

Private Cloud for Collaboration and e-Learning Services: from IaaS to SaaS

Frank Doelitzscher¹, Anthony Sulistio¹,
Christoph Reich², Hendrik Kuijs² and David
Wolf¹

the date of receipt and acceptance should be inserted later

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.

Keywords cloud computing, dynamic VM creation, Shibboleth, PaaS, e-Learning

1 Introduction

Although Cloud Computing is a popular trend, it is difficult to get a clear definition of it. Ian et al. [1] discuss the basic concepts of Cloud Computing and show the differences compared to Grid Computing. 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 [2]. In addition, Armbrust et al. [3] give a good overview of Cloud Computing by highlighting obstacles and proposing several opportunities to solve them.

1: Department of Computer Science
Hochschule Furtwangen University, Germany
E-mail: {frank.doelitzscher, anthony.sulistio, david.wolf}@hs-furtwangen.de

2: Information and Media Centre
Hochschule Furtwangen University, Germany
E-mail: {christoph.reich, hendrik.kuijs}@hs-furtwangen.de

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 of IT infrastructure and training personnel. Thus, SMEs can take advantage of using a cloud when the IT capacity needs to be increased 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, uses 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 avoids purchasing new hardware for this special event, and delivers this successful event in a short amount of time.

For companies with large IT infrastructure, such as Amazon and Google, becoming a cloud provider allow them to offer their resources to SMEs based on pay-as-you-go and subscription models, respectively. Because not all services need the full resources at the same time for a long period 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 demands mainly towards the end of a semester. Moreover, Cloud Computing can also be used to support e-Learning services [5] [6] [2]. With the aforementioned motivations and scenarios, the Hochschule Furtwangen University (HFU) acknowledges the potential benefits of Cloud Computing. As a result, HFU establishes 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 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.

The rest of this paper is organized as follows. Section 2 provides some related work, whereas Section 3 explains the CloudIA infrastructure. Section 4 shows the potential usage of the CloudIA project by providing IaaS to HFU students and staff. Section 5 and Section 6 present PaaS and SaaS use case scenarios in HFU, respectively. Finally, Section 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 in [5]. The CloudIA offerings, presented in this paper, demonstrate the feasibility of cloud computing for e-Learning services. In addition, the CloudIA ar-

chitecture (to be discussed in Section 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, e.g. Amazon.

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, unlike the CloudIA architecture. Security plays a major role in ensuring user access policies, and enabling a trust of federation among education providers and users, as mentioned in Section 6.2. In addition, the BlueSky framework only focuses on delivering IaaS, whereas the CloudIA project offer IaaS, PaaS and SaaS.

Virtual Computing Laboratory (VCL) [2], 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 VCL manages also 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. Furthermore, VCL does not offer collaboration features, like CollabSoft (see Section 6). VCL offers (IaaS and PaaS) platforms which could then be used to host collaboration systems (SaaS) on top of it.

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 is used to run web 2.0 applications, such as video teleconferencing, voice over IP, and remote management, over handheld devices and terminals. As Snow Leopard Cloud is targeted towards military usage, it has a 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, where users login once to access various services.

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 Section 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.

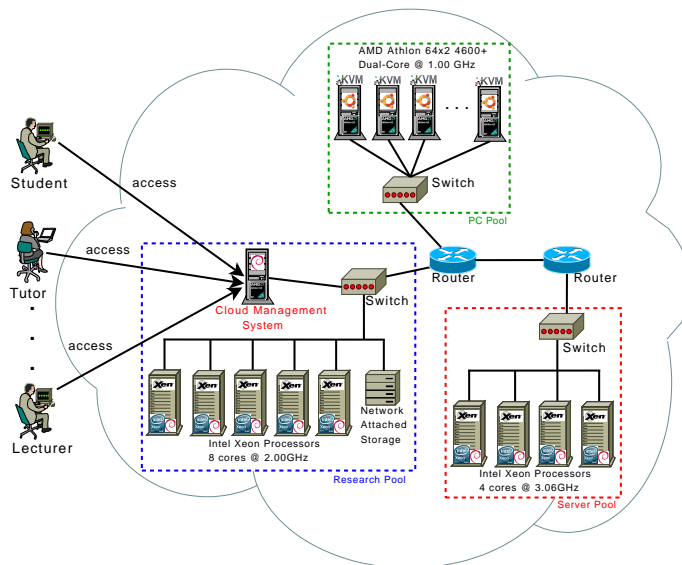


Fig. 1 Overview of the HFU's private cloud architecture CloudIA.

