J Grid Computing (2011) 9:3–26
DOI 10.1007/s10723-010-9171-y

Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach

Bhaskar Prasad Rimal · Admela Jukan · Dimitrios Katsaros · Yves Goeleven

Received: 9 March 2010 / Accepted: 26 October 2010 / Published online: 7 December 2010

Abstract Cloud Computing is a model of service delivery and access where dynamically scalable and virtualized resources are provided as a service over the Internet. This model creates a new horizon of opportunity for enterprises. It introduces new operating and business models that allow customers to pay for the resources they effectively use, instead of making heavy upfront investments. The biggest challenge in Cloud Computing is the lack of a de facto standard or single architectural method, which can meet the requirements of an enterprise cloud approach. In this paper, we explore the architectural features of Cloud Computing and classify them according to the requirements of end-users, enterprises that use the cloud as a platform, and cloud providers themselves. We show that several architectural features will play a major role in the adoption of the Cloud Computing paradigm as a mainstream commodity in the enterprise world. This paper also provides key guidelines to software architects and Cloud Computing application developers for creating future architectures.

Keywords Architecture · Cloud Computing · Grid Computing · Requirements · On-demand

B. P. Rimal (✉) · A. Jukan
Institute of Computer and Network Engineering, Technische Universität Carolo-Wilhelmina zu Braunschweig, Hans-Sommer-Straße 66, 38106 Braunschweig, Germany
e-mail: b.bprimal@gmail.com

A. Jukan
e-mail: jukan@ida.ing.tu-bs.de

D. Katsaros
Department of Computer and Communication Engineering, University of Thessaly, Volos 38221, Hellas, Greece
e-mail: dkatsar@inf.uth.gr

Y. Goeleven
CapGemini, Bessenveldstraat 19, 1831 Diegem, Belgium
e-mail: yves@goeleven.com

1 Introduction

The overarching goal of Cloud Computing is to provide on-demand computing services with high reliability, scalability, and availability in distributed environments. Despite this common goal, Cloud Computing [9] has been described in many different ways [2, 73] and no standard definition has been adopted until now. Two examples follow. Cisco Systems [41] defined Cloud Computing as, *IT resources and services that are abstracted from the underlying infrastructure and provided "on-demand" and "at scale" in a multitenant environment."* Recently, the Information Technology Laboratory at the National Institute of

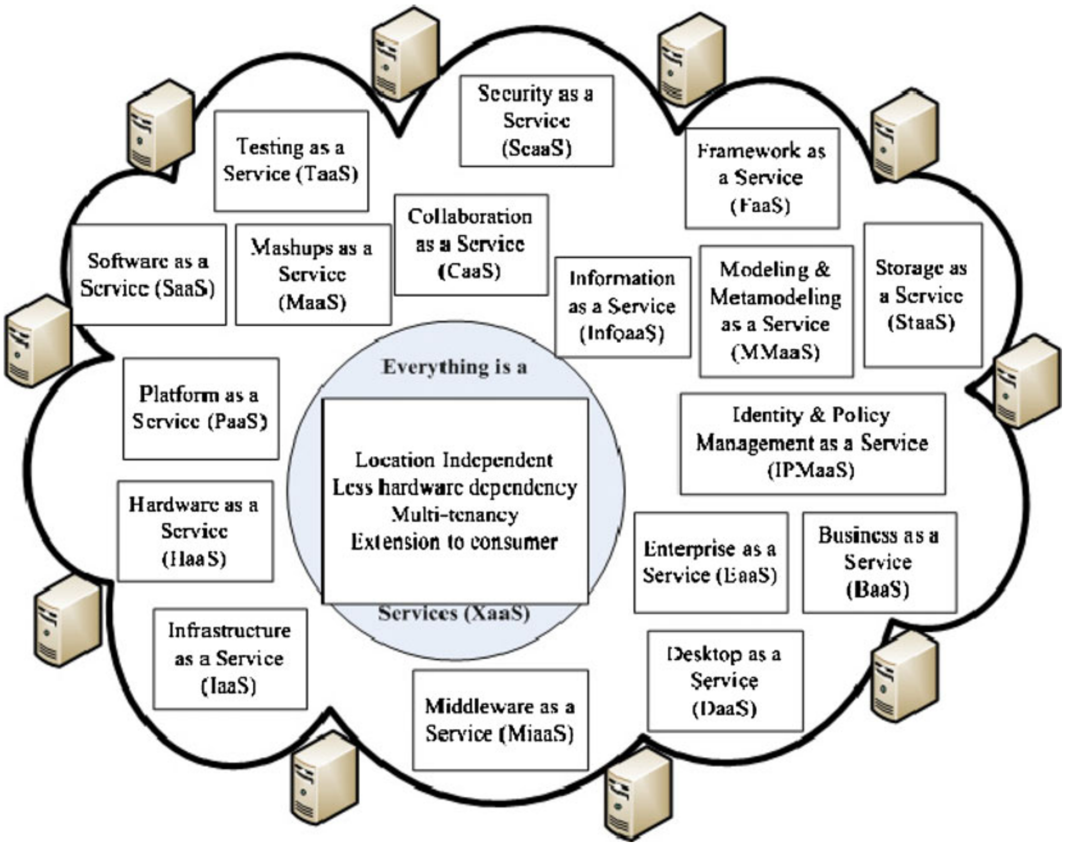

Fig. 1 Cloud computing: everything is a service model

Standards and Technology (NIST) [52] has posted a working definition of cloud computing: "*Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics (Rapid Elasticity, Measured Service, On-Demand Self-Service, Ubiquitous Network Access, Location-Independent Resource Pooling), three delivery models (Software as a Service, Platform as a Service, and Infrastructure as a Service), and four deployment models (Public Cloud, Private Cloud, Community Cloud and Hybrid Cloud)*."

Conceptually, in Cloud Computing everything is assumed as a service (XaaS) [33], such as TaaS (Testing as a Service), SaaS (Software as a Service), PaaS (Platform as a Service), HaaS (Hardware as a Service). This is illustrated in Fig. 1. To this end, a large number of cloud service providers and middleware suits have emerged, each providing different Cloud Computing services. These providers include Amazon EC2,[1] Google App Engine (GAE),[2] SalesForce.com (SFDC),[3] Microsoft Azure,[4] IBM Blue Cloud,[5] 3Tera,[6] to name a few. As we will discuss throughout the paper, the major challenge of the current Cloud Computing evolution, with multiple offerings and

[1] Amazon EC2, http://www.amazon.com/ec2/.
[2] Google App Engine, http://code.google.com/appengine/.
[3] SalesForce.com, http://www.salesforce.com/.
[4] Microsoft Azure, http://www.microsoft.com/windowsazure/.
[5] IBM Blue Cloud, http://www.ibm.com/ibm/cloud/.
[6] 3Tera, http://www.3tera.com/.

providers, is the lack of standardized APIs and usage model.

It is important to note that the concept of Cloud Computing is not new, but it represents the next evolutionary step of several initiatives carried out in the last few years, including Distributed Computing [31], Grid Computing [27], Utility Computing [61], Virtualization [4], and Server Clusters [30]. Examples of such efforts include the seminal work on Grid virtualization by Figueiredo et al. [23], lease-oriented mechanism for Grid system virtualization by Shirako [39], Autonomic Clouds on the Grid by Murphy et al. [54]. Other related work focuses on Dynamic Virtual Clustering [20], Violin [25], Virtual Workspaces [45], Virtual Cluster [28], In-VIGO [1], Grid and Cloud integration for next generation network [60] and Virtual Organization Clusters [53]. In particular, Cloud Computing is considered the business-oriented evolution of Grid Computing, which was focused on research and collaboration [29]. In addition, Cloud Computing provides a paradigm shift of business and IT infrastructure, where computing power, data storage and services are outsourced to third-parties and made available as commodities to enterprises and customers. The basic differences between Cloud Computing and Grid Computing are shown in Table 1.

In this paper, we would like to consider important architectural requirements for Cloud Computing systems from the point of view of enterprise users, such as infrastructure, storage, scalability, compliance and security. These considerations provide a guideline for creating architectural mechanism that will be helpful for software architects and developers while designing Cloud-based applications. The specific contributions in this paper are as follows:

- Delineation of motivation and issues and challenges specific to enterprise Cloud Computing.
- Classification of three layered architectural requirements for Cloud Computing, including *provider requirements*, *enterprise requirements* and *user requirements*.
- Discussion and a "vertical" analysis of the requirements common to all three layers.

The rest of this paper is organized as follows. Section 2 describes the architectural requirements of Cloud Computing systems within the three categories. Section 3 describes the discussion and a comparative analysis of the common requirements with partial mapping of requirements and finally, we conclude the paper with a summary in Section 4.

