



Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>
Eprints ID : 13105

To cite this version : Arcangeli, Jean-Paul and Chabridon, Sophie and Conan, Denis and Desprats, Thierry and Laborde, Romain and Leriche, Sébastien and Lim, Léon and Taconet, Chantal and Boujbel, Raja and Machara Marquez, Samer and Marie, Pierrick and Rottenberg, Sam
[Gestion de contexte multi-échelle pour l'Internet des objets](#). (2014) In: Journées francophones Mobilité et Ubiquité - UBIMOB 2014, 5 June 2014 - 6 June 2014 (Sophia Antipolis, France).

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

Gestion de contexte multi-échelle pour l'Internet des objets

Jean-Paul Arcangeli¹, Sophie Chabridon², Denis Conan², Thierry Desprats¹
Romain Laborde¹, Sébastien Leriche³, Léon Lim², Chantal Taconet²
Raja Boujbel¹, Samer Machara Marquez², Pierrick Marie¹, Sam Rottenberg²

¹IRIT UMR 5505 Université Paul SABATIER, 31062 TOULOUSE, France

<prénom>.<nom>@irit.fr

²Institut Mines-Télécom/Télécom SudParis, CNRS UMR 5157 SAMOVAR, 91011 Évry, France

<prénom>.<nom>@telecom-sudparis.eu

³ENAC, 31055 TOULOUSE, France

<prénom>.<nom>@enac.fr

RÉSUMÉ

Cet article présente une architecture logicielle de gestion de contexte multi-échelle pour l'Internet des objets. Nous présentons d'abord quelques défis spécifiques auxquels est confrontée cette nouvelle génération d'intergiciels, tels que la communication répartie en modes *push* et *pull*, la sensibilité multi-échelle, et la réponse à des exigences et des garanties de qualité de l'information de contexte et de respect de la vie privée. Nous donnons ensuite la vision du projet INCOME pour répondre à ces défis en fournissant une infrastructure répartie de gestion de contexte et en assurant son déploiement de manière autonome.

ABSTRACT

This article gives an overview of a software architecture able to support the multiscale distribution of context information over the Internet of Things. We first present some specific challenges that must be addressed by this new generation of middleware, like distributed *push* and *pull* communications, multiscale awareness, and both quality of information and privacy requirements and guarantees. Then we give the vision of the INCOME project to answer these requirements by providing an ad hoc context distribution infrastructure and a framework for its autonomic deployment.

Keywords

Middleware, Context Computing, Model Driven Engineering, Autonomic Deployment, Multiscale distributed systems

1. INTRODUCTION

Depuis plus de deux décennies, des solutions ont été proposées pour permettre à l'espace ambiant de percevoir les événements et les changements grâce à une variété de dispositifs de détection de contexte potentiellement mobiles [15]. Plus récemment, en particulier grâce aux progrès des télécommunications sans fil, le paradigme de l'Internet des objets (*IoT* en anglais), proposant de connecter entre eux non seulement des personnes mais aussi des choses ou des objets, connaît une évolution rapide [1].

Dans cet article, nous présentons notre vision des systèmes de gestion de contexte multi-échelle combinant des composants à différentes échelles allant des espaces ambiants locaux ou distants jusqu'aux nuages (*Cloud* en anglais). Cet aspect multi-échelle contribue à enrichir la capacité d'adaptation dont disposeront les nouvelles applications sensibles au contexte. Cette sensibilité n'est plus limitée à l'environnement ambiant qu'elles perçoivent localement, mais s'étend à des informations de contexte recueillies à partir de sources situées dans des espaces distants.

Cette vision multi-échelle des systèmes de gestion de contexte offre de nouvelles opportunités mais pose aussi de nouveaux défis [13]. L'IoT est caractérisé par l'extrême hétérogénéité des objets qu'il peut interconnecter et par la nature spontanée des interactions résultant de l'instabilité et de l'ouverture de l'environnement. Pour maîtriser l'énorme quantité d'informations, potentiellement transitoires, nous considérons que de nouvelles solutions sont nécessaires pour permettre de raisonner à l'échelle qui soit pertinente pour des prises de décisions opportunes et utiles. Les solutions que nous étudions dans le projet INCOME¹ sont intégrées dans une infrastructure de gestion de contexte qui peut être utilisée à des échelles multiples, qui est capable de raisonner à une échelle donnée, et qui permet aussi de communiquer des informations de contexte d'une échelle à l'autre. Les principales caractéristiques de notre proposition concernent i) la distribution de contexte multi-échelle selon les modes *push* (notification) et *pull* (observation) de communication, ii) la gestion conjointe de la qualité des informations de contexte et de la protection de la vie privée, et iii) le déploiement multi-échelle autonome. Les caractéristiques de notre proposition sont respectivement expliquées dans les sections 3 et 4. Elles répondent aux exigences résultant des nouveaux défis présentés dans la section 2.

¹anr-income.fr

2. VERS UNE NOUVELLE GÉNÉRATION DE GESTIONNAIRES DE CONTEXTE

Un gestionnaire de contexte est une entité logicielle capable de calculer des informations de haut niveau à partir de diverses sources de données de contexte. Ses fonctionnalités comprennent l'acquisition de données brutes, le traitement de ces données (fusion, agrégation, interprétation, inférence) et la présentation d'informations de plus haut niveau à des applications sensibles au contexte [6]. Ces applications sont classiquement vues comme des consommateurs finaux d'informations de contexte, alors que les entités qui fournissent des données brutes sont considérées comme des producteurs initiaux. Le composant qui, au sein d'un gestionnaire de contexte, est chargé de la transformation des données de contexte est un processeur de contexte qui est à la fois un consommateur et un producteur, de niveau intermédiaire, de données de contexte.

