Community-OrBAC : un modèle de contrôle d'accès établi à partir des agents pour les systèmes de collaboration centrés sur la communauté

Rodrigue N'goran[a,b] kouadio-rodrigue.ngoran@imt-atlantique.fr
Yvon Kermarrec[a] yvon.kermarrec@imt-atlantique.fr
Olivier Asseu[b] olivier.asseu@esatic.edu.ci
Jean-Louis Tetchueng[c] jean-louis.tetchuengfoping@univ-rennes1.fr

[a]Lab-STICC, IMT-Atlantique, 29280 Brest, France

[b]LASTIC,INP-HB,Yamoussoukro, Côte d'Ivoire

[c]Université Rennes 1, Rennes, France

Résumé

Le besoin accru et constant de partage de ressources au sein des entreprises et entre organisations favorise la création de systèmes de collaboration pour des communautés spécifiques. Ces infrastructures distribuées et complexes se caractérisent par la nécessité de garantir l'autonomie des entités membres et la sécurité des informations sensibles de diverses natures échangées. Par conséquent, La problématique de la mise en place de mécanismes de sécurité et de contrôle des collaborations se pose. Dans cet article, nous proposons le Community-OrBAC, un modèle de contrôle d'accès établi à partir des agents pour des systèmes centrés sur la communauté. Le modèle vise à fournir des techniques de négociation de contrat de collaboration et de spécification dynamique de règles de politique de sécurité. Par ailleurs, notre approche étend la notion de contexte du modèle de contrôle d'accès OrBAC et présente une étude de cas dans un cloud communautaire.

Mots-clés : *Multi-agents, Contrôle d'accès, Confiance, Cloud Communautaire, Négociation, OrBAC*

Abstract

The increased and constant need to share resources within companies and between organizations favors the creation of collaboration systems for specific communities. These distributed and complex infrastructures are characterized by the need to guarantee the autonomy of member entities and the security of sensitive information of various kinds exchanged. Therefore, the problem of setting up security and control mechanisms for collaborations arises. In this paper, we propose Community-OrBAC, an agent-based access control model for community-centric systems. The model aims at providing techniques for collaborative contract negotiation and dynamic specification of security policy rules. Furthermore, our approach extends the notion of context of the OrBAC access control model and presents a case study in a community cloud.

Keywords: *Multi-agents, Access control, Trust, Community Cloud, Negotiation, OrBAC*

1 Introduction

L'émergence des technologies de l'information et de la communication favorise de plus en plus le partage de ressources entre utilisateurs et entre organisations. Des systèmes de collaboration destinés à des communautés d'utilisateurs spécifiques peuvent ainsi être créés dans le but de produire des données, partager des ressources pour satisfaire leurs besoins respectifs. La diversité des organisations et des ressources, la sensibilité des données échangées font émerger des challenges en matière de sécurité, de confidentialité des données et d'autonomie des organisations. Il est donc primordial de mettre en place des mécanismes de protection des ressources partagées et de garantir la confiance au sein de ces communautés. Les modèles de contrôle d'accès se présentent comme l'un des moyens les plus utilisés dans la sécurisation des infrastructures informatiques. Plusieurs études ont été effectuées pour proposer des modèles de contrôle d'accès qualifiés de modèles classiques[3]. Par ailleurs, la nature dynamique et complexe des environnements de collaboration a favorisé des propositions de techniques de contrôle d'accès adaptés à ces systèmes[27]. Cependant, les systèmes collaboratifs construits autour de commu-

nauté d'organisations présentent certaines singularités. Ces communautés donnent une priorité aux besoins, aux intérêts et à l'autonomie de leurs membres, favorisent des relations sociales entre eux [24]. Par conséquent, le respect de ces exigences spécifiques dans la définition des politiques de sécurité pour ce type d'environnement est crucial.

Nous proposons dans cet article, le *Community-OrBAC*, un modèle de contrôle d'accès reposant sur un système multi-agents pour des environnements collaboratifs centrés sur la communauté. Ce modèle intègre une technique d'évaluation de la confiance entre les entités et un mécanisme de négociation et de création de contrats intelligents et dynamiques de collaborations. Le reste du document est organisé comme suit : la section 2 présente les travaux connexes. La section 3 décrit notre modèle *Community-OrBAC*. La section 4 présente un cas d'étude d'application et d'évaluation de notre modèle dans un environnement de cloud communautaire. Enfin, la section 5 conclut le document et présente des perspectives pour la suite de nos travaux.

2 Travaux connexes

Les modèles de contrôle d'accès ont longtemps fait l'objet d'une attention particulière de la part des chercheurs[3][27]. Parmi les propositions, le modèle *OrBAC* (Organization Based Access Control)[19] établi à partir des préceptes du *RBAC* (Role-Based Access Control Models)[25] présente des avancées significatives dans la définition de politiques de sécurité orientées organisation. La particularité de ce modèle est qu'il permet d'exprimer des politiques indépendamment de leurs mises en œuvre. Par ailleurs, *OrBAC* permet de formaliser des règles de sécurité en considérant le contexte et apporte de la souplesse dans l'administration des politiques[19]. Toutefois, le modèle *OrBAC* présente des limites dans la spécification de politiques de contrôle pour des systèmes de collaboration entre organisations autonomes. Dans la section suivante, nous présentons des travaux effectués en vue d'adapter *OrBAC* aux systèmes collaboratifs.

