Approche basée BRS pour la Spécification et l'Analyse d'une Architecture Sécurisée du Fog Computing

Ayoub Bouheroum*, Zakaria Benzadri**, Faiza Belala***

*ayoub.bouheroum@univ-constantine2.dz
**zakaria.benzadri@univ-constantine2.dz
***faiza.belala@univ-constantine2.dz
LIRE Laboratory, University of Constantine 2-Abdelhamid Mehri Constantine, Algeria.

Résumé. Le Fog Computing renvoie à une infrastructure matérielle et applicative distribuée offrant des services de calcul, de stockage et de mise en réseau entre les terminaux et les serveurs Cloud traditionnels. Il présente l'avantage de réduire le temps de latence des services et d'améliorer leur qualité perçue, ainsi que l'avantage de la distribution totale des données. Cependant, la segmentation, la distribution et le déploiement adaptatif de fonctionnalités sur cette série de dispositifs, allant de l'IoT au Cloud, posent de grands problèmes en raison de l'hétérogénéité, la structure hiérarchique et l'infrastructure à très grande échelle que ces taches devront exploiter. Le présent travail s'intègre dans ce nouveau paradigme du Fog Computing et vise la proposition d'un modèle formel générique spécifiant une architecture Fog, composée d'un ensemble de nœuds Fog sécurisés qui agissent à la fois comme filtres pour réduire la quantité de données envoyées au Cloud et comme unités de traitement tout prés des données collectées. Entre autre, les exigences de collaboration multi-niveaux (IoT, Fog et Cloud), engendrant une nouvelle série de problèmes de sécurité liés à la gestion des identités et la gestion de l'accès aux ressources, sont aussi considérées par le modèle formel CA-BRS (Control Agents and BRS) proposé, qui combine les agents et les BRS (Bigraphical Reactive Systems). L'exécution et l'analyse de ce modèle à travers l'outil BigraphER a permis de déduire des conclusions importantes.

1 Introduction

Le Fog computing est une expression inventée par Cisco en 2013 (Cisco, 2015) qui décrit un cadre de calcul et de réseau permettant de prendre en charge les applications IoT (Internet des objets). Bien que ce cadre ne soit pas exclusif à l'IoT, il existe d'autres applications, sensibles au temps de latence, qui peuvent exploiter cette architecture. D'autre part, le Fog computing rejoint dans ses caractéristiques l'informatique distribuée ("Distributed Computing"), une forme informatique dans laquelle les données et les applications sont réparties sur plusieurs ordinateurs ou systèmes, mais connectées et intégrées au moyen de services de réseau et de normes d'interopérabilité, de sorte qu'elles fonctionnent comme un seul environnement.

Ainsi, les systèmes dédiés aux Fog sont caractérisés de complexes en termes d'ouverture et de réaction, ils sont constitués de plusieurs types de sous systèmes qui communiquent de manière continue, lors de leur fonctionnement. Par conséquent, dans le cadre de l'Ingénierie des Systèmes, leur développement traditionnel n'est plus la solution souhaitée, de plus ils doivent satisfaire un certain nombre d'exigences sécuritaires. Un des challenges pour cette communauté du Génie Logiciel est de pouvoir intégrer de manière harmonieuse des considérations de sécurité dans les premières activités du cycle de vie d'un système dédié au Fog. En effet, si les exigences fonctionnelles d'un tel système peuvent être prises en charge et traitées de manière rationnelle, les exigences de sécurité n'ont pas généralement reçu le même type d'intérêt. Concevoir un système dédié au Fog "véritablement" sécurisé (c.-à-d. capable de se défendre contre toutes les menaces crédibles) est trop coûteux. Ainsi, en pratique, les ressources de développement limitées de ses systèmes imposent des compromis. A travers ce travail, nous soutenons fermement l'idée selon laquelle l'ingénierie de tels systèmes doit être unifiée avec l'ingénierie de la sécurité. Le manque d'interaction entre les chercheurs travaillant sur la modélisation des exigences et de la conception des systèmes dédiés au Fog (par exemple, dans la communauté UML) et celle des politiques de sécurité doit être surmonté. Notre travail rejoint ces travaux et partage les mêmes défis qui se résument principalement en :

D1 : Niveau d'abstraction assez élevé pour bien comprendre et commenter les entités d'un système Fog computing, aussi complexe.

D2 : Séparation claire et succincte entre les préoccupations en identifiant tous les niveaux de conception concernant ce type de système logiciel, en particulier le niveau sécurité, qui doit être considéré dès les premières phases de son développement.

D3 : Application des méthodes formelles à base rigoureuse pour des fins d'analyse.

