

JXTA

Alessandro Vasconcelos Ferreira de Lima

avfl@cin.ufpe.br

Roteiro

- Motivação
- Introdução
- Arquitetura de JXTA
- Elementos de JXTA
- Os Protocolos
- Comparações e Desvantagens
- Conclusão

Motivação

- Limitações do Modelo Cliente Servidor.
- O surgimento das comunidades P2P
 - Napster
 - Gnutella
- A busca pela interoperabilidade e independência entre serviços, protocolos, sistemas e linguagens.

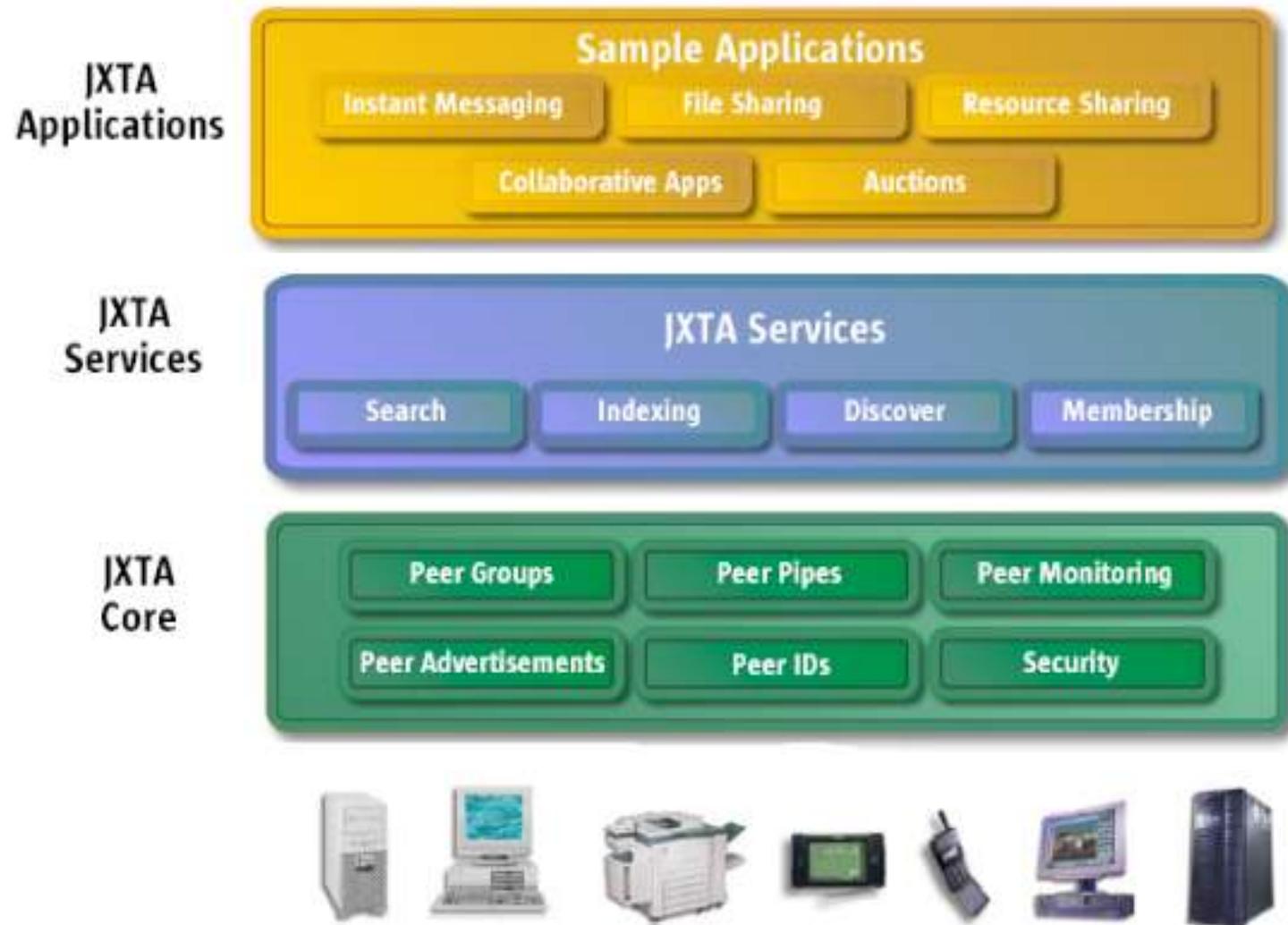
Introdução

- Arquitetura open source para desenvolvimento de aplicações P2P.
- Criada pela Sun Microsystems, e, hoje, mantida por colaboradores de todo o mundo.
- JXTA não vem de Java, vem do inglês juxtapose
- Tem como ideais:
 - Interoperabilidade.
 - Independência de plataforma.
 - Independência de linguagens.

