

6 Grundkonzepte

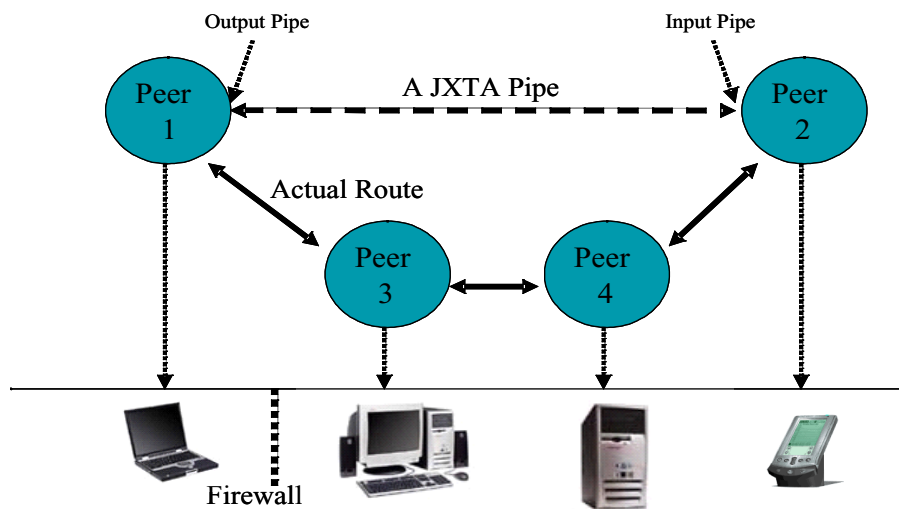
- Transportmechanismen
 - ◆ *Endpoint* - Start- und Endpunkt von Nachrichten aus Applikationssicht
 - Ergibt sich aus der Peer ID und allen Kontaktadressen
 - ◆ *Pipe* - Unidirektionale, asynchrone virtuelle Verbindung zwischen zwei oder mehr Endpoints
 - ◆ *Message* - Container für Daten die via Pipes zwischen Endpoints ausgetauscht werden
 - Message entspricht einem Datagramm
 - Jede Message verfügt über einen Umschlag (Sender und Empfänger Endpoint), einer variablen Anzahl von Protokoll spezifischen Headern sowie den Daten
 - Daten werden als eine geordnete Menge von (Name, Typ, Wert)-Tupel übertragen

6 Grundkonzepte

- Pipes
 - ◆ Virtuelle Verbindungen die zu verschiedenen Zeitpunkten mit unterschiedlichen Peer-Endpoints verbunden sein können
 - ◆ Jede Pipe kann durch eine JXTA ID eindeutig indentifiziert werden
 - ◆ Pipes sind unidirektional und besitzen mit *Output Pipe* eine Datenquelle und mit *Input Pipe* eine Datensenke
 - ◆ Es gibt zwei Arten von Pipes:
 - *Point-to-Point Pipes* für unidirektionale, asynchrone Verbindungen
 - Es gibt kein Acknowledgment oder Reply
 - Antworten werden über eine umgekehrt gerichtete Pipe versendet
 - *Propagate Pipes* für Multicast-Verbindungen

6 Grundkonzepte

■ Pipes (Fortsetzung)



6 Grundkonzepte

■ Advertisement

- ◆ XML basierte Metadaten-Beschreibungen angebotener Netzwerk-Ressourcen, eines Peers oder einer Peer Group
- ◆ Folgende Arten von Advertisements sind standardisiert
 - Peer, Peer Group, Pipe, Service, Rendezvous, Peer Endpoint
- ◆ Advertisements können weitere Advertisements beinhalten
- ◆ Advertisements können beliebig durch weitere Metadaten ergänzt werden
- ◆ Rendezvous-Peers unterstützen die Suche und Verbreitung von Advertisements
- ◆ Advertisements gelten nur zeitlich begrenzt

