Presented at CLOSER 2017 http://closer.scitevents.org/?y=2017

DECIDE: DevOps for Trusted, Portable and Interoperable Multi-Cloud
Applications towards the Digital Single Market

Juncal Alonso[1], Leire Orue-Echevarria[1], Marisa Escalante[1], Gorka Benguria[1]

[1]Tecnalia Research & Innovation, Parque Científico y Tecnológico de Bizkaia, Edificio 700, 48160 Derio, Bizkaia, Spain
{juncal.alonso, leire.orue-echevarria, marisa.escalante, gorka.benguria}@tecnalia.com

Keywords: Self-adaptation, Intermediation and Federation of Heterogeneous Resources, Run-Time Monitoring, Deployment Simulation, Multi-Cloud Architectural Patterns.

Abstract: The main objective of the DECIDE action is to provide a new generation of multi-cloud service-based software framework, enabling techniques and mechanisms to design, develop, and dynamically deploy multi-cloud aware applications in an ecosystem of reliable, interoperable, and legal compliant cloud services. Three use cases will be conducted to validate the proposed approach.

1   INTRODUCTION

The digital transformation from product to service economy means changes in the companies' operating environment: they need to transform into service providers from product providers and be able to flexibly change their role in the value chain and markets. In order to be able to foster the change, the companies IT infrastructure needs to be more flexible. Cloud services enable this to some degree, but as such create dependency to external partners for a company. In a world where new players come, others disappear, and conditions are continuously changing, how can the companies be sure that the architectural decisions that were taken in the past continue to be the best one? For example, while developing or migrating a web site, an organization can decide to build it in a dedicated internal computer, build it as an instance in a shared internal computer, build it in a dedicated external computer, or even build it as an instance in a shared external. The decision on using one, another, or several approaches simultaneously is driven by certain evaluation criteria (e.g. profitability, reliability, performance, security, legal or even ecological aspects). Cloud services providers (CSPs) themselves may fail too, so for the greatest measure of protection possible, an enterprise may wish to embark upon a multi-cloud strategy. There are several multi-cloud solutions available for solving specific problems, but to date, little attention has been paid to distributing the cloud risk, and managing multiple clouds from a single technology platform. Working with many CSPs means managing multiple relationships. Currently, most companies are already in the process of negotiating multiple contracts with multiple CSPs. That involves activities such as to analyze their service level agreements, manage multiple payments, storing passwords, and so on. Those cumbersome activities lead to question whether it would be possible to unify those efforts somehow in order to maximize both the efficiency and effectiveness of the usage of services coming from different CSPs. One way to achieve that is through Cloud Service Broker (CSB). Gartner defines a cloud services brokerage as a third-party software that adds value to cloud services on behalf of cloud service consumers (Gartner, 2016). The major benefit of using CSBs is that these allow organizations to focus on other critical day-to-day business needs instead of in how to manage multiple CSPs. As expressed by Gartner, a viable intermediator and federator of cloud services can make it less expensive, easier, safer (also in legal terms), interoperable and more productive for companies to discover, aggregate, consume and extend cloud services, particularly when they span multiple, diverse cloud services providers in different EU Member States (Gartner, 2012)

1.1 Multi-Cloud Applications: Main Challenges and Unsolved Issues

In the context of this paper, a multi-cloud native application is an application distributed over heterogeneous cloud resources whose components are deployed on different CSPs and still, they all work in an integrated way and transparently for the end-user. There are several reasons for deploying an application in a multi-cloud architecture, the most important ones being: non-compliance of the CSPs to the agreed SLAs, avoidance of vendor lock-in, increasing reliability or improving other QoS concerns such as increasing performance or security, and finally, reducing costs. The application types that would benefit the most from such a multi-cloud approach are on the one hand, those that are critical to the business and that need to respond efficiently to the user's needs in terms of performance, reliability and security and on the other hand, complex applications whose components need to be distributed over different cloud providers due to their specific needs and requirements. However, any application offered as SaaS can benefit from a multi-cloud architecture. Currently, this is solved by deploying the same application on several cloud providers following a master-slave or active-passive approach. This, however, poses also several risks, since the synchronization of all the data is critical for a correct functioning of the application if no data loss is wanted. The multi-cloud approach presented in this paper avoids synchronization risks and guarantees the fulfillment of the application providers' requirements, which can range from maintaining a constant cost structure to a certain response time, security issues or a certain performance level.

