Advanced Service Brokerage Capabilities as the Catalyst for Future Cloud Service Ecosystems

Anthony J. H.Simons
Department of Computer Science,
University of Sheffield
Regent Court, 211 Portobello
Sheffield S1 4DP, UK
+44 114 2221838
a.j.simons@sheffield.ac.uk

Konstantinos Bratanis,
Dimitrios Kourtesis
South East European
Research Centre
24 Proxenou Koromila
54622, Thessaloniki, Greece
+30 2310 253477
{kobratanis, dkourtesis}
@seerc.org

Iraklis Paraskakis,
Simeon Veloudis
South East European
Research Centre
24 Proxenou Koromila
54622, Thessaloniki, Greece
+30 2310 253477
{iparaskakis, sveloudis}
@seerc.org

Yiannis Verginadis,
Gregoris Mentzas
Institute of Communication and
Computer Systems,
9 Iroon Polytechniou
15773, Zografou, Greece
+30 2107 722489
{jverg, gmentzas}@mail.ntua.gr

Simone Braun
CAS Software AG
CAS-Weg 1-5
76131 Karlsruhe, Germany
+49 721 9638 137
simone.braun@cas.de

Alessandro Rossini
SINTEF
P.O. Box 124, Blindern
0314 Oslo, Norway
+ 47 2206 7633
alessandro.rossini@sintef.no

ABSTRACT
Market analysts have foreseen the emergence of cloud brokers in the mediation of cloud services. But rather than focus on current kinds of intermediary role, it is more constructive to consider the kinds of *brokerage capability* that could be offered in the future, which go far beyond the integration, aggregation and customization services available today. This paper identifies advanced capabilities for cloud service governance, quality assurance and optimization that will be critical in catalyzing the emergence of *cloud service ecosystems*, environments in which all parties will find their symbiotic niches. It shows the path whereby a platform provider could evolve to become the hub of a cloud service ecosystem, through gradually taking on more of these advanced brokerage capabilities. The paper provides an overview of work conducted by the EU FP7 *Broker@Cloud* project towards realizing these advanced brokerage capabilities.

Categories and Subject Descriptors
K.6.2 [**Installation Management**]: Performance and Usage Management, Pricing and Resource Allocation; K.7.3 [**Testing, Certification and Licensing**]; C.2.4 [**Distributed Systems**]: Cloud Computing

General Terms
Management, Measurement, Performance, Economics, Reliability, Standardization, Legal Aspects, Verification.

CrossCloud Brokers'14, December 8, 2014, Bordeaux, France.

Keywords
Cloud Computing, Cloud Service Broker, Intermediary, Platform-as-a-Service, Software-as-a-Service, Infrastructure-as-a-Service, Governance, Optimization, Failure Prevention, Recovery.

1. INTRODUCTION
Enterprises are increasingly moving their IT environments into the cloud, reducing operating costs by converting from a business model reliant on hardware and software ownership, to one based on utility service consumption. The interwoven mixture of infrastructure, platform and application services is often sourced from multiple providers, spanning not only different technologies and geographies, but also different domains of ownership and control, making the strategic and operational management of the cloud-based IT landscape a challenging exercise.

Technology analysts such as Gartner and Forrester foresee an increasing role for *cloud service brokers*, intermediaries who already offer related brokerage capabilities such as integration, customization or aggregation of software services. As more providers join the market, the notion of a *cloud service ecosystem* will emerge, in which software services from different providers are layered in tiers, with more complex services consuming more basic services. This trend can already be seen in the embedding of popular generic heat-map and calendar applications in enterprise-level systems. In this greatly enlarged cloud market, cloud service brokers will expand their activities to include service discovery, aggregation, integration, customization, quality assurance, optimization and governance. It is foreseen that the added value of continuous quality assurance and optimization will be most valuable for service consumers.

This paper deconstructs unhelpful arguments about broker roles, preferring to identify instead a set of advanced brokerage capabilities. It outlines the shape of future software platforms and operating models that, by adopting these capabilities, will bring about a successful transition to the cloud service ecosystem.

Platforms endowed with the projected advanced mechanisms for continuous quality assurance and optimization of cloud services will bring multiple benefits in terms of: added value to consumers, a gentle pressure towards standardization, higher quality of offerings and increased confidence in the ecosystem.

