# JXTA™ Technology for XML Messaging

## OASIS Symposium
## New Orleans, LA
## 27-April-2004


**Richard Manning**

**Senior Software Architect**

**Advanced Technology & Edge Computing Center**

**Sun Microsystems Inc.**


**www.jxta.org**
**www.java.net**
**www.sun.com**

Sun microsystems
We make the net work.

# Agenda

- Peer-to-Peer Computing
- JXTA™ Technology
  - Virtual network
  - Architecture
  - Concepts & components
- JXTA Messaging & Security
- Interoperability & Future
- Q & A

# What is Peer-to-Peer (P2P)?

- P2P is different things to different people…
  - Sharing files or swapping music
  - Instant messaging & pervasive devices communicating
  - Sharing CPU and storage resources
  - Distributed search and indexing
  - Collaborative work (and play)
  - New forms of content distribution and delivery
- P2P is not…
  - A specific architecture, technology, market, or business model
  - About eliminating servers or centralized services
- P2P is about any device easily connecting “directly” to other devices to enable a more cooperative, or social, style of computing

# What is JXTA Technology?

- An open set of XML-based protocols for creating peer-to-peer style network computing applications and services
  - A virtual network overlay
  - Protocol based --> language, OS, network, and service agnostic technology
  - Defines mechanisms, not policies
  - Open Source project: *www.jxta.org*

# An Open Source Model

- *www.jxta.org*
  - All source, projects, docs, examples are open
- Apache-style software license
  - No barriers to getting started
  - No royalties, no fees, no registration
- Meritocracy
  - The more you've done, the more you're allowed to do

# JXTA License & Governance

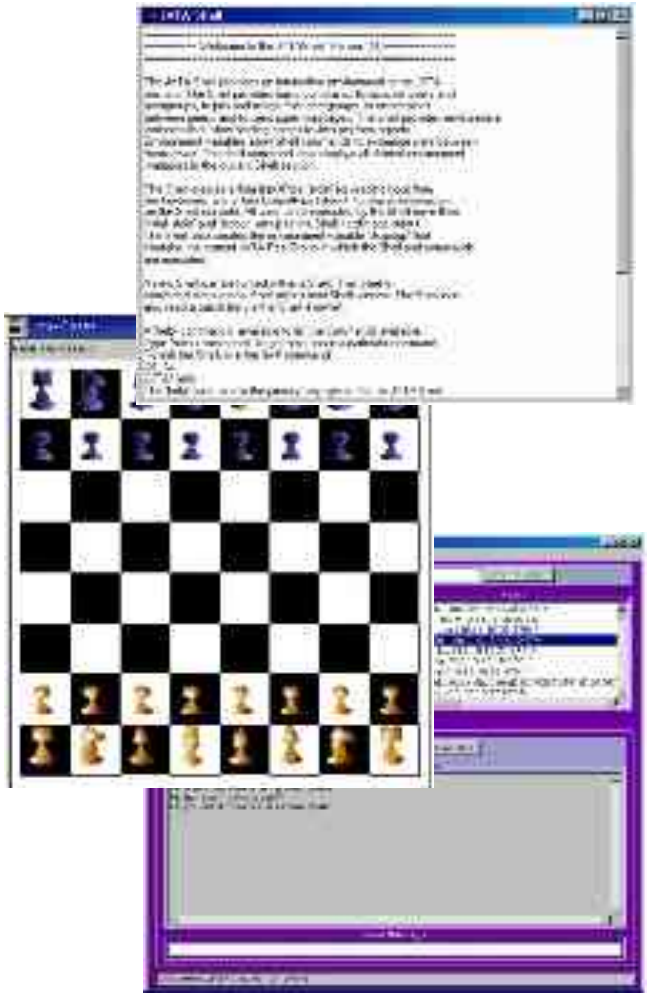
- Source code for Project JXTA has been released to the open source community under a variant of the Apache software License.
- Functionally equivalent to the Apache Software License with minor changes to reflect the Project JXTA name and Sun Microsystems as the original contributor.
- http://www.jxta.org/project/www/license.html
- http://www.jxta.org/project/www/govern.html

# What JXTA Technology Does
## Creating Connected Communities

- Brings devices, services, and networks together
- Enables interactions among highly dynamic resources
- Takes the complexity out of the network and operating environments so developers can quickly build peer-to-peer applications
- Users have better access to content across multiple devices, regardless of location
  - Find it, get it, use it

# JXTA Enables P2P Applications

- Content delivery and sharing
- Communication, collaboration, gaming
- Transactional Web Services
- Resource Sharing

