DHBW Stuttgart

Rechnerkommunikation und Vernetzung
Teil 3: Voice over IP

Dr. Leonhard Stiegler

Nachrichtentechnik

www.dhbw-stuttgart.de

Kursgliederung

DHBW Stuttgart

- Raspberry PI
- Netzwerkdiagnose
  - Kommandos
  - Analyse-Software Wireshark
  - Arbeiten mit Wireshark
- Asterisk – VoIP Einführung
- Asterisk Software
- Asterisk Programmierung

Laboraufbau : Raspberry PI

DHBW Stuttgart

Raspberry PI

- Einplatinen-Rechner mit Kommunikations- und Funktions-Schnittstellen
- ARM Prozessor
- OS: Debian Linux Derivat auf 8GB Typ10 SD-Speicherkarte
- Kommunikationsschnittstellen
  - RJ45 Ethernet, USB, HDMI, Video-Out
  - WLAN via USB-Stecker
- Funktionsschnittstellen
  - General-Purpose I/O (GPIO) mit I^2C, SPI, ...
- Anwendung: Netzwerkdiagnose (Wireshark)
- Anwendung: VoIP Telefonserver (Asterisk)

Kursgliederung

DHBW Stuttgart

- Raspberry PI
- Netzwerkdiagnose
  - Kommandos
  - Analyse-Software Wireshark
  - Arbeiten mit Wireshark
- Asterisk – VoIP Einführung
- Asterisk Software
- Asterisk Programmierung

Test und Diagnose Tools: ipconfig / ifconfig

DHBW Stuttgart

- IP Verbindungsanalyse (Connectivity)
  Zeigt die eigene IP- und MAC-Adresse an
  Windows: ipconfig (im DOS-Fenster) Linux/Mac: ifconfig
- Beispiel:

```
Ethernetadapter LAN-Verbindung 3:
        Verbindungspezifisches DNS-Suffix: Speedport_W_700V
         Beschreibung. . . . . . . . . . . : Ethernetadapter der AMD-PCNET-Familie #2
        Physikalische Adresse . . . . . . : 08-00-27-35-47-D6
        DHCP aktiviert. . . . . . . . . . : Ja
        Autokonfiguration aktiviert . . . : Ja
        IP-Adresse. . . . . . . . . . . . : 192.168.2.102
        Subnetzmaske. . . . . . . . . . . : 255.255.255.0
        Standardgateway . . . . . . . . . : 192.168.2.1
        DHCP-Server . . . . . . . . . . . : 192.168.2.1
        DNS-Server. . . . . . . . . . . . : 192.168.2.1
        Lease erhalten. . . . . . . . . . : Freitag, 6. September 2013 16:16:04
        Lease läuft ab. . . . . . . . . . : Dienstag, 10. September 2013 16:16:04
```

Test und Diagnose Tools: Netstat

DHBW Stuttgart

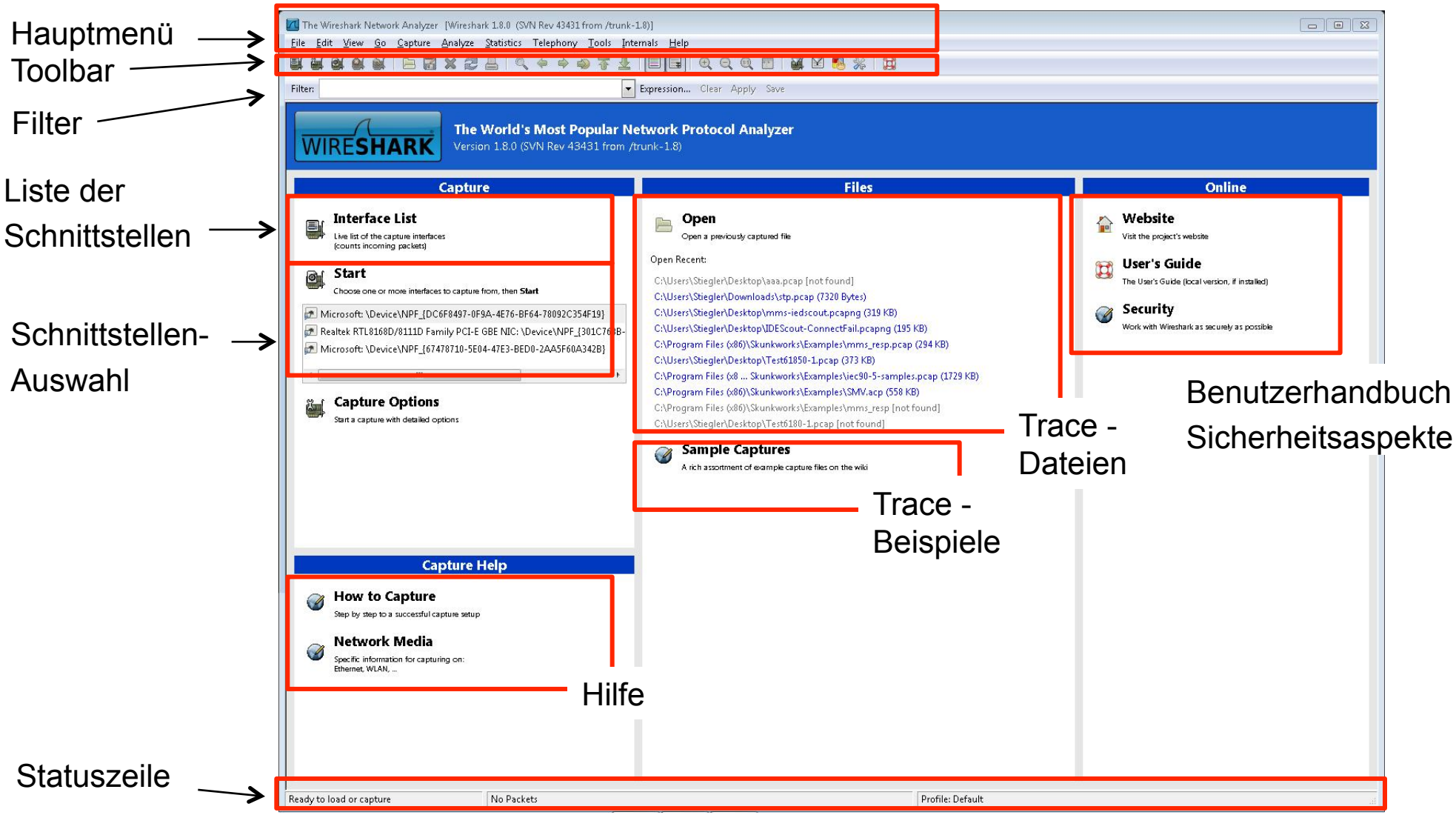
- IP Verbindungsanalyse (Connectivity)
  - Zeigt die aktiven Verbindungen (Windows: im CMD-Fenster: netstat)
- Beispiel

Aktive Verbindungen

Proto	Lokale Adresse	Remoteadresse	Status
TCP	vm-win:1201	localhost:44080	HERGESTELLT
TCP	vm-win:1203	localhost:44080	HERGESTELLT
TCP	vm-win:1205	localhost:44080	SCHLIESSEN_WARTEN
TCP	vm-win:1214	localhost:44080	HERGESTELLT
TCP	vm-win:44080	localhost:1201	HERGESTELLT
TCP	vm-win:44080	localhost:1203	HERGESTELLT
TCP	vm-win:44080	localhost:1205	FIN_WARTEN_2
TCP	vm-win:44080	localhost:1214	HERGESTELLT
TCP	vm-win:1202	95.100.97.67:http	HERGESTELLT
TCP	vm-win:1204	62.159.74.11:http	HERGESTELLT
TCP	vm-win:1215	62.156.238.46:http	HERGESTELLT

Protokollanalyse mit Wireshark : Startmenü

DHBW Stuttgart



Wireshark :Toolbar

DHBW Stuttgart


Interface Auswahl


Optionen Auswahl


START Trace


STOP Trace


STOP+Restart Trace


Datei öffnen


Datei speichern


Datei schließen


Aktuelle Trace Datei noch einmal öffnen


Drucken Dialog


Suchen Dialog


Capture Filter Dialog


Display Filter Dialog


Einstellungen Dialog


Farb-Einstellungen


Hilfe

Cursor bewegen:


Rückwärts


Vorwärts


Springen zu


Zum 1. Paket


Zum letzten Paket


Ausgabe vergrößern


Ausgabe verkleinern

Originalgröße

Wireshark Bildschirmbereiche

DHBW Stuttgart

Menü und Funktionsauswahl

Filter

Nachrichten-Liste

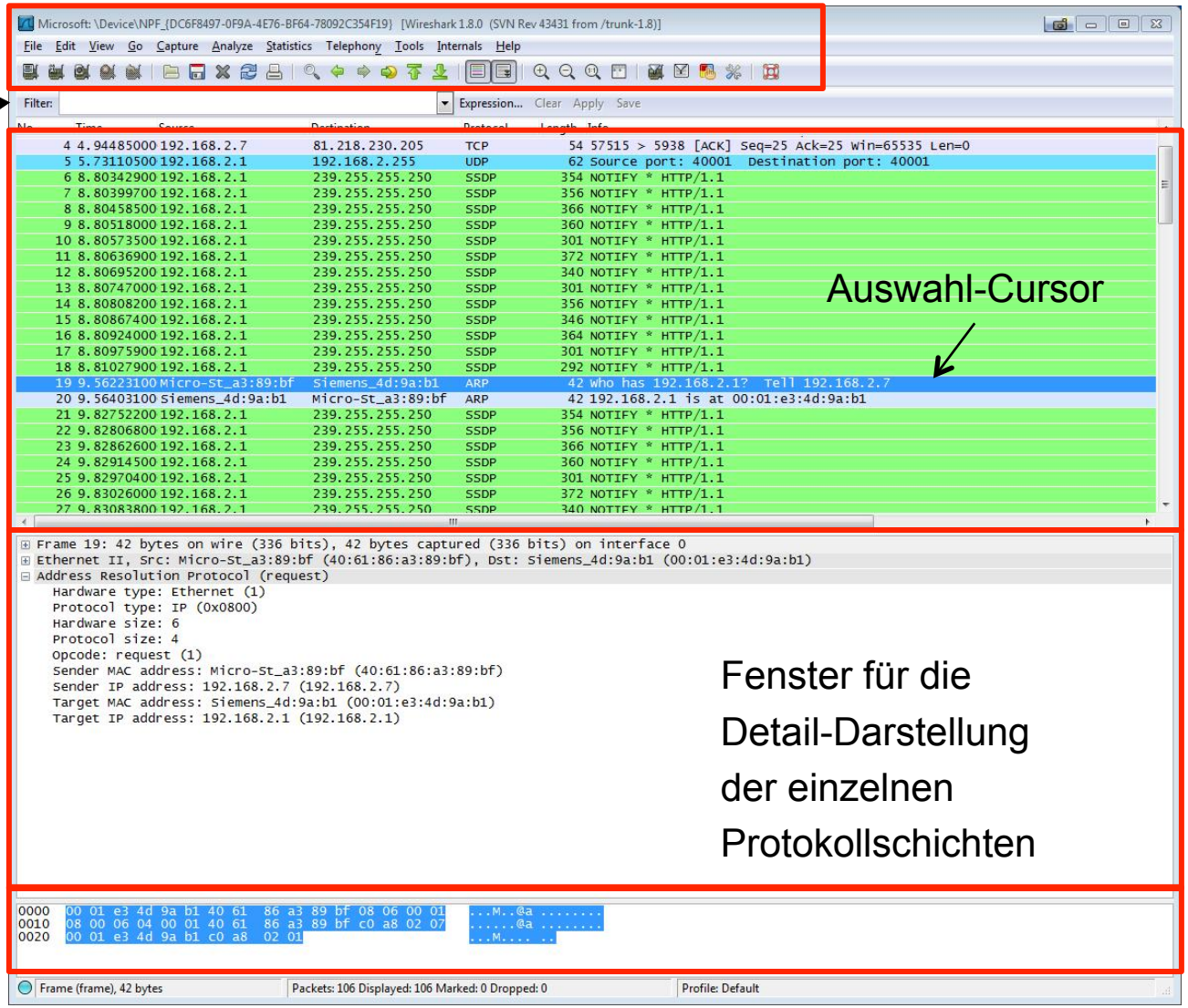
Auswahl-Cursor

Protokollschichten

Fenster für die Detail-Darstellung der einzelnen Protokollschichten

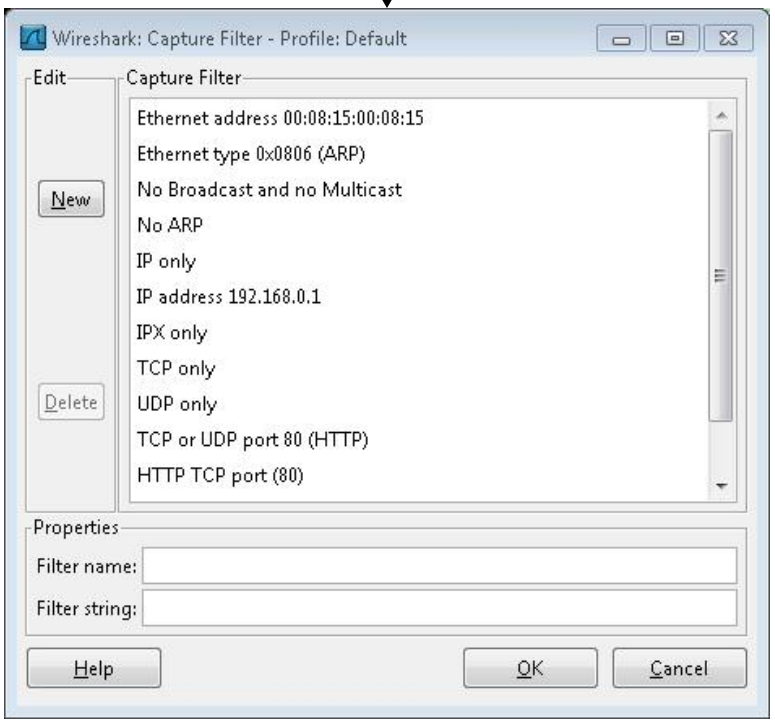
Hexadezimal-Darstellung (Hex-Dump)

Fußzeile



Filter-Arten

- **Capture Filter**:
  - Hauptmenü – Capture – Capture Filters …
  - Aufnahme-Filter
    Datenmenge wird bei der **Aufnahme** gefiltert
- **Display Filter**:
  - Hauptmenü – Analyze – Display Filters …
  - Anzeige-Filter
    Datenmenge wird bei der **Wiedergabe** gefiltert

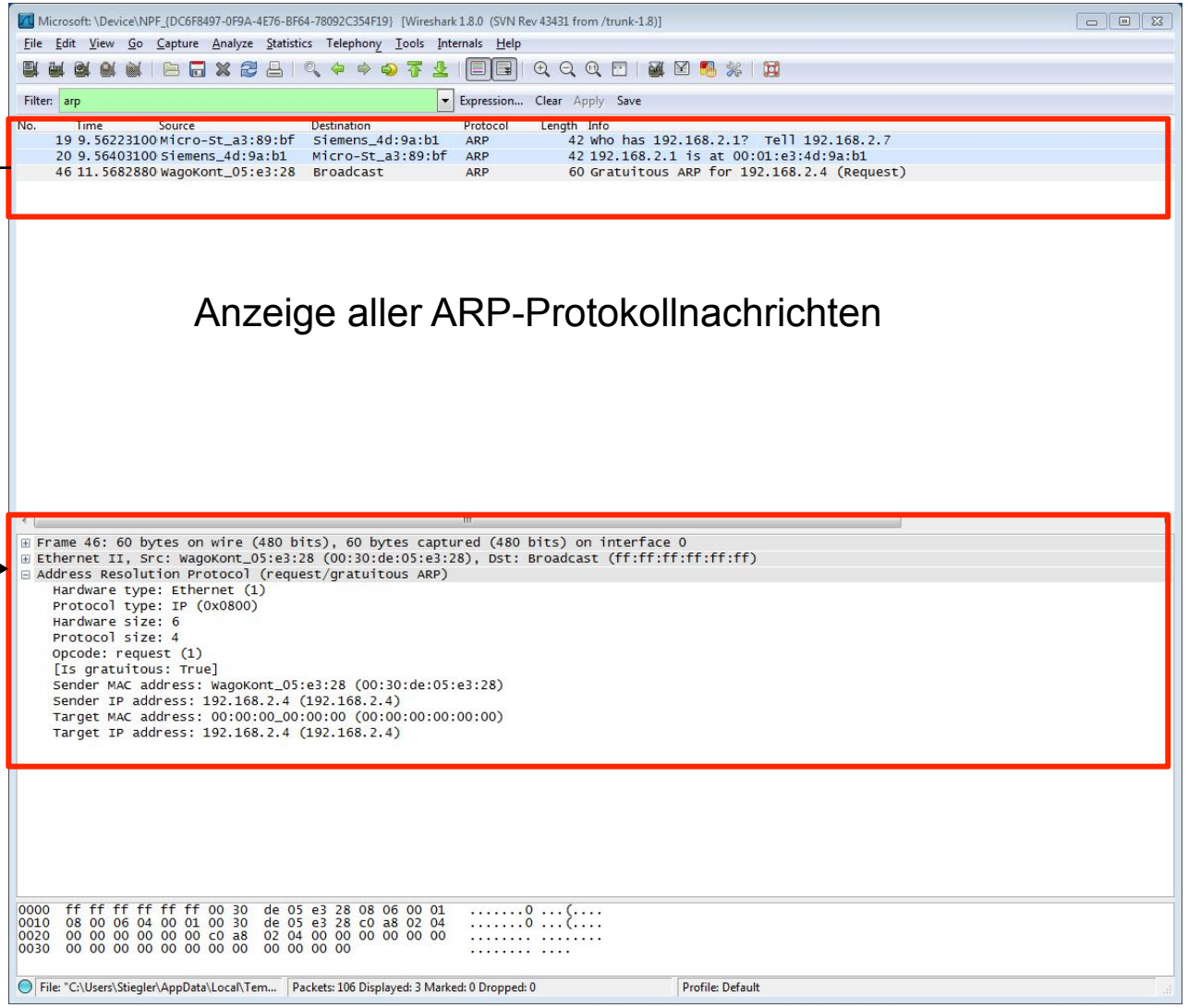


Protokoll-Filter Direkteingabe

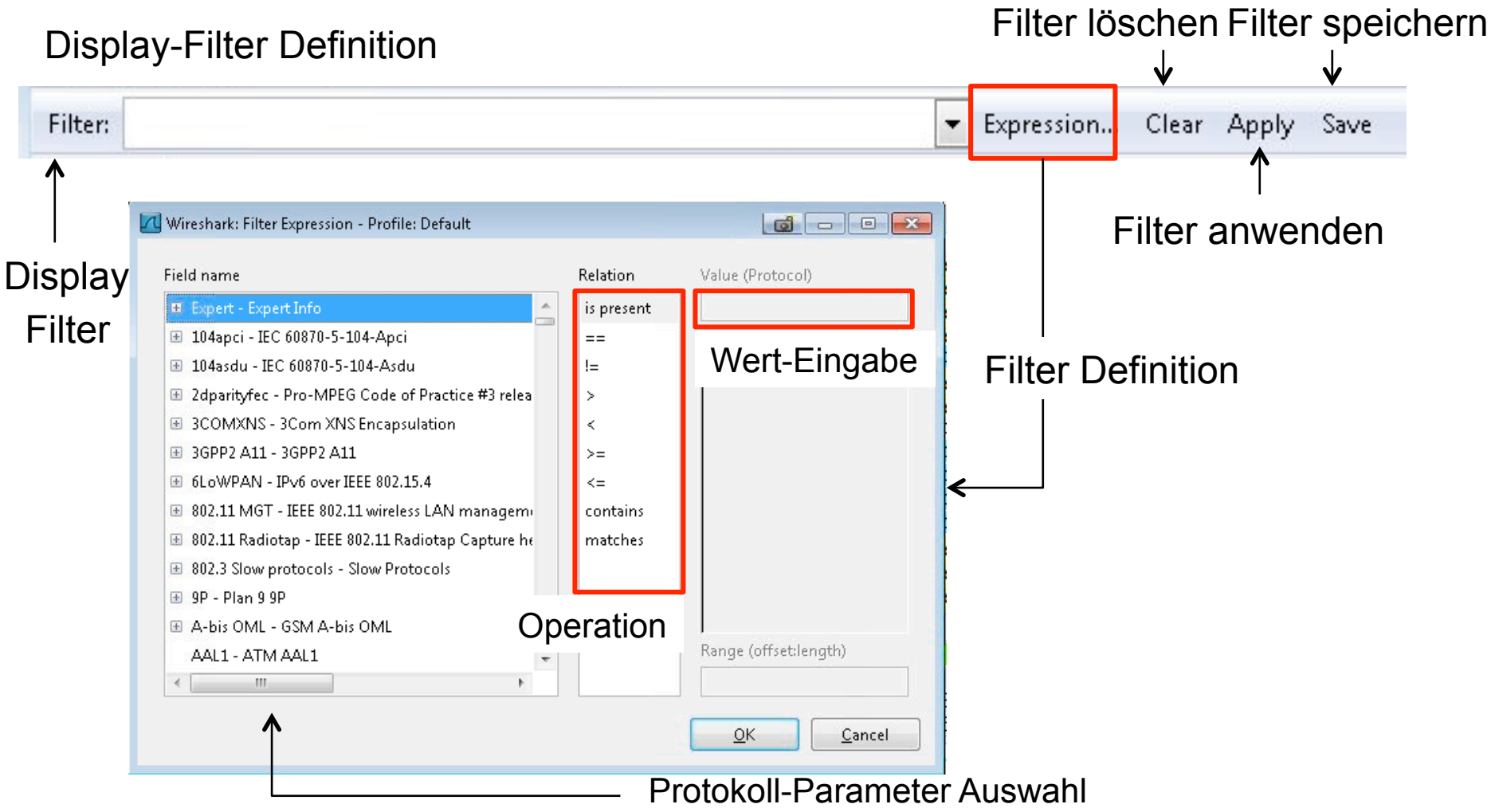
Filter = **arp**
Nur ARP-Nachrichten werden angezeigt

Dekodierung der ausgewählten Nachricht

ARP: Address Resolution Protocol

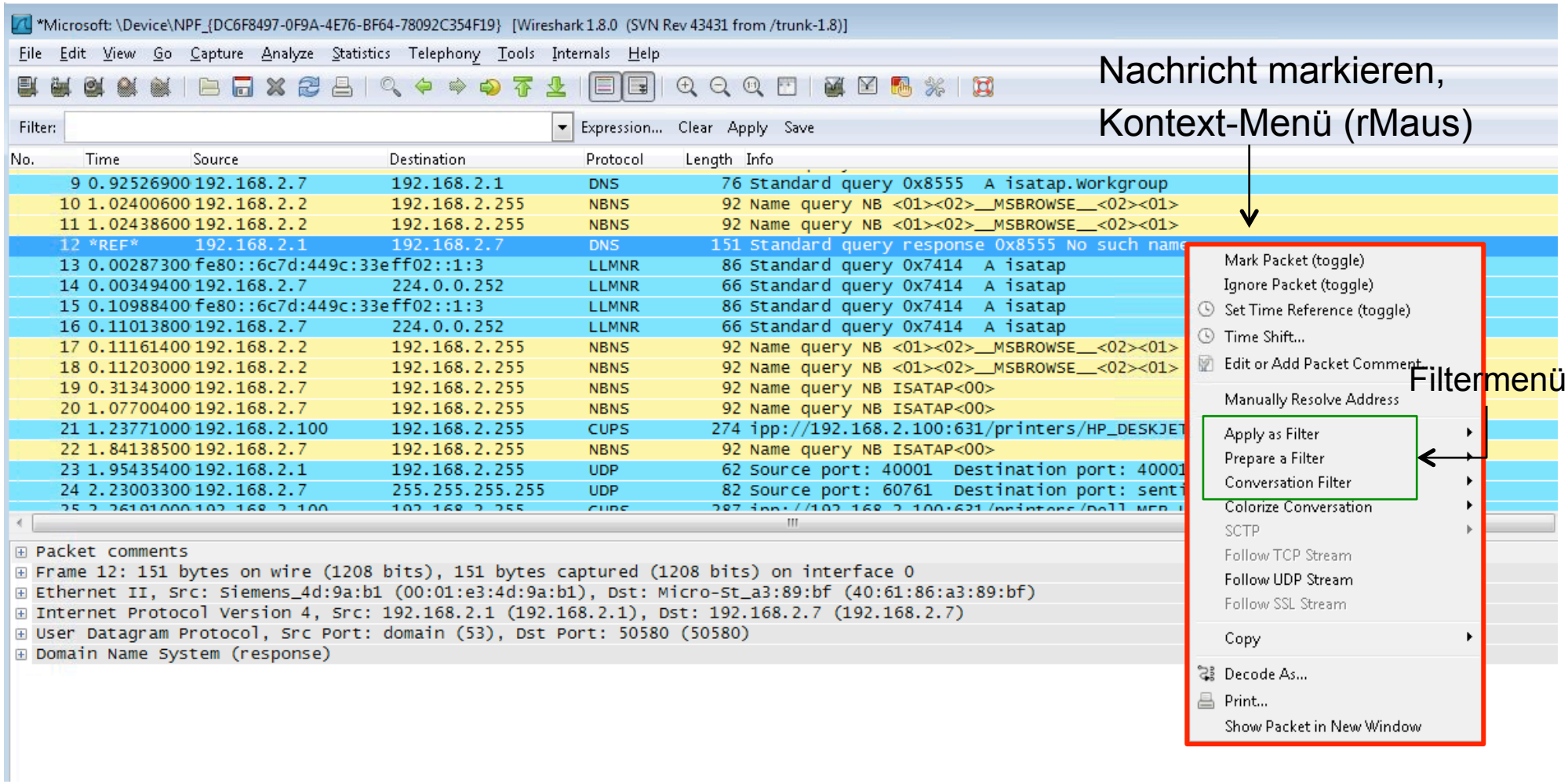


Manuelle Protokoll-Filter Definition



Automatische Protokoll-Filter Definition

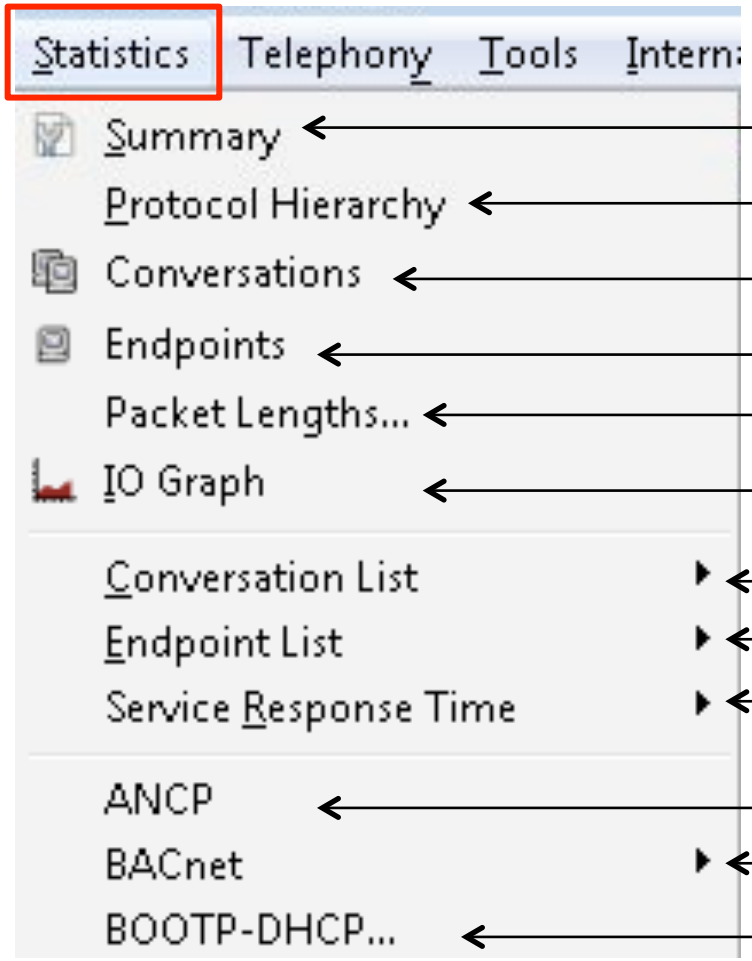
DHBW Stuttgart



Statistik-Menü (1)

