

Scalable, Distributed Conference Control in Heterogeneous Peer-to-Peer Scenarios with SIP

Eine Präsentation von
Alexander Knauf

Dept. Informatik
HAW Hamburg
Berliner Tor 7
D-20099 Hamburg, Germany

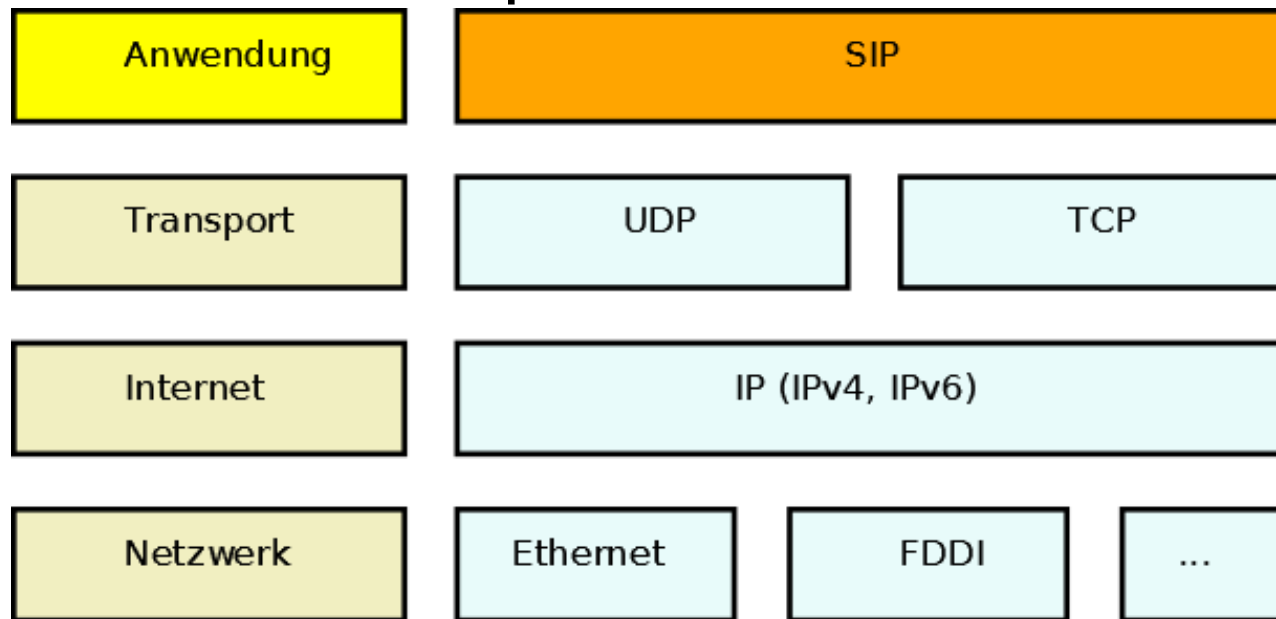


Übersicht

- Einführung
 - ◆ Grundlegendes zum *Session Initiation Protocol*
 - ◆ Das Prinzip des Focus
 - ◆ Das Problem des Focus
- Verteilter Focus
 - ◆ Zwei Lösungsansätze
 - ◆ „Hierarchische“ Verteilung
 - ◆ Overlay Network
- Unsere Lösung
 - ◆ Ermittlung neuer Focus
 - ◆ Verteilung der Last
 - ◆ Focus Ausfall
- Konsistenz der Zustände
 - ◆ Die Conference-Info Event Package Erweiterung
- Ergebnis

Einführung

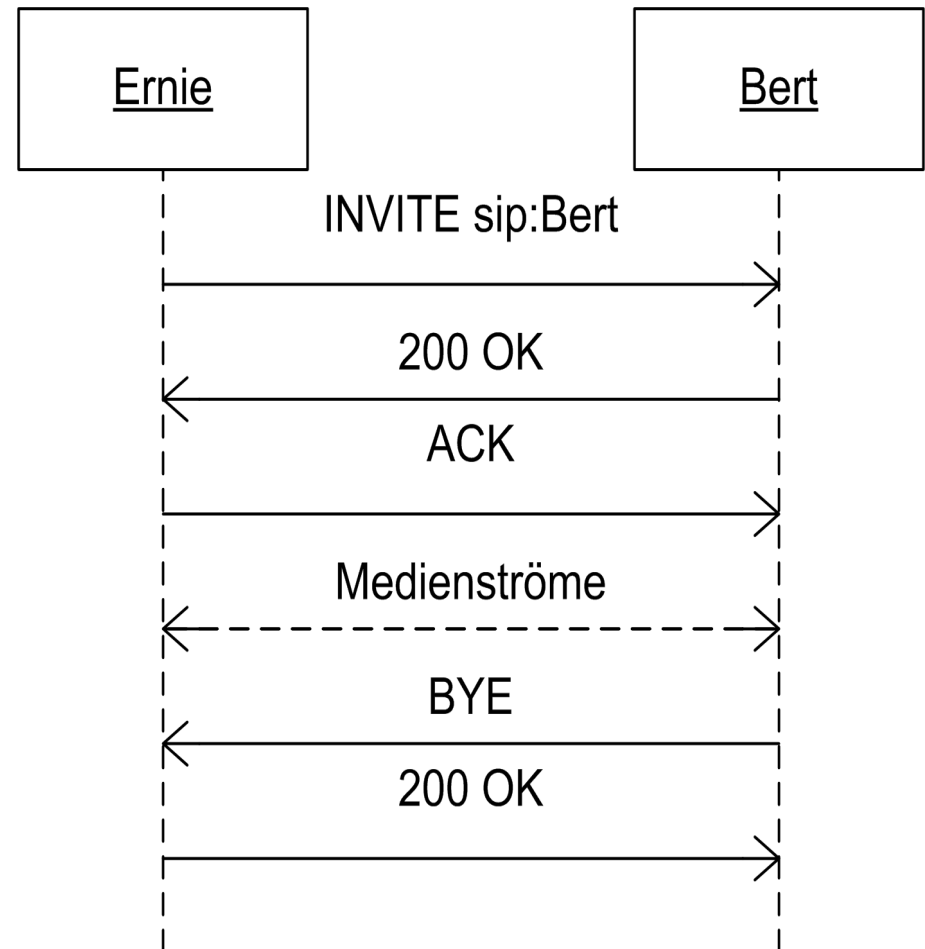
- SIP (**S**ession **I**nitiation **P**rotocol)
- Netzwerkprotokoll zum Aufbau und Steuerung von einer Kommunikationssitzung
- Wird im RFC 3261 spezifiziert



Einführung

- Kommunikationsaufbau:

- Eerie lädt Bert ein per SIP INVITE Request
- Wenn Eerie frei, Bestätigung mit OK Response
- ACK Nachricht zum Dialog bestätigen
- BYE Request zum Beenden des Dialogs
- Wird wieder bestätigt



