Mobile Netw Appl (2014) 19:161–170
DOI 10.1007/s11036-013-0488-1

# Cloud-assisted Computing for Event-driven Mobile Services

**Azzedine Boukerche · Antonio A. F. Loureiro · Eduardo F. Nakamura · Horacio A. B. F. Oliveira · Heitor S. Ramos · Leandro A. Villas**

Published online: 27 December 2013


**Abstract** Today, software developers for desktop computing build request and respond applications to do what end users tell them to do and answer what they ask. In mobile computing, software developers will need to develop sense and response applications that will interact with the end user. These applications will notify or ask users what they want based on what they have sensed or on a personal profile. Mobile cloud computing has the potential to empower mobile users with capabilities not found in mobile devices, combining different and heterogeneous data sets. In this work, we discuss the importance and challenges in designing event-driven mobile services that will detect conditions of interest to users and notify them accordingly.

A. Boukerche (✉)
Paradise Research Laboratory – EECS,
University of Ottawa, Ottawa, Canada
e-mail: boukerch@site.uottawa.ca

A. A. F. Loureiro
Department of Computer Science,
Federal University of Minas Gerais, Belo Horizonte,
Minas Gerais, Brazil

E. F. Nakamura · H. A. B. F. Oliveira
Department of Computer Science,
Federal University of Amazonas,
Maceió, Alagoas, Brazil

H. S. Ramos
Institute of Computing, Federal University of Alagoas,
Alagoas, Brazil

L. A. Villas
Institute of Computing, University of Campinas,
Campinas, São Paulo, Brazil

**Keywords** Cloud computing · Mobile services · Event-driven services

## 1 Introduction

We are starting to discover a new world for mobile computing where it seamlessly augments its capacity via cloud computing. By empowering mobile users with new enhanced experience, we have the opportunity to move to the ubiquitous access of information. The success of this vision relies on the understanding of three elements of a larger socio-technical ecosystem: the cloud, the mobile device, and the user. These three elements are the focus of this work, and they are deeply interleaved.

Nowadays, it is common for any person to store and access information on the cloud using a mobile device. Mobile users reading e-mail, interacting in social networks, putting Web queries, playing interactive games, or executing applications expect to continuously access those resources wherever and whenever they want. This technology disappears with the notion of *placeness* from computing since it is not possible to guarantee from someone else's perspective the correct location of a user when, for instance, filling a form in a Web page. Furthermore, it puts the notions of placeness and context in the forefront of users' wants and needs. In the not-so-distant past, computing resources demanded that users be in a certain physical place to access information. Now, users demand that information they access be relevant to their place and current needs, considering their physical and logical contexts. This is a tremendous change whose repercussions have only begun to be felt.

When we examine both the desktop and mobile computing paradigms we can better understand this fact. In a traditional desktop computing, users interact continuously with applications. On the other hand, in mobile computing we can expect to have a new breed of sense and respond applications that will detect conditions of interest to users and notify them through their mobile device. Today, software developers build most applications to run on the desktop or on the Web, but they are starting to build software that runs on multiple devices. Today, software developers for desktop computing build request and respond applications to do what end users tell them to do and answer what they ask. In mobile computing, software developers will need to develop sense and response applications that will interact with the end user. These applications will notify or ask users what they want based on what they have sensed or on a profile.

The main reason this ultimately will occur is that the delivery of timely and relevant information to users is a powerful value benefit for them in a mobile environment. All of us want to know immediately when something important happens and we do not want to be disturbed with non-critical notifications. In short, in this class of mobile applications the user interacts with a software system that continuously monitors the world obtaining data from a variety of heterogeneous sources, possibly using different sorts of devices, correlating multiple streams of events, some of them coming from the user's mobile device using a communication infrastructure while others from data sources in the cloud, and alerting the user only when necessary. This scenario is depicted in Fig. 1 that shows the *device*, which is used to collect heterogeneous data from different sources; *communication infrastructure*, responsible for all communication between the cloud and users; and *cloud*, which is responsible for all processing. This endorses the idea that placeness and context are two cornerstones to deliver personalized information to a given mobile user.

Of course, mobile devices are already used to browse the Web, read e-mail, run local applications and play games, which are important and useful for mobile users. Both proactive and reactive applications will encourage more people to carry mobile devices.

From a technical perspective, most of the technologies to design these applications are currently available. However, we still have many challenges ahead of us to fully deliver this promise, as discussed in this work.

## 2 Cloud computing

In the near future, we can expect to have a high availability of high-speed Internet access that will enable the delivery of new network-based services. Among them, cloud computing will have a fundamental role in the services to be offered to users.

