



HAL
open science

JXTA : une infrastructure générique pour le développement d'applications P2P

Guillaume Doyen, Olivier Festor, Emmanuel Nataf

► **To cite this version:**

Guillaume Doyen, Olivier Festor, Emmanuel Nataf. JXTA : une infrastructure générique pour le développement d'applications P2P. Groupe d'Exploitation des Réseaux Ethernet TCP/IP - GERET, 2004, Nancy, France, 24 p. inria-00099849

HAL Id: inria-00099849

<https://inria.hal.science/inria-00099849>

Submitted on 26 Sep 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

JXTA

*Une infrastructure générique pour développement
d'applications P2P*

Guillaume DOYEN, dirigé par Olivier FESTOR
Equipe MADYNES
LORIA – INRIA Lorraine
Email : Guillaume.Doyen@loria.fr

Plan de la présentation

- Introduction
 - Le modèle P2P
 - Objectifs
 - Historique
- Concepts
- Architecture & protocoles
- Implémentation
- Le shell Jxta
- Jxta 2.0
- Conclusions

Introduction

- Le modèle P2P

- *Peer*

- Rôle de client et serveur
 - Nature hétérogène
 - Ressources inutilisées

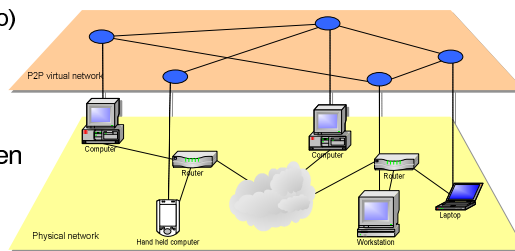
- Modèle décentralisé

- Différents niveaux
 - Pur (Freenet, Chord, PAST, Gnutella, ...)
 - Hybride (Jxta, Kazaa, ...)
 - Centralisé (Napster, Porivo)
 - Passage à l'échelle
 - Equilibre du trafic
 - Tolérance aux fautes
 - Partage des coûts de mise en œuvre et de maintenance
 - Auto organisation

- Connectivité Ad Hoc

- Réseau de type *overlay*

- Routage et nommage de niveau applicatif
 - Indépendance de la technologie sous-jacente



Objectifs

- Constats

- Explosion des applications P2P
 - Similarité des services (recherche, découverte, ...)
 - Pas de *toolkit* commun

- Objectif

- Définition de protocoles génériques
 - Proposition d'une implémentation
 - Déploiement sur de nombreux systèmes
 - Interopérabilité
 - Protocoles
 - Indépendance / plate-formes
 - Langages : C, Java, Python
 - Réseaux : Bluetooth, UMTS, ...
 - Terminaux : du téléphone portable au calculateur parallèle
 - Simplicité des concepts

Historique

- 2000
 - Multiplication des applications P2P
 - Gnutella, Napster, ...
- 2001
 - P2P devient un paradigme pour les applications distribuées
 - Sun lance une initiative autour du P2P : JXTA
 - Création de jxta.org
 - Jxta v1.0
- 2004
 - Jxta v2.0
- Aujourd'hui
 - 90 projets
 - 1 million de téléchargements
 - Domaines d'applications
 - Partage de contenu
 - Collaboration
 - Communication

Plan de la présentation

- Introduction
- Concepts
 - *Peer*
 - Communauté
 - Communication
 - Service
 - Annonce
 - Nommage
- Architecture & protocoles
- Implémentation
- Le shell Jxta
- Jxta 2.0
- Conclusions

Concepts : *peer* 1/3

- *Peer*
 - Nœud du réseau P2P
 - « *An entity capable of performing some useful work and communicating the results of that work to another entity over a network either directly or indirectly.* » [1]
 - Nature hétérogène
 - Du PDA à l'application distribuée
 - Plusieurs *peers* sur une machine
 - Différents types
 - *Minimal Peer*
 - *Simple Peer*
 - *Rendezvous Peer*
 - *Router Peer*

Concepts : *peer* 2/3

- *Minimal peer*
 - Exécution d'application P2P
 - Émission & réception de messages
 - Pas de réponse aux sollicitations d'annonce
 - Terminal de faible capacité
- *Simple peer*
 - Répond aux sollicitations de découverte
 - Pas de routage
- *Router peer*
 - Peer simple
 - Routage de messages entre *peers*
 - Firewall, NAT
 - Couche transport incompatible
 - Adressage Dynamique (~DNS)

