

Lezione n.11

JXTA: concetti generali
Materiale didattico
distribuito a lezione
Laura Ricci

JXTA: MOTIVAZIONI

- JXTA: tecnologia sviluppata da Sun Microsystems (team Bill Joy e Mike Clary) per lo sviluppo di applicazioni P2P
- JXTA: deriva da **juxtapose = giustapposto**
la tecnologia P2P si deve affiancare, piuttosto che sostituire, la tecnologia client/server (o web based)
- Motivazioni:
 - ogni applicazione P2P è basata su un **protocollo proprietario**
 - l'incompatibilità dei protocolli rende difficile l'integrazione di servizi offerti da reti P2P diverse
 - ogni rete P2P definisce una comunità chiusa, indipendente da altre comunità
 - necessità di definire un insieme di protocolli generali che possano essere utilizzati da applicazioni diverse

JXTA: SCELTE DI PROGETTO

- JXTA = insieme aperto di protocolli P2P
- Obiettivi:
 - Interoperabilità
 - Ubiquità
 - Indipendenza dalla piattaforma utilizzata
- **Interoperabilità**
i protocolli definiti possono essere utilizzati da ogni dispositivo connesso alla rete P2P per **comunicare** ed **interagire** con gli altri dispositivi
- **Ubiquità**
ogni dispositivo con un "digital hearthbeat" deve poter integire tramite una rete JXTA
cellulari, palmari, PC, Servers, supercomputers

JXTA: SCELTE DI PROGETTO

- Indipendenza da:

piattaforma (Windows, UNIX,...)

linguaggio

Esistono binding per JAVA, C, C++.....

i protocolli JXTA possono utilizzare diversi protocolli di rete TCP/IP, HTTP, Bluetooth,....

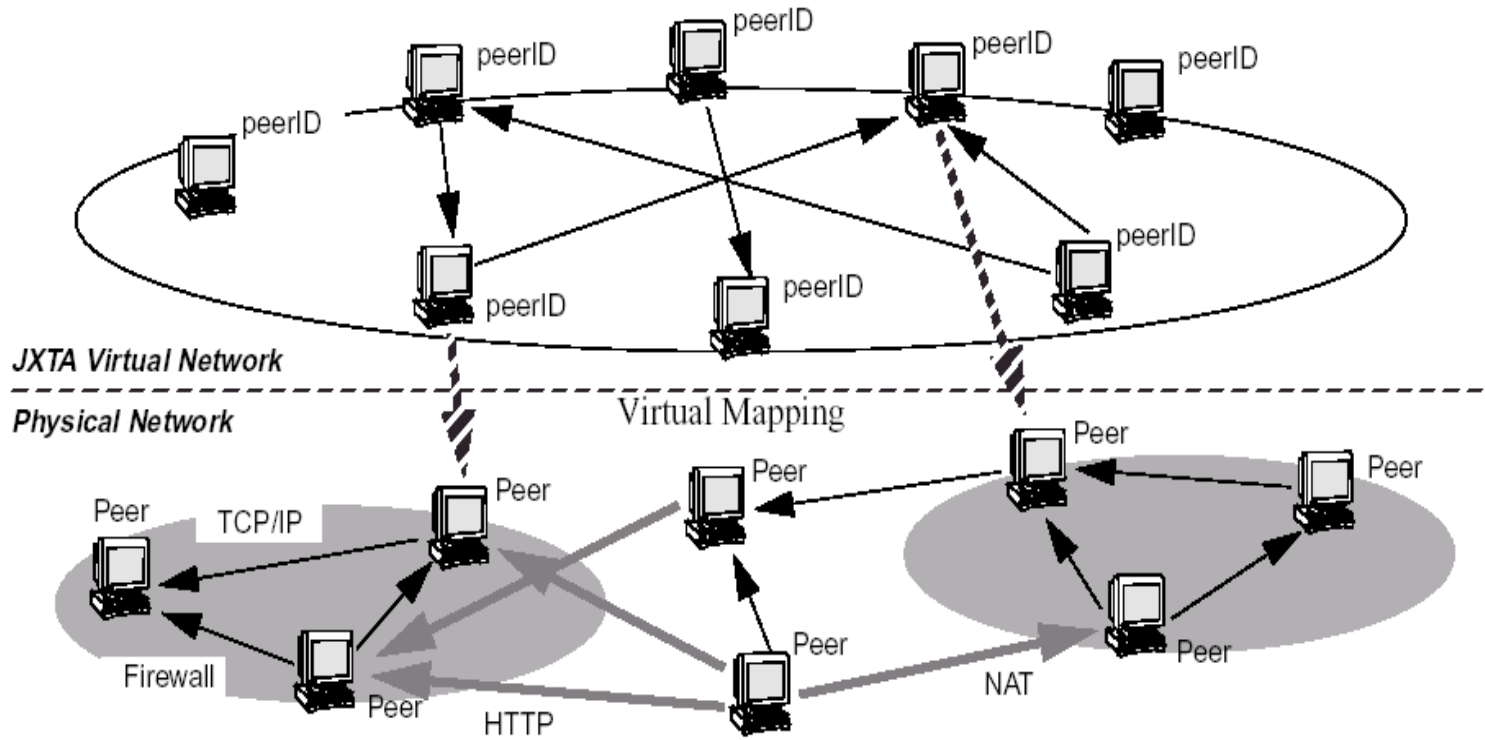
definizione dei servizi di rete

RMI, WSDL (Web Service Description Language)

