

Reihe: Telekommunikation @ Mediendienste · Band 14

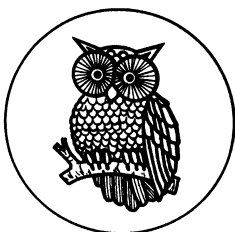
Herausgegeben von Prof. Dr. Dr. h. c. Norbert Szyperski, Köln, Prof. Dr. Udo Winand, Kassel, Prof. Dr. Dietrich Seibt, Köln, Prof. Dr. Rainer Kuhlen, Konstanz, Dr. Rudolf Pospischil, Brüssel, Prof. Dr. Claudia Löbbecke, Köln, und Prof. Dr. Christoph Zacharias, Köln

PD Dr.-Ing. habil. Martin Engelien
Dipl.-Inf. Jens Homann (Hrsg.)

Virtuelle Organisation und Neue Medien 2002

Workshop GeNeMe2002
Gemeinschaften in Neuen Medien

TU Dresden, 26. und 27. September 2002



JOSEF EUL VERLAG
Lohmar · Köln

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Virtuelle Organisation und Neue Medien 2002 / Workshop GeNeMe 2002 – Gemeinschaften in Neuen Medien – TU Dresden, 26. und 27. September 2002. Hrsg.: Martin Engeliens ; Jens Homann. – Lohmar ; Köln : Eul, 2002

(Reihe: Telekommunikation und Medienwirtschaft ; Bd. 14)

ISBN 3-89936-007-9

© 2002

Josef Eul Verlag GmbH

Brandsberg 6

53797 Lohmar

Tel.: 0 22 05 / 90 10 6-6

Fax: 0 22 05 / 90 10 6-88

<http://www.eul-verlag.de>

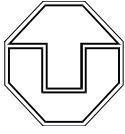
info@eul-verlag.de

Alle Rechte vorbehalten

Printed in Germany

Druck: RSP Köln

Bei der Herstellung unserer Bücher möchten wir die Umwelt schonen. Dieses Buch ist daher auf säurefreiem, 100% chlorfrei gebleichtem, alterungsbeständigem Papier nach DIN 6738 gedruckt.



Technische Universität Dresden
Fakultät Informatik • Institut für Angewandte Informatik
Privat-Dozentur Angewandte Informatik

PD Dr.–Ing. habil. Martin Engelen

Dipl.–Inf. Jens Homann

(Hrsg.)

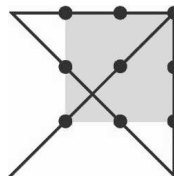


an der

Fakultät Informatik der Technischen Universität Dresden

in Zusammenarbeit mit der
Gesellschaft für Informatik e.V.,
GI-Regionalgruppe Dresden

gefördert von der Klaus Tschira Stiftung
gemeinnützige Gesellschaft mit beschränkter Haftung



am 26. und 27. September 2002

in Dresden

<http://pdai.inf.tu-dresden.de/geneme>

Kontakt: Thomas Müller (geneme@pdai.inf.tu-dresden.de)

B. Konzepte von GeNeMe

B.1. Peer-to-Peer – eine „verteilte Technologie auf der Suche nach einem „zentralen“ Verständnis

Claus Eikemeier,

Prof. Dr. Ulrike Lechner

Fachbereich für Mathematik und Informatik

Universität Bremen

„Ask ten different computer scientists what P2P is, and you may get ten different answers. And chances are, all will be correct. That`s the trouble with grand terms such as P2P ...“, H. M. Fattah [Fattah2002]

1. Einleitung

Seit ca. dem Jahr 2000 tritt eine neue Architektur immer stärker in den Vordergrund: Peer-to-Peer (P2P) Applikationen schicken sich an, eines der neuen IT-Paradigmen der Zukunft zu werden. Der Hype um P2P hat zur Gründung diverser Firmen geführt, die zum Teil mit mehreren Millionen Dollar Venture Capital oder durch andere Unternehmen unterstützt werden (Fattah, 2002, S. 169).

Was ist also Peer-to-Peer? Ist es nur eine Neubenennung von vorhandenen Netzwerken? Oder eine Bezeichnung einer Gruppe von Anwendungen? Welche ökonomischen und soziologischen Einflüsse und Auswirkungen hat diese Architektur bzw. wird sie in der Zukunft haben? Dieses Papier versucht hier Grundlagenarbeit zu leisten. Zunächst werden die zentralen Merkmale von Peer-to-Peer anhand der gängigen Definitionen aus der Literatur erarbeitet (Kapitel 2). Ausgehend von den Funktionen lässt sich die Software in Funktionsklassen einteilen (Kapitel 3). Basierend auf den wesentlichen Merkmalen in den vorhergehenden Kapiteln werden im Kapitel 4 ein Lebenszyklus für P2P Applikationen (Phasenmodell) entwickelt und die wesentlichen Phasen in einem Referenzmodell nochmals genauer untersucht (Kapitel 5). Das letzte Kapitel zeigt ausgewählte soziale und ökonomische Konsequenzen dieser Technologie auf. Es werden vorwiegend Aspekte betrachtet, die für virtuelle und reale Communities relevant sind.

2. P2P – Begriffsbildung

Der Begriff Peer-to-Peer rückte mit Applikationen wie Gnutella und Napster in den Fokus der Diskussion. Auch verschiedene andere Applikationen werden dem Bereich der P2P-Applikationen zugerechnet und sicherlich ist der Begriff Peer-to-Peer stark durch die Wahrnehmung der bekanntesten Vertreter dieser Architekturgattung geprägt. Nachdem diese Applikationen nicht nur aus technischer Hinsicht neu sind, sondern vor allem auch wirtschaftliche und soziologische Konsequenzen haben und für die User ein neues Kommunikationsverhalten bedeuten (Interaktion in einer Gemeinschaft – statt Berieselung durch Massenmedien), soll hier nun aus den verschiedenen Sichtweisen von Peer-to-Peer ein zentrales Verständnis dieses Begriffes entwickelt werden.

2.1 P2P als Architektur

Eines der wesentlichen Unterscheidungsmerkmale, das P2P auszeichnet ist die neue Architektur der Infrastruktur.

Das O`Reilly Network (OREillyNetworkDef) gibt eine sehr einfache Definition:

„Peer-to-Peer describes a decentralized network in which all computers function as equals.“

Diese Definition beschreibt ein P2P Netzwerk als ein Netzwerk von „Peer“-Knoten (Computern) mit gleichen Rechten und Pflichten. Viele Applikationen, wie Napster, SETI@home, eDonkey2000 gehören dieser Definition entsprechend nicht in die Klasse der P2P-Applikationen bzw. Netzwerke – dagegen wären Sendmail, Usenet und andere klassische Applikationen des Internet entsprechend (Harnessing the Power of Disruptive Technology) als P2P anzusehen. „Equal“, d.h. „gleich“, sind auch in Applikationen wie Gnutella niemals alle Clients, selbst wenn die gleiche Software läuft unterscheiden sich doch die Clients in der Qualität der Netzwerkverbindung, dem für die Software verfügbaren Speicherplatz, durch die Rechenleistung oder durch die verfügbaren Inhalte.