Einführung

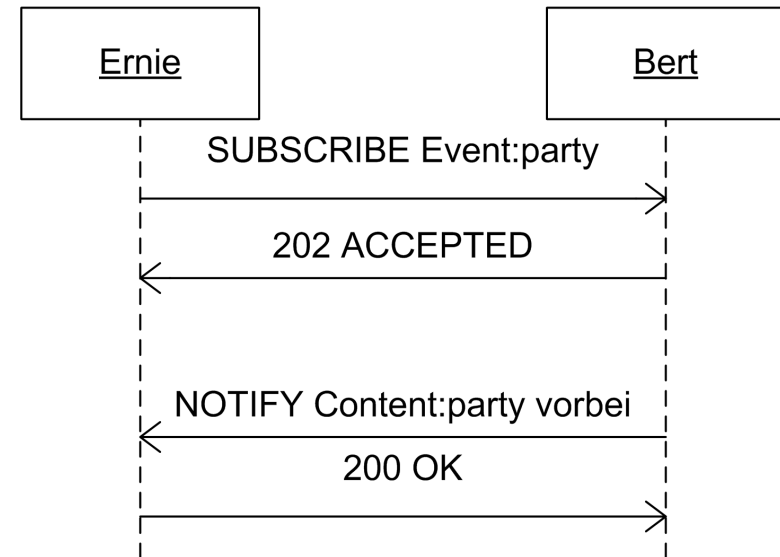
- SIP INVITE Request

```
INVITE sip:Bert@Sesamstrasse.de SIP/2.0
Call-ID: 0815@141.22.26.6
CSeq: 1 INVITE
From: "Ernie" <sip:Ernie@Sesamstrasse.de>;tag=134652
To: "Bert" <sip:Bert@Sesamstrasse.de>
Via: SIP/2.0/UDP 141.22.26.6:5060;
Max-Forwards: 70
Contact: <sip:Ernie@141.22.26.6>
Content-Length: 0
```

- Im Content Body noch weitere SDP Parameter möglich

Einführung

- Subscribe/Notify:
 - Client subscribed sich für ein spezifisches Event
 - 202 Accepted bestätigt
 - Wenn ein Ereignis auf das Event eintritt
Sendung eines SIP NOTIFY



Einführung

- Wie kommt man zu einer Konferenz?
- 2 Mögliche Wege:
 - Eine Organisation stellt SIP basierte Konferenz Applicationen zur Verfügung
 - Eine Client Application ist Fähig spontan eine Konferenz zu erzeugen

Einführung

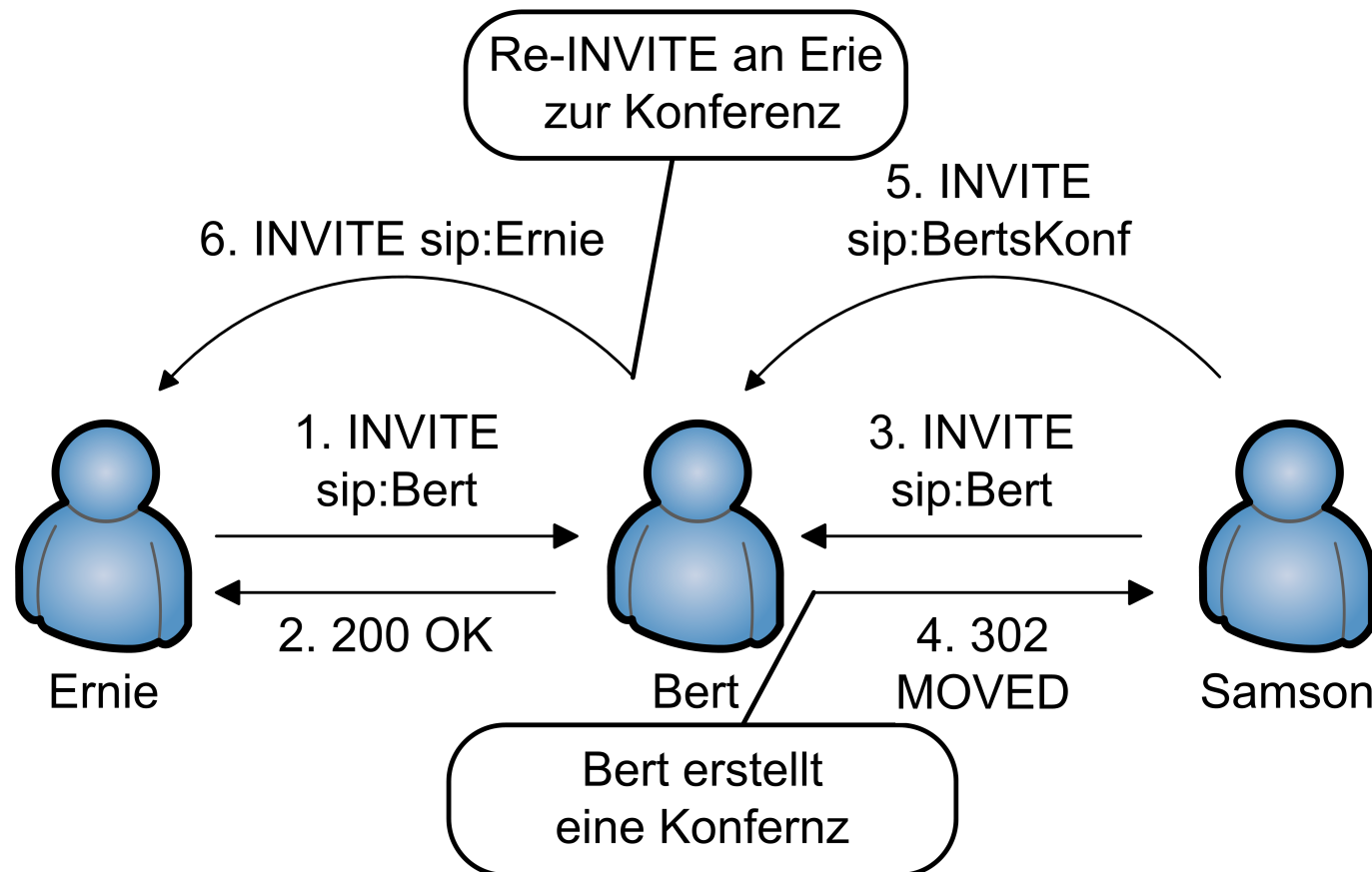
- 1. Fall:
 - SIP Client sendet SIP INVITE zu einer Konferenz Applikation (*Focus*)
 - Diese nimmt an und generiert eine Konferenz URI
 - Konferenz Applikation sendet ein neues SIP INVITE an den Client und teilt die Konferenz URI darin mit
 - Anfragende Client hat damit eine Konferenz eröffnet und kann andere SIP Clients dazu einladen. (z.B. via SIP REFER Request)

Einführung

- 2. Fall: (Für diesen Vortrag interessanter)
 - ♦ Zuerst 2 Party Dialogaufbau zweier SIP Clients
 - ♦ 3. Teilnehmer versucht einen Dialog zu einem der beiden SIP Client aufzubauen
 - ♦ Der angerufene erzeugt eine Konferenz URI und teilt sie dem 3. Teilnehmer mit und wird *Focus* der Konferenz
 - ♦ Dieser macht einen neuerlicher Verbindungsaufbau
 - ♦ Der *Focus* lädt den andern SIP Client zur Konferenz ein

Einführung

- Einleitung einer Konferenz



Einführung

- 302 MOVED Response

```
SIP/2.0 302 Moved Temporarily
Call-ID: 0816@141.22.26.235
CSeq: 1 INVITE
From: "Samson" <sip:Samson@Sesamstrasse.de>;tag=134652
To: "Bert" <sip:Bert@Sesamstrasse.de>;tag=43544
...
Contact: <sip:BertsKonf0@Sesamstrasse.de>;isfocus
Expires: 3600
...
```

Einführung

- Alle Teilnehmer der Konferenz können neue Mitglieder einladen via SIP REFER Request
- 2 Möglichkeiten:
 - ◆ Konferenzteilnehmer laden neue Mitglieder direkt selber ein
 - ◆ Konferenzteilnehmer fragen den Focus an neue Mitglieder einzuladen
- SIP Clients die „irgendwie“ die Konferenz URI kennen, können sich darüber einladen