2.1 Modèles de contrôle d'accès pour les systèmes collaboratifs

Les systèmes de collaboration permettent à des entités (utilisateurs ou organisations) de collaborer par le partage de données et de services. Frederic Cuppens *et al.* ont présenté dans [11], le modèle *O2O* permettant de gérer l'interopérabilité dans une collaboration entre des entités ayant défini leurs propres politiques de sécurité. Dans [18], les auteurs ont proposé une extension du modèle *OrBAC* à travers le concept de rôle dans l'organisation (*RiO*). Le modèle appelé *Multi-OrBAC* permet de spécifier des politiques de sécurité propre à chaque organisation et des règles pour gérer les interactions. Le *PolyORBAC*, introduit dans [12], est une approche qui utilise le modèle *OrBAC* pour définir la politique de sécurité au sein de chaque organisation d'une part et d'autre part, la technologie des services Web pour faciliter la collaboration et l'interopérabilité entre les organisations.

Les modèles présentés ci-dessus proposent des techniques pour résoudre la question de l'autonomie des organisations dans la définition des règles de contrôle d'accès aux ressources lors d'une collaboration. Cependant, la problématique de la confiance entre ces entités autonomes pour l'établissement de relations durables demeure et doit être abordée.

2.2 Modèles de contrôle d'accès et gestion de la confiance

La confiance entre les organisations est un facteur important pour inciter à la collaboration et garantir la sécurité des ressources partagées. Dans [29], les auteurs ont proposé le modèle *TRUST-OrBAC*. Ce modèle étend *OrBAC* avec la notion de confiance. Ils définissent des vecteurs de confiance permettant d'attribuer des rôles dynamiques aux utilisateurs, et ainsi définir des règles de sécurité pour des environnements multi-organisationnels. Le *Multi-Trust_OrBAC*, a été présenté dans [4]. Ce modèle introduit un tiers de confiance, le *TTP* (Third Trust Party), pour garantir la confiance entre les utilisateurs de différentes organisations dans le cloud. Dans [1], les auteurs ont proposé le modèle *Tr-OrBAC*. Il permet l'évaluation de la confiance entre les organisations sur la base de la logique floue. Les organisations prennent la décision de collaborer ou non en évaluant leurs pairs sur la base d'une valeur de confiance calculée.

En plus de la nécessité de garantir la confiance entre les acteurs, il est important d'apporter plus de flexibilité et de réduire les conflits de règle dans la définition des politiques de sécurité pour ces systèmes distribués de plus en plus complexes, hétérogènes et dynamiques.

2.3 Collaboration et Multi-Agents

La modélisation des interactions complexes entre diverses entités dans les environnements distribués de collaboration est un réel défi. Plusieurs axes de résolution ont été proposés sur la base des systèmes Multi-Agents (MAS)[13]. Un agent est une entité capable de s'adapter, de prendre des décisions et d'exécuter des actions complexes de manière autonome et intelligente. Il est également capable de négocier et de coopérer avec d'autres agents [28]. L'intégration des agents dans les systèmes collaboratifs permet de déployer des infrastructures composées d'entités proactives, autonomes et des mécanismes de partage flexible et dynamique [5]. Dans [2], les auteurs ont présenté un modèle de contrôle d'accès aux ressources partagées dans une coalition dynamique. Ce modèle garantit plus de flexibilité dans la gestion des départs ou l'intégration de nouveaux acteurs dans une coalition. Idrissi *et al.* ont proposé, dans [15], un modèle de contrôle d'accès établi à partir des agents mobiles et des principes du *RBAC*. Le modèle utilise les caractéristiques de mobilité et d'autonomie des agents mobiles pour combler les limites de communication. Dans [5], les auteurs ont proposé *MA-MOrBAC*, un modèle qui étend le *Multi-OrBAC* grâce à des agents mobiles pour des environnements collaboratifs distribués. Une architecture composée d'agents mobiles permet au modèle d'apporter des améliorations en termes de flexibilité et de robustesse. Un modèle de gestion dynamique de politiques de sécurité à partir d'agents a été proposé dans [22]. Dans cette approche, les agents sont utilisés pour la négociation et l'établissement de contrat intelligent pour le contrôle des accès aux données et de la protection de la vie privée. Un contrat intelligent permet de formaliser un accord applicable automatiquement. Cet accord porte sur les conditions de fourniture d'un service, sur la qualité attendue et les modifications en cas de violations des termes établis[26]. Ce type de contrat peut être déployé au travers d'algorithme auto exécutable utilisant les systèmes multi-agents et la technologie blockchain[6].

2.4 OrBAC

OrBAC est un modèle de contrôle d'accès dérivé du *RBAC*. Ce modèle a la particularité de permettre la définition de politique de contrôle d'accès en deux niveaux : un niveau constitué d'entités abstraites (Rôle, Vue, Activité) et un niveau d'entités concrètes (Sujet, Objet, Action)[19]. Les entités concrètes exécutent des actions sur des objets en fonction des règles de la politique de sécurité. À chaque entité abstraite est associée une entité concrète. Un rôle est alors une abstraction d'un groupe d'utilisateurs, une vue représente un ou des objets et une activité fait référence à une ou plusieurs actions. Le modèle *OrBAC* considère le contexte qui permet de modéliser les circonstances dans lesquelles les sujets sont autorisés à réaliser des actions sur des objets[19]. Ce modèle permet de spécifier les relations ci-dessous entre les entités de l'organisation :

- $Permission(org, r, v, a, c)$: l'organisation org autorise le rôle r à effectuer l'activité a sur la vue v dans un contexte c ;
- $Habilite(org, s, r)$: l'organisation org habilite un sujet s dans un rôle r ;
- $Utilise(org, o, v)$: l'organisation org utilise l'objet o dans la vue v ;
- $Considère(org, \alpha, a)$: l'organisation org considère l'action α comme faisant partie de l'activité a ;
- $Définit(org, s, \alpha, o, c)$: l'organisation org autorise l'action α du sujet s sur l'objet o si le contexte c est vrai.

