SERVICE-DRIVEN GROUP MANAGEMENT FOR MOBILE P2P SERVICES

A. Liotta[1], M. Ballette[1], L. Lin[1], M. Gasparoni[2], P. Brick[2], N. Papadoglou[2]
[1]ESE Department, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, United Kingdom. [2]Vodafone Group Services Ltd. Vodafone House, The Connection, Newbury Berkshire RG14 2FN, United Kingdom

Abstract: Given the considerable trend towards multi-party, peer-to-peer (P2P) communication, many are looking at the P2P computing paradigm as the means to extend the capability and scalability of Internet-based services. Current P2P frameworks are, however, largely incompatible with each other and do not address all the requirements of mobile computing. Here we propose a novel solution to peer group management which is 'autonomic', 'mobile friendly' and 'service driven'. We illustrate how our approach facilitates mobile P2P services by managing peer groups based on service semantics and resource availability. Our system allows efficient 'deep' search of user personal content stored in thin mobile terminals. Results are based on an experimental prototype and are demonstrated by a simple proof-of-concept mobile service.

Keywords: Mobility aware P2P; Mobile P2P applications; P2P Group Management.

1. INTRODUCTION

Mobile services over all-IP networks are revolutionizing the way people communicate, enhancing their ability to engage in virtual collaborations and share content. The classic client-server (C-S) interaction paradigm is still the backbone of Internet services. However, as these are increasingly based on multi-party, Peer-to-Peer (P2P) communication, P2P computing is gaining momentum as one of the fundamental building blocks of modern services.

Mobile P2P (MP2P) services bring the benefits of P2P computing (e.g., increased scalability, robustness, seamless resource pooling) to the mobile user. Despite their appeal, MP2P services are difficult to provide today since existing P2P frameworks are heavyweight and not geared towards mobility. On the other hand, mobile terminals are thin and involve intermittent connectivity that is not typical in conventional P2P computing.

Beyond the plethora of Internet services, mobile services represent an unprecedented revenue-raising opportunity for mobile network operators and value-added service providers. In particular, operators hold a unique position in the service-provisioning marketplace. They can rely on standardized service-centric frameworks (such as IMS[1]) and on a pre-existing range of subscribers. They have relevant know-how and are in the best position to add new dimensions to P2P services, namely security, authentication and quality of service.

Taking the operator's perspective, we present here a novel approach to deploying MP2P services which is 'autonomic', 'mobile friendly' and 'service driven'. In particular, we address the problem of providing effective group management in such a way which allows thin mobile terminals to dynamically join P2P services, share content and initiate virtual P2P communities.

Our approach is service-driven – peer groups or clusters are created on the basis of service semantics. We illustrate this new approach via a proof-of-concept P2P property selling application (Section 2), where peers are clustered depending on their postal code of interest. A completely different grouping strategy would be used, for instance, in a MP2P gaming service, where users may want to use group-joining strategies based on game name, user ability, location and so forth. Group structuring/organization is fundamental to P2P services. It affects the way P2P resources (including services) are, first, advertised and published and, then, discovered. The way adverts are organized, eventually affects the scalability of the MP2P service so we argue that the operator or service provider should directly affect that. This is why in our system grouping arises directly from service requirements.

Upon capturing the most relevant P2P systems (Section 3), we explain why our approach is mobility-aware (Section 4). Finally, in Section 5, we present the evaluation of our approach. Having realized a core set of MP2P protocols (publish, discover etc) and a MP2P service incurring a memory consumption of just 600Kbytes, we have proved the viability of our ideas.

2. MP2P SERVICE DEPLOYMENT SCENARIO

The application and use case scenario depicted in Fig. 1 have been developed for the purpose of illustrating the potential of mobile P2P services, demonstrating their viability in relation to state-of-the-art technologies, and present the service-driven group management approach proposed in Section 4. We are assuming a scenario in which mobile services are offered via a mobile network operator *e.g.* through the IP Multimedia System (IMS)[1], which supports the initial service discovery, registration, authentication and so forth. A typical sequence of interactions is depicted in Fig. 1. A user instantiates the service with the purpose of advertising a property. In addition to typical information (address, type of property, price etc) the user may use the terminal camera to grab/store pictures or video clips of the property. After entering the property data, an XML advert is generated (Fig. 1 step 1). This will be subsequently used during the semantic discovery phase (see Sect.4). Having stored the content and its corresponding XML advert, the user uses the IMS framework to authenticate itself and register to the property-selling service in the role of a "seller" (2).

