

JXTA

Ein Überblick

HS-Telematik: JXTA, von Ronny Heidenreich

Motivation

- bisher:
 - eigenständige, spezialisierte und inkompatible P2P-Lösungen
 - mit erheblichem Entwicklungsaufwand verbunden
- neuer Ansatz:
 - > **JXTA Framework**
 - neue, offene Protokolle, um das dezentrale Distributed Computing zu standardisieren
 - > **offener P2P-Standard** auf dem **beliebige Dienste** aufsetzen können
 - P2P-Netze auf jeder Plattform und jedem Gerät ermöglichen

Gliederung

- Einführung in P2P
- JXTA Geschichte & Ziele
- JXTA Architektur
 - *Core Layer*
 - *Service Layer*
 - *Application Layer*
- JXTA Anwendungen
- Fazit

3

Einführung in P2P

- Peer-to-Peer Begriffsklärung
 - Peer(s)
 - Peer-to-Peer
 - Peer-to-Peer-Kommunikation
 - P2P-Netzwerk
 - **Servents** (Client & Server gleichzeitig)
 - Peers sind dynamisch, unabhängig
- > **(total) dezentrale** Netzwerke (P2P)
- > im Gegensatz zu den **zentralen** Netzwerken (Client/Server)

4

Einführung in P2P

■ P2P-Einsatzbereiche

- kein neues Konzept, schon im frühen Internet
 - **Usenet** (zu Beginn mittels UUCP)
 - **DNS**
- heute wieder aktuell und in vielen Bereichen anzutreffen
 - **Instant Messaging** (ICQ, AIM, Jabber)
 - **Filesharing** (KaZaA, Gnutella, eDonkey, Overnet)
 - **Distributed Search Engines** (OpenCola, Copernic)
 - **Group Collaboration** (Groove)
 - **Distributed Computing** (Seti@Home, Parabon)

5

Einführung in P2P

■ Probleme in P2P Netzwerken

- Administration der Peers
- Anonymität und Sicherheit
- Vertraulichkeit
- Ausbreitung von unerwünschten Inhalten
- Verfügbarkeit von Inhalten
- Firewall, NAT umgehen
- Gruppenkommunikation
- Performance, Skalierbarkeit

6

Einführung in P2P

■ Hauptproblem bei P2P

- eine Vielzahl von verschiedenen Anwendungen, die alle ihr eigenes Programmierungs- und Kommunikationsmodell haben
- decken nur jeweils einen Einsatzbereich ab
- existieren unabhängig voneinander => **inkompatibel**

■ Lösung

- eine allgemeine P2P-Spezifikation, ein Framework
- eine **gemeinsame Grundlage**, um P2P-Anwendungen zu entwickeln

7

JXTA

■ Entwicklungsgeschichte von JXTA

- JXTA = "**juxtaposition**" (nebeneinander stellen)
- Konzept von Bill Joy (Chef Wissenschaftler bei Sun)
- erste Veröffentlichung vom 15.02.2001
- Web-Site (www.jxta.org) am 25.04.2001 eröffnet
- JXTA ist eine Spezifikation, die die Grundkonzepte zum Erstellen von P2P-Anwendungen definiert
- Beginn eines Open Source Projekts als Referenzimplementierung der Spezifikation, in Java
- > **Projekt JXTA**

8

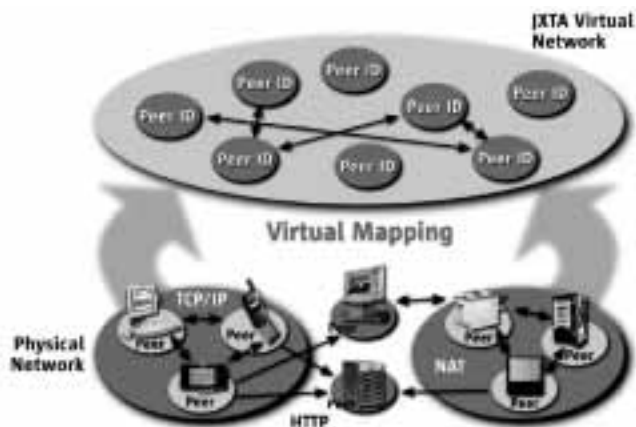
JXTA

■ Die 3 Hauptziele von JXTA

- **Interoperabilität** (*Interoperability*)
 - > um Peers, die verschiedenste Dienste anbieten, aufzufinden und mit ihnen kommunizieren zu können
- **Plattform Unabhängigkeit** (*Platform Independence*)
 - > unabhängig von Programmiersprachen, Kommunikationsprotokollen, und Betriebssystemen
- **Allgegenwärtigkeit** (*Ubiquity*)
 - > kann auf jedem digitalen Gerät eingesetzt werden

9

JXTA



10

JXTA

■ JXTA Architektur - 3 Hauptebenen

- **Core Layer**

-> enthält das Basiswissen für alle P2P-Operationen, beinhaltet die Funktionalität der JXTA Basisprotokolle

