Business Models in the Service World

Christof Weinhardt, Arun Anandasivam, Benjamin Blau, and Jochen Stößer
Universität Karlsruhe (TH), Germany

The authors provide a criteria catalogue to characterize cloud computing and their own Cloud Business Ontology Model to classify current product offerings and pricing models.

The popular press has recently promoted grid and cloud computing as two of the most promising trends in IT. Grid computing debuted first, in the early 1990s. Arising from the need for more computational power than clusters can provide, researchers soon found that distributed high-performance computing in virtual organizations could help them deal with large amounts of data. Research projects soon started all over the world, funded by governments as well as industry, in an attempt to fully exploit grid computing's computational advantages.

Lately, however, a new computing paradigm has emerged: cloud computing. Just as with the buzz around grid computing, this topic has generated a lot of discussion among scientists and researchers. But how does it differ from grid computing? Is it simply a new name for current technology, or does it pave the way for the commercial widespread use of large-scale IT resources?

In this article, we'll examine what cloud computing really is and introduce a new ontology for describing the different applications and business models for compute clouds. This ontology provides a clear framework to characterize and classify cloud offerings and application scenarios.

The Many Facets of Cloud Computing

Today, people use the term cloud computing in many ways—some consider it to be a pool of virtualized computer resources, whereas others say it's the dynamic development, composition and deployment of software fragments. In the former camp, Greg Boss and his colleagues[1] consider clouds to complement grid environments by supporting grid resource management. In particular, they believe clouds allow

- the dynamic scale-in and scale-out of applications by the (de-)provisioning of resources (for example, via virtualization), and
- the monitoring of resource utilization to support dynamic load-balancing and reallocations of applications and resources.

Most important, Boss and his colleagues stress that clouds aren't limited to grid environments, but also support interactive, user-facing applications, such as Web applications.

George Lawton describes types of cloud applications as Web-based applications accessed via browsers but with the look and feel of desktop programs.[2] Although this focus might be a bit too

 Published by the IEEE Computer Society

Table 1. Grid vs. cloud computing.

Criteria	Grid computing	Cloud computing
Virtualization	In its beginning	Essential
Type of application	Batch	Interactive
Application development	Local	In the cloud
Access	Via grid middleware	Via standard Web protocols
Organizations	Virtual	Physical
Business models	Sharing	Pricing (utility model; pay per use)
Service-level agreements/liability	Not yet enforceable	Essential
Control	Decentralized	Centralized (data center)
Openness	High	Low
Ease of use	Hard (until recently)	Easy
Switching cost	Low, due to standardization	High, due to incompatibilities

narrow, it corresponds to the type of applications currently emerging on Amazon's EC2 cloud platform, which are often Web 2.0 applications that need to grow and scale quickly.

For David Skillicorn, cloud computing implies component-based application construction,[3] so that instead of developing applications entirely from scratch, developers can dynamically retrieve application fragments such as simple (Web) services and third-party software libraries and assemble them in the cloud. This corresponds to the additional services that Amazon offers, such as Simple Storage and SimpleDB, or the model of Sun Microsystem's network.com platform, which offers so-called "published applications" for re-use.

Researchers like Aaron Weiss don't agree with the statement that cloud computing is a fundamentally new paradigm because it draws on existing technologies and approaches, such as utility computing, software as a service, distributed computing, and centralized data centers.[4] Cloud computing's innovation is that it combines and integrates these approaches—the combination of utility computing and data centers seems to especially differentiate cloud computing from grid computing. Although utility computing appeared earlier[5] and is, in principle, also applicable to grid computing, only recently have business models and pricing become accepted and implemented in a cloud computing context. Although some business-related research projects (such as Biz2Grid [www.biz2grid.de] and SORMA [www.sorma-project.eu]) have developed mechanisms and software to realize these concepts in grid environments, people working in science grids typically frown on such topics. Moreover, although grid computing is partly defined by its dispersed resources and decentralized control, cloud computing seems to aim for centralizing IT in data centers again, to economize on scale and scope.