L'objectif principal de cet article est de contribuer d'une part, au développement d'une architecture de référence générique, multi-vues pour surmonter les défis D1 et D2 et faciliter ainsi le traitement de certains aspects de sécurité dans les systèmes Fog, et d'autre part, à la définition d'un modèle formel permettant la spécification des exigences fonctionnelles du système Fog d'un côté, et ses exigences de sécurité de l'autre côté (s'attaquer au défi D3). Dans ce travail de recherche, nous adoptons les systèmes réactifs bigraphiques (BRS) (Milner et Jensen, 2004) et les agents pour la modélisation d'un système Fog sécurisé, du point de vue de sa structure physique en termes de bigraphes, ainsi que son fonctionnement et son contrôle en termes de règles de réaction. En outre, les agents qui constituent la structure additive de notre modèle (CA-BRS : "Control Agents and Bigraphical Reactive Systems") permettent d'observer (analyser) et de contrôler le fonctionnement sécurisé de ce type de système. L'implémentation, l'exécution et la vérification des comportements sécurisés d'un tel système sont réalisées à travers l'outil BigraphER (Sevegnani et Calder, 2016).

Dans la suite de l'article, la Section 2 décrit brièvement les concepts fondamentaux des BRS. La Section 3 présente un état de l'art relatif à notre problématique. Dans la Section 4, nous détaillons notre approche de modélisation formelle basée BRS d'une architecture sécurisée du Fog computing. Nous commençons tout d'abord par rappeler l'architecture de référence adoptée pour prendre en considérations les exigences de sécurités dans un système Fog, puis nous montrons comment transcrire tous les éléments architecturaux dans un modèle formel intégrant les bigraphes et les agents. La Section 5 est dédiée à la présentation de quelques résultats d'exécution et d'analyse model-cheking du modèle proposé. Finalement, la conclusion propose une synthèse du travail réalisé et des perspectives liées à la poursuite de ce travail.

2 Présentation brève des BRS

Les systèmes réactifs bigraphiques (BRS) sont un nouveau formalisme, développé par James J. Leifer, O. Jensen et R. Milner au laboratoire Computer de l'université de Cambridge (Milner et Jensen, 2004), qui diffère des formalismes traditionnels par son aspect graphique et sa capacité de représenter à la fois la localité et la connectivité des systèmes informatiques ubiquitaires et distribués. Les BRS aident à :

— Esquisser la structure et les schémas d'interaction d'un système composé de plus d'un composant en modélisant sa structure de localisation,
— Préciser les dépendances entre des différents composants d'un système,
— Décrire un ensemble de règles de réaction (transitions) pour modéliser le comportement d'un système concurrent, et
— Spécifier une image plus claire de propriétés du système.

Les BRS ont été utilisés pour la spécification de plusieurs types de systèmes tels que : les systèmes de téléphonie mobile (Højsgaard et Glenstrup, 2011), les services Web (WS-BPEL, HomeBPEL) (Bundgaard et al., 2008), les systèmes stochastiques (Sevegnani et Calder, 2015), les systèmes Cloud (Benzadri, 2016), les systèmes sensibles au contexte (Cherfia et al., 2014), les réseaux de communications (Boucebsi et Belala, 2015), etc. Un BRS est constitué d'un ensemble de bigraphes représentant l'état du système, et d'un ensemble de règles de réaction définissant la façon dont le système peut évoluer. Dans la suite, nous donnons la définition d'un bigraphe et celle d'une règle de réaction, pour plus de détails, le lecteur pourra consulter (Milner et Jensen, 2004), (Benzadri, 2016). Un bigraphe comme son nom l'indique est un graphe formé de deux structures mathématiques indépendantes : graphe de places et graphe de liens. Le graphe des places est une forêt (ensemble des arbres) avec les nœuds V, représentant la hiérarchie des nœuds (inclusion de certains nœuds dans d'autres) et des interfaces sortantes et entrantes. Un hypergraphe avec les mêmes nœuds V et des arêtes E peut constituer le graphe des liens d'un bigraphe, incluant aussi des interfaces. Les interfaces sortantes et entrantes du graphe des places sont respectivement ses racines (dites aussi régions) et ses sites, ils sont disjoints de ses nœuds. Les racines peuvent être les parents des nœuds et sites mais il n'y a pas le parent pour eux ; les sites peuvent être les fils des racines et nœuds mais il n'y a pas le fils pour eux. Les interfaces sortantes et entrantes du graphe des liens sont des ensembles de noms sortants et entrants. Ils précisent les liens potentiels avec d'autres bigraphes.

La Figure 1, présente un exemple d'un bigraphe avec $V = \{v0, v1, v2, v3, v4, v5\}, E = \{e0, e1, e2\}$. Notons la hiérarchie des nœuds, par exemple le nœud $v0$ qui contient $v1$ et $v2$, et de même le nœud $v2$ qui contient $v3$, etc. Pour le graphe des places, on choisit l'interface sortante $2 = \{0, 1\}$ fournissant les racines (régions) distinctes comme les parents des nœuds $v0$, $v4$ et le site . D'un autre côté, l'interface entrante 1, indique qu'il possède un seul site 0 dont le parent est la région . On écrit le graphe des places comme :
$B^P : 1 \rightarrow 2$;
Pour le graphe de liens, on choisit l'interface sortante $\{x\}$ qui est liée aux parties des hyperliens « ouverts », et l'interface entrante $\emptyset$, désignant l'absence d'interface entrante. On écrit le graphe des liens comme : $B^L : \emptyset \rightarrow \{x\}$.
Le bigraphe est ainsi une paire formée du graphe des places et du graphe des liens, notée : $B = (B^P, B^L) : (2, \{x\}) \rightarrow (1, \emptyset)$. Ses interfaces sortante et entrante sont respectivement les interfaces sortantes et entrantes de B^P et B^L.

