iClouds – Peer-to-Peer Information Sharing in Mobile Environments

Andreas Heinemann*, Jussi Kangasharju, Fernando Lyardet, and Max Mühlhäuser

Telecooperation Group, Department of Computer Science
Darmstadt University of Technology
Alexanderstr. 6, 64283 Darmstadt, Germany
aheine@gkec.tu-darmstadt.de
{jussi, fernando, max}@tk.informatik.tu-darmstadt.de

Abstract. The future mobile and ubiquitous computing world will need new forms of information sharing and collaboration between people. In this paper we present *iClouds*, an architecture for spontaneous mobile user interaction, collaboration, and transparent data exchange. iClouds relies on wireless ad hoc peer-to-peer communications. We present the iClouds architecture and different communication models, which closely resemble familiar communication forms in the real world. We also design a hierarchical information structure for storing the information in iClouds. We present our prototype implementation of iClouds which runs on wireless-enabled PDAs.

1 Introduction

People living in the future ubiquitous computing world will need new ways to share information and interests as well as collaborate with each other. Given the success of wireless communications, such as mobile telephones, 802.11b, or Bluetooth, many of these activities will benefit from or rely on wireless ad hoc communications.

This paper presents the iClouds project in which we study the architectures and mechanisms required to support mobile user interaction, collaboration, and transparent data exchange. The iClouds project is part of the Mundo research activity, which we will present in the next section.

Our motivation behind iClouds can be expressed as follows: "Whenever there is a group of people, they may share a common goal or have a related motivation. Information of interest may be in possession of only a few of them." The goal of iClouds is to make this information available to the whole group, based on individual user contribution, through peer-to-peer communications and data exchange.

* The author's work was supported by the German National Science Foundation (DFG) as part of the PhD program "Enabling Technologies for Electronic Commerce" at Darmstadt University of Technology.

H. Kosch, L. Böszörményi, H. Hellwagner (Eds.): Euro-Par 2003, LNCS 2790, pp. 1038–1045, 2003.

(a) Communication horizon (b) iClouds with 3 peers

Fig. 1. Information clouds

Consider a person walking with a wireless-enabled PDA. The communication range of the wireless device defines a sphere around that node. We call this sphere or communication horizon an *information cloud* or *iCloud* (see Fig. 1(a)). In practice this will not be an ideal sphere due to radio signal interference with buildings and other structures. The limited communication range (a few hundred meters at most) is a desired property, because it allows for easy ad hoc meetings and collaboration. When several nodes come close together, as shown in Fig. 1(b), the devices can communicate with each other and exchange information depending on what information the users provide and need. This exchange happens automatically, without any need for direct user intervention.

We have identified several application scenarios in which iClouds is beneficial:

– **Local Information Acquisition.** Residents of a city publish information about their city which tourists are interested in. This could be information about sights, restaurants, or useful telephone numbers, such as taxi number, etc.
– **Common Goal Pursuit.** iClouds can bring people with common interests together to help them collaborate. For example, consider students in a classroom. Some students may have formed study groups and others might be interested in joining those groups. Students already in the groups could publish the groups and interested students could directly contact them and join the group.
– **Advertisement and mCommerce.** A store can publish ads which are picked up by interested customers. These customers further pass the ads along to other interested users when they are away from the store, thus increasing the reach of the ads. If any of the users who have received ads in this way actually make a purchase, the store could give a bonus to the person who passed the ad along. This bonus could be, for example, points or a discount on the next purchase.

This paper is organized as follows. Section 2 describes the underlying Mundo project. Section 3 presents the iClouds architecture and communication mechanisms. In Section 4 we describe our prototype implementation of iClouds. Section 5 discusses related work. Finally, Section 6 concludes the paper and presents directions for future work.

2 Mundo Overview

In this section, we provide a brief overview of the Mundo project. A more complete description can be found in [1]; below we will briefly present the different entities in Mundo.

ME (Minimal Entity)

We consider ME devices as the representation of their users in the digital world. The personal ME is the only entity always involved in the user's activities. Our decision for one individually owned device rather than a publicly available infrastructure comes from *trust establishment*. As an increasing amount of computer based activities have (potential) legal impact and involve privacy issues, it is vital that users can trust the device carrying out such transactions.

