A Framework for Multiscale-, QoC- and Privacy-aware Context Dissemination in the Internet of Things

Sophie Chabridon[1(✉)], Denis Conan[1], Thierry Desprats[2], Mohamed Mbarki[3], Chantal Taconet[1], Léon Lim[1], Pierrick Marie[2], and Sam Rottenberg[1]

[1] SAMOVAR, Télécom SudParis, CNRS, Université Paris-Saclay, 9 Rue Charles Fourier, 91011 évry Cedex, France
{Sophie.Chabridon,Denis.Conan,Chantal.Taconet,Leon.Lim, Sam.Rottenberg}@telecom-sudparis.eu

[2] Université de Toulouse, IRIT UMR, 5505 Toulouse, France
{Thierry.Desprats,Pierrick.Marie}@irit.fr

[3] ARTAL Technologies, 227 Rue Pierre-Gilles de Gennes, 31681 Labège, France
Mohamed.Mbarki@artal.fr

Abstract. The tremendous amount of context information that can be generated by the Internet of Things (IoT) calls for new solutions able to dig for the relevant information fitting applications' needs. This paper proposes to leverage multiscale-, Quality of Context (QoC)- and privacy-awareness for the efficient filtering of context information disseminated between the decoupled producers and consumers of the IoT. We first discuss some specific challenges that must be addressed by next generation context managers, including multiscalability, distributed push and pull communications, and the consideration of both QoC and privacy constraints. We then answer these challenges with a new context dissemination framework involving visibility and forwarding filters and illustrate it through the implementation of a collaborative social welfare scenario.

Keywords: Context management · Internet of Things · Context dissemination · Multiscalability · Quality of context · Privacy · Distributed event-based system

1 Introduction

The Internet of Things (IoT) is characterized by the extreme heterogeneity of the things it may interconnect and by their spontaneous interaction mode [1]. This leverages the adaptation capability of new context-aware applications. They are no more limited to their perceived ambient environment but they can collect context information from sources situated at other remote spaces and possibly at other scales too. In this article, we present our vision of multiscale context management systems combining components at different scales from local to remote ambient spaces and from the Cloud. This vision comes with new opportunities but also with new challenges. To master the tremendous amount of highly

S. Guo et al. (Eds.): CollaborateCom 2015, LNICST 163, pp. 207–218, 2016.
DOI: 10.1007/978-3-319-28910-6_19

dynamic and uncertain information collected from the IoT, new solutions are required to reason at the relevant scale and to manipulate context information of sufficient quality. This quality level should be considered jointly with privacy for not disclosing more information than necessary [7].

Our contributions are threefold: (i) multiscale context dissemination through distributed push and pull communication modes with (ii) joint management of the quality of context (QoC) and (iii) privacy protection.

The remainder of this paper is structured as follows. Section 2 defines the required properties of a new generation of context managers for the IoT and overviews the current state of the art. Section 3 describes the characteristics of our proposed framework for answering the requirements raised by context dissemination in the IoT. We then present a qualitative evaluation of our framework in Sect. 4.3. We conclude with a summary and an outline of further research in Sect. 5.

2 Toward a New Generation of Context Managers

2.1 Requirements for Context Management over the IoT

With the IoT, the need for distributing context management components raises dramatically. Indeed, a huge number of things can be connected to the global network infrastructure at any time during their life cycle either temporarily or permanently. Moreover, some things are situated in fixed locations while others are mobile. Finally, heterogeneity is the rule as small things may collaborate with powerful systems following the concept of mobile cloud computing [10]. **(R1) Distributed context management should be deployable on various network infrastructures exposing multiple scales**. Things are no longer considered only as context producers. They are independent actors that can also autonomously exchange, store and process information as well as interact to cooperatively compute useful data for human users, without involving them. Regarding the usefulness of context data for applications, one solution consists in describing context quality through meta-data: QoC is related to any information that describes the quality of context data [6]. **(R2) The choice of the relevant QoC metadata should be open and flexible**. Privacy preservation is another important concern of the public for the acceptance of the IoT [14]. **(R3) The knowledge provided by context data and their associated QoC metadata should respect the privacy of the context owners and not reveal more information than necessary**.

