4CaaSt: Comprehensive management of Cloud services through a PaaS

Sergio García-Gómez, Manuel Escriche-Vicente, Pablo Arozarena-Llopis.
Telefónica Digital, Spain
e-mail: sergg@tid.es, mev@tid.es, pabloa@tid.es

Francesco Lelli, Yehia Taher
ERISS, University of Tilburg, The Netherlands
e-mail: F.Lelli@uvt.nl, Y.Taher@TilburgUniversity.edu

Christof Momm, Axel Spriestersbach, Jürgen Vogel
SAP Research Center Karlsruhe & Zürich, Germany
e-mail: christof.momm@sap.com, axel.spriestersbach@sap.com, juergen.vogel@sap.com

Andrea Giessmann, Frederic Junker
University of St. Gallen, Switzerland
e-mail: andrea.giessmann@unisg.ch, frederic.junker@unisg.ch

Miguel Jiménez-Gañán
Universidad Politécnica de Madrid, Spain
e-mail: mjimenez@fi.upm.es

József Biro
Nokia Siemens Networks, Budapest, Hungary
e-mail: jozsef.biro@nsn.com

Goulven Le Jeune
Bull, Grenoble, France
e-mail: goulven.le-jeune@bull.net

Michel Dao, Stephane P. Carrie
Orange Labs, Paris, France
e-mail: michel.dao@orange.com, stephanepierre.carrie@orange.com

Joerg Niemoller, Dimitri Mazmanov
Ericsson Research, Sweden
e-mail: joerg.niemoeller@ericsson.com, dimitri.mazmanov@ericsson.com

Abstract— **The 4CaaSt project aims at developing a PaaS framework that enables flexible definition, marketing, deployment and management of Cloud-based services and applications. The major innovations proposed by 4CaaSt are the blueprint and its lifecycle management, a one stop shop for Cloud services and a PaaS level resource management featuring elasticity. 4CaaSt also provides a portfolio of ready to use Cloud native services and Cloud-aware immigrant technologies.**

Keywords-component; Cloud, Platform as a Service, Service Composition

I. INTRODUCTION

Cloud computing is transforming the way applications and services are created, provided and consumed. The virtualization of infrastructures has lowered the barriers of entry such as cost and provisioning time for many providers, especially SMEs [1][2]. Through the virtualization of platforms, SMEs can compete in an almost equal basis with the established players [2].

The European Union-funded 4CaaSt project[1] aims to create an advanced PaaS implementation, which supports the optimized and elastic hosting of Internet-scale, multi-tier applications and enabling the creation of a true business ecosystem [4]. Applications coming from different providers can be tailored to different users, integrated, mashed up and traded together.

In [5], the 4CaaSt value proposition is highlighted as:

- A higher level of abstraction regarding applications and services deployment, hiding the operational complexity while providing a resource efficient solution.
- A broad set of built-in programming libraries, building blocks and specific functionalities, as well as common facilities beyond what is offered and fostered by State-of-the-Art PaaS Clouds, easing development of 'killer applications' showing the value of the 4CaaSt platform.
- An attractive business ecosystem supporting facilities to promote and monetize applications and services, as well as create an active community of users, providers and developers.
- The necessary tools to monitor the execution and manage the lifecycle of applications.

Overall, the project will bring significant benefits to the European economy via a greatly simplified design and delivery model for services and service compositions thus contributing to the establishment of new and highly dynamic and innovative service ecosystems.

[1] http://www.4caast.eu/

This paper focuses on the description of the technical innovations that are being developed in the 4CaaSt project, highlighting the main benefits for service developers and providers through a simplified example scenario. The 4CaaSt platform revolves around the innovative concept of *blueprint*, an abstract description of an application or service that decouples what they offer from the resources required from the various layers of the Cloud stack. The blueprint leverages a great flexibility for the creation, deployment and marketing of applications and services in the Cloud. The different usage models supported by 4CaaSt are presented in Section II. Section III describes the most important innovations of the project. Section IV explains the 4CaaSt validation scenarios and a simplified example to illustrate 4CaaSt platform process, and finally V summarizes the conclusions and the most important benefits to be obtained from the platform.