- **Service Layer**

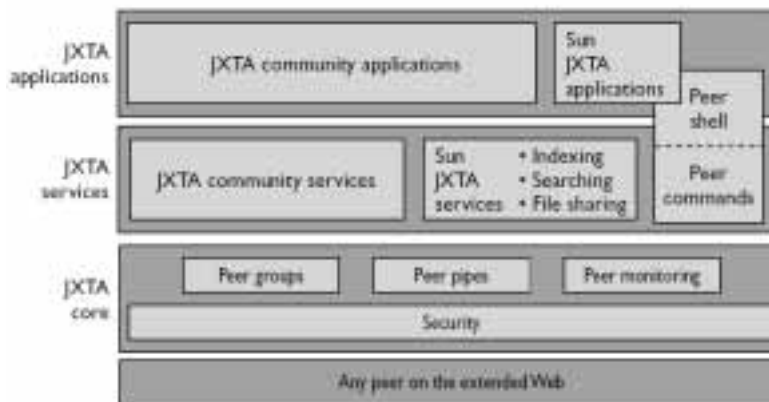
-> enthält die Dienste, die ein P2P-Programm benötigt

- **Application Layer**

-> Benutzerschnittstelle (UI), nutzt die Dienste

11

JXTA



12

JXTA - Core Layer

■ Funktion & Bestandteile

- definiert die grundlegende Funktionalität für die P2P-Dienste und Anwendungen, z.B.
 - Basiskommunikation und Erkundungsmechanismus
 - grundlegende Sicherheitskonzepte und Mitgliedschaftsmechanismen
 - grundlegende Monitoring-Mechanismen

-> **Konzepte**

-> **Protokolle**

13

JXTA - Core Layer - Konzepte

■ Bestandteile des Core Layers

- | | |
|---------------------------|----------------------|
| - Peers | - Messages |
| - Peer Groups | - Identifiers |
| - Network Services | - Credentials |
| - Pipes | - Contents |

14

JXTA - Core Layer - Konzepte

■ Peer

- jeder Teilnehmer am Netzwerk, der JXTA benutzt
- besitzt eine eindeutige ID, zum sicheren Adressieren
- interagiert zufällig mit anderen Teilnehmern
- kann selber Dienste anbieten
- kann Informationen zwischenspeichern
- kann mehrere Netzwerkschnittstellen haben
- interagiert typischerweise nur mit einer kleinen Anzahl an anderen Peers
- **verschiedene** Typen -> minimal, simple, relay, rendezvous Peer

15

JXTA - Core Layer - Konzepte

- **Relay Peers** (Nachrichten routen, Firewalls überwinden,...)
- **Rendezvous Peers** (Adv. cachen, Nachrichten verteilen,...)

■ Peer Group

- Ansammlung von Peers mit **gemeinsamen** Interessen
- bieten eine sichere, spezialisierte, beobachtbare Umgebung
- bieten verschiedene **Core Services**, z.B.
 - > Discovery-, Membership-, Pipe-, Resolver-, Monitoring Service
- ein Peer kann verschiedenen Gruppen beitreten
- alle Peers gehören der World Peer Group an

16

JXTA - Core Layer - Konzepte

■ Pipes

- **virtuelle Kommunikationskanäle**, zum Senden & Empfangen von Nachrichten
- asynchron, unidirektional und unsicher
- unidirektional, d.h. es gibt Input Pipes & Output Pipes
- die Bindung geschieht dynamisch
- verschiedene Basistypen:
 - > **Point-to-Point Pipes**
 - > **Propagate Pipes** (Point-to-Multipoint)
- Erweiterung:
 - > bidirektionale, zuverlässige, sichere & selbstdefinierte Pipes

17

JXTA - Core Layer - Konzepte

■ Advertisements

- alle Netzwerkressourcen werden als Advertisements dargestellt (*Peers, Peer Groups, Pipes, Dienste...*)
- **Metadaten** zum Beschreiben von Ressourcen, in **XML**
- Advertisements werden verteilt, um verfügbare Ressourcen **bekannt zu machen** und sie so **finden zu können**
- haben eine Lebenszeit

■ Network Services

- **Peer Services**
- **Peer Group Services**

18

JXTA - Core Layer - Konzepte

■ Messages

- Grundformat für den Datenaustausch zwischen den Peers
- in **XML** oder **binär**, vom Einsatzzweck abhängig
- XML, weil sprachunabhängig, selbstbeschreibend
- besteht aus Name/Wert Paaren
- enthält Credentials

■ Identifiers

- zum Eindeutigen identifizieren - lokalisieren vom Peers, Pipes, usw.
- Darstellung als **URNs** (Uniform Resource Names), **UUIDs**
- eindeutig, einzigartig, beständig, ortsunabhängig

19

JXTA - Core Layer - Konzepte

- Beispiel ID: urn:jxta:uuid-59616261646162614A7874.....DA53903

■ Credential

- Informationen, die den Sender gegenüber dem Empfänger **identifizieren**
- Beispiel:
 - eine Unterschrift
 - Informationen zur Verschlüsselung eine Nachricht

■ Content

- **jegliche Daten**, die zwischen den Peers ausgetauscht werden können (Codat -> Code und Daten)
- ist eindeutig identifizierbar

