

Positionnement sur le Web Sémantique des Objets

Nicolas Seydoux^{1,2,3}

Maxime Lefrançois⁴

Lionel Médini⁵

¹ LAAS-CNRS

7 avenue du colonel Roche, F-31400 Toulouse, France

² IRIT Toulouse, France

³ UNIV. DE TOULOUSE,

INSA, LAAS, F-31400 Toulouse, France

⁴ Mines Saint-Étienne, Univ Lyon, Univ Jean Monnet, IOGS, CNRS, UMR 5516, LHC,

Institut Henri Fayol, F - 42023 Saint-Étienne France

⁵ Univ. Lyon, Université Lyon 1 LIRIS, CNRS UMR5205,

F-69622, Villeurbanne, France

Résumé

L'Internet des Objets (IoT en anglais) est un domaine riche en enjeux scientifiques, technologiques et sociétaux. Afin de faire face aux problématiques d'interopérabilité faisant obstacle au développement de l'IoT, les principes et les technologies du Web Sémantique sont intégrées aux réseaux d'objets connectés. Cette intégration est désignée sous le nom de Web Sémantique des Objets (SWoT en anglais). L'émergence de ce domaine repose sur l'adaptation des technologies du Web Sémantique aux contraintes de l'IoT. Ce papier de positionnement vise à décrire les vocabulaires principaux utilisés dans le SWoT, ainsi que l'architecture dans lequel la pile technologique du Web Sémantique se déploie, afin d'ensuite identifier les problématiques émergentes dans le domaine du SWoT. Ces problématiques sont des domaines d'intérêts potentiels pour les futurs travaux de la communauté Ingénierie des Connaissances (IC).

1 Introduction

Les objets connectés sont désormais une réalité du quotidien, et depuis le premier emploi du terme en 1999, les enjeux scientifiques, techniques et sociétaux autour de la notion d'Internet of Things (IoT) n'ont fait que croître (Ashton, 2009). En particulier, le déploiement de réseaux dits IoT a des enjeux dans des domaines divers, tels que la ville intelligente, la télé-santé, l'agriculture numérique, ou l'industrie du futur. La grande hétérogénéité de ces domaines d'application a mené à l'émergence de "silos", avec l'intégration verticale de l'ensemble de la chaîne de traitement, depuis l'objet collectant la donnée jusqu'au programme l'intégrant dans son processus de collecte et de décision, sans oublier l'interface de l'application finalement consultée par l'utilisateur. Cette fracturation verticale est une source de problèmes d'interopérabilité et limite donc l'émergence d'applications complexes et multi-domaines, ainsi que l'intégration horizontale d'objets ou de services. Depuis sa création il y a 30 ans, le Web vise à masquer l'hétérogénéité des plateformes matérielles et logicielles, et

à permettre l'utilisation d'applications aussi spécifiques et performantes que possibles, en standardisant les langages et protocoles destinés à chaque type d'application. Dans cet esprit, le World Wide Web Consortium (W3C) s'est intéressé aux silos créés par l'IoT et est actuellement en train de finaliser plusieurs standards concernant le Web des Objets (WoT)¹. Depuis leurs origines (Berners-Lee et al., 2001), les principes et technologies du Semantic Web (SW) ont été dévolus à la communication entre agents complexes, et ont promu l'interopérabilité. L'utilisation de modèles riches et expressifs semble particulièrement indiquée dans le cas des objets connectés, car on se situe dans le cas d'une communication destinée principalement à être interprétée par une machine, et ce n'est qu'une fois transformée qu'elle sera éventuellement transmise à un utilisateur humain. La convergence entre les domaines de l'IoT, du WoT et celui du SW ont mené à l'apparition d'un nouveau domaine de recherche, le Semantic Web of Things (SWoT). D'abord représenté par les réseaux de capteurs sémantisés (Sheth et al., 2008), le SWoT s'est ensuite diversifié pour aussi intégrer des actionneurs (Wang et al., 2015). Cependant, l'intégration des technologies du SW dans les réseaux d'objets connectés n'est pas triviale. En effet, ce sont des technologies qui sont en général demandeuses en ressources, autant en termes de puissance de calcul pour le raisonnement qu'en bande passante pour le transfert de données ou qu'en mémoire pour le stockage. Cette consommation importante de ressources est en opposition directe à la nature contrainte des réseaux IoT, dans lesquels sont déployés de nombreux objets aux ressources limitées. En effet, afin de permettre la dissémination d'un grand nombre d'objets dans l'environnement, il est nécessaire de miniaturiser ces objets, ainsi que de potentiellement les rendre autonomes en énergie. La combinaison de ces deux contraintes amène à une limitation nécessaire de la puissance de calcul, de l'espace mémoire, et des capacités de communication de l'objet. De plus, d'après des estimations récentes, le nombre

1. <https://www.w3.org/WoT/>

d'objets connectés va continuer à croître rapidement, pour passer d'environ 23 milliards en 2018 à une estimation de 75 milliards en 2025. Cette augmentation importante du nombre d'objets s'accompagne nécessairement de l'augmentation du volume de données qu'ils collectent, amenant un problème de passage à l'échelle pour les technologies du SW.

Partant de ce constat, nous questionnons dans ce papier une possible convergence entre les domaines du SWoT et des objets contraints. En particulier, nous nous appuyons sur des travaux existants dans la communauté IC dans ces deux domaines, en termes d'intérêt et de faisabilité, pour mettre en lumière certaines questions qu'une telle convergence fait émerger. Après une présentation des ontologies principales utilisées pour produire des messages compréhensibles par les machines dans le SWoT, une architecture de référence permettant de situer les contributions du Web Sémantique dans l'IoT est détaillée. Les défis représentés par le déploiement du SWoT, ainsi que des contributions y faisant face, sont ensuite décrits, afin d'identifier de futurs domaines d'intérêt pour la communauté.

2 Interopérabilité entre objets

L'un des moteurs poussant au développement du SWoT est le besoin d'interopérabilité de l'IoT, et en particulier d'interopérabilité sémantique (Murdock et al., 2016). En effet, les données et les systèmes qui les collectent sont extrêmement divers, ce qui rend complexe l'intégration de multiples sources de données à grande échelle. Afin de décrire le domaine de l'IoT, de nombreuses ontologies ont été proposées, c'est pourquoi nous avons identifié celles ayant un intérêt particulier. Les ontologies que nous avons listées ici sont celles se rapportant directement aux réseaux d'objets connectés, permettant de décrire les dispositifs, la manière dont ils interagissent avec le monde réel, ou la manière dont leurs services peuvent être découverts ou sollicités. Les domaines d'application de l'IoT sont nombreux, et décrire les ontologies qui s'y rapportent sort du périmètre de ce papier.

2.1 Les ontologies standardisées

Dans les domaines technologiques, le premier recours au manque d'interopérabilité est la standardisation. Avec l'émergence du SWoT, les standards incluent des ontologies, de façon à promouvoir l'utilisation de quelques ontologies de référence, plutôt que d'observer un morcellement et la création de nouvelles ontologies avec chaque nouveau projet. Les organismes de standardisation qui contribuent aux ontologies du SWoT sont notamment : (1) Le W3C, organisme international de standardisation du Web et du Web sémantique. (2) L'European Telecommunication Standards Institute (ETSI), organisme européen de standardisation pour les télécommunications. (3) oneM2M, consortium international rassemblant des organismes de standardisation (dont l'ETSI), des organismes de recherche et des industriels autour d'un standard pour l'IoT. Le groupe d'incubation Semantic Sensor Network du W3C a publié en 2011 un rapport analysant les différents modèles conceptuels existants pour décrire les capteurs et leurs observations, ainsi qu'une proposition d'ontologie SSNX (Lefort et al., 2011). L'ontologie SSNX a été massivement réutilisée par

d'autres ontologies et jeux de données. Le groupe de travail Spatial Data on the Web Working Group commun aux organismes de standardisation Open Geospatial Consortium (OGC) et W3C ont récemment publié une mise à jour de cette ontologie, nommée **SOSA/SSN** (Haller et al., 2017; Janowicz et al., 2018; Haller et al., 2018). Celle-ci spécifie la sémantique des capteurs et actionneurs, entre autres. Elle permet de décrire notamment les capteurs, les propriétés des choses d'intérêt qu'ils observent, les observations qu'ils font et le résultat de ces observations. De manière analogue, elle permet de décrire les actionneurs, les propriétés des choses d'intérêt sur lesquelles ils peuvent agir, et les actionnements qu'ils font et le résultat de ces actionnements. L'ontologie SOSA/SSN est modulaire. Par exemple : (1) SOSA (Sensor, Observation, Sampler, and Actuator) est un module qui comprend seulement quelques termes et peu d'axiomatisation ; (2) SSN (Semantic Sensor Network) importe SOSA, introduit d'autres termes, et enrichit l'axiomatisation de SOSA ; (3) SSN-System (SSN System Capability module) importe SSN (et SOSA par transitivité), et définit des classes supplémentaires pour modéliser les capacités des systèmes, leurs domaine d'opération ou leurs conditions nominales de déploiement.

