

# Projektdokumentation

## Installation eines Linux-Clients mittels FAI-Server

Katharina Drexel  
Althöhensteigstr. 1  
83071 Stephanskirchen

Ausbildungsbetrieb: GeNUA  
Gesellschaft für Netzwerk- und Unix-Administration mbH  
Domagkstraße 7  
85551 Kirchheim

Projektverantwortlicher: Nikolay Sturm

Ausführungszeitraum: 26.02.2007 bis 14.03.2007

*Persönliche Erklärung:*

Ich versichere, dass ich das Projekt und die dazugehörige Dokumentation selbstständig erstellt habe.

Ort und Datum: Katharina Drexel

Kirchheim, den 4. Mai 2007 .....



# Inhaltsverzeichnis

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Projektrahmen</b>   | <b>1</b>  |
| 1.1      | Vorstellung des Unternehmens . . . . .   | 1         |
| 1.2      | Zustandekommen des Projekts . . . . .  | 1         |
| 1.3      | Ist-Zustand . . . . .  | 1         |
| 1.4      | Soll-Zustand . . . . .   | 2         |
| 1.5      | Kosten-Nutzen-Analyse . . . . .  | 2         |
| 1.6      | Projektschnittstellen . . . . .  | 3         |
| <b>2</b> | <b>Planung</b>   | <b>3</b>  |
| 2.1      | Planung der Testumgebung . . . . .   | 3         |
| 2.2      | Testhardware . . . . .   | 4         |
| 2.3      | Ablaufkonzept . . . . .  | 5         |
| <b>3</b> | <b>Realisierung</b>  | <b>5</b>  |
| 3.1      | Installation und Konfiguration des Servers . . . . .                                     | 5         |
| 3.2      | Konfiguration des Routers . . . . .  | 6         |
| 3.3      | Installation und Konfiguration von FAI sowie weiterer benötigter Dienste . . . . .       | 7         |
| 3.3.1    | Installation von FAI . . . . .   | 7         |
| 3.3.2    | Konfiguration von FAI . . . . .  | 7         |
| 3.3.3    | Anlegen des Nfsroots . . . . .   | 9         |
| 3.3.4    | Erstellen der für die Client-Installation benötigten Dateien . . . . .                   | 10        |
| 3.3.5    | Konfiguraton weiterer benötigter Dienste . . . . .                                       | 13        |
| 3.4      | Installation des Clients . . . . .   | 14        |
| 3.4.1    | Erzeugen einer Boot-CD . . . . .   | 14        |
| 3.4.2    | Starten der Installation . . . . .   | 15        |
| 3.4.3    | Monitoring und Logging . . . . .   | 15        |
| <b>4</b> | <b>Übergabe und Abnahme durch den Projektverantwortlichen</b>                            | <b>16</b> |
| <b>5</b> | <b>Fazit und Ausblick</b>  | <b>16</b> |
| <b>6</b> | <b>Literaturverzeichnis</b>  | <b>16</b> |
| <b>7</b> | <b>Abweichungen vom Konzeptpapier</b>  | <b>16</b> |
| <b>A</b> | <b>Anlagen</b>   | <b>17</b> |
| A.1      | Protokoll Kickoffmeeting . . . . .   | 17        |
| A.2      | Ganttplan . . . . .  | 18        |
| A.3      | etc/apt/sources.list für den FAI-Server, die FAI-Konfiguration und das Nfsroot . . . . . | 18        |
| A.4      | Startup-Skript für Masquerading mit iptables auf dem Router . . . . .                    | 18        |
| A.5      | fai.conf . . . . .   | 19        |
| A.6      | make-fai-nfsroot.conf . . . . .  | 19        |
| A.7      | update-exim4.conf.conf . . . . .   | 19        |
| A.8      | Dateien im FAI-Ordner /srv/fai/config . . . . .  | 20        |
| A.8.1    | class/50-host-classes . . . . .  | 20        |
| A.8.2    | files/etc/locale.gen/ETCHWS . . . . .  | 20        |
| A.8.3    | scripts/ETCHWS/10-copy-files . . . . .   | 20        |
| A.8.4    | hooks/faiend.ETCHWS . . . . .  | 21        |

|   |    |
|---|----|
| A.9 dhcpd.conf . . . . .                      | 21 |
| A.10 isolinux.cfg . . . . .                   | 21 |
| A.11 boot.msg . . . . .                       | 21 |
| A.12 Übergabe- und Abnahmeprotokoll . . . . . | 23 |

# 1 Projektrahmen

## 1.1 Vorstellung des Unternehmens

Die Firma GeNUA mbH, Gesellschaft für Netzwerk- und Unix-Administration, mit Firmensitz in Kirchheim bei München ist seit 1992 auf Lösungen im Bereich IT-Sicherheit spezialisiert. Während der letzten 15 Jahre wuchs das Unternehmen von 3 auf derzeit 81 Mitarbeiter an. Die Produktpalette umfasst Firewalls, Fernwartungssysteme, VPN-Gateways und Linkbalancer.

## 1.2 Zustandekommen des Projekts

Seit etwa 4 Jahren benutzt die interne Systemverwaltung der Firma GeNUA FAI (Fully Automatic Installation) zur automatischen Installation von Linux-Arbeitsplatzrechnern über das Netzwerk. Momentanes Client-Betriebssystem ist Debian GNU/Linux, Release Sarge. In naher Zukunft werden alle bestehenden Arbeitsplätze auf das Release Etch umgestellt bzw. neue Arbeitsplätze mit Etch installiert werden. Außerdem soll zur Entlastung des derzeitigen FAI-Servers, der zugleich auch als File-, Web-, Postgres- und Nameserver dient, die FAI-Software auf einen eigenen Server umgezogen werden. Aus diesem Anlass bietet es sich an, FAI mit einem aktuellen Release neu zu installieren und eine testweise Installation eines Etch-Clients mit GeNUA-spezifischen Anpassungen bezüglich Partitionierung der Festplatte, Auswahl erforderlicher Software und Anpassung bestimmter Konfigurationsdateien vorzunehmen.

Zu Beginn des Projekts wurden in einem Kickoff-Meeting mit dem Gruppenleiter der internen Systemverwaltung, Herrn Sturm, folgende Punkte diskutiert:

- Projektauftrag und -ziel
- Auftraggeber und Beteiligte
- Anforderungen an Installationssoftware und Testumgebung
- Zeitpunkt der Überprüfung und Anteil der Projektarbeit an der gesamten Aufgabe

Eine genauere Erläuterung der genannten Punkte findet sich im Anhang A.1.

## 1.3 Ist-Zustand

Zu Beginn der Projektarbeit war das bestehende FAI ein Stückwerk aus einer ersten Installation und nach und nach bei Bedarf zugefügten Skripten. Installiert worden war die Software 2003 auf der Basis des Debian-Releases Woody, das seither aus Gründen der Kompatibilität zu anderen Anwendungen nicht mehr geupdatet wurde. Fehler in der Software konnten nicht durch Updates der Version, sondern lediglich durch Eingriffe in den Source-Code gefixt werden. Um eine mit Sarge installierte Arbeitsplatzmaschine zu erhalten, musste auf der Maschine zunächst eine Woody-Installation durchgeführt werden, die anschließend per Skript auf Sarge aktualisiert wird. Sämtliche Konfigurationsdateien und Skripten für verschiedene Anwendungen wurden nach und nach bei Bedarf hinzugefügt, was die Gefahr von Redundanzen und Inkonsistenzen birgt. Außerdem wurde das momentan benutzte FAI nicht dafür eingerichtet, außer Workstations noch andere Arten von Installationen, wie vom Netzwerk unabhängige Einzelplatzmaschinen oder Server, durchzuführen. Eine diesbezügliche Änderung des bestehenden FAI-Servers wäre vermutlich komplizierter als das Aufsetzen eines neuen Systems.

## 1.4 Soll-Zustand

Aus o.g. Punkten entstand der Wunsch nach einem einheitlichen, aktuellen System, das alle Anforderungen von vorneherein berücksichtigt. Die eingesetzte Software sollte folgende Punkte erfüllen:

- Die Installation der Clients wird von einem einheitlichen Bootmedium gestartet.
- Das zu installierende Betriebssystem ist das Debian-Release Etch.
- Die Installation muss unterbrechungsfrei durchlaufen.
- Geeignete Formatierung und Partitionierung der Festplatte, gegebenenfalls auch hostspezifisch, ist eine Grundvoraussetzung für die Installation. Es besteht außerdem die Möglichkeit, vorhandene Partitionen beizubehalten.
- Eine Einteilung der Clients in die Gruppen Workstation, Server oder Standalone-Maschine ist möglich. Clients können dann in Abhängigkeit ihrer Gruppe installiert und konfiguriert werden.
- Dem Benutzer ist es möglich, für unterschiedliche Rechnergruppen verschiedene Softwarepakete zu installieren.
- Die Konfiguration der Clients erfolgt skriptgesteuert.
- Logfiles, die den Ablauf der Installation aufzeichnen, befinden sich nach der Installation des Clients auf dem FAI-Server.
- Das installierte System rebootet fehlerfrei.