BAC: Building Automation and Control

Hauptmenü



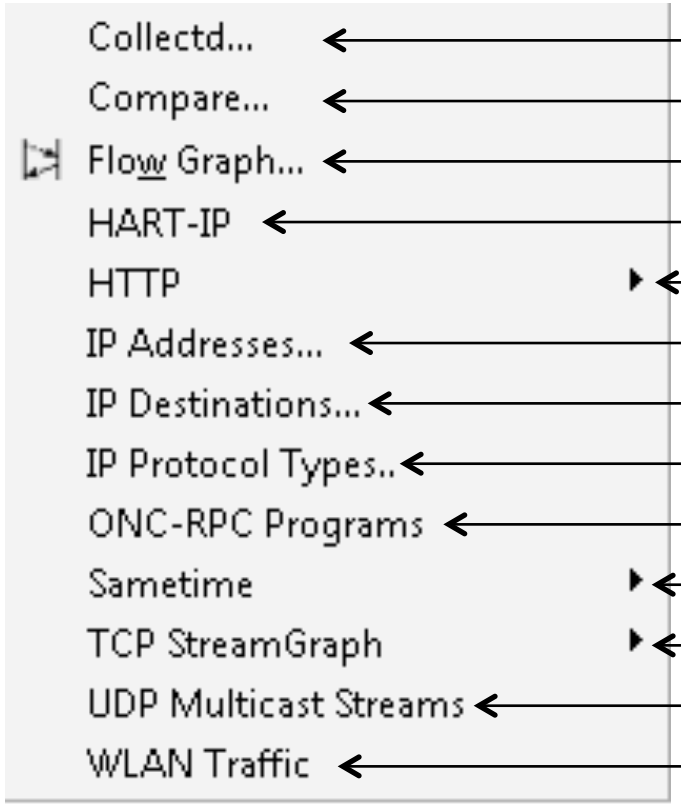
- Summary: Zusammenfassung der Trace-Daten
- Protocol Hierarchy: Trace-Daten: Protokollstatistik
- Conversations: Kommunikations-Statistik
- Endpoints: Adressen-Statistik
- Packet Lengths...: Statistik: Paket-Länge
- IO Graph: Statistik: Zeitverteilung
- Conversation List: Liste der Verbindungen
- Endpoint List: Liste der Adressen-Endpunkte
- Service Response Time: Liste der Antwortzeiten
- ANCP: Access Node Control Protocol Statistik
- BACnet: BAC-Network Statistik
- BOOTP-DHCP...: Bootstrap-Protocol und DHCP Statistik

Statistik-Menü (2)

Fortsetzung:

HART-IP: Highway Addressable Remote Transducer over IP

ONC-RPC: RFC 1831 Network File System (NFS) - Protokoll

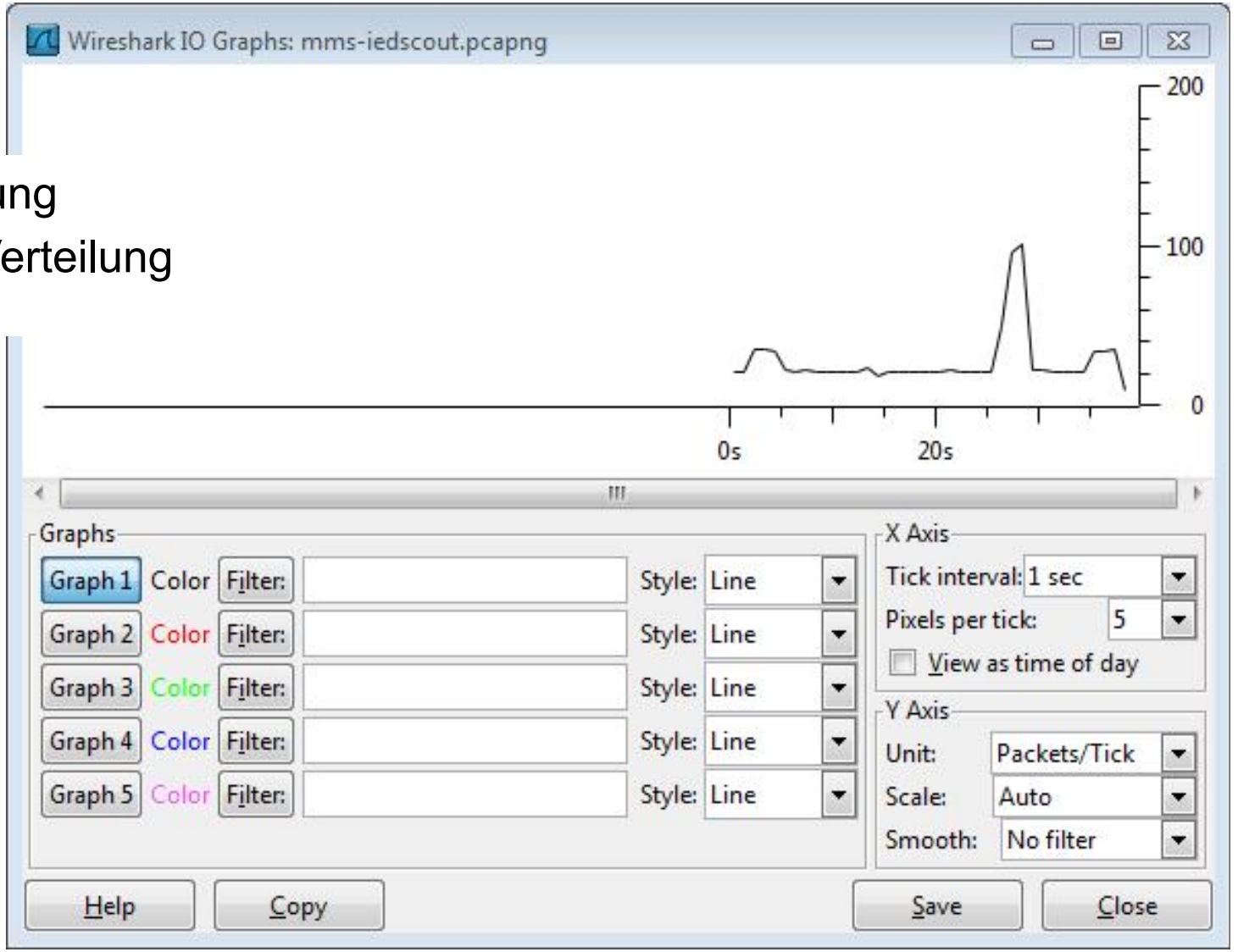


- Paketzähler und Filter
- Vergleich von Capture-Dateien
- Flussdiagramm erzeugen
- HART-IP Statistik
- Statistik: Paket-Zähler, Requests, Lastverteiluung
- Statistik: IP-Adressenverteilung
- IP-Adressen, Transportschicht und Portnummer
- Liste der Transportverbindungen
- Liste der ONC-RPC Applikationen
- Anzahl Nachrichten mit gleichem Zeitstempel
- TCP-Nachrichtentransport Statistik
- Liste der UDP-Multicast Streams
- WLAN - Verkehrsdaten

Statistik-Beispiel: Lastverteilung

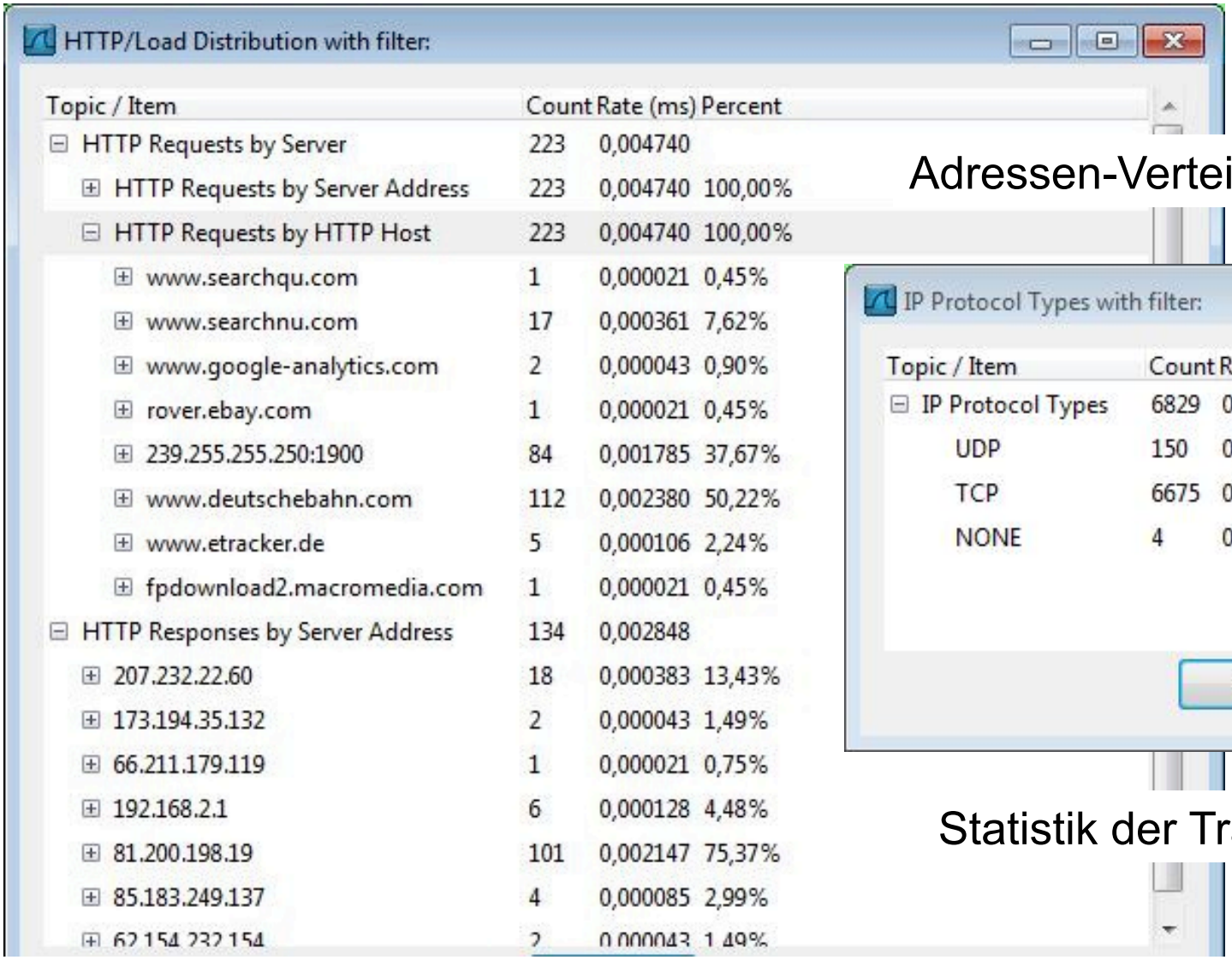
DHBW Stuttgart

Diese Darstellung
zeigt die Zeit-Verteilung
der Pakete

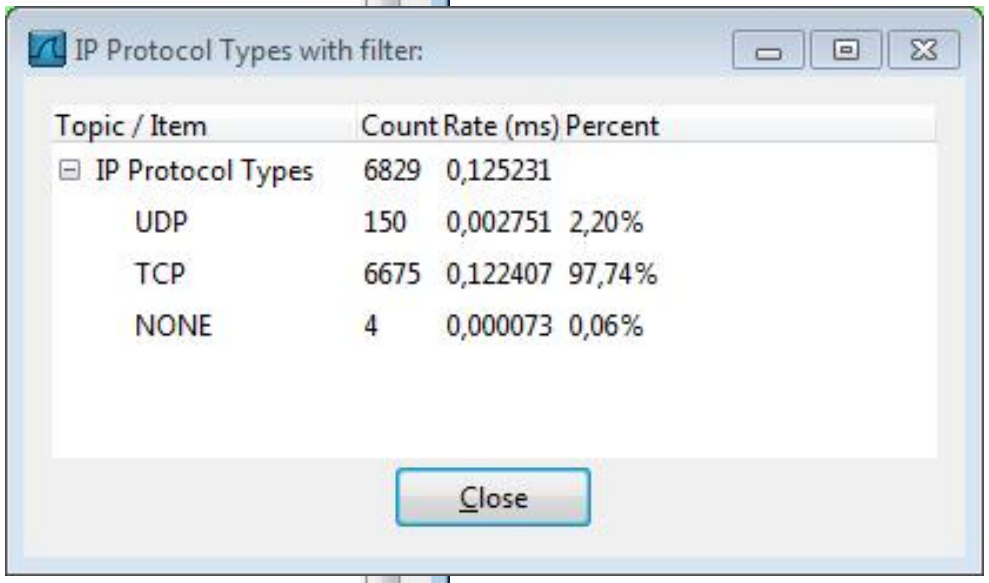


Statistik-Beispiel: Adressen- und Protokolle

DHBW Stuttgart



Adressen-Verteilung der Pakete



Statistik der Transportprotokolle

Kursgliederung

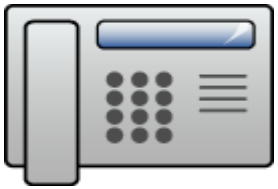
DHBW Stuttgart

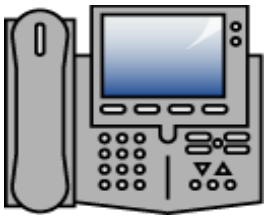
- Raspberry PI
- Netzwerkdiagnose
  - Kommandos
  - Analyse-Software Wireshark
  - Arbeiten mit Wireshark
- Asterisk – VoIP Einführung
- Asterisk Software
- Asterisk Programmierung

Evolution der Telefon-Dienste

DHBW Stuttgart

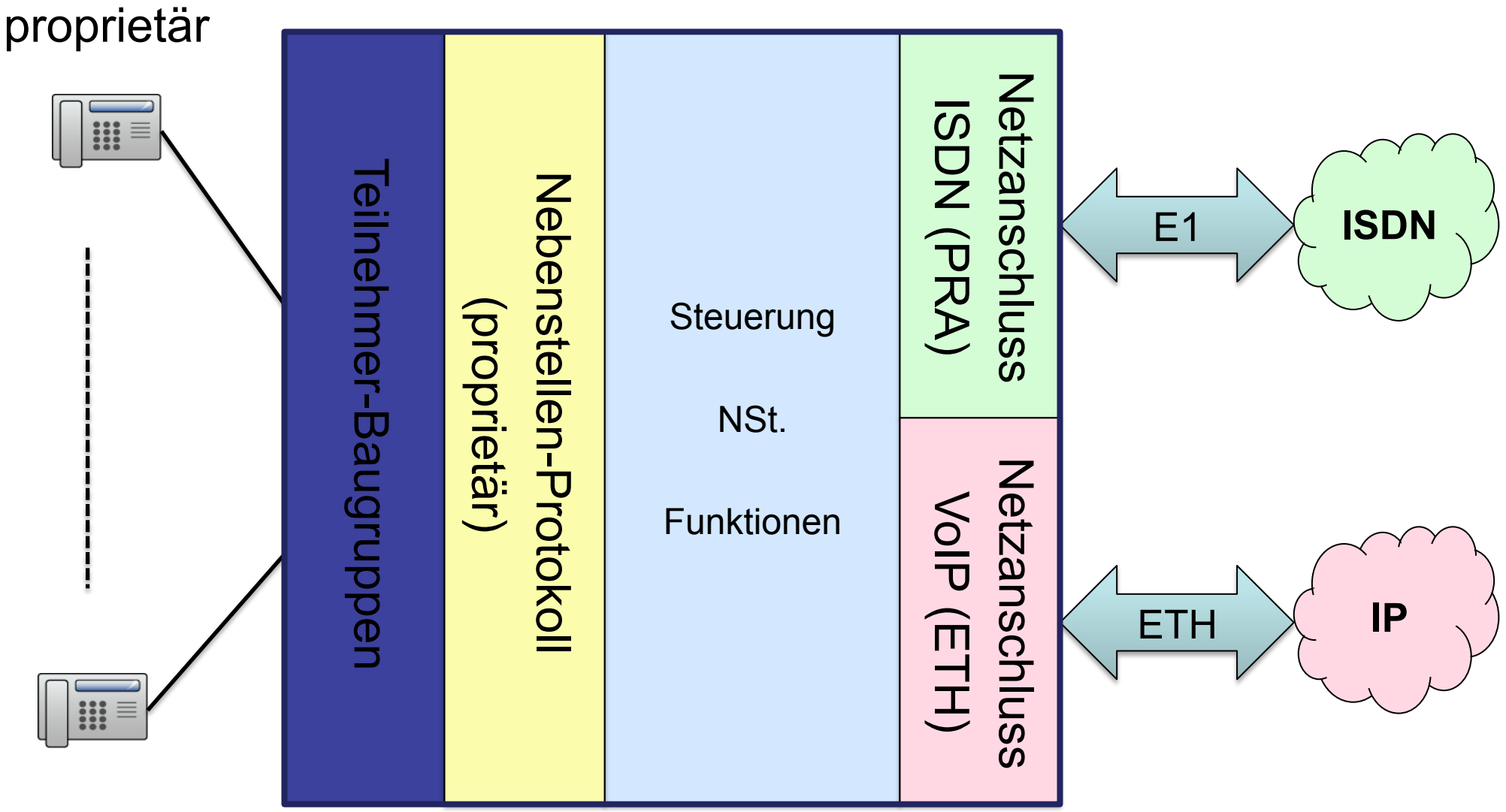
Analog → ISDN → VoIP



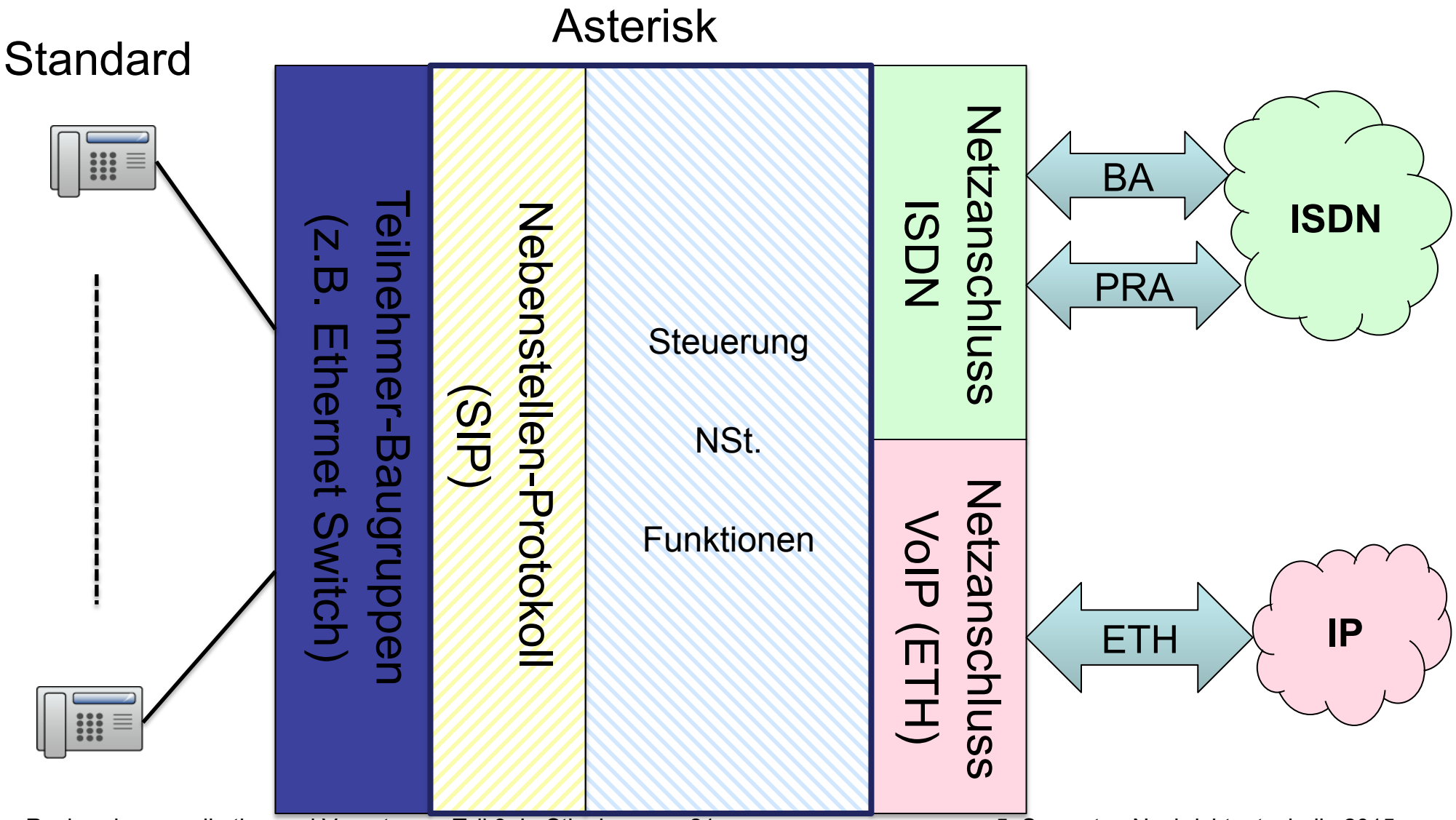


Dienste		
gering	Umfangreich	Umfangreich
eingeschränkte Nummernanzeige falls Display verfügbar	Nummernanzeige Rufumleitungen CCBS etc.	komplexe Implement. Datenintegration Comm. Server Open source universell

Nebenstellentechnik (Hardware)

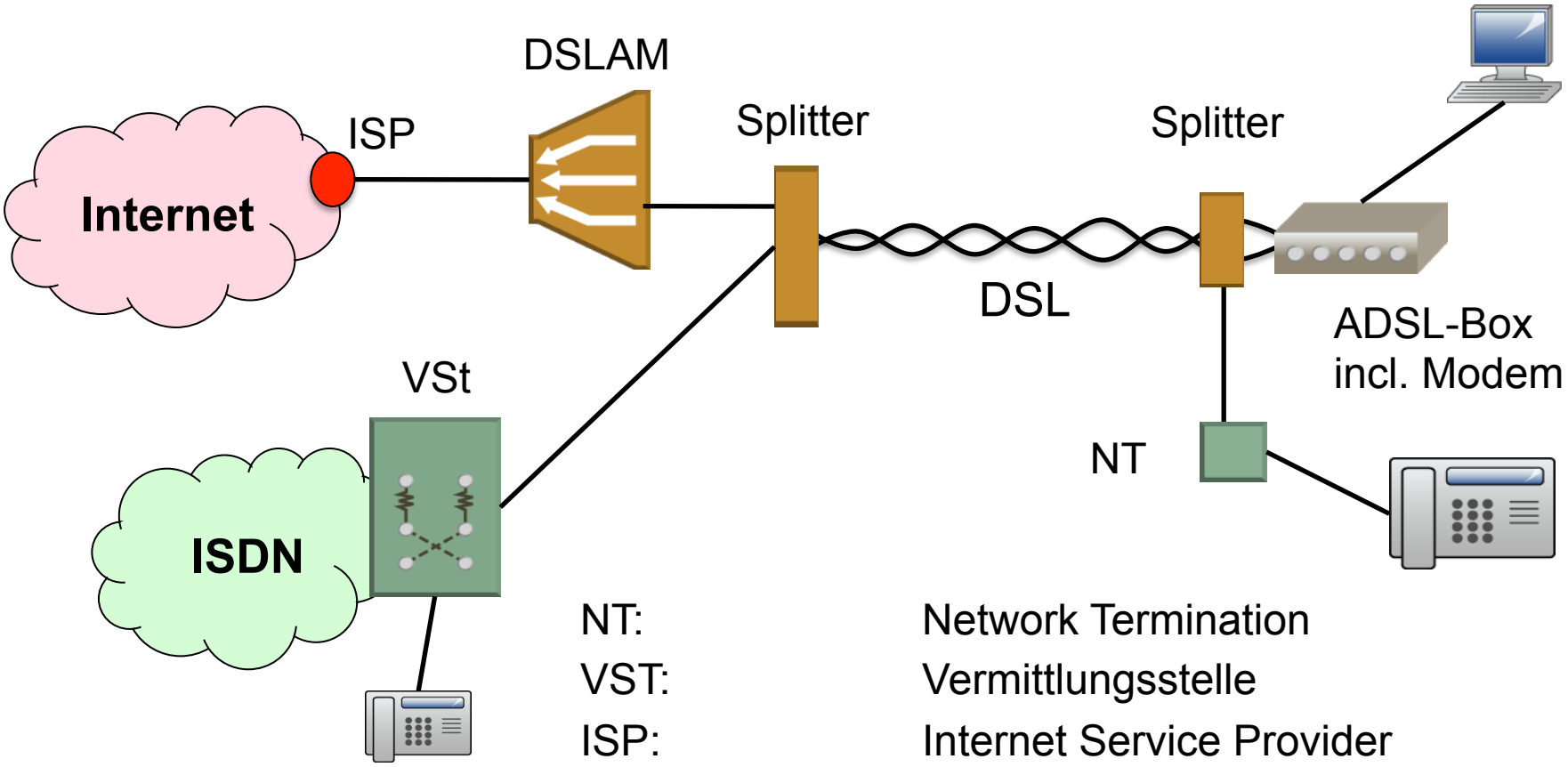


Nebenstellentechnik (Software)