Die Ressourcenverteilung ist ein weiteres architektonisches Merkmal, das P2P von anderen Applikationen zu unterscheiden scheint. Schollmeier (Schollmeier, 2001) betrachtet ausschließlich die Ressourcen:

„Eine verteilte Netzwerkarchitektur kann P2P Netzwerk genannt werden, wenn die Teilnehmer einen Teil ihrer eigenen Ressourcen (Rechenleistung, Speicherkapazität,

Verbindungsbandbreite, Drucker, etc.) zur Verfügung stellen. Diese geteilten Ressourcen sind notwendig, damit das Netzwerk den Dienst oder Inhalte zur Verfügung stellen kann (Beispiel.: gemeinsame Dateinutzung, gemeinsame Arbeitsdokumente). Auf die Ressourcen kann von jedem Peer direkt zugegriffen werden. Somit sind die Teilnehmer des P2P-Netzes gleichzeitig sowohl Anbieter als auch Nutzer von Ressourcen (Dienste, Inhalt): sie werden als Servent bezeichnet. “ (Schollmeier, 2001)

Auch diese Definition schließt – insbesondere bei Betrachtung des letzten Teils – Applikationen wie SETI@Home (SETI@Home) oder das Intel Philanthropic P2P-Programm (Intel, 2001) nicht mit ein.

Ein weiterer Ansatz definiert P2P über Eigenschaften der Netzwerke. Peers kommunizieren unmittelbar untereinander und das ganze System von Knoten organisiert sich selbst. Shirky definiert über Netzwerkeigenschaften und Autonomie der Knoten P2P Applikationen (Shirky, 2000):

„ Zu testen ist:

Wird standardmässig die wechselnde Anbindung der Geräte angenommen und werden die Netzwerkadressen normalerweise nur temporär vergeben? und

Wird den Aussenknoten des Netzwerkes eine erhebliche Eigenverantwortung zugesprochen?

Können beide Fragen mit „Ja“ beantwortet werden, so gehört die Applikation zur Gruppe der P2P Applikationen, wird auch nur eine mit „Nein“ beantwortet, gehört sie nicht dazu.“

Bei vielen Applikationen wie Gnutella oder Napster wechseln die Topologien der Netzwerke dynamisch – für den Austausch von Files bauen sich neue Beziehungen zwischen Knoten auf. Die Knoten entscheiden autonom – wie sich im Netzwerk beteiligen und welche Beziehungen sie eingehen. Die meisten P2P-Applikationen müssen außerdem initial mit Kommunikationspartnern versorgt werden, so dass sie das Netzwerk und mit ihm mehr Kommunikationspartner finden.

Im folgenden soll Dynamik der Topologie als wesentliche Eigenschaft von P2P noch näher erläutert werden: Die *Server*, d.h. die Computer, auf denen die Inhalte bereitgestellt werden, sind leistungsfähige Rechner (hohe Rechenleistung, großer Plattenspeicher) mit der Eigenschaft, dauerhaft (24/7) und insbesondere über die gleiche Adresse im Netzwerk erreichbar zu sein. Die *Clients*, welche die Schnittstelle zum

Benutzer darstellen, erhalten häufig erst beim Aufbau der Verbindung zu ihrem Provider eine i.d.R. wechselnde IP-Adresse zugeteilt. Auch das Domain Name System (DNS), welches die Umsetzung aus der symbolischen Adresse in die IP-Nummer durchführt, kann die dadurch entstehenden Probleme nicht ausgleichen. Sollte ein Client-Rechner als Anbieter von Informationen dienen, so wäre folglich nicht klar, wie man auf ihn zugreifen kann. Fattah (Fattah, 2002, Seite 19) (beruft sich dabei wiederum auf eine Aussage von Clay Shirky) bezeichnet somit einen Charakter von P2P-Systemen mit: das P2P-System muss es ermöglichen, dass sich die beteiligten Rechner jederzeit und auch aus dem herkömmlichen Internet heraus verbinden können. Abhilfe gegen diese dynamischen Topologien kann nur unter Umgehung des etablierten DNS, d.h. eines nicht auf IP-Nummern basierenden Adress-Konzeptes, erreicht werden.

Beispiele dafür sind ICQ bzw. JXTA. Bei der ersten Anmeldung am ICQ-System wird durch eine zentrale Stelle eine weltweit eindeutige Nummer vergeben, die ab dem Moment den Benutzer identifiziert. Sie ist der Schlüssel (ID) zu dem zentral gespeicherten Benutzerprofil und kann natürlich beim Wechsel des Gerätes (z.B. bei Wechsel des Betriebssystems oder wenn man sowohl mit PC als auch mit einem PDA arbeitet) mitgenommen werden. Im Projekt JXTA verwendet man für die Identifizierung aller Einzelkomponenten, die keinerlei weiterer Beschränkung unterliegen, eine eindeutige ID (zufällig erzeugt u.a. über Hashing) (vgl. SunJXTA, z.B. Programming Guide). Peers und auch Komponenten innerhalb der Peers lassen sich somit ohne direkte Verwendung der zugehörigen IP-Adresse ansprechen. Insbesondere wird die Verwendung von Handys und anderen nicht TCP/IP-fähigen Zugangsgeräten dadurch ermöglicht.

Zusammenfassend lässt sich feststellen, dass die Kriterien Ressourcenbeitrag, Dynamik der Topologie und Autonomie der Knoten bei vielen als P2P bezeichneten Applikationen zutreffen – die Gleichheit der „Peers“ jedoch ist – wie oben argumentiert – zweifelhaft und kann wohl in den seltensten Fällen wirklich angenommen werden.

Entsprechend wird der Begriff Peer-to-Peer in Bezug auf die Forderung nach „Gleichheit“ in der Literatur verfeinert. Man unterscheidet reine und hybride Peer-to-Peer-Systeme. So definiert Schollmeier (Schollmeier, 2001):

„Eine verteilte Netzwerk-Architektur wird als „reines“ (pure) P2P Netz bezeichnet, wenn es einerseits ein P2P Netzwerk ist und andererseits jedes willkürlich ausgewählte

Element aus dem Netz entfernt werden kann, ohne dass das Netzwerk mit dem angebotenen Dienst darunter leidet.“

Die Forderung nach der Willkürlichkeit des zu entfernenden Elementes kann von einigen Systemen nicht erfüllt werden. Sie werden als „hybride“, d.h. gemischte P2P Systeme bezeichnet.

„Eine verteilte Netzwerk-Architektur wird als „hybrides P2P“ bezeichnet, wenn es zum einen ein P2P-Netzwerk ist und andererseits aber eine zentrale Komponente zur Durchführung des angebotenen Dienstes notwendig ist.“ (Schollmeier, 2001)

Napster (NapsterProtokoll), eDonkey2000, Groove (GrooveNetworks) und andere Applikationen fallen unter die Kategorie hybrider P2P-Systeme. Gnutella und viele auf dem Gnutella Protokoll basierende Applikationen gehören der Kategorie der reinen P2P-Systeme an. Wie oben argumentiert sind auch die Knoten nicht gleich, sie unterscheiden sich durch die Inhalte und ihren Beitrag an Ressourcen. Letzteres soll als eine charakterisierende Eigenschaft von P2P angenommen werden. Weitere Eigenschaften betreffen die Verfügbarkeit von Komponenten und damit den Aufbau des Netzes (dynamische Topologie) und die Autonomie der Knoten. In Bezug auf die Austauschbarkeit der Knoten werden hybride und reine Peer-to-Peer Systeme unterschieden.