Da die skriptgesteuerte Konfiguration eines GeNUA-spezifischen Hostsystems einen erheblichen Aufwand darstellt, einigte man sich darauf, sich für die Projektarbeit auf die von FAI standardmäßig mitgelieferten Skripten zu beschränken, da zunächst nur ein Test der generellen Funktionsfähigkeit der FAI-Software erforderlich war.

Die Lektüre der FAI-Dokumentation<sup>1</sup> hatte ergeben, dass die aktuelle Version allen Anforderungen genügt, so dass FAI auch weiterhin als automatisches Installationstool für Debian-Clients in Betracht gezogen werden kann.

## 1.5 Kosten-Nutzen-Analyse

Eine Etch-Installation inklusive Partitionierung und allen bei GeNUA benötigten Softwarepaketen dauert schätzungsweise eine Stunde. Die manuelle Konfiguration aller Dateien dürfte selbst bei einem routinierten Mitarbeiter mindestens eine weitere Stunde dauern.

Die FAI-Installation beschränkt sich dagegen auf das Einlegen eines Bootmediums und einen kurzen Blick auf die Logfiles nach der Installation, ein Vorgang, der allenfalls 5 Minuten in Anspruch nimmt. Legt man die Kosten einer Mannstunde von 50 Euro zugrunde, ergibt sich daraus folgendes:

|        | <i>manuelle Installation</i> | <i>FAI-Installation</i> |
|--------|------------------------------|-------------------------|
| Dauer  | 120 min.                     | 5 min.                  |
| Kosten | 100 €                        | 4.17 €                  |

---

<sup>1</sup><http://www.informatik.uni-koeln.de/fai/>

Somit ergibt sich eine Differenz von 95.83€ pro installiertem Rechner.

Allerdings muss man noch den Aufwand miteinberechnen, den es kostet, ein funktionsfähiges FAI aufzusetzen. Zum momentanen Zeitpunkt beruht diese Rechnung allerdings auf vorläufigen Schätzwerten:

|   |        |
|---|--------|
| FAI-Server aufsetzen und konfigurieren, Boot-CD erzeugen:                     | 6 h    |
| gruppen- bzw. hardwarepezifische Partitionen, Software und Skripten anpassen: | 80 h   |
| Testinstallation unterschiedlicher Gruppen bzw. unterschiedlicher Hardware:   | 10 h   |
| Dokumentation aller Arbeitsschritte:  | 8 h    |
| <hr/>   |        |
| Summe:  | 104 h  |
| Kosten:   | 5200 € |

Unter Berücksichtigung der oben errechneten Kostenersparnis pro Installation von 95.83€ hat sich ab einer Anzahl von 55 installierten Arbeitsplätzen die Einführung von FAI gelohnt. In Anbetracht von etwa 80 (Tendenz steigend) Mitarbeitern und ebensovielen Arbeitsplatzrechnern bei GeNUA und des zeitnah bevorstehenden Etch-Upgrades dürfte sich die Investition in FAI bereits in wenigen Wochen amortisieren.

Da die Anforderungen an die Serverhardware sehr gering sind (s. Kapitel 2.2), musste ein FAI-Server nicht extra angeschafft werden, sondern konnte auf einer ehemaligen Arbeitsplatzmaschine aufgesetzt werden. Daher bleiben Hardwarekosten unberücksichtigt. Die Software ist Open Source und fließt in keine Berechnung mit ein.

Zudem müssen weitere Faktoren wie das Auftreten einer wesentlich höheren Fehlerquote bei manueller Konfiguration unberücksichtigt bleiben.

## 1.6 Projektschnittstellen

Der im nächsten Abschnitt beschriebene Versuchsaufbau fand zunächst in einer eigenen Testumgebung mit separatem IP-Adressbereich statt. Später sollte er als Grundlage für einen FAI-Server im Netz, in dem sich alle Arbeitsplatzmaschinen befinden, dienen. Die Konfiguration des Testservers wird bis auf kleine Änderungen (Hostnamen, IP-Adressen) vom künftigen, von der internen Systemverwaltung betriebenen, FAI-Server übernommen werden. Der Testserver wird außerdem als Basis für die Installation von Clients mit unterschiedlichen Hardwaretypen (IDE-/ SCSI-Festplatten, unterschiedliche Netzwerkkarten usw.) und zum Testen verschiedener Bootkernel verwendet werden. Nach Abschluss der Projektarbeit sollen schrittweise die GeNUA-spezifischen FAI-Modifikationen übernommen, auf Aktualität überprüft, korrigiert oder ergänzt werden.

## 2 Planung

Bevor es an die eigentliche Umsetzung ging, erfolgten zunächst Planungen zu Testumgebung, Hardware und Ablauf des Projekts.

### 2.1 Planung der Testumgebung

Bei GeNUA gibt es unterschiedliche Netze: Zum einen das Produktivnetz, in dem sich die Workstations der Mitarbeiter und die Server mit den zugehörigen Diensten befinden. Zum anderen existiert das Testnetz, in dem Mitarbeiter eigene Testmaschinen betreiben

können. Testnetz und Produktivnetz sind über einen Router miteinander verbunden. Außerdem gibt es kleinere abteilungsspezifische Netze, die hier aber nicht interessieren. Zunächst sollte der FAI-Server mit allen benötigten Diensten in einer separaten Testumgebung mit eigenen IP-Adressen, dem sog. FAI-Netz, laufen. Dadurch konnte unnötiger Netzwerkverkehr vermieden werden, und Dienste wie DHCP beeinflussten die anderen Netze nicht.

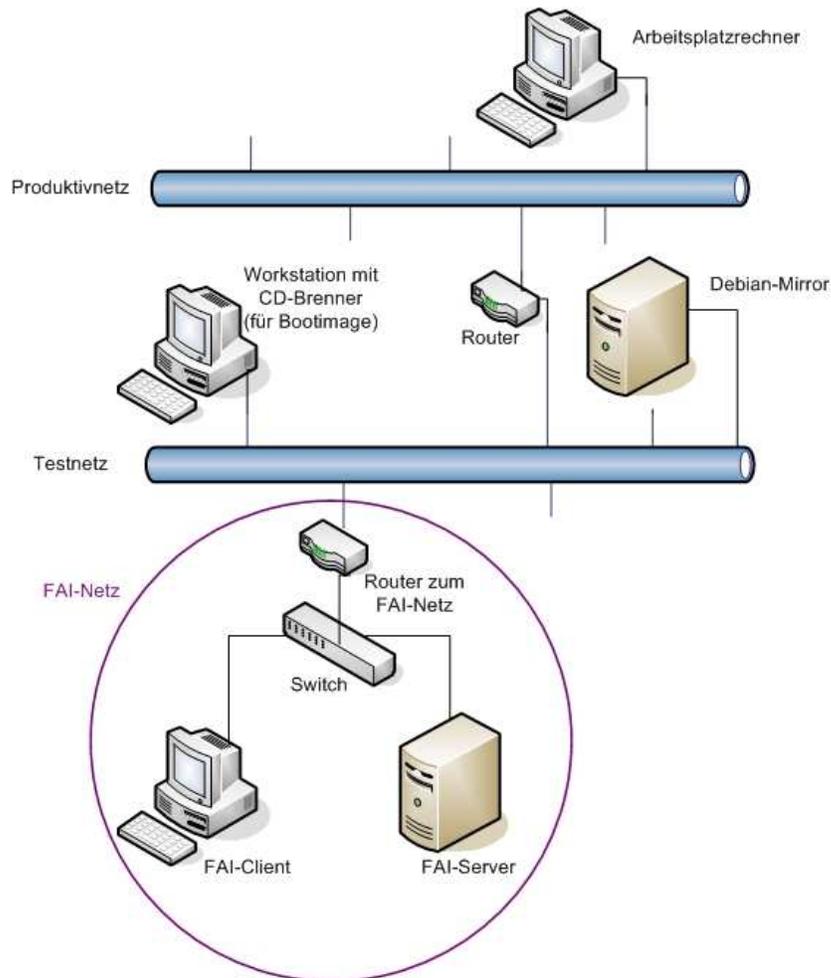


Abbildung 1: Netzplan der Testumgebung und angrenzender Netze

Andererseits musste auch auf Dienste im Testnetz zugegriffen werden können, da zum Installieren eines Clients ein Debian-Mirror sowie zum Brennen der Boot-CDs ein Rechner mit CD-Brenner benötigt wurde. Um ein effizienteres Arbeiten zu ermöglichen, sollte außerdem ein Zugriff auf die FAI-Testumgebung auch vom Arbeitsplatzrechner aus möglich sein. Für den Datenaustausch mit anderen Netzen musste daher ein weiterer PC als Router konfiguriert werden.

