Introduction to JXTA

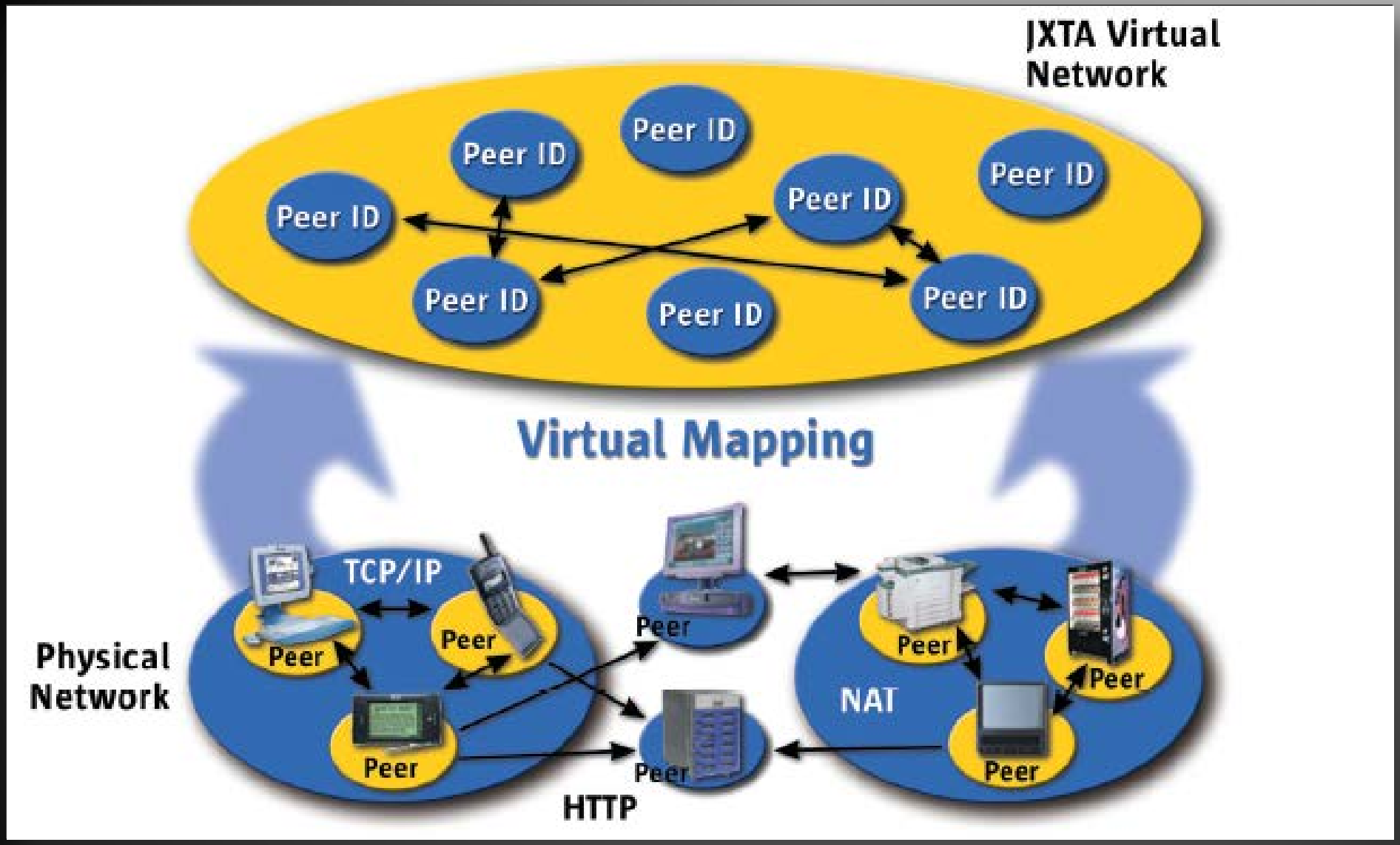
Chris Panayiotou

What is JXTA

- An open source peer-to-peer protocol specification begun by Sun in 2001
- JXTA protocols are defined as a set of XML messages which allow any device connected to a network to exchange messages and collaborate independently of the underlying network topology
- As JXTA is based upon a set of open XML protocols, it can be implemented in any modern computer language
  - Language, OS, network agnostic
  - Implementations are currently available for Java SE, C/C++, C# and Java ME
- JXTA peers create a virtual overlay network which allows a peer to interact with other peers even when some of the peers and resources are behind firewalls and NATs or use different network transports
  - Each resource is identified by a unique ID, a 160 bit SHA-1 URN in the Java binding, so that a peer can change its localization address while keeping a constant identification number
- Currently the JXTA/JXSE project is migrating from Java.net to Project Kenaï
  - Two new separate project on Kenai: JXTA for the protocols and JXSE for its implementation in the Java programming language
  - http://jxta.kenai.com/

Why Project JXTA?