2.2 P2P als Architektur für die Interaktion

Neben den technischen Ansätzen zur Beschreibung von P2P-Systemen gibt es auch eine Reihe von Ansätzen, P2P über die Interaktion zwischen Menschen zu charakterisieren. So tragen auch in virtuellen Gemeinschaften die Mitglieder für den Wert der Gemeinschaft entscheidende Ressourcen bei (Hummel und Lechner, 2002; Lechner und Schmid, 2001). Virtuelle Gemeinschaften entwickeln dynamische Beziehungsnetzwerke und die Mitglieder dieser Netzwerke sind autonom (Hummel 2002). Somit erfüllen Virtuelle Gemeinschaften ebenfalls die wesentlichen Eigenschaften von P2P-Architekturen (vgl. Kap. 2.1). Auch Napster hat sich als der Welt größte Gemeinschaft von Musikliebhabern verstanden (napster.com).

Schoder und Fischbach beschreiben, dass Virtuelle Gemeinschaften durch P2P-Systeme unterstützt werden (Schoder und Fischbach, 2002).

Als wesentliche Merkmale von P2P in diesem Kontext sind die gemeinsame Nutzung der vorhandenen bzw. beigetragenen Ressourcen und eine Beschleunigung von Informationsaustauschprozessen zu nennen. Insgesamt ermöglicht das P2P-Paradigma neuartige Arbeitsumgebungen für effiziente Kommunikation und Kollaboration innerhalb von virtuellen Gemeinschaften.

Das Internetlexikon WhatIs (whatis.com) definiert P2P über die in diesen Netzwerken typischen Austauschbeziehungen bzw. Interaktionen:

„P2P ist ein Kommunikations-Modell bei dem jeder Teil die gleichen Möglichkeiten hat und wo jede Partei eine Kommunikation eröffnen kann. ... In einigen Fällen wird die P2P-Kommunikation dadurch ermöglicht, dass jedem Kommunikations-Knoten sowohl Client- als auch Serverfunktionalität gegeben wird. ...“

Aber auch andere soziologische und organisatorische Aspekte von P2P scheinen bemerkenswert: P2P kennt innerhalb der Systeme (bisher noch) keine unterschiedlichen Rechte, d.h. ein Peer, der sich an einem Netzwerk anmeldet, hat die gleichen Rechte wie jeder andere Peer auch. Da die eigentliche Kommunikation (im Gegensatz zur Suche des Kommunikationspartners) nahezu ausnahmslos ohne Verwendung von zentralen Komponenten durchgeführt wird, lässt sich auch eine gewisse Anonymität erreichen. Die Nachrichten werden unter Umgehung zentraler Kontrollstellen übermittelt. Insgesamt ergibt sich eine Machtverschiebung von den zentralen Stellen im Zentrum bestehender Systeme (z.B. dem Betreiber des Servers) hin zum Rand des Netzes. Dieses bedeutet eine Verschiebung der Macht und Kontrolle von der Organisation hin zu den Individuen (Anderson, 1998, S. 76, Hummel und Lechner 2002).

Technologische Sicht und Interaktionssicht stehen dabei in enger Wechselwirkung. Hierfür seien zwei Beispiele genannt: zum einen die i.d.R. asymmetrische und damit für P2P-Anwendungen unpassende Struktur der „letzten Meile“, d.h. des Benutzer-Netzzugangs, und zum anderen die Besitzverhältnisse der am System beteiligten Geräte.

War bis jetzt der Internet-Nutzer nur Konsument von Informationen, so wird er in P2P-Systemen meist auch Anbieter (teilweise sogar Produzent) dieser (Shirky, 1998).

Napster, Gnutella und seine Clones, FreeNet, u.a. erweitern diesbezüglich die Möglichkeiten des Rechners, auf dem sie laufen. Neben dem „normalen“ Download von Daten von anderen Rechnern bietet der eigene Rechner diese Daten in der Regel dann auch an. Ideal wäre somit eine schnelle Download- und eine schnelle Upload-Verbindung. Heutige Anbindung an das Internet bieten jedoch meist asymmetrische

Verbindungen an, d.h. regelmäßig gibt es neben einer schnellen Downloadverbindung nur eine erheblich langsamere Upload-Verbindung. P2P-Systeme können daher auch über die Notwendigkeit einer symmetrischen Verbindung an das Internet identifiziert werden (quasi-notwendiges Kriterium).

Shirky gibt neben den oben genannten Testfragen als Thema für eine andere Prüfung die Frage des Besitzstandes der beteiligten Rechner an: die Frage, wem die Hardware gehört, auf der der Service läuft (Shirky, 2000), zeigt im Falle von P2P zu den Nutzern des Dienstes selber. Genau wie beim Client/Server-Modell sind bei P2P-Architekturen die Besitzer der Hardware für die Wartung des Systems zuständig. Allerdings wird der Dienst nun durch eine große Zahl an Benutzern gewartet. Fattah (Fattah, 2002, S. 19) bemerkt pointiert: „Napster hat 40 Millionen unbezahlte Netzwerkadministratoren, die sich um das Netzwerk kümmern“. Diese Eigenschaft unterscheidet natürlich die P2P-Anwendungen von herkömmlichen.

2.3 Zusammenfassung – Der Versuch eines einheitlichen Verständnisses

Technologische Sicht auf Netzwerke und die soziale Sicht auf die Interaktionsbeziehungen sind kongruent – sowohl aus Netzwerksicht bzw. Applikationssicht als auch aus Sicht der virtuellen Gemeinschaften lassen sich folgende drei Merkmale als charakteristisch aufzeigen:

- Beitrag der Ressourcen durch die Agenten¹,
- Dynamische Topologie der Verbindungen,
- Autonomie der Agenten.

Unterschieden werden in Bezug auf die Austauschbarkeit der Knoten hybride und reine Peer-to-Peer-Systeme.

¹Agenten sollen hier als autonome organisatorische Einheiten verstanden werden. Agenten können demnach sowohl Menschen als auch Softwareartefakte sein.

3. Funktionsklassen in P2P

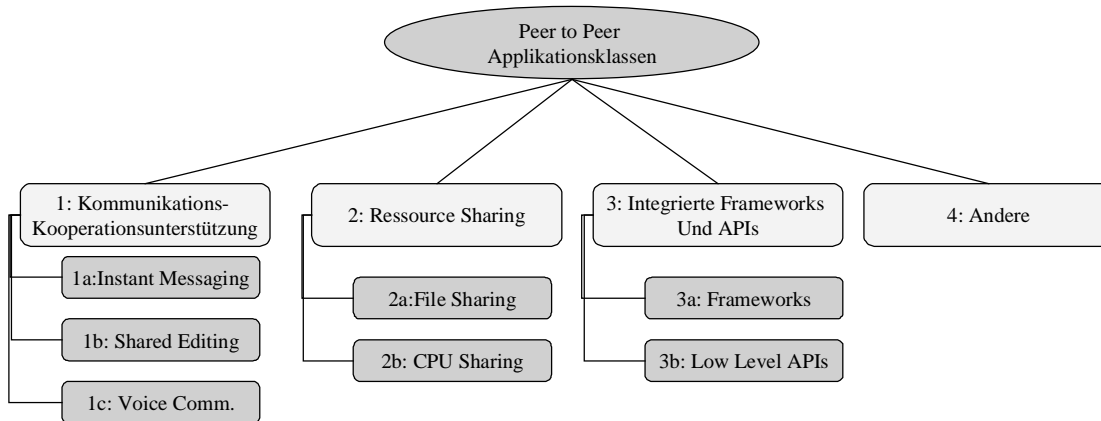


Abb. 1: Applikationsarten in P2P-Systemen

Im allgemeinen werden für eine effiziente Unterstützung von Communities mehrere Applikationen - auch aus unterschiedlichen Klassen - verwendet. Heutige Systeme integrieren daher z.B. neben einer File-Sharing-Applikation (Klasse 2a) auch häufig eine Kommunikationsunterstützung (Klasse 1a) oder eine Community Newsgroup (Klasse 1b). Je nach Art der Programmierung ergibt sich damit auch z.T. ein Zugriff auf schon bestehende Systeme, wie z.B. das ICQ-Netz. Hieraus ergeben sich positive Netzwerkeffekte.