Arquitetura de JXTA (1/2)

- A arquitetura de JXTA pode ser dividida em três camadas:
 - **Core:** encapsula as primitivas essenciais (descoberta de grupos e nós, transporte, etc.)
 - **Service:** acomoda serviços adicionais comumente utilizados ou desejáveis pelas aplicações (busca e indexação, diretórios, etc.)
 - **Application:** aplicações específicas que utilizam os serviços da rede.

Arquitetura de JXTA



Elementos de JXTA

- Peer
- Peer Group
- Transporte
- Serviço
- Advertisements

Peer (1/2)

- É qualquer entidade capaz de fazer algum trabalho útil e comunicar os resultados disso a outra entidade através da rede.
- São os nós numa rede P2P, que formam a unidade fundamental de qualquer solução P2P.
- Cada nó opera independente e de forma assíncrona dos outros.
- Os nós publicam os seus “peer endpoints”, por onde eles recebem conexões.

Peer (2/2)

- Os 3 principais tipos de peers são:
 - **Simple Peers:** servem a um usuário final. Podem prover serviços e/ou usar serviços de outros nós.
 - **Rendezvous Peers:** é usado para propagar mensagens dentro de um grupo de nós para um nó externo. Provêm para “Simple Peers” em uma rede privada a capacidade de fazer broadcast a outros membros do grupo que estão fora desta.
 - **Router Peers:** Provê mecanismos de comunicação com nós separados por um firewall ou por Network Address Translation (NAT).

Peer Group

- Dividem a rede em grupos de nós com objetivos em comum
- Pode prover serviços que não são acessados por outros nós da rede P2P.
- Pode utilizar autenticação para restringir o acesso aos serviços do grupo.
- Pode ser usado para prover serviços com tolerância a falha – se um membro ficar indisponível, outro pode tratar a requisição.

Transporte

- É responsável por todos os aspectos da transmissão de dados.
- Pode ser de baixo-nível (UDP ou TCP) ou alto-nível (HTTP ou SMTP)
- Pode ser dividido em 3 partes:
 - Endpoints: Corresponde as interfaces de rede usadas para mandar ou receber um dado.
 - Pipes: Canais virtuais de comunicação unidirecionais e assíncronos conectando dois ou mais endpoints.
 - Mensagens: Forma como os dados são transmitidos sobre um pipe de um endpoint a outro.

Serviços

- São funcionalidades oferecidas a nós remotos numa rede P2P.
- Podem ser transferência de arquivos, exibir status de nós, fazer um cálculo etc...
- Os serviços são divididos em duas categorias:
 - **Peer Services:** São funcionalidades oferecidas por um nó particular na rede a outros nós.
 - **Peer Group Services:** São funcionalidades oferecidas por um Peer Group a seus membros.

