Conception et développement d'un DSL pour le modèle de contrôle d'accès HoBAC ; maisons intelligentes comme étude de cas

Mémoire présenté

dans le cadre du programme de maîtrise en informatique

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

PAR

© MAMADOU MOUSLIM DIALLO

Décembre 2024

Composition du jury :

Mohamed Tarik Moutacalli, président du jury, UQAR

Mehdi Adda, directeur de recherche, UQAR

Thierno Barry, PhD, membre externe, Census Labs, UAE

Fehmi Jaafar, PhD, membre interne, UQAC

Dépôt initial le 26 septembre 2024 Dépôt final le 10 décembre 2024

DEDICACE

Ce mémoire est le fruit d'un long travail, un long processus de réflexion, de concentration, de patience et de recherche dans un monde nouveau pour moi. Atteindre le bout du tunnel a été un ouf de soulagement. Intégrer le club des scientifiques est un travail fastidieux, seuls le focus et le courage permettent d'y arriver. Pour cela, je dédie ce travail à ma famille, à mes parents et spécialement à ma maman qui a su m'inculquer et me donner les armes nécessaires pour accomplir cette mission. Je pense également à mon père, à mes frères, à ma sœur qui sont pour moi une source de motivation. Dédicace à vous tous !

REMERCIEMENTS

Je ne saurai commencer la rédaction de ce mémoire de recherche sans remercier les personnes qui, sans elles, ce projet n'aurait jamais abouti. Mes pensées vont à mon professeur et directeur de recherche Monsieur **Mehdi Adda** qui sans cesse n'a cessé de m'orienter dans ce travail en répondant à mes tonnes de questions. Il est à remercier pour sa patience, le temps qu'il m'a accordé ainsi que ses observations qui m'ont conduit dans la bonne direction. Grâce à lui, j'ai beaucoup appris. Je tiens à exprimer ma profonde gratitude aux membres du jury. Leur clairvoyance et leur expertise ont grandement contribué à enrichir mon travail.

Mes remerciements vont chaudement à l'endroit de mes parents qui m'ont soutenu dès le début et qui n'ont jamais cessé de le faire. Sans eux, je ne serais pas là où je suis actuellement. Mon esprit vogue vers la personne la plus importante au monde pour moi, ma **Maman** chérie qui m'a toujours accompagné par ses bénédictions et sa confiance. Sa fierté est mon cheval de bataille. Ce qu'elle a fait pour moi est incommensurable. Merci Maman !

Je remercie également l'université **UQAR** pour l'appui et l'accompagnement qu'elle offre aux étudiants que nous sommes, ainsi que pour les nombreuses opportunités qu'elle nous donne. Sans oublier mes collègues et amis à l'université et ailleurs avec lesquels j'ai pu échanger et partager des moments enrichissants.

RÉSUMÉ

Les systèmes IoT (Internet Of Things), en raison de leur nature évolutive, hétérogène et dynamique, présentent des défis de sécurité significatifs (Abbassi & Benlahmer, 2021). L'absence d'outils et de langages appropriés pour la spécification, l'évaluation et l'exécution des règles de sécurité exacerbe ces défis. Cette situation est particulièrement problématique dans des environnements contraignants tels que les maisons intelligentes. Dans ce contexte, le nouveau modèle de contrôle d'accès dénommé HoBAC (Higher-Order Attribute-Based Access Control) est proposé (Aliane & Adda, 2019). Ce modèle, applicable aux systèmes IoT et non IoT est une généralisation du modèle de contrôle d'accès basé sur les attributs ABAC (Attribute-Based Access Control). HoBAC se distingue par sa flexibilité et sa capacité de s'adapter à des environnements variés, utilisant des attributs pour spécifier les règles de contrôle d'accès. En combinant une hiérarchie d'entités (sujets, objets, contextes, actions) avec des attributs, ce modèle permet d'intégrer des politiques de sécurité adaptées aux besoins spécifiques des environnements IoT, y compris les maisons intelligentes. Notre recherche se focalise sur la conception et développement d'un langage spécifique au domaine (DSL) pour pallier le manque d'outils nécessaires à l'élaboration des politiques de contrôle d'accès basées sur le modèle HoBAC dans l'IoT. Conçu avec MPS (Meta Programming System), notre modèle HoBACDSL vise à abstraire les complexités de HoBAC, rendant ses concepts plus accessibles et simple à utiliser à travers un environnement de développement intégré complet. Nous illustrons comment HoBACDSL est utilisé pour spécifier, tester, valider et générer des politiques de contrôle d'accès compatibles aux spécifications XACML (Extensible Access Control Markup Language).

Mots clés : [Contrôle d'accès, Internet des Objets (IdO), Sécurité, Langage spécifique au domaine (DSL), Langage de balisage extensible pour le contrôle d'accès (XACML), Contrôle d'accès basé sur les attributs d'ordre supérieur (HoBAC), Système de Métaprogrammation (MPS)]

ABSTRACT

IoT (Internet of Things) systems, due to their scalable, heterogeneous and dynamic nature, present significant security challenges (Abbassi & Benlahmer, 2021). The lack of appropriate tools and languages for specifying, evaluating and enforcing security rules exacerbates these challenges. This is particularly problematic in constrained environments such as smart homes. In this context, a new access control model called HoBAC (Higher-Order Attribute-Based Access Control) is proposed (Aliane & Adda, 2019). This model, applicable to both IoT and non-IoT systems, is a generalization of the Attribute-Based Access Control (ABAC) access control model. HoBAC stands out for its flexibility and ability to adapt to a variety of environments, using attributes to specify access control rules. By combining a hierarchy of entities (subjects, objects, contexts, actions) with attributes, the HoBAC model enables the integration of security policies tailored to the specific needs of IoT environments, including smart homes. Our research focuses on the design and development of a domain-specific language (DSL) to overcome the lack of tools needed to develop access control policies based on the HoBAC model in the IoT. Designed with Meta Programming System (MPS), HoBACDSL aims to abstract HoBAC's complexities, making its concepts more accessible and easier to use through a complete integrated development environment. We illustrate how HoBACDSL is used to specify, test, validate and generate access control policies compatible with XACML (Extensible Access Control Markup Language) specifications.

Keywords: [Access Control, IoT, Security, HoBAC, MPS, XACML]

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES

AC	**Access Control**
ABAC	**Attribute-Based Access Control**
ACL	**Access Control List**
ADQL	**Access Definition and Query Language**
ALFA	**Abbreviated Language for Authorization**
AOS	**Asynchronous Object Substitution**
CBAC	**Context Based Access Control**

DSL	**Domain Specific Language**
DAC	**Discretionary Access Control**
DIC	**Disponibilité, Intégrité, Confidentiality**
Geo-RBAC.	**Geolocation Role-Based Access Control**
HoBAC	**Higher-Order Attribute-Based Access Control**
IoT	**Internet Of Things**
IdO	**Internet des Objets**
JSON	**JavaScript Object Notation**
LBAC	**Location-Based Access Control**
MPS	**Meta Programming System**
MAC	**Mandatory Access Control**
NAC	**Network Access Control**
NGAC	**Next Generation Access Control**
NIST	**National Institute of standard and Technology**
OM-AM	**Objectifs, Modèles, Architecture et Mécanismes**
OASIS	**Organization for the Advancement of Structured Information Standards**
OSI	**Open Systems Interconnection**
PAP	**Policy Administration Point**
PEP	**Policy Enforcement Point**
PDP	**Policy Decision Point**

PIP	**Policy Information Point**
PRP	**Policy Retrieval Point**
PML	**PERM Modeling Language**
PBAC	**Proximity-Based Access Control**
RBAC	**Role-Based Access Control**
ReBAC	**Relationship-Based Access Control**
TCP	**Transmission Control Protocol**
TRBAC	**Temporal Role-Based Access Control**
UIT	**Union International des Télécommunications**
UTRBAC	**Ubiquitous Role-Based Access Control**
UDP	**User Datagram Protocol**
XACML	**EXtensible Access Control Markup Language**
XML	**EXtensible Markup Language**

INTRODUCTION GÉNÉRALE

1. INTRODUCTION

1.1 Contexte général

Nous vivons à une époque où la technologie, en particulier l'internet des objets exerce une influence prépondérante sur notre vie quotidienne. De l'e-santé aux dispositifs de surveillance en passant par les appareils ménagers connectés tels que les réfrigérateurs et les poubelles intelligentes, l'IoT s'est largement intégré dans divers aspects de notre quotidien (Kumar et al., 2019). On estime que le marché de l'IoT pourrait atteindre 125 milliards en 2030 (Tanczer et al., 2018). Dans ce paysage technologique évolutive, le développement des maisons intelligentes se distingue comme l'un des progrès les plus significatifs. Elles représentent un ensemble de dispositifs interconnectés qui offrent divers services, y compris la sécurité, la surveillance à domicile pour les personnes âgées (Chan et al., 2005), et l'économie d'énergie. Bien que leur utilité soit indéniable, elles soulèvent des défis de sécurité qui pourraient impacter leur développement. Des études ont mis en lumière ces défis en abordant les limites, les exigences et les potentielles solutions (Mohanty et al., 2021). Toutefois, la protection des données personnelles et la prévention des attaques demeurent des préoccupations essentielles pour les consommateurs désireux de tirer parti des avantages de l'IoT (Emmanuel Bertin, 2019). L'application efficace de politiques de contrôle d'accès pour protéger ces systèmes constitue un défi de taille (Sicari et al., 2015). Ce défi englobe notamment la définition précise de règles adaptées à des scénarios variés ainsi que leur mise en œuvre dans des contextes hétérogènes et complexes. Pour relever ces enjeux, les langages spécifiques au domaine émergent comme des outils prometteurs. Un DSL peut simplifier la création et la gestion des politiques de contrôle d'accès dans les maisons intelligentes,

améliorant ainsi la sécurité globale. Au cours de la dernière décennie, plusieurs DSL ont été développés par les chercheurs. Le contrôle d'accès PML - PERM Modeling Language - et(Luo et al., 2019) le métamodèle de contrôle d'accès ADQL (Access Definition and Query Language) en sont des exemples (Andreas Sonnenbichler, 2013). Ces langages offrent une approche permettant de décrire les politiques de sécurité avec un niveau adapté aux exigences du domaine, facilitant la bonne communication entre les experts du domaine et les développeurs, mais aussi la vérification, la validation et la génération du code. Notre choix porté sur la sécurité des environnement IoT a un impact direct sur la vie quotidiens des citoyens (villes, maisons intelligentes, gadgets technologiques, etc.). L'un des défis les plus complexes est le développement de logiciels pour les systèmes IoT (Erazo-Garzón et al., 2022) puisque les outils traditionnels de génie logiciel ne répondent pas aux exigences des environnements IoT. L'expansion de l'IoT, loin d'être achevée compte tenu du nombre croissant d'objets connectés et les projections suggérées à l'horizon 2025, où le nombre d'appareils IoT pouvant atteindre 11 milliards souligne(Nagajayanthi, 2022) l'urgence de cette préoccupation. Face à cette expansion massive, il devient nécessaire d'implémenter des mécanismes de contrôle d'accès qui permettent de protéger ces systèmes interconnectés. Ces stratégies de contrôle d'accès, fondamentales en sécurité informatique, s'appliquent à des entités pouvant être physiques (bâtiments, parkings, véhicules, maisons) ou logiques telles que des services et des données. Dans l'environnement IoT, l'accès aux ressources est géré par des mécanismes de contrôle d'accès assurant que seuls les utilisateurs autorisés peuvent accéder à des services spécifiques. Ce processus de contrôle débute par l'identification et l'authentification des utilisateurs afin de déterminer qui est autorisé à réaliser quelles actions. Dans le cadre de notre mémoire, nous nous concentrons sur la conception d'un DSL qui intègre des mécanismes de contrôle d'accès spécifiques aux maisons intelligentes, où la protection des ressources constitue notre principale préoccupation. Le contrôle d'accès est un ensemble de mécanismes et de règles déterminant qui a le droit d'accéder à quelles ressources et quelles actions peuvent être entreprises une fois l'accès autorisé. Les architectures de contrôle d'accès bien que variées et parfois complexes, nécessitent une gestion efficace. Parmi les modèles les plus populaires, ABAC, se différencie par l'utilisation

des attributs des objets, des sujets et de l'environnement pour définir les politiques d'accès (Vijayalakshmi & Jayalakshmi, 2022). Toutefois, la complexité croissante des scénarios IoT rend la gestion des politiques d'accès plus difficile avec ce modèle (Zhang et al., 2021). Pour faire face à ce défi, des approches telles que NGAC (Next Generation Access Control) et des langages de spécification de politiques comme XACML sont développées. NGAC étend ABAC en utilisant une représentation graphique des données d'accès pour modéliser des politiques d'accès complexes et dynamiques. XACML, qui est notre choix, est un langage standardisé qui permet d'exprimer des politiques d'accès basées non seulement sur les attributs mais également sur d'autres types de politiques, offrant ainsi des syntaxes et des sémantiques pour décrire des règles d'accès. L'introduction du nouveau modèle HoBAC, une généralisation d'ABAC surmonte certaines limitations des modèles de contrôle d'accès existants (Adda & Aliane, 2020). Il permet de mettre en œuvre des politiques de contrôle d'accès flexibles, adaptées aux environnements IoT et non IoT. En introduisant de nouvelles primitives pour gérer la propagation des changements d'attributs et permettre l'évaluation dynamique de ceux-ci, HoBAC offre une approche innovante et plus flexible du contrôle d'accès. L'usage des hiérarchies d'entités (objets, sujets et contextes) et des opérations d'agrégation sur les attributs permettent une gestion plus efficace des autorisations d'accès. L'implémentation de ces modèles de contrôle d'accès peut nécessiter l'utilisation des langages de spécification de politiques d'accès offrant des syntaxes et des sémantiques pour décrire des règles d'accès. L'architecture XACML implique une variété de composants incluant des points de décision, des points d'application, des points de gestion des politiques et des points de gestion des attributs. Lorsqu'un utilisateur souhaite accéder à une ressource à travers un objet connecté, sa demande est soumise à une analyse approfondie incluant l'évaluation des règles et des politiques de sécurité, afin d'authentifier le demandeur. Cette analyse vise à vérifier les privilèges et les droits d'accès requis pour accéder à la ressource. Nous entreprendrons une étude approfondie des contrôles d'accès dans l'état de l'art, avec un penchant pour le modèle HoBAC, considérant qu'il répond mieux aux besoins et aux contraintes de l'IoT, en particulier pour les ressources d'une maison intelligente.