The rest of the paper is structured as follows. Section 2 examines critically the role of the cloud service broker. Section 3 describes how future cloud ecosystems will evolve, and how brokered capabilities will be integrated into future platforms. Section 4 outlines progress in realizing advanced capabilities required for continuous quality assurance and optimization. Before the conclusion, related work is presented in Section 5.

2. DEFINING FUTURE CLOUD SERVICE BROKERAGE CAPABILITIES

Technology market analysts Gartner Inc., Forrester Research and the US standards body NIST have failed in the past to agree on what is understood by the term *cloud broker*, or *cloud service broker*. For Gartner, a service broker is any person, or technology acting as an intermediary, to bring added value to a customer's use of a service [1]. Furthermore, such an agent must have a direct contractual relationship with consumers of cloud services, to qualify as a cloud service broker [2]. A broker performs brokerage, which is defined as intermediation between consumers and providers.

According to this liberal definition, Gartner highlights three distinct value propositions that already qualify as brokerage (and admits further roles are possible). The first kind of brokerage is *aggregation*, the grouping and delivery of multiple services to consumers, offered by various providers. This is seen simply as a matter of scaling, where bundled services are offered through normalized discovery, with a single point of access and billing. The second kind of brokerage is *integration*, where independent cloud services and ERP systems are made to work together, either by vertically integrating cloud services with commercial back-end systems, or integrating across clouds (such as synchronizing between Gmail and salesforce.com). The third kind of brokerage is *customization*, where the broker adds value to the capabilities of the cloud service (for example, by offering custom analytics with a service) [1].

Some argue that Gartner's views of brokerage are too narrowly shaped by the constituencies that pay for its research: software distributors, system integrators and independent software vendors (ISVs), whose roles correspond serendipitously to the above [3]. By contrast, Forrester Research argues that a cloud intermediary has to offer a far more complex value proposition in order to qualify as a cloud service broker [4]. Forrester regards brokerage as encompassing all three of the traditional business models offering infrastructure, consulting and software, but relates these to the levels of the cloud stack. *Simple cloud brokers* provide virtual sourcing of infrastructure (cf. IaaS); whereas *full infrastructure brokers* provide dynamic sourcing across public private cloud infrastructure (cf. PaaS); and *SaaS brokers* provide software services and integration, with unified billing and contract management (cf. SaaS). By this yardstick, no cloud service brokers exist yet in the marketplace [4].

The US National Institute of Standards and Technology (NIST) attempts to clarify the purview of the cloud broker, by scoping the limits of its role. Important actors in the cloud include the *consumer*, the *provider*, the *broker*, the *auditor* and the *carrier* [5]. Whereas the *broker* is seen as an intermediary negotiating between providers and consumers to manage the delivery of cloud services, the *auditor* takes on responsibility for cloud security and compliance with relevant legislation and ethical practice. While security is probably the most important concern for enterprises in the cloud, this topic is explicitly outside our remit (in the EU FP7 *Broker@Cloud* project; but is covered in other EU FP7 projects, such as *Optimis*), so we follow NIST's classification in the present discussion. NIST recognizes three kinds of brokerage (departing again from Gartner and Forrester). *Intermediation* is where a broker enhances services with added value (e.g. by providing unified access, identity management, or performance reporting). *Aggregation* is where the broker composes complex services out of simpler services, ensuring secure data movement between the component services. *Arbitrage* is an extension of this idea, where the broker selects dynamically from multiple service offerings (e.g. based on external credit scoring) to ensure an optimal set of services.

In resolving these conflicting definitions [6], we find it less useful to pigeon-hole specific *kinds of broker* (cf. Gartner), but more relevant to classify the *kinds of brokerage* that future cloud intermediaries may perform, respecting that a given entity may choose to fulfil fewer, or more of these functions, including:

- service discovery – a single point of access to multiple services offered by different providers;
- service integration – the vertical coupling of cloud services to ERP systems, or vertical connection of cloud services from different providers across the layers of the cloud stack;
- service aggregation – the simple bundling, or more creative composition, of cloud services to provide attractive consumer-facing packages, with a single point of access, identity management and billing;
- service customization – the extension or adaptation of generic cloud services to provide added value for bespoke customers, with mechanisms to allow and regulate the participation of ISVs;
- service optimization – the monitoring of service cost and performance, to offer arbitrage to consumers, who may select from alternatives according to pre-declared preferences;
- service quality assurance (QA) – encompassing service lifecycle governance, service certification, service monitoring for failure prevention and recovery.

