Cloud Computing and Computing Evolution

Cloud Computing and Computing Evolution

Markus Böhm, Stefanie Leimeister, Christoph Riedl, Helmut Krcmar

Technische Universität München (TUM), Germany

1 Front Matter

- Summarizes current scientific definitions of cloud computing
- Suggests a comprehensive definition of cloud computing
- Evaluates cloud computing from a technological and a IS delivery perspective
- Discusses innovative aspects of cloud computing
- Highlights some promising domains for cloud computing applications
- Provides a method to estimate whether an application is suitable for cloud computing or not

2 Introduction

The term cloud computing is sometimes used to refer to a new paradigm – some authors even speak of a new technology – that offers IT resources and services over the Internet. The technology analysts at Gartner see cloud computing as a so-called "emerging technology" (Fenn et al. 2008) on its way to the hype. When looking at the number of searches for the word pair "cloud computing" undertaken with the Google search engine one can get a feeling of the high interest on the topic. Even terms like "outsourcing", "Software-as-a-Service (SaaS)" or "grid computing" have already been overtaken (Google 2009).

Although, nearly everybody in the IT sector speaks about cloud computing, the concept remains somewhat unclear to many. With this contribution we aim to provide an understanding of cloud computing. Starting with a literature review on current definitions of cloud computing we will summarize the core characteristics and suggest a comprehensive definition. We will further examine the evolution of cloud computing from two different perspectives. From a technological point of view we will describe the phenomenon's development, highlighting reoccurring trends in computing history. Eventually we will show, which components constitute the cloud computing concept. Our second perspective draws on the delivery model for information systems that changed through cloud computing. Our understanding of cloud computing is that of a new business model, for delivering IT resources and services, flexible, on demand and on a pay-per-use basis. We will conclude our contribution with an estimation about what classes of applications appear to be promising for cloud computing.

3 Understanding the concept of cloud computing: defining a new phenomenon

Due to the current fashion, the term cloud computing is often used for advertising purposes in order to revamp existing offerings with a new wrap. Larry Ellison's (CEO of Oracle) statement at 2007s Analysts' Conference provides a felicitous example: *"We've redefined cloud computing to include everything that we already do. I can't think of anything that isn't cloud computing with all of these announcements. The* *computer industry is the only industry that is more fashion-driven than women's fashion"* (Fowler et al. 2009). In the following chapter we try to clarify the term to provide a common understanding.

3.1 Related work

To date there are few scientific contributions which strive to develop an accurate definition of the cloud computing phenomenon. Youseff et al. were among the first who tried to provide a comprehensive understanding of cloud computing and all its relevant components. They regard cloud computing as a "*collection of many old and few new concepts in several research fields like Service-Oriented Architectures (SOA), distributed and grid computing as well as Virtualization*" (Youseff et al. 2008). According to Youseff et al. "*cloud computing can be considered a new computing paradigm that allows users to temporary utilize computing infrastructure over the network, supplied as a service by the cloud-provider at possibly one or more levels of abstraction*" (Youseff et al. 2008). When speaking about levels of abstraction, the authors refer to their proposed cloud computing ontology which will be described in Chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**.

According to Armbrust et al. "*Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. The services themselves have long been referred to as Software as a Service (SaaS). The datacenter hardware and software is what we will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the general public, we call it a Public Cloud; the service being sold is Utility Computing. We use the term Private Cloud to refer to internal datacenters of a business or other organization, not made available to the general public. Thus, Cloud Computing is the sum of SaaS and Utility Computing, but does not include Private Clouds*" (Armbrust et al. 2009). In this way the authors as well understand cloud computing as a collective term, covering preexisting computing concepts such as SaaS and utility computing. Armbrust et al. especially perceive the following aspects as new: (1) the illusion of infinite computing capacity available on demand, (2) the elimination of up-front commitment to resources on the side of the cloud user, and (3) the usage-bound pricing for computing resources on a short-term basis (Armbrust et al. 2009).

Being grid computing scholars, Buyya et al. postulate a more technical focused approach, regarding cloud computing as a kind of parallel and distributed system, consisting of a collection of virtualized computers. This system provides resources dynamically, whereas Service Level Agreements (SLA) are negotiated between the service provider and the customer (Buyya et al. 2008).

In an attempt to provide a generally accepted definition, Vaquero et al. have derived similarities, based on Geelan's collection of expert opinions (Geelan 2009). They claim that "*clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for* *an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs*” (Vaquero et al. 2009).

The majority of definitions however originate from cloud computing service providers, consulting firms and market research companies. The market research company IDC for example defines cloud computing very general as “*an emerging IT development, deployment and delivery model, enabling real-time delivery of products, services and solutions over the Internet*” (Gens 2008). In that sense, cloud computing is the technical basis for cloud services, offering consumer and business solutions that are consumed in real-time over the internet. The technological foundation of cloud computing includes infrastructure, system software, application development and deployment software, system and application management software as well as IP-based network services. IDC also mentions usage-bound pricing as a core characteristic (Gens 2008). Another example of a market research company’s declaration is Gartner’s definition of cloud computing as “*a style of computing where massively scalable IT-enabled capabilities are delivered 'as a service' to external customers using Internet technologies*” (Plummer et al. 2008).

Autor[1]	**Service**	**Hardware**	**Software**	**Data**	**(Development) Platform**	**Pay-Per-Use**	**off-premise (public)**	**Scalability**	**No Upfront Commitment**	**Virtualization**	**SLA**	**Deterministic Performance**	**Internet/network**	**Automation**
Armbrust et al. [1]	x	x	x			x	x	x	x				x	
Breiter/Behrendt [2]	x	x				x		x		x				x
Briscoe/Marinos [3]	x	x	x					x		x			x	
Buyya [4]		x						x		x	x			
Foster et al. [5]	x	x	x		x			x		x			x	
Gartner [6]	x	x	x					x					x	
Grossman/Gu [7]	x	x		x				x					x	
Gruman/Knorr [8]	x	x	x		x								x	
IDC [9]	x	x				x		x					x	
Kim [10]	x	x	x	x		x		x	x				x	
McFredries [11]	x	x	x	x						x				

1 [1] (Armbrust et al. 2009) [2] (Breiter et al. 2008) [3] (Briscoe et al. 2009)
[4] (Buyya et al. 2008) [5] (Foster et al. 2008) [6] (Plummer et al. 2008) [7] (Grossman et al. 2009)
[8] (Gruman et al. 2008) [9] (Gens 2008) [10] (Kim 2009) [11] (McFredries 2008)
[12] (Nurmi et al. 2008) [13] (Vaquero et al. 2009) [14] (Vykoukal et al. 2009) [15] (Wang et al. 2008)
[16] (Weiss 2007) [17] (Youseff et al. 2008)

Nurmi et al. [12]	x	x	x					x				x		
Vaquero et al. [13]	x	x			x	x		x		x	x			
Vykoukal et al. [14]	x	x				x		x					x	
Wang et al. [15]	x	x	x	x										
Weiss [16]	x	x	x					x						
Youseff et al. [17]	x	x			x	x		x		x	x			
Nominations	**16**	**17**	**10**	**4**	**4**	**7**	**1**	**14**	**2**	**7**	**3**	**1**	**9**	**1**

Table 1: A comparison of various cloud computing definitions

3.2 A Definition of cloud computing

Table 1 summarizes key characteristics of cloud computing as they are understood by the respective authors. The list of definitions was compiled in May 2009 based on database queries and web searches. It is restricted to scientific contributions and statements of selected market research companies. The largest consent among the authors is spanning around the features service, hardware, software, scalability and internet/network. Furthermore, usage-bound payment models and virtualization are frequently mentioned as well. The latter, however, is considered a fundamental prerequisite (Armbrust et al. 2009) and is thus not explicitly mentioned by many authors.

Based on our literature review and our perception of cloud computing, we provide a definition that regards the concept holistically, from both the application and infrastructure perspective. Hereby we focus on the deployment of computing resources and applications, rather than on a technical description. Furthermore our definition stresses the ability of service-composition, allowing service providers to create new services by aggregating existing services, enabling customized solutions and varying distribution models. These two aspects might be driving forces, through which cloud computing could change the IT-Service Business. Thus,

> *cloud computing is an IT deployment model, based on virtualization, where resources, in terms of infrastructure, applications and data are deployed via the internet as a distributed service by one or several service providers. These services are scalable on demand and can be priced on a pay-per-use basis.*

4 The evolution of cloud computing

Cloud computing can be seen as an innovation in different ways. From a technological perspective it is an advancement of computing, applying virtualization concepts to utilize hardware more efficiently. Yet a different point of view is to look at cloud computing from an IT deployment perspective. In this sense cloud computing has the potential to revolutionize the way, how computing resources and applications are provided, breaking up traditional value chains and making room for new business models. In the following section we are going to describe the emergence of cloud computing from both perspectives.