Developing, deploying and operating multi-cloud applications present the following challenges:

- Applications need to be responsive to hybrid/multi-cloud model scenario, in which an application that is executing in a private cloud bursts into a public cloud when the demand for computing capacity spikes. This implies that the application architecture shall be re-designed to be "multi-cloud" aware simplifying the cloud application assembly and the deployment process. A possible solution to this problem is to deploy components of the same application on multiple clouds distributing the workload among them, so as to continuously guarantee the average QoS requirements, dynamically allocate resources and automatically reconfigure the deployment configuration when a Non-Functional Requirement (NFR) value is not met. Multi-cloud application solutions have to deal with a set of Non-Functional Property (NFP) (e.g. performance, security, availability, governance etc.) of the individual components given their complexity and distribution as well as with the overall NFP of the application including the communications and the data flow between components. Even if each CSP offers means and controls for measuring NFP, the multi-cloud application has to ensure integrated non-functional characteristics across the whole composition. Therefore, the overall non-functional characteristics depend on the NFP of the application components, which in turn depend on the NFP offered by the cloud resources they exploit.

- Means shall be provided to manage and assess cloud deployment alternatives to better support our cloud re-deployment decisions. This implies profiling and classifying application components and cloud nodes, as well as analyzing and simulating the behavior of the application under stressful conditions to support the deployment decision making process considering additional factors such as NFR, namely performance, availability, localization, cost, or risks associated with the change of cloud resources. Multi-cloud has value only when the right providers are selected, whether public or private (combined into different cloud deployment models), to meet functional and NFR.

- Existing cloud services shall be made available dynamically, broadly and cross border, so that software providers can re-use and combine cloud services, assembling a dynamic and re-configurable network of interoperable, legally secured, quality assessed (against SLAs) single and composite cloud services, facilitating the free flow of data and therefore impacting in the Digital Single Market (DSM).

1.2 DevOps for Multicloud Applications: Progress beyond the State of the Art

1.2.1 DevOps

DevOps refers to the emerging professional movement and philosophy that advocates for a collaborative working relationship between

Development and IT Operations, lowering barriers and silo-based teams, resulting in the fast flow of planned work, while simultaneously increasing the reliability, stability, resilience of the production environment. This is often called the "DevOps Paradox" (Edwards, 2015): "Going faster brings higher quality, lower costs, and better outcomes". Organizations such as Etsy, Netflix, Facebook, Amazon, Twitter and Google, but also ING, or Target, by applying the DevOps philosophy, they have been able to achieve levels of performance that were unthinkable even five years ago. DevOps pivots around three axes (UpGuard, 2016): processes, people and technology. From the people perspective, DevOps symbolizes a cultural change where collaboration and cooperation are key pillars, and this often results in an increased understanding to prioritize requests that the business needs. From the processes perspective, DevOps advocates for a more agile change processes, with an increased rate of acceptance for new features, improved quality in software developments, a decrease in number of incidents per release and an increased time to market and velocity to pass from development to production. Finally, from the technology point of view, DevOps results in an application with a reduced number of defects and therefore with more quality, and in an increased deployment of features.

The authors have analyzed several DevOps solutions existing on the market and in open source communities, such as The Eclipse Cloud Development project (Eclipse Cloud Development;, 2015), IBM Bluemix DevOps Services (IBM BlueMix, 2016), or Puppet+MCollective (Puppet labs, 2013) and Terraform (Terraforms, 2015) among others. From this analysis the authors conclude that the most of the current so-called "DevOps" solutions focus mostly on Continuous Delivery (CD – often also named continuous deployment in literature), while simultaneously applying practices like Continuous Integration (CI), Continuous Quality (CQ). CD practices allow for automated deployment, while CQ and CI practices allow for errors to be caught in an earlier phase of the development cycle which are thus cheaper to solve and with less rework, accompanied by a configuration management system. However, the tools hereby presented while aiming at cloud deployments, they do not fully support the multi-cloud approach.

1.2.2 Deployment Simulation

There are two main sources in the state of the art related to this topic, namely the model based testing and the cloud resource simulation.

