

In diesem Kapitel:

- *Was kann ein Netzwerk Programm für uns erledigen?*
- *Was gibt's denn sonst noch?*

1

Warum Netzwerk Programmierung mit Java?

1.1. Einführung in die Problematik

Java ist die erste Programmiersprache, die von Grund auf netzwerkfähig definiert wurde. Da das Internet zunehmend wichtiger und die Internet Technologien ebenso, steigt die Bedeutung von Java täglich.

Netzwerk-Programme lassen sich in jeder Programmiersprache schreiben; in Java ist es aber einfacher als in anderen Programmiersprachen. Java bietet mit dem Package `java.net` eine Basis, die ein Erstellen von Netzwerk Programmen wesentlich erleichtert.

In Java wurden auch spezielle Konstrukte zum ersten Mal definiert und eingeführt:

- Applets
- Servlets
- mobile Objekte und Remote Methode Invocation
- Java Spaces / Jini

1.2. Was kann ein Netzwerk Programm

Netzwerke ermöglichen viele Anwendungen, die lokal nicht möglich sind. Ein Programm kann Daten von einem der vielen Rechner, die zum Internet gehören, herunter laden oder Daten aus Programmen einer Niederlassung in Brasilien in den Rechner der Verwaltungsgesellschaft in einem steuergünstigen Land transferieren.

Es gibt verschiedene Kategorien von Netzwerk-Programmen:

- solche, die einfach Daten lesen und diese dann lokal weiter verarbeiten, zum Beispiel anzeigen
- solche, die Daten senden und empfangen
 - der Empfänger gehört dabei in der Regel zu einer der folgenden Kategorien
 - es handelt sich um einen Server, der die "Anfrage" verarbeitet und abhängig vom Ergebnis den Dialog, die Kommunikation mit dem Client weiterführt. Der Server hat je nach Ausprägung sehr unterschiedliche Funktionen, passive (Dateiserver) oder aktive (Applikationsserver)
 - es handelt sich um einen Client, der gleichberechtigt ist, und bestimmte Daten vom anderen Rechner ("Client") benötigt ("Peer-to-Peer" Kommunikation)
- moderne Anwendungen:
 - Agenten
 - diese nehmen Anfragen des "Meisters" entgegen und führen diese selbständig aus, reagieren proaktiv auf sich ändernde Situationen und zeigen "intelligentes" Verhalten. Die Agententheorie gehört zur "Verteilten Künstlichen Intelligenz" (Distributed Artificial Intelligence) und geht auf die späten 70er Jahre zurück.

NETZWERKPROGRAMMIERUNG MIT JAVA

Java, speziell die nun fast überall vorhandenen Java Virtual Machines erlauben es zum ersten Mal, Agenten im grossen Rahmen zu studieren und einzusetzen. Stellen Sie sich vor, wie ein Programm von einer Maschine zur andern wandern soll und ausführbar bleiben soll OHNE virtuelle Maschine!

Betrachten wir die obigen Kategorien von Netzwerk Programmen etwas genauer!

1.2.1. Lesen und Anzeigen von Daten

Die Grundstruktur eines Kommunikations-Programmes besteht aus einem Client und einem Server. Der Client liest Daten vom Server und zeigt diese in irgend einer Form an oder verarbeitet die Daten lokal.

Das konnten Programme schon lange vor Java, das ist alter Kaffee! Jeder Browser kann das. Browsers sind allerdings etwas beschränkt, oder eingeschränkt, wie auch immer. Web Browser können nur mit bestimmten Servern mit einem ganz bestimmten Protokoll (HTTP, eventuell auch noch FTP und SMTP, POP3) kommunizieren. Aber Sie haben keine Möglichkeit ein eigenes Protokoll zu definieren und damit Ihre Probleme zu lösen. Sie hätten eventuell gerne SQL oder XML als Sprachen eingesetzt. Das kann der Browser vermutlich aber nur ungenügend. Jedes Java Programm, welches Sie in eine HTML Seite einbetten oder als Standalone Applikation laufen lassen, kann all dies und noch viel mehr, sofern Sie Java beherrschen.

Java Zusätze, Applets im obigen Fall, sind flexibel, weil Java eine universell einsetzbare Sprache ist. Java Programme sehen eine Netzwerkverbindung als einen Strom von Daten, die interpretiert werden können und auf die geantwortet werden kann. Web Browser sehen nur einzelne Datentypen und können diese auf eine bestimmte Art und Weise interpretieren. Falls ein Browser Daten empfängt, die er nicht interpretieren kann, geschieht etwas unvorhersehbares.

