



**HAL**  
open science

## Combining fog architectures and distributed event-based systems for mobile sensor location certification

Fátima Castro-Jul, Denis Conan, Sophie Chabridon, Rebeca P. Díaz Redondo, Ana Fernández Vilas, Chantal Taconet

### ► To cite this version:

Fátima Castro-Jul, Denis Conan, Sophie Chabridon, Rebeca P. Díaz Redondo, Ana Fernández Vilas, et al.. Combining fog architectures and distributed event-based systems for mobile sensor location certification. UCAmI 2017: 11th International Conference on Ubiquitous Computing and Ambient Intelligence, Nov 2017, Philadelphia, United States. pp.27 - 33, 10.1007/978-3-319-67585-5\_3. hal-01634002

**HAL Id: hal-01634002**

**<https://hal.science/hal-01634002v1>**

Submitted on 30 Nov 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Combining Fog Architectures and Distributed Event-Based Systems for Mobile Sensor Location Certification

Fátima Castro-Jul<sup>1</sup>, Denis Conan<sup>2</sup>, Sophie Chabridon<sup>2</sup>, Rebeca P. Díaz Redondo<sup>1</sup>, Ana Fernández Vilas<sup>1</sup>, Chantal Taconet<sup>2</sup>

<sup>1</sup> I&C Lab, AtlantTIC Research Center, Universidade de Vigo, Spain  
{fatima,rebeca,avilas}@det.uvigo.es

<sup>2</sup> SAMOVAR, Télécom SudParis, CNRS, Université Paris-Saclay, Évry, France  
{firstname.lastname}@telecom-sudparis.eu

**Abstract.** Event filtering is of paramount importance in large-scale urban sensing, where an enormous quantity of data is generated. Multiple criteria can be considered for filtering, location being one of the most valuable ones. Obtaining high-quality (trustworthy, accurate) location information helps to contextualize the event content and provides trust both on the source producer and on the publication itself. However, IoT-based urban services rely often on cloud architectures, which have no means to support location certification. To meet the need for location certification support in urban distributed event-based systems (DEBS), we propose three different fog architectures targeted at scenarios with mobile event producers.

**Key words:** participatory sensing, smart city, internet of things

## 1 Introduction

The Smart City paradigm envisions a city where information technologies enhance citizens quality of life by improving urban services. The Internet of Things (IoT) plays a major role in building smart cities since it enables the interconnection of different devices, vehicles or any kind of objects. Additionally, Distributed Event-based Systems (DEBS) [1] have become a popular interaction paradigm for IoT-based smart city services [2]. These systems support event filtering and producer mobility. The former is essential in large-scale urban sensing, due to the tremendous quantity of data produced. The latter is a promising alternative to building a costly fixed sensing and communication infrastructure to cover the whole city.

Location is one of the main criteria to categorize and filter urban events. Providing spatial context in event distribution allows consumers to subscribe only to their geographical areas of interest. Moreover, even when the event is not location-dependent, receiving information on the producers' whereabouts increases trust in the event content. Implementing location evaluation mechanisms is a non-trivial task in cloud architectures. However, it becomes simpler when

part of the data processing is moved to the edges of the network, using a fog architecture.

Fog computing is a promising yet still unexploited possibility to build urban DEBS that provide sensor location certification. To analyze their potential, we have designed three architectures based on fog computing that combine proximity and cloud communication. Our contribution is targeted at providing proximity-based location verification on the edges of the network while employing an architecture that enables communication over a wide-spread area. The aim is to support the quality assessment of publications by relying on proximity-based communication. Thus, we leverage our previous works on quality of context information in DEBS [3] and on distributed proximity-based collaboration with peer devices [4].

This paper is structured as follows. First, we present an overview of related work (Section 2). Then, we detail our proposal (Sections 3 and 4). Finally, we conclude and present our ongoing work in Section 5.