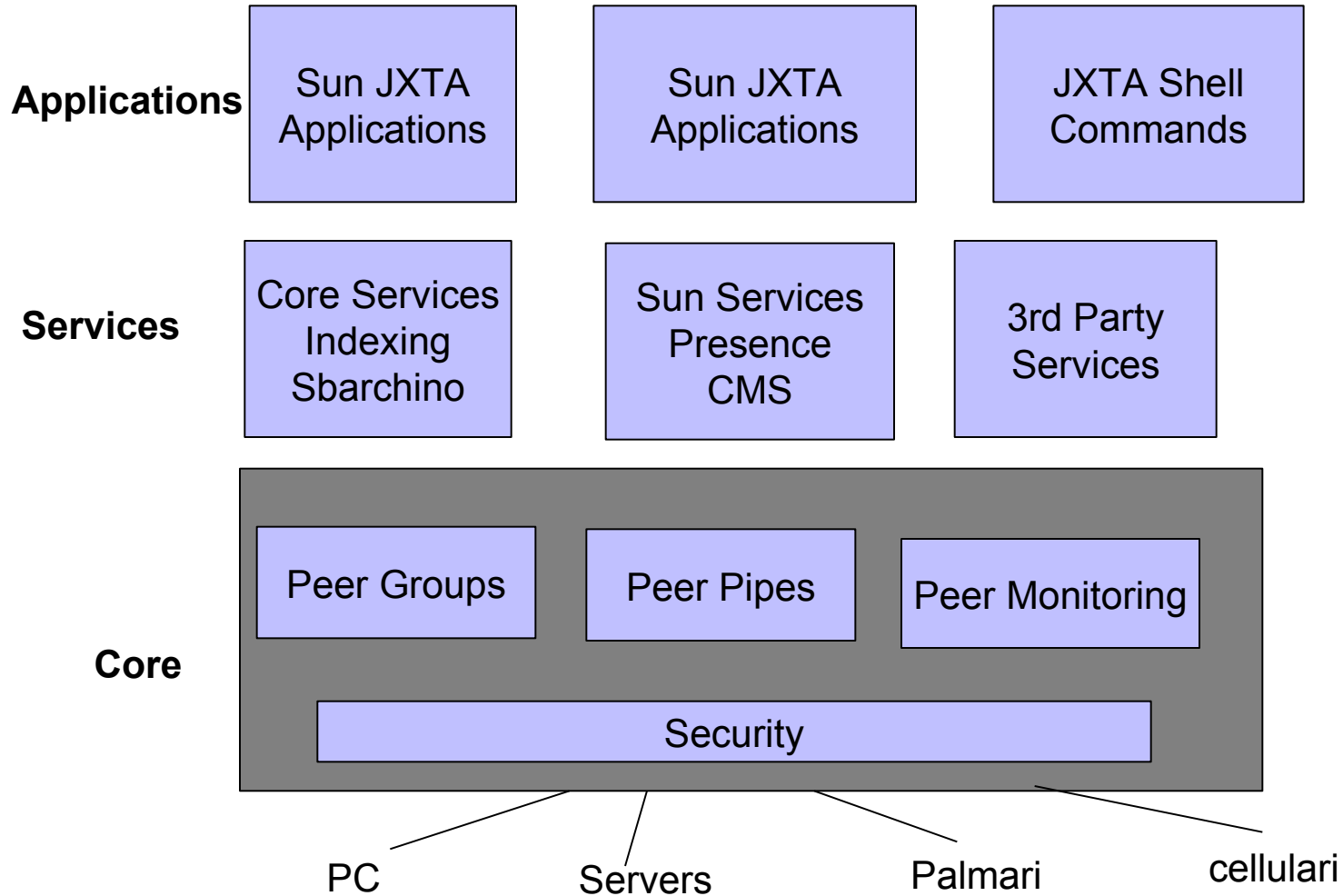
JXTA: SCELTE DI PROGETTO

- JXTA = definisce un insieme di protocolli che standardizzano il modo in cui i peer:
 - scoprono altri peer all'interno della rete
 - si auto-organizzano in peer groups
 - pubblicano e scoprono servizi di rete
 - comunicano direttamente (**meccanismo delle pipes**)
 - controllano l'attività degli altri peers (monitoraggio)
- ogni protocollo definisce un insieme di messaggi che vengono scambiati tra i partecipanti
- per ogni messaggio viene definito un formato pre-definito

JXTA: LA RETE VIRTUALE



JXTA LAYERS



JXTA LAYERS

- **JXTA Core:** incapsula un insieme minimo di primitive essenziali per il supporto di applicazioni P2P
 - creazione di peers e di gruppi di peers
 - sicurezza
 - scoperta di peers, servizi, risorse della rete
 - invio messaggi, gestione firewalls e NAT
- **JXTA Services:** definisce un insieme di servizi non strettamente necessari per una rete P2P, ma comuni nella maggior parte degli ambienti P2P (es: file sharing, distributed file systems,,...)
- **Applicazioni:** sviluppate da SUN (MyJXTA,...) e in genere dalla comunità di utenti

JXTA: I PEERS

- Peer: un qualsiasi dispositivo connesso ad una rete JXTA che implementa uno o più protocolli JXTA
- Ogni peer
 - è identificato da un identificatore unico (JXTA ID)
urn:jxta:uuid-
5966261646261624A7874615032033F3BC76FF13CBC0AB663666DA53903
 - è caratterizzato da un tipo che definisce l'insieme di responsabilità che il peer decide di assumersi
- Classificazione dei peers
 - Peer Semplici
 - Minimal Edge Peer
 - Full Fetured Edge Peer
 - Rendezvouz Peer
 - Relay (o routers) Peers

