# PIX-Grid: A Platform for P2P Photo Exchange[*]

Karl Aberer, Philippe Cudré-Mauroux, Anwitaman Datta, Manfred Hauswirth

Distributed Information Systems Laboratory
Ecole Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland

**Abstract.** The proliferation of digital camera devices (stand-alone or combined with cell phones), new protocols such as MMS and the desire of people to communicate and share their experience call for new systems to support these needs in new Internet-scale infrastructures. In this paper we outline our plans for an Internet-scale P2P system that enables users to share (globally or only within a group) and efficiently locate photographs based on semi-automatically provided meta-information (by the devices and the user).

## 1 A P2P photo exchange platform

P2P systems have been popular for sharing music and video information for a while. With the proliferation of advanced digital cameras and mobile phones that support MMS and may even be combined with a digital camera themselves, sharing digital photos may indeed become a very popular network application. The application scenario we envision is simple yet realistic: People take photos with their digital cameras which already associate some descriptive information (meta-information), for example, the date and time the photo was taken, the name of the user, or even some GPS information, that gives indication on where the photo was shot. The user may add further information, such as descriptive text and some keywords and then allow other people to access it.

This scenario may serve several purposes: Distributing photos to everyone or just offering access to a limited group, e.g., your family, or even provide a global photo database, if you were for example unable to take a photo at a certain place, or just efficiently discover photos of a place you would like to visit. A central requirement to put such scenarios into practice is that meta-information is provided to support enhanced, structured search. In fact this means that as much meta-information as possible should already be provided by the devices themselves and the user only needs to add further information of his choice (users do not want to spend much time—if at all—for providing meta-information).

So why is this scenario interesting? Actually it is interesting both from an application side and a technical side. People may really want to share files which would provide a sufficiently large user community to set up a system and do real-world evaluations.

[*] The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

The technical challenges that need to be tackled then are: How can we provide a system that supports a large number of users, that are online with only limited probability and may join and leave the system at any time, in sharing and discovering photo information easily. This involves providing structured meta-data-based search to efficiently discover interesting information and also must incorporate devices of limited computing power (cell phones, digital cameras) and new protocols (MMS). Additionally the question arises of how can we make the attached meta-information compatible at a schema level? For example, one user/device may use the notion of “place” while another one means exactly the same by tagging the respective information with “location”?

This basically reads like the requirements of a next generation self-organizing, decentralized P2P system with meta-data support. And that is exactly what we are aiming at in PIX-Grid.

Where are we now? At the moment the normal functionality supported by all decentralized P2P systems is to share, search, and access files based on filenames. Centralized systems such as Kazaa already support some kind of meta-data-based search but we will not further investigate this family of P2P systems because we think that the really challenging problems and interesting solutions can be found in the area of decentralized systems that exhibit self-organizing behaviours. Another argument against centralized systems would be that you cannot setup new infrastructures which require quite some investment into computing resources every time you come up with a new application scenario.

At the moment (with the exception of Gnutella) most of the systems in the class of decentralized P2P systems are based on distributed hash trees (DHT) and support exact searches for filenames. DHT-based systems can typically search only for exact information, but not substrings, or meta-information associated with files. Any reasonably sophisticated file-sharing system especially if the system should be usable by an unskilled user, however, ought to accommodate more complex searches based on substrings or meta-information. For example, the user may want to look for pictures of Lake Geneva taken during a sunny day in Lausanne. Further, many of the existing P2P systems do not provide proper authorization facilities which is essential to restrict access and build up user communities. It is more likely that a person will like to share some of his/her pictures only with friends and family rather than with everybody. Thus authorization supporting reasonable granularity is required and the authorization facility must work completely decentralized in an environment of very large user communities which imposes considerable requirements and leaves room for lots of research to be done.

Existing P2P systems also tend to focus on actively searching for files. The dual aspect, that of disseminating files to users who are interested in a certain type of information has not been addressed adequately in the large-scale environment we envision. We intend to include such capabilities through a publish/subscribe mechanism that also supports meta-data, i.e., the user can post a permanent query and the system will send all information to the user as soon as matching information becomes available.