Application service providers (ASPs), which emerged roughly a decade ago, seem to offer the missing link. ASPs were the first to implement business models and pricing and allowed a similar acquisition of software as clouds do today. However, the services offered in a cloud typically include simple hardware and complex (mashed-up) services, thus exceeding the former ASP concept. Furthermore, although ASP businesses never really evolved into something large scale, cloud computing already has—despite its relatively recent appearance, it has several customers and more expected. This is in part due to improved technical and more flexible frameworks evolved in Web2.0, including transmission techniques and security issues. According to Wolfgang Gentzsch, grids haven't had sustainable business models because they were largely based on public funding.[6] To him, clouds are a "useful utility that you can plug into your grid"—sort of the best of both worlds.

A Criteria Catalogue

To help solve the confusion surrounding cloud versus grid computing, we must first compare the two. We built a criteria catalogue for these two concepts and their respective technologies to determine some of the key characteristics and novelties. We found that this catalogue and our subsequent comparison provided hints as to why the cloud concept is currently deemed to be so promising.

Table 1 summarizes the key differences between grid and cloud computing, which we derived from literature as well as established grid and cloud implementations. Although the use of virtualization technologies in grid environments is still in its infancy, the abstraction from the underlying hardware pool is essential for cloud solutions. Virtualization technology lets cloud providers run multiple so-called virtual machines on a single physical machine. This has the benefit of being transparent to the application and its end users as well as making each machine customizable. From the provider's viewpoint, it makes the physical machine more efficient at load-balancing and energy consumption. Moreover, because a virtual machine essentially acts like a sandbox, it prevents side effects from cropping up between applications and users and prevents applications from compromising physical resources.

In grids, only the so-called head nodes of clusters are visible; users don't have direct access to resources. However, in clouds, each virtual machine has an IP address, so users can directly connect through their machines, thus enabling all kinds of interactive applications. The development approach differs for grids and clouds as well. In grids, users typically generate an executable file, which is then transferred and executed on a remote grid resource. Clouds, such as the network.com and salesforce.com platforms, offer ready-to-use application components so that users can dynamically assemble existing functionalities to a full application on the platform. Currently, access to grid resources happens via specific—sometimes very complex—middleware that must run on both the client and provider sides. In contrast, resource interaction in the cloud occurs via standard Web protocols. Initial participation in grid computing networks and thus the upfront investment can be very high, which might be why grid computing hasn't yet successfully established itself in many business scenarios. Accessibility and ease of use could be why cloud vendors have succeeded in boosting the numbers of nonacademic users in a relatively short time period.

Need pull quote here

Whereas grid environments are inherently decentralized and dynamic, organizational structures in clouds remain rather static. In grids, the interfaces between the participants have become increasingly standardized, which means that grids are actually quite open and, from an organizational perspective, easily accessible to new participants and providers. In clouds, relatively few (and primarily centralized) providers have so far offered resources via proprietary interfaces. This key point results in vastly different switching costs: although it's relatively easy for any grid user to switch from one grid provider's resources to another, cloud providers have no interest in standardizing, which ultimately makes it harder for potential customers to switch among providers.

The static and rather centralized cloud scenarios, however, also have a big advantage. Cloud providers can offer service-level agreements (SLAs) to their customers, which encourages the use of clouds even for mission-critical industrial services because SLAs with a single provider are easily enforceable. SLA enforcement in the grid environment is still in the research stage: the submission of a grid resource request to an allocated provider generally isn't known a priori, requiring a dynamic, automated SLA contracting and enforcement scheme, which is hard to realize in the field.

Cloud Business Models

Current trends in cloud computing lean heavily toward the business world: companies seem increasingly motivated to focus innovative business models on various aspects of cloud computing. To attain a better understanding and a common conceptualization, we propose a Cloud Business Model Ontology to provide a hierarchical classification of different business models and some well-known representatives within the cloud.

A Cloud Business Model Ontology

As Figure 1 shows, our ontology primarily consists of three layers analogous to the technical layers in most cloud realizations: infrastructure, platform as a service, and application.

