

Titel: Asterisk. The Definitive Guide
Autor: Leif Madsen, Jim Van Meggelen & Russell Bryant
Jahr: April 2011: Third Edition
Seitenzahl: 695
Preis: 45 EUR
ISBN: 978-0-596-51734-2

1 Allgemeine Bemerkungen

Das Buch „Asterisk. The Definitive Guide“ richtet sich in erster Linie an Leser, die eigentlich bereits Erfahrung mit Asterisk haben und Kenntnisse erweitern möchten und es als Referenzbuch verwenden möchten. Allerdings ist das Buch mit Einschränkungen auch für Anfänger geeignet, da die Autoren großen Wert darauf legen, von der Installation und Konfiguration bis hin zur der Einrichtung einer kleinen VOiP-Anlage alles umfassend zu erklären. Dabei werden Basis-Kenntnisse in Linux-Administration, Netzwerk-Management sowie weiteren IT-Bereichen vorausgesetzt (von Vorteil wären z.B. Basis-Kenntnisse in einer Skript-Sprache).

2 Gliederung des Buches

Kapitel 1: A Telephony Revolution

Im ersten Kapitel gehen die Autoren auf die Entstehungsgeschichte von Asterisk ein: im Jahr 1999 machte sich Jim Dixon, zunächst in Kombination mit dem „Zapata Telephony Project“, Gedanken über einen Chip, der direkt in den Computer eingebaut werden sollte, woraus ein selbständiges Asterisk PSTN-System resultierte. Aus diesem Projekt ging später das zur Zeit wohl verbreitetste Telefon-Interface – „Digium Asterisk Hardware Device Interface“ (DAHDI) – hervor.

Die Autoren verweisen auf Asterisk-betriebene Telefonanlagen, die sowohl von Hobby-Anwendern als auch kommerziell betrieben werden können. Dabei geben sie zur ersten Orientierung web-Adressen.

Kapitel 2: Asterisk Architecture

Den Hauptunterschied zu einer herkömmlichen Telefonanlage sehen die Autoren bei einer auf Asterisk basierten Telefonanlage darin, dass alle Anrufe hauptsächlich über den sog. Wählplan (engl. dial plan) gemanaged werden, da es technisch gesehen keine Unterschiede bei der Behandlung von eingehenden Leitungen bzw. Anrufen untereinander gibt.

Die Autoren stellen grafisch die beiden Architekturen (herkömmliche und auf Asterisk basierte Telefonanlage) gegenüber.

Modules

Die Autoren stellen die wichtigsten Module von Asterisk vor. Mithilfe von Modulen werden bestimmte Funktionalitäten beim Aufruf geladen und zu Verfügung gestellt.

Module werden in der Datei */etc/asterisk/modules.conf* konfiguriert und haben immer die Endung *.so*. Es gibt folgende Arten von Modulen:

- Applications (gemeint sind Wählplan-Anwendungen)
- Bridging modules
- Call detail recording (CDR) modules
- Channel event logging (CEL) modules
- Channel drivers
- Codec translators
- Format interpreters
- Dialplan functions
- PBX modules
- Resource modules
- Addons modules
- Test modules

Die Autoren gehen in diesem Kapitel einzeln auf die Module ein und geben in einer Liste kurz deren Aufgabe und den Status der Anwendungshäufigkeit an, d.h. ob das jeweilige Modul aktuell und notwendig ist oder bereits veraltet ist und daher davon abgeraten wird, es noch zu verwenden.

File Structure

In diesem Unterkapitel erfahren wir, wo Asterisk die wichtigsten Dateien ablegt:

| | | | |
|--|--------------------------|------------------|----------------------------------|
| Configurations Files Extensions.conf sip.conf modules.conf | <i>/etc/asterisk</i> | Modules | <i>/usr/lib/asterisk/modules</i> |
| The Resource Library MOH (Music On Hold) | <i>/var/lib/asterisk</i> | The Spool | <i>/var/spool/asterisk</i> |
| Logging CDR (Call Detail Records) | <i>/var/log/asterisk</i> | | |

Der Wählplan stellt das Herzstück von Asterisk dar und kann auf drei Weisen geschrieben werden:

Wählplan-Syntax stellt eine traditionelle und beliebte Methode dar, den Anrufswählplan zu konfigurieren. Zu finden ist er unter */etc/asterisk/extensions.conf*

Asterisk Extension Logic (AEL) wird später im Buch beschrieben, zu finden ist er unter `/etc/asterisk/extensions.ael`

LUA wird später im Buch beschrieben, zu finden ist er unter `/etc/asterisk/extensions.lua`

Weiter im Kapitel listen die Autoren Anbieter von Hardware auf, die notwendig ist, um Asterisk über eine herkömmliche Telefonanlage anzubinden. Auswahl:

- Digium (Sponsor, Besitzer und Hauptentwickler von Asterisk)
- Sangoma
- Rhino
- OpenVox
- Pika
- Voicetronix
- Junghanns
- Dialogic
- Xorcom
- beroNet

Schließlich erfahren wir, nach welcher Methode Asterisk versioniert wird.