While the research will focus on the problem areas described above we plan to validate our findings in a case study called PIX-Grid which will incrementally support the above functionalities. Such an application seems to have the potential of being an instant success, particularly given the recent advent of Multimedia Messaging Service (MMS) and the exponentially increasing proliferation of digital cameras.

In the following sections we outline the work we plan to do in PIX-Grid.

### 1.1 The P2P infrastructure: P-Grid

As we do not want to start from scratch we will base PIX-Grid on our previous work in the P2P area. We will use our P-Grid P2P system as PIX-Grid's underlying P2P infrastructure and enhance it with the missing functionalities. This section gives a brief overview of P-Grid.

P-Grid [2] is a peer-to-peer lookup system based on a virtual distributed search tree: Each peer only holds part of the overall tree, which comes into existence only through the cooperation of individual peers. Searching in P-Grid is efficient and fast even for unbalanced trees [1] ($O(\log(n))$, where $n$ is the number of leaves). Unlike many other peer-to-peer systems P-Grid is a truly decentralized system which does not require central coordination or knowledge. It is based purely on randomized algorithms and interactions. Also we assume peers to fail frequently and be online with a very low probability. Fig. 1 shows a simple P-Grid.

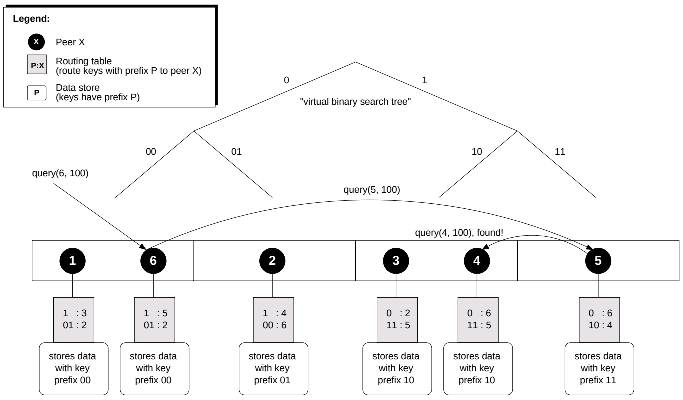

Legend:
X
Peer X
P:X
Routing table
(route keys with prefix P to peer X)
P
Data store
(keys have prefix P)
0
1
"virtual binary search tree"
00
01
10
11
query(6, 100)
query(5, 100)
query(4, 100), found!
1
6
2
3
4
5
1 : 3
01 : 2
1 : 5
01 : 2
1 : 4
00 : 6
0 : 2
11 : 5
0 : 6
11 : 5
0 : 6
10 : 4
stores data with key prefix 00
stores data with key prefix 00
stores data with key prefix 01
stores data with key prefix 10
stores data with key prefix 10
stores data with key prefix 11

**Fig. 1.** Example P-Grid

Every participating peer's position is determined by its path, that is, the binary bit string representing the subset of the tree's overall information that the peer is responsible for. For example, the path of Peer 4 in Fig. 1 is 10, so it stores all data items whose keys begin with 10. For fault-tolerance multiple peers can be responsible for the same path, for example, Peer 1 and Peer 6. P-Grid's query routing approach is simple but efficient: For each bit in its path, a peer stores a reference to at least one other peer that is responsible for the other side of the binary tree at that level. Thus, if a peer receives a binary query string it cannot satisfy, it must forward the query to a peer that is "closer" to the result. In Fig. 1, Peer 1 forwards queries starting with 1 to Peer 3, which is in Peer 1's routing table and whose path starts with 1. Peer 3 can either satisfy the query or forward it to another peer, depending on the next bits of the query. If Peer 1 gets a query starting with 0, and the next bit of the query is also 0, it is responsible for the query. If the next bit is 1, however, Peer 1 will check its routing table and forward the query to Peer 2, whose path starts with 01.