TABLE 1 – Spécification d'une permission avec OrBAC

$org \in Organisations, s \in Sujets, \alpha \in Actions, o \in Objets, a \in Activités, v \in Vues, c \in Contextes,$
Permission(org, r, v, a, c) $\wedge$
Habilite(org, s, r) $\wedge$
Utilise(org, o, v) $\wedge$
Considère(org, α, a)$\wedge$
Définit(org, s, α, o, c)$\wedge$
$\rightarrow$ *Est_Permis*(s, α, o)

OrBAC permet de formaliser des obligations, des permissions, des interdictions et des recommandations. Une occurrence d'une permission d'un sujet autorisé à effectuer une action sur un objet est présentée dans le tableau 1 ci-dessus. Cette règle signifie que si dans l'organisation org, le rôle r est autorisé à effectuer l'activité a sur la vue v quand le contexte c est vrai, et si le rôle r est assigné au sujet s, l'action α fait partie de l'activité a, l'objet o fait partie de la vue v, le contexte c est vrai pour les entités (org, s, α, o), alors le sujet s est autorisé à réaliser l'action α sur l'objet o.

2.5 Discussion

Les systèmes collaboratifs centrés sur la communauté sont composés d'entités réunies dans le but

de collaborer et de partager des ressources. Ces communautés présentent des spécificités particulières en plus de celle des systèmes collaboratifs classiques. En effet, ces systèmes sont définis par des relations transactionnelles, durables et évolutives. Ces relations représentent des connexions sociales contextuelles entre les entités. Par ailleurs, les systèmes axés sur la communauté sont hétérogènes et fondamentalement orientés sur les besoins et intérêts des membres de la communauté [24]. Il ressort de la littérature ci-dessus, que les modèles proposés, traitent soit de l'autonomie des entités collaboratrices dans la spécification des règles de sécurité, soit de la confiance dans les systèmes de contrôle d'accès ou de négociation et d'établissement de contrat. Par conséquent, aucun modèle présenté ne couvre toutes les exigences évoquées ci-dessus d'un système de collaboration centré sur la communauté. Il convient donc de proposer, un système de contrôle d'accès couvrant largement les caractéristiques fondamentales de cette catégorie de système collaboratif. Ce modèle pourra permettre, la définition dynamique et autonome des politiques de contrôle d'accès aux ressources, la gestion de la confiance entre les entités et la formalisation de l'engagement mutuel de chaque entité dans les processus de collaboration. La section suivante décrit notre proposition pour atteindre cet objectif.

3 Modèle proposé

3.1 Contexte dans les systèmes collaboratifs centrés sur la communauté

Une collaboration entre des entités dépend de différents paramètres contextuels liés à ces entités ou aux ressources partagées. Il est essentiel de considérer ces conditions dans la définition des règles de politiques de sécurité des entités engagées. Nous étendons le contexte dans *OrBAC* décrit dans [10] avec deux nouveaux concepts : le contexte de sécurité et le contexte social comme représenté dans la figure 1.

- **Le contexte de sécurité :**

  Le contexte de sécurité est un ensemble d'informations contextuelles permettant de caractériser le niveau de sécurité d'une entité (utilisateurs, ressources) et adapter son comportement en fonction de celui-ci. Ces informations contextuelles peuvent être de divers types, notamment les valeurs de niveau de sécurité, la robustesse des protocoles et des mécanismes de sécurité[17]. Le contexte

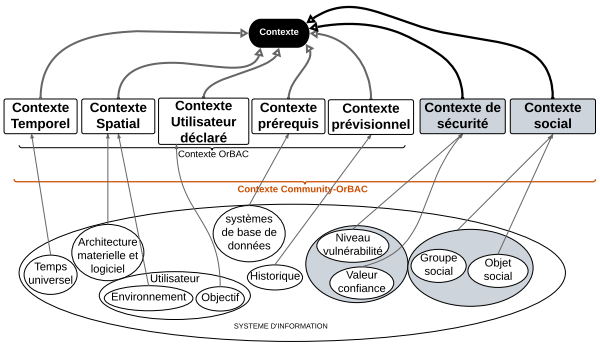

FIGURE 1 – Paramètres contextuels Community-OrBAC

de sécurité permet de spécifier qu'une action donnée d'un sujet sur un objet n'est autorisée qu'en fonction du niveau de vulnérabilité de cet objet et de la confiance accordée à ce sujet. Ainsi, la validation d'un accès requiert l'identification des niveaux de vulnérabilité des ressources et l'évaluation de la confiance entre les organisations engagées dans les interactions. À chaque niveau de vulnérabilité sera associé un seuil de valeur de confiance. Nous désignons le contexte de sécurité par $Contexte\_sécurité$ et ses deux composantes : $Niveau\_vulnérabilité$ et $Valeur\_confiance$. Les conditions requises pour un contexte donné sont exprimées formellement par des règles logiques et une algèbre de contexte dans des cas de contextes composites [10]. Le contexte de sécurité est formulé dans le tableau 2 ci-dessous et traduit que dans l'organisation org, un sujet s est autorisé à effectuer une action α sur un objet o donné si $Contexte\_sécurité$ composé de $Niveau\_vulnérabilité$ de l'objet et $Valeur\_confiance$ du sujet est vrai.

TABLE 2 – Règle de contexte de sécurité

$org \in Organisations, s \in Sujets, \alpha \in Actions, o \in Objets,$ *Définit*$(org, s, \alpha, o,$ *Niveau_vulnérabilité* $\&$ *Valeur_confiance*$)$ $\rightarrow$ *Définit*$(org, s, \alpha, o,$ *Contexte_sécurité*$)$

- **Le contexte social :**

  Une communauté requiert la mise en place de stratégies et d'actions communes au profit de ses membres. Par conséquent, nous décrivons le contexte social comme l'intérêt d'une action, d'un sujet ou d'une ressource pour la communauté. Cette perception permet de scinder la communauté en différents groupes

sociaux comme dans [17]. Le contexte social permet d'exprimer une règle qui accorde un privilège pour une action d'un sujet sur un objet en fonction de la portée communautaire de l'objet et du centre d'intérêt du groupe social auquel appartient le sujet. Nous exprimons le contexte social $Contexte\_social$ au travers de deux attributs : $Objet\_social$ et $Groupe\_social$. Le tableau 3 présente la définition d'une règle de contexte social. Elle signifie que dans l'organisation org, un sujet s peut exécuter une action α sur un objet o donné si le contexte social $Contexte\_social$ est vrai. Le contexte social est constitué de la portée sociale de l'objet $Objet\_social$ et du centre d'intérêt du groupe social du sujet $Group\_social$.