Infrastructures in the cloud. The infrastructure layer focuses on enabling technologies. Our ontology distinguishes between two categories of infrastructure business models: those providing storage capabilities and those supplying computing power. Amazon, for example, offers services based on its infrastructure as a computing service (EC2) and a storage service (S3). So far, pricing models are mostly pay per use or based

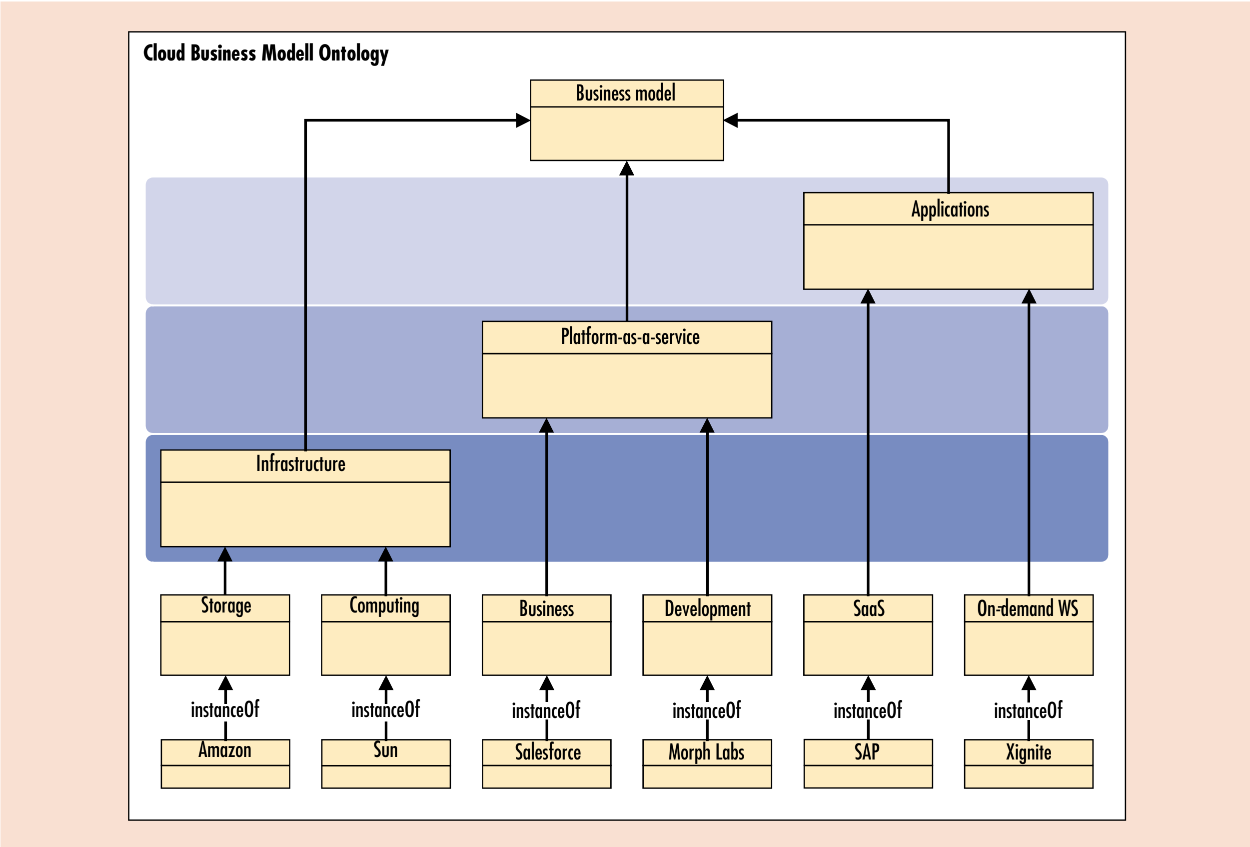

Figure 1. Cloud computing architecture. The components' location is significant: those further to the top facilitate encapsulated functionality from the layers beneath by aggregating and extending service components via composition and mashup technologies.

on subscription. In most cases, resource providers organize cloud computing infrastructures in a cluster-like structure to facilitate virtualization technologies. Nevertheless, a few business models focus on providing computing on demand through a server grid such as the Sun Grid Computing Utility (http://www.network.com). Among those that offer pure resource services, providers such as RightScale often enrich their offerings via value-added services that can manage the underlying hardware and are accessible through Web front ends.