II. USAGE MODELS

According to the most cited architectural concepts for Cloud computing, Platform as a Service is an important part of Cloud computing architecture. PaaS represents the middle layer connecting the IaaS and the SaaS layer, see for example [3][6]. However, this reflects a very simplified view on Cloud architecture. While 4CaaSt concentrates on the Platform as a Service layer of the Cloud stack, the way in which the project deals with the combination of services from the different layers, benefits from an analysis of how different roles can use them. First of all, the following roles are identified:

- *Service Provider*, who markets deployed services based on existing software, e.g. for SaaS, PaaS, IaaS, and any other XaaS layer.
- *Software Provider*, who provides new application software or platform (middleware) ready to be deployed on the IaaS/PaaS layer.
- *Customer*, who contracts a software to be deployed on Cloud resources, or any service in general (SaaS, PaaS, IaaS).

Beyond those external users, it must be taken into account that there is also a Cloud platform manager. The different cloud stack layers can be used and combined in different ways by the roles specified above, leading to an abundance of deployment scenarios for applications and services over Cloud Computing resources.

Figure 1. shows three examples of scenarios of such deployments. *Application over a platform (a)*: A software provider can develop an application (e.g., a content management system) and publish it so that it can be deployed using some platform software (a web container, a RDBMS…) offered as a service (PaaS). *SaaS over a platform (b)*: A service provider can develop and/or deploy an application (e.g.: a billing application) on top of several platform products to offer a service to any external customer that contracts it (SaaS model). *Pure platform over IaaS (c)*: A development company can contract several platforms (e.g. a web container, a RDBMS, etc.) to be deployed over an IaaS and use them during their own developments. They can also contract PaaS services (e.g., a Non-SQL Data Store) as part of the development/deployment environment of their applications.

Since the automatic provisioning and management of resources is one of the key features of Cloud Computing, managing the end-to-end lifecycle of any of those scenarios is a challenge. The innovative concept of the 4CaaSt *blueprint,* described in the following, in combination with the marketplace functionality and the integrated management of software, services and resources, enable the reification of many usage models and business models as described above. This flexibility constitutes a key differentiator of the 4CaaSt platform in relation to major competitors both from the market and academia.

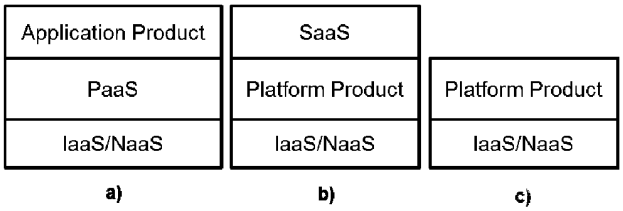

Figure 1. Examples of cloud usage models.

III. THE 4CAAST INNOVATIONS

A. Blueprints

Cloud Blueprinting is a powerful solution that aims at providing next-generation software developers with significant methods and tools that enable them to easily aggregate, configure and deploy virtual service-based application payloads on virtual machine and resource pools on the Cloud [8].

The long-term benefits of Cloud Blueprinting will address concerns at the heart of the Enterprise of the Future and global service marketplaces by:

- Enabling novel geography spanning, end-to-end service applications to be built.
- Encouraging innovation through novel integrative service/Cloud development.
- Empowering service developers to better meet changing application requirements and develop customized service applications.
- Allowing new, innovative business models to be developed through the use of "on-demand" service platforms, infrastructure and supporting services.

To achieve its aim, Cloud Blueprinting promotes autonomous services – at all levels of the Cloud stack – that adhere to the same principles of separation of concerns to minimize dependencies. This solution allows any service at any layer to be appropriately combined with a service at the same level of the Cloud stack or swapped in or out without having to stop and modify other components elsewhere. At the same time Cloud Blueprinting allows multiple (and possibly composed) resource/infrastructure or implementation options for a given service at the application-level. This enables forming service aggregations on demand

at any level of the Cloud stack that may potentially involve various SaaS/PaaS/IaaS providers by breaking up the current SaaS/PaaS/IaaS monolithic approach.