There exist cloud service intermediaries in the market who already offer capabilities such as integration, customization or aggregation brokerage; but in the future, it is clear that brokers will offer more sophisticated capabilities going far beyond what is available today. In practice there are two business models that brokers can adopt, depending on whether they are pure *intermediaries*, or also *providers*. An analogy may be drawn with online travel agencies. The first kind provides search and cost comparisons, but redirects the tourist to book with the primary travel agent(s). The second kind offers a complete booking service directly to the tourist, hiding the details of the primary operators that it uses.

The *passive* broker is a pure intermediary, offering mostly discovery and integration services, with limited aggregation (bundling only). This broker could marry a 3rd party platform to some infrastructure (cf. Heroku running on Amazon AWS), or seek a 3rd party platform on which to deploy software services selected from a catalogue. Customers will mostly be end-users seeking a suitable service package, or ISVs, seeking suitable platform outlets for their software services. SaaS to PaaS integration is expected to outstrip PaaS to IaaS integration, by volume of business.

The *active* broker will also provide, manage or license a platform that is capable of offering more, or all of the capabilities described above. This could integrate monitoring mechanisms on which optimization, failure prevention and recovery will depend. It could also integrate service governance mechanisms on which service certification and the regulation of ISV custom software contributions will depend. Customers as above will also be able to seek, or provide customized services. Again, brokered SaaS is expected to dominate any other kind of brokered product in the cloud, by volume of business.

3. IMAGINING THE EVOLUTION AND OPERATION OF CLOUD ECOSYSTEMS

The following fictional scenario is based on proprietary case study data from companies participating in the *Broker@Cloud* project. It describes how the incremental adoption of different kinds of advanced brokerage capability stimulates the emergence of the anticipated cloud service ecosystem. A CRM system provider *Office Systems AG* has for some years marketed the CRM suite *OfSys* as shrink-wrapped software, but is finding it increasingly hard to sell to new customers, who do not need the whole package, or who want customized parts. They decide to migrate *OfSys* into the cloud, offering its components as separately-billed services on a new cloud platform *OpenSys*, so that consumers may select which services they need, on a pay-per-use basis. *OpenSys* offers a virtual desktop, on which services are selected like apps. Enterprise-level services consume certain common micro-services (e.g. mail, calendar, or heat-map), so *OpenSys* functions as a *bona fide* PaaS.

To host the platform, *Office Systems AG* subcontracts to the infrastructure provider *DataCent SA*, the first partner in the ecosystem. *DataCent* provides elastic data storage and compute power, with internal monitoring and failover. Initially, this is a private cloud, used by *Office System'*s customers.

The business expands to include new kinds of customer, such as theatre agents and music impresarios, who desire new kinds of artist management services. Initially, *Office Systems* undertakes to develop these bespoke customized services for the platform, but finds increasingly that diverting effort from their core business is too disruptive and costly. They outsource the development to different ISVs, new partners in the ecosystem, who are able to respond quickly. The *OpenSys* platform now acts like a service broker, offering a selection of 3rd party services to consumers along with the original *OfSys* CRM services.

Maintaining the high quality of outsourced services becomes a challenge, so the *OpenSys* platform adopts automated policy-driven governance of all offerings on the platform. This ensures that all new services follow the same QA procedures and scrutiny, whether produced in-house or by an ISV. The platform tracks new services through the service lifecycle, enforcing the in-house service engineering rules of *Office Systems*, and the local legal regulatory framework for operation, which are described as sets of policies and interpreted by the governance tools.

The platform also offers automated QA. During the service onboarding phase, services are certified for compliance to the platform's standards. This includes checking the provider's SLAs against the platform's advertised SLAs, and testing the service for functional conformance to an agreed specification. Acting as a broker, the platform solicits and maintains a pool of specifications of the kinds of service offered, which are used as a reference by ISVs and have the effect of applying a gentle pressure towards service standardization. The specifications are checked by tools for consistency and completeness, and functional tests are generated from them automatically, to ensure that both new and upgraded services (still) comply with their specifications.