6 Grundkonzepte

■ Advertisement einer Peer Group

```
<?xml version="1.0" ?>
<!DOCTYPE jxta:PGA>

<jxta:PGA xmlns:jxta="http://jxta.org">

  <GID> urn:jxta:jxta-NetGroup</GID>

  <!-- Module Specification Id -->
  <MSID>urn:jxta:uuid-DEADBEEF2332342342...</MSID>

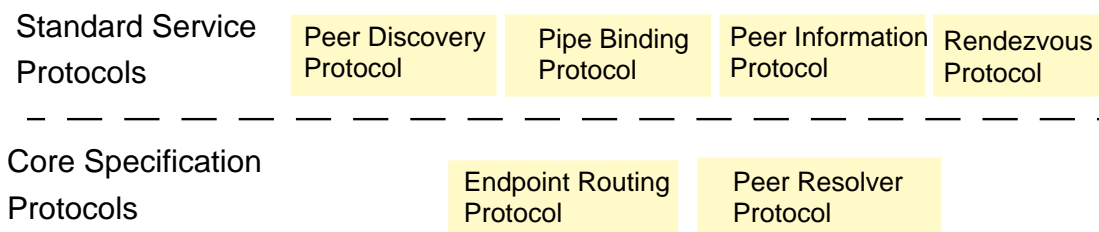
  <Name>NetPeerGroup</Name>

  <Desc>NetPeerGroup by default</Desc>

</jxta:PGA>
```

7 Protokolle

- Eine Suite von 6 Protokollen, die es ermöglichen sollen ad hoc ein Peer-to-Peer Netzwerk zu etablieren
- Die Suite kann in zwei Gruppen unterteilt werden
 - ◆ *Core Specification Protocols* - Minimales Protokollset für JXTA-Peers
 - ◆ *Standard Service Protocols* - Erweitertes Protokollset für bessere Interoperabilität und mehr Funktionalitäten



8 Endpoint Routing Protocol (ERP)

- Ermöglicht es einem Peer dynamisch die Route zu einem anderen Peer über mehrere Hops hinweg zu ermitteln
- Verändert sich die Topologie des Netzwerks und eine Route bricht zusammen ermöglicht das Protokoll den Aufbau einer neuen Route
- Verwendungszwecke
 - ◆ Integration von Rechnern die via NAT adressiert werden oder durch eine Firewall geschützt sind
 - ◆ Umsetzung zwischen verschiedenen Transportprotokollen
 - ◆ Routing innerhalb von Ad Hoc Netzwerken (z. B. Bluetooth)

8 Endpoint Routing Protocol (ERP)

- Allgemeiner Ablauf
 - ◆ Soll eine Nachricht versendet wird zuerst ermittelt ob im lokalen Cache ein passendes Route-Advertisement vorliegt
 - ◆ Ist dies nicht der Fall wird eine *Route Query Message* an Router/Relay Peers versendet
 - ◆ Verfügt ein Router-Peer über die gesuchte Route sendet er diese als eine Auflistung von *Hops* in Form einer *Route Response Message* an das anfragende Peer
 - ◆ Die Nachricht wird durch die Routinginformation ergänzt und kann nun entlang der Route an das Zielpaar versendet werden
 - ◆ Die Routinginformation wird entlang der Route aktualisiert und von den beteiligten Peers zwischengespeichert

9 Peer Resolver Protocol (PRP)

- Erlaubt es generische *Dienstanfragen* zu stellen
- Jede Anfrage besitzt einen spezifischen Namen der bei der Verarbeitung einer Anfrage dazu dient einen entsprechenden Handler zu identifizieren
 - ◆ Handler-Namen setzen sich in der Regel aus Dienst-Name, Group-Id und einer zusätzlichen Komponente zusammen
- Jede Dienstanfrage kann potenziell an alle Mitglieder einer Peer Group weitergeleitet werden

9 Peer Resolver Protocol (PRP)

- Allgemeiner Ablauf
 - ◆ Ein Peer versendet eine Anfrage als *Resolver Query Message* mit dem Rendezvous Protocol an alle verbundenen Peers und Rendezvous Peers
 - ◆ Jedes Peer das die Nachricht empfängt überprüft ob es einen entsprechenden Handler betreibt und stellt die Nachricht zu.
 - ◆ Der Handler überprüft die Anfrage und sendet eine *Resolver Response Message* wenn er sie beantworten kann.
 - ◆ Unterbindet der Handler nicht explizit die weitere Zustellung und handelt es sich um ein Rendezvous Peer wird die Nachricht weitergeleitet.

