

INSTITUT FÜR INFORMATIK
DER LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN



Diplomarbeit

Konzeption und Implementierung eines Webportals für das Management virtueller Maschinen

Jakob Lepiarczyk



Diplomarbeit

Konzeption und Implementierung eines Webportals für das Management virtueller Maschinen

Jakob Lepiarczyk

Aufgabensteller: Prof. Dr. Dieter Kranzlmüller
Betreuer: Dr. Nils Gentschen Felde
Markus Gillmeister
Dr. Tobias Lindinger
Christian Richter
Abgabetermin: 30. September 2011

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 30. September 2011

.....
(*Unterschrift des Kandidaten*)

Abstract

Der Dienstlebenszyklus einer individuell bereitgestellten virtuellen Maschine setzt sich aus mehreren Phasen, wie zum Beispiel der Beantragung, der Bereitstellung, dem Betrieb und der Kündigung, zusammen. Die vorhandenen Management-Anwendungen decken aber diesen nur unzureichend ab und konzentrieren sich in der Regel nur auf den Betrieb.

Diese Diplomarbeit nimmt sich dieses Defizites an. Mit Hilfe einer szenariobasierten Anforderungsanalyse und einer Nutzerbefragung wird ein Anforderungskatalog aufgestellt, auf dessen Grundlage ein Konzept für ein Webportal zum Management von virtuellen Maschinen entworfen und anschließend prototypisch implementiert wird. Dieses Webportal stellt für die Endnutzer eine einheitliche Oberfläche für die Einreichung von Anträgen und die Steuerung der virtuellen Maschinen bereit. Genauso erleichtert das Webportal den Administratoren die Verwaltung der virtuellen Infrastruktur, da es ihren gesamten Arbeitsfluss abdeckt und durch Automatisierung viele Fehler vermeidet.

Dabei verfügt das Webportal über einen generischen Aufbau, der es ermöglicht dieses mit jeder beliebigen Virtualisierungslösung einzusetzen. Ermöglicht wird dies durch eine Adapter-Komponente, für die im Rahmen der Arbeit eine generische REST-API entwickelt wird.

Zu den weiteren Eigenschaften des Webportals gehört eine hohe Skalierbarkeit und Ausfallsicherheit, sowie die Unterstützung von Ressourcepools. Das Webportal ist so konzipiert, dass es sich in nahezu jede IT-Infrastruktur fließend integrieren lässt. Dies wird unter anderem dadurch ermöglicht, dass das Webportal nicht auf einer eigenen Nutzerverwaltung aufsetzt, wie viele andere Lösungen, sondern die bereits vorhandene nutzt. Dadurch können parallel zum Webportal alle bisherigen Management-Anwendungen weiterbenutzt werden.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation	1
1.2. Vorgehensweise	2
2. Grundlagen	5
2.1. Vorteile und Nachteile der Virtualisierung	5
2.2. Techniken der Virtualisierung	6
2.3. Management virtueller Infrastrukturen	8
2.4. Die vSphere-Infrastruktur	9
2.5. Probleme mit dem vSphere-Management	11
3. Anforderungsanalyse	13
3.1. Szenariobasierte Anforderungsanalyse	13
3.1.1. Szenario 1 - Leibniz-Rechenzentrum	13
3.1.1.1. Beschreibung Variante A	14
3.1.1.2. Beschreibung Variante B	14
3.1.1.3. Abgeleitete Anwendungsfälle	14
3.1.2. Szenario 2 - Institut für Informatik	21
3.1.2.1. Beschreibung Variante A	21
3.1.2.2. Beschreibung Variante B	21
3.1.2.3. Abgeleitete Anwendungsfälle	22
3.1.3. An Hand der szenariobasierten Analyse ermittelte Anforderungen	24
3.2. Nutzerbefragung	26
3.2.1. Erstellung des Fragebogens	27
3.2.2. Durchführung	27
3.2.3. Ermittelte Daten	27
3.2.4. Durch die Nutzerbefragung ermittelte Anforderungen	29
3.3. Zusammenfassung der funktionalen Anforderungen	30
3.4. Nicht-funktionale Anforderungen	32
3.5. Anforderungskatalog	32
4. State of the Art	35
4.1. Einschub: libvirt	35
4.2. Überblick über bestehende Managementlösungen	36
4.3. Cloudmin	39
4.4. Abiquo	40
4.5. openQRM	40
4.6. Bewertung der vorgestellten Anwendungen im Kontext dieser Arbeit	41

5. Konzept	43
5.1. Komponenten der Gesamtarchitektur	43
5.2. Schnittstellen zwischen den Komponenten	47
5.3. Das Portal	51
5.3.1. Informationsmodell des Webportals	51
5.3.1.1. User	53
5.3.1.2. TAC	55
5.3.1.3. Request	55
5.3.1.4. Configuration	56
5.3.1.5. HDD	58
5.3.1.6. NIC	58
5.3.1.7. VM	58
5.3.1.8. RP	60
5.3.2. Portal-API(s)	60
5.4. Der Adapter	62
5.5. Adapter-API	63
5.5.1. Rückgabewerte	63
5.5.2. Requests	65
5.6. Prüfung der Anforderungen	77
6. Implementierung	81
6.1. Umfang der Implementierung	81
6.2. Verwendete Frameworks und Bibliotheken	82
6.2.1. Für das Portal verwendete Frameworks und Bibliotheken	82
6.2.2. Für den Adapter verwendete Frameworks und Bibliotheken	82
6.3. Implementierungsdetails	83
6.3.1. Realisierung der Konfigurationsvorlagen	85
6.3.2. Realisierung der Konsole	85
6.4. Installation und Konfiguration	86
6.4.1. Konfiguration des vCenters	86
6.4.2. Installation und Konfiguration des Portals	87
6.4.3. Installation und Konfiguration des Adapters	88
6.4.4. Installation und Konfiguration eines DHCP-Servers	89
6.5. Bedienkonzept des Webportals	89
6.6. Sicherheitskonzept / Erreichbarkeit der Komponenten	90
6.7. Prüfung der Anforderungen	95
7. Fazit	99
7.1. Zusammenfassung	99
7.2. Ausblick	100
A. Anhang	103
Abbildungsverzeichnis	109
Literaturverzeichnis	111

1. Einleitung

Schon seit einigen Jahren setzen sich im Hosting-Bereich Lösungen mit virtueller Infrastruktur immer weiter durch. Zu den Vorteilen gehören eine höhere Flexibilität, sowie eine bessere, gleichmäßigere Auslastung der physischen Hardware, wodurch wiederum ein günstigerer Betrieb möglich ist. Am Markt haben sich bereits viele Anbieter etabliert, die entsprechende Dienste zur Verfügung stellen. Die meisten bieten Pakete mit festgelegten Konfigurationen an. Der Kunde hat dann zum Beispiel vier verschiedene Konfigurationen zur Auswahl und muss gegebenenfalls Kompromisse eingehen. Möchte er zum Beispiel viel Arbeitsspeicher, so muss er unter Umständen ein Paket wählen bei dem zusätzlich eine große Festplatte angeboten wird, obwohl er diese nicht benötigt. Auf der anderen Seite gibt es ebenfalls Anbieter, unter anderem im akademischen Bereich, die frei konfigurierbare virtuelle Maschinen anbieten. Oft kann der Kunde hierbei — ohne weitere Vorgaben — seine Wunschkonfiguration beantragen. Die Bereitstellung geschieht in diesem Fall allerdings nicht automatisch, wie dies oft bei den paketbasierten Angeboten der Fall ist, sondern wird von einem Administrator angestoßen, welcher die Konfiguration des Kunden auf Realisierbarkeit prüft.

Diese Diplomarbeit betrachtet die zweite Sorte von Anbietern. Für solche Anbieter soll ein Webportal entwickelt werden, welches den gesamten Dienstlebenszyklus, also beginnend bei der Beantragung einer virtuellen Maschine und bei deren Löschung und Abrechnung endend, abdeckt.

1.1. Motivation

Betrachtet man die existierende Software-Landschaft wird man feststellen, dass bisher keine Management-Software verfügbar ist, die solche individuell ausgestatteten virtuellen Maschinen zufriedenstellend unterstützt. Zwar ist bei jeder Virtualisierungslösung auch irgendeine Managementlösung integriert, doch geht deren Funktionalität nicht über die Steuerung des Betriebs hinaus. Die Lösung, die im Rahmen dieser Arbeit entwickelt wird, soll diesen Teil des Dienstlebenszyklus zwar ebenfalls abdecken, aber darüber hinaus auch eine Schnittstelle für die Interaktion zwischen Administratoren und Endnutzern beziehungsweise Kunden bilden. Die auf diese Weise ermöglichte Kommunikation zwischen Administratoren und Endkunden ist essentiell für die Unterstützung der Teile des Dienstlebenszyklus, die über den reinen Betrieb hinausgehen. Der Aufgabenbereich des Webportals beginnt also bei der Beantragung von virtuellen Maschinen durch einen Kunden, erstreckt sich über den Genehmigungs- und Bereitstellungsvorgang und den Betrieb, inklusive möglicher Modifikationswünsche, bis hin zur Löschung und Abrechnung. Das Webportal soll also alle erforderlichen Handlungen, die ein Administrator oder Endnutzer über diesen Zeitraum ausführen muss, unter einer einheitlichen Oberfläche vereinen und so den Arbeitsablauf flüssiger und weniger fehleranfällig gestalten. Einige Unternehmen, die solche VMs anbieten, verfügen zwar über eine solche Software, doch sind dies proprietäre Systeme, die jeweils für den eigenen Bedarf entwickelt wurden, die der Abgrenzung von der Konkurrenz dienen und die anderen nicht zur Verfügung gestellt werden.

1. Einleitung

Doch dies ist nicht die einzige Motivation für die Entwicklung des Webportals. Neben der Abdeckung des gesamten Dienstlebenszykluses, soll das Webportal auch diverse Sicherheitsprobleme, die bei den bisherigen Managementlösungen bestehen, lösen. Zum Beispiel verfügen viele Virtualisierungslösungen über eine zentrale Steuerkomponente, auf welche alle Administratoren und Endkunden im Rahmen des Managements zugreifen. Doch wie immer bei zentral organisierten Infrastrukturen bildet das zentrale Element eine Schwachstelle. Es ist ein Leichtes diese zentrale Komponente anzugreifen und somit das Management der gesamten Infrastruktur außer Gefecht zu setzen. Das zu entwickelnde Webportal soll eine Lösung für diese und für weitere Schwachstellen bisheriger Managementlösungen anbieten.

Dabei soll das Webportal generisch sein, es soll sich also bei jedem Anbieter problemlos und maximal mit geringen Anpassungen installieren lassen. Das bedeutet auch, dass es unabhängig von einer bestimmten eingesetzten Virtualisierungslösung sein muss und nicht von einer konkreten Ausprägung einer bereits existierenden Management-Infrastruktur abhängig sein darf.

1.2. Vorgehensweise

Um die genannten Ziele zu erreichen, wird im Rahmen dieser Arbeit mit dem Leibniz-Rechenzentrum (LRZ) in Garching bei München zusammengearbeitet. Es handelt sich dabei um einen Anbieter der vorhin beschriebenen Sorte. Das LRZ bietet seine Dienste primär den Münchner Hochschulen an, welche die virtuellen Maschinen für Lehre und Forschung einsetzen. Gerade im Bereich der Forschung werden sehr individuelle Ansprüche an die VMs gestellt, so dass im voraus definierte Konfigurationen nicht akzeptabel wären. Als zweiter Anbieter, damit keine zu starke Orientierung an den Bedürfnissen einer Organisation entsteht, wird das Institut für Informatik an der LMU herangezogen. Dieses betreibt eine eigene virtuelle Infrastruktur, um VMs für seine Mitarbeiter und Studenten anzubieten.

Im Einzelnen soll bei dieser Arbeit wie im Folgenden beschrieben vorgegangen werden.

Zunächst müssen die genauen Anforderungen, die an ein solches System gestellt werden, ermittelt werden. Zu diesem Zwecke müssen zunächst Gespräche mit Administratoren am LRZ geführt werden, die darlegen müssen, wie die IT-Infrastruktur am LRZ aufgebaut ist und wo Zwangspunkte sind an denen sich die zu entwickelnde Lösung orientieren muss. Damit sich die Lösung auch funktional an den Bedürfnissen der zukünftigen Nutzer orientiert, wird eine Bedarfsanalyse durchgeführt. Dies geschieht einerseits szenariobasiert und andererseits mit Hilfe eines Fragebogens, mit dessen Hilfe sowohl die Administratoren als auch die Endnutzer der genannten Institutionen befragt werden. An Hand der gewonnenen Informationen werden Anwendungsfälle beschrieben und Anforderungen abgeleitet, welche das Webportal abdecken muss. Sobald dieser Anforderungskatalog erstellt ist, wird eine State of the Art Analyse durchgeführt. Zunächst wird recherchiert, welche Software es bereits am Markt gibt. Anschließend wird durch eine Prüfung, in wie weit diese den Anforderungskatalog erfüllt, festgestellt, ob sie bereits für den angedachten Einsatzzweck geeignet ist oder ob sich eine Modifikation dieser statt einer kompletten Neuentwicklung anbietet. Sobald feststeht, dass keine sich am Markt befindende Software den Anforderungen gewachsen ist, wird auf Grundlage des Katalogs ein Konzept erarbeitet, welches beschreibt, wie ein solches Webportal aufgebaut sein muss und wie es in eine bestehende Infrastruktur integriert werden kann. Um die Umsetzbarkeit des Konzepts zu demonstrieren wird zum Schluss eine prototypische Implementierung angefertigt, die am LRZ eingesetzt werden soll.

Das LRZ ist für eine solche Implementierung besonders geeignet. Einerseits verwaltet das LRZ seine virtuelle Infrastruktur derzeit über mehrere nicht integrierte Systeme. Die Management-Software der eingesetzten Virtualisierungslösung, bei welcher es sich um VMwares vSphere handelt, unterstützt nur den Betrieb. Bestellungen von VMs finden zum Beispiel per E-Mail statt, was einen entsprechenden Mehraufwand und eine hohe Fehleranfälligkeit bedeutet. Andererseits wird die webbasierte Management-Komponente von vSphere, WebAccess, nicht mehr weiterentwickelt, so dass an dieser Stelle sowieso Ersatz erforderlich ist. Sicherheitsprobleme wie sie bereits kurz angerissen wurden, sind auch in diesem Fall existent. Die Implementierung des Webportals wird an diesem praktischen Beispiel aufzeigen, dass das entwickelte Konzept den realen Bedingungen gewachsen ist. Dabei wird das Webportal nicht nur das Management der vSphere-Infrastruktur übernehmen müssen, sondern sich auch in die bisherigen LRZ-weiten Management-Systeme integrieren lassen müssen. Da das Konzept eine generische Lösung darstellt, sollte sich die Implementierung für das LRZ anschließend auch einfach an die Infrastruktur an der LMU anpassen lassen.

Das Ergebnis dieser Arbeit soll also erstens ein Konzept für ein Webportal zum Management virtueller Maschinen sein und zweitens eine prototypische Implementierung für das LRZ, welche die Tragfähigkeit des Konzepts unter Beweis stellt.

1. *Einleitung*

2. Grundlagen

Bevor mit dem in der Einleitung beschriebenen Vorgehen begonnen wird, sollen zunächst einige Grundlagen geklärt werden. Zuerst erfolgt eine Einführung in die Hostvirtualisierung, andere Virtualisierungsformen spielen für diese Arbeit keine Rolle. Dabei werden zuerst die Vor- und Nachteile der Virtualisierung beschrieben und diverse Techniken vorgestellt. Anschließend erfolgt eine Einführung in das Management von virtuellen Infrastrukturen und den damit verbundenen Problemen. Im darauf folgenden Unterkapitel wird VMwares vSphere genauer beschrieben. Die Gründe hierfür sind erstens, dass es sich dabei um einen weit verbreiteten Virtualisierer handelt, der gut geeignet ist die bis dahin erfolgten theoretischen Ausführungen zu veranschaulichen und zweitens dies die Virtualisierungslösung ist, auf der die prototypische Implementierung aufbauen wird. Um also die Funktionsweise der Implementierung zu verstehen, ist auch ein grundlegendes Verständnis über die Funktionsweise von vSphere erforderlich. Zum Schluß wird schließlich genauer auf die Probleme eingegangen, die sich am LRZ beim Einsatz von vSphere ergeben. Wichtige Fachbegriffe, welche für die weitere Arbeit von Bedeutung sind, werden im Folgenden hervorgehoben dargestellt.

2.1. Vorteile und Nachteile der Virtualisierung

Die Virtualisierung hat sich in letzter Zeit gerade im Hosting-Bereich sehr stark durchgesetzt. Ein Großteil der Provider, die früher lediglich physische Server (Root-Server) angeboten haben, bieten mittlerweile auch virtuelle Server, meist vServer genannt, an. Die Vorteile der Virtualisierung sind vielfältig und liegen sowohl auf Anbieter als auch auf Abnehmerseite. Im Folgenden sollen diese dargestellt werden.

Der Betrieb von virtueller Infrastruktur bedeutet eine bessere Auslastung der physischen Hardware. Ein gewöhnlicher Root-Server ist die meiste Zeit des Tages untätig, die regelmäßig zu erledigenden Aufgaben, wie der Empfang von E-Mails oder die Auslieferung von Webseiten erzeugt nur eine geringe Grundlast. Dennoch muss der Server eine ausreichende Reserve an Ressourcen haben, um auf gelegentlich auftretende Lastspitzen zu reagieren. Laufen mehrere virtuelle Server auf einer physischen Hardware ist die Grundlast bereits höher. Da die Lastspitzen meistens nicht alle virtuellen Server gleichzeitig betreffen, können die Ressourcen dynamisch der virtuellen Maschine zugeordnet werden, die sie gerade benötigt. Dabei ist die dynamische Zuordnung natürlich nicht zwingend, auch eine statische Zuordnung der Ressourcen ist möglich, wenn der Anwendungsfall dies erfordert. Durch die gemeinsame Nutzung von physischer Hardware und deren bessere Auslastung kann der Anbieter virtuelle Server deutlich günstiger anbieten als physische. Zumal es so auch möglich ist sehr kleine virtuelle Server anzubieten, so dass man zum Beispiel einen gering ausgelasteten E-Mail-Server in einer eigenen abgeschotteten Umgebung betreiben kann, wofür sich der Einsatz eines physischen Rechners nicht gelohnt hätte. Durch eine solche Isolation von Diensten kann wiederum eine höhere Sicherheit gewährleistet werden. Das Aufsetzen von VMs ist einfach und schnell möglich, dies kann sogar binnen weniger Sekunden automatisch erfolgen und die Einrichtung, wie auch der Wechsel von Betriebssystemen kann durch den Einsatz von Templates

2. Grundlagen

ebenso ohne große Verzögerung erfolgen. Des Weiteren können die Ressourcen meist schnell und ohne Neustart erweitert werden. Den Nutzern können beim Einsatz von VMs diverse Funktionen angeboten werden, die bei Nutzung von physischen Servern nur schwierig und nur durch den Einsatz von teurer Spezialhardware möglich sind. So kann der Kunde über den Hypervisor Zugriff auf die Konsole einer VM und somit auch Zugriff auf das BIOS bekommen. Möchte man so etwas bei einem physischen Server realisieren, müsste man auf eine teure KVM-over-IP-Lösung zurückgreifen. Der Kunde kann bei Problemen seinen Server auch jederzeit in den Ursprungszustand versetzen. Ebenfalls wird oft eine Rescue-Shell angeboten, dabei wird ein zweiter temporärer vServer dynamisch erzeugt und das Dateisystem des (defekten) ursprünglichen vServers unter dem neuen vServer gemountet. Auch High-Availability-Installationen lassen sich mit virtueller Infrastruktur kostengünstig realisieren. So kann eine Kopie eines Servers irgendwo bereitliegen und jederzeit gestartet werden. Dabei verursacht diese Kopie im ausgeschalteten Zustand beinahe keinerlei Kosten. Einige Virtualisierungslösungen bieten zudem auch eine Live-Migration von VMs zwischen Hosts an. So kann eine VM auf einen neuen Host migriert werden, sobald sich erste Probleme mit der physischen Hardware abzeichnen oder wenn die Ressourcen nicht mehr ausreichen.

Die Virtualisierung kann auch Probleme verursachen. So ist gerade bei dynamischer Ressourcenzuteilung die Performance einer einzelnen VM immer von der Auslastung der anderen VMs abhängig. Auch für Anwendungen mit sicherheitskritischen oder geheimen Daten sind virtuelle Server nur bedingt geeignet, besonders dann wenn man sich den Host mit anderen Anwendern teilt. So kann es zum Beispiel durchaus passieren, dass bei einem defekten RAID-Controller Dateien aus der einen VM in einer anderen VM auftauchen.

2.2. Techniken der Virtualisierung

Um eine Virtualisierung durchzuführen benötigt man eine **Virtualisierungslösung**. Dabei handelt es sich um ein aus mehreren Softwarekomponenten bestehendes Gesamtpaket, welches dazu dient virtuelle Maschinen zu erstellen und zu betreiben. Die **VM** ist eine Art Container, welcher dem Gast, also einem in der VM laufenden Betriebssystem die virtuelle Hardware zur Verfügung stellt. Wichtigster Bestandteil einer Virtualisierungslösung ist der **Hypervisor**, bei welchem es sich um den Virtualisierer im engeren Sinn handelt. Ergänzt wird dieser um diverse Software zum Management desselben. Der Hypervisor muss auf einem **Host** installiert werden, dabei handelt es sich um den physischen Rechner, welcher die Ressourcen für die virtuellen Maschinen bereitstellt. Der Hypervisor, welcher die Laufzeitumgebung für die virtuellen Maschinen bereitstellt, kann, wie in Abbildung 2.1 dargestellt, auf zwei Arten realisiert werden. Die erste Möglichkeit ist, dass er als Programm unter dem Betriebssystem läuft, welches auf dem Host installiert ist. Diese Variante wird als Hosted oder Type 2-Hypervisor bezeichnet. Die zweite Möglichkeit ist, dass der Hypervisor das Betriebssystem selbst realisiert, was einen ressourcenschonenderen und performanteren Betrieb ermöglicht. Hierbei spricht man von einem Bare-Metal- oder Type 1-Hypervisor. Bei einigen Virtualisierern ist es möglich die physische Hardware, auf welcher die Virtualisierung läuft an das Innere des VM-Containers weiter zu reichen. So kann der Gast zum Beispiel Zugriff auf ein physisch existierendes Laufwerk erhalten. Je nach Hypervisor kann die Größe der zuweisbaren Ressourcen direkt von der Hardware des Hosts abhängen oder nicht. So teilen einige Hypervisor den physischen RAM auf die einzelnen VMs auf, so dass den VMs zusammen

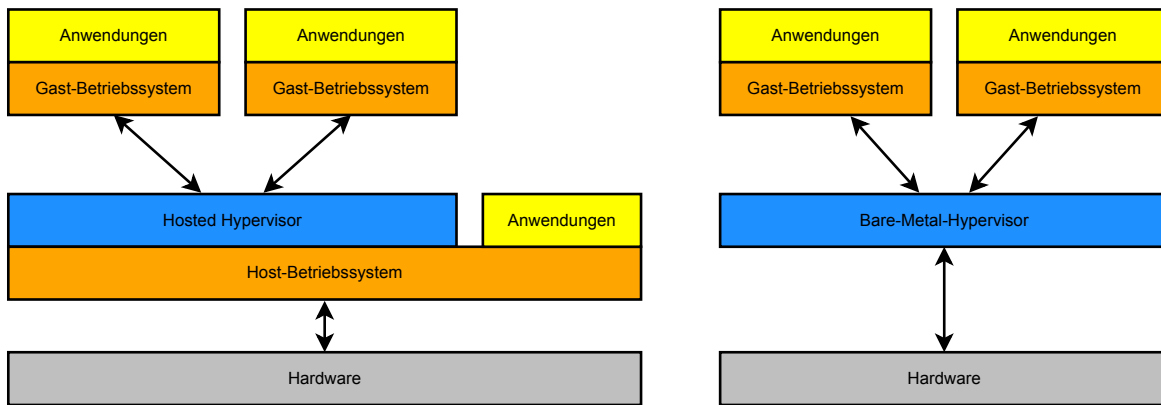


Abbildung 2.1.: Hosted- und Bare-Metal-Hypervisor

nicht mehr RAM zugewiesen werden kann als im Host verbaut. Andere Hypervisor haben, zum Beispiel dank der Möglichkeit Speicher auszulagern, solche Beschränkungen nicht.

Man kann dabei verschiedene Arten von Virtualisierung unterscheiden. Die folgende Kategorisierung basiert auf der Dissertation von Dr. Tobias Lindinger. [Lin09]

Der radikalste und zugleich flexibelste Ansatz ist die Emulation. Hierbei wird die gesamte Hardware eines Rechners in Software nachgebildet. Dadurch wird es möglich in der VM ein System laufen zu lassen, das für die darunterliegende physische Hardware nicht geeignet ist. Es können auch verschiedene Architekturen zur gleichen Zeit emuliert werden. Der große Nachteil dieser Lösung ist, dass die Umsetzung aller Hardwarezugriffe innerhalb der VM sehr rechenaufwendig ist, so dass die Emulation sehr langsam ist.

Ein schnellerer und auch relativ neuer Ansatz ist die Paravirtualisierung. Hierbei wird keine virtuelle Hardware erzeugt, sondern mehrere parallel laufende, speziell für diesen Einsatzzweck angepasste Betriebssysteme über eine Verwaltungsschicht voneinander getrennt. Die erforderlichen Eingriffe im Gastbetriebssystem können allerdings nur an quelloffenen System oder durch den Hersteller durchgeführt werden.

Ein weiterer Ansatz ist die Vollvirtualisierung. Diese ist eine Variante der Emulation, bei der jedoch nur Gastsysteme unterstützt werden, die auch auf der Hardware des Hosts laufen würden, wodurch der Übersetzungsaufwand reduziert und die Leistung somit gesteigert wird.

Zuletzt sollte noch die native Virtualisierung erwähnt werden. Hierbei handelt es sich hauptsächlich um eine Vollvirtualisierung, wobei jedoch für das Gastbetriebssystem spezielle angepasste Treiber verwendet werden, die einen performanteren Betrieb erlauben. Dies lässt sich zum Beispiel dadurch erreichen, dass die Anzahl der Kontextwechsel reduziert wird, welche vom Hypervisor abgefangen werden müssen, was das System ausbremst.

Die technische Grundlage auf der die Virtualisierungslösung aufbaut, die das in dieser Arbeit entwickelte Webportal steuern soll, spielt im Wesentlichen keine Rolle. Die Software könnte zum Beispiel mit folgenden Virtualisierungslösungen verwendet werden, die alle zu den bekannteren, weiter verbreiteten Lösungen zählen: Bochs (opensource), VirtualBox (Oracle), VirtualPC (Microsoft), Xen (Citrix, opensource), QEMU (opensource), OpenVZ (opensource), das auf OpenVZ aufbauende kommerzielle Virtuozzo (Parallels) und diverse Produkte von VMware zu denen VMware Server und vSphere gehören.

2.3. Management virtueller Infrastrukturen

Um den Hypervisor zu verwalten, beinhalten alle Virtualisierungslösungen irgendeine Form von Management-Software. Diese ist in der Regel produktspezifisch, so dass mit der Management-Software eines Herstellers auch nur der Virtualisierer dieses Herstellers gesteuert werden kann. Zu den grundlegenden Funktionen solcher Management-Anwendungen gehören die Möglichkeiten VMs zu erstellen, zu modifizieren, zu löschen und zu steuern, also diese zu starten und zu stoppen. Weiterführende Funktionen können die Rechteverwaltung, das Klonen von VMs, das Generieren von Templates und Erstellen von VMs aus diesen, das Anlegen von Snapshots oder die Einrichtung von hochverfügbaren Systemen sein. Bei diesen etwas anspruchsvolleren Funktionen stellt man allerdings schnell sehr große Unterschiede bei den einzelnen Produkten fest. Eine weitere sehr wichtige Funktion ist die (Remote-)**Konsole**. Hierbei handelt es sich um eine grafische Zugriffsmöglichkeit auf eine VM. Die Konsole ist im Prinzip vergleichbar mit RDP oder VNC, mit dem Unterschied, dass die Konsole durch den Hypervisor realisiert ist. Dadurch ist auch dann ein Zugriff ermöglicht, wenn kein Betriebssystem in der VM installiert ist oder sich dieses nicht starten lässt. Die Konsole ermöglicht somit auch einen Zugriff auf das BIOS der VM und bietet die Möglichkeit ein Betriebssystem zu installieren.

Betreibt man als großer Anbieter eine größere Anzahl an Hosts, auf welchen jeweils ein Hypervisor installiert ist, so wird das Management mit zunehmender Anzahl immer umständlicher. Deshalb beinhalten Virtualisierungslösungen, die darauf ausgelegt sind, eine weitere Abstraktionsebene. Zwischen die Hypervisoren und die Management-Anwendung wird eine neue Komponente eingesetzt, welche die zentrale Steuerung der gesamten Infrastruktur übernimmt. Der Administrator greift fortan mit seiner Management-Anwendung auf diese zentrale Steuerkomponente zu und erhält darüber einen Blick auf alle Hypervisoren. Dank dieser neuen Komponente können viele Management-Aufgaben vereinfacht oder sogar neue Funktionen eingeführt werden. So kann diese Komponente zum Beispiel die einzelnen Hosts kontinuierlich überwachen und bei Bedarf, etwa bei Überlastungen oder Hardwareproblemen, die sich auf diesem Host befindlichen VMs auf einen anderen Host migrieren.

In der Regel nutzen sowohl Administratoren als auch Endnutzer oder beliebige andere Nutzergruppen die gleiche Management-Anwendung. Deren unterschiedliche Berechtigungen stellen sicher, dass ein Administrator zum Beispiel VMs anlegen kann, ein Endnutzer diese aber nur starten und stoppen kann. Da also alle Nutzergruppen bereits eine gemeinsame Software nutzen, wäre diese die optimale Grundlage, um auch den verbleibenden Dienstlebenszyklus abzudecken. Ein Endnutzer sollte also zum Beispiel die Möglichkeit haben in dieser Software einen Antrag auf eine neue VM auszufüllen. Der Administrator sollte die Möglichkeit haben, die Angaben aus diesem Antrag direkt in eine VM-Konfiguration zu übernehmen. Doch leider weisen in der Hinsicht alle mitgelieferten Managementlösungen ein großes Defizit auf. Keine Lösung ermöglicht eine Interaktion zwischen den Benutzergruppen. Abbildung 2.2 veranschaulicht nochmals dieses Problem. Fairerweise muss man zugestehen, dass die Art, wie man eine solche Lösungen gestaltet, stark von der Umgebung abhängt, in der die Virtualisierungslösung eingesetzt wird, so dass es den Herstellern hier kaum möglich ist, eine für alle Anwendergruppen geeignete Lösung zu finden.

Viele Virtualisierungslösungen bieten neben der Steuerung über die mitgelieferte Management-Anwendung auch die Steuerung über eine API an. So ist es möglich eigene Anwendungen zu entwickeln, die den gewünschten Anforderungen entsprechen. Zudem kann man auf diese Weise Anwendungen schaffen, die die Virtualisierer von verschiedenen Herstellern

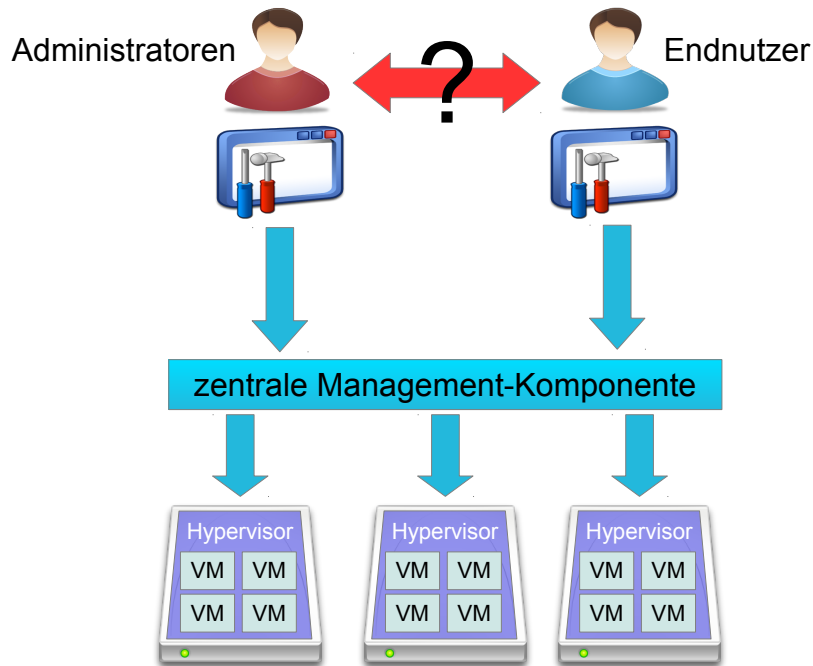


Abbildung 2.2.: Ein großes Defizit mitgelieferter Management-Anwendungen: keine integrierte Interaktionsmöglichkeit zwischen Administratoren und Endnutzern

steuern können. Mehr zu den verschiedenen existierenden Lösungen und zu der Frage, ob eine davon die Anforderungen dieser Arbeit erfüllt, findet man in Kapitel 4.

2.4. Die vSphere-Infrastruktur

Im Folgenden wird der Aufbau der vSphere-Infrastruktur und der Zusammenhang der einzelnen Komponenten beschrieben. vSphere dient dabei als beispielhafte Virtualisierungslösung, um die in den vorherigen Unterkapiteln beschriebene Theorie zu veranschaulichen. Zudem wird die Implementierung des Webportals, welche in Kapitel 6 beschrieben wird, auf die vSphere Infrastruktur aufbauen, so dass ein entsprechendes Verständnis über die Funktionsweise nötig ist.

Man kann feststellen, dass bei dieser Virtualisierungslösung alle im letzten Unterkapitel beschriebenen Komponenten vorkommen. Als Hypervisor kommt der so genannte **ESXi-Server** zum Einsatz. Dabei handelt es sich um einen Bare-Metal-Hypervisor. Auf dem Host, auf dem der ESXi-Server installiert ist, liegen zumeist auch die Konfigurations- und Festplattendateien der einzelnen VMs, sofern sie nicht auf einen Netzwerkspeicher ausgelagert wurden. Die verschiedenen Speichermöglichkeiten werden auf virtueller Ebene allerdings abstrahiert und erscheinen als ein großer Datenspeicher.

Kommen mehrere ESXi-Server zum Einsatz kann man die Verwaltung mit Hilfe des **vCenters** vereinfachen. Dieses wird auf einem Windows-Server als Dienst installiert und dient fortan als zentrale Steuereinheit innerhalb der gesamten Infrastruktur.

2. Grundlagen

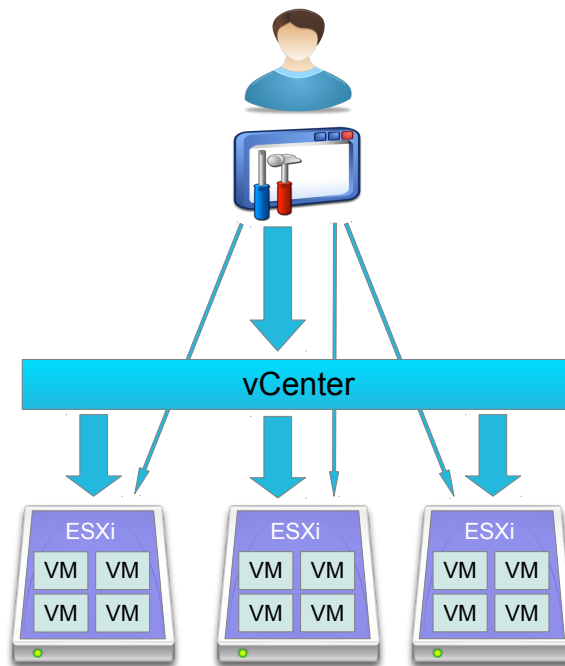


Abbildung 2.3.: Mit dem vSphere Client kann man die Infrastruktur sowohl über das vCenter steuern (dicke Pfeile), als auch auf die einzelnen ESXi-Server direkt zugreifen (dünne Pfeile)

Zur Steuerung bietet VMware primär den **vSphere Client**, auch VI Client genannt, an. Es handelt sich dabei um eine Windows-Software, welche über das Netz entweder direkt auf einen einzelnen ESXi-Server zugreift und diesen steuert, oder über das vCenter die gesamte Infrastruktur managt. In diesem Fall leitet das vCenter die Befehle, die von den vSphere Clients empfangen werden, an die ESXi-Server weiter, auf welchen diese vom vCenter Server-Agent empfangen und ausgeführt werden. Das vCenter administriert und überwacht die Infrastruktur allerdings auch dauerhaft, auch dann wenn kein Client angeschlossen ist und Aktionen ausführt. Ist ein vCenter im Einsatz, so bleibt die direkte Steuermöglichkeit des ESXi-Servers erhalten (siehe Abbildung 2.3). Der direkte ESXi-Zugriff kann zwar bei einer Störung des vCenters nützlich sein, wird aber oft aus Sicherheitsgründen nicht gewünscht. Die ESXi-Server müssen also mittels einer Firewall entsprechend abgesichert werden.

Der Funktionsumfang des vSphere Clients umfasst sowohl simple Tätigkeiten wie der Erstellung und Steuerung von VMs, die Überwachung der Ressourcenauslastung der VMs und des Hosts, den Zugriff auf die Konsole einer VM und den Zugriff auf den Datenspeicher des Hosts und setzt sich fort bis zu komplexen Abläufen wie der Live-Migration einer VM von einem ESXi-Server auf einen anderen.

Zu administrativen Zwecken ist es auch möglich direkt auf dem ESXi-Server ein Terminal, die so genannte Servicekonsole, zu nutzen oder einen SSH-Zugang zu aktivieren. Für den alltäglichen Betrieb, wenn also keine Störungen auftreten oder Wartungen anfallen, ist der Rückgriff hierauf aber nicht nötig.