The use of models for designing, simulating and testing such systems is currently one of the strongest industrial trends with significant impact on the overall development and quality assurance processes. Most of developments in that area, such as Fokus!MBT (Fraunhofer Fokus, 2015) have been mainly focused on the testing of the correctness of the model and not so much in the determination of the non-functional characteristics of the resulting system. However, some works have been carried out on the simulation of performance metrics for grid computing (Li, 2009); with can be used as a starting point for the definition of strategies for cloud resources performance simulation.

In the domain of cloud resource simulation, the most relevant input is Cloudsim (CLOUDS Laboratory, University of Melbourne, 2012). It provides a generalized and extensible simulation framework that enables seamless modelling, simulation, and experimentation of emerging Cloud computing infrastructures and application services. The usage of Cloudsim is based on coding the use cases in Java, and the provided example mainly make reference to IaaS infrastructures.

On other hand, there are several tools in the market that allow comparing IaaS services, such as CloudHarmony (CloudHarmony, n.d.), or Cloudorado (Cloudorado, 2016). There are also tools such as Cloudsleuth (Cloudsleuth, 2016) that allow comparing the performance of IaaS and PaaS providers, but are mostly focused in monitoring response-time and availability. However, all these decision support type tools usually ignore the complexity of multi-cloud environments, where combinations of cloud services need to be evaluated as well as non-functional properties.

1.2.3 Modeling of Dynamic and Reconfigurable Multi-Cloud Applications

Multi-Cloud is often defined as the serial or simultaneous use of services from diverse providers to execute an application (Petcu, 2013). At business level, Hybrid Cloud is the term commonly used. Gartner (Mazzuca, 2015) defines hybrid Cloud as the coordinated use of cloud services across isolation and provider boundaries among public, private and community service providers, or between internal and external cloud services. A number of scenarios demonstrate these serial or simultaneous interactions among hybrid heterogeneous private and public clouds and across all cloud layers (IaaS/PaaS/SaaS) (ETSI, 2013). Multi-cloud applications engineering as they are understood by the authors of this paper, with the application components distributed across heterogeneous cloud resources and still seamlessly

interoperating in a single whole, is not a common practice yet.

1.2.4 Automatic Re-Deployment

The automatic re-deployment of multi-cloud applications implies several technical issues like 1) supporting different cloud service interfaces (for distributing the application workload on different cloud platforms), 2) monitoring the application components (for verifying the metrics), and 3) implement strategies for migrating workload (namely applications or parts of applications) from one cloud platform to another.

Cloud computing applications can be monitored with a very broad set of monitoring solutions that range from services made available by Software as a Service (SaaS) providers (e.g. AppDynamics (APPDYNAMICS, 2016), BMC's TrueSight Pulse Monitoring-as-a-Service (BMC, 2016), New Relic (New Relic, 2016), RackSpace's Cloud Monitoring (Rackspace, 2015), etc.), services provided by the platform hosting the application (e.g. OpenStack Monasca (OpenStack, 2016), AWS CloudWatch (Amazon, 2016)) and tools implemented as stand-alone software packages (e.g. Nagios (Nagios, 2016), Zabbix (Zabbix, 2016)).

Adapting cloud applications for restoring the expected working conditions can be interpreted and implemented in several ways. Some cloud platforms (e.g. Google Cloud Platform (Google Cloud Platform, 2016) and AWS (Elastic Cloud Gate, 2014)) provide functions for periodically monitoring the status of a VM and, in case of unavailability, for restarting it. For sophisticated conditions, it is possible to adapt the application taking advantage of a fundamental capability of cloud computing which is named elasticity.

1.2.5 Cloud Service Brokers

"Cloud Marketplaces" are emerging to offer a mixture of service management, cloud deployment automation and application assembly, often in multi-cloud environments. Cloud providers such as Amazon WS (Amazon, 2016), HP (HPE, 2016) or IBM (IBM, 2016) have already launched their own cloud marketplace services. At the same time, both commercial solution providers (such as Appcara AppStack (Appcara, 2015) and Jamcracker Service Delivery Network (Jamcracker Platform, 2016) and Open Source initiatives (Ubuntu Juju (Ubuntu, 2016)) are developing solutions that enable the creation of customized Cloud marketplaces.