TABLE 3 – Règle de contexte social

$org \in Organisations, s \in Sujets, \alpha \in Actions, o \in Objets,$
Définit (org, s, $\alpha, o,$ ***Objet_social*** & ***Group_social***)
$\rightarrow$ ***Définit***$(org, s, \alpha, o,$ ***Contexte_social***$)$

3.2 Fonctionnement Community-OrBAC

La collaboration entre les membres de la communauté doit garantir leur autonomie et un accès régulé aux ressources partagées. Pour atteindre cet objectif, nous proposons le *Community-OrBAC*, un modèle de contrôle d'accès utilisant des agents autonomes. Notre modèle vise à fournir des moyens pour définir des politiques de sécurité fiables, dynamiques, tenant compte des paramètres contextuels décrits dans la figure 1 ci-dessus. La collaboration entre entités autonomes peut être source de conflit ou de renoncement partiel ou total à leur autonomie[30]. Nous apportons une réponse à cette problématique d'autonomie des organisations grâce aux systèmes multi-agents. En effet, les agents autonomes sont capables de s'engager dans divers types d'interactions sociales et de coopération avec d'autres agents. Le modèle de résolution de problèmes de coopération proposé dans [30] par M. Wooldridge constitue le fondement des phases du processus de collaboration entre entités de notre système. Ces différentes étapes sont : l'expression du besoin, l'engagement collectif, la négociation d'un contrat intelligent de coopération et l'action collective. Les entités ou sujets intervenants dans une collaboration sont représentés par des agents.

- **L'expression du besoin :**

  L'atteinte d'un objectif est conditionnée par l'identification des différentes actions à mener. Une entité s'engage dans une collaboration si elle reconnait le besoin de collaboration et croit en l'existence d'une entité ou groupe d'entités pouvant lui permettre d'atteindre son but[30]. Par conséquent, le processus de collaboration démarre avec l'expression explicite du besoin par l'agent demandeur de la ressource. Toutefois, une demande ne sera acceptée que si la ressource sollicitée est disponible à la période désirée et son accès est autorisé par la politique de sécurité de l'entité propriétaire. Nous pouvons alors introduire une nouvelle relation $Disponible$ entre les entités $organisation$ et $objet$ dans un contexte temporel. Cette relation sera associée aux relations $Demande(org, s, \alpha, o)$ et $Est\_Accepté(s, \alpha, o)$ exposées dans [10]. L'expression explicite d'un besoin présente plusieurs avantages. Elle constitue un moyen de suivi du respect des engagements et de gestion des conflits. Le tableau 4 ci-dessous présente la formalisation d'une règle d'acceptation d'une action d'un sujet sur un objet.

TABLE 4 – Règle d'acceptation d'une action sur un objet

$org \in Organisations, s \in Sujets, \alpha \in Actions, o \in Objets, contexte\_temporel \in Contextes,$
Demande(org, s, α, o)$\wedge$
Disponible(org, o, contexte_temporel) $\wedge$
Est_Permis(s, α, o)$\wedge$
$\rightarrow$ ***Est_Accepté***(s, α, o)

- **L'engagement collectif :**

  Cette étape consiste à trouver une entité partenaire capable de contribuer totalement ou partiellement à la réalisation de l'objectif visé. Ce processus commence par l'identification des sujets potentiels pour une collaboration donnée, ensuite l'évaluation de la confiance des entités et enfin la sélection du partenaire idéale.

  - **L'identification des potentiels partenaires :**

    Les membres de la communauté exposent dans le registre des objets de la communauté les ressources dont ils disposent. Les objets sont déclarés sur la base de règles de

contrôle d'accès établis dans les politiques locales de chaque entité. La prise en compte des contextes social et de sécurité est obligatoire dans le choix du partenaire. Ainsi, une matrice de gouvernance des objets doit définir les exigences de ces contextes. Cette matrice met en exergue le niveau de confiance requis pour un niveau de vulnérabilité d'un objet donné d'une part et d'autre part, le groupe social du sujet exigé pour une ressource de portée sociale donnée.

- **L'évaluation de la confiance :**

  La confiance entre les entités est un élément clé favorisant le partage et la collaboration. Cette confiance sera déterminée à partir d'un modèle d'évaluation de la confiance. Ce modèle a été exposé dans une précédente étude présentée dans[23]. Nous évaluons la valeur de confiance de chaque entité relativement à une action donnée sur un objet sur la base d'interactions directes ou recommandées et de la réputation spécifique. Cette valeur de confiance est exprimée comme ci-dessous :

  $$\omega_{O_j}^{O_i} = \beta DRT_{O_j}^{O_i} + (1-\beta) sr_{O_j} \qquad (1)$$

  $DRT_{O_j}^{O_i}$ est l'opinion de confiance sur la base d'interactions directes ou recommandées entre le propriétaire O_j d'une ressource et le demandeur O_i. Cette valeur est calculée grâce à la logique subjective [16]. sr_{O_j} est la réputation spécifique du propriétaire de l'objet et β le poids de la confiance directe ou recommandée représentant l'influence des interactions antérieures dans la détermination de la valeur la confiance.