DHBW Stuttgart

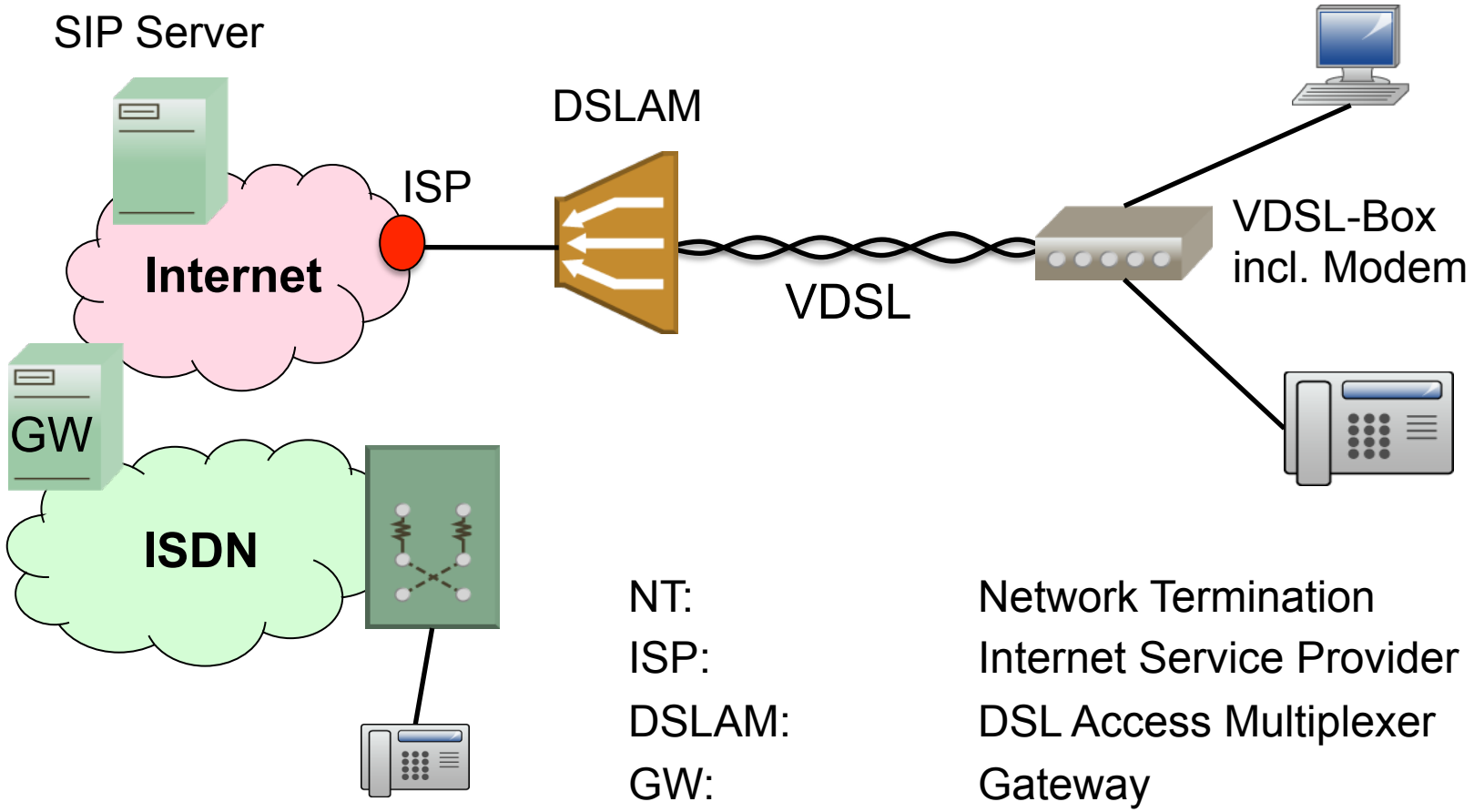
ADSL: Trennung von Sprache und Daten



NT:	Network Termination
VST:	Vermittlungsstelle
ISP:	Internet Service Provider
DSLAM:	DSL Access Multiplexer

DHBW Stuttgart

VDSL: Sprache und Daten kombiniert

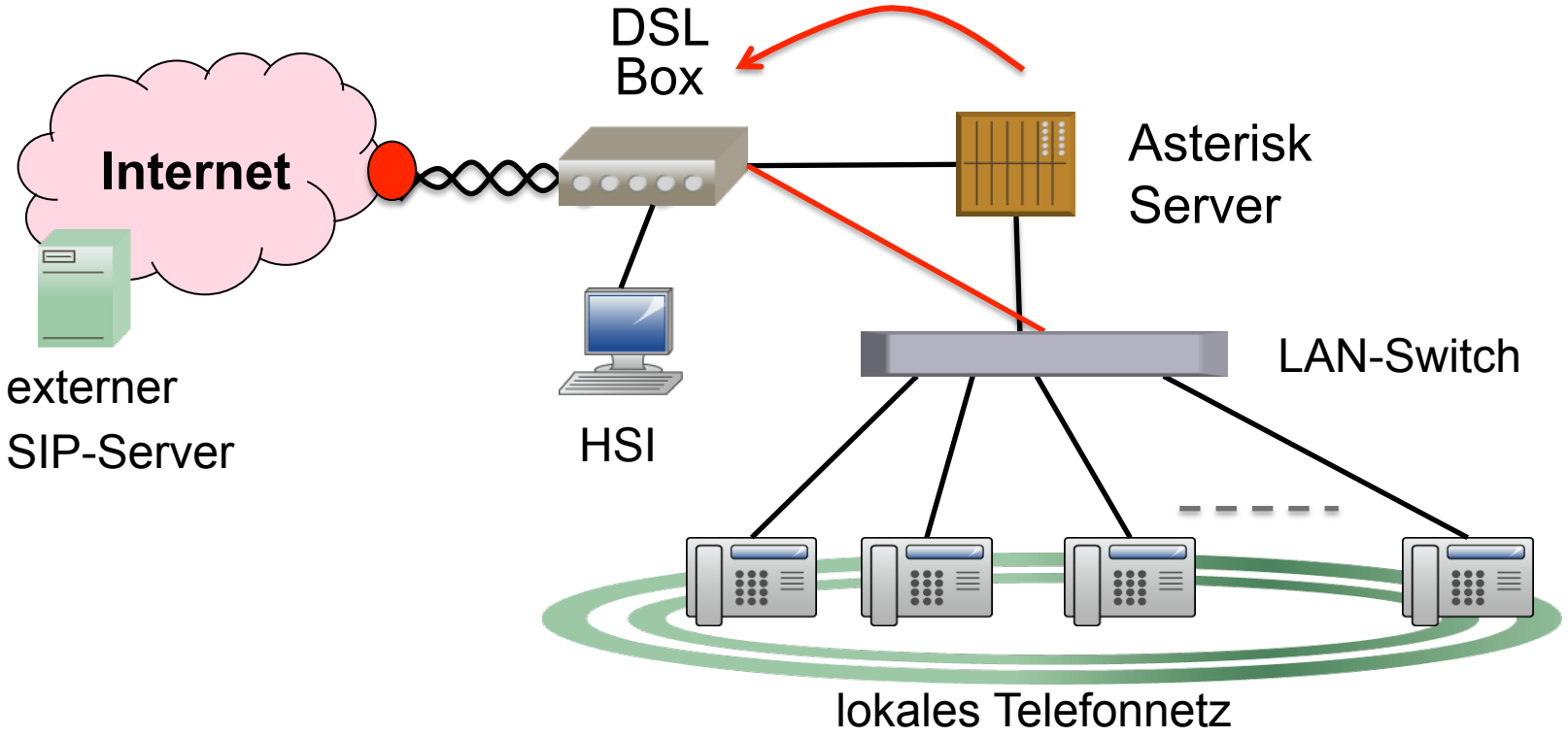


NT:	Network Termination
ISP:	Internet Service Provider
DSLAM:	DSL Access Multiplexer
GW:	Gateway

Asterisk Netzeinbindung

DHBW Stuttgart

Asterisk als lokale VoIP Vermittlungsstelle



HSI: High-Speed Internet

SIP Konzept

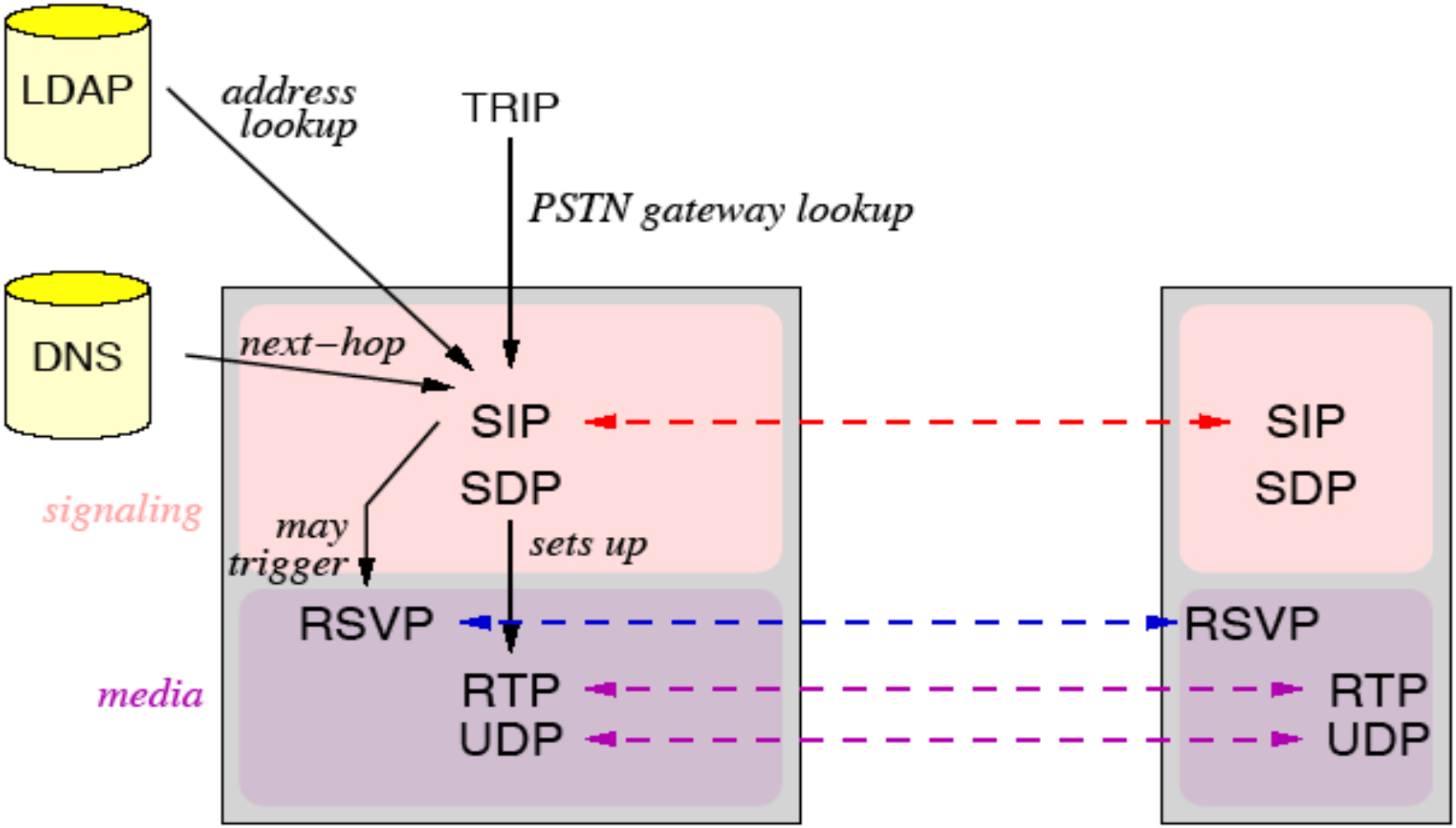
Die Protokollfamilie des Session Initiation Protocol (SIP) bildet eine Multimedia Architektur.

Andere dazu gehörende Protokolle sind :

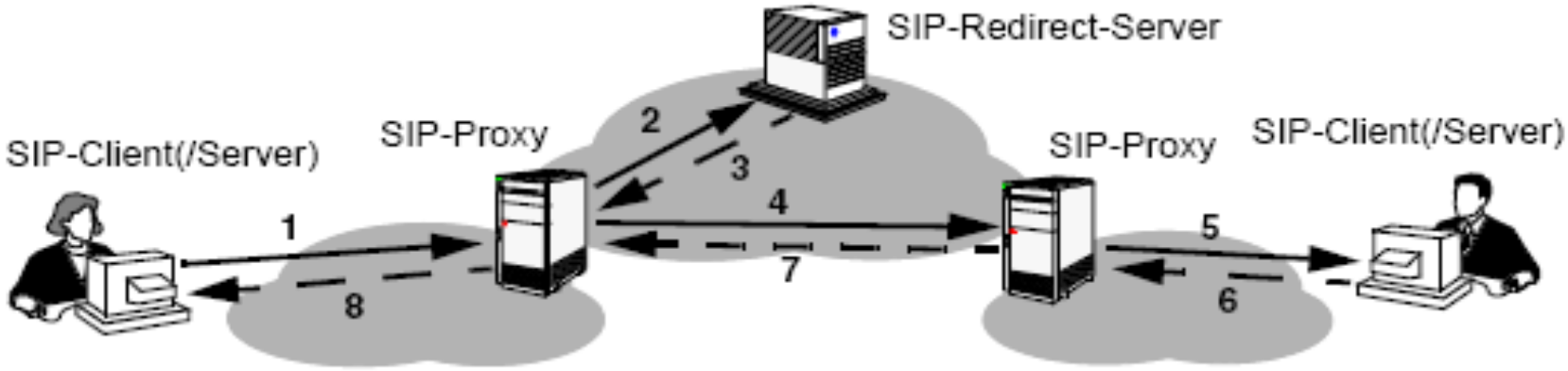
- Real Time Transport Protocol (RTP)
- Real Time Control Protocol (RTCP)
- Session Description Protocol (SDP)
- Real Time Streaming Protocol (RTSP)
- Gateway Control Protocol (MEGACO) etc.

Die grundlegenden SIP Funktionen werden durch diese Protokolle ergänzt damit vollständige Multimediadienste angeboten werden können.

SIP Protokollfamilie



SIP Netzelemente

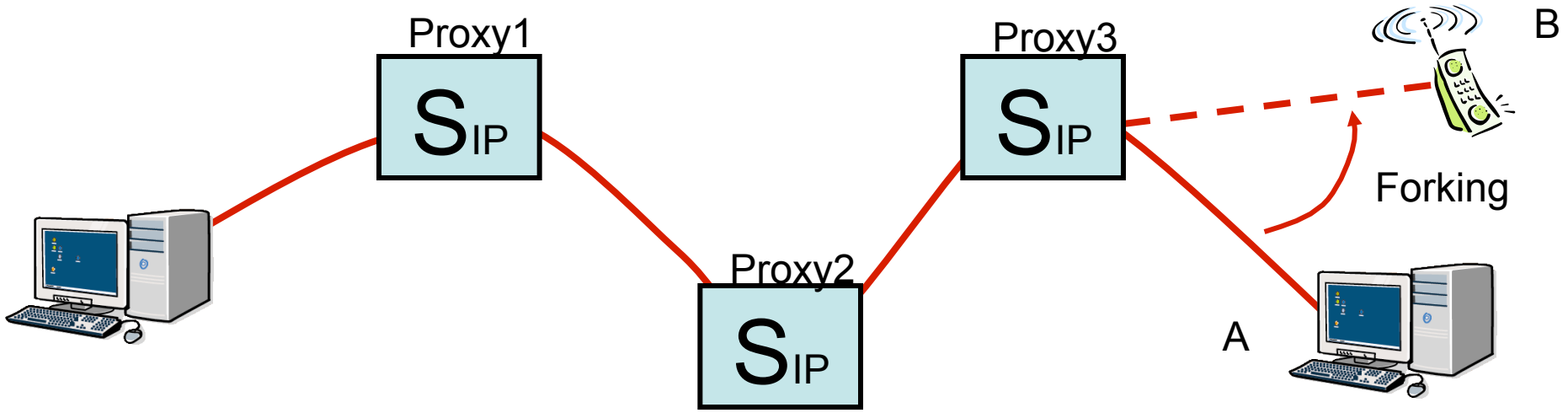


Nachrichtenfolge:

1. User Verbindungsaufbau (INVITE message) zu einem SIP-Proxy
2. Zieladresse wird vom redirection server ermittelt
3. Antwort: Zieladresse (z.B. Rufumleitung)
4. INVITE Nachricht zum Ziel-Proxy
5. INVITE zum SIP-Zielendgerät

6. – 8 Antworten vom SIP-Zielendgerät über den Signalisierungspfad

SIP Message Routing



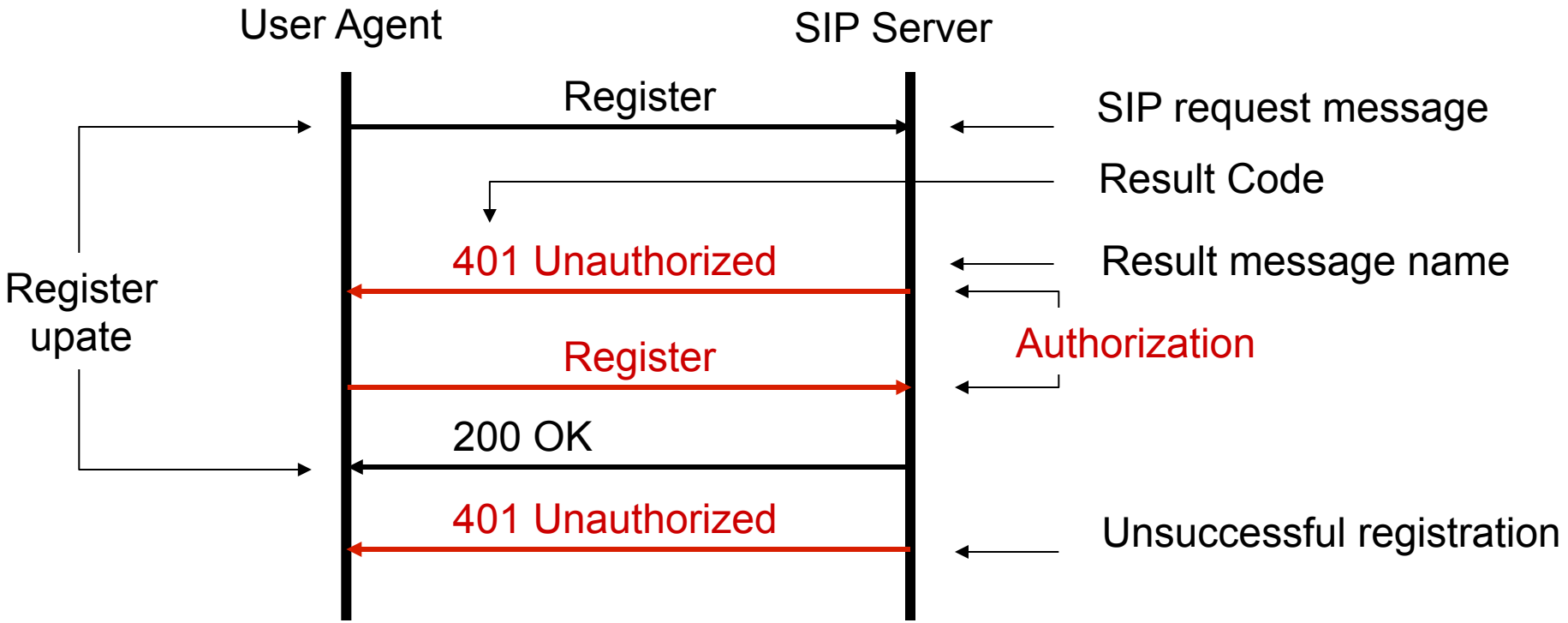
Pfad: SIP Proxy-1 - SIP Proxy-2 - SIP Proxy-3

Route: wird verwendet, um ein SIP-Request über Proxyrechner zum Ziel und zurück zu leiten Diese “Route list” + “Contact” - Parameter heissen “Route Set”.

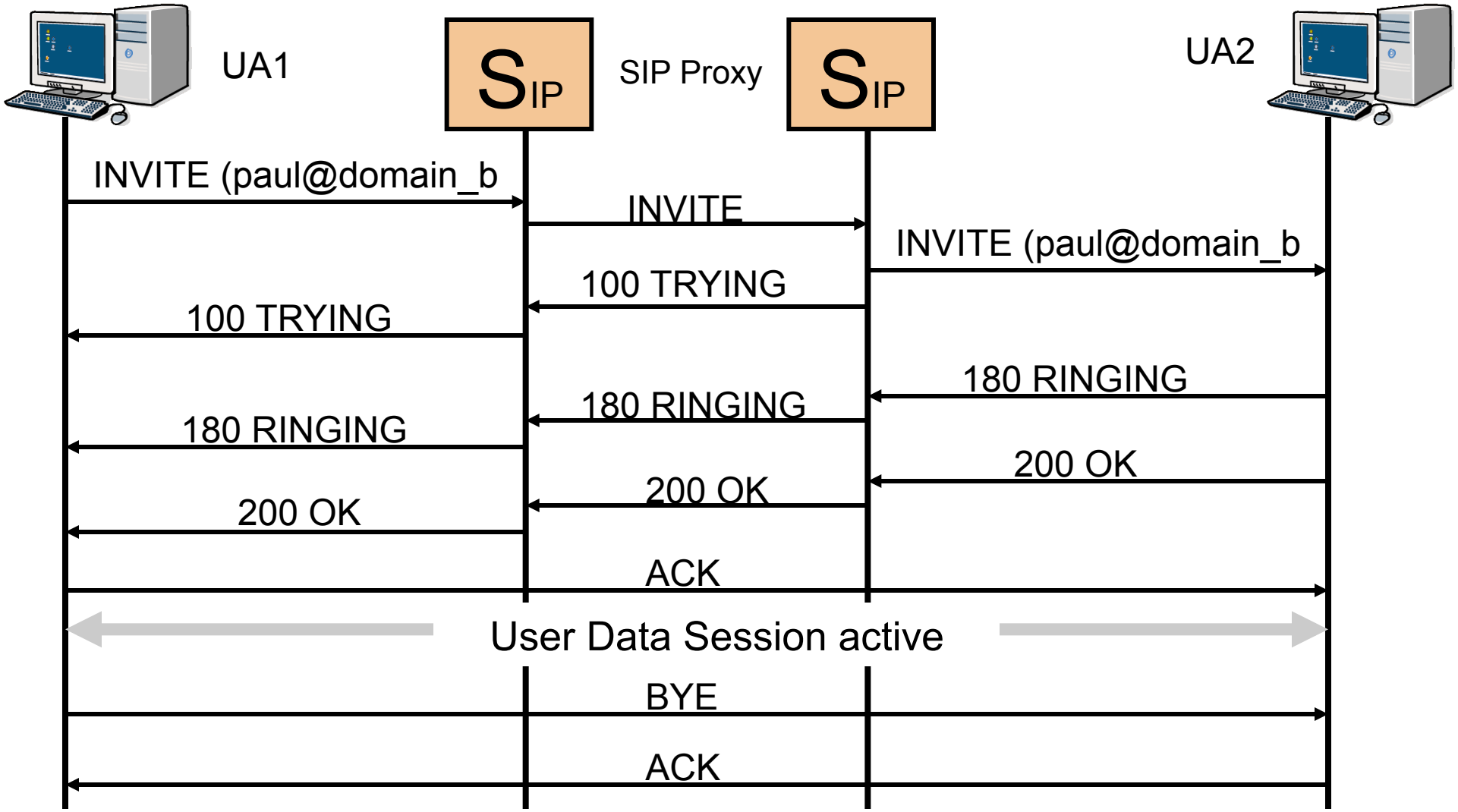
Register Prozedur

Die Registrierung verbindet eine Geräteadresse mit einem SIP user Address of Record (AOR)