Platforms in the cloud. The platform layer represents solutions on top of a cloud infrastructure that provide value-added services from both a technical and a business perspective, but our ontology clearly distinguishes between development and business platforms. Development platforms let developers write their applications and upload their code into the cloud, where the application is accessible and can be run in a Web-based manner. In this context, developers don't have to worry about issues such as system scalability when application usage grows. Prominent examples include Morph Labs and Google's App Engine, which provide platforms for deploying and managing Grails, Ruby on Rails, and Java applications in the cloud (www.mor.ph/ and http://code.google.com/appengine/). An additional example is BungeeLabs, which provides a platform that offers the functionality required for managing the whole Web application life cycle from development to productive provisioning (www.bungeelabs.com). Business platforms such as SalesForce with its programming language Apex and Microsoft with its business platform xRM (still in the development phase) have also gained attention by enabling the development, deployment, and management of tailored business applications (www.salesforce.com and www.xrm.com).

Applications in the cloud. The application layer is what most people recognize in cloud computing because it represents the customer's actual

Table 2. Offerings of services on demand.

Company/product	Service type	Pricing model	Ontology concept
Amazon EC2 and S3, SimpleDB, SQS, FPS, DevPay	Computing, storage, database, payment, billing	Pay per use	Infrastructure/platform as a service
Appian Anywhere	Business process management	Pay per use	Applications
Box.net	Storage	Pay per use	Applications
FlexiScale	Infrastructure	Pay per use	Infrastructure
Google App Engine	Infrastructure, Web applications	Pay per use	Infrastructure
Gmail Drive	Storage, email	Free/pay per use	Applications
MuxCLoud	Data processing (video); uses Amazon's EC2	Pay per use	Applications
Nirvanix	Storage	Pay per use	Applications
Network.com	Infrastructure	Pay per use	Infrastructure/platform as a service
OpSource	Billing	Subscription	Applications
Process Maker Live	Business process management	Pay per use	Applications
Salesforce.com	Platform	Pay per use	Platform as a service/ applications
MS SkyDrive	Storage	Free	Applications
SmugMug	Data sharing (photo)	Subscription	Applications
StrikeIron	Web services	Subscription/pay per use	Applications
XDrive	Storage	Subscription	Applications
XCalibre	Infrastructure	Subscription	Infrastructure
Zimory.com	Marketplace	Dynamic pricing	Applications

interface. This layer delivers applications via the opaque platform and infrastructure layers. Our ontology distinguishes between software-as-a-service applications and the provisioning of rudimentary Web services on demand. The most prominent examples in the software-as-a-service area are Google Apps, with their broad catalogue of Microsoft Office applications such as Word and Excel as well as easy-to-use email and calendar applications that are entirely accessible through a Web browser. An example from the B2B sector is SAP, which delivers its service-oriented business solution BusinessByDesign on a pay-per-use hosting model over the Web (www.sap.com/solutions/sme/businessbydesign/). Xignite and StrikeIron offer Web services hosted on a cloud on a pay-per-use basis as well (www.xignite.com and www.strikeiron.com).

Current Cloud Computing Offerings

Lately, we've noticed a rising number of Internet services on demand. Prominent providers such as Amazon, Google, Sun, IBM, Oracle, and SalesForce have extended their computing infrastructures and platforms to provide top-level services for computation, storage, databases, and applications, including those for email, MS Office programs, finance, media, and data processing. During our survey of current cloud offerings, we counted more than 70 providers of so-called cloud services; Table 2 gives an overview of some of them, the service types they offer, their pricing models, and a mapping to our ontology. We've categorized the offered services into service types: infrastructure, storage, database, business process management, marketplace, billing, accounting, email, data sharing, data processing, and Web services. We found the most common pricing model to be pay per use, in which customers pay a static price for a unit as they use it. The pay-per-use pricing model is simple: it associates units (or units per time) with fixed price values, and it's widely used for products (services) in which mass production and widespread delivery make price negotiation impractical.[7,8] Another pricing model is subscription, in which the customer subscribes (by signing a contract) to use a preselected combination of service units for a fixed price and a longer time frame, usually monthly or yearly.

