3.7.3. SalesForce.com

Salesforce [salesforce, 2018] est un Cloud de type logiciel comme service (SaaS) qui propose un logiciel de gestion de la relation client (CRM) en tant que service. Au lieu de maintenir le matériel et les licences des logiciels, les clients utilisent le logiciel hébergé sur les serveurs

Salesforce pour un coût minime. Les clients de Salesforce utilisent le logiciel comme s'il était le leur et n'ont pas à se soucier des coûts de sa maintenance. Cela inclut la fourniture du matériel, l'installation, ainsi que tous les logiciels requis et les mises à jour de routine. Cependant, Salesforce n'est applicable qu'aux clients qui ont besoin de logiciels existants. Salesforce n'offre que des logiciels de CRM et n'autorise pas l'hébergement de services personnalisés. Bien que Salesforce est un Cloud facile à utiliser mais il reste le moins flexible [Buyya R. et Al., 2011].

1.3.7.4. EC2 d'Amazon

Amazon Elastic Compute Cloud (EC2) est un service IaaS qui fournit une capacité de calcul élastique dans le Cloud. Ce service peut être exploité via des services web (SOAP ou REST), une console de gestion web AWS (Amazon Web Service) ou les outils de ligne de commande EC2. Le service Amazon fournit des centaines d'AMI (Amazon Machine Images) prédéfinies avec divers systèmes d'exploitation (Linux, OpenSolaris ou Windows) et des logiciels pré chargés. Il offre un contrôle complet de nos ressources informatiques et nous permet une exécution facile sur l'infrastructure d'Amazon. Le service EC2 d'Amazon réduit le temps nécessaire à l'obtention et au démarrage des instances d'un nouveau serveur en minutes, permettant ainsi une capacité et des ressources évolutives et rapides [Buyya R. et Al., 2011].

1.3.8. La différence entre les grilles informatiques et le Cloud Computing

Dans cette partie, nous allons comparer les grilles et le Cloud Computing en se basant sur différents aspects qui jouent un rôle important sur la performance et la fiabilité des deux technologies.

D'abord, il faut commencer par le modèle de calcul utilisé par chaque technologie, le calcul dans une grille se fait en divisant une tâche énorme en un grand nombre de sous-tâches indépendantes et non liées pour pouvoir les exécuter par un nombre de ressources (figure 1.13). Si l'une des ressources échoue et ne renvoie pas le résultat, cela n'aura pas d'importance car l'ensemble du processus ne sera pas affecté. Tout comme les grilles, le Cloud se constitue d'un grand nombre de ressources qui sont regroupés de manière à être prêtes pour l'exécution (virtualisation). Mais les ressources fournies par le Cloud sont pour accomplir une tâche spéciale, par exemple : un utilisateur peut utiliser une ressource du groupement pour déployer son application, ce n'est pas comme dans le cas ou il soumet sa tâche à la grille et laisse cette dernière la compléter. En outre, le Cloud prend en charge de

nombreuses opérations qui ne sont pas adaptées aux grilles, telles que la gestion efficace des applications interactives sensibles à la latence.

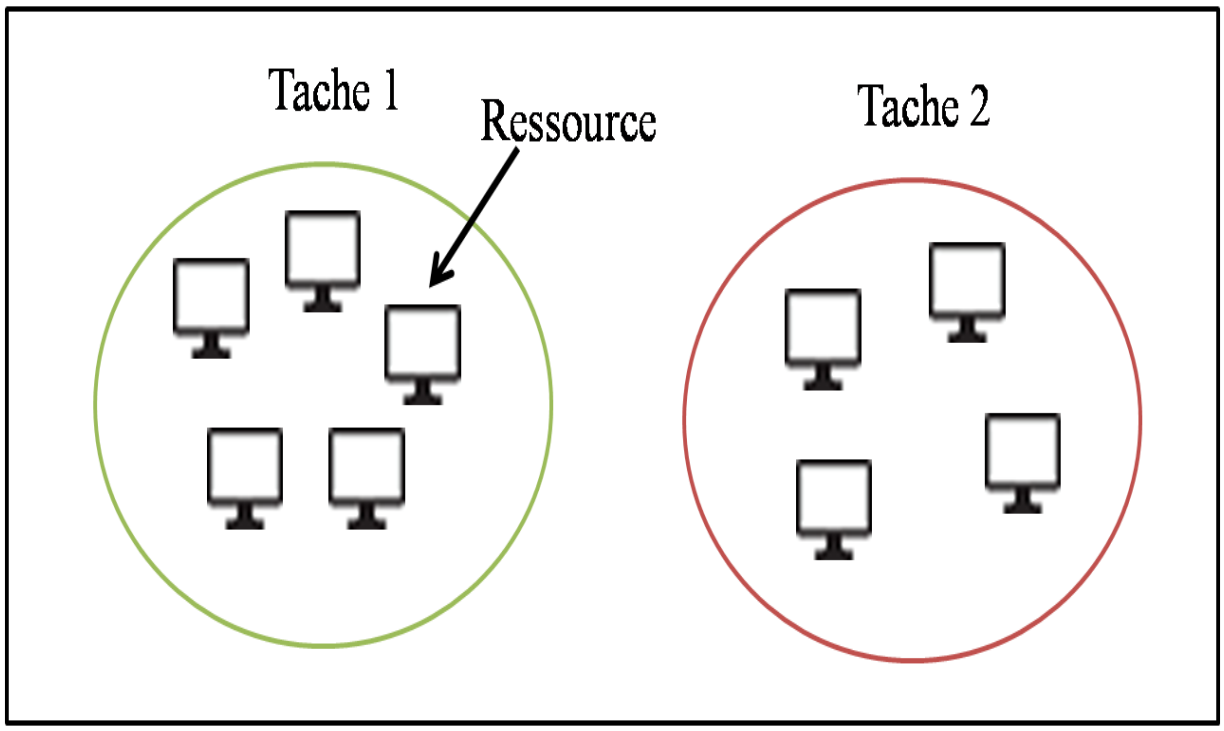

Figure 1. 13 Modèle de calcul d'une grille

En ce qui concerne le modèle économique, les modèles de logiciels traditionnels utilisaient le principe d'un paiement unique pour une utilisation illimitée du logiciel. L'avènement du Cloud Computing a donné naissance à un nouveau modèle économique, qui consiste à faire payer au client uniquement ce qu'il consomme. Par contre, le modèle économique des grilles est orienté projets où les organisations qui l'adoptent partagent les ressources de manière collaborative avec d'autres organisations faisant partie de cette grille. Les utilisateurs de la grille ont un accès direct aux ordinateurs, aux logiciels, aux données et aux autres ressources. Cet accès est possible en utilisant des stratégies conditionnelles de partage qui réalisent un processus organisationnel pour les utilisateurs et les fournisseurs de ressources [Pourqasem J et Al., 2014]. Dans le tableau 1.1 nous allons donner les différences qui existent entre les grilles et le Cloud Computing.

Paramètres	Les grilles	Le Cloud Computing
But	Partage collaboratif de ressources	Utilisation de services
Degré d'abstraction	Bas (plus de détail)	Elevé (Elimine les détails)
Degré de mise à l'échelle	Normal	Elevé
Degré de transparence	Bas	Elevé
Exécution	Pas à temps réel	Services à temps réel
Système d'exploitation	N'importe quel SE standard	Un hyperviseur (VM) sur lequel plusieurs SE s'exécutent
Objet critique	Ressource de calcul	Service
Nombre d'utilisateurs	Petit	Large
Ressources	Limitées car le matériel est limité	Illimitées (virtualisation)
Configuration	Difficile	Facile
Future	Cloud Computing	Une autre génération d'internet

Tableau 1. 1 Comparaison entre les grilles et le Cloud Computing [Pourqasem J et Al., 2014]

1.3.9. Les besoins de sécurité dans le Cloud

En plus des risques et des menaces inhérents aux technologies IT traditionnelles, le Cloud Computing présente une technologie avec son propre ensemble de problèmes de sécurité. L'un des risques les plus importants du Cloud, est que les fournisseurs doivent gérer potentiellement des millions de clients. Ce que cela représente, c'est que beaucoup de gens craignent que les fournisseurs de services du Cloud, ne soient pas en mesure de faire face à la grande échelle ou que l'infrastructure, ne puisse pas évoluer correctement avec un grand nombre d'utilisateurs. L'un des éléments qui poussent les entreprises à externaliser le stockage de leurs données, est qu'elles n'ont plus d'espace de stockage suffisant pour l'ensemble de ces données au sein de l'entreprise. Les héberger sur le Cloud est une très bonne solution mais avant d'aller vers une solution Cloud, l'entreprise veut avoir la garantie que ces données soient bien sécurisées ; c'est-à-dire elle veut assurer leur confidentialité, leur intégrité et leur disponibilité. En outre, le Cloud présente un niveau de risque supplémentaire

car les services essentiels sont souvent sous traités par une tierce partie, ce qui rend plus difficile le maintien et la gestion de la sécurité des données. Enfin, la sécurité du Cloud est l'une des issues les plus importantes qui peut empêcher son utilisation par les clients, ceci pousse les fournisseurs du Cloud à adopter les meilleures mesures de sécurité possibles pour gagner la confiance des clients. Il est toujours possible que l'infrastructure du Cloud soit sécurisée en fonction de certaines exigences et que les clients recherchent un ensemble d'exigences de sécurité différent.

1.4. Conclusion

De nos jours, la puissance de production de données de différents types (scientifique, santé, météorologique,…) augmente. Ces données-là ont besoin d'une capacité de stockage et de calcul d'où la nécessité d'une technologie répondant aux nouveaux besoins des organisations. Les grilles et le Cloud Computing sont les deux technologies qui répondent le mieux aux nouvelles exigences. Dans ce chapitre nous avons expliqué tout ce qui concerne les deux technologies. En outre, nous avons donné une brève description des besoins de sécurité dans les grilles et dans le Cloud Computing. Dans le deuxième chapitre, nous essayerons de détailler tout ce qui est en relation avec la sécurité des grilles, tandis que dans le troisième chapitre nous prenons en considération la sécurité dans le Cloud Computing.

Chapitre II : La sécurité dans les grilles informatiques (Etat de l’art)

2.1. Introduction

Le besoin d'une collaboration entre les scientifiques de différentes institutions ainsi que l'augmentation du taux de données à traiter et à stocker ont donné naissance aux technologies de grilles. Les grilles informatiques combinent des ressources hétérogènes, appartenant à plusieurs domaines administratifs géographiquement distribués, pour résoudre des problèmes de calcul énormes, stocker des données et permettre la collaboration scientifique. Bien que les grilles offrent d'énormes avantages, elles doivent faire face aux problèmes de sécurité et des implications possibles car en plus des problèmes traditionnels de sécurité, les grilles présentent des défis de sécurité plus importants en raison de leur nature hétérogène et leur distribution. Dans ce chapitre, nous nous intéressons à la sécurité des grilles. D'abord, nous allons identifier les différentes issues de sécurité citées dans la littérature. Ensuite, nous ferons des résumés des différentes solutions proposées pour résoudre ces problématiques.

2.2. Un aperçu sur la sécurité des grilles informatiques

La formation des grilles informatiques implique le partage organisé d'une variété de ressources avec des propriétaires diverses. La grille obtient sa puissance et sa fonctionnalité en utilisant les synergies résultant de la coopération, c'est-à-dire les propriétaires de ressources partageant un espace disque libre et un processeur inactif avec les utilisateurs pour résoudre des problèmes complexes que leurs propres ressources personnelles ne pouvaient gérer. Compte tenu de la distribution géographique des ressources et de la grande diversité des utilisateurs, chacun ayant des besoins et des objectifs uniques pour le système de grille, la gestion de la sécurité des utilisateurs et des ressources pose problème. Les utilisateurs d'une grille, qu'il s'agisse de calcul, de données ou d'information, peuvent avoir des intérêts contradictoires et souhaiteraient donc avoir l'assurance que leurs transactions soient à l'abri des yeux des autres utilisateurs. La grille passe également d'activités principalement gouvernementales et militaires à des initiatives plus commerciales ce qui pousse les utilisateurs à exiger un niveau de sécurité similaire à celui assuré par les autres outils commerciaux informatisés, tel que le commerce électronique. Par conséquent, sans sécurité, une configuration de grille serait vulnérable aux utilisateurs non autorisés, aux processus malveillants et à la falsification des données qui pourraient éventuellement la rendre inutilisable. Mais, la gestion d'une telle problématique n'est pas une tâche facile car la

sécurité de la grille fait face à plusieurs issues telles que: la gestion des identités des utilisateurs sur les réseaux locaux et mondiaux, la gestion de la diversité des politiques de sécurité des ressources et la sécurité des échanges entre les entités. Enfin, la conception d'une grille sécurisée implique d'une part la prise en considération des besoins des utilisateurs de la grille en offrant un accès à des ressources sécurisées, garantissant l'intégrité et la confidentialité de leurs données. D'autre part, il faut tenir compte des besoins des propriétaires de ressources en assurant que seules les personnes autorisées et fiables utilisent ces ressources.

2.3. Etat de l'art sur la sécurité des grilles

Dans cette partie, nous allons présenter et expliquer les différents travaux qui ont été proposés afin de résoudre une ou plusieurs issues de sécurité dans les grilles. Nous allons classer ces travaux selon le problème résolu et cela dans les classes suivantes : l'authentification, le contrôle d'accès (Autorisation), l'intégrité et la confidentialité. Les travaux qui prennent en considération plusieurs issues de sécurité ont été mis dans la classe : issues multiples.

2.3.1. Le contrôle d'accès

Le contrôle d'accès est l'issue de sécurité qui englobe : l'identification, l'authentification et l'autorisation [Firesmith D., 2004]. Cette dernière est souvent appelée contrôle d'accès car c'est à ce niveau que l'accès est réellement contrôlé. La première phase qui représente l'identification, consiste à l'introduction ou la présentation des informations d'identification par l'utilisateur. Les deux phases suivantes vont être expliquées en détail dans ce qui suit.

2.3.1.1. L'authentification

Est le processus qui permet de vérifier l'identité d'une entité (utilisateur, ressource ou service). Cette vérification est faite en utilisant des informations fournies par cette entité qui peuvent prendre la forme de quelque chose qu'on connait (ex : un mot de passe) ou qu'on sait faire (ex : une signature sur écran tactile), quelque chose qu'on possède (ex : une carte à puce) ou quelque chose qu'on est (ex : une empreinte digitale). Dans les grilles, le protocole d'authentification est assuré au niveau de la couche connectivité de l'architecture de la grille. Les protocoles d'authentification sont classés en trois familles selon le nombre de facteurs utilisés lors de la vérification. Si le protocole utilise un seul facteur, il s'agit d'une authentification simple. S'il utilise deux facteurs, il s'agit d'une authentification à deux

facteurs. S'il utilise plusieurs facteurs, on parlera alors d'une authentification forte (ou à plusieurs facteurs). Dans certains cas, il y'a une nécessité d'authentification des deux parties communicantes, il s'agit de l'authentification mutuelle. Enfin, il existe l'authentification unique qui permet à l'utilisateur de s'authentifier une seule et unique fois (SSO : Single Sign On).

- ***Les différents types de protocoles d'authentification***

❖ **Le protocole Single Sign On (SSO)**

Est un protocole qui permet à l'utilisateur de la grille de s'authentifier une seule fois (par exemple au démarrage des calculs). Ensuite, l'utilisateur pourra initier ses calculs, utiliser les ressources et les libérer sans authentification supplémentaire, ce qui permet aux utilisateurs de la grille d'accéder à un grand nombre de ressources distribuées géographiquement sans s'authentifier individuellement auprès de chaque ressource [Kaur J., 2013]. L'une des configurations de l'authentification unique est l'utilisation d'un mot de passe à usage unique (OTP). Ce dernier est un mot de passe dynamique qui est robuste face aux attaques par rapport aux mots de passe traditionnels (statiques). Dans ce mécanisme, chaque mot de passe est utilisé pour une session de travail, ce qui empêche sa réutilisation. Ce type d'authentification est considéré comme une authentification à deux facteurs car elle combine le facteur mot de passe (quelque chose qu'on connait) et le facteur jeton (quelque chose qu'on possède) [Kazemi A., 2014].

❖ **L'authentification utilisant une tierce partie**

Dans ce protocole d'authentification, l'utilisateur obtient un certificat digital d'une autorité de certification (CA) qui est connue comme une tierce partie. D'abord, les informations concernant l'utilisateur sont hachées puis signées par la clé privée de l'autorité de certification. Comme la clé publique du CA est largement connue, la validité du certificat ainsi que l'accès de l'utilisateur au système sera facile. Dans ce mécanisme, chaque utilisateur doit avoir une clé publique qui sera validée par l'autorité de certification. Ceci est difficile lorsqu'il s'agit d'un environnement largement distribué et avec un grand nombre d'utilisateurs tel que les grilles [Mohd Alif Hasmani A.G. et Al., 2012]. Un des exemples de l'autorité de certification est l'infrastructure à clés publiques (PKI). Cette infrastructure est basée sur les certificats standards X.509 ou bien une forme spéciale de certificat digital dérivé du X.509. Outre que l'authentification standard, l'infrastructure PKI offre aux utilisateurs un mécanisme

de délégation en utilisant un certificat proxy de type X.509. Ce dernier est un type spécial de certificat X.509 qui est signé par l'utilisateur et non par l'autorité de certification, il a une durée de vie limitée et assure une authentification unique [Mohd Alif Hasmani A.G. et Al., 2012]. Kerberos est aussi un mécanisme d'authentification par une tierce partie. Le système Kerberos doit disposer d'un centre de distribution de clés (KDC : Key Distribution Center) qui authentifie l'utilisateur à l'aide d'un mécanisme d'authentification standard tel que l'utilisation d'un mot de passe. Le KDC génère une clé de session chiffrée avec la clé publique du système qui permet l'accès de l'utilisateur [Mohd Alif Hasmani A.G. et Al., 2012].

- **L'authentification utilisant une cryptographie basée sur l'identité (IBC : Identity Based Cryptography)**

L'authentification basée sur les certificats numériques présente certaines faiblesses qui entravent l'évolutivité de la grille. Par conséquent, l'authentification sans certificat a émergé utilisant la cryptographie basée sur l'identité. Dans ce type de cryptographie, l'identifiant de l'utilisateur (par exemple : adresse IP ou adresse électronique) est utilisé comme clé publique. Par contre, la clé privée correspondante est récupérée auprès d'un tiers de confiance appelé générateur de clés privées (PKG : Private Key Generator). La clé privée est générée en utilisant l'identité de l'utilisateur et une clé secrète. En outre, la clé privée extraite doit être envoyée à l'utilisateur via un canal de communication sécurisé afin d'éviter sa divulgation [Farouk A. et Al., 2012].

- **L'authentification sans certificat**

Ce protocole utilise également un centre de génération de clé (KGC). Ce générateur n'a pas accès aux clés privées des entités contrairement au PKG de l'IBC. Le KGC fournit a une entité A une clé privée partielle D_A qu'il calcule en utilisant un identifiant de A (ID_A) et une clé maîtresse. La fourniture des clés privées partielles doit se faire d'une manière confidentielle. Ensuite, l'entité A combine sa clé partielle D_A avec quelques informations pour générer sa clé privée S_A. De cette manière, le KGC ne connait pas la clé privée de A mais l'entité A lui transmet ses informations secrètes pour qu'il calcule sa clé publique. La clé publique de l'entité A doit être transmise aux autres entités par message ou bien elle sera placée dans un annuaire public. C'est-à-dire, ce protocole n'utilise pas de certificat pour la clé publique [Al-Riyami S.S. and Paterson K.G., 2003].

❖ **Authentification utilisant une fédération d'identité**

La fédération d'identité est une technologie émergente, qui permet de transférer les informations d'identification à travers les différents domaines de la grille. En utilisant la fédération d'identité, les utilisateurs d'un domaine peuvent accéder à un autre domaine sans le besoin d'une relation de confiance directe entre l'utilisateur et les domaines accédés. C'est-à-dire, l'utilisateur peut introduire ses informations d'identification dont il a l'habitude d'utiliser au sein du domaine auquel il appartient. En outre, les domaines n'ont pas besoin de maintenir les informations d'identification des utilisateurs externes [Qiang W. and Konstantinov A., 2010].

- ***Les différents travaux qui ont été proposés pour résoudre l'issue d'authentification dans les grilles***

Dans cette partie, nous allons présenter et expliquer les différents travaux qui ont été présentés pour résoudre l'issue d'authentification dans les grilles.

❖ **La technique d'authentification proposée par Qiang W. and Konstantinov A., 2010**

Dans cet article, les auteurs ont décrit une infrastructure à authentification unique (SSO) développée comme une partie du middleware ARC (Advanced Resource Connector) de la grille NorduGrid. Cette technique utilise la fédération d'identité standard (SAML 2). Les utilisateurs s'authentifient une seule fois en utilisant leur compte habituel sans avoir besoin de maintenir des certificats X.509. Dans l'infrastructure proposée, les auteurs ont utilisé l'IdP (Identity Provider) Sibboleth [Internet2, 2018] pour accomplir les fonctionnalités du fournisseur d'identité. En outre, un navigateur web SAML 2 profile SSO, a été utilisé pour implémenter les fonctionnalités du fournisseur de service ainsi que celles de l'utilisateur. Pour l'authentification, l'utilisateur commence d'abord par accéder au fournisseur de service avec les informations du fournisseur d'identité. Dès la réception des informations sur le fournisseur d'identité, le fournisseur de service crée une demande d'authentification SAML qui permettra à l'utilisateur d'être réorienter vers l'IdP Sibooleth pour s'authentifier en utilisant son identifiant et son mot de passe. Ensuite, l'utilisateur recevra une réponse SAML incluant une confirmation SAML et des attributs. La réponse reçue sera transférée au fournisseur de service qui vérifiera et stockera la confirmation SAML. Dès que l'utilisateur reçoit la confirmation d'authentification avec succès, il pourra alors invoquer les services web. Les auteurs ont également implémenté un service web qui permet d'obtenir des certificats X.509

basés sur la fédération d'identité afin de permettre la compatibilité avec les applications grilles qui nécessitent le certificat X.509 [Qiang W. and Konstantinov A., 2010].

- **Technique d'authentification proposée par Chen M. et Al., 2010**

Dans cet article, les auteurs ont proposé un protocole d'authentification et de négociation de clé (AK) qui permet à l'utilisateur et au fournisseur de service de s'authentifier mutuellement tout en négociant la clé de session secrète. Le protocole proposé se compose de trois phases, à savoir : setup, génération de clé et négociation de clé. Les deux premières phases utilisent les six algorithmes suivants : setup, fixer la valeur secrète-publique, extraire la clé privée partielle, fixer la clé privée, fixer la clé publique et valider la clé publique. Dès la fin de la phase de génération de clé, l'utilisateur possède sa clé publique et sa clé privée et le fournisseur de ressource également. En outre, la troisième phase permet à l'utilisateur ainsi qu'au fournisseur de ressource de faire certains calculs et échanger les résultats. Après ces échanges, les deux participants vérifient la validité de la signature de la contrepartie et calculent la clé de session. Enfin, le protocole proposé a permis de contourner les problèmes de clés qui sont hérités des schémas basés sur l'identité ainsi que la charge de gestion des certificats dans les crypto-systèmes basés sur PKI [Chen M. et Al., 2010].

- **Technique d'authentification proposée par Hedayati M. et Al., 2010**

Dans cet article, les auteurs ont proposé un protocole à clé publique secrète basé sur l'identité (ID-SPK) pour sécuriser les protocoles d'authentification basés sur les mots de passe, cette nouvelle technique évite la devinette des mots de passe par un attaquant. Dans ce protocole, lors de la phase d'inscription de l'utilisateur auprès du serveur d'authentification, il doit choisir un mot de passe « *pwd* » et envoyer son image « *PW* » au serveur. De cette manière, le serveur ne connait pas le mot passe « *pwd* » mais uniquement son image. En outre, les identifiants de ce protocole sont secrets et sont connus uniquement par l'utilisateur et le serveur. Ces identifiants peuvent être générés en liant une valeur secrète au mot de passe. La clé secrète publique basée sur l'identité (PK) peut être générée par l'utilisateur ou le serveur. Enfin, après avoir présenté les protocoles à clé publique secrète basée sur l'identité pour deux parties et trois parties, les auteurs ont intégré le protocole ID-SPK proposé dans le protocole TLS [Hedayati M. et Al., 2010].

❖ **Technique d'authentification proposée par Bhowmick A. et Al., 2012**

Les auteurs de cet article ont utilisé une technique d'authentification basée sur quatre phases différentes, à savoir : la phase d'inscription de l'utilisateur, la phase de connexion de l'utilisateur, la phase d'accès et la phase de changement de mot de passe. Dans la première phase, l'utilisateur UR appartenant au domaine AD_1 doit s'inscrire auprès du serveur SR appartenant au domaine AD_2. Pour cela, UR envoie une requête d'inscription via un canal sécurisé contenant son identifiant I et son mot de passe P_1. Dès la réception de ces informations par le serveur, il calcule $K= h(P \text{ XOR } I)$ où h est une fonction de hachage à un seul sens, ensuite le serveur enregistre les éléments (h, K, P_1, I, E et D) dans une carte à puce (CS) où E est une clé de cryptage et D est une clé de décryptage générées par le serveur utilisant RSA. Cette première phase se termine par l'envoi de la carte à puce (CS) à l'utilisateur UR via un canal sécurisé. Dans la deuxième phase (connexion), l'utilisateur introduit sa carte CS auprès du serveur SR. En outre, il doit introduire son identifiant I' et son mot de passe P_1'. La carte à puce (CS) vérifie les valeurs introduites avec celles qu'elle contient. Ensuite, CS calcule $C= h(K \text{ XOR } Tu)$ où Tu est le temps d'accès de l'utilisateur. La demande de connexion contenant toutes les informations nécessaires est alors envoyée au serveur SR via un canal public. Dès la réception de cette demande, le serveur SR calcule $C^*= h(C)$, si $C= C^*$ la demande de connexion sera acceptée sinon elle sera refusée. La troisième phase qui est la phase d'accès de l'utilisateur, est exécutée pour vérifier l'authenticité de l'utilisateur UR lorsqu'il essaye d'accéder au serveur SR. Elle est exécutée à un intervalle régulier de temps. La carte CS demande à l'utilisateur d'introduire son identifiant et son mot de passe pour les vérifier avec ceux qu'elle contient. Si les informations introduites sont correctes, CS fait de nouveaux calculs et envoie une demande d'authentification avec les informations nécessaires. Dès la réception de cette demande, le serveur fait d'autres calculs et vérifie avec ce qu'il a reçu de CS. Si les résultats sont identiques, l'accès est accepté sinon il sera refusé. Enfin, la dernière phase a pour but de changer le mot de passe de l'utilisateur afin d'éviter son utilisation par un attaquant. Au cours de cette phase, un générateur aléatoire génère un nombre aléatoire de même longueur que le mot de passe P_1. Ensuite, un XOR avec l'ancien mot de passe est fait pour obtenir un nouveau mot de passe. Une fois l'utilisateur authentifié, on lui demande d'introduire le nouveau mot de passe qui remplacera l'ancien. De cette manière, l'utilisateur pourra se connecter en utilisant le nouveau mot de passe. Cette technique d'authentification est une technique à deux facteurs

(carte à puce et mot de passe) et en même à plusieurs niveaux car les informations de l'utilisateur sont vérifiées lors de la phase de connexion et celle d'accès. En outre, dans cette dernière elles seront vérifiées après chaque intervalle de temps fixe [Bhowmick A. et Al., 2012].

- **Technique d'authentification proposée par Kazemi A., 2014**

Dans cette technique, l'utilisateur doit se présenter au niveau de l'agence de la grille à laquelle il veut s'inscrire. Après avoir confirmé son identité et son authenticité, un modèle logiciel et un identifiant unique lui sont attribués. Cet identifiant a été créé en utilisant les informations personnelles de l'utilisateur (Nom, Prénom et date de naissance) et chiffré avec des algorithmes de hachage. Pour chaque connexion, l'utilisateur introduit son identifiant et reçoit un code challenge du serveur. En utilisant le modèle logiciel, la fonction de hachage ainsi que le code challenge, un mot de passe à usage unique est généré. Ce mot de passe est envoyé au serveur pour l'authentification. L'avantage de cette méthode est qu'il existe un modèle logiciel unique pour chaque utilisateur [Kazemi A., 2014].

- **Technique d'authentification proposée par Nandakumar V., 2014**

Dans cet article, l'auteur a pris en considération l'authentification mutuelle entre l'utilisateur de la grille et le fournisseur de la ressource. Pour assurer ce type d'authentification, une clé partagée (SK) de 128 bits a été utilisée. Lors de la phase d'inscription de l'utilisateur auprès de la grille, ses informations personnelles (Nom, date de naissance ainsi que l'adresse de sa machine) sont utilisées pour former les 128 bits de cette clé qui est donc divisée en 3 parties, K_{1a} K_{2b} et K_{3c}. Cette clé sera utilisée pour crypter les messages entre l'utilisateur et le fournisseur de ressources [Nandakumar V., 2014].

2.3.1.2. Le contrôle d'accès (autorisation)

Les grilles informatiques comportent un grand nombre d'utilisateurs et un grand nombre de ressources appartenant à différents domaines administratifs. Chaque utilisateur a ses propres données de calcul qui peuvent être confidentielles, d'où la nécessité d'un mécanisme de contrôle d'accès. Par définition, le rôle du contrôle d'accès consiste à contrôler et limiter les actions ou les opérations effectuées par un utilisateur sur un ensemble de ressources de la grille. En bref, il applique la politique de contrôle d'accès du système. Enfin, le contrôle d'accès dans la littérature est également appelé autorisation d'accès ou simplement autorisation [Gouglidis A. and Mavridis I., 2012].

- **Les phases d'élaboration du contrôle d'accès**

L'élaboration d'un système de contrôle d'accès s'effectue par une approche multi-phases basée sur les concepts suivants (Figure 2.1): politique de contrôle d'accès, modèle de contrôle d'accès et mécanisme de contrôle d'accès [Haddad A., 2005].

❖ **Les mécanismes de contrôle d'accès**

A un niveau élevé, les politiques de contrôle d'accès sont appliquées via un mécanisme qui traduit la demande d'accès d'un utilisateur, souvent en termes de structure fournie par un système. Il y a une grande variété de structures; par exemple : une simple recherche dans une table peut être effectuée pour accorder ou refuser l'accès. Bien qu'il n'existe pas encore de norme bien acceptée pour déterminer leur support politique, certains mécanismes de contrôle d'accès sont des implémentations directes de concepts de politique de contrôle d'accès formels [Hu V.C. et Al., 2006].

❖ **Définition d'une politique de contrôle d'accès**

Une politique de contrôle d'accès dans les grilles peut être définie comme une exigence de sécurité, qui spécifie comment et quand un utilisateur peut accéder à une ressource spécifique. Une telle politique, peut être appliquée dans un système de grille via un mécanisme de contrôle d'accès. Ce dernier, est chargé d'accorder ou de refuser l'accès d'un utilisateur à une ressource [Gouglidis A. and Mavridis I., 2012].

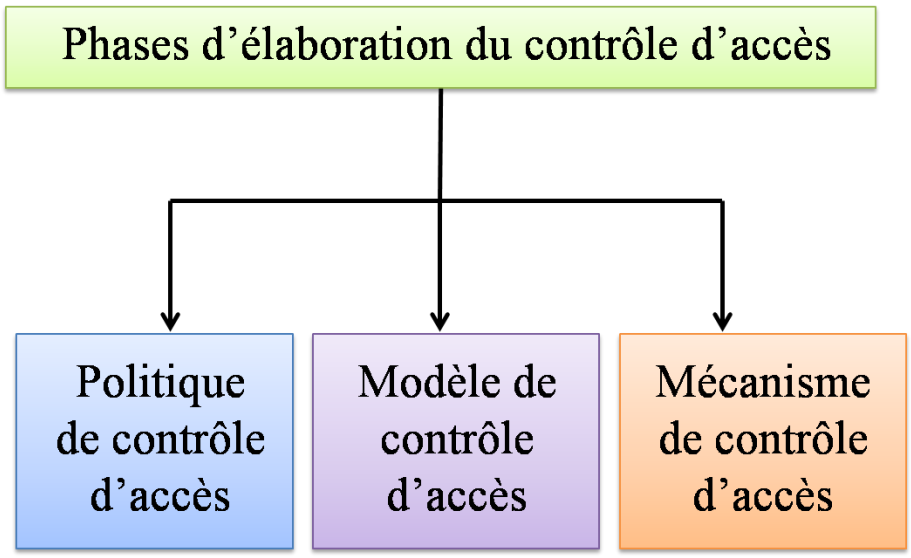

Figure 2. 1 Phases d'élaboration du contrôle d'accès

❖ **Un modèle de contrôle d'accès**

Un modèle de contrôle d'accès peut être défini comme un conteneur abstrait d'une collection d'implémentations de mécanisme de contrôle d'accès, qui est capable de préserver la prise en charge du raisonnement des politiques du système à travers un cadre conceptuel. Le modèle de contrôle d'accès comble le fossé d'abstraction existant entre le mécanisme et la politique dans un système. Dans ce qui suit nous allons voir les différents modèles de contrôle d'accès qui existent [Gouglidis A. and Mavridis I., 2012].

✓ ***Le modèle de contrôle d'accès discrétionnaire (DAC)***

Dans ce modèle de contrôle d'accès, le propriétaire de la ressource peut accorder ou refuser l'accès à cette ressource. Les politiques discrétionnaires définissent un contrôle d'accès basé sur l'identité des demandeurs et des règles d'accès explicites déterminant, qui peut ou ne peut pas exécuter une certaine action sur une ressource particulière [Bokefode Jayant. D. et Al., 2014]. Ce modèle a été proposé par Lampson [Thion R., 2007], puis il a été entièrement redéfini par Graham et Dening. Enfin, les auteurs Harrison, Ruzzo, et Ullmann ont formalisé le modèle en proposant HRU. Dans ce qui suit, nous allons expliquer ces différents modèles (figure 2.2).

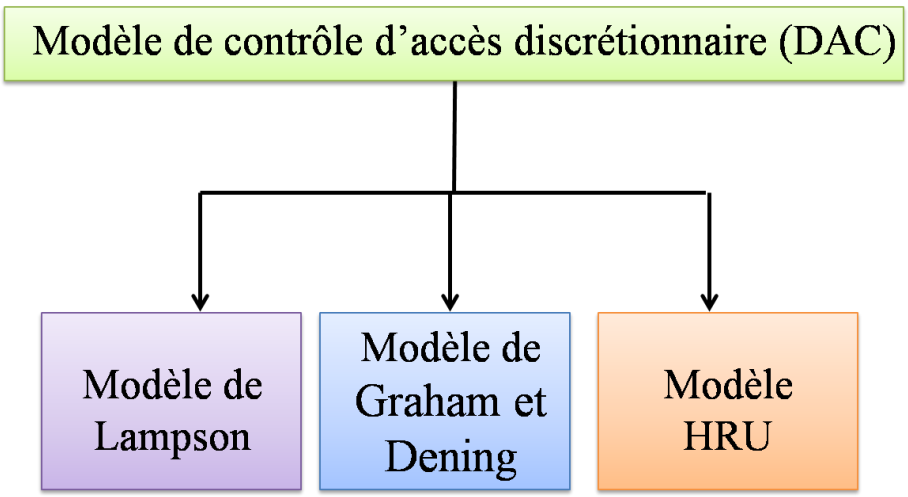

Figure 2. 2 Les différents exemples du modèle DAC

a. Le modèle Lampson

En 1974, Lampson a introduit les notions formelles : sujet, objet et matrice de contrôle d'accès. L'auteur a défini cette matrice comme une représentation simple dans laquelle chaque entrée (i, j) spécifie les opérations accordées au sujet i sur la ressource j [Thion R.,

2007]. Un exemple est illustré dans le tableau 2.1 où l'utilisateur Karim peut écrire les enregistrements administratifs et lire les prescriptions.

E : écrire, L : lire

	Enregistrement médical	Enregistrement administratif	Prescriptions
Imene	E, L	E	L
Karim		E	L
Mohamed		L	
Nadine	E, L	L	E, L

Tableau 2. 1 Exemple d'une matrice de contrôle d'accès

D'un point de vue ligne, cette matrice peut être interprétée comme une liste de capacités définissant ce qui est permis pour chaque utilisateur. D'un point de vue colonne, cette matrice peut être interprétée comme une liste de contrôle d'accès définissant les permissions qui sont accordées sur chaque objet [Thion R., 2007].

b. Le modèle de Graham et Dening

Les créateurs de ce modèle, ont amélioré le modèle Lampson en ajoutant huit commandes permettant de mettre à jour la matrice de contrôle d'accès. Ces matrices prennent en considération la création et la destruction des objets par les sujets ainsi que la transmission des autorisations entre les sujets. En outre, chaque objet possède un propriétaire qui a des droits spéciaux et chaque sujet possède un autre sujet qui le contrôle (a des droits spéciaux sur lui). Par exemple : une action A[S, O] contient les droits que le sujet S a sur l'objet O tels que : propriétaire, exécuter. Lors de l'exécution de l'une des huit commandes, par exemple la création d'un objet, la matrice de contrôle d'accès est modifiée : une nouvelle colonne est ajoutée pour cet objet et le sujet qui l'a créé devient son propriétaire. En outre, chaque règle est associée à une condition préalable, par exemple : si le sujet X veut supprimer l'objet O il doit être son propriétaire (A[X, O] contient le droit « propriétaire ») [Revolvy, 2018].

c. Le modèle HRU

Ce modèle est une amélioration du modèle Lampson qui a été présenté en 1976 par Harisson M.A, Ruzzo W.L. et Ullman J.D d'où vient son nom HRU. Ce modèle utilise également une matrice de contrôle d'accès classique en précisant les commandes qui peuvent lui être

appliquées. Ces dernières permettent d'assigner des droits d'accès (lire, écrire, posséder, …) ainsi que créer et supprimer des sujets et des objets. Dans ce modèle, si le droit « propriétaire » est associé à une paire (S, O), le sujet S sera considéré comme le propriétaire de l'objet O et pourra céder ses droits d'accès sur l'objet O à d'autres sujets. En d'autres termes, ce droit permet au sujet de définir les permissions sur la colonne entière. Les opérations primitives possibles sont : entrer (ajout des droits), supprimer (suppression de droit), créer un sujet, créer un objet, détruire un sujet et détruire un objet. Les commandes de ce modèle sont créées à partir des opérations primitives citées au dessus et peuvent avoir comme arguments les sujets et les objets. Ces commandes sont également divisées en deux parties, la première conditionnelle et la deuxième opérationnelle [Ennahbaoui M. and Elhajji S., 2013].

- ✓ ***Le modèle de contrôle d'accès obligatoire (MAC)***

Est un modèle de contrôle d'accès qui ne permet pas aux propriétaires de ressources d'accorder ou de refuser l'accès à ces ressources. Dans ce modèle, la politique de sécurité du système impose que les décisions de protection ne soient pas prises par les propriétaires de ressources. Ce modèle n'est pas convenable pour un environnement tel que les grilles car certaines ressources peuvent avoir leurs propres politiques de sécurité [TechTarget, 2018]. Le contrôle d'accès mandataire peut être divisé en deux types, à savoir : les modèles multi-niveaux et les modèles multilatéraux. Nous allons voir dans ce qui suit la définition de ces modèles et quelques exemples de chaque type (Figure 2.3).

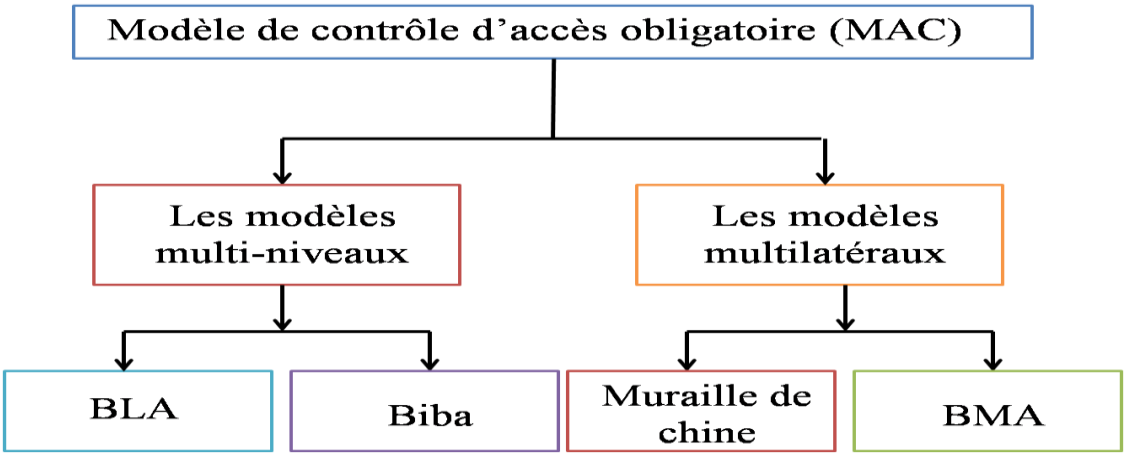

Figure 2. 3 Les différents exemples du modèle MAC

a. Les modèles multi-niveaux

Dans ce type, les sujets sont attribués à différentes classes selon la confiance qu'on peut leur accorder et les objets sont classés en différents niveaux selon leur degré de sensibilité (figure 2.4) : confidentiel, secret, top secret ou ouvert. Ce type de modèle utilise les deux notions importantes suivantes [Bokefode Jayant. D. et Al., 2014] :

- **Niveau de transparence** : indique le taux de confiance qu'on peut accorder à un utilisateur [Bokefode Jayant. D. et Al., 2014].
- **Niveau de classification** : indique le niveau de sensibilité attribuée à une ressource ou une donnée [Bokefode Jayant. D. et Al., 2014].

Figure 2. 4 Les différents niveaux de sécurité des objets [Anderson R.J., 2008]

Il existe plusieurs modèles multi-niveaux, dans ce qui suit nous allons voir le modèle Bell Lapadula et le modèle de Biba.

i. Modèle Bell Lapadula

Ce modèle a été proposé par David Bell et Len Lapadula en 1973 pour officialiser la politique de sécurité multi-niveaux du département de la défense des Etats-Unis [Sumtsova I. et Al., 2018]. Dans les installations gouvernementales et militaires, une classification des objets est faite selon leur niveau de sensibilité (confidentiel, secret, top secret ou ouvert). Les utilisateurs sont aussi classés selon le degré de confiance qu'on pourra leur accorder, ce degré est nommé « Niveau de transparence » (Security clearance). Lorsqu'un utilisateur essaye d'accéder à un fichier ou une ressource, une vérification de son degré de transparence est faite pour savoir si cet utilisateur peut accéder à une ressource avec un tel niveau de classification.

ii. Modèle Biba

Le modèle d'intégrité de Biba a été publié en 1977 à la Mitre Corporation. Le modèle Bell-LaPadula garantit la confidentialité des données mais pas leur intégrité. En conséquence, Biba a créé un modèle pour répondre au besoin de renforcer l'intégrité dans un système informatique. Ce modèle a proposé un ensemble de politiques d'intégrité pouvant être utilisées. Chacune des politiques utilise des conditions différentes pour assurer l'intégrité de l'information [Anderson R.J., 2008]. Les sujets sont classés en différentes classes d'intégrité qui sont totalement ordonnées, par exemple : Crucial (C), Important (I) et Non classifié (N). Ce niveau d'intégrité est attribué au sujet selon la confiance qu'on peut lui attribuer concernant la modification, l'ajout ou la suppression des données. Par contre, Le niveau de classification d'intégrité des objets permet de représenter le danger que peut constituer la modification de l'information contenue dans ce dernier. Le contrôle d'accès est imposé par deux principes [Haddad A., 2005]:

- Pas de lecture vers le bas (No read down) : un sujet peut lire un objet si la classe d'accès de l'objet domine celle du sujet.
- Pas d'écriture vers le haut (No write up) : un sujet peut écrire dans un objet si la classe d'accès du sujet domine celle de l'objet.

b. Les modèles multilatéraux

Dans ce type de modèle, l'objectif n'était pas d'empêcher la circulation des informations dans une hiérarchie mais plutôt d'empêcher leur circulation entre les départements (Figure 2.5). Ces frontières latérales peuvent être organisationnelles comme dans le cas d'une organisation de renseignement qui veut garder secret les noms de ses agents travaillant à l'étranger. Ces informations doivent rester secrètes par rapport au département responsable de l'espionnage du pays dans lequel ils sont infiltrés. Ou comme en médecine, lorsque l'accès à un dossier médical est limité à un service hospitalier particulier [Anderson R.J., 2008].

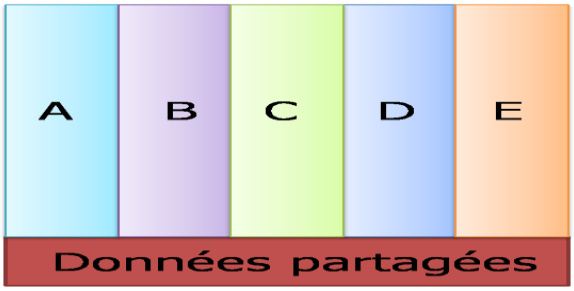

Figure 2. 5 Sécurité multilatérale [Anderson R.J., 2008]

i. Le modèle Muraille de Chine

Ce modèle a été développé par Brewer et Nash. Son nom vient du fait que les sociétés de services financiers (les banques d'investissement) ont des règles internes conçues pour prévenir les conflits d'intérêts, qu'elles appellent « Chinese Walls ». La portée du modèle est plus large que la banque d'investissement. De nombreuses entreprises professionnelles et de services ont des clients qui peuvent être en concurrence les uns avec les autres: les fournisseurs de logiciels, les agences de publicité et les comptables sont d'autres exemples. Une règle typique est que «un partenaire qui a travaillé récemment pour une entreprise dans un secteur d'activité peut ne pas avoir accès aux documents de toute autre entreprise dans ce secteur.» Donc, un rédacteur publicitaire qui a travaillé sur, disons, le compte Shell, ne sera pas autorisé à travailler sur le compte d'une autre compagnie pétrolière pendant une période déterminée [Anderson R.J., 2008].

ii. Le modèle BMA

Ce modèle a été proposé par l'association médicale britannique (BMA) en 1995. Le but principal de ce modèle était de garantir la confidentialité des données médicales. En outre, sa politique repose sur neuf principes [Biba M., 2017], à savoir :

- **Le contrôle d'accès :** chaque dossier clinique identifiable doit avoir une liste de contrôle d'accès nommant les personnes ou les groupes de personnes pouvant le lire et y ajouter des données. Le système doit empêcher toute personne ne figurant pas dans la liste d'accéder à l'enregistrement.
- **L'ouverture de l'enregistrement :** un clinicien (docteur, pharmacien, chirurgien dentiste, infirmier, …) peut ouvrir l'enregistrement seul ou avec le patient et les autres cliniciens figurant dans la liste de contrôle d'accès.
- **Le contrôle :** un des cliniciens sur la liste de contrôle d'accès doit être marqué 'Responsable'. Ce dernier peut changer cette liste en ajoutant uniquement d'autres cliniciens.
- **Le consentement et la notification :** le clinicien responsable doit informer le patient des noms figurant sur la liste de contrôle d'accès de son dossier lors de son ouverture, de tous les ajouts ultérieurs et de la transmission de responsabilité. Son consentement doit également être obtenu, sauf en cas d'urgence.
- **La persistance :** personne ne peut supprimer un enregistrement médical jusqu'à l'expiration du temps nécessaire.

- **L'attribution :** tous les accès aux enregistrements cliniques doivent être marqués sur l'enregistrement avec le nom du sujet, la date et l'heure.
- **Le flow d'information :** les informations provenant de l'enregistrement A peuvent être ajoutées à l'enregistrement B si et seulement si la liste de contrôle d'accès de B est contenue dans celle de A.
- **Le contrôle d'agrégation :** des mesures efficaces doivent être prises pour empêcher l'agrégation des informations personnelles de santé. En particulier, les patients doivent recevoir une notification spéciale si on propose d'ajouter une personne à sa liste de contrôle d'accès qui a déjà accès à des informations personnelles de santé sur un grand nombre de personnes.
- **Un système informatique de confiance :** les systèmes qui traitent les informations de santé doivent comporter un sous système qui applique les principes cités au dessus de manière efficace.

Les modèles MAC et DAC n'ont pas été utilisé dans les grilles informatiques. Un environnement de grilles est généralement composé d'un grand nombre de ressources et d'utilisateurs appartenant à différents domaines administratifs. L'utilisation d'un modèle basé sur l'identité n'est pas appropriée pour un environnement hautement distribué et dynamique tel que les grilles. Donc ces deux modèles sont efficaces pour des systèmes distribués fermés et relativement immuables qui ne prennent en charge qu'un ensemble d'utilisateurs connus qui accèdent à un ensemble de services connus.

✓ ***Contrôle d'accès basé sur les rôles (RBAC)***

Le contrôle d'accès à base de rôle (RBAC : Role Based Access Control) est un modèle qui attribue un ou plusieurs rôles à chaque utilisateur du système et en même temps, il attribue des permissions à ces rôles (Figure 2.6). Par exemple : soit l'utilisateur U_1 qui possède le rôle « *directeur d'hôpital* », les permissions qui sont attribuées à ce rôle sont : (gérer, soigner, acheter). De plus, gérer l'accès à un système utilisant ce modèle nécessite le respect de certaines politiques de contrôles d'accès qui sont faites à base de rôle, par exemple : Pour qu'un utilisateur puisse accéder aux informations médicales d'un patient de l'hôpital, il doit avoir le rôle « *médecin* ». De cette manière l'utilisation du rôle offre une indépendance logique dans la spécification des autorisations d'utilisateur. En outre, cette indépendance simplifie la gestion de la politique de sécurité car lorsqu'un nouvel utilisateur rejoint

l'organisation, l'administrateur du système doit attribuer des rôles particuliers selon les responsabilités du poste. Si les responsabilités d'un utilisateur sont modifiées, l'administrateur devra modifier les rôles associés à cet utilisateur. Lorsqu'une nouvelle tâche ou un nouveau programme est ajouté au système de sécurité, l'administrateur doit décider quels rôles sont fournis pour l'exécuter. L'un des points positifs du modèle RBAC, est le principe d'hiérarchie de rôles qu'il fournit avec l'héritage de permissions. Ces deux principes simplifient l'administration en diminuant le nombre d'attributions « utilisateur-rôle » et « rôle-permission ». Enfin, l'un des points négatifs du modèle RBAC est dû à l'attribution statique des rôles, ce qui peut être considérer comme un problème dans un environnement distribué et dynamique car son implémentation sera difficile.