The cloud computing service model involves the availability of possibly large amounts of high-performance computing resources and high-capacity storage devices to end users. There are different cloud service models, but in general, end users subscribe to the cloud service that is responsible for hosting their data providing computing resources on demand from the pool of infrastructure resources. The cloud service may also provide software applications required by the end user such as databases and applications. A successful cloud service needs a high-speed communication infrastructure to provide connectivity between end users and the cloud service.

Cloud computing can provide several advantages to both end users and cloud service providers [20]. End users can share a large set of computing resources, software and applications rather than owning and managing their own systems. Cloud service providers can potentially increase the use of their hardware and software resources [11], which need to be highly available using infrastructure and management systems, and charge end users a time-based or usage-based fee. Since along the time there will be active and inactive end users, cloud service providers can benefit from economies of scale, which often leads to a greater operational efficiency and a lower variable cost as well. On the other hand, end users can benefit from having data and services available at any location anytime, allowing them to focus on their tasks, really decoupling their activities from the operational aspects. This last point is certainly very important for many end users since it liberates them from the physical and costly management aspects.

There are different definitions of cloud computing, and discussion within the information technology industry continues over the possible services, standards and architectures to be offered to end users. The wide scope of cloud computing is succinctly summarized in [13]: "*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*." The proposed cloud model is composed of five essential characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service), three service models (Software as a Service – SaaS, Platform as a Service – PaaS, and Infrastructure as a Service – IaaS), and four deployment models (private cloud, community cloud, public cloud, and hybrid cloud).

