

Projekt JXTA

Übersicht einer Technologie

Vortrag zum Proseminar Funk und P2P-Netze
TUM SS2003

Boian Tzonev
erstellt mit L^AT_EX

Einführung JXTA ist eine Netzprogrammierungs- und Computerplattform, die mit dem Ziel entworfen ist, eine Reihe von Problemen in der gegenwärtigen Informatik und besonders im Gebiet der Peer-to-peer Netze zu lösen.

Warum Projekt JXTA Projekt JXTA hat als Ziel die Behebung einiger Mängel und Schwächen der zur Zeit existierenden oder entwickelnden Peer-to-peer-Systeme. Das sind insbesondere:

- **Kompatibilität.** Die Technologie JXTA ist konzipiert so, dass verbundene Benutzer in der Lage sind, sich gegenseitig zu finden, miteinander zu kommunizieren, in gemeinsamen Aktionen teilzunehmen und gegenseitig Dienste ohne Unterbrechung zwischen verschiedenen P2P-Systemen und verschiedenen Gemeinschaften anzubieten. Viele P2P-Systeme sind entwickelt, um eine einzige Art von Diensten zu liefern. Zum Beispiel Napster ermöglicht den Austausch von Musikdateien, Gnutella ermöglicht den allgemeinen Dateiaustausch und AOL Instant Messenger bietet instant messaging. Mit den unterschiedlichen Eigenschaften dieser Dienste und die Abwesenheit einer gemeinsamen zugrunde liegenden P2P-Infrastruktur ist jeder Softwareanbieter gezwungen, inkompatible Systeme zu entwickeln. Das bedeutet, dass jeder Softwareanbieter seine eigene P2P-Benutzergemeinschaft schafft und so Ressourcen in die Entwicklung von Software und Systemgrundbausteine verliert, die von allen P2P-Systemen verwendet werden. Außerdem muss ein Benutzer, der an verschiedenen Gemeinschaften teilnehmen will, die durch verschiedene P2P-Anwendungen organisiert sind, auch mehrere Anwendungen unterstützen.
- **Betriebssystemunabhängigkeit.** JXTA ist unabhängig von Programmiersprachen, Betriebssystemen und Netzwerkprotokollen. Viele der heutigen P2P-Systeme bieten ihre Dienste und Leistungen nur im Zusammenhang mit einem gewissen API, Betriebssystem und Netzwerkprotokoll. Demzufolge ist ein P2P-Entwickler gezwungen, eine bestimmte Plattform und so einen Kundenkreis im voraus zu wählen. Will er seinen Kundenkreis erweitern, dann muss er ein und dasselbe Dienst für mehrere P2P-Plattformen zu entwickeln. JXTA dagegen ist entwickelt so, dass sie von allen Entwicklern unabhängig von der bevorzugten Programmiersprache oder Betriebssystem angewendet werden kann.
- **Geräteunabhängigkeit.** JXTA kann auf jedem digitalen Gerät einschließlich Sensoren, elektronischen Haushaltsgeräten, PDAs, Netzwerkroutern, Personalcomputer, Datenserver u. a. angewendet werden.

Die JXTA Technologie Projekt JXTA wurde initiiert von Sun Microsystems, Inc. und ist open-source. Am Anfang haben die JXTA Entwickler verschiedene P2P-Softwarearchitekturen analysiert und eine gemeinsame Struktur gefunden.

