

M306

IT-Kleinprojekte abwickeln

Emanuel Duss, Arno Galliker, Semir Jahic

Gruppe: SP4

# Projektantrag

Einführung von LTSP (Linux Terminal Server Project)

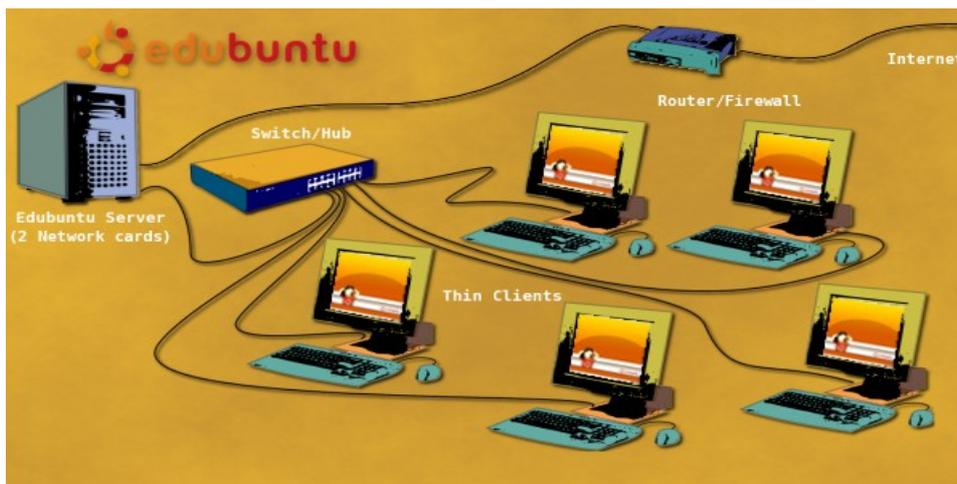


Abbildung 1: [http://www.edubuntu.org/images/ltsp\\_inet.png](http://www.edubuntu.org/images/ltsp_inet.png)

Projektbezeichnung	Einführung von LTSP (Linux Terminal Server Project)
Projektleiter	Emanuel Duss
Verantwortlich	Projektleiter
Erstellt	2008-09-09
Letzte Änderung	2008-09-17 um 14:25:50
Zustand	Zur Prüfung
Pfad	/ media/APACER_2GB/Schule/3_Lehrjahr/306_IT_Kleinprojekt_abwickeln/02_Projekt/ 01_Projektantrag/01_Projektantrag.odt

## Projektmitglieder

Emanuel	Duss	Im Bienz 15	6170 Schüpfheim	emanuel.duss@gmail.com	EDU
Semir	Jahic	Moosmatte 32	6182 Escholzmatt	semir.jahic@gmail.com	SJA
Arno	Galliker	Margrethenstrasse 8	6275 Ballwil	arno.galliker@gmail.com	AGA

## Änderungen

Datum	Version	Kapitel	Beschreibung	Autor
2008-09-09	0.1	Alle	Erster Entwurf	EDU
2008-09-10	0.2	Alle	Überprüfung	AGA
2008-09-12	0.3	Alle	Ergänzungen, Korrektur	SJA
2008-09-17	0.4	Alle	Überarbeitung	Alle

## Prüfungen / Reviews

Datum	Version	Kapitel	Beschreibung	Autor
	0.4	Alle		GAS

## Infos

Zuletzt bearbeitet 2008-09-17

Heute 2009-11-23

Bearbeitungszeit 04:35:21

Lehrjahr des Moduls 3. Lehrjahr; 2008 / 2009

Pfad

/  
media/APACER\_2GB/Schule/3\_Lehrjahr/306\_IT\_Kleinprojekt\_abwickeln/02\_Projekt/01\_Projektantrag/01\_Projektantrag.odt

CC-Lizenz



Creative Commons Namensnennung-Keine kommerzielle Nutzung-Weitergabe unter gleichen Bedingungen 2.5 Schweiz

<http://creativecommons.org/licenses/by-nc-sa/2.5/ch/>

Powered by



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung.....</b>	<b>4</b>
1.1	Zweck.....	4
<b>2</b>	<b>Ausgangslage.....</b>	<b>5</b>
2.1	Software.....	5
2.2	Hardware.....	5
2.3	Weitere Kritikpunkte.....	5
<b>3</b>	<b>Projektziele und Systemvorstellungen.....</b>	<b>6</b>
3.1	Gedanken zum Projekt.....	6
3.2	Projektziele.....	6
3.3	Systemvorstellungen.....	7
3.4	Systemübersicht.....	7
<b>4</b>	<b>Eckwerte der Planung.....</b>	<b>8</b>
4.1	Zeitliche vorgaben.....	8
<b>5</b>	<b>Glossar.....</b>	<b>9</b>
<b>6</b>	<b>Informationen / Links / Interessantes.....</b>	<b>10</b>
6.1	LTSM.....	10
6.2	Installation von LTSM.....	10

# 1 Einleitung

## 1.1 Zweck

Die Schule besitzt ältere Computer, vor allem auch alte Notebooks mit kleiner Festplatte. Eine Neuanschaffung ist zu teuer. Trotzdem sollen die Systeme auf einen aktuellen Stand gebracht werden. Die Lizenzen für Microsoft Windows, Microsoft Office, Adobe Photoshop, Adobe InDesign, spezielle Bildungssoftware, Microsoft Encarta, PowerDVD etc. kosten viel Geld. Eine Alternative soll her. Wir empfehlen der Schule komplett auf OpenSource zu setzen und ein wenig Geld für einen Server statt Lizenzen auszugeben, der als Terminalserver dient. Die alten PCs können weiterhin benutzt werden, da alles auf dem Server abläuft.

Als Betriebssystem empfiehlt sich das Ubuntu-Add-On „Edubuntu“, welches viele Lernsoftware beinhaltet.



## 2 Ausgangslage

### 2.1 Software

Im Moment sieht es ziemlich alt aus bezüglich Software:

- Microsoft Windows 98
- Microsoft Office 2000
- Microsoft Encarta 2000
- Adobe Photoshop 4
- Adobe InDesign 3

### 2.2 Hardware

Auch die Hardware ist schon etwas älter:

- Intel Pentium 4; 1,7 Ghz
- 256 MB RAM
- 30 GB HD
- Onboard-Grafik

### 2.3 Weitere Kritikpunkte

- Als Dateiserver dient ein normaler Client mit einem Samba-Server auf einem Debian 2.2.
- die IP-Adressen werden durch einen DHCP-Fähigen Home-Internet-Router vergeben.
- Für keinen Schüler / Lehrer besteht ein Benutzerkonto.
- Es gibt pro PC eine lokale Anmeldung. Es sind keine Berechtigungen vergeben. Der einzige „Schutz“ ist eine Dr. Kaiser PC-Wächter Karte.

### 3 Projektziele und Systemvorstellungen

#### 3.1 Gedanken zum Projekt

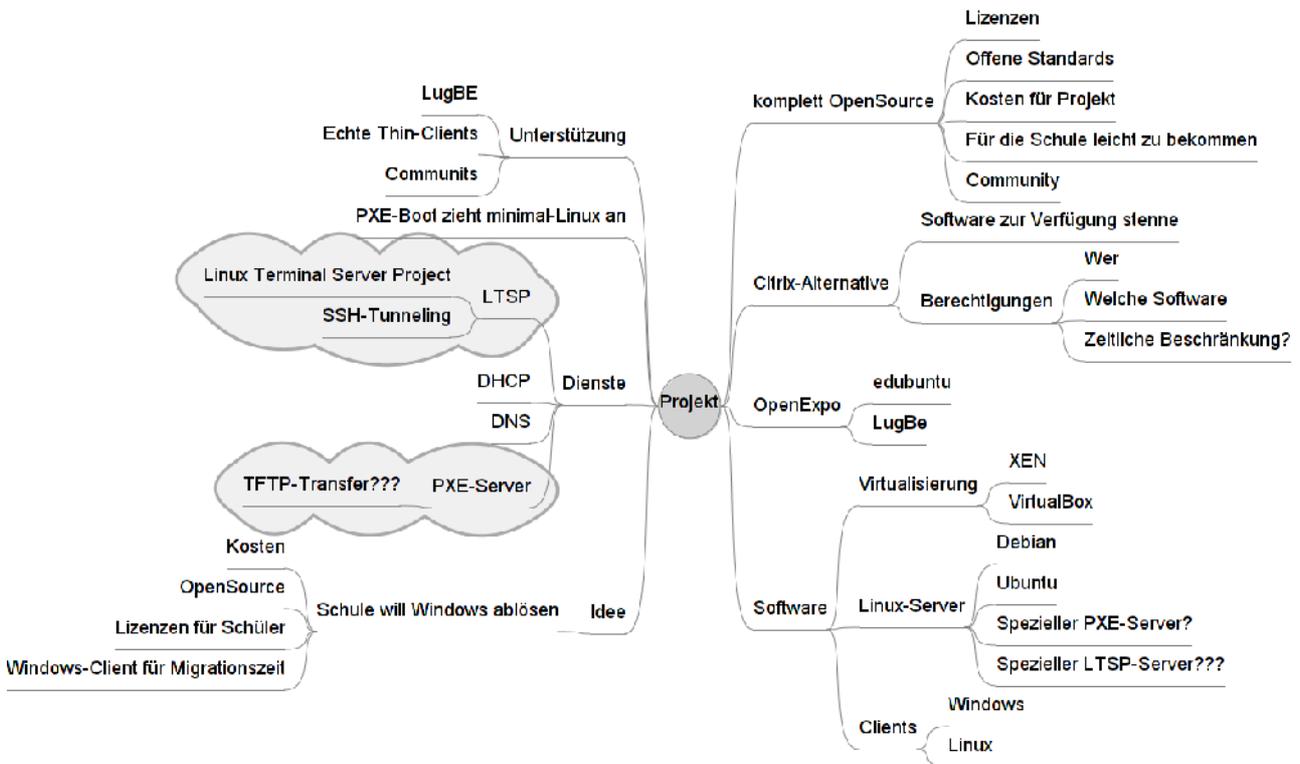


Abbildung 2: MindMap zum Projekt

#### 3.2 Projektziele

- Anschaffen eines neuen Servers.
- Die alte Hardware soll beibehalten werden.
- Die Schule wird komplett auf OpenSource setzen.
- Alle Daten sollen zentral gespeichert werden.
- Jeder Schüler soll sich persönlich anmelden können.
- Die Benutzersession läuft komplett auf dem Server ab. Der Client zeigt nur das vom Server gesehene Bild an. (Terminalserver)

### 3.3 Systemvorstellungen

- Alte Hardware behalten
- Server: Neuwertiger und Leistungsfähiger PC?
- Backup per rsync auf zweite Platte.
- Ubuntu-Server mit LTSP (Linux Terminal Server Projekt)
- Alles OpenSource-Software
- Edubuntu als Client-OS (mit PXE von der Netzwerkkarte starten und per TFTP Minimal-Linux in den RAM laden. Keine HD mehr notwendig!)

### 3.4 Systemübersicht

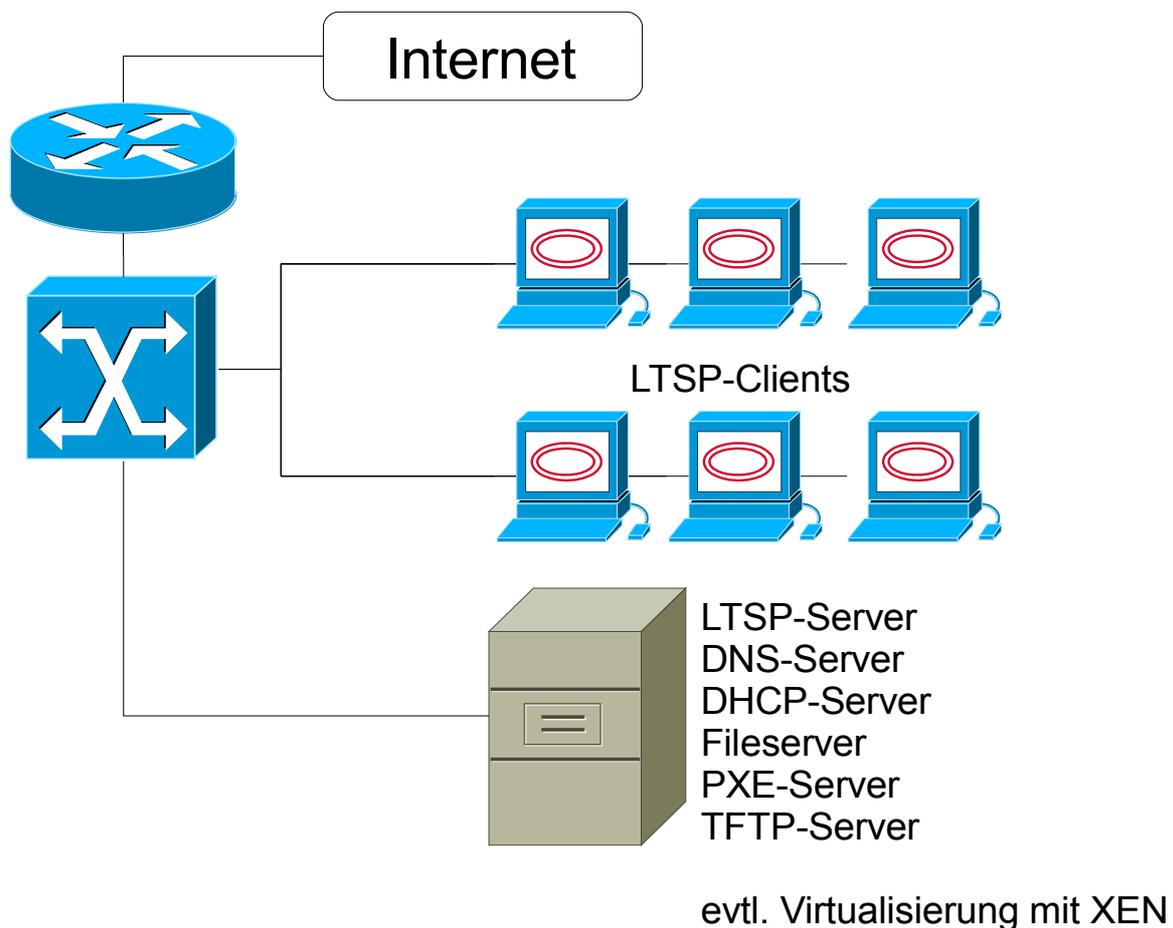


Abbildung 3: Systemübersicht

## 4 Eckwerte der Planung

### 4.1 Zeitliche vorgaben

Einreichen Projektauftrag	KW 38
Projektentscheid	KW 42
Anforderungsspezifikation	...
Abschluss Detailkonzept	
Projekttag	

## 5 Glossar

LTSP	Linux Terminal Server Project
------	-------------------------------

## 6 Informationen / Links / Interessantes

### 6.1 LTSM

- Wikipedia-Artikel: [http://de.wikipedia.org/wiki/Linux\\_Terminal\\_Server\\_Project](http://de.wikipedia.org/wiki/Linux_Terminal_Server_Project)
- Offizielle Seite: <http://ltsp.org/>

### 6.2 Installation von LTSM

- Installation unter Ubuntu: <https://help.ubuntu.com/community/UbuntuLTSP/LTSPQuickInstall>
- Anleitung auf Ubuntuusers.de: <http://wiki.ubuntuusers.de/LTSP>

M306

IT-Kleinprojekte abwickeln

Emanuel Duss, Arno Galliker, Semir Jahic

Gruppe: SP4

# Anforderungsanalyse

Einführung von LTSP (Linux Terminal Server Project)

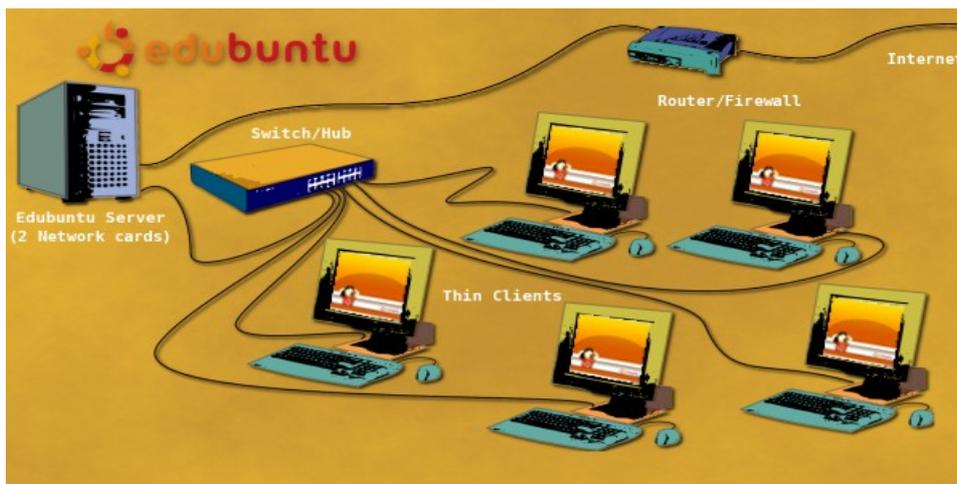


Abbildung 1: [http://www.edubuntu.org/images/ltsp\\_inet.png](http://www.edubuntu.org/images/ltsp_inet.png)

Projektbezeichnung	Einführung von LTSP (Linux Terminal Server Project)
Projektleiter	Emanuel Duss
Verantwortlich	Projektleiter
Erstellt	15.10.08
Letzte Änderung	2009-01-16 um 10:39:17
Zustand	Zur Prüfung
Pfad	/ media/APACER_2GB/Schule/3_Lehrjahr/306_IT_Kleinprojekt_abwickeln/02_Projekt/ 02_Anforderungsanalyse/02_Anforderungsanalyse.odt

## Projektmitglieder

Emanuel	Duss	Im Bienz 15	6170 Schüpfheim	emanuel.duss#gmail.com	EDU
Semir	Jahic	Moosmatte 32	6182 Escholzmatt	semir.jahic#gmail.com	SJA
Arno	Galliker	Margrethenstrasse 8	6275 Ballwil	arno.galliker#gmail.com	AGA

## Änderungen

Datum	Version	Kapitel	Beschreibung	Autor
2008-10-15	0.1	Alle	Grundgerüst vom Dokument	EDU
2008-10-22	0.2	Alle	Einfügen der Textinhalte von dem jeweiligen Autor	Alle
2008-10-25	0.3	Alle	Gemeinsame Überarbeitung	Alle
2008-10-29	0.4	Alle	Vollzogene Änderungen wurden eingetragen	Alle
2008-11-05	0.5	Alle	Weitere Kapitel wurden aufgeteilt und besprochen	EDU
2008-11-07	0.6	2.2	Erstellt	EDU
2008-11-07	0.7	2.3	Erstellt	AGA
2008-11-07	0.8	2.4	Erstellt	SJA
2008-11-07	0.9	2.4	Erstellt	SJA
2008-11-08	0.9	2.7	Ergänzt	AGA
2008-11-08	0.9	2.5	Erstellt	EDU
2008-11-09	1	Alle	Alle Änderungen wurden besprochen und überarbeitet	Alle

## Prüfungen / Reviews

Datum	Version	Kapitel	Beschreibung	Autor
	1	Alle		GAS

## Infos

Zuletzt bearbeitet	2009-01-16
Heute	2009-11-23
Bearbeitungszeit	09:57:30
Lehrjahr des Moduls	3. Lehrjahr; 2008 / 2009
Pfad	/ media/APACER_2GB/Schule/3_Lehrjahr/306_IT_Kleinprojekt_abwickeln/02_Projekt/02_Anforderungsanalyse/02_Anforderungsanalyse.odt

CC-Lizenz



Creative Commons Namensnennung-Keine kommerzielle Nutzung-Weitergabe unter gleichen Bedingungen 2.5 Schweiz

<http://creativecommons.org/licenses/by-nc-sa/2.5/ch/>

Powered by



## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung.....</b>	<b>5</b>
1.1	Zweck des Dokumentes.....	5
1.2	Referenzierte Dokumente.....	5
<b>2</b>	<b>Gesamtüberblick.....</b>	<b>6</b>
2.1	Allgemeine Beschreibung.....	6
2.2	Ziele.....	6
2.3	Spezielle Aspekte.....	7
<b>3</b>	<b>Analyse des IST-Systems.....</b>	<b>8</b>
3.1	Beschreibung.....	8
3.1.1	Textuell.....	8
3.1.1	Bildlich.....	8
3.2	Schwachstellen.....	8
<b>4</b>	<b>Zielkatalog.....</b>	<b>9</b>
4.1	Beschreibung des SOLL-Systems.....	9
4.2	Verbesserungen.....	9
4.3	Die Lösungsidee.....	9
4.1	Bildliche Darstellung.....	10
<b>5</b>	<b>Anforderungen.....</b>	<b>11</b>
5.1	Ziele.....	11
5.1.1	Muss-Ziele.....	11
5.1.1	Kann-Ziele.....	11
5.2	Akteure.....	12
5.2.1	Akteurbeschreibung.....	12
5.3	Funktionale Anforderungen.....	13
5.3.1	Anforderungen aus der Sicht des Users (Schüler, Lehrer).....	13
5.3.2	Anforderungen aus der Sicht des Administrators.....	13
5.3.3	Anforderungen aus der Sicht des Systems.....	13
5.4	Use-Cases.....	14
5.4.1	UC01 „Starten der Umgebung“.....	14
5.4.2	UC02 „Server warten“.....	14
5.4.3	UC03 „Verwendung von Applikationen“.....	14
5.4.4	UC04 „Verwendung der öffentlichen Shares“.....	14
5.5	Use-Case Diagramme.....	15
5.5.1	UC01 und UC02.....	15
5.5.2	UC03 und UC04.....	16
5.6	Aktivitätsdiagramm.....	17
5.7	Glossar.....	18

## Abbildungsverzeichnis

Abbildung 1:	<a href="http://www.edubuntu.org/images/ltsp_inet.png">http://www.edubuntu.org/images/ltsp_inet.png</a> .....	1
Abbildung 2:	Somalia (An der östlichen Küste).....	5
Abbildung 3:	IST-System: Alles läuft lokal. Funktioniert nicht gut.....	7
Abbildung 4:	SOLL-System: Alles läuft Zentral über den Server.....	9
Abbildung 5:	Das Aktivitätsdiagramm (User meldet sich an).....	12

# 1 Einleitung

## 1.1 Zweck des Dokumentes

Hierbei handelt es sich um die Anforderungsspezifikation vom Projekt „Einführung von LTSP (Linux Terminal Server Project)“. Vgl. [1].

Das Projekt wird für die Schule „Somalia High-School“ realisiert. In diesem Dokument wird beschrieben, welche Anforderungen das Projekt erfüllen muss.

## 1.2 Referenzierte Dokumente

[1] Projektantrag vom 9.9.2008

## 2 Gesamtüberblick

### 2.1 Allgemeine Beschreibung

In Afrika, wo die Entwicklungsländer den technologischen Fortschritt noch nicht so erlebt haben wie die westliche Welt, ist Bildung rar. Darum werden wir in Somalia ein LTSP Projekt mit Edubuntu initiieren.

Bei unserem Projekt ist die Somalia High-School unser Kunde, es ist eine Schule in Ostafrika. In der Kleinstadt Balad befindet sich diese kleine Schule, die etwa 50km Nord-östlich der Hauptstadt Mogadischu liegt. Die Schule hat nur sehr begrenzte finanzielle Möglichkeiten. Darum kann die Schulleitung sich keine kostspieligen Investitionen leisten. Trotz der Einschränkungen wird die Schule eine gute Ausstattung erhalten. Die Hardware, die wir nutzen ist nicht brandneu und auf dem neusten technischen Stand. Es sind alte Computer die nun als „Thin-Clients“ fungieren.



Abbildung 2: Somalia (An der östlichen Küste)

Die Hardware kann ohne Probleme noch weiter verwendet werden, denn Thin-Clients haben keine hohen Hardwareanforderungen. Wir werden es möglichst kostengünstig für die Schule gestalten. Die Software soll kostenlos sein und dies können wir mit Open-Source realisieren.

Es geht darum, dass die Schule möglichst viel für wenig machen kann und dies wird so möglich, Ubuntu wird das zu verwendende Betriebssystem, welches wir mit dem für Schulen entwickelten Addo-On <sup>1</sup> ergänzen.

