Chapter 1
Introduction

Elisabetta Di Nitto and Dana Petcu

1.1 Context

Cloud computing is a major trend in the ICT industry. The wide spectrum of available Clouds, such as those offered by Microsoft, Google, Amazon, HP, AT&T, and IBM, just to mention big players, provides a vibrant technical environment, where even small and medium enterprises (SMEs) use cheap and flexible services creating innovative solutions and evolving their existing service offer. Despite this richness of environments, Cloud business models and technologies are characterized by critical issues, such as the heterogeneity between vendor technologies and the resulting **lack of interoperability** between Clouds. In this setting a number of challenges for systems developers and operators can be identified, especially for SMEs that have limited resources and do not have the strength to influence the market. In particular:

- **Vendor Lock-in** [1, 2]. Cloud providers offer proprietary solutions that force Cloud customers to decide, at the early stages of software development the design and deployment models to adopt (e.g., public vs. hybrid Clouds) as well as the technology stack (e.g., Amazon Simple DB vs. Google Bigtable).
- **Risk Management**. There are several concerns when selecting a Cloud technology such as payment models, security, legal and contractual, quality and integration with the enterprise architecture and culture.

E. Di Nitto (✉)
Politecnico di Milano - DEIB, Piazza L. da Vinci, 32, 20133 Milan, Italy
e-mail: elisabetta.dinitto@polimi.it

D. Petcu (✉)
Institute e-Austria TimiŞoara and West University of TimiŞoara,
B-dul Vasile Pârvan 4, 300223 TimiŞoara, Romania
e-mail: petcu@info.uvt.ro

E. Di Nitto et al. (eds.), *Model-Driven Development and Operation of Multi-Cloud Applications*, PoliMI SpringerBriefs,
DOI 10.1007/978-3-319-46031-4_1

- **Quality Assurance**. Cloud performance can vary at any point in time. Elasticity may not ramp at desired speeds. Unavailability problems exist even when 99.9 % up-time is advertised (e.g., Amazon EC2 and Microsoft Office 365 outages in 2011).

The above issues can be addressed by enabling companies to develop their applications for multiple Cloud targets, by offering them proper tools to analyze the risks, performance and cost of various solutions and identify the ones that best suit the needs of the specific case, and by supporting a multi-Cloud deployment of applications to ensure a level of availability that is greater than the one offered by each specific Cloud. In this context, within the MODAClouds project, we focused on the following objectives:

- Deliver an advanced software engineering model-driven approach and an Integrated Development Environment (IDE) to support systems developers in building and deploying applications, together with related data, to multi-Clouds spanning across the full Cloud stack (Infrastructure as a Service, shortly IaaS, Platform as a Service, shortly PaaS, and Software as a Service, shortly SaaS).
- Define quality measures, monitoring mechanisms, prediction models, and adaptive policies to provide quality assurance in Clouds and multi-Clouds.
- Provide support to costs and risks analysis to increase trust in Clouds.
- Develop an integration framework between design tools and run-time.
- Create relevant and complex case studies for the entire risks assessment and software engineering methodologies.
- Analyze and validate project outcomes through case studies.
- Ensure distribution of project results via dissemination activities on relevant publication channels, training, and standardization.
- Provide community-based open source solutions supporting the full applications life-cycle.

In this chapter we provide a motivation for the adoption of a multi-Cloud approach and of a model-driven, quality aware development and operation paradigm (Sect. 1.2), offer a brief overview of related work (Sect. 1.3), introduce the MODAClouds approach and toolset (Sects. 1.4 and 1.5), and, finally, define the goals of this book (Sect. 1.6).

1.2 Motivation

The main drivers for exploiting a multiple Cloud approach can be of various nature, from the need to avoid dependence from a single provider to the need to follow local constraints and laws, to the opportunity to replicate software in order to enhance availability. The main factors we have identified are summarized in Fig. 1.1. In the figure we distinguish between those drives that imply the simultaneous usage of services from multiple Clouds and those that are more concerned with the possibility