US (Ubiquitous aSsociable object)

Minimalization pressure will not permit feature-rich MEs in the near future. Hence, they must be able to connect to local devices such as memory, processors, displays, and network devices in an ad hoc manner. We call this process association and we call such devices *ubiquitous associable objects* (US). Through association, the ME can personalize the US to suit the user's preferences and needs. For privacy reasons, any personalization of an US must become unavailable if it is out of range of the user's ME device.

IT (smart ITem)

There are also numerous smart items that do not support association that would turn them into an US. Vending machines, goods equipped with radio frequency IDs, and landmarks with "what is" functionality are just a few examples. We define such *smart items* as ITs. An IT is any digital or real entity that has an identity and can communicate with the ME. Communication may be active or passive; memory and computation capabilities are optional.

WE (Wireless group Environment)

We expect ad hoc networking to be restricted to an area near to the user of a ME device, as connections with remote services will involve a non-ad hoc network infrastructure. The functionality of a WE is to bring together two or more personal environments consisting of a ME and arbitrary US entities each. A WE makes connections between the devices possible and also allows for sharing and transferring hardware (e.g., US devices) and software or data between WE users.

iClouds is our most advanced project (in terms of realization) within the WE sub-domain of Mundo. While iClouds focuses on pure information sharing, other WE-related activities consider hardware sharing and corresponding security aspects.

THEY (Telecooperative Hierarchical ovErlaY)

We regard overlay cells as the backbone infrastructure of Mundo. These *telecooperative hierarchical overlay cells* (THEY) connect users to the (non-local) world, deliver services, and data to the user. THEY support transparent data access, for example, by caching frequently used data on US devices, and offer transparent cooperation between different physical networks. Because iClouds is based on ad hoc networking, it does not require the services of THEY.

3 iClouds System Description

iClouds devices are small mobile devices (like PDAs) with mobile communication support for a maximum of a few 100 meters; one example is a PDA with 802.11b support. There is no need for any central servers in the iClouds architecture; instead, each device is completely independent.

The diameter of the iClouds communication horizon (Fig. 1(a)) should not exceed a few hundred meters. We want to give iCloud users the option for spontaneous collaboration and when two iCloud users "see" each other, they should be within a short walking distance from each other (a couple of minutes at maximum). To allow for this easy collaboration, we specifically exclude multi-hop communication. Therefore, iClouds does not require any routing protocols; all communications happen directly between the concerned parties.

3.1 Data Structures, Communication Pattern, and Information Exchange

The two most important data structures found on the iClouds device are two information lists (*iLists* for short):

- *iHave*-list (information have list or information goods)
  The iHave list holds all the information the user wants to contribute to the iCloud. The items could be, for example, simple strings, or more complex entities expressed in XML. We discuss these issues in Section 3.2.
- *iWish*-list (information wish list or information needs)
  In the iWish list, the user specifies what kind of information he is interested in. The exact format of the items on this list depend on the format of the entries on the iHave list. Typically they would be search patterns which are matched against the entries on the iHave lists of other users. Note that the items on the iWish list are more private, since they reflect the needs of the user, which the user may not want to disclose to others.

Each iClouds device periodically scans its vicinity to see if known nodes are still active and in communication range and also to see if any new nodes have appeared. Information about active nodes is stored in a data structure called *neighborhood*.

In the second stage, the iClouds devices align their information goods and needs. This is achieved by exchanging iLists. Items on the *iWish*-lists are matched against items on the *iHave*-lists. On a match, information items move from one iHave-list to the other.

Table 1. Information Flow Semantics (from Alice's point of view)

	pull (from Bob)	push (to Bob)
iHave-List	Standard search	Advertise
iWish-List	Active service inquiry	Active search

For example, consider two iClouds users, Alice and Bob, who meet on the street. When their iClouds devices discover each other, they will exchange their iHave lists and match them locally against their iWish lists. If an item on Bob's iHave list matches an item on Alice's iWish list, her iClouds device will transfer that item onto her iHave list.

We have two main communication methods for transferring the iLists. Peers can either *pull* the iLists from other peers or they can *push* their own iLists to peers they encounter. In addition, either of these two operations is applicable to both lists, which gives us four distinct possibilities of communication. We summarize these possibilities, along with their real-world equivalents, in Table 1.