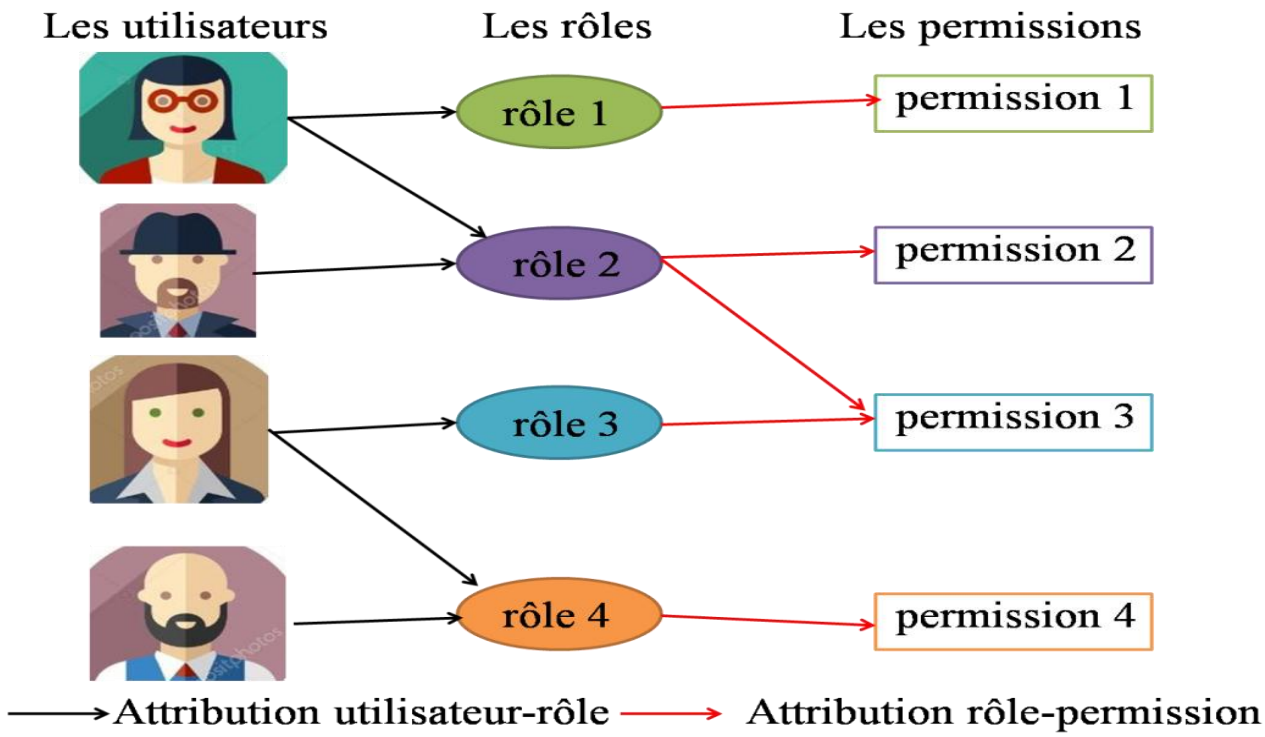

Figure 2. 6 Principe du modèle RBAC

✓ ***Contrôle d'accès basé sur les attributs (ABAC)***

Le modèle de contrôle d'accès basé sur les attributs (ABAC : Attribute Based Acces Control) est un modèle logique qui permet de contrôler l'accès aux objets en comparant les valeurs des attributs (Sujet, Objet, Opération, Environnement) d'une requête avec celles qui sont exigées par la politique de sécurité. ABAC offre un contrôle d'accès précis, qui permet un plus grand nombre d'entrées variables, fournissant un plus grand ensemble de combinaisons possibles de

ces variables pour refléter un ensemble plus large et plus définitif de règles possibles [Hu C.V. et Al., 2014]. Les listes de contrôle d'accès (ACL) ainsi que RBAC sont à certains égards des cas particuliers d'ABAC en termes d'attributs utilisés. Les ACL fonctionnent sur l'attribut « identité ». RBAC travaille sur l'attribut "rôle". La différence clé avec ABAC est le concept de politiques qui expriment un ensemble de règles booléennes complexes qui permettent l'évaluation de plusieurs attributs en même temps. Bien qu'il soit possible d'atteindre les objectifs d'ABAC en utilisant des ACL ou RBAC, mais en déployant ces deux modèles la réponse aux exigences du contrôle d'accès est difficile et restera coûteuse. Un autre problème avec les modèles ACL ou RBAC est que si la politique de sécurité est modifiée, il peut être difficile d'identifier tous les endroits où l'implémentation ACL ou RBAC doit être mise à jour [Hu C.V. et Al., 2014]. En général, ABAC évite que des permissions soient directement attribuées aux demandeurs ou à leurs rôles avant que la demande ne soit faite. Au lieu de cela, lorsqu'un sujet demande l'accès, le moteur ABAC peut prendre une décision en se basant sur les attributs du demandeur, les attributs de l'objet, les conditions d'environnement et un ensemble de politiques spécifiées en fonction de ces attributs. De cette manière, les politiques peuvent être créées et gérées sans référence directe à de nombreux utilisateurs et objets potentiellement nombreux. D'autre part, les utilisateurs ainsi que les objets peuvent être provisionnés sans référence à la politique [Hu C.V. et Al., 2014].

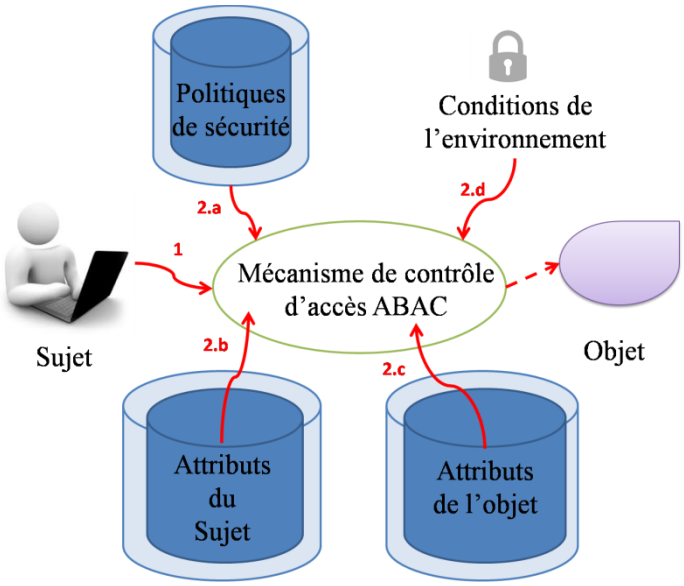

Figure 2. 7 Principe de fonctionnement du modèle ABAC [Hu C.V. et Al., 2014]

Le principe de fonctionnement du modèle ABAC est simple, comme l'illustre la figure 2.7, lorsqu'un utilisateur demande l'accès à un objet (1), le mécanisme de contrôle d'accès évalue les politiques de sécurité (2.a), pour cette évaluation il aura besoin des attributs du sujets (2.b), des attributs de l'objet (2.c) ainsi que des conditions de l'environnement (2.d). Si les valeurs de ces attributs coïncident avec celles des attributs de la politique, l'accès sera accepté sinon refusé (ligne rouge discontinue).

- ✓ ***Contrôle d'accès basé sur les organisations (OrBAC)***

Le modèle OrBAC est une extension du modèle RBAC qui détaille les permissions tout en restant indépendant de l'implémentation. Son objectif principal était d'exprimer les politiques de sécurité uniquement avec des entités abstraites en séparerant complètement la représentation de la politique de sécurité de son implémentation. Le modèle OrBAC est basé sur les rôles, les vues, les activités et un contexte. L'entité rôle est utilisée pour créer le lien entre le sujet et l'organisation. En outre, les objets qui satisferont une propriété commune sont spécifiés via les vues. Par contre, les activités sont utilisées pour abstraire les actions et le contexte peut être temporel, spatial ou bien déclaré par l'utilisateur. Les règles de sécurité de ce modèle sont de la forme : *Permission (org, r, v, a, c)* qui veut dire dans le contexte c, l'organisation *org* accorde au rôle *r* la permission d'effectuer l'activité *a* sur la vue *v* [Abou El Kalam A. et Al., 2003]. Par exemple, la politique de sécurité de l'hôpital Ibn Sina contient la règle de sécurité suivante : Persmission (Ibn Sina, médecin, consulter, dossier médical, urgence) qui signifie que l'hôpital Ibn Sina accorde aux médecins la permission de consulter n'importe quel dossier médical dans le contexte de l'urgence. Dans le modèle OrBAC, une organisation peut être divisée en sous-organisations, un rôle peut être divisé en sous-rôles, l'activité en sous-activités et la vue en sous-vues. Ces décompositions génèrent des relations hiérarchiques, OrBAC introduit les prédicats sous-rôle (org, R_1, R_2) qui signifie que dans l'organisation org, le rôle R_1 est un sous- rôle du rôle R_2 et même chose avec sous-activité (org, A_1, A_2) et sous-vue (org, V_1, V_2) [Kassid A. and El Kamoun N., 2016]. Une fois la politique de sécurité spécifiée au niveau organisationnel, il est possible de l'instancier en affectant des entités concrètes à des entités abstraites. Pour ce faire, trois prédicats ternaires ont été définis pour affecter un sujet à un rôle, une action à une activité et un objet à une vue [Orbac, 2016]:

- *empower(org, subject, role)* : Spécifie que dans l'organisation *org*, le sujet *subject* est affecté au rôle *role*.
- *consider(org, action, activity)*: spécifie que dans l'organisation *org*, l'action *action* implémente l'activité *activity*.
- *use(org, object, view)*: spécifie que dans l'organisation *org*, l'objet *objet* est utilisé dans la vue *view*.

- ***Les outils d'implémentation du contrôle d'accès***

Le processus du contrôle d'accès peut être divisé en deux sous-ensembles, le premier prend en charge la création des politiques de sécurité du système, c'est-à-dire la spécification ou l'écriture de ces politiques. Le second, prend en considération le processus du contrôle d'accès lui-même (réponse à une requête d'accès). En d'autres termes, quelles sont les entités qui vont réaliser ce processus ? Quels sont les échanges possibles entre ces entités et sous quelle forme? Pour mettre en œuvre le processus de contrôle d'accès, un outil d'implémentation est nécessaire mais avant de le choisir, il faut se focaliser sur quel sous-ensemble du processus va être pris en charge par cet outil et quel est le modèle de contrôle d'accès utilisé. Il existe plusieurs outils d'implémentation du processus de contrôle d'accès, certains sont spécifiques à la spécification des politiques de sécurité, d'autres prennent en considération le processus de contrôle d'accès et la spécification des politiques de sécurité en même temps. Dans ce qui suit, nous allons voir quelques outils d'implémentation en spécifiant leurs rôles et le modèle de contrôle d'accès qui peut être pris en charge par cet outil.

- **L'outil XACML**

XACML (eXtensible Access Control Markup Language) [OASIS, 2015] est une norme OASIS [OASIS, 2016][c] qui décrit à la fois un langage d'écriture de politique de sécurité et un langage de demande / réponse de décision de contrôle d'accès (tous deux écrits en XML). Son premier rôle lui permet de décrire les exigences générales de contrôle d'accès. Par contre, le langage requête / réponse permet de demander si une action donnée est autorisée, et interpréter le résultat. La décision comprend toujours une réponse indiquant si la requête a été autorisée en utilisant l'une des quatre valeurs suivantes: Permis, Refusé, Indéterminé (une erreur est survenue ou une valeur requise est manquante, une décision ne peut être prise) ou Non Applicable (la demande ne peut pas avoir de réponse par ce service) [OASIS, 2016]. L'outil XACML fait appel à plusieurs composants où chacun a un rôle bien déterminé. Dans ce qui suit, nous allons voir les composants les plus importants de cet outil ainsi que les

éléments nécessaires pour écrire les différentes politiques de sécurité. Enfin, nous présenterons une explication de son fonctionnement d'une manière très simple.

✓ ***Les composants de l'outil XACML***

Comme l'illustre la figure 2.9, l'outil XACML se compose des éléments suivants [OASIS, 2015] :

- **PEP** (Policy Enforcement Point) : est le point d'application de la politique, c'est ce composant qui se chargera de créer les requêtes d'accès et de les envoyer au PDP. En outre, dès la réception d'une réponse du PDP, le PEP la transmettra à l'utilisateur.
- **PDP** (Policy Decision Point) : est le point qui se chargera de prendre la décision d'accès en comparant les valeurs des attributs de la requête d'accès avec celles des attributs de la politique de sécurité.
- **PIP** (Policy Information Point) : est le point d'information de la politique, ce point se chargera de récupérer les attributs manquants concernant les ressources, les utilisateurs et l'environnement afin que le PDP puisse évaluer la politique de sécurité.
- **La base des politiques de sécurité :** cette base contient toutes les politiques du système rédigées utilisant le langage XACML qui est basé sur XML.

✓ ***Les composants du langage XACML***

La première phase qui précède le contrôle d'accès est celle où l'administrateur (A) du système soumet les politiques de sécurité. Ces politiques seront rédigées en utilisant le langage XACML (premier rôle cité précédemment). Le langage XACML définit trois niveaux d'éléments de la politique de sécurité (figure 2.8): l'élément **<Rule>** qui représente la règle de sécurité, l'élément <**Policy**> qui représente la politique de sécurité et l'élément <**PolicySet**> qui représente un ensemble de politiques [OASIS, 2015].

- **L'élément <Rule>** : contient une expression booléenne qui ne pourra pas être évaluée individuellement, cet élément existe de manière isolée uniquement dans la base des politiques où il pourra former l'unité de gestion de base et être utilisé dans plusieurs politiques. Il se compose des éléments suivants :
  - **L'élément <target>** : cet élément définit les ressources, les sujets et les actions auxquels la règle de sécurité est applicable.
  - **L'élément <effect> :** représente l'effet de la règle qui peut être refusé (Deny) ou accepté (permit).

- **L'élément <condition >**: qui indique les conditions sous lesquelles cette règle est satisfaite.

- **L'élément <Policy>** : contient un ensemble d'éléments **<Rule>,** une procédure spécifique pour combiner les résultats de leur évaluation (algorithmes de combinaison) ainsi que l'élément **<target>** qui spécifie quand est ce que cette politique est applicable. Cet élément représente l'unité de base qui sera utilisée par le PDP pour la prise de décision.
- **L'élément <PolicySet> :** contient un ensemble d'éléments **<Policy>** ou d'autres éléments **<PolicySet>,** une procédure spécifique pour combiner les résultats de leur évaluation (algorithmes de combinaison) ainsi que l'élément **<target>** qui spécifie quand est ce que cet ensemble de politiques est applicable. Cet élément représente le moyen standard pour combiner des politiques distinctes en une seule politique combinée.
- **Les algorithmes de combinaison :** Le langage XACML définit un ensemble d'algorithmes de combinaison pouvant être identifiés par un attribut **<RuleCombiningAlgId>** ou **<PolicyCombiningAlgId>** dans les deux éléments **<Policy>** et **<PolicySet>**. L'algorithme de combinaison des règles de sécurité, définit une procédure pour arriver à une décision d'autorisation en prenant en considération les résultats d'évaluation individuelle d'un ensemble de règles. De même, l'algorithme de combinaison des politiques de sécurité, définit une procédure pour arriver à une décision d'autorisation en prenant en considération les résultats d'évaluation individuelle d'un ensemble de politiques. Les différents algorithmes de combinaison utilisés dans XACML sont :
  - **Deny-overrides :** cet algorithme prend le résultat d'évaluation finale refusé (Deny) s'il existe au moins un seul refus dans les résultats d'évaluation individuelle des règles ou des politiques.
  - **Permit-overrides :** cet algorithme prend le résultat d'évaluation finale permis (Permit) s'il existe au moins une seule permission dans les résultats d'évaluation individuelle des règles ou des politiques.
  - **First-applicable :** le résultat d'évaluation utilisant cet algorithme est le même résultat obtenu lors de l'évaluation de la première règle ou politique.

- **Only-One-Applicable** : cet algorithme est applicable uniquement au niveau des politiques. Le résultat de cet algorithme assure qu'uniquement une et une seule politique est applicable.

En plus des éléments cités au dessus, les trois éléments **<Rule>**, **<Policy>**, **<PolicySet>** peuvent contenir les deux éléments facultatifs **<obligation>** et **<advice>**. Une obligation facultativement spécifiée dans une règle, une politique ou PolicySet est une directive du PDP au PEP sur ce qui doit être effectué avant ou après l'approbation ou le refus d'une demande d'accès. Le conseil est similaire à une obligation, sauf que le PEP peut ignorer ce conseil [Ferraiolo D. et Al., 2016].

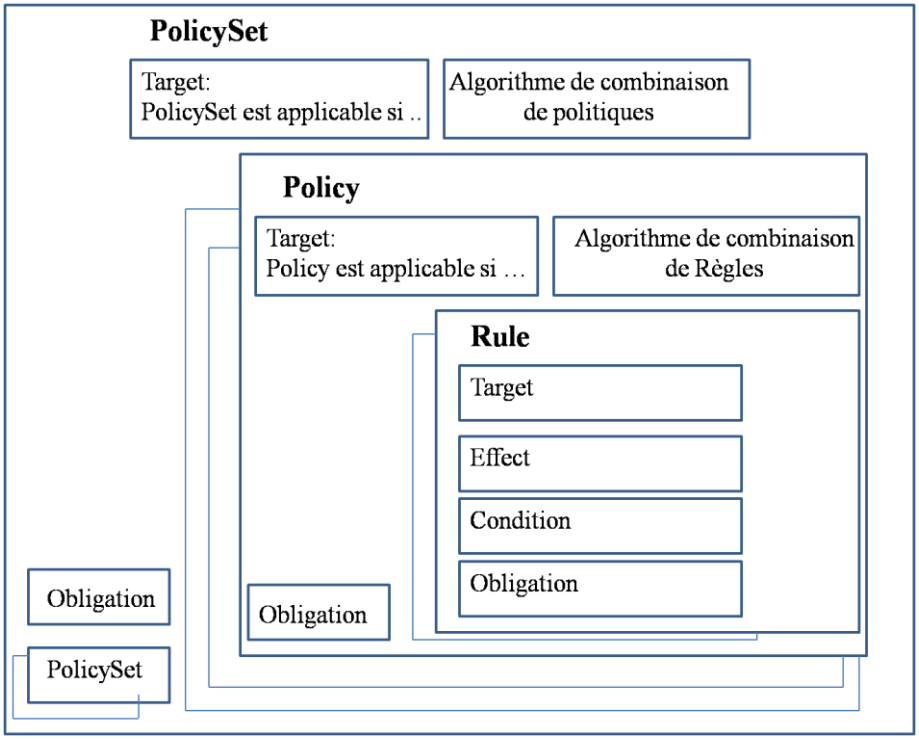

Figure 2. 8 Eléments du langage XACML [Ferraiolo D. et Al., 2016]

✓ ***Fonctionnement de l'outil XACML***

Après la soumission des politiques de sécurité rédigées en XACML par l'administrateur du système (flèche discontinue dans la figure 2.9), à chaque nouvelle demande de contrôle d'accès reçue par le PEP (1). Une requête XACML (Figure 2.9) est créée et envoyée au PDP (2). Dès la réception de cette requête, le PDP demande au PIP les valeurs manquantes des attributs (3) tels que les attributs concernant l'utilisateur (3.a), les attributs concernant les ressources (3.b) ou ceux concernant l'environnement (3.c). Après la réception de ces attributs

(4) le PDP évalue la requête en comparant les valeurs des attributs contenues dans la requête avec celles des attributs de la politique de sécurité applicable qui est stockée dans la base des politiques de sécurité (5). Ensuite, le PDP envoie une réponse XACML contenant le résultat de l'évaluation (6). Dès la réception de ce résultat par le PEP, il transmet la réponse à l'utilisateur (7).

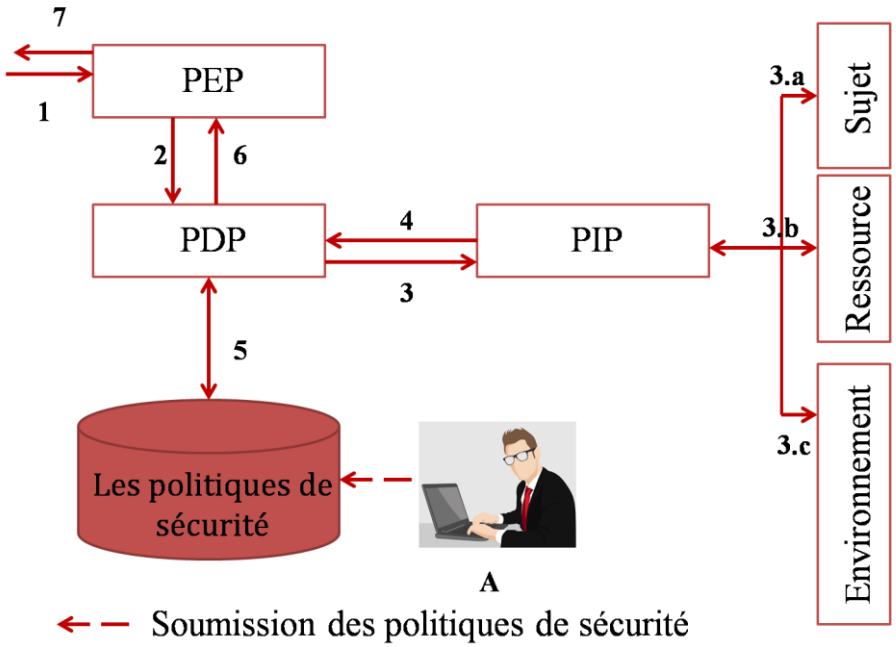

Figure 2. 9 Les composants de l'outil XACML

❖ **L'outil MotOrBAC**

L'outil MotOrBAC est un outil permettant la spécification et l'administration des politiques de sécurité basées sur le modèle OrBAC [MotOrBAC, 2018]. Cet outil est composé des quatre modules suivants [Cuppens C. et Al., 2006] :

- ✓ **L'interface graphique** : à travers cette interface, l'administrateur peut saisir les politiques de sécurité propres au système en utilisant les différentes entités (organisation, sous-organisation, rôle, activité, vue et contexte).
- ✓ **Le module de communication** : ce module reçoit les politiques saisies, déclenche l'analyse de cohérence et stocke les politiques de sécurité dans la base des politiques.
- ✓ **Module d'analyse de la cohérence** : étant donné que le modèle OrBAC, permet l'expression de politiques mixtes contenant à la fois des permissions et des interdictions, il y'aura surement des conflits de politiques. Ce module se charge

d'analyser la cohérence entre les politiques existantes et les politiques à ajouter. MotOrBAC intègre également des stratégies de résolution de conflits qu'il applique aux politiques de sécurité introduite par l'administrateur pour rétablir la cohérence.

- ✓ **La base de données** : contient toutes les politiques.

Il existe deux versions de MotOrBAC , la première version a été développée en utilisant java et prolog alors que la nouvelle version est écrite en pur Java. La nouvelle version est également plus modulaire car elle utilise l'API OrBAC [Orbac, 2017]. MotOrBAC vise à donner à l'utilisateur la possibilité de spécifier toutes ses exigences de sécurité indépendamment de son application. Pour ce faire, MotOrBAC implémente le modèle OrBAC qui spécifie les exigences de sécurité au niveau organisationnel. Chaque composant de sécurité peut être représenté comme une sous-organisation de l'organisation représentant le système d'information qui gère la sous-partie de la politique globale associée à ce composant. Une fois la spécification de la politique au niveau organisationnel est terminée, des entités concrètes correspondant aux utilisateurs, aux actions et aux objets du système d'information peuvent être introduites. De cette façon, le concepteur de politique peut simuler la politique de sécurité en vérifiant les règles de sécurité concrètes inférées par MotOrBAC [Autrel F. et Al., 2008].

- ***Les différents travaux qui ont été proposés pour résoudre l'issue du contrôle d'accès dans les grilles***

Dans cette partie, nous allons présenter et expliquer les différents travaux qui ont été présentés pour assurer le contrôle d'accès dans les grilles.

❖ **Modèle de contrôle d'accès proposé par Zhu X.J. et Al., 2010**

Dans cet article, les auteurs ont proposé un modèle d'autorisation dynamique basé sur un mécanisme de confiance dans un environnement de grille. Le système proposé se compose de (figure 2.10) :

- ✓ **Un centre d'autorisation :** il se compose des modules suivants:
  - **Un module d'autorisation**: module central, qui accorde l'autorisation aux utilisateurs en fonction des politiques d'accès locales, du rôle de l'utilisateur et de l'application d'une tâche spécifique. En appelant le module de condition, il garantit que l'autorisation est donnée si les conditions d'exécution des tâches sont vérifiées.

- **Le module de condition:** pour examiner les informations sur l'environnement, les informations sur l'utilisateur et les attributs du service objectif afin de s'assurer que les exigences pour l'exécution des tâches sont satisfaites.
- **Le module de surveillance:** pour effectuer une autorisation dynamique. Pendant le processus, la tâche en cours d'exécution surveille le changement des attributs de l'utilisateur, les attributs du service objectif et ceux de l'environnement. Une fois qu'une modification soit découverte, le module d'autorisation doit l'autoriser à nouveau. Lorsque l'autorisation mise à jour ne remplit pas les conditions requises pour l'exécution de la tâche, le module de surveillance notifiera au service objectif d'interrompre temporairement le service et demandera à l'utilisateur de s'ajuster en conséquence.

✓ **Centre de confiance :** il comporte les modules suivants :

- **Module de calcul du degré de confiance**: se charge d'évaluer dynamiquement la fiabilité de l'utilisateur et du service objectif en utilisant le modèle de confiance proposé.
- **Module de confiance** : selon le degré de confiance calculé de l'utilisateur et du service objectif, ce module décide d'accorder ou non l'autorisation en prenant en considération le seuil d'échange.
- **Module de collection des informations de retour** : ce module se charge de collecter et stocker les informations de retour, en utilisant les deux tables : QoSTT (Service Quality Trust Table) qui concerne la qualité du service, la violation du contrat, etc et la table TQTT (Trade Quality Trust Table) qui implique l'équilibre entre l'efficacité, le mode de paiement, la violation du contrat, etc.

Le client envoie une demande à un service objectif. Le service d'objectif transfère l'application au centre de confiance, pour lui demander de déterminer la confiance et accorder l'autorisation appropriée. Après confirmation de l'identité de l'utilisateur, le module déterminant la confiance donnera une valeur de confiance primaire en interrogeant la base de données de confiance ou en faisant appel au module de calcul de confiance, puis déterminera si l'utilisateur a atteint le seuil requis pour le rôle en termes de politiques de confiance locales.

Si le seuil est atteint, l'application d'autorisation sera envoyée au centre d'autorisation. Sinon, le service objectif sera informé pour refuser l'application de service. Dans le centre d'autorisation, le module de condition détermine d'abord si les attributs de l'environnement d'exécution, ceux de l'utilisateur et ceux du service objectif répondent aux exigences pour l'exécution de la tâche. Lorsque les exigences sont satisfaites, le module d'autorisation prend la décision en fonction du rôle de l'utilisateur, des politiques d'accès et de l'application spécifique de cette tâche.

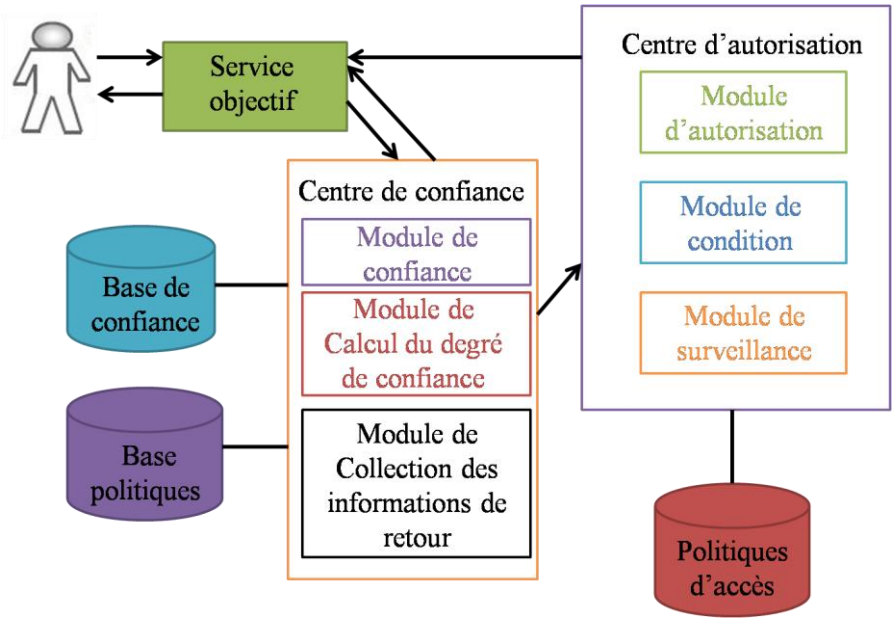

Figure 2. 10 Modèle d'autorisation dynamique basé sur un mécanisme de confiance dans un environnement de grille [Zhu X.J. et Al., 2010]

Le centre d'autorisation génère l'affirmation d'attribut SAML qui inclut les informations d'autorisation et l'envoie au service objectif. Le service objectif exécute l'application de service. Au cours du processus d'exécution, le module de surveillance du centre d'autorisation surveille l'utilisateur, le service en cours de traitement et l'environnement d'exécution. Une fois qu'un changement est découvert, le module d'autorisation sera réutilisé pour générer une nouvelle autorisation. S'il s'avère que la tâche ne peut pas se dérouler comme prévu, l'utilisateur et le service objectif seront avertis de procéder à un ajustement en conséquence. Une fois l'exécution du service demandé est terminée, les résultats seront envoyés au client. Les deux parties prenantes devront remplir les deux tables QoSTT et TQTT comme référence de l'évaluation commerciale. Le centre de confiance mettra à jour la confiance des deux parties prenantes [Zhu X.J. et Al., 2010].

❖ **Modèle de contrôle d'accès proposé par Kaiiali M. et Al., 2010**

Dans cet article, les auteurs se sont intéressés à la manière dont sont stockées les politiques de sécurité dans un environnement de grilles. Ils ont fait un résumé de tous les mécanismes existants pour le stockage des politiques de sécurité. D'abord, ils ont commencé par définir l'élément « table de sécurité » qui représente la base de ces mécanismes. Cette table a été définie comme une matrice n×m (n ressources et m règles de sécurité). Chaque entrée (i,j) dans la table de sécurité est égale à 0 ou 1 indiquant si la $j^{ième}$ règle de sécurité appartient à la politique de sécurité de la $i^{ème}$ ressource. Un exemple de cette table de sécurité est illustré dans le tableau 2.2 où Badji-Mokhtar, professeur, étudiant, Master 1 sont des règles de sécurité et $R_1 \ldots R_{12}$ sont des ressources. Comme un exemple, un utilisateur peut accéder à la ressource R_{11} s'il est étudiant en Master 1 à l'université Badji Mokhtar [Kaiiali M. et Al., 2010].

R_{id}	Badji-Mokhtar	Professeur	Etudiant	Master I
R_1	1	0	0	0
R_2	1	0	0	0
R_3	1	1	0	0
R_4	1	1	0	0
R_5	1	0	1	0
R_6	1	0	1	0
R_7	1	0	1	0
R_8	1	0	1	0
R_9	1	0	1	0
R_{10}	1	0	1	1
R_{11}	1	0	1	1
R_{12}	1	0	1	1

Tableau 2. 2 Exemple d'une Table de sécurité [Kaiiali M. et Al., 2010]

Ensuite, les auteurs ont présenté le premier mécanisme qui a été utilisé pour représenter les politiques de sécurité. Cette méthode est appelée approche à force brute (BFA : Brute Force Approach), la figure 2.11 représente une illustration de la politique de sécurité du tableau 2.2 avec ce mécanisme [Kaiiali M. et Al., 2010]. Ce dernier mécanisme nécessite la vérification de toutes les politiques de sécurité et toutes les règles afin de trouver le groupe de ressources autorisées, c'est ce qui donne des répétitions énormes. Les auteurs ont amélioré cette méthode

en proposant le mécanisme de regroupement primitif (PCM : Primitive Clustering Mechanism) qui consiste à regrouper les ressources qui ont des politiques de sécurité identiques (Un exemple de PCM sur les politiques de sécurité du tableau 2.2 est illustré dans la figure 2.12). Les auteurs ont remarqué que PCM élimine la redondance lors de la vérification des politiques de sécurité identiques mais ne peut pas supprimer la redondance de la vérification des règles de sécurité identiques. Dans la figure 2.12, la redondance dans la vérification des politiques de sécurité identiques a été éliminée en regroupant les politiques des ressources R_5, R_6, R_7, R_8 et R_9 dans un seul nœud parent. Par contre la règle de sécurité Badji-Mokhtar a besoin d'être vérifiée 4 fois. Afin de réduire cette redondance, les auteurs ont proposé un mécanisme de regroupement hiérarchique (HCM : Hierarchical Clustering Mechanism).

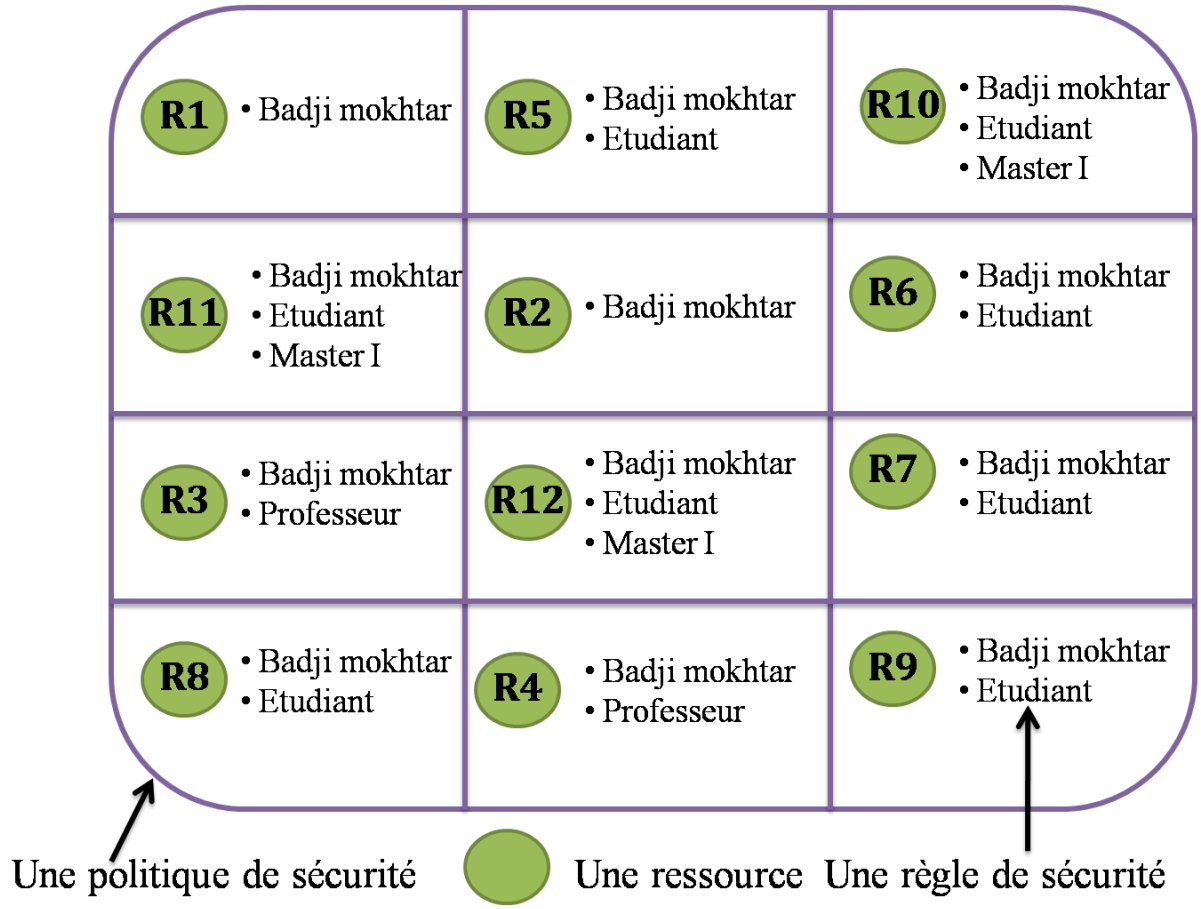

Figure 2. 11 Représentation des politiques de sécurité du tableau 2.2 par le mécanisme BFA

Ce mécanisme considère les informations des nœuds parents de l'arbre PCM comme des données pour générer un regroupement hiérarchique de ces nœuds selon leurs règles de

sécurité partagées (La figure 2.13 illustre l'arbre de décision HCM des politiques de sécurité du tableau 2.2). Pour construire cet arbre, les auteurs ont proposé un algorithme de comptage qui permet de le faire d'une manière efficace [Kaiiali M. et Al., 2010].

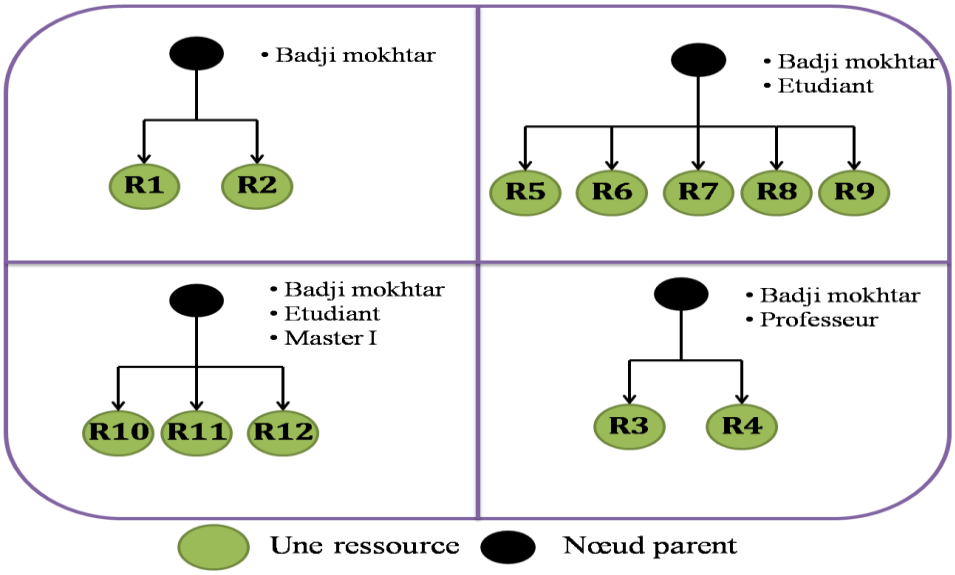

Figure 2. 12 Un exemple du mécanisme PCM [Kaiiali M. et Al., 2010]

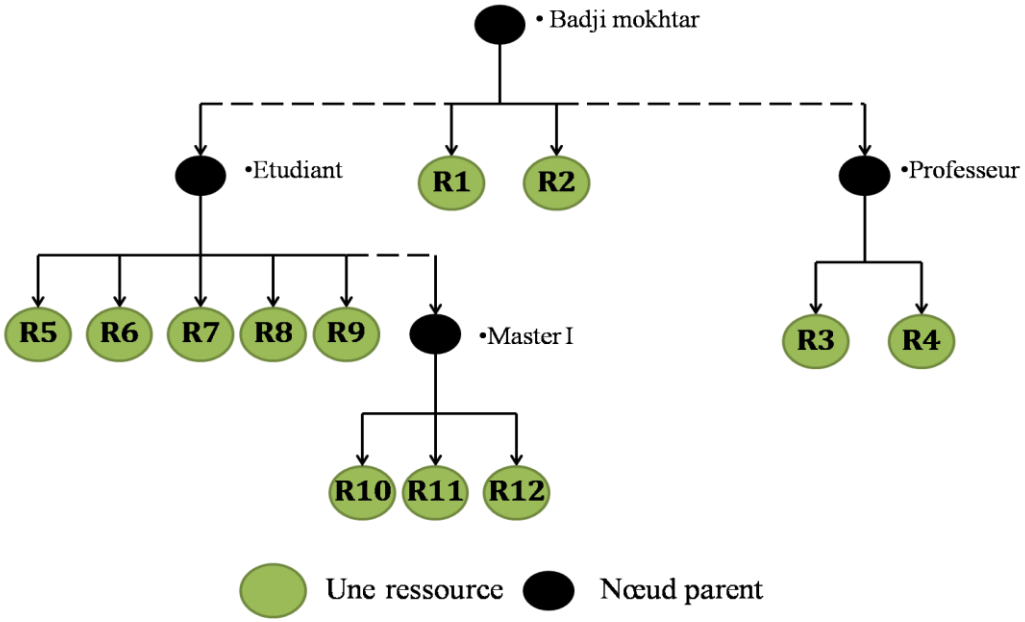

Figure 2. 13 Un exemple du mécanisme HCM [Kaiiali M. et Al., 2010]

❖ **Modèle de contrôle d'accès proposé par Zhao T. and Shoubin D., 2010**

Dans cet article, les auteurs ont proposé un modèle de contrôle d'accès basé sur ABAC en étendant l'architecture XACML avec un engin de confiance. Comme le montre la figure 2.14, l'architecture proposée se compose de :

- ✓ **Point d'administration de la politique (PAP) :** crée et gère les politiques de sécurité selon le besoin du système.
- ✓ **Point d'application de la politique (PEP) :** effectue le contrôle d'accès en créant les requêtes de contrôle d'accès et en appliquant les décisions de contrôle d'accès. Il intercepte les demandes d'accès de l'utilisateur et les envoie au PDP.
- ✓ **Point d'information de la politique (PIP):** crée et gère les attributs des sujets, des ressources, de l'environnement et ceux de la confiance.
- ✓ **Point de décision de la politique (PDP) :** prend la décision d'accès en prenant en considération les politiques de sécurité soumises par le PAP.
- ✓ **Engin de confiance (TE : Trust Engine) :** est déployé comme une source des attributs de confiance. Cet engin permet de gérer la confiance locale du domaine. Il comporte les 4 modules suivants (figure 2.15) :
  - **Module de confiance directe (DTM : Direct Trust Module)** : qui calcule la confiance directe.
  - **Module de confiance recommandée (RTM : Recommended Trust Module)** : est utilisé pour calculer la confiance recommandée
  - **Module de la confiance compréhensive (CTM : Comprehensive Trust Module)** : est désigné pour calculer la confiance compréhensive en utilisant la confiance directe et la confiance recommandée.
  - **Un agent de mise à jour de la confiance (TMA : Trust Management Agent) :** est responsable de la gestion de l'attribut de confiance.

Lorsque le PEP reçoit une demande d'accès, il envoie cette demande au PDP. Le PDP demande les attributs (concernant le sujet, la ressource, l'environnement et la confiance) au PIP. Le PIP à son tour, demande les attributs de confiance au TE. Ce dernier, vérifie si la valeur de confiance a été mise à jour. Si oui, l'agent de gestion de la valeur de confiance lit cette valeur de la base de confiance et la renvoie au PIP. Si la valeur n'a pas été mise à jour, alors il fait appel à DTM, RTM et CTM pour la calculer, puis il retourne la valeur au PIP et la

stocke dans la base de confiance. Le PIP retourne les attributs demandés au PDP. Le PDP prend une décision d'accès selon les politiques introduites par le PAP et renvoie la réponse au PEP. Enfin, le PEP accompli les conditions, si la décision d'accès est « accepté », le PEP permet l'accès à la ressource sinon l'accès sera refusé [Zhao T. and Shoubin D., 2010].

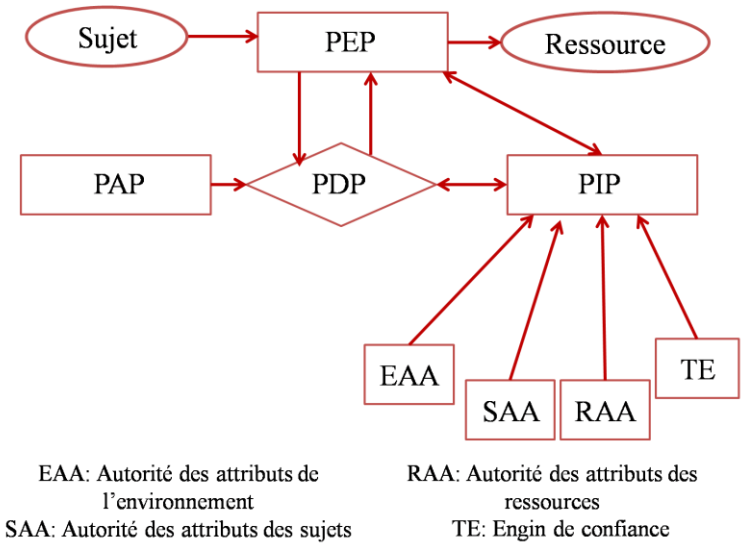

Figure 2. 14 L'architecture étendue XACML proposée [Zhao T. and Shoubin D., 2010]

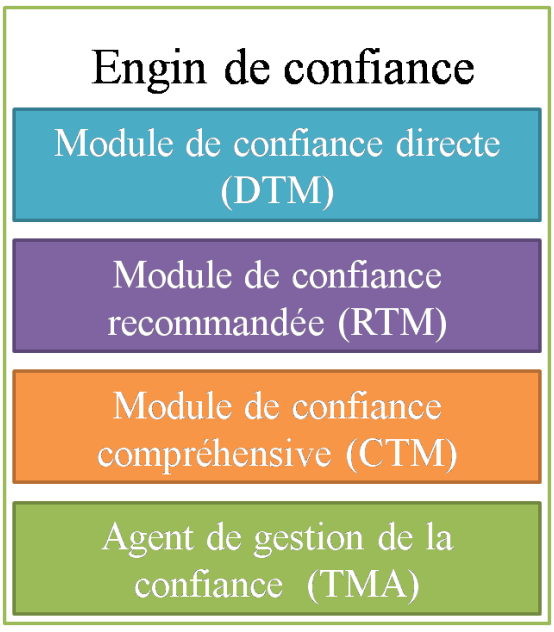

Figure 2. 15 Engin de confiance [Zhao T. and Shoubin D., 2010]

❖ **Modèle de contrôle d'accès proposé par Gupta B. et Al., 2011**

Dans cet article, une architecture de grilles multi-agents est proposée où chaque entité (fournisseur/ ressource) présente dans la grille a son propre agent. Lorsqu'un utilisateur X veut exécuter son job sur la grille, il soumet le job avec un ensemble d'exigences (telles que l'échéance, le coût, la réputation minimale du fournisseur de ressources, etc.) à son agent. Ce dernier utilise ensuite l'expérience de ses pairs de confiance auprès de divers fournisseurs de ressources pour sélectionner un ensemble de fournisseurs dignes de confiance. Une fois que l'agent utilisateur reçoit les recommandations de ses connaissances fiables, il les agrège pour identifier les fournisseurs fiables. L'agent utilisateur envoie ensuite une demande d'accès aux ressources de certains fournisseurs réputés. Lorsque cette demande atteint les fournisseurs de ressources sélectionnés, c'est maintenant au tour des agents des fournisseurs de décider si l'accès à la ressource doit être accordé ou non en calculant la fiabilité de l'utilisateur. Ceci est nécessaire pour vérifier si l'utilisateur est un utilisateur malveillant ou non. Pour calculer cette fiabilité, chaque agent du fournisseur de ressources sélectionné envoie un questionnaire à l'agent utilisateur pour obtenir ses attributs. Sur ces attributs explicatifs de l'utilisateur, la régression logistique est appliquée pour calculer sa fiabilité. Dans le modèle proposé, les utilisateurs sélectionnent les fournisseurs de ressources en prenant des recommandations de leurs agents fiables et les fournisseurs de services utilisent la régression logistique sur leurs propres données d'utilisateurs déjà desservis pour prédire la fiabilité des utilisateurs. En effet, les utilisateurs ne sont pas concurrents et peuvent agir en tant que communauté et s'entraider pour trouver de bons fournisseurs de services, mais les fournisseurs de services sont concurrents et ne s'entraident pas pour identifier les utilisateurs malveillants et donc les fournisseurs de ressources dépendent uniquement de leurs propres données [Gupta B. et Al., 2011].

❖ **Modèle de contrôle d'accès proposé par Kaustav, R. and Avijit, B., 2012**

Dans cet article, les auteurs ont proposé une politique d'autorisation inter-domaines dans un environnement de grille qui est basée sur le modèle RBAC. L'architecture proposée se compose des éléments suivants :

- Deux domaines administratifs A et B
- Le domaine A se compose de deux sous-domaines A-U et A-R où A-U représente le sous-domaine des utilisateurs et A-R représente le sous-domaine des ressources.

- Le domaine B se compose de deux sous-domaines B-U et B-R où B-U représente le sous-domaine des utilisateurs et B-R représente le sous-domaine des ressources.
- Dans le domaine A, il existe un serveur d'autorisation des utilisateurs (UAS1) et un serveur d'autorisation des ressources (RAS1).
- Dans le domaine B, il existe un serveur d'autorisation des utilisateurs (UAS2) et un serveur d'autorisation des ressources (RAS2).
- Un serveur d'évaluation 1 qui consiste à évaluer les sous-domaines du domaine A.
- Un serveur d'évaluation 2 qui consiste à évaluer les sous-domaines du domaine B.
- Un serveur d'évaluation global qui gère les redirections.

Ce modèle utilise le concept de classement des rôles, un rôle local d'un nœud est converti à un classement de rôle global afin que l'autorisation puisse être effectuée de manière transparente entre plusieurs domaines ou organisations virtuelles. Trois paramètres ont été choisis pour attribuer une valeur de rôle à un nœud utilisateur: le calcul, le stockage et le transfert de données. En combinant les valeurs de ces trois paramètres, huit valeurs différentes peuvent être générées, 000 ... 111 en binaire et converties en décimal, ainsi 7 désigne un nœud qui peut effectuer toutes les fonctions. Les notations sont données sur une échelle de 10. Les sous-domaines reçoivent également un classement des rôles basé sur l'importance et la hiérarchie sur une échelle de 10. Les nœuds de ressources ont été classés en trois catégories: système clusters, ordinateurs centraux et dispositifs de stockage dédiés ayant des rôles de 10,9 et 8 respectivement. Dès qu'une ressource reçoit une requête d'accès, elle demande au serveur d'autorisation local si le demandeur de la ressource a le droit de l'utiliser. Le serveur d'autorisation locale la redirige ainsi vers le serveur de classement global pour récupérer les informations d'identification de l'utilisateur. Le serveur global transmet la requête au serveur d'autorisation du domaine dans lequel réside l'utilisateur. Puis, le serveur d'autorisation crée un jeton et envoie sa réponse par le même chemin dans le sens inverse. A chaque étape, la notation de rôle des domaines parents est pondérée à la notation globale de l'utilisateur. Après avoir obtenu le jeton final, le rang de l'utilisateur est normalisé sur une échelle de 1. Une valeur d'interaction (IV) est également contenue dans le jeton qui a une valeur 1 s'il y avait une interaction précédente entre les deux ou 0 en cas de non interaction. Le rôle minimum pour accéder à une ressource est égal au rôle du domaine dans lequel réside l'utilisateur [Kaustav, R. and Avijit, B., 2012].

❖ **Modèle de contrôle d'accès proposé par Kaiiali M. et Al., 2013**

Dans cet article, les auteurs ont analysé le mécanisme de représentation de politique de sécurité (HCM) présenté dans [Kaiiali M. et Al., 2010], ils ont trouvé que ce mécanisme ne peut pas représenter les politiques de sécurité basées sur le « Ou » (OR-based security policies). En outre, malgré que HCM réduise la redondance dans la vérification des règles de sécurité par rapport à BFA et PCM, mais il ne peut pas l'éliminer totalement. Pour faire face aux points négatifs d'HCM, les auteurs ont proposé un graphe d'autorisation dans les grilles (GAG : Grid Authorization Graph). GAG a introduit un nouveau type d'arc nommé « Arc de correspondance » qui peut être utilisé pour éliminer totalement la redondance dans la vérification des règles de sécurité.