Um von einem Rechner ohne vSphere Client die Infrastruktur verwalten zu können, stellt das vCenter auch noch den WebAccess zur Verfügung. Dabei handelt es sich um ein Web-

portal, welches die wichtigsten Aktionen, welche über den Client ausgeführt werden können, ebenfalls zur Verfügung stellt. Die Weiterentwicklung des WebAccess wurde allerdings mit Version 4.0U2 eingestellt. [WSW10]

Neben der Nutzung dieser von VMware entwickelten Tools, gibt es auch die Möglichkeit eigene Anwendungen zu entwickeln, welche auf die von den ESXi-Servern beziehungsweise vom vCenter bereitgestellte Web Service API zugreifen. Der Zugriff auf den Server erfolgt dabei über das Simple Object Access Protocol (SOAP), welches auf dem HTTP- beziehungsweise HTTPS-Protokoll aufsetzt. Die Schnittstelle wird über ein Web Services Description Language (WSDL) Dokument beschrieben. Im Prinzip kann für die Erstellung einer solchen Anwendung jede beliebige Programmiersprache benutzt werden. VMware bietet allerdings für Java, C# und Perl SDKs an, welche die Nutzung der API erleichtern, so dass man zum Beispiel nicht selbst SOAP-Nachrichten erzeugen muss. Im Rahmen dieser Arbeit wird das SDK für Perl verwendet.

Zur leichteren Verwaltung kann die gesamte vSphere-Infrastruktur auf verschiedenen Ebenen unterteilt werden. Man unterscheidet Cluster, welche vor allem der Ressourcenverwaltung dienen, Datacenter, über welche Namensräume realisiert werden können, Ordner, welche vor allem die Übersicht erleichtern sollen und Resourcepools, welche die Rechte delegation an einem Teil der Ressourcen eines Hosts oder Clusters ermöglichen.

Für die Nutzerverwaltung können lokal auf dem ESXi eingerichtete Benutzer, eine Windows-Domäne oder ActiveDirectory verwendet werden. Den einzelnen Nutzern können Rollen zugewiesen werden. Diesen Rollen oder auch einzelnen Nutzern können wiederum bestimmte Berechtigungen zugewiesen werden.

Man könnte die vSphere-Architektur an dieser Stelle viel weiter vertiefen und zum Beispiel auf Server-Module und -Komponenten eingehen. Doch da diese für diese Arbeit keine Rolle spielen, sei an der Stelle auf die entsprechende Dokumentation verwiesen.

2.5. Probleme mit dem vSphere-Management

Das Grundlagen-Kapitel wird an dieser Stelle mit einer kurzen Ausführung über die Probleme, die sich beim Management von vSphere ergeben, abgeschlossen. Als Beispiel soll das LRZ dienen, bei welchem eben diese Virtualisierungslösung zum Einsatz kommt.

Die im Kapitel 2.3 beschriebenen Defizite der mitgelieferten Management-Anwendungen treten auch beim vSphere Client auf. Dieser bietet keinerlei Funktionalität, die über den Betrieb hinausgeht, vor allem besteht also auch keine Möglichkeit der Interaktion zwischen Endnutzer und Administrator. Das LRZ behilft sich deshalb mit anderen Lösungen. Anträge für VMs werden zum Beispiel per Mail angenommen, was allerdings ein umständliches und fehleranfälliges Vorgehen ist.

Ein weiteres Problem am LRZ ist, dass man Endnutzern ungenügend den vSphere Client zur Steuerung ihrer VMs zur Verfügung stellt. Auf der einen Seite steht der Zwang zur Installation der Software, für welche ein Windows-Rechner benötigt wird und die umfangreichen Optionen in der Oberfläche, die einen Gelegenheitsnutzer überfordern könnten. Auf der anderen Seite bestehen Sicherheitsbedenken, da man für den Einsatz des vSphere Clients den Endnutzern direkten Zugriff auf das vCenter, die zentrale Steuerkomponente, geben muss. Mittels eines Denial-of-Service-Angriffs wäre es ein Leichtes, das vCenter zu überlasten und so die Steuerung der gesamten vSphere-Infrastruktur zu stören.

2. Grundlagen

Zumindest dem ersten Punkt konnte man begegnen, indem man Endnutzer auf WebAccess verwiesen hat. Dieses ist Betriebssystem unabhängig und bietet zudem eine einfachere Oberfläche. Das Sicherheitsproblem besteht allerdings weiterhin, da es sich beim WebAccess um eine Komponente handelt, die unter einem Tomcat-Server auf dem gleichen Rechner wie das vCenter läuft. Hinzu kommt die Problematik, dass die Weiterentwicklung von WebAccess eingestellt wurde. Ein Webportal, wie das in dieser Arbeit entwickelte, würde nicht auf dem vCenter-Rechner laufen und auf diesen per API zugreifen. In so einem Fall könnte man den Zugriff auf das vCenter für die Endnutzer blockieren. Eine Überlastung des Portals hätte nicht automatisch eine Überlastung des vCenters zur Folge.

3. Anforderungsanalyse

Im letzten Kapitel wurden die Grundlagen vermittelt, die für das weitere Verständnis der Arbeit notwendig sind. Nachdem diese Basis gelegt ist, kann der Fokus auf das zu entwickelnde Webportal zum Management von virtuellen Maschinen gerichtet werden. Doch bevor ein Konzept für dieses erarbeitet werden kann, müssen zunächst die Anforderungen ermittelt werden, die an ein solches gestellt werden. Der Anforderungskatalog, der das Ergebnis dieses Kapitels sein soll, wird einerseits zur Bewertung existierender Softwaresysteme bei der State of the Art Analyse genutzt und andererseits auch als Grundlage für die Entwicklung des Konzepts dienen.

Zu deren Ermittlung werden zwei Verfahren verwendet. Zunächst wird dem objektorientierten Softwareentwurf entsprechend eine szenariobasierte Anforderungsanalyse durchgeführt. Anschließend wird ein Fragebogen entwickelt, mit Hilfe dessen schließlich Administratoren und Endnutzer am LRZ und an der LMU befragt werden. Zum Ende dieses Kapitels werden die aus den Szenarios abgeleiteten Anforderungen und die ermittelten Bedürfnisse zusammengefasst, priorisiert und der Anforderungskatalog erstellt.

3.1. Szenariobasierte Anforderungsanalyse

Für die zunächst erfolgende szenariobasierte Anforderungsanalyse wurden zwei Szenarios entwickelt, die sich jeweils in zwei Unterszenarien teilen. Das erste Szenario beschreibt eine Situation am LRZ, das zweite Szenario eine an der LMU. Auf die Beschreibung von jedem Szenario folgt direkt eine Ableitung der jeweiligen Anwendungsfälle. Am Ende von diesem Unterkapitel werden aus den Anwendungsfällen Anforderungen abgeleitet.

3.1.1. Szenario 1 - Leibniz-Rechenzentrum

Ein Wissenschaftler an der Ludwig-Maximilians-Universität benötigt für seine Forschungen im Rahmen eines Physik-Projekts 100 VMs die bereits mit Debian Linux fertig eingerichtet sein sollen. Er ist an seinem Lehrstuhl einem technischen Administrator unterstellt, bei welchem er diese VMs beantragen muss. Dieser wiederum wendet sich an den Systemadministrator am LRZ, damit dieser die benötigte Anzahl an fertig eingerichteten VMs, die den gewünschten Anforderungen entsprechen, bereitstellt. Bei der Beantragung muss der Administrator des Lehrstuhls die Preise und die AGB des LRZ akzeptieren. Bei den VMs selbst bietet das LRZ zwei verschiedene Dienstypen an, die in den folgenden Varianten beschrieben sind. Sobald sich der technische Administrator für einen Dienstyp entschieden hat, die übrigen Angaben zur Ausstattung der VM getätigt und den Antrag abgeschickt hat, wird dieser durch einen Systemadministrator am LRZ geprüft und eine Genehmigung erteilt. Anschließend erstellt er die VMs und setzt auf diesen entsprechende Rechte damit der Kunde darauf zugreifen kann. Der technische Administrator installiert auf den VMs die vom Wissenschaftler benötigte Software, wozu er diverse ISO-Images mounten muss, und richtet für diesen ein Konto im Betriebssystem ein. Nach einigen Wochen stellt sich heraus, dass die

3. Anforderungsanalyse

VMs keine ausreichende Ausstattung haben. Der technische Administrator muss in diesem Fall beim LRZ eine Erhöhung der Ressourcen beantragen. Diese Erhöhung muss wieder von einem Systemadministrator am LRZ genehmigt werden. Sollte der technische Administrator irgendwann seine Arbeitsstelle wechseln, muss er vorher die Berechtigungen für die VMs an den neuen technischen Administrator übertragen. Sobald die VMs nicht mehr benötigt werden, reicht er beim LRZ eine Kündigung ein, woraufhin die VMs sofort automatisch gelöscht werden. Der Kunde kann je nach seinen Anforderungen, zum Beispiel auf Grund von rechtlichen Auflagen, entscheiden, ob Backups die während des Betriebs von seinen VMs angelegt wurden, ebenfalls sofort zum Zeitpunkt der Kündigung gelöscht werden sollen oder ob dies verzögert geschehen kann.

3.1.1.1. Beschreibung Variante A

Der technische Administrator entscheidet sich für das sogenannte Attended Hosting, das heißt es werden VMs bereitgestellt auf denen bereits das gewünschte Betriebssystem installiert ist. Auf diesen VMs werden zudem automatisiert Updates eingespielt. Damit dies möglich ist, behält das LRZ eine Zugangsmöglichkeit zu dem System.

3.1.1.2. Beschreibung Variante B

Der technische Administrator entscheidet sich an Stelle des Attended Hostings für das Unattended Hosting. Hierbei erhält er die VMs ohne vorinstalliertes Betriebssystem und kann dieses über die Konsole selbst installieren. Um die weitere Pflege muss er sich im Weiteren entsprechend auch selbst kümmern.

3.1.1.3. Abgeleitete Anwendungsfälle

An Hand der vorangegangenen Beschreibung kann man feststellen, dass es insgesamt zwei Arten von Akteuren gibt: Administratoren und Endnutzer. Die Endnutzer beantragen VMs und nutzen sie. Administratoren bearbeiten die durch die Endnutzer eingereichten Anträge und stellen ihnen die VMs bereit. Auf dieser Grundlage lassen sich Anwendungsfälle ermitteln, die im Folgenden sowohl beschrieben als auch durch ein Diagramm dargestellt werden. Diese Diagramme beschreiben auf anschauliche Weise wann welcher Akteur mit dem Portal interagieren muss, wann das Portal Aktionen durchführt und wann dieses mit dem Adapter und dieser wiederum mit dem Virtualisierer kommuniziert. Beim hier erwähnten Adapter handelt es sich um einen Vorgriff auf das Kapitel zum Konzept, es handelt es sich dabei um eine Komponente die generische Befehle des Portals in virtualisiererspezifische Befehle übersetzt. Der Adapter wird bereits an dieser Stelle in den Ablaufdiagrammen erwähnt, um ein vollständiges Bild der Abläufe zu gewährleisten.

Name: AntragVMNeu

Kurzbeschreibung: Einreichung und Bearbeitung eines Antrags auf das Bereitstellen einer neuen VM

Visualisierung: siehe Abbildung 3.1

Akteure: Initiiert vom Endnutzer; kommuniziert mit Administrator

Anfangsbedingung: Endnutzer und Administrator verfügen über ein Konto und sind über

dieses am System angemeldet.

Ereignisfluss: Primärszenario:

1. Endnutzer wählt einen gewünschten Diensttyp aus.
2. Endnutzer kann in Abhängigkeit vom gewählten Diensttyp eine VM durch die Angabe der gewünschten Ausstattungs- und Leistungsparameter beantragen.
3. System prüft ob gewünschte Parameter im zulässigen Bereich sind und fordert Endnutzer unter Umständen zur Korrektur auf.
4. System berechnet Preis und zeigt diesen dem Endnutzer an.
5. Endnutzer akzeptiert Preis und AGBs.
6. Administrator bekommt Antrag angezeigt und genehmigt diesen.
7. System erstellt VM mit gewählten Parametern.
8. Administrator kann weitere optionale Aufgaben und Modifikationen an der VM außerhalb des Systems vornehmen.
9. Administrator schließt Vorgang ab.
10. System gewährt dem Endnutzer Zugriff auf die VM und benachrichtigt ihn per Mail.

Abschlussbedingungen: VM wurde erstellt, Endnutzer hat Zugriff auf diese und wurde über die Zugriffsmöglichkeit per Mail informiert.

Ereignisfluss: Sekundärszenario:

1. Administrator lehnt Antrag ab.
2. Administrator gibt Grund für Ablehnung an; zusätzlich kann eine alternative Konfiguration vorgeschlagen werden, auf deren Grundlage der Endnutzer einen neuen Antrag einreichen kann.
3. System benachrichtigt Endnutzer über Ablehnung und Grund per Mail.

Abschlussbedingungen: Endnutzer wurde über abgelehnten Antrag per Mail informiert.

Name: AntragVMMod

Kurzbeschreibung: Einreichung und Bearbeitung eines Antrags auf das Modifizieren einer VM

Akteure: Initiiert vom Endnutzer; kommuniziert mit Administrator

Anfangsbedingung: Endnutzer und Administrator verfügen über ein Konto und sind über dieses am System angemeldet.

Ereignisfluss: Primärszenario:

1. System zeigt Endnutzer zur Verfügung stehende VMs an.
2. Endnutzer wählt VM und gewünschte Aktion (Modifikation).
3. Endnutzer wählt neue Ausstattungs- und Leistungsmerkmale für VM.
4. System prüft ob gewünschte Parameter im zulässigen Bereich sind und fordert Endnutzer unter Umständen zur Korrektur auf.

3. Anforderungsanalyse

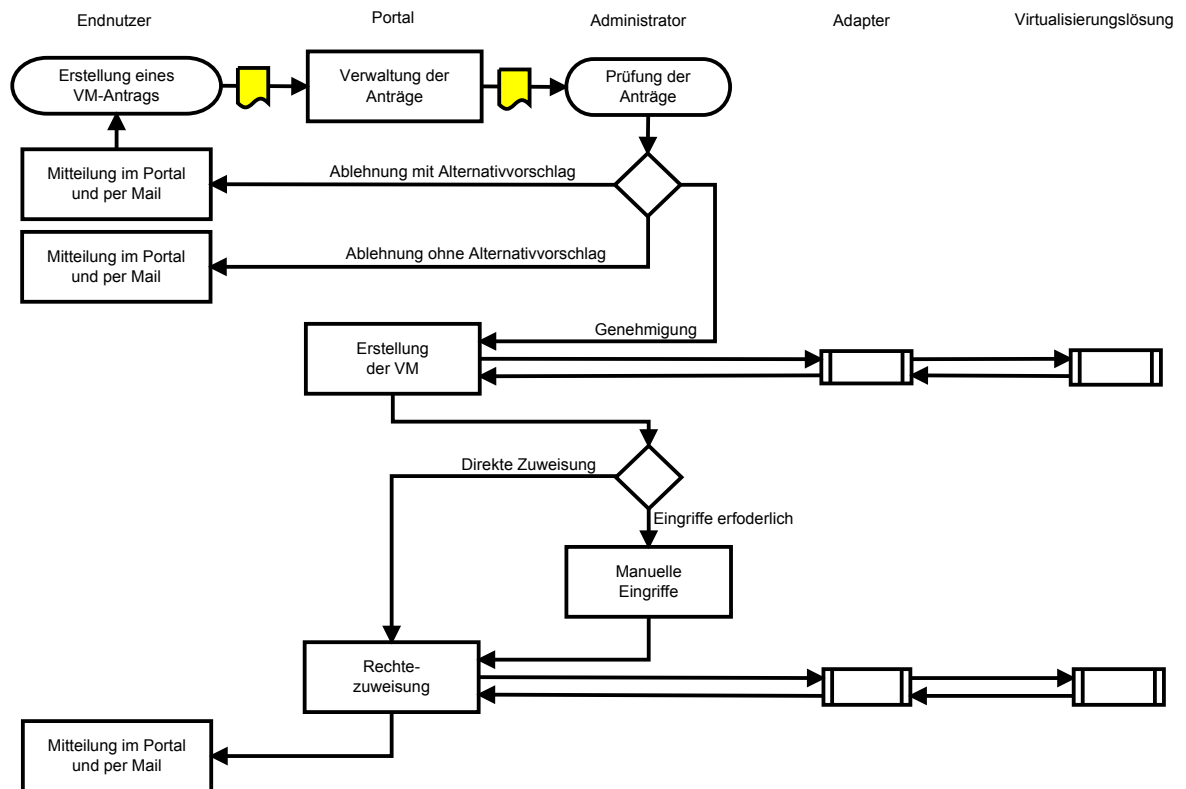


Abbildung 3.1.: Ablaufdiagramm des Anwendungsfalls AntragVMNeu

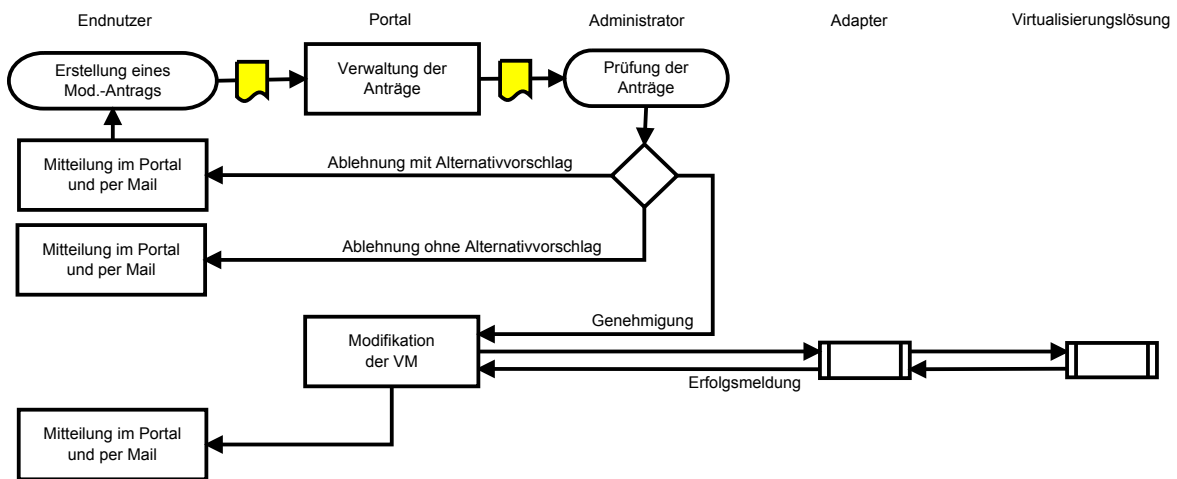


Abbildung 3.2.: Ablaufdiagramm des Anwendungsfalls AntragVMMMod

5. System berechnet Preis und zeigt diesen dem Endnutzer an.
6. Endnutzer akzeptiert Preis und AGBs.
7. Administrator bekommt Antrag angezeigt und genehmigt diesen.
8. System modifiziert VM.
9. System benachrichtigt Endnutzer per Mail.

Abschlussbedingungen: Gewählte VM verfügt über neue Ausstattung.

Ereignisfluss: Sekundärszenario:

1. Administrator lehnt Antrag ab.
2. Administrator gibt Grund für Ablehnung an; zusätzlich kann eine alternative Konfiguration vorgeschlagen werden, auf deren Grundlage der Endnutzer einen neuen Antrag einreichen kann.
3. System benachrichtigt Endnutzer über Ablehnung und Grund per Mail.

Abschlussbedingungen: Endnutzer wurde über abgelehnten Antrag per Mail informiert.

Name: AntragVMLöschung

Kurzbeschreibung: Einreichung und Bearbeitung eines Antrags auf das Löschen einer VM

Visualisierung: siehe Abbildung 3.3

Akteure: Initiiert vom Endnutzer

Anfangsbedingung: Endnutzer verfügt über ein Konto und ist über dieses am System angemeldet.

Ereignisfluss:

1. System zeigt Endnutzer zur Verfügung stehende VMs an.
2. Endnutzer wählt VM und gewünschte Aktion (Löschen).

3. Anforderungsanalyse

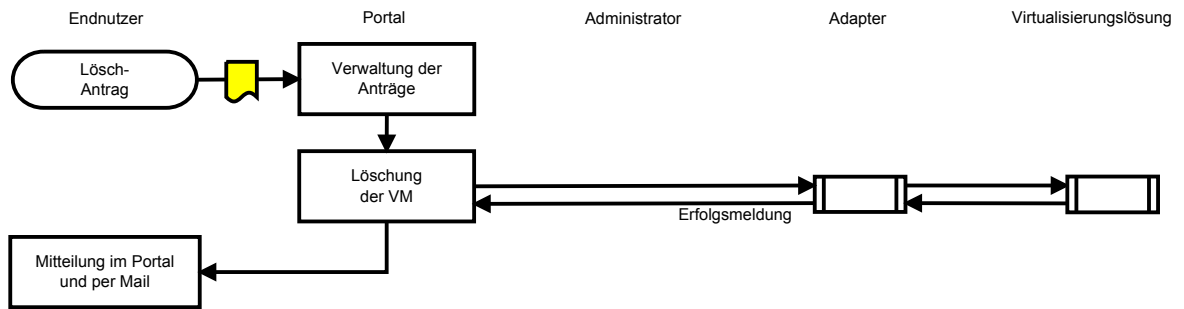


Abbildung 3.3.: Ablaufdiagramm des Anwendungsfalls AntragVMLöschung

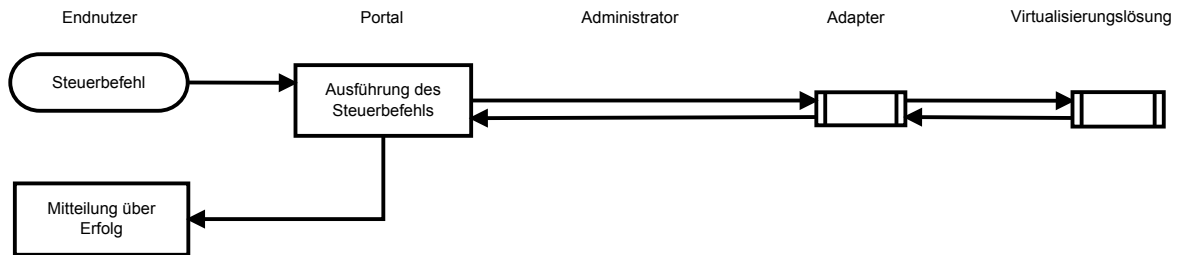


Abbildung 3.4.: Ablaufdiagramm der Anwendungsfälle SteuereVM, SteuereVM_Bulk, MountISO, SnapshotVM und KloneVM

3. System entzieht Endnutzer Zugriffsberechtigung und löscht VM.
4. System signalisiert Endnutzer erfolgreiche Ausführung und verschickt zur Bestätigung an diesen eine Mail.

Abschlussbedingungen: Gewählte VM existiert nicht mehr, Löschung wurde dem Endnutzer per Mail bestätigt.

Name: SteuereVM

Kurzbeschreibung: Start und Stopp einer VM

Visualisierung: siehe Abbildung 3.4

Akteure: Initiiert vom Endnutzer oder Administrator

Anfangsbedingung: Nutzer verfügt über ein Konto und ist über dieses am System angemeldet.

Ereignisfluss:

1. System zeigt Nutzer zur Verfügung stehende VMs an.
2. Nutzer wählt VM und gewünschte Aktion (Start, Stopp, Reset).
3. System führt gewünschte Aktion aus.
4. System signalisiert Nutzer erfolgreiche Ausführung.

Abschlussbedingungen: Gewählte Aktion wurde ausgeführt, VM wurde also gestartet, gestoppt oder zurückgesetzt.

Name: SteuereVM_Bulk

Kurzbeschreibung: Start und Stopp mehrerer VMs gleichzeitig

Visualisierung: siehe Abbildung 3.4

Akteure: Initiiert vom Endnutzer oder Administrator

Anfangsbedingung: Nutzer verfügt über ein Konto und ist über dieses am System angemeldet.

Ereignisfluss:

1. System zeigt Nutzer zur Verfügung stehende VMs an.
2. Nutzer wählt mehrere VMs und gewünschten Befehl.
3. System führt gewünschten Befehl nacheinander auf allen gewählten VMs aus.
4. System signalisiert Nutzer erfolgreiche Ausführung.

Abschlussbedingungen: Gewählte Aktion wurde auf allen VMs ausgeführt, VMs wurden also gestartet, gestoppt oder zurückgesetzt.

Name: MountISO

Kurzbeschreibung: Einhängen von ISO-Images

Visualisierung: siehe Abbildung 3.4

Akteure: Initiiert vom Endnutzer oder Administrator

Anfangsbedingung: Nutzer verfügt über ein Konto und ist über dieses am System angemeldet.

Ereignisfluss:

1. System zeigt Nutzer zur Verfügung stehende VMs an.
2. Nutzer wählt VM und gewünschte Aktion (Mount).
3. System zeigt Nutzer zur Verfügung stehende Images an.
4. Nutzer wählt Image.
5. System mountet gewünschtes Image.
6. System signalisiert Nutzer erfolgreiche Ausführung.

Abschlussbedingungen: Gewähltes Image ist in gewählter VM gemountet.

Name: StarteKonsole

Kurzbeschreibung: Start und Nutzung einer VM-Remote-Konsole

Visualisierung: siehe Abbildung 3.5

Akteure: Initiiert vom Endnutzer oder Administrator

Anfangsbedingung: Nutzer verfügt über ein Konto und ist über dieses am System angemeldet.

Ereignisfluss:

1. System zeigt Nutzer zur Verfügung stehende VMs an.
2. Nutzer initiiert Konsole.

3. Anforderungsanalyse

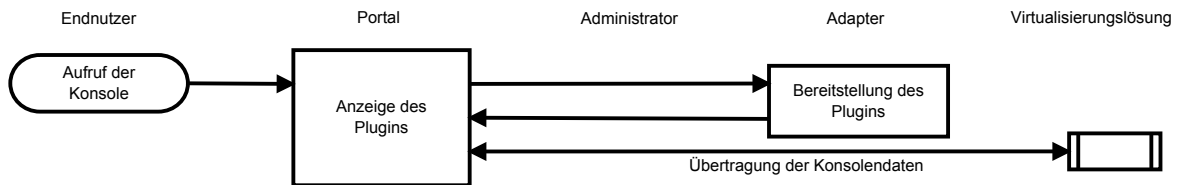


Abbildung 3.5.: Ablaufdiagramm des Anwendungsfalls StarteKonsole

3. System stellt Nutzer Konsolenzugriff ohne weitere Logins zur Verfügung.

Abschlussbedingungen: Nutzer hat Konsolenzugriff und kann diesen jederzeit beenden.

Name: ÄndereBesitzer

Kurzbeschreibung: Einreichung und Bearbeitung eines Antrags auf Übertragung einer VM an einen neuen Besitzer

Visualisierung: siehe Abbildung 3.6

Akteure: Initiiert vom Endnutzer oder Administrator; kommuniziert mit anderem Endnutzer
Anfangsbedingung: Nutzer verfügen über ein Konto und sind über dieses am System angemeldet.

Ereignisfluss: Primärszenario:

1. System zeigt Nutzer zur Verfügung stehende VMs an.
2. Nutzer wählt VM und gewünschte Aktion (Besitzerwechsel).
3. Nutzer gibt Kennung des neuen Besitzers ein.
4. System verschickt Mail an neuen Besitzer und fordert diesen zur Bestätigung auf.
5. Neuer Besitzer bekommt Details der VM (Ausstattung, Preis) angezeigt und bestätigt die Übernahme.
6. Administrator bekommt Antrag angezeigt und bestätigt diesen.
7. System ändert Zugriffsberechtigungen.
8. System versendet Benachrichtigung an alten und neuen Endnutzer.
9. System signalisiert erfolgreiche Ausführung.

Abschlussbedingungen: Neuer Nutzer kann auf Ressource zugreifen, altem Nutzer wurden Rechte entzogen.

Ereignisfluss: Sekundärszenario:

1. Administrator lehnt Antrag ab.
2. Administrator gibt Grund für Ablehnung an.
3. System benachrichtigt alten und neuen Endnutzer über Ablehnung und Grund per Mail.

Abschlussbedingungen: Endnutzer wurden über abgelehnten Antrag per Mail informiert.

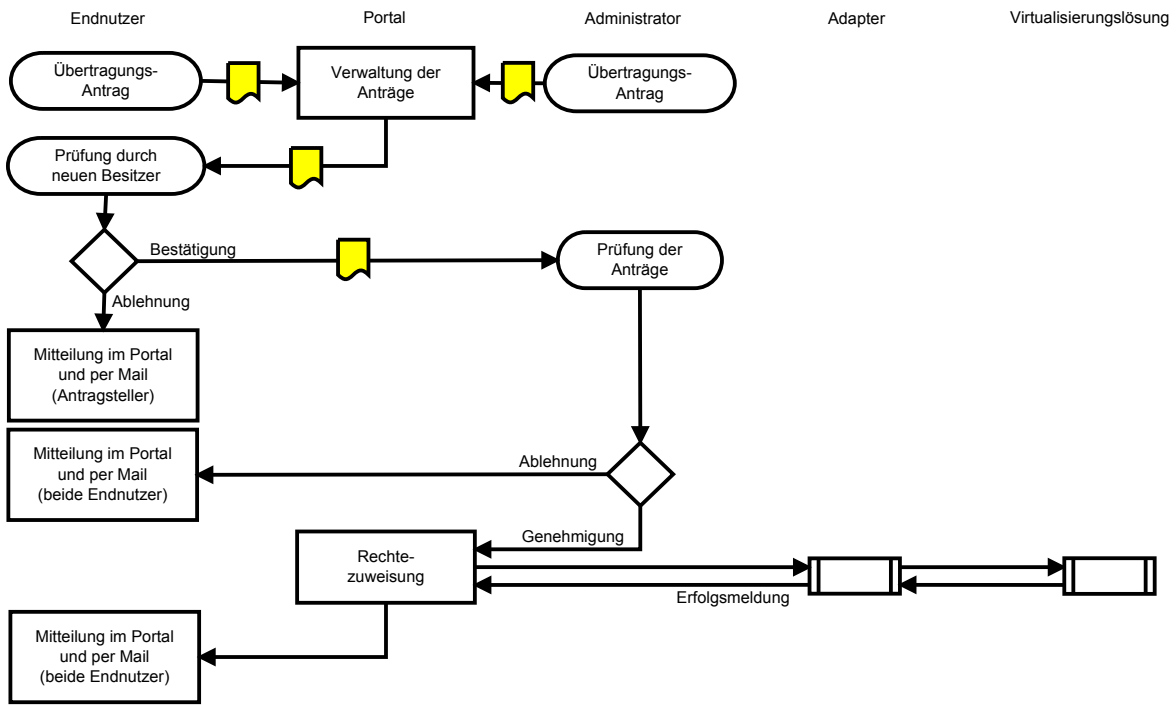


Abbildung 3.6.: Ablaufdiagramm des Anwendungsfalls ÄndereBesitzer

3.1.2. Szenario 2 - Institut für Informatik

Am Institut für Informatik an der LMU gibt es einen Systemadministrator, welcher für die IT-Infrastruktur inklusive des VM-Clusters verantwortlich ist. Des Weiteren gibt es einen wissenschaftlichen Mitarbeiter der für die Organisation und Durchführung eines Praktikums zuständig ist. Bei diesem Praktikum sollen die Studenten VMs zur Verfügung bekommen. Es bestehen prinzipiell zwei Möglichkeiten wie den Studenten die VMs zur Verfügung gestellt werden. Diese werden im Folgenden beschrieben.

3.1.2.1. Beschreibung Variante A

In dieser Variante beantragen die Studenten die VMs selbst direkt beim Systemadministrator. Der Systemadministrator prüft die Anträge und stellt die VMs bereit. Die Studenten haben ab diesen Zeitpunkt die Möglichkeit einige einfache Management-Aufgaben, wie das Starten und Stoppen selber auszuführen. Die Studenten können auch Snapshots anlegen, um bei einem Fehler jederzeit zum alten Stand zurückkehren zu können. Sobald das Praktikum abgeschlossen ist, werden die VMs wieder gelöscht.

3.1.2.2. Beschreibung Variante B

In dieser Variante beantragt der wissenschaftliche Mitarbeiter beim Systemadministrator einen Resourcepool. Hierbei gibt er an wie viele virtuelle CPUs, wie viel RAM und wie viel Speicherplatz er benötigt. Sobald der Systemadministrator den Antrag genehmigt und den Resourcepool zur Verfügung stellt, kann der wissenschaftliche Mitarbeiter im Rahmen seines Kontingents selbst VMs anlegen, modifizieren und löschen. Außerdem hat er die Möglichkeit

3. Anforderungsanalyse

auf einer VM Software zu installieren, die für das Praktikum benötigt wird und diese VM dann beliebig oft zu klonen. Nachdem das Praktikum beendet ist, kann der wissenschaftliche Mitarbeiter bestimmte Daten die während des Praktikums entstanden sind zu Archivierungszwecken sichern und anschließend eine Löschung des Resourcepools beantragen, woraufhin die Ressourcen auf dem VM-Cluster wieder freigegeben werden.

3.1.2.3. Abgeleitete Anwendungsfälle

Wie schon beim ersten Szenario können auch in diesem Fall einzelne Anwendungsfälle identifiziert werden. Neu ist bei diesem Szenario die Gruppe der Resourcepool-Nutzer. Bei diesen handelt es sich um Endnutzer, die einen Resourcepool zugewiesen bekommen haben und somit über erweiterte Rechte verfügen. Diese erlauben es ihnen VMs anzulegen oder zu modifizieren, ohne dies vorher beantragen zu müssen. Ein Resourcepool-Nutzer kann parallel zu seinem Resourcepool auch beantragte Einzel-VMs besitzen und kann somit gleichzeitig ein gewöhnlicher Endnutzer sein. Die bereits definierten Anwendungsfälle `SteuereVM`, `SteuereVM_Bulk`, `MountISO`, `StarteKonsole` und `ÄndereBesitzer` können auch von einem Resourcepool-Nutzer initiiert werden.

Im Folgenden werden weitere aus dem zweiten Szenario abgeleitete Anwendungsfälle beschrieben und wieder durch Ablaufdiagramme ergänzt.

Name: SnapshotVM

Kurzbeschreibung: Erzeugung eines neuen Snapshots einer VM

Visualisierung: siehe Abbildung 3.4

Akteure: Initiiert vom Endnutzer, Resourcepool-Nutzer oder Administrator

Anfangsbedingung: Nutzer verfügt über ein Konto und ist über dieses am System angemeldet.

Ereignisfluss:

1. System zeigt Nutzer zur Verfügung stehende VMs an.
2. Nutzer wählt VM und gewünschte Aktion (Snapshot).
3. System führt gewünschte Aktion aus.
4. System signalisiert Nutzer erfolgreiche Ausführung.

Abschlussbedingungen: Gewählte Aktion wurde auf VM ausgeführt, Snapshot wurde angelegt.

Name: KloneVM

Kurzbeschreibung: Erzeugung eines Klons einer VM

Visualisierung: siehe Abbildung 3.4

Akteure: Initiiert vom Resourcepool-Nutzer oder Administrator

Anfangsbedingung: Nutzer verfügt über ein Konto und ist über dieses am System angemeldet.

Ereignisfluss:

1. System zeigt Nutzer zur Verfügung stehende VMs an.

2. Nutzer wählt VM und gewünschte Aktion (Klonen).
3. System kloniert VM und gewährt Resourcepool-Nutzer Rechte.
4. System signalisiert erfolgreiche Ausführung.

Abschlussbedingungen: Klon von gewählter VM wurde erstellt.

Name: ErstelleRPVM

Kurzbeschreibung: Erstellung einer VM in einem Resourcepool

Visualisierung: siehe Abbildung 3.7

Akteure: Initiiert vom Resourcepool-Nutzer

Anfangsbedingung: Resourcepool-Nutzer verfügt über ein Konto und ist über dieses am System angemeldet.

Ereignisfluss:

1. Resourcepool-Nutzer wählt gewünschte Ausstattungs- und Leistungsparameter.
2. System prüft ob gewünschte Parameter im zulässigen Bereich sind und ob ausreichend freie Poolressourcen zur Verfügung stehen; fordert Resourcepool-Nutzer unter Umständen zur Korrektur auf.
3. System erstellt automatisch VM mit gewählten Parameter und passt Zugriffsberechtigungen an.
4. System signalisiert erfolgreiche Ausführung..

Abschlussbedingungen: VM wurde erstellt, Resourcepool-Nutzer hat Zugriff auf diese.

Name: ModifiziereRPVM

Kurzbeschreibung: Modifikation einer VM in einem Resourcepool

Visualisierung: siehe Abbildung 3.7

Akteure: Initiiert vom Resourcepool-Nutzer

Anfangsbedingung: Resourcepool-Nutzer verfügt über ein Konto und ist über dieses am System angemeldet.

Ereignisfluss:

1. System zeigt Resourcepool-Nutzer zur Verfügung stehende VMs an.
2. System wählt zu modifizierende VM aus.
3. Resourcepool-Nutzer wählt zu ändernde Ausstattungs- und Leistungsparameter.
4. System prüft ob gewünschte Parameter im zulässigen Bereich sind und ob ausreichend freie Poolressourcen zur Verfügung stehen; fordert Resourcepool-Nutzer unter Umständen zur Korrektur auf.
5. System modifiziert automatisch VM mit gewählten Parameter.
6. System signalisiert erfolgreiche Ausführung.

Abschlussbedingungen: VM steht mit neuer Ausstattung zur Verfügung.

3. Anforderungsanalyse

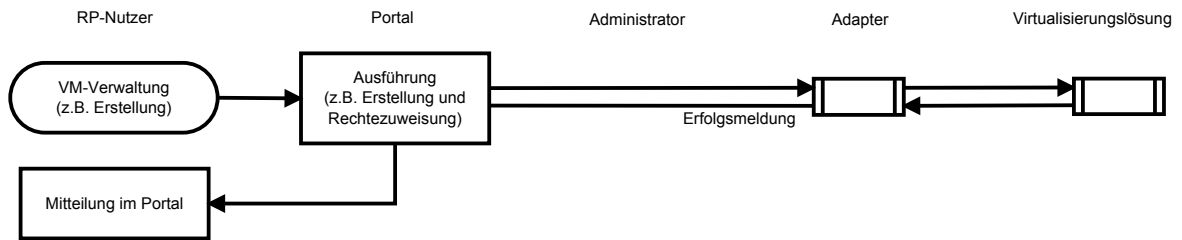


Abbildung 3.7.: Ablaufdiagramm der Anwendungsfälle ErstelleRPVM, ModifiziereRPVM und LöscheRPVM

Name: LöscheRPVM

Kurzbeschreibung: Löschung einer VM in einem Resourcepool

Visualisierung: siehe Abbildung 3.7

Akteure: Initiiert vom Resourcepool-Nutzer

Anfangsbedingung: Resourcepool-Nutzer verfügt über ein Konto und ist über dieses am System angemeldet.

Ereignisfluss:

1. System zeigt Resourcepool-Nutzer zur Verfügung stehende VMs an.
2. Resourcepool-Nutzer wählt zu löschende VM aus.
3. System löscht automatisiert VM.
4. System signalisiert erfolgreiche Ausführung.

Abschlussbedingungen: VM wurde gelöscht.