Les points de liaison entre les nœuds et les arcs sont appelés ports. Un contrôle (type) est attaché à chaque nœud du bigraphe ; il indique le nombre de ports qu'un nœud peut avoir et permet de déterminer s'il est atomique (nœud vide), actif (nœud permettant l'application de règles de réaction à l'intérieur) ou bien passif. La signature d'un bigraphe prend la forme (K, ar) ; avec K étant l'ensemble des contrôles sur les nœuds (types), et $ar : K \rightarrow N$, est une fonction attribuant un nombre de ports (arité) à chaque type de nœud. Les nœuds du bigraphe peuvent appartenir à des types différents. Par exemple, une signature K définie pour le bigraphe B (Figure 1) est donnée par $\Sigma = (K, ar) = \{Ctrl_{v0} : 1, Ctrl_{v1} : 2, Ctrl_{v2} : 0, Ctrl_{v3} : 2, Ctrl_{v4} : 2, Ctrl_{v5} : 1\}$, où $Ctrl_{vi}, i = 1$ à 5 sont les contrôles des nœuds ; et 1, 2, 0, 2, 2, 1 sont respectivement leur cardinalité.

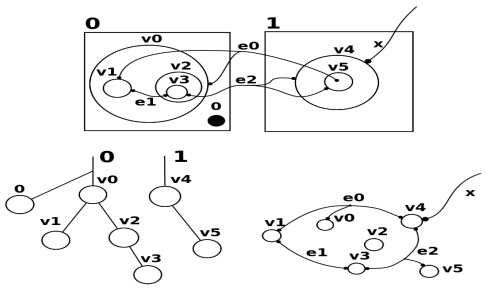


FIG. 1 – *Exemple d'un bigraphe B et les graphes de places et liens correspondants*

L'évolution d'un système exprimée au moyen de bigraphes est représentée par des règles de réaction. Une règle de réaction est une paire de bigraphes, *Reactum* et *Redex* où le Reactum est le bigraphe modélisant l'état courant du système et le Redex est celui qui modélise l'état suivant du système après l'exécution de la règle. Par exemple, considérons la règle de réaction de la Figure 2, elle permet au nœud v1 de se déplacer de la région 0 à la région 1 et qui reste connecté par le lien e1 au nœud v3. Le lien e0 reliant v1 à v0 et v4 est détruit.

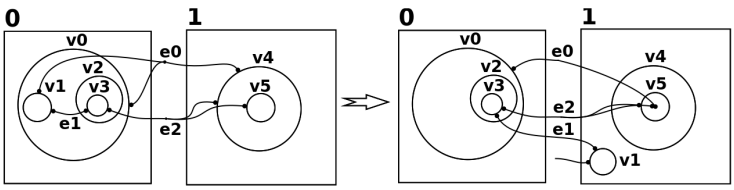


FIG. 2 – *Exemple d'une règle de réaction*

Depuis leur première introduction en 2004 (Milner et Jensen, 2004), plusieurs extensions et raffinements ont été rajoutés aux BRS classiques, motivant leurs intérêts et leur application dans plusieurs domaines pratiques. Certains travaux se sont intéressés à redéfinir les contraintes sur la localité des arcs dans les graphes de liens en ajoutant une probabilité aux arcs donnant lieu aux bigraphes stochastiques (Krivine et al., 2008). D'autres travaux ont proposé l'ajout

d'arcs orientés dans les graphes de liens (bigraphes dirigés) (Damgaard et Birkedal, 2006). Les bigraphes avec partage de nœuds ont aussi été définis introduisant la possibilité qu'un nœud puisse hériter de plusieurs nœuds parents (Sevegnani et Calder, 2015). Dans ce travail, nous nous basons sur une extension assez générique des BRS (Pereira et al., 2012) prenant en charge un type particulier des nœuds appelés Agents dotés d'une certaine intelligence leur permettant d'observer, d'analyser et d'exécuter des actions (sous forme de règles de réaction) sur les bigraphes qui les hébergent.

3 Travaux Connexes

Le Fog computing constitue l'une des principales améliorations de l'environnement Cloud, il permet d'offrir des communications géo-distribuées et moins lentes, tout en facilitant l'intégration des systèmes informatiques et physiques (Yousefpour et al., 2019). Malgré l'importance de cette technologie, peu de travaux mettant en œuvre des techniques formelles se sont intéressés à la modélisation, à la vérification et à la validation des architectures Fog computing pour assurer leur sécurité (Souri et al., 2018). Quelques approches, résumées dans la Table 3, ont retenu notre attention.