1.2 Revue de la littérature

1.2.1 La sécurité dans l'internet des objets, défis et perspectives

L'internet des objets ou Ido, englobe divers domaines d'application tels que la domotique, la santé, l'industrie, l'agriculture où des dispositifs connectés collectent, transmettent et traitent des données à travers internet. Ce concept, introduit en 1999 par Kevin (Ashton, 2009) a donné lieu à de multiples définitions. L'IoT est un réseau mondial d'objets interconnectés adressables de manière unique sur la base de protocoles de communication standard (Gubbi et al., 2013). Nous pouvons définir l'Internet des objets comme l'intégration de divers dispositifs en vue d'échanger des données, allant au-delà des machines traditionnelles, grâce à l'utilisation d'Internet. En référence à l'automatisation et l'intelligence artificielle, l'IoT est un processus qui autorise des objets, qu'ils soient identiques ou non, à interagir de manière autonome via Internet. L'objectif principal de l'IoT est d'accroître la productivité, d'économiser les ressources et d'optimiser les processus. La capacité de surveiller à distance ou d'automatiser les tâches domestiques représente un avantage non négligeable de l'IoT. L'ère actuelle, orientée vers l'automatisation, voit l'IoT jouer un rôle prépondérant, tandis que l'émergence de l'intelligence artificielle accélère ce processus en améliorant la précision, la rapidité et l'efficacité. Des exemples concrets illustrent comment l'IoT est utilisé pour améliorer la sécurité, surveiller les équipements distants, automatiser les processus et réduire l'impact environnemental notamment par les organisations et les municipalités. Par exemple, la surveillance du niveau des réservoirs dans les applications industrielles et agricoles peut améliorer l'efficacité et la sécurité (Locke, 2022). Cependant, si ces systèmes ne sont pas sécurisés, ils peuvent être vulnérables aux cyberattaques, entraînant des perturbations dans la distribution de l'eau ou des produits chimiques. De même, la surveillance et la détection d'obstacles sur les voies ferrées représentent des avantages significatifs (Jaber et al., 2022). Dans le domaine médical, l'IoT permet le suivi des patients et l'automatisation des processus médicaux, contribuant ainsi à une meilleure gestion des soins de santé (Déry, 2021). La montée rapide de l'IoT a élargi la surface

d’attaque et généré des préoccupations en matière de sécurité. On se souvient de Stuxnet, considéré comme une attaque IoT qui a endommagé physiquement des centrifugeuses iraniennes en détournant les instructions envoyées par les automates programmables (Trautman & Ormerod, 2017). L'une des principales inquiétudes est liée à l'exploitation des données des utilisateurs. La sécurité est une préoccupation majeure pour les utilisateurs lors de l’adoption de l’Internet des objets, avec une crainte accrue des cyberattaques qui pourraient avoir des conséquences physiques tangibles en raison de l’évolution des objets connectés (Abbassi & Benlahmer, 2021). Cette étude met en évidence l'ampleur de la croissance de l'Internet des objets, révélant des implications majeures pour la société et l'industrie. Des efforts de recherche approfondis sont indispensables pour aborder des problématiques telles que la sécurité, l'identification des dispositifs, et l'optimisation des réseaux dans le contexte des maisons intelligentes (Cvitić et al., 2021). L’expansion rapide de l’IoT renforce l'urgence de résoudre ces défis de sécurité pour garantir un environnement IoT sûr et fiable.

1.2.2 Défi de sécurité dans les maisons intelligentes

De nombreux secteurs sont révolutionnés par l’IoT, y compris celui de la domotique. Chaque domaine présente des problématiques de sécurité uniques en raison des caractéristiques et des types de dispositifs impliqués. La sécurité dans l'IoT ne peut pas être abordée de manière uniforme compte tenu des risques et les enjeux différents d'un secteur à l'autre. Les travaux scientifiques qui s’y rapportent, notamment le développement des maisons intelligentes démontrent plusieurs technologies associées (Alam et al., 2012). Avec la croissance démographique mondiale projetée des Nations Unies avec une estimation de 8 milliards de personnes en 2022, 8,5 milliards en 2030 et 9,7 milliards en 2050 (United Nations, 2022), combiné à l’accroissement considérable du nombre d'appareils IoT, passant potentiellement de 15,1 milliards en 2020 à plus de 29 milliards en 2030 (Transforma Insights, 2023), les maisons intelligentes revêtent une importance croissante fournissant divers services de confort, tels que la sécurité, la surveillance à domicile pour les personnes âgées (Chan et al., 2005). La domotique, un domaine clé de l'IoT, utilise des dispositifs

connectés à Internet pour améliorer le confort de vie des individus. Elle manifeste également comme un système médical intelligent, capable de détecter proactivement les symptômes sans nécessiter d'hospitalisation (Mukhopadhyay & Jayasundera, 2017). Les avancées technologiques, notamment les capteurs IoT, permettent la collecte et l'analyse continues des données et l'aide des personnes à mobilité réduite pour surmonter l'isolement social auquel elles sont souvent confrontées (Chan et al., 2008). Cependant, l'application des mécanismes de sécurité et de contrôle d'accès dans des environnements contraints que les maisons intelligentes présentent des défis complexes. Une analyse des systèmes de domotique révèle plusieurs failles de sécurité existantes, soulevant des inquiétudes quant à la sécurité des occupants (Jose & Malekian, 2015). Les défis de sécurité exposés dans la structure en couches de l'IoT, en particulier dans le contexte des maisons intelligentes sont également préoccupants (Touqeer et al., 2021), tout comme les risques en matière de sécurité si une maison est laissée sans surveillance (Chitnis et al., 2016) en sont des exemples. Notre recherche se concentre sur le développement de HoBACDSL pour le modèle HoBAC, contribuant à l'expression des politiques de contrôle d'accès basées sur les entités des maisons intelligentes.

1.2.3 Le contrôle d'accès et leur évolution

Le contrôle d'accès a connu une évolution technologique importante dans les années 1960. L'un des premiers modèles apparaît vers 1973 dans le domaine militaire(Bell & LaPadula, 1989), puis avec la protection des accès avec MULTICS (Saltzer & Schroeder, 1975). Cette évolution a conduit à l'émergence des modèles classiques de contrôle d'accès, notamment le DAC (Discretionary Access Control), MAC (Mandatory Access Control) et RBAC (Role-Based Access Control) pour des applications en sécurité informatique (Ubale Swapnaja et al., 2014). Le modèle ABAC a unifié ces trois modèles classiques (Hu et al., 2015) en offrant une approche plus flexible (Jin et al., 2012). Chaque modèle présente des avantages et des inconvénients adaptés à différents contextes et besoins. Le contrôle d'accès, en tant que processus déterminant *"qui fait quoi à quoi"* en fonction d'une politique (Kizza, 2024), est une composante fondamentale de la sécurité informatique avec des systèmes conçus pour

n'autoriser l'accès qu'aux individus autorisés, en utilisant des technologies telles que la biométrie ou les cartes à puce. Cette approche permet aux responsables d'entreprise de gérer efficacement les autorisations d'accès et de suivre les mouvements du personnel en temps réel. L'accès à des systèmes d'information nécessite l'usage d'un ensemble de moyens techniques, humains, physiques et juridiques pour maintenir l'intégrité, la disponibilité et la confidentialité des données. Trois concepts clés sont essentiels : la Disponibilité, l'Intégrité et la Confidentialité, regroupés sous l'acronyme DIC (Yee & Zolkipli, 2021). Dans les années 80, de solutions technologiques ont été mises en place pour gérer l'accès aux bâtiments et aux zones sensibles. Le modèle DAC reste un modèle décentralisé, permettant à l'utilisateur, propriétaire d'un objet de donner accès (écriture, lecture, etc.) à d'autres utilisateurs. Cependant, le DAC présente plusieurs limitations. L'absence de centralisation peut nuire à la gestion des permissions. De plus, le DAC a du mal à garantir une stricte confidentialité et intégrité des données (Ausanka-Crues, 2001). Le modèle MAC (Mandatory Access Control) en revanche est centralisé et offre une sécurité accrue. Les autorisations d'accès sont gérées par le système lui-même à un niveau élevé, nécessitant une bonne configuration et une gestion régulière des autorisations par l'administrateur. Cependant, le MAC peut être rigide et complexe à gérer. En 1992, RBAC (Role-Based Access Control) a été normalisé par David Ferraiolo et Rick Kuhn (FERRAIOLO D et KUHN, 1992). Largement adopté dans divers secteurs grâce à son approche flexible basée sur les rôles, l'impact économique de RBAC a permis aux industries d'économiser plusieurs milliards de dollars (O'Connor & Loomis, 2010). Il implémente le DAC ou le MAC dans la restriction des accès. Cependant, RBAC peut être complexe à mettre en œuvre et à gérer efficacement. Le manque de précision et l'augmentation des rôles peut rendre la gestion plus complexe (Fatima et al., 2016). De plus, le RBAC peut conduire à une explosion des rôles, où le nombre de rôles devient ingérable. Il a été mis à jour en 2004 et 2012 par ANSI/INCITS 359-2004 pour renforcer la sécurité d'accès, entre autres en incorporant des fonctionnalités de contrôle d'accès basé sur les attributs (Kuhn et al., 2010). En raison de la complexité d'attribution des rôles aux utilisateurs dans le modèle RBAC, plusieurs autres modèles ont vu le jour. Parmi eux, le modèle TRBAC (Temporal Role-Based Access Control) qui introduit une dépendance temporelle entre les

actions effectuées par l'utilisateur (Bertino et al., 2000). En 2005, les travaux de Bertino ont abouti à un autre modèle, le GEO-RBAC qui définit la géolocalisation de l'utilisateur en tenant compte de sa position physique ou logique (Damiani et al., 2007), outre LBAC – Location-Based Access Control - (Van Cleeff et al., 2010). Les auteurs introduisent un concept de schéma de rôle pour améliorer la flexibilité et la réutilisabilité. En 2006, le modèle UTRBAC (Ubiquitous Role-Based Access Control) a été proposé, intégrant les informations de localisation pour permettre un contrôle minimal du privilège de l'utilisateur (Chae et al., 2006). Il s'adapte dynamiquement à plusieurs environnements. Par ailleurs, le contrôle d'accès ABAC, normalisé par NIST - National Institute of standard and Technology apporte une approche flexible grâce à la richesse de ses attributs. Il propose une méthodologie « de contrôle d'accès logique où l'autorisation d'effectuer un ensemble d'opérations est déterminée en évaluant les attributs associés au sujet, à l'objet, aux opérations demandées et, dans certains cas, aux conditions d'environnement par rapport à la politique, aux règles ou aux relations qui décrivent les opérations autorisées »(Hu et al., 2013). Cependant, le modèle ABAC affiche une dégradation des performances lors de l'évaluation des politiques et des attributs pour chaque demande d'accès. De plus, il nécessite une conception préalable pour être correctement mis en place. Outre le modèle ABAC, le Proximity-Based Access Control (PBAC) adopte une approche particulière de contrôle d'accès avancée, basé sur la proximité, dynamique et contextuel (Lang & Schreiner, 2015). Il gère ses stratégies d'accès en tenant compte d'un ensemble de politiques prédéfinies, notamment des facteurs environnementaux et contextuels. Le modèle CBAC (Context Based Access Control) est également une autre spécialisation d'ABAC (Systems, 2020), utilisé dans la sécurisation des réseaux pour filtrer les paquets TCP (Transmission Control Protocol) et UDP (User Datagram Protocol) lors des échanges en se référant à la couche application du Modèle OSI (Open Systems Interconnection). Une autre approche (Al-Wahah & Farkas, 2019) offre cette capacité de délégation de contexte de manière plus dynamique et adaptative, sans modifier la politique sous-jacente. Par ailleurs, le XACML et le NGAC sont deux normes qui apportent également deux approches différentes pour gérer le contrôle d'accès dans les systèmes informatiques (Ferraiolo et al., 2016). Bien que ces deux normes partagent des objectifs similaires, elles