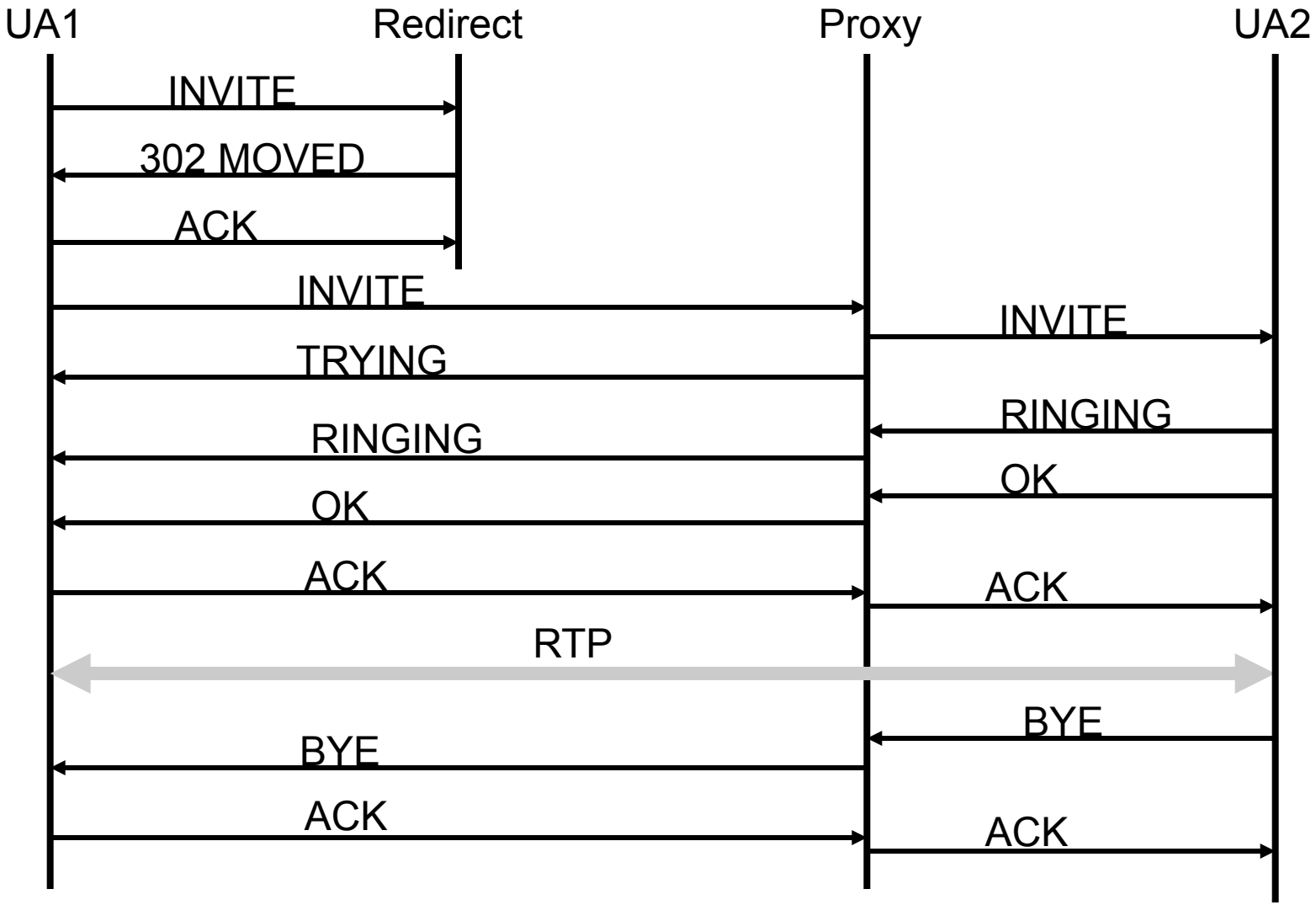
Die Registrierung läuft nach einer gewissen Zeit aus und muss periodisch erneuert werden



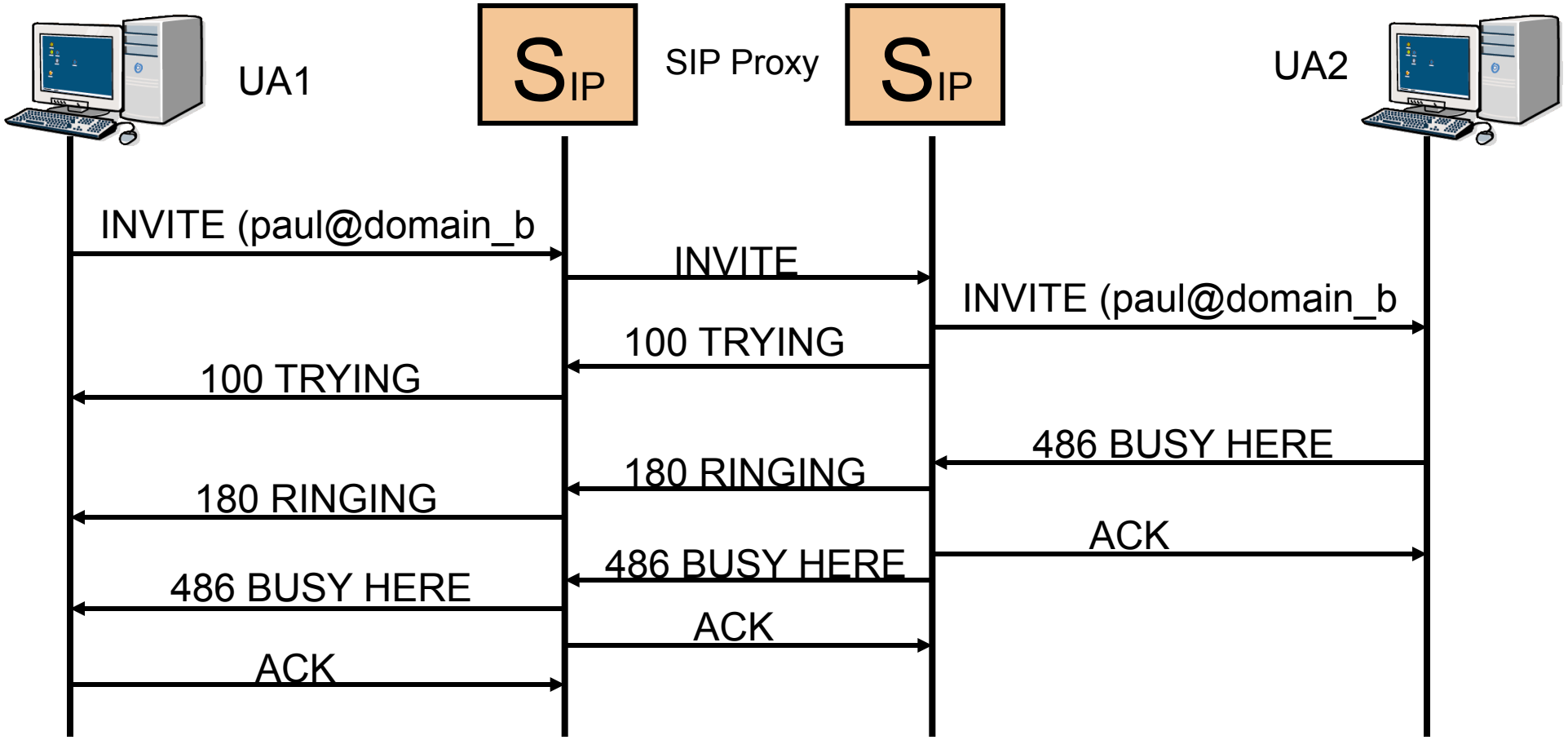
Verbindungsaufbau



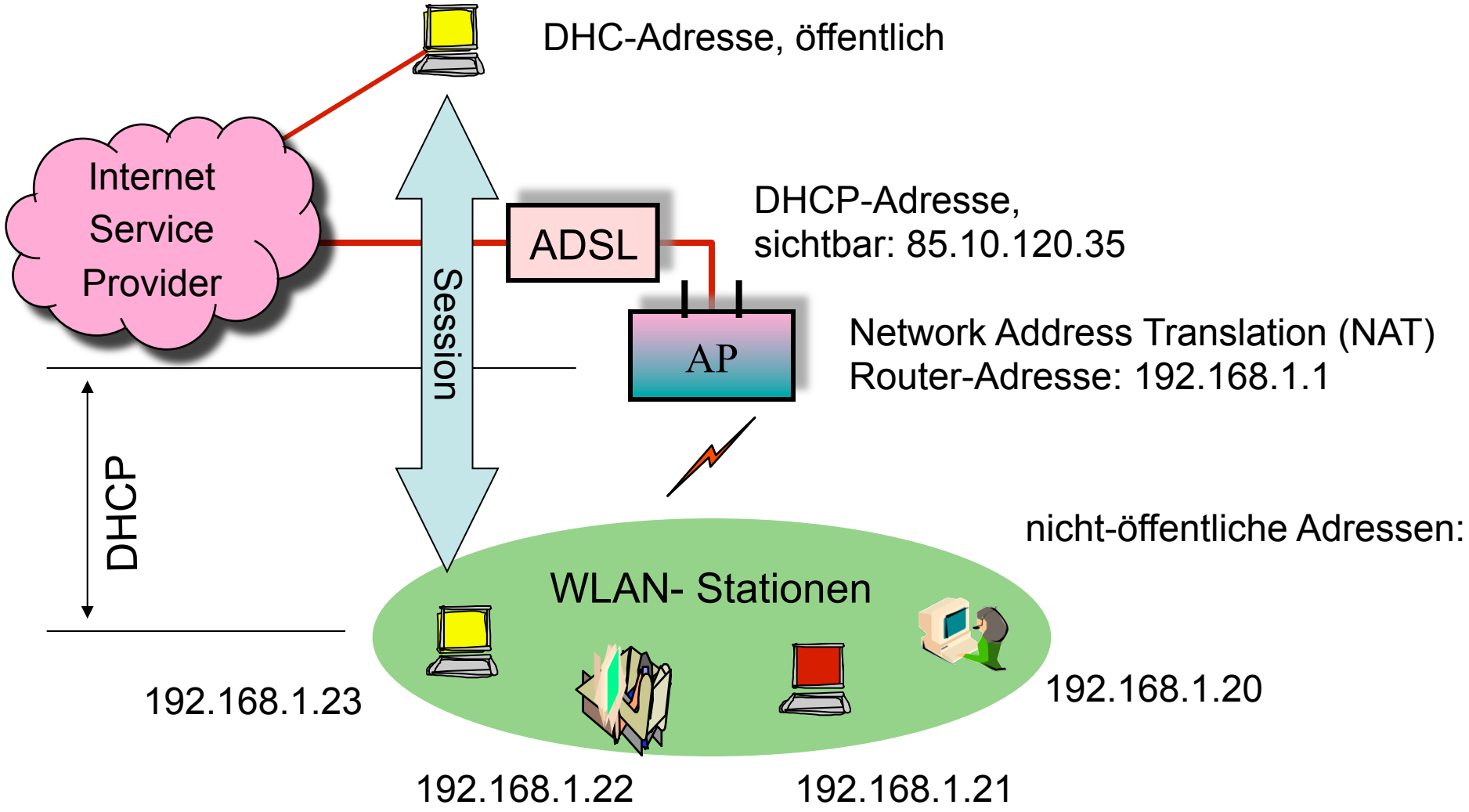
Redirect Server



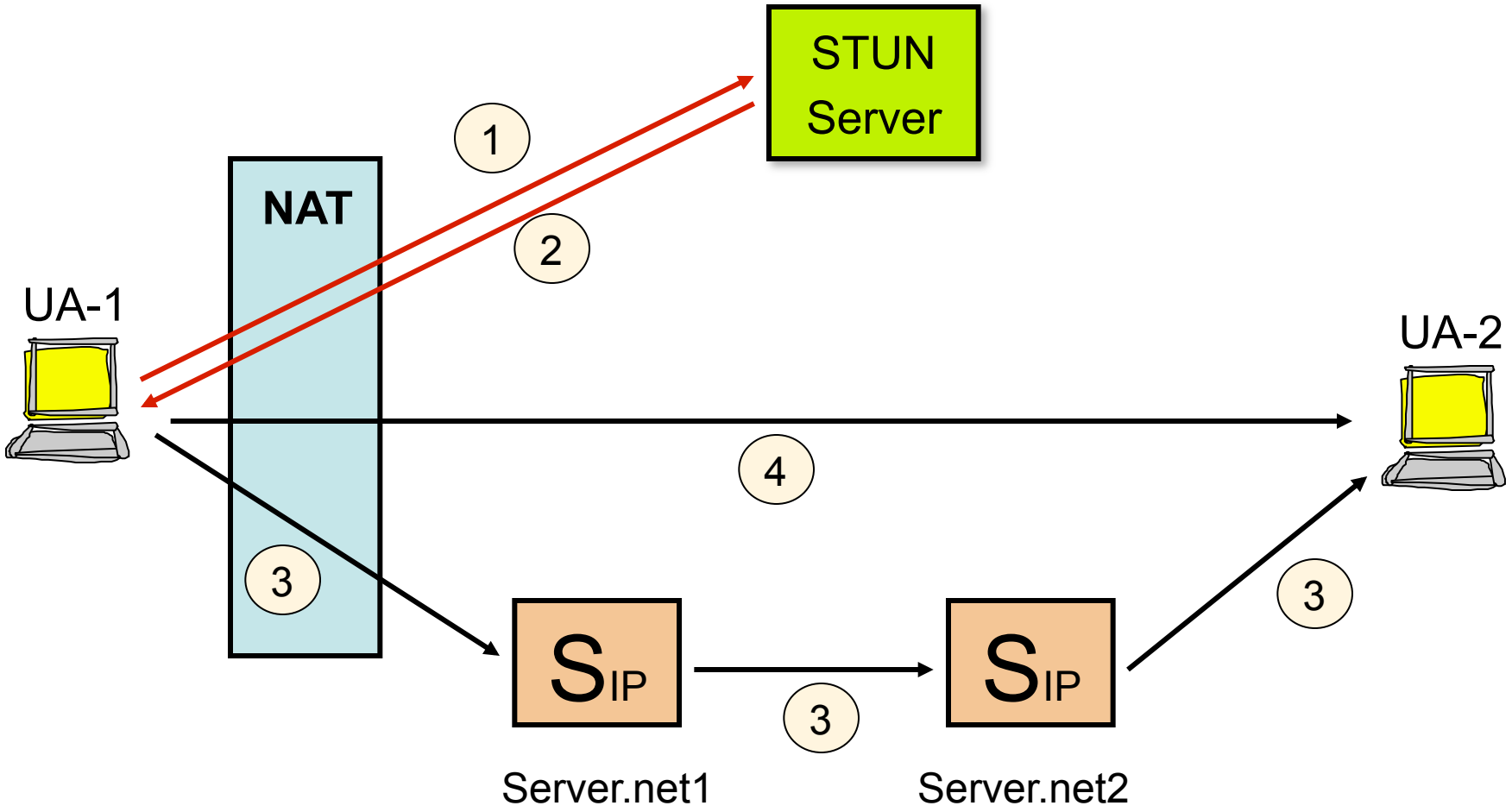
Teilnehmer Besetzt (User Busy)



NAT Traversal: Beispielkonfiguration

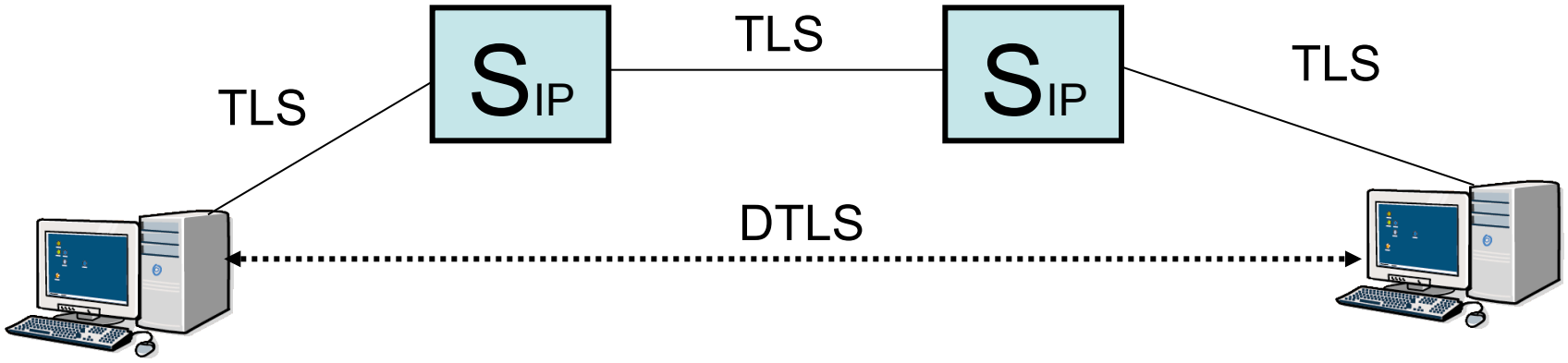


NAT Traversal



DHBW Stuttgart

Channel Security, AA



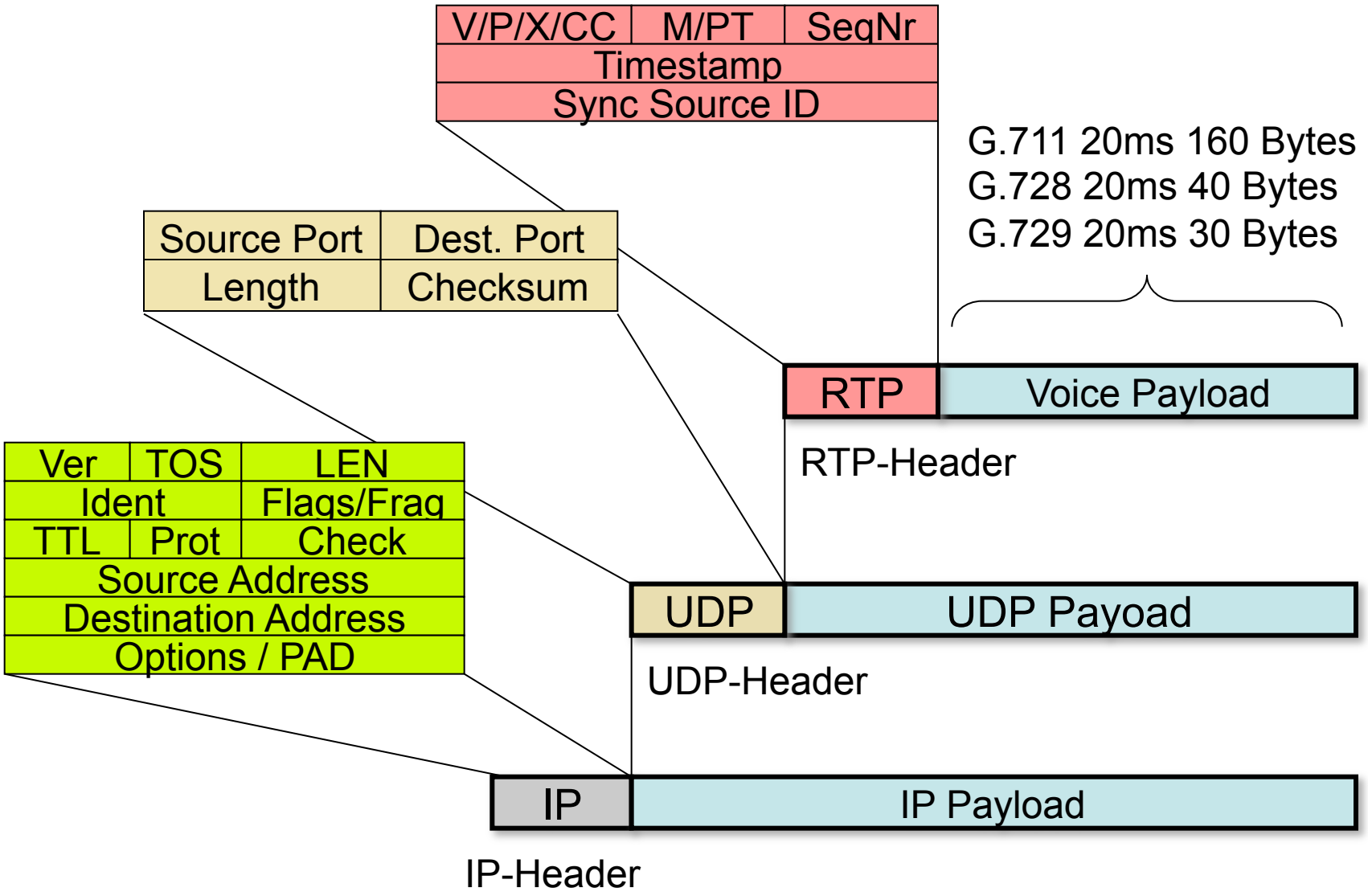
Verwendet Transport Layer Security (TLS)

Datagram TLS (DTLS) für UDP

Authentisierung : Proxy überprüft user
Proxy überprüfen einander
UA verifiziert proxy
UA(S) verifiziert UA(C) mittels S/MIME

AA: Authentication & Authorization

RTP Protokollschichten



RTP Payload Types

DHBW Stuttgart

Payload Art	Kodierung	Audio/Video	Abtasttakt	Kanäle
0	PCMU	A	8000	1
2	G.721	A	8000	1
3	GSM (FR)	A	8000	1
9	G.722	A	8000	1
15	G.728	A	8000	1
26	JPEG	V	90,000	n.a.
31	H.261	V	90,000	n.a.
96 - 127	dynamic	dynamic	dynamic	dynamic

Session Description Protocol (SDP)

SDP definiert in der RFC 2327.

SDP beschreibt Multimedia Sessions:

Parameter Gruppen:

- session description (e.g. Name, owner/creator ..)
- time description (Aktive Zeit, Wiederholungszeit)
- media description (Titek, Bandbreiteninfo, Verschlüsselung, ..)

SDP ermöglicht die Teilnahme an einer Multimedia Session

SDP enthält kein Transportprotokoll

SDP Protokoll-Information wird im SIP-Body transportiert

Sprachqualität

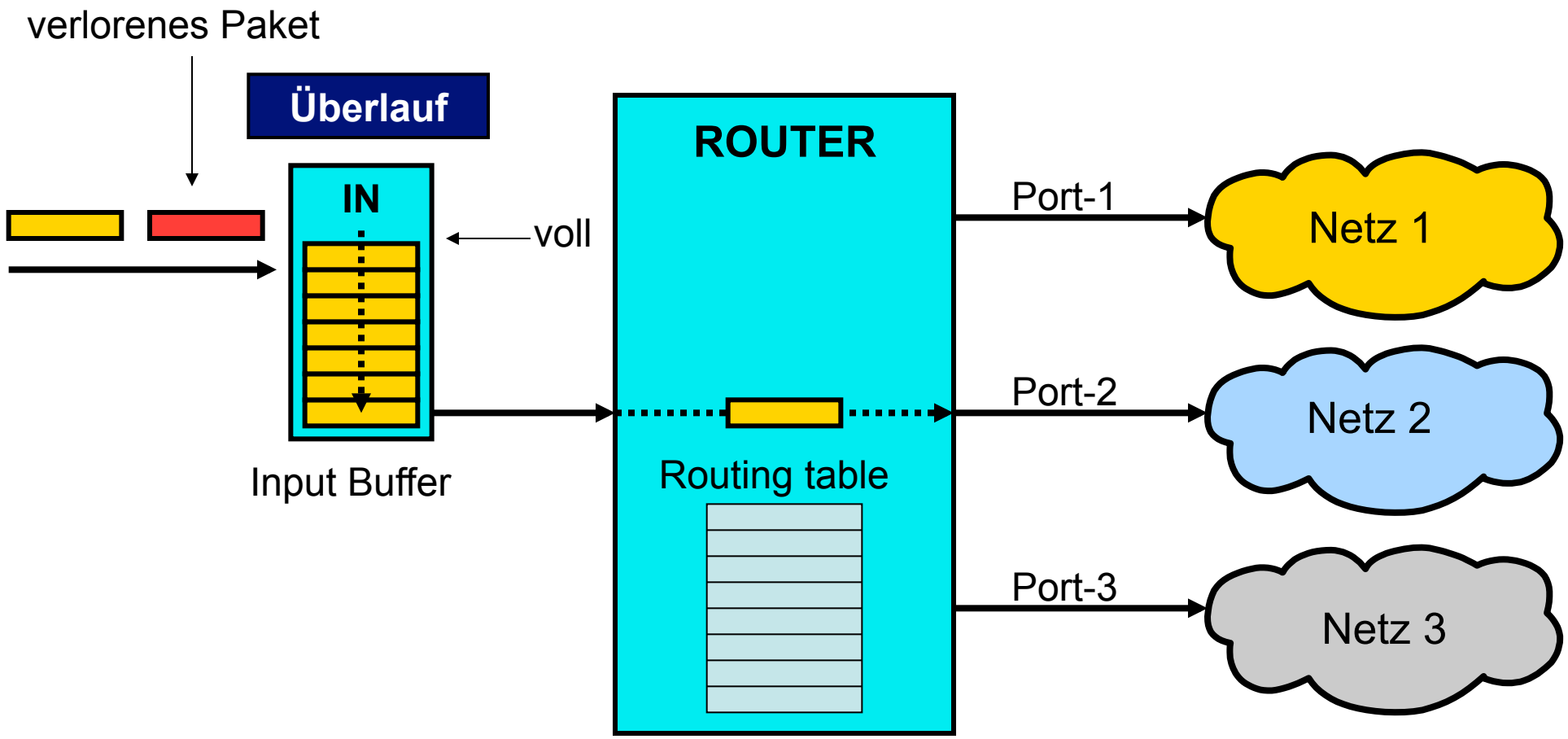
Objektive Sprachqualitätsmessungen verwenden VQA (Voice Quality Analysis) Technik.

Sbjektive Sprachqualitätsmessungen verwendet MOS (Mean Opinion Score) Skala bestehend aus 5 Stufen (excellent – bad) gemäß ITU-T P. 800.

