

A Microservice Architecture Use Case for Persons with Disabilities

Andrea Melis¹(✉), Silvia Mirri², Catia Prandi², Marco Prandini²,
Paola Salomoni², and Franco Callegati¹

¹ Department of Electrical and Information Engineering,
Università di Bologna, Bologna, Italy

² Department of Computer Science and Engineering,
Università di Bologna, Bologna, Italy

{a.melis,silvia.mirri,catia.prandi2,marco.prandini,paola.salomoni,
franco.callegati}@unibo.it

Abstract. Applications supporting the independent living of people with disabilities are usually built in a monolithic fashion for a specific purpose. On the other hand, a crucial sector for the livability of urban spaces such as mobility is undergoing a deep transformation, heading towards flexible composition of standardized services. This paper shows how this approach allows to build better applications for people with specific needs, making them seamlessly integrated in the most modern approach to smart mobility.

Keywords: Mobility-as-a-Service · Microservices · Mobility aids for blind persons

1 Introduction

The advent of smart mobility has enabled the creation and development of applications for mobility support addressed to different categories of people, including those with disabilities [1–3]. Yet, the majority of these development paradigms provides vertical applications for accessible mobility, separated from the wealth of services available for average-capability travelers. For these applications, the integration of collected data, the possibility to share common sub-services, and the ability to interact with the same standards is rather difficult.

On the other hand, the rapidly growing diffusion of ICT solutions in every field of mobility is fostering the creation of novel approaches, such as the so-called *Mobility-as-a-Service* (MaaS) paradigm [4]. The enabling factor for these approaches to succeed is a strong drive towards standardization of data formats and access methods. According to the most modern model, every access to a piece of data, as well as every execution of a simple computation is offered as a service [5]. The collection of domain-specific services, completed by a set of commonly used infrastructural services, composes a service-oriented architecture, on which it is comparatively simple and fast to develop any application by orchestrating the needed services [6].

The purpose of this paper is to present a specific use case that shows how it is possible to create a scalable and integrated mobility service for blind users, as a result of the coordination of reusable components on a microservice architecture. After discussing goals and purposes of the MaaS philosophy in Sect. 2, we introduce our microservice development approach in Sect. 3, focusing in particular on the advantages that this paradigm introduces on the basis of a real, currently developing infrastructure, which is called *Smart Mobility for All* (SMAll). Then, Sect. 4 details our use case developed in the urban region of the city of Bologna. Finally, Sect. 5 concludes the paper with some final remarks and directions for future work.

2 The MaaS Approach

Mobility as a Service [4] is an innovative approach that aims to break the barrier between public and private transport. It is made viable by the integration in a coordinated infrastructure of the technologies exposed by every single operator. Born in the city of Helsinki [4], this paradigm is starting to spread throughout Europe and beyond, aiming to establish standards for the interoperability between different (even in terms of country) operators, and to encourage the creation of alternative solutions to the standard “winner-take-all” paradigm. For our purpose, this approach can be seen from 2 different point of view, the business one and the technological one.

2.1 The Business Logic

Very briefly, the principle of MaaS is that as long as every detail of the demand and supply for transportation services is known in real-time, there is no need for passengers to commit on specific means. Instead, they will enjoy a broad spectrum of alternatives from which to choose, taking into account the needs of the moment. For example, one could specify a very strict set of constraints in terms of comfort and timing, likely to result in a choice of premium means, while another could simply express the need for reaching a destination at the best price, getting a virtual ticket, and receiving real-time instructions about which means to use to complete the trip. Many business models are possible [7].

In its simplest form, a MaaS operator could be a smart broker for planning and paying trips on existing networks. A more innovative approach would be that of selling mobility packages allowing travelers to use pre-configured amounts of travel on different means. From the transport operators viewpoint, a MaaS platform could be a great opportunity to leverage integration and to exploit unused capacity. For example, a taxi company exposing vehicle availability and position in real-time could offer lower prices during off-peak times, thus appearing as a good alternative to mass transit; data-mining could allow operators to foresee correlations between various conditions (events, weather, accidents) and transportation needs, to allocate materials in the best possible way [8].