### 2.2 Ziele

Die Schulleitung wünscht sich ein einfaches System. Es soll keine hohen Administrationsaufgaben geben, es soll einfach zu überschauen und schnell einzusetzen sein. Was vor allem wichtig ist, es soll die Bildung der Schüler fördern. Darum sind Faktoren wie Usability und Einfachheit von sehr hoher Priorität. Es soll möglich sein Anwendungen wie den Writer vom OpenOffice Suite einfach zu starten und Texte zu schreiben. Genauso soll man spielerisch Rechnen lernen, mit Lernprogrammen die intuitiv zu bedienen sind. Ein weiterer wichtiger Punkt ist ein Lernprogramm für Sprachen, um so Englisch, die am meisten genutzte Sprache weltweit zu erlernen.

Neben der Lernsoftware soll es auch möglich sein Daten auszutauschen, die Schüler sollen so die Möglichkeit haben eigene Daten zu erstellen und diese mit anderen auszutauschen.

Die Lehrperson, welche die Klasse betreut, soll ohne viele Kenntnisse die Möglichkeit haben, den Schülern zu helfen, zum Beispiel via Fernadministration.

Der zentrale Server im Klassenzimmer soll stets auf dem neusten Stand sein und alle Updates empfangen, der Zugang zum Internet soll uneingeschränkt möglich sein. Es ist wichtig das die Software stets auf dem neusten Stand ist.

---

<sup>1</sup> Edubuntu ist seit LTS-Version 8.04 ein Add-On von Ubuntu und keine eigenständige Distribution mehr.

## 2.3 Spezielle Aspekte

Unsere Schule hat grundsätzlich keine Daten die sensibel sind und von Zugriffen geschützt werden müssen. Darum werden wir auf den speziellen Aspekt der Sicherheit nicht viel Wert legen. Jedoch soll ein Backup sehr zuverlässig sein. Das System benötigt eine hohe Verfügbarkeit, denn das Klassenzimmer ist das einzige mit Informatikinfrastruktur und muss für über 250 Schüler und Schülerinnen zur Verfügung stehen. Ein weiterer wichtiger Aspekt ist, dass die Systeme sicher vor Viren sein müssen und dies garantieren wir grundsätzlich mit Linux, jedoch sollen diverse Internetseiten mit pornografischem Inhalt gesperrt werden.

Das System soll eine zentrale Steuerung haben, darum werden wir einen Hauptserver einsetzen. Sonstige spezielle Aspekte gibt es gegenwärtig noch nicht.

## 3 Analyse des IST-Systems

### 3.1 Beschreibung

#### 3.1.1 Textuell

Zur Zeit sieht es folgendermassen im Schulhaus aus:

1. Schüler / Lehrer startet den PC (Windows 98)
2. Schüler / Lehrer meldet sich mit einer allgemeinen lokalen PC-Anmeldung an
3. Schüler greifen über eine Verknüpfung auf eine Samba-Freigabe auf ein Verzeichnis zu. In diesem Verzeichnis sind alle Daten der Schüler gespeichert. Alle Schüler haben auf alle Ordner Zugriff.
4. Beim herunterfahren werden alle Einstellungen verworfen. Das erledigt die Dr. Kaiser PC-Wächter Karte (PCI).

#### 3.1.1 Bildlich

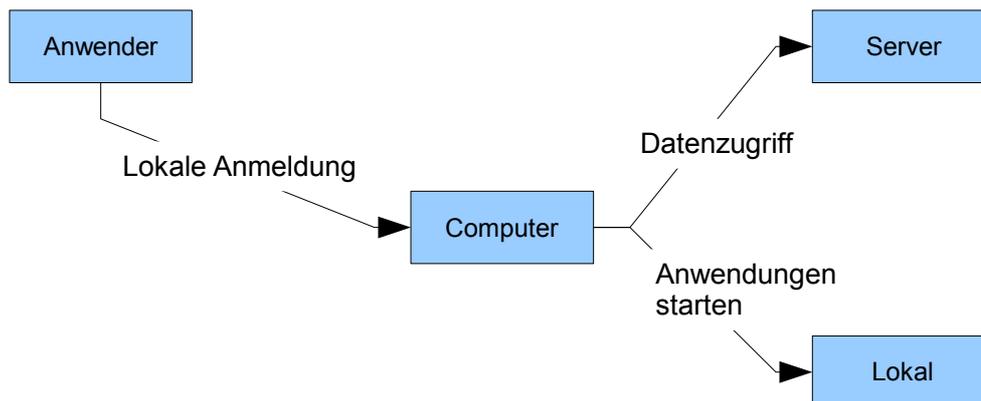


Abbildung 3: IST-System: Alles läuft lokal. Funktioniert nicht gut.

### 3.2 Schwachstellen

- Kein Benutzerprofil pro Schüler / Lehrer
- Die Einstellungen werden nicht persönlich gespeichert
- Alle Einstellungen gehen verloren
- Alle Daten sind für alle einsichtig
- Da die Hardware sehr alt ist, läuft alles sehr langsam.
- Keine zentrale Administration

## 4 Zielkatalog

### 4.1 Beschreibung des SOLL-Systems

Die Schule soll nun besser ausgestattet werden. Alles soll schnell laufen aber trotzdem nicht viel kosten. Wir schlagen der Schule in Somalia folgende Lösung vor:

### 4.2 Verbesserungen

1. Jeder Schüler erhält ein persönliches Benutzerprofil
2. Jeder Schüler erhält ein eigenes Home-Verzeichnis
3. Die Software soll stets aktuell bleiben
4. Eine zentrale Verwaltung ist notwendig
5. Das System soll angenehm und nicht langsam laufen

### 4.3 Die Lösungsidee

- LTSP wird in den Einsatz kommen
- Der Benutzer startet den bisherigen PC
  - Der PC bootet von der Netzwerkkarte (PXE-Boot)
  - Der PC holt eine IP-Adresse vom DHCP-Server
  - Der PC zieht ein Minimallinux von von einem TFTP-Server
  - Das Minimallinux wird gestartet
  - Das Minimallinux verbindet sich mit dem LTSP-Server
  - Das Bild wird vom LTSP-Server übertragen und auf den Client angezeigt
- Der Benutzer kann sich nun „auf dem Server“ anmelden.
- Dort hat er sein Profil mit den jeweiligen Einstellungen, sein Home-Verzeichnis, die Austausch-Verzeichnisse und Programme verfügbar.

## 4.1 Bildliche Darstellung

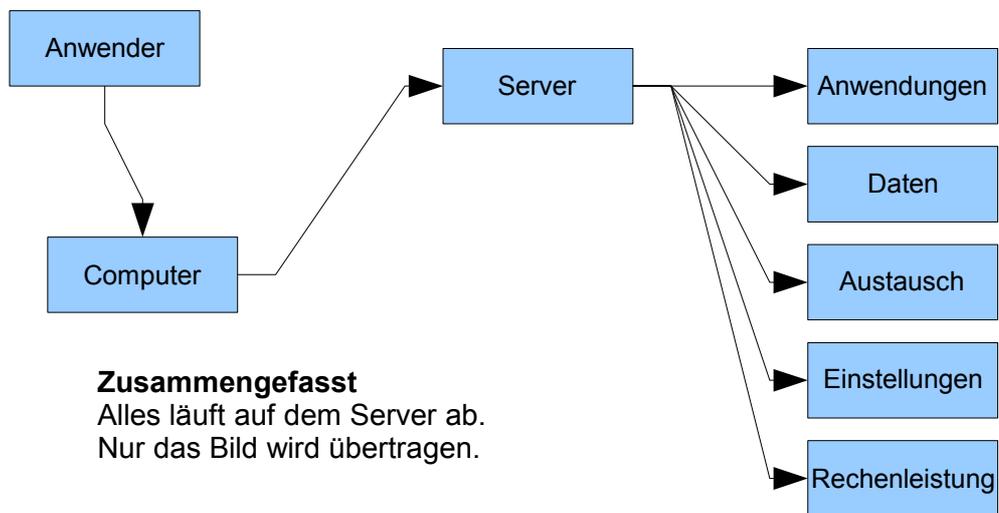


Abbildung 4: SOLL-System: Alles läuft Zentral über den Server

## **5 Anforderungen**

### **5.1 Ziele**

#### **5.1.1 Muss-Ziele**

1. Alte Hardware weiterhin einsetzen
2. Anschaffung eines leistungsstarken Servers
3. Einfache Integration der Geräte
4. Keine Lizenzkosten
5. Günstiger und einfacher Betrieb der gesamten Anlage
6. Zentrale Datenhaltung (Homes, Shares)
7. Zentrale Administration

#### **5.1.1 Kann-Ziele**

1. Komplette Open-Source
2. Alle Dienste (PXE, DHCP usw.) auf einem Server
3. Virtualisierungslösungen
4. Persönliche Benutzerlogins für alle Akteure
5. Überwachungsprogramme

## 5.2 Akteure

### 5.2.1 Akteurbeschreibung

Akteur-Name	Beschreibung
Schüler	<ul style="list-style-type: none"> <li>● Der Schüler besucht die Schule und erwartet eine gute Infrastruktur. Zu dieser gehört auch das Netzwerk mit den Computern.</li> <li>● Der Schüler meldet sich an einem Computer an und erwartet seine gewohnte Umgebung.</li> <li>● Der Schüler nutzt den ThinClient</li> </ul>
Lehrer	<ul style="list-style-type: none"> <li>● Der Lehrer funktioniert genau wie ein Schüler, ausser dass er Schreibrechte auf ein für die Schüler nur ReadOnly-Direcotry hat.</li> <li>● Der Lehrer nutzen den ThinClient.</li> </ul>
Administrator	<ul style="list-style-type: none"> <li>● Der Administrator wartet das Netzwerk.</li> <li>● Durch eine zentrale Verwaltung geschieht das viel einfacher. Er kann alle Einstellungen auf dem Server vornehmen.</li> <li>● Die Clients müssen praktisch nie konfiguriert werden.</li> </ul>
Thin-Client	<ul style="list-style-type: none"> <li>● Der Client stellt beim Booten automatisch eine Verbindung zum Server her.</li> <li>● Die Hardwareanforderungen sind sehr gering.</li> </ul>
Server	<ul style="list-style-type: none"> <li>● Der Server stellt alle Dienste zur Verfügung. So z.B. DHCP, DNS, LTSP, NFS, TFTP, PXE, etc... (Das wird später genauer behandelt.)</li> <li>● Auf dem Server wird alles konfiguriert. Vom Adressbereich bis zu den Usereinstellungen und Datenshares.</li> </ul>

## **5.3 Funktionale Anforderungen**

### **5.3.1 Anforderungen aus der Sicht des Users (Schüler, Lehrer)**

Der User (Lehrer, Schüler) will jederzeit von einem ThinClient aus auf die Umgebung und auf seine persönlichen Daten sowie die Austausch-Laufwerke zugreifen können.

### **5.3.2 Anforderungen aus der Sicht des Administrators**

Der Administrator kann das Netzwerk sowie den zentralen Server jederzeit konfigurieren und warten. Die Integration neuer Geräte soll dabei sehr einfach von statten gehen.

Der Administrator kann auf dem zentralen Server User und deren Profile verwalten, sowie alle weiteren Dienste, welche der Umgebung zur Verfügung stehen.

### **5.3.3 Anforderungen aus der Sicht des Systems**

Der zentrale Server soll jederzeit verfügbar sein und seine verschiedenen Dienste (z.B. DHCP, DNS, LTSP, NFS, TFTP, PXE, etc...) zuverlässig anbieten.

Die ganze Umgebung soll jederzeit einfach durch weitere ThinClients erweiterbar sein.

## 5.4 Use-Cases

### 5.4.1 UC01 „Starten der Umgebung“

Auslöser	Schüler oder Lehrer
Voraussetzung	Thinclient funktionsfähig im Netzwerk
Funktionsablauf	Installation eines minimalen Linuxbetriebssystems
Endzustand	Thinclient steht zur Verwendung der Personen im Klassenzimmer bereit

### 5.4.2 UC02 „Server warten“

Auslöser	Administrator
Voraussetzung	Server funktionsfähig im Netzwerk
Funktionsablauf	Login auf dem Server, Internetverbindung für Updates etc.
Endzustand	Server ist wieder up to date und kann weiterverwendet werden

### 5.4.3 UC03 „Verwendung von Applikationen“

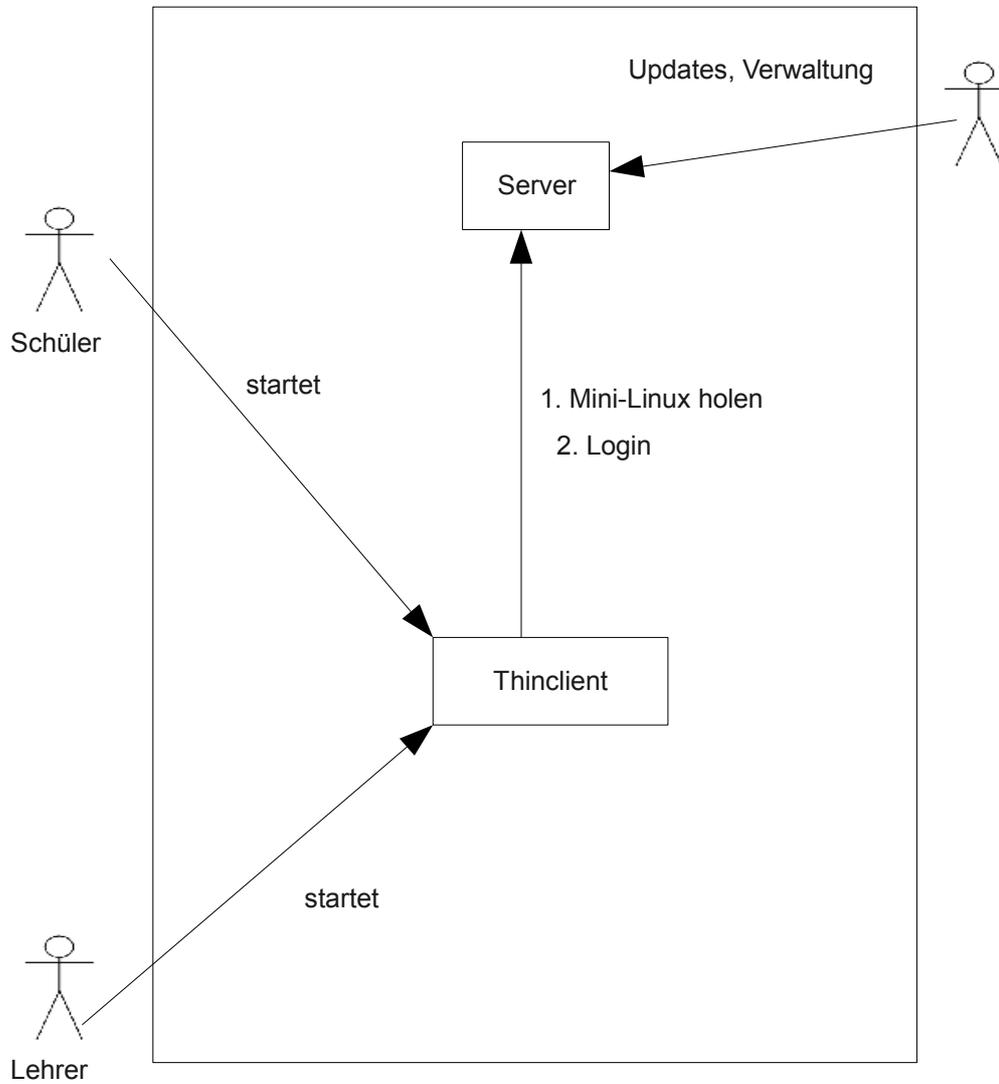
Auslöser	Schüler oder Lehrer
Voraussetzung	Client am Netz mit Konnektivität
Funktionsablauf	Starten einer Applikation durch das Betriebssystem durch den Benutzer
Endzustand	Verwendung der gestarteten Applikation

### 5.4.4 UC04 „Verwendung der öffentlichen Shares“

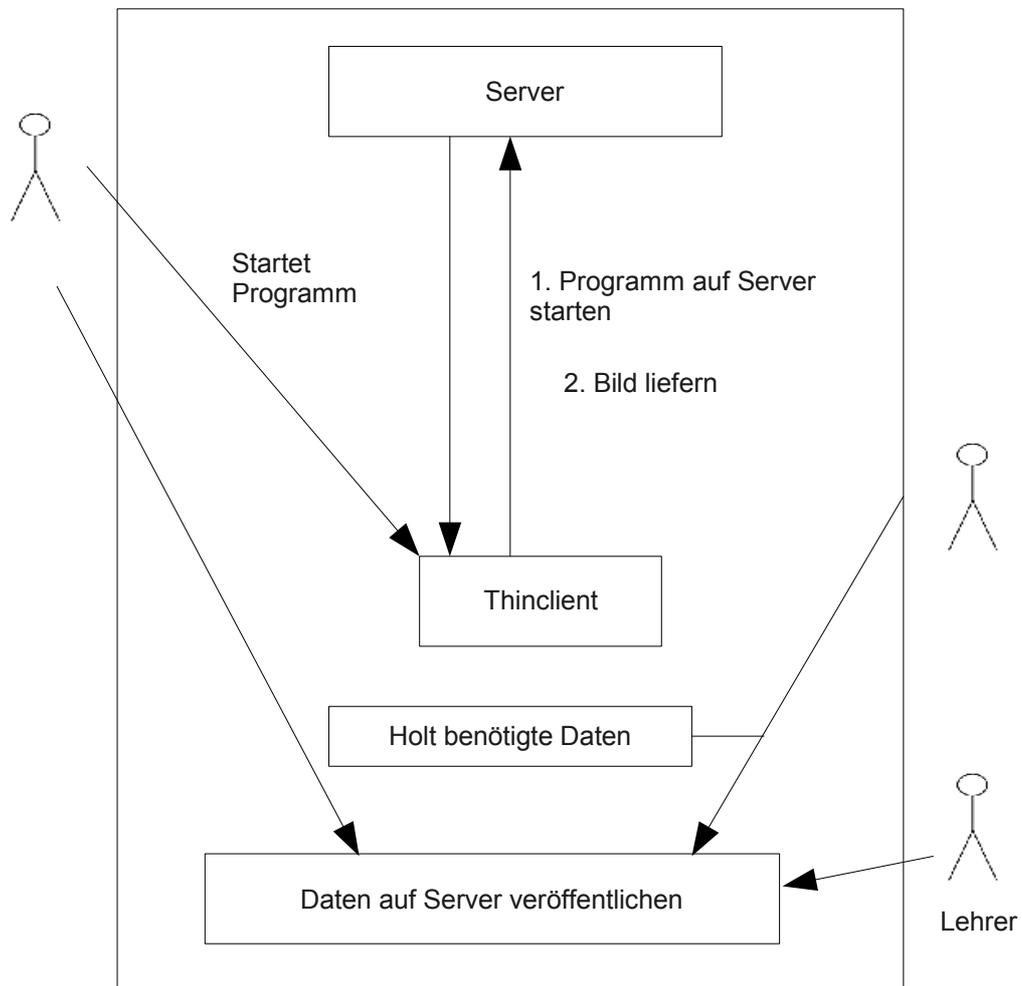
Auslöser	Schüler oder Lehrer
Voraussetzung	Client am Netz mit Konnektivität
Funktionsablauf	Daten auf dem öffentlichen Share speichern
Endzustand	Daten sind übers Netzwerk ausgetauscht worden.

## 5.5 Use-Case Diagramme

### 5.5.1 UC01 und UC02



### 5.5.2 UC03 und UC04



## 5.6 Aktivitätsdiagramm

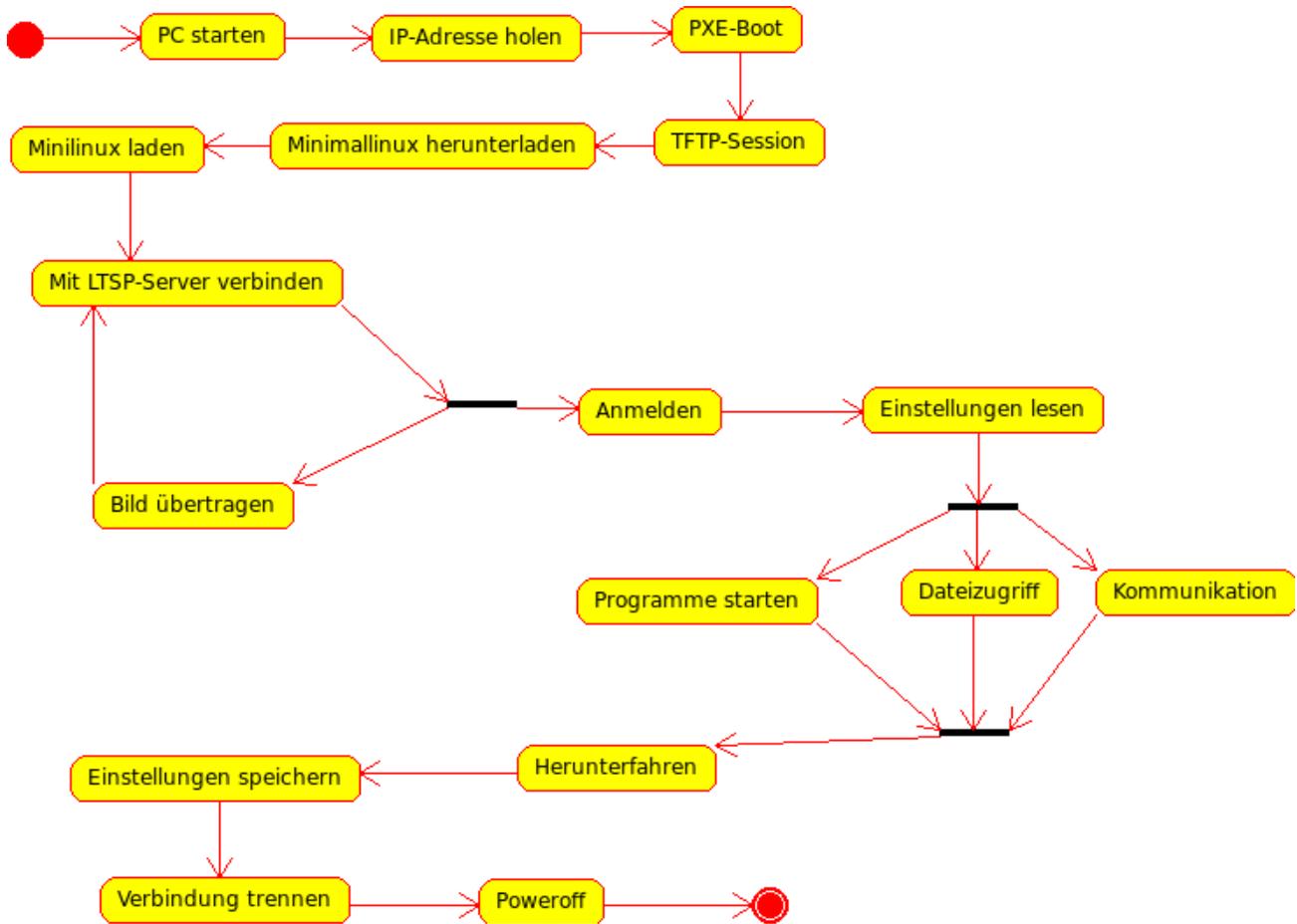


Abbildung 5: Das Aktivitätsdiagramm (User meldet sich an)

## 5.7 Glossar

Folgende Begriffe sind für einen Normalsterblichen genauer erklärt:

<b>Begriff</b>	<b>Erklärung</b>
DHCP	Die Thin-Clients holen sich vom DHCP-Server selber eine IP-Adresse
DNS	Die Pcs sollen einen für einen Menschen lesbaren „Namen“ bekommen.
IP-Adresse	Eindeutige „Hausnummer“ eines Computers in einem Netzwerk.
LTSP	Unsere Hauptsoftware heisst LTSP und steht für Linux Terminal Server Project.
NFS	Network File System. Man kann auf Ordner auf dem Server zugreifen.
PXE	Statt von der HD wird von der Netzwerkkarte gestartet.
TFTP	Primitives Verfahren zur Datenübertragung zwischen zwei PCs.