Es ist nicht auszuschließen, dass sich weitere P2P-Systemklassen herausbilden. Es wird diskutiert, ob z.B. Multi-User Games eine eigene Klasse bilden können (siehe z.B. OpenSkies). Derzeit wird jedoch noch für Zugangskontrolle und Ergebnisspeicherung jeweils eine zentrale Instanz verwendet. Abhängig von der Durchführung der Kommunikation zwischen den Spieler-Clients könnte eine hybride - ähnlich wie in Napster - oder eine reine P2P-Struktur genutzt werden.

Die Funktionalitäten unterscheiden die verschiedenen P2P-Applikationen. Sie lassen sich - zumindest entsprechend ihrer wesentlichen Anwendung - in die oben genannten vier Hauptgruppen einteilen. Dabei weisen die verschiedenen Arten durchaus unterschiedliche Ausprägungen der charakteristischen Merkmale auf (vgl. Kap 2). Im File Sharing werden andere Ressourcen beigetragen als bei einem P2P-Kommunikations- oder -Kollaborationssystem. Auch die Topologie und die Dynamik unterscheiden sich: File Sharing als „primitive Anwendung“ erfordert weit weniger an stabilen sozialen Strukturen als z.B. komplexe Transaktionen (Hummel, 2002, siehe

auch Kap. 6). Mobile Anwendungen strukturieren und verbreiten sich eher entlang bestehender sozialer Strukturen und weisen daher eine andere Topologie und Dynamik

<i>Nr.</i>	<i>Programm</i>	<i>Funktionalität</i>
Kommunikation (1a-c)		
1	ICQ	Instant Messaging (IM), 1to1-Chat, Aktivitäts-Monitor,
2	MSN Messenger	IM, 1to1-Chat, Shared Whiteboard, File Sharing, Shared Screen
3	AOL Instant Messenger	IM, Chat,
4	Jabber	XML-basierte Kommunikations-Middleware auf Client/Server-Basis; ermöglicht, eigene Komm.-Strukturen zu kreieren,
Ressource Sharing: FILE (2a)		
5	Napster	Datei-Suche, Zentraler Server mit Indexinformation, Datei-Austausch als P2P-Kommunikation
6	Gnutella	Datei-Suche, Reines P2P, d.h. sowohl Suche als auch Austausch per P2P-Kommunikation, Starthosts: gnutellahosts.com
7	FreeNet	Datei-Suche über Index, Dateien verschlüsselt, in Teile geteilt und verteilt gespeichert, echter Speicherort unbekannt, Autor ggf. anonymisiert, selten angeforderte Dokumente „verschwinden“
8	eDonkey2000	Dateitauschbörse, eigene dezentrale Server-Organisation
9	FreeHaven	Fokus auf Speicherung von Dateien (Persistenz), Anonymität
10	LimeWire	Gnutella-Clone, neues Frontend, vordefinierte Startknoten
11	Morpheus	Gnutella-Clone, mit weiteren Diensten wie Chat (u.a. zu AIM), Buddylist, etc. , Bandbreitensteuerung, verbesserter Performance
12	AudioGalaxy	Napster-artige Musiktaschbörse, wenig Bedeutung, da rechtliche Probleme, download wurde zum Teil mit Spy-Software verbunden
13	KaZaA	Client für Gnutella-Netzwerk mit erheblich verbesserter Performance (durch ressourcenbestimmte Selbstorganisation); findet derzeit starke Verbreitung
Resource Sharing: CPU (2b)		

<i>Nr.</i>	<i>Programm</i>	<i>Funktionalität</i>
9	SETI@home	Verteilte Signal-/Daten-Verarbeitung, Client-/Server-App. Jedoch wechselt die Funktion des Servers bei der Berechnung auf den Client-Rechner (zentrales Programm fragt Rechenergebnis beim ehemaligen Client, nun Server ab)
10	Parabon (Frontier SW-Paket)	Verteilte Datenverarbeitung (Krebsforschung,, nur teilweise non-profit), mit Entwicklungspaket für eigene Anwendungen
11	Intel Philantropic P2P Programm	Basisprogramm, Einsatz z.B. für Krebsforschung, Optimierung
Framework (3a) und Low Level API (3b)		
12	Groove	Plattform, Erweiterbar durch eigene Funktionen
13	JXTA	Offene Protokoll-Sammlung für P2P-Programme, Implementierung u.a. in Java, Projekte werden von Community gepflegt
Weitere (4)		
14	FASD	Such-Layer für FreeNet; vereinfacht Umgang mit FreeNet-IDs
15	CHORD	Optimierte Suchapplikation für P2P-Basisapplikation
16	Content Addressable Network (CAN)	allg. Suchlayer für optimiertes Retrieval von Dokumenten u.a. in P2P-Netzen (ermöglicht effiziente Scalability von P2P Netzen)

Tab. 1: Übersicht über Applikationsklassen und Beispiele (Auswahl)

auf als offene Peer-to-Peer Systeme für das File Sharing, die üblicherweise für fest installierte Knotenrechner gedacht sind.

Neben der Funktionalität als wesentlichem Merkmal von Peer-to-Peer-Applikationen lassen sich auch die Phasen innerhalb eines Lebenszyklus entsprechend ihrer Implementierung in einem tradierten oder P2P-Paradigma unterscheiden. Das soll im folgenden Kapitel untersucht werden.

4. P2P im Lebenszyklus

P2P kann ganz unterschiedliche Ausprägungen haben – dies wurde bereits im zweiten Kapitel dieses Artikels aufgezeigt. Die Art der Anwendung charakterisiert P2P – in welchen Aspekten einer Technologie bzw. einer Gemeinschaft sich Merkmale von P2P

wiederfinden lassen, ist von Applikation zu Applikation unterschiedlich. Im folgenden werden fünf Phasen im Lebenszyklus unterschieden.

Phase 1: Implementierung und Erstinstallation des Systems

In der Implementierung unterscheiden sich die P2P-Applikationen. Einige werden dem P2P-Paradigma in der Implementierung folgend von einer Vielzahl von Beteiligten in einem Open-Source-Prozess implementiert, andere werden von einer zentralen Stelle implementiert und bereitgestellt. Offene Protokolle, wie z.B. Gnutella, unterstützen einen P2P-gemäßen, verteilten Ansatz der Implementierung mit Ressourcenbeiträgen von vielen Einzelnen. Gnutella's Implementierung kann als P2P-Ansatz verstanden werden – Napster hat sich im Laufe des Lebens von einem tradierten zum einem offenen Ansatz gewandelt – auch für Napster waren gegen Ende seiner Existenz verschiedene Clients verfügbar und das Protokoll war offengelegt (NapsterProtokoll). Entsprechend sind dann auch verschiedene Server entstanden.

Diese Phase ist bei vielen der P2P-Applikationen ähnlich. Ausgehend davon, dass heute die überwiegende Zahl der Programme auf dem Betriebssystem Windows mit seinen de facto Vorgaben basieren, wird die Erstinstallation des Programms durch einen Wizard unterstützt. Der Wizard erzeugt u.a. Speicherstrukturen als Speicherort für heruntergeladene Daten und gleichzeitig zur Ablage von Upload-Dateien.