Table 1 Differences between Grid and Cloud

Parameters	Grid Computing	Cloud Computing
Focus	Predominantly on science, research, and collaboration purposes	On business and web-based applications
Resource pattern	Collaborative use of heterogeneous resources such as hardware/software configurations, access interfaces and management policies	Mostly the behavior of resource units are homogeneous and virtualized (e.g., Google and Amazon, in general contain homogeneous resources, operated under central control)
Management	Decentralized management systems, which spans across geographically distributed sites	Clouds operate as the centralized management systems from the viewpoint of users with a single access point
Business model	There is no fundamental business model for Grid	Pay-per-use/pay-as-you-go business model is a de-facto assumption for cloud services
Interoperability	Grid technology deals with the interoperability between providers	Still vendor lock-in and integration problem
Middleware	There are many middleware such as Unicore[a], gLite[b] etc. but Globus middleware [26] is a de-facto standard	There are many middleware such as Globus Nimbus [44], Open Nebula [64, 65], Eucalyptus [55] etc but no single de facto standard

[a]Unicore project, http://www.unicore.org/

[b]gLite, http://glite.web.cern.ch/glite/

2 Architectural Requirements for Cloud Computing Systems

Many Cloud Computing systems are available from industry as well as academia. Despite significant advances, there are several open issues with cloud such as security, availability, scalability, interoperability, service level agreement, data migration, data governance, trusty pyramid, user-centric privacy, transparency, political and legal issues, business service management etc. The major challenges for the Cloud Computing paradigm, however, will be the standardized API and usage model. As of today, there is no general design guideline that architects and developers can follow for Cloud-based applications. As discussed in [5, 14], defining a standard and scalable architecture will be pivotal to the success of Cloud Computing, as it will serve as a means of coordination and communication between the service providers and end users. These issues can be addressed by proper design of Cloud Computing architecture. To this end, it is important to start deriving general architectural requirements as the first step towards the future architecture of the cloud systems.

The architectural requirements are classified according to the requirements of cloud *providers*, the *enterprises* that use the cloud, and *end-users*. A three-layered classification of architectural requirements of cloud systems is illustrated in Fig. 2. As it can be seen from this figure, from the service provider's perspective, highly efficient service

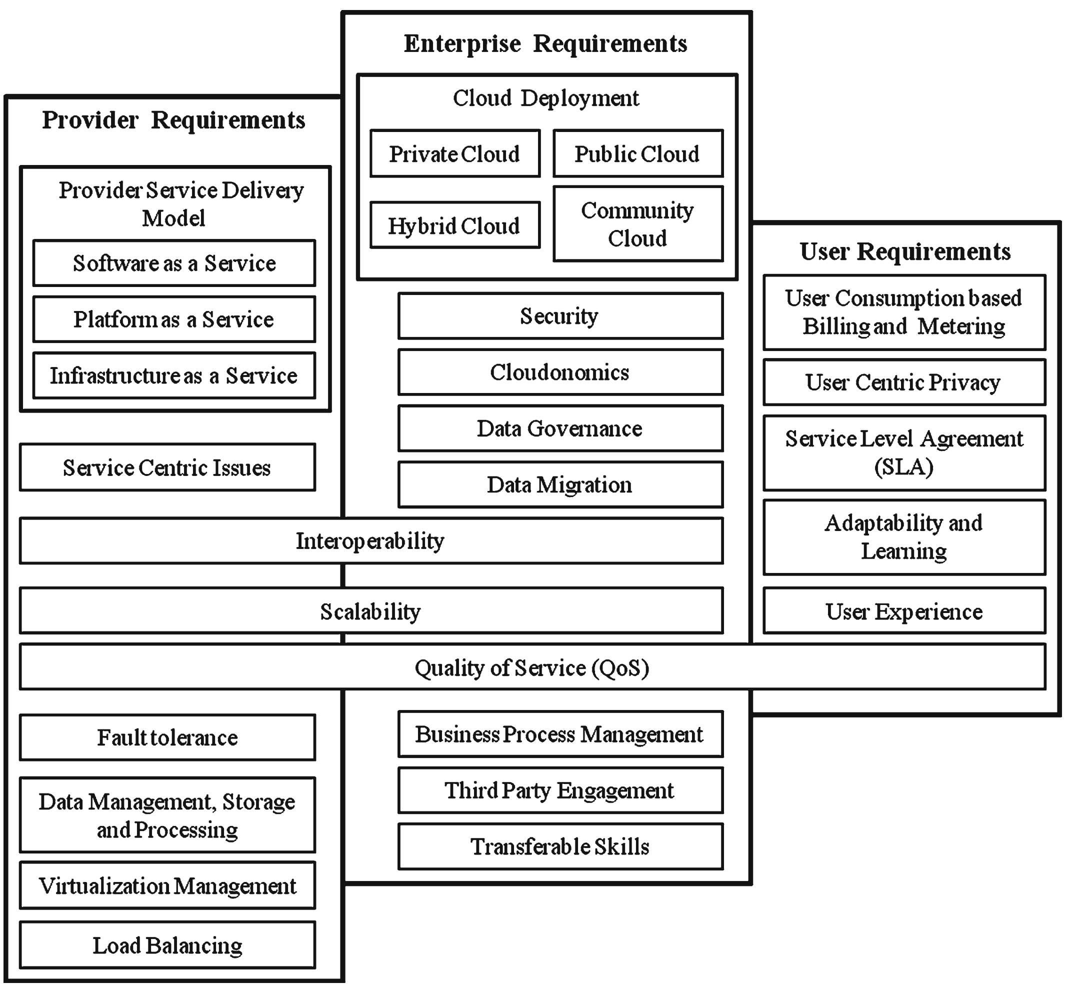

Fig. 2 Three layered architectural requirements

architecture to support infrastructure and services is needed in order to provide virtualized and dynamic services. A well-organized and secured data management and storage mechanism is herewith required along with an attractive cost model.

From the enterprise's perspective in the first place, a QoS-enabled, secure and scalable system is needed. The enterprises should have a capability of providing their business management services with internal/external interoperable mechanism which can be easily deployed within different cloud models. Finally, from the user's perspective, the basic requirement is a simplified interface with adaptability and self-learning capability that should address transparent pricing, metering and service level agreements (SLAs). A user-centric privacy, encryption/decryption will increase the stability and usability of the cloud services.

There is a correlation between those requirements. For example, if the cloud service providers are not fully aware on the issues of transparency for billing and data governance, there will always be concerns about trust. How user trusts the cloud service provider's services, depends on the level of assurance they are given. In a similar fashion, if there are data security issues in the cloud services, large corporate firms will not host their applications/data in the cloud. In addition, the aspects of security, compliance, reliability, migration, and SLAs are directly associated with the performance of cloud services. Those requirements will play a vital role in the enterprises to develop their business strategies over the cloud.

In the following subsections, we provide more details on the requirements related to each of the layers presented here, i.e., provider, enterprise and user.

2.1 Provider Requirements

According to J. Dean and S. Ghemawat [18], Google currently processes over 20 terabytes of raw web data per day. This shows the computational volume and complexity that a cloud provider has to deal with. For example, this kind of load typically increases the complexity of the architecture and thus the cost of operating a cloud provider system. To try and reduce the load, cloud service providers could for example utilize price discrimination based on several key factors such as usage patterns, bandwidth, and service level. This section describes the requirements of provider service delivery model requirements, service-centric issues, data management, storage and processing, fault-tolerance, virtualization management, and load balancing.

2.1.1 Provider Service Delivery Model

The provider service delivery model can be classified according to the services that they offer. Basically, in cloud systems three service delivery models can be considered such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). These services are usually exposed as industry standard interfaces, such as web services, service oriented architecture (SOA) [68] or REpresentational State Transfer (REST) [22] services or any proprietary services.

Software-as-a-Service (SaaS) Software as a Service (or application as a service) is a multi-tenant platform. It uses common resources and a single instance of both the object code of an application as well as the underlying database to support multiple customers simultaneously. SaaS [13, 63, 72] commonly referred to as the Application Service Provider (ASP) model, is heralded by many as the new wave in application software distribution. Examples of the key providers are the Salesforce Customer Relationships Management (CRM) system, NetSuite, and Google Office Productivity application. The major considerations in SaaS are the integration requirements with other applications. Another critical component is the composition of different types of technology like J2EE, .Net, Hibernate, Spring, Scalable Infrastructure and Services. At the application level, the important aspects of scalability, performance, multi-tenancy, configurability and fault-tolerance are primary considerations for the architect.

Platform-as-a-Service (PaaS) The idea behind PaaS is to provide developers with a platform including all the systems and environments comprising the end-to-end life cycle of developing, testing, deploying and hosting of sophisticated web applications as a service delivered by a cloud based platform. Key examples are:

- Integrated-oriented platform, i.e., a platform for realization of e-business and apps that provide a wide range of interaction between platforms, languages in use or users, such as Facebook F8, Salesforge App Exchange.
- Development-oriented platform, i.e., a platform that provides an environment for developers to develop, test and deploy their application easily such as Google App Engine, Bunzee connect, and SF force.com etc.
- Infra-oriented platform, i.e., a platform that provides developers a scalable infrastructure and storage such as Amazon EC2, Simple Storage, Simple DB etc.

Compared with conventional application development, PaaS can significantly reduce the development time, and also offers hundreds of readily available services. PaaS provides a solution for offering multiple applications on the same platform thus increasing the economy of scale and reducing complexity.

PaaS is completely reliant on the provider for complete uptime and operation. There is a high possibility of "lock-in" if PaaS needs to offer proprietary service interfaces or development languages. Therefore, PaaS providers should be aware these features as well as be capable to redefine the system discovery and services. Open Platform as a Service (OPaaS or Open PaaS) [58] is another step in the PaaS evolution. The OPaaS provides a wide range of facilities to the developers such as any programming language environment, development tools, servers, database etc.