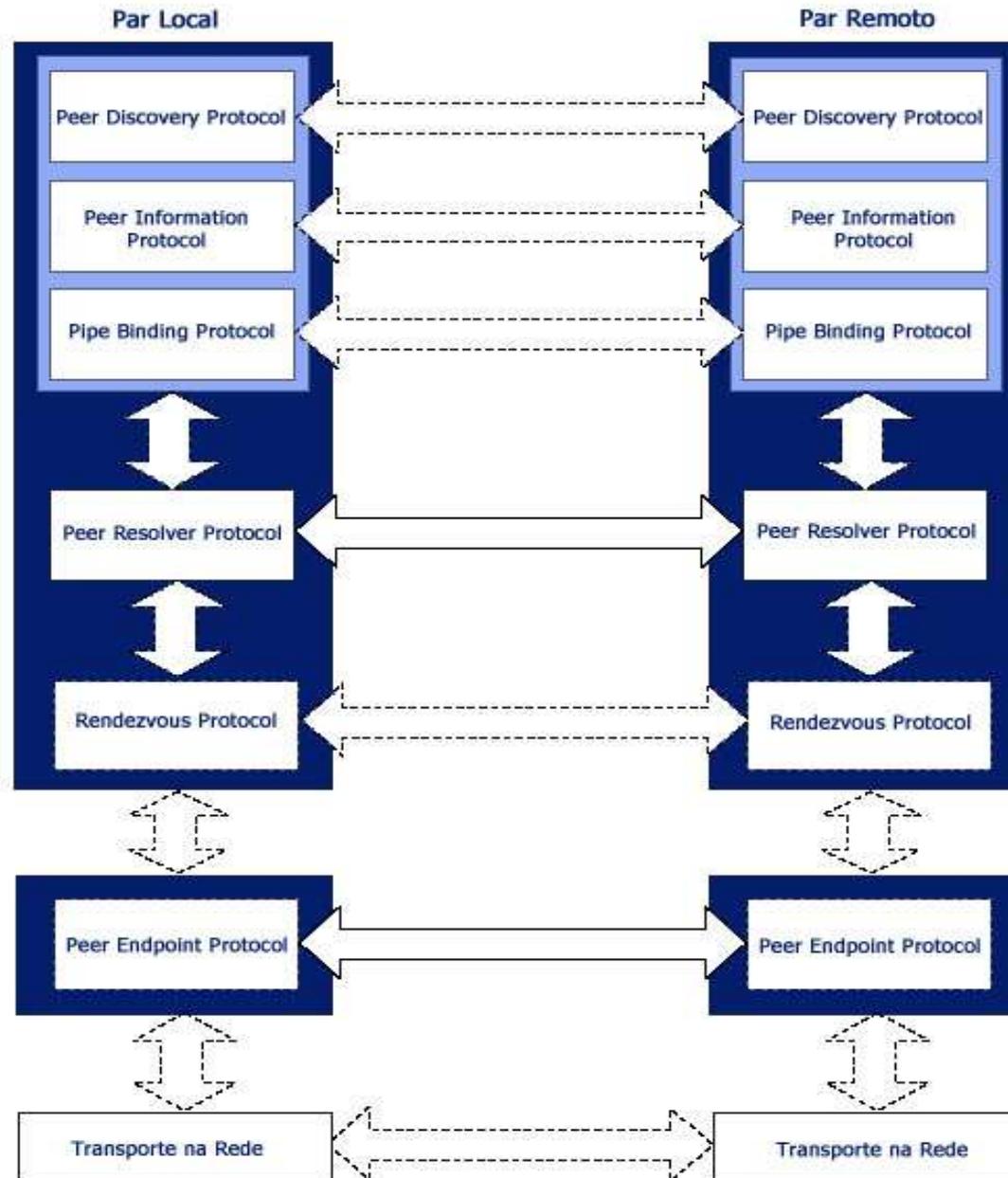
Advertisements

- São os anúncios utilizados para descoberta de serviços, nós, grupos e pipes.
- São representados por documentos XML e também utilizam TTL para manutenção da arquitetura.
- Na implementação para J2SE, é provido pelos rendezvous peers um serviço de indexação para otimizar a busca.

Os Protocolos de JXTA

- Visam oferecer as funcionalidades requeridas pelos pares numa rede.
- A especificação dos protocolos define estados que os nós da rede devem assumir para qualquer tipo de dispositivo.
- Foram definidos um conjunto de 6 protocolos, baseados em mensagens XML.
- Cada protocolo cobre um aspecto fundamental de uma rede P2P, e é dividido em uma parte responsável pelo nó local, e outra pelo nó remoto.