## 2 Related Work

The importance of mobile event producers has considerably grown over the last ten years, due to the emergence of participatory sensing [5] and mobile crowdsensing systems [6, 7]. These systems rely on collaborative data gathering and can be implemented using different architectures. When data is collected in a centralized manner, a cloud entity is in charge of receiving all the sensor readings and analyzing them. In this case, a blind confidence is placed in the locations claimed by data producers. In order to enhance trust in the provided location, a possible solution is to switch to a proximity-based totally ad hoc network that ensures producers' communication directly with consumers in their very area [4]. This is a good solution for small and densely populated urban areas but it becomes expensive when producers and their subscribed consumers are far away from each other. An alternative is to migrate part of the data processing from the cloud to the edges of the network, using a fog computing architecture. Fog computing [8] has become a useful paradigm for IoT architectures since it provides heterogeneity and fast mobility support [9], low latency and scalability [10]. [11] targets better localized accuracy through location-based customization but does not verify the trustworthiness of location information. To the best of our knowledge, the potential of fog architectures to support location certification schemes has remained unexploited.

## 3 DEBS with location certification support

Our DEBS is targeted at participatory sensing scenarios with high producer mobility and periodic publications of sensed data. We can think of, for instance, a participatory noise measurement system that relies on citizens smartphones. Using a DEBS architecture, event distribution is not limited to the close producer's neighborhood but extends outside the city boundaries and beyond. Consumers

subscribe by providing filters that specify the publications they are interested in. Publications are routed from producers to consumers by an overlay network of brokers.

Our goal is to improve publication quality by increasing trust on the publication’s source location. As a result, we focus our attention on producers’ mobility, and for the sake of simplicity we assume the broker and consumer structure to be fixed. A producer is connected to at most one broker, namely its access broker. When producers move, they may disconnect and reconnect to a different broker.

Location information may be included in every publication: it consists of an explicit representation using coordinates, or of the identifier of a specific area or neighborhood. When the location can be verified, the publication is enriched with a location certificate. The location is added by the producer and may be verified either by the producer or the broker but it is always the broker, assumed to be trustworthy, that decides whether to provide the certificate (Algorithm 1). Consumers choose whether they ask the DEBS to filter out uncertified location.

---

**Algorithm 1** : at broker  $B$ , handle a publication

---

```

1: procedure HANDLEPUBLICATION(Producer  $X$ , Publication  $n$ )
2:   if  $B$  is the access broker of  $X$  then
3:      $locOk \leftarrow$  VerifyLocation( $n$ )           // verify location information included in  $n$ 
4:     if  $locOk$  then
5:       ProvideLocationCertificate( $n$ )           // add to  $n$  a location certificate
6:     Forward  $n$  to interested neighbor brokers and local consumers

```

---

A publication is certified if the location claim has been verified. The verification consists in assessing the location information using either extra information provided by the producer to support the location claim or the collaboration of other peers in the area. In both cases, the verification relies on proximity-based communication. Proximity-based communication connects devices that are placed closely, at a distance from each other that allows them to be reached using a wireless short-range communication technology, such as Bluetooth or ad hoc WiFi. If a producer that claims to be in a certain area cannot be reached by other devices known to be there, we can conclude that it is not providing its actual location. Following these principles, we foresee three different DEBS architectures supporting location verification and detailed in Section 4.

## 4 Fog-based architectures

In traditional DEBS cloud architectures, clients (producers and consumers) are connected through an overlay of brokers that are hosted in a cloud. An access broker may serve clients from different locations (Figure 1.a). It has no knowledge about the clients’ location and no way to verify whether they are claiming false locations. As an alternative, we introduce three different fog architectures for a DEBS with location verification support. The different combinations of cloud and short-range communication determine how the location verification process is performed.

