Exploiting SAaaS in Smart City Scenarios

Salvatore Distefano[1], Giovanni Merlino[2,3], and Antonio Puliafito[2]

[1] Dip. di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milano, Italy
salvatore.distefano@polimi.it
[2] Dip. di Ingegneria, Università di Messina, 98166 Messina, Italy
{gmerlino,apuliafito}@unime.it
[3] Dip. di Ingegneria (DIEEI), Università di Catania, 95125 Catania, Italy
giovanni.merlino@dieei.unict.it

Abstract. Most of the current shortcomings in relation to Clouds made up of sensing resources can be addressed, as detailed in past work from the authors, following a Sensing and Actuation as a service (SAaaS) approach, i.e. enabling an Infrastructure-oriented (IaaS-like) provisioning model for sensors and actuators. The aforementioned approach matches the technological requirements and constraints springing around certain application domains belonging to the Future Internet research area. One of the most prominent scenarios is computer-assisted data treatment and automation of urban areas and public facilities, going under the umbrella term of a so-called Smart City.

Aim of this paper is to provide specific processes, rules and guidelines for the adoption of the SAaaS paradigm in IT infrastructure powering Smart Cities. Such artifacts are for assessing the feasibility of deploying Smart City applications over SAaaS, moreover the design of some specific use cases is presented and discussed.

Keywords: Cloud, sensors and actuators, Smart Cities, mobile crowdsensing.

1 Introduction

According to current ICT trends (Internet of Things, Future Internet) and estimates soon there are going to be myriads of devices dispersed and meshed into Internet-wide networks over geographic distances, calling for infrastructure management systems and facilities at previously never attempted before reach and scale. Yet up till now sensing resources to be involved in a Cloud have been envisioned only as data collection outlets. The authors took a different trajectory in [1] to tackle transducers in a Cloud environment: sensing and actuation devices should be handled along the same lines as computing and storage abstractions in traditional Clouds, i.e. virtualized and multiplexed over (scarce) hardware resources. This brings about a so-called "Sensing and Actuation as a Service" (SAaaS), a provisioning model where (virtual) sensors and actuators are to be exposed as Infrastructure and provided as a (Web) service to customers.

In a SAaaS-enabled scenario nodes can be either fixed or roaming, contributed on a voluntary basis by owners, thus joining and leaving the system at will, unpredictably.

D.-S. Huang et al. (Eds.): ICIC 2013, LNCS 7995, pp. 638–647, 2013.

Our device-driven approach implies giving customers full control over virtual resources under rental. Among interesting applications for SAaaS, Smart City-related ones could play a remarkable role. Sensing and network technologies are core elements for designing and building new Smart City facilities, as well as for monitoring and maintaining the existing ones. Moreover, production, business and social processes could be profiled to optimize, e.g., yields and reduce energy consumption and environmental impact.

The Smart City application domain is reaching a prominent position among innovation trends. Future Internet and Internet of things place Smart Cities as a key concept for future technological developments [2]. The transformational changes towards smarter cities will require innovation in planning, management, and operations. Several ongoing projects illustrate the opportunities and challenges of this evolutionary step [3] in diverse application fields, e.g., networking, decision support-systems, power grids, energy-aware platforms, service oriented architecture, highlighting the need to equip the cities of the future with different types of urban sensors. Plenty of applications can be envisioned for a Smart City, from traffic monitoring to energy management, from e-health to e-government, from crowd management to emergency management and disaster recovery, etc. envisioning and unlocking new forms, schemes and mechanisms of (micro/crowd-) business activities and funding. This all-encompassing and very ambitious scenario calls for adequate ICT technologies. In particular, infrastructural solutions for managing underlying physical sensing and actuation resources are required. This paper focuses on ways to deal with Smart City challenges through the SAaaS infrastructure. Selected guidelines, rules and algorithms are specified in the following, describing how to setup both the SAaaS infrastructure and the application, and also how to deploy the latter into the SAaaS infrastructure thus obtained.

Several works deal with issues related to Smart Cities. An interesting investigation about similarities and differences between "smart" and "digital" cities is provided by [4]. Starting from this, the authors of [5] propose a platform for managing urban services that include convenience, health, safety, and comfort. Also Cloud computing infrastructure recently found useful application in the context of Smart Cities [6-7]. Even if a lot of applications in the Smart City scenario have been proposed so far, there is a lack of common initiatives and strategies to address infrastructural issues. Indeed, the infrastructure adopted in such applications implements ad-hoc solutions. SAaaS aims at filling this gap between Smart City applications and the underlying infrastructure.