2.2 Existing Solutions for Context Dissemination

At first glance, context data dissemination can be addressed by existing solutions to the general problem of data distribution, that is by using a middleware providing some type of coordination models [12]: transactional, RPC-like, tuple-space, message-oriented, content-based, peer-to-peer or publish/subscribe middleware. However, context data distribution possesses original needs, which can be expressed in the concept of "uninformed" and "informed" context data distribution, that is whether the routing according to context needs is performed blindly or not [5]. The solution presented in this paper specifically targets three issues of informed context data distribution: (i) context data production/consumption decoupling with enforcement of context data visibility scopes, (ii) QoC-based context data distribution, and (iii) preservation of context owners' privacy. We review below related works regarding these three issues.

Following the publish/subscribe communication model [9], distributed event-based systems (DEBS) [17] are becoming popular as an enabler for context data dissemination in the IoT: Their interaction pattern decouples in space and time the things that produce events from applications that consume these events. With the concept of "scope" introduced in [11], the visibility of a notification can be limited to a subset of the consumers. In this paper, we argue that the system concept of scale matches with the DEBS concept of scope, and we propose multiscoping —i.e. distributed routing is impacted by the visibility of notifications that are analysed according to several dimensions. The REBECA [11] system implements monoscoping where only one scope graph is used at a time. Here we extend this concept to multiscoping and we allow the interoperability with scope-agnostic applications. [2] or [13] propose techniques for rewiring the broker overlay in order to avoid involving pure forwarder brokers. However, as highlighted by [3], a bad placement may result in a high number of messages being propagated between brokers. We favor a more distributed solution in which brokers can manage visibility filters.

[19] motivates the need for context managers to take QoC into account by application adaptation, middleware efficiency and users' privacy enforcement. In their proposed solution, all the parties should have an agreement in advance about the QoC they provide and require before sharing their context information. This mechanism needs further enhancement for the case of the IoT where the parties do not know each other in advance. Moreover, a common definition of the list of criteria used to qualify the context information within the context managers is still an open problem [5]. We therefore propose the unified QOCIM meta-model [16] allowing application developers to represent their own meta-data about QoC.

Concerning privacy protection in the IoT, most of the works offer confidentiality and anonymity using encryption mechanisms [4], k-anonymity or l-diversity models [20,22]. Our approach is complementary by granting access to consumers based on their intended use of context data expressed in contracts.

3 A Framework for Multiscale Context Management with QoC- and Privacy-awareness

Unlike large scale distribution focusing on a quantitative issue, multiscale distribution mainly addresses the heterogeneity of a system. We define *multiscalability* as the ability to cover several scales in at least one viewpoint which is associated to at least one dimension [18]. A *viewpoint* leads to a particular architectural view of a system which can be studied independently. A *dimension* corresponds to a measurable characteristic, numerical or semantic, of a system view for a given viewpoint. Using a measure, a dimension can then be divided using an ordered set of *scales*. Some examples of viewpoint are device (with the dimensions of storage capacity and processing power) and geography (with the dimensions of distance and adminstration area).

3.1 Main Components of the Global Architecture

Figure 1 illustrates the main architectural components of our context management infrastructure that provides context data dissemination between all categories of producers and consumers. A component can be either a collector or a capsule. A context collector is an element that provides raw context data and corresponds to a producer. A context capsule is a functional element of the context manager. It plays both the consumer and producer roles. Context-aware applications are clients that consume context information. A brokering service, distributed over the underlying physical network infrastructure, supports an efficient propagation of context data. It ensures the context delivery to the relevant context consumers. It may be deployed on several brokering overlay networks.