Einführung

- SIP REFER Request

```
REFER sip:BertsKonf@Sesamstrasse.de SIP/2.0
Via: SIP/2.0/UDP 141.22.26.235:5060;branch=z9hG4bK23a
CSeq: 2 REFER
Call-ID: 0816@141.22.26.235
From: "Samson" <sip:Samson@Sesamstrasse.de>;tag=13465
To: <sip:BertsKonf@Sesamstrasse.de>;tag=43544
Max-Forwards: 70
Contact: "Samson" <sip:Samson@Sesamstrasse.de>
Refer-To: <sip:Bibo@Sesamstrasse.de>
Content-Length: 0
```

Einführung

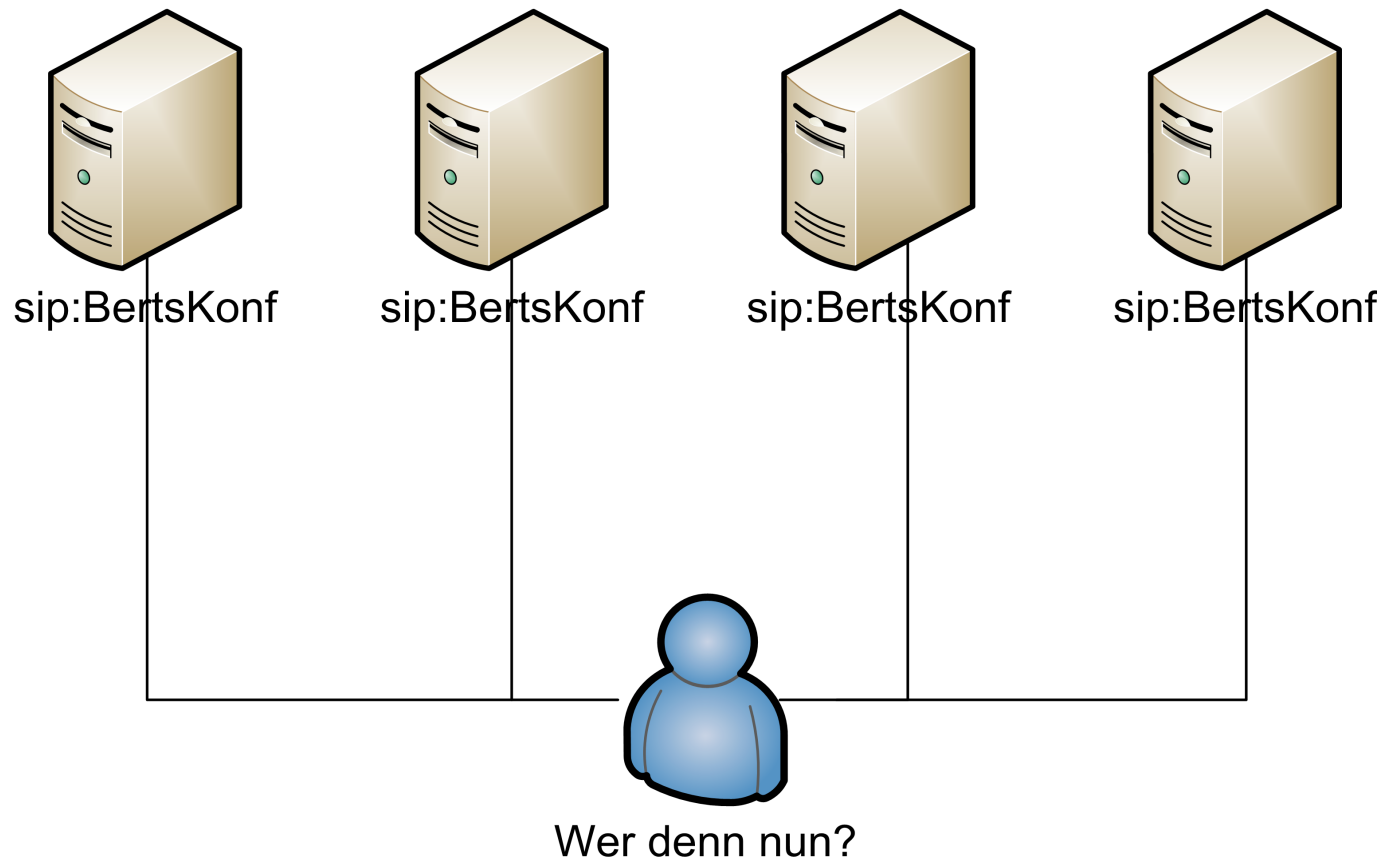
- Vorteile:
 - ◆ Ad-hoc Konferenz Aufbau
 - ◆ Keine externe Infrastruktur nötig
 - ◆ Keine externen P2P Protokolle nötig
- Nachteil
 - ◆ Focus Entität „nur“ normaler Consumer Computer
 - ◆ Bei wachsender Größe der Konferenz kann der Focus überlastet werden

Einführung

- Nativer Lösungsansatz für überlastete Knotenrechner:
 - ♦ Lastverteilung auf mehrere Knotenrechner!
- Probleme dabei:
 - ♦ Focus muss eine Globally Routable Useragent URI besitzen (GRUU)
 - ♦ Der Status der Verteilten Knotenrechner muss konsistent sein
 - ♦ Die Verteilung der Konferenz Focus muss transparent gegen über den SIP Clients sein

Einführung

- Problemstellung:



Übersicht

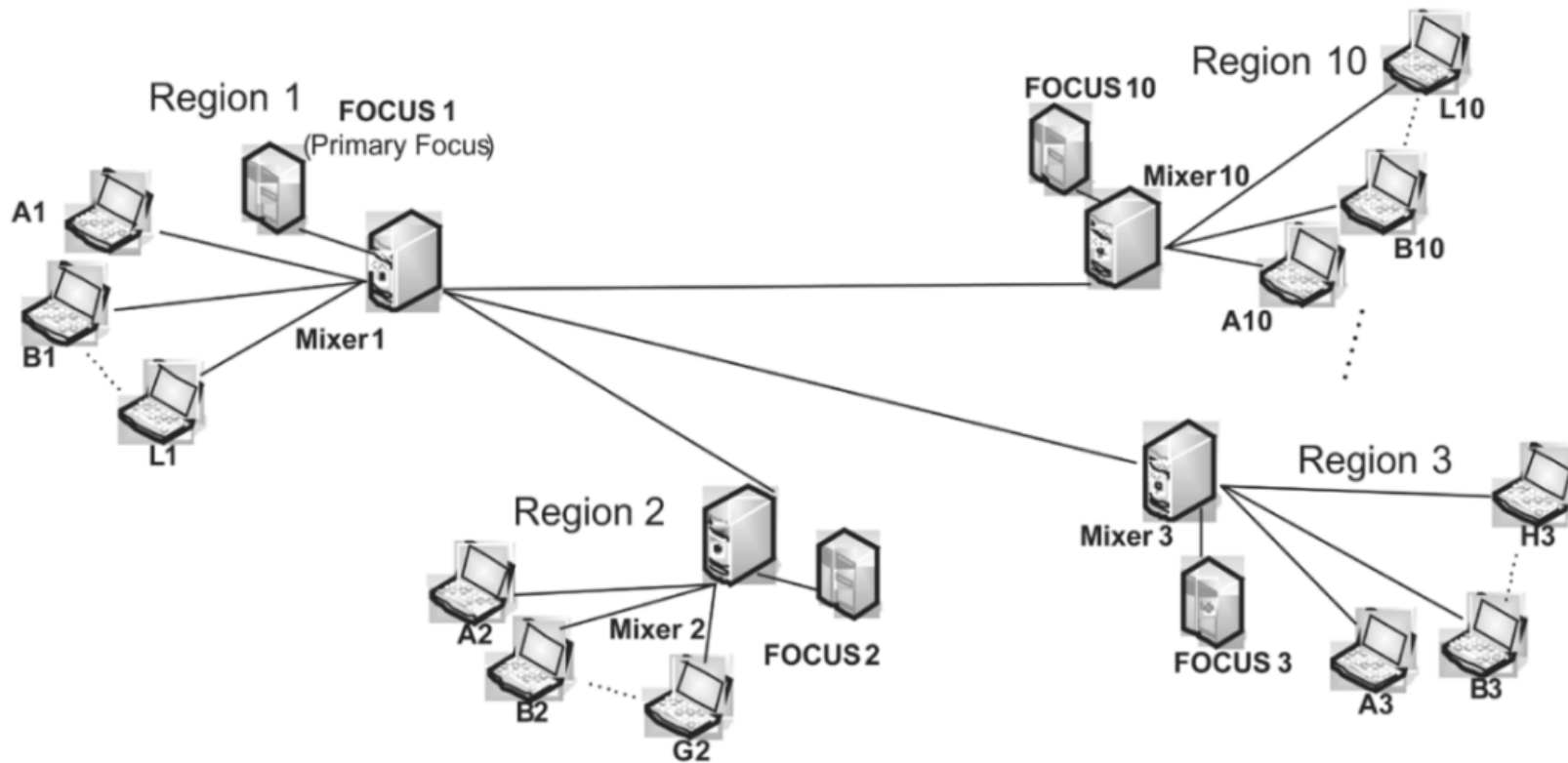
- Einführung
 - ◆ Grundlegendes zum *Session Initiation Protocol*
 - ◆ Das Prinzip des Focus
 - ◆ Das Problem des Focus
- **Verteilter Fokus**
 - ◆ Zwei Lösungsansätze
 - ◆ „Hierarchische“ Verteilung
 - ◆ Overlay Network
- Unsere Lösung
 - ◆ Ermittlung neuer Focus
 - ◆ Verteilung der Last
 - ◆ Focus Ausfall
- Konsistenz der Zustände
 - ◆ Die Conference-Info Event Package Erweiterung
- Ergebnis

Verteilter Focus

- Lösungsansatz von Yeong-Hun Cho und Team:
- „Hierarchischer“ Aufbau:
 - ♦ Aufteilung des Focus in *Primary Focus* und *Regional Focus*
 - ♦ Primary Focus: Zentraler Anlauf -und Verteilerknoten (Halter der GRUU)
 - ♦ Regional Focus: Konferenz Kontrolle für eine Teilmenge der Konferenzmitglieder
- Focus Status Konsistenz durch eine Erweiterung des CPCP (Conference Policy) für Verteiltes Conferencing

Verteilter Focus

- Hierarchische Konferenzkontrolle:

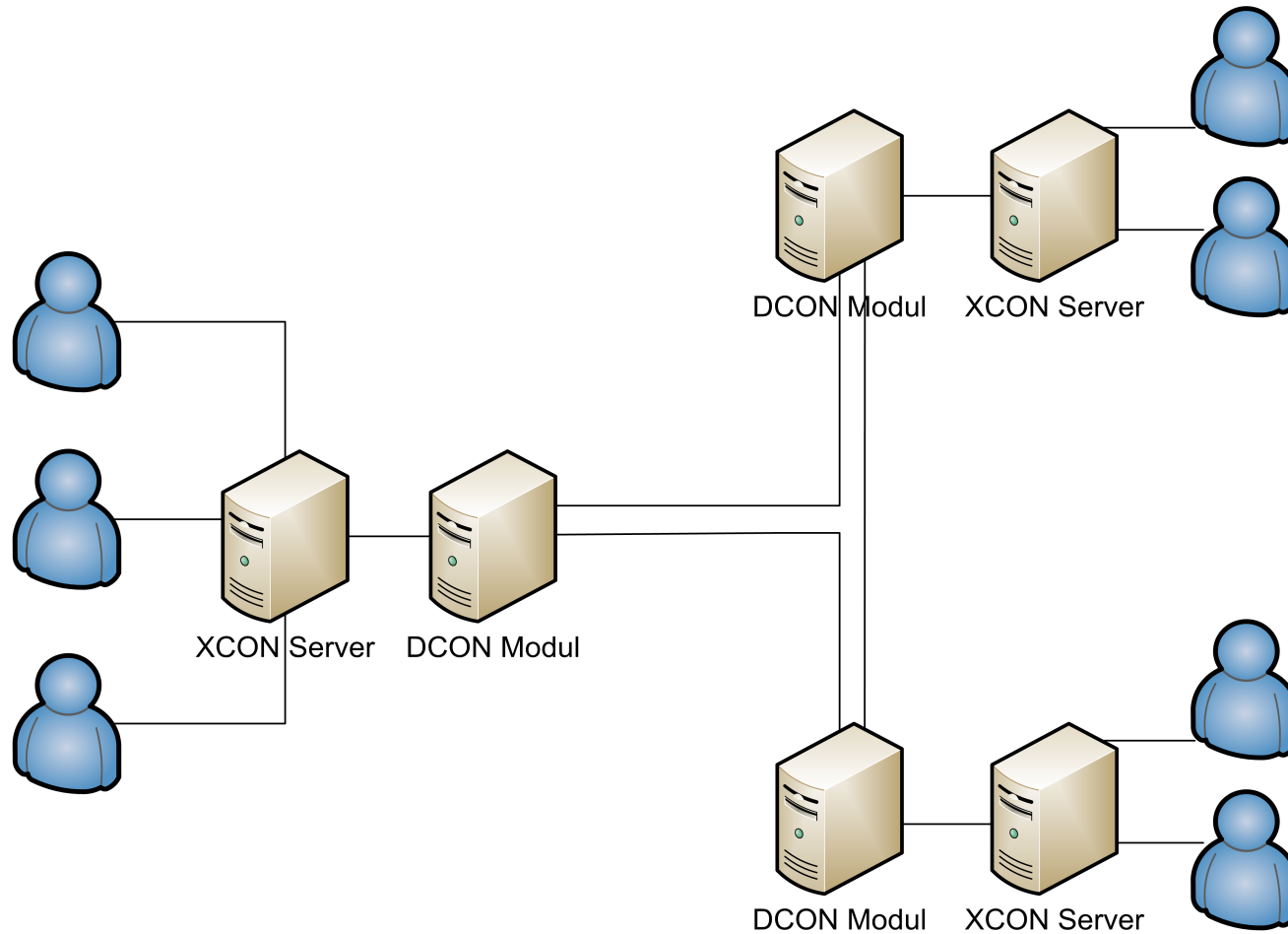


Verteilter Focus

- Lösungsansatz von P. Romano und Team (IETF Network WG)
- Zentralisierte *XCON* Konferenz Controller kommunizieren über ein *DCON* Overlay Network miteinander
- XCON und DCON Entität bilden eine *Conferencing Cloud*
- DCON Applikationen nutzen ein externes Protokoll für Verteilung und Konsistenz

Verteilter Focus

- DCON Architektur:



Übersicht

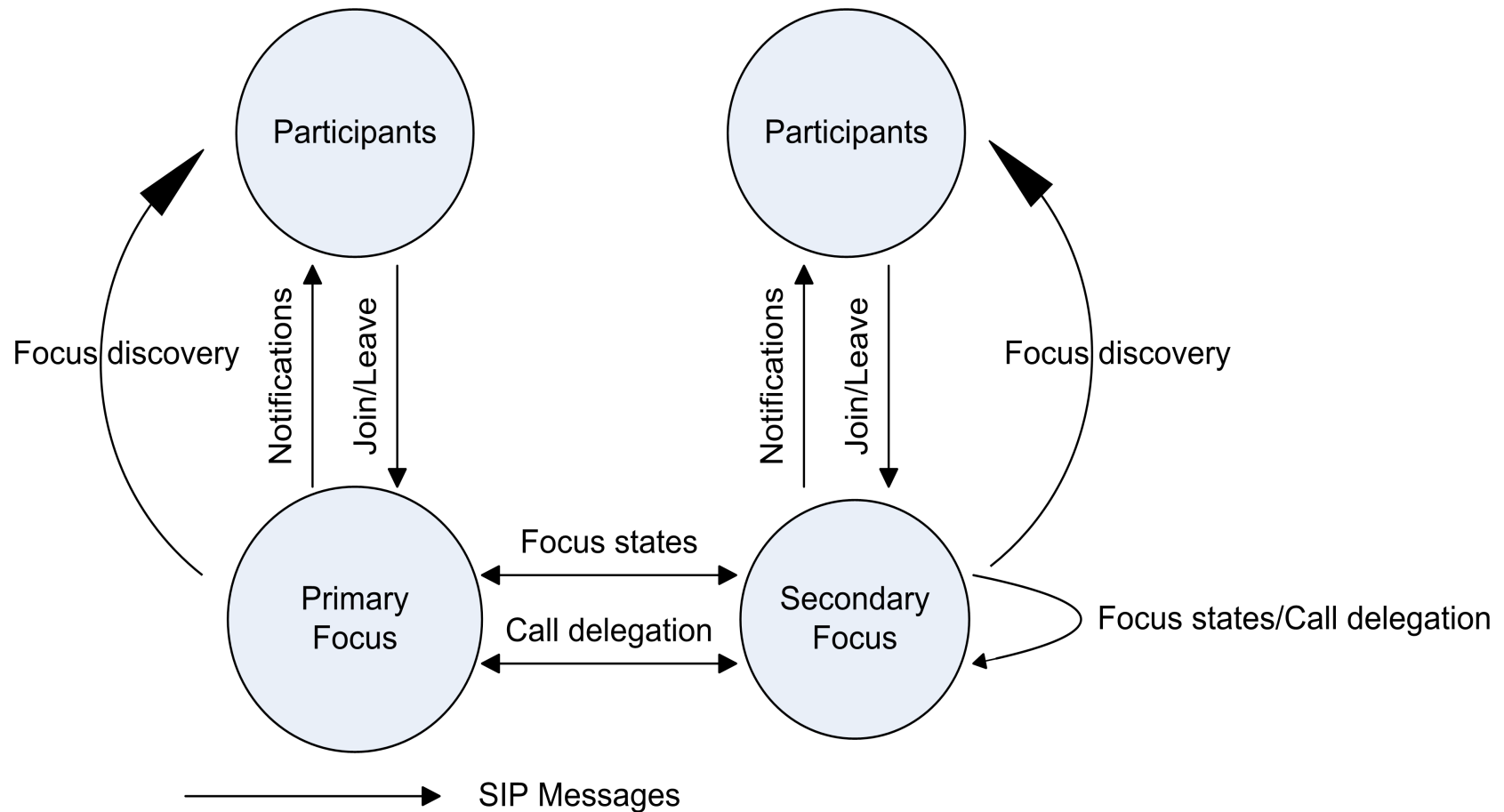
- Einführung
 - ◆ Grundlegendes zum *Session Initiation Protocol*
 - ◆ Das Prinzip des Focus
 - ◆ Das Problem des Focus
- Verteilter Fokus
 - ◆ Zwei Lösungsansätze
 - ◆ „Hierarchische“ Verteilung
 - ◆ Overlay Network
- **Unsere Lösung**
 - ◆ Ermittlung neuer Focus
 - ◆ Verteilung der Last
 - ◆ Focus Ausfall
- Konsistenz der Zustände
 - ◆ Die Conference-Info Event Package Erweiterung
- Ergebnis

Unsere Lösung

- Konferenzinitiator wird Halter der Konferenz URI
- Bei Bedarf sucht der Initiator neue potenzielle Focus Knoten (*Secondary Focus*)
- Secondary Focus Knoten intercepten SIP Nachrichten von an ihnen verbundenen SIP Clients
- Jeder Focus kann bei Überlastung SIP Calls an bekannte Focus Knoten weiterreichen
- Focus Knoten *Subscribe* sich gegenseitig, um einen konsistenten Zustand zu halten

Unsere Lösung

- Schematischer Aufbau:

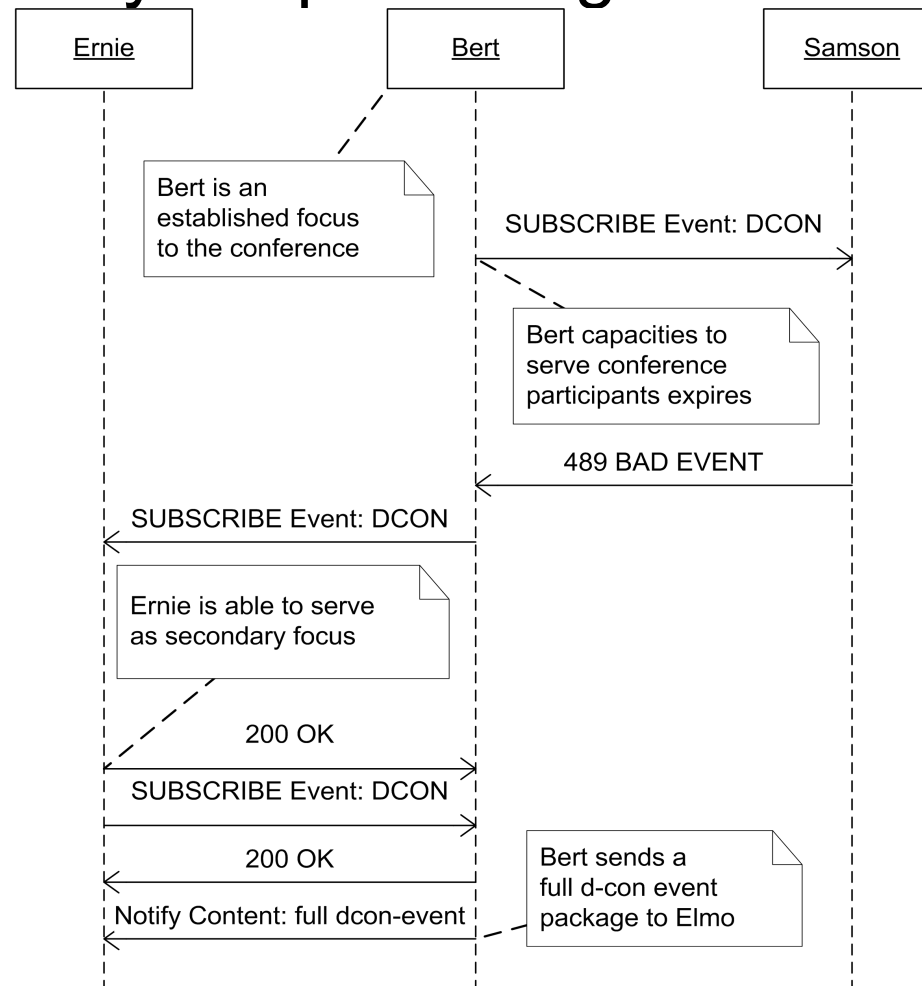


Ermittlung neuer Focus

- Szenario:
 - ◆ Ein Focus nähert sich max. Anzahl an Clients, die er vorsorgen will
 - ◆ Suche nach potenziellen Focus Knoten innerhalb der Konferenz (iterativ)
 - ◆ Via SIP SUBSCRIBE Request anfragend nach dem *inet-ci-multifocus-ext* Event Package
 - ◆ Beim Empfangen einer 200 OK Response Nachricht ist ein neuer Focus gefunden
 - ◆ Neuer Focus subscribed den alten Focus