JXTA: CLASSIFICAZIONE DEI PEERS

Minimal Edge Peer

peer con risorse limitate (palmare, cellulare,...), possibilmente allocato a monte di un NAT o di un firewall. A questi peer sono attribuite responsabilità minime

può inviare e ricevere messaggi

non inoltra messaggi e non memorizza informazioni per conto di altri peer

non possiede una cache per memorizzare informazioni scoperte sulla rete

Full featured Edge Peer

ha le stesse funzionalità di un minimal edge peer

possiede una cache locale dove memorizza informazioni ricevute dalla rete

JXTA: CLASSIFICAZIONE DEI PEERS

Rendez-vous peer: (Rendez-vous = Punto di incontro)

- ha le stesse funzionalità di un full-edged peer
- offre ulteriori servizi
 - routing: può inoltrare queries per conto di altri peers
 - memorizzazione: può memorizzare riferimenti ad informazioni memorizzate su altri peers
- per offrire questi servizi deve possedere risorse di calcolo e di memorizzazione superiori a quelle di un edge peer

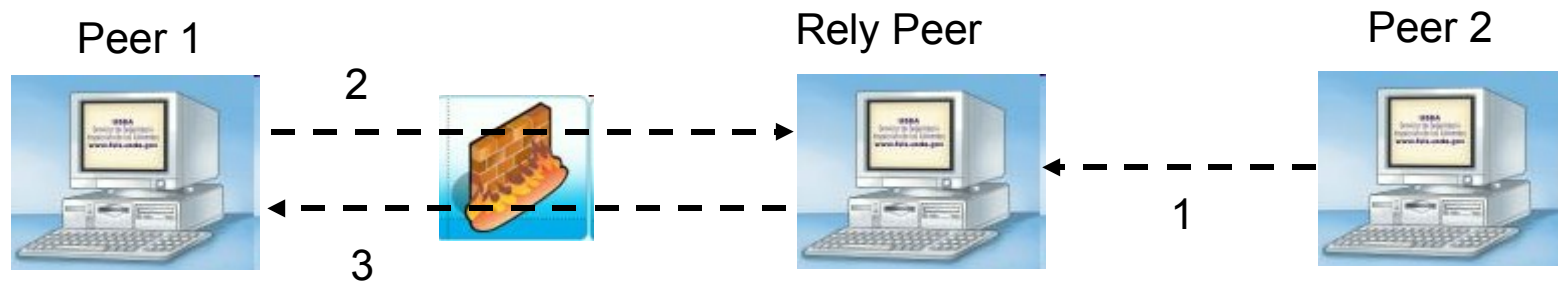
Relay Peer: (Relay=fare affidamento su) inoltrano messaggi per conto di peers che si trovano a monte di NATs o di firewalls

JXTA RELY PEERS

- **Rely Peers** utilizzati per permettere la connessione ad una rete JXTA di peer che si trovano all'interno:
 - di una rete protetta da un firewall oppure
 - di una rete privata i cui indirizzi privati vengono mappati da un NAT in indirizzi pubblici
- **Firewalls**
 - bloccano connessioni provenienti dall'esterno verso gli hosts della rete protetta
 - restringono le connessioni aperte verso l'esterno da hosts della rete protetta (es: solo protocollo HTTP, porte utilizzate,...)
- **NAT**
 - un peer esterno alla rete privata può comunicare con un host H della rete solo se H inizia la comunicazione

JXTA RELY PEERS: FIREWALL O NAT SINGOLO

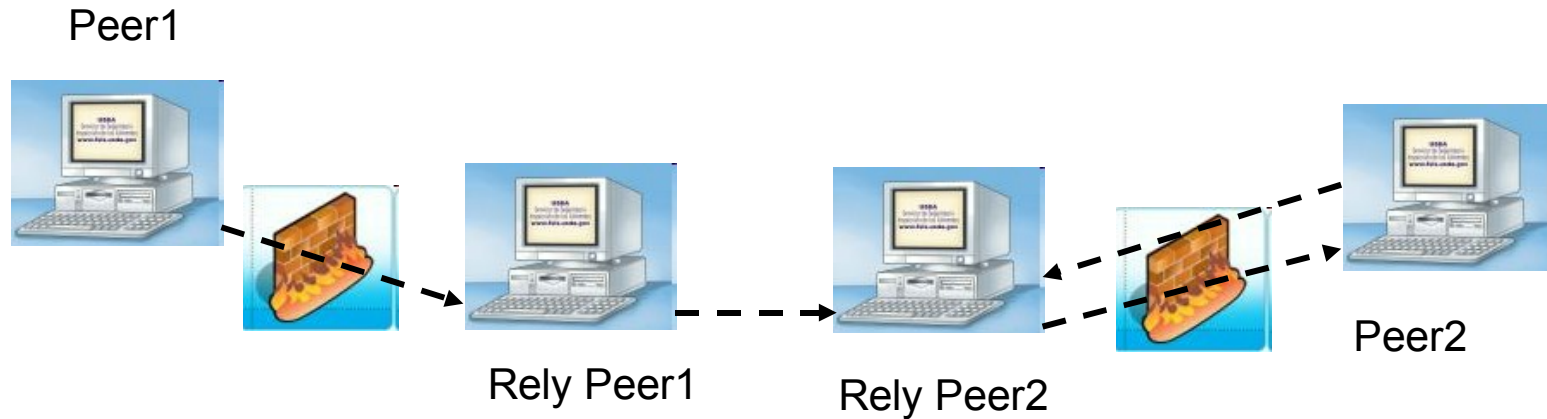
Un peer appartenente ad una rete privata protetta da un firewall utilizza un **rely peer** esterno al firewall (o appartenente alla rete privata, ma visibile dall'esterno) per attraversare il firewall