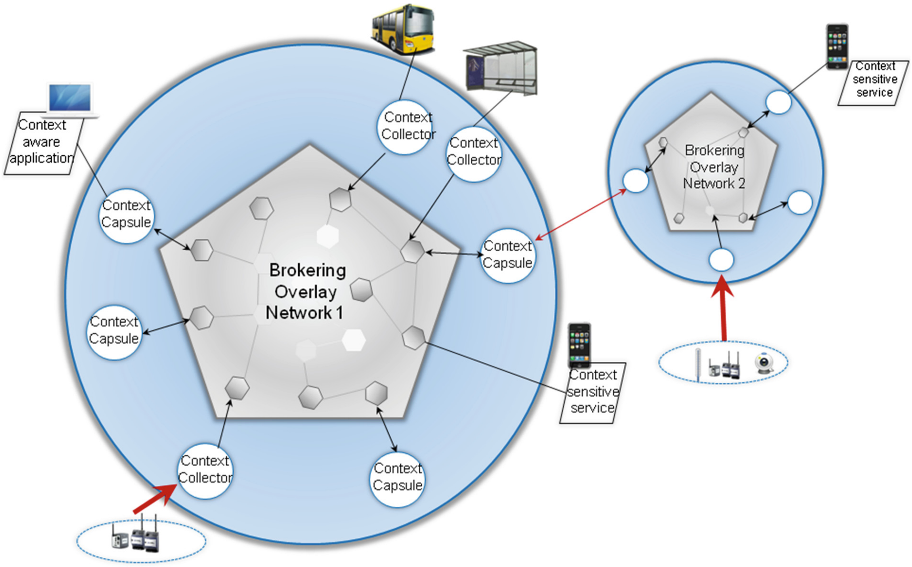

Fig. 1. Generic infrastructure with different types of overlay network

3.2 Distribution of Context Information with Multiscoping Using muDEBS

We have developed the muDEBS framework[1] as a Multiscale Distributed Event-Based System (DEBS). It allows to distribute context information in the IoT,

[1] https://fusionforge.int-evry.fr/www/mudebs/.

which is made up of brokers interconnected to build an overlay network of brokers. Clients of the system (producers and/or consumers of context data) also take part to the overlay network by being attached to brokers (every client is attached to one broker called the access broker). All these components (brokers and clients) communicate by asynchronous message passing. In order to ease the work of application designers, filters are content-based—i.e. they are constraints expressed on the whole content of notifications [9]. In addition, in order to remain as open and interoperable as possible, we assume a semi-structured data model *à la* XML.

The concept of "scope" is used in the field of DEBS to put the concept of visibility of notifications forward [11]. The visibility of a notification limits the set of consumers that may get access to this notification. We associate the system concept of scale instance (of multiscale distributed systems) with the concept of scope (of DEBS), and we go further by abstracting the customisability of DEBS with multiscoping [15]. Therefore, we complement the API of regular DEBS with the management of multiscoping. Figure 2 illustrates a scope graph for the *administration area* dimension and its projection onto the overlay network of brokers.

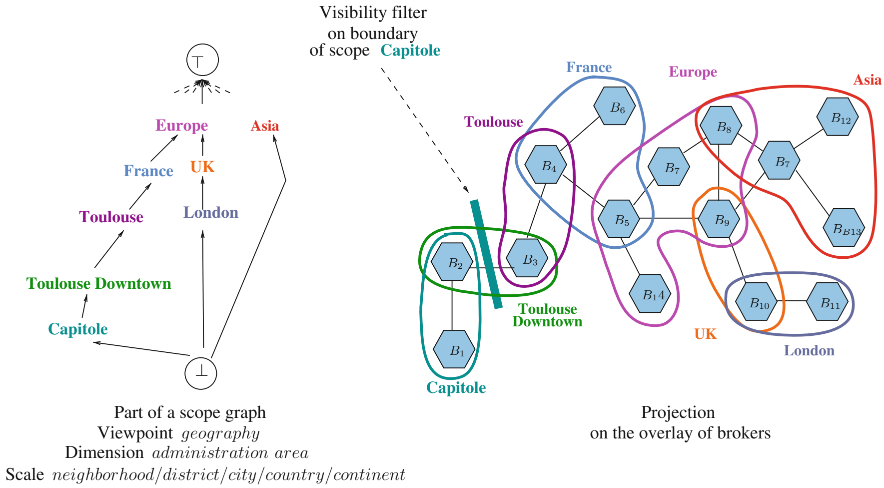

Fig. 2. Projection of a scope graph onto the overlay network of brokers

3.3 Contractualizing Multiscale-, QoC- and Privacy-awareness

The decoupling of context consumers and producers inherent to the IoT calls for new solutions for handling QoC and protecting the privacy of the persons concerned by the context data being collected. We handle this decoupling by using contracts where QoC, privacy and multiscale requirements and/or guarantees are stated. This contract model is implemented as part of the muContext framework[2]. We define two kinds of context contract, one for the context producer