M306

IT-Kleinprojekte abwickeln

Emanuel Duss, Arno Galliker, Semir Jahic

Gruppe: SP4

# Detailkonzept

Einführung von LTSP (Linux Terminal Server Project)



**eos opensource solutions**

Projektbezeichnung	Einführung von LTSP (Linux Terminal Server Project)
Projektleiter	Emanuel Duss
Verantwortlich	Projektleiter
Erstellt	15.10.08
Letzte Änderung	2009-06-03 um 11:25:36
Zustand	Zur Prüfung
Pfad	/ media/APACER_2GB/Schule/3_Lehrjahr/306_IT_Kleinprojekt_abwickeln/02_Projekt/ 03_Detailkonzept/03_Detailkonzept.odt

## Projektmitglieder

Emanuel	Duss	Im Bienz 15	6170 Schüpfheim	emanuel.duss@gmail.com	EDU
Semir	Jahic	Moosmatte 32	6182 Escholzmatt	semir.jahic@gmail.com	SJA
Arno	Galliker	Margrethenstrasse 8	6275 Ballwil	arno.galliker@gmail.com	AGA

## Änderungen

Datum	Version	Kapitel	Beschreibung	Autor
2008-11-26	0.1	ALLE	Grundgerüst vom Dokument	EDU
2009-01-09	0.2	ALLE	Erstellung der Inhalte	ALLE
2009-01-16	0.5	ALLE	Zusammenführung der erstellten Kapitel	ALLE

## Prüfungen / Reviews

Datum	Version	Kapitel	Beschreibung	Autor
	1	Alle		GAS

## Infos

Zuletzt bearbeitet	2009-06-03
Heute	2009-11-23
Bearbeitungszeit	18:19:03
Lehrjahr des Moduls	3. Lehrjahr; 2008 / 2009
Pfad	/ media/APACER_2GB/Schule/3_Lehrjahr/306_IT_Kleinprojekt_abwickeln/02_Projekt/03_Detailkonzept/03_Detailkonzept.odt

CC-Lizenz



Creative Commons Namensnennung-Keine kommerzielle Nutzung-Weitergabe unter gleichen Bedingungen 2.5 Schweiz

<http://creativecommons.org/licenses/by-nc-sa/2.5/ch/>

Powered by



## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung.....</b>	<b>5</b>
1.1	Zweck des Dokumentes.....	5
1.2	Referenzierte Dokumente.....	5
<b>2</b>	<b>Allgemeines.....</b>	<b>6</b>
2.1	Neu zu erstellen.....	6
2.2	Anpassungen an bestehenden Komponenten.....	6
<b>3</b>	<b>Systemübersicht SOLL-System.....</b>	<b>7</b>
3.1	Konfigurationsdateien auf dem Server.....	7
3.2	Server (Hardware).....	8
3.2.1	Funktionsbeschreibung.....	8
3.2.2	Funktionale Anforderungen.....	8
3.2.3	Use-Cases.....	8
3.3	Client (Hardware).....	10
3.3.1	Funktionsbeschreibung.....	10
3.3.2	Funktionale Anforderungen.....	10
3.3.3	Use-Cases.....	10
3.4	LTSP-Server.....	12
3.4.1	Funktionsbeschreibung.....	12
3.4.2	Funktionale Anforderungen.....	12
3.4.3	Use-Cases.....	13
3.5	OpenSSH-Server.....	15
3.5.1	Funktionsbeschreibung.....	15
3.5.2	Funktionale Anforderungen.....	15
3.5.3	Use-Cases.....	16
3.6	DHCP-Server QUELLE.....	17
3.6.1	Funktionsbeschreibung.....	17
3.6.2	Funktionale Anforderungen.....	17
3.6.3	Use-Case.....	18
3.7	NBD-Server.....	19
3.7.1	Funktionsbeschreibung.....	19
3.7.2	Funktionale Anforderungen.....	19
3.7.3	Use-Cases.....	19
3.7.4	Weitere Infos / Quellen.....	19
<b>4</b>	<b>TFTP-Server.....</b>	<b>20</b>
4.1.1	Funktionsbeschreibung.....	20
4.1.2	Funktionale Anforderungen.....	21
4.1.3	Use-Cases.....	21
<b>5</b>	<b>Glossar.....</b>	<b>22</b>

## Abbildungsverzeichnis

Abbildung 1:	Soll-System Netzwerkübersichtsplan.....	7
--------------	---	---

# 1 Einleitung

## 1.1 Zweck des Dokumentes

Dieses Dokument stellt das Detailkonzept zur Einführung des Projekts „Einführung von LTSP (Linux Terminal Server Project)“ dar.

Die einzelnen Systemteile und deren Zusammenwirken, sowie die Anbindung an bestehende Systeme werden in diesem Dokument detailliert beschrieben und bilden die Grundlage für die Implementierung des Systems.

## 1.2 Referenzierte Dokumente

[1] Projektantrag vom 9.9.2008

[2] Anforderungsspezifikation

## 2 Allgemeines

Unser Projekt besteht aus folgenden Hauptkomponenten, welche entweder von Grund auf neu erstellt werden müssen oder welche abgeändert werden müssen:

### 2.1 Neu zu erstellen

- Server (Hardware)
  - Anforderungen HW
- LTSP-Server
- Hauptsystem von unserem Projekt
- OpenSSH-Server
  - SSH-Zugriff
  - SCP
- Zertifikatbasierte Authentifikation
- TFTP-Server
  - (x)inetd oder dämon?
  - Um das Minimallinux anzubieten
- NBD
- LTSPFS
- Falls Clients z.B. USB-Sticks verwenden möchten

### 2.2 Anpassungen an bestehenden Komponenten

- Client (Hardware)
  - Netzwerkkarte muss PXE-Fähig sein
  - Boot-Reihenfolge
- DHCP-Server
  - Früher wurde an der Schule der Router benutzt als DHCP-Server. Diese Aufgabe wird nun vom LTSP-Server übernommen, weil wir alles zentral verwalten möchten!

### 3 Systemübersicht SOLL-System

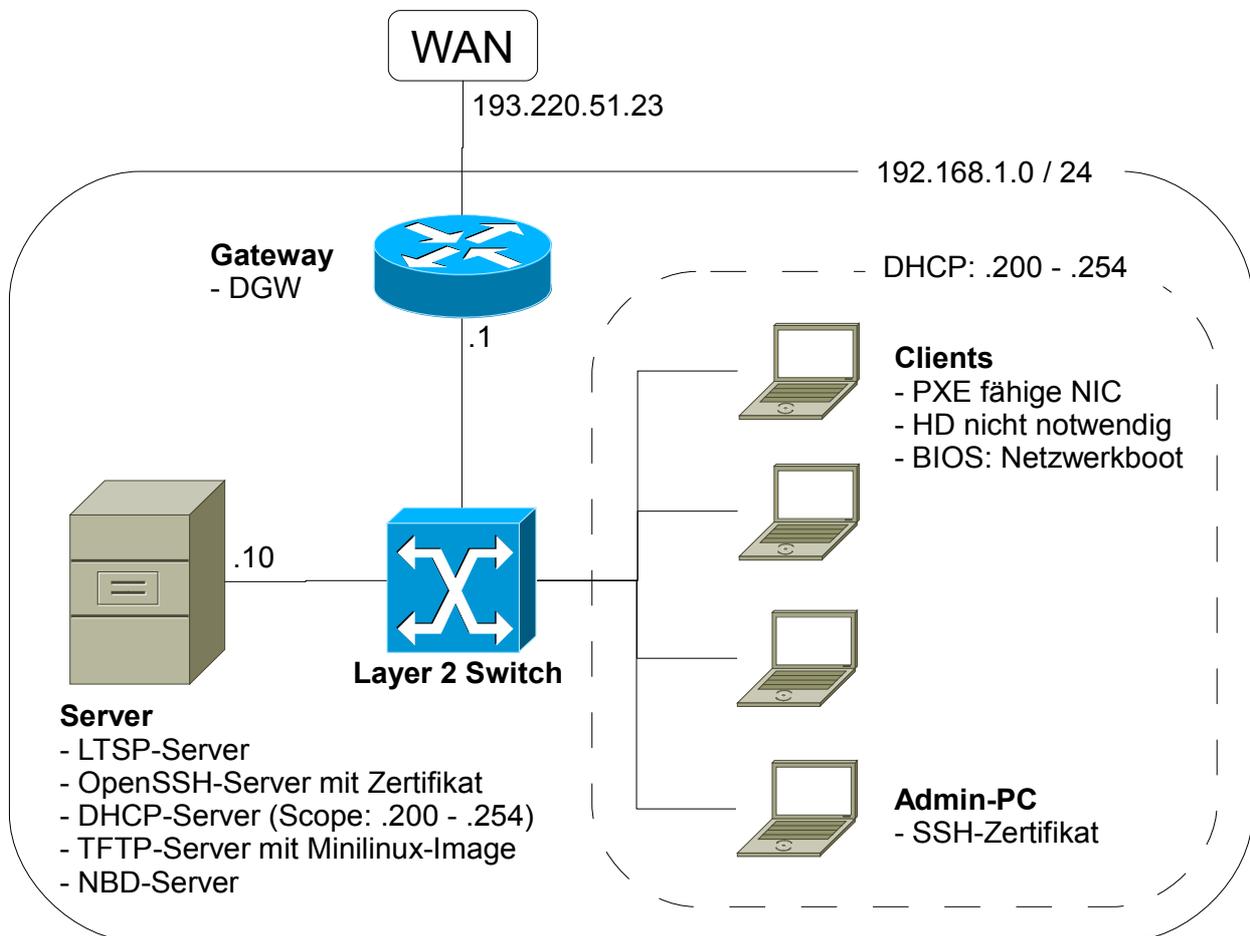


Abbildung 1: Soll-System Netzwerkübersichtsplan

#### 3.1 Konfigurationsdateien auf dem Server:

Datei	Zweck
~.vimrc	Damit man ordentlich mit dem VIM arbeiten kann
/etc/	Hier werden diverse Konfig-Files abgelegt. Beschreibung weiter unten in den Details.
.screenrc	Damit man mehrere virtuelle Terminals z.B. Per SSH zur Verfügung hat

## 3.2 Server (Hardware)

### 3.2.1 Funktionsbeschreibung

Auf dem Server laufen alle Dienste, werden die Ressourcen für alle Clients gebraucht und sind alle Daten zentral gespeichert. Daher ist es nötig, dass die Serverhardware möglichst ausfallsicher, schnell und zuverlässig zur Verfügung steht.

### 3.2.2 Funktionale Anforderungen

#### Festplatten

Der Speicherplatz ist nach den Anforderungen des Kunden zu wählen. Weiter sollten die Festplatten schnell und möglichst ausfallsicher sein. Es ist ein RAID (Spiegelung usw.) zu empfehlen.

#### CPU

Eine CPU mit möglichst hoher Leistung ist Voraussetzung, da die Rechenleistung ausschliesslich auf dem Server stattfindet.

Weiter stellt unser Server auch weitere Dienste zur Verfügung, die ebenfalls Rechenaufwand mit sich bringen.

#### RAM

Auch hier ist möglichst viel und schneller Arbeitsspeicher für ein flüssiges Arbeiten an den Terminals von Vorteil.

#### Ausfallsicherheit

Alle Komponenten sollten so gewählt, verbaut und konfiguriert werden, dass eine gewisse Ausfallsicherheit gewährleistet werden kann. Ohne das zuverlässige Dasein des zentralen Server ist die ganze Umgebung schnell unbrauchbar.

### 3.2.3 Use-Cases

#### Funktion 1

<b>Auslöser</b>	Administrator
<b>Voraussetzung</b>	Hardware bereit
<b>Funktionsablauf</b>	Administrator kann den Server und die dazu benötigte Software konfigurieren.
<b>Endzustand</b>	Der Server ist konfiguriert.

## Funktion 2

<b>Auslöser</b>	Administrator
<b>Voraussetzung</b>	Funktion 1
<b>Funktionsablauf</b>	Software installieren
<b>Endzustand</b>	Software installiert.

## Funktion 3

<b>Auslöser</b>	Administrator
<b>Voraussetzung</b>	Funktion 1 und 2
<b>Funktionsablauf</b>	User kann die Dienste des Servers benutzen.
<b>Endzustand</b>	Terminalserver-Umgebung kann genutzt werden.

## 3.3 Client (Hardware)

### 3.3.1 Funktionsbeschreibung

Da bei einem Terminal nur die Ein- und Ausgangs-Signale (Maus, Tastatur, Bildschirm) zum Terminal Server übertragen werden, reicht ein ThinClient eigentlich völlig aus. Jedoch können auch PC oder Notebooks dazu benutzt werden.

Im folgenden werden die Mindestanforderungen beschrieben.

### 3.3.2 Funktionale Anforderungen

#### System

Ein allgemein funktionierendes System (CPU, RAM, Grafikkarte usw.) muss vorhanden sein. Eine Harddisk ist jedoch nicht nötig, da nur Daten in den flüchtigen Speicher geschrieben werden.

#### Netzwerkkarte

Eine funktionierende, PXE-fähige Netzwerkkarte, welche mindestens 100 Mbit/s übertragen kann.

#### Ein- und Ausgabegeräte

Ein Bildschirm mit einer Auflösung von mindestens 800x600 Pixel (weniger funktioniert auch, ist aber sehr unkomfortabel).

Tastatur und Maus müssen vorhanden sein.

### 3.3.3 Use-Cases

#### Funktion 1: Hochfahren des Computers

<b>Auslöser</b>	User
<b>Voraussetzung</b>	Funktionsfähiger PC
<b>Funktionsablauf</b>	User schaltet Computer an.
<b>Endzustand</b>	Der Computer ist gestartet.

#### Funktion 2: Computer benutzen

<b>Auslöser</b>	User
<b>Voraussetzung</b>	Funktionsfähiger PC
<b>Funktionsablauf</b>	User bedient den Computer mit Maus und Tastatur. Grafikausgabe am Bildschirm.
<b>Endzustand</b>	Der User ist zufrieden mit dem System. Lokal wird nichts gemacht. Nur auf dem Terminalserver.

**Funktion 3: Computer herunterfahren**

<b>Auslöser</b>	User
<b>Voraussetzung</b>	Funktionsfähiger PC
<b>Funktionsablauf</b>	User schaltet Computer aus.
<b>Endzustand</b>	Der Computer ist heruntergefahren

## 3.4 LTSP-Server

### 3.4.1 Funktionsbeschreibung

Das Linux Terminal Server Project (LTSP) ist ein Linux-Programmpaket, das dazu dient, Benutzern von Terminals (z.B. ThinClients) Zugriff auf den Terminal-Server zu gewähren, von dem aus Anwendungen ausgeführt und mittels des Terminals gesteuert werden können.

Bei einem Terminal werden nur die Ein- und Ausgangs-Signale (Maus, Tastatur, Bildschirm) zum Terminal Server übertragen. Die eigentliche Verarbeitung der Daten, also die gesamte Rechenleistung ist auf dem Terminal Server.

### 3.4.2 Funktionale Anforderungen

#### Bootvorgang mit PXE

LTSP unterstützt diverse Boot-Methoden (z.B. Etherboot, PXE, RPL).

Wir haben uns für PXE entschieden.

Per PXE wird dem Client auf den TFTP-Server verbunden, der dann dem Client das Systemimage schickt.

#### Oberfläche

Die aktuellen LTSP-Versionen liefern den Clients eine Umgebung, die einer regulären Linux-Distribution entspricht. Dadurch kann der Benutzer das Linux-System der Thin Clients beliebig erweitern und Pakete nach eigenem Wunsch installieren.

#### Lokale Medien

Der LTSP-Server kann auch lokal an den Client angeschlossene Medien (Drucker, Sound usw.) auf der Linux-Session verfügbar machen.

#### Betriebssystem

Wir installieren ein Ubuntu 8.10 System. Zum einrichten von LTSP gibt es eine spezielle Instalations-CD. Mit der Alternate Instal CD kann man beim Bootvorgang auswählen, ob wir eine LTSP-Server-Installation durchführen wollen:



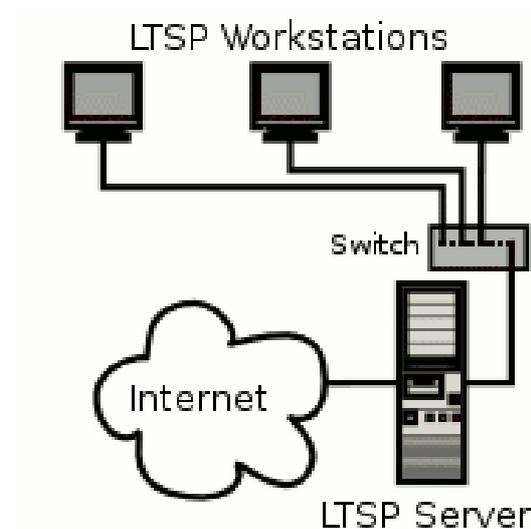
Folgende eingaben müssen wir während der Installation machen:

- System: Ubuntu 8.10
- Installation: Ubuntu 8.10 Alternate installation (LTSP-Server-Installation)

- Land: Schweiz
- Sprache: Deutsch
- Servername: ltsp
- Username: ltsp
- Passwort: ltsp

Die meisten Konfigurationen sind bereits gemacht. Die Konfigurationen können in der `lts.conf` vorgenommen werden.

### 3.4.3 Use-Cases



#### Funktion 1: Am Server anmelden

<b>Auslöser</b>	User
<b>Voraussetzung</b>	Funktionsfähiger PC +LTSP-Server
<b>Funktionsablauf</b>	User meldet sich am Terminalserver an.
<b>Endzustand</b>	Der User ist angemeldet und kann den Desktop bedienen.

#### Funktion 2: Anmeldung entgegennehmen

<b>Auslöser</b>	User
<b>Voraussetzung</b>	Funktionsfähiger PC +LTSP-Server
<b>Funktionsablauf</b>	Der LTSP-Server nimmt die Anmeldung entgegen und konfiguriert den Desktop so, wie es für den User bestimmt ist mit allen Berechtigungen und allen gemounteten Devices.
<b>Endzustand</b>	Der User ist angemeldet und kann den Desktop bedienen.

#### Funktion 3: Abmelden

<b>Auslöser</b>	User
-----------------	------

<b>Voraussetzung</b>	Funktionsfähiger PC +LTSP-Server
<b>Funktionsablauf</b>	User meldet sich ab. Der LTSP-Server speichert dabei die Einstellungen und alle Daten. Die Daten liegen auf dem LTSP-Server.
<b>Endzustand</b>	Der Computer kann heruntergefahren werden (vgl. UseCase „Computer herunterfahren“)

#### Funktion 4: User verwalten

<b>Auslöser</b>	Administrator
<b>Voraussetzung</b>	Funktionsfähiger LTSP-Server
<b>Funktionsablauf</b>	Der Administrator kann die User und Gruppen und alle Berechtigungen und Einstellungen auf dem LTSP-Server verwalten.
<b>Endzustand</b>	Die User sind so konfiguriert, damit diese sich anmelden können und zufrieden arbeiten können.

## 3.5 OpenSSH-Server

- Der OpenSSH-Server ist nach der Installation vom System bereits installiert. Dabei wird die Installation aus dem Software-Repository von Ubuntu verwendet.
- Einsatzzwecke
  - ◆ Fernzugriff via SSH
  - ◆ Dateien kopieren mit SCP

### 3.5.1 Funktionsbeschreibung

- Mit dem OpenSSH-Server kann man den Server über das Netzwerk / Internet von aussen steuern.
- Wenn man per SSH auf den Server zugreift, hat man ein Terminal, das man ganz normal bedienen kann.
- Somit geht das warten einfacher.
- Nötigenfalls kann man X-Anwendungen per X-Forwarding starten.
- Dateien können per SCP einfach kopiert werden.

### 3.5.2 Funktionale Anforderungen

#### Funktion 1: SSH-Fernzugriff

Nach der Installation vom OpenSSH-Server ist diese Funktion aktiv. Man kann ab dann mit einem SSH-Client (z.B. ssh) darauf zugreifen.

Zum Beispiel:

```
ssh username@hostname
```

#### Funktion 2: SCP

Nach der Installation vom OpenSSH-Server ist diese Funktion aktiv. Man kann ab dann mit einem SCP-Client (z.B. scp) darauf zugreifen.

Zum Beispiel:

```
scp .vimrc username@host:~.vimrc
```

#### Funktion 3: Zertifikat basiertes authentifizieren

Mit dieser Funktion kann man sich einfacher am SSH-Server anmelden. Man ist nicht mehr verpflichtet, ein Passwort einzugeben. Es werden zwei Keys generiert. Ein Public und ein Private.

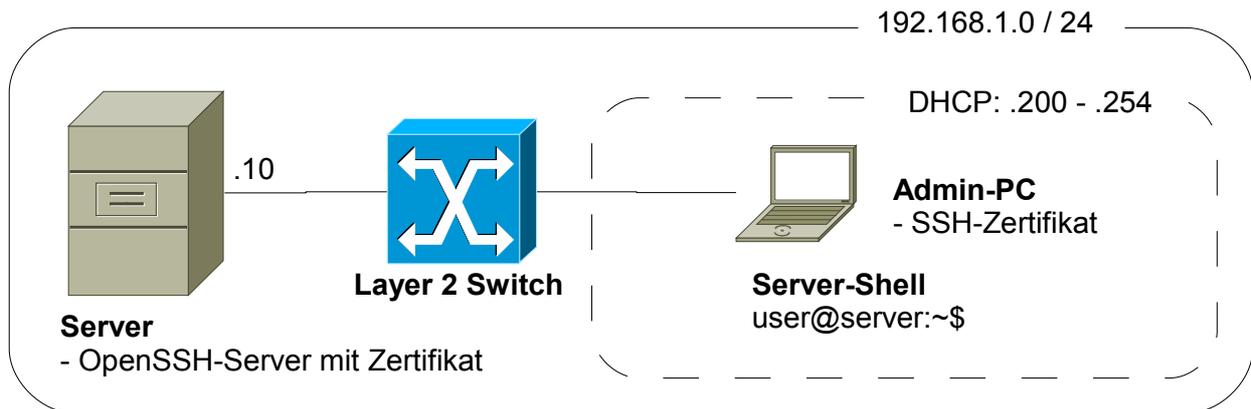
Auf dem SSH-Client erstellt man die zwei Keys und verteilt dann den öffentlichen auf den Server.

Keys: id\_rsa (Private Key) und id\_rsa.pub (Public Key; dieser muss verteilt werden)

### 3.5.3 Use-Cases

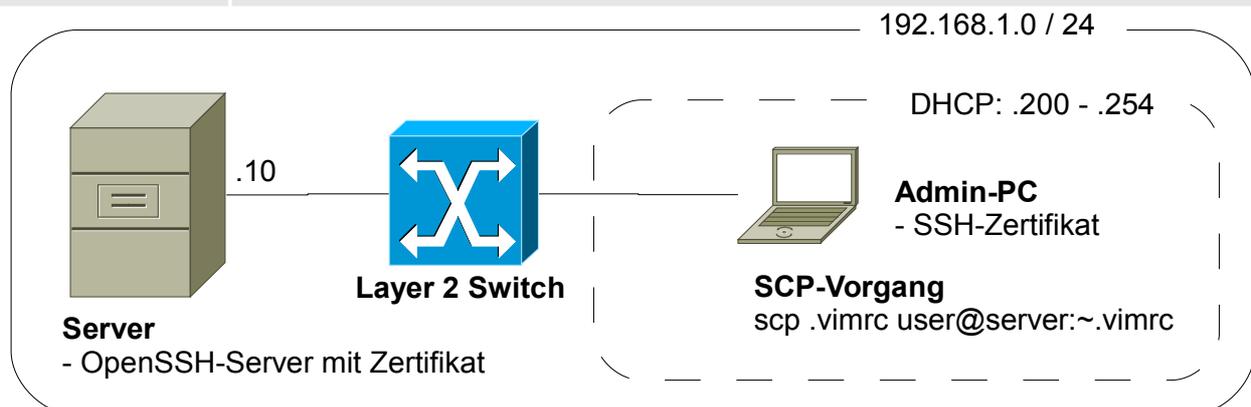
#### Funktion 1: SSH-Zugriff

<b>Auslöser</b>	Administrator
<b>Voraussetzung</b>	OpenSSH-Server und OpenSSH-Client
<b>Funktionsablauf</b>	Administrator verbindet sich mit dem SSH-Client „ssh“ mit dem Server.
<b>Endzustand</b>	Der Administrator sieht eine Shell vom Server.



#### Funktion 2: SCP

<b>Auslöser</b>	Administrator
<b>Voraussetzung</b>	OpenSSH-Server und OpenSSH-Client
<b>Funktionsablauf</b>	Der Administrator kopiert mittels SCP (mit dem SCP-Client „scp“) Dateien auf den Server.
<b>Endzustand</b>	Die Datei ist auf dem Server.



#### Funktion 3: Zertifikat basiertes authentifizieren

<b>Auslöser</b>	Administrator
<b>Voraussetzung</b>	OpenSSH-Server und OpenSSH-Client
<b>Funktionsablauf</b>	Der Administrator will eine Shell über ssh auf dem Server starten oder mit scp eine Datei kopieren. Mit Hilfe des Zertifikats muss er kein Passwort eingeben.
<b>Endzustand</b>	Der Zugriff erfolgt ohne Passwort.