1. Il Peer2 non può connettersi con il Peer1, che si trova a monte di un firewall. Peer2 si connette al Rely Peer ed invia ad esso i messaggi destinati a Peer1. Il Rely Peer inoltra a Peer1 i messaggi ricevuti da Peer2
2. Peer1 si connette periodicamente al rely peer per dare ad esso la possibilità di inoltrare i messaggi ricevuti da Peer2. Peer1 utilizza un protocollo che gli consente di attraversare il firewall (es: HTTP)
3. Il rely peer invia a Peer1, per conto di Peer2 i messaggi ricevuti

JXTA RELY PEERS: FIREWALL/NAT MULTIPLI

Se i due peer che intendono comunicare si trovano entrambi a monte di un firewall/NAT si utilizza una coppia di rely peers



JXTA RELY PEERS

- Per inviare un messaggio ad un peer posizionato dietro ad un firewall, occorre conoscere il rely peer corrispondente
- Questa informazione può essere ottenuta mediante un processo di discovery di routing informations effettuato in modo distribuito
- L'invio di messaggi diversi tra la stessa coppia di peers può utilizzare diversi rely peers

JXTA: PEER GROUPS

- nelle applicazioni P2P 'classiche' (Gnutella, ICQ, Freenet), ogni applicazione definisce una diversa overlay network
- ogni overlay network utilizza un protocollo diverso
- in JXTA: lo stesso insieme di protocolli utilizzato per applicazioni diverse
- **JXTA Peer Groups**: suddividono l'overlay network JXTA in base dell'applicazione a cui i peer partecipano
- **Peer Group**= Insieme di peers caratterizzati da **interessi o scopi comuni**. Ogni gruppo fornisce ai partecipanti un insieme di servizi che non sono accessibili da peers JXTA che non partecipano al gruppo

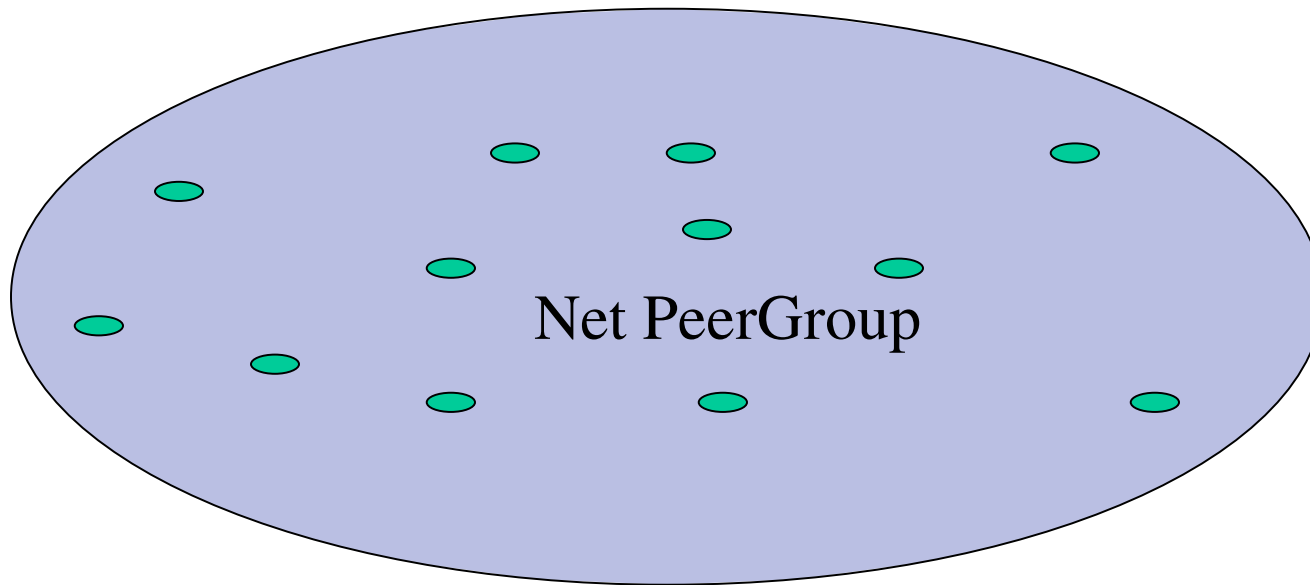
JXTA: PEER GROUPS

- JXTA definisce un insieme di protocolli per gestire la creazione dei gruppi, la richiesta di partecipazione ad un gruppo,...
- I gruppi vengono creati dalle applicazioni
- I peer groups definiscono delle 'regioni astratte' all'interno della rete JXTA
- Un peer group viene utilizzato per definire:
 - un insieme di servizi e risorse accessibili dai partecipanti al gruppo
 - un insieme di politiche di sicurezza associate al gruppo (membership requirements)
 - un meccanismo di scoping
 - un ambiente di monitoraggio

JXTA: PEER GROUPS

- i peer si organizzano dinamicamente in peer groups
- all'inizio della esecuzione ogni peer appartiene, per default, a `NetPeerGroup`, un gruppo che fornisce un insieme di servizi base
- ogni peer può unirsi a diversi gruppi durante la sua esecuzione
- ogni gruppo viene creato a partire da un un unico gruppo esistente
- `Albero di peer groups` = descrive le relazioni tra i gruppi attivi durante l'esecuzione di un programma JXTA
- Ogni gruppo offre un insieme di servizi ai peer che partecipano al gruppo