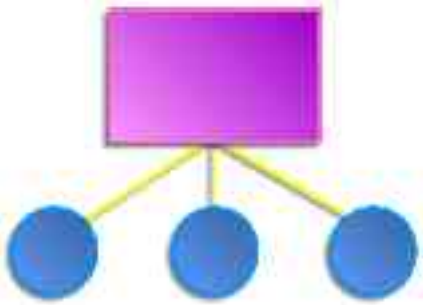


# Problems JXTA Technology Solves

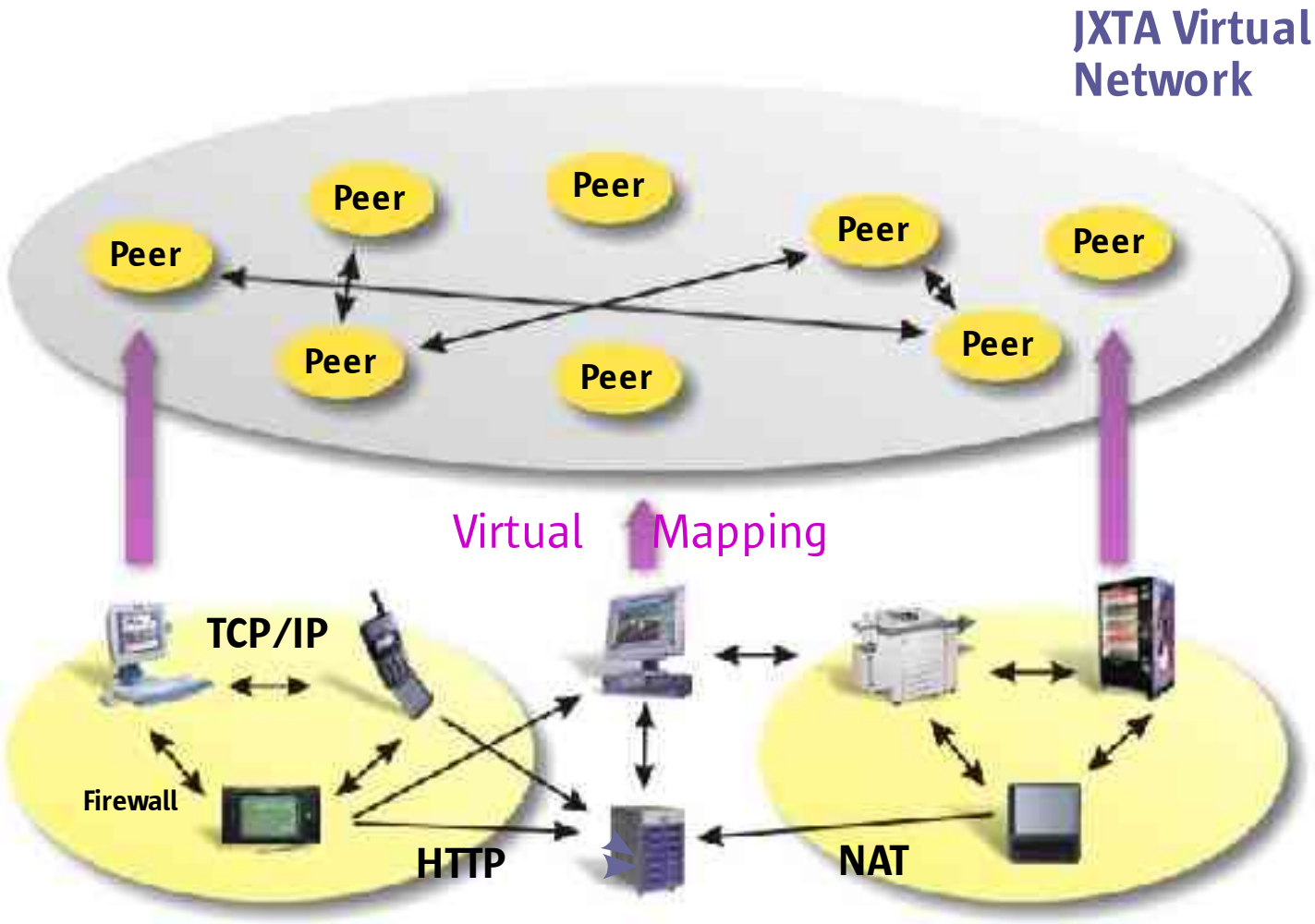
- Provides a set of building blocks that provide a foundation for P2P applications
- Provides an open and interoperable set of protocols that do not have special licensing requirements
- Quick time to market for new products and services

# JXTA Technology Objectives

- Interoperability
  - Across different P2P systems and communities
- Platform independence
  - Programming languages, system platforms, and networking platforms
- Ubiquity
  - Every device with a digital heartbeat
- Security and Monitoring
  - For commercial and enterprise deployment