diffèrent considérablement dans la manière dont elles spécifient et gèrent les politiques de contrôle d'accès et les attributs, ainsi que dans la manière dont elles calculent et appliquent les décisions. Dans ce contexte, ALFA (Abbreviated Language for Authorization) offre une alternative intéressante pour les développeurs cherchant à réduire la complexité inhérente aux politiques d'accès. Depuis son adoption par l'OASIS XACML Technical Committee en 2014, ALFA a été standardisé en tant qu'outil open source, favorisant son utilisation dans de contextes variés. C'est un DSL orienté gestion des accès et des autorisations. Il est conçu pour simplifier la définition et la gestion des politiques d'accès, tout en s'appuyant sur les principes structurels de XACML (PolicySet, Policy, Rule). Il se distingue par une formulation plus claire et concise des politiques d'autorisation en prenant en charge divers types de données. En outre, ALFA supporte plusieurs modèles de contrôle d'accès, notamment le contrôle d'accès ABAC permettant de manipuler et d'évaluer des attributs complexes au sein des règles, le contrôle d'accès RBAC, en définissant des rôles et en attribuant des permissions basées sur ces rôles, le contrôle d'accès ReBAC, un modèle basé sur les relations. ALFA gère également les requêtes avec des algorithmes de combinaison (Brossard, 2023). De plus, les politiques définies avec ALFA peuvent être traduites de manière transparente en XACML, garantissant ainsi une compatibilité totale avec les systèmes et outils existants basés sur XACML. Cette adaptabilité en fait un outil flexible, capable de s'adapter à des contextes hétérogènes où les ressources et les exigences en matière de sécurité évoluent rapidement. Intégrable dans des pipelines CI/CD et des systèmes de gestion de versions comme GitHub, ALFA favorise la collaboration et l'automatisation du déploiement des politiques d'accès dans des environnements complexes (Giambiagi et al., 2015). L'intégration d'ALFA dans HoBAC permettrait de tirer parti de la notation légère et claire d'ALFA pour définir les relations complexes entre entités (objets, sujets, contextes) propres à HoBAC.

1.2.3.1 XACML (EXTENSIBLE ACCESS CONTROL MARKUP LANGUAGE)

La mise en œuvre de politiques de contrôle d'accès flexibles et distribuées constitue l'un des principaux objectifs sécuritaires de l'IoT. Pour répondre à ce besoin, il est nécessaire de disposer d'un langage de politique d'accès approprié, offrant une approche flexible et distribuée adaptée aux différents scénarios rencontrés dans les systèmes IoT (Kalaria et al., 2024). XACML (Standard, 2013) se positionne comme une solution répondant à ces exigences. Examiné par de nombreux experts et utilisateurs, il présente de nombreux avantages par rapport à d'autres langages de contrôle d'accès (Atlam et al., 2018). Il offre une compatibilité avec la plupart des modèles de contrôle d'accès tels que ACL (Access Control List), RBAC et ABAC (Parducci et al., 2010). XACML est une norme de OASIS (Organization for the Advancement of Structured Information Standards) qui a vu le jour en 2003 avec sa première version. Au fil des années, XACML a évolué pour donner naissance à plusieurs autres versions dont la 2.0 ratifiée par OASIS (Tim, 2005) et la version 3.0 Plus Errata 01, par (Standard, 2013). Ces avancées, notamment le JSON Profile of XACML 3.0 Version 1.0 édité par (Brossard, 2014) ont contribué à l'avancement de celui-ci et à son utilisation dans le contrôle d'accès basé sur les attributs. Ce langage, basé sur XML (Riad & Cheng, 2021) permet de gérer les autorisations d'accès dans les environnements complexes tels que l'IoT. Dans le contexte spécifique des environnements IoT, XACML peut être employé pour gérer les autorisations d'accès aux diverses ressources IoT. En effet, les environnements IoT se caractérisent souvent par leur hétérogénéité, leur dynamisme et leur complexité, ce qui rend la définition et l'application de politiques de contrôle d'accès particulièrement délicates. Les règles de contrôle d'accès sont définies à l'aide de fichiers XML (De la Rosa Algarín et al., 2014) dans une approche modulaire permettant aux développeurs de personnaliser les politiques en fonction des besoins spécifiques de leur système. L'architecture du modèle XACML (Pereira et al., 2017) repose sur plusieurs concepts clés. Un exemple de requête émise par l'un de ces composants, appelé Point d'Application de Politique (PAP), illustre brièvement les fondements de notre étude sur la

manière dont XACML exprime les politiques d'accès, en particulier dans le domaine de l'IoT et des maisons intelligentes.

- Le premier concept est le PAP (Policy Administration Point), un composant de gestion des règles et des politiques de contrôle d'accès. Il permet la création, la modification et la suppression des règles d'accès pour divers services tels que les bases de données, les applications ou les serveurs ;

- Le PEP (Policy Enfoncement Point) représente le point d'application de la politique, garantissant la protection des données applicatives. Il intercepte les demandes commerciales et les convertit en demandes d'autorisation avant de les envoyer au PDP;

- Le PDP (Policy Décision Point) est responsable des décisions d'autorisation ou de refus après avoir évalué les données par rapport aux sujets, aux objets ou à l'environnement. Le PDP peut solliciter le PIP (Point d'Information des Politiques) pour récupérer des informations manquantes lors de l'évaluation, agissant ainsi comme un serveur de décision;

- Le PIP (Policy Information Point) relie d'autres éléments au PDP, notamment les sources d'informations externes grâce à la méthode AOS, assurant la collecte des informations liées à la demande ;

- Le PRP est chargé de stocker les politiques dans une base de données.

- Les éléments de politiques XACML comprennent plusieurs composants fondamentaux utilisés par le PDP pour prendre des décisions. Le premier élément de base est <Policy> qui renforce la politique <Rule>. Voici comment le langage XACML est structuré :

1. <Policyset> contient le <Policy> et également inclure un autre <Policyset>. Il assure la combinaison des résultats pour une évaluation.

2. <Rule> est un élément de base de la politique qui peut se situer au point d'administration PAP. Il exprime des conditions contenant des fonctions booléennes et d'attributs. Les fonctions peuvent être imbriquées.
3. <Target> représente la cible à laquelle un élément <Policy> peut être appliqué. Les éléments Policyset, Policy et Rule contiennent l'élément Target pour spécifier les demandes.
4. <Policy> est un élément essentiel de la politique utilisé par le PDP pour prendre une décision d'autorisation.

Dans l'évaluation des politiques XACML, plusieurs algorithmes de combinaison sont utilisés pour calculer les décisions des politiques. Ces algorithmes déterminent la manière dont les résultats des règles sont combinés pour produire une décision. Lorsqu'une requête est émise, plusieurs types de réponses peuvent être générés : non applicable, autorisation (permit), refus (deny) ou indéterminé.

- Permit-Overrides : Dans cet algorithme, lors de l'évaluation, si une règle attribue une autorisation (permit), cette décision prévaut sur toute autre décision de refus ou d'indétermination ;
- Deny-Overrides : Contrairement à l'algorithme précédent, ici, si une règle attribue un refus (deny) lors de l'évaluation, cette décision prévaut sur toute autre décision d'autorisation ou d'indétermination (Ait El Hadj et al., 2018);
- First-Applicable : Cet algorithme exige que les règles soient évaluées dans un ordre spécifique, en commençant par la première règle. Dès qu'une règle est applicable, son effet est immédiat et aucune autre règle n'est évaluée.

XACML, bien qu'en constante évolution, présente certaines limitations. Il n'est pas compatible qu'avec les systèmes d'exploitation traitant du XML, principalement utilisé sur la plateforme Java. Les tentatives d'intégration avec des systèmes tels qu'Active Directory de Microsoft et le .NET sont en cours mais rencontrent des difficultés (Klenk et al., 2009). L'écriture de code en XML peut générer des fichiers volumineux (Abbassi & Benlahmer, 2021). La diversité des profils et des versions de XACML, peut rendre la gestion des politiques complexe. Cependant, malgré ces défis, XACML est une solution efficace pour la

gestion des politiques de contrôle d'accès. Grâce à ses règles et politiques de contrôle d'accès, XACML facilite la communication et l'interaction entre divers services, y compris les dispositifs IoT(Ashutosh et al., 2023). Cela justifie notre choix d'utiliser XACML pour générer notre code, ce qui permettrait une gestion efficace et sécurisée des accès dans l'environnement de la maison intelligente.

1.2.3.2 HOBAC, vers un contrôle d'accès dans les environnements IoT

Le modèle HoBAC se positionne comme une solution réelle dans le domaine de la sécurité en se distinguant par sa flexibilité et sa capacité à s'adapter à une variété d'environnements qu'ils soient liés ou non à l'IoT (Adda & Aliane, 2020). Fondé sur des hiérarchies d'entités telles que les sujets, les objets et les contextes, HoBAC repose sur des opérations d'agrégation des attributs pour construire ses politiques de contrôle d'accès. Cette approche constitue une évolution du modèle ABAC qui évalue les attributs plutôt que les rôles pour déterminer l'accès. L'introduction de HoBAC apporte une avancée dans la conception des politiques de contrôle d'accès en offrant la possibilité de mettre en place des politiques de contrôle d'accès flexibles, adaptées aussi bien aux systèmes IoT qu'aux systèmes traditionnels. Cette souplesse est rendue possible par l'utilisation des hiérarchies d'entités construites grâce à une composition fine et des opérations d'agrégation sur les attributs. Cette approche se révèle particulièrement pertinente dans le contexte de l'IoT, où la dynamique de l'environnement et la diversité des dispositifs représentent des défis majeurs en matière de sécurité (Adnan & Ahmad Zukarnain, 2020). En outre, la polyvalence de HoBAC lui permet également de s'appliquer aux systèmes informatiques conventionnels. Dans le contexte des systèmes IoT, HoBAC offre un cadre favorable dans la définition des politiques d'accès, en particulier pour les maisons intelligentes.

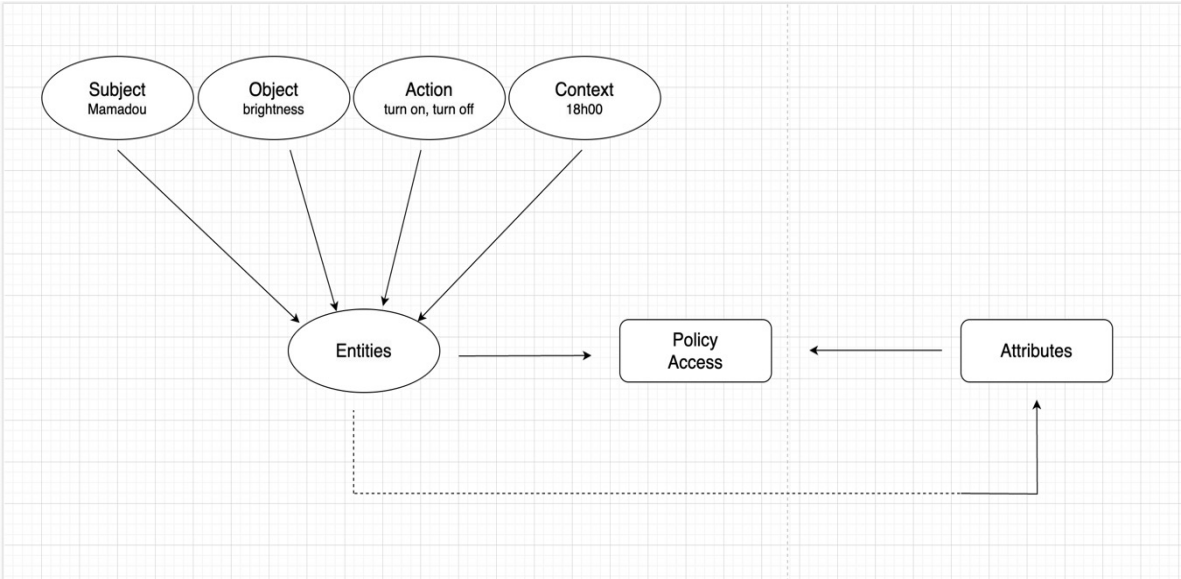

Figure 1 - Illustration de HoBAC et ses concepts

Ce modèle se compose de quatre concepts fondamentaux, chacun doté d'attributs spécifiques permettant de caractériser les entités impliquées dans le contrôle d'accès :

- Sujet, il représente l'entité cherchant à accéder à des ressources. Il est défini par des attributs tels que le nom, le prénom, l'âge et le rôle, qui déterminent ses droits d'accès.
- Objet, il désigne la ressource à laquelle l'accès est demandé. Dans le contexte des maisons intelligentes, il s'agit souvent de dispositifs connectés tels que des capteurs, des actionneurs ou des appareils domestiques. Chaque objet est caractérisé par des attributs tels que le nom, le type et l'état.
- Contexte, il représente les conditions environnementales qui influencent les décisions d'accès. Ces conditions incluent des variables telles que la température, l'heure ou d'autres facteurs pertinents pour les opérations de contrôle d'accès.

