



Fachbereich Wirtschaft

Bachelorarbeit zur Erlangung des Grades
Bachelor of Science
im Studiengang Wirtschaftsinformatik

Vom klassischen PC zum zentralisierten Desktop:
Technische Implementierung einer Thin-Client
Infrastructure und vergleichende Analyse mit einer
bestehenden Virtual Desktop Infrastructure
am Universitätsrechenzentrum Greifswald

Vorgelegt von: Christian Schuldt
Matrikel-Nr.: 12857
Hans-Beimler-Str. 8a
17491 Greifswald

Erstgutachter: Prof. Dr. rer. nat. Gerold Blakowski
Zweitgutachter: Dr. Gordon Grubert

Abgabetermin: 14.01.2016

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbst angefertigt habe und alle von mir benutzten Hilfsmittel und Quellen angegeben wurden. Alle wörtlichen Zitate und Entlehnungen aus fremden Arbeiten sind als solche gekennzeichnet.

Ort, Datum

Unterschrift: Christian Schuldt

Greifswald, den 14.01.2015

Inhaltsverzeichnis

1	Einleitung	2
1.1	Vorgehensweise	3
1.2	Kurzzusammenfassung	3
2	Virtualisierung Grundlagen - Theorieteil	4
2.1	Virtualisierung	4
2.2	Desktopvirtualisierung	5
2.3	Virtual Desktop Infrastructure	6
2.3.1	Hypervisor	6
2.3.2	Connection Broker	7
2.3.3	Übertragungsprotokoll	7
2.4	Thin-Client Infrastructure	8
2.4.1	iSCSI	8
2.4.2	Unterscheidung Thin-Client und Zero-Client	9
3	Thin Client Infrastructure	10
3.1	Aufbau und Netzwerkstruktur	10
3.2	Implementierung	12
3.2.1	Boot Vorgang	12
3.2.2	Übertragungssicherheit	14
3.2.3	Authentifizierungsmethode	17
3.2.4	Cluster	18
3.3	iSCSI - Server	22
3.3.1	Daemon	22
3.3.2	Installation	22
3.4	Konfiguration	24
3.4.1	Netzwerk	24
3.4.2	TFPT Server	25
3.4.3	Festplatten	26
3.4.4	iSCSI Enterprise Target	29
3.4.5	DHCP	31
3.4.6	Backup	31
3.5	Performance	32
3.5.1	Kernel IO-Scheduler	32

3.5.2	Mount Optionen	33
3.6	Thin-Client	34
3.7	Automatisierung.....	35
3.8	Fazit und Ausblick.....	37
4	Virtual Desktop Infrastructure.....	39
4.1	VDI – technische IST-Analyse des URZ Greifswald.....	39
4.2	Verwendete Komponenten IST-Zustand.....	40
4.3	Aufbau und Netzwerkstruktur IST-Zustand VDI.....	41
4.4	Schwachstellenanalyse.....	42
4.5	Aufbau und Netzwerkstruktur SOLL – Zustand.....	44
4.6	Implementierung VMware Sicherheitsserver	45
4.7	Zusammenfassung VDI.....	47
5	Kostenanalyse	48
5.1	Kostengegenüberstellung	48
5.1.1	Investitionskosten.....	49
5.1.2	Wartungs- und Personalaufwand	52
5.2	Anforderungsprofile	53
5.2.1	Zero-User	53
5.2.2	Thin-User.....	54
6	Fazit.....	55
7	Literatur- und Quellenverzeichnis	57
8	Anhang	i
8.1	Installationsleitfaden iSCSI-Server	i
8.2	Automatisierungsscript	ii
8.3	iSCSI-Start Script.....	iv
8.4	Fragebogen Wartungs- und Personalaufwand	v

Abbildungsverzeichnis

Abbildung 1 Virtualisierung Hypervisor Typ 1 (Eigene Darstellung)	4
Abbildung 2 Gegenüberstellung vollständige und Paravirtualisierung (Eigene Darstellung).....	6
Abbildung 3 VDI Connection Broker (Eigene Darstellung)	7
Abbildung 4 Server und Netzwerkstruktur iSCSI (Eigene Darstellung).....	11
Abbildung 5 Bootvorgang iSCSI (Eigene Darstellung)	12
Abbildung 6 iSCSI Übertragungsgeschwindigkeit Quelle: Weiping, Wandong (2005, S. 457-462).....	15
Abbildung 7 iSCSI CPU Auslastung Quelle: Weiping, Wandong (2005, S. 457-462) ..	15
Abbildung 8 Serverauslastung IOSTAT (Screenshot iSCSI1).....	20
Abbildung 9 iSCSI1 Netzwerkkonfiguration (Eigene Darstellung)	24
Abbildung 10 iSCSI Network Interface (Datei).....	25
Abbildung 11 TFTP Konfiguration (Datei)	25
Abbildung 12 iSCSI SAN-Anbindung (Eigene Darstellung)	27
Abbildung 13 iSCSI IET Konfiguration (Datei)	29
Abbildung 14 DHCP Optionen (Eigene Darstellung)	31
Abbildung 15 Thin-Client Dell OptiPlex 7020 (Quelle: http://www.dell.com/)	34
Abbildung 16 BPMN Automatisierung iSCSI-Target (Eigene Darstellung).....	36
Abbildung 17 IST-Zustand Netzwerkinfrastruktur VDI (Eigene Darstellung).....	41
Abbildung 18 SOLL-Zustand VDI (Eigene Darstellung).....	44
Abbildung 19 Option Secure Gateway (Screenshot)	46

Tabellenverzeichnis

Tabelle 1 Installierte Dienste iSCSI-Target.....	23
Tabelle 2 VMware Komponenten IST-Zustand	40
Tabelle 3 Kostengegenüberstellung VDI / Thin-Client Infrastructure	50
Tabelle 4 Kosten pro Client	51

Abkürzungsverzeichnis

DHCP	Dynamic Host Configuration Protocol
iSCSI	internet Small Computer Interface
PXE	Preboot Execution Enviroment
TFTP	Trivial File Transfer Protocol
Daemon	Disk and execution monitors
RFC	Request For Comments
IPSec	Internet Protokoll Security
CPU	Central Processing Unit
SSH	Secure Shell
LUN	Logical Unit Number
VLAN	Virtual Local Area Network
BIOS	Basis Input Output System
MAC	Media Access Control
DRBD	Distributed Replicated Block Device
IPSec	Internet Protokoll Security
VDI	Virtual Desktop Infrastructure
SAN	Storage Area Network
DMZ	Demilitarized Zone
FQDN	Fully Qualified Domain Name
VDA	Microsoft Virtual Desktop Access

Vorwort

Ich sitze gerade vor meinem heimischen Schreibtisch an einem Linux Computer und schreibe dieses Vorwort in Microsoft Office 2010 Standard, welches mir das Rechenzentrum der Ernst-Moritz-Arndt Universität Greifswald zur Verfügung gestellt hat. In der heutigen Zeit gibt es mit Sicherheit unterstützende Software, die es mir ermöglicht ein für Windows entwickeltes Microsoft Office auf einem Linux Betriebssystem zu installieren, aber für mich ist das gar nicht wichtig. Es ist für mich aus dem Grund nicht wichtig, weil ich von zu Hause auf einem entfernten Desktop arbeite, welcher mir auf den Servern im Rechenzentrum der Ernst-Moritz-Arndt Universität Greifswald zur Verfügung gestellt wurde. Mein virtueller Desktop mit allen zugehörigen Programmen und Ressourcen, die ich auch am Arbeitsplatz im Büro vor finde. Ich bin dadurch sehr flexibel bezüglich meiner Arbeitszeiten und meines Arbeitsortes.

Für mich war es ein lehrreiches, sehr praxisorientiertes und spannendes Projekt, wofür ich mich gerne beim gesamten Team des Rechenzentrums der Ernst-Moritz-Arndt Universität Greifswald bedanken möchte, ausdrücklich beim technischen Leiter des Rechenzentrums Dr. Gordon Grubert, der es ermöglicht hat, dieses Thema zu bearbeiten.

1 Einleitung

Durch die ständig steigenden Anforderungen an die heutigen IT-Systeme und der damit verbundenen Herausforderungen, diesen gerecht zu werden, hat sich das Konzept der Virtualisierung in den letzten Jahren in den Rechenzentren fest etabliert. Die Herausforderung für das IT-Management ist es bei steigender Komplexität der IT-Systeme die Hardwareressourcen, Kosten und den Energieverbrauch dabei nicht weiter steigen zu lassen. Neben den genannten Herausforderungen gelten für viele Unternehmen ebenso die Motivationsgründe, das administrative Personal durch die Zentralisierung der IT-Infrastruktur zu entlasten.

Das Einsparpotenzial durch die Konsolidierung und Virtualisierung von Serversystemen wurde auch beim Konzept der Desktopvirtualisierung erkannt und angewendet. Im Vergleich zu einer herkömmlichen Client / Server Architektur liegt der Vorteil einer Desktopvirtualisierung beim zentralen Management aller vollwertigen Maschinen. Der Umkehrschluss bei der Virtualisierung von Desktops liegt bei den technischen Leistungsmerkmalen, die durch die Anforderungen anspruchsvoller Anwendungen nicht immer ausreichend sind.

Es besteht die Herausforderung ein Konzept zu entwickeln, welches die Anforderungsspezifikationen von anspruchsvollen Nutzern erfüllt und dabei ähnlich wie die Desktopvirtualisierung auf ein zentrales Management zurückgreifen kann.

Schließt das Konzept der *Thin-Client Infrastructure* die Lücke zum Konzept der Desktopvirtualisierung und ermöglicht es so einem ausgewählten Kreis von Benutzern die Verarbeitung von Anwendungen mit leistungsstärkeren Anforderungen? In den folgenden Kapiteln wird genau auf dieses Ziel hingearbeitet. Flexibilität am Arbeitsplatz spielt in der heutigen Informationsgesellschaft eine zunehmende Rolle. Wie könnte mit der vorhandenen Infrastruktur die Desktopvirtualisierung zukünftig für Mitarbeiter attraktiver gestaltet werden? Die theoretische Beantwortung und die praktische Umsetzung der gestellten Fragen erfolgt im weiteren Verlauf dieser Bachelorthesis.

1.1 Vorgehensweise

Die vorliegenden Bachelorthesis beschäftigt sich mit der Thematik: *Der Wandel vom klassischen PC zum zentralisierten Desktop*. Aus dem Portfolio aller zur Verfügung stehenden Konzepte werden im Folgenden zwei davon technisch und wirtschaftlich gegenüber gestellt. Innerhalb der nächsten Kapitel geht es nicht darum, das beste Konzept der Desktopvirtualisierung am Markt heraus zu kristallisieren und es hier zu präsentieren, sondern vielmehr liegt der Fokus auf einer Analyse und der Erweiterung der bereits vorhandenen Desktopvirtualisierung im Rechenzentrum der Ernst-Moritz-Arndt Universität Greifswald.

Parallel dazu erfolgt die praktische Implementierung einer neuen *Thin-Client Infrastructure*. Die neue Thin-Client Infrastructure soll den Bedarf von Mitarbeitern mit anspruchsvollen Anwendungen decken, welches nicht durch die vorhandene *Virtual Desktop Infrastructure* gewährleistet werden kann.

Im Abschluss erfolgt eine Kostengegenüberstellung der beiden behandelten Konzepte und die anschließende Beschreibung von Anforderungsprofilen, welche für den Einsatz innerhalb der Konzepte geeignet sind.

1.2 Kurzzusammenfassung

Projektauslöser der Bachelorarbeit war die aufgezeigte Einschränkung der vorhandene Desktopvirtualisierung bei der Verarbeitung von leistungsgesteigerten Anwendungen. Es wurde ein Konzept implementiert, welches genau diese anspruchsvollen Anwendungen ermöglicht, aber dabei trotzdem zentral gemanagt werden kann. Zum Einsatz kam eine Thin-Client Infrastructure auf der Basis des iSCSI Protokolls. Die Thin-Client Infrastructure ist keine Virtualisierung durch einen Hypervisor, ermöglicht aber die zentrale Verwaltung der im SAN (Storage Area Network) gespeicherten Festplatten, Bootloader und Konfigurationsdateien.

Die vorhandene Desktopvirtualisierung wurde anhand einer IST-Analyse bewertet und anschließend durch ein Security-Gateway erweitert, welches den weltweiten Zugriff auf die virtuellen Desktops ermöglicht.

2 Virtualisierung Grundlagen - Theorieteil

2.1 Virtualisierung

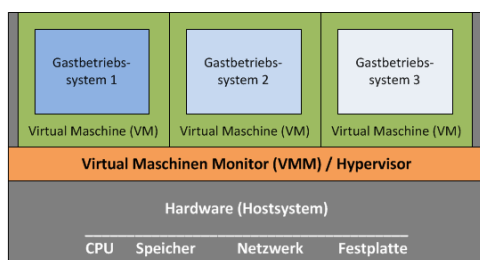
Das folgende Zitat könnte eine allgemein gültige Definition darstellen, welches versucht, die Begrifflichkeit „Virtualisierung“ zu erläutern.

„Virtualisierung bezeichnet Methoden, die es erlauben, Ressourcen (wie Server, Applikationen, Desktop, Storage etc.) mit Hilfe von Software zu abstrahieren und damit die Möglichkeit zum zentralen Zusammenfassen oder Aufteilen zu erhalten“.

[Vogel, Kocoglu, Berger(2010, S.7)]

Aus der Definition geht hervor, dass Ressourcen durch den Einsatz von Software entweder aufgeteilt oder zusammengefasst werden können. Bei der Virtualisierung von Storage-Systemen spricht man z.B. vom Zusammenfassen von Ressourcen, wobei man innerhalb der Netzwerktechnik beispielsweise beim VLAN (*Virtual Local Area Network*) von Aufteilen spricht.

Bei der Servervirtualisierung erfolgt ebenfalls eine Aufteilung von Hardwareressourcen, indem auf einem großen, physischen Server mehrere kleinere, virtuelle Maschinen installiert werden. Die Aufteilung der Hardware durch mehrere virtuelle Maschinen ist sehr effektiv, weil das physische Server System dadurch effizienter ausgenutzt wird. *[Vogel, Kocoglu, Berger(2010, S.7)]*. Um das Konzept der Virtualisierung zu verstehen, sind die Begriffe *Hostsystem*, *Virtuelle Maschine*, *Gastbetriebssystem* und der *Virtual Maschinen Monitor* vorab zu erläutern. Eine virtuelle Maschine erscheint dabei wie ein vollwertiger Rechner, der aber in einer isolierten Umgebung auf einem physischen System (*Hostsystem*) läuft, welche sich mit anderen virtuellen Maschinen dieselben Hardwarekomponenten teilt. Das Gastbetriebssystem wird innerhalb einer virtuellen Maschine ausgeführt,



es arbeitet mit der Hardware des Hostsystem als hätte es die volle Kontrolle darüber, aber in Wirklichkeit verwaltet der VMM (Virtual Maschinen Monitor), auch bekannt als Hypervisor *Kapitel 2.3.1, die Hardware*.

Abbildung 1 Virtualisierung Hypervisor Typ 1 (Eigene Darstellung)

2.2 Desktopvirtualisierung

Unternehmen setzen vermehrt auf eine Modernisierung ihrer IT-Infrastruktur. Die Zentralisierung mit Hilfe von Virtualisierung dient zur Steigerung ihrer Flexibilität und zur Optimierung der Wirtschaftlichkeit.

Die Virtualisierung von Desktops verlagert die Rechenleistung und die Ressourcen des herkömmlichen Desktop-PCs in das Rechenzentrum einer IT-Infrastruktur [Frauenhofe (2011, S.8)]. Durch die Entkopplung des Betriebssystems können auf den entfernten physikalischen Hosts mehrere logische Desktopbetriebssysteme zeitgleich verarbeitet werden. Die Desktopvirtualisierung stellt sich schematisch gleich dar wie die Abbildung 1. Zusätzlich gehört zur Infrastruktur der Desktopvirtualisierung ein Connection Broker Kapitel 2.3.2. Die zentral gespeicherten und gemanagten Desktopimages bestehen aus einem Betriebssystem und deren zugehörigen Programmen, die der Benutzer bereits von seinem klassischen Computer kennt. Der Zugriff auf den Desktop erfolgt mittels eines Zero-Clients, Thin-Clients oder eines mobilen Endgerätes [Lampe (2010, S.78)]. Die Desktopvirtualisierung ist die logische Weiterentwicklung der Server- und Speichervirtualisierung. Somit besitzt das Konzept der Desktopvirtualisierung auch die ähnlichen Vorteile, wie z.B. das effizientere Ausnutzen der Hardware, die Reduzierung von Anschaffungskosten für die Clients, der geringerer Energieverbrauch und das Minimieren des Personalaufwandes bei der Administration vor Ort. Der Lebenszyklus eines Zero-Clients oder Thin-Clients erhöht sich auf 5 Jahre im Vergleich zum traditionellen PC, der nur einen Zyklus von 3 Jahren besitzt.

2.3 Virtual Desktop Infrastructure

In der Einleitung wurde die Begrifflichkeit Virtual Desktop Infrastructure, verallgemeinert als Desktopvirtualisierung betitelt. Der Begriff VDI (Virtual Desktop Infrastructure) wurde von dem Unternehmen VMware ins Leben gerufen und beinhaltet das Bereitstellen von Desktopbetriebssystemen als virtuelle Maschinen, welche auf zentralen Servern gespeichert und verwaltet werden.

Weitere bekannte Anbieter sind Citrix mit *XenDesktop* und Microsoft mit der *VDI Suite*. Zu einer vollständigen VDI gehören immer mehrere Komponenten, wie z.B. ein Hypervisor, ein Connection Broker, die Management-Tools für die Verwaltung und das Übertragungsprotokoll zur Übermittlung und Darstellung der virtuellen Bildschirmhalte. Jeder der genannten Anbieter verwendet dafür seine selbstentwickelten Komponenten und versucht diese am Markt zu etablieren. In den folgenden Unterkapiteln erfolgt eine allgemeine Erläuterung der einzelnen Komponenten, die zu einer VDI gehören, ohne dabei auf Spezifikationen einzelner Hersteller einzugehen.