Infrastructure-as-a-Service (IaaS) IaaS is also sometimes called Hardware as a Service (HaaS). According to Nicholas Carr [10], "*the idea of buying IT hardware—or even an entire data center—as a pay-as-you-go subscription service that scales up or down to meet your needs. But as a result of rapid advances in hardware virtualization, IT automation, and usage metering and pricing, I think the concept of hardware-as-a-service—let's call it HaaS—may at last be ready for prime time...*" This model is advantageous to the enterprise users, since they do not need to invest in building and managing the IT systems hardware. Aside from the higher flexibility, a key benefit of IaaS is the usage-based payment scheme. This allows customers to pay as they grow. Another important advantage is to buy and use the latest technology. On-demand, self-sustaining or self-healing, multi-tenant, customer segregation are the key requirements of IaaS. GoGrid,[7] Mosso/Rackspace,[8] MSP On Demand,[9] masterIT,[10] NewServers Inc[11] are the pioneer IaaS providers. Dynamically configuring and managing of network, storage and computing resources in a holistic approach is the significant challenges. In addition, there are some critical issues in the architectural strategies to incorporate IaaS successfully, for instance in answering the following questions.

- How will the applications behave due to dynamic infrastructure scaling? Can the users design their application in such a way as to maximize the scaling advantage?
- How to understand the Cloud workload (e.g., transactional database, fileserver, web server, application server, batch data processing) to design successful IaaS?
- How to asses the power consumption and environmental impact?
- How to isolate of service failure within each tenant?
- How to minimize the response time of elastic demand and maximize the throughput of requests?

2.1.2 Service-Centric Issues

Cloud Computing as a service needs to respond to the real world requirements of enterprise's IT

[7]GoGrid, http://www.gogrid.com/.

[8]Rackspace, http://www.rackspacecloud.com/.

[9]MSP On Demand, http://www.mspondemand.com/.

[10]masterIT, http://master-it.com/index.html.

[11]NewServers, http://www.newservers.com/.

management. To fulfill the requirements of enterprise's IT management, cloud architecture needs to deal with unified service-centric approach. The cloud services should be:

- Autonomic: Cloud systems/applications should be designed to adapt dynamically to changes in the environment with less human assistantship. Autonomic behavior of services can be used to improve the quality of services, fault-tolerance and security.
- Self-describing: Self-describing service interfaces can depict the contained information and functionality as reusable, and context-independent way. The underlying implementation of a service can be changed simultaneously without reconfigurations when the service contract is updated. Every service may be validated against each potential cloud in runtime before it is deployed. Self-describing services are advantageous, because they can notify the client application exactly how they should be called and what type of data they will return.
- Low cost composition of distributed applications as well as it should provide infrastructure for multi-party interactions and collaboration.

Overall, the service-centric model for Cloud Computing needs to include processes related to provisioning and deployment, service decommissioning, and service planning.

2.1.3 Interoperability

Interoperability focuses on the creation of an agreed-upon framework/ontology or open data format or open protocols/APIs that enables easy migration and integration of applications and data between different cloud service providers and also facilities for the secure information exchange across platforms. It is essential requirements for both service providers and enterprises. For enterprises, it is important to provide interoperability between enterprise clouds and cloud service providers. It is about more to its utility as a viable enterprise platform and industry-wise standard.

The issues of interoperability are either to allow applications to be ported between clouds, or to use multiple cloud infrastructures before critical business applications are delivered from the cloud. In addition, some issues are relevant in the context of enterprises such as mechanism of incrementally migration of enterprise applications to cloud platforms, procedure to extend enterprises policies and governance to cloud deployments, and design of public clouds to meet enterprise systems scope etc. Furthermore, interoperability might focus how to integrate entire cloud and grid systems into each other across service providers and enterprises as well as to investigate the relevancy of clouds of grids and grids of clouds.

There are many organizations working in the field of enterprise cloud interoperability. Cloud Computing Interoperability Forum (CCIF) [67] is one of them. In this forums, it is discussed whether a mechanism like network weather map is required to monitor the cloud and also what level of user control is needed to allow for interoperability.

2.1.4 Quality of Service (QoS)

In general, QoS provides the guarantee of performance and availability as well as other aspects of service quality such as security, reliability and dependability etc. QoS requirements are associated with service providers and end-users. SLAs play a facilitator key role to make agree upon QoS between service providers and end-users.

Transparent management systems to monitor resources, storage, network, virtual machine, service migration, and fault-tolerance are subjected by QoS. The state of art of QoS concept is exactly similar to the Grid Computing paradigm with some additional issues, such as virtualization of IT and network resources, virtual machine image migration. In the context of cloud service providers, QoS should emphasis on the performance of virtualization and monitoring tools.

The question is what the performance requirements are of applications and services that user plan to utilize from the cloud. In the case of high performance SLAs, service provider may still not be able to satisfy the performance levels at all the time due to inherent network latency of Internet. Since users expectations on QoS will always remain high, it is important to set the tolerance level of enterprise business processes.

2.1.5 Fault-tolerance

Fault-tolerance enables the systems to continue operating in the event of the failure of some of its components. In general, fault-tolerance requires fault isolation to the falling components, and availability of reversion mode etc. Fault-tolerance systems are characterized in terms of outages.

Cloud Computing outages are associated with the platforms. Some of the outages reported in the past were quite lengthy, see Table 2. For example, Microsoft Azure had an outage that lasted 22 hours in March 13th and 14th, 2008 which was deadly. If critical data/corporate business applications are hosted in the cloud and face such problems, perhaps it might be loss of billions of dollars. Thus, cloud reliance can cause significant problems if the control of downtime and outages are removed from cloud service provider's control. Table 2 also shows failover records for some of the cloud service providers. Every year, thousands of websites struggle with unexpected downtime, and hundreds of networks break or have other issues. Indeed, the major problem for cloud provider is how to minimize such kind of outage/failover to provide reliable services.

This table also illustrates the service availability and reliability of service providers. If the duration attribute is more, this means less reliability and availability of that particular service in the specified date. For instance, from table it seems that FlexiScale and Microsoft Azure are less reliable and having less availability on that particular date mention above for that service.

Cloud providers need to detect failure in cloud systems/applications with proper tools and mechanism such as application-specific Self-Healing and Self-Diagnosis mechanism. Furthermore, inductive systems like classification or clustering would be helpful not only for detection of failure but also to determine its possible cause.

2.1.6 Data Management, Storage, and Processing

The shift of computer processing, local storage like RAID storage and software delivery away from the desktop and local servers, across the Internet, and into Cloud Computing facilities, results in limitations as well as new opportunities regarding data management. The data will be replicated across large geographic distances where its availability and durability is paramount for cloud service providers. If the data is stored at un-trusted hosts that can create enormous risks for data privacy. Another characteristic of clouds is that the computing power is elastic in order to face the changing conditions. For instance, additional computational resources can be allocated on the fly to handle the increased demand.

Table 2 Outages in different cloud services

	Service and outage	Duration	Date
1	Microsoft Azure: malfunction in Windows Azure [46]	22 h	March 13–14, 2008
2	Gmail and Google Apps Engine [15]	2.5 h	Feb 24, 2009
3	Google search outage: programming error [51]	40 min	Jan 31, 2009
4	Gmail: site unavailable due to outage in contacts system [40]	1.5 h	Aug 11, 2008
5	Google AppEngine partial outage: programming error [59]	5 h	June 17, 2008
6	Google AppEngine experienced an outage that ranged from partial to complete [6]	5 h 50 min	July 2, 2009
7	S3 outage: authentication service overload leading to unavailability [66]	2 h	Feb 15, 2008
8	S3 outage: Single bit error leading to gossip protocol blowup [3]	6–8 h	July 20, 2008
9	FlexiScale: core network failure [71]	18 h	Oct 31, 2008

The focus of Cloud Computing is high performance computation, infinite capacity on-demand and no up-front cost. However, these characteristics may be easier to interpret in the context of computation than of storage. For the latter, the Cloud Computing providers must ensure that the storage infrastructure is capable of providing rich query languages which is based on simple data structures to allow for scale-up and scale-down on-demand. In addition, the providers need to offer performance guarantees with the potential to allow the programmer some form of control over the storage procedures. These goals are challenging, and currently no commercial solution exists incorporating all these features.

On one side of the spectrum of challenges related to storage is the Elastic Block Store (EBS) offered by Amazon that looks like a physical storage and where the programmer is responsible for managing the entire software stack. Although this approach gives the programmer full control over the processes, it is very difficult to provide scale-up on-demand and also to allow for the design of automatic replication facilities by the cloud provider, because the data storage model is application dependent. On the other side of the spectrum is Google's AppEngine, whose storage model is supported by the Bigtable [12], which provides excellent scalability, but it is tied to a very specific application model, i.e., that of Web services. Thus, it is not straightforward for this approach to support general purpose computing so as to allow various business applications to run on top of it. Finally, Microsoft's proposal based on Azure and a restricted view of structure query language (SQL) is an intermediate approach striving to gain from both worlds.