Eventually, the volume of *OpenSys* business grows to the point that *DataCent* cannot host it all. *OpenSys* bursts onto the public cloud and new instances run on Amazon and Google infrastructures. The monitoring of resource usage and performance is part of the platform's contract with the IaaS providers, since it is important to know when to scale-out, or scale-in, to minimize operating costs. The platform offers failure prevention and recovery mechanisms, such that if *DataCent's* hosted instances drop below a certain performance threshold and the consumer's SLA is about to be violated, affected instances may migrate out into the public cloud. Monitoring allows pro-active adaptation to take place, before failure occurs.

A new ISV partner *Bottom Line plc* joins the ecosystem, to provide a range of related financial accounting and taxation services. *Bottom Line* develops a tax return app, to help the self-employed and SMEs fulfil their tax obligations. This app is a wrapper, integrating vertically with a number of back-end banking services offered by different banks at variable cost. The platform can detect which tax return service is currently the cheapest, and provides arbitrage to its customers, who may specify in advance whether they wish to be notified, or whether the app should switch automatically to a different provider, when the cost reaches a given threshold.

The pressure to provide standardized services matching a common specification eventually leads to many comparable offerings from different ISVs. The platform is then more able to take advantage of opportunities to optimize its offered service recommendations, on the basis of cost, performance or reliability. End-users may express their preferences in terms of exact rules, or fuzzy trade-offs between different service qualities.

4. REALIZING CONTINUOUS QUALITY ASSURANCE AND OPTIMIZATION

The scenario above extended existing brokerage capabilities with new advanced capabilities to support continuous quality assurance and optimization in the gestation and delivery of cloud services. This has been the focus of the EU FP7 *Broker@Cloud* project. We summarize the requirements below, which are reported in full in the project deliverable [6].

1. Brokerage framework interfaces and platform-neutral descriptions of cloud services: a precondition for delivering the other capabilities outlined below is the ability of the brokerage framework mechanisms to interact with the cloud service delivery platform in which they are to be integrated, using appropriate platform-independent interfaces, as well as the availability of platform-neutral methods for description of cloud services.

2. Cloud service governance and quality control: managing the lifecycle of cloud services as they evolve; evaluating services for compliance to policies addressing technical, business and legal aspects of service delivery; continuously monitoring services for conformance to SLAs; repeatedly testing services after creation or modification to certify conformance to specifications and compatibility with expected behavior.

3. Cloud service failure prevention and recovery: reactive and proactive detection of cloud service failures; selection of suitable adaptation strategies to prevent or to recover from problematic situations as these surface; recommendation or (where possible) automated enactment of adaptation actions such as service substitution or renegotiation of service terms.

4. Continuous optimization of cloud services: continuously identifying opportunities to optimize the set of services consumed by an enterprise with respect to different quality goals; exploiting a large number of QoS attributes, such as functionality, agility, availability, cost, performance and usability; ranking of alternative services through a unified multi-criteria decision-making approach using quantitative (precise) and qualitative (imprecise, or fuzzy) measures of service- and provider-characteristics.

Below, we give an overview of progress made towards realizing these goals, referring the reader to the public project deliverables for the full details. In the furtherance of goal (1) above, the complete brokerage framework has now been described in the reference architecture deliverable [7], in which the different reference processes were grounded to the phases of a service's lifecycle, and related to the capabilities above. Each capability was described as a black-box with inputs and outputs. The APIs through which these brokerage components are to be accessed were specified in the REST style [8], also indicating their interdependencies.

Furthermore, an ontological framework for the specification of service descriptions, broker policies and other relevant artefacts such as consumer preferences has been created [9] using minimal extensions to *Linked USDL* [11], a lightweight service description ontology. This is intended to support platform-neutral data exchanges between brokerage-enabled platforms and the materialized brokerage capabilities; and leverages the power of semantic reasoning frameworks for discovery and matching.

In the furtherance of goal (2) above, prototype software has been developed to support a set of mechanisms for cloud service governance and quality control [10]. These are implementations of the reference processes described in [7]. They break down into mechanisms for policy-driven governance, mechanisms to evaluate functional behavior and mechanisms to evaluate non-functional aspects of service delivery (e.g. SLAs).