## 3.6 DHCP-Server QUELLE

<http://ltsp.mirrors.tds.net/pub/ltsp/docs/ltsp-4.1-de.pdf>

Der DHCP-Server ist nach der Installation vom System bereits installiert.

### 3.6.1 Funktionsbeschreibung

DHCP muss so eingestellt werden, dass `fixed-address`, `filename`, `subnet-mask`, `broadcast-address` und `root-path` eingetragen sind. Dann wird die Konfigurationsdatei geschrieben.

Jede Workstation benötigt eine automatisch vergebene IP-Adresse. Die Informationen, die von einem DHCP Server vergeben werden können sind vielfältig. Hier eine Auflistung:

- IP-Adresse
- Hostname
- Server-IP-Adresse
- Default Gateway

Während dem Ablauf des `ltspcfg`-Skripts werden die Werte der Konfiguration übertragen. Auch ist es wichtig, dass die DHCP-Anfrage beim PXE-Boot problemlos abläuft, dies ist nur möglich, wenn der Server korrekt läuft.

### 3.6.2 Funktionale Anforderungen

#### Funktion 1

Adressvergabe an Clients die gestartet werden. PXE-Boot-Anfrage beantworten.

#### Funktion 2

Weitere Netzwerkkonfiguration an den Client senden und ihn somit im Netzwerk kommunikationsfähig machen.

### 3.6.3 Use-Case

#### Funktion 1 und 2

<b>Auslöser</b>	User
<b>Voraussetzung</b>	Funktionierender Client und DHCP-Server
<b>Funktionsablauf</b>	Der Bootcode wird in den Speicher geladen die Netzwerkkarte wird gefunden und initialisiert eine DHCP-Anfrage wird ins Netzwerk geschickt der DHCP-Server antwortet--> der Kernel wird vom Server heruntergeladen und gestartet. Der Kernel erkennt die Hardware X startet und ein grafisches Login erscheint auf dem Monitor.
<b>Endzustand</b>	Client ist bereit zum Login.

## 3.7 NBD-Server

Der NBD-Server ist nach der Installation vom System bereits installiert. Es handelt sich hierbei um eine Art virtuelle Festplatte, ausgeschrieben heisst es Network Block Device. Auf diese Festplatte kann via Netzwerk über TCP/IP zugegriffen werden.

### 3.7.1 Funktionsbeschreibung

Die virtuelle Festplatte wird vom NBD-Server bereitgestellt. Der Client stellt eine Verbindung zum Server her und kann die Platte wie ein Lokale bedienen. Um die Funktion aber überhaupt zu ermöglichen, muss der Support der NBD in der Linux-Kernel Konfiguration eingestellt sein. Ein Userspace-Hilfsprogramm namens ndb-client stellt eine TCP Verbindung zum Server her. Es wäre möglich Rechner ohne Festplatte zu nutzen

### 3.7.2 Funktionale Anforderungen

#### Funktion 1

Nachdem die TCP-Verbindung hergestellt wurde, wird man eine Festplatte nutzen können, als ob sie lokal wäre.

### 3.7.3 Use-Cases

#### Funktion 1

Nachdem auf dem Server der ndb-server gestartet wurde, erfolgt der Start des ndb-client und eine Verbindung wird ermöglicht.

<b>Auslöser</b>	User
<b>Voraussetzung</b>	NBD-Dienst gestartet
<b>Funktionsablauf</b>	Nachdem auf dem Server der ndb-server gestartet wurde, erfolgt der Start des ndb-client und eine Verbindung wird ermöglicht.
<b>Endzustand</b>	Client ist bereit zum Login.

### 3.7.4 Weitere Infos / Quellen

[http://de.wikipedia.org/wiki/Network\\_Block\\_Device](http://de.wikipedia.org/wiki/Network_Block_Device)

## 4 TFTP-Server

### 4.1.1 Funktionsbeschreibung

Trivial File Transfer Protocol oder kurz einfach nur TFTP. Dies ist ein sehr rudimentäres Protokoll, welches lediglich für das Schreiben und Lesen von Daten konzipiert ist. Nicht zu verwechseln ist dies mit dem File Transfer Protocol - FTP - denn dieses beherrscht mächtige Funktionen. Diese sind beispielsweise Rechtevergabe und Benutzerauthentifizierung.

TFTP ist bei uns ein essentielles Protokoll, denn die Bootimages werden über dieses Protokoll gesteuert. Die Verteilung des Betriebssystems übernimmt somit TFTP.

#### Funktion 1: Verteilung des Betriebssystems

Dass unsere Dienste später auf der Umgebung alle tadellos funktionieren, müssen wir noch einige Konfigurationen anpassen.

Die Datei `/etc/default/tftpd-hpa` muss wie folgt aussehen:

```
#Defaults for tftpd-hpa
RUN_DAEMON="yes"
OPTIONS="-l -s /var/lib/tftpboot"
```

Zur Bearbeitung der Datei wird ein Editor mit Root-Rechten benötigt. Um dies zu können benötigen wir Rechte, die wir uns also erst holen müssen. Am einfachsten mit dem Terminal und dem Befehl `sudo`.

Diese Zeile `#Defaults for tftpd-hpa` beschreibt die Standardmässigen Einstellungen für diese Datei, durch die Route bzw. Gartenhag am Anfang wird die Zeile auskommentiert. Die Zeile hat also keine eigentliche Funktion, ausser dem Benutzer mitzuteilen, was die Datei ist.

Diese Zeile `RUN_DAEMON="yes"` mit dieser Zeile wird der Dämon gestartet. Hier wäre auch Alternativ ein `"no"` möglich, jedoch benötigen wir den Dämon, nur so können wir TFTP benutzen. TFTP würde auch über `inetd` funktionieren. Doch wir wollen, das dieser "Dienst" immer läuft. Deshalb starten wir ihn als Dämon. >Würden wir dies nicht in der Konfiguration automatisch machen, dann müsste man in der Konsole `sudo /etc/init.d/tftpd-hpa start` von Hand - was sonst - eingeben.

Diese Zeile `OPTIONS="-l -s /var/lib/tftpboot"` setzt noch einige benötigte Optionen. Die Option `"-l"` ist eine Clientoption, welche die spezielle Behandlung vom Doppelpunkt als ein Trennzeichen für Hostnamen überschreibt.

#### Funktion 2: Bereitstellen des Dienstes per Dämon

Damit wird der `tftpd-hpa` als Dämon gestartet. Dies steht allerdings in Widerspruch zur normalen Ubuntu-Konfiguration, in welcher der Dienst über den `inetd` aufgerufen wird. Damit dies zu keinen Konflikten führt, muss man in der Datei `/etc/inetd.conf` die zuständige Zeile mit einem `#` auskommentieren oder vollständig löschen:

```
# tftp dgram udp wait root usr/sbin/in.tftpd /usr/sbin/in.tftpd -s
/var/lib/tftpboot
```

Damit der `inetd` diese Änderung mitbekommt, muss er die Konfiguration neu einlesen:

```
/etc/init.d/openbsd-inetd reload
```

Quellen: <http://rpmfind.net/linux/RPM/opensuse/10.3/i586/tftp-0.48-39.i586.html>

## 4.1.2 Funktionale Anforderungen

### 4.1.3 Use-Cases

#### Funktion 1: Image anbieten

<b>Auslöser</b>	Client
<b>Voraussetzung</b>	Der TFTP-Dämon läuft auf dem Server.
<b>Funktionsablauf</b>	Der Client verbindet sich mit dem TFTP-Server und lädt das Image herunter.
<b>Endzustand</b>	Der Zugriff erfolgt ohne Passwort.

#### Funktion 2: Ständig verfügbar

<b>Auslöser</b>	Administrator
<b>Voraussetzung</b>	Dämon
<b>Funktionsablauf</b>	Der Server bietet dem Client die TFTP-Funktion an. Diese Funktion wird als Dämon laufen, damit diese Funktion immer verfügbar.
<b>Endzustand</b>	Der Dämon läuft

## 5 Glossar

Folgende Begriffe sind für einen Normalsterblichen genauer erklärt:

Begriff	Erklärung
DHCP	Dienst, der die IP-Adressen an die Clients verteilt.
DNS	Löst Namen in IP-Adressen auf. Damit man sich keine „komischen“ IP-Adressen merken muss.
IP-Adresse	Eindeutige Adresse eines Computers in einem Netzwerk. Quasi die Hausnummer eines Pcs.
LTSP	Linux Terminal Server Project: So heisst unsere Hauptsoftware vom Projekt.
NFS	Network File System: Dateien über das Netzwerk organisieren.
PXE	Preboot eXecution Environment: Beim Bootvorgang mit einem Server verbinden.
TFTP	Trivial File Transfer Protocol. Hiermit wird das Minilinux-System heruntergeladen nach dem PXE-Boot
NBD	Ein Network Block Device (engl. für Netzwerk-Blockgerät, abgekürzt NBD) ist eine Art virtuelle Festplatte, auf die ein Rechner via TCP/IP zugreifen kann. Das NBD wird von einem NBD-Server bereitgestellt. Er bietet hierfür eigene Festplatte, Festplattenpartition oder eine Datei als NBD bestimmten anderen Rechnern (Clients) an. Ein anderer Rechner (oder auch der gleiche) kann sich über eine TCP-Verbindung mit dem NBD-Server verbinden und anschließend das NBD wie eine eigene lokale Festplatte benutzen. (Quelle: Wikipedia)
vim	Klassischer und sehr mächtiger Texteditor.

M306

IT-Kleinprojekte abwickeln

Emanuel Duss, Arno Galliker, Semir Jahic

Gruppe: SP4

# Testkonzept

Einführung von LTSP (Linux Terminal Server Project)



**eos opensource solutions**

Projektbezeichnung	Einführung von LTSP (Linux Terminal Server Project)
Projektleiter	Emanuel Duss
Verantwortlich	Projektleiter
Erstellt	15.10.08
Letzte Änderung	2009-04-27 um 12:15:11
Zustand	Zur Prüfung
Pfad	/ media/APACER_2GB/Schule/3_Lehrjahr/306_IT_Kleinprojekt_abwickeln/02_Projekt/ 04_Testkonzept/04_Testkonzept.odt

## Projektmitglieder

Emanuel	Duss	Im Bienz 15	6170 Schüpfheim	emanuel.duss#gmail.com	EDU
Semir	Jahic	Moosmatte 32	6182 Escholzmatt	semir.jahic#gmail.com	SJA
Arno	Galliker	Margrethenstrasse 8	6275 Ballwil	arno.galliker#gmail.com	AGA

## Änderungen

Datum	Version	Kapitel	Beschreibung	Autor
2008-11-26	0.1	ALLE	Grundgerüst vom Dokument	EDU
2009-01-09	0.2	ALLE	Erstellung der Inhalte	ALLE
2009-01-16	0.5	ALLE	Zusammenführung der erstellten Kapitel	ALLE
2009-04-26	1.0	ALLE	Alle Testfälle wurden ergänzt und in Wolhusen besprochen!	ALLE

## Prüfungen / Reviews

Datum	Version	Kapitel	Beschreibung	Autor
	1	Alle		GAS

## Infos

Zuletzt bearbeitet	2009-04-27
Heute	2009-11-23
Bearbeitungszeit	20:27:59
Lehrjahr des Moduls	3. Lehrjahr; 2008 / 2009
Pfad	/ media/APACER_2GB/Schule/3_Lehrjahr/306_IT_Kleinprojekt_abwickeln/02_Projekt/04_Testkonzept/04_Testkonzept.odt

CC-Lizenz



Creative Commons Namensnennung-Keine kommerzielle Nutzung-Weitergabe unter gleichen Bedingungen 2.5 Schweiz

<http://creativecommons.org/licenses/by-nc-sa/2.5/ch/>

Powered by



## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung.....</b>	<b>5</b>
1.1	Test Werkzeuge.....	5
1.2	Zweck des Dokumentes.....	5
1.3	Zielsetzung.....	5
1.4	Referenzierte Dokumente.....	5
1.5	Testwerkzeuge.....	5
<b>2</b>	<b>Testplanung.....</b>	<b>6</b>
2.1	Aktivitätsdiagramm für ein User.....	6
2.2	Systemübersicht.....	7
2.3	Liste der Testfälle.....	8
2.4	Teststufenplan.....	9
2.4.1	Komponenten Tests.....	9
2.4.2	Integrationstest.....	9
2.4.3	Systemtest.....	9
2.4.4	Abnahmetests.....	10
2.5	Beschreibung der Testfälle.....	11
2.5.1	Testfall 1.....	11
2.5.1	Testfall 2.....	11
2.5.2	Testfall 3.....	12
2.5.1	Testfall 4.....	12
2.5.1	Testfall 5.....	12
2.5.2	Testfall 6.....	13
2.5.1	Testfall 7.....	13
2.5.2	Testfall 8.....	13
2.5.3	Testfall 9.....	14
2.5.1	Testfall 10.....	14
2.5.2	Testfall 11.....	14
2.5.3	Testfall 12.....	15
2.5.1	Testfall 13.....	15
2.5.1	Testfall 14.....	15
2.5.1	Testfall 15.....	16
2.5.1	Testfall 16.....	16
2.5.1	Testfall 17.....	16
2.5.2	Testfall 18.....	17
2.5.1	Testfall 19.....	17
2.5.1	Testfall 20.....	17
<b>3</b>	<b>Glossar.....</b>	<b>18</b>

## Abbildungsverzeichnis

Abbildung 1: Aktivitätsdiagramm.....	6
Abbildung 2: Netzwerklayout.....	7

# 1 Einleitung

## 1.1 Test Werkzeuge

Zum Testen stehen uns folgende Werkzeuge zur Verfügung:

- PuTTY
- Wireshark
- Texteditor vim
- SSH-Client ssh
- Netbook Asus EEEPC 1000H zum testen von DHCP
- dhclient

## 1.2 Zweck des Dokumentes

Dieses Dokument stellt das Testkonzept für den Aufbau des Netzwerks der Firma Kuhumus AG dar. In diesem Dokument werden alle durchgeführten Tests und deren Ergebnisse ausführlich und detailliert beschrieben. Die Testabläufe sind klar beschrieben und dokumentiert. So wird klar ersichtlich, was in der Umgebung alles funktionstüchtig ist.

## 1.3 Zielsetzung

- Es geht darum, Fehler festzustellen, die Leistungen gemäss Anforderungsspezifikation beeinträchtigen.
- Dazu wird nach dem Black-Box Testverhalten gearbeitet.
- Aus Zeitgründen werden keine White Box Tests dokumentiert.

## 1.4 Referenzierte Dokumente

Referenzierte Dokumente werden in diesem Dokument nicht mit vollem Namen erwähnt, sondern mit den Zahlen in den eckigen Klammern aus der untenstehenden Liste deklariert. Alle Referenzen, welche im Text verwendet werden, sind in dieser Liste enthalten.

1. Projektauftrag
2. Anforderungsanalyse
3. Detailkonzept

## 1.5 Testwerkzeuge

Sind die Werkzeuge, die für die diversen Tests benutzt werden. Dazu gehören alle benötigten Server und Netzwerkkomponenten, welche zum Einsatz kommen.

## 2 Testplanung

### 2.1 Aktivitätsdiagramm für ein User

Der User soll am Schluss folgende Funktionen nutzen können:

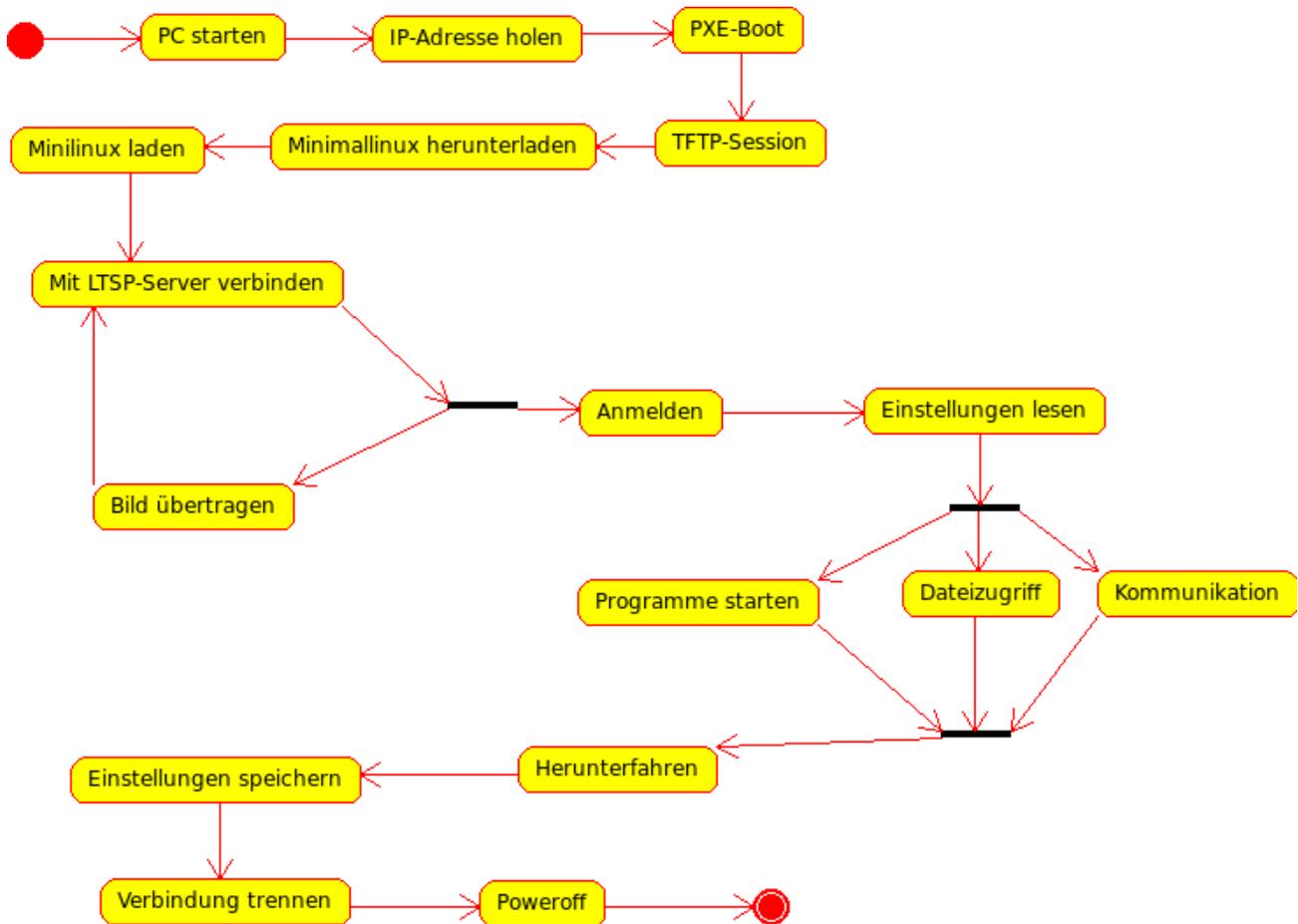


Abbildung 1: Aktivitätsdiagramm

## 2.2 Systemübersicht

Das System sieht am Schluss so aus und soll so funktionieren:

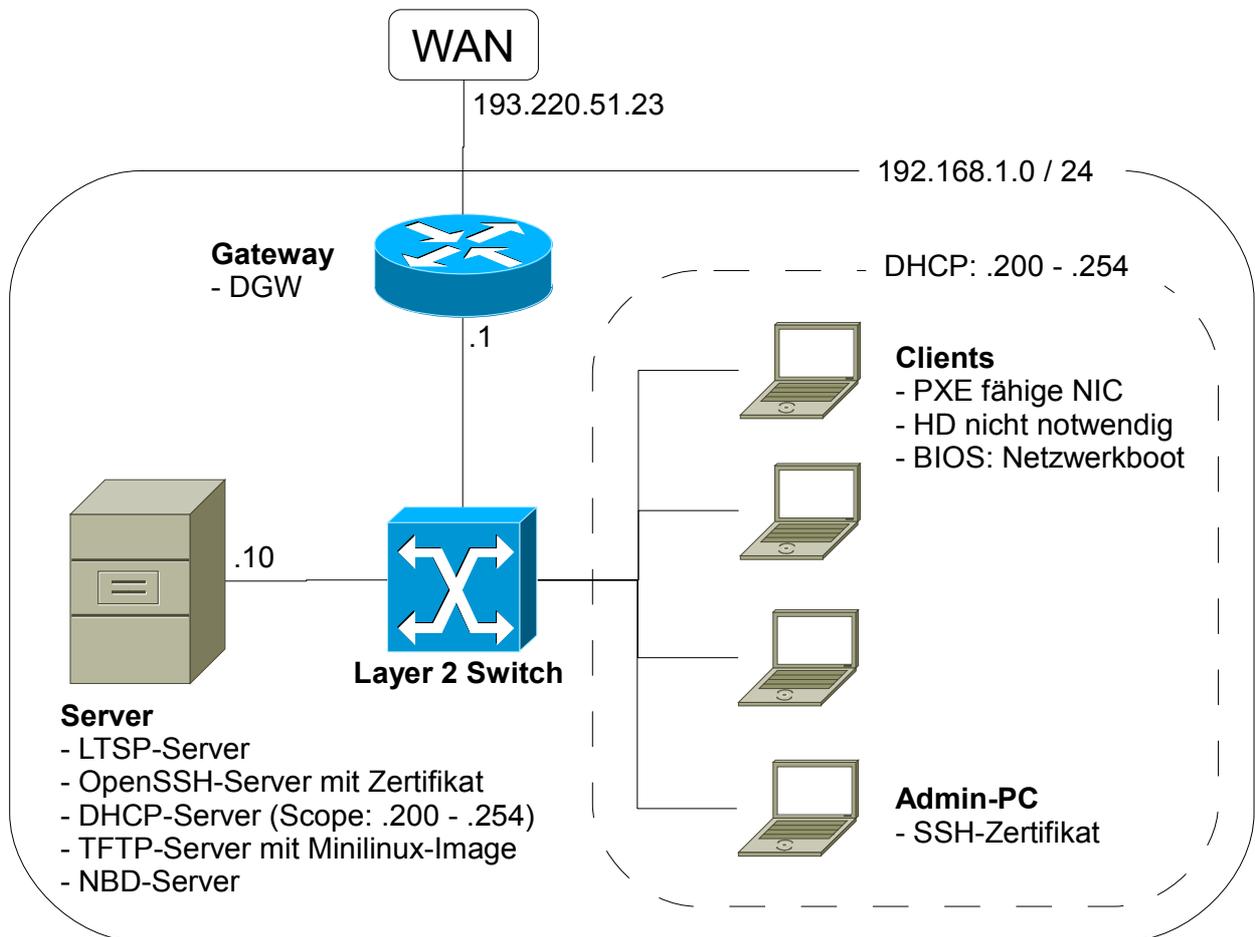


Abbildung 2: Netzwerklayout

## 2.3 Liste der Testfälle

Nr.	Name	Beschreibung
1	Hochfahren der Clients	Der Computer der User soll gestartet werden können.
2	Hochfahren des Servers und starten aller Daemons	Der Server soll eingeschaltet werden können und alle nötigen Daemons werden automatisch gestartet.
3	Computer benutzen	Der User soll den Computer benutzen können.
4	Computer herunterfahren	Der User soll sich wieder abmelden können.
5	Per DHCP eine IP-Adresse bekommen	Beim Bootvorgang wird eine IP-Adresse bezogen.
6	Per TFTP das Minilinux herunterladen	Der DHCP-Server gibt dem PXE-Client den TFTP-Pfad zum Minilinux-Image, welches per TFTP heruntergeladen werden soll.
7	Das Minilinux starten	Das Minilinux wird gestartet bis zum Anmeldebildschirm. Der Loginscreen (LDM LTSP Display Manager) ist ein Teil vom Minilinux.
8	Am Server anmelden	Ein SSH-Tunnel wird aufgebaut und der User kann sich am Server anmelden.
9	Anmeldung entgegennehmen	Der Server nimmt die Authentifikation entgegen und zeigt dem User die grafische Oberfläche an.
10	Vom Server abmelden	Der User kann die Gnome-Sitzung zum Server beenden.
11	User verwalten	Der Administrator kann neue User erfassen.
12	Berechtigungen für die User setzen	Der Administrator kann User speziell berechtigen.
13	Den Server per SSH verwalten	Der Administrator kann sich am Server per SSH anmelden.
14	SCP machen	Der Administrator kann Dateien mit SSH per SCP auf den Server kopieren.
15	Zertifikatbasierte authentifikation (SSH)	Die ganze Administrator-Server-SSH-Geschichte funktioniert mit Zertifikaten, damit es komfortabler funktioniert.
16	PXE-Boot der Clients	Die Clients wählen die Netzwerkkarte als erstes Bootmedium.
17	Audioausgabe bei den Clients	Alle Soundausgaben werden beim User am Client ausgegeben. Das Minilinux setzt dabei die Systemvariablen für den Sound.
18	Geschwindigkeitstest	Die User sollen bei Büroarbeiten ungestört arbeiten können ohne Geschwindigkeitseinbußen.
19	Verwendung von Applikationen	Der User kann Applikationen starten.
20	Verwendung der öffentlichen Shares	Der User kann auf Verzeichnisse auf dem Server zugreifen.