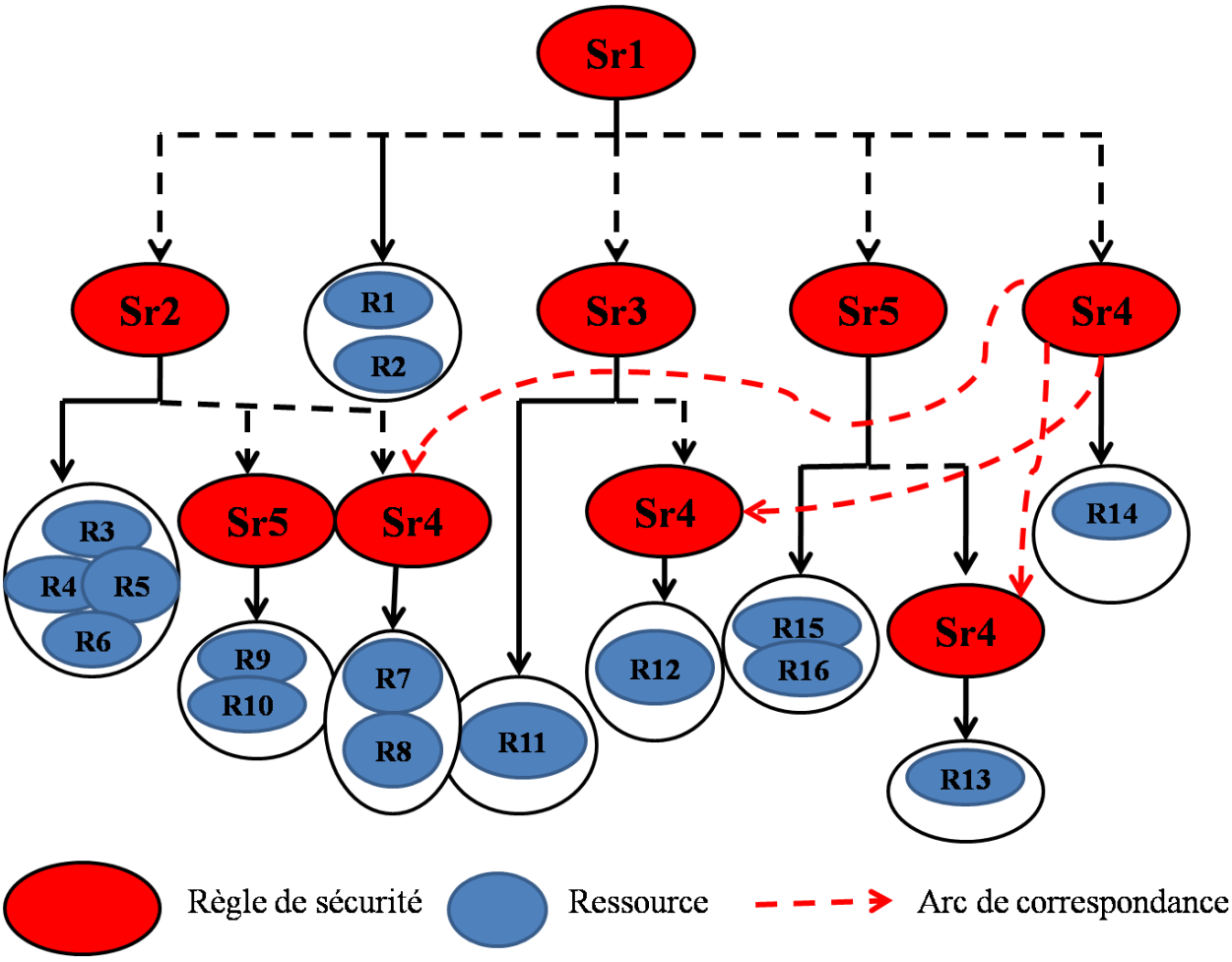

Figure 2. 16 Un exemple de représentation utilisant GAG [Kaiiali M. et Al., 2013]

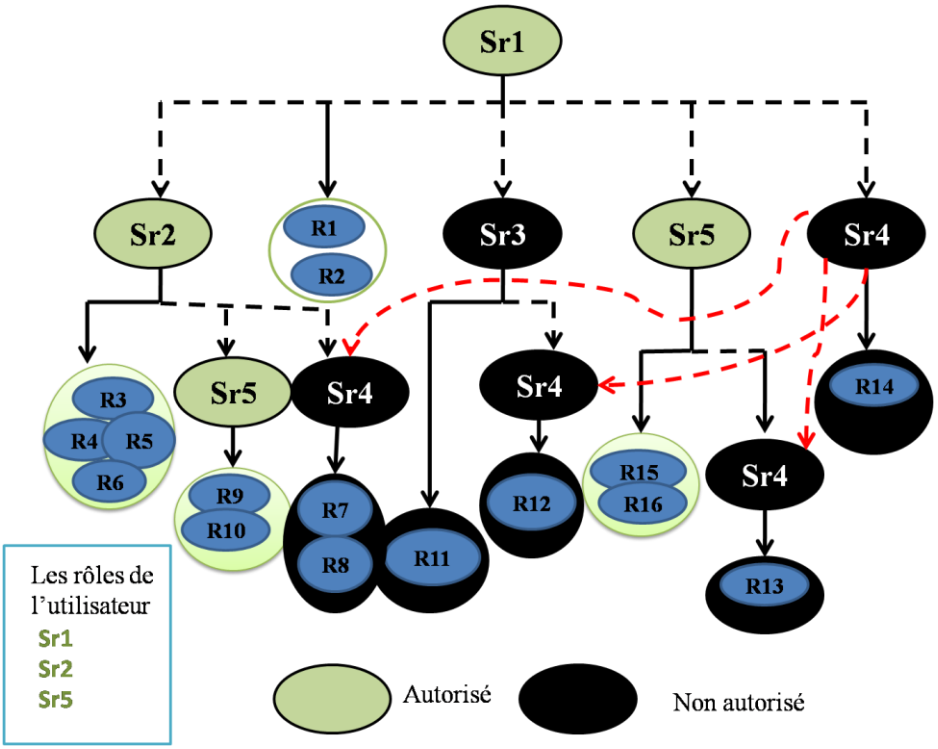

Figure 2. 17 Exemple de vérification des règles de sécurité utilisant GAG [Kaiiali M. et Al., 2013]

La figure 2.16 représente l'arc ajouté en ligne rouge discontinue, c'est un arc dessiné entre les nœuds redondants dans le graphe et il sera utilisé comme suit :

Dès qu'une règle de sécurité est vérifiée pour la première fois, un parcours en largeur (BFS : Breadth First Search) d'un seul niveau sur ses bords de correspondance consiste à marquer tous les nœuds redondants de cette règle avec le résultat de sa première vérification. Lorsqu'un processus d'autorisation arrive à un nœud redondant, il le trouve déjà marqué et n'aura pas besoin de le vérifier encore une fois (Figure 2.17) [Kaiiali M. et Al., 2013].

2.3.2. L'intégrité

Généralement, l'intégrité concerne la protection d'un contenu vis-à-vis des modifications non autorisées et intentionnelles. Elle peut être classée en trois groupes : l'intégrité des données, l'intégrité du matériel et l'intégrité des logiciels. Les mécanismes utilisés pour assurer l'intégrité peuvent être divisés en deux catégories, la première représente les mécanismes préventifs tels que le contrôle d'accès. Ces mécanismes évitent les modifications non autorisées. La seconde catégorie concerne les mécanismes défectifs, qui consistent à détecter les modifications après l'échec du premier type de mécanismes. Lorsque l'on considère le

contexte de la grille, les données peuvent se référer à des données résultant d'expériences ou simulations. Ces données sont généralement organisées dans des bases de données accessibles aux utilisateurs de grilles. En outre, ces données peuvent appartenir à des utilisateurs de grille connus ou anonymes. Dans les deux cas, ces utilisateurs veulent avoir l'assurance que les données consultées n'ont pas été altérées par des mains non autorisées. Donc, l'intégrité des données dans un environnement tel que les grilles, est le type le plus considéré. Pour assurer l'intégrité des données, dans [Tu M. et Al., 2010], les auteurs ont utilisé des schémas de partitionnement avec une réplication dynamique. L'objectif principal de cet article était de développer des algorithmes de placement pour allouer des réplicas partagés de sorte que les coûts de communication et la latence d'accès soient minimisés. Les auteurs ont d'abord introduit un algorithme heuristique pour déterminer les clusters qui devraient héberger les parties de données. Ensuite, un algorithme heuristique pour l'allocation de partages dans un cluster est présenté.

2.3.3. La confidentialité

Les mesures prises pour assurer la confidentialité visent à empêcher que les informations sensibles atteignent les mauvaises personnes, tout en s'assurant que les bonnes personnes puissent effectivement les obtenir. L'accès doit être limité aux personnes autorisées à consulter les données en question. Il est également fréquent que les données soient catégorisées en fonction de la quantité et du type de dommages qui pourraient être causés s'ils tombaient entre des mains non intentionnelles. Des mesures plus ou moins rigoureuses peuvent alors être mises en œuvre selon ces catégories. L'authentification ainsi que le contrôle d'accès garantissent généralement la confidentialité, car si on assure que la personne qui prétend être X, est bien X et que X peut accéder uniquement aux données et ressources dont il a le droit, donc on est sûr d'assurer la confidentialité de ces données. Des méthodes cryptographiques sont également utilisées pour assurer la confidentialité. Dans [Cebuc E. et Al., 2010], les auteurs ont présenté un système qui implémente quelques algorithmes cryptographiques connus (algorithmes symétriques, algorithmes asymétriques, fonctions de hachage, générateurs de nombres aléatoires et testes statistiques pour générateurs de nombres aléatoires) sur une infrastructure de grille utilisant le middleware gLite. Le système présenté offre une vaste bibliothèque d'algorithmes cryptographiques (plus de 10 algorithmes dans chaque catégorie). D'abord, le système expose un nouveau format de fichier (le format de fichier .job) qui correspond directement à la taxonomie considérée. La structure de ce type de

fichier a été définie comme un fichier XML avec des tags structurés sous la forme d'un arbre. Pour l'analyse des performances, le système communique avec le système gLite via l'outil « generatejdls » qui mappe les fichiers .job aux fichiers .jdl. Une caractéristique importante de la mise en œuvre est qu'elle offre le support nécessaire au parallélisme automatique de tout type d'application. En d'autres termes, les applications existantes peuvent être adaptées pour fonctionner sur une plateforme grille en créant simplement un fichier .job qui les inclut. Enfin, les auteurs ont mené une étude de performance en développant un ensemble d'expériences appropriées qui a révélé une diminution du temps d'exécution lorsque le degré de parallélisme des applications testées augmente.

2.3.4. Issues multiples

Cette classe comporte les travaux qui prennent en considération plusieurs issues de sécurité, les travaux qui n'ont pas pu être attribués à l'une des classes citées précédemment sont expliqués et détaillés dans cette partie.

2.3.4.1. Travail proposé par Sudalai Muthu T. et Al., 2010

Dans cet article, les auteurs ont proposé un protocole de sécurité qui assure la confidentialité, l'intégrité et le contrôle d'accès dans la grille de données « Desktop ». Les auteurs ont utilisé la cryptographie symétrique dans laquelle, avant de transférer un ficher vers les clients de stockage volontaires (VSC), le serveur du projet crypte le fichier de données avec une clé symétrique maitresse composée du hachage du fichier ajouté à un nonce. Avec cette technique, la confidentialité des données a été assurée car les VSC ignorent la clé de cryptage. En outre, l'utilisation d'une fonction de hachage ainsi que la fragmentation des données ont permis de garantir l'intégrité. Enfin, pour assurer le contrôle d'accès, les auteurs ont proposé l'utilisation de la cryptographie à clé publique pour crypter la clé symétrique maitresse garantissant que seul le propriétaire légitime de la clé privée correspondante peut décrypter la clé symétrique et le fichier demandé [Sudalai Muthu T. et Al., 2010].

2.3.4.2. Travail proposé par Razieh M. et Al., 2010

Dans cet article, les auteurs ont proposé un protocole d'échange de clé ID-KEX basé sur IBC pour assurer l'authentification mutuelle et la confidentialité des informations échangées entre le système d'information de la grille (GIS : Grid Information System) et les ressources. En outre, les auteurs ont évalué les performances du protocole proposé par rapport à celles du protocole SSL (Secure Socket Layer) qui est utilisé dans l'infrastructure de sécurité de la

grille (GSI : Grid Security Infrastructure) du middleware Globus pour assurer une communication sécurisée entre la GSI et les ressources. Le protocole ID-KEX a donné de meilleures performances en termes de temps et de consommation d'espace, ce qui semble plus efficace en présence d'un grand nombre de ressources [Razieh M. et Al., 2010].

2.3.4.3. Travail proposé par Rajesh I. and Sivakumar G., 2010

Dans cet article, les auteurs ont proposé une architecture de sécurité étendue de la grille (EGSI) pour supporter la communication sécurisée de groupe. En outre, les auteurs ont présenté un schéma d'authentification et de contrôle d'accès au niveau de l'organisation virtuelle. La figure 2.18 montre les différents composants de l'architecture proposée.

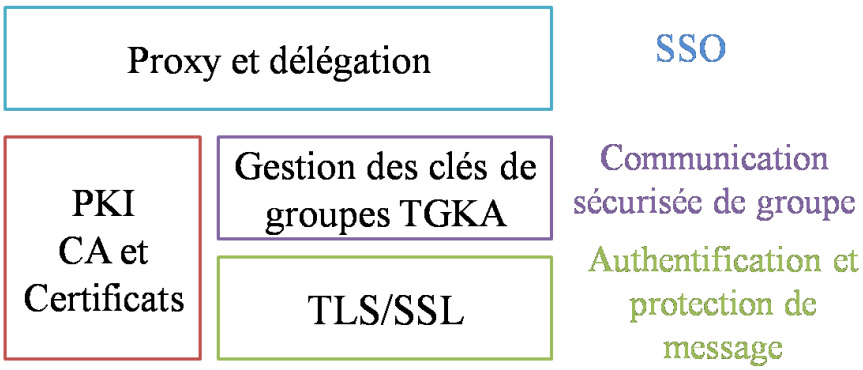

Figure 2. 18 Architecture d'EGSI [Rajesh I. and Sivakumar G., 2010]

EGSI permet aux utilisateurs et aux applications d'accéder en toute sécurité aux ressources. Elle est basée sur PKI. Les proxys et les fonctions de délégation sont nécessaires pour l'authentification unique. SSL est utilisé pour l'authentification et la protection des messages. Le composant de gestion des clés de groupe est introduit pour prendre en charge la communication de groupe sécurisée [Rajesh I. and Sivakumar G., 2010].

2.3.4.4. Travail proposé par Ashrafijoo B. et Al., 2010

Dans cet article, les auteurs ont présenté une nouvelle méthode pour évaluer et gérer la confiance dans un environnement de grille. Dans le modèle proposé, la théorie des probabilités a été utilisée pour évaluer et estimer la fiabilité et la confiance des ressources. De plus, dans ce modèle, les demandes sont définies dans un tableau en fonction du niveau de confiance d'accessibilité. Une des entités est sélectionnée en fonction de la qualité requise des programmes demandés. Il n'est pas toujours nécessaire de sélectionner une entité dans un domaine qui a le niveau de confiance le plus élevé, mais le critère de sélection d'une entité est

la qualité du service présenté par cette entité. Le taux de confiance de l'entité cliente est calculé en le comparant à l'entité serveur et si l'entité serveur a la confiance ainsi que la réputation nécessaires, l'entité cliente aura l'autorisation d'obtenir les services et cette position sera conservée pour sa connexion ultérieure [Ashrafijoo B. et Al., 2010].

2.3.4.5. Travail proposé par Khider H. et Al., 2010

Dans cet article, les auteurs ont proposé un modèle basé sur ABAC et qui utilise les deux technologies : SAML pour une authentification unique et XACML pour l'autorisation. Dans le modèle proposé les auteurs ont supposé que les fournisseurs de service (SP : Service Provider) et les fournisseurs d'identité (IdP : Identity Provider) se font mutuellement confiance au sein d'une organisation virtuelle (VO). D'abord, l'utilisateur tente d'accéder à un SP dans la VO pour exécuter une application spécifique. Comme le fournisseur de service et l'application ne connaissent pas l'utilisateur, alors il sera dirigé vers l'AAproxy pour un processus d'autorisation. En fonction de la demande de l'utilisateur, AAproxy génère une requête SAML et l'envoie à l'IdP pour récupérer ses attributs. Ensuite, l'utilisateur sera redirigé vers la VO qui contient la liste des institutions membres pour qu'il choisisse son IdP. La VO redirige alors l'utilisateur vers l'IdP choisi pour qu'il fournisse ses données d'authentification selon le mécanisme choisi (Identifiant/mot de passe ou certificat ou autre). Puis, en fonction de la demande SAML reçue, l'autorité d'attribut obtient les attributs de l'utilisateur utilisant le protocole LDAP conformément à la politique de libération d'attribut (ARP). Une réponse SAML est générée contenant les attributs de l'utilisateur et est ensuite envoyée à l'AAproxy. Après avoir obtenu des attributs authentifiés de l'utilisateur, le réalisateur d'attribut (AR: Attribute Realiser) qui fait partie de l'AAproxy envoie les attributs de l'utilisateur au service d'autorisation web. Ensuite, en fonction des attributs reçus, le générateur de requête (RG : Request Generator) génère une requête XACML qui sera évaluée par le PDP. Le PEP reçoit alors une décision d'autorisation qu'il envoie à l'AAproxy. Si la décision est « refusé », l'AAproxy demande à l'utilisateur s'il y'a une possibilité de fournir d'autres attributs en utilisant d'autres IdP et refaire les étapes, sinon l'AAproxy prend la décision finale concernant l'application et l'utilisateur est enfin autorisé ou non à exécuter son job [Khider H. et Al., 2010].

2.3.4.6. Travail proposé par Jaspher G.W.K. et Al., 2011

Dans cet article, les auteurs ont proposé une architecture de grille multi-domaines où chaque domaine se compose des éléments suivants :

- **Le service (S) :** représente le fournisseur de service de la grille.
- **La politique de service (SP) :** chaque propriétaire de service a sa propre politique de service basée sur l'organisation virtuelle et le domaine auquel il appartient.
- **Le domaine (DO) :** représente l'organisation où réside chaque service individuel, ce n'est pas nécessaire que tous les membres du même domaine appartiennent à la même VO.
- **L'organisation virtuelle (VO) :** elle regroupe temporairement des ressources appartenant à différents domaines pour atteindre un but commun.
- **La politique de la VO (VP) :** représente la politique de la VO qui est suivie par tous les membres de cette VO. Cette politique est un ensemble basé sur la combinaison des politiques du domaines (DO), politiques des services (SP), politiques d'authentification (AuP), politiques d'autorisation (AuZP) et d'autres politiques (OP).
- **Le client de sécurité (SC) :** cette fonction de sécurité est présente dans chaque service / ressource de l'organisation virtuelle (VO). C'est surtout pour la sécurité au niveau de l'hôte. Elle fournit une action contre les virus et autres attaques au niveau de l'hôte. Cette fonction de sécurité va analyser les ressources et les processus pour tout compromis de sécurité imminent. Une fois qu'un compromis est identifié, le détail est immédiatement envoyé au Gestionnaire de sécurité (SM) de la VO.
- **Le Gestionnaire de sécurité (SM) :** le SM est présent dans chaque organisation virtuelle. Il participe activement à la collecte des détails de l'attaque depuis son organisation virtuelle et la transmet au CSM. Les SM vont alternativement s'entrecroiser pour trouver la disponibilité des autres SM et CSM. Le SM enverra et recevra des mises à jour récentes concernant les divers compromis. C'est le SM qui enverra des informations à ses propres membres de l'organisation virtuelle et au gestionnaire chef de sécurité (CSM). Si le CSM lui-même a échoué, les SM de toutes les VO dans l'environnement de la grille commenceront un vote pour sélectionner un nouveau CSM immédiatement.
- **Gestionnaire Chef de Sécurité (CSM) :** ce gestionnaire de sécurité est sélectionné au hasard selon la méthode de vote parmi les organisations virtuelles existantes. Le CSM recueille tous les détails des compromis recueillis auprès des autres VO et de ses propres membres. Enfin, le CSM consolide les détails et propage les vrais compromis à tous les autres VO. Il devrait être capable d'analyser les compromis et de fournir des

solutions appropriées aux autres VO à travers leur SM. Ainsi, tous les autres VO sont capables de fournir des mesures de sécurité à leurs membres et également mettre à jour d'autres VO concernant les attaques récentes.

Pour assurer l'authentification, les auteurs ont utilisé une authentification à trois facteurs (Données biométriques, chaine secrète et une carte à puce). Pendant la phase d'enregistrement, l'utilisateur choisit une chaîne secrète qui est un nombre aléatoire et fournit les données biométriques qui peuvent être une reconnaissance de l'iris ou une empreinte digitale ou autre. Dans cette phase, un XOR entre la chaîne secrète et les données biométriques est fait, le résultat est stocké dans la carte à puce. Lors de la phase de connexion, le capteur de la carte à puce recueille les données biométriques de l'utilisateur. Puis une comparaison est faite au niveau du serveur distant entre les données de la carte à puce et les données collectées à partir du capteur de la carte. Le serveur distant ne contient aucune donnée de l'utilisateur mais seulement la fonction de comparaison des données (stockées et collectées) pour trouver la similarité/ Différence. Ainsi, la confidentialité de l'utilisateur est maintenue et le serveur distant n'a pas besoin d'une gestion de stockage sécurisé supplémentaire. Pour le contrôle d'accès, l'architecture proposée est basée sur le modèle RBAC. Les politiques de sécurité sont formulées en se basant sur la combinaison des politiques du domaine auquel l'utilisateur appartient, les politiques de la VO dans laquelle l'utilisateur est membre et les besoins spécifiques de l'utilisateur. Le modèle RBAC est implémenté en utilisant SAML/ XACML [Jaspher G.W.K. et Al., 2011].

2.3.4.7. Travail proposé par Kumari A.K. et Al., 2011

Dans cet article, les auteurs ont proposé un modèle d'authentification, d'autorisation et de distribution de clé pour assurer une communication de groupe sécurisée dans un environnement de grille. Les informations d'authentification sont stockées sur deux serveurs. Lors de la phase de connexion, si l'utilisateur est authentifié une clé aléatoire est générée et distribuée entre les membres du groupe. Le message digest du message est crypté puis diffusé au groupe sur le canal. Enfin, les récepteurs le décryptent et le décodent pour obtenir le message original [Kumari A.K. et Al., 2011].

2.4. Conclusion

Dans ce chapitre, nous avons donné un aperçu sur la sécurité des grilles. D'abord, nous avons présenté et détaillé les différents travaux qui ont été proposés pour résoudre les différentes issues de sécurité dans les grilles. Puis, ces travaux ont été classés selon l'issue qu'ils prennent en considération telle que : l'authentification, le contrôle d'accès, l'intégrité, la confidentialité et les issues multiples. Nous avons trouvé que chaque issue a des méthodes spécifiques qui ont été utilisées seules ou combinées. En outre, l'intégrité et la confidentialité n'étaient pas des issues visées par les chercheurs dans les travaux récents. Par contre, le contrôle d'accès a pris l'attention de plusieurs chercheurs car plusieurs solutions ont été proposées pour résoudre cette issue. Dans le chapitre suivant, nous allons explorer ce qui a été fait dans le domaine de la sécurité du Cloud Computing.

Chapitre III : La sécurité dans le Cloud Computing (Etat de l'art)

3.1. Introduction

Le Cloud Computing offre de nombreux avantages aux organisations mais, la sécurité est l'une des issues qui entravent son évolution. Il est clair que le problème de sécurité a joué le rôle le plus important dans l'acceptation du Cloud car pour la plupart des organisations, stocker les données et exécuter des applications sur un disque dur de quelqu'un d'autre utilisant un CPU d'une tierce personne semble décourageant. De plus, la perte des données, le hameçonnage et le botnet constituent de sérieuses menaces pour les données et les logiciels de l'organisation. Dans ce chapitre, nous allons voir les différents problèmes de sécurité détectés dans un environnement de Cloud. Ensuite, nous présenterons les différents travaux qui ont été proposés pour résoudre chaque problème. Ces travaux ont été classés selon le problème résolu : authentification, contrôle d'accès, intégrité, confidentialité et issues multiples.

3.2. Les problèmes de sécurité introduits par le Cloud Computing

Comme déjà mentionné, la sécurité est l'une des plus grandes préoccupations dans la mise en œuvre et l'utilisation des services du Cloud Computing. D'un point de vue technique, la majorité des risques de sécurité associés au Cloud, sont déjà présents dans les centres traditionnels de données. La virtualisation augmente l'impact de certains de ces risques, car les attaques réussies sur la machine hôte (où se trouve l'hyperviseur) peuvent compromettre potentiellement toutes les machines virtuelles hébergées. Les failles sur les plateformes de virtualisation elles-mêmes représentent un risque évident. En outre, la disponibilité est également une préoccupation majeure même s'il n'y a pas de différences fondamentales, d'un point de vue strictement technique, entre les services traditionnels et les services en nuage, en termes simples, la disponibilité signifie qu'une organisation dispose de son ensemble complet de ressources informatiques accessibles et utilisables à tout moment. Donc dans un environnement Cloud, la majorité des problèmes de sécurité sont liés au besoin implicite de faire confiance à des parties externes pour maintenir des informations critiques et fournir des services informatiques critiques. Ce besoin était déjà présent dans les services informatiques traditionnels, chaque fois qu'un nouvel équipement ou de nouvelles applications étaient déployés dans le centre de données, il y avait un besoin implicite de faire confiance aux fournisseurs associés. Néanmoins, il existe une différence fondamentale: dans le Cloud Computing, il est beaucoup plus difficile de gérer la chaîne de confiance, car il n'y a pas de

vision claire (pour le client) de la manière dont le service est fourni. Le client connaît uniquement le fournisseur des services, mains l'ensemble des composants de services sous-traités est généralement non vérifiable par des tiers.

3.3. Etat de l'art sur la sécurité du Cloud Computing

Dans cette partie, nous allons présenter et détailler les différents travaux qui ont été proposés pour assurer la sécurité dans un environnement de Cloud. En se basant sur l'issue de sécurité traitée, nous avons classé ces travaux selon les classes présentées dans la section 2.3 du chapitre 2.

3.3.1. Le contrôle d'accès

Nous avons vu dans la section 2.3.1 du chapitre 2, que cette issue de sécurité peut être divisée en trois parties, à savoir : l'identification, l'authentification et l'autorisation (contrôle d'accès).

3.3.1.1. L'authentification

Dans un environnement de Cloud, l'authentification est importante tout comme dans les grilles, les utilisateurs des services du Cloud aimeraient bien avoir à faire à des fournisseurs de services authentifiés. D'un autre coté, ces fournisseurs veulent fournir leurs services à des utilisateurs authentifiés. Nous avons présenté les différents types de protocoles d'authentification dans le chapitre 2. Plusieurs travaux ont été proposés pour assurer l'authentification dans un environnement de Cloud, nous allons voir ces travaux dans ce qui suit.

- **Technique d'authentification proposée par Dinesha H.A. and Agrawal V.K., 2012**

Dans cet article, les auteurs ont proposé une technique d'authentification qui permet de générer un mot de passe et le concaténer avec ceux générés à plusieurs niveaux. L'accès au Cloud est basé sur le mot de passe concaténé au niveau-feuille mais l'utilisateur peut accéder aux services si l'authentification est réussie à tous les niveaux précédents. Le premier niveau représente la génération du mot de passe au niveau de l'organisation (utilisant les informations sur l'organisation). Le second niveau permet de générer un mot de passe au niveau de l'équipe (les informations sur l'équipe sont utilisées), ceci permet d'authentifier une équipe pour un service Cloud particulier. Enfin, le dernier niveau permet la génération du mot de passe au niveau de l'utilisateur final (en utilisant les informations de l'utilisateur). Ces trois

mots de passe générés sont concaténés. Cette technique est simple, car malgré qu'elle soit à plusieurs niveaux, à chaque étape l'utilisateur n'a qu'à se souvenir d'un seul mot de passe [Dinesha H.A. and Agrawal V.K., 2012].

- **Technique d'authentification proposée par Velciu M.A. et Al., 2014**

Dans cet article, les auteurs ont proposé une méthode d'authentification basée sur le cryptage biométrique dans un environnement de Cloud. Ils ont utilisé l'algorithme fuzzy-vault qui est une construction bio-cryptographique, conçue pour fonctionner avec des caractéristiques biométriques (les données d'entrée doivent toujours être représentées comme des ensembles de données non ordonnées). La valeur secrète cryptée peut être récupérée uniquement si les données de requête à cet effet chevauchent sensiblement l'ensemble de données utilisées pour le codage. Pendant la phase de cryptage de cet algorithme, la valeur secrète S (PIN, mot de passe ou une clé cryptographique) est divisée en groupes de b bits chacun (les coefficients), pour former la construction polynomiale P de degré k. En utilisant la requête biométrique Q et la construction polynomiale, l'ensemble de points authentiques G est calculé comme G = (xi; P (xi)); i = 1..t, où xi représente chaque valeur de donnée de gabarit biométrique et t représente le nombre total de points de consigne de données de gabarit biométrique qui sont codés dans la construction. La dernière étape nécessaire pour achever le processus de cryptage (et probablement la plus importante) est la génération d'un grand nombre de points de réglage aléatoires, C = (ci; di); i = 1..m, qui ne repose pas sur le polynôme P, dans le but de diffuser les points authentiques. Le bio-cryptogramme fuzzy-vault est obtenu comme la réunion brouillée des deux ensembles de données, les points de paillettes véritables et aléatoires: V = G U C. le décryptage peut être réalisé en fournissant une autre requête biométrique fraîche Q. À partir de ce nouvel échantillon, les points candidats sont identifiés par comparaison avec les valeurs d'abscisse des points résidant dans la construction du fuzzy-vault. Si un nombre suffisant de points sur le polynôme P peut être identifié, alors l'interpolation de Lagrange peut être utilisée pour reconstruire le polynôme P et récupérer le secret S. Le nombre minimum de points nécessaires pour reconstruire un polynôme de degré k est k + 1 [Velciu M.A. et Al., 2014].

- **Technique d'authentification proposé par Sarvabhatla M. et Al., 2014**

De nombreux chercheurs ont proposé des schémas cryptographiques, qui permettent une authentification mutuelle entre un serveur et un utilisateur du Cloud avec une efficacité

variée. Dans [Nimmy K. and Sethumadhavan M., 2014], Les auteurs ont proposé un nouveau protocole d'authentification mutuelle dans un environnement de Cloud utilisant le partage secret et la stéganographie. Le système est composé de quatre étapes: l'enregistrement, la connexion, l'authentification et le changement de mot de passe. Dans cet article, les auteurs ont montré que le schéma présenté par Nimmy K. and Sethumadhavan M., souffre des attaques de devinette de mot de passe et celles du déni de service. En outre, ils ont expliqué que les opérations cryptographiques et stéganographiques requièrent d'énormes ressources de calcul et conviennent mieux au côté serveur plein de ressources. Dans le schéma de [Nimmy K. and Sethumadhavan M., 2014], les périphériques côté client sont limités en ressources mais prennent en charge d'énormes opérations cryptographiques et stéganographiques, ce qui rend le système impossible à adopter pour le scénario du Cloud. Enfin, les auteurs ont proposé un système d'authentification mutuelle amélioré, léger sur les côtés client et serveur, et résistant à toutes les attaques cryptographiques majeures [Sarvabhatla M. et Al., 2014].

- **Technique d'authentification proposée par Singh A. and Chatterjee K., 2015**

Dans cet article, les auteurs ont proposé un schéma d'authentification multi-niveaux pour accéder aux services du Cloud. Ce schéma est divisé en deux niveaux. Dans le premier niveau, l'utilisateur entre le nom d'utilisateur et le mot de passe, ce qui représente un schéma d'authentification simple. Tandis que la technique d'authentification du deuxième niveau est basée sur une séquence d'activité prédéterminée sur un écran virtuel. Cette séquence doit être identique à celle que l'utilisateur a effectuée pendant l'enregistrement. Cet écran virtuel est chargé par un observateur (il s'agit du programme d'application exécuté côté client) [Singh A. and Chatterjee K., 2015].

- **Technique d'authentification proposée par Mansour A. and Sadik M., 2015**

Dans cet article, les auteurs ont proposé une technique d'authentification à plusieurs facteurs basée sur les biométries multimodales (MFA-MB) dans un environnement de Cloud. Le premier facteur représente un mot de passe (facteur de connaissance), le second facteur représente un jeton (facteur de possession) et le troisième facteur consiste à l'utilisation de la biométrie multimodale (facteur biométrique) où plusieurs types de biométries ont été fusionnés en utilisant la méthode du degré du score correspondant [Mansour A. and Sadik M., 2015].

- **Technique d'authentification proposée par Al-Attab B.S. and Fadewar H.S., 2016**

Dans cet article, les auteurs ont proposé un schéma d'authentification qui se compose de trois phases : l'enregistrement, la connexion, l'authentification et de deux activités : le changement du mot de passe ainsi que le backup du jeton. Dans cette proposition, les auteurs ont supposé que l'environnement du Cloud est protégé par un générateur de jeton USB (UTG : Usb Token Generator). Ce générateur génère des jetons USB pour chaque utilisateur durant la phase d'enregistrement où l'utilisateur choisit, un identifiant et un mot de passe qui seront utilisés avec une fonction de hachage, le résultat obtenu sera ensuite stocké dans le jeton qui sera remis à l'utilisateur. Dans le schéma proposé, l'utilisateur doit fournir le jeton et introduire son identifiant et son mot de passe. Si l'utilisateur perd son jeton USB, personne ne peut accéder à son compte sans le mot de passe et pour plus de sécurité, l'utilisateur peut bloquer son jeton gratuitement en utilisant son identifiant de backup qui a été crée lors de la phase d'enregistrement [Al-Attab B.S. and Fadewar H.S., 2016].

- **Technique d'authentification proposée par Talkhaby H.R. and Parsamehr R., 2016**

Dans cet article, les auteurs ont proposé un mécanisme d'authentification basé sur l'amélioration du protocole Kerberos 5 avec l'algorithme d'échange de clé Diffie-Hellman-DSA et les empreintes digitales de l'utilisateur dans un environnement de Cloud. L'utilisation des données biométriques a fourni un mécanisme de non répudiation et a résolu les limites d'authentification basées sur le mot de passe. En outre, l'utilisation de l'algorithme d'échange de clé de Diffie-Hellman-DSA avec une authentification mutuelle a permis de résoudre le problème de l'attaque d'obtention du mot de passe dans le protocole Kerberos 5 [Talkhaby H.R. and Parsamehr R., 2016].

3.3.1.2. Le contrôle d'accès (Autorisation)

La première façon avec laquelle on peut assurer la sécurité d'un système consiste à contrôler l'accès à ses données et ses ressources. Les décisions de contrôle d'accès sont très importantes pour tout système partagé. Cependant, pour un grand système distribué comme le Cloud Computing, la décision d'accès doit être plus flexible et plus évolutive. Ces dernières années, la proposition des mécanismes de contrôle d'accès dans un environnement du Cloud est devenue un sujet de recherche brûlant. Le but principal de ces travaux est de garantir l'accès aux services du Cloud uniquement aux utilisateurs légaux et d'empêcher toute utilisation non autorisée. Dans ce qui suit, nous allons présenter et détailler les différents travaux qui ont été

présentés pour résoudre l'issue du contrôle d'accès (autorisation) dans le Cloud Computing. Les différents modèles utilisés dans ces propositions ont été expliqués en détails dans le chapitre 2.

- **Modèle de contrôle d'accès proposé par Mon E. and Naing T., 2011**

Dans cet article, les auteurs ont proposé un modèle de contrôle d'accès qui combine les deux approches RBAC et ABAC pour assurer la confidentialité des données dans un Cloud privé. Ce modèle reste un modèle purement théorique car aucune implémentation n'a encore été proposée [Mon E. and Naing T., 2011].

- **Modèle de contrôle d'accès proposé par Sun L. et Al., 2012**

Dans cet article, les auteurs ont proposé un modèle de contrôle d'accès sémantique basé sur RBAC. Des vocabulaires structurés et hétérogènes ont été utilisés avec les ontologies dans le système e-healthcare, ce qui a permis de résoudre le problème d'un contrôle d'accès distribué dans un environnement dynamique tel que le Cloud Computing. Ce modèle reste un modèle purement théorique car aucune implémentation n'a encore été proposée [Sun L. et Al., 2012].

- **Modèle de contrôle d'accès proposé par Chunlei W. et Al., 2012**

Dans cet article, les auteurs ont introduit une valeur de permission, un rôle quantifié et une valeur de comportement pour construire un modèle de contrôle d'accès basé sur un rôle quantifié. Ce modèle a été validé dans un prototype du Cloud Computing, ce qui a donné une réduction du nombre de rôles, une amélioration du processus d'autorisation et une implémentation dynamique des permissions [Chunlei W. et Al., 2012].

- **Modèle de contrôle d'accès proposé par Yue-qin F. and Yong-sheng Z., 2012**

Les auteurs de cet article ont proposé un modèle de contrôle d'accès spécifique au Cloud Computing qui repose sur RBAC ainsi qu'un modèle d'accès basé sur les tâches. Ce qui a permis d'avoir les avantages des deux modèles en intégrant une valeur de réputation. Cette valeur a pour but de diminuer le nombre d'accès non autorisés. Le point faible de cet article est que les auteurs n'ont pas précisé comment calculer la valeur de réputation [Yue-qin F. and Yong-sheng Z., 2012].

- **Modèle de contrôle d'accès proposé par Dos Santos D. et Al., 2013**

Dans cet article, les auteurs ont proposé un modèle de contrôle d'accès à base de risque au sein d'une fédération de Cloud sans la nécessité d'une fédération d'identité. Le modèle

proposé utilise des politiques de risques sous la forme de fichiers XML, lors d'une demande d'accès par un utilisateur du même Cloud, la requête est gérée par le modèle ABAC classique, par contre lors d'une demande d'accès par un utilisateur appartenant à un autre Cloud, si il n'y'a pas une fédération d'identité entre les deux Cloud, le modèle de contrôle d'accès à base de risque est activé [Dos Santos D. et Al., 2013].

- **Modèle de contrôle d'accès proposé par Auxilia M and K. Raja, 2014**

Dans cet article, les auteurs ont proposé un contrôle d'accès dynamique pour sécuriser l'accès aux données dans un Cloud. Cette approche considère une relation entre le demandeur, la donnée demandée et l'action qui va être réalisée sur cette donnée. Le modèle proposé utilise une architecture de contrôle d'accès basée sur le contexte sémantique. Cette architecture se compose des éléments suivants (Figure 3.1):

- **La base des ontologies :** elle contient les ontologies des domaines à savoir : les ontologies sujets (utilisateurs), les ontologies objets (les données) et les ontologies actions (les actions).
- **La base d'autorisation :** est une base qui contient les politiques de contrôle d'accès qui ont été définies à l'avance. Ces politiques prennent en considération les ontologies sujets, les ontologies objets et les ontologies actions.
- **Le gestionnaire des ontologies** : reçoit comme entrée la base des ontologies et applique la propriété de subsomption (inclusion entre les concepts) sur les différentes ontologies de domaine de contrôle d'accès afin de réduire les inférences possibles entre elles.
- **L'engin d'analyse du contexte:** il analyse la situation dans laquelle l'utilisateur effectue la demande. Il reçoit une entrée de l'ontologie de contexte et, en fonction de la relation entre les domaines du contexte, il saura si l'utilisateur effectue des requêtes dans un contexte correct.
- **Le moteur d'inférence :** est l'élément important de cette architecture. Ce moteur reçoit la demande d'autorisation du sujet, puis il la compare avec les règles d'autorisation prédéfinies dans la base d'autorisation et obtient également le contexte à partir du moteur d'analyse contextuelle. Si la demande d'autorisation correspond à l'une des règles de la base d'autorisation, les droits d'accès sont acceptés pour effectuer l'action demandée sur l'objet requis. Sinon, les droits d'accès sont refusés.

Le moteur d'inférence reçoit la requête d'accès de l'utilisateur, il contacte le gestionnaire des ontologies qui à son tour obtient les ontologies des différents domaines de contrôle d'accès. Le gestionnaire des ontologies dérive l'interrelation entre ces domaines et réduit les ontologies en utilisant la propriété de subsomption. Ces ontologies réduites sont données comme entrée au moteur d'inférence. Le moteur d'inférence compare ensuite la demande d'accès effectuée par l'utilisateur avec les politiques de contrôle d'accès déjà stockées dans une base d'autorisation. En outre, le moteur d'inférence obtient le contexte de l'engin d'évaluation du contexte. Lorsque la demande d'autorisation d'un utilisateur correspond aux règles de la base d'autorisation, il accorde les droits d'accès à l'utilisateur, sinon il ne le fait pas [Auxilia M and Raja K., 2014].

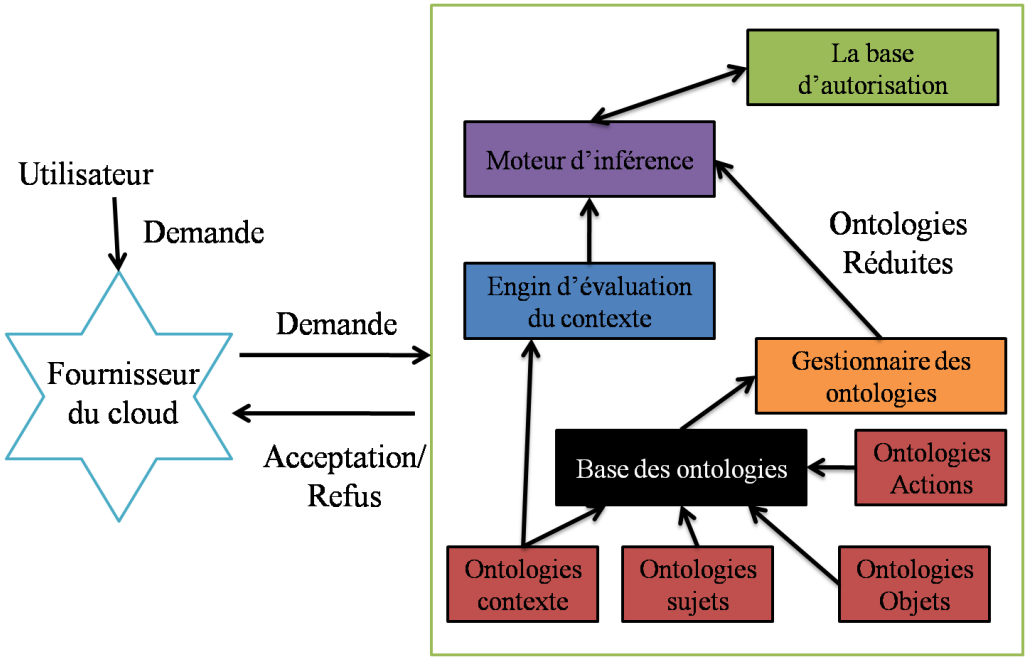

Figure 3. 1 Modèle de contrôle d'accès dynamique **[**Auxilia M and K. Raja, 2014]

- **Modèle de contrôle d'accès proposé par Ayache M. et Al., 2015**

Dans cet article, les auteurs ont proposé un middleware qui fournit une vérification de haut niveau des politiques de sécurité en utilisant des requêtes de contrôle d'accès rapides. Cela garantit au propriétaire des données d'imposer sa propre politique de sécurité et de gérer l'accès à ses ressources tout en les partageant avec d'autres. L'approche proposée vise à appliquer la politique de sécurité sans affecter la demande de l'utilisateur. En effet, l'utilisateur peut envoyer une requête d'accès qui est alors reçue par un middleware noté curlX qui la traduit en requête XACML. Cette requête sera envoyée à l'outil d'implémentation de politique

XACML (contenant l'ensemble des politiques définies) pour décider si l'utilisateur est autorisé à accéder aux données ou non. Pour la validation de l'approche proposée, les auteurs ont implémenté un tel middleware en utilisant le stockage d'objets Openstack (Swift). Par contre, pour la politique et l'étude de cas, ils ont considéré un scénario de référence d'accident vasculaire cérébral. Ce scénario consiste en une collaboration entre trois types d'organisations médicales pour regrouper plusieurs compétences pertinentes afin de contribuer à l'obtention d'un diagnostic aussi précis et correct que possible. Chaque organisation a sous-traité ses données médicales à un fournisseur de Cloud (Openstack dans ce cas) et impose un ensemble de règles de sécurité pour gérer leurs accès. La requête envoyée par l'utilisateur peut être une requête d'accès ou bien une requête acl. Dans le cas d'une requête acl, le middleware curlX la réorganise comme une règle de sécurité et l'ajoute à la base des politiques XACML comme une nouvelle règle via le point d'administration de la politique (PAP) [Ayache M. et Al., 2015].

- **Modèle de contrôle d'accès proposé par Varadharajan V. et Al., 2015**

Dans cet article, les auteurs ont proposé un modèle de contrôle d'accès centré sur le rôle et sur les attributs. Cette combinaison a permis d'assurer la flexibilité complète d'ABAC tout en améliorant l'efficacité de l'audit. La modification a été faite en utilisant les opérateurs logiques (ET/ OU). En outre, les auteurs ont défini l'attribut « rôle » comme un attribut obligatoire dans chaque expression booléenne de la politique, ce qui permet d'avoir les avantages du modèle RBAC. Les auteurs ont également proposé un arbre d'accès (Figure 3.2). Dans cet arbre, le nœud racine doit être un 'OU' et chaque enfant de ce nœud doit être un 'ET'. Dans chaque sous-arbre du niveau 1 du 'ET', l'enfant droit doit être un attribut T(k) ou bien un arbre vide, et l'enfant gauche doit être un rôle R(k) et sera vide uniquement si l'enfant droit est vide. Une fois que l'arbre est défini, le reste du modèle est un contrôle d'accès basé sur ABAC [Varadharajan V. et Al., 2015].

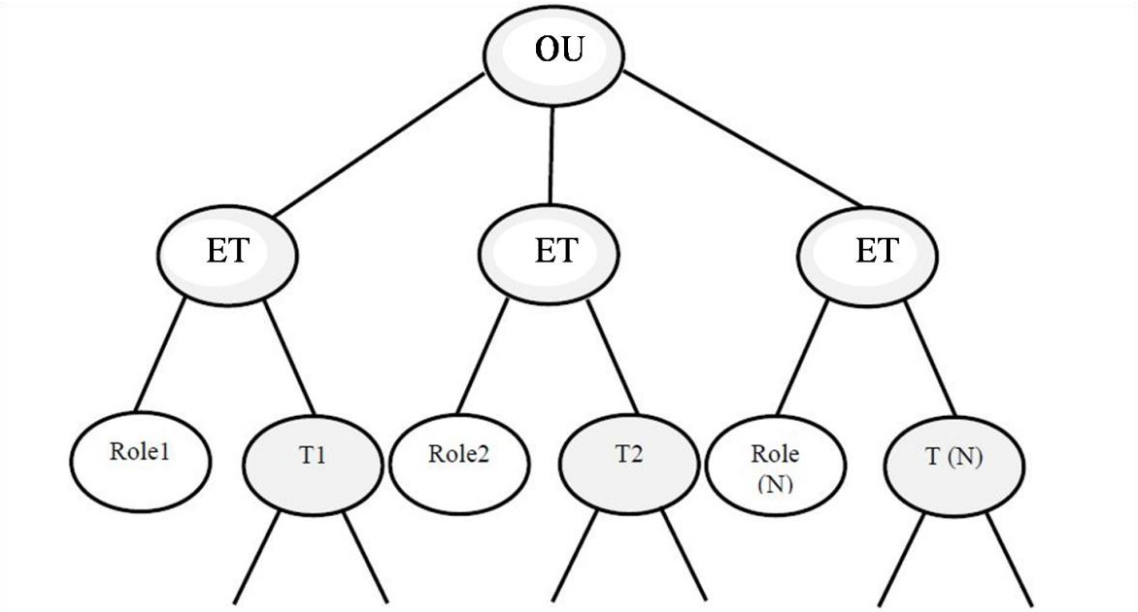

Figure 3. 2 Arbre d'accès [Varadharajan V. et Al., 2015].

- **Modèle de contrôle d'accès proposé par Li N. et Al., 2015**

Le Cloud hybride multi-niveaux est composé de différents modes de Cloud dont chacun a des niveaux de sécurité et des exigences de sécurité différents, restant relativement indépendant, assurant mutuellement l'interopérabilité et le partage des données, ce qui pose des problèmes de sécurité entre les Clouds, les niveaux et les domaines. Dans cet article, les auteurs ont proposé un modèle de gestion de l'autorisation inter-domaines dans un environnement de Cloud hybride et à plusieurs niveaux. Le modèle proposé a divisé le rôle traditionnel du modèle RBAC en deux parties. La première consiste en un rôle de position et la deuxième partie représente un rôle d'application. Ceci a permis de satisfaire les besoins pratiques du niveau organisation et du niveau application en même temps. En outre, les auteurs ont présenté une conversion de rôle unidirectionnelle pour réaliser une autorisation et une collaboration inter-domaines sécurisée. Enfin, pour décrire le modèle proposé, les auteurs ont utilisé la logique de description dynamique [Li N. et Al., 2015].

- **Modèle de contrôle d'accès proposé par Khaled R. et Al., 2015**

Dans cet article, les auteurs ont proposé le modèle de contrôle d'accès AR-ABAC qui utilise le modèle ABAC classique et une nouvelle notion présentée par les auteurs nommée les règles d'attributs (AR). Ces règles d'attributs permettent l'accès aux objets selon leurs degrés de

sensibilité. D'autre part, elles permettent de déterminer combien d'attributs et quel type d'attributs sont utilisés pour prendre une décision d'accès [Khaled R. et Al., 2015].

- **Modèle de contrôle d'accès proposé par Aluvalu R.K. and Muddana L., 2016**

Dans cet article, les auteurs ont proposé un modèle de contrôle d'accès hybride dans un environnement de Cloud. Ce modèle d'hybridation peut être utilisé seul et peut également être utilisé avec n'importe quel modèle de contrôle d'accès basé sur le cryptage des attributs. La figure 3.3, présente l'hybridation du mécanisme d'autorisation dans un système de stockage distant en ajoutant l'engin de risque au système de contrôle d'accès existant. Cet engin calcule le seuil de risque en utilisant les attributs et leurs valeurs fournis par l'utilisateur lors de la phase d'enregistrement. Le consommateur des données demande une autorisation d'accès à l'engin de risque lorsqu'il obtient un refus du modèle de contrôle d'accès traditionnel, ce qui rend le système proposé plus dynamique [Aluvalu R.K. and Muddana L., 2016].

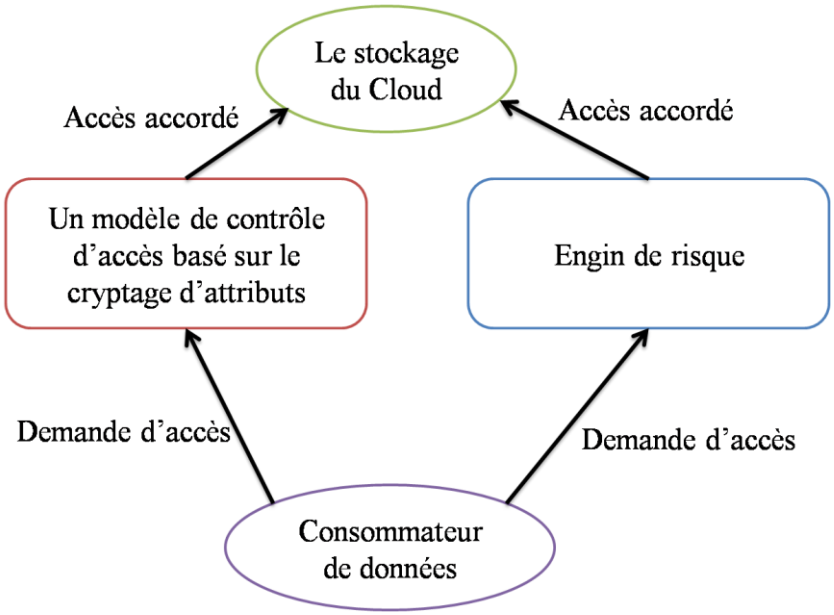

Figure 3. 3 Modèle de contrôle d'accès hybride [Aluvalu R.K. and Muddana L., 2016]

- **Modèle de contrôle d'accès proposé par Chen A. et Al., 2016**

Dans cet article, les auteurs ont proposé un modèle de contrôle d'accès dynamique basé sur le risque (DRAC), ce modèle repose sur le modèle ABAC ainsi qu'un mécanisme d'évaluation de risques. Les auteurs ont analysé la régression itérative en se basant sur une fenêtre de flux de données, c'est ce qui a permis de calculer efficacement les facteurs de risque environnementaux d'un demandeur [Chen A. et Al., 2016].

- **Modèle de contrôle d'accès proposé par Khan F. et Al., 2016**

Dans cet article, les auteurs ont proposé un modèle de contrôle d'accès basé sur le cryptage d'attributs à plusieurs autorités. Le modèle proposé permet au propriétaire de données de spécifier quelle partie de données peut être accédée par un utilisateur en utilisant des attributs cryptés. D'autre part, le modèle proposé a pris en considération la réduction du coût de cryptage en diminuant les attributs répétitifs. Ce modèle n'a pas pris en considération le cas où les utilisateurs quittent le système et reste un modèle purement théorique car aucune implémentation n'a encore été proposée [Khan F. et Al., 2016].

- **Modèle de contrôle d'accès proposé par Jianan H. et Al., 2017**

Dans cet article, les auteurs ont proposé un modèle de contrôle d'accès qui combine les deux facteurs : temps et attribut dans un environnement de Cloud. En se basant sur le schéma proposé, les auteurs ont également proposé une approche efficace pour concevoir des politiques d'accès confrontées à diverses exigences d'accès pour les données sensibles au temps. Une analyse approfondie de la sécurité et des performances montre que le système proposé est très efficace et répond aux exigences de sécurité pour le stockage de données sensibles au temps dans un Cloud public [Jianan H. et Al., 2017].

3.3.2. L'intégrité

Dans un environnement de Cloud, nous n'avons pas trouvé un grand nombre de travaux récents qui traitent uniquement l'issue d'intégrité. Le seul travail que nous avons trouvé était celui de [Dinesh C, 2018], où l'auteur a proposé une technique qui permet de garantir l'intégrité des données hébergées dans le Cloud, en utilisant un algorithme de protocole de lecture. Cet algorithme est déployé du coté utilisateur pour mesurer les données à héberger. Une fois que ces données sont externalisées vers le Cloud, ce même algorithme est utilisé pour les mesurer à nouveau. Un autre algorithme de comparaison de données au niveau de plusieurs serveurs du Cloud a été utilisé pour chaque donnée externalisée. Cet algorithme permet de gérer la récupération des données. Dans le cas de défaillance du serveur du Cloud, des données entières peuvent être affectées. En outre, l'utilisateur ne peut pas prévoir l'intégrité de ses données car ceci dépend du fournisseur de service, qui peut masquer la perte de certaines données. Pour répondre à cette problématique, les auteurs ont proposé un algorithme de gestion des données au niveau des serveurs et un protocole automatique qui permet de connaitre l'échange entier des données avant et après l'insertion de ces dernières

dans les serveurs multiples du Cloud. Cet algorithme permet à l'utilisateur de savoir si des modifications ou des suppressions ont eu lieu.

