

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

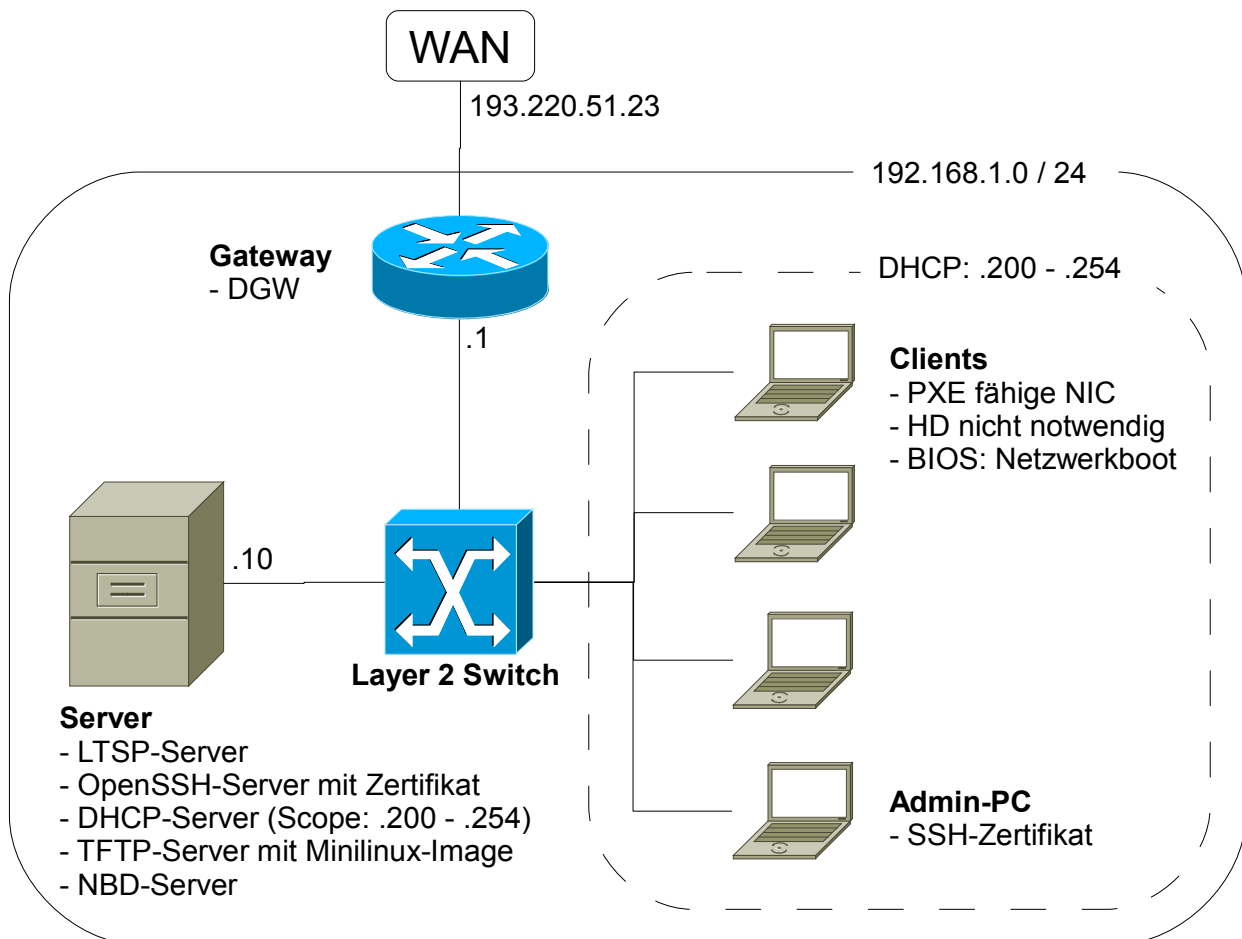


Abbildung 1: Soll-System Netzwerkübersichtsplan

#### 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:



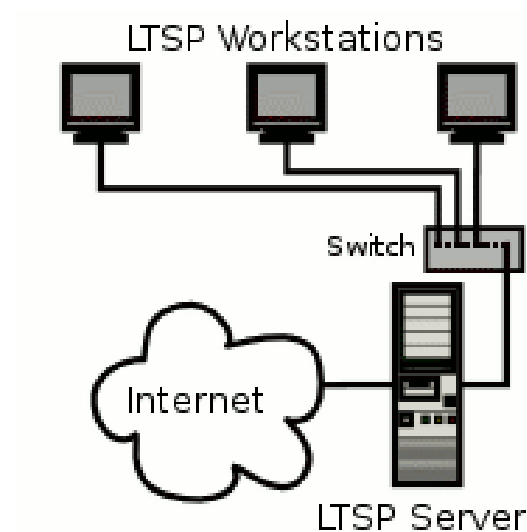
Folgende eingaben müssen wir während der Installation machen:

- 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.