Le W3C comprend un autre groupe de travail qui contribue directement au SWoT : le W3C Web of Things working group². Entre autre, ce groupe a pour tâche la conception d'une ontologie, la **Thing Description ontology**, ainsi que son association à un ensemble de métadonnées décrivant les données exposées aux applications par les objets, leurs règles de sécurité, ou les informations de connexion à leurs points d'accès. La spécification de l'ontologie associée au modèle Thing Description est disponible dans (Kaebisch et al., 2019), encore en cours de développement. Une Thing est définie dans (Kovatsch et al., 2019) comme l'abstraction d'une entité (virtuelle ou physique) devant être manipulée par une application IoT, *e.g.* un objet, un service, ou une entité logique telle qu'une pièce ou un bâtiment. L'ontologie Smart Appliances Reference (**SAREF**) (Daniele et al., 2015), dont le développement est soutenu par l'ETSI, est centrée sur le concept de Device en tant qu'objet physique remplissant une ou plusieurs fonctions. Initialement conçue pour décrire le domaine de la maison intelligente, SAREF a été étendue par des modules spécialisés pour l'énergie, les bâtiments, l'environnement ou l'usine du futur. (Poveda Villalón et al., 2018) a analysé l'alignement possible entre SAREF et SSN, notamment pour l'étendre à la description d'un scénario d'agriculture numérique.

La **oneM2M Base Ontology**³ est documentée dans l'une des spécifications techniques⁴ du standard international oneM2M par le consortium éponyme. La oneM2M Base Ontology est une ontologie de type "cœur de domaine" : elle ne décrit qu'un ensemble réduit de concepts de haut niveau qui vise à être étendu par les implémentations du standard. Ces concepts, par exemple Device, Service ou Variable, font référence à

2. W3C WoT Charter - <https://www.w3.org/2016/12/wot-wg-2016.html>

3. https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl

4. TS-0012 : <http://www.onem2m.org/technical/published-drafts>

des éléments du standard. Un alignement entre SAREF et la oneM2M base Ontology est aussi distribué par oneM2M⁵. La prochaine version de SAREF devrait également comporter un alignement avec SOSA/SSN.

Il subsiste des questions ouvertes sur la modélisation et l'utilisation de ces ontologies, voici deux points de départ. (1) La section (Haller et al., 2017, §7) discute de questions ouvertes de modélisation avec SOSA/SSN. Les différentes options présentées n'ont pas encore fait l'objet d'articles de recherche. (2) L'ontologie TD étant maintenant stabilisée, il reste à ré-étudier l'alignement et l'interopérabilité possible avec SOSA/SSN.

2.2 Autres ontologies importantes du SWoT

Ces ontologies ne sont pas proposées par des organismes de standardisation, mais sont elle-même construites sur des ontologies de référence, issues d'organismes comme le W3C, ou de standards *de facto* dont le statut est acquis par l'usage.

L'ontologie **SEAS**⁶ (Lefrançois, 2017) est une ontologie modulaire dont tous les termes sont contenus dans le même espace de noms. SEAS étend SOSA/SSN, et propose un cœur de quatre patrons ontologiques décrivant les systèmes physiques et leurs connexions, les valeurs associées à leurs propriétés, et les processus par lesquels ces associations de valeurs sont faites. Ces patrons sont ensuite instanciés dans chaque module pour un domaine particulier. Dans le contexte d'un projet ETSI Specialist Task Force 556⁷ ces patrons ontologiques sont partiellement incorporés dans SAREF.

L'ontologie **S3N - Smart Semantic Sensor Networks**⁸ (Sagar et al., 2018) est une ontologie modulaire qui étend SSN pour décrire des capteurs intelligents, c'est à dire des plateformes hébergeant au moins un capteur, un dispositif communicant, et un micro-contrôleur.

IoT-O⁹ est une ontologie modulaire pour l'IoT, construite selon un ensemble de bonnes pratiques visant à permettre sa réutilisation (Seydoux et al., 2015, 2016a). Elle est notamment basée sur SSN initialement, et dispose d'une version mise à jour basée sur SOSA. Les différents modules de IoT-O, e.g. Capteurs et Observation ou Actionneurs et Actions, couvrent chacun un domaine de l'IoT, et sont indépendants les uns des autres.

schema.org est une ontologie légère décrivant un large panel de concepts se rapportant aux ressources publiées en ligne (e.g. blogs, boutiques en ligne ou contenu multimédia). Ce cœur a été ensuite étendu avec des vocabulaires se rapportant à un domaine spécifique présent sur le Web, comme les références bibliographiques¹⁰, et en particulier l'IoT, avec **iot.schema.org**. Un point de départ pour suivre le développement de cette extension ou contribuer est le groupe collaboratif **GitHub** <https://github.com/iot-schema-collab/>.

5. <https://git.onem2m.org/MAS/BaseOntology>

6. <https://w3id.org/seas/>

7. <https://portal.etsi.org//STF/STFs/STFHomePages/STF556>

8. <https://w3id.org/s3n/>

9. <http://irit.fr/recherches/MELODI/ontologies/IoT-O>

10. <https://bib.schema.org/>

3 Une architecture de référence pour le SWoT

La maturité du domaine du Web Sémantique, et le développement rapide du domaine de l'IoT, amènent l'émergence de problématiques spécifiques au SWoT. Ces problématiques s'articulent autour d'une dichotomie entre les **ressources nécessaires** au déploiement des technologies du Web Sémantique et les **capacités limitées** des nœuds présents dans les réseaux IoT. Ces contraintes sont caractérisées dans la suite de la présente section par l'introduction d'un patron architectural de référence pour le SWoT, utilisé pour situer les problématiques émergentes. Ce patron est ensuite instancié par une architecture orientée avatar permettant d'envisager la construction d'applications WoT et SWoT.

3.1 Que sont le Cloud et le Fog computing ?

Le Cloud computing (Mell and Grance, 2011) est un paradigme dans lequel les traitements sont effectués à distance sur des machines dotées d'importantes ressources : puissance de calcul, mémoire, stockage, bande passante, voire systèmes et applications. L'**accessibilité via le Web** et l'**élasticité** des machines du Cloud en font des relais intéressants pour les applications du SWoT. Dans ce modèle, les données sont collectées par les objets connectés, puis concentrées sur les nœuds Cloud où elles sont stockées, traitées, et intégrées à des applications.

Cependant, les réseaux d'objets connectés contiennent par nature des objets contraints, et très disséminés dans leur environnement. Leurs capacités de communication limitées peuvent être un obstacle à la communication systématique avec des nœuds distants. De plus, la concentration des données dans un serveur Cloud peut introduire un délai important, dépendant du volume de données qui transitent et de la qualité du lien réseau. Pour faire face à cette fracture entre les infrastructures Cloud et les sources de données de l'IoT, un nouveau paradigme est apparu, le Fog Computing (Bonomi et al., 2012), basé sur l'utilisation des ressources de calcul et de stockage disponibles **en bordure de réseau**. Les nœuds Fog sont donc **massivement distribués, hétérogènes**, et disposent de **capacités de calcul limitées**, et se situent entre les objets connectés et les machines du Cloud. Cette définition englobe notamment les passerelles que l'on retrouve dans de nombreux déploiement d'objets connectés, et qui assurent la connexion entre les protocoles de l'IoT et Internet, permettant ainsi de transporter les données des objets jusqu'aux serveurs distants du Cloud.