En premier, les auteurs des travaux dans (Wazid et al., 2019) ont présenté un schéma d'authentification des utilisateurs et de clés sécurisées pour le Fog, appelé SAKA-FC, préservant les propriétés d'anonymat. Il utilise une fonction de hachage cryptographique unidirectionnelle pour les appareils intelligents, qui se révèlent être sécurisés contre les attaques connues. Cela se fait en adoptant la sécurité basée sur un modèle formel : "Real-Or-Random" (ROR), utilisant des protocoles de sécurité Internet (AVISPA) largement connus et utilisés (Abdalla et al., 2005).

Dans leur article sur la résolution des problèmes de sécurité rencontrés lors de l'externalisation des données du client Fog vers le nœud Fog, les auteurs dans (Samman et al., 2017) se sont concentrés sur le déploiement du protocole de sécurité et de contrôle d'accès entre domaines, dit "Shibboleth", dans un réseau FogIoT pour une architecture sécurisée gérant le problème d'accès. En outre, pour prouver son exactitude par rapport aux propriétés de sécurité considérées, les auteurs ont également formellement vérifié l'architecture proposée à l'aide des réseaux de Petri de haut niveau (HLPN). Le solveur Z3 SMT a ensuite été utilisé pour analyser les règles de flux d'information, ce qui a prouvé la pertinence de l'architecture par rapport aux trois propriétés de sécurité : propriétés de métadonnées, d'authenticité et de proxy.

Dans (Gupta et al., 2017), les auteurs ont introduit iFogSim pour modéliser et simuler des environnements informatiques IoT, Fog et Edge. Ils ont analysé l'architecture iFogSim proposée ainsi que sa conception et son implémentation. Le modèle adopté considéré pour cette architecture est : "Sense-Process-Actuate", dans lequel les capteurs publient des données sur des réseaux IoT, les applications fonctionnant sur des dispositifs Fog s'abonnent et traitent les données provenant de capteurs, et les informations finales obtenues sont ensuite converties en actions transmises aux actionneurs. En outre, la boîte à outils iFogSim mise en œuvre permet de mesurer l'impact des politiques de gestion des ressources applicables aux environnements Fog en ce qui concerne la latence, la consommation d'énergie, la congestion du réseau et les coûts opérationnels. Enfin, les auteurs ont évalué l'évolutivité d'iFogSim en termes de consommation de RAM et de temps d'exécution dans différentes circonstances.

Par ailleurs, afin de prendre en charge la collaboration sécurisée et l'interopérabilité entre diffé-

rentes ressources demandées par les utilisateurs dans le Fog computing, les auteurs de (Dsouza et al., 2014) ont proposé un cadre préliminaire de gestion des ressources dans le Fog, élargissant la plateforme informatique actuelle du Fog en termes de critères de spécification de politique et schémas pertinents. Ils ont présenté les principaux concepts du Fog et ils ont identifié des problèmes de gestion des stratégies qui sont essentiels pour la prise en charge de la collaboration et de la réutilisation des données dans un environnement hétérogène. En outre, les auteurs ont démontré la faisabilité et le caractère pratique de l'approche proposée par le biais d'un ensemble de scénarios de cas d'utilisation.
Un nouveau concept de schéma d'agrégation anonyme et sécurisée (ASAS) dans le Cloud computing public basé sur le Fog est introduit dans (Wang et al., 2018). Le modèle ASAS formalisé consiste en un nœud Fog qui regroupe les données des nœuds terminaux et les transmet au serveur Cloud public. Cela peut aider les terminaux à télécharger leurs données sur le PCS ("Public Cloud Server ") et économiser la bande passante entre le nœud Fog et le PCS. Par ailleurs, les auteurs ont évoqué une technique de cryptage "homomorphic" qui protège non seulement l'identité des terminaux en utilisant des pseudonymes, mais garantit également les données. Enfin, les auteurs ont également prouvé que la sécurité et la performance de leur proposition étaient suffisamment sûres et suffisamment efficaces pour être déployées dans la pratique.

La véritable difficulté de la formalisation du Fog Computing, compte tenu de sa nature distribuée, est de fournir un modèle suffisamment expressif tout en intégrant de manière transparente les considérations de sécurité. Sur cette base, nous remarquons que certes les approches décrites précédemment sont principalement axées sur les exigences de sécurité. Cependant, une attention négligeable est consacrée à la formalisation des concepts du Fog computing (expressivité). Par ailleurs, tous ces travaux s'intéressent à la vérification d'un certain nombre de contraintes de sécurité de manière pratique, après avoir conçu le système Fog en question qui n'est dans aucun cas un système "véritablement" sécurisé, c.-à-d., capable de se défendre contre toutes les menaces crédibles.
Par rapport à ces approches, notre démarche est générale et plus expressive (Table 3, dernière ligne). Elle permet la modélisation formelle des éléments (structuraux et logiques) de l'architecture du Fog tout en supportant certaines politiques de sécurité. L'intérêt porté à la prise en charge des exigences fonctionnelles d'un système dédié au Fog d'un côté, et ses exigences de sécurité de l'autre côté, est le même. D'autre part, à notre connaissance, ce travail de recherche est le premier à faire adopter systématiquement les systèmes réactifs bigraphiques (BRS) pour la spécification d'une architecture Fog géo-distribuée, supportant les aspects :

- Structure physique, en termes de bigraphes,
- Fonctionnement et contrôle, en termes de règles de réaction,
- Sécurité, en termes d'agents abstraits qui constituent la structure additive de notre modèle (CA-BRS) ; ils permettent d'observer (analyser) et de contrôler le fonctionnement sécurisé de ce modèle.