After having studied relevant literature [18], a proliferation of solutions for Cloud service development has been observed [12][13][14][15][16]. But, such methods have clearly shown considerable shortcomings to provide an efficient solution to deal with important aspects related to Cloud service-based applications. Some of these aspects are the *elasticity* and *multi-tenancy* of SaaS applications used to compose service-based applications. Current Cloud service offerings are often provided as a monolithic one-size-fits-all solution and give little or no opportunity for further customization. As a result, these stand-alone Cloud service offerings are more likely to show failure in meeting the business requirements of several consumers due to a lack of flexibility and interoperability.

The Cloud blueprinting approach introduces a series of Blueprint templates used to abstract and describe the components of Cloud Blueprinting-based applications. The use of templates provides a fast and simplified method for provisioning and automating Cloud services. It can be seen as a way for providing an understanding of the features used to deliver reliable and scalable Cloud deployments, and achieving better interconnection between physical and virtual infrastructures. To better manage Blueprint templates, the Blueprint framework interlaces several inter-related components [9]: (1) a declarative Blueprint Definition Language (BDL) that provides the necessary abstraction, constructs to describe the operational, performance and capacity requirements of Cloud services; (2) a Blueprint Constraint Language (BCL) that specifies any explicitly stated rule or regulation that prescribes any aspect of Cloud service; (3) a Blueprint Manipulation Language (BML) which provides a set of operators for manipulating, comparing and achieving mappings between blueprints, and (4) a simple Blueprint Query Language (BQL) for querying collections of blueprints.

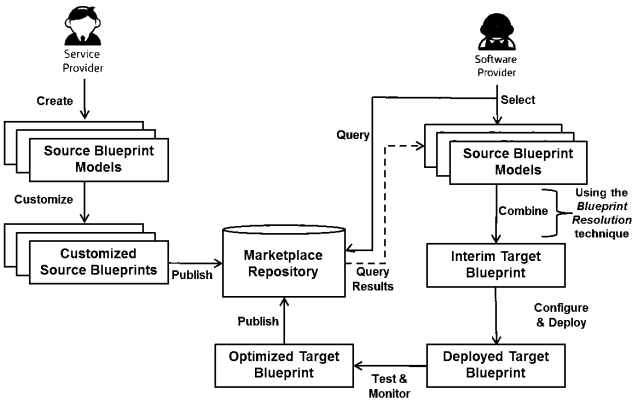

Figure 2. Blueprint support for the Cloud service lifecycle

The Blueprint model helps managing services when they transit through lifecycle stages: design, deployment, testing and monitoring. As illustrated by Figure 2. , after a provider has created the components of a service, the software provider begins the process of making it available to Cloud consumers by creating a source Blueprint model that defines the content of, and the interface to the service. Initially, during design each provider describes all relevant aspects of an offered service in a structure called a *source blueprint*. A service provider (might be distinct from the software provider) customizes the source blueprint templates to create a service offering for consumption by one or more consumers.

During design, an interim *target blueprint* model is created by combining a set of source blueprint models that a developer has selected. Combining source blueprints to satisfy the functional and non-functional requirements of the target blueprint relies on the blueprint resolution technique. A Cloud service developer normally starts designing a new and unresolved target blueprint that captures his to-be services. During the resolution process, in order to fulfill all resource requirements in the target blueprint, he relies on the offerings of other available third-party source blueprints that can be queried and purchased from the marketplace. The blueprint resolution is an iterative process that ends up with a set of blueprints that fulfill all the requirements and constraints needed for actual deployment; the whole solution is referred to as *abstract resolved blueprint*. Subsequently, a deployment plan with configurability points is generated. This plan drives platform resources and virtual machine placement and network configuration.

