Introduction to P2P
and the
JXTA framework

Slides from Sébastien Baehni
modified with permission by
Vincent Matossian

Visit: http://lpdwww.epfl.ch/sbaehni/work/presentations/presentations.html
For original slides
Plan of the presentation

• Introduction to P2P (7 slides)
• JXTA (39 (quick) slides)
• Conclusion
Introduction to P2P

• What is a Peer?
  – « one that is of equal standing with another » [MW01]

• What is P2P?
  – « P2P is a network architecture in which each computer has equivalent capabilities and responsibilities » [Mil01]
  – P2P network content provided by the peers
  – Peers are dynamic and autonomous
  – No central authority, point of failure

• What isn’t P2P?
  – Basically: client-server architecture
Introduction to P2P
Introduction to P2P

• Old concept (~ 30 years)
  – Arpanet (leading to Internet)
  – Usenet (UUCP leading to NNTP)
  – DNS

• Interesting now
  – Low cost of PCs
  – More and more people using a PC
  – Expansion of the Internet (~ 513 million, 08.01 [NUA02])

• 3 main architectures
  – Centralized (*Napster* (originally))
  – Partially decentralized (*Kazaa*)
  – Totally decentralized (*Gnutella* (originally))
Introduction to P2P

Centralized

Partially decentralized

Totally decentralized
Introduction to P2P

• Problems raised with P2P
  – Administration of peers
  – Privacy/Anonymity and Security
  – Trustworthiness
  – Propagation of undesirable content
  – Intellectual property
  – Availability of content
  – Firewall, NAT bypassing
  – Group membership communication
  – Performance, scalability
Introduction to P2P

• Today’s applications
  – Instant Messaging (ICQ, AIM, Jabber, …)
  – File Sharing (Napster, Morpheus, Gnutella, Freenet, …)
  – Distributed Search Engines (OpenCola, Copernic, …)
  – Group Collaboration (Groove, NetMeeting, …)
  – Distributed Computing (Seti@Home, Parabon, …)

• Tomorrow’s applications
  – Distributed Storage
  – Licensed Media Distribution
  – Commerce (Intel)
  – Intelligent Agent
  – Online Gaming
  – Writeable Web
  – Wireless network
  – War applications
Introduction to P2P

• Main problem
  – Lot of different applications, own programming model

• Solution
  – Having a common P2P « specification », framework
JXTA

• Abbreviation of juxtaposition:
  – « putting things next to each other »
• Concept of Bill Joy (Chief Scientist, Sun)
• Fairly new concept
  – First talk in 15.02.2001
  – Web-site (www.jxta.org) opened the 24.04.2001
• Specification defining the basic concepts for creating P2P applications [sJxta02]
• An open-source project developed around the specification (Project JXTA)
JXTA

• 3 main goals
  – Interoperability
    • Across different P2P systems and communities
  – Platform Independence
    • Programming languages, System platforms, Network Platforms
  – Ubiquity
    • Every electronic device with a digital heartbeat
JXTA

refridgerator, TinyOS, Perl

Sensor, TinyOS, Obj-c

MainFrame, UNIX, C++

Mobile, Symbian, C

Mac, MacOsX, Python

PC, Win2K, Java
JXTA

• 3 main layers
  – Core layer (platform)
    • Minimal, essential primitives common to the P2P network
    • Basic communication, discovery mechanisms
    • Basic security, membership mechanisms
    • Basic monitoring mechanisms
  – Service layer
    • Set of services on JXTA
    • Not necessary but desirable
    • cms, jxtaspaces, jxta-rmi…
  – Application layer
    • May use the service layer
    • Boundary not rigid with the service layer
    • myJxta, jxAuction, vop2p…
JXTA

JXTA MODEL

JXTA Applications

Community JXTA Applications

Sun JXTA Applications

JXTA Services

Community JXTA Services

Sun JXTA Services
- Indexing
- Searching
- Filesharing

JXTA Core

Peer Groups
Peer Pipes
Peer Monitoring

Security

Any Peer on the Expanded Web

[Images of different devices]
JXTA

• Core layer is based on:
  – Concepts
  – Protocols

• Concepts
  – Peers
  – Peer groups
  – Network Services
  – Pipes
  – Advertisements
  – Messages
  – Identifiers
  – Credentials
  – Contents
JXTA

• Peer
  – Any networked device using JXTA
  – Has a unique ID, allowing to be addressed independently of its physical location
  – Interact with each other spontaneously
  – May provide network services
  – May cache information
  – No assumption about availability
  – May have multiple network interfaces (for sending/receiving data)
  – Interact with a small number of peers
JXTA

• Different types of peers
  – Relay peers
  – Rendez-vous peers

• Relay peers
  – Used to route messages
    • Support multi-hop message transfer
    • Maintain dynamic route information
    • Cache messages for temporarily unavailable peers
  – Used for traversing firewall and NAT

• Rendez-vous peers
  – Used to cache advertisements
  – Used to propagate messages
  – Maintain a list of its connected rdv and clients
  – Act as entry points for the JXTA network
JXTA