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 Section 5) and on-demand Collaboration Software (CollabSoft) (see Section 6).

The CloudIA project is a joint collaboration between IT (IMZ) and Computer Science (CS) departments of 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 Figure 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 as followed:

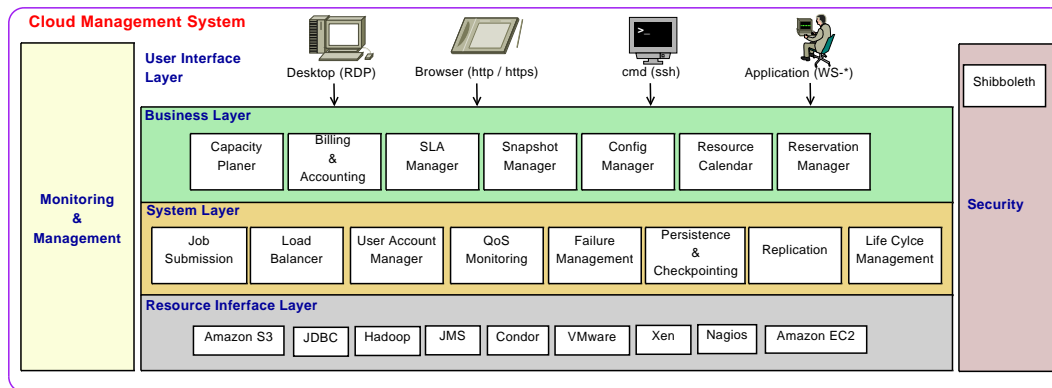


Fig. 2 Cloud Management System of CloudIA.

- **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 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].
- **Monitoring & Management Component:** To ensure the reliability of each layer in the cloud, a monitoring and management component is needed. Thus, this component allows 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 is also addressing 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 Section 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 Section 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, Shibboleth plays a critical role in integrating these cloud services to existing IT infrastructure seamlessly.

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

Shibboleth can be used for accessing the CloudIA platform. The example (Figure 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 Service Provider 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 precreated certificates for possible Service Providers.

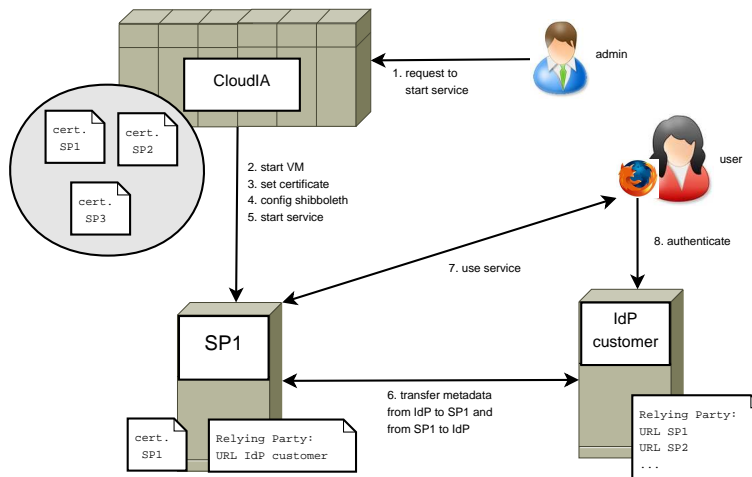


Fig. 3 CloudIA and SSO

Figure 3 explains the steps until 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 Service Provider certificate is copied to the VM (e.g. certificate SP1), (4) the Shibboleth configuration is processed (e.g. URL of the customers Identity Provider 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 Identity Provider and the Service Provider. Please notice, that the URL of the Service Provider must be known by the Identity Provider as "relying

party”. (7) user calls service the first time and (8) the user must authenticate at his/her Identity Provider, 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 followed:

- **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 subnets. We found that this is obstructive for 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, i.e. slowing the application down. In addition, it consumes the whole physical resource. Thus, preventing other VMs running on the same resource. For example, 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` was constantly consumed the available resource. 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 duration of the practical semester and they need to be maintained and re-configured every six months. This section shows that by offering these PCs as VMs, the CloudIA project helps IMZ to improve productivity in managing various hardware and software requirements needed for the students’ projects.