2.1 Gestion de contexte répartie pour l'IoT

Plusieurs styles d'architectures ont été proposés pour la distribution de flux d'informations de contexte [2]. Les travaux de recherche existants ne concernent essentiellement que les réseaux locaux et les milieux ambiants. Avec l'IoT, la nécessité de répartir les composants de gestion de contexte est renforcée. En effet, de très nombreux objets peuvent être connectés à l'infrastructure de réseau mondial à tout moment au cours de leur cycle de vie et de façon temporaire ou permanente. En outre, certains objets sont situés dans des emplacements fixes tandis que d'autres sont mobiles. Les objets ne sont plus considérés seulement comme des producteurs de contexte. Ils sont des acteurs indépendants qui peuvent également échanger, enregistrer et transformer l'information de manière autonome ainsi que collaborer pour calculer des informations utiles pour des utilisateurs humains sans les impliquer. En ce qui concerne l'utilité des données de contexte, une solution consiste à la décrire au moyen de méta-données de qualité : la qualité du contexte (QoC en anglais) correspond à toute information qui décrit la qualité des données de contexte comme indiqué par la définition originelle de [4]. *Le choix des méta-données pertinentes doit être ouvert et flexible (R1). Les connaissances supplémentaires fournies par ces méta-données doivent également respecter la vie privée des propriétaires de contexte et ne pas divulguer plus d'informations que nécessaire (R2).*

De très petits systèmes (par exemple, les objets du paradigme de l'IoT tels que des essais de minuscules capteurs ayant des capacités limitées de traitement, de stockage, d'énergie et de communication) collaborent avec des systèmes puissants (comme ceux pouvant être trouvés dans le *Cloud*) pour fournir efficacement, à des applications sensibles au contexte, des données collectées dans le monde réel. Certaines de ces coopérations impliquent également des ordinateurs intermédiaires, tels que des appareils mobiles ou des serveurs de proximité. *Une gestion de contexte répartie doit être construite au-dessus de tous ces systèmes et permettre des communications en mode push (notifications) et pull (observations) (R3).*

Faire de l'IoT une réalité contribue à étendre la capacité de sensibilité au contexte de nombreuses applications, qui peuvent ainsi bénéficier de données de contexte obtenues à partir d'espaces distants. À cet effet, l'évolution de l'IoT d'aujourd'hui, composé d'îlots de réseaux de capteurs sans

fil, à une infrastructure mondiale composée de plusieurs infrastructures multi-échelles, est nécessaire. *Le déploiement des entités de gestion de contexte doit être autonome pour gérer la volatilité des producteurs et des consommateurs à toutes les échelles (R4).*

2.2 Gestion de contexte multi-échelle

Dans cette section, nous introduisons le concept de gestion de contexte multi-échelle pour l'IoT. Nous présentons quelques cas d'utilisation et résumons les exigences liées à la capacité multi-échelle identifiées précédemment.

Certains travaux de recherche étudient les systèmes multi-échelles mais considèrent différentes échelles de façon indépendante. Par exemple, le Centre d'étude des systèmes multi-échelles² étudie en trois thèmes distincts les systèmes répartis de collecte d'informations et de contrôle, les systèmes à grande échelle et les systèmes à petite échelle, et aborde ces thèmes séparément. [14] a proposé une architecture hybride combinant les modèles hiérarchique et pair-à-pair, sur la base de l'infrastructure multi-échelle du projet européen GLOSS, mais en se limitant à une répartition géographique. Nous traitons plutôt la capacité multi-échelle en tant que telle, ce qui permet de raisonner à l'échelle pertinente et ainsi de pouvoir changer d'échelle si nécessaire. Cela ouvre la voie à des solutions inédites et efficaces pour la gestion de contexte.

La distribution multi-échelle est un concept différent de la distribution à grande échelle. Le passage à l'échelle possède un sens quantitatif, alors que le multi-échelle s'attache à l'hétérogénéité. L'hétérogénéité d'un système peut provenir par exemple de la différence de temps de latence dans les réseaux concernés, ou de la différence de capacité de stockage des dispositifs concernés. Chaque caractéristique de l'hétérogénéité peut conduire à des problèmes spécifiques en termes de distribution, de performances ou de sécurité.

Comme représenté dans la figure 1, nous considérons, dans le projet INCOME, le multi-échelle selon plusieurs points de vue : l'appareil (p.ex., les objets intelligents, les appareils mobiles et les ordinateurs dans les nuages appartiennent à des échelles de périphériques différents), la géographie (p.ex., ambiant, ville, pays), réseau (p.ex., réseau de capteurs, LAN, WAN) et administratif (p.ex., équipe, filiale, société). Chaque point de vue est associé à des mesures et chaque mesure est associée à un ensemble d'échelles.

Le multi-échelle conduit à deux types d'impact. Tout d'abord, en termes de déploiement, il détermine le choix des dispositifs où les composants de gestion de contexte sont déployés. Ensuite, en termes de gestion de contexte, il influence l'acheminement et la livraison de données de contexte, depuis les producteurs vers les consommateurs. La prise en compte du multi-échelle dans la gestion de contexte est un moyen de réduire le volume global des données transmises sur l'IoT.