[2] https://fusionforge.int-evry.fr/www/mucontext/.

side and one for the context consumer side. On one hand, a context producer contract includes clauses about the production of context data: (1) the privacy requirements, in the form of an XACML policy [21], stating context data they accept to provide to a context manager and how these data may be shared, (2) the QoC guarantees defining the level of QoC, compliant with QoCIM [16], they are committed to provide and (3) possibly multiscale requirements identifying in what scopes the context data should be visible. On the other hand, context consumers, encompassing context-aware applications and context end-users, describe their QoC and multiscale requirements, and privacy guarantees, in the form of ABAC (Attribute-Based Access Control) information [8], in context consumer contracts.

4 Qualitative Validation

We have experimented the muDEBS and muContext frameworks through the implementation of the 4ME (*Mobile, MultiModal, Multiscale Experience*) application as part of the INCOME project[3]. 4ME provides collaborative services to mobile users in major urban cities. A network of muDEBS brokers is deployed in those cities. End users define their general privacy rules. They have the ability to adjust those rules for each application.

We describe in this section how to implement and deploy 4ME and give some lessons learned in this development.

4.1 Illustrative Scenario

A social welfare association proposes leisure activities to its members (students, employees, etc.) and provide them with the 4ME collaborative context-aware application they can use on their smartphone. 4ME allows members to register/unregister to activities, get in touch with members sharing common activities, be aware of their current location and arrange to meet on the road while heading to the activities'premises in the city.

The *Easy Cooking* activity takes place on the last Friday of each month in the *Montaudran* district in the city of Toulouse, and the *Play Soccer* activity is proposed every Friday in the neighborhood of the *Capitole* place of the *Toulouse Downtown* district. Both activities are scheduled from 12:30pm to 1:30pm. A member can register to several activities, change its activity registrations, but cannot be registered to two activities at the same time slot.

The progress of an instance of the scenario is the following. Julian has registered this week to the soccer activity. It is already 12:15pm when he gets out of a meeting. He launches the 4ME app to determine whether he can meet another participant on his way to the activity's premises. As Julian has previously chosen the soccer activity, his 4ME app is subscribed and receives the notifications of the *Play soccer* scope. Each member enrolled in an activity publishes his/her

[3] http://anr-income.fr.

current position in the appropriate scope. The muDEBS framework is responsible for managing the spread of each position and its delivery in the form of a notification to all members belonging to the same scope with the appropriate QoC according to the privacy rules. Julian therefore gets notified of the positions of the members of the *Play soccer* group. He notes that no member is nearby and that he is already late and will not be able to arrive on time. Julian then decides to cancel his participation to the *Play soccer* activity. He checks the status of the *Easy Cooking* activity (the activity's premises are nearer) and learns that there are remaining places. He decides to enroll in this activity, and his agenda gets updated. Following this change in his agenda, the 4ME app unsubscribes from the *Play soccer* scope and subscribes to the *Easy Cooking* scope. Likewise, his position is not visible anymore to members of the *Play soccer* group and is henceforth visible to members of the *Easy Cooking* group. Julian is no longer notified of the positions of members of the *Play soccer* group. He is now notified of the positions of members of the *Easy Cooking* group. Julian sees that Corina is on the way and that she is nearby. He therefore decides to contact her in order to go to the activity's premises with her. The *Easy Cooking* teacher, with a simple look to the position of the participants on his tablet, is aware of their future arrival time.

4.2 Scenario Implementation

We show in this section some details of our implementation of the illustrative scenario. We show how filtering mechanisms are installed for this usecase.