Table 2 shows the popularity of the pay-per-use and subscription pricing models; seemingly, users and providers prefer simple, static models in which it's easy to predict payments.[9] However, Kevin Lai found that dynamic pricing policies could achieve more economically efficient allocations and prices for high-value services.[10] In a market in which cloud providers have scarce resources and thus high demand, capacity allocation depends on customer choice, classification, and appropriate pricing. Cloud providers can gain a higher revenue by offering customized products with additional services based on the same commodity.[11]

Cloud computing and its related paradigms—grid computing, utility computing, and voluntary computing—have seen their share of discussion in academia and industry. Our proposed cloud business model ontology offers a clear framework to characterize and classify cloud offerings. Cloud users and providers can use it to map products, identify customers and suppliers, and set pricing schemes. New research areas in the context of cloud computing will undoubtedly focus on the necessity for a standard cloud API, security issues, new business models, and dynamic pricing systems for complex services.[12]

References

1. G. Boss et al., *Cloud Computing*, tech. report, IBM High-Performance on Demand Solutions, 2007; http://download.boulder.ibm.com/ibmdl/pub/software/dw/wes/hipods/Cloud_computing_wp_final_8Oct.pdf.
2. G. Lawton, "Moving the OS to the Web," *Computer*, vol. 41, no. 3, 2008, pp. 16–19.
3. D.B. Skillicorn, "The Case for Data-Centric Grids," *Proc. Int'l Parallel and Distributed Processing Symp.*, IEEE CS Press, 2002, pp. 247–251.
4. A. Weiss, "Computing in the Clouds," *netWorker*, vol. 11, no. 4, 2007, pp. 16–25.
5. M.A. Rappa, "The Utility Business Model and the Future of Computing Services," *IBM Systems J.*, vol. 43, no. 1, 2004, pp. 32–42.
6. W. Gentzsch, "Grids Are Dead! Or Are They?" *GRIDToday*, 16 June 2008; www.Gridtoday.com/Grid/2381106.html.
7. P.R. Wurman, "Dynamic Pricing in the Virtual Marketplace," *IEEE Internet Computing*, vol. 5, no. 2, 2001, pp. 36–42.
8. G. Bitran and R. Caldentey, "An Overview of Pricing Models for Revenue Management," *Manufacturing & Service Operations Management*, vol. 5, no. 3, 2003, pp. 203–229.
9. P.C. Fishburn and A.M. Odlyzko, "Competitive Pricing of Information Goods: Subscription Pricing versus Pay-Per-Use," *Economic Theory*, vol. 13, no. 2, 1999, pp. 447–470.
10. K. Lai, "Markets Are Dead, Long Live Markets," *ACM SIGecom Exchanges*, vol. 5, no. 4, 2005, pp. 1–10.
11. A. Anandasivam and D. Neumann, "Managing Revenue in Grids," *Proc. Int'l Conf. System Sciences*, IEEE CS Press, 2009, pp. 1–10.
12. B. Blau et al., "Provisioning of Service Mashup Topologies," *Proc. 16th European Conf. Information Systems*, Editors: W. Golden, T. Acton, K. Conboy, H. van der Heijden, and V.K. Tuunainen, 2008, pp. 551–562.

Christof Weinhardt *heads the Institute of Information Management and Systems at the School of Economics and Business Engineering at Universität Karlsruhe, where he also serves as the vice dean of the School of Economics and Business Engineering. His research focuses on combining innovative questions from the areas of information management and engineering, service sciences, and market engineering with various computing scenarios. Contact him at weinhardt@iism.uni-karlsruhe.de.*

Arun Anandasivam *is a research assistant and PhD student in the Institute of Information Systems and Management at Universität Karlsruhe. His research work comprises pricing policies and decision frameworks of grid and cloud computing providers. Contact him at anandasivam@iism.uni-karlsruhe.de.*

Benjamin Blau *is a PhD student and research assistant in the Institute of Information Systems and Management and the newly founded Karlsruhe Service Research Institute at Universität Karlsruhe. His main research interest is in the field of service-oriented computing and mechanism design in the context of service auctions and service pricing. Contact him at blau@iism.uni-karlsruhe.de.*

Jochen Stößer *is a postdoctorate researcher in the Institute of Information Systems and Management at Universität Karlsruhe. His main interest is in the design of market mechanisms for an efficient allocation of computer resources. Contact him at stoesser@iism.uni-karlsruhe.de.*