3.3.3. La confidentialité

Dans un environnement de Cloud, les auteurs dans [Zardari M.A., 2014], ont proposé une approche de classification des données basée sur K-voisins les plus proches (KNN : K-Nearest Neighbors) pour assurer la confidentialité des données. La classification est basée sur les besoins de sécurité des données (données sensibles et données non sensibles). Les données sensibles sont cryptées à l'aide de l'algorithme RSA.

3.3.4. Issues multiples

Dans cette partie, nous allons présenter les différents travaux qui ont pris en considération plusieurs issues de sécurité dans un environnement de Cloud.

3.3.4.1. Travail proposé par Luo W. and Bai G., 2011

Dans cet article, les auteurs ont proposé un schéma qui se compose des quatre algorithmes suivants :

- **KeyGen** : est un algorithme qui permet de générer une clé, il est exécuté par l'utilisateur afin de mettre en marche le schéma.
- **SigGen** : cet algorithme est déployé par l'utilisateur pour générer des métadonnées de vérification. Ces métadonnées peuvent être des signatures ou d'autres informations utilisées pour l'audit.
- **GenProof** : exécuté par le serveur du Cloud pour générer la preuve de l'exactitude du stockage des données.
- **VerifyProof :** exécuté par la tierce partie d'audit (TPA) pour vérifier la preuve du serveur du Cloud.

Le système de vérification publique passe par les deux phases suivantes :

- **La phase d'installation (Setup)** : dans cette phase, l'utilisateur initialise les paramètres secrets et publics du système en exécutant *KeyGen* et prétraite le fichier de données F en utilisant *SigGen* pour générer les métadonnées de vérification. Puis, l'utilisateur enregistre le fichier de données F dans le serveur du Cloud. Ensuite, il

supprime toutes les copies locales du fichier et publie les métadonnées de vérification à la TPA pour le futur audit.

- **La phase d'audit** : dans cette phase, la TPA génère un message d'audit ou un challenge pour le serveur du Cloud. Ce message permet de s'assurer que le serveur du Cloud a conservé le fichier F correctement au moment de l'audit. Ensuite, le serveur va dériver un message de réponse d'une fonction du fichier de donnés F en exécutant *GenProof*. En utilisant les métadonnées la TPA vérifie la réponse via *VerifyProof*.

Dans cet article, les auteurs ont pu assurer l'authentification en utilisant la signature RSA, l'intégrité, en utilisant une fonction de hachage et la confidentialité en utilisant le chiffrement symétrique [Luo W. and Bai G., 2011].

3.3.4.2. Travail proposé par Zhu T. et Al., 2011

Dans cet article, les auteurs ont proposé un modèle de contrôle d'accès nommé CoRBAC (Cloud optimized RBAC) qui est basé sur le modèle RBAC, plus précisément sur le RBAC distribué (dRBAC). Dans le modèle proposé, les auteurs ont fusionné les services d'authentification distribuée de (dRBAC) et ont étendu la fonction de l'autorité de certification (CA), en offrant l'authentification inter-domaines et l'affectation de rôles inter-domaines. En outre, le modèle proposé a ajouté une racine CoRBAC qui se chargera de : stocker les informations sur les domaines des utilisateurs, fournir les conditions d'accès aux services et gérer les rôles des utilisateurs. Enfin, dans cette proposition les auteurs ont amélioré l'efficacité du contrôle d'accès en ajoutant des caches hiérarchiques, ils ont utilisé deux niveaux de caches d'authentification afin de réduire la charge sur le serveur racine CoRBAC global [Zhu T. et Al., 2011].

3.3.4.3. Travail proposé par Chugh S. and Peddoju S.K., 2012

Dans cet article, les auteurs ont utilisé un Cloud hybride (Cloud public et Cloud privé). Le Cloud public a été utilisé uniquement pour l'hébergement des documents. Par contre, le Cloud privé a été utilisé pour la gestion des clés, l'authentification et le contrôle d'accès. Les auteurs ont utilisé une authentification à un seul facteur (identifiant/ mot de passe). D'abord le propriétaire crypte le fichier et le stocke dans le Cloud. Le mécanisme de chiffrement utilisé inclut des métadonnées attachées à l'objet protégé. En outre, ces métadonnées font partie de l'entête du fichier crypté et sont toujours insérées au début du fichier. Dans ce modèle, le contrôle d'accès est basé sur une table d'accès contenant les utilisateurs, les actions permises

ainsi que la raison de la permission (permission des groupes et appartenance des utilisateurs aux groupes) [Chugh S. and Peddoju S.K., 2012].

3.3.4.4. Travail proposé par Gonzalez N.M. et Al., 2013

Dans cet article, les auteurs se sont concentrés sur l'étude de l'état de l'art relatif aux architectures de gestion des informations d'identification (credentials). Puis, ils ont donné une classification pour ces informations selon les types suivants :

- **Métadonnées** : qui sont utilisées pour stocker les informations concernant l'entité, telles que : les attributs d'identification, la migration entre Cloud, l'enregistrement du changement des états du cycle de vie, utilisation et propriétaire.
- **Autorisation** : ces informations spécifient les actions pouvant être effectuées par une entité, quelles ressources cette entité peut accéder et tout autre avantage qu'elle pourra avoir.
- **Obligation** : ces informations définissent les règles et les politiques qu'une entité doit suivre.

Ensuite, les auteurs ont donné une comparaison détaillée entre ces informations d'identification (credentials) et les identités. Enfin, un cadre d'authentification et d'autorisation basé sur ces informations a été proposé dans un environnement de Cloud [Gonzalez N.M. et Al., 2013].

3.3.4.5. Travail proposé par Liu X. et Al., 2013

Dans cet article, les auteurs ont traité trois issues de sécurité, à savoir : l'authentification, le contrôle d'accès et la confidentialité. Les auteurs ont utilisé une signature basée sur les attributs (ABS : Attribute Based Signature) [Maji H. et Al., 2011] qui permet d'assurer l'authentification de l'utilisateur sans divulguer son identité. Dans le mécanisme ABS, les utilisateurs possède un prédicat de revendication associé à un message, ce prédicat aide à identifier l'utilisateur comme un utilisateur autorisé sans révéler son identité. Pour assurer un contrôle d'accès efficace, les auteurs ont utilisé une politique chiffrée qui utilise un chiffrement basé sur les attributs (CP-ABE : Ciphertext Policy Attribute Based Encryption) [Bethencourt J. et Al., 2007]. Dans le schéma CP-ABE standard, il y'a une seule autorité qui gère les attributs et la distribution des clés. Or, dans un environnement de Cloud, un utilisateur peut posséder des attributs générés par plusieurs autorités. D'un autre coté, le

propriétaire peut vouloir partager des données avec des utilisateurs administrés par plusieurs autorités. Pour faire face à cette problématique, les auteurs ont étendu le schéma CP-ABE en utilisant une structure hiérarchique de plusieurs autorités. En outre, le langage XACML a été utilisé comme un langage de description de la politique. Enfin, la confidentialité des données hébergées a été assurée en utilisant un mécanisme de chiffrement symétrique [Liu X. et Al., 2013].

3.3.4.6. Travail proposé par Sun L. et Al., 2013

Dans cet article, les auteurs ont proposé un modèle de contrôle d'accès basé sur RBAC et le cryptage des données hébergées dans un Cloud. Dans ce modèle, au lieu de fournir les clés d'accès aux utilisateurs, les rôles reçoivent les clés d'autorisation pour accéder aux données. En outre, les données stockées chez le fournisseur de service sont associées à des opérations et des permissions (Figure 3.4). Les auteurs ont utilisé une méthode de dérivation de clé, qui utilise des jetons à la place des clés afin de réduire le nombre des clés et faciliter leur gestion. Dans cette technique, les clés restent secrètes mais les jetons sont publics. En plus, les auteurs ont proposé une méthode d'attribution et de dérivation de clé, qui utilise le principe suivant : chaque rôle est associé à une seule clé générée par le propriétaire des données et chaque donnée est cryptée par une seule clé [Sun L. et Al., 2013].

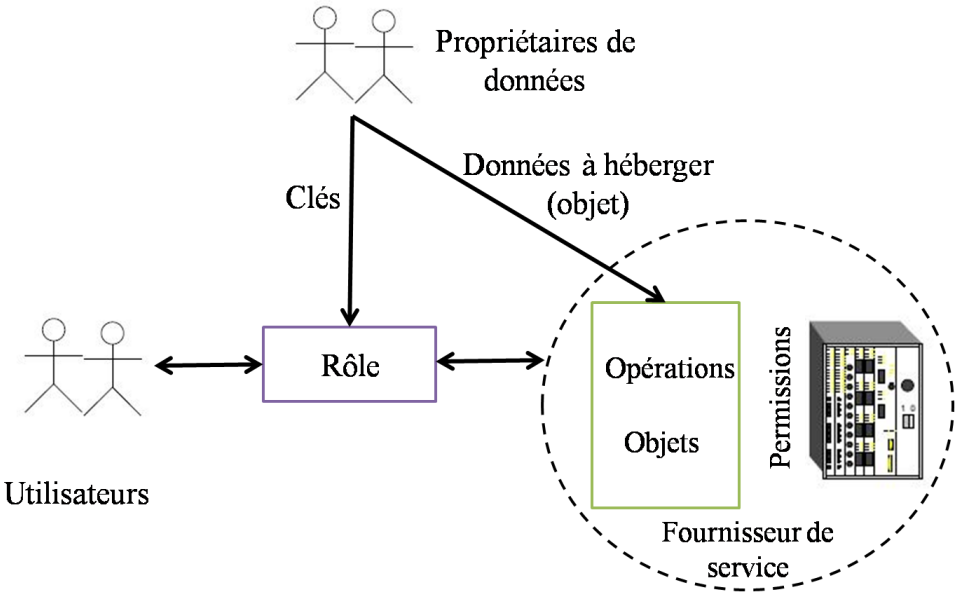

Figure 3. 4 Un modèle de contrôle d'accès basé sur RBAC dans un environnement de Cloud [Sun L. et Al., 2013]

3.3.4.7. Travail proposé par Zhou L. et Al., 2013

Dans cet article, les auteurs ont conçu un système de stockage du Cloud basé sur un RBAC sécurisé, où les politiques de contrôle d'accès utilisent une cryptographie basée sur le rôle. Le propriétaire des données crypte les données, de sorte que seuls les utilisateurs ayant les rôles appropriés, spécifiés par une règle RBAC puissent déchiffrer et afficher les données. Le rôle permet d'accorder et de révoquer les autorisations aux utilisateurs. Le fournisseur du Cloud (qui stocke les données) ne pourra pas voir le contenu des données s'il n'a pas le rôle approprié. En outre, le schéma proposé est capable de gérer les hiérarchies de rôles ainsi que les rôles héritant des autorisations d'autres rôles. Un utilisateur peut rejoindre un rôle après que le propriétaire ait chiffré les données pour ce rôle. L'utilisateur pourra alors accéder à ces données et le propriétaire n'aura pas besoin de recrypter les données. Un utilisateur peut être révoqué à tout moment et n'aura accès à aucune future donnée cryptée pour ce rôle. Avec le nouveau RBE (Role Based Encryption) proposé, la révocation d'un utilisateur d'un rôle n'affecte pas les autres utilisateurs. En outre, les auteurs ont externalisé une partie du décryptage dans le schéma vers le Cloud, dans lequel seuls les paramètres publics sont impliqués. En utilisant cette approche, le système RBE réalise un décryptage efficace du côté client. Les auteurs ont également utilisé la même stratégie d'externalisation pour améliorer l'efficacité de la gestion de l'adhésion des utilisateurs aux rôles, en ne faisant intervenir que des paramètres publics [Zhou L. et Al., 2013].

3.3.4.8 Travail proposé par Abbdal S.H., 2014

Les schémas qui ont été proposés auparavant pour assurer l'intégrité des données dans un environnement de Cloud, ont utilisé le principe d'une tierce partie auditrice (TPA), or cette partie peut être compromise par un attaquant. Dans cet article, les auteurs ont proposé un nouveau modèle TPA qui diffère des modèles précédents et surmonte le problème mentionné. Le système proposé comporte trois composants principaux: l'utilisateur du Cloud (propriétaire des données), le TPA et le serveur Cloud (CS), où seul l'utilisateur a l'autorité sur ses données et traite d'autres composants en tant que services. La proposition bénéficie de plusieurs avantages, tels que la préservation de la confidentialité et le secret des clés de session. Le schéma proposé combine les deux caractéristiques importantes des préoccupations d'efficacité et de sécurité: les fonctionnalités d'une TPA et il prend en charge la confidentialité du côté de l'utilisateur [Abbdal S.H., 2014].

3.3.4.9 Travail proposé par Khedkar S.V. and Gawande A.D., 2014

Dans cet article, les auteurs ont pris en considération deux issues de sécurité dans le stockage du Cloud. La première issue concerne la confidentialité des données et la deuxième issue concerne l'intégrité. Pour assurer la confidentialité, les auteurs ont utilisé l'algorithme RSA. Toutes les données sont cryptées avant de les héberger dans le Cloud. Pour l'intégrité des données, les auteurs ont proposé un algorithme de partitionnement avec l'application de MD5 (Message Digest 5). La fonction de partitionnement joue un rôle important dans ce travail. Elle permet de diviser les fichiers volumineux en plus petites parties afin de stocker les données de manière efficace et rapide, améliorant ainsi l'accès. Les données d'origine sont complexes et il est difficile de les stocker dans le Cloud. La fonction de partitionnement est donc utilisée pour faciliter le stockage dans le Cloud. En outre, le partitionnement s'effectue par ordre alphabétique en utilisant la méthode d'index, il récupère les deux premières lettres et les vérifie dans le dossier avec le présent ayant la même lettre. Si ce n'est pas le cas, il crée un dossier et stocke le fichier dans ce dossier. Les fichiers partitionnés sont cryptés, codés avec la clé publique et stockés dans le Cloud. Le partitionnement s'effectue automatiquement lorsque les données sont prêtes pour être stockées. Le fichier original est également reconstruit dès le besoin [Khedkar S.V. and Gawande A.D., 2014].

3.3.4.10 Travail proposé par Sulochana M. and Dubey O., 2015

Dans cet article, les auteurs ont proposé une architecture multi-Clouds, où les applications ainsi que les données sont stockées sur deux différents Clouds publics, tandis que l'administrateur est sur un Cloud privé. Dans cette architecture, les auteurs ont pris en considération trois issues de sécurité, à savoir : l'authentification, l'intégrité et la confidentialité des données. Pour l'authentification, les auteurs ont utilisé une méthode à un seul facteur de connaissance (identifiant et mot de passe). Après l'introduction des informations d'identification à travers le Cloud A, l'utilisateur demande à l'administrateur d'accéder au Cloud B. Seuls les utilisateurs authentifiés sont autorisés à accéder au stockage dans le Cloud. Un utilisateur peut uniquement accéder aux données après l'authentification correcte. Pour assurer la confidentialité ainsi que l'intégrité des données, les auteurs ont utilisé un algorithme de chiffrement ainsi que le partitionnement des données. Les données collectées à partir du Cloud A sont cryptées à l'aide de l'algorithme de chiffrement RSA. De plus, les données cryptées sont segmentées en blocs de données et stockées sur différents

emplacements dans le Cloud B. Chaque fragment est dans un format illisible pour garantir une sécurité des données de haut niveau [Sulochana M. and Dubey O., 2015].

3.3.4.11 Travail proposé par Nair N.K. and Navin K. S, 2015

Dans cet article, les auteurs ont proposé un mécanisme efficace d'authentification de groupe qui supporte la confidentialité, la fraicheur et l'authenticité de la clé dans un environnement de Cloud Computing. L'authentification de groupe est toujours effectuée entre les membres d'un groupe. Ce type d'authentification se produit en utilisant des messages cryptés pour partager le secret. Chaque utilisateur doit s'inscrire pour devenir membre d'un groupe particulier. Le gestionnaire de ce dernier assume la responsabilité d'approuver ou de rejeter l'utilisateur dans le groupe. Une fois que l'utilisateur a reçu l'approbation du responsable, il devient membre de ce groupe et il peut communiquer avec les autres membres. Chaque groupe possède une clé de groupe et cette dernière est générée lors de sa création. Le gestionnaire du groupe ne connaît que les clés de ce groupe et il donne une clé de session pour chaque membre de ce dernier utilisant la clé de groupe. Les clés de session sont utilisées par les membres pour partager des messages secrets. Des clés de session sont également utilisées pour chiffrer les messages secrets et plus tard, elles seront déchiffrées par d'autres membres en utilisant leurs clés de session. Lorsqu'un utilisateur quitte / rejoint un groupe, la clé du groupe est modifiée pour que les clés de session soient également modifiées. Le protocole de transfert de la clé de groupe dépend d'une seule entité de confiance qui est le centre de génération de clé (KGC : Key Generator Center). Ce centre se charge de choisir et distribuer les clés d'une manière sécurisée sur les membres du groupe. Chaque utilisateur doit s'enregistrer auprès du centre de clés pour s'inscrire au service de distribution de clés. Un double cryptage est réalisé en utilisant le schéma de chiffrement RSA [Nair N.K. and Navin K. S, 2015].

3.3.4.12 Travail proposé par Pawar P. and Sheikh R., 2016

Dans cet article, les auteurs ont proposé une architecture de sécurité pour un environnement de Cloud. Cette architecture a pris en considération toutes les issues de sécurité, à savoir : l'authentification, le contrôle d'accès, l'intégrité et la confidentialité. Pour l'authentification, l'architecture proposée vérifie en premier lieu le nom de l'utilisateur et le mot de passe, puis pour un Cloud privé, il y aura une vérification de l'adresse IP. Par contre, pour un Cloud public, il y aura une vérification de l'OTP. En outre, les auteurs ont utilisé un contrôle d'accès basé sur le modèle RBAC utilisant uniquement trois rôles : administrateur, gestionnaire et

client. Enfin, le MD5 a été utilisé pour garantir l'intégrité et l'algorithme de chiffrement RSA a été déployé pour garantir la confidentialité [Pawar P. and Sheikh R., 2016].

3.3.4.13 Travail proposé par Rucha D. et Al., 2017

Dans cet article, les auteurs ont proposé un mécanisme de contrôle d'accès qui est basé sur le cryptage d'attribut nommé KP-TSABE (Key-Policy Attribute Based Encryption with Time-Specified attributes). Dans le schéma KP-TSABE, chaque texte chiffré est étiqueté avec un intervalle de temps tandis que la clé privée est associée à un instant. Les données sensibles seront auto-détruites après une durée d'expiration spécifiée par l'utilisateur. Des comparaisons complètes des propriétés de sécurité indiquent que le schéma KP-TSABE répond aux exigences de sécurité et est meilleure que les autres schémas existants [Rucha D. et Al., 2017].

3.3.4.14 Travail proposé par Balusamy B. et Al., 2017

Dans cet article, les auteurs ont proposé un nouveau schéma nommé SCFAP (Storage Correctness and Fine-grained Access Provision) qui offre un contrôle d'accès à granularité fine ainsi qu'un stockage amélioré des données au sein du Cloud. Ce schéma permet d'offrir à l'utilisateur un accès exclusif grâce à l'utilisation d'une structure hiérarchique qui est une combinaison d'attributs uniques et communs. En outre, les auteurs ont utilisé le concept de système de jeton de permission, qui permet aux utilisateurs de vérifier l'exactitude des emplacements des données externalisées sans la récupération des fichiers respectifs. Les jetons ont été dérivés des métadonnées contenant l'emplacement du fichier. Cela réduit le temps de récupération de fichier associé à la demande d'accès aux données par les utilisateurs [Balusamy B. et Al., 2017].

3.4. Conclusion

Dans ce chapitre, nous avons commencé par définir les problèmes de sécurité introduits par le Cloud Computing. Ensuite, nous avons fait un état de l'art présentant tout ce qui a été fait pour résoudre les différentes issues de sécurité dans un environnement de Cloud. Ces travaux ont été classés selon l'issue traitée telle que : l'authentification, le contrôle d'accès, l'intégrité, la confidentialité et les issues multiples. Une comparaison plus détaillée entre ces travaux sera réalisée dans le chapitre suivant.

Chapitre IV : Une étude comparative entre la sécurité des grilles et celle du Cloud

4.1. Introduction

Dans les chapitres précédents, nous avons expliqué les différents travaux qui ont été proposés pour améliorer la sécurité dans les grilles et dans le Cloud Computing. Dans ce chapitre, nous allons faire une étude comparative entre les différents travaux qui ont été proposés dans chaque classe des issues de sécurité présentées dans les chapitres précédents et cela dans les grilles ainsi que dans le Cloud Computing. Cette comparaison va se focaliser sur certains critères que nous jugeons importants. Par la suite, nous ferons une comparaison entre les différentes techniques proposées dans chaque classe des grilles avec celles proposées dans la même classe du Cloud.

4.2. Etat de l'art détaillé sur la sécurité dans les grilles vs le Cloud Computing

Dans notre première contribution, nous avons fait des comparaisons entre les différents travaux qui ont été présentés dans le chapitre 2 et le chapitre 3 et cela pour chaque classe des issues de sécurité suivantes : authentification, contrôle d'accès (autorisation), intégrité, confidentialité et issues multiples. Le but principal de cette comparaison était de connaitre les différentes techniques de sécurité employées dans les grilles puis celles utilisées dans le Cloud en se basant sur certains critères. Après avoir comparé les différentes méthodes utilisées pour résoudre une issue de sécurité dans les grilles, nous avons fait la même chose pour le Cloud Computing. Enfin, pour chaque issue de sécurité, nous avons comparé les techniques proposées dans les grilles avec celles utilisées dans le Cloud.

4.2.1. La sécurité dans les grilles

Dans cette partie, nous allons faire une étude comparative entre les différents travaux qui ont été proposés pour assurer la sécurité dans un environnement de grille. Cette comparaison va être réalisée entre les travaux de chaque classe en prenant en considération des critères spécifiques pour chacune de ces classes.

4.2.1.1. L'authentification

Les travaux utilisés pour l'étude comparative de cette partie ont été expliqués en détail dans le chapitre 2, les critères de comparaison utilisés sont : la technique d'authentification, les avantages et la validation de la contribution.

☒ Satisfait ● Non satisfait

Articles	Années	Techniques d'authentification	avantages	Validation
Qiang W. and Konstantinov A	2010	▪ Identifiant/ mot de passe ▪ Fédération d'identité ▪ Service web ▪ SSO	▪ La mise à l'échelle ▪ La compatibilité avec les applications qui nécessite X.509	☒
Chen M. et Al.	2010	▪ Un protocole de négociation de clé authentique basé sur une signature sans certificat	▪ Sécurisé et efficace	☒
Hedayati M. et Al.	2010	▪ Une clé publique secrète basée sur l'identité	▪ Bien sécurisée contre les attaques de vol de mot de passe en ligne et les autres attaques	☒
Bhowmick A. et Al.	2012	▪ Mot de passe ▪ Carte à puce ▪ Signature RSA	▪ Simple et facile à utiliser	☒
Kazemi A.	2014	▪ Un mot de passe à usage unique (OTP : One Time Password)	▪ Chaque utilisateur peut avoir son propre modèle logiciel	●
Nandakumar V.	2014	▪ Partage de clé	▪ Une clé de 128 bits est difficile à tracer ▪ Sa formation de 3 parties la rend impossible à casser	●

Tableau 4. 1 Comparaison entre les différents travaux proposés pour résoudre l'issue d'authentification dans les grilles

Selon le tableau 4.1, les différents travaux qui ont été présentés pour résoudre l'issue d'authentification dans un environnement de grille, ont utilisé une authentification à un seul facteur, à deux facteurs ou les signatures. Dans [Kazemi A., 2014], les auteurs ont proposé une authentification à un seul facteur qui représente un mot de passe à usage unique, ce mot

de passe est divisé en deux parties, la première qui est un code d'identification fixe et la seconde est de nature variable (temps, nombre aléatoire, compteur, … etc). L'avantage de cette contribution réside dans le fait que même lorsqu'un attaquant intercepte le mot de passe, il ne pourra pas l'utiliser pour voler l'identité d'un utilisateur légitime. Nous pouvons dire également que l'authentification traditionnelle basée sur un mot de passe statique n'a pas été utilisée dans les grilles car elle est considérée comme inadéquate pour un tel environnement qui est dynamique et gérant un grand nombre d'utilisateurs. Une authentification à deux facteurs a été utilisée dans [Bhowmick A. et Al., 2012], où les auteurs ont combiné un facteur de connaissance (mot de passe) avec un facteur de possession (carte à puce) pour assurer une authentification forte. Dans [Nandakumar V., 2014], les auteurs ont proposé une nouvelle clé partagée pour assurer une authentification mutuelle dans un environnement de grille. En outre, ils ont proposé un nouvel algorithme de chiffrement et de déchiffrement en utilisant une technique mathématique dans le partage de clé. Dans [Qiang W. and Konstantinov A., 2010], les auteurs ont développé une nouvelle infrastructure à authentification unique utilisant une fédération d'identité standard. Cette infrastructure permet aux utilisateurs d'accéder aux grilles en utilisant leurs informations d'identification habituelles (identifiant/mot de passe) sans être débordé par la gestion et le maintien des certificats X.509. Le mécanisme proposé utilise un facteur de connaissance (le mot de passe), mais le facteur de possession (certificat X.509) a été remplacé par une fédération d'identité. Dans [Chen M. et Al., 2010], les auteurs ont proposé un protocole de négociation de clé authentique basé sur une signature sans certificat, le but principal de ces auteurs était d'éliminer d'une part les issues liées à la clé qui sont héritées par les schémas basés sur les identités. D'autre part, ils voulaient éliminer le fardeau de gestion des certificats dans les crypto systèmes basés sur PKI. Dans [Hedayati M. et Al., 2010], les auteurs ont présenté un protocole à clé publique secrète basé sur l'identité. Ce protocole permet l'échange de clés entre deux et trois parties pour assurer une authentification mutuelle dans un environnement de grille.

4.2.1.2. Le contrôle d'accès

Les travaux utilisés pour l'étude comparative de cette partie ont été expliqués en détail dans le chapitre 2, les critères de comparaison utilisés sont : les approches utilisées, le stockage et la gestion des politiques de sécurité ainsi que la validation de la contribution. Les modèles de contrôle d'accès proposés dans les grilles sont basés sur le modèle RBAC [Zhu X.J. et Al., 2010 ; Kaiiali M. et Al., 2010 ; Kaiiali M. et Al., 2013], le modèle ABAC [Zhao T. and

Shoubin D., 2010] ou autre [Gupta B. et Al., 2011]. Le modèle RBAC utilise les rôles des utilisateurs au lieu d'utiliser leurs identités pour accorder ou refuser l'accès. Tandis que le modèle ABAC est basé sur les attributs des utilisateurs (rôle, nom, email, …), c'est un modèle dynamique et plus convenable pour des environnements dynamiques et multi-domaines tels que les grilles. Selon les travaux cités dans le tableau 4.2, un processus de contrôle d'accès est divisé en deux parties : la première concerne le stockage et la gestion des politiques de sécurité. Accorder ou refuser l'accès d'un utilisateur à une ressource dépend d'une politique de sécurité. Cette dernière se compose d'un ensemble de règles de sécurité.

☒ Satisfait ● Non satisfait

Le modèle de contrôle d'accès	L'article	L'année	Les approches utilisées	Le stockage et la gestion des politiques de sécurité	La validation
RBAC	Zhu X.J. et Al.	2010	▪ Confiance ▪ Tâche ▪ Condition	●	☒
	Kaiiali M. et Al.	2010	▪ BFA ▪ PCM ▪ HCM	☒	☒
	Kaiiali M. et Al.	2013	▪ GAG ▪ XACML	☒	☒
ABAC	Zhao T. and Shoubin D.	2010	▪ Facteur de confiance ▪ XACML	●	☒
Autres	Gupta B. et Al.	2011	▪ Une architecture de confiance multi-agents	●	☒

Tableau 4. 2 Comparaison entre les différents travaux proposés pour résoudre l'issue du contrôle d'accès dans les grilles

L'une des problématiques qui peut influencer l'efficacité d'un modèle de contrôle d'accès représente la manière dont sont stockées ces règles de sécurité et comment sont elles gérées ? La seconde partie prend en charge le processus de contrôle d'accès lui-même, c'est-à-dire répondre aux requêtes d'accès efficacement. Dans [Kaiiali M. et Al., 2010], les auteurs ont utilisé une approche à force brute (BFA : Brute Force Approach) pour représenter les politiques de sécurité. Ce mécanisme nécessite la vérification de toutes les politiques de

sécurité pour trouver le groupe de ressources auxquelles l'utilisateur peut accéder, ce qui donne une répétition énorme. Ensuite, les auteurs ont introduit le mécanisme de groupement primitif (PCM : Primitive Clustering Mechanism) où ils ont regroupé les ressources qui ont la même politique de sécurité afin de réduire la redondance. Le mécanisme PCM élimine la répétition de la vérification des politiques de sécurité identiques mais il ne peut pas éliminer la répétition de la vérification des règles de sécurité identiques. Pour éviter cette répétition, les auteurs ont proposé le mécanisme de groupement hiérarchique (HCM : Hierarchical Clustering Mechanism). Ce dernier considère les informations des nœuds parents du PCM comme données pour générer un groupement hiérarchique de ces nœuds parents selon leurs règles de sécurité partagées. Dans [Kaiiali M. et Al., 2013], les auteurs ont trouvé que HCM réduit la répétition mais ne l'élimine pas totalement, ils ont proposé un graphe d'autorisation dans les grilles (GAG : Grid Authorization Graph) qui ajoute des arcs liant les nœuds redondant dans HCM. GAG permet d'éliminer la redondance lors de la vérification des règles de sécurité et permet de représenter les politiques de sécurité basées sur le « OU ». L'outil XACML a été utilisé dans [Kaiiali M. et Al., 2013, Zhao T. and Shoubin D., 2010] pour implémenter le modèle de contrôle d'accès. Cet outil peut être utilisé pour l'implémentation du modèle RBAC ainsi que du modèle ABAC. XACML représente également un langage de spécification de politiques de sécurité, il est générique, standard, distribué et puissant. Le modèle de contrôle d'accès présenté dans [Gupta B. et Al., 2011], n'est basé ni sur le modèle RBAC ni sur ABAC. Il utilise une architecture de grille multi-agents où chaque élément de la grille (utilisateur ou ressource) à son propre agent. Ce dernier reçoit un ensemble des besoins de l'utilisateur tels que : le coût, le temps, la réputation minimum du fournisseur de la ressource, … etc. Ensuite, l'expérience des agents pairs de confiance est utilisée pour sélectionner un ensemble de fournisseurs de services de confiance.

4.2.1.3. L'intégrité

Cette issue de sécurité n'a pas été beaucoup prise en considération par les recherches actuelles qui ont été présentées dans un environnement de grille. Nous avons trouvé un seul travail qui a été présenté dans la section 2.3.2 du chapitre 2. Dans ce travail, les auteurs ont utilisé le partitionnement des données pour éviter qu'elles soient alternées.

4.2.1.4. La confidentialité

Cette issue de sécurité n'a pas été largement considérée par les chercheurs car dans les contributions actuelles sur la sécurité des grilles, nous avons trouvé un seul travail dans lequel

les auteurs ont implémenté quelques algorithmes cryptographiques connus pour assurer la confidentialité dans un environnement de grille. Ce travail a été présenté en détail dans la section 2.3.3 du chapitre 2.

4.2.1.5. Issues multiples

Dans cette partie, nous allons comparer les différents travaux qui ont pris en considération plusieurs issues de sécurité en même temps dans un environnement de grille. Ces travaux ont été présentés en détail dans la section 2.3.4 du chapitre 2.

☒ Satisfait ● Non satisfait

Articles	Année	Techniques	Issues de sécurité	Validation
Sudalai Muthu T. et Al.	2010	▪ La cryptographie symétrique ▪ Le hachage ▪ Algorithme de dispersion d'information ▪ PKI	▪ Contrôle d'accès ▪ Intégrité ▪ Confidentialité ▪ Authentification	☒
Razieh M. et Al.	2010	▪ Authentification mutuelle ▪ Protocole d'échange de clés basé sur l'identité	▪ Authentification ▪ Confidentialité	☒
Rajesh I. and Sivakumar G.,	2010	▪ Communication de groupe sécurisée ▪ SSO ▪ SSL/TLS ▪ PKI ▪ Gestion de clé de groupe	▪ Authentification ▪ Contrôle d'accès	☒
Ashrafijoo B. et Al.	2010	▪ Théorie de la probabilité ▪ Processus aléatoire	▪ Confiance ▪ Réputation	☒
Khider H. et Al.	2010	▪ AAproxy ▪ XACML ▪ SAML	▪ Authentification ▪ Contrôle d'accès	☒
Jaspher G.W.K. et Al.	2011	▪ RBAC ▪ SAML/ XACML ▪ Chaine secrète plus données biométriques	▪ Authentification ▪ Contrôle d'accès	●
Kumari A.K. et Al.	2011	▪ Communication de groupe sécurisée ▪ Distribution de clé ▪ Communication VO-VO	▪ Authentification ▪ Contrôle d'accès	☒

Tableau 4. 3 Comparaison entre les différents travaux présentés dans un environnement de grille

Un processus de contrôle d'accès commence par la phase d'identification où l'utilisateur introduit ses informations. La prochaine phase représente l'authentification qui consiste à

vérifier que l'utilisateur est bien celui qu'il prétend être. Enfin, il y'a la phase de contrôle d'accès lui-même ou bien l'autorisation. Dans cette dernière phase, il y'aura une vérification des droits d'accès, en d'autres termes, c'est la phase où on essayera de vérifier si l'utilisateur a bien le droit d'accéder à la ressource demandée. Cette vérification est faite en se basant sur la politique de contrôle d'accès du modèle. Selon le tableau 4.3, certains travaux ont pris en considération le processus du contrôle d'accès en entier (authentification + autorisation) [Rajesh I. and Sivakumar G., 2010 ; Sudalai Muthu T. et Al., 2010 ; Khider H. et Al., 2010 ; Jaspher G.W.K. et Al., 2011 ; Kumari A.K. et Al., 2011]. Dans [Sudalai Muthu T. et Al., 2010], les auteurs ont proposé un mécanisme de contrôle d'accès qui assure deux métriques de sécurité, à savoir : l'intégrité (Hachage) et la confidentialité (chiffrement). Tandis que dans [Razieh M. et Al., 2010], les auteurs ont pris en considération l'issue d'authentification avec celle de la confidentialité. Enfin, d'après cette analyse, nous pouvons dire que le contrôle d'accès (authentification + autorisation) est l'issue de sécurité la plus ciblée par les chercheurs dans un environnement de grille.

4.2.2. La sécurité dans le Cloud Computing

Dans cette partie, nous allons faire une comparaison entre les différents travaux qui ont été présentés pour assurer la sécurité dans un environnement de Cloud. Cette comparaison sera faite entre les travaux de chaque groupe des classes suivantes : authentification, contrôle d'accès (autorisation), intégrité, confidentialité et issues multiples. Pour chaque classe, nous prenons en considération les mêmes critères de comparaison utilisés pour la comparaison des travaux de cette même classe dans les grilles.

4.2.2.1. L'authentification

Les travaux utilisés pour la comparaison ont été expliqués en détail dans la section 3.3.1.1 du chapitre 3. Les critères de comparaison utilisés sont : la technique d'authentification, les avantages et la validation de la contribution. Nous avons déployé les mêmes critères de comparaison utilisés dans la section 4.2.1.1 pour pouvoir comparer à la suite, les techniques d'authentification dans les grilles avec celles proposées dans le Cloud. Comme le montre le tableau 4.4, les recherches actuelles ont proposé plusieurs techniques d'authentification dans un environnement de Cloud. Elles se sont limitées à la deuxième phase du contrôle d'accès qui est l'authentification. Dans [Dinesha H.A. and Agrawal V.K., 2012], les auteurs ont présenté une authentification à plusieurs niveaux qui utilise un facteur de connaissance (mot

de passe). Les niveaux utilisés sont : le niveau utilisateur, le niveau équipe et le niveau organisation. Dans [Velciu M.A. et Al., 2014], les auteurs ont proposé une authentification à un seul facteur qui est biométrique (la voix), ce facteur a été combiné avec la cryptographie afin de garantir une authentification forte.

Dans [Sarvabhatla M. et Al., 2014], les auteurs ont proposé une authentification à deux facteurs où le premier représente un facteur de connaissance (mot de passe) et le second est un facteur de possession (carte à puce), ces deux facteurs ont été combinés avec la stéganographie et le chiffrement pour assurer une authentification solide. Dans [Singh A. and Chatterjee K., 2015], les auteurs ont présenté une authentification à deux facteurs. Le premier représente un facteur de connaissance (mot de passe) et le second représente un facteur de savoir faire (activité prédéterminée sur un écran tactile). Dans [Al-Attab B.S. and Fadewar H.S., 2016], les auteurs ont également utilisé une authentification à deux facteurs. Le premier représente aussi un mot de passe (facteur de connaissance) mais le second est un jeton USB (facteur de possession). Les auteurs ont utilisé une fonction de hachage pour le mot de passe afin de garantir sa sécurité. En outre, ils ont utilisé l'algorithme d'échange de clé Diffie-Helleman pour assurer une authentification mutuelle. Dans [Mansour A. and Sadik M., 2015], les auteurs ont proposé une authentification à trois facteurs. Le premier représente un facteur de connaissance (mot de passe), le second est un facteur de possession (jeton) et le troisième est un facteur biométrique. L'un des points positifs de cette proposition est que le dernier facteur biométrique est multimodal, c'est-à-dire les auteurs ont utilisé plusieurs types de biométries et ils ont fait une fusion entre ces dernières. Enfin, dans [Talkhaby H.R. and Parsamehr R., 2016], les auteurs ont proposé une authentification avec le protocole Kerberos 5, cette dernière est considérée comme une authentification avec une tierce partie. En outre, les auteurs ont utilisé les empreintes digitales de l'utilisateur (facteur biométrique) ainsi que le protocole d'échange de clés Diffie-Helleman combiné avec la signature DSA (Digital Signature Algorithm) pour assurer une authentification mutuelle forte.

☒ Satisfait ● Non satisfait

Articles	Année	Technique d'authentification	avantages	Validation
Dinesha H.A. and Agrawal V.K.	2012	▪ Mot de passe à plusieurs niveaux	▪ L'utilisateur a le fardeau de connaitre un seul mot de passe	●
Velciu M.A. et Al.	2014	▪ Mécanisme d'authentification basé sur la voix	▪ Combinaison des biométries avec des clés cryptographiques pour résoudre les vulnérabilités des systèmes cryptographiques traditionnels	☒
Sarvabhatla M. et Al.	2014	▪ Mot de passe ▪ Carte à puce ▪ Stéganographie ▪ Partage de clé	▪ Réponse rapide aux utilisateurs ▪ Résistant à toutes les attaques cryptographiques majeures	☒
Singh A. and Chatterjee K.	2015	▪ Mot de passé ▪ Une séquence d'une activité prédéterminée sur un écran virtuel	▪ Pas besoin de logiciel et de matériel supplémentaire	☒
Mansour A. and Sadik M.	2015	▪ Mot de passe ▪ Jeton ▪ Biométrie multimodale	▪ Elimination des inconvénients de la biométrie uni-modale	●
Al-Attab B.S. and Fadewar H.S.	2016	▪ Mot de passe ▪ Jeton USB ▪ Fonction de hachage ▪ Authentification mutuelle ▪ Echange de clé Diffie-Hellman	▪ Le choix et la mise à jour du mot de passe sont gratuits ▪ Le blocage du jeton USB est gratuit en cas de perte	☒
Talkhaby H.R. and Parsamehr R.	2016	▪ Protocole Kerberos 5 ▪ Algorithme d'échange de clé Diffie-Hellman-DSA ▪ Empreintes digitales de l'utilisateur ▪ Authentification mutuelle	▪ Résolution de l'issue de vol de mot de passe ▪ La non répudiation est assurée grâce à la biométrie	☒

Tableau 4. 4 Comparaison entre les différents travaux proposés pour résoudre l'issue d'authentification dans le Cloud Computing

4.2.2.2. Le contrôle d'accès

Les travaux utilisés pour l'étude comparative de cette partie, ont été expliqués en détail dans la section 3.3.1.2 du chapitre 3, les critères de comparaison utilisés sont : les approches

utilisées, le stockage et la gestion des politiques de sécurité ainsi que la validation de la contribution. Comme le montre le tableau 4.5, les modèles de contrôle d'accès proposés dans un environnement de Cloud sont basés soit sur le modèle RBAC, ABAC, ARBAC ou autre. Les modèles qui sont basés sur RBAC [Sun L. et Al., 2012 ; Chunlei W. et Al., 2012 ; Yue-qin F. and Yong-sheng Z., 2012], utilisent des politiques de sécurité qui sont centrées sur le rôle de l'utilisateur au lieu de son identité. Ceux qui sont basés sur le modèle ABAC [Dos Santos D. et Al., 2013 ; Aluvalu R.K. and Muddana L., 2016 ; Chen A. et Al., 2016 ; Khan F. et Al., 2016], utilisent des politiques de sécurité qui sont centrées sur les attributs des utilisateurs et leurs valeurs (nom, adresse mail, rôle).

Dans un environnement de Cloud, nous remarquons l'utilisation du nouveau modèle ARBAC [Mon E. and Naing T., 2011 ; Varadharajan V. et Al., 2015 ; Ayache M. et Al., 2015], qui est la combinaison des modèles RBAC et ABAC, où le rôle de l'utilisateur est considéré comme un attribut mais les politiques de sécurité sont basées sur ce dernier. Il existe également un autre type de modèles tels que [Auxilia M and Raja K., 2014], qui ne sont basés ni sur le rôle ni sur les attributs, ils ont utilisé d'autres approches pour assurer le contrôle d'accès, il s'agit de la classe « autre ». Nous remarquons également que l'utilisation du modèle RBAC, a toujours été combinée avec une nouvelle approche. Dans [Sun L. et Al., 2012], les auteurs ont utilisé des vocabulaires structurés et des ontologies. Dans [Chunlei W. et Al., 2012], les auteurs ont utilisé un rôle quantifié, une valeur de permission et une valeur de comportement. Dans [Yue-qin F. and Yong-sheng Z., 2012], les auteurs ont utilisé une valeur de réputation et un modèle de contrôle d'accès basé sur la tâche. Les modèles de contrôle d'accès basés sur ABAC ont utilisé une politique de risque ou un chiffrement des attributs (ABE : Attribute Based Encryption). La gestion des attributs peut être faite par une seule autorité ou par plusieurs comme dans [Khan F. et Al., 2016]. Les modèles de contrôle d'accès basés sur ARBAC, ont combiné les deux modèles (RBAC et ABAC) afin de profiter de la hiérarchie des rôles et de la fonction de conversion des rôles du modèle RBAC, ainsi que la souplesse et la flexibilité du modèle ABAC en même temps et cela dans un environnement dynamique tel que le Cloud Computing. L'outil XACML a été utilisé pour l'implémentation d'un modèle de contrôle d'accès basé sur ABAC [Dos Santos D. et Al., 2013] et pour l'implémentation d'un modèle de contrôle d'accès basé sur ARBAC [Ayache M. et Al., 2015]. Enfin, aucun des travaux cités dans le tableau 4.5 n'a pris en considération l'issue du stockage et de la gestion des politiques de sécurité dans un environnement de Cloud.

☒ Satisfait ● Non satisfait

Modèle	Article	Année	Approches	Stockage et gestion des politiques	Validation
RBAC	Sun L. et Al.	2012	▪ Un modèle de contrôle d'accès sémantique ▪ Les ontologies ▪ Des vocabulaires structurés et hétérogènes	●	●
	Chunlei W. et Al.	2012	▪ Un rôle quantifié ▪ Une valeur de permission ▪ Une valeur de comportement	●	☒
	Yue-qin F. and Yong-sheng Z.	2012	▪ Un contrôle d'accès basé sur la tâche ▪ Une valeur de réputation	●	☒
ABAC	Dos Santos D. et Al.	2013	▪ Politique de risque ▪ Politique ABAC ▪ XACML	●	☒
	Aluvalu R.K. and Muddana L.	2016	▪ Contrôle d'accès basé sur le risque ▪ Chiffrement basé sur les attributs (ABE)	●	☒
	Chen A. et Al.	2016	▪ Politique de risque ▪ Politique ABAC	●	☒
	Khan F. et Al.	2016	▪ Contrôle d'accès basé sur ABE ▪ Plusieurs autorités d'attributs	●	☒
ARBAC	Mon E. and Naing T.	2011	▪ Niveaux de sécurité des utilisateurs ▪ Niveaux de sécurité des objets	●	●
	Varadharajan V. et Al.	2015	▪ Arbre d'accès ▪ Centré sur le rôle ▪ Centré sur l'attribut	●	●
	Ayache M. et Al.	2015	▪ Un middleware ▪ XACML ▪ Acl ▪ Plusieurs fournisseurs ▪ OpenStack	●	☒
Autres	Auxilia M and Raja K.	2014	▪ Contrôle d'accès dynamique ▪ Contexte sémantique ▪ Les ontologies	●	☒

Tableau 4. 5 Comparaison entre les différents travaux proposés pour résoudre l'issue du contrôle d'accès dans le Cloud Computing

4.2.2.3. L'intégrité

Cette issue de sécurité n'a pas vraiment attiré l'attention des chercheurs dans les travaux récents sur la sécurité du Cloud Computing. Nous avons trouvé un seul travail qui a traité l'intégrité comme une issue à part entière, cet article a été expliqué en détail dans la section 3.3.2 du chapitre 3.

4.2.2.4. La confidentialité

Les travaux récents sur la sécurité du Cloud Computing, n'ont pas largement considéré cette issue. Nous avons trouvé un seul travail qui a traité cette problématique. La contribution proposée a été expliquée en détail dans la section 3.3.3 du chapitre 3.

4.2.2.5. Issues multiples

Dans cette partie, nous allons comparer les différents travaux qui ont pris en considération plusieurs issues de sécurité en même temps dans un environnement de Cloud Computing. Ces travaux ont été présentés en détail dans la section 3.3.4 du chapitre 3.

Comme le montre le tableau 4.6, l'issue du contrôle d'accès est l'issue de sécurité la plus ciblée par les chercheurs dans un environnement de Cloud. Cette issue a été traitée de différentes manières, certains travaux ont pris en considération uniquement sa deuxième phase, qui est l'authentification comme dans [Luo W. and Bai G., 2011 ; Sulochana M. and Dubey O., 2015, Nair N.K. and Navin K.S., 2015]. Dans le premier article, les auteurs ont assuré l'authentification en utilisant une signature RSA. En outre, ils ont utilisé une fonction de hachage pour assurer l'intégrité et un chiffrement symétrique pour assurer la confidentialité des données. Dans le deuxième article, les auteurs ont proposé une technique d'authentification à un seul facteur (identifiant/ mot de passe) dans une architecture multi-clouds. De plus, ils ont utilisé le partitionnement des données pour assurer leur intégrité et le chiffrement RSA pour assurer la confidentialité. Dans [Nair N.K. and Navin K.S., 2015], les auteurs ont proposé une technique d'authentification de groupe qui prend en charge la génération et le rafraichissement de clés. En outre, ils ont utilisé un double chiffrement pour assurer la confidentialité des données. Certains travaux ont tenu compte de la troisième phase du contrôle d'accès qui est l'autorisation comme dans [Sun L. et Al., 2013], où les auteurs ont assuré la confidentialité en utilisant un chiffrement pour la protection des informations sensitives. Enfin le processus de contrôle d'accès a également été pris en considération en entier (authentification + contrôle d'accès) dans [Chugh S. and Peddoju S.K., 2012 ; Gonzalez

☒ Satisfait ● Non satisfait

Article	Année	Technique	Issue de sécurité	Validation
Luo W. and Bai G.	2011	▪ Signature RSA ▪ Fonction de hachage ▪ Chiffrement symétrique	▪ L'authentification ▪ L'intégrité ▪ La confidentialité	●
Chugh S. and Peddoju S.K.	2012	▪ Mot de passe ▪ Table de permissions des utilisateurs ▪ Table d'appartenance des utilisateurs aux groupes ▪ Chiffrement des données	▪ Authentification ▪ Contrôle d'accès ▪ Confidentialité	●
Gonzalez N.M. et Al.	2013	▪ Architecture de gestion de certificats	▪ Authentification ▪ Contrôle d'accès	●
Sun L. et Al.	2013	▪ Chiffrement ▪ Protection des informations sensitives ▪ RBAC	▪ Contrôle d'accès ▪ Confidentialité	●
Liu X. et Al.	2013	▪ ABS ▪ CP-ABE ▪ XACML ▪ Chiffrement symétrique des données ▪ Plusieurs autorités d'attributs	▪ Authentification ▪ Contrôle d'accès ▪ Confidentialité	☒
Abbdal S.H.	2014	▪ Nouveau modèle de TPA ▪ Chiffrement progressif	▪ Intégrité ▪ Confidentialité	●
Khedkar S.V. and Gawande A.D.	2014	▪ Partitionnement des données ▪ MD5 ▪ Algorithme RSA	▪ Intégrité ▪ Confidentialité	●
Sulochana M. and Dubey O.	2015	▪ Architecture à plusieurs Cloud ▪ Identifiant/ mot de passe ▪ Partitionnement des données ▪ Chiffrement RSA	▪ Authentification ▪ Intégrité ▪ Confidentialité	●
Nair N.K. and Navin K.S.	2015	▪ Mécanisme d'authentification de groupe ▪ Génération et rafraichissement de clés ▪ Double chiffrement	▪ Authentification ▪ Confidentialité	☒
Pawar P. and Sheikh R.	2016	▪ OTP ▪ RBAC ▪ MD5 ▪ Algorithme RSA	▪ Authentification ▪ Contrôle d'accès ▪ Intégrité ▪ Confidentialité	☒

Tableau 4. 6 Comparaison entre les différents travaux présentés dans un environnement de Cloud Computing

N.M. et Al., 2013 ; Liu X. et Al., 2013 ; Pawar P. and Sheikh R., 2016]. L'intégrité et la confidentialité dans un environnement de Cloud ont été traitées ensemble dans [Abbdal S.H., 2014 ; Khedkar S.V. and Gawande A.D., 2014] et cela en utilisant un algorithme de chiffrement avec le partitionnement des données ou une tierce partie auditrice. Dans [Pawar P. and Sheikh R., 2016], les auteurs ont traité toutes les issues de sécurité présentées dans le chapitre 2 et le chapitre 3. Ils ont assuré l'authentification en utilisant un mot de passe à usage unique (OTP). En outre, le modèle RBAC a été utilisé pour assurer le contrôle d'accès. De plus, afin de garantir l'intégrité des données, les auteurs ont utilisé l'algorithme MD5 (Message Digest 5). Enfin, l'algorithme RSA a été utilisé pour assurer la confidentialité des données.

4.2.3. Comparaison entre la sécurité dans les grilles et la sécurité dans le Cloud Computing

Dans cette partie, nous allons comparer les différentes techniques proposées pour assurer la sécurité dans les grilles avec celles proposées dans un environnement de Cloud et cela pour chacune des issues de sécurité suivantes : l'authentification, le contrôle d'accès (autorisation) l'intégrité, la confidentialité et les issues multiples.

4.2.3.1. L'authentification

Selon le tableau 4.1 et le tableau 4.4, nous pouvons noter que les solutions proposées pour résoudre l'issue d'authentification n'ont pas utilisé une authentification traditionnelle basée sur un mot de passe statique et cela dans les deux environnements : grille et Cloud Computing. L'authentification à un seul facteur a été utilisée dans les grilles comme dans le Cloud. Par contre, dans un environnement de grille l'authentification s'est limitée à deux facteurs (type : connaissance et possession). Dans un environnement de Cloud, une authentification a deux facteurs a été utilisée avec un facteur de savoir faire, puis les auteurs ont proposé une authentification à trois facteurs où ils ont utilisé un facteur de connaissance, un facteur de possession et un facteur biométrique. L'authentification utilisant la biométrie a été largement utilisée dans le Cloud contrairement aux travaux qui ont été présentés dans un environnement de grilles. Ce type d'authentification a utilisé un seul type de biométrie (voix, empreintes digitales) ou plusieurs types en même temps (multimodale) qui seront fusionnés en utilisant des algorithmes spécifiques. En outre, l'authentification à plusieurs niveaux a été utilisée dans un environnement de Cloud mais pas dans un environnement de grille. Nous

remarquons également que la plupart des méthodes utilisées pour l'authentification dans le Cloud ont toujours fait appel à un mécanisme supplémentaire tel que : le chiffrement, le hachage, l'échange de clés, … Enfin, nous pouvons dire que les solutions proposées pour résoudre l'issue d'authentification dans un environnement de Cloud sont plus sophistiquées que celles proposées pour résoudre la même issue dans un environnement de grille. Le terme « plus sophistiquées » désigne : l'effort supplémentaire de l'utilisateur, une complexité plus importante ou la nécessité d'un matériel plus coûteux (fusion des biométries).