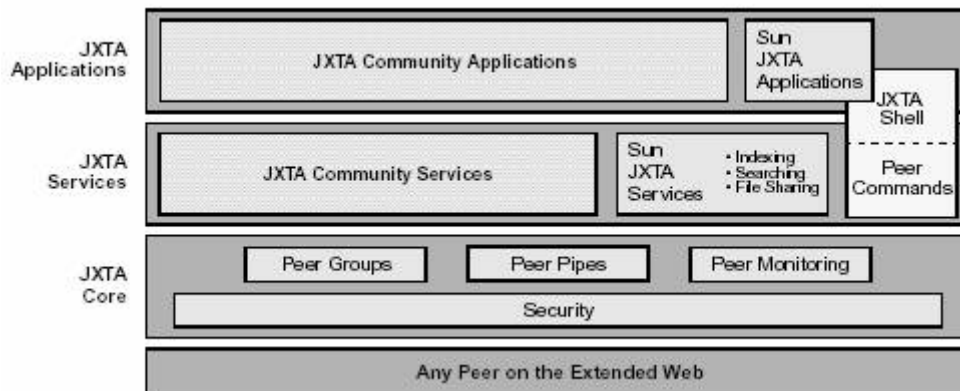


Abbildung 1:P2P Software Architecture

Demgemäss kann man einen typischen P2P-Softwarestack in drei Ebenen teilen. Ganz unten befindet sich die Kernebene (core layer), welche sich mit dem Lokalisieren von Peers und Kommunikationsmanagement wie Routing befasst. In der Mitte befindet sich eine Dienstebene(service layer), welche sich mit höheren Konzepten wie Indexing, Suchen und Dateiaustausch befasst. Die oberste Ebene ist die Ebene für Anwendungen wie E-maildienste, Speichersysteme usw. Einige Eigenschaften, wie z. B. Sicherheit, sind in allen drei Ebenen präsent, auch wenn in verschiedener Form gemäss der Position im gesamten P2P-System. Die JXTA Technologie ist gedacht als oberste Ebene, in der Dienste und Anwendungen realisiert werden.

Konzepte der JXTA-Technologie JXTA Technologie. Auf höchster Abstraktionsebene stellt JXTA eine Zusammenstellung von Protokollen dar. Jedes Protokoll wird durch eine oder mehrere Nachrichten bestimmt, die zwischen Teilnehmer am Protokoll ausgetauscht werden. Jede Nachricht hat ein vordefiniertes Format und kann verschiedene Datenfelder enthalten. JXTA ist transportabhängig und kann TCP/IP wie auch andere Transportprotokolle benutzen. Es sind folgende Protokolle definiert:

- Peer Discovery Protocol
- Peer Resolver Protocol
- Peer Information Protocol
- Rendezvous Protocol
- Pipe Binding Protocol
- Endpoint Routing Protocol

Diese Protokolle bauen auf einige Konzepte wie peer, peer group und advertisement auf, die im folgenden kurz erklärt werden.

Kennungen. JXTA verwendet UUID, eine 128-bit Datenrepräsentation, um sich auf eine Instanz (Peer, advertisement, Dienst usw.) zu beziehen. Es ist leicht sicherzustellen, dass jede Instanz eine eindeutige UUID zur Laufzeit bekommt,

aber es gibt keine absolute Möglichkeit eine solche UUID in einer globalen Umgebung sicherzustellen. Das aber ist nicht so ein großes Problem, weil eine UUID als eine interne Kennung benutzt wird.

Ankündigungen. Ein advertisement ein XML strukturiertes Dokument, das die Existenz einer Ressource wie Peer, Peer Group, Pipe oder Dienst benennt, beschreibt und veröffentlicht. Es gibt einige Grundtypen von Ankündigungen, die mittels XML-Schemata erweitert werden können.

Peers. Ein Peer ist eine Instanz, die die für einen Peer notwendigen Protokolle beherrscht. Als Peer kann ein Prozessor, Prozess, Gerät oder Benutzer handeln. Wichtig ist, dass ein Peer nicht unbedingt alle obenangegebenen Protokolle beherrschen muss.

Nachrichten. Die Nachrichten sind gedacht so, dass sie über einen asynchronen, nichtssicheren und in einer Richtung verlaufenden Transport verwendet werden. Aus diesem Grunde stellt eine Nachricht ein Datenpaket dar, das aus einem Umschlag und Protokollheaders mit Hauptteil besteht. Der Umschlag enthält einen Header, Auszug aus der Nachricht und optional Quellendpunkt und Zielendpunkt. Ein Endpunkt ist ein logisches Ziel in der Form einer URL auf jedem Netzwerktransportprotokoll, das Datenpakete übertragen kann.