Ermittlung neuer Focus

- Focus Discovery Sequenzdiagramm:



Ermittlung neuer Focus

- SIP SUBSCRIBE Request für inet-ci-multifocus-ext:

```
SUBSCRIBE sip:Ernie@Sesamstrasse.de SIP/2.0
Call-ID: 0817@141.22.26.55
CSeq: 1 SUBSCRIBE
From: "Bert" <sip:Bert@Sesamstrasse.de>;tag=134653
To: "Ernie" <sip:Ernie@Sesamstrasse.de>
...
Contact: <sip:Bert@Sesamstrasse.de>
Event: inet-ci-multifocus-ext+xml
Expires: 3600
...
```

Ermittlung neuer Focus

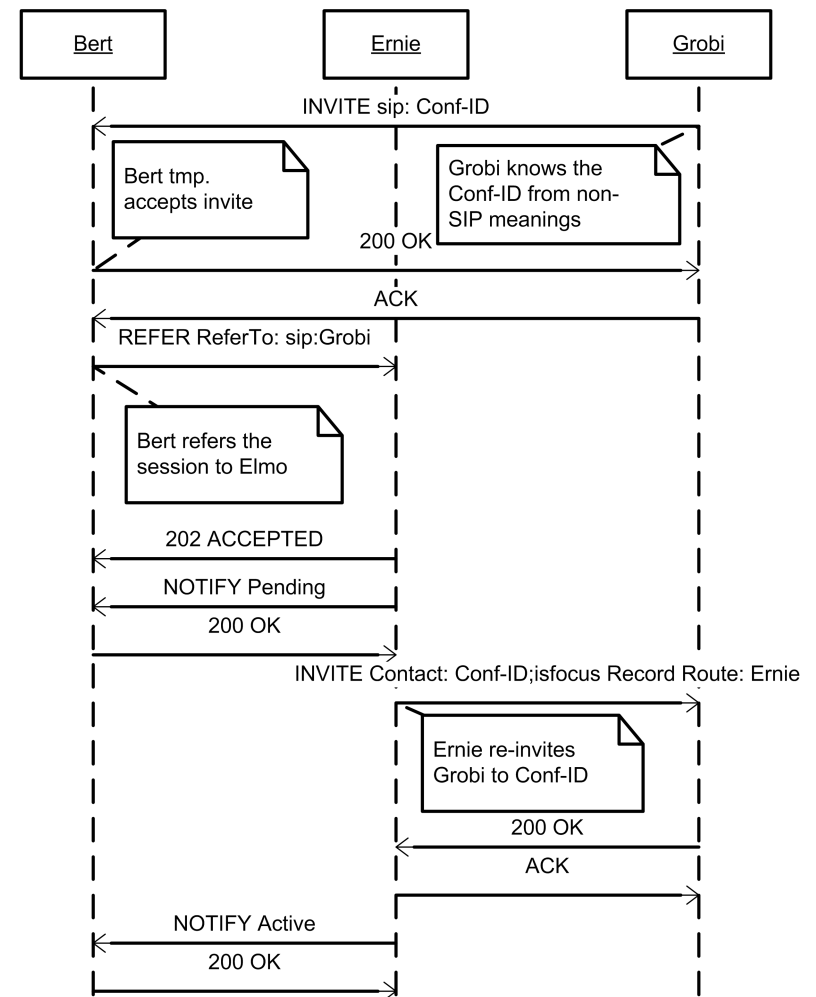
- Jeder Focus kann bei bedarf neue Focus erfragen
- Jeder neue Focus bekommt eine komplette Statusmeldung über die Konferenz (Focus, Teilnehmer, etc)
- Anhand dieser Statusmeldung soll der neue Focus kurze Wege zu anderen Focus Knoten erstellen
- Vermeidung von Focusketten!

Verteilung der Last

- 2 Wege in eine bestehende Konferenz:
 - ◆ INVITE eines neuen Teilnehmers auf die Konferenz URI
 - ◆ Ein Teilnehmer lädt via Focus einen neuen Teilnehmer ein via REFER Request
- Verteilung im 1. Fall:
 - ◆ Wenn noch Kapazitäten frei → Annahme des Calls
 - ◆ Sonst:
 - Der angefragte Focus nimmt den SIP Call vorläufig an
 - Call Weiterleitung an 2nd Focus via SIP REFER
 - Wenn 2nd Focus akzeptiert lädt er den neuen Teilnehmer erneut ein

Verteilung der Last

- 1. Fall:
- Focus Ernie ist jetzt zuständig für Konferenzkontrolle für Grobi
- Ernie erscheint für Grobi als Konferenzknoten sip:BertsKonf
- Alle Requests von Grobi an die Konferenz behandelt nun Ernie
- Nutzung eines Record-Route Header (später genauer)



Verteilung der Last

- Verteilung im 2. Fall: (Einladung via REFER)
 - ◆ REFER Requests werden **nicht** automatisch zum Konferenzinitiator geroutet!
 - ◆ Teilnehmer, die von einem Secondary Focus eingeladen wurden, routen ihre Nachrichten über diesen
 - ◆ Hinzufügen eines *Record-Route Headers* bei Einladung
 - ◆ Bewirkt ein Source Routing für nachfolgende Requests
 - ◆ 2nd Focus intercepten Konferenzrequests ihrer Teilnehmer

Verteilung der Last

- SIP INVITE mit Record-Route Header:

```
INVITE sip:Tiffy@Sesamstrasse.de SIP/2.0
Call-ID: 0818@141.22.26.55
CSeq: 1 INVITE
From: <sip:BertsKonf@Sesamstrasse.de>;tag=134652
To: "Tiffy" <sip:Tiffy@Sesamstrasse.de>
Via: SIP/2.0/UDP 141.22.26.6:5060;branch=z9hG4bKf1
Max-Forwards: 70
Contact: <sip:BertsKonf@Sesamstrasse.de>
Record-Route: <Ernie@Sesamstrasse.de>
Content-Length: 0
```