2.3.1 Hypervisor

Die zentrale Software zur Abstraktion der Systeme ist der Hypervisor. Eine grundsätzliche Kategorisierung lässt sich in vollständige Virtualisierung und Paravirtualisierung gliedern. Die Paravirtualisierung findet sehr oft Anwendung bei Heimarbeitsplätzen, sie wird charakterisiert durch eine Software wie z.B. *Virtual PC* oder *VMware Fusion*, die als Voraussetzung immer ein Gastgeberbetriebssystem benötigt. Die fachspezifische Bezeichnung der Paravirtualisierung, nennt sich Hypervisor Typ 2. Die vollständige Virtualisierung beschreibt den Typ 1 des Hypervisor, bekannt als *Bare Metal Hypervisor* und ist auch für die Desktopvirtualisierung in Rechenzentren von Relevanz.

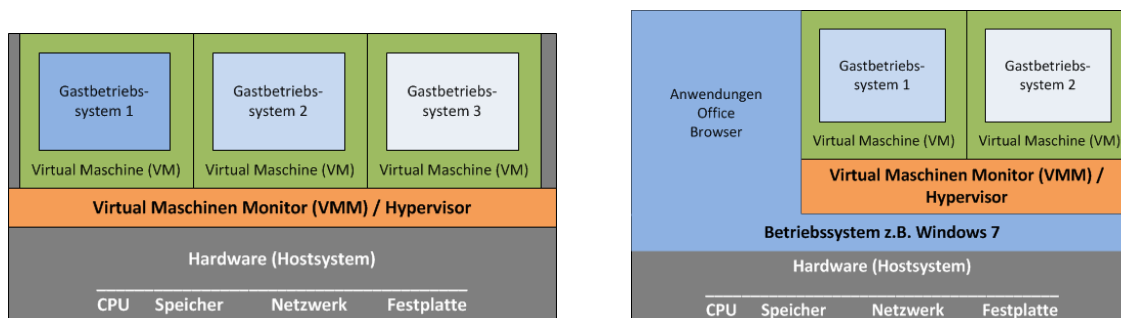


Abbildung 2 Gegenüberstellung vollständige und Paravirtualisierung (Eigene Darstellung)

2.3.2 Connection Broker

Bei der VDI ist der Connection Broker das zentrale Bindeglied zwischen dem Benutzer und seinem entfernten virtuellen Desktop. Bei Bedarf stellt er die Verbindung zum virtuellen Desktop her. Der Benutzer authentifiziert sich und anhand seiner Identität wird ihm der entsprechende Desktop zugewiesen.

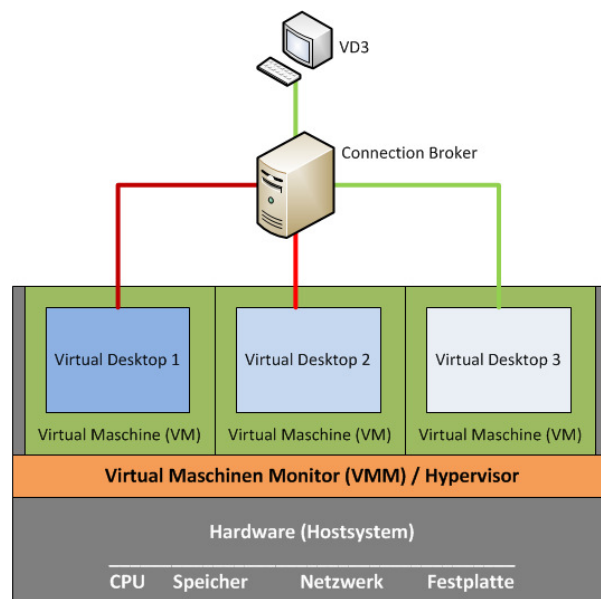


Abbildung 3 VDI Connection Broker (Eigene Darstellung)

2.3.3 Übertragungsprotokoll

Die Übertragungsprotokolle gewährleisten die Übermittlung der Remote-Displays vom Server des Rechenzentrums zum jeweiligen Endgerät. Das Protokoll überträgt die Daten und ist z.B. verantwortlich für die Darstellung der Monitorauflösung, für eine Komprimierung der Übertragung, das Verwenden von USB am Zero-Client oder das Mappen von Laufwerken. Zu den bekanntesten Übertragungsprotokollen gehören RDP/RemoteFX (*Remote Desktop Protocol*) von Microsoft, PCoIP (PC-over-Internet-Protocol) von Teradici [teradici] und HDX (*High-Definition Performance*) von Citrix. Bei jedem der Protokolle erfolgen die Verarbeitung der Darstellung und die Übertragung auf andere Art und Weise. So erfolgt z.B. beim PCoIP das Rendern von Grafiken durch die servereigene CPU.

2.4 Thin-Client Infrastructure

Die Definition für eine Thin-Client Infrastructure ist nicht eindeutig wiederzugeben. Thin-Clients werden im Zusammenhang mit dem Zugriff auf zentralisierte Ressourcen häufiger verwendet, so z.B. auch beim Konzept *Server Based Computing*. Server Based Computing ist eine Alternative zu einer Server / Client Architektur und basiert auf den Zugriffen von Thin-Clients auf Terminalserver.

Für die hier beschriebene Thin-Client Infrastructure ist der Zugriff durch Thin-Clients auf zentralisierte Festplatten, die sich im entfernten SAN (Storage Area Network) eines Rechenzentrums befinden, gemeint. Die Kommunikation zwischen dem Thin-Clients und ihren Festplatten erfolgt über das herkömmliche Netzwerk. Die verwendete Übertragung aller Daten erfolgt auf der Basis des iSCSI-Protokolls *Kapitel 2.4.1*. Die Installation von Software gewährleistet eine zentrale Verwaltung aller zur Verfügung stehenden Festplatten, welche mittels eines Netzwerkboots durch den entfernten Thin-Client, gestartet werden können.

2.4.1 iSCSI

iSCSI (*Internet Small Computer System Interface*) leitet sich aus dem Protokoll SCSI (*Small Computer System Interface*) ab. SCSI stammt aus der Datentechnik und ermöglicht die Anbindung mehrerer Geräte (Festplatte, CD-Laufwerke, Bandlaufwerke) an einen Host-Adapter. Somit war es möglich, in einem einzelnen Computer mehrere Peripheriegeräte parallel zu betreiben.

iSCSI nutzt diesen Befehlssatz von SCSI zum Ansprechen von Geräten und versendet ihn verpackt in TCP/IP Pakete über das Netzwerk. Dabei gibt es zwei Kommunikationspartner bei einer iSCSI Verbindung. Ein Client, welcher als iSCSI-Initiator bezeichnet wird, kann sich mit einem Server (iSCSI-Target) verbinden.

2.4.2 Unterscheidung Thin-Client und Zero-Client

Für das Projekt ist der Einsatz eines Zero-Clients oder eines Thin-Clients im Universitätsrechenzentrum ein wesentlicher Unterschied, welcher hier kurz hervorgehoben werden sollte.

Die Hauptaufgabe des Zero-Clients ist die primitive Darstellung von Bild, Ton und der Übermittlung von Benutzerinteraktionen des virtuellen Desktops. Der Anwender arbeitet auf dem entfernten Server mit seinem persönlichen Desktop. Der Zero-Client besitzt keine Festspeichereinheit wie z.B. eine Festplatte. Die benötigte Rechenleistung und die Bereitstellung von Grafikressourcen erfolgt vom entfernten Server innerhalb der Virtual Desktop Infrastructure.

Der Thin-Client hingegen besitzt im Vergleich zum Zero-Client ein anderes Konzept zur Verarbeitung und Bereitstellung der Rechenleistung. Der wesentliche Unterschied ist neben der Bauform und Größe auch die zur Verfügung stehende Grafikkarte innerhalb der Thin-Clients. Der Thin-Client übernimmt die Rechenleistung und die Verarbeitung von Grafikressourcen selbstständig. Genau wie der Zero-Client besitzt der Thin-Client ebenfalls keine eigene, eingebaute Festplatte.

3 Thin Client Infrastructure

In diesem Kapitel erfolgt eine technische Implementierung einer Thin-Client Infrastructure auf Basis des Übertragungsprotokolls iSCSI. Dabei liegt der Fokus auf der Auswahl einer geeigneten Lösung und deren praktischen Umsetzung.

3.1 Aufbau und Netzwerkstruktur

Bei der Thin-Client Infrastructure handelt es sich um ein *zentrales Desktop Management*. Die Verwaltung, Speicherung und Bereitstellung von virtuellen Festplatten erfolgt an einer zentralen Stelle. Der Einsatz eines Hypervisor, der eine Virtualisierung ermöglicht, ist für dieses Konzept nicht vorgesehen. In der Theorie bedeutet Virtualisierung die Bereitstellung von mehreren logischen (virtuellen) Systemen auf einem physischen Serversystem sowie deren Nutzung von zugeteilten Hardwareressourcen. Bei der folgenden Implementierung kommt es zur Speicherung von mehreren virtuellen Festplatten durch sogenannte LUN's (Logical Unit Number) auf einem Serversystem, welche aber nicht zum Betrieb der Thin-Clients auf die physischen Hardwareressourcen der Server zurückgreift. Die Rechenleistung zur Verarbeitung der entfernt gespeicherten Festplatten erfolgt durch die verwendeten Thin-Clients.

Der Einsatz zentralisierter Desktops erfolgt auf der Basis eines iSCSI-Targets. Das iSCSI-Target gewährleistet die zentrale Verwaltung oben genannter LUN's. LUN's werden über das iSCSI-Protokoll adressiert und ermöglichen so die Anbindung des SAN (Storages Area Networks) des Universitätsrechenzentrums an das iSCSI-Target. LUN's sind im SAN über ein DISK-Array verteilte virtuelle Festplatten, die vom Betriebssystem (Windows/Linux) als eigenständige Festplatte erkannt und genutzt werden können. Das Booten der virtuellen Festplatten erfolgt über das Netzwerk durch einen konfigurierten PXE-Bootloader. Die folgende Abbildung stellt den Aufbau der Thin-Client Infrastructure und den gleichzeitigen Ablauf einer Verbindung vom Thin-Client zum iSCSI-Target dar.

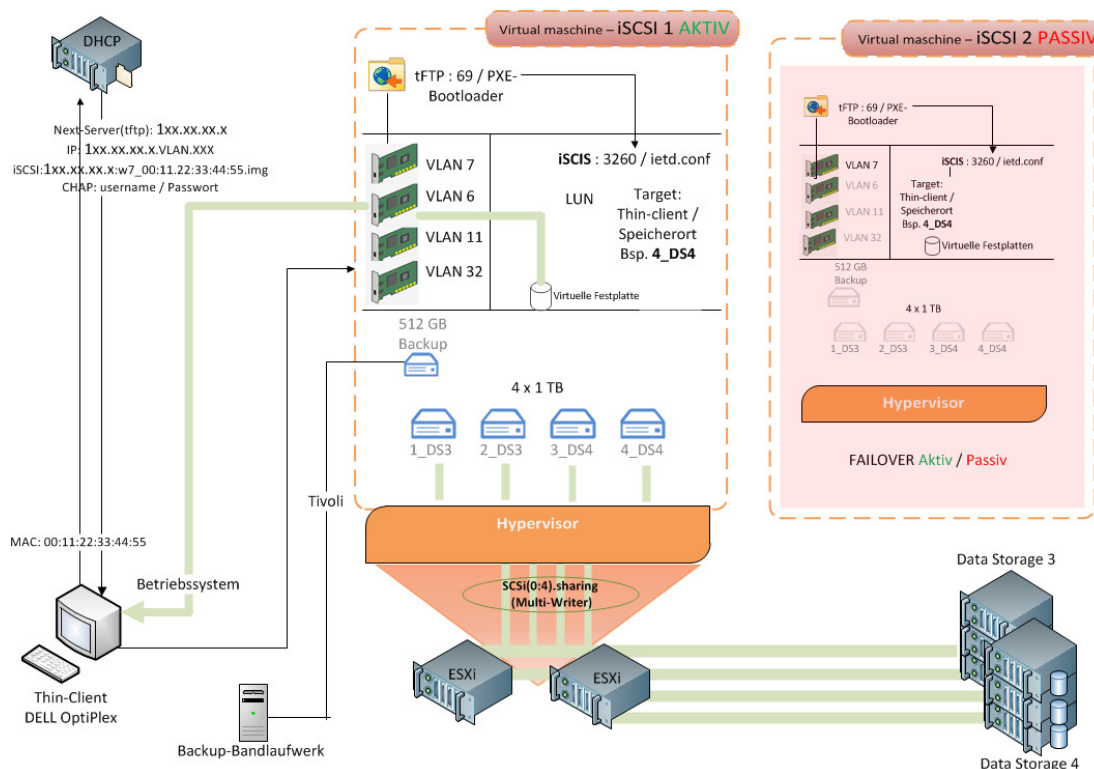


Abbildung 4 Server und Netzwerkstruktur iSCSI (Eigene Darstellung)

Die Darstellung zeigt im linken Teil, die Thin-Client Autorisierung am zentralen DHCP-Server und die Kontaktaufnahme zum iSCSI-Target über die im DHCP hinterlegten Konfigurationen. Der mittlere Bereich, stellt die iSCSI-Server dar. iSCSI1 und iSCSI2 sind zwei virtuelle Maschinen und sie befinden sich auf unterschiedlichen ESXi-Hosts (VMware Hypervisor) innerhalb der Servervirtualisierung des Universitätsrechenzentrums Greifswald. Die beiden Server sind in einem aktiv / passiv Konstrukt so konzeptioniert und konfiguriert, dass bei einem Ausfall des aktiven Servers der passive Server mit dem gleichen Konfigurationsstand die aktive Rolle übernehmen könnte. Im rechten Teil erfolgt die Anbindung des SAN an die virtuellen Server. Die Anbindung des Storage ermöglicht die Speicherung der virtuellen Festplatten auf unterschiedlichen Disk-Arrays innerhalb des Speichernetzwerkes. Eine Backup Festplatte von 512 GB steht für die Sicherung aller Konfigurationsdateien, Skripte und Master-Images zur Verfügung.

3.2 Implementierung

3.2.1 Boot Vorgang

Eines der entscheidenden Merkmale einer Thin-Client Infrastructure auf der Basis von iSCSI ist das Booten eines Betriebssystems über das Netzwerk. Der Bootvorgang unterscheidet sich beim herkömmlichen Starten von einer lokalen Festplatte im Wesentlichen davon, dass beim iSCSI Boot alle notwendigen Startinformationen in einem Bootloader zusammengefasst werden und dieser auf einem entfernten Server bereitgestellt wird. Die Thin-Clients sind durch das Entfernen ihrer lokalen Festplatte nicht in der Lage, einen normalen Systemstart auf herkömmliche und altbekannte Weise durchzuführen. Die elementar wichtigste Voraussetzung für das Booten über ein Netzwerk ist die verbaute Netzwerkkarte, welche einen sogenannten PXE-Bootloader (Preboot Execution Environment) integriert haben muss. Der Bootvorgang wurde in der folgenden Abbildung visualisiert.

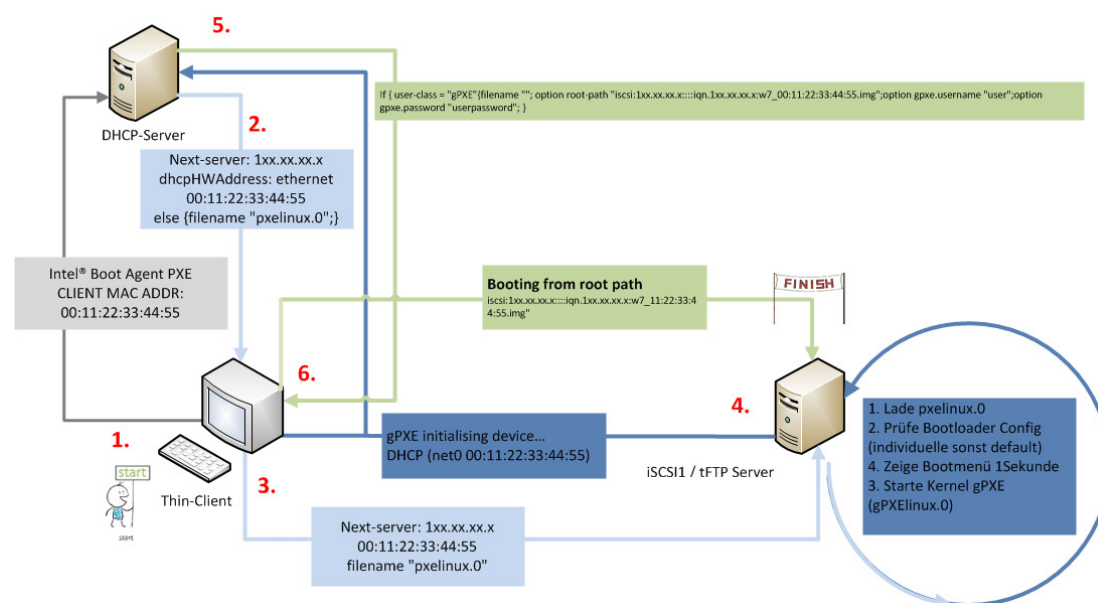


Abbildung 5 Bootvorgang iSCSI (Eigene Darstellung)

Der gesamte Bootvorgang setzt einzelne Ablaufschritte voraus, welche zusammengesetzt zu einem erfolgreichen Starten des Betriebssystems führt. Im Folgenden werden die einzelnen Schritte zum Verständnis kurz und mit ihren wichtigsten Konfigurationsparametern zusammengefasst. Dies ermöglicht einen

groben Überblick und erläutert den Zusammenhang des Bootvorgangs. Die Voraussetzung für den Thin-Client ist, dass im BIOS die Netzwerkkarte auf die Position eins der Bootreihenfolge gesetzt ist.

Der **1. Ablaufschritt** startet mit dem Einschalten des Thin-Clients.

Der PXE-Bootloader versucht eine Kontaktaufnahme zu einem im Netzwerk befindlichen DHCP-Server. Es wird ein mehrstufiges Verfahren, das sogenannte *3-way handshake* Verfahren, ausgelöst, welches zur Autorisierung zwischen Client und Server dient. Bei der Kontaktaufnahme übermittelt der Thin-Client seine MAC-Adresse an den DHCP-Server.

