Private cloud for collaboration and e-Learning services: from IaaS to SaaS

**Frank Doelitzscher · Anthony Sulistio ·
Christoph Reich · Hendrik Kuijs · David Wolf**

Abstract   The idea behind cloud computing is to deliver Infrastructure-, Platform-
and Software-as-a-Service (IaaS, PaaS and SaaS) over the Internet on an easy pay-
per-use business model. The Hochschule Furtwangen University (HFU) is running
their own private cloud infrastructure, called Cloud Infrastructure and Application
CloudIA. The targeted users of the CloudIA project are HFU staff and students running
e-Learning applications, and external people for collaboration purposes. Therefore, in
this paper, we introduce our work in building a private cloud. More specifically, this
paper shows how our cloud offerings in each of the cloud service models, i.e. IaaS,
PaaS and SaaS, address the requirements and needs of e-Learning and collaboration
in an university environment.

F. Doelitzscher (✉) · A. Sulistio · D. Wolf
Department of Computer Science, Hochschule Furtwangen University, Furtwangen, Germany
e-mail: frank.doelitzscher@hs-furtwangen.de

A. Sulistio
e-mail: anthony.sulistio@hs-furtwangen.de

D. Wolf
e-mail: david.wolf@hs-furtwangen.de

C. Reich · H. Kuijs
Information and Media Centre, Hochschule Furtwangen University, Furtwangen, Germany
e-mail: christoph.reich@hs-furtwangen.de

H. Kuijs
e-mail: hendrik.kuijs@hs-furtwangen.de

1 Introduction

Although cloud computing is becoming very popular, it is difficult to get a clear definition of it. The key of cloud computing lies in its component-based nature, i.e. reusability, substitutability (e.g. alternative implementations, specialized interfaces and runtime component replacements), extensibility, customizability and scalability [1]. Foster et al. [2] discuss the basic concepts of cloud computing and identify the differences compared to Grid computing. In addition, Armbrust et al. [3] give a good overview of cloud computing by highlighting some of the obstacles and opportunities in the field.

From our point of view, cloud computing delivers Infrastructure-, Platform-, and Software as a Service (IaaS, PaaS, and SaaS) on a simple pay-per-use basis. For small- and medium-sized enterprises (SMEs), cloud computing enables them to avoid over-provisioning IT infrastructure and training personnel. Thus, SMEs can take advantage of using a cloud when the IT capacity needs to be modified on the fly. Typically, more resources are needed for services that are available only for a certain period. For example, AF83, a company specializing in social networking and live web solutions, used Amazon IT infrastructure to deliver a live concert via the Web and mobile devices [4]. The concert attracted 7,000 simultaneous users. By using cloud computing, AF83 avoided purchasing new hardware for this special event, and delivered a successful event in a short amount of time.

For companies with large IT infrastructure, such as Amazon and Google, becoming a cloud provider allowed them to offer their resources to SMEs based on pay-as-you-go and subscription models, respectively. Because not all services require full resources at the same time for a long periods of time, these companies can still use and lease their existing infrastructure with a relatively small cost. Hence, they can reduce the total cost of ownership (TCO) and increase hardware utilization [3].

In a typical university scenario, PC labs and servers are under-utilized during the night and semester breaks. In addition, these resources are on high demand mainly towards the end of a semester. Moreover, cloud computing can also be used to support e-Learning services [1,5,6]. With the aforementioned motivations and scenarios, the Hochschule Furtwangen University (HFU) acknowledged the potential benefits of cloud computing. As a result, HFU established a new project called Cloud Infrastructure and Application (CloudIA).

The targeted users of the CloudIA project are HFU staff and students running e-Learning applications and external people for collaboration purposes. Therefore, in this paper, we introduce our work in building CloudIA. More specifically, this paper shows how our cloud offerings in each of the cloud service models, i.e. IaaS, PaaS and SaaS address the requirements and needs of e-Learning and collaboration in an university environment. In addition, for PaaS and SaaS, our work interacts with Amazon for scalability and high-availability purposes.

The rest of this paper is organized as follows. Section 2 provides some related work, whereas Sect. 3 explains the CloudIA infrastructure. Section 4 shows the potential usage of the CloudIA project by providing IaaS to HFU students and staff. Sections 5 and 6 present PaaS and SaaS use case scenarios in HFU, respectively. Finally, Sect. 7 concludes the paper and shows future work.

2 Related work

Dong et al. [5] proposes the use of cloud computing as a base for modern e-Learning applications. Hence, these applications can leverage cloud computing for dynamic assignable storage and compute resources. To realize this objective, the paper presents a general and simple architecture with ad-hoc modules, such as monitoring, policy and provision. However, no concrete implementations and e-Learning scenarios are presented. The CloudIA offerings, presented in this paper, demonstrate the feasibility of cloud computing for e-Learning services. In addition, the CloudIA architecture (discussed in Sect. 3) addresses other important functionalities for enabling an e-Learning ecosystem in the cloud, such as authentication and integration with existing IT infrastructure, creation of customized on-demand virtual machines, and integration with a public cloud provider.

BlueSky cloud framework [6], developed by Xi'an Jiaotong University (China), enables physical machines to be virtualized and allocated on-demand for e-Learning systems. The BlueSky framework has similar architectural layers as the CloudIA project, such as layers dealing with physical resources, provisioning, monitoring, and user interface. However, the BlueSky framework does not have a security layer like the CloudIA architecture. Security plays a major role in ensuring user access policies, and enabling a trust among education providers and users. In addition, the BlueSky framework only focuses on delivering IaaS, whereas the CloudIA project offer IaaS, PaaS and SaaS.

The Virtual Computing Laboratory (VCL) [1], developed by North Carolina State University (USA), enables students to reserve and access virtual machines (VMs) with a basic image or specific applications environments, such as Matlab and Autodesk. While the also VCL manages physical computing hardware, the CloudIA project concentrates on virtual machines. Therefore, our work does not need any specific server hardware or blade centers as a computing base. However, the VCL does not offer collaboration features such as CollabSoft (see Sect. 6). The VCL offers IaaS and PaaS platforms which could then be used to host collaboration systems (SaaS) on top of it.

The Snow Leopard Cloud [7] provides PaaS for North Atlantic Treaty Organization (NATO) to run its various military exercises and mission events. In addition, Snow Leopard Cloud is used to run Web 2.0 applications, such as video teleconferencing, voice over IP, and remote management over handheld devices and terminals. As the Snow Leopard Cloud is targeted towards military usage, it has multi-level security and the network infrastructure is encrypted. In the CloudIA project, high privacy and security standards are respected by using a single-sign on approach.