## 2.2 Testhardware

Benötigt wurden drei Testmaschinen: ein FAI-Server, ein zu installierender Client und ein Router zum Testnetz. Ein vorinstallierter Router mit zwei Netzwerkkarten war bereits vor-

handen und konnte ohne größere Konfigurationsänderungen verwendet werden.

Für den Server wurde eine Maschine mit einer zweiten Festplatte ausgewählt, da zu Beginn noch nicht klar war, wie viel Platz die FAI-Installation mit allen zugehörigen Komponenten und Konfigurationsdateien einnehmen würde.

Der FAI-Client benötigte als Voraussetzungen eine Netzwerkkarte, ein funktionierendes CDROM-Laufwerk und eine Festplattenkapazität von mindestens 3 GB. Da FAI eine rein textbasierte Anwendung ist, war mit keinen allzu hohen Anforderungen an die Hardware zu rechnen. Die einzelnen Testrechner wiesen folgende Hardwarekomponenten auf.

Router:

CPU: GenuineIntel, Pentium II (Deschutes), 334 Mhz

RAM: 256 MB DDR-SDRAM

HD: 31 GB IDE

Server:

CPU: AMD-K6(tm) 3D processor, 350 MHz

RAM: 128 MB DDR-SDRAM

HD: 14 GB, 19 GB IDE

Client:

CPU: AMD-K6(tm) 3D processor, 400 MHz

RAM: 512 MB DDR-SDRAM

HD: 6 GB IDE

## 2.3 Ablaufkonzept

Als erste Schritte waren die Grundinstallation des FAI-Servers von CDROM sowie die Konfiguration des Routers zum Testnetz geplant. Danach sollte die Installation von FAI und weiterer benötigter Software erfolgen. Das Vorhandensein eines funktionsfähigen FAI-Servers stellt eine Grundvoraussetzung für das weitere Vorgehen dar und wurde daher zum Meilenstein für das Projekt bestimmt.

Nach der Einrichtung des Servers waren das Erstellen einer Boot-CD sowie die Installation des FAI-Clients vorgesehen. Nach deren erfolgreichen Ausführung war zu überprüfen, ob ein Logging der einzelnen Schritte stattgefunden hat und ob ein erfolgreicher Reboot des Clients durchgeführt werden kann.

## 3 Realisierung

Nach der Planungsphase fand die Umsetzung der Testumgebung statt. Dazu gehörten die Installation des FAI-Servers, die Konfiguration des Routers, Installation und Konfiguration der FAI-Software und weiterer für FAI benötigter Dienste, das Vorbereiten einer Boot-CD und die Installation eines Testclients.

### 3.1 Installation und Konfiguration des Servers

Als Betriebssystem für den Server diente das Debian-Release Etch, das zum Entstehungszeitpunkt dieser Arbeit auch die Bezeichnung „Testing“ hatte. Installationsmedium war eine CDROM. Als Installationsart wurde eine Standardpartitionierung (eine /-Partition

und eine swap-Partition) und ein Basissystem ohne weitere Dienste gewählt, da diese im späteren Verlauf von Hand bzw. über Abhängigkeiten installiert und konfiguriert wurden. Die Konfiguration des Netzwerks fand bereits während der Installation statt, da Debian an geeigneter Stelle entsprechende Fragen stellt. Für das FAI-Netz wurde der IP-Adressbereich 192.168.1.0/24 vergeben. Folgende Parameter wurden bei der Installation festgelegt:

|                               |               |
|-------------------------------|---------------|
| Bezeichner der Netzwerkkarte: | eth0          |
| Art der IP-Adressvergabe:     | statisch      |
| IP-Adresse:                   | 192.168.1.10  |
| Hostname:                     | server        |
| Domainname:                   | azubis.de     |
| Default gateway:              | 192.168.1.254 |
| DNS-Server:                   | 10.0.1.4      |

Um FAI und weitere benötigte Softwarepakete zu installieren, war es außerdem nötig, als Installationsmethode http sowie den entsprechenden Debian-Mirror und das Release anzugeben. Da während der Installation noch kein Routing zum Testnetz bestand, war als bisherige Installationsmethode „cdrom“ angegeben worden.

Nach der Konfiguration des Routers erfolgte für weitere Installationen von Softwarepaketen das Editieren der Datei /etc/apt/sources.list. In dieser Datei werden die Installationsquellen, das Paketverzeichnis, die Version (stable, testing...) und der Typ (main, contrib, non-free) gespeichert. Eingetragen wurden der firmeneigene Mirror im Testnetz und das Verzeichnis etch. Die vollständige sources.list findet sich im Anhang A.3.

### 3.2 Konfiguration des Routers

Auf dem PC, der als Router dienen sollte, befand sich aufgrund früherer Tests bereits ein installiertes Debiansystem (Release Sarge) mit zwei konfigurierten Netzwerkkarten, eine für das FAI-Netz mit der Adresse 192.168.1.254 und eine für das Testnetz mit der Adresse 10.1.254.148.

Die einzige Anpassung, die nötig wurde, war NAT<sup>2</sup> für das FAI-Netz in Richtung Testnetz. Hintergrund ist, dass im Testnetz keine Route zum FAI-Netz bestand. Somit mussten alle IP-Pakete, die aus dem FAI-Netz über den Router ins Testnetz gingen, als Quelladresse eine 10er-Adresse (sinnvollerweise die des Routers) bekommen. Realisiert wurde dies durch Masquerading als spezielle Form von Source-NAT mit iptables, ein Linux-Tool für Paketfilterung und NAT. Das benötigte Kommando lautet:

```
iptables -t nat -A POSTROUTING -s 192.168.1.0/255.255.255.0 \  
-o eth0 -j MASQUERADE
```

Alle IP-Adressen des FAI-Netzes wurden somit durch die IP-Adresse des äußeren Router-Interfaces (eth0) maskiert. Die Wirkung eines solchen Kommandos besteht jedoch nur bis zum nächsten Reboot. Um iptables bereits beim Booten zu starten, musste im Ordner /etc/init.d ein Startup-Skript erstellt und im Verzeichnis des entsprechenden Runlevels<sup>3</sup> verlinkt werden. Das zugehörige Kommando lautete:

<sup>2</sup>Network Address Translation, Adressinformationen von Datenpaketen werden automatisiert und transparent durch andere ersetzt

<sup>3</sup>Ein Runlevel definiert die Konstellation aktiver Prozesse eines Systems, die während des Bootvorgangs initiiert wurden.

```
ln -s /etc/init.d/nat.sh /etc/rc2.d/S98-natting
```

Der Code für das Startup-Skript befindet sich im Anhang A.4.

### 3.3 Installation und Konfiguration von FAI sowie weiterer benötigter Dienste

Vor der eigentlichen FAI-Installation wurde zunächst `ssh`<sup>4</sup> als Remote-Verbindung installiert, um künftig vom eigenen Arbeitsplatz aus auf dem Server zugreifen zu können.

Nach Ausführung des Befehls „`aptitude`<sup>5</sup> install ssh“ war es möglich, sich mit dem Befehl „`ssh root@server`“ über den Router auf den Server einzuloggen und vom Arbeitsplatz aus weiterzuarbeiten. Danach erfolgte die eigentliche FAI-Installation.

#### 3.3.1 Installation von FAI

FAI besteht aus drei Paketen: `fai-server` (ausführende Shell- und Perlskripten), `fai-kernels` (vorkonfiguriertes Kernelpaket zum Booten mit FAI, optional) und `fai-doc` (Beispiele für Partitionierung und Konfiguration eines Clients, optional). Außerdem wird bei FAI ein DHCP-Server mitinstalliert (`dhcp3-server`). Es wurde die zum Installationszeitpunkt aktuellste FAI-Version, 3.1.7, verwendet. Ein weiteres von FAI mitgeliefertes Paket, `ftpd-hpa`, fand keine Verwendung.