A buyer joining the server will go through the usual operator's authentication process (3, 4), followed by a discovery phase. P2P publish and discovery are described in Section 4, while Fig. 1 shows that, once the buyer discovers the seller, any further communication between them is in P2P mode – i.e. the operator is bypassed for efficiency reasons (5). Being part of a P2P system, the buyer now transparently acts also as a server for any content it has downloaded from the seller (6). Other users may, then, instantiate the service either as buyers, sellers or both (steps 7-10).

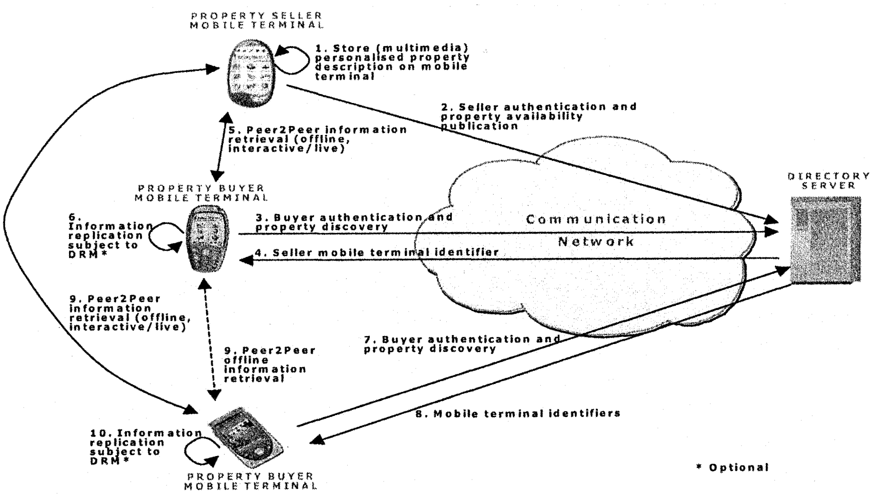

PROPERTY SELLER MOBILE TERMINAL
1. Store (multimedia) personalised property description on mobile terminal
2. Seller authentication and property availability publication
5. Peer2Peer information retrieval (offline, interactive/live)
PROPERTY BUYER MOBILE TERMINAL
DIRECTORY SERVER
6. Information replication subject to DRM*
3. Buyer authentication and property discovery
Communication
Network
4. Seller mobile terminal identifier
9. Peer2Peer information retrieval (offline, interactive/live)
9. Peer2Peer offline information retrieval
7. Buyer authentication and property discovery
8. Mobile terminal identifiers
10. Information replication subject to DRM*
PROPERTY BUYER MOBILE TERMINAL
* Optional

Figure 1. A personalized MP2P property selling application.

3. EXISTING P2P SYSTEMS

The Peer-to-Peer (P2P) computational paradigm[2] is in essence an alternative to the Client-Server (C-S) model. We can identify four main approaches to P2P computing. *Hybrid P2P systems* such as Napster[3], Jabber[4] and ICQ[5] represent the first generation of P2P systems. They use a centralized indexing server to facilitate the interaction between peers – but this introduces, again, a single point of failure and a bottleneck.

In a *pure unstructured P2P system* all nodes communicate and find each other directly through a P2P overlay *with no central server intervention.*

This solution is highly *scalable* and *reliable* since a failure in one or more nodes does not affect the communication between the remaining nodes. Examples are Gnutella[6] and Freenet[7]. The inherent drawbacks are the limited search horizon and the large network overhead (queries are propagated via flooding). Hence, unstructured systems have been superseded by *structured P2P systems* which are based on *Distributed Hash Tables* (DHT)[8]. The latter systematically reduces search complexity - examples are Chord[9], Pastry[10], and Tapestry[11]. What limits the applicability of DHT systems to the area of mobile services is their simplicity in terms of searching.

Super-Peer networks such as JXTA[12,13] consist on some special nodes that operate as a server to a set of clients and as an equal in a network of super-peers. These systems improve manageability without compromising *scalability* and *fault tolerance.* They combine the efficiency of centralized search with the autonomy, load balancing and robustness provided by distributed search. The key benefits exploited also in our system are the reduced search time and the limited signaling overhead. If we consider M Super Peers in charge of N peers (with $N>>M$), a search will take O(N) in a pure unstructured P2P overlay and O(M) in a Super Peer-based overlay.