Le Cloud comme le Fog computing visent à fournir une abstraction des ressources de calcul pour permettre une gestion plus souple des tâches. Cependant, il existe des différences fondamentales entre ces paradigmes. **(I)** Les nœuds du Fog sont situés à **proximité** des objets connectés, fournissant ainsi des capacités de calcul plus proche des sources de données dans le cas de l'IoT. **(II)** Le traitement distribué supporté par le Fog computing a des propriétés de résilience, par l'absence de point critique sur lequel repose l'intégralité du calcul. **(III)** Cloud et Fog offrent des approches différentes pour le passage à l'échelle. Le Cloud s'appuie sur deux axes : l'attribution de machines supplémentaires à la tâche en cours, ou l'attribution de ressources

supplémentaires aux machines qui y sont déjà dédiées. Dans le Fog computing, les ressources des nœuds n'étant pas flexibles comme dans le Cloud, seule l'augmentation du nombre de machines est possible. Cependant, cette augmentation se faisant localement par rapport aux sources de données, et s'appuyant sur des machines faisant potentiellement déjà partie du déploiement d'objet, elle s'intègre naturellement au développement massif des grands réseaux d'objets, amenant des ressources complémentaires à celles des serveurs Cloud (Dastjerdi and Buyya, 2016). (IV) La nature distribuée des nœuds constituant l'infrastructure Fog rend aussi plus fluide la répartition de la puissance de calcul face à la mobilité des besoins, ce qui est en particulier adapté au domaine de l'IoT (Dastjerdi and Buyya, 2016), où les objets sont disséminés dans un espace géographiquement étalé dans lequel les utilisateurs et certains objets se déplacent. Le Fog computing n'a pas vocation à remplacer le Cloud computing, mais bien à le compléter. La capacité de calcul limité des nœuds Fog, leur mobilité, et la localité de leur déploiement ne sont pas adaptés à la prise en charge de cas d'utilisation du Cloud computing. Le rapprochement de ces deux paradigmes est un vecteur intéressant pour fournir des infrastructures soutenant le développement de l'IoT en général, et du SWoT en particulier.

3.2 Structurer une architecture de référence du SWoT

Tous les nœuds de l'IoT ne sont pas équivalents, et ce domaine se distingue par l'hétérogénéité des capacités des machines impliquées, depuis des objets très contraints jusqu'aux serveurs Cloud dotés d'une très grande puissance de calcul. Les principales caractéristiques distinguant les nœuds sont : la puissance de calcul, la mémoire, le stockage, les capacités de communications, et le type de source d'énergie.

Trois classes de nœuds se détachent par l'analyse de la littérature via ces critères :

1. Les **nœuds Cloud**, caractérisés par leur puissance de calcul importante, leur large capacité de communication (nombreux protocoles, connectés en permanence), ainsi que leur mémoire et leur capacité de stockage étendue. Cette classe de nœuds inclut les serveurs déployés dans le Cloud, et peut aussi être étendue aux nœuds dont la puissance est importante relativement aux autres machines du même déploiement, pouvant contextuellement s'appliquer à une station de travail classique. De tels nœuds sont présents dans les architectures mises en jeu dans des travaux tels que (Ben-Alaya et al., 2015; Szilagy et al., 2016; Su et al., 2018), avec des appellations variables.
2. Les **nœuds Fog**, caractérisés en premier lieu par leurs importantes capacités de communication, et leur rôle de pourvoyeur de contenu entre objets connectés et nœuds Cloud en assurant la continuité des protocoles. Ce rôle fait que ces nœuds sont souvent désignés par le terme de "passerelle" dans la littérature (Compton et al., 2009; Ben-Alaya et al., 2015).
3. Les **objets contraints**, caractérisés par leurs capacités de traitement et de communication limitée, leur autonomie en énergie, et leur très faible capacité de stockage. Ces nœuds sont par définition présents dans tous les réseaux IoT, dont ils

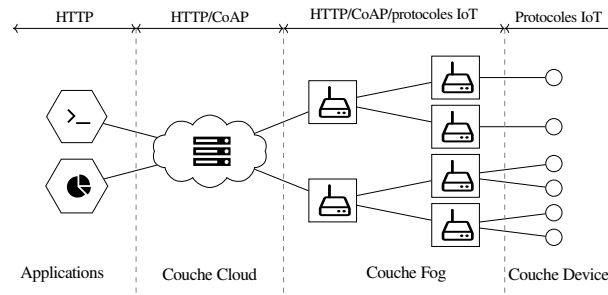


FIGURE 1 – Patron architectural de référence pour le SWoT

sont la source principale d'informations sur l'environnement.

3.3 Un patron architectural de référence pour le SWoT

Les classes de nœuds précédemment définies constituent un patron architectural récurrent dans la littérature, découpé en trois couches : **Cloud-Fog-Devices**, représenté sur la Fig. 1. Dans ce patron architectural, les services sont exposés aux utilisateurs par un nœud Cloud via des protocoles Web, lequel collecte des données produites par des objets via l'entremise de nœuds Fog. Ce patron architectural, et les différentes couches qu'il définit, seront utilisés comme référence dans la suite de ce papier.

3.4 Composition de services métier orientée-avatars

Depuis les automates programmables et les robots, des éléments de programmation ont été associés aux objets. Avec l'avènement des communications en réseau (sans fil), ils ont évolué vers des systèmes de contrôle distribués, des systèmes embarqués et, plus récemment, l'intelligence ambiante et la robotique distribuée (Galloway and Hancke, 2013). L'IoT est une conséquence directe de cette évolution et vise à exploiter les capacités de communication d'Internet, la puissance de calcul quasi illimitée des infrastructures de la couche Cloud et des interfaces utilisateur modernes pour fournir aux utilisateurs finaux des applications utiles. Le WoT s'appuie sur l'IoT et encourage l'utilisation des standards du Web.

Mais dès que des objets physiques entrent en jeu, la programmation de telles applications devient plus complexe et moins déterministe. Les données des capteurs souffrent d'imprécisions, les actionneurs ont besoin de boucles de rétro-action, les processus critiques et de synchronisation nécessitent une attention constante et les communications réseau peuvent perdre des données ou être interrompues. Les défenseurs des Systèmes Cyber-Physiques (CPS) affirment que de telles difficultés proviennent de phénomènes physiques et doivent être modélisés avec des modèles et des processus informatiques (Lee, 2008). Parmi les nombreuses architectures CPS qui ont été conçues, bon nombre d'entre elles sont à base de composants et incluent des entités logicielles qui modélisent ces préoccupations.

Dans la communauté WoT, nous avons proposé (Jamont et al., 2014; Mrissa et al., 2015) la notion d'**avatar** pour désigner les artefacts logiciels attachés à un objet et agréger le code né-

cessaire pour exécuter des applications WoT. Comme détaillé ci-dessous, les avatars reposent sur une architecture interne à base de composants, de sorte que toutes les préoccupations nécessaires du point de vue des CPS puissent être décrites. Plus récemment, le groupe de travail WoT du W3C a proposé la notion de « servient » actuellement définie pour normaliser les objets logiciels dans les applications WoT. Les servients sont très proches des avatars : ils fournissent un accès aux objets, peuvent être exécutés sur ceux-ci dans la couche Device, mais aussi dans les couches Fog et Cloud, et peuvent interagir avec d'autres serveurs. Les deux s'appuient également sur des technologies sémantiques pour échanger des données compréhensibles par une machine. Comme les standards du WoT doivent faire face à une variété de cas d'utilisation et de plateformes, l'architecture cliente ne spécifie que des blocs de construction (« environnement d'exécution », « modèle de ressource », etc.). Les avatars peuvent être considérés comme une spécialisation de servients, davantage centrés sur le déploiement et l'exécution d'applications WoT, et s'appuyant sur une architecture à base de composants pour tirer parti des avancées liées aux préoccupations et aux exigences spécifiques dans divers domaines.

