# Wolfgang Kellerer

# Serverarchitektur zur netzunabhängigen Dienststeuerung in heterogenen Kommunikationsnetzen

Herausgegeben von

Prof. Dr.-Ing. Jörg Eberspächer Lehrstuhl für Kommunikationsnetze Technische Universität München

in der Reihe

Kommunikationstechnik



Herbert Utz Verlag · Wissenschaft München Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Ein Titeldatensatz für diese Publikation ist bei Der Deutschen Bibliothek erhältlich

Zugleich: Dissertation, München, Techn. Univ., 2002

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Wiedergabe auf photomechanischem oder ähnlichem Wege und der Speicherung in Datenverarbeitungsanlagen bleiben – auch bei nur auszugsweiser Verwendung – vorbehalten.

Copyright © Herbert Utz Verlag GmbH 2002

ISBN 3-8316-0123-2

Printed in Germany

Herbert Utz Verlag GmbH, München Tel.: 089/277791-00 – Fax: 089/277791-01

# Inhaltsverzeichnis

Kapitel Einführ			1
1.1	Ausg	angssituation und Motivation	1
1.2	Zielse	etzung, Lösungsansatz und Merkmale der Serverarchitektur	2
1.3		dnung der Arbeit	
1.4	Beitra	ag der Arbeit	5
1.5		lerung	
		6	
Kapitel Dienstu		and Anforderungen an die Steuerung von Diensten	7
2.1		imediale Informations- und Kommunikationsdienste	7
2.2		ogenität der Netze	
2.3		starchitektur: Grundstruktur und Aufgaben der Dienststeuerung	
2.5	2.3.1	Aufgaben	
	2.3.1	Grundstruktur: Partitionierung und Schnittstellen	
	2.3.3	Intelligenzverteilung	
2.4		ce Engineering	
	2.4.1	Betrachtung des Dienstlebenszyklus unter Kostengesichtspunkten	
	2.4.2	Dienstentwicklung: Vorgehensmodell	
	2.4.3	Eine integrierte Methode für die Kostenschätzung	
	2.4.4	Folgerungen für Dienstarchitekturen	22
2.5	Merk	male einer universellen Dienstarchitektur	
2.6	Beson	ndere Anforderungen an eine netzunabhängige Dienstarchitektur	23
Kapitel Verglei		Bewertung von Architekturen zur Dienststeuerung	28
3.1	Verte	ilte, protokollbasierte Dienstarchitekturen	29
	3.1.1	Breitband-ISDN	29
	3.1.2	Die Signalisierungsarchitekturen MAGIC und AMSA	30
3.2	Diens	starchitekturen mit getrennter Dienst- und Rufsteuerung	31
	3.2.1	Intelligente Netze	31
	3.2.2	Broadband Intelligent Network	33
3.3	Verte	ilte, Middleware-basierte Dienstarchitekturen	34
3.4	Verteilte, Endgeräte-basierte Dienstarchitekturen: Internet-Architekturen		
	3.4.1	Internet-Dienstarchitektur gemäß H.323	
	3.4.2	Internet-Dienstarchitektur gemäß SIP	
	3.4.3	Dienstarchitektur mit Megaco/H.248	40
2.5	A DI 1	pociartar Angotz	42