Im **2. Ablaufschritt** prüft der DHCP-Server seine Konfiguration und vergleicht die kontaktierte MAC-Adresse mit den hinterlegten Konfigurationseinträgen. Bei einer Übereinstimmung werden die Informationen an den Client übermittelt.

Der Thin-Client wird angewiesen, einen weiteren Server zu kontaktieren, um den dortigen Bootloader *pxelinux.0* zu starten. Der *Next-Server* ist eine DHCP-Option und stellt in der umgesetzten Implementierung einen TFTP-Server dar. Die Aufgabe des TFTP Server ist so trivial wie der Name selbst. Der TFTP (trivial File Transfer Protocol) gewährleistet eine Freigabe im Netzwerk, in der sich der Bootloader *pxelinux.0* befindet.

Im **3. Ablaufschritt** kontaktiert der Thin-Client den TFTP-Server mit dem Ziel den *pxelinux.0* Bootloader auszuführen.

Im **4. Ablaufschritt** verarbeitet der *pxelinux.0* Bootloader die ankommende Anfrage und prüft erneut anhand der MAC-Adresse, ob eine für diesen Thin-Client individuelle Bootloaderkonfiguration hinterlegt ist. Wenn dies nicht der Fall ist, wird eine *default* Konfiguration gestartet. Inhalt der Bootloaderkonfiguration ist z.B. das Bootmenü, welches dem Anwender beim Starten angezeigt wird. Ein individualisierter Bootloader wird verwendet, wenn der Benutzer zum Starten zwei Betriebssysteme zur Auswahl hat.

Im **5. Ablaufschritt** startet ein kleiner Kreislauf, welcher erneut den zentralen DHCP-Server kontaktiert. Der DHCP-Server erkennt jetzt, im Vergleich zur ersten Kontaktaufnahme, dass der Thin-Client einen anderen Bootloader gestartet hat. Der

gPXE-Bootloader ist dem DHCP-Server bekannt und er hält entscheidende Konfigurationsparameter zum Kontaktieren des iSCSI-Targets für den Thin-Client vor.

Im **6. Ablaufschritt** erfolgt die Kontaktaufnahme mit dem iSCSI-Target. Die übermittelte Konfiguration beinhalten den *root-path*, *username* und *password* zum erfolgreichen kontaktieren des iSCSI-Target's.

Dieses Boot-Schema ermöglicht den Administratoren eine zentrale Verwaltung aller Bootoptionen jedes einzelnen Thin-Clients. Der Thin-Client wird einmalig bei der Einrichtung durch die BIOS Einstellung vorbereitet und im späteren Verlauf nicht weiter betrachtet. Dies hat zum Vorteil, dass der zuständige Administrator wenig Arbeitszeit vor Ort beim Anwender am Thin-Client aufwenden muss.

3.2.2 Übertragungssicherheit

iSCSI überträgt den Datenverkehr über ein herkömmliches TCP/IP Netzwerk. Durch die Übertragung über TCP/IP dient iSCSI als eine kostengünstige Alternative zur Anbindung an ein SAN im Vergleich zum Fibre Channel Standard. Die Übertragung des gesamten TCP/IP Traffic inklusive der iSCSI Datenpakete erfolgt im Klartext. Zu einer Konsequenzen iSCSI Implementierung gehören effektive Sicherheitsmechanismen, um Angriffen präventiv entgegenzuwirken. In den Request of Comments *[RFC 7143 iSCSI]* lautet der empfohlene Verschlüsselungsmechanismus IPsec. Auf Basis von IPsec würde der gesamte TCP/IP-Traffic verschlüsselt werden. IPsec verschlüsselt jedes Datenpaket, was aber die Beeinträchtigung mit sich bringt, dass zusätzlich zur Übertragung der iSCSI Pakete auch jedes Paket vorher verschlüsselt und am Endpunkt wieder entschlüsselt werden müsste. In der Ressourcenauslastung bedeutet dieses Verfahren Einbußen in der Performance der Netzwerkübertragung und einer zusätzlichen Arbeitslast der CPU. In der Theorie ist die Verschlüsselung mit IPsec die beste Wahl, wenn die Sicherheit der Datenübertragung als einziges Kriterium betrachtet wird. Die Benutzer, die täglich mit einem Thin-Client auf einem iSCSI-Target arbeiten, setzen ein verzögerungsfreies und schnelles Arbeiten voraus. Bei einer IPsec

Verschlüsselung würde die Übertragungsgeschwindigkeit zwischen dem iSCSI-Initiator und dem iSCSI-Target Verluste erleiden. Im Folgenden wird eine Studie [Weiping, Wandong (2005, S.457-462)] zur Sicherheit von iSCSI Übertragungen herangezogen. Die Gegenüberstellung der Ergebnisse verdeutlicht die zusätzliche Ressourcenauslastung beim Einsatz von Verschlüsselungsmethoden. Aus der Studie geht hervor, dass IPsec die empfohlene und sicherste Übertragung ist, aber es bis zu 100% mehr CPU-Last benötigt. Die Abbildung 6 zeigt einen Vergleich zwischen iSCSI + IPsec, iSCSI + SSH (Secure Shell) und iSCSI ohne Verschlüsselung.

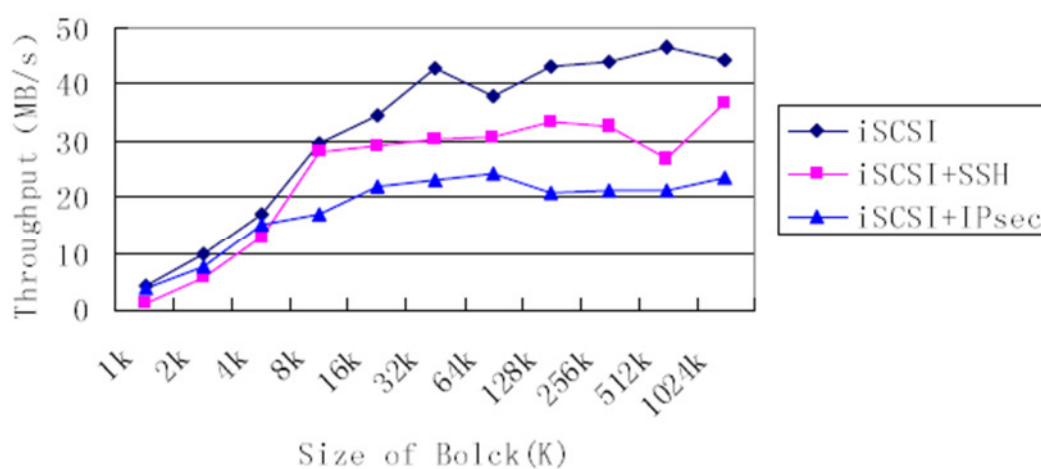


Abbildung 6 iSCSI Übertragungsgeschwindigkeit Quelle: Weiping, Wandong (2005, S. 457-462)

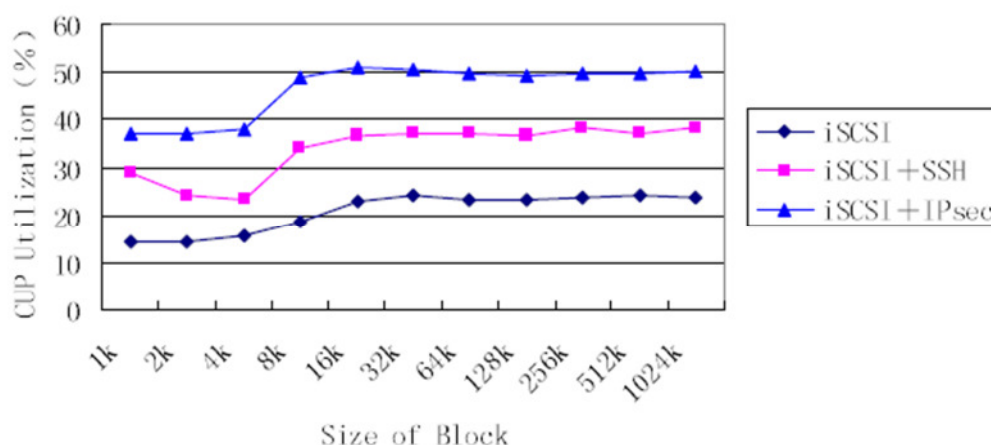


Abbildung 7 iSCSI CPU Auslastung Quelle: Weiping, Wandong (2005, S. 457-462)

Die prozentualen Performancewerte aus der Studie sind nicht auf die eigene Umgebung übertragbar, weil sie auch ein Spiegel der eingesetzten Hardware und IT-Infrastruktur sind. Sie dienen an dieser Stelle als Illustration für den Verlust von Geschwindigkeit und Rechenressourcen bei dem Einsatz von Verschlüsselungen. Die SSH Verschlüsselung benötigt weniger Leistung im Vergleich zur IPSec Verschlüsselung. Der Einsatz in der geplanten Thin-Client Infrastruktur im Universitätsrechenzentrum ist vorerst aber ausgeschlossen, weil die Implementierung und Verfügbarkeit eines SSH Clients, wie z.B. *openSSH*, noch nicht nativ auf dem Thin-Client Betriebssystem Windows verfügbar ist. Die Verwendung von IPSec wird aus Performancegründen nicht für die Thin-Client Infrastruktur favorisiert.

Bei einer Übertragung im Klartext könnte ein potentieller Angreifer den Datenverkehr mitlesen. Um den Datenverkehr zwischen dem Thin-Client und dem iSCSI Target mitzulesen, müsste der Angreifer ein sogenanntes ARP-Spoofing [*ARP-Spoofing*] durchführen.

Das ARP Spoofing dient zur Manipulation der ARP-Tabellen im Netzwerk und ermöglicht das Abhören von zwei oder mehr Computersystemen gleichzeitig. Das ARP-Spoofing ist der Einstieg für einen oft beschriebenen MAN-In-The-Middle-Angriff. Um das Mitlesen über ein MAN-In-The-Middle-Angriff fast unmöglich zu machen, wird der Traffic vom Thin-Client zum iSCSI-Target in seiner Zielstruktur physikalisch nicht über den Hauptrouter geleitet. Der Hauptrouter befindet sich auf Layer 3 des OSI Schichtenmodells und würde ein ARP-Spoofing ermöglichen. Die Thin Client Infrastruktur wird in einem eigenen VLAN implementiert, welches über einen Switch auf Layer 2 Ebene angeschlossen ist. Durch die Umsetzung dieser physikalischen Zielstruktur bleibt die Performance erhalten und die Überwindbarkeit für ein gezieltes Mitschneiden des TCP/IP Traffic wird erhöht.

3.2.3 Authentifizierungsmethode

iSCSI unterstützt drei unterschiedliche Authentifizierungsmethoden [RFC 7143, (S.223)]: KerberosV5, Secure Remote Password und Challenge Handshake Authentifikation Protocol. Jede der drei Authentifizierungsmöglichkeiten erhöht die Sicherheit und verhandelt vor dem Login die Integrität der Verbindung. Ein persönlicher Test mit einer Ubuntu Linux Distribution als iSCSI-Initiator hat ergeben, dass es ohne die Verwendung einer Authentifizierungsmethode sehr leicht möglich ist, innerhalb des Netzwerkes eine Verbindung zu jedem iSCSI-Target aufzubauen, welches vom iSCSI-Server zur Verfügung gestellt wurde. Aus diesem Anlass ist die Implementierung einer praktikablen Authentifizierungsmethode unbedingt notwendig. Die mit am häufigsten verwendete und von den meisten iSCSI Speicherlösungen unterstützte Methode ist *CHAP (Challenge Handshake Authentifikation Protocol)*. Die Implementierung ist simpel, aber effektiv, wobei man z.B. für eine Kerberos-Authentifizierung zusätzliche Implementierungen benötigt. Aus diesen Gründen fiel die Entscheidung auf die CHAP [CHAP RFC 1994] Authentifizierungsmethode. Jedes Target auf dem iSCSI-Server kann mit einer Kombination aus einem Benutzernamen und einem Passwort abgesichert werden. Auf dem Server befindet sich dazu, innerhalb der zentralen Konfigurationsdatei **ietd.conf**, die Option **IncomingUser** mit dem Benutzernamen und dem Passwort. In der Zielstruktur ist vorgesehen, dass jeder Thin-Client unterschiedliche Benutzer- und Passwort-Kombinationen zugewiesen bekommt. Die Kombination wird als Konfigurationsparameter im DHCP-Server hinterlegt, bei welchem sich der Thin-Client während des Bootens autorisiert und diese übermittelt bekommt. Die CHAP-Authentifizierung erfolgt in einer 3-Way Handshake Challenge zwischen dem Client und dem Server, in dem versteckt aus Zufallszahlen und dem Passwort ein Hashwert ausgetauscht wird. Dieser Hashwert ist einmalig und mit MD5-Algorithmus [MD5 RFC1321] kryptisch verschleiert. Die Challenge erfolgt nicht nur vor sondern wiederholt sich periodisch auch während einer aktiven Verbindung, um die Authentizität fortlaufend zu gewährleisten.

Dabei ist zu beachten, dass MD5 heute nicht mehr als sicher gilt. MD5 weist Schwächen auf, die bereits anhand von praktischen Beispielen demonstriert wurden [BSI]. Aktuelle Alternativen zum MD5-Algorithmus sind die geeigneten Hashfunktion der SHA-2 Familie, sie gewährleisten laut der Bundesnetzagentur ein langfristiges Sicherheitsniveau [Bundesnetzagentur].

3.2.4 Cluster

In der Abbildung 4 wurde visuell verdeutlicht, dass das entworfene Konzept zur Ausfallsicherheit einen zweiten Server (*iSCSI2*) vorhält, welcher beim Ausfall von *iSCSI1* aktiviert wird. In diesem Abschnitt wird auf den Einsatz eines Cluster-Filesystem eingegangen, welches durch ein Testszenario auf Tauglichkeit für den aktiven Einsatz geprüft wurde. Bei einem Cluster-Filesystem ist es möglich, dass ein oder mehrere Rechner auf dasselbe Dateisystem zugreifen. Die erste Herangehensweise, um ein redundantes Vorhalten der virtuellen Festplatten auf beiden Servern zu ermöglichen, war das Implementieren eines verteilten Dateisystems über zwei Nodes. Nodes beschreiben die Cluster-Knoten innerhalb einer Cluster-Architektur, welche in diesem Fall *iSCSI1* und *iSCSI2* darstellen. Innerhalb der Open-Source Community gibt es eine Auswahl unterschiedlichster Cluster-Systeme, wie z.B. DRBD [DRBD], XTREEMFS [XTREEMFS] oder GlusterFS [GlusterFS]. Letzteres wurde ausgewählt, weil im Universitätsrechenzentrum bereits Erfahrungen im aktiven Einsatz mit GlusterFS gesammelt wurden. Das GlusterFS wurde im Replicated Mode konfiguriert und die Testumgebung aufgebaut.

Die Testdaten, die zur Replizierung mit GlusterFS zum Einsatz kommen, sind die zentralisierten Festplatten aller Thin-Clients, jeweils mit einer Größe von 80GB.

Es wurde ein Test durchgeführt, um herauszufinden, ob das GlusterFS für den späteren Einsatz geeignet ist. Es wurde ein Szenario aufgebaut, welches im aktiven Betrieb das Ausrollen und Initialisieren von Thin-Clients darstellt. Der Initialisierungsdurchlauf eines Thin-Clients erfolgte mit dem ausgerollten Master Image, dabei wurde der Rechner automatisch mit SYSPREP [SYSPREP] konfiguriert

und in die Windows-Domain des Rechenzentrums mit aufgenommen. Somit sind die folgenden Zeitangaben nicht mit einem gewöhnlichen Bootvorgang zu vergleichen.

Testszenario (mit GlusterFS / 2 CPU / 1GB RAM / Storage 1 x 4TB):

Die Testumgebung bestand aus einem PC-Labor mit 20 Thin-Clients, welche über ein VLAN mit den iSCSI1 Server verbunden waren. Die Serverhardware beschränkte sich für den ersten Testdurchlauf auf 2 CPU's, 1GB Arbeitsspeicher und einer 4TB Platte aus dem SAN, adressiert über iSCSI. Der erste und einfachste Test war das Starten eines Thin-Clients, er belief sich auf eine Referenzzeit von *2.35 min*, welches mit der Stoppuhr ermittelt wurde. Die Startzeit ist ein erster greifbarer Indikator für die Performance des gesamten Systems.

Das zweite Testszenarium beinhaltete das Starten aller 20 Thin-Clients gleichzeitig, welches aber nicht mehr im Rahmen einer akzeptablen Zeit von 34 min erfolgreich abgeschlossen werden konnte. Die Ursache war die Auslastung der CPU's auf dem iSCSI-Server, welche durch die Synchronisation des GlusterFS nahtlos bei 100% verweilte.

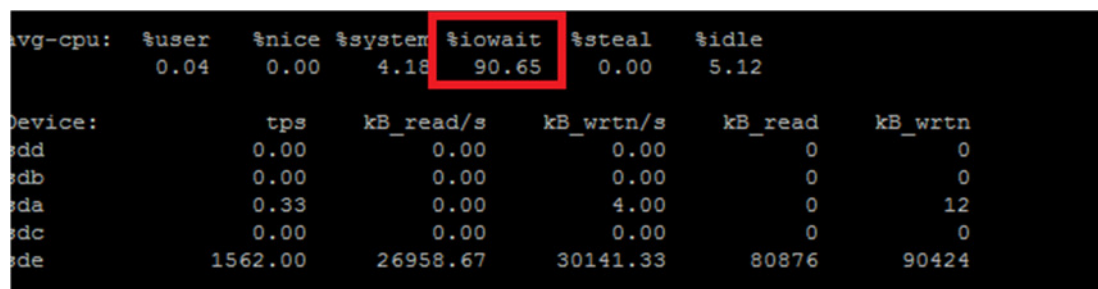
Testszenario (mit GlusterFS / 8 CPU's / 8GB RAM / Storage 1 x 4TB)

Ein Vorteil virtueller Maschinen ist es, in kurzer Zeit neue Hardwareressourcen zuzuteilen. Somit wurden die Ressourcen auf 8 Kerne (vCPU's) und 8GB Arbeitsspeicher erhöht. Die 4TB Festplatte behielt weiterhin ihren Bestand. Das erneute Starten von 20 Thin-Clients ergab eine durchgängige Auslastung von 50-70% der 8 CPU's. Die Startzeit reduzierte sich auf 4.30 min für 20 Thin-Clients gleichzeitig. Der Durchlauf konnte abgeschlossen werden, aber das Ergebnis war nicht zufriedenstellend. Auch wenn sich die erreichte Zeit im Vergleich zum ersten Test durch die Aufstockung der Hardware wesentlich verbessert hat, ist die Performance noch nicht ausreichend. Das zusätzliche Aufstocken der Serverhardware würde die Zeit bei 20 Thin-Clients ggf. noch verkürzen, aber bei einer Skalierung von mehr als 20 Thin-Clients würde es nicht zu einem zufriedenstellenden Ergebnis kommen.