In a business environment, the RESERVOIR project [8] proposes a modular and extensible architecture to support federation of cloud providers. In the RESERVOIR model, the providers are independent and clearly separated into two roles, i.e. service and infrastructure. The service providers offer SaaS, whereas the infrastructure providers offer IaaS. In contrast, our work does not have this distinction, since HFU, as a cloud provider, can offer both IaaS and SaaS. Moreover, the main focus of our work is to build a private cloud on an existing infrastructure. Thus, federation of cloud providers is considered as a secondary objective.

The Google App Engine [9] provides a Java web framework. It is based on the servlet container Jetty and BigTable for data storage. Applications written for Google App Engine are scaled automatically by Google. Similarities to the Servlet Container Platform (SCP) in Sect. 5 are the uses of servlets. For the data storage CloudIA uses MySQL. The advantage of SCP is the complete support of the newest standards (e.g. Servlet 2.5 and JSP 2.1). Hence, existing servlet applications do not have to be rewritten, since they use a SQL-compatible database, instead of Google's BigTable. The disadvantage is not as scalable as Google applications.

3 Private cloud architecture: CloudIA

To harness the potentials of cloud computing for internal usages at the university, the Cloud Infrastructure and Application (CloudIA) project has been established. The main objective of this project is to build a private cloud for the purpose of creating on-demand infrastructures and running e-Learning applications, such as Servlet Container Platform (see Sect. 5) and on-demand Collaboration Software (CollabSoft) (see Sect. 6).

The CloudIA project is a joint collaboration between IT (IMZ) and Computer Science (CS) departments at HFU. IMZ has a role in maintaining IT services in HFU and providing e-Learning services to HFU staff and students. Thus, IMZ plays an important role in integrating CloudIA offerings into existing HFU's IT infrastructure.

Figure 1 shows a general overview of our HFU's private cloud architecture that builds on top of an existing hardware infrastructure. It consists of three computer pools (PC Pool, Research Pool and Server Pool), that are located at different locations and use separate IP subdomains within the university. The PC Pool has 18 computers with Ubuntu operating system (OS) and KVM [10] installed, whereas the other pools have Debian OS and Xen [11] configured. The PC and Server pools are used by students and staff for teaching purposes during the day. In contrast, the Research Pool is used for research and development purposes.

The pools are managed by a CloudIA's Cloud Management System (CMS), as shown in Fig. 2. The CMS is divided into several layers for extensibility and maintainability. However, the *Monitoring and Management* and *Security* components are incorporated across all layers to ensure high reliability and secured services. The descriptions of each layer are:

– *User interface layer*: This layer provides various access points to users and/or administrators of the CMS in accessing our cloud system
– *Business layer*: This layer aims to regulate resource supply and demand through the use of economy (price) and Service-Level Agreements (SLA). In addition, this layer enables users to reserve VMs in advance and manage their personal VMs
– *System layer*: This layer is responsible for daily operation of the CMS, such as submitting jobs, managing user accounts and monitoring Quality of Service (QoS)
– *Resource interface layer*: This layer deals with the physical hardware. It provides interfaces and plugins to various virtualization, database and distributed systems as well as other technologies, such as Xen, Amazon EC2, Amazon S3, and Nagios [12]
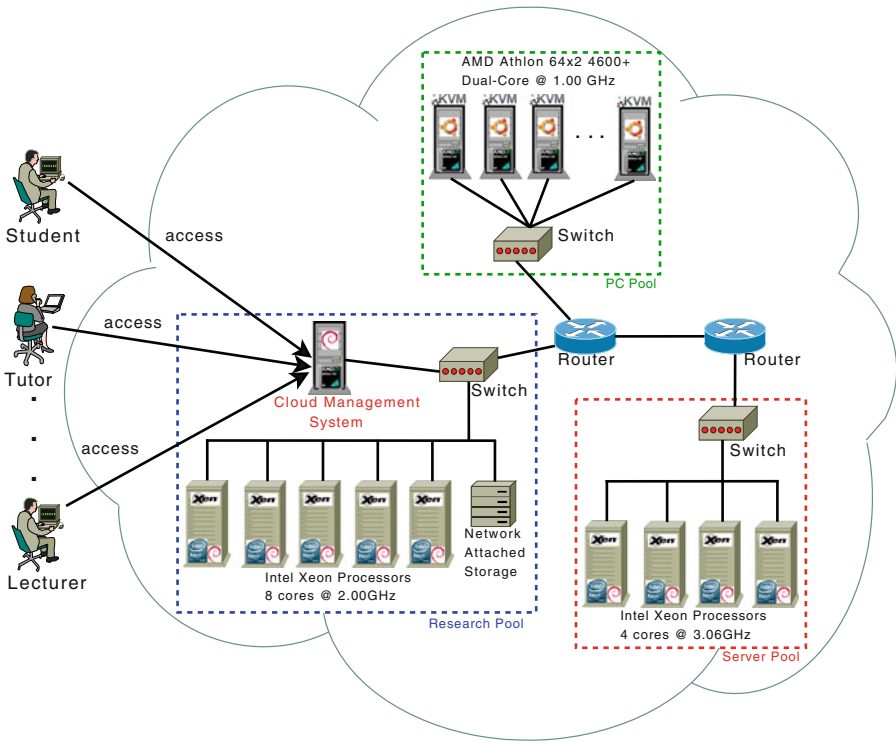
Fig. 1 Overview of the HFU's private cloud architecture, CloudIA
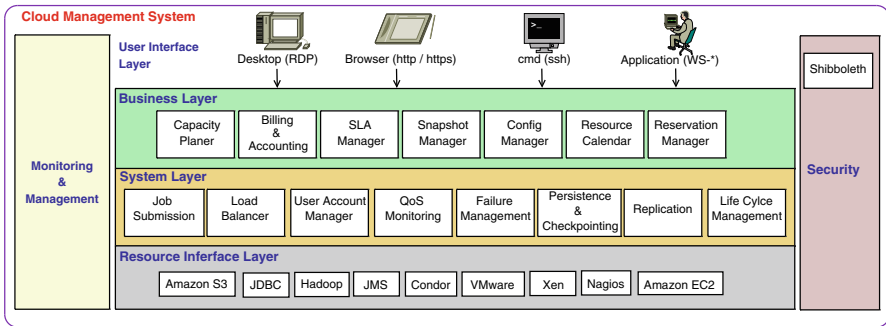

Fig. 2 Cloud management system of CloudIA

– *Monitoring and management component*: To ensure the reliability of each layer in the cloud, a monitoring and management component is needed. Thus, this component allows the system administrator to monitor and to initiate activities of each layer. In case of failures, conflicts with SLA objectives, under- or over-utilized resources

– *Security component*: To ensure the privacy, recovery, integrity and security of user data and transactions, a security feature on all layers is required. Besides the

technical solutions, issues in areas such as regulatory compliance and data auditing are important. Therefore, this component also addresses these issues

In this paper, we only focus on describing some components from the *System* and *Security* layers to address different needs of students and staff. Few components from the *System* layer, such as *Job Submission* and *QoS Monitoring* will be discussed in Sect. 4, whereas the *Shibboleth* component of the *System* layer is explained next. Finally, the use of Amazon services, such as EC2 and S3 will be mentioned in Sect. 5.

