Cloud Computing

Frank Leymann, University of Stuttgart

Cloud Computing is often considered as disruptive technology, i. e., technology that will change the way we use IT: from a user perspective IT resources become available on demand, and from a provider perspective IT resources will be offered in a utility manner. "On demand" refers to the fact that users of IT resources access them when they need them, for how long they need them, and only pay for this actual usage. "Utility" refers to the fact that providers of IT resources offer and manage them in a way similar to how resources like electricity, telephony, water etc. are offered and managed, i. e., in an industrial manner, especially shared by all their customers and tenants. These two viewpoints have implications on infrastructures required for offering and accessing IT resources (also called "cloud services" in some of the articles of this special issue), the way how applications to be offered and run in clouds should be built, and on the economic aspects of IT resources. The articles collected in this special issue present and discuss these implications.

Breiter and his co-authors review definitions of cloud computing and summarize cloud layering models as well as cloud service models. They introduce a comprehensive lifecycle of a cloud service beginning with the definition of a cloud service and ending with its termination. The need for automating the management of cloud services is worked out. The notion of a service template is introduced that captures the information required to support such an automated management enabling cloud providers to host and offer any corresponding cloud service. A reference architecture for an overall cloud environment encompassing consumers, providers, and creators of cloud services is presented. Important success criteria for the acceptance of clouds are sketched.

Grid technology is often considered as predecessor technology that is closest to cloud technology. Brandic and Dustdar compare both technologies. First, they discuss main characteristics of both, grids and clouds. Next, they identify the major differences between these technologies, which are in the area of business models, resource management, provisioning models, and availability. They sketch a sample grid project as well as a sample cloud project and exemplify based on these projects the aspect of managing quality of services in corresponding environments: the differences in service provisioning, QoS management, and SLA attainment strategies are worked out.

Azeez and his co-authors sketch the implications of cloud technology for the future use of IT. They argue that a cloud platform for building large service oriented applications is a key and present an architecture for such a platform. A corresponding implementation called "Stratos" is described. First hand, a cloud platform for service oriented applications is beneficial for application owners. In order to make such a platform beneficial for the users of the corresponding applications too, the authors identify six properties ("Cloud Native" properties) that such a platform must support. Use cases motivate these properties. Selective platform as a service offerings are compared based on the Cloud Native properties.

Fehling and Mietzner propose how applications that target multiple tenants should be designed to fully benefit from clouds. They argue that such applications should offer points of variability to support tenants in customizing both, the functional as well the non-functional properties of an application. In case the application is component-based its flexibility is further increased and its different components may even be deployed in different clouds. A metamodel for points of variability of component-based applications is presented and a corresponding modeling tool is sketched. To ease the management of cloud applications the architecture of a management infrastructure is worked out. Based on this infrastructure the management of cloud applications becomes individual being derived from an application's variability.

Leukel and Kirn use the concept of a "cloud value system" to describe a service provider and his relation to his infrastructure providers as well as his users. The concept is recursive, i. e., complex value systems can be described and evaluated by this concept. The authors propose a corresponding framework to explain and analyze economic activities in a cloud environment. They apply transaction cost theory to derive the determinants of transaction costs in cloud value systems. Based on agency theory the authors identify two fundamental problems: (i) conflicting goals with respect to resource sharing between a service provider and its infrastructure providers, and (ii) sharing risks by means of proper service contracts. Property rights theory strives towards efficient distribution of all usage rights between participants in a cloud value system maximizing the total utility. Concrete research problems related to their framework are sketched.

Other important aspects could not be covered because of the space limitations of a special issue of a journal. Most notably, the issues of security, trust and compliance rules companies have to obey and that affect the categories of applications or application components that might be hosted in cloud environments are not touched. Nevertheless, a huge number of applications (for example catalogue applications, timetable applications, or emergency planning applications – to name but a few) are only marginally or not at all affected by such concerns and are immediate candidates to be run in a cloud environment. Thus, cloud technology can be seriously considered as integral part of the IT landscape. This special issue provides much basic information for this.

Frank Leymann

Received: May 21, 2011

Prof. Dr. Frank Leymann, since 2004, Frank Leymann is a full professor of computer science and director of the Institute of Architecture of Application Systems at the University of Stuttgart, Germany. His research interests include service oriented computing and middleware, workflow- and business process management, Cloud Computing. He worked on corresponding research projects funded by EU, DFG, BMBF, BMWi, and industry. He is (co-)author of more than 200 publications and editor of several scientific journals.

Before accepting his professor position Frank Leymann worked for two decades for IBM Software Group building database and middleware products: He built tools supporting conceptual and physical database design for DB2; built performance prediction and monitoring tools for an object database system; was co-architect of a repository system; built both, a universal relation system as well as a complex object database system on top of DB2; and was co-architect of the MQSeries family. In parallel to that, Frank Leymann worked continuously since the late eighties on workflow technology and became the father of IBM's workflow product set. As an IBM Distinguished Engineer and elected member of the IBM Academy of Technology he contributed to the architecture and strategy of IBM's entire middleware stack as well as IBM's On Demand Computing strategy. From 2000 on, Frank worked as co-architect of the Web Service stack. He is co-author of many Web Service specifications, including WSFL, WS-Addressing, WS-Metadata Exchange, WS-Business Agreement, the WS-Resource Framework set of specifications, WS-HumanTask and BPEL4People; together with Satish Thatte, he was the driving force behind BPEL4WS. Also, he is co-author of BPMN 2.0.

Address: University of Stuttgart, Institute of Architecture of Application Systems, Universitätsstraße 38, D-70569 Stuttgart, Germany,
e-mail: leymann@iaas.uni-stuttgart.de