2 DECIDE: AN ADVANCED MULTI-CLOUD SERVICES BASED FRAMEWORK

2.1 DECIDE Framework: Proposed Approach

DECIDE focuses on a novel concept of multi-cloud applications. The application of this new concept encompasses several challenges that pave the way for the proposed innovations and for improving the competitive advantage of DECIDE partners compared to what the market offers today. The design of efficient multi-cloud application requires a set of established architectural patterns. DECIDE will develop architectural patterns and modeling practices focused on the description of the system architecture in terms of cloud resource dependencies as well as in terms of NFR of the system as a whole. DECIDE will provide the architect with suggestions on which pattern has to be applied, how, when and the potential trade-offs, for multi-cloud applications deployed on heterogeneous cloud providers configuring different cloud layers (as in an IoT environment), making use of heterogeneous resources and services, in some cases, provided by the DECIDE Advanced Cloud Service intermediator.(ACSmI) ACSmI will be user and resource-centric, searching always for the best opportunistic choices while fulfilling the requirements set by the user. Moreover, ACSmI will develop its activities in a context which will be legally secured by adequate innovative contractual and policy solutions and will foster cross-border interoperability. The DECICE OPTIMUS simulation tool will provide on one hand, the automation of the provisioning resources and deployment scripts for multi-cloud native applications based on the modeling of cloud resources at multiple cloud layers (IaaS, PaaS) and of multiple CSPs, and on the other, it will profile and classify the application components, which will be used to simulate the application behavior under certain conditions.

Multi-cloud applications demand supporting tools, such as DevOps, not only for their design, development and deployment but also for their operation. In order to remain sustainable, a cloud based application cannot stop its operation and it is expected that it is self-adaptive with respect to the new topology needed to fulfill the users' requirements at all times. DECIDE aims to offer a dynamic monitoring of NFRs as set by the user or potential SLA violations, which will trigger the self-adaptability and reconfiguration of the application at run-time through the DECIDE ADAPT multi-cloud

application self-adaptation tool. It will pro-actively adjust the running configuration of the application based on measurements that are derived from the dynamic monitoring activities of both the application and the NFPs of the CSPs and cloud offerings where the application is deployed and making use of.

The assembling of the mentioned novel components along with other DevOps natural components (such as continuous integration (CI), continuous quality (CQ), and continuous delivery (CD)) will set up a DevOps framework for development and operation of multi-cloud native applications in compliance with the DevOps paradigm.

DECIDE workflow starts from the design of the multi-cloud native application that is sensitive to the changeable situation in a multi-cloud based environment. For that, developers establish a set of quantitative (i.e. Mean Time Between Failures (MBTF)), availability, response time, lag, cost, throughput) and qualitative (i.e. security, location, financial, low / high technological risk) NFPs that the application must comply with and uses DECIDE's ARCHITECT tool to support the design and development process of the distributed application and its components through the architectural implementation of patterns and the recommendations derived from the tool on which patterns to apply in which components. Qualitative NFR will only be applicable for the selection of the cloud services, while quantitative will be used for monitoring and simulation purposes. After the application of this initial set of multi-cloud based architectural patterns, the developer follows with the implementation process (following the CQ, CI and again continuous architecting DevOps approach). For the implementation (continuous development, CI, CQ, CD), DECIDE will integrate open environments such as Eclipse, Git, Puppet, Chef, Docker, Jenkins, and Vagrant, among others.

As a next step, DECIDE will support the selection of the deployment topology and the underlying selection of the most suitable cloud services through the OPTIMUS simulation tool. The OPTIMUS tool will base the simulation on the profiling of the different components to be considered: profiling of the multi-cloud application, profiling of the cloud services to be used (data bases, processing clusters, etc.), profiling of the communications between nodes, and profiling of external services to be used by the multi-cloud application. For the modeling of the profiling information so that it can be processed, represented and used, existing technologies such as CloudML and OpenTosca will be evaluated. Optimization algorithms such as genetic algorithms, Harmony

search, or Dandelion codes will be used by OPTIMUS to provide a set of potential combination of cloud services that fulfills the established user requirements. Along with these simulation results, OPTIMUS will provide the developer with information about the required changes in the application structure/schema/code to achieve the required configuration deployment and the technological risk that each of these configurations imply (low technological risk or high technological risk), i.e. moving from an IaaS to a PaaS, move from one PaaS to another like OpenShift vs. Cloudfoundry.