The P-Grid construction algorithm [2] guarantees that the routing tables always provide at least one path from any peer receiving a request to one of the peers holding a replica so that any query can be satisfied regardless of the peer queried. P-Grid as the underlying infrastructure already takes into account low online probabilities of the participating peers. Thus each peer has multiple routing entries for each level of the P-Grid tree which enables successful routing even in the presence of many unavailable peers. Other problems to be addressed in the environment targeted by PIX-Grid are disconnected operations and mobility of the peers. Also this problem is already addressed by P-Grid's functionalities as described in [8]. The basic idea of the approach is that each peer inserts a (replicated) mapping of a unique identifier to its current location (IP address) and updates this mapping every time it changes location or becomes online again. The routing process thus exploits the unique identifiers instead of the changing IP addresses. The algorithms used for that are secure and efficient (proven analytically and by simulations). So routing is successful with a very high probability even if many peers are offline or change location. For content (photos) the situation is alike. However, it can occur that the location can be found in the routing process but the peer holding the data is offline and possibly none of the peers who have already retrieved the data and thus replicate it is available. In this case the data location can be found but retrieval is not possible. This cannot be avoided but is not a severe restriction.

## 2 PIX-Grid

This section describe the functionalities to be included in PIX-Grid and how we plan to accomplish them.

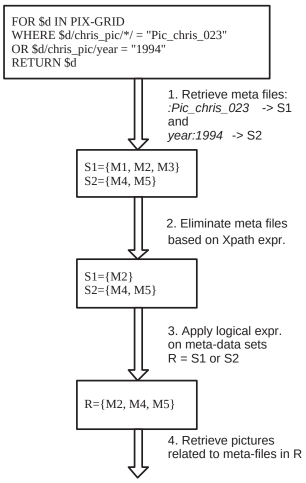
### 2.1 Powerful search functionality

P2P technologies have been mainly used for building general purpose file-sharing systems so far, limiting their capability to answer complex queries or to support user-defined schemas. Some centralized or hierarchical systems (e.g., Kazaa) define a rigid schema for file annotations, imposing the meta-data the users have to use. Completely decentralized or DHT-based systems presently do not provide any annotation facility, supporting either exact matches or substring searches based on file names (so far only P-Grid and CAN support substring search). This will definitely be a hindrance for the development of any reasonable application, and hence we will explore the indexing and searching of meta-information (not conforming to a fixed schema but rather based on custom XML documents) in order to locate the associated resources (pictures).

As a starting point this may be done in a brute force manner by treating the content of the meta-data file associated with the document as pure text and then index each of its sub-structures separately. Using standard text-retrieval techniques we could thus offer keyword-search capabilities at the expense of the structure of the meta-data information which is completely lost in the process. Instead, we choose to index the meta-data files themselves (pointing to the picture they are related to) using series of *attribute-name:value* expressions as keys in P-Grid. This mean that we take every attribute in the XML file and index this file using the concatenation of the attribute name and its value as key (additionally, we perform a suffix closure operation on the keys in order to allow for substring searches in P-Grid). Combining this with some post-processing on the retrieved meta-data files enables us to treat simple XQuery-like expressions as follows:

1. extract the different *attribute name:value* pairs from the XPath expressions in the query, use them as keys and retrieve the corresponding meta-data files from P-Grid
2. inspect the meta-data files retrieved and eliminate those not strictly complying to the aforementioned XPath expressions
3. create sets of meta-data files by grouping the files by the XPath expression used to retrieve them; apply on these sets the same logical expressions as applied in the query for their corresponding XPath expressions
4. look into the remaining meta-data files to get the names of the documents to be retrieved; retrieve those documents from the P-Grid.