The remainder of the paper is thus organized as follows. Section 2 describes and investigates the Smart City scenario from the infrastructural perspective, contextualizing the SAaaS and providing a high-level setup process overview. Then, Section 3 details the first setup phase, the SAaaS provider and infrastructure setup, also briefly discussing the SAaaS approach and related modular architecture, while Section 4 deals with the application setup and deployment, detailing different Smart City use cases. Section 5 closes the paper with some final remarks and discussion.

2 Scenario

Urban performance currently depends not only on a city's endowment of hard infrastructure, but also, and increasingly so, on the availability and quality of communication infrastructures and services. ICT has a strategic role in this scenario, since it can optimize the distribution of knowledge and offer advanced technologies to create innovative services. The SAaaS is poised to be a very strategic solution to provide infrastructure for Smart City applications, where the involvement of both SN and mobiles would be a peculiar advantage of our vision, as in urban settings each kind of source can at a minimum serve for Cloudbursting or, in certain contexts, even augment the other due to diversification of resources and behaviors, i.e. motion patterns, user preferences, metadata, etc. In particular mobiles bring issues of random engagement to the Cloud that need to be addressed by means of volunteer mechanisms, and lead naturally to a multi-tenant scenario with regards to resource ownership. Moreover smart techniques for node management (e.g. energy optimization) are required when it comes to mobiles. Sensing resources are needed to satisfy application requirements that can continuously change to the environment conditions and the events occurring. This can be translated into a demand of high adaptability to the infrastructure, which has to provide adequate resources to promptly satisfy the requests. The SAaaS infrastructure can meet such demand by enrolling and dynamically managing the resources to satisfy fluctuating requirements. Indeed, a prominent distinction between other current sensing Clouds efforts and our vision is the offering of IaaS-like services based on sensors, i.e. proper sensing infrastructure. Here abstracting means offering homogeneous treatment of, and access to, devices, e.g. exporting SensorML-enabled representation of resources on (typically) standards-alien platforms with regards to sensing, e.g. Android, iOS, TinyOS, Contiki, etc. Leveraging abstraction, virtualization brings enablement of sensing resource morphing, composition or splitting, exposing virtualized instances of sensors, that have to be provided to the customer as devices, not exposing just data, i.e. the customer is going to be able to turn virtual knobs on the sensor handles she gets access to. The feasibility of this approach lies in the customizability of the software that drives sensing resources, i.e. our ability to offer a way to inject custom-made modules (even customer-provided possibly) into the systems in order to repurpose devices, thus both an SDK and a provider-assisted field deployment system are to be provided as part of this framework. Given the tools and infrastructure, a provider has everything in place to offer contextualization services, as per IaaS definition. Last but not least, a remarkable advantage of exposing devices is the ability to offer actuating resources as well, leading to fully feedback-enabled flows, whereas pure data-driven approaches obviously fail to engage this kind of devices.

Several case studies can be developed on a SAaaS infrastructure, where resources are gathered from diverse contributing nodes (SN, smart devices, laptops, PDA, etc.), geographically scattered across sites, as shown in Fig. 1. Building upon infrastructure services offered by an SAaaS-powered provider, a wide set of applications belonging to a Smart City scenario can be developed and deployed, among which we decided to focus on smart mobility and smart surveillance systems, also involving mobile crowdsensing as discussed in Section 4. Through these use cases we intend to demonstrate how SAaaS can meet outstanding requirements, which are enabling in terms of Smart City applications, such as devices abstraction, due to the great variety of devices; virtualization, due to customization and security needs; scalability, due to variable and not predictable expansions of Sensor Networks; flexibility, due to heterogeneous parameters to track; fault-tolerance, due to ensure services continuity.

