Project Acronym: STORM CLOUDS

Grant Agreement number: 621089

Project Title: STORM CLOUDS – Surfing Towards the Opportunity of Real Migration to CLOUD-based public Services

Deliverable 5.2

Book about public services migration to the cloud and the use of cloud-based public services in four European cities

Work Package: WP5

Version: 1.0

Date: 30/3/2017

Status:

Nature: Report

Dissemination Level: PUBLIC

Editor: Christina Kakderi, Nicos Komninos Panagiotis Tsarchopoulos (Aristoteleio Panepistimio Thessalonikis - AUTH)

Authors:

Contributors: Christina Kakderi, Nicos Komninos, Panagiotis Tsarchopoulos, Anastasia Panori (AUTH) Marco Battarra, Marco Consonni, Samuele De Domenico, Andrea Milani (HP), Agustín González-Quel (ASI), Miguel Tavares (ÁGUEDA), Dimitris Simitopoulos (THESSALONIKI), Julián Arroyo (VALLADOLID), Alkiviadis Giannakoulias (ED)

Reviewed by:

Version Control

Modified by	Date	Version	Comments
Panagiotis Tsarchopoulos	30/3/2017	1.0	The published papers of the special issue is bundled.

Table of Contents

1. Journal of Smart Cities

The Journal of Smart Cities offers a multidisciplinary forum for publishing original research in the area of smart cities, bringing together scholars from the fields of engineering, urban planning, computer science, and information and knowledge management. The Journal of Smart Cities will serve as an international platform by sharing the latest research achievements on smart cities globally, offering a comprehensive access to researchers and the academic communities.

The Journal of Smart Cities adopts a wide perspective on Smart Cities, as the broad use of information and telecommunication technologies to support an innovative and sustainable urban infrastructure and environment; as participatory action within cities, engagement of citizens, and investment in ICT, Internet technologies and smart objects, which increase the intelligence and problem solving capacity of cities; as an urban laboratory, a Living Lab, and a generator of solutions for wicked problems of inclusion, sustainability, climate change, and global competitiveness.

Main objectives of the Journal of Smart Cities is to promote research for the improvement of all sub-systems of cities, including urban infrastructure, construction, mobility, energy saving and renewable energy, health and social care, safety in the public space, work and innovation, and make the city easily accessible, effective, efficient, and an attractive place to live. Information and communication technologies, the Internet, and embedded devices into the physical space of cities is a means to achieve these objectives, which are mainly realized through the empowerment of city institutions, organizations, citizens, and end-users to make better, more informed, and intelligent decisions.

The Journal of Smart Cities aims to publish articles in all areas of smart cities, by scholars and experts in the field, but primarily from the fields of Civil and Environmental Engineering, Urban Planning, and Transport, while relevant contributions about the development of smart city technologies from the fields of Electrical Engineering, Informatics, and Computer Science are also welcome. Emphasizing on brilliant-quality research and timely publication, the Journal of Smart Cities invests in the management of best technological platform in an innovation context.

2. Special Issue on Smart Cities and Cloud Computing

The Journal of Smart Cities announced a call for papers for the special issue "**Smart Cities and Cloud Computing**" to be published on June 2016. Guest editors were: Christina Kakderi, Nicos Komninos, and P. Tsarchopoulos, researchers at URENIO Research. This issue will focus on smart city solutions and cities around the globe that develop strategies, infrastructure, applications and e-services using cloud technologies and business models. The focus is on the challenges and solutions related to the use of cloud computing for the development or the migration of smart city services to the cloud, such as the use of cloud-based infrastructure and platforms, adaptation of applications to cloud environment, distributed organization of resources, business models, data security, analytics and other.

Papers should be between 6.000 – 9.000 words covering aspects of literature, empirical survey findings, analysis and discussion. Detailed description of concrete examples of cloud-based solutions with a distinctive contribution to discussion will be appreciated. Also welcome are papers that refer to case studies from cities around the world that highlight how cities take advantage from cloud based solutions in infrastructure, platforms and application (IaaS, PaaS, SaaS); lessons learnt from experimentation; obstacles and enablers in adopting cloud technologies; use of specific business models; changes along the migration of existing e-services on the cloud.

We encouraged papers related to:

1. The stack of technologies enabling cloud-based smart city solutions
2. data management and security in public, private or hybrid clouds
3. innovation in city services or the deployment of new e-services with cloud solutions
4. changes in the urban system because of the introduction of cloud computing.

We also solicited methodologically oriented papers on new, non-traditional approaches to citizen-centric innovation for smart cities related to cloud-based platforms and services.

Particular topics to be addressed might include, but are not limited to the following:

- Cloud computing, service models and smart city solutions enabling innovation
- IaaS, PaaS, and SaaS for smart city solutions
- Cloud platforms and deployment of smart city applications
- Transformation of city services due to their migration the cloud
- Challenges of instant scalability of smart city services
- Security issues related to data protection and sensitive information concerning users
- Smart city business models adapted to cloud
- Costs and benefits from the deployment of smart city services on the cloud
- Plus and minus of public, private, and hybrid clouds for the deployment of smart city services
- Living lab experiments for cloud based smart city solutions
- Analytics and cloud-based services
- Standardisation and open interfaces of smart city systems, platforms and applications
- Case studies for cloud-base solutions of smart city services.

Time schedule

-Full paper submission: 31 January 2016.
-Notification of acceptance: 29 February 2016
-Revised submission: 31 March 2016
-Final acceptance notification: 30 April 2016
-Publication: June 2016

3. Published Papers

The published papers are available online as open access papers at:

http://ojs.whioce.com/index.php/jsc/issue/view/12

By the end of March 2017, the papers had 1.280 views on the publisher's website. The papers were also distributed on other academia web platforms such as ResearchGate.net.net and Academia.edu

EDITORIAL

Smart cities and cloud computing: Introduction to the special issue
Christina Kakderi, Nicos Komninos, Panagiotis Tsarchopoulos

RESEARCH ARTICLES

Smart cities and cloud computing: lessons from the STORM CLOUDS experiment
Christina Kakderi, Nicos Komninos, Panagiotis Tsarchopoulos

Storm Clouds Platform: a cloud computing platform for smart city applications
Marco Battarra, Marco Consonni, Samuele De Domenico, Andrea Milani

Social network services for innovative smart cities: the RADICAL platform approach
Fotis Aisopos, Antonios Litke, Magdalini Kardara, Konstantinos Tserpes, Pablo Martínez Campo, Theodora Varvarigou

Migration of applications to the Cloud: a user-driven approach
Anastasia Panori, Agustín González-Quel, Miguel Tavares, Dimitris Simitopoulos, Julián Arroyo

Cloud-based architectures for geo-located blogosphere dynamics detection
Athena Vakali, Stefanos Antaris, Maria Giatsoglou

Cloud computing security: protecting cloud-based smart city applications
Alkiviadis Giannakoulias

EDITORIAL

Smart cities and cloud computing: Introduction to the special issue

Guest Editors
Christina Kakderi, Nicos Komninos and Panagiotis Tsarchopoulos

URENIO Research, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

http://dx.doi.org/10.18063/JSC.2016.01.001.

The special issue "**Smart Cities and Cloud Computing**" of the *Journal of Smart Cities* focuses on smart city solutions that are deployed over various types of cloud environment and discuss challenges and solutions related to the use of cloud computing, and mainly the migration of smart city services to the Cloud.

The papers included are based on experiments that highlight how cities can take advantage from cloud based solutions in infrastructure, platforms and application (IaaS, PaaS, SaaS), lessons learnt from experimentation with cloud platforms, obstacles and enablers in using cloud technologies, and changes to applications and e-services along their migration to cloud environments. Some contributions also offer recommendations and guidelines to public authorities to help them in developing smart city services with cloud computing and make the required internal organisational changes to this end. Six papers and contributions by twenty-two authors are selected for the special issue. Following an introductory paper on smart cities and cloud computing, two papers discuss platforms and cloud-based environments for the deployment of services, two papers focus on cloud-based applications and big data that reveal trends and behaviours in smart cities, while the final paper looks at the challenging issue of data security in cloud-based environments.

The paper "**Smart cities and cloud computing: lessons from the STORM CLOUDS experiment**", by Christina Kakderi, Nicos Komninos and Panagiotis Tsarchopoulos, looks at the advantages that the cloud computing paradigm can offer to city governments for the deployment of smart city applications and services. The advantages are related to cost reduction, quality of e-services provision, enriching skills and infrastructure by outsourcing, adopting more efficient management of IT resources, and managing the large amount of heterogeneous data that are produced by the functioning of cities. The paper examines how these advantages appeared in the STORM CLOUDS project and its smart city experiments with respect to various tasks that were needed to develop or migrate applications to the Cloud. Main tasks for cloud-based deployment of smart city applications include the development of the cloud environment to host the applications, the portfolio of smart city applications and services, the roadmap for the migration of public services to the Cloud, and the business models for scalability and sustainability. The conclusions discuss both the general advantages of cloud computing and some new scientific directions and future challenges with regards to smart cities and cloud computing, and the specific lessons learnt from the platform created and smart city services deployed in the framework of the STORM CLOUDS project.

The paper "**Storm Clouds Platform: a cloud computing platform for smart city applications**", by Marco Battarra, Marco Consonni, Samuele De Domenico and Andrea Milani, describes a platform for running application services and the needed resources to be activated on-demand. In principle, any cloud-based application can run on a public cloud, like Amazon WebServices™ or Microsoft® Azure, which provide computational resources on a pay-per-use basis. However, there are some disadvantages in these large scale clouds due to their proprietary and global nature. Vendor lock-in is a problem, together with the lack of control on the location where applications run and

data are stored. This is an important barrier to cloud-based smart city solutions, in particular when applications manage personal data and the provider has legal obligation for securing data privacy. Given these challenges, the paper presents a new cloud computing platform that was developed on open-software components. It can be used either for implementing private clouds or for porting smart city applications to public clouds. The platform is based on the OpenStack, an open source software for creating private and public clouds, and offers a layered architecture in which the Infrastructure as a Service layer (IaaS) works as the foundation of the whole system. The IaaS layer provides basic IT capabilities like computing services (e.g., Virtual Machines), storage services (e.g., Virtual Volumes) and networking services (e.g., Virtual Networks) enabling the implementation of upper layers, such as the Data Service layer, the Access layer, and the Platform as a Service layer. The platform has been developed and tested successfully in the STORM CLOUDS project and all the software components used for the implementation are available under an Open Source Software license.

The paper "**Social network services for innovative smart cities: the RADICAL platform approach**", by Fotis Aisopos, Antonios Litke, Magdalini Kardara, Konstantinos Tserpes, Pablo Martínez Campo and Theodora Varvarigou, presents another platform for cloud-based smart city applications. The RADICAL platform is a software stack that enables the combination of social network services and Internet of Things (IoT) in the context of innovative smart city solutions. The benefits of combining social networking and IoT in smart cities have been discussed in the literature, yet there is no easy way to develop, customise, deploy and operate such services. The RADICAL platform focuses on this challenge and makes possible to combine multi-sensory and socially-aware services for smart city services. Core component of the platform is a Virtual Machine mechanism which works in a sophisticated cloud environment. The architecture of the platform includes three distinct layers: from top to bottom, the Service Application layer, the Platform layer, and the Data Source layer, where IoT devices, smartphone city apps and various Social Networks provide data. Eight pilot services were tested upon the RADICAL platform dealing with cycling safety, products carbon footprint management, object-driven data journalism, participatory urbanism, augmented reality, eco-consciousness, sound mapping in the city, and crowdsourcing for collecting movement information from citizens smartphones. A large number of citizens from different countries were actively involved in the co-creation, validation and evaluation of the platform, using Living Lab methods for collaborative product development.

The paper "**Migration of applications to the Cloud: a user-driven approach**", by Anastasia Panori, Agustín González-Quel, Miguel Tavares, Dimitris Simitopoulos and Julián Arroyo, focuses on the migration of existing smart city services to cloud infrastructure (IaaS), the selection of most suitable applications, as well as their adaptation to fit in the new cloud environment. The paper presents the main findings of a migration process that has been followed in the cities of Agueda (Portugal), Thessaloniki (Greece) and Valladolid (Spain). The migration process consists of four stages, which include actions dealing with the selection of applications/services to be migrated to the Cloud, technical or procedural problems related to adaptation of the applications to cloud environment, and the migration itself of applications to the Cloud. The definition of monitoring indicators for assessing the results of the migration process constitutes an additional aspect of the proposed procedure. A roadmap is designed that codifies the main instances of the migration process. The paper discusses the three case studies where a migration process took place and offers some advice that might guide cities and city governments to transfer applications and services they provide to cloud environments and take advantage of the cloud computing paradigm.

The paper "**Cloud-based architectures for geo-located blogosphere dynamics detection**", by Athena Vakali, Stefanos Antaris and Maria Giatsoglou, focuses on microblogging content under cloud-based infrastructure. Social networking data emerge rapidly and crowd-driven big data streams are valuable for detecting trends, opinions and citizen behaviour in smart cities. This type of information is of major importance, due to the fact that trends can be utilised to spot collective emergent or evolving patterns and phenomena. However, in such cases, conventional data mining approaches are inadequate to deal with the dimensionality and scalability of data. The Cloud4Trends framework, proposed in the paper, for collecting and analysing geo-located microblogging content under cloud-based infrastructure addresses this challenge. The proposed process involves three series of tasks: (i) collection of data in a streaming manner from Twitter as well as from a pool of selected blogs focused on a number of

geographic areas, (ii) application of an online data clustering method to detect trending topics, and (iii) refinement and ranking of clusters that trends are detected and visualised. The paper shows that Cloud4-Trends offers a viable solution for online social web geo-located and time-related data mining applications, enabling massive data analysis and reducing the need for real-time applications data processing time.

Finally, the paper "**Cloud computing security: protecting cloud-based smart city applications**", by Alkiviadis Giannakoulias, discusses a key challenge to all cloud-based solutions, that of data security. This is a major concern in cloud computing as such environments offer scope for intruders to attack. Cloud computing security is a rising sub-domain of cyber or computer security and refers to a broad set of policies, technologies, and controls used to protect data, applications, and the cloud infrastructure. Security issues related to cloud computing can be classified in two categories, those of cloud service providers and those of cloud users and customers. In particular, the paper focuses on security issues that arise when public sector organisations consider transitioning to an IaaS cloud. The paper examines legal implications; regulatory and standards compliance; vulnerabilities in virtualisation technologies; data integrity issues; and security checks to be performed on the services prior to their movement to the Cloud. The paper lists and explains twenty-one different types of cloud computing security threats, from ease of use to vulnerable data transmission, insecure APIs, malicious insiders to data loss, data breach, and authentication and authorisation. Then security recommendations and the virtualisation of security are presented. This analysis of security challenges leads to a series of recommendations concerning the operating system and firewall software, introspection in public deployment models, regular checks for new updates, and channelling in-going and out-going traffic to virtual machines.

Altogether, the six papers of the special issue discuss three main pillars of developing or moving smart city applications and services to cloud environments. First, the design and deployment of the cloud environment that will host the applications, and the related issues of data management, security, interoperability, second, the selection and adaptation of applications to fit with the type of the cloud environment selected (IaaS, PaaS, SaaS), and third, the use of data mining and analytics to gain insight from the existence of large volume data on the Cloud. These three dimensions of cloud-based smart city applications and services should be addressed concurrently, taking into account interdependences, interoperability, and adaptation of applications to the type of cloud.

RESEARCH ARTICLE

Smart cities and cloud computing: lessons from the STORM CLOUDS experiment

Christina Kakderi*, Nicos Komninos and Panagiotis Tsarchopoulos

URENIO Research, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

Abstract: Since the emergence of cloud computing paradigm, there has been an increasing interest on the adoption of cloud computing from municipalities and city governments towards their effort to address complex urban problems. This paper explores the significant role that cloud computing can play in helping cities on their way to become smart. We focus on the STORM CLOUDS paradigm as a solution for municipalities everywhere in order to (i) deploy a portfolio of smart cities applications related to governance, economy and quality of life on a single cloud-based platform and (ii) use the platform and its accompanied tools to migrate their existing applications to the cloud environment. Besides the conclusions from the STORM experience, the paper closes with a number of research trends and future challenges that are expected to define the adoption of cloud computing from municipalities and city governments in the following years.

Keywords: smart cities, cloud computing, cloud platform, portfolio of cloud-based services, applications' migration

*Correspondence to: Christina Kakderi, URENIO Research, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece; Email: christina@urenio.org

Received: February 10, 2016; **Accepted:** April 29, 2016; **Published Online:** May 10, 2016

Citation: Kakderi C, Komninos N and Tsarchopoulos P, 2016, Smart cities and cloud computing: lessons from the STORM CLOUDS experiment. *Journal of Smart Cities*, vol.2(1): 4–13. http://dx.doi.org/10.18063/JSC.2016.01.002.

1. Introduction

Cloud computing has emerged during the last years as a disruptive model, with the ability to transform IT organisations, helping them to become more responsive and agile than ever before. Due to its multiple and significant benefits, cloud computing can particularly help complex organisations such as cities run more efficiently, providing new opportunities and opening up new business models. Although clearly at an early stage, the discussion on how cloud computing can help cities towards their effort of becoming smart has revealed a plethora of applications regardless of the size and level of organisation, and wealth of a city. The paper aims to contribute to this discourse both at the theoretical and empirical level, firstly, by reviewing the literature on the cloud computing paradigm and the way smart cities can benefit from it, secondly, by introducing a cloud-based solution for smart city services and thirdly, by doing a short foresight on future research challenges with relation to cloud computing adoption by smart cities.

The cloud computing model represents a fundamental change in the way that information technology hardware and software are created, developed, deployed, scaled, updated, maintained and paid[1]. It serves as an enormous step towards delivering global computing as a utility (like traditional utilities such as water, electricity and telephony)[2] by changing the traditional access model, where data and applications are fully contained in the same physical location (local computing), to a new one, where the users access their data and applications outside their own local computing environment through the internet.

The US Government's National Institute of Standards and Technology (NIST) defines cloud computing

as "*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*"[3], while the IT research and advisory company Gartner uses a simplified definition and defines cloud computing "*as a style of computing in which scalable and elastic IT-enabled capabilities are delivered as a service using Internet technologies*"[4]. A common analogy to understand cloud computing is renting versus buying. Actually, someone rents IT capacity (computing power, disc space, applications, etc.) from a cloud service provider and consume them over the internet, instead of buying his own IT equipment. Moreover, he pays only for the resources he consumes.

Cloud computing promises economic benefits, speed, agility, flexibility, rapid elasticity and more innovation. The motivations of the organisations for the migration of their applications to the Cloud are closely related to the following key cloud computing characteristics:

- **On-demand self-service.** A consumer can unilaterally provision IT resources (e.g., storage, processing power, memory, bandwidth, etc.), as needed automatically without requiring human interaction with the cloud provider.
- **Broad network access.** IT resources are available over the Internet and accessed through heterogeneous devices (e.g., mobile phones, tablets, laptops, and workstations). There is a sense of location-independence because the customer generally is not aware of the exact location of the provided resources.
- **Rapid elasticity.** A cloud environment offers to the consumer the ability to rapidly scale up or down the IT infrastructure commensurate with demand. To the consumer, the capabilities available for provisioning usually appear to be limitless and can be reserved in any quantity at any time.
- **Measured service.** Cloud systems monitor, control and report the use of IT resources by leveraging a metering capability at some level of abstraction suitable for the type of service. This leads to a transparent relationship between the consumer and provider of the cloud service.

The above-mentioned characteristics create a highly efficient, scalable and elastic computing environment, which is available through a business model where consumers buy only the capacity and capabilities needed at any given time, instead of buying and deploying all the components of a computing centre.

Using this **"pay for what you consume"** approach, an organisation can significantly reduce the up-front costs by avoiding the procurement of hardware and software in advance, as well as the resultant infrastructure depreciation. Moreover, cloud adoption can decrease the operational costs, as the organization will maintain at a given time only the required resources, which could be scaled up and down through rapid elasticity. Also, other IT costs can be reduced by buying general purpose capabilities such as asset management, security, collaboration, etc. as a service, instead of maintaining a specialised in-house IT department. As organisations operate in an unstable economic environment, smart consumption-based procurement allows them to scale up to fulfil new demands and reduce spending, if necessary, to address changes in budgets and funding.

Besides IT cost reduction, the adoption of cloud computing increases significantly the effectiveness of an organisation in fulfilling its mission. The following benefits are clearly related to the ability of an organisation to deliver more and better results[5,6]: (i) Just-in-time infrastructure, (ii) more efficient resource utilisation, (iii) agility to respond to emerging needs, (iv) increased reliability and performance through the adoption of new technologies, best practices and security enhancements, (v) better information sharing and collaboration through the heterogeneous, location-dependent access, and (vi) evaluation and optimisation of business processes through enhanced real-time visibility and audit of applications and infrastructure.

Cloud computing can also serve as an enabler for innovation in the organisations that adopt it in a lot of ways: (i) reduces time to market by allowing the scale-up of resources in a cost-effective manner, (ii) gives constant access to commercial best practices and new capabilities, which can be incorporated to already existed services, instead of developing them in-house, (iii) enables the launching of new initiatives based on new applications that are available from the cloud provider, and (iv) connects to new and emerging technologies. Moreover, in government organisations, cloud computing can: (i) encourage entrepreneurial culture by reducing the risk of launching of new initiatives and by allowing the low-cost experimentation in new applications and services, and (ii) give them access to

innovations developed by the private sector[7].

Organisations should also take into account, when they plan on their Cloud strategy, the deferent service categories and deployment models of cloud computing[3]. Table 1 summarises the options.

As organisations start migrating their applications to the Cloud, it is important to determine which applications fit better into this environment. The best candidates are the applications taking advantage of the cloud computing elasticity. For example, applications that are designed to spread their workload across multiple servers, applications that run occasionally but require significant computing resources when they run, and applications with unpredictable or cyclical usage patterns, will benefit from the Cloud's ability to automate the dynamicity of resources to match the current demand. For these applications, the rapid elasticity combined with the pay-by-usage characteristic of the Cloud can lead to significant financial savings[8].

2. Smart Cities and Cloud Computing

City governments and municipalities everywhere constitute for one thing complex public organisations that have more reasons to invest in cloud computing than any other public organisation. Despite their limited resources they have to provide a wide number of municipal services (ranging from sanitation, water, schools, health, transportation etc.) and serve the needs of their citizens in their daily life. At the same time, they face a variety of challenges including job creation, economic growth and environmental pollution. While it is widely accepted that increasing urbanisation strains the limited resources of cities and affects their resilience, it also highlights the need for sustainable urban development; especially in terms of much more efficient management of natural resources, such as energy and water, as well as better planning and collaborative decision making[9]. In this context, cloud computing can play a significant role, facilitating cities in meeting the above-mentioned tasks.

Over the past years, the term 'smart cities' has evolved to denote the cognitive processes combined with the deployment of ICTs, institutional settings for innovation and physical infrastructure, which taken altogether increased the problem solving capability of a city or a community[10]. The main features of a smart city include applications that connect, manage and optimise data from a complex set of devices, sensors, people and software, creating real-time, context-specific information intelligence and analytics, which aim to transform the urban environment and address its specific needs[11]. Managing such enormous amount of heterogeneous data requires, among others, high storage capacity and performance computing power[12]. For this, the latest developments in cloud computing and the Internet of Things (IoT) are widely deployed in smart cities[13].

More specifically, smart cities have to use a wide variety of ICT solutions to deal with urban problems and monitor their functions; they do not only require the use of new technologies and devices such as sensors, RFID (radio-frequency identification) devices, smartphones, smart household appliances, etc., to collect land use, transport, census, and environmental monitoring data which are generated every minute in the urban environment, but also the capacity to manage and process all this large scale data (Big data) in real time, in an interconnected and service/applications' specific way[14]. The emergence of cloud computing paradigm facilitates big data storage and big data integration, visualisation, processing and analysis in acceptable time frames. Fu, Jia and Hao[12] listed six reasons for the convergence of cloud and IoT in smart cities: (i) the immense storage capacities of cloud computing infrastructure, (ii) central processing capacity to perform complex computing, (iii) the ability for dynamic reconfiguration of resources, securing sufficient computational resources at any time, (iv) the

Table 1. Cloud computing service categories and deployment models

Service Categories	Deployment Models
• *Software-as-a-Service (SaaS):* The capability provided to the consumer is to use the provider's applications by running on a cloud infrastructure. • *Platform-as-a-Service (PaaS):* The capability provided to the consumer is to use the provider's development platform in order to create, test and host new applications. • *Infrastructure-as-a-Service (IaaS):* The capability provided to the consumer is for provisioning the process, storage, networks, and other fundamental computing resources in order to build a customised computing environment.	• *Private Cloud:* The cloud infrastructure is used exclusively for internal applications within an organisation. • *Community Cloud:* The cloud infrastructure is used exclusively by multiple organisations that have similar interests for collaboration. • *Public Cloud:* the general public access a provisioned cloud infrastructure for open use. • *Hybrid Cloud:* The cloud infrastructure is a composition of two or more distinct cloud models (private, community, or public). Data can be easily transferred between the different infrastructures.

ability to conduct system level comprehensive analysis with sample data, (v) high accessibility to various objects in IoT through user friendly applications and customised portals and (vi) high speed network and disaster recovery capabilities. Most significantly, cloud based big data mining and analytic tools can deal effectively with multi-disciplinary city data, characterised by environmental dynamism and spatiotemporal attributes, and formulate a variety of smart city application scenarios[9,15].

Apart from tackling persistent urban problems, cities of today also feel significant pressures to become sustainable and energy efficient. Cloud computing opens up new possibilities for sustainable solutions; it is not only the cloud's economies of scale which contribute to economic and environmental sustainability, it is also the fact that sustainability of future cities is mainly based on their ability to manage increasingly large and complex data (on the environment, waste, water usage etc.), a task that can be performed more effectively through the cloud. Smart buildings for example, which constitute the core of smart cities, need to rely on enterprise scalable cloud architecture so that they can benefit from functions such as stream analytics, machine learning, Hadoop, Spark, etc.[16]. Technologies in place today enable designers to integrate technical specification data about the materials, systems and equipment to yield greater efficiencies in terms of energy performance and better management throughout a buildings' lifetime. Just as any other aspect of a city, in order to manage buildings efficiently one has to metered its sub-systems such as lighting, electrical, mechanical, security etc. in an instrumented and unifying way; in fact, to do it on an aggregate level, for a group of buildings and across neighborhoods. This requires the ability to access, collect and analyse a large volume of mostly private data, which can be done by cloud computing in a more efficient and cost-effective way than traditionally dedicated computing solutions[17].

Real time processing of big data and the deployment of multiple applications is possible due to virtualisation, which is one of the main enabling technologies of cloud computing. Virtualisation software abstracts the physical infrastructures as virtual machines (VMs) and makes servers, workstations, storage and other systems independent of the hardware layer creating various dedicated resources according to the needs of the users. It increases infrastructure utilisation, enables more efficient use of the hardware and allows for true scalability and increased uptime[18].

Smart city solutions are applicable to all three service models of cloud computing (IaaS, PaaS, SaaS). Firstly, as already mentioned, cloud solution providers integrate IoT infrastructure (devices, networks) through virtualisation, offering computer resources as a utility (IaaS).

Secondly, cloud computing can shift domain specific tightly coupled systems (i.e., systems focusing on energy, transportation etc., orchestrated by domain specific service providers) to open domain-independent and scalable smart city services, provided through cloud based and domain-independent service-delivery platforms[19]. These constitute a type of Platform as a Service (PaaS) offering, which integrates and processes real time data from IoT and other data sources, and allows domain specific applications to employ both IoT and cloud resources on demand.

Thirdly, cloud computing can enable standardisation of smart city applications and turnkey solutions for software as a service (SaaS), providing on-demand self-services and decreasing significantly the associated development costs[13,20]. The provision of smart city services through flexible usage models and billing schemes, gives the opportunity to other cities with relatively limited budgets to make use of them. Standardisation of core city services, platforms and applications is extremely important as it accelerates technology diffusion and the uptake of proven smart city solutions, while at the same time enhances the emergence of collaborative innovation systems in these areas[20].

Much like open innovation, cloud-based smart city solutions require the collaboration among different actors of the urban ecosystem, meaning citizens, enterprises and the public sector. Besides, the existence of technologies being able to manage the cloud of things within cities does not automatically guarantee the development of smart city services[21]. According to Clohessy, Acton and Morgan[22], cloud computing smart city initiatives could harness the capabilities of open innovation paradigms such as living laboratories and crowdsourcing, taking full potential of the emerging collective intelligence. To this end, the use of open data is imperative for the development of innovative solutions and the opening up of new business opportunities.

Governments across the EU have initiated Government Cloud (G-Cloud) programs to deliver computing, storage and software capabilities to central and

local governments using cloud computing. Government G-Clouds are considered as promising models for smart cities, which can create urban clouds that reduce IT costs, offering platforms for business applications and e-services[23]. Contrary to the abundance of G-cloud initiatives, Clohessy, Acton and Morgan[22] proposed the development of a single G-Cloud, with the collaboration of government, citizens, businesses, and researchers, which will implement a number of cloud technologies on a hosted platform to create and deliver an integrated pool of smart city services and solutions. Examples of smart city cloud based platforms include SCOPE, implemented in Boston, USA as well as SOFIA2, implemented in Coruna, Spain. Finally, in line with the above, we also see the emergence of a number of joint initiatives aiming to battle the fragmentation of efforts towards smart cities and cloud computing:

- EU's Memorandum of Understanding (MoU) on Smart Cities Open and Interoperable Urban Platforms, which aims to integrate data flows within and across city systems with the use of modern technologies (cloud services, analytics, social media) enabling cities to shift from fragmented applications and to create confidence on the demand side. The initiative is part of the European Innovation Partnership on Smart Cities and Communities[24].
- Cloud28+[25], a European-based community aimed at increasing the visibility and revenue of its members and accelerating adoption of cloud technologies through the creation of a cloud service catalogue, with a strong focus on compliance with the European rules on data privacy and security
- Eurocloud[26], an independent non-profit organisation acting as a pan-European hub, working towards the maintenance of a constant open dialogue and the sharing of knowledge between cloud computing customers and providers, start-ups and research centers.

3. Moving Smart Cities Applications to the Cloud — the STORM CLOUDS Experience

The STORM CLOUDS project[27] aims to accelerate the pace at which public authorities move to the cloud computing. The project provides a methodology for the Cloud migration process, mainly from the point of view of the end-users, as well as the essential IT tools that will support this process. By taking the STORM CLOUDS approach, public authorities can take full advantage of the cloud-computing model and provide to the citizens in a highly reliable and innovative services, despite resource constraints.

The main assets of the project are as follows:

- **STORM CLOUDS Platform (SCP).** The platform provides the Cloud environment, which can host Smart City applications.
- **Portfolio of Smart City Services.** The portfolio includes many cloud-based, open source applications, which are ready for public authorities use.
- **Roadmap for the migration of public services into the Cloud.** The roadmap consists of guidelines that help public authorities to address the technical and business challenges in the adoption of cloud computing.
- **Best practices for cloud-based public services deployment.** The best practices include software techniques and methodological approaches, which facilitate the adoption of cloud services in the public sector.
- **Business models for the scalability and sustainability of the project's results.** The business models cover the exploitation of the SCP and Smart City services, as well as the viability of the already 'cloudified' (turned into cloud-computing) services in the project's pilots.

The STORM CLOUDS Platform (SCP) has been implemented using open source technologies and solutions. This approach not only lowers the cost of the solution but also helps public authorities to avoid vendor lock-in, as it can be easily transferred in different Cloud providers. Another valuable feature of the platform is that it supports all Cloud deployment models (i.e., private, community, public and hybrid), as well as two of the Cloud service categories (Platform as a Service and Infrastructure as a Service). Figure 1 presents the logical architecture of the SCP. Table 2 presents the components of each layer.

The STORM CLOUDS platform offers great flexibility to the cities that will use it as the Cloud environment for their Smart City applications. Cities are free to select an external Cloud provider or build their own private Cloud or have a hybrid solution. In the latter case, the private Cloud will host high-risk applications — those with high privacy and security requirements (i.e., applications that contain customer data and other sensitive information) — while the public Cloud will host the rest of them. Moreover, cites are able to deploy its applications to the IaaS or to the PaaS Layer.

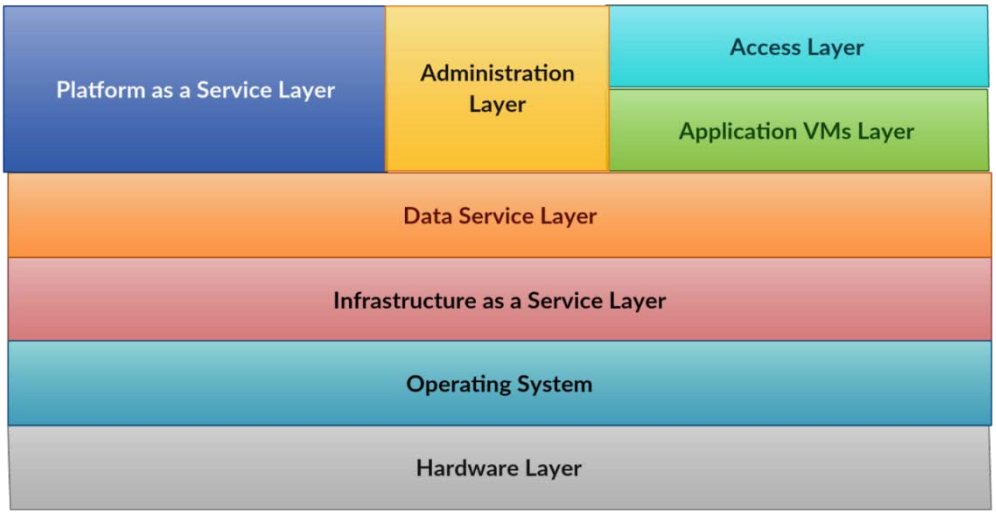

Figure 1. The logical architecture of STORM CLOUDS Platform.

Table 2. The components of STORM CLOUDS Platform

Component	Description
Hardware Layer	Physical resources (i.e., servers, storage, network, etc.) that host the platform.
Operating System	A Linux distribution that supports OpenStack (Debian 7.0, openSUSE, SUSE Linux Enterprise Server, Red Hat Enterprise Linux, CentOS, Fedora and Ubuntu).
IaaS Layer	It is implemented using OpenStack[28], the most popular and most adopted open source IaaS solution[29]. OpenStack can be seen as a cloud operating system that controls large pools of compute, storage, and networking resources (the hardware) and gives administrators a number of tools in order to deploy their applications.
Data Service Layer	Contains the database and the file servers, both deployed on VM clusters and provide high-availability and scalability. The currently supported databases are the MySQL and PostgreSQL.
Application VMs Layer	Contains the Virtual Machines that host the Smart City applications.
Access Layer	It is the front-end for the Smart City services. It receives the http(s) requests from users and redirects them to the suitable application's VM. It also implements the Load Balancer, which improves the overall performance of applications by distributing the workload in different VMs.
PaaS Layer	It is implemented using Cloud Foundry[30], a very popular open source platform for cloud applications.
Administration Layer	Provides to the platform's administrators and to the application owners, the tools that allow them to manage and monitor the components of the platform, as well as the Smart City applications. The tools include: • The Monitoring Module, which monitors the resources (CPU load, disk space occupation, network traffic, number of processes, etc.) used by the platform's services or by the applications. • The Database Administration Module, which administers the supported databases. • The Backup Module, which takes backups from databases and file-systems. • The Platform Administrator's Console, which allows the SCP administrator to have full control of the layers of the platform. Through the console, (s)he can manage the databases, the file-system, the IaaS layer, and the PaaS layers. Moreover, (s)he can deploy the Smart City applications to the IaaS layer automatically using a number of predefined scripts.

To decide which of the two options (IaaS or PaaS) they will follow, the public authorities should evaluate the pros and cons of each solution. On one hand, the IaaS offers excellent flexibility, as it does not require architectural changes to the applications, and full control of the resources used for the deployment. However, it increases the deployment complexity, as the applications' owners must take care of the installing and configuring of all components for its high availability and scalability. On the other hand, the PaaS "hides" the complexity of the underlying infrastructure and allows developers to deploy their web applications to the cloud without having to take care of the infrastructure. However, the applications may require significant changes[31] to comply with the PaaS principles and take full advantage of high availability and scalability features.

To address this problem, the SCP takes a hybrid approach and enhances the IaaS solution with two modules that provide the high-availability and scal-

ability features in a way that is transparent to the applications' owners. By accompanying the IaaS layer with the Data Service and Access layers, the data and the HTTP traffic management are delegated to the platform while the application's business logic is still contained on the VM(s). The hybrid solution offers great flexibility as it does not require architectural changes to the applications but also keeps the deployment complexity low because the application owner "leverages" the high-availability and scalability features of the platform. The only drawback of this solution comparing with the SaaS is that the application's owners are not entirely independent from platform's administrators, as the later should configure the high-availability and scalability features per application.

Apart from the platform, public authorities have the access to a portfolio of Smart-Cities applications[32]. These applications have been successfully cloudified and deployed in a STORM CLOUDS pilot city. Moreover, they have been validated by citizens and public authorities in different European cities and, at the same time, are general enough to be transferred and deployed in other cities worldwide. The project's partners offer consultation and development services to the cities wishing to deploy the applications in their Cloud environment. The portfolio also contains all the necessary documents and guides, so that public authorities can evaluate the differences of the services when considering a cloud-based service deployment. Table 3 summarises the available services.

The Smart City applications are related to (i) City Governance, (ii) Innovation Economy and (iii) Quality of Life. Table 4 presents the applications per category.

The STORM CLOUDS methodologies and IT tools have been validated in a number of European cities. They constitute a valuable asset for the cities that want to leverage the cloud computing and be more efficient, agile, and innovative. The SCP is flexible enough to support the different needs and requirements of the cities, as it can support a variety of cloud deployment models and services' categories. Public authorities can use the already cloudified applications from Smart City services portfolio as a starting point to their journey to the Cloud. Moreover, by using the platform's tools, they can easily migrate from their existing applications to the Cloud environment.