In each of the four cases shown in Table 1, the matching operation is always performed on the peer who receives the list (Alice's in pull and Bob's in push). Each of the four possible combinations corresponds to some interaction in the real world:

– *Standard search.* Alice pulls iHave-List from Bob.
  This is the most natural communication pattern. Alice asks for the information stored on Bob's device and performs a match against her information needs (specified in her iWish-List) on her device. We can also see the user as just passively "browsing" what is available.
– *Advertise.* Alice pushes her iHave-List to Bob.
  This is a more direct approach. Alice gives her information goods straight to Bob and it's up to Bob to match this against the things he is interested in. As an example, consider iClouds devices mounted on shopping mall doorways pushing advertisements onto customer devices when they enter the building.
– *Active service inquiry.* Alice pulls iWish-List from Bob.
  This is best suited for shopping clerks. They learn at a very early stage, what their customers are interested. An example of this query could be: "*Can I help you, please show me what are you looking for?*".
  In general, especially for privacy reasons and user acceptance, we believe it is a good design choice to leave the iWish-list on the iClouds device. Hence, this model of communication would likely be extremely rare in the real world.
– *Active search.* Alice pushes her iWish-List to Bob.
  With active search, we model the natural "*I'm looking for X. Can you help me?*". This is similar to the standard search mechanism, except that the user is actively searching for a particular item, whereas in the standard search the user is more passive.

3.2 Information Modeling and List Matching

A key issue in iClouds is a set of rules or constraints for the information items on the iHave-list, as well as for the search patterns stored on the iWish-list.

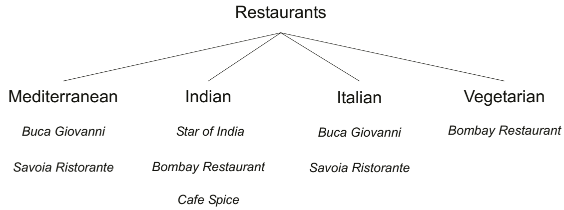

Fig. 2. Hierarchical data organization

For example, consider free text for both lists. A hypothetical item *Best Italian Restaurant on Market Street* will not match an information wish *Interested in Mediterranean Food*. Therefore the chances are close to zero, that any information is passed through iClouds, although one node holds information the other node is interested in.

To overcome this, we propose a hierarchical system, as shown in Fig. 2, similar to the organization of the Usenet, the product catalog of eBay, or the online catalog of Yahoo. Items are divided into categories (e.g., restaurants, hotels, etc.) which are further divided into subcategories (e.g., Italian, Indian, etc. restaurants). Note that entries may belong to several categories. For example, the Bombay Restaurant in Fig. 2 is listed under Indian and Vegetarian, because it offers both Indian and vegetarian food.

These hierarchies can be stored at a central server and can be downloaded to an iClouds device while it is in its cradle during power recharge or a normal sync operation. This system should be open for extension and categories could be moderated by different people to improve quality.

Hierarchies are well understood by users and, in addition, technologies, such as XML and XPATH, support the construction of hierarchically organized information and search very well. The names of the most common tags, such as address or telephone, will need to be standardized. We will also need a standardized way for user extensions to the hierarchy. These are topics of our on-going work.

3.3 User Notification and Privacy

Collaboration regarding a common goal or a related motivation requires that users are notified by the iClouds device on a successful match. This can be a simple beep or a vibrating alarm as found in mobile phones today. For each item (search pattern) on the iWish-list, the user can specify, whether a notification should be sent or not. This enables a user to differentiate between pure collect pattern, e.g. *Good Restaurants?*, and patterns that require some kind of action, e.g. talk to the other person.

iClouds devices are linked to their owners, broadcast information, and are traceable, hence they raise the question of user privacy. To protect user privacy, iWish-lists never leave the device, unless explicitly allowed by the user. Therefore, to construct a user profile is not possible. A user can mark each item on the iHave-list as private. Private items will then be unavailable to other parties. In addition, the comparatively short communication range constitutes a natural protection for user privacy.

4 Prototype

To gain more practical experiences with iClouds, we have built a first prototype and set up a testbed. The prototype runs on Toshiba Pocket PC e740 (Windows CE) with integrated 802.11b cards. For the underlying link layer network connectivity, we run the PDAs in 802.11b ad hoc mode with statically assigned IP addresses The prototype was developed using PersonalJava from the Java2 Micro Edition (J2ME).