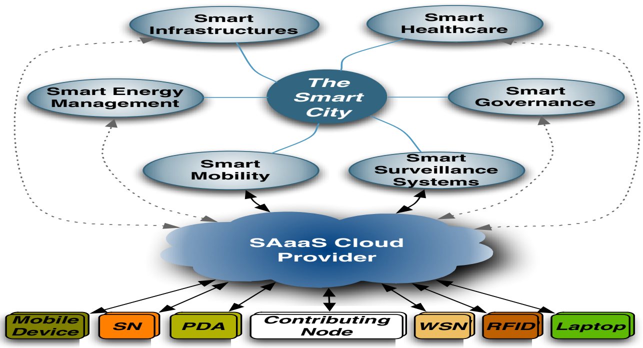

Fig. 1. Smart Cities scenario and use cases in SAaaS

This can be translated into a demand of high adaptability of the infrastructure, which has to provide adequate resources to promptly satisfy the requests. All those requirements find a natural outlet in a Cloud-based implementation that allows to adequately meet such demand by enrolling and dynamically managing the resources to satisfy fluctuating requirements. On a higher level, such an IaaS-like offering enables Software as a Service providers and end-users alike to detect potential dangerous conditions leading to emergencies (overloading conditions, black out, etc.), or to monitor crowded areas (traffic jams, sport events, safety and emergency services, etc.), just to name a few uses, where potential customers could therefore include law enforcement officials and emergency management agents.

To adopt the SAaaS approach in the Smart City scenario we can broadly identify two main phases:

1. *SAaaS Provisioning System & Infrastructure Setup* - the SAaaS provisioning system and the infrastructure should be set up in order to provide the required (virtual) sensing and actuation resources to the application, as described in Section 3.
2. *Application Deployment* - the software has to be deployed by first selecting and then customizing the (virtual) resources provided by the SAaaS infrastructure, which can be also merged to the ones owned by the application provider in a Cloudbursting hybrid-Cloud fashion, as described in Section 4.

It is important to remark that the SAaaS provider and the application/software (SaaS) provider generally don't overlap since, according to the Cloud provisioning model, infrastructure and software providers identify two different stakeholders. However, they can also coincide, i.e. the application/software (SaaS) provider can decide to build up an SAaaS infrastructure to deploy his/her application in a private Cloud fashion. On the other hand, in order to deploy an application into a SAaaS infrastructure it is not mandatory to first build up the infrastructure, therefore the SAaaS provider & infrastructure setup phase would not be a prerequisite in that case.

3 SAaaS Provisioning System and Infrastructure Setup

This section describes the first step of the workflow described in Section 2. With this aim, an overview of the SAaaS system is provided in Section 3.1, while setup details are discussed in Section 3.2.

3.1 Overview of SAaaS Sensing Cloud

SAaaS is a paradigm aimed at developing a sensing infrastructure based on sensors and actuators from both mobiles and SNs, in order to provide virtual sensing and actuation resources in a Cloud-like fashion.

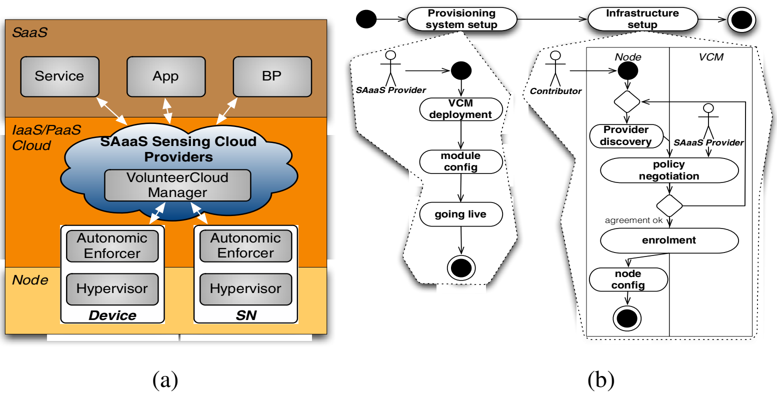

Fig. 2. Architectural schema and modules (a) and SAaaS Provisioning System and Infrastructure setup (b)

In order to build a Cloud of sensors in [1] we introduced the whole SAaaS stack and a rough schema of the architectural modules based on the functionalities to be provided and thus specifying the three main components shown in Fig. 2(a), i.e. Hypervisor, Autonomic Enforcer and VolunteerCloud Manager. The SAaaS stack and modules span to all the layers thus identified: from the highest Cloud one, providing PaaS support to SaaS software applications and services, to the IaaS infrastructure.