Auf der Suche nach der Schwachstelle, ergab sich ein hoher **IOWAIT Wert** von über 90% der gesamten Testlaufzeit. Der IOWAIT kann mit folgendem Befehl ermittelt werden.

```
iostat 3
```

Die 3 gibt das Aktualisierungsintervall in Sekunden an. Folgende Abbildung zeigt einen Ausschnitt der IOSTAT während des Testdurchlaufs.



avg-cpu:	%user	%nice	%system	%iowait	%steal	%idle
	0.04	0.00	4.18	90.65	0.00	5.12
Device:	tps	kB_read/s	kB_wrtn/s	kB_read	kB_wrtn	
edd	0.00	0.00	0.00	0	0	
edb	0.00	0.00	0.00	0	0	
eda	0.33	0.00	4.00	0	12	
edc	0.00	0.00	0.00	0	0	
ede	1562.00	26958.67	30141.33	80876	90424	

Abbildung 8 Serverauslastung IOSTAT (Screenshot iSCSI1)

IOWAIT gibt den Zeitanteil in Prozent an, den das System damit verbringt, auf einen I/O Request der Festplatte zu warten.

Das Testergebnis erbrachte die Erkenntnisse, dass dabei zwei entscheidende Faktoren zum Tragen kommen. Zum einen die hohe CPU-Last, verursacht durch die Replizierung auf den iSCSI2 durch das GlusterFS. Die Auslastung von über 50% bei 8 zur Verfügung stehenden Kernen bei einer Anzahl von nur 20 Thin-Clients lässt eine steigende Skalierung von weiteren Thin-Clients in der Aktivumgebung nicht zu.

Zum anderen gibt es ein weiteres Problem: Die eine 4 TB Festplatte mit allen darauf gespeicherten Images kann die Verarbeitung aller anstehenden Aktivitäten nicht performant gewährleisten.

Das Fazit aus dem Testszenario ist, dass das GlusterFS eine gute Möglichkeit darstellt, um Daten auf mehreren Servern über ein verteiltes Dateisystem zu replizieren. Zu beachten ist dabei aber das Einsatzszenario. Für die Replizierung mit großen Dateien, wie in den Testszenarien mit den Festplatten (20 x 80 GB), auf denen permanent zugegriffen wird, und wo es enorm auf Latenzzeiten ankommt, ist

es im aktivem Umfeld eines Rechenzentrums leider nicht einsatzfähig. Somit musste die Entscheidung getroffen werden, ohne ein Cluster Filesystem weiterzumachen und eine Alternative für die Redundanz der Daten zu finden. Die Beschreibung zur Alternative befindet sich im *Kapitel 3.4.3 Festplatten*.

3.3 iSCSI - Server

Die zentrale Komponente der Thin-Client Infrastructure ist das iSCSI-Target. Das iSCSI-Target basiert in dieser Implementierung auf dem Open Source Betriebssystem Linux und wird betreut durch das Projekt *IET (The iSCSI Enterprise Target Project)* [IET]. Als Installationsgrundlage wurde die Distribution Debian in der Version 8 ausgewählt. Open Source Serverbetriebssysteme gehören zur Philosophie des Universitätsrechenzentrums und werden auch in diesem sehr häufig eingesetzt. Die Entscheidung liegt nahe, Debian als Grundlage aller folgenden Installationen auszuwählen.

3.3.1 Daemon

Vor der Installation soll eine kurze Erläuterung von Daemons erfolgen. Daemons sind kleine Programme, die auf UNIX-Systemen oder auf UNIX ähnlichen Systemen im Hintergrund laufen und bestimmte Dienste zur Verfügung stellen. Sie stellen eine zentrale Betriebssystemkomponente dar und werden sehr häufig ohne Benutzerinteraktionen durch das System beim Bootvorgang gestartet.

Von einem „Dämon“ zu sprechen wäre laut der Schreibweise nicht richtig. Die technische Abkürzung lautet „**disk and execution monitors**“, aber der bildhafte Begriff „Dämon“ ist auch in Fachbüchern verbreitet [Wolf (2006, Kap.7.12)].

3.3.2 Installation

Die Installation der benötigten Dienste für den reibungslosen Betrieb eines iSCSI-Targets auf Basis der Distribution Debian Jessie beläuft sich auf die Installation des Betriebssystems Debian selber, einen TFTP Server für das Speichern des PXE-Bootloader und dem Dienst iSCSI-Target. Da es sich bei der Installation des Debian um eine Standardinstallation handelt, wird auf die einzelnen Installationsschritte hier nicht eingegangen. Beginnend mit der Implementierung werden die folgenden Dienste installiert.

Tabelle 1 Installierte Dienste iSCSI-Target

Programm / Dienst / Port	Bemerkung	Zentrale Konfiguration
tFTP / in.tftpd /:69	Trivial FTP Server– Bereitstellung des gPXE- bootloader	/etc/default/tftpd-hpa
iSCSI-Target / ietd /:3260	Zentrales Desktop Management	/etc/iet/ietd.conf
iSCSI-Initiator	Zur ersten Überprüfung vom iSCSI-Target (Verbindungsaufbau, Login eines Target)	/etc/iscsi/iscsid.conf
SSH Server / sshd /:22	Fernadministration über SSH	

Eine detaillierte Installationsanleitung für Administratoren inklusive aller Befehle und deren Optionsparametern befindet sich im Anhang 8.1.

3.4 Konfiguration

In diesem Abschnitt werden die wichtigsten Parameter, wie die Netzwerkkonfiguration, Festplattenkonfiguration, iSCSI-Target und die Einstellungen des TFTP-Servers, erläutert und dargestellt.

3.4.1 Netzwerk

Die Bereitstellung der Desktops erfolgt in unterschiedlichen VLANs. Das iSCSI-Target sollte Mitglied in dem VLAN sein, in welchem es die Images an die Thin-Clients verteilt. Entsprechend besitzen beide iSCSI-Targets mehrere Netzwerkschnittstellen. Beide Server haben neben ihren VLAN-Schnittstellen die primäre eth0 Schnittstelle mit der IP-Adresse: 141.xx.xx.xxx (iSCSI1) und der IP-Adresse 141.xx.xx.xxx(iSCSI2).

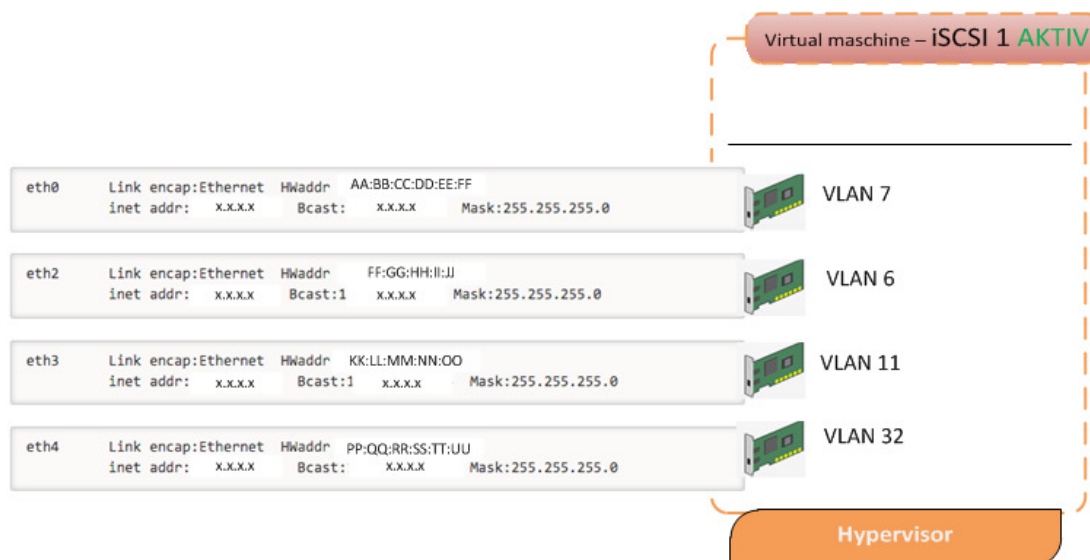


Abbildung 9 iSCSI1 Netzwerkkonfiguration (Eigene Darstellung)

Der iSCSI2 befindet sich im Passivmodus, was zur Konsequenz hat, dass die Netzwerkschnittstellen vorkonfiguriert sind, aber erst im Falle des Wechselszenarios von passiv auf aktiv hochgefahren werden dürfen. Selbst beim iSCSI1, der vorrangig den aktiven Posten besitzt, dürfen keine Netzwerkschnittstellen oder Dienste automatisch beim Systemstart hochgefahren werden.

Das automatische Starten der Netzwerkschnittstelle wird durch das Auskommentieren der Befehlszeile „*#allow-hotplug eth4*“ in der zugehörigen Konfigurationsdatei verhindert. Hier zu sehen am Beispiel von eth4.

/etc/network/interfaces.d/eth4



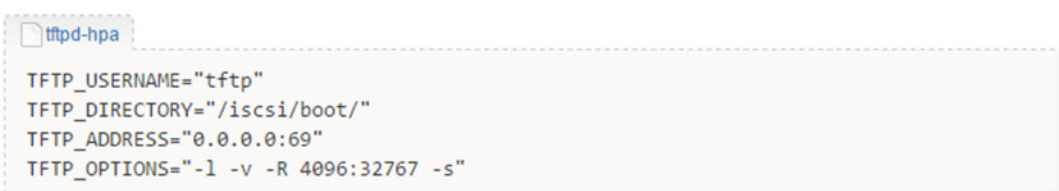
```

eth4
#allow-hotplug eth4
iface eth4 inet static
    address 1xx.xx.xx.x
    netmask 255.255.255.0
    network 1xx.xx.xx.0
  
```

Abbildung 10 iSCSI Network Interface (Datei)

3.4.2 TFTP Server

Der TFTP Server gewährleistet die Freigabe eines zentralen Ordners *TFTP_DIRECTORY="/iscsi/boot/"*, in welchem die Bootloader Konfiguration hinterlegt wird, die zum Starten über PXE vom iSCSI-Initiator benötigt werden. Dieser Server könnte auf einer beliebigen Maschine im Netzwerk installiert werden. Für diese Implementierung ist es für die eingeführte Automatisierung leichter, ihn auf dem iSCSI-Target (iSCSI1) zu installieren. Die zentrale Konfigurationsdatei befindet sich unter */etc/default/tftpd-hpa*.



```

tftpd-hpa
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/iscsi/boot/"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="-l -v -R 4096:32767 -s"
  
```

Abbildung 11 TFTP Konfiguration (Datei)

3.4.3 Festplatten

Die Erkenntnisse des Performancetestes aus dem *Kapitel 3.2.4* sorgten zum Umdenken bei der Festplattenkonfiguration und zur Suche nach einer neuen Lösung. Es wird im Folgenden eine Alternative dargestellt, welche die Performance beim Starten von 20 Thin-Clients gleichzeitig – im Vergleich zum selben Testszenarium aus Kapitel 3.2.4 – auf 1.30 min reduziert (ohne GlusterFS).

Das Universitätsrechenzentrum ist im Besitz von mehreren physisch voneinander getrennten Storage-Systemen. Somit ist es möglich, die Festplatten auf unterschiedliche Storage-Systeme zu verteilen. Das Konzept wurde von einer 4 TB Platte zu 4 x 1 TB abgeändert. Damit das Storage-System selber noch entlastet wird, wurden zwei der 1 TB Platten aus dem Datastorage 3 und die restlichen zwei aus dem DataStorage 4 zur Verfügung gestellt. Innerhalb des jeweiligen DataStorage werden die einzelnen Platten zudem auch noch auf unterschiedlichen Enclosures bereitgestellt. Die Abbildung skizziert die Festplattenkonstellation und deren Anbindung an das SAN.

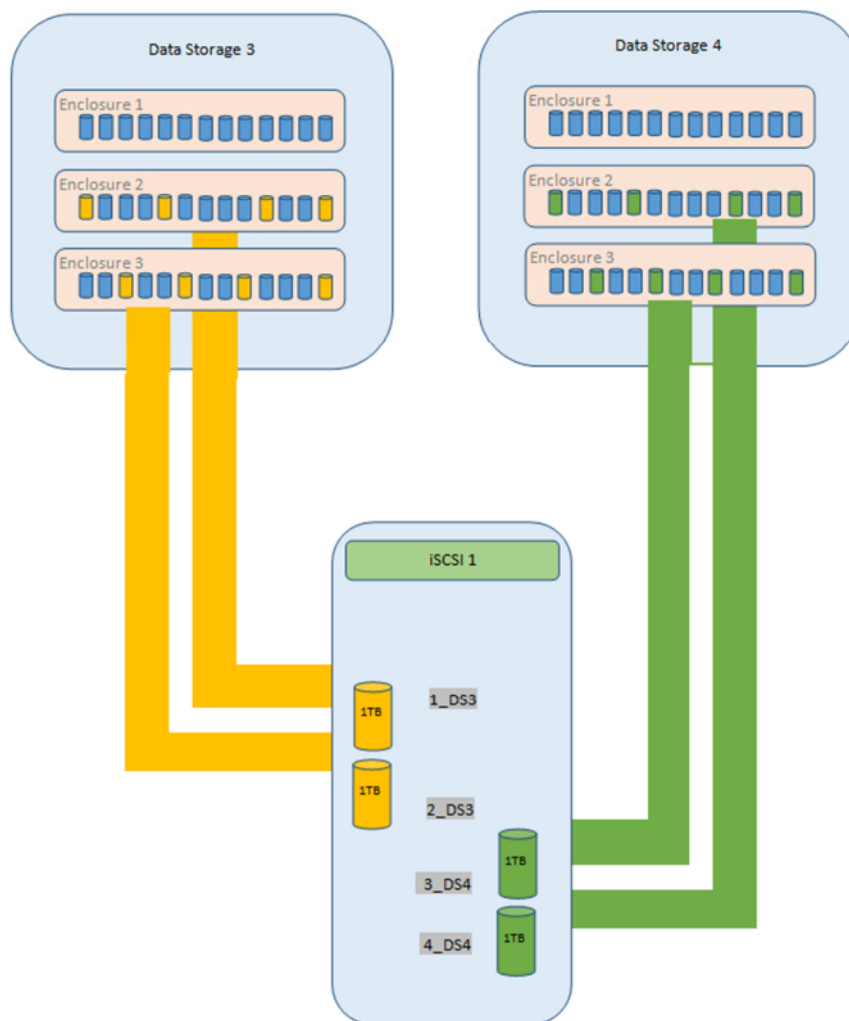


Abbildung 12 iSCSI SAN-Anbindung (Eigene Darstellung)

Diese Konstellation lässt eine absehbare Steigerung der Nutzerzahlen von Thin-Clients zu und ermöglicht durch Hinzufügen weiterer Festplatten eine angemessene Skalierung.

Die Verwendung eines Cluster-Filesystems wurde durch die erhöhte CPU-Auslastung verworfen. Das Problem die laufenden Thin-Clients bei einem Ausfall von iSCSI1 unbemerkt auf den iSCSI2 zu schwenken bleibt aber weiterhin bestehen. Es handelt sich nicht um physikalische Festplatten die in dem iSCSI1 Server eingebaut sind,

sondern um sogenannte RAW-Device aus den Storage-Systemen, die über einen *mount* Befehl an den Server angehängt werden können.

Das Notfallszenario wurde so konzeptioniert, dass bei einem Ausfall des iSCSI1 alle Festplatten durch ein ausführbares Script am iSCSI2 Server gemountet werden. Diese Konstellation gewährleistet einen weiterführenden Betrieb aller laufenden Thin-Clients.

Diese Art der Bereitstellung von Festplatten ist innerhalb der VMware vSphere 5.5 Umgebung nicht standardmäßig vorgesehen, daher bedarf es einiger Vorbereitungen und die Beachtung von Sicherheitshinweisen [*VMware Multi-Writer*]. iSCSI1 und iSCSI2 sind virtuelle Maschinen, die auf einem ESXi-Host (VMware Hypervisor) installiert sind. Die RAW-Device aus dem SAN werden innerhalb der VMware vSphere Umgebung in eine virtuelle Festplatte (VMDK) umgewandelt und einem Server fest zugeordnet. Nach der Zuordnung an einen ESXi-Host ist es mit der Standardkonfiguration nicht mehr möglich, diese Festplatten auch für einen zweiten Server zur Verfügung zu stellen. Dafür wird eine erweiterte Konfiguration aller Festplatten zu einer *Sharing Multi-writer* Festplatte [*VMware Multi-writer*] vorgenommen. Ist jede Festplatte mit dem Multi-writer Flag versehen, ist es möglich, die Platten in der Administrationsoberfläche auch parallel am iSCSI2 zu betreiben.

Nach einem Neustart vom iSCSI2 erscheinen im System auch dort die 4 x 1TB. Zur Überprüfung kann folgender Befehl verwendet werden.