Approches	**Intérêt**	**Modèle/Schéma de base**	**Propriétés préservées**	**Outil**
Wazid et al. (2019)	Authentification utilisateur Schéma Accord-clé	SAKA-FC, Modèle Real-Or-Random (ROR)	Anonymat, Proxy	AVISPA [9]
Samman et al. (2017)	Externalisation de données	Réseaux de Petri de haut niveau	Métadonnées, Authenticité/Proxy	Z3 SMT solver
Gupta et al. (2017)	Politiques de gestion des ressources	iFogSim, Modèle Sense-Process-Actuate	Latence, Consommation d'énergie, Congestion du réseau	iFogSim tool kit
Dsouza et al. (2014)	Collaboration et réutilisation des données dans un environnement hétérogène	Cadre de gestion basé politiques	Collaboration sécurisée, Interopérabilité	Scenarios de cas d'utilisation "Use-case"
Wang et al. (2018)	Agrégation et transmission de données vers un serveur Cloud public	Schéma d'agrégation anonyme et sécurisé (ASAS)	Agrégation sécurisée	Technique de cryptage
Notre Contribution	Modélisation de la structure distribuée d'une architecture Fog	Formalisation de certains services de sécurités et d'accès au réseau	CA-BRS (modèle à base des BRS et les Agents), Collaboration sécurisée, Interopérabilité, Authenticité, Séparation de domaine et contrôle d'accès	BigraphER

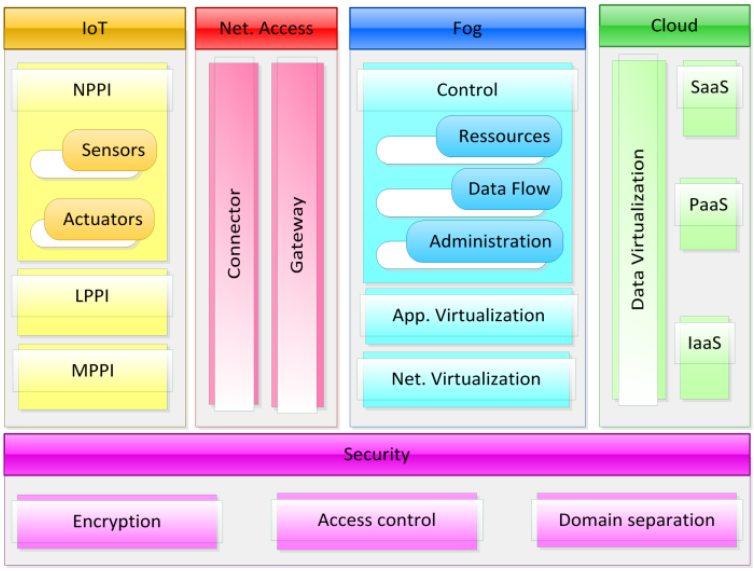
TAB. 1 – *Synthèse des approches existantes*

4 Spécification Formelle

L'objectif de cette section est double, d'une part nous proposons une architecture de référence multi vues pour le Fog computing prenant en charge les aspects de sécurité offerts par ses services, d'autre part, nous associons à cette architecture un cadre sémantique modélisant à la fois ses entités physiques et leur dispersion géographique, ainsi que ses entités virtuelles consacrées aux fonctionnalités de sécurité. Nous montrons par la suite (Section 5) comment ce modèle peut être exploité pour spécifier et vérifier des propriétés comportementales d'un nœud Fog sécurisé de manière formelle.

4.1 Architecture de référence SecAFog

Les architectures Fog Computing se diffèrent d'une référence à une autre ; OpenFog (OpenFog et al., 2017), Cisco (McKendrick, 2016) et NIST (Iorga et al., 2018) mais partagent globalement trois constituants : Objets IoT, Fog computing et Cloud. Chaque constituant possède ses propres propriétés et complète les autres constituants pour assurer leurs tâches.