Once the application is implemented and the cloud services are selected, the developer needs to define the service level agreement that the application will offer to end-users (Multi-Cloud SLA - MCSLA). This MCSLA will be influenced by the SLAs of the underlying (combination of) cloud services to be contracted. DECIDE Multi-cloud native applications DevOps framework will support the definition of these composite MCSLAs (Multi Cloud Service Level Agreement) and the corresponding SLOs (Service Level Objectives) of the application and the dependencies and needs on the underlying (combination of) cloud services in a machine-readable format for the representation. This composite MCSLA will be then assessed at run time to check if it is being accomplished.

To finish this first cycle of the development phase, the developer will select the deployment scripts based on the selected configuration from the simulation phase through the continuous deployment supporting tools and the architectural patterns for deployment. Each deployment configuration will be stored in the multi-cloud native application controller, maintaining the current deployment configuration situation as well as the historic of the previous deployment configuration used, so that they can be checked in the re-deployment phase.

Once the application has been developed, the operation phase starts. The application owner contracts the corresponding (combination of) cloud services (accomplishing the required MCSLAs) and deploys the application over different clouds using the ADAPT continuous deployment tool.

During the application operation phase, the DECIDE self-adaptation application provisioning tool will continuously monitor and assess the fulfillment of the established NFR and MCSLA. If a violation of any of the former metrics occurs, the self-adaptation tool through the ACSmI will assess the operation of the (combination of) cloud services selected and discard those that are affecting the MCSLA. If the application configuration has been established as of low technological risk, the multi-cloud application will be self-adaptive and it will be

redeployed automatically, following a new deployment configuration. In case the application has been identified as high technological risk, once it has identified the aspects that are affecting the malfunctioning of the application, it will alert the operator and using the OPTIMUS tool it will look for new (combination of) cloud services to set up a new deployment schema. The DECIDE application controller and the continuous deployment supporting tools will support the selection of the new deployment scripts (based on the architectural patterns for deployment), and thus semi-automatically re-deploy it. DECIDE will also support parallel re-deployment strategies definition and multiple cloud layers. In this case the new operation phase will start again contracting the new services and deploying the new scripts into the new configuration of cloud services.

2.2 DECIDE: Proposed Technical Architecture

Figure 1 presents the proposed technical architecture for DECIDE components.

3 DECIDE TESTS BEDS

The resulting assets of this approach will be validated in several test beds. They have been selected bearing in mind that the main beneficiaries of this framework are, namely developers and operators of applications that need to be legally aware and compliant, and need to fulfill high demanding requirements of performance, availability and reliability, without reaching high costs.

3.1 Case Study 1: High Availability

High availability (HA) is usually offered in the same ISP, as a high cost dedicated hardware solution distributed, when available, in different Datacenters or in different ISPs using different control panel and services. The main aim of this use case is to provide a low-cost HA service that users can manage through a single control panel. Through DECIDE ACSmI, any customer will be able to add redundancy to their system in order to eliminate a single point of failure. Developers will also be able to detect failures so as to maintain availability across the different cloud platforms.

3.2 Case Study 2: e-Health

The second validation case comes from a cloud solutions provider for health data based applications and digital services. This provider is based on England even though it provides the services in England and the UK, it hosts European wide Clinical Data Entry Tools, as well as applications that have users all over Europe. Currently the problems faced around hosting health data across Europe are: (1) Legal Compliance, (2) Resilience, and (3) Performance. Often there is a compromise to be made in one of these areas to facilitate the implementation of services using health data In order to solve such challenges, it is of paramount importance to simulate potential multi-cloud deployments that comply with the legal , cross-border and interoperability requirements associated with e-health data services.
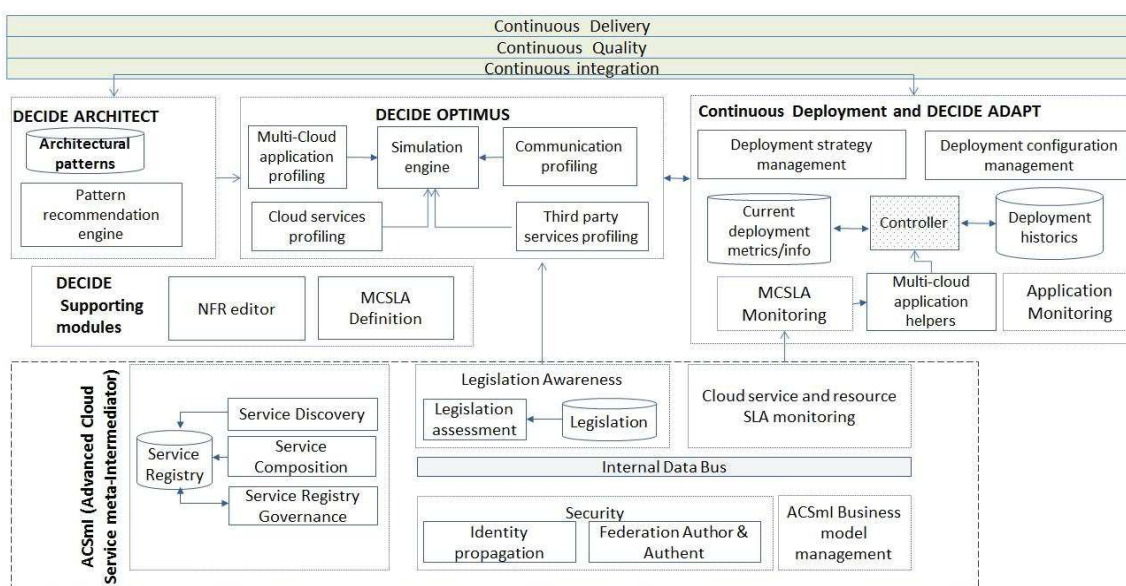