Die Sprachqualität hängt von folgenden Faktoren ab:

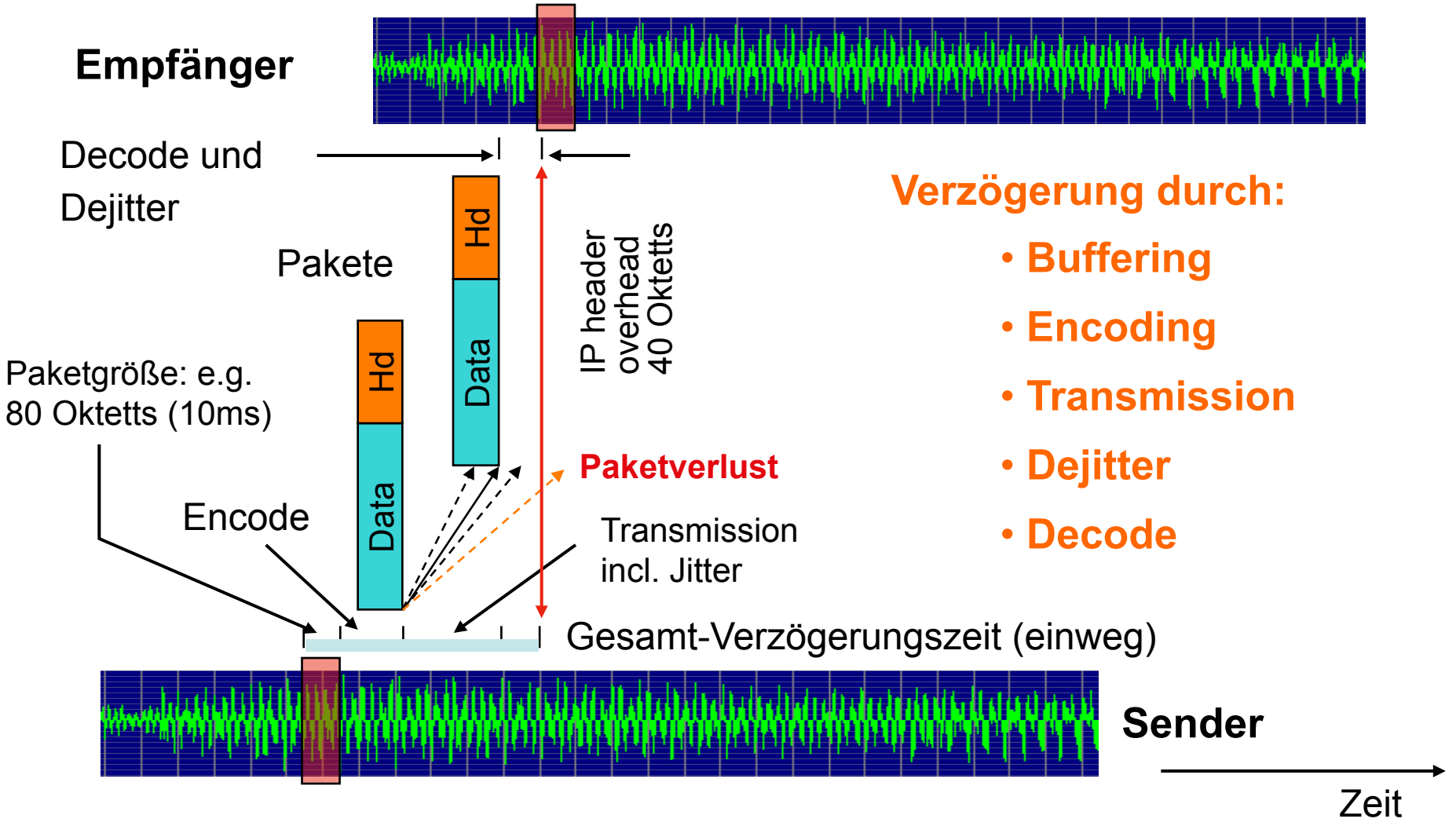
Packet Loss Rate	packets received / packets sent
End-to-end delay	packet received time - packet sent time
Delay jitter	Inter-packet delay time variation

Paketverlust



Ursache für Paketverlust:
Buffergröße nicht ausrechend

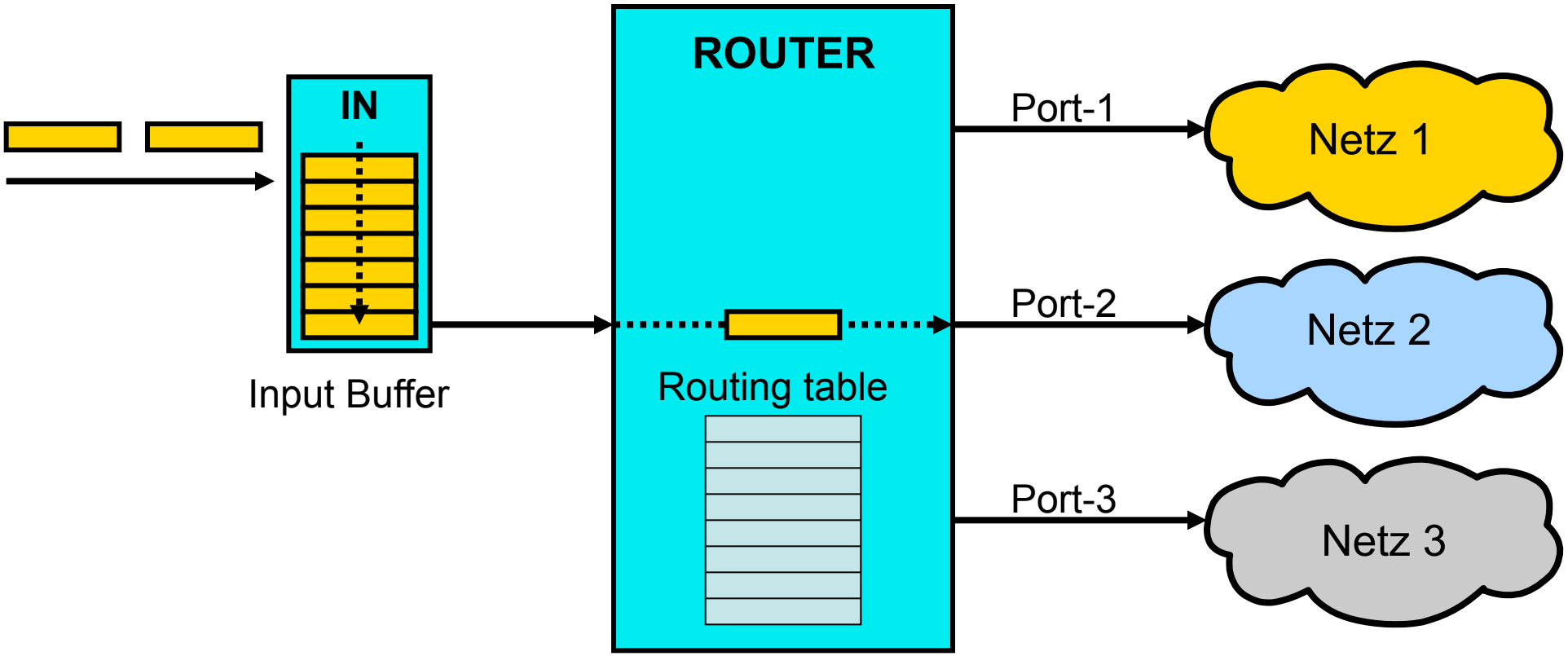
Paketverzögerung (1)



Verzögerung durch:

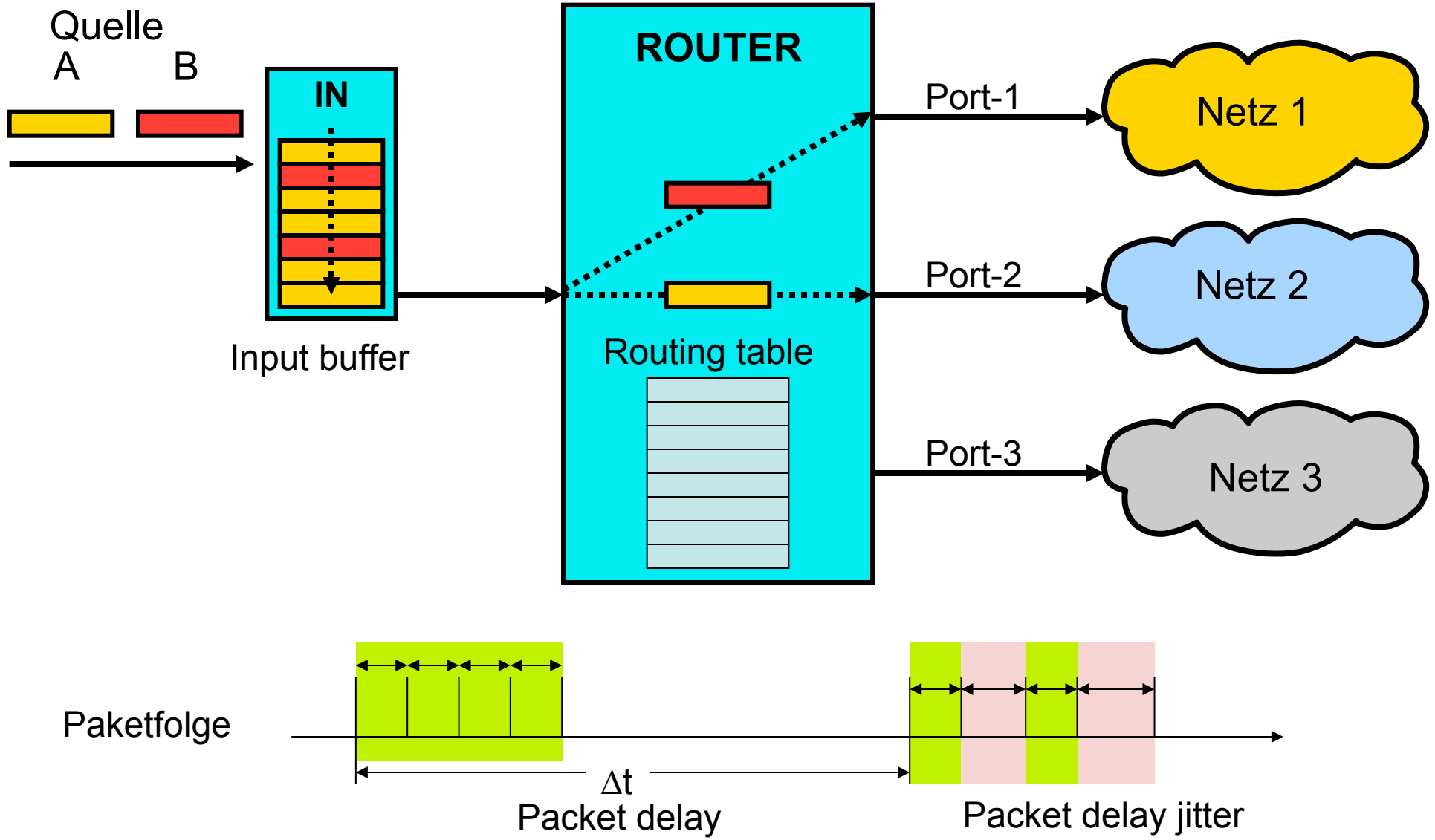
- **Buffering**
- **Encoding**
- **Transmission**
- **Dejitter**
- **Decode**

Paketverzögerung (2)



Ursache der Paketverzögerung im Router:
I/O Operations, Prozessorzeit

Paket Jitter

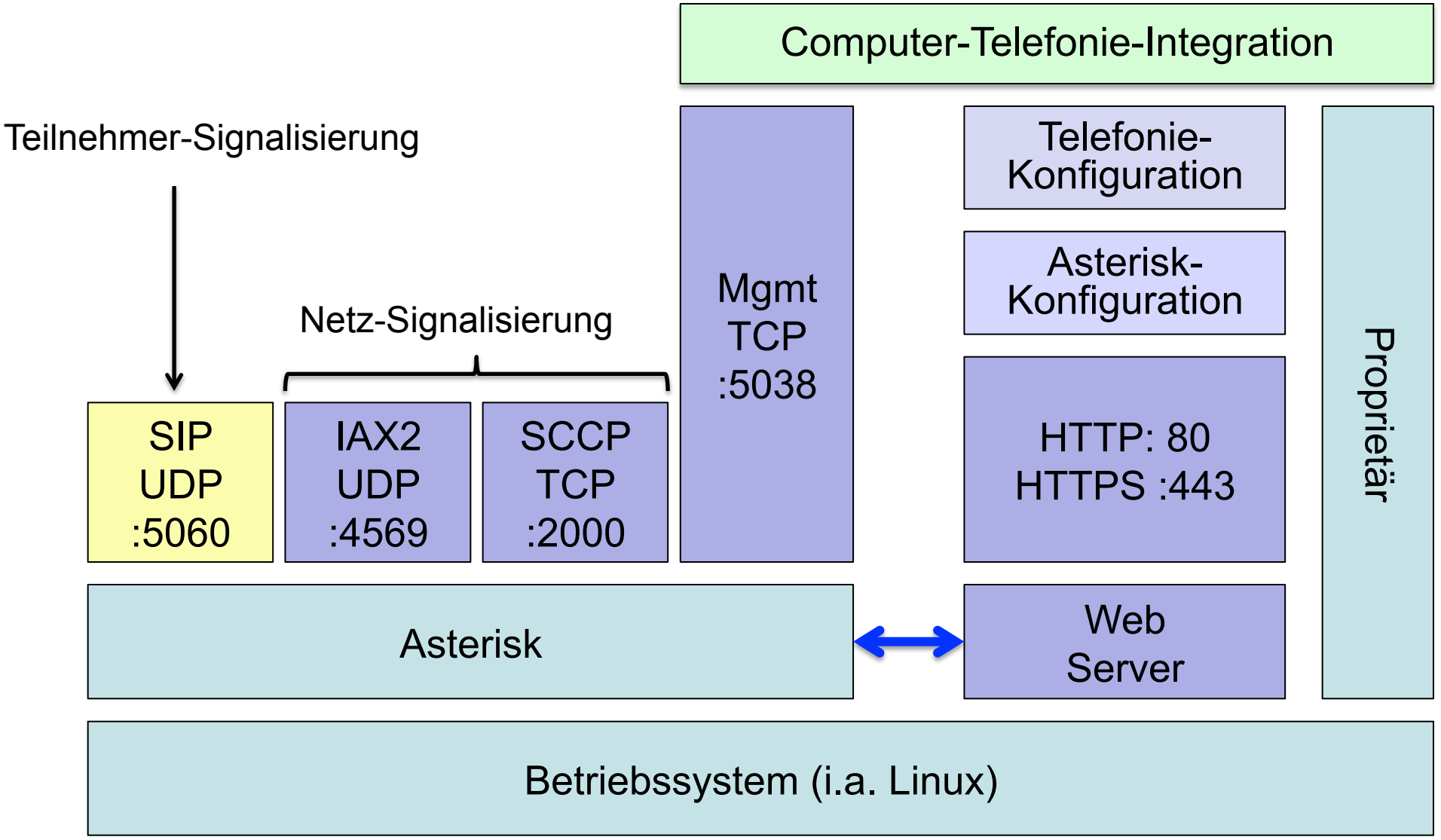


Kursgliederung

DHBW Stuttgart

- Raspberry PI
- Netzwerkdiagnose
  - Kommandos
  - Analyse-Software Wireshark
  - Arbeiten mit Wireshark
- Asterisk – VoIP Einführung
- Asterisk Software
- Asterisk Programmierung

Asterisk Server Aufbau



Asterisk Installation: Software Pakete

DHBW Stuttgart

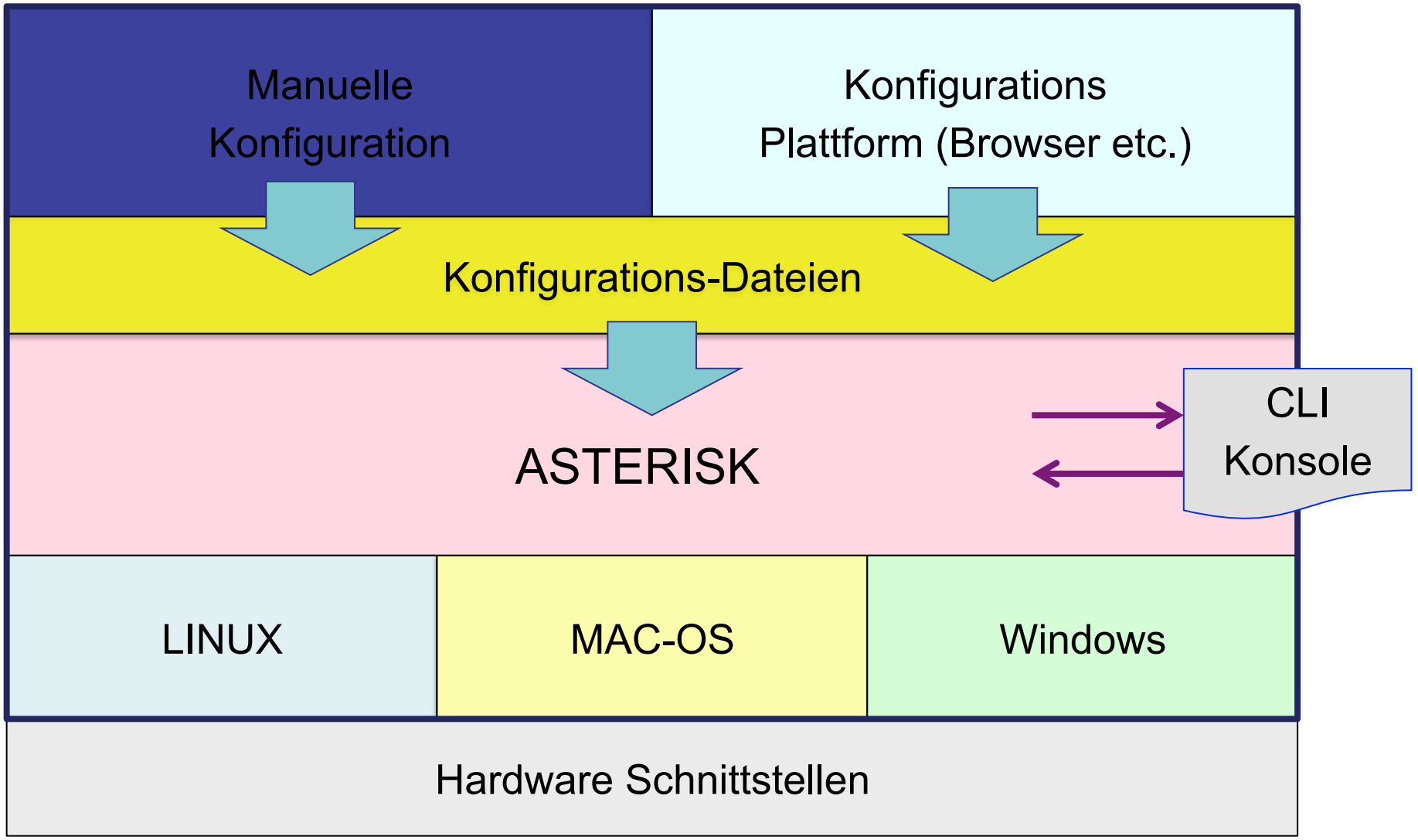
- **Software-Pakete unter Linux:**
  - DEBIAN und UBUNTU : Asterisk mit **APT** installieren
    Benutzerschnittstelle für die Verwaltung von Software-Paketen
  - Red Hat und CentOS : Asterisk mit **YUM** installieren
    Software-Paketmanagement System
  - Software-Komponenten: Basispaket: Asterisk
    DAHDI : Hardware Treiber
- **Windows :**
  - Asterisk Win32 mit PBX-Manager Softwareplattform
  - 3CX Asterisk-basierte Softwarelösungen
  - AsteriskNOW von DIGIUM

Asterisk Installation: Hardware

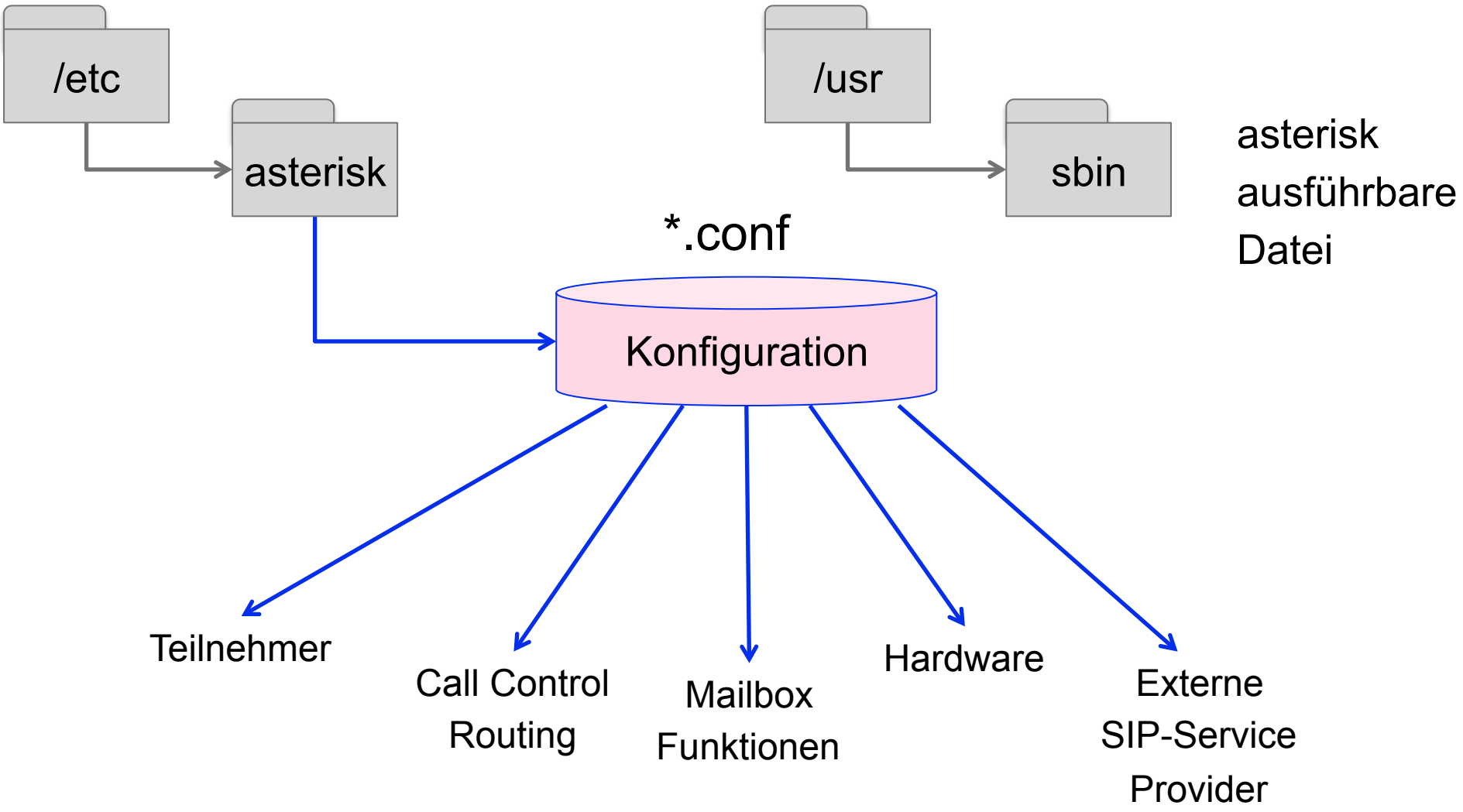
DHBW Stuttgart

- Hardware Dimensionierung
  - Anzahl gleichzeitiger Telefongespräche
  - Anzahl und Art abgehender Telefonleitungen (analog, ISDN (BRA, PRA), Ethernet)
  - Art der Telefongeräte (Analog/ISDN, SIP, H.323,…)
  - Art der Sprachkodierer (G.711, …)
  - Erforderliche Features (Echokompensator, Sprach-Mailbox, Konferenz-Funktionen,…)
  - Anforderungen bezüglich Verfügbarkeit, Erweiterungsfähigkeit
  - IP-Netzanforderungen: Echtzeitfähigkeit, Dienstgüte (QoS)