Phase 2: Start der Applikation

Viele reine P2P-Applikationen benötigen für den Start mindestens einen ersten Kontakt zu anderen Peers im vorhandenen Netz. Dafür wird i.d.R. eine Anzahl von Peer-Adressen bei der Installation mitgeliefert. Somit ist dem System ein Zugang zum Netzwerk als Default bekannt und bietet dem Nutzer die Möglichkeit, sofort mit dem System, ohne weitere Angaben machen zu müssen, zu starten. Die Rechner der mitgelieferten Adressen werden meist von den Bereitstellern der Software betrieben; häufig sind es keine vollwertigen Peers, sondern stellen nur die Systemanbindung sicher (d.h. bieten z.B. keinen Speicherplatz an). Ein Ansatz, bei dem nicht ein zentraler Einstiegspunkt ins Netz vorhanden sind – sondern viele, die auch dynamisch ändern können, entspricht dem P2P Paradigma.

Phase 3: Auswahl der Kommunikationsverbindung („search“-Phase)

Diese dritte Phase ist meist der „interessanteste“ Teil der Applikation. Hier wird der spätere Kommunikationspartner, d.h. das Peer, mit dem kommuniziert werden soll,

ausgewählt. Die Applikationen unterscheiden sich stark darin, wie diese Phase tatsächlich ausgestaltet ist. Die bekanntesten Implementierungen nutzen eine von zwei Such-Methoden: entweder wird die Suchinformation von einem zentralen Indexserver (z.B. Napster, ICQ, eDonkey2000) geladen oder die Applikation verbreitet die Anfrage lawinenartig in ihrer Umgebung (flooding, verwendet im Gnutella-Protokoll).

Phase 4: Aufbau und Nutzung der Verbindung („load“-Phase, Nutzung)

Man kann diese Phase auch als Arbeit mit dem System bezeichnen, da hier die Nutzdaten ausgetauscht werden. Dabei kann es sich sowohl um Dateien handeln oder auch um das Senden- und Empfangen von Kommunikationsnachrichten, z.B. im Rahmen eines Chats. Diese Phase soll im folgenden Kapitel nochmals untersucht werden.

Phase 5: Übergang in den Ruhemodus, Hintergrundarbeit der Applikation

Neben der eigentlichen „Arbeit“ der Load-Phase ist bei P2P auch das „Mitglied sein im Netzwerk“ interessant. Die Knoten sind als Teil des Netzes auch dann aktiv und bieten Dienste an, wenn sie nicht für den Auftraggeber handeln. Solche Servents können dann Aufgaben im Routen von Nachrichten, im Beantworten von Anfragen (Queries) und im Bereitstellen von Informationen oder Daten übernehmen. So melden sich Systeme z.B. beim zentralen Server mit dem dynamischen Index und zeigen ihre Aktivität an bzw. wie sich ihr Datenbestand verändert hat. Auch andere Arten von Statistik können nun problemlos bearbeitet werden. Dieses passive Bereitstellen von Diensten und das Verfügbarsein im Netz sind wesentliche Bestandteile von P2P. Applikationen, die gleichermaßen Client und Server im Netz sind und die aktiv Teil des Netzes bleiben, auch wenn sie gerade selbst nicht arbeiten, entsprechen einem P2P-Paradigma. Applikationen, die einladen, sie zu schließen und nicht aktiv offen zu lassen, entsprechen dagegen dem P2P-Paradigma weniger. Applikationen, die im Hintergrund arbeiten, Aufgaben (Tasks) wie z.B. das Herunterladen von Dateien autonom schedulen und die auf passende Interaktionspartner warten, sind somit erfolgversprechender für den Aufbau eines guten P2P-Netzes als Applikationen, die ohne unmittelbaren Komfortverlust vom User sofort nach Erledigung einer Aufgabe geschlossen werden können. Sie laden nicht dazu ein, das Servent-Prinzip beim einzelnen User zu implementieren.

Die Betrachtung unterschiedlicher Systeme zeigt nun auf, dass im Falle von Napster die Phase 3 nicht reines P2P beinhaltet (Anfrage bei zentralem Napster-Server), bei Verwendung eines Gnutella-Clients (z.B. LimeWire) jedoch schon (Flooding in die

Nachbarknoten). Die Phase 4 unterscheidet sich jedoch nicht, d.h. in der Phase 4 wird in beiden Programmen das P2P-Paradigma implementiert. Die Betrachtung dieser zwei Phasen erlaubt eine Charakterisierung der gesamten Applikation, da die weiteren Phasen nur wenig zur Funktion selber beitragen.

Im folgenden sollen nun die unterschiedlichsten und für die eigentliche Arbeit solcher Systeme relevanten Komponenten herausgegriffen werden und in einem eigenen Modell vertieft betrachtet werden.

5. Organisation der Interaktion

Nachdem die Applikationen entsprechend ihren Funktionalitäten unterschieden wurden und die verschiedenen Phasen des Lebenszyklus von P2P-Applikationen unterschieden wurden, wird nun der Fokus der Betrachtung auf die Search- und die Load-Phase gelegt. Im folgenden sollen die verschiedenen Arten der Interaktion sowie die verschiedenen Sichtweisen der Beschreibung für die Charakterisierung der einzelnen Arten von Interaktion innerhalb von P2P-Applikationen unterschieden werden. Als Rahmenmodell wird dazu das Medienreferenzmodell (Schmid, 1999) verwendet.

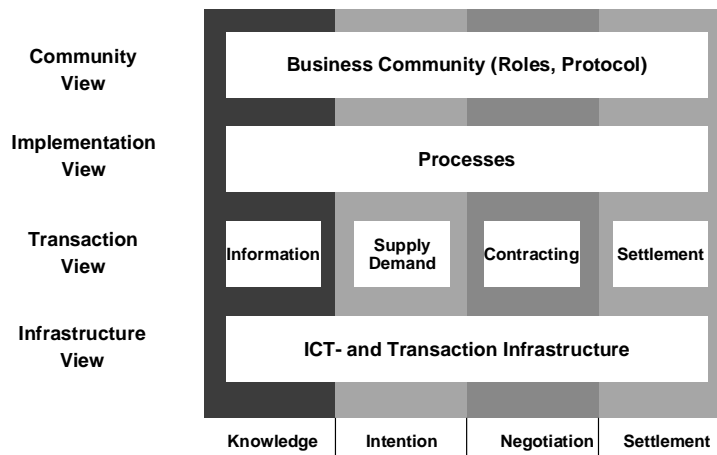


Abb. 2: Das Medien-Referenz-Modell (Schmid, 1999)

Das Medienreferenzmodell (MRM) MRM ist in Abbildung 2 dargestellt und wird nachfolgend vorgestellt.

Die *Phasen* (Spalten im MRM, v.l.n.r.) bilden die logisch notwendigen Schritte ab, die zur Abwicklung einer Transaktion notwendig sind (Schmid, 1999).

In der *Wissensphase (Knowledge Phase)* wird assertorisches Wissen ausgetauscht, d.h. bedingtes oder unbedingtes Tatsachenwissen.

In der *Absichtsphase (Intention Phase)* werden konkrete Austauschabsichten (*Supply and Demand*) direkt ausgetauscht bzw. ermittelt und ausgetauscht. Dienste dieser Phase sind z.B. elektronische Produktkataloge.