Table 3. STORM CLOUDS portfolio of Smart City services

Service	Description
Virtual City Market	The application — on one hand — provides the possibility to every commercial enterprise located in the city to create its own virtual shop, and — on the other hand— enables customers to access a variety of retailers using a shared site. In its simplest form the service provides a list of the existing shops located in the city (and their location on a map) as well as their offers. The virtual city market enhances collaboration schemes among retailers offering the opportunity to create open malls and organise the shops per street or district.
CloudFunding	The application supports local communities to collect money for social and charitable purposes. It gives citizens and organisations the ability to co-finance projects for: (i) the improvement of the urban environment, (ii) social entrepreneurship and (iii) knowledge-intensive and technology-based youth entrepreneurship.
City Branding	The application promotes the identity of a city using interactive maps, 360º panoramas, video and images. It enables Municipalities to focus on different target groups, which are associated with various aspects of city's identity by supporting the differentiation of the commons according to the target group in which the visitor belongs. Moreover, the application allows services provisioning based on the public space of the city and in particular around public commons, which are connected to local shops and services providers. In this way, local business communities can offer services tailored to the tourists' needs and expectations.
Improve My City	The application enables the citizens to report non-emergency local problems such as potholes, illegal trash dumping, faulty streetlights, broken tiles on sidewalks, and illegal advertising boards. The submitted issues are displayed on the city's map. Users may add photos and comments. Moreover, they can suggest solutions for improving the environment of their neighbourhood. Through this service, the Municipalities enable citizens and local actors to take action to improve their neighbourhood.
Vive	The application enables citizens to share information about leisure events in the city, map located and classified according to its cost (free, or low costs) and type (festival, movie, concert, etc.). The user could also see events close to her/his location.
Colabora	The application can be used to report issues in the street, but it has a very flexible data model that can be easily modified to represent many collaborative issues. A number of variations are available for: (i) urban issues, (ii) free or low-cost events and (iii) crime-spotting.
Have you Say	The application supports bottom-up decisions, by enabling Municipalities to ask the citizens' opinion about the future city plan, theme or issue. Both public authorities and citizens can upload photos and documents to support their opinion. Real-time statistics are also available for promotion of transparency in the participation process.

Table 4. Smart City services per category

City Governance	Innovation Economy	Quality of Life
• Have Your Say • Colabora • Improve My City	• Virtual City Market • CloudFunding • City Branding	• Live the City

4. Research Trends and Future Challenges

Based on the advances made so far, this section aims to highlight new scientific directions and future challenges with regards to smart cities and cloud computing. Although the trends that define the future of cloud computing can be numerous, ranging from technological aspects to new business models/opportunities, we identified four areas that are about to play a significant role with regards to cloud computing adoption from municipalities and city governments.

As Petrolo, Loscri and Mitton[33] mentioned, "in Smart City context, Cloud of Things (CoT) is expected to play a significant role in making a better use of distributed resources, achieving higher throughput and tackling large scale of computational problems, to enable the horizontal integration of various vertical IoT platforms and the Smart City vision". This means that over the next years, the focus will be on cloud platforms dedicated for IoT and on technologies for real time processing of big data and linked data. Advanced analytics over millions of data streams coming from highly distributed, heterogeneous, decentralised, real and virtual devices and data sources[9,15] which bring a new vision on the notion of cloud scalability. Here, issues of interoperability, privacy and security should be carefully considered.

The SCP could be expanded in order to enable the rapid deployment of city wide networked sensors and actuators, as well as of IoT applications. Three new modules should be developed to support the core requirement of the IoT solutions:

- The Control Module that will handle the device interfacing and will enable the time-critical response. The module will consist of autonomous control applications and drivers which support a broad range of existing and new equipment and protocols.
- The Data Provisioning Module that will provide a unified sensor data acquisition in order to enable further processing steps. The module will store the acquired data in the SCP Data Service Layer. Data transformation techniques, such as data masking and data obfuscation, will be implemented in order to address privacy and security issues.
- The Big Data Analytics Module that will provide historic as well as real-time analytics. The historic analytics will identify data patterns of significance and will enable the optimisation of algorithms, services, and solution delivery while the real-time analytics will evaluate data as they come into the system in order to produce insights in near-real-time for immediate exploitation.

Apart from the development of the new modules, the platform's Data Service Layer should be enhanced with big data handling capabilities.

Although SaaS applications can offer higher flexibility and lower cost, integration between SaaS and on-premises legacy applications has been identified as a significant obstacle to adopt and deploy SaaS and other web-based applications. However, cloud integration does not only refer to cloud and on-premises integration, but also refers to integration among different clouds. Services convergence and multi cloud integration[34] is a promising paradigm, which creates new system design possibilities but also presents technological and management challenges, such as portability, compliance, elasticity and high availability[35]. These issues will be the focus of interest over the following years.

Despite the significant benefits of cloud computing in public administrations (cost savings related to scalability, increased efficiency, accelerated innovation etc.), there is still a lot of effort associated to the development and running of composite applications on the cloud. Based on the principles which are similar to the component-based software development (CBD) and service oriented architecture (SOA), cloudification of core application components and component portability can create repositories of the essential building blocks of smart city applications, reducing the cost of development and enhancing the emergence of new business models. The development of online libraries with re-usable software components providing different services for processing requests (e.g., database connection for the performance of database queries, user authentication, mail composing and sending, etc.) has had a significant background[36]. However, delivery of a service over different cloud providers requires specific considerations including issues of interoperability, reusability, open standards, etc.

Cloud computing encourages automation because the infrastructure is programmable. To ensure a high

level of automation along with accuracy in the migration of the applications to the cloud, a set of tools and procedures have been developed and integrated into the SCP. The automatic deployment is implemented using OpenStack's orchestration engine to launch multiple composite cloud applications based on templates. The aim of orchestration is to create a human-and machine-accessible service for managing the entire lifecycle of infrastructure and applications within the SCP Cloud environment. Using these tools, the city of Miskolcin, Hungary, managed to move its applications to the Cloud within a period of two months.

Finally, we see the growth of cloud-based platforms and/or marketplaces being able to offer standardised core smart city services with on-demand infrastructure aiming to make smart city application deployment a simple process to follow. Marketplaces have emerged over the last few years as a way to provide a unified channel of distributing high quality cloud services, bringing together cloud service providers, data centre operators and technology partners[37]. The idea of having third party cloud-based applications integrated into one application suite has started gaining momentum as most municipalities do not have the resources, such as funds, expertise and available technology, which will enable them to move towards the vision of becoming a smart city. Cloud marketplaces and cloud-based application suites can help in speeding up the implementation process of smart city turn-key solutions in different domains of cities including transportation, sanitation, urban planning, policy making and so on. Research on this topic will have to focus also on issues such as Service Licence Agreements (SLAs), open standards etc.

Apart from the automated migration process, the SCP accelerates the uptake of cloud computing by public authorities in service provisioning, by offering a marketplace of commonly used smart city applications. Cities can provide cloud-based smart city services by activating their applications of choice and customise them depending on the specific characteristics of their cities. Using this approach, the cities, which participate in the STORM CLOUDS project, enriched their smart city portfolio by incorporating the applications from a catalogue of applications that has been cloudified in other pilot cities. The STORM CLOUDS marketplace will increase the number of generic smart city applications over the time, as the new cities that will migrate their applications to the latter platform will be offered the opportunity to add them to the catalogue while earning revenue each time another city uses them.

Conflict of Interest and Funding

No conflict of interest has been reported by the authors.

Acknowledgements

This paper is based on the experience gained from STORM CLOUDS, a project co-funded by the CIP-ICT-PSP program of the European Commission.

References

1. Marston S, Li Z, Bandyopadhyay S, *et al*. 2011, Cloud computing — the business perspective. *Decision Support Systems*, vol.51(1): 176–189.
2. Buyya R, Yeo C S, Venugopal S, *et al*. 2009, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, *Future Generation Computer Systems*, vol.25(6): 599–616.
3. U.S. Department of Commerce, National Institute of Standards and Technology, 2011, *The NIST definition of cloud computing*, viewed January 24, 2016, <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
4. Gartner, *IT Glossary: Cloud Computing*, viewed January 25, 2016, <http://www.gartner.com/it-glossary/cloud-computing>
5. Jinesh V, 2011, *Architecting for the Cloud: best practices*, Amazon Web Services, viewed January 24, 2016, <https://media.amazonwebservices.com/AWS_Cloud_Best_Practices.pdf>
6. *Best practices for federal agency adoption of commercial cloud solutions*, 2015, Professional Services Council, viewed January 25, 2016, <https://www.pscouncil.org/Downloads/documents/PSC-Cloud-WEB%20-%2012-10-15.pdf>
7. U.S. Chief Information Officer, 2011, *Federal cloud computing strategy*, White House.
8. Cloud Standards Customer Council, 2013, *Migrating applications to public cloud services: roadmap for success*, viewed January 25, 2016, <http://www.cloud-council.org/Migrating-Apps-to-the-Cloud-Final.pdf>
9. KhanZ, Anjum A, Soomro K, *et al*. 2015, Towards cloud based big data analytics for smart future cities. *Journal of Cloud Computing: Advances, Systems and Applications*, vol.4(2): 1–11. http://dx.doi.org/10.1186/s13677-015-0026-8.
10. Komninos N, 2013, Smart cities and the future internet: innovation ecosystems of embedded spatial intelligence,

Proceedings of International Conference for Entrepreneurship, Innovation and Regional Development, ICEIRD, 2013.

11. Mitchel S, Villa N, Stewart-Weeks M, *et al*. 2013, *The Internet of everything for cities: connecting people, process, data, and things to improve the 'livability' of cities and communities, point of view*, Cisco, viewed January 10, 2016, <http://www.cisco.com/c/dam/en_us/solutions/industries/docs/gov/everything-for-cities.pdf>
12. Fu Y, Jia S and Hao J, 2015, A scalable cloud for the Internet of Things in smart cities. *Journal of Computers*, vol.26(3): 63–75.
13. Schaffers H, Komninos N, Pallot M, *et al*. 2011, Smart cities and the future internet: towards cooperation frameworks for open innovation, in J Domingue, *et al*. (eds.) *The Future Internet — Future Internet Assembly 2011: Achievements and Technological Promises*, 431–446.
14. Mitton N, Papavassiliou S, Puliafito A, *et al*. 2012, Combining cloud and sensors in a smart city environment, *EURASIP Journal on Wireless Communications and Networking 2012*, 247.
15. Suciu G, Vulpe A, Halunga S, *et al*. 2013, Smart cities built on resilient cloud computing and secure Internet of Things, in *Control Systems and Computer Science (CSCS), 19th International Conference, 29–31 May 2013*, 513–518.
16. Aditi Technologies, 2015, *Building "smart" cities on the Cloud*, viewed January 10, 2016, <https://blog.aditi.com/cloud/building-smart-cities-cloud/>
17. Microsoft, 2011, *The central role of cloud computing in making cities energy-smart*, Microsoft Corporation, viewed January 10, 2016, <https://www.microsoft.com/en-us/download/confirmation.aspx?id=5960>
18. Jarvis G, 2014, *Why virtualisation isn't enough in cloud computing*, CloudTech, viewed January 15, 2016, <http://www.cloudcomputing-news.net/news/2014/feb/12/why-virtualisation-isnt-enough-cloud-computing/>
19. Dustar S, Vögler M, Sehic S, *et al*. 2014, *The Internet of Things meets cloud computing in smart cities*, Bridges vol.41, OpEds & Commentaries, viewed January 15, 2016, <http://ostaustria.org/bridges-magazine/item/8280-the-internet-of-things-meets-cloud-computing-in-smart-cities>
20. Komninos N, 2014, *The Age of Intelligent Cities*, Routledge, London and New York.
21. Clohessy T and Acton T, 2013, Open innovation as a route to value in cloud computing, *26th Bled eConference, eInnovations: Challenges and Impacts for Individuals, Organizations and Society, 9–13 June, 2013, Bled, Slovenia*.
22. Clohessy T, Acton T and Morgan L, 2014, Smart City as a Service (SCaaS): a future roadmap for e-government smart city cloud computing initiatives, *UCC '14 Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, 836–841*.
23. Schaffers H, Komninos N and Pallot M, 2012, *FIREBALL White Paper: smart cities as innovation ecosystems sustained by the future internet*, viewed January 10, 2016, <http://www.urenio.org/wp-content/uploads/2012/04/2012-FIREBALL-White-Paper-Final.pdf>
24. *Market Place of the European Innovation Partnership on Smart Cities and Communities*, n.d., viewed January 14, 2016, <https://eu-smartcities.eu>
25. *Cloud28+: Europe's Cloud of Clouds*, n.d., viewed January 15, 2016, <http://www.cloud28plus.eu>
26. *EuroCloud*, n.d., viewed January 21, 2016, <https://www.eurocloud.org>
27. *STORM CLOUDS Project — main page*, viewed January 27, 2016, <http://stormclouds.eu>
28. *OpenStack — main page*, viewed January 27, 2016, <http://www.openstack.org>
29. *The top open source cloud projects of 2014*, viewed January 29, 2016, <https://www.linux.com/news/enterprise/cloud-computing/784573-the-top-open-source-cloud-projects-of-2014>
30. *Cloud Foundry — main page*, viewed January 29, 2016, <https://www.cloudfoundry.org>
31. *The twelve-factor app — a methodology for building software-as-a-service apps*, viewed January 29, 2016, <http://12factor.net>
32. *Storm Clouds Services Portfolio — main page*, viewed January 29, 2016, <http://stormclouds.urenio.org>
33. Petrolo R, Loscri V and Mitton N, 2014, Towards a smart city based on Cloud of Things, *WiMobCity '14, August 11, 2014, Philadelphia, PA, USA*.
34. Li Q, Wang Z, Li W, *et al*. 2013, Model-based services convergence and multi-clouds integration. *Computers in Industry*, vol.64(7): 813–832. http://dx.doi.org/10.1016/j.compind.2013.05.003.
35. Paraiso F, Merle P and Seinturier L, 2014, soCloud: a service-oriented component-based PaaS for managing portability, provisioning, elasticity and high availability across multiple clouds. *Computing*, vol.98(5): 539–565. http://dx.doi.org/10.1007/s00607-014-0421-x.
36. Heineman G T and Councill W T, 2001, *Component-based Software Engineering: Putting the Pieces Together*, Addison-Westley Publishing, Boston, USA.
37. Fugate R and Tang G, 2015, *Cloud market transformation, industry prediction update*, Cartesian, viewed January 29, 2016, <http://www.cartesian.com/cloud-market-transformation/>

RESEARCH ARTICLE

Storm Clouds Platform: a cloud computing platform for smart city applications

Marco Battarra, Marco Consonni*, Samuele De Domenico and Andrea Milani

Hewlett Packard Italiana S.R.L., 9, Via Di Vittorio Giuseppe – 20063 Cernusco Sul Naviglio, Italy

Abstract: This paper describes our work on STORM CLOUDS[1], a project with the main objective of migrating smart-city services, that Public Authorities (PAs) currently provided using traditional Information Technology, to a cloud-based environment. Our organization was in charge of finding the technical solutions, so we designed and implemented a cloud computing solution called **Storm Clouds Platform** (SCP), for that purpose. In principle, the applications we ported could run on a public-cloud service, like Amazon Web Services™[2] or Microsoft® Azure[3], that provide computational resources on a pay-per-use paradigm. However, these solutions have disadvantages due to their proprietary nature: vendor lock-in is one of the issues but other serious problems are related to the lack of full control on how data and applications are processed in the cloud. As an example, when using a public cloud, the users of the cloud services have very little control on the location where applications run and data are stored, if there is any. This is identified as one of the most important obstacles in cloud computing adoption, particularly in applications manage personal data and the application provider has legal obligation of preserving end user privacy[4]. This paper explains how we faced the problem and the solutions we found. We designed a cloud computing platform — completely based on open-software components — that can be used for either implementing private clouds or for porting applications to public clouds.

Keywords: smart city, cloud computing, infrastructure as service, OpenStack

*Correspondence to: Marco Consonni, Hewlett Packard Italiana S.R.L., 9, Via Di Vittorio Giuseppe – 20063 Cernusco Sul Naviglio, Italy; Email: marco.consonni@hpe.com

Received: February 3, 2016; **Accepted:** April 12, 2016; **Published Online:** May 9, 2016

Citation: Battarra M, Consonni M, De Domenico S, *et al*. 2016, Storm Clouds Platform: a cloud computing platform for smart city applications. *Journal of Smart Cities*, vol.2(1): 14–25. http://dx.doi.org/10.18063/JSC.2016.01.003.

1. Introduction

Cloud computing is a delivery model for technology enabled services that drives greater agility, speed and cost savings. It provides on-demand access, via a network, to an elastic pool of shared computing resources (e.g., services, applications, frameworks, platforms, servers, storage, and networks) that can be rapidly provisioned and released with minimal service provider interaction and scaled as needed according to a pay-per-use paradigm.

We recently participated in STORM CLOUDS, a project partially funded by the European Commission within the 7th Framework Program (Grant Agreement No. 621089) with the main objective of deploying smart-cities application services to a cloud-based environment. In the project, we were in charge of designing and implementing the technical solutions taking into account some general but fundamental requirements:

- the tools used to port and/or deploy the applications should be based on open source technology in order to avoid vendor lock-in issues,
- the solution was required to support different deployment models ranging from situations in which the applications run on data-centers owned

by PAs, to scenarios in which they run on a public cloud, made available by some cloud service provider,
- the solution should support the implementation of a catalogue of easily deployable applications: the idea was that any ported application could be reused by PAs not directly participating in the project.

This article briefly describes the experience gained while working on the project, illustrating the problem and the solution found. Storm Clouds Platform (SCP) was the platform we implemented and this article briefly describes the functions it implements and the architecture.

2. The Problem

In order to find a solution for the generic problem of "porting applications to cloud", first we tried to understand the problem better, focusing on two aspects: the project context and the applications to port.

2.1 Project Context

We started by identifying the "entities" (e.g., organizations, users, other systems, etc.) participating in the project and/or interacting with applications to port. Figure 1 summarizes what we found out.

As reported[1], we needed a 'digital space' for running application services and, as required by the cloud computing paradigm, the resources for running them should be activated on-demand. We called this digital space Storm Clouds Platform (SCP).

Analyzing the applications, we found that they could inter-operate with each-other and/or with External Digital Services in order to achieve the required functionality. For instance, several applications used Google Maps[5], a service available on Internet through a web service interface, for rendering maps on HTML pages. Citizens and/or public servants were identified as the End Users of the applications while Public Administrations (e.g., municipalities) were the Application Providers, responsible for providing services to citizens. Application Creators were the organizations in charge of implementing the applications. We found that usually Independent Software Vendors (ISVs) cover this role but, in some cases, the Application Provider is also the Application Creator (e.g., a municipality might also be the creator of the application). Generally, it was important to distinguish these two roles because, while application creators are usually interested in the technical details for porting applications to a cloud environment, application providers give more importance to aspects like the cost of the solution, the economic benefits, whether data are processed according to regulations (e.g., if and how security/privacy problems are addressed), etc.

In the project, we played the role of the Platform Provider, in charge of designing, implementing and

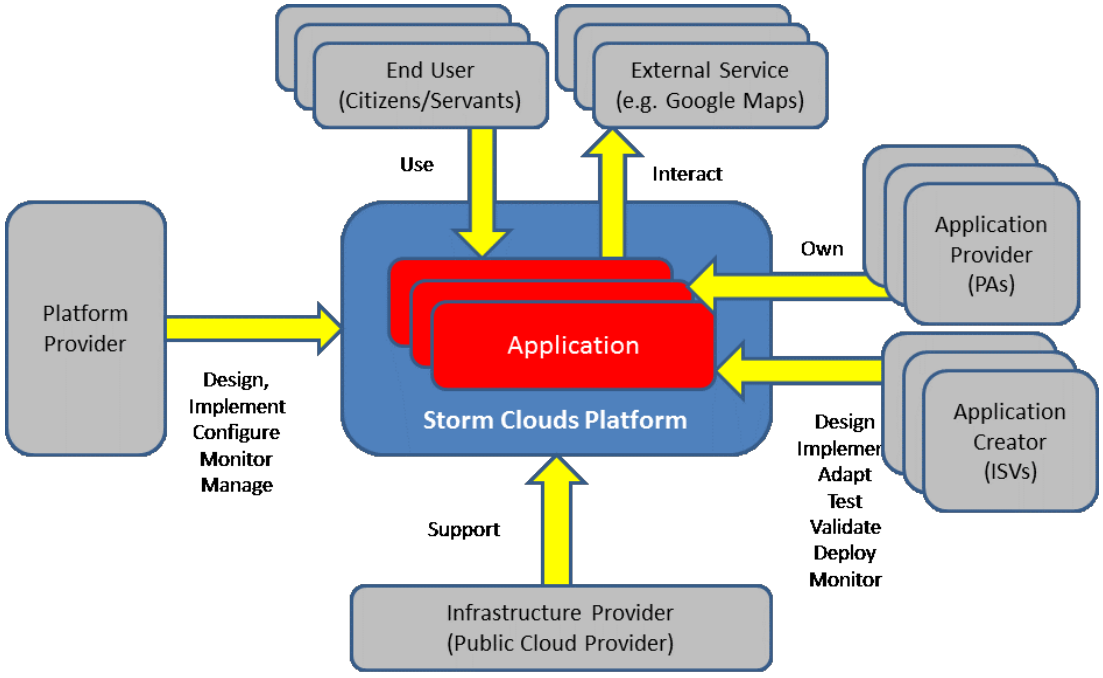

Figure 1. STORM CLOUDS project context.

operating the SCP. We decided to engage an Infrastructure Provider for supplying the physical resources (e.g., physical servers, disk storage and network connectivity) for hosting the platform and the applications. The distinction between these two roles, in addition to being useful for the project because it was not necessary to make an up-front investment for the physical resources, was also aligned with the general requirement of providing a solution that could be deployed on-premises (i.e., on the municipality's data centre), off-premises (i.e., a data centre of a third party) or even using services of a public cloud operator. In fact, after the project termination, PAs could decide to implement the solution at their own data centres, use a data centre of a third party or even deploy the applications on a public cloud.

2.2 Application Services

The applications were selected from a set of completely implemented smart-city services that the partners developed in previous projects. We conducted an assessment for gathering information that served as requirements for the platform. We asked the application proponents to give technical details on their applications in order to characterize the kind of workload we should support. Table 1 summarizes the results of the assessment.

Most of the applications were implemented using open source products at different levels (e.g., Linux operating system; MySQL/MariaDB or PostgreSQL databases; PHP, Python, Java and JavaScript programming languages, etc.), whilst few were based

Table 1. Application assessment results

Application	Operating System					Language								Database					
	Ubuntu	CentOS	Linux (*)	OpenWrt	Windows	Javascript	PHP	Java	Perl	C#	Python	PL/SQL	Ruby	PostgreSQL	MySQL	SQLite	MSSQL	Mongo	Oracle
Blue Parking Valladolid		1			1	1		1			1	1						1	1
City Navigator											1								
Co-Labora			1		1		1								1				
Crowd Dunding													1	1					
Emissão Plantas de Localização	1					1	1	1						1					
Honolulu Answers	1												1	1					
HotSpot CMA	1			1		1	1								1				
Ideol – Innobarómetro	1					1	1	1			1			1	1				
Improve My City	1	1	1			1	1								1				
LimeSurvey		1	1			1	1		1					1			1		
LocalGIS	1	1			1	1		1						1					
LocalWiki	1	1									1			1					
OpenCivic	1	1	1			1	1								1				
OpenTripPlanner								1											
Participação Pública (PPGIS)	1	1	1			1								1					
Plano Director Municipal	1					1	1	1						1					
SEDOC	1						1	1		1					1	1			
Sense the City	1	1	1			1	1								1				
SIMPLEXT	1							1											
Urbanismo en Red - UeR	1							1						1					
Virtual City Marketplace	1	1	1			1	1								1				
Virtual City Tour	1	1	1			1	1								1				
Vive Valladolid	1	1	1			1	1								1				
We The People Petitions	1					1	1								1			1	
Total	**18**	**11**	**9**	**1**	**3**	**15**	**14**	**9**	**1**	**1**	**4**	**1**	**2**	**10**	**11**	**1**	**1**	**2**	**1**

Note: (*) Linux stands for any Linux distribution.

on proprietary technologies such as Oracle DBMS, Windows Server 2008, etc. The steering committee of the project decided to port only applications based on open source software packages, in order to be sure that the act of "porting the application to cloud" did not constitute an infringement of the licensing rights when the application was activated.

During the assessment we also found out that some applications managed sensitive information like, for example, citizens' personal data. This was a very important aspect with implications both on the applications to migrate during the project and the solution we designed. In fact, the Application Providers had the legal responsibility of preserving citizens' privacy[4] and — although during the project some activities were addressed to verify the application security — they preferred to migrate services that did not manage sensitive information. Application Providers were not in the position of fully controlling the way data was managed because, as described above, all the applications were to run on servers made available by the Infrastructure Provider, an organization external to the project. However, the Application Owners required that the solution could be hosted on any data centre so that, after the project termination, there would have the possibility of creating a Storm Clouds Platform instance at their own premise. As a result, we had to fulfill the requirement of defining a solution that could be implemented both on-premises and off-premises.

2.3 High-level Requirements

The analysis of the project context and the results of the application assessment can be briefly summarized in a list of high-level requirements.

- **Open Source Technology**: the solution should be implemented using open source software packages for avoiding vendor lock-in issues and for controlling the cost of the porting. This was also in accordance with the indications of the European Commission (EC) that "*supports Free/Open Source Software (FOSS) as a development model since it is a very effective way to collaboratively develop software with fast take-up and improvement cycles*" [6].
- **LAMP Workload**: the solution should support and facilitate the porting of LAMP applications. According to a broadly accepted definition[7], "*The acronym LAMP refers to first letters of the four components of a solution stack, composed entirely of free and open-source software, suitable for building high-availability heavy-duty dynamic web sites*". The meaning of the LAMP acronym depends on which specific components are used as parts of the actual bundle. In the case of the project, "L" stands for Linux; "A" for Apache HTTP Web Server; "M" identifies MySQL, MariaDB or MongoDB, the database management system; "P" is for PHP or Python, the programming languages used for dynamic web pages and web development[1].
- **Deployment Models**: the solution should support various cloud computing deployment models like *Private Cloud* in which "*the cloud infrastructure is provisioned for exclusive use by a single organization*"[8] (this is the case in which a municipality has its own data centre, *Community Cloud* in which "*the cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns*"[8] (when, for example, all the municipalities use a single cloud provided at the National level) and *Public Cloud* in which "*the cloud infrastructure is provisioned for open use by the general public*"[8].
- **Management**: the solution should implement tools for managing the applications and the components of the solution itself. In this area, the requirement was directed to tools for administering, monitoring and automating the deployment of the services.

3. The Solution

Figure 2 shows the logical architecture of the Storm Clouds Platform (SCP), i.e., the solution we designed for the project.

SCP is a layered architecture in which the Infrastructure as a Service Layer (IaaS Layer) works as the foundation of the whole solution. IaaS Layer provides basic IT capabilities as compute services (e.g., Virtual Machines), storage services (e.g., Virtual Volumes) and networking services (e.g., Virtual Networks) for the implementation of the upper layers.

Definitely the IaaS layer should be run on some physical resources, such as servers, disks and network equipment that are represented by the Hardware Layer. This layer is implemented by the hardware of the data

[1] As described in the following pages, the architecture we designed also supports PostgreSQL database management system and other programming languages like Java, Perl and Ruby.

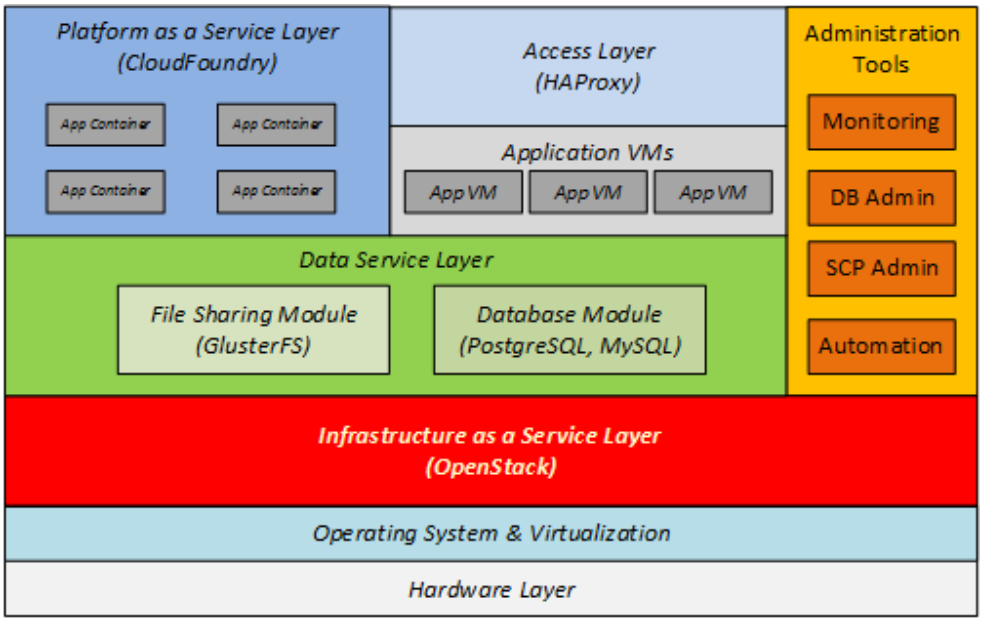

Figure 2. Storm Clouds Platform architecture.

centre(s) where the platform is hosted and was considered out of scope. The only requirement for the Hardware Layer is that the servers are equipped with an Operating System supporting OpenStack [2], the software solution we selected for implementing the IaaS Layer.

The platform supports the Applications in two alternative ways: as Application VMs activated at the IaaS Layer for running the application logic or as Application Containers hosted by the platform as a Service Layer (PaaS Layer). There are pros and cons for these two alternative approaches, as described in following sections of this article; we decided to implement both in order to have a more flexible solution.

While analysing the Applications we observed common functions that we 'factored-out' and support at the platform level. The Data Service Layer implements database and file sharing functionality, while the Access Layer manages the requests coming from the end users of the applications.

Finally, SCP provides Administration Tools for monitoring and managing the resources (i.e., services and applications) activated in the platform.

3.1 Infrastructure as a Service Layer

The Infrastructure as a Service Layer provides services for creating virtual resources (like virtual machines, virtual disks and virtual networks) used instead of their physical counterparts, for deploying and running application software. Resources are provided as services, meaning that they are 'created' when needed, used to run applications, and 'removed' when the application is not needed anymore. The actual computation happen at the physical level but physical resources and applications are not tightly bound together. This makes it easier to reuse the physical infrastructure for several purposes, usually at different times. This is obtained with the extensive use of virtualization technology[9] that is part of the IaaS Layer.

For the implementation of the IaaS Layer, we selected OpenStack[10], an open source technology (all source code is freely available under the Apache 2.0 license). The main reason for selecting this technology was because it was the most popular and most adopted open source IaaS solution[11].

Figure 3 shows the OpenStack high-level logical architecture. OpenStack is composed of the following modules mapping the fundamental IaaS services:

- Nova provides computation services (Virtual Machines);
- Neutron provides networking services (Virtual Networks);
- Cinder provides block storage services (Virtual Disks);
- Swift implements object storage services (Files);
- Horizon provides a web front-end for managing and controlling the resources allocated in the cloud;

[2] Possible distributions are Debian 7.0, openSUSE, SUSE Linux Enterprise Server, Red Hat Enterprise Linux, CentOS, Fedora and Ubuntu 12.04/14.04 (LTS).

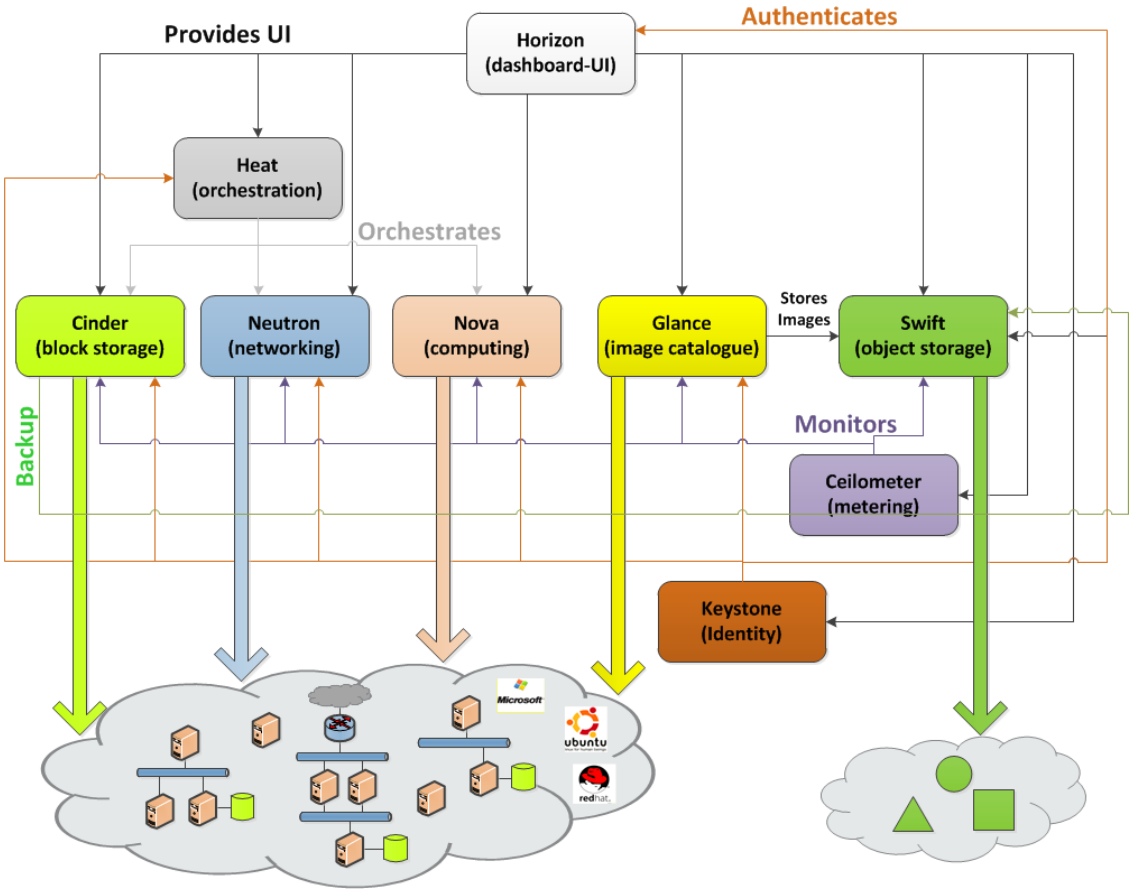

Figure 3. OpenStack logical architecture.

- Glance implements a catalogue for storing virtual machine images;
- Keystone implements authentication and authorization functions;
- Heat uses the other components for orchestrating the creation/deletion of virtual resource groups described by script files called “stacks”;
- Ceilometer monitors the usage of resources for metering and accounting purposes.

In principle, the IaaS Layer could be sufficient for deploying any application we assessed. We could activate a virtual machine, install all the required software packages and run the application. This situation is described by Figure 4.

This scenario, albeit supported by the platform, does not address some important issues of a production-ready situation. What happens if a VM is switched off accidentally or voluntary (for example for maintenance reasons)? What if a single machine is not enough for handling the requests of an increased number of users? In other words, how do we address high-availability and scalability issues?

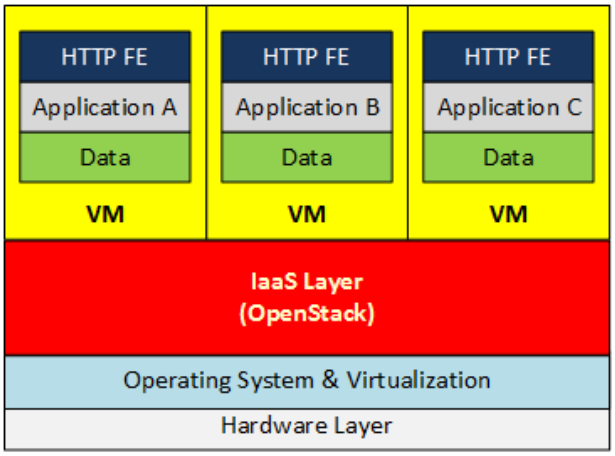

Figure 4. Deploying applications on IaaS.

In addition, from a functional perspective, the essence of an application is represented by the business logic it implements (the part in grey), while functions like HTTP traffic management and data management (e.g., database management) are common to all applications. We decided to implement them at the platform level and make them available as services, in order to

facilitate the application deployment. Following these ideas, we thought of 'decoupling' the three levels of the application stack (namely, the HTTP Front End (HTTP FE), the Application logic and the data management) and provide the HTTP FE functions and the data management functions at SCP level. In addition, we considered that several applications could benefit from Platform-as-a-Service solutions specifically designed for LAMP stacks. As a result, we envisioned the solution shown in Figure 5.

All the layers were implemented using VMs activated at the IaaS Layer level and implement high-availability and scalability, enabling the implementation of similar features for the application services[3].

While working directly at the IaaS layer provides great flexibility and full control, the PaaS layer provides a convenient alternative that facilitates the deployment of applications by 'hiding' the complexity of the underlying infrastructure. In fact, the application developer does not explicitly activate VMs for running her applications; she just uploads software packages to the PaaS Layer that takes care of activating the computing resources on her behalf. Furthermore, the PaaS can handle scalability and high-availability automatically when applications are designed properly.