- Action, correspond aux opérations que le sujet est autorisé à effectuer sur les ressources de la maison intelligente. Ces actions peuvent inclure la lecture de données à partir de capteurs, la modification de paramètres ou l'activation/désactivation d'appareils.

Les politiques de contrôle d'accès sont établies en utilisant des règles logiques qui font référence aux attributs des entités mentionnées ci-dessus. Une règle de politique peut spécifier par exemple que si le sujet est un résident adulte (attribut=âge), alors il est autorisé à contrôler le thermostat (objet) si la température ambiante dépasse 18 degrés Celsius (attribut=température). Les concepts HoBAC, appliqués à une maison intelligente, englobent des attributs définis par un identifiant unique et une valeur, offrant une structure organisée aux entités et à leurs relations. Ces attributs sont regroupés en types d'attributs de manière sémantique, facilitant la création d'instances de types d'attributs à partir d'attributs individuels. Chaque type d'attribut est caractérisé par un nom et une fonction d'ordre supérieur, représentée par la S-Structure (Adda & Aliane, 2020). Dans le cadre de notre maison intelligente, diverses entités telles que les personnes et les dispositifs IoT comme le thermostat, la télévision, les pièces et les stores peuvent interagir. Le modèle HoBAC permet de structurer et de gérer ces entités et leurs relations, offrant ainsi une représentation précise de la maison intelligente et facilitant la gestion des interactions entre celles-ci. Une étude antérieure a également démontré la faisabilité de la fédération d'attributs dans le modèle HoBAC. Par exemple, les données de capteurs comme un thermostat ajustant la température et un capteur allumant les lumières lors de la détection d'une personne entrant dans une pièce peuvent être agrégées pour fournir une vue globale de l'état de la maison. Des politiques de contrôle d'accès peuvent ensuite être appliquées aux personnes souhaitant accéder à ces capteurs. Par exemple, si un individu nommé Mamadou souhaite accéder aux données de température du thermostat, une règle d'accès est mise en place. Si les attributs de Mamadou correspondent à ceux du thermostat dans un contexte donné, tel que les heures entre 9h00 et 18h00, l'accès est autorisé. Cette décision est basée sur l'égalité des attributs, également connue sous le nom de fonction d'unification (Fm). L'étude des principes de conception de HoBAC met en lumière l'introduction de nouvelles primitives pour la gestion des

changements et l'évaluation dynamique des attributs. Ces primitives, réparties en trois types, offrent une flexibilité dans la gestion des attributs en permettant des mises à jour à différents niveaux. L'architecture d'un système de contrôle d'accès basé sur HoBAC repose sur un cadre de politique d'AC comprenant des règles, des stratégies et un exécuteur de politique. Ces éléments définissent les relations entre les attributs et permettent de déterminer les autorisations d'accès. HoBAC trouve des applications dans divers scénarios réels, notamment dans l'IoT, où il améliore la sécurité et la flexibilité des systèmes d'AC en permettant une gestion fine et dynamique des autorisations en fonction des attributs des utilisateurs et des ressources. En tirant parti des fonctionnalités avancées de HoBAC, les systèmes d'AC peuvent mieux s'adapter aux environnements complexes de l'IoT, assurant ainsi une protection adéquate des données et des ressources sensibles (Adda & Aliane, 2020).

1.2.4 Les langages spécifiques au domaine : concepts, applications et perspectives

Les langages spécifiques aux domaines (DSL) sont des langages de programmation conçus pour résoudre des problèmes spécifiques dans un domaine d'application particulier, offrant une précision et une flexibilité supérieures par rapport aux langages à usage général comme Python, Java, etc.(Mernik, 2017). Ils sont particulièrement efficaces dans la gestion des politiques de sécurité où une précision fine est essentielle pour garantir la protection des données. Bien que ces langages aient existé aussi longtemps que l'informatique elle-même, leur exploration reste relativement limitée par rapport aux langages à usage général qui sont largement utilisés dans de nombreux domaines. Les techniques de développement DSL étaient souvent méconnues bien que les DSL soient des langages succincts, axés sur un aspect spécifique d'un système logiciel(Fowler, 2010). Deux types de DSL sont principalement utilisés. Les DSL internes qui sont des extensions de langages généraux existants (Java, Python, etc.), utilisant leur syntaxe et leur compilateur pour définir des abstractions spécifiques au domaine. Ils peuvent être comparés à des variantes d'un langage de programmation commun. En revanche, les DSL externes sont des langages indépendants avec leur propre syntaxe et leur propre compilateur. Bien qu'ils offrent une plus grande

expressivité et flexibilité, leur conception et leur mise en œuvre nécessitent davantage d'efforts (Voelter & Solomatov, 2010). Plusieurs DSL ont été proposés dans la littérature pour répondre aux besoins spécifiques en matière de sécurité. Parmi eux, on trouve des langages centrés sur les politiques de contrôle d'accès, tels que PML (Luo et al., 2019) et ADQL "Access Definition and Query Language" (Andreas Sonnenbichler, 2013). Ils illustrent l'objectif de créer des langages informatiques ciblés sur des problèmes spécifiques plutôt que des langages généralistes(Fowler, 2010). Des DSL permettent également de spécifier, vérifier ou de concevoir des systèmes IoT sécurisés. En effet, les systèmes IoT présentent des risques de vulnérabilités, d'attaques ou de pannes compte tenu de nombreux dispositifs connectés qui interagissent entre eux. La génération d'un méta-modèle de vérification pour les systèmes cyber-physiques IoT permet de définir un processus de vérification pour les environnements intelligents (Meixner et al., 2021). De plus, IoTSec est un DSL qui permet de spécifier des exigences de sécurité pour les systèmes IoT en utilisant une approche basée sur les modèles et les patrons (Minoli et al., 2017). Puis, IoTECS, un DSL qui permet de modéliser des architectures IoT sécurisées avec l'apprentissage profond (Yue et al., 2021). Ces DSL permettent de renforcer la sécurité des systèmes IoT, en utilisant des abstractions adaptées au domaine et en facilitant l'analyse et la génération de code. L'adoption d'approches décentralisées comme illustré par l'utilisation de la blockchain (Dorri et al., 2017) constitue un autre défi. Parallèlement, une approche alternative consiste à apporter une stratégie de modélisation graphique et à spécifier des configurations d'appareils IoT à l'aide d'une notation textuelle concise et intuitive (Kolovos & de la Vega, 2023). Il est également important de répondre au défi posé par le développement de logiciels pour les systèmes IoT (Erazo-Garzón et al., 2022) étant donné que les outils traditionnels de génie logiciel ne répondent pas aux exigences des environnements IoT. Un DSL permet d'exprimer des problèmes et des solutions dans les termes et les concepts propre au domaine, en utilisant une syntaxe et une sémantique dédiées. Cette approche favorise non seulement une communication efficace entre les experts du domaine et les développeurs, mais également la vérification, la validation et la génération de code (Mernik et al., 2005). Des projets ont été modélisés pour des systèmes IoT (Alfonso et al., 2021) illustrant l'utilisation de l'outil MPS

sur des architectures IoT auto adaptatives. C'est pourquoi nous nous concentrons sur la sécurité des environnements IoT, qui a un impact direct sur la vie quotidienne des citoyens (villes, industries, maisons, outils technologiques). L'approche distincte de la programmation orientée langage permet de créer des langages adaptés à des domaines spécifiques fournissant des notations et des abstractions propres au domaine pour simplifier le développement et la compréhension des applications. Pour concevoir de tels DSL, des outils comme MPS sont disponibles pour assister les développeurs dans le processus, résultant en des DSL avec des notations variées telles que des tableaux, des diagrammes ou des formules mathématiques. MPS, développé par JetBrains, a été fondé en 2000 et a émergé de la vision exposée dans l'article intitulé : « Language Oriented Programming: The Next Programming Paradigm » (Dmitriev, 2004) où le fondateur exposait sa vision d'une programmation basée sur les langages. La première version publique de MPS est lancée en 2009 (Jetbrains, 2009) et depuis, l'outil s'est enrichi des nouvelles fonctionnalités et de nouveaux langages sous la licence Apache 2.0 offrant une utilisation open source. MPS est également utilisé par d'autres sociétés ou organisations comme mbeddr, un système de développement embarqué basé sur C (Voelter et al., 2019), ou PEoPL, un outil pour l'ingénierie des lignes de produits logiciels (Behringer & Fey, 2016).

1.3 Problématique

L'essor fulgurant de l'IoT a redéfini les paradigmes technologiques, notamment dans le domaine des maisons intelligentes, soulignant ainsi la nécessité de concevoir des mécanismes de sécurisation adaptés. Notre recherche s'inscrit dans cette dynamique, explorant une problématique liée au manque d'outils appropriés pour le développement des systèmes IoT sécurisés, spécifiquement pour l'implémentation du modèle de contrôle d'accès HoBAC. Ce déficit d'outils compromet son intégration dans les environnements IoT, révélant une dimension critique dans la gestion de la sécurité au sein d'un écosystème hétérogène et complexes des systèmes IoT. Le déploiement des systèmes IoT sécurisés, particulièrement dans de secteurs essentiels tels que la santé, l'industrie et les infrastructures intelligentes met en évidence l'importance de résoudre cette problématique. En l'absence d'un cadre

d'application adapté pour le modèle HoBAC, ces systèmes sont exposés à des vulnérabilités significatives en matière de sécurité, les rendant susceptibles d'être ciblés par diverses attaques (Sivasakthi et al., 2024). Le contrôle d'accès émerge comme un enjeu majeur dans les systèmes IoT caractérisés par leur évolutivité, leur hétérogénéité et leur dynamique (Ragothaman et al., 2023). L'expansion fulgurante de l'internet des objets a intensifié ces enjeux de sécurité accentuant l'impératif de trouver des solutions appropriées. Bien que les DSL se révèlent performants dans la spécification des politiques de contrôle d'accès (Vistbakka et al., 2018), l'absence d'outils adéquats pour une intégration efficace du modèle HoBAC conduit les développeurs à se heurter à des obstacles significatifs lors de la conception et de l'implémentation de solutions IoT sécurisées. Les développeurs pourraient se forcés de recourir à des solutions de contournement susceptibles de compromettre davantage la sécurité du système informatique. L'emploi de XACML suggéré comme langage de politique d'accès adapté aux systèmes IoT pourrait constituer une plateforme flexible et compatible avec divers scénarios de contrôle d'accès, favorisant ainsi l'intégration du modèle HoBAC dans les applications IoT.

1.4 Objectifs

L'objectif central de cette recherche est de contribuer à l'enrichissement de l'écosystème d'outils associés à HoBAC par le développement d'un langage spécifique au domaine dénommé HoBACDSL. Conçu spéciquement pour faciliter la modélisation et l'implémentation des politiques de contrôle d'accès propres au modèle HoBAC, HoBACDSL s'appuiera sur MPS comme plateforme de développement. Son application dans le contexte des maisons intelligentes émane de l'impératif de renforcer la sécurité dans ces environnements hétérogènes et complexes. En réponse à la problématique énoncée précédemment, HoBACDSL permettra de générer des politiques d'accès exploitables au format XACML, un langage basé sur du XML facilitant l'articulation des politiques de

contrôle d'accès. Ces dernières, une fois générées, serviront à réguler l'accès aux ressources des maisons intelligentes en définissant des règles spécifiques. Cette approche s'appuie sur la technique de l'ingénierie des langages telle qu'illustrée par les travaux de (Voelter et al., 2019). Cette étude met en avant l'efficacité de l'utilisation de MPS pour le développement de mbeddr. Par cette démarche, nous visons à abstraire les complexités techniques inhérentes à HoBAC, en s'appuyant sur divers attributs associés aux dispositifs (ressources), aux utilisateurs, aux actions et aux contextes pour définir les politiques d'accès tout en respectant les exigences du modèle. L'objectif est de déployer une solution cohérente qui renforce efficacement la gestion de la sécurité au sein des maisons intelligentes.