In der *Vereinbarungsphase (Negotiation Phase)* findet die Verhandlung statt, die im Erfolgsfall mit einem Vertrag endet. Kommuniziert werden Angebote, Gegenangebote und Annahme oder Ablehnung von Angeboten. Es werden die Handlungen und Austauschbeziehungen zwischen den Mitgliedern der Gemeinschaft verhandelt, formalisiert und als Kontrakte externalisiert.

In der *Abwicklungsphase (Settlement Phase)* werden die in den Kontrakten spezifizierten Leistungen erbracht. In dieser Phase wirken die güter- und finanzlogistischen Transaktionen mit ihren unterschiedlichen Prozessen und Dienstleistern.

Die *Sichten* (Reihen, v.u.n.o.) des Referenzmodells unterscheiden

Die IKT- oder Infrastruktursicht, welche die physische Plattform als Basis der Austauschbeziehungen beschreibt.

Die Dienste, die, nach den Phasen unterschieden, in einem Medium die Verarbeitung von Informationen und den Austausch von Information übernehmen.

Die Prozesssicht, in der die Prozesse der Leistungserstellung detailliert werden.

Die Community- oder Gemeinschaftssicht mit der sozialen Textur eines Mediums. Zu dieser sozialen Struktur gehören die digitalen Repräsentationen der Mitglieder einer Gemeinschaft und ihrer Rollen in der Gemeinschaft und die Regeln (Protokolle) für die verschiedenen Austauschbeziehungen. Die Prozesse der Prozesssicht implementieren diese Protokolle.

Dieses Rahmenwerk wird nun angewendet, um die Peer-to-Peer Applikationen zu charakterisieren.

Infrastruktursicht: Die Infrastruktur einer Peer-to-Peer-Applikation kann, muss aber nicht dem Peer-to-Peer-Paradigma entsprechen (vgl. auch Kapitel 2 – die Gemeinschaft kann P2P sein). So können die Identifikation der Peers innerhalb des Netzwerks, die Verbindungen der Peers und vor allem die Verteilung der verschiedenen Ressourcen Peer-to-Peer-Prinzipien entsprechen, so dass die Identifikatoren nicht zentral vergeben werden, die Ressourcen jedoch gleichmäßig verteilt bzw. beigetragen werden, und auch

der Informationsaustausch zwischen den Knoten einem Peer-to-Peer-Prinzip folgt. Unterschiede gibt es hier z.B. zwischen Freenet, Gnutella und Napster. Bei Napster wird der Identifikator für den Client zentral vergeben, Gnutella stützt sich auf die Domain-Adressierung des Domain Name Service (DNS). Dieser ist aber hierarchisch organisiert und verwaltet. Freenet geht in der Identifizierung von Knoten ein Stück weiter – hier können die Namen von Knoten und auch von Informationsobjekten (Files) dezentral erzeugt werden. Der Mechanismus von Freenet entspricht damit am stärksten dem P2P-Paradigma.

Dienste: Einzelne Dienste für die Verarbeitung oder den Austausch von Information können unterschiedlich dem P2P-Paradigma entsprechen. Information kann vollkommen dezentral organisiert werden, genauso wie der Zugriff auf diese Informationen. So bieten sogenannte Ameisenalgorithmen die Möglichkeit, Informationen dezentral auf einer Peer-to-Peer-Infrastruktur abzulegen und sie auch – ohne zentrale Kontrolle oder Kenntnis der Struktur – wiederzufinden. Auch bei Freenet ist die Organisation von Information dezentral einem Peer-to-Peer-Prinzip folgend organisiert. Die wesentlichen Applikationen im Peer-to-Peer computing betreffen den Austausch von digitalisierter Information, während eine Verknüpfung mit Profilen oder Organisationsstrukturen von Information kaum Verwendung findet.

Prozesssicht: Prozesse können einem tradierten zentralistischen genauso wie einem dezentralistischen Design entsprechen. In einem zentralistischen oder hierarchischen Paradigma liegt die Kontrolle der Aktionen bei einer Stelle – in einem dezentralistischen, Peer-to-Peer-System gleichmäßig bei allen involvierten Instanzen. Die Suche ist in Gnutella entsprechend einem Peer-to-Peer-Paradigma organisiert: alle Knoten partizipieren auf ähnliche Weise. Dabei entsprechen dezentral organisierte Prozesse im Grunde genommen Multi-Agentensystemen in denen die Agenten autonom handeln, um ein gemeinsames Ziel zu erreichen. Beschränkungen sind der dezentralen Organisation durch Sicherheitsaspekte vorgegeben: Mobiler oder selbstreplizierender Code kann ein entsprechendes Sicherheitsrisiko in sich tragen.

Community-Sicht: Auf der Community-Ebene gibt es Modelle, in denen einzelnen Agenten eine zentrale Bedeutung zukommt. Zu dieser Klasse von Modellen gehören alle Applikationen mit zentralisierten Index Servern, aber auch Anwendungen wie Groove, in denen der Server (Groove Relay Server) wesentlich für den Austausch von Daten und die Koordinaten ist. Diese Server zeichnen dann auch ihre Betreiber als wesentliche Mitglieder einer Gemeinschaft aus. Einer Gruppe von Knoten entspricht damit eine Community von „Eigentümern“ oder Betreibern dieser Servents. Genau wie

die Servents kommt auch den Mitgliedern der Gemeinschaft zunächst eine gleiche soziale Position zu. Unterschiede ergeben sich durch die Beiträge der Einzelnen. So zeigen verschiedene Arbeiten, wie z.B. (Adar und Hubermann, 2000), auf, wie das soziale Momentum von Peer-to-Peer-Architekturen wirkt: In Bezug auf die Inhalte ergibt sich auf der Peer-to-Peer-Architektur eine „klassische“ hierarchische und zentralistische Organisation. Nur wenige mächtige Server bedienen wenige Clients. Dabei zeigt sich als Paradoxon des Peer-to-Peer-Paradigmas, dass die Beiträge die Anbieter („Geber“) im Endeffekt über negative Feedback Effekte bestrafen (Hummel und Lechner, 2002). Der Beitrag des Einzelnen zieht als Konsequenz nach sich, dass die beigetragenen Ressourcen stark belastet werden – jeder Download kostet denjenigen, der Ressourcen zur Verfügung stellt weitere Ressourcen. Beheben lässt sich dies durch eine Verteilung von Informationsobjekten oder durch eine soziale Struktur der Gemeinschaft. Dieses soziale Gefüge kann denen, die aktiv beitragen eine Bühne für Bestätigung und Anerkennung bieten (Hummel 2002) und so für eine Kompensation für Beiträge sorgen. Eine soziale Struktur kann auch als eine Kontrollinstanz wirken, die darauf achtet, dass „Geben“ und „Nehmen“ ausgeglichen sind. Hier lassen sich verschiedene Arten von Applikationen unterscheiden. Die ersten, reinen Peer-to-Peer-Systeme hatten keinerlei solche sozialen Strukturen. Beispiel ist Gnutella – in all diesen Applikationen sind darüber hinaus die Mitglieder mehr oder weniger anonym, so dass sich auch keine soziale Struktur bilden kann. In Napster gab es Ansätze, das Feedback zu managen: die Bandbreite die für Downloads zur Verfügung gestellt wurde, konnte gemanagt und vor allem nach oben hin festgelegt werden. Applikationen wie Napster oder eDonkey2000 machen den User wiedererkennbar.

Die oben aufgezählten Peer-to-Peer Systeme sind offen – sie sprechen User an, die sich mit dem Download der entsprechenden Software dem Netz anschließen können. Sie gehen damit zunächst von einem schwachen sozialen Zusammenhalt einer Gemeinschaft aus.