3.5 Avatars

Certains composants de l'architecture de l'avatar (Figure 2) sont dédiés au contrôle des objets et d'autres implémentent le comportement autonome, adaptatif et collaboratif des avatars. La configuration physique est découplée de son architecture logique : un avatar peut adapter de manière dynamique la distribution de ses composants à travers le patron architectural de référence pour améliorer leur efficacité. Nous avons regroupé les composants d'avatar en 8 modules fonctionnels.

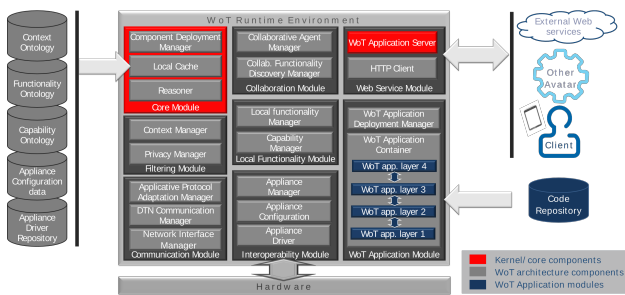


FIGURE 2 – Architecture logicielle de l'avatar

Le **module Core** comprend des composants utilisés à plusieurs étapes du cycle de vie d'un avatar. Le gestionnaire de déploiement de composants définit quels composants d'avatar seront instanciés par rapport aux fonctionnalités de l'objet, et où¹¹. Chaque avatar intègre un moteur d'inférences, utilisé par d'autres composants pour traiter les informations sémantiques relatives aux capacités, aux fonctionnalités et au contexte. Il en

11. Les composants d'un avatar peuvent être situés : sur l'objet si celui-ci dispose de suffisamment de capacités de traitement ou pour des modules de code possédant des contraintes temporelles ; dans la couche Fog pour les processus impliquant une communication inter-avatars localisée ; ou dans la couche Cloud pour les processus gourmands en calculs.

va de même pour le cache local, qui stocke des informations sémantiques provenant de sources diverses (objet, référentiels, contexte externe) et reflète l'état actuel de l'avatar.

Le **module d'interopérabilité** fournit aux autres modules de l'avatar une interface uniforme pour interagir avec l'objet auquel il est attaché. Cette interface consiste en un ensemble de *capacités* représentant l'API de l'objet. Il charge les pilotes depuis un dépôt de la plateforme WoT (voir plus loin) et les utilise pour identifier les schémas de communication compris par l'objet. Finalement, il télécharge sur l'objet la configuration appropriée.

Le **module de filtrage** limite l'exposition des fonctionnalités et les échanges de données. Si, pour des raisons de confidentialité ou de sécurité, certaines fonctionnalités ne sont pas réalisées par l'avatar, elles seront filtrées par le gestionnaire de la confidentialité. Le gestionnaire de contexte a un rôle plus complexe, qui permet notamment de décider quelles fonctionnalités peuvent être exposées, à partir de quelles capacités de l'objet les réaliser, et avec quels autres avatars collaborer pour composer des fonctionnalités complexes (Terdjimi et al., 2016).

Le **module de communication** assure une communication fiable avec l'objet. Il sélectionne l'interface réseau appropriée (Ethernet, Wi-Fi, Zigbee...) et les protocoles (CoAP, HTTP...) en fonction des objectifs de la communication et des besoins en performances (débit / consommation d'énergie). Il prend également en charge les interruptions de connectivité (Mahéo et al., 2012).

Le **module de services Web** permet aux avatars de communiquer entre eux et avec le monde externe dans le respect des standards du Web. Ainsi, les avatars peuvent : interagir avec la plateforme WoT pour interroger des dépôts de code, répondre aux demandes des clients concernant les fonctionnalités qu'ils exposent en tant que ressources RESTful, échanger des données avec d'autres avatars pour réaliser des fonctionnalités collaboratives et interroger des services Web externes pour enrichir leurs propres données.

Le **module de fonctionnalités locales** gère les *fonctionnalités* de haut niveau réalisables à l'aide des capacités atomiques de l'objet et de leur composition. Il s'appuie sur des technologies sémantiques pour lier la couche physique (capacités) avec la couche applicative (fonctionnalités) de manière déclarative et faiblement couplée, en garantissant l'interopérabilité des applications pour les objets (Mrissa et al., 2014). Lorsque l'avatar est créé, le CapabilityManager interroge le module d'interopérabilité pour connaître les fonctionnalités de l'objet et l'ontologie des fonctionnalités de la plateforme pour obtenir leurs descriptions sémantiques. Il est interrogé par le LocalFunctionalityManager, qui charge également les descriptions de fonctionnalités et utilise le raisonneur pour déduire les fonctionnalités locales de l'avatar. Pour chaque fonctionnalité inférée, le LocalFunctionalityManager interroge le gestionnaire de contexte pour déterminer si elle doit être exposée aux clients. Les fonctionnalités exposées sont liées à un registre, afin de permettre leur découverte.

Le **module de Collaboration** gère les fonctionnalités nécessitant une collaboration entre plusieurs avatars. Le CollaborativeFunctionalityDiscoveryManager interroge le raisonneur

comme décrit ci-dessus pour identifier, à partir des fonctionnalités locales, celles auxquelles il pourrait participer aux niveaux supérieurs. Ensuite, il interroge le répertoire des fonctionnalités de la plateforme pour rechercher les fonctionnalités manquantes localement. Si de telles fonctionnalités sont disponibles auprès d'autres avatars, il charge le CollaborativeAgentManager, qui gère la négociation avec ces autres avatars, de mettre en oeuvre ces fonctionnalités collaboratives (Jamont, 2016).

Le **module d'Application WoT** fournit et contrôle des « conteneurs d'applications WoT » qui exécutent des modules de code implémentant les différents aspects d'une application (voir ci-dessous). De tels conteneurs peuvent être répliqués sur l'objet, sur la passerelle et sur l'infrastructure cloud grâce au gestionnaire de déploiement, afin que les modules soient exécutés à l'emplacement approprié.

3.6 Plateforme orientée-avatar

Les solutions IoT existantes peuvent être utilisées en tant que couche de support pour connecter les objets ¹². Les plateformes WoT sont déployées en tant que « serveurs d'applications WoT » au-dessus de ces solutions. Pour permettre le déploiement et l'exécution d'applications WoT sur des environnements informatiques ubiquitaires, les plateformes WoT doivent fournir un accès à des installations de stockage d'informations et de connaissances et à une puissance de calcul supplémentaire dans la couche Cloud. Dans la mesure où nous promovons ici les plateformes WoT basées sur des avatars, elles devraient également prendre en charge la gestion, l'exécution et la (dé)serialisation des avatars. Le mécanisme de sérialisation permet un passage à l'échelle horizontale en répliquant les plateformes et en déplaçant les avatars entre elles à travers les couches Fog et Cloud. La figure 3 décrit l'infrastructure de telles plateformes, où on retrouve instancié le patron architectural décrit en Section 3.3 ¹³.

Les principaux éléments de cette infrastructure sont le contrôleur d'infrastructure et l'environnement d'exécution WoT. Le premier interagit avec la plateforme IoT et est chargé de décider de créer ¹⁴, de mettre à jour ¹⁵ ou de supprimer des avatars, au fur et à mesure que les objets sont connectés et déconnectés de la couche IoT. Le second est le conteneur qui isole les avatars et gère leur cycle de vie. Pour des raisons de performances, il est également connecté à la passerelle, de sorte que les avatars puissent directement interagir avec les objets via Fog computing, sans traverser la pile de la plateforme IoT à chaque requête. À l'intérieur du conteneur, les avatars peuvent également : accéder aux ressources Web externes via le proxy Web, partager des informations sur les fonctionnalités exposées par chaque avatar à l'aide du registre de fonctionnalités, interroger les différents dépôts pour accéder aux descriptions sémantiques dont ils ont besoin pour fonctionner, et récupérer

12. Nous avons développé une *Couche d'Interopérabilité Matériels-Applications* (CIMA) sur ce principe, à partir de la plateforme IoT OM2M (Alaya et al., 2014) : <https://github.com/ucbl/CIMA>

13. La Gateway est une passerelle, élément commun de la couche Fog.

14. Chaque avatar est construit de manière à pouvoir accéder à l'objet auquel il se rapporte à travers la passerelle. Les capacités de l'objet sont injectées et les composants de l'avatar instanciés.