4. GROUP MANAGEMENT IN PEERMOB

The work presented herein is part of the PeerMob project, whose aim is to build a service framework facilitating the deployment of mobile P2P services by network operators. Given the scope of this article, below we focus on the group management aspects of PeerMob.

4.1 Semantic Clustering for Service-Driven Grouping

PeerMob adopts a Super Peer, n-tier hybrid architecture where the number of hierarchical levels and the structure of the P2P overlay are determined by service semantics. We illustrate this novel approach by means of an example, using the application of Section 2. For the sake of simplicity, Fig. 2 depicts a possible solution based on a 3-level hierarchy. The main components are:

- *PeerMob centralized server*: holds the service schema which ultimately influences they way peers are dynamically grouped, acting as a meta-information repository of Super Peer (SP) identifiers;
- *Super Peers*: realize typical cluster head functionality, acting as information hubs (for discovery requests across groups), and store the meta-adverts corresponding to the whole content stored in their group peers.

- *Simple Peers (or peers)*: Store the actual data content (e.g. pictures, video clips, property description) to be shared with other peers.

Looking at the requirements and specification of the P2P application, the service developer designs the service schema. We have used XML to facilitate semantic data processing. For efficiency, our prototype makes use of a relational database. Fig.3 illustrates a snippet of the server-side meta-information w.r.t. the application of Section 2 and the scenario depicted in Fig.2. Fig.3A, illustrates the grouping based on postal code; Fig.3B, shows the three SP identifiers; and Fig.3C identifies the allocation of SPs to Groups (one SP per group in this simple case).

Service semantics influences also the structure of the meta-information maintained by Super Peers. Our SPs have a lightweight relational database analogous to the one present at the server side. However, while the server-side meta-information relates groups with SPs, the SP meta-information relates SPs with simple peers.

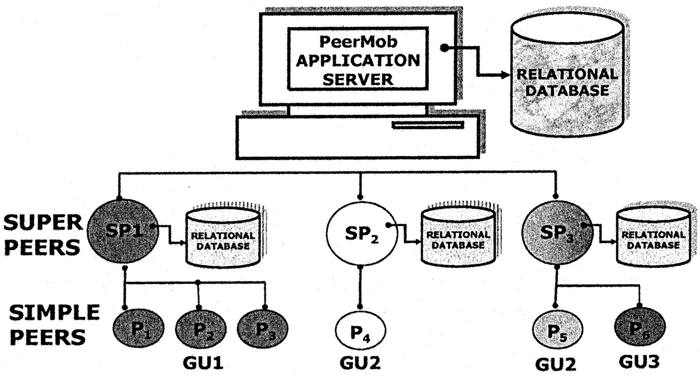


Figure 2. The PeerMob system in relation to the application of Section 2.

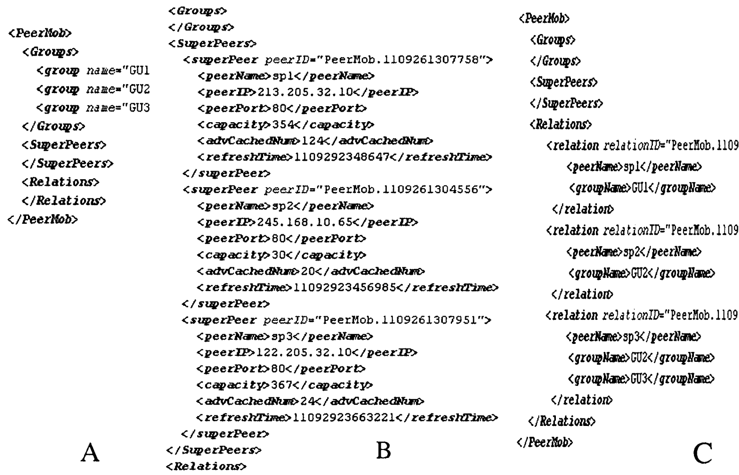
4.2 Mobility-aware Group Management

All meta-information is dynamically updated as new groups appear and new users join/disjoin those groups. Clearly, when the size of a group grows (for instance as a result of an increased interest in the properties of a particular area), the number of SPs per group should grow too. Since we are assuming a mobile P2P scenario, relatively thin terminals may act as SPs; so another important element of group management is the run-time level of congestion of SPs. Finally, since we are assuming that any peer (including SPs) may be mobile, the system should keep track of who is connected and react appropriately when peers lose connectivity. In particular, since SPs store the adverts relating to the data content shared by simple peers, it is essential to maintain replicas of those adverts. Similarly, the actual content stored by simple peers should be transparently replicated to increase availability and tolerance to loss of connectivity.