20 Stücke reichen aus um das System auf Magen und Nieren zu testen.

## 2.4 Teststufenplan

### 2.4.1 Komponenten Tests

Nr.	Name	Beschreibung
1	Hochfahren der Clients	Der Computer der User soll gestartet werden können.
2	Hochfahren des Servers und starten aller Daemons	Der Server soll eingeschaltet werden können und alle nötigen Daemons werden automatisch gestartet.
3	Computer benutzen	Der User soll den Computer benutzen können.
4	Computer herunterfahren	Der User soll sich wieder abmelden können.
16	PXE-Boot der Clients	Die Clients wählen die Netzwerkkarte als erstes Bootmedium.
17	Audioausgabe bei den Clients	Alle Soundausgaben werden beim User am Client ausgegeben. Das Minilinux setzt dabei die Systemvariablen für den Sound.
18	Geschwindigkeitstest	Die User sollen bei Büroarbeiten ungestört arbeiten können ohne Geschwindigkeitseinbußen.

### 2.4.2 Integrationstest

Nr.	Name	Beschreibung
5	Per DHCP eine IP-Adresse bekommen	Beim Bootvorgang wird eine IP-Adresse bezogen.
6	Per TFTP das Minilinux herunterladen	Der DHCP-Server gibt dem PXE-Client den TFTP-Pfad zum Minilinux-Image, welches per TFTP heruntergeladen werden soll.
7	Das Minilinux starten	Das Minilinux wird gestartet bis zum Anmeldebildschirm. Der Loginscreen (LDM LTSP Display Manager) ist ein Teil vom Minilinux.

### 2.4.3 Systemtest

Nr.	Name	Beschreibung
8	Am Server anmelden	Ein SSH-Tunnel wird aufgebaut und der User kann sich am Server anmelden.
9	Anmeldung entgegennehmen	Der Server nimmt die Authentifikation entgegen und zeigt dem User die grafische Oberfläche an.
10	Vom Server abmelden	Der User kann die Gnome-Sitzung zum Server beenden.
11	User verwalten	Der Administrator kann neue User erfassen.
12	Berechtigungen für die User setzen	Der Administrator kann User speziell berechtigen.
13	Den Server per SSH verwalten	Der Administrator kann sich am Server per SSH anmelden.
14	SCP machen	Der Administrator kann Dateien mit SSH per SCP auf den Server kopieren.
15	Zertifikatbasierte authentifikation (SSH)	Die ganze Administrator-Server-SSH-Geschichte funktioniert mit Zertifikaten, damit es komfortabler funktioniert.
19	Verwendung von	Der User kann Applikationen starten.

	Applikationen	
20	Verwendung der öffentlichen Shares	Der User kann auf Verzeichnisse auf dem Server zugreifen.

#### 2.4.4 Abnahmetests

Die Abnahmetests werden bei uns nicht durchgeführt, da die Zeit am Zahn nagt... :-)

## 2.5 Beschreibung der Testfälle

Die Testfälle werden anhand der UseCase Tabellen beschrieben. Folgende Informationen sind für Testfälle wichtig:

- Name des UseCase / Testat
- Vorbedingungen
- Testablauf
- Erwartetes Ergebnis.

### 2.5.1 Testfall 1

<b>Nummer</b>	1
<b>Name</b>	Hochfahren der Clients
<b>Beschreibung</b>	Der Computer der User soll gestartet werden können.
<b>Teststufenplan</b>	Komponenten Tests
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• Funktionsfähige Hardware</li> <li>• Geschulte User</li> </ul>
<b>Testablauf</b>	Durch Beobachtung der Benutzer kann man feststellen, ob das Starten für den Benutzer offensichtlich und einfach ist.
<b>Erwartetes Ergebnis</b>	Das Mainboard stellt durch den gedrückten Schalter einen Kurzschluss her, der den Computer aufstarten lässt. Dann durchläuft er den POST im BIOS.

### 2.5.1 Testfall 2

<b>Nummer</b>	2
<b>Name</b>	Hochfahren des Servers und starten aller Deamons
<b>Beschreibung</b>	Der Server soll eingeschaltet werden können und alle nötigen Deamons werden automatisch gestartet.
<b>Teststufenplan</b>	Komponenten Tests
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• Erfolgreicher POST des Computers</li> <li>• Deamons müssen so in der Konfiguration eingestellt sein, dass die Daemons starten</li> </ul>
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>• Überprüfung ob, nach dem Start alle Daemons laufen durch den Befehl</li> <li>• Mit dem Befehl „top“ im Terminal, werden die laufenden Dienste angezeigt, wo auch unsere Daemons drin sein müssen</li> <li>• --&gt; Interessant: Welche Prozesse werden unter top angezeigt, wenn top auf einem LTSP-Client ausgeführt wird?</li> </ul>
<b>Erwartetes Ergebnis</b>	Alle Daemons laufen problemlos nach dem Start.

### 2.5.2 Testfall 3

<b>Nummer</b>	2
<b>Name</b>	Computer benutzen
<b>Beschreibung</b>	Der User soll den Computer benutzen können.
<b>Teststufenplan</b>	Komponenten Tests
<b>Vorbedingungen</b>	Geschulter User Grundkenntnisse über Bedienung von Computern
<b>Testablauf</b>	Ein User versucht den Computer zu benutzen
<b>Erwartetes Ergebnis</b>	Der User hat es geschafft den Computer erfolgreich zu benutzen

### 2.5.1 Testfall 4

<b>Nummer</b>	4
<b>Name</b>	Computer herunterfahren
<b>Beschreibung</b>	Der User soll sich wieder abmelden können.
<b>Teststufenplan</b>	Komponenten Tests
<b>Vorbedingungen</b>	Der User ist angemeldet am Computer Er kann die Grafikoberfläche bedienen
<b>Testablauf</b>	Durch Abmelden auf dem entsprechenden Button auf der Oberfläche, versucht der User sich abzumelden
<b>Erwartetes Ergebnis</b>	Die Abmeldung ist erfolgt und eine Anmeldemaske erscheint auf dem Flüssigkristallbildschirm

### 2.5.1 Testfall 5

<b>Nummer</b>	5
<b>Name</b>	Per DHCP eine IP-Adresse bekommen
<b>Beschreibung</b>	Beim Bootvorgang wird eine IP-Adresse bezogen.
<b>Teststufenplan</b>	Integrationstest
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• Ein DHCP Server ist im Netzwerk erreichbar</li> <li>• Der Computer wurde gestartet</li> <li>• Der Computer ist mit dem Netzwerk verbunden</li> <li>• Der Computer ist auf DHCP-IP-Adressenbezug konfiguriert</li> </ul>
<b>Testablauf</b>	Starten des Computers weiterlaufen lassen
<b>Erwartetes Ergebnis</b>	Eine IP-Adresse erhalten haben

## 2.5.2 Testfall 6

<b>Nummer</b>	6
<b>Name</b>	Per TFTP das Minilinux herunterladen
<b>Beschreibung</b>	Der DHCP-Server gibt dem PXE-Client den TFTP-Pfad zum Minilinux-Image, welches per TFTP heruntergeladen werden soll.
<b>Teststufenplan</b>	Integrationstest
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• Der Server hat ein Minilinux bereit</li> <li>• Übermittlungsbedingungen sind gewährleistet</li> <li>• Der Computer hat eine IP-Adresse erhalten</li> </ul>
<b>Testablauf</b>	Beim Bootvorgang wird das PXE-Image erfolgreich heruntergeladen
<b>Erwartetes Ergebnis</b>	Minilinux auf dem lokalen Rechner, welches erfolgreich bootet

## 2.5.1 Testfall 7

<b>Nummer</b>	7
<b>Name</b>	Das Minilinux starten
<b>Beschreibung</b>	Das Minilinux wird gestartet bis zum Anmeldebildschirm. Der Loginscreen (LDM LTSP Display Manager) ist ein Teil vom Minilinux.
<b>Teststufenplan</b>	Integrationstest
<b>Vorbedingungen</b>	TFTP-Image wurde erfolgreich heruntergeladen und gestartet
<b>Testablauf</b>	Warten nach dem TFTP-Boot, bis der Anmeldebildschirm erscheint
<b>Erwartetes Ergebnis</b>	Korrektes Erscheinungsbild der Anmeldemaske

## 2.5.2 Testfall 8

<b>Nummer</b>	8
<b>Name</b>	Am Server anmelden
<b>Beschreibung</b>	Ein SSH-Tunnel wird aufgebaut und der User kann sich am Server anmelden.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Erfolgreiches Booten und Verbinden zum Server
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>• Den SSH-Tunnel erstellen lassen (automatisch)</li> <li>• Anmelden mit Benutzernamen und Passwort</li> </ul>
<b>Erwartetes Ergebnis</b>	Sicher am Server angemeldet sein

### 2.5.3 Testfall 9

<b>Nummer</b>	9
<b>Name</b>	Am Server anmelden
<b>Beschreibung</b>	Ein SSH-Tunnel wird aufgebaut und der User kann sich am Server anmelden.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Verbindung mit dem Server kann hergestellt werden (Ping, SSH)
<b>Testablauf</b>	Wir melden uns mit einem eingerichteten User an.
<b>Erwartetes Ergebnis</b>	Wir sehen den Gnome-Desktop vor uns.

### 2.5.1 Testfall 10

<b>Nummer</b>	10
<b>Name</b>	Anmeldung entgegennehmen
<b>Beschreibung</b>	Der Server nimmt die Authentifizierung entgegen und zeigt dem User die grafische Oberfläche an.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• SSH-Tunnel aufgebaut</li> <li>• Computer ist korrekt im Netzwerk verbunden</li> </ul>
<b>Testablauf</b>	Während des Anmeldevorgangs warten und die grafische Oberfläche betrachten
<b>Erwartetes Ergebnis</b>	<ul style="list-style-type: none"> <li>• Authentizitätsbestätigung</li> <li>• Oberfläche wird angezeigt</li> </ul>

### 2.5.2 Testfall 11

<b>Nummer</b>	11
<b>Name</b>	User verwalten
<b>Beschreibung</b>	Der Administrator kann neue User erfassen.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Root-Berechtigung auf zentralem Server Physikalischer oder Remote-Zugang zum Server
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● Anmelden am Server (root)</li> <li>● User-Verwaltungstool öffnen und User eintragen</li> <li>● User per Konsole erfassen</li> </ul>
<b>Erwartetes Ergebnis</b>	Der User wurde korrekt erfasst. Die Anmeldung am ThinClient am LTSP-Server funktioniert.

### 2.5.3 Testfall 12

<b>Nummer</b>	12
<b>Name</b>	Berechtigungen für die User setzen
<b>Beschreibung</b>	Der Administrator kann User speziell berechtigen.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Root-Berechtigung auf zentralem Server Physikalischer oder Remote-Zugang zum Server
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● Anmelden am Server (root)</li> <li>● User-Verwaltungstool öffnen und Rechte eintragen oder</li> <li>● User per Konsole editieren und Rechte eintragen</li> </ul>
<b>Erwartetes Ergebnis</b>	Der User kann speziell berechtigt werden und die Berechtigungen haben ihre Wirkung.

### 2.5.1 Testfall 13

<b>Nummer</b>	13
<b>Name</b>	Den Server per SSH verwalten
<b>Beschreibung</b>	Der Administrator kann sich am Server per SSH anmelden.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Der OpenSSH-Dienst ist korrekt installiert und konfiguriert
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● SSH-Verbindung (mit Putty und ssh) auf den Server</li> <li>● Zertifikat des OpenSSH-Servers hinzufügen</li> <li>● mit berechtigtem User einloggen</li> </ul>
<b>Erwartetes Ergebnis</b>	Der Administrator kann eine SSH-Verbindung zum Server aufbauen und sich erfolgreich einloggen.

### 2.5.1 Testfall 14

<b>Nummer</b>	14
<b>Name</b>	SCP machen
<b>Beschreibung</b>	Der Administrator kann Dateien mit SCP per SSH auf den Server kopieren.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Der OpenSSH-Dienst ist korrekt installiert und konfiguriert Ein SCP-Client ist installiert
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● SSH-Verbindung (mit Putty) auf den Server</li> <li>● Zertifikat des OpenSSH-Servers hinzufügen</li> <li>● mit berechtigtem User einloggen</li> <li>● Datei per SCP (Secure Copy Protocol -&gt; Steuerdaten und Dateiübertragung per SSH verschlüsselt!) übertragen: <code>scp Test1.bsp Benutzer@Host:Verzeichnis/Zieldatei.bsp</code></li> </ul>
<b>Erwartetes Ergebnis</b>	Der Administrator kann Dateien remote und verschlüsselt auf den Server übertragen.

### 2.5.1 Testfall 15

<b>Nummer</b>	15
<b>Name</b>	Zertifikatbasierte authentifikation (SSH)
<b>Beschreibung</b>	Die ganze Administrator-Server-SSH-Geschichte funktioniert mit Zertifikaten, damit es komfortabler funktioniert.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Der OpenSSH-Dienst ist korrekt installiert und konfiguriert Ein SCP-Client ist installiert
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● SSH-Verbindung (mit Putty oder ssh) auf den Server</li> <li>● Zertifikat des OpenSSH-Servers hinzufügen</li> <li>● mit berechtigtem User einloggen</li> </ul>
<b>Erwartetes Ergebnis</b>	Beim einloggen per SSH auf den Server muss kein Passwort mehr eingegeben werden.

### 2.5.1 Testfall 16

<b>Nummer</b>	16
<b>Name</b>	PXE-Boot der Clients
<b>Beschreibung</b>	Die Clients wählen die Netzwerkkarte als erstes Bootmedium.
<b>Teststufenplan</b>	Komponenten Tests
<b>Vorbedingungen</b>	Der PC funktioniert einwandfrei und wir gelangen ohne Probleme ins BIOS
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● BIOS des Clients starten</li> <li>● Bootreihenfolge so ändern, dass die Netzwerkkarte an 1. Position steht</li> <li>● Einstellungen speichern und neustarten.</li> </ul>
<b>Erwartetes Ergebnis</b>	Die Clients wählen die Netzwerkkarte als erstes Bootmedium.

### 2.5.1 Testfall 17

<b>Nummer</b>	17
<b>Name</b>	Audioausgabe bei den Clients
<b>Beschreibung</b>	Alle Soundausgaben werden beim User am Client ausgegeben.
<b>Teststufenplan</b>	Komponenten Tests
<b>Vorbedingungen</b>	Das Minilinux ist korrekt konfiguriert und setzt die Systemvariablen für den Sound.
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● Die Lautsprecherausgabe auf eine höhere Stufe stellen (nicht Mute)</li> <li>● Clientseitig eine Sounddatei öffnen und lauschen</li> </ul>
<b>Erwartetes Ergebnis</b>	Die Frequenzen erklingen über die Lautsprecher.

## 2.5.2 Testfall 18

<b>Nummer</b>	18
<b>Name</b>	Geschwindigkeitstest
<b>Beschreibung</b>	Die User sollen bei Büroarbeiten ungestört arbeiten können ohne Geschwindigkeitseinbussen.
<b>Teststufenplan</b>	Komponenten Tests
<b>Vorbedingungen</b>	Die Terminalsession ist auf die Geschwindigkeit optimiert.
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● Wir öffnen den Internetbrowser und einen Texteditor</li> <li>● Wir können wie gewohnt ohne grosse Verzögerungen multitaskfähig arbeiten.</li> </ul>
<b>Erwartetes Ergebnis</b>	Verzögerungsfreies Arbeiten (mit gewissen Einschränkungen wie z.B. Videowiedergabe) ist möglich.

## 2.5.1 Testfall 19

<b>Nummer</b>	19
<b>Name</b>	Verwendung von Applikationen
<b>Beschreibung</b>	Der User kann Applikationen starten.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Die Terminalsession lässt sich öffnen und die Benutzeroberfläche erscheint.
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● Wir öffnen den Internetbrowser und einen Texteditor</li> <li>● Die Applikationen starten auf und funktionieren</li> </ul>
<b>Erwartetes Ergebnis</b>	Der User kann Applikationen starten.

## 2.5.1 Testfall 20

<b>Nummer</b>	20
<b>Name</b>	Verwendung der öffentlichen Shares
<b>Beschreibung</b>	Der User kann auf Verzeichnisse auf dem Server zugreifen.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Die User-Shares sind korrekt freigeben und die Berechtigungen korrekt gesetzt.
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● Wir öffnen als User clientseitig einen Share (mounten) cd /media/share/transfer ls</li> <li>● Wir können Dateien lesen und schreiben</li> </ul>
<b>Erwartetes Ergebnis</b>	Der User kann auf Dateien und Verzeichnisse auf dem zentralen Server zugreifen.

### 3 Glossar

Folgende Begriffe sind für einen Normalsterblichen genauer erklärt:

Begriff	Erklärung
DHCP	Dienst, der die IP-Adressen an die Clients verteilt.
DNS	Löst Namen in IP-Adressen auf. Damit man sich keine „komischen“ IP-Adressen merken muss.
IP-Adresse	Eindeutige Adresse eines Computers in einem Netzwerk. Quasi die Hausnummer eines Pcs.
LTSP	Linux Terminal Server Project: So heisst unsere Hauptsoftware vom Projekt.
NFS	Network File System: Dateien über das Netzwerk organisieren.
PXE	Preboot eXecution Environment: Beim Bootvorgang mit einem Server verbinden.
TFTP	Trivial File Transfer Protocol. Hiermit wird das Minilinux-System heruntergeladen nach dem PXE-Boot
NBD	Ein Network Block Device (engl. für Netzwerk-Blockgerät, abgekürzt NBD) ist eine Art virtuelle Festplatte, auf die ein Rechner via TCP/IP zugreifen kann. Das NBD wird von einem NBD-Server bereitgestellt. Er bietet hierfür eigene Festplatte, Festplattenpartition oder eine Datei als NBD bestimmten anderen Rechnern (Clients) an. Ein anderer Rechner (oder auch der gleiche) kann sich über eine TCP-Verbindung mit dem NBD-Server verbinden und anschließend das NBD wie eine eigene lokale Festplatte benutzen. (Quelle: Wikipedia)
vim	Klassischer und sehr mächtiger Texteditor.
SCP	Secure Copy Protocol (Abk. SCP) ist ein Protokoll sowie ein Programm zur verschlüsselten Übertragung von Daten zwischen zwei Computern über ein Rechnernetz.

M306

IT-Kleinprojekte abwickeln

Emanuel Duss, Arno Galliker, Semir Jahic

Gruppe: SP4

# Arbeitsjournal

Einführung von LTSP (Linux Terminal Server Project)



**eos opensource solutions**

Projektbezeichnung	Einführung von LTSP (Linux Terminal Server Project)
Projektleiter	Emanuel Duss
Verantwortlich	Projektleiter
Erstellt	15.10.08
Letzte Änderung	2009-05-08 um 09:53:42
Zustand	Zur Prüfung
Pfad	/ media/APACER_2GB/Schule/3_Lehrjahr/306_IT_Kleinprojekt_abwickeln/02_Projekt/ 05_Arbeitsjournal/05_Arbeitsjournal.odt

## Projektmitglieder

Emanuel	Duss	Im Bienz 15	6170 Schüpfheim	emanuel.duss@gmail.com	EDU
Semir	Jahic	Moosmatte 32	6182 Escholzmatt	semir.jahic@gmail.com	SJA
Arno	Galliker	Margrethenstrasse 8	6275 Ballwil	arno.galliker@gmail.com	AGA

## Änderungen

Datum	Version	Kapitel	Beschreibung	Autor
2008-11-26	0.1	ALLE	Grundgerüst vom Dokument	EDU
2009-01-09	0.2	ALLE	Erstellung der Inhalte	ALLE
2009-01-16	0.5	ALLE	Zusammenführung der erstellten Kapitel	ALLE
2009-04-26	1.0	ALLE	Alle Testfälle wurden ergänzt und in Wolhusen besprochen!	ALLE

## Prüfungen / Reviews

Datum	Version	Kapitel	Beschreibung	Autor
	1	Alle		GAS

## Infos

Zuletzt bearbeitet	2009-05-08
Heute	2009-11-23
Bearbeitungszeit	00:12:12
Lehrjahr des Moduls	3. Lehrjahr; 2008 / 2009
Pfad	/ media/APACER_2GB/Schule/3_Lehrjahr/306_IT_Kleinprojekt_abwickeln/02_Projekt/05_Arbeitsjournal/05_Arbeitsjournal.odt

CC-Lizenz



Creative Commons Namensnennung-Keine kommerzielle Nutzung-Weitergabe unter gleichen Bedingungen 2.5 Schweiz

<http://creativecommons.org/licenses/by-nc-sa/2.5/ch/>

Powered by



## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung.....</b>	<b>5</b>
1.1	Referenzierte Dokumente.....	5
<b>2</b>	<b>Zeitplanung Soll.....</b>	<b>6</b>
<b>3</b>	<b>Zeitplanung Ist.....</b>	<b>7</b>
<b>4</b>	<b>Glossar.....</b>	<b>10</b>

## Abbildungsverzeichnis

Abbildung 1: Aktivitätsdiagramm.....	6
Abbildung 2: Netzwerklayout.....	7

# 1 Einleitung

Unser Arbeitjournal enthält folgende Teile:

- Zeitplan
- Zeiterfassung
- Soll / Ist Vergleich und Planungsanpassungen
- Reflexion

## 1.1 Referenzierte Dokumente

Referenzierte Dokumente werden in diesem Dokument nicht mit vollem Namen erwähnt, sondern mit den Zahlen in den eckigen Klammern aus der untenstehenden Liste deklariert. Alle Referenzen, welche im Text verwendet werden, sind in dieser Liste enthalten.