1.5 Méthodologie

Pour atteindre notre objectif de conception de HoBACDSL, nous avons initié notre projet par une analyse de la littérature sur l'IoT, le contrôle d'accès, et les DSL. Cette phase préliminaire a permis d'identifier les défis de sécurité et les contraintes existantes dans ces domaines, tout en acquérant une compréhension des principes sous-jacents. Parallèlement, nous avons investi du temps dans l'apprentissage des concepts, de la syntaxe et des fonctionnalités du MPS. Cette étape, incluant l'étude des langages, des éditeurs, des contraintes, etc., s'est révélée essentielle pour notre capacité à développer un DSL fonctionnel qui générera du code XML conforme aux spécifications XACML pour les politiques de contrôle d'accès HoBAC. En nous appuyant sur la composition d'un DSL (syntaxe abstraite, concrète, sémantique), nous avons modélisé différents concepts intervenants dans une maison intelligente en établissant une représentation textuelle de ces entités, accompagnée d'une liste d'attributs associés à chaque concept. Les dispositifs IoT, par exemple, sont représentés par des concepts tels que Miroir, Thermostat, TV etc. Le processus de conception d'un DSL implique plusieurs étapes, notamment l'élaboration d'un modèle de données au sein de la syntaxe abstraite, transposée de manière textuelle dans la syntaxe concrète, destinée à être manipulée par les utilisateurs. Cette séquence d'étapes imbriquées nous permettra à définir la sémantique de notre langage. Le générateur reproduira le code source en entrée (syntaxe abstraite) dans un autre langage cible, le langage XML

(Rabinovich et al., 2017). La modélisation des concepts XACML s'articule autour du concept de haut niveau Policy qui hérite de PolicySet. Nous avons défini d'autres concepts tels qu'AttributeDesignator, Attribute, AttributeValue, Match, Rule, etc., en les imbriquant selon la structure XACML. En mappant ces concepts et ceux de la maison intelligente, l'utilisateur est en mesure de manipuler les concepts, de définir des règles et générer du code XML. Ce code reproduit les capacités de HoBAC en termes d'expressivité, de flexibilité ainsi que les opérations d'agrégation sur les attributs sans oublier sa propriété de hiérarchisation des entités. À l'aide des diagrammes modélisés, nous avons établies une meilleure structure entre les concepts et les attributs utilisés pour créer des hiérarchies d'entités et qui serviront de base à la création de notre HoBACDSL. Notre approche d'évaluation met en évidence la capacité à spécifier les règles d'accès et à générer le code XACML dans le DSL conformément à ces règles. De plus, nous envisageons d'intégrer les principes d'utilisabilité du DSL comme indiqué dans cette étude (Poltronieri et al., 2021). Ces principes pourraient nous aider à évaluer et à améliorer l'utilisation de HoBACDSL.

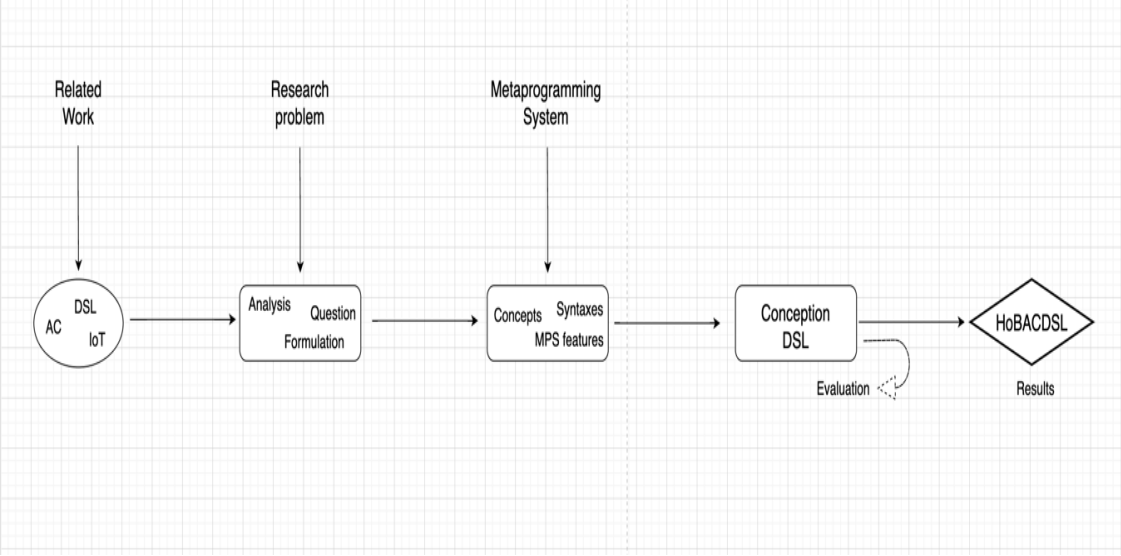

Figure 2 – Schéma de méthodologie

1.6 Contributions

Dans cette recherche, nous avons mis l'accent sur l'évaluation de notre DSL en montrant les résultats obtenus grâce à notre approche de modélisation MPS appliquée au contrôle d'accès HoBAC dans les maisons intelligentes. Exploitant le paradigme de la programmation orientée Langage pour la sécurité IoT avec MPS, nous avons développé un DSL modulaire, adapté aux dispositifs IoT et générant des politiques de contrôle d'accès converties en code XACML. L'introduction des règles syntaxiques et sémantiques et l'applicabilité de ce DSL dans la mise en œuvre des politiques d'accès montrent sa capacité à générer du code XACML conforme aux normes. Diverses règles telles que PolicyAccess, ont été implémentées pour répondre à des scénarios d'accès spécifiques. Par exemple, l'ouverture de la porte d'entrée pour les livreurs sous conditions ou l'ajustement du thermostat pour les propriétaires. L'évaluation de HoBACDSL vise à prouver sa capacité à répondre aux besoins de sécurité spécifiques aux maisons intelligentes, en fournissant une solution modulaire pour la formulation et la gestion des politiques de contrôle d'accès.

1.7 Structure du mémoire

Ce mémoire structuré en trois parties adopte le format d'un mémoire par articles. La première partie donne une vue d'ensemble du domaine en exposant le contexte et la problématique de recherche inhérentes aux défis de sécurité dans les systèmes l'IoT, les maisons intelligentes en particulier, les contrôle d'accès et les DSL. L'établissement des objectifs de l'étude en détaillant les approches, la contribution et la méthodologie de travail terminent ce premier segment. Le premier chapitre présente un article scientifique proposant HoBACDSL comme langage spécifique au domaine axé sur HoBAC, publié dans la revue Procedia Computer Science, accessible en open source (Diallo & Adda, 2024). Le mémoire se termine par une conclusion générale de l'étude en présentant un résumé des principaux résultats, en discutant des implications et perspectives pour les travaux futurs.

CHAPITRE 1
HOBACDSL : LANGAGE SPÉCIFIQUE AU DOMAINE DU CONTRÔLE D'ACCÈS CENTRÉ SUR HOBAC

1.1 RÉSUMÉ EN FRANÇAIS DE L'ARTICLE

Le contrôle d'accès dans les systèmes IoT présente des défis importants. Ces systèmes sont évolutifs, hétérogènes et dynamiques, ce à quoi s'ajoute le manque d'outils appropriés pour spécifier et concevoir des politiques de contrôle d'accès. Malgré diverses propositions, la complexité persiste, en particulier dans des environnements restrictifs comme l'IoT. Dans cette perspective, le contrôle d'accès basé sur les attributs d'ordre supérieur HoBAC a été proposé comme un modèle général de contrôle d'accès qui étend le contrôle d'accès basé sur les attributs ABAC. Il permet de concevoir des modèles et des politiques de contrôle d'accès flexibles applicables aux systèmes IoT et non IoT. Les travaux présentés dans cet article visent à répondre au besoin d'outils pour soutenir l'adoption de HoBAC pour les systèmes IoT en proposant HoBACDSL, un langage spécifique au domaine (DSL). HoBACDSL fait abstraction des complexités de HoBAC et rend ses concepts plus accessibles. Nous illustrons comment ce DSL est concrètement utilisé pour spécifier et générer des politiques de contrôle d'accès dans la norme XACML (Extensible Access Control Markup Language) pour un cas d'utilisation de maison intelligente. Notre article est publié dans la revue Procedia Computer Science, accessible en open source (Diallo & Adda, 2024).

1.2 HoBACDSL : langage spécifique au domaine du contrôle d'accès centré sur HoBAC

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2019) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

The 19th International Conference on Future Networks and Communications (FNC)
August 5-7, 2024, Marshall University, Huntington, WV, USA

HoBACDSL: HoBAC-focused Access Control Domain Specific Language

Mamadou Mouslim Diallo[a], Mehdi Adda*[a]

[a]*Department of Mathematics, Computer Science and Engineering, University of Quebec at Rimouski, Quebec, Canada*

Abstract

Access control (AC) in IoT (Internet of Things) systems presents significant challenges. These systems are evolving, heterogeneous, and dynamic, combined with the lack of appropriate tools to specify and design Access Control Policies (ACP). Despite various proposals, complexity persists, especially in restrictive environments like IoT. In this perspective, Higher-Order Attribute-Based Access Control (HoBAC) was proposed as a general AC model that extends Attribute Based AC (ABAC). It allows the design of flexible AC models and policies applicable to IoT and non-IoT systems. The work presented in this paper focuses on addressing the need for tools to support the adoption of HoBAC for IoT systems by proposing HoBACDSL, a Domain Specific Language (DSL). HoBACDSL abstracts the complexities of HoBAC and makes its concepts more accessible. We illustrate how this DSL is concretely used to specify, and generate AC policies in the Extensible Access Control Markup Language (XACML) standard for a Smart Home use case.

Keywords: Access Control; IoT; HoBAC; MPS; XACML ;

1. Introduction

The Internet of Things (IoT) systems play a predominant role in our daily lives, integrating connected devices in various fields such as health, agriculture, industry, and Smart Homes. These connected devices include smart watches, surveillance systems, and household equipment such as fridges and smart dustbins [12]. The benefits of IoT, such as improving quality of life, optimizing resources, and creating new services, are considerable, not to mention the vast market opportunity it offers[9]. However, these technological advances raise significant security issues that could affect its development, exposing data and services to risks of attacks, thefts, or diversions[13]. These issues include

* Corresponding author. Tel.: +1-514-703-7709 ;
E-mail address: diam0247@uqar.ca

protecting our personal data and preventing attacks and intrusions. The IoT differs from conventional IT platforms in that it is heterogeneous, consisting of various devices with varying capabilities, communication protocols, and manufacturing origins. This heterogeneous aspect presents a unique challenge for establishing a homogeneous access control system. In addition, IoT environments are scalable and dynamic, involving frequent changes in the network context, hence the need for a flexible and adaptive access control system. Moreover, many IoT devices have limited computing and storage capacities, further complicating the implementation of conventional access control models [3]. Access control (AC) is a fundamental security aspect. It aims to define and apply policies by specifying who can access what, when, where, and how [22]. In this perspective, Aliane and Adda proposed a Higher-order Attribute-Based general AC model (HoBAC)[5, 2]. Even though several models exist to define AC models and policies, HoBAC stands out for its flexibility and the expression of adaptive access policies to IoT and non-IoT environments. HoBAC extends the Attribute-Based Access Control model (ABAC)[11], which is one of the most popular models and has received particular attention in recent years [23, 24]. HoBAC is still in its infancy and lacks the tools and ecosystem to support its adoption. This article, contribute to addressing this need by designing and implementing a Domain Specific Language (DSL) dedicated to HoBAC. DSLs are programming languages specialized in a particular domain that simplify and improve software development by using concepts and notations adapted to the domain[16, 15]. HoBACDSL is a DSL we designed and implemented using JetBrains' MPS workbench[25, 26]. It abstracts the HoBAC high-level concepts, makes them more accessible, and automatically generates AC policies according to the XACML (eXtensible Access Control Markup Language) standard.

The remainder of this paper is structured as follows: Section 2 highlights existing work in the field. Section 3 presents the HoBAC main concepts. Sections 5 and 6 detail the design methodology and implementation HoBACDSL. Section 7 discusses our contributions, before presenting the conclusion in Section 8.

2. Related work

The IoT paradigm encompasses various application areas, such as health, industry, agriculture, or home automation. These areas involve connected devices that collect, transmit, and process data across the internet[6, 10, 7]. Moreover, the rapid rise of IoT usage has generated major security concerns. One of the main concerns is the exploitation of user data[1]. People are particularly concerned about security and privacy. This 2022 study shows that most do not believe IoT products are secure. "Less than a third of users trust manufacturers, service providers, and governments to safeguard their privacy" [27]. IoT devices are particularly vulnerable to network attacks such as data theft, phishing attacks, spoofing, and denial-of-service attacks. These attacks can lead to other cybersecurity threats, such as ransomware attacks and serious data breaches that can cost companies a lot of money and effort to recover [28].

A survey conducted by[14] highlights various security challenges in its layered structure, particularly in the context of Smart Homes. This underscores the importance of implementing home security systems, as highlighted by this investigative study[8]. On the other hand, traditional access control models, such as DAC, MAC, RBAC, and ABAC, show their limitations in IoT contexts due to their rigidity and inability to handle the dynamic and heterogeneous nature of IoT devices [22]. HoBAC addresses these limitations by offering more granular, context-sensitive access control.