A Pilha de Protocolos



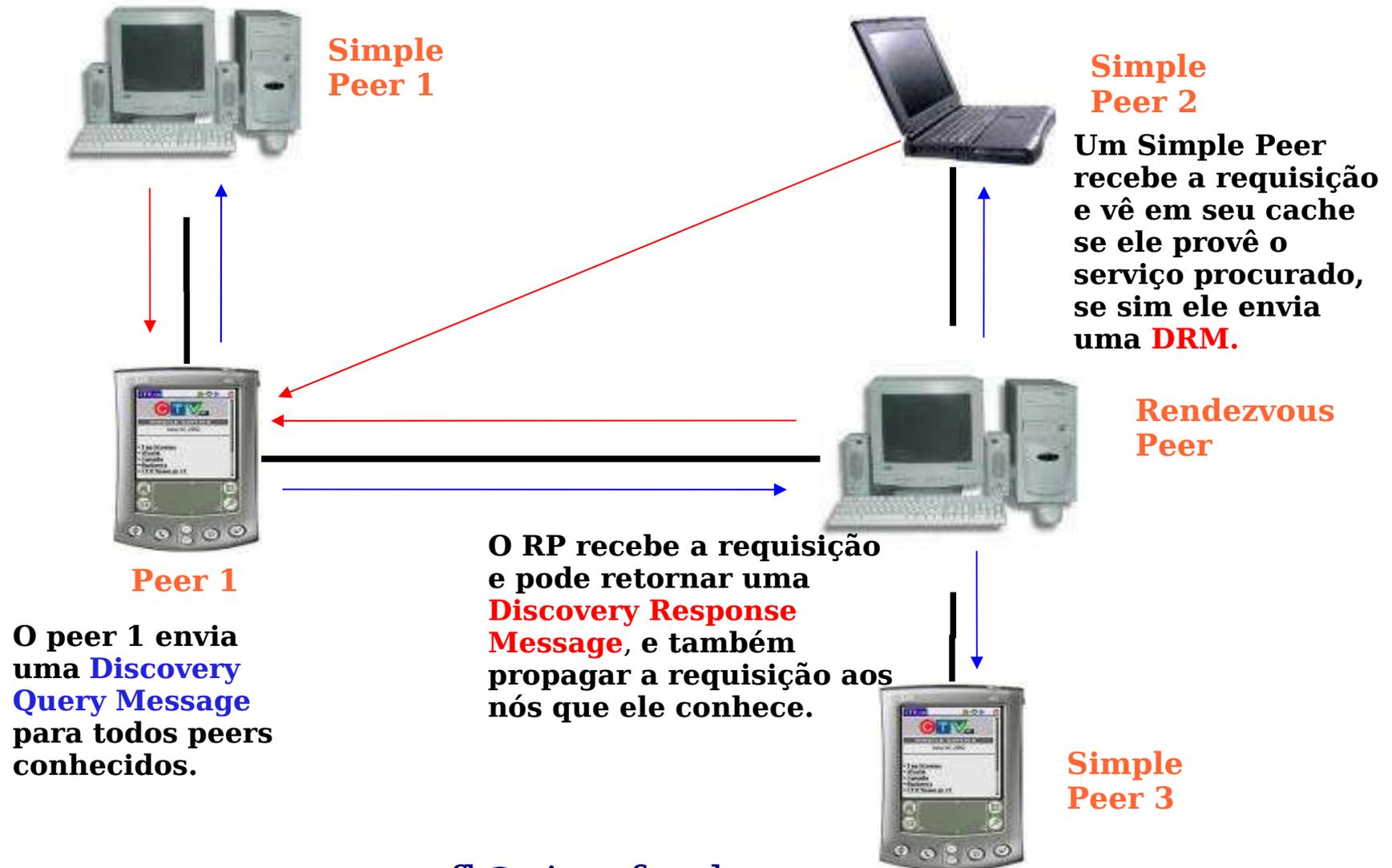
Peer Resolver Protocol

- O PRP define um protocolo de envio de uma mensagem genérica para um outro nó e processa uma resposta genérica para uma requisição.
- Possui 2 tipos de mensagens:
 - **Resolver Query Message:** Para enviar requisições.
 - **Resolver Response Message:** Para enviar respostas.

Peer Discovery Protocol(1/2)

- Definem como os nós requisitam notificação e respondem a requisição de outros nós.
- Os nós descobrem os recursos da rede enviando requisições a outros nós, geralmente a um “redesvour peer”.
- Peer Discovery Protocol dispõe de duas mensagens:
 - **Discovery Query Message:** uma mensagem de requisição, para perguntar por recursos.
 - **Discovery Response Message:** uma mensagem de resposta para a mensagem anterior.

Peer Discovery Protocol(2/2)



Rendezvous Protocol (1/3)

- É responsável por reservar um usuário para propagar mensagens para outros nós através de um Rendezvous Peer e também por prover rendezvous peer services para outros nós na rede.
- Pra isso cada nó deve conectar-se ao rendezvous peer e obter um "aluguel". Esse "aluguel" especifica o tempo que um nó poderá usar o rendezvous peer, antes de precisar renovar esse "aluguel".