4.1 A new computing paradigm

To judge, whether cloud computing is a new computing paradigm, one needs to reflect its development in the context of computing history. In this chapter we are going to provide a short summary of the history starting with the calculating machine, describing the development of computers and the internet and eventually the beginnings of cloud computing. We also describe a layered model of the constituting components of cloud computing

4.1.1 From the calculating machine to cloud computing

Computing history can be traced back to the invention of the first *calculating machine*. In 1623 Wilhelm Schickard was the first who documented the assembly of such a calculating machine, which worked according to the principle of Napier's bones, a sort of abacus (Freytag-Löringhoff et al. 2002). Another milestone in computing history is Charles Babbage's description of the *Analytical Engine* in 1837, a mechanical calculating machine for general purpose tasks (Babbage 1864). Additional milestones were Allan Marquand's draft of an *electrical logic machine* (1885) and Herman Hollerith's development of a *tabulating machine* (1890). The electrical logic machine was first realized in 1936 by Benjamin Burack (1949).

However, the actual history of the modern computer began with Konrad Zuse's construction of the *Z3* in 1941. Zuse's machine was the first functioning digital computer, it was based on the binary digit system, programmable, and Turing capable (Rojas 1997). In 1945, John Mauchley and J. Presper Eckert have built *ENIAC*, an Electronic Numerical Integrator and Computer, which was considered the first fully electronic tube computer (Goldstine et al. 1946). The *ENIAC* was as well programmable and Turing capable, but was based on the decimal system. Depending on the definition of the term "Computer", either the *Z3* or the *ENIAC* is considered the first Computer in the world.

With the invention of the transistor in 1947 advancements in the development of the Computer emerged at a good pace. In 1957 IBM introduced the *704* as the first mass produced mainframe computer with floating-point arithmetic. Eventually, in 1964 the IBM *System/360* followed. The highlight of this product family was that peripheral components were exchangeable and that the software was executable on all computers of this product family (Bashe 1986). Further developments and the miniaturization of the mainframe computers eventually lead to independent machines, so called Minicomputers such as DEC's *PDP-8* in 1964 or Xerox's *Alto* in 1974 (Freiberger et al. 2000).

The development of the *Personal Computer (PC)* began as recently as in the 1970ies, with the construction of the first microprocessor 4004 in 1969 and the later 8008 in 1971 by Intel. The latter was the basis for the first Home Computer, the *Micral* by André Thi Truong in 1973 (Freiberger et al. 2000). Construction manuals for the *TV Typewriter* (Lancaster 1973) or the *Mark 8* (Titus 1974) were primarily targeting hobbyists. In 1975 MITS was selling the *Altair 8800* as construction set. It was one of the first Home Computers, which for Microsoft has developed a BASIC Interpreter (Freiberger et al. 2000). More and more Home Computers followed, such as by Apple, Atari, Commodore and others. As late as 1981 IBM entered this market segment and coined the name *Personal Computer (PC)*. Microsoft has developed the operating system for the IBM-PC, which soon became the standard platform, with which many PC manufacturers where compatible with (Freiberger et al. 2000). Since then the development and diffusion of PC's gained pace, significant performance leaps could be achieved, graphical user interfaces were established and the continuing miniaturization eventually lead to the development of laptops and mobile devices.

Another important milestone was the development of the *Internet*. This can be traced back to a research project at the Advanced Research Projects Agency (ARPA). A communication system, which would stay available if one of its nodes would be broken, was developed in 1969, on behalf of the US ministry of defense. Eventually the ARPAnet was developed out of this project. In 1981 around 200 institutions were connected to this network. In 1983 the net's protocol was switched to TCP/IP, which made it possible to connect whole subnets to the ARPAnet. This network of networks was soon called *Internet*. While at the beginning it was primarily used for military and scientific purposes, its opening and commercialization began in 1988 with services like mail, telnet and usenet (Freiberger et al. 2000).

However, the Internet achieved its real breakthrough with Tim Berners-Lee's invention of the *World Wide Web* in 1989. Tim Berners-Lee conceptualized an information management system for the European Organization for Nuclear Research (CERN), which was based on Hypertext, a network structure, where knowledge entities are referenced through logical references, so called hyperlinks. Traditional hyper textual structures are for example content tables or cross references. The modern hypertext concept can be traced back to Vannevar Bush (1945). With the increasing diffusion of the web browser *Mosaic*, the *World Wide Web* eventually gained great popularity (Freiberger et al. 2000; Berners-Lee 1989).

Further increasing bandwidths and technologies like Java, PHP or Ajax made it possible to develop more and more elaborate, interactive websites. Due to this development, we can today find many multimedia websites, online shops and numerous applications that are deployed in the Internet. Some examples are route planners, communication platforms, social networks and even whole office applications like word processors or spread sheet applications. This deployment concept, usually referred to *Software-as-a-Service* gained popularity around the year 2000 (Finch 2006; Bennett et al. 2000). Similar deployment concepts were developed for the deployment of hardware resources, especially computing power and storage. Primarily in academia *Grid Computing* got established as such a concept already at the beginning of the 1990ies (Foster et al. 2003). The term Cloud Computing was coined in 2007, typically refereeing to a joint hardware and software deployment concept. First research initiatives were started by Google and

IBM, in cooperation with six American universities (Lohr 2008). Figure 1 gives an overview of important milestones of computing history.

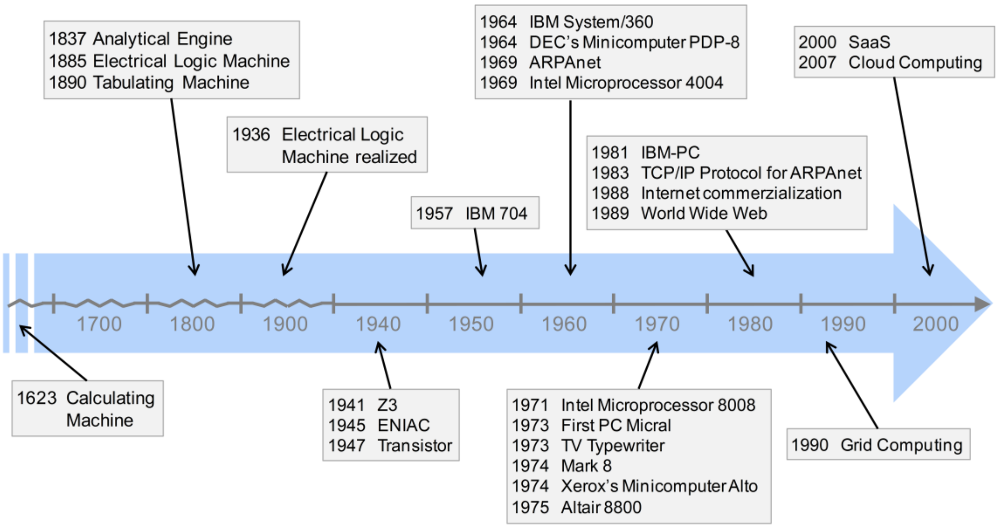

Figure 1: Milestones of computing history (own exhibit)

When looking back at the operating models of the 60ies and 70ies, one realizes that with cloud computing an old trend is coming back: centralized, shared computing resources. The so called time-sharing concept, where idle CPU times where dynamically distributed to several users, can be traced back to John McCarthy in 1957 (Lee et al. 1992). In the early days of data processing, computers as well as their operation were expensive. Therefore the operators were looking for ways to utilize them in the best way possible. Thus, companies with large mainframe computers offered their computing resources to external users. This eventually led to the foundation of independent service providers which specialized on the deployment of computing resources. Companies that offered time-sharing-systems were for example General Electric's Information Service Company (GEISCO), IBM's subsidiary The Service Bureau Corporation or Tymshare Inc. Users of these time-sharing services could access the mainframe computer through dial-up connections, using terminals that were simple input/output devices. The mainframe computers typically provided a complete working environment, including different software packages, programming environments, file storage and printing services. Pricing models were commonly based on a fixed rental fee for the terminal and variable costs for connection time, consumed CPU time and storage usage (Lee et al. 2000; Greaves 2008). With the increasing diffusion of more and more powerful PCs data processing and data storage shifted to those local devices. In the 1990ies, the next trend could be observed, back again towards a centralization of information technology, especially a centralization of data storage. This trend appears to be continuing with cloud computing.