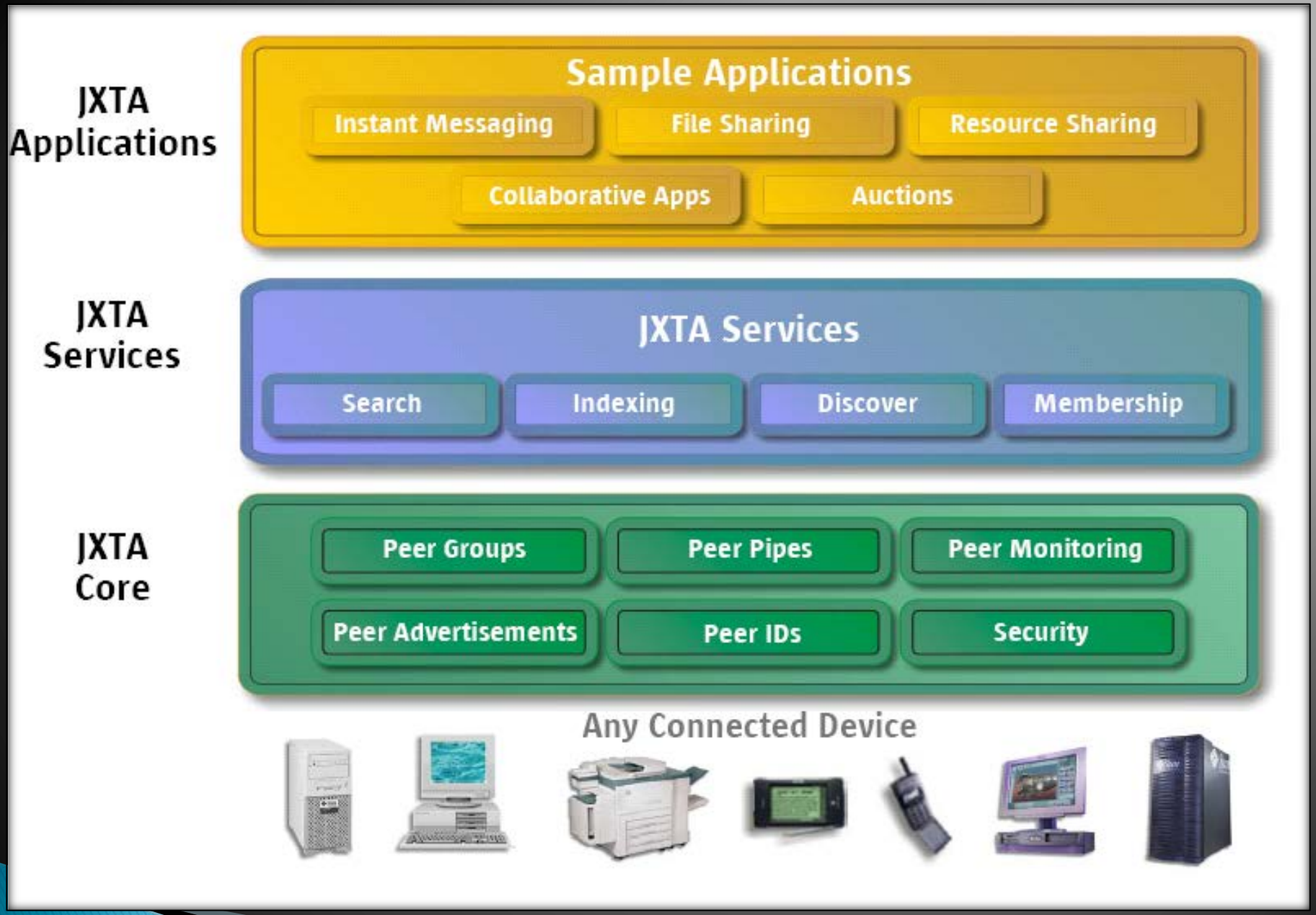
- As the Web continues to grow in both content and the number of connected devices, peer-to-peer computing is becoming increasingly popular
  - E.g. file sharing, distributed computing and instant messenger services
- While each of these applications performs different tasks, they all share many of the same properties, such as:
  - Discovery of peers
  - Searching
  - File or data transfer
- Currently, application development is inefficient, with developers solving the same problems and duplicating similar infrastructure implementation
  - And, most applications are specific to a single platform and are unable to communicate and share data with other applications