We have addressed the aforementioned requirements via an autonomic, self-management approach. This is based on periodic messaging that is confined between adjacent hierarchical levels. Going back to the example of Fig.2, peers periodically report their presence directly to their SP (or SPs to cater for data replication). In turn, SPs report their presence and level of resource utilization (the 'capacity' field in Fig.3B) directly to the centralized server.

In this way, when a simple peer loses connectivity from its group, the relevant SP can trigger a P2P fault-tolerance mechanism. For instance, if the disconnected peer contains data which is requested by some other peer, the SP dynamically redirects any request meant for the disconnected peer to the peer containing its replicas. Loss of connectivity of a SP is handled in a similar way but involves the intervention of the centralized server.

Group size is managed on the bases of the 'capacity' information relayed by SPs to the centralized server and of the 'peer to SP' ratio. So if a SP is hit by 'too many' requests, and as soon as its run-time resources go below a certain threshold, a new SP election mechanism is triggered under the control of the server. A similar reaction is triggered as the number of peers per group increases.



```
<PeerMob>
  <Groups>
    <group name="GU1
    <group name="GU2
    <group name="GU3
  </Groups>
  <SuperPeers>
  </SuperPeers>
  <Relations>
  </Relations>
</PeerMob>
```

A

```
<Groups>
</Groups>
<SuperPeers>
  <superPeer peerID="PeerMob.1109261307758">
    <peerName>sp1</peerName>
    <peerIP>213.205.32.10</peerIP>
    <peerPort>80</peerPort>
    <capacity>354</capacity>
    <advCachedNum>124</advCachedNum>
    <refreshTime>1109292348647</refreshTime>
  </superPeer>
  <superPeer peerID="PeerMob.1109261304556">
    <peerName>sp2</peerName>
    <peerIP>245.168.10.65</peerIP>
    <peerPort>80</peerPort>
    <capacity>30</capacity>
    <advCachedNum>20</advCachedNum>
    <refreshTime>11092923456985</refreshTime>
  </superPeer>
  <superPeer peerID="PeerMob.1109261307951">
    <peerName>sp3</peerName>
    <peerIP>122.205.32.10</peerIP>
    <peerPort>80</peerPort>
    <capacity>367</capacity>
    <advCachedNum>24</advCachedNum>
    <refreshTime>11092923663221</refreshTime>
  </superPeer>
</SuperPeers>
<Relations>
```

B

```
<PeerMob>
  <Groups>
  </Groups>
  <SuperPeers>
  </SuperPeers>
  <Relations>
    <relation relationID="PeerMob.1109261307951.GU1">
      <peerName>sp1</peerName>
      <groupName>GU1</groupName>
    </relation>
    <relation relationID="PeerMob.1109261307951.GU1">
      <peerName>sp2</peerName>
      <groupName>GU2</groupName>
    </relation>
    <relation relationID="PeerMob.1109261307951.GU1">
      <peerName>sp3</peerName>
      <groupName>GU2</groupName>
      <groupName>GU3</groupName>
    </relation>
  </Relations>
</PeerMob>
```

C

Figure 3. Server-side XML schema for the property selling application of Section 2.

4.3 Semantic Publishing and Discovery

P2P services critically rely on two functions, publish and discovery. Publishing, allows a peer to advertise its content, resources or services in such a way as to make them shareable with other peers. Discovery, in turn, allows a peer to find resources of interest. PeerMob adopts a semantic approach to publish and discovery. The first issue for a seller is to determine the group where its advert should be propagated. In our case, semantic indexing is achieved by mapping the seller's advert (in XML) with the server DB information. Given the semantics of this particular service, the server returns SPs which store adverts w.r.t. the relevant postal code. Multiple SPs are returned in order to allow the replication of adverts. At this point, the peer joins the relevant peer group and publishes the advert in SP_1 and SP_2. A similar semantic-based procedure is followed by the buyer. In this case, the initial task is to determine the relevant group that holds adverts regarding the geographic area of interest. Upon joining a group, the buyer can query the group's SP, issuing an XML query which is easily matched against the SPs list of adverts. The SP finally returns the list of sellers to the requester who can now engage in P2P communication with the buyers.