INHALTSVERZEICHNIS IX

	3.5.1	Parlay	42
	3.5.2	Java APIs for Integrated Networks (JAIN)	43
3.6	Netzunabhängige Dienstarchitekturen		
	3.6.1	IN-basierte Architekturen	45
	3.6.2	Weiterentwicklung der TINA-Ansätze	46
	3.6.3	Agentenunterstützte Ressourcensteuerung	47
	3.6.4	Programmierbare Netze	48
	3.6.5	Internet-basierte Architektur: ICEBERG	49
3.7	Disku	assion und Fazit	50
Kapitel	4		
Modelli	erung d	les neuen Dienststeuerservers	53
4.1	Grun	dkonzepte von SAMSON	53
	4.1.1	Trennung von Dienstebene und Netzebene	54
	4.1.2	Intelligenzverteilung	55
	4.1.3	Dienste und Dienstaufruf	56
	4.1.4	Kosteneffizienz	56
	4.1.5	Software-Architektur	56
4.2	Modellbildung: Sichtweisen auf die Systemarchitektur		
	4.2.1	Modellierung im ODP-Referenzmodell	57
	4.2.2	Objektorientierte Modellierung	58
	4.2.3	Formale, mathematische Modelle	58
	4.2.4	Modellierung nicht-funktionaler Anforderungen	58
	4.2.5	Ein neuer Modellierungsansatz für Dienstarchitekturen	59
	4.2.6	Zusammenfassung	62
4.3	Gescl	häftsmodell	62
4.4	Mode	ellierung der Steuerungsbereiche durch Sessions	64
	4.4.1	Session in SAMSON	64
	4.4.2	Modellierung von Dienste-Mobilität	65
4.5	Mode	ellierung der Information	66
	4.5.1	Modellierung der Dienstqualität	68
	4.5.2	Informationsbeschreibung der User Session	72
	4.5.3	Informationsbeschreibung der Service Session	73
	4.5.4	Informationsbeschreibung der Communication Session	74
	4.5.5	Aufbau der Teilnehmer- und Ressourcen-Adressierung	74
4.6	Kom	ponenten	76
	4.6.1	Zentralisierte Teilnehmerverwaltung (User Control)	
	4.6.2	Zentralisierte Dienststeuerung (Service Control)	78
	4.6.3	Zentralisierte Kommunikationssteuerung (Communication Control)	78
	4.6.4	Schnittstellen zu den Netzen: Anpassungseinheiten	79
4.7	Komi	munikation	79

		4.7.1	Anforderungen an das Signalisierungsverfahren	79	
		4.7.2	Signalisierungsprinzip		
		4.7.3	Zusammenfassung	82	
	4.8	Funkt	ionsweise der Dienstarchitektur	82	
	4.9	Zusar	nmenfassung	88	
	pitel : ezifik		er Dienstebene	89	
	5.1	Ein neues Signalisierungsprotokoll für die Dienststeuerung			
		5.1.1	Ausgangspunkt: Das IETF Session Initiation Protocol	90	
		5.1.2	Überblick über das Session Control Protocol		
		5.1.3	Nachrichtenformat und Parameter	95	
		5.1.4	Dienstbeschreibung	98	
		5.1.5	Transport der Signalisierungsnachrichten und Adressierung	100	
		5.1.6	Abgrenzung	101	
		5.1.7	Zusammenfassung und Ausblick	102	
	5.2	Teilne	ehmerverwaltung und Dienstzugang	103	
		5.2.1	Teilnehmerverwaltung	103	
		5.2.2	Dienstzugang und Dienstaufruf	105	
		5.2.3	Neues Konzept für die Netzunabhängigkeit des Dienstzuganges	106	
		5.2.4	Mobilitätsunterstützung	107	
	5.3	Zentra	ale Dienststeuereinheit	107	
		5.3.1	Struktur und Verhalten einer Dienstinstanz	107	
		5.3.2	Dienstlogik	108	
		5.3.3	Teilnehmer-Dienst-Interaktion	114	
		5.3.4	Interaktionen mit Datenservern	115	
	5.4	.4 Zusatzdienste		115	
	5.5	Komr	munikationssteuerung	116	
		5.5.1	Aufgaben der Kommunikationssteuerung		
		5.5.2	Anforderung durch die Dienststeuerung	118	
		5.5.3	Abfrage und Abstimmung der Teilnehmerprofile	119	
		5.5.4	Abbildung der Informationspfade auf Netzressourcen	121	
		5.5.5	Signalisierung zur Einrichtung von Verbindungen	123	
		5.5.6	Zusammenfassende Betrachtung des Communication Session Managers	125	
	5.6	Zusar	nmenfassendes Signalisierungsbeispiel	127	
Ka	pitel	6			
	•		Anpassung an unterschiedliche Netze	129	
	6.1	Netza	daptoren zur Steuerung der Dienstausführung	129	
		6.1.1	Paralleler Verbindungsaufbau durch Netz-APIs	131	
		6.1.2	Sequentieller Aufbau von Kommunikationsverbindungen	133	
		6.1.3	Aufbau von Kommunikationsverbindungen durch Gateways	134	