Overlay Network Example

JXTA Goals

- To provide a platform with the basic functions necessary for a P2P network.
- Also, to overcome potential shortcomings in many of the existing P2P systems:
  - *Interoperability* — JXTA technology is designed to enable peers providing various P2P services to locate each other and communicate with each other
  - *Platform independence* — JXTA technology is designed to be independent of programming languages, transport protocols, and deployment platforms
  - *Ubiquity* — JXTA technology is designed to be accessible by any device with a digital heartbeat, not just PCs or a specific deployment platform



JXTA Architecture

Protocols in JXTA

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

Categories of Peers

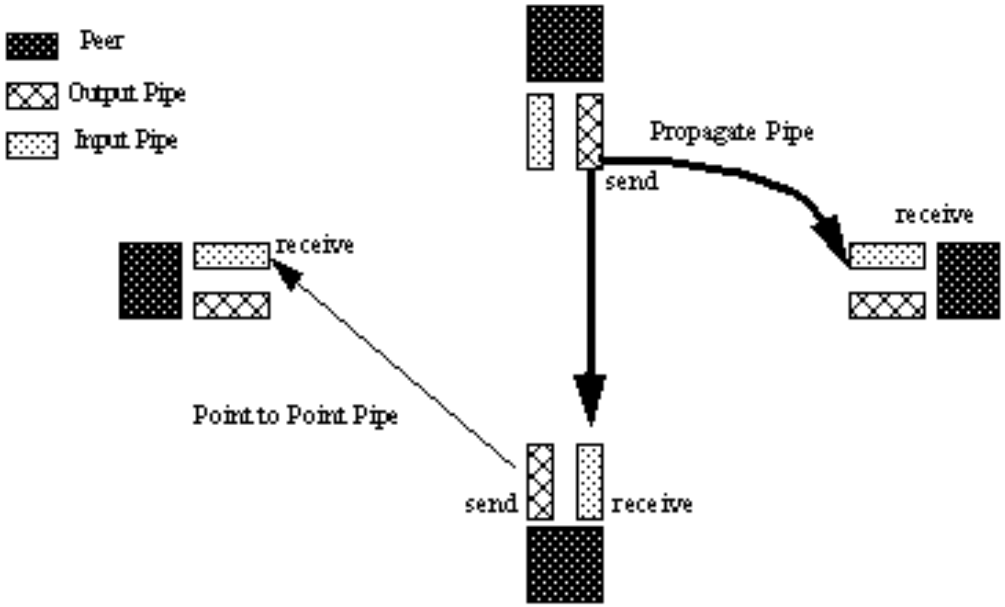
- JXTA defines two main categories of peers: *edge peers and super-peers*
  - The super-peers can be further divided into rendezvous and relay peers
  - Each peer has a well defined role in the JXTA peer-to-peer model
- The edge peers are usually defined as peers which have transient, low bandwidth network connectivity
  - They usually reside on the border of the Internet, hidden behind corporate firewalls or accessing the network through non-dedicated connections
- A Rendezvous peer is a special purpose peer which is in charge of coordinating the peers in the JXTA network and provides the necessary scope to message propagation
  - If the peers are located in different subnets then the network should have at least one Rendezvous peer
- A Relay peer allows the peers which are behind firewalls or NAT systems to take part in the JXTA network
  - This is performed by using a protocol which can traverse the firewall, like HTTP
- Any peer in a JXTA network can be a rendezvous or relay as soon as they have the necessary credentials or network/storage/memory/CPU requirements

Advertisements and Pipes

- An *Advertisement* is an XML document which describes any resource in a P2P network (peers, groups, pipes, services, etc.)
  - The communication in JXTA can be thought as the exchange of one or more advertisements through the network
- *Pipes* are a virtual communication channel used by JXTA to exchange messages and data
  - Pipes are asynchronous, unreliable, and unidirectional
  - There are basically three types of pipes:
    - Unicast
    - Unicast Secure
    - Propagate

Pipes

- Peers can host input pipes (incoming messages), or output pipes (outgoing messages)
- Pipes can be chained to link peers across multiple logical hops, and can be one-to-many
- Pipes are bound to peer ids, not IP address.