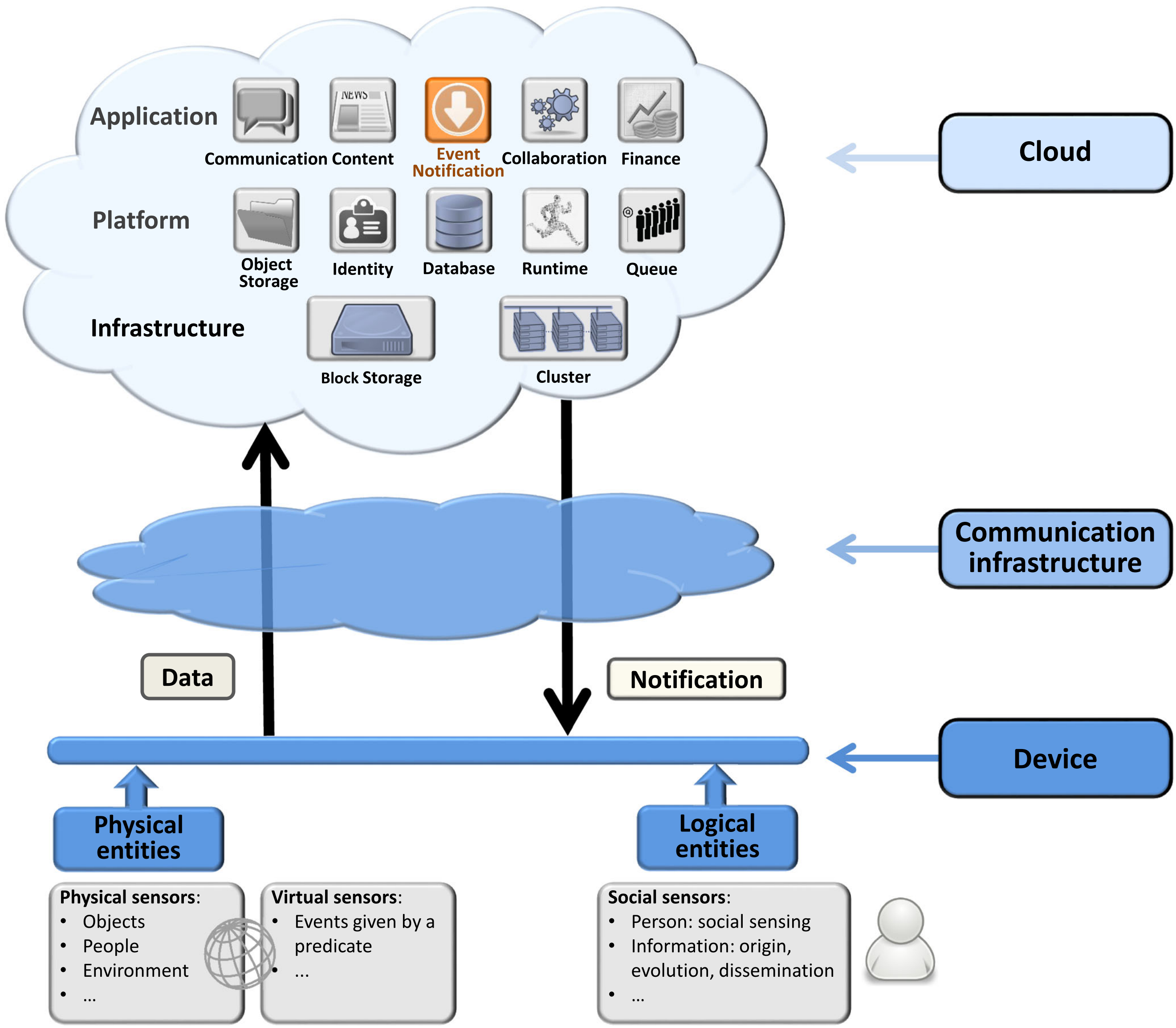


**Fig. 1** Cloud computing

## 3 Mobile devices and sensors

Sensing elements are becoming ubiquitous and can be found in many devices that are part of the hardware solution for different applications such as automotive industry, climate monitoring, health care, industrial control, oil exploration and petroleum industry, smart grid infrastructure, smart highway infrastructure, and smart homes, just to mention a few. The rapid spread of sensors into all domains of the human life is mainly due to the MEMS technology that reached a maturity stage and is becoming a mainstream. In the future, we can expect to have more and more products based on the MEMS technology, which will in turn power further penetration of sensors and new applications.

Currently, smartphones are being manufactured with an increasing number of powerful embedded sensors of different categories such as acoustic, sound, vibration (e.g., microphone), electric current, electric potential, magnetic, radio (e.g., magnetometer), navigation (e.g., digital compass, gyroscope), optical, light, imaging (e.g., ambient light, backside illumination, camera), position, angle, displacement, distance, speed, acceleration (e.g., accelerometer, GPS), pressure (e.g., barometer), and proximity, presence (e.g., proximity, touch). These are examples of sensors currently available in smartphones. In fact, in high-end smartphones, the number of sensors has already reached a half dozen and is quickly moving towards a dozen per device. Other sensors in different categories could be easily incorporated into a smartphone such as chemical (e.g., carbon dioxide and carbon monoxide) and thermal, heat (e.g., temperature). Sensors are extremely important to devise new smart features to all sorts of mobile devices.

More sensors on smartphones and other mobile devices mean more data flowing through these devices, and, eventually to the cloud with its vast data and processing facilities. It also means that processing them to lead to new information and knowledge is even more critical. Currently, smartphones are already used for many personal sensing applications [9] such as the monitoring of some physical exercise (e.g., running, walking) for personal use. In this case, the smartphone suffices to provide the user with the desired information since it has the embedded sensors and the corresponding application software.

Crowd sensing applications aim to monitor large-scale phenomena and require the active involvement of people to make available their sensed data [10]. We foresee that social sensing applications will be fundamental to the human interaction in the future, leveraging our awareness to different aspects of our lives. There are so many interesting opportunities and great challenges ahead of us in the deployment of mobile-driven mobile services for this class of mobile applications [17, 18].

## 4 End user and the case for event-driven applications

The extensive use of smartphones is one of the main factors for the consumerization of information technology. Consumerization means the tendency for new IT solutions to appear first in the consumer market and then disseminate into business and government organizations. There is no doubt that the advent of strong consumer markets is one of the main impulses for the information technology innovation and a major shift in the IT present in business and government. Consumerization is guided by end users who are taking technology innovation into their own hands, i.e., they acquire their own devices, install their own applications, contract their own services, and use different ways to connect to their friends and other people in different environments.

It is clear that the smartphone is here to stay with the associated computing paradigm it enables. Probably one of the main differences between the desktop computing paradigm and the mobile computing paradigm is the user's attention to the device. When working in a desktop, users are usually concentrated on the different activities they are carrying on, typically jumping among them by pressing a few keys or just clicking on an icon. Users's eyes usually rest on the desktop screen almost all the time.

On the other hand, in the mobile computing paradigm, users cannot afford to retain their attention to the mobile device's screen, mainly when they are moving. Possibly the canonical example of this fact is texting while driving, i.e., reading, composing, sending text messages, email, or making other similar use of an application on a mobile device while driving a vehicle. This practice has been reported as the cause of vehicle accidents, and in many places has been outlawed. As part of the initiative to combat distracted driving, the US Department of Transportation and the National Highway Traffic Safety Administration launched a website to advise the public about distraction-related accidents.[1] Thus, it behooves us to study and analyze the mobile computing paradigm to understand how to best exploit it and design appropriate and useful applications.

Yelp's Monocle[2] represents some of first consumer technologies that aggregate data collected from sensors and other pieces of information available at the cloud, and present contextually relevant information integrated with end users' current activities, surroundings and context. For instance, Yelp's Monocle uses GPS and compass to display markers for places (e.g., restaurants) on top of the camera's view.