Kapitel 3: Installing Asterisk

Im dritten Kapitel wird die Installation und Basiskonfiguration von Asterisk auf den Linux-Distributionen CentOS und Ubuntu vorgeführt.

Kapitel 4: Initial Configuration Tasks

Im vierten Kapitel werden die wichtigsten Anfangskonfigurationsdateien von Asterisk vorgestellt:

asterisk.conf In dieser Konfigurationsdatei werden generelle Einstellungen vorgenommen, die die Arbeitsweise von Asterisk beeinflussen.

modules.conf In dieser Konfigurationsdatei werden die eingebundenen Module eingestellt. Ohne diese Konfigurationsdatei kann Asterisk nicht richtig arbeiten.

indications.conf In dieser Konfigurationsdatei werden Parameter für verschiedene Klingeltöne gesetzt, z.B. unterschiedliche Ton-Zuordnungen für bestimmte Länder.

musiconhold.conf In dieser Konfigurationsdatei werden die Einstellungen für die Warteschleifenmusik vorgenommen.

Kapitel 5: User Device Configuration

Im fünften Kapitel stellen die Autoren Techniken vor, wie man Geräte und Asterisk derart konfiguriert, dass sie einander gegenseitig erkennen und miteinander kommunizieren können.

Es wird kurz das SIP-Protokoll (Session Initiation Protocol) vorgestellt und erklärt, wie es mit Asterisk zusammenarbeitet:

- Asterisk fungiert als Brücke zwischen eingehenden Anrufen und dem Empfängergerät.
- die SIP-Transaktionen erfolgen bis zu Asterisk und dann von Asterisk zum Empfängergerät.

Telephone Naming Concepts

In Asterisk hat sich eingebürgert, dass die Leitungen Namen erhalten (channel names). Es gibt kein Konzept für die Rolle eines Benutzers (außer bei Asterisk GUI). Die Durchwahl wird verwendet, um eingehende Anrufe durch das System zu leiten.

Die Autoren empfehlen, als Namen für ein Telefongerät seine MAC-Adresse zu verwenden.

Hardphones, Softphones, and ATAs

Es werden drei typische Endgeräte vorgestellt, die im Zusammenhang mit Asterisk und VOiP-Telephonie eingesetzt werden:

Hardphones stellen physische Geräte dar, die alle Eigenschaften eines Gerätes einer herkömmlichen Telefonanlage haben, nur mit dem Unterschied, dass sie über Netz via SIP-Protokoll verbunden werden. Diese Telefongeräte werden SIP-Telefone bzw. VOiP-Telefone genannt.

Vorteile: gute akustische Qualität.

Nachteile: nicht-portabel, relativ teuer.

Softphones stellen eine Software-Anwendung dar, mit deren Hilfe eine Telefonverbindung realisiert wird, die auf dem PC bzw. Laptop läuft, wenn dieser entsprechend mit Soundsystem ausgestattet ist. Zur Ausstattung hierfür gehört normalerweise auch noch das Headset. Aktuell gibt es mehr und mehr solcher Softphone-Anwendungen für Smartphones.

Vorteile: portabel, leicht zu installieren, mit anderen Funktionen kombiniert einsetzbar (z.B. Webcam, Videoanrufe).

Nachteile: es besteht die potentielle Gefahr, dass die Softphones nicht immer angeschaltet sind. Man braucht beim Telefonieren ständig ein Headset. Und nicht zuletzt kann die Software des Softphones im schlimmsten Fall die Ausführung anderer Programme behindern.

Analog Terminal Adaptors (ATAs) stellt einen Adaptor in Form einer Box dar, die aus zwei Konnektoren besteht: einem RJ-11-Konnektor zum analogen Telefonapparat einerseits und einem RJ-45-Konnektor zum Netz andererseits.

Vorteile: die Vorteile ergeben sich aus der primären Funktion eines Adapters: die analogen Telefonapparate mit dem Netz via SIP-Protokoll zu verbinden. Aber auch mit Hilfe von Karten bestimmter Hersteller (z.B. Digium), die direkt in den Server eingesetzt werden, kann man eine SIP-Telefonverbindung herstellen.