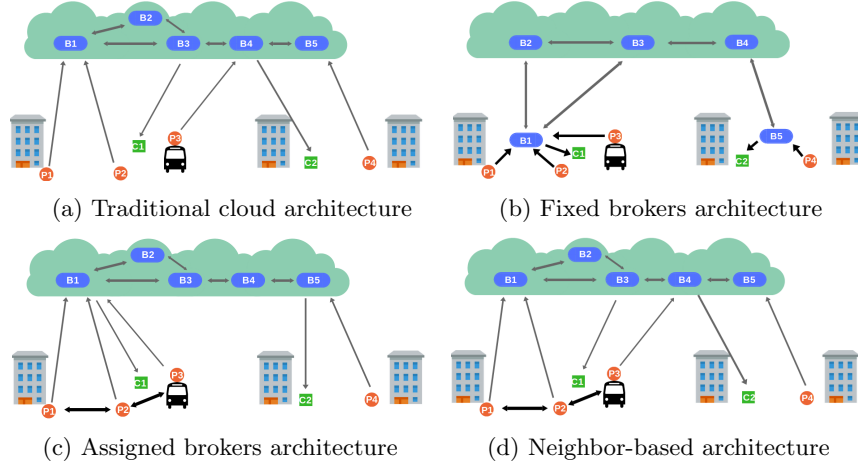


Fig. 1: Fog-based architectures

#### 4.1 Fixed brokers architecture

This architecture considers two levels of brokers: border brokers that are in the vicinity (e.g., B1, B5 in Figure 1.b) and inner brokers that are hosted in the cloud (e.g., B2, B3, B4). Border brokers are placed all over the city covering different areas; they are fixed, thus being assigned an immutable location. Clients in the same area are connected to the same access broker using a short-range transmission technology that ensures the veracity of the clients' location. As a result, location verification is straightforward. The access broker will certify the location of every publication whose location attribute coincides with the broker location (Algorithm 2).

A mobile producer may have to disconnect temporary. Publications created by disconnected producers are stored in a local queue, to be issued to the new access broker when reconnecting. If the location is different from the new access broker's location, they are distributed but not considered for location certification because a broker can not certify a location different from its own.

---

#### Algorithm 2 Verify location information (Fixed brokers architecture)

---

- 1: **procedure** VERIFYLOCATION(Publication  $n$ )
  - 2:     **return** (location included in  $n$  "equals" access broker's location)
- 

#### 4.2 Assigned brokers architecture

Even though a hybrid architecture is a cheaper alternative to deploying a purely fixed sensor network architecture, its cost is still high. As a result, we have designed a second architecture, which still exploits location verification through proximity-based communication but gives up the requirements for a fixed infrastructure. By considering location-based network partitions, we maximize the probability of having all the clients in an area connected to the same broker (Fig-

ure 1.c). Thus, they can still be in charge of producers’ location verification even though they cannot communicate with them using a short-range transmission technology. Producers obtain the IP address of their access brokers by providing their location to a discovery service. Producers include in their publications a set of other producers in the area with whom they have established a short-range communication link and who are connected to the same broker (*neigh*). If nearly all<sup>1</sup> of the producers included in the set are registered as local producers of the same access broker, the location is confirmed (Algorithm 3). Producers claiming a false location are not able to provide a valid set of neighbors.

---

**Algorithm 3** Verify location information (Assigned brokers architecture)

---

1: **procedure** VERIFYLOCATIONINFORMATION(Publication *n*)  
 2:     **return** (location included in *n* “equals” access broker’s location  $\wedge$  “most of” the producers  $\in$  *neigh* are local producers)

---

### 4.3 Neighbor-based architecture

Unlike the two previous architectures, brokers are not assigned to specific areas and location verification takes place between peer neighbor producers. Producers can be connected to any access broker, which may be different from the one their close neighbors are connected to (Figure 1.d). By exchanging short-range messages, they are always aware of who their neighbors are. When a producer is going to send a publication, it triggers a consensus process between its neighbors. In this process, neighbor producers propose location values and decide on one of them, which will be included in the publication. Since we are dealing with a dynamic scenario where it may be difficult to decide due to variations in the number of neighbors, we consider abortable consensus [12, 13]: the result of the consensus is either a location value or *aborted*. If no location value can be reached by consensus, the producer includes its own proposed location value in the notification. Then, the publication sent to the broker includes, in addition to the location, an indication about the consensus result (*successful* or *aborted*). The broker does not take part in the verification process and certifies every publication that includes an indication of successful consensus (Algorithm 4). Remark that changing the access broker does not necessarily imply a change of producers’ location and does not invalidate a location certification.