4.2.3.2. Le contrôle d'accès

Les modèles de contrôle d'accès présentés dans un environnement de grilles ont été divisés selon le modèle utilisé en trois classes, à savoir : ceux basés sur le modèle RBAC, ceux basés sur le modèle ABAC et la dernière classe qui concerne les modèles qui ne sont basés ni sur RBAC ni sur ABAC. Par contre dans un environnement de Cloud, en plus des classes citées précédemment, il existe une nouvelle classe nommée ARBAC qui combine les deux modèles RBAC et ABAC. Cette classe permet d'une part, de garder l'importance du rôle de l'utilisateur par rapport à son identité avec le principe de la hiérarchie de rôles. D'autre part, elle permet de profiter de la flexibilité du modèle ABAC dans un environnement dynamique et distribué tel que le Cloud Computing. Le processus de contrôle d'accès (autorisation) comprend deux phases : la première prend en charge la manière avec laquelle sont stockées et gérées les politiques de sécurité. La deuxième phase prend en considération le contrôle d'accès lui-même, c'est-à-dire quels sont les droits d'accès pour un utilisateur particulier ? Selon le tableau 4.2 et le tableau 4.5, nous pouvons dire que les travaux présentés dans un environnement de grille pour résoudre l'issue du contrôle d'accès ont tenu compte des deux phases, par contre ceux présentés dans un environnement de Cloud n'ont pas pris en considération l'issue du stockage et de gestion des politiques de sécurité. Il n'y a aucun travail qui a mentionné les mécanismes de stockage des politiques de sécurité, comment sont-elles stockées ? Comment sont elles vérifiées ? Or que cette issue est très importante car un environnement de Cloud a généralement un grand nombre d'utilisateurs et un grand nombre de services, si les politiques de sécurité sont stockées et vérifiées efficacement, cela améliorera le processus entier (temps de réponse et faux positifs). Nous remarquons également que l'outil XACML a été utilisé pour l'implémentation des modèles de contrôle d'accès proposés dans les grilles et dans le Cloud Computing et cela dans le cas de RBAC, ABAC ou ARBAC. Tous les modèles basés sur RBAC dans un environnement de Cloud, ont

combiné un nouveau principe (Valeur de permission, valeur de comportement, valeur de réputation, chiffrement, les ontologies) avec la notion de rôle, ce qui rend la validation de ces modèles un peu plus compliquée. En outre, tous les modèles basés sur ABAC dans un environnement de Cloud, ont ajouté une politique de risque ou un chiffrement d'attribut à la politique ABAC. Enfin, tous les modèles de contrôle d'accès présentés dans un environnement de grille, ont été validés contrairement à ceux présentés dans un environnement de Cloud, ceci est dû à la complexité des techniques de contrôle d'accès proposées dans un environnement de Cloud par rapport à celles proposées dans un environnement de grille, c'est ce qui a pu entraver leur validation.

4.2.3.3. L'intégrité

Cette issue de sécurité n'a pas été largement considérée dans les travaux récents et cela dans les deux environnements grille et Cloud. Nous avons trouvé un seul travail qui a traité cette issue dans les grilles et un seul dans le Cloud Computing.

4.2.3.4. La confidentialité

Cette issue de sécurité n'a pas attiré l'attention des chercheurs actuellement et cela dans les deux environnements grille et Cloud. Nous avons trouvé un seul travail qui a traité cette issue dans les grilles et un seul dans le Cloud Computing.

4.2.3.5. Les issues multiples

Selon le tableau 4.6 et le tableau 4.3, nous pouvons dire que le contrôle d'accès est l'issue de sécurité la plus visée par les chercheurs et cela dans les deux environnements grilles et Cloud Computing. Mis à part l'étape d'identification qui consiste à l'introduction des informations d'identification par l'utilisateur, l'issue du contrôle d'accès peut être divisée en deux étapes importantes, à savoir : l'authentification de l'utilisateur, c'est-à-dire le processus qui vérifie que ce dernier est bien celui qu'il prétend être et la deuxième étape représente le contrôle d'accès lui-même (autorisation), c'est-à-dire le processus qui vérifie que l'utilisateur peut accéder uniquement aux services autorisés. Certains travaux dans les grilles comme dans le Cloud se sont limités à la première étape (Tableau 4.1 et Tableau 4.4), d'autres se sont limités à la deuxième étape (Tableau 4.2 et Tableau 4.5), et enfin y'en a ceux qui ont pris en considération le processus en entier. Cette issue est la plus importante car si un attaquant peut accéder à des services ou à des ressources non autorisés, il pourra compromettre la confidentialité et l'intégrité du système.

4.3.Conclusion

Dans ce chapitre, nous avons fait un état de l'art détaillé sur tout ce qui a été proposé pour résoudre la problématique de sécurité dans les grilles et dans le Cloud Computing. Pour chaque environnement, nous avons attribué les travaux que nous avons trouvés à l'une des classes suivantes, selon l'issue de sécurité traitée : l'authentification, le contrôle d'accès (autorisation), l'intégrité et la confidentialité. Nous avons attribué les travaux qui ont traité plusieurs issues de sécurité en même temps à la classe « issues multiples ». Après cette classification, nous avons fait une comparaison entre les travaux proposés pour résoudre la même issue dans un environnement de grille en se basant sur certains critères importants. La même chose a été faite pour les travaux de chaque issue dans un environnement de Cloud. Enfin, une comparaison entre les travaux de chaque issue des grilles a été faite avec ceux de cette même issue dans le Cloud Computing. Comme conclusion, nous pouvons dire que l'issue de sécurité la plus importante pour les deux environnements représente le contrôle d'accès. Ce processus peut être divisé en deux étapes qui sont l'authentification et le contrôle d'accès lui-même (autorisation). Pour cette dernière étape, elle peut également être divisée en deux parties, la première qui prend en considération le stockage et la gestion des politiques de sécurité et la deuxième qui essaye d'assurer un processus de contrôle d'accès rapide et efficace. La première phase a été largement considérée dans un environnement de grille, mais dans le Cloud Computing, aucun des travaux trouvés n'a géré cette sous-problématique. Dans nos travaux futurs, nous voudrons axer nos efforts sur l'issue de contrôle d'accès, d'abord nous voulons savoir si les mécanismes proposés pour stocker et gérer les politiques de sécurité dans les grilles peuvent avoir des améliorations. En outre, nous voulons explorer la façon dont sont stockées les politiques de sécurité dans un environnement de Cloud. Pour la deuxième phase, nous essayerons de proposer un modèle de contrôle d'accès rapide et efficace.

Chapitre V : Les différents modèles de contrôle d'accès proposés dans un environnement de grilles et de Cloud

5.1. Introduction

Dans le deuxième chapitre, nous avons attribué les différents travaux trouvés dans la littérature pour assurer la sécurité des grilles à différentes classes selon l'issue de sécurité traitée, à savoir : le contrôle d'accès qui englobe l'authentification et l'autorisation, l'intégrité et la confidentialité. Les travaux qui ont traité plusieurs issues de sécurité en même temps ont été attribués à la classe « issues multiples ». Dans le troisième chapitre, nous avons fait la même chose avec les différents travaux qui ont été présentés pour assurer la sécurité dans un environnement de Cloud. Dans le quatrième chapitre, nous avons fait une étude comparative entre les différentes techniques proposées pour résoudre chaque issue de sécurité et cela pour chaque environnement. Ensuite, nous avons fait une comparaison entre les différentes techniques proposées dans les grilles pour résoudre une issue particulière et celles proposées pour résoudre cette même issue dans le Cloud. C'est ce qui nous a permis de savoir que l'issue du contrôle d'accès représente la problématique la plus visée par les recherches actuelles car son importance est remarquable par rapport aux autres issues et cela dans les grilles comme dans le Cloud. En outre, nous avons pu savoir que cette issue peut être traitée de deux manières : la première consiste à proposer un mécanisme efficace de gestion et de stockage des politiques de sécurité, tandis que la deuxième se limite à proposer un modèle de contrôle d'accès qui gère efficacement les requêtes d'accès. Dans ce chapitre, nous allons présenter nos quatre contributions. Dans la première contribution, nous avons proposé un modèle de contrôle d'accès parallèle dans un environnement de grille inter-domaines. Dans la seconde proposition, nous avons présenté le graphe pondéré d'autorisation dans les grilles (WGAG). Dans la troisième contribution, nous avons proposé un graphe pondéré basé sur l'action pour l'autorisation dans les grilles (Action-WGAG). Enfin dans notre dernière contribution, nous avons présenté un modèle de contrôle d'accès à différents niveaux de données hébergées dans un environnement de Cloud

5.2. Un modèle de contrôle d'accès parallèle dans un environnement de grille inter-domaines

A partir de cette contribution, lorsqu'on parle de contrôle d'accès, il s'agit de la deuxième étape qui est l'autorisation. Dans cette contribution, nous nous sommes focalisés sur la phase

qui prend en charge le contrôle d'accès en tant que processus, c'est-à-dire est ce qu'un tel utilisateur a le droit d'accéder à une telle ressource dans un environnement de grille ? Cet environnement est souvent composé de plusieurs domaines administratifs où chaque domaine comprend un grand nombre de ressources et un grand nombre d'utilisateurs, ce qui donne un grand nombre de requêtes d'autorisation à traiter. En outre, contrôler l'accès à un tel système est une tâche difficile à gérer car le rôle d'un utilisateur dans un domaine A est vraiment différent du rôle qu'il pourra obtenir dans un domaine B. Cette issue nous pousse à chercher une certaine politique de conversion de rôles entre les domaines administratifs respectant la politique de sécurité de chaque domaine. Afin de faire face à cette problématique, nous avons présenté un modèle de contrôle d'accès parallèle dans un environnement de grille multi-domaines. Le modèle proposé représente une architecture XACML étendue qui est basée sur le modèle ARBAC avec une politique de conversion de rôles permettant la collaboration entre plusieurs domaines.

5.2.1. Comparaison entre les travaux existants

Dans cette partie, nous avons fait une comparaison entre les travaux existants en se basant sur les critères suivants : les approches utilisées, la validation de la contribution, la collaboration inter-domaines (CID), la réduction du temps de réponse à une requête d'accès (RTRRA). Ces travaux ont été présentés en détail dans le chapitre 2.

5.2.2. Analyse et synthèse

Après une analyse approfondie des travaux cités dans le tableau 5.1, nous pouvons dire que :

- Un modèle de contrôle d'accès est basé sur RBAC, ABAC, ARBAC ou autre
- Tous les modèles basés sur ABAC ont été validés, mais ceux basés sur RBAC ne l'ont pas tous été. Ce point montre la flexibilité du modèle ABAC par rapport au modèle RBAC dans un environnement dynamique et distribué tel que les grilles.
- La collaboration entre les domaines dans un environnement de grille a été prise en considération uniquement dans [Martino, L.D. et Al., 2008 ; Li N. et Al., 2015 ; Kaustav, R. and Avijit, B., 2012 ; Yue-qin F. and Yong-sheng Z., 2012 ; Dos Santos D. et Al., 2013 ; Alfieri R. et Al., 2003 ; Ceccanti A. et Al., 2015], ce qui montre la difficulté de la gestion de ce point sans violer les politiques internes de chaque domaine et la hiérarchie de rôles.

- La réduction du temps de réponse à une requête de contrôle d'accès est un critère très important car il influence l'efficacité du modèle. Aucun des travaux n'a pris en considération ce critère.
- Dans [Mon E. and Naing T., 2011], les auteurs ont combiné le modèle RBAC et le modèle ABAC pour profiter de la flexibilité du modèle ABAC et de l'importance du rôle d'un utilisateur dans une organisation au lieu de son identité ce qui est réalisable par le modèle RBAC.

☒ Satisfait ● Non satisfait

Modèle	Citation	Approches	Validation	CID	RTRRA
RBAC	Martino, L.D. et Al., 2008	▪ Un modèle de contrôle d'accès privé	☒	☒	●
	Li N. et Al., 2015	▪ Une architecture de rôle à deux tiers	☒	☒	●
	Kaustav, R. and Avijit, B., 2012	▪ Mécanisme de classement de rôles	●	☒	●
	Chunlei W. et Al., 2012	▪ Un rôle quantifié	☒	●	●
	Sun L. et Al., 2012	▪ Un modèle de contrôle d'accès sémantique	●	●	●
	Yue-qin F. and Yong-sheng Z., 2012	▪ Un modèle de contrôle d'accès basé sur la tâche	☒	☒	●
ABAC	Khaled R. et Al., 2015	▪ La sensibilité des objets	☒	●	●
	Chen A. et Al., 2016	▪ Contrôle d'accès basé sur le risque	☒	●	●
	Khan F. et Al., 2016	Un modèle de contrôle d'accès basé sur le chiffrement	☒	●	●
	Dos Santos D. et Al., 2013	▪ Un modèle de contrôle d'accès basé sur le risque	☒	☒	●
ARBAC	Mon E. and Naing T., 2011	▪ Degré de sensibilité des données	●	●	●
Autres	Alfieri R. et Al., 2003	▪ Organisation virtuelle	☒	☒	●
	Ceccanti A. et Al., 2015	▪ XACML	☒	☒	●

Tableau 5. 1 Comparaison entre les travaux existants (contribution 1)

5.2.3. Le paradigme du calcul parallèle

Le principe de calcul parallèle est un type de calcul dans lequel plusieurs calculs sont réalisés simultanément, partant du principe que les gros problèmes ont la possibilité d'être divisés en petits problèmes qui peuvent être résolus en même temps [Wikipedia, 2017]. Les recherches actuelles sur le paradigme de l'informatique parallèle, se concentrent sur l'analyse de performance en utilisant différents algorithmes mais n'abordent pas le problème de l'utilisation de ce paradigme dans différents domaines tels que: La sécurité, le Grid Computing, le Cloud Computing ... afin de pouvoir profiter du parallélisme dans tous les niveaux. Dans [Kulkarni P. and Pathare S., 2014], les auteurs ont analysé les performances des algorithmes parallèles par rapport aux algorithmes séquentiels, les résultats de simulation montrent que les algorithmes avec de petits ensembles de données donnent de bonnes performances lorsqu'ils sont exécutés séquentiellement. Si l'exécution parallèle est utilisée sur un grand ensemble de données, alors elle aura donné de meilleurs résultats que l'exécution séquentielle. C'est ce qui nous a conduit à introduire un modèle de contrôle d'accès parallèle dans un environnement de grille inter-domaines, le parallélisme sera utilisé entre deux processus différents, le premier prend la décision d'autorisation pour les requêtes d'accès inter-domaines et le second prendra en charge les décisions d'accès pour les demandes locales. De plus, dans [Kulkarni P. and Pathare S., 2014] les auteurs ont indiqué que le paramètre le plus important pour mesurer la performance du calcul parallèle est le temps d'exécution, c'est pourquoi les résultats de la simulation discutés dans les sections suivantes utiliseront ce paramètre.

5.2.4. Le mécanisme de conversion de rôle inter-domaines

Un environnement de grille se constitue souvent d'un ensemble de domaines administratifs, donc l'interopérabilité entre ces domaines doit être prise en considération. Le premier modèle qui a proposé un mécanisme de conversion de rôles entre plusieurs domaines était le modèle IRBAC 2000 [Al-Muhtadi J., 2000]. Cette fonction a été utilisée dans le modèle que nous proposons comme suit :

- Soit R_1= {ri| i=1…n} un ensemble des rôles du domaine AD_1
- Soit R_2= {ri| i=1…n} un ensemble des rôles du domaine AD_2
- Soit H_1 une hiérarchie de rôles du domaine AD_1 (Figure 5.1)
- Soit H_2 une hiérarchie de rôles du domaine AD_2 (Figure 5.1)

- H_1 et H_2 ont la même structure mais elles diffèrent dans leurs sémantiques
- $r_i > r_j$ veut dire que le rôle r_i est supérieur au rôle r_j
- Les deux domaines ont le rôle « invité ». Si un rôle n'est pas clair, il pourra être considéré comme le rôle « invité ».
- Soit R_1R_2 la conversion dynamique de rôles entre ceux du domaine AD_1 et ceux du domaine AD_2

Pour l'instant, la flèche discontinue allant du rôle $Directeur_{R2}$ de H_2 vers le rôle $Enseignant_{R1}$ dans H_1 veut dire que le rôle $Directeur_{R2}$ a été converti au rôle $Enseignant_{R1}$. On dénote cette fonction par $Directeur_{R2} \longrightarrow Enseignant_{R1}$ où la relation ' $\longrightarrow$ ' désigne 'converti à '. Ceci est équivalent à l'écriture (Directeur, Enseignant) appartient à R_2R_1.Il faut noter que le rôle $Administrateur_{R2}$ est sénior au rôle $Directeur_{R2}$ donc le rôle $Administrateur_{R2}$ peut utiliser la conversion du rôle $Directeur_{R2}$ d'où l'appartenance de (Administrateur, Enseignant) à R_2R_1implicitement. Enfin, nous pouvons dire que la relation ' $\longrightarrow$ ' est transitive. Selon la figure 5.1, l'ensemble R_2R_1 se compose des éléments suivants :

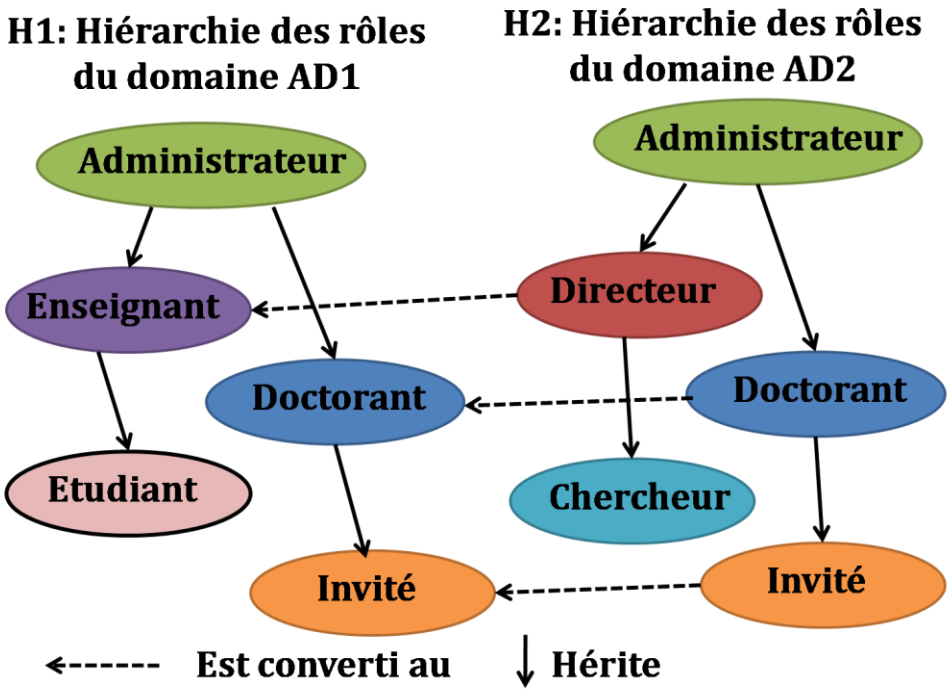

Figure 5. 1 La hiérarchie de rôles des domaines AD1 et AD2 [Al-Muhtadi J., 2000]

- R_2R_1 = {(Directeur, Enseignant) ; (Administrateur, Enseignant) ; (Doctorant, Doctorant) ; (Administrateur, Doctorant) ; (Invité, Invité) ; (Chercheur, Invité) ; (Doctorant, Invité) ; (Directeur, Invité) ; (Administrateur, Invité)}.

Pour offrir plus de flexibilité au modèle d'interopérabilité, des associations non transitives ont été introduites entre les rôles. Par exemple, dans la figure 5.2, l'association entre $Directeur_{R2}$ et $Enseignant_{R1}$ est représentée par une flèche discontinue avec le labelle 'NT' qui désigne que c'est une association non transitive. Elle sera dénotée $Directeur_{R2}$ NT ⟶ $Enseignant_{R1}$. Une association non transitive prévient les conversions implicites du domaine AD_2. Dans ce cas, (Directeur, Enseignant) appartient à R_2R_1 mais (Administrateur, Enseignant) n'appartient pas à R_2R_1. Selon la figure 5.2 l'ensemble R_2R_1 se compose des éléments suivants :

R_2R_1 = {(Directeur, Enseignant) ; (Doctorant, Doctorant) ; (Administrateur, Doctorant) ; (Invité, Invité) ; (Chercheur, Invité) ; (Doctorant, Invité) ; (Directeur, Invité) ; (Administrateur, Invité)}.

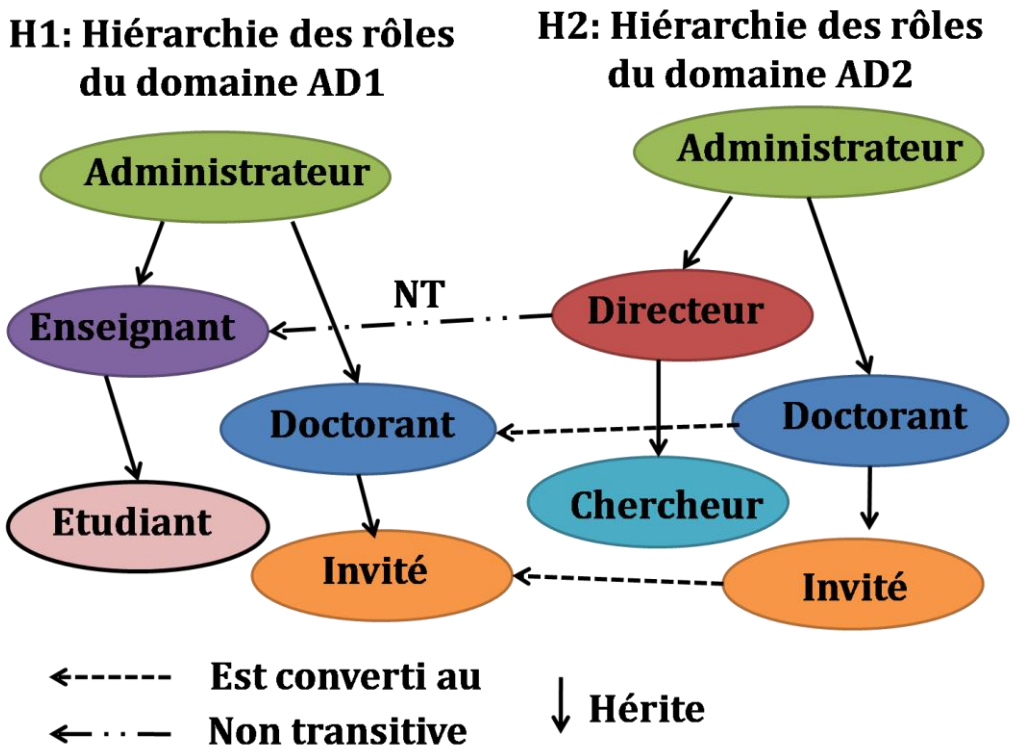

Figure 5. 2 Association non transitive pour respecter les politiques du domaine [Al-Muhtadi J., 2000]

Les résultats obtenus par ce modèle peuvent être mis dans une table (Tableau 5.2), cette table est nommée « table de conversion de rôles (RMT) ».

R_1	R_2
Directeur	Enseignant
Doctorant	Doctorant
Administrateur	Doctorant
…	…

Tableau 5. 2 Table de conversion des rôles (RMT)

5.2.5. Description du modèle de contrôle d'accès parallèle proposé

5.2.5.1. Motivation

La grille de calcul est l'application de plusieurs ressources provenant de plusieurs institutions pour résoudre un seul problème en même temps [Kiranjot K. and Anjandeep K. R., 2014]. Le contrôle d'accès à ces ressources est une question cruciale car non seulement les données doivent être sécurisées, mais aussi les ressources et les calculs. Une requête de contrôle d'accès générée par un utilisateur, peut cibler une ressource provenant du même domaine ou d'un domaine différent, mais le rôle d'un utilisateur dans un domaine diffère de manière significative dans un autre domaine. D'un autre côté, chaque domaine a sa propre politique de sécurité qui doit toujours être respectée. L'environnement de grille peut également être défini comme un grand système distribué contenant un grand nombre d'utilisateurs et de ressources, ce qui conduit à un grand nombre de demandes de contrôle d'accès à traiter. Comment un modèle de contrôle d'accès peut-il gérer un grand nombre de demandes tout en garantissant l'efficacité et un temps de réponse minimal? Comment gérer le mécanisme de conversion de rôles entre domaines sans violer les politiques de sécurité de chacun?

Le processus d'autorisation inter-domaines est un facteur essentiel dans la politique de contrôle d'accès de plusieurs domaines. En règle générale, un environnement de grille est composé de nombreux domaines et de sous-domaines jouant différents rôles et ayant différentes responsabilités. Chaque domaine a un administrateur qui entreprend l'ensemble du domaine comme l'ajout, la suppression et la modification des utilisateurs, l'attribution des rôles, la création de règles d'autorisation et leur stockage. Le rôle d'un nœud dans un domaine diffère de manière significative dans un autre domaine. Le besoin est d'avoir un certain modèle de contrôle d'accès qui permet à différents utilisateurs de différents domaines de partager leurs ressources et de collaborer ensemble pour résoudre différents problèmes, et ce, quel que soit le domaine auquel ils appartiennent. Comme l'illustre la figure 5.3, le modèle

proposé est composé de deux domaines AD_1 et AD_2, chaque domaine contient un ensemble d'utilisateurs, un ensemble de ressources et un serveur d'autorisation qui gère le processus de contrôle d'accès.

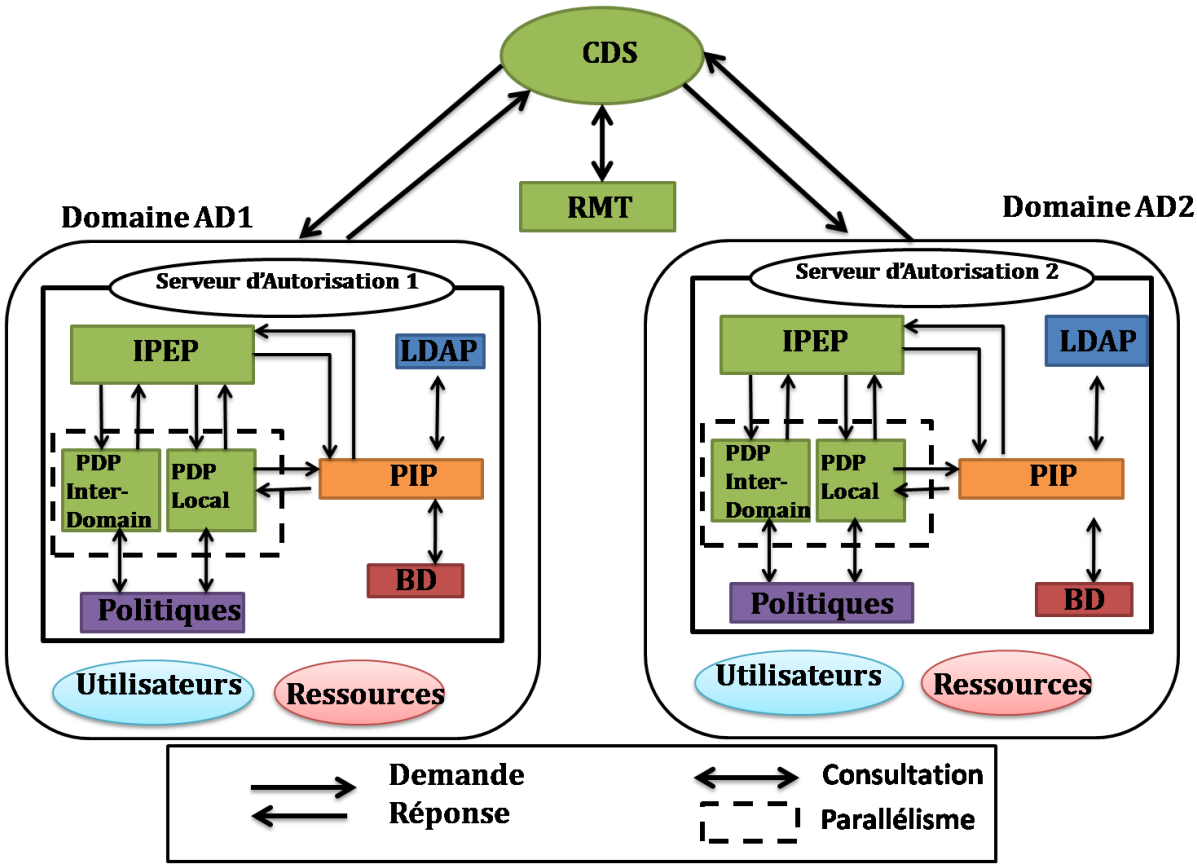

Figure 5. 3 L'architecture proposée

5.2.5.2. L'architecture d'autorisation XACML étendue

XACML définit un langage de politique qui utilise les attributs du demandeur, des ressources et de l'environnement [Rissanen E., 2017]. En outre, XACML fournit une architecture d'autorisation qui supporte le modèle ARBAC [Yuan, E. and Tong, J., 2005]. Pour répondre aux besoins d'un environnement de grille multi-domaines, nous avons étendu l'architecture XACML standard (chapitre 2). Notre architecture se compose des éléments suivants (Figure 5.3) :

- **Point d'application de la politique intelligent (IPEP : Intelligent Policy Enforcement Point)** : il reçoit les demandes d'autorisation de l'utilisateur, si la

demande vient d'un utilisateur du même domaine, il la transmettra au PDP local, sinon il l'enverra au PDP inter-domaines.

- **PDP inter-domaines** : il prend les décisions d'autorisation pour les requêtes générées par des utilisateurs appartenant à un autre domaine. Il évalue la requête selon la politique de sécurité, et rend la réponse à l'IPEP.
- **PDP local** : il prend les décisions d'autorisation pour les requêtes générées par des utilisateurs locaux, il demande les attributs manquants au PIP, puis évalue la requête et rend la réponse à l'IPEP.
- **Point d'information de la politique (PIP)** : il crée et gère les attributs des sujets, des ressources et de l'environnement. Dans le cas où le PDP évalue une requête d'accès et trouve des attributs manquants par rapport à la politique de sécurité, il demande ces attributs au PIP.
- **Politiques** : contient les politiques de sécurité du système.
- **Serveur inter-domaines (CDS : Cross Domain Server)** : il gère le contrôle d'accès inter-domaines entre le domaine AD_1 et le domaine AD_2 en utilisant une fonction de conversion de rôle et en transférant les informations des utilisateurs.
- **Table de conversion de rôle (RMT : Role Mapping Table) :** contient les résultats de la fonction de conversion de rôles entre le domaine AD_1 et le domaine AD_2 (générée dans la section 5.2.4).
- **LDAP :** contient toutes les informations sur les utilisateurs et les ressources du domaine [Rissanen E., 2017].
- **La base de donnés (DB) :** contient les informations sur les rôles des utilisateurs, la délégation de rôles, …

5.2.5.3. Cas d'une demande d'accès générée par un utilisateur d'un autre domaine

Lorsqu'un utilisateur U_1 du domaine AD_1 veut accéder à une ressource R_1 du domaine AD_2, le processus d'autorisation est comme suit (illustré dans la figure 5.4) :

1. L'utilisateur U_1 envoie une requête d'autorisation à la ressource R_1du domaine AD_2.
2. La ressource R_1 demande au serveur d'autorisation 2 la décision d'accès.
3. L'IPEP intercepte la requête, comme le demandeur est d'un autre domaine, l'IPEP du domaine AD_2 crée une requête d'information de l'utilisateur U_1 et l'envoie au serveur inter-domaines (CDS) (illustré dans la figure 5.5)

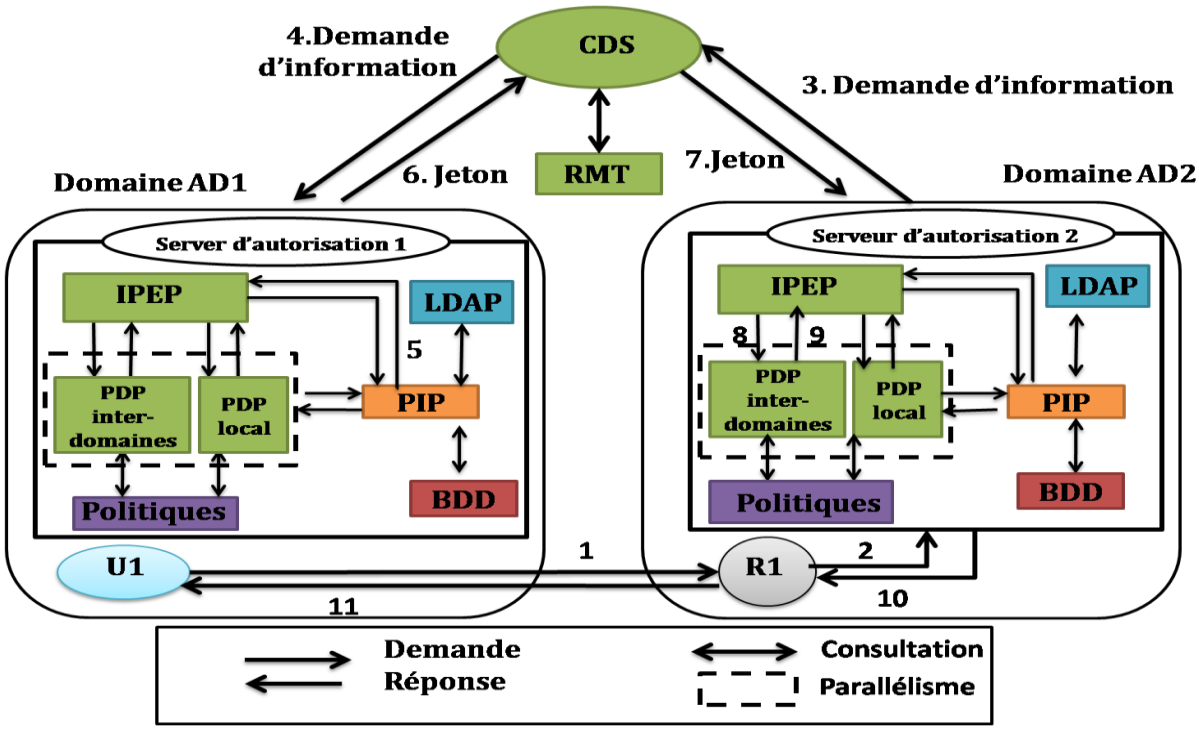

Figure 5. 4 Cas d'une demande d'accès générée par un utilisateur d'un autre domaine

4. Le CDS redirige la requête d'information vers le domaine de l'utilisateur (AD_1)
5. L'IPEP du domaine AD_1 intercepte la requête d'information, demande au PIP du domaine AD_1 les attributs concernant l'utilisateur U_1 tels que : le rôle. Puis l'IPEP crée un jeton contenant toutes ces informations (Illustré dans la figure 5.5).
6. Le jeton suit le même chemin dans le sens contraire, lorsque le CDS reçoit le jeton, il prend le rôle (local au domaine AD_1) de l'utilisateur et le convertit au rôle correspondant du domaine AD_2 en utilisant la table de conversion de rôle (RMT) (illustré dans le tableau 5.2).
7. Le CDS renvoie le jeton avec la nouvelle valeur de rôle comme réponse à la requête d'information de l'IPEP.
8. L'IPEP du domaine AD_2 reçoit finalement le jeton qui contient les informations concernant l'utilisateur, il envoie alors une requête d'autorisation au PDP inter-domaines.
9. Le PDP inter-domaines évalue la politique de sécurité puis envoie la décision d'autorisation à l'IPEP.
10. L'IPEP transfère la décision d'autorisation à R_1.

11. R_1 répond U_1 si l'accès est accepté ou refusé.

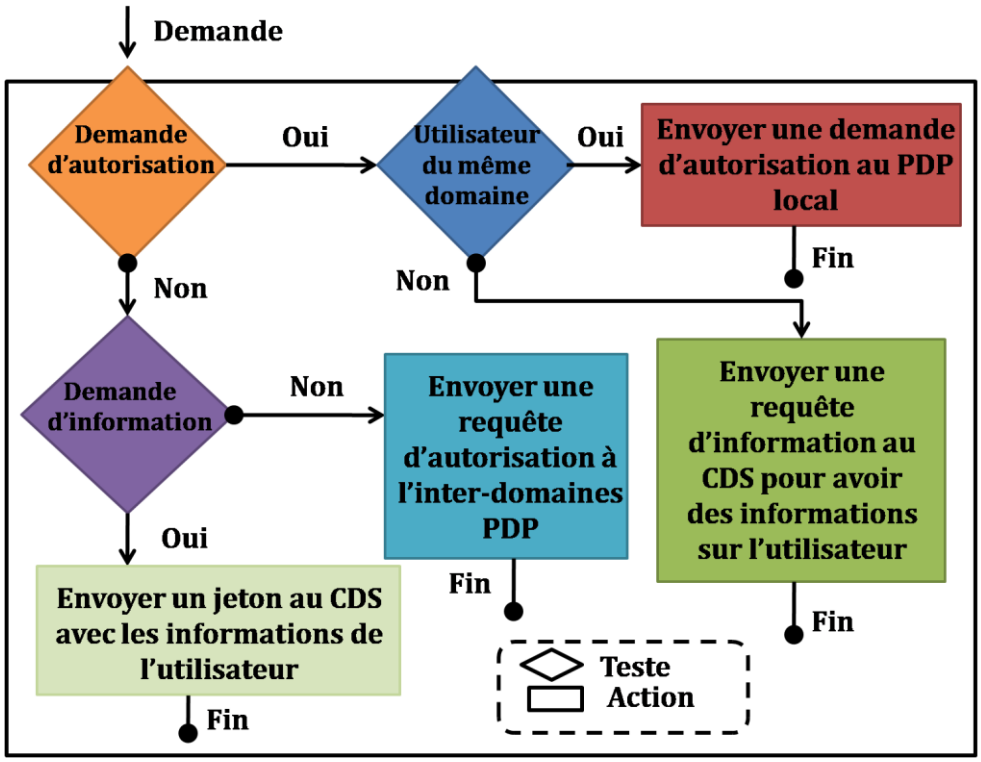

Figure 5. 5 Algorithme de l'IPEP

5.2.5.4. Cas d'une demande d'accès générée par un utilisateur du même domaine

Lorsqu'un utilisateur U_1 du domaine AD_1, veut accéder à une ressource R_1 du domaine AD_1, le processus d'autorisation est comme suit (illustré dans la figure 5.6) :

1. L'utilisateur U_1 envoie une requête d'autorisation à la ressource R_1du domaine AD_1.
2. La ressource R_1 demande au serveur d'autorisation 1 la décision d'accès.
3. L'IPEP intercepte la requête, comme l'utilisateur est du même domaine, il envoie une requête d'autorisation au PDP local (figure 5.5).
4. Le PDP local demande les attributs manquants de l'utilisateur au PIP, puis évalue la politique.
5. La décision d'autorisation est envoyée à l'IPEP.
6. L'IPEP transfert la décision d'autorisation à R_1.
7. R_1 répond U_1 si l'accès est accepté ou refusé.

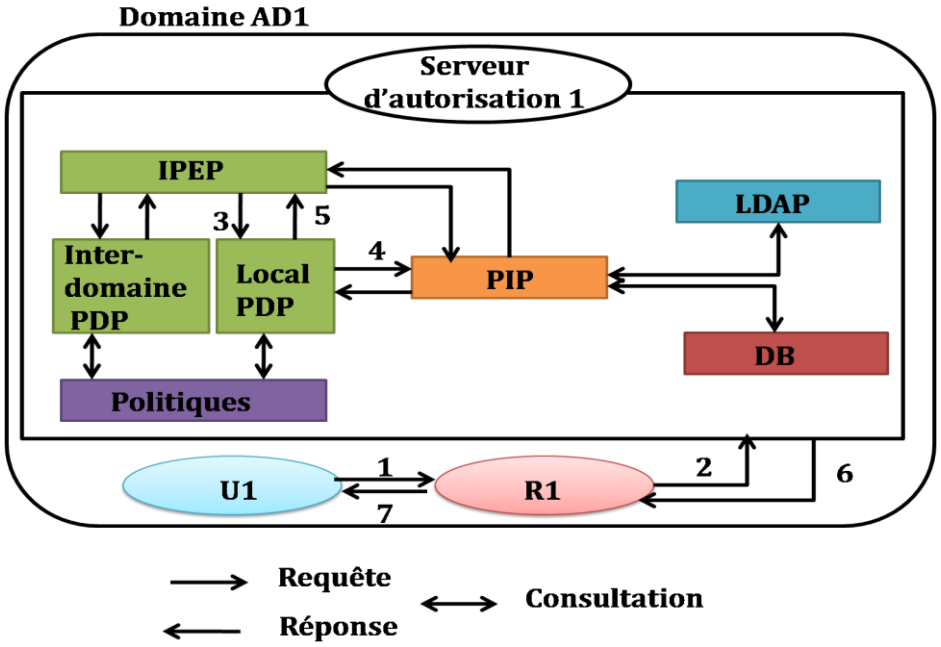

Figure 5. 6 Demande d'accès générée par un utilisateur du même domaine

5.2.6. Simulation et résultats

Pour tester l'efficacité du modèle proposé, nous avons développé un simulateur d'autorisation parallèle inter-domaines (CD-PAS : Cross Domain Parallel Access control Simulator) dans un environnement de grille. Ce simulateur est une application basée sur le langage C#. Le simulateur CD-PAS utilise le principe de table de sécurité qui a été introduit par [Kaiiali, M. et Al., 2013] comme une matrice (n× m) où n représente le nombre de ressources et m représente le nombre de règles de sécurité. Chaque entrée (i,j) à la table est égale soit à 0 ou à 1 indiquant que la $j^{ième}$ règle de sécurité appartient ou non à la politique de sécurité de la $i^{ème}$ ressource (un exemple est illustré dans le tableau 5.3). Le simulateur CD-PAS a utilisé deux tables de sécurité, la première pour représenter les politiques de sécurité du domaine AD_1 et la deuxième pour représenter les politiques de sécurité du domaine AD_2. Chaque table de sécurité contient 200 ressources.

Ressources/ Règles de sécurité	Sr_1	Sr_2	Sr_3
R_1	0	1	1
R_2	1	0	1
R_3	1	1	0
R_4	0	0	1

Tableau 5. 3 Exemple d'une table de sécurité

Les politiques de sécurité des ressources, les rôles des utilisateurs, des demandes d'accès par des utilisateurs à des ressources appartenant au même domaine, des demandes d'accès par des utilisateurs à des ressources appartenant à d'autres domaines ainsi qu'une table de conversion de rôle ont été générées aléatoirement en utilisant la fonction aléatoire du langage C#. Puis nous avons programmé les deux cas suivants :

- **Cas 1** : Un modèle d'autorisation avec un seul point de décision de politique (PDP) qui gère des demandes d'autorisation à des ressources appartenant au même domaine de l'utilisateur et celles générées par des utilisateurs appartenant à d'autres domaines (programmation séquentielle).
- **Cas 2** : Un modèle d'autorisation avec deux points de décision de politique (threads), le premier prend en charge les requêtes générées par des utilisateurs locaux et le deuxième gère les requêtes d'autorisation générées par des utilisateurs appartenant à un autre domaine, les deux points de décision (PDPs) travaillent en parallèle.

Pour un environnement de grille contenant deux domaines AD_1 et AD_2, chaque domaine contient 180 utilisateurs et 200 ressources avec différentes politiques de sécurité. Nous avons simulé 6000 différentes requêtes d'autorisation pour chaque cas. La comparaison des performances des modèles de contrôle d'accès reste une tâche difficile car il existe plusieurs métriques de comparaison, mais en ce qui concerne le parallélisme, nous avons vu dans la section 5.2.3 que le paramètre le plus important représente le temps d'exécution et c'est pourquoi nous l'avons utilisé pour la comparaison des deux cas. Les résultats de la simulation des deux cas : cas 1 et cas 2, ont été représentés dans le tableau 5.4 et la figure 5.7 (L'axe X représente le nombre des requêtes et l'axe Y représente le temps d'exécution d'un processus d'autorisation en milliseconde).

Cas/ nombre de requêtes	50	100	400	800	1600	3000	6000
Un seul PDP	17	168	548	794	849	629	972
Deux PDPs	35	90	281	447	341	290	386

Tableau 5. 4 Les résultats de la simulation

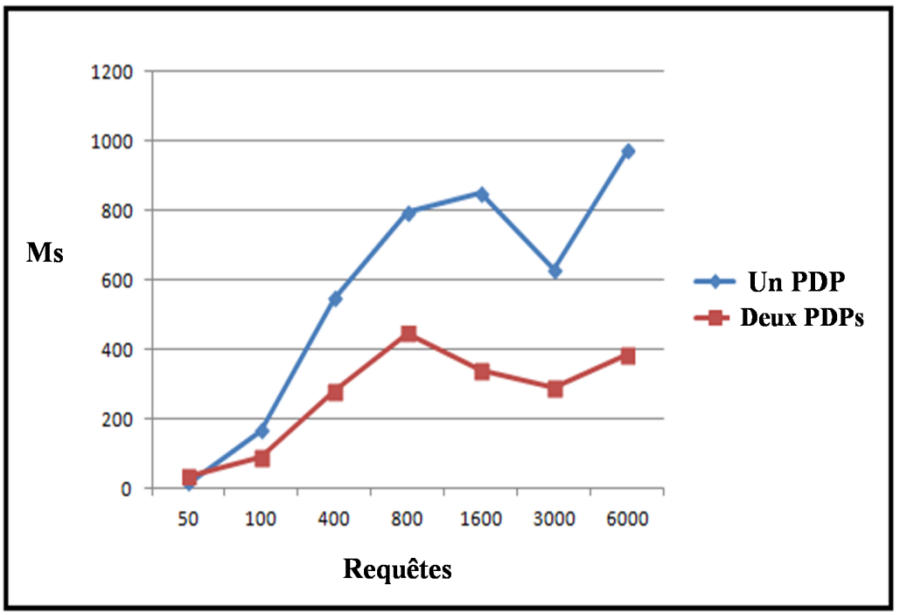

Figure 5. 7 Résultats de la simulation (Un seul PDP vs deux PDPs)

Les résultats (figure 5.7) montrent que l'exécution de 50 requêtes d'autorisation dans le premier cas (un seul PDP utilisant la programmation séquentielle) a pris moins de temps que leur exécution dans le second cas (deux PDPs en parallèle), ceci est dû au traitement parallèle qui prend un temps significatif lorsque la quantité de données n'est pas importante. Par contre, lorsque le nombre de requêtes d'autorisation augmente, le temps d'exécution utilisant un seul PDP devient plus important que le temps d'exécution utilisant deux PDPs en parallèle. Par ailleurs, les auteurs ont remarqué que les résultats obtenus ont un comportement non monotone, c'est-à-dire que le temps d'exécution augmente et diminue. L'augmentation du temps dans certains cas, est due au fait que lorsque le nombre de demandes d'autorisation augmente, le système aura besoin de plus de temps pour traiter ces demandes. Mais, traiter une requête implique plusieurs phases telles que: définir la nature de la requête (même domaine ou autre domaine), rechercher les rôles de l'utilisateur, convertir les rôles en cas de requêtes inter-domaines et vérifier les rôles des utilisateurs avec les politiques de sécurité des ressources. La diminution du temps d'exécution est due à la nature des requêtes générées, car les requêtes inter-domaines prennent plus de temps, y compris la phase de conversion de rôle.

En outre, la ressource demandée joue un rôle important dans ce comportement, car l'analyse de la table de sécurité pour trouver la ressource R1 prend moins de temps que l'analyse de la table de sécurité pour trouver la ressource R180. Ainsi, le comportement non monotone est dû à la nature aléatoire des demandes de contrôle d'accès et celle du choix de la ressource. L'utilisation du temps d'exécution comme paramètre de comparaison, a montré que le modèle proposé est sensible au nombre de demandes d'autorisation, c'est-à-dire lorsque le nombre de requêtes d'autorisation est important; le modèle proposé a donné de bons résultats. D'autre part, l'évaluation de la qualité d'un modèle de contrôle d'accès est généralement estimée par le nombre de faux positifs (situations où l'accès a été accordé, alors qu'il devra être refusé et vice versa). Les résultats illustrés dans le tableau 5.5 donnent le nombre d'accès acceptés et refusés dans le modèle proposé (deux PDP) et dans le modèle standard (un PDP). Pour chaque cas (deux PDP / un PDP), le nombre d'accès acceptés et refusés est le même et cela de 50 à 6000 requêtes. En conséquence, le modèle de contrôle d'accès inter-domaines et parallèle proposé est bénéfique pour l'architecture d'une grille parce que l'utilisation du parallélisme a permis aux auteurs de gagner du temps sans avoir une influence négative sur la qualité de la décision de contrôle d'accès. Le résultat obtenu pour une demande de contrôle d'accès est le même dans le cas du modèle proposé ou standard, mais dans le modèle proposé, il prend un temps de réponse plus court.

A : accepté R : refusé

Requêtes	50		100		400		800		1600		3000		6000	
Résultat de l'accès	A	R	A	R	A	R	A	R	A	R	A	R	A	R
Un PDP	21	29	43	57	184	216	364	436	744	856	1409	1591	2820	3180
Deux PDPs	21	29	43	57	184	216	364	436	744	856	1409	1591	2820	3180

Tableau 5. 5 Evaluation des accès acceptés et refusés

5.2.7. Evaluation du modèle proposé

Dans le modèle de contrôle d'accès proposé, notre attention était focalisée sur les axes de recherches les plus importants tels que : le contrôle d'accès, la collaboration entre les domaines et la performance. L'originalité de ce travail réside dans le fait qu'il prend en charge tous ces aspects par rapport aux recherches précédentes (Tableau 5.1) et cela comme suit :

5.2.7.1. Le contrôle d'accès

Le contrôle d'accès est nécessaire dans chaque système pour empêcher les attaques malveillantes, cette architecture fournit un contrôle d'accès qui combine les modèles ABAC [Yuan, E. and Tong, J., 2005] et RBAC [Barkley J. et Al., 1999], ABAC a été utilisé car il a le caractère d'être plus flexible et dynamiquement adapté au processus d'autorisation dans des environnements distribués tels que les grilles. Par contre, le modèle RBAC a été utilisé pour l'importance qu'il donne au rôle de l'utilisateur au lieu de son identité. D'autre part, le modèle RBAC a une fonction de correspondance de rôle solide qui associe un rôle local à un rôle global et cela entre différents domaines. La combinaison des deux modèles RBAC et ABAC donne le modèle nommé ARBAC qui garantit les avantages de chaque modèle.

5.2.7.2. Une architecture multi-domaines

Dans un environnement multi-domaines, les ressources sont distribuées dans différents endroits, et le contrôle d'accès inter-domaines devient un axe de recherche hautement ciblé par les chercheurs. Dans l'architecture proposée, nous avons pris en considération l'aspect inter-domaines car le modèle proposé fonctionne soit dans le cas où la ressource et l'utilisateur sont dans le même domaine ou dans deux domaines différents. Un cadre XACML étendu est utilisé pour séparer le processus d'autorisation entre la demande d'autorisation inter-domaines et la demande d'autorisation locale. D'autre part, une fonction de conversion de rôles est utilisée pour permettre à différents utilisateurs de différents domaines de partager leurs ressources et de collaborer ensemble pour résoudre différents problèmes et ce, quel que soit le domaine auquel ils appartiennent. En outre, la fonction de conversion de rôles utilisée est basée sur le modèle IRBAC 2000 [Al-Muhtadi J., 2000] et gère la collaboration sans violer les politiques de sécurité propre à chaque domaine.

5.2.7.3. La performance

A notre connaissance, prouver qu'une proposition est bonne réside dans l'évaluation de ses performances. Le modèle proposé a obtenu de bons résultats, ceci est dû à l'utilisation du parallélisme. Notre proposition réduit effectivement le temps de réponse à une requête de contrôle d'accès. En outre, l'utilisation du parallélisme n'a pas affecté les résultats de l'autorisation car dans le cas d'un seul PDP ou de deux PDPs, le modèle vérifie la même base des politiques et donne le même résultat éliminant les faux positifs.

Ce travail a été publié par le journal « International Journal of Embedded and Real-Time Communication Systems (IJERTCS) », Volume 9, Issue 1, Article 3. Notre modèle possède plusieurs avantages par rapport aux modèles existants (Tableau 5.6), parmi ces avantages, nous citons : Il est basé sur le modèle ARBAC ce qui lui permet de garantir les avantages du modèle RBAC ainsi que ceux du modèle ABAC en même temps. En outre, il permet la collaboration inter-domaines tout en respectant les politiques de sécurité internes de chaque domaine en utilisant une forte fonction de conversion de rôles, il réduit le temps de réponse à une requête de contrôle d'accès en éliminant les faux positifs. L'un des inconvénients de ce modèle, est qu'il s'est limité au deuxième processus d'autorisation sans se soucier du stockage et de la gestion des politiques de sécurité.