- **La sélection du partenaire de collaboration :**

  L'entité idéale pour la collaboration est choisie sur la base de la valeur de confiance calculée, de l'algorithme de sélection présenté dans [23] et des exigences définies dans les règles de la politique de sécurité d'accès. La sélection d'une entité ayant exposé volontairement sa capacité à collaborer en vue de contribuer à satisfaire un besoin formellement exprimé par une entité paire constitue l'engagement collectif des deux sujets[30].

- **Négociation et création d'un contrat dynamique de collaboration :**

  Lors de cette étape, les agents engagés doivent être d'accord sur les différentes actions à mener dans le but d'atteindre l'objectif de la collaboration. En effet, l'exigence d'autonomie des organisations et l'éventualité de conflits dans les actions nécessitent de parvenir à un accord entre les entités sur la conduite à tenir [30]. Une négociation pourra permettre d'aboutir à cet accord.
  La négociation se traduit généralement par des propositions, des contre-propositions et des suggestions pour aboutir à un consensus sur le résultat final. Les actions à effectuer et le cadre de suivi de cette collaboration seront consolidés dans un contrat dynamique et intelligent. Ce processus de négociation est crucial et peut s'avérer complexe. D'où l'utilisation des agents intelligents, autonomes, capables de rendre dynamique la définition, le suivi des politiques et d'éviter ou réduire l'intervention humaine [30]. La mise en place du contrat sera effectuée par des agents représentant les organisations et charger de négocier, de définir et d'actualiser éventuellement les termes de cet accord.
  Plusieurs travaux ont mis en exergue la négociation, la définition et la gestion de contrat électronique dans les collaborations entre entités d'organisations distribuées [26][22]. Nous proposons une approche fondée sur le *Web Services Agreement* (*WS-Agreement*)[14], un standard de spécification des accords de niveaux de service intégrant un protocole de négociation et de renégociation de contrat. Ce protocole composé de trois couches (négociation, accord et service) est adapté à différents types d'environnements dont les infrastructures utilisant les systèmes multi-agents [22][7]. Par ailleurs, le *WS-Agreement* a été associé à des règles de sécurité *OrBAC* afin de permettre à des utilisateurs et des fournisseurs de services cloud de négocier, créer et surveiller des accords de niveau de service (*SLA*) dans [20].
  Pour illustrer le fonctionnement de ce protocole, considérons un agent $orgA$ demandeur d'une ressource et un agent $orgB$ fournisseur. La négociation commence par l'envoi d'une requête de demande d'une ressource exposée par l'agent $orgB$ dans le registre des objets. À la réception de la requête, $orgB$ répond en transmettant la structure de base d'un contrat *WS-Agreement*. Sur la base de ce modèle

et de ses besoins, $orgA$ construit une offre puis l'envoie à $orgB$. Cette offre peut être validée directement ou faire l'objet d'une contre-proposition en cas de non-conformité avec les contraintes prédéfinies par $orgB$. Cette opération est répétée jusqu'à ce qu'un accord soit trouvé ou non. Une fois l'offre validée, l'agent $orgB$ crée un contrat qu'il signe et le soumet à $orgA$. Ce dernier à son tour marque son approbation en signant l'accord et le partage avec l'agent $orgB$.

- **L'action collective :**

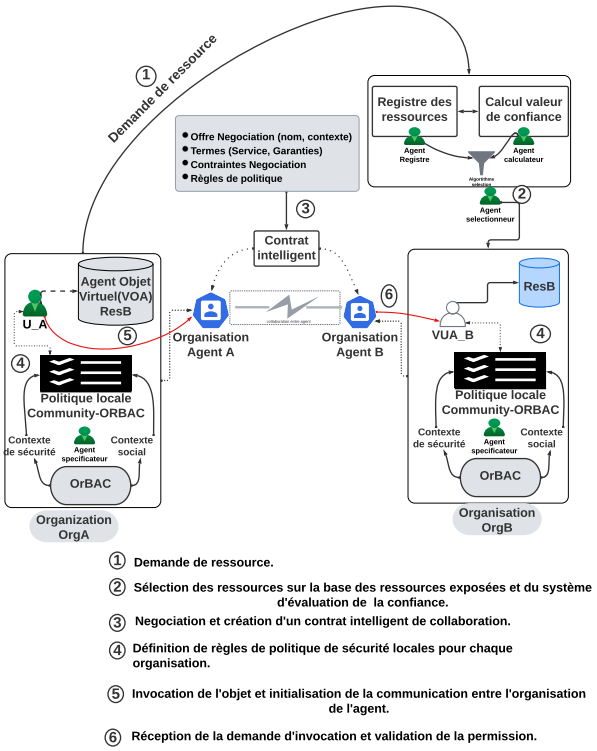

FIGURE 2 – Architecture Community-OrBAC

Les négociations effectuées, les entités ont donc un accord sur l'action collective à réaliser et un contrat pour suivre le processus en vue de l'atteinte de l'objectif. En partant du scenario présenté lors de la phase de négociation, le but de l'action collective est de permettre à un utilisateur u_A de l'organisation $orgA$ d'accéder à une ressource Res_B de l'organisation $orgB$. Afin de préserver l'autonomie des entités dans la définition des règles de sécurité, l'approche adoptée est la procédure d'invocation et de partage de services présentée dans [12] en introduisant les notions d'objet agent virtuel (VOA) et de sujet agent virtuel(VUA). Ces éléments représentent respectivement l'objet distant invoqué et le sujet dans l'organisation fournisseur de l'objet.

De ce fait, un objet VOA_Res_B est créé dans la politique locale de $orgA$ et lié à une action $invoquer$. Une permission autorisant le rôle associé au sujet u_A d'exécuter l'activité correspondant à l'action $invoquer$ sur la vue représentant l'objet VOA_Res_B est également définie dans la politique locale de $orgA$. Cette règle est exprimée dans le tableau 5 ci-dessous.