The operators of today's datacenters, such as Amazon or Google, are confronted with a similar situation as it has been in the 1960ies and 1970ies. They as well strive to utilize their immense resources in a better way. Topical approaches, such as virtualization are efficient means to grant third party users to dynamically access their infrastructure and harness computing power and storage capacities. Platforms that were developed for cloud computing enable the development of applications that are deployed on the hardware of those cloud computing infrastructure providers.

4.1.2 A layered model of cloud computing

Cloud computing is based on a set of many pre-existing and well researched concepts such as distributed and grid computing, virtualization or Software-as-a-Service (SaaS). Although, many of the concepts do not appear to be new, the real innovation of cloud computing lies in the way it provides computing services to the customer. Various business models have evolved in recent times to provide services on different levels of abstraction. These services include providing software applications, programming platforms, data-storage or computing infrastructure.

Classifying cloud computing services along different layers is common practice in the industry (Reeves et al. 2009; Sun Microsystems 2009; Kontio 2009). Wang et al. for example describe three complementary services, Hardware-as-a-Service (HaaS), Software-as-a-Service (SaaS) and Data-as-a-Service (DaaS). These services together form Platform-as-a-Service (PaaS), which is offered as cloud computing (Wang et al. 2008). In an attempt to obtain a comprehensive understanding of cloud computing and its relevant components, Youseff, Butrico and Da Silva (2008) were among the first who suggested a unified ontology of cloud computing. According to their layered model (see Figure 2), cloud computing systems fall into one of the following five layers: applications, software environment, software infrastructure, software kernel, and hardware. Each layer represents a level of abstraction, hiding the user from all underlying components and thus providing simplified access to the resources or functionality. In the following section we are going to describe each layer of Youseff's, Butrico's and Da Silva's (2008) model.

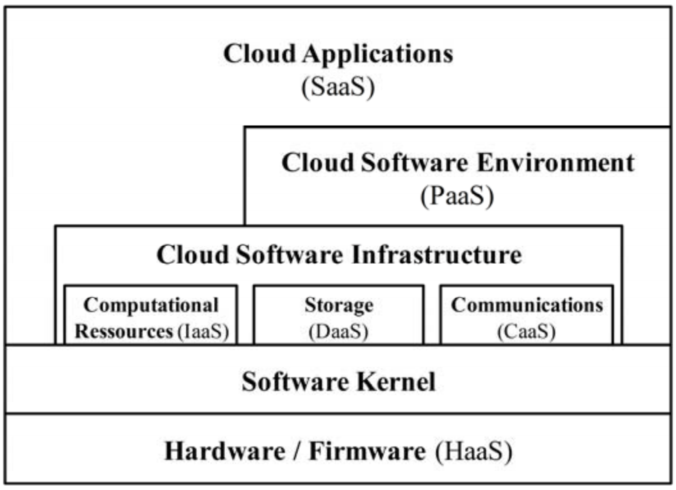

Figure 2: The layers of cloud computing (Youseff et al. 2008)

4.1.2.1 Cloud Application Layer

When it comes to user interaction, the cloud application layer is the most visible layer to the end-customer. It is usually accessed through web-portals and thus builds the front-end, the user interacts with when using cloud services. A Service in the application layer may consist of a mesh of various other cloud services, but appears as a single service to the end-customer. This model of software provision, normally also referred to as *Software-as-a-Service (SaaS),* appears to be attractive for many users. Reasons for this are the reduction of software and system maintenance costs, the shift of computational work from local systems into the cloud, or a reduction of upfront investments into hardware and software licenses. Also the service provider has advantages over traditional software licensing models. The effort for software upgrades is reduced, since patches and features can be deployed centrally in shorter cycles. Depending on the pricing model a continuous revenue stream can be obtained. However, security and availability aspects are issues that still need to be addressed. Also the migration of user data is a task that should not be underestimated.

Examples for applications in this layer are numerous, but the most prominent might be Salesforce's Customer Relationships Management (CRM) system[2] or Google's Apps, which include word-processing, spreadsheet and calendaring[3].

4.1.2.2 Cloud Software Environment Layer

The cloud software environment layer (also called software platform layer) provides a programming-language environment for developers of cloud applications. The software environment also offers a set of well-defined application programming interfaces (API) to utilize cloud services and interact with other cloud applications. Thus developers benefit from features like automatic scaling and load balancing, authentication services, communication services or graphical user interface (GUI) components. However, as long as there is no common standard for cloud application development, lock-in effects arise, making the developer dependent on the proprietary software environment of the cloud platform provider. This service, provided in the software environment layer is also referred to as *Platform-as-a-Service (PaaS)*.

A known example of a cloud software platform is Google's App Engine[4], which provides developers a Phyton runtime environment and specified APIs to develop applications for Google's cloud environment. Another example is Salesforce's AppExchange platform[5] that allows developers to extend the Salesforce CRM solution or even develop entire new applications that runs on their cloud environment.

As we will highlight in Chapter 4.2.2.2 one can also look at the cloud platform from a value network or business model perspective. In that sense, the cloud platform can act as a market place for applications.

---

[2] http://www.salesforce.com
[3] http://apps.google.com
[4] http://code.google.com/intl/de-DE/appengine
[5] http://sites.force.com/appexchange/home

4.1.2.3 Cloud Software Infrastructure Layer

The cloud software infrastructure layer provides resources to other higher-level layers, which are utilized by cloud applications and cloud software platforms. The services offered in this layer are commonly differentiated into computational resources, data storage, and communication.

Computational resources in this context are usually referred to as *Infrastructure-as-a-Service (IaaS)*. Virtual Machines are the common form of providing computational resources to users, which they can fully administrate and configure to fit their specific needs. Virtualization technologies can be seen as the enabling technology for IaaS, allowing data center providers to adjust resources on demand, thus utilizing their hardware more efficiently. The downside of the medal is the lack of a strict performance allocation on shared hardware resources. Due to this, infrastructure providers can’t give strong performance guarantees which results in unsatisfactory service level agreements (SLA). These weak SLAs propagate upwards in the cloud stack, leading to possible availability problems of cloud applications. The most prominent examples of IaaS are Amazon’s Elastic Compute Cloud[6] and Enomalism’s Elastic Computing Infrastructure[7]. There are also some academic open source projects like Eucalyptus[8] and Nimbus[9].

In analogy to computational resources data storage within the cloud computing model is offered as *Data-Storage-as-a-Service (DaaS)*. DaaS allows users to obtain demand-flexible storage on remote disks which they can access from everywhere. Like for other storage systems, trade-offs must be made between the partly conflicting requirements: high availability, reliability, performance, replication and data consistency, which in turn are manifested in the service providers SLAs. Examples of DaaS are Amazon’s Elastic Block Storage (EBS)[10] or its Simple Storage Service (S3)[11] and Rackspace’s Cloud Files[12]. In addition, to simple storage space, data can be offered as service as well. Amazon for example offers the human genome or the US census as public data sets to use for other services or analytics[13].

A fairly new idea is *Communication-as-a-Service (CaaS)*, which shall provide quality of service ensured communication capabilities such as network security, dedicated bandwidth or network monitoring. Audio and video conferencing is just one example of cloud applications that would benefit from CaaS. So far this service is more of a research interest than in commercial use. However, Microsoft’s Connected Service Framework (CSF)[14] can be counted into this class of services.

---

[6] http://aws.amazon.com/ec2
[7] http://www.enomalism.com
[8] http://www.eucalyptus.com
[9] http://workspace.globus.org
[10] http://aws.amazon.com/ebs
[11] http://aws.amazon.com/s3
[12] http://www.rackspacecloud.com/cloud_hosting_products/files
[13] http://aws.amazon.com/publicdatasets/
[14] http://msdn.microsoft.com/en-us/library/bb931207.aspx

As Figure 2 shows, cloud applications must not necessarily be developed upon a cloud software platform, but can also run directly on the cloud software infrastructure layer or even the software kernel, thus by-passing the aforementioned layers. Although this approach might offer some performance advantages, it is directly dependent on lower level components and does not make use of development aids such as the automatic scaling provided by the cloud software platform.