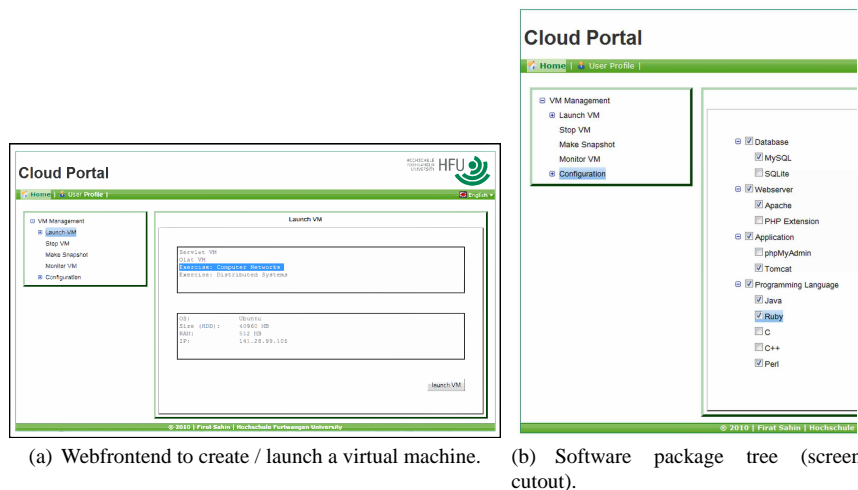


Fig. 4 CloudIA IaaS Webfrontend

4.1 VM Creation

Instead of providing bare metal machines (e.g. server and desktop PCs), IMZ can provide VMs. For example, networking exercises like routing labs or firewall configuration, each group needs multiple PCs to complete the exercise. Thus, VMs are suitable for this purpose. However, VMs need to be created, managed and monitored automatically. Thus a web front-end (see Figure 4(a)) is developed, where students of the aforementioned faculties can login and create, suspend, or delete own VMs using a wizard, according to their project demands. Per default, a maximum of 3 VMs per student with a total CPU time of 100 hours per semester, and a maximum of 1GB RAM for each VM is set. Unique compared to other IaaS providers is the creation and monitoring of VMs in CloudIA. Despite 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 add 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 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 in chronological order: The user who wants to create a new VM need to access the CloudIA's IaaS component through a web front-end (1), as shown in Figure 4(a). Login is handled by Shibboleth (see Section 3.1), which authenticates against HFU's identity system, i.e. OpenLDAP. The web front-end provides a wizard to create new VMs. Common (virtual) machine's

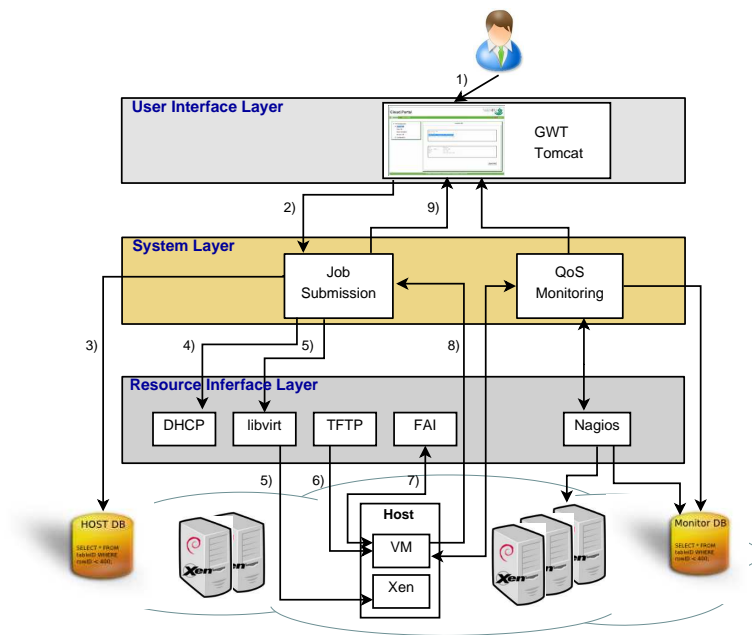


Fig. 5 Automatic VM creation architecture.