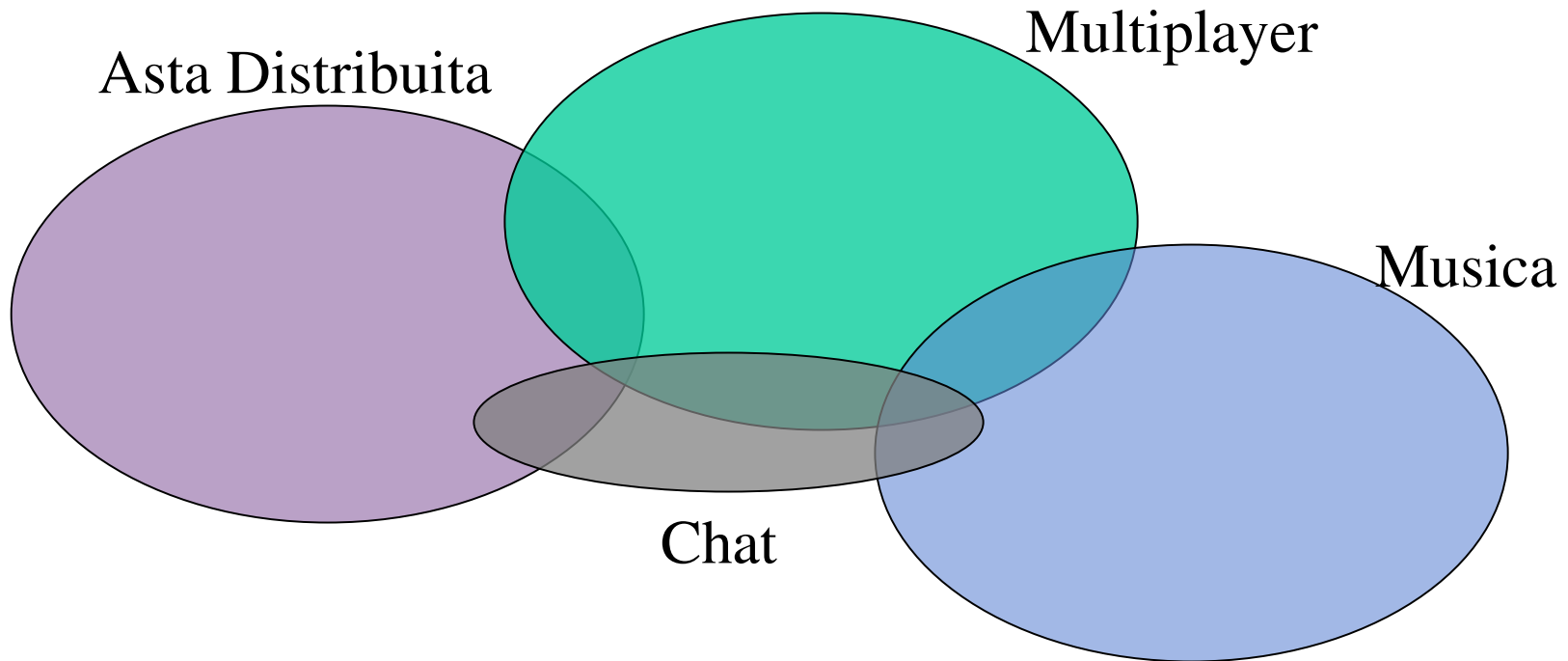
IL NET-PEER-GROUP



All'inizio della esecuzione, ogni peer

- è membro del NetPeerGroup
- può scoprire risorse all'interno di quel peer group e comunicare con altri peer appartenenti al peer group

JXTA PEER GROUPS



- Ogni peer può creare, unirsi e lasciare un PeerGroup
- Ogni peer può appartenere a più di un peer group contemporaneamente
- Le ricerche di risorse vengono effettuate rispetto ad un peergroup di riferimento

JXTA: MECCANISMI DI COMUNICAZIONE

- Meccanismi definiti da JXTA per realizzare lo scambio di messaggi tra servizi/ applicazioni
 - Pipe:** canale di comunicazione virtuale **unidirezionale ed asincrono**.
Connette uno o più endpoints
 - Pipe EndPoint:** estremo della pipe, **input pipe** (estremo da cui si ricevono i dati) ed **output pipe** (estremo su cui si immettono i dati)
 - Peer EndPoint:** astrae il concetto di interfaccia di un peer (Es: indirizzo IP+ porta)
 - Messaggi:** Contenitori utilizzati per trasmettere dati sulla rete
- Il meccanismo delle pipe definisce un'astrazione che fornisce l'illusione di disporre di **mailboxes virtuali** che non fisicamente collegate ad uno specifico peer.

JXTA PIPES

- Le pipes possono connettere due o più endpoints
- Gli estremi di una pipe vengono collegati dinamicamente ad uno o più peer endpoints, mediante il pipe binding protocol
- la stessa pipe può quindi essere collegata a tempo di esecuzione a diversi peer
utile per mantenere il sistema tollerante ai guasti
un peer guasto P1 può essere sostituito da un nuovo peer P2
P2 prende la responsabilità di gestire le comunicazioni dirette a P1,
collegando ad un proprio end point gli estremi delle pipe utilizzate in
precedenza da P1