Die Anwendungsfälle AntragRPNeu, AntragRPMod und AntragRPLöschung entsprechen in wesentlichen den Anwendungsfällen AntragVMNeu, AntragVMMod und AntragVMLöschung und werden deshalb nicht ausformuliert.

Das Use-Case-Diagramm in Abbildung 3.8 fasst die Anwendungsfälle beider Szenarien zusammen und macht nochmals deutlich welche Nutzergruppe bei welchem Anwendungsfall beteiligt ist.

3.1.3. An Hand der szenariobasierten Analyse ermittelte Anforderungen

Auf Grundlage dieser Anwendungsfälle können die folgenden funktionalen Anforderungen abgeleitet werden.

- Endnutzer können durch Einreichung eines Antrags VMs beantragen.
- Antragsteller müssen unter Umständen AGBs und den Preis akzeptieren.
- Administratoren können Beantragungen einsehen und diese genehmigen oder ablehnen.
- Bei einer Genehmigung erstellt das System die VM oder den Resourcepool und stellt diese dem Nutzer sofort zur Verfügung oder gewährt dem Administrator vorher noch die Möglichkeit administrative Eingriffe vorzunehmen.

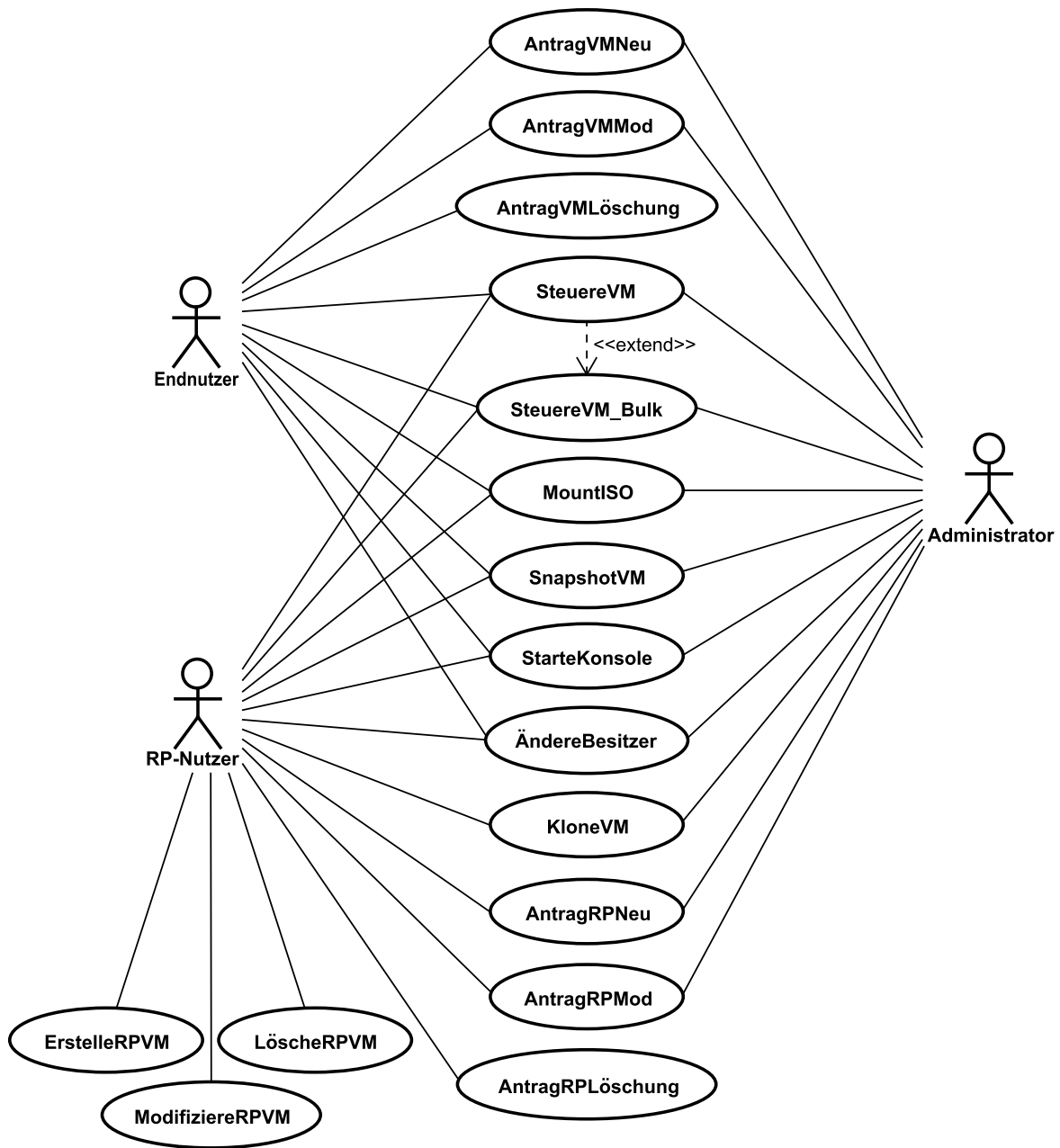


Abbildung 3.8.: Use-Case-Diagramm

3. Anforderungsanalyse

- Bei einer Ablehnung muss der Administrator einen Grund angeben. Außerdem kann er eine alternative Konfiguration vorschlagen. Der Endnutzer kann auf Grundlage dieses Alternativvorschlags einen neuen Antrag einreichen.
- System benachrichtigt automatisch die Antragsteller per E-Mail über bereitgestellte VMs.
- Nutzer können für die Parameter mit denen eine VM beantragt wurde eine Änderung beantragen.
- Nutzer können folgende Aktionen auf den VMs ausführen:
 - Starten
 - Stoppen
 - Zurücksetzen
 - ISO-Images einhängen
 - Snapshots anlegen
 - Auf die Konsole zugreifen
 - Mindestens Start, Stopp und Reset sollten als Bulk-Aktion verfügbar, also auf mehreren VMs gleichzeitig ausführbar sein.
- Neben VMs können auch Ressourcepools beantragt werden. Die Abläufe bei der Beantragung und Bereitstellung sind die selben wie bei einer VM.
- Ressourcepool-Nutzer können im Vergleich zu Endnutzern die folgenden zusätzliche Aktionen durchführen:
 - Erstellen neuer VMs
 - Modifizieren bestehender VMs
 - Klonen von VMs
- Administratoren können das Klonen zur Bereitstellung von Attended VMs verwenden.
- Wünscht ein End- oder Ressourcepool-Nutzer die Löschung einer VM, wird dies sofort automatisch ausgeführt.
- Endnutzer, Ressourcepool-Nutzer und Administratoren können für eine VM einen Besitzerwechsel beantragen. Ein solcher Antrag muss sowohl vom neuen potentiellen Besitzer, als auch anschließend von einem Administrator bestätigt werden.

3.2. Nutzerbefragung

Im jetzt durchzuführendem zweiten Schritt zur Ermittlung der Anforderungen wird eine Befragung unter potentiellen Nutzern eines solchen Portals erfolgen. Um zu vermeiden, dass sich die Ergebnisse zu sehr an den Anforderungen einer einzelnen Institution orientieren, wird die Befragung sowohl am LRZ als auch an der LMU durchgeführt. Beide Institutionen befinden sich zwar im akademischen Umfeld, ansonsten sind die Umstände aber durchaus unterschiedlich.

Das primäre Ziel dieser Befragung soll es sein, die Bedürfnisse der Administratoren und Endnutzer an diesen Institutionen zu ermitteln. Möglicherweise lassen sich dabei Widersprüche zwischen den Wünschen der Administratoren und der Endnutzern feststellen, die so

nicht umgesetzt werden können. In diesem Fall sollen diese Widersprüche die Grundlage für weitere Gespräche bilden. Außerdem lassen sich bei den Gesprächen möglicherweise Fehler beziehungsweise praxisferne Situationen in den Szenarios feststellen.

In den nun folgenden Unterkapiteln wird zunächst erläutert, wie der Fragebogen, mit Hilfe dessen die Befragung durchgeführt wurde, erstellt wurde, welche Personen interviewt wurden und welche Daten ermittelt wurden. Zum Schluss werden auf diesen Daten basierende Anforderungen abgeleitet.

3.2.1. Erstellung des Fragebogens

Um die Befragung durchzuführen, wurde ein Fragebogen entwickelt. Als Grundlage für diesen Fragebogen dienten die Szenarios und die bereits daraus abgeleiteten Anforderungen. Indem diese Informationen in den Fragebogen miteinfließen, können sie so nochmals verifiziert werden. Der Fragebogen unterteilt sich zunächst in zwei Hauptteile. Der erste Teil umfasst Fragen, die lediglich von den Administratoren beantwortet werden können. Der zweite Teil richtet sich an die Endnutzer, wird aber ebenfalls von den Administratoren beantwortet, da diese Erfahrung mit den Wünschen ihrer Endnutzer haben. Der Fragebogen enthält zudem einige Abschnitte, die nur an die Administratoren und Endnutzer am LRZ gerichtet sind, da bei der LMU einerseits der Dienstyp Attended Hosting nicht angeboten wird und andererseits die VMs an die Mitarbeiter kostenlos vergeben werden, folglich Fragen zur Abrechnung und zu den AGBs keine Rolle spielen. Die Endnutzer können unter Umständen einige Fragen, die sich auf das Attended oder Unattended Hosting beziehen nicht beantworten, da sie nur einen dieser beiden Dienste nutzen. Einige der Fragen mögen für die Ermittlung von Anforderungen, um auf deren Grundlage ein Konzept zu erarbeiten nicht relevant erscheinen. Sie sind aber für die spätere Implementierung, die primär beim LRZ, aber auch bei der LMU laufen soll umso wichtiger und wurden deshalb ebenfalls in diesen Fragebogen mitaufgenommen. Ein vollständiger Fragebogen kann im Anhang eingesehen werden. Diesem können sowohl alle 33 Fragen, als auch die möglichen Antwortmöglichkeiten entnommen werden.

3.2.2. Durchführung

Insgesamt wurden am LRZ und an der LMU jeweils drei Administratoren und am LRZ zudem zwei Endnutzer befragt. Die befragten Personen wurden von den Betreuern dieser Arbeit vermittelt. Vor Beginn der Befragung wurden die Interviewpartner aufgefordert nicht nur auf die Fragen entsprechend der möglichen Antwortvorgaben zu antworten, sondern auch darüber hinaus alle Gedanken zu äußern, die ihnen im jeweiligen Zusammenhang relevant erscheinen. Auf diese Weise sollen nicht nur quantitative Daten erhoben werden, die unter anderem für die Priorisierung der verschiedenen Wünsche wichtig sind, sondern auch Wissen über mögliche Aspekte gewonnen werden, die bisher keine Beachtung gefunden haben.

3.2.3. Ermittelte Daten

Die Auswertung der Daten ist in zwei Schritten erfolgt. Im ersten Schritt wurden die quantifizierbaren Daten zusammengefasst und an Hand dieser Daten eine Priorisierung der Wünsche vorgenommen. Im zweiten Schritt wurden die sonstigen Aussagen der Interviewten analysiert und auf dieser Grundlage die Szenarios, Anwendungsfälle und daraus abgeleiteten Anforder-

3. Anforderungsanalyse

rungen verifiziert und verfeinert. Außerdem wurden die Antworten der Administratoren mit denen der Endnutzer verglichen, um eventuelle Widersprüche zu finden.

Im Folgenden findet sich eine beispielhafte Auswahl einiger ermittelter Daten. Die Zahlen in Klammern geben jeweils an wie viele von wie vielen Nutzer diese Antwort gaben.

- Bei Erstellung eines Antrags sollen folgende Parameter wählbar sein: Anzahl virtueller CPUs (6/6), Größe des RAM (6/6), Größen der Festplatten (6/6), Betriebssystem (2/6), Netz/Erreichbarkeit (3/6), Anzahl Netzwerkkarten (1/6), Hostname (1/6), Mountpoints (1/6).
- Dieselben Parameter sollen zu einem späteren Zeitpunkt veränderbar sein (5/6).
- Resourcepools kommen sowohl am LRZ als auch an der LMU nicht zum Einsatz, ihre Unterstützung wird aber als sinnvoll angesehen und gewünscht.
- Am LRZ ist die Netztopologie, an die die VMs angeschlossen werden, flach. Es werden mehrere VLANs mit unterschiedlicher Erreichbarkeit betrieben. An der LMU können auf Wunsch äußerst komplizierte Topologien eingerichtet werden.
- Die Entscheidung ob eine VM genehmigt wird fällt immer individuell, dieser Vorgang lässt sich durch vorgegebene Grenzwerte nicht automatisieren. An der LMU existieren zudem minimale Grenzwerte.
- IP-Adressen werden an beiden Institutionen manuell zugewiesen, eine Automatisierung davon wird gewünscht.
- Häufig durchgeführte Aktionen (zum Beispiel Starten und Stoppen) sollen als Bulk-Aktionen verfügbar sein (5/6).
- Die Möglichkeit Snapshots anzulegen besteht entweder immer für alle Nutzer (3/5) oder wird auf Antrag gewährt (2/5).
- Benachrichtigungen sollen bevorzugt per Mail verschickt werden (6/6).
- Der Preis für eine VM lässt sich über eine Formel ermitteln, Großkunden erhalten unter Umständen Rabatt.
- Der Nutzer kann beim Attended Hosting nur zwischen Linux und Windows wählen.
- Eine mobile Version des Webportals wird nicht benötigt (8/8).
- Zugriff auf das BIOS wird in der Regel nicht benötigt (5/8).
- Für die Konsole wird ein Browser-Plugin gegenüber eine Standalone-Anwendung bevorzugt (7/8).

Abseits der quantitativen Daten konnte aus den Gesprächen die Erkenntnis gewonnen werden, dass die Backupsysteme weitestgehend automatisch für jede VM eingerichtet werden und der Kunde hierauf im Normalfall keinen Einfluss hat. Dies bedeutet, dass die Art des Backups bei der Antragsstellung nicht berücksichtigt werden muss. Von Seiten der Endnutzer wird ein solcher Einfluss auch nicht benötigt.

Des Weiteren wurde von den Administratoren am LRZ besonders nachdrücklich der Wunsch geäußert, dass das Webportal über eine integrierte Verwaltung und Zuweisung von IP-Adressen verfügen soll. Freie IP-Adressen werden zur Zeit über manuell geführte Listen verwaltet, was sehr umständlich ist.

Beim Vergleich der Wünsche von Administratoren und Endnutzern wurden folgende Erkenntnisse gewonnen:

- Das LRZ bietet seinen Kunden keine Standardkonfigurationen zur Auswahl an, solche werden von den Endnutzern aber gewünscht. Eine weitere Rücksprache mit den Administratoren hat ergeben, dass nichts gegen eine solche Auswahl sprechen würde, folglich wird eine solche vorgesehen.
- 50% der Administratoren sind dagegen den Endnutzern einen Zugriff auf das BIOS der VMs zu gewähren. Entweder sie erachten dies als unnötig oder haben in geringerem Maße sogar Sicherheitsbedenken. Auf Seiten der Endnutzer wird der BIOS-Zugriff nur von einer Minderheit gewünscht. Da im Rahmen eines Konsolenzugriffs, der für das Unattended Hosting zweifellos nötig ist, ein Zugriff auf das BIOS aber nicht verhindert werden kann, ist eine weitere Vertiefung dieser Problematik nicht nötig.
- Die Administratoren gewähren den Endnutzern nur eingeschränkte Möglichkeiten eigene ISO-Images einzubinden (zum Beispiel von der lokalen Festplatte über die Konsole). Von den Endnutzern wird aber die Möglichkeit auch eigene Images hochladen zu können gewünscht. Eine Rücksprache mit den Administratoren hat ergeben, dass, wenn die Managementlösung eine solche Möglichkeit bietet und die Endnutzer keinen gegenseitigen Zugriff auf die Images bekommen, nichts dagegen spricht.
- Die Administratoren des LRZ bieten standardmäßig nur Attended Hosting mit automatischen Updates oder Unattended Hosting ohne installiertes Betriebssystem an. Von den Endnutzern wird allerdings durchaus der Wunsch nach Attended Hosting ohne automatische Updates geäußert. Eine Rücksprache mit den Administratoren hat ergeben, dass die Updates abgeschaltet werden können, dies allerdings einen manuellen Eingriff erfordert und deshalb nur in Ausnahmefällen praktiziert wird. Das Webportal sollte folglich eine Automatisierung hiervon vorsehen und die Updates als wählbaren Parameter des Attended Hosting einführen.

3.2.4. Durch die Nutzerbefragung ermittelte Anforderungen

Um den Rahmen dieser Arbeit nicht zu sprengen, wurden bei den durch die Befragungen ermittelten Wünschen, jeweils die am häufigsten genannten berücksichtigt. Konkret bedeutet dies, dass jeweils von nur einer Person genannte Punkte in der Regel nicht berücksichtigt wurden. Dies gilt im Besonderen dann, wenn ein Punkt in eine völlig andere Richtung führt, als die Anforderungen der übrigen Nutzer. So wurde zum Beispiel von einer Administratorin an der LMU gefordert, dass bei der Beantragung einer VM der Antragsteller keine konkreten Angaben zur Ausstattung der VM tätigen soll, sondern dass dieser lediglich einen Fragebogen beantworten soll, in dem er Angaben zu seinen Plänen und Absichten mit der VM machen soll. Über die Hardware-Ausstattung würde anschließend ein Administrator auf Grundlage des vom Antragsteller beschriebenen Einsatzzweckes entscheiden. Auf der anderen Seite wurden von einzelnen Personen geäußerte Wünsche berücksichtigt, wenn sich diese vor allem auf die konkrete Implementierung bezogen und keinen besonderen Aufwand erfordern beziehungsweise durch eine andere Anforderung bereits zum großen Teil abgedeckt sind. Folgende Anforderungen, die über die bereits definierten hinausgehen, können aus den Ergebnissen der Nutzerbefragung abgeleitet werden:

- Bei einem Antrag auf eine neue VM können folgende Parameter beeinflusst werden: Anzahl virtueller CPUs, Größe des RAM, Größe der Festplatten, Netz/Erreichbarkeit, Art des Hostings, Betriebssystem (nur Attended Hosting), Updates (nur Attended Hosting).

3. Anforderungsanalyse

- Endnutzer und Resourcepool-Nutzer können für die Zusammenstellung ihrer VMs auf vordefinierte Konfigurationen zurückgreifen und diese modifizieren.
- Angaben des Nutzers müssen auf Plausibilität geprüft werden und müssen in einem definiertem Wertebereich liegen.
- Endnutzer müssen AGBs nur akzeptieren, wenn die aktuelle Fassung bisher noch nicht akzeptiert wurde.
- Das System soll IP-Adressen verwalten und diese den VMs zuweisen.
- Start, Stopp und Reset müssen als Bulk-Aktionen verfügbar sein.
- Administratoren sollen darüber informiert werden, wenn ein Endnutzer eine seiner VMs löscht.
- Endnutzer und Resourcepool-Nutzer sollen die Möglichkeit haben eigene ISO-Images hochzuladen. Diese hochgeladenen Images sollen nur dem jeweiligen Nutzer zur Verfügung stehen.
- Für alle Nutzergruppen sollen Statistiken und Performancedaten zu den VMs abrufbar sein. Resourcepool-Nutzer sollen zudem einsehen können welchen Anteil ihrer Ressourcen sie bereits verbraucht haben.

3.3. Zusammenfassung der funktionalen Anforderungen

Im Folgenden ist eine Zusammenfassung der funktionalen Anforderungen, die über die szenariobasierte Analyse und die Nutzerbefragung ermittelt wurden, zu finden. Die folgende Übersicht vermittelt die Anforderungen gegliedert nach den verschiedenen Phasen des Dienstlebenszykluses.

Bereitstellung einer VM

- Endnutzer können VMs beantragen, wofür von ihnen, je nach angebotenen Diensten, folgende Angaben gemacht werden müssen: Anzahl virtueller CPUs, Größe des RAM, Größe der Festplatten, Netz/Erreichbarkeit, Art des Hostings, Betriebssystem (nur Attended Hosting), Updates (nur Attended Hosting).
- Endnutzer können für die Zusammenstellung ihrer VMs auf vordefinierte Konfigurationen zurückgreifen und diese modifizieren.
- Angaben der Endnutzer müssen auf Plausibilität geprüft werden und in einem definiertem Wertebereich liegen.
- Endnutzer müssen die aktuelle Version der AGBs akzeptieren, falls noch nicht geschehen.
- Endnutzer bekommen vor Abschluss eines Antrags einmalige und monatliche Gebühren angezeigt.
- Administratoren können Beantragungen einsehen und sie entweder genehmigen oder ablehnen.
- Bei einer Genehmigung erstellt das System auf Grundlage der vom Endnutzer gemachten Angaben die VM, ordnet dieser eine freie IP-Adresse zu und stellt diese entsprechend der Auswahl eines Administrators sofort bereit oder ermöglicht diesem vorher noch einen manuellen Eingriff.

- Bei einer Ablehnung muss vom Administrator eine Begründung angegeben werden, die vom System an den Endnutzer übermittelt werden muss. Zudem können die Administratoren Alternativkonfigurationen vorschlagen, auf deren Basis die Endnutzer neue Anträge einreichen können.
- Endnutzer werden über fertige VMs automatisch per E-Mail informiert.

Betrieb einer VM

- Endnutzer können für die Parameter, mit denen die VM beantragt wurde, eine Änderung beantragen.
- Administratoren können Anträge für Modifikationen auf die gleiche Weise bearbeiten wie bei einem Bereitstellungsantrag.
- Administratoren und Endnutzer können folgende Aktionen auf den VMs ausführen:
 - Starten
 - Stoppen
 - Reseten
 - Eigene ISO-Images hochladen und diese und optional lokal verfügbare Images einhängen
 - Den Betriebsstatus und wichtige Performance-Daten und Statistiken einsehen
 - Snapshots anlegen
 - Auf die Konsole zugreifen (am besten über ein Browser-Plugin)
 - Mindestens Start, Stopp und Reset sind als Bulk-Aktion verfügbar.
- Endnutzer und Administratoren können für eine VM einen Besitzerwechsel beantragen. Ein solcher Antrag muss sowohl vom neuen potentiellen Besitzer als auch anschließend von einem Administrator bestätigt werden.

Kündigung einer VM

- Endnutzer können für ihre VMs Löschungen beantragen, diese werden sofort automatisch durchgeführt.
- Administratoren erhalten Mails bezüglich solcher Löschungen.

Bereitstellung, Betrieb und Kündigung eines Ressourcepools

- wie bei einer VM mit folgenden Unterschieden
- Ressourcepool-Nutzer können folgende zusätzliche Aktionen durchführen:
 - Erstellen neuer VMs
 - Modifizieren bestehender VMs
 - Klonen von VMs
- Administratoren und Ressourcepool-Nutzer können Einsehen welcher Anteil der Ressourcen bereits verbraucht wurde beziehungsweise noch zur Verfügung steht.

3.4. Nicht-funktionale Anforderungen

Unabhängig von den funktionalen Anforderungen, die sich in erster Linie aus der szenario-basierten Anforderungsanalyse und der Nutzerbefragung ergeben, gibt es eine Reihe nicht-funktionaler Anforderungen. Ein Großteil von diesen leitet sich aus der allgemeinen Problemstellung und der Motivation, die zur Erstellung eines Webportals zum Management virtueller Maschinen geführt haben, ab. Andere ergeben sich aus Sicherheits- und Verfügbarkeitsanforderungen, wie sie zum Beispiel am LRZ gegeben sind.

- Das System sollte sich leicht an die Bedürfnisse der einsetzenden Institution anpassen lassen. Um dies zu ermöglichen sollte die Software am besten quelloffen sein und unter einer freien Lizenz stehen. Auf proprietäre Lösungen sollte verzichtet werden.
- Das System muss auf die Rechteverwaltung des Virtualisierers aufsetzen, um weiterhin die Nutzung der bisherigen Managementlösungen beziehungsweise einen fließenden Wechsel zu ermöglichen. Da das Portal in eine bestehende Infrastruktur integriert wird, kann ein radikaler Wechsel der Management-Software nicht vorausgesetzt werden. Damit existierende Software weiterhin genutzt werden kann, muss auf der existierenden Rechteverwaltung aufgebaut werden. Würde das Webportal auf ein eigenes Rechtemanagement setzen, so müsste es auch die Funktionalität aller bisheriger Anwendungen übernehmen. Da aber ansonsten die verbleibende Software problemlos weiter eingesetzt werden kann, können sich diese und das Webportal entsprechend ergänzen.
- Für externe Anwendungen muss eine Möglichkeit geschaffen werden auf Nutzungsdaten zuzugreifen. Dies wird zum Beispiel für das Accounting benötigt. In direkter Folge bedeutet dies auch, dass entsprechende Nutzungsdaten, also vor allem wer von wann bis wann was für eine VM oder eine Ressourcepool genutzt hat, gespeichert werden müssen.
- Das System muss auf eine große Anzahl von gleichzeitigen Nutzern ausgelegt sein (Skalierbarkeit).
- Das System muss ausfallsicher aufgesetzt werden können (Verfügbarkeit).
- Das System muss leicht für die Verwaltung neuer Virtualisierer angepasst werden können. Diese Anforderung leitet sich daraus ab, dass das Webportal generisch sein soll und nicht nur für den Einsatz an einer speziellen Institution geplant sein soll.

3.5. Anforderungskatalog

Die in den letzten beiden Unterkapiteln ausführlich beschriebenen funktionalen und nicht-funktionalen Anforderungen werden in Tabelle 3.1 kurz und knapp zusammengefasst. Diese Tabelle wird die Grundlage für die Bewertung existierender Managementlösungen im nächsten Kapitel bilden. Des Weiteren wird sie der abschließenden Bewertung des Konzepts und der Implementierung dienen.

Benutzerverwaltung	Passende Benutzergruppen werden unterstützt
Bestell- und Bereitstellungs-vorgang	Bestellmöglichkeit besteht Parameter frei wählbar Vorlagen / fertige Angebote verfügbar Einzelne Parameter bei Vorlagen modifizierbar AGBs werden angezeigt Preis wird angezeigt Administrator kann Antrag ablehnen Administrator kann Alternative vorschlagen Vor Bereitstellung ist manueller Eingriff möglich Benachrichtigungen werden verschickt
Betrieb	Steuerung des Betriebs möglich VMs können gestartet/gestoppt/resetet werden Snapshots können angelegt/verwaltet werden Mounten von Nutzer-ISO-Images möglich Status-/Performace-Daten sind abrufbar Fertige VMs können modifiziert werden Zugriff auf die Konsole ist möglich Löschungen können beantragt werden
Ressourcepools	Ressourcepools werden unterstützt VMs können geklont werden Verbrauchte Ressourcen werden angezeigt
Nicht-funktionale Anforderungen	System kann einfach angepasst werden Bisherige Rechteverwaltung wird verwendet Schnittstellen/APIs sind verfügbar System ist skalierbar System kann hochverfügbar aufgesetzt werden
Virtualisierer- unterstützung	vSphere wird unterstützt ESXi-Server werden unterstützt Anderer Virtualisierer werden unterstützt

Tabelle 3.1.: Anforderungskatalog

3. Anforderungsanalyse

4. State of the Art

Vor der Entwicklung einer neuen Software soll geprüft werden, ob es bereits eine existierende Lösung für das Problem gibt oder ob es Open-Source-Software gibt, auf der man aufbauen kann. Zunächst kann man die existierenden Lösungen in drei Kategorien einteilen: CLIs (Command Line Interfaces), Desktop-Anwendungen und Webanwendungen.

CLIs sind als Managementoberfläche für Endkunden natürlich denkbar ungeeignet, schließlich kann man bei diesen keinerlei Erfahrung im Umgang mit der Kommandozeile erwarten. Denkbar wäre aber der Aufbau einer grafischen Oberfläche auf einem existierenden CLI.

Desktop-Anwendungen sind für diese Arbeit naturgemäß weniger interessant, da die angestrebte Lösung eine Webanwendung sein soll. Dennoch wäre es nicht sinnvoll ein Webportal zu entwickeln, wenn sämtliche Anforderungen durch eine Desktop-Anwendung erfüllt wären und deren einziger Nachteil die Notwendigkeit der Installation wäre. Somit sollen auch diese in der folgenden Übersicht berücksichtigt werden.

Webanwendungen entsprechen schließlich der angestrebten Anwendungsart.

In den folgenden Unterkapiteln soll zunächst libvirt vorgestellt werden, welches als Grundlage für viele Anwendungen dient. Anschließend wird in einem groben Überblick die Fülle der bestehenden Managementlösungen aufgezeigt. Diese Betrachtung findet ohne Festlegung auf konkrete zu unterstützende Virtualisierungslösungen statt. Aus dieser Übersicht werden die drei aussichtsreichsten webbasierten Managementlösungen herausgegriffen und einer tieferen Analyse unterzogen, in welcher geprüft wird, inwiefern sie die im letzten Kapitel aufgestellten Anforderungen erfüllen.

4.1. Einschub: libvirt

Beim bereits erwähnten libvirt (<http://libvirt.org/>), welches zuerst betrachtet werden soll, da viele Anwendungen darauf aufbauen, handelt es sich um eine C-Bibliothek, welche eine API zur Steuerung diverser Virtualisierungslösungen bietet. Für andere Programmiersprachen werden Bindings angeboten, so zum Beispiel für C#, PHP, Perl, Python und andere. Unterstützt werden unter anderem Xen, QEMU, KVM, OpenVZ, VirtualBox und andere. In den neueren Versionen wird neben der Unterstützung von ESXi-Servern auch die Steuerung eines vCenters angeboten. Im Falle der Steuerung einer vSphere-Infrastruktur greift libvirt direkt auf die von VMware angebotene API zurück. Im Falle anderer Virtualisierer muss auf den entsprechenden Hosts zusätzlich ein Agent installiert werden, welcher die Kommandos von der auf libvirt aufsetzenden Anwendung entgegen nimmt (siehe Abbildung 4.1).

Es gibt eine große Anzahl an Management-Anwendungen, die auf libvirt aufsetzen. Diese konzentrieren sich oft entweder auf einen sehr beschränkten Aufgabenbereich (z.B. libguestfs zum Bearbeiten von VM-Dateisystemen) oder auf die Steuerung nur eines einzelnen oder weniger Hypervisoren. Diese Anwendungen finden sich in allen zuvor genannten Kategorien. Bei einer libvirt-Installation mitgeliefert wird die Kommandozeilenanwendung virsh. Dies ist

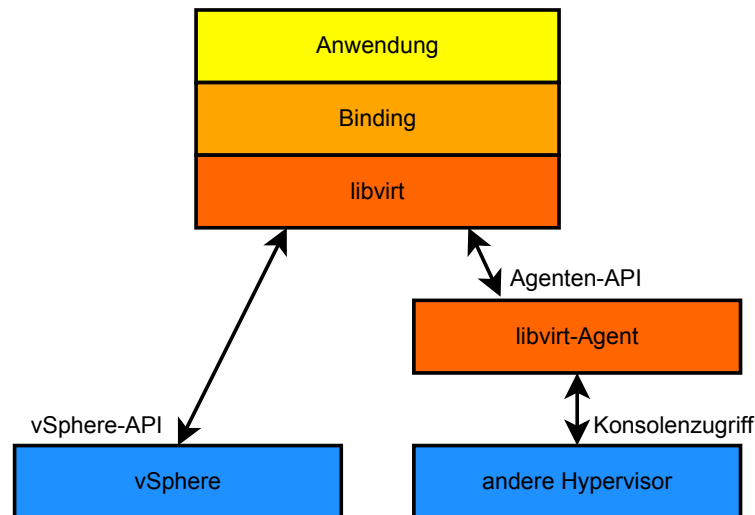


Abbildung 4.1.: Steuerung einer Virtualisierungslösung mit Hilfe von libvirt

zudem die einzige Anwendung die für alle von libvirt unterstützten Virtualisierungslösungen angelegt ist.

Da im Rahmen dieser Arbeit auch eine Implementierung des Webportals für das LRZ steht, ist es erforderlich, dass die Management-Software auch die vSphere-Infrastruktur unterstützt, doch neben virsh gibt es keine Anwendung die eine umfassende Steuerung dieser ermöglicht. Doch auch der Einsatz von virsh oder die Entwicklung einer eigenen Anwendung unter Nutzung von libvirt ist nicht möglich. Die Unterstützung des vCenters ist noch in einer sehr frühen Phase. So ist es zum Beispiel nicht möglich übergreifende Daten über alle durch das vCenter gesteuerte Hosts abzurufen. Bei einer Anfrage muss immer ein konkreter Host genannt werden. [LVV] Möglicherweise ließe sich dieses Problem noch durch eigene Erweiterungen umgehen, viel schwerwiegender ist jedoch die Tatsache, dass libvirt einen root-Zugriff auf den Virtualisierer benötigt. Mit beschränkten Nutzerrechten führen bereits simple Statusabfragen zu einem Abbruch mit einer Fehlermeldung. Denkbar wäre die Zugriffskontrolle in das Webportal zu verlagern und dieser selbst einen root-Zugriff auf die Infrastruktur zu gewähren. Das setzt aber voraus, dass das Webportal die einzige Zugriffsmöglichkeit auf den Virtualisierer darstellt. Dies widerspricht aber der Anforderung, dass das Webportal die bereits existierende Nutzerverwaltung nutzen soll, um weiterhin die Nutzung sich bereits im Einsatz befindender Management-Anwendungen zu ermöglichen. Des Weiteren würden sich weitere sicherheitskritische Probleme zum Beispiel beim Zugriff auf die Konsole ergeben, bei welcher der Client direkt auf den ESXi-Server zugreifen muss. Es zeigt sich also, dass auf einem Einsatz von libvirt in Verbindung mit vSphere in der jetzigen Entwicklungsphase verzichtet werden muss. Der Einsatz zu einem späteren Zeitpunkt lässt sich aber nicht ganz ausschließen und sollte berücksichtigt werden.

4.2. Überblick über bestehende Managementlösungen

Die Informationen, die in diesem Überblick vermittelt werden, entstammen im Wesentlichen den Produktbeschreibungen auf den Webseiten der einzelnen Lösungen. Die Auswahl selbst basiert auf einer längeren Recherche im Internet und sollte einen Großteil der verfügba-

ren Lösungen abdecken. Projekte, bei denen die Entwicklung erst am Anfang steht und noch keine fertige Software angeboten wird, wurden nicht berücksichtigt. Die drei aussichtsreichsten webbasierten Anwendungen aus diesem Unterkapitel werden anschließend genauer untersucht, wozu diese auch installiert und ausprobiert werden.

Im Folgenden soll der Blick zunächst auf existierende CLIs geworfen werden. Virsh wurde bereits im Rahmen von libvirt angesprochen. Für vSphere muss natürlich auch die offizielle vSphere CLI von VMware angesprochen werden. Damit lassen sich natürlich sämtliche Funktionen von vSphere problemlos nutzen, doch ist dieses CLI, wie der Name schon andeutet, auf vSphere begrenzt. Solche spezialisierten CLIs lassen sich für die meisten Virtualisierer finden. So gibt es zum Beispiel für VirtualBox VBoxManage und für Xen xe, sm und xm. Interessanter sind CLIs über die sich mehrere Virtualisierer steuern lassen, doch ist die Auswahl an nicht auf libvirt aufbauenden Lösungen sehr begrenzt. Ein Beispiel wäre Ganeti, welches Xen und KVM unterstützt. [GAN] Eine Lösung die auch das vCenter unterstützt gibt es nicht.

Breiter ist die Auswahl bei den Desktop-Anwendungen. Es gibt unter anderem:

Virtual Machine Manager Dies ist eine libvirt-basierte Anwendung für das Management von Xen, QEMU und KVM. Neben den üblichen Funktionen zum Betrieb bietet dieses Programm unter anderem grafische Performance-Statistiken und die Provisionierung von Gast-Betriebssystemen mittels Templates. [VMM]

Gnome KVM Manager (GKVM) Dies ist eine sehr einfach gestaltete GUI für KVM. Dieses Programm bietet nur die wichtigsten Funktionen zur Erstellung, Modifikation und Ausführung von VMs. Seit 2007 gab es allerdings keine neuen Versionen. [GKV]

AQEMU Diese Anwendung ist eine GUI für QEMU und KVM. Sie unterstützt die meisten Einstellungsoptionen dieser Virtualisierer und wird auch weiterentwickelt. [AQE]

Virtualbricks Dies ist eine Management-Anwendung für VDE, QEMU und KVM. Unterstützt wird neben dem Betrieb von VMs auch die Erstellung komplexer virtueller Netze. [VBR]

SYSVM Dies ist eine kostenpflichtige Lösung für OpenVZ und Xen. Der Funktionsumfang geht weit über die üblichen Basisfunktionen hinaus. Unter anderem gibt es ein Billing-Modul und einen Dateimanager. [SYS]

VMware Powerpack for PowerGUI Bei PowerGUI handelt es sich um einen grafischen Editor für die Windows PowerShell. Das VMware Powerpack stellt über diesen umfangreiche Management-Möglichkeiten für ESXi-Server oder ganze vSphere-Infrastrukturen bereit. [Mun]

Unter den Webanwendungen gibt es die meiste Auswahl. Es gibt unter anderem die folgenden Produkte:

Univention Virtual Machine Manager (UVMM) Diese von der gleichnamigen Firma angebotene Lösung für Xen und KVM ist Teil der größeren ITSM-Software Univention Corporate Server. Der UVMM selbst ist libvirt-basiert und beschränkt sich wie die meisten Lösungen auf den Betrieb. [UVM]

4. State of the Art

ConVirt Diese Anwendung von Convirture für Xen und KVM wird in drei Versionen angeboten, einer kostenlosen Open-Source-Variante und zwei kostenpflichtigen Varianten mit größerem Funktionsumfang. Neben Standardaufgaben unterstützt ConVirt High Availability-Funktionen, VLANs, Live Migrationen und Backups. [CON]

Karesansui Diese, über eine REST-basierte Architektur verfügende, japanische Software nutzt libvirt und unterstützt Xen und KVM. Die Unterstützung weiterer Virtualisierungslösungen ist geplant. [KAR]

openQRM Auch dies ist wieder eine Lösung die in einer kostenlosen und einer kostenpflichtigen Variante angeboten wird. Bemerkenswert ist einerseits die große Anzahl an unterstützten Virtualisierern, nämlich VMwares ESXi-Server, Xen, KVM, lxc Linux Containers und OpenVZ, sowie ein Plugin welches Beantragungen und Provisionierungen ermöglicht. [OQR]

Abiquo Diese Lösung, ebenfalls in zwei Versionen verfügbar, unterstützt als einzige der hier vorgestellten Lösungen vSphere und darüber hinaus auch Hyper-V, Virtual Box, Xen und KVM. [ABI]

Cloudmin Diese Software basiert auf dem weit verbreiteten Webmin, welches zum webbasierten Management von Unix-Servern dient. Unterstützt wird mit Xen, OpenVZ, KVM, VServers und Solaris Zones ebenfalls eine Vielzahl von Virtualisierern. [CLO]

Proxmox Virtual Environment Dies ist eine Open-Source-Lösung der Proxmox Server Solutions GmbH für Xen und KVM. Der Funktionsumfang ist mit dem der bisher erwähnten umfangreicheren Lösungen vergleichbar. [PRO]

Es gibt auch Komplettpakete, welche vergleichbar mit vSphere, sowohl den Hypervisor mit einem Bare-Metal-Installer als auch die Managementplattform integrieren. Hierzu gehören:

Parallels Virtuozzo Containers Als vServer für den Privatkundenmarkt populär wurden, war es diese Software, auf der die meisten, vor allem kleineren, Anbieter aufgebaut haben. Als Virtualisierer wird OpenVZ verwendet. [PVC]

OpenNode Cloud Platform Dieses Komplettpaket basiert auf OpenVZ und KVM und verwendet libvirt. Für den Bare-Metal-Installer kommt CentOS/RHEL zum Einsatz. [ONO]

Abschließend kann man feststellen, dass es eine breite Auswahl an diversen Management-Anwendungen gibt. Viele dieser Produkte können auch mehr als eine Virtualisierungslösung managen, wobei gerade die Unterstützung für VMwares Produkte selten ist. Noch seltener kommt es vor, dass auch das vCenter unterstützt wird. Sämtliche libvirt-basierten Lösungen, von denen es in der Tat eine große Anzahl gibt, fallen wegen den bereits genannten Beschränkungen aus.