Der Hauptteil jedes Protokolls enthält verschiedene Anzahl von Bytes und ein oder mehr Merkmale, die zum Identifizieren des Senders vom Empfänger benötigt werden. Diese Merkmale haben kein vorbestimmtes Format und Inhalt. So kann zum Beispiel eine Nachricht verschlüsselt sein und die Merkmale können Informationen zum Entschlüsselung beinhalten.

Peer groups. Eine peer group ist eine virtuelle Instanz, die die Peergruppenprotokolle beherrscht. Normalerweise bietet eine peer group gemeinsame Dienste für eine Gruppe von kooperierenden Benutzern. Die JXTA-Spezifikation aber besagt nicht wann, wo oder warum eine peer group hergestellt werden soll. Ausserdem gibt es keine Einschränkungen zu wie viel Gruppen ein Benutzer gehören kann. Sie bestimmt nur wie man peer group mittels dem Peer Discover Protocol entdeckt werden können.

Es gibt eine auch besondere Gruppe, die World Peer Group, die alle JXTA peers einschließt.

Kanäle(pipes). Das sind Kommunikationskanäle zum Senden und Empfangen von Nachrichten. Sie verlaufen immer nur in einer Richtung, so dass es Eingangs- und Ausgangskanäle gibt.

Normalerweise wird ein Kanal dynamisch zur Laufzeit zu einem Peer mittels dem Pipe Binding Protocol gebunden. Das heißt also, dass ein Kanal verlegt und zu verschiedenen Peers zu verschiedenen Zeiten gebunden werden kann.

Es gibt point-to-point-Kanäle, die genau zwei Peers miteinander verbinden, und multicast-Kanäle, die einen Ausgangskanal mit mehreren Eingangskanälen verbinden.

Die Protokolle der JXTA Technologie Am Anfang wurden im Projekt JXTA folgende sechs Protokolle definiert. Es können weitere Protokolle definiert werden.

Peer Discovery Protocol. Dieses Protokoll ermöglicht einen Peer Ankündigungen über andere Peers zu finden und kann benutzt werden, um Peers, peer groups, oder Ankündigungen zu finden.

Peer Resolver Protocol. Dieses Protokoll ermöglicht einen Peer Suchanfragen nach Peers, peer groups, Kanäle und andere Information zu senden und zu empfangen. Normalerweise wird dieses Protokoll nur von Peers benutzt, die Zugang zu Datenbeständen haben und erweiterte Suchmöglichkeiten anbieten.

Peer Information Protocol. Dieses Protokoll gibt einem Peer die Möglichkeit, Informationen über den Status und Möglichkeiten eines Peers zu gewinnen. So kann zum Beispiel ein Peer eine ping-Nachricht zu einem Peer senden, um zu prüfen, ob der letzte online ist.

Rendezvous Protocol. Dieses Protokoll gibt einem Peer die Möglichkeit, eine Nachricht innerhalb einer peer group zu verbreiten.

Pipe Binding Protocol. Dieses Protokoll gibt einem Peer die Möglichkeit eine Kanalankündigung zu einem Kanalendpunkt zu binden, um so zu zeigen wo eigentlich Nachrichten über den Kanal laufen. In einem gewissen Sinne kann ein Kanal als eine abstrakte, benannte Nachrichtenwarteschlange gesehen werden, die eine Reihe von abstrakten Operationen wie Herstellen, öffnen, Schließen, Löschen, Senden und Empfangen unterstützt.

Endpoint Routing Protocol. Dieses Protokoll erlaubt es einem Peer, einen Router nach verfügbaren Wegen zu einem Zielppeer zu fragen. Oft können zwei kommunizierende Peers nicht direkt miteinander verbunden sein. Beispiel dafür sind Peers, die nicht ein und dasselbe Netzwerktransportprotokoll benutzen, oder Peers, die durch eine Firewall getrennt sind. Peer Router antworten auf Anfragen mit Information über verfügbare Wege, die eine Liste von Gateaways entlang dem Weg darstellt. Jeder Peer kann ein Peer Router sein, indem er das Routing Protocol implementiert.

