

A Publish-Subscribe Scheme Based Open Architecture for Crowd-Sourcing

Róbert Szabó, Károly Farkas

► **To cite this version:**

Róbert Szabó, Károly Farkas. A Publish-Subscribe Scheme Based Open Architecture for Crowd-Sourcing. 19th Open European Summer School (EUNICE), Aug 2013, Chemnitz, Germany. pp.287-291, 10.1007/978-3-642-40552-5_27. hal-01497025

HAL Id: hal-01497025

<https://hal.inria.fr/hal-01497025>

Submitted on 28 Mar 2017

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.



A Publish-Subscribe Scheme Based Open Architecture for Crowd-sourcing

Róbert L. Szabó^{1,2} and Károly Farkas^{1,3}

¹ Inter-University Centre for Telecomm. and Informatics, Debrecen, Hungary

² HSNLab, Dept. of Telecommunications and Media Informatics,

³ Dept. of Networked Systems and Services,

Budapest University of Technology and Economics, Budapest, Hungary

szabo.robert@etik.hu, farkask@hit.bme.hu

Abstract. Participatory sensing, when a crowd of users collaborate to collect useful information, based applications are getting popular these days thanks to the proliferation of powerful mobile devices. The built-in sensors of smartphones offer an easy and handy way to monitor the environment and collect data which can serve as the basis of smart applications. However, the quick and flexible development and deployment of these applications call for a unifying open architecture. In this paper we propose a publish-subscribe based open participatory sensing architecture.

1 Introduction

The vision of collecting information by the assistance of the crowd and exploiting its voluntary work has been boosted recently thanks to the emergence of the web and mobile technologies. This is called participatory sensing whose power is that basically everybody has the possibility to contribute on a voluntary basis. In essence, a participatory sensing system enlists and utilizes a crowd of users to collaborate in building a system that is beneficial to the whole community.

The form of participatory sensing has been influenced by the development of technologies. Smart mobile devices open the way for mobile participatory sensing. They do not only serve as the key computing and communication platforms of choice, but also come with a rich set of embedded sensors, such as accelerometer, digital compass, gyroscope, GPS, microphone and camera. Collectively, these sensors are enabling new applications across a wide variety of domains, such as healthcare, social networks, safety, environmental monitoring and transportation [1]. Thus, mobile smartphones are possible bridges of everyday objects to our enriched world and a plethora of Internet of Things (IoT) applications can be envisioned that will be available in the future through community based sensing and monitoring [2].

Today the development of a participatory sensing application follows the Silo approach. These applications are developed in a stand-alone manner recreating all the necessary building blocks from scratch on both the client and the server/cloud side. This makes the development slow, cumbersome, error-prone

and inflexible. On the contrary, instead of the vertically integrated Silo approach our goal is to design a horizontally layered application framework and an open architecture enabling rapid prototyping to experiment with participatory sensing based innovative use-case scenarios. In such a situation, an architecture based on the publish-subscribe (pubsub) communication scheme is well suited where information sources publish events autonomously to the pubsub service whenever something happens. Users can subscribe to event contents and the pubsub service delivers the events to all the interested parties.

In the next section, we propose an XMPP publish subscribe based architecture for mobile participatory sensing.

2 Open Participatory Sensing Architecture

2.1 Requirements

Unifying open architecture: A typical participatory sensing application today has two *application specific* components: *i*) one at the user's device and *ii*) another one in the cloud [2]. This results in many parallelism, unnecessary developments and slow application innovation cycle.

We envision that the separation of application logics and some core communication, privacy and trust functions could result in a *unifying open architecture* where innovation can be done at the end systems focusing on the application and presentation layers. This could boost the developments similarly to the innovation at the application front enabled by IP. Such unifying open architecture shall not only allow independent application development but also the independent development of application support services.

Extensible information model: We want to allow independent application innovation. This can only be done if the middleware can transparently pass extra information which applications may attach to the basic messages.

Decoupling between producers and consumers: If we accept the fact that by participatory sensing we aim at monitoring a larger-scale phenomena that cannot be easily monitored by individuals, then we need to decouple producers and consumers efficiently in space, time and synchronization.

2.2 System Model

We propose a generic publish-subscribe interaction scheme with an extensible event language as our system model. The roles in this model are *Producers*, *Service Providers* and *Consumers*. They communicate through event based pubsub nodes as the core service. The model must be easily extensible and support the needs of the basic pubsub operations.

Producers: They are original *participatory* information sources in our model and are central to a participatory sensing setup. These individuals are actively involved in contributing their sensor data related to a larger-scale phenomena. They produce the raw data to the system.