Natürlich unterstützen diese Lösungen nicht alle virtualisiererspezifischen Funktionen, wie die mitgelieferten Anwendungen, doch diese werden für die definierte Aufgabenstellung auch nicht benötigt. Zumal die Software eines Drittanbieters kaum mit den immer wieder neu entwickelten Funktionen eines Virtualisierers mithalten könnte. Das Webportal soll vor allem den alltäglichen Serverbetrieb und die Interaktion mit den Kunden unterstützen. Für

kompliziertere administrative Eingriffe, wie zum Beispiel eine Live-Migration stehen weiterhin die virtualisiererspezifischen Lösungen zur Verfügung.

Für die genauere Untersuchung, bei welcher die Software installiert, ausprobiert und mit den Anforderungen abgeglichen wird, wurden die folgenden drei Produkte ausgewählt:

- Cloudmin, da es mit Webmin eine seit vielen Jahren entwickelte und weit verbreitete Software zum Management von Unix-Systemen als Basis hat,
- Abiquo, da dies die einzige Lösung ist, die vSphere unterstützt und
- openQRM, da dies die einzige Management-Anwendung ist, die außer dem Betrieb weitere große Bereiche des Dienstlebenszykluses abdeckt.

4.3. Cloudmin

Cloudmin ist eine Erweiterung von Webmin. Bei Webmin handelt es sich um eine webbasierte Managementlösung zur Verwaltung von Unix-Systemen. Cloudmin ermöglicht, auf der Funktionalität von Webmin aufbauend, die Verwaltung von virtuellen Maschinen, wobei diverse Virtualisierer — auch parallel — eingesetzt werden können. Die kostenlose GPL-Version erlaubt lediglich den Einsatz von Xen und KVM, die kommerzielle Pro-Version diverse weitere bereits in der Übersicht aufgezählte Virtualisierer. Das Rechtemodell von Cloudmin baut primär auf Ressourcepool-Nutzern auf. Der Administrator erstellt Benutzerkonten denen er eine bestimmte Menge an Ressourcen, zum Beispiel die maximale Anzahl an VMs oder die Größe des verfügbaren Speicherplatzes, und feingranular einzelne Rechte, wie etwa das Recht neue Maschinen anzulegen oder die Konsole aufzurufen, zuweisen kann. Dabei kann man die Ressourcen und Rechte auf einzelne der zentral verwalteten Hosts beschränken. Bekommt der Nutzer das Recht, VMs auf allen Hosts anzulegen, kann das System beim Anlegen neuer VMs automatisch den Host auswählen, der die meisten freien Ressourcen hat. Das Anlegen eines Nutzers ist zwar immer mit der Zuweisung eines Ressourcepools verknüpft, doch könnte man über eine entsprechende Rechtevergabe (kein Recht VMs anzulegen oder zu modifizieren) auch Endnutzern Zugang zum System ermöglichen, damit diese ihre VMs verwalten (also zum Beispiel starten oder stoppen) können. Bei der Erstellung von VMs ist es möglich auf fertige System-Images zurückzugreifen. Es ist möglich über Cloudmin auf eine Textkonsole und eine VNC-basierte grafische Konsole zuzugreifen. Es können diverse Statistiken und Graphen zum Ressourcenverbrauch eingesehen werden. IP-Adressen werden zentral verwaltet und über DHCP an die VMs zugeteilt, des weiteren wird der Hostname der VMs automatisch in eine DNS-Zone eingetragen. Zudem gibt es die Möglichkeit für die VMs ein regelmäßiges Backup einzurichten. Auf sämtliche Funktionen kann über eine API zugegriffen werden.

Zusammenfassend erscheint Cloudmin für den Betrieb von VMs eine gute Lösung zu sein, zumal man diese, auch Dank der Möglichkeit zur Replikation, ausfallsicher aufstellen kann. Da es sich um Open-Source-Software handelt, ist auch die Anpassung an eigene Bedürfnisse einfach. Der gesamte Dienstlebenszyklus wird durch Cloudmin allerdings nicht abgedeckt. Cloudmin bietet keine Möglichkeit Bestellung aufzugeben oder Maschinen zu kündigen. Es finden sich auch keinerlei sonstige Möglichkeiten zur Interaktion zwischen Endnutzer und Administrator. Außerdem wird vSphere nicht unterstützt was einen Einsatz am LRZ unmöglich macht.

4.4. **Abiquo**

Abiquo, welches von der gleichnamigen Firma entwickelt wird, ist in zwei Versionen verfügbar: in der unter der LGPL veröffentlichten Community Edition und der kostenpflichtigen Enterprise Edition mit erweitertem Funktionsumfang. Abiquo hat den größten Funktionsumfang der drei vorgestellten Lösungen und ist entsprechend so komplex aufgebaut, dass hier nur auf einzelne Aspekte eingegangen werden kann. Abiquo unterstützt neben der bereits aufgeführten großen Anzahl an Virtualisierern verschiedene Speichersysteme, wie LVM und iSCSI Linux, ZFS OpenSolaris und Nexenta. Die Zuteilung von IP-Adressen erfolgt über eine Anbindung an den ISC DHCP-Server. Das System deckt alle Basisfunktionalitäten, die zum Betrieb von VMs benötigt werden ab. Es gibt eine starke Ausrichtung auf die Provisionierung von virtuellen Appliances. Es existiert die Möglichkeit VLANs einzurichten. Außerdem ist es möglich auf komplexen Infrastrukturen aufzusetzen. So lassen sich zum Beispiel VMs in mehreren physikalischen Rechenzentren zentral steuern. Sehr spezifisch ist die Rechteverwaltung, Abiquo setzt auf drei vordefinierten Rollen auf: Cloud Admins, Enterprise Admins und Users. Sowohl Enterprise Admins als auch User verfügen allerdings über einen Resourcepool und können selbst VMs erstellen und löschen. Einfache Endnutzer, die VMs nur steuern sollen, sind nicht vorgesehen. Das System ist in erster Linie auf das Management von Enterprise-Clouds ausgerichtet und für den Endkundenmarkt weniger geeignet. Folglich gibt es auch keinerlei Funktionalität, die über den Betrieb hinausgeht.

4.5. **openQRM**

Beim ebenfalls appliance-basierten openQRM, welches von openQRM Enterprise entwickelt wird, gibt es genauso wie bei Abiquo eine kostenlose Open-Source-Version und eine kostenpflichtige Enterprise-Version, wobei die Unterschiede zwischen den Versionen in diesem Fall deutlich geringer ausfallen. OpenQRM kommt den geforderten Anforderungen von den getesteten Systemen am nächsten. OpenQRM ist modular aufgebaut, verschiedene Funktionen, wie die Unterstützung verschiedener Speicherlösungen oder diverser Virtualisierungslösungen, sind in Plugins ausgelagert. Besonders interessant ist das Cloud-Plugin, welches ein vollautomatisiertes Beantragungs- und Provisionierungssystem zur Verfügung stellt. Über eine spezielle Benutzeroberfläche können sich die Endnutzer registrieren und anschließend VMs beantragen. Dabei ist es möglich sich die VM beliebig zusammenzustellen. Die zur Verfügung stehenden Optionen für die einzelnen Parameter (z.B. RAM, Kernel, Virtualisierungslösung) werden vorher vom Administrator festgelegt. Während der Zusammenstellung der VM werden dem Benutzer sofort die anfallenden Kosten angezeigt. Die Bezahlung erfolgt über sogenannte CC-Units, die der Benutzer zuvor außerhalb des Systems erwerben muss. Bei der Bestellung muss der Endnutzer auch Zeitpunkte angeben zu denen die VM zur Verfügung gestellt werden soll und ab wann sie nicht mehr benötigt wird. Ein solcher Antrag kann anschließend von einem Administrator bestätigt oder abgelehnt werden. Bei einer Bestätigung wird die VM vollautomatisch provisioniert. Der Endbenutzer kann seine VM auch vorzeitig selbstständig deprovisionieren. Resourcepools sind im Rahmen des Cloud-Plugins nicht vorgesehen. IP-Adressen können wie bei den bisherigen Lösungen über DHCP verwaltet werden und können auch im DNS eingetragen werden. Ebenfalls interessant ist ein Plugin, über welches ein Monitoring auf Nagios-Basis realisiert wird. Mit openQRM können auch physische Systeme verwaltet werden. Auch eine Migration eines physischen Systems auf

ein virtuelles und umgekehrt ist möglich. Des Weiteren sind private System-Images möglich, also Images die nur einem Nutzer zur Verfügung stehen. In neueren Versionen von openQRM ist neben einer Text-Konsole auch eine VNC-basierte grafische Konsole verfügbar.

Besonders ist bei openQRM die Einbindung neuer Hosts. Zwar ist es möglich auf bereits existierenden Hosts eine Clientanwendung zu installieren, doch um alle Vorteile von openQRM ausnutzen zu können, müssen die Hosts über PXE booten. Dabei registrieren sie sich automatisch am openQRM-Server und können von dort für diverse Zwecke konfiguriert werden.

4.6. Bewertung der vorgestellten Anwendungen im Kontext dieser Arbeit

An dieser Stelle soll die Eignung der drei im Detail untersuchten Managementlösungen festgestellt werden. Hierzu wird den Anwendungen der Anforderungskatalog gegenübergestellt und geprüft, welche Anforderungen erfüllt werden. Die Tabelle 4.1 zeigt das Ergebnis dieser Analyse. Klammern bedeuten, dass diese Funktion nur sehr eingeschränkt beziehungsweise auf eine sehr spezifische Art und Weise zur Verfügung steht.

Abiquo unterstützt als einzige der aufgezählten Lösungen neben der Anbindung an ESX- und ESXi-Servern auch das vCenter. Dieses ist aber zugleich das ungeeignetste der getesteten Systeme, da es klar auf den Enterprise-Bereich ausgerichtet ist und deshalb ein völlig ungeeignetes Rechtesystem implementiert, bei dem es unter anderem keine gewöhnlichen Endnutzer gibt, die keine VMs anlegen können. Cloudmin bietet eine zufriedenstellende Lösung für den Betrieb, unterstützt aber genauso wie Abiquo den sonstigen Dienstlebenszyklus nicht. In dieser Hinsicht hat nur openQRM etwas zu bieten. Mit dem Cloud-Plugin kommt es der gewünschten Funktionalität bereits recht nahe, wenn auch noch diverse Anforderungen nicht erfüllt sind. So entspricht der Beantragungsvorgang an diversen Stellen nicht den Anforderungen, es werden zum Beispiel keine Vorlagen unterstützt, AGBs werden nicht angezeigt und der Administrator kann bei einer Ablehnung keine Alternative vorschlagen. Auf ein vCenter kann zudem auch nicht zugegriffen werden.

Die meisten nicht-funktionalen Anforderungen werden von allen drei Anwendungen erfüllt, allerdings ist ihnen in diesem Bereich auch ein großes Defizit gemein: alle bauen auf einem eigenem internem Rechtesystem auf und nutzen nicht das vorhandene. Das bedeutet unter anderem, dass man bei allen diesen Produkten vollständig auf die neue Lösung setzen müsste. Bisherige Management-Anwendungen ließen sich nicht weiter verwenden, eine fließende Einführung wäre nicht möglich.

Es zeigt sich also, dass ein Webportal, welches den ganzen Dienstlebenszyklus abbildet und eine Steuerung der vSphere-Infrastruktur ermöglicht, welches sowohl für Administratoren, wie auch Endnutzer gleichermaßen geeignet ist, welches für weitere Virtualisierer erweiterbar ist und welches sich problemlos in die bestehende Infrastruktur integrieren und auf diese anpassen lässt, erst entwickelt werden muss.

4. State of the Art

		Cloudmin	Abiquo	openQRM
Benutzerverwaltung	Passende Benutzergruppen werden unterstützt	(✓)		(✓)
Bestell- und Bereitstellungs- vorgang	Bestellmöglichkeit besteht			✓
	Parameter frei wählbar			✓
	Vorlagen / fertige Angebote verfügbar			
	Einzelne Parameter bei Vorlagen modifizierbar			
	AGBs werden angezeigt			
	Preis wird angezeigt			✓
	Administrator kann Antrag ablehnen			✓
Betrieb	Administrator kann Alternative vorschlagen			
	Vor Bereitstellung ist manueller Eingriff möglich			
	Benachrichtigungen werden verschickt			✓
	Steuerung des Betriebs möglich	✓	✓	✓
	VMs können gestartet/gestoppt/resetet werden	✓	✓	✓
	Snapshots können angelegt/verwaltet werden		(✓)	✓
	Mounten von Nutzer-ISO-Images möglich	✓	✓	(✓)
	Status-/Performance-Daten sind abrufbar	✓	✓	✓
Fertige VMs können modifiziert werden	✓	✓	✓	
Ressourcepools	Zugriff auf die Konsole ist möglich	✓	✓	✓
	Löschungen können beantragt werden			✓
	Ressourcepools werden unterstützt	✓	✓	
Nicht-funktionale Anforderungen	VMs können geklont werden	✓	(✓)	
	Verbrauchte Ressourcen werden angezeigt	✓	✓	
	System kann einfach angepasst werden	✓	✓	✓
	Bisherige Rechteverwaltung wird verwendet	✓	✓	✓
Virtualisierer- unterstützung	Schnittstellen/APIs sind verfügbar	✓	✓	✓
	System ist skalierbar	✓	✓	✓
	System kann hochverfügbar aufgesetzt werden	✓	✓	✓
	vSphere wird unterstützt		✓	
Anderer Virtualisierer werden unterstützt	ESXi-Server werden unterstützt		✓	✓
		✓	✓	✓

Tabelle 4.1.: Anforderungserfüllung durch die drei getesteten webbasierten Management-Anwendungen

5. Konzept

Nachdem im vorherigen Kapitel festgestellt wurde, dass die existierenden Management-Anwendungen die aufgestellten Anforderungen nicht erfüllen, wird in diesem Kapitel das Konzept für eine Lösung aufgestellt, welche diesen nachkommt. Die grundsätzliche Idee hinter dem Webportal ist dabei, dass alle Nutzergruppen, also Administratoren, Resourcepool-Nutzer und Endnutzer über eine gemeinsame GUI sowohl untereinander interagieren als auch die virtuelle Infrastruktur steuern. Damit das Portal selbst generisch ist, wird dieses über einen Adapter, der eine Übersetzung der Kommandos vornimmt, an den Virtualisierer angeschlossen. Da auf die bisherige Rechteverwaltung aufgesetzt werden soll, können die bisherigen Management-Tools von allen oder auch nur von einer Teilgruppe weiter genutzt werden. Das Webportal muss also nicht den kompletten Funktionsumfang der bisherigen Anwendungen umsetzen. Abbildung 5.1 veranschaulicht diese grundsätzliche Idee. Diese soll in diesem Kapitel im Folgenden weiter verfeinert werden. Dazu werden zunächst die einzelnen Komponenten ermittelt, die im gesamten System eine Rolle spielen, deren Schnittstellen untereinander definiert und schließlich ein Informationsmodell aufgestellt, auf welchem das Webportal aufbauen kann.

5.1. Komponenten der Gesamtarchitektur

Zunächst muss ein Gesamtüberblick über die Komponenten geschaffen werden, mit denen das Webportal interagieren muss. In einem ersten Schritt werden die Komponenten aus den Anforderungen abgeleitet. Diese Komponenten sind zum Betrieb des Webportals zwingend erforderlich. Der Grund, dass einige in der folgenden Aufzählung dennoch als optional gekennzeichnet sind liegt darin, dass die Funktionalität von einer anderen Komponente übernommen werden kann. Anschließend werden weitere optionale Komponenten beschrieben, die sich nicht direkt aus den Anforderungen ergeben, aber dennoch häufig vorhanden sind und berücksichtigt werden müssen.

Virtualisierungslösung mit virtueller Infrastruktur Das Portal soll eine virtuelle Infrastruktur managen. Diese virtuelle Infrastruktur wird von einer Virtualisierungslösung bereitgestellt. Beides wird für dieses Konzept zu einer abstrakten Komponente zusammengefasst. Diese enthält somit sowohl die Hypervisoren und die zentrale Steuerkomponente, als auch die darüber gesteuerten und betriebenen VMs. In den Abbildungen wird diese Komponente verkürzt als “virtuelle Infrastruktur” bezeichnet.

Authentication and Authorization Infrastructure (optional) Aus den Anforderungen ergibt sich, dass das Portal mehrere Nutzer unterstützen soll. Diese Nutzer lassen sich einerseits in die Administratoren und andererseits in die Endnutzer unterteilen. Unter letzteren findet sich zudem die Gruppe der Resourcepool-Nutzer. Der Zugriff auf das Webportal soll nur berechtigten Nutzern gewährt werden. Damit das Portal feststellen

5. Konzept

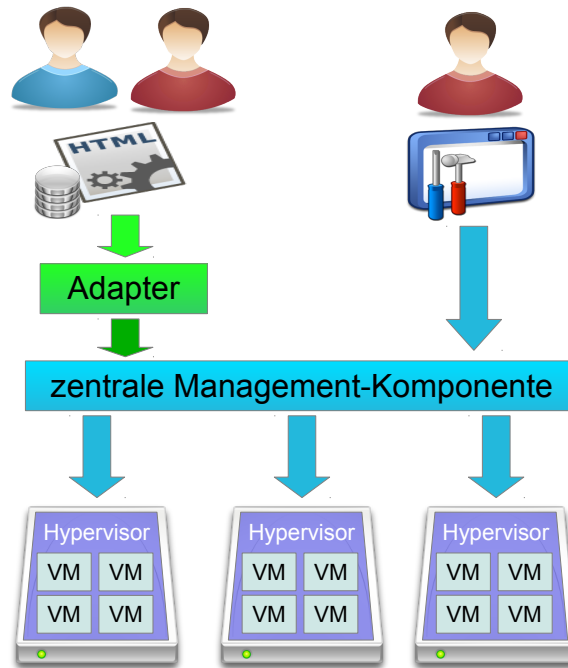


Abbildung 5.1.: Einordnung des Webportals in das bisherige Management

kann, ob ein Nutzer die Berechtigung zur Nutzung desselben hat und zu welcher Gruppe er gehört, wird eine Authentication and Authorization Infrastructure benötigt. Bei der AAI handelt es sich um ein System, über welches sich Nutzer anmelden können, um Zugriff auf bestimmte Webanwendungen mit bestimmten Berechtigungen zu erhalten. Ein Beispiel für eine AAI ist Shibboleth. Wenn die Virtualisierungslösung bereits über eine ausreichende eigene Zugriffsverwaltung verfügt, so kann auf den Einsatz einer dedizierten AAI verzichtet und auf Authentifizierungs- und Autorisierungsmechanismen dieser zurückgegriffen werden.

Benutzerdatenbank (optional) Die AAI alleine verfügt über keine Informationen bezüglich der existierenden Benutzer. Damit also die AAI einen sich einloggenden Nutzer einer bestimmten Identität zuordnen kann, muss sie auf eine Benutzerdatenbank zugreifen. In dieser sind alle Nutzer gespeichert, die an einer Institution Zugriff auf die verschiedenen IT-Systeme erhalten sollen, dies trifft auch auf die Nutzer der virtuellen Infrastruktur zu. Im einfachsten Fall kann die eingesetzte Virtualisierungslösung eine integrierte Benutzerdatenbank aufweisen. Werden aber neben der virtuellen Infrastruktur auch andere Dienste angeboten, so sollte eine dedizierte Benutzerdatenbank verwendet werden. Dabei kann es sich zum Beispiel um ActiveDirectory oder um LDAP handeln.

Notification-System Der Anforderungskatalog schreibt den Versand von Benachrichtigungen vor. Endnutzer sollen zum Beispiel benachrichtigt werden, wenn ihnen VMs zur Verfügung gestellt werden oder wenn Anträge abgelehnt werden. Werden nur selten Anträge über das System eingereicht, kann es auch sinnvoll sein Administratoren zu benachrichtigen, damit sich diese nicht regelmäßig einloggen müssen, um zu prüfen, ob

es etwas neues gibt. Zudem kam bei der Befragung der Wunsch auf Administratoren über Löschungen durch die Kunden zu benachrichtigen. Um diese Anforderungen zu erfüllen, wird ein Notification-System benötigt, welches die durch das Portal erzeugten Nachrichten zustellt. Im einfachsten Fall kann es sich hierbei um ein Mail-System handeln. Vorstellbar wären aber auch Benachrichtigungen über ein Instant-Messaging-System oder die Integration von Nachrichten an Administratoren in Ticketing-Systeme.

IP-Konfiguration (optional) Das Webportal soll die komplette Bereitstellung einer VM automatisch durchführen. Hierzu gehört auch die Zuweisung einer IP-Adresse, da ohne diese eine VM in der Regel nicht sinnvoll eingesetzt werden kann. Da die meisten Virtualisierungslösungen über keine integrierte Verwaltung von IP-Adressen verfügen, muss dies durch externe Komponenten sichergestellt werden. Die Verwaltung freier Adressen und die Zuteilung dieser zu den einzelnen VMs wird von der Webportal-Komponente übernommen. Es wird allerdings eine weitere Komponente benötigt, die diese IP-Adressen einer bootenden VM mitteilt. Normalerweise setzt man für diesen Einsatzzweck einen DHCP-Server ein. Wichtig ist dabei, dass die Entscheidung darüber, welche IP-Adresse welcher VM zugeordnet wird, nicht von DHCP-Server gefällt, sondern vom Webportal vorgegeben wird. Der eingesetzte DHCP-Server kann freilich auch anderen Systemen, als den VMs, IP-Adressen zuteilen. Für die Verwaltung dieser Adressen ist er dann selbst verantwortlich. Einige Virtualisierungslösungen haben einen DHCP-Server bereits integriert, so dass kein separater erforderlich ist.

Adapter Laut den Anforderungen soll das Portal diverse Virtualisierer unterstützen beziehungsweise sich leicht an neue Virtualisierer anpassen lassen. Wie bereits in der Einleitung zu diesem Kapitel geschrieben, wird dies durch einen Adapter realisiert. Das Webportal selbst wird generisch sein und an die Virtualisierungslösung generische Befehle senden. Damit diese von dieser verstanden werden, werden sie vom Adapter übersetzt. Bei dieser Übersetzung muss der Adapter unter Umständen die generischen Informationen um virtualisiererspezifische Parameter erweitern. Folglich müssen im Adapter auch diverse virtualisiererspezifische Einstellungen vorgenommen werden. Beispielsweise benötigt vSphere für die Bereitstellung einer VM die Angabe eines Clusters. Diese Information erhält der Adapter nicht vom Webportal, sondern muss selbst über den benötigten Wert verfügen. Bietet der Virtualisierer eine eigene Zugriffskontrolle, kann bei der Implementierung die AAI als Bestandteil des Adapters realisiert werden. Aus Gründen der Ausfallsicherheit und der Lastverteilung können auch mehrere Adapter gleichzeitig im Betrieb sein und per Zufallsverfahren vom Portal angesprochen werden.

Webportal Zuallerletzt sei die eigentlich zu entwickelnde Kernkomponente erwähnt: das Webportal. Ein großer Teil des Anforderungskatalogs umfasst diverse Details bezüglich der geforderten Antragsverwaltung. Da eine solche Antragsverwaltung nicht von der Virtualisierungslösung bereitgestellt wird, muss dieses vollständig vom Webportal übernommen werden. Das Webportal muss also einerseits entsprechend der Anforderungsbeschreibung Anträge von Endnutzern annehmen und andererseits den Administratoren eine Möglichkeit geben diese zu bearbeiten, also zu genehmigen oder abzulehnen. Eine Genehmigung geht in der Regel mit Veränderungen der virtuellen Infrastruktur einher. Diese Veränderungen muss ebenfalls das Webportal anstossen. Dieser Punkt führt auch schon in den zweiten großen Aufgabenbereich des Webportals ein, nämlich die Steuerung der virtuellen Infrastruktur. Sowohl Endnutzern als auch Administra-

5. Konzept

toren soll es möglich sein verschiedene Aktionen auf den VMs und Resourcepools auszuführen und Informationen über diese abzurufen. So soll beispielsweise über das Webportal abrufbar sein, ob eine VM läuft oder nicht und sich diese entsprechend starten oder stoppen lassen.

Die folgenden Komponenten lassen sich nicht direkt aus den Anforderungen ableiten. Einige dieser Komponenten wurden zwar bei der Nutzerbefragung am Rande erwähnt, die Notwendigkeit zu dessen Berücksichtigung ergab sich allerdings erst in weiteren vertiefenden Gesprächen mit den Administratoren am LRZ und an der LMU. Es handelt sich dabei um optionale Komponenten, die zum Betrieb der virtuellen Infrastruktur nicht notwendig sind, die aber gerade in größeren Institutionen außerhalb von Testumgebungen eine wichtige Rolle spielen.

Accounting (optional) Werden die Dienste kommerziell angeboten, so wird eine Komponente benötigt, welche für das Accounting zuständig ist. Diese Komponente stellt einerseits Preise für die diversen Produkte zur Verfügung und führt andererseits die Abrechnung der bereitgestellten Ressourcen durch.

Externe ITSM-Software (optional) Es besteht die Möglichkeit, dass bei der Institution, bei welcher das Webportal eingesetzt werden soll, bereits irgendeine Software für das IT-Service-Management eingesetzt wird. Da eine solche Software oft das zentrale Werkzeug für das Management der gesamten Infrastruktur ist, wäre es kontraproduktiv, wenn man für das Management der virtuellen Maschinen auf das Webportal, also eine andere Software mit völlig anderer Oberfläche und Funktionsweise, zugreifen müsste. Deshalb sollte eine mögliche Integration bestehender ITSM-Software berücksichtigt werden.

Configuration Management Database (optional) Bei der CMDB handelt es sich um eine zentrale Datenbank die der Konfiguration und dem Zugriff auf Configuration Items dient. Bei den Configuration Items handelt es sich um alle Objekte die in der IT verwaltet werden müssen. Genauere Erläuterung zur Thematik der CMDB findet man zum Beispiel in der IT Infrastructure Library. Damit die Datensätze in der CMDB vollständig sind, muss diese auch die vom Webportal verwalteten Items beinhalten. Optimal wäre es im Sinne der CMDB, wenn das Webportal die CMDB selbst als Datenbank für seine Informationen verwenden könnte. Da sich aber die CMDBs bei verschiedenen Institutionen stark voneinander unterscheiden, ist dies bei einer generischen Lösung nicht machbar. Zudem soll das Webportal auch dort funktionieren, wo keine CMDB zum Einsatz kommt. Damit dennoch in der CMDB die notwendigen Informationen verfügbar sind, muss eine Möglichkeit zur Synchronisation zwischen der Datenbank des Webportals und der CMDB vorgesehen werden.

Update-System (optional) Falls die Institution, welche die virtuelle Infrastruktur betreibt einen Dienstyp Attended Hosting anbietet, bei welchem auf Wunsch automatisch Updates auf den VMs eingespielt werden, so wird ein Update-System benötigt, welches diese Aufgabe übernimmt.

Backup-System (optional) Falls die Institution für die VMs irgendeine Art Backup anbieten will, sei es ein regelmäßiges Backup der gesamten VM oder ein inkrementelles Backup auf Dateiebene, so wird hierfür ein Backup-System benötigt. Einige Virtualisierungslösungen bieten integrierte Backuplösungen, die zum Beispiel snapshot-basiert arbeiten. In diesem Fall kann auf ein dediziertes Backup-System verzichtet werden.

5.2. Schnittstellen zwischen den Komponenten

Im letzten Kapitel wurden alle benötigten beziehungsweise unter Umständen zu berücksichtigenden Komponenten aufgezählt. In diesem Kapitel wird genauer auf die Interaktion zwischen diesen eingegangen. In Abbildung 5.2 ist zu sehen welche Daten zwischen den einzelnen Komponenten ausgetauscht werden. Diese Abbildung soll dabei lediglich einen Überblick verschaffen und erhebt keinen Anspruch auf Vollständigkeit. Damit dieser Informationsaustausch stattfinden kann, müssen die einzelnen Komponenten Schnittstellen bereitstellen. Abbildung 5.3 zeigt welche Komponente welche API bereitstellt. Die Bezeichnungen der APIs, die in diesem Diagramm zu finden sind, finden sich in der folgenden Auflistung wieder, in der die einzelnen Schnittstellen genauer beschrieben und spezifiziert werden.

Accounting-API Bei der Einreichung eines Antrags muss das Webportal dem Antragsteller die Kosten anzeigen, die mit seinem Antrag verbunden sind. Die Berechnung der Kosten kann zwar das Webportal an Hand einer festgelegten Formel selbst berechnen, doch sollte dies aus Konsistenzgründen idealerweise vom Accounting-System durchgeführt werden. Für die Abfrage des Preises muss das Accounting-System eine API zur Verfügung stellen. Hierfür würde sich eine einfache Webservice-API anbieten. Im Wesentlichen muss so eine API nur eine einzige Methode unterstützen. Das Portal schickt an das Accounting-System eine Auflistung sämtlicher gewünschter Ausstattungsparameter der VM beziehungsweise des Resourcepools und erhält den monatlich zu zahlenden Preis und eventuell eine Einrichtungsgebühr zurück. Wie das Accounting-System den Preis berechnet hängt vom Tarifmodell des Anbieters ab. Oft wird dabei für jeden Dienstyp ein Grundpreis herangezogen und auf diesen jeweils die Preise für die beantragten Komponenten (zum Beispiel die Größe der Festplatte) hinzuaddiert.

Notification-API Das Notification-System muss eine API zur Verfügung stellen, über welche das Webportal Benachrichtigungen einreichen kann. Laut Anforderungsanalyse wird hierfür eindeutig E-Mail bevorzugt. Entsprechend handelt es sich beim Protokoll für diese API um SMTP. Um eine Nachricht zu verschicken wird also die dem Webportal bekannte Benachrichtigungsadresse des Nutzers und der Inhalt der Nachricht benötigt.

Adapter-API Bei dieser API handelt es sich um eine Schnittstelle, an welche das Webportal vom verwendeten Virtualisierer unabhängige generische Befehle senden kann, damit diese entsprechend umgewandelt werden. Eine solche API wird im Rahmen dieser Arbeit in Kapitel 5.5 definiert.

AAI-API Über diese Schnittstelle kann das Webportal prüfen, ob ein Nutzer zur Nutzung des Systems berechtigt ist und über welche Rolle er im System verfügt. Hierfür gibt es mehrere existierende Protokolle, ein mögliches wäre RADIUS. Im einfachsten Fall wird an die AAI der Benutzername und ein Passwort übertragen. Die AAI beantwortet die Anfrage damit, ob die Authentifizierung erfolgreich war und informiert im Erfolgsfall darüber, welcher Gruppe der Nutzer angehört. Bietet der Virtualisierer eine integrierte Benutzerverwaltung, kann die AAI-Schnittstelle auch als Bestandteil der Adapter-API implementiert werden. In die in Kapitel 5.5 beschriebene Adapter-API ist eine solche Funktionalität integriert.

Benutzerdatenbank-API Über diese Schnittstelle kommuniziert die AAI mit der Benutzerdatenbank. Die AAI ermittelt über diese API die Identität eines Nutzers, der sich

5. Konzept

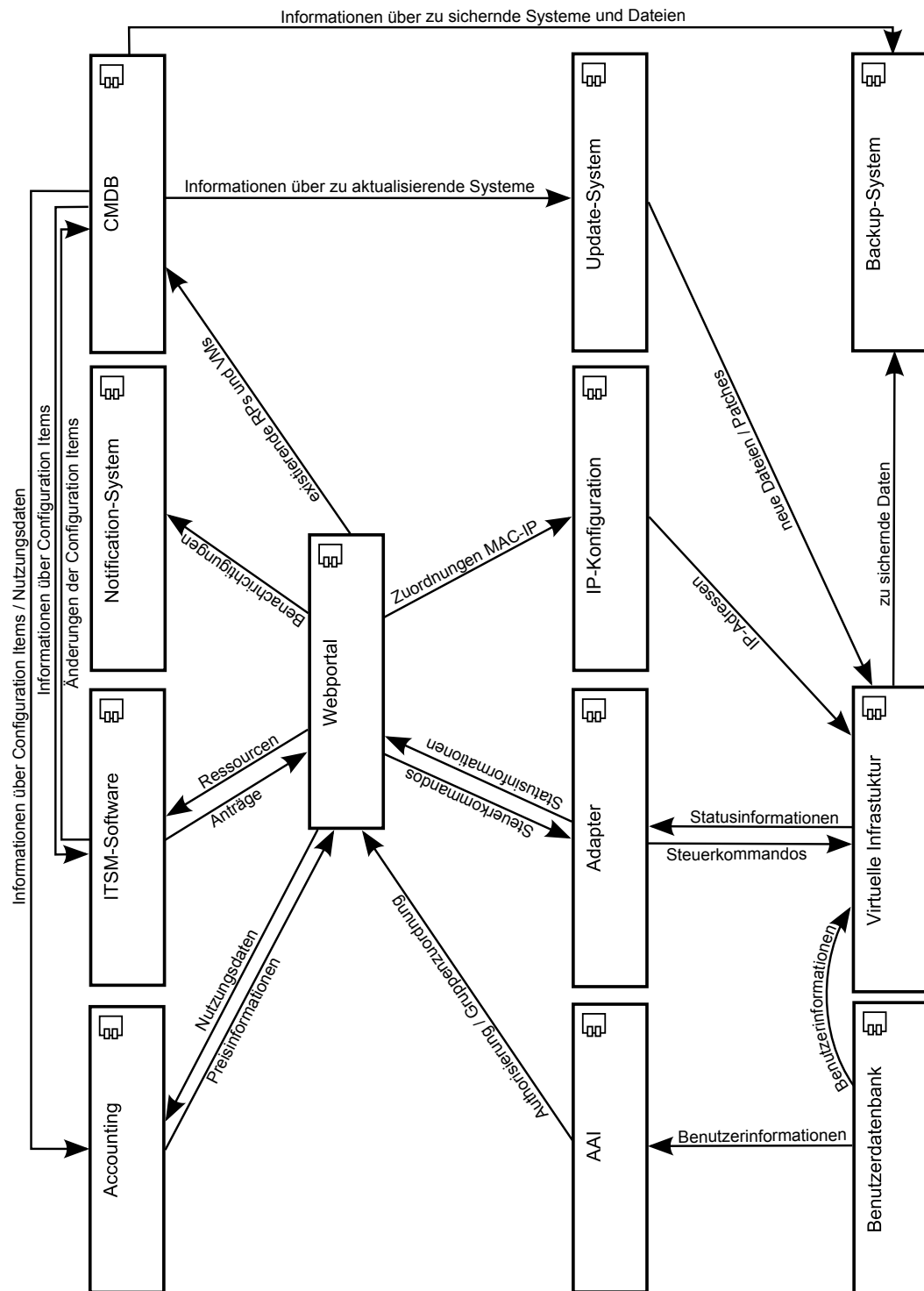


Abbildung 5.2.: Komponentendiagramm mit Richtungen des Informationsaustauschs

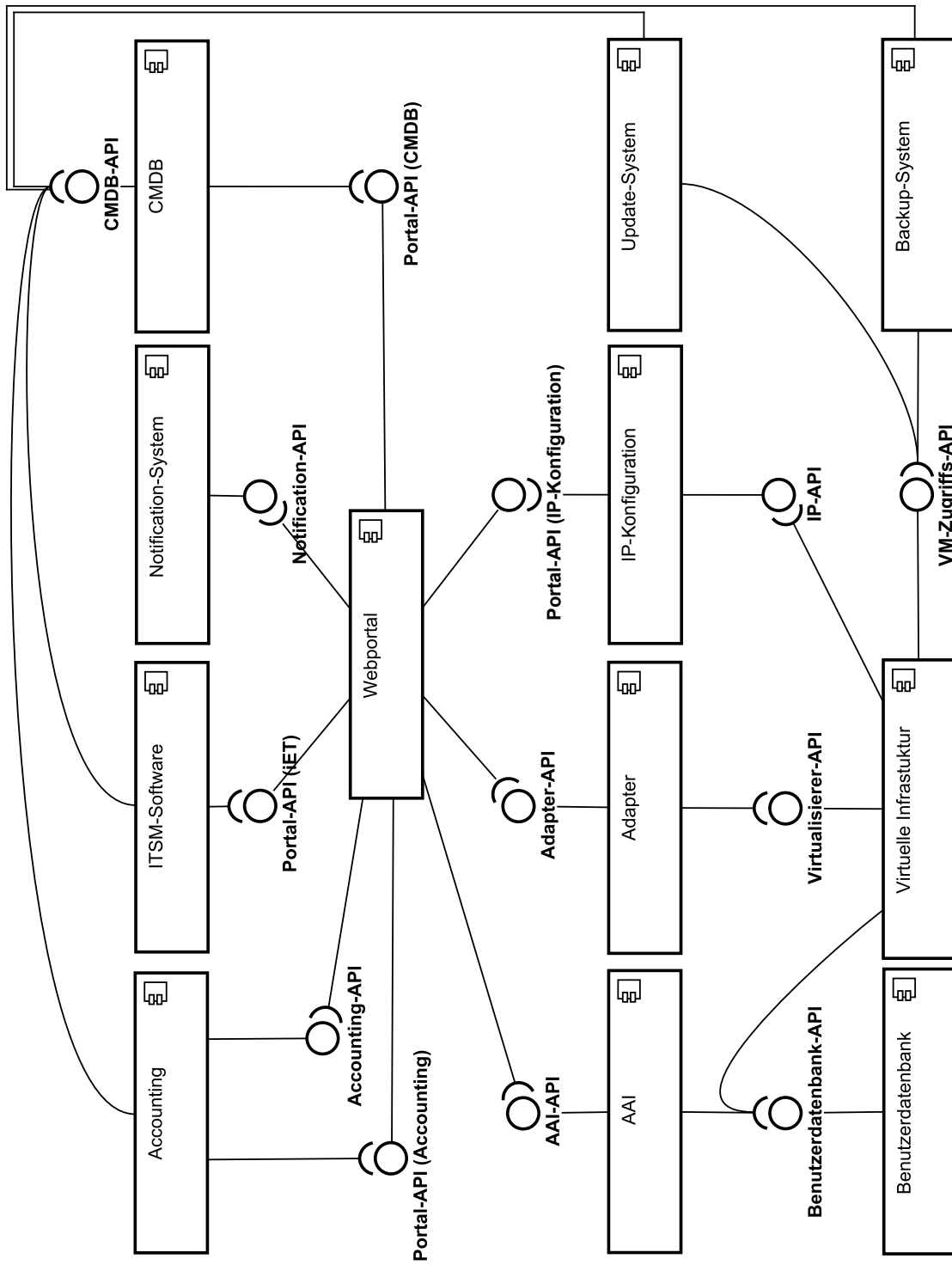


Abbildung 5.3.: Komponentendiagramm mit bereitgestellten APIs

5. Konzept

einzufragen versucht. In der Benutzerdatenbank ist auch gespeichert, welchen Gruppen ein Nutzer angehört. Ein mögliches Protokoll hierfür wäre LDAP.