#### 3.3.2 Konfiguration von FAI

Vor der Erläuterung der einzelnen Konfigurationsschritte soll kurz auf das allgemeine Konzept von FAI eingegangen werden.

Das eigentliche Konfigurationsverzeichnis befindet sich nach der Installation von FAI im Ordner `/etc/fai`. Dort liegen die Konfigurationsdateien `fai.conf`, `make-fai-nfsroot.conf`, `NFS-ROOT` und `apt/sources.list`. Innerhalb dieser Dateien werden Shellvariablen definiert, die im Lauf der Installation durch die FAI-Skripten ausgewertet werden.

Nach der FAI-Konfiguration erfolgt als nächster Schritt das Anlegen eines Nfsroots. Dieses stellt das Verzeichnis `/srv/fai/nfsroot` dar. Zu Beginn der Installation des Clients wird das Nfsroot-Verzeichnis des FAI-Servers zum Wurzel- oder `/`-Verzeichnis des FAI-Clients. Im weiteren Verlauf des Installationsvorgangs legt FAI im `/tmp`-Verzeichnis des Clients einen Ordner namens „`target`“ an. In diesem wird der Verzeichnisbaum für den installierten Client eingerichtet. Das `/tmp/target`-Verzeichnis ist zunächst ein Abbild des Nfsroots, kann aber mittels Skripten und Konfigurationsdateien nach eigenem Bedarf eingerichtet werden.

Um einen Client zu installieren, benötigt FAI ein Verzeichnis, das sich standardmäßig unter `/srv/fai/config` befindet und Skripten und Konfigurationsdateien enthält. Dieser Ordner wird zu Beginn des Installationsvorgangs in das Verzeichnis `/var/lib/fai` des FAI-Clients gemountet (s. Abb. 2).

Nach dem Reboot des Clients wird sein ehemaliges `/tmp/target`-Verzeichnis zum `/`-Verzeichnis<sup>6</sup>.

Die Grafik auf der nächsten Seite soll den Zusammenhang zwischen den Verzeichnissen des FAI-Servers und -Clients verdeutlichen.

<sup>4</sup>Secure Shell, ursprünglich als OpenSSH vom OpenBSD-Projekt entwickelt

<sup>5</sup>Installationstool unter Debian, erkennt Abhängigkeiten zwischen Softwarepaketen und installiert benötigte Pakete automatisch mit

<sup>6</sup>Das Wurzelverzeichnis eines UNIX-Systems

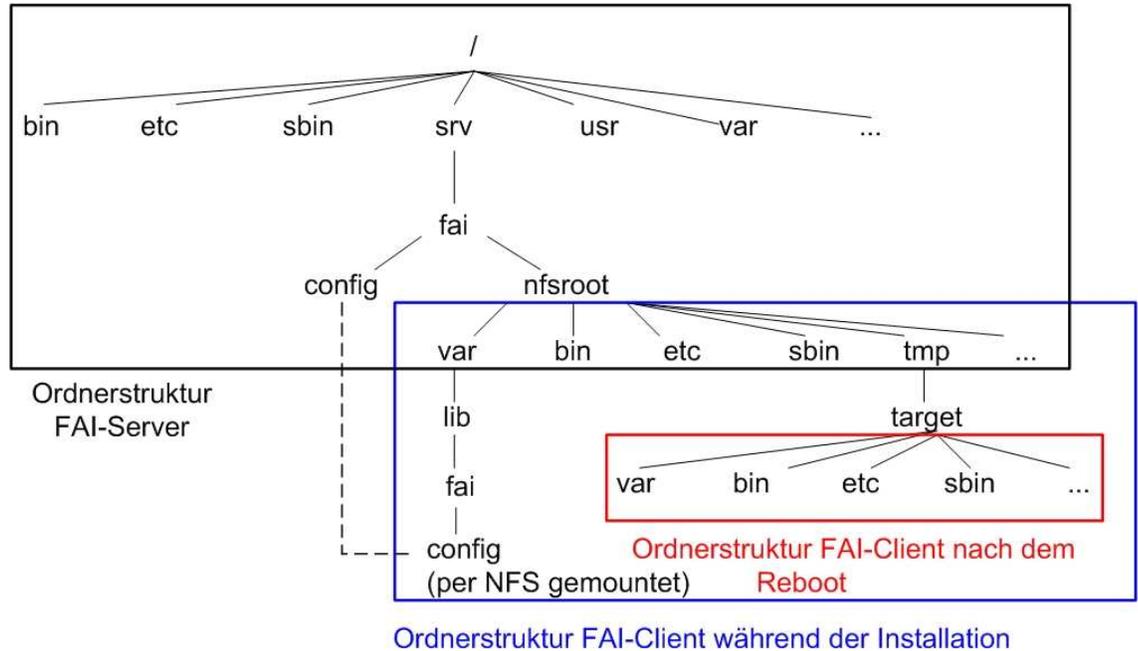


Abbildung 2: Ordnerstruktur von FAI-Server und -Client

Zunächst erfolgte eine Konfiguration der Dateien innerhalb des Verzeichnisses `/etc/fai`. In der Datei `fai.conf` können folgende Variablen gesetzt werden:

- `FAI_CONFIGDIR`: Das Verzeichnis, in dem sich die Dateien für die Installation und Konfiguration des Clients befinden (Default: `/srv/fai/config`)
- `FAI_CONFIG_SRC`: per NFS (s.3.3.5) gemountetes Konfigurationsverzeichnis auf dem Client
- `FAI`: Der Mountpoint des unter `FAI_CONFIGDIR` angegebenen Verzeichnisses auf dem Client
- `FAI_LOGPROTO`: Methode, wie die Logfiles (s.Kapitel 3.4.3) nach der Installation vom Client auf den Server gelangen (ssh, rsh oder ftp).
- `LOGUSER`: Name des Benutzers, unter dem die Logdateien auf den Logserver übertragen werden (Default: `fai`)
- `MNTPOINT`: Mountpoint zum Übertragen von Softwarepaketen vom Mirror auf den Client (nur, wenn die Übertragung nicht per http, sondern per NFS erfolgt)
- `LOGSERVER`: Name oder IP-Adresse des Servers, auf dem die Logdateien gespeichert werden. Wird hier nichts eingetragen, sind die Logdateien nach dem Reboot des Clients gelöscht.

Die Konfigurationsdatei `make-fai-nfsroot.conf` wird zum Aufsetzen des Nfsroots (s. nächstes Kapitel) benötigt und enthält folgende Einträge:

- `NFSROOT`: Das Nfsroot-Verzeichnis auf dem Server (Default: `/srv/fai/nfsroot`)

- NFSROOT\_ETC\_HOSTS: Eintrag zur Namensauflösung, falls kein Nameserver vorhanden (nur 1 Eintrag zulässig)
- FAI\_DEBOOTSTRAP: Host-/Verzeichnisname, aus dem zusätzliche Software installiert wird
- FAI\_ROOTPW: MD5-Hash zum Einloggen während der Installation (Standard-Passwort: fai)
- KERNELPACKAGE: Pfad zum Kernelpackage, das zum ersten Booten des Clients verwendet wird
- FAI\_DEBOOTSTRAP\_OPTS: Optionen zum Installieren des Nfsroot

Die verwendeten Konfigurationsdateien befinden sich im Anhang A.5 und A.6.

Die Datei sources.list enthielt den Namen des Mirrors und das Releaseverzeichnis, von dem das Nfsroot und später der Client installiert werden. Diese Datei war identisch mit der sources.list im /etc/apt-Verzeichnis des Servers (A.3).

Die Datei NFSROOT wird im nächsten Kapitel behandelt.

### 3.3.3 Anlegen des Nfsroots

Das Nfsroot ist das temporäre /-Verzeichnis des FAI-Clients während der Installation.

Da eine Anforderung war, dass der Client nach der Installation eine Hinweis-E-Mail schicken soll, musste innerhalb des Nfsroots zusätzliche Software installiert werden: Exim als MTA und mailutils als Werkzeug zum kommandozeilenbasierten Mailen. Der Eintrag „exim4 mailutils“ in die Datei /etc/fai/NFSROOT bewirkt, dass diese Pakete beim Erstellen des Nfsroots mitinstalliert werden.

Zum Einrichten des Nfsroots wird folgender Befehl ausgeführt:

```
fai-setup -v (für verbose)
```

Dieses Kommando löscht das Nfsroot-Verzeichnis (falls bereits vorhanden) und installiert es neu. Folgende Meldungen deuten auf eine erfolgreiche Ausführung hin:

```
make-fai-nfsroot finished properly.  
FAI setup finished
```

Nun konnte der Exim innerhalb des Nfsroots konfiguriert werden.