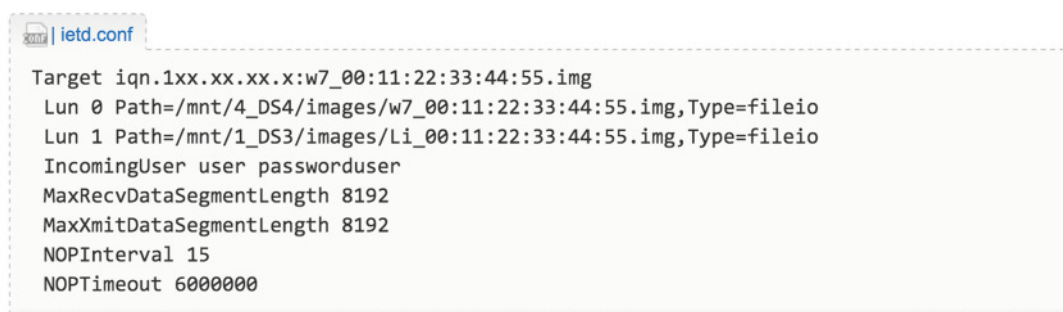
```
fdisk -l
```

Die Sicherheitsrisiken, die bei dieser Konstellation zu beachten sind, können bei Missachtung kritische Auswirkungen auf das Dateisystem der Festplatten haben. Es sollte unbedingt vermieden werden, dass die Festplatten gleichzeitig an beiden Servern verwendet werden. Schreiben beide Server zur selben Zeit in das Filesystem der Festplatten, kann es mit hoher Wahrscheinlichkeit zur Inkonsistenz und somit zu einem defekten Dateisystem kommen, was wiederum den Verlust der Images

bedeuten würde. Da die Performance innerhalb der Thin-Client Infrastructure von großer Bedeutung ist, wurde unter Beachtung der Hinweise und dem geregelten Ablauf eines Notfallszenarios eine praktikable Lösung gefunden.

3.4.4 iSCSI Enterprise Target

Das iSCSI-Target vom Open Source Projekt *iSCSI Enterprise Target* ist der Ausgangspunkt aller zentral verwalteten Festplatten der Thin-Client Infrastructure. Jeder iSCSI-Initiator verbindet sich beim Starten mit dem iSCSI-Target. Die zentrale Konfiguration erfolgt innerhalb der Datei *ietd.conf*, welche sich auf dem Server im Pfad `/etc/iet` befindet. Ein Ausschnitt aus dieser Konfigurationsdatei zeigt einen Eintrag für einen iSCSI-Initiator, jeder iSCSI-Initiator benötigt genau einen hinterlegten Konfigurationseintrag.



```
Target iqn.1xx.xx.xx.x:w7_00:11:22:33:44:55.img
Lun 0 Path=/mnt/4_DS4/images/w7_00:11:22:33:44:55.img,Type=fileio
Lun 1 Path=/mnt/1_DS3/images/Li_00:11:22:33:44:55.img,Type=fileio
IncomingUser user passworduser
MaxRecvDataSegmentLength 8192
MaxXmitDataSegmentLength 8192
NOPIInterval 15
NOPITimeout 6000000
```

Abbildung 13 iSCSI IET Konfiguration (Datei)

Die Zuordnung des jeweiligen iSCSI-Initiator beginnt mit dem Eintrag **Target** und dem folgenden global gültigen (*iqn*) Namen, welcher definiert ist durch den iSCSI-Standard [*iSCSI RFC 7143, (S. 49)*].

Die **LUN** klassifiziert die zugehörige Festplatte inklusive des hinterlegten Speicherortes auf dem Server. In diesem Beispiel ist eine zweite **LUN** definiert, welche dem iSCSI-Initiator eine zweite vollwertige Festplatte unabhängig von der ersten Festplatte zur Verfügung stellt. Diese Konstellation würde z.B. ermöglichen, ein Betriebssystem auf LUN 0 zu installieren und LUN 1 als eine eigene physikalisch getrennte Datenplatte zu verwenden. Die Festplatten werden alle innerhalb einer Betriebssystemumgebung angezeigt und zur Verfügung gestellt. Ein zweites

mögliches Anwendungsbeispiel von mehreren Festplatten wäre, dem iSCSI-Initiator ein zweites Betriebssystem anzubieten. Der Nutzer hat dann die Möglichkeit, beim Booten eine Auswahl zutreffen.

Die Option **IncomingUser** realisiert die Authentifizierung zwischen dem iSCSI-Initiator und dem iSCSI-Target, was im Kapitel 3.2.4 detailliert beschrieben wurde. Damit sind die wichtigsten Parameter für einen erfolgreichen Verbindungsaufbau gegeben. Weitere Optionen, wie z.B. **NOTimeout**, ermöglichen ein Timeout. Sollte das iSCSI-Target seine Verbindung zum iSCSI-Initiator verlieren, wird somit ein Verbindungsabbruch für die Zeit des Timeouts verhindert. Wenn der Server innerhalb des angegebenen Timeouts seine Verbindung wiederherstellt, kann der Client ohne einen Neustart oder sonstigen Verlust von Daten normal weiterarbeiten.

3.4.5 DHCP

In diesem Abschnitt wird die Notwendigkeit der Konfigurationseinträge vom Kapitel 3.4.4 hervorgehoben. Was das iSCSI-Target innerhalb der *ietd.conf* gespeichert hat, hat jeder iSCSI-Initiator im DHCP hinterlegt. Mit den tabellarisch dargestellten Optionen nimmt der iSCSI-Initiator mit dem iSCSI-Target Kontakt auf, authentifiziert sich und bekommt seine Festplatten zugeordnet. Alle DHCP Informationen werden während des Bootvorgangs an den iSCSI-Initiator übermittelt, eine ausführliche Beschreibung erfolgte im Kapitel 3.2.1.

Option	
root-path	iscsi:1xx.xx.xx.x:::iqn.1xx.xx.xx.x:w7_00:11:22:33:44:55.img
username	user
password	min12Zeichenlangespasswort
ip	1xx.xx.xx.x
netmask	255.255.255.0
gateway	1xx.xx.xx.1
mac	00:11:22:33:44:55
dhcp-server	1xx.xx.xx.x
dns	1xx.xx.xx.x
next-server (tftp)	1xx.xx.xx.x

Abbildung 14 DHCP Optionen (Eigene Darstellung)

3.4.6 Backup

Im iSCSI-Target befindet sich eine zusätzliche Festplatte zum Sichern der notwendigsten Daten. Die Arbeitsphilosophie der Anwender sollte möglichst so gestaltet sein, dass alle Nutzer der Thin-Clients die zentralen Speicherressourcen des URZ verwenden und nicht die Desktops als Speicherort benutzt werden. Es wird kein einmaliges und auch keine inkrementelles Backup der Thin-Client Images erstellt. In das Sicherungsverfahren werden nur Konfigurationsdateien, Automatisierungsscripte und die unterschiedlichen Master Images aufgenommen.

3.5 Performance

Die Bedeutsamkeit schneller Input/Output Operationen der eingesetzten Festplatten wurde während des Testszenarios in Kapitel 3.2.4 deutlich. Die I/O-Last wurde von 1 x 4 TB Festplatte auf 4 x 1 TB Festplatten verteilt. Damit die Anzahl der zu verarbeitenden Operationen gesenkt werden kann, wurden für alle Festplatten der *noop* Kernel IO-Scheduler [IO-Scheduler] aktiviert und die Protokollierung von Dateizugriffsinformationen deaktiviert. Diese Einstellungen waren noch kein Bestandteil im Testszenario aus Kapitel 3.2.4.

3.5.1 Kernel IO-Scheduler

Im Linux Kernel sind drei verschiedene IO-Scheduler vorhanden. Jeder der drei IO-Scheduler verwendet einen eigenen Algorithmus, um Prozessoperationen an den Hardware Controller der Festplatte zu übergeben. Der *noop* Kernel IO-Scheduler ist dafür konzipiert, dabei sehr wenig IO-Last zu generieren.

Da es sich bei den Festplatten um mehrere RAW-Device handelt und der Hypervisor die Schreib- und Lesevorgänge regelt, wurde der *noop* Kernel IO-Scheduler für die virtuelle Maschine aktiviert, hier am Beispiel der Festplatte „*sde*“. Für alle weiteren Laufwerksbezeichnungen ist diese Durchführung ebenfalls notwendig.

```
cat /sys/block/sde/queue/scheduler  
noop deadline [cfq]
```

```
echo noop > /sys/block/sde/queue/scheduler  
cat /sys/block/sde/queue/scheduler  
[noop] deadline cfq
```

Alternativ kann die Aktivierung des *noop* Kernel IO-Scheduler für alle Festplatten im Grub-Bootloader vorgenommen werden.

```
GRUB_CMDLINE_LINUX_DEFAULT="elevator=noop"
```

Das Aktualisieren der Konfiguration erfolgt mit dem Befehl.

```
update-grub
```

3.5.2 Mount Optionen

Mit dem *mount* Befehl werden die Festplatten am iSCSI1 und iSCSI2 angehängt. Der *mount* Befehl gestattet weitere Optionen, die für die Performance relevant sind. Für die 1 TB Festplatten wurde die Option **noatime** und **nobarrier** [*xfs*] angewandt. Die Option **Noatime** verhindert die standardmäßige Speicherung von Dateizugriffszeiten. Sollten diese nicht benötigt werden, bringt dieses einen kleinen Geschwindigkeitsvorteil.

Write Barrier kann die Integrität des Dateisystems bei einem Stromausfall gewährleisten, allerdings nur mit Verlust von Performance. **Nobarrier** wird also nur empfohlen, wenn der Cache des Raid-Arrays zusätzlich abgesichert ist. Bei allen Festplatten kommt das Dateisystem *xfs* zum Einsatz, die erwähnten Optionen sind aber auch auf anderen Dateisystemen wie z.B. *ext4* anwendbar.

Folgender Befehl zeigt ein *mount* Beispiel mit den beiden genannten Optionen unter Verwendung der uuid der Festplatte.

```
mount -o noatime,nobarrier /dev/disk/by-uuid/3d385c0f-2dc2-409a-92f8-e350e3fee7d1 /mnt/1_DS3
```

3.6 Thin-Client

Das Endgerät der Thin-Client Infrastructure ist ein Dell OptiPlex 7020, welcher für die Verarbeitung von anspruchsvollen Aufgaben ausreichend Hardware besitzt (Abbildung 15). Der Unterschied zur VDI und dem Zero-Client geht aus dem Abschnitt 2.4.2 eindeutig hervor. Die Praxistests bestätigten eine gute Leistung, die zur Bearbeitung von Programmen wie z.B. *GIS (Geoinformationssystem)* oder Grafikbearbeitungsprogramme wie *Adobe Illustrator* ausreichend sind.



Element	Wert
Hersteller	Dell
Modell	OptiPlex 7020
Prozessor	Intel(R) Core(TM) i3-4150 CPU @ 3.50Ghz, 2 Kerne, 4 logische Prozessoren
Bios	Dell Inc. A05, 06.05.2015
Arbeitsspeicher	8GB
Grafikkarte	Intel(R) HD Graphics 4400 1024MB
Netzwerkkarte	Intel(R) Ethernet Connection I217-LM
Display-Anschlüsse	2x Displayport / 1x VGA

Abbildung 15 Thin-Client Dell OptiPlex 7020 (Quelle: <http://www.dell.com/>)

Als Betriebssystem wird Windows 7 Professional, Windows 8.1 und Ubuntu Linux Desktop 15.10 für die Benutzer zur Verfügung gestellt. Die drei Betriebssysteme wurden einmalig installiert und als Master-Image auf dem iSCSI-Server zur Verfügung gestellt. Der Benutzer kann bei der Beantragung eine Auswahl des jeweiligen Betriebssystems treffen. Wer gerne auf ein Linux zugreifen möchte, aber auf ein Windows nicht verzichten kann, dem steht ein Dualboot beider Systeme zur

Verfügung. Die Master-Images der Windows Betriebssysteme werden nach dem ersten Booten automatisch individualisiert und in die Windows-Domäne *UNI-GREIFSWALD.DE* aufgenommen. Ubuntu Linux wird derzeit durch eine manuelle Konfiguration der Domain *UNI-GREIFSWALD.DE* hinzugefügt, aber es bestehen dadurch keine Einschränkungen bei dem Zugriff auf interne Ressourcen, wie z.B. die Home-Laufwerke.

3.7 Automatisierung

Zur Reduzierung des administrativen Aufwandes wurden wiederkehrende Tätigkeiten identifiziert und durch die Ausführung von Scripts zum Teil automatisiert. Bei der Automatisierung werden z.B. die Konfigurationseinträge für den iSCSI-Initiator und dem iSCSI-Target erstellt. Das Anlegen der Festplatten und das Anpassen des Bootloader sind weitere Teilschritte die dem Administrator die Arbeit beim Ausrollen eines Thin-Clients erleichtern.

In dem vorliegenden Abschnitt wird der aktuelle IST-Zustand der Automatisierung beschrieben, welcher im Rahmen dieser Bachelorthesis erarbeitet wurde. Die Komplexität bis zu einer vollständigen Automatisierung wäre eine eigenständige Arbeit. Ein Ausblick auf die zukünftigen Ziele der Automatisierung liefert das Fazit im *Kapitel 3.8*.

Es werden wiederkehrende Aufgaben identifiziert und Lösungen präsentiert, die eine Teilautomatisierung durch Linux-Scripts gewährleistet.

Das *BPMN (Business Process Modeling and Notation) [BPMN]* visualisiert den groben Geschäftsprozessablauf von der Beantragung bis zum Initialisierungsprozess eines Thin-Clients. Die in der grünen Farbe hinterlegten Tasks wurden als wiederkehrende Tätigkeiten innerhalb des gesamten Geschäftsprozesses identifiziert. Im Folgenden wird der Ablauf erläutert und das zugehörige Skript *autoSetupMain.sh* befindet sich im Anhang 8.2.

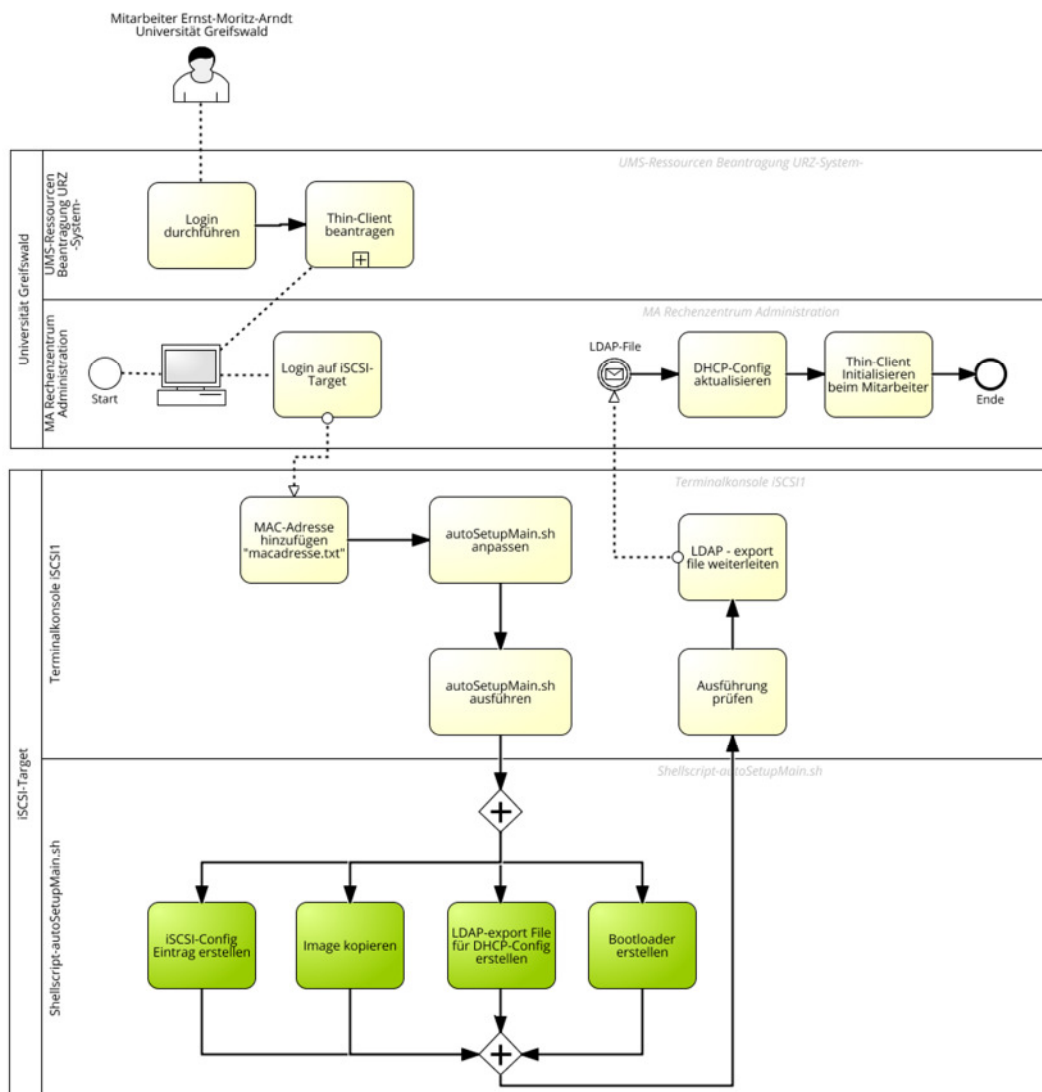


Abbildung 16 BPMN Automatisierung iSCSI-Target (Eigene Darstellung)

Der Prozess startet mit der Beantragung durch einen Mitarbeiter am UMS-System. Das UMS ist das Ressourcenportal der Ernst-Moritz-Arndt Universität Greifswald. Jeder Berechtigte kann über das Portal seine benötigten Ressourcen wie Hardware, IP-Adressen oder sonstiges beantragen. Die förmliche Beantragung erreicht einen zuständigen Mitarbeiter des Universitätsrechenzentrums, dieser leitet das Ausrollen des Thin-Clients in die Wege. Dafür loggt er oder der zuständige Administrator sich auf dem iSCSI-Server ein und trägt die ausgewählte MAC-Adresse eines freien Thin-Clients in die vorgesehene Datei *macadressen.txt* ein. Das Ausrollen mehrerer

Clients gleichzeitig ist problemlos auf demselben Wege möglich. Im Folgeschritt wird die *autoSetupMain.sh* auf die geforderten Einstellungen angepasst. Das Script ermöglicht eine individuelle Konfiguration, welche sich nach dem gewünschten Betriebssystem, welches VLAN für den Nutzer vorgesehen ist, und der zugehörige DHCP-Konfigurationseintrag richtet. Die manuelle Vorkonfiguration ist notwendig, damit die entsprechenden Parameter beim Ausführen der *autoSetupMain.sh* die spätere fehlerfreie Funktion gewährleisten. Nach der Ausführung wird durch den Administrator die Ausführung auf Vollständigkeit und Fehler geprüft, bevor der Initialisierungsprozess beim Mitarbeiter gestartet werden kann. Ein Teil der DHCP Konfiguration wird manuell an den zuständigen Mitarbeiter des Universitätsrechenzentrums weitergeleitet und am DHCP Server eingepflegt.