Peer A
urn:jxta:uuid-0032….03

- cache
- jxtaspaces
- jxta-rmi

UDP
HTTP
1802.11

Peer B
Peer C
Peer D
JXTA

• Peer group
  – Collection of peers sharing a common interest
  – Create a secure, scoped and monitored environment
  – Provide a set of core services
    • Discovery service
    • Membership service
    • Pipe service
    • Resolver service
    • Monitoring service
  – A peer may join several peer groups
  – All peers belong to the « world peer group »
  – Java implementation: a peer is in the « NetPeerGroup »
JXTA

NetPeerGroup

World peer group
“Services provide functionality that peers can engage to perform ‘useful work’ on a remote peer. This work may include transferring a file, providing status information, performing a calculation, or basically anything that you might want a peer in a P2P network to be capable of doing. Services are the motivation for gathering devices into a P2P network.”

Brendon Wilson
JXTA

• Network Services
  – Peer services
    • Peer failed, service failed
  – Peer groups services
    • Service failed when all peers supporting the service failed
  – Similar as a plug-in (finding, installation)

• Pipes
  – Virtual communication channels
  – Asynchronous, uni-directionnal and unreliable (basic pipe)
  – Binding done dynamically with the Pipe Binding Protocol
  – Uni-directionnal ➔ input and output pipes
  – Different kind of pipes
    • Uni-directionnal (many-to-one)
    • Propagate pipe (many-to-many, in the scope of a peer group)
    • Enhanced pipe (secure, bidirectionnal)
JXTA

• Advertisements
  – All networked resources are advertisements
    • Peer advertisement
    • Peer group advertisement
    • Pipe advertisement
    • Module (service) advertisement
    • Content advertisement
    • …
  – Metadata structures for describing resources
  – Represented in XML
  – Have a local and remote age
JXTA

• Peer Advertisement

<?xml version="1.0"?>
<!DOCTYPE jxta:PA>
<jxta:PA xmlns:jxta="http://jxta.org">
  <PID> urn:jxta:uuid-59616261646162...C88E081668F0E4BDA903 </PID>
  <GID> urn:jxta:uuid-FF1D3F7597584238F2F89F9E87AD51F602 </GID>
  <Name> Seb </Name>
  <Svc>
    <MCID> urn:jxta:uuid-DEADBEEFDEAFB...0000000805 </MCID>
    <Parm>
      <Addr> tcp://128.178.73.43:9701/ </Addr>
      <Addr> jxtatls://uuid-59......03/TlsTransport/jxta-
        WorldGroup </Addr>
      <Addr> jxta://uuid-59......903/ </Addr>
      <Addr> http://JxtaHttpClientuuid-59......903/ </Addr>
    </Parm>
  </Svc>
</jxta:PA>
JXTA

• Messages
  – Basic unit of data exchanged between peers
  – Two representations
    • XML/Binary
      • Depends on the transport protocol for convenience purpose
  – Sets of name/value pair (element)
  – Element
    • Added while passing down the protocol stack
    • Removed while passing back up the stack
  – XML because
    • Language agnostic
    • Self-describing
    • Ensure correct syntax
JXTA

• Messages (XML format)

```xml
<?xml version="1.0"?>
<!DOCTYPE Message>
<Message version="0">
  <Element name="jxta:SourceAddress" mime_type="text/plain"> tcp://123.456.205.212 </Element>
  <Element name="stuff" encoding="base64" mime_type="application/octet-stream">
    AAECAwQFBgcICQoLDA0ODxAREhMUFRYXGBkaGxwdHh8gISIjJCUmJygpKissLS4vMD
    EyMzQ1Njc4OTo7PD0+P0BBQkNERUZHSElKS0xNTk9QUVJTVFVWW1hZWltcXV5fY
    GFi Y2RlZmdoaWprbG1ub3BxcnN0dXZ3eHl6e3x9fn+AgYKDhIWGh4iJiouMjY6PkJGSk5SVm
    ZqbnJ2en6ChoqOkpaanqKmq q6ytrq+wsbKztLW2t7i5uru8vb6/wMHCw8TFxsc=
  </Element>
</Message>
```
JXTA

• Identifiers
  – Unambiguously identify an entity and serve to locate it
  – No central name server
  – Presented as URN
  – Properties:
    • Unambiguous
    • Unique
    • Canonical
    • Opacity
    • 64 bytes array
  – Typical id:
    • urn:jxta:uuid-59616261646162...C8168F0E4BDA903
JXTA

• Credential
  – Used to identify a sender
  – Provides:
    • Confidentiality
    • Authorization
    • Data integrity
    • Refutability

• Content (codat)
  – Any data (even an executable process)
  – Shared among peers and peer groups
  – Uniquely identify
  – Known by a Content advertisement
JXTA Protocols

- Protocols
  - Peer Resolver Protocol (PRP)
  - Peer Discovery Protocol (PDP)
  - Peer Information Protocol (PIP)
  - Pipe Binding Protocol (PBP)
  - Peer Endpoint Protocol (PEP)
  - Rendezvous Protocol (RVP)