Rendezvous Protocol (2/3)

- Para prover essas funcionalidades o RVP define três tipos de mensagens:
 - **Lease Request Message:** é enviada quando um nó requisita um aluguel ao rendezvous peer.
 - **Lease Granted Message:** é enviada pelo rendezvous peer para aprovar uma Lease Request Message e determinar o tempo do aluguel.
 - **Lease Cancel Message:** é enviada por um par para se desconectar do rendezvous peer.
- Aqui as mensagens não são especificada e definidas em XML, por questão de eficiência é usada uma representação compacta binária para a transmissão.

Rendezvous Protocol (3/3)



Simple Peer 2

O peer 1 recebe a LGM e começa a mandar **mensagens** ao RP para serem propagadas aos outros peers.



Peer 1

O peer 1 quer usar o Rendezvous Peer para propagar sua mensagem. Ele deve antes "alugar" uma conexão mandando uma **Lease Request Message**.



Rendezvous Peer

Após receber a LRM, o RP decide se aluga a conexão. Caso alugue ele cria uma conexão e manda uma **Lease Granted Message** ao peer 1.



Simple Peer 3

Peer Information Protocol(1/2)

- Permite que um nó obtenha informações sobre o status de outros nós previamente descobertos.
- Esse status pode ser o tempo no ar e a quantidade de tráfego processada por um nó remoto por exemplo.
- Um nó pode querer monitorar o status do nó remoto para tomar decisões sobre como usar-lo mais efetivamente.

Peer Information Protocol(2/2)

- Monitoramento é uma parte essencial de uma rede P2P, prover informação dos nós leva a um uso mais eficiente dos recursos da rede.
- Usa dois tipos de mensagens:
 - **Peer Info Query Message:** é enviada para requisitar o status de um nó remoto.
 - **Peer Info Response Message:** é enviada para prover o status de um nó a outros nós.

Pipe Binding Protocol(1/2)

- Define o processo de ligação de um pipe a um endpoint.
- Antes de um pipe ser usado ele deve ligado a um endpoint de um nó.
- Ligando um pipe a um endpoint cria-se um input pipe para receber dados ou um output pipe para mandar dados.

Pipe Binding Protocol(2/2)

- O PBP define um conjunto de mensagens que um nó pode usar para buscar nós remotos que tenham um endpoint apropriado para um dado pipe.
- Essas mensagens são de dois tipos:
 - **Pipe Binding Query Message:** é enviada para requisitar um nós remoto se ele está ligado a um pipe com o mesmo Pipe ID.
 - **Pipe Binding Answer Message:** é enviada para responder a requisição.

Endpoint Routing Protocol(1/2)

- Provê um mecanismo para determinar a rota para um endpoint.
- A funcionalidade do endpoint não tem um protocolo definido.
- A forma como os dados serão enviados na rede, são de responsabilidade de uma implementação particular do protocolo.
- O desenvolvedor poder usar o Endpoint Service para obter uma implementação particular do ERP, embora isso torne a solução menos flexível.

Endpoint Routing Protocol(2/2)

- Tipos de implementação do ERP disponíveis:
 - **TCP:** Usa um socket para conectar-se diretamente ao nó remoto.
 - **HTTP:** Tem como vantagem passar por firewalls, mas não fornece broadcast.
 - **Servlet HTTP:** Para ser usado em aplicações que suportem Servlets.
 - **TLS:** Provê comunicação segura, com uso de autenticação.

Comparações

- JXTA vs JINI
 - Tem alguns objetivos em comum.
 - JINI é dependente de LP, JXTA não.
 - JINI usa RMI, JXTA usa XML.
- JXTA vs .NET
 - Propósitos diferentes
 - .NET é baseado em cliente servidor

Desvantagens

- Overhead por não estar presa a um tipo de protocolo de transporte.
- Dependendo da aplicação o overhead de mensagens XML pode ser um problema.
- Não dá suporte a padrões de invocação de serviços como Web Service Description Language.

Conclusão

- É preciso balancear flexibilidade com performance na hora de implementar aplicações P2P. JXTA pode não ser a solução mais eficiente, mas com certeza é a melhor plataforma para produzir aplicações flexíveis e integrá-las num ambiente P2P.

Referências

- www.jxta.org - Site oficial do projeto.
- www.sunsource.net - Site onde podem ser encontrados todos os projetos OpenSource da Sun.

Obrigado!
Essa apresentação foi feita
usando software open source.