Ein Java Programm kann beispielsweise sehr einfach beim Anbinden des Systems an das Netzwerk rasch die Systemzeit synchronisieren, oder eine Textseite herunterladen, oder Bilder, oderJava Programme können auch mit grafischen Oberflächen versehen werden, wenn auch noch mit einigem Aufwand. Aber auch hier gibt es vorgefertigte Packages, mit deren Hilfe das Look & Feel wesentlich verbessert werden kann.

Der Verbindungsaufbau zu einem Server ist denkbar einfach, da Java standardmässig mit Netzwerkkomponenten ausgeliefert wird (java.net.*). Java unterstützt vorallem TCP und UDP, also IP basierte Kommunikationsprotokolle. Aber mit zusätzlich erhältlichen Packages wie COMM kann man auch auf die COM und LPT Ports zugreifen. Wir werden später sehen, wie man auf einer höheren Ebene Protokolle (Protokoll Handler und Content Handler) implementieren kann. AppletTalk, Netware's SPX/IPX und viele weitere Protokolle werden heute (noch) nicht unterstützt. Aber das kann sich täglich ändern.

Nachdem mit Java die Verbindung zu einem entfernten Host hergestellt wurde, liegt es an Ihnen korrekt zu kommunizieren und die gesendeten / empfangenen Daten zu interpretieren. Das Packen und Entpacken der Daten ist in der Regel wesentlich komplexer als die Verbindung herzustellen. Java umfasst auch Klassen, die Ihnen dabei helfen. Beispielsweise stellt Java Klassen und Methoden zur Verfügung, um GIF oder JPEG Daten zu interpretieren. Wie bereits erwähnt, können Sie mittels eigener oder sonst irgendwo erhältlicher Handler sowohl die Protokolle (Protocol Handler) als auch die Interpretation der Inhalt der Pakete (Data Handler) beliebig ausweiten.

NETZWERKPROGRAMMIERUNG MIT JAVA

Java schmeisst auch automatisch alle nicht mehr benötigten Protokoll Handler aus dem Speicher und optimiert das System, mittels Garbage Collector.

1.1.1. Wiederholtes Lesen von Daten

Web Browser lesen die Daten je nach Bedarf. Der Benutzer verlangt eine Seite oder genauer eine URL und der Browser liest die Daten aus dieser Adresse. Das Modell ist nett solange Sie die Daten genau einmal benötigen. Es wird komplexer, sobald Sie Daten wiederholt und in gleicher Form oder zeitlich dem Markt angepasst, oder lesen möchten. Mit HTTP kann's bereits komplexer werden: Server Push ist zwar eine Idee, die einmal herum geisterte, sich aber nicht durchsetzen konnte. Sie kennen sicher auch Web Dienste, die Sie periodisch informieren, falls eine bestimmte Seite geändert wurde (beispielsweise SelfHTML).

Der Java Client hat's da etwas einfacher. Er kann beliebig oft Daten vom Server lesen, überprüfen, ob sich Daten verändert haben, eventuell mehrere Versionen zwischenspeichern und vieles mehr. Beispielsweise Börsenticker: diese lassen sich sehr einfach in Java implementieren, es handelt sich sozusagen um Netzwerk "Hello World" Programme.

Solange Daten auf dem Web erhältlich sind, kann ein Java Anwendungsprogramm sie auch lesen und interpretieren, sofern es darauf Zugriff hat.

1.1.2. Senden von Daten

Web Browser sind auf das Lesen von Daten optimiert worden. Sie senden nur limitiert Daten an einen Server, beispielsweise mittels POST in CGI Skripts oder Java Server Pages oder Applets oder ...Ein Java Programm ist in dieser Beziehung viel flexibler. Ein Vorteil ist auch, dass das selbe Programm senden und empfangen (und interpretieren) kann. Damit haben Sie also viel mehr Möglichkeiten.