TABLE 5 – Permission au niveau de l'organisation demandeur

$orgA \in Organisations, u_A \in Sujets, invoquer \in Actions, VOA\_Res_B \in Objets, consulter \in Activités, c \in Contexte,$
Permission$(orgA, r, consulter, vue\_VOA\_Res_B, c) \wedge$
Habilite$(orgA, u_A, r) \wedge$
Utilise$(orgA, VOA\_Res_B, vue\_VOA\_Res_B) \wedge$
Considère$(orgA, invoquer, consulter) \wedge$
Définit$(orgA, u_A, invoquer, VOA\_Res_B, c) \wedge$
$\rightarrow$ ***Est_Permis***$(u_A, invoquer, VOA\_Res_B)$

Par ailleurs, dans la politique locale de $OrgB$, un utilisateur virtuel vua_B est créé et associé à un rôle disposant d'une permission permettant d'exécuter une activité sur la vue de l'objet Res_B. L'action $invoquer$ va déclencher une communication entre l'agent associé à l'objet virtuel de $OrgA$ et celui de l'organisation $OrgB$. Cette communication se fait conformément aux dispositions du contrat établi entre les deux entités. À la réception du message de l'agent de l'organisation $OrgA$ suite à l'invocation, les actions liées au sujet vua_B sur l'objet Res_B seront exécutées comme illustré dans la figure 2.

Le tableau 6 ci-dessous présente la règle dans la politique locale de $OrgB$.

TABLE 6 – Permission au niveau de l'organisation fournisseur

$orgB \in Organisations, vua_B \in Sujets, exécuter \in Actions, Res_B \in Objets, Afficher \in Activités, c \in Contexte,$
Permission$(orgB, r, Afficher, vue\_Res_B, c) \wedge$
Habilite$(orgB, vua_B, r) \wedge$
Utilise$(orgB, Res_B, vue\_Res_B) \wedge$
Considère$(orgB, exécuter, Afficher) \wedge$
Définit$(orgB, vua_B, exécuter, Res_B, c) \wedge$
$\rightarrow$ ***Est_Permis***$(vua_B, exécuter, Res_B)$

Notons que toute action de toute entité de la collaboration est conditionnée par une authentification. Cependant, le processus d'authentification des entités est hors du cadre de notre étude.

4 Étude de cas

Cette section décrit une étude de cas de l'application de **Community-OrBAC** dans un cloud communautaire.

Un Cloud Communautaire (*3C*) permet de regrouper au sein d'une communauté des organisations ayant des exigences (sécuritaire, juridique, etc) et des besoins communs. Le *3C* vise à favoriser la coopération entre les membres, améliorer la sécurité de l'infrastructure et réduire les coûts d'investissements [21]. Notre étude repose sur une architecture de *3C* fédéré.
Tout d'abord, nous présentons l'architecture de notre cloud communautaire. Ensuite, le scénario d'échange et de négociation d'un contrat intelligent de coopération et enfin la spécification de règles sur la base du modèle *Community-OrBAC*.

4.1 Architecture Cloud communautaire

L'architecture de notre cloud communautaire est présentée dans la figure 3 ci-dessous. Elle est composée de startups regroupées dans une communauté d'entreprise *Com_Startup*. Chaque startup représente une organisation. Ces organisations sont réunies dans l'optique de partager des ressources de type : *Software as a Service(SaaS)*, *Platform as a Service(PaaS)* et *Infrastructure as a Service(IaaS)*. Ces ressources sont référencées dans un registre de ressources. Chaque organisation membre est soit fournisseur et/ou demandeur de ressources.

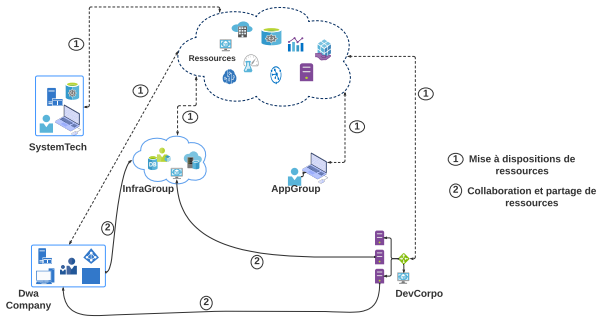

FIGURE 3 – Architecture Cloud Communautaire Com_Startup

4.2 Scénario de collaboration et spécification de règles Community-OrBAC

Nous décrivons dans cette partie un partage de ressources entre les organisations et la définition de règles de politiques d'accès sur la base du modèle *Community-OrBAC*. Considérons que la startup $DevCorpo$ sollicite un cluster de serveurs pour la mise en place d'infrastructures de développement d'applications métiers. La ressource correspondante à ce besoin dans le registre est le $Clusterkub$ fourni par la startup $InfraGroup$. Les deux organisations négocient et mettent en place un contrat de collaboration dynamique et évolutif durant toute la période de partage. La figure 4 ci-dessous présente le scénario de collaboration entre les agents $InfraGroup$ et $DevCorpo$ représentant respectivement le fournisseur et le demandeur.

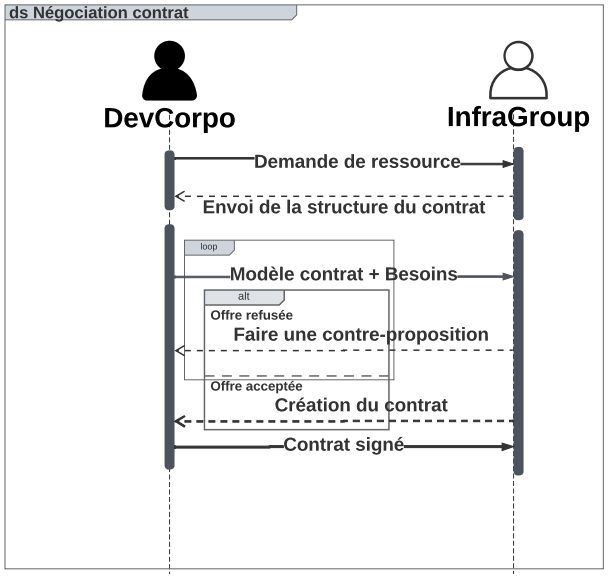

FIGURE 4 – Négociation de contrat de coopération entre deux agents

Le contrat de collaboration établi, nous présentons ci-dessous les règles *Community-OrBAC* pour le partage de la ressource $Clusterkub$. Comme présenté dans le fonctionnement du *Community-OrBAC*(section 3.2), l'accès à une ressource distante nécessite la création de l'agent objet virtuel $VOA\_clusterkub$, représentant la ressource désirée $clusterkub$ dans l'organisation demandeur, et de l'agent utilisateur virtuel vua_B dans la politique de sécurité du propriétaire de la ressource (voir figure 2).