INHALTSVERZEICHNIS XI

	6.1.4 Fazit	135				
6.2	Teilnehmerinteraktion					
6.3	Steuerung zusätzlicher Ressourcen					
6.4	Neuartige Lösung für die Ressourcenverwaltung					
	6.4.1 Problemstellung	139				
	6.4.2 Attribute der Adaptoren	139				
	6.4.3 Lösung durch das Service Discovery-Konzept	141				
	6.4.4 Aufbau und Funktionsweise der Resource Registry	142				
6.5	Zusammenfassung	144				
Kapitel 7		145				
Prototypische Realisierung						
7.1	XML-basierter Teilnehmerzugang	146				
7.2	Spezifikation und Simulation des Signalisierungssystems					
7.3	Realisierung der Ressourcenverwaltung mit Jini					
7.4	Anbindung an reale Netze	151				
7.5	Fazit	151				
Zusamm	enfassung und Ausblick	152				
Literaturverzeichnis 15						
Anhang	Anhang					
A	Erläuterungen zu den Informationsmodellen der Sessions	1				
В	Details zur Dienstbeschreibung mit SDP+	3				
C	Document Type Definition der Session Programming Language (SPL)	6				
D	SesCP-Signalisierungsbeispiel	7				
E	Performance-Analyse des Signalisierungsprotokolls SesCP	12				
F	Beschreibung des Sessionmodells als Use-Case	16				
G	Erläuterungen zur Unified Modeling Language	18				

# Kapitel 1 Einführung

"The success of new service provision platforms will largely depend on their ability to blend with existing technologies" IEEE Communications Magazine [GHH+99]

# 1.1 Ausgangssituation und Motivation

Durch den rasanten technologischen Fortschritt in der Übertragungstechnik wird die reine Übertragungskapazität bald nicht mehr der allein erfolgsentscheidende Faktor bei Kommunikationsnetzen sein. Vielmehr wird der wirtschaftliche und technische Nutzen von Kommunikationssystemen in Zukunft in hohem Maße von den **Diensten** bestimmt, die den angeschlossenen Teilnehmern zur Verfügung gestellt werden. Die Teilnehmer erwarten über die Dienste, die ein Kommunikationssystem anbietet, die Deckung ihres Kommunikationsund Informationsbedarfs [Ebe96].

Bisher waren die Dienste sehr eng mit der Netztechnologie verknüpft und wurden stets zusammen mit einem Kommunikationsnetz eingeführt, wie z.B. dem Telefonnetz oder dem Bildschirmtext-System. Um die enge Bindung von Dienst und Netz aufzuheben und damit die Einführung und Entwicklung neuer Dienste in bestehenden Systemen zu erleichtern, wurden bereits Anfang der 80er Jahre Architekturen, wie die Intelligenten Netze [FGK97] eingeführt. Doch diese erweisen sich bei genauerer Betrachtung als in sich geschlossen und nur für bestimmte Mehrwertdienste in der Telefonie geeignet. Eine vollständige **Entkopplung** der Dienste von den Netzen, die für eine getrennte Evolution der Dienste notwendig ist, vollzieht sich nur sehr langsam.

In den letzten Jahren hat allerdings in verschiedenen Bereichen der Telekommunikation ein deutlicher **Wandel** stattgefunden, der sich direkt oder indirekt auf die Dienste und die Diensterbringung auswirkt [EP00]. Während zuvor eher eine stetige Evolution der Kommunikationssysteme getrennt in unterschiedlichen Anwendungsbereichen zu beobachten war, sind die derzeitigen Entwicklungen stark von **Kooperation** und **Integration** zwischen verschiedenen Systemen geprägt. Wichtige Einflußfaktoren sind dabei nicht unbedingt ausschließlich technische Neuerungen, sondern auch marktpolitische Entscheidungen, wie die Deregulierung auf dem Telekommunikationsmarkt.