1. Projektauftrag
2. Anforderungsanalyse
3. Detailkonzept
4. Testkonzept

## 2 Zeitplanung Soll

Datum	Uhrzeit	Dauer [h]	Vorgang
04.05.09	08.00-10.00	2	Installation Grundsystem LTSP
04.05.09	10.00-12.00	2	Einrichten Client, Bootoption
04.05.09	10.00-12.00	2	Einrichten Server, Dienste
04.05.09	13.00-15.00	2	Booten von Client überprüfen
04.05.09	15.00-16.30	2.5	Dokumentation erstellen und aktualisieren
06.05.09	08.00-10.00	2	Gemäss Testkonzept Integrationsprüfung durchführen und die Testfälle durchgehen (Testfälle 5,6,7)
06.05.09	10.00-14.00	4	Gemäss Testkonzept Komponententests durchführen und die Testfälle durchgehen (Testfälle 1-4, 16-18)
06.05.09	14.00-16.30	2.5	Dokumentation aktualisieren
07.05.09	08.00-12.00	3	Gemäss Testkonzept Systemtests durchführen und die Testfälle durchgehen (Testfälle 8-20)
07.05.09	12.00-16.30	4.5	Dokumentation aktualisieren
08.05.09	08.00-12.00	4	Experimentieren mit Softwarepakete etc./Präsentation erstellen
08.05.09	12.00-14.00	2	Dokumentation aktualisieren
08.05.09	14.00-16.30	2.5	Präsentation

### 3 Zeitplanung Ist

Datum	Uhrzeit	Dauer [h]	Differenz [min] +=länger, -=kürzer	Bemerkung	Vorgang
04.05.09	08.00-10.00	2	0	Ja, Grundsystem erfolgreich mit Ubuntu 9.04 aufgesetzt und die Installation wurde auf den aktuellsten Stand gebracht.	Installation Grundsystem LTSP
04.05.09	10.00-12.00	2	30	Clients wurden eingerichtet, jedoch wurde nicht immer ordnungsgemäss davon gebootet. Es stellte sich heraus, dass es höchst wahrscheinlich an der Hardware liegt.	Einrichten Client, Bootoption
04.05.09	10.00-12.00	2	30	Mit dem Einrichten des Server hat nicht alles so reibungslos funktioniert wie erwartet. Probleme gab es hauptsächlich mit dem DHCP-Server. Der Zeitplan wurde hier um etwa 30 Minuten nicht eingehalten überschritten.	Einrichten Server, Dienste
04.05.09	13.00-15.00	2	-60	Anmeldung gab es anfänglich kleinere Probleme, nachdem die Benutzer jedoch eingerichtet waren und wir den Server einmal neu gestartet haben, hat alles reibungslos funktioniert Die Verspätung beim Einrichten des Servers, wurde hier somit wieder wett gemacht.	Booten von Client überprüfen
04.05.09	15.00-16.30	2.5	150	Wir haben noch mit den Clients diverser ausprobiert und getestet, um zu sehen, was alles auf so einem Mini-Linux möglich ist, so haben wir die Dokumentation nicht gross angefangen. Lediglich das Grunddokument, welches es zu füllen gilt, wurde erstellt.	Dokumentation erstellen und aktualisieren
06.05.09	08.00-10.00	2	-20	Das Booten und sich anmelden hat wunderbar funktioniert und das PXE-Image wurde wie erwartet heruntergeladen und gestartet. Da dies geklappt hat, gehen wir davon aus, dass die IP-Adresse erfolgreich bezogen wurde. Dies kontrollieren wir nach der ersten Anmeldung und stellten das Erwartete auch fest.	Gemäss Testkonzept Integrationsprüfung durchführen und die Testfälle durchgehen (Testfälle 5,6,7)
06.05.09	10.00-14.00	4	60	Grundsätzlich hat das Booten wunderbar funktioniert. Leider sind die Computer etwas	Gemäss Testkonzept

				<p>widerspenstig, was den Bezug der IP-Adresse und des PXE-Images angeht. Dies fällt aber nur bei einem Client auf, somit schliessen wir daraus, dass es am Client selbst und nicht an unserer Einrichtung liegt.</p> <p>In dieser Zeit haben wir auch ein Login-Screen erstellt, der wie der aus unserem Projekt aussieht, dieser soll dann erscheinen, wenn der Client gestartet ist. Das Einbetten werden wir später machen.</p> <p>Hier wollten wir auch kurz das Fernwartungstool testen, welches standardmässig im Ubuntu-LTSP integriert ist, leider hat nach etlichen versuchen, keine Verbindung zustande kommen wollen. VNC hat hier leider versagt. So mussten wir uns überlegen, wie das Problem zu meistern wäre und haben die Anleitungen im Internet durchforstet und gedacht, da alle diese auf Ubuntu 8.10 basieren, unseren Server neu aufzusetzen und es damit nochmal zu versuchen.</p> <p>Die im Testfall 18 beschriebene Prüfung der Geschwindigkeit verlief äusserst erfolgreich. Die Geschwindigkeit war sehr hoch und proportional hoch zu dieser auch unser Erstaunen, wie gut LTSP wirklich funktioniert.</p>	Komponententests durchführen und die Testfälle durchgehen (Testfälle 1-4, 16-18)
06.05.09	14.00-16.30	2.5	-10	<p>Um ca. 15.15, also nach der Pause wurde intensiv an der Dokumentation unseres Projekts gearbeitet, darin wurden die bisherigen Schritte, die wir gemacht haben festgehalten. Währenddessen wurde unser Server neu installiert und vorbereitet, die Einstellungen haben wir vom vorherigen Server übernommen und aus der Doku auslesen können.</p>	Dokumentation aktualisieren
07.05.09	08.00-12.00	3	-60	<p>Der am Vortag entdeckte Fehler in Ubuntu 9.04 wurde gemeldet und zwar direkt im offiziellen Forum für Bugreports in Ubuntudistributionen.</p> <p>Am Server kann man sich erfolgreich anmelden und die Authentifikation funktioniert makellos. Zuvor wurden auf dem Server im Terminal Pseudobnutzer eingerichtet. Diese haben wir Bob und Lea genannt. Bob und Lea waren fähig, sich problemlos am Server anzumelden.</p> <p>Die Clients können abgemeldet und neugestartet werden, die verläuft auch ohne Hürden. Die Benutzer wurden in verschiedene Gruppen importiert und können nun auch so verwaltet werden.</p> <p>Währenddessen wurden per SSH mit SCP einige Dateien auf den Server übertragen. Dies funktionierte wie in den Testfällen beschrieben und verursachte keine Zeitverzögerungen. Des weiteren kann man sogar von den Windows-Rechnern im gleichen Subnetz, dass wir erstellt haben, um leichter zu arbeiten, auch Befehle auf dem Server ausführen und beispielsweise Verzeichnisse erstellen oder weitere Benutzer anlegen.</p>	Gemäss Testkonzept Systemtests durchführen und die Testfälle durchgehen (Testfälle 8-20)
07.05.09	12.00-16.30	4.5	0	<p>Mit der Doku konnten einige Gruppenmitglieder weiterfahren, auch wurde das Image in dieser Zeit noch angepasst, um so den neu gestalteten Login-Screen darin einzubetten. Als</p>	Dokumentation

				nächstes werden auch noch Applikationen für Benutzer eingebettet. Auch musste ein funktionsfähiges Administrationsprogramm gefunden werden, um die Clients zu überwachen und Präsentationen durchführen zu können. Wir haben uns hier für iTalc entschieden und nach einigen Einstellungen, wie den Keys und so weiter, hat das Programm wunderbar funktioniert.	aktualisieren
08.05.09	08.00-12.00	4	-45	Nachdem wir die Testfälle in den vorherigen Tagen relativ gut bearbeiten konnten und uns Probleme etwas aufgehalten haben, hat sich das Delta im Projektzeitplan relativiert. Nun schreibe ich das Arbeitsjournal und der Server läuft. Die Präsentation wir jetzt noch erstellt und geplant. Es ist eine Live-Demo, die Clients zeigt, die gestartet werden und überwacht werden können.	Experimentieren mit Softwarepakete etc./Präsentation erstellen
08.05.09	12.00-14.00	2		ja	Dokumentation aktualisieren
08.05.09	14.00-16.30	2.5		ja	Präsentation

## 4 Glossar

Folgende Begriffe sind für einen Normalsterblichen genauer erklärt:

Begriff	Erklärung
DHCP	Dienst, der die IP-Adressen an die Clients verteilt.
DNS	Löst Namen in IP-Adressen auf. Damit man sich keine „komischen“ IP-Adressen merken muss.
IP-Adresse	Eindeutige Adresse eines Computers in einem Netzwerk. Quasi die Hausnummer eines Pcs.
LTSP	Linux Terminal Server Project: So heisst unsere Hauptsoftware vom Projekt.
NFS	Network File System: Dateien über das Netzwerk organisieren.
PXE	Preboot eXecution Environment: Beim Bootvorgang mit einem Server verbinden.
TFTP	Trivial File Transfer Protocol. Hiermit wird das Minilinux-System heruntergeladen nach dem PXE-Boot
NBD	Ein Network Block Device (engl. für Netzwerk-Blockgerät, abgekürzt NBD) ist eine Art virtuelle Festplatte, auf die ein Rechner via TCP/IP zugreifen kann. Das NBD wird von einem NBD-Server bereitgestellt. Er bietet hierfür eigene Festplatte, Festplattenpartition oder eine Datei als NBD bestimmten anderen Rechnern (Clients) an. Ein anderer Rechner (oder auch der gleiche) kann sich über eine TCP-Verbindung mit dem NBD-Server verbinden und anschließend das NBD wie eine eigene lokale Festplatte benutzen. (Quelle: Wikipedia)
vim	Klassischer und sehr mächtiger Texteditor.
SCP	Secure Copy Protocol (Abk. SCP) ist ein Protokoll sowie ein Programm zur verschlüsselten Übertragung von Daten zwischen zwei Computern über ein Rechnernetz.

M306

IT-Kleinprojekte abwickeln

Emanuel Duss, Arno Galliker, Semir Jahic

Gruppe: SP4

# Testprotokoll

Einführung von LTSP (Linux Terminal Server Project)



**eos opensource solutions**

Projektbezeichnung	Einführung von LTSP (Linux Terminal Server Project)
Projektleiter	Emanuel Duss
Verantwortlich	Projektleiter
Erstellt	15.10.08
Letzte Änderung	2009-05-08 um 15:34:09
Zustand	Zur Prüfung
Pfad	/ media/APACER_2GB/Schule/3_Lehrjahr/306_IT_Kleinprojekt_abwickeln/02_Projekt/

	06_Testprotokoll/06_Testprotokoll.odt
--	---------------------------------------

## Projektmitglieder

Emanuel	Duss	Im Bienz 15	6170 Schüpfheim	emanuel.duss#gmail.com	EDU
Semir	Jahic	Moosmatte 32	6182 Escholzmatt	semir.jahic#gmail.com	SJA
Arno	Galliker	Margrethenstrasse 8	6275 Ballwil	arno.galliker#gmail.com	AGA

## Änderungen

Datum	Version	Kapitel	Beschreibung	Autor
2008-11-26	0.1	ALLE	Grundgerüst vom Dokument	EDU
2009-01-09	0.2	ALLE	Erstellung der Inhalten	ALLE

## Prüfungen / Reviews

Datum	Version	Kapitel	Beschreibung	Autor
	1	Alle		GAS

## Infos

Zuletzt bearbeitet	2009-05-08
Heute	2009-11-23
Bearbeitungszeit	20:39:52
Lehrjahr des Moduls	3. Lehrjahr; 2008 / 2009
Pfad	/ media/APACER_2GB/Schule/3_Lehrjahr/306_IT_Kleinprojekt_abwickeln/02_Projekt/06_Testprotokoll/06_Testprotokoll.odt

CC-Lizenz



Creative Commons Namensnennung-Keine kommerzielle Nutzung-Weitergabe unter gleichen Bedingungen 2.5 Schweiz

<http://creativecommons.org/licenses/by-nc-sa/2.5/ch/>

Powered by



## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung.....</b>	<b>5</b>
1.1	Test Werkzeuge.....	5
1.2	Zweck des Dokumentes.....	5
1.3	Zielsetzung.....	5
1.4	Referenzierte Dokumente.....	5
1.5	Liste der Testfälle.....	6
1.6	Teststufenplan.....	7
1.6.1	Komponenten Tests.....	7
1.6.2	Integrationstest.....	7
1.6.3	Systemtest.....	7
1.7	Beschreibung der Testfälle.....	9
1.7.1	Testfall 1.....	9
1.7.1	Testfall 2.....	9
1.7.2	Testfall 3.....	10
1.7.1	Testfall 4.....	10
1.7.1	Testfall 5.....	10
1.7.2	Testfall 6.....	11
1.7.1	Testfall 7.....	11
1.7.2	Testfall 8.....	11
1.7.3	Testfall 9.....	12
1.7.1	Testfall 10.....	12
1.7.2	Testfall 11.....	12
1.7.3	Testfall 12.....	13
1.7.1	Testfall 13.....	13
1.7.1	Testfall 14.....	13
1.7.1	Testfall 15.....	14
1.7.1	Testfall 16.....	14
1.7.1	Testfall 17.....	14
1.7.2	Testfall 18.....	15
1.7.1	Testfall 19.....	15
1.7.1	Testfall 20.....	15
<b>2</b>	<b>Glossar.....</b>	<b>16</b>

## Abbildungsverzeichnis

Abbildung 1: Aktivitätsdiagramm.....	6
Abbildung 2: Netzwerklayout.....	7

# 1 Einleitung

## 1.1 Test Werkzeuge

Zum Testen stehen uns folgende Werkzeuge zur Verfügung:

- PuTTY
- Wireshark
- Texteditor vim
- SSH-Client ssh
- Netbook Asus EEEPC 1000H zum testen von DHCP
- dhclient

## 1.2 Zweck des Dokumentes

Dieses Dokument stellt das Testprotokoll dar.

## 1.3 Zielsetzung

Das Testprotokoll überprüft die im Testkonzept erstellten Forderungen.

## 1.4 Referenzierte Dokumente

Referenzierte Dokumente werden in diesem Dokument nicht mit vollem Namen erwähnt, sondern mit den Zahlen in den eckigen Klammern aus der untenstehenden Liste deklariert. Alle Referenzen, welche im Text verwendet werden, sind in dieser Liste enthalten.

1. Projektauftrag
2. Anforderungsanalyse
3. Testkonzept

## 1.5 Liste der Testfälle

Nr.	Name	Beschreibung
1	Hochfahren der Clients	Der Computer der User soll gestartet werden können.
2	Hochfahren des Servers und starten aller Daemons	Der Server soll eingeschaltet werden können und alle nötigen Daemons werden automatisch gestartet.
3	Computer benutzen	Der User soll den Computer benutzen können.
4	Computer herunterfahren	Der User soll sich wieder abmelden können.
5	Per DHCP eine IP-Adresse bekommen	Beim Bootvorgang wird eine IP-Adresse bezogen.
6	Per TFTP das Minilinux herunterladen	Der DHCP-Server gibt dem PXE-Client den TFTP-Pfad zum Minilinux-Image, welches per TFTP heruntergeladen werden soll.
7	Das Minilinux starten	Das Minilinux wird gestartet bis zum Anmeldebildschirm. Der Loginscreen (LDM LTSP Display Manager) ist ein Teil vom Minilinux.
8	Am Server anmelden	Ein SSH-Tunnel wird aufgebaut und der User kann sich am Server anmelden.
9	Anmeldung entgegennehmen	Der Server nimmt die Authentifikation entgegen und zeigt dem User die grafische Oberfläche an.
10	Vom Server abmelden	Der User kann die Gnome-Sitzung zum Server beenden.
11	User verwalten	Der Administrator kann neue User erfassen.
12	Berechtigungen für die User setzen	Der Administrator kann User speziell berechtigen.
13	Den Server per SSH verwalten	Der Administrator kann sich am Server per SSH anmelden.
14	SCP machen	Der Administrator kann Dateien mit SSH per SCP auf den Server kopieren.
15	Zertifikatbasierte authentifikation (SSH)	Die ganze Administrator-Server-SSH-Geschichte funktioniert mit Zertifikaten, damit es komfortabler funktioniert.
16	PXE-Boot der Clients	Die Clients wählen die Netzwerkkarte als erstes Bootmedium.
17	Audioausgabe bei den Clients	Alle Soundausgaben werden beim User am Client ausgegeben. Das Minilinux setzt dabei die Systemvariablen für den Sound.
18	Geschwindigkeitstest	Die User sollen bei Büroarbeiten ungestört arbeiten können ohne Geschwindigkeitseinbußen.
19	Verwendung von Applikationen	Der User kann Applikationen starten.
20	Verwendung der öffentlichen Shares	Der User kann auf Verzeichnisse auf dem Server zugreifen.

20 Stücke reichen aus um das System auf Magen und Nieren zu testen.

## 1.6 Teststufenplan

### 1.6.1 Komponenten Tests

Nr.	Name	Beschreibung
1	Hochfahren der Clients	Der Computer der User soll gestartet werden können.
2	Hochfahren des Servers und starten aller Daemons	Der Server soll eingeschaltet werden können und alle nötigen Daemons werden automatisch gestartet.
3	Computer benutzen	Der User soll den Computer benutzen können.
4	Computer herunterfahren	Der User soll sich wieder abmelden können.
16	PXE-Boot der Clients	Die Clients wählen die Netzwerkkarte als erstes Bootmedium.
17	Audioausgabe bei den Clients	Alle Soundausgaben werden beim User am Client ausgegeben. Das Minilinux setzt dabei die Systemvariablen für den Sound.
18	Geschwindigkeitstest	Die User sollen bei Büroarbeiten ungestört arbeiten können ohne Geschwindigkeitseinbußen.

### 1.6.2 Integrationstest

Nr.	Name	Beschreibung
5	Per DHCP eine IP-Adresse bekommen	Beim Bootvorgang wird eine IP-Adresse bezogen.
6	Per TFTP das Minilinux herunterladen	Der DHCP-Server gibt dem PXE-Client den TFTP-Pfad zum Minilinux-Image, welches per TFTP heruntergeladen werden soll.
7	Das Minilinux starten	Das Minilinux wird gestartet bis zum Anmeldebildschirm. Der Loginscreen (LDM LTSP Display Manager) ist ein Teil vom Minilinux.

### 1.6.3 Systemtest

Nr.	Name	Beschreibung
8	Am Server anmelden	Ein SSH-Tunnel wird aufgebaut und der User kann sich am Server anmelden.
9	Anmeldung entgegennehmen	Der Server nimmt die Authentifikation entgegen und zeigt dem User die grafische Oberfläche an.
10	Vom Server abmelden	Der User kann die Gnome-Sitzung zum Server beenden.
11	User verwalten	Der Administrator kann neue User erfassen.
12	Berechtigungen für die User setzen	Der Administrator kann User speziell berechtigen.
13	Den Server per SSH verwalten	Der Administrator kann sich am Server per SSH anmelden.
14	SCP machen	Der Administrator kann Dateien mit SSH per SCP auf den Server kopieren.
15	Zertifikatbasierte authentifikation (SSH)	Die ganze Administrator-Server-SSH-Geschichte funktioniert mit Zertifikaten, damit es komfortabler funktioniert.
19	Verwendung von	Der User kann Applikationen starten.

	Applikationen	
20	Verwendung der öffentlichen Shares	Der User kann auf Verzeichnisse auf dem Server zugreifen.

## 1.7 Beschreibung der Testfälle

Die Testfälle werden anhand der UseCase Tabellen beschrieben. Folgende Informationen sind für Testfälle wichtig:

- Name des UseCase / Testat
- Vorbedingungen
- Testablauf
- Erreichtes Ergebnis.

### 1.7.1 Testfall 1

<b>Nummer</b>	1
<b>Name</b>	Hochfahren der Clients
<b>Beschreibung</b>	Der Computer der User soll gestartet werden können.
<b>Teststufenplan</b>	Komponenten Tests
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• Funktionsfähige Hardware</li> <li>• Geschulte User</li> </ul>
<b>Testablauf</b>	Durch Beobachtung der Benutzer kann man feststellen, ob das Starten für den Benutzer offensichtlich und einfach ist.
<b>Erreichtes Ergebnis</b>	Das Mainboard stellt durch den gedrückten Schalter einen Kurzschluss her, der den Computer aufstarten lässt. Dann durchläuft er den POST im BIOS.

### 1.7.1 Testfall 2

<b>Nummer</b>	2
<b>Name</b>	Hochfahren des Servers und starten aller Deamons
<b>Beschreibung</b>	Der Server soll eingeschaltet werden können und alle nötigen Deamons werden automatisch gestartet.
<b>Teststufenplan</b>	Komponenten Tests
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• Erfolgreicher POST des Computers</li> <li>• Deamons müssen so in der Konfiguration eingestellt sein, dass die Deamons starten</li> </ul>
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>• Überprüfung ob, nach dem Start alle Daemons laufen durch den Befehl</li> <li>• Mit dem Befehl „top“ im Terminal, werden die laufenden Dienste angezeigt, wo auch unsere Daemons drin sein müssen</li> <li>• --&gt; Interessant: Welche Prozesse werden unter top angezeigt, wenn top auf einem LTSP-Client ausgeführt wird?</li> </ul>
<b>Erreichtes Ergebnis</b>	Alle Daemons laufen problemlos nach dem Start.

### 1.7.2 Testfall 3

<b>Nummer</b>	2
<b>Name</b>	Computer benutzen
<b>Beschreibung</b>	Der User soll den Computer benutzen können.
<b>Teststufenplan</b>	Komponenten Tests
<b>Vorbedingungen</b>	Geschulter User Grundkenntnisse über Bedienung von Computern
<b>Testablauf</b>	Ein User versucht den Computer zu benutzen
<b>Erreichtes Ergebnis</b>	Der User hat es geschafft den Computer erfolgreich zu benutzen

### 1.7.1 Testfall 4

<b>Nummer</b>	4
<b>Name</b>	Computer herunterfahren
<b>Beschreibung</b>	Der User soll sich wieder abmelden können.
<b>Teststufenplan</b>	Komponenten Tests
<b>Vorbedingungen</b>	Der User ist angemeldet am Computer Er kann die Grafikoberfläche bedienen
<b>Testablauf</b>	Durch Abmelden auf dem entsprechenden Button auf der Oberfläche, versucht der User sich abzumelden
<b>Erreichtes Ergebnis</b>	Die Abmeldung ist erfolgt und eine Anmeldemaske erscheint auf dem Flüssigkristallbildschirm

### 1.7.1 Testfall 5

<b>Nummer</b>	5
<b>Name</b>	Per DHCP eine IP-Adresse bekommen
<b>Beschreibung</b>	Beim Bootvorgang wird eine IP-Adresse bezogen.
<b>Teststufenplan</b>	Integrationstest
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• Ein DHCP Server ist im Netzwerk erreichbar</li> <li>• Der Computer wurde gestartet</li> <li>• Der Computer ist mit dem Netzwerk verbunden</li> <li>• Der Computer ist auf DHCP-IP-Adressenbezug konfiguriert</li> </ul>
<b>Testablauf</b>	Starten des Computers weiterlaufen lassen
<b>Erreichtes Ergebnis</b>	Eine IP-Adresse erhalten haben

### 1.7.2 Testfall 6

<b>Nummer</b>	6
<b>Name</b>	Per TFTP das Minilinux herunterladen
<b>Beschreibung</b>	Der DHCP-Server gibt dem PXE-Client den TFTP-Pfad zum Minilinux-Image, welches per TFTP heruntergeladen werden soll.
<b>Teststufenplan</b>	Integrationstest
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• Der Server hat ein Minilinux bereit</li> <li>• Übermittlungsbedingungen sind gewährleistet</li> <li>• Der Computer hat eine IP-Adresse erhalten</li> </ul>
<b>Testablauf</b>	Beim Bootvorgang wird das PXE-Image erfolgreich heruntergeladen
<b>Erreichtes Ergebnis</b>	Minilinux auf dem lokalen Rechner, welches erfolgreich bootet

### 1.7.1 Testfall 7

<b>Nummer</b>	7
<b>Name</b>	Das Minilinux starten
<b>Beschreibung</b>	Das Minilinux wird gestartet bis zum Anmeldebildschirm. Der Loginscreen (LDM LTSP Display Manager) ist ein Teil vom Minilinux.
<b>Teststufenplan</b>	Integrationstest
<b>Vorbedingungen</b>	TFTP-Image wurde erfolgreich heruntergeladen und gestartet
<b>Testablauf</b>	Warten nach dem TFTP-Boot, bis der Anmeldebildschirm erscheint
<b>Erreichtes Ergebnis</b>	Korrektes Erscheinungsbild der Anmeldemaske

### 1.7.2 Testfall 8

<b>Nummer</b>	8
<b>Name</b>	Am Server anmelden
<b>Beschreibung</b>	Ein SSH-Tunnel wird aufgebaut und der User kann sich am Server anmelden.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Erfolgreiches Booten und Verbinden zum Server
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>• Den SSH-Tunnel erstellen lassen (automatisch)</li> <li>• Anmelden mit Benutzernamen und Passwort</li> </ul>
<b>Erreichtes Ergebnis</b>	Sicher am Server angemeldet sein

### 1.7.3 Testfall 9

<b>Nummer</b>	9
<b>Name</b>	Am Server anmelden
<b>Beschreibung</b>	Ein SSH-Tunnel wird aufgebaut und der User kann sich am Server anmelden.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Verbindung mit dem Server kann hergestellt werden (Ping, SSH)
<b>Testablauf</b>	Wir melden uns mit einem eingerichteten User an.
<b>Erreichtes Ergebnis</b>	Wir sehen den Gnome-Desktop vor uns.

### 1.7.1 Testfall 10

<b>Nummer</b>	10
<b>Name</b>	Anmeldung entgegennehmen
<b>Beschreibung</b>	Der Server nimmt die Authentifizierung entgegen und zeigt dem User die grafische Oberfläche an.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>• SSH-Tunnel aufgebaut</li> <li>• Computer ist korrekt im Netzwerk verbunden</li> </ul>
<b>Testablauf</b>	Während des Anmeldevorgangs warten und die grafische Oberfläche betrachten
<b>Erreichtes Ergebnis</b>	<ul style="list-style-type: none"> <li>• Authentizitätsbestätigung</li> <li>• Oberfläche wird angezeigt</li> </ul>

### 1.7.2 Testfall 11

<b>Nummer</b>	11
<b>Name</b>	User verwalten
<b>Beschreibung</b>	Der Administrator kann neue User erfassen.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Root-Berechtigung auf zentralem Server Physikalischer oder Remote-Zugang zum Server
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● Anmelden am Server (root)</li> <li>● User-Verwaltungstool öffnen und User eintragen</li> <li>● User per Konsole erfassen</li> </ul>
<b>Erreichtes Ergebnis</b>	Der User wurde korrekt erfasst. Die Anmeldung am ThinClient am LTSP-Server funktioniert.

### 1.7.3 Testfall 12

<b>Nummer</b>	12
<b>Name</b>	Berechtigungen für die User setzen
<b>Beschreibung</b>	Der Administrator kann User speziell berechtigen.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Root-Berechtigung auf zentralem Server Physikalischer oder Remote-Zugang zum Server
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● Anmelden am Server (root)</li> <li>● User-Verwaltungstool öffnen und Rechte eintragen oder</li> <li>● User per Konsole editieren und Rechte eintragen</li> </ul>
<b>Erreichtes Ergebnis</b>	Der User kann speziell berechtigt werden und die Berechtigungen haben ihre Wirkung.