Gerade mobile Applikationen und Peer-to-Peer-Systeme gehen hier einen anderen Weg. Die soziale Struktur der Peer-to-Peer-Gemeinschaft formt sich auf bestehenden sozialen Strukturen und liegt abgebildet in den Buddy Lists oder anderen Systemen von Repräsentationen existierender Kommunikationspartner (vgl. auch (Pennock et al., 2002)). Hier wird die Peer-to-Peer-Gemeinschaft auf einem bestehenden sozialen Netz aufgebaut. Diese Systeme sind für den Austausch von nur vergleichsweise wenigen Nutzern, die einen sozialen Verbund bilden, gedacht. Es gibt hierbei – im Gegensatz zu den oben erwähnten offenen „asozialen“ Peer-to-Peer-Gemeinschaften – positive und

negative Netzwerkeffekte, wie sie zur Bildung einer sozialen Gemeinschaft notwendig sind (Hummel 2002, Hummel und Lechner, 2002).

6. Soziale und ökonomische Aspekte des P2P Einsatzes

Will man das Phänomen P2P umfassend begreifen, so darf es nicht bei der Beschreibung der Anwendungen bleiben – man muss sowohl soziale als insbesondere auch die ökonomischen Aspekte betrachten. Obwohl ein Teil der sozialen Effekte schon im vorhergehenden Abschnitt angesprochen wurde und hier weitere genannt werden, wird auch dieses trotzdem nur eine Auswahl der relevanten Aspekte sein.

Das P2P-Paradigma ermöglicht durch offene Standards und Protokolle, dass sich die Nutzergemeinschaften dynamisch entwickeln können. Fast immer besteht eine Auswahl von Programmen, um am P2P-Informationsaustausch teilzunehmen. So sind Gnutella, LimeWire, BearShare u.a. Anwendungen zur Teilnahme am Gnutella-Netzwerk, d.h. all diese Programme greifen auf die gleichen Daten zu. Ähnlich wie man es z.B. bei Email beobachten konnte, werden dadurch die Inhalte der Nachrichten in den Vordergrund gerückt. Sind die Inhalte vom Nutzer nicht beeinflussbar (z.B. beim Download von MP3 Dateien), bilden sich lose, offene und zielgerichtete Ad hoc-Communities, deren gemeinsames Ziel die Optimierung des Zugangs zu den Dateien darstellt. Sind die Inhalte variabel (z.B. Chat, Foren, etc.), so bilden sich festere Strukturen. Hier sind weitere Bereiche wie Identität, Vertrauen, Bewertung der Inhalte etc. zu betrachten, die im Widerspruch zur möglichen Dynamik so gearteter Strukturen stehen. Prinzipiell wird einem unbekanntem Kommunikationspartner Misstrauen entgegengebracht. Erst nach einiger Zeit wird sich ein Vertrauensverhältnis entwickeln (Chen und Yeager, 2001), über das dann Hinweise und Informationen mit verbindlichem Charakter ausgetauscht werden (z.B. Werbung mittels des gesprochenen Wortes, Upshaw, 2001). Die P2P-basierte Unterstützung von Kommunikation und Kollaboration wird daher von bestehenden, etablierten Communities verwendet. Hier ermöglicht sie eine natürliche, spontane Kommunikation zwischen den Peers (siehe auch Kap. 3).

P2P-Netze sind von ihrem Charakter her demokratisch: dadurch, dass es keine speziell ausgezeichneten Komponenten gibt, haben alle Peers gleiche Rechte. Dieses wird sich in zukünftigen P2P-Applikationen ändern. Ansätze, z.B. eines Member-Managements, sind verfügbar (z.B. Rechtevergabe und Zugangsregelung zu Peer-Gruppen, JXTAJoinPeerGroup, 2001) und ermöglichen somit die Einrichtung von nicht-öffentlichen Gemeinschaften. In Verbindung mit verschlüsselten Nachrichten-

Übertragungen (Wearden, 2001) finden Communities mit festen Strukturen eine passende Plattform.

Die Möglichkeit, anonym und geschützt zu kommunizieren und kollaborieren, führt zur Betrachtung ökonomischer Aspekte von P2P-Applikationen. Nachdem das Ende des wirtschaftlichen Höhenfluges in den Technologie-Startups die finanziellen Möglichkeiten der i.d.R. kleinen Firmen der P2P-Entwickler einschränkt, werden auch immer mehr rechtliche Fragen gegen den P2P-Sektor entschieden. Der Gerichtsfall Napster ist der bekannteste Fall, bei dem letztendlich eine aktive, virtuelle Gemeinschaft (ca. 40 Mio Mitglieder) durch Vernichtung der IT-Basis zerstört wurde. Napster repräsentiert den Kampf der neuartigen Technologie gegen die etablierten Industrien. Hinter vielen Projekten und Applikationen stehen schon heute etablierte Firmen, welche die bestehenden Märkte gegenüber den neuen Möglichkeiten schützen wollen oder aber dabei sein möchten, wenn sich ein erfolgreiches Geschäftsmodell ergibt (Cohen, 2001). Heute erfolgreiche (d.h. stark gewinnbringende) Geschäftsmodelle widersprechen in der Regel jedoch den Ideen von P2P (Mayne, 2000). Die Kommerzialisierung kann die „Leichtigkeit“ der P2P-Nutzung stark beeinträchtigen. Dieses würden dynamische virtuelle Gemeinschaften am stärksten merken. Etablierte Gemeinschaften können in den „Untergrund“ flüchten, so wie es geschieht, wenn Strukturen bedroht sind (z.B. Übergang von Napster nach Gnutella, Idee von Freenet bzw. Publius). Dynamische ad hoc Communities benötigen jedoch dauerhaft öffentliche Plätze, damit neuen Mitgliedern zumindest die Anbindung ermöglicht wird. Und die große Anzahl von aktiven Mitgliedern ist der wichtigste Erfolgsgarant für eine P2P-basierte Community.

7. Zusammenfassung

Häufig wird in Diskussionen über P2P nur die technische Seite betrachtet. Es gibt verschiedene Definitionen, die i.d.R. nicht genau sind. Das Servent-Konzept beschreibt, dass die Agenten, auch Peers genannt, sowohl Anbieter von Daten (Server) als auch Nutzer (Client) sind. In reinen P2P-Netzen gibt es nur gleichartige Servents, die die Arbeit verrichten. In Hybridsystemen existieren ausgezeichnete Rechner, die dann eine Serverfunktion verrichten. Die Bezeichnung P2P geht heute weit über den PC-Sektor hinaus, es können sich auch PDAs und Handys bzw. andere Geräte mit oder ohne TCP/IP-Verbindung am Netz anmelden.

Eher noch als technische Eigenschaften zeichnen sich P2P-Netze durch hohe Dynamik und durch das Verrichten eines substantiellen Beitrags der Arbeit in den autonomen

Aussenknoten des Netzes aus. P2P-Netze unterstützen die Kommunikation bzw. Kollaboration in Gemeinschaften u.a. mittels Instant-Messaging-Applikationen und bieten Standards bzw. Frameworks für Resource Sharing (Rechenleistung, Speicherplatz) an. P2P-Programme sind i.d.R. auf eine Funktion fokussiert.

Die Arbeit mit P2P-Software wird durch zwei Phasen bestimmt:

- a) der Suche nach dem passenden Transaktionspartner und
- b) der eigentlichen Kommunikation, sei es die Übertragung einer Datei oder die Übertragung von Nachrichten.