3.1 Single sign-on using Shibboleth

To avoid multiple user managements, HFU bases their authentication systems on a Single Sign-On (SSO) solution using Shibboleth [13]. This includes access to the CloudIA platform, and the collaboration and e-Learning services, such as SCP and CollabSoft. Therefore, we use Shibboleth for a seamless integration between these cloud services and existing IT infrastructure.

Shibboleth's main components are Identity Provider (IdP), Service Provider (SP) and Discovery Service for localization of the IdP. Federations can be build easily, where a person authenticates through his/her home IdP and the authorization information is retrieved by the SP of that home IdP.

Shibboleth can be used for accessing the CloudIA platform. The example (Fig. 3) shows the sequence of calls using Shibboleth in a multi-tenant application (e.g. Media Wiki). One requirement of a Shibboleth architecture is that all SP present a valid certificate. Since the name of the instance is mapped dynamically, the certificate of a instance is set at runtime. Therefore, CloudIA has a pool of pre-created certificates for possible Service Providers.

Figure 3 explains the steps before a user can use a web service. (1) The admin requests a web service to start at the CloudIA platform. (2) A VM with the installed web service (e.g. wiki) is started. (3) The relevant SP certificate is copied to the VM (e.g. certificate SP1). (4) The Shibboleth configuration is processed (e.g. URL of the customers IdP is set). and (5). The web service is started. Now a Shibboleth enabled web service is running. Next Shibboleth takes over and exchanges the meta data of the IdP and the SP. Note that the URL of the SP must be known by the IdP as "relying party". (7) User calls service the first time. (8) The user must authenticate at his/her IdP, and if successful can then use the service.

3.2 Early experiences in building a cloud infrastructure

Currently, many organizations are building their own private clouds from their existing infrastructure [3]. Our experiences and difficulties of building a cloud infrastructure are as follows:

– *Internal IT policies*: Our university has two departments which are responsible in maintaining and managing IT resources, i.e. IT and Computer Science departments. Each department has its own PC pools, data centers and secured network. Thus, with this separation, each department has their own firewall rules and IP
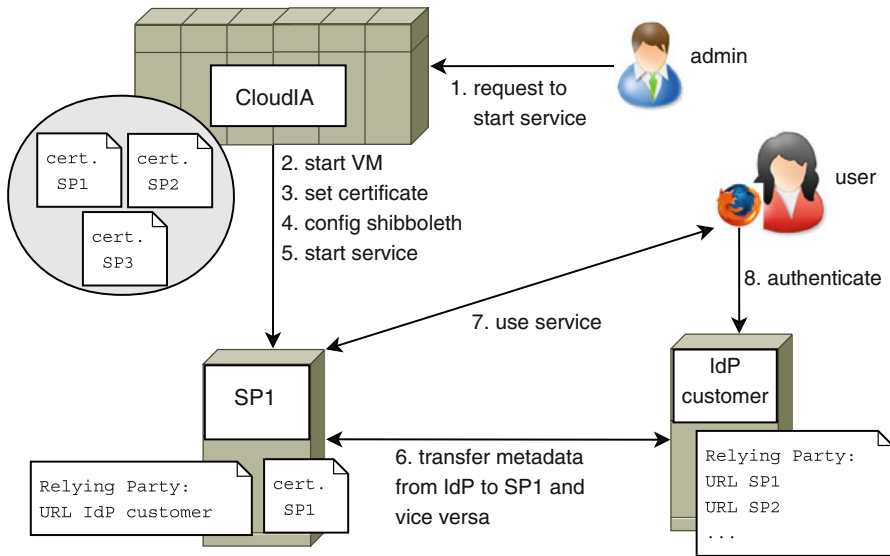
Fig. 3 Using Shibboleth for a single sign-on solution on the CloudIA architecture

subnets. We found that this was an obstruction to building a cloud computing infrastructure using heterogeneous resources in the university. We later faced a problem of running out of IP addresses, due to the dynamic creation of virtual instances on the host PCs that belong to same subnet. As the network plays a critical role in the whole concept of cloud computing, an organization wanting to create a private cloud needs to be wary of the IT policies and practices deployed across internal divisions

– *Running appropriate services and applications on the cloud*: Hosting a particular service or running a legacy application on the cloud can be counterproductive. For example it may slow the application down. In addition, it may consume the whole physical resource and prevent other VMs from running. In our university's time table and room management software, `TimeEdit` [14] was tested on a VM on the HP ESX server. Although the hardware specification meets the requirement of running several virtual instances, `TimeEdit` constantly consumed the available resources. This is because the underlying database of `TimeEdit` could not be efficiently run on a VM. Hence, it is important to monitor VM consumption periodically, and to run resource-intensive services and applications on dedicated machines

4 Infrastructure as a service: automatic creation of customized VMs

Each semester, lab exercises and practical projects require hardware with specific software requirements for student projects. Fourth and sixth semester students of Business Informatics and Computer Science faculties are required to take two practical semester projects in various topics such as designing and implementing new software

architectures, and evaluating new networking or programming techniques. Therefore, a big pool of project PCs (mostly state of the art hardware) needs to be provided by the HFU's IMZ. These project PCs are used throughout the semester and they need to be maintained and re-configured every 6 months. This section shows that by offering these PCs as VMs, the CloudIA project helps IMZ improve productivity in managing various hardware and software requirements for the students' projects.