Virtualisierer-API Über diese API kann die verwendete Virtualisierungslösung verwaltet werden. Diese API ermöglicht es VMs und Resourcepools anzulegen, zu modifizieren und zu löschen und VMs zu steuern, also zu starten, zu stoppen, ISO-Images zu mounten, Snapshots anzulegen und so weiter. Zudem können über diese API diverse Statusinformationen abgerufen werden, also zum Beispiel ob eine VM gerade läuft oder welche Anteile eines Resourcepools genutzt werden. Das Protokoll ist abhängig von der verwendeten Software. Beim Einsatz von VMware's vSphere wäre dies die vSphere API.

IP-API Über die IP-API fragt eine VM ab, welche IP-Adressen sie für ihre Netzwerkkarten verwenden soll. In der Regel erfolgt die Abfrage über das Dynamic Host Configuration Protocol (DHCP). Bei DHCP handelt es sich um ein Protokoll über welches IP-Adressen, sowie weitere Konfigurationsparameter in einem Subnetz verteilt werden können. Hierbei stellt eine bootende VM die Anfrage für die Zuteilung einer IP-Adresse, der DHCP-Server antwortet auf Grundlage der MAC-Adresse der VM mit der IP-Adresse, welche das Webportal der VM zugeordnet hat.

VM-Zugriffs-API Das Update-System für das Attended Hosting benötigt einen direkten Zugriff auf die VMs um neue Dateien und Patches einzuspielen. Gleiches gilt für das Backup-System, falls ein dateibasiertes (inkrementelles) Backup durchgeführt werden soll. Die Schnittstelle für einen solchen Zugriff muss vom in der VM installierten Gastsystem zur Verfügung gestellt werden. Eine Möglichkeit dies zu realisieren besteht in der Installation eines SSH-Servers auf welchen über ein spezielles Konto zugegriffen werden kann, um die entsprechenden Aktionen auszuführen. Da Updates und ein dateibasiertes Backup nur für Attended VMs angeboten werden, kann ein solcher SSH-Server bereits Bestandteil der Vorlage sein, aus welcher die Attended VMs erzeugt werden.

CMDB-API Da die CMDB bei der Verwaltung der Infrastruktur eine zentrale Rolle einnimmt, benötigen entsprechend viele Systeme Zugriff auf diese. Das Accounting muss zum Beispiel Informationen über von einem Nutzer, auch außerhalb der virtuellen Infrastruktur, verwendeten Systeme abrufen. Die ITSM-Software kann etwa den Status diverser Systeme abfragen oder Änderungen auf den Configuration Items durchführen. Das Update-System ruft aus der CMDB Informationen über die Systeme ab, die aktualisiert werden müssen. Das Backup-System benötigt Angaben darüber, für welche Systeme welche Art von Backup ansteht. Das Management-Portal muss der CMDB Informationen über existierende VMs und Resourcepools zur Verfügung stellen, selbst ruft es aber von dort keine Informationen ab. Bei der CMDB handelt es sich um ein System das oft bei der Einführung des Management-Portals bereits vorhanden ist. Die konkrete Ausprägung einer CMDB ist von Institution zu Institution sehr unterschiedlich, es macht also keinen Sinn darüber zu spekulieren, welche konkreten Informationen mit der CMDB ausgetauscht werden müssen. Die Einführung des Management-Portals in eine bestehende IT-Infrastruktur beeinflusst aber in keiner Weise die Informationen, die die genannten Komponenten mit der CMDB austauschen. Es müssen also an einer bestehenden CMDB-API auch keine Modifikationen vorgenommen werden. Als Protokoll kann im einfachsten Fall SQL verwendet werden.

Portal-API Das Portal muss APIs für verschiedene Anwendungsbereiche bieten. Das Accounting benötigt Nutzungsinformationen zu den einzelnen VMs und Ressourcepools, also Informationen darüber wann welche VM beziehungsweise welcher Ressourcepool mit welchen Parametern einem Nutzer zur Verfügung gestellt wurde, wann diese beziehungsweise dieser gegebenenfalls modifiziert oder an einen anderen Nutzer übertragen wurde und wann sie beziehungsweise er schließlich gelöscht wurde. Die ITSM-Software kann im extremsten Fall als voll funktionsfähige Alternative zur GUI des Webportals dienen und benötigt entsprechend umfassenden Zugriff auf alle Informationen und unter Umständen auch die Möglichkeit Aktionen wie das Deployment anzustoßen. Die CMDB benötigt Informationen über die aktuell existierenden VMs, um die eigenen Daten entsprechend aktualisieren zu können. Und schließlich benötigt die Komponente, welche den VMs ihre IP-Adressen mitteilt, also in der Regel der DHCP-Server, Informationen darüber, welcher MAC-Adresse welche IP-Adresse zugeordnet werden soll. Es besteht die Möglichkeit diese vier Bereiche getrennt zu betrachten und für jeden eine eigene API zu spezifizieren, ebenso können aber auch einzelne oder alle Bereiche zu einer multifunktionalen API zusammengefasst werden. Eine genauere Analyse darüber wie die einzelnen Funktionsbereiche abgedeckt werden können, findet sich im Kapitel zum Portal.

5.3. Das Portal

In diesem Kapitel wird die Komponente, um deren Entwicklung es in dieser Arbeit primär geht, genauer beschrieben. Das Webportal, welches in den vorherigen Kapiteln recht abstrakt als eine Komponente mit diversen Schnittstellen nach außen betrachtet wurde, lässt sich selbst in diverse Unterkomponenten unterteilen. Einen Überblick über diese verschafft Abbildung 5.4.

Zunächst benötigt das Webportal eine Datenbank, in welcher sämtliche benötigten Informationen zu den Anträgen und VMs gespeichert werden. Hierfür ist ein relationales Datenbanksystem angedacht. Das Datenbankschema baut dabei auf dem Informationsmodell auf, welches im nächsten Unterkapitel beschrieben wird. Die Datenbank bietet eine Schnittstelle für den Zugriff auf sich an. Dabei handelt es sich in der Regel um ODBC, das darüber gesprochene Protokoll ist SQL.

Das eigentliche Webportal ist objektorientiert aufgebaut. Für die Implementierung bietet sich eine Model-View-Controller Architektur an. Das Modell baut damit primär auf dem noch zu beschreibenden Informationsmodell auf. Zudem ist im Modell eine Komponente integriert, die für die Verwaltung der IP-Adressen zuständig ist. Das Modell ist es auch, welches die Kommunikation mit den APIs der anderen Komponenten übernimmt. Um einen Datenaustausch zwischen dem objektorientierten Modell und der relationalen Datenbank zu ermöglichen, wird eine Komponente benötigt die eine objektrelationale Abbildung durchführt. Diese wird als ORM (object-relational mapping) bezeichnet.

5.3.1. Informationsmodell des Webportals

Das Informationsmodell beschreibt Klassen für die Objekte, auf denen das Management-Portal operiert. Modelle für das Management virtueller Infrastrukturen wurden bereits oft aufgestellt. So zum Beispiel in der Diplomarbeit von Mathias Gehrung, der ein Managementkonzept für virtuelle Infrastrukturen unter besonderer Berücksichtigung unterschiedlicher

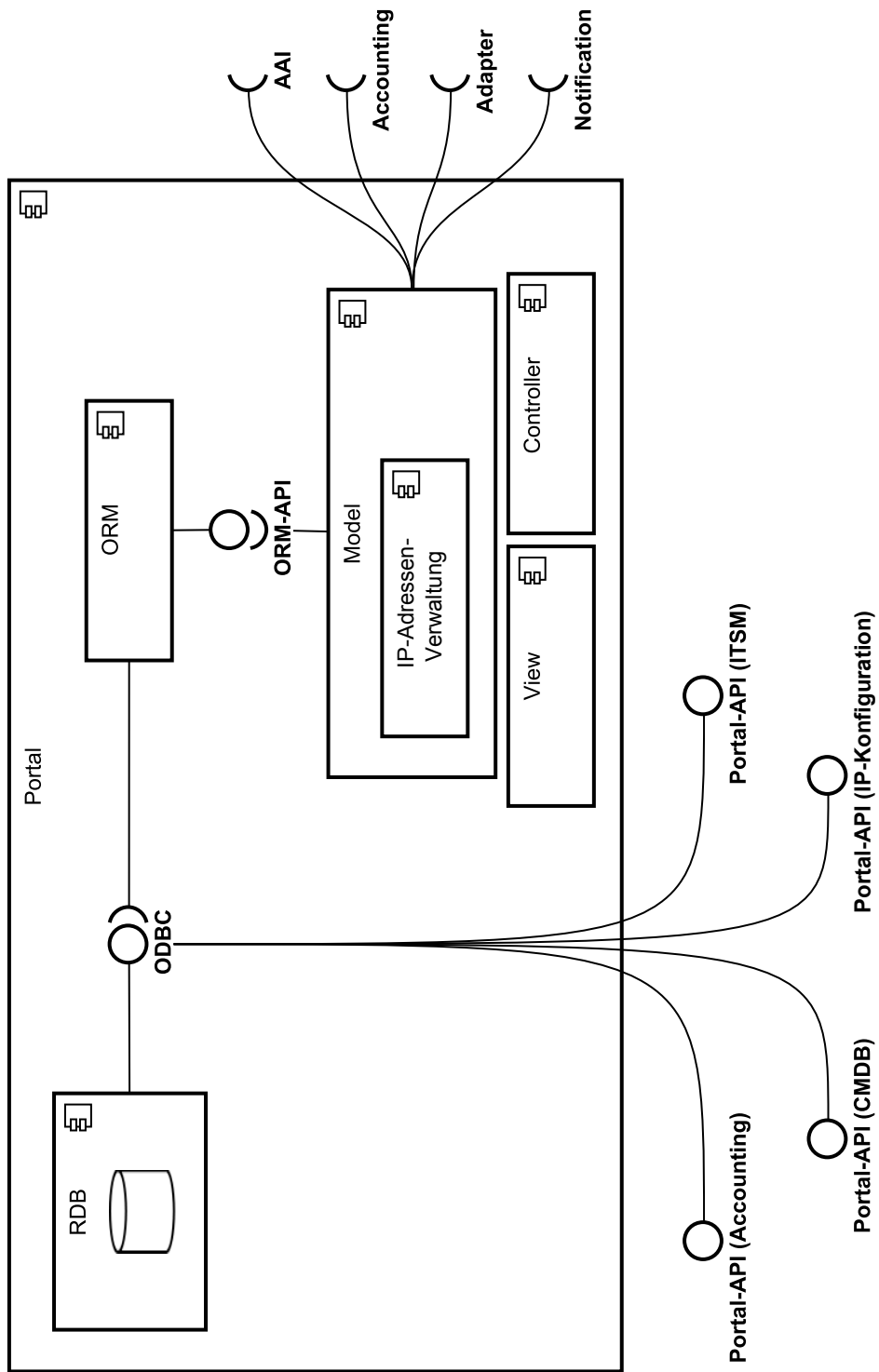


Abbildung 5.4.: Webportal-Komponente im Detail

Nutzeranforderungen aufstellt. [Geh08] Oder in der Diplomarbeit von Florian Bittner, der eine erweiterbare Management-Plattform für Hostvirtualisierungslösungen entwickelt hat. [Bit09] Ebenfalls erwähnen sollte man das Open Virtualization Format der DTMF, welches ein herstellerunabhängiges Modell zur Beschreibung von VMs verwendet. [OVF10] All diesen Informationsmodellen ist gemeinsam, dass sie für das Management der gesamten Infrastruktur inklusive der Hosts auf denen die VMs laufen gedacht sind. Das in dieser Arbeit zu entwerfende Webportal soll auf einer höheren Abstraktionsebene laufen. Die Hosts spielen in diesem Zusammenhang keine Rolle. Sie werden durch das Virtualisierungssystem abstrahiert und sind weder für die Endnutzer noch für die Administratoren von Interesse. Die Virtualisierungslösung entscheidet selbstständig auf welchem Host eine VM erzeugt wird. Die zentrale Managementkomponente ist dafür zuständig, dass Befehle an eine konkrete VM an den richtigen Host weitergeleitet werden. Auch die managementrelevanten Bestandteile, zum Beispiel einer VM, werden in diesen Modellen sehr detailliert beschrieben. Eine solche Tiefe ist für diese Arbeit nicht erforderlich. Die Modellbestandteile, die sich auf die virtuellen Maschinen konzentrieren, können allerdings teilweise angepasst und übernommen werden.

Der große neue Aspekt bei dieser Arbeit ist, dass das Management der existierenden virtuellen Infrastruktur mit einer Antragsverwaltung verknüpft ist. Es gibt zum Teil sehr spezielle Anforderungen, zum Beispiel diejenige, dass Nutzer neue AGBs nur dann akzeptieren müssen, falls sie die jeweils aktuelle Version noch nicht akzeptiert haben. Selbst wenn also ein existierendes Modell zur Beschreibung der VMs verwendet wird, so muss dieses dennoch um Klassen, die diesen speziellen Beantragungsvorgang abbilden, erweitert werden.

Das entwickelte Informationsmodell (siehe Abbildung 5.5) baut auf acht Klassen auf. Da das System mehrere Nutzer unterstützen soll, wird eine Klasse User benötigt. Da die Nutzer AGBs angezeigt bekommen sollen, wird eine Klasse TAC (Terms and Conditions) benötigt. Die diversen Antragsarten, die die Nutzer einreichen können, werden durch die Klasse Request abgebildet. Die den Nutzern zur Verfügung gestellten VMs und Ressourcepools werden durch die Klassen VM und RP repräsentiert. Die Konfiguration einer VM oder eines Ressourcepools die ein Nutzer bei einem Antrag angibt oder über die eine VM verfügt, wird durch die Klasse Configuration modelliert. Die zu einer Konfiguration gehörenden Angaben bezüglich vorhandener Festplatten und Netzwerkadapter wird durch zwei weitere Klassen, HDD und NIC, abgebildet, weil deren Anzahl bei einer Konfiguration nicht beschränkt ist. Diese acht Klassen werden im Folgendem im Detail beschrieben.

5.3.1.1. User

Diese Klasse beschreibt einen Nutzer des Webportals, unabhängig davon zu welcher Benutzergruppe dieser gehört. Die Information zu welcher Gruppe ein Nutzer gehört wird übrigens nicht durch das Webportal gespeichert, sondern beim Login-Vorgang jedes mal von neuem über die AAI ermittelt. Damit das Portal an einen Benutzer Benachrichtigungen verschicken kann, enthält diese Klasse die für diesen Zweck vorgesehene Adresse. Zwar wäre es auch denkbar, dass man eine solche Adresse aus der Benutzerdatenbank abrufen, doch ist die Existenz einer solchen Information in dieser nicht sicher. Zudem kann es durchaus sein, dass ein Nutzer für die Benachrichtigungen des Webportals eine andere Adresse angeben möchte, als die allgemein gültige in der Benutzerdatenbank. Ein weiteres Attribut ist die ID der zuletzt akzeptierten AGB-Version. Diese Information wird benötigt, um beim Einreichen eines neuen Antrags festzustellen, ob dem Nutzer die aktuelle Version zur Akzeptanz angezeigt werden muss.

5. Konzept

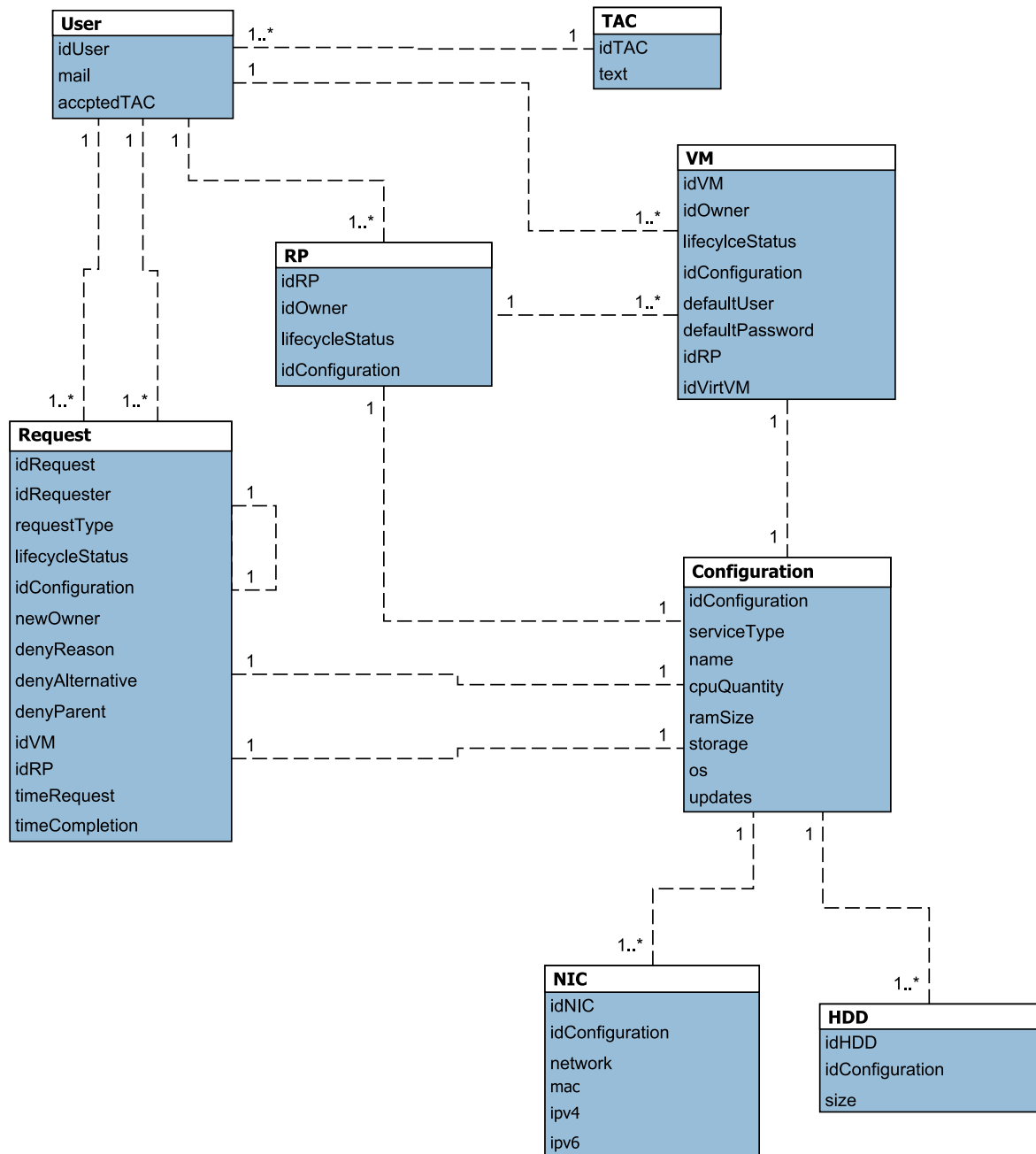


Abbildung 5.5.: Informationsmodell

5.3.1.2. TAC

Diese Klasse beschreibt eine Fassung der Allgemeinen Geschäftsbedingungen. Dies ist eine sehr einfach aufgebaute Klasse, welche lediglich aus der ID und dem Text der AGB-Fassung besteht.

5.3.1.3. Request

Diese Klasse beschreibt einen beliebigen Antrag und ist damit das zentrale Element der gesamten Auftragsverwaltung. Es gibt die folgenden vier Antragsarten, zwischen denen durch ein entsprechendes Attribut unterschieden wird:

create Dies ist ein Antrag auf Bereitstellung einer neuen VM oder eines neuen Ressourcepools.

modify Dies ist ein Antrag auf Modifikation einer VM oder eines Ressourcepools.

delete Dies ist ein Antrag auf Löschung einer VM oder eines Ressourcepools. Dies ist die einzige Antragsart bei welcher eine automatische Ausführung erfolgt und kein weiterer Eingriff durch einen Administrator erforderlich ist.

ownerChange Dies ist ein Antrag auf einen Besitzerwechsel. Normalerweise wird ein solcher Antrag durch den aktuellen Besitzer der VM eingereicht. In Ausnahmefällen kann es aber vorkommen, dass er dazu nicht mehr im Stande ist, zum Beispiel weil er unerwartet die Arbeitsstelle gewechselt oder erkrankt ist. Ist so einem Fall kann der Antrag auf einen Besitzerwechsel auch von einem Administrator eingereicht werden, damit die VM an einen anderen Mitarbeiter, der die Arbeit des ausgefallenen Kollegen fortsetzt, übergehen kann.

Von seiner Einreichung bis zur Erfüllung oder Ablehnung durchläuft ein Antrag mehrere Phasen, die durch den Lifecyclestatus beschrieben werden. Folgende Stati sind möglich:

pending Sobald der Antrag eingereicht wird, befindet er sich, mit Ausnahme eines Delete-Requests, in der Pending-Phase. Ein Create- und Modify-Request verbleibt in dieser Phase bis er von einem Administrator genehmigt oder abgelehnt wird. Ein Owner-Change-Request geht in die nächste Phase über, sobald der neue Besitzer die Übernahme bestätigt hat.

inprogress Bei einem Create- und Modify-Request beginnt diese Phase mit der Genehmigung durch den Administrator und endet mit der Bereitstellung der (modifizierten) VM beziehungsweise des (modifizierten) Ressourcepools. Da der Administrator vor der Bereitstellung noch manuelle Eingriffe vornehmen kann, kann diese Phase durchaus länger dauern. Finden solche Eingriffe nicht statt, ist die Dauer im Normalfall auf wenige Sekunden beschränkt. Bei einem OwnerChange-Request beginnt diese Phase mit der Bestätigung der Übertragung durch den neuen Besitzer. Einem Delete-Request wird dieser Status bereits bei der Einreichung zugewiesen.

fulfilled Wurde der Antrag erfüllt, wechselt der Antrag in den Fulfilled-Zustand und verbleibt in diesem.

denied Lehnt ein Administrator einen Antrag ab, geht dieser aus der Pending-Phase (bei OwnerChange-Requests aus der In-Progress-Phase) direkt in den Denied-Zustand über und verbleibt in diesem.

In Abbildung 5.6 werden diese Stati den verschiedenen Phasen des Dienstlebenszykluses zugeordnet.

Im Falle eines Create- oder Modify-Requests verweist die Klasse auf eine Configuration, welche die gewünschte Ausstattung der VM beziehungsweise des Ressourcenpools beschreibt. Diese Information wird nicht in der Request-Klasse selbst gespeichert, da die Konfigurationsinformationen auch mit der noch zu definierenden VM verknüpft werden sollen, was eine Auslagerung in eine separate Klasse erforderlich macht.

Alle Anträge, mit Ausnahme von abgelehnten Create-Anträgen, verfügen spätestens nach Abschluss über die ID der zugehörigen VM beziehungsweise des zugehörigen Ressourcenpools. Es kann sein, dass diese ID auf eine bereits gelöschte VM oder einen gelöschten Ressourcenpool verweist. Benötigt wird diese ID, um alle Anträge zusammenfassen zu können, die eine bestimmte VM oder einen bestimmten Ressourcenpool betreffen. Dies ist erforderlich um die Geschichte einer Ressource rekonstruieren zu können, was zum Beispiel für das Accounting erforderlich ist.

Ebenfalls für das Accounting von Interesse ist die Information zu welchem Zeitpunkt ein Antrag abgeschlossen wurde, denn dies ist der Zeitpunkt ab dem einem Nutzer eine Ressource zur Verfügung gestellt, diese modifiziert oder gelöscht wurde. Neben diesem Zeitstempel wird ein weiterer für den Zeitpunkt der Antragsanmeldung benötigt. Diese Information kann zum Beispiel für Administratoren hilfreich sein, damit diese bereits seit langem eingereichte Anträge bevorzugt behandeln können.

Wird ein Antrag abgelehnt, so erhält er als weiteres Attribut mindestens den vom Administrator angegebenen Ablehnungsgrund. Außerdem wird auf eine weitere Configuration verwiesen, wenn der Administrator einen Alternativvorschlag eingereicht hat. Da ein abgelehnter Create-Request keine VM- oder Ressourcenpool-ID erhalten kann, müssen die auf einem Alternativvorschlag aufbauenden Anträge einen Verweis auf den zuvor abgelehnten Antrag erhalten, um hier einen möglicherweise über mehrere Ablehnungen und darauf folgende Modifikationen erfolgenden Genehmigungsvorgang nachvollziehen zu können.

5.3.1.4. Configuration

Die Configuration beschreibt die Ausstattung einer VM oder eines Ressourcenpools. Eine Konfiguration kann nur mit einem Antrag verknüpft sein oder mit einem Antrag und einer dazugehörigen VM. Der erste Fall tritt auf, wenn zum Beispiel sich ein Antrag in der Pending-Phase befindet oder der Antrag abgelehnt wurde. Der zweite Fall tritt auf, wenn der Antrag genehmigt wurde und es sich um die aktuelle Konfiguration der VM beziehungsweise des Ressourcenpools handelt.

Die Attribute einer Configuration lassen sich direkt aus den Anforderungen ablesen, immerhin wurden die Parameter auf die ein Nutzer Einfluss nehmen soll während der Befragung konkret benannt. Dies wären der gewünschte Diensttyp (Attended Hosting, Unattended Hosting, Ressourcenpool), die Anzahl virtueller CPUs und die Größe des verfügbaren RAM. Hinzu kommt eine beliebige Anzahl an Festplatten mit einer bestimmten Größe und Netzwerkadapter, die an ein bestimmtes Netz angeschlossen werden sollen. Im Falle von Attended Hosting kommen als weitere Informationen der Name des Betriebssystems und eine Angabe, ob automatische Updates gewünscht sind, hinzu. Handelt es sich um ein Configuration

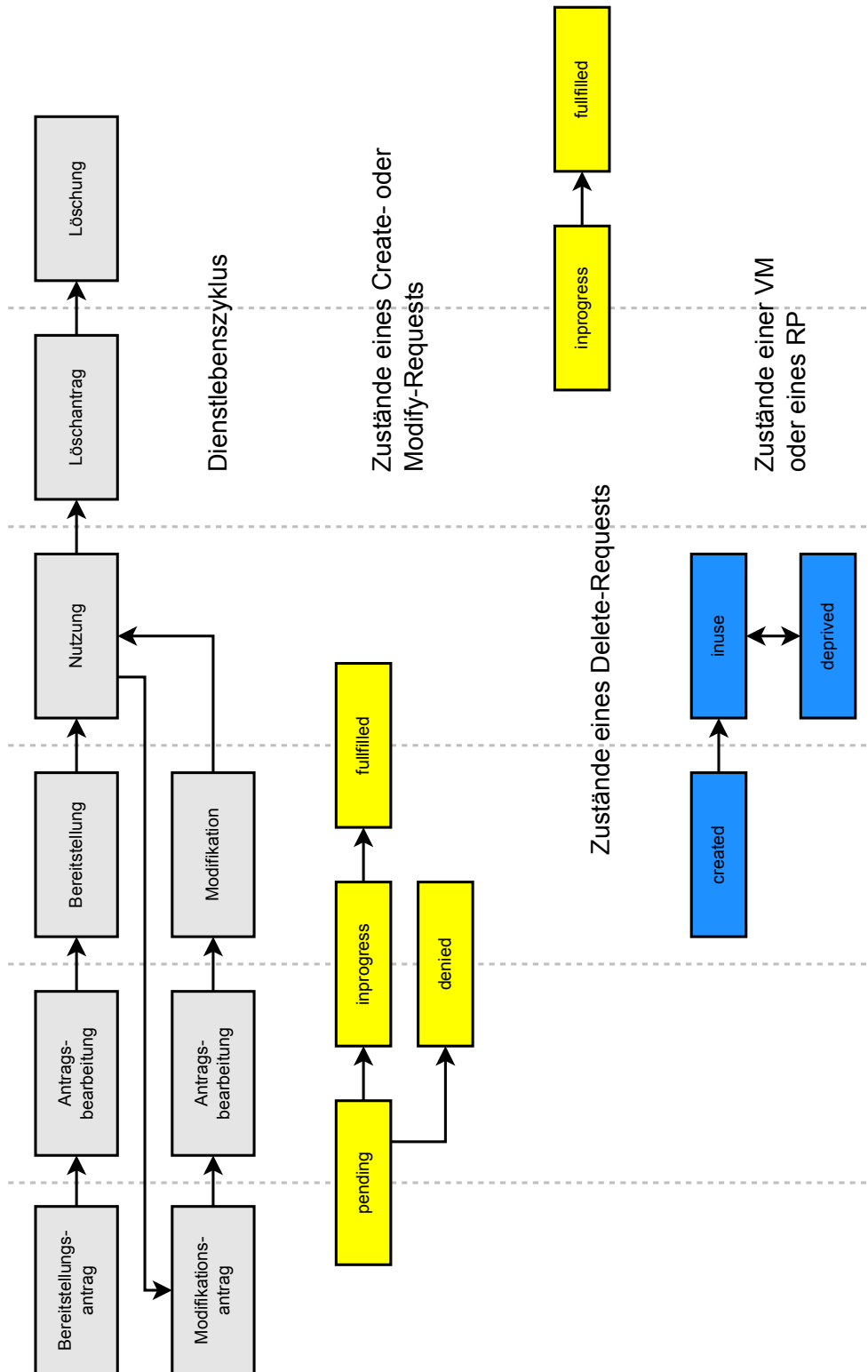


Abbildung 5.6.: Zustände von Anträgen, VMs und Resourcepools und ihre Zuordnung zu den Phasen des Dienstlebenszykluses

5. Konzept

für eine Resourcepool so wird die Größe des insgesamt zur Verfügung stehenden Speichers angegeben.

Die Parameter, die die Hardwareausstattung einer VM beschreiben, entsprechen der Arbeit von Gehring. In seiner Arbeit werden CPUs, Hauptspeicher, Festplatten und Netzwerkadap-ter allerdings als separate Klassen definiert. Da bei einer VM-Konfiguration im Sinne dieser Arbeit lediglich die Anzahl der CPUs und die Größe des Hauptspeichers von Interesse ist, sind keine separaten Klassen erforderlich, diese Angaben können über Attribute der Configu-ration abgebildet werden. Festplatten und NICs kann eine VM dagegen mehrere enthalten. Deshalb wird hier die Realisierung über separate Klassen beibehalten.

Aus Usability-Gründen wird zudem als weiteres Attribut der Configuration ein Name eingeführt. Über diesen soll der Nutzer seine VMs und Resourcepools besser unterscheiden können, als über irgendwelche internen in der Regel numerischen IDs.

5.3.1.5. HDD

Dieses Klasse beschreibt eine Festplatte. Es handelt sich dabei um eine Unterklasse der Configuration. Einer Configuration kann eine beliebige Anzahl an HDDs zugewiesen werden. Es handelt sich dabei um eine sehr einfache Klasse, deren einziges Attribut die Größe der Festplatte ist.

5.3.1.6. NIC

Dieses Klasse beschreibt einen Netzwerkadapter. Es handelt sich dabei um eine Unterklasse der Configuration. Einer Configuration kann eine beliebige Anzahl an NICs zugewiesen wer-den. Ein Netzwerkadapter kann an ein lokales oder ein globales Netz angeschlossen werden. Im ersten Fall ist die VM nur innerhalb der Institution beziehungsweise über VPN erreichbar, im zweiten Fall wird der VM eine global gültige IP-Adresse zugewiesen, so dass diese aus dem gesamten Internet erreicht werden kann. Diese Zuordnung wird über ein Attribut abgebildet.

Weitere Attribute eines NIC sind eine MAC-Adresse, eine IPv4-Adresse und eine IPv6-Adresse. Eine Rücksprache mit den Administratoren am LRZ und an der LMU hat ergeben, dass die Unterstützung von mehreren IP-Adressen pro Netzwerkadapter nicht notwendig ist. Die diversen Adressen werden erst zugewiesen, wenn auf Grundlage der zugehörigen Configuration eine VM erstellt wird. Wird die VM gelöscht oder wird dieser eine neue Confi-guration zugeordnet, so werden diese Attribute wieder gelöscht. Auf diese Weise werden die IP-Adressen wieder freigegeben und können einer neuen VM zugewiesen werden. Die Adress-Attribute sind also nur gesetzt, wenn die zum NIC gehörende Configuration mit einer VM verknüpft ist.

5.3.1.7. VM

Diese Klasse beschreibt vorhandene VMs. Betrachtet man die weiter oben erwähnten Mo-delle, wird man feststellen, dass sie alle eine solche Klasse besitzen, wenn auch diese an-ders genannt wird. Die Attribute beziehungsweise Unterklassen werden dabei zum Teil sehr ausführlich beschrieben. Viele dieser Attribute und Unterklassen sind für ein Management-Portal im Sinne dieser Arbeit aber nicht erforderlich. Als Beispiel sei der Zustand einer VM herausgegriffen. Das Management-Portal muss über diesen keine Kenntnis besitzen. Möchte der Nutzer den Zustand einer VM wissen, fragt das Portal diesen vom Virtualisierer ab

und zeigt ihn dem Nutzer an. Möchte der Nutzer eine VM starten, sendet das Portal einen entsprechenden Befehl ab, auch dann wenn die VM bereits läuft. Für eine sinnvolle Abarbeitung dieser Befehle ist die Virtualisierungslösung und in geringerem Umfang der Adapter verantwortlich. Die hier definierte VM-Klasse, die die existierenden VMs beschreibt, enthält deshalb keine Zustandsinformationen die sich vom Virtualisierer abrufen lassen. Ein anderes Beispiel wären Informationen über vorhandene Snapshots oder verfügbare mountbare ISO-Images. Auch diese Daten werden bei Bedarf vom Virtualisierer abgerufen und nicht redundant im Webportal vorgehalten. Auch Informationen zu gelöschten VMs werden nicht aufgehoben. Die Nutzungsdaten einer VM lassen sich allein an Hand der Requests und der mit diesen verknüpften Configurations rekonstruieren.

Genauso wie ein Request verfügt eine VM über einen Lifecyclestatus. Die Folgenden werden unterschieden:

created Während der Created-Phase ist die VM bereits erstellt, dem Nutzer wurden aber noch keine Zugriffsrechte gewährt. Dies kann zum Beispiel der Fall sein, wenn der Administrator noch manuelle Eingriffe vornimmt. Solange eine VM in der Created-Phase ist, kann der dazugehörige Request nicht in die Fullfilled-Phase wechseln.

inuse Sobald die VM dem Nutzer zur Verfügung gestellt wird, wechselt sie in die In-Use-Phase. Gleichzeitig wechselt der dazugehörige Request in die Fullfilled-Phase.

deprived Eine solche Phase lässt sich nicht aus den Anforderungen ableiten. Die Existenz einer solchen Phase erscheint allerdings sinnvoll, so dass sie hier definiert werden soll. Eine VM befindet sich in der Deprived-Phase, wenn dem Nutzer die Nutzungsrechte temporär entzogen wurden, nachdem er die VM bereits genutzt hat, die VM also in der In-Use-Phase war. Dies kann zum Beispiel sinnvoll sein, wenn der Verdacht aufkommt, dass die VM zu illegalen Zwecken missbraucht wird oder der Nutzer mit seinen Zahlungen im Rückstand ist. Eine VM kann beliebig oft zwischen dem In-Use- und Deprived-Status wechseln.

In Abbildung 5.6 werden diese Stati den verschiedenen Phasen des Dienstlebenszykluses zugeordnet.

Die Klasse verweist auf eine Configuration, dabei handelt es sich um die gleiche Configuration, die mit dem Create-Request beziehungsweise dem letzten erfüllten Modify-Request verknüpft ist.

Des weiteren werden in dieser Klasse ein Standard-Benutzername und ein Standard-Passwort für ein im Rahmen von Attended Hosting bereitgestelltes Betriebssystem gespeichert. Sollte der Nutzer diese vergessen haben, bevor er sie geändert hat, können diese Angaben so dem Nutzer erneut zugeschickt werden. Eine solche Speicherung der Zugangsdaten kann allerdings unter Umständen rechtlich problematisch sein und sollte gegebenenfalls von einem Juristen geprüft werden.

Zu den weiteren Attributen gehört die ID des Besitzers, die unter anderem dafür benötigt wird, um eine Liste aller VMs eines Benutzers zu erstellen und die virtualisiererinterne ID der VM, die sich von der portalinternen ID unterscheiden kann. Die virtualisiererinterne ID wird benötigt, um beim Versand von Kommandos an den Virtualisierer die zu steuernde VM zu identifizieren.

Gehört eine VM einem Ressourcepool an, wird auch auf eben jenen verwiesen. So lassen sich alle VMs die einem bestimmten Ressourcepool angehören ermitteln.

5.3.1.8. RP

Diese Klasse beschreibt einen Resourcepool. Dies ist wieder eine eher einfach aufgebaute Klasse. Da es sich beim Resourcepool genauso um eine beantragte Ressource wie eine VM handelt, müssen für diesen lediglich die relevanten Attribute übernommen werden. Dies wären der Verweis auf den Besitzer und auf eine Configuration und der Lifecyclestatus mit den selben Zuständen.