Even though these approaches will survive in the years to come, especially because they are supported by the today's software "giants" such as Amazon, Google, Microsoft, there are two parameters that these approaches currently do not provide. First, they do not provide the advances in storage technologies related to solid state disks (SSDs), i.e., flash-based disks. Second, they do not address the need by the scalable storage to support both MapReduce [18] operations over the raw data, and also declarative data management, as it is offered by traditional database management systems. The noticeable point here is SSD consumes less power in idle case, while hard disk drives (HDDs) need significant energy consumption.

In terms of storage technologies, there should be a shift from HDDs to solid-state drives [36, 47] or, since the complete replacement of hard disks is prohibitively expensive, the design of hybrid hard disks–hard disks augmented with flash memories [49], as the latter provide reliable and high performance data storage. The biggest barriers to adopting SSDs in the data centers have been price, capacity and, to some extent, the lack of sophisticated query processing techniques for SSDs. However, this is about to change, because the IOPS (input/output operations per second) benefits of SSDs are becoming important [48], for their capacity increases at fast pace, and new algorithms and data structures tailored to solid state disks [70].

The integration of magnetic and solid state disks in data centers is expected to boost the performance of applications running at cloud providers, but it will depend on the ability of the underlying infrastructure to recognize which applications are read-bound (e.g. data mining, processing of data coming from e-Science experiments), and which are write-bound (e.g. on-line transactional systems such as e-commerce applications).

Currently, the programming model of data centers supported by the current industry giants is a Lisp-based form, termed MapReduce. Despite its claimed generality, effort should be spent to convert traditional algorithms to this programming paradigm, since it is too low level and rigid, and leads to a great deal of custom user code that is hard to maintain and reuse.

Although MapReduce is a perfect fit for many tasks related to Web information retrieval (e.g., PageRank computations, inverted index construction), higher level abstractions are necessary to allow it to be used in enterprise-related tasks that require ad-hoc analysis of extremely large data sets. Towards this direction, new languages and systems must be developed to realize hybrid designs among DBMSs and Map-Reduce-like systems. The Pig Latin [57] and Clustera [19] comprise characteristic examples of a language and a system that address this goal.

2.1.7 Virtualization Management

Virtualization [35] refers to the abstraction of logical resources away from their underlying physical resources in order to improve agility, flexibility, reduce costs and thus enhance business value. Handling a number of virtualization machines on the top of operating systems and evaluating, testing servers and deployment to the targets are some of the important concerns of virtualization. The basic elements of the hypervisor include CPU, memory management, and I/O which provide the greatest performance, reliability and compatibility. Virtualization in the cloud includes server virtualization, client/desktop/application virtualization, storage virtualization (Storage Area Network), network virtualization, and service/application infrastructure virtualization, i.e.,

- Server virtualization: A common interpretation of server virtualization is the mapping of single physical resources to multiple logical representations or partitions.
- Client/desktop virtualization: Thin client technique is one of the cheapest models of achieving cloud virtualization, with security being the major concern.
- Storage virtualization: It provides the abstraction between logical storage and physical storage. Capacity, performance, durability, and availability are some of the related considerations.
- Network virtualization: It provides an environment to run multiple networks which can be customized for specific purpose at the same time over a shared substrate. It is the combination of hardware resources and software resources.
- Service/application virtualization: It virtualizes the application and provides an access through a centralized web server. Application licensing and user provisioning is easy to manage as well as it reduces application deployment costs, and enables delivery of applications as services.
- Infrastructure virtualization: It deals with the separation of application and infrastructure logic which enables the application developers to focus on building application instead of building infrastructure code. Infrastructure can be virtualized on the top of physical infrastructure. Virtual resource pool with virtual net resources facilitates to connect virtual infrastructure and resource virtualization.
- Resource virtualization: The idea behind resource virtualization is to customize resource provisioning in cloud and data center environment to meet workload requirements and to provide the ability to control the virtual resources execution. Several resources can be virtualized on the top of physical resources. Developing the strategies for service-specific policies and resource-specific policies are always focal concern in resource virtualization.

Virtualization is therefore well suited to a dynamic cloud infrastructure, because it provides important advantages in sharing of cloud facilities, managing of complex systems as well as isolation of data/application. Additionally, the significant idea is to ensure whether the data centers are locked or not into a particular operating system environment through their choice of a virtualization.

2.1.8 Scalability

Scalability deals with the ability of the software system to manage increasing complexity when given additional resources. Scalability with large data set operations is a requirement for Cloud Computing. Horizontal scalability is what clouds provide through load balancing and application delivery solutions. Distributed hash table (DHT), column-orientation, and horizontal partitioning are examples of horizontal scalability. Vertical scalability is related to resources used, much like the old mainframe model. If an application doesn't scale well vertically, it's going to increase the costs to run in the cloud. Applications that fail to vertically scale may end up costing more when deployed in the cloud because of the additional demand on compute resources required as demand increases.

Force.com was designed to run a simple business application and it is based on a database centric architecture. It seems to be limited in scalability as it partitions its database per application. This means that if an application needs to scale more than what a single database can provide, there is a challenge.

Even though DHT, column-oriented store or document-centric approaches were designed to address the issues of scaling or write heavy applications, they cannot support for complex joins, foreign keys as well as reporting. They may be considered as a part of an overall architecture and they can play a vital role to reduce writing heavy bottlenecks. However, they should not be considered a replacement for a relational database.

Considerations of Cloud-based scaling are basically dependent on the nature of the applications and the expected volume of usage. Werner [74] suggested that architect must be carefully inspect along which axis we expect the system to grow, where redundancy is required, and how one should handle heterogeneity in the system, and make sure that architects are aware of which tools they can use under which condition, and what the common pitfalls are. Architecting a scalable enterprise application is not a trivial task.

2.1.9 Load Balancing

Load Balancing is an integral part of Cloud Computing and elastic scalability which can be provided by software or hardware or virtualware. It is the mechanism of self-regulating the workloads properly within the Cloud's entities (one or more servers, hard drives, network, and IT resources). The cloud infrastructures and datacenters need huge computing hardware, network and IT resources which are always subjected to the failover when the demand exceeds. Load Balancing is often used to implement failover. The failover occurs due to the limited resources allocation, hardware failure, power and network interruption etc. The service components are monitored continually and when one becomes non-responsive, the load balancer is informed and no longer sends traffic to it. This is an inherited feature from Grid-based computing that has been transferred to Cloud-based platforms. A load balancer is a key requirement to build dynamic cloud architecture. It provides the ways by which application instances can be provisioned and de-provisioned. There are some architectural considerations while designing the load balancers:

- design to provide scalability that could be at the CPU level, at the machine level, at the network level, or even at the application level and data center level
- capability to manipulate the client requests and forward it to the selected target resources by using load balancing policies
- scalability of the request handling capacity automatically
- fault tolerance for applications
- handling of more complex and higher traffic needs such as Apache traffic server. Apache traffic server[12] provides high performance web proxy cache to improve network efficiency and performance and improves content delivery for enterprises backbone providers and large intranets by maximizing existing and available bandwidth.

2.2 Enterprise Requirements

Gartner predicated that, by 2012, 20% of enterprise market e-mail will be hosted on the cloud [32]. Therefore, potential cloud adopters should do their assessment before being "absorbed" by this trend, thus reading external sources of information and talking to analysts, technology providers and experts in order to understand the real challenges and benefits of this new technology. According to Robbins [17], an enterprise should always make sure that they know what services they are paying for, and should pay careful attention to the issues like service levels, privacy matters, compliances, data ownership, and data mobility.

This section describes cloud deployment requirements for enterprises, including a long list of requirements ranging from security, cloudonomics, data governance over to business process management requirements and transferable skills requirements.

[12] Apache traffic server, http://trafficserver.apache.org/.

2.2.1 Cloud Deployment for Enterprises

The cloud services are ubiquitous as a single point of access with four types of deployment models. These are public, private, community and hybrid clouds. Each of types has its trade-offs. Private cloud involves enterprise firms to deploy their key enabling technologies such as virtualization and multi-tenant applications to create their own private cloud data centers. Individual businesses can pay for using standardized services in line with agreed charge-back mechanisms. For many enterprises, this approach is more attractive than move to the public cloud.

- Public Cloud: The concept of sharing the services and infrastructure provided by off-site third-party service provider (own and manage physical infrastructure) in a multi-tenant environment, usually called as public cloud. It describes Cloud Computing in the traditional mainstream sense, whereby resources are dynamically provisioned on a fine-grained, self-service basis over the Internet, via web applications/web services. Generally, enterprises do not want to move their mission-critical and core-business applications in the public cloud due to security and control.
- Private Cloud: The idea behind private cloud is to share services and infrastructure provided by an organization or its specified service provider in a single-tenant environment. It is not as cost effective as the public cloud, but it is cheaper than buying and maintaining a data center. Similarly, it gives companies a high level of control over the use of cloud resources. Public clouds can access the data on the private cloud with data services. The services in the public cloud communicate with a data service layer. The latter figures out how to retrieve the physical data from the appropriate location over a secure protocol. The best way to achieve enterprise cloud is to introduce the private cloud which can be built from both internal and external computing resources. Private cloud provides trusted self-services capabilities.
- Community Cloud: According to NIST [52], the community cloud is shared by several organizations and is supported by a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations).
- Hybrid Cloud: It consists of multiple internal or external providers. It can provide cost savings, scalable on-demand infrastructure and security. The combination of hybrid cloud with traditional infrastructure may be a viable solution for most of the companies. However, hybrid clouds introduce the complexity of determining how to distribute applications across both a public and private cloud. If the data amount is small, or the application is stateless, a hybrid cloud can be much more successful than if a large amount of data must be transferred into a public cloud for a small amount of processing.