4.1 VM creation

Instead of providing physical machines (e.g. server and desktop PCs), IMZ can provide VMs. For example, for networking exercises like routing labs or firewall configuration, each group needs multiple PCs to complete the exercise. VMs are suitable for this purpose. However, the VMs need to be created, managed and monitored automatically. Thus a web front-end (see Fig. 4a) was developed, where students can login and create, suspend, or delete VMs using a wizard, according to their project demands. Per default, a maximum of three VMs per student with a total CPU time of 100 h per semester, and a maximum of 1GB RAM for each VM is set. Unique amongst to other IaaS providers is the creation and monitoring of VMs in CloudIA. In comparison to other systems that provide customers with preconfigured virtual machine images, our approach allows users to decide which software should automatically get installed during the VM's creation time.

The creation of VMs is done using minimal operating system (OS) images (e.g. Debian Linux), and adds software packages on demand. To realize this approach, a Fully Automatic Installation (FAI) [15] service has been included into the CloudIA platform to provide the necessary services. Figure 5 shows the involved components of the automatic creation of customized VMs process. The major components are at
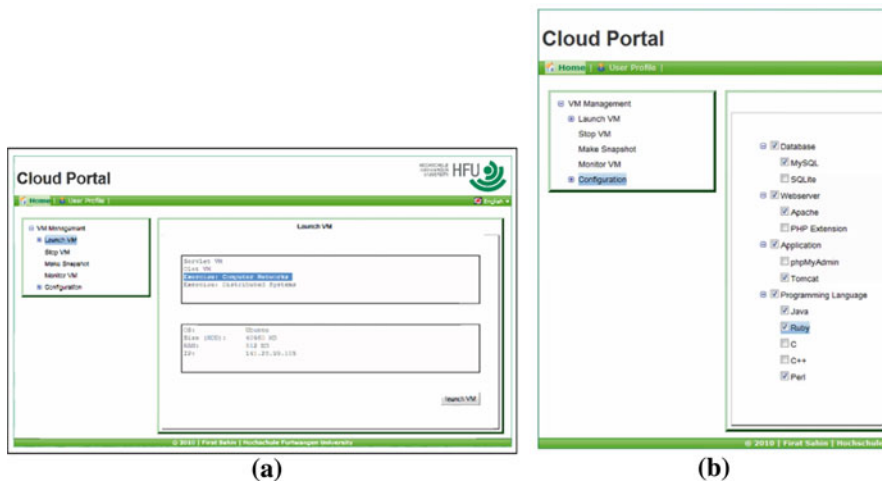

Fig. 4 CloudIA IaaS webfrontend: **a** Webfrontend to create/launch a virtual machine, **b** software package tree (*screen cutout*)
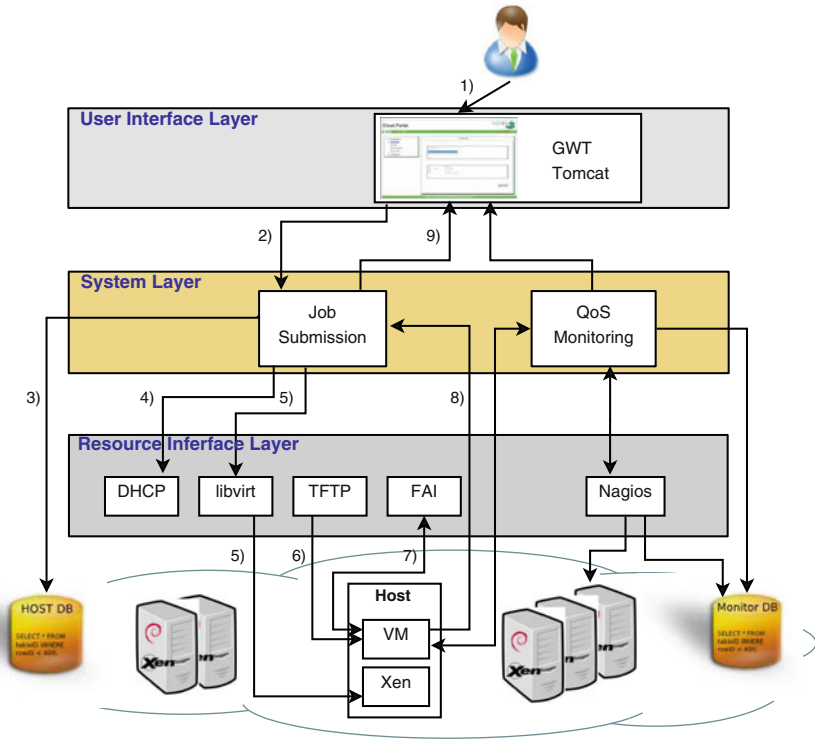
Fig. 5 Automatic VM creation architecture

the resource interface layer, such as DHCP (by providing IPs), libvirt (an interface to manage VMs), TFTP (for image transfer), and most importantly the FAI service (for image creation).

The following steps describe the installation process of a new VM in a chronological order. The user who wants to create a new VM needs to access the CloudIA's IaaS component through a web portal (1), as shown in Fig. 4a. Login is handled by Shibboleth which authenticates against HFU's identity system, i.e. OpenLDAP. The web portal provides a wizard to create new VMs. Common (virtual) machine's parameters like CPU, RAM size and hard disk size are asked before the user can choose software packages from a software tree (Fig. 4b), which then gets automatically installed on the VM. After finishing the wizard, an XML description of the new VM is created and transferred via GWT-RPC to the *Job Submission* module of CloudIA (2). The module contacts a host database (3) to find a free (physical) host for the new VM and a corresponding IP address. The host database gets updated with the resources needed and an individual DHCP entry for the VM gets created on the DHCP server (4). The *Job Submission* module forwards the XML description of the VM to the libvirt [16] module on the (physical) target host (5). According to the XML description, a VM gets created using the libvirt API.

The newly created VM boots with a network boot protocol (PXE) enabled. Then, the VM downloads a Linux kernel via a TFTP network from the FAI server (6).

The VM boots the kernel from memory. Hardware detection then takes place, the hard disk gets partitioned, and according to the chosen software packages, a corresponding FAI class gets contacted. This starts the automated installation of the necessary software packages (7).

To reduce transfer time and network bandwidth consumption, the software packages are downloaded from the university's local software repository (e.g. a Debian mirror). In addition to the package installation, customized scripts are executed to change the VM's root credentials, to delete temporary log and package files, and to inform the *Job Submission* module about the completion of the installation (8). Afterwards, the VM boots off its local file system and is ready for the user. The *Job Submission* module informs the web portal that the VM is available (9).