☒ Satisfait ● Non satisfait

Modèle	Approches	Validation	CID	RTRRA
ARBAC	• Parallélisme • Conversion de rôle • XACML • Respect des besoins de sécurité des domaines	☒	☒	☒

Tableau 5. 6 Les critères satisfaits par le modèle proposé

5.3. Le graphe pondéré d'autorisation dans les grilles (WGAG: Weighted Grid Authorization Graph)

Dans cette contribution, nous nous sommes intéressés au stockage et à la gestion des politiques de sécurité dans un environnement de grille. Plusieurs mécanismes ont été proposés pour gérer cette issue. Dans le chapitre 2, nous avons expliqué en détail ces mécanismes. Nous rappelons qu'après l'étude détaillée de ces derniers, nous avons trouvé que dans l'approche par force brute (BFA) [Kaiiali M. et Al., 2010], toutes les politiques de sécurité doivent être vérifiées afin de trouver le groupe de ressources auxquelles l'utilisateur peut accéder. C'est ce qui donne des répétitions énormes. Pour faire face à cette problématique, les auteurs ont proposé le mécanisme de groupement primitif (PCM) [Kaiiali M. et Al., 2010] qui permet de regrouper les ressources qui ont les mêmes politiques de sécurité. Ce mécanisme élimine la répétition lors de la vérification des mêmes politiques de sécurité mais il ne peut pas éliminer la répétition lors de la vérification des règles de sécurité identiques. Pout éliminer cette redondance, le mécanisme de groupement hiérarchique (HCM) [Kaiiali M. et Al., 2010]

a été proposé. Ce mécanisme considère les informations des nœuds parents de l'arbre PCM comme des données pour générer un regroupement hiérarchique de ces nœuds selon leurs règles de sécurité partagées. HCM a pu réduire la répétition lors de la vérification des règles de sécurité mais cette redondance existe toujours. En outre, ce mécanisme n'a pas pu représenter les politiques de sécurité basées sur le 'OU'. Pour contourner les inconvénients du mécanisme HCM, les auteurs ont proposé un graphe pour l'autorisation dans un environnement de grille (GAG) [Kaiiali M. et Al., 2013]. Ce dernier est un arbre de décision dérivé d'HCM en introduisant un nouveau type d'arc. GAG a totalement éliminé la redondance. Après une analyse approfondie du dernier mécanisme proposé, nous avons pu constater que : bien que GAG élimine totalement la répétition lors de la vérification des règles de sécurité, il y'a certaines règles de sécurité qui n'ont pas besoin d'être vérifiées dès le début mais GAG les vérifie quand même. C'est ce qui nous a poussés à proposer un nouveau mécanisme nommé 'un graphe pondéré pour l'autorisation dans un environnement de grille (WGAG)'. Lorsque nous proposons un nouveau mécanisme de stockage de règles de sécurité, nous sommes obligés de présenter le modèle de contrôle d'accès (autorisation) qui le gère.

5.3.1. Description du modèle de contrôle d'accès proposé

Un environnement de grille se compose généralement d'un ensemble de domaines administratifs organisés d'une manière hiérarchique. Chaque domaine a un administrateur qui gère l'ensemble des scénarios de grille pour ce domaine, en ajoutant les politiques de contrôle d'accès des ressources appartenant à ce dernier. Ces politiques de sécurité sont généralement écrites dans des fichiers XML, nous avons vu précédemment que le meilleur mécanisme proposé pour le stockage et la gestion des politiques de sécurité était le graphe d'autorisation (GAG). Comme chaque arc dans un graphe peut avoir un poids, nous avons attribué un poids (un entier non négatif) à chaque arc du GAG, ce poids représente le degré d'importance de la règle de sécurité à partir de laquelle l'arc émerge. Ensuite, un attribut nommé niveau de classification $RCL\ (ri)$ a été attribué à chaque ressource ri. Ce degré représente une valeur numérique qui est égale à la somme des poids des arcs du chemin le plus court de la racine à cette ressource (figure 5.8). En outre, chaque utilisateur Ui appartenant à la grille possède une clairance de sécurité $USC(Ui)$ qui est dérivée de l'ensemble de ses rôles. Avec cette proposition, nous pouvons éliminer le parcours du graphe pour les utilisateurs qui ont une clairance de sécurité inférieure au niveau de classification des ressources $(USC(Ui)) < (RCL(rj))$.

Le graphe pondéré d'autorisation dans les grilles (WGAG) a été développé avec les détails suivants :

- Soit $SR= \{sr_j \mid j=1 \ldots l\}$ l'ensemble des règles de sécurité.
- Soit $IDSR: SR \rightarrow N: IDSR(sr_j)$ = le degré d'importance de la règle de sécurité sr_j.
- Soit $G(V; E)$ le graphe d'autorisation dans les grilles (GAG), où V est l'ensemble des sommets (règles de sécurité) et E est un ensemble d'arcs.

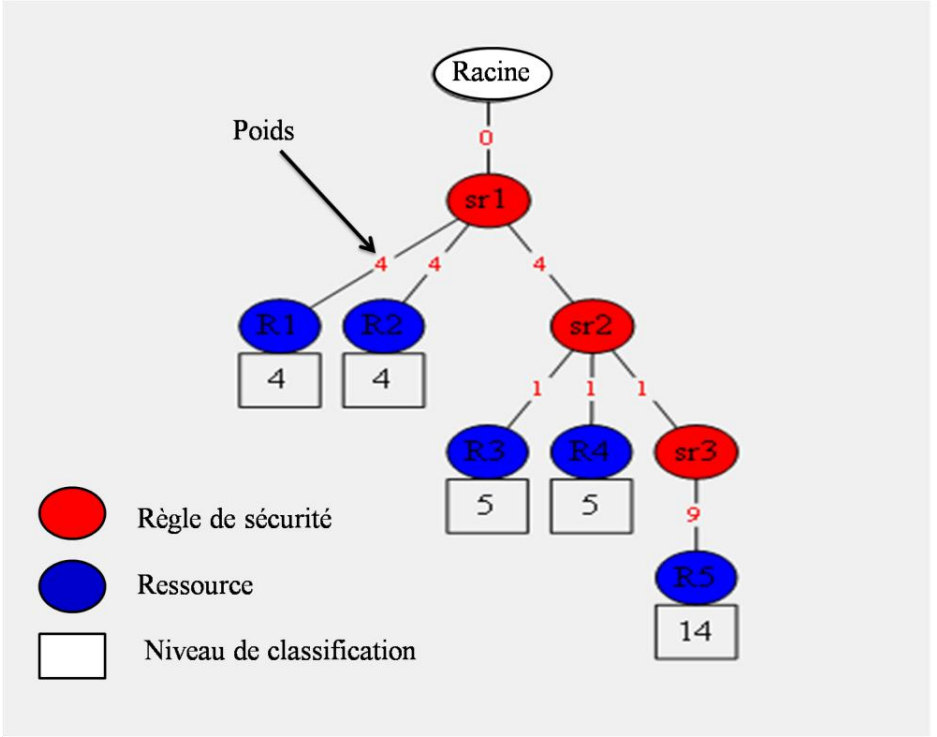

Figure 5. 8 Graphe généré par WGAG

- Soit $W: E \rightarrow N: W(e_{ij})$ = IDSR (sr_i) une fonction qui définit le poids de l'arc dans GAG. où sr_i représente la règle de sécurité à partir de laquelle l'arc e_{ij} démarre.
- Soit $R = \{r_j \mid j=1 \ldots n\}$ un ensemble de ressources de la grille.
- Soit $SP: R \rightarrow N: SP(r_j)$ = Le poids du plus court chemin de la racine à la ressource r_j.
- Soit $RCL: R \rightarrow N: RCL(r_j) = SP(r_j)$ une fonction qui calcule pour chaque ressource son niveau de classification
- Soit $Role = \{role_i \mid i=1 \ldots s\}$ un ensemble des rôles des utilisateurs
- Soit $IDR: Roles \rightarrow N: IDR(role_j)$ = le degré d'importance du rôle $role_j$.

- Soit $U = \{Ui \mid i = 1 \ldots m\}$ l'ensemble des utilisateurs.
- Soit $UR \subseteq U \times Roles$: l'ensemble des relations d'attribution des rôles aux utilisateurs.
- Soit $USR: U \rightarrow Roles : USR(u_i) = \{role_j \mid (u_i, role_j) \in UR\}$ une fonction derivée de UR qui attribut à chaque utilisateur un ensemble de rôles.
- Soit $USC: U \rightarrow N: USC(U_i) = \sum_{roles \in USR(Ui)} IDR(role_j)$ une fonction qui attribut à chaque utilisateur sa clairance de sécurité.

Pour bien comprendre notre contribution, nous proposons l'exemple suivant :

Soit un environnement de grille avec 5 ressources R= $\{r_1, r_2, r_3, r_4, r_5\}$ et 3 règles de sécurité SR= $\{sr_1, sr_2, sr_3\}$ avec leurs degrés d'importance (illustré dans le tableau 5.7).

- Sr_1 demande que l'utilisateur fasse partie de l'université XYZ
- Sr_2 demande que l'utilisateur ait le rôle «Etudiant »
- Sr_3 demande que l'utilisateur ait le rôle «Programmeur »

Règles de sécurité	Degrés d'importance
Sr_1	4
Sr_2	1
Sr_3	9

Tableau 5. 7 Les degrés d'importance des règles de sécurité

Les cinq ressources ont les politiques de sécurité suivantes :

- r_1, r_2 nécessitent sr_1.
- r_3 nécessite sr_1 et sr_2.
- r_4 nécessite sr_1, sr_2 et sr_3.
- r_5 nécessite sr_1, sr_2 et sr_3.

L'architecture de contrôle d'accès proposée est illustrée dans la figure 5.9, le serveur d'autorisation est basé sur XACML et contient les éléments suivants : le point d'information de la politique (PIP), le point d'application de la politique (PEP), le point de décision de la politique (PDP) et d'autres modules supplémentaires pour rendre le cadre XACML compatible avec WGAG tels que : analyseur et exécuteur de requête (RAP : Request Analyser and Processor), Engin de recherche WGAG, Base de données WGAG, parseur XACML.

Comme le montre la figure 5.9, les politiques de sécurité des ressources sont soumises par l'administrateur en utilisant le langage de rédaction de politique SAML [OASIS, 2016][b] ou XACML [Ferraiolo D. et Al., 2016]. Ensuite, le parseur XML parcourt tout le fichier XML et donne comme résultat une table de sécurité contenant toutes les ressources avec leurs politiques de sécurité (Tableau 5.8). L'engin générateur WGAG crée le graphe pondéré d'autorisation dans les grilles (WGAG) en utilisant la table de sécurité générée par le parseur XML, pratiquement c'est une implémentation directe de l'algorithme WGAG qui sera présenté dans les sections qui suivent. Dans ce graphe, les sommets représentent les règles de sécurité et les poids prendront pour valeur le degré d'importance des règles de sécurité, par contre, les feuilles du graphe représentent les ressources. Pour chaque ressource r_i, le graphe sera parcouru pour calculer le chemin le plus court de la racine au nœud r_i, la valeur obtenue sera prise comme niveau de classification de la ressource r_i (Figure 5.8). Le graphe WGAG résultant sera stocké dans la base de données WGAG pour qu'il soit parcouru à chaque requête de demande d'accès par l'engin de recherche WGAG.

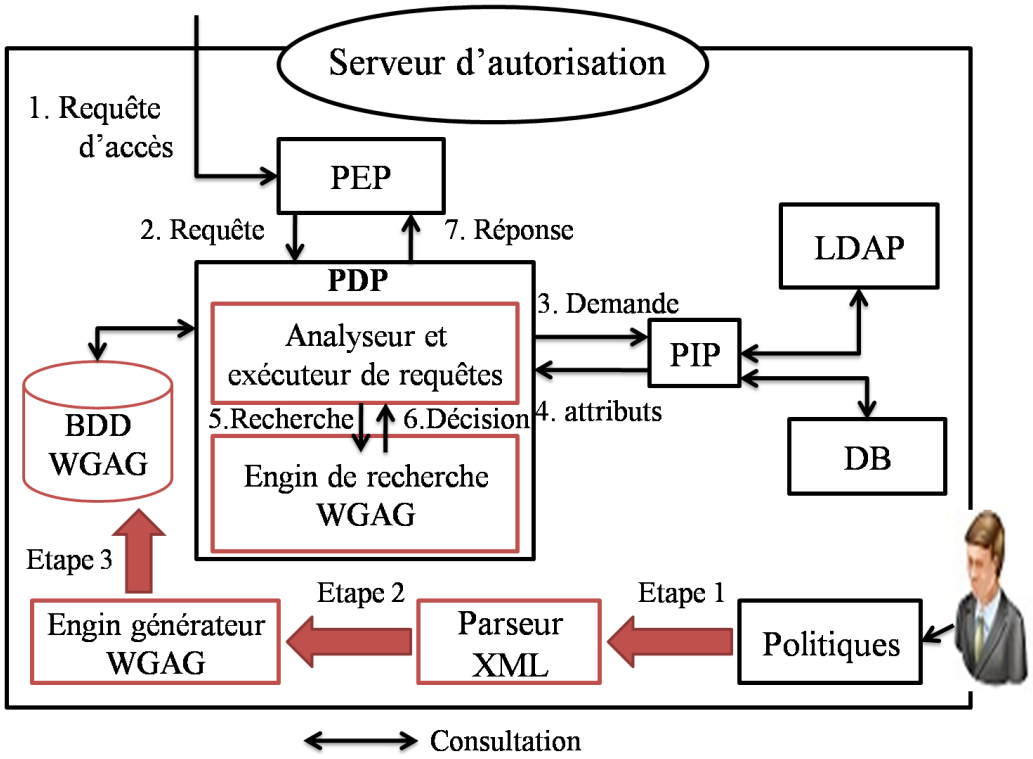

Figure 5. 9 Extension de l'architecture XACML avec le mécanisme WGAG (les éléments dessinés en rouge représentent notre contribution sauf l'analyseur et l'exécuteur de requêtes ainsi que le parseur de politiques qui existaient déjà dans le GAG)

sr_i / r_i	Sr_1	Sr_2	Sr_3
r_1	1	0	0
r_2	1	0	0
r_3	1	1	0
r_4	1	1	1
r_5	1	1	1

Tableau 5. 8 Résultat du parseur XML

5.3.1.1. Mode d'autorisation « groupe de ressources »

Lorsqu'un utilisateur génère une requête d'accès, le PEP l'intercepte et la diffuse au PDP. La requête est mise dans une queue dans le PDP. Puis le RAP qui est un simple écouteur d'action, prend la requête, demande au PIP les attributs manquants concernant l'utilisateur tels que : la clairance de sécurité USC(U_i). Ensuite, le RAP envoie la requête à l'engin de recherche WGAG. Ce dernier parcourt le graphe d'autorisation stocké dans la base de données WGAG, et donne le groupe de ressources qui satisferont : $RCL(ri) \leq USC(Ui)$. Enfin, le groupe de ressources est envoyé au RAP, puis au PEP (Figure 5.10).

5.3.1.2. Mode d'autorisation « Une seule ressource »

Lorsqu'un utilisateur génère une requête d'accès, le PEP l'intercepte et la diffuse au PDP. La requête est mise dans une queue dans le PDP. Puis le RAP prend la requête, demande au PIP les attributs manquants concernant l'utilisateur tels que : la clairance de sécurité USC(U_i). Ensuite, le RAP envoie la requête à l'engin de recherche WGAG. Ce dernier parcourt le graphe d'autorisation stocké dans la base de données WGAG pour la ressource spéciale r_i, puis il donne comme résultat, un accès accepté si : $RCL(ri) \leq USC(Ui)$. Sinon un accès refusé sera envoyé au RAP, puis au PEP.

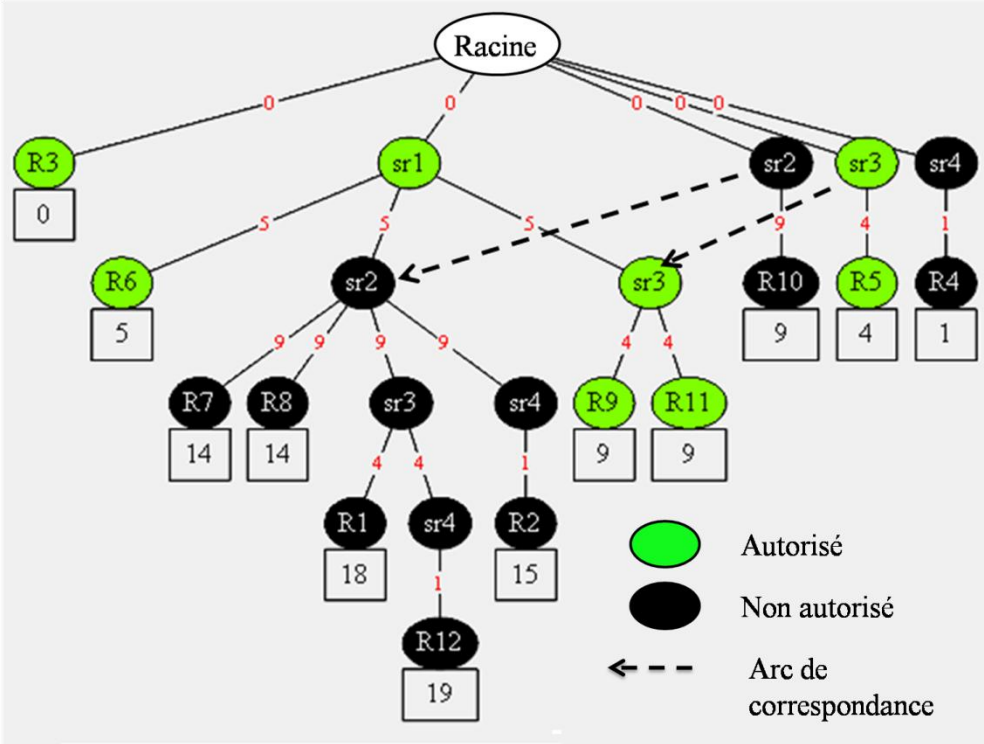

Figure 5. 10 Groupe des ressources autorisées pour un utilisateur avec Sr1 (degré d'importance= 5) et sr3 (degré d'importance = 4)

5.3.1.3. Algorithme WGAG

Entrées : Table de sécurité (**ST**), table des degrés d'importance des règles de sécurité (**IDT**).

Sorties : Le graphe pondéré d'autorisation dans les grilles (WGAG).

Les variables :

1. **SRV** : un vecteur de toutes les règles de sécurité
2. Chaque nœud (**N**) dans le graphe est une structure de trois champs :
   - La règle de sécurité (**sr**).
   - La table de sécurité provisoire (**ST**).
   - Un arc correspondant allant du nœud à la cellule représentative **sr** dans **SRV.**
3. Chaque arc (**e**) dans le graphe a un poids **w** qui est un entier représentant le degré d'importance de la règle de sécurité **sr** dans la table des degrés d'importance des règles de sécurité (**IDT**)**.**
4. Chaque ressource r_i dans le graphe est une structure de deux champs :
   - Un libellé
   - **CL** : Un entier qui représente le niveau de classification de la ressource.

Début

Etape 1 :

- Initialiser l'arbre de décision par un nœud racine qui a une valeur « **racine** » comme libellé de la règle de sécurité.
- Construire la table de sécurité (**ST)** qui représente la politique de sécurité entière du système. L'attribuer comme propriété table de sécurité (**ST)** du nœud racine.
- Construire la table des degrés d'importance des règles de sécurité (**IDT**) qui représente le degré d'importance de chaque règle de sécurité.
- Exécuter l'étape 2 pour le nœud racine.

Etape 2 :

- Ajouter la ressource $\mathbf{r_i}$ qui a une ligne égale à zéro dans la table de sécurité (**ST)** du nœud (**N)** comme une ressource enfant du nœud **(N).**
- Pour chaque colonne de la table de sécurité (**ST)** du nœud **(N),** calculer la somme et la référer comme « **Compte** ».
- Choisir la règle de sécurité $\mathbf{sr_j}$ qui a le **compte** le plus élevé.
- Diviser la table de sécurité (**ST)** en deux tables, excluant la j^{ieme} colonne comme suit :
  - La première table $\mathbf{T_1}$ contient les lignes des ressources qui demandent $\mathbf{sr_j}$ (chaque ligne qui a la $\mathbf{j^{ieme}}$ cellule > **0**)**.**
  - La seconde table $\mathbf{T_2}$ contient les lignes des ressources qui ne demandent pas $\mathbf{sr_j}$ (chaque ligne qui a la $\mathbf{j^{ieme}}$ cellule = **0**)**.**
- Ajouter un nœud enfant gauche (**LCN)** au nœud (**N)** avec $\mathbf{sr_j}$ comme règle de sécurité **sr** et $\mathbf{T_1}$ comme table de sécurité (**ST)**, le degré de $\mathbf{sr_j}$ dans (**IDT)** comme le poids de (**LCN)**. L'arc correspondant de (**LCN)** réfère à la cellule $\mathbf{sr_j}$ dans **SRV**.
- Ajouter un nœud enfant droit (**RCN)** au nœud (**N)** avec **NULL** comme règle de sécurité **sr** et $\mathbf{T_2}$ comme table de sécurité (**ST)**, 0 comme le poids de (**RCN)**. L'arc correspondant de (**RCN)** réfère à **NULL**.

Etape 3 : Répéter l'étape 2 pour chaque nœud enfant jusqu'à ce qu'on trouve un nœud avec une table de sécurité vide.

Etape 4 : Pour chaque ressource $\mathbf{r_i}$ dans le graphe, calculer la somme des poids du chemin le plus court de la racine à cette ressource, la valeur trouvée est considérée comme **CL** de la ressource $\mathbf{r_i}$.

Etape 5 : Effacer toutes les tables de sécurité provisoires pour libérer l'espace.

Fin

5.3.2. Simulation et résultats

Pour un environnement de grille de 100 ressources et 8 règles de sécurité, 100 différentes requêtes d'autorisation ont été initiées. Pour chaque processus d'autorisation, l'analyse du GAG et du WGAG a été faite. Les résultats de la simulation sont représentés dans la figure 5.11 (l'axe X représente le numéro du processus d'autorisation (Numéro de l'expérience) et l'axe Y représente la complexité de l'autorisation (le nombre des règles de sécurité vérifiées)).

Comme le montre la figure 5.11, nous pouvons dire que la complexité de WGAG (ligne rouge) est toujours inférieure à celle du GAG (ligne bleue), ceci est dû à l'utilisation des attributs : niveau de classification et clairance de sécurité dans le mécanisme WGAG. Dans ce dernier cas, lorsqu'un utilisateur a une clairance de sécurité inférieure au niveau de classification de la ressource, le graphe résultant WGAG ne sera pas du tout parcouru tandis que GAG parcourt tout le graphe pour vérifier si les rôles des utilisateurs satisferont la politique de sécurité de la ressource. L'un des points les plus remarquables du résultat, est que la complexité de WGAG atteint le zéro parfois, par contre celle du GAG est toujours supérieure à zéro. Ceci est dû à l'utilisation du mode « une seule ressource » car lorsque la clairance de sécurité d'un utilisateur est inférieure au niveau de classification de la ressource, il y'n'aura pas de règles de sécurité à vérifier dans le cas de WGAG, par contre dans le cas du GAG, il y'aura la vérification au moins d'une règle de sécurité. Donc, l'analyse des résultats de la simulation indique que WGAG réduit effectivement la complexité de vérification des règles de sécurité. Enfin, nous pouvons dire que WGAG améliore le contrôle d'accès et donne de meilleurs résultats que GAG.

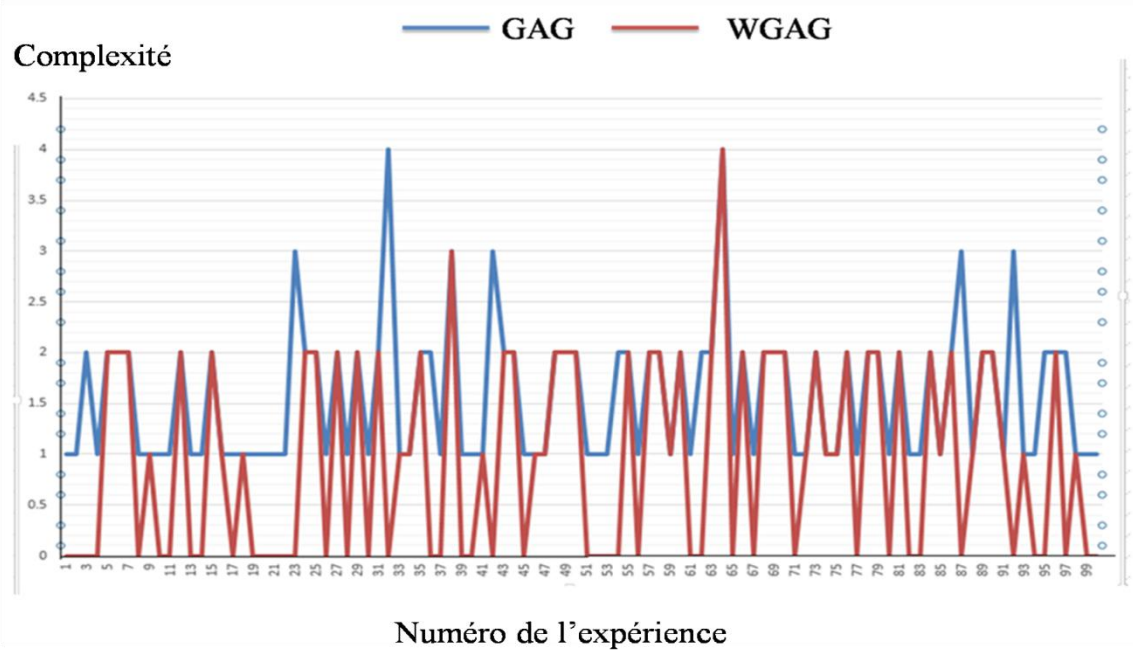

Figure 5. 11 Résultats de la simulation (Complexité GAG vs complexité WGAG)

Ce travail a été accepté et présenté à la conférence « The International Conference on Advanced Communication Technologies and Networking (**ComNet 2017**) »

Dans ce travail, nous avons étudié les points faibles des mécanismes de stockage et de gestion des politiques de sécurité. Puis, nous avons remarqué que le dernier et meilleur mécanisme proposé était le graphe d'autorisation dans les grilles (GAG), après une analyse approfondie de ce mécanisme, nous avons trouvé qu'on pourra l'améliorer pour éviter la vérification de certaines règles de sécurité dès le début. Ce point a été réalisé en proposant le graphe pondéré d'autorisation dans les grilles (WGAG), ce dernier a réellement diminué le nombre de règles de sécurité à vérifier ce qui permet logiquement de diminuer le temps de réponse à une requête de contrôle d'accès. Par contre, parmi les points négatifs de cette proposition et celle qui la précède, nous pouvons citer les deux suivants :

- Dans le cas du GAG comme du WGAG, nous avons supposé qu'il existe un parseur XML qui parcourt des fichiers XML de politiques de sécurité et donne comme résultat une table de sécurité. Cette dernière sera utilisée comme entrée dans le cas du GAG et du WGAG, mais pour l'instant il n'y a pas un algorithme ou une application qui existe réellement et nous permet de le faire.

- Une requête de contrôle d'accès réelle contient généralement un utilisateur, une ressource et une action à réaliser, les deux mécanismes n'ont pas tenu compte de cette dernière, ceci rend l'utilisation du GAG et du WGAG dans des environnements de grille réelle un peu difficile.

5.4. Le graphe pondéré basé sur l'action pour l'autorisation dans les grilles (Action-WGAG : Action-Weighted Grid Authorization Graph)

Dans cette contribution, nous avons essayé de résoudre les points négatifs de notre troisième contribution. Notre point de départ, était de rapprocher au maximum le mécanisme WGAG de la réalité pour pouvoir l'utiliser dans des serveurs d'autorisation réels utilisant des politiques de sécurité spécifiées en langage XACML. Cette contribution est une amélioration du WGAG, nous proposons un graphe pondéré d'autorisation dans les grilles qui est basé sur l'action (Action-WGAG). En outre, nous allons présenter l'algorithme du parseur XML que nous avons implémenté, nous avons appelé ce parseur, le parseur des politiques de sécurité (SP-Parser : Security Policy Parser).

5.4.1. Comparaison entre les différents travaux existants

Dans cette partie, nous allons faire une comparaison entre les différents travaux existants (présentés en détails dans le chapitre 2, la section 3.3.1.2 du chapitre 3 et la section 5.3 du chapitre 5). Ces travaux ont été divisés en deux groupes : le premier contient ceux qui ont pris en considération les mécanismes de stockage et de gestion des politiques de sécurité. Le second groupe comporte les travaux qui ont proposé des contributions concernant le processus d'autorisation lui-même. Pour chaque groupe, nous avons défini un ensemble de critères de comparaison que nous jugeons importants. Les critères de comparaison utilisés pour le premier groupe sont : l'élimination totale de la répétition lors de la vérification, l'élimination de certaines vérifications dès le début, la réduction de la complexité (nombre de règles de sécurité vérifiées), la réduction du temps de réponse à une requête d'accès et l'expressivité (représenter n'importe quelle politique de sécurité réelle).

Plusieurs travaux sur la représentation des politiques de sécurité ont été présentés dans le tableau 5.9. D'après les critères de comparaison utilisés, nous pouvons noter les remarques suivantes :

- **L'élimination totale de la redondance** : le mécanisme BFA nécessite la vérification totale des règles de sécurité ce qui mène vers des répétitions énormes. Par contre, PCM et HCM ont réduit la redondance dans la vérification des règles mais elle n'a pas été totalement éliminée. Ce critère a été satisfait uniquement par GAG et WGAG.

☒ Satisfait ● Non satisfait

Articles	Mécanismes	Elimination totale de la répétition	Elimination de certaines vérifications dès le début	Réduction de la complexité	Réduction du temps de réponse	Expressivité
Kaiiali M. et Al., 2013	BFA	●	●	●	●	●
	PCM	●	●	☒	●	●
	HCM	●	●	☒	●	●
	GAG	☒	●	☒	●	●
Namane S. et Al., 2017	WGAG	☒	☒	☒	●	●

Tableau 5. 9 Comparaison entre les travaux existants sur la représentation des politiques de sécurité

- **Elimination de certaines vérifications dès le début** : ce critère a été satisfait uniquement par WGAG, aucun des autres travaux ne l'a pris en considération.
- **Réduction de la complexité** : tous les mécanismes à l'exception de BFA ont réduit la complexité (le nombre de règles de sécurité vérifiées). Le meilleur résultat a été obtenu par WGAG.
- **Réduction de temps** : tous les mécanismes n'ont pas pris en charge ce critère.
- **Expressivité** : tous les mécanismes proposés ont été pauvres du point de vue expressivité car ils ne permettent pas de représenter des politiques de sécurité des systèmes réels qui utilisent les principes suivants : utilisateur, ressource et action.

Une comparaison entre un autre groupe de modèles de contrôle d'accès a été réalisée dans le tableau 5.10. D'après les critères utilisés nous pouvons noter les remarques suivantes :

- **Le mode d'autorisation** : les accès peuvent être accordés en fonction des trois modes, à savoir : l'identité de l'utilisateur, le rôle attribué à l'utilisateur ou des attributs de l'utilisateur (sujet). Les politiques basées sur les identités des utilisateurs définissent directement les autorisations que disposent ces derniers et doivent être gérées pour chaque utilisateur [Sadegh D.N. and Rasool J., 2016]. Par conséquent, leur maintenance représente une lourde charge pour les administrateurs. Cependant, les politiques basées sur les rôles des utilisateurs sont plus générales et plus faciles à gérer. Elles permettent à l'utilisateur d'exécuter directement toutes les autorisations associées à ce rôle. L'approche la plus flexible et la plus générale est basée sur les attributs des utilisateurs. Dans cette approche, les rôles attribués, les appartenances à des groupes, les affiliations et d'autres attributs dynamiques ou statiques tels que la fiabilité peuvent être référencés afin de déterminer les permissions des utilisateurs à la demande [Sadegh D.N. and Rasool J., 2016]. Nous pouvons remarquer que tous les travaux cités ci-dessous n'ont pas utilisé le mode « identité », certains d'entre eux ont utilisé le mode « rôles » et à partir de 2016, la plupart des travaux ont utilisé le mode « attributs des utilisateurs ».
- **La validation** : la plupart des travaux basés sur le mode « attributs des utilisateurs » ont été validés, par contre ceux basés sur le rôle ne l'ont pas tous été. Ceci est dû à la flexibilité des modèles basés sur les attributs par rapport à ceux basés sur les rôles et cela dans un environnement dynamique et distribué tel que les grilles et le Cloud Computing.
- **La réduction de temps** : ce critère n'a pas été largement considéré dans les travaux récents, il a été pris en charge uniquement dans [Balusamy B. et Al., 2017]. C'est un critère important car il influence directement sur l'efficacité du modèle de contrôle d'accès.
- **Les faux positifs** : ce critère fait référence aux situations où l'accès est refusé alors qu'il devra être accepté et vice versa. Aucun des travaux cités auparavant n'a pris en considération ce critère.

☒ Satisfait ● Non satisfait

Articles	Approches	Mode d'autorisation	Validation	Réduction de temps	Faux positifs
Sun L. et Al., 2012	▪ Les ontologies ▪ RBAC	Rôles des utilisateurs	●	●	●
Sun L. et Al., 2013	▪ Chiffrement ▪ RBAC	Rôles des utilisateurs	●	●	●
Zhou L. et Al., 2013	▪ Chiffrement basé sur le rôle ▪ RBAC	Rôles des utilisateurs	☒	●	●
Li N. et Al., 2015	▪ Rôle de position ▪ Rôle d'application ▪ Mécanisme de conversion de rôle	Rôles des utilisateurs	●	●	●
Chen A. et Al., 2016	▪ Mesure de risque ▪ Seuil de risque dynamique	Attributs des utilisateurs	☒	●	●
Khan F. et Al., 2016	▪ Chiffrement des attributs ▪ Elimination des attributs répétitifs ▪ Plusieurs autorités d'attributs	Attributs des utilisateurs	☒	●	●
Rucha D. et Al., 2017	▪ Chiffrement des attributs ▪ Attributs spécifiques temps	Attributs des utilisateurs	●	●	●
Balusamy B. et Al., 2017	▪ Structure d'accès hiérarchique ▪ Un système de jeton de permission	Attributs des utilisateurs	☒	☒	●
Jianan H. et Al., 2017	▪ Combinaison des facteurs : temps et attributs ▪ Contrôle d'accès pour des données sensibles au temps	Attributs des utilisateurs	☒	●	●

Tableau 5. 10 Comparaison entre les travaux existants sur le contrôle d'accès

5.4.2. Le modèle proposé

5.4.2.1. Motivation

Le contrôle d'accès aux ressources de la grille englobe deux parties, la première concerne la façon de stocker les règles de sécurité et la seconde prend en considération la manière de vérifier efficacement ces règles. Les articles cités précédemment ont été divisés en deux ensembles en fonction de la façon de représenter les règles de sécurité ou d'assurer un modèle de contrôle d'accès efficace. En comparant le premier ensemble de travaux connexes, le meilleur résultat a été donné par le graphe pondéré d'autorisation dans les grilles (WGAG), il a éliminé la vérification de certaines règles de sécurité qui n'avaient pas besoin d'être vérifiées dès le début, c'est ce qui a permis de réduire la complexité par rapport au graphe d'autorisation dans les grilles (GAG). Est ce que cette réduction est optimale ou d'autres améliorations peuvent être apportées sur WGAG? D'un autre côté, une demande de contrôle d'accès utilise des éléments importants tels que: l'utilisateur, la ressource et l'action. Le WGAG et tous les mécanismes proposés auparavant avaient une faible expressivité, le WGAG peut-il être plus expressif pour représenter des politiques de sécurité réelles? Tous les mécanismes cités dans la section précédente, ont parlé du principe de la table de sécurité mais aucun de ces mécanismes n'a spécifié comment est-elle générée à partir d'un fichier de sécurité XACML. Les auteurs ont seulement indiqué que l'analyseur SAX [Pan Y. et Al., 2008] peut être utilisé pour cela; est-il possible d'implémenter un analyseur basé sur SAX qui lit les fichiers XACML et donne comme résultat une simple table de sécurité résumant l'ensemble des politiques de sécurité des ressources? Enfin, est ce que la complexité ou, en d'autres termes, le nombre de règles de sécurité qui doivent être vérifiées sont les seuls critères que nous pouvons utiliser pour prouver l'efficacité d'un mécanisme proposé?

5.4.2.2. Description du modèle proposé

Un environnement de grille se compose généralement de plusieurs domaines administratifs organisés d'une manière hiérarchique [Kaustav, R. and Avijit, B., 2012]. Chaque domaine contient un ensemble d'utilisateurs, un ensemble de ressources et un serveur d'autorisation qui comporte les politiques de sécurité du système rédigées et gérées par l'administrateur de ce domaine. Chaque ressource r_j a sa propre politique de sécurité SP_j. Cette politique se compose des éléments suivants : la ressource, l'action qui sera réalisée sur la ressource et une ou plusieurs règles de sécurité qui déterminent qui peut accéder à cette ressource avec une telle action. L'architecture que nous proposons représente une extension de l'architecture

XACML basée sur le modèle ABAC (une autorisation basée sur le mode « attributs des utilisateurs »). Cette proposition est une amélioration du mécanisme WGAG proposé dans la contribution précédente. On considère une règle de sécurité comme une paire {Claim, Valeur du claim} où le claim est considéré comme une information sur un sujet particulier. Il peut être toute caractéristique liée au sujet ou tout ce dont il peut posséder tel que : le nom, e-mail, rôle, téléphone, organisation, … [WSO2 Inc, 2015][a]. La valeur du claim (valeur de l'attribut) : représente la valeur qu'on peut donner à ce claim telle que : pour le claim nom, nous pouvons donner comme valeur « Bouteflika ». Dans ce qui suit, nous allons présenter les détails du modèle proposé :

- Soit EnvironmentGrille = {$Domaine_1$, $Domaine_2$, $Domaine_3$, …} l'ensemble des domaines de la grille.
- Soit Domaine = {Utilisateurs, Ressources, Politiques} l'ensemble des éléments du domaine.
- Soit Utilisateurs = {U_1, U_2, U_3, …} l'ensemble des utilisateurs de la grille.
- Soit Ressources = {r_1, r_2, r_3, …} l'ensemble des ressources de la grille.
- Soit Actions = {Lire, Ecrire, Exécuter, Modifier, …} l'ensemble des actions qui peuvent être réalisées sur les ressources de la grille.
- Soit Politiques = {$Politique_1$, $Politique_2$, $Politique_3$, …} l'ensemble des politiques de sécurité de la grille.
- Soit Politique = {Ressources, Actions, ReglesDeSecurité } l'ensemble des éléments de la politique.
- Soit ReglesDeSecurité = {sr_1, sr_2, …} l'ensemble des règles de sécurité.
- Soit sr = {Claims, ValeurClaim} l'ensemble des éléments d'une règle de sécurité.
- Soit Claims = {Nom, Organisation, Role, …} l'ensemble des claims.
- Soit claimValue une chaine de caractère représentant la valeur du claim.
- Soit DemandesAcces = {Req_1, Req_2, …} un ensemble des requêtes d'accès.
- Soit Req = {Utilisateurs, Ressources, Actions} un ensemble des éléments de la requête.

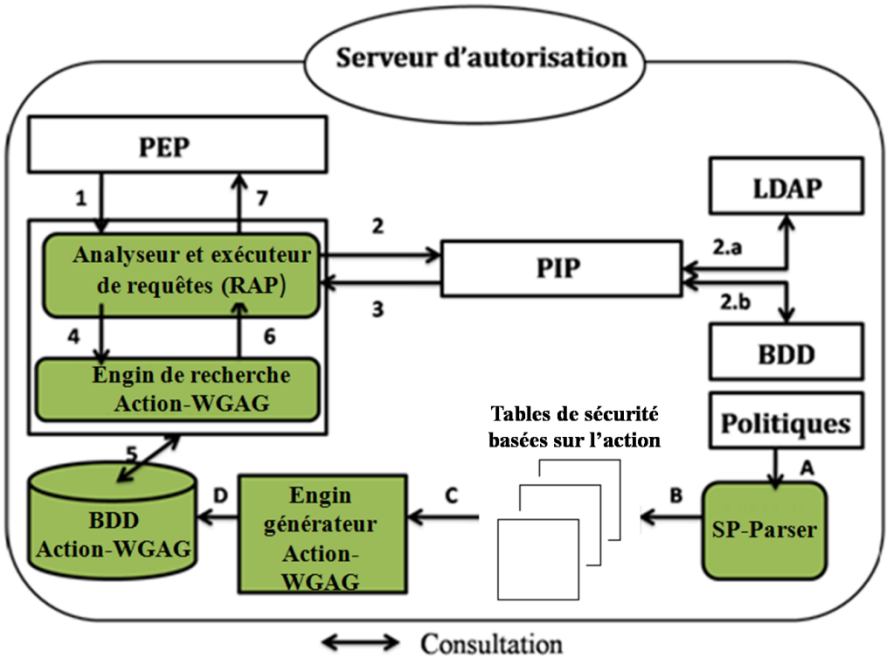

Figure 5. 12 Extension de l'architecture XACML pour compatibilité avec Action-WGAG (les éléments en vert représentent notre contribution sauf le RAP qui existait déjà dans le GAG)

Comme le montre la figure 5.12, dans un environnement de grille, l'administrateur soumet toutes les politiques de sécurité des ressources de la grille dans des fichiers XML rédigés en langage XACML V3. Ces fichiers vont être utilisés comme entrée au parseur de politique (A). Ce dernier va parcourir les fichiers XML et rendre comme résultat un ensemble de tables de sécurité qui regroupent les ressources selon l'action (B). L'engin générateur du Action-WGAG va utiliser ces tables de sécurité pour générer des graphes pondérés d'autorisation dans les grilles selon l'action mais en se basant sur l'algorithme WGAG (C). Les graphes générés seront stockés dans la base Action-WGAG pour être parcourus lors d'une réception d'une requête de contrôle d'accès (D). Lorsqu'un utilisateur génère une requête d'accès, le PEP l'intercepte et la diffuse au PDP (1). La requête est mise dans une queue dans le PDP. Puis le RAP prend la requête, demande au PIP les attributs manquants concernant l'utilisateur tels que : la clairance de sécurité USC(U_i) (2). Ensuite après la réception de ces attributs (3), le RAP envoie la requête à l'engin de recherche Action-WGAG (4). Ce dernier parcourt le graphe d'autorisation de l'action qu'il trouve dans la requête et donnera le groupe des ressources qui satisferont : $RCL(ri) \leq USC(Ui)$.

Pour bien comprendre notre contribution, nous proposons l'exemple suivant : Soit un environnement de grille avec 13 ressources R= {r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12}, r_{13}}, les

actions réalisées sur ces ressources sont Action = {Lire, Ecrire, Modifier, Exécuter} et les politiques de sécurité suivantes :

- La politique de sécurité de la ressource r_1 (Domaine 1)
  - Le rôle **Etudiant** à l'**Université Badji Mokhtar** comme **organisation** peut réaliser l'action **Lire**.
  - Le rôle **Programmeur** à l'**Université Badji Mokhtar** comme **organisation** peut réaliser l'action **Ecrire** et l'action **Exécuter**.
- La politique de sécurité de la ressource r_2 (Domaine 1)
  - Le rôle **Enseignant** à l'**Université Badji Mokhtar** comme **organisation** peut réaliser l'action **Ecrire.**
  - Le rôle **Enseignant** à **l'Université Hyderabad** comme **organisation** peut réaliser l'action **Modifier**.
- La politique de sécurité de la ressource r_3 (Domaine 1)
  - Le rôle **Enseignant** à **l'Université Hyderabad** comme **organisation** peut réaliser l'action **Ecrire**.
  - Le rôle **Enseignant** à l'**Université Badji Mokhtar** comme **organisation** peut réaliser l'action **Modifier**.
  - Le rôle **Etudiant** à **l'Université Hyderabad** comme **organisation** peut réaliser l'action **Exécuter**.
- La politique de sécurité de la ressource r_4 (Domaine 1)
  - Le rôle **Programmeur** à l'**Université Badji Mokhtar** comme **organisation** peut réaliser l'action **Ecrire** et l'action **Exécuter**.
- La politique de sécurité de la ressource r_5 (Domaine 1)
  - Le rôle **Etudiant** à **l'Université Hyderabad** comme **organisation** peut réaliser l'action **Lire**.
- La politique de sécurité de la ressource r_6 (Domaine 1)
  - Le rôle **Programmeur** à **l'Université Hyderabad** comme **organisation** peut réaliser l'action **Exécuter**.
  - Le rôle **Enseignant** à l'**Université Badji Mokhtar** comme **organisation** peut réaliser l'action **Ecrire**.
  - Le rôle **Enseignant** peut réaliser l'action **Exécuter**.
- La politique de sécurité de la ressource r_7 (Domaine 1)

- Le rôle **Etudiant** à l'**Université Badji Mokhtar** comme **organisation** peut réaliser l'action **Lire**.

- La politique de sécurité de la ressource r_8 (Domaine 1)
  - Le rôle **Enseignant** à **l'Université Hyderabad** comme **organisation** peut réaliser l'action **Ecrire**.
  - Le rôle **Programmeur** peut réaliser l'action **Exécuter.**
- La politique de sécurité de la ressource r_9 (Domaine 1)
  - Le rôle **Etudiant** à l'**Université Badji Mokhtar** comme **organisation** peut réaliser l'action **Lire**.
  - Le rôle **Etudiant** à **l'Université Hyderabad** comme **organisation** peut réaliser l'action **Exécuter**.
- La politique de sécurité de la ressource r_{10} (Domaine 1)
  - Les utilisateurs de **l'Université Hyderabad** comme **organisation** peuvent réaliser l'action **Ecrire**.
- La politique de sécurité de la ressource r_{11} (Domaine 1)
  - Le rôle **Programmeur** à **l'Université Hyderabad** comme **organisation** peut réaliser l'action **Modifier**.
  - Le rôle **Enseignant** à l'**Université Badji Mokhtar** comme **organisation** peut réaliser l'action **Ecrire**.
- La politique de sécurité de la ressource r_{12} (Domaine 1)
  - Le rôle **Enseignant** à l'**Université Badji Mokhtar** comme **organisation** peut réaliser l'action **Ecrire**.
  - Le rôle **Enseignant** à **l'Université Hyderabad** comme **organisation** peut réaliser l'action **Modifier**.
- La politique de sécurité de la ressource r_{13} (Domaine 1)
  - Le rôle **Programmeur** à l'**Université Badji Mokhtar** comme **organisation** peut réaliser l'action **Modifier.**
  - Le rôle **Enseignant** à **l'Université Hyderabad** comme **organisation** peut réaliser l'action **Ecrire**.

Pour rédiger ces politiques de sécurité en langage XACML, le serveur d'identité WSO_2 a été utilisé. Ce dernier gère les identités et les droits d'accès en garantissant la sécurité lors de la

connexion et en gérant des identités multiples entre plusieurs applications [WSO2 Inc, 2015][b]. Le serveur WSO_2 permet aux architectes d'entreprise et aux développeurs d'améliorer l'expérience des clients grâce à un environnement sécurisé avec une authentification unique (SSO) [WSO2 Inc, 2015][b] . En outre, WSO_2 fournit un éditeur de politique XACML pour la création de politiques XACML version 3.0. L'éditeur offre une manière simple de définir des règles et des politiques. Un exemple de l'élément <PolicySet > est illustré dans la (Figure 5.13). Chaque politique de sécurité des ressources est représentée par un élément <Policy> et aura un identifiant. L'élément <PolicySet> contient un ensemble d'éléments <Policy>, si une politique appartient à <PolicySet>, son identifiant sera cité dans la définition <PolicySet>.

Le parseur de politique de sécurité (SP-Parser) parcourt le document PolicySet, prend les identifiants des politiques. Puis pour chaque identifiant de politique trouvé, un parcours du fichier de cette dernière est fait. Le but principal de ce parcours est d'extraire la ressource, l'action et la règle de sécurité. De cette manière, le parcours de tous les fichiers des politiques de sécurité permet au parseur (SP-Parser) de créer les tables de sécurité en se basant sur les actions trouvées. Le problème à résoudre est : Comment extraire une règle de sécurité ? Pour résoudre ce problème, le serveur WSO_2 a utilisé la notion de claim. Ce qui nous a permis de diviser chaque règle de sécurité en deux parties : le claim et la valeur du claim. Dans notre exemple, nous avons utilisé deux types de claims : Claim= {Role, organisation}.