Concepts : *peer* 3/3

- *Rendezvous Peer*
 - *Peer* simple
 - « Lieu de rencontre et de découverte entre *peers* »
 - Connaissance de *peers*
 - Maintient une liste de *peers*
 - Cache des annonces
 - Optimisation de recherche
 - *Forward* de requêtes de découverte
 - Vers autres *RDVpeers*
 - Vers *peers* simples
 - Point de ralliement par défaut
 - *Peers* émettent des requêtes de recherche vers des *RDVpeers*
 - *Peers* se connectent à un/des *RDVpeers*

Concepts : communauté

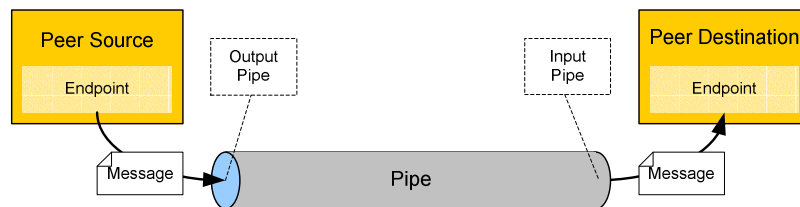
- Problématique
 - Association Application / Protocole
 - Incompatibilité
 - Exemples
 - Gnutella
 - ICQ
- *PeerGroup*
 - Division du réseau
 - Regroupement par intérêts communs
 - Restriction service / groupe
 - Environnement sécurisé
 - Authentification
 - Monitoring d'autre *peers*
 - Groupe par défaut : **NetPeerGroup**

```
JXTA>whoami -g
<PeerGroup>
  NetPeerGroup
</PeerGroup>
<Description>
  NetPeerGroup
  by default
</Description>
<PeerGroupId>
  urn:jxta:
  jxta-NetGroup
</PeerGroupId>
```

Exemple de *PeerGroup*

Concepts : communication

- *Network Transport*
 - Ensemble de mécanismes qui assure l'échange de données entre *peers*
 - Séparation des données en paquets
 - Ajout d'entêtes
 - Eventuel acquittement
 - Deux niveaux
 - Bas : TCP, UDP, ...
 - Haut : HTTP, SMTP, ...
- *Constitution*
 - *Endpoints*
 - Interface réseau virtuel
 - Source, destination
 - *Pipe*
 - Unidirectionnel
 - Asynchrone
 - Connecte 2 *endpoints* ou plus
 - *Message*
 - Conteneur de données
 - Transmis dans un *pipe*



Concepts : service

- Valeur ajoutée d'un réseau P2P
 - Fonction qu'un *peer* peut engager sur un *peer* distant
 - Transfert de fichiers, virement bancaire, inversion de matrice, ...
- 2 types de services
 - Service de *peer*
 - 1 seul *peer* héberge le service
 - Disponible ssi le *peer* est présent
 - Service de groupe
 - Service distribué dans un groupe
 - Disponible ssi au moins 1 *peer* du groupe est présent
- Composition de service
 - *Jxta Core Services*

Concepts : annonce

- *Advertisement*
 - Représentation structurée
 - XML
 - Tout est annonce !
 - *Peers*, groupes
 - Services
 - *Pipes*, endpoints
 - Ressources
- Méta données fondamentales
 - Cœur du processus de découverte
 - Liaison aux ressources

```
JXTA>monadv = mkadv -g FestorTestGroup
JXTA>cat monadv
<?xml version="1.0"?>
<!DOCTYPE jxta:PGA>
  <jxta:PGA xmlns:jxta="http://jxta.org">
    <GID>
      urn:jxta:uuid-CE1ABC502A
      E2404486BD8C1DB8116BDD02
    </GID>
    <MSID>
      urn:jxta:uuid-DEADBEEFDEA
      FBABAFEEDBABA000000010306
    </MSID>
    <Name>
      FestorTestGroup
    </Name>
    <Desc>
      created by mkadv
    </Desc>
  </jxta:PGA>
```

Exemple d'annonce de *PeerGroup*

Concepts : nommage

- Identification
 - Entités
 - *Peer*, groupe
 - Service
 - *Pipe*, endpoint
 - Ressources
 - Nommage indépendant
 - Système d'exploitation
 - Réseau
 - Adresse
 - Protocole de transport
 - Identification conviviale
 - Nom sous forme de chaîne de caractère
 - Identification unique sur le réseau
 - URN de 17 ou 30 octets

```
<Peer>
Olivier's peer
</Peer>
<PeerId>
urn:jxta:uuid-59616261646162614A78
746150325033C02C36A2
5F3E4F039585816D690B
9F0703
</PeerId>
```