The details of the software configuration are stored in the configuration files on the FAI install server. These configuration files are shared among groups (classes) of computers. With this class concept, FAI does not care whether 10 or 100 VMs need to be installed, aside from performance and latency effects [15]. Thus, there is no need to create a configuration file for every single VM. Since an updated mirror of software is used, new VMs are always at the actual patch level. From our experience, FAI offers a scalable method for installing customized software packages into the VMs.

4.2 Monitoring of virtual machines

A challenge of using VMs is that the infrastructure seems to be infinitely available. While allocating physical hardware was considered very carefully, new virtual machines are deployed less deliberately by administrators. This behaviour is supported by the overbooking feature of physical hardware. However, this approach has disadvantages as well. Physical hardware is still limited. Therefore, a monitoring component is needed to provide quick overview about physical and virtual resources. Quality of Service (QoS) serves as an assurance of performance characteristics to individual applications during runtime. Such characteristics can be CPU load, memory usage and network traffic. By monitoring these QoS parameters, IMZ ensures a certain level of performance of the virtual infrastructure.

Monitoring for IMZ administrators: For monitoring the physical hosts, Nagios [12] is used. Its configuration allows the administration to be alerted via several communication channels, e.g. via an e-mail when servers and services go into a critical stage, if a heavy load for a certain duration is recognized. Another advantage of Nagios is that these hosts and services can be visualized through a web browser. NagiosGrapher [17], a plug-in for Nagios that displays the real-time data into an online chart, is used to provide a real time overview of the infrastructure. A Simple Network Management Protocol (SNMP) is used to monitor the VMs with regards to CPU and memory usages, and network bandwidth. SNMP uses an UDP-based network protocol and therefore requires an agent to be installed in each VM. The *QoS Monitoring* module of CloudIA acts as an SNMP master that periodically collects real-time information from agents and stores the data into a monitor database. This enables IMZ administrators to monitor the existing infrastructure and also shows the utilization over larger periods of time.

This allows a better decision about the deployability if additional VMs are required for a project.

Monitoring for cloud users: Compared to many other IaaS systems available on the market, CloudIA shows the user about the progress of the installation until the VM is ready to use. For example, Amazon EC2 customers only get a "processing" status, without any further information. Using the FAI concept for automatic VM creation, the CloudIA system is able to present the user a more detailed status of the overall installation process. A prototype GUI pictured in Fig. 6 shows the different steps. Experiments show that an average VM installation time of 5 min can be achieved depending on the amount of software which needs to be installed additionally on the VM.

4.3 Advantages for the IMZ

Before using this approach, PCs were assigned among project groups at the beginning of each semester project. Changes to the number of projects in each semester make it difficult to predict on how much hardware will be needed. Often hardware requirements change throughout the project, so additional hardware may be needed. At the end of each semester project, PCs need to be collected and maintained. Maintenance of these PCs includes a functionality check of all components, secure wiping of installed software, and new installation of a different OS.

An internal study shows that for each used project PC, this process takes up 15 min of an IMZ administrator's time. This sums up to a considerable high amount depending on how many project PCs were used. Using CloudIA's IaaS module optimizes this process to a total of 2 h work for an administrator for a complete semester. Since eligible students are able to create their own VMs, no additional hardware needs to be provided. Due to the restriction of CPU hours, students are forced to use the infrastructure in an economic and responsible manner. It also enables an overbooking of existing hardware, since the maximum possible number of VMs can be higher than physical hardware exists. As a result, a better utilization of existing hardware is obtained.

For fixed exercises which are part of certain lectures from different faculties (e.g. routing labs), predefined VMs are provided. Thus, all attending students use the same development environment which is also known to the lecturer. Errors due to OS misconfiguration on missing software components or missing programming libraries can also be avoided. For all other semester projects, the approach using FAI as an automatic VM creation mechanism provides the following advantages:

Fig. 6 Installation status provided by FAI

- Automatic management of software package dependencies
- Up-to-date patch level of new installed systems
- Low network load since VM images are created directly on the target host
- Detailed information of VM creation status

5 Platform as a service: servlet container platform

Many programming courses at HFU require special programming environments for students. The proposed PaaS is a Servlet Container Platform (SCP) for HFU courses, such as middleware, Java frameworks, and web frameworks. Thus, this platform allows students to develop, deploy and test their own applications in a servlet container, without the need of software installation and configuration.

In order to reduce a high learning curve for students, course coordinators can create VMs using the approach described in Sect. 4. For example, a VM image containing MySQL, Tomcat, PHP, and Apache web server is created. Since the VMs are running in isolation, the IMZ administrator can integrate SCP easily by developing a web front-end embedded in Online Learning And Training (OLAT) [18], a web-based open source Learning Management System (LMS) used by HFU, for authentication purposes.

After starting an instance in the SCP, students are able to reserve VMs, e.g. for 100 h with predefined packages and software for their courses. To access these VMs, students login to the web portal, as shown in Fig. 7, using HFU's SSO mechanism. After authentication, the students are able to use servlet container VMs with the following functionalities:

- Start and stop servlet container instances, since students can only use these instances for a limited amount of time. The time limitation is introduced to encourage students to utilize SCP optimally and effectively
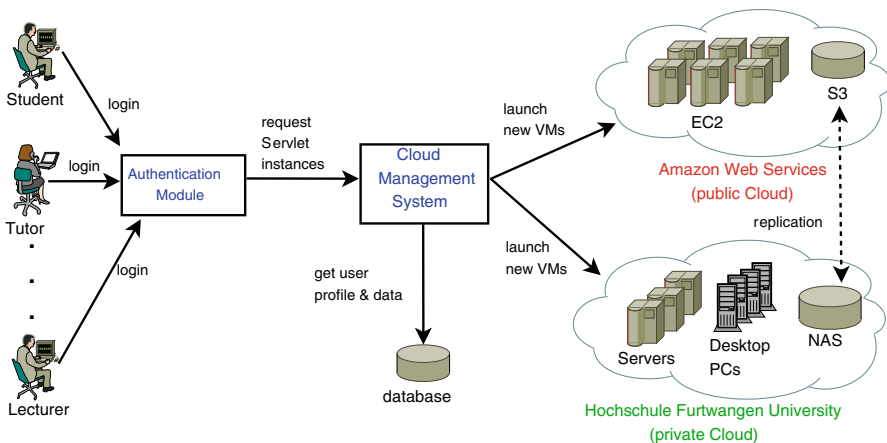

Fig. 7 High level overview of running the servlet container in the cloud

– Save and delete snapshots of work. In this scenario, a snapshot refers to the Tomcat and MySQL configuration details and data, such as `*.war` files and database tables
– Start a VM based on a saved snapshot
– Downloads of snapshots to a local hard disk. Thus, students only need to send their snapshots to submit their exercise to an examiner

By default, servlet container instances are running on HFU's private cloud. However, if there are not enough resources available, instances can be outsourced to a public cloud, i.e. using Amazon's small EC2 instances, as shown in Fig. 7. Currently, internal IP and Amazon's dynamic public DNS addresses are displayed in the web portal, to allow students to access Tomcat and MySQL through the Web. However, these addresses are no longer valid for new instances. In the future, a DNS server will be installed to enable the mapping of the internal IP and Amazon's dynamic DNS of a VM to public names. Therefore, students can access their instances using the same website address throughout semester.