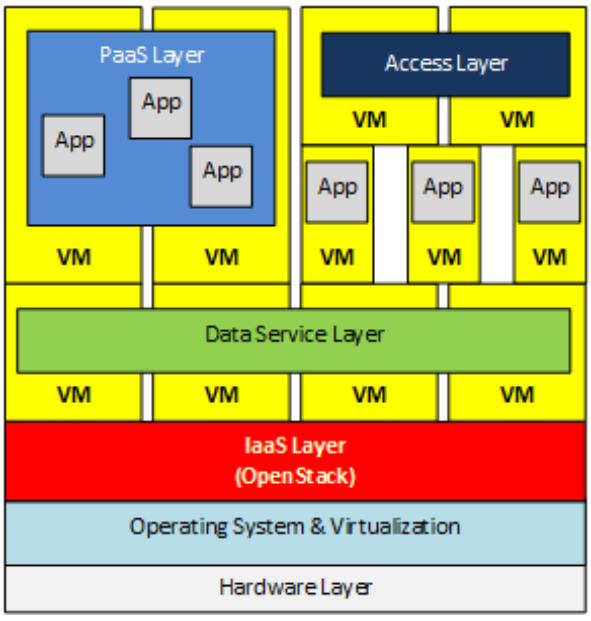

Figure 5. Decoupling the application stack.

[3] High availability and scalability of an application can be obtained only when all the layers support such features, application layer included. For this reason, we cannot claim that their implementation at the Access and the Data Service Layer is sufficient; it also depends on how the application is designed.

In this perspective, SCP enables different application migration options summarized in the Table 2.

The following sections describe the layers built on top of the IaaS Layer in greater details.

3.2 Data Service Layer

The applications that analyzed store structured data in a database and use a file system for unstructured data like images, music and videos. As mentioned above, one of the objectives of the Data Service Layer is to provide mechanisms for implementing H/A; therefore two or more VMs — hosting the business logic of an application — should be able to share data so that, in case one VM becomes unavailable, the other(s) can continue the service.

For these reasons Data Service Layer implements both database and file sharing functions with two subcomponents: the Database Services Module, implementing database functions, and the File Sharing Services Module, providing file server functionality. They are both deployed on VM clusters and implement high-availability and scalability.

At the time of writing, the Database Service Module supports MySQL/MariaDB and PostgreSQL database engines; it is deployed as an active/stand-by VM cluster but the architecture supports also other deployment topologies[4], such as active/active, N+M, N-to-N, etc.[12]

The File Sharing Service Module is implemented as a cluster of VMs hosting Gluster[13], a scale-out network-attached storage file system. Gluster implements a client/server architecture in which the servers are aggregated into one large parallel network file system; while clients, equipped with the Gluster client software package, mount shared volumes that are seen as local file systems.

The Data Service Layer provides each application with a private database and a private volume but multiple VMs of an application can share the same data allowing the implementation of H/A and scalability.

3.3 Access Layer

The Access Layer implements the HTTP Front End for web based services. It receives HTTP requests directed to applications and, having knowledge of what VMs host the required service, dispatches the request accordingly, trying to balance the traffic among all the VMs. In addition, it continuously monitors the availability

[4] The topology of a cluster defines how many nodes are used and/or how the work is distributed among them. For more information see[12].

Table 2. Migration options

Option	Description	Pros	Cons
Full IaaS	All the application components are deployed on VM(s) explicitly managed by the application owner	- No architectural change of the application - Full control on the resources used for the deployment	- Great deployment complexity because the application owner must take care of installing and configuring all the components for H/A and scalability
IaaS + Data Service Layer + Access Layer (optional)	Data management and (optionally) HTTP traffic management are delegated to the platform while the application business logic is still deployed on VM(s)	- No architectural change of the application - Less deployment complexity because the application owner 'leverages' the high-available and scalable features of the platform layers	- Because of the centralized administration of the shared functions (e.g. data service layer), application owners cannot deploy their applications in full autonomy
PaaS + Data Service Layer	Applications are hosted by the PaaS Layer and use the Data Service Layer for storing data	- No infrastructure management required by the user: the platform does it for her	- Applications can require significant changes to comply with PaaS principles

of the VMs and, in case one of them becomes unavailable, redistributes the traffic among the remaining ones.

The Access Layer implements the Load Balancer Module based on the open source HAProxy software[14] and is implemented as a set of VMs deployed 'in front' of the VMs hosting the applications, as described in Figure 6.

The HAProxy VMs receive the HTTP request and dispatch it to the Application VMs distributing the workload according to a load balancing algorithm. In addition, HAProxy VMs periodically monitor the Application VMs and, in case a VM appears unavailable, redistribute the upcoming requests to the remaining ones. As shown in Figure 6, HAProxy can be deployed on several VMs (in the picture we have a two VM cluster) implementing H/A.

3.4 Platform as a Service Layer

The Platform as a Service Layer (PaaS Layer) is a sophisticated solution that allows developers to deploy their web applications to the cloud, without having to take care of the underlying infrastructure. In fact, while IaaS Layer focuses on managing the fundamental

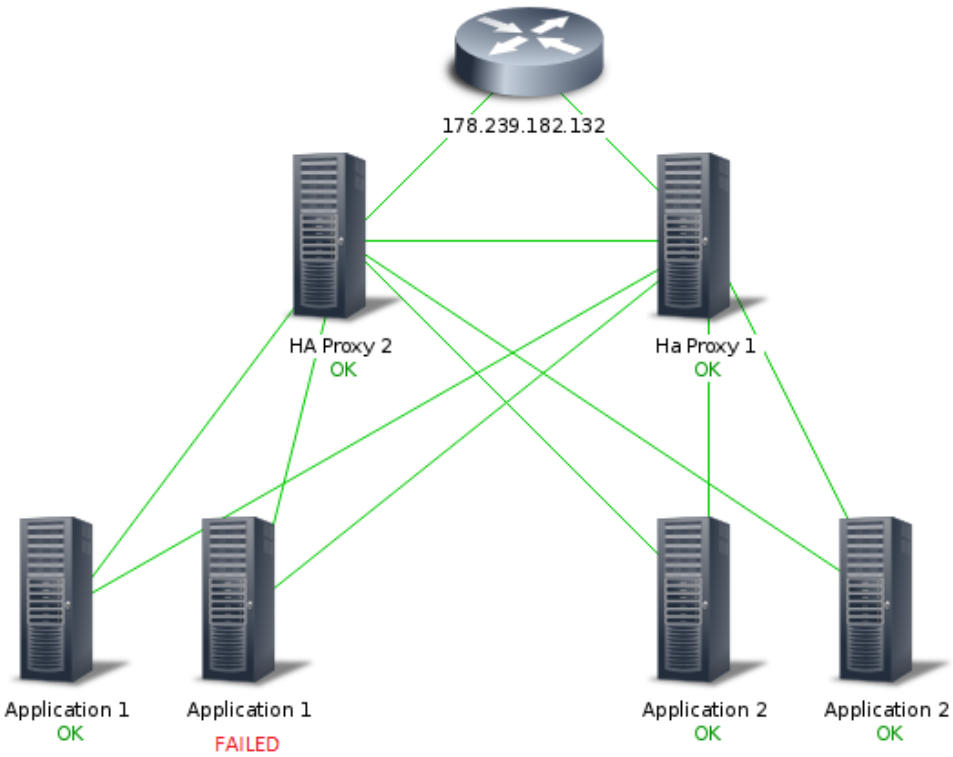

Figure 6. Load balancer module (HAProxy).

infrastructure building blocks in a cloud environment, thus allowing the transfer any existing deployment to the cloud with little or no architectural changes, PaaS Layer goes one step further and focuses on managing applications instead of infrastructure. The developer who can deploy an application to the PaaS and expects it to perform, delegates all infrastructure management tasks to the PaaS and focuses on development work instead. As a consequence, the primary resources involved in deploying an application to a PaaS are not virtual machines, virtual storage and virtual network objects, but application services, configuration and artifacts, as shown in Figure 7.

The picture shows that the PaaS Layer uses resources (mainly VMs) provided by the IaaS Layer 'hiding' the correspondent complexity. Programmers can deploy scalable and highly available applications without requiring advanced infrastructure skills because the PaaS Layer takes care of activating/deactivating VMs for hosting applications on their behalf. The workload is automatically load balanced, similar to what the Access Layer does for IaaS, if the developer chooses to start multiple instances of the application.

Cloud Foundry[15] is the open source solution we selected for implementing the PaaS Layer released under the Apache License 2.0 and supported by the Cloud Foundry Foundation, established in December 2014, with EMC, HP, IBM, Intel, Pivotal, SAP and VMware as platinum members. Cloud Foundry is based on Linux Container (LXC) technology that isolates applications using operating system containers; this feature permits to run several applications on a single machine (virtual or physical) optimizing the resource usage. It is worth mentioning that, in case the programmer requires to activate several copies of an application for H/A, it is up to Cloud Foundry to transparently deploy them on different (virtual) machines.

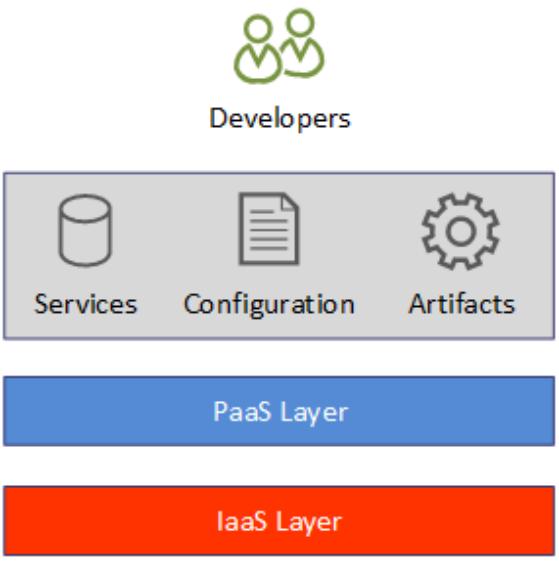

Figure 7. Platform as a service layer.

3.5 Administration Tools

SCP implements functions that both the platform administrator and the application owners are able to use for managing, monitoring and administering the cloud platform components as well as applications running in the cloud. The actions a user can perform depend on her role: the platform administrator has full control on all the objects deployed in the cloud (platform components and applications), whilst application owners have full control on their applications and can perform only some actions on the platform components. For instance, application owners have full control on databases and shared volumes used by their own applications but do not have any control on databases and shared volumes of other application owners.

3.6 Monitoring Module

The Monitoring Module is a component for verifying the working conditions of the resources in the SCP. The cloud administrator can monitor the resources used for implementing the platform services (e.g., the VMs used for the different layers as well as the physical servers of the IaaS) while the application owners can keep under control only the VMs of their own applications.

The Monitoring Module is implemented by Zabbix[16], a tool available under GNU General Public License (GPL) version 2 for monitoring the availability and performance of IT infrastructure components. According to the configuration, Zabbix, continuously gathers information from the servers under control and, in case one or more parameters reach a threshold value, it notifies the operator. Zabbix offers several monitoring options ranging from simple checks for verifying the availability/responsiveness of a server, to sophisticated measurements of parameters like CPU load, disk volume occupation, network traffic, number of processes, etc. Zabbix provides several ways for representing monitoring data in both graphical and textual/tabular format (Figure 8).

Zabbix can inform operators when a problem occurs with a server by sending an e-mail message, an Instant Messages (IM) or an SMS; we used such a feature for

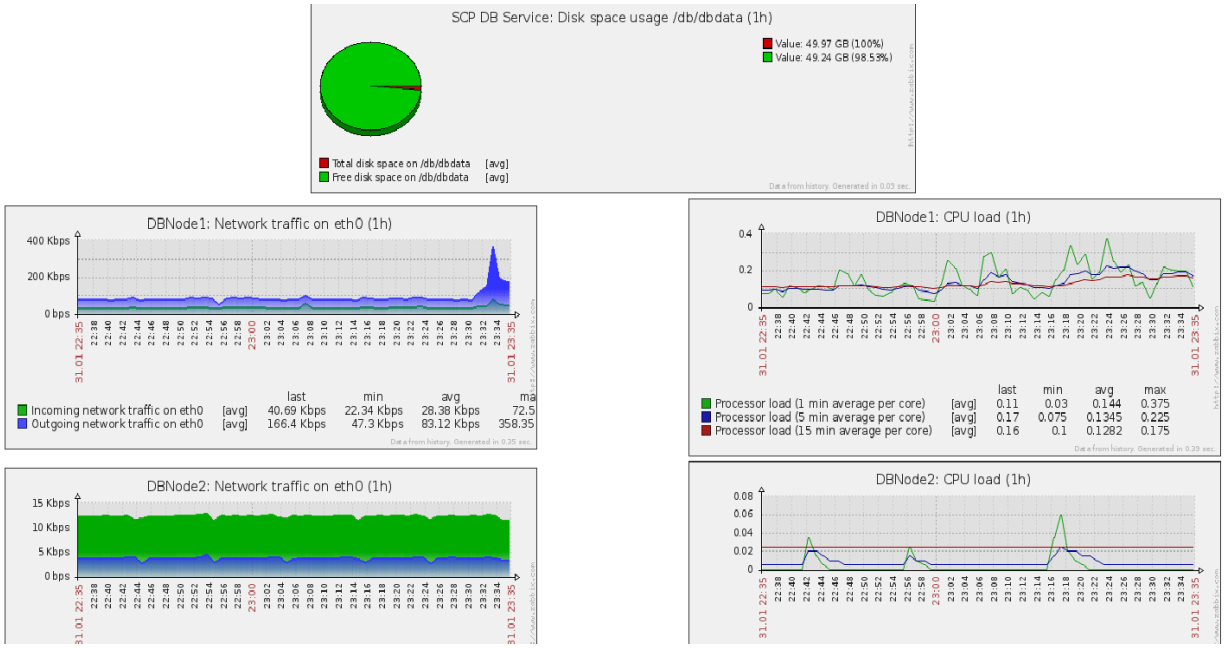

Figure 8. Zabbix monitoring page.

notifying the platform administrator or the application owners in order to request their intervention.

3.7 Database Administration Module

SCP provides web based tools for administering the supported database engines: we selected phpMyAdmin[17], for MySQL administration, phpPgAdmin[18], for PostgreSQL (Figure 9). They implement very similar functions for the corresponding database engine, such as creating, modifying and deleting databases and database objects (e.g., tables, indexes, etc.), submitting queries, importing/exporting data, managing database accounts, etc.

In the SCP context, the cloud administrator has full control on all the objects and configures database accounts for the application owners, giving them the rights of managing only the database objects created for their own applications.

3.8 Platform Administrator's Console

The Platform Administrator's Console is designed exclusively for the SCP administrator who needs to have full control on all the resources in the platform. Through the console, the administrator is able to manage all the elements at any level, IaaS Layer level included.

The console provides a Command Line Interface (CLI) suitable for recurrent tasks that can be automated using CLI scripts. The console is implemented as an Ubuntu Linux Server with the installation of the CLI interfaces of all the other components of the platform:

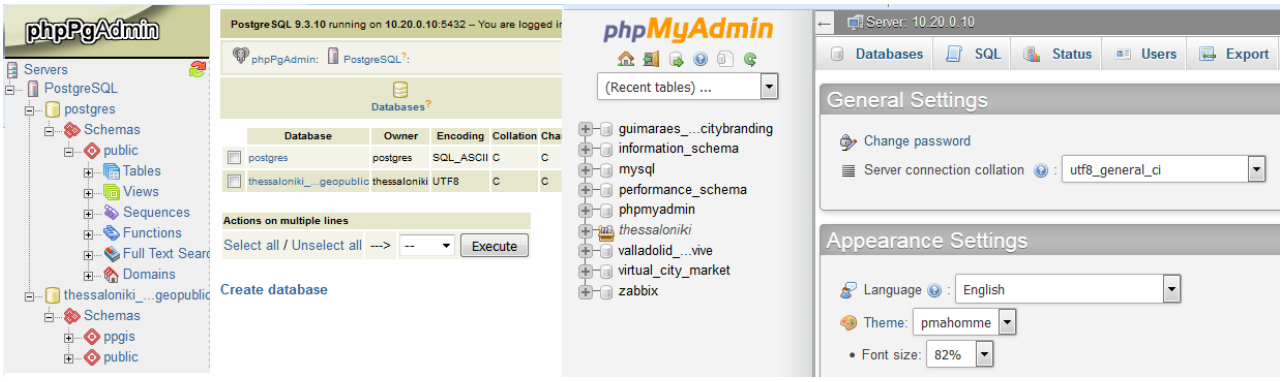

Figure 9. Database Administration — phpPgAdmin and phpMyAdmin.

- MySQL client and PostgreSQL client for managing the corresponding DB engines of the Database Services Module;
- Gluster client, for managing the File Sharing Services Module;
- OpenStack CLI software packages, for managing the IaaS Layer
- Cloud Foundry CLI, for managing the PaaS Layer.

The console is activated as a VM in the IaaS cloud when SCP is hosted in an OpenStack public cloud. However, it can also be deployed on a physical server when SCP is deployed on private cloud.

During the project, we heavily used the console for automating the deployment and the configuration both of the platform components and the application services. For this purpose we extensively used OpenStack Heat that permits the IaaS cloud user to describe all the IaaS objects that are needed for an application in a script — called stack — and to "*control the entire lifecycle of infrastructure and applications within OpenStack clouds*"[19]. In this perspective, the activation and deactivation of the IaaS objects can be simply obtained by 'submitting a stack' to Heat that is in charge of automatically creating/destroying the listed IaaS objects (e.g., VMs, Virtual Disks, etc.).

4. Conclusion

This article described the experience we gained on STORM CLOUDS, a project experimenting the migration of smart-city digital services to a cloud computing paradigm.

After analysing the problem, both from the organizational and the technical point of views, we decided to implement Storm Clouds Platform (SCP), a cloud computing infrastructure designed for hosting the applications selected by the project consortium.

During the project, we implemented two instances of the platform: one at Hewlett Packard Entreprise's premises (SCP@HPE), the other hosted at a public cloud-computing operator (SCP@Operator). We used SCP@HPE mainly for testing purposes and for supporting the 'cloudification process', consisting in the technical activities for porting the selected applications to cloud (e.g., adaptation, configuration, automation, etc.). SCP@Operator, on the other hand, was used as the "production environment", for making the migrated applications available to the end users on Internet.

SCP@HPE is a private cloud providing services for exclusive use of the STORM CLOUDS project partners while SCP@Operator is hosted on a public cloud. This exercise demonstrated that our solution supports both public and private deployment models allowing the project partners (in particular PAs) to decide how to manage their applications once the project terminates. In some cases, they may decide to keep their applications on a public cloud operator or, as an alternative, they can deploy services on equipment at their own sites, for instance for fulfilling privacy and security requirements.

All the software components used for the SCP implementation are available under an Open-Source Software (OSS) license, fulfilling one of the main requirements of the project. We selected broadly adopted software packages in order to guarantee long term support for the solution.

The architecture presented here is a baseline for future extensions and modifications with the objective of improving the way functions are implemented or for adding new functions not currently available. As an example, today — when an application owner uses the Data Service Layer — the platform administrator needs to create the database(s) and the shared volume (s) for the programmer to use. This operational model does not fulfils one of the fundamental requirements for a pure "as-a-service" paradigm in which services (in this case the database and the shared volumes) should be provided in a self-service manner without any intervention of the cloud administrator. OpenStack community is actively working on these aspects that are respectively addressed by Trove[20] and Manila[12] projects. Similar problems affect the monitoring functions we implemented and are addressed by another OpenStack project called Monasca[21]. When the project was started, these solutions were not available or they were in a very primitive state not suitable for a production-ready environment; consequently we decided to implement those functions following a more traditional, yet more proven, approach. Moreover, the solution we provide can be more easily replicated on public clouds based on OpenStack because, according to the OpenStack Market Place web page[22], at the time of writing only one operator offers Database-as-a-Service (implemented with Trove) and none of them implements Manila or Monasca.

Evolutions can be directed to support new programming and deployment paradigms like, for example, Docker[23] that is based on Linux containers (LXC), the same technology used by our Cloud Foundry based PaaS Layer. Actually, Docker can substitute Cloud Foundry but presents similar adoption problems: the

application architecture need to be reviewed in order to exploit containers at their full potential and this requires a strong commitment by the application owners with the related costs. It is worth to remember that this was the main reason why we designed a solution supporting both IaaS-based deployment paradigm (more traditional and with less impact on the applications) and a PaaS deployment paradigm (that would require adaptations).

In conclusion, we think that our experience demonstrates that cloud computing is a viable solution for implementing smart city services. Stakeholders can take great advantage of the inherent delivery model that promotes agility, speed and cost savings. Certain issues such as lack of control on how/where data and applications are managed/deployed and vendor-lock-in are still obstacles for the adoption of public cloud computing models. However, we feel that solutions like the one described in this paper are on the direction of alleviating the problem.

Conflict of Interest and Funding

No conflict of interest was reported by all authors. STORM CLOUDS, short form for "Surfing Towards the Opportunity of Real Migration to Cloud-based public Services", is a project partially funded by the European Commission within the 7th Framework Program (Grant Agreement No. 621089)[24].

References

1. STORM CLOUDS Consortium, 2013, *Surfing Towards the Opportunity of Real Migration to CLOUD-based public Services*, viewed January 30, 2016, <http://stormclouds.asi-soft.com/the-project/>
2. Amazon.com Inc., n.d., *Amazon Web Services — Main, Page*, viewed January 7, 2016, <https://aws.amazon.com>
3. Microsoft Corporation, n.d., *Microsof Azure — Main Page*, viewed January 10, 2016, <https://azure.microsoft.com>
4. *Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data*, n.d., viewed January 11, 2016, <http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=urisery:OJ.L_.1995.281.01.0031.01.ENG>
5. Google Inc., n.d., *Google Maps — Main Page*, viewed January 2, 2016, <https://maps.google.com>
6. *CORDIS — Free and open source software activities in European Information Society initiatives*, n.d., viewed January 4, 2016, <http://cordis.europa.eu/fp7/ict/ssai/foss-home_en.html>
7. *LAMP (software bundle) — Wikipedia page*, n.d., viewed January 30, 2016, <https://en.wikipedia.org/wiki/LAMP_(software_bundle)>
8. Mell P and Grance T, 2011, *The NIST definition of cloud computing*, viewed January 29, 2016, <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
9. *Virtualization — Wikipedia page*, n.d., viewed January 20, 2016, <https://en.wikipedia.org/wiki/Virtualization>
10. *OpenStack project*, n.d., viewed February 1, 2016, <https://www.openstack.org>
11. *Top 10 open source projects of 2014*, n.d., viewed February 1, 2016, <https://opensource.com/business/14/12/top-10-open-source-projects-2014>
12. *OpenStack Manila — Wiki page*, n.d., viewed January 22, 2016, <https://wiki.openstack.org/wiki/Manila>
13. *Gluster — main page*, n.d., viewed January 15, 2016, <https://www.gluster.org>
14. *HAProxy — main page*, n.d., viewed January 17, 2016, <http://www.haproxy.org>
15. *Cloud Foundry — main page*, n.d., viewed January 11, 2016, <https://www.cloudfoundry.org>
16. Zabbix Company, n.d., *Zabbix — main page*, viewed January 29, 2016, <http://www.zabbix.com>
17. *phpMyAdmin — main page*, n.d., viewed January 21, 2016, <https://www.phpmyadmin.net>
18. *phpPgAdmin — main page*, n.d., viewed January 21, 2016, <http://phppgadmin.sourceforge.net/doku.php>
19. *OpenStack Heat — Wiki page*, n.d., viewed January 21, 2016, <https://wiki.openstack.org/wiki/Heat>
20. *OpenStack — Trove page*, n.d., viewed January 22, 2016, <https://wiki.openstack.org/wiki/Trove>
21. *OpenStack Monasca — Wiki Page*, n.d., viewed January 22, 2016, <https://wiki.openstack.org/wiki/Monasca>
22. *OpenStack marketplace — web page*, n.d., viewed January 23, 2016, <https://www.openstack.org/marketplace/public-clouds>
23. Docker Inc., n.d., *Docker main page*, viewed January 23, 2016, <https://www.docker.com>
24. *Storm Clouds Project — European Commission project page*, n.d., viewed July 1, 2014, <http://ec.europa.eu/digital-agenda/en/storm-clouds-project-cloud-public-services>
25. *Vendor lock-in — Wikipedia page*, n.d., viewed January 12, 2016, <http://en.wikipedia.org/wiki/Vendor_lock-in>
26. *OpenStack — Neutron/LBaaS page*, n.d., viewed January 15, 2016, <https://wiki.openstack.org/wiki/Neutron/LBaaS>
27. *Puppet open source*, n.d., viewed January 24, 2016, <https://puppet.com/product/open-source-projects>
28. *Zabbix license page*, n.d., viewed January 25, 2016, <http://www.zabbix.com/license.php>
29. *High-availability cluster — Wikipedia page*, n.d., viewed January 26, 2016, <https://en.wikipedia.org/wiki/High-availability_cluster>

RESEARCH ARTICLE

Social network services for innovative smart cities: the RADICAL platform approach

Fotis Aisopos[1*], Antonios Litke[1], Magdalini Kardara[1], Konstantinos Tserpes[1], Pablo Martínez Campo[2] and Theodora Varvarigou[1]

[1] Distributed Knowledge and Media Systems Group, National Technical University of Athens, Zografou Campus, Athens, 15773, Greece

[2] Communications Engineering Department, University of Cantabria, Santander, Cantabria, 39005, Spain

Abstract: In this paper we present the RADICAL platform, a software stack that enables the combination of social network (SN) services and Internet of Things (IoT) in the context of innovative smart cities. RADICAL makes possible the development and deployment of interoperable pervasive multi-sensory and socially-aware services; facilitates smart governance and flexible replication of services across cities and regions through a Virtual Machine generation mechanism in a sophisticated cloud environment. A large scale piloting of the platform integrates, deploys and tests various services in the areas of Cycling Safety, Participatory Urbanism, Augmented Reality and others while a large group of citizens from different countries are actively involved in the co-creation, validation and evaluation of the RADICAL approach on the basis of an innovative Living Labs approach.

Keywords: smart cities, cloud environment, internet of things, social networks, Living Labs

*Correspondence to: Fotis Aisopos, Distributed Knowledge and Media Systems Group, National Technical University of Athens, Zografou Campus, Athens, 15773, Greece; Email: fotais@mail.ntua.gr

Received: January 22, 2016; **Accepted:** April 18, 2016; **Published Online:** May 9, 2016

Citation: Aisopos F, Litke A, Kardara M, *et al*., 2016, Social network services for innovative smart cities: the RADICAL platform approach. *Journal of Smart Cities*, vol.2(1): 26–40. http://dx.doi.org/10.18063/JSC.2016.01.004.

1. Introduction

Modern cities are increasingly turning towards ICT technology for confronting pressures associated with demographic changes, urbanisation, climate change[1] and globalisation. Therefore, most cities have undertaken significant investments during the last decade in ICT infrastructure including computers, broadband connectivity and recently, sensing infrastructures. These infrastructures have empowered a number of innovative services in areas such as participatory sensing, urban logistics and ambient assisted living. Such services have been extensively deployed in several cities, thereby demonstrating the potential benefits of ICT infrastructures for businesses and the citizens themselves.

During the last few years, we have also witnessed an explosion of sensor deployments and social networking services along with the emergence of social networking[2] and Internet of Things (IoT) technologies[3,4]. Social and sensor networks can be combined in order to offer a variety of added-value services for smart cities, as has already been demonstrated by various early IoT applications (such as WikiCity[5], CitySense[6], GoogleLatitude[7]), as well as applications combining social and sensor networks[8–10].

Recently, the benefits of social networking and IoT deployments for smart cities have also been demonstrated in the context of a range of EC co-funded projects[11,12]. Despite the proliferation of social networking infrastructures and sensor networking infrastructures and deployments, there is still no easy way

to develop, customise, deploy and operate such services in smart cities. Smart cities today have various needs for becoming smarter. They need to optimally exploit their already existing infrastructures (IoT devices such as sensors throughout the urban landscape) for providing new innovative services to the citizens. The ultimate goal is to increase the participation levels of end users/citizens in the daily activities of the city and increase their well-being.

There are various initiatives on the European and global scale on how cities can become smart cities. Among others, the most established initiatives include EUROCITIES[13], OASC[14] and the development efforts made in the context of EIT's digital action line "Urban Life and Mobility"[15]. Furthermore, the coordination actions project FP7 CA FIREBALL[16] is providing a substantial analysis on how European cities are currently developing strategies in order to become smart cities and the lessons we can draw for the future. In the research efforts, we can find interesting publications and results in journals such as the ISJ special issue on smart cities[17].

Motivated by the modern challenges in smart cities, the RADICAL (Rapid Deployment for Intelligent Cities and Living) approach[18] opens new horizons in the operation of intelligent services in smart cities, notably services that could be flexibly and successfully customised and replicated across multiple cities. Its main goal is to provide the means for cities and ICT companies to rapidly develop, deploy, replicate, and evaluate a diverse set of sustainable ICT services that leverage established IoT and SN infrastructures. Eight distinct pilot services were built upon the RADICAL platform architecture dealing with: (i) Cycling Safety Improvement, (ii) Products Carbon Footprint Management, (iii) Object-driven Data Journalism, (iv) Participatory Urbanism, (v) Augmented Reality, (vi) Eco-consciousness, (vii) Sound map of a city, and (viii) City-R-Us—a crowdsourcing app for collecting movement information using citizens smartphones.

RADICAL provides an easy way to develop and customise smart city services bridging both IoT and SN domains. This is due to the fact that it provides special tools to do so, namely the Application Development toolkit, which makes the creation, adaptation and configuration of services a user friendly, clear and easy to manage procedure. The primary targeted users of the RADICAL platform are city authorities who can become the early adopters. Their needs are summarised on how to optimally leverage existing ICT infrastructures (e.g., smart city infrastructures with sensors, etc.) with social networks in order to engage the citizens, interact with them and offer them innovative services.

Replication across multiple cities covers the following need—to have a RADICAL platform with maximum usability and adaptability to other city's contexts. This is particularly beneficial for multiple stakeholders: (i) ICT companies for providing analytics, consultancy or other added value services to city authorities, (ii) citizens for being aware about what is happening around them, (iii) policy makers for taking decisions either in a local government level or at national level. Policy makers (city officials), through the sustainability and replicability of added value services, are able to understand which services fit with their organisational, infrastructural, geographical and socioeconomic characteristics. Accordingly, smart cities could benefit from a technical infrastructure enabling the rapid and effective customisation of the identified services in their environments.

Thus, RADICAL can provide a series of benefits and value proposition for a series of end-users: (i) for city officials by enabling them to have a better control and knowledge on their infrastructures, while engaging citizens in communication and participation with authorities and being able to offer them innovative services on top of IoT and SN; (ii) for city developers as they can build innovative services for the smart cities (deploying IoT their infrastructures and making use of social networks), and (iii) for citizens who can receive smart and innovative services through the joint collaboration of SMEs (providers of services) and the city authorities.

The specific objectives of the current paper are focused on:

(i) Showcasing the RADICAL platform approach on how to aggregate and combine IoT and SN data for the benefit of the city authorities and citizens of smart cities,

(ii) Presenting and analysing the validated results of the pilots implemented through a Living Labs approach,

(iii) Demonstrating the RADICAL approach on cross-city data correlation.

The rest of the paper is structured as follows: Section 2 gives an overview of related and similar works that can be found in the international literature and in projects funded by the European Commission; Section 3 presents the RADICAL architecture and approach;

Section 4 provides the pilot scenarios and the city experiments' configuration and results, along with the citizens' feedback, while in Section 5 we provide the future work to be planned in the context of RADICAL and the conclusions we have come into.

2. Related and Similar Works

There are various efforts today that are focused in providing innovative services in the context of future smart cities. We present some of them along with a comparative analysis against the approach of RADICAL below.

The European Platform for Intelligent Cities[19] is an EU-funded project that aims to wed state-of-the-art cloud computing technologies with mature e-Government service applications to create the first truly scalable and flexible pan-European platform for innovative, user-driven public service delivery. The EPIC project aimed to provide a way for Cities to deliver and share 'smarter services' in a flexible and cost effective way which would not involve large-scale reorganisation of their ICT infrastructure. A cloud-centric vision on Smart Cities is also provided by Gubbi *et al.*[20] where the current trends in IoT research propelled by applications is presented. The authors also discuss the need for convergence in several interdisciplinary technologies and a case study of data analytics on the Aneka/Azure cloud platform is given.

A proof-of-concept of Smart City IoT deployment was provided in the city of Padova, Italy, in the context of Padova Smart City project[21]. In this case, a full range of IoT solutions and services were interconnected with the data network of the city municipality and data collected were further analysed and presented as a relevant example of application of the IoT paradigm to smart cities.

The SMARTiP project[22] is aimed at enhancing the ability of the cities to grow and sustain a 'smart city' ecosystem which can support new, emerging opportunities for a dynamic co-production process resulting in a more inclusive, higher quality and efficient public services which can then be made replicable and scalable for cross-border deployment on a larger scale. Citadel on the Move[23] aims to make it easier for citizens and application developers alike from across Europe to use Open Data to create the type of innovative mobile applications that they want and need. Citadel on the Move aims to fulfill this need by: (i) creating formats that make it easier for local government to release data in usable, interoperable formats, and (ii) providing templates that make it easier for citizens to create mobile applications that can be potentially shared across Europe, creating services that can be used on any device, anytime, anywhere.

The objective of PERIPHÈRIA[24] is to deploy convergent Future Internet (FI) platforms and services for the promotion of sustainable lifestyles in and across emergent networks of 'smart" peripheral cities in Europe, dynamic realities with a specific vocation for green creativity. Its Open Service Convergence Platform, an "Internet by and for the People", extends and enhances the Save Energy project's Social Information Architecture, integrating key new components—sensor networks, real time 3D and mobile location-based services—with the FI paradigms of Internet of Things (IoT), Internet of Services (IoS) and Internet of People (IoP). PERIPHÈRIA develops the Living Lab premise of shifting technology R&D out of the laboratory and into the real world in a systemic blend of technological with social innovation.

Open Cities[25] was a project co-funded by the European Commission that aimed to validate how to approach open and user driven innovation methodologies to the public sector in a scenario of FI services for smart cities by leveraging existing tools, trials and platforms in crowdsourcing, open data, Fiber to the Home (FTTH) and Open Sensor Networks in seven major European cities—Helsinki, Berlin, Amsterdam, Paris, Rome, Barcelona and Bologna. The project provided different types of contributions such as new understandings on how to approach open innovation from the public sector, functioning platforms for open data and open networks, and actual FI services provided by developers using these platforms.

Even though the sustainability of the above results was ensured by the presence of all relevant actors—city authorities, research institutions, companies and more importantly, real users using the services and creating the necessary demand for FI services in Smart Cities—its services are not as easy to replicate as RADICAL's. Besides, services developed in the context of Open Cities project do not exploit the potential of the sentiments analysis expressed through Social Networks which is one of the key strengths of RADICAL.

Other recent ongoing efforts include projects such as GrowSmarter[26] and Triangulum[27]. The former brings together cities and industry to integrate and demonstrate 12 smart city solutions in energy, infrastructure and transport while the latter is one of the three Euro-

pean smart cities and communities lighthouse projects, set to demonstrate, disseminate and replicate solutions and frameworks for Europe's future smart cites. It will serve as a testbed for innovative projects, focusing on sustainable mobility, energy, ICT, and business opportunities. The project will demonstrate real smart city solutions with working business and social value models and will facilitate and replicate them across three more follower cities.

Overall, the RADICAL platform approach is set apart from all these efforts by trying to merge in an innovative way based on its unique value proposition, the benefits and functional capabilities of IoT and social networking services. The wide set of application services that are deployed and piloted are collecting all the necessary features for a thorough validation of the RADICAL concept and the benefit it can bring for citizens of the smart cities in the future. Table 1 gives a comparative analysis between RADICAL and the most recent efforts in the area of smart cities.

3. The RADICAL Approach

The RADICAL platform integrates components and tools from SocIoS[28] and SmartSantander[29] projects, in order to develop innovative smart city services leveraging information stemming from social networks (SN) and IoT devices.

Most existing smart city solutions focus on the IoT data aggregation, in order to provide intelligent services to the citizens. RADICAL platform on the other hand, uses SmartSantander IoT infrustructure as a basis, enriched by the social networking services and analytics of SocIoS, along with added value Governance and application-development components that will be analysed in the following sections to provide a complete and sophisticated smart city solution. Thus, the RADICAL platform is able to collect, analyse, combine, process, visualise and provide uniform access to three main types of data—Social Network content (from SocIoS), Internet of Things data collected from sensors and devices (from SmartSantander) and smartphone data specific to pilot scenarios collected by city services themselves.

The architecture of RADICAL is depicted in Figure 1[30] along with a legend of colors explaining the use of each module group. As can be seen, there are three distinct architectural layers from the top to the bottom:

- The **Service Application Layer**: This is external from the RADICAL platform and presents an open list of the City Services (installed in local municipalities) providing a Front-end to the IoT/SN data aggregated in the RADICAL platform. Some services (depending on their functionality) allow citizens to view analysed platform data
- The **Platform Layer**, including Application Management and Platform tools and modules, connecting and exposing its data through Application Programming Interfaces (API)
- The **Data Sources Layer**, including IoT devices, smartphone City Applications and various Social Networks, feeding the RADICAL platform with the three types of data mentioned above. For Social Networks, RADICAL can also post data (e.g., tweets) thus connections are bidirectional

Data coming from the lower layer, with the exception of Social Network-related ones, are saved in the RADICAL platform Repository by the relevant **Data Storage** modules. The **IoT Data** Modules on the left were adapted from existing SmartSantander platform components and include components parsing and pushing IoT device measurements into the Repository (**Service Aggregator** and **Storer —** one direction data flow), as well as components managing the IoT devices registered (Register Manager, IoT Manager).