Also the role of public administration can undergo a significant change. Some administrations could choose to play the role of a central MaaS operator, exerting a stronger control on the local mobility agenda. Others could leave the field to private companies, hoping to benefit from market-driven optimization of citizens' patterns of mobility. They could also accurately monitor citizens, using collected data to plan investments and direct incentives towards specific goals [9].

In this context, our use case became interesting. Typically, scenarios with few users involved, and that require specific technologies, as it happens in the case of people with disabilities, have always been profitable only through a heavy vertical investment [10]. Thanks to the MaaS principle, where single services can be shared and public administrations can direct investments on the development of specific needs, even this kind of use case can be seen as an opportunity for MaaS operators.

2.2 The Technological Side

To effectively support the creation of MaaS providers, we envision the creation of an ecosystem of reusable components within a Service-Oriented Architecture, where standardized services efficiently and flexibly combine heterogeneous data sources, such as available transport options, real-time data regarding vehicles and infrastructures, pricing, etc., to provide customized travel planning, information and ticketing to final users, as well as monitoring and planning tools to policy-makers.

This infrastructure aims to create services to track timing, position, and availability of trains, buses, subways, shared bikes, shared cars, and to enable crowdsourcing of users' report, crowdsensing through devices [11,12], etc. in an overall effort of opening data and standardizing the interfaces to access them. Both operators and users can benefit from this plethora of mobility services.

A real example of this kind of infrastructure is the one that we are currently developing as a marketplace for mobility services, within the EIT Digital¹ called *Smart Mobility for All* (SMAll) [13]. In our vision, SMAll is the enabling technology to solve the challenges of the MaaS market, from developing user-contributed, crowdsourced applications, to launching a MaaS operator, to planning effective and sustainable transport policies for smart cities [14].

3 The Microservice Architecture Paradigm

The microservice architecture is a development style inspired by service-oriented computing that has recently started gaining popularity [15]. A microservice architecture can be defined as: "a distributed application where all its modules are microservices [16]."

The reasons why this architecture has been proposed is to try to overcome the limits of the so-called monolithic architecture [17]. In addition, we have to

¹ <http://www.eitdigital.eu/>.

consider the fact that this paradigm is a reasonable consequence of a world that is more and more oriented to the development on cloud, and therefore the need of connection and interoperability between the different platforms is increasing. Defined as [18] “a single-tiered software application in which the user interface and data access code are combined into a single program from a single platform”, the main limits of the monolithic approach are:

- The maintain process.
- The dependency chain of libraries and modules explode.
- The deployment phase
- The scalability of the program.

It is clear that this kind of architecture is obsolete and it cannot be the right solution for today’s service-oriented developments.

On the other side, the microservice architecture has been proposed with a diametrically opposite approach. Rather than having a single block composed of several force-dependent modules from each other, the microservice philosophy extracts each module making an independent and autonomous microservice following the statement “everything can be a microservice” [19].

This kind of distributed compositionality becomes his greatest strength. As described in [16], even the internal components of software are autonomous services, independent components conceptually deployed alone with dedicated technologies, both software and hardware. This means that this architectural style does not foster or forbid any particular programming, but it provides only a partition structure of the component the developer has to follow. Since all the components of a microservice architecture are microservices, its distinguishing behavior derives from the composition and coordination of its components via messages. Thus, the core of a microservice platform will be the orchestration phase, namely the composition of microservices, tools, and processes invoked, and the connection and automation of workflows that create the final service [20].

However, the microservice paradigm introduces also new challenges and enlarges known threats. The distributed programming nature is at the same time its main advantage (as shown earlier) and disadvantage. As described in [21], preventing programming errors in this context is harder, and the programmer has to face and work with different technologies that typically have different means of specifying contracts for the composition of services.

Other issues introduced by the microservice architecture are the ones related to the security aspects. These issues are certainly not new, as they apply to SOA and in general to distributed computing, but they become even more challenging in the context of microservices. Description of this kind of issues can be found in [22–24].

In Fig. 1 is represented a microservice platform, to enable the creation of mobility services. This architecture is inspired from our project SMAll [6].