5.3.2. Portal-API(s)

Wie bereits bei der Auflistung der Schnittstellen erwähnt, muss das Portal Schnittstellen für diverse Funktionen zur Verfügung stellen. Der einfachste Fall ist, dass alle Bereiche abgedeckt werden, indem die betroffenen Komponenten direkten Zugriff auf das interne relationale Datenbanksystem des Portals bekommen. Das verwendete Protokoll wäre in diesem Fall SQL. Dank des sehr umfangreichen Rechtssystemes von RDBs, wie zum Beispiel MySQL, können die Zugriffsrechte der einzelnen Komponenten sehr detailliert gesteuert werden. Das Diagramm in Abbildung 5.4 zeigt die Realisierung der Portal-API auf diese Weise. Im Folgenden soll analysiert werden, in wie weit diese Lösung für die einzelnen Funktionsbereiche geeignet ist und ob es nicht besser wäre für einige dieser Funktionsbereiche dedizierte APIs zur Verfügung zu stellen.

Accounting Das Accounting benötigt Informationen darüber welcher Nutzer über welchen Zeitraum, welche VM beziehungsweise welchen Resourcepool mit welcher Konfiguration genutzt hat. Laut dem Informationsmodell werden die Requests mit den damit verbundenen Configurations dauerhaft gespeichert. Aus diesen Informationen kann man alle für das Accounting benötigten Nutzungsinformationen ableiten. Dies sei an einem Beispiel verdeutlicht: Um das Accounting für einen Nutzer durchzuführen, müssen zunächst alle Request gesucht werden, die von diesem Nutzer erstellt wurden. An Hand der VM-ID beziehungsweise Resourcepool-ID, die zu jedem nicht abgelehnten Request — und nur erfüllte Requests sind für das Accounting von Interesse — gespeichert wird, können alle Anträge, die eine bestimmte VM oder einen bestimmten Resourcepool betreffen, verknüpft werden. Aus dem Antragstyp und den abgespeicherten Zeitstempeln können die Zeitpunkte für die Bereitstellung, eventuell aufgetretene Modifikationen und unter Umständen bereits erfolgte Kündigungen abgeleitet und daraus die entsprechenden Nutzungszeiträume berechnet werden. Gesondert müssen noch VMs und Resourcepools betrachtet werden, die der Nutzer durch einen Besitzerwechsel erhalten oder abgegeben hat. Aber auch hierfür sind alle benötigten Informationen in der Datenbank vorhanden. An diesem Beispiel zeigt sich, dass das Accounting problemlos durchgeführt werden kann, in dem lediglich ein direkter Zugriff auf die Datenbank gewährt wird. Die Accounting-Komponente muss allerdings eine entsprechende Logik implementieren, um eine Auswertung der Tabellen durchzuführen. Es könnte also durchaus sinnvoll sein für das Accounting eine dedizierte Web-API bereitzustellen, über welche die Nutzungsdaten bereits in aggregierter, aufbereiteter Form zur Verfügung gestellt werden. Würde man eine solche API mit einem Benutzernamen und einem Zeitraum als Parameter abfragen, würde man eine Auflistung sämtlicher durch den Nutzer in diesem Zeitraum genutzter VMs und Resourcepools erhalten, wobei jeweils die diversen Konfigurationen und deren Geltungszeitraum aufgeführt wären.

CMDB Die CMDB benötigt in erster Linie Informationen über die gerade bestehenden VMs und Ressourcepools. Diese Informationen liegen in den entsprechenden Tabellen bereit. Da es sich bei der CMDB oftmals ebenfalls um eine relationale Datenbank handelt, stellt eine Synchronisation der Daten über SQL kein Problem dar. Eine dedizierte API ist in diesem Fall nicht erforderlich.

IP-Konfiguration Die Komponente, welche die IP-Konfiguration der VMs vornimmt, also der DHCP-Server, benötigt Informationen darüber, welche IP-Adressen welcher MAC-Adresse zugeordnet werden müssen. Hierfür bieten sich mehrere Lösungsvarianten an: Die erste ist, dass das Portal bei einer Änderung der IP-Konfiguration eine neue Konfigurationsdatei für den DHCP-Server erstellt, diese ihm zur Verfügung stellt und ein Signal an ihn schickt, damit dieser die Konfigurationsdatei neu einliest. Bei dieser Variante muss der DHCP-Server entsprechende Schnittstellen zur Verfügung stellen, das Portal muss dabei über die erforderlichen Rechte für den Zugriff auf den DHCP-Server verfügen. Die zweite Variante ist eine abgeschwächte Form der ersten. Das Portal erstellt auch in diesem Fall Konfigurationsdateien für den DHCP-Server, dieser erkennt Änderungen an diesen aber automatisch oder liest die Konfiguration in regelmäßigen Abständen neu ein. Die dritte Variante, die als die eleganteste erscheint, und die auch in den bisherigen Grafiken dargestellt wurde ist, dass der DHCP-Server direkten Zugriff auf die Datenbank erhält und aus dieser immer die aktuelle IP-Konfiguration ausliest. Dieser Fall wird durch die multifunktionale Portal-API abgedeckt. Dem DHCP-Server genügt in diesem Fall nur ein lesender Zugriff auf die relevanten Spalten der Tabelle IPConfig. Eine dedizierte API ist hier nicht sinnvoll.

ITSM Die Integration der ITSM-Software kann unterschiedlich stark erfolgen und in Abhängigkeit davon ist die multifunktionale Portal-API vollkommen ausreichend oder mangelhaft. Soll zum Beispiel die ITSM-Software nur in der Lage sein als alternative Schnittstelle zur Einreichung von Anträgen zu dienen, so reicht die SQL-basierte multifunktionale API aus, es muss lediglich ein entsprechender Eintrag in der Datenbank erfolgen. Dabei ist die ITSM-Software selbst für die Einhaltung der Grenzwerte und eine Plausibilitätskontrolle verantwortlich. Die Datenbank kann nur den passenden Datentyp für die unterschiedlichen Parameter sicherstellen. Eine Überschreitung der Grenzwerte oder unvollständige Anträge stellen aber kein größeres Problem dar, da sie sowieso von Administratoren geprüft werden. Sollen über die ITSM-Software zum Beispiel auch Anträge bearbeitet werden, wird es komplizierter. Eine Möglichkeit dies zu realisieren wäre, die Datenbank in regelmäßigen Abständen auf Veränderungen zu untersuchen und beim Auftreten von solchen entsprechende Aktionen einzuleiten. Wird zum Beispiel der Wechsel des Lifecyclestatus einer VM von “created” auf “inuse” erkannt, so werden dem Nutzer Zugriffsrechte auf seine VM eingeräumt. Eine solche Prüfung der Datenbank auf Veränderungen ist aber fehleranfällig und nicht sehr elegant. Soll also die ITSM-Software Aktionen des Portals anstoßen, so sollte eine entsprechende dedizierte API für diesen Funktionsbereich entwickelt werden. Soll die ITSM-Software als vollwertige Alternative zur Weboberfläche dienen, so muss die API sämtliche Funktionen abdecken, die in dieser verfügbar sind. Benötigt werden also Methoden für die Antragsanmeldung, -abfrage und -bearbeitung, Methoden für die Abfrage von VM-Informationen und die Ausführung von Steuerkommandos und für die Ressourcepools Methoden zur Erstellung, Modifikation und Löschung von VMs. Hinzu kommen Me-

5. Konzept

thoden zur Verwaltung des Nutzerkontos, um etwa die akzeptierte AGB-Fassung zu aktualisieren oder die Benachrichtigungsadresse zu ändern.

Zusammenfassend kann man eine SQL-basierte multifunktionale Portal-API wie folgt bewerten: Die Grundbedürfnisse werden damit zufriedenstellend abgedeckt. Für das Accounting kann eine separate API entwickelt werden, wenn man die Komplexität des Accounting-Systems möglichst gering halten will. Sollen über die ITSM-Software nicht nur Anträge eingereicht oder Informationen abgerufen werden, soll die ITSM-Software also auch Aktionen des Webportals anstoßen, so sollte eine dedizierte API hierfür entwickelt werden. Die Methoden die eine solche API zur Verfügung stellen muss entsprechen im Wesentlichen den Funktionen die über die Oberfläche des Webportals verfügbar sind.

5.4. Der Adapter

Der Adapter ist als eine sehr einfache Komponente entworfen, welche primär lediglich die per Adapter-API empfangenen Befehle in ein vom verwendeten Virtualisierer verstandenes Format übersetzt. Der Adapter ist zustandslos, d.h. der Adapter loggt sich auch für die Dauer einer Sitzung eines Benutzers nicht dauerhaft beim Virtualisierer ein, sondern loggt sich nach der Ausführung von jedem Befehl wieder aus. Auf diese Weise können auch problemlos mehrere Adapter zur Lastverteilung oder aus Gründen der Ausfallsicherheit eingesetzt werden, und im Round-Robin-Verfahren vom Portal angesprochen werden. Die Einfachheit des Adapters soll es ermöglichen schnell und unkompliziert neue Versionen zu entwickeln und so problemlos neue Virtualisierer anzubinden oder auf Veränderungen in der API des Virtualisierers zu reagieren. Die Logik, zum Beispiel die einzelnen Phasen der Bereitstellung einer VM, finden sich im Portal. Das Portal muss also nacheinander mehrere Befehle an den Adapter schicken, um die Bereitstellung einer VM abzuschließen. Dies beginnt mit der Erstellung der VM und endet mit der Zuteilung der Zugriffsrechte. Sollte die Verbindung zwischen Adapter und Virtualisierer unterbrochen werden oder sollte die Ausführung eines Befehls aus irgend einem anderen Grund scheitern, so unternimmt der Adapter selbstständig keine erneuten Versuche zu einem späteren Zeitpunkt. Wird ein solches Verhalten gewünscht, so muss dieses auf Seiten des Portals implementiert werden. Der Adapter führt niemals selbstständig Operationen aus.

Die Befehle, die ein Adapter über seine API erhält, sollten sich bei den meisten Virtualisierern direkt in einen passenden Befehl für diesen übersetzen lassen. Kann der Adapter ein bestimmtes Kommando nicht abbilden, weil dieses vom Virtualisierer nicht unterstützt wird oder weil man auf eine Implementierung dieser Funktion im Adapter verzichtet hat, muss dieser eine entsprechende Fehlermeldung zurückliefern. In speziellen Fällen besteht auch die Möglichkeit über den Adapter Datenstrukturen zu speichern, wenn diese sich nicht auf die Virtualisierungslösung abbilden lassen. Angenommen der Virtualisierer kann keine Beschreibung für Snapshots speichern. In diesem Fall kann der Adapter die vom Portal übermittelte Beschreibung verwerfen, versuchen andere Datenstrukturen des Virtualisierers für diesen Zweck zu verwenden, was in der Regel nicht empfehlenswert ist, oder diese Information in einer eigenen Datenstruktur außerhalb des Virtualisierers ablegen. Bei einem Abruf der Snapshots müsste der Adapter im letzteren Fall wieder die Informationen vom Virtualisierer mit seinen eigenen Informationen verknüpfen und an das Portal senden. Solche Maßnahmen sollten allerdings selten notwendig sein, da der vom Adapter abgebildete

Funktionsumfang auf einer recht weit verbreiteten Basis aufbaut und von der Mehrheit der Virtualisierungslösungen unterstützt wird.

5.5. Adapter-API

Im Folgenden wird die API definiert, über welche das Portal den Adapter anspricht. Es handelt sich dabei um eine REST-API. Zunächst findet sich eine Beschreibung der Rückgabewerte und Fehlertypen, es folgt in Tabelle 5.2 eine Übersicht über sämtliche URIs mit einer kurzen Beschreibung. Anschließend werden die einzelnen Requests mit den benötigten Parametern und einer beispielhaften Rückgabe ausführlich beschrieben. URI-Bestandteile die mit einem Doppelpunkt beginnen entsprechen Parametern, die als Bestandteil der URI anzugeben sind. Dabei handelt es sich um die IDs der gewünschten Ressourcen, die angesprochen werden sollen, also zum Beispiel um eine VM oder einen Resourcepool. Die Rückgabe erfolgt in der Regel im XML-Format. Das Root-Element ist immer `vmmgmt`, welches als Attribut die Versionsnummer der verwendeten Adapter-API trägt, um bei zukünftigen Versionen des Portals oder Adapters Versionskonflikte erkennen zu können. In dieser Arbeit ist die Version immer die 1.0. Einzige Ausnahme bildet der Aufruf der VM-Konsole, bei welchem die Rückgabe ein HTML-Fragment ist. Mehr Informationen hierzu findet man bei der Beschreibung des entsprechenden Requests. Zeitangaben innerhalb der XML-Daten haben im XML Schema `DateTime` Datentyp zu erfolgen, also die Form `YYYY-MM-DDThh:mm:ss` aufzuweisen.

Um eine Authentifizierung des Adapters mit passenden Rechten am Virtualisierer zu ermöglichen, muss das Portal die dazu benötigten Daten bei jedem Aufruf der API mitsenden. Dies erfolgt mittels Basic Authentication über HTTP wie in RFC 2617 [FHBH⁺99] definiert. Das Passwort wird dabei lediglich base64-codiert übertragen, da aber die Kommunikation zwischen Portal und Adapter (wie auch zwischen Adapter und Virtualisierer) verschlüsselt zu erfolgen hat, besteht keine Gefahr, dass das Passwort mitgehört werden könnte. Eine Ausnahme hiervon kann aus Performance-Gründen erfolgen, wenn Portal und Adapter auf der gleichen Maschine laufen.

5.5.1. Rückgabewerte

Im Erfolgsfall antwortet der Adapter auf die meisten GET- und PUT-Requests mit dem HTTP-Statuscode 200 `OK` und übermittelt im Body die gewünschten Informationen beziehungsweise die neue Konfiguration in XML-Form. Nach dem Anlegen einer neuen Ressource mittels eines POST-Requests wird mit 201 `Created` geantwortet. In diesem Fall enthält der Body zur Bestätigung die Parameter mit denen die neue Ressource beantragt wurde, sowie die ID, über welche die Ressource zukünftig angesprochen werden kann. Neben der ID im Rahmen der XML-Daten erfolgt auch eine Angabe der URI für die neue Ressource mittels Location-Header. Wird eine Ressource mittels eines HTTP-DELETE-Requests gelöscht, erfolgt eine Antwort mit dem Code 204 `No Content`, der Body ist in diesem Fall leer.

Tritt ein Fehler bei der Abarbeitung des Requests auf, wird ein Statuscode aus der 400er- oder 500er-Familie geschickt. Sämtliche Fehler werden in Tabelle 5.1 aufgeführt.

5. Konzept

Soweit möglich wird die Fehlermeldung ebenfalls im XML-Format ausgegeben, wie im folgenden Beispiel zu sehen:

```
<vmmgmt version=" 1.0">
  <error type="MissingRequiredQueryParameter">
    Snapshot-ID required.
  </error>
</vmmgmt>
```

HTTP-Statuscode	Fehler	Beschreibung
400	MissingRequiredQueryParameter	Ein benötigter Parameter wurde nicht angegeben.
400	InvalidQueryParameterValue	Ein Parameter weist einen ungültigen Wert auf.
400	InvalidUri	Die angegebene URI verweist auf keine Ressource.
400	InvalidHttpVerb	Ein nicht unterstütztes HTTP-Verb wurde verwendet.
403	AuthenticationFailed	Die Authentifizierung ist fehlgeschlagen.
403	InsufficientAccountPermissions	Unzureichende Berechtigung für die gewünschte Aktion.
404	ResourceNotFound	Die angegebene Ressource, z.B. eine VM mit der angegebenen ID, existiert nicht.
500	InternalServerError	Interner Fehler. Tritt auch dann auf, wenn es zu einem Problem bei der Kommunikation zwischen Adapter und Virtualisierer kommt. Eine unter Umständen vorhandene Fehlermeldung des Virtualisierers wird in die Antwort an den Client eingefügt.
501	NotImplemented	Eine gewünschte Funktionalität ist nicht vorhanden, da diese entweder vom Adapter nicht implementiert wurde oder vom Virtualisierer nicht unterstützt wird.

Tabelle 5.1.: Fehlertypen der Adapter-API

5.5.2. Requests

GET login

Prüfung des Logins und Abfrage der Rolle, dient der Anmeldung am Portal ohne weitere Interaktion mit dem Virtualisierer. Entsprechend der Aufteilung der Funktionen auf die einzelnen Komponenten, wäre dies eigentlich eine Aufgabe der AAI. Da allerdings viele Virtualisierer eine integrierte Benutzerverwaltung enthalten, kann die Autorisierung über die API des Virtualisierers erfolgen und folglich kann es auch eine entsprechende Methode in der Adapter-API geben. Werden Login-Parameter bei diesem Request akzeptiert, so sollten sie auch bei allen weiteren Requests akzeptiert werden.

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <login id="user" group="enduser">
</vmmgmt>
```

POST vm

Erstellen einer neuen VM

Parameter:

cpuquantity: Anzahl der CPUs

ramsize: Größe des RAM in MB

hdd: Größen der Festplatten in MB, bei mehreren Festplatten werden die einzelnen Werte durch Kommata getrennt

nic: Netze der Netzwerkadapter, mögliche Werte sind `global` und `local`, mehrere Werte werden durch Kommata getrennt

os: Bezeichnung des Betriebssystems für Attended Hosting (optional)

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <vm id="32552_testvm">
    <hardware>
      <cpuquantity>2</cpuquantity>
      <ramsize>256</ramsize>
      <hdd>10000</hdd>
      <hdd>20000</hdd>
      <nic>00:11:22:33:44:55</nic>
    </hardware>
    <status>created</status>
  </vm>
</vmmgmt>
```

PUT vm/:id/hardware

Ändern der Hardwareausstattung einer VM

Parameter:

id: ID der zu modifizierenden VM

Request	Beschreibung
GET login	Prüfung des Logins und Abfrage der Rolle
POST vms PUT vm/:id/hardware PUT vm/:id/privileges PUT vm/:id/status PUT vm/:id/tools GET vm/:id/hardware GET vm/:id/privileges GET vm/:id/status GET vm/:id/stats GET vm/:id/console GET vm/:id/iso GET vm/:id/snapshot DELETE vm/:id POST vm/:id/iso PUT vm/:id/iso/:isoid DELETE vm/:id/iso/:isoid POST vm/:id/snapshot PUT vm/:id/snapshot/:snapid DELETE vm/:id/snapshots/:snapid	Erstellen einer neuen VM Ändern der Ausstattung einer VM Ändern der Zugriffsrechte auf eine VM Ändern des Betriebszustands einer VM Virtualisiererspezifische Tools auf der VM installieren Abfrage der Ausstattung einer VM Abfrage der Zugriffsrechte auf eine VM Abfrage des Betriebszustands einer VM Abfrage der Statistiken einer VM Starten der Konsole einer VM Abfrage verfügbarer mountbarer ISO-Images Abfrage verfügbarer Snapshots einer VM Löschen einer VM Upload eines neuen ISO-Images ISO-Image mounten oder unmounten Löschen eines ISO-Images Anlegen eines Snapshots Laden eines Snapshots Löschen eines Snapshots
POST rp PUT rp/:id/hardware PUT rp/:id/privileges PUT rp/:id/vm/:vmid GET rp/:id/usage GET rp/:id/stats GET rp/:id/vm DELETE rp/:id	Anlegen eines neuen Ressourcepools Ändern der Ausstattung eines Ressourcepools Ändern der Zugriffsrechte auf einen Ressourcepools Hinzufügen/Entfernen einer VM zu einem Ressourcepool Abfrage der Auslastung eines Ressourcepools Abfrage der Statistiken eines Ressourcepools Abfrage sämtlicher zu einem Ressourcepool gehörender VMs Löschen eines Ressourcepools

Tabelle 5.2.: Zusammenfassung der Adapter-API

cpuquantity: Anzahl der CPUs (optional)
 ramsize: Größe des RAM in MB (optional)
 hdd: Größen der Festplatten in MB (optional), bei mehreren Festplatten werden die einzelnen Werte durch Kommata getrennt; soll eine Platte gelöscht werden, ist als Größe "0" anzugeben
 nic: Netze der Netzwerkadapter (optional), mögliche Werte sind `global` und `local`, mehrere Werte werden durch Kommata getrennt; soll ein Adapter gelöscht werden, ist an seiner Stelle eine "0" anzugeben

Beispielhafte Rückgabe:

```

<vmmgmt version="1.0">
  <vm id="32552_testvm">
    <hardware>
      <cpuquantity>2</cpuquantity>
      <ramsize>256</ramsize>
      <hdd>15000</hdd>
      <nic>00:11:22:33:44:55</nic>
      <nic>00:23:32:42:42:62</nic>
    </hardware>
  </vm>
</vmmgmt>
  
```

PUT vm/:id/privileges

Ändern der Zugriffsrechte auf eine VM

Parameter:

id: ID der zu verändernden VM
 user: ID des Benutzers für den die Zugriffsrechte verändert werden sollen
 privilege: Neue Zugriffsrechte für den Benutzer. Zur Auswahl stehen die Werte `enduser`, `rpuser` und `none`

Die Rückgabe enthält eine Auflistung aller Benutzerrechte inklusive der mit dem Request modifizierten.

Beispielhafte Rückgabe:

```

<vmmgmt version="1.0">
  <vm id="32552_testvm">
    <permission id="eu2" privilege="enduser" />
    <permission id="rpu4" privilege="rpuser" />
  </vm>
</vmmgmt>
  
```

PUT vm/:id/status

Ändern des Status einer VM, also das Starten und Stoppen

Parameter:

id: ID der zu steuernden VM
 status: die gewünschte Aktion, zur Auswahl stehen: `on`, `off` und `pause`

5. Konzept

Beispielhafte Rückgabe:

```
<vmmgmt version=" 1.0">  
  <vm id=" testvm">  
    <status>off</status>  
  </vm>  
</vmmgmt>
```

PUT vm/:id/tools

Installation virtualisiererspezifische Tools in der VM (z.B. VMware Tools)

Parameter:

id: ID der zu verändernden VM

Beispielhafte Rückgabe:

```
<vmmgmt version=" 1.0">  
  <vm id=" testvm">  
    <tools>installed</tools>  
  </vm>  
</vmmgmt>
```

GET vm/:id/hardware

Abfrage der Ausstattung einer VM. Diese Informationen liegen auch in der Datenbank des Portals vor und werden normalerweise dort abgerufen. Dennoch kann es in bestimmten Fällen, zum Beispiel zu Kontrollzwecken, sinnvoll sein diese Informationen direkt vom Virtualisierer abzurufen.

Parameter:

id: ID der abzufragenden VM

Beispielhafte Rückgabe:

```
<vmmgmt version=" 1.0">  
  <vm id=" 32552_testvm">  
    <hardware>  
      <cpuquantity>2</cpuquantity>  
      <ramsize>256</ramsize>  
      <hdd>25000</hdd>  
      <nic>00:23:32:42:42:62</nic>  
    </hardware>  
  </vm>  
</vmmgmt>
```

GET vm/:id/privileges

Abfrage der Zugriffsrechte auf eine VM

Parameter:

id: ID der abzufragenden VM

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <vm id="32552_testvm">
    <permission id="eu2" privilege="enduser" />
  </vm>
</vmmgmt>
```

GET vm/:id/status

Abfrage des Betriebszustands einer VM. Aus Performancegründen kann mit dieser Methode auch der Betriebszustand von mehreren VMs gleichzeitig abgefragt werden, da eine solche Abfrage sehr oft auftritt und dies zudem meistens, zum Beispiel im Rahmen von Übersichten, für mehrere VMs gleichzeitig erfolgt.

Parameter:

id: ID der VM beziehungsweise kommasetrennte Liste von IDs

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <vm id="32552_testvm">
    <status>on</status>
  </vm>
  <vm id="32662_testvm2">
    <status>off</status>
  </vm>
</vmmgmt>
```

GET vm/:id/stats

Abfrage der Statistiken einer VM. Dies ist ein äußerst schwieriger Punkt, da die Statistiken, die über eine VM abgerufen werden können, sich von Virtualisierer zu Virtualisierer stark unterscheiden. Die hier ausgewählten Parameter sind sehr grundlegende Werte die von vielen Virtualisierungslösungen unterstützt werden. Dennoch kann es Virtualisierer geben, die diese nicht zur Verfügung stellen können. Wenn sehr ausführliche Statistiken gewünscht werden, so kommt man um eine virtualisierspezifische Modifikation des Portals nicht herum. Als Alternative wäre zudem noch denkbar die Generierung der Statistiken in den Adapter zu verlagern und von diesem an das Portal ein fertiges HTML-Fragment senden zu lassen.

Parameter:

id: ID der abzufragenden VM

Bei den Rückgabewerten handelt es sich um die CPU-Anforderung und CPU-Nutzung der VM in MHz, um den RAM-Verbrauch der VM und des Hosts, um Private-Memory-Anteil, Shared-Memory-Anteil und Swapped-Memory-Anteil der VM, alle jeweils in MB, und um die Uptime in Sekunden.

5. Konzept

Beispielhafte Rückgabe:

```
<vmmgmt version=" 1.0">
  <vm id=" 32552_testvm">
    <stats>
      <cpuUsage>100</cpuUsage>
      <cpuDemand>100</cpuDemand>
      <ramVmUsage>32</ramVmUsage>
      <ramHostUsage>12</ramHostUsage>
      <ramPrivate>10</ramPrivate>
      <ramShared>1</ramShared>
      <ramSwapped>1</ramSwapped>
      <uptime>12842</uptime>
    </stats>
  </vm>
</vmmgmt>
```

GET vm/:id/console

Aufruf der Konsole einer VM. In diesem Fall erfolgt keine XML-Rückgabe, sondern die Angabe eines HTML-Fragments, welches das Portal verwenden muss, um das Konsolen-Plugin anzuzeigen. Im Falle von vSphere ist für den Konsolenaufruf keine Kommunikation über die vSphere API erforderlich, das Konsolen-Plugin baut die Verbindung selbstständig auf. Die Konsole ist allerdings immer eine virtualisiererspezifische Lösung und wäre diese fest auf Seite des Portals implementiert, würde ein Wechsel des Virtualisierers auch eine Veränderung am Portal erfordern. Der Aufruf der Konsole über den Adapter hat den Zweck eine strikte Trennung von der generischen Portalfunktionalität und virtualisiererspezifischen Code zu erreichen. Des weiteren kann der Adapter, sollte es eine Virtualisierungslösung erfordern, so die Konsole über die Virtualisierer-API initialisieren, bevor der Code für das Plugin an das Portal gesendet wird.

Parameter:

id: ID der VM

Beispielhafte Rückgabe:

```
<object width=" 800" height=" 600">
  [...]
</object>
```

GET vm/:id/iso

Abfrage der zur Verfügung stehenden mountbaren beziehungsweise gemounteten ISO-Images

Parameter:

id: ID der abzufragenden VM

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <vm id="testvm">
    <isos>
      <iso id="img001"
        description="Beispielimage" state="mounted" />
      <iso id="img002"
        description="Testimage" state="unmounted" />
    </isos>
  </vm>
</vmmgmt>
```

GET vm/:id/snapshot

Abfrage verfügbarer Snapshots einer VM

Parameter:

id: ID der abzufragenden VM

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <vm id="testvm">
    <snapshots>
      <snapshot id="snap001"
        description="Snapshot vor Installation"
        time="2011-06-20T12:11:10" />
      <snapshot id="snap002"
        description="Snapshot nach Installation"
        time="2011-06-20T16:22:23" />
    </snapshots>
  </vm>
</vmmgmt>
```

DELETE vm/:id

Löschen einer VM

Parameter:

id: ID der zu löschenden VM

POST vm/:id/iso

Upload eines neuen ISO-Images. Der Upload erfolgt als normaler HTML-Upload, also gemäß RFC 1867 [NM95].

Parameter:

id: ID der VM

description: Beschreibung des Images

5. Konzept

Die Rückgabe enthält eine Auflistung aller verfügbaren Images der VM, inklusive des neu hochgeladenen.

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <vm id="testvm">
    <isos>
      <iso id="img001"
        description="Beispielimage" state="mounted" />
      <iso id="img002"
        description="Testimage" state="unmounted" />
    </isos>
  </vm>
</vmmgmt>
```

PUT vm/:id/iso/:isoid

ISO-Image einhängen oder aushängen

Parameter:

id: ID der VM

isoid: ID des einzuhängenden oder auszuhängenden Images

action: Gewünschte Aktion, zur Verfügung stehen mount und unmount

Die Rückgabe enthält eine Auflistung aller zur Verfügung stehenden und gemounteten Images, inklusive der Änderung durch den Request.

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <vm id="testvm">
    <images>
      <image id="img001"
        description="Beispielimage"
        state="mounted" />
      <image id="img002"
        description="Testimage"
        state="mounted" />
    </images>
  </vm>
</vmmgmt>
```

DELETE vm/:id/iso/:isoid

Löschen eines ISO-Images

Parameter:

id: ID der VM

isoid: ID des zu löschenden Images

POST vm/:id/snapshot

Anlegen eines neuen Snapshots

Parameter:

id: ID der VM

description: Name beziehungsweise Beschreibung des Snapshots

Die Rückgabe enthält eine Auflistung aller angelegten Snapshots der VM, inklusive des neu angelegten.

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <vm id="testvm">
    <snapshots>
      <snapshot id="snap001"
        description="Snapshot vor Installation"
        time="2011-06-20T12:11:10" />
      <snapshot id="snap002"
        description="Snapshot nach Installation"
        time="2011-06-20T16:22:23" />
    </snapshots>
  </vm>
</vmmgmt>
```

PUT vm/:id/snapshot/:snapid

Laden eines Snapshots

Parameter:

id: ID der VM

snapid: ID des zu ladenden Snapshots

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <vm id="testvm">
    <snapshot id="snap001"
      description="Snapshot vor Installation" />
  </vm>
</vmmgmt>
```

DELETE vm/:id/snapshot/:snapid

Löschen eines Snapshots

Parameter:

id: ID der VM

snapid: ID des zu löschenden Snapshots

5. Konzept

POST rp

Anlegen eines neuen Ressourcepools

Parameter:

`cpuquantity`: Anzahl der CPUs

`ramsize`: Größe des RAM in MB

`storage`: Größen der Speichers in MB

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <rp id="12515_testrp">
    <hardware>
      <cpuquantity>50</cpuquantity>
      <ramsize>102400</ramsize>
      <storage>20480000</storage>
    </hardware>
    <status>created</status>
  </rp>
</vmmgmt>
```

PUT rp/:id/hardware

Ändern der Hardwareausstattung eines Ressourcepools

Parameter:

`id`: ID des zu modifizierenden Ressourcepools

`cpuquantity`: Anzahl der CPUs (optional)

`ramsize`: Größe des RAM in MB (optional)

`storage`: Größe des Speichers in MB (optional)

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <rp id="12515_testrp">
    <hardware>
      <cpuquantity>100</cpuquantity>
      <ramsize>102400</ramsize>
      <storage>20480000</storage>
    </hardware>
  </rp>
</vmmgmt>
```

PUT rp/:id/privileges

Ändern der Zugriffsrechte auf einen Ressourcepool

Parameter:

`id`: ID des zu verändernden Ressourcepools

`user`: ID des Benutzers für den die Zugriffsrechte verändert werden sollen

`privilege`: Neue Zugriffsrechte für den Benutzer. Zur Auswahl stehen die Werte `rpuser` und `none`.

Die Rückgabe enthält eine Auflistung aller Benutzerrechte inklusive der mit dem Request modifizierten.

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <rp id="12515_testrp">
    <permission id="rpu4" privilege="rpuser" />
  </rp>
</vmmgmt>
```

PUT rp/:id/vm/:vmid

Hinzufügen einer VM zu einem Ressourcepool oder Entfernen einer VM aus einem Ressourcepool

Parameter:

`id`: ID des Ressourcepools

`vmid`: ID der VM

`action`: Gewünschte Aktion, zur Auswahl stehen `add` und `remove`.

Die Rückgabe enthält eine Auflistung aller VMs innerhalb des Ressourcepools.

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <rp id="12515_testrp">
    <vm id="23525_rpvml" />
    <vm id="23544_rpvml2" />
  </rp>
</vmmgmt>
```

GET rp/:id/usage

Abfrage der Auslastung eines Ressourcepools

Parameter:

`id`: ID des abzufragenden Ressourcepools

Bei den Rückgabewerten handelt es sich um Prozentangaben.

Beispielhafte Rückgabe:

```
<vmmgmt version="1.0">
  <rp id="12515_testrp">
    <usage>
      <cpuUsage>12</cpuUsage>
      <ramUsage>33</ramUsage>
      <storageUsage>80</storageUsage>
    </usage>
  </rp>
</vmmgmt>
```

5. Konzept

```
| </rp>  
| </vmmgmt>
```

GET rp/:id/stats

Abfrage der Statistiken eines Ressourcenpools

Parameter:

id: ID des abzufragenden Ressourcenpools

Die Rückgabewerte werden über Addition der Werte der einzelnen VMs ermittelt.

Beispielhafte Rückgabe:

```
| <vmmgmt version="1.0">  
|   <rp id="12515_testrp">  
|     <stats>  
|       <cpuUsage>on</cpuUsage>  
|       <cpuDemand>on</cpuDemand>  
|       <ramVmUsage>on</ramVmUsage>  
|       <ramHostUsage>on</ramHostUsage>  
|       <ramPrivate>on</ramPrivate>  
|       <ramShared>on</ramShared>  
|       <ramSwapped>on</ramSwapped>  
|     </stats>  
|   </rp>  
| </vmmgmt>
```

GET rp/:id/vm

Abfrage sämtlicher zu einem Ressourcenpool gehörender VMs

Parameter:

id: ID des abzufragenden Ressourcenpools

Die Rückgabe enthält eine Auflistung aller VMs innerhalb des Ressourcenpools.

Beispielhafte Rückgabe:

```
| <vmmgmt version="1.0">  
|   <rp id="12515_testrp">  
|     <vm id="23525_rpv1" />  
|     <vm id="23243_rpv3" />  
|   </rp>  
| </vmmgmt>
```

DELETE rp/:id

Löschen eines Ressourcenpools

Parameter:

id: ID des zu löschenden Ressourcenpools

5.6. Prüfung der Anforderungen

Im Folgenden soll dieses Konzept dem Anforderungskatalog gegenübergestellt werden.

Die Erfüllung vieler funktionaler Anforderungen kann man aus den vorangegangenen Unterkapiteln ableiten. Das Management baut auf den passenden Benutzergruppen, also Administratoren, Endnutzern und, da Resourcepools unterstützt werden, Resourcepool-Nutzern auf. Der Bestell- und Bereitstellungsvorgang ist elementarer Bestandteil des Webportals. Dass die verschiedenen Steuerfunktionen berücksichtigt wurden, kann man am besten an Hand der Spezifikation der Adapter-API erkennen. Einige der Anforderungen sind primär eher von der konkreten Implementierung als vom Konzept abhängig. So ist zum Beispiel die Anforderung, dass beim Bestellvorgang Vorlagen verfügbar sein sollen, von einer entsprechenden Implementierung hiervon in der View abhängig. Steht das Konzept einer Implementierung einer bestimmten Funktion nicht im Wege, so wird diese Anforderung an dieser Stelle als erfüllt angesehen.

Die nicht-funktionalen Anforderungen bedürfen einer genaueren Analyse.

Anpassbarkeit Die Anpassbarkeit an individuelle Bedürfnisse der Institution hängt primär nicht vom Konzept, sondern von der Implementierung ab. Erfolgt die Implementierung als Open-Source-Projekt mit einer freien Lizenz, steht Anpassungen und Erweiterungen nichts im Wege.

Nutzung bisheriger Rechteverwaltung Das System nutzt eine dedizierte separate AAI oder verwendet die interne Rechteverwaltung der Virtualisierungslösung. Die Nutzer können weiterhin die virtualisiererspezifischen Management-Anwendungen, wie zum Beispiel den vSphere Client benutzen - vorausgesetzt die Administratoren gewähren netzseitig Zugriff. Bei Bedarf können den Endnutzern, Resourcepool-Nutzern und Administratoren auch Rechte für Funktionen gewährt werden, die im Portal nicht zur Verfügung stehen.

Schnittstellen/APIs In dieser Arbeit wurde als Basisvariante eine multifunktionale SQL-basierte Schnittstelle beschrieben, die durch einem direkten Zugriff auf die Datenbank des Portals realisiert ist. Für das Accounting und die Anbindung der ITSM-Software können dedizierte APIs je nach Anforderungen der jeweiligen Institution Sinn machen. Die Anforderung wird deshalb an dieser Stelle als mit Einschränkungen erfüllt bewertet.

Skalierbarkeit Das System ist hoch skalierbar. Es können alle Komponenten, also Datenbank, Portal und Adapter, auf einem Server installiert werden, genauso ist es möglich jede Komponente auf einem separatem Server zu installieren. Es kann sogar jede Komponente auf mehreren Server parallel betrieben werden. Die Lastverteilung für das Webportal kann dann im einfachsten Fall DNS-basiert erfolgen. Die Lastverteilung für die Adapter kann dadurch realisiert werden, dass das Webportal für jede Aktion einen Adapter zufällig auswählt. Dies ist dank der zustandslosen Adapter-API problemlos möglich. Die Art wie mehrere Datenbankserver parallel betrieben werden können, hängt von der verwendeten Software ab, die meisten Datenbanksysteme bieten hierfür irgendeine Form von Replikation an.

5. Konzept

Verfügbarkeit Die Eigenschaften des Systems die eine hohe Skalierbarkeit gewährleisten, ermöglichen auch eine hohe Verfügbarkeit. Dadurch, dass jede Komponente mehrfach parallel betrieben werden kann, ist ein hohes Maß an Ausfallsicherheit gegeben.

Die Unterstützung diverser Virtualisierer beziehungsweise die leichte Anpassbarkeit an neue Virtualisierer wird durch das Adapter-Konzept umgesetzt. Der Adapter ist eine relativ einfach aufgebaute Übersetzungskomponente, die schnell implementiert werden kann, das Portal selbst muss dabei nicht verändert werden.

Abschließend gibt die Tabelle 5.3 einen Überblick über die erfüllten Anforderungen.