A governance registry system has been created (on top of the WSO2 platform [12]) that provides service lifecycle management features; this interprets declarative policies expressed in [9], as opposed to offering hardwired governance [13-15], a key advance needed for open standards of governance. Also, an enterprise publish/subscribe system has been built on top of the WSO2 platform to transport service-related artefacts to the various components of the brokerage framework. An ontology-driven tool has been created that checks service descriptions for their compliance against pre-specified policies concerning business and technical (non-functional) aspects of service delivery.

A service specification language based on EFSM models and expressed in XML has been constructed; and tools have been created that interpret this language [16, 17]. Verification and validation tools ensure that the service workflow, expressed as a finite state machine, is consistent with the individual function specifications, and that the latter are consistent, complete and deterministic. Test-generation tools create high-level functional test suites expressed in XML, which may be grounded for specific SOAP, REST or other service implementation technology.

In the furtherance of goal (3) above, prototype software has been developed to support continuous failure prevention and recovery [18]. This consists firstly of a monitoring and analysis prototype based on the WSO2 Complex Event Processor, a lightweight open source CEP server. A CEP engine can derive higher-order events relating to impending service failure from low-level events reporting high CPU loads or slow response times, detected at the infrastructure or platform level. The high-level events are transmitted to the second component, a reasoner which controls how adaptation or recovery is performed, based on the EU FP7 DiVA project's model-based open-source platform for managing the variability of dynamically adaptive systems (DAS). Treating cloud services as a special kind of DAS, the Broker@Cloud reasoner reimplements DiVA to overcome some limitations in performance and flexibility. It interprets declarative rules that express alternative recovery strategies and seeks to maintain the overall Quality-of-Service (QoS).

In the furtherance of goal (4) above, a method has been devised to support continuous optimization of cloud service delivery, based on the fuzzy AHP (analytic hierarchy process [19]) approach. This offers a unified method for multi-criteria decision making, based on precise (i.e. measurable) and imprecise (i.e. fuzzy) decision criteria. Service consumers may express their preferences for service optimization using exact numerical or imprecise linguistic terms, which are known to be closer to human perception in their deliberate vagueness. The framework provides optimal multi-criteria hierarchical decision-making over these metrics, yielding more satisfactory outcomes than traditional service ranking methods [20]. A software prototype called PuLSaR (Preference-based cLoud Service Recommender) is under construction and will implement this approach.

As a means to validate the above, we plan to build two prototype service brokerage platforms that each adopt different selections of the above brokerage capabilities. One will be hosted by CAS Software AG (Karlsruhe), as an extension to the *CAS Open* platform. The other will be hosted by Singular Logic (Athens), as an extension of the *Orbi* platform.

5. RELATED WORK

An earlier survey of the state-of-the-art in relation to cloud service governance and quality control can be found in [21]. Current work in the related field of SOA governance [22, 23] has focused on registry and repository systems that check whether service-related artefacts conform to business policies [13, 22] and a similar approach has also been applied to SaaS [14, 15]. These examples use hardwired checking algorithms integrated with the format of the policies and checked data. They fail to decouple the expression of policies from the checking of data against the policies, a weakness in current tools. We envisage a future in which these concerns are separated [13], by leveraging Semantic Web technologies and Linked Data principles.

There are no industry standards for assuring the quality of software services. SaaS providers follow their own in-house development lifecycle, which typically involves code inspections and limited functional testing. Proposals for explicit service testing methods have largely been interface-based [24, 25]. Some have also sought to capture service execution behavior with graph transformation rules [26], or semantically augmented WSDL [27-30], where the semantics are expressed as UML state machines or OCL pre- and post-conditions. One precursor to our approach [31] showed how EFSMs could be reverse-engineered from protocol specifications with IOPE descriptions (inputs, outputs, preconditions and effects). The resulting EFSM was amenable to the Stream X-Machine method for complete functional testing [32, 33], on which our approach is based [16, 17].

Monitoring and adaptation for service-based systems [34] and cross-layer adaptation and monitoring for service-based applications [35] have been researched as precursors for advanced systems for failure prevention and recovery in the cloud. The extensive state-of-the-art survey [21] identified challenges to be addressed, which include (i) what data should be collected and what metrics used; (ii) how brokers should manage large volumes of events collected from heterogeneous sources; and (iii) what kinds of analysis and prediction techniques should be used to support proactive failure prevention. Architectural issues supporting extensibility, flexibility and dynamic response have received initial consideration [36].