Over the past decade, researchers have proposed several DSLs to express AC rules clearly, concisely, and abstractly. For example, PML, introduced by Luo et al.[17], is an AC policy language that operates on an interpreter for Web services. Additionally, the AC meta-model, which closely resembles SQL, is named ADQL (Access Definition and Query Language), as described by Sonnenbichler in [18]. It is also important to mention XACML, a declarative XML-based AC policy language, designed to express and enforce AC policies [19, 21]. The Axiomatics Language for Authorization (ALFA) is a domain-specific language designed to simplify the creation and management of XACML policies[20]. By providing a more intuitive, higher-level syntax than XACML's XML-based structure, ALFA makes it easier for policy authors to define, read, and maintain complex AC policies. It is important to note that existing AC DSLs and tools lack support for HoBAC, which motivates this work to propose a DSL tailored specifically for HoBAC.

3. HoBAC: Overview and Key Concepts

HoBAC, a generalization of ABAC, is based on attributes that make up four basic concepts called entities: Subject, Object, Context, and Action. To illustrate HoBAC in the context of Smart Homes, we define each entity class by a symbolic representation where:

- HS = Subject, is a user of the Smart Home who has attributes such as first name, last name, age, role, etc;
- HO = Object, is a connected device of the Smart Home, which is the entity representing the resource that also has attributes like name, type, state;
- HC = Context, represented by environmental conditions influencing access decisions. It also has attributes like temperature, time;
- HA = Action, which specifies the operations that the subject wants to perform on the resource;

As HoBAC favors the construction of hierarchies of entities using aggregation and composition relationships. Notably, the instantiation of those entities may lead to the definition of distinct hierarchies [2].

Access control policies are thus defined using logical rules based on these attributes of the Smart Home entities. For example, rule R1 specifies that the subject who has named "Diallo," wants to access the Thermostat resource and perform the action of lowering the temperature if the context is that the temperature is equal to or greater than 22 degrees Celsius.

$$R1 = (HS.firstname = "Diallo") \wedge (HO.type = "Thermostat") \wedge (HA.name = "Lower") \wedge (HC.temperature \geq 22) \rightarrow Authorize!$$

This rule comprises four parts (HS.name, HO.type, HA.name, HC.Temperature), each corresponding to a condition on an attribute of an entity followed by the decision "Authorize" if the expression is evaluated to be True (cf. Fig. 1).

Concepts	**Description**	**Attributes**
Subject	Admin, Owner, Visitor, guest	Name, login, password, address, fonction, etc
Object	House, Room, Piece, Camera, Store, Thermostat, Lamp, Alarm, TV	name, size, state, number, consumption
Action	Turn On, View, Activate, Turn off, Deactivate, Adjust	
Context	Temperature, Time, Brightness, Day	date, GPS position
Access Rights	Admin access, owner, guest, visitor, Authorize, Refuse, Delete, Modify	
Process	Identification, Authentication, Authorization	

Fig. 1. HoBAC Concepts Applied to Smart Home

Fig. 2 illustrates one of the access principles of home automation where different users (admins, owners, and guests) are granted different levels of access to control various devices in the house. Depending on their previligies, users can control various IoT devices, such as TVs and security cameras.

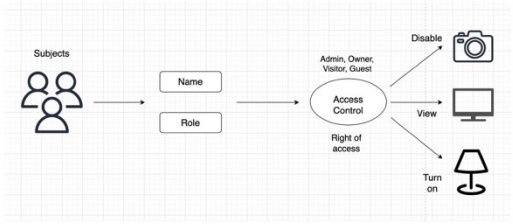

Fig. 2. Example of an access control scheme related to home automation

4. Methodology

The section details the creation and fine-tuning of HoBACDSL, starting with using the Meta Programming System (MPS) for defining essential concepts and syntaxes. Key to this process was establishing a DSL that can generate XML code aligned with XACML specifications for HoBAC policies. The development steps covered designing the DSL's core components—abstract syntax, concrete syntax, and semantics—and mapping Smart Home use case concepts to specific access policy attributes. Finally, the DSL was tested and evaluated for its effectiveness in integrating with HoBAC to manage access policies in Smart Home environments.

5. Design and Modeling of HoBACDSL

5.1. DSL Concepts

This section explains the HoBACDSL design process under MPS workbench, its application in modeling techniques for IoT systems, focusing on Smart Homes. The process is depicted in Fig. 3. Step 1 defines the key concepts and their logical structure, forming the abstract syntax. In step 2, these concepts are rendered visually through an editor, forming the concrete syntax. Editors enable the transition from abstract to concrete syntax, and these two aspects are intimately linked, as evidenced by the bidirectional arrows. Step 3 involves the conversion of concrete syntax into semantics by applying rules defined in sandboxes (Step 4). These sandboxes establish directives that govern the behavior of language elements in an interactive context. In Step 5, the DSL rules, representing the semantic part of the language, are transformed into a code structure in the target language, in this case XML. This transformation enables us to move from the conceptual to the concrete, generating operational code from the language specifications.

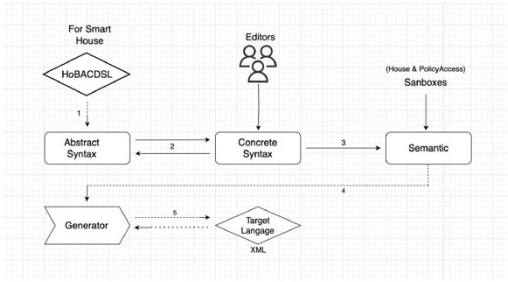

Fig. 3. DSL design workflow

5.1.1. Abstract Syntax

The abstract syntax refers to the high-level domain concepts and their interrelations within the language framework. The diagram of Fig 4 showcases a Smart Home and its components. This high-level representation serves as a preliminary example, setting the stage for a more detailed examination of the use case in the subsequent sections. The concept of the House is central and represents our Smart Home, characterized by attributes such as "ID" and "name" which allow the identification and designation of the house. Each concept can have a list of attributes. A composition relationship exists between the Room and the House, as a house can consist of several rooms. Bedroom, Living Room, Bathroom, and Kitchen are specific types of rooms. IoT devices are represented by concepts such as Mirrors, Smart Faucets, TVs, Microwaves, Lamps, and Smart Lock, which can be found in several rooms. AC policy rules manage access to those devices through HoBACDSL. This UML diagram of Fig. 4 highlights the entities and their relationships.

5.1.2. Concrete Syntax

Concrete syntax is the visual or textual representation of the elements of the language and their relations. From this point of view, it defines the form of the domain concepts that the language allows to express. There can be several

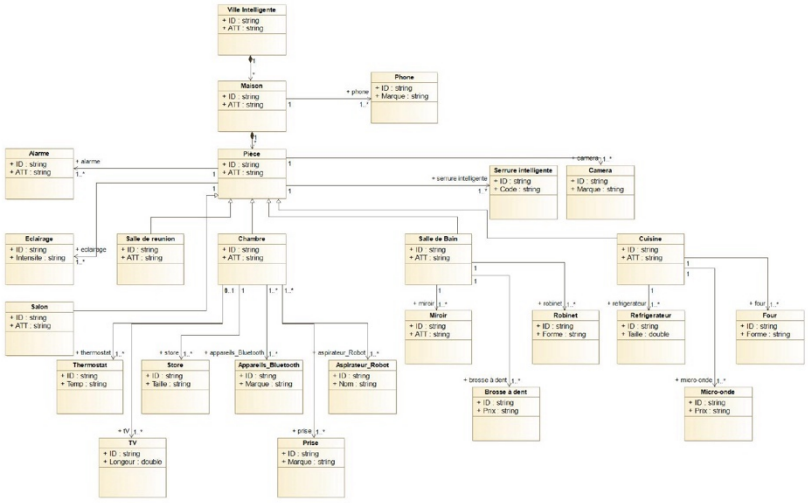

Fig. 4. Diagram of Smart Home concepts and their relations

concrete syntaxes for the same abstract syntax. For HoBACDSL, we have chosen to use a textual concrete syntax. It allows modeling all the concepts by creating easily manipulable editors. A block of text representing each room in the house has sub-blocks for each device in that room, including their hierarchical relationships. We have used various keywords to represent the devices. (Fig. 5) and (Fig. 6) present an illustration of the editor of the Policy concept and the Piece concepts along with their hierarchies. The need for flexibility and adaptability motivates our choice of HoBACDSL design to meet the specific requirements of IoT environments. MPS provides adequate tools for syntax definition, transformation and code generation.

Semantics defines the meaning of the domain concepts that one manipulates. It defines the interpretation or execution of the language elements and their relations. MPS allows the creation of semantics with actions or transformations, which correspond to operations on the language elements. HoBACDSL semantics are based on actions that allow checking or modifying the access policies to resources shared by IoT devices.

For example, we can define a checkAccess action to verify if a user has the right to access a resource in a house. We can proceed similarly to define the actions of the other elements of HoBACDSL, like Devices, TV, Thermostat, etc.

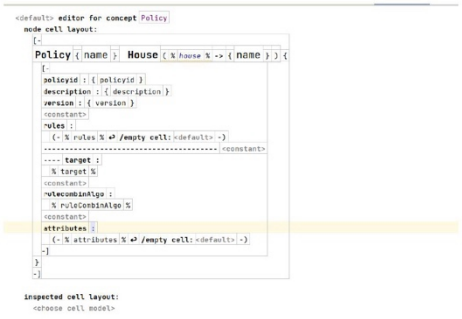

Fig. 5. Projection editor of the policy concept

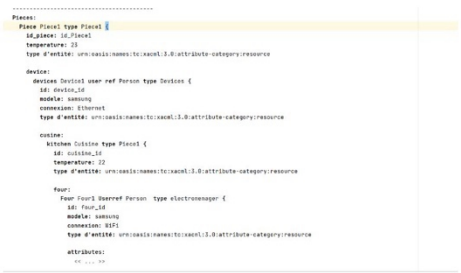

Fig. 6. Sandbox editor of hierarchies of concepts

6. Implementation

This section focuses on the concrete implementation of HoBACDSL including code snippets and examples of how the DSL translates high-level policies into XACM in the context of a Smart Home system. Having defined the essential concepts and their textual representations, we transition to defining semantics and generating XACML-complaint AC policy. We also discuss the challenges encountered.

6.1. Modeling

The approach to modeling XACML concepts in HoBACDSL centers on the high-level Policy concept, an extension of PolicySet. This includes defining essential elements like AttributeDesignator, Attribute, AttributeValue, Match, Rule, and more, organized in alignment with the XACML standard. We created two Sandboxes illustrated in Fig. 8 to facilitate user interaction with Smart Home and XACML concepts. These sandboxes enable users to define rules and generate XML code by manipulating entities and their attributes. Key XACML elements such as AttributeDesignator, which identifies a specific attribute for rule evaluation, and Attribute, which denotes a particular characteristic related to a subject, resource, action, or environment, are utilized. AttributeValue denotes an attribute's value, and Match compares the AttributeValue with an AttributeDesignator's value.

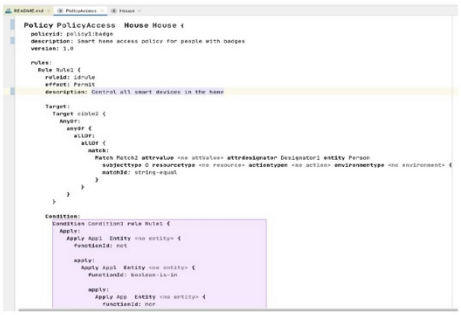

Fig. 7. Sandbox editor of the DSL for PolicyAcess

Fig. 8. Sandbox editor of DSL for Home

To facilitate XML generation, we integrated a module dedicated to XML manipulation. This module uses XML templates where we mapped several concepts from the generator. Policy and House are two high-level concepts to which other concepts are linked. The "Map configuration main" function contains the list of mapped elements, including several reduction rules of Smart Home concepts as shown in figures (Fig. 9) and (Fig. 10).

Fig. 9. Example of Mapping on the Policy Concept

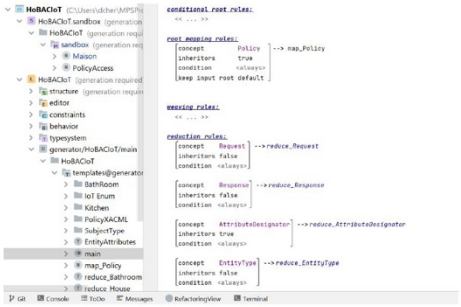

Fig. 10. HoBACDSL concepts mapped in the generator

An ”EntityType” concept is created to represent all types of entities, with other concepts such as House, Person, and Devices inheriting from it. Respecting the structure of XACML and the concepts we modeled, a reference relationship is established between Policy and “House” to define access rules.

Modeling the complex relationships and dependencies between the various HoBAC and XACML concepts required careful planning and structuring to avoid redundancy and ensure smooth integration. We used UML diagrams to test and validate relationships before final implementation. Generating XACML-compliant XML code was also tricky, requiring careful attention to syntax. We integrated XML templates into MPS and used mapping functions to standardize and simplify this process.