1.1.2.1. Dateispeicher

Applets Anwender möchten oft Daten zwischen den einzelnen Aufrufen abspeichern, zum Beispiel den erreichten Level eines Spieles. Aber Applets können ohne spezielle Bewilligung oder Security Policy, nicht auf lokale Datenträger schreiben. Der Ausweg besteht darin, dass entweder der Benutzer um Erlaubnis gebeten wird, oder indem die Daten auf dem Server zwischengespeichert werden. Der Host / Server kann die Daten mittels eines CGI-Skripts auf seinen Datenspeicher schreiben oder mittels Servlets, Java Server Pages, oder Falls der Server FTP implementiert und akzeptiert können die Daten auch per FTP an den Server übermittelt werden.

Hier einige der vielen Möglichkeiten, die sich mit Java bieten:

1.1.2.2. Massiv paralleles Rechnen

Da Java Applets entweder mittels Sandbox Modell oder ab Version 1.2 mittels Security Policy abgesichert werden können, ist es problemlos möglich, dem Applet auf unterschiedlichen Rechnern CPU Zeit zur Verfügung zu stellen. Viele wissenschaftliche Berechnungen benötigen massive Rechnerressourcen, die am besten durch viele parallel arbeitende CPUs zur Verfügung gestellt werden können. Eine mögliche Anwendung wären Wetterberechnungen. Diese müssen viele Messpunkte berücksichtigen und berechnen mittels nichtlinearer Gleichungen das Wettergeschehen für die nächsten Stunden oder Tage, je nach Rechnerkapazität. In einer solchen Anwendung könnte ein Steuerprozess die Daten an dedizierte Rechner übermitteln. Diese Berechnen einen Teil der Lösung und informieren den Steuerprozess sobald die Berechnung abgeschlossen ist. Der Steuerprozess kann dann

NETZWERKPROGRAMMIERUNG MIT JAVA

eventuell weitere Prozessoren beauftragen, die Daten und Berechnungen zusammenzufassen und daraus ein Gesamtergebnis herzuleiten. Java bietet für solche Berechnungen eine ideale Plattform, weil die Programme plattformunabhängig entwickelt werden können, die Kommunikation relativ leicht möglich ist und das Sicherheitsmodell von Java dafür sorgt, dass nur berechnete Operationen ausgeführt werden.

1.1.2.3. Smarte Eingabeformulare

Mit Swing oder früher mit AWT (Abstract Window Toolkit) stehen Benutzerinterface-Komponenten zur Verfügung, mit deren Hilfe komplexe Anwendungen und GUIs entwickelt werden können. Damit lassen sich Eingabeformulare entwickeln, die weit über eine reine HTML oder DHTML Lösung hinausgehen. Datenfelder können auf Grund der Eingabe eingeblendet werden (das geht auch noch in DHTML), Wertetabellen können dynamisch aus Serverdatenbanken gelesen werden (das geht auch in JSP: auch hier mittels Java) und vieles mehr.

Da Applets echte Programme sind, kann ein Dialog voll ausprogrammiert werden, im Gegensatz zu einer reinen HTML Lösung, bei der dauernd die Eingabemaske an den Server gesendet werden muss.

Applets können auch Dateneingaben validieren. Dies ist innerhalb DHTML nur bedingt möglich. Java bietet wesentlich mehr Flexibilität, da Standarddatentypen und eigene Datentypen zur Verfügung stehen.

1.1.3. Peer-to-Peer Interaktion

Die obigen Modelle folgen alle dem Client-Server Muster. In Java sind jedoch auch andere Modelle denkbar und praktisch. Beispielsweise könnten Applets mit Applets kommunizieren, sofern dies der Applet Security Manager erlaubt und der Browser die Kommunikation unterstützt.

1.1.3.1. Spiele

Spiele können entweder einen zentralen Server oder ein Peer-to-Peer Kommunikationsmodell zugrunde legen. Beide Modelle haben Vorteile und Nachteile. Der Vorteil einer Peer-to-Peer Lösung ist der, dass auch ohne Server gespielt werden kann. Sie finden viele unterschiedliche Java Spiele auf dem Web, die meisten sind noch eher primitiv. Aber das kann sich ja noch ändern.

1.1.3.2. Chat

Im Kommunikationsbereich und der Netzwerkprogrammierung sind Chat Programme fast so etwas wie das "Hello World". Wir werden viele unterschiedliche Varianten von Chat Programmen entwickeln. Auch auf dem Web finden Sie fast jede denkbare Form, von 2D bis Avatar Chats mit 3D virtuellen Chat Räumen. Es liegt nur an Ihnen diese zu finden.

1.1.3.3. Kollaboration

Groupware Applikationen zu implementieren.