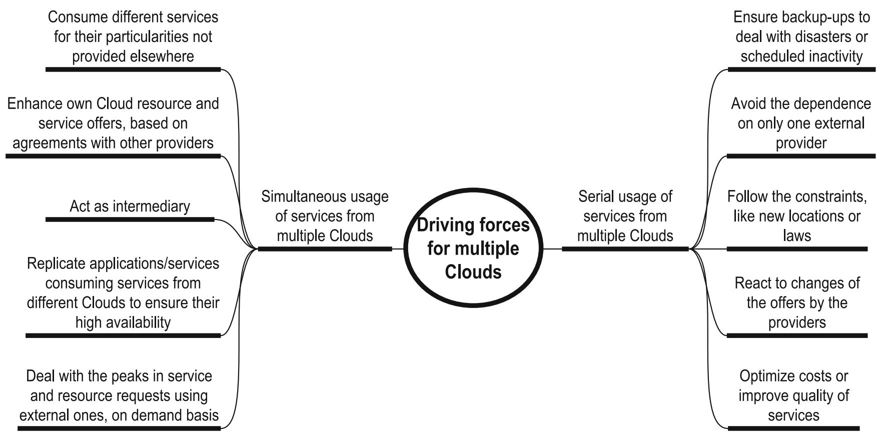
Driving forces for multiple Clouds
Simultaneous usage of services from multiple Clouds
Consume different services for their particularities not provided elsewhere
Enhance own Cloud resource and service offers, based on agreements with other providers
Act as intermediary
Replicate applications/services consuming services from different Clouds to ensure their high availability
Deal with the peaks in service and resource requests using external ones, on demand basis
Serial usage of services from multiple Clouds
Ensure backup-ups to deal with disasters or scheduled inactivity
Avoid the dependence on only one external provider
Follow the constraints, like new locations or laws
React to changes of the offers by the providers
Optimize costs or improve quality of services

Fig. 1.1 Drivers for multi-Cloud adoption

of preparing a software system to be run on multiple Clouds but still using a single Cloud at a time during operation.

To exemplify concrete needs in an industrial context, we refer to the case of a small company that we call MODAFIN, specialised in IT applications for financial services. Its main product line is a proprietary solution for stock market operations, cash administration, and lending management.

MODAFIN most profitable activities are software customization and life-cycle management for this product line.

Customisation involves development of custom modules to accommodate new functional requirements. Moreover, it includes application integration with existing databases and legacy business systems at the customer' site.

Life-cycle management needs to assure high-availability for real-time computations during market hours, scalability and low operational costs for batch analytic workloads running after-hours. MODAFIN fulfills these quality requirements with a capacity management consultancy team following the application life-cycle.

The consultancy team has been working for a long time at the customers' site, where the system is deployed in the operation environment. Thanks to the diffusion of the Cloud, however, new needs have arisen. At night, some customers want to run their batch analytic workloads at the cheapest operational costs of Amazon on-spot instances. During the day, they expect calculation engines to ramp-up computing power at an unprecedented pace when the stock market gets volatile. Moreover, some customer applications are collecting and processing stock market data directly on the Cloud using PaaS datastore services such as Google Bigtable or Amazon SimpleDB. At the same time, customers are cutting spending in consultancy services for life-cycle management as they are relying more and more on SaaS services.

To remain competitive, MODAFIN solution must evolve addressing all above requirements. To do so, the Company needs to apply advanced software engineering methodologies revising both the software development process and its life-cycle management services:

- It needs to develop a solution that can be executed on a broad spectrum of customers IaaS/PaaS, also supporting Cloud bursting, that is, the ability to move part of the system on a different Cloud to manage pick of traffic when needed.
- It must develop a flexible architecture for the system so that it could be adapted to new Cloud offers emerging in the next 5–10 years to adapt to changes of context and requirements.
- It needs libraries and connectors to integrate various data storage tools and services to address different needs in terms of performance, data locality, scalability and the like.
- It needs simple to use tools to perform what if analyses and optimizations on the system configuration in order to allow for the fulfillment of the required QoS.
- It needs a multi-Cloud environment for execution, which supports monitoring, smart load balancing, scale-in and out on several Clouds to avoid that availability or performance outages of a single Cloud provider would turn into a disaster for MODAFIN's own business.

All above needs result not only in the adoption of a multi-cloud approach, but also in the exploitation of a proper development and operation set of tools and methods, which are specifically built to support multi-Cloud.

Within the MODAClouds approach we have experimented with model-driven development enhanced with the possibility of exploiting models not only as part of design but also as part of the runtime. In this case the system model becomes a live object that evolves with the system itself and can be used to send back to the design time powerful information that enables a continuous improvement of the system. In new terms, this approach goes into the direction of offering a valid tool to support DevOps, that is, the ability to support development and operation in a seamless way.