7. HoBACDSL AC Policy example

In this section, we showcase the effectiveness of HoBACDSL to generate XACML-compliant AC policy rules. Due to the lack of space, only one rule is illustrated below. More rules and the source code of HoBACDSL may be found on GitHub[1]

- A rule that grants a ”delivery person” access to open the front door if they possess a valid access code (CODE) and the owner has confirmed the delivery (indicated by the attribute CodeValidated:House being ’TRUE’). Access is permitted under these conditions (Fig. 11).

Fig. 11. Access Rule for Delivery Person Authorization

This XML rule shows how HoBACDSL can translate high-level access control rules into precise XACML specifications.

8. Concluding remarks

This paper introduces HoBACDSL, a DSL to streamline the creation of XACML-compliant HoBAC access control policies for Smart Homes. By demonstrating the DSL’s ability to abstract access control concepts and facilitate policy specification, we have highlighted its potential to simplify security management in IoT environments. We recognize the need for further refinement and real-world testing to enhance its functionality and applicability in addressing the nuanced security demands of general IoT environments.

Acknowledgement

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference number 06351]).

[1] https://github.com/addame/HoBACDSL/tree/master

References

[1] Abbassi, Younes and Benlahmer, Habib, "An overview on the security of the internet of things (IOT)", in *Symposium on Connected Objects and Systems-COC'2021*, Conference Proceedings, 2021.

[2] Adda, Mehdi and Aliane, Linda, "HoBAC: Fundamentals, principles, and policies", *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 5927-5941, 2020.

[3] Moore, Samuel J and Nugent, Chris D and Zhang, Shuai and Cleland, Ian, "IoT reliability: a review leading to 5 key research directions", *CCF Transactions on Pervasive Computing and Interaction*, vol. 2, pp. 147-163, 2020.

[4] Ragothaman, Kaushik and Wang, Yong and Rimal, Bhaskar and Lawrence, Mark, "Access control for IoT: A survey of existing research, dynamic policies and future directions", *Sensors*, vol. 23, no. 4, pp. 1805, 2023.

[5] Aliane, Linda and Adda, Mehdi, "HoBAC: Toward a higher-order attribute-based access control model", *Procedia Computer Science*, vol. 155, pp. 303–310, 2019.

[6] Ashton, Kevin, "That 'internet of things' thing", *RFID journal*, vol. 22, no. 7, pp. 97-114, 2009.

[7] Atzori, Luigi and Iera, Antonio and Morabito, Giacomo, "The internet of things: A survey", *Computer networks*, vol. 54, no. 15, pp. 2787-2805, 2010.

[8] Chitnis, S and Deshpande, N and Shaligram, A, "An investigative study for Smart Home security: Issues, challenges and countermeasures. Wireless Sensor Network", vol. 8, no. 4, pp. 61-68, 2016.

[9] Emmanuel Bertin, Dina Hussein, Cigdem Sengul et Vincent Frey, "Access control in the Internet of Things A survey on existing approaches and open research questions", 2019.

[10] Gubbi, Jayavardhana and Buyya, Rajkumar and Marusic, Slaven and Palaniswami, Marimuthu, "Internet of Things (IoT): A vision, architectural elements, and future directions", *Future generation computer systems*, vol. 29, no. 7, pp. 1645-1660, 2013.

[11] Hu, Vincent C and Kuhn, D Richard and Ferraiolo, David F and Voas, Jeffrey, "Attribute-based access control", *Computer*, vol. 48, no. 2, pp. 85-88, 2015.

[12] Kumar, Sachin and Tiwari, Prayag and Zymbler, Mikhail, "Internet of Things is a revolutionary approach for future technology enhancement: a review", *Journal of Big data*, vol. 6, no. 1, pp. 1-21, 2019.

[13] Mohanty, Jayashree and Mishra, Sushree and Patra, Sibani and Pati, Bibudhendu and Panigrahi, Chhabi Rani, "IoT security, challenges, and solutions: a review", *Springer Proceedings of ICACIE 2019, Volume 2*, pp. 493-504, 2021.

[14] Touqeer, Haseeb and Zaman, Shakir and Amin, Rashid and Hussain, Mudassar and Al-Turjman, Fadi and Bilal, Muhammad, "Smart Home security: challenges, issues and solutions at different IoT layers", *The Journal of Supercomputing*, vol. 77, no. 12, pp. 14053-14089, 2021.

[15] Voelter, Markus and Benz, Sebastian and Dietrich, Christian and Engelmann, Birgit and Helander, Mats and Kats, Lennart CL and Visser, Eelco and Wachsmuth, GH, *DSL engineering-designing, implementing and using domain-specific languages*, M Volter/DSLBook. org, 2013.

[16] Fowler, Martin, *Domain-specific languages*, Pearson Education, 2010.

[17] Luo, Yang and Shen, Qingni and Wu, Zhonghai, *Pml: An interpreter-based access control policy language for web services*, arXiv preprint arXiv:1903.09756, 2019.

[18] Sonnenbichler, Andreas, *An Access Definition and Query Language: Towards a Unified Access Control Model*, KIT Scientific Publishing, 2014.

[19] OASIS, "eXtensible Access Control Markup Language (XACML) Version 3.0", *OASIS Standard*, January 22, 2013. Available: https://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html.

[20] Brossard, David and Gebel, Gerry and Berg, Mark, "A Systematic Approach to Implementing ABAC", *ACM Proceedings of the 2nd ACM Workshop on Attribute-Based Access Control*, pages 53–59, 2017. DOI: 10.1145/3041048.3041051.

[21] Godik, Simon and Moses, Tim, "Oasis Extensible Access Control Markup Language (XACML)", *OASIS Committee Specification cs-xacml-specification-1.0*, 48, 2002.

[22] Ragothaman, Kaushik and Wang, Yong and Rimal, Bhaskar and Lawrence, Mark, "Access control for IoT: A survey of existing research, dynamic policies and future directions", *Sensors*, volume 23, number 4, pages 1805, 2023. MDPI.

[23] Ameer, Safwa and Benson, James et al., "Hybrid approaches (ABAC and RBAC) toward secure access control in smart home IoT", *IEEE Transactions on Dependable and Secure Computing*, 2022, IEEE.

[24] Vijayalakshmi, K and Jayalakshmi, V, "A study on current research and challenges in attribute-based access control model", In *Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2021*, pp. 17–31, 2022, Springer.

[25] Bucchiarone, Antonio and Cicchetti, Antonio and Ciccozzi, Federico and Pierantonio, Alfonso, "Domain-specific languages in practice: with JetBrains MPS", 2021, Springer Nature.

[26] Prinz, Andreas, "Teaching Language Engineering Using MPS", In *Domain-Specific Languages in Practice: with JetBrains MPS*, pp. 315–336, 2021, Springer.

[27] Schuster, Frederik and Habibipour, Abdolrasoul, "Users' privacy and security concerns that affect IoT adoption in the home domain", *International Journal of Human–Computer Interaction*, pp. 1–12, 2022, Taylor & Francis.

[28] Husar, A, "IoT Security: 5 cyber-attacks caused by IoT security vulnerabilities", Accessed: Jul, vol. 7, pp. 2023, 2022.

CONCLUSION GÉNÉRALE

2. RÉSUMÉ

Notre recherche a exploré la conception et le développement de HoBACDSL avec pour objectif l'adaptation du modèle de contrôle d'accès d'ordre supérieur HoBAC aux besoins spécifiques des maisons intelligentes. L'intégration de HoBACDSL avec XACML a permis de formaliser la structuration des politiques de sécurité dans les systèmes IoT. En tenant compte des particularités inhérentes à HoBAC, telles que le sujet, l'objet, le contexte et l'action, nous avons modélisé ses concepts en respectant les structures hiérarchiques requises. Diverses règles d'accès reflétant ces éléments ont été implémentées. HoBACDSL est textuellement éditable dans des environnements de type Sandbox offrant une flexibilité en s'adaptant aux besoins spécifiques des utilisateurs. Cette flexibilité se manifeste par la simplicité avec laquelle les utilisateurs peuvent personnaliser les règles d'accès, en tenant compte de la diversité des scénarios de sécurité dans les maisons intelligentes. HoBACDSL abstrait les concepts de haut niveau de HoBAC, les rendant plus accessibles, et génère automatiquement du code conforme à XACML. Cette fonctionnalité facilite la spécification et l'exécution des politiques de contrôle d'accès pour les experts du domaine. Nous soulignons l'importance évidente de HoBACDSL dans la définition et l'implémentation des politiques de contrôle d'accès pour les maisons intelligentes. Les perspectives offertes par les travaux futurs pourraient se concentrer sur l'extension des fonctionnalités de HoBACDSL pour répondre aux exigences croissantes de sécurité. Ces améliorations envisagées pourraient inclurent l'intégration de fonctionnalités avancées telles que la gestion avancée des autorisations, le support de politiques de contrôle d'accès dynamiques basées sur des changements contextuels, l'intégration de techniques d'apprentissage automatique pour une détection proactive des menaces, ainsi que l'implémentation de mécanismes de chiffrement

et d'authentification pour sécuriser les communications entre les appareils connectés. Cette évolution renforcera davantage l'applicabilité de HoBACDSL dans des scénarios concrets de maisons intelligentes.

3. LIMITATIONS

Les limitations identifiées concernant HoBACDSL pour la génération des politiques de contrôle d'accès pour HoBAC en XACML dans le contexte des maisons intelligentes mettent en évidence la nécessité d'améliorer le DSL. L'absence d'un langage dédié capable de simplifier l'application des concepts du modèle HoBAC dans l'environnement IoT s'est avérée être un défi majeur. Cette lacune a été exacerbée par la nécessité de définir précisément et de manière cohérente les contraintes des attributs des maisons intelligentes sous MPS, ainsi que par la désignation des valeurs des attributs pour une entité donnée. La corrélation et la mise en correspondance entre les concepts de la maison intelligente et ceux du modèle HoBAC ont nécessité une analyse en profondeur ajoutant ainsi une complexité supplémentaire au processus de développement du DSL. Malgré ces défis, notre capacité à exprimer des politiques d'accès conformes aux exigences XACML pour interroger les ressources d'une maison intelligente en se basant sur les concepts fondamentaux combinés aux concepts du modèle de contrôle d'accès a été démontrée. Cependant, des pistes d'amélioration restent à explorer afin d'optimiser notre DSL pour fournir une solution plus cohérente et efficace pour la gestion de la sécurité dans une maison intelligente. En outre, notre utilisation exclusive du modèle de contrôle d'accès HoBAC pour concevoir notre DSL ainsi que l'utilisation d'un seul langage cible (XACML) pour générer du code limitent la portée et la généralité de notre approche, ce qui peut affecter la compatibilité et l'interopérabilité de HoBACDSL. En dépit d'une dizaine de règles Policy implémentés, l'évaluation de notre DSL sur un nombre limité de scénarios et de cas d'étude peut restreindre la validité et la fiabilité de nos résultats en soulignant la nécessité d'étendre nos tests à des situations plus variées et complexes.

4. OBJECTIFS FUTURS

Les limitations identifiées soulignent la nécessité d'apporter des améliorations à HoBACDSL afin d'en optimiser sa portée. Une validation pratique, reposant sur des scénarios réels pourrait révéler des opportunités d'optimisation pour résoudre les problèmes identifiés et renforcer l'efficacité de HoBACDSL dans l'expression des politiques d'accès. Pour améliorer la cohérence et réduire les redondances, des énergies devront être orientées vers une optimisation du processus de génération de code XACML et une extension des fonctionnalités. L'intégration de nouveaux concepts XACML accompagnés d'une modélisation avancée, ainsi que la simplification des syntaxes du DSL pourrait être envisagée. Ces ajustements visent à rendre HoBACDSL plus intuitif, sans pour autant compromettre sa flexibilité nécessaire aux besoins variés. En plus, des efforts devront être concentrés sur une modélisation plus intuitive adaptée aux maisons intelligentes afin de surmonter les complexités liées à la définition précise des contraintes d'attributs et des valeurs associées. Étant donné le défi de sécurité dans l'IoT, l'évaluation de HoBACDSL devrait s'étendre à des scénarios plus complexes et diversifiés, impliquant des ressources IoT issues de différents domaines tels que la santé, l'éducation ou le transport, afin de valider la robustesse et l'applicabilité de HoBACDSL dans divers contextes.