Benutzerverwaltung	Passende Benutzergruppen werden unterstützt	✓
Bestell- und Bereitstellungs- vorgang	Bestellmöglichkeit besteht	✓
	Parameter frei wählbar	✓
	Vorlagen / fertige Angebote verfügbar	✓
	Einzelne Parameter bei Vorlagen modifizierbar	✓
	AGBs werden angezeigt	✓
	Preis wird angezeigt	✓
	Administrator kann Antrag ablehnen	✓
	Administrator kann Alternative vorschlagen	✓
	Vor Bereitstellung ist manueller Eingriff möglich	✓
Benachrichtigungen werden verschickt	✓	
Betrieb	Steuerung des Betriebs möglich	✓
	VMs können gestartet/gestoppt/resetet werden	✓
	Snapshots können angelegt/verwaltet werden	✓
	Mounten von Nutzer-ISO-Images möglich	✓
	Status-/Performance-Daten sind abrufbar	✓
	Fertige VMs können modifiziert werden	✓
	Zugriff auf die Konsole ist möglich	✓
	Löschungen können beantragt werden	✓
Ressourcepools	Ressourcepools werden unterstützt	✓
	VMs können geklont werden	✓
	Verbrauchte Ressourcen werden angezeigt	✓
Nicht-funktionale Anforderungen	System kann einfach angepasst werden	✓
	Bisherige Rechteverwaltung wird verwendet	✓
	Schnittstellen/APIs sind verfügbar	(✓)
	System ist skalierbar	✓
	System kann hochverfügbar aufgesetzt werden	✓
Virtualisierer- unterstützung	vSphere wird unterstützt	✓
	ESXi-Server werden unterstützt	✓
	Anderer Virtualisierer werden unterstützt	✓

Tabelle 5.3.: Anforderungserfüllung des Konzepts

5. Konzept

6. Implementierung

Im Folgenden soll die prototypische Implementierung des Webportals und des Adapters für vSphere dokumentiert werden. Der primäre Nutzer der Implementierung soll dabei das LRZ sein, aber auch ein Einsatz an der LMU soll möglich sein. Zunächst wird der Umfang dieser festgelegt, anschließend werden die Frameworks und Bibliotheken, die für die Implementierung verwendet wurden, beschrieben, einzelne Komponenten der Implementierung erklärt und erläutert wie man das Webportal installiert und konfiguriert. Zum Schluss wird die Bedienung des Portals dargestellt und auf das Sicherheitskonzept eingegangen. Abschließend findet sich wieder eine Bewertung, in wie weit die Implementierung die Anforderungen erfüllt.

6.1. Umfang der Implementierung

Da es aus zeitlichen Gründen nicht möglich ist sämtliche im Konzept beschriebene Funktionalität zu implementieren, muss der Umfang der Implementierung eingeschränkt werden. Dabei sollten die Funktionen, auf die verzichtet wird, nach Möglichkeit so ausgewählt werden, dass weiterhin aufgezeigt werden kann, dass das Portal den gesamten Dienstlebenszyklus abdeckt.

Als erster großer Funktionskomplex dieser Art wurde die Unterstützung von Ressourcepools ausgewählt. Beim LRZ kommen Ressourcepools zum gegebenen Zeitpunkt nicht zum Einsatz, deren Unterstützung wäre nach Aussage der LRZ-Administratoren zwar eine Funktion, die man gerne hätte, welche aber nicht unbedingt notwendig ist und auf welche zunächst verzichtet werden kann.

Eine weitere Funktion die nicht implementiert werden muss, ist das Hochladen eigener ISO-Images. Zunächst erscheint die Möglichkeit eigene ISO-Images einzuhängen unverzichtbar, um auf einer Unattended VM ein Betriebssystem zu installieren. Doch da sich die Implementierung auf vSphere konzentriert muss darauf nicht verzichtet werden. Die vSphere Konsole ist nämlich weit mehr als eine einfache Konsole. Sie bietet diverse zusätzliche Funktionen, wie zum Beispiel die Möglichkeit die VM zu starten und zu stoppen oder lokale ISO-Images zu mounten. Somit ist die Möglichkeit eigene ISO-Images einzubinden gewährleistet und auf die entsprechende Funktionalität im Portal selbst kann verzichtet werden.

Außerdem wird auch noch auf die für den Betrieb nicht so essentiellen Funktionen zum Anlegen und Verwalten von Snapshots verzichtet.

Das Modell der Implementierung wird eine unbegrenzte Anzahl an Festplatten und Netzwerkadaptern pro Konfiguration unterstützen, in der View, also in den diversen Formularen, wird die Anzahl aber auf jeweils zwei beschränkt.

Zuallerletzt wird das Portal so implementiert, dass es nur einen einzelnen Adapter ansprechen kann. Das Konzept sieht eigentlich vor, dass aus Gründen der Skalierbarkeit und Ausfallsicherheit mehrere Adapter gleichzeitig betrieben und per Round-Robin-Verfahren angesprochen werden können.

Die hier beschriebenen Einschränkungen betreffen sowohl die Webportal-Komponente als auch den Adapter. In diesem werden nur die Teile der API implementiert, die für die zu implementierenden Funktionen benötigt werden.

6.2. Verwendete Frameworks und Bibliotheken

In diesem Kapitel werden kurz die einzelnen Frameworks und Bibliotheken aufgezählt, welche zur Implementierung des Portals und des Adapters verwendet wurden.

6.2.1. Für das Portal verwendete Frameworks und Bibliotheken

Das Webportal wird auf Grund der größten Erfahrung mit dieser Sprache in PHP implementiert. Es stellte sich die Frage welches Framework eingesetzt werden soll. Hierbei wurde vom LRZ der Wunsch geäußert Symfony (<http://www.symfony-project.org/>) einzusetzen, da dieses bereits bei vielen anderen Projekten am LRZ zum Einsatz kam, und somit den Mitarbeitern eine Weiterentwicklung am Portal erleichtern würde. Da es ansonsten keinerlei Argumente gegen den Einsatz von Symfony gab, wurde dieses Framework schließlich auch verwendet. Es handelt sich dabei um ein sich an Ruby on Rails orientierendes Framework, welches dem Model-View-Controller-Pattern folgt. Symfony verfügt über ein objektrelationales Mapping (ORM), eine Template-Engine und eine Session-Verwaltung, es unterstützt die Nutzung von YAML-Konfigurationsdateien, die Prüfung von Formularen (auch auf Schadcode) und Request-Handling für hübsche URLs. Eine vom Portal nicht verwendete weitere Eigenschaft von Symfony ist die Unterstützung von mehrsprachigen Anwendungen. Eine Erweiterung des Portals in der Hinsicht ist dank Symfony also relativ einfach möglich.

Zu dem Zeitpunkt als mit der Implementierung begonnen wurde, war bereits Symfony 2 als Beta verfügbar. Es wurde überlegt, ob bereits diese Version für die Implementierung verwendet werden soll, da Symfony 1.4 nur noch bis November 2012 unterstützt wird. Auf Grund der deutlich besseren und umfangreicheren Dokumentation für Symfony 1.4, fiel die Entscheidung schließlich aber dennoch auf diese.

Neben Symfony wird aus dem PHP Extension and Application Repository PEAR das Paket HTTP_Request2 (http://pear.php.net/package/HTTP_Request2) verwendet, welches für die Kommunikation des Portals mit dem Adapter benötigt wird.

Um das Portal möglichst dynamisch zu gestalten und dem Benutzer einen hohen Grad an Usability zu bieten, ist der Einsatz von Ajax nicht vermeidbar. Folglich wird auch ein Javascript-Framework benötigt. Hierbei fiel die Wahl auf das weit verbreitete jQuery (<http://jquery.com/>).

6.2.2. Für den Adapter verwendete Frameworks und Bibliotheken

Der Adapter wurde in Perl implementiert. Zum Einsatz kommt hier natürlich das vSphere SDK for Perl (<http://vmware.com/support/developer/viperltoolkit>), welches eine Schnittstelle zur vSphere API bildet. Verwendet wird hierbei die Version 4.1.

Des weiteren kommt das Framework Perl Dancer (<http://perldancer.org/>) zum Einsatz. Dabei handelt es sich um ein sehr einfaches und kleines Framework, welches optimal dafür geeignet ist eine REST-API zu realisieren. Sobald man Dancer in ein Perl-Skript importiert, wird dieses Skript zu einer Webanwendung. In dem Skript müssen lediglich Routen definiert werden. Jede Route besteht aus einer HTTP-Methode, einem Pfad-Pattern und

einem Codeblock. Eine Dancer-Anwendung kann direkt ausgeführt und sofort über HTTP aufgerufen werden. Bei einem Aufruf wird entsprechend der Routen-Konfiguration der passende Codeblock ausgeführt und von diesem die Antwort auf die Anfrage geliefert.

Das Dancer-Skript wird zudem um Plack (<http://plackperl.org/>) erweitert um die Unterstützung für HTTP-Authorization hinzuzufügen.

6.3. Implementierungsdetails

Das von der Implementierung verwendete Datenbankschema basiert auf dem Informationsmodell. Wie in Abbildung 6.1 zu sehen ist, wurden allerdings Änderungen bei der Abbildung der Netzwerkadapter vorgenommen. Die Tabelle NIC wird um eine Tabelle IPConfig ergänzt, welche die zu einem NIC gehörende MAC-, IPv4- und IPv6-Adresse enthält. Der Grund für die Abspaltung dieser Attribute ist der Folgende: Da die Requests archiviert werden, unter anderem um ein Accounting zu ermöglichen, müssen auch die mit diesen verknüpften Configurations archiviert werden, was sich wiederum auf die HDDs und NICs fortplant. Die verschiedenen Adressen werden allerdings erst dann zugewiesen, wenn eine VM über den Virtualisierer angelegt wird. Wird eine VM gelöscht, müssen auch die zugewiesenen Adressen gelöscht werden, damit diese einer anderen VM erneut zugewiesen werden können. In Folge dessen würde die Datenbank sehr viele Einträge mit nicht gesetzten Attributen enthalten. Durch die Auslagerung der IP-Konfiguration in eine separate Tabelle verhindert man nicht nur diese, sondern vereinfacht auch das Management erheblich. Wird eine VM gelöscht, kann einfach die dazugehörige IPConfig gelöscht werden. Ändert sich die Konfiguration einer VM, können die IPConfigs einfach mit den NICs der neuen Konfiguration verknüpft werden, so dass sich die MAC- und IP-Adressen der VM nicht ändern.

Im nächsten Schritt wurde mit Symfony-Hilfsmitteln aus diesem Datenbankschema ein objektorientiertes Modell erstellt und anschließend um benötigte Funktionalität erweitert. Da das Datenbankschema Ressourcepools berücksichtigt, wurden automatisch auch die entsprechenden Basisklassen erzeugt. Auf eine weitere Modifikation und Erweiterung dieser wurde entsprechend des festgelegten Implementierungsumfangs verzichtet. Diese Klassen können aber jederzeit verwendet werden, um das Portal entsprechend zu ergänzen.

Neben diesen direkt aus dem Datenmodell abgeleiteten Klassen, wurden zusätzlich die folgenden Klassen implementiert:

adapterCom Diese Klasse ist für die Kommunikation des Portals mit dem Adapter zuständig. Zum Einsatz kommt hierbei das bereits erwähnte PEAR-Paket HTTP_Request2. Für die Authentifizierung am Adapter und somit im weiteren am Virtualisierer werden die in der Session gespeicherten Login-Parameter verwendet.

priceCalc Diese Klasse ist für die Berechnung der Preise einer VM-Konfiguration zuständig. Die Preisberechnung basiert dabei auf den in der Konfiguration angegebenen Komponentenpreisen. Der Gesamtpreis einer VM wird dabei durch eine Addition des Grundpreises für den jeweiligen Dienstyp und den Stückpreisen der einzelnen Komponenten multipliziert mit der jeweiligen Anzahl (Anzahl CPUs, Megabyte RAM, Megabyte HDD) berechnet. Im Falle des produktiven Einsatzes sollte diese Berechnung natürlich nicht intern im Portal erfolgen, sondern vom Accountingsystem übernommen werden. Die priceCalc-Klasse muss zu diesem Zweck entsprechend modifiziert werden, so dass eine Preisanfrage an das Accounting-System gestellt wird.

6. Implementierung

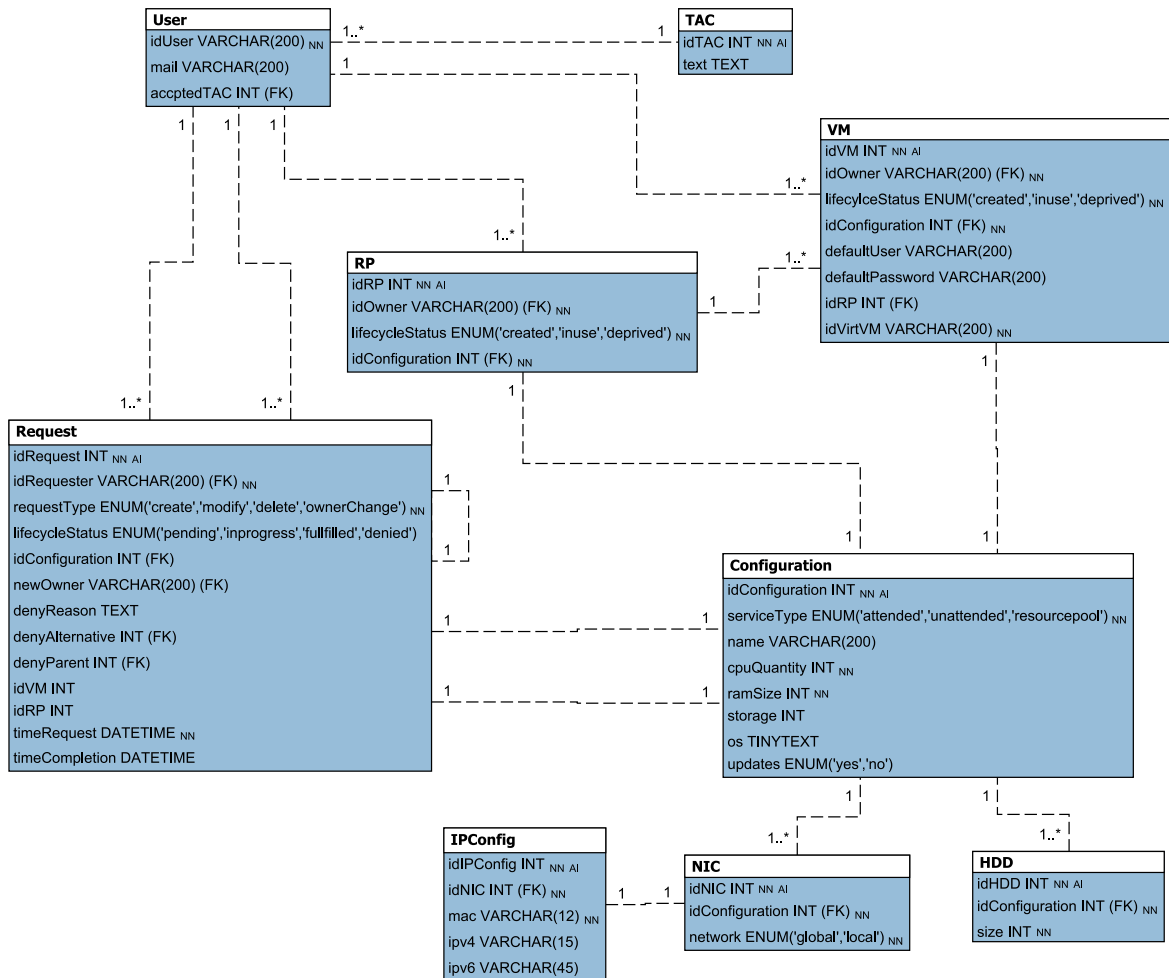


Abbildung 6.1.: Datenbankschema

Auf Controller- und View-Ebene wurden vier Module erstellt:

user Dieses Modul ist für die Login- und Logout-Vorgänge zuständig. Bei einem Loginvorgang wird die Benutzergruppe festgelegt und die dazu passende GUI angezeigt. Auch das Ändern der E-Mail-Adresse geschieht über das user-Modul.

request Dieses Modul ist für die Generierung jeglicher Art von Anträgen zuständig.

approval Dieses Modul kann im Gegensatz zu den anderen Modulen nur von den Administratoren verwendet werden. Es dient der Bearbeitung sämtlicher Anträge und im Rahmen dessen auch der Bereitstellung und Modifikation von VMs.

vms Dieses Modul ist für die Darstellung von Informationen über existierende VMs und deren Steuerung zuständig.

Für jedes Modul gibt es symfony-typisch eine Datei `actions.class.php`, in welcher der Controller implementiert ist und für jede Aktion ein separates Template, das die View beschreibt.

In den folgenden Unterkapiteln werden einzelne Bestandteile des Portals beschrieben, welche bei der Implementierung eine besondere Herausforderung waren oder die zusätzlicher

genauerer Erläuterungen bedürfen.

6.3.1. Realisierung der Konfigurationsvorlagen

Die Konfigurationsvorlagen wurden als eine vollständig clientseitige Funktionalität in JavaScript implementiert. Der Client ruft hierzu per Ajax-Request eine XML-Datei ab, welche die verschiedenen Vorlagen beschreibt und erstellt aus diesen Daten eine Auswahlliste. Bei Wahl eines Templates werden die Parameter des Templates in die einzelnen Formularfelder geschrieben. Der Nutzer kann diese Parameter dann daraufhin weiterhin verändern oder ein anderes Template wählen. Während des gesamten Vorgangs findet keine weitere Kommunikation mit dem Server statt. Wird der Antrag schließlich abgeschickt, enthält dieser keine Information darüber, ob dieser Antrag auf einem Template basiert oder nicht. Diese Information ist auch unwichtig, zumal der Nutzer ein Template so stark modifizieren kann, dass von seiner ursprünglichen Konfiguration nichts mehr übrig ist.

6.3.2. Realisierung der Konsole

Die meisten Virtualisierungslösungen verwenden für die Konsole das VNC-Protokoll. Dank dieses offenen Protokolls und vieler bereits dafür vorhandener Browser-Plugins ist es ein Leichtes die Konsole im Portal zu realisieren.

Bei VMware ist dies allerdings nicht der Fall. Für die Konsole wird ein proprietäres Protokoll verwendet, das neben der eigentlichen Konsole, also dem Übertragen von Bildschirmhalten und Tastatur- und Mauseingaben, auch weitere Funktionen bereitstellt. So kann man über die VMware-Konsole auch VMs starten und stoppen und lokale ISO-Images mounten. Da das Protokoll proprietär ist, gibt es dafür auch keine frei verfügbaren Plugins. Die einzige Möglichkeit also die Konsole auf die gewünschte Weise zu realisieren, liegt in der Verwendung des Plugins, welches von WebAccess verwendet wird. Mit der bereits eingestellten Weiterentwicklung von WebAccess geht allerdings einher, dass auch dieses Plugin nicht mehr weiterentwickelt wird. Die Version für den Internet Explorer lässt sich zwar weiterhin problemlos installieren, bei der Firefox-Version ist dies allerdings nicht mehr der Fall. Hier müssen die im Plugin gespeicherten Informationen über die unterstützten Firefox-Versionen modifiziert werden. Führt man diese Modifikation durch, gelingt die Installation auch unter einem aktuellen Firefox. Dann stößt man aber direkt auf das nächste Problem. Eine Einbindung des Plugins in eigene Software war von VMware nicht vorgesehen, so dass es hierzu keinerlei Dokumentation gibt. Da das Plugin nicht mehr unterstützt wird, ist mit einer Änderung dieser Situation nicht zu rechnen. Die letzte verbleibende Möglichkeit ist, die Einbindung des Plugins unter WebAccess zu untersuchen und zu versuchen daraus abzuleiten, wie man das Plugin im Webportal initialisieren müsste. Dies wurde über einen längeren Zeitraum probiert, ist aber in mehreren Versuchen misslungen. Aus diesem Grund wurde schließlich von der Integration der Konsole über ein Plugin Abstand genommen und diese stattdessen über eine Standalone-Anwendung realisiert. Die Funktionsweise der VMware-Standalone-Konsole ist ausreichend dokumentiert und deshalb problemlos zu bewerkstelligen. Die Bereitstellung der Standalone-Konsole wird dabei genauso wie im Falle des Plugins über den Adapter realisiert. Dieser stellt an Stelle eines HTML-Fragments, das das Plugin initialisieren soll, ein HTML-Fragment mit dem Download-Link der Standalone-Anwendung bereit. Außerdem enthält dieses HTML-Fragment einen an die konkrete VM angepassten

6. Implementierung

Terminal-Befehl, über den der Nutzer die Konsole starten muss. Auf diese Weise wird versucht den Komfort-Verlust, der sich durch den Rückgriff auf ein separates Programm ergibt, so gering wie möglich zu halten.

Unabhängig davon ob weiterhin die Standalone-Anwendung verwendet wird oder ob in Zukunft die Initialisierung des Plugins doch noch gelingt, ist die rechtliche Lage bei Verwendung dieser Software im Rahmen des Webportals durch einen Juristen zu prüfen.

6.4. Installation und Konfiguration

In diesem Kapitel wird beschrieben, wie man das Portal und den Adapter installiert und konfiguriert. Dabei wird davon ausgegangen, dass bereits ein funktionsfähiges Linuxsystem mit installiertem Apache-Webserver und MySQL-Datenbank, beziehungsweise im Falle des Adapters mit installierten Perl vorhanden ist. Da die hier beschriebene Installation vor allem Demonstrationszwecken dienen soll, wird außerdem auf die Konfiguration der SSL-Verschlüsselung verzichtet. Bei einem produktiven Einsatz sollte der Apache-Webserver natürlich entsprechend konfiguriert werden. Verbindungsversuche auf dem unverschlüsselten Port 80 sollten auf den verschlüsselten Port, in der Regel 443, umgeleitet werden. Der erwähnte Apache-Webserver und die MySQL-Datenbank sind übrigens nicht verpflichtend, es handelt sich dabei um die Software unter welcher das Portal entwickelt wurde und für die die folgende Anleitung geschrieben wurde. Bei entsprechender Konfiguration wird das Portal auch unter einem anderem Webserver (z.B. lighttpd) oder einer anderen Datenbank (z.B. PostgreSQL) funktionieren. Mit Symfony können für alle unterstützten Datenbanksysteme automatisch basierend auf dem Modell die entsprechenden Tabellen erzeugt werden.

6.4.1. Konfiguration des vCenters

Im vCenter muss ein Cluster und ein Resourcepool angelegt werden, in welchen das Portal die VMs erzeugen kann.

Außerdem müssen drei neue Rollen angelegt und mit passenden Rechten versehen werden. Die Rollen Portal-Administrator und Portal-Endnutzer müssen dann den entsprechenden Nutzern zugewiesen werden.

Portal-Administratoren Diese Rolle kann aus einer Kopie der normalen Administrator-Rolle erzeugt werden. Sie hat sämtliche Berechtigungen und wird auf dem für das Portal vorgesehenem Cluster den Nutzern zugewiesen, die die Anträge bearbeiten sollen.

Portal-Endnutzer Diese Rolle dient lediglich der Feststellung, ob ein Nutzer zur Nutzung des Webportals berechtigt ist. Im vCenter selbst soll diese Rolle deshalb über keinerlei Rechte verfügen. Sie wird ebenfalls auf dem für das Portal vorgesehenem Cluster den Nutzern zugewiesen, die das Portal als Endnutzer verwenden sollen.

VM-Endnutzer Diese Rolle wird jeweils für eine VM an den jeweiligen Besitzer vergeben. Sie entspricht der standardmäßig vorhandenen Beispielrolle "Benutzer virtueller Maschinen" mit zusätzlichen Berechtigungen zum Erstellen, Umbenennen, Entfernen und Wiederherstellen von Snapshots, sowie der Berechtigung zum Entfernen der VM aus der Bestandsliste. Die Snapshot-Berechtigungen werden zwar auf Grund des eingeschränkten Umfangs der Implementierung nicht benötigt, seien hier aber der Vollständigkeit

halber erwähnt. Die Rolle wird zunächst keinem Nutzer zugewiesen, dies übernimmt das Portal beziehungsweise der Adapter im Rahmen der Zuweisung einer VM.

6.4.2. Installation und Konfiguration des Portals

Neben den bereits genannten Voraussetzungen, muss für ein korrektes Funktionieren des Portals das PEAR-Paket HTTP_Request2 installiert werden. Dies ist mit folgendem Befehl möglich:

```
| > pear install HTTP_Request2
```

Zunächst muss der Ordner `portal` auf das System an einen beliebigen Ort kopiert werden. Hierzu bietet sich entweder das Benutzerverzeichnis oder der Ordner `/opt` an. Letzterer wird in den folgenden Beispielen verwendet. Anschließend muss die Apache-Konfiguration angepasst werden. Man kann entweder für das Portal einen eigenen `vHost` anlegen oder, falls der Server sonst keine weiteren Aufgaben erfüllt, die Default-Konfiguration anpassen. Im Folgenden ist ein Ausschnitt aus der Konfiguration mit allen relevanten Einstellungen aufgelistet:

```
| DocumentRoot "/opt/portal/web"
| DirectoryIndex index.php
|
| <Directory "/opt/portal/web">
|     AllowOverride All
|     Order allow,deny
|     Allow from All
| </Directory>
|
| Alias /sf /opt/portal/lib/vendor/symfony/data/web/sf
| <Directory "/opt/portal/lib/vendor/symfony/data/web/sf">
|     AllowOverride All
|     Order allow,deny
|     Allow from All
| </Directory>
```

Anschließend muss unter MySQL eine neue Datenbank und ein neuer Benutzer angelegt werden. Dies kann über folgende Shell-Befehle erfolgen:

```
| > mysql -u root -p
| mysql> CREATE DATABASE vmmgmt;
| mysql> GRANT ALL PRIVILEGES ON vmmgmt.* TO vmmgmtuser@localhost
|     IDENTIFIED BY 'password';
| mysql> FLUSH PRIVILEGES;
```

Jetzt kann die Datenbank mit dem soeben gewählten Passwort in der Datenbank-Konfiguration des Portals eingetragen werden. Diese befindet sich in der Datei `/opt/portal/config/databases.yml`. Anschließend wechselt man in der Shell in den Ordner `/opt/portal` und führt dort folgende Befehle aus, welche die benötigten Tabellen in der Datenbank anlegen:

```
| > php symfony doctrine:build-sql
| > php symfony doctrine:insert-sql
```

6. Implementierung

Im gleichen Ordner wie die Datei `databases.yml` befindet sich die Datei `app.yml`. Hier können fast alle weiteren Einstellungen, die das Portal betreffen, vorgenommen werden. Die Datei ist in fünf Abschnitte unterteilt. Zwingend erforderlich ist im Abschnitt `adapter` die Angabe des Host und des Ports unter welchen der Adapter erreichbar ist. Wird dieser auf dem gleichen Server installiert, können die voreingestellten Werte belassen werden. Im Abschnitt `disabled` kann die Anzeige des Preies und der AGBs deaktiviert werden. Diese Funktionalität wird zwar am LRZ benötigt, in anderen Anwendungsfällen, wie zum Beispiel bei einem Einsatz am Lehrstuhl für Informatik der LMU, bei dem die VMs nicht abgerechnet werden, kann eine Deaktivierung sinnvoll sein. Im Abschnitt `limits` können Grenzwerte für die Anzahl der CPUs und die Größe des Arbeitsspeichers und der Festplatten definiert werden. Werden diese nicht eingehalten, wird ein Antrag gar nicht erst angenommen. Es ist sinnvoll wenigstens Mindestwerte zu definieren, so sollte ein VM zum Beispiel über mindestens eine CPU und über mindestens 128 MB RAM verfügen. Im Abschnitt `ips` können die IP-Adressbereiche für das lokale und das globale Netz konfiguriert werden. Unter `prices` können die Basis- und Komponentenpreise für die interne Preisberechnung des Portals angegeben werden. Am Ende dieser Konfigurationsdatei im Abschnitt `admin` können Benachrichtigungen für die Administratoren konfiguriert werden. Das Portal verfügt zwar für jeden Nutzer unabhängig von der Nutzergruppe über eine Kontaktadresse, es fehlt allerdings die Angabe, welche Nutzer Administratoren sind. Folglich könnte das Portal keine Benachrichtigungen an diese schicken, wenn neue Anträge eingehen oder sich neue Benutzer anmelden. Deshalb kann man in der Konfigurationsdatei eine Verteiler-Adresse angeben, an welche solche Nachrichten geschickt werden sollen.

Die dritte und letzte Datei, über die man die Konfiguration des Portals anpassen kann, ist `/opt/portal/web/xml/templates.xml`. Hierbei handelt es sich um die XML-Datei, die in Kapitel 6.3.1 erwähnt wurde. Das Portal ruft diese über einen Ajax-Request ab und erstellt auf deren Grundlage die Vorlagen für die Antragserstellung. Die Konfigurationsvorlagen in dieser Datei können nach Belieben modifiziert und erweitert werden. Die Art und Weise wie diese zu beschreiben sind, kann den Beispielen in dieser Datei entnommen werden.

6.4.3. Installation und Konfiguration des Adapters

Die Installation und Konfiguration des vSphere-Adapters ist weitaus einfacher zu bewerkstelligen. Bevor mit dieser begonnen werden kann, muss aber zuerst das vSphere SDK for Perl in der Version 4.1 installiert werden. Alle benötigten Pakete und der genaue Ablauf sind im vSphere SDK for Perl Installation Guide beschrieben. [sdk10]

Anschließend muss der Ordner `adapter` wie bereits im Falle des Portals auf den Server kopiert werden. Hierbei bieten sich wieder die gleichen Orte an, in den Beispielen wird also wieder vom Ordner `/opt` ausgegangen.

Im nächsten Schritt müssen einige Konfigurationsvariablen in der Datei `/opt/adapter/config.yml` gesetzt werden. Angegeben werden müssen die Adresse des vCenters, die Bezeichnung des Datacenters und Clusters in welchem die VMs angelegt werden sollen, die Bezeichnung des Speichers für die VM-Daten, die Bezeichnung des virtuellen Switches und der virtuellen Portgruppe, an welche die VMs angeschlossen werden sollen, die Namen für die Templates aus denen die Attended VMs durch Klonen erzeugt werden sollen, die URL von welcher die Konsolenanwendung heruntergeladen werden kann und die IDs der verschiedenen Rollen.

Anschließend kann der Adapter über den Befehl `plackup /opt/adapter/bin/app.pl` gestartet werden. Mit angefügten Ampersand kann man die Anwendung im Hintergrund starten.

Wird der Adapter in dieser einfachen Form ausgeführt, bietet er keine Unterstützung für SSL-Verschlüsselung, ist also nur für den Einsatz auf dem gleichen Host wie das Portal geeignet. Für den Einsatz auf einem separaten Host empfiehlt es sich Plack über den Apache-Webserver anzusprechen und dessen SSL-Unterstützung zu nutzen. Wie dies funktioniert ist unter [PDD] beschrieben.

6.4.4. Installation und Konfiguration eines DHCP-Servers

Der DHCP-Server ist eigentlich nicht Teil der Implementierung des Webportals. Um die Zuweisung der IP-Adressen zu realisieren, sollte der eingesetzte DHCP-Server einfach über die multifunktionale Portal-API die Zuordnung von MAC- und IP-Adressen abrufen. Es besteht allerdings das Problem, dass die bei den diversen Linuxdistributionen enthaltenen DHCP-Server — in der Regel handelt es sich dabei um den ISC DHCPD — eine solche Funktionalität nicht bieten. Deshalb wird an dieser Stelle eine mögliche Lösung für dieses Problem vorgestellt.

Um die gewünschte Funktionalität zu erhalten sollte man auf den Quellcode des ISC DHCPD einen Patch von Alex Bligh anwenden und die Software anschließend neu kompilieren. [Bli10] Der Patch ergänzt den ISC DHCPD mit Hilfe von libdbi um die Funktionalität, die Konfiguration für statische IP-Adressen aus einer Datenbank abzurufen. Hierfür wird die Konfigurationsdatei des Servers um die Angabe eines SQL-Servers, von Zugangsdaten und eines SQL-Statements ergänzt. Das SQL-Statement muss für den Einsatz mit dem Webportal wie folgt aussehen:

```
| SELECT ipv4 AS 'fixed-address' FROM IPConfig WHERE mac = '%h'
```

Des Weiteren muss in der Konfiguration auf die übliche Art und Weise das globale und lokale Netz konfiguriert werden, damit der DHCP-Server neben der IP-Adresse auch die Netzmaske und das Standard-Gateway zuweisen kann.

Die übrige Funktionalität des DHCP-Servers wird durch diesen Patch nicht beeinflusst, man kann ihn also wie gewohnt konfigurieren und zum Beispiel auch für die Zuweisung von dynamischen IPs an andere Systeme als die VMs einsetzen.

Der Patch wird vom Autor selbst als nicht für den produktiven Einsatz geeignet beschrieben. Folglich sollte er vor einem solchen nochmals geprüft oder neu implementiert werden.

6.5. Bedienkonzept des Webportals

An dieser Stelle wird die Bedienung des Webportals beschrieben, wodurch nochmals deutlich wird welche Funktionen das Webportal bietet und welche Anwendungsfälle abgedeckt sind.

Ein Endnutzer gelangt nach einem erfolgreichen Login zunächst zur Übersichtsseite, auf der er auf einen Blick alle seine VMs sehen kann (siehe Abbildung 6.2). Unterhalb dieser Übersicht befindet sich eine Zusammenfassung seiner aktuell laufenden beziehungsweise abgelehnten Anträge. Er sieht zu jeder VM die aktuellen Ausstattungsparameter und den aktuellen Zustand der VM. Letzterer wird alle paar Sekunden mittels eines Ajax-Requests aktualisiert. Über die Icons auf der rechten Seite kann er die VM starten, stoppen, neu starten, die Konsole aufrufen, Statistiken abrufen, die Geschichte der VM einsehen, einen

6. Implementierung

Modifikationsantrag erstellen, einen Antrag auf einen Besitzerwechsel erstellen und die VM kündigen.

Die Geschichte der VM ist eine chronologische Auflistung aller die VM betreffenden Anträge. Sie enthält alle Informationen zu den Bereitstellungsanträgen, zu Modifikationsanträgen und zu Besitzerwechseln. Es werden sowohl erfolgreiche, als auch abgelehnte Anträge inklusive Ablehnungsgründen und eventuell vorhandenen Alternativvorschlägen angezeigt. Zu jedem Antrag ist der Einreichungszeitpunkt und der Zeitpunkt der Erfüllung beziehungsweise Ablehnung einsehbar. An Hand dieser Funktion lässt sich demonstrieren, dass alle für ein Accounting benötigten Informationen verfügbar sind.

Über den Menüpunkt “Antrag erstellen” kann der Nutzer einen Antrag auf eine neue VM einreichen (siehe Abbildung 6.4). Dabei kann er auf Konfigurationsvorlagen zurückgreifen und diese auch modifizieren. Das Formular passt sich dabei automatisch dem gewählten Diensttyp an. Betriebssystem und Updates werden also nur bei der Beantragung einer Attended VM angezeigt. Die Formulareingaben werden auf Plausibilität geprüft und müssen innerhalb der in der Konfiguration angegebenen Grenzwerte liegen. Bevor der Antrag angenommen wird, müssen in einem weiteren Schritt Preis und AGBs akzeptiert werden, sofern dies in der Konfiguration aktiviert ist.

Wird ein Antrag abgelehnt erfährt der Nutzer darüber, von der Mail-Benachrichtigung abgesehen, in der Antragsübersicht, wo auch der Grund für die Ablehnung angegeben ist und wo sich gegebenenfalls auch ein Link zu einem Alternativvorschlag befindet. Über diesen gelangt er zu einem Formular, über welches er den Alternativvorschlag direkt als neuen Antrag einreichen oder nochmals modifizieren kann (siehe Abbildung 6.6).

Des weiteren kann der Nutzer seine E-Mail-Adresse ändern und sich ausloggen.

Die Oberfläche der Administratoren ist sehr ähnlich aufgebaut. Dazu muss man wissen, dass in der Implementierung Administratoren selbst VMs beantragen und besitzen können. Die Anträge können sie sich selber genehmigen. Ein Administrator sieht in der VM-Übersicht neben seinen eigenen VMs, alle VMs der Endnutzer und kann auf diesen dieselben Steuerfunktionen ausführen.

In der Administrator-Oberfläche ist der zusätzliche Menüpunkt “Anträge genehmigen” verfügbar. Über diesen gelangt er zu einer Übersicht aller Anträge, welche nach Antragsart unterteilt sind (siehe Abbildung 6.3). Zu finden sind hier Anträge für die Bereitstellung neuer VMs, Anträge für Modifizierungen und Anträge für einen Besitzerwechsel. Letztere sind solche, die bereits durch den potentiellen neuen Nutzer bestätigt wurden. Außerdem befindet sich hier eine Auflistung von VMs die genehmigt, aber noch nicht bereitgestellt wurden, weil zum Beispiel ein manueller Eingriff erforderlich war. Wählt der Administrator einen Antrag aus, bekommt er alle Details des Antrags angezeigt und kann über das weitere Vorgehen entscheiden (siehe Abbildung 6.5).