Sicherheitsüberlegungen Die Sicherheitsanforderungen eines P2P-Systems ähneln sehr denjenigen jedes anderen Computersystems. Die drei Hauptanforderungen sind Vertraulichkeit, Vollständigkeit und Verfügbarkeit. Die spiegeln sich in spezifische Funktionalitätsanforderungen wie Authentifizierung, Zugangskontrolle, Prüfung, Verschlüsselung, sichere Kommunikation und andere.

Solche Anforderungen werden im Allgemeinen mittels einem passenden Sicherheitsmodell oder architektur erfüllt, was meistens in der Form von Subjekten, Objekten und Aktionen implementiert wird.

Da JXTA auf die Konzepte des Peers und der peer groups aufbaut, kann man sich eine Sicherheitsarchitektur vorstellen, in der Peer- und Gruppenkennungen

als Subjekte auf niedriger Ebene (wie uid und gid in UNIX), Code und Daten als Objekte (wie Dateien) angesehen werden. Aktionen sind die Operationen auf Peers, peer groups, Code und Daten.

Beim Betrachten der Sicherheitsarchitektur ist es wichtig, dass die Sicherheitsanforderungen von JXTA durch einige einzigartigen Eigenschaften beeinflusst wird:

- JXTA ist eine Plattform, die sich auf Mechanismen und nicht auf Politiken konzentriert. Zum Beispiel werden UUIDs überall verwendet, aber sie haben keine äußere Bedeutung. Ohne zusätzliche Benennungs- und Bindungsdienste sind UUIDs nur Zahlen und können nicht auf einen Benutzer oder zurückgeführt werden. Aus diesem Grund bestimmt JXTA kein Sicherheitsmodell auf höchster Ebene wie zum Beispiel information flow, Bell-LaPadula, oder Chinese Wall.

Werden UUIDs zu externen Namen oder Instanzen gebunden, kann man die Echtheit der Bindung durch Anhängen von Sicherheitsattributen wie zum Beispiel digitale Signaturen sicherstellen.

- JXTA ist neutral gegenüber kryptographische Protokolle oder Sicherheitsalgorithmen. In dieser Weise wird keine spezifische Sicherheitslösung vorgeschrieben. Es werden nur genug Schnittstellen und Platzhalter zur Verfügung gestellt, so dass verschiedene Sicherheitslösungen implementiert werden können. So hat zum Beispiel jede Nachricht ein Merkmal, das zum Speichern von sicherheitsrelevanter Information verwendet werden kann.
- JXTA kann manchmal Sicherheitsanforderungen auf verschiedenen Ebenen erfüllen. Zur Illustration dessen kann man folgende Anforderung in Betracht nehmen:

Kommunikationssicherheit. Die Kommunikation zwischen Peers erfolgt mittels Kanälen. Angenommen es werden Vertraulichkeit und Integrität von einem Kanal erfordert. Eine Lösung ist VPNs zu nutzen. Eine andere Lösung ist, eine sichere Version des Kanals herzustellen, so dass jede Nachricht, die durch diesen Kanal übertragen wird, gesichert wird. Eine dritte Lösung ist, gewöhnliche Kommunikationsmechanismen zu nutzen, wobei der Entwickler spezifische Daten mit Verschlüsselungstechniken oder digitalen Signaturen sicherstellt. All das kann mit JXTA verwirklicht werden.

Typische P2P-Anwendungen

1. Content Sharing Netzwerke

Die Idee hinter den Content Sharing Netzwerken ist der gemeinsame Zugriff auf Dateien mit beliebigem Inhalt. Die bisher gängigste Vorgehensweise dabei war die zentralisierte. Bei dieser Vorgehensweise werden die Daten auf einen FTP- oder Webserver gespeichert, von dem dann Dateien von Klienten heruntergeladen werden können. Eine andere Lösung bringen P2P-Netze, bei denen jeder Peer Daten zum Download zur Verfügung stellen kann. Einige der ersten und bekanntesten solcher Anwendungen sind Napster und Gnutella. Bei Gnutella handeln die Peers nicht nur als Server, sondern sind auch für die Beantwortung von Anfragen und Routen von Nachrichten zu anderen Peers verantwortlich.