5. SYSTEM EVALUATION

We have realized a first prototype of the MP2P core along with the application of Section 2, proving the viability of the MP2P concept involving thin mobile terminals – the total footprint is only 600Kbyte. In the remainder we present a significant sample of results aiming at assessing response time, overheads, and scalability.

5.1 Super Peer Response Time

Super Peers store adverts and handle queries coming from group members or from other SPs. We are assuming a mobile environment in which wireless enabled PDAs can act as SP. We assess here the ability of such thin devices to act as SPs. The experimental set-up is illustrated in Fig. 4. The PDA is an HP IPAQ 5550 with a 400 MHz Intel XScal PXA255 processor, 128 Mbytes RAM, 48Mbytes ROM and integrated WiFi (802.11b). Multiple simple peers run on a single laptop. The PDA stores 1000 XML adverts in its cache and is hit by an increasing query rate ranging between 2 and 32 requests per second. Fig. 5 illustrates the results, including average values and the 95% confidence bands related to 100 repetitions. The important result is that response time does not increase dramatically, proving that even thin terminals can be effective SPs. Scalability is achieved thanks to the distributed search mechanism offered by P2P.

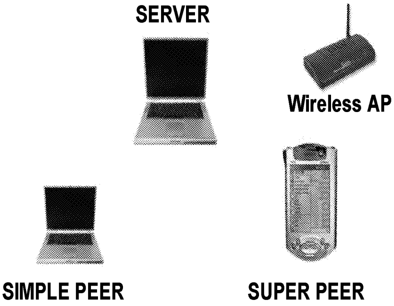


Figure 4. Experimental set-up.

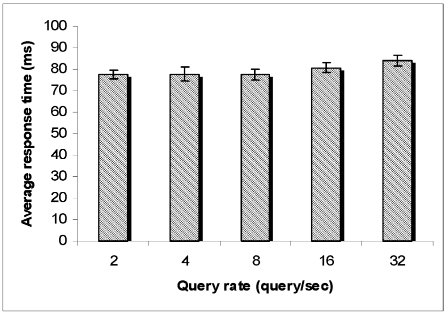


Figure 5. Super Peer response time.

5.2 System signaling

Like for any other distributed system, P2P systems offer advantages in terms of response time and scalability which are paid at the expenses of other forms of overheads. In PeerMob there is periodic messaging between adjacent hierarchical levels (Fig.2). The advantage of our approach is that messages generated by simple peers are confined within their group (i.e. do not propagate up to the server). There is, however, messaging between SPs and server – this is needed for effective group management (Section 4). Messaging around the server is the main determinant of bottlenecks – we report here our findings based on mathematical modeling (for brevity, the model description has been omitted).

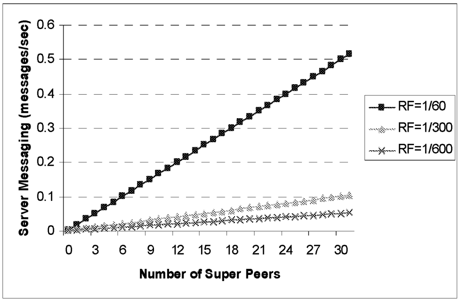


Figure 6. Server messaging Vs. SPs.

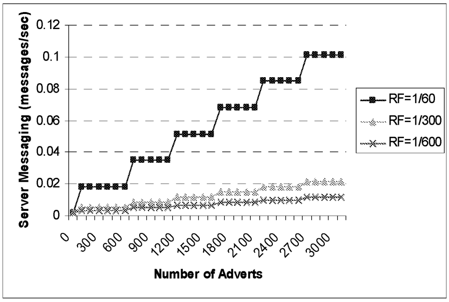


Figure 7. Server messaging Vs. adverts.

The linear behavior of Fig.6 was expected since the size of messages does not change with other system parameters, whereas message rate is determined by the self-management system. As the system becomes more dynamic, the message refresh rate (RF) has to increase and so will the slope of the line.