The lowest block of the stack, the Hypervisor one, operates at the level of a single node, where it abstracts the available sensors. The node can be a standalone resource-rich device, such as a smartphone, or it can be an embedded system which belongs to a network (such as a WSN). The main duties of the Hypervisor are: relaying commands and data retrieval, communications and networking, abstraction of devices and capabilities, virtualization of abstracted resources, customization, isolation, intra-node management facilities, semantic labeling and thing-enabled services.

The Cloud modules, under the guise of an Autonomic Enforcer and a Volunteer-Cloud Manager, deal with issues related to the interaction among nodes, belonging to a single Cloud, when generating a Cloud of Sensors. The former is tasked with enforcement of Cloud policies, local vs. global (i.e. relayed) policy tie-breaking, subscription management, cooperation on overlay instantiation, where possible through autonomic approaches. The latter is instead in charge of exposing the generated Cloud it hides by means of Web Service interfaces, framing reward mechanisms and policies in synergy with SLA matching, to be mediated by QoS metrics and monitoring, as well as indexing duties to allow for efficient discovery of resources.

3.2 Setup

In order to build up an SAaaS system it is first necessary to configure the SAaaS provisioning system managing the requests and therefore to enroll the underlying sensing physical nodes at the basis of the SAaaS provisioning model, after abstraction and virtualization. Thus, as shown in Fig 2(b), this process can be split into two main sequential steps: provisioning system and infrastructure setup.

The former step aims at establishing the SAaaS provisioning system, instantiating, deploying and configuring the upper level components of the architecture above described, i.e. the ones belonging to the Volunteer Cloud Manager (VCM) module, up to the SAaaS provider. More specifically, as shown on the left side of Fig. 2(b)_, the VCM modules have to be deployed into one or more physical machines (**VCM deployment**), exposed as a Web service. The VCM deployment could be performed either in a centralized or (hopefully) in a distributed way to avoid a single point of failure.

After deployment it is necessary to configure the VCM modules (**module config**). In particular the management policies (business models, SLA policies, etc.) should be specified as well as the user interface (frontend, protocols, knowledge base, etc.). Once the system is configured and ready to the delivery the SAaaS provider has to be exposed and indexed (**going live**) in order to be discovered and selected by both interested users and potential contributors.

The second phase, the infrastructure setup, then mainly aims at enrolling sensing resources and thus it is triggered by contributors. Indeed, a potential contributor that intends to share his/her sensing resources with the SAaaS community, has to first select a specific SAaaS provider (**provider discovery**) and therefore to start the negotiation process with the latter (**policy negotiation**), negotiating on his/her involvement, the SLA/credit/reward policies and similar issues. Such process (discovery + negotiation) is reiterated upon agreement by the involved parties. In case of successful agreement the node is enrolled into the SAaaS infrastructure (**enrolment**) and thus configured as required by the SAaaS provider and agreed by the parties (**node config**).

In this way, when a number of contributing resources provide their availability to the SAaaS provider, the system is ready to accept and manage requests of (virtual) sensing and actuation resources.

4 Smart City Application Deployment in SAaaS

4.1 SAaaS Application Setup

As depicted in Fig. 3, we have five (macro) actions involved in the deployment of custom applications powered by SAaaS for any service provider to start offering her software (SaaS) or platform (PaaS) over the Web. The deployment is kickstarted with a look-up (**resource search**) for sensing resources according to predefined constraints, as per requirements of the application to be offered as a service, and thus it is triggered by the app / platform provider. This phase is followed, upon selection of the resources to be employed, by instantiation of either physical or virtualized resources (**virtual instantiation**), to be carried out Cloud-side. As soon as the app provider can get hold of handles for these instances, the setup of the whole app can commence, first by setting up (app) server components and configuring them to reference the aforementioned handles (**server-side app setup**), and then sending requests for customization of selected nodes, where needed, to be acted upon by the SAaaS machinery (**node customization**). Upon completion and acknowledgement of customization duties, it's time for the app provider to expose its web service, possibly including any frontend or UI (**web service deployment**), to be put online for customers to peruse.