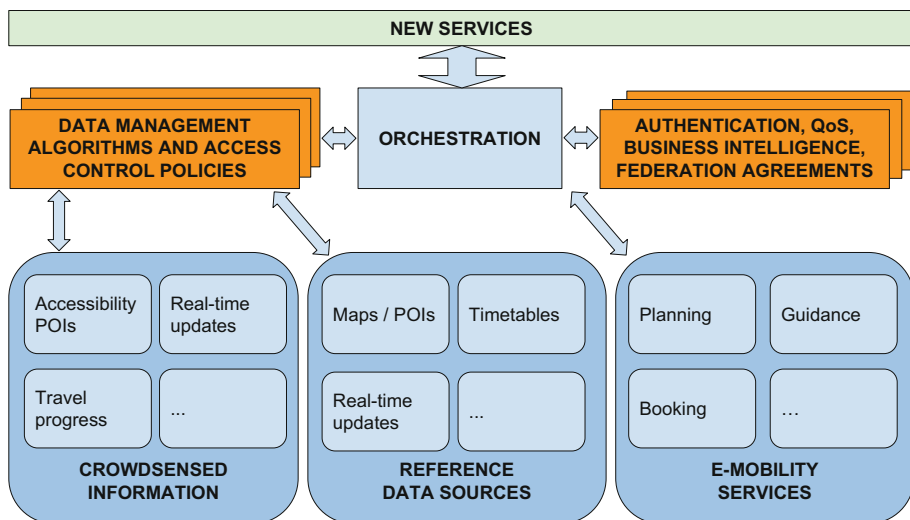


Fig. 1. A microservice platform to enable the creation of mobility services

4 A Use Case

In this section we describe a specific use case which has been developed by means of the SMALL platform. The goal of such a use case is to show how the microservice architecture can be exploited in the context of smart mobility, meeting different kinds of users' needs.

In particular, we present a scenario of a use case illustrating a blind user who requests a personalized path along the city of Bologna (Italy), by using his own smartphone. In particular, let us consider a male undergraduate student equipped with a white cane who uses to move across Bologna from home to reach the University campus and among offices and classrooms of the University, which are spread all over the city.

He has set up his profile with preferences related to the crossing facilities (he LIKES zebra crossings, traffic lights, and audible traffic lights, since it is much easier for him crossing streets by exploiting such urban elements), surfaces (he LIKES tactile pavings, which help him in walking on the street in an independent and safe way), bus and bus stops characteristics (he LIKES tactile information and acoustic cues and announcements), and other urban elements (such as pathways, obstructions and so on). He LIKES steps and stairs because they represent a landmark, helping him in orientating while he is wandering the city.

In this use case, our user asks for a specific path (including bus routes) starting from the School of Engineering and Architecture to reach the School of Art, Humanities and Cultural Heritage of the University of Bologna. The most commonly used geospatial mapping platforms (such as Google Maps) usually propose the path shown in Fig. 2.

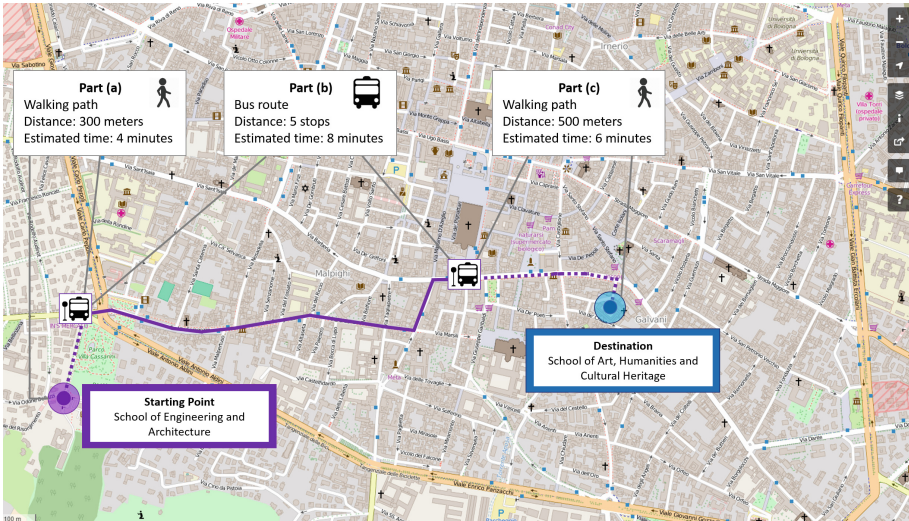


Fig. 2. The shortest path between the starting point and the destination

Such a path is the shortest one (it is expected to take 18min) and it is structured in three parts, as follows:

1. a walking part to reach the bus stop; this part is supposed to take 4 min (for a 300 m distance);
2. a part of a bus route (between the bus stop icons); this part is supposed to take 8 min (with five in-between stops);
3. another walking part from the arrival bus stop to the final destination; this part is supposed to take 6 min (for a 500 m distance).