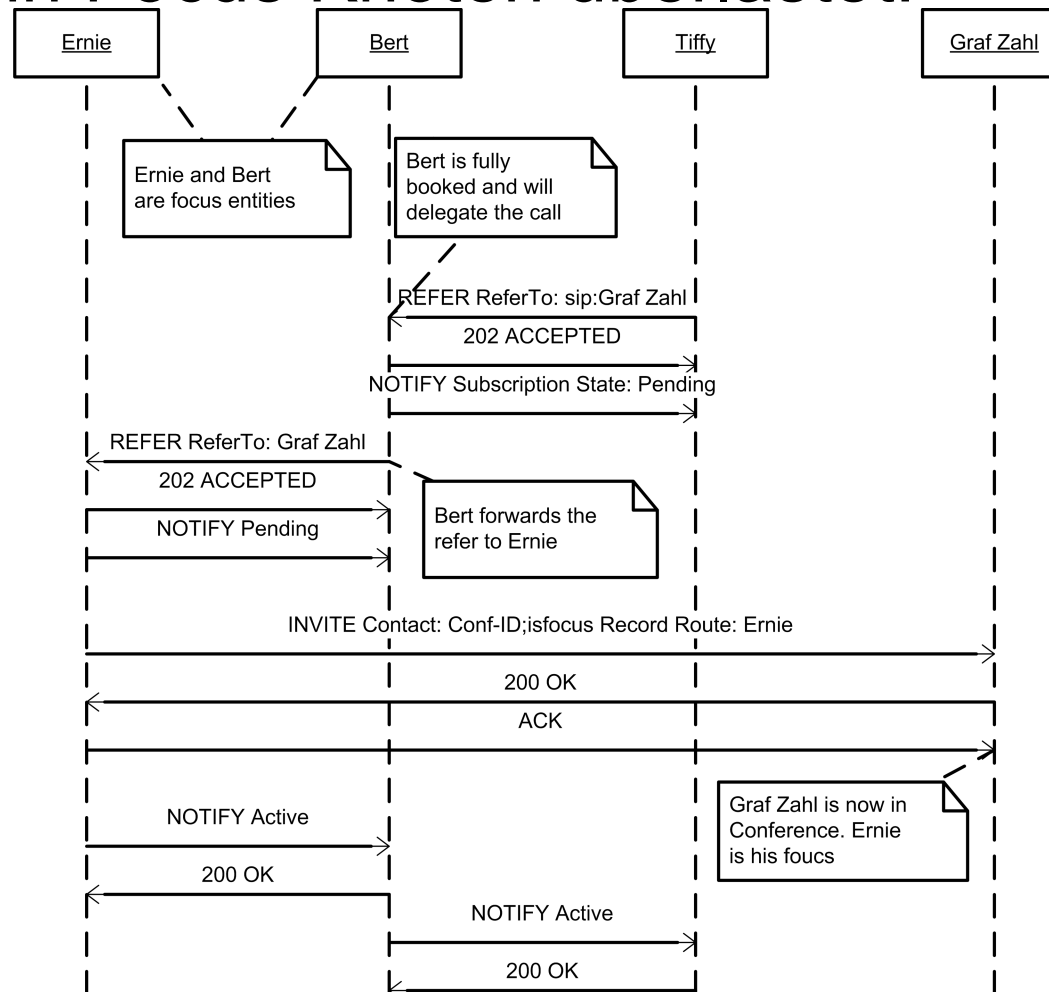
- Zwingt Tiffy Requets via Ernie zu senden

Verteilung der Last

- Fall 2, wenn Focus Knoten überlastet:
 - ♦ Ankommender REFER Request wird an einen freien Focus Knoten weitergeleitet
 - ♦ Dieser erkennt, dass es sich um ein weitergeleitetes REFER handelt und lädt den neuen Teilnehmer in die Konferenz ein
 - ♦ 2nd Focus erscheint dem neuen Teilnehmer als Konferenz Controller

Verteilung der Last

- Fall 2, wenn Focus Knoten überlastet:



Focus Ausfall

- Focus Knoten verlässt die Konferenz:
 - ◆ Muss seine SIP Calls an andere Focus Knoten übergeben via SIP REFER
- Focus Knoten fällt aus:
 - ◆ Knoten der den Ausfall bemerkt sollte vorläufig alle Teilnehmer übernehmen
 - ◆ Danach sollte er die Teilnehmer wieder via REFER an andere Focus Knoten verteilen
 - ◆ Weitere Möglichkeiten noch offen

Übersicht

- Einführung
 - ◆ Grundlegendes zum *Session Initiation Protocol*
 - ◆ Das Prinzip des Focus
 - ◆ Das Problem des Focus
- Verteilter Fokus
 - ◆ Zwei Lösungsansätze
 - ◆ „Hierarchische“ Verteilung
 - ◆ Overlay Network
- Unsere Lösung
 - ◆ Ermittlung neuer Focus
 - ◆ Verteilung der Last
 - ◆ Focus Ausfall
- **Konsistenz der Zustände**
 - ◆ Die Conference-Info Event Package Erweiterung
- Ergebnis

Konsistenz der Zustände

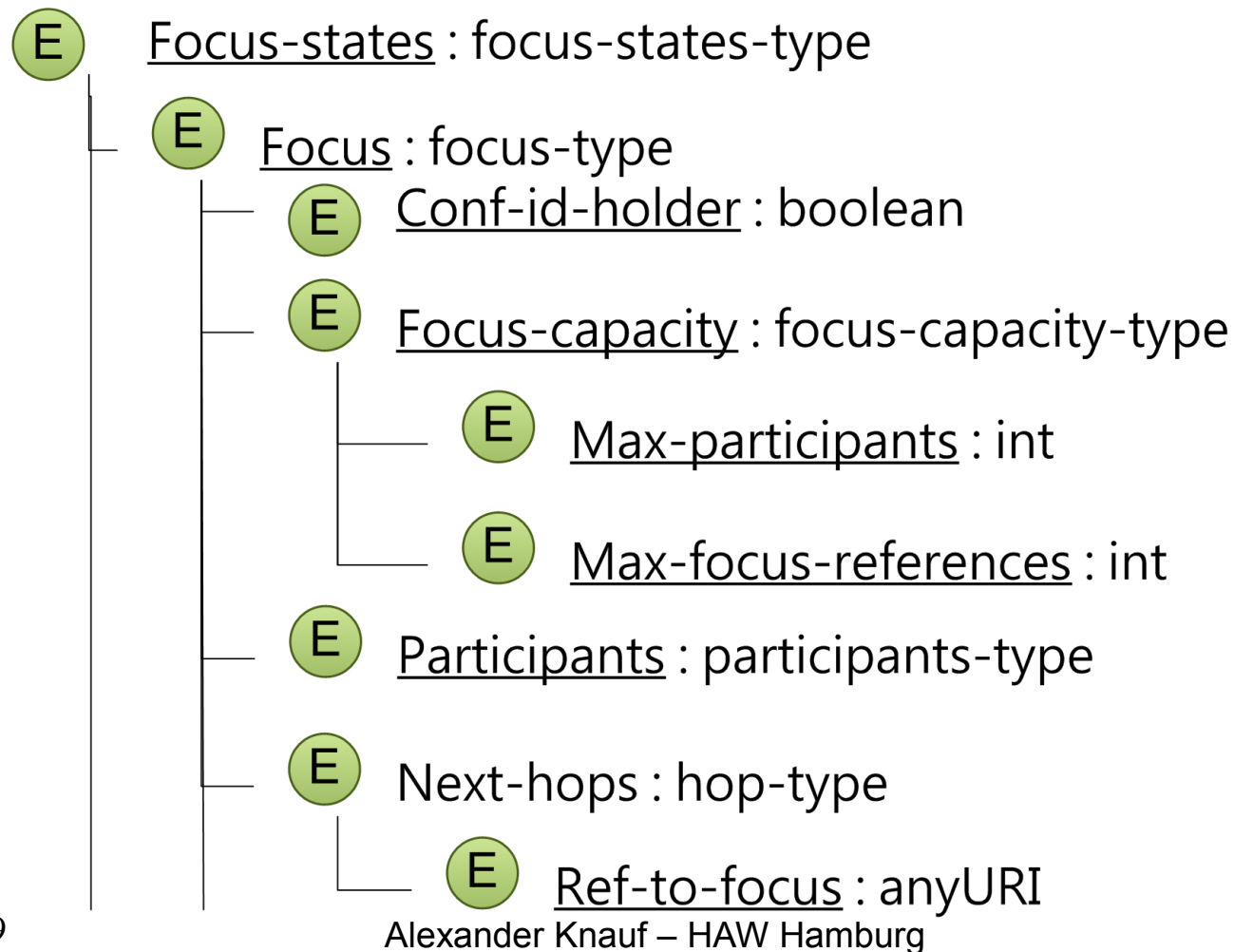
- Focus Knoten agieren autonom
 - ◆ Nehmen Calls an
 - ◆ Suchen neue Focus Knoten
 - ◆ Delegieren Calls weiter usw.
- Focus Knoten müssen über den Zustand der gesamten Konferenz Bescheid wissen
 - ◆ Wer sind die Konferenzteilnehmer
 - ◆ Wer sind die Focus Knoten
 - ◆ Wer hält welche Teilnehmer usw.