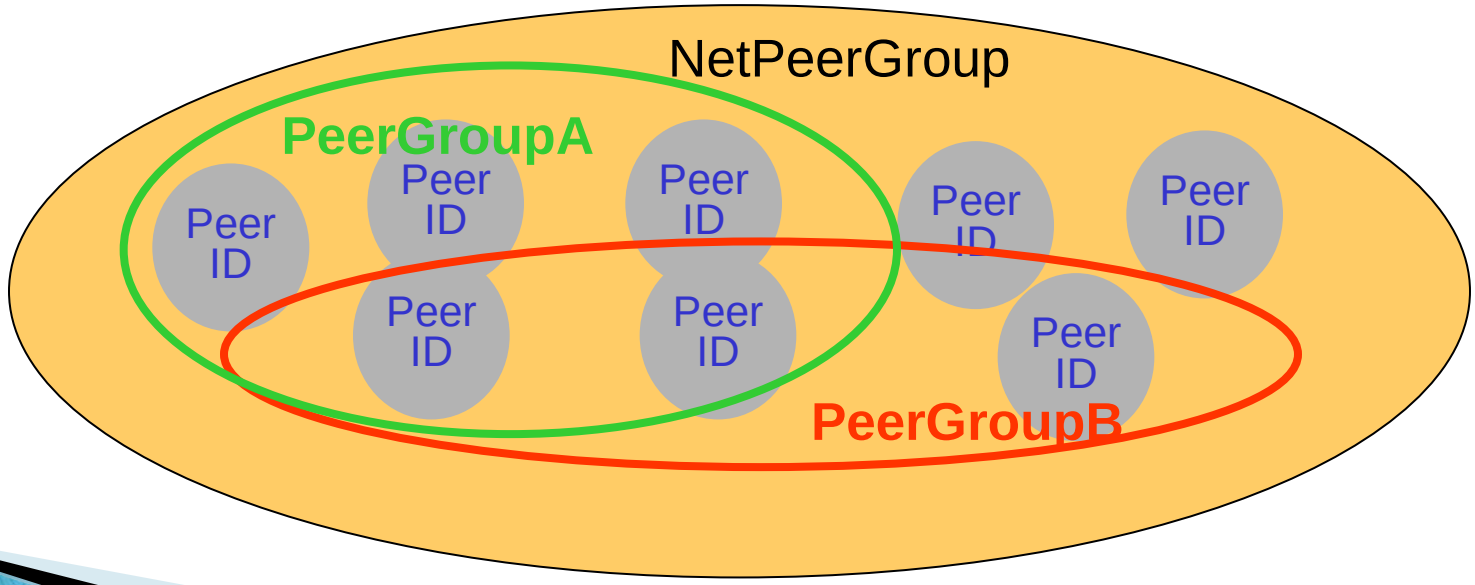


Peer Groups

- A peer group provides a scope for message propagation and a logical clustering of peers
  - In JXTA, every peer is a member of a default group, NetPeerGroup, but a given peer can be member of many sub-groups at the same time
  - A peer may play different roles in different groups; it may act as an edge peer in one group, but a rendezvous in another
- Each group should have at least one rendezvous peer and it is not possible to send messages between two groups

Peer Groups

- Why Peer Groups?
  - Provide a “group” identity (common interest)
  - Create secure & protected domains
  - Scope peer operations (discovery, search, communications)
  - Enable monitoring



Rendezvous Network

- The Rendezvous peers have an optimized routing mechanism which allows an efficient propagation of messages pushed by edge peers connected to them
  - Achieved through the use of a loosely consistent network
- Each Rendezvous peer maintains a Rendezvous Peer View (RPV), a list of known rendezvous peers ordered by the Peer ID
  - There is not any mechanism to enforce the consistency of all RPVs across the JXTA network, so a given RPV can have a temporary or permanent inconsistent view of the other rendezvous peers
  - As soon as there is a low churn rate, that is, a stable network where peers don't join or leave too frequently, the RPV list of each peer will converge as each rendezvous peer exchange a random subset of its RPV with other rendezvous peers from time to time

Rendezvous Network

- When an edge peer publishes an *Advertisement*, the index of this advertisement is pushed to the rendezvous through a system called Shared Resource Distributed Index (SRDI)
- After that, the rendezvous applies a Distributed Hash Table (DHT) function so that it can forward the index to another peer in the RPV list
  - For replication purposes, it will send this index to the neighbors of the chosen rendezvous peer in the RPV list
- The lookup process requires the use of the same DHT function to discover the rendezvous peer which is in charge of storing that index
- Once the rendezvous peer is reached it will forward the query to the edge peer which published the advertisement and this peer will get in touch with the peer which issues the query
  - If the DHT function cannot find a peer which is in charge of the advertisement then the query will be forwarded up and down the RPV list until a match is found, the query is aborted, or it reaches the limits of the RPV list
    - This process is called random walk