La permission dans la politique locale $DevCorpo$ est présentée dans le tableau 7 ci-dessous. Cette permission est accordée dans un contexte c considérant le niveau de vulnérabilité et la portée sociale de la ressource $VOA\_clusterkub$.

TABLE 7 – Règle dans la politique locale du demandeur $DevCorpo$

$DevCorpo \in Com\_Startup, s \in Sujets, invoquer \in Actions, VOA\_clusterkub \in Objets, consulter \in Activités, c \in Contexte, r \in Roles$
Permission(DevCorpo, r, consulter, vue_VOA_clusterkub, c) $\wedge$ *Habilite* (DevCorpo, s, r) $\wedge$ *Utilise*(DevCorpo, VOA_clusterkub, vue_VOA_clusterkub) $\wedge$ *Considère* (DevCorpo, invoquer, consulter) $\wedge$ *Définit* (DevCorpo, s, invoquer, VOA_clusterkub, c) $\rightarrow$ *Est_Permis*$(s, invoquer, VOA\_clusterkub)$

Le tableau 8 décrit la règle d'accès dans la politique du propriétaire $InfraGroup$ dans un contexte évalué sur la base de la valeur de confiance et du groupe social du sujet vua_B.

TABLE 8 – Règle dans la politique locale du fournisseur $Infragroup$

$InfraGroup \in Com\_Startup, vua_B \in Sujets, exécuter \in Actions, Clusterkub \in Objets, Afficher \in Activités, c \in Contexte, r \in Roles,$
Permission (InfraGroup, r, Afficher, Vue_ clusterkub, c) $\wedge$ *Habilite* (InfraGroup, vua$_B, r) \wedge$ *Utilise* (InfraGroup, Clusterkub, Vue_clusterkub) $\wedge$ *Considère* (InfraGroup, exécuter, Afficher) $\wedge$ *Définit* (InfraGroup, vua$_B, excuter, Clusterkub, c)$ $\rightarrow$ *Est_Permis*$(vua_B, excuter, Clusterkub)$

L'étude de cas présente différents avantages apportés par notre modèle dans une collaboration entre deux entités de façon générale et spécifiquement entre membres d'une même communauté. En effet, l'utilisation des agents autonomes et la démarche de négociation proposée exposent des interactions de type pair à pair, consensuelles, garantissant une souplesse et une indépendance de chaque entité dans la gestion de ses utilisateurs et du contrôle des accès à ses ressources. Par ailleurs, au regard des apports significatifs apportés par la combinaison des systèmes multi-agents et des technologies blockchain [9][6], la négociation et la création de contrats dynamiques peuvent être déployées grâce à des contrats intelligents auto-exécutables. Les agents serviront de sources de données (modifications des clauses du contrat, pénalités, etc) pour des algorithmes représentant les contrats intelligents. En outre, l'identification des niveaux de sécurité des ressources partagées et l'évaluation de la confiance est fortement recommandée dans la mise en place de stratégie de cybersécurité dans le *cloud computing*[8]. De façon spécifique pour des organisations centrées sur la communauté, les membres peuvent mettre à la disposition de leurs pairs des ressources avec des exigences d'accessibilité réduites (niveau de confiance, sans contrepartie financière, etc). Cette action visant à enrichir la communauté et dans l'intérêt de ses membres peut être une prérogative à l'adhésion. Un tel scénario serait difficilement envisageable pour des systèmes non communautaires. Le contexte de sécurité et le contexte social constituent par ailleurs des critères supplémentaires et fiables pour définir des règles de sécurité robustes et personnalisées.

5 Conclusion and Perspectives

Dans cet article, nous avons présenté le *Community-OrBAC*, un modèle de contrôle d'accès aux ressources utilisant les agents pour les systèmes de collaboration centrés la communauté. Le modèle s'appuie sur des agents autonomes pour représenter les entités du système et favoriser la spécification dynamique et autonome des politiques de sécurité. Les systèmes multi-agents permettent d'exposer un processus de collaboration consensuel et évolutif à travers la négociation et la création de contrat de coopération entre les entités. Par ailleurs, le modèle étend *OrBAC* avec les notions de contexte de sécurité et contexte social et intègre un système d'évaluation de la confiance entre les entités. En outre, un cloud communautaire a servi de cadre d'application du modèle. Dans de futurs travaux, nous envisageons de proposer des mécanismes de sécurisation des échanges, d'authentification des agents et de suivi des accords négociés. Une implémentation du modèle est également en cours d'expérimentations.

Références

[1] N. A. Aali, A. Baina, and L. Echabbi. Tr-OrBAC : Towards a Trust Framework for Collaborative Sys-

tems in Critical Information Infrastructures. *Journal of Network and Innovative Computing*, 4 :106–115, 2016.

[2] B. I. Abdelkrim, A. Baina, C. Feltus, J. Aubert, M. Bellafkih, and D. Khadraoui. Coalition-OrBAC : An agent-based access control model for dynamic coalitions. *Advances in Intelligent Systems and Computing*, 745(May 2020) :1060–1070, 2018.