Beginnend 1984 in den USA wurden seit 1998 auch in Europa die (staatlichen) Monopole für die Telekommunikation aufgelöst [Wit98]. Die damit einhergehende **Deregulierung** und somit Liberalisierung des Telekommunikationsmarktes führte zu einem vorher nicht existierenden **Wettbewerb** zwischen den ehemaligen Monopolisten und den neuen Anbietern.

2 KAPITEL 1

Attraktive **Dienste** für die Teilnehmer sind die entscheidenden Faktoren, um sich im Wettbewerb um neue Kunden auf dem Markt zu differenzieren. Der Wettbewerb zwingt die Anbieter, den Teilnehmern in immer kürzeren Abständen neue Dienste zur Verfügung zu stellen. Es ist daher Ziel, dieselben Dienste auf möglichst vielen Netzen anzubieten, ohne sie jedesmal neu implementieren zu müssen. Denn die Teilnehmer fordern nicht nur einfach zu bedienende, auf sie persönlich zugeschnittene, hoch funktionelle Dienste. Sie wollen diese auch **unabhängig von den Netzen** aufrufen, je nachdem, welche Anschlüsse gerade verfügbar sind.

Bestehende Systeme können diese Anforderung nach einem einheitlichen Dienstzugang nur unzureichend erfüllen. Die Bestrebungen Mitte der 90er Jahre, mit B-ISDN/ATM [Boc97] eine einheitliche Dienste- und Netzinfrastruktur zu schaffen, waren nicht erfolgreich. Auch beim künftigen Mobilfunkstandard der dritten Generation IMT2000/UMTS [HT01] ist es ungewiß, ob wirklich alle Dienste integriert werden können. Demgegenüber entstehen fortwährend neue, oftmals konkurrierende Netze und Steuerungsprotokolle, z.B. für *Voice over IP*: ITU-H.323 [OT99] und IETF-SIP [SR99]. Zudem werden viele der bestehenden Systeme weitergenutzt, weil sie bestimmte Dienstklassen besser erbringen (z.B. PSTN für Sprachtelefonie) oder zusätzliche Bandbreite bereitstellen (z.B. digitale Rundfunk-Verteilnetze). Bisher unbeachtete Systeme erfahren einen neuen Nutzen (z.B. *Powerline Communication*). Diese **Heterogenität** der Netzinfrastruktur im Teilnehmerzugang und im Übertragungsnetz ist bei der Erbringung von Diensten zu berücksichtigen. Sie stellt hohe Ansprüche an die Dienstplattform.

Auch im **Software Engineering** allgemein und speziell für die Informations- und Kommunikationstechnik haben sich Änderungen ergeben, die eine verbesserte Dienst- und Plattformentwicklung entscheidend unterstützen. Die Komplexität neuer Dienste kann durch den Einsatz von objektorientierten Sprachen, formalen Beschreibungstechniken und durch den konsequenten Einsatz von formalen Methoden im Entwicklungsprozeß drastisch reduziert werden [Bro97]. Dies ist insbesondere in einem kooperativen Umfeld verteilter Architekturen notwendig, in dem Systeme verschiedener Hersteller und Betreiber zusammenwirken. Um eine schnelle Dienstentwicklung ohne aufwendige Einarbeitung zu gewährleisten, gewinnen Standard-Software-Entwicklungsumgebungen (z.B. für Java) im Gegensatz zu proprietären, aber bisher vorherrschenden Entwicklungsumgebungen (*Service Creation Environments*) immer mehr an Bedeutung. Dies wird dadurch begünstigt, daß offene Programmierschnittstellen (APIs), eingebettet in Middlewareplattformen, proprietäre Protokollschnittstellen langsam verdrängen (z.B. Parlay [Par00]).