Taking into account the preferences shown in Listing 1 (in JSON format), the path depicted in Fig. 2 presents some issues our blind citizen has to face: (i) the absence of tactile pavings and of acoustic traffic lights; (ii) the absence of tactile information and of acoustic cues at the bus stops; (iii) the absence of acoustic announcements and of tactile information on the bus of that line; (iv) the information about bus arrival time is derived from a time table, instead of referring to the real bus position and availability.

When our user asks for a path from the starting point to the destination, then our system computes a personalized route taking into account the user's profile (i.e. avoiding those barriers which affect him and including as much as possible the LIKEd facilities).

Our system computes a personalized path, by taking into account real data about bus availability and the user's profile, in terms of barriers to avoid and LIKEd facilities to include as much as possible. In particular, our system matches the user's preferences with the information about the aPOIs (accessibility Points of Interest [2]) all over the city and real time open data about the public means of

transport. A personalized path that meets these issues, computed by our system is shown in Fig. 3.

Such a personalized path is longer than the original one, and it is expected to take 24min. According to a survey [25], citizens would face a longer path in urban environments if it meets their preferences. In particular, 88% of them were ready to face a path up to 30% longer to reach their destination if such a route was tailored to their preferences and needs, while 12% of the users would face a personalized path more than 30% longer.

Listing 1. User’s profile fragment

```
{
  "user_profile": {
    "crossing": {
      "zebra_crossings": {
        "-type": "facility",
        "-pref": "like"
      }, "traffic_lights": {
        "-type": "facility",
        "-pref": "like"
      }, "audible_traffic_lights": {
        "-type": "facility",
        "-pref": "like"
      }
    }, "surface": {
      ...
      "tactile_paving": {
        "-type": "facility",
        "-pref": "like"
      }
    }, ...
    "bus_stop": {
      ...
      "tactile_info": {
        "-type": "facility",
        "-pref": "like"
      },
      "acoustic_cues": {
        "-type": "facility",
        "-pref": "like"
      }
    }, "bus": {
      ...
      "acoustic_stop_announcements": {
        "-type": "facility",
        "-pref": "like"
      },
      "audible_output_devices": {
        "-type": "facility",
        "-pref": "like"
      }
    }
  }
}
```

The personalized path of our use case is structured in three parts, which involve a different bus line in its second part. In particular, such a path lets the user exploits urban facilities he declared as “LIKE” in his profile (e.g. zebra crossings, tactile pavings, acoustic traffic lights, bus stops equipped with tactile and acoustic cues), which improve his independence in moving across the city. During the third part of his personalized route, the user meets two stairs. Incidentally, his smartphone senses them, thanks to our system, making the user also a provider of sensed data related to urban barriers.