1.3 Related Work

Model-driven engineering (MDE) allows developers to build the system at various level of abstractions. It is often summarized as "model once, generate anywhere" and, as such, becomes particularly relevant when it comes to provisioning and deployment of applications and services across multiple Clouds, as well to migration of source code and data from one service provider to another.

Services Oriented Architecture (SOA) related technologies are often used to define Cloud-enabled applications without going into the fine details of deployment. Services are often modeled by means of general purpose languages such as UML. Service-specific languages have also been designed for SOA approach (e.g. SoaML[1]). USDL[2] goes even further, by allowing designers to specify, beside services and their

[1]http://www.omg.org/spec/SoaML/1.0/Beta2/.

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

interfaces, non-functional aspects of these services (e.g. pricing, legal, certification, documentation).

Other approaches are related to the specific concept of Web Service: WSDL[3] enables the specification of a list of services, interfaces, data types and orchestration processes at a syntactical level, OWL[4] is a semantic Web language which enables the specification of the semantics of the services, besides their syntax. Both these approaches do not allow for the description of non-functional requirements and constraints. However, they can be complemented with the OMG UML profile for QoS, QFTP,[5] which allows a designer to specify QoS requirements and to connect them to service descriptions.

While the above approaches are Cloud-agnostic, modeling concepts and technologies for supporting provisioning, deployment and adaptation of applications in the Cloud have been recently developed. They exploit the uniform interfaces provided by various libraries for application deployment and control at run-time. We can mention here the most successful ones: jclouds,[6] libcloud,[7] δ-cloud[8] or fog.[9] For example, the jclouds model includes the description of nodes with metadata (like CPU, RAM, security policy), parameters (like minCPU, OS type) and a set of commands to be executed on nodes, as well as on the groups of nodes to be managed together.

Most of the above mentioned libraries are providing a common access to multiple Clouds, but are dependent on the programming language. Typically, they provide a code-based model of the infrastructure and do not offer any mechanism for automatic provisioning and deployment of applications on the Clouds. Moreover, they work at the IaaS level and do not expect applications and services to be presented in terms of models. To fill this gap, MODAClouds offers a complete set of model-based tools from design to deployment and run-time control of the applications.

Recently, several frameworks for managing multi-Cloud services and applications have been developed. They provide capabilities for the provisioning, deployment, monitoring, and adaptation of applications without being language-dependent. We mention here three of them: Cloudify,[10] Scalr[11] and CloudFoundry.[12] For example, the Cloudify model for deploying applications includes recipes for information like: (i) required infrastructure and how it should be used, (ii) clusters of service instances that make up an application tier, (iii) configuration (including provisioning and scaling rules) of an application and the services it is made of, (iv) probes used to monitor the status of the system. These frameworks are important to optimise performance,

[3]http://www.w3.org/TR/wsdl.

[4]http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/.

[5]http://www.omg.org/spec/QFTP/.

[6]http://jclouds.apache.org.

[7]http://libcloud.apache.org.

[8]http://deltacloud.apache.org.

[9]http://fog.io.

[10]http://www.cloudify.org.

[11]http://scalr.com.

[12]http://www.cloudfoundry.org.

availability, and cost of multi-Cloud systems. However, they do not come with any structured guideline/methodology, thus, developers and operators are typically left hacking at code level rather than engineering multi-Cloud systems following a structured tool-supported methodology.

The models@runtime paradigm, often used in MDE, proposes to leverage models during the execution of adaptive software systems to monitor and control the way they adapt. This approach enables the continuous evolution of the system with no strict boundaries between design-time and runtime activities. Models@runtime provides an abstract representation of the running system causally connected to the underlying state of the system which facilitates reasoning, simulation and enactment of adaptation actions. A change in the running system is automatically reflected in a model of the current system. A modification applied to this model can be enacted on the running system on demand. Thanks to the use of models, well-defined interface are provided to monitor the system and adapt it. The models also provide a way to measure the impact of changes in the system and analyse them before their enactment on the running system. In MODAClouds we adopt the models@runtime concept in order to tame the complexity of adaptation and ease the reasoning process for self-adaptation.