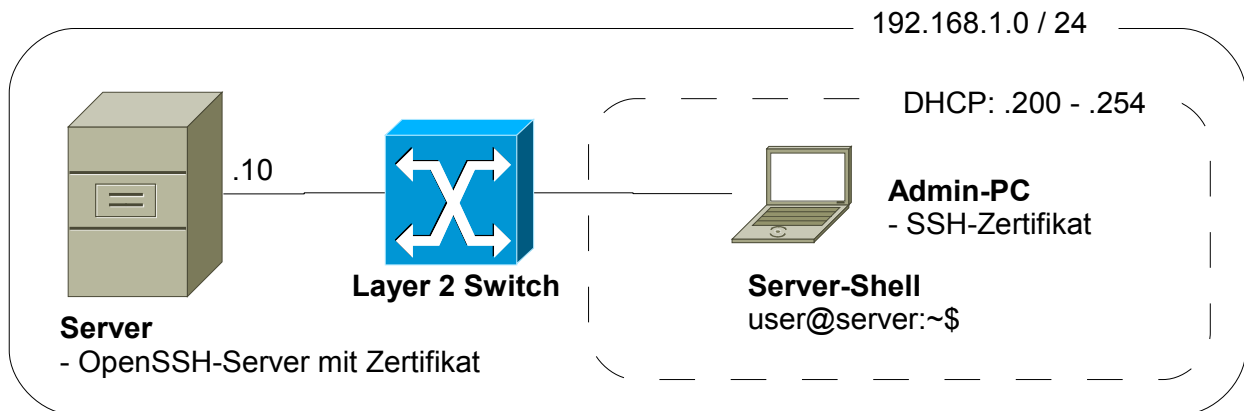
Auf dem SSH-Client erstellt man die zwei Keys und verteilt dann den öffentlichen auf den Server.

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

### 3.5.3 Use-Cases

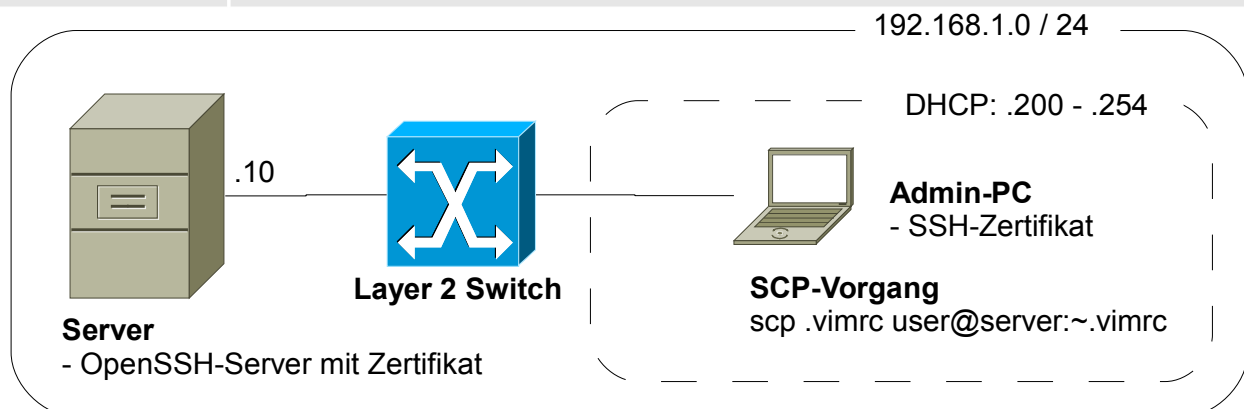
#### Funktion 1: SSH-Zugriff

<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.