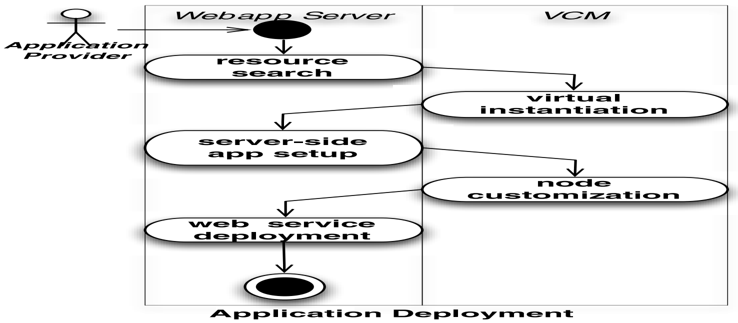

Fig. 3. Custom application deployment

4.2 Smart City UC Deployment on SAaaS

Here two compelling use cases, concerning mobility and homeland security, for SAaaS in a Smart City environment are discussed, where the novelty of our device-driven approach lends those feasibility and effectiveness. According to the proposed approach, in order to implement a Smart City application it is necessary to start from the underlying sensing infrastructure. As discussed in Section 3.2, the SAaaS infrastructure usually is up and ready to provide virtual sensors and actuators. But, in order to show the whole process, here we assume the SAaaS provisioning system and infrastructure has not yet been established and therefore we have to first set it up. We also assume the SAaaS infrastructure gets readied to cover all the applications described in the use cases below. Following the workflow of Fig. 2(b)_, the stakeholder that wants to build up the SAaaS infrastructure (a company, a community, etc., e.g. the ACME company) has to first setup the SAaaS provisioning system. The ACME SAaaS provider has to thus identify the physical computing infrastructure on which to deploy and configure the SAaaS modules, including the web service frontend. Then, the enrolment phase is triggered, involving the parties interested in contributing their sensing resources, negotiating with the ACME provider their involvement as shown on the right side of Fig. 2(b). We are particularly interested in GPS, accelerometers, cameras and similar sensors to implement the use cases. Thus, once the ACME SAaaS infrastructure is ready to serve handles on virtual sensors and actuators, the use cases can be deployed on it. According to the diagram depicting the deployment of custom applications, the activities that mandate an infrastructure-oriented paradigm like SAaaS are the instantiation of virtualized sensing resources and the customization of nodes, whereas none of the data-driven approaches to sensing Clouds are able to offer this kinds of functionality.

Table 1. Smart surveillance infrastructure: key elements

Kind of sensing node	***Associated services***
IP cameras / CCTV	- Motion detection - Presence detection (e.g. heat mapping) - Environment characterization (e.g. ambient light sensing)
Environmental & Mobile	- Motion detection (e.g. feature extraction from captures) - Presence detection (e.g. feature extraction from captures) - Environment characterization (e.g. ambient light sensing)

Smart Surveillance Systems. In a smart city setting, an outstanding selling point of a SAaaS-enabled infrastructure can be any application that fulfills homeland security requirements. A feasible one could be based on surveillance systems where, e.g., IP cameras, in their default configuration, are employed as CCTV, yet can be repurposed for, e.g., motion or presence detection (i.e. by means of heat mapping) as well as environmental characterization (i.e. aided by ambient light sensing and such), once enabled under the guise of Infrastructure as a Service, as soon as ACME has the infrastructure (e.g. IP cameras) enrolled and set up. This kind of flexibility can be achieved by means of customization of the virtualized instances of the devices being exposed, i.e., "beaming" software modules to the embedded platforms hosting the environment these cameras would be running upon, therefore shifting many higher-level duties to the smart sensing block. The customer will have the choice to move the knob in either way to set one's own favorite tradeoffs in terms of, e.g., higher bandwidth consumption coupled with lower processing on the embedded system vs. lower communication overhead while carrying on more (remote-side) computational duties, and so on. Once again, none of the above could be possible in a data-driven environment, as a prerequisite for repurposing resources is access to node customization facilities. The nodes, and their associated services, to be deployed for such use case are specified in Table 1.

Smart Traffic Control. One more offering leveraging SAaaS in a Smart City context could be in terms of traffic control facilities, by means of a Mobile Crowd Sensing (MCS) approach. Its quite obvious that mobile handsets can help a lot in a traffic control applications, as those devices are usually equipped with lots of relevant sensors, like those detecting position (GPS, WiFi, GSM) and motion (gyros), and are usually enabled for (always-on) bidirectional communication with the Internet, and with neighboring handsets as well.

Table 2 summarizes the parameters set of this use case.