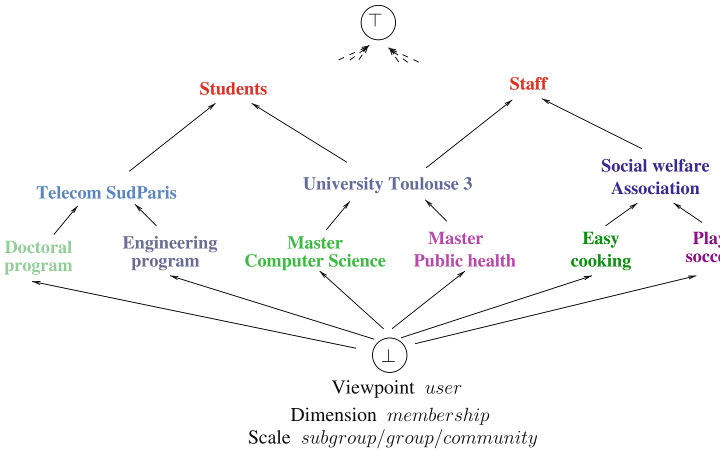

Fig. 3. Scope graph for *Membership* dimension of *User* viewpoint

Deployment and Administration of the MuDEBS Framework. In a first place, the context management administrator defines the different scope graphs corresponding to the multiscale characterization involved in the application. The scenario considers the *geography* and the *user* viewpoints. For each of these viewpoints, the scenario implies only one dimension, *administrative area* dimension and the *membership* dimension respectively.

Figure 2 (see Sect. 3.2) shows a scope graph for the *administrative area* dimension and its projection as an overlay on the network of brokers. Figure 3 shows the scope graph for the *membership* dimension.

In order to reduce the message traffic and to increase the scalability of the platform, the administrator may configure generic visibility filters. In our example, the administrator has chosen to limit the broadcast of messages inside a city. So the location notifications produced in the city of Toulouse are not forwarded to brokers associated by the projection to other cities.

For our experiment, we have deployed a network of 28 brokers. An excerpt of the deployment script is given in Listing 1.

Listing 1. Filtering on scope

```
startbroker --uri $B1_URI #start broker 1
startbroker --uri $B2_URI #start broker 2
#connect two brokers
broker --uri $B1_URI --command connect --neigh $B2_URI
# define scope projection and install visibility filters
broker --uri $B1_URI --command joinscope --dimension
     $ADMIN_AREA --subscope $P0 --superscope $FRANCE --
     mapupfile $MAPUP_FILTER --mapdownfile $MADOWN_FILTER
```

Implementation of the 4ME Application. Two components are installed on each end user mobile phone: a collector that publishes the position of the user with various QoC levels, and the 4ME application. muDEBS provides an API to write collectors and applications (see Listing 2). To publish positions, the collector (1) advertises and then (2) publishes. To receive positions, 4ME (3) subscribes and then (4) receives position.

Listing 2. MuDEBS API

```
1 advertise(advertise_filter, XACML_policy, SCOPES);
2 push(report);
3 subscribe(subscribe_filter, ABAC_request, SCOPES);
4 pull(report)
```

On each mobile phone, collectors are started by the 4ME application. A first one is started half an hour before the activity with a hundred meters accuracy, a second one is started when the user arrives in the district of the activity premises with a ten meter accuracy. The collector advertises with the privacy policies defined by the end user through an XACML description. The collector defines with an advertisement filter the QoC of the publication. The advertisement may define for each dimension the scope of the publications. For the scenario, the scope is restricted to the membership dimension in order to limit the diffusion to the users who share the same activity, there is no additional restriction on the geography dimension.

The 4ME application is started on each mobile phone. It is configured differently on Corina and Julian's phones and on the teacher tablet. Indeed, Corina and Julian require positions with a good accuracy in order to be able to meet on the road, while the teacher needs to receive the position of all the meeting participants even with a lower accuracy. 4ME sends a subscription with a specific filter indicating the relevant scopes and with the definition of the purpose of the subscription through an ABAC description. An excerpt of Corina's subscription filter is given in Listing 3. Line 3 chooses the observation to receive the users positions, line 5 chooses the QoC criterion corresponding to accuracy (with identifier 4.1), and line 7 defines the required accuracy value.