Nachteile: Funktionalitäten sind gegenüber einem SIP-Telefon sehr begrenzt.

Nach den Abwägungen von Vor- und Nachteilen kamen die Autoren zu dem Schluss, dass sie Softphones bevorzugen.

Zum Schluss wird erläutert, wie man Asterisk mittels der Dateien *sip.conf* und *extensions.conf* konfiguriert und diese Dateien anschließend mit der Asterisk-Kommandozeileninterpreter lädt und testet. Außerdem wird demonstriert, wie man Geräte registrieren kann. Nicht zuletzt stellen die Autoren den ersten einfachen Wählplan vor, den man gleich ausprobieren kann.

Kapitel 6: Dialplan Basics

Im sechsten Kapitel werden Grundlagen des Wählplans vorgestellt. Für die Autoren stellt ein Wählplan das Herzstück eines Asterisk-Systems dar, denn dieser regelt die Art und Weise, wie die ankommenden und ausgehenden Anrufe behandelt werden.

Insbesondere wird die Syntax eines Wählplans erläutert, der in der *extensions.conf* in einer Art Skriptsprache definiert ist. Dabei werden folgende Sektionen von *extensions.conf* genauer unter die Lupe genommen:

Contexts listet die Abschnitte eines Wählplans auf, die Instruktionen für die Ausführung einer bestimmten Aktion in Bezug auf Anrufe beinhalten. Ein Wählplan ist in mehrere, von einander unabhängig agierende *contexts* unterteilt, die durch das Token **[context name]** dargestellt werden. Es gilt folgende Regel: ein *context* gilt solange bis der nächste *context* ihn ablöst.

In der Datei *sip.conf* muss eine Referenz auf die jeweiligen *contexts* in der Form *context=context name* existieren, damit der Wählplan von dieser Leitung bzw. von diesem Kanal aus starten kann.

Extentions listet Anwendungen auf, die durch eine Anzahl von Anweisungen hinsichtlich der Handhabung von Anrufen charakterisiert sind. Ein *context* kann aus mehreren solcher *extensions* bestehen.

Man erkennt eine *extension* an folgender Auszeichnung: **exten** =>, gefolgt von drei Komponenten:

- Name bzw. Nummer der *extension*
- Priorität der *extension*
- Anwendung bzw. Befehl, der beim Aufruf des jeweiligen Kontextes ausgeführt werden soll

(vgl. z.B. **exten** => **101,1,Dial (Sip/0000FFFF0002)**).

Priorities Mittels Nummerierung wird die Reihenfolge mehrerer *extensions* festgelegt. Wenn es sich allerdings um ein und die selbe Nummer handelt, auf die eine Anwendung angewandt wird, wird, um Redundanz im Codes zu vermeiden, nur bei der Initialisierung eine Priorität gesetzt und weitere Schritte durch **n**-Priorität markiert. (**n** steht hierbei für **next**), und *exten* => wird durch den **same**-Operator ersetzt:

```
exten => 123,1,Answer()
same => n,do something
same => n,do something else
same => n,do one last thing
same => n,Hangup()
```

Man kann zwecks besserer und lesbarer Referenz zusätzlich eine Priorität mit einem Label versehen:

```
exten => 123,n(label),application()
```

Im weiteren Verlauf des Kapitels erläutern die Autoren grundlegende Applikationen von Asterisk, die wichtige Standardfunktionen rund ums Telefonieren implementieren:

- **Answer()** antwortet auf einen eingehenden Anruf; hat keine Argumente
- **Playback(Filename oder relativer bzw. absoluter Pfad)** spielt einen zuvor aufgenommenen Ton ab
- **Hangup()** beendet die aktuelle Leitung; hat keine Argumente

Anschließend wird ein einfacher statischer Wählplan vorgestellt, mit dessen Hilfe eine Stimme, nachdem ein Anruf angekommen ist, zunächst „Hello World“ sagt und dann auflegt:

```
exten => 200,1,Answer()
same => n,Playback(hello-world)
same => n,Hangup()
```

Es werden außerdem weitere Funktionen vorgestellt, die verwendet werden, um einen dynamischen Wählplan zu erstellen:

- **Goto(context,extension,priority)** dient der Weiterleitung des Anrufes auf einen anderen Teil des Wählplans, in dem als Ziel der Name des betreffenden *contextes* aufgeführt ist (z.B. *exten => 201,1,Goto(TestMenu,start,1)*).
- **Background()** spielt einen zuvor gespeicherten Ton ab; wird meistens verwendet, um Stimmen-Menüs zu erstellen, damit der Anrufer mit der Kombination von Wähltasten weiter verbunden bzw. umgeleitet werden kann. Die Syntax ist mit der von *Playback()* identisch.
- **WaitExten()** ermöglicht es, die Eingabe des Anrufers solange zu ignorieren, bis der Ton zu Ende gespielt ist.