## 5 Sensor fusion at different scales

Sensors such as 3D-accelerometer, 3D-gyroscope, and 3D-magnetometer are becoming standard features in mobile devices to provide users with an enhanced experience. The accelerometer sensor measures the $x$, $y$, and $z$ linear motion variables, the gyroscope sensor measures the pitch, roll, and yaw rotational angles, and the magnetometer sensor measures the $x$, $y$, and $z$ axis magnetic field. Each one of these sensors has powerful capabilities but also presents some limitations that impact their accuracy in applications. For instance, accelerometers are sensitive to vibrations and can generate a signal even when mobile devices are at rest, gyroscopes suffer from zero bias drift (signal output from the gyroscope when it is not experiencing any rotation), and magnetometers are sensitive to magnetic interference and can also create an undesired signal.

The signals from these three sensors can be sampled and passed directly to different applications such as a pedometer (application to count someone's steps as they walk). In this class of applications, the goal is to receive and process the signals individually, leading to simple and sometimes trivial solutions. As expected, the quality of the result will depend very much on the quality of the input signal.

[1] http://www.distraction.gov/

[2] http://www.yelp.com/

On the other hand, if the signals from these three sensors are sampled at the same time and processed properly, the shortcomings of each sensor can be overcome and a more useful output can be obtained. Data fusion algorithms are employed to obtain more sophisticated results whereby the output is more meaningful than the sum of individual data. A sensor fusion solution that combines data from those three sensors (3D-accelerometer, 3D-gyro and 3D-magnetometer) is called a 9-DoF (nine degrees of freedom) or 9-SFA (nine sensor fusion axis) solution. Compass applications, enhanced navigation and 3D-games are some applications that make use of the 9-DoF technique and provide users with a more improved experience.

It is very important to observe that we can continue this process at the same level or at upper and broader levels. For instance, if we add an additional sensing quantity such as barometric sensing to the 9-DoF technique, we have a 10-DoF solution. In this case, we can use a barometer to enable altitude detection between floors in a building leading to yet more enhanced navigation. Sensor fusion at this scale appears in some platforms for mobile devices as is the case of Windows 8.

The reality is that we have a plethora of mobile devices with a great variety of embedded sensors producing different sorts of data about the physical world, including ourselves, such as our blood pressure. On the other hand, we have users that can also input participatory data from their physical and logical contexts. The reality is that information is perceived and mixed in a number of ways, for a number of purposes, and at various scales. This broad spectrum of devices and sensed data is depicted in the lower part of Fig. 1.

Fusing data from multiple heterogeneous sensors can provide more complete situational awareness assessments for the end users. Heterogeneous sensors have the potential to produce a more comprehensive view of a given scenario by contributing either orthogonal or complementary measurements. Data fusion is defined as a framework for the processing of data originating from different sources in order to obtain higher quality information when compared with the individual sources.

## 6 Event-driven systems

In general, people have an intuitive idea of event, which is defined in [6] as the following: "an *event* is an occurrence within a particular system or domain; it is something that has happened, or is contemplated as having happened in that domain. The word *event* is also used to mean a programming entity that represents such an occurrence in a computing system". These two meanings are used in practice to represent occurrences in the real world that have happened or considered to have happened, and a programming entity in an event computing system.

Event processing systems collect, analyze, and react to real-world events as they occur. Basically, an ***event-driven system*** executes "rules" that generate appropriate responses, such as a notification, whenever the rule is "satisfied". By rule we mean a computational process for evaluating events according to a particular problem domain. How to collect, analyze and react poses the most interesting challenges in the design of event-driven systems. The analysis part, which is encapsulated by rules, are based on different techniques such as information fusion, machine learning, ontologies, probabilistic reasoning, data mining, databases, etc.

The Event Processing Agent (EPA) is the basic element of an event-driven system as depicted in Fig. 2, a simplification of Fig. 1. The input to the EPA consists of data collected by mobile devices and other heterogeneous data sources.

An event-driven system can be viewed as a set of interconnected event processing agents, having different functionalities as depicted in Fig. 3. This diagram shows a hierarchical interconnection of EPAs, just for the sake of illustration, and any other appropriate structure for the problem at hand could be envisioned. The point is that there will be different types of agents to perform specific tasks on data streams such as filter, transform, and pattern detection, considering a particular end user. As expected, this process can be further refined depending on the function of an agent, the input data stream and the notification to be sent to the mobile user.