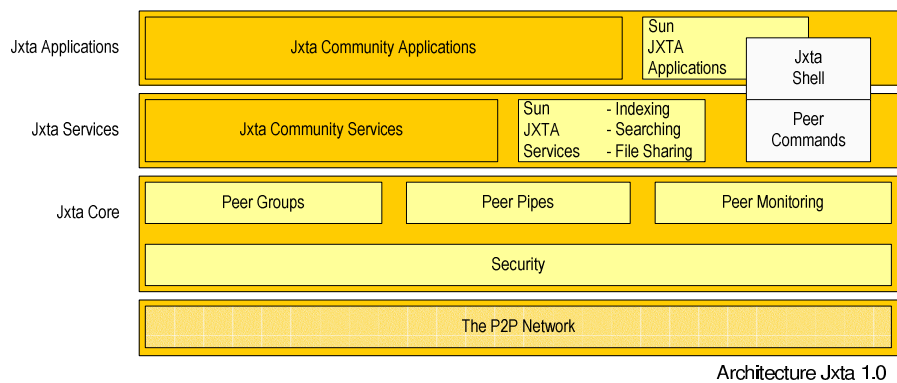
Exemple d'identificateur de *peer*

Plan de la présentation

- Introduction
- Concepts
- Architecture & protocoles
 - Les différentes couches
 - Découverte
 - Routage
 - Communication
- Implémentation
- Le shell Jxta
- Jxta 2.0
- Conclusions

L'architecture Jxta 1.0 1/4

- Architecture générale



L'architecture Jxta 1.0 2/4

- La couche *Core*

- Éléments essentiels au concept P2P

- *Peers*
 - Groupes
 - Transport
 - *Pipes*
 - *Endpoints*
 - Messages
 - Annonces
 - Nommage des entités

- Protocoles

- Découverte
 - Communication
 - Monitoring

- Sécurité

- Authentification

Jxta Applications

Jxta Services

Jxta Core

L'architecture Jxta 1.0 3/4

- La couche *Service*

- Valeur ajoutée
 - Non nécessaire
 - Indépendance
 - Application
 - Couche ouverte
 - Développement de nouveaux services

- Exemples

- Découverte de ressources
 - Partage de documents
 - Authentification

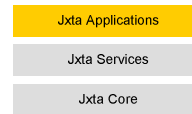
Jxta Applications

Jxta Services

Jxta Core

L'architecture Jxta 1.0 4/4

- La couche application
 - Repose sur couche services
 - Application vs service
 - Agrégation de services
 - Interface utilisateur

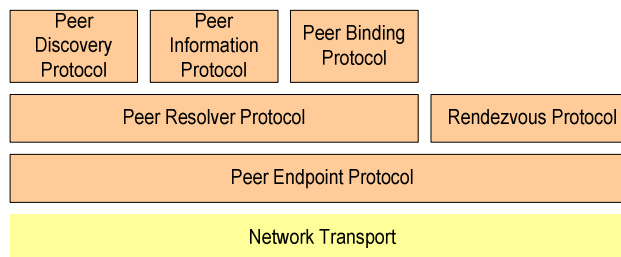


- Exemples

- [angelopeerrendezvous](#) : Communication intra-entreprise
- [gameplatform](#) : Plate-forme de jeux multijoueurs
- [gnovella](#) : Stockage de documents en entreprise
- [jxcube](#) : Collaboration
- [jxta-httpd](#) : Publication Web
- [myjxta](#) : Outil de démonstration Jxta
- [p2pconference](#) : Télé-conférences (texte)
- [radiojxta](#) : Diffusion audio
- [rosettachat](#) : Messagerie instantanée
- [shell](#) : Ligne de commandes Jxta
- [vop2p](#) : Voix sur P2P

La pile de protocoles Jxta

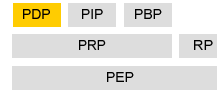
- Principe
 - Généricité
 - applications
 - Indépendance
 - Machine
 - Système d'exploitation
 - Transport
- Protocoles
 - Aspects fondamentaux du P2P
 - Messages XML
 - Semi-indépendance