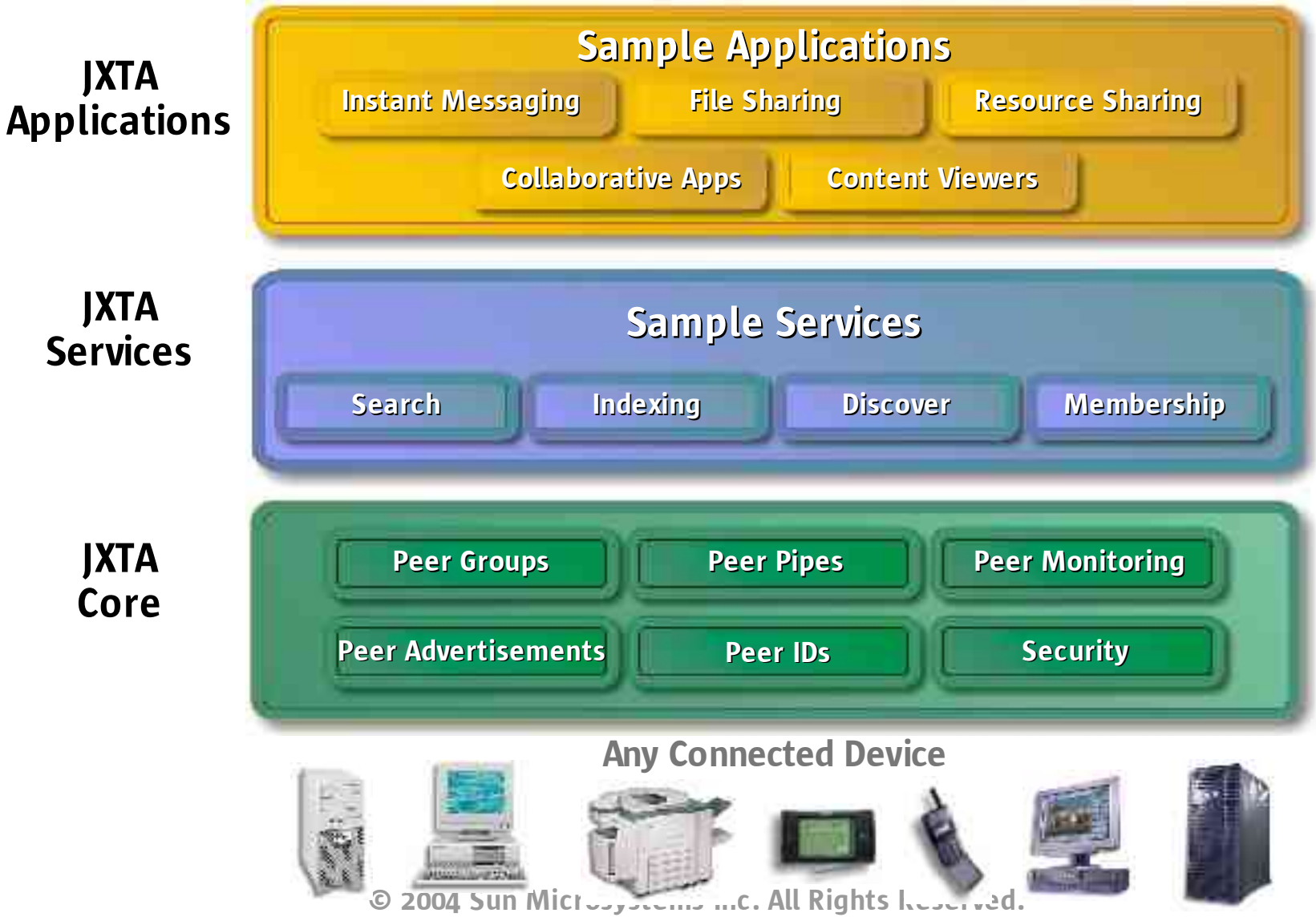
# JXTA Virtual Network



# JXTA Virtual Network Building Blocks

- Uniform peer addressing
  - Peer IDs
- Dynamically configurable peer domains
  - Peer groups
- Uniform resource representation
  - Advertisements
- Virtual communication channels
  - Pipes
- Security and Monitoring

# JXTA Software Architecture



# Peers

- Any networked device that implements one or more JXTA protocols
  - PC, server, PDA, cell phone, etc.
- Operate independently, asynchronously
- Spontaneously discover each other on the network
  - Transient relationships
  - Persistent relationships (peer groups)

# Peer Types

- Micro peer
- Peer
- Super Peer
  - Relay
  - Rendezvous
  - Proxy

# Identifiers

- JXTA IDs uniquely identify resources: peers, peer group, pipes, etc.
- Uniform peer addressing scheme
  - Unique Peer IDs enable peers to be addressed independently of their physical network location
  - Example Peer ID:
    Urn:jxta:uuid-59616261646162614E5047205032
    50338E3E786229EA460DADC1A176B69B731504

# Peer Endpoints

- Network interface(s) published by peer
- Example:
  - TCP/IP (tcp://129.127.29.65:9700)
  - HTTP (http://JxtaHttpClientuuid-...)
- Used to establish point-to-point connections between two peers
- Direct connections not required; intermediary peers can route messages

# Protocols

- JXTA technology defines XML message formats, or protocols, for communication between peers
- Protocols used to discover peers, advertise and discover resources, communicate and route messages, and provide monitoring
- Asynchronous; based on query/response model
- Can be implemented in any language