15. En requêtant périodiquement la plateforme IoT.

le code des modules applicatifs dans le dépôt de code. Les programmes *Device Installer* et *Application Installer* sont chargés d'alimenter les différents référentiels et sont indépendants de la notion d'avatar. Les utilisateurs interagissent en toute sécurité avec la plateforme pour télécharger des pilotes d'objets, installer des applications WoT à partir d'un « marketplace » en ligne ou les exécuter à l'aide de leur navigateur Web, via le proxy qui bloque les demandes entrantes non identifiées.

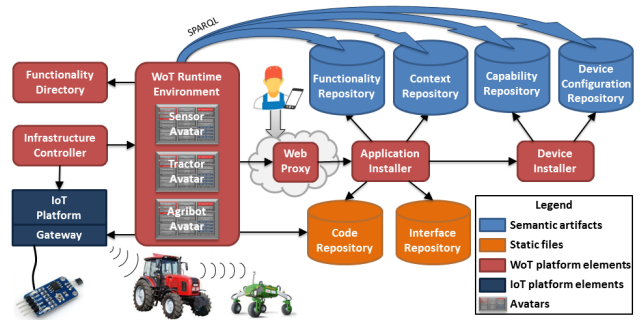


FIGURE 3 – Architecture d'une plateforme orientée-avatar pour le WoT.

3.7 Applications WoT

Afin de faciliter la conception des applications WoT et de conserver une indépendance vis-à-vis des caractéristiques des objets disponibles, une application WoT ne traite que la couche fonctionnalité. Elle décrit principalement une hiérarchie de fonctionnalités, dont les nœuds terminaux sont des fonctionnalités terminales (c-à-d. devant être implémentées par une capacité de l'objet ¹⁶), et tous les autres nœuds sont des fonctionnalités composées (i.e. qui nécessitent des sous-fonctionnalités et les interrogent à l'aide de modules applicatifs). Certaines de ces fonctionnalités composées peuvent nécessiter d'utiliser les capacités de plusieurs objets et, par conséquent, une collaboration entre plusieurs avatars. La fonctionnalité de niveau supérieur correspond alors à l'application que l'utilisateur final souhaite utiliser. Une application WoT est packagée dans un fichier compressé, composé de : un fichier *Manifest*, décrivant son contenu ; la hiérarchie de fonctionnalités susmentionnées ; les modules de code correspondant aux algorithmes implémentant les fonctionnalités composées ¹⁷ ; le modèle de contexte de l'application, contenant une description sémantique du domaine d'application et un ensemble de règles d'adaptation (Terdjimi et al., 2016) ; un ensemble de fichiers statiques constituant l'interface de l'application et permettant aux utilisateurs finaux d'exécuter et de contrôler l'application via leur navigateur

16. Les capacités sont sémantiquement liées à ces fonctionnalités terminales lors de l'initialisation de l'avatar.

17. L'exigence d'interopérabilité impose à ces applications d'être décrites de manière générique, de manière à pouvoir être déployées et exécutées sur différentes configurations. Par conséquent, nous recommandons de les décrire à l'aide de langages déclaratifs, tels qu'une machine à états finis, et lors de l'installation, de les pré-transpiler dans des langages exécutables sur l'objet, sur la passerelle et sur la plateforme cloud sur lesquels ils peuvent être déployés.

Web en interrogeant les fonctionnalités des avatars à l'aide de standards du Web (ressources RESTful, WebSockets, etc.).

4 Problématiques émergentes

Représenter les données d'une manière qui soit à la fois interopérable et adaptée aux contraintes de l'IoT n'est pas le seul défi auquel fait face le SWoT : l'ensemble des technologies du Web Sémantique est concerné par ce besoin d'adaptabilité. La Section 3 donne l'architecture dans laquelle les composants du SWoT évoluent, ainsi que les fonctionnalités qui y sont regroupées par la notion d'avatar. Cette section montre les défis que représentent l'intégration du SWoT dans cette architecture. En particulier, différents aspects des objets de l'IoT amènent des problématiques dont la communauté du SWoT s'empare peu à peu. C'est notamment le cas de leur nature fortement contrainte, l'aspect éclaté des réseaux qu'ils forment, la forte contextualité des applications qui y sont liées, et les problèmes de passage à l'échelle inhérentes à leur déploiement.

4.1 Comblent l'écart entre ontologies et objets

Les ontologies sont des vocabulaires riches, et celles que nous avons identifiées sont toutes représentées en RDF¹⁸. Basé sur des URI comme identifiants de ressource, le RDF est un langage dont certaines sérialisations peuvent s'avérer assez verbeuses, ce qui implique une consommation d'espace mémoire et de bande passante, qui sont des ressources limitées pour les objets connectés. De ce fait, pour pouvoir manipuler des données sémantisées dans les couches Fog et Cloud, il est nécessaire de considérer les contraintes de la couche Device. Cette contrainte peut être intégrée aux modules de communication ou de filtrage des avatars. Nous identifions deux approches principales : faire en sorte que les objets puissent directement manipuler le RDF, ou assurer une traduction entre RDF et des formats plus adaptés aux contraintes des objets.

Amener le RDF aux objets.... Afin de rendre possible la manipulation du RDF par les objets, il est nécessaire de proposer des syntaxes RDF plus compactes que les syntaxes les plus répandues, comme le RDF/XML ou la notation N3. (Su et al., 2018) propose un panorama des sérialisations RDF dans l'optique de les manipuler dans un environnement contraint. Par exemple, dans (Charpenay et al., 2018d), les auteurs proposent une sérialisation binaire de JSON-LD afin de permettre l'échange plus efficace des données dans le contexte d'objets contraints. De plus, ce travail est étendu dans (Charpenay et al., 2018b), afin de s'appuyer sur ce format pour stocker et manipuler les bases de connaissance en RDF directement sur les objets connectés. De plus, ces travaux prennent aussi en compte les contraintes en terme de protocoles des communications entre les couches Fog et Device : les protocoles du Web Sémantique, comme HTTP et SPARQL, reposent sur TCP, qui demande l'établissement d'une connexion, ce qui peut être coûteux en bande passante. De la même façon, (Loseto et al., 2016) étend la spécification de Linked Data Platform (LDP)¹⁹, une recommandation du

W3C qui décrit un protocole d'exposition de données liées, potentiellement en RDF. Dans la recommandation originale, LDP est associé aux verbes HTTP, que les auteurs ont remplacé par leur équivalent CoAP²⁰, un protocole construit sur UDP, spécialement conçu pour les échanges dans la couche Device.

... ou assurer la traduction depuis et vers le RDF. Même si des technologies émergent pour rendre possible la manipulation de RDF sur des objets contraints, un grand nombre d'objets et de services déjà en circulation n'ont pas cette capacité. Pour pouvoir intégrer cette masse d'objets au SWoT, il est donc nécessaire d'établir une transformation entre les formats de données spécifiques à l'IoT et le RDF. Transformer des données vers du RDF est une problématique dont la communauté du Web Sémantique s'est déjà largement emparé : on parle dans ce cas d'**enrichissement**. Cependant, les objets de l'IoT ne sont pas uniquement des producteurs de données, ils en sont aussi des consommateurs, par exemple dans le cas des actionneurs, ou pour la configuration. Il est donc non seulement nécessaire d'enrichir les données issues des objets, mais aussi d'effectuer la transformation inverse, à laquelle on pourrait faire référence comme un **appauvrissement**. En effet, la transformation du RDF vers un format moins expressif s'accompagne d'une perte de contexte, qui n'est plus explicite, mais qui correspond au contexte implicite dans lequel la donnée ainsi appauvrie sera consommée par l'objet destinataire de la communication. La problématique de la traduction bidirectionnelle entre RDF et formats contraints a été abordée dans divers travaux récents. Dans (Seydoux et al., 2016b), l'approche proposée se destine principalement aux formats arborescents aux schémas explicites, de type XML ou JSON. À partir d'associations entre les balises du langage et les ressources d'un graphe RDF, le système effectue la transformation de manière semi-automatique. Il est particulièrement adapté aux langages formalisés par un standard, dans lequel le schéma est fixé. Les travaux de (Charpenay et al., 2018c) visent eux aussi à assurer la traduction entre différents modèles standards en passant par l'enrichissement. Dans ce cas, la traduction n'est considérée que dans le sens de l'enrichissement. La méthode vise à offrir de l'interopérabilité entre les standards en construisant de manière semi-automatique des règles de traduction d'un modèle enrichi à l'autre. Dans (Lefrançois, 2018), une approche plus souple est proposée en s'ancrant sur les principes de l'architecture du Web : les agents sur le Web, potentiellement contraints, s'échangent des représentations de graphes RDF sous formes de flux d'octets typés par des mediatypes, comme l'illustre la Figure 4. Ce cadre conceptuel, nommé RDF Presentation, définit les termes RDF **lowering** (abaissement, étape 1 sur la Figure 4) et **lifting** (élévation, étape 3 sur la Figure 4) au lieu de enrichissement et appauvrissement, étant donné que les représentations RDF sont supposées encoder toutes les caractéristiques du graphe RDF. À partir d'en-têtes HTTP, d'options CoAP, ou d'autres mécanismes, la source ou le destinataire de la communication peut indiquer les règles de transformations entre le format échangé et RDF. Un exemple de langage expressif d'enri-