NPPI : Non Processing Power IoT, LPPI : Low Processing Power IoT,
MPPI : Medium Processing Power IoT

FIG. 3 – *Architecture multi vues proposée pour le Fog computing sécurisé*

Le Cloud, regroupe les centres de données ("Datacenters") et assure le stockage et le traitement des données volumineuses.
Le Fog, représente une couche intermédiaire sur laquelle se base l'architecture Fog. Elle englobe des nœuds Fog capables du stockage, calcul et de mise en réseau.
Le constituant IoT, regroupe les dispositifs responsables de collecte des données ainsi que les

actionneurs ("Actuators") pour exécuter les actions.
Ces architectures possèdent pratiquement le même principe de fonctionnement : les objets IoT de différents types envoient les données collectées continuellement au nœud Fog le plus proche pour que celui-ci exécute des mécanismes d'analyse et de calcul et décide à quel actionneur il doit répondre. Les nœuds Fog sont connectés au Cloud pour transmettre des données d'exception ou volumineuses.
De façon assez similaire, nous avons suggéré dans un travail antérieur (Bouheroum et al., 2019) une architecture complète, générique et à vues multiples dite SecAFog (voir la Figure 3) conçue pour faciliter le traitement de certains aspects de la sécurité dans les nœuds Fog. L'objectif principal du présent travail est de montrer comment intégrer de manière harmonieuse des considérations de sécurité dans les premières activités du cycle de vie d'un système dédié au Fog, en particulier, nous transcrivons une telle architecture en un modèle formel, dédié à la prise en charge des politiques de sécurité pour le Fog Computing. SecAFog est composée de cinq niveaux : Cloud, Fog, IoT, Accès réseau et Sécurité. Plusieurs vues peuvent être considérées, la vue structurelle, la vue fonctionnelle et la vue de sécurité.

Niveau IoT : Il regroupe les dispositifs responsables de collecte des données à travers le niveau accès réseau, provenant de trois types de dispositifs que nous avons classé selon leur puissance de traitement :

- NPPI ("No Processing Power IoT") : un dispositif IoT dans ce groupe est caractérisé par le manque de puissance de traitement et l'utilisation de protocoles de communication industriels non utilisés dans le réseau Internet, ainsi il doit être connecté au Fog à travers le composant Connecteur de la couche Accès réseau (Voir Figure 3). Généralement, nous rassemblons dans ce groupe de dispositifs : 1) Les capteurs ("Sensors") dont le rôle est de collecter des données et de les envoyer pour qu'elles soient analysées et traitées afin de prendre des mesures, d'enregistrer des activités ou d'améliorer le modèle des applications (à des fins d'apprentissage automatique), 2) Les actionneurs ("Actuators") constituent une gamme variée d'appareils chargés d'exécuter des actions définies par les applications. Ils peuvent être une vanne, un ventilateur, une LED, un moteur, etc.
- LPPI ("Low Processing Power IoT") : La pensée bien connue en matière de cryptographie est une tâche très exigeante en termes de temps de calcul, car de tels périphériques qui ont une faible puissance de calcul en raison de la réduction des coûts ne peuvent pas réaliser cette tâche. Le héros de la sauvegarde est le composant de passerelle, il est responsable de l'exécution de la tâche de chiffrement. Un exemple de ces dispositifs peut être une montre intelligente, un drone, un Arduino, etc.
- MPPI ("Medium Processing Power IoT") : ce sont les autres dispositifs qui doivent uniquement être identifiés, contrôlés et autorisés pour accéder au réseau.

Niveau accès réseau : Nous déléguons ce composant de l'architecture SecAfog comme le seul apte à recevoir les données de sa couche IoT et de tous ses autres constituants, il fonctionne grâce à un connecteur (Connector") ou une passerelle ("Gateway") comme suit :

- Connecteur : ce composant est dédié à la réalisation de deux tâches principales : 1) le cryptage et la génération de la clé cryptographique, 2) la transformation ou

l'adaptation du protocole utilisé dans le but d'acheminer les données générées sur le réseau Internet vers les nœuds Fog ou le Cloud, si nécessaire.

— Passerelle : les principaux services offerts par ce composant sont : 1) vérifier quels sont les appareils IoT autorisés à accéder au réseau, 2) s'assurer que les appareils utilisent le bon protocole de cryptage. Ceci est généralement réalisé grâce à la coopération des composants de cryptage et de contrôle d'accès présents dans la couche de sécurité.

Niveau Fog : Nous représentons à ce niveau les parties logiques que l'on peut trouver dans un nœud Fog idéal, il comporte trois sous-composants :

— Contrôle : gère toutes les actions pouvant être effectuées par le nœud Fog. Trois modules distincts sont identifiés dans ce composant : 1) Ressource responsable de la gestion des ressources disponibles du nœud Fog (mémoire, capacité de stockage, temps processeur), 2) Le module de flux de données ("Data Flow") qui implémente les règles définies par le sous-composant de séparation de domaine dans la couche de sécurité, il définit également le sens de flux des données transférées et 3) Le module d'administration qui inclut les fonctionnalités logicielles. En particulier, un administrateur donne l'ordre d'installer, de mettre à jour ou de supprimer une application, ou de diffuser les mises à jour concernant les stratégies et règles dictées par le Cloud aux différents nœuds Fog.

— Virtualisation des applications : elle consiste en une plate-forme de virtualisation au format : IOX (Cisco, 2016), VMWare (Chaubal, 2008) ou même un conteneur (comme un Docker (Merkel, 2014) contenant toutes les applications installées dans le nœud Fog. En effet, chaque application est installée dans un conteneur de machine virtuelle différent, en maintenant le principe de sécurité bien connu (sendbox) qui permet d'empêcher l'application d'accéder aux autres zones de mémoire.