Les exemples décrits ci-après illustrent des cas de communication entre producteurs et consommateurs avec conscience de l'échelle. Les consommateurs peuvent souhaiter contraindre l'échelle géographique des données de contexte reçues. Par exemple, un automobiliste peut limiter les informations sur

²www.musyc.org

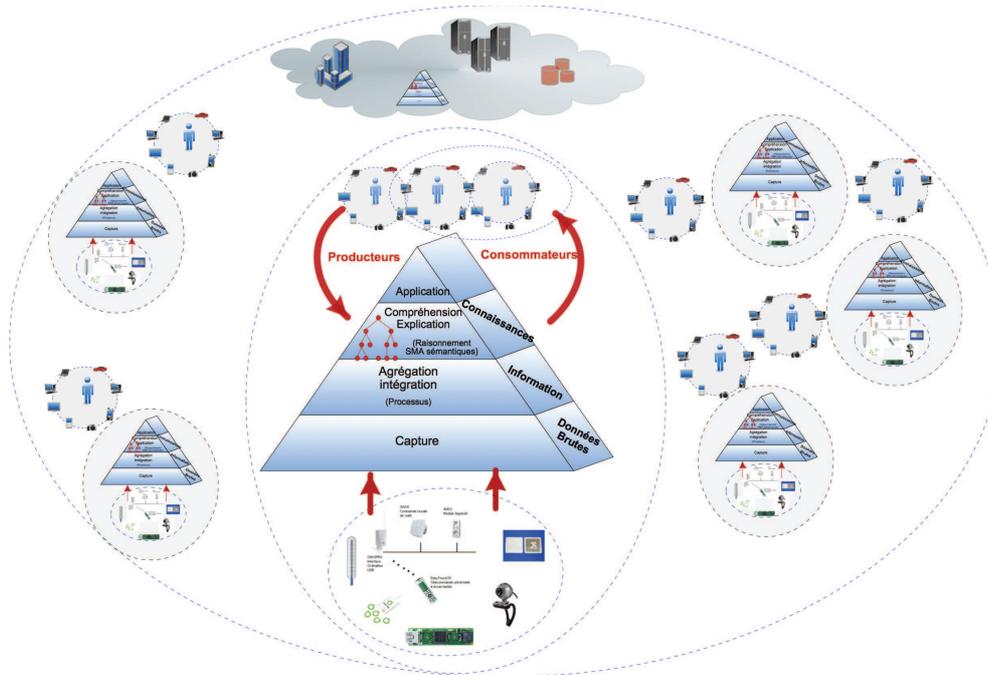


Figure 1. Gestion répartie de contexte pour l'IoT

les parkings disponibles à ceux situés à une certaine distance de marche de sa destination. Les informations concernant des parkings situés au-delà de la distance à pied ne seront pas reçues. La livraison des informations de contexte peut aussi être limitée par les producteurs. Par exemple, un étudiant peut limiter la transmission de sa localisation à des amis qui sont dans la même université que lui. Dans les systèmes de communication inter-véhiculaires, une société de location de voitures peut choisir de limiter la diffusion de la localisation d'un de ses véhicules exclusivement aux autres voitures qui lui appartiennent.

En conclusion, nous identifions la nécessité de *définir des contraintes d'échelle en termes de géographie, d'utilisateur, d'administration, d'appareil et de réseau pour la livraison des données de contexte et pour le déploiement de la gestion de contexte (R5).*

2.3 Synthèse des exigences

La table 1 résume les exigences les plus importantes que nous avons identifiées pour la gestion de contexte multi-échelle. Pour chacune d'elles, la dernière colonne indique la(les) section(s) dans laquelle(lesquelles) nous présentons la solution proposée par le projet INCOME.

3. PRÉSENTATION DE L'ARCHITECTURE

La figure 2 illustre les fonctionnalités et les extra-fonctionnalités de l'infrastructure INCOME. Les fonctionnalités sont classiquement réparties en trois couches : la livraison des données de contexte, le traitement des données de contexte et la présentation de ces données aux applications sensibles au contexte [6]. Cette architecture revisite celle précédemment proposée par [14] en l'étendant pour prendre en compte la répartition des producteurs et des consommateurs de contexte

No	Exigences	Sections
R1	Sensibilité à la QoC : capacité à qualifier l'information de contexte au moyen de méta-données	3.3 et 3.4
R2	Protection de la vie privée : capacité à contrôler comment, quand, où, avec qui et pourquoi une information est partagée	3.3 et 3.4
R3	Modes de communication répartie <i>push</i> et <i>pull</i>	3.2
R4	Autonomie : capacité à être autonome et à s'adapter aux changements du contexte opérationnel	4
R5	Multi-échelle : capacité à couvrir plusieurs échelles dans au moins un point de vue tel que géographie, utilisateur, administration, appareil ou réseau	3.4 et 4

Table 1. Exigences de la gestion de contexte multi-échelle

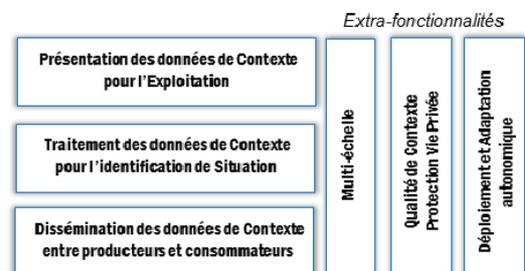


Figure 2. Vue logique de l'architecture INCOME

et les extra-fonctionnalités spécifiquement étudiées dans le projet INCOME.

Dans ce qui suit, nous décrivons l'infrastructure que nous avons conçue pour permettre la distribution multi-échelle de données de contexte entre les composants logiciels qui sont impliqués dans la réalisation des tâches liées à cette architecture fonctionnelle globale (section 3.2) et les relations avec deux des extra-fonctionnalités mentionnées, la sensibilité à la qualité de contexte et au respect de la vie privée (section 3.3) et la prise en compte du multi-échelle (section 3.4). Nous commençons tout d'abord par une présentation générale des principaux composants de cette architecture.