18. <https://www.w3.org/RDF/>

19. <https://www.w3.org/TR/ldp/>

20. <http://coap.technology/>

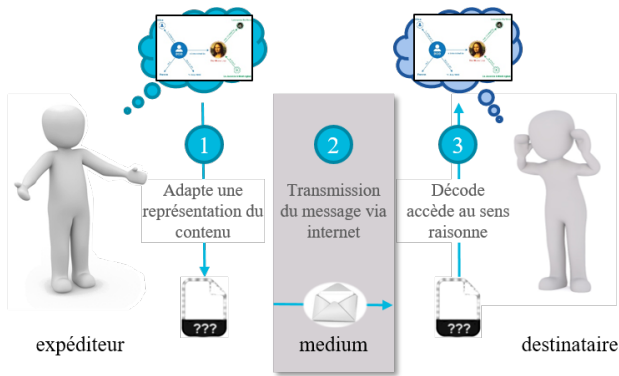


FIGURE 4 – RDF Presentation : utilisation de transformations pour échanger du RDF sur le web (Lefrançois, 2018)

chissement ou d'élévation est SPARQL-Generate²¹ (Lefrançois et al., 2017), qui est une extension de SPARQL 1.1 qui permet de requêter à la fois du RDF et des documents dans des formats hétérogènes (XML, JSON, CBOR, CSV, HTML, GeoJSON, CBOR, ...) ou des flux de tels documents (WebSocket, MQTT).

4.2 Distribuer le raisonnement

Les technologies du Web Sémantique ont initialement été déployées sur des machines du Cloud pour leur permettre de traiter des données issues d'objets connectés, sans être limitées par leurs contraintes. Le développement du Fog computing offre de nouvelles ressources intégrées aux réseaux d'objets connectés, avec une couche Fog mobile et plus proches de objets. Bien que les nœuds Fog soient largement moins puissants que les serveurs du Cloud, ils peuvent cependant supporter dans une mesure raisonnable les technologies du Web Sémantique sans adaptation particulière de protocole ou de format de donnée. De ce fait, la couche Fog est un candidat privilégié dans le rôle de médiateur entre les formats et les protocoles les plus répandus du Web Sémantique, et ceux spécifiques à la couche Device de l'IoT. La traduction proposée dans la Section 4.1 est de ce fait souvent poussée vers les nœuds Fog, qui ont le double avantage de pouvoir communiquer avec les objets connectés, et de gérer un nombre limité d'objet. Il est donc possible pour les passerelles Fog d'utiliser leur connaissance du contexte et des objets auxquels elles sont connectées pour enrichir les données issues des objets, capturant dans la représentation de ces données leur contexte de collecte. Ainsi, les données peuvent être enrichies sémantiquement à partir de leur entrée dans la couche Fog. Le découpage des avatars en modules de fonctionnalité, et la mobilité de ces modules, permet d'envisager leur distribution à travers l'architecture de référence. En particulier, la description des données à partir d'ontologies permet le déploiement de traitements tirant parti du formalisme ainsi introduit, en particulier du raisonnement. Un raisonneur peut, à partir d'une base de connaissance, identifier des inconsistances ou dériver de nouvelles connaissances. La production de nouveaux triplets peut se baser sur des règles d'inférence génériques capturées

dans les langages de modélisation comme OWL, mais peut aussi être décrite par des règles d'inférence métier, décrites dans des langages spécifiques. En particulier, SHACL²² est la recommandation la plus récente du W3C en matière de représentation de contrainte sur des graphes RDF, et une de ses extensions porte sur la représentation de règles²³. En s'appuyant sur des requêtes SPARQL CONSTRUCT, SHACL permet d'écrire des règles d'inférence menant à la production de nouvelles connaissances. Une fois sérialisées, les règles de déduction peuvent être échangées, rendant possible la distribution du raisonnement, notamment à travers la couche Fog. Dans (Seydoux et al., 2018a,b), les applications consommant des données issues d'un réseau d'objets sont représentées sous la forme d'une collection de règles, lesquelles sont ensuite propagées de proche en proche entre les nœuds du réseau dans une approche nommée EDR. La propagation est complètement décentralisée, et elle est guidée par une définition de politique directement embarquée dans la règle, rendant l'approche complètement flexible. Un exemple de politique visant à amener les règles au plus près des objets est détaillé dans (Seydoux et al., 2018b). Les nœuds échangent entre eux des informations concernant les types de données qu'ils produisent, et les règles sont annotées pour identifier les types de données qu'elles consomment. Les nœuds transmettent la règle aux autres nœuds produisant l'ensemble des données qu'elle consomme, jusqu'à arriver à un point critique, qui est l'ancêtre commun le plus proche des différents producteurs de données nécessaires à la règle. Pour assurer un placement des règles adaptatif aux évolutions du réseau, les mises à jour de la topologie sont propagées dynamiquement par les nœuds, qui remettent en permanence en question la distribution des règles. EDR suppose une disposition hiérarchique des nœuds du réseau en un graphe acyclique, que l'on retrouve dans le patron présenté en Section 3.3. La distribution du traitement permet de réduire la quantité d'information à traiter par chaque nœud appliquant la règle, amenant ainsi à des temps de raisonnement plus court, réduisant ainsi le délai entre l'observation d'un ensemble de signes caractérisant un phénomène et sa prise en compte par une application de haut niveau. De plus, distribuer les règles permet de contextualiser leur application, en ne les considérant que dans les sous-ensemble du réseau dans lequel elles sont pertinentes.

4.3 Distribuer les données

Par nature, les données issues des réseaux d'objets sont fortement liées à un contexte spatio-temporel, exprimant des observations n'ayant de valeur de vérité que si elles sont considérées pour un lieu et un instant donné. Ces données sont produites par des sources très distribuées géographiquement, lesquelles sont potentiellement opérées par des entités indépendantes les unes des autres. De plus, quand bien même les données sont enrichies, soit directement par les objets par l'utilisation des formats spécifiques (*c.f.* Section 4.1) soit par l'entremise de nœuds tiers (*c.f.* Section 4.1), les ontologies utilisées par les

21. <https://w3id.org/sparql-generate>

22. <https://www.w3.org/TR/shacl/>

23. <https://www.w3.org/TR/shacl-af/>

différentes parties ne sont pas nécessairement les mêmes, ou alignées entre elles. Pour permettre l'accès aux données, deux axes d'approche sont possibles : leur duplication, ou leur fédération.