Once the students stop or shutdown their instances, only their snapshots are saved. The VMs and/or Amazon Machine Images (AMIs) are deleted to save storage. Therefore, these snapshots are replicated between Amazon S3 and our internal storage. In addition, Amazon S3 stores a template AMI for the servlet container with the same basic configurations as the VM hosted on HFU.

5.1 Advantages of servlet container platform

By having SCP as PaaS in the cloud, students can start working on their projects, instead of spending time on installing a webserver, Tomcat and MySQL on their computers. The same scenario can also be applied to a tutor or lecturer in testing and marking the students' work. In addition, students can start a basic VM if they misconfigure the Tomcat and/or MySQL configuration details.

The concept of SCP is similar to Google App Engine [9]. Both provide a servlet container. However, the major difference is the data storage. Google uses BigTable, whereas SCP uses MySQL. Our approach has an advantage where students can program their code with standard APIs (e.g. JDBC), instead of Google's specific APIs.

Another advantage is that SCP can be used multiple times for students and staff for different purposes, such as lab modules, projects, experiments, and teaching environments. Therefore, SCP as PaaS provides a fool-proof environment and enhances the e-Learning experience. In addition, SCP is easy to maintain and upgrade, since software updates and installations are needed only on the master VM. Moreover, SCP supports high availability since VMs and data are replicated on both Amazon S3 and our internal storage.

Since SCP has been included into OLAT (see Sect. 6), IMZ can support students and staff through the use of thin clients, which enable the usage of higher computing resources without purchasing new hardware. As a result, IT costs on hardware purchase, hardware maintenance, and administration can be reduced.

In addition, with this approach, IMZ can monitor the usage of SCP to control the use of Amazon EC2 and to prevent miscellaneous access, since IMZ is reponsible for

the payment of Amazon services. Then, IMZ can charge VMs running on Amazon to the appropriate faculties.

6 Software as a service: on-demand Collaboration Software (CollabSoft)

For e-Learning and lecture support, IMZ at HFU provides OLAT (Online Learning And Training). OLAT offers a flexible online course system expanded by the versioning system Subversion (SVN) and a XMPP based instant messaging server (Jabber). In OLAT, it is very simple to setup collaboration environments (working groups) themselves. Each user can create their own working groups and manage them easily by adding various functionalities, such as calendar, wiki, forum or chat.

Our experience at the HFU has shown that students adapted to the OLAT platform very well. OLAT is used for organizing programming exercises and storing course materials, as a central project management platform and as a web-based blackboard to share and discuss exercise solutions using wiki and forum functions. Figure 8 shows active OLAT users over 1 month—consider that HFU has 4100 active students in total, distributed over three campuses.

6.1 The need for OLAT on-demand

Acknowledging the possibilities of the OLAT system, several faculties at the HFU started developing special requirements for this system for very special tasks. For example, the faculty of Product Engineering arranges a two-week OLAT-based online assessment center once every semester. Web-based exercises, quick tests and group tasks are hosted in a special OLAT course which is only accessible by participants. To prohibit cheating, only this special course must be available in OLAT. After the assessment period, this OLAT installation gets shutdown until the following semester.

Another example is the HFU learning services department that organizes an e-Learning course with a final online exam once per semester. Therefore, a single OLAT instance needs to be provided preventing cheating as mentioned earlier. Special events, like the 20-year celebration of the faculty of Digital Media, require a collaboration and project management platform which can be OLAT. Using the existing
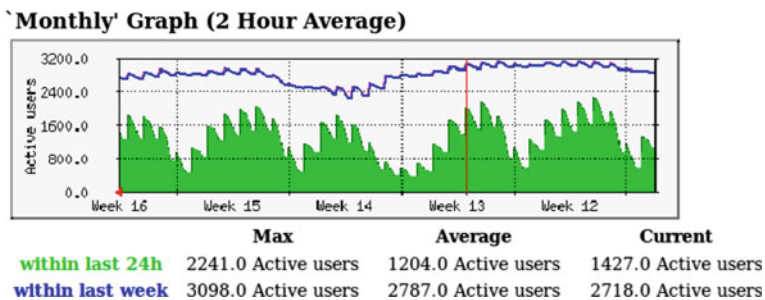

	Max	Average	Current
within last 24h	2241.0 Active users	1204.0 Active users	1427.0 Active users
within last week	3098.0 Active users	2787.0 Active users	2718.0 Active users

Fig. 8 Active monthly OLAT user in HFU

HFU-wide OLAT would require creation of user accounts for external project partners as well. By providing a separate OLAT installation for this event, HFU students and employees work in a known environment while external partner do not automatically get access to HFU internal OLAT courses.

It becomes clear that there is a high demand for OLAT systems aimed at a special event, existing only for a limited time. In contrast, installation and configuration of an OLAT system takes a whole day done by an experienced OLAT administrator of the IMZ. Difficult parts include the initial connection to HFU's LDAP [19] based user directory and adoption of HFU's corporate design. Therefore, the on-demand Collaboration Software (CollabSoft), developed by the IMZ, provides an on-demand OLAT system installed in a VM with customized add-ons.