10 Peer Rendezvous Protocol (RVP)

- Organisiert die Verteilung von Nachrichten in einer Peer Group
- Folgende Teilaufgaben werden durch das Protokoll adressiert
 - ◆ Regelt wie sich Peers bei Rendezvous Peers registrieren
 - ◆ Vernetzung der Rendezvous Peers
 - ◆ Verteilung von Nachrichten zwischen den Rendezvous Peers
- Wird von dem Peer Resolver Protocol und Pipe Binding Protocol genutzt um Nachrichten zu versenden, setzt auf dem Endpoint Routing Protocol auf

10 Peer Rendezvous Protocol (RVP)

- Verbindung von Peers und Rendezvous Peers
 - ◆ Es werden Leases genutzt um zeitbeschränkt die Nutzung eines Rendezvous Peers als Vermittler zu ermöglichen.
 - ◆ Ein Lease kann von beiden Seiten zu beliebigen Zeitpunkten aufgehoben werden.
- Allgemeiner Ablauf
 - ◆ Peer sendet eine *Lease Request Message* an einen Rendezvous Peer
 - ◆ Sollte der Rendezvous Peer noch freie Kapazitäten besitzen antwortet dieser mit einer *Lease Granted Message*
 - ◆ Wird der Lease nicht mehr benötigt so sendet der Peer eine *Lease Cancel Message*

10 Peer Rendezvous Protocol (RVP)

- Vernetzung der Rendezvous Peers
 - ◆ Zielsetzung ist das jedes Rendezvous Peer eine möglichst aktuelle und konsistente Sicht auf die Gruppe der Rendezvous Peers (*Peer View*) besitzt
- Allgemeiner Ablauf
 - ◆ Jedes Rendezvous Peer versendet *PeerView Messages* als *Probe* Nachricht an andere Rendezvous Peers. Eine *Probe PeerView Message* enthält ein *Rendezvous Advertisement* des anfragenden Peers
 - ◆ Die Probe Anfrage wird durch ein *Response PeerView Message* beantwortet. Diese kann weitere *Rendezvous Advertisements* beinhalten.

10 Peer Rendezvous Protocol (RVP)

- Kontrolle der Ausbreitung von Nachrichten innerhalb der *Peer View*
- Alle Nachrichten die innerhalb der *Peer View* vermittelt werden, sind in eine *RendezVousPropagat Message* eingebettet
- Die Parameter der *RendezVousPropagat Message* ermöglichen
 - ◆ Vermeidung von Schleifen und Dublikation
 - ◆ Kontrolle der Ausbreitung (TTL)

11 Peer Discovery Protocol (PDP)

- Veröffentlichen von Ressourcen eines Peers via Advertisement
- Suche nach Ressourcen (Peers, Peer Groups, Pipes ...)
- Allgemeiner Ablauf
 - ◆ Zur Suche nach einer Ressource wird eine *DiscoveryQuery Message* versendet
 - Angabe der Art der gesuchten Ressource
 - Maximale Anzahl der Antworten
 - Attribute zur Spezifikation der Suche
 - ◆ Sollte die Anfrage beantwortet werden können wird eine *DiscoveryResponse Message* versendet
 - Anzahl der Treffer und Antworten
- *DiscoveryResponse Message* dienen auch zum Veröffentlichen von Ressourcen

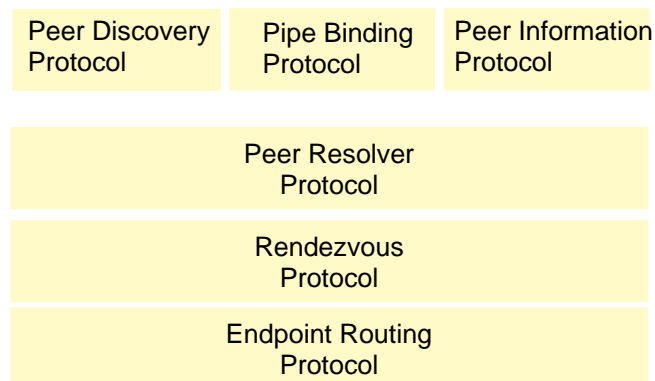
12 Peer Information Protocol (PIP)