Für die Einrichtung des Mailclients musste zunächst die Datei /etc/exim4/exim4.conf.template bearbeitet werden. Zu beachten ist nämlich, dass während der Installation das Nfsroot-Verzeichnis nur lesbar gemountet ist und somit keine E-Mails im Spoolverzeichnis, welches ein- und ausgehende E-Mails zwischenlagert, gespeichert werden können. Einzige Ausnahme stellt das /tmp-Verzeichnis dar, das les- und schreibbar gemountet wird. Daher muss das Spoolverzeichnis von /var/spool/exim4 auf /tmp umgemappt werden. Dies geschieht durch Setzen der Variable:

```
SPOOLDIR = /tmp
```

Desweiteren sollten E-Mails an den Mailserver bei GeNUA ausgeliefert werden. Die entsprechenden Einstellungen wurden in der Datei /etc/exim4/update-exim4.conf.conf getätigt (s. Anhang A.7).

Der Test erfolgte per mail-Kommando aus der Chroot-Umgebung:

```
echo "mail von nfsroot" | mail -s 'test' drexel@genua.de
```

Da die E-Mail ankam, wurde deutlich, dass der Exim korrekt konfiguriert war.

### 3.3.4 Erstellen der für die Client-Installation benötigten Dateien

Für die Installation und Konfiguration des FAI-Clients benötigt FAI das Verzeichnis `/srv/fai/config`. Die einfachste Methode, dieses Verzeichnis zu erzeugen, besteht darin, eine Vorlage aus dem FAI-Examples-Ordner<sup>7</sup> nach `/srv/fai/config` zu kopieren. Dazu wird der Benutzer während des Anlegens des Nfsroots aufgefordert. Das so entstandene Verzeichnis `/srv/fai/config` enthält folgende Unterordner: `class`, `debconf`, `disk_config`, `files`, `hooks`, `package_config` und `scripts`. In diesen befinden sich bereits exemplarische Konfigurationsdateien, die beliebig ersetzt oder ergänzt werden können.

Zum besseren Verständnis für den Ablauf einer FAI-Installation soll hier kurz auf die Unterordner des config-Verzeichnisses in der Reihenfolge, in der FAI sie abarbeitet, eingegangen werden:

- **class:** FAI konfiguriert Hosts unterschiedlich, abhängig davon, welchen Klassen der Client angehört. Der Benutzer kann eigene Klassen definieren. Zusätzlich benutzt FAI eigene Standardklassen, die z.B. aufgrund von Prozessortyp oder zu installierendem Betriebssystem definiert sind.
- **disk\_config:** Mit FAI können Partitionen beliebiger Größe festgelegt werden. Anzugeben sind außerdem der Filesystem-Typ und der Mountpoint.
- **package\_config:** In diesem Ordner sind Paketlisten anzugeben, die klassenabhängig auf dem FAI-Client zu installieren sind.
- **debconf:** Der FAI-Benutzer erhält zusätzlich die Möglichkeit, Pakete nach der Installation zu konfigurieren.
- **scripts:** In diesem Verzeichnis werden Skripten abgelegt, die während der Installation des Testrechners ausgeführt werden. Akzeptiert werden Shell, Perl und cfengine<sup>8</sup> (cfagent).
- **hooks:** Zusätzlich gibt es Skripten, die zu einem bestimmten Zeitpunkt ausgeführt werden sollen (z.B. vor der Installation der Softwarepakete). Diese werden im Ordner „hooks“ gespeichert.
- **files:** Mit FAI lassen sich Vorlagen von Dateien vom config-Verzeichnis auf den Client kopieren. Die Pfade sind relativ zum „files“-Ordner.

**Einteilung in Klassen** Standardmäßig erstellt FAI bei Linux-Installationen auf einer x86-Architektur folgende Klassen: `DEFAULT`, `LINUX`, `I386`, `LAST` und eine Klasse, die den Hostnamen des Testrechners trägt, in diesem Fall `faiclient`. Innerhalb des Unterordners „class“ befindet sich außerdem ein Skript (`50-host-classes`), das den FAI-Client weiteren Klassen zuordnet: `FAIBASE`, `DHCP` und `GRUB`. Alle Klassennamen werden, mit Ausnahme des Hostnamens, grundsätzlich groß geschrieben. Um den Testrechner später nach eigenen Vorstellungen konfigurieren zu können, wurde die Möglichkeit getestet, weitere eigene Klassen hinzuzufügen: `GENUA` als Klasse, die allen installierten Rechnern gemeinsam sein soll, und `ETCHWS` als eine Klasse speziell für Workstations. In diesem Fall wurde exemplarisch eine Workstation installiert, die sich in insgesamt 10 Klassen befand (Abb. 3).

<sup>7</sup>`/usr/share/doc/fai-doc/examples`

<sup>8</sup>network configuration engine; Programmierumgebung zum automatischen Konfigurieren von UNIX-Hosts

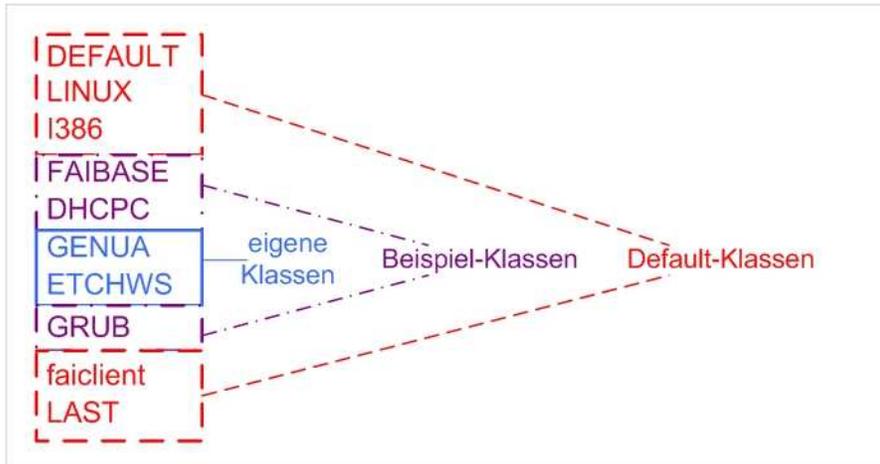


Abbildung 3: Zur Testinstallation verwendete FAI-Klassen, Reihenfolge von oben nach unten entspricht der zeitlichen Reihenfolge der Klassenzuordnung

Um die zusätzlichen Klassen hinzuzufügen, wurde das Skript „50-host-classes“ durch die Einträge „GENUA“ und „ETCHWS“ ergänzt (vgl. Anhang A.8.1).

Die Klassen spielen bei FAI insofern eine Rolle, als Konfigurationsdateien bzw. Skripten in der Reihenfolge abgearbeitet werden, in der Klassen definiert wurden (DEFAULT wird somit immer zuerst abgearbeitet, LAST zuletzt). Zum Teil wird aber auch nur eine Klasse, nämlich diejenige, die als letzte definiert wurde, berücksichtigt. Herauszufinden, bei welchem Installationsschritt wie viele Klassen ausgewertet werden, war ebenfalls Aufgabe dieser Projektarbeit.

**Partitionierung der Festplatte** Bei GeNUA werden die Festplatten von Workstations einheitlich partitioniert: 23 MB für die /boot-Partition, 512 MB swap, 10 GB für die /-Partition und der Rest für eine Partition namens space. Letztere dient zum Zwischenlagern persönlicher Daten, die nicht zwingend im /home-Verzeichnis gespeichert werden, welches vom Backup erfasst wird. Da die Festplatte des FAI-Clients nur eine Größe von 6 GB hatte, wurde hier ein eigenes Partitionierungsschema für die Klasse „faiclient“ angelegt. Dadurch konnte auch gleich das Funktionieren einer hostspezifischen Klasse getestet werden. Um eine individuelle Partitionierung zu gewährleisten, wurde im Ordner „disk.config“ folgende Datei mit dem Namen „faiclient“ erstellt:

```
# <type> <mountpoint> <size in mb> [mount options]      [;extra options]
disk_config disk1
# 1 Disk, SCSI or IDE
primary          /boot                23      defaults,grpid  ;-j ext3 boot
primary          swap                  256     defaults
primary          /                    5000    defaults,grpid  ;-j ext3
primary          /space                 1-      defaults,grpid  ;-j ext3 lazyformat
```

In diesem Fall wurden 4 primäre ext3-Partitionen mit 23 MB, 256 MB, 5 GB und dem verbleibenden Plattenplatz (ca. 800 MB) erstellt.