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-
algorithm:first-applicable" PolicySetId="Grid-Computing-Policy"  Version="1.0">
  <Description>This is Grid-Computing-Policy policy set</Description>
 <Target></Target>
  <PolicyIdReference>Resource-One-Read-Policy</PolicyIdReference>
  <PolicyIdReference>Resource-One-Run-Policy</PolicyIdReference>
  <PolicyIdReference>Resource-One-Write-Policy</PolicyIdReference>
  <PolicyIdReference>Resource-Two-Modify-Policy</PolicyIdReference>
  <PolicyIdReference>Resource-Two-Write-Policy</PolicyIdReference>
  <PolicyIdReference>Resource-Three-Modify-Policy</PolicyIdReference>
  <PolicyIdReference>Resource-Three-Run-Policy</PolicyIdReference>
  <PolicyIdReference>Resource-Three-Write-Policy</PolicyIdReference>
  <PolicyIdReference>Resource-Four-Run-Policy</PolicyIdReference>
  <PolicyIdReference>Resource-Four-Write-Policy</PolicyIdReference>
  <PolicyIdReference>Resource-Five-Read-Policy</PolicyIdReference>
  </PolicySet>
```

Figure 5. 13 Exemple de l'élément PolicySet

Le résultat du parseur (SP-Parser) sera : Une table de sécurité de l'action « Lire » (tableau 5.11), une table de sécurité de l'action « Ecrire » (Tableau 5.12), une table de sécurité de l'action « Modifier » (Tableau 5.13), une table de sécurité de l'action « Exécuter » (Tableau 5.14) et une table de conversion contenant la règle de sécurité et la paire {Claim, valeur du Claim} (Tableau 5.15).

Res \ Sr	Sr_1	Sr_2	Sr_5
r_1	1	1	0
r_5	1	0	1
r_7	1	1	0
r_9	1	1	0

Tableau 5. 11 Table de sécurité de l'action "Lire" réalisée par le SP-Parser

Res \ Sr	Sr_3	Sr_2	Sr_4	Sr_5
r_1	1	1	0	0
r_2	0	1	1	0
r_3	0	0	1	1
r_4	1	1	0	0
r_6	0	1	1	0
r_8	0	0	1	1
r_{10}	0	0	0	1
r_{11}	0	1	1	0
r_{12}	0	1	1	0
r_{13}	0	0	1	1

Tableau 5. 12 Table de sécurité de l'action "Ecrire" réalisée par le SP-Parser

Res \ Sr	Sr_4	Sr_5	Sr_2	Sr_3
r_2	1	1	0	0
r_3	1	0	1	0
r_{11}	0	1	0	1
r_{12}	1	1	0	0
r_{13}	0	0	1	1

Tableau 5. 13 Table de sécurité de l'action "Modifier" réalisée par le SP-Parser

Sr \ Res	Sr_3	Sr_2	Sr_1	Sr_4	Sr_5
r_1	1	1	0	0	0
r_3	0	1	1	0	0
r_4	1	1	0	0	0
r_6	0	0	0	1	0
r_6	1	0	0	0	1
r_8	1	0	0	0	0
r_9	0	0	1	0	1

Tableau 5. 14 Table de sécurité de l'action "Exécuter" réalisée par le SP-Parser

Règle de sécurité	Claim	Valeur du claim
Sr_1	Role	Etudiant
Sr_2	Organisation	Université Badji Mokhtar
Sr_3	Role	Programmeur
Sr_4	Role	Enseignant
Sr_5	Organisation	Hyderabad

Tableau 5. 15 Une table de conversion contenant la règle de sécurité et la paire {Claim, valeur du Claim}

5.4.2.3. L'implémentation du parseur de politiques de sécurité (SP-Parser)

- ***Détails d'implémentation du parseur***

Dans cette partie, nous allons expliquer les détails d'implémentation du parseur (SP-Parser) que nous proposons. Il faut noter que ce dernier prend en charge les politiques de sécurité rédigées en langage XACML. En outre, il est basé sur le parseur Sax [Pan Y. et Al., 2008] qui est implémenté en langage Java. Sax utilise l'interface *Content Handler*, cette interface fait appel aux méthodes suivantes [Oracle, 2015]:

- **void startDocument()** : Appelée au début d'un document.
- **void endDocument() :** Appelée à la fin d'un document.
- **void startElement**(String uri, String localName, String qName, Attributes atts) : Appelée au début d'un élément.
- **void endElement**(String uri, String localName,String qName): Appelée à la fin d'un élément.
- **void characters(char[] ch, int start, int length) :** Appelée lorsque des données « caractères » sont rencontrées.

Le parseur Sax a été amélioré pour créer le parseur de politiques de sécurité (SP-Parser) en ajoutant deux procédures (Méthodes) de parcours. La première est nommée « PolicySet Parsing », elle a comme entrée un fichier de politique xml contenant tous les identifiants des politiques de sécurité des ressources. Lorsqu'un identifiant d'une politique de ressource particulière est trouvé, la sous procédure « *SubPolicy Parsing* » est appelée avec le fichier Xml de la politique de sécurité de la ressource comme paramètre. Le rôle le plus important de la deuxième procédure de parcours est l'extraction du nom de la ressource, l'action et le nombre de règles de sécurité dans le fichier de politique. Pour chaque règle de sécurité, la procédure prend le claim et sa valeur. Tous ces résultats sont rendus à la première procédure de parcours où plusieurs tests seront faits. Le premier consiste à vérifier si la paire {Claim,

valeur du Claim} existe dans la table de conversion (Tableau 5. 15). Si la paire existe déjà, la procédure testera alors l'existence de la règle de sécurité dans la table de sécurité de l'action extraite précédemment. Si la règle de sécurité existe, la ressource sera ajoutée et la cellule (ressource, règle de sécurité) aura 1 comme valeur. Sinon, la règle de sécurité sera ajoutée et le même traitement sera fait avec la ressource. Nous voulons préciser que le parseur Sax est toujours existant dans le parseur SP-Parser, ce qui veut dire que toutes les méthodes et les variables de Sax sont également utilisées dans SP-Parser.

- ***L'algorithme du parseur de politique de sécurité (SP-Parser)***

Les entrées : un fichier XML de type PolicySet ; un ensemble de fichier XML de type Policy.

Les sorties : Table-de-sécurité -Lire-, Table-de-sécurité -Ecrire-, Table-de-sécurité -Modifier, Table-de-sécurité -Exécuter- , Table-conversion-Regle-claim.

Variables

node : une chaine de caractère définie dans le parseur Sax.

Str : une chaine de caractère définie dans le parseur Sax.

AttrValue : une chaine de caractère définie dans le parseur Sax.

namea : une chaine de caractère définie dans le parseur Sax.

PoliciesList : une liste qui va contenir tous les identifiants trouvés dans le fichier PolicySet.

ResultOne : un tableau de type chaine de caractère contenant quatre éléments : la ressource extraite, l'action, le claim et la valeur du claim de la première règle de sécurité dans le fichier de la politique.

ResultTwo : un tableau de type chaine de caractère contenant quatre éléments : la ressource extraite, l'action, le claim et la valeur du claim de la deuxième règle de sécurité dans le fichier de la politique.

NbrRule : un nombre entier qui représente le nombre de règles de sécurité dans le fichier de la politique.

Resource : une chaine de caractère représentant le nom de la ressource.

Action : une chaine de caractère représentant l'action.

Claim : une chaine de caractère qui représente le claim.

ClaimValue : une chaine de caractère qui représente la valeur du claim.

Sr : une chaine de caractère qui va contenir la règle de sécurité retournée.

k : un entier qui va contenir l'index de la règle de sécurité trouvé dans la table de sécurité.

R : un entier qui représente l'index de la ressource dans la table de sécurité.

Fonctions et procédures

Existclaim (claim: string, claimValue: string, Table-conversion-Regle-claim: array [x][3]) **:** une fonction qui cherche la paire {Claim, ClaimValue}dans la table de correspondance entre les règles de sécurité et la paire {Claim, claimValue}. Si la paire existe, la fonction retournera la règle de sécurité correspondante, sinon elle retournera une chaine vide.

AddResource (Resource: String, Table: array[x][y]): une procédure qui permet d'ajouter une ressource à la table de sécurité.

AddSecurityRule (sr: String, Table: array[x][y]): une procédure qui permet d'ajouter une règle de sécurité à la table de sécurité.

SetValueToOne (resource: integer, sr: integer): une procédure qui permet de mettre 1 comme valeur de la cellule (resource, sr).

FindSecurityRule(sr: string, table : array [x][y]): une fonction qui permet de retourner la position de la règle de sécurité sr si elle existe dans la table, sinon elle retournera -1.

Debut

nbrRule ← 0; k ← 0; R ← 0

Commencer le parcours du fichier PolicySet

Si node = "PolicyIdReference" **alors** ajouter str à la PolicyList

FinSi

Pour chaque élément dans PolicyList

appeler ParseSubPolicy(element, nbrRule, ResultOne, ResultTwo)

FinPour

Si nbrRule > = 1 **alors**

Resource ← ResultOne[0]

Action ← ResultOne[1]

Claim ← ResultOne[2]

ClaimValue ← ResultOne[3]

Sr ← Existclaim (Claim, claimValue, Table-conversion-Regle-claim)

Si Sr est vide **Alors**

Ajouter une nouvelle règle de sécurité Sr à la Table-conversion-Regle-claim

Ajouter le Claim correspondant à cette règle de sécurité

Ajouter ClaimValue corresponante à cette règle de sécurité

Sr ← la nouvelle règle de sécurité créée

FinSi

Si Action est égale à 'Read' **alors**

k ← FindSecurityRule (Sr, Table-de-sécurité -Lire-)

si k= -1 **alors**

AddSecurityRule (Sr, Table-de-sécurité -Lire-)

AddResource (Resource, Table-de-sécurité -Lire-)

SetValueToOne (R, k)

Sinon

AddResource (Resource, Table-de-sécurité -Lire-)

SetValueToOne (R, k)

FinSi

Sinon Si Action est égale à 'Write' **Alors**

k ← FindSecurityRule (Sr, Table-de-sécurité -Ecrire-)

Si k= -1 **Alors**

AddSecurityRule (Sr, Table-de-sécurité -Ecrire-)

AddResource (Resource, Table-de-sécurité -Ecrire-)

SetValueToOne (R, k)

Sinon

AddResource (Resource, Table-de-sécurité -Ecrire-)

SetValueToOne (R, k)

FinSi

Sinon Si Action est égale à 'Modify' **Alors**

k ← FindSecurityRule (Sr, Table-de-sécurité -Modifier-)

Si k= -1 **Alors**

AddSecurityRule (Sr, Table-de-sécurité -Modifier-)

AddResource (Resource, Table-de-sécurité -Modifier-)

SetValueToOne (R, k)

Sinon

AddResource (Resource, Table-de-sécurité -Modifier-)

SetValueToOne (R, k)

FinSi

Sinon Si Action est égale à 'Run' **Alors**

k ← FindSecurityRule (Sr, Table-de-sécurité -Executer-)

Si k= -1 **Alors**

AddSecurityRule (Sr, Table-de-sécurité -Executer-)

AddResource (Resource, Table-de-sécurité -Executer-)

SetValueToOne (R, k)

Sinon

AddResource (Resource, Table-de-sécurité -Executer-)

SetValueToOne (R, k)

FinSi

FinSi

Si nbrRule > 1 **Alors**

Resource ← ResultTwo[0]

Action ← ResultTwo[1]

Claim ← ResultTwo[2]

ClaimValue ← ResultTwo[3]

Sr ← Existclaim (Claim, claimValue, Security-rules-vs-claims)

Si Sr est vide **Alors**

Ajouter une nouvelle règle de sécurité Sr à la Table-conversion-Regle-claim

Ajouter le Claim correspondant à cette règle de sécurité

Ajouter ClaimValue corresponante à cette règle de sécurité

Sr ← la nouvelle règle de sécurité créée

FinSi

Si Action est égale à 'Read' **alors**

k ← FindSecurityRule (Sr, Table-de-sécurité -Lire-)

Si k= -1 **Alors**

AddSecurityRule (Sr, Table-de-sécurité -Lire-)

AddResource (Resource, Table-de-sécurité -Lire-)

SetValueToOne (R, k)

Sinon

AddResource (Resource, Table-de-sécurité -Lire-)

SetValueToOne (R, k)

FinSi

Sinon Si Action est égale à 'Write' **Alors**

k ← FindSecurityRule (Sr, Table-de-sécurité -Ecrire-)

Si k= -1 A**lors**

AddSecurityRule (Sr, Table-de-sécurité -Ecrire-)

AddResource (Resource, Table-de-sécurité -Ecrire-)

SetValueToOne (R, k)

Sinon

AddResource (Resource, Table-de-sécurité -Ecrire-)

SetValueToOne (R, k)

FinSi

Sinon Si Action est egale à 'Modify' **alors**

k ← FindSecurityRule (Sr, Table-de-sécurité -Modifier-)

Si k= -1 **Alors**

AddSecurityRule (Sr, Table-de-sécurité -Modifier-)

AddResource (Resource, Table-de-sécurité -Modifier-)

SetValueToOne (R, k)

Sinon

AddResource (Resource, Table-de-sécurité -Modifier-)

SetValueToOne (R, k)

FinSi

Sinon Si Action est égale à 'Run' **Alors**

k ← FindSecurityRule (Sr, Table-de-sécurité -Executer-)

Si k= -1 **Alors**

AddSecurityRule (Sr, Table-de-sécurité -Executer-)

AddResource (Resource, Table-de-sécurité -Executer-)

SetValueToOne (R, k)

Sinon

AddResource (Resource, Table-de-sécurité -Executer-)

SetValueToOne (R, k)

FinSi

FinSi

Fin

Procedure ParseSubPolicy(File: string , var nbrRule: integer, var ResultOne: array [4]of string, var ResultTwo: array [4] of string)

Debut

Si node= 'Rule' **Alors** nbrRule ← nbrRule+1

FinSi

Si node= 'AttributeValue' **Alors**

Si str commence par R suivi d'un nombre **Alors**

ResultOne[0] ← str

Sinon Si str= 'Read' or str= 'Write' or str= 'Modify' or str='Run' **Alors**

Si nbrRule >1 **Alors**

ResultTwo[0] ← str

Sinon ResultOne[0] ← str

FinSi

Sinon Si str n'est pas vide **Alors**

Si nbrRule >1 **Alors**

ResultTwo[1] ← str

Sinon ResultOne[1] ← str

FinSi FinSi FinSi FinSi

Sinon Si node='AttributeDesignator' and namea='AttributeId' and AttrValue= 'http://wso2.org/claims/organization' **Alors**

Si nbrRule >1 **Alors**

ResultTwo[2] ← 'organization'

Sinon ResultOne[2] ← 'organization'

FinSi

Sinon Si node='AttributeDesignator' and namea='AttributeId' and AttrValue= 'http://wso2.org/claims/role' **Alors**

Si nbrRule >1 **Alors** ResultTwo[3] ← 'role'

Sinon ResultOne[3] ← 'role'

FinSi FinSi

FinSi FinSi

Fin

Règles de sécurité	Degrés d'importance
Sr_1	3
Sr_2	2
Sr_3	1
Sr_4	4
Sr_5	6

Tableau 5. 16 Table des règles de sécurité et leur degré d'importance

Les tables de sécurité résultantes du parcours des fichiers de politique par le SP-Parser sont utilisées avec la table des degrés d'importance des règles de sécurité (tableau 5.16) comme entrée à l'engin générateur d'Action-WGAG pour générer les graphes pondérés d'autorisation dans les grilles (WGAG) basés sur l'action en utilisant l'algorithme WGAG [Namane S. et Al., 2017]. On obtiendra comme résultat un graphe pondéré pour chaque table de sécurité : un WGAG pour l'action lire (figure 5.14), un WGAG pour l'action écrire, un WGAG pour l'action modifier et un WGAG pour l'action exécuter. Lorsqu'un utilisateur génère une requête d'accès, cette dernière contient : la ressource et l'action. Selon l'action trouvée, l'Action-WGAG va parcourir uniquement le graphe de cette action, ce qui évitera le parcours d'un graphe global contenant les politiques de toutes les ressources.

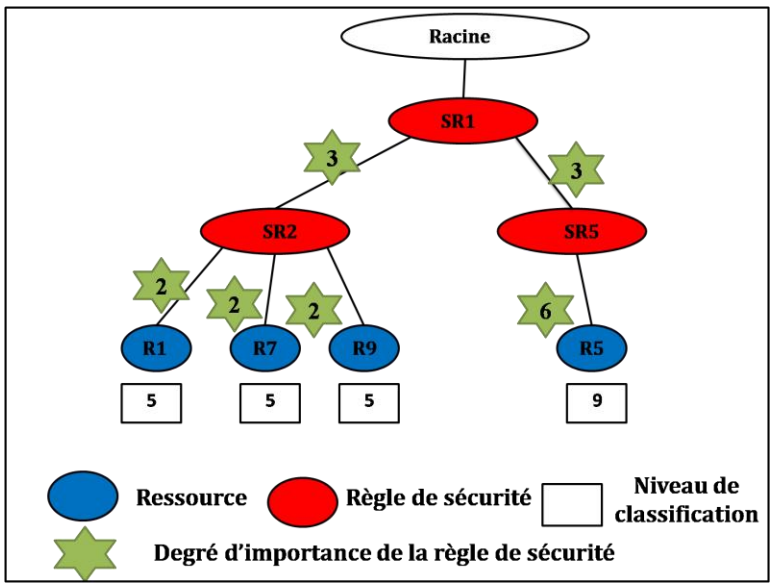

Figure 5. 14 Graphe WGAG pour l'action « lire »

5.4.3. Simulations et résultats

Le graphe pondéré d'autorisation dans les grilles (WGAG) a été amélioré dans cette contribution pour avoir le graphe pondéré basé sur l'action pour l'autorisation dans les grilles (Action-WGAG). Donc, le simulateur (Action-WGAG) est une amélioration du simulateur (WGAG), c'est une application programmée en C# également. Pour un environnement de grilles contenant 96 ressources et 7 règles de sécurité, les deux cas suivants ont été implémentés :

- **Premier cas** : quatre tables de sécurité ont été créées contenant chacune 24 ressources et quelques règles de sécurité appartenant à Sr = {sr_1, sr_2, sr_3, sr_4, sr_5, sr_6, sr_7}. Chaque table contient les politiques de sécurité qui doivent être satisfaites pour accéder aux ressources avec une certaine action. Les quatre tables de sécurité suivantes ont été utilisées : Table-de-sécurité -Lire-, Table-de-sécurité -Ecrire-, Table-de-sécurité -Modifier-, Table-de-sécurité -Exécuter-. Ces tables ont été remplies aléatoirement en utilisant la fonction aléatoire du langage C#. Un graphe pondéré d'autorisation dans les grilles (WGAG) est créé pour chaque table de sécurité citée précédemment.
- **Deuxième cas :** une table de sécurité contenant 96 ressources a été créée (un exemple est illustré dans le tableau 5.17). Pour chaque ressource, il existe quatre lignes dans la table, chaque ligne correspond à la politique de sécurité qui doit être vérifiée pour accéder à cette ressource avec une action particulière. Un index indiquant l'action est ajouté au nom

de la ressource tel que : r pour lire, w pour écrire, m pour modifier et u pour l'action exécuter. Les politiques de sécurité utilisées dans le premier cas ont été copiées dans cette table de sécurité. Ensuite, le graphe pondéré d'autorisation dans les grilles (WGAG) est crée en utilisant cette table de sécurité.

Sr / Res	Sr_1	Sr_2	Sr_3	Sr_4	Sr_5	Sr_6	Sr_7
Rr_1	0	1	1	0	1	0	0
Rw_1	1	0	0	1	0	1	0
Rm_1	0	0	1	0	1	1	1
Ru_1	1	0	0	0	1	1	0
Rr_2	0	1	0	0	1	0	0
Rw_2	0	0	0	1	0	1	0
Rm_2	1	0	1	0	1	0	0
Ru_2	1	0	0	0	1	0	0

Tableau 5. 17 Un exemple d'une table de sécurité qui peut être utilisée pour la simulation du deuxième cas

Pour montrer l'efficacité du modèle proposé, nous avons fait plusieurs simulations utilisant les critères suivants : la complexité (nombre de règles de sécurité vérifiées), le nombre des ressources parcourues dans le graphe et le temps de réponse à une requête de contrôle d'accès. En outre, pour démontrer l'efficacité d'un modèle de contrôle d'accès, l'estimation du nombre de faux positifs (Situation où l'accès est accepté alors qu'il devra être refusé et vice versa) est un critère très important.

5.4.3.1. La complexité

100 différents processus d'autorisation ont été initiés. Pour chaque processus l'analyse du WGAG (cas 2) et de l'Action-WGAG (cas 1) a été faite. Les résultats obtenus sont représenté dans la figure 5.15 (l'axe X représente le numéro du processus d'autorisation (Numéro de l'expérience) et l'axe Y représente la complexité d'autorisation (le nombre de règles de sécurité vérifiées)).

Comme le montre la figure 5.15, nous pouvons dire que la complexité d'Action-WGAG est toujours inférieure ou égale à celle du WGAG. Dans certains cas, Action-WGAG vérifie un nombre de règle de sécurité inférieur à celui vérifié par le WGAG, ceci est dû à l'ordre dans lequel l'algorithme a dessiné les règles de sécurité parentes d'une ressource. Nous savons que l'algorithme WGAG fait la somme de chaque colonne de la table de sécurité, puis la règle de sécurité qui aura la somme la plus élevée sera dessinée la première. Ce principe est utilisé

pour les deux cas (cas 1 et cas 2), sauf que les tables de sécurité sont différentes, donc pour une ressource particulière, les règles de sécurité parentes peuvent être dessinées dans des ordres différents dans chaque cas. Cette différence peut augmenter la complexité du WGAG car le nombre de règles de sécurité est très important dans un aussi grand graphe. Dans d'autres cas, l'Action-WGAG et le WGAG vérifient le même nombre de règles car la ressource possède le même niveau de classification dans les deux cas. Donc pour un utilisateur avec une certaine clairance de sécurité, des règles de sécurité dessinées dans le même ordre, le nombre de règles de sécurité à vérifier sera le même dans les deux cas. Enfin, nous pouvons dire que la complexité d'Action-WGAG est toujours inférieure ou égale à celle du WGAG.

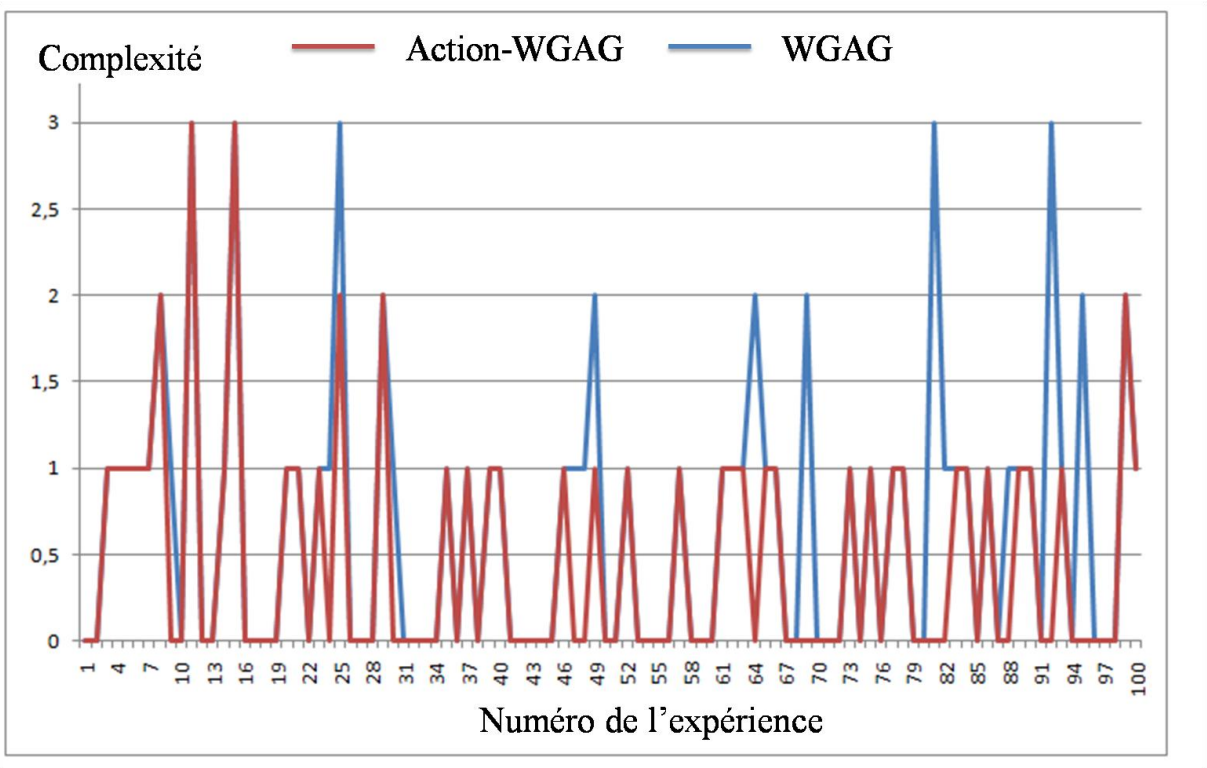

Figure 5. 15 Evaluation de la complexité d'Action-WGAG vs complexité de WGAG

5.4.3.2. Le nombre de ressources parcourues dans le graphe pour trouver la ressource désirée

100 différents processus d'autorisation ont été initiés. Pour chaque processus l'analyse du WGAG (cas 2) et de l'Action-WGAG (cas 1) a été faite. Les résultats obtenus sont représentés dans la figure 5.16 (l'axe X représente le numéro du processus d'autorisation

(Numéro de l'expérience) et l'axe Y représente le nombre de ressources parcourues dans le graphe pour trouver la ressource désirée).

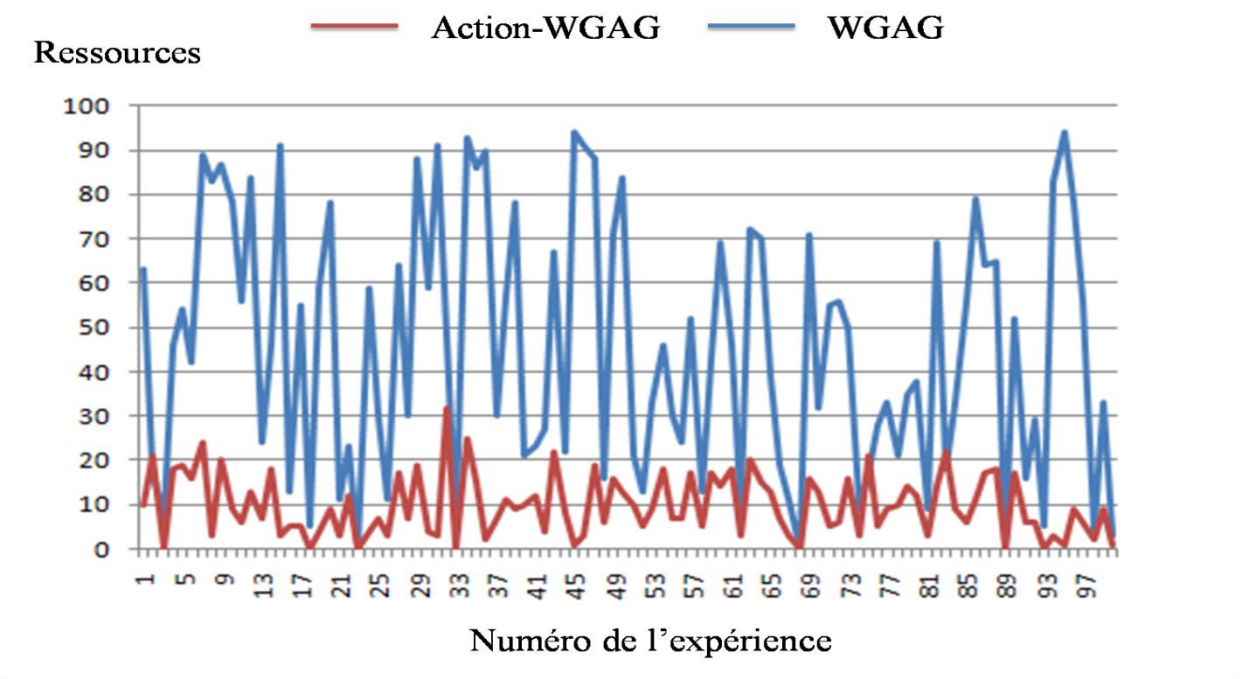

Figure 5. 16 Nombre de ressources parcourues dans le graphe pour trouver la ressource désirée

Selon les résultats illustrés dans la figure 5.16, nous pouvons dire que le nombre de ressources parcourues dans le WGAG pour trouver la ressource désirée est toujours élevé par rapport au nombre de ressources parcourues dans Action-WGAG. Ceci est dû à la taille du graphe WGAG par rapport à celle d'Action-WGAG. Ce dernier comporte 24 ressources au maximum alors que le graphe WGAG contient 96 ressources. Donc trouver la ressource désirée par un utilisateur est plus facile dans le cas d'Action-WGAG que dans le graphe WGAG.

5.4.3.3. Analyse du temps de réponse à une requête de contrôle d'accès

Dans les deux cas : WGAG ou Action-WGAG, la réponse à une requête de contrôle d'accès nécessite le passage par deux étapes : la première qui consiste à rechercher la ressource demandée dans le graphe. Tandis que la deuxième consiste à vérifier les règles de sécurité parentes de cette ressource pour voir si elles sont satisfaites par les rôles des utilisateurs ou non. Par conséquent, le temps de réponse à une requête de contrôle d'accès, sera égal au temps de recherche de la ressource plus le temps de vérification des règles de sécurité parentes. Dans la section 5.4.3.1, nous avons montré que l'Action-WGAG vérifie un nombre

de règles de sécurité inférieur ou égal à celui du WGAG. Ceci nous permet de dire que le temps de vérification des règles de sécurité dans Action-WGAG est inférieur ou égal à celui du temps de vérification des règles dans WGAG. Dans la section 5.4.3.2, nous avons montré que l'Action-WGAG, parcourt moins de ressources que le WGAG pour trouver la ressource demandée. Ceci nous permet de dire que le temps nécessaire pour trouver la ressource demandée dans le cas d'Action-WGAG est inférieur à celui du WGAG. Enfin, en tenant compte de ces deux confirmations, nous pouvons conclure qu'Action-WGAG prendra moins de temps pour répondre à une requête de contrôle d'accès que WGAG.

5.4.3.4. Les faux positifs

Nous avons démontré qu'Action-WGAG prend moins de temps que le WGAG pour répondre à une requête d'accès. Ceci est un point positif, mais reste à savoir si le résultat de la décision d'accès est juste. Dans [Andale, 2015], l'auteur a montré que l'efficacité d'un modèle de contrôle d'accès est mesurée par le nombre de faux positifs. Pour être sûr qu'Action-WGAG prend des décisions d'accès de qualité, nous allons évaluer le nombre de faux positifs.

A : Accepté, R : Refusé

Simulation	100		200		300		400	
Résultat d'accès	A	R	A	R	A	R	A	R
WGAG	32	68	65	135	101	199	144	256
Action-WGAG	32	68	65	135	101	199	144	256

Tableau 5. 18 Evaluation des accès acceptés et refusés

Les résultats obtenus (Tableau 5.18) montrent que le nombre des accès refusés et le nombre des accès acceptés est le même dans le cas de WGAG et Action-WGAG. Donc ce dernier donne la même décision d'accès que WGAG mais en consommant moins de temps et en vérifiant un nombre de règles de sécurité inférieur ou égale à celui du WGAG. Nous pouvons conclure qu'Action-WGAG ne donne pas de faux positifs.

Ce travail a été soumis à un journal et il est en cours d'évaluation. Cette contribution satisfait tous les critères cités dans le tableau 5.9 et ceux qui sont cités dans le tableau 5.10 et cela comme suit :

- **Le mode d'autorisation** : Action-WGAG est basé sur le modèle ABAC, il utilise les attributs des utilisateurs au lieu d'utiliser leurs identités. Ce point permet au modèle d'être plus flexible dans un environnement dynamique tel que les grilles.

- **La validation** : le modèle proposé a été validé en utilisant le simulateur Action-WGAG. Les résultats obtenus ont montré que notre contribution a apporté un aspect positif au processus de contrôle d'accès (autorisation) dans un environnement de grille.
- **La réduction de temps** : ce critère fait référence au temps de réponse à une requête de contrôle d'accès. Nous avons démontré dans la section 5.4.3.3 qu'Action-WGAG réduit ce temps.
- **Les faux positifs** : l'efficacité d'un modèle de contrôle d'accès dépend du nombre de faux positifs qu'il donne. Nous avons montré dans la section 5.4.3.4 qu'Action-WGAG ne donne pas de faux positifs et que la décision d'accès qu'il donne est la même que celle du WGAG sauf que Action-WGAG prend cette décision en peu de temps.
- **La réduction de complexité** : Action-WGAG réduit dans certains cas le nombre de règles de sécurité vérifiées.
- **Elimination de la vérification de certaines règles dès le début** : l'utilisation de l'action qu'un utilisateur veut réaliser permet d'éliminer la vérification d'autres règles de sécurité dès le début (politiques concernant d'autres actions).
- **La répétition dans la vérification des règles de sécurité** : vu qu'Action-WGAG est une amélioration du WGAG et ce dernier n'a pas de répétition dans la vérification des règles de sécurité, nous pouvons dire qu'Action-WGAG n'a pas une répétition également.
- **Expressivité** : nous pouvons dire qu'Action-WGAG est plus expressif que WGAG car il permet de représenter n'importe quelles politiques de sécurité écrites en langage XACML. Il prend en charge tous les éléments nécessaires : ressource, action et règle de sécurité.

5.5. Un modèle de contrôle d'accès à des données organisées d'une manière hiérarchique dans un environnement de Cloud

Dans cette dernière contribution, nous nous sommes intéressés à la sécurité du Cloud Computing, plus précisément au processus de contrôle d'accès. Le Cloud Computing est un modèle permettant d'accéder à un réseau partagé de ressources informatiques configurables

(réseaux, serveurs, stockages, applications et services), qui peuvent être rapidement provisionnées et libérées avec un minimum d'effort de gestion [Mell P. and Grance T., 2011]. Le stockage des données au sein du Cloud est un service qui a atteint un développement sans précédent. C'est ce qui a permis aux propriétaires des données de différentes organisations de stocker leurs données locales sur différentes zones de stockage virtuelles hébergées par le Cloud avec la possibilité d'atteindre leur contenu dès le besoin. Cependant, le paradigme de stockage de données introduira également quelques problèmes de sécurité tout en offrant beaucoup de commodités. Il est évident que les propriétaires de données s'inquiètent que leurs données soient mal utilisées ou accédées par des utilisateurs non autorisés. La confidentialité des données, la préservation de la vie privée ainsi que l'efficacité entravent l'expansion rapide du Cloud. Un mécanisme de contrôle d'accès efficace et sécurisé devient un moyen pour faire face à ce dilemme. En outre, les politiques de sécurité peuvent utiliser différents formats de données allant d'une simple donnée à d'énormes bases, comment gérer l'accès à plusieurs données demandées par une requête quelque soit sa complexité tout en garantissant la souplesse et l'efficacité ? Dans cet article, une extension du standard XACML basée sur ARBAC (Attribute and Role Based Access Control) est proposée afin de garantir la sécurité des données hébergées au Cloud tout en fournissant la possibilité d'accéder à différents niveaux de la structure d'une base de données (niveau base de données, niveau tables ou niveau colonnes) et omettre l'accès à d'autres.

5.5.1. Comparaison entre les différents travaux existants

Dans cette partie, nous avons fait une comparaison entre les travaux existants selon les critères suivants : Les approches utilisées, les techniques, la validation (V), la gestion des attributs (GA), la décomposition des données (DD) et la minimisation des politiques de sécurité à vérifier (MPSV). Ces travaux ont été présentés en détail dans la section 3.3.1.2 du chapitre 3.

L'étude des travaux cités dans le (Tableau 5.19) sur le contrôle d'accès dans un environnement de Cloud, nous a permis de trouver ce qui suit :

☒ Satisfait ● Non satisfait

Modèle	Citation	Approches	V	GA	DD	MPSV
RBAC	Zhu T. et Al., 2011	▪ Authentification inter-domaines ▪ Affectation de rôle inter-domaines ▪ Caches hiérarchiques	☒	●	●	●
	Chunlei W. et Al., 2012	▪ Un rôle quantifié ▪ Valeur de permission ▪ Valeur de comportement	☒	●	●	●
	Sun L. et Al., 2012	▪ Les ontologies ▪ Des vocabulaires structurés ▪ Contrôle d'accès sémantique	●	●	●	●
	Yue-qin F. and Yong-sheng Z., 2012	▪ Un modèle de contrôle d'accès basé sur la tâche ▪ Valeur de réputation	☒	●	●	●
ABAC	Dos Santos D. et Al., 2013 Chen A. et Al., 2016	▪ Contrôle d'accès basé sur le risque	☒	●	●	●
	Khaled R. et Al., 2015	▪ La sensibilité des objets ▪ Politique ABAC ▪ Règles d'attributs	☒	☒	●	●
	Khan F. et Al., 2016	▪ Un modèle de contrôle d'accès basé sur le chiffrement ▪ Plusieurs autorités d'attributs	☒	●	☒	●
ARBAC	Mon E. and Naing T., 2011	▪ Degré de sensibilité des données ▪ Niveau des utilisateurs ▪ Gestionnaire de sécurité	●	●	●	●

Tableau 5. 19 Comparaison entre les travaux existants (contribution N° 4)

- Les modèles de contrôle d'accès basés sur ABAC ont été tous validés dans un environnement de Cloud, tandis que ceux utilisant RBAC ne l'ont pas tous été, ce qui montre la souplesse du modèle ABAC par rapport à RBAC dans un environnement distribué et dynamique tel que le Cloud Computing.
- La gestion des attributs n'est pas applicable dans les modèles basés sur RBAC, car ceux-ci n'utilisent pas d'attributs. Les travaux basés sur ABAC n'ont pas pris en considération ce point, c'est uniquement dans [Khaled R. et Al., 2015] que les auteurs

ont utilisé des règles d'attributs pour déterminer le nombre et le type d'attributs à utiliser lors d'une décision d'accès.

- Dans [Khan F. et Al., 2016], les auteurs ont proposé un modèle de contrôle d'accès qui a permis de spécifier la partie de données accessible par un utilisateur contrairement aux autres travaux qui n'ont pas tenu compte de ce point important qui pourra faire partie des exigences d'un propriétaire de données. Tous les travaux cités auparavant n'ont pas tenu compte de la minimisation du nombre de politiques de sécurité à vérifier, ce qui pourra réduire d'une manière significative le temps de réponse à une requête d'accès.
- Dans [Mon E. and Naing T., 2011], les auteurs ont proposé un modèle de contrôle d'accès basé sur les deux modèles RBAC et ABAC à la fois, ce modèle n'a géré aucun des critères cités auparavant.

5.5.2. Contrôle d'accès à des données organisées d'une manière hiérarchique basé sur les modèles RBAC, ABAC et une extension du standard XACML

5.5.2.1. Motivation

Le contrôle d'accès est un mécanisme qui permet d'assurer la sécurité des données hébergées au sein du Cloud en spécifiant les permissions acceptées pour chaque utilisateur. Selon les travaux cités dans la section précédente, un modèle de contrôle d'accès est basé sur RBAC, ABAC ou ARBAC. Une requête générée par un utilisateur peut demander l'accès à différents types de données. Ces données sont généralement organisées dans des bases de données composées de plusieurs tables. Chaque table contient plusieurs colonnes qui représentent les attributs de la table. D'un point de vue physique, ces données sont vues comme une hiérarchie de données. D'un autre côté le propriétaire peut spécifier la partie des données accessible pour certains utilisateurs et omettre l'accès à une autre partie au sein d'une même table en utilisant des politiques de sécurité spéciales. Comment un modèle de contrôle d'accès peut-il gérer des requêtes d'accès à des données appartenant à différents niveaux hiérarchiques de la structure d'une base de données (niveau base de données, niveau table, niveau colonnes) tout en respectant les politiques de sécurité exigées par le propriétaire des données ? Quel langage pouvons-nous utiliser pour spécifier des politiques de sécurité d'un ensemble de données organisées d'une telle manière tout en assurant un contrôle d'accès efficace ? Comment réduire le nombre de politiques à vérifier afin d'améliorer le processus de contrôle d'accès et

diminuer le temps de réponse à des requêtes d'accès dans un environnement distribué et avec un grand nombre d'utilisateurs tel que le Cloud ?

5.5.2.2. Le modèle proposé

Un environnement de Cloud Computing est souvent composé d'un fournisseur de service (A) qui fournit le service de stockage et une organisation (B) qui fait appel à ce service pour héberger ses données dans des bases de données (i). Le modèle proposé (figure 5.17) se compose des éléments suivants :

(a) le propriétaire des données, est celui qui héberge les données au sein du Cloud, spécifie les règles et les politiques de sécurité.

(b) Les utilisateurs de données : ce sont ceux qui génèrent des requêtes pour manipuler les données.

(c) Le gestionnaire de données : il intercepte les requêtes SQL, extrait les données concernées et les envoie au PEP.

(d) PEP (Policy Enforcement Point) fait partie du modèle XACML, d'habitude nous le trouvons dans le serveur d'autorisation avec les autres composants, il se charge de créer les requêtes XACML et les envoyer au PDP.

(e) le serveur d'autorisation qui est une extension du modèle XACML, contient le PDP (Policy Decision Point) : le module qui permet de vérifier les valeurs des attributs des requêtes d'accès avec les valeurs d'attributs des politiques de sécurité et des métapolitiques. Le PIP (Policy Information Point) est l'élément qui fournit les attributs manquants concernant les utilisateurs de l'organisation. La base de politiques contient toutes les politiques de sécurité concernant les tables et les colonnes. La base des métapolitiques contient les politiques de sécurité concernant les bases de données hébergées.

Dans cette contribution, les deux modèles RBAC et ABAC ont été combinés et déployés dans une architecture XACML distribuée, le choix du modèle RBAC est dû à l'importance qu'il donne aux rôles des utilisateurs par rapport à leurs identités. Dans une organisation, la notion de rôle est importante pour déterminer le contenu que peut consulter un utilisateur, l'utilisation de RBAC au sein de notre modèle a permis de gérer les données de manière efficace sachant qu'un utilisateur peut occuper plusieurs rôles. En outre, un rôle peut être affecté à plusieurs employés ce qui nous permet de gérer les permissions affectées aux rôles et

non les permissions affectées à chaque employé. Le modèle ABAC a été utilisé pour sa souplesse et son dynamisme.

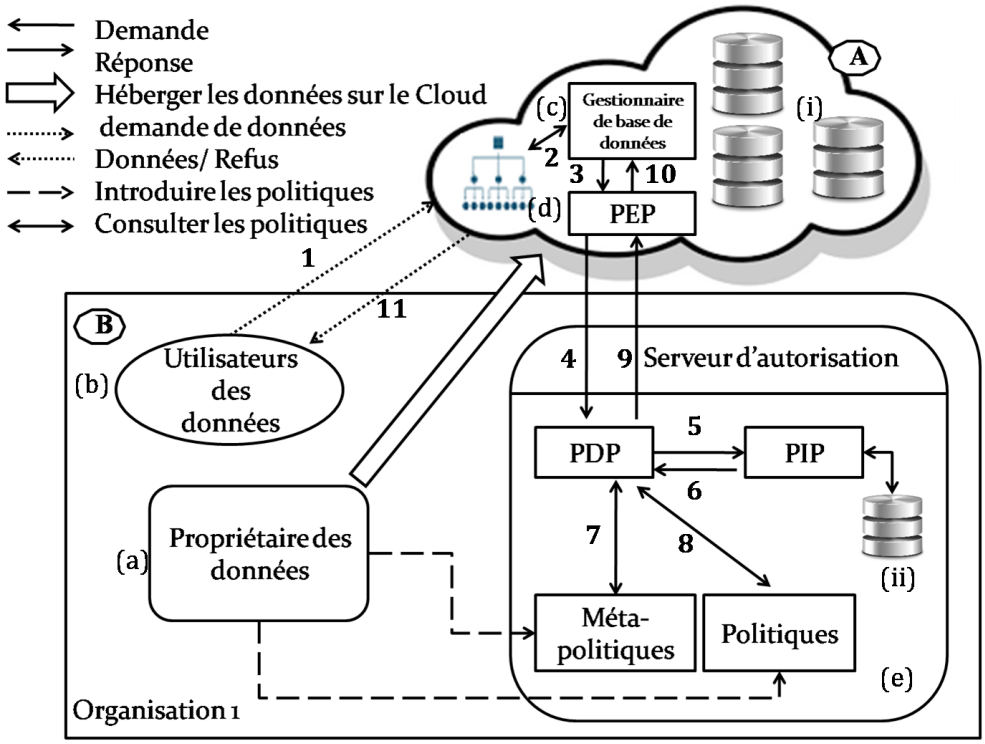

Figure 5. 17 Architecture proposée pour contrôler l'accès à différents niveaux de données dans un environnement de Cloud

Ce modèle vérifie les valeurs des différents attributs avec les objets de la politique d'accès, l'un de ses points faibles est lié à la gestion d'attribut en termes de nombre et de type à utiliser pour prendre une décision concernant une demande d'accès. Pour faire face aux inconvénients des deux modèles, nous les avons fusionnés pour tirer profit des avantages de chacun. Le modèle proposé utilise le standard XACML [Rissanen E., 2017] pour spécifier les politiques de sécurité et les métapolitiques pour des données organisées de manière hiérarchique. D'autre part, le PEP a été mis du côté du fournisseur pour permettre la création des requêtes XACML, tandis que les politiques et leur vérification sont du côté propriétaire pour garantir leur intégrité et éviter leur cryptage qui ajoutera des tâches administratives supplémentaires au propriétaire. La division des politiques en deux ensembles facilitera la tâche à l'administrateur et évitera la vérification de l'ensemble des politiques de sécurité par le PDP ce qui permet d'avoir un modèle plus efficace.

5.5.2.3. Exemple d'un contrôle d'accès à des données d'une organisation hébergées dans un environnement de Cloud

Soit l'exemple suivant : un ensemble de données d'une organisation qui sont organisées dans deux bases de données (BDD 1 et BDD 2) avec différentes tables (achats, ventes,…), chaque table contient plusieurs attributs (Produit, fournisseur, …) (Figure 5.18). Les étapes suivantes sont réalisées lors du passage d'une organisation à une solution Cloud.

Etape 1 : Hébergement des données par le propriétaire

Les données d'une entreprise ou organisation sont souvent sous la forme d'une base de données. Cette dernière est composée de tables et chaque table d'un ensemble de colonnes, l'organisation de ces données est faite de manière hiérarchique car la demande d'accès sera pour une colonne dans une table qui appartient à une base de données. Chaque donnée est identifiée en utilisant son URI [Berners-Lee T. et Al., 2016] car les données peuvent être répétitives, de cette manière l'URI permet de trouver la donnée demandée par l'utilisateur.

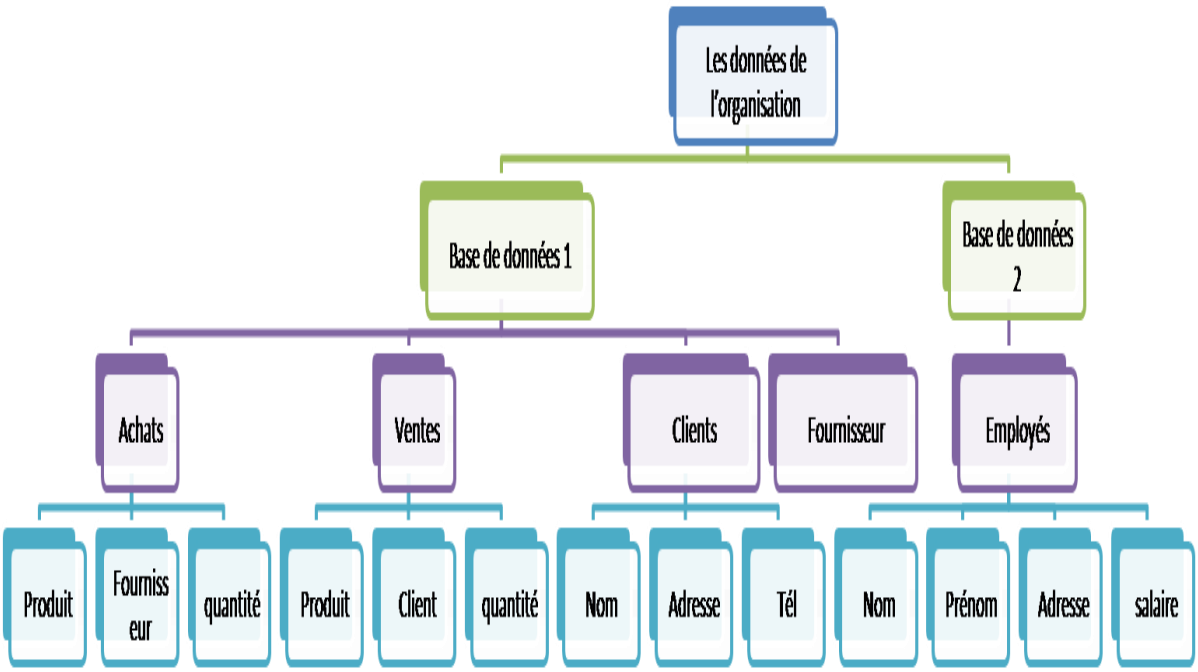

Figure 5. 18 Plan des données du propriétaire

Lorsqu'un propriétaire décide d'héberger ses données dans un Cloud, il crée les bases de données adéquates, crée les tables ainsi que le plan de ses données (Figure 5.18). Le

propriétaire doit définir les politiques d'accès aux bases de données (Métapolitiques) ainsi que les politiques d'accès aux tables et aux colonnes (politiques) en utilisant le rôle des utilisateurs au sein de l'organisation, la spécification de ces politiques est faite en langage XACML 3 pour ressources hiérarchiques avec décisions multiples (Figure 5.19). Les données sont ensuite hébergées sur le Cloud, tandis que les métapolitiques et les politiques sont mises dans les deux bases spécifiques aux politiques au sein de l'organisation.

Etape 2 : Demande d'accès par un utilisateur

Chaque utilisateur au sein de l'organisation, a un ou plusieurs rôles qu'il occupe. Tous ces rôles ainsi que toutes les informations relatives aux utilisateurs sont stockés au niveau de l'organisation dans une base de données au sein du serveur d'autorisation (ii). Lorsqu'un utilisateur U_1 envoie au fournisseur de service une requête de manipulation de données de type (Figure 5.20) : SELECT, UPDATE ou autre (1) ; le gestionnaire de données retire de cette requête les données concernées, cherche dans le plan des données pour trouver la base concernée (2), ensuite il envoie toutes ces informations au PEP (3). Ce dernier crée une requête XACML de type : données hiérarchiques avec décisions multiples et l'envoie au PDP (4). Le PDP retire la base concernée du plan envoyé par le propriétaire, demande au PIP les attributs manquants concernant l'utilisateur (5), le PIP cherche tout le nécessaire concernant les utilisateurs et les données puis les renvoie au PDP (6), ensuite le PDP vérifie les métapolitiques de sécurité (7) , si l'utilisateur peut accéder à cette base, le PDP continue de vérifier les politiques concernant le reste des données demandées (8), sinon il envoie un refus comme réponse au PEP (9) qui la transmettra au gestionnaire de données (10), ce dernier la transmet à l'utilisateur (11). Si l'utilisateur a accès à la base, après vérification de l'accès aux autres données, il envoie sous forme d'une réponse multiple la réponse de chaque donnée demandée (Figure 5.21). Cette réponse multiple sera envoyée au gestionnaire de données. (Illustré dans la Figure 5.17).

Etape 3 : Gestion d'un résultat avec réponse multiple

Lors de la réception du résultat multiple par le gestionnaire de données, il doit faire des tests pour savoir s'il donne l'accès aux données demandées et à quelle partie exactement l'utilisateur y est autorisé. Si le résultat est refusé pour toutes les données alors le gestionnaire de données envoie un refus à la demande de l'utilisateur. Le gestionnaire suit la règle

suivante : Si l'accès à une donnée de niveau supérieur dans la hiérarchie est refusé, alors que l'accès pour l'une de ses descendantes est accepté alors le résultat sera refusé, sinon il sera le même que celui obtenu, si l'utilisateur a accès à une partie de données et pas à une autre, le gestionnaire lui demandera de refaire sa requête car il demande l'accès à des données non autorisées (Figure 5.22).