2. Instant Messaging

Instant Messaging Anwendungen sind nicht neu. Einige Instant Messaging Anwendungen existieren seit den Zeiten der frühesten Mehrbenutzer- Großrechner-Systeme. Beispiel dafür ist das Kommando `talk` auf Unix. Viele von diesen Anwendungen benutzen viele P2P-Lösungen, aber trotzdem basieren fast alle auf zentralisierten Lösungen, um:

- (a) Einen minimalen Grad an Bedienung sicherzustellen
- (b) Die Benutzung des Systems zu kontrollieren und zu überwachen.

Bei dieser zentralisierten Vorgehensweise müssen Nutzer als erstes sich bei einem Server anmelden. Sie zeigen ihre Anwesenheit dem Server und müssen sich immer wieder mit dem Server verbinden, um aktiv zu bleiben. Nutzer, die zu anderen Nutzern Verbindung herstellen möchten, müssen sich zuerst mit dem Server verbinden. Nachdem sie die zum Herstellen einer Verbindung notwendige Information erhalten haben, kann der Nutzer entweder eine direkte Verbindung herstellen oder mittels dem zentralisierten Server mit anderen Nutzern kommunizieren. JXTA dagegen hat eine reine P2P Vorgehensweise. Jeder Peer sendet eine Ankündigung über seine Anwesenheit, die solange aktiv bleibt, bis der Nutzer online ist. Ein Peer, der Verbindung zu einem Anderen sucht, macht eine Suchanfrage nach der Ankündigung verfügbar des gesuchten Peers. Der Ausgangspeer verbindet sich mit dem Eingangskanal des Zielpeers.

Praktische Anwendungen

Als nächstes werden anhand einiger Beispielanwendungen die Möglichkeiten der Technologie JXTA gezeigt. An neuen Projekten wird im Rahmen von Open-Source Projekten auf freiwilliger Basis gearbeitet. Es werden auch kommerzielle Anwendungen angeboten.

Open-Source Projekte

gnougat

Gnougat ist eine Anwendung für Dateiaustausch, die sehr ähnlich zu Gnutella ist, mit dem Unterschied, dass die Verantwortung für den Dateiaustausch unter alle Peers im Netzwerk verteilt werden kann. Außerdem erfolgt bei gnougat die Suche nach dem Inhalt einer Datei und nicht nach Metadaten-Information und eine Datei kann in mehrere Teile geteilt werden und so parallel von verschiedenen Nutzern heruntergeladen werden.

p2pconference

P2pconference ist eine Anwendung zur Durchführung von Konferenzen in einem Netzwerk. Dabei wird eine Konferenz von einem sogenannten Direktor organisiert. Der Direktor wählt das Hauptthema, die Besprechungspunkte und lädt Teilnehmer an der Konferenz ein. Der Moderator führt und kontrolliert die Diskussion und kündigt Themen zur Abstimmung an. Mit p2pconference können drei Arten von Konferenzen durchgeführt werden:

- Treffen
Bei einem Treffen hat der Moderator nur geringe Kontrollkraft, da er störende Teilnehmer nur frieren kann.
- Präsentation
Hier wird von einem Peer eine Rede gehalten, und das Publikum kann ihm (nach Anmeldung) Fragen stellen.
- Panel Das ist eine verallgemeinerte Form der Präsentation. Hier gibt es mehrere Sprecher, die sich anmelden, wenn sie sprechen wollen. Jeder Peer als dem Publikum kann wählen, welchem der Sprecher er eine Frage stellen will.

jxtaChess

JxtaChess ist ein Programm zum Schachspielen. Ein Nutzer hat die Möglichkeit ein Spiel zwischen anderen Peers zu beobachten oder selbst zu Spielen, nachdem man sich einen Gegner ausgewählt hat.

picshare

Picshare ist eine Anwendung, die Bilder in einer Peergruppe verbreitet. Dabei kann jedes Bild mit Kommentaren versehen werden.

Kommerzielle Anwendungen

VistaRepository

VistaRepository stellt eine standardisierte und sichere Umgebung, in welcher verschiedene Dienste angeboten werden können, um eine verteilte oder zentralisierte Speicherung und einen flexiblen Zugriff auf Objekte verschiedener Art zur Verfügung zu stellen. VistaRepository erleichtert die P2P-Kommunikation, den gemeinsamen Dateizugriff und die Netzwerkzusammenarbeit.