Service Providers: They would like to introduce *extra* value to the raw data collected by participatory sensing. Thus, Service Providers *add extra value* to the information flow from participatory Producers to the Consumers. They might collect (Consumer role), store and analyze Producers' information to produce (Service Provider role) new information. Therefore they might, but not necessarily must, act at the same time as both Service Provider and Consumer. For example, data can be introduced into the participatory application from external systems/sources. If this data is larger in scope what an individual can possibly provide, then we call the associated role as Service Provider.

Consumers: They are the beneficiaries of the information flow. In general, participatory sensing is considered to be an open call for anybody to participate typically on-line in a task. This task can either belong to an owner (e.g., a company seeking solutions to its problem) or to a community or group of users. Nevertheless, the task (collection, analysis and dissemination of data) aims to give value to some beneficiaries or Consumers. Sometimes, these Consumers also act as Producers. When users act on both the Producer and the Consumer side of the process we call them as *Prosumers*.

Event Service: It realizes the *publish-subscribe* communication scheme, whereby decoupling Producers (publishers) and Consumers (subscribers) in time, space and synchronization.

Producers create original information by sensing their environment. They publish (marked with empty arrowheads) their information to event nodes (raw information nodes marked with blue). Service Providers subscribe (black arrowheads) to raw event nodes and receive information asynchronously. They combine their information with the participatory sensed data to publish value added information to Consumers. The information they produce are published to different content nodes. Consumers subscribe to content nodes according to their topics of interest and receive information from these nodes asynchronously.

2.3 Architecture

Our model can be directly mapped to the Extensible Messaging and Presence Protocol's (XMPP) [3] publish-subscribe service as follows (see Fig. 1):

A Service Provider creates a raw pubsub data node to collect Producers' data for their services. Producers must have at least *publisher-only* affiliation to be able to publish into these nodes. This requires either a pre-authorization or an extended access right management at the XMPP side. Similarly to the *open* node access mode, an *open-publish-only* node access can be introduced, whereby Consumers can openly publish to the corresponding nodes, but only affiliated Consumers and the owner can retrieve events. Producers with proper access rights can publish (stream) their sensors' data or annotations to these raw data nodes at the XMPP server. Service Providers collect these data by the pubsub subscription service and map their value added services to a structure, which allows reasonable content filtering for their Consumers. Prosumers participate in sensing and publish their sensor data or annotations into some raw XMPP pubsub nodes. Service Providers subscribed to these raw pubsub nodes collect,

store and analyze data and derive new information realizing value added services (related to a larger scale phenomena). This information is published into a structured collection of pubsub nodes, to which Consumers (Prosumers) can subscribe. The structure of pubsub service nodes can utilize XMPP's aggregation feature through collection nodes, where a collection node simply receives all events of its child nodes. Note however, that XMPP's collection node cannot filter events throughout the aggregation mechanism, therefore scalable content aggregation can only be realized through the Service Provider role. Such use-case is shown in the figure, where XMPP's aggregations are depicted with dark circles at the container node while empty circles represent only logical containment and intelligent aggregation through the service logic.

3 Summary

We proposed a horizontally layered, open, participatory sensing architecture based on XMPP's publish-subscribe interaction scheme. We described players and their roles related of the architecture. Use cases and implementation experiments will be shown in the accompanying poster.

Acknowledgments: The publication was supported by the TÁMOP-4.2.2.C-11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed by the European Social Fund. This work has been partially supported by the KIC ICTLabs under the activity 13064 CityCrowdSource of the action line Digital Cities. Károly Farkas has been partially supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References

1. Lane, N., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.: A survey of mobile phone sensing. *Communications Magazine, IEEE* **48**(9) (2010) 140 – 150
2. Ganti, R., Ye, F., Lei, H.: Mobile Crowdsensing: Current State and Future Challenges. *IEEE Communications Magazine* (November 2011) 32–39
3. Saint-Andre, P.: Extensible Messaging and Presence Protocol (XMPP): Core. RFC 6120 (Proposed Standard) (March 2011)

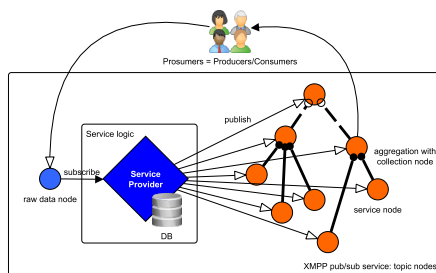


Fig. 1: XMPP Based Publish-Subscribe Architecture