```
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17" PolicyId="Resource-Five-Get-Policy"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides" Version="1.0">
  <Target> <AnyOf> <AllOf>
<Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
  <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">BDD1/ACHAT/PRODUIT</AttributeValue>
  <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
 Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource" DataType="http://www.w3.org/2001/XMLSchema#string"
 MustBePresent="true"/> </Match> </AllOf> </AnyOf> </Target>
<Rule Effect="Permit" RuleId="Rule-1"> <Target><AnyOf><AllOf>
<Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
 <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">SELECT</AttributeValue>
 <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action" DataType="http://www.w3.org/2001/XMLSchema#string"
MustBePresent="true"/> </Match> </AllOf> </AnyOf> </Target> <Condition>
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
  <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
  <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">GESTIONNAIRE_STOCK</AttributeValue>
  <AttributeDesignator AttributeId="http://wso2.org/claims/role" Category="urn:oasis:names:tc:xacml:1.0:subject-
category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
</Apply> </Condition>
</Policy>
```

Figure 5. 19 Un exemple d'une politique de sécurité écrite en XACML 3 qui dit que pour accéder à la colonne produit de la table achat de la base de données 1 avec une action Select il faut que l'utilisateur ait un rôle « Gestionnaire de Stock ».

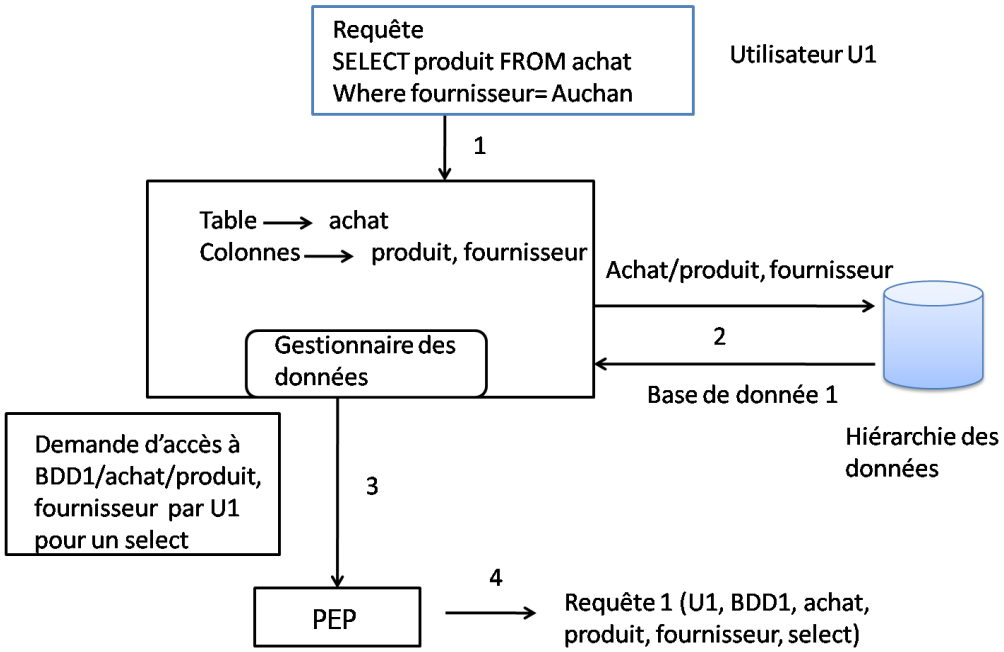

Figure 5. 20 Demande d'accès au niveau du fournisseur

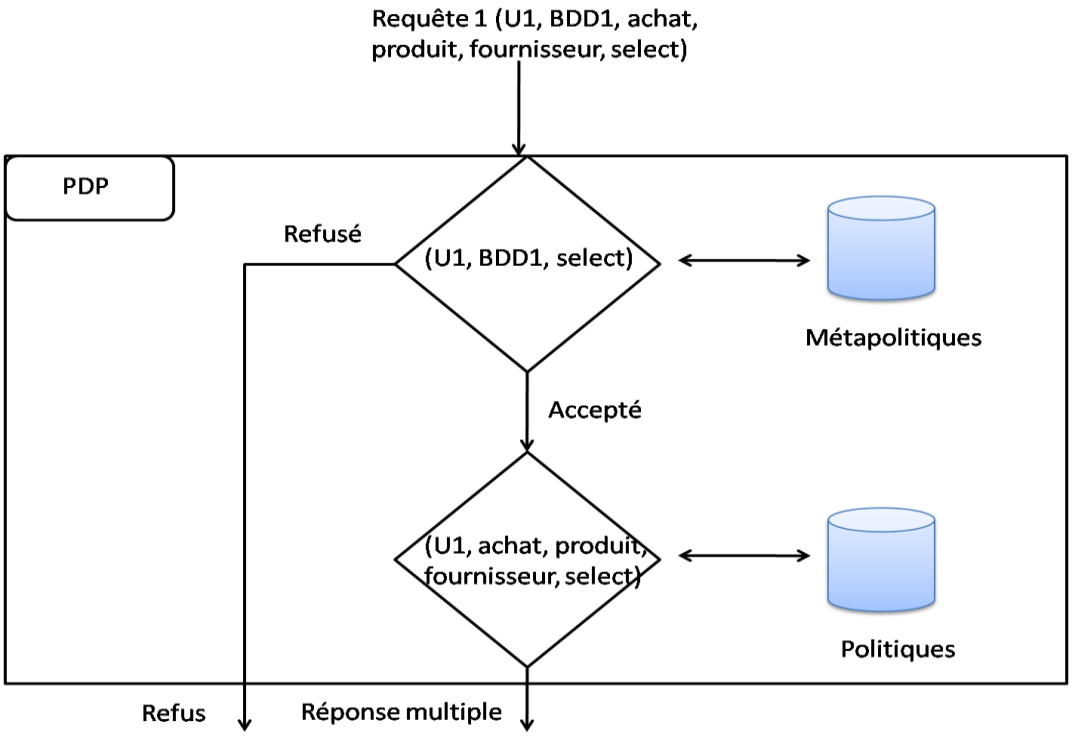

Figure 5. 21 La prise de décision par le PDP

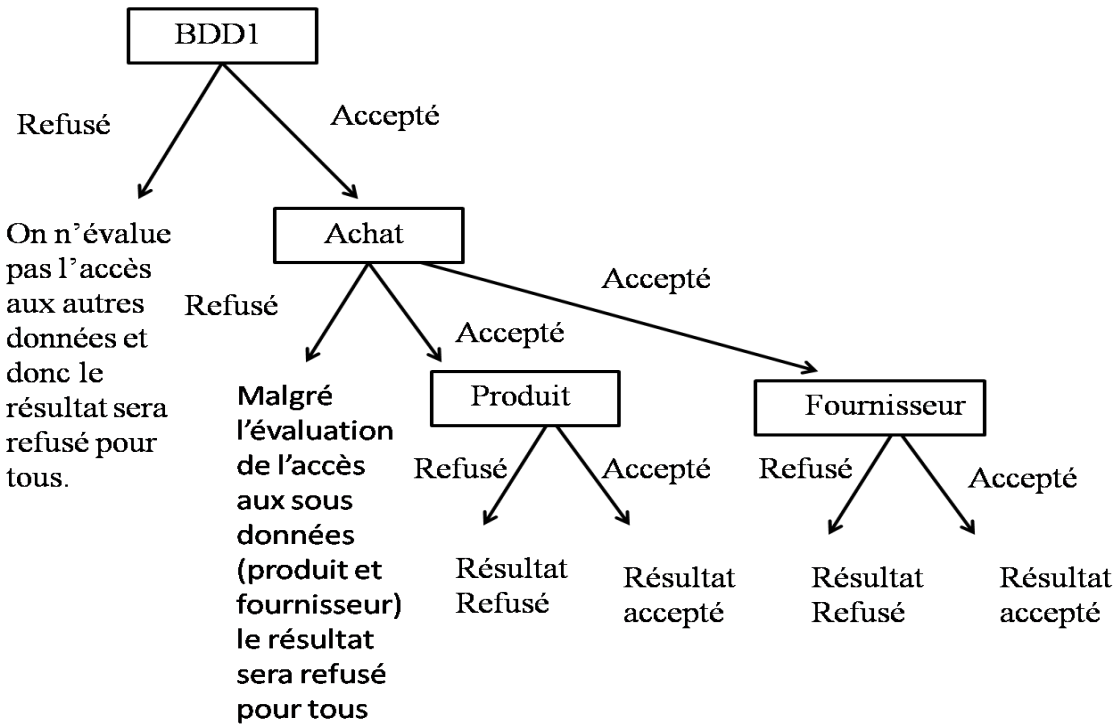

Figure 5. 22 Exemple d'un résultat multiple

Ce travail a été présenté à la conférence « Advances on Decisional Systems (ASD 2017) », Dans cette contribution nous avons proposé un modèle de contrôle d'accès qui est une extension du standard XACML basée sur ARBAC. Ce modèle a de nombreux avantages tels que : la combinaison des deux modèles RBAC et ABAC qui permet de profiter des avantages de ces deux modèles, la prise en charge du niveau de données auquel l'utilisateur peut accéder ce qui facilite la tâche au propriétaire et lui permet de gérer l'accès à ces données selon le niveau souhaité. Le modèle proposé utilise le standard XACML 3 pour la spécification des politiques de sécurité et des métapolitiques, ce standard permet de gérer les demandes d'accès à des ressources hiérarchiques avec des décisions multiples ce qui réduit le parcours de longs fichiers XML de politiques et donne un contrôle d'accès efficace. L'un des points qui reste à établir, est l'implémentation du modèle de contrôle d'accès proposé dans un environnement du Cloud Computing. Donc, nous pouvons dire que le seul point négatif de ce modèle réside dans le fait qu'il n'a pas encore été testé.

5.6. Conclusion

Dans ce chapitre, nous avons présenté nos quatre contributions. Dans la première contribution, nous avons proposé un modèle de contrôle d'accès parallèle qui prend en charge la collaboration dans un environnement de grilles inter-domaines. Dans la deuxième contribution, nous nous sommes intéressés aux mécanismes de stockage et de gestion des politiques de sécurité dans les grilles, nous avons proposé le graphe pondéré d'autorisation dans les grilles (WGAG), ce mécanisme a pu éliminer la vérification de certaines règles de sécurité dès le début, mais il ne permettait pas de représenter toutes les politiques de sécurité réelles. C'est ce qui nous a poussés à proposer le graphe pondéré basé sur l'action pour l'autorisation dans les grilles (Action-WGAG) avec le parseur de politique de sécurité (SP-Parser). Enfin, nous avons proposé une extension du standard XACML basée sur le modèle ARBAC pour contrôler l'accès à différents niveaux de données hébergées dans un environnement de Cloud.

Conclusion générale

Conclusion générale

Plusieurs issues de sécurité telles que : l'authentification, le contrôle d'accès, l'intégrité et la confidentialité, ont été traitées dans les grilles et dans le Cloud Computing. Le contrôle d'accès représente l'issue de sécurité la plus considérée par les chercheurs car elle peut assurer plusieurs dimensions de sécurité en même temps. Un processus de contrôle d'accès englobe deux parties : la première consiste à trouver la meilleure manière de représenter les politiques de sécurité soumises par l'administrateur. Puis, comment gérer ce mécanisme de manière à avoir un contrôle d'accès efficace. Tandis que la deuxième, elle prend en considération la manière de répondre à une requête d'accès efficacement et rapidement.

1. Les contributions

Dans notre thèse, nous avons proposé quatre contributions. Dans la première, nous avons présenté un modèle de contrôle d'accès parallèle qui permet d'assurer la collaboration inter-domaines dans un environnement de grille. Les résultats de simulations obtenus ont montré que le modèle proposé est efficace car il réduit le temps de réponse à une requête d'accès, de plus la décision d'accès ne contient pas de faux positifs. Ensuite, dans notre deuxième contribution, nous nous sommes focalisés sur les mécanismes de stockage et de gestion des politiques de sécurité dans un environnement de grille. Nous avons trouvé que le graphe d'autorisation dans les grilles (GAG) était le meilleur mécanisme disponible car il a pu éliminer toute répétition lors de la vérification des règles de sécurité. Mais, nous avons pu constater que certaines vérifications peuvent être éliminées dès le début dans le GAG, c'est ce qui nous a permis de proposer le graphe pondéré d'autorisation dans les grilles (WGAG). Après évaluation de ce mécanisme, nous avons trouvé qu'il est meilleur par rapport aux mécanismes. Un des points négatifs de ce mécanisme, réside dans le fait qu'il ne peut pas être appliqué à toutes les politiques de sécurité d'un système réel car il ne contient pas tous les éléments nécessaires. Ceci nous a éclairé le chemin pour proposer notre troisième contribution. Cette dernière consiste en un graphe pondéré basé sur l'action pour l'autorisation dans les grilles (Action-WGAG). Cette contribution a donné de meilleurs résultats que le WGAG. En outre, elle a permis la proposition d'un parseur de politique de sécurité (SP-Parser) qui pourra être très utile pour transformer de longs fichiers de politiques de sécurité à de simples tables de sécurité. L'évaluation de ce modèle a pu démontrer qu'il est plus efficace que le WGAG car : le nombre de règles de sécurité vérifiées est inférieur ou égal à celui des règles vérifiées par WGAG, le nombre de nœuds parcourus pour trouver la ressource demandée est inférieur à celui du WGAG, la décision d'accès ne contient pas de

faux positifs et le temps de réponse à une requête d'accès est inférieur à celui du WGAG. Enfin, notre dernière contribution s'est focalisée sur le contrôle d'accès dans un environnement de Cloud. Le seul point négatif de cette dernière est qu'elle n'a pas encore été testée.

2. Perspectives

Dans nos travaux futurs, nous allons d'abord tester l'efficacité de notre dernière contribution dans un environnement de Cloud. Nous envisageons aussi d'importer le graphe pondéré basé sur l'action pour l'autorisation dans les grilles (Action-WGAG) dans un environnement de Cloud pour améliorer le stockage et la gestion des règles de sécurité. De plus, nous allons essayer de proposer des mécanismes de contrôle d'accès basés sur les modèles que nous avons proposés en assurant plus de dimensions de sécurité.

Les références

Références

[Abou El Kalam A. et Al., 2003] Abou El Kalam A., El Baida R. and Balbiani P, (2003). Organization based access control. Proceedings policy 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks, DOI: 10.1109/POLICY.2003.1206966.

[Acqnotes, 2018] Acqnotes, (2018). Global Information Grid (GIG). http://acqnotes.com/acqnote/careerfields/global-information-grid-gig (Consulté en 2018).

[Akihiko K., 2006] Akihiko K. (2006), Trends in life science grid: from computing grid to knowledge grid, BMC Bioinformatics. 2006; 7(Suppl 5): S10. Published online 2006 Dec 18. doi: 10.1186/1471-2105-7-S5-S10.

[Al-Attab B.S. and Fadewar H.S., 2016] Al-Attab B.S. and Fadewar H.S., (2016). Authentication Scheme for Insecure Networks in Cloud Computing, 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication.

[Alfieri R. et Al., 2003] Alfieri R. , Cecchini R., Ciaschini V. , dell'Agnello L. , Frohner Á. , Gianoli A., Lörentey K., Spataro F. (2003). Voms: an authorization system for virtual organizations. European Across Grids Conference, pp. 33-40.

[Al-Muhtadi J., 2000] Al-Muhtadi J., (2000). The A-IRBAC 2000 model: Administrative interoperable role-based access control. Technical Report, UIUCDCS.

[Al-Riyami S.S. and Paterson K.G., 2003] Al-Riyami S.S. and Paterson K.G., (2003) Certificateless Public Key Cryptography. In: Laih CS. (eds) Advances in Cryptology - ASIACRYPT 2003. ASIACRYPT 2003. Lecture Notes in Computer Science, vol 2894. Springer, Berlin, Heidelberg.

[Aluvalu R.K. and Muddana L., 2016] Aluvalu R.K. and Muddana L., (2016). A dynamic attribute-based risk aware access control model (DA- RAAC) for cloud computing, 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), DOI: 10.1109/ICCIC.2016.7919618.

[Andale, 2015] Andale, (2015). False Positive and False Negative: Definition and Examples. http://www.statisticshowto.com/false-positive-definition-and-examples/ (Consulté en 2015).

[Anderson R.J., 2008] Anderson R.J., (2008). Security Engineering: A Guide to Building Dependable Distributed Systems. Pubished by Wiley Publishing Inc, ISBN: 978-0-470-06852-6.

[Ashrafijoo B. et Al., 2010] Ashrafijoo B., Navin A.H., Nia M.M., Abedini S., Azari N., (2010).Trust Management in Grid Computing Systems Based on Probability Theory, 201O 2nd International Conference on Education Technology and Computer (ICETC).

[Autrel F. et Al., 2008] Autrel F., Cuppens C., Cuppens-Boulahia N. and Coma C., (2008). MotOrBAC 2: a security policy tool. In: Proceedings of the 3rd Joint Conference on Security in Network Architectures (SAR) and Security of Information Systems(SSI) , Loctudy, France, Editions Publibook, pp. 273–288.

[Auxilia M and Raja K., 2014] Auxilia M and Raja K., (2014). Dynamic Access Control Model for Cloud Computing, 2014 Sixth International Conference on Advanced Computing (ICoAC), DOI: 10.1109/ICoAC.2014.7229744.

Références

[Ayache M. et Al., 2015] Ayache M., Erradi M and Freisleben B., (2015). Access Control Policies Enforcement in a Cloud Environment: Openstack; 2015 11th International Conference on Information Assurance and Security (IAS).

[Baktash H.A. et Al., 2010] Baktash H.A.; Karimi M.B.; Meybodi M.R. and Bouyer A., (2010). 2L-RBACG: A new framework for resource access control in grid environments, 2010 Fifth International Conference on Digital Information Management (ICDIM), DOI: 10.1109/ICDIM.2010.5662244.

[Balon N., Thabet I., 2004] Balon N., Thabet I., (2004). The Biba Security Model. https://pdfs.semanticscholar.org/7360/c680906617622f27ef2596c7efcc902795db.pdf (Consulté en 2017).

[Balusamy B. et Al., 2017] Balusamy B., Krishna P.V., Tamizh Arasi G.S., and Chang, V., (2017). A Secured Access Control Technique for Cloud Computing Environment Using Attribute Based Hierarchical Structure and Token Granting System. International Journal of Network Security, Vol.19, No.4, PP.559-572, July 2017 (DOI: 10.6633/IJNS.201707.19(4).9).

[Barkley J. et Al., 1999] Barkley J., Beznosoz K. and Uppal J., (1999). Supporting Relationships in Access Control Using Role Based Access Control. Proceeding of the ACM workshop on RBAC, Fairfax, Virginia, USA, 28-29 October.

[Berners-Lee T. et Al., 2016] Berners-Lee T., Fielding R. and Masinter N. (2016). Uniform Resource Identifiers (URI): Generic Syntax, IETF RFC 3986, https://www.ietf.org/rfc/rfc3986.txt (consulté en 2016).

[Bethencourt J. et Al., 2007] Bethencourt J., Sahai A. and Waters B., (2007). Ciphertext-policy attribute-based encryption. In IEEE Symposium on Security and Privacy, pages 321–334.

[Bhowmick A. et Al., 2012] Bhowmick A. , Koner C. and Bhunia C.T., (2012). A Novel Time based Authentication Technique for Enhancing Grid Computing Security . National Conference on Communication Technologies & its impact on Next Generation Computing CTNGC (2012).

[Biba M., 2017] Biba M., Multilateral Security, Secure Systems and Applications. http://www.marenglenbiba.net/seceng/lesson13.pdf (Consulté en 2017)

[Bokefode Jayant. D. et Al., 2014] Bokefode Jayant. D., Ubale Swapnaja A., Apte Sulabha S., Modani Dattatray G., (2014). Analysis of DAC MAC RBAC Access Control based Models for Security. International Journal of Computer Applications (0975 – 8887) Volume 104 – No.5, October 2014.

[Brighthub, 2016] Brighthub, (2016). Grid Computing - Definition and Disadvantages. http://www.brighthub.com/environment/green-computing/articles/107038.aspx (Consulté en 2016).

[Buyya R. et Al., 2011] Buyya R., Broberg J., Goscinski A., (2011). CLOUD COMPUTING Principles and Paradigms. Published by John Wiley & Sons, Inc., Hoboken, New Jersey Published simultaneously in Canada.

Références

[Buyya R. et Al., 2013] Buyya R., Vecchiola C., ThamaraiSelvi S., (2013). Mastering Cloud Computing Foundations andApplications Programming. ISBN: 978-0-12-411454-8; MorganKaufmann (ELSEVIER).

[Buyya R et Al., 2009] Buyya R., Yeoa C.S., Venugopal S., Broberg J. and Brandic I., (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems 25 (2009) 599–616.

[Cebuc E. et Al., 2010] Cebuc E., Suciu A., MartonK., Dolha S. and Muresan L., (2010). Implementation of cryptographic algorithms on a Grid infrastructure, 2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR) Year: 2010, Volume: 2 Pages: 1 - 6, DOI: 10.1109/AQTR.2010.5520814.

[Ceccanti A. et Al., 2015] Ceccanti A., Tschopp V., Jouvin M. and Caberletti M., (2015). Argus. Retrieved June 2, 2017 from http://argus-documentation.readthedocs.io/en/latest/release_notes.html#v1-7-0.

[Chakrabarti A., 2007] Chakrabarti A., (2007). Grid Computing Security. Library of Congress Control Number: 2007922355 ACM Computing Classification (1998): C.2, D.4.6, K.6.5 ISBN 978-3-540-44492-3 Springer Berlin Heidelberg New York.

[Chen A. et Al., 2016] Chen A., Xing H. , She K. and Duan G., (2016). A Dynamic Risk-based Access Control Model for Cloud Computing, IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom).

[Chen M. et Al., 2010] Chen M.,Wu K., Wu C. and Wu Z.,(2010). Certificateless Signature Based Authenticated Key Agreement Protocol for Grid, 2010 Fifth Annual ChinaGrid Conference, DOI: 10.1109/ChinaGrid.2010.52.

[Chugh S. and Peddoju S.K., 2012] Chugh S. and Peddoju S.K., (2012). Access Control Based Data Security in Cloud Computing, Sonam Chugh, Sateesh Kumar Peddoju/ International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012, pp.2589-2593.

[Chunlei W. et Al., 2012] Chunlei W., Zhongwei L. and Xuerong C,(2012). An Access Control Method of Cloud Com-puting Resources Based on Quantified-Role, 14th International Conference on Communication Technology, IEEE.

[Cody E et Al., 2008] Cody E., Sharman R., Rao R.H. and Upadhyaya S.,(2008), Security in grid computing: A review and synthesis, Decision Support Systems Volume 44, Issue 4, March 2008, Pages 749-764.

[Computerweekly, 2018] Computerweekly, (2018). A history of cloud computing. http://www.computerweekly.com/feature/A-history-of-cloud-computing (Consulté en 2018).

[Computerworld, 2016] Computerworld, (2016). TeraGrid project awarded $150M. https://www.computerworld.com/article/2557127/computer-hardware/teragrid-project-awarded--150m.html (Consulté en 2016).

Références

[Crandall R., 2018] Crandall R., (2018). L'inventeur des grilles de calcul, http://www.itespresso.fr (Consulté en 2018).

[Cuppens F. et Al., 2006] Cuppens F., Cuppens-Boulahia N. and Coma C., (2006). MotOrBAC : un outil d'administration et de simulation de politiques de securité. First joint conference on security in network architectures (SAR) and security of information systems (SSI), 6-9 june, Seignosse, Landes, France, 2006.

[Dinesh C, 2018] Dinesh C, (2018). Data Integrity and Dynamic Storage Way in Cloud Computing, https://arxiv.org/abs/1111.2418 (Consulté en 2018)

[Dinesha H.A. and Agrawal V.K., 2012] Dinesha H.A. and Agrawal V.K., (2012). Multi-level authentication technique for accessing cloud services, 2012 International Conference on Computing, Communication and Applications, DOI: 10.1109/ICCCA.2012.6179130.

[Dos Santos D. et Al., 2013] Dos Santos D. Westphall C.M. and Westphall C.B, (2013).Risk-based Dynamic Access Control for a Highly Scalable Cloud Federation, SECURWARE 2013: The Seventh International Conference on Emerging Security Information, Systems and Technologies.

[DTIC, 2017] DTIC, (2017).Campus-Wide Computing: Early Results Using Legion at the University of Virginia. http://www.dtic.mil/dtic/tr/fulltext/u2/a447083.pdf. (Consulté en 2017).

[Dugénie P., 2006] Dugénie P., (2006). Orientations et Usages de l'Architecture de Services Grid OGSA. https://hal-lirmm.ccsd.cnrs.fr/lirmm-00121616/document (Consulté en 2017).

[eci, 2018] eci, (2018).The History of Cloud Computing. https://www.eci.com/cloudforum/cloud-computing-history.html (Consulté en 2018).

[eci, 2018]^{b} eci, (2018). The Eze Castle Integration Story. https://www.eci.com/about/about-us/our-story.html (Consulté en 2018).

[EGEE-II, 2014] EGEE-II, (2014). An overview of grid middleware and gLite, EGEE-II INFSO-RI-031688, www.eu-egee.org (Consulté en 2014).

[Ennahbaoui M. and Elhajji S., 2013] Ennahbaoui M. and Elhajji S., (2013). Study of Access Control Models. Proceedings of the World Congress on Engineering 2013 Vol II, WCE 2013, July 3 - 5, 2013, London, U.K.

[Erwin D.W. and Snelling D.F, 2001] Erwin D.W. and Snelling D.F. (2001). UNICORE: A Grid Computing Environment. In: Sakellariou R., Gurd J., Freeman L., Keane J. (eds) Euro-Par 2001 Parallel Processing. Euro-Par 2001. Lecture Notes in Computer Science, vol 2150. Springer, Berlin, Heidelberg.

[Farouk A. et Al., 2012] Farouk A., Abdelhafez A.A. and Fouad M.A., (2012). Authentication Mechanisms in Grid Computing Environment: Comparative Study. 2012 International Conference on Engineering and Technology (ICET).

[Ferraiolo D. et Al., 2016] Ferraiolo D., Chandramouli R., Kuhn R. and Hu V., (2016). Extensible Access Control Markup Language (XACML) and Next Generation Access Control (NGAC). ABAC'16: 2016 ACM International Workshop on Attribute Based Access Control Proceedings.

Références

[Firesmith D., 2004] Firesmith D., (2004). Specifying Reusable Security Requirements, ETH Zurich, Chair of Software Engineering, Vol. 3, No. 1, January-February –(2004).

[Foster I., 2005] Foster I., (2005). Globus toolkit version 4: software for service-oriented systems. Published in: Proceeding NPC'05 Proceedings of the 2005 IFIP international conference on Network and Parallel Computing Pages 2-13 ISBN:3-540-29810-X 978-3-540-29810-6 doi>10.1007/11577188_2.

[Foster I. and Kesselman C., 2002] Foster I. and Kesselman C., (2002). The Grid 2, Second Edition: Blueprint for a New Computing Infrastructure (The Elsevier Series in Grid Computing) 2nd Edition. Elservier, ISBN: 1-55860-933-4 (2002).

[Foster I. et Al., 2002] Foster I., Kesselman C., Nick J.M. and Tuecke S., (2002). Grid Services for Distributed System Integration, Journal Computer Volume 35 Issue 6, June 2002 Page 37-46 IEEE Computer Society Press Los Alamitos, CA, USA.

[Foster I. et Al., 1998] Foster, I., Kesselman, K. and Tzudik, G. (1998) A Security Architecture for Computational Grids. Proceedings of the 5th ACM Conference on Computer and Communication Security, San Francisco, 83-92.

[Foster I et Al., 2008] Foster I., Zhao Y., Raicu I. and Lu S., (2008), Cloud Computing and Grid Computing 360-Degree Compared, Grid Computing Environments Workshop, 2008. GCE '08, DOI: 10.1109/GCE.2008.4738445

[Gentzsch, W. et Al., 2011] Gentzsch, W., Girou, D., Kennedy, A. et al. J Grid Computing (2011) 9:259. https://doi.org/10.1007/s10723-011-9183-2 (Consulté en 2018).

[Globus, 2016] Globus, (2016). Globus Toolkit 3.2.0 release notes, http://toolkit.globus.org/toolkit/releasenotes/3.2.0/ (Consulté en 2016).

[Globus, 2018] Globus, 2018. Globus Toolkit, http://toolkit.globus.org/toolkit/about.html (consulté en 2018).

[Globus, 2016]^{b} Globus, (2016). News About the Globus Toolkit. http://toolkit.globus.org/toolkit/news.html#164(Consulté en 2016).

[Globus, 2016]^{c} Globus, (2016). GT Data Management: Replica Location Service (RLS). http://toolkit.globus.org/toolkit/data/rls/ (Consulté en 2016).

[Globus, 2017] Globus, (2017). Overview of the I-WAY: Wide Area Visual Supercomputing. http://toolkit.globus.org/alliance/publications/papers/iway_overview.pdf (Consulté en 2017).

[Gonzalez N.M. et Al., 2013] Gonzalez N.M., Torrez Rojas M.A. and Maciel da Silva M.V., (2013). A framework for authentication and authorization credentials in cloud computing, 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications.

Références

[Gouglidis A. and Mavridis I., 2012] Gouglidis A. and Mavridis I., (2012). Grid access control models and architectures. Computational and Data Grids: Principles, Applications and Design. DOI: 10.4018/978-1-61350-113-9.ch008.

[grid-deployment, 2017] grid-deployment, (2017). gLite. http://grid-deployment.web.cern.ch/grid-deployment/glite-web/introduction (Consulté en 2017).

[Gtnews, 2017] Gtnews, (2017). Grid Computing Adoption in the Financial Services Sector, https://www.gtnews.com/articles/grid-computing-adoption-in-the-financial-services-sector/ (Consulté en 2017).

[Gupta B. et Al., 2011] Gupta B.; Kaur H.; Bedi N. and Bedi P., (2011). Trust Based Access Control for Grid Resources, 2011 International Conference on Communication Systems and Network Technologies, DOI: 10.1109/CSNT.2011.146.

[Haddad A., 2005] Haddad A., (2005). MODELISATION ET VERIFICATION DE POLITIQUES DE SECURITE. These de DEA, Université Joseph Fourier sciences.technologie.santé, 2005.

[Hashemi S.M. and Bardsiri A.K., 2012] Hashemi S.M. and Bardsiri A.K., (2012). Cloud Computing Vs. Grid Computing. ARPN Journal of Systems and Software ISSN 2222-9833, VOL. 2, NO.5, MAY 2012.

[Hedayati M. et Al., 2010] Hedayati M., Kamali S.H. and Shakerian R., (2010). Using Identity-Based Secret Public Keys Cryptography for Heuristic Security Analyses in Grid Computing, 2010 5th International Symposium on Telecommunications (IST'2010), DOI: 10.1109/ISTEL.2010.5734028.

[Hogan M et Al., 2011] Hogan M., Liu F., Sokol A. and Tong J., (2011). NIST Cloud Computing Standards Roadmap. National Institute of Standards and Technology Special Publication 500-291 Natl. Inst. Stand. Technol. Spec. Publ. 500-291, 83 pages (5 July 2011).

[Hu V.C. et Al., 2006] Hu V.C., Ferraiolo D.F., Kuhn D.R., (2006). Assessment of Access Control Systems. National Institute of Standards and Technology Interagency Report 7316, 60 pages (September 2006).

[Hu C.V. et Al., 2014] Hu C.V., Ferraiolo D.F., Kuhn D.R., Schnitzer A., Sandlin S., Miller R., Scarfon K., (2014). Guide to Attribute Based Access Control (ABAC) Definition and Considerations. National Institute of Standards and Technology Special Publication 800-162 Natl. Inst. Stand. Technol. Spec. Publ. 800-162, 45 pages (January 2014).

[IGI Global, 2018] IGI Global, (2018). What is Information Grid, https://www.igi-global.com/dictionary/information-grid/14411(Consulté en 2018).

[ikbooks, 2017] ikbooks, (2017). Cloud computing Architecture. https://www.ikbooks.com/home/samplechapter?filename=92_Sample_Chapter.pdf (Consulté en 2017).

[Internet 2, 2017] Internet 2, (2018). TeraGrid Project. https://spaces.internet2.edu/display/InCCollaborate/TeraGrid+Project (Consulté en 2017).

Références

[Internet2, 2018] Internet2, (2018). SHIBBOLETH. https://www.internet2.edu/products-services/trust-identity/shibboleth/ (consulté en 2018).

[Inria, 2017] Inria, (2017). GRID'5000 Plate-forme de recherche experimentale en informatique. https://www-sop.inria.fr/aci/grid/public/Library/rapport-grid5000-V3.pdf (Consulté en 2017).

[in2p3, 2018] in2p3, (2018). Grille de calcul : l'internet du calcul intensif. http://www.in2p3.fr/presentation/thematiques/grille/grille.htm (Consulté en 2018).

[iphc, 2017] iphc, (2017). Grille de calcul. http://www.iphc.cnrs.fr/IMG/Grille-IPHC_gazette-IPHC-juillet 2009.pdf (Consulté en 2017).

[IRMA, 2018] Information Resources Management Association (IRMA), 2012. Grid and Cloud Computing: Concepts, Methodologies, Tools and Applications (4 Volumes). https://books.google.dz/books id=ulKr4CXuQW4C&pg=PA13&lpg=PA13&dq=1992+(Smarr+and+Catlett&source=bl&ots=yZTEDCldWY&sig=idZ8XUIMaUCw8Zs840VS2XYzkJU&hl=fr&sa=X&ved=0ahUKEwjuIT34pvZAhXHUhQKHSoiBWcQ6AEITDAI#v=onepage&q=1992%20(Smarr%20and%20Catlett&f=false (Consulté en 2018).

[Jacq N., 2006] Jacq N., (2006), WISDOM : Grid-enabled Virtual High Throughput Screening, Biomed meeting, Lyon, 2006/04/28.

[Jaspher G.W.K. et Al., 2011] Jaspher G.W.K., Raj B.E. and Kirubakaran E., (2011). A novel security framework for computational grid, 2011 3rd International Conference on Electronics Computer Technology Year: 2011, Volume: 1 Pages: 103 - 107, DOI: 10.1109/ICECTECH.2011.5941569.

[Jianan H. et Al., 2017] Jianan H., Kaiping X., Wei L., & Yingjie X. (2017). TAFC: Time and Attribute Factors Combined Access Control on Time-Sensitive Data in Public Cloud. IEEE Transactions on Services Computing (Volume: PP, Issue: 99), DOI: 10.1109/TSC.2017.2682090, ISSN: 1939-1374.

[Joshy J. and Craig F., 2003] Joshy J. and Craig F., (2003), Grid computing, Prentice Hall PTR Pub Date : December 30, 2003 ISBN : 0-13-145660-1 Pages : 400.

[journaldunet, 2018] journaldunet, (2018). Microsoft a 40 ans : retour sur une incroyable histoire. http://www.journaldunet.com/solutions/dsi/les-40-ans-de-microsoft-1504.shtml (Consulté en 2018).

[Julius M. and Elyjoy M., 2017] Julius M. and Elyjoy M. (2017), Grid Computing For Collaborative Research Systems In Kenyan Universities. The International Journal of Engineering and Science (IJES) || Volume || 6 || Issue || 4 || Pages || PP 24-31 || 2017 || ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805 DOI: 10.9790/1813-0604022431 www.theijes.com Page 24.

[Kaiiali M. et Al., 2010] Kaiiali M., Wankar R.; Rao C.R. and Agarwal A., (2010). New Efficient Tree-Building Algorithms for Creating HCM Decision Tree in a Grid Authorization System, 2010 Second International Conference on Network Applications, Protocols and Services, DOI: 10.1109/NETAPPS.2010.8.

Références

[Kaiiali, M. et Al., 2013] Kaiiali, M., Wankara, R., Rao, C.R., Agarwal, A. and Buyya R, (2013). Grid Authorization Graph. Future Generation Computer Systems 29 1909–1918.

[Kassid A. and El Kamoun N., 2016] Kassid A. and El Kamoun N., (2016). Evaluation of a Security Policy Based on OrBAC Model Using MotOrBAC: Application E-learning. Advances in Ubiquitous Networking Proceedings of UNet'15.

[Kaur J., 2013] Kaur J., (2013). Single Sign-On Under Grid Security. IJCST Vol. 4, Issue 2, April - June 2013, ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print).

[Kaustav, R. and Avijit, B., 2012] Kaustav, R. and Avijit, B., (2012). A Proposed Mechanism for Cross-Domain Authorization in Grid Computing Environment. International Journal of Emerging Technology and Advanced Engineering ISSN 2250-2459, Volume 2, Issue 4.

[Kazemi A., 2014] Kazemi A., (2014). Improving Grid Networks Security by One-time Password Security Mechanism. International Journal of Advances in Computer Science and Technology, ISSN 2320 – 2602, Volume 3, No.2, February 2014.

[Khaled R. et Al., 2015] Khaled R., Zhu Y., Hongxin H. and Ahn G., (2015). AR-ABAC: A New Attribute Based Access Control Model Supporting Attribute-Rules for Cloud Computing, IEEE Conference on Collaboration and Internet Computing.

[Khan F. et Al., 2016] Khan F., Li H., and Zhang L., (2016). Owner Specified Excessive Access Control for Attribute Based Encryption, DOI 10.1109/ACCESS.2016.2632132, IEEE Access.

[Khedkar S.V. and Gawande A.D., 2014] Khedkar S.V. and Gawande A.D., (2014). Data Partitioning Technique to Improve Cloud Data Storage Security, Swapnil V.Khedkar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 3347-3350.

[Khider H. et Al., 2010] Khider H.; Osman T. and Sherkat N., (2010). Attribute-Based Authorization for Grid Computing, 2010 International Conference on Intelligent Systems, Modelling and Simulation, DOI: 10.1109/ISMS.2010.24.

[Kiranjot K. and Anjandeep K. R., 2014] Kiranjot K. and Anjandeep K. R. (2014).A Comparative Analysis: Grid, Cluster and Cloud Computing. International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 3, March 2014.

[Kónya B. ,2004] Kónya B. (2004). Advanced Resource Connector (ARC) – The Grid Middleware of the NorduGrid. In: Kranzlmüller D., Kacsuk P., Dongarra J. (eds) Recent Advances in Parallel Virtual Machine and Message Passing Interface. EuroPVM/MPI 2004. Lecture Notes in Computer Science, vol 3241. Springer, Berlin, Heidelberg.

[Kulkarni P. and Pathare S., 2014] Kulkarni P. and Pathare S., (2014). Performance Analysis of Parallel Algorithm over Sequential using OpenMP. IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. X, PP 58-62 (Mar-Apr. 2014).

Références

[Kumari A.K. et Al., 2011] Kumari A.K., Sadasivam S.G. , Senthil Prabha R. and Saranya G., (2011). Grid Based Security Framework for Online Trading, 2011 International Conference on Process Automation, Control and Computing.

[lebigdata, 2018] lebigdata, (2018). Amazon Cloud – Tout savoir sur Amazon Web Services. https://www.lebigdata.fr/amazon-cloud-amazon-web-services (Consulté en 2018).

[Li N. et Al., 2015] Li N., Dong Y., Che, T. and Wang C., (2015). Cross-Domain Authorization Management Model for MultiLevels Hybrid Cloud Computing. International Journal of Security and Its Applications Vol.9, No.12, pp.357-366.

[Liu X. et Al., 2013] Liu X., Xia Y., Jiang S., Xia F. and Wang Y., (2013). Hierarchical Attribute-based Access Control with Authentication for Outsourced Data in Cloud Computing, 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications.

[Liu X. et Al., 2012] Liu X., Yuan D., Zhang G., Li W., Cao D., He Q., Chen J. and Yang Y., (2012). Cloud Workflow System Architecture. In: The Design of Cloud Workflow Systems. SpringerBriefs in Computer Science. Springer, New York, NY.

[Luo W. and Bai G., 2011] Luo W. and Bai G., (2011). Ensuring The Data Integrity In Cloud Data Storage, 2011 IEEE International Conference on Cloud Computing and Intelligence Systems.

[Maji H. et Al., 2011] Maji H., Prabhakaran M. and Rosulek M., (2011). Attribute-based signatures. Proceedings of the 11th International Conference on Topics in Cryptology, pages 376–392, 2011.

[Mansour A. and Sadik M., 2015] Mansour A. and Sadik M., (2015). Essaid Sabir; Multi-factor Authentication based on Multimodal Biometrics (MFA-MB) for Cloud Computing; 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA).

[Martino, L.D. et Al., 2008] Martino, L.D., Ni Q., Lin, D. and Bertino E. (2008). Multi-domain and Privacy-aware Role Based Access Control in eHealth. Pervasive Computing Technologies for Healthcare, Pervasive Health. Second International Conference.

[medium, 2018] medium, 2018. An Annotated History of Google's Cloud Platform. https://medium.com/@retomeier/an-annotated-history-of-googles-cloud-platform-90b90f948920 (Consulté en 2018).

[Mell P. and Grance T., 2011] Mell P. and Grance T., (2011). The NIST Definition of Cloud Computing. National Institute of Standards and Technology Special Publication 800-145 7 pages (September 2011).

[Minoli D., 2005] Minoli D., (2005). A Networking Approach to Grid Computing. Published by John Wiley & Sons, Inc., Hoboken, New Jersey. Published simultaneously in Canada. ISBN 0-471-68756-1.

[Mohd Alif Hasmani A.G. et Al., 2012] Mohd Alif Hasmani A.G., , Badlishah R.A. and Naemah A.W., (2012). Grid Authentication Mechanisms Survey. The 2nd International

Malaysia-Ireland Joint Symposium on Engineering, Science and Business 2012 (IMiEJS2012).

[Mon E. and Naing T., 2011] Mon E. and Naing T., (2011). The privacy-aware access control system using attribute-and role-based access control in private cloud, proceedings of IEEE IC-BNMT.

[MotOrBAC, 2018] MotOrBAC. MotOrBAC :An OrBAC security policy editor. http://motorbac.sourceforge.net/index.php?page=home&lang=fr (Consulté en 2018).

[mturk, 2018] mturk, (2018). Amazon Mechanical Turk (MTurk). https://www.mturk.com/. Date d'accès: 2018.

[Nair N.K. and Navin K. S, 2015] Nair N.K. and Navin K. S, (2015). An efficient group authentication mechanism supporting key confidentiality, key freshness and key authentication in cloud computing, 2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), DOI: 10.1109/ICCPEIC.2015.7259477.

[Nandakumar V., 2014] Nandakumar V., (2014). A novel shared key for security in grid computing, 2014 International Conference on Smart Structures and Systems (ICSSS), DOI: 10.1109/ICSSS.2014.7006190.

[Namane S. et Al., 2017] Namane S., Kaiiali M., Ghoualmi N., (2017). Weighted Grid Authorization Graph (WGAG). 2017 Sixth International Conference on Communications and Networking (ComNet). DOI: 10.1109/COMNET.2017.8285589.

[Nimmy K. and Sethumadhavan M., 2014] Nimmy K. and Sethumadhavan M., (2014). "Novel mutual authentication protocol for cloud computing using secret sharing and steganography", Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT), feb, 2014, India.

[Nordugrid, 2018] Nordugrid, (2018). Nordugrid – The Nordic Testbed For Wide Area Computing And Data Handling. http://www.nordugrid.org/documents/nordugrid-final.pdf (Consulté en 2018).

[Noureddine R., 2010] NOUREDDINE R., (2010). Métaheuristiques Sur Grilles de Calcul. Thèse de Magister soutenue le 15 décembre à l'Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf - USTO - MB, 2010.

[OASIS, 2015] OASIS. eXtensible Access Control Markup Language (XACML) Version 1.0. https://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf (Consulté en 2015).

[OASIS, 2016] A brief introduction to XACML. https://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html (Consulté en 2016).

[OASIS, 2016][b] OASIS, (2016). SAML, https://www.oasis-open.org/standards#samlv1.0. (Consulté en 2016).

[OASIS, 2016][c] OASIS, (2016). OASIS, https://www.oasis-open.org/org (Consulté en 2016).

[OASIS, 2016][d] OASIS, (2016). XACML, https://www.oasis-open.org/standards#xacmlv1.0, (Consulté en 2016).

[OASIS, 2016][e] OASIS, (2016). OASIS Members Organize to Define Stateful Resources Using Web Services, https://www.oasis-open.org/news/pr/oasis-members-organize-to-define-stateful-resources-using-web-services. Date d'accès: 2016.

[OASIS, 2017] OASIS, (2017). OASIS Security Services (SAML) TC, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security (Consulté en 2017).

[OASIS, 2017][b] OASIS, (2017). eXtensible Access Control Markup Language (XACML) Version 1.0, https://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf (Consulté en 2017).

[Oracle, 2015] Oracle, (2015). Parsing an XML File Using SAX. https://docs.oracle. com/javase/tutorial/jaxp/ sax/parsing.html (Consulté en 2015).

[Orbac, 2016] Orbac. OrBAC: Organization based access control. http://orbac.org/?page_id=21 (Consulté en 2016).

[Orbac, 2017] Orbac. OrBAC API. http://orbac.org/?page_id=16 (Consulté en 2017).

[Ortiz A, 2009] Ortiz A., Contrôle de la concurrence dans les grilles informatiques : Application au projet ViSaGe. Thèse de doctorat soutenue le 17 décembre à l'Université Toulouse III - Paul Sabatier, 2009.

[Pan Y. et Al., 2008] Pan Y., Zhang Y. and Chiu K., (2008). Hybrid parallelism for XML SAX parsing. IEEE International Conference on Web Services, ICWS08, 23–26 September 2008, pp. 505–512.

[Pawar P. and Sheikh R., 2016] Pawar P. and Sheikh R., (2016). Implementation of Secure Authentication Scheme and Access Control in Cloud Computing; 2016 International Conference on ICT in Business Industry & Government (ICTBIG).

[Pourqasem J et Al., 2014] Pourqasem J., Karimi S. and Edalatpanah S.A., (2014). Comparison of Cloud and Grid Computing. American Journal of Software Engineering. 2014, 2(1), 8-12. DOI: 10.12691/ajse-2-1-2.

[Qiang W. and Konstantinov A., 2010] Qiang W. and Konstantinov A., (2010). The design and implementation of standards-based Grid single sign-on using federated identity, 2010 12th IEEE International Conference on High Performance Computing and Communications.

Références

[Rajesh I. and Sivakumar G., 2010] Rajesh I. and Sivakumar G., (2010). EGSI: TGKA Based Security Architecture for Group Communication in Grid, 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, DOI: 10.1109/CCGRID.2010.28.

[Razieh M. et Al., 2010] Razieh M., Ho S.B. and Nithiapidary M., (2010). An empirical study on secure communication for grid information service, 2010 International Conference on Computer Applications and Industrial Electronics Year: 2010, DOI: 10.1109/ICCAIE.2010.5771166.

[Resinfo, 2018] Resinfo, 2018. Et les grilles dans tout ça? https://resinfo.org/IMG/pdf/josy-grilles-Bruno-Bzeznik.pdf (Consulté en 2018).

[Revolvy, 2018] Revolvy, (2018). Graham-Denning model. https://www.revolvy.com/main/index.php?s=Graham-Denning%20model (Consulté en 2018).

[Rissanen E., 2017] Rissanen E. (2017). eXtensible Access Control Markup Language (XACML) Version 3.0. OASIS Standard. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html (Consulté en 2017).

[Rucha D. et Al., 2017] Rucha D., Shubham S. and Gaurav G.(2017). Time Domain Attribute Base Access Control for Cloud Based Content Sharing: A Cryptographic Approach. International Journal for Modern Trends in Science and Technology Volume: 03, Issue No: 01, January 2017 ISSN: 2455-3778.

[Sadashiv N. and Dilip Kumar S.M., 2011] Sadashiv N. and Dilip Kumar S.M., (2011). Cluster, Grid and Cloud Computing: A Detailed Comparison. The 6th International Conference on Computer Science & Education (ICCSE 2011) August 3-5, 2011. SuperStar Virgo, Singapore

[Sadegh D.N. and Rasool J., 2016] Sadegh D.N. and Rasool J., (2016). TIRIAC: A trust-driven risk-aware access control framework for Grid environments. Future Generation Computer Systems · February 2016, DOI: 10.1016/j.future.2015.03.003.

[salesforce, 2017] salesforce, (2017). Qu'est-ce que le « cloud » ? https://www.salesforce.com/fr/learning-centre/tech/cloudcomputing/ (Consulté en 2017).

[salesforce, 2018] salesforce, (2018). Qu'est-ce que Salesforce ? https://www.salesforce.com/fr/products/what-is-salesforce/ (Consulté en 2018).

[salesforce, 2018]^{b} salesforce, (2018). Notre histoire : aider les entreprises à grandir. https://www.salesforce.com/fr/company/about-us/ (Consulté en 2018).

[Sarvabhatla M. et Al., 2014] Sarvabhatla M., Giri M. and Vorugunti C.S., (2014). A Secure Mutual Authentication Protocol for Cloud Computing using Secret Sharing and Steganography. 2014 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM).

Références

[Singh A. and Chatterjee K., 2015] Singh A. and Chatterjee K., (2015). A secure multi-tier authentication scheme in cloud computing environment, 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015], DOI: 10.1109/ICCPCT.2015.7159276.

[Smarr L. and Catllet C.E., 1992] Smarr L. and Catllet C.E., (1992). Metacomputing, Magazine Communications of the ACM CACM Homepage archive Volume 35 Issue 6, June 1992 Pages 44-52 ACM New York, NY, USA.

[Sudalai Muthu T. et Al., 2010] Sudalai Muthu T., Vadivel R., Ramesh A. and VasanthG., (2010). A novel protocol for secure data storage in Data Grid environment, Trendz in Information Sciences & Computing(TISC2010) Year: 2010 Pages: 125 - 130, DOI: 10.1109/TISC.2010.5714622.

[Sulochana M. and Dubey O., 2015] Sulochana M. and Dubey O.,(2015). Preserving Data Confidentiality using Multi-Cloud Architecture, 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15).

[Sumtsova I. et Al., 2018] Sumtsova I., Shcheglov S.A. and Shendryk V.V. Bell–LaPadula model of computer security. https://core.ac.uk/download/pdf/14060453.pdf (Consulté en 2018).

[Sun L. et Al., 2013] Sun L., Wang H. and Betino E., (2013). Role based access control to outsourced data in cloud computing; Proceedings of the Twenty-Fourth Australasian Database Conference (ADC 2013), Adelaide, Australia.

[Sun L. et Al., 2012] Sun L., Wang R., Yong J. and Wu G., (2012). Semantic access control for cloud computing based on e-Healthcare, Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design.

[Talkhaby H.R. and Parsamehr R., 2016] Talkhaby H.R. and Parsamehr R., (2016). Cloud computing authentication using biometric-Kerberos scheme based on strong Diffi-Hellman-DSA key exchange, 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), DOI: 10.1109/ICCICCT.2016.7987926.

[TechTarget, 2018] TechTarget, (2018). Mandatory access control (MAC). http://searchsecurity.techtarget.com/definition/mandatory-access-control-MAC (Consulté en 2018).

[Thion R., 2007] Thion R., (2007). Access Control Models. Cyber Warfare and Cyber Terrorism, IGI global, 10.4018/978-1-59140-991-5.ch037.

[Tiezhu Z. and Shoubin D., 2010] Tiezhu Z. and Shoubin D., (2010). Trust-GSM: A Trust Aware Security Model for Multi-domain Grid, 2010 Fifth Annual ChinaGrid Conference Year: 2010 Pages: 43 - 47, DOI: 10.1109/ChinaGrid.2010.49.

Références

[Tu M. et Al., 2010] Tu M., Li P., Yen I.L., (2010). Secure Data Objects Replication in Data Grid; IEEE transactions on dependable and secure computing, vol. 7, no. 1, january-march.

[Tuecke et Al., 2003] Tuecke S., Foster I., Frey J., Graham S. , Kesselman C., Maquire T., Sandholm T., Snelling D., Vanderbilt F. P., (2003). Open Grid Services Infrastructure (OGSI). Global Grid Forum.

[Vachhani M.K. and Atkotiya K.H., 2012] Vachhani M.K. and Atkotiya K.H., (2012). Globus Toolkit 5 (GT5): Introduction of a tool to develop Grid Application and Middleware. International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 7, July 2012).

[Varadharajan V. et Al., 2015] Varadharajan V., Amid A. and Rai S., (2015). Policy Based Role Centric Attribute Based Access Control Model; 2015 Intl. Conference on Computing and Network Communications (CoCoNet'15), Dec. 16-19, 2015, Trivandrum, India.

[Velciu M.A. et Al., 2014] Velciu M.A., Patrascu A. and Patriciu V.V., (2014). Bio-cryptographic authentication in cloud storage sharing; 9th IEEE International Symposium on Applied Computational Intelligence and Informatics • May 15-17, 2014 Timişoara, Romania.

[Wang L et Al., 2009] Wang L., Jie W. and Chen J., (2009). Grid Computing: Infrastructure, Service, and Applications. CRC Press, 16 avr. 2009 - 528 pages.

[Wikipedia, 2017] Wikipedia, (2017). Parallel computing. https://en.wikipedia.org/wiki/Parallel_computing (Consulté en 2017).

[Wlcg, 2018] Wlcg, 2018. Welcome to the Worldwide LHC Computing Grid. http://wlcg.web.cern.ch/ (Consulté en 2018).

[WSO2 Inc, 2015][a] WSO2 Inc, (2015). Claim Management in WSO2 identity server version 5. https://docs.wso2.com/display/IS500/Claim+Management (Consulté en 2015).

[WSO2 Inc, 2015][b] WSO2 Inc, (2015). The WSO2 Identity Server Documentation. https://docs.wso2.com/display/IS510/WSO2+Identity+Server+Documentation (Consulté en 2015).

[Yuan, E. and Tong, J., 2005] Yuan, E. and Tong, J., (2005). Attributed Based Access Control (ABAC) for Web Service. The 2005 IEEE International conference on web service (ICWS‟05).

[Yue-qin F. and Yong-sheng Z., 2012] Yue-qin F. and Yong-sheng Z. (2012). Trusted Access Control Model Based on Role and Task in Cloud Computing, 7th International Conference on Information Technology in Medicine and Education .

[Zardari M.A., 2014] Zardari M.A., Jung L.T., Zakaria N., (2014). K-NN Classifier for Data Confidentiality in Cloud Computing, 2014 International Conference on Computer and Information Sciences (ICCOINS).

[Zhao T. and Shoubin D., 2010] Zhao T. and Shoubin D., (2010). A Trust Aware Grid Access Control Architecture Based on ABAC, 2010 IEEE Fifth International Conference on Networking, Architecture, and Storage, DOI: 10.1109/NAS.2010.18.

[Zhou L. et Al., 2013] Zhou L., Varadharajan V. and Hitchens M,(2013). Achieving Secure Role-Based Access Control on Encrypted Data in Cloud Storage. IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 8, NO. 12, DECEMBER 2013.

[Zhu T. et Al., 2011] Zhu T. , Liu W. and Song J.,(2011). An efficient Role Based Access Control System for Cloud Computing, 11th IEEE International Conference on Computer and Information Technology.

[Zhu X.J. et Al., 2010] Zhu X.J., Lv S.Q., Yu X.L. and Zuo G.P., (2010). Dynamic Authorization of Grid Based on Trust Mechanism, 2010 International Symposium on Intelligence Information Processing and Trusted Computing, DOI: 10.1109/IPTC.2010.113.

Annexes : Liste des publications

Annexe : liste des publications

[Namane S. and Ghoualmi N., 2018] Namane S. and Ghoualmi N., (2018). Parallel Access Control Model in Cross Domain Grid Computing Environment. International Journal of Embedded and Real-Time Communication Systems (IJERTCS) volume: 9 issue: 1. DOI: 10.4018/IJERTCS.2018010103.

[Namane S. et Al., 2017] Namane S., Kaiiali M., Ghoualmi N., (2017). Weighted Grid Authorization Graph (WGAG). 2017 Sixth International Conference on Communications and Networking (ComNet). DOI: 10.1109/COMNET.2017.8285589.

[Namane S. et Al., 2017] Namane S., Harbi N., Ghoualmi N., (2017). Une extension du standard XACML basée sur ARBAC pour contrôler l'accès à différents niveaux de données hébergées dans un environnement de cloud. The Conference on Advances on Decisional Systems (ASD 2018).

[Namane S. et Al., 2018] Namane S., Kaiiali M., Ghoualmi N., (2018). Action Weighted Grid Authorization Graph (Action-WGAG) (En cours d'examen).

[Namane S. and Ghoualmi N., 2018] Namane S. and Ghoualmi N., (2018). Grid and Cloud computing Security: State of the art. (En cours d'examen).