One the other hand, Social Network Data Retrieval Modules access data in real time from the underlying Social Networks via the **SN Core Services** and **Adaptors**, originated from the SocIoS platform. For each

Table 1. RADICAL analytical comparison with recent similar works

Name	Triangulum	Open Cities	GrowSmarter	RADICAL
Exploitation of SN Capabilities	No	Yes	No	Yes
Cloud Hosting	Yes	No	Yes	Yes
Provides APIs	Yes	Yes	Not explicitly mentioned	Yes
Living Labs methodology	Yes	Yes (Urban Labs)	Yes	Yes
What are differences/complementarities with RADICAL	3 Lighthouse cities and 3 followers. Focus on replication. Applications in energy saving, urban planning and others.	Test bench for innovative Apps and services. Free support for 6 months for technical tests and obtaining real and formalised feed- backs from 60 users.	3 Lighthouse Cities. Low energy district, waste heat recovery, smart waste collection.	6 cities/ regions

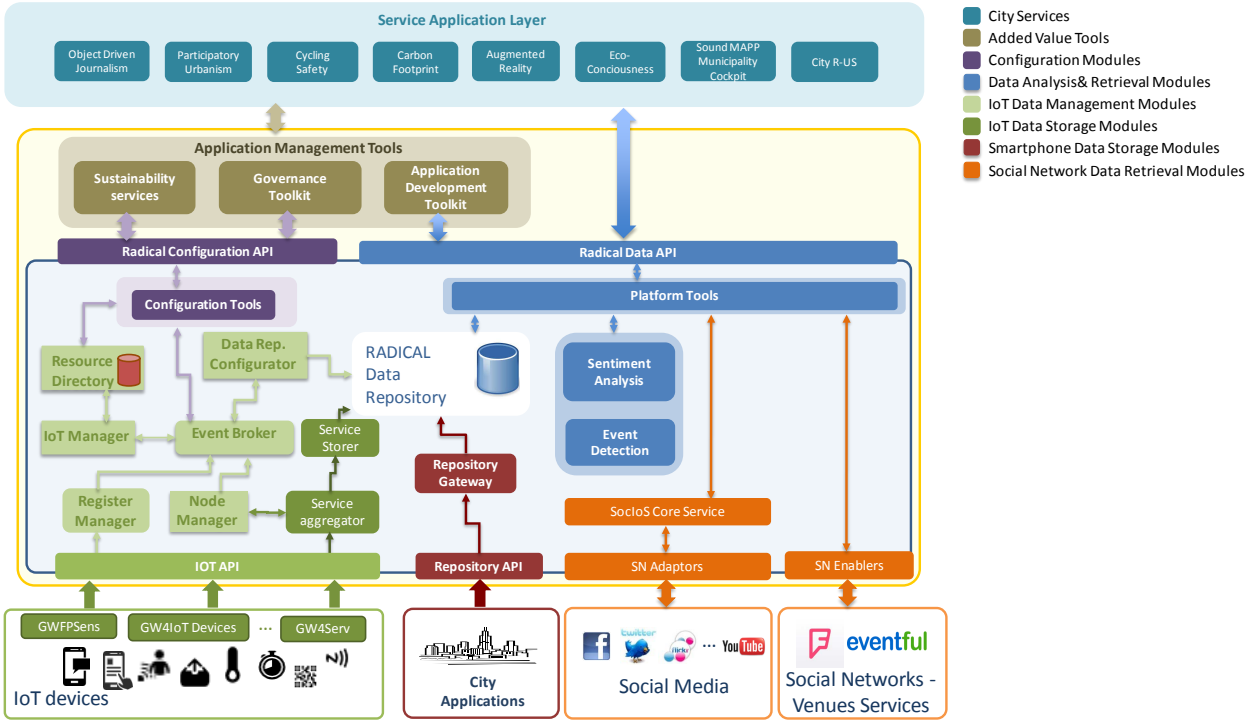

Figure 1. RADICAL architecture.

supported SN a respective adaptor is provided, encapsulating its data structures and functionality. Apart from SocIoS-parsed data, **Social Enablers** are also used to retrieve venue-related information data for more plain Social Networks like Foursquare, where SN Adaptors' implementation does not make sense.

Platform and **Configuration tools** as well as **Data Analytics** services (Sentiment Analysis, Event Detection) provide more sophisticated use and combination of RADICAL data, as needed for City Services Front-end presentation (Service Application Layer). Any data transfer between the RADICAL platform and City Services or IoT infrastructures is made possible through the respective **APIs** (Configuration API, Data API, IoT API, Repository API) that will be presented below.

Lastly, on top of the core platform, RADICAL delivers a set of tools (**Application Management Tools**) allowing City Service administrators and developers to configure the RADICAL platform or combine its various functionalities. Most notably, the **Application Development Toolkit** added value service is the one combining different types of data originated from various cities and will be further analysed in Section 3.3.

3.1 RADICAL APIs, Authentication and Data Security

The main functionality of the RADICAL platform is exposed through the **Data API** which allows smart city services to access the different sources of information (social networks, IoT infrastructures, city applications), perform analysis and combine data by using the appropriate platform tools. In addition to that, the platform exposes a second API, the RADICAL **Configuration API**, which allows smart city service administrators to configure the RADICAL platform through the **Data Repository Configurator** and manage the IoT devices listed in the **Resource Directory**. Data flows for both APIs are bidirectional, as API consumers can pull (retrieve SN/IoT data, extract platform configuration) or push data (make a new SN post, set configuration parameters).

Finally, aiming at data forwarding to the RADICAL platform, the RADICAL **IoT API** provides the interface for registering devices and feeding relevant IoT measurements and reported events as observations, while the **Repository API** allows City applications to push their data observations to the RADICAL platform and store them into the RADICAL data repository.

Regarding the aforementioned APIs, a major concern raised was to ensure the integrity and confidentiality of all data sent or retrieved through them, in order to protect citizens' privacy when using smart city services. Thus, a sophisticated authentication system was built for restricting usage of those APIs, following the same logic as Social Networks' APIs' au-

thentication — city service administrators are provided with specific credentials, which must be passed to an authentication method that returns an "API key", in the form of a 32-character hash. API keys are updated periodically (as dictated by the corresponding configuration parameter) and are required for any API call that requests or sends city data.

Moreover, to install and use a smart city app connected to RADICAL; end-users are asked to provide their consent and are informed about all anonymised data that their smartphones provide to RADICAL, such as GPS location or events reporting. Reassuring end-users that their data is being processed anonymously and confidentially is crucial, as much concern is now publicly raised on privacy and security around online networks and smartphone-related applications that citizens use in their everyday life[31,32].

3.2 Social Network Adaptors and Data Analytics Services

The main Social Networking related functionality in RADICAL is provided by the SocIoS services, i.e., a set of tools and mechanisms for leveraging the potential of Social Networking Sites. The SocIoS framework is a software stack that operates on top of Social Networking Sites' APIs. It provides an abstraction layer for aggregating data and functionality from a multitude of underlying social media platforms as well as a set of analytical tools for leveraging that functionality.

The SocIoS tools that are integrated into the RADICAL platform are the SocIoS API[33], an abstraction layer providing uniform access to the data and functionality of the most popular Social Networks, and two analysis services, i.e., the Event Detection Service and the Sentiment Analysis Service.

The SocIoS API exposes operations that encapsulate the functionality of the underlying SN APIs. It is based on its own generic object model, encapsulating the entities and relationships residing in the underlying SN.

For each supported SN API, an adaptor has been implemented transforming SN data to the SocIoS object model, allowing the API consumers to access data from the respective underlying SN in a uniform way. The design overview of the SocIoS API and the supported Social Networks is depicted in Figure 2[30].

The Event Detection Service aims to enable RADICAL end users to detect and monitor real-world events that are defined by citizens' activity. The service analyses the comments, tweets and other text messages generated by the citizens and classifies them to categories that are most likely to relate to events. It also generates a set of keywords that define each event so as to assist the end user in understanding the event context.

The goal of the Sentiment Analysis Service is to extract sentiment expressive patterns from user-generated content in social networks or any other types of text posts. The service comes to the aid of RADICAL's city services' administrators, helping them to categorise sentimentally charged texts, e.g., analyse citizens' posts, to separate the subjective from objective opinions or count the overall positive and negative feedback concerning a specific topic.

3.3 Application Development: Integrate and Analyse SN and IoT Data from Various Cities

One of the main added values that RADICAL platform provides to Smart Cities is the ability to form sophisticated Applications integrating social and IoT data from different cities and perform data analytics tasks

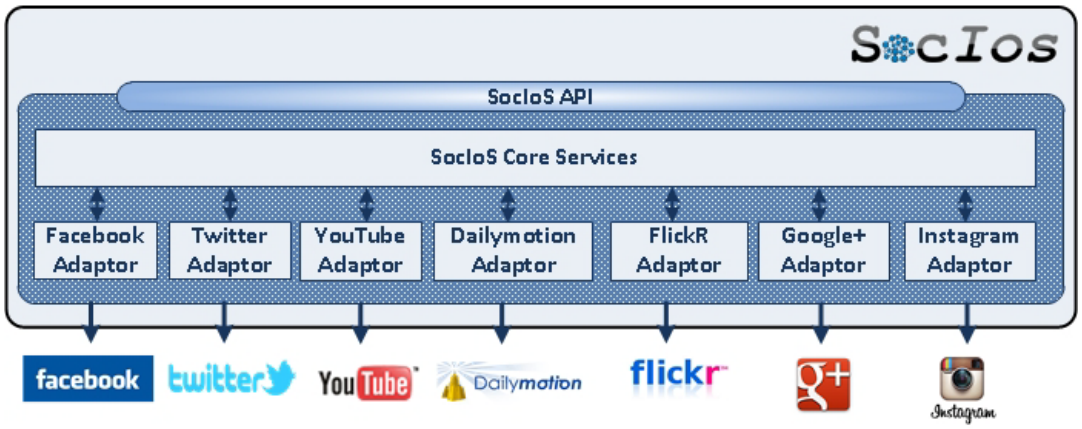

Figure 2. SocIoS API.

on the aggregated data. The building of such Applications is achieved through the RADICAL Application Development Toolkit.

The Application Development Toolkit based on the WebHookIt open-source tool[34], is located in the Application Management layer (Figure 1) and combines various data through web services offered by the platform. It can be used both as a mash-up tool, enabling the concurrent APIs' and services' consumption and output aggregation, as well as a service composition tool, constructing sequential service workflows.

For this purpose, as can be observed in the architecture figure, it consumes the RADICAL Data API, to get access to the RADICAL Repository data and the integrated Social Networks. The tool functionality is exposed to the City Service developers via a usable graphical interface (GUI), where they can create complex RADICAL Applications, using simple drag and drop actions. Through RADICAL Applications, the city service administrators can multi-query for SN and IoT from various cities and get an analysis of the combined results.

The Application Development Toolkit also includes a scheduling functionality for periodic Application executions; an Exception Handler for a meaningful management of the runtime errors, and a Service Discovery component which is using a Yahoo! Query Language module for services retrieval from a Web Application Description Language (WADL) document. After the list of services is retrieved, the tool end-user can use or combine services into a workflow Application, by dragging and dropping them into the wiring and mapping input and output.

Figure 3 presents an example wiring workflow, mining a Social Network (Twitter) for posts with specific keywords (concerning the Fort d'Issy POI) and applying data analytics (Topic Detection) to the results to get a parsed JSON output. For this purpose the respective RADICAL Data API web methods are called (findTextMediaItems, getTopicsFromTextItems) with the relevant parameters (sns="Twitter", keywords="issy, fort") and the result is parsed by an XML parsing component (xml2js) to be presented as a list of topics in the output panel below.

Note that the resulting wiring workflow can form the basis of a web-based Application. To this end, a City Service developer can create a web panel as part of the city administration portal, presenting the output results of an Application in a user friendly manner. In the example shown above, through this panel the city administrators can be informed of what is reported in the city through IoT devices or Social Networks in the form of a list of events.

3.4 Correlating Cross-city Data for a RADICAL City Service

In Smart Cities, a major need of city officials in order to administer City Services is a tool for controlling

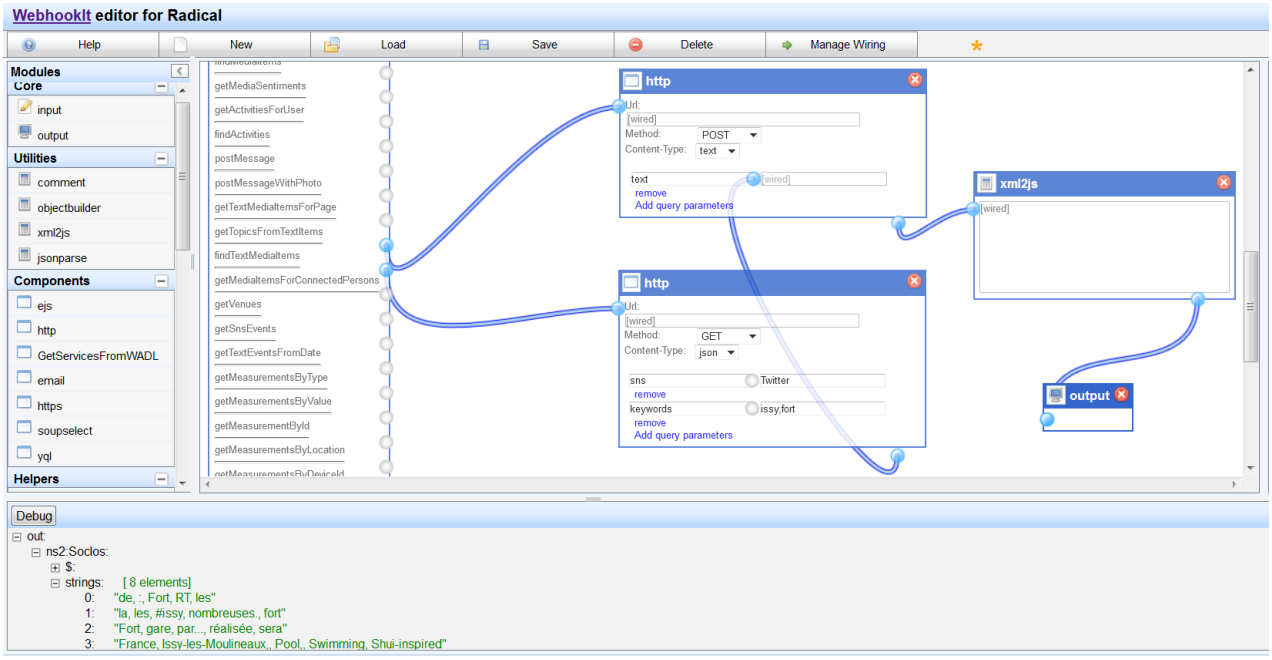

Figure 3. Wiring application services in WebhookIt editor GUI.

large amount of data related to any facet of those services. This tool must present aggregated usage, as well as qualitative characteristics of citizens' activity for every Smart City Service.

Towards this direction, a Smart Cities Dashboard was introduced by RADICAL, implemented as a web interface that presents information about IoT data sent by any device registered in the platform, such as smartphones sending citizens' geo-location and activity, weather stations with climate measurements or even bicycles carrying RFID tags. By the use of this web application, City Service administrators can view aggregated statistics from one or various cities for the same City Service during a specific time frame. Diagrams and charts with all devices' measurements as well as registrations are provided per day for each service, for every city that this service running. This way, the overall usage and user acceptance of a specific City Service can be estimated and also compared with another RADICAL city running a service of the same nature (e.g., Augmented Reality for POIs).

Thus, depending on the City Service chosen, city officials are able to retrieve reliable statistics about citizens' behavior on work weeks, weekends, bank holidays or areas popular among citizens. A typical example of this can be seen in Figure 4 for the Augmented Reality pilot scenario (described in Section 4) comparing Santander and Cantabria Smartphone and POIs' reports Registrations.

Another example involves services related to weather monitoring that provide precise insight into climate or gas emissions such as carbon footprint. Figure 5 presents graph comparisons of daily values sent to the carbon footprint monitoring City Service from relevant sensors, between 2 pilot cities (Genoa and Cantabria).

In addition to sensor measurements, RADICAL Pilot Services' scenarios also report events, e.g., when citizens report incidences taking place in an area. In this case, a correlation of activity and event types is provided by the Dashboard, as shown in Figure 6, presenting device registrations and events reported in the context of the Participatory Urbanism City Service in the cities of Santander and Issy Les Moulineaux.

4. RADICAL Experiments: Configuration, Pilot Scenarios and Evaluation Results

4.1 Evaluation Method

RADICAL evaluation aims at demonstrating the platform's technical soundness, in terms of the fine-grained implementation and operation of the platform tools, and the ease of the virtualised smart solution replication for a city, as well as the illustration of its added value for the various types of end-users (citizens, city officials, etc.). More specifically, the purpose

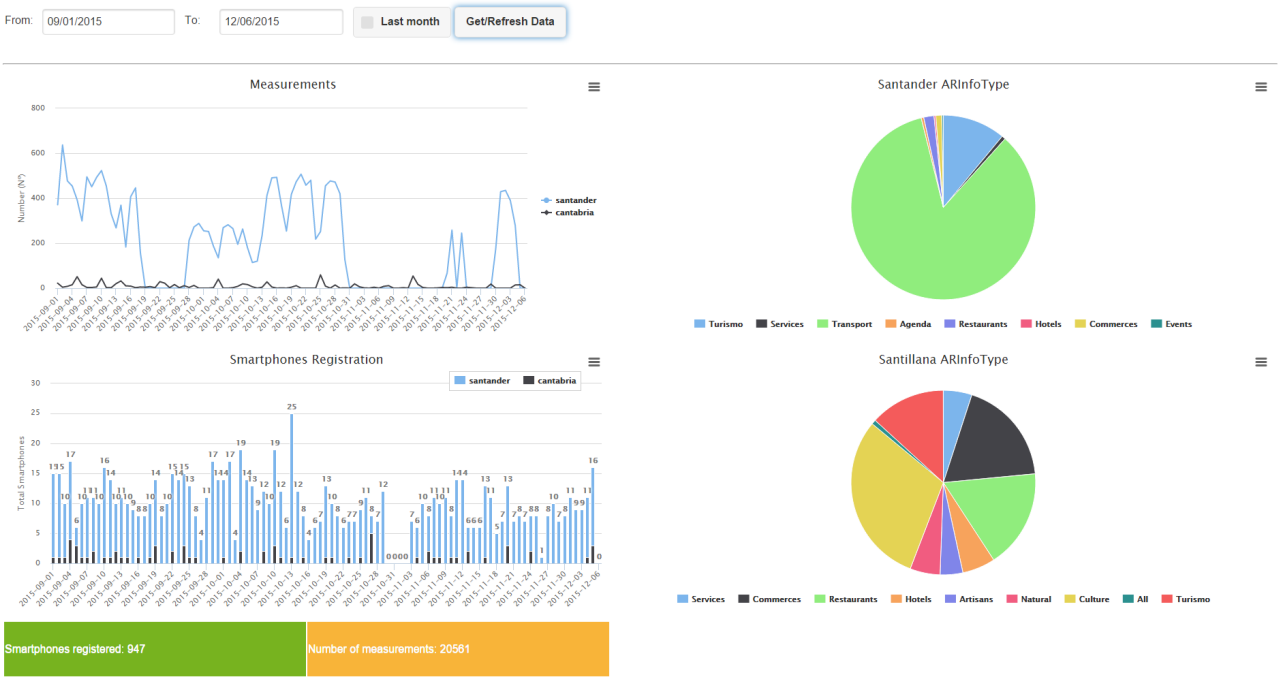

Figure 4. RADICAL Cities Dashboard presents aggregated citizens' activity.

Figure 5. RADICAL Cities Dashboard compares climate-related sensor measurements between cities.

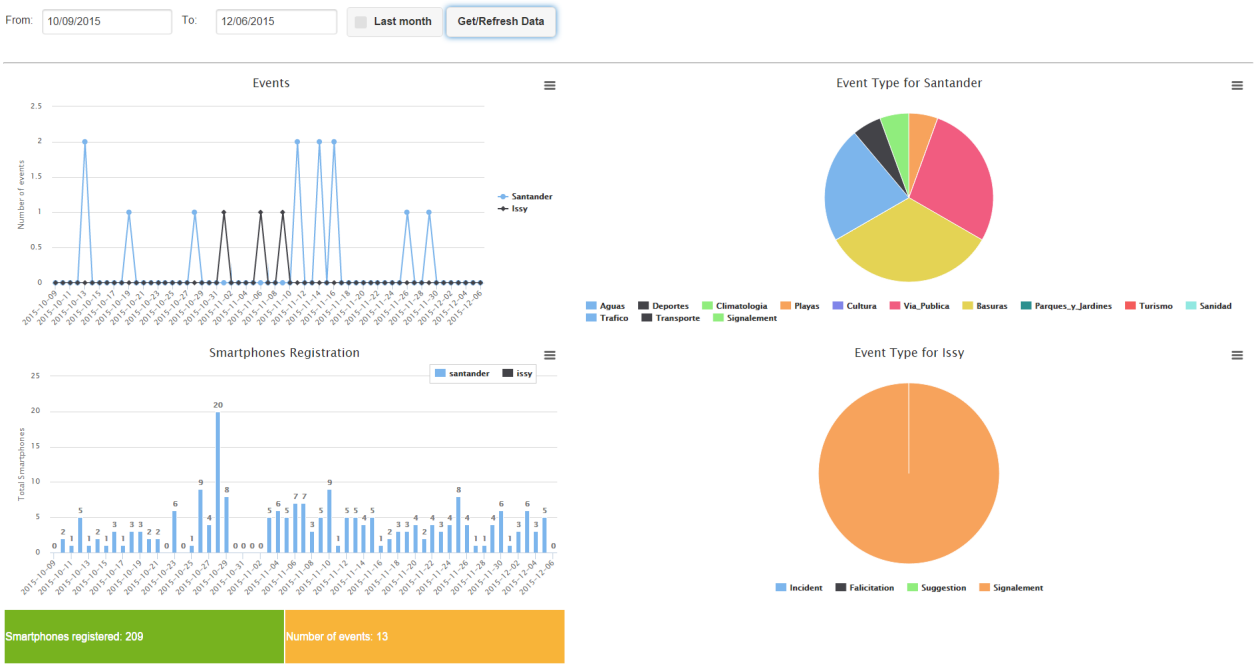

Figure 6. RADICAL Cities Dashboard compares events in the Participatory Urbanism service of different cities.

of evaluation performed in the context of RADICAL was three-fold:

- **Technical Coherence**: A set of stress tests was employed to validate the platform operation under heavy data load
- **Smart Exploitation Potential**: City administrators with the help of technical experts evaluated RADICAL in terms of usability, sustainability and replicability
- **User Acceptance**: A major concern was to evaluate citizens' satisfaction and added value for a Smart City derived from the RADICAL operation

Regarding the latter, assuring the engagement and validation of citizens' participation in pilot cities, we

decided to follow an innovative Living Labs approach to observe RADICAL's impact on dedicated citizens' groups in the long term. This involved representative samples of the population, in order to study the penetration and sustainability of the services within each individual city.

The employed Living Labs approach[35] is going beyond most conventional Living Labs that commonly take into account representative samples in testing and experimentation of novel ICT services. RADICAL extended existing LL methodologies by involving the end-users not only in the requirements of the capture and design phases but also in the testing validation, evaluation and feedback phase. While such an approach has been taken in other solutions, it is the breadth of the user communities and the involvement of our multi-disciplinary expert communities (technology SMEs, public bodies, citizens with different interests) supporting the process, which maximises the impact of RADICAL[36].

Unlike previous efforts, RADICAL's approach is broader as it involves expert users from all relevant disciplines and citizens' contribution in all phases, to multiple, concurrently developed modules (from application services for cyclists to one Platform supporting the services). Thus Living Labs' participants contribute in multiple aspects of RADICAL's developments and not only to the Labs that they have participated in.

4.2 Pilots Setup and Scenarios

For the establishment of Living Labs in different areas, RADICAL was piloted in six cities (Aarhus, Athens, Genoa, Issy les Moulineaux, Santander and Cantabria region), with the support of respective municipalities. To logically separate city repositories and data control access, it was decided to follow a "one platform instance per city" deployment approach.

For this purpose, the BonFIRE[37] cloud infrastructure was employed, providing one Ubuntu Linux Virtual Machine (VM) for each city where the corresponding RADICAL Platform instance and Data Repository (MySQL Database) were installed. A "Template" VM with generic data was also kept for backup (Figure 7). The advantages of this approach are apparent in case a new city decides to adopt RADICAL—the "template" instance can be easily replicated for every new entry with a minimum effort of parameterising and defining the domain name of the new instance.

RADICAL pilot scenarios were selected with a view

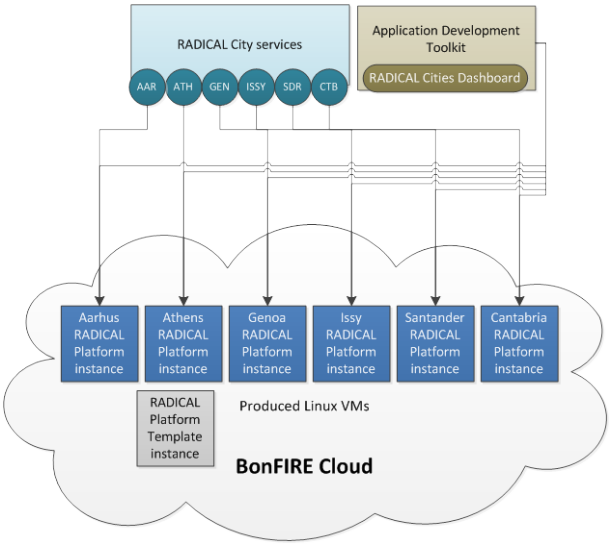

Figure 7. RADICAL deployment diagram.

to maximise its impact and meeting the challenges of future smart cities. Specifically, the following criteria have driven the selection:

- **Societal Challenges**: RADICAL scenarios are representative cases of ICT services addressing the emerging societal challenges in the urban environment (demographic trends, climate change).
- **Involvement of Multiple Stakeholders**: RADICAL introduces a holistic approach to the development, deployment and operation of ICT services in urban environments. Thus, RADICAL scenarios involve all the envisaged stakeholders.
- **Innovation**: The services combine knowledge from SN and IoT to provide real-time information and intelligence to citizens. Such services have not been deployed at large in the scope of modern cities.

City Services present a fair degree of heterogeneity and diversity in terms of their socio-economic and legal characteristics. This is intentional and serves the purpose of studying different deployment cases and associated governance schemes. An overview of these services is presented below, along with the benefits for citizens and relevant stakeholders:

(1) Cycling Safety: Cyclists, acting as human sensors, report the situation in city streets through their smartphone apps (Figure 9). Benefit for the citizens: Cyclists benefit by sharing routes, road conditions (including traffic and infrastructure issues) and events in real time. **Benefit for city officials:** Real time reporting of potential problems in city infrastructure, recording of preferred cyclist routes for the implementation of measures to accommodate them.

(2) Augmented Reality in Points of Interest: Tourists use their smartphone apps (Figures 8 and 9) to receive information about points of interest in a city. Coordinates, sceneries, or QR codes are used for the identification of location in different cities. **Benefit for the citizens**: Visitors get or share useful information about the place they are visiting in their SN accounts. **Benefit for city officials**: Points of interest in a city can be promoted. **Benefit for local businesses**: Visitors can be diverged to specific areas thus strengthening the local market.

(3) Citizen journalism/Participatory Urbanism: Citizens report events of interest by posting images, texts and metadata through RADICAL's smartphone apps (Figure 8) in a city-dedicated page or in SNs. Others can then consume the information perhaps in a curated/structured way or alternatively crawl into Social Networks and detect events of interest in the city. **Benefit for the citizens**: Citizens get involved in public issues by reporting through their phones. **Benefit for city officials**: Events of interest to the city administration are reported in real time.

(4) Monitoring the carbon footprint of products, people and services: A range of sensors monitors the CO_2 emissions in specific places in a city. **Benefit for city officials**: Increasing awareness about the processes that generate CO_2 emissions and assist in the creation of a zero-emissions city policy. **Benefit for local businesses**: Taking measures for the reduction of their service's or product's carbon footprint, leading to a reduction of relevant taxes and promoting CO_2-free processes.

(5) Propagation of eco-consciousness: Leveraging on the viral effect in the propagation of information in social networks as well as the recycling policy of a city through monitoring and reporting relevant actions from eco-conscious citizens' smartphones. **Benefit for city officials**: Eco-conscious activities are monitored and city officials spend relevant resources in a more direct manner.

(6) Social-Oriented Urban Noise Decibel Measurement Application: This scenario aims at creating a socially-enabled and aware service which allows citizens to better monitor and report environmental noise in the streets. Noise sensors are employed for that purpose and citizens are able to report and comment noise-related issues through SNs under specific hashtags. **Benefit for city officials**: Having a clear sentiment-based map overview of how the noise is affecting the city.

(7) City Reporting application for the use of Urban Services: This service gathers sensory data along with SN check-ins in city venues to construct a traffic map throughout the city, leveraging the process load of any centralised decision making processes. **Benefit for city officials**: Local governments can listen to the mobility needs of citizens, combine it with traffic sensor data and advanced analytics to deliver smart transportation solutions and better infrastructures.

4.3 Evaluation Results

Citizen participation in the context of RADICAL pilots was evident throughout the cities, with the highest activity observed in more mature smart cities, like Santander and Genoa. Table 2 presents aggregated IoT data statistics by pilot and by city platform instances,

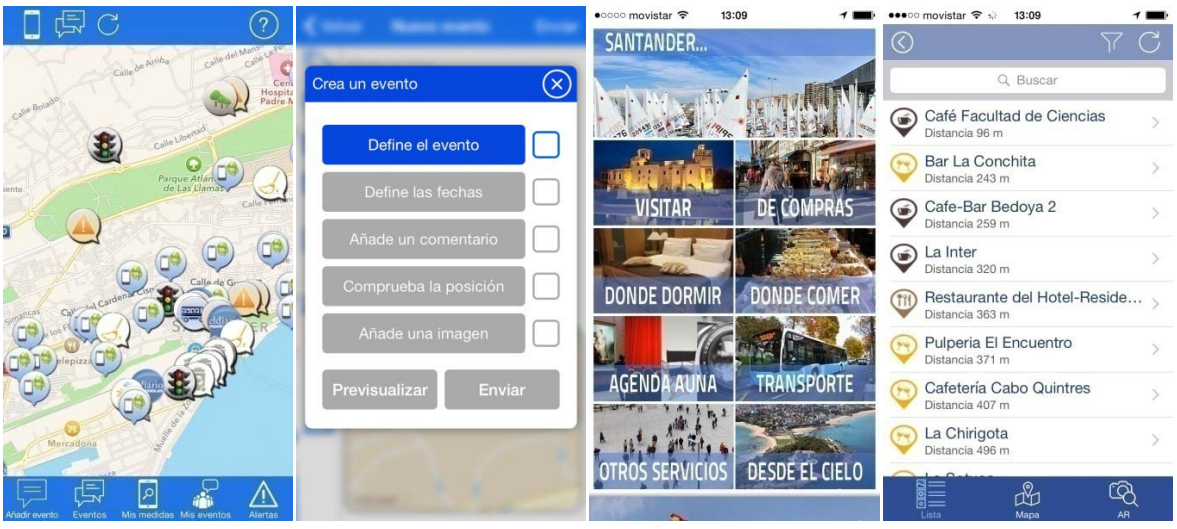

Figure 8. Participatory urbanism and augmented reality RADICAL apps in the city of Santander.

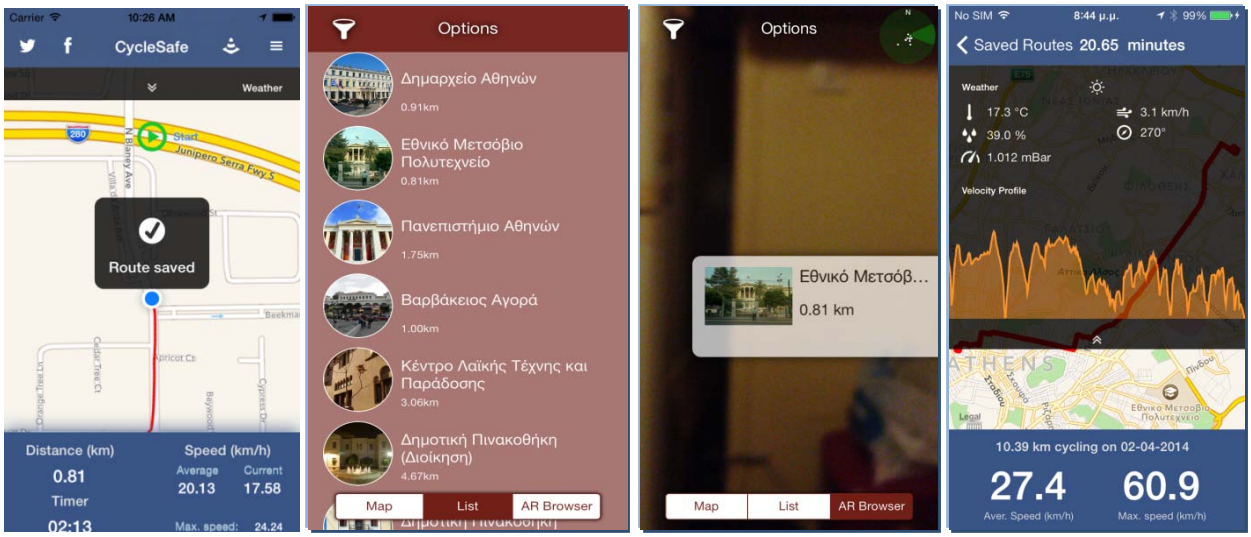

Figure 9. Augmented reality and cycling safety RADICAL apps in the city of Athens.

Table 2. Aggregated RADICAL usage statistics per city and pilot service

City/Services		Cycling safety improvement	Carbon footprint management	Citizen journalism	Participatory urbanism	Augmented reality	Propagation of eco-consciousness	Total
	# of IoT devices			4		2		6
Issy	# of Observations			750		10		760
	# of Measurements			8083		50		8133
	# of IoT devices				470	4270	185	4925
Santander	# of Observations				5011	56818	12	61841
	# of Measurements				16362	341842	72	358276
	# of IoT devices	45				2		47
Athens	# of Observations	139				162		301
	# of Measurements	139				162		301
	# of IoT devices		19				7	26
Genoa	# of Observations		3015				230074	233089
	# of Measurements		7545				1839928	1847473
	# of IoT devices		1			628		629
Cantabria	# of Observations		412258			7999		420257
	# of Measurements		3179598			55993		3235591
	# of IoT devices	3						3
Aarhus	# of Observations	12005						12005
	# of Measurements	12002						12002
	# of IoT devices	48	20	4	470	4902	192	5636
Total	# of Observations	12144	415273	750	5011	64989	230086	728253
	# of Measurements	12141	3187143	8083	16362	398047	1840000	5461776

extracted from the RADICAL Governance Toolkit's added value tool. Those results provided the big picture of citizens' engagement in the RADICAL pilot, showing overall IoT devices (smartphones or installed sensors) and data sent per service for each participating city.

Device-related data as dictated by the RADICAL object model are presented in the form of Observations or Measurements. Observations correspond to general IoT events, for example sensor reports or bi-

cycle "check-in" events, while Measurements cover more specific metrics included in an Observation like Ozone measurements (mpcc) or the average bicycle speed (km/h).

As can be observed, data handled by RADICAL's repository reach the range of millions. Thus the aim of the technical evaluation performed was to stress the RADICAL platform and tools with parallel calls for retrieving IoT and SN data. More specifically, the following tests were run to validate the platform's technical stability:

- Parallel calls for retrieving various IoT (devices, events, observations) large datasets (**results scale: ~10.000 results per dataset**)
- Parallel calls for retrieving IoT and SN large datasets (**results scale: ~100 results per dataset, 5 topics identified**)
- Parallel calls for retrieving large sets of measurements in many pages (**results scale: 10.000 results per page**)
- Parallel calls to various cities for retrieving IoT data (**results scale: ~2.000 results from 5 cities' platforms**)

RADICAL's APIs successfully replied parallel requests in a matter of less than 3 sec with resource utilisation being balanced. Server crashes did not appear, proving the platform's soundness in terms of technical integrity.

Regarding citizens' and city officials' platform evaluation, feedback was collected through online surveys, presenting user questionnaires in +Spaces[38] dedicated poll apps in Facebook and Twitter. Living Labs' participants in pilot cities were asked to evaluate RADICAL's city-provided application's usability and operation, and its added value in general.

City Service administrators and developers on the other hand were asked about the functionality and operation of RADICAL as well as the sustainability potential in the context of Smart City.

Overall, 337 participants (235 citizens and 102 city officials) coming from the 6 cities evaluated the various aspects concerning RADICAL. 12 Living Labs were established overall (2 for each city), thus 12 questionnaires were finally deployed to citizens, asking them to evaluate the impact RADICAL had in their everyday lives. The final validation also included 3 questionnaires for city officials (service admins and technicians) in order to evaluate the technical aspects of the platform (functionality, replicability, security and privacy), as well as its business potential (sustainability).

Feedback collected was promising, as presented in some indicative questions in Figures 10 and 11, extracted from +Spaces Data Analysis page. Full questionnaires and detailed numbers for RADICAL Living Labs can be found on the project website[39].

5. Future Work and Conclusion

Future work in the RADICAL platform is aligned to the progress of the RADICAL project and includes the operation of pilots for large groups of people participating in Living Labs. These people will validate the benefits of the core concepts and technical implementations. In particular, the main challenge is to prove that the combination of knowledge gathered and deployed in social networks combined with sensors and Internet of Things elements in smart cities contexts can provide a wide set of services that are beneficial

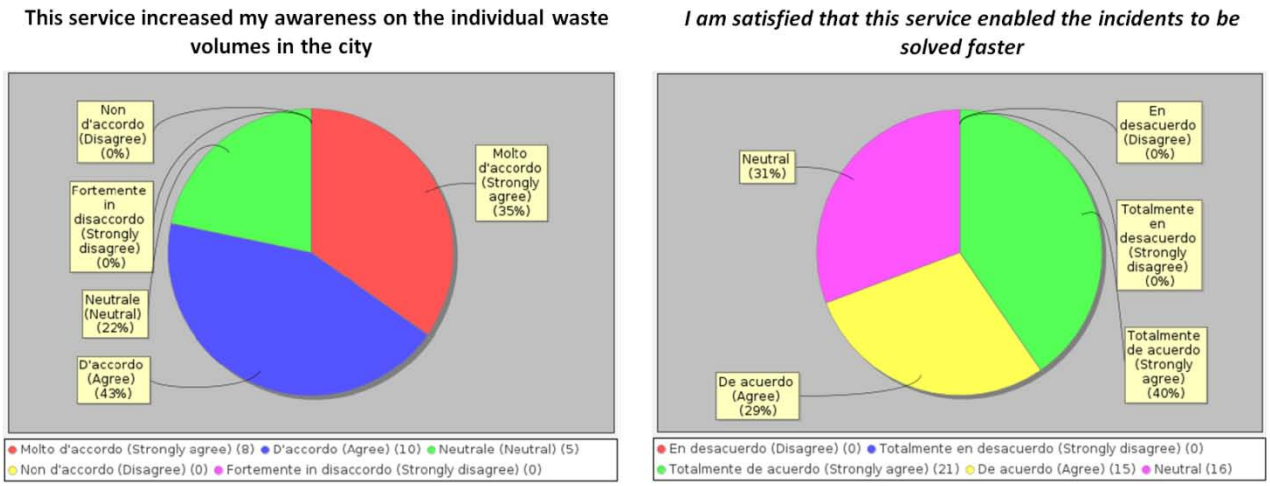

Figure 10. Indicative results of citizens' polls (Eco-consciousness service in Genoa/Participatory Urbanism service in Santander).