### 1.7.1 Testfall 13

<b>Nummer</b>	13
<b>Name</b>	Den Server per SSH verwalten
<b>Beschreibung</b>	Der Administrator kann sich am Server per SSH anmelden.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Der OpenSSH-Dienst ist korrekt installiert und konfiguriert
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● SSH-Verbindung (mit Putty und ssh) auf den Server</li> <li>● Zertifikat des OpenSSH-Servers hinzufügen</li> <li>● mit berechtigtem User einloggen</li> </ul>
<b>Erreichtes Ergebnis</b>	Der Administrator kann eine SSH-Verbindung zum Server aufbauen und sich erfolgreich einloggen.

### 1.7.1 Testfall 14

<b>Nummer</b>	14
<b>Name</b>	SCP machen
<b>Beschreibung</b>	Der Administrator kann Dateien mit SCP per SSH auf den Server kopieren.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Der OpenSSH-Dienst ist korrekt installiert und konfiguriert Ein SCP-Client ist installiert
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● SSH-Verbindung (mit Putty) auf den Server</li> <li>● Zertifikat des OpenSSH-Servers hinzufügen</li> <li>● mit berechtigtem User einloggen</li> <li>● Datei per SCP (Secure Copy Protocol -&gt; Steuerdaten und Dateiübertragung per SSH verschlüsselt!) übertragen: <code>scp Test1.bsp Benutzer@Host:Verzeichnis/Zieldatei.bsp</code></li> </ul>
<b>Erreichtes Ergebnis</b>	Der Administrator kann Dateien remote und verschlüsselt auf den Server übertragen.

### 1.7.1 Testfall 15

<b>Nummer</b>	15
<b>Name</b>	Zertifikatbasierte authentifikation (SSH)
<b>Beschreibung</b>	Die ganze Administrator-Server-SSH-Geschichte funktioniert mit Zertifikaten, damit es komfortabler funktioniert.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Der OpenSSH-Dienst ist korrekt installiert und konfiguriert Ein SCP-Client ist installiert
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● SSH-Verbindung (mit Putty oder ssh) auf den Server</li> <li>● Zertifikat des OpenSSH-Servers hinzufügen</li> <li>● mit berechtigtem User einloggen</li> </ul>
<b>Erreichtes Ergebnis</b>	Beim einloggen per SSH auf den Server muss kein Passwort mehr eingegeben werden.

### 1.7.1 Testfall 16

<b>Nummer</b>	16
<b>Name</b>	PXE-Boot der Clients
<b>Beschreibung</b>	Die Clients wählen die Netzwerkkarte als erstes Bootmedium.
<b>Teststufenplan</b>	Komponenten Tests
<b>Vorbedingungen</b>	Der PC funktioniert einwandfrei und wir gelangen ohne Probleme ins BIOS
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● BIOS des Clients starten</li> <li>● Bootreihenfolge so ändern, dass die Netzwerkkarte an 1. Position steht</li> <li>● Einstellungen speichern und neustarten.</li> </ul>
<b>Erreichtes Ergebnis</b>	Die Clients wählen die Netzwerkkarte als erstes Bootmedium.

### 1.7.1 Testfall 17

<b>Nummer</b>	17
<b>Name</b>	Audioausgabe bei den Clients
<b>Beschreibung</b>	Alle Soundausgaben werden beim User am Client ausgegeben.
<b>Teststufenplan</b>	Komponenten Tests
<b>Vorbedingungen</b>	Das Minilinux ist korrekt konfiguriert und setzt die Systemvariablen für den Sound.
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● Die Lautsprecherausgabe auf eine höhere Stufe stellen (nicht Mute)</li> <li>● Clientseitig eine Sounddatei öffnen und lauschen</li> </ul>
<b>Erreichtes Ergebnis</b>	Die Frequenzen erklingen über die Lautsprecher.

## 1.7.2 Testfall 18

<b>Nummer</b>	18
<b>Name</b>	Geschwindigkeitstest
<b>Beschreibung</b>	Die User sollen bei Büroarbeiten ungestört arbeiten können ohne Geschwindigkeitseinbussen.
<b>Teststufenplan</b>	Komponenten Tests
<b>Vorbedingungen</b>	Die Terminalsession ist auf die Geschwindigkeit optimiert.
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● Wir öffnen den Internetbrowser und einen Texteditor</li> <li>● Wir können wie gewohnt ohne grosse Verzögerungen multitaskfähig arbeiten.</li> </ul>
<b>Erreichtes Ergebnis</b>	Verzögerungsfreies Arbeiten (mit gewissen Einschränkungen wie z.B. Videowiedergabe) ist möglich.

## 1.7.1 Testfall 19

<b>Nummer</b>	19
<b>Name</b>	Verwendung von Applikationen
<b>Beschreibung</b>	Der User kann Applikationen starten.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Die Terminalsession lässt sich öffnen und die Benutzeroberfläche erscheint.
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● Wir öffnen den Internetbrowser und einen Texteditor</li> <li>● Die Applikationen starten auf und funktionieren</li> </ul>
<b>Erreichtes Ergebnis</b>	Der User kann Applikationen starten.

## 1.7.1 Testfall 20

<b>Nummer</b>	20
<b>Name</b>	Verwendung der öffentlichen Shares
<b>Beschreibung</b>	Der User kann auf Verzeichnisse auf dem Server zugreifen.
<b>Teststufenplan</b>	Systemtest
<b>Vorbedingungen</b>	Die User-Shares sind korrekt freigeben und die Berechtigungen korrekt gesetzt.
<b>Testablauf</b>	<ul style="list-style-type: none"> <li>● Wir öffnen als User clientseitig einen Share (mounten) cd /media/share/transfer ls</li> <li>● Wir können Dateien lesen und schreiben</li> </ul>
<b>Erreichtes Ergebnis</b>	Der User kann auf Dateien und Verzeichnisse auf dem zentralen Server zugreifen.

## 2 Glossar

Folgende Begriffe sind für einen Normalsterblichen genauer erklärt:

Begriff	Erklärung
DHCP	Dienst, der die IP-Adressen an die Clients verteilt.
DNS	Löst Namen in IP-Adressen auf. Damit man sich keine „komischen“ IP-Adressen merken muss.
IP-Adresse	Eindeutige Adresse eines Computers in einem Netzwerk. Quasi die Hausnummer eines Pcs.
LTSP	Linux Terminal Server Project: So heisst unsere Hauptsoftware vom Projekt.
NFS	Network File System: Dateien über das Netzwerk organisieren.
PXE	Preboot eXecution Environment: Beim Bootvorgang mit einem Server verbinden.
TFTP	Trivial File Transfer Protocol. Hiermit wird das Minilinux-System heruntergeladen nach dem PXE-Boot
NBD	Ein Network Block Device (engl. für Netzwerk-Blockgerät, abgekürzt NBD) ist eine Art virtuelle Festplatte, auf die ein Rechner via TCP/IP zugreifen kann. Das NBD wird von einem NBD-Server bereitgestellt. Er bietet hierfür eigene Festplatte, Festplattenpartition oder eine Datei als NBD bestimmten anderen Rechnern (Clients) an. Ein anderer Rechner (oder auch der gleiche) kann sich über eine TCP-Verbindung mit dem NBD-Server verbinden und anschließend das NBD wie eine eigene lokale Festplatte benutzen. (Quelle: Wikipedia)
vim	Klassischer und sehr mächtiger Texteditor.
SCP	Secure Copy Protocol (Abk. SCP) ist ein Protokoll sowie ein Programm zur verschlüsselten Übertragung von Daten zwischen zwei Computern über ein Rechnernetz.

M306

IT-Kleinprojekte abwickeln

Emanuel Duss, Arno Galliker, Semir Jahic

Gruppe: SP4

# Dokumentation

Einführung von LTSP (Linux Terminal Server Project)



**eos opensource solutions**

Projektbezeichnung	Einführung von LTSP (Linux Terminal Server Project)
Projektleiter	Emanuel Duss
Verantwortlich	Projektleiter
Erstellt	04.05.09
Letzte Änderung	2009-05-08 um 15:00:59
Zustand	Zur Prüfung
Pfad	/ media/APACER_2GB/Schule/3_Lehrjahr/306_IT_Kleinprojekt_abwickeln/02_Projekt/ 07_Dokumentation/07_Dokumentation.odt

## Projektmitglieder

Emanuel	Duss	Im Bienz 15	6170 Schüpfheim	emanuel.duss#gmail.com	EDU
Semir	Jahic	Moosmatte 32	6182 Escholzmatt	semir.jahic#gmail.com	SJA
Arno	Galliker	Margrethenstrasse 8	6275 Ballwil	arno.galliker#gmail.com	AGA

## Änderungen

Datum	Version	Kapitel	Beschreibung	Autor
2009-05-04	0.1	ALLE	Grundgerüst vom Dokument	EDU
2009-05-08	1.0	ALLE	Fertigstellung und Abgabe	ALLE

## Prüfungen / Reviews

Datum	Version	Kapitel	Beschreibung	Autor
	1	Alle		GAS

## Infos

Zuletzt bearbeitet	2009-05-08
Heute	2009-11-23
Bearbeitungszeit	18:34:05
Lehrjahr des Moduls	3. Lehrjahr; 2008 / 2009
Pfad	/ media/APACER_2GB/Schule/3_Lehrjahr/306_IT_Kleinprojekt_abwickeln/02_Projekt/07_Dokumentation/07_Dokumentation.odt
CC-Lizenz	Creative Commons Namensnennung-Keine kommerzielle Nutzung-Weitergabe unter gleichen Bedingungen 2.5 Schweiz <a href="http://creativecommons.org/licenses/by-nc-sa/2.5/ch/">http://creativecommons.org/licenses/by-nc-sa/2.5/ch/</a>

# Inhaltsverzeichnis

<b>1</b>	<b>Netzwerkübersicht.....</b>	<b>5</b>
1.1	IP-Konzept.....	5
1.2	Netzwerkplan.....	5
<b>2</b>	<b>Installation vom LTSP-Server.....</b>	<b>6</b>
2.1	Grundinstallation.....	6
2.2	Weitere Software.....	7
<b>3</b>	<b>Konfiguration der Dienste.....</b>	<b>8</b>
3.1	DHCP-Server.....	8
3.2	TFTP-Server.....	8
3.3	LTSP.....	8
3.3.1	Anpassungen.....	8
3.4	LTSP Thin Client Manager.....	9
3.5	iTALC.....	9
3.5.1	Installation.....	9
3.5.2	Clients hinzufügen.....	9
3.5.3	Übersicht GUI.....	10
3.5.4	Funktionen von iTALC.....	10
<b>4</b>	<b>Clients.....</b>	<b>11</b>
4.1	BIOS.....	11
4.2	Aufstarten.....	11
<b>5</b>	<b>Benutzer einrichten.....</b>	<b>12</b>
5.1	Grundbedingung.....	12
5.2	User hinzufügen.....	12
5.3	User löschen.....	12
5.4	Gruppen hinzufügen.....	12
5.5	Gruppen entfernen.....	12
5.6	Gruppen verwalten.....	12
<b>6</b>	<b>Bug in 9.04.....</b>	<b>13</b>
6.1	Report a bug.....	13
<b>7</b>	<b>Detailbeschreibung.....</b>	<b>15</b>
7.1	Bootvorgang.....	15
7.1.1	DHCP.....	15
7.1.2	TFTP.....	15
7.2	Login am Server.....	15
7.3	VNC-Zugriff.....	16
7.3.1	VNC-Zugriff auf die Clients.....	16
7.3.2	VNC-Zugriff auf den Server.....	16
<b>8</b>	<b>Glossar.....</b>	<b>17</b>

## Abbildungsverzeichnis

Abbildung 1: Netzwerkübersicht.....	6
Abbildung 2: Wir wählen die LTSP-Installationsvariante aus.....	7
Abbildung 3: Clients zu iTALC hinzufügen.....	10
Abbildung 4: GUI iTALC.....	12
Abbildung 5: Unser Login-Screen.....	13
Abbildung 6: DHCP-Anfrage.....	17
Abbildung 7: TFTP-Download.....	17
Abbildung 8: Verbindungen zum Server.....	17
Abbildung 9: Fast nur SSH-Traffic.....	17

# 1 Netzwerkübersicht

## 1.1 IP-Konzept

- Server: 192.168.1.10 /24
- Clients (DHCP-Range): 192.168.1.20 -250
- Default-Gateway: 192.168.1.1

## 1.2 Netzwerkplan

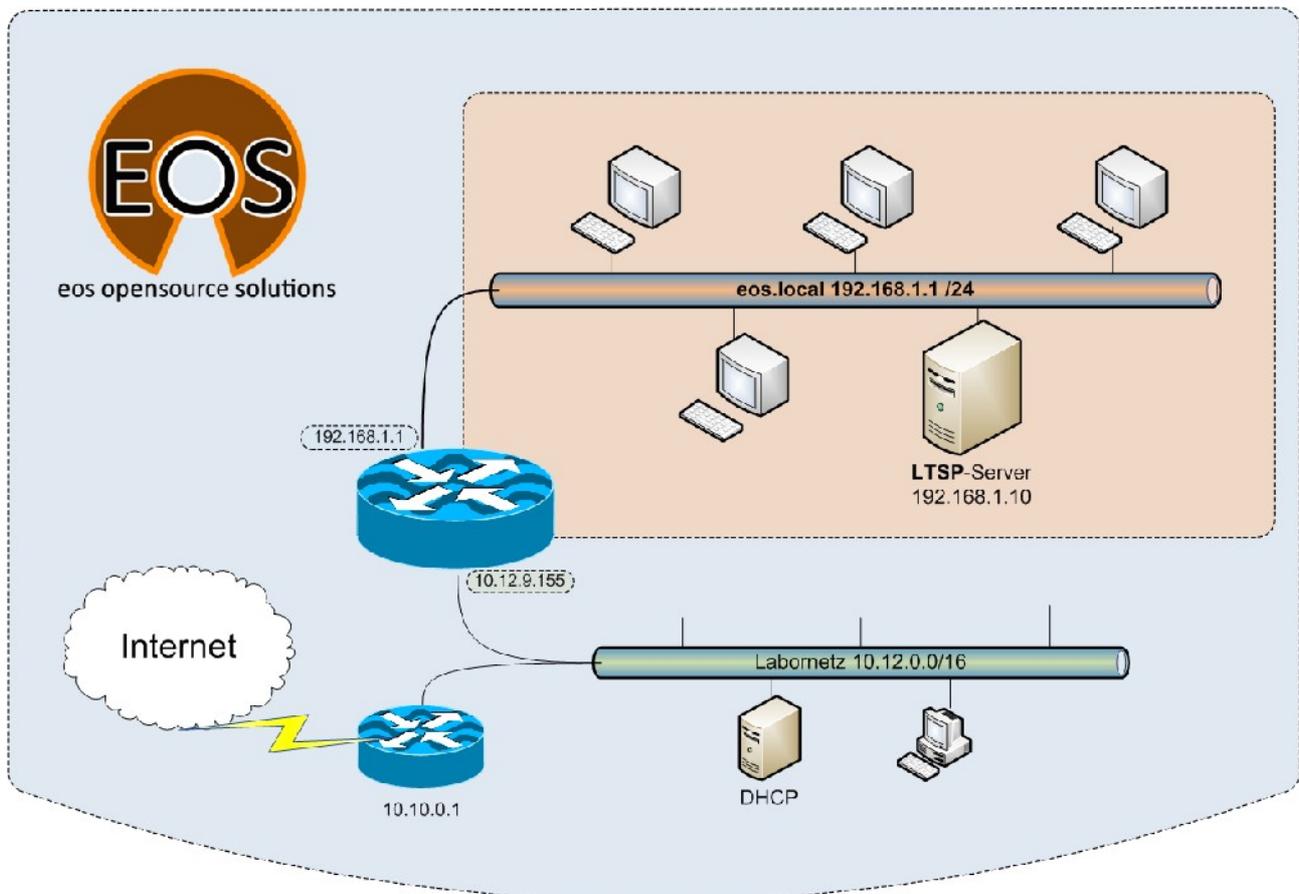


Abbildung 1: Netzwerkübersicht

## 2 Installation vom LTSP-Server

### 2.1 Grundinstallation

Wir haben einen bootfähigen USB-Stick erstellt, von dem wir die Ubuntu Alternate CD starten können. Folgende Angaben haben wir während der Installation gemacht:

- Optionen (F4): LTSP-Server
- IP-Adresse: 192.168.1.10
- Netzmaske: 255.255.255.0
- Default-Gateway: 192.168.1.1
- Servername: ltsp
- Domain: eos.local
- Username: ltsp
- Passwort: just4us



Abbildung 2: Wir wählen die LTSP-Installationsvariante aus

## 2.2 Weitere Software

Wir installieren zusätzliche Software, die wir für unsere Arbeit brauchen. Folgende Software haben wir installiert:

- Texteditor vim

```
ltsp@ltsp:~$ sudo apt-get install vim
```

- Wireshark um den Netzwerktraffic anzuzeigen. Alles läuft über den Server!

```
ltsp@ltsp:~$ sudo apt-get install wireshark
```

- Der Portscanner nmap um auf das Minilinux der Clients zu „schauen“.

```
ltsp@ltsp:~$ sudo apt-get install nmap
```

- iptraf um zu zeigen, dass SSH das meistgenutzte Protokoll in unserem Netzwerk ist

```
ltsp@ltsp:~$ sudo apt-get install iptraf
```

## 3 Konfiguration der Dienste

### 3.1 DHCP-Server

Der DHCP-Server vergibt beim Booten die IP-Konfiguration an die Clients und gibt den Pfad zum TFTP-Server.

```
root@ltsp:~# cat /etc/ltsp/dhcpd.conf
authoritative;

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.20 192.168.1.250;
    option domain-name "eos.local";
    option domain-name-servers 192.168.1.1;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.1;
    option subnet-mask 255.255.255.0;
    option root-path "/opt/ltsp/i386";
    if substring( option vendor-class-identifier, 0, 9 ) = "PXELinux" {
        filename "/ltsp/i386/pxelinux.0";
    } else {
        filename "/ltsp/i386/nbi.img";
    }
}
```

### 3.2 TFTP-Server

```
root@ltsp:~# cat /etc/default/tftpd-hpa
#Defaults for tftpd-hpa
RUN_DAEMON="no"
OPTIONS="-l -s /var/lib/tftpboot"
```

Wir sehen, dass der TFTP-Server nicht als Daemon läuft. Doch wie wird er gestartet? Mit dem Superserver inetd.

Folgender Eintrag in der Datei /etc/inetd ist für den dynamischen start des TFTP-Servers zuständig:

```
tftp          dgram        udp          wait        root    /usr/sbin/in.tftpd
/usr/sbin/in.tftpd -s /var/lib/tftpboot
```

### 3.3 LTSP

#### 3.3.1 Anpassungen

##### LTSP-Login-Screen anpassen

Wir ändern das default-Logo des LTSP-Login-Screen's im folgenden Ordner.

```
/opt/ltsp/i386/usr/share/ldm/themes/default
Logo.png überschrieben
```

Danach muss das LTSP-Minilinux-Image neu erstellt werden.

```
ltsp-update-image
```

Nach dem Update müssen die Clients neu gestartet werden, damit sie das neue Image erhalten.

## 3.4 LTSP Thin Client Manager

Da der Thin Client Manager nicht mehr weiter entwickelt wird und bei der Installation und Konfiguration auf Ubuntu 9.04 und 8.10 Probleme bereitete und trotz langem probieren keine Lösung in Sicht war, haben wir uns entschieden für die Remoteüberwachung der Clients auf das Produkt iTalc zu setzen.

## 3.5 iTALC

iTALC (Intelligent Teaching And Learning with Computers) ist eine freie Bildschirmsteuersoftware für Linux und Windows, die vor allem auf den schulischen Bereich abzielt. Mit Hilfe von iTALC können Lehrer Schüler-Computer beobachten, fernsteuern, sperren, eine Demo zeigen, an-/ausschalten uvm. Als Grundlage für alle Funktionen dient ein erweitertes RFB-Protokoll.

ITALC ist also perfekt für unsere Schule in Somalia.

### 3.5.1 Installation

Auf dem Server installieren wir iTALC.

```
sudo apt-get install italc
```

### 3.5.2 Clients hinzufügen

Damit die Clients im iTALC-Manager auftauchen, müssen wir diese hinzufügen.

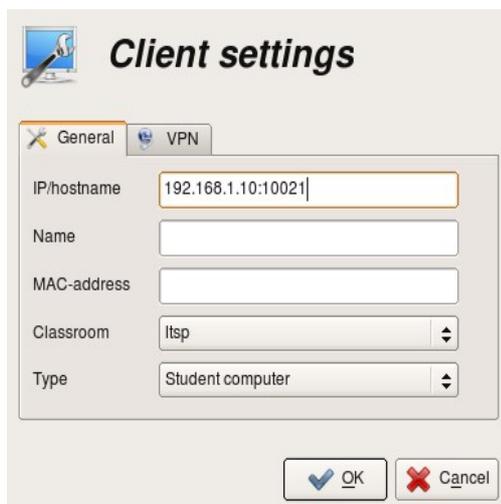


Abbildung 3: Clients zu iTALC hinzufügen

Die IP-Adresse setzt sich folgendermassen zusammen:

serverip:10xxx

xxx entspricht den letzten drei Zahlen der IP-Adresse von einem Client.

Wieso ist das so? Vgl. Detailbeschreibung.

### 3.5.3 Übersicht GUI

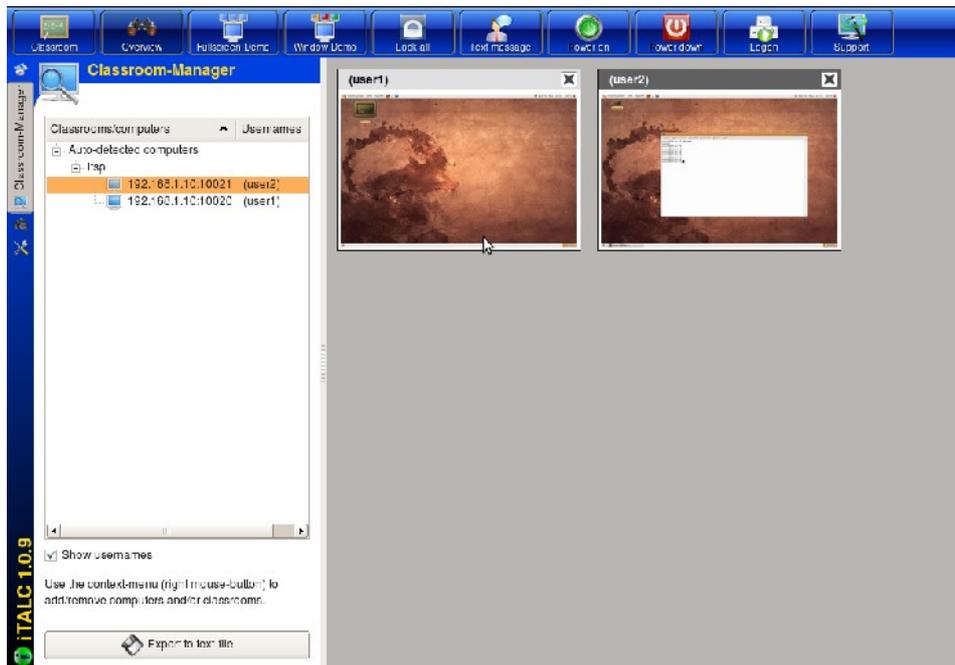


Abbildung 4: GUI iTALC

### 3.5.4 Funktionen von iTALC

- Bildschirmüberwachung
- Bildschirm der Schüler übernehmen
- Rechner herunterfahren
- Rechner neu starten
- Screenshots von den Clients machen
- Den Desktop von einem Schüler an alle Clients senden

## 4 Clients

### 4.1 BIOS

Wir aktivieren im BIOS PXE-Boot und stellen die Boot-Reihenfolge so ein, dass als erstes die Netzwerkkarte gebootet wird.

Danach können wir den Client neu starten. Zusätzlich können wir die Verbindung zur Harddisk trennen, um sicherzustellen, dass kein lokales Speichermedium nötig ist.

### 4.2 Aufstarten

Nachdem der Client beim PXE-Booten vom LTSP-Server einen DHCP-Release erhalten haben, bekommt er das Mini-Linux per TFTP geschickt. Dieses erkennt alle Hardware und startet nachher den LTSP-Client.