- Dient zu Ermittlung von Statusinformationen
- Ablauf
 - ◆ Anfrage nach Statusinformationen durch eine *PeerInfoQuery Message* mit optionaler, zusätzlicher Anfrage
 - ◆ Antwort erfolgt durch eine *PeerInfoResponse Message*
 - Uptime, Verkehr, Last usw. sowie zusätzliche Informationen
- PIP verwendet PRP für die Vermittlung von Anfragen

13 Pipe Binding Protocol (PIB)

- Ermöglicht es virtuelle Verbindungen aufzubauen
- Auf der Empfangsseite können ein oder mehrere Peers verbunden werden
- Allgemeiner Ablauf
 - ◆ Es wird ein Pipe Advertisement veröffentlicht
 - ◆ Ein Peer bindet eine Input Pipe zur ID des Advertisements
 - ◆ Ein anderes Peer sucht eine entsprechende Pipe mittels einer *PipeResolver Message* mit dem Message Typ *Query*
 - ◆ Ein Peer welches die Input Pipe gebunden hat versendet eine *PipeResolver Message* mit dem Message Typ *Answer*
 - ◆ Anfragende Peer kann nun eine Output Pipe binden

14 Protokollstack

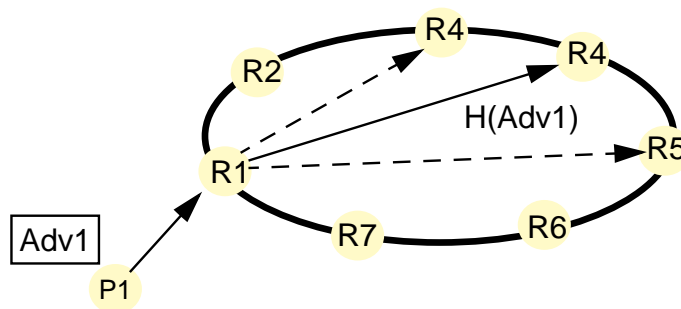


15 Bewertung des Protokollstack

- Geringer Overhead
- Geringe Anforderungen an das Transportsystem
 - ◆ unidirektionale, asynchrone Verbindungen
- Geringe Anforderungen an die Ausführungsumgebung des Peers
 - ◆ Je nach Anwendung und Peer muss nur ein Untermenge der Protokolle implementiert werden
- Geeignet für eine Vielzahl von P2P-Anwendungen
 - ◆ Dynamische Netzwerkstruktur und ständiger Wechsel von Peers

16 Rendezvous Netzwerk der Referenzimplementierung

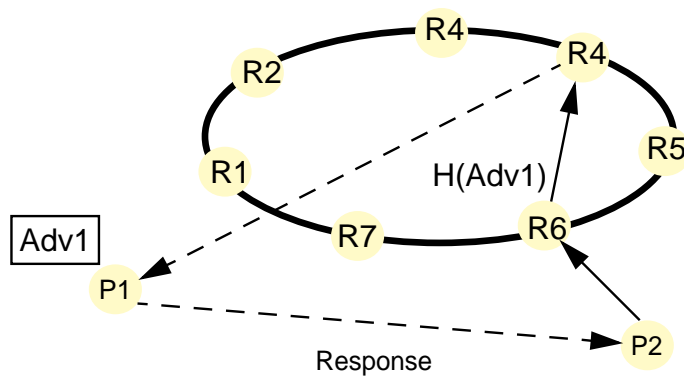
- Veröffentlichung eines Advertisement



- ◆ Rendezvous Peers verwalten nur noch einen Advertisement-Index
- ◆ Für jedes Advertisement wird ein Hashwert bestimmt der festlegt welche Rendezvous Peers Informationen über das Advertisement speichern

16 Rendezvous Netzwerk der Referenzimplementierung

- Suche nach einem Advertisement



- ◆ Das zuständige Rendezvous Peer R4 leitet die Anfrage an P1 weiter.

17 Zusammenfassung

- JXTA versucht eine Standardumgebung für Peer-to-Peer Anwendungen zu etablieren
 - ◆ Ist dies schon jetzt sinnvoll?
- JXTA wird in der Regel nicht immer die beste und effizienteste Lösung bieten können um eine Peer-to-Peer Anwendung zu entwickeln