The iBICOOP middleware: Enablers and Services for Emerging Pervasive Computing Environments

Amel Bennaceur, Singh Pushpendra, Pierre-Guillaume Raverdy, Valérie Issarny

The iBICOOP middleware: Enablers and Services for Emerging Pervasive Computing Environments

Amel Bennaceur, Pushpendra Singh, Pierre-Guillaume Raverdy, Valérie Issarny
ARLES Project-Team, INRIA Paris-Rocquencourt
Domain de Voluceau, Rocquencourt, Le Chesnay 78153, France
Email: firstname.lastname@inria.fr

***Abstract*—Content sharing and social networking are two activities that users now routinely engage in on a range of connected devices. However users have to face arcane configuration panels for setting up connectivity, heterogeneity in the human-machine interaction models, differences in devices' capabilities and a profusion of tools/services/applications to perform their tasks. Adding to this complexity is the need to interact with people not known a priory. To support users in such environments, we currently develop the iBICOOP middleware to minimize user effort in collaborating with other users and also provide seamless access to user data across different networks and devices.**

The iBICOOP middleware provides core services for distributed content storage, partnerships, naming & discovery and multi-network communications management. More complex services, such as data synchronization or content sharing, are then built on top of these core functionalities. iBICOOP is aimed to be deployed on home/enterprise servers, mobile devices and also in the 3G IMS infrastructure in order to truly leverage both Telecoms networks and the Internet. Central to our development is the use of standards in order to achieve interoperability with similar proprietary/closed systems.

I. Introduction

Nowadays, users rely on various sets of connected devices and services to interact with each other: home gateways, laptops, smartphones, UMPC, enterprise servers, Web/cloud computing services... While for a long time interactions have been mostly Web-centric, users now routinely engage in user-centric interactions such as content sharing and social networking through mobile devices. Indeed, these devices have grown very powerful in terms of computational, visual, and storage capabilities. They are able to retrieve, display and manipulate content or information easily. They also offer compelling advantages with the realization of the fixed/mobile convergence that enables anywhere/anytime IP connectivity to all terminals with extended networking capabilities (e.g., 3G/HSPDA, Wi-Fi, Bluetooth). Cellular and IT networks can be seamlessly used by mobile phones, as well as laptops or mobile multimedia players that are now fitted with mobile phone connectivity. Multiple devices with multiple communication interfaces give users more than one way to access content and interact depending on their context and needs.

While mobile devices have become very capable of communicating with other devices, accessing or exchanging content is far from being easy: communication costs are still high, communication protocols are not uniformly supported, communication links are not always reliable... Another problem lies in the role usually assigned to mobile devices. Although mobile devices are the primary terminals for interaction, they have essentially focused on (i) acting as clients accessing services in the infrastructure (e.g., enablers in the IMS infrastructure for Telecoms networks, or Web services in the Internet), or (ii) manipulating content stored locally (e.g., multimedia content playback). However, there has been little success in truly integrating these mobile devices in the pervasive computing environment (i.e., acting also as resource or service providers)[1], [2].

Furthermore, the use of multiple devices, having large storage capabilities, has resulted in the scattering of content on a set of devices, which, ironically, is making access to all the content that one possesses a big challenge (where is this file? Is it the latest version?). This problem becomes acute when it comes to impromptu collaboration (e.g., in business meetings, conferences, or on the road). Such collaborations often require exchange of content, and have to wait till the user gets physical access to the device over which the required content (or the correct/latest version) is currently stored. Delay in accessing the content often cause frustration and missed opportunities.