Beim Eintreten des Notfallszenarios müssen alle Dienste und Festplatten am iSCSI2 Server gestartet und angefügt werden, dafür befinden sich auf jedem Server Skripte, die diese Tätigkeiten nach dem Starten selbstständig ausführen.

Das Skript befindet sich im Anhang 8.3.

3.8 Fazit und Ausblick

Die Realisierungsphase der neuen Thin-Client Infrastructure wurde begleitet durch unterschiedliche konzeptionelle Schwerpunkte, wie z.B. dem Einsatz eines Cluster-Filesystems, welches durch ausgiebige Testverfahren leider die Erprobungsphase nicht überstanden hatte und somit keine Berücksichtigung in der aktiven Umgebung fand. Der Verzicht auf eine komplette Verschlüsselung des gesamten TCP/IP Traffic wurde bewusst unter der Voraussetzung der höheren Performance und somit einer größeren Akzeptanz der Benutzer implementiert.

Das Angebot eines Linux Betriebssystems fand in bestimmten Interessenskreisen große Nachfrage.

Mit dem Abschluss der Implementierung begann die Bewährungszeit im aktiven Einsatz des Rechenzentrums der Ernst-Moritz-Arndt Universität Greifswald. Insgesamt wurden 30 Thin-Clients bis zum Dezember 2015 ausgerollt. Die eingesetzten Thin-Clients lösten Zero-Clients an Arbeitsplätzen und einem

Schulungsraum ab, an denen bisher die Verarbeitung von grafisch anspruchsvollen Programmen nicht möglich war bzw. nur durch ein kostenintensives vGPU Sharing [vGPU Sharing] der VDI erbracht werden konnten.

Das zentrale Desktopmanagement gekoppelt mit der Automatisierung *Kapitel 3.7* am iSCSI-Target reduziert die administrativen Tätigkeiten und die Bereitstellungszeiten eines Desktop im Vergleich zum klassischen Computer erheblich.

Ohne an dieser Stelle detailliert die Kosten zu betrachten, wurde auf der Basis von iSCSI ein Konzept implementiert, welches es ermöglicht, auf vorhandene Hardware und Netzwerkressourcen zurückzugreifen und damit ein alternatives Konzept zur bereits vorhanden VDI zur Verfügung zu stellen.

Für den Zeitraum nach der Bachelorthesis würden der Ausbau der Automatisierung und deren Anpassung an die Geschäftsprozesse des Universitätsrechenzentrums eine zusätzliche Entlastung des administrativen Personals bedeuten. Vorstellbar wäre, dass durch die Beantragung eines Thin-Client im UMS Ressourcenportal das Ausrollen und Bereitstellen der Festplatten automatisch ohne jegliche manuelle Vorkonfiguration erfolgen könnte.

4 Virtual Desktop Infrastructure

In diesem Kapitel liegt der Schwerpunkt auf der Analyse und der technischen Erweiterung der bestehenden VDI am Rechenzentrum der Ernst-Moritz-Arndt Universität Greifswald. Es erfolgt die Aufnahme des IST-Zustandes mit einer anschließenden Implementierung eines VMware Sicherheitsservers für einen weltweiten Zugriff auf die virtuellen Desktops der Ernst-Moritz-Arndt Universität Greifswald. Das Rechenzentrum zieht in Erwägung, seine VDI konzeptionell und infrastrukturell so zu erweitern, dass die Vorzüge eines virtuellen Desktops auch außerhalb des internen Netzwerkes genutzt werden könnten. Die Implementierung des VMware Sicherheitsserver beinhaltet eine Erweiterung der vorhandenen VDI-Netzwerkstruktur durch die Installation und Konfiguration weiterer Komponenten.

4.1 VDI – technische IST-Analyse des URZ Greifswald

Im eingehenden Theorieteil wurde die VDI im Allgemeinen betrachtet, ohne dabei auf herstellerspezifische Angaben einzugehen. In der technischen Analyse wird genauestens auf das Produkt *VMware vSphere 5.5 und Horizon View 5.3.0* eingegangen, welches die zentralen Komponenten der VDI des Rechenzentrums der Ernst-Moritz-Arndt Universität Greifswald darstellen. Die Analyse ist Voraussetzung für das Verständnis, aber auch eine Grundvoraussetzung, um eine infrastrukturelle Erweiterung der aktiven VDI durchzuführen. Die Kompatibilität der Softwarekomponenten, Versionen und Lizenzen zueinander gleicht einem Informationsdschungel, den es zu durchdringen gilt.

Im letzten Teil der Analyse erfolgt eine Bewertung der möglichen Schwachstellen, welche durch Handlungsempfehlungen und deren praktischen Umsetzungen die VDI zukünftig noch zuverlässiger ausrichtet.

4.2 Verwendete Komponenten IST-Zustand

Die folgende Übersicht liefert eine Aufschlüsselung der gesamten VMware Komponenten, die zum Zeitpunkt der Analyse im Einsatz waren.

Tabelle 2 VMware Komponenten IST-Zustand

Komponente	Version	Verwendung als
VMware vSphere		
ESXi	5.5 (1)	Bare Metal Hypervisor
VMware vCenter		
VMware vCenter Server	5.5.0	Zentrales Management
VMware Single Sign On	5.5.0	Authentifizierungsdienst
VMware vCenter Inventory Service	5.5.0	Katalogspeicher vCenter-Instanzen
VMware vCenter Orchestrator	5.5.1	Automatisierung
VMware vCenter Java-Komponenten	5.5.0	Java
VMware vSphere Client	5.5.0	Verwaltungsclient ESXi-Host
VMware vSphere Update Manager	5.5.0	Aktualisierung ESXi-Hosts
VMware vSphere Web-Client	5.5.0	Web Verwaltung ESXi-Host
VMware View-Composer (optional)	5.3.0	Automatisch Linked-Clone
Connection-Broker		
VMware Horizon View	5.3.0	Bereitstellung von Desktops
Virtual Desktop		
VMware View-Agent	5.3.0	Kommunikation zwischen View-Verbindungsserver, vCenter und virtual Desktop
Zero-Client		
VMware Horizon View Client		Zugriff auf virtual Desktop

4.3 Aufbau und Netzwerkstruktur IST-Zustand VDI

Der IST-Netzwerkzustand der vorhandenen VDI wird schematisch mit allen zusammenhängenden Komponenten (Tabelle 2) in der folgenden Abbildung dargestellt.

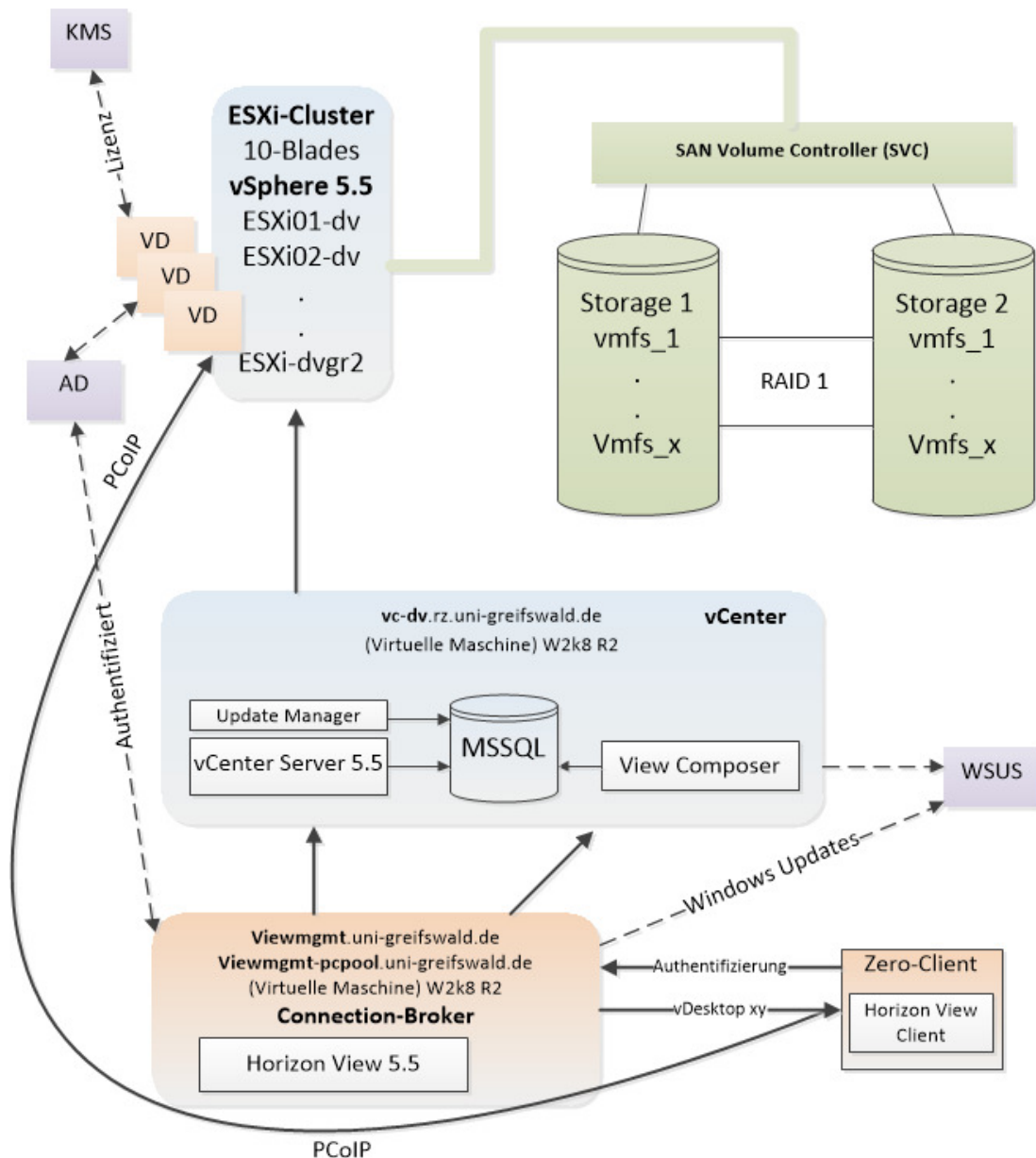


Abbildung 17 IST-Zustand Netzwerkinfrastruktur VDI (Eigene Darstellung)

Zum Aufbau einer Verbindung authentifiziert der Nutzer sich über ein Horizon View Client beim Connection-Broker, welcher die Authentifizierung des Nutzers mit dem

Active Directory abgleicht. Bei erfolgreicher Anmeldung übergibt der Connection-Broker dem Zero-Client alle für den Nutzer zur Verfügung stehenden Desktops zur Anzeige. Mit der Auswahl des gewünschten Desktops stellt der Zero-Client über das Übertragungsprotokoll PCoIP eine direkte Verbindung zum virtuellen Desktop her. Das vCenter [VMware vCenter] stellt eine zentrale Managementplattform der gesamten VMware-vSphere Umgebung inklusive aller darauf laufenden virtuellen Desktops dar. Der View-Composer ist verantwortlich für das Bereitstellen von Linked-Clones [VMware Linked Clones] und ist eine Komponente der Desktopvirtualisierung. Der Update Manager ist ein Plug-In auf dem vCenter, welches die VMware-vSphere Umgebung mit Patches versorgt. Alle drei Bestandteile, das vCenter, der View-Composer und der Update Manager greifen auf die Datenbanksoftware MSSQL zu. Innerhalb von MSSQL steht jeder Komponente eine eigene Datenbank für die Verwaltung ihrer Tätigkeiten zur Verfügung.

4.4 Schwachstellenanalyse

Das Ziel dieser Schwachstellenanalyse ist es, dass die IT-Umgebung erfasst wird und mögliche kritische Punkte aufgezeigt werden. Eine umfangreiche Schwachstellenanalyse beinhaltet neben einer technischen Bewertung auch eine bauliche Bewertung wie Serverräume, Kühlung, Stromversorgung, Zugangskontrollen und viele weitere Möglichkeiten, welche aber an dieser Stelle zu umfangreich werden würden. Die Analyse beschränkt sich auf die VDI und deren zugehörigen Komponenten. Dabei sind die eingesetzten Hardwareressourcen und die VDI Komponenten genauer zu analysieren.

Für die Desktopvirtualisierung steht ein eigenes *Dell M1000e Bladecenter* zur Verfügung, welches mit 8 Servern (*Blades*) + 2 Grafikserver ausgestattet ist. Die 8 Blades arbeiten im Verbund, sodass eine Redundanz innerhalb des Bladecenter gewährleistet werden kann.

Für den zukünftigen Ausbau der VDI wird empfohlen, ein zweites Bladecenter für die Desktopvirtualisierung zu implementieren und vorerst 4 der 8 Blades auf das

zweite Bladecenter zu migrieren. Mit der empfohlenen Redundanz könnte ein technischer Ausfall eines Bladecenters kompensiert werden.

Bei den VDI Komponenten ist während der Analyse aufgefallen, dass es zum heutigen Zeitpunkt eine neuere Version der gesamten *VMware* Umgebung gibt. Die Empfehlung geht zum Upgrade der Version 5.5 auf die Version 6, welches in naher Zukunft durchgeführt werden sollte. Die Gründe für ein Upgrade sind neben dem allgemeinen LifeCycle der Version 5.5, auch die neuen zusätzlichen Features, die in der Version 6 für eine bessere Skalierung von virtuellen Desktops zur Verfügung stehen. Zu beachten sind allerdings beim Upgrade der Hardware-Support [*VMware HW Kompatibilität*] die Installationsreihenfolge [*VMware Upgrade*] und das Upgrade der Lizenzierung.

Bei dem Ergebnis der Bewertungen der einzelnen VDI-Komponenten ist weiter aufgefallen, dass keine Redundanz für den View-Verbindungsserver (Connection-Broker) (Abbildung 17) zur Verfügung steht. Beim Ausfall des View-Verbindungsserver könnten sich keine Benutzer mehr an ihrem virtuellen Desktop anmelden. Der View-Verbindungsserver ist als eine virtuelle Maschine innerhalb der Servervirtualisierung gehostet. Es sollte in der SOLL-Struktur unbedingt ein zusätzlicher View-Verbindungsserver implementiert werden. Die Installation und Implementierung obliegt dem Rechenzentrum und ist nicht Bestandteil dieser Arbeit, wird aber zeitgleich bei der Realisierung der SOLL-Struktur durchgeführt.

Die Schwachstellenanalyse ermöglichte einen kritischen Blick auf die vorhandene VDI und führte zu Handlungsempfehlungen im Bereich der Hardwareredundanzen, dem Versionsupgrade auf *VMware* 6 und der Implementierung eines zusätzlichen View-Verbindungservers, welche dazu dienen, die bereits sehr gut ausgelegte Desktopvirtualisierung noch zuverlässiger darzustellen.

4.5 Aufbau und Netzwerkstruktur SOLL – Zustand

Das Bild zeigt die schematische Erweiterung des Netzwerkes, wie es in der SOLL-Struktur umgesetzt wurde.

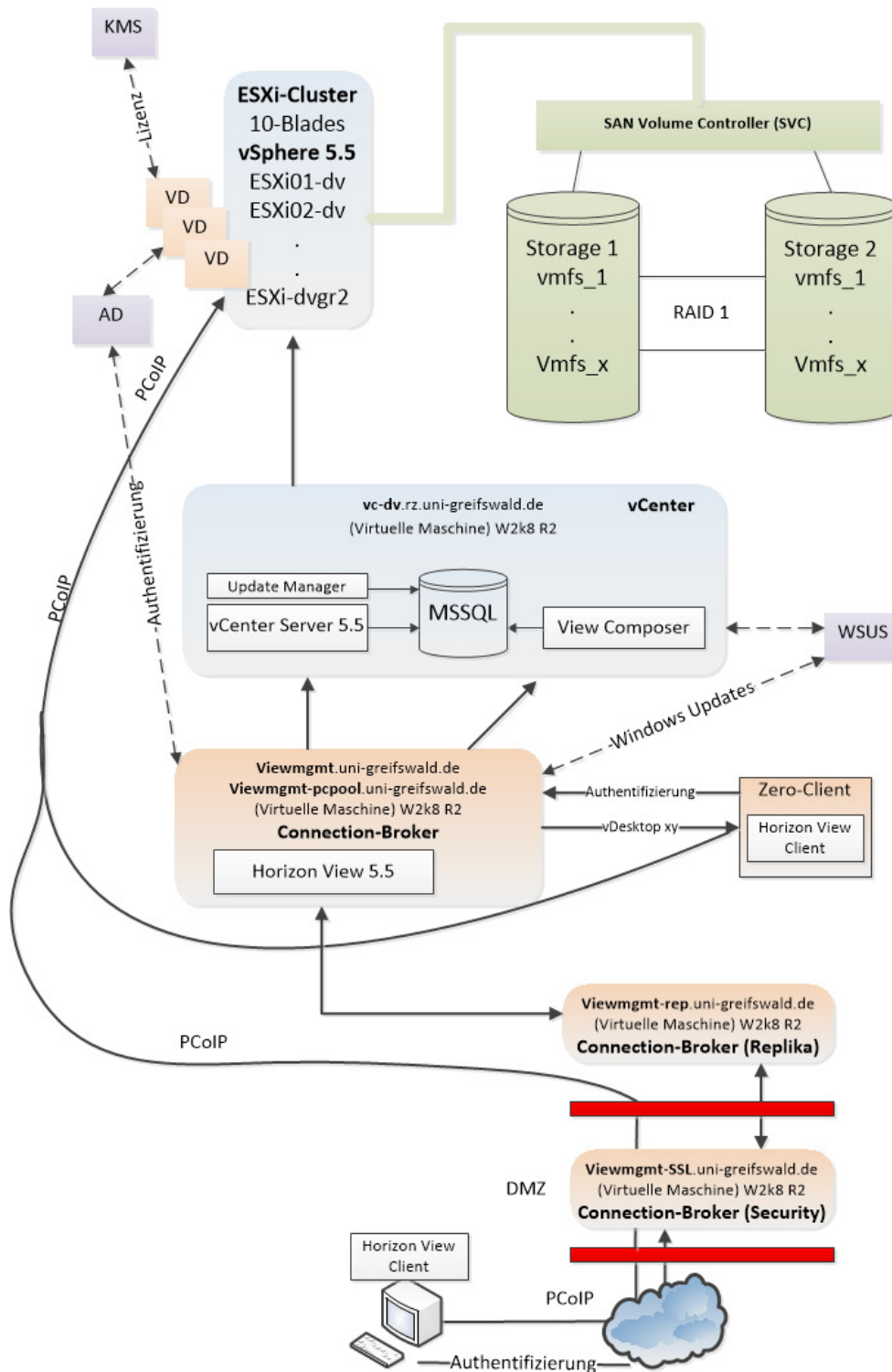


Abbildung 18 SOLL-Zustand VDI (Eigene Darstellung)

4.6 Implementierung VMware Sicherheitsserver

Während der Implementierungsphase geht es um die praktische Umsetzung eines Secure-Gateways für einen weltweiten Zugriff auf die virtuellen Desktops der Ernst-Moritz-Arndt Universität Greifswald (Abbildung 18). Es wird konzeptionell erläutert, welche Komponenten dafür benötigt werden, wo diese zu installieren sind und wie die Vorgehensweise bei der Durchführung ist.