Bei der zukünftigen Entwicklung von Diensten sind alle oben beschriebenen Faktoren von hoher Bedeutung. Insbesondere die Existenz einer wachsenden Vielzahl von leistungsfähigen Netzen, Systemen, Standards und Endgeräten muß entsprechend berücksichtigt werden. Die Heterogenität stellt *die* Herausforderung dar, um neue Dienste effizient einzuführen und gleichzeitig die Charakteristika der bestehenden Systeme intelligent zu nutzen. Dazu ist es notwendig, die Steuerung von Diensten möglichst unabhängig zu machen von den spezifischen Technologien der bestehenden, heterogenen Kommunikationsnetze. In der vorliegenden Arbeit wird eine Lösung für diese Problemstellung in Form einer Serverarchitektur aufgezeigt.

## 1.2 Zielsetzung, Lösungsansatz und Merkmale der Serverarchitektur

Ziel der vorliegenden Arbeit ist es, eine Konvergenz der unterschiedlichen Kommunikationssysteme auf Dienstebene zu erreichen, um Dienste unabhängig von den Netzen und Teilnehmerzugängen steuern zu können und darüber hinaus Dienste netzübergreifend anbieten zu können. Als Lösung wird eine neuartige Serverarchitektur beschrieben, die die Dienststeuerung von der netzspezifischen Signalisierung entkoppelt. Die Dienststeuerung soll nicht nur unabhängig von einem Netztyp sein, sondern es sollen Dienste für die Teilnehmer in verschie-

EINFÜHRUNG 3

denen, heterogenen Kommunikationsnetzen **übergreifend** erbracht werden. Dafür ist eine einheitliche Schnittstelle der Dienststeuerung zu den Netzen zu definieren. Das Modell der betrachteten Dienste oberhalb dieser Schnittstelle ist so flexibel zu gestalten, daß alle Arten von multimedialen Informations- und Kommunikationsdiensten mittels der Architektur steuerbar sind.

Eine Dienststeuerung ist ein hochkomplexes Kommunikationssystem, für das keine allgemeine Entwicklungsmethodik existiert [Kel98]. Bestehende Softwareentwicklungsmethoden erweisen sich als zu allgemein, da sie nicht auf die besonderen Bedingungen der Dienste eingehen, und sind deshalb entsprechend für die Dienstentwicklung zu spezialisieren.

Daher wurde ausgehend von obigem Lösungsansatz folgendes **Vorgehen** gewählt, um die Serverarchitektur als technische Lösung zu entwickeln. Die Analysephase ist durch die Validierung der Ideen mit Dienst-Szenarien geprägt [HKS97]. Um der Komplexität von Kommunikationssystemen Rechnung zu tragen, wird das Gesamtsystem aus ausgewählten Sichtweisen abstrakt beschrieben. Zur Umsetzung des entstandenen Modells werden existierende Ansätze und Systeme herangezogen und diese entsprechend angepaßt. Auf diese Weise wird die abstrakte Idee anschaulich und in einer bestehenden Umgebung schnell implementierbar gemacht. Da entsprechende mathematische Verifikationsmethoden für diese hochkomplexen Systeme noch Stand der Forschung sind (z.B. [Hin98b]), wird die Funktionsweise der Systemkomponenten mit Prototypen nachgewiesen. Insbesondere formale Beschreibungssprachen haben sich im Vorfeld der Arbeit für die Analyse, für Simulationen und für Rapid Prototyping als äußerst geeignet erwiesen [Hin98a, Kel99b, KIR96, VKK98].

Mit diesem Vorgehen wird in der vorliegenden Arbeit eine Serverarchitektur entworfen. Sie weist folgende **Merkmale** auf:

- Die Dienststeuerung in heterogenen Netzen wird durch Adaptoren realisiert.
  Diese arbeiten als Signalisierungsgateways zwischen dem Signalisierungsprotokoll
  der Serverarchitektur und der netzspezifischen Signalisierung. Sie ermöglichen
  eine Auswahl und Kombination verschiedener Kommunikationssysteme für die
  Diensterbringung.
- Für die Steuerung von Diensten in den unterschiedlichen Netzen wird auf bestehende, standardisierte APIs zurückgegriffen. Somit kann die Architektur einfach umgesetzt werden, ohne Änderungen in den Netzen zu erfordern.
- Die Serverarchitektur ermöglicht das Geschäftsmodell eines neuen, Netzanbieterunabhängigen Dienstanbieters.
- Die Serverarchitektur wird in drei getrennt verwaltete Teilbereiche unterteilt: Teilnehmersteuerung, Dienststeuerung und Kommunikationssteuerung.
- Die Teilnehmersteuerung regelt den Teilnehmerzugang und die Teilnehmerprofilverwaltung unabhängig von der Dienststeuerung. Sie unterstützt damit die bisher wenig berücksichtigte, aber derzeit wachsende Personalisierung und die Mobilität bei Diensten.
- Objekt-orientierte Beschreibung der Dienstzustände (Call Model) zur generischen Modellierung eines umfassenden Dienstspektrums, das nicht auf bestimmte Dienstklassen beschränkt ist.
- Dynamisch verfeinertes Qualitäts-Modell: Die Beschreibung des Dienstes wird bei der Verarbeitung durch die Serverkomponenten ausgehend vom Dienstaufruf des Teilnehmers per Dienstname bis zu einer detaillierten Kommunikationsdienst-Beschreibung schrittweise konkretisiert.

4 KAPITEL 1

Für die Signalisierung innerhalb der Dienststeuerung wird ein neues Signalisierungsprotokoll spezifiziert, das auf dem Session Initiation Protocol der IETF (SIP) basiert. Damit wird erstens durch die Ausrichtung auf das Internet eine einfache Integration der Architekturkomponenten in eine IP-basierte Infrastruktur ermöglicht. Zweitens können bestehende Netzkomponenten für die Signalisierung eingesetzt werden.

- Der Zugang zum Server ist durch die Verwendung von XML-Technologie [Mar00] und der Umsetzbarkeit von XML-Inhalten in verschiedene Ausgabeformate unabhängig von der Informationsdarstellung im Endgerät.
- Das Service Discovery-Konzept wird auf die Dienststeuerung übertragen. Damit kann die Auswahl von Kommunikationssystemen unterstützt werden und die Serverarchitektur automatisch an eine sich ändernde Umgebung angepaßt werden.

Die wesentlichen Bestandteile der Serverarchitektur wurden in einer prototypischen Realisierung umgesetzt, um die Funktionsweise der hier vorgestellten Konzepte zu validieren.

# 1.3 Einordnung der Arbeit

An dieser Stelle wird kurz auf Ansätze und Systeme eingegangen, die eng mit der in dieser Arbeit vorgestellten Serverarchitektur zusammenhängen. Weitere Details zu diesen Architekturen und eine eingehende Analyse des Stands der Technik können in Kapitel 3 nachgelesen werden.

#### Intelligente Netze

Das Konzept der Intelligenten Netze (IN) [FGK97] wird seit den 80er Jahren in der Erbringung von leistungsfähigen Zusatzdiensten für die Telefonie eingesetzt. Hauptmerkmal ist die logische Trennung der Basisvermittlung (Basisdienst) von einer zentralisierten Steuerung von Zusatzdiensten wie z.B. Freephone, Virtual Private Network. Ziel war es, die Abhängigkeit der Dienstentwicklung von den Switch-Herstellern zu reduzieren und die Dienstentwicklung durch die Wiederverwendung von dedizierten Dienstbausteinen zu beschleunigen. Die IN-Architektur ist stark auf Telefoniedienste eingeschränkt und ermöglicht nur eine teilweise Unabhängigkeit der Dienststeuerung von der Netzinfrastruktur. Trotzdem ist das Grundkonzept der zentralen, vom Netz abgesetzten Dienststeuerung Grundlage jeder modernen Dienststeuerung. In diesem Sinn fließen die Erkenntnisse aus der Analyse der IN-Architektur in die vorliegende Arbeit ein

#### TINA - Telecommunication Information Networking Architecture

Aus einer internationalen Initiative der wichtigsten Telekommunikationsfirmen (TINA-C) ist Mitte der 90er Jahre eine abstrakte Beschreibung einer umfassenden Dienst-, Netz- und Management-Architektur entstanden, die auf ATM-basierte Netze ausgerichtet ist [ILM99]. Wenngleich der tatsächliche Einsatz dieser Architektur bisher nicht nachgewiesen wurde, so sind viele darin enthaltenen Konzepte wegweisend für die weitere Entwicklung der Kommunikationssysteme. Die vorliegende Arbeit baut auf dem TINA-Konzept zur Trennung der Dienststeuerung in Teilnehmer-, Dienst- und Kommunikations-Session für die funktionale Dekomposition der Dienstarchitektur auf.