COMUNICAZIONE MEDIANTE PIPES

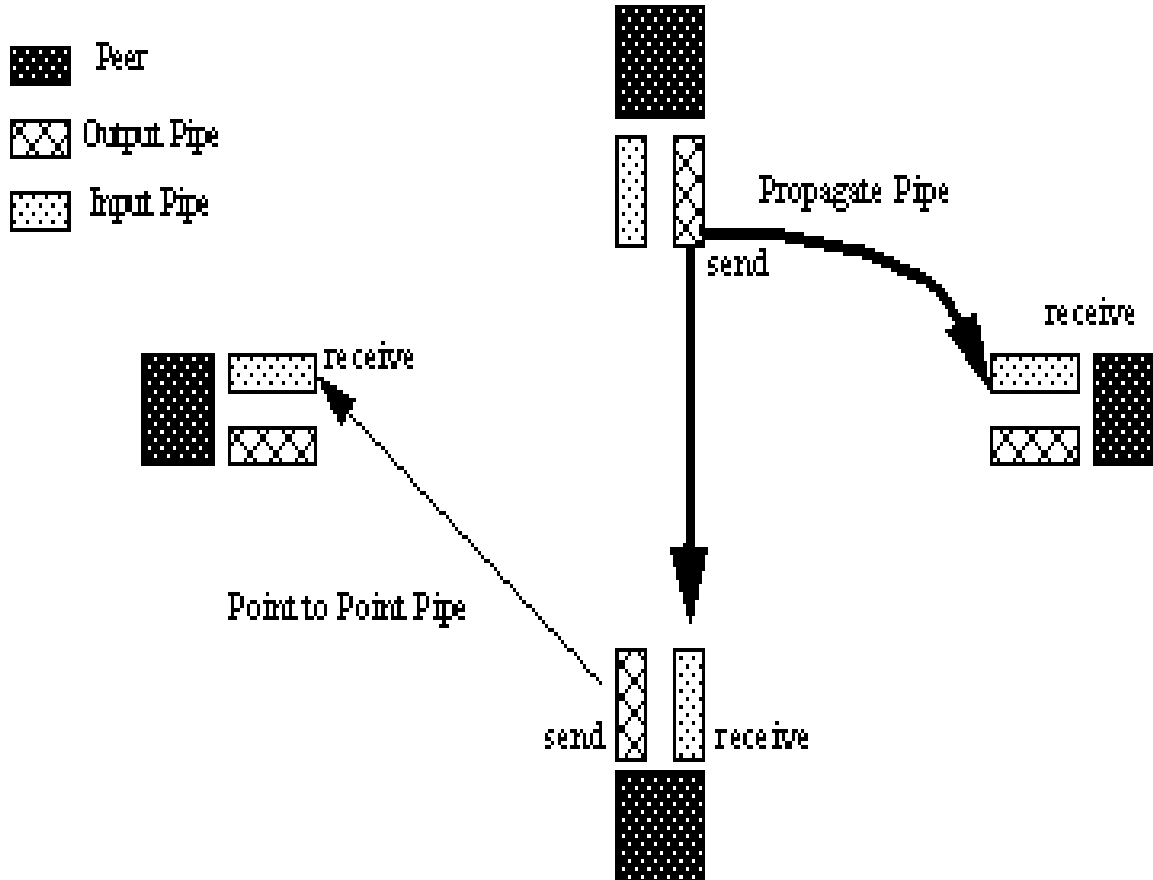
- Le pipes sono *risorse condivise* all'interno di un peer group
- Un peer può
 - creare una pipe e collegarne un estremo ad un proprio endpoint
 - pubblicare la pipe all'interno di un gruppo, in modo da renderne nota la esistenza a tutti i membri del gruppo
 - ricercare una pipe per comunicare con un altro peer
- Per comunicare, una coppia di peer deve:
 - utilizzare la stessa pipe (canale)
 - il mittente connette dinamicamente un estremo della pipe ad un suo endpoint (output pipe)
 - il destinatario connette dinamicamente l'altro estremo della pipe ad un suo endpoint (input pipe)
 - pipe binding protocol: protocollo per la gestione delle pipes

JXTA: TIPI DI PIPES

- **Point-to-point (Unicast) Pipe:** mette in collegamento due peer. Utilizza il protocollo TCP/IP, non effettua il crittaggio dei messaggi
- **Secure Unicast Pipes:** point to point pipe che fornisce un servizio di comunicazione
- **Propagate Pipes:** pipe di tipo **uno a molti**. Connette una output pipe ad un insieme di input pipes. Le input e le output pipes devono appartenere al solito gruppo

Esempio: implementazione di un servizio di chat. Un messaggio deve essere trasmesso a tutti i partecipanti ad una chat

JXTA: TIPI DI PIPES



JXTA: MESSAGGI

- Messaggio = oggetto scambiato tra JXTA peers
- Messaggio = insieme di coppie (nome, valore)
- Rappresentazione dei messaggi
 - Binaria
 - XML

JXTA: ADVERTISEMENTS

- **Advertisement** = Rappresentazione strutturata di una entità, di un servizio, o di una risorsa resa disponibile da un peer all'interno di un peer group
- Rappresentazione mediante documenti XML
- Ogni peer
 - pubblica un insieme di advertisements per descrivere le risorse che mette a disposizione
 - ricerca advertisements sulla rete
- Ogni advertisement è caratterizzato da un ciclo di vita, dopo cui viene eliminato dalla rete
- Ogni peer mantiene **una cache** locale in cui memorizza gli advertisements ricevuti dalla rete

JXTA UN ESEMPIO DI ADVERTISEMENTS

```
<?xml version="1.0"?>
```

```
<!DOCTYPE jxta: PipeAdvertisement>
```

```
<jxta: PipeAdvertisement xmlns:jxta="http://7jxta.org">
```

```
  <id>
```

```
urn:jxta:uuid
```

```
59616261646162614E504720503250338E3E786229EA460DADC1A176B69B7354
```

```
  </id>
```

```
  <type>
```

```
    JxtaUnicast
```

```
  </type>
```

```
  < name
```

```
    TestPipe
```

```
  </name>
```

```
</jxta:PipeAdvertisement>
```

JXTA: TIPI DI ADVERTISEMENTS

- peer advertisements
 - nome del peer, identificatore del peer, endpoints, se è un rendez-vous peer,...
- peer group advertisement
 - nome del gruppo, identificatore, servizi offerti,....
- pipe advertisement
 - anche **una pipe** è considerata una risorsa condivisa. Il peer che la crea, **ne notifica l'esistenza** agli altri peer del gruppo tramite un advertisement
 - identificatore, tipo della pipe,....
- peer info advertisements
 - contiene informazioni sullo stato del peer, ad esempio, l'uptime, quanti messaggi ha ricevuto in un certo intervallo di tempo.