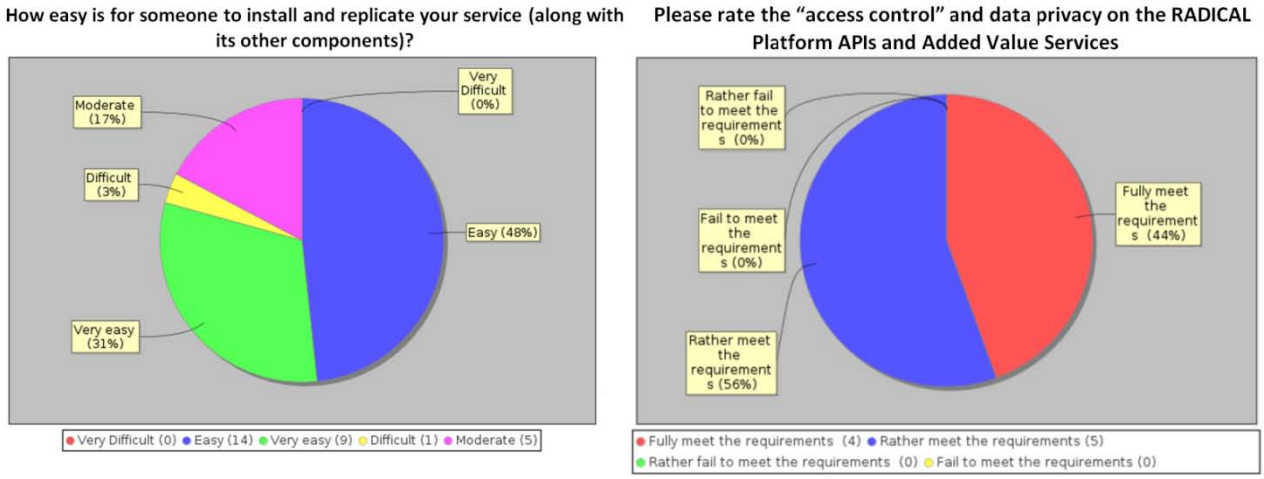

Figure 11. Indicative results of city officials' polls (Ease of RADICAL installation and replicability/Data privacy evaluation).

for the citizens and the urban planning of the public administration and policy makers.

The evaluation from city officials demonstrated that the described approach promotes sustainable integration of social networking and IoT services, enabling third parties (i.e., cities and service developers including SMEs) to successfully engage in the co-design and take-up of similar services. In relation to replicability, the proposed approach seems to be achieving its goals. Evaluation showed that almost 79% of city officials could replicate RADICAL's services with ease.

Finally, we showed that the RADICAL platform is a technically sound solution while the currently hosted services serve their purpose in raising awareness and providing access to government services.

The main task set as future work is the development of best practices that will not only cover technical and technological issues but will mainly emphasise on planning, financing, sustainability, operational and legal aspects. For example, as shown before, privacy and security are regarded as crucial factors in smart city platforms.

Along with these best practices, RADICAL will produce roadmaps associated with sustainable deployment and operation of social networking services in smart cities, illustrating the roles and responsibilities of all stakeholders towards sustainable development and operation. Based on the best practices and roadmaps, RADICAL will also elicit a number of guidelines that will be provided to policy makers in order to shape policies at the regional, national and EU levels, in a way that boosts the development and adoption of social networking and IoT services in urban environments.

Conflict of Interest and Funding

No conflict of interest was reported by all authors.

Acknowledgements

The authors would like to acknowledge the RADICAL consortium for their collaboration during the research project. This work was supported by the European Union's Competitiveness and Innovation Framework Programme under grant agreement no. 325138.

References

1. Romero-Lankao P, 2008, Urban areas and climate change: review of current issues and trends. *Issues Paper for the 2011 Global Report on Human Settlements*.
2. Conti M, Passarella A and Pezzoni F, 2011, A model for the generation of social network graphs. *Proceedings from the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011*, 1–6. http://dx.doi.org/10.1109/WoWMoM.2011.5986141.
3. Sundmaeker H, Guillemin P, Friess P, *et al*. (eds) 2010, *Vision and Challenges for Realising the Internet of Things*. Publications Office of the European Union, Luxembourg.
4. Perera C, Zaslavsky A, Christen P, *et al*. 2014, Sensing as a service model for smart cities supported by Internet of Things. *Transactions on Emerging Telecommunications Technologies Special Issue: Smart Cities — Trends & Technologies*, vol.25(1), 81–93. http://dx.doi.org/10.1002/ett.2704.
5. Calabrese F, Kloeckl K and Ratti C, 2007, WikiCity: real-time location-sensitive tools for the city. *IEEE Pervasive Computing*, July–September 2007.
6. Murty R, Gosain A, Tierney M, *et al*. 2007, *TR-13-07. CitySense: a vision for an urban-scale wireless networking testbed*, Harvard University Computer Science Group, viewed January 21, 2016,

<http://ftp.deas.harvard.edu/techreports/tr-13-07.pdf>

7. Page X and Kobsa A, 2010, Navigating the social terrain with Google Latitude. *Proceedings of iConference, Urbana-Champaign, IL, 2010*, 174–178.
8. Breslin J G and Decker S, 2007, The future of social networks on the internet: the need for semantics. *IEEE Internet Computing*, vol.11(6), 86–90. http://dx.doi.org/10.1109/MIC.2007.138.
9. Breslin J G, Decker S, Hauswirth M, *et al*. 2009, Integrating social networks and sensor networks. *Proceedings from the W3C Workshop on the Future of Social Networking, 15–16 January 2009, Barcelona*.
10. Miluzzo E, Lane N D, Eisenman S B, *et al*. 2007, CenceMe: injecting sensing presence into social networking applications. *Proceedings of the 2nd European Conference on Smart Sensing and Context (EuroSSC'07)*.
11. Hernández-Muñoz J M, Vercher J B, Muñoz L, *et al*. 2011, Smart cities at the forefront of the future internet, in *The Future Internet*. Springer-Verlag Berlin, Heidelberg, 447–462.
12. Sanchez L, 2010, SmartSantander: experimenting the future internet in the city of the future. *Proceedings from the 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2010), 26–29 September 2010, Istanbul, Turkey*.
13. *EUROCITIES project*, n.d., viewed December 7, 2015, <http://www.eurocities.eu/eurocities/projects>
14. *Open and Agile Smart Cities (OASC)*, n.d., viewed December 15, 2015, <http://oascities.org>
15. EIT Digital IVZW, 2010, *Urban life and mobility*, viewed December 19, 2015, <http://www.eitdigital.eu/innovation-entrepreneurship/urban-life-mobility>
16. Schaffers H, Komninos N and Pallot M, 2012, *FIREBALL White Paper: smart cities as innovation ecosystems sustained by the future internet*, viewed January 10, 2016, <http://www.urenio.org/wp-content/uploads/2012/04/2012-FIREBALL-White-Paper-Final.pdf>
17. European Network of Living Labs ENoLL, n.d., *Interdisciplinary Studies Journal — a special issue on smart cities*, viewed December 9, 2015, <http://www.openlivinglabs.eu/news/isj-special-issue-smart-cities>
18. *RADICAL: Rapid Deployment for Intelligent Cities and Living*, n.d., viewed January 17, 2016, <http://www.radical-project.eu>
19. Menychtas A, Kranas P, van der Graaf S, *et al*. 2013, EPIC: A holistic approach for smart city services. *Proceedings from the CMI's 6th Annual International Conference: Developing the Future ICT Infrastructure Technologies, Markets, and Policies, Aalborg, Denmark*.
20. Gubbi J, Buyya R, Marusic S, *et al*. 2013, Internet of Things (IoT): a vision, architectural elements, and future directions. *Future Generation Computer Systems*, vol.29(7): 1645–1660. http://dx.doi.org/10.1016/j.future.2013.01.010.
21. Zanella A, Bui N, Castellani A, *et al*. 2014, Internet of Things for smart cities. *IEEE Internet of Things Journal*, vol.1(1), 22–32. http://dx.doi.org/10.1109/JIOT.2014.2306328.
22. Paskaleva K, 2013, *Smart citizens in smart cities: implementation of SmartiP Evaluation Framework*, viewed September 1, 2015, <https://www.escholar.manchester.ac.uk/uk-ac-man-scw:213230>
23. *Citadel on the Move homepage*, n.d., viewed January 19, 2016, <http://www.citadelonthemove.eu>
24. *Peripheria — networked smart peripheral cities for sustainable lifestyles*, n.d., viewed January 15, 2016, <https://www.itas.kit.edu/english/iut_completed_pask10_peripheria.php>
25. *Open Cities project*, n.d., viewed January 17, 2016, <http://www.opencities.net>
26. *GrowSmarter project*, n.d., viewed January 18, 2016, <http://www.grow-smarter.eu/home>
27. *Triangulum project*, n.d., viewed January 20, 2016, <http://www.triangulum-project.eu>
28. *SocIoS: Exploiting Social Networks for Building the Future Internet of Services*, n.d., viewed January 18, 2016, <http://www.sociosproject.eu>
29. *SmartSantander, FP7EU funded research project*, n.d., viewed June 9, 2015, <http://www.smartsantander.eu>
30. *RADICAL project deliverable D2.3: architecture of the Open RADICAL Platform*, n.d., viewed January 11, 2016, <http://www.radical-project.eu/wp-content/uploads/RADICAL-D2.3-Architecture-of-the-Open-RADICAL-Platform.pdf>
31. Bertot J C, Jaeger P T and Hansen D, 2012, The impact of polices on government social media usage: issues, challenges, and recommendations. *Government Information Quarterly*, vol.29(1): 30–40.
32. Chin E, Felt A P, Sekar V, *et al*. 2012, Measuring user confidence in smartphone security and privacy. *Proceedings of the Eighth Symposium on Usable Privacy and Security (SOUPS '12), ACM, New York, USA*.
33. Kardara M, Kalogirou V, Papaoikonomou A, *et al*. 2014, SocIoS API: a data aggregator for accessing user generated content from online social networks, *Web Information System Engineering — WISE 2014 Workshops*, vol.9051: 93–104. http://dx.doi.org/10.1007/978-3-319-20370-6_8.
34. *WebHookIt open-source project*, n.d., viewed December 8, 2015, <http://neyric.github.io/webhookit/docs/>
35. Eriksson M, Niitamo V P and Kulkki S, 2005, *State-of-the-art in utilizing Living Labs approach to user-centric ICT innovation — a European approach*, Center for Distance-spanning Technology, Luleå University of Technology, Sweden, viewed January 4, 2016, <http://www.vinnova.se/upload/dokument/Verksamhet/TITA/Stateoftheart_LivingLabs_Eriksson2005.pdf>
36. *RADICAL deliverable D5.3: pilot operations plans*, n.d., viewed January 12, 2016, <www.radical-project.eu/wp-content/uploads/RADICAL-D5.3-Pilot-Operations-Plans_Final.pdf>
37. *BonFIRE*, 2014, viewed January 20, 2016, <http://www.bonfire-project.eu/home>
38. CORDIS Community Research and Development Information Service, 2016, *+Spaces: policy simulation in virtual spaces*, viewed January 21, 2016, <http://cordis.europa.eu/project/rcn/93838_en.html>
39. *RADICAL deliverable D7.2.1: end-users evaluation*, n.d., viewed January 22, 2016, <http://www.radical-project.eu/publications/public-deliverables/>

RESEARCH ARTICLE

Migration of applications to the Cloud: a user-driven approach

Anastasia Panori[1*], Agustín González-Quel[2], Miguel Tavares[3], Dimitris Simitopoulos[4] and Julián Arroyo[5]

[1] URENIO Research, Department of Urban and Regional Development and Planning, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

[2] Ariadna Servicios Informáticos, C/ Cronos 63, 4ª. 28037 Madrid, Spain

[3] Municipality of Agueda, Portugal

[4] Department of Operational Planning & ICT Systems, Municipality of Thessaloniki, Greece

[5] City of Valladolid, Spain

Abstract: During the last decade, there has been an increased interest on cloud computing and especially on the adoption of public cloud services. The process of developing cloud-based public services or migrating existing ones to the Cloud is considered to be of particular interest—as it may require the selection of the most suitable applications as well as their transformation to fit in the new cloud environment. This paper aims at presenting the main findings of a migration process regarding smart city applications to a cloud infrastructure. First, it summarises the methodology along with the main steps followed by the cities of Agueda (Portugal), Thessaloniki (Greece) and Valladolid (Spain) in order to implement this migration process within the framework of the STORM CLOUDS project. Furthermore, it illustrates some crucial results regarding monitoring and validation aspects during the empirical application that was conducted via these pilots. These findings should be received as a helpful experience for future efforts designed by cities or other organisations that are willing to move their applications to the Cloud.

Keywords: cloud migration, STORM CLOUDS, user-driven open innovation process, smart cities

*Correspondence to: Anastasia Panori, URENIO Research, Department of Urban and Regional Development and Planning, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece; Email: apanori@urenio.org

Received: February 10, 2016; **Accepted:** March 22, 2016; **Published Online:** May 9, 2016

Citation: Panori A, González-Quel A, Tavares M, *et al*. 2016, Migration of applications to the Cloud: a user-driven approach. *Journal of Smart Cities*, vol.2(1): 41–52. http://dx.doi.org/10.18063/JSC.2016.01.005.

1. Introduction

Cloud computing has emerged as a key enabling technology to support more efficient, cost effective and quick deployment of IT services. It allows optimising service provisioning, automatically and seamlessly adjusting resources (bandwidth, infrastructures, data, etc.) to real-time demands, resulting in the optimisation of IT related costs while at the same time constituting an easier and more flexible alternative for service provision to citizens[1,2].

The European Commission eGovernment Action plan, the Digital Agenda for Europe and the Cloud Computing Strategy have set the effective exploitation of the benefits of information and communication technologies (ICT) as one of their main goals. After having highlighted the innovative character of public services migrated to the Cloud, the use of cloud computing is being promoted for the creation of more agile, trusted and transparent administrative services.

Under this context, STORM CLOUDS project (www.stormclouds.eu) was born, partially funded by

the European Commission within the CIP-FP7 Program. The main objective of the project focuses on exploring the shift to a cloud-based paradigm for deploying services that public authorities currently provide, based on their internal ICT infrastructure. The project aims to define useful guidelines on how to implement the process of moving applications to the cloud and it is based on direct experimentation with pilot projects conducted in three cities participating in the consortium.

Following this short introduction, the rest of the paper is organised as follows: Section 2 provides a brief literature review concerning the cloudification and migration processes. For Section 3, we show how cloudification process starts with the selection of the applications that will be migrated to the cloud based on a user-driven open innovation methodology. Municipalities' personnel work along with the cities' stakeholders were included in order to accomplish this selection. Once suitable technical actions were taken, an effective deployment of the applications on the cloud infrastructure was carried out.

Moving on, experiences and findings regarding the empirical implementation of this process on three pilot projects are thoroughly described in Section 4. The main goal of this section is to evaluate the adequacy of implementing an open innovation methodology in a strongly technical process like this one. We firmly believe that this methodology is highly suitable, provided that technical issues are communicated in the best way, so that all involved stakeholders are able to understand the implications of the migration process. This means that although a final user may not be aware whether an application is hosted on the City Hall server or on a public cloud, he/she will be concerned about the application's implications on costs, flexibility or duration.

In general, information provided in this article is particularly valuable for municipalities and public administration bodies, aiming to implement a migration process of their applications to a cloud infrastructure. One of the main findings suggested that this action was received as a positive policy movement for most stakeholders. These findings, along with a small discussion on the process and overall conclusions, are presented in Section 5.

2. Public Services on the Cloud: Literature Review

During the last decade, a lot of efforts have been made in order to define and in some cases portray the art of cloud computing. Starting from characterisations such as "*phrase du jour*"[3], "*perfect market buzzword*"[4] and moving on to "*a buzzword almost designed to be vogue*"[5], it became clear that cloud computing could not be ignored as an innovative methodological approach. Back in 2009, *The Economist*[6] stated that "the rise of the Cloud is more than just another platform shift that gets geeks excited. It will undoubtedly transform the information technology (IT) industry, but it will also profoundly change the way people work and companies operate". Furthermore, the diversified nature of applications that are gradually shifting into the Cloud constituted a crucial parameter which acts as a catalyst regarding the constantly changing definition of cloud computing[7].

Throughout time, the advantages offered by this innovative approach to internet-based computing stated to seem quite appealing to government agencies which gradually began adopting it. In general, cloud infrastructure is able to provide public administrations with mechanisms for improving their capability to fulfil citizens' demands. More specifically, advantages such as high levels of communication and collaboration effectiveness[8] are considered to be particularly appealing characteristics, which act as an essential comparative advantage when it comes to public sector agencies. Meanwhile, initiatives aiming to promote aspects of "smart city" evolution are gradually starting to accelerate as their policy design tool character is strictly interconnected with broader concepts of development, sustainability and inclusion[9–11]. The design and development of applications targeting certain domains of smart city advancement, such as energy and transport, resulted in the production of a large number of services which in many cases need to be redefined in order for their effectiveness to be improved[12].

Alongside the existence of practical benefits of the adoption of cloud based services, the recent economic crisis played an important role to their rapid expansion. Condon[13], referring specifically to the case of the U.S., characterises the combinatorial interaction between economic recession and web-savvy policies obtained by the government as the "perfect storm" that cloud computing is riding through its attempt to penetrate the public sector. He also pointed out that if cloud computing can work well for businesses, it should also be appropriate for the public sector[14]. A similar point of view was also adopted by a large number of politicians and scholars during the same period. Schatz[15] quoted Aneesh Chopra, the first Chief

Technology Officer (CTO) for the US government, who tried to promote a more extensive adaptation of applications based on cloud computing techniques by the US government. Hoover[16] quoted another important US CTO, Vivek Kundra, who highlighted another very essential advantage of cloud-based applications. Specifically, he pointed out the extremely less expensive character of such kind of applications compared to the expensive investments in infrastructure that should have otherwise taken place.

Apart from the undoubtedly high interest expressed by the US government on the extended use of cloud-based services throughout its public sector, there seems to be a large number of other international examples following a similar policy. Starting from the UK, the Digital Britain Report announced in June 2009, promoted a wide-raging digital strategy for the country through the improvement and migration of more governmental applications on the cloud[17,18]. Japan also constituted another important case of a major cloud computing initiative through the development of the "Kasumigaseki Cloud"[19,20], an initiative which aims to create a private cloud environment that would eventually host all governmental applications. In this case, the environmentally friendly character of such an innovative IT policy was also highlighted[21]. Other international examples referring to cloud computing initiatives are Thailand, New Zealand, Vietnam, Singapore and China.

Regarding the EU policy towards the creation of a cloud of public services, there has been a lot of effort during the last few years. In September 2012, the European Commission adopted the European Cloud Computing Strategy [EC14]. The strategy was designed to speed up and increase the use of cloud computing across European economy, as the gain in jobs and overall economy indicators was clearly identified. In particular, public sector organisations have much to gain by taking a cloud computing approach for service delivery in their information and communications technology environments[22]. An important number of research projects (eEnviPer, InGeoCloudS, OASIS, OpenDAI and SEED) were funded during a first call by the EU in 2011 whose initial success triggered a second round of projects in 2013 (CLIPS, CloudOpting, ECIM, STORM CLOUDS, STRATEGIC and Virgo). All of them target on experimenting with the migration of public services to the cloud, demonstrating possible interoperability and validating common specifications of cloud-based services. This process resulted in specifying the anticipated impact with respect to the Cloud Computing Strategy implementation.

Given the complexity of the process and the rigidity of public organisations, governmental strategies targeting a more cloud-oriented public infrastructure should be accompanied with a plan regarding the selection and migration of the appropriate applications to the Cloud, in addition to the technical transformations needed to fit in the new environment. A number of studies have been conducted in order to determine strategies referring to the process of organising the cloudification procedure, each offering a new approach to the challenges entailed by this movement towards a 'cloud of public services'.

According to Wauters *et al.*[23], the migration process should be conceived as a segmented procedure focusing on different sub-domains, targeting on a simplified implementation of the cloudification process. They also highlighted the fact that already existing public services should be used as a base upon which new ones could be added during the migration process. In addition to that, Seo *et al.*[24] viewed the whole process as a combination of three main sub-procedures that include (a) the definition of standards for selecting the applications to be migrated, (b) technical guielines regarding the cloudification process and (c) a set of instructions concerning an economic feasibility analysis. Concerning the selection of applications to be migrated, Bonneau *et al.*[25] moved to the creation of three general categories of services encompassing procurement and marketplace apps, resource pooling and standalone applications, respectively. Moreover, pilot projects were conducted in 10 European countries covering these categories[26].

Finally, KPMG's technical report[27] on the cloudification of public services stated some key points aiming to offer a better understanding of how governmental agencies could optimise the costs and positive impacts by this migration process. These included issues such as addressing the Cloud as a potential "ecosystem" incorporating diversified participants, applying strong initiatives, managing potential risks and collaborating with providers and private sectors. In this paper, we focused on a user-driven approach for migrating public services to the Cloud, promoting broad participation of stakeholders during the whole migration process.

3. Migration Process Design

Cloud computing characteristics are particularly inter-

esting in supporting the provision of *Smart City* applications provided to citizens by governmental authorities but the migration process is not always a smooth procedure. Several issues may arise when public sector organisations consider transitioning to cloud computing. Some prominent concerns are related to assuring control of ICT systems by public managers and quality of service, ownership and liability issues, security and privacy, trust in reliability and resilience of infrastructures and services, interoperability and standards, potential dependencies with vendors, regulation, risk management, governance and culture.

In general, the migration process consists of four stages which are illustrated in Figure 1. These stages include actions regarding (i) the selection of applications/services to be migrated to the cloud, (ii) addressing technical or procedural challenges, (iii) migration to the pre-production cloud and (iv) moving applications to the production cloud.

During this procedure, the involvement of a number of stakeholders was essential not only as a methodological requirement but also due to the fact that stakeholders' involvement would produce helpful feedback for the overall process to be closer to citizens and public employees. Moreover, another reason why this mechanism was adopted by STORM CLOUDS as compared to a mere selection of applications by the authority's technical staff is the fact that this involvement would strengthen stakeholders' awareness regarding the efforts made by local authorities towards a modernisation of the public sector.

During the first stage, potential services to be cloudified or deployed were selected. In the second stage, problems regarding technical, procedural or financial issues arising during the migration of services to the cloud were specified and enumerated. Stage 3 constitutes a pre-production stage where pilot projects, demonstration and other activities take place under the scope of service validation whereas stage 4 refers to moving the selected applications to the production cloud. Alongside these four stages, active participation of the stakeholders was an integral part of this procedure, promoting the user-driven character of this methodology.

The process of application selection was based on three main criteria: political priorities, technical specifications/restrictions and user driven aspects. Regarding the stakeholders participating during the whole procedure, their profile was carefully selected in order to derive a high quality input for the process. Generally, it is essential to select adequate stakeholders and keep them engaged in the project, thus groups or individuals with any interest in having publicly available services operated from the cloud were mostly preferred. The initial list of stakeholders that was proposed included citizens, local SMEs, municipality personnel, technicians as well as financial, managerial and political representatives.

Although it was important to involve a satisfactory number of stakeholders, their motivation to participate and stay active throughout the whole procedure was even more essential. In order to cope with this challenge,

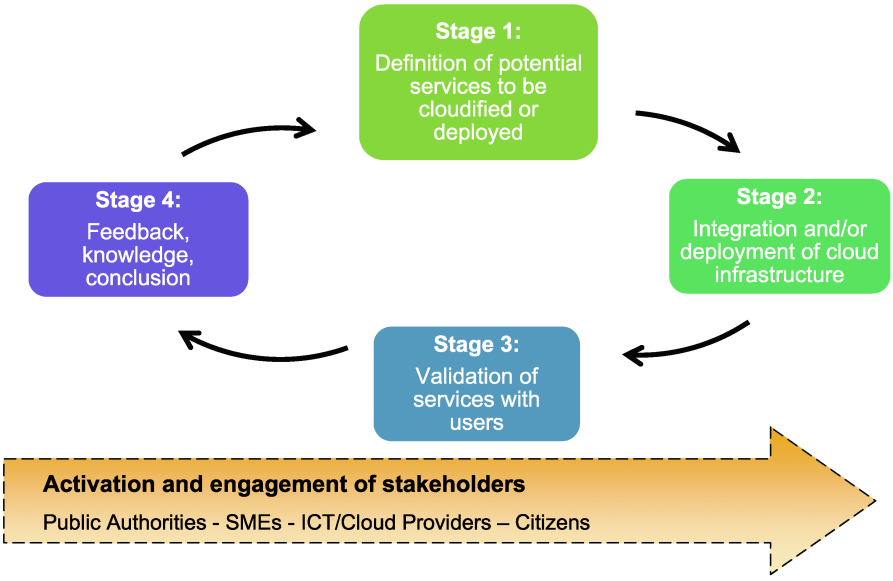

Figure 1. Four stages of a user-driven migration process.

some crucial elements were adopted while interacting with the stakeholders. These included first of all the selection of a meaningful number of people from each group and a detailed explanation of the project, as well as the proposed services. Furthermore, the use of non-technical language for a fluent communication to be achieved could give the end-users the opportunity to freely ask questions and be kept interested in the process. In addition, details such as providing enough time to fill in questionnaires and acknowledging the importance of end-users' opinion of the procedure's success played a crucial role for fulfilling this premise. Finally, in order to keep their engagement, it was important to provide stakeholders with short and mid-term rewards. In this sense, information about the progress of the whole process and the ways in which their involvement affected its formation were also given to them.

Another important methodological aspect of the project was the definition of a monitoring procedure in order to keep track of all activities related to user-driven open innovation methods. For this purpose, a mechanism consisting of three steps was endorsed. One of these steps included the identification of monitoring criteria and the definition of specific indicators, each depending on the task (stakeholder selection, application selection, etc.). Furthermore, the selection of information throughout the entire process of migration to the cloud, starting from the application selection until the adaptation of these applications to fit on the cloud, was an important part of the monitoring procedure for each one of the pilot cities. Additionally, an analysis concerning the extent of usage and acceptance of new applications and any possible variations on their usage patterns in the case of legacy applications was also considered to be essential for the monitoring process.

A summary of the process described above is given in Table 1. This table was completed by each one of the 3 pilot cities. In general, it constituted a practical roadmap for problem solving throughout the migration process of applications/services used in the public sector to a cloud infrastructure. These problems are usually recurring, thus the guide presented here might be perceived as an additional help to organisations facing future cloud migration processes in order to avoid loss of effectiveness through the deployment process. Nevertheless, the identification and suggestion of probable solutions referring to that kind of problems and situations were also the main objectives of the experiment.

4. Empirical Implementation

The above-described processes were followed in the cities of Agueda, Thessaloniki and Valladolid. Before examining each case separately, Table 2 provided a short description of the applications chosen to be migrated to the cloud during the STORM CLOUDS experiment.

4.1 Agueda

In the case of Agueda, the selection of services was guided initially by technical characteristics. These included criteria such as applications already owned by Agueda or open source applications for ensuring code availability. Moreover, the use of technologies known by existing Agueda technical staff was considered to be another crucial parameter for service selection.

The selected service for Agueda was *Eu Participo*. Several technical issues were identified including changes needed to be carried out in the application in order to reinforce its usability and change its logic to accommodate the requirements, preferences and expectations of a wider user reference group. Although

Table 1. Monitoring the migration process of public sector applications/services to cloud infrastructures

Stages	Details
STAGE 1: Application/ Service selection	How many different stakeholders participated in the service selection?
	How are stakeholders going to be activated and their participation maintained?
	How is the initial selection of services to be migrated to the cloud carried out?
STAGE 2: Technical adaptations	Problems found
STAGES 3 & 4: Cloud deployment	Analyse the technical experience and support provided by the cloud provider.
	Study how specific characteristics of the cloud could ease the process.
Feedback during the process	From stakeholders.
	From the Municipality's technical personnel.

Table 2. Applications chosen to be migrated to the cloud for each city

City	Name	Description
Agueda	Eu Participo	This WebGIS app allows citizens to express their opinion on a theme under discussion. Each topic is related to some geographic feature and users can upload photographs relevant to it. This application has an administrative interface to manage discussion topics. Privileged users (either from the local administration or community members) can add new themes for discussion.
Thessaloniki	Virtual City Mall (VCM)	Virtual City Marketplace enables the creation of a smart marketplace managed by the local shopping community. It empowers the city's local market by bringing together customers and merchants. Local shops are displayed over a city map, along with consumer reviews and promotional offers.
Valladolid	Urbanismo en Red (UeR)	Urbanismo en Red (UeR) was created with the purpose of publishing the municipal development plans across the Internet, enabling citizens to access them easily. It is designed to increase and enhance transparency in the public management of urban sectors. Moreover, it provides full interoperability between the various authorities and stakeholders.

the application was already available as an open source, its code files were transferred to Github and a new documentation was written in order to make it accessible to developers.

Regarding the selection of stakeholders, local authorities played an important role. The strong commitment shown by the City Mayor on technological progress in all aspects of the city was remarkable and extremely helpful during the whole process. This feature constituted a key parameter due to the fact that it facilitated the removal of a number of obstacles encountered, while at the same time it resulted in a diversified synthesis of municipality participants in the project.

Three main groups of stakeholders were defined—people working in the municipality, external stakeholders collaborating with the municipality and external users. These groups were defined using two main selection criteria; (a) the degree to which their decision affected the cloudification process (e.g., policy makers and politicians) and (b) the degree to which their everyday life was affected by this change (e.g., technical staff, accounting, procurement processes personnel). The widely diversified number of stakeholders selected for the city of Agueda resulted in the creation of a rich set of information that was used as input in the process of application/service selection. However, the difficulties on managing and consolidating the conclusions were received as a potential disadvantage of the wide range of stakeholders, whose activation and participation was stimulated by discussions, meetings and online questionnaires.

Most feedback received regarding *Eu Participo* was positive, despite the fact that citizens were not interested in knowing the technical details of the application. Generally, it was difficult to get feedback from users regarding the validation process as such technical issues were not clear to them. Regarding feedback coming from the municipality's technical personnel, these include the strong dependence on the municipality's internal IT units, arising from migrating public services to the cloud while a visible benefit was the possibility to deploy applications in a very fast and easy way.

It has not been possible to perform any technical or economic comparisons, as relevant comparable information was not available for the previously existing applications in Agueda. On the other hand, migration of an already existing application to the cloud was in practice a much faster and easier procedure than expected. Resources were ready to be used and all security threats were already handled by the cloud infrastructure. Besides that, there were no great benefits gained from this process since Agueda was using the cloud as an IaaS provider and the city already has an internal infrastructure of servers and virtual machines.

Two main indicators were defined for measuring the migration process: (1) users' acceptance degree which was measured by questionnaires filled in the meetings concerning the continuation of the user-driven open innovation process and (2) proportional change in the number of users. This latter aspect did not present any particular variation but the acceptance degree was particularly good as 89.23% of the stakeholders formed a positive/neutral view of the *Eu Participo* application (Table 3).

4.2 Thessaloniki

In the case of Thessaloniki, the whole migration process started with the user-driven dynamics for stakeholder/service selection. The municipality of Thessaloniki in the beginning organised general meetings with stakeholders and municipal services and later, training sessions and validation sessions with end users

Table 3. Monitoring results in Agueda – migration acceptance

Stakeholder group	Migration acceptance
Political representatives	Positive
External IT specialists related to the Municipality	Positive
Human resources staff	Positive
Administrative staff	Positive
Information Technology staff	Positive
Citizens	Neutral

as well as dissemination activities. During these meetings/events, the pilot partners distributed informative materials about the overall project and leaflets including the different candidates for migration. The municipality was particularly keen to select applications related to entrepreneurship and quality of life in the city. External stakeholders also selected applications for tourism promotion of the city.

During the second stage of the migration process, a number of problems were detected through feedback received and are explained in detail below. First, there were some organisational issues regarding the overall geographical coverage of the service, given that there was a strong interest from specific stakeholders to expand it to other areas of the wider metropolitan area of Thessaloniki and not only the city centre. These stakeholders were interested in sustaining the service after the end of the project. Another main issue referred to the entities that will be able to be present on the marketplace, the catalogue and the promotion pages. In general, it was decided that while professionals would be able to present themselves to the catalogue and be able to make offers, they could not be present on the marketplace which would be restricted only to shops and not service providers. Finally, a problem referring to the type of business model ensuring that the service would be sustainable after the end of the project also aroused. The most prevailing business model so far was one of multiple ownerships by different stakeholders managing different types of entities.

Regarding the monitoring process, a four dimensional group of indicators was established incorporating supply, demand, dissemination and level of validation (Table 4). Although four applications were selected to migrate to the cloud, only the Virtual City Mall was released early enough to complete the whole validation cycle, resulting in a complete dataset of monitoring indicator values.

Given that indicators were defined prior to the development of applications, no analytical record has been held beforehand which would give us a correct number of users providing feedback for the application (participants of the events and training sessions giving comments, individuals representing stakeholders and municipal employees proposing improvements, etc.). Furthermore, although a large number of stakeholders were involved in the whole process of services' selection, only a few of them were interested in providing feedback for this specific application—the Commercial Chamber of Thessaloniki, the Association of Professionals of Thessaloniki and the three departments of Thessaloniki Municipality, the Department of Entrepreneurship, the Department of Volunteerism and the Department of Tourism. Referring to the number of modifications based on the feedback received, these have been tracked after the first release of the application, when end users and professionals started using the application. In previous steps, feedback received and the respective modifications were somehow more informal and were not recorded in detail.

All applications that are either already or will be cloudified in the Thessaloniki pilot were new applications, thus any cost comparison would be inaccurate and could lead to erroneous conclusions. In general, a cost estimation task for an application's participation on the public cloud would require estimating separately the cost of buying hardware, software, operating systems and virtualisation software. In addition to that, the maintenance with personnel within the Municipality and with subcontractors, power consumption in case of hardware, telecom charges for Internet connection, system architecture and interoperability design should also be taken under consideration throughout the process. However, such a task would be infeasible and if tried it would provide very erroneous and questionable results.

4.3 Valladolid

In the case of Valladolid, four different applications were pre-selected by the internal staff involved in the project (Blue Parking, Ideal Innobarometer, LocalGIS, UeR) instead of presenting to stakeholders a wide range of candidate applications to be migrated to the cloud. In general, the field of applications/services was considered as a priority in the innovation strategy for the municipality. The criteria used to identify the list of applications included issues about citizens' demands and users' requests, as well as municipality strategic lines for developing both technical and public services.

Table 4. Indicators and monitoring results for Virtual City Mall, Thessaloniki

Supply	No. of shops participating in the app	62
	% of shops participating in the platform/shops in the area (total)	25.72
	No. of shops that have extended their online presence in the platform	27
	No. of shops making online transactions through the platform	0
	No. of offers per shop	0.24
	No. of synergies between two or more shops	0
Demand	No. of users – visitors (since 01.01.2015)	3893
	No. of registered users	41
	Sex	F: 45.85%
		M: 54.15%
	Total presence of the platform in third party websites (until 28.07.2015)	42
	No. of users providing feedback for the application	≈50
	No. of stakeholders providing feedback for the application	5
	No. of modifications based on the feedback received	3
Dissemination	Total presence of the platform in third party websites (until 28.07.15)	42
Validation	No. of users providing feedback for the application	≈50
	No. of stakeholders providing feedback for the application	5
	No. of modifications based on the feedback received	3

Project stakeholders were selected among two different groups, internal and external to the municipality. Also, the main criteria for the selection within each of these groups were diversified. In the first case, the main criterion was the extent to which migration of some applications to the cloud would affect their work. For instance, procedures followed by the procurement department were differentiated when contracting a cloud service, instead of buying hardware. Moreover, in the case of external stakeholders, the main goal was to achieve a diversified representation of groups of citizens living in the city—people from different age groups or professional skills, local SME, entrepreneurs, etc. This was complemented by choosing organisations that play a key role in the coordination of city actors.

After having defined the main selection criteria in each case, two groups were created. In the case of the internal group, municipal employees were mostly selected due to their good knowledge of each application and the fact that they could suggest improvements to be made, concerning the applications' functionality. The selected internal stakeholders were from the Department of Urban Planning, Traffic Management and Entrepreneurship. All staff working in some other key departments that usually provide support to the rest of the municipal units who thus have a comprehensive view of municipality needs and strengths was also selected. As a result, the additional internal stakeholders selected were from the Department of Information Technology, Accounting, Legal Services and Local Innovation Agency.

Regarding the external stakeholders' group, this was composed of citizens, small companies and associations clearly focusing on innovation. Using this criterion, members were selected from *Agile CyL*, a regional community of agile technologies developers; *ePunto*, a company devoted to fostering innovation in Castilla y León and linked to the local Chamber of Commerce; and *Geocyl*, a start-up in the field of geographic information systems consulting.

Stakeholders were contacted by email or phone, in order to get involved in the project while some regular short meetings were organised for applying the open innovation methodology. Additionally, as communication events were a strategic line, *Smart City* activities were presented in Valladolid and their participation to STORM CLOUDS project was evaluated as an important element.