Die Option „lazyformat“ bei der /space-Partition sorgt dafür, dass diese Partition bei zukünftigen Neuinstallationen erhalten bleibt. Um dies zu testen, wurde innerhalb der

/space-Partition eine Datei namens „test.txt“ hinterlegt. Nach einer zweiten Installation konnte verifiziert werden, ob diese Datei im /space-Verzeichnis zu finden war.

Es stellte sich heraus, dass bei der Partitionierung nur eine FAI-Klasse, in diesem Fall die Klasse „faicient“, ausgewertet wird.

**Installation von Softwarepaketen** Um festzulegen, welche Debianpakete FAI installieren soll, legt man im Ordner „package\_config“ eine oder mehrere Dateien mit entsprechendem Klassennamen an. Die Datei muss eine Kopfzeile und die Auflistung aller entsprechenden Paketnamen aufweisen. Um eine klassenabhängige Installation zu verifizieren, wurde eine Datei namens GENUA mit folgendem Inhalt erzeugt:

```
PACKAGES install
linux-image-2.6.20.1
gpm
```

Ein derartiger Eintrag stellt sicher, dass der 2.6.20-Kernel und das Debian-Paket „gpm“ auf Maschinen der Klasse GENUA installiert werden. FAI wertet alle Dateien innerhalb des „package\_config“-Ordners aus, deren Name eine zum Client zugehörigen Klasse ist, und installiert die Softwarepakete anschließend in alphabetischer Reihenfolge.

**Ausführen von Skripten und Übertragen von Dateien** Zur Ausführung von Skripten auf dem FAI-Client müssen innerhalb des Verzeichnisses „scripts“ Unterverzeichnisse mit dem entsprechenden Klassennamen angelegt werden. Die Namen der auszuführenden Skripten werden innerhalb dieser Unterverzeichnisse gespeichert und beginnen mit einer Nummer (z.B. „10-copy-files“).

FAI führt die Skripten in der Reihenfolge der definierten Klassen aus (DEFAULT, LINUX, ... LAST). Innerhalb des Klassenordners werden sie in Reihenfolge der Nummerierung ihres Dateinamens abgearbeitet. Zu beachten ist außerdem, dass es sich bei einem Skript um eine ausführbare Datei handeln muss, sonst wird das Skript ohne Fehlermeldung von FAI übergangen.

Zum Testen wurde ein Skript für die Klasse ETCHWS erstellt, das Dateien auf den FAI-Client kopiert. Die entsprechenden Dateien müssen dann im Verzeichnis „files“ im jeweiligen Pfad mit dem korrekten Klassennamen hinterlegt sein. Zum Testen wurden folgende Dateien hinterlegt und kopiert: files/etc/apt/sources.list/ETCHWS und files/etc/locale.gen/ETCHWS<sup>9</sup>. Das Skript und die entsprechenden Dateien befinden sich im Anhang A.8.2.

Testweise wurde außerdem eine sources.list für die Klasse GENUA in files/etc/apt/sources.list hinterlegt. Es stellte sich heraus, dass FAI lediglich eine Datei kopiert, nämlich die, die zur später definierten Klasse gehört (in d. Fall ETCHWS).

Zusätzlich zur Abarbeitung des „scripts“-Verzeichnisses gibt es die Möglichkeit, Skripten an einem bestimmten Punkt der FAI-Installation auszuführen. Zum Testen dieses Features muss im Ordner „hooks“ ein Skript erstellt werden, dessen Name aus einem Prefix und einem Klassennamen, getrennt durch einen Punkt, besteht. Das Prefix ist der Name eines sog. „Tasks“. Tasks sind Subroutinen, die während der einzelnen Installationsschritte von FAI ausgeführt werden, z.B. „defclass“ zum Zuordnen der Klassen oder „partition“ zum Partitionieren der Festplatte. Ein Skript mit dem Namen „partition.FAIBASE“ würde beispielsweise vor dem Task „partition“ ausgeführt werden.

---

<sup>9</sup>Pfadangabe relativ zum Verzeichnis /srv/fai/config

Im Fall der Testinstallation sollte am Schluss ein Skript ausgeführt werden, das den Benutzer per E-Mail darauf aufmerksam macht, dass die Installation beendet ist. Die Datei wurde unter dem Namen „faiend.ETCHWS“ gespeichert. Dies bedeutet, dass sie vor dem Task „faiend<sup>10</sup>“ für die Klasse ETCHWS ausgeführt wird. Das Skript faiend.ETCHWS befindet sich im Anhang A.8.4.

An welcher Stelle welche Skripte ausgeführt werden, lässt sich anhand der Logdateien (s. 3.4.3) nachvollziehen.

### 3.3.5 Konfiguraton weiterer benötigter Dienste

**NFS** Wie bereits erwähnt, mountet der von CD gebootete Client per NFS (Network File System) das Nfsroot-Verzeichnis des Servers nach /. Um dem Client den Zugriff auf die entsprechenden Verzeichnisse des FAI-Servers zu ermöglichen, muss auf diesem ein NFS-Dienst laufen. Zunächst musste NFS mit „aptitude install nfs-kernel-server“ installiert werden. Eine Installation **vor** Anlegen des Nfsroots empfiehlt sich, da in diesem Fall der NFS-Dienst automatisch konfiguriert wird. Zur Kontrolle, ob die benötigten Einstellungen vorgenommen wurden, empfiehlt sich ein Blick in die Datei /etc/exports. Diese wies die folgenden beiden Einträge auf:

```
/srv/fai/config 192.168.1.0/24(async,ro,no_subtree_check)
/srv/fai/nfsroot 192.168.1.0/24(async,ro,no_subtree_check,\
    no_root_squash)
```

**DHCP** Der Client besitzt beim Booten keine IP-Adresse. Andererseits muss er zum Mounten des /-Verzeichnisses über eine IP-Adresse verfügen, um auf den NFS-Server zugreifen zu können. Die Lösung hierfür besteht in der Einrichtung eines DHCP-Servers (Dynamic Host Configuration Protocol), der Clients IP-Adressen automatisch zuweist.

Das Paket dhcp3-server DHCP wird bei der Installation von FAI gleich mitgeliefert und braucht daher nicht extra installiert zu werden. Als Konfigurationsparameter wurden in die Datei dhcpd.conf Domainname, Netzmaske und Nameserver eingetragen. Eine vollständige Konfigurationsdatei befindet sich im Anhang A.9.

Die Festlegung einer Range, innerhalb derer IP-Adressen vergeben werden können, erfolgte nicht. Stattdessen wurde der zu installierende Client mit einer fixen IP-Adresse eingetragen. Der dhcpd.conf-Eintrag für den Host „faiclient“ lautete folgendermaßen:

```
host faiclient {
    hardware ethernet 00:90:27:75:48:d7;
    option host-name "faiclient";
    fixed-address 192.168.1.209;
}
```

Da die Fertigstellung des FAI-Servers ein Meilenstein des Projektes darstellte, überprüfte Herr Sturm an dieser Stelle verschiedene Punkte, z.B. das Vorhandensein und Funktionieren der Chroot-Umgebung, die per FAI konfigurierten Parameter und das ordnungsgemäße Starten der Dienste NFS und DHCP. Da es hier keine Probleme gab, konnte der nächste Schritt, die Installation eines FAI-Clients, erfolgen.

<sup>10</sup>Beendet die FAI-Installation und gibt auf der Shell aus, dass der Benutzer den FAI-Client rebooten kann

### 3.4 Installation des Clients

Die FAI-Client-Installation gliederte sich in die Schritte Erstellen eines Bootmediums, die eigentliche Installation und das Überprüfen derselben anhand von Logdateien.

#### 3.4.1 Erzeugen einer Boot-CD

Der nächste Schritt war die Herstellung eines bootfähigen Mediums. Dieses muss in der Lage sein, den FAI-Client soweit zu booten, bis er das Nfsroot in sein /-Verzeichnis mounten kann. Erst danach kann das FAI-Startskript ausgeführt werden.