parameters like CPU, RAM size and hard disk size are asked before the user can choose software packages from a software tree (Figure 4(b)), 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 new created VM boots with network boot protocol (PXE) enabled. A TFTP network share on the FAI server offers a Linux kernel which gets transferred to the VM (6). The VM boots the kernel without using its hard disk. Hardware detection 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 save time and network bandwidth, the software packages are downloaded from the university's local 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 from its local file system and is ready for the user. The Job Submission module informs the web front end that the VM is available (9).

The details of the software-configuration are stored in the configuration files on the install server. Configuration files are shared among groups (classes) of computers which are similar. Therefore FAI does not care whether 10 or 100 hosts need to be installed, aside from performance and latency effects. [17] FAI creates a configuration file for every single VM unnecessary. Since an updated mirror of software is used, new VMs are always at actual patch level. Hence, FAI is a scalable method to install an infrastructure offered as a service with a great number of VMs.

4.2 Advantages for the IMZ

Before using this approach, at the beginning of each semester project, PCs are needed to be assigned among project groups. 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 different OS.

An internal study shows that for each used project PC, this process takes up 15 minutes 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 hours work for an administrator for a complete semester. Since eligible students are able to create their own VMs, no 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. This 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.

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 developing 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:

- 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 (see Section 4.2.1).

4.2.1 Challenges and Monitoring

A resulting challenge of using VMs is that the infrastructure seems to be infinitely available. While giving out physical hardware was considered very carefully, new virtual machines are deployed less deliberately. This 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 to performance characteristics of individual applications during runtime. Such characteristics can be CPU load, memory usage and network traffic. By monitoring these QoS parameters, IMZ provides 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 or 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. Nagios-Grapher [18], 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 a semesters time. This allows a better decision about the deployability if additional VMs are required for a project.



Fig. 6 Automatic VM Creation FAI process.

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 are only getting 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 Figure 6 shows the different steps. Experiments show that an average VM installation time of 5 minutes can be achieved depending on the amount of software which needs to be installed additionally on the VM.

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, to deploy and to 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 Section 4, and configure them. For example, a VM image containing MySQL, Tomcat, PHP, and Apache web server is created. Since 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) [19], 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 hours with predefined packages and software for their courses. To access these VMs, students login to the web portal, as shown in Figure 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.
- Save and deletes 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 resource available instances can be outsourced to a public cloud, i.e. using Amazon's small EC2 instances, as shown in Figure 7. Currently, internal IP and dynamic Amazon's 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 dynamic Amazon's 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 being 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' works. 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,

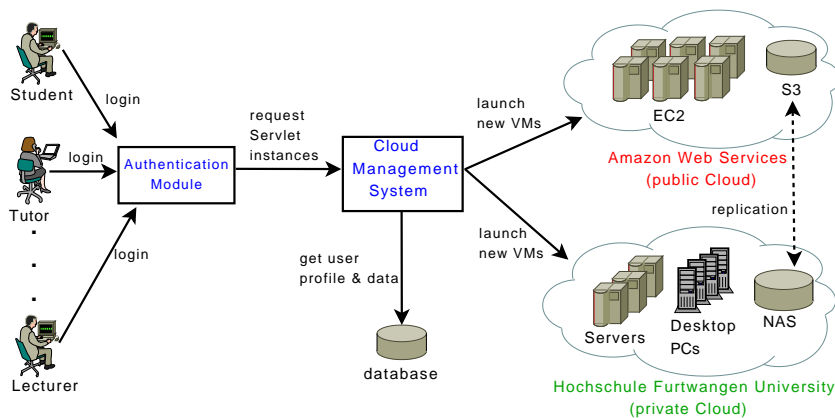


Fig. 7 High level overview of running the servlet container in the cloud.

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 environment. 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.