La duplication de données. Dans certains cas, les données issues de sources multiples sont stockées par une entité intermédiaire qui les expose ensuite aux requêtes client. Les données sont donc potentiellement toujours disponibles dans leurs sources respectives, et elles sont dupliquées par l'intermédiaire, ce qui fait que le client interroge un interlocuteur unique pour accéder à des données issues de sources multiples. Ce modèle est instancié par le projet FIESTA-IoT, un projet H2020 visant à exposer via une plateforme unique des données liées issues d'un ensemble hétérogène de sources, telles que des villes intelligentes ou des bâtiments instrumentés. Toutes les sources ne produisant pas leurs données en RDF, n'étant pas nécessairement accessibles en continu, et ne stockant pas nécessairement d'historique, les clients viendront directement interroger la plateforme FIESTA-IoT. Celle-ci se propose donc de collecter les données dans leur format original, de les enrichir à l'aide de leur propre vocabulaire, et de les stocker dans son propre espace mémoire pour fournir un historique aux utilisateurs. FIESTA-IoT est donc hébergée sur un nœud Cloud. Dans le cas de systèmes produisant déjà leurs données en RDF, là aussi la plateforme vient les collecter et les stocker dans son propre espace, à condition que celles-ci soient annotées avec l'ontologie prévue par la plateforme. Le problème de l'interopérabilité est donc résolu par l'utilisation d'un vocabulaire unique au sein de la plateforme. Cependant, les sources de données peuvent tout à fait continuer leur activité de manière indépendante après leur intégration à la plateforme. C'est par exemple le cas du bâtiment ADREAM, au LAAS-CNRS, dont les données sont enrichies pour être publiées sur un portail de données ouvertes²⁴, en plus d'être publiées dans la plateforme FIESTA annotées avec un vocabulaire différent.

La fédération de données du SWoT. Dans le cas de requêtes fédérées sur un ensemble de sources de données, contrairement au cas précédent, les données ne sont pas dupliquées. C'est le moteur de requête qui va agréger les données issues des différentes sources, afin de fournir une réponse à l'utilisateur. Ce type d'approche ne pose pas le problème de la fraîcheur des données qui peut être observé dans le cas de duplication qui ne soit pas faite en continu, mais par lots de manière ponctuelle. Dans les requêtes fédérées, les sources de données originales sont accédées au moment de la requête. Une telle approche, adaptée au SWoT, est proposée dans (Jaiswal and Lefrançois, 2017). Les auteurs proposent d'étendre la clause SERVICE des requêtes SPARQL, permettant l'interrogation fédérée, afin d'y intégrer les fonctionnalités de SPARQL-Generate. De cette façon, toute source de donnée, même non-RDF, peut être intégrée à une requête fédérée, à la seule condition d'exposer les données à travers un point d'accès HTTP. Cependant, les requêtes fédérées ont aussi leurs limitations, et le choix de l'un ou l'autre des modèles dépend des besoins applicatifs. En particulier, une requête fédérée sera

toujours aussi lente que la plus lente des sources de données interrogée, ce qui peut être un problème dans le cas de données directement exposées par des objets parfois peu fiables. De plus, interroger l'ensemble des sources de données à chaque requête peut générer un trafic réseau (et donc une consommation énergétique) plus importante que d'interroger un dépôt centrale dans lequel les données sont stockées après une unique collecte.

4.4 Découvrir les objets et les services

Les réseaux IoT sont des réseaux dynamiques, dans lesquels les changements de topologie dus à des déplacements, des connexions et des déconnexions de nœuds sont fréquents, notamment dans les couches Fog et Device. Cette accessibilité intermittente des entités amène la question de la découverte : comment, à un instant donné, identifier les nœuds offrant un service utile à une application ? Il est important de remarquer dans ce cas le rôle double des ontologies mises en jeu : non seulement elles décrivent les données métiers manipulées par les objets, mais aussi les objets eux-même. La distribution des données évoquée dans la Section 4.3 peut donc à la fois porter sur les données collectées ou sur la description des objets. Ce second cas est au cœur de la contribution de (Charpenay et al., 2018a). Le cas d'utilisation que présentent les auteurs de ce papier repose sur la description d'objets à partir de SOSA/SSN, et de Thing Description, les deux vocabulaires de référence du W3C. Chaque objet est dépositaire de sa propre description, et pour découvrir l'ensemble des objets et de leurs relations, le client envoie une requête en mode broadcast à l'ensemble du réseau. Il collecte les différentes descriptions envoyées dans les réponses à cette requête, les agrège, et exécute l'étape de découverte en attribuant une description sémantique aux nœuds physiques. La question de la découverte des objets peut aussi être comprise dans les protocoles standards. C'est par exemple le cas dans oneM2M, qui définit une procédure de découverte par laquelle un client peut récupérer l'ensemble des ressources connues par un serveur et dont la description est validée par une requête SPARQL.

4.5 Respect de la vie privée

Le développement des réseaux d'objets connectés, et la présence de plus en plus pervasive d'objets dans notre environnement pose d'évidentes questions de respect de la vie privée. Au-delà des questions de sécurisation des objets eux-même, qui est déjà une problématique en soi, les multiples fuites de données récentes amènent à questionner la pertinence de la concentration de masses de données sensibles dans un entrepôt centralisé. C'est pourquoi les approches réparties, très représentées dans les problématiques émergentes (c.f. Section 4), sont particulièrement intéressantes. En effet, la distribution du traitement amène un changement de paradigme : les utilisateurs finaux, qui sont ceux qui produisent les données personnelles consommées par leurs fournisseurs de services, peuvent dans ce cas garder la maîtrise de leurs données. Les traitements offerts par les fournisseurs de services sont dans ce type d'approche décorrélés des serveurs contrôlés par ces même fournisseurs, et ne reposent plus sur le transfert de propriété des données par l'utilisateur. En diminuant la concentration des données, le risque de fuite

24. <https://syndream.laas.fr>

massive est réduit, ainsi que l’attrait pour un potentiel attaquant. Les ontologies comme acl²⁵, ou les travaux comme ceux de (Daga et al., 2015), dans lequel les auteurs automatisent la vérification de la conformité des licences d’utilisation des données tout au long de leur processus d’intégration, sont des pistes par lesquelles les principes et les technologies du Web Sémantique peuvent accompagner la distribution du SWoT pour renforcer le respect de la vie privée.

5 Conclusion

L’émergence du SWoT représente un nouveau champ scientifique et technologique où les principes et les technologies du Web Sémantique peuvent représenter un apport majeur. Apporter une réponse aux problèmes d’interopérabilité dus à l’hétérogénéité des objets et des domaines d’application de l’IoT est le moteur du développement du SWoT. Ce besoin explique l’émergence de plusieurs ontologies et standards visant à faciliter l’interaction entre systèmes. Les déploiements d’objets connectés, de par les contraintes qui leurs sont propres, tendent à reproduire un patron architectural en trois niveaux, Cloud-Fog-Device. Dans ce patron architectural, les composants logiciels permettant le contrôle des objets, que l’on peut structurer sous forme d’avatars, sont distribués selon les besoins et les capacités des nœuds qui les supportent. Le fait que les nœuds composant les réseaux IoT aient des capacités si diverses, et en particulier les objets connectés et les fortes contraintes, est l’aspect qui demande le plus d’adaptation des technologies du Web Sémantique pour développer le SWoT. Pour être manipulables par les objets, le RDF doit être exprimé dans des syntaxes plus compactes et plus légères. Il est aussi nécessaire de prendre en compte l’incapacité de certains objets à manipuler du RDF, et d’assurer dans ce cas une traduction dans les couches Fog ou Cloud. La distribution dynamique du raisonnement et des données est aussi un aspect sur lequel des contributions de la communauté amènent un éclairage nouveau. L’adoption de vocabulaires et technologies du Web Sémantique dans des standards issus de communautés dans lesquelles ils n’étaient pas d’habitude intégrés est signe de leur démocratisation toujours plus large. Les contraintes de l’IoT, et les innovations qu’elles amènent dans le développement du SWoT, bénéficient au Web Sémantique tout entier, en faisant considérer à la communauté des défis auxquels elle doit faire face, mais aussi en lui donnant accès à une masse de données dont l’exploitation ne peut être que très riche. Les problématiques émergentes identifiées dans ce papier sont autant de domaines d’intérêt pour la communauté IC, et promettent d’intéressants développements dans le futur.

Références

Alaya, M. B., Banouar, Y., Monteil, T., Chassot, C., and Drira, K. (2014). OM2M : Extensible ETSI-compliant M2M Service Platform with Self-configuration Capability. In *the 5th Int. Conf. on Ambient Systems Networks and Technologies*.

Ashton, K. (2009). That ‘internet of things’ thing. *RFID journal*, 22(7).