In summary, enterprises need to build a strategy that leverages all four options mentioned above. The virtual private networks help to isolate the computing resources, thereby extending the existing IT resources of enterprises; this is called *Virtual Private Clouds* that help to optimize the business service deployment. Table 3 describes the general summary of requirements of cloud deployment models.

2.2.2 Security

Usually security is the focal concern in terms of data, infrastructure and virtualization. In a survey conducted by IDC[13] (2009), almost 75% of IT executives reported that security was their primary concern, followed by performance and reliability, i.e. how enterprise data is safeguarded in a shared environment. It is today widely accepted that cloud introduces new security risks [8, 34]. Corporate information is not only a competitive asset, but it often contains information of customers, consumers and employees that, in the wrong hands, could create a civil liability and possibly criminal charges.

In Cloud Computing, a data center holds the information that would more traditionally have been stored on the end-user's computer. This

[13] IDC, http://www.idc.com/.

Table 3 Summary of cloud deployment models and the corresponding requirements

Parameters / Deployment models	Cost	Migration	Client base	Security	Control over the use of cloud resources	Legal issues
Private cloud	Expensive	Standard API is needed between public, private and hybrid cloud	Large enterprises and corporations	High	High	
Public cloud	Less expensive than private cloud	Standard API to make data movement seamless	Large as well as SMEs	– Low, more chances of malicious activities such DDoS attacks – Trusted virtual data center is required	Low	National boundary for data storage
Community cloud	Relatively cheaper than other three cloud models	Standard API is needed to migrate data within specific community	Small SMEs	Low	Low	
Hybrid cloud	Cost savings	Standard API to make data movement seamless	Multiple internal and/or external providers	Application compatibility issues	High	National boundary for data storage

raises concerns regarding user privacy protection, since the users do not "own" their data. Also, the move to centralized services may affect the privacy and security of users' interactions. Security threats may happen in resource provisioning and during distributed execution of user applications. Also, new threats are likely to emerge. For instance, hackers can use the virtualized infrastructure as a launching pad for new attacks. Cloud services should preserve the integrity of data and the privacy of users. In this context, new data protection mechanisms to secure data privacy, resource security, and content copyrights should be investigated along with the traditional ones.

To this end, the authorization management and control in Web 2.0 are considered important and critical. Authorization can be course-grained within an enterprise. Private clouds will need granular authorization (such as role-based controls) that can be persistent throughout the data's lifecycle. Authentication may be multi-factor authentication with one-time password technology, federation within and across enterprises, or risk-based authentication that measures behavior history, current context etc. Federated-type single sign-on models based on SAML (Security Assertion Markup Language) protocol [56] and OpenID[14] are positioned to manage the authentication issues between users and cloud application providers. The user identity is also important in the growth of sensitive data and confidential relationships online [76].

2.2.3 Cloudonomics

The Economics of Cloud Computing is called Cloudonomics [75]. The enterprises will have a choice of cloud vendors based on their pricing, and billing systems for *pay-per-use* model. The problem with Cloud Computing is the lack of cost based transparency. It is actually very difficult to quantify the cost benefits of using traditional infrastructure versus using a remote service providers (Amazon EC2, GoGrid etc). For example, cost of buying a cluster assuming that

[14]OpenID Foundation, http://openid.net/.

runs 24*7 for 5 years versus cost of same number of hours on cloud infrastructure.

The emergence and popularity of Cloud Computing does not necessarily mean that it is the best solution for any enterprises from the economic perspective. It might be the best alternative at some point in enterprise's lifetime and under specific market conditions. To understand the economy of Cloud Computing, we recall the basic premises of clouds:

- they provide true on-demand services, by multiplexing demand into a common pool of dynamically allocated resources,
- (large) cloud providers operate at a scale much greater than even the largest private enterprises,
- whereas enterprises' private data centers try to reduce cost via consolidation and concentration, the clouds exploit geographic dispersion.

Based on these observations, Joe Weinman [75] documented the superiority of the clouds using the "10 Laws of Cloudonomics". Herewith, we will summarize the major points of Cloudonomics, focusing our description around the *pay-as-you-go* (i.e., usage-based) pricing model employed by Cloud Computing.

Contrary to leased, owned or financed services, even in the cases when the cost per unit time is higher for cloud providers, this cost is amortized over the usage (Cloudonomics Law #1)—the cloud user pays for just the resources that a particular computation requires; computations that require additional resources, simply request them from the cloud up to the cloud's overall capacity. Therefore, Cloud Computing does not require upfront investments, and lets user's access capacity exactly when they need it (Cloudonomics Law #2).

Within the cloud, the laws of probability give the cloud provider great leverage through statistical multiplexing of varying workloads (Cloudonomics Law #3), thus enabling him to support a wide range of peak demands which due to the multiplexing appear smother (Cloudonomics Law #4).

The magnitude of the infrastructure of a (large) cloud provider provides tremendous economic advantages, since it can acquire hardware/software at lower prices (Cloudonomics Law #5), and implement parallel processing at a much greater scale (Cloudonomics Law #6 and #7) even for real-time tasks.

Finally, the ability of the cloud provider to implement its data centers at geographically dispersed regions provides unique opportunities for latency reduction (Cloudonomics Law #8), similarly to the Content Distribution Networks that move data close to the edge of the Internet, for survivability and availability (Cloudonomics Law #9), and allow for enterprise "mobility" (Cloudonomics Law #10), since an enterprise's private data center can rarely be build on locations away from the enterprise's geographic basis, even though such a decision would be more profitable, e.g., for its users.

2.2.4 Data Governance

Geographical and political issues are the key requirements for enterprise cloud. When data begins to move out of the organizations, it is vulnerable to disclosure, or loss. The act of moving sensitive data outside the organizational boundary may violate national regulations for privacy. In Germany[15] passing data across national territory can be a federal offense [21, 38]. Also in some of the European countries require services and customer data be retained within a country's borders. Governance in the cloud "who and how" is a big challenge for enterprise cloud. There are some questions that need to be solved before mission critical data and functionality can be moved outside a controllable environment. How SLA controlled and enforced, what will happen if a provider goes insolvent? Some of the data might reside in China or India while it is processed in Europe or North America. What will happen if a country changes its laws unexpectedly? The missing control could easily bring a business down when relying on cloud services that do not deliver anymore. This is a serious issue if *mission critical*

[15] German data protection law called "Informationelle Selbstbestimmung".

data is inaccessible for a longer period. Therefore, many government agencies do not want to move their mission critical data like defense data, government strategy and certain important policy data.

In a true market-based economy, a client is able to select the provider of the goods s/he needs based on a combination of factors related to cost, quality etc. This means a user should be able to move data/programs to another cloud provider, if the second provider is more profitable. In other words, the cloud users must be protected against "data lock-in".

One of the central questions here is, which technology and development platform are being used by your organization? Vendor lock in is one of the important considerations for an enterprise. There are currently no standardized ways to plug into a cloud, and this makes it hard to switch to a new vendor. For example, users can try to move their MySpace profile and its content to Facebook without manually retyping the profiles. Cloud service providers can provide a service for their customer to move data in and out of their systems without too much hassle by adopting open standards. This would be a good chance for providers to compete on openness from the outset.

Currently, due to the lack of interoperability among cloud platforms and the lack of standardization efforts, cloud providers cannot guarantee that a cloud user can move its data/programs to another cloud provider on demand. Cloud Computing would become much more appealing if protection against data lock-in is fully implemented, since it would liberate the users from possible monopolies, make them less vulnerable to price increases, and increase competition among cloud providers for offering better quality vs. price ratios. Finally, it would also guarantee longevity of the cloud users, since they would not be afraid of cloud providers going out of business.

For the time being, this issue is still unresolved, and the current situation implies that once the cloud users upload their data to the data center, the data is locked-in. This issue is of paramount importance and, together with the security topics, comprise a key factor that will determine the extent to which Cloud Computing will be widely adopted by the enterprises.

2.2.5 Data Migration

The issue of distributing information to web users in an efficient and cost-effective manner is a challenging problem, especially, under the increasing requirements emerging from a variety of modern applications, e.g., voice-over-IP, and streaming media. Starting from the early approaches based on proxy caches, currently, Content Distribution Networks (CDNs), e.g., Akamai, have met these challenges by providing a scalable and cost-effective mechanism for accelerating the delivery of web content, based on more or less sophisticated data migration (outsourcing) policies for the surrogate servers of a CDN [42].

In the context of Cloud Computing though, such policies are ineffective, since they must collectively ensure the following goals:

1. No data loss: The system must ensure with a high probability that data will not be lost permanently.
2. High availability: Data must be available to users/owners when they want the data with a reasonably high probability, though some occasional temporary outages are acceptable.
3. High performance: The system should not perform significantly worse than the current usual alternatives notably NFS.
4. Scalability: The system must scale to large numbers of clients, large numbers of storage 'nodes', large aggregate storage spaces, etc.
5. Cost efficiency: Since there are existing solutions to buy large, reliable storage, the system must be inexpensive in hardware, software and maintenance.
6. Security: The system must be able to match the confidentiality, data integrity and authorization standards expected by users. This is particularly challenging given that data will be stored on the remote machines of cloud providers.