Besides research community, recently a number of industry initiatives have also been announced that try to address these issues (Section II). However, these initiatives are mostly vendor centric, and focus on Internet-centric solutions (i.e., the "mobile as a client" approach). Indeed, providing an integrated environment for all the user's devices in a fixed-mobile pervasive environment poses a number of specific challenges (Section III) such as ubiquitous content access, efficient connectivity management, flexible communications, or security and privacy. To answer these challenges, and better support interactions between mobile users, we develop the iBICOOP middleware (Section IV). Our middleware addresses these challenges by targeting both fixed and mobile devices, leveraging their characteristics (e.g., always on and unlimited storage for home/enterprise servers, ad hoc communication link between mobile devices), and by leveraging the capabilities of all available networks (e.g., ad hoc networks, Internet, Telecoms infrastructure networks). It also relies on Web and Telecoms standards to promote interoperability

II. Background

As mobile computing emerged in the early 90's with the advent of notebooks and pre-Wi-Fi networks, much of the re-

search focused on supporting disconnected operations, mobile file systems[3], [4], and algorithms for file or object replication [5]. However, these approaches focused on powerful terminals (for the time) and mobile IP-based networks.

For normal users. mobile computing was still vague. Mobile phones however started to incorporate Personal Information Management (PIM) applications (e.g., contacts, calendar...) and vendors started to release tools for synchronizing a user's desktop and mobile phone PIM data and content (e.g., Microsoft's ActiveSync or Nokia's PC Suite). Nowadays, users not only need tools to organize and synchronize their content and PIM information, they also want to share and exchange data with friends, family and colleagues. So, content distribution over a large set of devices and mobile services in general have attracted more attention from both major industry players and the research community. Microsoft has recently launched LiveMesh[1] that proposes to offer a unified view of all the user's devices through *live desktop*; users can add or remove device to their live desktop and can manage applications and data over any of the device. They are proposing the LiveMesh sync service to synchronize data across multiple devices. Currently, only Windows-based PCs are supported but they plan to extend it to mobiles and other devices (Mac, Symbian etc.) in future. LiveMesh also allows creating shared spaces for multiple users. LiveMesh security is based around a combination of AES and RSA, along with SSL, for secure data transfer, while user identity relies on Windows Live ID.

Other approaches, offered by Apple MobileMe[2] and Nokia OVI[3], go beyond simple synchronization services that they used to offer through iTunes and PC Suite. OVI is a collection of services that help users in managing their personal information and media through the use of the Web-site of Ovi.com. Users get a personal space called "personal dashboard" and they can share photos with friends and manage their music or other medias. The main idea is to have a single location where users can manage their data (emails, contacts, bookmarks,...). Similarly, MobileMe is a set of services to keep the personal information such as e-mail, calendars, address books etc. synchronized on all user computers - Macs, Windows PCs, iPhones and iPod Touches. The idea is to use a server to store the master copy of all information over the Internet and when user machines come on-line, they synchronize. Users can get space in form of a virtual disk, where they can store and share their data, using the Web-site Me.com - managed by Apple.

Pervasive computing, mainly investigated in the research community, aims to provide computing and communication services anytime, anywhere, transparently, and invisibly, to the user using devices embedded in the surrounding environment. In the past few years, a number of pervasive computing systems have been developed. Smart Personal Information Network (Smart PIN)[6] is a performance and cost-aware personal information network, which uses a novel user-centric utility-based data replication scheme to exchange content automatically based on both network performance and user interests. There are also several research works that address content sharing in pervasive environments. In HomeViews [7], authors propose a peer-to-peer middleware system for building personal data management. They provide abstractions and services for data organization and distributed data sharing. To do so, HomeViews exposes, to applications, a database view abstraction of the file system, provides a capability-based access control scheme and defines a distributed query execution layer. XMIDDLE [8] enables transparent sharing of XML documents across heterogeneous mobile hosts, allowing on-line and off-line access to data using wireless LAN and UDP; but it does not deal with heterogeneous networks. LightPeers [9] proposes a platform for lightweight mobile pure P2P networking using different mobile devices such as Tablet PCs, PDAs and mobile phones to produce, organize, present and share digital material. The LightPeer architecture is composed of: a software stack of four component layers (Network, P2P, Sensors/Actuator and Application), a data model and a set of P2P networking protocols for transmitting data between devices. None of the three last middleware deals with security which is a key requirement of today's systems. AdhocFS [10] investigates this issue and also offers manging coherency, dynamic ad hoc groups [11] and enhanced data availability on mobile nodes [12]. Another system that deals with security is Mobile Gaia [13]. Mobile Gaia is a services-based middleware that integrates resources of various devices. It manages several functions such as forming and maintaining device collections, sharing resources among devices and enables seamless service interactions. It also provides an application framework to develop applications for the device collection. Furthermore, Gaia Microserver extensions [14], developed with J2ME, enables thin clients to fully access device specific features while respecting security through a standard interface as Gaia services.