Finally, since SCP has been included into OLAT (see section 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. In addition, IMZ can charge VMs running on Amazon to the appropriate faculties. As a result, IT costs on hardware purchase, hardware maintenance, and administration can be reduced.

6 Software as a Service: On-demand Collaboration Software (CollabSoft)

For e-Learning and lecture support, IMZ at HFU provides an OLAT (Online Learning And Training). It 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) by the user themselves. Each user can create 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 adopted 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 one month - consider that HFU has 4100 active students in total, distributed over three campuses.

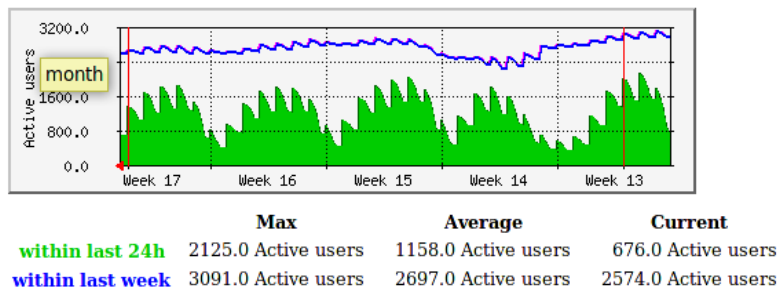


Fig. 8 Active OLAT user HFU (monthly)

6.1 The need for OLAT on-demand

Acknowledging the possibilities of the OLAT system, several faculties at the HFU start developing special demand 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 for the assessment 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, an own OLAT instance needs to be provided due to the same cheating-prevention 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 HFU-wide OLAT would imply to create 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 for a special event, existing only for 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 [20] based user directory and especially adoption of HFU's corporate design. Therefore, the on-demand Collaboration Software (CollabSoft), developed by the IMZ, providing 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 user-suited for small quantities. The OLAT administrator uses an administrator GUI to create/delete individual accounts.

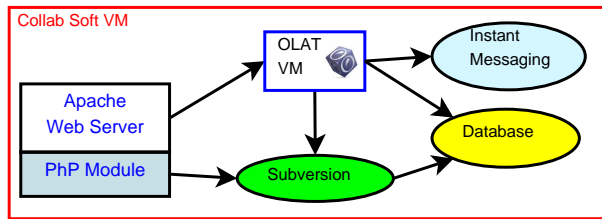


Fig. 9 Collaboration Software running inside a Virtual Machine

- *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 LDAP can access the instance.
- *Shibboleth*: If an institution is member of a federation, such as DFN-AAI [21], 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 be made to the Identity Provider (IdP), the home organization. For a more detailed description see Section 3.1.

6.2 Advantages of CollabSoft

With the CollabSoft VM image, the IMZ is now able to setup the 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).

HFU faculties are using CollabSoft as general project management software. With CollabSoft they are using a tool which they are used 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 can be created within a couple of hours without importing user-data from remote member-directory-services. Figure 10 shows a map of all DFN-AAI members who are using the wiki.



Fig. 10 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 administration by the user.
- No administration of single users needed if using Shibboleth. Only configuration of access level by attributes and rules.
- No additional user management necessary if used by HFU organizations due to preconfigured connection to LDAP directory.
- Persistent storage of instance possible due to snapshots of a CollabSoft VM.

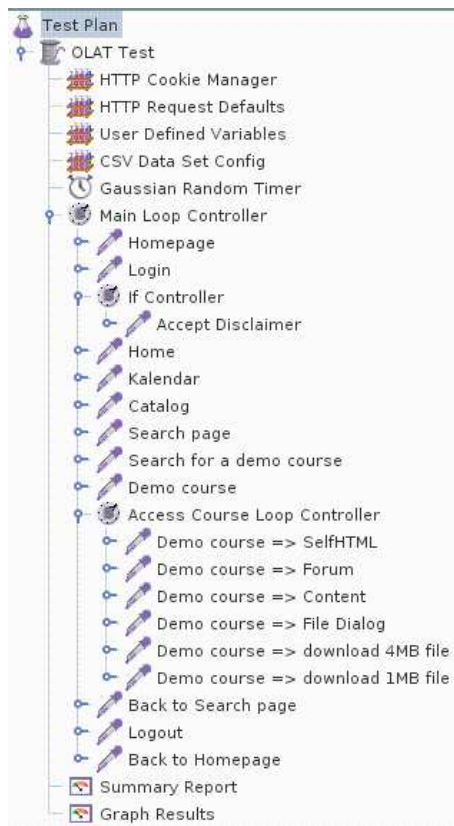


Fig. 11 JMeter screen shot of the Test Plan

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
t10	in demo course using forum
t11	in demo course using content page
t12	in demo course using file dialog
t13	in demo course download 4MByte file
t14	in demo course download 1MByte file
t15	search for demo course
t16	logout
t17	go to start page (dmz)