- Only the needed one have to be implemented
JXTA

• Peer Resolver Protocol (PRP)
  – Used to register handlers of specific messages
  – Bound a query/response to the specific handler
  – Each query has a unique ID
  – Does not assume any kind of transport protocol
  – Used by the PDP and PIP
JXTA

Peer

H1

H2

Resolver service

query

response

query
JXTA

• Peer Discovery Protocol (PDP)
  – Allows to advertise the resources of a peer
  – Allows to discover any published resource
  – Done in a group context
  – IP multicast propagation into a LAN (Java implementation)
  – Uses rendezvous/relay into a WAN (RVP)
  – Uses TTL to prevent flooding
  – Check messages id to avoid redundant discovery
JXTA

• Peer Information Protocol (PIP)
  – Used to get information about a peer
• Pipe Binding Protocol (PBP)
  – Allows to establish a virtual communication channel between two or several peers
  – Binds the ends (input/output(s)) of the connection
  – On top of the PEP
  – Uses the PRP
  – Once the pipe is bound, the PipeService is responsible for keeping it bound
  – Query: contains the PipeAdvertisement to found
  – Reply: contains the associated PeerAdvertisement
JXTA

pipeQuery

pipeResponse(peerAdv)

pipeAdv

pipeQuery

sendMsg

pipeAdv

pipeAdv

Input pipe

Output pipe
JXTA

• Peer Endpoint Protocol (PEP)
  – Used to dynamically find a route to send a msg to another peer
    • Uses the PRP (queries send to other routers)
    • Caches route information locally and uses remote peers (gateway)
  – Uses the relay service for peers that are not reachable (firewall)
    • Leasing mechanism
  – Uses the Endpoint service to send messages
    • The PeerAdvertisement contains a list of Transport Protocols
    • Endpoint service used to delegates the sending part to the appropriate protocol
JXTA

- Routing
JXTA

- Relay

Firewall, http out only

Peer relay

Lease manager
JXTA

• RendezVous Protocol (RVP)
  – Used to propagate a message into a peergroup
  – Propagates the queries to all its connected clients and rendezvous
  – Uses a leasing mechanism
  – Used by the PDP
JXTA

connection

Rdv peer
JXTA

• Summary of the core layer

- Pipe Service
- Propagation Service
- Router and relay services
- TCP
- UDP
- HTTP
- XML messages
JXTA

• Service layer
  – Services on the core layer
  – Expanding list of services:
    • jxta-rmi
    • jxta-wire (now into the core layer)
    • ...
  – Created using the Module(Spec/Class/Impl)Advertisement

43
JXTA

• Application layer
  – Regroup the registered applications developed using JXTA
  – 20 well known applications:
    • shell
    • myJxta
    • vop2p
    • …
JXTA

• Shell
  – Command shell allowing to interact directly with the JXTA platform
    • Create advertisements
    • Talk to other peers
    • …
  – Command line interface
  – Allows to run scripts
  – Allows to add dynamically new commands
  – Check out more at http://shell.jxta.org
The JXTA Shell provides an interactive environment to the JXTA platform. The Shell provides basic commands to discover peers and peer groups, to join and resign from peer groups, to create pipes between peers, and to send pipe messages. The Shell provides environment variables that permit binding symbolic names to Jxta platform objects. Environment variables allow Shell commands to exchange data between themselves. The shell command 'env' displays all defined environment variables in the current Shell session.

The Shell creates a Jxta InputPipe (stdin) for reading input from the keyboard, and a Jxta OutputPipe (stdout) to display information on the Shell console. All commands executed by the Shell have their initial 'stdin' and 'stdout' set up to the Shell's stdin and stdout pipes. The Shell also creates the environment variable 'stdgroup' that contains the current JXTA PeerGroup in which the Shell and commands are executed.

A new Shell can be forked within a Shell. The 'Shell -s' command starts a new Shell with a new Shell window. The Shell can also read a command script file via the 'Shell -f myfile'.

A 'man' command is available to list the commands available. Type 'man <command>' to get help about a particular command. To exit the Shell, use the 'exit' command.

JXTA>
JXTA

• myJxta
  – Demonstration of possibilities of JXTA
    • Instant Messaging
    • File sharing
    • Secure chat
  – Check out more at http://myjxta.jxta.org
myJXTA
JXTA

• JXTA web sites characteristics
  – Members: 15000+
    • Apache-like license to be able to contribute
  – Projects: 50+
  – Downloads:
    • JXTA: 165000
    • Doc: 181000
  – Home page: 2.5 million unique visits
  – Sun team: 25 and 17 are engineers
  – Check out more at www.jxta.org
References


[BBD02] Nitin Borwankar, Daniel Brookshier et al., *JXTA: Java P2P Programming,* Sams publishing, March 2002


[jxta02] JXTA web site, [http://www.jxta.org](http://www.jxta.org)

[Li02] Sing Li, *JXTA Peer-to-Peer Computing with Java,* Wrox, January 2002


References