Für die Implementierung des Secure-Gateways kommt nur die *DMZ* (Demilitarized Zone) des Universitätsrechenzentrums als Aufstellungsort in Frage. Eine Voraussetzung für die reibungslose Funktion eines Secure-Gateways ist die Konfiguration der DMZ-Firewall gemäß den Spezifikationen von VMware [*VMware Firewall*]. Da es sich bei dem künftigen Secure-Gateway um eine virtuelle Maschine handelt und die DMZ durch ein VLAN realisiert wurde, ist es physisch nicht notwendig, einen separaten Aufstellungsort zu wählen. Eine neue virtuelle Maschine mit entsprechender VLAN Konfiguration innerhalb der Servervirtualisierung ist dafür ausreichend ausgelegt. Eine weitere Voraussetzung, die auf dem bereits vorhandenen View-Verbindungsserver getroffen werden muss, ist die Vergabe eines Passwortes zur Aufnahme einer weiteren View-Serverinstanz, welches bei der Installation des Secure-Gateways angegeben werden muss.

Der Server benötigt nach der Installation des Windows Server Betriebssystems eine statische IP-Konfiguration, Namen und Domainregistrierung sowie die Installation eines Webbrowsers und eines aktuellen Flash-Players für die Administration über den Web-Client. Sind alle Vorbereitungen und Voraussetzungen getroffen, wird auf dem Server ***viewmgmt-ssl.uni-greifswald.de*** eine weitere VMware Horizon View Instanz installiert. Die Installationsroutine ist trivial und beinhaltet lediglich die Auswahl der Instanz *View Security Server*, dem *FQDN (Fully Qualified Domain Name)* des internen View-Verbindungservers und die automatische Konfiguration der lokalen Windows-Firewall. Unter der Angabe des Passwortes erfolgt die Aufnahme des Sicherheitsservers in die vorhandene Horizon-View Umgebung.

Mit Abschluss der Installation findet sich auf dem bereits vorhanden internen View-Verbindungsserver im Konfigurationsmenü der neue Sicherheitsserver wieder. Die

Aktivierung des Secure-Gateways erfolgt durch das Setzen der folgenden Option auf dem internen View-Verbindungsserver.

PCoIP Secure Gateway

☒ PCoIP Secure Gateway für PCoIP-Verbindungen zum Computer verwenden

PCoIP - Externe URL: Beispiel: 10.0.0.1:4172 ?

Abbildung 19 Option Secure Gateway (Screenshot)

Damit wird jeder Client, egal ob intern oder extern, angehalten, die Verbindung über das Secure-Gateway zu wählen. Diese Konstellation, dass sich alle internen Clients auch über das Secure-Gateway verbinden, entspricht nicht den Vorstellungen eines gewünschten Verbindungsablaufs innerhalb der VDI. Zur Lösung des Problems muss eine weitere View-Verbindungsinstanz (**Replikation**) geschaffen werden, die es ermöglicht, die internen von den externen Zugriffen zu trennen. Somit ist ein weiterer View-Verbindungsserver **viewmgmt-rep.uni-greifswald.de** notwendig. Die Vorbereitungen der virtuellen Maschine werden adäquat zum Secure-Gateway durchgeführt. Eine DMZ-Konfiguration ist für den weiteren View-Verbindungsserver nicht notwendig, weil dieser im internen Netzwerk zur Verfügung steht.

Mit dem Abschluss der Implementierung begann die Testphase. Getestet wurden die Zugriffe über den Sicherheitsserver. Dabei sind die fehlerfreie Funktion sowie die Performance und die Latenzzeiten entscheidende Faktoren. Der Erfolg für das umgesetzte Projekt steht oder fällt mit der Akzeptanz der Anwender. In der Testphase konnte durch unterschiedliche Probanden festgestellt werden, dass ein flüssiges Arbeiten möglich ist, aber es verstärkt auf die Qualität der Internetleitung ankommt. Probanden mit geringerer Bandbreite aus ländlichen Gegenden konnten nur ein mangelndes Arbeiten feststellen, die Tester mit einer Bandbreite von mindestens 3 Mbit/s aufwärts konnten dagegen komfortabel mit ihrem Desktop arbeiten. Bei einem direkten Leistungsvergleich gegenüber internen und externen Zugriffen gibt es sicher Einschnitte der externen Zugriffe, die es aber für bestimmte

Situationen erträglich machen, z.B. wenn eine persönliche Anwesenheit am Arbeitsplatz nicht möglich ist, aber trotzdem der betriebliche Desktop in Anspruch genommen werden möchte.

4.7 Zusammenfassung VDI

Das Kapitel der Virtual Desktop Infrastructure stand ganz im Fokus der Analyse und der Weiterentwicklung durch die Implementierung eines Secure-Gateways, welches den Mitarbeitern und Administratoren der Ernst-Moritz-Arndt Universität Greifswald einen entfernten Zugriff auf ihren betrieblichen Desktop ermöglicht. Die Implementierung ist erfolgreich verlaufen. Durch die Implementierung des Secure-Gateway ist zusätzlich ein sehr positiver Nebeneffekt entstanden: Die Einwahl über das Secure-Gateway erfolgt ohne einen vorherigen VPN Verbindungsaufbau in das Netzwerk der Ernst-Moritz-Arndt Universität Greifswald. Dies stellte für nicht versierte IT-Mitarbeiter bisher eine zusätzliche Hürde dar.

Mit der Implementierung kommen ganz neue Ansätze wie *BYOD* (Bring Your Own Device) zum Tragen. BYOD geht in die Richtung, dass die Mitarbeiter ihre privaten Endgeräte für das Arbeiten verwenden könnten. Es erfolgt eine Verschmelzung zwischen dem privaten und dem dienstlichen Nutzen, welches vom IT-Management des Universitätsrechenzentrums in mehrfacher Hinsicht in Zukunft berücksichtigt werden muss.

Wer das Arbeiten auf einem Smartphone oder Tablet-PC bevorzugt, kann dieses auf den meisten herkömmlichen Betriebssystemen, wie z.B. Android, IOS, MAC-OS, Linux oder Windows über das Secure-Gateway gerne in Anspruch nehmen. Voraussetzung ist immer der Horizon View Client, welcher für viele Geräte zur Verfügung steht.

Zur abschließenden Bewertung passt die kurze Zusammenfassung, dass mit der Implementierung der Komfort und die Flexibilität für das Arbeiten erweitert wurden.

5 Kostenanalyse

In diesem Kapitel werden die vorgestellten Konzepte in zukünftige Einsatzszenarien kategorisiert und den Anforderungsprofilen der Ernst-Moritz-Arndt Universität Greifswald zugeordnet. Es erfolgt die Benennung und Beschreibung der Anforderungsprofile. Die Eingliederung dient als Grundlage für IT-Managemententscheidungen, wann welches Konzept eingesetzt wird.

5.1 Kostengegenüberstellung

Die Kosten für eine VDI innerhalb eines Rechenzentrums unterscheiden sich im Wesentlichen von anderen IT-Infrastrukturen kleinerer Unternehmen. In einem Rechenzentrum gibt es gewisse Voraussetzungen, die bereits vorhanden sind, wie z.B. gekühlte Serverräume, Hochleistungsserver, performante Anbindungen an Storage-Netzwerke, eine abgesicherte Stromversorgung und die Kommunikation über redundante Netzwerkkomponenten. All diese Verfügbarkeiten lassen sich auch auf die Betriebskosten einer VDI umlegen. In diesem Abschnitt geht es nicht um die Identifizierung, wie viel Kosten die VDI umgerechnet auf die gesamte IT-Infrastruktur verursacht, sondern vielmehr darum, wo wesentliche Unterschiede bei den Kostenindikatoren im Vergleich zur neuen Thin-Client Infrastructure bestehen. Gemeinsame Kosten der beiden Konzepte lassen sich identifizieren und im Wesentlichen für die Betrachtung dadurch neutralisieren. Gemeinsame Kosten bringen keinen Unterschied und werden somit nicht für die Berechnung herangezogen, wie z.B. das Storage Area Network.

Somit bleibt an dieser Stelle der Hinweis, dass die tatsächlichen Kosten pro Client dadurch höher ausfallen können als Sie im weiteren Verlauf dargestellt werden.

5.1.1 Investitionskosten

In diesem Abschnitt werden die Anschaffungskosten und die laufenden Kosten aller benötigten Komponenten für die VDI und der Thin-Client Infrastructure identifiziert. Dieses betrifft explizit die verwendete Hardware, Software und alle dazugehörigen Lizenzen. Da es sich vermehrt um Kosten für die zentrale Bereitstellung von Desktops handelt, wird die Gesamtsumme auf die Anzahl der verwendeten Desktops umgelegt.

Die folgende Tabelle identifiziert die Investitionskosten für die VDI und die Thin-Client Infrastructure. Die dargestellten Kosten sind aus Beschaffungsaufträgen entnommen, die das Rechenzentrum der Ernst-Moritz-Arndt Universität Greifswald in der Vergangenheit getätigt hat. Der Start mit der Desktopvirtualisierung erfolgte im Jahre 2012. Eingehend wurde erwähnt, dass die Desktopvirtualisierung eine konsequente Weiterentwicklung der Servervirtualisierung ist. So ist es auch innerhalb des Rechenzentrums, dass die Desktopvirtualisierung historisch aus der Servervirtualisierung entstanden ist. Historisch meint, dass z.B. übrige Lizenzen aus der Servervirtualisierung auf die Desktopvirtualisierung angewandt wurden. Die Desktopvirtualisierung benötigt die gleichen Lizenzen wie die Servervirtualisierung, z.B. das VMware vSphere Enterprise Paket und das vCenter Server Standard. Zusätzlich für die Desktopvirtualisierung werden VMware View Lizenzen pro eingesetzten Zero-Client angeschafft. Ein Teil der Lizenzkosten entfällt auch an die Firma Microsoft, welche es ermöglicht, durch die Lizenzierung VDA (*Microsoft Virtual Desktop Access*) auf ein Microsoft Betriebssystem zuzugreifen, welches sich in einer Virtualisierungsumgebung befindet. Dafür entfallen die bekannten Windows Betriebssystem-Lizenzen, die z.B. für jeden Thin-Client beschafft werden müssen.

Tabelle 3 Kostengegenüberstellung VDI / Thin-Client Infrastructure

	Virtual Desktop Infrastructure	In EUR (netto)	Thin-Client Infrastructure	In EUR (netto)
Einmalig Server	Bladecenter Desktopvirtualisierung 8 Blades + 2x Server (nVidia grid K2) Dell BC M1000e	83.721		
	VMware vSphere Enterprise ESX-Host – Lizenzen (1153€ / CPU) 10 Blades (20 CPU's)	23.050		
	Bladecenter Servervirtualisierung anteilig 4 VM (Server vCenter, 3x View) Bladecenter SV inkl. Hardware u. Lizenzen 103.734€ / 325 virtuelle Server = 319€ p. VM (stand 12/2015 70% Auslastung)	1.276	Bladecenter Servervirtualisierung anteilig 2 VM (Server iSCSI1 u. iSCSI2)	638
	Softwarelizenzen: vCenter Server Standard	2.180	Softwarelizenzen: iSCSI Enterprise Target GNU General Public License Version 2	0
	Betriebssystemlizenzen: 4x Windows Server 2008 Enterprise R2	1.200	Betriebssystemlizenzen: Debian 8.0 GNU General Public License Version 2	0
	Einmalige Investitionskosten	111.427		638
	* (exkl. IT-Infrastruktur Universitätsrechenzentrum)			
Jährlich Server	VMware vSphere Enterprise Support and Subscription (SnS) Lizenz 10 Blades x 300€	3.000		
	vCenter Server Standard Lizenz	522		
	Jährliche Investitionskosten	3.522		0
	Insgesamt Investition + 1 Jahr Betrieb	114.949		638
Einmalig Client	Zero-Client EVGA PD06	256	Thin-Client Dell OptiPlex 7020	393
	VMware View Addon Lizenz (Horizon View / View Composer) (1 Jahr, danach 57€ jedes weitere)	145	Betriebssystemlizenzen: Windows 7 Professional	127
	Einmalige Investitionskosten Client	434		520
Jährliche Client	VMware View Lizenz (Maintenance) ab dem 2 Jahr	57		
	Microsoft Windows Virtual Desktop Access (VDA)	33		
	Energiebedarf 7 Watt 250 Arbeitstage entspricht 41,5 Kw/h 0,25€ x Kw/h	10	Energiebedarf 17 Watt 250 Arbeitstage entspricht 102 Kw/h 0,25€ x Kw/h	26
	Jährliche Investitionskosten pro Client	100		26
	Insgesamt Investition + 1 Jahr Betrieb	534		546

Die Kostengegenüberstellung zeigt eine sehr große Differenz bei den Anschaffungskosten der VDI im Vergleich zur Thin-Client Infrastructure. Die Zahlen sind verschieden interpretierbar, zum Einen trübt der erste Blick, dass die Desktopvirtualisierung sehr teuer im Verhältnis zur Thin-Client Infrastructure ist, zum Anderen schrecken die hohen Lizenzkosten ab. Die Anschaffung und die Investitionskosten der VDI erfolgten im Jahr 2012, somit unterliegen die Server, Clients und Lizenzkosten bereits einer jährlichen Abschreibung, welche sich gewinnmindernd auswirkt. Die VDI hat einen positiven Skalierungseffekt: Je mehr virtuelle Desktops ausgerollt werden, umso geringer werden die Kosten für jeden einzelnen. Dieser Spareffekt ist bei den Thin-Clients aufgrund der geringen Anfangsinvestition nicht so stark spürbar. Folgende Tabelle stellt diesen Skalierungseffekt exemplarisch dar. Die verwendete Formel zur Berechnung der Kosten pro Client lautet

$$K.p.Client = \frac{(Clientanzahl * Client Invest.) + Server Invest}{Clientanzahl}$$

Tabelle 4 Kosten pro Client

Virtual Desktop Infrastructure	In EUR (netto)	Thin-Client Infrastructure	In EUR (netto)
Server Investition + 1 Jahr Betrieb	114.949	Server Investition + 1 Jahr Betrieb	638
Zero-Client Client Investition + 1 Jahr Betrieb	534	Thin-Client Client Investition + 1 Jahr Betrieb	543
Kosten pro Client		Kosten pro Client	
1x Zero Client	115.483	1x Thin Client	1.181
100x Zero Clients	1.683,49	25x Thin Clients	568,52
160x Zero Clients	1.252,43	100x Thin Clients	549,38
300x Zero Clients	917,16	300x Thin Clients	545,13

Durch den anfänglich hohen Investitionsaufwand bei der VDI wirkte die gesamte Infrastruktur im Vergleich zur Thin-Client Infrastructure überteuert. Der heutige Stand von 175 eingesetzten virtuellen Desktops innerhalb der VDI mit einer durchschnittlichen Serverauslastung von 20% gewährleistet genügend Reserven, um

die Anzahl der virtuellen Desktops auf 300 Stück auszuweiten, vorausgesetzt ist eine ausgewogene Belastung der 300 virtuellen Maschinen. Das Bladecenter ist derzeit mit 10 von möglichen 16 Blades belegt, jedes weitere Blade könnte für eine zusätzliche Investition von 7000€ inkl. Lizenzen 30 weitere virtuelle Desktops bereitstellen.

Bei der Thin-Client Infrastructure ist der Spareffekt nicht entscheidend, was aber auch für das Konzept und seine Einsatzmöglichkeiten spricht. Der niedrige Investitionsaufwand ermöglicht die Implementierung von geringen Stückzahlen, wo die VDI mit ihren technischen Möglichkeiten am Ende ist.

5.1.2 Wartungs- und Personalaufwand

Der Einsatz einer Desktopvirtualisierung ermöglichte bereits Einsparungen beim Personalaufwand. Durch die VDI sind administrative Aufgaben bei der Bereitstellung von Desktops oder beim Patchmanagement weniger zeitintensiv, als im Vergleich zur klassischen Server / Client Architektur.

Durch die Implementierung der Thin-Client Infrastructure sind neue Tätigkeiten für das Personal hinzugekommen. Die zusätzlichen Aufgaben sind für das administrative Personal der Thin-Client Infrastructure zwar neu, lassen sich aber mit den herkömmlichen Aufgaben einer VDI vergleichen. Die Hauptaufgaben für beide Konzepte sind:

- Bereitstellung von Desktops
- Wartung der zentralen Masterimages
- Organisation des Desktopmanagement
- Pflege und Wartung der Infrastruktur
- Monitoring und Fehlerbehebung
- Neuinstallation und Upgrade

Zur Ermittlung des tatsächlichen Aufwandes wurde ein Fragebogen *Anhang 8.4* mit den wesentlichen Aufgaben definiert, welcher durch das administrative Personal zeitlich bewertet wurde. Aus den Fragebögen lässt sich feststellen, dass es keinen gravierenden Unterschied bei der Erfüllung der grundlegenden Aufgaben gibt.