III. Collaboration in Pervasive Environments

Enabling users to have full control over their data with all-time access is the major motivation behind our work. In this section, we present our motivations through a simple scenario, and then derive our requirements for a solution that should overcome the identified challenges.

A. Scenario

We consider Alice who leads a R&D team in a large company, and is on her way to a standardization meeting with a colleague and a contractor. Their documents for this meeting are scattered on several devices: the company servers, their laptops, and their mobile phones. Due to past interactions they all have each others certificates and device descriptions. Midway through her presentation, Alice needs specific results from recent experiments performed by the contractor. With his permission, she connects to his shared work folder, fetch the document, and display the relevant tables. For this interaction, the system needs to quickly identify which interface is the best

[1]https://www.mesh.com/

[2]http://www.apple.com/mobileme

[3]http://www.ovi.mobi/ovi/app/ovi/index

to get the document on Alice's laptop immediately. As it is a work-related request, the company-paid 3G connection can be used to quickly connect to the company servers instead of using ad hoc connections to the consultant's smartphone using Bluetooth OBEX or the overloaded WiFi network.

Later, her colleague Stan makes his own presentation on the software developed by the team. During the demo, a participant asks some details about one of the tools used to develop the software. Alice connects to both Stan and the contractor's work folders and search for all documents on this topic. She then sends some relevant white papers to the person to complete Stan's response. In this case, Alice sends a search request to the local devices of her colleague and the contractor over WiFi. As processing the request may be power intensive, both devices forward the request to the enterprise server. The contractors device also forwards the request to the contractor's network. The returned documents are displayed on the work folder of Alice's laptop, but have only been copied on the enterprise server. The enterprise server thus processes her email request.

After the meeting, the same participant presents the tools they are developing to Alice. They exchange their contact information with their mobile phones and the person is then able to send some flyers and videos to Alice. As this is not an urgent matter, the new content will be synchronized on the companys server when she returns. At this stage, a new partnership needs to be established between Alice and the participant. As it is a first encounter, both provide limited information and access credentials (i.e., mobile phones only). And also the contents are exchanged in an ad-hoc way.

B. Requirements for a user-centric pervasive middleware

From the above scenario, we identify that despite the device and environment heterogeneity, users need seamless communications with each others as well as ubiquitous content access. Furthermore, these functionalities should be provided, taking into account key transversal issues such as context-awareness and trust. Finally, naming of resources should be user-friendly so as to solve conflicts easily and their discovery should be simple and efficient. These requirements are now detailed.

Device heterogeneity: The tremendous growth of the mobile market has resulted in variety of devices available for end users. While some countries like Japan have a dominant player that enforces software and hardware specifications for these mobile devices, others like Europe or the U.S. have very large variations in the actual capabilities of the devices. Furthermore, operators, there, also have very complex subscription plans. It then becomes complex and costly for application developers to address a significant share of the mobile market. This explains the slow growth of mobile data and also, ironically, the many incompatible initiatives for new mobile platforms. This problem is however for the device and system manufacturers, and Telecoms operators to solve.

In iBICOOP, we stress the use of well-accepted standards and promote solutions that interoperate with existing protocols and services using such standards.