Apparently, the 4th and 5th items are "by-definition" offered by the cloud infrastructure itself, since cloud users are not supposed to worry about the hardware and scalability.

The first three items are extremely significant since current data centers suffer from failures due to overheating, power failures, rack failures, network failures, hard drive failures, network rewiring and maintenance, and even due to natural disasters (e.g., a tornado may destroy a whole data center), or due to malicious human behavior (DDoS, terrorism). These situations require fail-over processing and migration for seamless services.

To collectively address the first three goals, data replication in the cloud seems the most convenient approach. Replication presents some fundamental differences when viewed in the cloud context: it must be fully distributed, must be dynamically adapted the number of replicas to the query load and the capabilities of the underlying servers, must be provided different data consistency levels (strong, weak, differentiated), and must be cost-effective, ensuring the highest possible availability for the consumers of the data and also for the owner of the data.

Although several caching and replication algorithms are widely available to address each goals individually (or subsets of these goals), novel techniques are required towards this direction. The relevant cloud literature lacks sophisticated caching/replication solutions and only a couple of recent proposals seem to have understood the importance of the problem [7, 16]. Process migration is a valuable resource management tool that requires continuity of process to keep its process in local stack of data and operations. The major difficulty lies in providing an efficient method for naming resources that is entirely independent of their location.

2.2.6 Business Process Management (BPM)

Business process management systems provide a business structure, security and consistent rules across business processes, users, organization and territory. This classical concept is enhanced in the context of Cloud-based BPM, as cloud delivers a Business Operating Platform for enterprises such as combining SaaS and BPM applications (e.g., customer relationship management (CRM), workforce performance management (WPM), enterprise resource planning (ERP), e-commerce portals etc.) which helps for the flexibility, deployability and affordability for complex enterprise applications. When the enterprises adopt Cloud-based services or business processes, the return of investment (ROI) of overall business measurement is important. There are still many debates are on-going about Cloud Computing ROI and the methodology of its measurement and associated parameters. Cloud Business Artifacts (CBA) project of The Open Group Cloud Computing Work Group [69] identified the following issues; how to measure

- Pricing and costing of cloud services
- Funding approaches to cloud services
- Return of Investment (ROI)
- Capacity and utilization (called as Key Performance Indicators (KPIs))
- Total cost of ownership
- Risk management
- Decision and choices evaluation processes for cloud services

CBA also describes approaches to measuring this ROI, absolutely and in comparison with traditional approaches to IT, by giving an overview of cloud KPIs and metrics. Reusability of business processes can help the enterprises to maximize their profits and some of the intelligent/innovative business processes such as *situational business processes* [24] would be fruitful to drive the business values.

2.2.7 Third Party Engagement

The involvement of third-party in enterprises can help for establishing a robust communication plan with provider landscape, continuity of cloud service engagements, legal implications (e.g., SLAs, potential intellectual property (copyrights, trademarks, and patents) infringement issues), cloud audit, and reporting capabilities etc.

A key requirement to move the on-premise IT systems to Cloud Computing is the existence of the business model. An enterprise should take the onus of encouraging the supply chain management (SCM), enterprise resource planning (ERP) and enterprise content management (ECM) package vendors to come-up with a *Cloud License* model which takes machine instances to consideration. The major problem with offerings like Amazon EC2 or Google AppEngine is, that enterprises might get a refund if their SLAs with

Amazon have been violated, but there is no way to have stronger contractual bounds with service provider. Therefore, enterprises might have to use smaller, more specialized semi-private cloud offerings, and those again might be more subjected to attacks.

2.2.8 Transferable Skills

Transferable skills deals with the technology dissemination, technical supports, discussion with consulting expert groups, or *offshore* outsourcing that help for the adaptation and stability of systems/applications. Cloud Computing comes with its own set of management tasks that need to be executed by the enterprises staff, for example staff needs to monitor current computing capacity and to increase or decrease it depending on usage, or developers might have to implement a new feature of the system as business processes change. Before choosing a cloud provider system, the enterprise should have a look at the skill set of their existing workforce to identify those skills that are transferable to the new environment in order to make the transition as fluent as possible because there is a wide variation in maturity of enterprise cloud software and services.

2.3 User Requirements

Users' requirements are the third key factor to the adoption of any cloud system within an enterprise. Cloud should be trustworthy enough to migrate critical user data. Users need assurance that their sensitive data and information are protected from compromise and loss that their data is available when required from anywhere of the world. For users, the trust issues are a major concern to the adoption of the cloud services. Trust-based [34] cloud is therefore an essential part to the success of enterprise cloud. Stability and security can play a vital role to increase the trust between user and service providers. Cloud-based applications should be architected to be able to support personalization, localization and internationalization to make user-friendly environment.

This section describes user consumption-based billing and metering requirements, user centric privacy requirements, service level agreements, adaptability and learning requirements, and user experience requirements.

2.3.1 User Consumption-based Billing and Metering

The individual end-user consumption-based billing and metering in cloud systems is similar analogy with the consumption measurement and allocation of costs of water, gas and electricity consumption on a consumption unit basis. Cost management is important for planning and controlling decision. It helps to check the utilized resources versus cost. Cost breakdown analysis, tracing the utilized activity and adaptive cost management are important considerations as well. Users always want transparency of consumption and billings. What is the frequency of usage of services? How frequent is the service usage? Very frequent usage in fact makes less economic sense to go for cloud a based *pay-per-use* model. An architect needs to pay special attention to provide this visibility while architecting cloud systems. Activity-Based Costing (ABC) [50] is a profiler that the user can use it to understand how much their implementation will cost in terms of Cloud Computing charges. Such kind of application may be helpful for user to test cost transparency.

2.3.2 User-Centric Privacy

The main consideration regarding Cloud Computing for end users is related to the storage of personal/enterprise sensitive data. Cloud Computing brings with it the fact that most of the users' creations, data that the user would regard as his/her personal intellectual property, will be stored at mega-data centers located around the world. In this environment, privacy becomes a major issue [11]. Therefore, the cloud providers are making all to efforts to win is the trust of their users; in general, a corporation would not be willing to store any sensitive data on the cloud. Even though this seems to be an unsolvable problem, there are indications that it can be overcome. First, many users have already been exposed to some form of Cloud Computing, mainly in a form of email service (e.g., Gmail) or a Web 2.0 application (Facebook, Myspace etc), without much concern about privacy. Therefore, users may be used to a level of privacy as they know today. Second, there are currently technologies that can assure data integrity, confidentiality, and security for the clouds to be trusted. These technologies include:

- data compressing and encrypting at the storage level, thus the cloud provider can't really use the sensitive data,
- virtual LANs, that can offer secure remote communications,
- network middleboxes (e.g., firewalls, packet filters), to further failsafe secure communications.

These technologies are quite mature and there is no technical difficulty *per se* in providing these tools for clouds. They will certainly need adjustments and improvements in order to be applied at the huge scale of clouds, but this is something that can be achieved. For example, the Sector [37] provides authentication, authorization, and access control, and, as measured by the TeraSort benchmark, is faster than Hadoop, which currently does not have user-level authentication or access control.

2.3.3 Service Level Agreements (SLAs)

The mutual contract between providers and users is usually called SLAs, i.e., the ability to deliver the services according to pre-defines agreements. Currently, many cloud providers offer SLAs, but these are rather weak on user compensations on outages. The lists of important architectural issues are:

- How and who will measure the delivery of services?
- How to develop an agreed method of monitoring performance?
- What will happen if the provider unable or fails to deliver the services as contract?
- What will be the mechanism to change SLA over time?
- What will be the compensation mechanism if service provider violates any elements of the SLA?

Users always want stable and reliable system services. Cloud architecture is considered to be highly available, up and running 24 hours by 7 days. Many cloud service providers have made huge investments to make their system as reliable as they can give the scale of their system. Most cloud vendors today do not provide high availability assurances; this is particularly an issue with enterprise Mashups [43, 62] that need a set of web services hosted in various Cloud Computing environments and many may stop working at any time. If a service goes down (refer back to Table 2) for whatever reasons, what can users do? How can users access their documents stored in the cloud? In such case, the provider may offer to pay a fine to the consumer as compensation.

2.3.4 Adaptability and Learning

Cloud Infrastructure must handle more resources, data, services and users; all of this makes Cloud-based enterprise application/systems more complex to control, to keep coherence between services and resources. The biggest challenge for every user is to get acquainted with an application presented by the enterprises when trying to deal with clouds. To this end, users must be empowered to execute effective controls over their personal information [11]. Cloud Computing is being adopted by business leaders and most of the processes in Cloud Computing are for sensitive business processes such as online purchases of products or acquisition of services. If the users do not fully aware about its usage, they would not be benefited with those services or could even be exposed to different types of security attacks. Some of the guidelines are listed below to make learning and users adaptation in the Cloud-based application/systems:

- identify the suitable approaches based on observation, data recording and users demand to make learning and adaptation environment to the users so that they can easily use the systems
- design the applications to meet the goal of reconfiguration, personalization and customization

- intelligent, and interactive demonstration would be helpful to aware the users about the usage of application. Artificial intelligence and semantic techniques are also helpful here.