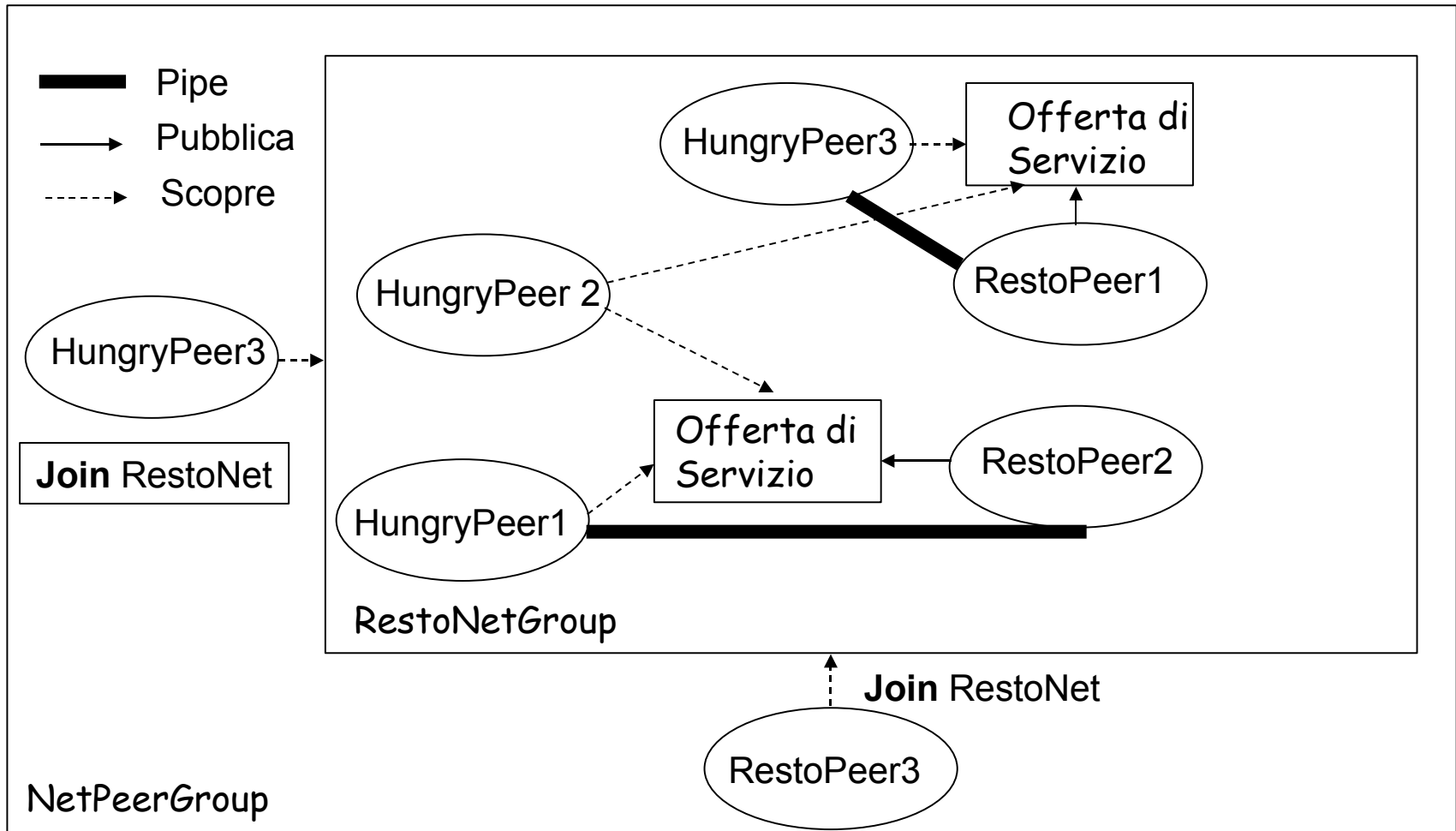
Altri tipi di advertisements verranno introdotti quando descriveremo altre risorse/servizi corrispondenti

JXTA: UN ESEMPIO

Si vuole progettare un'applicazione P2P per la ricerca distribuita di servizi di ristorazione. Il sistema prevede:

- un **peergroup RestoNet** a cui si uniscono i ristoratori che presentano alla comunità le proprie proposte di servizio ed i clienti che ricercano le offerte più vantaggiose per loro
- **RestoPeer**: forniscono servizi di ristorazione. Pubblicano le proprie offerte e rispondono alle richieste di servizio che possono soddisfare.
- **Hungry Peers**: sono alla ricerca di un locale in cui ristorarsi. Ricercano offerte di ristorazione. Possono selezionare il ristorante in base a diversi criteri (prezzo, tipo di cibo, distanza,....)