B. Cloud eMarketplace

4CaaSt provides a cloud *One Stop Shop* marketplace that supports the trading of all types of XaaS (SaaS, PaaS, etc.) services in a unified way. A single access point to all type of service offerings such as SaaS, PaaS, IaaS, or NaaS will be provided. A tight relationship with the 4CaaSt service engineering layer (by means of blueprints) supports the specification of commercial offerings for any type of service, using the most appropriate revenue models from the available ones, for each specific service: different price models (pay per use, subscription, etc.), revenue sharing, advertisements, etc.

The 4CaaSt marketplace may support different usage models as we discussed in Section II: the service providers may contract resources, platforms and even software (or deploy their own software) and sell services by themselves or through the 4CaaSt marketplace, while the customers may contract access to running services. Software developers may enable applications to be contracted and deployed on demand. Finally, service customers may contract a private instance of an application that has to be deployed. All these usages and combinations are supported by the 4CaaSt blueprint and its lifecycle. This behaviour enhances the flexibility of software and service providers to realize multiple business models in 4CaaSt. It also allows the 4CaaSt platform providers to foster truly dynamic business ecosystems through the support of multiple price models and settlement/revenue sharing mechanisms. An increased offering of services and applications can therefore be expected for end- or corporate users.

After analysing the most prominent Cloud services marketplaces (Windows Azure Marketplace, SuiteApp, Zoho Marketplace, Google Apps Marketplace, Google Play (formerly known as Android Market), and Force.com AppExchange) , it has been realised than no other platform supports the full suite of functionalities envisioned by the 4CaaSt integrated marketplace [17]. The most comprehensive offering is provided by Force.com, although it only allows the trading of applications based on their exposed development APIs. Furthermore, it has been detected a trend for IaaS providers to include PaaS capabilities in their offerings, or SaaS providers to enable the application development over their platform [19]. However, no other platform supports in the same way a combined offering of services in the different levels of the Cloud stack. This can therefore be considered as a unique feature of Cloud marketplaces.

On the other hand, most currently established Cloud service marketplaces attempt to adapt their offerings to the customer. However, the peculiarities of the individual users are hardly taken into account, if at all [7]. The social enhancement of the 4CaaSt marketplace brings a number of benefits. For example, individualization based on social data will distinctly improve the user experience, social enhancement can support or even replace the statistical estimation approaches currently employed by market analysis tools in other marketplaces, and the relevance of search results can be increased by using social data.

The 4CaaSt marketplace trades products based on a single blueprint. Every product has an associated price (based on the Universal Service Modelling Language[2] for pricing standard), either defined by the service provider in advance or dynamically after the blueprint resolution by considering the actual aggregated services of the solution. The resolution and the contracting processes are driven by a set of attributes that can be customized both by the service provider and by the customer.

Based on this model, 4CaaSt manages all the phases of a marketplace: information of products and stakeholders; negotiation and resolution of products; contracting and settlement of services; and money flows. Furthermore, the marketplace is tightly integrated with the 4CaaSt platform so that information sources such as monitoring data from the service execution can be leveraged by the marketplace.

C. *PaaS Deployments and Elasticity*

Using the 4CaaSt blueprints, application providers can choose across different platform layers and services to run their applications, including different infrastructure, middleware, and applications components/services. Based on this selection, 4CaaSt is able to automatically generate suitable deployment designs and automatically provision the corresponding resources. Resource provisioning includes selection of the most appropriate deployment designs considering resources, QoS levels, as well as scalability requirements, the automated technical construction and configuration of virtual machines, and their final deployment.

Current PaaS offerings, like the Google App Engine, Amazon Beanstalk, Force.com and Windows Azure rely on a dedicated, homogenous set of infrastructure resources and middleware components. They are able to automatically provision resources but are limited to their choice of technology. 4CaaSt adds support for heterogeneous environments and as a consequence, eases switching between alternative components to avoid vendor lock-ins.

The blueprint resolution process described in Section III.A is the input for the deployment process of resources in the 4CaaSt platform. 4CaaSt in the first step generates applicable deployment designs based on generic deployment templates and architectural constraints defined within the resolved blueprint. In the second step, the 4CaaSt PaaS resource manager performs the actual resource allocation, configuration, and deployment.