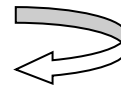
5.2 Anforderungsprofile

Es erfolgt eine Kurzbeschreibung der zwei möglichen Anforderungsprofile, die für den Einsatz innerhalb der VDI oder der Thin-Client Infrastructure in Frage kommen. Die Einstufung der Mitarbeiter in Anforderungsprofile ermöglicht die Zuweisung eines passenden Endgerätes für die Bearbeitung ihrer Aufgaben.

5.2.1 Zero-User

Der Zero-User entspricht einem Mitarbeiter mit herkömmlichen Verwaltungsaufgaben, wie z.B. das Bearbeiten von Office Dokumenten, elektronische Kommunikation, Fernwartungen, Abrechnungen oder das Vorbereiten von Präsentationen.

Virtuell Desktop Infrastructure



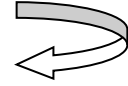
Der Zero-User passt in das Profil der VDI und würde einen virtuellen Desktop bekommen. Für die Bearbeitung steht dem Zero-User ein virtueller Desktop mit dem Betriebssystem Windows 7 oder Windows 8.1 zur Verfügung, welcher bei Gelegenheit auch von außerhalb des betrieblichen Umfeldes aufgerufen werden könnte. Die Stärken und Schwächen der VDI sind:

- + Home Office über Security-Gateway
- + Hardware unabhängig
- Performance
- Nur Windows

5.2.2 Thin-User

Der Thin-User beinhaltet die gleichen Anforderungen wie ein Zero-User, aber zusätzlich benötigt er die Verarbeitung performanter Anwendungen. In den meisten Fällen handelt es sich dabei um das Verarbeiten von 3D Anwendungen.

Thin-Client Infrastructure



Dem Thin-User steht es frei, auf welchem Betriebssystem (Windows oder Linux) er seine täglichen Aufgaben am Arbeitsplatz durchführen möchte. Die Stärken und Schwächen der Thin-Client Infrastructure sind:

- + Performance bei einfacher 3D Verarbeitung
- + Dualboot (Windows & Linux)
- Hardware gebunden
- Kein Zugriff über das Security-Gateway

Die Gegenüberstellung der Anforderungsprofile ermöglicht eine Einschätzung ggf. unter Berücksichtigung und Einfluss des Mitarbeiters, welches Endgerät den Zweck zur Bewerkstellung der Tätigkeiten am geeignetsten erscheint.

6 Fazit

Am Anfang der Bachelorarbeit standen zwei wesentliche Aspekte im Raum. Zum Einen war es das Ziel, die VDI durch die Implementierung eines Security-Gateways konzeptionell und infrastrukturell zu erweitern und zum Zweiten stand die Einführung einer Thin-Client Infrastructure auf der Agenda.

Das Rechenzentrum der Ernst-Moritz-Arndt Universität Greifswald begann im Jahre 2012 mit der Einführung der Desktopvirtualisierung. Bis zum heutigen Stand sind 175 virtuelle Desktops im aktiven Einsatz. Die vorliegende Arbeit erfüllte die Maßgabe, die vorhandene Desktopvirtualisierung konzeptionell und infrastrukturell so zu erweitern, dass sämtliche virtuelle Desktops über das Internet erreichbar sind. Dadurch wurden neue Möglichkeiten geschaffen, wie z.B. das zeit- und ortsunabhängige Zugreifen auf das betriebliche Umfeld. Die Implementierung eines Security-Gateway revolutioniert nicht die gesamten Arbeitsabläufe und -zeiten der Mitarbeiter der Universität Greifswald, aber es ermöglicht die Verfügbarkeit des betrieblichen Umfeldes durch einen virtuellen Desktop, zu der Zeit und an dem Ort, an dem er benötigt wird.

Den praktisch größeren Anteil nahm die Implementierung der Thin-Client Infrastructure mit der Erarbeitung einer geeigneten Lösung zur Deckung des Bedarfs an die erhöhten Anforderungen von anspruchsvollen Anwendungen ein. Dabei lag der Fokus auf dem zentralen Desktopmanagement basierend auf dem iSCSI Protokoll. Die Testphase von 2 Monaten verlief so positiv, dass im nächsten Jahr die Weiterentwicklung der Thin-Client Infrastructure fortgesetzt wird.

Ein Thin-Client besitzt alle Vorteile, die ein herkömmlicher PC auch ermöglicht, zusätzlich aber zentral durch das administrative Personal verwaltet werden kann.

Das Investitionsverhältnis der Thin-Client Infrastructure im Vergleich zur VDI ermöglicht einen flächendeckenden Einsatz, aber es ermöglicht eben auch den Einsatz weniger Thin-Clients an Stellen mit speziellen Anforderungen.

Das IT-Management des Rechenzentrums der Ernst-Moritz-Arndt Greifswald besitzt mit Abschluss dieser Bachelorarbeit die Wahl, alle Nutzer der Ernst-Moritz-Arndt

Universität Greifswald anhand von Anforderungsprofilen zu kategorisieren und daraufhin die Entscheidung zu fällen, welches Endgerät zum Einsatz kommt.

Es wurde ein weiterer Schritt absolviert weg vom herkömmlichen PC, hin zum zentralisierten Desktop.

7 Literatur- und Quellenverzeichnis

ARP-Spoofing

<http://www.elektronik-kompodium.de/sites/net/1910171.htm>, 11.12.2015

BPMN

<http://www.bpmn.org/>, 12.12.2015

BSI

url:

https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/_content/g/g04/g04035.html, 06.01.2016

Bundesnetzagentur

url: http://www.bundesnetzagentur.de/cln_1911/DE/Service-Funktionen/QualifizierteelektronischeSignatur/WelcheAufgabenhatdieBundesnetzagentur/GeeigneteAlgorithmenfestlegen/geeignetealgorithmenfestlegen-node.html, Algorithmenkatalog 2015

CHAP RFC 1994

W. Simpson. PPP Challenge Handshake Authentication Protocol (CHAP). RFC 1994, August 1996.

DRBD

url: <http://drbd.linbit.com/>, 27.10.2015

Fraunhofer (2011, S.8)

Fraunhofer-Institute für Umwelt-, Sicherheit- und Energietechnik UMSICHT: Thin Clients 2011 - Ökologische und ökonomische Aspekte virtueller Desktops [Online], http://publica.fraunhofer.de/eprints/urn_nbn_de_0011-n-1537716.pdf, 17.09.2015

GlusterFS

<http://www.gluster.org/>, 11.12.2015

IET

url:<http://iscsitarget.sourceforge.net/>, 28.10.2015

IO-Scheduler

<https://wiki.ubuntuusers.de/SSD/Scheduler>, 13.12.2015

iSCSI RFC 7143

M. Chadalapaka. Internet Small Computer System Interface (iSCSI). RFC 7143, April 2014

Lampe (S.78)

Lampe, Frank: Green-it, Virtualisierung und Thin Clients, 1. Auflage,
Vieweg+Teubner Verlag | Springer Fachmedien Wiesbaden GmbH, 2010

Laudon(S.239-243)

Laudon K., Laudon J.P., Schoder D.: Wirtschaftsinformatik eine Einführung, 2.
aktualisierte Auflage, Pearson Studium, 2010.

MD5 RFC1321

R. Rivest. The MD5 Message-Digest Algorithm (MD5). RFC1321, April 1992

SYSPREP

[https://technet.microsoft.com/de-de/library/cc721940\(v=ws.10\).aspx](https://technet.microsoft.com/de-de/library/cc721940(v=ws.10).aspx), 11.12.2015

Teradici

<http://www.teradici.com/pcoip-technology>, 17.12.2015

vGPU Sharing

<https://www.vmware.com/products/vsphere/features/vGPU>, 14.12.2015

VMware (Dokumentationscenter)

url: <https://pubs.vmware.com/view-52/index.jsp#com.vmware.view.planning.doc/GUID-57D362EB-AC04-45B8-87AA-05A15A998211.html>, 01.12.2015

VMware (Firewall)

url: <https://pubs.vmware.com/view-52/index.jsp#com.vmware.view.planning.doc/GUID-B8D3225D-0CB2-42D3-B2B8-EB7DED0F3B5E.html>, 30.11.2015

VMware HW Kompatibilität

<https://www.vmware.com/resources/compatibility/search.php>, 14.12.2015

VMware Linked Clones

https://www.vmware.com/support/ws55/doc/ws_clone_overview.html,
14.12.2015

VMware (Multi-writer)

url: http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1034165, 28.10.2015

VMware Upgrade

<https://pubs.vmware.com/vsphere-60/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-60-upgrade-guide.pdf>, 17.12.2015

VMware vCenter

<https://www.vmware.com/de/products/vcenter-server>, 17.12.2015

Vogel, Kocoglu, Berger(2010, S.7)

R. Vogel, T.Kocoglu, T. Berger: Desktopvirtualisierung, 1. Auflage,
Vieweg+Teubner Verlag | Springer Fachmedien Wiesbaden GmbH, 2010.

Weiping, Wandong (2005, S.457-462)

Weiping Liu, Wandong Cai: Computational Intelligence and Security, Part II,
LNAI 3802, S.457-462, 2005.

Wolf (2006, Kap.7.12)

Wolf, Jürgen: Linux-UNIX-Programmierung[Online],
http://openbook.rheinwerk-verlag.de/linux_unix_programmierung/Kap07-011.htm#RxxKap07011040002021F048100, 18.09.2015

xfs

http://xfs.org/index.php/XFS_FAQ, 13.12.2015

XTREEMFS

url: <http://www.xtreemfs.org/>, 27.10.2015

8 Anhang

8.1 Installationsleitfaden iSCSI-Server

Die folgenden Installationsschritte beruhen auf einem Debian GNU/Linux 8 und sollten möglichst auf beiden iSCSI-Targets (iSCSI1 & iSCSI2) identisch durchgeführt werden, ansonsten könnte ein unterschiedlicher Konfigurationsstand auf beiden Servern zustande kommen.

Installation der Hauptpakete

```
aptitude -y install iscsitarget iscsitarget-dkms
```

Nachladen benötigter Kernel-Module

```
aptitude install module-assistant debhelper
```

Aktualisieren der Kernel-Module

```
m-a a-i iscsitarget
```

Starten des iSCSI-Targets

```
/etc/init.d/iscsitarget start  
[ ok ] Starting iSCSI enterprise target service:.. ok
```

8.2 Automatisierungsscript

```
#!/bin/bash
# AutoSetupMain.sh
# Christian Schuldt
# Stand 15.11.2015
#
# Macadressen einlesen
typeset -i i=0
while read array[$i]
do
    i=i+1
done < macadressen.txt
# Passwortdatei einlesen
typeset -i i=0
while read arraypassword[$i]
do
    i=i+1
done < JoeDoe.txt
# initialisierung
# Speicherort Masterimage
sysprepPath="/mnt/5_BA1/images/windows7sysprep2_FG_2015_11_19"
prefix="w7_"
#vlan
vlan="141.5x.x.x"

#Festplatten Kapazitaet ermitteln uns speichern in einer Variable
#sde1
df_1_DS3=$(df -h /dev/disk/by-uuid/3d385c0f-2dc2-409a-92f8-e350e3fee7d1 | grep ^/ | awk '{print $4}' | tr -d 'G')
#sdf1
df_2_DS3=$(df -h /dev/disk/by-uuid/69c7fc35-a0dc-4e7b-bc42-6e9c0081be11 | grep ^/ | awk '{print $4}' | tr -d 'G')
#sdg1
df_3_DS4=$(df -h /dev/disk/by-uuid/b3e556de-13c3-4d80-9b07-b155a738361e | grep ^/ | awk '{print $4}' | tr -d 'G')
#sdh1
df_4_DS4=$(df -h /dev/disk/by-uuid/c6643b75-283b-46a4-82c7-93dfa1698c63 | grep ^/ | awk '{print $4}' | tr -d 'G')

# Images kopieren inkl. Kontrolle der Speicherkapazitäten
for ((i=0; i<${#array[@]-1};i++))
do
    # i= %(Modulo) 'Anzahl der Platten' ergibt gleich Rest 0-3 gleichmaeßige Verteilung ueber alle Platten
    # dynamisch Erweiterbar: bei 5 Platten i= % 5 ist gleich Rest 0-4, ergänzen einer elif Bedingung
    # Kopiert kein image auf die Platte, wenn nicht mehr wie 100GB frei sind.
    mod=$((i % 4))
    if [ $mod -eq 1 ] && [ $df_1_DS3 -gt 100 ]
    then
```

```

        targetpath="/mnt/1_DS3/images"
        cp $sysprepPath $targetpath/$prefix${array[i]}.img

    elif [ $mod -eq 2 ] && [ $df_2_DS3 -gt 100 ]
    then
        targetpath="/mnt/2_DS3/images"
        cp $sysprepPath $targetpath/$prefix${array[i]}.img

    elif [ $mod -eq 3 ]&& [ $df_3_DS4 -gt 100 ]
    then
        targetpath="/mnt/3_DS4/images"
        cp $sysprepPath $targetpath/$prefix${array[i]}.img

    elif [ $mod -eq 0 ] && [ $df_4_DS4 -gt 100 ]
    then
        targetpath="/mnt/4_DS4/images"
        cp $sysprepPath $targetpath/$prefix${array[i]}.img

    else
        echo "Fehlerausgabe: ${array[i]} wurde nicht kopiert"
        targetpath=""
    fi

# ietd.conf Eintrag erstellen
iqn="Target iqn.$vlan:w7_${array[i]}.img
    Lun 0 Path=$targetpath/$prefix${array[i]}.img,Type=fileio
    MaxRecvDataSegmentLength      8192
    IncomingUser user ${arraypassword[i]}
    MaxXmitDataSegmentLength      8192
    NOPInterval                    15
    NOTimeout                      6000000 "

    echo -e $iqn >> /etc/iet/ietd.conf
    echo ${array[i]} user ${arraypassword[i]} >> MACplusJoeDoe.txt

#ldif import-Datei erstellen (DHCP-Server Konfig iSCSI-initiator)
LDAP="dn: cn=${array[i]},cn=iSCSI-
neu,ou=Groups,cn=1xx.xx.xx.x,ou=HS,ou=Subnets,cn=Settings,dc=DHCP,dc=Konfigurationen,dc=TLD\nobjectClass:
dhcpHost\nobjectClass: top\ncn: ${array[i]}\ndhcpHWAddress: ethernet ${array[i]}\ndhcpStatements: if exists user-class and
option user-class = \"gPXE\"{filename \"\"; option root-path \"iscsi:$vlan::::iqn.$vlan:w7_${array[i]}.img\";option
gpxe.username \"user\";option gpxe.password \"${arraypassword[i]}\"; } else {filename \"pxelinux.0\";}\n"
echo -e $LDAP >> LDAP_Eintrag.txt
done

```

8.3 iSCSI-Start Script

```
#!/bin/bash
# Christian Schuldt
# Stand 12.12.2015

# VLAN 38 aktivieren
ifup eth1

# VLAN 6 aktivieren
ifup eth2

# VLAN 32 aktivieren
ifup eth3

# VLAN 11 aktivieren
ifup eth4

# VLAN 36 aktivieren
ifup eth5

sleep 1

#sde1
mount -o noatime,nobarrier /dev/disk/by-uuid/3d385c0f-2dc2-409a-92f8-e350e3fee7d1 /mnt/1_DS3
#sdf1
mount -o noatime,nobarrier /dev/disk/by-uuid/69c7fc35-a0dc-4e7b-bc42-6e9c0081be11 /mnt/2_DS3
#sdg1
mount -o noatime,nobarrier /dev/disk/by-uuid/b3e556de-13c3-4d80-9b07-b155a738361e /mnt/3_DS4
#sdh1
mount -o noatime,nobarrier /dev/disk/by-uuid/c6643b75-283b-46a4-82c7-93dfa1698c63 /mnt/4_DS4
#backup Plate 512GB
mount -o noatime,nobarrier /dev/disk/by-uuid/f81f2abf-72f9-4726-bd1b-39406bebbe3b /mnt/5_BA1/

# Dienste Starten
/etc/init.d/iscsitarget start
/etc/init.d/tftpd-hpa start
/etc/init.d/tivoli start
```

8.4 Fragebogen Wartungs- und Personalaufwand

VDI (Virtual Desktop Infrastructure)

Tätigkeit	Aufwand in Personenstunden
Installation und Einrichtung der gesamten VDI Umgebung (geschultes Personal)	min. 2 Arbeitstage von 8 h
Erstellen eines Masterimages	3 h
Wartung des Masterimages in h / Monat	2 h
Bereitstellung eines virtuellen Desktops (vollwertige Maschine)	0,5 h
Aufbau und Inbetriebnahme eines Zero-Clients	Ohne Anreise 15 min
Administration am Arbeitsplatz des Nutzers h / Monat	keine
Monitoring und Fehlerbehebung in h / Tag	2 h
Organisation Desktopmanagement in h / Tag	1 h

beantwortet durch Michael Bartsch am 2.12.2015

Thin-Client Infrastructure

Tätigkeit	Aufwand
Installation und Einrichtung der gesamten Thin-Client Infrastructure (geschultes Personal)	min. 1 Arbeitstag
Erstellen eines Masterimages	2 h (je nach Softwareausstattung) 1 h (Vorbereitung + Test Sysprep)
Wartung des Masterimages in h / Monat	4 h (mit 1x Update / Woche)
Bereitstellung eines Desktops (vollwertige Maschine)	10 min Spezifikationen erarbeiten 15 min Konfiguration + kopieren 10 min Initialisieren
Aufbau und Inbetriebnahme eines Thin-Clients	< 0,5 h für Hardware Aufbauen BISO Einstellungen
Administration am Arbeitsplatz des Anwenders h / Monat	Keine, Telefonsupport falls erforderlich
Monitoring und Fehlerbehebung in h / Tag	2 h / Monat
Organisation Desktopmanagement in h / Tag	1 h

beantwortet durch Albert Vass am 2.12.2015