Das Bootmedium sollte für alle Hosts einheitlich verwendet werden. Da alle PCs bei GeNUA über ein CDROM-Laufwerk verfügen, wurde beschlossen, alle FAI-Clients per CDROM zu booten. Als Bootmethode wurde isolinux<sup>11</sup> gewählt. Um ein Isolinux-Bootimage zu erzeugen, erstellt man ein Verzeichnis namens isolinux. Darin sollten folgende Dateien enthalten sein:

- das Binary isolinux.bin
- die Datei memdisk
- die Konfigurationsdatei isolinux.cfg
- eine Textdatei als Bootmessage (optional)
- ein Bootkernel

Die Dateien isolinux.bin und memdisk können von der Syslinux-Homepage<sup>12</sup> heruntergeladen werden. Die Datei isolinux.cfg enthält Variablen mit allen Informationen, die den Client beim Booten mitgegeben werden, z.B.:

- ein Auswahlmenü für den Fall, dass mit unterschiedlichen Kernen gebootet werden kann (in d. Fall unnötig)
- ein Timeout, innerhalb dessen der Benutzer eine Auswahl treffen kann
- den Namen des Bootkernels
- die Art der IP-Addressvergabe (hier dhcp)
- den Pfad zum Nfsroot
- den Namen der Datei, die einen Bootprompt enthält (hier boot.msg)
- Zusatzoptionen (Starten eines ssh-Servers auf dem Client, Anlegen virtueller Terminals, mehr Ausgaben auf der Konsole)

Als Bootkernel war bereits ein von einem Mitarbeiter erstelltes Kernepaket (2.6.20) auf dem Mirror vorhanden, das passende Module für alle bei GeNUA betriebenen Maschinen enthält. Aus diesem Kernepaket wurde der eigentliche Kernel extrahiert und unter dem Namen vmlinux in den isolinux-Ordner verschoben.

Nach Erstellen aller benötigten Dateien, ließ sich das Isoimage erzeugen. Zunächst musste hierfür das Paket „mkisofs“ installiert werden. Danach wechselt man eine Ebene über das Verzeichnis, in dem sich der isolinux-Ordner befindet und erzeugt das Image mit folgendem Befehl:

---

<sup>11</sup>Bootloader für Linux zum Erstellen bootfähiger CDROMS

<sup>12</sup><http://www.kernel.org/pub/linux/utils/boot/syslinux/>

```
mkisofs -o fai_cd_etch.iso -r -b isolinux/isolinux.bin -no-emul-boot \  
-boot-load-size 4 -boot-info-table -c isolinux/boot.cat bootcd
```

wobei bootcd der Ordnername ist, in dem sich das isolinux-Verzeichnis befindet.

Das entstandene Isoimage wurde per scp (Secure Copy) auf einen Rechner mit CD-Brenner im Testnetz übertragen und dort mit dem cdrecord-Kommando auf eine CDROM gebrannt.

### 3.4.2 Starten der Installation

Ist alles korrekt konfiguriert, muss der Benutzer lediglich den FAI-Client ans Netzwerk anschließen, einschalten und eine Boot-CD einlegen. Rückmeldungen zu den einzelnen Installationsschritten erfolgen direkt auf der Shell und in Logdateien (s.u.).

Ist die Installation beendet, kann die CDROM entnommen und der Rechner neu gebootet werden.

Insgesamt wurden drei Installationen vorgenommen, eine für einen ersten Test, eine zweite zur Fehlerkorrektur und eine dritte für eine abschließende Überprüfung aller Anforderungen.

### 3.4.3 Monitoring und Logging

FAI speichert zu jedem Installationsschritt Logfiles im Verzeichnis /tmp/fai. Beispiele für die wichtigsten Logdateien sind:

- FAI.CLASSES: In welche Klassen wurde der Client einsortiert ?
- partition.hda: Wie wurde die Festplatte formatiert ?
- software.log: Welche Softwarepakete wurden installiert und in welcher Reihenfolge?
- fcopy.log: Welche Dateien aus dem files-Verzeichnis wurden auf den FAI-Client kopiert ?
- status.log: Rückgabewert der ausgeführten Skripten (OK oder FAILED + Exitcode)
- shell.log: Ausgaben der ausgeführten Skripten
- error.log: gesammelte Fehlermeldungen über die ganze Installation

Außerdem werden alle Logmeldungen, die beim FAI-Client direkt auf der Shell ausgegeben werden, in der Datei fai.log gespeichert. Ist ssh aktiviert, kann man sich auf dem FAI-Client einloggen und mit dem Befehl

```
tail -f fai.log
```

die Installation live mitverfolgen.

Wird der FAI-Client nach der Installation neu gebootet, wird das /tmp-Verzeichnis (mit Ausnahme des Unterverzeichnisses „target“) geleert. Damit verschwinden auch die Logfiles. Um diese auch später noch zur Verfügung zu haben, sollten diese auf einem Server gespeichert werden. Dies kann durch richtiges Setzen der Variable LOGSERVER in der Datei fai.conf (s. 3.3.2) erreicht werden. Am sinnvollsten ist es, als Logserver den FAI-Server anzugeben. Dort werden die Dateien standardmäßig im Verzeichnis /var/log/fai/<hostname>/install-<Datum>.<Uhrzeit> gespeichert.

## 4 Übergabe und Abnahme durch den Projektverantwortlichen

Zum Abschluss fand ein letztes Meeting zum Überprüfen der Ergebnisse statt. Hierfür wurde ein weiterer Testlauf vorgenommen. Die getesteten Punkte sind im Protokoll im Anhang A.12 aufgeführt. Nach erfolgreicher Überprüfung wurde das Projekt von Herrn Sturm abgenommen.

## 5 Fazit und Ausblick

Nach Überprüfung der Installation zeigte sich, dass FAI durchaus als automatisches Installationstool für Maschinen, die unter Etch betrieben werden sollen, geeignet ist. Aufbauend auf den gewonnenen Erkenntnissen können künftig weitere Schritte vorgenommen werden, z.B. Installation von Clients mit anderer Hardwareausstattung, Installation und Konfiguration weiterer Rechnerklassen (Server, Standalonemaschinen/Laptops), Update der GeNUA-spezifischen Paketlisten von Sarge auf Etch und Überprüfung der Funktionalität der Skriten vom alten FAI-Server bei der Ausführung unter Etch. Für diese Punkte stellt der erfolgte Testaufbau eine Grundlage dar.

FAI wird also auch in Zukunft für Installationen und Updates bei GeNUA verwendet werden.

## 6 Literaturverzeichnis

<http://www.informatik.uni-koeln.de/fai/>  
<http://syslinux.zytor.com/faq.php>  
<http://www.cpqlinux.com/dhcpd.html>  
<http://www.linuxfibel.de>  
<http://www.selflinux.org/selflinux>  
<http://www.wikipedia.org>

## 7 Abweichungen vom Konzeptpapier

Die Projektarbeit weicht in folgenden Punkten vom Konzeptpapier ab:

- Die Realisierungsphase nahm 3 h länger als geplant in Anspruch. Die Gesamtdauer von 35 h wurde nicht überschritten.
- Die firmeneigene Dokumentation erfolgt erst am Ende des gesamten Projekts. Daher enthält der Anhang keine Auszüge aus derselben.
- Die Anlagen wurden um ein Meetingprotokoll und einen Ganttplan ergänzt.

## A Anlagen

### A.1 Protokoll Kickoffmeeting

Datum: 22.01.2007, Zeitraum: 13:30 Uhr - 13:45 Uhr, Ort: Seminarraum 2-41

Anwesende: Nikolay Sturm, Katharina Drexel, Protokoll: Katharina Drexel

Tagesordnung:

1. Projektauftrag
2. Projektziel
3. Auftraggeber und Projektbeteiligte
4. Anforderungen an die Software
5. Anforderungen an die Testumgebung
6. Zeitplan
7. weitere Schritte

zu 1.)

Herr Sturm erläuterte, dass die mittelfristige Umstellung aller Arbeitsplätze auf Etch ein guter Anlass wäre, die veraltete FAI-Software durch eine neue Version zu ersetzen. Ein erster Schritt hierfür stellt der Aufbau einer separaten Testumgebung dar.

zu 2.)

Ziel der Projektarbeit ist die Verifikation eines funktionierendes FAI-Servers unter Etch sowie einer automatischen Installation eines Clients. Die Konfiguration auf dem Server soll später dahin ausgebaut werden können, dass eine auf die Erfordernisse eines GeNUA-spezifischen Arbeitsplatzrechners angepasste Maschine installiert werden kann.

zu 3.)

Auftraggeber des Projekts ist die interne Systemverwaltung. Die Bearbeitung erfolgt zunächst durch die Verfasserin. Im Produktivbetrieb (der nicht mehr Bestandteil der Projektarbeit ist) wird die Software von allen Mitarbeitern der internen Systemverwaltung gepflegt.

zu 4.)