Before the publication of the positions, muDEBS must first of all determine whether the ABAC information does match the privacy policy associated with an advertisement in order to authorise or not access to the publication message and determine the required associated QoC level.

Listing 3. Filter with QoC guarantees

```
1 // Constraint on context data including QoC-value constraint
2 if (xpath.evaluate("//observable[uri='#location'] and
3                    //qocindicator[@id='4' and
4                    qoccriterion[@id='[4.1]']/
5                    qocmetricdefinition[@id='4.1']
6                    and qocmetricvalue[@value<='10']]",
7                    doc, XPathConstants.NODESET).length == 0) {
8                          return true;}
```

Figure 4 shows the screen of Corina when Julian enters the district of the activity premises.

4.3 Lessons Learned

4ME has been implemented as an evaluation prototype by the industrial partner of the project. muDEBS is able to filter publications for different purposes. Scale filtering is achieved easily with muDEBS through a graph of scopes, either for administration purpose (visibility filter), or for application specific purpose. This does not require any additional development by using the subscription/advertisement scope parameter. muContext allows decoupled producers and consumers to express their requirements in terms of QoC and privacy. Subscription filters enable developers to specify various QoC requirements and allow applications to receive information with the appropriate QoC level. Privacy concerns are taken into account as early as possible in the design of applications through the use of the XACML standard. In this experiment, the industrial partner has been able to define easily all the appropriate filters.

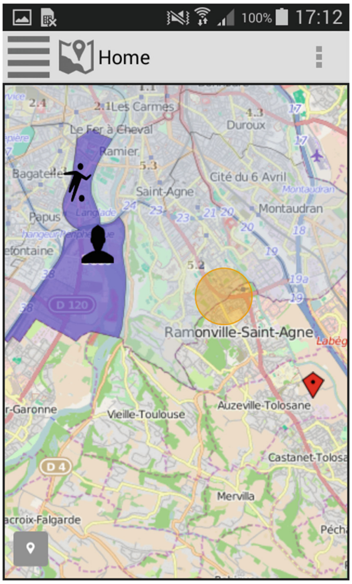

Fig. 4. On the way to a social activity - Screen copy

5 Conclusion

The new challenges brought by the IoT in terms of spontaneous interaction, instability, amount of transient context data, call for innovative solutions that can reason at the relevant scales. We identified three requirements to be fulfilled by the new generation of context managers for enabling their deployment at multiple scales over the IoT. We then described the key features proposed by the INCOME project to address these requirements in an integrated way: (i) An open architecture hosting entities of various power and size encompassing context producers, context processors and context consumers. (ii) Distributed push and pull communication modes well adapted to the spatio-temporal decoupling of these entities while guaranteeing context data delivery thanks to specific advertisement and subscription filters, and thanks to efficient routing algorithms. (iii) An uniform contract modeling approach for specifying QoC criteria, privacy properties and multiscale dimensions that allows generic, configurable and intelligent routing filters. Such filters avoid superfluous context propagations in the brokering overlay network by first checking contract compliance. Facing the big amount of data generated by the IoT, this will preserve performance. We have presented the 4ME application for collaborative social welfare activities and the lessons learned in the development of a real scenario.[4]

[4] http://anr-income.fr.

Acknowledgments. This work is part of the French National Research Agency (ANR) project INCOME (ANR-11-INFR-009, 2012-2015). The authors thank all the members of the project who contributed directly or indirectly to this article.