Table 2. MCS-powered TC parameters set

Observed / Controlled Parameters	*Source (sensing)*	*Target (actuation)*
Geoposition (latitude, longitude, altitude)	mobile	-
Relative position (distance plus direction, orientation)	mobile	-
Motion detection (direction, orientation, rotation)	mobile	-
Scene capture	mobile	-
Human-Computer Interaction interfaces	-	mobile
Mobile platform notification system	-	mobile (virtual)

As information for traffic behavior profiling would come from users on the road, their mobility is key. Dealing with this kind of infrastructure i.e., privately owned mobiles, requires at the very least volunteer engagement and node management techniques, as pointed out before, but our efforts go far beyond, as we want to offer sensing Infrastructure under a Cloud provisioning scheme, leveraging what is a natural outcome of the scenario, i.e., multi-tenancy of resources, to be offered by one or more providers, tasked with gathering, managing and offering device-like resources to customers. Any effective road monitoring scheme would need at least a way to search (i.e., by area), filter and select sensors, including repurposed ones (i.e., cameras, with matching software to take shots autonomously, e.g., when certain events occur) by virtue of a Cloud-like provisioning model. At last, no full featured traffic control system may afford to lack mechanisms to influence traffic patterns, in our example to be implemented by feeding relevant info back to the contributors (e.g., road users) by means of (virtual) actuators, e.g., on-board signaling devices of any kind (buzzers, haptic interfaces) or other (platform) notification system piggybacking. Not to mention the possibilities disclosed by virtualizing those same devices, e.g., composing gyros and compass to expose a gyrocompass, or morph a camera into a motion detector, thus packaging up the virtual instances of sensing/actuation devices for the task at hand. Virtualization is thus achieved by means of contextualization, as unlocked by (virtual) device driver reconfiguration capabilities, where deployment takes the shape of provider-assisted packaging and delivery of customized modules for field use. In this case our ACME provider has to set up a Cloud made up of lots of mobiles for this kind of use case to be feasible. Here there are many facilities unique to the SAaaS approach being leveraged: whether we talk about virtualization, exploited on many levels, about actuating resources or even about customization, there's a lot of requirements for enabling such a scenario, and no way to tackle any of these issues by means of a data-driven approach, where they are totally out of scope.

5 Conclusions

Combining sensing resources and Cloud infrastructure is appealing if challenging, especially when tackling the problem from a novel IaaS-like perspective. SAaaS aims at bridging the gap between sensing pervasive scenarios and service-oriented provisioning models through a device-driven approach offering facilities such as virtualization and customization of sensing and actuation devices.

In this work we have focused on a Smart City scenario, comprising two use cases, i.e. Surveillance Systems and Traffic Control, to put SAaaS unique features up on display, and validate the usefulness of our approach against quite a compelling and trendy technology domain, the former all the more relevant when compared to other solutions for sensing Clouds, where much needed facilities are out of scope and outright unfeasible.

We believe that, as ongoing development efforts progress in parallel with those related to applications for any of these use cases, we will have the chance to further investigate and obtain proofs about our claims.

References

1. Distefano, S., Merlino, G., Puliafito, A.: Sensing and actuation as a service: A new development for clouds. In: Proceedings of the 2012 IEEE 11th International Symposium on Network Computing and Applications, NCA 2012, pp. 272–275. IEEE Computer Society, Washington, DC (2012)
2. Galache, J., Santana, J., Gutierrez, V., Sanchez, L., Sotres, P., Munoz, L.: Towards experimentation-service duality within a smart city scenario. In: 2012 9th Annual Conference on Wireless On-demand Network Systems and Services (WONS), pp. 175–181 (January 2012)
3. Naphade, M., Banavar, G., Harrison, C., Paraszczak, J., Morris, R.: Smarter cities and their innovation challenges. Computer 44(6), 32–39 (2011)
4. Su, K., Li, J., Fu, H.: Smart city and the applications. In: 2011 International Conference on Electronics, Communications and Control (ICECC), pp. 1028–1031 (September 2011)
5. Lee, J., Baik, S., Lee, C.: Building an integrated service management platform for ubiquitous cities. Computer 44(6), 56–63 (2011)
6. Li, Z., Chen, C., Wang, K.: Cloud computing for agent-based urban transportation systems. IEEE Intelligent Systems 26(1), 73–79 (2011)
7. Mitton, N., Papavassiliou, S., Puliafito, A., Trivedi, K.S.: Combining cloud and sensors in a smart city environment. EURASIP J. Wireless Comm. and Networking 2012, 247 (2012)