20

JXTA - Core Layer - Protokolle

- Die 6 Basisprotokolle die JXTA definiert
 - *Peer Resolver Protocol (PRP)*
 - *Peer Discovery Protocol (PDP)*
 - *Peer Information Protocol (PIP)*
 - *Pipe Binding Protocol (PBP)*
 - *Peer Endpoint Protocol (PEP)*
 - *Rendezvous Protocol (RVP)*

21

JXTA - Core Layer - Protokolle

- Peer Resolver Protocol
 - stellt einen *allgemeinen* Weg dar, um *Anfragen* an andere Peers zu richten und *Antworten* darauf zu empfangen
 - Anfragen/Antworten sind an einen bestimmten "handler" gebunden und werden von diesem bearbeitet
- Peer Information Protocol
 - eingesetzt, um *Informationen* über einen Peer *abzufragen*
 - beim Peer Monitoring eingesetzt (z.B. zum Überprüfen der Erreichbarkeit und der Auslastung eines Peers)

22

JXTA - Core Layer - Protokolle

■ Peer Discovery Protocol

- ermöglicht es **Ressourcen** eines Peers zu veröffentlichen und nach veröffentlichten Ressourcen zu **suchen**
- Broadcast im LAN (Subnetz), Rendezvous/Relay im WAN
- Einsatz von **TTL** und überprüfen der **MessageID**, um das endlose Kreisen von Nachrichten zu verhindern

■ Pipe Binding Protocol

- zum **Einrichten** von **Pipes**
- bindet die Enden der Pipes an die jeweiligen Peer Endpoints

23

JXTA - Core Layer - Protokolle

■ Peer Endpoint Protocol

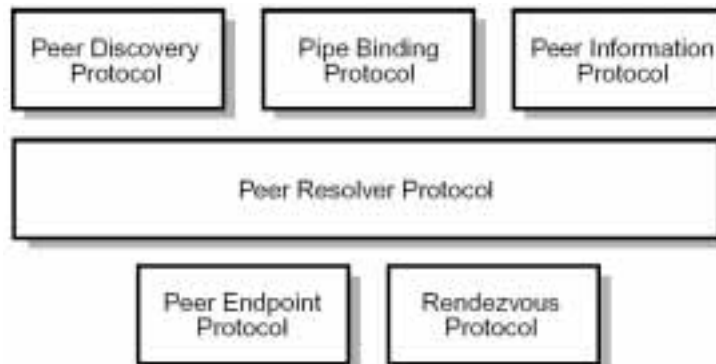
- zum **dynamischen Finden** von **Routen**, die es dann ermöglichen Nachrichten zu anderen Peers zu verschicken
- **Informationen** zu Routen entweder aus **lokalem Cache** oder von **Relay Peers** abgefragt
- nutzt den Relay Service für Peers hinter einer Firewall

■ Rendezvous Protocol

- zum **Verbreiten** von **Nachrichten** in einer Peer Group
- Rendezvous Peers können damit Nachrichten an alle ihre Klienten und andere Rendezvous Peers weiterleiten

24

JXTA - Core Layer - Protokolle



25

JXTA - Service Layer

■ Funktion

- **beinhaltet** die **Dienste**, die auf dem Core Layer aufbauen
- es ist nicht notwendig, dass jeder Peer alle Dienste unterstützt
- Dienste können von jedem selbst entwickelt werden

- es gibt bereits einige fertig gestellte Dienste, z.B.
 - **cms** (Content Management System)
 - **jxta-rmi**

26

JXTA - Application Layer

■ Funktion

- diese Ebene ist die **Benutzerschnittstelle** (UI) und ermöglicht die Nutzung der Services
- dient zum Einordnen der registrierten Anwendungen

- einige bekannte Anwendungen sind z.B.
 - **JXTA Shell**
 - **myJxta** (früher InstantP2P)
 - **vop2p** (Voice over P2P)

27

JXTA - Anwendungen

■ JXTA Shell

- fester Bestandteil der JXTA-Architektur
- **zeilenorientierter Kommandointerpreter**, Konsole
- erlaubt es **direkt** mit der **JXTA** Plattform zu **interagieren**
- enthält Kommandos für die verschiedensten Aufgaben, z.B.
 - zum Erstellen von Advertisements
 - um mit anderen Peers zu kommunizieren

- kann beliebig um weitere Kommandos erweitert werden

28

JXTA - Anwendungen

■ myJxta

- dient zur Demonstration der Möglichkeiten von JXTA
- ermöglicht ein einfaches navigieren im JXTA Netzwerk
- bietet:
 - **Instant Messaging**
 - **Filesharing**
 - **Secure Chat**

29

Fazit

- **P2P**
 - ein aktuelles Thema in Forschung und Unternehmen
 - nach anfänglichem Hype wieder etwas abgekühlt und es wird nun versucht brauchbare Anwendungen zu entwickeln
 - ist eine Alternative zu Client/Server Architekturen
- **JXTA**
 - bietet ein interessantes Framework zur Erstellung von interoperablen P2P-Anwendungen (plattformunabhängig)
 - erfährt ein konstante Verbesserung und Weiterentwicklung
 - große Entwicklergemeinde

30