During the service deployment, a number of problems have risen regarding the selected apps. Firstly, there were small delays in the planning of tasks and recruitment of external experts to cope with the cloud application installation. Another issue was ownership. The municipality was not the owner of the selected applications, as they were the property of the Spanish

Ministry of Industry and thus any implementation of additional features required a declaration of conformity from the Ministry. On a positive side, stakeholders were pleased by the current applications' functionality and modifications were not necessary.

A number of essential conclusions were recorded from the whole migration process. Initially, feedback coming from the final users of the cloudified version of *Urbanismo en Red* showed that the application's performance is similar to the existing one and the supplied information is exactly the same. Thus, it is absolutely transparent for users to use either the local or the cloudified version of the application. Moreover, there was no change in the number of users after the cloud-migration process. However, municipality technical staff mentioned that using a cloudified version of the application gave them more flexibility in the provision of services. Based on this approach, a temporal upgrade was feasible, cheap and easy to use during limited periods of time like urban plans submitted to public debate for a time. Furthermore, after the presentation and a testing period by specialised staff, positive reviews were received concerning the application.

Technical effort regarding system's maintenance has proven to be easier in the cloudified version, as most of the tasks being routine jobs, the staff can be devoted to more creative work. The relative ease of getting the service recovered from a crash by just restoring the instance in a short time and reducing downtime period is also remarkable for the technical staff. In addition, provision of telecommunications service is another major advantage, leading to a decreased number of new contracts with partners for using their services only for a few weeks.

At the political side, migration to the cloud was also perceived as a positive trend as it allows buying the application provision as a service which seems to be more feasible for the municipality than buying the required hardware and the service for its maintenance. Some feedback was also received by the municipality managers which showed that 80% of the interviewed managers characterised this application as very useful for the deployment of municipal policies and for enhancing transparency on urban projects. On the other hand, 20% of them expressed a sceptic attitude towards the impact of cloudification on citizens' lives. These conclusions are summarised in Table 5.

Table 5. Monitoring results for the city of Valladolid

Indicators	Before and after cloudification
Performance	No variations
Information supplied	No variations
Flexibility in service providing	Improvement
Improvement possibility	Improvement
Technical effort	Lower
Political management	Positive trend

5. Discussion and Conclusion

Table 6 summarises the main findings of *STORM CLOUDS* experiment regarding the four main steps of migration process followed here. In this integrated table, a systematic description of each city's results is attempted, corresponding to the categorisation proposed in Table 1. In all three cases, both similarities and disparities existed regarding the selection of stakeholders/applications, the encountered technical/procedural challenges and the monitoring indicators. Moreover, during the whole process a variety of methods were applied for activating and engaging stakeholders.

Starting from the selection of services/applications to be migrated to the cloud, a list of parameters affecting the final decision in each case was derived. As illustrated in Table 6, these components included various aspects such as technical considerations, political priorities and user-driven decisions. Although in the case of Agueda, technical restrictions were considered as the main driving forces for applications' selection; in the other two cases political priorities and user-driven criteria were used as the main selection incentives.

The selection of stakeholders is another crucial issue concerning the degree of effectiveness of this user-driven open innovation methodology. It was important to understand that the selection process should take into account not only the technical staff of the municipality but also a broader group of stakeholders whose everyday life might probably be affected by the migration process. A common pattern seemed to underpin all pilot projects, revealing the three main groups of stakeholders—an internal group of municipality departments, an external group of experts including academia, IT companies, commercial associations and a group of external users (citizens, local businesses, SMEs, etc.). In general, promoting the involvement of a wide range of end-users during the design process helps to better understand their needs. In this sense, involving external organisations and citizens was a wise decision leading to excellent results.

Table 6. Monitoring results for pilot projects

Monitoring Criteria			Agueda	Thessaloniki	Valladolid
STAGE 1 Stakeholder/ Service selection	*Selection of services*		Technical considerations: - Application owned by Agueda or open source applications - The code should be available - Use of a technology mastered by existing Agueda technical staff - Citizen personal data would not be managed	Political priorities: Thessaloniki Municipality decided to cloudify applications that are related to entrepreneurship and quality of life in Thessaloniki	Political & User-driven priorities: - Four different applications were selected by the internal staff of Valladolid municipality - City's most relevant problems were identified - Citizens' demands and users' request
	Selection of stakeholders	Internal members	Mayor and municipality dept.	Municipality and Dept. of Crowdfunding Volunteerism	Municipality departments
		External group	Academia, cities, IT companies	Commercial Association of Thessaloniki, Professional Chamber of Thessaloniki	Companies and associations with a clear focus on innovation: Agile CyL, ePunto, Geocyl
		External users	Citizens, professionals, SMEs	Local businesses	Citizens with different profiles
	Activation/ Participation		Direct contact	Meetings with stakeholders and municipal services	Direct contact (email or phone)
			Project brochure and STORM website	Training sessions	Regular short meetings
			Online questionnaires	Collection of comments and improvement suggestions	Small incentives to stakeholders (theatre tickets, invitations to cultural events)
			Discuss with stakeholders the name of the application		
STAGE 2 Technical adaptations	*Problems found* (Technical issues)		Change app logic to accommodate the requirements, preferences and expectations of a wider user community of international citizens from different municipalities Move code to Github and write new documentation	Thessaloniki Municipality servers are mostly running Windows instead of Linux Most are three- or more-tiered applications and are much heavier None of the existing applications is using Mysql In most cases, the applications don't use a web server	An unsuitable manual for the application installation procedure was only available The municipality wasn't the owner of the selected application
STAGES 3–4 Cloud deployment	*Study how specific characteristics of the cloud could ease the process*		Migration a much faster and easier procedure than expected All security threats were handled by the cloud infrastructure	A cost estimation task for the migration of an app to the public cloud would provide very erroneous and questionable results	Simple and routine jobs Easier to recover from crash Easier to telecommunications service providers relation
Feedback during the process	*From final users/ citizens*		Difficult to get feedback from final users about validating the services based on cloud computing as this issue was not clear in their minds	Expansion of the overall geographical coverage of the service, entities that will be able to be present to the marketplace, the catalogue and the promotions page	It is absolutely transparent using either the physical or cloudified version of the application
	From municipality technical personnel		Internal organisation made each unit highly dependent on the internal IT unit, regarding anything related with IT	The type of business model that will ensure that the service is sustainable after the end of the project	Using a cloudified version of the applications gave more flexibility in providing service
			The most visible benefit was the possibility to deploy an application in a very fast and easy way, independently of the municipality's internal IT department	The type of business model that will ensure that the service is sustainable after the end of the project	Technical effort for systems maintenance was lower with the cloudified version as most of the tasks that were routine jobs were eliminated and the staff can devote themselves to more creative work

Technical and procedural challenges addressed in this process also indicated an interesting variation throughout the pilot projects. It is particularly important to have a detailed technical plan in order to ensure that all required elements will be available prior to the migration process. This plan should incorporate aspects referring to (a) source code and documentation, (b) availability of technical support, either internal or external, (c) similarities/differences between the existing IT environment and the cloud environment and

ways to cope with any probable incompatibilities and (d) coordination of these activities with specific technical partners that could provide the necessary help and support to successfully complete the cloudification process.

Design and definition of monitoring indicators for assessing the output of the migration process constituted some additional key points of the proposed procedure. The choice of baseline indicators for evaluating public services that have been migrated to the cloud should cover multiple aspects, regarding the advantages of a cloud-based logic. These aspects include financial impacts, in addition to maintenance and security issues. Monitoring information targets a continuous improvement of cloudified services through feedback processes, as well as the detection of underlying multiplier effects, derived by a wide use of similar applications.

In terms of methodology, we firmly believed that the adoption of a user-driven open innovation approach under the framework of the STORM CLOUDS project was completely appropriate, providing satisfactory results. Stakeholders' involvement throughout the whole cloudification process was an essential characteristic of the proposed methodology. Nonetheless, their participation required an additional set of actions in order to maintain and reinforce their engagement during all stages of migration process.

Depending on the nature of the stakeholder's group, these actions should of course be diversified in terms of communication. For example, public servants ought to be informed about the ways in which cloudification of public services would benefit their group, in combination with appropriate training to adapt their competencies and abilities to withstand the new work conditions. On the other hand, a particularly effective tool for increasing citizens' involvement and encouraging them to stay engaged in the process is by including gifts and benefits related to the municipality. Mixing different types of stakeholders in common presentations and information sessions enriched the dynamic character of the process and increased the interest of stakeholders themselves.

These results should be received as a helpful experience for cities or other organisations that are willing to promote a strategy for migrating public services to the Cloud. In a user-driven approach, the selection of services/applications to be cloudified should take into account the opinion of a wide range of users. It is important to understand that when all users become an active part of the technological evolution of their city, they do realize that their city is essentially interested in covering their needs in the best possible way. This condition constitutes a particularly effective tool for increasing stakeholders' interest, by making them feel more deeply involved in the management of their city. After all, involving different types of users in the management process of a city creates positive interactions between them, resulting not only in higher quality public services but also in a more efficient way of management overall.

Conflict of Interest and Funding

No conflict of interest was reported by the authors.

Acknowledgements

This paper was based on experiments conducted in the framework of STORM CLOUDS, a project co-funded by the CIP-ICT-PSP program of the European Commission.

References

1. Marston S, Li Z, Bandyopadhyay S, *et al*. 2011, Cloud computing—The business perspective. *Decision Support Systems*, vol.51(1), 176–189. http://dx.doi.org/10.1016/j.dss.2010.12.006.
2. Buyya R, Yeo C S, Venugopal S, *et al*. 2009, Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, vol.25(6), 599– 616. http://dx.doi.org/10.1016/j.future.2008.12.001.
3. Knorr E and Gruman G, 2008, *What cloud computing really means*, IDG News Service, viewed February 1, 2016, <http://www.nytimes.com/idg/IDG_002570DE00740E180025742400363509.html?ex=1365393600&en=9076c93ed5911518&ei=5088&partner=rssnyt&emc=rss>
4. Wayner P, 2008, *Cloud versus cloud: A guided tour of Amazon, Google, AppNexus, and GoGrid*, InfoWorld, viewed January 25, 2016, <http://www.infoworld.com/article/2652709/paas/cloud-versus-cloud--a-guided-tour-of-amazon--google--appnexus--and-gogrid.html>
5. Weiss A, 2007, Computing in the clouds. *Computing*, vol.16.
6. The Economist, 2008, *Let it rise*, viewed January 26, 2016, <http://www.economist.com/node/12411882>
7. Hartig K, 2009, *What is cloud computing? The cloud is a virtualization of resources that maintains and manages itself*, Cloud Computing Journal, viewed January 23, 2016, <http://cloudcomputing.sys-con.com/node/579826>

8. Beizer D, 2008, *Cloud computing comes into focus*, GCN, viewed January 20, 2016, <https://gcn.com/articles/2008/06/11/cloud-computing-comes-into-focus.aspx>
9. Komninos N, Tsarchopoulos P and Kakderi C, 2014, New services design for smart cities: a planning roadmap for user-driven innovation: *Proceedings of the 2014 ACM International Workshop on Wireless and Mobile Technologies for Smart Cities (WiMobCity '14)*: 29–39. http://dx.doi.org/10.1145/2633661.2633664.
10. Schaffers H, Komninos N and Pallot M, 2012, *FIREBALL White Paper: smart cities as innovation ecosystems sustained by the future internet*, viewed January 10, 2016, <http://www.urenio.org/wp-content/uploads/2012/04/2012-FIREBALL-White-Paper-Final.pdf>
11. Komninos N, 2014, *The age of intelligent cities*, Routledge, New York.
12. Komninos N, Bratsas C, Kakderi C, *et al*. 2015, Smart City Ontologies: Improving the effectiveness of smart city applications. *Journal of Smart Cities*, vol.1(1): 31–46. http://dx.doi.org/10.18063/JSC.2015.01.001.
13. Condon S, 2009, *Is Washington ready for cloud computing*?, CNet News, viewed February 3, 2016, <http://www.cnet.com/news/is-washington-ready-for-cloud-computing/>
14. Condon S, 2009, *Cloud computing comes to D.C. in .gov 'war game'*, CNet News, viewed February 3, 2016, <http://www.cnet.com/news/cloud-computing-comes-to-d-c-in-gov-war-game/>
15. Schatz A, 2009, *Obama CTO addresses cloud computing, cybersecurity*, The Wall Street Journal, viewed December 8, 2015, <http://blogs.wsj.com/digits/2009/05/21/obama-cto-addresses-cloud-computing-cybersecurity/>
16. Hoover J N, 2009, F*ederal CIO scrutinizes spending and eyes cloud computing*, Information Week, viewed November 1, 2015, <http://www.informationweek.com/architecture/federal-cio-scrutinizes-spending-and-eyes-cloud-computing/d/d-id/1077579?>
17. Glick B, 2009, *Digital Britain commits government to cloud computing*, Computing, viewed November 18, 2015, <http://www.computing.co.uk/ctg/news/1816113/digital-britain-commits-government-cloud-computing>
18. Department for Culture, Media and Sport and Department for Business, Innovation and Skills, 2009, *Digital Britain: final report*, viewed January 5, 2016, <https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/228844/7650.pdf>
19. Hicks R, 2009, The future of government in the cloud, *FutureGov*, vol.6(3): 58–62.
20. Ng K, 2009, Japan gov't plots private cloud, *FutureGov*.
21. Rosenberg D, 2009, *Supercloud looms for Japanese government*, CNet News, viewed December 1, 2015, <http://www.cnet.com/news/supercloud-looms-for-japanese-government/>
22. European Commission, 2013, *Towards a cloud of public services*, viewed November 8, 2015.
23. Wauters P, Declercq K, van der Peijl S, *et al*. 2011, *Study on cloud and service oriented architectures for egovernment: final report*, European Commission, viewed October 5, 2015, <https://ec.europa.eu/digital-single-market/en/news/study-cloud-and-service-oriented-architectures-e-government-smart-2010-0074-%E2%80%93-final-report>
24. Seo J-H, Min J-S and Lee H, 2014, Implementation strategy for a public service based on cloud computing at the government. *International Journal of Software Engineering and Its Applications*, vol.8(9): 207–220. http://dx.doi.org/10.14257/ijseia.2014.8.9.17.
25. Bonneau V, Mahieu B, Dudenbostel T, *et al*. 2013, *Analysis of cloud best practices and pilots for the public sector*, European Commission, 2013.
26. Bonneau V, Mahieu B, Dudenbostel T, *et al*. 2013, *Analysis of cloud best practices and pilots for the public sector* — city profiles, European Commission, 2013.
27. KPMG, 2012, *Exploring the Cloud: a global study of governments' adoption of Cloud*, viewed November 28, 2015, <https://www.kpmg.com/Global/en/IssuesAndInsights/ArticlesPublications/Documents/exploring-cloud.pdf>
28. Chesbrough H, Vanhaverbeke V and West J, 2006, *Open Innovation: Researching a New Paradigm*, Oxford University Press, Oxford.

RESEARCH ARTICLE

Cloud-based architectures for geo-located blogosphere dynamics detection

Athena Vakali*, Stefanos Antaris and Maria Giatsoglou

Department of Informatics, Aristotle University, 54124 Thessaloniki, Greece

Abstract: Social networking data threads emerge rapidly and such crowd-driven big data streams are valuable for detecting trends and opinions. For such analytics, conventional data mining approaches are challenged by both high-dimensionality and scalability concerns. Here, we leverage on the Cloud4Trends framework for collecting and analyzing geo-located microblogging content, partitioned into clusters under cloud-based infrastructures. Different cloud architectures are proposed to offer flexible solutions for geo-located data analytics with emphasis on incremental trend analysis. The proposed architectures are largely based on a set of service modules which facilitate the deployment of the experimentation on cloud infrastructures. Several experimentation remarks are highlighted to showcase the requirements and testing capabilities of different cloud computing settings.

Keywords: social networks and wisdom of the crowd, geo-located blogosphere dynamics, social geo-located data clustering, cloud service deployment.

*Correspondence to: Athena Vakali, Department of Informatics, Aristotle University, 54124 Thessaloniki, Greece; Email: avakali@csd.auth.gr

Received: February 28, 2016; **Accepted:** March 30, 2016; **Published Online:** May 10, 2016

Citation: Vakali A, Antaris S and Giatsoglou M, 2016, Cloud-based architectures for geo-located blogosphere dynamics detection. *Journal of Smart Cities*, vol.2(1): 53–65. http://dx.doi.org/10.18063/JSC.2016.01.006.

1. Introduction

Huge data threads are produced in social media constantly, with microblogging and blogging frameworks dominating people's trend to broadcast information in a real-time fashion. In Twitter for example, tweet threads reach massive sizes, currently in the scale of 500 million per day and around 200 billion tweets per year[1]. The dynamic and unstructured nature of such information broadcasting offers an abundance of data out of which unexpected latent information can be harvested. Moreover, the current practice of declaring geo-location offers new sources of metadata (such as time, point of interest, geo-coordinates, etc.) which can reveal important trends in a geo-bounded area such as in a city.

User-originating information posted on social media typically reflect topics of their actual interest, thus such crowd-sourced and in particular geo-tagged content is particularly useful for geo-located trends' detection. Detecting trends of an area (such as in a city) is of major importance due to the fact that trends can be utilized to spot collective emergent or evolving behavior and phenomena—useful for proceeding to appropriate decision and city policy making.

Raw information from Twitter has been exploited in research at several domains such as for predicting revenues and stock prices, real-time identification of phenomena, political standings, etc. Therefore, it is well acknowledged that the microblogging "sphere" forms a valuable source of latent information relevant to the dynamics involved in public opinions and views. This is further justified by the fact that such applications capture the dynamics and the co-evolution social pulse[2]. Blogosphere as well is a rich information source at which the dynamics and the "voice of the public" may be extracted and mined especially with respect to certain locations or events. Therefore, mi-

croblogging and blogging activities can serve as major social dynamics barometers. This is due to the fact that such parallel information flows embed valuable and often hidden information about trending users' interests and opinions in a geo-located area.

This paper places emphasis on both the design framework and on the leveraging of the cloud paradigm to stress-test data analytics for localized trending topics detection. Trends dynamics are harvested and may be analyzed over user-contributed content from both microblogging (Twitter) and blogosphere activities through an approach such as in incremental text clustering. To support such an approach, dealing with high processing and data management demands; different cloud-based architectures have been designed and proposed for various experimentation scales and requirements.

The proposed framework's contribution is summarized in its following objectives:

- *Dealing with large scale data production in Web 2.0 micro-blogsphere* (with huge and rapidly evolving data) by enabling methods for efficient implementation, useful for real world application settings
- *Supporting the analysis of text data from emergent web sources* which may be generated at various rates under a unified data processing cloud-leveraging manner
- *Proposing a methodology for unsupervised detection of local trends* by combining content from different sources to enrich detected geo-located related trends
- *Designing and experimenting with different Cloud-based infrastructures* to support geo-located social data processing scenarios under parallelized computational settings

The rest of the paper is structured as follows: Section 2 reviews the current state regarding trend detection approaches that leverage microblogging and blogging data, discussing their challenges; Section 3 outlines the idea for leveraging a cloud-based trends detection framework, under which potential, with its implementation details summarized in Section 4. Different cloud based architectures are proposed in Section 5 to enable different architecture focus ranging from lightweight to fully parallelized solutions. In Section 6, some indicative experimentation results are outlined in order to highlight cloud-based solutions' capabilities which can be exploited for different social media data threads. Finally, conclusions and indications for the proposed work's exploitation in real city-bounded use cases are also outlined in Section 6.

2. Micro-blogosphere Trends Detection: Status and Challenges

Localized trend detection and "public's pulse" monitoring strongly set the need for efficient scalable and/or summarizing methodologies and frameworks. Current data mining (such as clustering) approaches focus on detecting (e.g., in Twitter): (i) clusters of users densely associated via follower or message links, or (ii) groups of tweets using text mining techniques such as exploiting common word co-occurrences[3].

2.1 Mining for Trend Analysis

A typical approach to trend analysis involves tracking users' interests in different keywords across time. At present, temporal trend analysis based on keyword frequencies has appeared in several commercial blogs and Web search engines such as Google Hot Trends[4] and BlogPulse. Although Google Hot Trends analyzes millions of web searches to identify trends, it does not emphasize social data analytics. This overlooks a collective source of intelligence which embeds opinions, facts and sentiments. BlogPulse is an online service that discovers trends from blogs on a daily basis with statistical techniques for detecting trending phrases based on their frequency of appearance[5]. Other online services such as the Twitter-related Trendistic[6] outline term frequency trends, again under statistical methodologies. Twitter itself also exhibits local trends[7] (for some locations) as keywords which are popular at the current time and at a particular city.

Clustering has been widely applied on content generated in Web social media to uncover latent associations, while only recently the feature of time and the temporal evolution of clusters have been researched[8,9]. Social data hierarchical graph clustering approaches have been applied for trend detection, in cases where associations among cross-blogs are modeled with a graph structure[9,10]. This approach operates on a static dataset, as it is not tailored for real-time online operation.

In TwitterStand, news detection from tweets is based on data from Twitter's GardenHose service (with a sample of Twitter's public timeline)[11]. To deal with noise, TwitterStand also filters out tweets that are unrelated to news via a classification method based on the Naïve Bayes Classifier. After that, the tweets are

clustered with an online method that holds many similarities to the one followed in Cloud4Trends application[3]. In particular, TwitterStand's algorithm extracts TF-IDF feature vectors for the tweets and clusters and performs clustering based on their similarity, while also incorporating the temporal dimension in the clustering process in the same way as Cloud4Trends does.

TwitterMonitor[11] is another framework for online trend detection over Twitter, following an approach similar to BlogScope[10].

2.2 Challenges in Crowdsourced Trend Analysis

Several solutions have been proposed to surpass the challenges imposed on social text clustering algorithms. These challenges are largely due to the inherent main social data characteristics (summarized in Table 1):

Table 1. Online social data processing challenges

Data Characteristics	Challenges
Vast size: Huge amounts of textual content, e.g., posts, tweets, comments, etc., produced on social media are intrinsic characteristics for social data analysis	**Scalability:** A scalable clustering methodology is suitable to process the vast amount of social text data; social media mining requires features such as geo-location and time
Noisy Data: Social text data are mostly written in informal style and have simple phrases, abbreviations, etc.	**Data Preprocessing:** An efficient text preprocessing stage which identifies that the noisy data is of great value for knowledge extraction
Dynamic data: Social media users produce new textual data at unexpected rates of time	**Online Processing:** A real-time, adaptive text clustering approach is needed to process highly evolving social data
Social data are geo- and time-dependent: Social users generate textual content of similar topics at a specific time period, and at different locations	**Streaming Clustering:** Social text data processed online must meet memory and performance requirements in cases of streaming clustering algorithms.

- **Vast social data sizes:** Demand scalable solutions, dealing with computational time complexities required by conventional text mining algorithms. Emerging clustering approaches should be considered to result in efficient social text data analysis, since data need to be processed at a limited amount of time[12,13]. Current parallel and distributed infrastructures are proposed to meet the scaling demands of the clustering algorithms, and already several parallel or distributed clustering approaches have been proposed reducing both the computational cost and the execution time[12,14,15].
- **Noisy social networks data:** Pose the need for methodologies dealing with the multiple noise states, due to the non-formal and unstructured social networks expression. Several text preprocessing methodologies (e.g., emoticon identification, acronyms recognition, etc.) are proposed as vital for the refinement of the text content and the improvement of the clustering approach[3].
- **Dynamic data threads:** Demand fast and often real-time processing and monitoring therefore new adaptive methodologies should be considered and validated[9].
- **Social data geo and time-dependencies:** Require multiple features integration since both geo-location and time are crucial in several location based social networks analytics[16].

3. Leveraging Cloud4trends for Social Text Dynamics Detection

This work places emphasis on dynamics detection in a geo-located and time related context. It leverages on Cloud4Trends, a framework proposed by the authors to enable online identification of trends dynamics, using Twitter and the Blogosphere[3].

It is important to notice that some commercially available products have focused on offering trend analytics solutions, with two examples shown here[17,18]. These tools place emphasis on attracting social customers by unifying engagement, visual planning, and collective collaboration. Their focus is on performance analytics for real-time campaign decisions and they differ with Cloud4Trends in terms of their focus on customer interactions and not on the latent knowledge extraction. By using Cloud4Trends, text clustering is employed in an incremental manner for detecting and maintaining a set of dynamic clusters. This framework is based on the assumption that the analysis is implemented on a "document" level, instead of a "term" level, whereas the corresponding clustering approach follows the TwitterStand process[11]. It is important to note that with Cloud4Trends clusters which are active at a given time and locations express the so-called active topics which are of users' interest. By dynamically observing the clusters' updating rates, we identify trends at their peak and detect the topics that are no longer trending. This is followed instead of applying a fixed-threshold based method that sets inactive clusters after a predefined period of time. In our approach, we separately collect and cluster tweets that pertain to a desired geographical area, rather than

examining the geographical scope of the resulting clusters as a post-analysis process.

In our proposed process, we proceed to a microblogging analysis performed on a streaming fashion to capture constantly changing trending users' interests, with an analysis which further:

- exploits associations based on the broadcasting **time**, alleviating gaps in earlier efforts[2], which employs a clustering method after identifying a set of trending phrases and focuses only on the latter, in an offline fashion
- deals with the respective user's physical **location** (exploiting the tweet's geo-location feature)

Such mutual multi-feature analysis is expected to produce more fine-grained high-quality clusters of tweets which will correspond to actual topics that are popular at a given location and time period. It is also expected to alleviate the generally acknowledged problem of noisy microblogging data, since the joint consideration of location and time generally improves the clustering quality and contributes to filtering out noisy tweets.

The proposed process is outlined in Figure 1 and it actually involves a 3-tier design that deals with the (i) collection of data in a streaming manner from Twitter as well as from a pool of selected blogs focused on a number of geographic areas, (ii) application of an online clustering technique on the data to detect recent trending topics, and (iii) refinement and ranking of clusters such that trends are detected and visualized. These three tiers are summarized in the next subsections.

3.1 The Data Collection Tier

The data collection tier involves special online data aggregators for collecting recently published content from Twitter and the Blogosphere. The content corresponds to some specific geographic area (such as a city level); leveraging the Twitter Streaming API and Google Blogger API (other possibilities in blogging and microblogging platforms can also be considered). While the first API provides a continuous stream of recently generated posts the second one is based on REST requests. To this end, based on a collection of identifiers of blogs owned by Blogger users who have declared that they reside within the monitored geographic area, we use the API for requesting new posts for each blog at a fixed time interval (e.g., daily, which is reasonable since we do not expect blogs to be updated as frequently as Twitter contents).

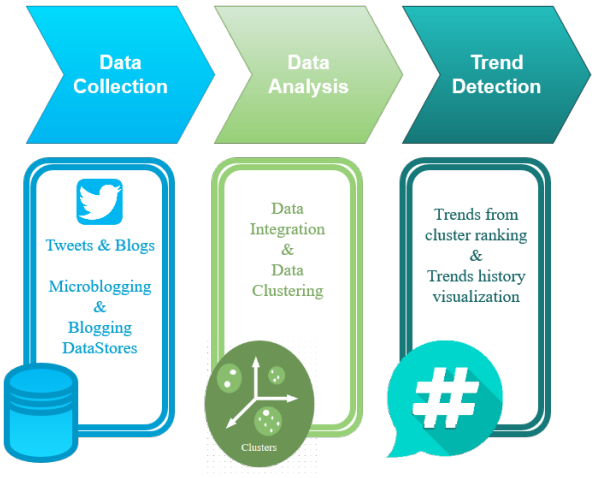

Figure 1. Microblogging trend detection outline.

3.2 The Data Analysis and Processing Tier

The retrieved posts (either tweets, blog posts, or extended tweets) are processed in order to produce clusters which contain posts pertaining to the same topic. Data are filtered to remove low quality content with typical approaches including filtering out tweets/blog posts with very few terms, etc. Text sanitization techniques are applied, filtering out common words (defined in a stop word lists) and to perform stemming. Next, resources are represented with a common model that includes a unique identifier, a TF-IDF-based key-value map, a timestamp, and the source (tweet, blog post, or extended tweet). For a given source, the key-value map structure includes keys to all of the source's unique terms, taken from the initial data model's *text*, *tags* and *title* (for blogs only) attributes. Using the Lucene™ Search Engine library[19], separate indexes are kept for each resource type and for each attribute. Through these indexes, TF-IDF key-value maps are obtained for each attribute.

3.3 The Trend Detection and Visualization Tier

This tier builds on the basis of the outcome of data processing tier which produces three sets of clusters for the datasets of tweets, blog posts, and extended tweets. A given cluster can be characterized as *active* or *inactive* based on whether it corresponds to topics that are popular at the given time or to topics that are no longer considered as trending. Clusters' update rates are monitored to determine when a cluster should be made inactive due to limited activity. To this end, additional information is maintained for each cluster—*the evolution of the temporal distance between the timestamps*

of the last two resources assigned to the given cluster. By taking the moving average of the aforementioned parameter to smoothen its evolution, we can identify periods of time when the cluster is increasingly rising in popularity. This is due to users' intense activity when it is at its peak. To improve clusters' quality, the tiny sized clusters (with very few members) are considered as noise and thus are eliminated.

Active clusters are considered as representative of topics that concern web users at given times, however, in order to identify the actual trends, clusters should be ranked in terms of an activity measure. To this end, for each type of content (and for a given monitored location) Cloud4Trends retrieves the *active* clusters and ranks them based on their members' number and their mean timestamp—under the assumption that the "hottest" topics are those that are referred to in many resources and that are additionally being created on average close to the current time.

The topics that characterize each cluster are identified as the terms with the highest scores in the cluster's mean key-value map. Cloud4Trends then generates a summary description for each cluster comprising of few member terms or phrases based on their scores and their significance (hashtags, title terms, etc.), while the high-ranked clusters shape the trending topics for the given time.

In Cloud4Trends, trends are therefore calculated based on the three different data sources for each location under investigation. Depending on the update rates of the sources (e.g., faster in Twitter while slower in blogs), one can decide on how often the clusters' "trending scale" will be recalculated.

4. Implementation Design of a Cloud-based Framework for Blogosphere Dynamics

Cloud4Trends is proposed to handle data-intensive use cases as it involves concurrent analysis of large sizes of social web data in an online fashion. In handling for example both tweets with unexpected peaks and blogs whose sizes may be considerably large, problems for handling large and fluctuating sizes of data arise. Feasible approaches should address parallel programming techniques required for many of our proposed operations, for example in the cases of:

- data which should be concurrently analyzed for the different geographic areas and their analysis should be done effectively
- blogs and Twitter data which should be collected in parallel
- data collection module which should be constantly available for receiving new data
- data processing which should be carried out for data already arrived and awaiting analysis under different concurrent process

Therefore, suggested ideas can heavily utilize the cloud computing paradigm which offers a significant ground for such social streams' mining applications due to its support via scalable and powerful infrastructures[8]. Our design requirements match well with the *MapReduce* computing paradigm which codifies a generic "recipe" for processing large datasets when this processing consists of more than one stage. The MapReduce technology matches the needs of the Cloud4Trends data analysis and processing tier, given that in a cloud-based deployment the *mapping* operations can be distributed into separate computer nodes. Prior to being ported to the Cloud, Cloud4Trends ran into a multi-core computer, designed over a software architecture which posed obstacles in aggregating and analyzing data from both Twitter and blogs. We believe that parallel approaches in cloud computing infrastructures constitute viable solutions for real-time large-scale data mining applications.

Cloud4Trends' aims are to validate the quality of the resulting clusters, observe, and quantify the differences in the trends resulting from the three data sources which represent different user groups. Thus Cloud infrastructure can be leveraged for efficiently handling both the data's high scalability, the requirement for real-time tweet processing and clusters' update, as well as for ensuring quality of service for an increasing number of end users (in line with the challenges stated in Table 1).

4.1 The VENUS-C Infrastructure

The suggested system (as described in the previous section) is currently implemented in the context of the so-called "Cloud4Trends" experiment entitled "*Leveraging the Cloud infrastructure for localized real-time trend detection in social media*", which runs on the VENUS-C infrastructure. VENUS-C (Virtual Multidisciplinary EnviroNments USing Cloud Infrastructures)[20] is a pioneering project that develops and deploys a cloud computing service for research and industry communities in Europe by offering an industrial-quality, service-oriented platform based on virtualization technologies and taking advantage of previous experience on Grids and Supercomputing to facilitate a range of research fields through easy dep-

loyment of end-user services.

VENUS-C offers several service components to allow a wide range of end-user applications (targeting mainly research groups and SMEs) to benefit from the advantages of a cloud computing platform, without having to develop custom Cloud-aware solutions. It offers a selection of programming models that, combined with appropriate data access mechanisms, constitute a convenient abstraction for deploying scientific applications on top of plain virtual machines. VENUS-C programming model enactment services expose their functionality via an *Open Grid Forums Basic Execution Service* (OGF BES)[21] and *Job Submission Description Language* (JSDL)[22] compliant web service interface, taking care of the enactment of a job at a given Cloud Provider. Each enactment service deploys a specific application on a number of either Windows Azure[23] virtual machines or Unix virtual resources from open source Cloud middlewares (OpenNebula[24] and EMOTIVE Cloud[25]).

A Cloud-based data management SDK is provided by VENUS-C for handling all data transfer operations between the Cloud infrastructure and the on-premises client applications, which supports Storage Networking Industry Association (SNIA) and Cloud Data Management Interface (CDMI) specifications[14]. Resource monitoring has also been taken into account in VENUS-C so that each end-user is able to identify its application's resource consumption via the VENUS-C Accounting Service.

4.2 The Cloud4Trends Cloud-based Architecture

The suggested design is implemented in Cloud4Trends under the previously described 3-tiered conceptual design structure. Cloud4Trends is implemented as a hybrid application based on the integration of on-premises client interface and multiple job execution components with different functionalities on top of the VENUS-C Cloud services infrastructure. In particular, Cloud4Trends uses VENUS-C Generic Worker programming model for job submission and application deployment on top of Azure.

Cloud infrastructure functionalities which enable effective data-driven and task-based job submissions are exploited. Figure 2 illustrates the three Cloud4Trends modules—the Collect module, the on-the-Cloud module, and the Cloud services module. These modules support the proposed 3-tier design (described above) such that the Client module implements Data Collection and Visualization tiers, whereas Cloud-based module implements Data Processing and knowledge extraction tier.

More details on this implementation are given[3], at which an indicative workflow is proposed to be orchestrated as follows: *Collect module*, which is responsible for collecting data from social Web (Twitter and Blogosphere) and initializing new experiments when required by the researchers, submits new Parsing jobs (executed by Twitter or Blog Parsers) to the cloud via the Job Submission Client of *Generic Worker*, which is hosted at the Execution Service, when new data are available. The required data for each given job are uploaded as a batch, using the Data Access Service to Azure Blobs. Cloud4Trends' *Indexing Service Module* consists of three separate Azure services (for indexing tweets, blog posts, and extended tweets) that are responsible for Full Text Indexing of the parsed data using the Lucene library for Azure. When an Indexing service completes its execution it initiates a *Splitter* Job using the Generic worker's Job Submission Client, which receives the new resources as input data. Splitter application also downloads the appropriate currently active clusters from the corresponding Azure Table. Different *Similarity* estimation Workers (Mapper jobs) are submitted by each Splitter job via the Execution Service module and in particular using the Generic Worker Local Job Submission service which calculate the similarity scores between the new resource and the respective active clusters. An *Aggregation* Worker (Reducer job) is also submitted by the Splitter Job via the Local Job Submission service—to

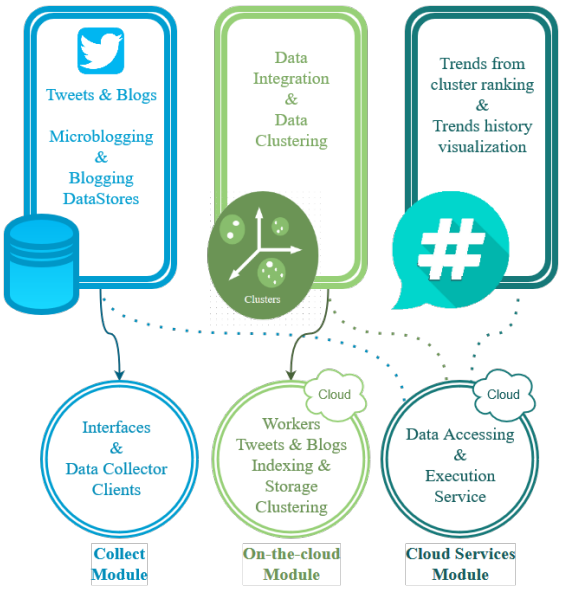

Figure 2. Trend detection cloud-based framework.

collect all similarity score combinations emitted by the corresponding Mapper jobs and identify the best match to an existing cluster for each resource.

5. Cloud Architectures for Social Data Analysis

Cloud4Trends has exploited the *Windows Azure* infrastructure which provides functionalities but also poses additional challenges since in cloud computing a predefined number of resources (i.e., number of CPU, RAM, etc.) is available to be utilized and shared. Each computer resource in the cloud executes a specific system's service; the *Worker Role* and Windows Azure provide several instances of resources to be defined on each Worker Roles. By initiating more than one instance of a Worker Role in order to provide parallelization, the number of reserved resources is increasing according to the defined resource's instance. As such, great emphasis should be given on the number of resources which are assigned on each *Worker Role* in order to efficiently divide the shared resources to the overall system's *Worker Roles*. An erroneous partition of the cloud resources over the *Worker Roles* may lead to non-scalable system architectures. Moreover, the *Azure Service Bus* service is provided to establish a fail-safe communication channel among the *Worker Roles*; *Virtual Machines* are hosted into the Windows Azure infrastructure which allocates same shared resources with the other Worker Roles, under the same communication network to avoid network delays during their communication.

Based on the above specifications, we propose five distinct system architectures emphasizing on the differences among single-node and cloud-computing architecture:

- *1st system architecture; as a lightweight suggestion* at which a sequential flow of the social text clustering process over the cloud is accomplished, as shown in Figure 3. No parallelization is provided in any of our framework's component and components of the same tier are included into a single Worker Role, providing the minimum number of Worker Roles and just two Virtual Machines are to be generated for efficiently executing the component's processes—*Token Identifier VM* and *Dimension Manager VM*.
- *2nd system architecture; to focus on the text processing* and at which distinction of the tweet collection and the Json parsing process is emphasized. While the process of roles do not require high memory allocation, small instances of each role is needed and multiple instances of the Json Parser Role and the Tweet Preprocessing Roles are initialized (Figure 4) to provide components parallelization and offer a more scalable solution.