Within the resource allocation step, additional meta-information for the Application Components (ACs), middleware components and infrastructure services regarding performance and resource consumption is used to determine the optimal initial deployment design and resource allocation.

Once the initial deployment is decided, 4CaaSt constructs and configures all required VMs together with the Platform Instance Components (PICs) (e.g.: PostgreSQL or JOnAS) for deploying the selected design. These are called Runtime Execution Containers (RECs) in 4CaaSt terminology. To this end, blueprints also have associated technical configuration scripts for installing and configuring the specified component. 4CaaSt is able to "compose" these scripts for the complete stack for a single VM and to orchestrate the construction and "wiring" of landscapes.

Once an application has been deployed, 4CaaSt automatically manages the dynamic scaling of the platform products and the application components deployed on those products. The elasticity mechanism is provided by two means: on the one hand, the customer is able to define a set of elasticity rules based on a number of KPIs that will govern how the application scales *vertically* and/or *horizontally* based on their preferences. On the other hand, the platform can take automatic control of an application using the monitoring data and performing an advanced analysis over them to decide which part of the multi-tier application should scale and how.

D. *Other innovations in a nutshell*

Apart from the described major innovations, 4CaaSt is working in many other advanced features. This section highlights some of them:

- *Network as a Service:* The envisioned 4CaaSt NaaS support provides the possibility to define virtual networks for the deployed applications both at Layer 3 and Layer 2. The clients can customize the defined networks (private or public, IP range, VLANs, QoS requirements like throughput or latency) and dynamically assign virtual machines to the defined networks.

[2] http://www.w3.org/2005/Incubator/usdl/XGR-usdl-20111027/

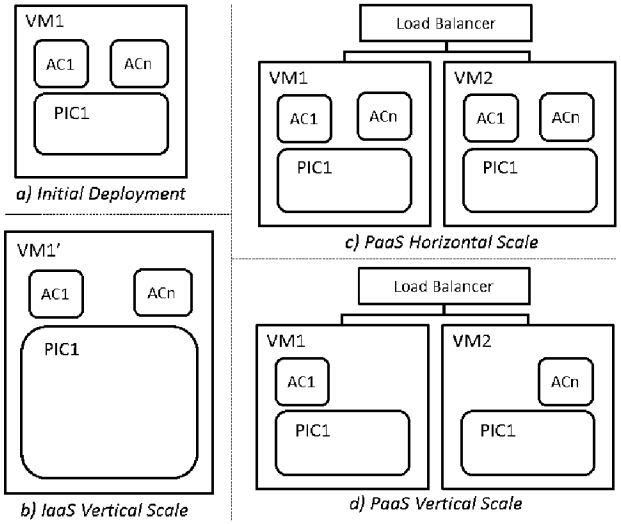

Figure 3. Various scalability mechanisms. The figure represents a) an initial (single VM) deployment with one platform product and several application components (ACs) . The subsequent diagrams represent the same application where the IaaS layer has vertically scaled b), or the PaaS layer has scaled horizontally c), or vertically d).

- *Native Service Technologies* include technologies developed specifically for the Cloud or that happen to be very suitable for it; therefore they are likely to be involved in Cloud-oriented applications. 4CaaSt will offer several native Cloud technologies as services, such as Context aaS, Network Enablers aaS or Cloud Data Store Capabilities that can be used and deployed in an on-demand fashion according to a client's workload and requirements. These technologies are packaged as services offered by 4CaaSt in a uniform fashion, following the concept of technology *enabler*, and thus offering a common interface used by the platform in different phases of applications lifecycle, namely deployment, configuration, monitoring, management and billing.
- *Immigrant PaaS technologies* refer to a set of tried and proven technologies that were available before the advent of Cloud computing and now need to be adapted for the new era. These technologies form a series of building blocks for the 4CaaSt platform and can be used on demand, either in conjunction or independently, when building applications and services based on the 4CaaSt platform. 4CaaSt focuses on the provisioning of infrastructure components for composite applications and services, and in particular on providing Cloud-ready databases, application servers and service composition engines, working when required in tandem through an integration layer that handles the communication with services external to the 4CaaSt platform. Instead of simply deploying these technologies on a Cloud environment, 4CaaSt works towards making them Cloud-aware, offering out-of-the-box capabilities like multi-tenancy, scalability and PaaS-oriented management and configuration. These technologies can be offered as installable software, or as services, similar to the Native Service Technologies.