Als nächstes werden die möglichen Argumente der **Dial()**-Funktion erläutert:

1. **Destination** besteht aus *technology/user*, vgl.:
exten => 201,1,Dial(DAHDI/1)
2. **Timeout** definiert, wie lange maximal auf die Abnahme des Hörers gewartet wird, vgl.:
exten => 201,1,Dial(DAHDI/1,10)
3. **Option** in Form eines oder mehrerer Buchstaben, die Dial()-Anwendung modifizieren. Der meistverwendete Modifikator ist **m** und steht für Musik, die, anstatt des Freizeichens gespielt wird, während man anruft, vgl.:
exten => 201,1,Dial(DAHDI/1,10,m)
4. **URI** stellt das vierte und letzte Argument der Dial()-Anwendung dar und wird angegeben, wenn man die URI des Anrufers angezeigt haben möchte.

Das zweite, dritte und das vierte Argument kann weggelassen werden. Zwingend erforderlich ist jedoch die Angabe des ersten Arguments. Bei fehlenden Argumenten muss ausserdem das Komma als Platzhalter beibehalten werden:

z.B. die Angabe des zweiten Arguments (*Timeout*) wird weggelassen:
exten => 201,1,Dial(DAHDI/1,,m)

Desweiteren erfahren wir, dass es folgende Variablentypen gibt:

- **Global variables**, die für die ganze Anwendung im Wählplan gelten. Sie werden im Kontext *[globals]* deklariert.
- **Channel Variables**, die nur in Verbindung mit dem jeweiligem Anruf gültig sind. Sie werden mit Hilfe der *Set()*-Funktion deklariert.
- **Environment variables**, die den Zugang zur Unix-Umgebung ermöglichen. Die Syntax lautet: *ENV(var)*

Anschließend erfahren wir, wie man in Asterisk pattern matching auszeichnet.

Zum Schluss erfahren wir, wie man externe Kontexte mit Hilfe des **include**-Befehls anbindet, vgl.:

include => context.

Kapitel 7: Outside Connectivity

Im siebten Kapitel werden die wichtigsten Techniken und Konfigurationen der Hauptleitungen vorgestellt, welche für die Verbindung eines Asterisk-Systems mit der Außenwelt relevant sind. Zuvor wird die Architektur eines Netzwerk-basierten Asterisk-Systems erläutert und dessen Unterschiede dem herkömmlichen analogen Telefonsystem kurz gegenübergestellt.

Schließlich wird vorgestellt, wie man Notrufnummern am geschicktesten auszeichnet, so dass sie auch von unterwegs und vor allem vom Ausland aus getätigt werden können.

Kapitel 8: Voicemail

Im achten Kapitel geht es um die Einrichtung bzw. Konfiguration des Voicemail- bzw. Sprachnachrichten-Systems unter Asterisk. Einige nützliche Funktionen werden aufgelistet. Eine Auswahl:

- Unlimited password-protected voicemail boxes, each containing mailbox folders for organizing voicemail
- Different greetings for busy and unavailable states
- Email notification of voicemail, with the voicemail optionally attached as a sound file
- Company directory of employees, based on voicemail boxes

Die Konfiguration erfolgt in der Datei `/etc/asterisk/voicemail.conf`, deren Optionen die Autoren den Hauptteil der Erläuterungen in Form von Tabellen widmen:

- `[general]`- hier werden allgemeine Einstellungen vorgenommen
- `[zonemessages]`- hier werden Zeitstempel in Bezug auf Zeitzonen konfiguriert
- `[default]` stellt einen sog. *context* dar. Wenn keine Trennung von den Voicemail-Usern besteht, dann wird empfohlen, dass der Eigentümer **default** als ausschließlicher **voicemail context** verwendet wird. Das Format für eine Mailbox ist das folgende:

```
mailbox => password[,FirstName LastName[,email addr[,pager addr[,options[|options]]]]]
```

Anschließend wird eine Beispiel-Konfiguration vorgestellt, mit einer Auswahl von durch die Autoren empfohlenen Optionen.