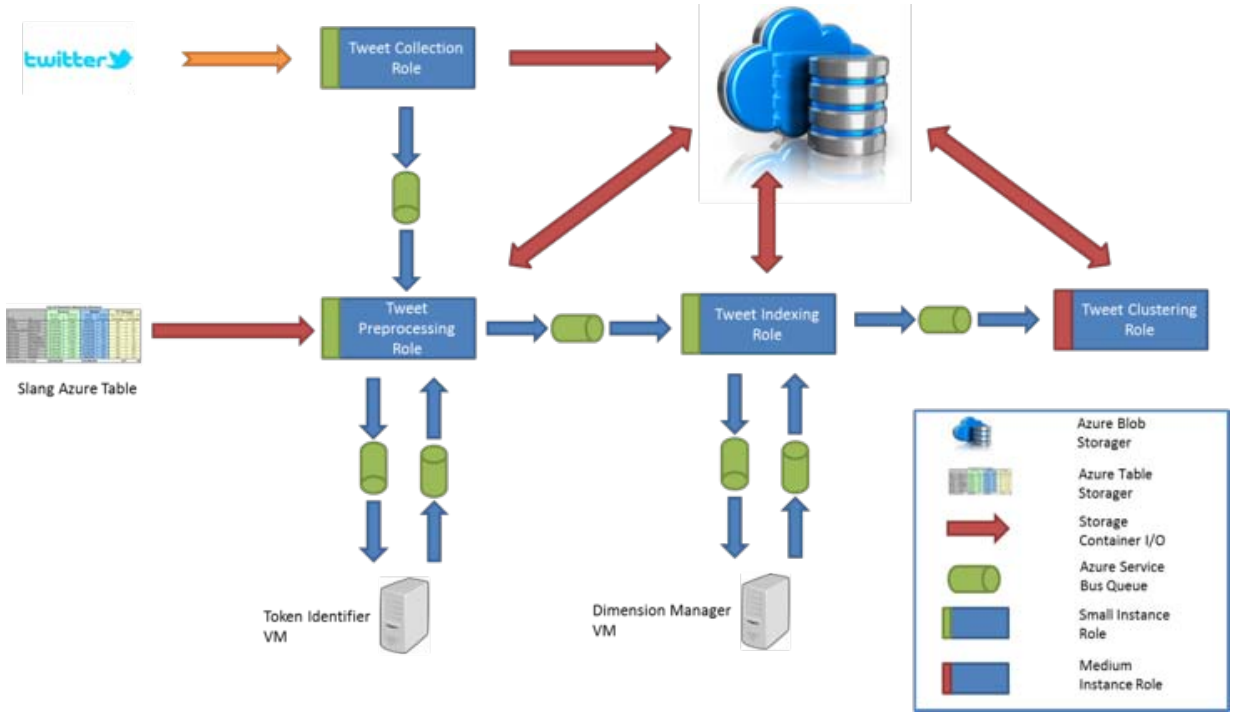

Figure 3. 1st system architecture (lightweight).

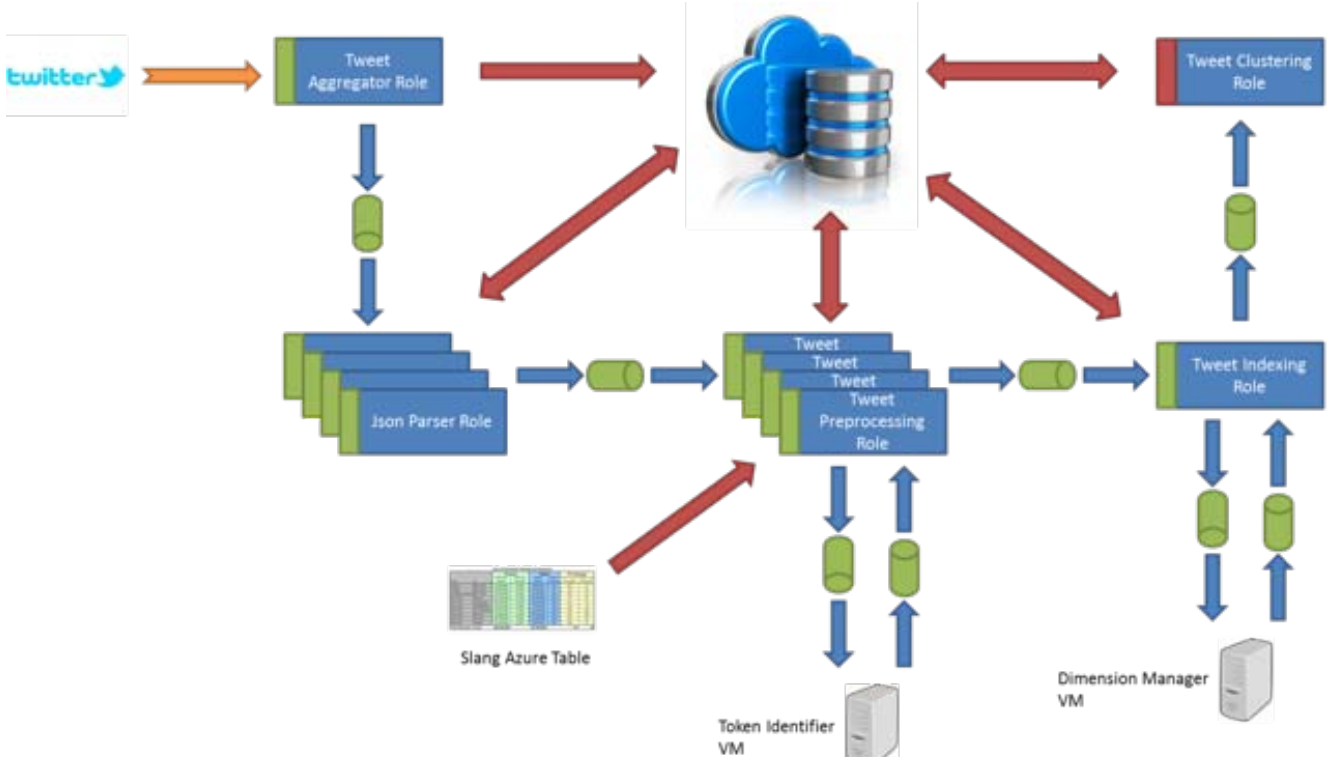

Figure 4. 2nd system architecture (text processing).

- *3rd system architecture; to support Tweet Indexing* and at which parallelization may be accomplished on the Vector Handler component, while the *Indexing* and the *Dimension Manager* components are not required to be parallelized (Figure 5). Since Windows Azure enables programming balancing of a Worker Role's instances, flexibility and dimensionality reduction is reached.
- *4th system architecture; to place emphasis on the parallelization of clustering components* and at which Text Clustering Role is divided into three distinct roles, as shown in Figure 6. This approach supports (i) Similarity Calculator Role, (ii) Results Comparator Role, and (iii) Cluster Manager Role, such that Similarity Calculator Roles are processed effectively.
- *5th system architecture; to offer parallelization in most of the framework's components* (Figure 7). Combining the 3rd and 4th system architectures, maximum parallelization is targeted, for scalable solutions in real-time, geo-located social text clustering. Medium instances of resources are assigned on *Cluster Manager Role* and *Dimension Manager VM* while small instances are assigned to the other of the system's roles.

6. Experimentation Results

For testing purposes, real-time social text clustering approach uses the FSD Corpus[26]. This dataset was collected by the Cross project which deals with identifying real-time story detection across multiple massive streams. Datasets consist of almost 52 million tweet IDs along with their corresponding user screen names (from Twitter's Streaming API). Additionally, the labeling process has been applied on a subset of the dataset to categorize the tweets on a distinct topic. This subset consists of 3034 tweets which were tagged as being on-topic for one of the 27 topics defined.

Great emphasis is given on the computer architectures used to apply our experiments, evaluating the benefits and limitations of each architecture. The same experiments are conducted over the different proposed (Section 5) cloud-computing architectures and also on single node solutions. Details for each of the architecture's service technical details are summarized in Table 2.

The initial experimentation focused on the two computing infrastructures paradigm and we stress-tested the proposed framework with a dimensionality reduction technique, for both cloud and traditional computing infrastructures.

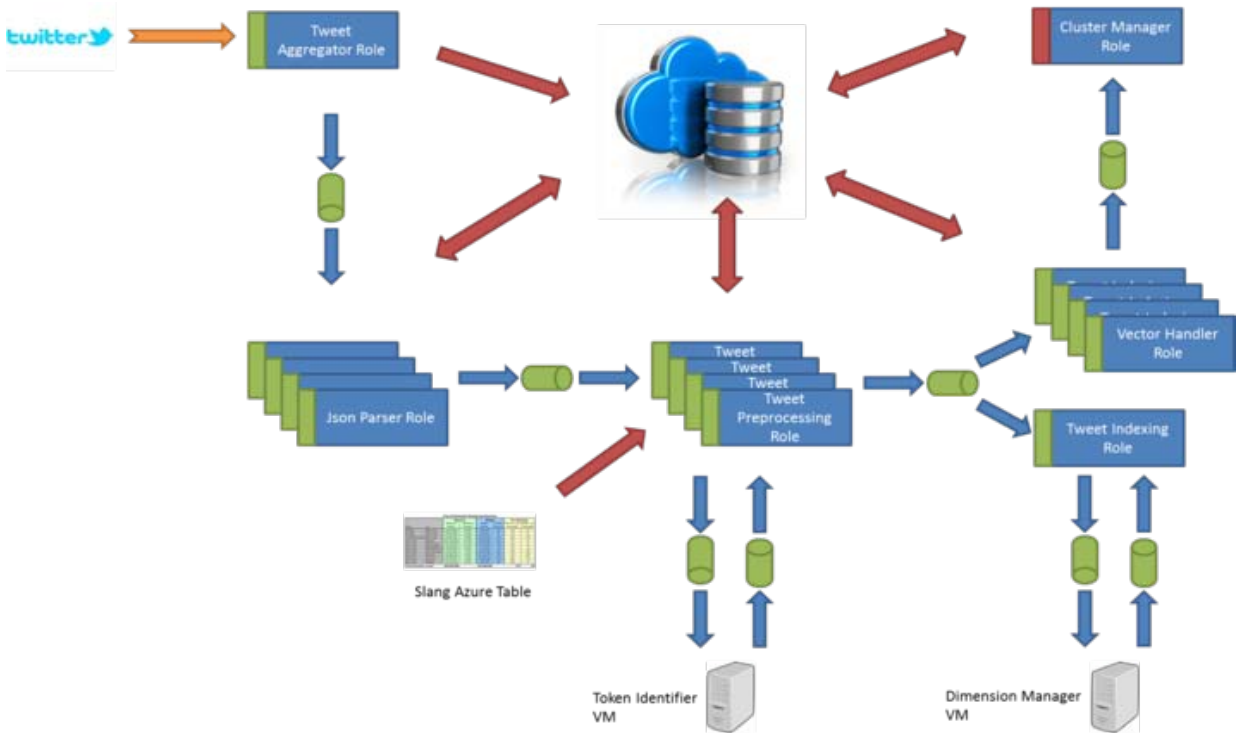

Figure 5. 3rd system architecture (indexing).

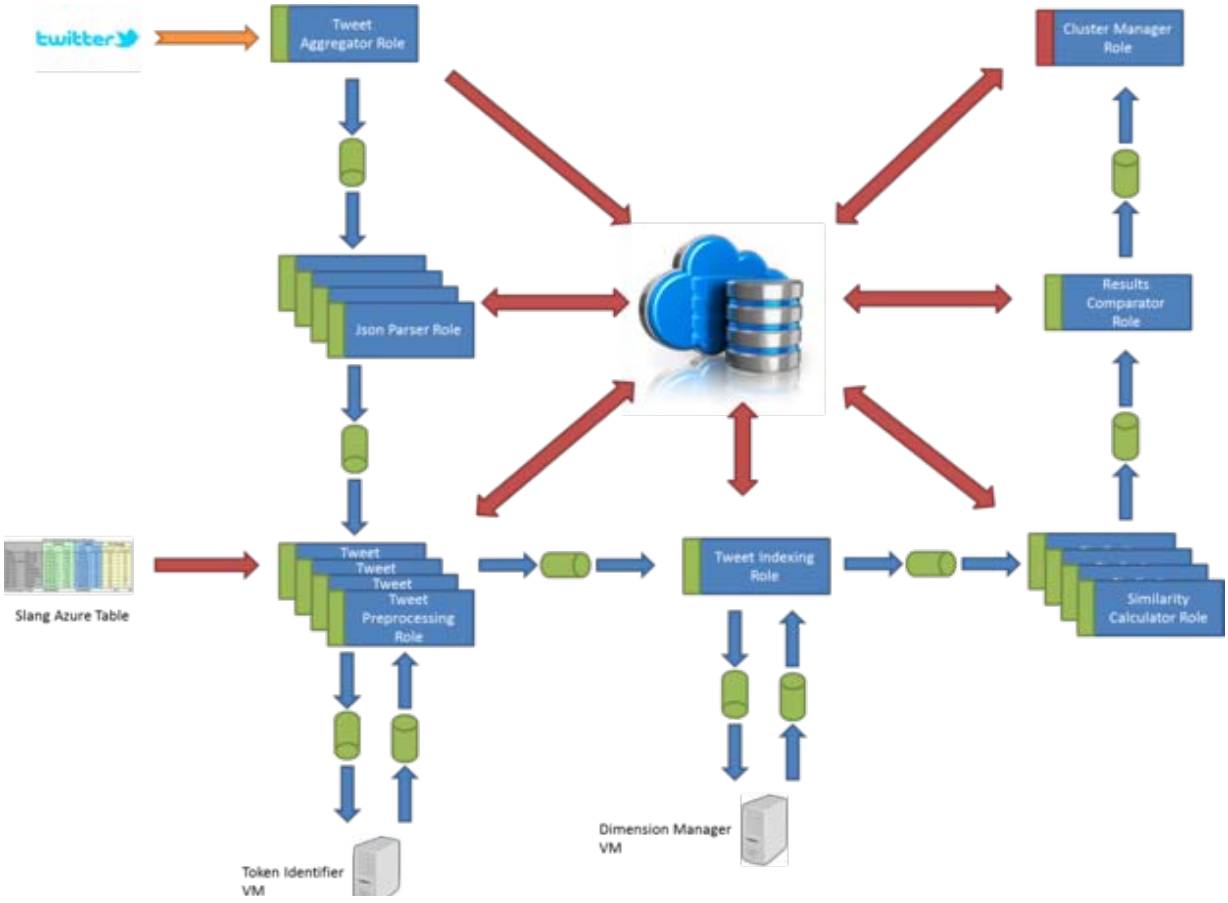

Figure 6. 4th system architecture (clustering).

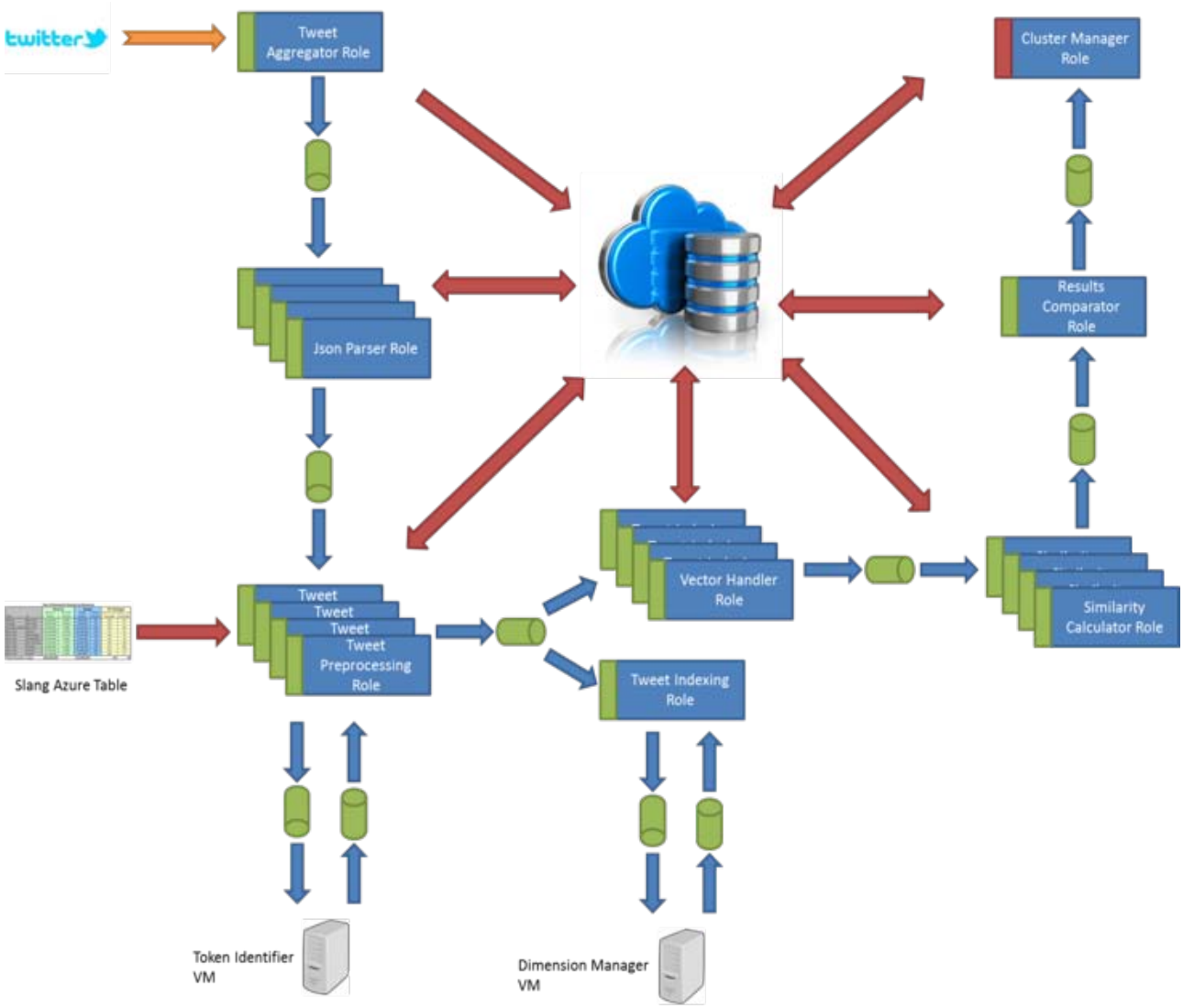

Figure 7. 5th system architecture (parallelization).

Table 2. Experimentation technical details

Services	System architectures				
	1st	2nd	3rd	4th	5th
Tweet Preprocessing Role	1x1 Core 1x1.75GB RAM	4x1 Core 4x1.75 GB RAM	3x1 Core 3x1.75 GB RAM	2x1 Core 2x1.75 GB RAM	2x1 Core 2x1.75 GB RAM
Tweet Indexing Role	1x1 Core 1x1.75GB RAM	1x1 Core 1x1.75GB RAM	1x1 Core 1x1.75GB RAM	1x1 Core 1x1.75GB RAM	1x1 Core 1x1.75 GB RAM
Tweet Clustering Role	1x2 Cores 1x3.5GB Ram	1x2 Cores 1x3.5GB Ram	1x2 Cores 1x3.5GB Ram	–	–
Results Comparator Role	–	–	–	1x1 Core 1x1.75GB RAM	1x1 Core 1x1.75GB RAM
Cluster Manager Role	–	–	–	1x2 Cores 1x3.5GB Ram	1x2 Cores 1x3.5GB Ram
Dimension Manager VM	1x4 Cores 1x7GB Ram	1x4 Cores 1x7GB Ram	1x4 Cores 1x7GB Ram	1x4 Cores 1x7GB Ram	1x4 Cores 1x7GB Ram
Total Number of Resources	10 Cores 17.5 GB Ram	17 Cores 29.75 GB RAM	20 Cores 35 GB Ram	20 Cores 35 GB Ram	20 Cores 35 GB Ram

Figure 8 showcases the fact that Windows Azure cloud computing infrastructure performed better when compared to the single-node infrastructure, and it was observed that regarding its high computational costs and parallelization demands, cloud computing infrastructure has provided the appropriate resources to efficiently support such a demanding and scalable solution.

The basic workflow of the experiments and the different parameters of our proposed methodology including the appropriate tweets' selection and the rate of the dimensionality reduction performed have been stress-tested under several scenarios. Here, we focus on the experimentation results which are relevant in the proposed architectures' emphasis and potential on geo-located time dependent knowledge extraction from social text threads.

Figure 9 summarizes the proposed cloud computing architectures (details at Section 5) and the execution time of each system architecture, when no dimensionality reduction is performed. As depicted in this figure, the 1st system architecture demands more hours than the other system architectures to accomplish the experiment. Instead, when parallelization occurred on the other system architectures, the experiments' execution times diminish. In the 5th system architecture where maximum parallelization is provided in most of its services, the total execution time is lower than the others. The scalability obstacle though is confronted

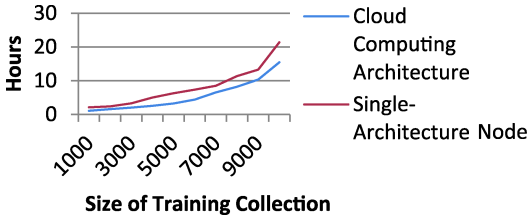

Figure 8. Comparison of computer infrastructures.

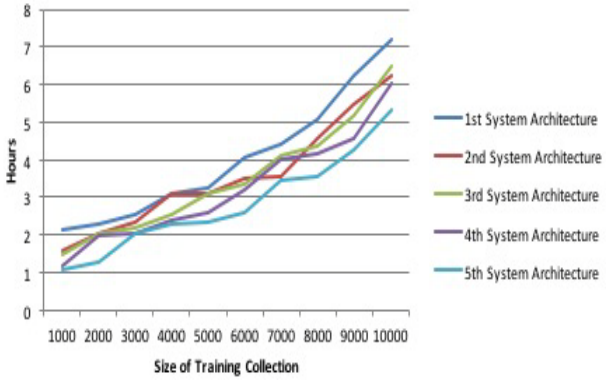

Figure 9. Azure hours of execution time — no dimensionality reduction.

with the support of the parallelization on the systems' services.

The dimensionality reduction approach has been further applied on cloud computing architectures; timed execution of each system architectures are shown in Figure 10. Similar to the previous approach, 1st system architecture requires the most execution time than the other architectures, while the 5th system architecture provides a more scalable solution.

Finally, the difference among the proposed system architectures when the dimensionality reduction technique is applied is stressed-tested, by considering that parallelization is provided on the distinct systems' services (Figure 11). The total execution time is shown to be decreased since the tweets' vector representation is reduced, and the similarity calculation along with the existing clusters is not of high-computational costs. Such cases exhibit no bottlenecks in the clustering process and their applicability in geo-located time dependent use cases is promising.

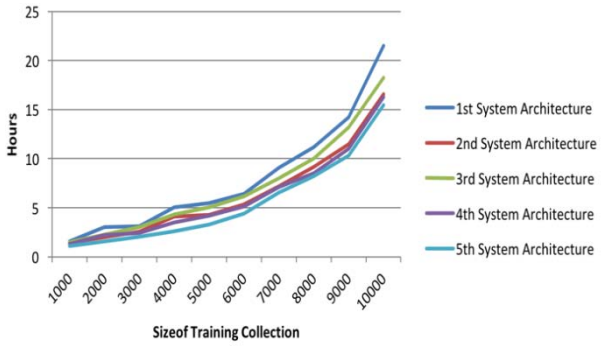

Figure 10. Azure hours of execution time — 25% latent factors dimensionality reduction.

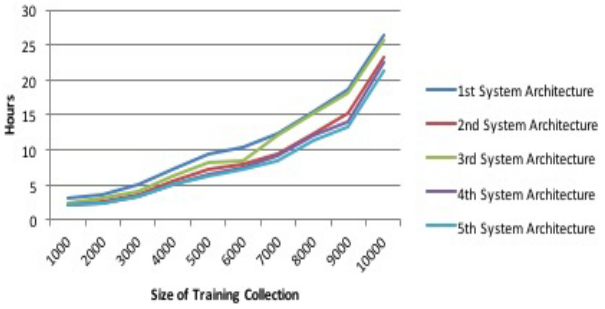

Figure 11. Single-node hours of execution time — 25% latent factors dimensionality reduction.

7. Conclusion and Future Work

This proposed work has focused on utilizing cloud computing paradigms, by the use of VENUS-C ena-

ctment services and under data-dependent jobs which stress-tested five different proposed architectures. Scalability issues are addressed by dedicated components which enable dedicated Cloud Tables for monitoring clusters' "activity" and for updating their states. Dynamics can thus be detected under clusters' representations at parameterized time intervals, with geo-spotted trends identification.

Future work is planned around the following axes: (i) the fine-tuning of the clustering and trend detection algorithm and the experimental evaluation of results, (ii) the implementation of a shard-based distributed Indexing service since for the time being the service for each type of resource is deployed on a single instance, (iii) measuring the system's performance for different design parameters, (iv) creating a web-based user interface (hosted either on Cloud or on premises) for visualization of the detected real-time trends and trends' analytics, and (v) stress-test the proposed framework and architectures under different cities datasets and requirements.

The benefit that Cloud4Trends and different architectures offer is that they verify that Cloud-based architectures constitute a viable solution for online social web geo-located and time-related data mining applications. In particular, the proposed work enables massive data analysis at a distributed setting, thus reducing the prerequisite for real-time applications' data processing times. At the same time, this allows easier testing of new scenarios, achieving high-quality results under different demands of cloud-based architectures which can improve an application's capability sharing. Overall, these are beneficial to both researchers and entrepreneurs in actual real-use cases.

In summary, the proposed framework (as realized by the Cloud4Trends experiment), demonstrates that porting trend detection into the Cloud is a very suitable solution considering the challenges posed by the data, geo-location features and time intensive processes involved in online collection and analysis of large and evolving geo-located Web 2.0 datasets.

Conflict of Interest and Funding

No conflict of interest was reported by the authors. This work is partly funded and realized within the Cloud4Trends pilot project of the VENUS-C project. VENUS-C (Virtual multidisciplinary EnviroNments USing Cloud infrastructures) is co-funded by the GÉANT and e-Infrastructures Unit, DG Information Society and Media, European Commission.

Acknowledgments

The authors thank the anonymous reviewers for their valuable comments and suggestions.

References

1. *Twitter usage statistics*, n.d., viewed February 21, 2016, <http://www.internetlivestats.com/twitter-statistics/>
2. Antoniades D and Dovrolis C, 2015, Co-evolutionary dynamics in social networks: a case study of Twitter. *Computational Social Networks*, vol.2(1): 1–21. http://dx.doi.org/10.1186/s40649-015-0023-6.
3. Vakali A, Giatsoglou M, and Antaris S, 2012, Social networking trends and dynamics detection via a cloud-based framework design. *Proceedings of the 21st International Conference on World Wide Web (WWW' 2012)*, 1213–1220. http://dx.doi.org/10.1145/2187980.2188263.
4. *Google trends*, n.d., viewed February 22, 2016, <https://www.google.com/trends/>
5. Glance N S, Hurst H and Tomokiyo T, 2004, BlogPulse: automated trend discovery for weblogs, in *WWW 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, May 2004*.
6. *Trendistic homepage*, n.d., viewed February 24, 2016, <http://trendistic.com/>
7. *FAQs about trends on Twitter*, n.d., viewed February 27, 2016, <https://support.twitter.com/articles/101125>
8. Livenson I and Laure E, 2011, Towards transparent integration of heterogeneous cloud storage platforms. *Proceedings of the Fourth International Workshop on Data-intensive Distributed Computing (DIDC '11)*, 27–34. http://dx.doi.org/10.1145/1996014.1996020.
9. Giatsoglou M and Vakali A, 2013, Capturing social data evolution using graph clustering. *IEEE Internet Computing*, vol.17(1): 74-79. http://dx.doi.org/10.1109/MIC.2012.141.
10. Uchida M, Shibata N and Shirayama S, 2007, Identification and visualization of emerging trend from blogosphere. *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 305–306.
11. Mathioudakis M and Koudas N, 2010, TwitterMonitor: trend detection over the twitter stream. *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD '10)*, 1155–1158. http://dx.doi.org/10.1145/1807167.1807306.
12. Clarg E and Araki K, 2011, Text normalization in social media: progress, problems and applications for a pre-processing system of casual English. *Procedia — Social and Behavioral Sciences: Special Issue of Computational Linguistics and Related Fields*, vol.27: 2–11.

http://dx.doi.org/10.1016/j.sbspro.2011.10.577.

13. Reuter T, Cimiano P, Drumond L, *et al*. 2011, Scalable event-based clustering of social media via record linkage techniques. *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*.
14. Storage Network Industry Association (SNIA), n.d., *Cloud Data Management Interface (CDMI)*, viewed February 11, 2016, <http://www.snia.org/cdmi>
15. Whang J J, Sui X and Dhillon I S, 2012, Scalable and memory-efficient clustering of large-scale social networks. *Proceedings of the 12th International Conference on Data Mining, 10–13 December 2012, Brussels*, 705–714. http://dx.doi.org/10.1109/ICDM.2012.148.
16. Bao J, Zheng Y, Wilkie D, *et al*. 2015, *Recommendations in location-based social networks: a survey*. *GeoInformatica*, vol19.(3): 525–565. http://dx.doi.org/10.1007/s10707-014-0220-8.
17. *Sysomos MAP social research engine*, n.d., viewed February 26, 2016, <https://sysomos.com/>
18. *Radian6 Buddy Media Social Studio*, n.d., viewed February 27, 2016, <http://www.exacttarget.com/>
19. *Lucene™*, n.d., viewed February 25, 2016, <http://lucene.apache.org/core/>
20. *Venus-C (Virtual Multidisciplinary EnviroNments USing Cloud Infrastructures)*, n.d., viewed February 27, 2016, <http://www.venus-c.eu/Pages/Home.aspx>
21. Foster I, Grimshaw A, Lane P, *et. al*. 2007, *Open Grid Services Architecture Basic Execution Service Version 1.0. Document GFD-R.108*, Open Grid Forum, viewed January 30, 2016, <https://www.ogf.org/documents/GFD.108.pdf>
22. Anjomshoaa A, Brisard F, Drescher M, *et al*. 2005, *Job Submission Description Language (JSDL) Specification, Version 1.0. Document GFD-R.056*, Open Grid Forum, viewed January 31, 2016, <https://www.ogf.org/documents/GFD.56.pdf>
23. *Microsoft Azure*, n.d., viewed February 18, 2016, <https://azure.microsoft.com/en-us/>
24. *OpenNebula.org*, n.d., viewed February 18, 2016, <http://opennebula.org/>
25. *EMOTIVE Cloud (Elastic Management Of Tasks In Virtual Environments) homepage*, n.d., viewed February 19, 2016, <http://www.emotivecloud.net/>
26. *Cross Project homepage*, n.d., viewed February 9, 2016, <http://demeter.inf.ed.ac.uk/cross/index.html>

RESEARCH ARTICLE

Cloud computing security: protecting cloud-based smart city applications

Alkiviadis Giannakoulias*

European Dynamics SA, 209, Kifissias Av. & Arkadiou Str., 15124 Maroussi, Athens, Greece

Abstract: Data security is a major concern in cloud computing environments as they provide much scope for intruders to attack. Data centres in cloud environments hold valid information that end-users would conventionally have stored on their computers. Moving information towards centralised services may have an adverse effect on the security of users' interactions with files kept in cloud cupboard spaces[1], for example accidental or deliberate alterations or deletions of information from the cloud server by the Cloud Service Provider (CSP). This necessitates the deployment of some sort of mechanism to ensure the safety of information integrity[2]. Public sector organisations have much to gain by adopting a cloud computing approach to service delivery in their ICT environments. However, these benefits must be reaped without compromising core requirements and institutional values.

This paper focuses on the security issues that may arise when public sector organisations consider transitioning to an Open Source Software (OSS) Infrastructure as a Service (IaaS) Cloud Infrastructure (OpenStack), although the same issues are likely to be found in other OSS cloud computing software like Apache CloudStack[3], Eucalyptus[4], and OpenNebula[5]. We examine legal implications, regulatory and standards compliance, new attack vectors resulting from vulnerabilities coming from virtualisation technologies, data integrity issues such as encryption and access controls, and security checks to be performed on the services prior to their movement to the cloud. In addition, some of the most important security threats in cloud computing are presented, followed by key recommendations on how to address them, namely security standards and certifications, service provider auditing, secure APIs, transport layer protection, authentication and encryption key management, and cloud service agreements.

Keywords: STORM CLOUDS, OpenStack, cloud security, certification, auditing, security by exception and group management, introspection, host-based security, firewall, network security architecture, transport layer protection, access right management, configuration management, patch management, cryptography

*Correspondence to: Alkiviadis Giannakoulias, European Dynamics SA, 209, Kifissias Av. & Arkadiou Str., 15124 Maroussi, Athens, Greece; Email: Alkiviadis.Giannakoulias@eurodyn.com

Received: February 4, 2016; **Accepted:** March 11, 2016; **Published Online:** May 9, 2016

Citation: Giannakoulias A, 2016, Cloud computing security: protecting cloud-based smart city applications. *Journal of Smart Cities*, vol.2(1): 66–77. http://dx.doi.org/10.18063/JSC.2016.01.007.

1. Introduction: Security Definition

Cloud computing security is an evolving sub-domain of information security and refers to a broad set of policies, technologies, and controls deployed to protect data, applications, and the associated infrastructure[6].

There is a number of security concerns associated with cloud computing; they can be broadly classified into two categories, namely issues faced by Cloud Service Providers (CSP) and those faced by Cloud Service Consumers (CSC). Providers must ensure that their infrastructure is secure and clients' data and applications are protected; consumers, on the other hand, must ensure that their provider has taken appropriate security measures to protect their information[6].

Current Cloud delivery models (whether implemented on an Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), or Software-as-a-Service (SaaS) model) are ruled by Service Level Agreements (SLA) that normally define mutual supplier and user expectations and obligations. The central idea behind such models is that the consumer ought to trust the supplier[7].

1.1 Delineation of Responsibility

Venturing into a public cloud environment, especially via an IaaS model, security becomes a shared responsibility. Although there are certain measures which a cloud provider will apply to ensure that Virtual Machines (VM) stay secure, a considerable number of tasks are left in the hands of the tenant (cloud consumer). Figure 1 shows the cloud stack[8], presented as an OSI model for the cloud. As can be seen, responsibility for security is equally split between tenant and provider.

To make our notions more precise, some definitions and explanations on the IaaS model are in order:

- **Provider** is the person/organisation that has built the cloud and offers the relevant service.
- **Tenants** are those asking the provider for access to that service.
- The **red line** identifies where the delineation of responsibility is depending on the cloud service model used.
- **Facility**, found at the bottom of the stack, refers to installations, such as buildings, doors that lock and other related objects. As tenants have no control here, responsibility for those lies with the provider.
- **Network**, refers to connected physical entities such as wires, cables, switches, routers, located inside the Facility.

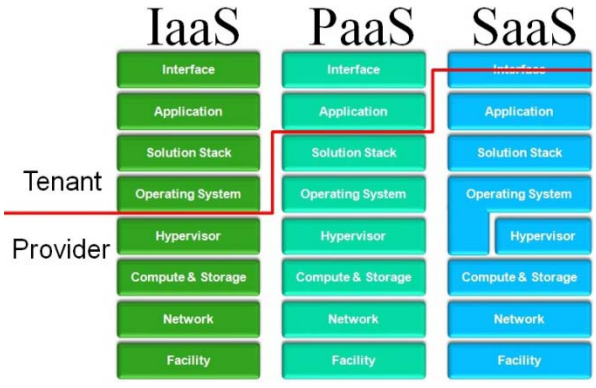

Figure 1. Cloud service models delineation of responsibility.[9]

- **Compute & Storage** refers to CPUs, motherboards, hard drives, etc.
- **Server virtualisation** is the technology for creating and managing VMs, implemented by hypervisors.
- **Virtual Machine** refers to the software container used to emulate hardware for the Operating System (OS) running inside of it.
- **Solution stack** refers to any type of application language running, such as .Net, Perl, Python, or others.
- **Applications** refer to specific for purpose software, for example web applications.
- **Interface** refers to implemented GUIs, graphic web interfaces, or even sets of RESTful APIs.

1.2 Cloud Computing Security Threats

Some of the most important security threats in cloud computing include[10–12]:

(1) Ease of use. The simplicity of cloud services (i.e., any resource that is provided over the Internet[13]) is appealing to attackers for malicious purposes like spamming, malware distribution, command-and-control servers, distributed denial-of-service (DDoS) attacks, password/hash cracking, etc.

(2) Vulnerable data transmission. Since data can be intercepted by man-in-the-middle attacks, data transferred from clients to the cloud needs to be properly encrypted by using Secure Socket Layer (SSL)/ Transport Layer Security (TLS).

(3) Insecure APIs. Since cloud services are exposed by APIs, it is imperative for the CSPs to secure them. The reason is that attackers can manipulate data with the right authentication/authorisation token.

(4) Malicious insiders. CSP personnel having complete access to enterprise data and resources can, undetected, gather confidential information. Hence, CSPs should employ security measures in place to track employee actions such as data viewing.

(5) Shared technology issues. Shared infrastructure resources amongst various tenants can lead to vulnerabilities, such as hypervisor exploitation, VM sandbox break-out, unauthorised access to shared data through side-channel attacks, and others.

(6) Virtualisation technology issues. Virtualisation is a critical part of cloud computing with virtualised operating systems being the backbone of IaaS[14]. They provide an important layer of abstraction from physical hardware, thus enabling the elasticity and

resource pooling commonly associated with cloud. Given its importance, it is only natural to ask: "what constitutes our biggest concern regarding security in virtualisation?" The answer is the hypervisor, which ties together all our operating systems. Traditionally, in the past, machines were plugged to a switch; the only way for machine A to interact with machine B was via the switch itself, in other words via the IP stack, i.e., via network-based communication. In virtualisation, a hypervisor manages all our VMs containing operating systems, thus creating a software bridge among them. As shown in Figure 2, this means that we have to secure the two connection points (depicted in red).