# JXTA Protocols

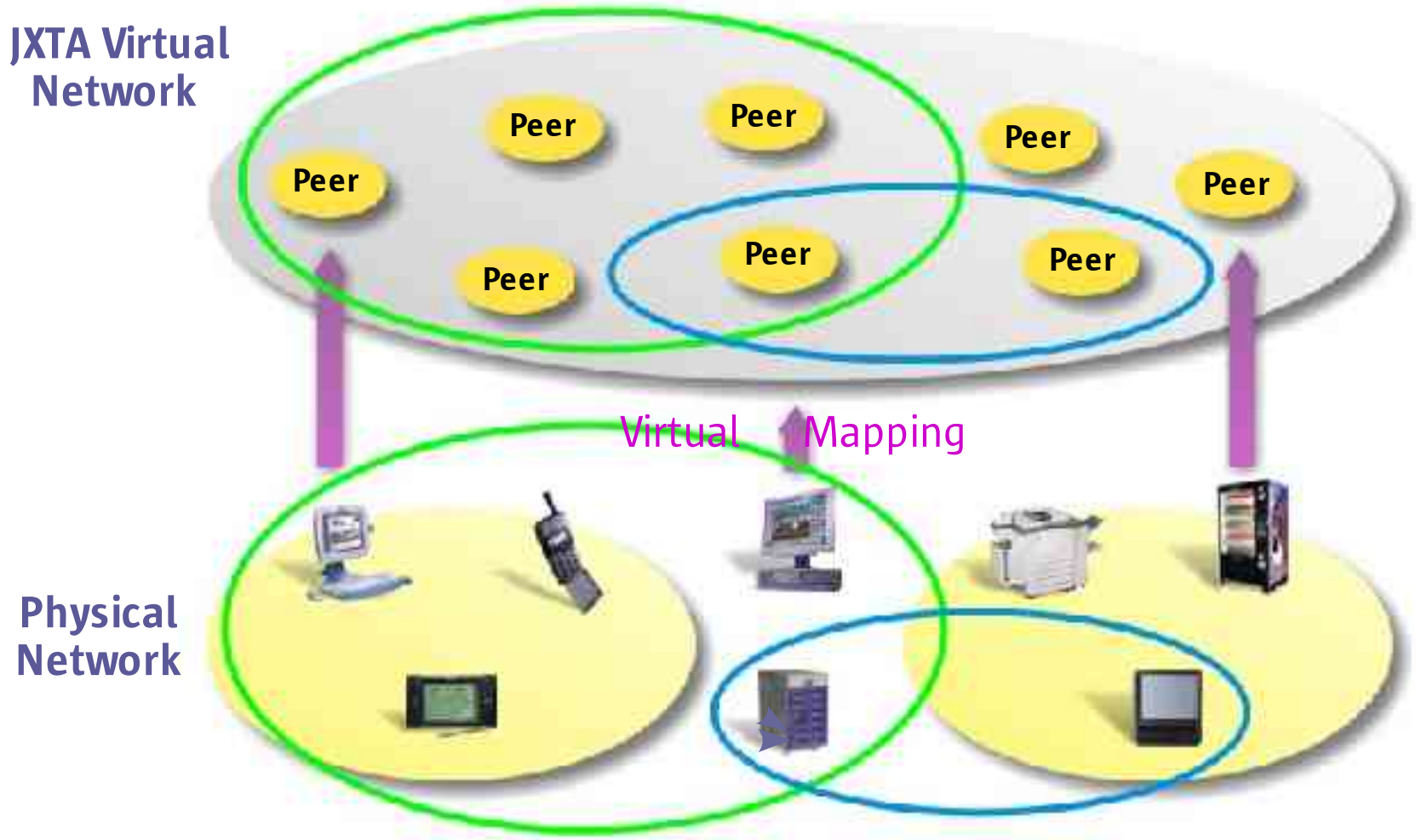
Super Peer

Peer

Micro Peer

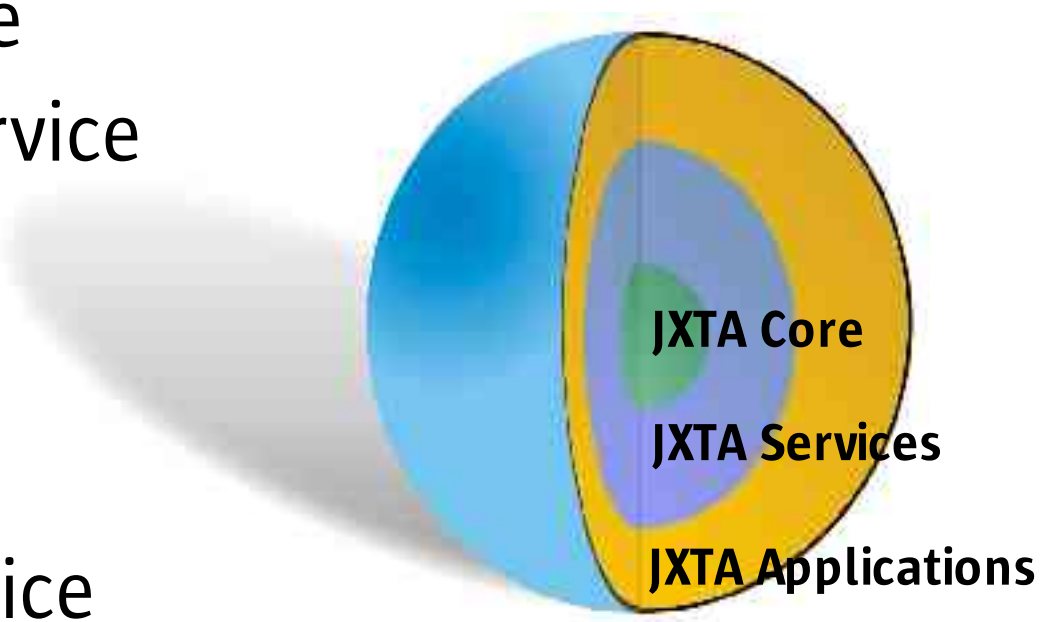
| Core Protocols | Micro Peer | Peer | Super Peer |
|---|---|---|---|
| | | | Peer Rendezvous Protocol |
| | | | Peer Discovery Protocol |
| | | Peer Information Protocol | Peer Information Protocol |
| | | Pipe Binding Protocol | Pipe Binding Protocol |
| | Peer Resolver Protocol | Peer Resolver Protocol | Peer Resolver Protocol |
| | Endpoint Routing Protocol | Endpoint Routing Protocol | Endpoint Routing Protocol |

# Peer Groups



# JXTA Core Peer Group Services

- Discovery Service
- Membership Service
- Access Service
- Pipe Service
- Resolver Service
- Monitoring Service




**Peer Groups are not required to implement all services; can use default net peer group services.**

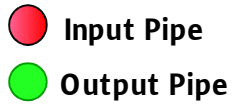
# Why Use Peer Groups?

- Create secure and protected domains
- Scope peer operations
  - Discovery, search, communications
- Provide a “group” identity
  - Group peers sharing a common interest
- Enable monitoring

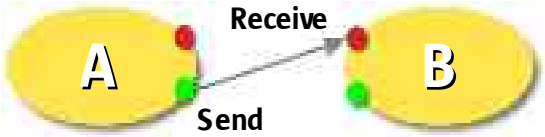
# Pipes

- Used to send/receive messages
- Asynchronous and unidirectional
- Support the transfer of any object
  - Binary code, data strings, etc.
- Dynamically bound
- Virtual communication channels
  - May connect peers that do not have direct physical link
  - Can be bound to more than one peer endpoint