References

1. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
2. Baldoni, R., Beraldi, R., Querzoni, L., Virgillito, A.: Efficient publish/subscribe through a self-organizing broker overlay and its application to SIENA. Comput. J. **50**(4), 444–459 (2007)
3. Barazzutti, R., Felber, P., Fetzer, C., Onica, E., Pineau, J.-F., Pasin, M., Rivière, É., Weigert, S.: StreamHub: a massively parallel architecture for high-performance content-based publish/subscribe. In: Proceedings of the ACM DEBS, July 2013
4. Barazzutti, R., Felber, P., Mercier, H., Onica, E., Rivière, E.: Thrifty privacy: Efficient support for privacy-preserving publish/subscribe. In: 6th ACM International Conference on Distributed Event-Based Systems, pp. 225–236. ACM, New York (2012)
5. Bellavista, P., Corradi, A., Fanelli, M., Foschini, L.: A survey of context data distribution for mobile ubiquitous systems. ACM Comput. Surv. **44**(4), 24:1–24:45 (2012)
6. Buchholz, T., Kupper, A., Schiffers, M.: Quality of context information: What it is and why we Need it. In: 10th International Workshop of the HP OpenView University Association (HPOVUA), Geneva, Switzerland, July 2003
7. Chabridon, S., Laborde, R., Desprats, T., Oglaza, A., Marie, P., Machara Marquez, S.: A survey on addressing privacy together with quality of context for context management in the internet of things. Ann. Telecommun. **69**, 1–2 (2014)
8. Covington, M.J., Sastry, M.R.: A contextual attribute-based access control model. In: On the Move to Meaningful Internet Systems, pp. 1996–2006 (2006)
9. Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.-M.: The many faces of publish/subscribe. ACM Comput. Surv. **35**(2), 41–88 (2003)
10. Fernando, N., Loke, S.W., Rahayu, W.: Mobile cloud computing: A survey. Future Gener. Comput. Syst. Elsevier **29**(1), 84–106 (2013)
11. Fiege, L., Cilia, M., Mühl, B.C.: Publish-subscribe grows up,: Support for management, visibility control, and heterogeneity. IEEE Internet Comput. **10**(1), 48–55 (2006)
12. Issarny, V., Caporuscio, M., Georgantas, N.: A perspective on the future of middleware-based software engineering. In: Proceedings of the Conference on Future of Software Engineering, Minneapolis, USA, pp. 244–258, May 2007
13. Sherafat Kazemzadeh, R., Jacobsen, H.-A.: Opportunistic multipath forwarding in content-based publish/subscribe overlays. In: Narasimhan, P., Triantafillou, P. (eds.) Middleware 2012. LNCS, vol. 7662, pp. 249–270. Springer, Heidelberg (2012)
14. Langheinrich, M.: Privacy by design-principles of privacy-aware ubiquitous systems. In: Abowd, G.D., Brumitt, B., Shafer, S. (eds.) UbiComp 2001. LNCS, vol. 2201, pp. 273–291. Springer, Heidelberg (2001)
15. Lim, L., Conan, D.: Distributed event-based system with multiscoping for multiscalability. In: Proceedings of the 9th Workshop on Middleware for Next Generation Internet Computing, MW4NG 2014, p. 3: Observation of strains. Infect Dis Ther. 3(1), 35–43.: 1–3: 6. ACM, New York 2014 (2011)

16. Marie, P., Desprats, T., Chabridon, S., Sibilla, M.: QoCIM: A meta-model for quality of context. In: Brézillon, P., Blackburn, P., Dapoigny, R. (eds.) CONTEXT 2013. LNCS, vol. 8175, pp. 302–315. Springer, Heidelberg (2013)
17. Mühl, G., Fiege, L., Pietzuch, P.R.: Distributed Event-Based Systems. Springer, Heidelberg (2006)
18. Rottenberg, S., Leriche, S., Taconet, C., Lecocq, C., Desprats, T., MuSCa: A multi-scale characterization framework for complex distributed systems. In: 4th FedCSIS Workshop on Model Driven Approaches in System Development (MDASD), Warsaw, Poland, September 2014
19. Sheikh, K., Wegdam, M., Van Sinderen, M.: Middleware support for quality of context in pervasive context-aware systems. In: 5th IEEE Conference on Pervasive Computing and Communications - Workshops, March 2007
20. Weixiong, R., Lei, C., Sasu, T.: Toward efficient filter privacy-aware content-based pub/sub systems. IEEE Trans. Knowl. Data Eng. **25**(11), 2644–2657 (2013)
21. eXtensible Access Control Markup Language (XACML) Version 3.0. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html, January 2013
22. Zhou, Y., Shou, L., Shang, X., Chen, K.: Dissemination of anonymized streaming data. In: 9th ACM International Conference on Distributed Event-Based Systems (DEBS 2015), Oslo, Norway, pp. 104–115, July 2015