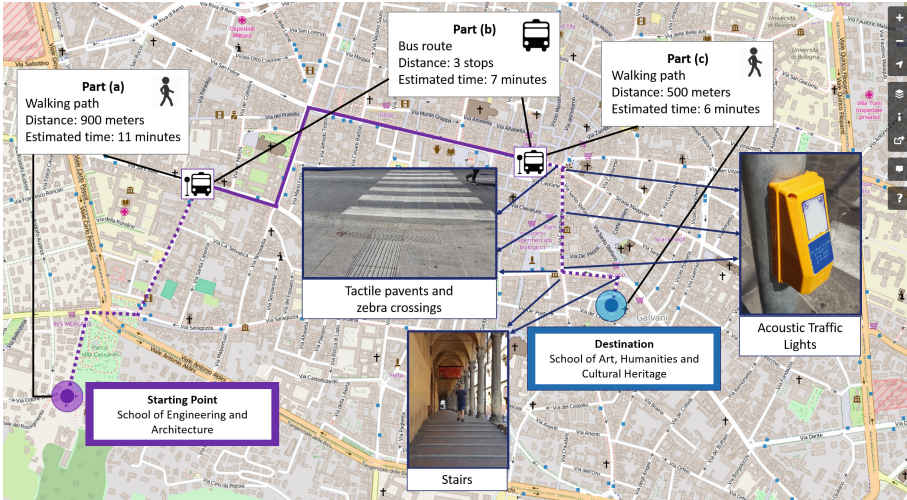


Fig. 3. The personalized path between the starting point and the destination

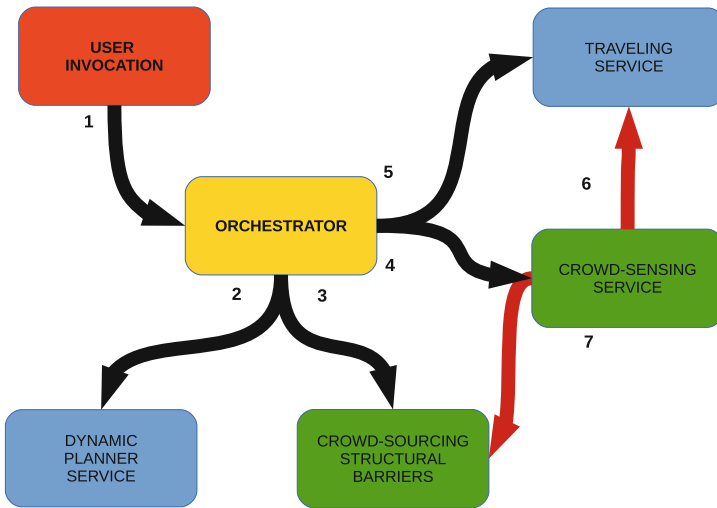


Fig. 4. The use case workflow represented as a microservice architecture in SMALL

A representation of the workflow of this use case in a microservice architecture is shown in Fig. 4. The user, in our case the blind user, invokes the routing service (1). The orchestrator intercept this call and he is the one in charge to create the correspondent workflow based on the invocation parameters. The invocation parameter are represented as the profile *LIKES*, and the orchestrator invoke the routing algorithms of the dynamic planner service (2) and the crowdsourcing service that map the structural barriers (3) based on the on this preferences. Once the trip has been proposed and choose the orchestrator call at the same

time the traveling service (5) and the crowdsensing one (4). The traveling service manage the assistance phase, guiding the blind user and exposing the information with in an appropriate manner, e.g. voice synthesizer. The crowdsensing service otherwise as twofold goal. Based on the hardware device used he will collect (possibly) new information about the path in place at that time, e.g. such as the presence of stairs, obstacles or other temporary interruptions. This information will be used to immediately interact with the user notifying him warnings or advisories and at the same time this information will feed the crowdsourcing service enriching its database for future requests.

5 Conclusion

Smart mobility is a key element to support citizens in their daily activities and to offer them a livable smart city. Information about urban transportation (including taxis, bus, train, car-sharing, etc.), urban barriers and facilities, pedestrian and multimodal paths would be of great benefit in this context, as well as all the information about the whole experience of traveling and wandering the city, including travel planning and payments. In order to provide a platform to manage such services and features, we designed and prototyped an infrastructure as a marketplace for mobility services, called Smart Mobility for All (SMALL). A prototype of such infrastructure has been developed and its microservice architecture has been described in the paper. A use case involving a blind user moving across a urban environment is detailed in the paper and it shows the feasibility and the potentialities of our approach.