Fig.7 captures what happens as we increase the number of adverts. In this case the maximum SP capacity was set to 500 adverts. This means that while the number of adverts increases linearly, the number of SP increases only when its maximum capacity is reached (every 500 adverts) – hence the step-like shape (message rate increases only with the number of SPs).

Being resilient to peer failure, PeerMob adopts a messaging rate which varies with the Peer Failure Frequency (PFF). Fig. 8 captures the influence of this parameter.

5.3 Search Failure Rate

Fig.9 captures the system resilience to SP failure. The result highlights a very attractive property which is typical of P2P systems, i.e. their ability to increase resource availability via data replication mechanisms. In our study we emphasize the effect of SP failure probability since this is an important factor in mobility-aware P2P systems. We can see that our system can compensate to a great extent SP failure by increasing data replication.

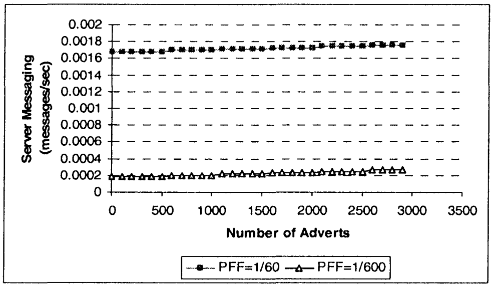


Figure 8. Effect of Peer Failure Frequency on server messaging.

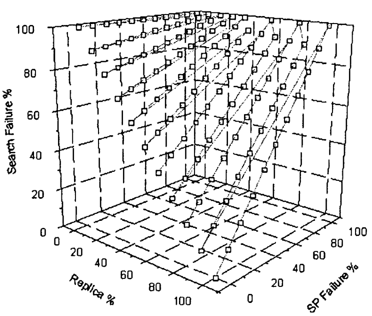


Figure 9. Resilience to SP failure.

6. CONCLUDING REMARKS

Behind the scenes of this article we are addressing the following questions: is it possible to satisfactorily bring the power of P2P to the mobile context? Whose role is it to provide the necessary level of management which can make this vision a reality? Can P2P-enabled mobile services spark a new generation of services?

The work carried out so far, indicates positive answers to the above questions and highlights the important role that operators may take.

Operators and service-providers already adopt a service-centric approach aimed at sparking a wealth of new mobile services. MP2P is certainly not the only enabler of revolutionary services but it can, indeed, play an important role if integrated with existing frameworks.

Our immediate plans are to integrate the PeerMob system in IMS and study the benefits that MP2P can bring into it, while also exploiting the standardized features of IMS in P2P. Security, authentication, and charging are certainly weak aspects of existing P2P frameworks that are also not geared towards mobility. A seamless integration of P2P computing with state-of-the-art service-centric frameworks such IMS seems the obvious step towards effective mobile P2P services.

ACKNOWLEDGEMENTS

The work described herein has carried out in the context of the PeerMob project sponsored by Vodafone Group Services Ltd., UK.

REFERENCES

1. M. Poikselka, et al., The IMS: IP Multimedia Concepts and Services in the Mobile Domain. John Wiley & Sons, 2004.
2. M.P. Singh, Peering at Peer-to-Peer Computing. IEEE Internet Comp., Vol. 5(1), 2001.
3. http://www.napster.com
4. http://www.jabber.org/
5. http://www.icq.com/
6. http://www.gnutella.com/
7. http://freenet.sourceforge.net/
8. Zhiyong Xu, Rui Min, Yiming Hu, Reducing Maintenance Overhead in DHT Based Peer-to-Peer Algorithms. Proc. of P2P'03.
9. I. Stoica, et al., Chord: A scalable peer-to-peer lookup service for internet applications. Proc. ACM SIGCOMM 2001.
10. A. Rowstron, P. Druschel, Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, November, 2001.
11. B. Zhao, J. Kubiatowicz, A. Joseph, *Tapestry: An Infrastructure for Fault-Resilient Wide-Area Location and Routing*. Report UCB/CSD-01-1141, U. C. Berkeley, 2001.
12. http://www.jxta.org/
13. S.M. Botros et al., Search in JXTA and Other Distributed Networks. Peer-to-Peer Computing, 2001.