Existing work on service optimization has largely focused on the multi-objective decision methodology [37, 38], rather than the overall goal of satisfying service consumers. This work has considered mainly IaaS and the effects of network bandwidth and virtual machine factors on QoS [39-41] and has not considered the variety of other conditions that can change in a cloud ecosystem. Other ideas include a service recommender system [39] and management of dynamic SLAs [40]. Optimization is based on quantitative evaluation using precise metrics [42], where decision theory would predict that qualitative evaluation with imprecise metrics would achieve a better service ranking [43].

Web services are described at interface-level using WSDL and SOAP; or observe proprietary REST conventions; or use non-standard AJAX streaming to rich client applications. There are various competing standardization efforts for the syntactic and semantic description and dynamic configuration of web services (e.g. OASIS TOSCA [44]). Different fragmentary approaches to describing services as pure interfaces, or with functional semantics, or with more comprehensive non-functional business considerations, culminated in the development of the *Unified Service Description Language* (USDL) [45]. This was intended as a unifying framework, but was later considered too monolithic. Its lightweight successor, *Linked USDL* [11] follows a linked data philosophy, adding service descriptions, SLAs and other business-related aspects to a framework that capitalizes on existing semantic ontologies (MSM; FOAF; GR; SKOS). Our work builds on this open approach; and to the best of our knowledge, there has been no other proposal for interface specifications supporting cloud service quality assurance and optimization.

6. CONCLUSIONS

Cloud service brokerage aims to help enterprises negotiate better deals, when consuming many cloud services from diverse sources. Rather than simply consider brokers as perpetuating the old roles of software distributors, systems integrators and ISVs, we have shown how advanced brokerage capabilities could be adopted gradually by any platform provider seeking to bring increased service quality and choice to consumers. Each step is motivated by a business need and also attracts a new kind of business partner to the service ecosystem, which develops in an organic fashion around the hub of the brokerage platform. Open standards for service gestation and certification will also encourage federation.

Platforms will evolve from marketplaces into sophisticated brokerage engines, offering mechanisms for the certification, continuous quality assurance and optimization of cloud services. These have a high value for consumers, since they increase service reliability, force standardization across service offerings and develop trust in the cloud service ecosystem; yet they are complex and difficult to implement. To this end, the EU FP7 Broker@Cloud project has been investigating the mechanisms required to deliver these advanced brokerage capabilities; and we have reported an overview of the project's progress, giving an index into the more detailed public deliverable reports.

7. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n°328392, the Broker@Cloud project (http://www.broker-cloud.eu).

8. REFERENCES

[1] Gartner Inc., *Cloud Services Brokerage Is Dominated by Three Primary Roles. Gartner, 23 Nov.* 2011.

[2] Gartner Inc., *Defining Cloud Services Brokerage: Taking Intermediation to the Next Level. Gartner, 8 Oct*. 2010.

[3] J. Bloomberg, *Cloud washing the Cloud Brokerage. ZapThink, 11 Nov.* 2012.

[4] Forrester Research Inc., *Cloud Brokers Will Reshape The Cloud - Getting Ready For The Future Cloud Business Models. Forrester, Sep.* 2012.

[5] NIST, *Cloud Computing Reference Architecture, Special Publication 500-292*, Information Technology Laboratory, NIST, Sep. 2011.

[6] Broker@Cloud, *Deliverable 20.3 - Requirements analysis report, Broker@ Cloud Project, Tech. Rep*. 2013. [Online] Available: http://www.broker-cloud.eu

[7] Broker@Cloud, *Deliverable 30.1 - Conceptual architecture for the cloud brokerage framework, Broker@Cloud Project, Tech. Rep.* 2013. [Online] Available: http://www.broker-cloud.eu

[8] Broker@Cloud, *Deliverable 30.3 - Specification of interfaces for enabling brokerage in enterprise cloud service delivery platforms, Broker@Cloud Project, Tech. Rep.* 2014. [Online] Available: http://www.broker-cloud.eu