[3] R. Ausanka-Crues. Methods for access control : Advances and limitations. *Harvey Mudd College*, pages 1–5, 2001.

[4] M. B. Saidi and A. Marzouk. Multi-Trust_OrBAC : Access Control Model for Multi-Organizational Critical Systems Migrated To the Cloud. *International Journal of Soft Computing and Engineering*, (3) :2231–2307, 2013.

[5] Z. B. Yahya, F. B. Ktata, and K. Ghedira. MA-MOrBAC : A distributed access control model based on mobile agent for multi-organizational, collaborative and heterogeneous systems. *Lecture Notes in Computer Science*, 10694 LNCS :101–114, 2018.

[6] Ricardo Barbosa, Ricardo Santos, and Paulo Novais. Smart Contracts Based on Multi-agent Negotiation. In Fernando De La Prieta, Alia El Bolock, Dalila Durães, João Carneiro, Fernando Lopes, and Vicente Julian, editors, *Highlights in Practical Applications of Agents, Multi-Agent Systems, and Social Good. The PAAMS Collection*, pages 104–114, Cham, 2021. Springer International Publishing.

[7] D. Battr, F. M. T. Brazier, K. P Clark, M. Oey, A. Papaspyrou, W. Oliver, P. Wieder, and W. Ziegler. A Proposal for WS-Agreement Negotiation. pages 233–241, 2010.

[8] Oliver Borchert and Allen Tan. Implementing a Zero Trust. *national insutute of standard and technology (NIST)*, (March), 2020.

[9] Davide Calvaresi, Alevtina Dubovitskaya, Jean Paul Calbimonte, Kuldar Taveter, and Michael Schumacher. Multi-agent systems and blockchain : Results from a systematic literature review. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10978 LNAI(June) :110–126, 2018.

[10] F. Cuppens and N. Cuppens-Boulahia. Modeling contextual security policies. *International Journal of Information Security*, 7(4) :285–305, 2008.

[11] F. Cuppens, N. Cuppens-Boulahia, and C. Coma. O2O : Virtual private organizations to manage security policy interoperability. *Lecture Notes in Computer Science*, 4332 LNCS :101–115, 2006.

[12] Y. Deswarte and A. A. E. Kalam. PolyOrBAC : An Access Control Model for Inter-Organizational Web Services. *IGI Global*, pages 901–923, 2009.

[13] Ali Dorri, Salil S. Kanhere, and Raja Jurdak. Multi-Agent Systems : A Survey. *IEEE Access*, 6 :28573–28593, 2018.

[14] John Rofrano Ibm. Web Services Agreement Specification (WS-Agreement). pages 1–81, 2007.

[15] H. Idrissi, M. Ennahbaoui, E. M. Souidi, A. Revel, and S. Elhajji. Access control using mobile agents. *International Conference on Multimedia Computing and Systems*, pages 1216–1221, 2014.

[16] A. Jøsang, S. Pope, and R. Hayward. Trust Network Analysis with Subjective Logic. *Conferences in Research and Practice in Information Technology Series (2006)*, 48 :85–94, 2006.

[17] V. Jovanovikj, D. Gabrijelčič, and T. Klobučar. A conceptual model of security context. *International Journal of Information Security*, 13(6) :571–581, 2014.

[18] A. A. El Kalam and Y. Deswarte. Multi-OrBAC : A New Access Control Model for Distributed, Heterogeneous and Collaborative Systems. *8th IEEE International Symposium on Systems and Information Security*, 1, 2006.

[19] A. A.E. Kalam, R. E. Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miege, C. Saurel, and G. Trouessin. Organization based access control. *IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, (May 2014) :120–131, 2003.

[20] Y. Li, N. Cuppens-Boulahia, J. M. Crom, F. Cuppens, and V. Frey. Expression and enforcement of security policy for virtual resource allocation in IaaS cloud. *IFIP International Federation for Information Processing*, 471 :105–118, 2016.

[21] K. Marzantowicz and L. Paciorkowski. Community cloud : Closing the gap between public and private. *IGI Global*, pages 39–55, 2017.

[22] J. E. Mokhtari, A. A. E. Kalam, S. Benhaddou, and J. P. Leroy. Dynamic Management of Security Policies in PrivOrBAC. *International Journal of Advanced Computer Science and Applications*, 12(6) :693–701, 2021.

[23] R. N'goran, J.-L. Tetchueng, G. Pandry, Y. Kermarrec, and O. Asseu. Trust Assessment Model Based on a Zero Trust Strategy in a Community Cloud Environment. *Engineering*, 14(11) :479–496, 2022.

[24] F. Paci, A. Squicciarini, and N. Zannone. Survey on access control for community-centered collaborative systems. *ACM Computing Surveys*, 51(1), 2018.

[25] Ravi S Sandhu, Hal L Feinstein, Charles E Youman, and Edward J Coyne. Role-Based Access Control Models. 29(2) :38–47, 1996.

[26] V. Scoca, R. B. Uriarte, and R. D. Nicola. Smart Contract Negotiation in Cloud Computing. *IEEE International Conference on Cloud Computing*, 2017-June :592–599, 2017.

[27] P. G. Shynu and K. John Singh. A comprehensive survey and analysis on access control schemes in cloud environment. *Cybernetics and Information Technologies*, 16(1) :19–38, 2016.

[28] Kwang Mong Sim. Agent-based approaches for intelligent intercloud resource allocation. *IEEE Transactions on Cloud Computing*, 7(2) :442–455, 2019.

[29] K. Toumi, C. Andrés, and A. Cavalli. Trust-OrBac : A trust access control model in multi-organization environments. *Lecture Notes in Computer Science*, 7671 LNCS(August 2015) :89–103, 2012.

[30] M. Wooldridge. *Reasoning about Rational Agents*. The MIT Press, 2003.