Figure 1: DECIDE Technical Architecture.

3.3 Case Study 3: Network Management

Network management is a key aspect of the operation of the telecom networks as performed by the Telecom Operators and other players. The network is crucial to every business, optimizing network performance effectively is necessary to achieve a competitive business society. Current network management tools are limited to the collection and presentation of a summary of the status of the network but do not meet present industry demands. There is a need for a very dynamic Development and Operation environment (DevOps) where maintenance activities, Long-term Network Configuration and Planning as well as updates of the operation tools and rules have to implemented and modified continuously with the business and operation feedback. This need to be updated on-the-fly but using specific tools to promote a culture of agile team work between development, test and operation.

4   FUTURE WORK

Following the approach proposed, next steps will deal with the specification of the technical design of the different tools as well as the implementations of the first prototypes.

These prototypes will be tested into real industrial use cases in the context of the DECIDE research action.

5   CONCLUSIONS

This paper presents how DECIDE action provides a novel DevOps multi-cloud application framework, enabling techniques and mechanisms to architect, develop, and dynamically deploy multi-cloud aware applications in an ecosystem of reliable, interoperable, and legal compliant cloud services. DECIDE will support software development companies in:

1. Enhancing their (multi cloud applications) development and operations processes,
2. Improving the developers' and operators' productivity,
3. While ensuring the application maintainability, Quality of Experience (QoE) and Quality of Service (QoS) in its whole life,
4. Decreasing the time-to-market.

The next activities include the actual implementation of all components described in this paper and the validation in the three use cases. For that, the authors will follow an iterative and incremental approach based on SCRUM and in alignment with a DevOps philosophy. The prioritisation of the functionalities will come from the priority analysis of functional requirements and the use cases.

ACKNOWLEDGEMENTS

REFERENCES

Amazon, 2016. Amazon CloudWatch. [Online] Available at: https://aws.amazon.com/cloudwatch/?nc1=h_ls [Accessed 17 November 2016].

Amazon, 2016. AWS Marketplace. [Online] Available at: https://aws.amazon.com/marketplace [Accessed March 2016].

Appcara, 2015. Appcara- AppStack. [Online] Available at: http://www.appcara.com/products/appstack/ [Accessed March 2016].

APPDYNAMICS, 2016. APPDYNAMICS. [Online] Available at: https://www.appdynamics.com/ [Accessed 17 November 2016].

BMC, 2016. TrueSight Pulse Monitoring-as-a-Service. [Online] Available at: http://www.bmc.com/truesightpulse [Accessed 17 November 2016].

CloudHarmony, n.d. CloudHarmony. [Online] Available at: https://cloudharmony.com/directory [Accessed 17 November 2016].

Cloudorado, 2016. Cloudorado. [Online] Available at: https://www.cloudorado.com/ [Accessed 17 November 2016].

CLOUDS Laboratory, University of Melbourne, 2012. CloudSim: A Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services. [Online] Available at: http://www.cloudbus.org/cloudsim/ [Accessed 17 November 2016].