L'APPLICAZIONE RESTONET



L'APPLICAZIONE RESTONET

Il RestoPeer

- effettua una ricerca per verificare se RestoNetGroup è già stato creato
- se la ricerca ha esito positivo si unisce al gruppo, altrimenti crea un nuovo gruppo, RestoNetGroup
- crea e pubblica una pipe RP da cui riceve le richieste di servizio da parte degli utenti
- Si mette in attesa di richieste e risponde ad esse

L'HungryPeer

- effettua una ricerca per verificare se RestoNetGroup è già stato creato
- se la ricerca ha esito positivo si unisce al gruppo, altrimenti termina
- ricerca annunci di ristorazione all'interno di RestoNetGroup (ricerca le pipes a cui connettersi per comunicare con i ristoranti)
- crea una pipe HP per ricevere le risposte dai ristoranti
- si connette ad un ristorante: collega l'input pipe ad un suo endpoint
- invia la richiesta ad uno dei ristoranti individuati ed attende la risposta su HP
- se la risposta non è soddisfacente si connette ad un altro ristorante

RICERCA DI UN GRUPPO

Ricerca e creazione/unione del/al gruppo RestoNetGroup

- Ogni gruppo è descritto mediante **un advertisement**
- RestoPeer ricerca un advertisement per RestoNetGroup
 - ricerca nella propria cache locale, per verificare se un advertisement per lo stesso RestoNetGroup è già stato creato
 - in caso negativo, invia una richiesta sulla rete per scoprire se esiste un advertisement per quel gruppo (utilizza PDP Peer Discovery Protocol)
 - JXTA crea un nuovo thread che attende dalla rete eventuali risposte alla richiesta
 - attende eventuali risposte per un certo intervallo di tempo.

CREAZIONE DI UN GRUPPO

Ricerca e creazione/unione del/al gruppo RestoNetGroup

- trascorso un certo intervallo di tempo se RestoPeer non ha ricevuto risposte, crea un nuovo RestoNetGroup, altrimenti si unisce a RestoPeerGroup
- creazione di un nuovo gruppo
 - creazione di un nuovo advertisement che descrive il nuovo gruppo
 - creazione del gruppo con le caratteristiche specificate nell'advertisement
 - al gruppo creato viene associato **un insieme di servizi**. Per default, si possono ereditare i servizi standard associati a NetPeerGroup
 - discovery
 - membership
 - pipe binding
 -
 - pubblicazione dell'advertisement che descrive il gruppo

RICHIESTA DI PARTECIPAZIONE AD UN GRUPPO

Richiesta di partecipazione ad un gruppo già esistente

- l'advertisement di RestoNetGroup, reperito nella rete, viene utilizzato per richiedere la partecipazione al gruppo
- al momento richiesta di partecipazione può essere sfruttato il servizio di membership del gruppo per autenticare il nuovo peer che si unisce al gruppo
- dopo essersi unito al gruppo, il peer può usufruire di tutti i servizi offerti dal gruppo

JXTA: RICERCA DI UN GRUPPO

1. Creazione del riferimento a NetPeerGroup e quindi reperimento del servizio di discovery associato a tale gruppo

```
try
```

```
{ PeerGroup netpg = PeerGroupFactory.newNetPeerGroup ( );  
}
```

```
catch (PeerGroupException e) {
```

```
System.exit(1); }
```

```
DiscoveryService hdisco = netpg.getDiscoveryService ( );
```

2. Ricerca di un advertisement per RestoPeerGroup
 - ricerca prima nella cache locale, poi invia una richiesta di ricerca sulla rete
 - la ricerca remota viene ripetuta un certo numero di volte
 - utilizza hdisco per effettuare la ricerca nella cache locale e la ricerca remota

JXTA: RICERCA DI UN GRUPPO

```
Enumeration ae = null;
int count = 3;
while (count-- >0)
    {try {
        ae = hdisco.getLocalAdvertisements(DiscoveryService.GROUP,
                                           "Name","RestoNet");

        if ((ae !=null) )
            break;

        hdisco.getRemoteAdvertisements(null, DiscoveryService.GROUP,
                                       "Name","RestoNet",1,null)

        try {
            Thread.sleep(timeout)
        } catch (InterruptedException ie) { }
    }
continua .....
```

JXTA: RICERCA DI UN GRUPPO

```
if (ae == null)
    <creazione del nuovo gruppo>
else <richiesta di partecipazione al gruppo>
catch (Exception e)
    {System.exit(1);}
```

- *Osservazione:*

questo metodo di ricerca può risultare inefficiente, poiché il peer effettua **un'attesa attiva** delle risposte provenienti dalla rete.

Il programma si blocca nell'attesa della ricezione degli advertisements

Ricerca (Remota) Asincrona degli Advertisements: può essere effettuata associando un **DiscoveryListener** al servizio

RICERCA DI UN GRUPPO: METODI UTILIZZATI

`getLocalAdvertisements(int type, String attribute, String value)`

- *type* = indica il tipo di advertisement che stiamo cercando (DiscoveryService.PEER, DiscoveryService.GROUP, DiscoveryService.ADV)
- *attribute* = nome di un elemento presente nel documento XML che descrive l'advertisement
- *value* = valore dell'attributo ricercato
- *attribute* + *value* vengono utilizzati per restringere la ricerca. Possono essere utilizzate wildcard-

`getRemoteAdvertisements(peerid id, int type, String attribute, String value, int threshold, DiscoveryListener listener)`

- *id* può essere utilizzato per indirizzare la ricerca su un solo peer. Se null, la ricerca avviene su tutti i peers appartenenti al gruppo
- *threshold* indica il massimo numero di risposte che si intende ricevere
- *DiscoveryListener* indica il Listener a cui devono essere inviate le risposte. Nel caso sia null, la ricerca è sincrona