3.1 Principaux composants de l'architecture

La figure 3 illustre les principaux composants de l'architecture INCOME assurant la diffusion de données de contexte entre différentes catégories de producteurs et de consommateurs. Chaque cercle bleu représente un **espace**. Un espace peut être caractérisé par des échelles liées aux capacités de l'infrastructure réseau sous-jacente. Un espace peut également être considéré comme une unité de haut niveau de dissémination de données de contexte entre les composants locaux qu'il inclut. Un composant peut être soit un collecteur, soit une capsule. Chacun est une unité de déploiement.

Un **collecteur de contexte** est un élément qui fournit des données de contexte brutes. Typiquement, un collecteur de contexte encapsule les comportements d'acquisition, en mode *push* ou *pull*, de données brutes issues du monde observé. Par exemple, un collecteur de contexte peut abstraire un client d'un lecteur RFID ou un serveur COAP collectant des données à partir d'un réseau de capteurs. Il est déployé à proximité de l'environnement qu'il observe. Un collecteur de contexte est exclusivement un producteur de données de contexte dans l'infrastructure.

Une **capsule** est un élément fonctionnel du gestionnaire de contexte qui transforme des données de contexte en d'autres données de contexte de plus haut niveau d'abstraction. En offrant un partage de ces données, une capsule joue à la fois les rôles de producteur et de consommateur.

Les **applications sensibles au contexte** sont des clients qui consomment des informations de contexte potentiellement distantes. Ces informations leur permettent de s'adapter dynamiquement alors qu'elles s'exécutent sur les appareils des utilisateurs connectés à l'espace local.

Un **service de courtage intra-espace**, réparti sur l'infrastructure réseau physique sous-jacente, prend en charge une propagation efficace des données de contexte entre toutes les entités actuelles d'un espace. Il assure la livraison des informations de contexte aux consommateurs de contexte.

Chaque espace comprend au moins une capsule passerelle dédiée à transmettre des données de contexte à d'autres espaces distants sur lesquels le gestionnaire de contexte a également été déployé, et *vice versa*. Une capsule passerelle est également reliée à un espace spécifique qui prend en charge le courtage inter-espace. La diffusion est effectuée selon les modes de communication *push* et *pull*.

3.2 Modes de communication répartie *push* et *pull*

Nous considérons un système réparti constitué de **courtiers** reliés entre eux pour construire un réseau de recouvrement. Producteurs et consommateurs de données de contexte participent également au réseau de recouvrement en étant attachés aux courtiers (chaque producteur ou consommateur est attaché à un courtier d'accès). Tous ces composants (courtiers, producteurs et consommateurs) communiquent de manière asynchrone par passage de messages. Dans notre premier prototype de l'architecture de gestion de contexte, la topologie de communication est connexe et peut contenir des cycles. Les composants ne sont pas statiques, mais peuvent rejoindre ou quitter volontairement le réseau de recouvrement en appelant explicitement des opérations spécifiques. Les liens de communication sont considérés comme étant fiables et respectant l'ordre FIFO en utilisant des connexions TCP.

Mode de communication push - Pour le mode de communication *push* ou par notification, nous choisissons le paradigme de communication basé sur les événements répartis [7] avec deux formes de mécanismes de routage à base de filtres : filtrage basé sur le contenu pour la distribution intra-espace et filtrage basé sur le sujet pour la dissémination inter-espace.

Dans le filtrage basé sur le contenu, les filtres sont évalués sur l'ensemble du contenu des messages envoyés : la livraison de messages est basée sur un prédicat défini par le producteur (filtre d'annonce) et un prédicat défini par le consommateur (filtre de souscription). Un filtre d'annonce correspond à un contrat que le producteur doit remplir et le filtre de souscription correspond à une demande exprimée par le consommateur.

Dans le filtrage basé sur le sujet, les consommateurs utilisent la correspondance de chaînes de caractères pour la sélection des messages transmis. Chaque message notifié est complété par une clé de routage définie comme un chemin absolu exprimé à partir de la racine d'un arbre de sujets. Une souscription correspond également à une clé de routage : par exemple, un producteur envoie des données de contexte avec la clé de routage `location.user` et un consommateur s'abonne au filtre `location.*` pour recevoir toutes les données de localisation. Le filtrage basé sur le sujet est beaucoup plus efficace que le filtrage basé sur le contenu, mais il est beaucoup moins expressif.

Pour garantir la sûreté, la configuration locale de routage des courtiers garantit la propriété de validité locale. Cette propriété stipule que seules les données de contexte vérifiant un filtre sont livrées et qu'elles le sont immédiatement. Pour garantir la vivacité, la configuration locale de routage assure la propriété de validité du routage monotone : quand un consommateur s'abonne à un filtre et reste abonné, les données de contexte qui sont transmises à un courtier et qui correspondent au filtre de souscription sont assurées d'être livrées au consommateur.

Lorsque la nécessité de passage à l'échelle est exigée, le filtrage basé sur le sujet est applicable. Pour cette partie, nous nous appuyons sur les intergiciels qui mettent en œuvre le protocole AMQP³. Les courtiers sont généralement déployés dans le *Cloud* ou sur des infrastructures à base de *cloudlets*, et sont capables de gérer de manière efficace les communications