Programme wie Napster haben den Nachteil, dass sie einen dedizierten Server benötigen. Dies ist bei einer Peer-to-Peer Lösung nicht mehr nötig. Natürlich gibt es solche Alternativen seit langem, auch für Napster.

NETZWERKPROGRAMMIERUNG MIT JAVA

Hier sind wir erst am Anfang der Entwicklung. E-Commerce Applikationen zeigen jedoch, speziell mit ihren 1:1 Marketing Fähigkeiten, in welche Richtung die Entwicklung geht: Sie erhalten beim Zugriff auf das WWW ein Ihren Fähigkeiten und Neigungen angepasstes Menü zur Verfügung gestellt. Dieses passt sich dynamisch Ihrem Verhalten an. Sie werden nicht durch unnötigen Ballast belästigt und gebremst.

1.1.4. Server

Neben den Clients steht Java auch die Welt der Server offen. Sun hat neben dem HotJava Browser, der sich dynamisch konfigurieren lässt, und jeweils benötigte Module direkt aus dem Web herunterlädt, auch eine Referenzimplementation eines Java Webservers publiziert. Diese steht Ihnen als Trial oder unlimitierte Version auf der Java CD zur Verfügung.

Es hält Sie aber niemand davon ab, selber weitere Server für unterschiedliche Protokolle zu implementieren, zum Beispiel FTP, Telnet, POP3, SMTP und Sie werden Teile solcher Server im Laufe der nächsten Wochen implementieren müssen.

Natürlich finden Sie auch dazu auf dem Web schon beliebig viel. Aber die eigene Implementation ist nicht besonders schwierig, ausgehend vom RFC, der Beschreibung des Protokolls.

Sie können aber auch Server für eigene spezielle Anwendungen schreiben. Beispiele wären ein Spielserver, der den Spielstand abspeichert, so dass Sie jederzeit mit dem Spiel weiterfahren können.

Mit Hilfe komplexerer Mechanismen lassen sich diese Beispiele beliebig erweitern. Wir werden mögliche bestehende und neu publizierte, jetzt noch nicht bekannte Ansätze kennenlernen, wann immer etwas neues erscheint, werden Sie Gelegenheit haben sich direkt damit zu beschäftigen.

1.1.5. Zukünftige Applikationen

Mit Java wird dem Entwickler endlich die Möglichkeit geboten, Anwendungen schnell und portabel zu entwickeln. Studien zeigen, dass die Entwicklungszeiten in Java mindestens 20% unter jenen von C++ liegen. Dazu kommen viele Packages, die Sie direkt ab WWW beziehen können, und überwiegend erst noch gratis.

Clients können intelligenter werden, die Applikationen können aufgeteilt werden, in mehr als zwei Tiers (Client / Server), beispielsweise mittels des MVC (Model-View-Controller) Design Patterns (Entwurfsmuster) oder als Multitier Applikation, mit einer Präsentationsschicht, einer Businesslogiksicht und einer Persistenz-/Datenspeicherungssicht. Die Unterscheidung "Client" oder "Server" wird oft nebensächlich. Das führt zu neuartigen Applikationen, an die man noch vor wenigen Jahren nicht denken konnte.

Java's Erfolg basiert sicher auf der starken Netzwerkausrichtung, von Grund auf! Zusätzlich spielt das Sicherheitsmodell, etwas Einzigartiges, in keiner andern Programmiersprache vorhandenes. Dieses wurde in der Version Java 2 wesentlich verfeinert.

Schauen wir uns einige bereits bestehende Applikation an, die wesentlich auf Java aufbauen.

NETZWERKPROGRAMMIERUNG MIT JAVA

1.1.5.1. Einkaufswagen

Ein Einkaufswagen sammelt Informationen über Artikel, die ein Kunde eines Online-Shops bestellen möchte. Im Rahmen von HTML wäre die Realisierung eines Online-Shops ausschliesslich mit DHTML und HTTP äusserst aufwendig, wenn überhaupt machbar. Auch die Belastung der Server CPU ist gewaltig, speziell im Vergleich zu einer Java Lösung.

Mit Java können Sie viele Arbeiten direkt zum Client verschieben. Informationen werden zum Server übermittelt sofern und sobald der Kunde dazu bereit ist. Sogar Kreditkarten Plausibilitätsprüfungen lassen sich lokal erledigen.