IV. VALIDATION

4CaaSt requirements and business view are being validated through three different scenarios:

- A marketplace for SMEs' applications.
- A marketplace for mass market applications.
- A hybrid cloud.

These three scenarios are implementing application prototypes using some of the native cloud services and immigrant technologies provided in 4CaaSt, and are being integrated with the overall 4CaaSt platform in order to evaluate the different features provided: blueprints and resolution, revenues settlement, deployment designs, elasticity, etc.

In order to illustrate the value proposition of 4CaaSt platform and validate the proposed approach of the project, this section describes a simple scenario where the main actors are involved, and their interactions with the system depicted:

1. The *PaaS (4CaaSt) provider* wants to have multiple technologies (software or services) in a 4CaaSt platform so that developers have a wider range of technological options to develop their applications. For instance, Context as a Service or a PostgreSQL-based service. For each of them, he must provide a blueprint with the offering and the requirements, and publish it in the marketplace according to a price model.
   External *Service Providers* could follow the same approach in order to provide services in a 4CaaSt compliant Cloud Provider, enabling them to do business in the 4CaaSt ecosystem.
2. The *Software provider* creates the application using his preferred tools (e.g., Eclipse). When he wants to integrate an existing service, he can browse through the 4CaaSt One Stop Shop in order to get information and download client libraries or other artifacts. He could also contract virtual machines or platform services (like a Tomcat deployment) to deploy and test his own application. Once the application is ready, he can create and publish the application blueprints and artifacts.
3. The application is now available and the 4CaaSt platform has all the information required to deploy and manage it. A *Service provider* can create a commercial offer of the application in the marketplace. He selects the application blueprint and defines the price model and the application is ready to be contracted. Any *Customer* will be able to obtain his own instance of the application.
4. When the *Customer* logs into 4CaaSt marketplace, he can select one out of many existing application and services and customize it if such action is allowed for the service.

5. Given the functions, price, performance, availability, or even the customer profile, the *4CaaSt Blueprint Resolution process* decides the best combination of services (blueprints) to deploy or provision for this particular *Customer*. *4CaaSt platform* decides also about which Cloud resources to use, including computing, networking, platforms and services. The application is deployed and the *4CaaSt platform* starts monitoring and accounting its usage.
6. The monitoring information is used by the *4CaaSt platform* to take runtime elasticity decisions according to the constraints defined by the *Customer*.
7. Finally, the *4CaaSt marketplace* charges the *Customer* and distributes incomes among the various *Service providers* involved, including revenue sharing policies when they are defined.

V. CONCLUSION

The 4CaaSt project aims at creating an innovative framework for creating, marketing, deploying and managing applications on the Cloud, both over platform products and platform as a service. 4CaaSt introduces the concept of blueprint, a technical description of an application or a service that decouples the various dependencies it has along the Cloud layers.

Thanks to the blueprint innovation and how applications and services are traded and provisioned/deployed, 4CaaSt may support multiple usage and business models, giving to software and service providers the flexibility to use the resources and services as they prefer.

As a summary, 4CaaSt allows software and service providers to focus on their business (both the software and the service monetization), leaving the underlying complexity of infrastructure and platforms out of their concerns.

To date, a first release of the platform has been delivered, implementing the main features for executing an end-to-end process, and the scenarios are being integrated and validated. Moreover, the focus of the future work will be in some of the most advanced and challenging features described in this paper, especially those related to PaaS elasticity and blueprint resolution, which will represent 4CaaSt unique value proposition.

REFERENCES

[1] Gartner. PaaS Road Map: A Continent Emerging (pp. 1 - 16). 2011. Stamford, Connecticut, USA.