Fig. 12 JMeter Test Plan with Index

6.3 CollabSoft Performance Evaluation

To evaluate the performance of CollabSoft in the HFU's private cloud and its deployability to a public cloud, a performance evaluation was done. A CollabSoft virtual machine is deployed on a single cloud host in CloudIA. The same one is 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.

	CloudIA instance	Amazon	
		large	extra large
no. of cores	8	2	4
CPU type (Intel(R) Xeon(R))	E5504	E5430	E5430
CPU Freq.	2.00GHz	2.66GHz	2.66GHz
total Mem (kB)	12330508	7872040	15736360

Table 1 Hardware Configuration for the CollabSoft Experiment.

After deployment, each CollabSoft instance has been stressed with a customized benchmark using Apache JMeter [22]. JMeter is designed to interactively create test plans for load test. The CollabSoft test plan pictured in figure 11 and figure 12 has been 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 1000 times with about 100 users.

The most important measurement for the user is the request/response time during the test. Figure 13 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.

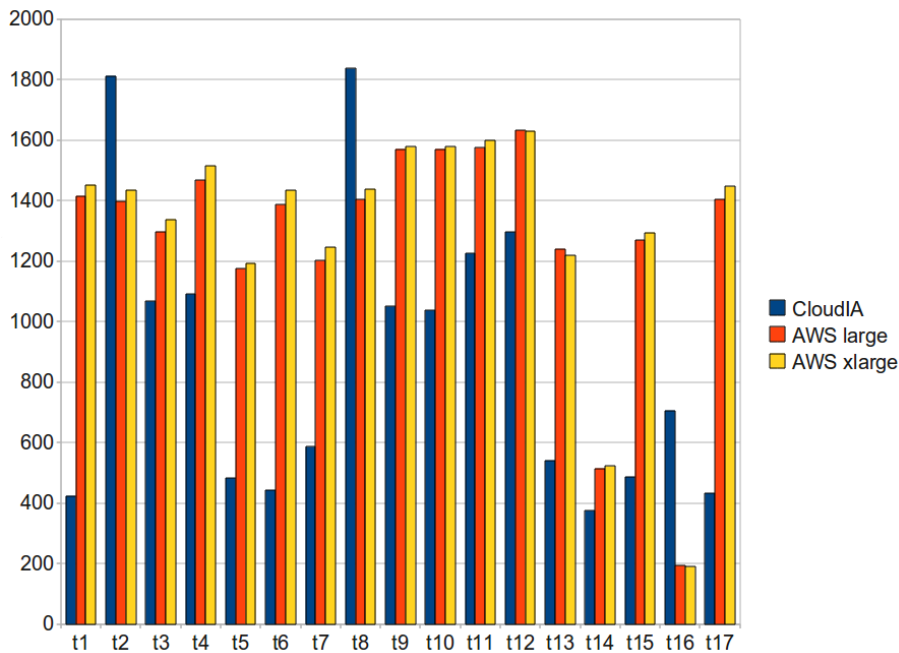


Fig. 13 ReqResTimeAll

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 EC2.

7 Conclusion and Future Work

The paper shows how Cloud Computing helps HFU’s IT department to improve periodical IT tasks like the management of project PCs or lecturer with a standard development environment for programming exercises. The private cloud computing ar-

chitecture *CloudIA* was explained and its special issue of Single-Sign-On using Shibboleth has been emphasized. Unique compared to other IaaS frameworks CloudIA is creating VM images 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 PaaS approach is supported by a Servlet Container Platform and mainly used for study courses at the HFU. It has been integrated into HFU's e-Learning management system with the support of Single-Sign-On. A very successful collaboration application CollabSoft has been adapted to SaaS with the support for different authentication technologies. Due to its flexibility it is now possible to support demand for a collaboration platform with an on-demand solution which HFU affiliates are already familiar with.