2.3.5 User Experience (UX)

The notion of UX is to provide the insight into the needs and behaviors of end-user that can help to maximize the usability, desirability, and productivity of the applications. UX-driven design and deployment may be the next step in the evolution of Cloud Computing. The screens or user interfaces must be simple, uncluttered, and designed according to a workflow or process you expect your users to follow. For example, in the case of SaaS application, AJAX based UIs, and Smart Clients have an interesting impact on users experience sharing and practice that can help to elaborate the scope of software products and services as well as its business value. On the other hand, cloud application and devices evolution such as mobile brings new opportunities to user involvement and interaction. Mobile devices have drag and drop features of Gadgets that provides information and fun-like news, pictures, games etc. Furthermore, Human-Computer Interaction, ergonomics, and usability engineering are the some of the important key requirements to design UX-based cloud applications. Some of the important considerations are listed below, i.e.,

- Identification of most potential services from users and business point of view
- analysis of service possibilities, context for the problem, and describes the rationale behind the solution by implementing UX design patterns
- ensuring that UX involvement from the beginning of service/product life-cycle
- incorporation of end-user involvement in design
- conducting reality demonstration of the services/applications

In addition, Cloud-based application/systems should be easy to use, capable of providing faster and reliable services, easily scalable, and customizable to meet the goal of localization and standardization. They should also ensure the impact of rich Internet applications (RIAs) such as EyeOS[16]. For instance, EyeOS 2.0 is the perfect development framework for quick and easy creation of RIAs that provides effectiveness, usefulness, versatility/appropriateness and comfortableness environment to the users.

3 Discussion and a Comparative Analysis of the Common Requirements

The Table 4 gives a comparative analysis of the requirements discussed in the prior sections. It also illustrates the relationships between architectural requirements layers and components described above.

There are various architectural styles of software architecture such as client/server, pipe-and-filter, distributed object, event-based integration, virtual machine etc. Table 5 summarizes the mapping of architectural requirements to the general architectural components.

Furthermore, some of the important issues are given below that should be considered while mapping the requirements to architectural components.

- What kind of architectural components are frequently made in building dynamic large cloud systems?
- How do architectural components relate to provider's and enterprise's system requirements?
- How can we able to extract the generic and reusable model to classify relations between requirements and architectural components/patterns?
- How do we abstract key architectural assessment made in existing cloud application/systems?

In addition, we have figured out three general architectural components to couple those architectural styles such as *workflow orchestration*, *single flow* and *types of requirements*. Workflow orchestration provides the pipe-and-filter architectural styles [5] to manage business, service

[16]EyeOS, http://www.eyeos.org.

Table 4 Comparative analysis of the common requirements

Parameters / Requirements	Security	Interoperability	Data management, storage and processing	SLA	Metering and billing	Virtualization management	User-centric privacy	Data migration
Provider requirements	Can build up the trust among tenants	Open data format, Open API is more helpful for interoperable cloud services	Should capable of providing rich query language to allow scale-up scale-down -flexible and elastic storage services with durability is needed	If they offer 99.99% of SLA service, this means their service is more reliable	Can built up the trust with enterprises and end-users by offering efficient metering and billing services	Can maximize profit by virtualizing application, network, database, infrastructure	Can build up the trust easily	Regional data center is needed if there is a country boundaries for data migration
Enterprise requirements	Organizational confidential data and mission critical data can move on cloud if there is a secure services	Allow to be ported between clouds before critical business application are delivered from clouds	Elastic storage, no need to worry about storage	Can claim the charge for service outage	Can build up the trust easily between tenants	Can maximized profits	Can build up the trust between tenants	Political and legal issues associated with it
Users requirements	Trusty environment will create to use cloud services easily	User can take services from any provider if the data portability is easy.	Can be benefited with on-demand storage service	Legal agreement is important if the service is down users can claim refund.	Transparent metering and billing affect the user's intension to use the services offered by providers.	Virtualization makes cloud services cheaper, so users can be easily benefited from it	User's point of view privacy is the primary concern to trust service providers.	Can move their data in their regional datacenter

Table 5 Partial mapping of requirements to architectural components

	NFR	NFR/FR	FR
Single flow	R5, R6, R8, R11, R12, R13, R16, R17, R19, R20, R21	R7, R18	R1, R4, R10, R14
Workflow orchestration	R9, R15, R22	R2	R3

Requirements	
R1	Service delivery model (SaaS, PaaS, IaaS)
R2	Service-centric
R3	Data management and storage
R4	Virtualization management
R5	Cloud deployment
R6	Fault tolerance
R7	Security
R8	Quality of service
R9	Cloudonomics
R10	Load balancing
R11	Interoperability
R12	Scalability
R13	Data governance
R14	Data migration
R15	Business process management
R16	Third party engagement
R17	Transferable skills
R18	User consumption-based billing and metering
R19	User-centric privacy
R20	Service level agreements
R21	Adaptability and learning
R22	User experience

and scientific data managements. The *functional requirements (FRs)* deal with the set of operation/functions that the system offers while *non-functional requirements (NFRs)* refer to the manner in such a way those functions are executed. Some of the NFRs need to be captured and analyzed from the very early stages of the application/system development process but NRFs are always coupled with FRs. The preliminary idea of dividing functional and non-functional requirement is to figure out the whole scenario of Cloud-based applications. That could be helpful for architects to set up the criteria for the solution in terms of the operational, and service-level objectives in the phase of requirement engineering.

4 Conclusions

Clear understanding of requirements and their relationships with respect to architectural choices is a critical for any enterprise system transforming the corporate culture into a co-operative business culture by sharing/reusing business processes with help of the cloud services. The biggest challenge in Cloud Computing is the lack of a de facto standard or single architectural method, which can meet the requirements of an enterprise cloud approach. In this paper, we explored the architectural features of Cloud Computing and classified them according to the requirements of end-users, enterprises that use the cloud as a platform, and cloud providers themselves. We show that several architectural features will play a major role in the adoption of the Cloud Computing paradigm as a mainstream commodity in the enterprise world. This paper also provides basic guidelines to software architects and Cloud Computing application developers for creating future architectures. Although the abstract nature of cloud makes it challenging to analyze the coverage of all requirements and architectural components with respect to a generic enterprise cloud system, our set of general, and preliminary requirements can be used as the first step towards a more specific set of properties and requirements for the emerging enterprise cloud services.

Acknowledgements The authors would like to thank the anonymous reviewers for their valuable comments which helped us to improve the quality of our initial manuscript. This work has been partially supported by the EU FP7-ICT-248657 GEYSERS project.

References

1. Adabala, S., Chadha, V., Chawla, P., Figueiredo, R., Fortes, J., Krsul, I., Matsunaga, A., Tsugawa, M., Zhang, J., Zhao, M., Zhu, L., Zhu, X.: From virtualized resources to virtual computing Grids: the In-VIGO system. Future Gener. Comput. Syst. **21**(6), 896–909 (2005)
2. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: Above the clouds: a Berkeley view of cloud computing. Technical Report No. UCB/EECS-2009-28, Electrical Engineering and Computer Sciences, University of California at Berkeley (2009)
3. AWS Service Health Dashboard: Amazon S3 availability event. Available online at http://status.aws.amazon.com/s3-20080720.html (2009). Accessed on March 2009
4. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. In: Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles (2003)
5. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. Addison-Wesley, Reading, Massachusetts (1998)
6. Beckmann, C.: Google App Engine. http://tinyurl.com/l949tp (2008). Accessed on July 2009
7. Bonvin, N., Papaioannou, T.G., Aberer, K.: Dynamic cost-efficient replication in data clouds. In: Proceedings of the 1st Workshop on Automated Control for Datacenters and Clouds (2009)
8. Brodkin, J.: Gartner: seven cloud-computing security risks. Available online at http://tinyurl.com/3okysm (2008). Accessed on March 2009
9. Carr, N.: The Big Switch—Rewiring the World from Edisson to Google. W. W. Norton (2008)
10. Carr, N.: Rough Type. Available online at http://www.roughtype.com (2010). Accessed on January 2010
11. Cavoukian, A.: Privacy in the Clouds—A White Paper on Privacy and Digital Identity: Implications for the Internet. Information and Privacy Commission of Ontario (2008)
12. Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., Gruber, R.E.: Bigtable: a distributed storage system for structured data. ACM Trans Comput. Syst. **26**(2), 1–26 (2008)
13. Choudhary, V.: Software as a service: implications for investment in software development. In: Proceedings of the 40th Hawaii International Conference on System Sciences (2006)
14. Clements, P., Northrop, L.M.: Software architecture: an executive overview. Software Engineering Institute, Carnegie Mellon University, Technical Report CMU/SEI-96-TR-003 (1996)
15. Cruz, A.: Gmail site reliability manager, update in Gmail. Available online at http://tinyurl.com/b2vzka (2009). Accessed on March 2009
16. Dash, D., Kantere, V., Ailamaki, A.: An economic model for self-tuned cloud caching. In: Proceedings of the IEEE International Conference on Data Engineering (2009)
17. David, R.: Cloud computing explained. Available online at http://tinyurl.com/qexwau (2009). Retrieved on 1 Sept 2009
18. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
19. DeWitt, D.J., Robinson, E., Shankar, S., Paulson, E., Naughton, J., Krioukov, A., Royalty, J.: Clustera: an integrated computation and data management system. In: Proceedings of the Very Large Databases (2008)
20. Emeneker, W., Stanzione, D.: Dynamic virtual clustering. In: IEEE International Conference on Cluster Computing (2007)
21. Federal Data Protection Act, Bundesdatenschutzgesetz (BDSG), http://tinyurl.com/37sf3ly (1990). Accessed 10 December 2009
22. Fielding, R.T.: The representational state transfer (REST). Ph.D. Dissertation, Department of Information and Computer Science, University of California, Irvine. Available online at http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm (2000)
23. Figueiredo, R.J., Dinda, P.A., Fortes, A.B.: A case for Grid computing on virtual machines. In: Proceedings of the 23rd International Conference on Distributed Computing Systems (2003)
24. Fingar, P.: Extreme Competition: Cloud Oriented Business Architecture. Business Process Trends (2009)
25. Flouris, M.D., Lachaize, R., Bilas, A.: Violin: a framework for extensible block-level storage. In: Knowledge and Data Management in Grids, CoreGRID series, vol. 3, pp. 83–98. Springer Verlag (2007)
26. Foster, I., Kesselman, C.: Globus ametacomputing infrastructure toolkit. Int. J. Supercomput. Appl. **11**(2), 115–128 (1997)
27. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the Grid: enabling scalable virtual organizations. Int. J. High Perform. Comput. Appl. **15**(3), 200–222 (2001)
28. Foster, I., Freeman, T., Keahey, K., Scheftner, D., Sotomayor, B., Zhang, X.: Virtual clusters for Grid communities. In: 6th IEEE International Symposium on Cluster Computing and the Grid (2006)
29. Foster, I., Zhao, Y., Raicu, I., Lu, S.: Cloud computing cloud computing and Grid computing 360 degree