Die eingesetzte Software soll eine unterbrechungsfreie Installation eines Etch-Clients gewährleisten. Alle Funktionalitäten der bestehenden FAI-Software, z.B. eine skriptgesteuerte Konfiguration des FAI-Clients oder die Loggingfunktion müssen erhalten bleiben und gegebenenfalls ausgebaut werden. Für den Umfang der Projektarbeit genügt zunächst die Einrichtung eines funktionierenden FAI-Servers sowie die Installation eines Clients mit den FAI-Defaultskripten und die Überprüfung aller genannten Funktionen. Eine genauere Auflistung der zu erfüllenden Anforderungen befindet sich im Punkt 1.4 (Sollzustand).

zu 5.)

Um alle im Zusammenhang mit FAI stehende Dienste wie DHCP oder NFS eigenständig betreiben zu können, soll zunächst ein separates FAI-Netz aufgebaut werden, in das später bei Bedarf weitere Testmaschinen integriert werden können. Für die Bereitstellung von Softwarepaketen soll der GeNUA-eigene Mirror im Testnetz dienen.

Die Testumgebung soll im Labor 2-44-R1 aufgebaut werden. Die Zusammensetzung der Testhardware erfolgt aus dem für Azubis vorgesehenen Hardwarepool.

zu 6.)

Als erster Überprüfungstermin wird der Abschluss der FAI-Serverinstallation gewählt (Meilenstein des Projekts). Eine abschließende Überprüfung findet nach Beendigung der Client-Installation statt.

Das gesamte Projekt soll spätestens zum offiziellen Release von Etch als „Debian Stable“ beendet sein.

zu 7.)

Zuletzt erwähnt Herr Sturm noch kurz die Schritte, die außerhalb der Projektarbeit stattfinden werden. Dazu gehört die Überprüfung der vorhandenen Skripten hinsichtlich Redundanz und Funktionalität auch unter Etch, die Auswahl geeigneter unter Etch zu installierender Softwarepakete, die Installation von Servern und Standalonemaschinen und das Verfassen einer Dokumentation auf dem internen Webserver der Firma.

## A.2 Ganttplan

s. Abb. auf der letzten Seite

## A.3 etc/apt/sources.list für den FAI-Server, die FAI-Konfiguration und das Nfsroot

```
deb http://mirror.genua.de/debian etch main contrib non-free
deb http://mirror.genua.de/debian-security etch/updates main \
contrib non-free
deb http://mirror.genua.de/debian-marillat/ etch main
deb http://mirror.genua.de/debian-genua/etch/ ./
```

## A.4 Startup-Skript für Masquerading mit iptables auf dem Router

```
#!/bin/sh
case $1 in
start)
    iptables -t nat -A POSTROUTING -s 192.168.1.0/255.255.255.0 \
        -o eth0 -j MASQUERADE
    ;;
stop)
    iptables -F POSTROUTING -t nat
    ;;
restart)
    iptables -F POSTROUTING -t nat
    iptables -t nat -A POSTROUTING -s 192.168.1.0/255.255.255.0 \
        -o eth0 -j MASQUERADE
    ;;
*)
    echo "Usage: start/stop/restart"
    exit 1
;;
esac
```

## A.5 fai.conf

```
LOGUSER=fai
LOGSERVER=192.168.1.10
FAI_LOGPROTO=ssh
FAI_CONFIGDIR=/srv/fai/config
FAI_CONFIG_SRC=nfs://192.168.1.10$FAI_CONFIGDIR
FAI=/var/lib/fai/config
```

## A.6 make-fai-nfsroot.conf

```
NFSROOT=/srv/fai/nfsroot
NFSROOT_ETC_HOSTS="192.168.1.10 server"
FAI_DEBOOTSTRAP="etch http://10.0.1.17/debian"
FAI_ROOTPW=' $1$RTiUric3$xEEAj2w7//6I1ZyC.aFU/.'
KERNELPACKAGE=/usr/lib/fai/kernel/linux-image-2.6.20.1_genua.0.3_i386.deb
FAI_DEBOOTSTRAP_OPTS="--exclude=dhcp-client,info --verbose"
```

## A.7 update-exim4.conf.conf

```
# /etc/exim4/update-exim4.conf.conf
dc_eximconfig_configtype='smarthost'
dc_other_hostnames='server'
dc_local_interfaces=''
dc_readhost=''
dc_relay_domains=''
dc_minimaldns='false'
dc_relay_nets=''
dc_smarthost='192.168.217.122'
CFILEMODE='644'
dc_use_split_config='false'
dc_hide_mailname=''
dc_mailname_in_oh='true'
dc_localdelivery='mail_spool'
```

## A.8 Dateien im FAI-Ordner `/srv/fai/config`

### A.8.1 `class/50-host-classes`

```

#!/bin/bash
# assign classes hosts
# use a list of classes for our demo machine
case $HOSTNAME in
    faiserver)
        echo "FAIBASE DEMO FAISERVER" ;;
    demohost)
        echo "FAIBASE DHCP DEMO" ;;
    gnomehost)
        echo "FAIBASE DHCP DEMO XORG GNOME" ;;
    atom*)
        echo "FAIBASE DHCP DEMO" ;;
    *)
        echo "FAIBASE DHCP GENUA ETCHWS" ;;
        #-----#
esac

(ifclass I386 || ifclass AMD64) && echo GRUB
exit 0

```

Der markierte (`#-----#`) Teil wurde hinzugefügt, der Rest ist FAI-Standard.

### A.8.2 `files/etc/locale.gen/ETCHWS`

```

de_DE@euro ISO-8859-15
de_DE ISO-8859-1
de_DE.UTF-8 UTF-8
en_US ISO-8859-1
en_US.UTF-8 UTF-8

```

Die Datei `files/etc/apt/sources.list/ETCHWS` ist identisch mit der entsprechenden Datei im `Nfsroot` (s. Kapitel A.3).

### A.8.3 `scripts/ETCHWS/10-copy-files`

```

#!/bin/sh
# generate locales and sources.list on client
fcopy -B etc/apt/sources.list
fcopy -B etc/locale.gen
$ROOTCMD dpkg-reconfigure -u locales

```

(`ROOTCMD=chroot /tmp/target`, von FAI exportierte Shellvariable)

(`fcopy`: FAI-spezifischer Befehl zum Kopieren von Dateien)

#### A.8.4 hooks/faiend.ETCHWS

```
#!/bin/sh
NAME='hostname'
echo "Installation type: Workstation" | mail -s \
    "Installation of $NAME finished" drexel@genua.de
```

#### A.9 dhcpd.conf

```
ddns-update-style none;
default-lease-time 600;
max-lease-time 7200;
authoritative;
log-facility local7;

option domain-name "azubis.de";
option subnet-mask 255.255.255.0;
option domain-name-servers 10.0.1.4;

host faiclient {
    hardware ethernet 00:90:27:75:48:d7;
    option host-name "faiclient";
    fixed-address 192.168.1.209;
}
```

#### A.10 isolinux.cfg

```
DEFAULT INSTALL
PROMPT 1
TIMEOUT 150
DISPLAY boot.msg
LABEL INSTALL
    KERNEL vmlinuz
    APPEND ip=dhcp nfsroot=192.168.1.10:/srv/fai/nfsroot
FAI\_ACTION=install FAI\_FLAGS="createvt,sshd,verbose"
```

#### A.11 boot.msg

```
*****
*** Fully Automatical Installation of a Debian system ***
*** *** All data may be overwritten ! *** *** ***
*****
To continue press ENTER, to reboot press Ctrl+Alt+Return,
timeout is 15 seconds
```



## A.12 Übergabe- und Abnahmeprotokoll

Übergabeprotokoll vom .....

Die im Projekt-Antrag beschriebene Funktionalität wurde getestet. Folgende Anforderungen wurden geprüft:

|   | erfolgreich | Projektdurchführender | Kontrolle |
|---|-------------|-----------------------|-----------|
| FAI und zugehörige Dienste wurden ordnungsgemäß konfiguriert  | ja o nein o | .....                 | .....     |
| Bootmedium ermöglicht problemfreies Starten des FAI-Clients   | ja o nein o | .....                 | .....     |
| Hinzufügen eigener Installationsklassen funktioniert  | ja o nein o | .....                 | .....     |
| Installationsschritte (Partitionieren und Formatieren, Grundsystem und Software installieren, Skripten ausführen, Dateien übertragen) werden ordnungsgemäß ausgeführt | ja o nein o | .....                 | .....     |
| erfolgreicher Reboot des Clients nach erfolgter Installation  | ja o nein o | .....                 | .....     |
| korrektes Funktionieren des installierten Systems (stichprobenartige Überprüfung)   | ja o nein o | .....                 | .....     |
| Logfiles werden auf dem FAI-Server gespeichert  | ja o nein o | .....                 | .....     |

Das Projekt wurde erfolgreich abgenommen

.....  
Projektverantwortlicher