Ben-Alaya, M., Medjiah, S., Monteil, T., and Drira, K. (2015). Toward semantic interoperability in oneM2M architecture. *IEEE Communications Magazine*, 53(12).

Berners-Lee, T., Hendler, J., and Lasilla, O. (2001). The Semantic Web. *Scientific American*, 284(5).

Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. (2012). Fog Computing and Its Role in the Internet of Things. In *Proc. of the MCC workshop on Mobile cloud computing*.

Charpenay, V., Käbisich, S., and Kosch, H. (2018a). A framework for semantic discovery on the web of things. In *Emerging Topics in Semantic Technologies, ISWC 2018 Satellite Events*.

Charpenay, V., Käbisich, S., and Kosch, H. (2018b). μ RDF Store : Towards Extending the Semantic Web to Embedded Devices. In *ESWC*.

Charpenay, V., Käbisich, S., and Kosch, H. (2018c). Semantic data integration on the web of things. In *IOT 2018*.

Charpenay, V., Sebastian, K., and Kosch, H. (2018d). Towards a Binary Object Notation for RDF. In *ESWC 2018*.

Compton, M., Henson, C., Lefort, L., Neuhaus, H., and Sheth, A. (2009). A survey of the semantic specification of sensors. In *SSN 2009*, volume 522.

Daga, E., D’Aquin, M., Gangemi, A., and Motta, E. (2015). Propagation of Policies in Rich Data Flows. In *K-CAP 2015*.

Daniele, L., den Hartog, F., and Roes, J. (2015). Created in close interaction with the industry : the smart appliances reference (saref) ontology. In *FOMI 2015*.

Dastjerdi, A. V. and Buyya, R. (2016). Fog Computing : Helping the Internet of Things Realize Its Potential. *Computer*, 49(8).

Galloway, B. and Hancke, G. P. (2013). Introduction to industrial control networks. *Communications Surveys & Tutorials, IEEE*, 15(2).

Haller, A., Janowicz, K., Cox, S. J., Lefrançois, M., Taylor, K., Le Phuoc, D., Lieberman, J., García-Castro, R., Atkinson, R., and Stadler, C. (2018). Sosa : A lightweight ontology for sensors, observations, samples, and actuators. *Semantic Web Journal*.

Haller, A., Janowicz, K., Cox, S. J. D., Le Phuoc, D., Taylor, K., and Lefrançois, M. (2017). Semantic Sensor Network Ontology. W3C Recommendation, W3C.

Jaiswal, S. and Lefrançois, M. (2017). Towards federated queries for web of things devices. In *Workshop on Semantic Interoperability and Standardization in the IoT, SIS-IoT*.

25. www.w3.org/ns/auth/acl

- Jamont, J., Médini, L., and Mrissa, M. (2014). A Web-Based Agent-Oriented Approach to Address Heterogeneity in Cooperative Embedded Systems. In *PAAMS 2014*, volume 293 of *Advances in Intelligent Systems and Computing*.
- Jamont, J.-P. (2016). Multi-agent approach, models and tools to collective cyber-physical system engineering. Habilitation thesis, université grenoble alpes.
- Janowicz, K., Haller, A., Cox, S. J., Le Phuoc, D., and Lefrançois, M. (2018). Sosa : A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*.
- Kaebisch, S., Kamiya, T., McCool, M., and Charpenay, V. (2019). Web of Things (WoT) Thing Description. Candidate Recommendation, W3C.
- Kovatsch, Matthias and Matsukura, R., Lagally, M., Kawaguchi, T., Toumura, K., and Kajimoto, K. (2019). Web of Things (WoT) Architecture. Candidate Recommendation, W3C.
- Lee, E. A. (2008). Cyber physical systems : Design challenges. In *ISORC 2008*. IEEE.
- Lefort, L., Henson, C., and Taylor, K. (2011). Semantic Sensor Network XG Final Report. W3C Incubator Group Report, W3C.
- Lefrançois, M. (2017). Planned ETSI SAREF Extensions based on the W3C&OGC SOSA/SSN-compatible SEAS Ontology Patterns. In *SIS-IoT 2017*.
- Lefrançois, M. (2018). Rdf presentation and correct content conveyance for legacy services and the web of things. In *Proc. of IOT 2018*.
- Lefrançois, M., Zimmermann, A., and Bakerally, N. (2017). A SPARQL Extension for Generating RDF from Heterogeneous Formats. In *ESWC*. Springer, Cham.
- Loseto, G., Ieva, S., Gramegna, F., Ruta, M., Scioscia, F., Sciascio, E. D., and I, B. (2016). Linked Data (in low-resource) Platforms : a mapping for Constrained Application Protocol. In *ISWC*.
- Mahéo, Y., Le Sommer, N., Launay, P., Guidec, F., and Dragone, M. (2012). Beyond Opportunistic Networking Protocols : a Disruption-Tolerant Application Suite for Disconnected MANETs. In *ExtremeCom 2012*, Zürich, Switzerland. ACM.
- Mell, P. and Grance, T. (2011). The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. *National Institute of Standards and Technology, Information Technology Laboratory*, 145.
- Mrissa, M., Médini, L., Jamont, J., Sommer, N. L., and Laplace, J. (2015). An avatar architecture for the web of things. *IEEE Internet Computing*, 19(2).
- Mrissa, M., Médini, L., and Jamont, J.-P. (2014). Semantic Discovery and Invocation of Functionalities for the Web of Things. In *IEEE International Conference on Enabling Technologies : Infrastructure for Collaborative Enterprises*.
- Murdock et al, P. (2016). Semantic Interoperability for the Web of Things. Technical Report October.
- Poveda Villalón, M., Nguyen, Q.-D., Roussey, C., de Vaulx, C., and Chanet, J.-P. (2018). Ontological requirement specification for smart irrigation systems : A SOSA/SSN and SAREF comparison. In *SSN 2018*.
- Sagar, S., Lefrançois, M., Rebai, I., Maha, K., Garlatti, S., Feki, J., and Médini, L. (2018). Modeling smart sensors on top of SOSA/SSN and WoT TD with the semantic smart sensor network (S3N) modular ontology. In *Emerging Topics in Semantic Technologies, ISWC 2018 Satellite Events*.
- Seydoux, N., Alaya, M. B., Hernandez, N., Monteil, T., and Haemmerlé, O. (2015). Sémantique et Internet des objets : d'un état de l'art à une ontologie modulaire. In *IC 2015*.
- Seydoux, N., Drira, K., Hernandez, N., and Monteil, T. (2016a). IoT-O, a core-domain IoT ontology to represent connected devices networks. In *EKAW*.
- Seydoux, N., Drira, K., Hernandez, N., and Monteil, T. (2016b). Lowering knowledge : Making constrained devices semantically interoperable. In *ISWC*.
- Seydoux, N., Drira, K., Hernandez, N., and Monteil, T. (2018a). Towards Cooperative Semantic Computing : a Distributed Reasoning approach for Fog-enabled SWoT. In *COOPIS*.
- Seydoux, N., Khalil, D., Hernandez, N., and Monteil, T. (2018b). A Distributed Scalable Approach for Rule Processing : Computing in the Fog for the SWoT. In *Web intelligence, Santiago, Chili*.
- Sheth, A., Henson, C., and Sahoo, S. S. (2008). Semantic Sensor Web. *IEEE Internet Computing*, 12(4).
- Su, X., Li, P., Riekkki, J., Liu, X., Kiljander, J., Soininen, J.-P., Prehofer, C., Flores, H., and Li, Y. (2018). Distribution of Semantic Reasoning on the Edge of Internet of Things. In *IEEE UbiComp*, number November.
- Szilagyi, I. and Wira, P. (2016). Ontologies and Semantic Web for the Internet of Things - a survey. In *IECON*. IEEE.
- Terdjimi, M., Médini, L., Mrissa, M., and Le Sommer, N. (2016). An Avatar-based Adaptation Workflow for the Web of things. In *WETICE 2016*, Paris, France.
- Wang, F., Hu, L., Zhou, J., and Zhao, K. (2015). A survey from the perspective of evolutionary process in the internet of things. *Int. J. Distributed Sensor Networks*, 2015.