[9] Broker@Cloud, *Deliverable 30.2 - Methods and tools for brokerage-enabling description of enterprise cloud services, Broker@Cloud Project, Tech. Rep.* 2014. [Online] Available: http://www.broker-cloud.eu

[10] Broker@Cloud, *Deliverable 40.1 - Methods and mechanisms for cloud service governance and quality control, Broker@ Cloud Project, Tech. Rep*. 2014. [Online] Available: http://www.broker-cloud.eu

[11] T. Leidig and C. Pedrinaci, *Linked USDL*, 2013. [Online] Available: http://www.linked-usdl.org

[12] WSO2, *WSO2 Governance Registry*. [Online] Available: http://wso2.com/products/governance-registry/

[13] D. Kourtesis, I. Parakakis, and A. J. H. Simons, Policy-driven governance in cloud application platforms: an ontology-based approach, *Proc. 4th. Int. Workshop on Ontology-Driven Information Systems Engineering*, 2012.

[14] D. Kourtesis and I. Paraskakis, A registry and repository system supporting cloud application platform governance, *Proc. 9th Int. Conf. on Service Oriented Computing, LNCS vol. 7221*, Springer Verlag, 2011, pp. 255–256.

[15] D. Kourtesis and I. Paraskakis, Governance in cloud platforms for the development and deployment of enterprise applications, *Proc. 3rd IEEE Int. Conf. on Cloud Computing Technology and Science,* 2011.

[16] M. Kiran, A. Friesen, A. J. H. Simons, and W. K. R. Scwhach, Model-based testing in cloud brokerage scenarios, in: A. R. Lumesco, et al. (eds.), *Service-Oriented Computing - ICSOC 2013 Workshops, LNCS vol. 8377*, Springer Verlag, 2013, pp. 192-208.

[17] R. Lefticaru and A. J. H. Simons, X-Machine based testing for cloud services, in: G. Ortiz et al. (eds.) *European Conf. on Service-Oriented and Cloud Computing - ESOCC 2014 Workshops, CCIS*, Springer Verlag, 2014 (to appear).

[18] Broker@Cloud, *Deliverable 40.2 - Cloud service failure prevention and recovery components of brokerage framework. Broker@Cloud Project, Tech. Rep*. 2014. [Online] Available: http://www.broker-cloud.eu

[19] T. L. Saaty, *The Analytic Hierarchy Process*, McGraw Hill, 1980.

[20] M. Godse and S. Mulik, An approach for selecting software-as-a-service (SaaS) product. *Proc. IEEE International Conference on Cloud Computing*, 2009, pp. 155–158.

[21] Broker@Cloud, *Deliverable D20.1 State of the art and research baseline. Broker@Cloud, Tech. Rep*. 2013. [Online] Available: http://www.broker-cloud.eu

[22] E.A. Marks, *Service-Oriented Architecture Governance for the Services Driven Enterprise*, John Wiley, 2008.

[23] L. J. Zhang and Q. Zhou, CCOA: cloud computing open architecture, *Proc. 7th IEEE International Conference on Web Services*, New York, USA, 2009, pp. 607–616.

[24] X. Bai, W. Dong, W. Tsai, and Y. Chen, WSDL-based automatic test case generation for web services testing, In Proc. *IEEE Int. Workshop Service-Oriented System Eng.*, Beijing, China, 2005, pp. 215-220.

[25] S. Chakrabarti and P. Kumar, Test-the-REST: an approach to testing RESTful web services, *ComputationWorld 2009: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns.* Athens, Greece: IEEE Computer Society, 2009, pp. 302–308.

[26] R. Heckel and L. Mariani, Automatic conformance testing of web services, In: Cerioli, M. (ed.) *Fundamental Approaches to Software Eng., 2005. LNCS, vol. 3442*, Springer Verlag, 2005, pp. 34–48.

[27] A. Bertolino, I. Frantzen, A. Polini and J. Tretmans, Audition of web services for testing conformance to open specified protocols, in: R. Reussner, J. A. Stafford and C. Szypersky (eds.), *Architecting Systems with Trustworthy Components, LNCS, vol. 3938*, Springer Verlag, 2006, pp. 1-25.