compared. In: Grid Computing Environments Workshop (2008)
30. Fox, A., Gribble, S.D., Chawathe, Y., Brewer, E.A., Gauthier, P.: Cluster-based scalable network services. In: Proceedings of the Sixteenth ACM Symposium on Operating Systems Principles (1997)
31. Garg, V.K.: Elements of Distributed Computing. Wiley-IEEE Press. ISBN 0471036005 (2002)
32. Gartner Research: Predicts 2009: Cloud Computing Beckons (2009)
33. Gathering Clouds of XaaS! http://www.ibm.com/developer (2008)
34. Gellman, R.: Privacy in the clouds: risks to privacy and confidentiality from cloud computing. In: World Privacy Forum (2009)
35. Golden, B.: Virtualization for Dummies. Wiley Publishing, Inc. (2008)
36. Graefe, G.: The five-minute rule twenty years later, and how flash memory changes the rules. In: Proceedings of the 3rd International Workshop on Data Management on New Hardware (2007)
37. Gu, Y., Grossman, R.L.: Sector and sphere towards simplified storage and processing of large scale distributed data. Philos. Trans. R. Soc. Lond. A. **367**(1897), 2429–2445 (2009)
38. Hoeren, T.: The new German data protection act and its compatibility with the European Data Protection Directive. J. Comput. Law, Security Review, Elsevier **25**(4), 318–324 (2009)
39. Irwin, D., Chase, J., Grit, L., Yumerefendi, A., Becker, D., Yocum, K.: Sharing networked resources with brokered leases. In: USENIX Technical Conference (2006)
40. Jackson, T.: We feel your pain, and we're sorry. Available online at http://tinyurl.com/58mqho (2008). Accessed on March 2009
41. Kapil Bakshi, K.: Cisco cloud computing-data center strategy, architecture, and solutions. Point of View White Paper for U.S. Public Sector (2009)
42. Katsaros, D., Pallis, G., Stamos, K., Vakali, A., Sidiropoulos, A., Manolopoulos, Y.: CDNs content outsourcing via generalized communities. IEEE Trans. Knowl. Data Eng. **21**(1), 137–151 (2009)
43. Kavis, M.: Enterprise mashups—the icing on your SOA. Available online at http://socialcomputingjournal.com (2008). Accessed on July 2009
44. Keahey, K., Foster, I., Freeman, T., Zhang, X.: Virtual workspaces: achieving quality of service and quality of life in the Grid. Sci. Program. J. **13**(4), 265–276 (2005)
45. Keahey, K., Foster, I., Freeman, T., Zhang, X., Galron, D.: Virtual workspaces in the Grid. In: 11th International Euro-Par Conference (2005)
46. Kolakowski, N.: Microsoft's cloud azure service suffers outage. Available online at http://tinyurl.com/yhqo5t3 (2009). Accessed on April 2009
47. Lee, S.W., Kim, W.: On flash-based DBMSs: issues for architectural re-examination. J. Object Technol. **6**(8), 39–49 (2007)
48. Lee, S.W., Moon, B., Park, C.: Advances in flash memory SSD technology for enterprise database applications. In: Proceedings of the ACM Conference on the Management of Data (SIGMOD) (2009)
49. Lim, K., Ranganathan, P., Chang, J., Patel, C., Mudge, T., Reinhardt, S.K.: Server designs for warehouse-computing environments. IEEE Micro **29**(1), 41–49 (2009)
50. Louth, W.: Metering the cloud: applying activity based costing (ABC) from code profiling up to performance & cost management of cloud computing. In: The International Conference on JAVA Technology (2009)
51. Mayer, M.: This site may harm your computer on every search results. Available online at http://tinyurl.com/cd76r3 (2009). Accessed on February 2009
52. Mell, P., Grance, T.: Perspectives on Cloud Computing and Standards. National Institute of Standards and Technology (NIST), Information Technology Laboratory (2009)
53. Murphy, M.A., Goasgue, S.: Virtual organization clusters self-provisioned clouds on the Grid. Future Gener. Comput. Syst. (Elsevier) **26**(8), 1271–1281 (2010)
54. Murphy, M.A., Abraham, L., Fenn, M., Goasguen, S.: Autonomic clouds on the Grid. J. Grid Computing **8**(1), 1–18 (2010)
55. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The eucalyptus open-source cloud-computing system. In: Proceedings of Cloud Computing and its Applications (2008)
56. OASIS: Security services technical committee, Security Assertion Markup Language (SAML). 2.0 Technical Overview Working Draft (2004)
57. Olston, C., Reed, B., Srivastava, U.: Pig latin: a not-so-foreign language for data processing. In: Proceedings of the ACM Conference on the Management of Data (2008)
58. Open Platform as a Service: Available online at http://www.openplatformasaservice.com/ (2009). Accessed on December 2009
59. Pete: App Engine outage today. Available online at http://tinyurl.com/2atu68l (2008). Accessed on April 2009
60. Rings, T., Caryer, G., Gallop, J., Grabowski, J., Kovacikova, T., Schulz, S., Stokes-Rees, I.: Grid and cloud computing: opportunities for integration with the next generation network. J. Grid Computing **7**(3), 375–393 (2009)
61. Ross, J.W., Westerman, G.: Preparing for utility computing: the role of IT architecture and relationship management. IBM Syst. J. **43**(1) (2004)
62. Siebeck, R., Janner, T., Schroth, C., Hoyer, V., Wörndl, W., Urmetzer, F.: Cloud-based enterprise mashup integration services for B2B scenarios. In: 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009) in Conjunction with the 18th International World Wide Web Conference (2009)
63. Software & Information Industry Association: Software as a service: strategic backgrounder (2001)
64. Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I: Capacity leasing in cloud systems using the

opennebula engine. In: Workshop on Cloud Computing and its Applications (2008)
65. Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I.: Virtual infrastructure management in private and hybrid clouds. IEEE Internet Computing **13**(5), 14–22 (2009)
66. Stern, A.: Update from Amazon regarding Friday's S3 downtime. Available online at http://tinyurl.com/2wtwdr7 (2008). Accessed on April 2009
67. The Cloud Computing Interoperability Forum (CCIF): Available online at http://www.cloudforum.org/ (2009). Accessed on March 2009
68. The Open Group, Service Oriented Architecture (SOA): Available online at http://www.opengroup.org/projects/soa/ (2010). Accessed on July 2009
69. The Open Group: Building return on investment from cloud computing. A white paper, cloud business artifacts project. Cloud Computing Work Group (2010)
70. Tsirogiannis, D., Harizopoulos, S., Shah, M.A., Wiener, J.L., Graefe, G.: Query processing techniques for solid state drives. In: Proceedings of the ACM Conference on the Management of Data (2009)
71. Tubanos, A.: theWHIR.com, FlexiScale suffers 18-hour outage. Available online at http://tinyurl.com/y965dv8 (2008). Accessed on January 2009
72. Turner, M., Budgen, D., Brereton, P.: Turning software into a service. Computer **36**(10), 38–44 (2003)
73. Vaquero, L.M., Rodero-Merino, L., Caceres, J., Lindner, M.: A break in the clouds towards a cloud definition. ACM SIGCOMM Comput. Commun. Rev. **39**(1), 50–55 (2009)
74. Vogels, W.: A word on scalability. Available online at http://tinyurl.com/yj2vpus (2006). Accessed on April 2009
75. Weinman, J.: The 10 laws of cloudonomics. Available online at http://tinyurl.com/5wv9d7 (2008). Accessed on 14 July 2009
76. Working Party on Information Security and Privacy: The role of digital identity management in the internet economy: a primer for policy makers (2009)