³<http://amqp.org>

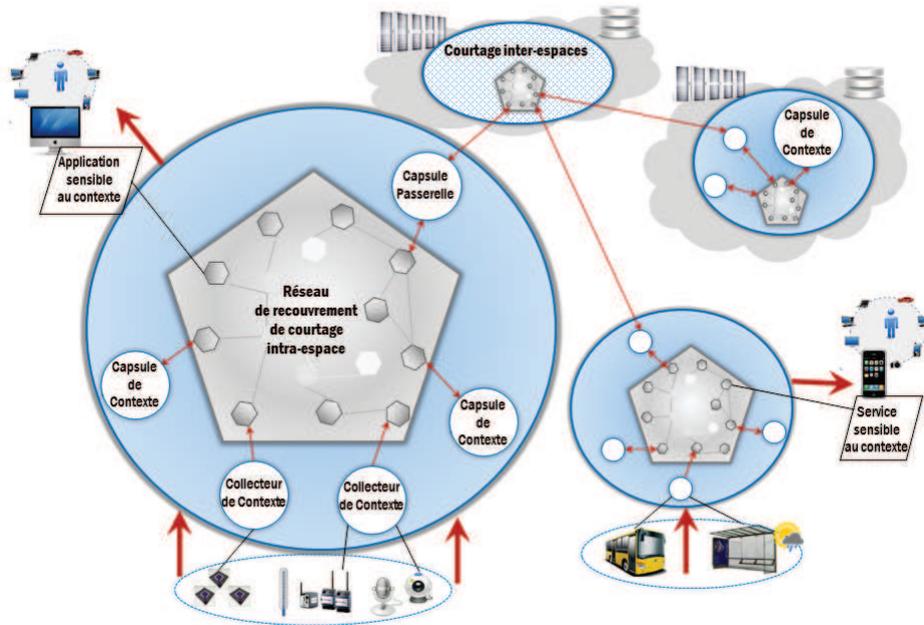


Figure 3. Vue d'ensemble de l'infrastructure INCOME

inter-espace. Une différence importante entre le filtrage basé sur le sujet et celui basé sur le contenu est que ce dernier exige que les données de contexte transmises soient marquées avec des méta-données organisées selon une hiérarchie de sujets, nécessitant un consensus sur ces méta-données et leurs rôles.

Différents modes de fonctionnement de la communication push - Nous proposons deux modes de fonctionnement pour les souscriptions et les annonces : souscription globale avec annonce locale, et souscription locale avec annonce globale. Dans le premier mode de fonctionnement, les annonces sont conservées localement par les courtiers d'accès des producteurs, et les courtiers transmettent les souscriptions à leurs courtiers voisins selon un mécanisme de routage simple : la transmission des annonces au sein du réseau logique des courtiers implique la construction d'un arbre couvrant de sorte que le courtier local du consommateur en est la racine et que les arcs sont dirigés vers cette racine. La taille des tables de routage peut augmenter de manière excessive, mais un tel routage simple minimise le trafic de notification. Dans le second mode de fonctionnement, les abonnements demeurent locaux pour les courtiers d'accès des consommateurs, et les annonces sont diffusées sur le réseau logique des courtiers. Ce mode de fonctionnement est plus efficace lorsque les consommateurs sont très volatiles et que les producteurs sont beaucoup plus stables. Ces deux modes de fonctionnement sont adaptés pour les systèmes à échelle restreinte.

Mode de communication pull - En plus du mode de communication *push*, les applications clientes et les processeurs de contexte peuvent parfois demander à consommer des données de contexte par l'émission de requêtes bloquantes. Ces requêtes correspondent à des observations ou des recherches,

tandis que les messages non sollicités sont des notifications ou des recommandations. Par conséquent, nous fournissons également aux développeurs d'applications sensibles au contexte un mécanisme de requête/réponse anonyme avec réponses multiples pour la communication intra-espace. Une demande d'un consommateur de contexte peut entraîner un nombre non connu *a priori* de réponses par des producteurs de contexte inconnus. L'algorithme pour traiter une telle requête est un algorithme de vague qui agit comme un écho au sein du réseau de recouvrement des courtiers.

3.3 Sensibilité à la QoC et au respect de la vie privée

La vision d'un Internet des objets pose de nouveaux défis pour gérer la qualité de l'énorme quantité d'informations de contexte pouvant être collectées tout en préservant la vie privée des utilisateurs sur lesquels portent les informations recueillies. Nous avons montré dans [5] que les notions de vie privée et de qualité de contexte (QoC) sont interdépendantes et parfois contradictoires nécessitant de les étudier de manière conjointe. En outre, il existe des modèles différents et incompatibles pour spécifier la QoC ou la protection de la vie privée. Nous présentons dans cette section la modélisation introduite par le projet INCOME combinant QoC et modèles de protection de la vie privée.

Modélisation de la QoC - La QoC spécialise la notion plus générale de qualité de l'information (QoI en anglais) pour les informations de contexte. La fraîcheur, la précision, la complétude ou l'exactitude sont des exemples de critères de QoC. Plusieurs travaux de recherche ont porté sur la modélisation et la gestion de la QoC. Dans [11], nous avons comparé les propositions successives de listes de critères de

QoC définis par différents auteurs et nous avons conclu sur la difficulté à converger vers une liste unique et exhaustive de critères. Nous avons donc créé un méta-modèle dédié appelé **QoCIM** (*Quality of Context Information Model* ou Modèle d'Information pour la QoC) qui offre une solution unifiée pour modéliser des méta-données hétérogènes de QoC. QoCIM offre des mécanismes pour :

1. associer des indicateurs de QoC à n'importe quelle donnée de contexte. Chaque indicateur est défini par un critère de QoC et peut être évalué à l'exécution par des valeurs de métriques.
2. spécifier une ou plusieurs définitions pour caractériser les métriques pour l'évaluation d'un critère. Dans une telle définition, des informations telles qu'une description textuelle, une formule de calcul, l'unité, l'ordre des valeurs, les valeurs limites, peuvent être spécifiées.
3. définir des critères composites afin de manipuler un seul indicateur général de QoC à la place d'un ensemble d'indicateurs individuels correspondant chacun à un critère simple.

Par conséquent, QoCIM permet l'exploitation et la manipulation de critères de QoC d'une manière expressive, calculable et générique.