RÉFÉRENCES BIBLIOGRAPHIQUES

1. Abbassi, Y., & Benlahmer, H. (2021, March 2021). Un aperçu sur la sécurité de l'internet des objets (IOT). Colloque sur les Objets et systèmes Connectés-COC'2021, IUT d'Aix-Marseille, Marseille, France.
2. Adda, M., & Aliane, L. (2020). HoBAC: Fundamentals, principles, and policies. Journal of Ambient Intelligence and Humanized Computing, 11, 5927-5941.
3. Adnan, M. H., & Ahmad Zukarnain, Z. (2020). Device-to-device communication in 5G environment: Issues, solutions, and challenges. Symmetry, 12(11), 1762.
4. Ait El Hadj, M., Benkaouz, Y., Khoumsi, A., & Erradi, M. (2018, December 11-13, 2017). Access Domain-Based Approach for Anomaly Detection and Resolution in XACML Policies. Innovations in Bio-Inspired Computing and Applications: Proceedings of the 8th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2017) held in December 11-13, 2017, Marrakech, Morocco.
5. Alam, M. R., Reaz, M. B. I., & Ali, M. A. M. (2012). A review of smart homes—Past, present, and future. IEEE transactions on systems, man, and cybernetics, part C (applications and reviews), 42(6), 1190-1203.
6. Alfonso, I., Garcés, K., Castro, H., & Cabot, J. (2021, October 2021). Modeling self-adaptative IoT architectures. 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C) (pp. 761-766). IEEE, Fukuoka, Japan.
7. Aliane, L., & Adda, M. (2019). HoBAC: Toward a higher-order attribute-based access control model. Procedia Computer Science, 155, 303-310.
8. Ashton, K. (2009). That 'internet of things' thing. RFID journal, 22(7), 97-114.
9. Ashutosh, A., Gerl, A., Wagner, S., Brunie, L., & Kosch, H. (2023). XACML for Mobility (XACML4M)—An Access Control Framework for Connected Vehicles. Sensors, 23(4), 1763.
10. Atlam, H. F., Alassafi, M. O., Alenezi, A., Walters, R. J., & Wills, G. B. (2018). XACML for Building Access Control Policies in Internet of Things. IoTBDS (pp. 253-260).
11. Ausanka-Crues, R. (2001). Methods for access control: advances and limitations. Harvey Mudd College, 301, 20.
12. Bell, D. E., & LaPadula, L. J. (1989). Secure computer systems: Mathematical foundations. National Technical Information Service.
13. Bertino, E., Bonatti, P. A., & Ferrari, E. (2000). TRBAC: A temporal role-based access control model. Proceedings of the fifth ACM workshop on Role-based access control (pp. 21-30).
14. Brossard, D. (2014). JSON Profile of XACML 3.0 Version 1.0. XACML Committee Specification, 1(11).
15. Brossard, D. (2023, 3 octobre). Ten years of ALFA. Consulté le 25 novembre 2024.

16. Chae, S.-h., Kim, W., & Kim, D.-K. (2006). uT-RBAC: Ubiquitous role-based access control model. IEICE transactions on fundamentals of electronics, communications and computer sciences, 89(1), 238-239.
17. Chan, M., Campo, E., & Estève, D. (2005). Assessment of activity of elderly people using a home monitoring system. International Journal of Rehabilitation Research, 28(1), 69-76.
18. Chan, M., Estève, D., Escriba, C., & Campo, E. (2008). A review of smart homes—Present state and future challenges. Computer methods and programs in biomedicine, 91(1), 55-81.
19. Chitnis, S., Deshpande, N., & Shaligram, A. (2016). An investigative study for smart home security: Issues, challenges and countermeasures. Wireless Sensor Network, 8(4), 61-68.
20. Cvitić, I., Peraković, D., Periša, M., Krstić, M., & Gupta, B. (2021). Analysis of IoT concept applications: Smart home perspective. International Conference on Future Access Enablers of Ubiquitous and Intelligent Infrastructures (pp. 167-180).
21. Damiani, M. L., Bertino, E., Catania, B., & Perlasca, P. (2007). GEO-RBAC: a spatially aware RBAC. ACM Transactions on Information and System Security (TISSEC), 10(1), 2-es.
22. De la Rosa Algarín, A., Ziminski, T. B., Demurjian, S. A., Rivera Sánchez, Y. K., & Kuykendall, R. (2014). Generating XACML enforcement policies for role-based access control of XML documents. Web Information Systems and Technologies: 9th International Conference, WEBIST 2013, Aachen, Germany, May 8-10, 2013, Revised Selected Papers 9 (pp. 21-36).
23. Déry, G. (2021). Le niveau de maturité des organisations québécoises à l'égard de l'intelligence artificielle: les déterminants de l'adoption de l'IA au Québec: le cas des communicateurs québécois [Maitrise, Université Laval]. Québec.
24. Diallo, M. M., & Adda, M. (2024). HoBACDSL: HoBAC-focused Access Control Domain Specific Language. Procedia Computer Science, 241, 40-47.
25. Dmitriev, S. (2004). Language oriented programming: The next programming paradigm. JetBrains onboard, 1(2), 1-13.
26. Fatima, A., Ghazi, Y., Shibli, M. A., & Abassi, A. G. (2016). Towards Attribute-Centric Access Control: an ABAC versus RBAC argument. Security and Communication Networks, 9(16), 3152-3166.
27. Ferraiolo, D., Chandramouli, R., Kuhn, R., & Hu, V. (2016). Extensible access control markup language (XACML) and next generation access control (NGAC). Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control, (pp. 13-24).
28. Fowler, M. (2010). Domain-specific languages. Pearson Education.
29. Giambiagi, P., Nair, S. K., & Brossard, D. (2015). Abbreviated language for authorization Version 1.0. OASIS eXtensible Access Control Markup Language (XACML) TC oasis-open. org/committees/download. php/55228/alfa-for-xacml-v1. 0-wd01. doc.

30. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems, 29(7), 1645-1660.
31. Hu, V. C., Ferraiolo, D., Kuhn, R., Friedman, A. R., Lang, A. J., Cogdell, M. M., Schnitzer, A., Sandlin, K., Miller, R., & Scarfone, K. (2013). Guide to attribute based access control (abac) definition and considerations (draft). NIST special publication, 800(162), 1-54.
32. Hu, V. C., Kuhn, D. R., Ferraiolo, D. F., & Voas, J. (2015). Attribute-based access control. Computer, 48(2), 85-88.
33. Jaber, A., Sattarpanah Karganroudi, S., Meiabadi, M. S., Aminzadeh, A., Ibrahim, H., Adda, M., & Taheri, H. (2022). On Smart Geometric Non-Destructive Evaluation: Inspection Methods, Overview, and Challenges. Materials, 15(20), 7187.
34. Jetbrains. (2009). Previous MPS Releases. Jetbrains, consulté le 12 novembre 2024.
35. Jin, X., Krishnan, R., & Sandhu, R. (2012, July 11-13, 2012). A unified attribute-based access control model covering DAC, MAC and RBAC Data and Applications Security and Privacy XXVI: 26th Annual IFIP WG 11.3 Conference, DBSec 2012. Proceedings 26, Paris, France.
36. Jose, A. C., & Malekian, R. (2015). Smart home automation security: a literature review. SmartCR, 5(4), 269-285.
37. Kalaria, R., Kayes, A., Rahayu, W., Pardede, E., & Salehi Shahraki, A. (2024). Adaptive context-aware access control for IoT environments leveraging fog computing. International Journal of Information Security, 23(4), 3089-3107.
38. Kizza, J. M. (2024). Access control and authorization. In Guide to Computer Network Security (pp. 195-214). Springer International Publishing.
39. Klenk, A., Heide, T., Radier, B., Salaun, M., & Carle, G. (2009, March 2-6, 2009). Pluggable Authorization and Distributed Enforcement with pam_xacml Kommunikation in Verteilten Systemen (KiVS) 16. Fachtagung Kommunikation in Verteilten Systemen (KiVS 2009) Kassel, 2.–6. März 2009 Eine Veranstaltung der Gesellschaft für Informatik (GI) unter Beteiligung der Informationstechnischen Gesellschaft (ITG/VDE) Ausgerichtet von der Universität Kassel, Germany (pp. 253-264).
40. Kumar, S., Tiwari, P., & Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: a review. Journal of Big data, 6(1), 1-21.
41. Locke, J. (2022, juillet 08). Directrice principale du marketing, Digi International. DIGI,
42. Luo, Y., Shen, Q., & Wu, Z. (2019). Pml: An interpreter-based access control policy language for web services. arXiv preprint arXiv:1903.09756.
43. Mernik, M. (2017, January). Domain-specific languages: A systematic mapping study. International Conference on Current Trends in Theory and Practice of Informatics, (pp. 464-472).
44. Mohanty, J., Mishra, S., Patra, S., Pati, B., & Panigrahi, C. R. (2021). IoT security, challenges, and solutions: a review. Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2019, Volume 2, 493-504.

45. Mukhopadhyay, S. C., & Jayasundera, K. P. (2017). HARNESSING POWER OF IOT FOR HEALTHCARE. APCTT Tech Monitor.
46. Nagajayanthi, B. (2022). Decades of Internet of Things towards twenty-first century: A research-based introspective. Wireless Personal Communications, 123(4), 3661-3697.
47. O'Connor, A., & Loomis, R. (2010). Economic analysis of role-based access control.
48. Parducci, E., Lockhart, H., & Rissanen, E. (2010). XACML v3. 0 privacy policy profile version 1.0. Policy, 1-11.
49. Pereira, Ó. M., Semenski, V., Regateiro, D. D., & Aguiar, R. L. (2017, April). The XACML standard-addressing architectural and security aspects. International Conference on Internet of Things, Big Data and Security, SCITEPRESS, (pp. 189-197).
50. Poltronieri, I., Pedroso, A. C., Zorzo, A. F., Bernardino, M., & de Borba Campos, M. (2021). Is usability evaluation of DSL still a trending topic? Human-Computer Interaction. Theory, Methods and Tools: Thematic Area, HCI 2021, Held as Part of the 23rd HCI International Conference, HCII 2021, Virtual Event, July 24–29, 2021, Proceedings, Part I 23 (pp. 299-317). Springer International Publishing.
51. Rabinovich, M., Stern, M., & Klein, D. (2017). Abstract syntax networks for code generation and semantic parsing. arXiv preprint arXiv:1704.07535.
52. Ragothaman, K., Wang, Y., Rimal, B., & Lawrence, M. (2023). Access control for IoT: A survey of existing research, dynamic policies and future directions. Sensors, 23(4), 1805.
53. Saltzer, J. H., & Schroeder, M. D. (1975). The protection of information in computer systems. Proceedings of the IEEE, 63(9), 1278-1308.
54. Sivasakthi, D. A., Sathiyaraj, A., & Devendiran, R. (2024). HybridRobustNet: enhancing detection of hybrid attacks in IoT networks through advanced learning approach. Cluster Computing, 1-15.
55. Standard, O. (2013). extensible access control markup language (xacml) version 3.0. A:(22 January 2013). URl: oasis-open. org/xacml/3.0/xacml-3.0-core-spec-os-en. html.
56. Systems, C. (2020). Configure Context-Based Access Control (CBAC). Retrieved 12 novembre 2024, from
57. Tanczer, L., Lopez-Neira, I., Parkin, S., Patel, T., & Danezis, G. (2018). Gender and IoT research report: The rise of the Internet of Things and implications for technology-facilitated abuse.
58. Tim, M. (2005). Extensible access control markup language (xacml) version 2.0. OASIS standard.
59. Touqeer, H., Zaman, S., Amin, R., Hussain, M., Al-Turjman, F., & Bilal, M. (2021). Smart home security: challenges, issues and solutions at different IoT layers. The Journal of Supercomputing, 77(12), 14053-14089.
60. Transforma Insights, e. d. s. e. (2023). Nombre d'appareils connectés à l'Internet des objets (IoT) dans le monde de 2019 à 2023, avec des prévisions de 2022 à 2030. Statista (Extrait le 17 octobre 2023),

61. Trautman, L. J., & Ormerod, P. C. (2017). Industrial cyber vulnerabilities: Lessons from Stuxnet and the Internet of Things. U. Miami L. Rev., 72, 761.
62. Ubale Swapnaja, A., Modani Dattatray, G., & Apte Sulabha, S. (2014). Analysis of dac mac rbac access control based models for security. International Journal of Computer Applications, 104(5), 6-13.
63. United Nations, D. o. E. a. S. A., Population Division. (2022). World Population Prospects 2022.
64. Vijayalakshmi, K., & Jayalakshmi, V. (2022). A study on current research and challenges in attribute-based access control model. Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2021, 17-31.
65. Vistbakka, I., Barash, M., & Troubitsyna, E. (2018). Towards creating a DSL facilitating modelling of dynamic access control in Event-B. Abstract State Machines, Alloy, B, TLA, VDM, and Z: 6th International Conference, ABZ 2018, Southampton, UK, June 5–8, 2018, Proceedings 6, (pp. 386-391). Springer International Publishing.
66. Voelter, M., Kolb, B., Szabó, T., Ratiu, D., & van Deursen, A. (2019). Lessons learned from developing mbeddr: a case study in language engineering with MPS. Software & Systems Modeling, 18, 585-630.
67. Yee, C. K., & Zolkipli, M. F. (2021). Review on confidentiality, integrity and availability in information security. Journal of ICT in Education, 8(2), 34-42.
68. Zhang, R., Liu, G., Li, S., Wei, Y., & Wang, Q. (2021). ABSAC: attribute-based access control model supporting anonymous access for smart cities. Security and Communication Networks, 2021, 1-11.