4.1.2.4 Software Kernel Layer

The software kernel layer represents the software management environment for the physical servers in the datacenters. These software kernels are usually implemented as OS kernel, hypervisor, virtual machine monitor or clustering middleware. Typically this layer is also the level where grid computing applications are deployed. Globus[15] is an example of a successful grid middleware. At this layer, cloud computing can benefit from the research already undertaken in the grid computing research community.

4.1.2.5 Hardware / Firmware Layer

At the bottom end of the layered model of cloud computing is the actual physical hardware, which forms the backbone of any cloud computing service offering. Hardware can also be subleased from datacenter providers to, normally, large enterprises. This is typically offered in traditional outsourcing plans, but in an as-a-service context also referred to as *Hardware-as-a-Service (HaaS)*. One example of this is IBM's Managed Hosting service[16].

With regard to the layered model of Youseff, Butrico and Da Silva (2008), described above, cloud computing can be perceived as a collection of pre-existing technologies and components. Therefore we see cloud computing as an evolutionary development and re-conceptualization, rather than a disruptive technological innovation. In our opinion cloud computing is rather an innovation in the delivery model of IT services, as we have highlighted it in our definition (see Chapter 3.2). Therefore we are showing the evolution of cloud computing in the context of IT provisioning in the following chapter.

4.2 The changing IS delivery model

The provision of IT resources in enterprises is closely linked with the general consideration whether information and communication technology should be kept in the enterprise or be sourced from external providers – a question that is established in business administration research as "make or buy" decision or vertical design (Behme 1995; Dillmann 1996). In recent years, the option to outsource IT services to an external service provider has grown in importance, because of cost, quality, flexibility and competency advantages. Outsourcing has become one of the most important organizational concepts in recent decades, especially in the light of the rapid development of information technology (Matiaske et al. 2002).

---

[15] http://www.globus.org
[16] http://www-935.ibm.com/services/de/index.wss/offering/ebhs/a1007253

To understand the evolution of cloud computing, a short summary of the history of outsourcing research is helpful. This might also help to rank and evaluate cloud computing in the context of IT service provisioning, from traditional IT provisioning models towards new concepts of IT service provision such as cloud

4.2.1 Cloud computing as an evolution of outsourcing

Although outsourcing has been an established topic since decades and the core of the concept is still around the lasting question of IT provisioning, the focus of the issue has quite shifted over time. At the beginning of the outsourcing phenomenon the focus has been on the *decision between an internal or external provision of IT services* and the *subject of outsourcing* (infrastructure, applications and processes). Later, the strategic outsourcing decision of Kodak in 1989 lead to a more differentiated approach, addressing the topic of *vertical design*. As a first step the motivation behind the *pro or contra outsourcing decisions* has been investigated. The central motives for outsourcing decisions are still mainly economical benefits, in particularly flexibility of costs and cost savings, technological advantages, innovation, strategic aims, and business-oriented advantages, such as an increasing service quality or an increasing flexibility of the business (Bongard 1994; Grover et al. 1994).

Following the discussion about outsourcing motives and potential benefits and risks the question of the appropriate *scope of outsourcing* became an issue that led to the distinction between selective and total outsourcing (Lacity et al. 1993). Within short time this has led to the consideration of what benefits and what performance advantages can be gained through an external sourcing of IT services. It was the question, which *efficiency gains* could be obtained through outsourcing, compared to the internal operation of IT (Loh et al. 1995). These questions often remained unanswered and the efficiency of outsourcing was very difficult to prove which resulted in a backward movement towards *insourcing or backsourcing*. Despite criticism the organizational concept of outsourcing has been established and today the design parameters of a successful outsourcing project are of interest. So far the focus has mainly been the *design of the contract* between the outsourcing partners (Saunders et al. 1997). Only recently, the awareness has developed that the contract on its own is not able to completely cover and specify the complexity of an outsourcing project. This is especially true, because the subject-matter of the contract, “information technology”, is a very volatile, fast changing asset and therefore requires flexibility in the outsourcing relationship (Häberle et al. 2005). Since that, new approaches to the “relationship management”, i.e., the maintenance of a good outsourcing relationship, are now seen as the key factor to a successful outsourcing project (Goles et al. 2005; Leimeister et al. 2008). Figure 3 summarizes the evolution of the outsourcing concept:

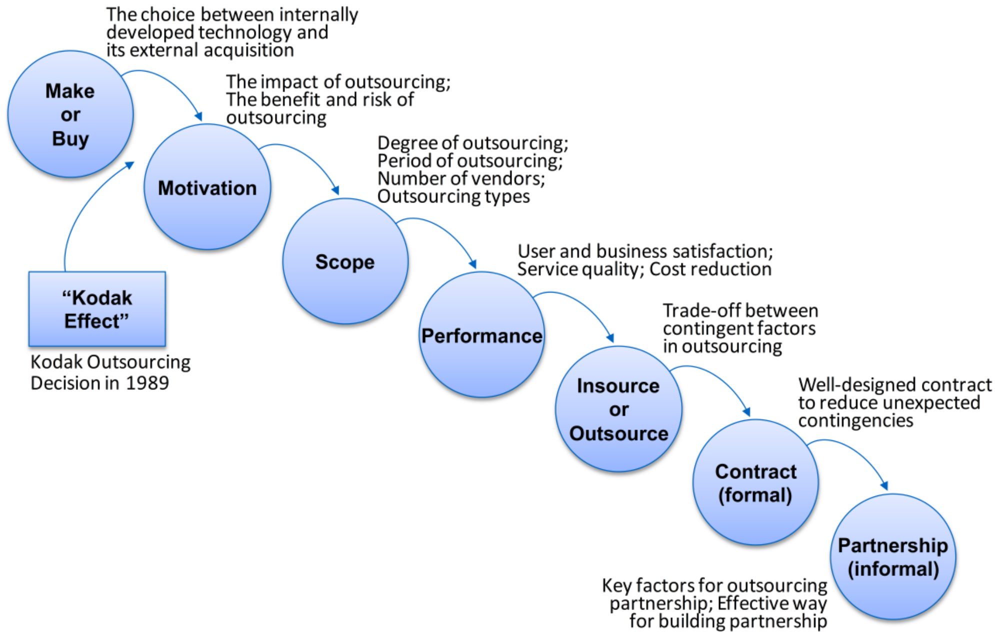

Figure 3: The evolution of external IT provisioning (Lee et al. 2003)

The correlation between cloud computing and outsourcing is best illustrated by taking current challenges into account. On the one hand, customers expect a cost-effective and efficient delivery of IT services from their service providers, paid on demand. On the other hand, more and more customers demand innovations or the identification of a customer-specific innovation potential from their service providers (Leimeister et al. 2008). From these general conditions, the new phenomenon of cloud computing has developed. New infrastructure providers that were previously active in other markets, such as Amazon or Google, have developed new business models to market their former byproducts (e.g., large storage and computing capacity) as new products. With this move, they entered the traditional outsourcing value chain (see Figure 4) and steped into competition with established outsourcing service providers. These new service providers offer innovative ways of IT provisioning through pay-per-use payment models and help customers to satisfy their needs for efficiency, cost reduction and flexibility. In the past the physical resources in traditional outsourcing models have been kept either by the customer or the provider. On the contrary, cloud computing heralds the paradigm of an asset free provision of technological capacities.

4.2.2 A value chain comparison

A value chain describes the interactions between different business partners to jointly develop and manufacture a product or service. Here, the manufacturing process is decomposed into its strategically relevant activities, thus determining how competitive advantages can be achieved. Competitive advantages are achieved by fulfilling the strategically important activities cheaper or better than the competition (Porter 1985). A value chain does not only contain different companies but also different business units inside one organization that jointly produce a product or service. The manufacturing process is seldom strictly linear and, thus, is often not seen as a value chain but rather as a value network. It is a network of relationships that generates economical value and other advantages through complex dynamical exchanges between companies (Allee 2002). Especially with regard to new internet services, value networks are often understood as a network of suppliers, distributors, suppliers of commercial services and customers that are linked via the internet and other electronic media to create values for their end customers (Tapscott et al. 2000).

4.2.2.1 Traditional IT Service Outsourcing Value Chain

In traditional IT service outsourcing the value chain is usually divided into the areas infrastructure, applications and business processes, which can be complemented by strategy and consulting activities (see Figure 4). In each of these four value chain steps the whole cycle of IT-services, often referred to as "plan, build, run", must be supported and implemented. Thus, single aspects of individual value chain steps may be outsourced, such as the development of applications. Purchasing and operating IT hardware as well as hosting can be further divided into services that are done by the customer himself and such that use resources of a hosting provider. Here, the myriad possibilities of combination may lead to complex outsourcing relationships.