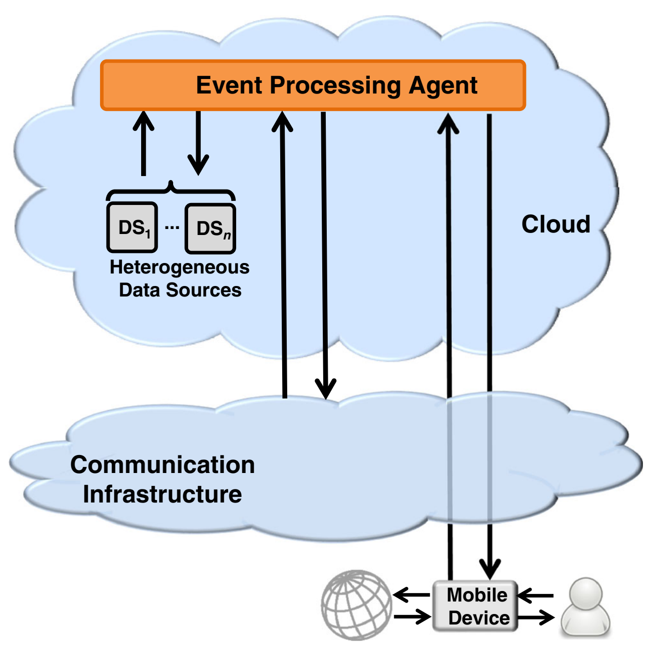


**Fig. 2** Event processing agent

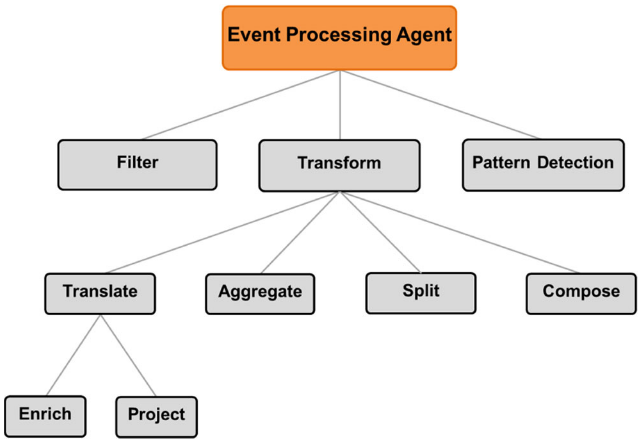


**Fig. 3** Types of event processing agents

Typically, the event processing agents that receive the first streams deal with simple filters and simple data models. Once this data is further processed, we have EPAs that deal with more complex models and lower data volumes. The characteristics of an event-driven system depend on different factors such as complexity and uniqueness of applications and performance requirements [15].

## 7 Mobile cloud computing

Cloud computing helps to bridge the resource gap of the mobile computing environment. For instance, Apple's iCloud can store automatically clients' files (e.g., music, photos, documents, etc.) and without any user intervention pushes them back to the user's device. We can say that Apple's iCloud is virtual since the real storage happens in Amazon EC2, a web service that provides resizable compute capacity in the cloud, and Microsoft Azure, Microsoft cloud computing platform. Recently, Amazon released Amazon Silk, a web browser developed for Kindle Fire that uses a split architecture to dynamically determine a division of tasks between the mobile device and EC2. The goal is to improve webpage loading performance, considering in this division some factors such as network conditions, page complexity and the location of any cached content. This is a new research area and is often referred as mobile cloud computing (mCloud computing) [2].

Mobile cloud computing has the potential to empower mobile users with some capabilities such as voice recognition and natural language processing, and computer vision, graphics and augmented reality. The restrictions on the mobile device's side and enormous availability on the cloud's side are doomed to stay, unless some technological breakthrough happens in the future, i.e., these are intrinsic limitations of the mobile computing paradigm. There is no doubt about the achievements and advances in the mobile computing technology in the last two decades. However, during this time the expectations and desires for this technology moved at a brisk pace.

It is seems clear that a very reasonable strategy is to devise solutions that combine the best of both paradigms: the great quality of current mobile devices with embedded sensors, friendly interfaces, and processing capability, on one side, and an incredible pool of both computing resources and data sources in the cloud, on the other side. Furthermore, given the distinct characteristics of mobile computing, in particular the limited user's attention to the mobile device, the ultimate goal is to deliver useful and important notifications to the user in a timely manner. This clearly justifies the need for offloading [8].

Event-driven services are closely associated with context-rich services that involve context extraction for user, device and communication infrastructure, recommendation based on user's desires and characteristics, and security aspects. Context extraction services perform some data mining analysis on heterogeneous data sets (e.g., social data, embedded sensor data, infrastructure data) to extract contextual traces related to a single user or a group of users. A basic contextual extraction service is to identify the user's activity based on embedded sensors (e.g., accelerometer, proximity), audio data and other data sources directly or indirectly related to the person (e.g., individual is in a room that provides an ambient light sensor) [14]. This will be a basic building block for other more sophisticated context-rich applications, as depicted in Fig. 3.