---

**Algorithm 4** Verify location information (Neighbor-based architecture)

---

1: **procedure** VERIFYLOCATIONINFORMATION(Publication *n*)  
 2:     **return** (indication of consensus in *n* equals *successful*)

---

## 5 Conclusion and Ongoing Work

This paper presents three different fog architectures, targeted at location certification support in DEBS for IoT urban services. They combine cloud and

<sup>1</sup> To tolerate disconnections and overlapping

proximity-based communication, which enables them to provide first-hand location verification.

We are currently working on the evaluation of the architectures. We plan to employ network simulations based on pedestrian traces generated in real urban maps. The assessment will be focused on the appropriate location verification and on the ability to handle producer’s mobility in highly dynamic urban scenarios.

**Acknowledgments** This work is funded by: the European Regional Development Fund (ERDF) and the Galician Regional Government under agreement for funding the Atlantic Research Center for Information and Communication Technologies (AtlantTIC), the Spanish Ministry of Economy and Competitiveness under the National Science Program (TEC2014-54335-C4-3-R) and a predoctoral grant financed by the Galician Regional Government (Consellería de Cultura, Educación e Ordenación Universitaria) and the European Social Fund.

## References

1. P. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec, “The Many Faces of Publish/Subscribe,” *ACM Computing Surveys*, vol. 35, June 2003.
2. A. Antonic, K. Roankovic, M. Marjanovic, K. Pripuic, *et al.*, “A mobile crowdsensing ecosystem enabled by a cloud-based publish/subscribe middleware,” in *FiCloud 2014*, pp. 107–114, IEEE, 2014.
3. L. Lim, P. Marie, D. Conan, S. Chabridon, T. Desprats, and A. Manzoor, “Enhancing Context Data Distribution for the Internet of Things using QoC-awareness and Attribute-based Access Control,” *Annals of Telecommunications*, vol. 71(3), pp. 121–132, 2016.
4. F. Castro-Jul, R. P. Díaz-Redondo, and A. Fernández-Vilas, “Have you also seen that? collaborative alert assessment in ad hoc participatory sensing,” in *UCAM I 2016*, pp. 125–130, Springer, 2016.
5. J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. Srivastava, “Participatory sensing,” *Center for Embedded Network Sensing*, 2006.
6. R. K. Ganti, F. Ye, and H. Lei, “Mobile crowdsensing: current state and future challenges,” *IEEE Communications Magazine*, vol. 49, no. 11, 2011.
7. B. Guo, Z. Yu, X. Zhou, and D. Zhang, “From participatory sensing to mobile crowd sensing,” in *PERCOM 2014*, pp. 593–598, IEEE, 2014.
8. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in *MCC’12*, pp. 13–16, ACM, 2012.
9. M. Yannuzzi, R. Milito, R. Serral-Gracià, D. Montero, and M. Nemirowsky, “Key ingredients in an iot recipe: Fog computing, cloud computing, and more fog computing,” in *CAMAD 2014*, pp. 325–329, IEEE, 2014.
10. K. Hong, D. Lillehun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, “Mobile fog: A programming model for large-scale applications on the internet of things,” in *ACM SIGCOMM, MCC ’13*, pp. 15–20, ACM, 2013.
11. A. Munir, P. Kansakar, and S. U. Khan, “IFCIoT: Integrated Fog Cloud IoT Architectural Paradigm for Future Internet of Things,” *CoRR*, [arxiv.org/abs/1701.08474](https://arxiv.org/abs/1701.08474), 2017.
12. R. Guerraoui and L. Rodrigues, *Introduction to Reliable Distributed Programming*. Springer-Verlag, 2006.
13. W. Chen, “Abortable Consensus and Its Application to Probabilistic Atomic Broadcast,” Tech. Rep. MSR-TR-2006-135, Microsoft Research, 2006.