Konsistenz der Zustände

- Focus Knoten Subscriben sich gegenseitig auf das *inet-ci-multifocus-ext* Event Package
- Basis ist das Event Package for Conference State (RFC 4575)
- Wenn sich der Status eines Focus ändert schickt er ein SIP NOTIFY an alle Knoten die auf das Package subscribed haben
- Konferenzstatus im Content Teil in XML Form
- Focus Knoten Verteilen diese Nachricht weiter, so das jeden Knoten informiert ist

Konsistenz der Zustände

- XML Schema des *inet-ci-multifocus-ext*:



Konsistenz der Zustände

- Focus Knoten halten einen Zustand
- Speicheraufwand gering, Konferenzen normal unter 200 Teilnehmern
- Keine externen Protokolle verwendet
- Erweiterungen noch möglich

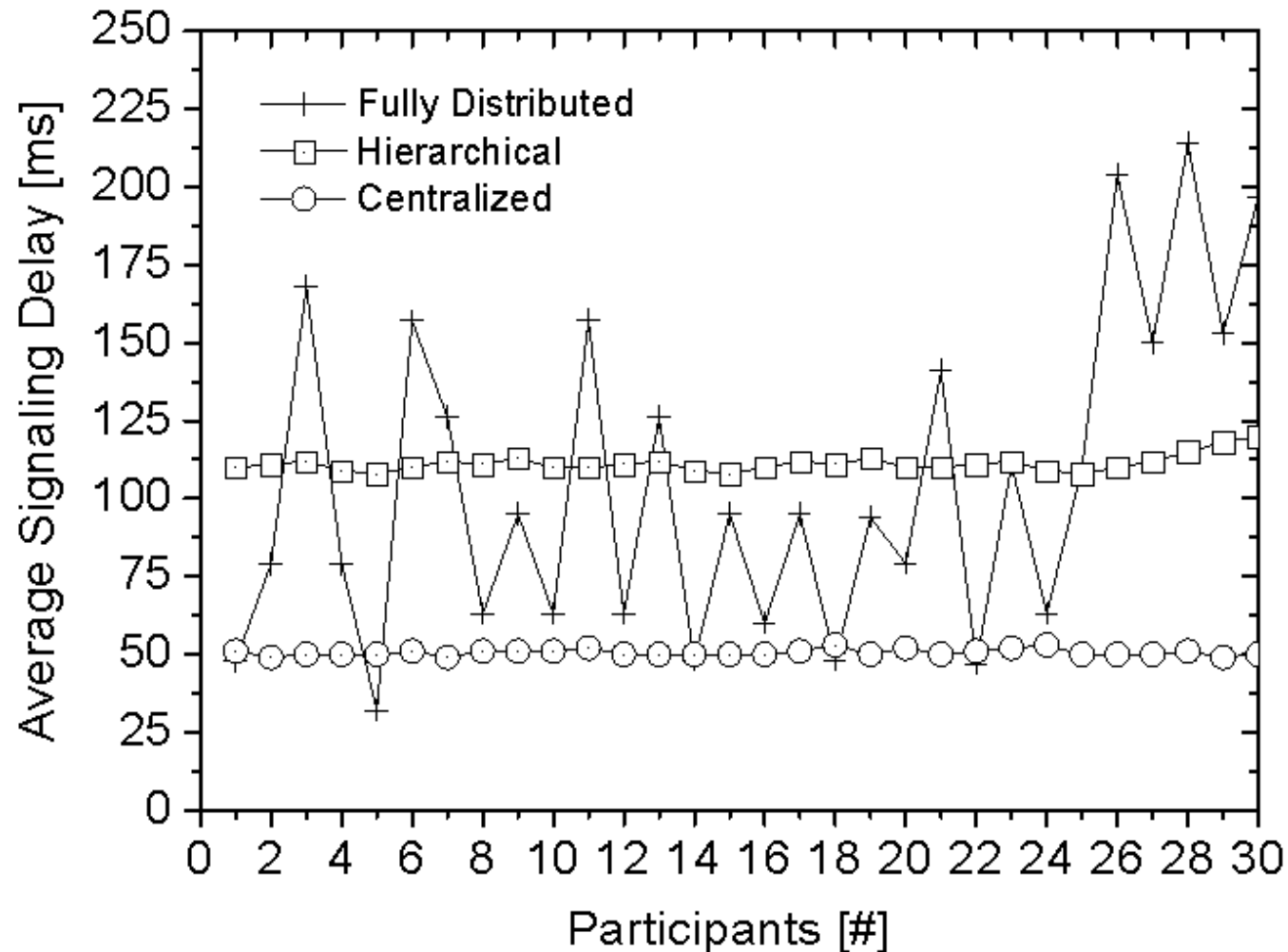
Übersicht

- Einführung
 - ◆ Grundlegendes zum *Session Initiation Protocol*
 - ◆ Das Prinzip des Focus
 - ◆ Das Problem des Focus
- Verteilter Fokus
 - ◆ Zwei Lösungsansätze
 - ◆ „Hierarchische“ Verteilung
 - ◆ Overlay Network
- Unsere Lösung
 - ◆ Ermittlung neuer Focus
 - ◆ Verteilung der Last
 - ◆ Focus Ausfall
- Konsistenz der Zustände
 - ◆ Die Conference-Info Event Package Erweiterung
- Ergebnis

Ergebnis

- Spontan Initiierte Konferenzen
- Keine externe Hardware nötig
- Transparent gegenüber den Konferenzteilnehmern
- Keine externen Protokolle nötig
- Zustände der Focus Knoten konsistent

Ergebnis



A Distributed Conference Control in Tightly Coupled SIP Scenarios

Vielen Dank für die Aufmerksamkeit!

Fragen und Diskussionen?

Referenzen (Auswahl)

- Y.-H. Cho, M.-S. Jeong, J.-W. Nah, W.-H. Lee, and J.-T. Park. Architecture for Large-Scale Enterprise Conferencing Service Using SIP. Selected Areas in Communications, IEEE Journal on, 23(10):1934-1949, Oct. 2005.
- A. Johnston and O. Levin. Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents. RFC 4579, IETF, Aug. 2006.
- A. B. Roach. Session Initiation Protocol (SIP)-Specific Event Notification. RFC 3265, IETF, June 2002. Updated by RFC 5367.
- S. Romano, A. Amirante, T. Castaldi, L. Miniero, and A. Buono. Requirements for Distributed Conferencing. Internet Draft { work in progress 04, IETF, Dec. 2008.
- J. Rosenberg. A Framework for Conferencing with the Session Initiation Protocol (SIP). RFC 4353, IETF, Feb. 2006.
- J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261, IETF, June 2002. Updated by RFCs 3265, 3853, 4320, 4916, 5393.
- J. Rosenberg, H. Schulzrinne, and O. Levin. A Session Initiation Protocol (SIP) Event Package for Conference State. RFC 4575, IETF, Aug. 2006.
- R. Sparks. The Session Initiation Protocol (SIP) Refer Method. RFC 3515, IETF, Apr. 2003.