Basic Mode of Operation

- The JXTA network consists of a series of interconnected nodes, or peers
- Each node in the JXTA network
  - Has a “peer id” – a “globally” unique ID (UUID)
    - Urn:jxta:idform3-31:08:66:42:67:::91:24::73
  - Is Autonomous and may operate independently of all peers
- Peers can self-organize into peer groups, which provide a common set of services
  - Examples of services that could be provided by a peer group include document sharing or chat applications
- JXTA peers advertise their services in XML documents called advertisements
  - Advertisements enable other peers on the network to learn how to connect to, and interact with, a peer’s services
- JXTA peers use pipes to send messages to one another
  - Pipes are an asynchronous and unidirectional message transfer mechanism used for service communication
  - Messages are simple XML documents whose envelope contains routing, digest and credential information

Discovery Query Example

```
<xs:element name="DiscoveryQuery" type="jxta:DiscoveryQuery"/>

<xsd:simpleType name="DiscoveryQueryType">
        <xsd:restriction base="xsd:string">
                <!-- peer -->
                <xsd:enumeration value="0"/>
        </xsd:restriction>
</xsd:simpleType>

<xs:complexType name="DiscoveryQuery">
        <xs:sequence>
                <xs:element name="Type" type="jxta:DiscoveryQueryType"/>
                <xs:element name="Threshold" type="xs:unsignedInt" minOccurs="0"/>
                <xs:element name="Attr" type="xs:string" minOccurs="0"/>
                <xs:element name="Value" type="xs:string" minOccurs="0"/>
                <!-- The following should refer to a peer adv, but is instead a whole doc for
historical reasons -->
                <xs:element name="PeerAdv" type="xs:string" minOccurs="0"/>
        </xs:sequence>
</xs:complexType>
```

Discovery

- LAN-based (broadcast)
- Invitation (in or out of band, via an advertisement)
- Cascaded (controlled view across discovered peers)
- Rendezvous (napster-esqe)

Resolution

- In general – a service that resolves advertisements into endpoints
- JXTA ships with one implementation – “Rendezvous” – in which hosts serve as switchboards for messages
- More complex/decentralized resolvers are possible, but not specified/provided

Hello JXTA!

```
import ...

public class SimpleJxtaApp {
    public static void main(String args[]) {
    try { // Create, and Start the default jxta NetPeerGroup
        PeerGroup  netPeerGroup =  PeerGroupFactory.newNetPeerGroup();

        // Obtain the peer advertisement
        PeerAdvertisement myPeerAdv = netPeerGroup.getPeerAdvertisement();

        //Get the discovery service
        DiscoveryService discovery = netPeerGroup.getDiscoveryService();

      //Publish the peer advertisement
        discovery.publish(myPeerAdv);
      } catch (PeerGroupException pge) {
          pge.printStackTrace();
      }
}
```

The Future of JXTA

- In November 2010, Oracle officially announced its withdrawal from the JXTA projects
  - No surprise since Sun had progressively reduced its contributions since 2008
- The elections should have been organized around July 2010 by the board
  - Unfortunately, it did not organize these, which leaves the community in a limbo state
- The community voted to move to the Apache Software Foundation (ASF). An application has been filled to the Apache Incubator
  - Unfortunately, this move can be performed only if the JXSE code base can be moved from the Apache 1.1 license to Apache 2.0 license
  - Appropriate requests have been made to Oracle to enable this
- Oracle announced that it would not transfer the JXTA trade name
  - The community voted for a new name: Chaupal.
- The finalization of the move to Kenai has been frozen
- As of August 2011, the JXTA project has not yet been continued or otherwise announced to retain operations, neither a decision was made on the assembly of its Board nor an answer by Oracle regarding a pending request to move the source-code to Apache 2.0 license

Comparison of P2P Technologies

	JXTA	Gnutella	Napster	Freenet/Chord
Key Difference	P2P Platform that allows the deployment of "any" P2P service	P2P Protocol tailored to the needs of file-sharers	P2P Protocol tailored to the needs of file-sharers	Algorithms for Object Location in P2P systems
Architecture	Any (Pure, Rendezvous (hybrid), or centralized)	Pure/Hybrid P2P (started out as Pure but with Limewire's Ultra-peers moving to Hybrid)	Centralized P2P	Structured P2P networks (peers are positioned at well-known places)
Transport	HTTP or TCP/IP	TCP/IP	TCP/IP	mostly TCP/IP
Communication	Sync/Async	Asynchronous	Synchronous	Asynchronous
Data Replication	Open (depends on the Application)	No (downloaded file not considered replication)	No (downloaded file not considered replication)	Required
Lang Bindings	JAVA, C/C++, …	Open source mostly in Java	No API. (proprietary)	Java and C++