Zum Schluss wird demonstriert, wie man ein Asterisk-System so konfiguriert, dass es als „Standalone Voicemail Server“ in Betrieb genommen werden kann.

Kapitel 9: Internationalization

Im neunten Kapitel erfahren wir, wie man eine asteriskbasierte Telefonanlage konfigurieren und technisch so ausrüsten kann, damit das Telefonieren in das und aus dem Ausland gewährleistet ist. Behandelt wird unter anderem folgende Funktion:

- die Möglichkeit der Konfiguration der Länder-spezifischen Klingeltöne, die auf deren Vorwahl basieren und beim getätigten Anruf zum Einsatz kommen.

Kapitel 10: Deeper into the Dialplan

Im zehnten Kapitel werden weitere Programmierbestandteile eines Wälplans erläutert:

- Ausdrücke mit der Auszeichnung `$(expression)`
- Operatoren, die man auch aus anderen Programmiersprachen kennt.
- Funktionen: `$(FUNCTION_NAME(argument))`
- Konditional-Ausdrücke: `GotoIf(expression?destination1:destination2)`
- zeitbasierte Konditionalausdrücke:
`GotoIfTime(times,days_of_week,days_of_month,months?label)`

Im weiteren Verlauf des Kapitels erläutern die Autoren, wie man Makros erstellt, mit deren Hilfe man Code-Redundanz im Wälplan vermeiden kann.

Desweiteren stellen die Autoren die Funktion **GoSub()** vor, mit deren Hilfe man Subroutinen schreiben kann, die es erlauben, nur einen Teil des Wälplans abzurufen, diesen mit einer bestimmten Information zu versorgen und, wenn nötig, anschließend einen Wert zurückzuliefern.

Anschließend wird gezeigt, wie lokale Leitungen aus dem Wälplan heraus definiert werden.

Die asteriskeigene Datenbank **AstDB** wird vorgestellt und deren Einbindung in den Wälplan wird vorgeführt.

Schließlich werden Anwendungen, die für Handy-Telefonie wichtig sind vorgestellt:

- **Zapateller()**- vermittelt unerwünschten Anrufern (z.B. Werbung) den Eindruck, die Verbindung sei unterbrochen. Nachdem die unerwünschten Teilnehmer den Hörer auflegen, markiert deren System die aktuell unerreichte Telefonnummer als „außer Betrieb“. Zumindest soll diese Maßnahme gegen einige Arten von Werbetelefonanrufen helfen, wenn sie auf einer automatischen Wählvorrichtung basieren.
- **Call parking** erlaubt das Parken von Gesprächen in einer Parkposition (parking lot), sodass sie von anderen *extensions* erreichbar sind. In der Datei `features.conf` kann man die Konfigurationen für das *call parking* vornehmen. In der Section **[general]** können folgenden Parameter konfiguriert werden:
 - parktext
 - parkpos
 - context
 - parkingtime

- **MeetMe()** stellt eine Audiokonferenz-Brücke dar, die es mehreren Telefonteilnehmern erlaubt, miteinander zu kommunizieren, als ob sie sich physisch in einem Raum befänden. Die wichtigsten Vorteile dieser Anwendung sehen die Autoren in folgenden Eigenschaften:
 - Möglichkeit passwortgeschützter Audiokonferenzen
 - Administration: z.B. stumm-geschaltete oder geschlossene Konferenzen uvm.
 - Stummschaltung für alle außer einem Teilnehmer
 - statische oder dynamische Erstellung von Audiokonferenzen

Kapitel 11: Parking and Paging

Im elften Kapitel wird erläutert, wie Gespräche geparkt (parking) und dann abgerufen werden (paging), auf Wunsch, mit Bekanntgabe wem der Anruf gilt und wie er angezeigt werden soll.

Die Definitionen zu zu parkenden Gesprächen werden in der Konfigurationsdatei *features.conf* vorgenommen:

- [general] - allgemeine Einstellungen (*parktext*, *parkpos*, *context* uvm.)
- [featuremap] - erlaubt es, bestimmte Tastenkombination zu definieren, die verschiedene Ereignisse auf der Leitungen auslösen können und die über die Funktionen *Dial()* bzw. *Queue()* überbrückt wurden.
- [applicationmap] - erlaubt es, den DTMF-Code auf den Wählplan so zu mappen, dass der Anruf solange in die Warteschleife umgeleitet wird, bis die Anwendung abgeschlossen ist.

Schließlich gehen die Autoren ausführlicher auf den Begriff Paging ein (siehe: **Overhead and „Underchin“ Paging (a.k.a. Public Address)** usw. ab S. 231-236 ff.).