La pile de protocoles Jxta

Les protocoles : découverte 1/2

- Problématique
 - Présence intermittente
 - NAT / *Firewall*
- Solution Jxta
 - Association ressource / annonce
 - Différentes méthodes
 - Cache local
 - Pas de trafic réseau
 - Problème d'actualité des données
 - Limitation de la durée de vie
 - A utiliser en conjonction avec la découverte
- Découverte directe
 - Restriction : LAN
 - *Broadcast* ou *multicast*
- Découverte indirecte
 - Echelle de l'Internet
 - *Peer* de rendez-vous
 - Propagation
 - Cache
 - Problèmes de boucles
 - Identification des messages
 - Messages zombies
 - TTL



Les protocoles : découverte 2/2

- *The Peer Discovery Protocol (PDP)*
 - Messages
 - Requête
 - Réponse
- *The Discovery service*
 - *Core service*
 - Une instance par groupe
 - Niveau d'abstraction
 - Messages
 - Fonctions
 - Recouvrer des annonces distantes
 - Recouvrer des annonces locales
 - Publier des annonces localement
 - Publier des annonces à distance
 - Effacer le cache local

```

<?xml version="1.0" encoding="UTF-8"?>
<jxta:DiscoveryQuery>
  <Type> </Type>
  <Threshold> </Threshold>
  <PeerAdv> </PeerAdv>
  <Attr> </Attr>
  <Value> </Value>
</jxta:DiscoveryQuery>

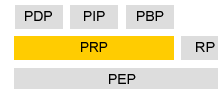
<?xml version="1.0" encoding="UTF-8"?>
<jxta:DiscoveryResponse>
  <Type> </Type>
  <Count> </Count>
  <PeerAdv> </PeerAdv>
  <Attr> </Attr>
  <Value> </Value>
  <Response>
    Expiration=" ExpTime "
  </Response>
</jxta:DiscoveryResponse>
    
```

Format des messages Query et Response

Les protocoles : résolution 1/2

- *Peer Resolver Protocol (PRP)*

- Protocole générique
 - *Handler*
- Diffusion
 - *Peers* simples connus
 - *Peers* de rendez-vous
- Messages
 - Requête
 - Réponse



```
<?xml version="1.0"
encoding="UTF-8"?>
<jxta:ResolverQuery>
  <HandlerName> </HandlerName>
  <Credential> </Credential>
  <QueryId> </QueryId>
  <SrcPeerId> </SrcPeerId>
  <Query> </Query>
</jxta:ResolverQuery>

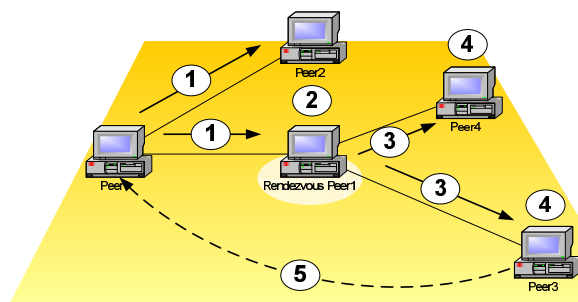
<?xml version="1.0" >
encoding="UTF-8"?>
<jxta:ResolverResponse>
  <HandlerName> </HandlerName>
  <Credential> </Credential>
  <QueryId> </QueryId>
  <Response> </Response>
</jxta:ResolverResponse>
```

Format des messages Query et Response

Les protocoles : résolution 2/2

- Exemple

- P1 -> P3
- Déroulement

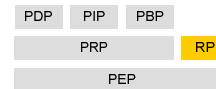


PRP : exemple d'échanges.

- 1 : P1 : Envoi à tous les *peers* connus
- 2 : RdvP1 : Vérifie si un *handler* correspond (si oui, répond)
- 3 : RdvP1 : Diffuse la requête
- 4 : P3 & P4 : Vérifient si un *handler* correspond
- 5 : P3 : Possède un *handler* correspondant. Communique la requête au service correspondant et éventuellement répond.