Figure 2 gives a simple example of this process.



```
FOR $d IN PIX-GRID
WHERE $d/chris_pic/*/ = "Pic_chris_023"
OR $d/chris_pic/year = "1994"
RETURN $d
```

**Fig. 2.** Meta-data search

### 2.2 Inter-operability

Built on top of existing infrastructures PIX-Grid has been designed to integrate seamlessly into today's state-of-the-art IT environment. Figure 3 depicts the process of picture insertion. On the left, the user sends an MMS (*Multimedia Messaging Service*) message encapsulating the picture itself and some related meta-data (partly defined by the user, partly automatically generated by the device) in XML. The message is received by the PIX-Grid component which takes care of inserting the meta-data and the picture into two separate P-Grid infrastructures. The P-Grids in turn take care of contacting the different peers in order to store the assets accordingly. This process is depicted in Fig. 3.

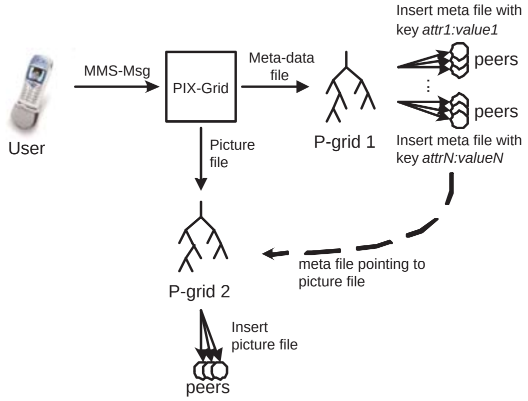

User
MMS-Msg
PIX-Grid
Meta-data file
P-grid 1
Insert meta file with key *attr1:value1*
peers
peers
Insert meta file with key *attrN:valueN*
Picture file
P-grid 2
meta file pointing to picture file
Insert picture file
peers

**Fig. 3.** Picture insertion

Similarly, requesting a picture would first require the user to create an SMS or HTTP XQuery message (using presumably some user-friendly tools) and to send it to the PIX-Grid component. PIX-Grid would then start querying the first P-Grid and retrieve the picture from the second P-Grid as explained above. Finally, the picture is sent back to the user using, again, some standard protocol or messaging system.

### 2.3 Authorization and access

File sharing is arguably the most successful P2P application yet and it already shows various challenges that need to be met including issues like copyright protection, privacy and free riders. Additionally, users may want to provide their resources (photos) only to a defined group. In order to address these issues a decentralized authorization and access scheme is required that provides means for simple definition of access policies that need to be enforced by the system.

We have already proposed a decentralized public key infrastructure (PKI) [5] that may be used as the basis for any authorization scheme. At this early stage it seems that an approach like that of SDSI [9], which is similar to DNS-like delegation, but with multiple roots, such that any peer can authorize a subspace to be used by any other peer, and may also authorize them to authorize the same space to others will meet the basic requirements of authorization at an agreeable granularity.

### 2.4 Information dissemination

P2P systems support only active searching. However, it is desirable to have the dual aspect of searching, that of disseminating the relevant information to interested peers automatically. Presently there are several paradigms for dissemination in information systems [7], for instance, request/response, polling, broadcast disks, publish/subscribe (P/S), etc. While some of these have strictly client-server or other asymmetric centralized architectures and cannot be used in ad-hoc networks, others, particularly the P/S paradigm may be realized for autonomous decentralized event producers (the offering peers) and event consumers (the peers looking for information). For example, Scribe [3] is a proposal for realizing topic-based P/S in a P2P system. Actually in a application like PIX-Grid many of the constraints of fully-blown P/S systems can be relaxed. Ordering, consistency, and completeness are of much lesser importance which in turn will simplify the implementation.

A tentative way to implement content-based P/S in a P2P system will involve indexing of queries (subscriptions), and then, whenever any content is published, the indexed queries may be used to push the content to the subscribers. Peers coming online will have to conduct pulls. At present it is difficult to see how event ordering may be achieved, but probabilistic consistency and eventual completeness seems feasible using a lazy push-pull updating scheme [6]. This will also depend on the advance we make in meta-information indexing (described above), since it will be necessary for content-based publish/subscribe.