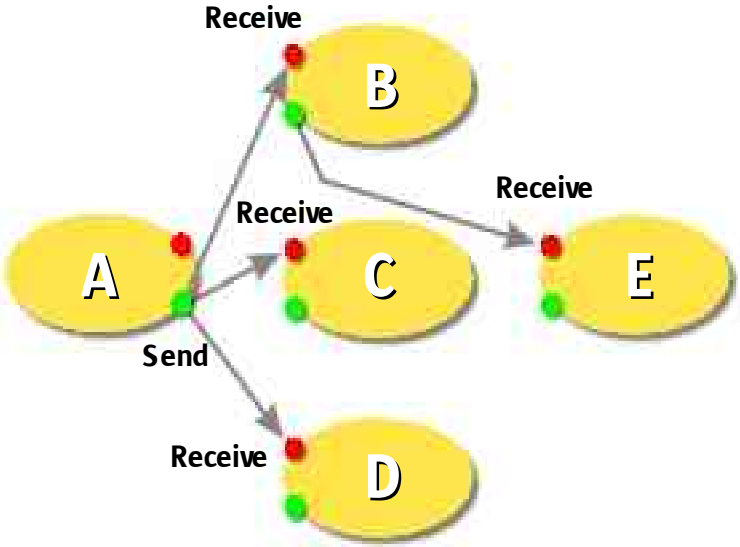
# Pipe Types




- Point-to-Point Pipe
  - Connects exactly two peer endpoints together




- Propagate Pipe
  - Connects one output pipe to multiple input pipes




**Additional pipe types can be created from the core types.**

# Additional Pipe Types

- BiDiPipe
  - JXTA bi-directional Pipe
  - Request-Response
- JXTASockets
  - JXTASocket
  - JXTAServerSocket
  - JXTAMulticastSocket
- May be designated reliable and/or secure reliable!

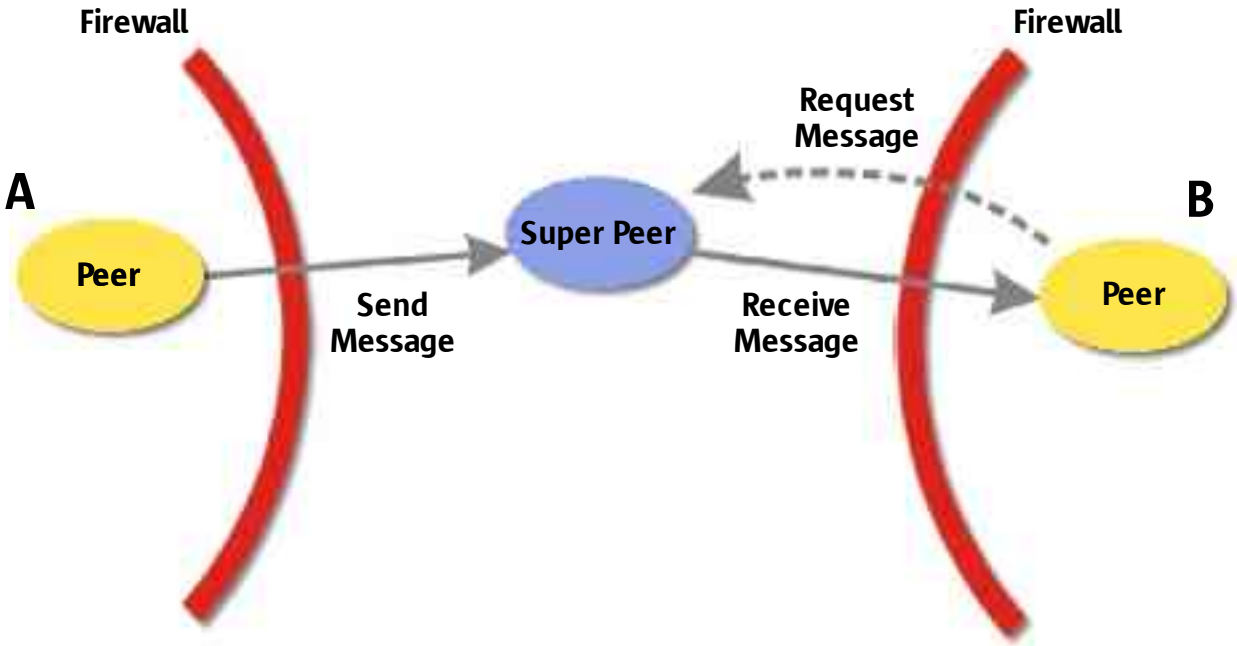
# Pipe Service Protocols

- JXTA Pipe Binding Protocol
  - Mechanism to resolve the location of pipes to a physical peer
  - Decentralized
- Pipe Resolver Protocol
  - Uses dynamic and adaptive search mechanism
  - Attempts at all times to find peers where an instance of the pipe is running