References

1. Cardonha, C., Gallo, D., Avegliano, P., Herrmann, R., Koch, F., Borger, S.: A crowdsourcing platform for the construction of accessibility maps. In: Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility, p. 26. ACM (2013)
2. Mirri, S., Prandi, C., Salomoni, P.: A context-aware system for personalized and accessible pedestrian paths. In: 2014 International Conference on High Performance Computing and Simulation (HPCS), pp. 833–840. IEEE (2014)
3. Palazzi, C.E., Teodori, L., Rocchetti, M.: Path 2.0: a participatory system for the generation of accessible routes. In: 2010 IEEE International Conference on Multimedia and Expo (ICME), pp. 1707–1711. IEEE (2010)
4. Pippuri, S., Hietanen, S., Pyyhti, K.: Maas finland. maas.fi/
5. Banerjee, P., Friedrich, R., Bash, C., Goldsack, P., Huberman, B., Manley, J., Patel, C., Ranganathan, P., Veitch, A.: Everything as a service: powering the new information economy. *Computer* **3**, 36–43 (2011)
6. Melis, A., Mirri, S., Prandi, C., Prandini, M., Salomoni, P., Callegati, F.: Crowdsensing for smart mobility through a service-oriented architecture. In: 2016 IEEE International Conference on Smart Cities Conference (ISC2). IEEE (2016)
7. Melis, A., Prandini, M., Sartori, L., Callegati, F.: Public transportation, IoT, trust and urban habits. In: Bagnoli, F., Satsiou, A., Stavrakakis, I., Nesi, P., Pacini, G., Welp, Y., Tiropanis, T., DiFranzo, D. (eds.) INSCI 2016. LNCS, vol. 9934, pp. 318–325. Springer, Cham (2016). doi:[10.1007/978-3-319-45982-0_27](https://doi.org/10.1007/978-3-319-45982-0_27)

8. El Fassi, A., Awasthi, A., Viviani, M.: Evaluation of carsharing network's growth strategies through discrete event simulation. *Expert Syst. Appl.* **39**(8), 6692–6705 (2012)
9. Kupfer, D., Finger, M., Bert, N.: Mobility-as-a-Service: from the Helsinki experiment to a European model? Technical report, Observer European Transport Regulation (2015)
10. Sampo Hietanen, I.-F.: The new transport model? Technical report, MaaS Finland (2014)
11. Bujari, A., Licar, B., Palazzi, C.E.: Movement pattern recognition through smartphone's accelerometer. In: 2012 IEEE Consumer Communications and Networking Conference (CCNC), pp. 502–506. IEEE (2012)
12. Prandi, C., Ferretti, S., Mirri, S., Salomoni, P.: Trustworthiness in crowd-sensed and sourced georeferenced data. In: 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), pp. 402–407. IEEE (2015)
13. Callegati, F., Giallorenzo, S., Melis, A., Prandini, M.: Data security issues in maas-enabling platforms. In: International Forum on Research and Technologies for Society and Industry (2016)
14. Mirri, S., Prandi, C., Salomoni, P., Callegati, F., Campi, A.: On combining crowd-sourcing, sensing and open data for an accessible smart city. In: 2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies, pp. 294–299. IEEE (2014)
15. Fowler, M., Lewis, J.: Microservices. <http://martinfowler.com/articles/microservices.html>
16. Dragoni, N., Giallorenzo, S., Lluch-Lafuente, A., Mazzara, M., Montesi, F., Mustafin, R., Safina, L.: Microservices: yesterday, today, and tomorrow. *CoRR*, vol. abs/1606.04036 (2016)
17. Merkel, D.: Docker: lightweight linux containers for consistent development and deployment. *Linux J.* **2014**(239), 2 (2014)
18. Machado, R., El-Khoury, R.: *Monolithic Architecture*. Prestel Publishing, Munich (1995)
19. Newman, S.: *Building Microservices*. O'Reilly Media Inc., Sebastopol (2015)
20. Erl, T.: *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, Upper Saddle River (2005)
21. Christensen, E., Curbera, F., Meredith, G., Weerawarana, S.: *Web Services Description Language (WSDL) 1.1, Official Specification*, W3C (2001). <http://www.w3.org/TR/wsdl>. Accessed 15 Mar 2001
22. Greene, W.: Providing secure data and policy exchange between domains in a multi-domain grid by use of a service ecosystem facilitating uses such as supply-chain integration with RFID tagged items and barcodes. US Patent App. 11/069,479, 1 March 2005
23. Greene, W.: System and method for use of mobile policy agents and local services, within a geographically distributed service grid, to provide greater security via local intelligence and life-cycle management for RFLD tagged items. US Patent App. 10/913,887, 5 August 2004
24. Lea, G.: Microservices security: all the questions you should be asking (2015). <http://www.grahamlea.com/2015/07/microservices-security-questions/>
25. Mirri, S., Prandi, C., Salomoni, P.: Personalizing pedestrian accessible way-finding with mPASS. In: 2016 13th IEEE Annual Consumer Communications and Networking Conference (CCNC), pp. 1119–1124. IEEE (2016)