As for future work, further extensions of CloudIA's IaaS systems are planned. First trials showed that users often choose a very similar configuration. Hence it will be more efficient to provide several templates and add individual software wishes to them. Furthermore integration of different user management systems are developed to support connection to existing external user directories like 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 regarding to the green-it paradigm.

Acknowledgements The authors would like to thank the following people: Jonas Ambrus, Markus Joos, Bastian Lorenz, Daniel Neidig, Christoph Raible, Roman Rietsche, Nikolas Schaetzle, Patrick Straub, Daniel Weng, Lars Heppler and Manuel Gawol for their work on the development of the automatic virtual machine creation environment. In addition, we would like to thank Joachim Saleyka for his efforts on the OLAT installation, and Patrick Brunner from the OLAT project at University of Zurich for providing us with a JMeter template. We also would like to thank Firat Sahin for the web portal screenshots. Finally we thank Amazon's Education Project for supplying us with research credits on Amazon EC2.

References

1. Foster, I., Zhao, Y., Raicu, I., Lu, S.: Cloud Computing and Grid Computing 360-Degree Compared. In: Proceedings of the Grid Computing Environments Workshop (GCE'08), Austin, Texas, USA (Nov. 16 2008)
2. Vouk, M., Averitt, S., Bugaev, M., Kurth, A., Peeler, A., Schaffer, H., Sills, E., Stein, S., Thompson, J.: "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). (May 16–17 2008)
3. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS-2009-28, Univ. of California at Berkeley (Feb. 2009)
4. Amazon Web Services: AF83 Case Study. <http://aws.amazon.com/solutions/case-studies/af83/> (May 2009)
5. Dong, B., Zheng, Q., Yang, J., Li, H., Qiao, M.: An E-learning Ecosystem Based on Cloud Computing Infrastructure. In: Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on. (July 2009) 125 –127
6. Dong, B., Zheng, Q., Qiao, M., Shu, J., Yang, J.: BlueSky Cloud Framework: An E-Learning Framework Embracing Cloud Computing. In: Proceedings of the 1st International Conference on Cloud Computing (CloudCom 2009), Beijing, China (Dec. 1–4 2009)

7. Cayirci, E., Rong, C., Huiskamp, W., Verkoelen, C.: 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 (Dec. 1–4 2009)
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.: The RESERVOIR Model and Architecture for Open Federated Cloud Computing. *IBM Systems Journal* (2009)
9. GoogleAppEngine: Google's PaaS infrastructure. <http://code.google.com/appengine/> (2010)
10. Qumranet: KVM: White Paper. <http://www.linux-kvm.org/> (2006)
11. 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 19th ACM Symposium on Operating Systems Principles (SOSP'03), New York, USA (Oct. 19–22 2003)
12. Nagios: Nagios: IT-infrastructure monitoring tool. <http://www.nagios.org/> (2010)
13. Shibboleth: A Single-Sign-On Solution. <http://shibboleth.internet2.edu/> (2010)
14. TimeEdit. <http://www.timeedit.com/> (May 2009)
15. Dipl. Inf. Thomas Lange: Fully Automated Installation (FAI). <http://www.informatik.uni-koeln.de/fai/> (2010)
16. Libvirt: virtualization API. <http://libvirt.org/>
17. Thomas Lange: Fai guide (fully automatic installation). Technical report, University of Colone (2010)
18. NagiosGrapher: Nagios graph system. <https://www.nagiosforge.org/gf/project/nagiosgrapher/> (2010)
19. OLAT: Online Learning And Training. <http://www.olat.org/> (2010)
20. LDAP: Lightweight Directory Access Protocol (RFC4510). <http://tools.ietf.org/html/rfc4510> (2010)
21. DFN-AAI: Deutsches Forschungsnetz federation. <https://www.aai.dfn.de/> (2010)
22. JMeter: Apache JMeter. <http://jakarta.apache.org/jmeter/> (2010)