— Virtualisation du réseau : c'est une couche logicielle qui offre les fonctionnalités du réseau, telles que le routage, le DNS, etc. (à ne pas confondre avec le middleware). Dans ce cas, cela pourrait correspondre à VMware ou Vsphere (Chaubal, 2008). Cette couche permet notamment d'installer sur une même machine, appartenant à un ISP ("Internet Service Provider" comme par exemple Telecom Algérie) de nombreux nœuds Fog pour différents clients de l'ISP pouvant se situer dans la même zone géographique (bâtiment partagé par certaines sociétés.

Niveau Cloud : Nous retrouvons dans cette couche les deux fonctionnalités principales à savoir les services Cloud (SaaS, PaaS et IaaS) et la virtualisation des données qui permet d'offrir une interface unifiée manipulant les données.

Niveau Sécurité : Nous remarquons que la couche de sécurité représente la couche horizontale (Figure 3), dont tous les composants des autres couches bénéficient de ses services que nous limitons dans ce travail à :

— Cryptage : spécifier les protocoles utilisés pour crypter les données transférées au sein du réseau et assurer leur continuité,

— Contrôle d'accès : spécifier quels utilisateurs (IoT ou nœuds Fog) sont autorisés à accéder au réseau,

— Séparation de domaine : définir les règles associant les nœuds Fog aux différents types d'applications, les données de message transférées ou le contrôle et la manière

dont le flux de données est conduit. Par exemple, le nœud Fog f1 envoie et reçoit des données de f2, mais le nœud f2 ne peut envoyer que des données à f1.

Les services de sécurité offerts par l'architecture SecAFog, respectant le triangle de sécurité largement utilisé comme référence de base (Bousquet, 2015) (Confidentialité, Disponibilité et Intégrité), sont illustrés dans la Figure 4. Nous donnons pour chaque propriété, les éléments de l'architecture intervenant dans son assurance selon des mécanismes appropriés. Par exemple, la confidentialité dans ce cas est assurée, soit par le mécanisme de chiffrement à clé publique, utilisé pour l'authentification : répondre à la question "Qui êtes-vous ? (rôle du Composant "Encryption" de SecAFog), soit par le principe d'autorisation : répondre à la question "Qu'est-ce que vous êtes autorisé à faire ?" (Assuré par le composant "Access Control" de SecAFog).

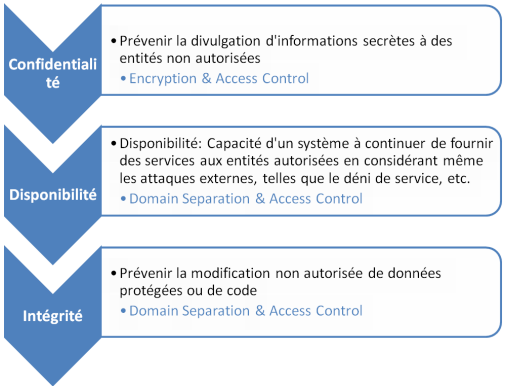


FIG. 4 – *Aspects de sécurité pris en charge par SecAFog*

L'architecture de référence SecAFog proposée est une abstraction du système Fog spécifié, elle est représentée sous la forme d'un ensemble de faits construits dans une intention considérée. En d'autres termes, elle permet de représenter une vue simplifiée d'une partie du système ou du système lui-même et de masquer certaines difficultés. Nous notons que tous les services de sécurité collectés dans la couche sécurité de SecAFog se trouvent implémentés de manière dispersée à travers pratiquement tous les nœuds de notre architecture physique d'un système Fog. Cette solution architecturale doit encore évoluer pour permettre plus de précision dans la description et l'utilisation de ces différents rôles. Le recours aux méthodes formelles pourrait apporter une solution efficace à cette préoccupation (section suivante).

4.2 CA-BRS : modèle formel pour le Fog sécurisé

Nous proposons un modèle nommé CA-BRS (Control Agent and BRS), une extension des systèmes réactifs bigraphiques (BRS) avec agents de contrôle qui s'avère être un cadre sémantique formel approprié pour la spécification de l'architecture SecAFog proposée. Il sépare, pour un système complexe, les problèmes de calcul et de localisation des entités physiques, respectivement en deux modèles différents, le modèle des agents virtuels et le modèle Bigraphe. Ce modèle est assez similaire aux deux modèles : BiAgents (Pereira et al., 2012), (Marir et al.,

2018) et BigActors (Pereira et al., 2013) ; il se base sur les mêmes formalismes Agents et BRS, néanmoins la définition de son modèle de calcul est simplifiée par rapport à celle des modèles précédents. Ainsi, une forme particulière attribuée aux règles de réaction est suggérée (RR), elle repose sur l'ajout d'informations de types déclencheur ("Observation") et contrôle ("ControlActions") aux règles de réaction (voir Figure 5a), afin de capturer la relation entre les différentes entités définies par le modèle CA-BRS. En outre, les agents abstraits dans ce contexte sont en mesure d'observer, de contrôler et d'adapter les services correspondants affectant l'état du système (voir Figure 5b). Les états possibles du système spécifié par un CA-BRS peuvent être atteints par l'exécution de ces règles de réaction RR. Ainsi, l'état d'un système à un moment donné est défini par le couple (B_j, H_j) reliant la position (emplacement $H_j \in H$) de ses agents opérationnels à sa structure physique actuelle (bigraphe B_j).