[28] S. Noikajana and T. Suwannnasart, An improved test case generation method for web service testing from WSDL-S and OCL with pair-wise testing technique, In: *Proc. 33rd Annual IEEE Int. Computer Software and Applications Conferences, vol. 1*, Seattle: IEEE Computer Society, 2009, pp. 115-123.

[29] W. T. Tsai, R. Paul, Y.Wang, C. Fan and D. Wang, Extending WSDL to facilitate web services testing, In: *Proc. 7th IEEE Int. Symp. on High Assurance Systems Engineering*, Tokyo, Japan, 2002, pp. 171-172.

[30] R. Heckel and M. Lohmann, Towards Contract-based Testing of Web Services, *Electronic Notes in Theoretical Computer Science, vol. 127, no. 3*, 2005, pp. 101-111.

[31] E. Ramollari, D. Kourtesis, D. Dranidis and A. J. H. Simons, Leveraging semantic web service descriptions for validation by automated functional testing, In: L. Aroyo and P. Traverso (eds.), *Proc. 6th European Semantic Web Conf., LNCS, vol. 5554*, 2009, pp. 593-607.

[32] W. M. L. Holcombe and F. Ipate, *Correct Systems: Building a Business Process Solution, Applied Computing Series*, London: Springer Verlag, 1998.

[33] F. Ipate and W. M. L. Holcombe, An integration testing method that is proven to find all faults, *Int. J. Computer Mathematics, vol. 63*, pp. 159-178, 1997.

[34] M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger, Eds., *Service research challenges and solutions for the future internet: S-cube - towards engineering, managing and adapting service-based systems*. Springer Verlag, 2010.

[35] K. Bratanis, D. Dranidis and A. J. H. Simons, SLAs for cross-layer adaptation and monitoring of service-based applications: a case study, in: D Bianculli, et al., (eds.) *Proc. Int. Workshop on Quality Assurance for Service-based Applications*, ACM Digital Library, 2011, 28-32.

[36] K. Bratanis, D. Dranidis and A. J. H. Simons, An extensible architecture for the run-time monitoring of conversational web services, in: D. Karastoyanova, R. Kazhamiakin and A. Metzger (eds.), *Proc 3rd. Int. Workshop on Monitoring, Adaptation and Beyond*, ACM New York, 2010, 9-16.

[37] H. J. Moon, Y. Chi, and H. Hacigumus, SLA-aware profit optimization in cloud services via resource scheduling, In *Proc. 6th World Congress on Services (SERVICES-1)*, 2010, pp. 152–153.

[38] J. Z. Li, M. Woodside, J. Chinneck, and M. Litoiu, CloudOpt: multi-goal optimization of application deployments across a cloud, *Proc. 7th Int. Conf. on Network and Services Management*, 2011, pp. 162–170.

[39] S.-M. Han, M. M. Hassan, C.-W. Yoon, and E.-N. Huh, Efficient service recommendation system for cloud computing market, *Proc. 2nd Int. Conf. on Interaction Sciences: IT, Culture and Human*, 2009, pp. 839–845.

[40] A. Lawrence, K. Djemame, O. Waldrich, W. Ziegler, and C. Zsigri, Using service level agreements for optimising cloud infrastructure services, in: M. Cezon and Y. Wolfsthal (eds.) *Towards a Service-Based Internet. ServiceWave 2010 Workshops*, 2010, pp. 38–49.

[41] P. Pawluk, B. Simmons, M. Smit, M. Litoiu, and S. Mankovski, Introducing STRATOS: A cloud broker service, *Proc. IEEE 5th International Conference on Cloud Computing (CLOUD)*, 2012, pp. 891–898.

[42] S. K. Garg, S. Versteeg, and R. Buyya, SMICloud: a framework for comparing and ranking cloud services, *Proc. 4th IEEE International Conference on Utility and Cloud Computing (UCC)*, 2011, pp. 210–218.

[43] J. Doyle and R. H. Thomason, Background to qualitative decision theory, *Artificial Intelligence, vol.20, no.2*, 1999, p. 55.

[44] OASIS, *Topology and Orchestration Specification for Cloud Applications Version 1.0, 2013*. [Online] Available: http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html

[45] D. Oberle, A. Barros, U. Kylau, and S. Heinzl, A unified description language for human to automated services, *Information Systems, vol. 38, no. 1*, pp. 155–181, 2013.