Hier kommt dann das Anmeldefenster.

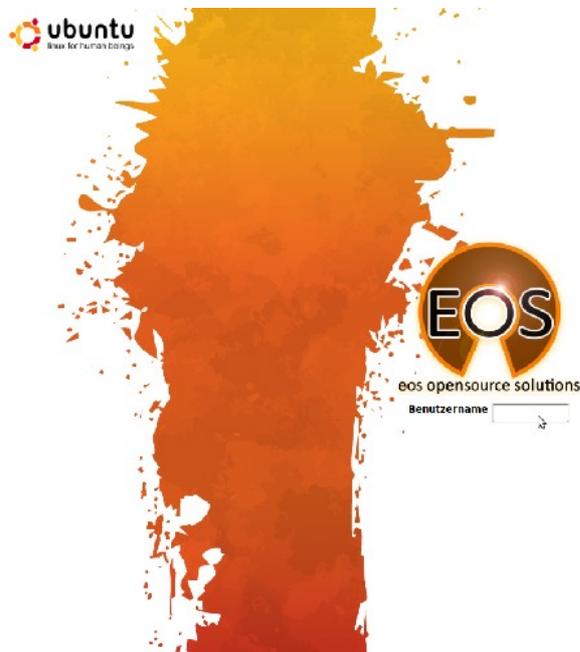


Abbildung 5: Unser Login-Screen

Wir können uns mit jedem vorhanden Benutzer anmelden.

Für die normalen User erstellen wir jedoch eingeschränkte Profile.

## 5 Benutzer einrichten

### 5.1 Grundbedingung

Eine Putty-Session muss auf den Server gestartet werden. Am Server sollte man sich anmelden und sofort Root-Rechte erlangen. Dies schafft man durch

```
sudo -s
```

und danach die Passworteingabe.

### 5.2 User hinzufügen

Durch den Befehl `adduser` und die dazugehörigen Parameter kann man den Benutzer – hier „John“ - hinzufügen und das Home-Verzeichnis wird ebenfalls gleich erstellt.

```
adduser --home /home/john john
```

### 5.3 User löschen

Um Benutzer zu löschen, kann man einfach `deluser` eingeben, danach den Parameter `remove`, der das Home-Verzeichnis, sowie den Benutzer löscht.

```
deluser --remove-home john
```

### 5.4 Gruppen hinzufügen

Der Befehl `addgroup` fügt eine Gruppe hinzu, in der später Benutzer angeordnet werden können.

```
addgroup accounts
```

### 5.5 Gruppen entfernen

Um eine Gruppe zu löschen, muss man `delgroup` ausführen. Falls die Gruppe nicht leer ist, löscht der unten stehende Befehl diese aber noch nicht.

```
delgroup --only-if-empty students
```

Hier wird die Gruppe einfach gelöscht

```
sudo delgroup students
```

### 5.6 Gruppen verwalten

Um einzelne Benutzer gewissen Gruppen hinzufügen zu können, kann man diesen einfach angeben und dahinter die jeweilige Gruppe klassifizieren.

Beim Löschen geht man gleich vor, die Befehle sind identisch, wie die beim Hinzufügen und Löschen der Benutzer.

Hinzufügen zu Gruppe:

```
adduser john students
```

Löschen aus Gruppe:

```
deluser john students
```

## 6 Bug in 9.04

Mit Ubuntu-Alternate 9.04 kamen zahlreiche gute Neuerungen hinzu, leider darunter auch ein Bug, der uns den Fortschritt im Projekt entscheidend erschwerte. Der Thin Client Manager, der zum administrieren der Thin Clients verantwortlich ist, funktioniert nicht korrekt. Der Fehler besteht darin, dass, nachdem er gestartet ist, die Clients nicht gesehen werden.

Wir haben den Server danach mit der Version 8.10 installiert und die gleiche Infrastruktur beibehalten, die Clients werden nach der Anmeldung nun angezeigt, sobald diese angemeldet sind.

### 6.1 Report a bug

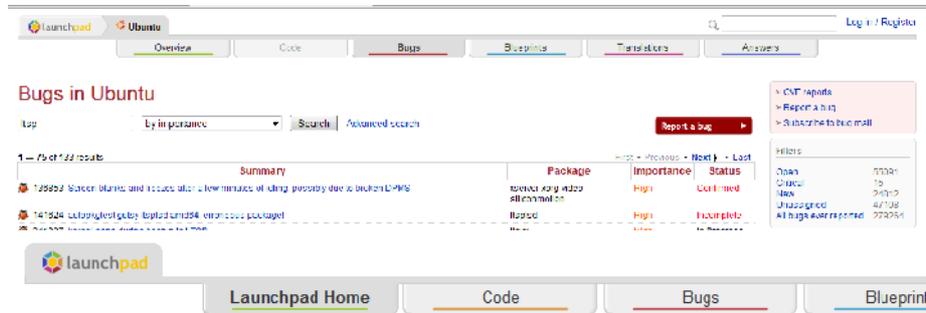
Diesen Fehler haben wir der Community natürlich gleich gemeldet und somit auch entdeckt, wie einfach es ist, über Fehler zu berichten.

Auf der Webseite <https://launchpad.net/ubuntu/+bugs> geht man wie folgt vor:

Der Bug wurde noch nicht Ubuntu gemeldet.

Nun haben wir *Report a bug* ausgewählt.

Die Anmeldung ist kurz und einfach, nur die E-Mail-Adresse muss angegeben werden, danach kommt ein Bestätigungslink, wo man den Benutzernamen und ein Passwort bestimmen kann.



#### Complete your registration

Display Name:

Your name as you would like it displayed throughout Launchpad. Most people use their full name here.

Hide my email addresses from other Launchpad users

Create password:

Retype the password:

Jetzt ist es bereits möglich den Bug zu beschreiben und ihn zu veröffentlichen.

#### Report a bug

Where did you find the bug?:

Distribution:  (Choose...)

Package:  (Choose...)

Project:  (Choose...)

Please describe the bug in a few words, for example, "weather applet crashes on logout":

Summary:

[Advanced reporting options](#)

[Contact us](#) | [Get help with Launchpad](#)

So sieht dann der veröffentlichte Bug aus, nun kann man nur auf Antworten und Bugfixes warten.

The screenshot shows a bug report interface. At the top, it says 'Bug #372100 reported by bszlein 4 weeks ago (Activity log)'. The title is 'Thin Client Manager shows nothing in 9.04'. Below the title is a table with columns: Affects, Status, Importance, Assigned to, and Milestone. The 'Affects' column contains 'Ubuntu' and 'New'. The 'Status' column contains 'In Progress'. The 'Importance' column contains 'Undecided'. The 'Assigned to' column is empty. The 'Milestone' column is empty. Below the table are several action links: 'Also affects project', 'Also affects distribution', 'Nominate for release', and 'This bug affects no one (change)'. The main text of the bug report reads: 'Thin Client Manager shows nothing in 9.04, there's neither a connected client nor a logged on user shown in the sidebar of the application. Despite that the machine is running and a user logged in. With 9.10 this works flawlessly under the same conditions, something must have changed due to update in Ubuntu. (9.04)'. Below the text are more action links: 'Update description / tags', 'Link a related branch', 'Link to CV', and 'Add a comment/attachment'. On the right side, there is a sidebar with a list of users: 'Unsubscribe', 'Subscribe', 'Subscribers', 'Upvote', 'Also notified', 'Colin Milner', 'bzt', 'Andreas Strohmann', 'joshkatz', 'Gerrit J. Bahner', 'vague', 'Jesse Jones', and 'Jury'.

## 7 Detailbeschreibung

### 7.1 Bootvorgang

#### 7.1.1 DHCP

Wir sehen, dass der Client per DHCP eine IP-Adresse bezieht:

0.0.0.0	255.255.255.255	DHCP	DHCP Discover
192.168.1.10	192.168.1.20	DHCP	DHCP Offer
0.0.0.0	255.255.255.255	DHCP	DHCP Request
192.168.1.10	192.168.1.20	DHCP	DHCP ACK

Abbildung 6: DHCP-Anfrage

#### 7.1.2 TFTP

Nun lädt der Client das Minilinux vom Server herunter:

```

192.168.1.20 192.168.1.10 TFTP Read Request, File: /.tsp/i386/pxelinux.0\000, Transfer type: octet\000, tsize\000=0\000
192.168.1.10 192.168.1.20 TFTP Option Acknowledgement, tsize\000=14776\000
192.168.1.20 192.168.1.10 TFTP Error Code, Code: Not defined, Message: TFTP Aborted\000
192.168.1.20 192.168.1.10 TFTP Read Request, File: /.tsp/i386/pxelinux.0\000, Transfer type: octet\000, blksize\000=1456\000
192.168.1.10 192.168.1.20 TFTP Option Acknowledgement, blksize\000=1456\000
192.168.1.20 192.168.1.10 TFTP Acknowledgement, Block: 0
192.168.1.10 192.168.1.20 TFTP Data Packet, Block: 1
192.168.1.20 192.168.1.10 TFTP Acknowledgement, Block: 1
192.168.1.10 192.168.1.20 TFTP Data Packet, Block: 2
192.168.1.20 192.168.1.10 TFTP Acknowledgement, Block: 2
    
```

Abbildung 7: TFTP-Download

## 7.2 Login am Server

Die Verbindung zwischen Client und Server läuft über einen SSH-Tunnel.

```

IPTraf
TCP Connections (Source Host:Port) ----- Packets --- Bytes Flags Iface
192.168.1.10:22 > 55 4984 --A- eth0
192.168.1.20:39966 > 37 6526 -PA- eth0
192.168.1.10:22 > 52 4752 --A- eth0
192.168.1.21:35928 > 35 6110 -PA- eth0
    
```

Abbildung 8: Verbindungen zum Server

Der meiste Traffic, der über unseren Server läuft, wird von SSH produziert:

```

IPTraf
Proto/Port ----- Pkts --- Bytes --- PktsTo --- BytesTo PktsFrom BytesFrom ---
TCP/22 20549 15625076 11744 15069340 8805 555736
UDP/137 12 936 12 936 12 936
    
```

Abbildung 9: Fast nur SSH-Traffic

## 7.3 VNC-Zugriff

### 7.3.1 VNC-Zugriff auf die Clients

Wie greift der Server auf die Clients zu? Da wir iTALC nicht im Minilinux installiert haben, können wir ja nicht die IP-Adresse vom Client nehmen:

```
ltsp@ltsp:~$ nmap 192.168.1.20

Starting Nmap 4.62 ( http://nmap.org ) at 2009-05-08 08:51 CEST
Interesting ports on 192.168.1.20:
Not shown: 1714 closed ports
PORT      STATE SERVICE
6007/tcp  open  X11:7

Nmap done: 1 IP address (1 host up) scanned in 0.146 sec
```

Der Server erledigt das auf eine ganz elegante Art: Er öffnet einen neuen Port, auf dem der Bildschirm vom Client per VNC dargestellt werden kann:

```
ltsp@ltsp:~$ netstat | grep local:10
tcp        0      0 ltsp.eos.local:53230    ltsp.eos.local:10020    VERBUNDEN
tcp        0      0 ltsp.eos.local:10020    ltsp.eos.local:53230    VERBUNDEN
```

### 7.3.2 VNC-Zugriff auf den Server

Um von den Clients auf den Server zu schauen (z.B. Um eine Lernpräsentation zu halten) wird der Port 5900 verwendet:

```
ltsp@ltsp:~$ nmap localhost

Starting Nmap 4.62 ( http://nmap.org ) at 2009-05-08 08:51 CEST
Warning: Hostname localhost resolves to 2 IPs. Using 127.0.0.1.
Interesting ports on localhost (127.0.0.1):
Not shown: 1710 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
631/tcp   open  ipp
2000/tcp  open  callbook
5800/tcp  open  vnc-http
5900/tcp  open  vnc

Nmap done: 1 IP address (1 host up) scanned in 0.114 seconds
```

## 8 Reflexion

Es war ein sehr interessantes Projekt und die Durchführung machte sehr viel Spass. Anfangs war eine Sache schwierig: Wie es so häufig ist, musste man sich irgendwie eingrenzen, denn der Themenbereich ist so breit gefächert. Diese Wahl bescherte bereits die erste Herausforderung.

Nach reiflicher Überlegung war es schliesslich doch möglich einen passenden Rahmen für die ganze Sache zu finden.

Die Zeit verging schnell während der Arbeit. Anfangs gab es kleinere Hürden, die nicht geplant waren. Trotz dessen war der Erfolg der ganzen Unternehmung nicht gefährdet. Mit Durchhaltewillen und viel Zeit sowie Geduld, war es möglich ans Ziel zu gelangen.

Am Ende kann man zufrieden sagen, das Projekt ist anspruchsvoll aber realisierbar und dementsprechend gelungen.

Der persönliche Eindruck spiegelt sich selbstverständlich in der Bewertung des Ganzen wieder und ist somit sehr gut ausgefallen.

Es ist grundsätzlich nicht zu vergessen, dass die Chance, ein Thema so eingängig zu behandeln und zu erforschen, nicht immer gegeben ist.

Deshalb sollte man immer froh sein, denn so ein Projekt ist stets lehrreich und für das Leben hilfreich. So war es auch mit diesem Projekt wieder.

## 9 Glossar

Folgende Begriffe sind für einen Normalsterblichen genauer erklärt:

Begriff	Erklärung
DHCP	Dienst, der die IP-Adressen an die Clients verteilt.
DNS	Löst Namen in IP-Adressen auf. Damit man sich keine „komischen“ IP-Adressen merken muss.
IP-Adresse	Eindeutige Adresse eines Computers in einem Netzwerk. Quasi die Hausnummer eines Pcs.
LTSP	Linux Terminal Server Project: So heisst unsere Hauptsoftware vom Projekt.
NFS	Network File System: Dateien über das Netzwerk organisieren.
PXE	Preboot eXecution Environment: Beim Bootvorgang mit einem Server verbinden.
TFTP	Trivial File Transfer Protocol. Hiermit wird das Minilinux-System heruntergeladen nach dem PXE-Boot
NBD	Ein Network Block Device (engl. für Netzwerk-Blockgerät, abgekürzt NBD) ist eine Art virtuelle Festplatte, auf die ein Rechner via TCP/IP zugreifen kann. Das NBD wird von einem NBD-Server bereitgestellt. Er bietet hierfür eigene Festplatte, Festplattenpartition oder eine Datei als NBD bestimmten anderen Rechnern (Clients) an. Ein anderer Rechner (oder auch der gleiche) kann sich über eine TCP-Verbindung mit dem NBD-Server verbinden und anschließend das NBD wie eine eigene lokale Festplatte benutzen. (Quelle: Wikipedia)
vim	Klassischer und sehr mächtiger Texteditor.
SCP	Secure Copy Protocol (Abk. SCP) ist ein Protokoll sowie ein Programm zur verschlüsselten Übertragung von Daten zwischen zwei Computern über ein Rechnernetz.
Bug	Fehler in Programmen, die eine bestimmte Funktion verhindern oder erschweren

Präsentation  
LTSP  
Linux Terminal Server Project

# Präsentation ¶

Modul 305 //  
BBZWITS

\*\*\*\*\*

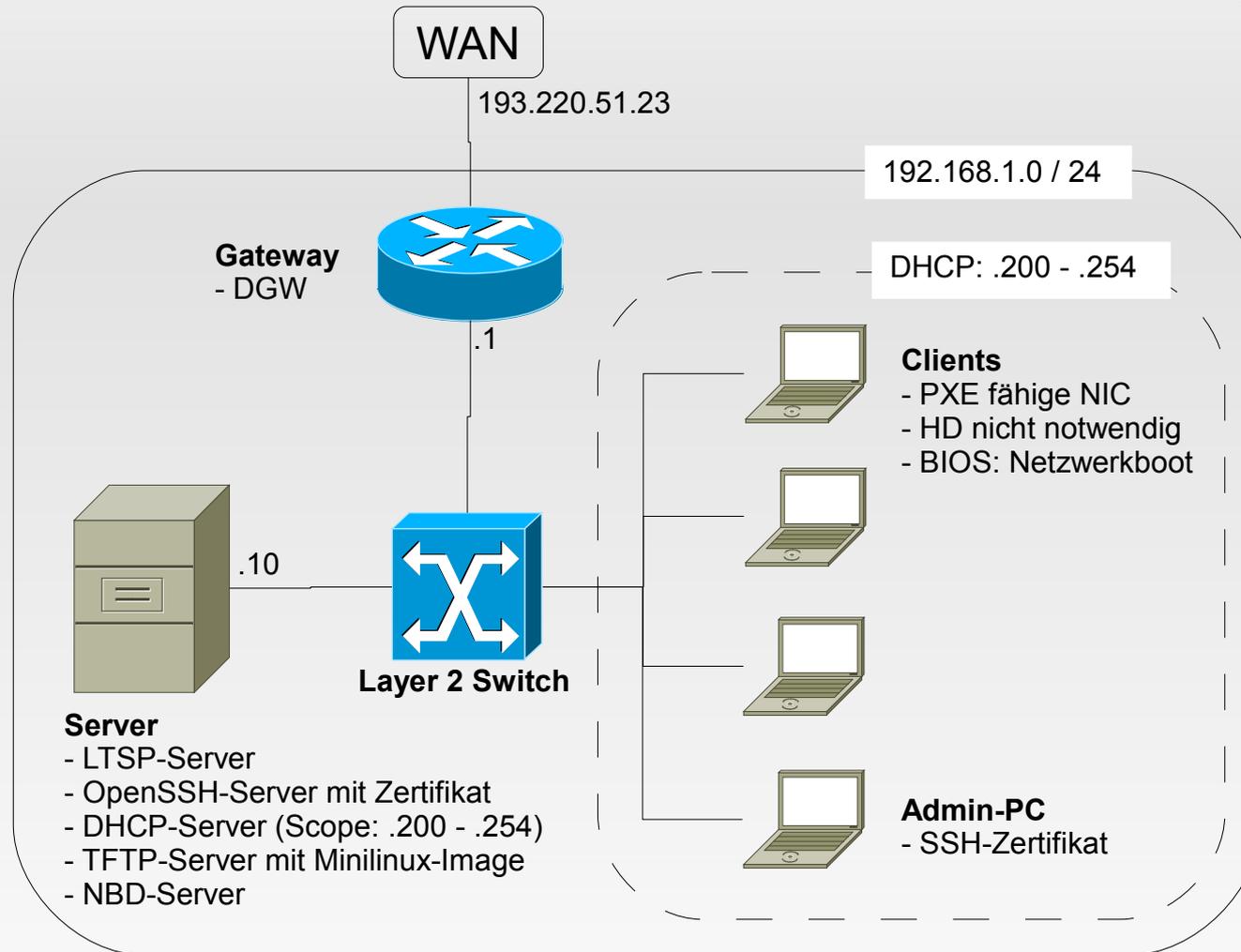
## Inhalt ¶

- Installation, Konfigurati
- Dateien, Koron ¶
- 

## Präsentation ¶

- - > Demonstration ¶

# Netzwerklayout



# Installation



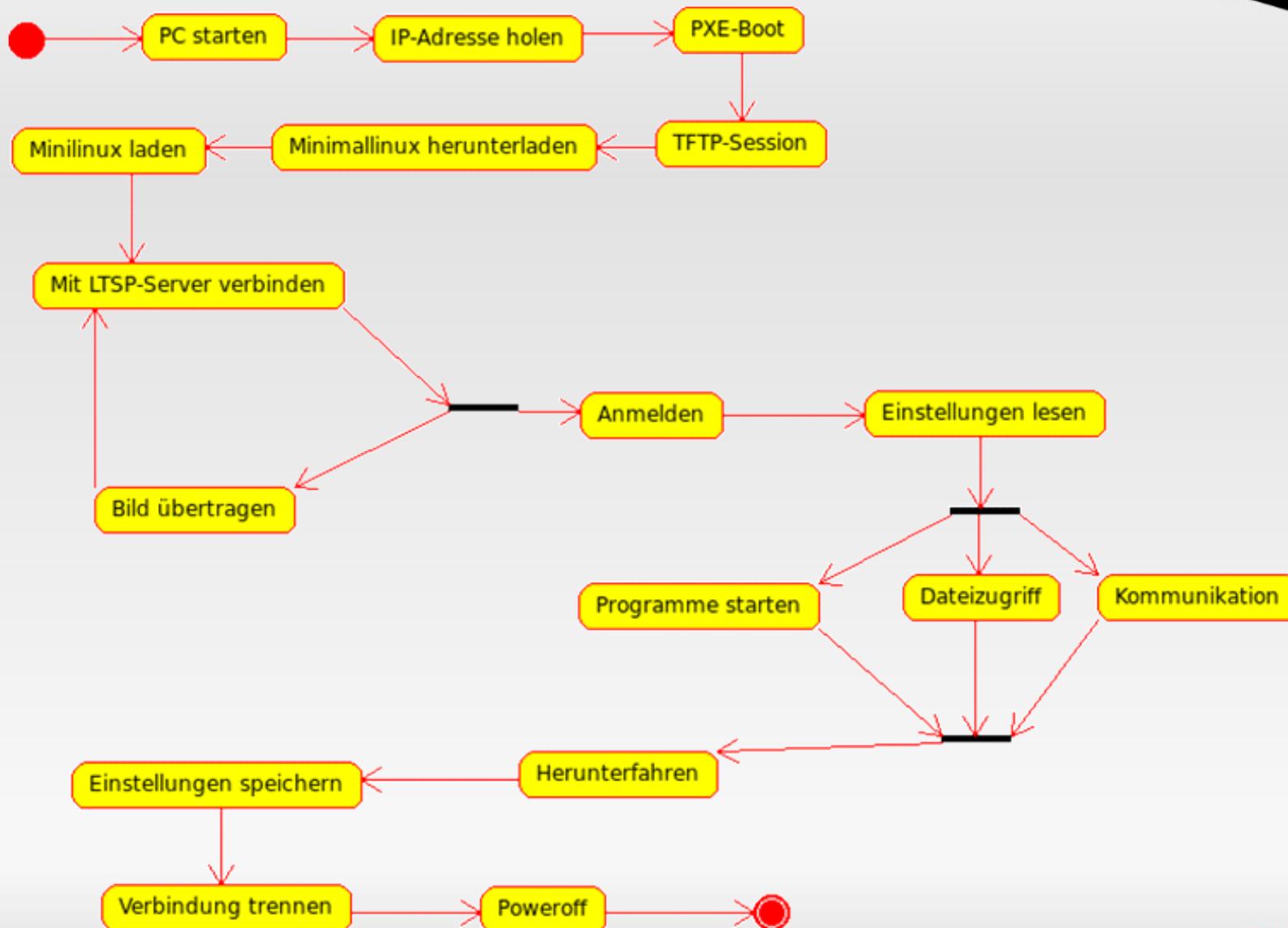
- Ubuntu 8.10, da 9.04 fehlerhaft war
  - Bugreport
- Dienste
  - DHCP-Server, TFTP-Server, SSH-Server
- Tools
  - iTALC

# Der Bootvorgang



- PXE-Boot (Boot von Netzwerkkarte)
- DHCP
- TFTP-Link lädt Minilinux in den RAM (150 MB)
- X-Server startet
- Logon am LTSP-Server über SSH-Tunnel

# Aktivitätsdiagramm



# Konfiguration DHCP



- /etc/ltsp/dhcp.conf

```
authoritative;
```

```
subnet 192.168.1.0 netmask 255.255.255.0 {  
    range 192.168.1.20 192.168.1.250;  
    option domain-name "example.com";  
    option domain-name-servers 192.168.1.1;  
    option broadcast-address 192.168.1.255;  
    option routers 192.168.1.1;  
#    next-server 192.168.0.1;  
#    get-lease-hostnames true;  
    option subnet-mask 255.255.255.0;  
    option root-path "/opt/ltsp/i386";  
    if substring( option vendor-class-identifier, 0, 9 ) = "PXEClient"  
    {  
        filename "/ltsp/i386/pxelinux.0";  
    } else {  
        filename "/ltsp/i386/nbi.img";  
    }  
}
```

# Anpassung Minilinux



- Chroot-Umgebung (/opt/ltsp/i386/)
  - Softwareinstallation: iTALC
  - Dateien: iTALC-Zertifikate, Logonbild
- Danach das Image neu erzeugen
  - `sudo ltsp-update-image`

# Präsentation iTALC



- Sehet selbst!
- Wireshark: Bootvorgang
- Iptraf: SSH-Tunnel
- Nmap: offene Ports

# Präsentation

Modul 305 //  
BBZWI<sup>TS</sup>

\*\*\*\*\*

5 Tonnen Flachs

## Inhalt

- Installation, Konfiguration
- Dateieren, Konfiguration
- **Vragen?**

## Präse