Kapitel 12: Internet Call Routing

Im zwölften Kapitel erläutern die Autoren die wichtigsten Protokolle und Standards, die im Zusammenhang mit der Übertragung von Daten via Internet bereits aus anderen Bereichen des Web bekannt sind und wie sie in Asterisk verwendet werden können. Eine Auswahl der Themen:

- *DNS* und *SIP URI*
- *ENUM*, *E.164*
- *ISN*, *ITAD*

Außerdem wird erklärt, wie man auf Sicherheit achtet.

Kapitel 13: Automatic Call Distribution (ACD) Queues

Im dreizehnten Kapitel werden Techniken vorgestellt, mit denen man die automatischen Rufumleitungs-Warteschleifen (*Automatic Call Distribution (ACD) Queues*) konfigurieren kann:

- *queues.conf*
- *agents.conf*

Im weiteren Verlauf des Kapitels erläutern die Autoren, wie man den Status der Anrufe in Bezug auf eine Priorisierung der Annahme festlegen kann.

Schließlich erfahren wir, wie man Statistiken erstellen kann. Für die Definition stehen folgende Optionen in der Konfigurationsdatei *logger.conf* (zu finden unter */var/log/asterisk*) zur Verfügung:

- *queue_log* - Aktivieren/Deaktivieren des Logging
- *queue_log_to_file* - soll in eine Datei geschrieben werden (Ja/Nein)
- *queue_log_name* den Namen für das *queue_log* vergeben

Kapitel 14: Device States

Im vierzehnten Kapitel wird erläutert, welche Konzepte der Geräte-Status-Überwachung es in Asterisk gibt, und wie die Geräte und Anwendungen an die Status-Informationen kommen und diese verwenden können.

Die Autoren unterscheiden zwischen zwei Arten von Geräten, auf den Gerätestatus referieren können:

- **reale Geräte:** z.B. SIP-Geräte
- **virtuelle Geräte:** alles, was sich innerhalb von Asterisk befindet, aber nützliche Statusinformationen liefert

Zum Funktionsarsenal der Gerätestatus-Überwachung von Asterisk gehören folgende:

- ***DEVICE_STATE()*** kann direkt im Wählplan ausgezeichnet werden, um den Gerätestatus zu testen, z.B.:
exten => 7012,1,Answer()
same => n,Verbose(3,The state of SIP/0004F2060EB4 is \
\${DEVICE_STATE(SIP/0004F2060EB4)})
same => n,Hangup()

Folgende Statusabfrage-Resultate können zurückgeliefert werden(eine Auswahl):

- UNKNOWN
- INUSE
- BUSY u.a.

- **Extension States** stellt ein weiteres wichtiges Konzept in Asterisk dar. Dieser Status wird durch sog. *hints* (Hinweise bzw. Spuren) ermittelt und mit deren Hilfe kann einer bestimmten Extension das jeweilige Gerät bzw. der jeweilige Benutzer zugeordnet werden, um später den Status des Gerätes zu überwachen. Die Zuordnung zwischen Extension und Geräten wird im Dialplan vorgenommen, wobei das Schlüsselwort *hint* an Stelle der Priorität kommt und diese somit ersetzt, vgl:
 exten => 1234, hint,SIP/phoneA/&SIP/phoneB/&SIP/phoneC
 exten => 5555, DAHDI/1

Anschließend müssen noch die Konfigurationen in *sip.conf* vorgenommen werden (eine Auswahl):

- callcounter
- busylevel
- call-limit
- allowsubscribe

Schließlich erfahren wir, wie man das eXtensible Messaging and Presence Protocol (XMPP) (bekannt durch *Jabber*) unter Asterisk anwenden kann. Es wird erläutert, wie man XMPP installiert, Asterisk entsprechend konfiguriert und testet.

Kapitel 15: The Automated Attendant

Im fünfzehnten Kapitel werden die Einstellungen des Automated Attendant vorgestellt. Die Autoren betonen dabei, dass es sich hierbei nicht um Interactive Voice Response (IVR), also ein automatisiertes Dialogsystem handelt, sondern bloß um ein System, mit dessen Hilfe man zu einem bestimmten Menüpunkt weitergeleitet werden kann, ohne dass eine Aktion des Anrufers erforderlich ist.

Die Autoren empfehlen, den „*automated attendant*“ so einfach wie möglich einzurichten, so dass der Anrufer so schnell wie möglich mit einer lebenden Person verbunden wird und nicht minutenlang die Maschinenstimme anhören muss.