Figure 9 describes the architecture of a CollabSoft VM. The VM image contains an Apache web server, the servlet container Tomcat with the OLAT software, a MySQL database server, an instant messaging server and the versioning system Subversion. A PHP module connects the SVN and the service GUI. There are three different possibilities for the user management in CollabSoft:

– *Direct administration* is suited for a small number of users. The OLAT administrator uses an administrator GUI to create/delete individual accounts
– *Lightweight Directory Access Protocol (LDAP)* is a common directory service for data management of identities. Due to the configuration at the beginning of a new CollabSoft instance, it is possible to integrate an existing LDAP server of an organization by configuration of its URI. After authentication of the CollabSoft administrator to the CloudIA platform and starting a CollabSoft instance, the user known by the LDAP can access the instance
– *Shibboleth*: If an institution is member of a federation, such as DFN-AAI [20], it can provide Single Sign-On (SSO) to the services of their member. Only minor configuration changes to the CollabSoft instance are necessary. Compliance with common rules and standards is guaranteed by the membership of the federation. Authentication and accesses to the respective service providers can be made to the Identity Provider (IdP), the home organization. For a more detailed description see Sect. 3.1
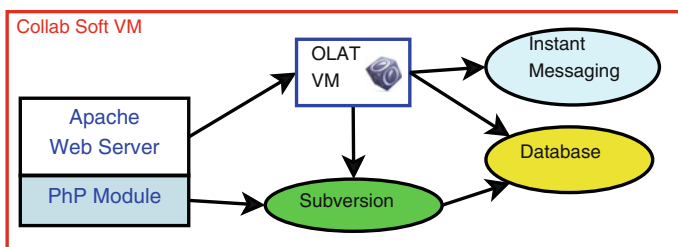

Fig. 9 Collaboration Software (CollabSoft) running inside a virtual machine

6.2 CollabSoft performance evaluation

To evaluate the performance of CollabSoft in the HFU's private cloud and its deploy-ability to a public cloud, a performance evaluation was done. A CollabSoft virtual machine was deployed on a single cloud host in CloudIA. The same one was also exported to Amazon's public cloud. Due to RAM limitations, *large* and *extra large* Amazon instance were used. The hardware configuration of the hosts is listed in Table 1.

Table 1 Hardware Configuration for the CollabSoft Experiment

	CloudIA instance	Amazon	
		Large	Extra large
No. of cores	8	2	4
CPU type (Intel(R) Xeon(R))	E5504	E5430	E5430
CPU freq.	2.00 GHz	2.66 GHz	2.66 GHz
Total mem (kB)	12,330,508	7,872,040	15,736,360
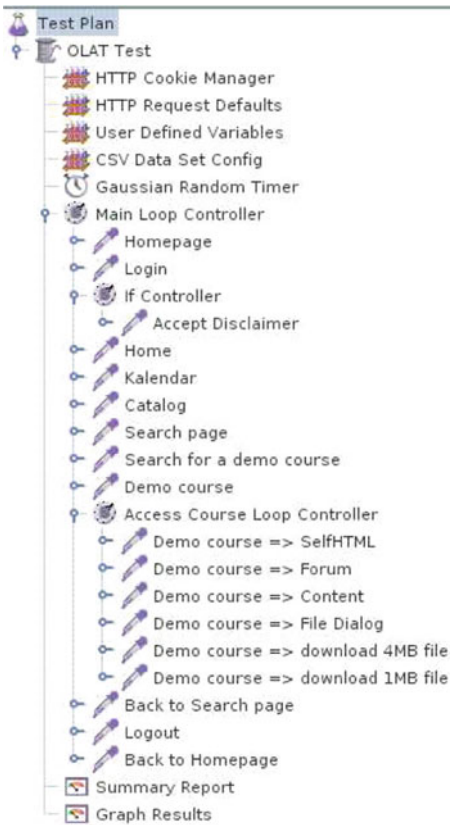
Fig. 10 JMeter screen shot of the test plan

Test Case No.	Test Case Description
t1	go to start page (dmz)
t2	login
t3	calling home address
t4	usgin calendar
t5	using catalog
t6	search for document
t7	search for demo course
t8	using demo course
t9	using SelfHTML in demo course
t10	using forum in demo course
t11	using content page in demo course
t12	using file dialog in demo course
t13	download 4MByte file
t14	download 1MByte file
t15	search for demo course
t16	logout
t17	go to start page (dmz)

Fig. 11 JMeter test plan with index

After deployment, each CollabSoft instance was stressed with a customized benchmark using Apache JMeter [21]. JMeter is designed to interactively create test plans for a load test. The CollabSoft test plan pictured in Figs. 10 and 11 was used. It represents a typical usage scenario of a user, e.g. login, search for a specific course, use the course, use a forum, up or download files, etc. All interactions have been done 1,000 times with about 100 users.

The most important measurement for the user is the request/response time during the test. Figure 12 compares this measurement between the three different hardware platforms (CloudIA, Amazon Large, Amazon Extra large). It can be seen that at t2 "login" and at t8 "using demo course" the response time of the CloudIA instance is slightly slower. This is an exception and can not be re-produced.

Overall, the experiment shows that at peak time, when the private cloud is fully utilized, the CollabSoft can operate with acceptable response times even in Amazon.

6.3 Advantages of CollabSoft

With the CollabSoft VM image, the IMZ is now able to setup CollabSoft on the HFU's private cloud architecture or any other public cloud provider in a matter of minutes by defining essential configurations, such as an LDAP host, disk space and a logo. The main strength of CollabSoft comes with the functionality of OLAT. The system is used by students for on demand lab exercises and by researcher groups for collaboration purposes (e.g. CMS).
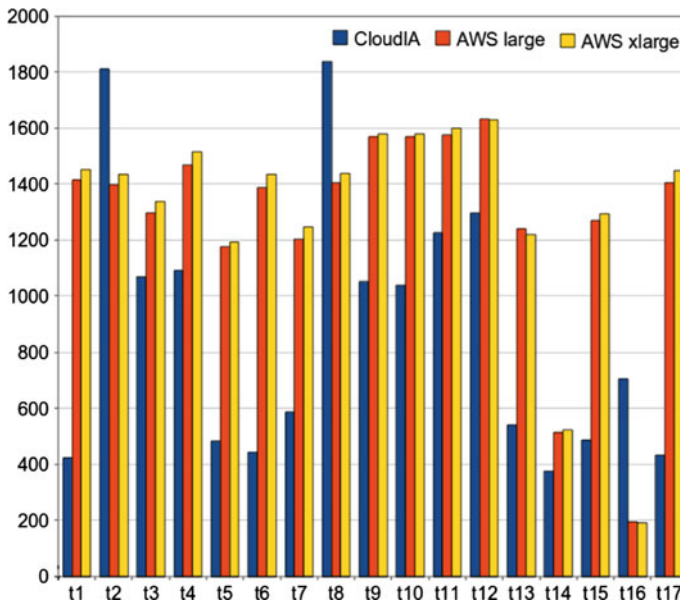

Fig. 12 ReqResTimeAll

HFU faculties are using CollabSoft as general project management software. With CollabSoft they are using a tool which they are used to and which follows the HFU's corporate design. This optimizes the working experience and strengthens HFU's collaboration with external partners. The latest project using a CollabSoft VM is a new HFU alumni platform.