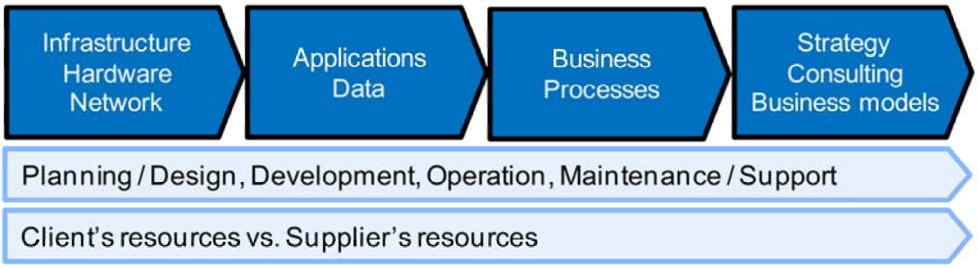

Figure 4: A traditional IT service outsourcing value chain (own exhibit)

4.2.2.2 Cloud Computing Value Chain

A general trend from products to services can be observed (Jacob et al. 2008). This trend is not only restricted to the IT world, but becomes evident also in many other industries. In the transport industry, for example, the service offering is mobility, instead of solely cars. The trend does not only lead to more outsourcing, but also from the classical hardware-based outsourcing of data centers to computing as a service (see Chapter 4.1.2.3). A similar trend can be found in the software business, which leads away from delivering software products off the shelf towards offering software as a service (see Chapter 4.1.2.1). Cloud computing links these two areas of a stronger service-oriented hardware outsourcing to the "as-a-service" concept for software. Here, cloud computing shows two big facets: infrastructure-based services are now offered dynamically to the needs of customers, often referred to as utility computing, where the customer is charged according to its actual usage.

Secondly, new cloud computing platforms emerged, to integrate both hardware and software as-a-service offerings. These platforms allow creating new, single as well as composed applications and services that support complex processes and interlink multiple data sources. From a technical point of view these platforms provide programming and runtime environments to deploy cloud computing applications (see Chapter 4.1.2.2). Looking at these platforms from a value chain perspective, they can be perceived as some kind of market place, where various cloud computing resources from different levels (infrastructure, platform services and applications) are integrated and offered to the customer. By composing different services, complex business processes and can be supported and accessed via a unified user interface. The as-a-service concept of cloud computing allows to develop new complex service-oriented applications that consist of a mixture of on-premise and off-premise services as well as pure cloud applications. Examples, how different business models utilize the new concept provided with cloud computing are given in Chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**.

From the layers of the cloud computing services model, described in chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**, we can derive three major actors within the value network: the service provider, the platform provider and the infrastructure provider. The infrastructure provider supplies the value network with all the computing and storage services needed to run applications within the cloud. The platform provider offers an environment within which cloud applications can be deployed. He also acts as some kind of catalogue or market within which applications are offered to the customer through one simple portal. The service provider develops applications that are offered and deployed on the cloud computing platform. As we especially want to highlight the aspect of service composition, we have added the aggregator role to the simplified cloud computing value network depicted in Figure 3. The aggregator is a specialized form of the service provider, offering new services or solutions by combining pre-existing services.

Within this value network value is created by providing services that are valuable for other participants of the network. Infrastructure services for example are essential for all other actors within the value network, who consume this service to provide their service offering. All the actors within the value network exchange services for money, add value for other actors through service refinement and eventually provide services that fulfills the customers' needs. As it can be observed in practice, one company of course also act in more than one roles. Salesforce for example is platform provider (AppExchange) and application provider (CRM) at the same time[17]. It can also host its own infrastructure or partly source it from third party infrastructure providers. Various service providers can offer their applications on the Salesforce platform which customers can utilize in conjunction with or separately of Salesforce's CRM solution. Aggregators might combine different services to easily provide a customized solution for the customer.

---

[17] http://www.salesforce.com

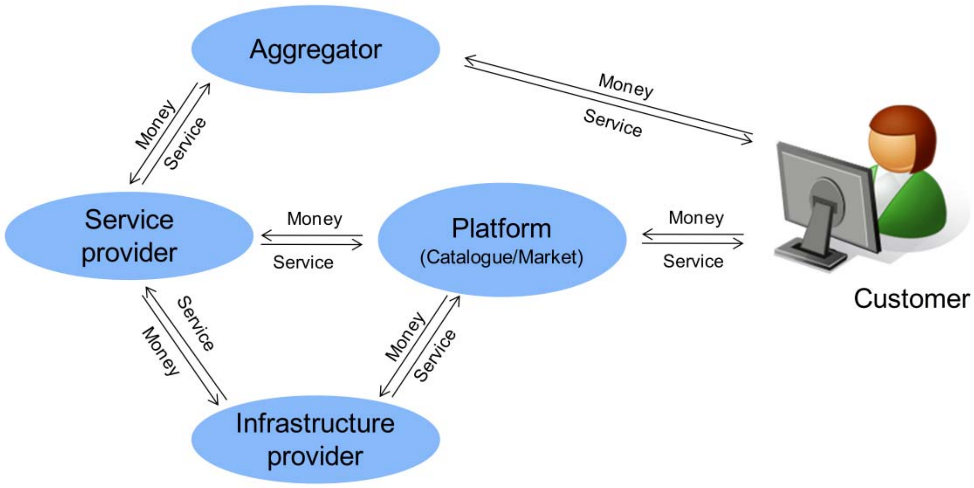

Figure 5: A simplified value network of cloud computing (own exhibit)

4.2.2.3 Comparison

Through an increased service orientation and a continuing technical standardization, the classical value chain has broken up. The model of "single-provider, one-stop provision of outsourcing" is replaced by a network of different service providers, offering a wide range of services and products on different levels.

The main characteristics of cloud computing, from a users perspective, compared to traditional IT outsourcing is the flexible deployment of virtual and asset-free resources and services. This model allows the implementation of flexible, pay-per-use business models. Comparing cloud computing with classical outsourcing shows how the value chain has broken up and how fine-grained services can be offered. This allows service providers, to provide existing customers a new flexibility, and to access entirely new customer groups with new services and business models. In addition, the cloud computing model allows modifying existing services without large investments, extending them and offering them with new business models. JungleDisk[18] for example uses the hardware-related infrastructure services of Amazon to offer user-friendly storage services for end-users.

4.3 What is new about cloud computing?

The computer pioneer John McCarthey has already predicted back in 1961 that "*computation may someday be organized as a public utility*"(in Foster et al. 2008). This statement and the reflections on computing history (see Chapter 4.1.1) show that cloud computing is no entirely new idea. Critics may even say that cloud computing is simply another name for grid computing. However, although cloud computing and grid computing have a lot in common, there are also some differences. According to Foster (2002), grid computing describes "*a system that coordinates resources which are not subject to centralized con-*

[18] http://www.jungledisk.com

trol, using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service“. Foster et al. (2008) identified the main differences regarding security aspects, programming, compute and data model, abstractions, applications and the business model.

Today’s clouds usually do not focus on the coordination of distributed infrastructure resources that are under the control of various parties. Instead, cloud computing providers commonly manage their own infrastructure that is probably more homogeneous than that of a typical grid. The reason for this lies in the different objectives of the two concepts. While cloud computing addresses Internet-scale computing problems, utilizing a large pool of computing and storing resources, grid computing aimed at large-scale computing problems by harnessing a network of resource-sharing commodity computers, dedicating resources to a single computing problem (Foster et al. 2008). Computing grids were designed upon the assumption that resources are heterogeneous and dynamic, being owned by different parties who want to remain their own administration domain and operating autonomy. This is the reason, why security is a fundamental aspect of the grid computing architecture. According to Foster et al. (2008) this is another important distinguishing factor between these two concepts.

Another difference can be found regarding the computation model. While cloud computing harnesses the power of virtualization to allow users to share resources simultaneously, most computing grids use a batch-scheduled computing model. Meaning that dedicated resources are governed by a queuing system, potentially long queuing times might occur in grid computing. Thus, most grids are not supporting interactive applications natively. Also the business models of both concepts differ significantly. The business model of grid computing is typically project oriented, where users are assigned a certain number of service units (i.e. CPU hours), which they can spend. In contrast, cloud computing is based on a consumption basis, where users can utilize as many resources as they need and only pay for what they have consumed (Foster et al. 2008).