Kapitel 16: Relational Database Integration

Im sechzehnten Kapitel wird erläutert, wie man die PostgreSQL- und MySQL-Datenbanken in Asterisk integriert, konfiguriert und im Wählplan auszeichnet (mit *fun_odbc*), so dass die Datenbanken verwendet werden können, z.B. um Kontaktdaten zu speichern.

Kapitel 17: Interactive Voice Response

Im siebzehnten Kapitel gehen die Autoren ausführlicher auf das Thema der automatisierten Dialogsysteme ein, zu denen das *Interactive Voice Response* (IVR) gehört. Die Autoren erläutern, aus welchen Komponenten das IVR besteht, nämlich aus einem Prompt

und aus einer zweiten Komponente in Form einer Methode, welche das von dem Anrufer eingegebene Material liest, das Ergebnis bearbeitet und in irgendeiner Form zurück liefert. Zu IVR gehört ebenfalls die Bearbeitung von Sprachsignalen, d.h. der Anrufer kann sprechen, anstatt auf die Tasten zu drücken, Asterisk bearbeitet die Sprachsignale und leitet entsprechend zum jeweiligen Menüpunkt bzw. interagiert mit dem Anrufer.

Kapitel 18: External Services

Im achtzehnten Kapitel werden Techniken vorgestellt, mit denen man externe Dienste bzw. nützliche Funktionen konfiguriert, die in Asterisk eingebunden werden können:

- Kalender-Funktion
- *Internet Message Access Protocoll* (IMAP) für VoiceMail
- *XMPP* (Jabber) für Nachrichten-Sendungen
- *Skype*
- *Lightweight Directory Accesss Protocoll* (LDAP)

Kapitel 19: Fax

Im neunzehnten Kapitel werden Techniken vorgestellt, wie man Asterisk so einrichtet, dass man Fax-Dokumente empfangen und schicken kann. Dafür benötigt man die *spandsp*-Bibliothek, die separat installiert werden muss.

Kapitel 20: Asterisk Manager Interface (AMI)

Im zwanzigsten Kapitel wird das *Asterisk Manager Interface* (AMI) vorgestellt - die Schnittstelle zum Überwachungs- und Managementsystem, das von Asterisk eigens zur Verfügung gestellt wird. Mit Hilfe von AMI ist es möglich, von Asterisk direkt Aktionen zum Ausführen anzufordern.

Die Autoren erläutern, wie man das AMI aktivieren kann, und wie man es konfiguriert. Die entsprechende Datei findet man unter `/etc/asterisk/manager.conf`, wo man Einstellungen vornehmen kann.

Die Autoren erwähnen Entwicklertools, mit deren Hilfe man einen Code in einer anderen Programmiersprache direkt in AMI verknüpfen kann:

- Adhearsion - *Ruby*
- StarPy - *Python*
- Asterisk-Java - *Java*

Schließlich werden folgende Anwendungen erwähnt, die in Asterisk integriert werden können, bzw. die Verwaltung von Asterisk erleichtern können:

- *Computer-Supported Telecommunications Applications* (CSTA) (siehe <http://opencsta.org>)

- AsteriskGui (siehe <http://www.asterisk.org/asterisknow/>)
- Flash Operator Panel (siehe <http://www.asternic.org>)

Kapitel 21: Asterisk Gateway Interface (AGI)

Im einundzwanzigsten Kapitel werden Techniken vorgestellt, wie man mit einer anderen Programmiersprache geschriebene Codes in Asterisk integrieren kann. Die entsprechenden Konfigurationen werden zunächst in der Datei `/etc/asterisk/extensions/conf` vorgenommen.

Kapitel 22: Clustering

Im zweiundzwanzigsten Kapitel wird das Clustering-System von Asterisk am Beispiel eines Call-Center-Modells vorgestellt.

Kapitel 23: Distributed Universal Number Discovery (DUNDi)

Im dreiundzwanzigsten Kapitel wird das Protokoll *Distributed Universal Number Discovery* (DUNDi) vorgestellt und erläutert, wie es funktioniert, bzw. wie und wo es konfiguriert wird, nämlich in der Datei `dundi.conf`. Was DUNDi ist, erklären die Autoren folgendermaßen (S. 503):

Think of DUNDi as a large phone book that allows you to ask peers if they know of an alternative VoIP route to an extension number or PSTN telephone number.

Kapitel 24: System Monitoring and Logging

Im vierundzwanzigsten Kapitel wird erläutert, wie man Systemüberwachung und Protokolle in Asterisk einrichten kann.