### 2.5 Handling hot-spots

P-Grid provides a distributed and persistent storage facility, but does not directly allow for wider dissemination of popular content. This is important in order to minimize the service time of so-called *flash-crowds*, i.e., large groups of users requesting an asset approximatively at the same time. We have already devised a method for handling flash-crowds and replicating popular content (here, popular pictures) in a completely decentralized manner [4].

Basically, it is possible to disseminate the assets proportionally to their (estimated) popularity in a greedy but transparent fashion, i.e., without modifying the routing tables of the underlying structure, thus balancing the overall effort and minimizing the service time for the clients. Such techniques would fit very well here, since our application is targeted for ubiquitous computing where bandwidth is scarce and where users behavior can be very dynamic.

## 3 PIX-Grid Architecture

So far the intended architecture is not fully specified. Fig. 4 provides a rough sketch of the architecture we envision.

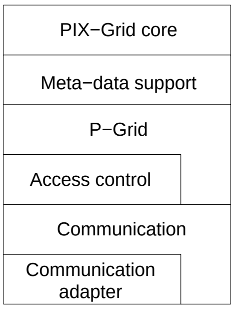


**Fig. 4.** PIX-Grid's architecture

For simplification Fig. 4 abstracts from the distribution issues and only shows the functional architecture. Actually each of the components is distributed over the peers constituting the overall system. The communication component provides the transport services required by the higher layers via a uniform interface. The transport services work in cooperation with the communication adapter that translates back and forth between various special protocols such as MMS. In fact this component consists of a simple framework with some standard conversion services like the before-mentioned MMS but can be extended easily with further adapters. On top of the communication facilities the access control component is in charge of enforcing the user-defined access control policy.

The functional core of the system consists of the upper three layers: The PIX-Grid core is the heart of the system and coordinates and processes all requests. Since we want to uses meta-data all this is based on the services of the meta-data component which in turn relies on P-Grid to accomplish its tasks. P-Grid itself operates on top of the access control component for external communication (user requests) and on the plain communication layer if intra-system issues are concerned.

## 4 Conclusion and summary

Our envisioned PIX-Grid system provides an interesting application and testbed to study large-scale information dissemination, meta-data-based search and various kinds of problems in decentralized systems such as authorization and access. We plan to base PIX-Grid on our P-Grid P2P system and will add the higher-level functionalities such as meta-data support and access schemes. The application domain (sharing photos) seems to be interesting for the average user so that we will have a fair chance of seeing the system put in place at a large scale which will give us the opportunity to do large-scale testing and statistics to discover further research issues and address them.

## References

1. Karl Aberer. Scalable data access in P2P systems using unbalanced search trees. In *WDAS*, 2002.
2. Karl Aberer, Manfred Hauswirth, Magdalena Punceva, and Roman Schmidt. Improving data access in P2P systems. *IEEE Internet Computing*, 6(1), Jan./Feb. 2002.
3. M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron. SCRIBE: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in communications (JSAC)*, 2002.
4. Philippe Cudre-Mauroux and Karl Aberer. A decentralized architecture for adaptive media dissemination. In *International Conference On Multimedia And Expo, ICME2002*, 2002.
5. Anwitaman Datta, Manfred Hauswirth, and Karl Aberer. Beyond "web of trust": enabling p2p e-commerce. Technical Report IC-2003-06, EPFL, 2003.
6. Anwitaman Datta, Manfred Hauswirth, and Karl Aberer. Updates in highly unreliable, replicated peer-to-peer systems. In *To appear in proceedings of the 23rd International Conference on Distributed Computing Systems, ICDCS2003*, 2003.
7. Michael J. Franklin and Stanley B. Zdonik. Dissemination-based information systems. *Data Engineering Bulletin*, 19(3):20–30, 1996.
8. Manfred Hauswirth, Anwitaman Datta, and Karl Aberer. Handling Identity in Peer-to-Peer Systems. Technical Report IC-2002-67, EPFL, 2002.
9. Ronald L. Rivest and Butler Lampson. SDSI – A simple distributed security infrastructure. CRYPTO'96 Rumpsession, 1996.