A set of contextual traces, defining a history for a person or group of people, is the basic element for other more elaborated event processing agents in the cloud. For instance, based on these traces, we can devise recommendation services that provide customized notifications to an individual or entire group based on their personal profiles. These recommendations can be very simple and creative and will help people to come up more interesting and useful services. For instance, combine the person's/group's current location with social data to create customized multimedia content [3], such as a song playlist or a recommended video [17, 18], that can be sent to a nearby audio jukebox or video screen aware of their presence.

A more complex solution is to consider different services that use different and heterogeneous data sources (e.g., social, mobile, infrastructure), process them, and output a customized notification [5]. Using event processing agents specialized in data mining and inference techniques for those particular data sets, we can extract contextual traces.

A key aspect of all these services is privacy that must be considered in context-aware mobile cloud services. The challenge in the design of those services is to determine the proper tradeoff between providing personal data to use a given contextual service and disclosing private information for that service. For instance, in general, people want to have location privacy and we have to trust the cloud that this information is not revealed to third parties. However, more sophisticated privacy services will be needed mainly to protect user data from data mining services that analyze contextual traces of individuals, or protect and anonymize those traces from social networks [4] and embedded sensors [7, 16]. Furthermore, we will need to consider not just an individual but a group of individuals as well when protecting their data.

## 8 Challenges

Some facts that can help us to understand the motivations and challenges in the design of event-driven mobile services are discussed in the following.

*Mobile devices are the norm.* As computing becomes a popular commodity, users look to mobile devices first. While fewer people consider desktops their primary device, more and more users choose a mobile-first strategy. Corporations like Google have announced that more of the company's focus is now on smartphones than on desktops, i.e., "Mobile First" [19]. Google developers are now creating versions of new services for smartphones before creating their counterparts to run on desktops. Recently, Microsoft announced Windows 8 that introduced significant changes to the operating system's platform, primarily targeting the user experience on mobile devices.

*The nature of applications is changing.* Applications are now uniquely mobile, designed to understand user identity and location, and make different sorts of predictions. A mobile device is not just a better option, but may be the only option for many people. This is similar to another phenomenon that has happened for some years now that is more and more people are ditching their landlines in favor of cellphones. When people get to that point, they tend to acquire not any cellphone but a smartphone.

*The density of mobile devices is accelerating the use of IPv6.* Exponential growth in mobile computing has created a need for additional blocks of IP addresses, which in turn is helping spur growth of IPv6 capable mobile devices. In fact, in some regions of the planet, like Europe and Asia, IPv6 is almost the only available option to connect new mobile devices to the Internet. On the other hand, the high availability of IPv6 addresses will boost the dissemination of Internet-based mobile devices.

*Application is increasingly important.* It is preeminent for the IT industry to understand the sort of applications people are using in their mobile devices. Since the experience of a "good" mobile computing environment is very recent for all involved players, the challenge is to understand what are the applications users are willing/want to have in their mobile devices. In this paper, we make the point that event-driven applications is undoubtedly an important class of applications for mobile computing.

The design of event-driven mobile applications establishes new challenges that we need to address to effectively use them. They can be broadly categorized into three groups: fundamentals, applications, infrastructure. Fundamental challenges deal with the theory, principles, techniques, methodologies, and tools to design them. Application challenges deal with the different aspects to be considered in the design of event-driven applications for the mobile environment. Finally, infrastructure challenges deal with communication aspects in event-driven applications.

### 8.1 Fundamentals

We can summarize the main challenge related to the fundamentals and principles in the design of event-driven mobile applications as the following: understand first potential applications, users' requirements and then propose the fundamentals. Usually, we advance the state of the art in a given area as a response to well-defined problems and challenges in that area. This is the main motivation for researchers to conduct applied research.

Once we have identified user's requirements, we need to design the event-driven mobile application using an event processing agent, which will use different functions (Fig. 3). As discussed above, we have to devise rules to generate appropriate notifications to the user based on different heterogeneous data sources, the communication infrastructure and the user's context. Depending on the analysis to be performed we need to use different techniques from different areas. However, we still need to consolidate the underlying theories [6, 15], which in turn will define the principles to be encapsulated in methodologies available to the designers as software tools.

Regarding some potential applications, let us consider the case of social networks that continue to make impressive strides, gaining more and more attention in our daily lives. Without entering into this discussion and tendency, we imagine that there will be scenarios where we want to receive different types of notifications from our social networks.

Social networks are primarily modeled and analyzed using graph theory where vertices represent people and edges represent the relationships between them. Research on social networks in the past half century has provided a wealth of results about the structure and evolution of these networks, including metrics and techniques for assessing, forecasting, and visualizing network behavior more generally. Much of the relevant results use social networks analysis in which the behavioral patterns and social interactions among people are evaluated using graph theoretic metrics. Social networks analysis defines a rich body ok knowledge based on graph theory, social sciences, economics, evolutionary biology, statistics, among others.