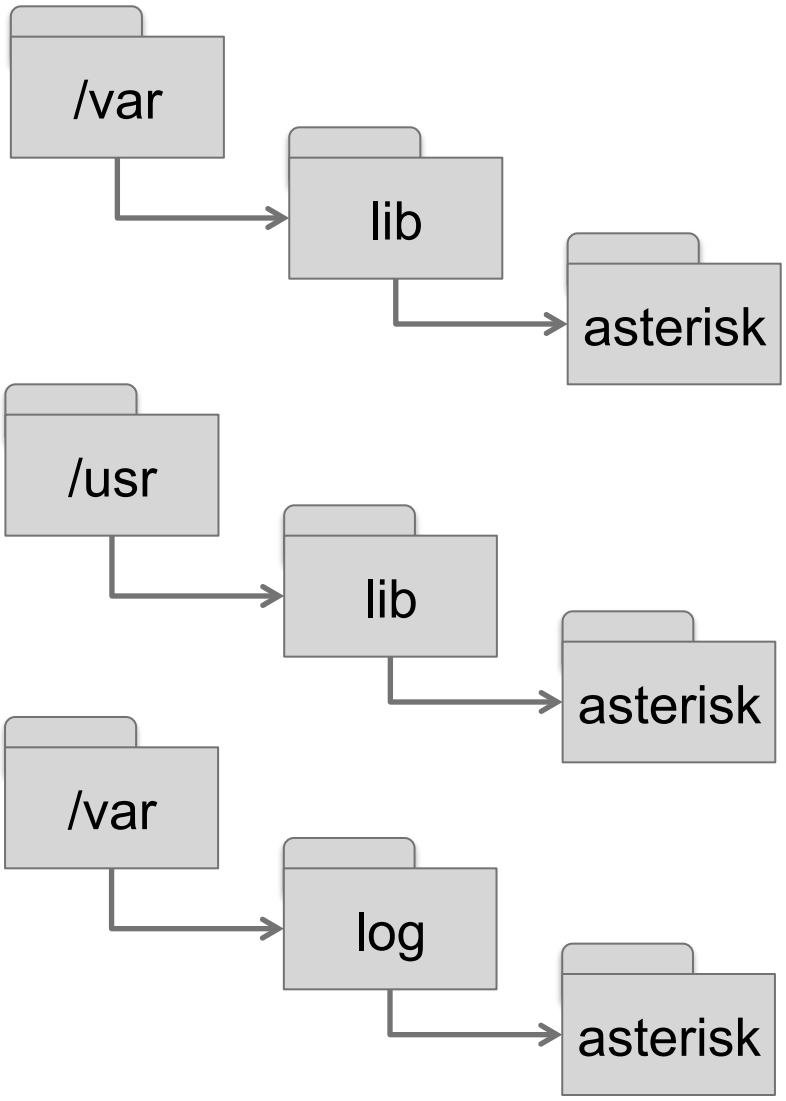
Asterisk Software Struktur



Asterisk Konfigurationsverzeichnis



Weitere Asterisk Verzeichnisse



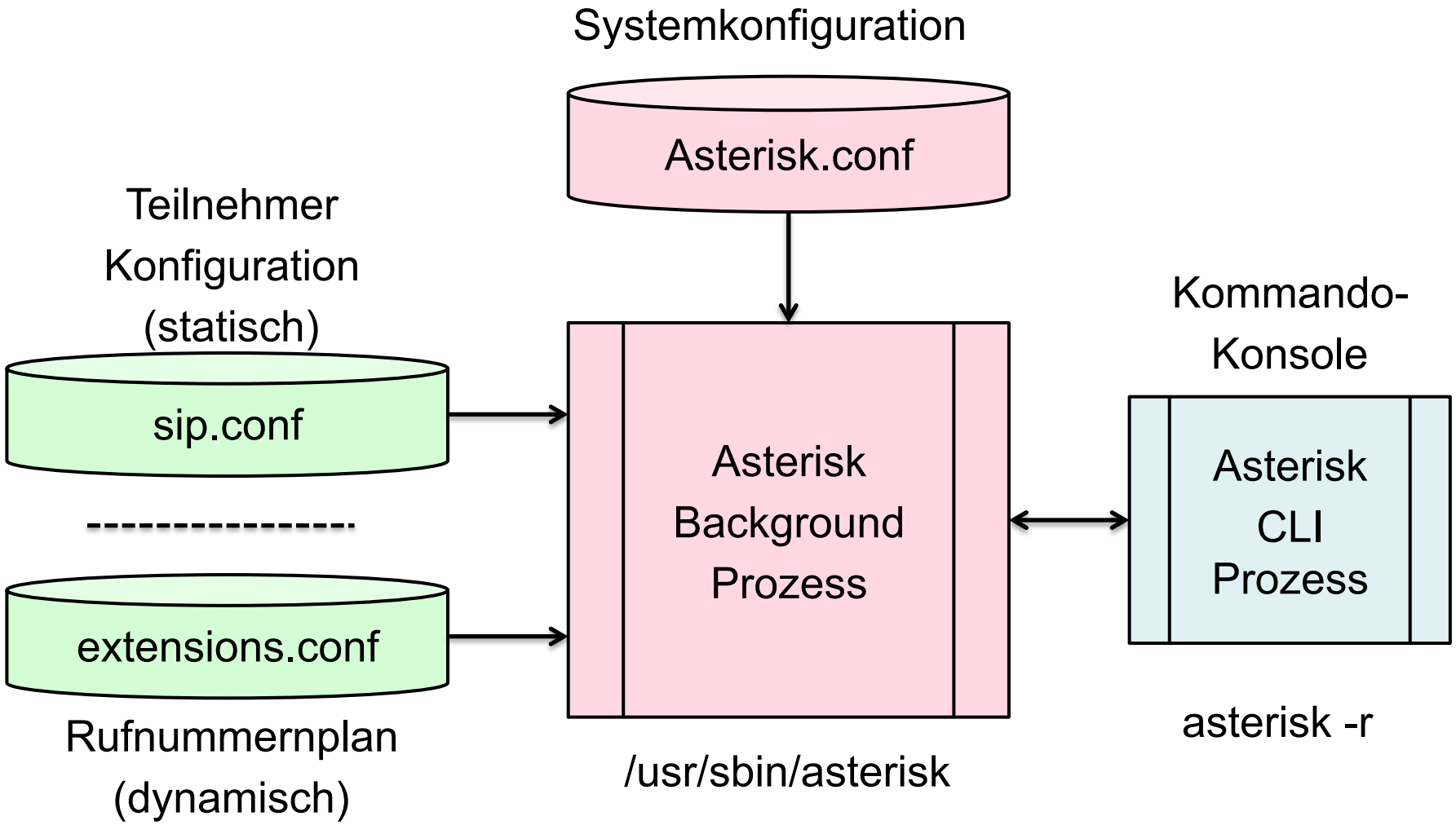
Unterverzeichnisse:

- agi-bin: Script-Dateien
- firmware: Hardware Gerätedateien
- keys: öffentliche Schlüssel
- mohmp3: Haltemusk-Dateien
- sounds: Sprache-Ansagedateien

Asterisk Module
(Anwendungen, Codecs, Formate, Channels

Asterisk Log-Information

Asterisk Prozesse



Asterisk Kommando Konsole - CLI

DHBW Stuttgart

- CLI-Start mit dem Kommando: “asterisk -r”
- CLI-Kommandogruppen
- CLI-Liste der Kommandos: help <gruppenname>

core	DAHDI	dialplan	IAX2
System Kommandos	Hardware Kommandos	Routing Kommandos	Inter-Asterisk Exchange Kommandos
manager	**sip**	**voicemail**	**rtp/rtcp**
Asterisk Management	SIP Parameter	Voicemail Kommandos	rtp/rtcp Monitor

Wichtige Core-Kommandos

DHBW Stuttgart

Asterisk CLI-Kommandos sind Versions-abhängig
Übersicht über die verfügbaren Kommandos mit: “help”

- Core – CLI-Kommandos
  - **core show sysinfo:**
    Anzeige der Prozesse und Speichervolumen
  - **core show settings:**
    System-Auslastung, Verzeichnisse, Subsysteme , Zeitgeber
  - **core show codecs:**
    Anzeige der unterstützten Codecs (Sprache, Bild, Video)
  - **core show setting :**
    Anzeige der SIP-Einstellungen
  - **core restart/stop (now):**
    Asterisk restart/stop

Wichtige CLI SIP-Kommandos

DHBW Stuttgart

- SIP – CLI-Kommandos
  - **sip show peers :**
    Anzeige der SIP-Telefone
  - **sip show registry :**
    Statusanzeige der registrierten Telefone
  - **sip set debug on :**
    Anzeige der SIP – Signalisierung
  - **sip show setting :**
    Anzeige der SIP-Einstellungen
  - **sip show users :**
    Liste der SIP-User

Wichtige CLI Dialplan-Kommandos

DHBW Stuttgart

- Dialplan – CLI-Kommandos
  - **dialplan show :**
    Anzeige des Dialplans
  - **dialplan add/remove extension :**
    Telefon hinzufügen / entfernen
  - **dialplan reload :**
    Dialplan laden – nach einer Veränderung
  - **dialplan show globals:**
    Anzeige der globalen Dialplan-Parameter
  - **dialplan show ?:**
    Liste der Dialplan Anzeigemöglichkeiten

Konfiguration: sip.conf

DHBW Stuttgart

- Definition der einzelnen SIP-Telefone
- Registrierung und Konfiguration der VoIP-Parameter

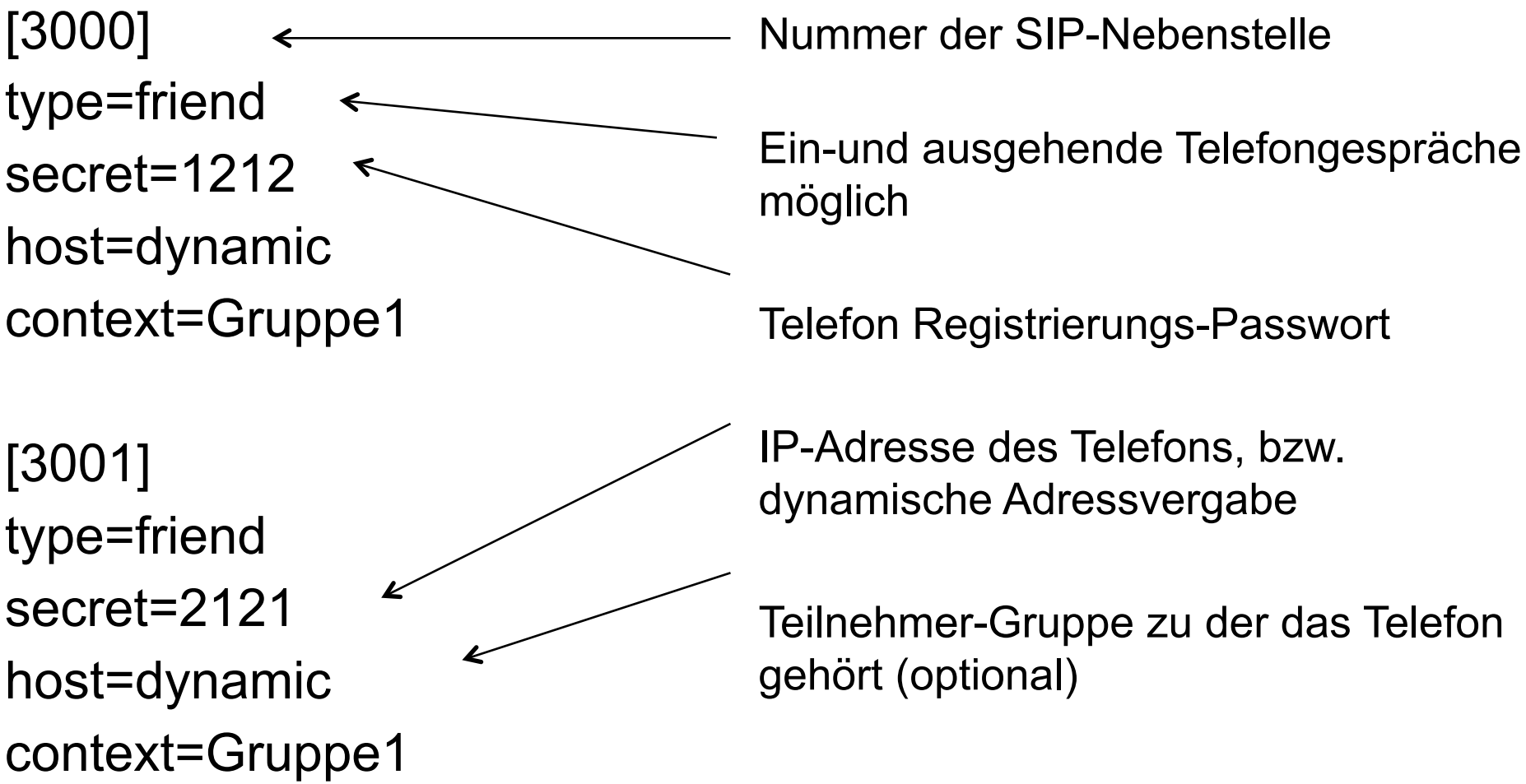
Allgemeiner Teil: [general]

- IP-Adresse und Port-Nummer des Asterisk Servers

Spezieller Teil: [<nr>]

- Beschreibung der SIP-Telefone
  - SIP-Id
  - Caller-Id-Name + Caller-Id-Nummer
  - Dynamische IP-Adresse
  - User, secret: Identifikationsdaten <nr>,
  - Server-Adresse (Domain-Name)
  - NAT-Router vorhanden ?
  - Typ: friend = ein- und ausgehende Verbindungen erlaubt
  - Mailbox-nummer

Beispiel-Konfiguration: sip.conf



SIP Provider Definition

Beispiel-Definition für einen Provider: “provider1” in sip.conf:

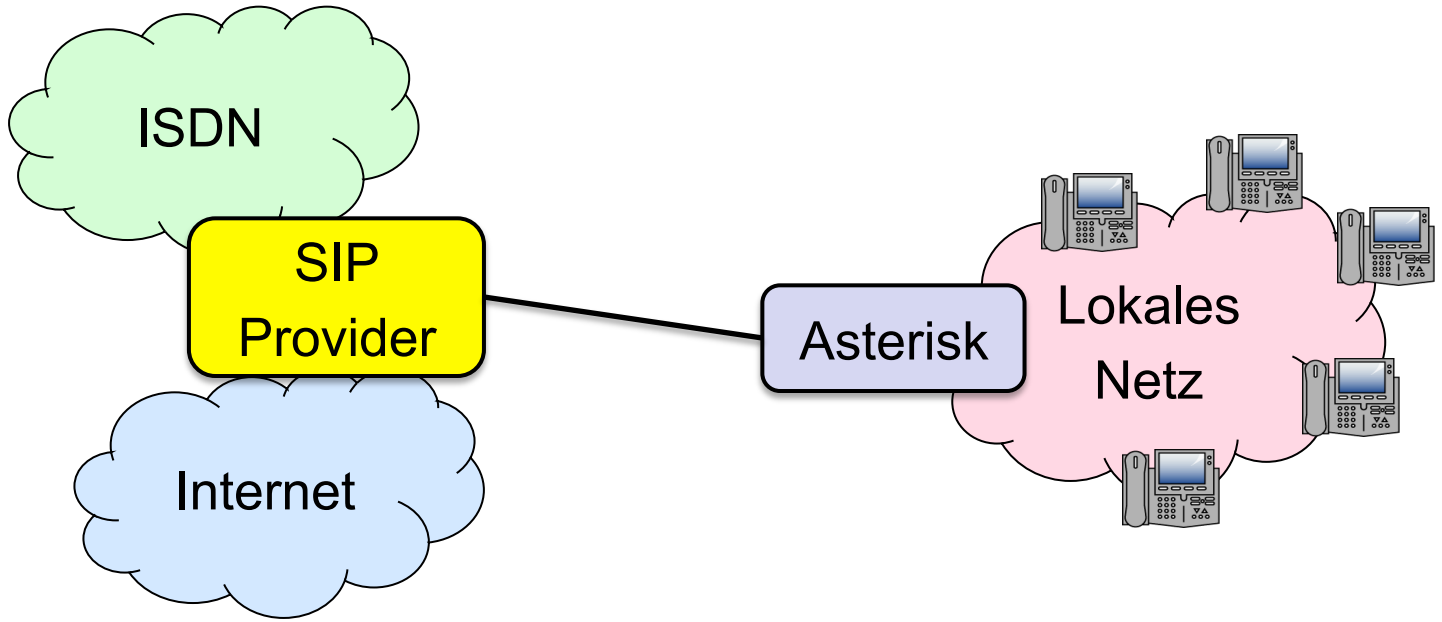
register => 123456:passwort@sip-provider1.de/123456

User Passwort Provider User

```
[provider1]
type=friend
host=sip.provider1.de
fromdomain=sip.provider1.de
username=123456
fromuser=123456
secret=passwort
callbackextension=3000
transport=udp,tcp
nat=yes
```

Externe Registrierung

- Asterisk muss sich bei einem externen SIP-Server registrieren.
- Die Registrierung wird periodisch durchgeführt
- Das entsprechende Kommando lautet:
  - register => *user*[:*passwort*[:*authuser*]]@*host*[:*port*][/*extension*



Konfiguration: Extensions.conf

DHBW Stuttgart

- Enthält den Rufnummern-Plan (Dialplan)
- Dialplan Aufgaben
  - logische Abarbeitung einer Telefon-Transaktion
  - logische Verbindungssteuerung
  - enthält Aktionen und Funktionen
  - ist in unterschiedliche Bereiche untergliedert
  - verwendet eine Script-Sprache: Asterisk Extension Language
  - allgemeines Script-Format:
    exten => extension,priority,command(parameters)

Dialplan Syntax

exten => extension,priority,command(parameters)

- extension: Rufnummer der Nebenstelle oder Name
- priority: Reihenfolge der Aktionen, beginnt mit 1
  keine Numerierungs-Lücken,
  ab nr. 2 kann Platzhalter “n” verwendet werden
- command: Steuerungs-Befehle (Dialplan Applications
- parameters: Befehl-Parameter

Dialplan Beispiele

Beispiele:

```
exten => 123,1, Answer( )
exten => 123,n,Playback(Ansage1)
exten => 123,n,Hangup( )
```

Falls ein Telefon die Nr. 123 wählt wird, so geschieht folgendes:

1. Ruf wird angenommen
2. Ansage1 wird abgespielt
3. Ruf wird beendet (auflegen)

```
exten => 3000,1,Dial(SIP/3000)
```

1. Verbinden mit Nummer 3000

```
exten => 3000,1,Dial(SIP/${EXTEN},60)
exten => 3000,2,Hangup()
```

1. Verbinden mit Nummer (EXTEN = 3000), Timer: 60 sek.
2. Falls nicht erfolgreich:
   Auflegen

```
exten => 123456,1,Dial(SIP/3000)
```

externe SIP-UserId

Eintreffendes Gespräch von 123456 erreicht die Extension 3000 “123456” muss in sip.conf definiert sein.

Mailbox Funktionen

DHBW Stuttgart

- Konfiguration in Datei: voicemail.conf
- Syntax: MailboxNr => Paßwort, Name, E-Mail, Pager, Optionen
  Mailbox-Nummer = Extension
- Beispiele (in voicemail.conf):
  - 3000 => 000,Mailbox3000
    3001 => 111,Mailbox3001
    3002 => 222,Mailbox3002
- Verwendung (in extensions.conf):
  - exten => 3001,1,Dial(SIP/${EXTEN},60)
    exten => 3001,2,VoiceMail(${EXTEN},u)
  - exten => 999,1,VoiceMailMain (${CALLERID(num)},s)

Mailbox der Extension 3000:
Paßwort=000,
Name: Mailbox3000

Mailbox der Extension 3001:
Paßwort=111,
Name: Mailbox3001

Mailbox der Extension 3002:
Paßwort=222,
Name: Mailbox3002

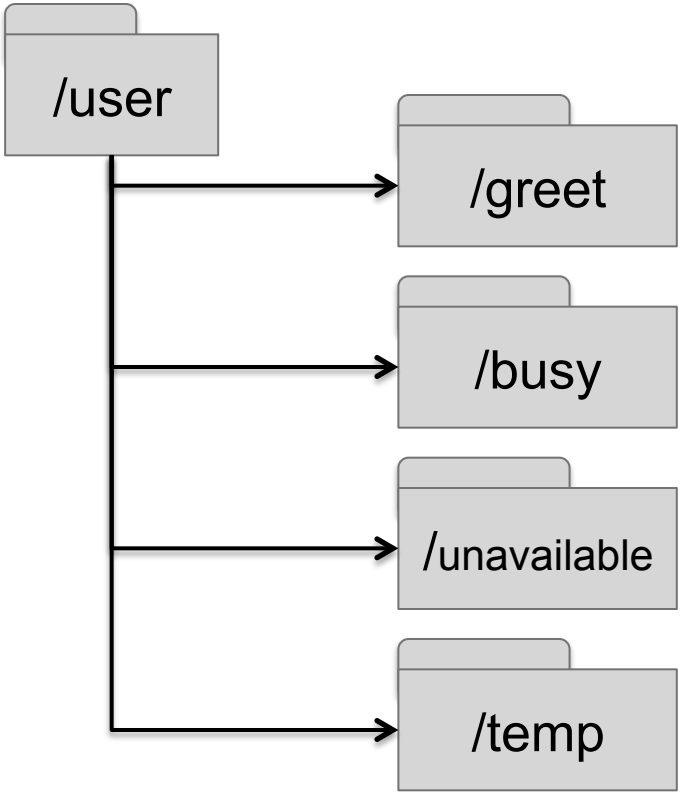
Aufruf der Mailbox 3001:

Abfrage der Mailbox mit der Nummer: 999

Asterisk Mini-VoiceMail (MiniVM)

- MiniVM steht ab Release 1.6 zur Verfügung
- Verzeichnis-Struktur:

/var/spool/asterisk/voicemail/domain



Eigene Benutzeransagen

Ansage: Begrüßung

Ansage: Besetzt, im Gespräch

Ansage: Nicht erreichbar

Ansage: Temporäre Ansage

Dialplan Sonderzeichen

DHBW Stuttgart

Spezielle Zeichen:

_	Beginn einer Zeichenfolge mit Platzhaltern
X	jede Ziffer von 0 – 9
Z	jede Ziffer von 1 – 9
N	jede Ziffer von 2 – 9
[15-7]	Ziffernfolgen: 1 und 5 – 7 = 1, 5, 6, 7
.	Ersatz für einen oder mehrere Buchstaben
!	Ersatz für null oder mehrere Buchstaben

```
[gruppe12]
exten => _12X,1,Answer()
exten => _12X,2,Playback(Ansage1)
exten => _12X,3,Hangup()
```

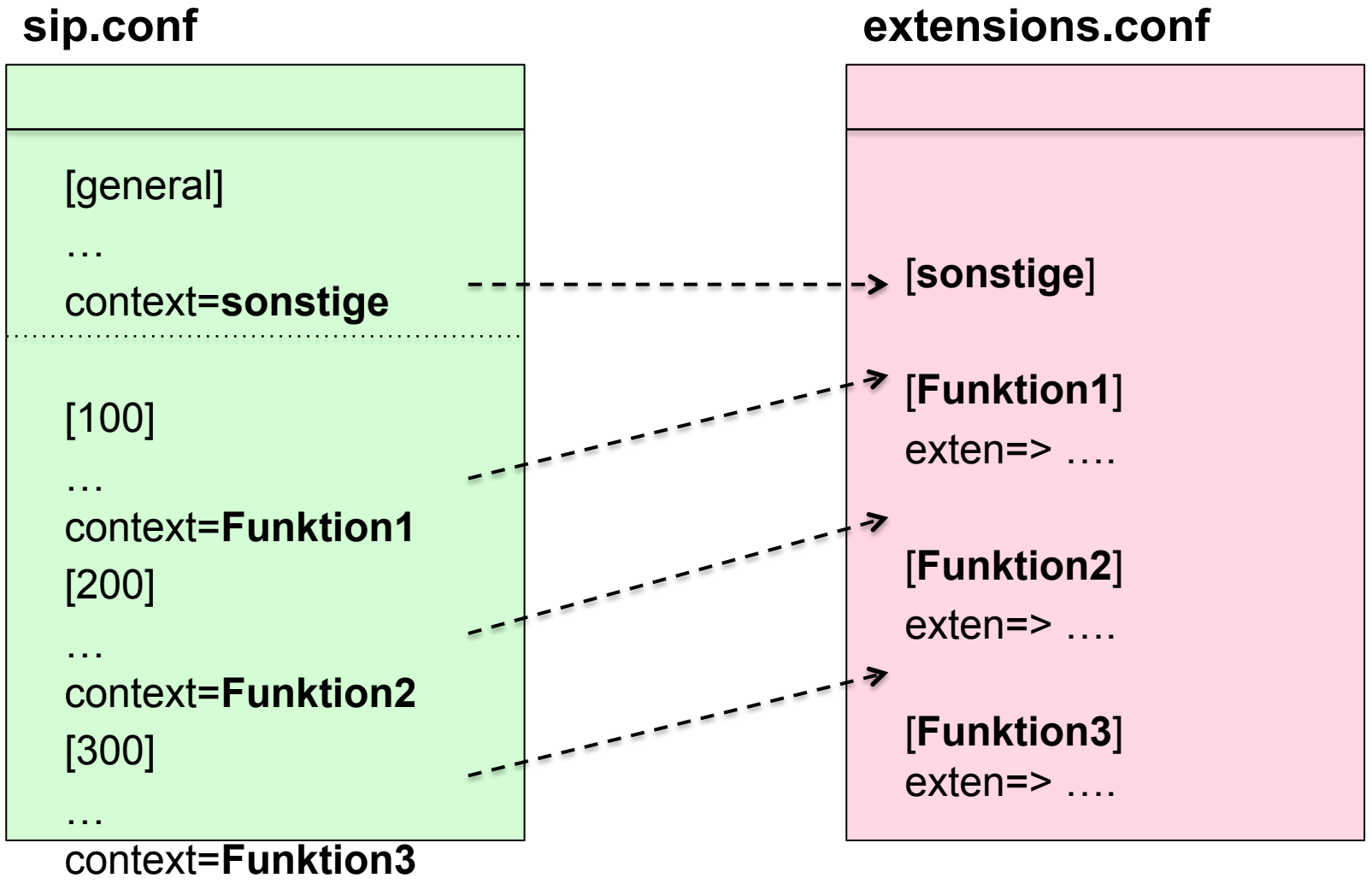
Die Abfolge von Abheben, Einspielen einer Ansage und Auflegen wird hier für die Nebenstellen "120" bis "129" festgelegt.