EINFÜHRUNG 5

### **IETF Session Initiation Protocol (SIP)**

In RFC 2543 wurde 1999 von der IETF ein Ende-zu-Ende Signalisierungsprotokoll für die Internet-basierte Multimediakommunikation standardisiert [SR99]. Das Protokoll ist durch eine universelle Verwendbarkeit für beliebige Dienste gekennzeichnet. Dies bewirken die Trennung von Transaktionsprotokoll und transportierter Dienstbeschreibung und ein einfacher Aushandlungsmechanismus. Daher wurde SIP in der vorliegenden Arbeit als Basis für das Signalisierungsprotokoll zur Dienststeuerung verwendet.

#### Parlay-API

Die *Parlay Group* wurde 1998 von namhaften Telekommunikationsfirmen gegründet [Par00]. Sie hat das Ziel, eine einheitliche Schnittstelle (Parlay-API) zu definieren, über die Dienstanbieter auf fremde Netze zugreifen können. Dem Parlay-API wird derzeit eine große Bedeutung für die zukünftige Dienstentwicklung beigemessen. Es ist auch ein wichtiger Faktor für die in dieser Arbeit vorgestellte Serverarchitektur, die oberhalb einer solchen Schnittstelle anzusiedeln ist, da ein Parlay-Interface erstmals einen standardisierten Zugang zu Netzen für unabhängige Dienstanbieter ermöglicht. Parlay definiert nur ein API, aber keine Komponenten für eine Dienststeuerung, z.B. für die Teilnehmerverwaltung oder den Teilnehmerzugang.

### 1.4 Beitrag der Arbeit

Die vorliegende Arbeit liefert aufgrund der beschriebenen Merkmale der Serverarchitektur Beiträge in wichtigen Gebieten der Kommunikationsnetze und des Software Engineerings:

- Konvergenz der Netze: Es wird ein Kommunikationssystem beschrieben, das auf Dienstebene durch eine strikte Entkopplung von Diensten und Netzen die Forderung nach der Konvergenz der Netze unterstützt. Die beschriebene Architektur ermöglicht es, Dienste unabhängig von der vorhandenen Infrastruktur anzubieten.
- Dienstarchitekturen: Die vorliegende Arbeit spezifiziert eine komponentenorientierte Architektur und ein objektorientiertes Dienstmodell für die Steuerung von Diensten über Netz-APIs. Sie bietet nicht nur eine neuartige, technische Lösung für die obige Zielsetzung, sondern zeigt auch allgemein nötige, abstrakte Komponenten und eine generelle Struktur einer idealen Dienststeuerung. Für Netz-APIs wie z.B. Parlay sind bisher keine Architekturen definiert, die wie hier Teilnehmersteuerung, Dienststeuerung und Kommunikationssteuerung integrieren.
- Signalisierung für die Dienststeuerung: Für die Kommunikation der Serverkomponenten wird ein neuartiges Signalisierungsprotokoll basierend auf dem IETF Session Initiation Protocol entwickelt. Da IP-basierte Signalisierung und speziell SIP in vielen Bereichen als Grundlage für die Signalisierung verwendet wird (z.B. bei der dritten Generation Mobilfunk), stellt das neue Protokoll einen Beitrag zur laufenden Standardisierung dar.
- Software Engineering: Um die Software-Architektur der Dienststeuerung auf Systemebene abstrakt darstellen zu können, wird ein Systemmodell vorgestellt. Dessen Sichtweisen sind auf die Spezifikation von Dienstarchitekturen abgestimmt.