Les protocoles : rendez-vous

- *Rendezvous Protocol (RVP)*
 - Connexion à un *peer* de rendez-vous
 - Principe de bail
 - Temps limité
 - Messages
 - Requête de connexion
 - Acceptation de connexion
 - Annulation de connexion
 - Pas de spécification XML
 - Prévention des boucles
 - Information de chemin
 - TTL
- *Rendezvous Service*
 - Abstraction du protocole
 - Renouvellement automatique des connexions
 - Monitoring des connexions



Les protocoles : monitoring

- *Peer Information Protocol (PIP)*
 - Protocole optionnel
 - Monitoring de *peer* distant
 - Trafic
 - Date d'arrivée
 - Messages
 - Requête
 - Réponse
 - Propagation
 - Diffusion
 - Limitation au groupe
 - Illimitée !
 - Pas complètement déployé (sur Jxta 1.0)
- *Peer info service*
 - Abstraction du protocole
 - Utilisation d'un cache local



```
<?xml version="1.0"
encoding="UTF-8"?>
<jxta:PeerInfoQueryMessage>
  <sourcePid>      </sourcePid>
  <targetPid>     </targetPid>
  <request>       </request>
</jxta:PeerInfoQueryMessage>

<?xml version=" 1.0 "
encoding="UTF-8"?>
<jxta:PeerInfoResponse>
  <sourcePid>     </sourcePid>
  <targetPid>     </targetPid>
  <uptime>        </uptime>
  <timestamp>     </timestamp>
  <response>      </response>
  <traffic>       </traffic>
</jxta:PeerInfoResponse>
```

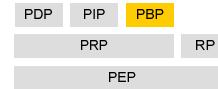
Format des messages Query et Response

Protocoles : liaison 1/2

- *Pipe*
 - Abstraction de connexion
 - Unidirectionnel
 - Unicast, propagé, sécurisé
 - *Pipe advertisement*

```
<?xml version=«1.0»
encoding=«UTF-8»?>
<jxta:PipeAdvertisement>
  <Id>                </Id>
  <Type>              </Type>
  <Name>              </Name>
</jxta:PipeAdvertisement>
```

- *Pipe Binding Protocol (PBP)*
 - Association *Pipe / Endpoint*
 - Messages
 - Requête
 - Réponse



```
<?xml version=«1.0»
encoding=«UTF-8»?>
<jxta:PipeResolver>
  <MsgType>Query </MsgType>
  <PipeId>        </PipeId>
  <Type>          </Type>
  <Cached>        </Cached>
  <Peer>          </Peer>
</jxta:PipeResolver>
```

```
<?xml version=« 1.0 »
encoding=«UTF-8»?>
<jxta:PipeResolver>
  <MsgType>Answer </MsgType>
  <PipeId>        </PipeId>
  <Type>          </Type>
  <Peer>          </Peer>
  <Found>         </Found>
  <PeerAdv>       </PeerAdv>
</jxta:PipeResolver>
```

Format des messages Query et Réponse

Protocoles : liaison 2/2

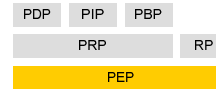
- Exemple
 - *Pipe* P2 -> P1
 - Annonce de *pipe* connue de P1 et P2
 - Déroulement
 - 1 : P1 : Création d'un *input pipe*
 - 2 : P2 : Envoi d'une requête *Pipe Binding*
 - 3 : P1 : Reçoit la requête, consulte son cache et répond
 - 4 : P2 : Reçoit la réponse, extrait le *endpoint* et crée le *output pipe*



- *Pipe service*
 - Abstraction du protocole
 - *Pipes* sécurisés
 - *Transport Security Layer*
 - *Pipes* bidirectionnels

Protocoles : endpoints 1/2

- Endpoints
 - « *Entities responsible for conducting the actual exchange of information over a network* »
 - Interface réseau virtuel
 - Avec le niveau transport
 - Indépendance de la technologie sous-jacente
- *Endpoint service*
 - Point d'accès à l'ensemble des *endpoints* disponibles sur un *peer*
 - Indépendance des groupes
 - Implémentations Jxta
 - TCP
 - HTTP
 - Servlet HTTP
 - TLS (*Transport Layer Security*)
 - BEEP
 - Abstraction du routage



Protocoles : endpoints 2/2

- *Endpoint service (suite)*
 - Adresses
 - `<Protocol>://<Network address>/<Service name>/<Service parameters>`
 - Exemple : `tcp://10.6.18.38:80/PipeService/<PipeId>`
 - Format de messages
 - XML
 - Binaires
 - Propagation de messages
 - Réseaux locaux
 - TCP
 - Filtrage de messages
 - Traitement avant transmission au niveau supérieur