1.1.5.2. 3D Welten

Virtual Reality Modelling Language (VRML) gestattet es Ihnen 3D Welten aufzubauen, zu entwerfen und anzuzeigen. Dabei muss man sich im Klaren sein, dass VRML eigentlich ein Dateiformat ist. Das heisst, dass Sie Ihre Welten mit irgend einem CAD System entwerfen und anschliessend als VRML Modell abspeichern. Java bietet auf unterschiedlichen Ebenen 3D Unterstützung, insbesondere im 3D API. Dieses wurde zur Modellierung komplexer Systeme im Bereich der Hochenergieforschung und in der Architektur, dem Turbinenbau, ... eingesetzt. Sie finden auf dem Java Website von Sun einige Links zu diesen Arbeiten. Viele Teile der Spezifikation wurden von Silicon Graphics entwickelt und stehen nun allen Java Programmieren zur Verfügung.

1.1.5.3. Spiders

Java Programme können durch das Web wandern und Informationen sammeln. Solche Programme bezeichnet man als *Spider*. Spiders durchwandern das WWW indem sie Links folgen. Sie geben eine Startseite an, der Rest wird durch den Spider erledigt. Es ist relativ einfach Spiders in Java zu schreiben. Ob es sinnvoll ist, selber Spiders zu entwickeln sei dahin gestellt. Sie können dafür ja einfach eine Suchmaschine wie etwa <http://www.google.com> einsetzen.

Dass das Spidergeschäft durchaus real ist, zeigt der kommerzielle Erfolg von Eurospider von Peter Stäubli: <http://www.eurospider.com>. Seine Spiders werden zum grossen Teil von Firmen eingesetzt. Die Herausforderung liegt vorallem in der Definition und Implementation cleverer Algorithmen.

Sie werden in absehbarer Zeit Gelegenheit haben einen Spider selbst zu entwickeln. Spiders benötigen einige Intelligenz, um die Anzahl Besuche einer Seite auf ein Minimum zu reduzieren und die vorhandenen Informationen möglichst optimal, was das auch immer heisse, zu repräsentieren.

1.1.6. Intelligente Agenten

Intelligente Agenten sind Programme, die sich von einem Host zum andern bewegen können. Es gibt verschiedene kommerzielle Agentensysteme zum Beispiel Aglets von IBM. Aber es ist relativ leicht in Java Agenten zu programmieren, auch lernfähige. Intelligente Agenten können zum Beispiel eingesetzt werden, um das günstigste Angebot einer Produkts im WWW zu finden. Die Technologie ist bereits seit mehreren Jahrzehnten bekannt, speziell aus dem Forschungsgebiet "Distributed Artificial Intelligence". Erste Systeme wurden bereits in den 70er Jahren vorgestellt. Sie finden auf dem Web beliebig viele Beispiele von Forschungsarbeiten zu diesem Thema. Allerdings zeigt es sich zunehmend, dass heutige Agentenanwendungen eher dumm sind und auch dumm bleiben werden. Eine echte

NETZWERKPROGRAMMIERUNG MIT JAVA

Killerapplikation fehlt offensichtlich bis heute. Trotzdem werden Sie Gelegenheit haben, die Technologie zu testen und selber Agenten zu programmieren.

Einer der Hauptgründe für die sehr langsame Verbreitung der Agententechnologie war eine fehlende Portabilität. Diese ist jetzt mit Java gegeben. Allerdings muss bei der Entwicklung von Agenten darauf geachtet werden, dass nicht Viren gebaut werden.

Ein Internet Worm ist in gewissem Sinne ein pervertierter Agent. Java bietet zum Glück viele Sicherheitsmechanismen, mit deren Hilfe der Schaden durch solche Gebilde eingeschränkt werden können.

Auch Spider Anwendungen könnten mit Hilfe von Agenten realisiert werden. Sie könnten einem Agenten den Auftrag geben, relevante Informationen für Ihre Projekte im WWW zu sammeln. Damit Sie nicht alle Links testen müssen, könnten Sie weitere Kriterien angeben und beispielsweise ein semantisches Netz anlegen lassen, in dem alle wesentlichen Begriffe eingeordnet werden. Sie könnten periodisch über den aktuellen Status des semantischen Netzes informiert werden. Der Spider würde laufend oder periodisch suchen und Sie über die neusten Links informieren.