Modélisation du respect de la vie privée - À partir de l'analyse des lois américaines et européennes sur le respect de la vie privée et après une étude des solutions de protection de la vie privée existantes, nous avons identifié un sous-ensemble de quatre dimensions pertinentes pour les données de contexte dans l'IoT, à savoir la finalité (défini dans quel but les données de contexte seront utilisées), la visibilité (indique qui est autorisé à accéder aux données de contexte), la rétention (spécifie combien de temps les données de contexte peuvent être conservées) et la QoC (détermine le niveau de qualité auquel les données de contexte sont fournies). Nous avons proposé dans [10] un cadre de modélisation pour définir les caractéristiques communes de ces dimensions. Nous avons créé trois méta-modèles génériques pour les dimensions *purpose*, *visibility* et *retention*. La dimension QoC repose sur QoCIM ce qui traduit l'interdépendance des préoccupations de QoC et de respect de la vie privée.

3.4 Contractualisation de la sensibilité à la QoC, à la vie privée et au multi-échelle

Le découplage des consommateurs et des producteurs de contexte inhérent à l'IoT nécessite de nouvelles solutions pour le traitement de la QoC et pour protéger la vie privée des personnes concernées par les données de contexte recueillies.

Nous traitons ce découplage en utilisant des contrats où les besoins et/ou les garanties de QoC, vie privée et multi-échelle sont indiqué(e)s. Deux types de contrat de contexte peuvent être définis, l'un du côté du producteur de contexte et l'autre du côté du consommateur de contexte. D'une part, un **contrat de producteur de contexte** comprend des clauses mentionnant : 1) les exigences en matière de protection de la vie privée indiquant les informations personnelles que le producteur accepte de fournir à un gestionnaire de contexte et la façon dont

ces données peuvent être partagées, 2) les garanties de QoC définissant le niveau de qualité des données de contexte que le producteur est en mesure de fournir et 3) éventuellement des exigences multi-échelle identifiant à quelles échelles les données de contexte doivent être livrées. D'autre part, les consommateurs de contexte, comprenant les applications sensibles au contexte et les utilisateurs finaux de contexte, décrivent leurs exigences en termes de QoC et de multi-échelle, et les garanties qu'ils offrent pour le respect de la vie privée des utilisateurs observés dans des **contrats de consommateur de contexte**.

Les exigences et les garanties de respect de la vie privée sont décrites selon les dimensions de *purpose*, *visibility* et *retention* évoquées précédemment. En ce qui concerne les exigences et les garanties de QoC, elles sont définies en suivant le modèle QoCIM. Les exigences concernant le multi-échelle sont décrites en termes de points de vue et de mesures d'échelle (tels que cités dans la section 2.2).

Chaque contrat de contexte est ensuite traduit en un filtre et enregistré dans le composant de communication répartie basée sur les événements de l'infrastructure INCOME (voir la section 3.2). Un contrat de producteur de contexte se traduit en un ou plusieurs filtre(s) d'annonce et un contrat de consommateur de contexte se traduit en un ou plusieurs filtre(s) de souscription. Au moment de la notification, les données contextuelles fournies par un producteur de contexte sont d'abord comparées aux garanties de QoC du producteur et ne sont transmises par les courtiers de messages que si elles remplissent ces garanties. Du côté des consommateurs, les données de contexte sont livrées aux consommateurs ayant des filtres de souscription conformes en termes de respect de la vie privée et de multi-échelle.

4. DÉPLOIEMENT MULTI-ÉCHELLE AUTONOMIQUE

Le déploiement de logiciel [8] est un processus de post-production complexe qui a pour objectif de rendre un logiciel disponible et opérationnel. Cela regroupe des activités étroitement liées telles que l'installation, l'activation, la mise à jour ou l'adaptation. À l'origine, le déploiement de logiciels était géré manuellement. Aujourd'hui, dans des environnements ubiquitaires, répartis, ouverts et hétérogènes, cela exige de l'automatisation et même de l'autonomie. Cette exigence est renforcée dans le cadre du déploiement des composants de gestion de contexte à toutes les échelles de l'IoT.

4.1 Défis du déploiement multi-échelle

Le multi-échelle impacte le déploiement des composants de gestion de contexte de plusieurs façons. L'instabilité et l'ouverture de l'IoT exigent d'abord de gérer un grand nombre de dispositifs hétérogènes, et une diversité de versions de logiciels. En outre, l'environnement peut être très dynamique où la disponibilité et la qualité des ressources et des services peuvent varier. Par conséquent, le déploiement doit faire face à l'apparition et à la disparition d'appareils, à l'arrivée de nouveaux composants logiciels de gestion de contexte ou à la mobilité de ceux déjà présents. Le respect de la vie privée et la gestion de la QoC apportent des exigences supplémentaires que les stratégies de déploiement doivent satisfaire (p.ex., deux capsules de contexte ne peuvent être déployées que sur des appareils appartenant à la même personne). De même, la

gestion de contexte peut exiger le déploiement d'un composant donné sur n'importe quel appareil situé à une échelle particulière (p.ex., sur tous les smartphones d'un réseau Wi-Fi local ou tous les ordinateurs dans une zone géographique donnée).

Le déploiement n'est pas un processus ponctuel en raison de l'évolution dynamique à la fois du système de gestion de contexte et du domaine de déploiement – le domaine de déploiement est l'ensemble des dispositifs en réseau sur lesquels les composants peuvent être déployés. Au contraire, il accompagne l'exécution du logiciel. Ainsi, au-delà de la définition classique de déploiement, nous mettons en évidence deux formes particulières de déploiement. Le **déploiement incrémental** consiste à déployer un nouveau composant dans un système logiciel déjà déployé. Le **déploiement continu** consiste à gérer le déploiement d'un composant sur un dispositif entrant (ou sortant) du domaine. En d'autres termes, le déploiement incrémental gère la dynamique du système logiciel déployé tandis que le déploiement continu gère la dynamique du domaine de déploiement.