Topology

Topology ist ein JXTA-fähiger XML/SGML/HTML-editor, entwickelt besonders für die Lösung von Problemen von Entwicklern und Entwicklerteams bei der Arbeit mit Markup-Dokumenten. Topology fördert die Zusammenarbeit, weil Nutzer screenshots austauschen und annotieren und sich Nachrichten senden können. Außerdem können plugins über Topology aktualisiert werden, was zu einem geringeren Verwaltungsaufwand führt.

IamDeveloping

IamDeveloping ist Teil von einem Paket Anwendungen für Zusammenarbeit, das auf JXTA basiert. Es ist ein Integrated Development Environment (IDE). Mit IamDeveloping können mehrere Programmierer gleichzeitig denselben Code oder Konfigurationsdateien zu sehen und zu bearbeiten. Man kann Code gemeinsam im Netzwerk kompilieren, testen und korrigieren. Während ein Nutzer an einer Datei arbeitet, kann ihn ein Anderer dabei beobachten oder selbst an einer verschiedenen Datei arbeiten. IamDeveloper kann zusammen mit einem Telefongespräch oder einem Instant Messaging Programm benutzt werden.

Zukunft von JXTA

JXTA ist eine wichtige Evolutionstechnologie, die viel Zukunftspotenzial aufweist. Zur Zeit wird in der JXTA-Entwicklergemeinschaft an vielen Projekten gearbeitet. Einige richten sich an die Lösung von neuen Problemen im Gebiet der Peer-2-Peernetze, andere an die Erweiterung der Kernreferenzanwendung von JXTA oder des Kerns von JXTA auf andere Sprachen als Java. Der künftige Erfolg von JXTA hängt von der Fähigkeit solcher Anwendungen und Dienste die Vorteile und die Anwendbarkeit von JXTA zu demonstrieren. Im Folgenden werden einige der am meisten entwickelten Anwendungen vorgestellt.

Content Addressable Networks

Content Addressable Networks (CAN) sind ein neues Forschungsgebiet. Die Idee dabei ist, das ganze Netzwerk als eine riesige Hash-tabelle zu betrachten, die Schlüssel-wertpaare beinhaltet. Die Implementierung erfordert die Wartung eines verteilten, nichtzentralisierten Informationsindex, der die hashing unterstützt. Dieser verteilte Index wird normalerweise von den teilnehmenden Peers gespeichert. Diese nichtzentralisierte Indeximplementierung wird sicherstellen, dass die Netzwerkhashtabelle immer verfügbar ist, solange ein Knoten online oder aktiv im Netzwerk bleibt. JXTA kann solche Netzwerke durch folgende Methoden unterstützen, durch die rendezvousbasierte Entdeckung, IP Multicasting, Hashtabelle.

Remote Method Invocation

Die Remote Method Invocation API, eingeführt mit Java 2 Standard Edition erlaubt einem Programm Methoden auf Java-Objekten auszuführen und Objektinstanzen zu transportieren, die sich auf andere JVM oder anderen Rechner in einem Netzwerk befinden. Die JXTA-RMI erlaubt einem Programmierer als Transportmechanismus JXTA-Kanäle statt TCP/IP zu benutzen. Der Vorteil dieser Vorgehensweise ist, dass sie RMI-basierten Anwendungen erlaubt, auf bestehenden JXTA-Anwendungen ohne große Veränderungen aufzubauen.

Weitere Information über das Projekt JXTA kann auf www.jxta.org finden. Dort kann man sich Sourcecode downloaden, sich für Mailinglisten anzumelden und an verschiedenen Projekten teilnehmen und arbeiten.

Literatur:

Easy Adopter JXTA, Sing Li, 2001, Wrox Press Ltd.

JXTA, Brendon J. Wilson

(kostenlos unter <http://www.brendonwilson.com/projects/jxta/pdf/JXTA.pdf> zu beziehen)

www.jxta.org