Für den Einkauf stehen viele Agenten im WWW zur Verfügung. Suchen Sie mal kurz im WWW und Sie werden sicher einige davon finden. Am MIT wurde auch ein Agentensystem entwickelt, welches "Interessensgemeinschaften" aufbauen kann: sie geben an, welche Präferenzen Sie haben (Musik, ...) und der Agent, Ihr Agent, sucht andere Agenten im Web, welche ein ähnliches Profil haben. Das Ganze kann auch erweiterbar sein: Ihr Agent lernt von andern Agenten, was deren "Meister" sonst noch so interessiert. Ihr Agent lernt dies und teilt es Ihnen mit. Sie entscheiden, was mit dieser Zusatzinformation geschieht: soll in Zukunft diese Information als Erweiterung Ihres Profils verwendet werden, oder soll die Information ignoriert werden.

1.1.7. Electronic Commerce

Alle gängigen e-Commerce Lösungen, die heute signifikanten Marktanteil besitzen, verwenden Java in hohem Masse. Eine Realisierung eines e-Shops oder eines Katalogsystems mit DHTML und CGI Skripts ist so gut wie unmöglich, speziell falls komplexere Funktionen, wie Gruppenbildung (Communities), 1:1 Marketing und vieles mehr berücksichtigt werden sollen.

Sie haben die Möglichkeit beliebig viele Shopsysteme zu testen und selber Shops aufzubauen. Ihnen stehen dazu die gängigen Shop und Katalogsysteme zur Verfügung.

1.3. Aber das ist nur der Anfang

In den folgenden Kapiteln befassen wir uns mit den Grundlagen moderner Netzwerkanwendungen. Sie werden aber, falls neue Techniken oder Konzepte vorgestellt werden, sofort Gelegenheit haben, die Konzepte kennen zu lernen und auszutesten.

Es ist also ein Ziel, Sie ganz nach vorne zu bringen und Ihnen die aktuellsten Entwicklungen aufzuzeigen. Dies geschieht in der Regel schon bevor es kommerzielle Implementationen gibt. Denn in der Regel dauert es einige Jahre (1-3) bis eine neue Idee auf der konzeptionellen Ebene sich in kommerziellen Produkten zeigen. Aber die Entwicklung beschleunigt sich dauernd. Heute geht es darum, neue Ideen möglichst innerhalb von Monaten vermarkten zu können.

NETZWERKPROGRAMMIERUNG MIT JAVA

Einige der Konzepte, die Sie kennen lernen, sind heute vermutlich noch nicht bekannt, werden aber im Laufe des Jahres bekannt werden.

Macht es Sinn so aktuelle Fragen und Konzepte anzugehen?

Ich glaube schon, selbst wenn sich ein Konzept nicht durchsetzen wird, lernen Sie einiges.

Sie werden einige Programme schreiben müssen, einige stehen Ihnen einfach so zur Verfügung. Aber das zeitliche Engagement dürfte nicht so ganz ohne sein, sofern Sie die Konzepte auch verstehen wollen und anwenden möchten.

NETZWERKPROGRAMMIERUNG MIT JAVA

WARUM NETZWERK PROGRAMMIERUNG MIT JAVA?	1
1.1. EINFÜHRUNG IN DIE PROBLEMATIK.....	1
1.2. WAS KANN EIN NETZWERK PROGRAMM.....	1
1.2.1. <i>Lesen und Anzeigen von Daten</i>	2
1.1.1. <i>Wiederholtes Lesen von Daten</i>	3
1.1.2. <i>Senden von Daten</i>	3
1.1.2.1. <i>Dateispeicher</i>	3
1.1.2.2. <i>Massiv paralleles Rechnen</i>	3
1.1.2.3. <i>Smarte Eingabeformulare</i>	4
1.1.3. <i>Peer-to-Peer Interaktion</i>	4
1.1.3.1. <i>Spiele</i>	4
1.1.3.2. <i>Chat</i>	4
1.1.3.3. <i>Kollaboration</i>	4
1.1.4. <i>Server</i>	5
1.1.5. <i>Zukünftige Applikationen</i>	5
1.1.5.1. <i>Einkaufswagen</i>	6
1.1.5.2. <i>3D Welten</i>	6
1.1.5.3. <i>Spiders</i>	6
1.1.6. <i>Intelligente Agenten</i>	6
1.1.7. <i>Electronic Commerce</i>	7
1.3. ABER DAS IST NUR DER ANFANG.....	7