From a theoretical perspective, the hypervisor appears as a less secure solution. However, from a practical perspective, a hypervisor can augment our security.

(7) Data loss. Data stored in the cloud could be lost due to a number of reasons, namely hard drive failure, CSP going out of business, accidental data deletion by a CSP employee, data-theft by an attacker, etc. The best way to protect against such threats is via data backups for subsequent restore.

(8) Data breach. Side-channel attacks refer to a situation where VMs running on the same physical host can access the data of another VM, leading to a data breach.

(9) Insecure or incomplete data deletion. Requests to delete cloud resources may not result in actual data wiping. Data deletion could be incomplete either because extra copies are being stored but not deleted, or because the physical resources are being used by other clients.

(10) Security incident handling. CSCs rely on CSPs to handle detection, reporting and management of security breaches.

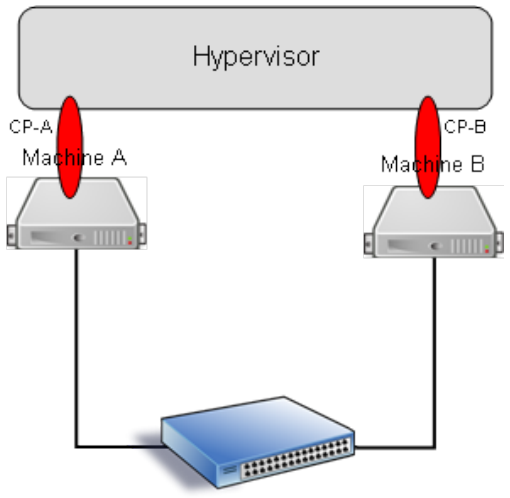

Figure 2. The lack of an air gap.

(11) Data protection. Issues like exposure or release of sensitive data as well as loss or unavailability of data are major data protection risks for both cloud consumers and providers. Cloud consumers have no effective information about the data handling practices of the CSP so as to ensure that data is handled in a lawful way.

(12) Account/service hijacking. If cloud access is only password protected, an attacker who knows the password will have equally easy access. It is therefore better to use two-factor authentication when available.

(13) Unknown risk profile. Lack of knowledge of a cloud provider's security protocols and policies can contribute to making it harder to "calculate" a risk profile. Security by obscurity may be a low effort but it can result in unknown exposures[15].

(14) Denial of service (DoS). Cloud services can get disrupted by attackers issuing a DoS attack against the cloud service rendering it inaccessible.

(15) Lack of understanding. Users should understand the cloud security threats in order to properly defend against them. This means that organisations should invest time and resources in education and training before moving to the cloud.

(16) Access privileges. CSPs should be able to demonstrate that they enforce adequate hiring processes, oversight, and access controls to enforce administrative delegation.

(17) Regulatory compliance. Cloud consumers are accountable for their own data even when this is in a public cloud; they should ensure that CSPs are ready and willing to undergo audits. Cloud consumers should use cloud computing services in a responsible way and should ensure that the CSP has appropriate certifications in place.

(18) Data segregation. Most public clouds are shared environments; it is critical to ensure that hosting providers can guarantee complete data segregation for secure multi-tenancy.

(19) Loss of governance. Public cloud consumers unavoidably tend to handle control to CSPs over a number of issues affecting security without at the same time being able to impose strict SLA commitments on the part of the CSP. This creates gaps in their security defences.

(20) Responsibility ambiguity. Responsibility for security aspects is spread across both the cloud con-

sumers and CSPs, potentially resulting in vital parts being left unattained in case of failure to allocate responsibility clearly.

(21) Vendor lock-in. Use of proprietary services from a specific CSP that does not support portability of applications and data to other CSPs can lead to vendor lock-in and higher risks of data and service unavailability.

(22) Authentication and Authorisation. Assurance regarding the identity of users (employees, contractors, partners, and customers) is important as resources are accessed from anywhere. Strong authentication and authorisation becomes a critical concern.

2. Security Recommendations

According to David S. Linthicum, senior vice president of Cloud Technology Partners, CSCs are responsible for securing their data and for providing the security requirements that the CSPs should meet via appropriate technical means[16]. CSCs must determine their security requirements and must map those to the appropriate technology.

What we should keep in mind that the level of security provided in the cloud environment should be equal to or better than the security provided by traditional IT environments, otherwise we could be facing higher costs and potential loss of information. This would eliminate any potential benefits of cloud computing.

2.1 Service Provider Certification and Auditing

Certification of cloud computing services allows CSPs to show their customers that they meet certain standards, for example on network and information security.

Certification provides assurance to CSCs that their critical security requirements are being met. Therefore, they should identify which security certifications are important to them and to insist that their CSPs demonstrate their conformance.

Regrettably, many providers have taken a **security through obscurity** approach—they don't want to talk about what security controls they have put in place. The idea is that if they don't talk about their controls, then it becomes harder for an attacker to break in.

Moreover, as there is no single accepted security assessment program for CSCs that can help them evaluate the types of controls CSPs have in place, making risk assessments can be difficult. Programs like the United States federal government's **FedRAMP**[17] and the United Kingdom government's **G-Cloud**[18] are only focused on government cloud use. One effort currently underway is the **Cloud Security Alliance's Open Certification Framework** (CSA OCF). CSA is the world's leading organisation dedicated to defining and raising awareness of best practices to help ensure a secure cloud computing environment[19]. It comprises of the following programs[20]:

- **CSA Security Trust and Assurance Registry (STAR) Certification**[21]: The STAR Certification relies on an independent third-party assessment of a cloud provider against the ISO 27001 standard, as well as the CSA Cloud Controls Matrix (CCM).
- **CSA STAR Attestation**[22]: The STAR Attestation phase provides a report via the audit-reporting standard for customer consumption known as the Statement on Standards for Attestation Engagements Service Organisation Control (SSAE SOC) 2 Report[23].
- **CSA STAR Continuous**: STAR Continuous was originally planned for release in 2015. This service will provide a scanning and monitoring console for users to remotely assess cloud providers' control statements via the CloudAudit XML-based tag format and the Cloud Trust Protocol (CTP) for data transmission and retention.

Additional frameworks are available from the **Shared Assessments Program** and the **European Union Agency for Network and Information Security** (ENISA).

Hence, CSPs should state the security validations they have obtained or they should identify and describe the security they have implemented inside their environment using the CSA STAR program. This will provide more detailed information and a better understanding of the CSP's security posture.

Even in cases where the provider under consideration is not listed in the STAR registry, there is a solution, namely the **Consensus Assessments Initiative Questionnaire** (CAIQ), downloadable from the CSA website, which the provider can be asked to complete[24].

Additionally, the Cloud Standards Customer Council (CSCC) "Security for Cloud Computing: Ten Steps to Ensure Success"[12] provides a prescriptive series of 10 steps that should be taken by CSCs to evaluate and manage the security of their cloud environment with the goal of mitigating risk and delivering an appropriate level of support.

2.2 Security by Exception and Group Management

To assist public sector organisations in deploying their servers and services to an OpenStack IaaS cloud model, we leverage automation. The automatic deployment is obtained using Heat[25]. The automation process includes: (a) The configuration of the VM hosting the application and (b) The installation and configuration of the application and its dependencies. Validation of the automation process is made in collaboration with the public sector organisations and includes functional tests in order to ensure that the deployed application performs as designed. Once the automation process is validated it results in a master VM.

When it comes to security, we need not focus as much on the instances that get generated but on that master VM, since all VMs should be identical. Even if we deploy multiple VMs serving out slightly different content, then:

- OS is going to be the same
- Patches are going to be the same
- Configurations should be the same
- Same processes will be running in memory

Rather than having to look at them as a one-off, we relate them all back to the master VM. In this way, we can change the way we go about security from the one-off security mode to **security by exception**.

Security by exception is similar to the "spot the difference game". Should one of the VMs deployed appear different compared to the master configuration, we then have a strong indication of compromise (Figure 3). Using this way to look at the data and managing it as a **collective group**, makes things easier. For example, we need to know if it is normal for a Web Server to serve content without HTTPS in a secure site, or whether directory listing is enabled, or whether the X-Frame- Options HTTP response header is set on all web pages returned by the application. Not having this **intimate knowledge** may not lead us to the compromised server.

Another great example of the power of doing management via collective groups is shown in Figure 4. Any patching and configuration management should not be performed at each deployed VM. If a VM is missing a patch, it is because it is also missing from the scripts implementing the automation process and thus from the master. Update the scripts, re-launch out another copy and we are good to go! Once more, rather than having to do one-off security management, we are doing security management by exception.

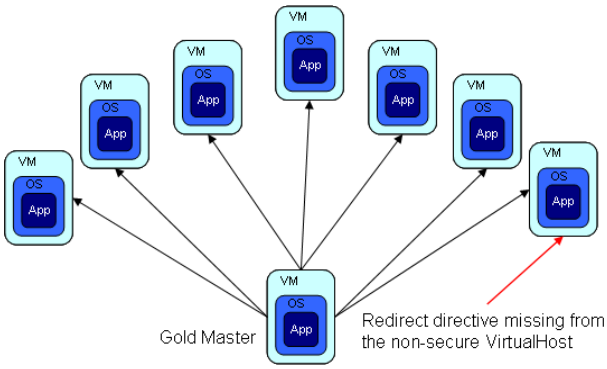

Figure 3. Spot the difference – case 1.

Now, there are two ways we can look at this. We can look at our deployed VMs and ask, "Does any of them differ from the master?" But there is another way we can parse this data too. Let us say that we are missing three patches on our master; we did build the master a month ago, three patches have come out since then, and all our VMs running in the cloud are currently missing three patches. A look at Figure 4 reveals that the VM on the right is fully patched and up-to-date, despite the fact that the master is not. The conclusion is that somebody has done something to that VM.

The idea is to compare the VMs to the master so as to investigate possible inconsistencies. In our example above, we had three missing patches on the master but would we really have cared had three missing patches appeared on all of our VM instances? The answer is yes and no. Yes, because we would want to install these three patches and burst back out again—this is an administrator's stand point. We wouldn't necessarily care from a security stand point; there is no indication that somebody has broken into the VM. If we now normalise out the three missing patches across the system and something appears to have been patched differently, then this is an indication of a serious problem. What is interesting here is that old school one-off server management would never catch such a problem—what we would have looked for would be for a patched and up-to-date machine. A one-off server, patched and up-to-date would not make us suspect that something is wrong. However, if we look at it in the context of Figure 4, then we have **an exception**. It is common practice for hackers once they break into a machine to pullout on their toolkit and patch the machine so some other script can break in exactly the same way. In other words, when all our other systems have not been patched, it is not uncommon for a single system to appear patched because of it being actually

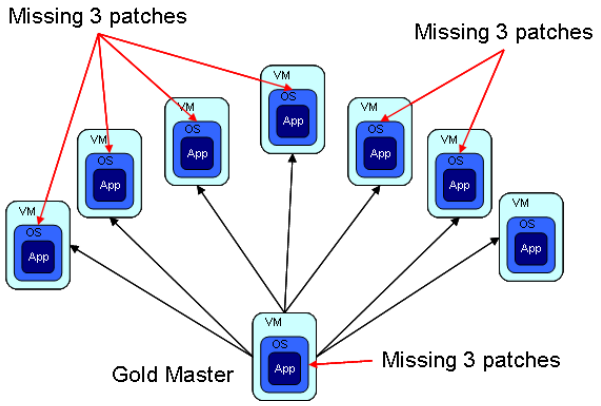

Figure 4. Spot the difference – case 3.

compromised. This is not something which needs intimate knowledge of each VM. It just shows right up.

To summarise, the process of managing by exception:

(i) simplifies the work,

(ii) offers new security controls which were not previously available.

For IaaS deployments:

- The CSP is responsible for their (physical) machines and for taking care of updating the software running on them (host OS, OpenStack, virtualisation technology, etc.). As such, the selected public CSP has to confirm that they support this security feature.
- The CSC is responsible for the (virtual) machines deployed and for taking care of updating the software running on them (OS, applications, frameworks, etc.). As such we should periodically review our VMs and apply updates as soon as they are available.

2.3 Operating System Firewall

When we deploy a firewall to protect a corporate network, we typically install it as an appliance on the perimeter, as shown in Figure 5. Here, each server is connected to a unique switch, which is then routed into the firewall. The servers are:

- logically segregated from each other, as the only routed path between networks is through the firewall,
- physically segregated from each other, as the only point of connectivity between them is through the firewall.

In other words, pull the firewall out of the mix and there is no possible way for the servers to talk to each other or to the Internet. This is one of the security benefits of physical segregation[26].

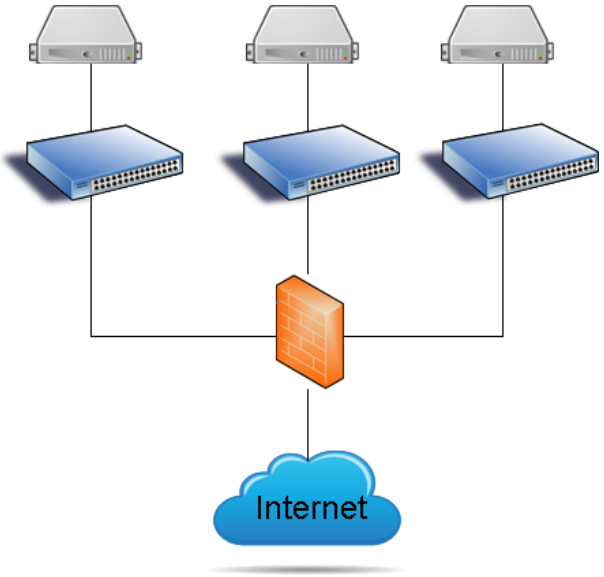

Figure 5. Classic firewall installation.

Can a virtual firewall appliance provide the same level of risk mitigation as a physical firewall appliance?

Figure 6 shows an identical configuration, only applied to an IaaS cloud with a virtual firewall appliance.

When virtual switches are deployed, we tend to think in terms of having multiple virtual switches; only one virtual switch is deployed, separated into logical partitions. Hence, apart from the risk that comes from the hypervisor (a software connection between the VMs), an additional identifiable risk comes from the virtual switch being another connection point among all the VMs. Should the virtual switch be compromised, access to the VMs (which normally would not be possible due to them being petitioned off) could be gained. If two VMs figure out a way to get connectivity

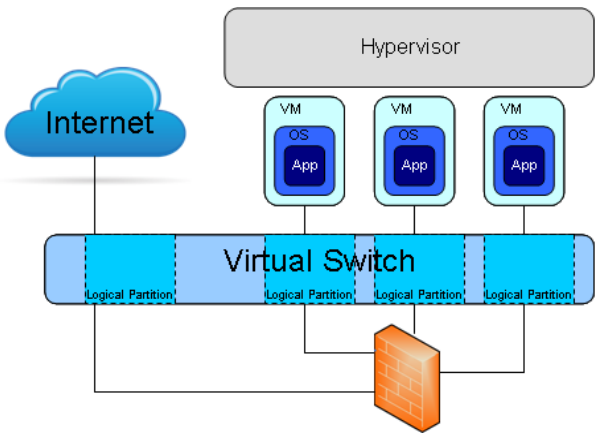

Figure 6. Virtual switch installation.

through the virtual switch, any risks that may exist within the virtual switch itself are propagated to the VMs. Here, firewall protection is not effective as these acts outside the virtual switch. By deploying a virtual appliance, we are protected from the Internet but not necessarily from other users operating in the same cloud environment.

In order to mitigate any risks coming from the virtual switch, we need to move the firewall to the other side of the switch.

One solution is to leverage a **hypervisor-based firewall**. Such firewalls are not only vendor specific but in some cases, CSPs do not even support them. Moreover, they require the use of introspection which introduces additional risks in public cloud environments as we analyse below.

Alternatively, we could leverage the **VM operating system firewall** software. Since such a solution controls traffic as it passes in and out of the VM, it is more than capable of negating any potential risks within the virtual switch. This option presents two advantages:

(i) It is the least expensive; the firewall is already present within the VM operating system.

(ii) It is the most portable—if the VM is moved to another CSP, host-based firewall protection obviously moves with it.

The main disadvantage is that it introduces an additional point of management.

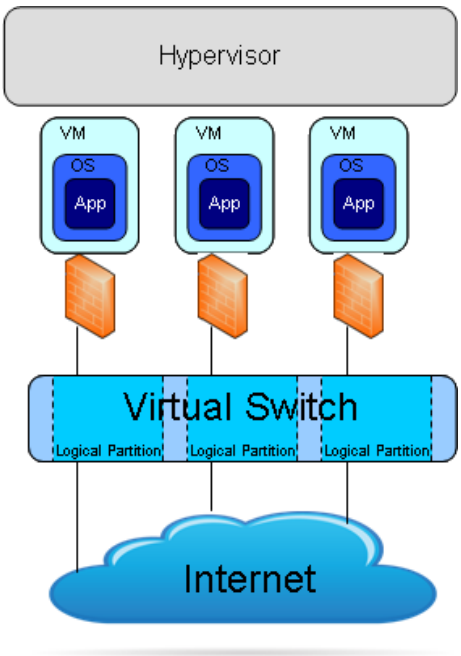

Figure 7. Hypervisor based firewall.

2.4 Introspection

Introspection is a security tool that we can leverage via the hypervisor in order to implement security on each VM. The capability of introspection can be leveraged for a wide range of security applications such as malware control, data loss prevention, firewalling between VMs, network intrusion detection between VMs, and forensics. If a hypervisor can see and interact with all the VMs running within a certain platform, it almost makes sense to ask whether it can be leveraged to implement security. With that being said, a hypervisor based solution secures ONLY a specific cloud group; if the VM is to be migrated to another, CSP protection is lost, resulting in vendor lock-in. Introspection is thus similar to a firewall—if a machine stays on the one side of the firewall, it can be protected; if it is moved, protection is not there anymore.

Despite some advantages, additional security issues introduced by introspection can elevate risk against an environment. The first is hypervisor bloat. Much of hypervisor security is predicated on keeping the code as small as possible. The fewer the lines of code, the less likely an attacker will find a problem which can be leveraged for malicious gain. By adding introspection capabilities to the hypervisor we increased the amount of code being processed[27].

Introspection also enlarges the attack surface of the hypervisor; as the hypervisor is made to interact more with each VM, increased interaction with an untrusted source results in increased probability of compromise.

A good example is Network-based Intrusion Detection Systems (NIDS) that sits on the wire and passively monitor traffic by processing packets and matching them against pre-defined malicious patterns. The problem here lies in the fact that NIDS interacts with the passing packets. Attackers can figure out a way to create a packet that would not hurt the targeted system but the NIDS sitting between the attacker and the target system. Such a packet can make the NIDS blow up and fall off-line. Hence, while using the hypervisor to monitor security within the VMs, hackers may find a way to create files or processes that don't necessarily harm the VMs themselves but instead attack the introspection system. In this way, a hacker can gain full access to the hypervisor which in turn has access to everything. The result will be a cascade effect leading to VMs themselves becoming compromised.

Another concern is that introspection can potential-

ly break segregation of duties[27], leading to uncontrolled access to VM data. Other similar arguments state that introspection generally creates a backdoor access to every VM.

For public deployment models, introspection is not suitable because any access to tenant data is unlogged. When the hypervisor monitors file processing within each VM, the VM itself does not monitor any activity. When the provider deploys introspection to monitor processes inside a VM, the tenant has no available audit trail of what took place. Two problems arise here:

- Possible uncontrolled access that cannot be audited
- Bulls-eye on the provider's back—if attacking the VM directly to access tenant's data is not possible, a provider (having access to all data running on the VMs) who uses introspection can be attacked

In contrast to public deployment, introspection makes a good architecture for private deployment models. It should be used when focus is placed in protecting the network, when risks to the hypervisor itself can be controlled, and where segregation of duties is not an issue. Moreover, introspection secures the infrastructure; by migrating a VM to another virtualisation infrastructure, risk mitigation provided via introspection may be lost. Fear of this happening can result in vendor lock-in.

2.5 Data Misplacement (Resource Allocation)

Data leaks are a risk when physical memory or data storage used by one virtual machine is reallocated to another[28]. Leaks occur when a VM shrinks or is no longer needed and freed resources are allocated to another VM. It is possible for the new VM receiving the additional resources to use forensic investigation techniques to acquire an image of the whole physical memory and data storage. It is not clear whether we can leverage this. It is certainly hard to argue that attackers keep on waiting until we make our partition smaller in order to try and get our data. It is entirely possible; however, that this is opportunistic—attackers may be able to find valuable information, not necessarily because they have a specific target but simply because the data is there to be had.

Hence, when reallocating resources from one VM to another, both must be properly secured. The old data present in the physical memory and in the data storage should be nullified so as to prevent other VMs from pulling data out of them and gaining access to important information still contained there.

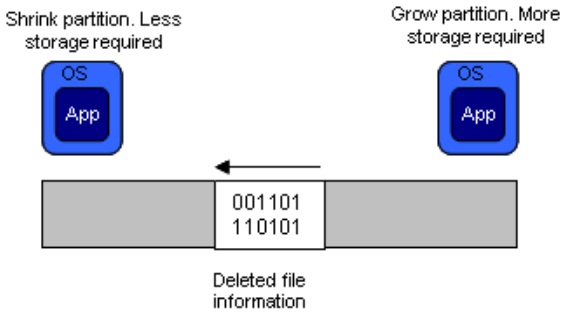

Figure 8. Data misplacement.

By design, OpenStack allows zeroing of all the data used by a virtual resource (VM or virtual volume) once the resource is released. This happens via internal KVM procedures without any intervention needed from the CSP's side. However, in order to clean the used memory, CSPs should ensure that they have updated the Cinder configuration (available in /etc/cinder/cinder.conf file) in order to delete blocks that have some written data.

2.6 Configure Access and Security

According to the principle of defence-in-depth[29], layered security mechanisms increase the security of the system as a whole. We should therefore use the additional **OpenStack** hardening mechanism related to the network ports, namely the **security groups**[30]. These are used to define a number of IP firewalling rules, describing the kind of network traffic allowed to go to or come from the VMs[31].

As these rules are project/application specific, CSCs should carefully review which protocol to use and the port range. However, at a minimum, SSH ONLY access should be allowed.

To support this operation, CSCs should import their public keys (**SSH credentials**) into OpenStack that will be **injected into the VM** when it is launched.

Here we should note that even when a VM is compromised, the security group rules still provide the required level of security. This is because they are implemented in the host operating system, i.e., the operating system of the physical machine where the VM is hosted.

2.7 Continuous System Management

Arguably, one of the most important elements of cloud security is configuration management, including patch management[32]. Since the web will always be prone to

bugs, it is of essence to be prepared to apply security updates and general software updates using configuration management and deployment tools.

2.7.1 Vulnerability Management

Vulnerability management depends on the cloud delivery model and is a shared activity between the CSP and CSC. However, for IaaS and PaaS models it is up to the CSP to perform vulnerability scanning and penetration testing activities. This should be done on a regular basis in order to evaluate the security posture of systems exposed to the Internet.

These activities should be performed using some kind of automated tools. Security auditing tools help automate the process of verifying that a large number of security controls are satisfied for a given system configuration. Combining configuration management and security auditing tools create a powerful combination—auditing tools will highlight deployment concerns while configuration management tools will simplify the process of changing each system to address the concerns highlighted in the audit. Penetration testing tools have been developed to assist in the automation of the process. These are:

1. Zed Attack Proxy (ZAP)[33]
2. OpenVAS[34]
3. SQL Inject Me[35]
4. HTTP Directory Traversal Scanner[36]
5. Burp Suite[37]
6. Qualys SSL Server Test[38]
7. Tamper Data (Samurai WTF)[39]
8. Vega[40]

The testing tools above can be used to conduct a security assessment on own VMs only. However, one should **always** check the terms of service of the selected CSP to determine whether running security tests on the CSP infrastructure is allowed, even if their own machines are the target. If this is not so, either a CSP that allows penetration tests on own VMs must be chosen, or have tests run on a development or testing environment before deployment to the production environment.

2.7.2 Configuration Management

Configuration management allows avoiding the many pitfalls inherent in building, managing, and maintaining complex infrastructures. We should always use tools to automate configuration and deployment. This eliminates human error and allows the cloud to scale much more rapidly[41].

For IaaS deployments we should lock down VMs as securely as possible. Ultimately, we should manage our own VMs—given the exposure level within the cloud, both the application code and the underlying software stack are our responsibility. We should decommission all unnecessary services and applications, remove any unneeded codes, limit user and group access to the bare minimum, and consistently keep systems patched.

For PaaS deployments, CSPs offer a computing platform that could include an operating system, programming languages, an execution environment, a database, and a web server. The management of the components of a PaaS deployment is left to the CSP, who must be able to meet service level agreements (SLA).

2.7.3 Patch Management

Patch management refers to controlling the implementation of fixes so as to resolve the defects/problems identified[42]. One of the reasons why a patch should be deployed can be due to vulnerability management.

A patch manager should keep track of deployed patches while securing the necessary approvals before every patch is deployed. Although automatic updates for operating systems and applications represent a good approach, they can lead to unexpected behaviour and problems. It is therefore recommended to first test these patches in the development/test environment before applying them to the production environment. In addition, as patch deployment can lead to outages one should deploy them in batches to the possible extent, choose lean usage time periods to run the deployment process, and inform users in advance.

Basic steps governing the upgrades and/or patches process, a necessary part of any IT system, are as follows:

(i) Identify the responsible partner to lead the implementation of the requested changes.

(ii) Identify the required changes and outline a “chain of command”, a project plan and a timeline, to test and implement those changes.

(iii) Identify the process for resolving issues introduced by an upgrade, including a clearly defined set of responsibilities and methods for resolving those issues.

(iv) Define a rollback process to restore an upgrade to its initial state should the changes cause unexpected and major failures.

2.8 Decommissioning of Unnecessary Services

An important principle in network security is to only

run the services that are absolutely necessary, thus reducing the ways an attacker might compromise systems. It's important to periodically review the necessity of the services provided and to completely decommission any unnecessary services.

Moreover, insecure services should be avoided wherever possible as they can be exploited by an attacker. Such services include:

- Telnet (use SSH instead)
- Plain FTP
- Open mail (SMTP) relays

Additionally, periodic port-scanning should be used to check for unnecessary services enabled inadvertently, as well as to ensure that services intended for local use only are not made publicly-available, such as:

- File- and print-sharing services (SAMBA, NFS, CUPS)
- Memcached
- Direct database access
- Universal Plug 'n' Play (UPnP)
- Remote Procedure Calls (RPC)
- Simple Network Management Protocol (SNMP)

Finally, one should maintain a list of which services should be made available, while periodically reviewing the necessity of the services provided and restricting or decommissioning those which are not necessary. In addition, one should record any temporarily installed services which will eventually be disabled/decommissioned. Attention should be paid in ensuring (for example by using port scanners) that the decommissioning procedure has actually succeeded.

2.9 Cryptographic Operations

In an IaaS cloud deployment, the CSC deploys computing resources (VMs) from a shared pool of configurable computing resources. This involves the following operations:

(i) Authentication of the offered pre-built images to ensure that they are from authorised sources and have not been tampered with.

(ii) Authentication via the management interface of the hypervisor, needed to launch the VM and to perform subsequent lifecycle operations on that VM (Stop, Pause, Restart, Kill, etc.).

(iii) Secure interaction with the running VM instances.

To assure the integrity of the VM templates the CSP should either[43]:

(i) Digitally sign the templates (with the private key being securely stored at the CSP's premises and protected while in use) using strong (e.g., FIPS 140-2 compliant) algorithms, while the corresponding public key is made available to the CSC in an authenticated manner.

(ii) Use strong cryptographic algorithms, such as AES, RSA public key cryptography, and SHA-256 or better, computed over the VM code, with the corresponding cryptographic hash made available to the CSC in an authenticated manner.

(iii) Use a Hash-based Message Authentication Code (HMAC), using a cryptographic algorithm and a secret key that both the CSC and CSP share.

(iv) Regenerate images on a daily basis.

(v) Allow CSCs to upload their own image templates on the cloud image repository.

To assure the integrity of hypervisor API calls the CSP should implement functionality whereby the VM Management Interface of the hypervisor only accepts and executes authenticated API calls. This is possible via a private/public key pair generation, used for signing the calls submitted to the VM Management Interface. A third-party trusted authority can be used to sign the public key certificate. The certificate is then made available to the VM Management Interface of the hypervisor to verify the signature of the calls submitted by the consumer to the VM instance[43].

To assure the integrity of communication with running VM instances[43]:

(i) A CSC should upload their public keys and inject them in the activated VMs.

(ii) A CSP should deploy a Secure Shell (SSH) protocol. This strong cryptographic authentication prevents anonymous connection attempts to the VM instance, as well as authentication attacks. When SSH is used, not only is the administrator authenticated but all commands, responses, and payloads are protected in both directions from eavesdropping and undetected modifications, in addition to being cryptographically authenticated.

3. Conclusion

When considering a move to cloud computing, we must have a clear understanding of the associated potential security benefits and risks so as to set realistic expectations on our CSPs. Attention must be paid to the different service models (IaaS, PaaS or SaaS) as each model carries different security requirements and responsibilities.

Cloud computing services certification is an impor-

tant aspect—it provides assurance that our critical security requirements are being met. We should therefore identify which security certifications are important to us and insist that our CSP demonstrates their conformance.

Our analysis of the security issues associated with virtualisation lead us to the following principles:

- The VM operating system firewall software should be leveraged to secure the VM.
- Introspection should not be used in public deployment models—host-based security should.
- Regularly check for new updates and apply them accordingly.

Finally, securing our cloud infrastructure means not only implementing controls for the layers we are able to but also auditing our CSP regarding actions taken to lock-down the tenant instances. We must conduct our own analysis of our needs—assess, select, engage, and oversee the cloud services that can best fulfil those needs.

When we talk about securing our cloud infrastructure, part of it is what we are going to do to implement controls for the layers that we have control over and how we can audit the CSP regarding the actions taken to lock-down the tenant instances.

References

1. Wang C, Wang Q, Ren K, *et al*. 2009, *Privacy-preserving public auditing for data storage security in cloud computing*, Cryptology ePrint Archive, viewed February 1, 2016, <https://eprint.iacr.org/2009/579.pdf>
2. Zhu Y, Wang H, Hu Z, *et al*. 2010, *Efficient provable data possession for hybrid clouds*, Cryptology ePrint Archive, viewed February 1, 2016, <http://eprint.iacr.org/2010/234.pdf>
3. *Apache CloudStack*, n.d., viewed January 28, 2016, <http://cloudstack.apache.org/>
4. *Eucalyptus*, n.d., viewed January 24, 2016, <https://www.eucalyptus.com>
5. *OpenNebula*, n.d., viewed January 25, 2016, <http://opennebula.org/>
6. European Commission, 2013, *What does the Commission mean by secure cloud computing services in Europe?*, viewed January 12, 2016, <http://europa.eu/rapid/press-release_MEMO-13-898_en.htm>
7. Raju R V, Vasanth V and Udaykumar P, 2013, Data integrity using encryption in cloud computing, *Journal of Global Research in Computer Science*, vol.4(5): 40–43, viewed January 14, 2016, <http://www.rroij.com/open-access/data-integrity-using-encryption-in-cloud-computing-40-43.pdf>
8. Ali M, 2014, *What is cloud computing stack (SaaS, PaaS, IaaS)*, Maizkglobal Global IT Solutions, viewed January 18, 2016, <http://www.mazikglobal.com/blog/cloud-computing-stack-saas-paas-iaas/>
9. Brenton C, 2012, *Delineation of cloud responsibility*, SANS Information Security Training, viewed October 17, 2015, <https://www.sans.org/cloud/2012/07/02/delineation-of-cloud-responsibility>
10. Lukan D, 2014, *Addressing the most critical cloud security threats*, TechTarget SearchCloudSecurity, viewed January 21, 2016, <http://searchcloudsecurity.techtarget.com/tip/Addressing-the-most-critical-cloud-security-threats>
11. *Choosing a Cloud Provider with Confidence*, n.d., viewed January 4, 2015, <https://www.geotrust.com/resources/whitepapers/choosing-cloud-provider.pdf>
12. Cloud Standards Customer Council, 2015, *Security for cloud computing – 10 steps to ensure success, version 2.0*, viewed May 14, 2015, <http://www.cloud-council.org/deliverables/CSCC-Security-for-Cloud-Computing-10-Steps-to-Ensure-Success.pdf>
13. Rouse M, 2011, *Definition of cloud services*, TechTarget SearchCloudProvider, viewed January 29, 2016, <http://searchcloudprovider.techtarget.com/definition/cloud-services>
14. Cloud Security Alliance, n.d., *Virtualization Working Group*, viewed January 30, 2016, <https://cloudsecurityalliance.org/group/virtualization>
15. Adams D, 2010, *Top 7 threats to cloud computing — part 1*, Patriot Technologies Inc., viewed December 3, 2015, <http://patriot-tech.com/top-7-threats-to-cloud-computing-part-1>
16. Linthicum D, 2015, *Minimize threats through public cloud security testing*, TechTarget SearchCloudComputing, viewed January 20, 2016, <http://searchcloudcomputing.techtarget.com/tip/Minimize-threats-through-public-cloud-security-testing>
17. *FedRAMP*, n.d., viewed September 21, 2015, <https://www.fedramp.gov/>
18. U.K. Government Digital Service, 2013, *The G-cloud framework on the Digital Marketplace*, viewed September 21, 2015, <https://www.digitalmarketplace.service.gov.uk/g-cloud/framework>

19. *Cloud Security Alliance*, n.d., viewed September 13, 2015, <https://cloudsecurityalliance.org/about/>
20. Horrigan B L, 2015, *Securing your cloud deployment*, Information Security — Insider Edition, viewed September 30, 2015, <http://docs.media.bitpipe.com/io_12x/io_121990/item_1101004/ISM_InsideEdition_Securing%20Your%20Cloud%20Deployment_final.pdf>
21. Cloud Security Alliance, n.d., *STAR Certification*, viewed October 12, 2015, <https://cloudsecurityalliance.org/star/certification/>
22. Cloud Security Alliance, *STAR Attestation*, viewed October 12, 2015, <https://cloudsecurityalliance.org/star/attestation/>
23. SSAE-16, n.d., *SSAE SOC 2 report — Trust Services Principles*, viewed November 12, 2015, <http://www.ssae-16.com/soc-2/>
24. Cloud Security Alliance, 2015, *Consensus Assessments Initiative Questionnaire v3.0.1 Info Sheet*, viewed November 15, 2015, <https://cloudsecurityalliance.org/download/consensus-assessments-initiative-questionnaire-v3-0-1/>
25. OpenStack, n.d., *Heat*, viewed May 12, 2015, <https://wiki.openstack.org/wiki/Heat>
26. Brenton C, 2012, *Virtual firewall appliances — trust misplaced?*, Cloud Security Alliance, viewed December 1, 2015, <https://cloudsecurityalliance.org/wp-content/uploads/2012/02/VirtualFirewallAppliances-TrustMisplaced.pdf>
27. Brenton C, 2011, *Hypervisor versus host based security*, Cloud Security Alliance, viewed November 16, 2015, <https://cloudsecurityalliance.org/wp-content/uploads/2011/11/hypervisor-vs-hostbased-security.pdf>
28. Lukan D, 2014, *How to limit security risks during cloud computing virtualization*, TechTarget SearchCloudSecurity, viewed December 2, 2015, <http://searchcloudsecurity.techtarget.com/tip/How-to-limit-security-risks-during-cloud-computing-virtualization>
29. OWASP Open Software Security Community, 2015, *Defense in depth*, viewed November 10, 2015, <https://www.owasp.org/index.php/Defense_in_depth>
30. OpenStack, n.d., *Security Groups*, viewed August 10, 2015, <http://docs.openstack.org/openstack-ops/content/security_groups.html>
31. OpenStack, n.d., *Configure access and security for instances*, viewed May 12, 2015, <http://docs.openstack.org/user-guide/cli_nova_configure_access_security_for_instances.html>
32. Shackleford D, 2014, *Assessing cloud security controls key in repelling cloud attacks*, TechTarget SearchCloudSecurity, viewed November 20, 2015, <http://searchcloudsecurity.techtarget.com/tip/Assessing-cloud-security-controls-key-in-repelling-cloud-attacks>
33. OWASP Open Software Security Community, n.d., *Zed Attack Proxy Project*, viewed March 8, 2016, <https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project>
34. *OpenVAS*, n.d., viewed March 8, 2015, <http://www.openvas.org/download.html>
35. *SQL Inject Me 0.4.5 end-user license agreement*, n.d., viewed March 8, 2015, <https://addons.mozilla.org/en-US/firefox/addon/sql-inject-me/eula/88410?src=collection&collection_id=203cc10a-26b3-5921-12ef-6ba80b06fe07>
36. AutoSec Tools, 2016, *HTTP directory traversal scanner*, viewed March 10, 2016, <http://www.autosectools.com/Page/HTTP-Directory-Traversal-Scanner>
37. PostWigger Web Security, n.d., *Burp Suite*, viewed March 8, 2016, <https://portswigger.net/burp/>
38. Qualys SSL Labs, n.d., *SSL server test*, viewed March 8, 2015, <https://www.ssllabs.com/ssltest/index.html>
39. *Samurai Web Testing Framework*, n.d., viewed March 8, 2015, <https://addons.mozilla.org/el/firefox/collections/rsiles/samurai/>
40. Subgraph, n.d., *Vega vulnerability scanner*, viewed March 8, 2015, <https://subgraph.com/vega/>
41. OpenStack, n.d., *Openstack Security Guide*, viewed February 5, 2016, <http://docs.openstack.org/sec/>
42. Sidana H S, 2012, Cloud computing: managing security. *Infosys Labs Briefings*, vol.10(1).
43. Chandramouli R, Iorga M and Chokhani S, 2013, *Cryptographic key management issues & challenges in cloud services*, National Institute of Standards and Technology, viewed January 27, 2016, <http://dx.doi.org/10.6028/NIST.IR.7956>