Our information list data structures consist of strings. Currently, we have not yet implemented hierarchies. iList comparison is based on a simple substring matching function. A successful match will copy iHave-list items to new devices. The user is notified of this event by a beep from the device. This allows the user to check her updated information goods and plan further action.

We use a UDP based ping/pong mechanism for scanning the vicinity for new nodes. New node discovery is done by periodically pinging every IP address within a given set. A node sends a ping message to other nodes in and waits for a pong message. Upon receiving a pong, the new node is added to the active neighborhood. Otherwise, after a certain timeout, the node is removed from the neighborhood. When a node encounters a new node, it pulls the iHave list from the new node using an HTTP GET request.

Because a PDA is not a good device for data input, we believe that the iLists should be managed in a desktop application and the lists should be synced to the PDA during the normal sync operation.

5 Related Work

The Proem Platform [3] targets very similar goals. The main difference to iClouds is that they focus on Personal Area Networks (PAN) for collaboration. We believe that it is fruitful to focus on a wider area (mobile networks that cover several 100 meters in diameter) and that it is not necessary to encounter communication partners physically for information exchange.

Sharing information among mobile, wireless users is also subject of the 7DS Architecture [5,6]. In contrast to iClouds, in 7DS the users are intermittently connected to the Internet and cache information, i.e., HTML pages, which is accessed during that time frame. Later these caches are used to fulfill requests among the nodes.

The Usenet-on-the-fly system [2] makes use of channels to share information in a mobile environment. But the information spreading is limited by a hop count in the message. This has the disadvantage, that an unlucky user might be one hop to far away from the information source, although she might be interested in receiving the information.

Mascolo *et al.* describe XMIDDLE [4], a peer-to-peer middleware that allows the synchronization of XML documents across several mobile devices of one user. The data modeling and exchange in iClouds has to fulfill similar tasks, it has to work between different users though.

Basic information services require contributions from users. This is true for many current systems. The Usenet news is certainly one of the most prominent and successful systems. Tveit [7] proposes a peer-to-peer based network of agents, that support product and service recommendations for mobile users. Recommendations are provided by aggregating and filtering individual user input. Tveit focuses on infrastructured wireless networks, e.g. mobile phone networks.

6 Conclusion

In this paper we have presented iClouds, an architecture for supporting spontaneous user interaction, collaboration, and transparent data exchange in a mobile ubiquitous computing context. iClouds achieves these goals through ad hoc peer-to-peer communications between wireless-enabled devices. iClouds supports many natural forms of interaction, such as browsing for information, searching, and advertising. We have devised a hierarchical information structure for storing and matching information in iClouds. We have also implemented a prototype of iClouds which runs on PDAs with wireless LAN cards.

As part of our future work, we will develop a model for specifying the iClouds information hierarchies and ways to extend them. In addition, we plan on investigating ontologies as a possible way to improve the matching between items on the iLists.

References

1. Hartl, A. *et al.*, Engineering Multimedia-Aware Personalized Ubiquitous Services. In IEEE MSE 2002, Newport Beach, CA, December 2002.
2. Becker, C., Bauer, M., Hähner, J., Usenet-on-the-fly - Supporting Locality of Information in Spontaneous Networking Environments, Workshop on Ad Hoc Communications and Collaboration in Ubiquitous Computing Environments (at CSCW), New Orleans, LA, November 2002.
3. Kortuem, G. *et al.*, When Peer-to-Peer comes Face-to-Face: Collaborative Peer-to-Peer Computing in Mobile Ad-hoc Networks, International Conference on Peer-to-Peer Computing, Linköping, Sweden, August 2001.
4. Mascolo, C., Capra, L., Emmerich, W.: An XML-based Middleware for Peer-to-Peer Computing. International Conference on Peer-to-Peer Computing, Linköping. Sweden, August 2001.
5. Papadopouli, M., Schulzrinne, H., Seven degrees of separation in mobile ad hoc networks, IEEE GLOBECOM, 2000.
6. Papadopouli, M., Schulzrinne, H., Design and implementation of a peer-to-peer data dissemination and prefetching tool for mobile users, First NY Metro Area Networking Workshop, 2001.
7. Tveit, A., Peer-to-peer based Recommendations for Mobile Commerce, First International Workshop on Mobile Commerce, Rome, Italy, 2001.