Concluding, Foster et al. (2008) claim that both grid and cloud computing share the same vision, “*to reduce the cost of computing, increase reliability, and increase flexibility by transforming computers from something that we buy and operate ourselves to something that is operated by a third party*”. However, they predict that in future local systems and large-scale infrastructure providers will coexist, distributing load dynamically. As Figure 6 shows, Foster et al. see an overlap of grid and cloud computing, where the first one is more application oriented and the latter one more service oriented. Computing grids may be used as infrastructure basis for cloud computing, maybe they even become more service oriented. Nevertheless, grid computing will probably remain a domain of scientific computing, ensuring dedicated resources and high security standards.

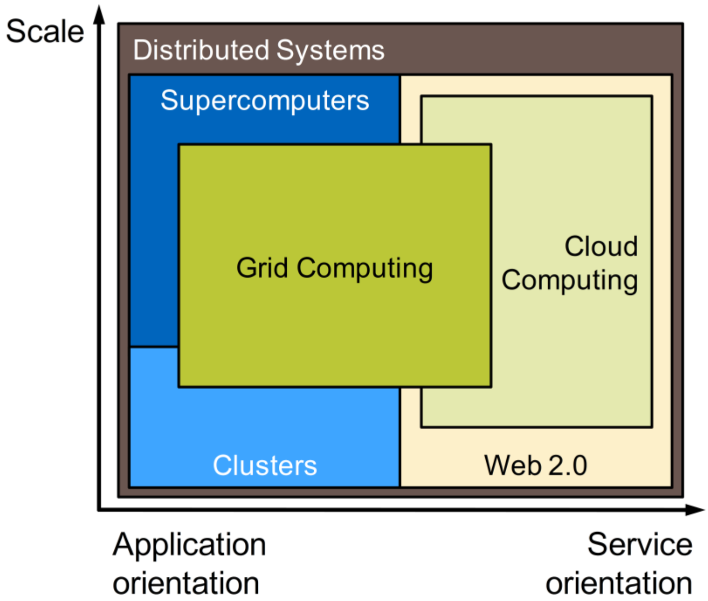

Figure 6: An overview of grid and cloud computing (Foster et al. 2008)

Another concept that is often mentioned in conjunction with cloud computing is Software-as-a-Service (SaaS). As we have described in Chapter 4.1.2.1, SaaS can be regarded as the top layer in the presented cloud computing model. Nevertheless, traditional SaaS models, occurring around the year 2000, are based on a regular (e.g. monthly) fee for service provision, independent of whether the service was used or not (Buxmann et al. 2008). Thus, traditional SaaS can be differentiated from cloud computing, which is grounded in consumption based pricing, where users only pay for what they have consumed.

Concluding, we see cloud computing as a set of preexisting technologies that offers some new features. Probably the most evident feature is the elasticity cloud computing offers. Computing resources as well as higher level services such as software can be utilized on-demand. In conjunction with this one must also mention the pay-per-use pricing model, which does not require any long term commitment and only charges for what has been consumed. In addition cloud computing is not restricted to certain users or institutions, but open for everybody, offering large computing resources to everyone with a credit card. Compared to grid computing, the cloud computing concept offers a higher level of abstraction. Development platforms increase the user friendliness by for example offering graphical user interfaces, application programming interfaces (API) and automatic scalability. Thus, cloud computing is accessible for a larger group of people, both developers and end-users.

However, in our opinion, the really new about cloud computing is not the technology behind, but the IT service ecosystem it shapes. The innovation of cloud computing is the way of how IT resources are deployed, allowing different actors to provide, consume and aggregate services on different levels. This simple deployment model reduces the entry barriers and builds the basis for a fast growing services landscape, where different actors provide various services to solve individual problems.

5 Promising applications for cloud computing

In media and advertising cloud computing often appears to be the solution for all and everything, however, it is not suitable for all domains. According to Armbrust et al. (2009) major obstacles are an insufficient availability of the service, proprietary software environments leading to data lock-in, data confidentiality and auditability issues, data transfer bottlenecks over the network, performance unpredictability, inadequate scalability of storage, bugs in large distributed systems, slow scaling of resources, legal liability issues, as well as software licensing issues.

While some of these obstacles such as bugs in distributed systems, certain security aspects or even the data transfer bottleneck will probably dissolve in the not all too distant future, others will remain due to concept inherent tradeoffs. Infrastructure providers for example can only obtain efficiency advantages over on-premise datacenters if they utilize their hardware better. By virtualizing resources, they gain the flexibility to schedule computing tasks better, however at the cost of lower availability and predictability of the service. Thus, the service requestor can be sure that his task will be completed, but not whether it is completed immediately or not. This is the reason, why cloud computing, at least at the current technological state, is for example not suitable for time critical applications.

This raises the question, for what domains is cloud computing useful? Armbrust et al. (2009) for example see opportunities in mobile interactive applications. Such services would profit from the possibility to store large datasets in the cloud, the possibility to combine different data sources and services in the form of mashups and the possibility to run the computation in the cloud, while mobile devices are not connected. A second opportunity they see in parallel batch processing and the benefits this would provide for analytics. Data conversion, video rendering or computer animations are just some examples. Also scientific simulations or business analytics appear to be predominated for cloud computing, due to its cost associativity. This means that harnessing thousand computing units for one hour costs as much as using one computing unit for thousand hours. Yet another domain Armbrust et al. mention is the extension of compute-intensive desktop applications. Matlab and Mathematica for example are able to perform extensive evaluations in the cloud. Especially applications that require high computing capacities but involve few data that needs to be transferred via the network appear to be predestinated for cloud computing.

Swaminathan et al. (2009) add some other interesting domains to this list. In their eyes cloud computing might be especially useful for software development and testing since development resources can be obtained on the fly and realistic load and performance tests can be conducted without cost-prohibitive infrastructure. Another domain is business continuity and disaster recovery since cloud computing infrastructure providers usually operate highly distributed, robust and scalable infrastructure at different geographical locations. This might be especially interesting for small and medium companies which cannot afford cost intensive fail save storage systems. Swaminathan et al. also see a potential for cloud computing in desktop productivity tools and other commodity software such as e-mail or calendaring. Finally the authors mention that cloud computing might be good to compensate peak load demands such as retailers are confronted with during holiday season or enterprises run into when compiling their annual financial statements.

To evaluate which applications are especially suitable for cloud computing, one can look at the distinctive features of this concept and estimate how a specific application would be supported or jeopardized by each feature. From the characteristics of cloud computing, discussed in Chapter 3, we have derived *demand volatility* and *computing power* as two exemplary domains, along which we can classify applications to evaluate how well cloud computing meets their requirements (see Figure 7). Demand volatility describes how continuously services or resources are required. An application that has a low demand volatility is used on a frequent basis, whereas an application with a high demand volatility is used seldom and often on an irregular basis. Computing power refers to the consumption of computing resources. Applications with a high workload require more computing power than applications that primarily focus on displaying information. In both domains cloud computing can offer significant advantages, since computing resources can be used on demand, without long-term commitment.

Our aim was to provide an intuitive graphical representation of classes of applications and their perspectives for cloud computing. As we have grouped applications into classes, individual applications could be positioned differently from where we have estimated the application class. A comprehensive evaluation of a specific application would of course also include additional requirement domains such as geographical distribution, cost impact, security, data transfer, auditability or service availability.