Protocoles : routage 1/2

- *Endpoint Routing Protocol (ERP)*
 - Protocole de routage Jxta
 - Découverte de routes
 - Messages
 - Requête
 - Réponse
 - Router message
- *Endpoint Router Transport Protocol*
 - Implémentation de ERP
 - Invocation automatique
 - Adresses
 - Jxta://<PeerId>
 - Pas de propagation

```

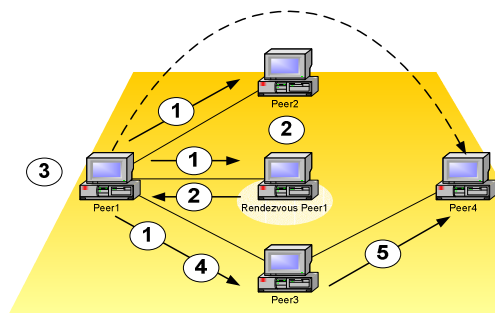
<?xml version="1.0" encoding="UTF-8"?>
<jxta:EndpointRouter>
  <Type>RouteQuery      </Type>
  <DestPeer>            </DestPeer>
  <RoutingPeerAdv>     </RoutingPeerAdv>
</jxta:EndpointRouter>

<?xml version="1.0" encoding="UTF-8"?>
<jxta:EndpointRouter>
  <Version>             </Version>
  <Type>RouteResponse  </Type>
  <DestPipeIdTag>      </DestPeerIdTag>
  <RoutingPeerIdTag>   </RoutingPeerIdTag>
  <NbOfHops>           </NbOfHops>
  <RoutingPeerAdvTag>  </RoutingPeerAdvTag>
  <GatewayForward>     </GatewayForward>
</jxta:EndpointRouter>

<jxta:EndpointRouter>
  <jxta:Src>            </jxta:Src>
  <jxta:Dest>          </jxta:Dest>
  <jxta:Last>          </jxta:Last>
  <jxta:NBOH>          </jxta:NBOH>
  <jxta:GatewayForward> </jxta:GatewayForward>
  <jxta:GatewayReverse> </jxta:GatewayReverse>
</jxta:EndpointRouter>
  
```

Format des messages Query et Response, et Router message

Protocoles : routage 2/2



ERP : Exemple d'échanges

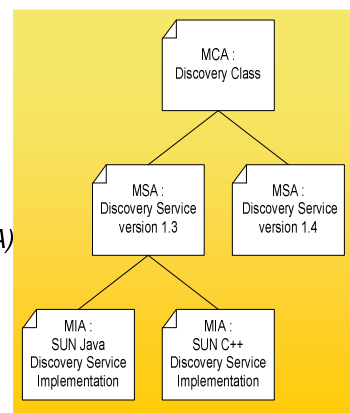
- Exemple
 - Envoi P1 -> P4
 - Déroulement
 - 1 : P1 : Envoi requête de route
 - 2 : RdvP1 : Connaît une route vers P4. Envoie sa réponse
 - 3 : P1 : Ajoute un *endpoint* intermédiaire à son message et l'envoie.
 - 4 : P3 : Détermine le prochain *peer* sur la route et envoie le message.
 - 5 : P4 : Est la destination finale du message.

Plan de la présentation

- Introduction
- Concepts
- Architecture & protocoles
- Implémentation
 - Modules
 - Amorce
- Le shell Jxta
- Jxta 2.0
- Conclusions

Implémentation : modules 1/2

- Module
 - « *some distributed chunk of functionality that a peer can initialize, start and stop.* »
 - Module de service
 - Module d'application
- Annonce de module
 - Annonce de spécification
 - *Module specification advertisement (MSA)*
 - Fonctionnalités du module
 - Annonce d'implémentation
 - *Module implementation advertisement (MIA)*
 - Distinction des implémentations (Java, C, COM)
 - Annonce de classe
 - *Module class advertisement (MCA)*
 - Classe de fonctionnalité



Exemple d'annonces de modules

Implémentation : modules 2/2

```
<?xml version=« 1.0 » encoding=«UTF-8»?>
<jxta:MSA>
  <MSID>          </MSID>
  <Name>          </Name>
  <Crtr>         </Crtr>
  <SURI>         </SURI>
  <Vers>         </Vers>
  <Desc>         </Desc>
  <Parm>         </Parm>
  <jxta:PipeAdvertisement>
</jxta:PipeAdvertisement>
  <Proxy>        </Proxy>
  <Auth>         </Auth>
</jxta:MSA>
```

```
<?xml version=«1.0» encoding=«UTF-8»?>
<jxta:MCA>
  <MCID>          </MCID>
  <Name>          </Name>
  <Desc>          </Desc>
</jxta:MCA>

<jxta:MIA>
  <MSID>          </MSID>
  <Comp>          </Comp>
  <Code>          </Code>
  <PURI>          </PURI>
  <Prov>          </Prov>
  <Desc>          </Desc>
  <Parm>          </Parm>
</jxta:MIA>
```