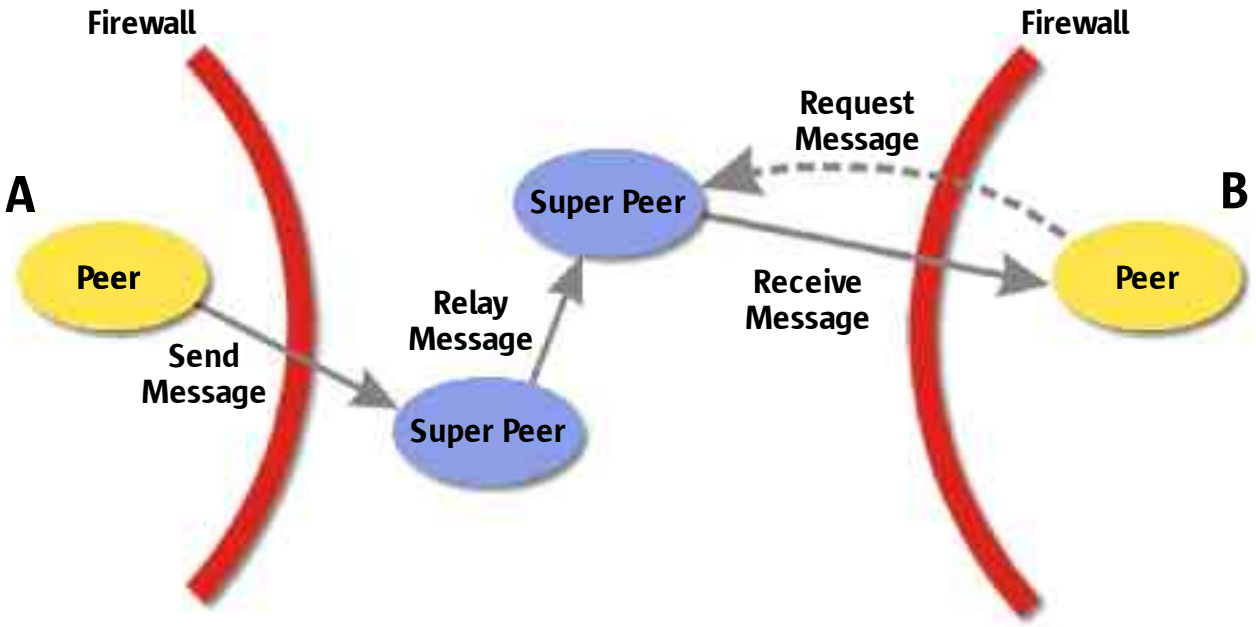
# Messages

- Object sent between JXTA peers; basic unit of data exchange
- Ordered sequence of named/typed contents called Elements
- Contains its own routing information
- XML and binary representations are used

# Message Routing Via Relay Peers



# Message Routing Via Relay Peers



# Services

- Set of functions that a provider offers
- Provider peer publishes service advertisement
- Pipes used to communicate with service
- Types of services:
  - Peer Services
  - Peer Group Services (discovery, membership, etc.)

# Advertisements

- All JXTA resources represented by advertisements
- Language-neutral XML documents
- Peers cache, publish, and exchange advertisements
- Each advertisement published with a lifetime (time-to-live)
  - Enables deletion of obsolete resources without requiring centralized control

# Resolvers

- In JXTA technology, all “binding” operations are simple discovery of advertisement(s)
- Example resolution operations
  - DNS (search for Peer or Peer Group advertisement)
  - Directory Service (search for a Peer adv.)
  - Socket Binding (search for a Pipe adv.)

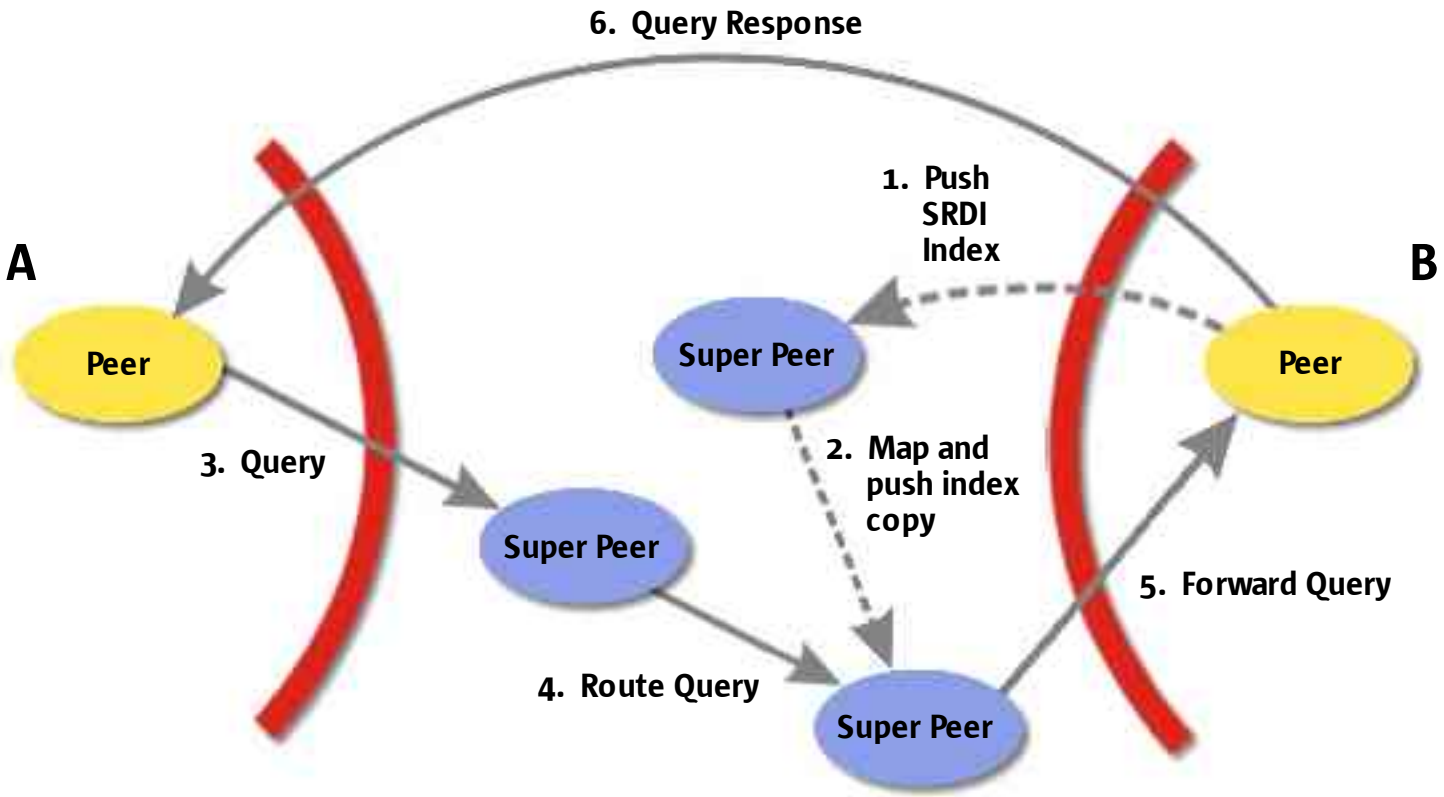
# Advertisement Discovery

- Local neighbor discovery
  - TCP/IP multicast
- Rendezvous peers
  - Discovery requests forwarded between rendezvous peers
  - Any peer may be a rendezvous peer
  - Cache a large number of advertisements
  - Each peer group has a set of rendezvous peers
- Out-of-band discovery