Die unterschiedlichen Anwendungen unterscheiden sich häufig nur in diesen Bereichen. Anhand des Medien-Referenz-Modells wurden verschiedene Sichtweisen auf die Such- bzw. die Arbeitsphase erläutert. Auf den Ebenen Infrastruktur, Dienste, Prozess und Gemeinschaft zeigt sich, dass viele Aspekte des P2P-Paradigmas mit Entsprechungen in virtuellen Gemeinschaften übereinstimmen. Gemeinsame Themen wie FreeRiding oder Vertrauen belegen dieses.

Das P2P-Paradigma wurde stark durch technologische Entwicklungen geprägt. Immer mehr erkennt man jedoch, dass die Strukturen der P2P-Welt gut mit soziologischen Strukturen in Virtuellen Gemeinschaften übereinstimmen und diese dadurch optimal unterstützen. Für eine ökonomische Nutzung von P2P-Architekturen müssen neuartige Geschäftsmodelle geschaffen werden, da die zugrunde liegenden Ideen sowohl bei virtuellen Gemeinschaften als auch bei P2P-Software zu den heute vorhandenen Geschäftsideen nicht mehr passen.

8. Literatur

- [1] Adar, E. und Huberman, B.A. (2000), Free Riding on Gnutella, FirstMonday.dk, online bei URL: citeseer.nj.nec.com/article/adar00free.html (2002-08-10)
- [2] Anderson, D. (1998) SETI@home (Kapitel 5), in Oram, A. (Ed.): Peer-to-Peer – Harnessing the Power of Disruptive Technologies, O'Reilly, Beijing/Cambridge, 1998.
- [3] Chen, R. und Yaeger, W. (2001), Poblano – A distributed trust model for Peer-to-Peer Networks, online bei URL: <http://www.jxta.org/docs/trust.pdf> (2002-06-18).
- [4] Cohen, G. (2001) Building a successful P2P Business Model, Vortrag bei The O'Reilly Peer-to-Peer and Web Services Conference, Washington, D.C., 2001, online bei URL: http://conferences.oreillynet.com/cs/p2pweb2001/view/e_sess/1601 (2002-08-10).

-
- [5] Fattah, H.M. (2002): P2P - How Peer-to-Peer Technology Is Revolutionizing the Way We Do Business, Dearborn Trade Publishing, Chicago, 2002 (Zitat: S. 18).
 - [6] Fischbach, K. (2002) PEER-TO-PEER (P2P): Technologies, Architectures and Applications, Call-for-Papers zum Sonderheft Wirtschaftsinformatik, Ankündigung in Mailingliste ISWorld
 - [7] Groove Networks Inc., Homepage, online bei URL: <http://www.groove.net> (2002-08-01).
 - [8] Hummel, J. (2002). Online Gemeinschaften als sozio-ökonomisches Geschäftsmodell. Habilitationsschrift. Universität St.Gallen, 2002. To appear.
 - [9] Hummel, J. und Lechner U. (2001) The Community Model for Content Management, Int. Journal of Media Management (JMM), Vol 3. Issue 1, 2001. Auch: www.mediajournal.org/netacademy/publications.nsf/all_pk/1875.
 - [10] Hummel, J und Lechner, U. (2002) Profiling Virtual Communities. In: R. Sprague (ed). Proc. Of the Hawaiian Int. Conf. on System Sciences (HICSS 2002). 2002.
 - [11] Intel Corp. (2001), Intel Philanthropic Peer-to-Peer Programm, online bei URL: <http://www.intel.com/cure/> (2002-08-02).
 - [12] JXTAJoinPeerGroup (2001), Small Tutorial for Creating and Joining Peer Groups in JXTA's JAVA implementation, online bei URL: <http://platform.jxta.org/java/tutorial/PeerGroupSmallTutorial.htm> (2002-08-04).
 - [13] Lechner, U. und Schmid, B. F. (2001): Communities - Business Models and System Architectures: The Blueprint of MP3.com, Napster and Gnutella Revisited, in: Sprague, E., (Ed.) Hawaiian Int. Conf. on System Sciences (HICSS 2001), IEEE Press, <http://www.mediamanagement.org/modules/pub/view.php/mediamanagement-7>, (2002-08-10)
 - [14] Mayne, W (2000), Entrepreneurs dream of making p2p pay, Technologie-Artikel bei USA Today, online bei <http://www.usatoday.com/life/cyber/ccarch/cckev040.htm> (2002-08-10).
 - [15] NapsterProtokoll, david.weekly.org: Protokoll des Napster-Systems, online bei <http://david.weekly.org/code/napster.php3> (2002-08-02).
 - [16] OpenSkies, Massive Multiplayer Online Gaming SDK (MMPOG), (z.B. Folie 2), online bei URL: www.openskies.net/files/Openskies_MMPOG.pdf (2002-08-06).

-
- [17] O'ReillyNetworkDef, OpenP2P.com Web Site Aims to help Developers Create the Internet's Next Generation, online bei URL:
<http://www.oreillynet.com/pub/a/mediakit/pressrelease/20010123.html>.
- [18] Pennock et al: David Pennock, Gary Flake, Steve Lawrence, Eric Glover, C. Lee Giles Winners don't take all: Characterizing the competition for links on the web. In *Proceedings of the National Academy of Sciences*, 99(8), 5207-5211, April 2002. Auch: modelingtheweb.com/.
- [19] Schmid, B.F. (1999) Elektronische Märkte - Merkmale, Organisation und Potentiale. In: Hermanns, A. and Sauter, M., (Eds.) *Handbuch Electronic Commerce*, Vahlen Verlag.
- [20] Schoder, D. und Fischbach, K. (2002) Peer-to-Peer – Ökonomische, technische und juristische Perspektiven, Springer (Vorabdruck, Probekapitel), online bei <http://www.whu-koblenz.de/ebusiness/p2p-buch/buch/>.
- [21] Schollmeier, R. (2001): A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications, Extended Abstract In Proceedings of the IEEE 2001 International Conference on Peer-to-Peer Computing (P2P2001), Linköping, Sweden, August 27-29, 2001 (Übersetzung: Autor).
- [22] SETI@Home, Homepage, online at <http://setiathome.ssl.berkeley.edu/> (2002-08-04).
- [23] Shirky, C. (1998) Listening to Napster (Kapitel 2) in: Oram, A. (Ed.): *Peer-to-Peer – Harnessing the Power of Disruptive Technologies*, O'Reilly, Beijing/Cambridge, 1998.
- [24] Shirky, C. (2000) Clay Shirky on P2P, Befragung im Rahmen von davenet.com, online bei URL: <http://davenet.userland.com/2000/11/15/clayShirkyOnP2p> (2002-08-03), (Übersetzung: Autor).
- [25] Sun Microsystems, Projekt JXTA Homepage, online bei URL: <http://www.jxta.org> (2002-08-01).
- [26] Upshaw, L.B. (2001), Building a Brand.com, Ausschnitte aus The Design Management Journal, online bei URL:
<http://www.brandbuilding.com/books/building.html> (2002-07-28).
- [27] Wearden, G. (2001) P2P: From Piracy to Productivity, ZDNET (UK), online bei URL: <http://zdnet.com.com/2100-1105-531304.html> (2002-08-02).
- [28] WhatIs / Techtargent Enterprises, WhatIs.com – IT-Specific Encyclopedia; Suchwort: Peer-to-Peer, online bei URL:
http://whatis.techtargent.com/definition/0,,sid9_gci212769,00.html (2002-08-04).