Most of the social network analysis happens on static snapshots of the network, usually targeting small communities such as those observed among a class of students or within a study group. In mobile computing, we need to better understand the network dynamics from the point of view of a mobile user. Furthermore, for this class of application, we need to introduce another elements, not present in traditional social network analysis, that are the user's context and desires, and the state of the communication infrastructure to send the person the most appropriate notification at a given moment. This is far from being a trivial enhancement to social network analysis that deserves to be investigated.

Another fundamental aspect related to the wireless infrastructure communication, broader than mobile computing, is a general capacity theory for wireless networks, which describes the fundamental performance limits of such networks [1]. These results are important to the event processing agent as depicted in Fig. 2, since the notification to the user should consider the state of the wireless communication infrastructure.

### 8.2 Event-driven mobile applications

*Useful, easy and interesting first* Identify useful and interesting event-driven driven applications to show the importance of designing this class of applications. To illustrate this point, let us consider applications for social networks and a sustainable society, certainly two categories of applications that call our attention.

Organizations usually have some polity regarding the use of social networks during working hours. However, a challenge is how to use social networks to enhance the work experience of employees at a given organization. Recently, IBM proposed Beehive, an internal social networking site, which provides IBMers a "rich connection to the people they work with" on both a personal and a professional level. The goal of Beehive is to help employees make new connections, track current friends and coworkers, and renew contacts with people they have worked with in the past. The natural question that arises is: what sorts of notifications should Beehive send to an employee considering the interests of both the organization and worker?

As the different actors in an organized society steadily act to discourage wasteful and environmentally unsustainable work and living practices, the ITC industry will play a central role in helping forging new practices and possibilities for the civil society as a whole. For instance, given a window of time (e.g., one hour), what is the best moment an application running in the cloud should notify a person to set foot to work or return home? This result should consider both the person's criteria to go to that place and all surrounding variables at that day/time (e.g., transportation schedule and traffic conditions). Basically, the result is a departure schedule not only for the person but for a group of related people, i.e., we need to consider a broader view rather than a local view. The idea with this application is to limit greenhouse gas emissions better orchestrating people's tasks, and, of course, saving their time. Many other scenarios could be devised in which we will need to propose environmentally-friendly solutions.

*Enhance user's experience using new gadgets* Besides smartphones, we can expect to have different gadgets to access the Internet in a mobile environment. An example in that direction is the different glass and watch projects. Its goal is to develop an augmented reality head-mounted display that allows pictorial communications to other people through images and video. The products of this project will allow people to quickly access information on the Internet [11, 20]. One of the first prototypes of this project resembles a pair of normal eyeglasses where the lens is replaced by a head-up display.

In the future, this gadget could be used as a canonical example to convey the idea that mobile users should receive only desired notifications in their devices. In this case, it does not make sense to start receiving all sorts of notifications that do not make sense when a person is working or just walking using this head-mounted display.

Since this will be a richer scenario, the challenge here is to design event-driven systems that consider the users' context together with the real environment surrounding them. Furthermore, the distraction time, when sending a notification to a person, should be minimized, i.e., notifications should be more meaningful.

*Cloud offload vs. local computing* Most of the modern smartphones can run complex applications that were commonly seen only in high-end desktop computers. Complex games, image processing applications, computer vision and video decoder, for instance, are now widely available in those devices. Despite the benefits, there is a tradeoff: this new scenario leads to a high energy burden most of the time higher than the phone's battery capacity [12]. The available technology for batteries is not evolving at the same pace as other components presented in the new smartphone generations, hence, energy consumption remains as one of the most challenging issues. To mitigate this problem, applications can take advantage of the high computing capacity available on the cloud, which means that code snippets can be offloaded to the cloud [8]. To do so, we might rely either on the programmer's ability for annotating the code indicating which part may migrate to the cloud, or on automatically migrating the code whenever it is worth it.

The challenge here is to design a management solution that collects meaningful information about the mobile device and applications running on it to decide whether the code snippet should be sent to the cloud or not.

### 8.3 Communication infrastructure

Traditionally, the wireless network is built on the principle of best effort. However, supposedly "good enough" networks are no longer good enough. In an ideal scenario, the wireless network must provide a high quality, consistent user experience by automatically recognizing applications and prioritizing "more important" traffic, as it might be the case when sending notifications to users. This means that they should receive notifications in a timely manner.