6.6. Sicherheitskonzept / Erreichbarkeit der Komponenten

Ein Ziel des Management-Portals ist es den Endnutzern keinen direkten Zugriff auf das vCenter zu gewähren und es so vor Angriffen zu schützen. Das Webportals alleine leistet aber noch keinen Schutz, es bietet lediglich eine Schnittstelle für die Nutzer an, so dass diese keinen direkten Zugriff auf die zentrale Managementkomponente, also zum Beispiel das vCenter, benötigen. Die Abschirmung muss durch eine entsprechende Konfiguration der Firewalls erfolgen. Auf diese Weise kann man den Endnutzern (und wenn man nur auf diese

VM-Management-Portal

[VM-ÜBERSICHT](#)
[ANTRAG ERSTELLEN](#)
[E-MAIL-ADRESSE ÄNDERN](#)

[LOGOUT](#)

Ihre VMs

	Name	Diensttyp	CPUs	RAM	1.HDD	2.HDD	OS	Updates	1.NIC	2.NIC	Status	Aktionen
<input type="checkbox"/>	Dev-VM	Unattended VM	1	128	1024				192.168.1.1 lokal		gestartet	
<input type="checkbox"/>	Mini-VM 23	Unattended VM	1	128	1024				192.168.1.2 lokal		gestartet	
<input type="checkbox"/>	Experimentier-VM	Unattended VM	4	2048	10000				192.168.1.4 lokal	10.10.0.2 global	gestoppt	

Ihre laufenden Anträge

Antragsart	Name	Diensttyp	CPUs	RAM	1.HDD	2.HDD	OS	Updates	1.NIC	2.NIC	Status
Besitzerwechsel	Dev-VM	Unattended VM	1	128	1024				lokal		in Bearbeitung
Modifikation	Dev-VM	Unattended VM	2	256	1024				lokal		abgelehnt Begründung: Zur Zeit keine Ausreichenden Ressourcen verfügbar.
Bereitstellung	Windows-Test-Server	Attended VM	2	2048	20000		Windows	ja	global		eingereicht
Bereitstellung	Test-VM	Attended VM	1	128	4096	16000	Linux	ja	lokal		abgelehnt Begründung: Nur mit kleinerer 2. Platte zu haben. Alternativvorschlag

Abbildung 6.2.: Übersicht über die bestehenden VMs mit diversen Steuermöglichkeiten und Auftragsübersicht

VM-Management-Portal

[VM-ÜBERSICHT](#)
[ANTRAG ERSTELLEN](#)
[ANTRÄGE GENEHMIGEN](#)
[E-MAIL-ADRESSE ÄNDERN](#)

[LOGOUT](#)

Anträge für neue VMs

Nutzer	Diensttyp	CPUs	RAM	1.HDD	2.HDD	OS	Updates	1.NIC	2.NIC	Aktion
TestUser1	Attended VM	1	128	4096	16000	Linux	ja	lokal		Antrag bearbeiten
TestUser1	Unattended VM	4	2048	10000				lokal	global	Antrag bearbeiten
TestUser1	Attended VM	2	2048	20000		Windows	ja	global		Antrag bearbeiten
TestUser2	Unattended VM	1	256	1024				lokal		Antrag bearbeiten

Anträge für Modifizierungen

Nutzer	Diensttyp	CPUs alt CPUs neu	RAM alt RAM neu	1.HDD alt 1.HDD neu	2.HDD alt 2.HDD neu	OS	Updates alt Updates neu	1.NIC alt 1.NIC neu	2.NIC alt 2.NIC neu	Aktion
TestUser1	Unattended VM	1 2	128 256	1024 1024				lokal lokal		Antrag bearbeiten

Erstellte aber noch nicht bereitgestellte VMs

Nutzer	Diensttyp	CPUs	RAM	1.HDD	2.HDD	OS	Updates	1.NIC	2.NIC	Aktion
TestUser1	Unattended VM	1	128	1024				lokal		Bereitstellen
TestUser2	Unattended VM	1	128	40000				lokal		Bereitstellen

Anträge für Besitzerwechsel

Alter Besitzer	Neuer Besitzer	Diensttyp	CPUs	RAM	1.HDD	2.HDD	OS	Updates	1.NIC	2.NIC	Aktion
TestUser1	TestUser2	Unattended VM	1	128	1024				lokal		Antrag bearbeiten

Abbildung 6.3.: Überblick eines Administrators über die eingereichten Anträge, aufgeteilt in die verschiedenen Antragsarten

Neuer Antrag

Konfigurationsvorlagen

Diensttyp*

Name*

CPU-Anzahl*

RAM-Größe (MB)*

1. Festplatte (MB)*

2. Festplatte (MB)

Betriebssystem*

Automatische Updates*

Erreichbarkeit des 1. NICs*

Erreichbarkeit des 2. NICs

Abbildung 6.4.: Antragsformular für neue VMs

Bereitstellungsantrag genehmigen

Diensttyp* Attended VM

CPU-Anzahl* 1

RAM-Größe (MB)* 512

1. Festplatte (MB)* 10000

2. Festplatte (MB) 20000

Betriebssystem* Linux

Automatische Updates* nein

Erreichbarkeit des 1. NICs* global

Erreichbarkeit des 2. NICs

Abbildung 6.5.: Auftragsbearbeitung mit Auswahlmöglichkeiten

Alternativen Antrag einreichen

Ihr alter Antrag für die VM "Test-VM" wurde von einem Administrator mit der folgenden Begründung abgelehnt:
Nur mit kleinerer 2. Platte zu haben.

Der Administrator schlägt folgende alternative Konfiguration vor, welche Sie direkt übernehmen oder nochmals modifizieren können:

Diensttyp*

CPU-Anzahl*

RAM-Größe (MB)*

1. Festplatte (MB)*

2. Festplatte (MB)

Betriebssystem*

Automatische Updates*

Erreichbarkeit des 1. NICs*

Erreichbarkeit des 2. NICs

Abbildung 6.6.: Anzeige eines durch einen Administrator erstellten Alternativvorschlags mit Möglichkeit zur weiteren Modifikation

6. Implementierung

Lösung setzt und der Einsatz zum Beispiel des vSphere Clients entbehrlich ist, auch den Administratoren) den direkten Zugriff auf das vCenter entziehen. Das Webportal dient dabei quasi als Proxy. Ein direkter Zugriff auf die Hosts lässt sich im Gegensatz zum vCenter nicht unterbinden, da dieser zum Beispiel für die Konsolenzugriff nötig ist. Sollte man dies wirklich für nötig erachten, kann man an der Stelle über den Einsatz eines Circuit-Level-Gateways nachdenken.

Im Folgenden werden die Ports aufgezählt die für einen minimalen Betrieb in Zusammenhang mit dem Webportal benötigt werden. Anspruchsvollere vSphere Dienste, wie zum Beispiel vMotion, werden nicht in Betracht gezogen. Bei Bedarf müssen also weitere Ports freigeschaltet werden, zum Beispiel auch für den Maintenance-SSH-Zugang auf den ESXi-Servern.

- vCenter
 - Port 443/TCP: Auf diesem Port kommuniziert der vSphere Client mit dem vCenter, außerdem ist hier die alte WebAccess-Oberfläche verfügbar. Dieser Port muss für alle Nutzer erreichbar sein, die diese Management-Tools weiterhin nutzen sollen.
 - Port 902/TCP und 903/TCP: Über diese Ports kommunizieren die ESXi-Server mit dem vCenter. Der Zugriff aus dem ESXi-Netz muss also entsprechend freigeschaltet werden.
 - Port 8080/TCP (unverschlüsselt) und Port 8443/TCP (verschlüsselt): Auf diesen Ports werden die VMware VirtualCenter Management Services zur Verfügung gestellt. Über diese API nimmt das Management-Portal die Steuerung vor, weshalb diese Ports mindestens für den Adapter und eventuell weitere diese Schnittstelle nutzende Anwendungen freigegeben sein müssen.
- ESXi-Server
 - Port 902/TCP und 903/TCP: Über diese Ports kommuniziert das vCenter mit den ESXi-Servern. Außerdem erfolgt über diese Ports der Zugriff auf die Konsole und den Datastore. Uploads auf den Datastore werden theoretisch, was allerdings in der prototypischen Implementierung nicht umgesetzt ist, durch den Adapter geleitet. Wegen der Konsole müssen diese Ports aber dennoch für alle Portalnutzer erreichbar sein.
- Portalserver
 - Port 443/TCP: Der Server auf dem das Management-Portal läuft muss über diesen Port für den HTTPS-Zugriff erreichbar sein. Optional kann auch Port 80 freigeschaltet sein, falls das Portal unverschlüsselt angeboten werden soll oder falls auf diesem Port eine automatische Weiterleitung auf die verschlüsselte Verbindung erfolgen soll.
 - Port 3306/TCP (optional): Läuft die Datenbank auf dem gleichen Server und soll die multifunktionale Portal-API verfügbar sein, so muss dieser Port zusätzlich freigegeben sein.
- Adapter (optional)
 - Port 443/TCP: Wird der Adapter auf einem separatem Server betrieben und wurde für diesen eine verschlüsselte Verbindung eingerichtet, so muss dieser Ser-

ver über Port 443 erreichbar sein. Ohne Verschlüsselung ist der Adapter standardmäßig über Port 5000 zu erreichen.

- Datenbank (optional)
 - Port 3306/TCP: Falls die Datenbank auf einem separaten Server betrieben wird, so muss sie über diesen Port mindestens für den Portalserver erreichbar sein. Soll die multifunktionale Portal-API genutzt werden, so muss der Zugriff für alle weiteren relevanten Komponenten freigeschaltet sein.

6.7. Prüfung der Anforderungen

Die folgende Tabelle 6.1 gibt einen Überblick darüber, welche Anforderungen von der prototypischen Implementierung erfüllt werden. Der Grund für die Nichterfüllung einiger Anforderungen, nämlich der Unterstützung von Resourcepools und der Verwaltungsmöglichkeit für Snapshots, liegt in der Beschränkung des Funktionsumfangs aus zeitlichen Gründen.

Ist ein Punkt in der Tabelle mit Klammern versehen, so bedeutet dies, dass die Anforderung nur eingeschränkt erfüllt wurde.

Welche Funktionen bei der Antragsverwaltung und bei der Steuerung der VMs umgesetzt wurden, kann man am besten der Beschreibung des Bedienkonzepts in Kapitel 6.5 entnehmen. Diese listet die diversen Funktionen nacheinander auf.

Nur eingeschränkt umgesetzt wurde die Konsole, da sie nicht über ein Plugin realisiert wurde, so wie es anfangs angedacht war und von den Nutzern auch gewünscht wurde, sondern über eine Standalone-Anwendung. Einerseits bringt dies eine größere Unabhängigkeit vom verwendeten Browser, bedeutet aber andererseits eine Komfortminderung bei der Bedienung.

Auch das Mounten von ISO-Images wurde nur eingeschränkt erfüllt, da die Möglichkeit hierzu nur über die Konsolenanwendung besteht und nicht im eigentlichen Portal implementiert wurde. Über die Konsolenanwendung besteht zudem keine Möglichkeit ISO-Images auf den Datastore hochzuladen. Es können nur dort bereits existierende Images oder Images vom lokalen System des Portalnutzers gemountet werden.

Bei den nicht-funktionalen Anforderungen stellt sich die Lage wie folgt dar:

Anpassbarkeit Die Implementation wurde für das LRZ und die LMU erstellt, der Quellcode steht diesen Institutionen zur freien Verwendung bereit. Einer Erweiterung der Software und einer Anpassung an neue Umstände steht also nichts im Wege. Zudem wurde die Implementierung, auf Wunsch des LRZ, auf Basis des Symfony-Frameworks durchgeführt, was eine weitere Arbeit an dieser zusätzlich vereinfachen sollte.

Nutzung bisheriger Rechteverwaltung Die Implementierung verwendet die Rechteverwaltung von vSphere. Es müssen lediglich in dieser zusätzliche Rollen für das Portal erstellt werden. Im Hintergrund greift vSphere wiederum auf das institutsweite ActiveDirectory zu. Ein Nutzer der über das Webportal eine VM bereitgestellt bekommt, kann diese folglich auch über den vSphere Client steuern - eine entsprechende Firewall-Konfiguration vorausgesetzt.

Schnittstellen/APIs Implementiert wurde nur die Basisvariante des Konzepts, also der direkte Zugriff auf die Datenbank. Eine Prüfung der Datenbank auf Veränderungen durch externe ITSM-Software findet ebenfalls nicht statt.

6. Implementierung

Betrachtet man die konkrete Situation am LRZ, so stellt man fest, dass dort für das IT Service Management iET-ITSM von iET Solutions eingesetzt wird. In dieser Software können auf eine einfache Art und Weise Formulare erzeugt werden. Die Kommunikation mit externen Systemen findet über Java-Klassen statt, die man im System hinterlegen kann. Da Java auch objektrelationales Mapping unterstützt wäre es problemlos möglich, ein Formular in iET-ITSM zu erstellen, über welches Mitarbeiter am LRZ VMs beantragen könnten. Die Bearbeitung der Anträge müsste aber aus den genannten Gründen über das GUI im Webportal erfolgen.

Die Anforderung nach den benötigten Schnittstellen wird also nur teilweise erfüllt.

Skalierbarkeit Die Skalierbarkeit leitet sich zum Teil bereits aus dem Konzept ab. Das eigentliche Webportal, die dazugehörige Datenbank und der Adapter können auf jeweils unterschiedlichen Hosts installiert werden. Das Webportal und der Datenbankserver können bei der Implementierung auch mehrfach parallel betrieben werden. Die Implementierung unterstützt allerdings nicht mehrere Adapter, die per Round-Robin-Verfahren angesprochen werden. Man könnte die Lastverteilung jedoch weiterhin anders realisieren, zum Beispiel DNS-basiert. Bei einer prototypischen Implementierung, die primär Demonstrationszwecken dienen soll, hat eine hohe Skalierbarkeit zudem nur eine geringe Priorität.

Auf der anderen Seite verfügt die Implementierung über zwei weitere Merkmale, die sich positiv auf die Skalierbarkeit auswirken. Zum einen sind dies die in Symfony eingebauten Caching-Mechanismen, die eine hohe Leistung bei der Ausführung der PHP-Skripte gewährleisten. Zum zweiten besteht die Möglichkeit einen PHP Accelerator zu installieren, um die Leistung des Portals nochmals zu erhöhen.

Da das einzige Defizit der Implementierung das fehlende Round-Robin-Verfahren ist, und dieses relativ leicht umzusetzen ist, wird diese Anforderung als erfüllt betrachtet.

Verfügbarkeit Die Angaben zur Skalierbarkeit treffen auch für die Verfügbarkeit zu. Folglich gilt auch diese Anforderung als erfüllt.

Betrachtet man zum Schluß noch die Unterstützung der diversen Virtualisierungslösungen, so stellt man fest, dass nur vSphere unterstützt wird. Einzelne ESXi-Server können wegen dem spezifischen Konfigurationsaufbau des Adapters nicht verwendet werden. Es war allerdings von Anfang an das Ziel der Implementierung nur einen Adapter für vSphere zu schreiben.

Benutzerverwaltung	Passende Benutzergruppen werden unterstützt	✓
Bestell- und Bereitstellungs- vorgang	Bestellmöglichkeit besteht	✓
	Parameter frei wählbar	✓
	Vorlagen / fertige Angebote verfügbar	✓
	Einzelne Parameter bei Vorlagen modifizierbar	✓
	AGBs werden angezeigt	✓
	Preis wird angezeigt	✓
	Administrator kann Antrag ablehnen	✓
	Administrator kann Alternative vorschlagen	✓
	Vor Bereitstellung ist manueller Eingriff möglich	✓
Benachrichtigungen werden verschickt	✓	
Betrieb	Steuerung des Betriebs möglich	✓
	VMs können gestartet/gestoppt/resetet werden	✓
	Snapshots können angelegt/verwaltet werden	
	Mounten von Nutzer-ISO-Images möglich	(✓)
	Status-/Performance-Daten sind abrufbar	✓
	Fertige VMs können modifiziert werden	✓
	Zugriff auf die Konsole ist möglich	(✓)
Löschungen können beantragt werden	✓	
Ressourcepools	Ressourcepools werden unterstützt	
	VMs können geklont werden	
	Verbrauchte Ressourcen werden angezeigt	
Nicht-funktionale Anforderungen	System kann einfach angepasst werden	✓
	Bisherige Rechteverwaltung wird verwendet	✓
	Schnittstellen/APIs sind verfügbar	(✓)
	System ist skalierbar	✓
	System kann hochverfügbar aufgesetzt werden	✓
Virtualisierer- unterstützung	vSphere wird unterstützt	✓
	ESXi-Server werden unterstützt	
	Anderer Virtualisierer werden unterstützt	

Tabelle 6.1.: Anforderungserfüllung der Implementierung

6. Implementierung

7. Fazit

An dieser Stelle soll das Vorgehen zur Konzeptionierung und Implementierung des Webportals für das Management virtueller Maschinen nochmals kurz zusammengefasst und die Erkenntnisse wiederum hervorgehoben werden. Anschließend erfolgt noch ein Ausblick auf in der Zukunft mögliche Weiterentwicklungen des Portals.

7.1. Zusammenfassung

Die Arbeit begann mit der Idee ein VM-Management-Portal für Institutionen zu schaffen, die ihren Kunden oder Mitarbeitern individuelle VMs bereitstellen wollen. Das Webportal sollte dabei den gesamten Dienstlebenszyklus abdecken, also von der Beantragung bis zur Kündigung beziehungsweise Löschung und Abrechnung. Es sollte einerseits die virtuelle Infrastruktur steuern und andererseits eine Schnittstelle zwischen Administrator und Endnutzer bilden. Das Portal sollte die Arbeit der Administratoren und Endnutzer vereinfachen, indem es den gesamten Arbeitsfluss unter einer gemeinsamen Oberfläche vereint. Dabei sollte das Webportal weder auf eine konkrete Virtualisierungslösung festgelegt sein, noch sich an der IT-Infrastruktur einer einzelnen Institution orientieren. Es sollte generisch und an diverse Bedürfnisse leicht anpassbar sein. Neben der Vereinfachung des Managements sollte das Webportal auch diverse Sicherheitslücken, die beim Einsatz der bisherigen Managementlösungen auftraten, schließen. So sollte unter anderem der direkte Zugriff auf die zentrale Steuerkomponente der Virtualisierungslösung durch die Endnutzer verhindert werden. Die Lösung sollte sich fließend in das bisherige Management integrieren lassen und deshalb auf die vorhandene Rechteverwaltung aufsetzen.

Im Rahmen der Realisierung eines solchen Webportals wurde zunächst im Kapitel 3 eine umfangreiche Anforderungsanalyse durchgeführt. Im ersten szenariobasierten Teil, im Unterkapitel 3.1, konnte bereits eine große Anzahl an Anwendungsfällen identifiziert und daraus viele Anforderungen abgeleitet werden. Um die konkreten Wünsche und Bedürfnisse der zukünftigen Nutzer nicht aus den Augen zu verlieren, wurde im zweiten Teil, im Unterkapitel 3.2, eine Befragung durchgeführt. Um beide Seiten kennenzulernen und eventuelle Widersprüche zu identifizieren, wurden dabei sowohl Administratoren als auch Endnutzer befragt. Des weiteren erfolgte die Befragung an zwei verschiedenen Institutionen, um die Ergebnisse nicht allzu stark von der Situation an einem konkreten Standort abhängig zu machen. Die Befragung half die Anwendungsfälle des ersten Teils zu konkretisieren und dabei sogar Falschannahmen aufzudecken. Aus den gewonnenen Daten konnten weitere Anforderungen mit unterschiedlichen Prioritäten gewonnen werden. Zum Vorschein kamen dabei nicht nur konzeptrelevante Informationen, sondern auch detaillierte Wünsche bezüglich der Implementierung. Die Ergebnisse der beiden Teile wurden kombiniert und im Unterkapitel 3.5 ein Anforderungskatalog erstellt.

Nach Abschluss der Anforderungsanalyse wurde im Kapitel 4 nach bereits existierenden Lösungen gesucht, die den Anforderungen entsprechen. Entstanden ist dabei eine Übersicht vieler verschiedener Managementlösungen in diversen Kategorien. Doch bei der genauen

7. Fazit

Untersuchung der vielversprechendsten Anwendungen hat sich gezeigt, dass keine von ihnen die Anforderungen an eine Lösung im Sinne dieser Arbeit zu erfüllen vermag.

Durch die Erkenntnis bestärkt, etwas noch nicht dagewesenes zu entwickeln, wurde im Kapitel 5 ein Konzept für das Webportal aufgestellt. Dabei wurde das Webportal in eine Gesamtarchitektur integriert, die neben den zwingend erforderlichen Komponenten, diverse optionale Komponenten berücksichtigt, die an einer Institution vorhanden sein können aber nicht müssen. Im Unterkapitel 5.3.1 wurde ein Informationsmodell aufgestellt, das sowohl die Bereitstellung von VMs als auch von Ressourcepools unterstützt. Zudem ermöglicht es die Extraktion von Nutzungsdaten für das Accounting ohne diese dediziert zu speichern. Um das Webportal selbst generisch zu gestalten, wurde für die Kommunikation mit dem Virtualisierer ein Adapter vorgesehen, der die Befehle entsprechend übersetzt. Für diesen wurde sogleich im Unterkapitel 5.5 eine generische REST-API entworfen.

Um die Umsetzbarkeit des Konzepts zu demonstrieren, wurde dieses prototypisch implementiert. Diese Implementierung, welche im Kapitel 6 beschrieben wurde, wurde primär für das LRZ angefertigt und deshalb auf Wunsch der dortigen Administratoren mit Hilfe des Symfony-Frameworks umgesetzt. Zusammen mit dem vSphere-Adapter lässt es sich aber an beliebigen anderen Institutionen mit einer vSphere-Infrastruktur einsetzen. Es wurde aufgezeigt wie man bei einem Einsatz des Webportals das von Angriffen gefährdete vCenter mit einer Firewall-Konfiguration abschirmen kann.

Der Abgleich des Konzepts mit den Anforderungen (siehe Unterkapitel 5.6) zeigt, dass das Konzept den kompletten Katalog, wenn auch mit einer kleinen Einschränkung, erfüllt. Die Implementierung erfüllt ebenfalls einen Großteil des Katalogs (siehe Unterkapitel 6.7). Die nicht erfüllten Punkte liegen vor allem am eingeschränkten zeitlichen Rahmen, sowie, im Falle der Konsole, vor allem an der mangelnden Dokumentation derselben.

Damit wurde mit dem Webportal eine Software entwickelt, die Nutzern, die individuell eingerichtete virtuelle Maschinen benötigen und Institutionen, die solche anbieten, das Management über den gesamten Dienstlebenszyklus ermöglicht und damit die Arbeitsabläufe vereinfacht und fehlerfreier gestaltet.

7.2. Ausblick

Nach der Zusammenfassung der Ergebnisse dieser Arbeit soll an dieser Stelle auf mögliche Weiterentwicklungen eingegangen werden.

Zunächst seien einige kleinere Weiterentwicklungsmöglichkeiten der Implementierung erwähnt. Neben einer Implementierung der nicht eingebauten Funktionalität, wäre es vor allem bei einer Verwendung von vSphere natürlich weiterhin wünschenswert, wenn man die Konsole als Plugin umsetzen könnte, zumal dies laut Nutzerbefragung eindeutig bevorzugt wird. Des weiteren könnte man die Benutzerfreundlichkeit für die Administratoren erhöhen. So könnte man die diversen Konfigurationsparameter, die in einer YAML-Datei bearbeitet werden müssen, auch über das GUI zugänglich machen. Außerdem könnte man ein Vorlagen-Management einführen. Die Vorlagen könnte man dann ebenfalls in der Datenbank speichern und die vom Ajax-Request angeforderte XML-Datei dynamisch erzeugen.

Eine weitere Idee für mögliche Verbesserungen betrifft das Rechtemanagement. Das bisherige Konzept sieht keine Möglichkeit vor, dass ein Nutzer Rechte für seine VMs weitervergeben kann. Gerade bei Ressourcepools wäre dies aber sinnvoll. Bei der Verwendung von Virtualisierungslösungen, die eine integrierte Rechteverwaltung wie vSphere haben, ließe sich

dies ohne eine weitere Anpassung des Informationsmodells realisieren. Dies würde wie folgt funktionieren: Zunächst werden Gastrechte definiert, dies sind zum Beispiel das Einsehen des Statuses einer VM sowie das Starten und Stoppen. Entsprechend dieser Rechte wird im vCenter eine neue Rolle angelegt. Das Webportal wird um eine Funktion erweitert einen Gastzugriff auf eine VM einzurichten. Hierzu muss der Besitzer der VM lediglich die ID des zukünftigen Gastes eingeben. Das Portal weißt dann über die bereits im Adapter existierende Methode "privileges" die entsprechende Gastrolle auf der VM zu. Damit die VM unterhalb des Kontos des Gastes sichtbar wird, fragt das Portal über den Adapter ab, für welche VMs der Nutzer über die Gastrolle verfügt, verknüpft diese Information mit den in der Portal-Datenbank vorhandenen Informationen zu dieser VM und zeigt sie entsprechend an. Probleme ergeben sich, wenn die Virtualisierungslösung über keine entsprechende Rechteverwaltung verfügt. Man müsste wohl in so einem Fall das Informationsmodell entsprechend erweitern, um auch die Gastberechtigungen zu speichern.

Verbesserungswürdig ist auch die Auswertung der Statistik- und Performance-Daten. Möchte man dies aber generisch lösen, so bedeutet das einen größeren Aufwand. Man könnte versuchen ein virtualisiererunabhängiges Schema für Statistiken zu entwerfen. Es ist anzunehmen, dass dies sehr schwierig umzusetzen sein wird, wenn man sich nicht auf nur wenige Basiswerte beschränken möchte. Bereits bei der in dieser Arbeit beschriebenen Adapter-API, die für die Statistik nur wenige Werte überträgt, könnten sich einzelne Parameter nicht problemlos auf andere Virtualisierer übertragen lassen. Eine mögliche Alternative wäre, wie schon bereits in der API-Beschreibung angedeutet, die Statistikauswertung in den Adapter zu verlagern. Dieser würde dann, ähnlich der Konsole, ein HTML-Fragment an das Portal schicken. Der Adapter könnte auch Grafiken erzeugen und diese zusammen mit dem Fragment an das Portal schicken, damit dieses diese Informationen wiederum dem Nutzer verfügbar macht.

Und wo bereits von unterschiedlichen Virtualisierern die Rede ist, so wäre es ebenfalls eine Überlegung wert unterschiedliche Virtualisierungslösungen gleichzeitig zu betreiben und über das Portal gleichzeitig zu steuern. Hierfür müsste sicherlich auch das Informationsmodell erweitert werden. Man müsste zum Beispiel speichern, unter welcher Virtualisierungslösung eine VM läuft, damit die Steuerkommandos an den richtigen Adapter geschickt werden können. Man müsste sich auch überlegen beziehungsweise durch eine Befragung herausfinden, ob die Endnutzer eine VM unter einer bestimmten Virtualisierungslösung beantragen können sollen oder ob die Entscheidung über welchen Virtualisierer eine VM provisioniert wird bei den Administratoren liegen soll. Für die Entwicklung von Adaptern für weitere Virtualisierer sollte übrigens geprüft werden, ob sich hierfür libvirt nicht als Grundlage verwenden ließe.

Ideen für mögliche Weiterentwicklungen gibt es jedenfalls genug und diese Arbeit wird hierfür hoffentlich eine gute Grundlage sein.

7. Fazit

A. Anhang

Fragebogen der Anforderungsanalyse

Fragen an Administratoren allgemein

1. Auf welche Parameter einer VM kann der Nutzer Einfluss nehmen?
2. Unterscheiden sich die Parameter welche der Nutzer nachträglich im Betrieb ändern kann, von denen bei der Beantragung einer VM?
 Ja Nein
Wenn ja, welche?
3. Gibt es vordefinierte Standardkonfigurationen aus denen der Nutzer wählen kann?
 Ja Nein
Wenn nein, wäre es sinnvoll solche definieren zu können?
 Ja Nein
Wenn ja, könnten solche Standardkonfiguration automatisch ohne Eingriff eines Administrators eingerichtet werden?
 Ja Nein
4. Über welche Parameter wird ein Ressourcenpool definiert?
5. Welche Formen von virtueller Netzinfrastruktur können beantragt werden? Sind beliebige Konfigurationen möglich? Kann den Nutzern eine Auswahl an möglichen Konfigurationen angeboten werden?

A. Anhang

6. Fällt die Entscheidung, ob es gewünschte Infrastruktur genehmigt wird, individuell oder erfolgt dies an Hand festgelegter Grenzwerte? Gibt es z.B. Maximalwerte für möglichen RAM oder Speicherplatz?

individuell an Hand von Grenzwerten, und zwar den Folgenden:

7. Welche Statusmeldungen bzgl. der VMs (Zustand, Leistung) sollen im Webportal sichtbar sein, welche davon soll auch der Endnutzer sehen?

8. Wie werden IP-Adressen zugeteilt?

9. Werden Bulk-Aktionen benötigt?

Nein Ja, für folgende Aktionen:

10. Wird die Möglichkeit VMs zu klonen benötigt?

Ja Nein

11. Muss die Möglichkeit Snapshots anzulegen beantragt werden oder steht diese Funktion immer zur Verfügung? Wer kann diese nutzen?

12. Welche Einflussmöglichkeiten auf das Backup der VMs hat der Endnutzer?

13. Auf welchem Wege sollen die Nutzer darüber informiert werden, dass Ihre Infrastruktur fertig eingerichtet ist?

14. Kommt es regelmäßig zu Nachfragen bzgl. der beantragten Infrastruktur beim Endnutzer? Weshalb?

Fragen an Administratoren am LRZ

15. Gibt es eine Formel mit deren Hilfe der Preis an Hand der gewählten Parameter berechnet werden kann? Gibt es preisliche Unterschiede für verschiedene Nutzergruppen?

Formel	<input type="checkbox"/> Ja, siehe unten	<input type="checkbox"/> Nein
Unterschiede	<input type="checkbox"/> Ja, siehe unten	<input type="checkbox"/> Nein

16. Genügt es wenn der Endnutzer die AGBs einmalig bei der ersten Nutzung des Webportals akzeptiert?

Ja Nein

Die folgenden Fragen beziehen sich auf unattended hosting.

17. Auf welchem Wege kann der Nutzer sein gewünschtes Betriebssystem installieren? Werden dem Endnutzer auf dem Datastore ISO-Images bereitgestellt, die er mounten kann? Kann der Endnutzer seine eigenen ISO-Images irgendwo hochladen, damit sie gemountet werden? Soll der Endnutzer die Möglichkeit haben seine lokal verfügbaren ISO-Images direkt zu mounten? Wird Netbooting angeboten?

ISOs auf Datastore	<input type="checkbox"/> Ja	<input type="checkbox"/> Nein
ISOs hochladen	<input type="checkbox"/> Ja	<input type="checkbox"/> Nein
lokale ISOs mounten	<input type="checkbox"/> Ja	<input type="checkbox"/> Nein
Netbooting	<input type="checkbox"/> Ja	<input type="checkbox"/> Nein

A. Anhang

18. Soll den Nutzern Zugriff auf das BIOS der VMs gewährt werden?

- Ja Nein

Die folgenden Fragen beziehen sich auf attended hosting.

19. Kann der Endnutzer nur aus einer fest definierten Liste von Betriebssystem wählen oder kann er beliebige Wünsche äußern?

- Liste Wünsche

20. Kann der Endnutzer beantragen, dass keine weitere Pflege seiner VMs stattfindet?

- Ja Nein

21. Werden die Betriebssysteme in verschiedenen Versionen angeboten (z.B. mit unterschiedlicher vorinstallierter Software)?

- Ja Nein

Fragen an Endnutzer allgemein

22. Wenn Sie VMs mit beliebigen Parametern beantragen können, würden Sie trotzdem gerne auf vordefinierte Standardkonfigurationen zurückgreifen?

- Ja Nein

23. Werden Bulk-Aktionen benötigt?

- Nein Ja, für folgende Aktionen:

24. Wird die Möglichkeit Snapshots anzulegen benötigt?

- Ja Nein

25. Wird ein Einfluss auf das automatische Backup der VMs, z.B. aus rechtlichen Gründen, benötigt?

Ja Nein

26. Wird eine mobile Version des Portals für die Steuerung von unterwegs benötigt?

Nein Ja, für folgende Aktionen:

Fragen an unattended hosting Endnutzer des LRZ bzw. Endnutzer an der LMU

27. Wird ein Zugriff auf das BIOS der VM benötigt?

Ja Nein

28. Genügt für die Installation des Systems eine Auswahl an ISO-Images des Anbieters oder werden eigene Images benötigt?

Auswahl Eigene

29. Wird für den Konsolenzugriff ein Browserplugin oder eine Standalone-Anwendung bevorzugt?

Browser Standalone

Fragen an attended hosting Endnutzer des LRZ

30. Würden Sie beim attended hosting gerne beantragen, dass keine weitere Pflege der VMs stattfinden soll?

Ja Nein

31. Wird ein Konsolenzugriff benötigt oder genügen die Zugriffsmöglichkeiten der Betriebssysteme?

Ja Nein

Fragen an technische Administratoren / Nutzer eines Ressourcenpools

32. Wird bei der Nutzung eines Ressourcenpools die Möglichkeit benötigt VMs zu klonen?

Ja Nein

33. Werden VMs aus einem Ressourcenpool von den Endnutzern formal beantragt oder hat man als Nutzer eines Ressourcenpools bereits Kenntnis darüber welche VMs man anlegen will? Wird eine Interaktionsmöglichkeit zwischen technischem Administrator und Endnutzer über das Webportal benötigt?

Abbildungsverzeichnis

2.1.	Hosted- und Bare-Metal-Hypervisor	7
2.2.	Ein großes Defizit mitgelieferter Management-Anwendungen: keine integrierte Interaktionsmöglichkeit zwischen Administratoren und Endnutzern	9
2.3.	Mit dem vSphere Client kann man die Infrastruktur sowohl über das vCenter steuern (dicke Pfeile), als auch auf die einzelnen ESXi-Server direkt zugreifen (dünne Pfeile)	10
3.1.	Ablaufdiagramm des Anwendungsfalls AntragVMNeu	16
3.2.	Ablaufdiagramm des Anwendungsfalls AntragVMMod	17
3.3.	Ablaufdiagramm des Anwendungsfalls AntragVMLöschung	18
3.4.	Ablaufdiagramm der Anwendungsfälle SteuereVM, SteuereVM_Bulk, MountISO, SnapshotVM und KloneVM	18
3.5.	Ablaufdiagramm des Anwendungsfalls StarteKonsole	20
3.6.	Ablaufdiagramm des Anwendungsfalls ÄndereBesitzer	21
3.7.	Ablaufdiagramm der Anwendungsfälle ErstelleRPVM, ModifiziereRPVM und LöscheRPVM	24
3.8.	Use-Case-Diagramm	25
4.1.	Steuerung einer Virtualisierungslösung mit Hilfe von libvirt	36
5.1.	Einordnung des Webportals in das bisherige Management	44
5.2.	Komponentendiagramm mit Richtungen des Informationsaustauschs	48
5.3.	Komponentendiagramm mit bereitgestellten APIs	49
5.4.	Webportal-Komponente im Detail	52
5.5.	Informationsmodell	54
5.6.	Zustände von Anträgen, VMs und Resourcepools und ihre Zuordnung zu den Phasen des Dienstlebenszykluses	57
6.1.	Datenbankschema	84
6.2.	Übersicht über die bestehendes VMs mit diversen Steuermöglichkeiten und Auftragsübersicht	91
6.3.	Überblick eines Administrators über die eingereichten Anträge, aufgeteilt in die verschiedenen Antragsarten	92
6.4.	Antragsformular für neue VMs	93
6.5.	Auftragsbearbeitung mit Auswahlmöglichkeiten	93
6.6.	Anzeige eines durch einen Administrator erstellten Alternativvorschlags mit Möglichkeit zur weiteren Modifikation	93

Literaturverzeichnis

- [ABI] *Abiquo Overview*. abgerufen am 3.6.2011
<http://www.abiquo.com/products/abiquo-overview.php>.
- [AQE] *AQEMU*. abgerufen am 1.6.2011
<http://sourceforge.net/projects/aqemu/>.
- [Bit09] BITTNER, FLORIAN: *Eine erweiterbare Management-Plattform für Hostvirtualisierungslösungen*. Diplomarbeit, Ludwig-Maximilians-Universität München, 2009. <http://www.mnm-team.org/pub/Diplomarbeiten/bitt08/PDF-Version/bitt08.pdf>.
- [Bli10] BLIGH, ALEX: *Adding SQL support to ISC dhcpd*, September 2010. abgerufen am 5.9.2011
<http://blog.alex.org.uk/2010/09/05/adding-sql-support-to-isc-dhcpd/>.
- [CLO] *Webmin Cloudmin*. abgerufen am 2.6.2011
<http://www.webmin.com/cloudmin.html>.
- [CON] *ConVirt Open Source*. abgerufen am 2.6.2011
http://www.convirture.com/products_opensource.php.
- [FHBH⁺99] FRANKS, J., P. HALLAM-BAKER, J. HOSTETLER, S. LAWRENCE, P. LEACH, A. LUOTONEN und L. STEWART: *HTTP Authentication: Basic and Digest Access Authentication*. RFC 2617 (Draft Standard), Juni 1999. <http://www.ietf.org/rfc/rfc2617.txt>.
- [GAN] *ganeti*. abgerufen am 1.6.2011
<http://code.google.com/p/ganeti/>.
- [Geh08] GEHRUNG, MATHIAS: *Managementkonzept für virtuelle Infrastrukturen unter besonderer Berücksichtigung unterschiedlicher Nutzeranforderungen*. Diplomarbeit, Ludwig-Maximilians-Universität München, 2008. <http://www.nm.ifi.lmu.de/pub/Diplomarbeiten/gehr09/PDF-Version/gehr09.pdf>.
- [GKV] *GKVM: Gnome KVM Manager*. abgerufen am 1.6.2011
<http://gkvm.sourceforge.net/>.
- [KAR] *Karesansui Project*. abgerufen am 2.6.2011
<http://karesansui-project.info/>.
- [Lin09] LINDINGER, TOBIAS: *Optimierung des Wirkungsgrades virtueller Infrastrukturen*. Doktorarbeit, Ludwig-Maximilians-Universität München, 2009. http://edoc.ub.uni-muenchen.de/11120/1/Lindinger_Tobias.pdf.

- [LVV] *libvirt - VMware ESX hypervisor driver*. abgerufen am 27.5.2011
<http://libvirt.org/drvesx.html>.
- [Mun] MUNRO, KIRK: *VMware vSphere Management*. abgerufen am 6.6.2011
<http://www.powergui.org/entry.jspa?externalID=1802&categoryID=290>.
- [NM95] NEBEL, E. und L. MASINTER: *Form-based File Upload in HTML*. RFC 1867, November 1995. <http://www.ietf.org/rfc/rfc1867.txt>.
- [ONO] *OpenNode Cloud Platform*. abgerufen am 2.6.2011
<http://opennode.activesys.org/>.
- [OQR] *openQRM Enterprise*. abgerufen am 3.6.2011
<http://www.openqrm-enterprise.com/>.
- [OVF10] DISTRIBUTED MANAGEMENT TASK FORCE, INC.: *Open Virtualization Format Specification*, 1.1.0 Auflage, Januar 2010. http://dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.0.pdf.
- [PDD] *Dancer::Deployment - Running from Apache with Plack*. abgerufen am 10.9.2011
<https://metacpan.org/module/Dancer::Deployment#Running-from-Apache-with-Plack>.
- [PRO] *Proxmox Virtual Environment*. abgerufen am 3.6.2011
<http://pve.proxmox.com/>.
- [PVC] *Parallels Virtuozzo Containers*. abgerufen am 3.6.2011
<http://www.parallels.com/de/products/pvc/>.
- [sdk10] VMWARE, INC.: *vSphere SDK for Perl Installation Guide (Release 4.1)*, 2010. http://www.vmware.com/support/developer/viperltoolkit/viperl41/doc/vsp_4_41_vsperl_install.pdf.
- [SYS] *SYSVM Vps Control Panel*. abgerufen am 1.6.2011
<http://sysvm.com/>.
- [UVM] *Univention Corporate Server - Univention Virtual Machine Manager (UVMM)*. abgerufen am 2.6.2011
<http://www.univention.de/en/products/ucs/ucs-components/virtualisierung/univention-virtual-machine-manager/>.
- [VBR] *Virtualbricks*. abgerufen am 1.6.2011
<http://virtualbricks.eu/>.
- [VMM] *Virtual Machine Manager*. abgerufen am 1.6.2011
<http://www.ietf.org/rfc/rfc2617.txt>.
- [WSW10] *VMware vCenter Server 4.0 Update 2 Release Notes*, September 2010. abgerufen am 15.5.2011
http://www.vmware.com/support/vsphere4/doc/vsp_vc40_u2_rel_notes.html.