Dialplan Struktur: Context

DHBW Stuttgart

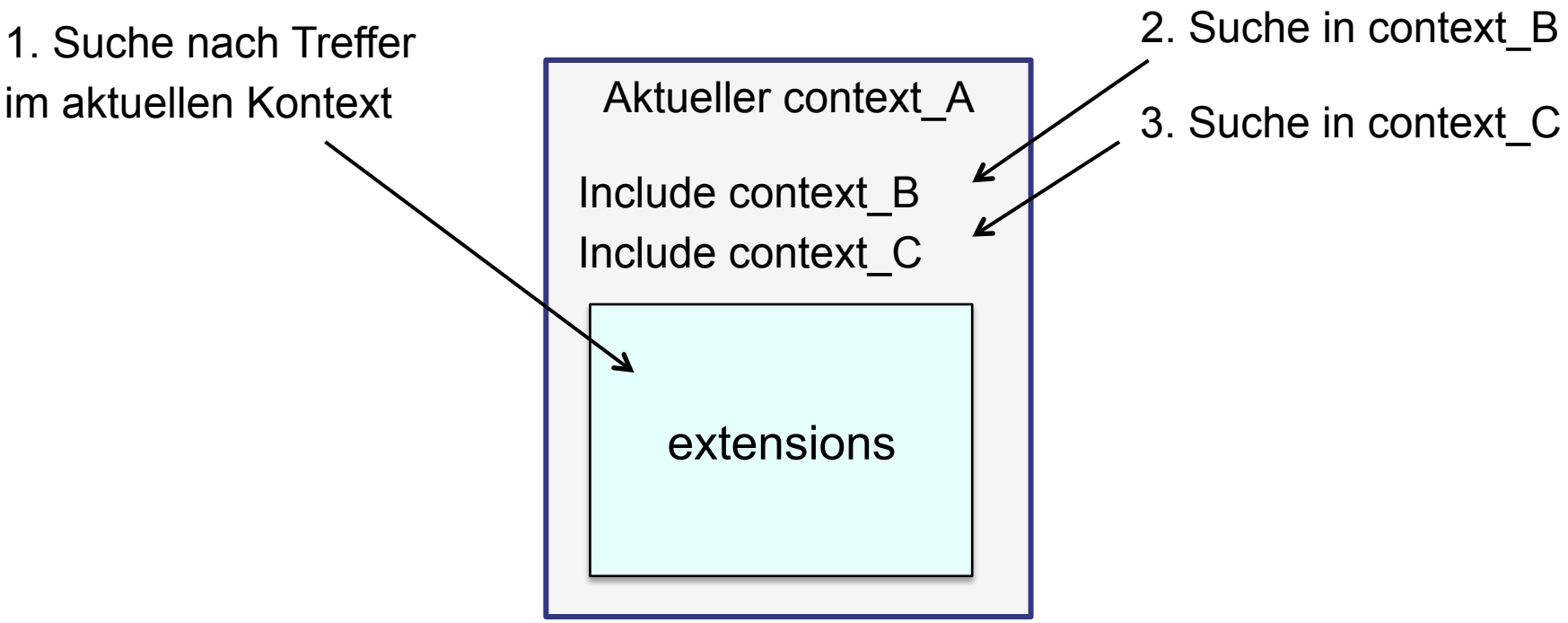
- Kontexte gliedern den Rufnummernplan
  - Syntax: [Kontextname]
  - Vordefinierter Kontext:
    [globals] für die Definition globaler Variablen
    [general] für allgemeine Konfigurationen
- Die Gültigkeit eines Kontextes endet am folgenden Kontext
- SIP-Kontextnamen werden Extensions zugeordnet
- Mittels Kontexten kann die Sicherheit eines Asterisk-Systems erhöht werden.

Context - Verwendung



Dialplan Struktur: Includes

Verwendung: include => ContextName



Bei erfolgreicher Suche wird der Treffer benutzt und der Dialplan weiter abgearbeitet.

Dialplan Effizienz

```
[general]

[intern]
exten => 101,1,Answer()
exten => 101,2,Playback(Text)
exten => 101,3,Hangup()

exten => 102,1,Answer()
exten => 102,2,Playback(Text)
exten => 102,3,Hangup()

exten => 103,1,Answer()
exten => 103,2,Playback(Text)
exten => 103,3,Hangup()
   .......
exten => 109,1,Answer()
exten => 109,2,Playback(Text)
exten => 109,3,Hangup()
```

Durch die Verwendung von “Wildcard” – Zeichen wird der Dialplan im rechten Beispiel wesentlich vereinfacht.

```
[general]

[intern]
exten => _10X,1,Answer()
exten => _10X,2,Playback(Text)
exten => _10X,3,Hangup()
```

Dialplan Variablen

DHBW Stuttgart

- Globale Variablen: [globals]
  - gelten für all Extensions in allen Kontexten
  - Definition zu Beginn der extensions.conf Datei
- Channel Variablen:
  - gelten nur für den aktuellen Call und für den dadurch aktivierten Kanal.
- ${EXTEN} enthält die Wahlziffern
- ${EXTEN:x}
  - Entfernung der ersten x Zeichen
- ${EXTEN:-x}
  - Entfernung der letzten x Zeichen

Wichtige Dialplan Applikationen (1)

DHBW Stuttgart

- Answer( )
  - Akzeptiert einen Verbindungsversuch (Hörer abnehmen)
- Hangup( )
  - Verbindung wird getrennt (Hörer auflegen)
- Playback(Soundfile)
  - Abspielen einer Datei aus dem Verzeichnis: /var/lib/asterisk/sounds
- Wait(SekundenDauer)
  - Pause mit SekundenDauer
- VoiceMail(BoxNummer,Option)
  - Sprachnachricht auf BoxNummer, Option

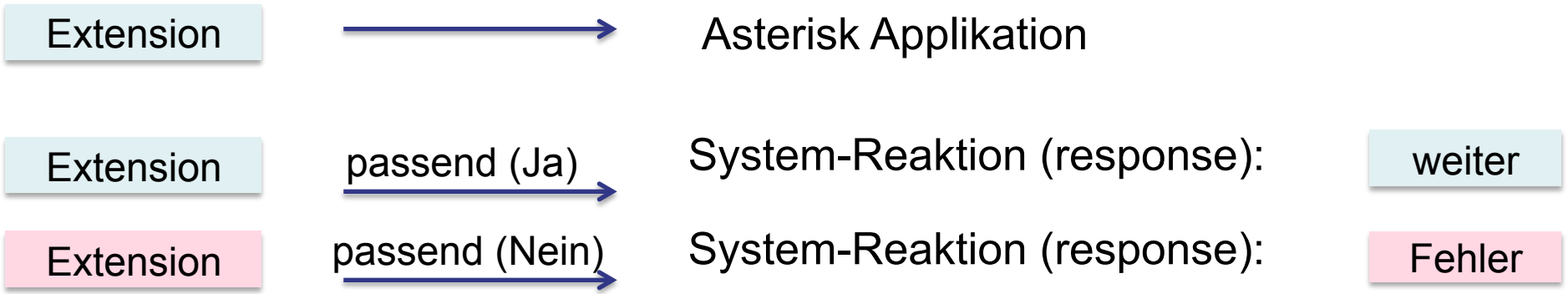
Wichtige Dialplan Applikationen (2)

DHBW Stuttgart

- VoiceMailMain(MailboxNummer, Optionen)
  - Zugang zum Voicemail System
- Dial( )
  - Verbindet Kanäle
- Background()
  - Im Hintergrund eine Sounddatei abspielen
- BackgroundDetect()
  - Background() mit Spracherkennung
- DateTime()
  - Datum/Uhrzeit ansagen

Asterisk Extension Language – AEL

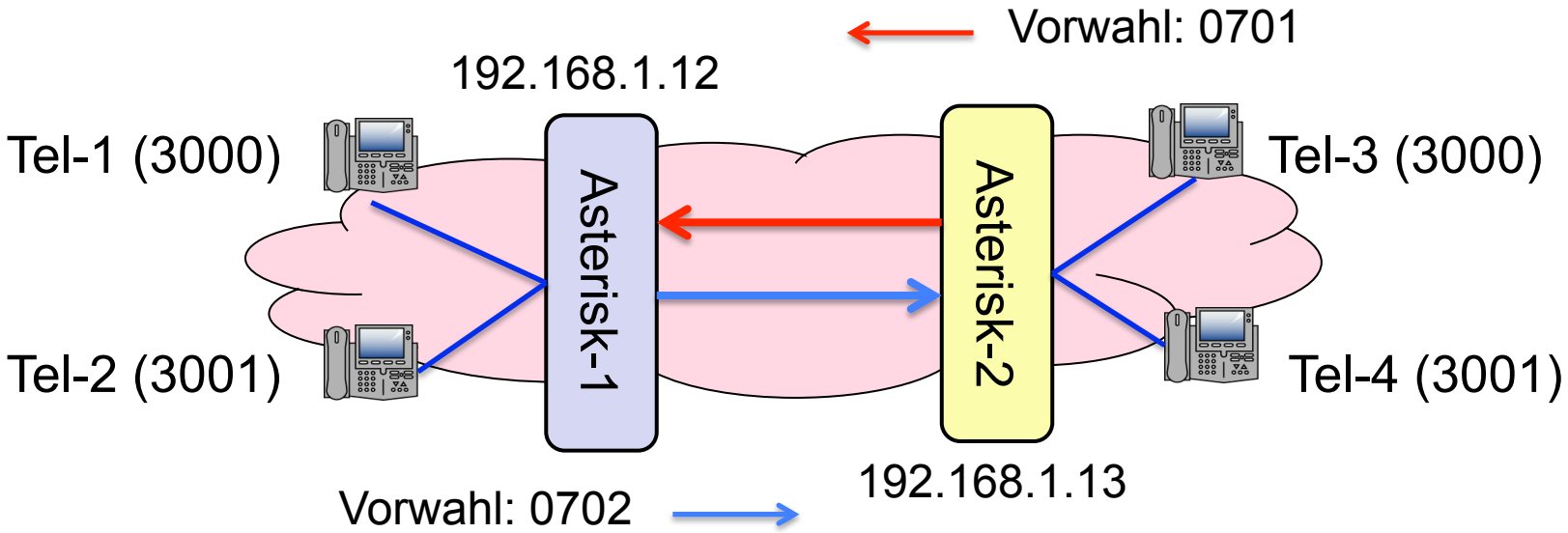
- AEL ist die Beschreibungssprache für den Rufnummernplan
- Extensions können zu Kontexten (Context) gruppiert werden
- Kontexte können geschachtelt sein
- Vordefinierte Extensions (Asterisk Rel. 1.8):
  - s: Start-Extension; Beginn der Kontext-Aktivierung
  - t: Timeout
  - i: ungültige Antwort (invalid response)
- Priorität: Reihenfolge der Abarbeitung



Asterisk – Asterisk Kommunikation : IAX

- IAX (Inter Asterisk Exchange) Protokoll ist das Asterisk-eigene VoIP-Protokoll.
- IAX wird optimal verwendet für die Kommunikation zwischen Asterisk Systemen

Beispiel: Workshop-Konfiguration



Asterisk-1 Programmierung

DHBW Stuttgart

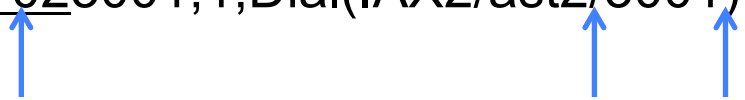
iax.conf

```
[ast2]
type = friend
host = 192.168.1.12
secret = 1234
context = test-telefone
permit = 0.0.0.0/0.0.0.0
```

- [ast2] ← Asterisk-2 Definition
- type = friend ← Kommunikation in beide Richtungen
- host = 192.168.1.12 ← IP-@ von Asterisk-1
- secret = 1234 ← Passwort
- context = test-telefone ← Standard-Kontex für den Dialplan
- permit = 0.0.0.0/0.0.0.0 ← Alle Verbindungen sind zugelassen

dialplan.conf

```
[via-asterisk2]
exten => 07023000,1,Dial(IAX2/ast2/3000)
exten => 07023001,1,Dial(IAX2/ast2/3001)
```



externe Vorwahl externe Verbindung

DHBW Stuttgart

iax.conf

```
[ast1]
type = friend
host = 192.168.1.11
secret = 1234
context = test-telefone
permit = 0.0.0.0/0.0.0.0
```

Asterisk-1 Definition
Kommunikation in beide Richtungen
IP-@ von Asterisk-2
Passwort
Standard-Kontex für den Dialplan
Alle Verbindungen sind zugelassen

dialplan.conf

```
[via-asterisk1]
exten => 07013000,1,Dial(IAX2/ast1/3000)
exten => 07013001,1,Dial(IAX2/ast1/3001)
```

externe Vorwahl

externe Verbindung

Asterisk – zu – Asterisk Verbindung

DHBW Stuttgart

Asterisk-1 (Ast1)

Asterisk-2 (Ast2)

3000 →  → 3000 

extensions.conf

```
exten => 3000,1,Dial(IAX2/Ast1-user:
Passwort@asterisk2.example.com/
${EXTEN}@IAX_incoming)
exten => 300,2,Hangup
```

Ast1-user :
Kontext, in dem Tln. 3000 definiert ist

asterisk2.example.com :
ist die IP-Adresse von Ast2
IAX_incoming :
verweist auf den Ast2-Kontext, für die Rufnummer 3000

iax.conf

```
[Ast1-user]
type=user
secret=Passwort
context=IAX_incoming
```

extensions.conf

```
[IAX_incoming]
exten => 3000,1,Dial(SIP/3000,30,Ttm)
exten => 3000,n,Hangup()
```

Dialplan-Beispiel: Rufumleitung

DHBW Stuttgart

- Aktivierung der Rufumleitung durch die Vorwahl: 99 + Zielnummer

```
exten => _99X.,1,Answer()
exten => _99X.,n,Set(DB(CF/${CALLERID(num)})=${EXTEN:2})
exten => _99X.,n,SayDigits(${EXTEN:2})
exten => _99X.,n,NoOp(Weiterleitung fuer ${CALLERID(num)} auf ${EXTEN:2} aktiviert.)
exten => _99X.,n,Hangup()
```

➢ Deaktivierung der Rufumleitung durch Wahlziffern: 99

```
exten => 99,1,Answer()
exten => 99,n,DBdel(CF/${CALLERID(num)})
exten => 99,n,Playback(auth-thankyou)
exten => 99,n,NoOp(Weiterleitung fuer ${CALLERID(num)} deaktiviert.)
exten => 99,n,Hangup(
```

Rufumleitung im Dialplan

exten => _X.,1,NoOp(Anruf von ${CALLERID(num)} fuer ${EXTEN})
; Ausgabenachricht: CALLERID(num) = Nummer des Anrufers
: ${EXTEN} = Ziel-Rufnummer

exten => _X.,n,GotoIf($[foo${DB(CF/${EXTEN})} != foo]?normal:forward)
; Abfrage : DB(CF/${EXTEN}) CF-Eintrag in der Datenbank ?
; Eintrag vorhanden : 0 -> Sprungziel = normal ; 1 -> Sprungziel=forward

exten => _X.,n(normal),Dial(SIP/${EXTEN})
; Wahlvorgang : normale Verbindung

exten => _X.,n(forward),NoOp(Anruf fuer ${EXTEN} wird verbunden zu ${DB(CF/${EXTEN})})
; Wahlvorgang

exten => _X.,n,Dial(local/${DB(CF/${EXTEN})})

Kursgliederung

DHBW Stuttgart

- Raspberry PI
- Netzwerkdiagnose
  - Kommandos
  - Analyse-Software Wireshark
  - Arbeiten mit Wireshark
- Asterisk – VoIP Einführung
- Asterisk Software
- Asterisk Programmierung

AEL2 Einführung

DHBW Stuttgart

- AEL2 Aktivierung durch Modul “pbx_ael.so”
- AEL2:
  - Programmiersprache zur Dialplan-Programmierung
  - AEL2 Syntaxdefinition im BNF-Format
  - Datei-Erweiterung von AEL2-Dialplan: .ael2
  - Datei-Erweiterung von Standard-Dialplan: .conf
- Standard-Dialplan Programmierung: .conf
- AELPARSE als Übersetzer von .ael2 -> .conf
  - AELPARSE als Testprogramm für AEL2-Dateien

AEL2 Syntax (1)

Kommentar: // Text bis zum Zeilenende

Kontext:

```
Context default {  // Kontextname in der selben Zeile wie „context“
.....              // Klammer „{„ in der selben Zeile wie Block-Name
}
```

Extensions:

```
context default {
07231 => Playback(audio-1);  // Wiedergabe-Funktion
 8000 => {                   // Liste abarbeiten
NoOp(Text1);                 // NoOp = CLI-Ausgabe: „Text1“
NoOp(Text2);                 //                     „Text2“
NoOp(Text3);                 //                     „Text3“
};                           // Ende der Liste
 _5XXX => NoOp(Ziffernmuster);     //                „Ziffernmuster“
  };
```

AEL2 Dialplan-Beispiele

```
123 => {
        Answer()
        Playback(Ansage)
        Dial(SIP/${EXTEN},20)
        Voicemail(${EXTEN},u)
}
```

Ist gleichbedeutend (in conf-Schreibweise) mit:

```
exten => 123,1,Answer()
same => n,Playback(Ansage)
same => n,Dial(SIP/${EXTEN},20)

same => n,Voicemail(${EXTEN},u)
```

kopierfähig für jede Nummer

AEL2 Syntax (2)

AEL2 Variablen-Definition

```
globals {                                  // Globale Variablen in einem Block
CONSOLE=Console/dsp;                       // Wertzweisung: CONSOLE
TRUNK=Zap/g2;                              // Wertzweisung: TRUNK
};
context default{                           // Variablendefinition in der extension
555 => {                                   // entspricht dem Set – Befehl
x=5;                                       // Variable x: Wert = 5
y=nix;                                     // Variable y: Wert = “nix”
div=10/2;                                  // Variable div = 5
NoOp(x is ${x} und y is ${y} !);           // CLI-Ausgabe: “x=5 und y=nix
};
};
```

AEL2 Syntax (3)

DHBW Stuttgart

Bedingungen: if … else

```
context conditional {                          // Kontext = “conditional”
_8XXX => {                                     // 1. Ziffer = 8
Dial(SIP/${EXTEN});                            // Wähle: Rufnummer
if ("${DIALSTATUS}" = "BUSY") {                // Falls besetzt:
Voicemail(${EXTEN}|b);                         // Ansage: besetzt
} else {                                       // else-Zweig in Klammern
Voicemail(${EXTEN}|n);                         // Ansage: nicht anwesend
 }
};
```

Schleifensteuerung : Break, Continue

```
context loops {
1 => {                                          // Extension = 1
for (x=0; ${x} < 3; x=${x} + 1) {               // Schleifenbedingung
Verbose(x is ${x} !);
if( ${x} == 2 && ${y} == 17) break;             // Abbruchbedingung (if)
if(${x} == 2 && ${y} == 16) continue } }        // weiter
2 => {                                          // Extension = 2
z=10
y=10; while (${y} >= 0) {                       // Schleifenbedingung
Verbose(y is ${y} !);
z=${z} + 1                          // Increment Abbruchbedingung
if (${z}>20) break;                 // Abbruchbedingung (if)
y=${y}-1; }                         // Abbruchbedingung (while)
}
}
```

Anwendungsbeispiel : Gesprächszähler

DHBW Stuttgart

```
[globals]
; Zählerdefinition
ZAEHLEN=1
; sollen die Extensions der laufenden Server-Instanz gezählt werden? (ja = 1)
ANZAHL=NULL      ; Startwert
GESPRAECHE=0 ; Startwert
; zaehlen und weiter sind Sprungmarken
exten => _300[0-3],1,GotoIf($[${ZAEHLEN} = 1}]?zaehlen:weiter)
exten => _300[0-3],n(zaehlen),Set(GLOBAL(GESPRAECHE)=$[${GESPRAECHE}+1])
exten => _300[0-3],n(weiter),Dial(SIP/${EXTEN},10,tT)
; 10 Sek. timer. tT aktiviert Vermitteln & Parken fuer beide Seiten
exten => _300[0-3],n,VoiceMail(${EXTEN},u)
; Mailbox falls Verbindung nicht zustande kommt
```

Macro (1)

- Macro ist ein “Funktionsaufruf”, die von Extensions verwendet werden
- Macro Einsatz:
  - für wiederholt auftretende Ereignisse
  - für zentralisierte Änderung am Dialplan
- Syntax: Makro-Definition: im Kontext: [**Macro-**Macroname] extensions ….
- Syntax: Macro-Aufruf: Macroname(Par1,.., Par-n)
- Vorgegebene Variablen:
  - ${MACRO_CONTEXT}, ${MACRO_EXTEN}, ${MACRO_PRIORITY}, $ARG1}, …, $ARGn},

Macro (2)

AEL2 Macro definition

```
Macro norm-exten( ext , dev ) {  // 2 Parameter: extension, device
Dial(${dev}/${ext},20);          // z.B. SIP/123

switch(${DIALSTATUS}) {          // Abfrage von DIALSTATUS
case BUSY:                       // falls besetzt:
Voicemail(${ext},b);             // BUSY-Ansagetext
break;                           // Switch verlassen

default:                         // Switch-Ausgang: sonst
Voicemail(${ext},u);             //  Nicht-Anwesend-Ansagetext

}

}
```

Asterisk Variablen

DHBW Stuttgart

Einige Built-in Variablen:

- ${CALLERID(num)} Anrufernummer
- ${CONTEXT} aktueller Kontext
- ${EXTEN} Rufnummer
- ${CHANNEL} Channelname
- ${PRIORITY} aktuelle Dialplan-Priorität
- ${HANGUPCAUSE} Auslösegrund

Eigene Variablen definieren:

- same => n,Set(Variable1=10)
  same => n,Set(Variable2=5)
  same => n,Set(Variable3="Ergebnis = ")

Verwenden:

- same => n,NoOp(${Variable3}${Variable1}/${Variable2})

Spezielle Extensions

- s-Extension: wird verwendet, wenn das Ziel nicht bekannt ist.
  s-Extensions werden z.B. in Macros verwendet
- i-Extension: wird für eine ungültiges (invalid) Ziel verwendet
- t-Extension: wird für ein Timeout verwendet
- h-Extension: markiert die Beendigung eines Gesprächs
- o-Extension: Operator Extension durch Eingabe von Null (0) im Voicemailmenü
- a-Extension: Abbruch durch “*” – Eingabe im Voicemailbox Menü

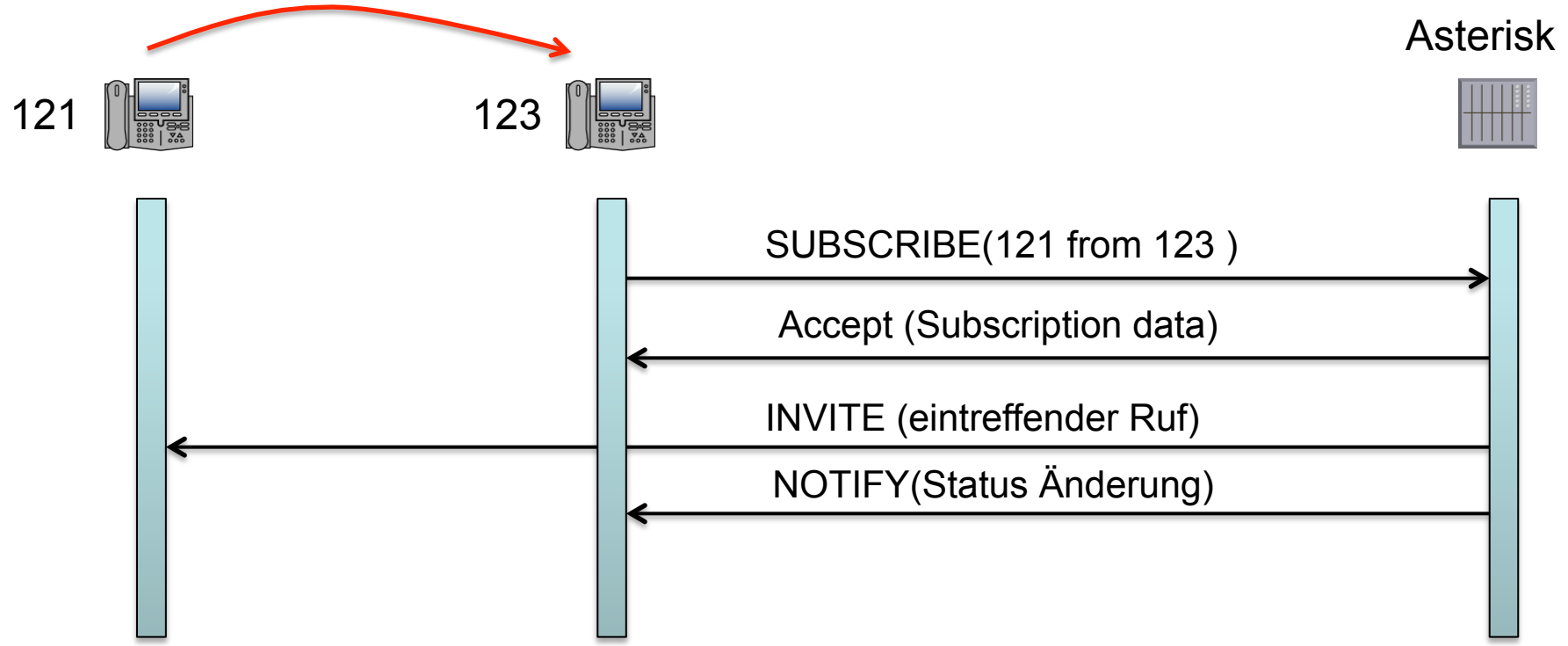
Leistungsmerkmal: Call Pick-Up

- **Vorbereitung**:
  - Programmierung des Telefons-B zur Übernahme der Gespräche des Telefons-A
  - Meldung an das Telefon-A (SUBSCRIBE)
  - Aktivierung des Leistungsmerkmals
- **Durchführung des Leistungsmerkmals**:
  - Eintreffender Ruf am Telefon-A: Telefon-A klingelt
  - Information an Telefon-B (NOTIFY) und Anzeige am Telefon-B
  - Übernahme durch Telefon-B (Funktionstaste oder Zeichenfolge)