MODAClouds was developed together with two siblings projects, PaaSage and ARTIST. The scope of PaaSage[13] was to extend application models with annotations concerning platform and user's goals and preference. The language used for this is called Cloud Application Modelling and Execution Language (CAMEL). CAMEL integrates various domain-specific languages using the Eclipse Modelling Framework. Within this context, PaaSage has extend and adapt MODAClouds' CloudML to support model-based provisioning and deployment of Cloud-based systems. CloudML is used also by the ARTIST initiative,[14] which offers a set of methods and tools for an end-to-end assisted migration of legacy software to the Cloud. ARTIST followed an earlier initiative, REMICS[15] which proposed a leap progress in legacy systems migration to Service Clouds by providing a model driven methodology and tool following the Architecture Driven Modernization concept (use knowledge discovery to recover application models and rebuild the applications following the discovered models).

The MONDO initiative[16] focused not on MDE for Clouds, but on Clouds for MDE: aiming to achieve scalability in MDE, MONDO provided an integrated open-source and Cloud-based platform for scalable model persistence, indexing and retrieval facilities, support for collaborative modelling, and parallel execution of model queries and transformations, and an Eclipse-based developer workbench that include tooling for developing queries and transformations, for querying model indexes, and for constructing large models and domain specific languages. The HEADS initiative.[17]

[13]http://www.passage.eu.

[14]http://www.artist-project.eu.

[15]http://www.remics.eu.

[16]http://www.mondo-project.org.

[17]http://www.heads-project.eu.

leveraged MDE to provide an open-source Integrated Development Environment (IDE) supporting the collaboration between platform experts (platform for mobile devices, sensors, smart objects, etc.) and Cloud-based service developers and including a domain specific modeling language and a methodology for the specification, validation, deployment and evolution of software-intensive services distributed across the future computing continuum (composed of a wide set of heterogeneous platforms).

1.4 The MODAClouds Approach

Figure 1.2 shows an overview of the MODAClouds development approach. In particular, it shows how an application is designed and packaged for deployment according to a Cloud-tailored model-driven approach. Software designers start from defining the application structure and the corresponding Quality of Service (QoS) requirements at the *Cloud Independent Model* level (CIM). In the example shown in the figure, the application is composed of three components, two of which are further decomposed in sub-components. Availability and response time requirements are defined and associated to two of the application components. At this level there is no reference to specific Cloud services and resources as the focus is exclusively on the high level design of the application itself.

From the CIM level description the designer moves then to focus on introducing Cloud-specific aspects at the *Cloud-Provider Independent Model* level (CPIM). At this level, he/she may decide, for instance, to select a certain class of database service (e.g., key-value NoSQL) and certain kinds of computational and memory resources. All these are then associated to the application logic elements they contribute to realize. At this point the developer can start running the MODAClouds QoS analysis tool that, based on the defined QoS requirements and on the typical characteristics of the selected kinds of Cloud resources and services, can provide some feedback about the realizability of the application on specific Clouds and can suggest possible optimizations.

As soon as the designer is satisfied with the specified solution, he/she can move to the *Cloud-Provider Independent Model* level (CPSM) from where he/she can finalize the selection of specific providers and services/resources for the application, run more precise QoS analyses and, finally, generate proper deployment, monitoring and self-adaptation scripts to support the runtime phases.

In all analysis and design phases, the application designers as well as the decision makers from the company can be supported in the definition of risks and benefits for the application and in the identification of the candidate Cloud services and resources based on these.

Finally, at runtime, the models defined at design time are exploited to monitor and manage the application by enabling smart self-adaptation. Moreover, the values of specific metrics characteristic for the running applications are collected and passed to the development team that can exploit them to fine-tune the application.