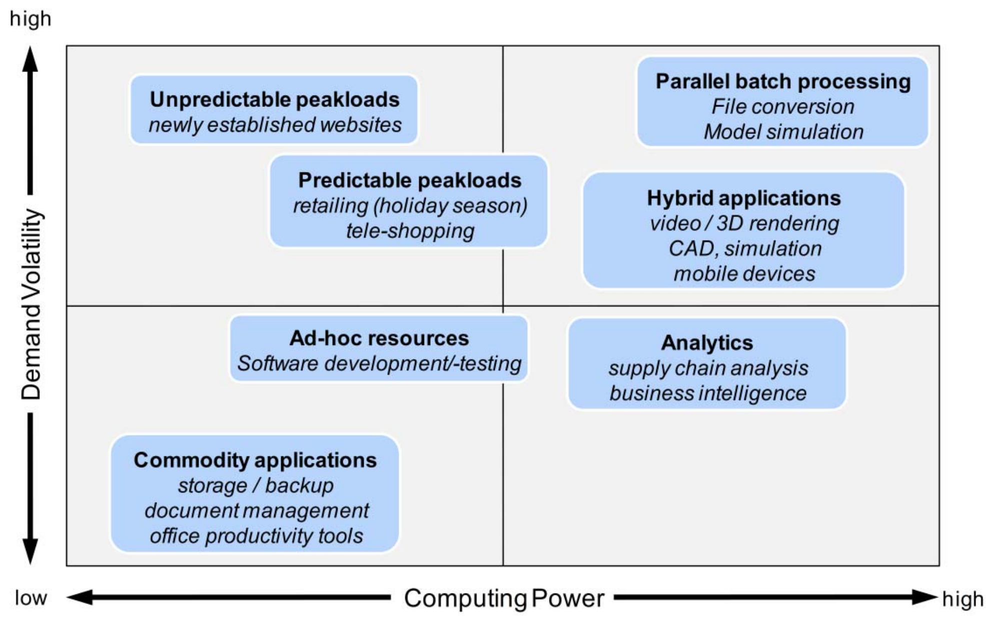

Figure 7: A classification of cloud computing applications (own exhibit)

As Figure 7 shows, applications use cloud computing features to a different extent. *Commodity applications*, such as remote backup and storage, centralized document management, or office productivity tools typically require few computing resources and are usually used on a regular basis. However, these applications might still be an interesting domain for cloud computing due to other features. Cost advantages might be realized compared to traditional licensing models and the avoidance of upfront-investments might also be of advantage, especially for startup companies or during economically hard times, where budgets are low.

Software development and –testing are examples, where resources are not required constantly. Cloud computing might be interesting in this domain, because it can provide *ad-hoc resources*. Thus, investments in, mostly underutilized, computing hardware can be avoided and realistic load and performance tests can be conducted. Cloud computing could also be a mean to simplify the often bureaucratic procurement or datacenter usage requests at large enterprises.

Peakloads are temporary high demands for resources. This demand can be *predictable*, such as in the case of retailers, whose online shops are more frequented during holiday season. The same counts for example for teleshopping providers, whose systems are facing heavier load during the time the product is advertised on TV. *Unpredictable peakloads* are especially difficult to handle with conventional hardware because of the uncertainty in capacity planning. A good example for this are newly set up websites, which's success can hardly be predicted. In both cases cloud computing is predestinated to cover these temporary resource requirements.

Analytics is another domain where cloud computing makes sense. Although the demand for analytics, especially in the business context is fairly constant, e.g. monthly statistics, it still requires considerable computing resources.

Applications that are characterized by both high computing power and high demand volatility show great potential for cloud computing. *Hybrid applications* that utilize the resources of the local computer or mobile device, but also have the ability to source additional resources from the cloud belong into this category. Thus, video and 3D rendering, CAD applications, or simulation software can provide faster results by partially outsourcing computing tasks into the cloud. Especially mobile devices, with their restricted computing power and storage limitations might also profit from cloud computing. Computations on large datasets can be conducted in the cloud, while the results can be received and visualized on the mobile device. First steps into this direction have already been undertaken, see for example Balan et al. (2003) or Palmer et al. (2009).

Yet another class of applications that shows great potential for cloud computing are parallel batch processing tasks. Otherwise long lasting scientific model simulations, calculations or file conversions can be executed in the cloud, utilizing immense computing power for a short period of time, benefiting from cost associativity.

6 Back Matter

6.1 Summary and conclusions

With the rise of the recent phenomenon of cloud computing, a myriad of terms, concepts, and approaches have emerged. Although, nearly everybody in the IT sector speaks about cloud computing, the concept remains somewhat unclear to many. In this contribution we have provided an understanding of cloud computing, based on a literature review on current definitions. From this we have derived our definition of cloud computing as an IT deployment model, that is based on virtualization, where resources, in terms of infrastructure, applications and data are deployed via the internet as a distributed service by one or several service providers. These services are scalable on demand and can be priced on a pay-per-use basis.

We have further examined the evolution of cloud computing from two different perspectives. From a technological point of view we have placed cloud computing into the context of computing history, seeing it as a reoccurring trend back to centralized computing resources. Our second perspective drew on the delivery model for information systems that changed through cloud computing. Seeing cloud computing in the context of outsourcing, we understand it as a new business model, for delivering IT resources and services, flexible, on demand and on a pay-per-use basis. We predict that this development will change the value network of IT service provision, leading to an ecosystem of IT services. Eventually we concluded with the question, which applications appear to be promising for cloud computing. In our opinion, cloud computing is no solution for everything. It offers specific features of which certain applications can profit. Other applications on the other hand are not suitable for cloud computing. Applications that are characterized by both high computing power and high demand volatility, for example, appear to have great potential for cloud computing.

6.2 Review Questions

- What are the core characteristics of cloud computing?
- What were the cornerstones in the technological development of cloud computing?
- How does cloud computing differ from traditional outsourcing?
- How can applications be evaluated for their potential use with cloud computing?

6.3 Discussion Questions

- Is it possible to establish a comprehensive definition of cloud computing?

- Is cloud computing just the next development cycle in computing history, back to some kind of mainframe computing, where the cloud is the mainframe? Will there be another counter development towards decentralized computing devices?
- Is cloud computing a disruptive innovation?
- What are requirements that need to be considered when evaluating applications for their appropriateness for cloud computing?

6.4 Further reading

- Armbrust et al. (2009) present the Berkeley View on cloud computing and provide a nice introduction to cloud computing
- Ian Foster et al. (2008) provide a detailed differentiation of cloud computing from grid computing on different aspects: business model, architecture, resource management, programming model, application model and security model.
- Briscoe et al. (2009) write about digital ecosystems in the cloud, presenting the concept of a community cloud as an alternative to cloud computing as presented here.

6.5 Acknowledgments

This research received funding from the German Federal Ministry of Economics and Technology (BMWi) under grant code 10MQ07024. This research has also been funded by Siemens IT Solutions & Services in the context of the Center for Knowledge Interchange at Technische Universität München (TUM), Germany. The responsibility for the content of this publication lies with the authors.

6.6 References

Allee, V. 2002. *The future of knowledge: Increasing prosperity through value networks*: Butterworth-Heinemann.

Armbrust, Michael, Armando Fox, Rean Griffith, et al. 2009. Above the Clouds: A Berkeley View of Cloud Computing. Berkeley: EECS Department, University of California.

Babbage, Charles. 1864. *Passages from the life of a philosopher*. London,: Longman, Green, Longman, Roberts, & Green.

Balan, RK, M Satyanarayanan, SY Park, et al. 2003. Tactics-based remote execution for mobile computing. Paper read at International Conference On Mobile Systems, Applications And Services, at San Francisco, California.

Bashe, Charles J. 1986. *IBM's early computers*, *MIT Press series in the history of computing*. Cambridge, Mass.: MIT Press.

Behme, W. 1995. Outsourcing. *WISU - Wirtschaftsstudium* 24 (12).

Bennett, K, P Layzell, D Budgen, et al. 2000. Service-based software: the future for flexible software. In *Seventh Asia-Pacific Software Engineering Conference (APSEC)*. Singapore.

Berners-Lee, Tim. *Information Management: A Proposal*. World Wide Web Consortium, 1998 1989 [cited 2009-07-02. Available from http://www.w3.org/History/1989/proposal.html.

Bongard, Stefan. 1994. *Outsourcing-Entscheidungen in der Informationsverarbeitung. Entwicklung eines computergestützten Portfolio-Instrumentariums*. Edited by W. Becker and J. Weber, *Unternehmensführung & Controlling*. Wiesbaden: DUV.