# Discovery Service

- Asynchronous mechanism for discovering advertisements (peers, peer groups, pipes, services)
- Can retrieve advertisements in local cache
- Can send Discovery Query Message
  - To a specific peer
  - Propagated to the JXTA network

# Request Propagation via Rendezvous Super Peers



# Security

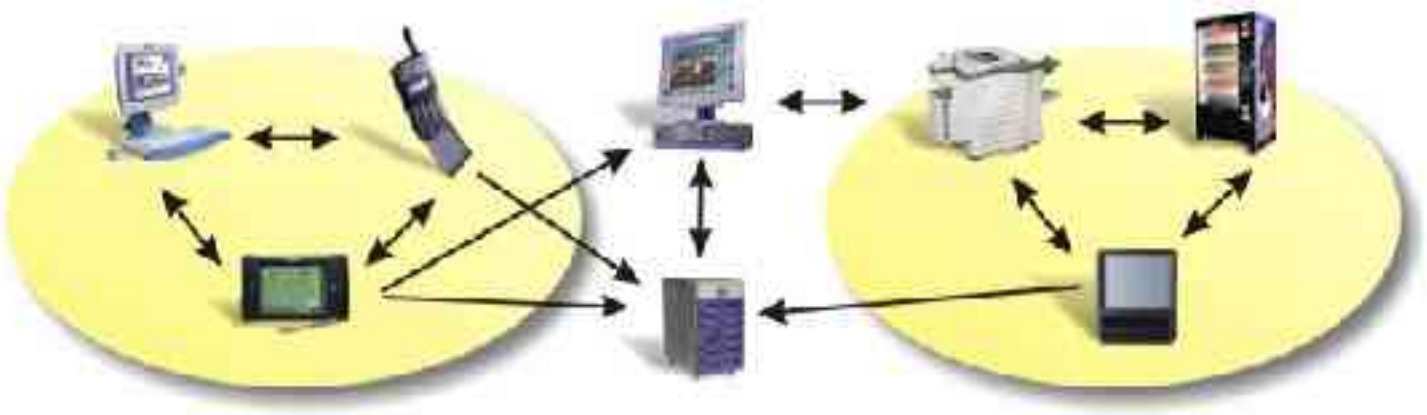
# Security Requirements

- Confidentiality
- Authentication
- Authorization
- Data integrity
- Refutability

# Intrinsic Security in P2P Networks

- Decentralization
- Privacy
- Locality
- “Web of Trust”

Physical Network

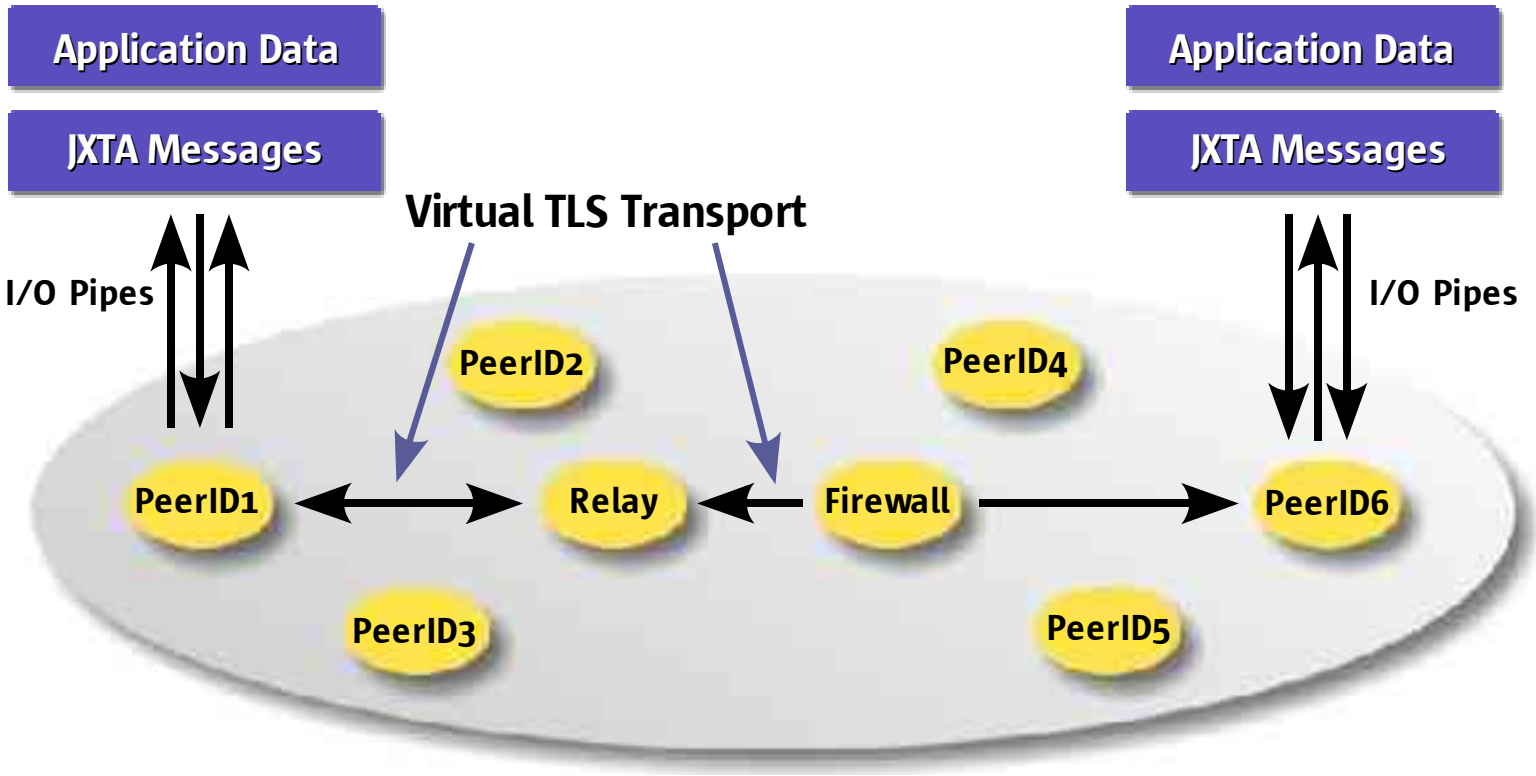


# JXTA Security Technologies

- Transport Layer Security (TLS)
- End-to-end transport independence of JXTA protocols
- Digital certificates and certificate authorities
- Encryption