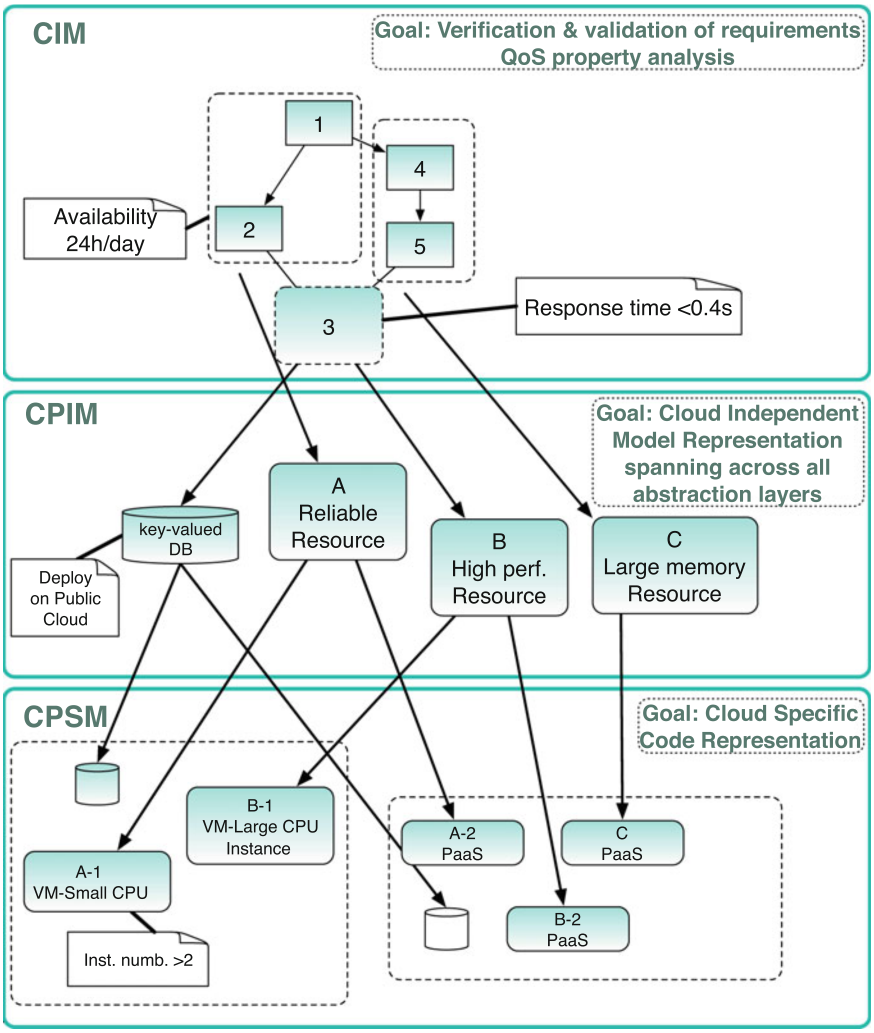

Fig. 1.2 Model-driven development in MODAClouds

As described in Chap. 10, this enables the adoption of a DevOps approach [3] that supports development and operation in a coherent manner.

1.5 The MODAClouds Toolbox

The MODAClouds model-driven approach is supported by the MODAClouds Toolbox (see Fig. 1.3). The toolbox helps lowering existing barriers between Development and Operations Teams and helps embracing DevOps practices within IT teams.

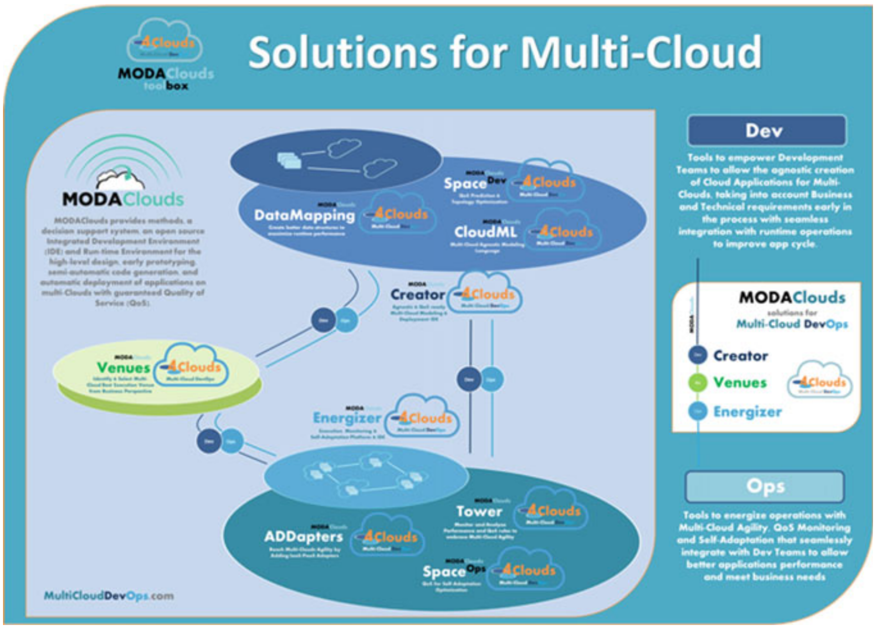
Solutions for Multi-Cloud
MODAClouds toolbox
MODAClouds
MODAClouds provides methods, a decision support system, an open source Integrated Development Environment (IDE) and Run-time Environment for the high-level design, early prototyping, semi-automatic code generation, and automatic deployment of applications on multi-Clouds with guaranteed Quality of Service (QoS).
DataMapping 4Clouds
Space Dev 4Clouds
CloudML 4Clouds
Creator 4Clouds
Venues 4Clouds
Energizer 4Clouds
Tower 4Clouds
ADDapters 4Clouds
Space Ops 4Clouds
Dev
Tools to empower Development Teams to allow the agnostic creation of Cloud Applications for Multi-Clouds, taking into account Business and Technical requirements early in the process with seamless integration with runtime operations to improve app cycle.
MODAClouds solutions for Multi-Cloud DevOps
Creator
Venues
Energizer
Ops
Tools to energize operations with Multi-Cloud Agility, QoS Monitoring and Self-Adaptation that seamlessly integrate with Dev Teams to allow better applications performance and meet business needs
MultiCloudDevOps.com