Several German universities and established research organizations are organized in an "Authentication and Authorization Infrastructure" (DFN-AAI). While they are physically distributed within Germany (and no centralized user directory exists), members are able to login to a DFN-AAI enabled service using Shibboleth. With the flexibility of CloudIA and CollabSoft, IMZ is able to provide an *IMZ wiki* to share information about common IMZ problems (every German university has an IMZ) and especially Shibboleth-related solutions. A DFN-AAI member can login to the wiki using his/her existing user name and password which gets validated through the trust relation provided by Shibboleth. Since CollabSoft is fully Shibboleth-enabled, a new DFN-AAI collaboration service an be created within a couple of hours without importing user-data from remote member-directory-services. Figure 13 shows a map of all DFN-AAI members who are using the wiki.

Fig. 13 Map of German DFN-AAI member using IMZ wiki

The following list sums up the advantages of CollabSoft:

– Fast setup time of a new collaboration/project management environment
– Uniform environment that follows a corporate design
– Self adminstration by the user
– No administration of single users is needed if using Shibboleth Only configuration of access level by attributes and rules is required
– No additional user management is necessary if used by HFU organizations due to preconfigured connection to LDAP directory
– Persistent storage of instances is possible due to snapshots of a CollabSoft VM

7 Conclusion and future work

This paper shows how cloud computing helps Hochschule Furtwangen University (HFU)'s IT department improve periodic IT tasks such as the management of project PCs with a standard development environment for programming exercises. HFU's private cloud architecture called Cloud Infrastructure and Application (CloudIA) was explained and its special issue of Single Sign-on (SSO) using Shibboleth has been emphasized.

Unique amongst to other Infrastructure as a Service (IaaS) frameworks, CloudIA creates a virtual machine by choosing a base image and post installing software packages selected by the user on the fly. This approach gives high flexibility, saves disk space, guarantees installation of newest software versions, and enables monitoring of the installation progress. The Platform as a Service (PaaS) approach is supported by a Servlet Container Platform and is mainly used for study courses at the HFU. It has been integrated into HFU's e-Learning management system with the support of SSO. A successful collaboration application, Collaboration Software (CollabSoft), has been adapted to a Software as a Service (SaaS) model with support for different authentication technologies. Due to its flexibility, it is now possible to support heavy demands on a collaboration platform with an instant solution which HFU affiliates are already familiar with.

This paper also presents a case for cloud computing where it can be used to support collaboration and e-Learning services in a university environment—from IaaS to SaaS. Therefore, our work on cloud computing can be used as an example for other universities to follow.

As for future work, further extensions of CloudIA's IaaS systems are planned. First trials indicate that users often choose a very similar configuration. Hence, it will be more efficient to provide several templates and add individual software configurations to them. Furthermore, integration of different user management systems will be developed to support connection to existing external user directories such as a partner's LDAP server. Thus, CollabSoft could be offered to external parties as well. Finally, CloudIA's monitoring module needs to be extended to support efficient Quality of Service (QoS) monitoring to optimize infrastructure usability in regards to the green IT paradigm.

References

1. Vouk M, Averritt S, Bugaev M, Kurth A, Peeler A, Schaffer H, Sills E, Stein S, Thompson J (2008) Powered by VCL—using Virtual Computing Laboratory (VCL) Technology to Power Cloud Computing. In: Proceedings of the 2nd international conference on the virtual computing initiative (ICVCI'08)
2. Foster I, Zhao Y, Raicu I, Lu S (2008) Cloud computing and grid computing 360-degree compared. In: Proceedings of the grid computing environments workshop (GCE'08), Austin, Texas, USA
3. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M (2009) Above the clouds: a Berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, University of California at Berkeley
4. Amazon Web Services (2009) AF83 Case Study. http://aws.amazon.com/solutions/case-studies/af83/
5. Dong B, Zheng Q, Yang J, Li H, Qiao M (2009) An E-learning ecosystem based on cloud computing infrastructure. In: Ninth IEEE international conference on advanced learning technologies, ICALT 2009, pp 125–127
6. Dong B, Zheng Q, Qiao M, Shu J, Yang J (2009) BlueSky cloud framework: an e-learning framework embracing cloud computing. In: Proceedings of the 1st international conference on cloud computing (CloudCom 2009), Beijing, China
7. Cayirci E, Rong C, Huiskamp W, Verkoelen C (2009) Snow leopard cloud: a multi-national education training and experimentation cloud and its security challenges. In: Proceedings of the 1st international conference on cloud computing (CloudCom 2009), Beijing, China
8. Rochwerger B, Breitgand D, Levy E, Galis A, Nagin K, Llorente I, Montero R, Wolfsthal Y, Elmroth E, Caceres J, Ben-Yehuda M, Emmerich W, Galan F (2009) The RESERVOIR model and architecture for open federated cloud computing. IBM Syst J **53**(4):4:1–4:11
9. Google App Engine (2010) http://code.google.com/appengine/
10. Qumranet (2006) KVM—White Paper. http://www.linux-kvm.org/
11. Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebauer R, Pratt I, Warfield A (2003) Xen and the art of virtualization. In: Proceedings of the 19th ACM symposium on operating systems principles (SOSP'03), New York, USA
12. Nagios (2010) The industry standard in IT infrastructure monitoring. http://www.nagios.org/
13. Shibboleth (2010) http://shibboleth.internet2.edu/
14. TimeEdit (2010) http://www.timeedit.com/
15. FAI (2010) Fully automated installation. http://www.informatik.uni-koeln.de/fai/
16. Libvirt (2010) The virtualization API. http://libvirt.org/
17. NagiosGrapher (2010) https://www.nagiosforge.org/gf/project/nagiosgrapher/
18. OLAT (2010) Online learning and training. http://www.olat.org/
19. Zeilenga K (ed) (2006) Lightweight directory access protocol (LDAP): technical specification road map (RFC4510). http://tools.ietf.org/html/rfc4510
20. DFN-AAI (2010) Deutsches Forschungsnetz Federation. https://www.aai.dfn.de/
21. JMeter (2010) Apache JMeter. http://jakarta.apache.org/jmeter/