Cloudsleuth, 2016. Cloudsleuth. [Online]
Available at: https://cloudsleuth.net
[Accessed November 2016].

CloudSurfing, 2016. CloudSurfing. [Online]
Available at: https://www.cloudsurfing.com/
[Accessed 17 November 2016].

Eclipse Cloud Development;, 2015. Eclipse Cloud Development. [Online]
Available at: http://www.eclipse.org/ecd/
[Accessed 17 November 2016].

Edwards, D., 2015. DevOps paradox: Going Faster Brings Higher Quality, Lower Costs, & Better Outcomes, s.l.: s.n.

Elastic Cloud Gate, 2014. EC2 Monitoring - Always On - Elastic Cloud GateElastic Cloud Gate. [Online]
Available at: http://blog.ecloudgate.com/ec2-monitoring-always-on/
[Accessed 17 November 2016].

ETSI, 2013. Cloud Standards Coordination - Final Report, s.l.: s.n.

Fraunhofer Fokus, 2015. Fokus!MBT Features. [Online]
Available at:
http://www.fokusmbt.com/fmbt/key_features
[Accessed 17 November 2016].

Gartner, 2012. The Business Landscape of Cloud Computing. [Online]
Available at: https://www.cloudeassurance.com/white-papers/Gartner_Daryl_Plummer_and_Financial_Times_e-
Book_The_Business_Landscape_of_Cloud_Computing.pdf
[Accessed 17 November 2016].

Gartner, 2016. Cloud Services Brokerage. [Online]
Available at: http://www.gartner.com/it-glossary/cloud-services-brokerage-csb/
[Accessed 17 November 2016].

Google Cloud Platform, 2016. Setting Instance Scheduling Options - Compute Engine. [Online]
Available at:
https://cloud.google.com/compute/docs/instances/setting-instance-scheduling-options
[Accessed 17 November 2016].

HPE, 2016. HEP Helion. [Online]
Available at: http://www8.hp.com/us/en/cloud/helion-overview.html
[Accessed March 2016].

IBM BlueMix, 2016. IBM BlueMix DevOps services. [Online]
Available at: hub.jazz.net
[Accessed 2017 November 2016].

IBM, 2016. IBM Cloud. [Online]
Available at:
https://www.ibm.com/marketplace/cloud/us/en-us?lnk=mp
[Accessed March 2016].

Jamcracker Platform, 2016. Jamcracker Services Delivery Network. [Online]

Available at: http://www.jamcracker.com/jamcracker-services-delivery-network-jsdn
[Accessed March 2016].

Li, H. a. R. B., 2009. Model-based simulation and performance evaluation of grid scheduling strategies. In: Future Generation Computer Systems. s.l.:s.n., pp. 460-465.

Mazzuca, J., 2015. Survey Analysis: Cloud Adoption Across Vertical Industries Exhibits More Similarities Than Differences, s.l.: Anderson.

Nagios, 2016. Industry Standard In IT Infrastructure Monitoring. [Online]
Available at: https://www.nagios.com
[Accessed 17 November 2016].

New Relic, 2016. Application Performance Monitoring Features. [Online]
Available at: http://newrelic.com/application-monitoring
[Accessed 17 November 2016].

OpenStack, 2016. Monasca- OpenStack. [Online]
Available at: https://wiki.openstack.org/wiki/Monasca
[Accessed 17 November 2016].

Petcu, D., 2013. Multi-Cloud: Expectations and Current Approaches. New York, s.n.

Puppet labs, 2013. MCollective Introduction: Programmatic Execution of Systems Administrations Actions - Puppet Labs. [Online]
Available at: https://puppetlabs.com/mcollective
[Accessed March 2016].

Rackspace, 2015. Rackspace Monitoring. [Online]
Available at:
https://www.rackspace.com/cloud/monitoring
[Accessed 17 November 2016].

Terraforms, 2015. Terraform. [Online]
Available at: https://www.terraform.io/
[Accessed April 2016].

Ubuntu, 2016. Ubuntu Juju. [Online]
Available at: http://community.ubuntu.com/
[Accessed March 2016].

UpGuard, 2016. The four prerequisites for DevOps success, s.l.: s.n.

Zabbix, 2016. The Enterprise-class Monitoring Solution for Everyone. [Online]
Available at: http://www.zabbix.com
[Accessed March 2016].