4.2 MuScADeL : un langage pour le déploiement multi-échelle

La première étape du processus de déploiement consiste à définir un plan de déploiement, c'est-à-dire une mise en correspondance des composants à déployer et des dispositifs (ou sites) de déploiement. Or, exprimer cette correspondance peut être complexe et fastidieux pour des raisons de nombre et de diversité. De plus, les domaines étant ouverts, tous les sites ne sont pas connus *a priori*. Par conséquent, plutôt que d'exprimer directement le plan de déploiement, nous proposons de le construire à partir d'expressions déclaratives de propriétés de déploiement. Par exemple, nous pouvons exprimer qu'un composant doit être déployé sur un périphérique qui exécute le système d'exploitation Linux avec au moins 1 Mo de mémoire libre, et que le dispositif doit être situé dans la même échelle du point de vue réseau qu'un autre composant.

Pour exprimer ces propriétés de déploiement multi-échelle complexes, nous estimons que le concepteur de déploiement pourrait bénéficier d'un langage spécifique au domaine (DSL, *Domain-Specific Language*). De manière générale, un DSL améliore l'expressivité en fournissant des idiomes et des abstractions spécifiques à un domaine, et des outils modernes de génie logiciel peuvent fournir plusieurs niveaux de validation. Dans le projet INCOME, nous avons fourni une première version d'un tel DSL, nommé MuScADeL (*MultiScale Autonomic Deployment Language*) [3], qui est inspiré de Deladas [9] et de j-ASD [12], et qui considère les préoccupations multi-échelles et extra-fonctionnelles. La grammaire de ce DSL et sa sémantique, ainsi qu'un exemple complet, sont présentés dans [3].

4.3 Le déploiement en action

Dans notre approche, le plan de déploiement est la solution d'un problème de satisfaction de contraintes, formulé en traduisant les propriétés de déploiement en contraintes qui sont ensuite traitées par un solveur de contraintes. En plus de ces contraintes, le solveur prend en entrée l'état du domaine de déploiement, qui est construit par un ensemble de sondes précédemment installées sur les différents appareils. Le morceau de logiciel qui encapsule les sondes est un programme d'amorce ou *bootstrap* qui permet tout d'abord de rassembler toutes les

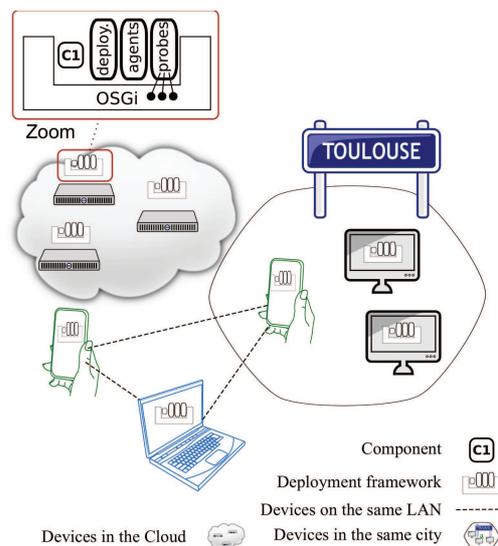


Figure 4. Exemple de déploiement multi-échelle

autorisations nécessaires du propriétaire de l'appareil ou de son administrateur, puis d'enregistrer cet appareil comme un site potentiel pour le déploiement des entités de gestion de contexte.

L'amorce de chaque dispositif comprend également un conteneur OSGi qui est utilisé pour le déploiement local. La contrainte pour les développeurs du gestionnaire de contexte est de fournir les composants sous la forme de *bundles* OSGi natifs (composants OSGi) ou d'envelopper les composants dans des *bundles* OSGi. Deux autres *bundles* OSGi doivent être présents sur chaque appareil et contenir : les sondes qui contrôlent dynamiquement les propriétés de déploiement, et un environnement d'exécution pour des agents logiciels. Ces agents logiciels déployés sur les différents dispositifs sont les éléments du système réparti multi-échelle chargés de la réalisation du déploiement. Par exemple, les agents présents sur un dispositif scrutent en permanence les sondes locales. Lorsqu'ils détectent une violation de contrainte, un système hiérarchique d'agents répartis sur chaque échelle du système adapte le plan de déploiement. Dans certains cas, des interactions avec la couche de gestion de contexte sont nécessaires.

La figure 4 illustre un déploiement impliquant plusieurs points de vue et échelles : le point de vue « dispositif » avec les échelles « smartphone », « ordinateur portable », et « nuage » ; le point de vue « géographique » ; et le point de vue « réseau ». L'utilisation d'agents pour le déploiement autonome n'a jamais été expérimentée dans des environnements ouverts et de grande envergure tels que les systèmes multi-échelles. L'autonomie des agents, leur capacité de communication asynchrone et leur potentiel de collaboration en font de bons candidats pour permettre l'autonomie et l'adaptation dynamique du déploiement multi-échelle.