Annonces des modules de classe, spécification et implémentation

Implémentation : Amorce

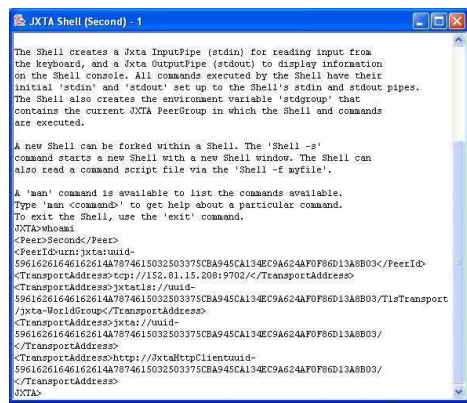
- Groupe
 - Sous-ensemble du réseau
 - Restriction service / groupe
- Amorce de Jxta
 - Création du **WorldPeerGroup**
 - Groupe de configuration local au *peer*
 - Identification particulière
 - Annonce des fonctionnalités du *peer*
 - Services
 - Transport
 - Applications
 - Annonce du groupe
 - Annonce du MIA
 - Services
 - Création ou Découverte du **NetPeerGroup**
 - Connexion au réseau P2P
 - Démarrage des services

Plan de la présentation

- Introduction
- Concepts
- Architecture & protocoles
- Implémentation
- Le shell Jxta
 - Principe
 - Exemples
- Jxta 2.0
- Conclusions

Le shell Jxta

- Objectifs
 - Interface ligne de commandes
 - ~ Shell Unix
 - Redirections
 - Variables d'environnement
 - Scripts
 - Interaction directe
 - Peers
 - Groupes
 - Pipes
 - Codats
 - Extension possible



```
JXTA Shell (Second) - 1
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>showall
<Peer>Seconds:/Peer>
<PeerId>urn:jxta:uid-
59616261646162614a7874615032503375CBA945CA134EC9A624AF0F86D13A8B03/<PeerId>
<TransportAddress>tcp://152.81.15.208:9702/</TransportAddress>
<TransportAddress>jxta:is://uid-
59616261646162614a7874615032503375CBA945CA134EC9A624AF0F86D13A8B03/TlsTransport
/jxta-WorldGroup/</TransportAddress>
<TransportAddress>jxta://uid-
59616261646162614a7874615032503375CBA945CA134EC9A624AF0F86D13A8B03/
</TransportAddress>
<TransportAddress>http://jxta:HttpClient/uid-
59616261646162614a7874615032503375CBA945CA134EC9A624AF0F86D13A8B03/
</TransportAddress>
JXTA>
```

Le shell Jxta : exemples 1/2

```
JXTA>peers -r
peer discovery message sent
JXTA>peers
peer0: name = SYSTEM@VSNETSVR@219.101.38.52
peer1: name = dream@db2.ddwave.com@211.10.20.66
...
peer10: name = JXTA.ORG 39
peer11: name = Doyen
JXTA>cat peer11
<?xml version="1.0"?>
<!DOCTYPE jxta:PA>
<jxta:PA xmlns:jxta="http://jxta.org">
  <PID>
    urn:jxta:uuid-
59616261646162614A78746150325033A0FB36BD121147C4B6E8AE123CCA411903
  </PID>
  <GID>
    urn:jxta:jxta-NetGroup</GID>
  <Name>
    Doyen</Name>
  ...
</jxta:PA>
```

Le shell Jxta : exemples 1/2

```
JXTA>groups -r
group discovery message sent
JXTA>groups
group0: name = GE323PNet
group1: name = iZooloo
...
group5: name = testGrp
group6: name = Hello
JXTA>join -d group6
# join - Stopping rdv
Doyen - Enter the identity you want to use for group 'Hello' :
Identity : waladze
JXTA>search -r
# search - JXTA Advertisement search message sent
JXTA>search
JXTA Advertisement adv0 [jxta:MIA]
JXTA Advertisement adv1 [jxta:MIA]
...
JXTA Advertisement adv8 [jxta:PipeAdvertisement]
JXTA Advertisement adv9 [jxta:RA]

JXTA>JXTA>inpipe = mkpipe -i adv8
JXTA>msg = recv inpipe
```