Es werden folgende Dateien bzw. Protokolle vorgestellt, die für die Systemüberwachung und Protokollierung wichtig sind:

- Einstellungen in der Datei `/etc/asterisk/logger.conf`
- *Call Detail Records* (CDR) in der Datei `cdr.conf`
- *Channel Event Logging* (CEL)
- *Simple Network Management Protokoll* (SNMP) - es wird erläutert, wie man es installiert und konfiguriert.

Kapitel 25: Web Interfaces

Im fünfundzwanzigsten Kapitel werden die beiden meist verwendeten Webanwendungen vorgestellt, welche web-basierte Schnittstellen zum Asterisk-System ermöglichen:

- Flash Operator Panel
- A2Billing

Kapitel 26: Security

Im sechszwanzigsten Kapitel wird die Anwendung *Fail2ban* (<http://www.fail2ban.org>) vorgestellt und erläutert wie es installiert und konfiguriert wird.

Kapitel 27: Asterisk: A Future for Telephony

Im siebenundzwanzigsten Kapitel wird über die Zukunft der Telefonie, gesprochen, insbesondere die der traditionellen. Die Unterschiede zwischen der traditionellen und der Telefonie, die auf Asterisk basiert, ist kultureller Natur (S. 578):

The cultural differences are obvious. Telecommunications technologies tend to be closed, confusing and expensive, while networking technologies are comparatively open, well documented and competitive.

Zum Schluss sprechen die Autoren über die Zukunft von Asterisk, die einen positiven Ausgang nimmt, wenn man den Autoren zu glauben vermag, denn Asterisk gehört zu den „Open Source“-Produkten und demnach gilt (S. 594):

Open source telephony creates limitless opportunities.

Appendix A: Understanding Telephony

Im Anhang A werden die Grundlagen von Telefonsystemen erläutert: von Analog-Telefonen bis Digitaltelefonen. Es werden die Signalübertragungen bei verschiedenen Telefonsystemen vorgestellt. Außerdem erfahren wir, wie die Schaltkreise aufgebaut sind und welche Protokolle verwendet werden (z.B. *ISDN* - digitales Protokoll)

Appendix B: Protocols for VoIP

Im Anhang B werden die Protokolle ausführlicher vorgestellt, die in Zusammenhang mit VoIP verwendet werden können:

- IAX
- SIP
- H.323
- MGCP
- Skinny/SCCP
- UNISTIM

Codecs:

- G.711
- G.726
- G.729A

- GSM
- iLBC
- Speex
- MP3

Als weitere Protokolle werden erwähnt:

- TCP
- UDP
- SCTP

Schließlich wird noch einmal gesondert der Augenmerk auf das Thema Sicherheit gelegt.

Appendix C: Preparing a System for Asterisk

Im Anhang C erfahren wir, welche technischen Mindestvoraussetzungen unser System haben sollte, um Asterisk darauf betreiben zu können.

3 Kritik

3.1 Tippfehler

Seite 109 Figure 6-1, extensions.conf `exten => 101,1,Dial (Sip/0000FFFF0002]`
 muss geändert werden in:
`exten => 101,1,Dial (Sip/0000FFFF0002)`

Seite 228 Parking Lots, dritter Absatz, Kommentar zum Code `parkpos => 701-720;`
`What extensions to park calls on (dafault parking lot)`
 muss geändert werden in:
`default`

4 Fazit:

Mit Asterisk ist nicht nur Telekommunikation möglich, sondern auch andere Technologien können damit kombiniert eingesetzt werden. Man kann Asterisk für kleine Heimanlagen aber auch für große Unternehmen, d.h. verbunden mit einer umfangreichen Konfiguration, einsetzen.

Das Buch stellt einen guten Einstieg in Asterisk als Kommandozeilenplattform dar und vermittelt solide Grundkenntnisse, ohne dass eine störende GUI dazwischenfunkt. Somit ist man gut gewappnet, später bei der Konfiguration mittels GUI-basierter Programme wie TrixBox, mit grundlegendem Sachverstand viel schneller fertig zu werden, als wenn man versucht, sich durch die Menüs zu hangeln. Das gilt nicht nur für die Konfiguration,

sondern erst recht bei der Fehlersuche.

Jeder, der mit dem Gedanken spielt, Asterisk einzusetzen, sollte dieses Buch gelesen oder zumindest überflogen haben, um einen Überblick zu gewinnen, was in Asterisk möglich ist, und wo man die Konfiguration findet. Das Buch ist als Community-Basiswerk geschrieben, quasi von Usern für User, und darf auf keinem Schreibtisch fehlen, wenn es um Telefonie mit Asterisk geht.