**By adopting a security model that relies on existing, trusted technologies, JXTA can provide strong security quickly, adopt new technologies, and retain flexibility.**

# Transport Layer Security (TLS)



# Security in JXTA

- TLS Endpoint Transport
- Simple cryptography library
- Peer security
  - Every peer has its own root certificate
  - Public key certificate part of peer advertisements
  - Credential certificate embedded in every JXTA protocol message
- Authentication framework
- Password-based login scheme

# Security Resources

- JXTA Security project
  - http://security.jxta.org
  - discuss@security.jxta.org
- White papers
  - http://www.jxta.org/white_papers.html
- TLS
  - http://www.ietf.org.rfc.rfc2246.txt
  - http://www.claymoresystems.com (Pure TLS)
- Cryptography
  - http://www.bouncycastle.org/

# Interoperability Examples

- JXTA SOAP
  - Designed to allow SOAP communication over the JXTA P2P network
  - Leverages JXTA virtual network for dynamic discovery
  - Community Project at: http://soap.jxta.org/
- JMS-for-JXTA
  - Designed to allow JMS over JXTA
  - Designed to allow JXTA over JMS
  - Community Project at: http://jms-for-jxta.jxta.org/

# Interoperability Example

- Java Web Services
  - JXTA can be integrated at many levels within the Java Technologies for Web Services:
    - Java API for XML-Based RPC (JAX-RPC)
    - Java API for XML Messaging (JAXM)
    - Java API for XML Registries (JAXR)
    - Java API for XML Processing (JAXP)
    - Java Architecture for XML Binding (JAXB)
    - SOAP with Attachments API for Java (SAAJ)
- Please remember JXTA is hardware/OS platform, programming language and network agnostic!

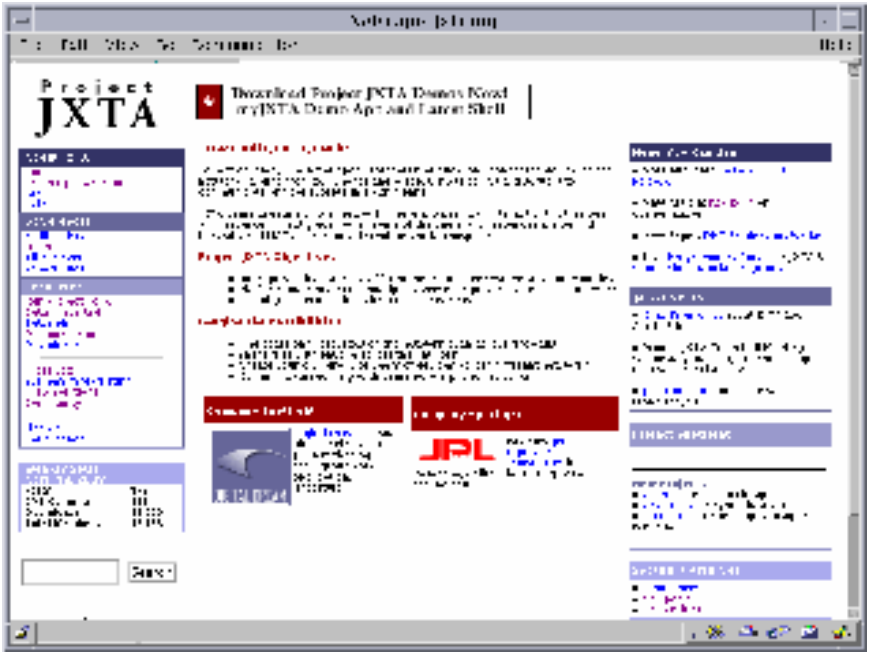
# JXTA Technology Status

- JXTA technology specification, code, demos, docs, and tutorials on-line
- Virtual network beginning to build
- Active community - contributing and integrating technology

# JXTA Community Momentum

## www.jxta.org (4/2001 – 4/2004)

- 2,500,000+ downloads
- 80+ Projects
- 17,500+ members
- Active discussion groups
- Community actively contributing and integrating technology




*Please join our efforts!*

# JXTA Implementation Platforms

- J2SE™ Implementation
  - Full implementation of JXTA protocols
  - Standard and Super Peer functionality
  - APIs and functionality frozen
- JXTA-C
  - Standard Peer functionality only
  - Runs on Linux, Solaris™ OE, and Windows
- JXTA Technology for J2ME™
  - Micro Peer functionality only
  - MIDP-1.0 compliant
  - iappli compliant

# Community Projects

- Python
- Perl
- Objective-C
- Ruby
- Smalltalk
- Services
- Applications
- And many others...

# Future Directions

- Enhanced Performance, Scalability, and Security
  - SAML
  - expanded use of credentials
- New services and opportunities
  - E.g. identity, integration with Web services, content management, digital rights, presence
- Specification standardization through public organization
  - http://spec.jxta.org/

# Q & A

**Richard Manning**
**(o) 678-448-4019**
**richard.manning@sun.com**

**www.jxta.org**
**www.java.net**
**www.sun.com**

Sun microsystems
We make the net work.