Plan de la présentation

- Introduction
- Concepts
- Architecture & protocoles
- Implémentation
- Le shell Jxta
- Jxta 2.0
 - Améliorations
 - Architecture
- Conclusions

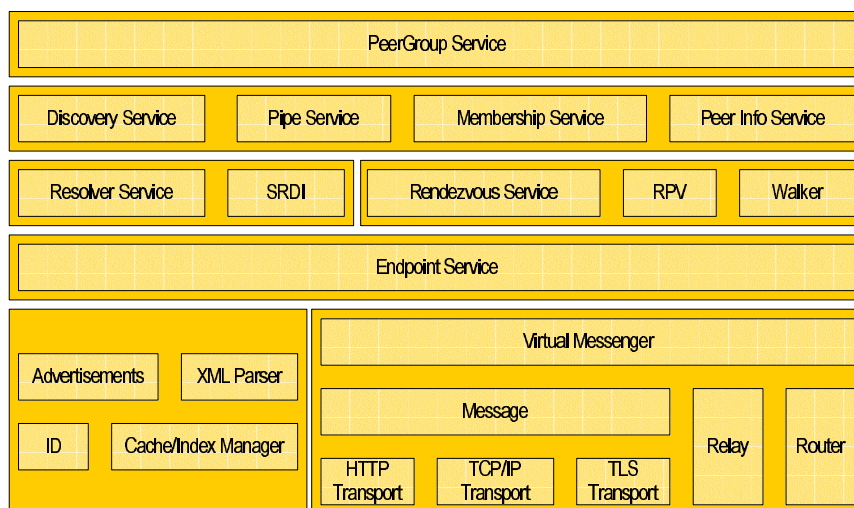
Jxta 2.0 1/3

- Améliorations
 - Performances
 - Passage à l'échelle
- Rendezvous peer
 - Index d'annonces
 - *Shared Resource Distributed Index (SRDI)*
 - Nouvelle propagation des requêtes
 - Réseau de *rendezvous peers*
 - Dynamique
 - Composé de *Peers* simples
 - ~ *Distributed Hash Table (DHT)*
 - *Rendezvous Peer View (RPV)*
 - Maintien de la cohérence
 - *Seeding rendezvous peers*
 - Propagation des requêtes
 - Voisins
 - *Walker*

Jxta 2.0 2/3

- *Peer* relais
 - Intermédiaire de connexion
 - *Firewall, NAT*
 - Hétérogénéité du transport
 - Dynamicité
 - Bail
 - *Peers*
 - Maintenance d'une liste de relais
 - Routes
 - Annonces
 - Association *PeerId / endpoint*
 - Indépendance de la source
 - Unidirectionnelles
- Sécurité
 - Modèle de confiance
 - Rôles (Rendezvous, relais, ...)
 - Transport (SSL, TLS, IPSec, ...)

Architecture fonctionnelle



Architecture Jxta 2.0

Plan de la présentation

- Introduction
- Concepts
- Architecture & protocoles
- Implémentation
- Le shell Jxta
- Jxta 2.0
- Conclusions

Conclusions

- Une vraie plate-forme de développement d'applications P2P
 - Spécification précise
 - Documentation
 - Tutoriels
- Interopérabilité
 - Compromis flexibilité / performances
- Communauté de développeurs
 - Plus de 9000 développeurs
 - Listes de diffusion
 - Exemples d'applications
- Domaine d'application croissant
 - Intégration
 - J2EE
 - Web Services
 - JMX

Références

- [1] Wilson, B. J. ; *Jxta* ; New Riders Publishing
- [2] Festor, O ; *Jxta : un framework pour le P2P* ; Cours ESIAL 3ème année
- [3] www.jxta.org
- [4] Oaks, S.and Travesat, B. and Gong, L. ; *JXTA in a Nutshell* ; O'Reilly & Associates ; 2002
- [5] Gradecki, J. and Wiley & Sons , *Mastering JXTA: Building Java Peer-to-Peer Applications* ; 1 edition ; 2002
- [6] Li, S. ; *Early Adopter JXTA: Peer-to-Peer Computing with Java* ; Wrox Press Inc ; ; 1st edition ; 2001
- [7] Govoni, D. and Krishnan, N. and Soto, J. C. ; *JXTA 2.0 : Java P2P Programming* by Brookshier, D.(Ed), Sams; 1st edition ; 2002