SIP-Prozedur: Call Pick-Up

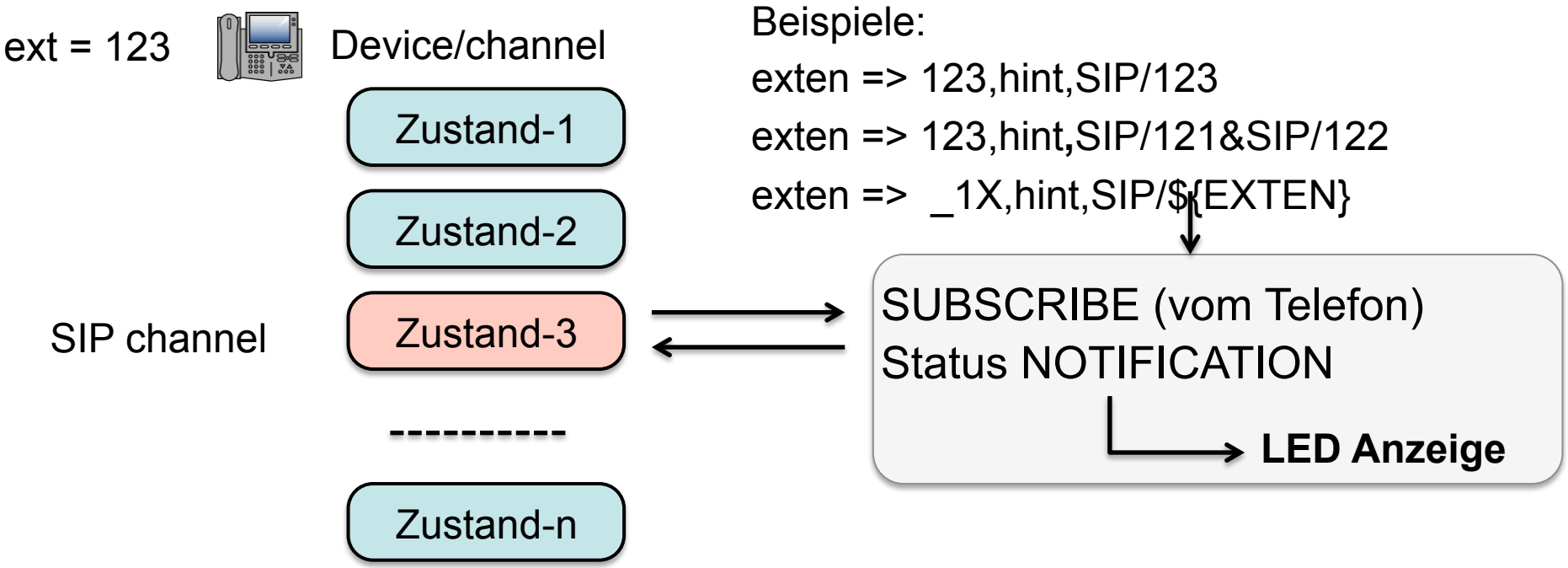
DHBW Stuttgart

SIP Beispiel : Call Pick-Up Nr. 121 durch Nr. 123



Telefon-Ansteuerung - BLF

- **BLF** : Besetzt-Anzeige (Busy Lamp Field) durch die Telefonanlage
- **Hint** – Priorität verknüpft:
  - Extension = Folge von Funktionen/Anwendungen mit dem Channel (Gerät, Technologie) und dessen Zustand



Umsetzung in sip.conf

Anzeigesteuerung Telefon-123:

```
[general]
allowsubscribe = yes        /* SUBSCRIBE Prozedur erlauben */
notifyringing = yes         /* NOTIFY bei eintreffendem Ruf */
notifyhold = yes
limitonpeers = yes
```

Context – Ergänzungen:

```
[123]
……
Subscribecontext=interne-verbindungen   /* Teilnehmer-Kontext */
call-limit=10                           /* Gesprächszähler */
callgroup=2                             /* Rechteverwaltung */
pickupgroup=2                           /* Pickup-Gruppe */

…….
```

Dialplan Programmierung

DHBW Stuttgart

[**interne-verbindungen**]

exten => _2X,hint,SIP/${EXTEN}

exten => _2X,1,Dial(SIP/${EXTEN},30)

exten => _2X,n,VoiceMail(${EXTEN},u)

; Gesprächsübernahme mit *8+Nr
; z.B. mit *8121 wird 121 herangeholt

exten => _*8X.,1,Set(nst=${EXTEN:2})

exten => _*8X.,1,Pickup(${nst}@interne-verbindungen)

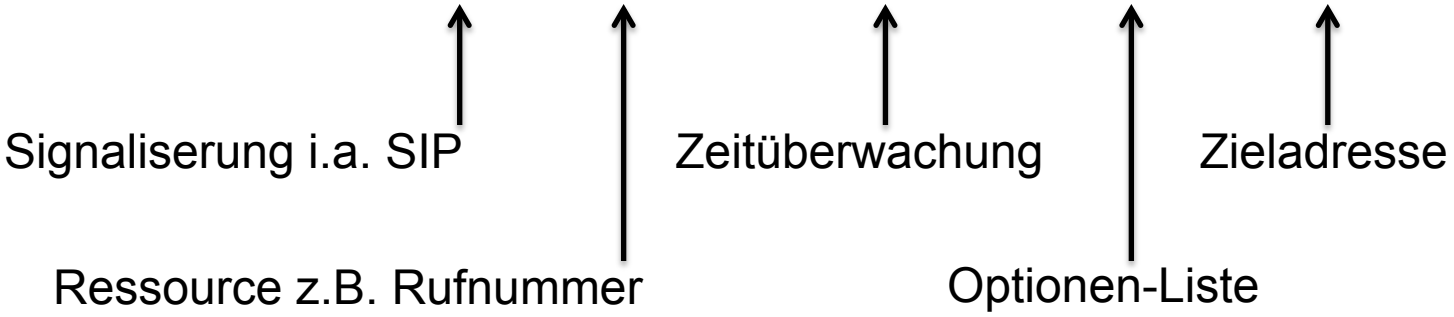
```
context interne-verbindungen {
    hint(SIP/${EXTEN})  _2X => {
     Dial(SIP/${EXTEN},30);
    VoiceMail(${EXTEN},u);
 }
}
// Gesprächsübernahme mit *8+Nr //
_*8X. => {
Set(nst=${EXTEN:2});
Pickup(${nst}@interne-benutzer);
}
}
```

Ergebnis

- Meldung eines Statuswechsel des überwachten Telefons an das überwachende Telefon
- LED-Steuerung (Telefon-Funktion):
  - Keine Aktivität: LED aus
  - Blinkende-LED bei eintreffenden Ruf
  - Rufannahme mit *8 + Nummer des überwachten Telefons
  - Dauer-LED, falls das überwachte Telefon ein aktives Gespräch führt
- Konsolen-Meldungen bei Status-Wechsel

Dial-Funktion und Call Transfer

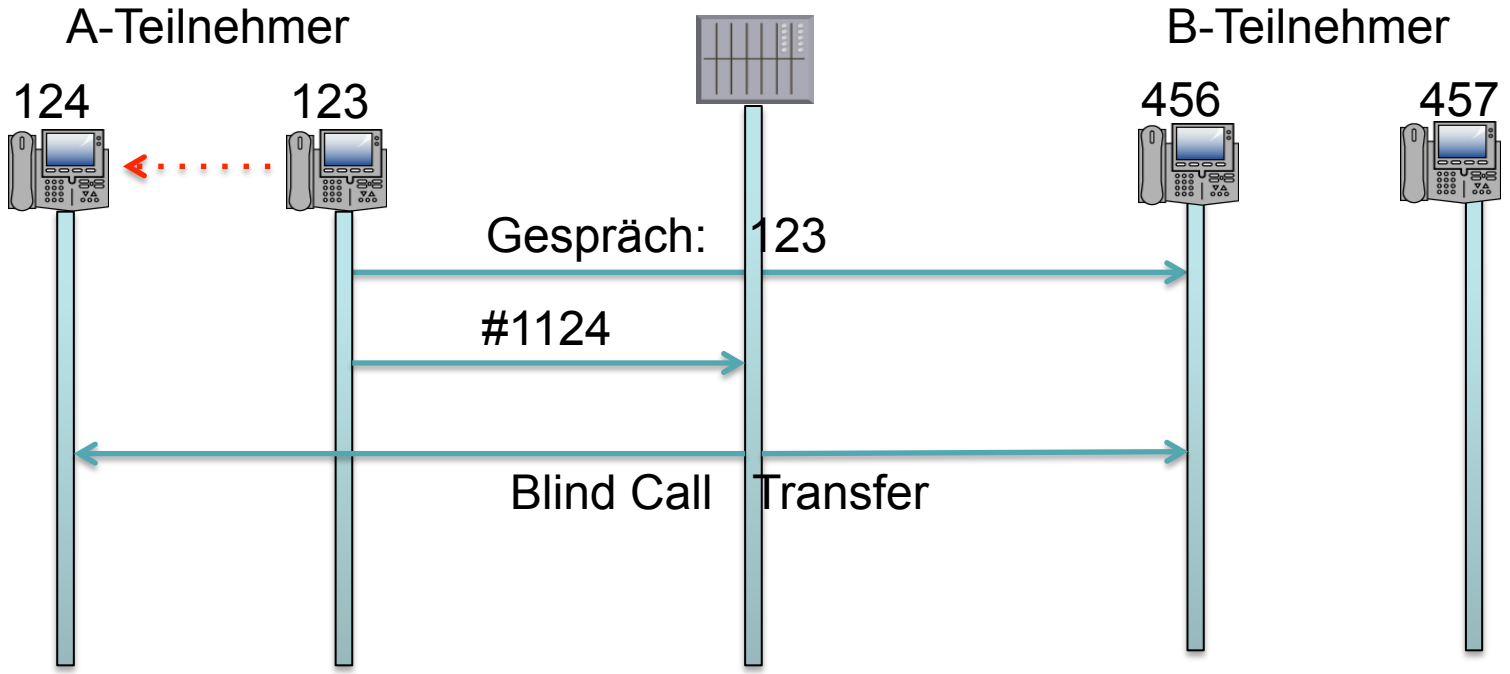
Dial Syntax: **Dial(*Tech*/*Resource*,*Timeout*,*Optionen*,*URL*)**



Wichtige Dial – Optionen:

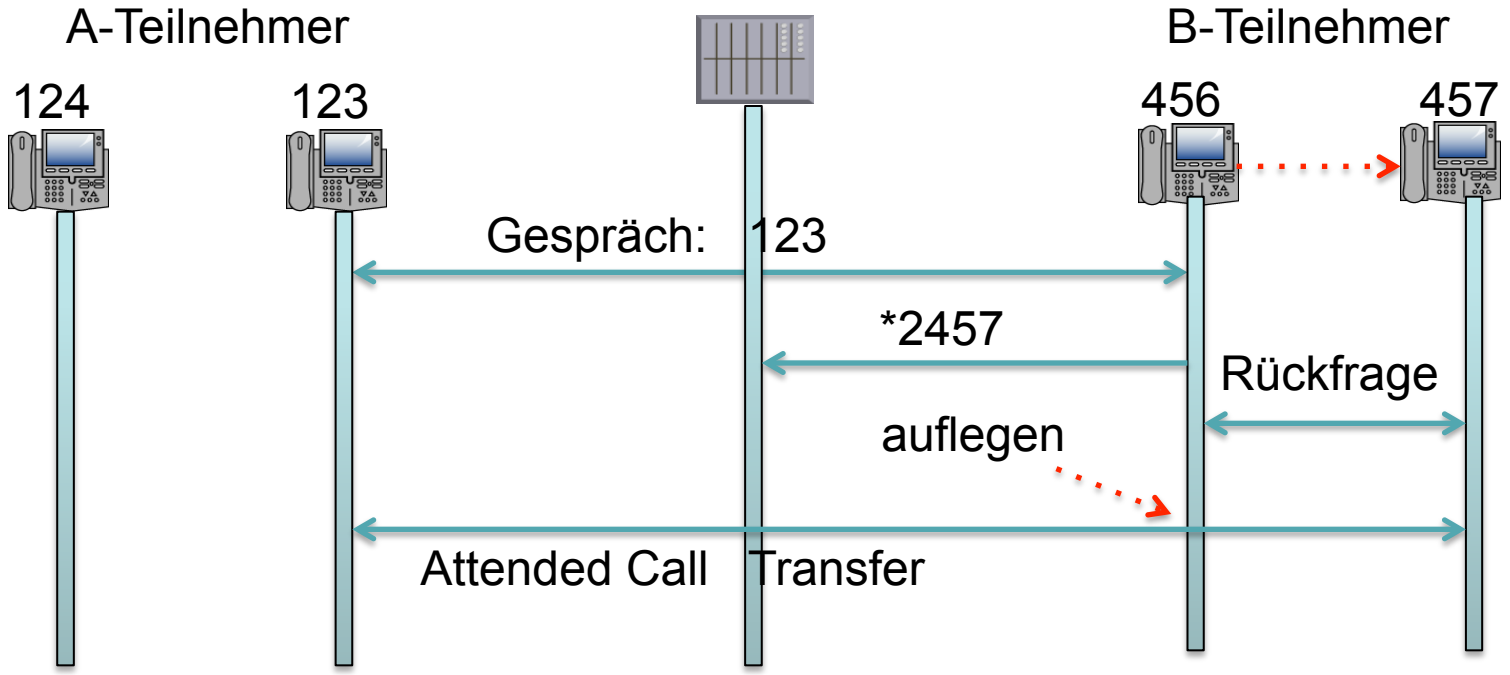
- **t/T:** Transfer durch den angerufenen/rufenden Teilnehmer durch drücken der #-Taste ermöglicht
- w/W: Aufnahme des Gesprächs durch den angerufenen/rufenden Teilnehmer
- M(x[arg]): Ausführen des Makros x[arg] bei der Rufannahme
- L(x): Begrenzt die Gesprächsdauer

Blind Transfer – ohne Rückfrage



- exten => same,1,Dial(SIP/${EXTEN},tT)
  Call Transfer für rufenden/gerufenen Teilnehmer erlaubt
- Standard-Transfer-Kommando: #1 + Zielrufnummer

Blind Transfer – mit Rückfrage



- exten => same,1,Dial(SIP/${EXTEN},tT)
  Call Transfer für rufenden/gerufenen Teilnehmer erlaubt
- Standard-Transfer-Kommando: #1 + Zielrufnummer

Dialplan : Labels und Sprünge

DHBW Stuttgart

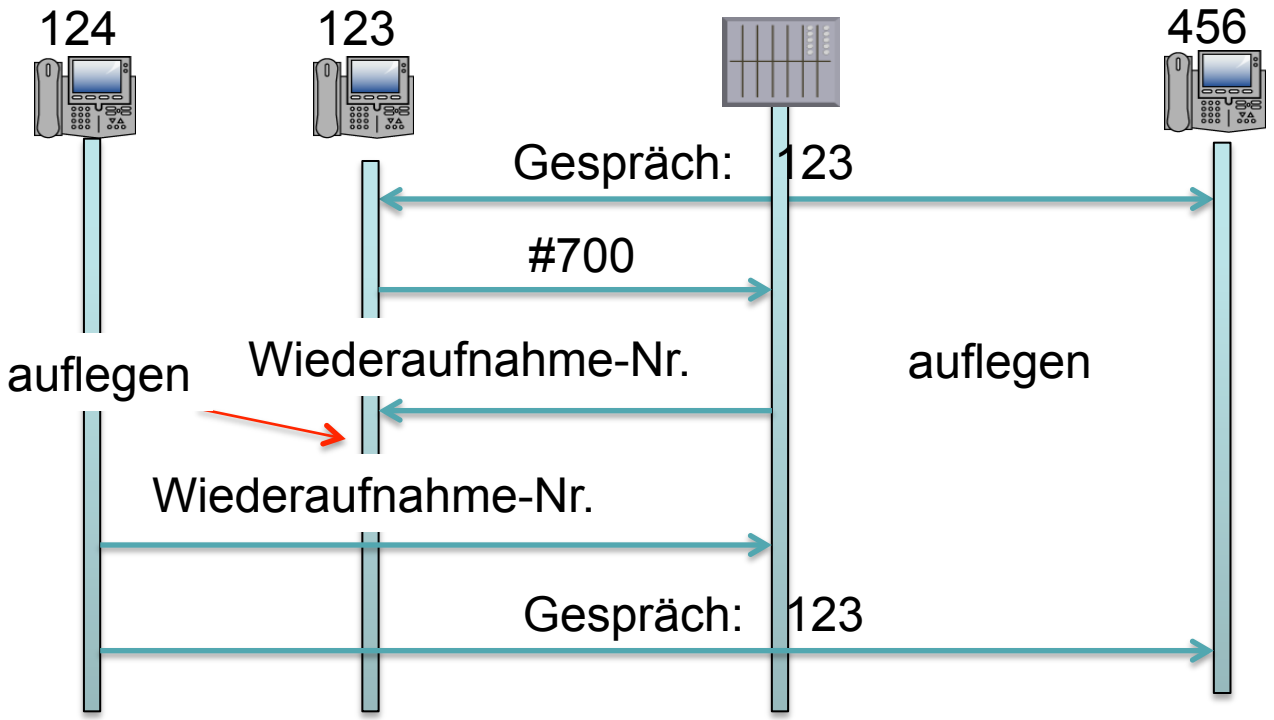
```
; Definition von Sprungzielen (Label):
exten => 123,1,Answer()
        same =>
        same => n(Anfang),Playback(Ansage)
        same => n,Dial(SIP/${EXTEN},20)
        same => n,Voicemail(${EXTEN},u)

; Unbedingter Sprung (Goto):
exten => 124,1,Answer()
exten => 124,n, Goto(123,Anfang)
```

Park-Funktion

Park-Prozedur:

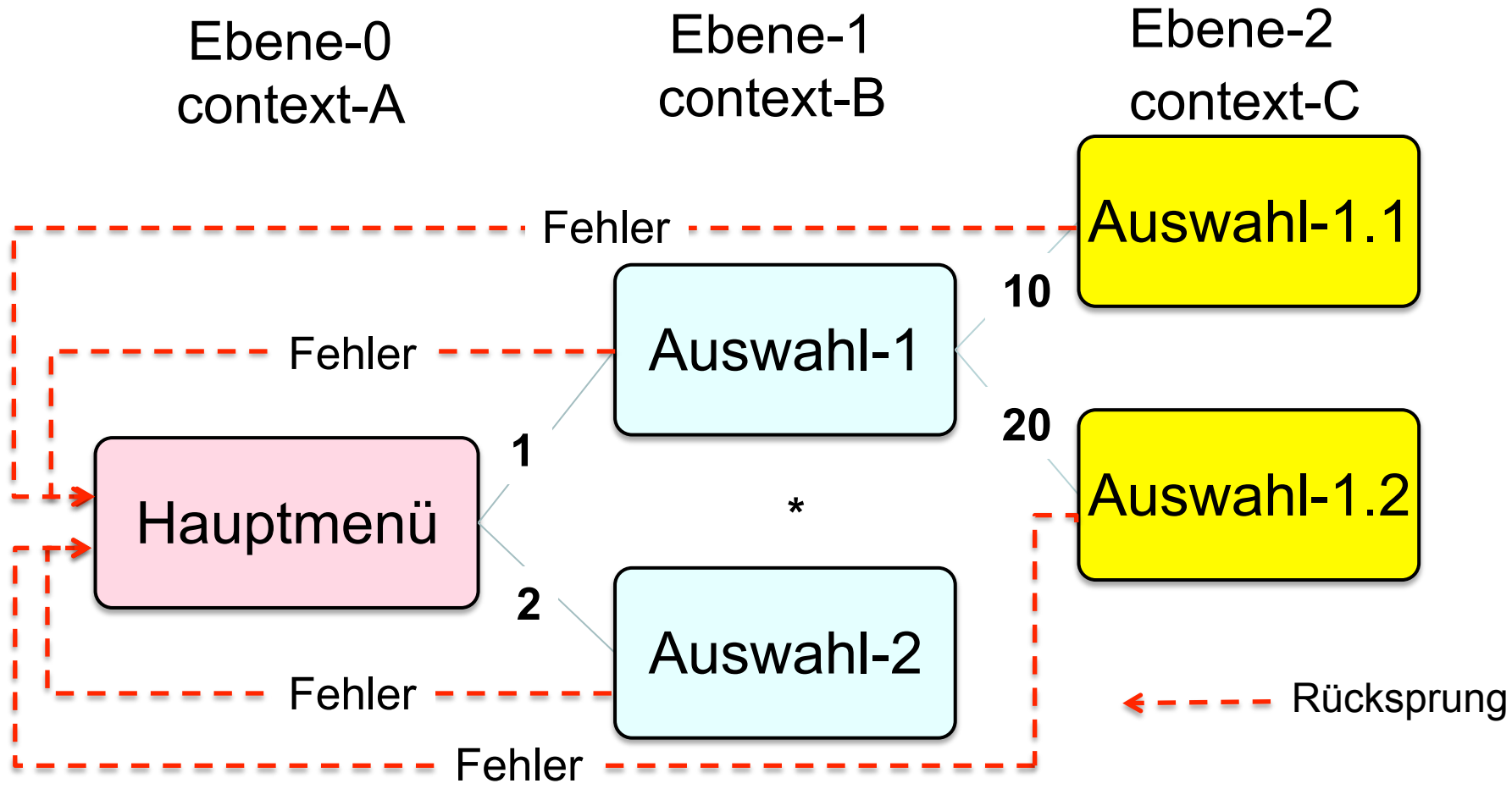
- Ein Telefongespräch soll über ein anderes Telefon fortgesetzt werden.
- Park-Kommando: #700



Interactive Voice Response - IVR

- Mittels IVR erhält der Anrufer ein akkustisches Auswahlmenü und antwortet darauf durch Spracheingabe oder durch Telefon-Tastatureingabe
- Asterisk verwendet die Telefon-Tastatureingabe
- Funktionen zur Abspielen der Menünachricht :
    - Background(Audio-Datei)
    - Playback(Audio-Datei)
- Die Tastatureingabe wird als Extension behandelt.
- Fehlerhafte Eingaben können durch die „i-Extension“ abgefangen werden.
- Mehrstellige Eingaben werden mittels Tastatur-Timeout überwacht.

Mehrstufige IVR Menüs



Jede Ebene besitzt ihren eigenen Kontext, dadurch können Extensions (Tastatureingaben) mehrfach verwendet werden.

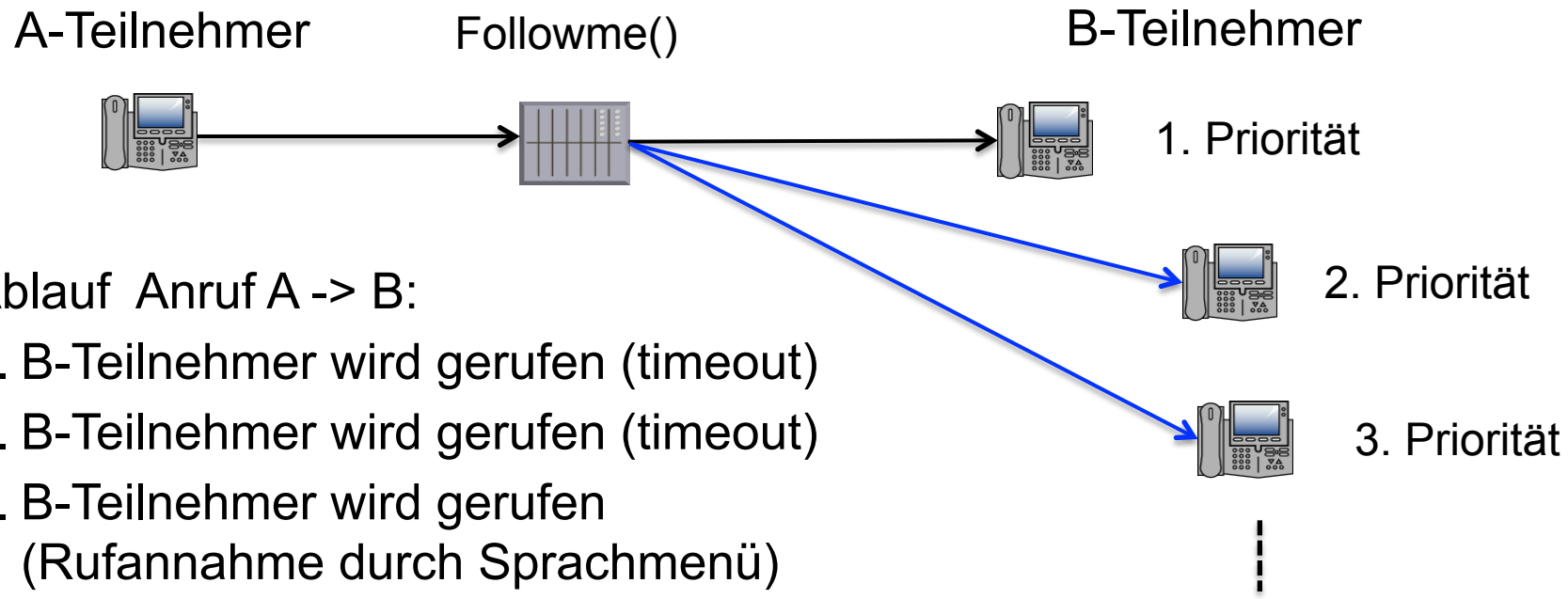
Zählerfunktionen

```
[CounterIncrement]
exten => start,1,Verbose(2,Increment the counter variable)
same => n,Set(CounterVariable=1) /* Zählervariable setzen */
same => n,Verbose(2,Zählerstand: ${CounterVariable})
same => n,Set(CounterVariable=${INC(CounterVariable)})
same => n,Verbose(2,Neuer Zählerstand: ${CounterVariable})
same => n,Hangup()
```

```
[CounterDecrement]
exten => start,1,Verbose(2,Increment the counter variable)
same => n,Set(CounterVariable=3) /* Zählervariable setzen */
same => n,Verbose(2,Zählerstand: ${CounterVariable})
same => n,Set(CounterVariable=${DEC(CounterVariable)})
same => n,Verbose(2,Neuer Zählerstand: ${CounterVariable})
same => n,Hangup()
```

Follow-Me Funktion

- Follow-me:
  - Nachbildung der ISDN-Festnetz Funktion
  - Erreichbarkeit mehrerer Ziele (Liste)
  - Sprachsteuerung z.B. Rufannahme-Menü



Ablauf Anruf A -> B:

1. B-Teilnehmer wird gerufen (timeout)
2. B-Teilnehmer wird gerufen (timeout)
3. B-Teilnehmer wird gerufen (Rufannahme durch Sprachmenü)

Dialplan Diagnose

Diagnosemöglichkeiten:

- Textausgaben aus dem Dialplan:
  exten => same,n,Verbose(2, Die Extension ist: ${EXTEN})
  exten => same,n,NoOp(“Die Extension ist: “ ${EXTEN})
- Ausgabe an der CLI-Konsole
- Verbose() ermöglicht die Ausgabe in Abhängigkeit vom eingestellten verbosity-Level: Diagnose-Switch
- NoOP erzeugt eine CLI-Ausgabe ab Level-2
- CLI Kommandos : dialplan show, etc.

Kursgliederung

DHBW Stuttgart

- Raspberry PI
- Netzwerkdiagnose
  - Kommandos
  - Analyse-Software Wireshark
  - Arbeiten mit Wireshark
- Asterisk – VoIP Einführung
- Asterisk Software
- Asterisk Programmierung