Mobile computing requires a network-centric computing model, based on a seamless access to multiple wireless data networks. Vertical handoff should be a basic feature of any mobile device since users want to roam among multiple wireless networks, based on different criteria such as connection price and QoS, in a manner that is completely transparent to applications and that disrupts connectivity as little as possible. Furthermore, the mobile computing era dictates that the wireless network will become the primary access network. To prepare for an event-driven mobile service, the communication infrastructure should be able to provide the event processing agent with information about the communication infrastructure (Fig. 2), so it can consider the network state when processing the notifications.

### 8.4 Some comments

At a higher level, cloud governance concerns the decision making processes and the set of criteria and policies regarding the planning, architecture, acquisition, deployment, operation and management of a cloud computing capability. The incorporation of governance to a mobile cloud computing based on event-driven services seems to be more difficult due to the challenges discussed above and the scalability of such systems when we consider millions of users.

This type of system continuously receives events from event sources that typically require real-time processing and low latency for event notification. Furthermore, events can arrive in bursts and out of order, be lost or intentionally omitted, and have imprecise timestamps. The processing of such events to provide mobile services usually concern relationships between events and their patterns, rather than with individual events. We can expect to combine data from multiple sources and infer from them a notification to be sent to a user based on both the arrival of new events and historical data.

All these challenges may represent a current limitation of cloud-assisted computing for event-driven mobile services but, on the other hand, have the potential to advance the state of the art becoming a reality this type of system.

## 9 Conclusion

The ubiquity of mobile devices, such as the smartphone, is owed, in part, to its multifunction capability that incorporates a series of tradeoffs that make it useful for many applications. It is becoming increasingly clear that the next generations of mobile devices will need new interaction and application models to facilitate new forms of communication and meet increasing user desires. Furthermore, we can expect to have a high availability of high-speed Internet access that will enable the delivery of new cloud-assisted computing for event-driven mobile services.

The design of those applications establishes new challenges that we need to address to effectively use them. In this work we discussed the importance of this class of applications in mobile computing, and some of the interesting opportunities and great challenges in designing this class of mobile applications.

## References

1. Andrews J et al (2008) Rethinking information theory for mobile ad hoc networks. IEEE Comm Mag 46(12):94–101
2. Bahl P, Han RY, Li LE, Satyanarayanan M (2012) Advancing the state of mobile cloud computing. In: Proceeding of the third ACM workshop on mobile cloud computing and services, pp 21–28
3. Beach A et al (2008) WhozThat? Evolving an ecosystem for context-aware mobile social networks. IEEE Netw 22(4):50–55
4. Beach A, Gartrell M, Han R (2010) q-Anon: rethinking anonymity for social networks. In: Proceeding of the 2010 IEEE second international conference on social computing, pp 185–192
5. Beach A et al (2010) Fusing mobile, sensor, and social data to fully enable context-aware computing. In: Proceeding of the eleventh workshop on mobile computing systems & applications, pp 60–65
6. Etzion O, Niblett P (2010) Event processing in action. Manning Publications Co, USA, p 384
7. Gilbert P et al (2010) Toward trustworthy mobile sensing. In: Proceeding of the eleventh workshop on mobile computing systems & applications, pp 31–36
8. Kumar K et al (2013) A survey of computation offloading for mobile systems. J Mob Netw Appl 18(1):129–140
9. Lane ND et al (2010) A survey of mobile phone sensing. IEEE Comm Mag 48(9):140–150

10. Lane ND et al (2011) Exploiting social networks for large-scale human behavior modeling. IEEE Pervasive Comput 10(4):45–53
11. Liu F et al (2013) Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications. IEEE Wireless Comm Mag 20(3):14–22
12. Liu J et al (2012) Energy efficient GPS sensing with cloud offloading. In: Proceeding of the 10th ACM conference on embedded network sensor systems, pp 85–98
13. Mell P, Grance T (2011) The NIST definition of cloud computing. NIST Special Publication 800-145. National Institute of Standards and Technology. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf
14. Miluzzo E et al (2008) Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In: Proceeding of the 6th ACM conference on embedded network sensor systems, pp 337–350
15. Mühl G, Ludger F, Pietzuch P (2006) Distributed event-based systems. Springer, Berlin, p 406
16. Saroiu S, Wolman A (2010) I am a sensor, and I approve this message. In: Proceeding of the eleventh workshop on mobile computing systems & applications, pp 37–42
17. Wang X et al (2013) Cloud-assisted adaptive video streaming and social-aware video prefetching for mobile users. IEEE Wirel Comm Mag 20(3):72–79
18. Wang X et al (2013) AMES-Cloud: a framework of adaptive mobile video streaming and efficient social video sharing in the clouds. IEEE Trans Mult 15(4):811–820
19. Wroblewski L (2012) Mobile first. A book apart. New York, p 144
20. Xu Y, Mao S (2013) A survey of mobile cloud computing for rich media applications. IEEE Wirel Comm Mag 20(3):46–53