Daneben zeigt die Übertragung von neuen Konzepten aus der Informationstechnik (das Metadatenformat XML [Mar00] und das *Service Discovery*-Prinzip) auf die Steuerung von Informations- und Kommunikationsdiensten, wie deren Vorteile für zukünftige 6 KAPITEL 1

Kommunikationssysteme nutzbar gemacht werden können. Die Anwendung wurde in prototypischen Realisierungen gezeigt.

### 1.5 Gliederung

Kapitel 2 beschreibt das Umfeld von Informations- und Kommunikationsdiensten anhand von Begriffsdefinitionen, der Analyse der Netzinfrastruktur, Grundprinzipien von Dienstarchitekturen, der Dienstentwicklung und den Kosten im Dienstlebenszyklus. Daraus werden die generellen Merkmale von Dienstarchitekturen entwickelt. Anhand von Beispielszenarien werden konkrete Anforderungen an eine netzunabhängige Dienststeuerung abgeleitet, die im Schwerpunkt der weiteren Betrachtungen liegt.

Kapitel 3 gibt einen Überblick über existierende Architekturen und Ansätze zur Steuerung von Diensten und bewertet sie anhand der in Kapitel 2 aufgestellten Anforderungen. Dabei wird besonders auf bestehende Ansätze für netzunabhängige Architekturen eingegangen und der Stand der Technik analysiert, der in engem Zusammenhang mit der beschriebenen Serverarchitektur steht.

In Kapitel 4 wird das Konzept der Serverarchitektur erläutert, die sich in eine Dienstebene und in eine Anpassungsschicht unterteilt. Anschließend wird ein Modell für Dienstarchitekturen vorgestellt. Die zugehörigen Sichtweisen erlauben es, die Architektur aus verschiedenen Blickwinkeln abstrakt zu beschreiben. Das Geschäftsmodell unterscheidet die an einem Dienst beteiligten Aktoren und definiert externe Schnittstellen der Dienststeuerung. Aus der Informations-Sichtweise wird die Datenstruktur und aus der Komponenten-Sichtweise wird die Strukturierung in Komponenten mit den zugehörigen Schnittstellen beschrieben. Die Session-Sichtweise fokussiert die Teilnehmerinteraktionen. Aus der Betrachtung der Architektur aus der Kommunikations-Sichtweise folgen die Anforderungen an das Signalisierungsverfahren zwischen den Komponenten. Ein ausführliches Beispielszenario zeigt die Funktionsweise der Architektur und das Zusammenspiel der einzelnen Komponenten.

Aufbauend auf der abstrakten Modellierung der Dienstarchitektur in Kapitel 4 werden in Kapitel 5 die interagierenden Komponenten der Dienststeuerungsebene spezifiziert. Als erstes wird das neue Signalisierungsprotokoll SesCP (Session Control Protocol) beschrieben. Anschließend werden die Komponenten der Teilnehmerverwaltung und des Dienstzugangs, die Komponenten der Dienststeuerungsmodule und die Dienstlogik, sowie die Komponenten der Kommunikationssteuerung spezifiziert. Kapitel 5 schließt mit einem detaillierten Beispiel eines Signalisierungsablaufs.

In Kapitel 6 wird die Anpassungsschicht der Serverarchitektur beschrieben, welche die Einrichtung von Kommunikationsbeziehungen in den Netzen realisiert. Hauptkomponenten sind die Adaptoren für die Kommunikationssteuerung, für den Dienstzugang, für die Steuerung von zusätzlichen Ressourcen und für die Ressourcenverwaltung. Es wird eine Klassifikation vorgeschlagen und die Funktion der Adaptoren beschrieben.

Kapitel 7 gibt einen Überblick über die Spezifikation und die prototypische Realisierung ausgewählter Teile der Serverarchitektur. Zum Nachweis der Realisierbarkeit der Dienstarchitektur wurde ein Prototyp des Teilnehmerzugangs, der Ressourcenverwaltung und ein SDL-Simulator für das Signalisierungsprotokoll erstellt.