Breiter, Gerd,Michael Behrendt. 2008. Cloud Computing Concepts. *Informatik Spektrum* 31 (6):624-628.
Briscoe, G,A Marinos. 2009. Digital Ecosystems in the Clouds: Towards Community Cloud Computing. *Arxiv preprint arXiv:0903.0694*.
Burack, Benjamin. 1949. An Electrical Logic Machine. *Science* 109 (2842):610-611.
Bush, Vannevar. 1945. As We May Think. *Atlantic Monthly* (July).
Buxmann, Peter, Sonja Lehmann,Thomas Hess. 2008. Software as a Service. *WIRTSCHAFTSINFORMATIK* 50 (6):500-503.
Buyya, Rajkumar, Chee Shin Yeo,Srikumar Venugopal. 2008. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. Paper read at International Conference on High Performance Computing and Communications
Dillmann, Lutz. 1996. Outsourcing in der Produktentwicklung. Eine transaktionskostentheoretische Betrachtung der zunehmenden Fremdvergabe pharmazeutischer Produktentwicklungsaufgaben in der BRD. Dissertation, Universität Gießen, Frankfurt.
Fenn, Jackie, Nikos Drakos, Whit Andrews, et al. 2008. Hype Cycle for Emerging Technologies, 2008. In *Research*, edited by Gartner: Gartner.
Finch, Curt. 2006. The Benefits of the Software-as-a-Service Model. *Computerworld Management* (2. Januar), http://www.computerworld.com/s/article/107276/The_Benefits_of_the_Software_as_a_Service_Model.
Foster, Ian. 2002. What is the grid? a three point checklist. *GRID today* 1 (6):22–25.
Foster, Ian,Carl Kesselman. 2003. *The grid: blueprint for a new computing infrastructure*. 2 ed. Amsterdam: Morgan Kaufmann.
Foster, Ian, Yong Zhao, Ioan Raicu, et al. 2008. Cloud Computing and Grid Computing 360-Degree Compared. In *Grid Computing Environments Workshop (GCE)*. Austin.
Fowler, Geoffrey A.,Ben Worthen. 2009. The Internet Industry Is on a Cloud - Whatever That May Mean. *The Wall Street Journal* (March 26), http://online.wsj.com/article/SB123802623665542725.html.
Freiberger, Paul,Michael Swaine. 2000. *Fire in the valley : the making of the personal computer*. 2 ed. New York: McGraw-Hill.
Freytag-Löringhoff, Bruno von,Friedrich Seck. 2002. *Wilhelm Schickards Tübinger Rechenmaschine von 1623*. 5 ed. Tübingen: Univ.-Stadt Tübingen, Kulturamt.
Geelan, Jeremy. 2009. Twenty-One Experts Define Cloud Computing. *Virtualization Journal*, http://virtualization.sys-con.com/node/612375.
Gens, Frank. 2008. Defining “Cloud Services” and “Cloud Computing”. *IDC eXchange*, http://blogs.idc.com/ie/?p=190.
Goldstine, H.H.,A. Goldstine. 1946. The electronic numerical integrator and computer (ENIAC). *Mathematical Tables and Other Aids to Computation*:97-110.
Goles, Tim,W. W. Chin. 2005. Information systems outsourcing relationship factors: detailed conceptualization and initial evidence. *SIGMIS Database* 36 (4):47-67.
Google. 2009. Insights for Search. http://www.google.com/insights/search/.
Greaves, Jon. 2008. The Datacenter of the Future – What Was Once Old Is Now New Again. *The Datacenter Journal* (13. November).
Grossman, Robert L.,Yunhong Gu. 2009. On the Varieties of Clouds for Data Intensive Computing. *IEEE Computer Society Bulletin of the Technical Committee on Data Engineering* 32 (1):44-51.
Grover, Varun, Myun J. Cheon,James T. C. Teng. 1994. A Descriptive Study on the Outsourcing of Information Systems Functions. *Information & Management* 27 (1):33-44.
Gruman, Galen,Eric Knorr. 2008. What cloud computing really means. *Infoworld*, http://www.infoworld.com/print/34031.
Häberle, Oliver, Stefanie Jahner,Helmut Krcmar. 2005. Beyond the On Demand Hype: A Conceptual Framework for Flexibility in Outsourcing. Paper read at European Academy of Management Annual Conference (EURAM), Germany, May 4th - 7th 2005, at TUM Business School Munich.
Jacob, Frank,Wolfgang Ulaga. 2008. The transition from product to service in business markets: An agenda for academic inquiry. *Industrial Marketing Management* 37 (3):247-253.
Kim, Won. 2009. Cloud Computing: Today and Tomorrow. *Journal of Object Technology* 8 (1):65-72.
Kontio, Mikko. 2009. *Architectural manifesto: An introduction to the possibilities (and risks) of cloud computing*. IBM 2009 [cited 2009-07-30 2009]. Available from http://www.ibm.com/developerworks/library/ar-archman10/.

Lacity, Mary C.,R. A. Hirschheim. 1993. *Information Systems Outsourcing - Myths, Metaphors and Realities*. Chichester, New York: John Wiley & Sons.
Lancaster, Don. 1973. TV Typewriter. *Radio Electronics* 44 (9):43-52.
Lee, Jae-Nam, Minh Q. Huynh, Kwok Ron Chi-wai, et al. 2000. The evolution of outsourcing research: What is the next issue? In *33rd Annual Hawaii International Conference on System Sciences*. Island of Maui.
Lee, Jae-Nam, Minh Q. Huynh, Ron Chi-Wai Kwok, et al. 2003. IT Outsourcing Evolution--Past, Present, and Future. *Communications of the ACM* 46 (5):84-89.
Lee, JAN, J McCarthy,JCR Licklider. 1992. The beginnings at MIT. *IEEE Annals of the History of Computing* 14 (1):18-30.
Leimeister, Stefanie, Tilo Böhmann,Helmut Krcmar. 2008. IS Outsourcing Governance in Innovation-Focused Relationships: An Empirical Investigation. Paper read at 16th European Conference on Information Systems, June 9-11, 2008, at Galway, Ireland.
Loh, Lawrence,N. Venkatraman. 1995. An empirical study of information technology outsourcing: Benefits, risk and performance implications. In *Sixteenth International Conference on Information Systems*. Amsterdam.
Lohr, Steve. 2008. Google and I.B.M. Join in 'Cloud Computing' Research. *New York Times* (8. Oktober), http://www.nytimes.com/2007/10/08/technology/08cloud.html?_r=1&ex=1349496000&en=92627f0f65ea0d75&ei=5090&partner=rssuserland&emc=rss&oref=slogin.
Matiaske, Wenzel,Thomas Mellewigt. 2002. Motive, Erfolge und Risiken des Outsourcings - Befunde und Defizite der empirischen Outsourcing-Forschung. (With English summary.). *Zeitschrift fur Betriebswirtschaft* 72 (6):641-59.
McFredries, P. 2008. Technically speaking: The cloud is the computer. *Spectrum, IEEE* 45 (8):20-20.
Nurmi, D, R Wolski, C Grzegorczyk, et al. 2008. The Eucalyptus Open-source Cloud-computing System. In *Cloud Computing and Its Applications*. Chicago.
Palmer, Nicholas, Roelof Kemp, Thilo Kielmann, et al. 2009. Ibis for mobility: solving challenges of mobile computing using grid techniques. In *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*. Santa Cruz, California: ACM.
Plummer, Daryl C., Thomas J. Bittman, Tom Austin, et al. 2008. Cloud computing: Defining and describing an emerging phenomenon. In *Research*, edited by Gartner.
Porter, Michael E. 1985. *Competitive Advantage: Creating and Sustaining Superior Performance*. New York: The Free Press.
Reeves, Drue, Dan Blum, Richard Watson, et al. 2009. Cloud Computing: Transforming IT Midvale: Burton Group.
Rojas, R. 1997. Konrad Zuse's legacy: the architecture of the Z1 and Z3. *IEEE Annals of the History of Computing* 19 (2):5-16.
Saunders, Carol, Mary Gebelt,Qing Hu. 1997. Achieving Success in Information Systems Outsourcing. *California Management Review* 39 (2):63-79.
Sun Microsystems. 2009. Take your Business to a Higher Level. Sun Microsystems.
Swaminathan, Kishore, Paul Daugherty,Joseph Tobolski. 2009. What the Enterprise Needs to Know about Cloud Computing. Accenture Technology Labs.
Tapscott, Don, David Ticoll,Alex Lowy. 2000. Digital capital: harnessing the power of business Webs. *Ubiquity* 1 (13).
Titus, Jonathan. 1974. Build the Mark 8 Minicomputer. *Radio Electronics* 45 (7):29-33.
Vaquero, Luis M., Luis Rodero-Merino, Juan Caceres, et al. 2009. A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review* 39 (1):50-55.
Vykoukal, Jens, Martin Wolf,Roman Beck. 2009. Service-Grids in der Industrie – On-Demand-Bereitstellung und Nutzung von Grid-basierten Services in Unternehmen. *WIRTSCHAFTSINFORMATIK* 51 (2):206-214.
Wang, Wang, Tao Tao, Marcel Kunze, et al. 2008. Scientific Cloud Computing: Early Definition and Experience. Paper read at High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on.
Weiss, Aaron. 2007. Computing in the clouds. *netWorker* 11 (4):16-25.
Youseff, L., M. Butrico,D. Da Silva. 2008. Toward a Unified Ontology of Cloud Computing. In *Grid Computing Environments Workshop*.