5. CONCLUSION

Dans cet article, nous avons montré que les nouveaux défis posés par l'IoT en termes d'interaction spontanée, de volatilité des composants, de quantité de données de contexte transitoires, nécessitent des solutions innovantes permettant de mieux raisonner à l'échelle pertinente. Nous avons identifié cinq

conditions à remplir par la nouvelle génération de gestionnaires de contexte pour permettre leur déploiement sur de multiples échelles de l'IoT. Nous avons ensuite décrit les principales solutions proposées par le projet INCOME pour répondre à ces besoins d'une manière intégrée :

- une architecture ouverte de dispositifs et d'objets de différentes puissances et tailles hébergeant des producteurs, des processeurs et des consommateurs de contexte ;
- des modes de communication répartie *push* et *pull* bien adaptés au découplage spatio-temporel des objets de l'IoT, tout en garantissant la livraison des données de contexte grâce à des filtres d'annonce et de souscription, et grâce à des algorithmes de routage efficaces. Nous avons validé notre approche par la preuve formelle de nos algorithmes de routage ;
- une approche de modélisation uniforme de contrats pour exprimer les critères de QoC, les propriétés attendues pour le respect de la vie privée et les dimensions multi-échelles pour la mise en place de filtres de routage génériques, configurables et intelligents. Ces filtres évitent les disséminations de contexte superflues dans le réseau logique des courtiers de contexte en vérifiant d'abord le respect des contrats. Face à la grande quantité de données générées par l'IoT, ceci permet d'offrir une gestion de contexte performante ;
- une solution de déploiement multi-échelle et autonome comprenant i) un DSL pour la spécification du déploiement et ii) des agents mobiles pour sa réalisation et son adaptation à l'exécution en cas d'évolution de l'infrastructure multi-échelle.

Un premier prototype d'une expérience améliorée de transport citadin avec recommandation du meilleur itinéraire selon les situations de contexte a été mis au point pour la validation de notre approche. Nous travaillons actuellement à la consolidation de nos premiers résultats sur d'autres exemples d'applications. Nous cibons tout d'abord des applications grand public exploitant les informations collectées par l'IoT, mais les solutions actuellement explorées pourront contribuer à d'autres domaines tels que la gestion de réseau sensible au contexte.

6. REMERCIEMENTS

Ce travail s'inscrit dans le cadre du projet INCOME soutenu par l'Agence Nationale de la Recherche (ANR-11-INFR-009, 2012-2015).

7. REFERENCES

- [1] ATZORI, L., IERA, A., AND MORABITO, G. The Internet of Things: A survey. *Computer Networks* 54, 15 (2010), 2787–2805.
- [2] BELLAVISTA, P., CORRADI, A., FANELLI, M., AND FOSCHINI, L. A Survey of Context Data Distribution for Mobile Ubiquitous Systems. *ACM Computing Surveys* 44, 4 (Aug. 2012), 24:1–24:45.
- [3] BOUJBEL, R., LERICHE, S., AND ARCANGELI, J.-P. A DSL for Multi-Scale and Autonomic Software Deployment . In *ICSEA - International IARIA Conference on Software Engineering Advances* (Venice, Italy, Oct. 2013), pp. 291–296.
- [4] BUCHHOLZ, T., KUPPER, A., AND SCHIFFERS, M. Quality of Context Information: What it is and why we Need it. In *10th Int. Workshop of the HP OpenView University Association (HPOVUA)* (Geneva, Switzerland, July 2003), Hewlett-Packard OpenView University Association.
- [5] CHABRIDON, S., LABORDE, R., DESPRATS, T., OGLAZA, A., MARIE, P., AND MACHARA MARQUEZ, S. A survey on addressing privacy together with quality of context for context management in the internet of things. *Annals of Telecommunications* 69, 1 (Feb. 2014), 47 – 62.
- [6] COUTAZ, J., AND REY, G. Foundations for a Theory of Contextors. In *Proc. 4th International Conference on Computer-Aided Design of User Interfaces* (Valenciennes (France), May 2002), C. Kolski and J. Vanderdonck, Eds., Kluwer, pp. 13–34.
- [7] CUGOLA, G., AND MARGARA, A. Processing Flows of Information: From Data Stream to Complex Event Processing. *ACM Computing Surveys* 44, 3 (June 2012), 15:1–15:62.
- [8] DEARLE, A. Software deployment, past, present and future. In *Future of Software Engineering* (2007), pp. 269–284.
- [9] DEARLE, A., KIRBY, G., AND MCCARTHY, A. J. A Framework for Constraint-Based Deployment and Autonomic Management of Distributed Applications. In *ICAC* (2004), IEEE Computer Society, pp. 300–301.
- [10] MACHARA MARQUEZ, S., CHABRIDON, S., AND TACONET, C. Trust-based Context Contract Models for the Internet of Things. In *The 2013 International Symposium on Ubiquitous Intelligence and Autonomic Systems (UIAS'2013)* (Vietri Sul Mare, Italy, Dec. 2013), pp. 115 – 120. In conjunction with The 10th IEEE International Conference on Ubiquitous Intelligence and Computing (UIC'2013) and The 10th IEEE International Conference on Autonomic and Trusted Computing (ATC'2013).
- [11] MARIE, P., DESPRATS, T., CHABRIDON, S., AND SIBILLA, M. QoCIM : a meta-model for quality of context. In *CONTEXT'13 : Eighth International and Interdisciplinary Conference on Modeling and Using Context* (Oct. 2013), Springer, Ed., vol. 8175. Collection : Lecture Notes in Computer Science.
- [12] MATOUGUI, M. E. A., AND LERICHE, S. A middleware architecture for autonomic software deployment. In *ICSNC '12 : The Seventh International Conference on Systems and Networks Communications* (Lisbon, Portugal, 2012), XPS, pp. 13–20. 12619 12619 .
- [13] PERERA, C., ZASLAVSKY, A., CHRISTEN, P., AND GEORGAKOPOULOS, D. Context aware computing for the internet of things: A survey. *Communications Surveys Tutorials, IEEE* (Feb. 2014).
- [14] REY, G. *Contexte en Interaction Homme-Machine : le contexteur*. PhD thesis, Université Joseph Fourier, Grenoble, France, Aug. 2005.
- [15] SADRI, F. Ambient intelligence: A survey. *ACM Computing Surveys* 43, 4 (2011), 36.