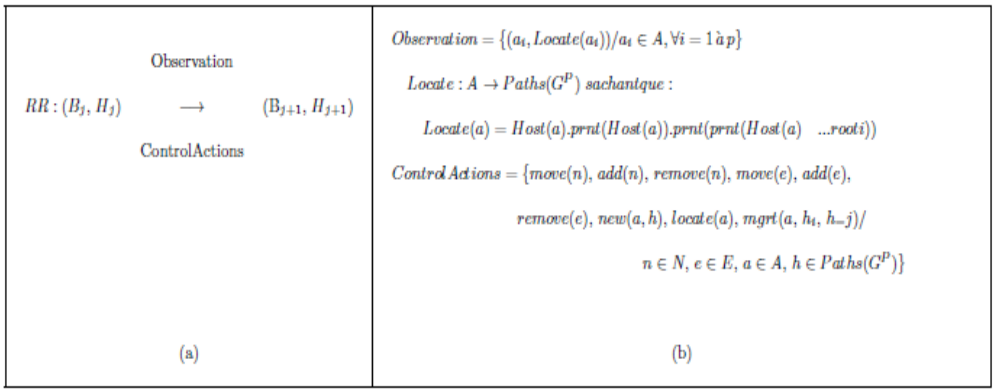


FIG. 5 – *RR : Forme étendue d'une règle de réaction dans CA-BRS*

L'architecture SecAFog qui représente un système distribué ayant deux niveaux distincts : un physique incluant tous les composants physiques (routeur, capteurs, nœud Fog, etc.) et un autre virtuel pour les programmes et les protocoles logiciels qui y interviennent, peut être spécifiée mathématiquement par un modèle CA-BRS. Un tableau de correspondances illustrant la sémantique attribuée aux différents éléments constituant SecAFog est déduit (Table 2) selon les cinq niveaux repérés dans cette architecture. La structure physique dans ce contexte permet de modéliser les trois niveaux IoT, Fog et Cloud sous forme de régions d'un bigraphe. Les nœuds appartenant aux différentes régions représentent les services offerts ou les dispositifs physiques ("Sensors", "Actuators", etc) qui se trouvent dans chacun des trois niveaux de SecAFog. La structure virtuelle de CA-BRS modélise le niveau de sécurité à travers un ensemble d'agents, chaque catégorie d'agents gère un service particulier, par exemple les agents de type *AgE* s'occupe de l'authentification ("Encryption"). Elle contient aussi deux autres types d'agents *AgC* et *AgA* qui détiennent les services d'administration et de contrôle dans un nœud Fog.

Eléments de l'architecture Fog	Concepts de CA-BRS
Fog level	Root0 : Fog
Fog Node	Node of control : FN
Control	Control Agent : AgC / host(AgC) = Services, NV, AV
Resources	Node of control : Resources
Memory(RAM)	Node of control M and parent Resources
CPU	Node of control CPU and parent Resources
Storage(Disks)	Node of control D and parent Resources
Data flow	Node of control : DF
Administration	Control Agent : AgA / host(AgA)=FN
Applications virtualization Services	Node of control : AV
Network virtualization Services	Node of control : NV
Interaction between Fog node elements	Edges : e1, e2, e3
Network between Fog Nodes	Edge : FogFog links NV nodes of each Fog Node
Relationships between Fog and Cloud	Edge : FogCloud links NV nodes of each Fog Node to DV node of Cloud root
IOT level	Root1 : IOT
NPPI	Node of control : NPPI
Sensor device	Node of control Sensor and parent NPPI
Actuators device	Node of control Actuators and parent NPPI
LPPI	Node of control : LPPI
MPPI	Node of control : MPPI
Cloud level	Root2 : Cloud
Services cloud (Saas,Paas,Iaas)	Node of control : Services
Data virtualization	Node of control : DV
Network Access level	Edges
Gateway	Edges : gateway links LPPI or MPPI nodes to NV nodes
Connector	Edges : ConnectorGateway links Sensors or Actuators Nodes to NV nodes
Security level	Control Agents
Encryption function	Control Agent : AgE / host(AgE)=NPPI,LPPI,MPPI
Domain Separation function	Control Agent : AgD / host(AgD)=DF
Access Control function	Control Agent : AgAC / host(AgAC)=DF

TAB. 2 – *Une sémantique basée CA-BRS pour les éléments de l'architecture SecAFog*

Les interactions entre les différents nœuds d'une même région ou des régions différentes sont modélisées par les liens d'un bigraphe. Le niveau "Network Access" de l'architecture proposée est aussi formalisé par deux types de liens "ConnectorGateway" et "Gateway". Nous explicitons la définition formelle de CA-BRS appliqué à un système Fog comme suit :