Fig. 1.3 MODAClouds toolbox

Thanks to it, organizations of any size can Build and Run Cloud Applications driven by business and technical needs and quality requirements. The toolbox is comprised of the following elements: (1) *Creator 4Clouds*, an Integrated Development Environment (IDE) for high-level application design; (2) *Venues 4Clouds*, a decision support system that helps decision makers identify and select the best execution venue for Cloud applications, by considering technical and business requirements; (3) *Energizer 4Clouds*, a Multi-Cloud Run-time Environment energized to provide automatic deployment and execution of applications with guaranteed Quality of Service (QoS) on compatible Multi-Clouds.[18]

Creator 4Clouds, in turn, includes plugins focusing on (i) analysing the QoS/cost trade-offs of various possible application configurations (Space 4Clouds^{Dev}), (ii) mapping high level data models into less expressive but more scalable NoSQL, (iii) deploying the resulting application on multi-Cloud by exploiting the CloudML language. Overall, Creator 4Clouds is a unique tool supporting design, development, deployment and resource provisioning for multi-Cloud applications. It limits lock-in and provides features to assess the QoS guarantees required by the application. Moreover, it offers support to the definition of the application SLA.

[18] All these tools are available as open source, see http://www.modaclouds.eu/software/.

Energizer 4Clouds includes the frameworks to support monitoring (Tower 4Clouds) and self-adaptation (Space 4Clouds^{Ops}), together with utilities that perform ancillary tasks in the platform (ADDapters 4Clouds). Energizer 4Clouds is one of the few approaches that addresses, in a single framework, the needs of operators willing to run their applications in a multi-Cloud environment. Through Tower 4Clouds, operators are able to perform complex monitoring and data analyses from multiple sources. Moreover, thanks to Space 4Clouds for Ops, it identifies and actuates proper self-adaptation actions that take into account the current and foreseen state of the system under control.

We have included in the design of the MODAClouds architecture what we call *Feed-Back Loop* technologies that extend capabilities offered by Creator, Venues and Energizer 4Clouds. Thanks to the Feed-Back Loop approach, Tower 4Clouds connects with Creator 4Clouds and Venues 4Clouds, respectively. The first connector is responsible for providing developers and the QoS engineers with the perspective of the application behavior at runtime to improve the development process and incorporate DevOps techniques and tools into the process. The second connector allows Venues 4Clouds to adapt its knowledge base according to real live data. This helps in offering to users an updated vision of services quality for future recommendations. The capability of the runtime to influence the design time is in line with current research and is a very important feature to empower multi-Cloud application developers.

1.6 Book Objectives

The objective of this book is to: (i) present the methods and tools composing the MODAClouds solution as well as the business needs they address, and (ii) to show their validity and utility through four industrial cases. The presentation will highlight both development and operation aspects and the way they are integrated to support a DevOps approach.

References

1. Gartner (2012) 2012 Cloud Computing Planning Guide
2. Forbes (2011) Cloud computing's vendor lock-in problem: why the industry is taking a step backward
3. Debois P (2011) DevOps: a software revolution in the making? J Inf Technol Manage