[2] IDC.. Market Analysis - Worldwide Public Application Development and Deployment as a Service: 2010 – 2014 Forecast (pp. 1 - 31). Framingham, MA, USA. 2010.

[3] Mell, P., & Grance, T. The NIST Definition of Cloud Computing. Nist Special Publication (Vol. 145, pp. 1-2). Gaithersburg, Maryland: National Institute of Standards and Technology, Information Technology Laboratory. 2011. Retrieved from http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_Cloud-definition.pdf

[4] Moore, J. F. Predators and Prey: A New Ecology of Competition. Harvard Business Review, May-June 1993 (93309), 75 – 86.

[5] Oliveros Díaz, E., García Gómez, S. 4CaaSt Value Proposition - A 4CaaSt Whitepaper. October 2011 (pp. 1-17). Brussels. Available at http://www.4caast.eu.

[6] Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. A Break in the Clouds: Towards a Cloud De nition. ACM SIGCOMM Computer Communication Review, 39(1), 50-55. 2009.

[7] M. Sarkar, B. Butler, and C. Steinfield, "Cybermediaries in Electronic Marketspace: Toward Theory Building," Journal of Business Research, vol. 41, no. 3, pp. 215–221, 1998.

[8] Michael P. Papazoglou, Willem-Jan van den Heuvel: Blueprinting the Cloud. IEEE Internet Computing 15(6): 74-79 , 2011.

[9] Amal Elgammal, Oktay Türetken, Willem-Jan van den Heuvel, Mike P. Papazoglou: Root-Cause Analysis of Design-Time Compliance Violations on the Basis of Property Patterns. ICSOC 2010: 17-31

[10] J. Cáceres, L. M. Vaquero, L. Rodero-Merino, A. Polo, and J. J. Hierro. Service Scalability over the Cloud. In B. Furht and A. Escalante, editors, Handbook of Cloud Computing, pages 357–377. Springer US, 2010.

[11] D. Moran, L.M. Vaquero, F. Galan "Elastically Ruling the Cloud: Specifying Application's Behavior in Federated Clouds", 4th IEEE International Conference on Cloud Computing (CLOUD 2011), July 2011, Washington DC, USA.

[12] Mietzner, R.: A method and implementation to define and provision variable composite applications, and its usage in Cloud computing. Dissertation, Universitaet Stuttgart, Fakultaet Informatik, Elektrotechnik und Informationstechnik, Germany, August 2010.

[13] Galan, F., Sampaio, A., Rodero-Merino, L., Loy, I., Gil, V., Vaquero, L.M.: Service specification in Cloud environments based on extensions to open standards. In: Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE. COMSWARE '09, New York, NY, USA, ACM, 2009 19:1-19:12

[14] Chapman, C., Emmerich, W., Marquez, F.G., Clayman, S., Galis, A.: Software architecture definition for on-demand Cloud provisioning. In: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing. HPDC '10, New York, NY, USA, ACM, 2010, 61-72

[15] Wei-Tek Tsai; Xin Sun; Balasooriya, J., "Service-Oriented Cloud Computing Architecture," Seventh International Conference on Information Technology: New Generations (ITNG), vol., no., pp.684-689, 12-14 April 2010.

[16] Hyun Jung La and Soo Dong Kim: A Systematic Process for Developing High Quality SaaS Cloud Services. In *Proceedings of the 1st International Conference on Cloud Computing* (CloudCom '09). Springer-Verlag, Berlin, Heidelberg, 278-289, 2009.

[17] García-Gómez, S. (editor) D3.1.1 "Marketplace Functions: Scientific and Technical Report", 4CaaSt Project Deliverable D3.1.1, 2011. Available at http://www.4caast.eu.

[18] Lelli, F. (editor) D2.1.1 "Blueprinting the Cloud: Scientific and Technical Report", 4CaaSt Project Deliverable D3.1.1, 2011. Available at http://www.4caast.eu.

[19] Giessmann, A. (editor). D9.1.2"Market and Competition Analysis", 4CaaSt Project Deliverable D9.1.2, 2012. Available at http://www.4caast.eu.