Communication and reachability: nowadays, mobile devices have multiple networking interfaces (e.g., 3G, WiFi, Bluetooth). However, these interfaces, and the underlying networks they connect to, are mostly aimed at anytime, anywhere Internet access (i.e., mobile device acting as a client only). They are not well suited for pervasive computing interactions. Some devices prevent background processes to preserve energy, others have a shared hardware radio for Bluetooth and WiFi that limits concurrent communications, not all recent mobile devices support the PAN profile for Bluetooth (i.e., IP network). Telecoms operators also only provide dynamic IP addresses, block ports, and do not allow incoming connection to mobile devices for all customers. Finally, interactions in pervasive computing environments also often rely on lightweight group communication (i.e., IP multicast) that is not available in 2.5/3G networks. Distributed services working on both fixed and mobile devices then require advanced communication support able to bypass such limitations.

This requires to provide mechanisms that permit easy connection between different devices. As current day devices have more than one network interfaces available, the mechanisms should be able to exploit it.

Context-aware content management: content replication and sharing in mobile environments is more than a decade-old problem and many solutions have already been investigated. However, multi-devices, multi-users environments add new constraints for moving around content: multiple sources and/or multiple destinations may be available and the correct coupling depends on context information such as the communication cost, the use and the importance of the content.

Trusted interactions: One of the major requirements of content access and sharing is secure communication. While existing and well-established security mechanisms should be used, careful integration should be considered. Indeed, the security mechanisms employed to keep content access secure should not be too intrusive and complex to be at the risk of being ignored by users. Some solutions are emerging in the internet work in terms of Single-Sign-On[4] and Windows Cardspace[5], but they have been designed for Web-site security and not pervasive computing devices. Impromptu collaborations also bring challenges for privacy along with security.

Users should be in control and aware of the content that they share. In particular it must be possible to provide transient information, or to reveal little information initially, and later complete one's information stored by other people as required.

Naming and discovery: With content and services scattered over sets of heterogeneous devices, a flexible naming scheme is required to easily and accurately locate resources. Moreover, it is important for end-users that this scheme is in human readable form to easily identify/validate the targeted resource. The process of assigning names should be easy and intuitive and users should be able to resolve conflicts if that happens.

[4]http://openid.net/

[5]http://msdn.microsoft.com/en-us/library/aa480189.aspx

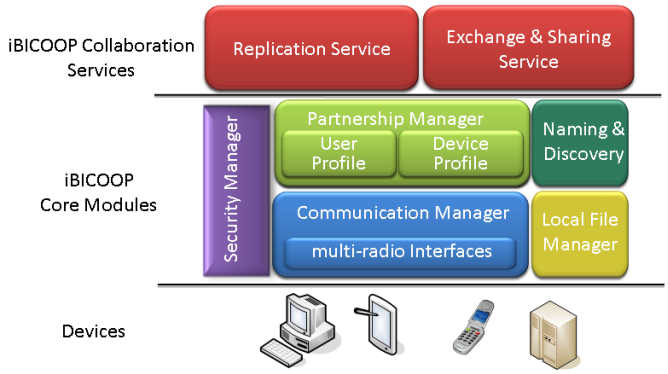

Fig. 1. the iBICOOP Architecture

Given the heterogeneous nature of the underlying networks (e.g., IP multicast support) and the limited resources of mobile devices, especially energy which prevents continuous communication, a lightweight mechanism is required to announce and discover resources, making the best use of available infrastructure services. In particular, it should be possible for a client searching a specific resource to wait for its availability, or to setup a request for connection at a specific time (e.g., for content synchronization). Finally, searching or registering resources raise concerns for privacy and trust (knowing which services a user provides or search for may lead to specific attacks)

IV. iBICOOP Architecture

The base architecture of the iBICOOP middleware, as depicted in Figure 1, consists of core modules on top of which we can develop applications that may arise in up-coming multi-device, multi-user world. Major constraints come in our work mainly from three dimensions - network characteristics, device characteristics, and software running on them - and we aim to solve them through the *core modules* of iBICOOP. Collaboration Services i.e., *Replication Service* and *Exchange & Sharing Service* are two major applications that we provide with iBICOOP. We believe that functionalities provided by iBICOOP are generic in nature and can be used to develop different applications in near future.

A. Core Modules

The *core modules* in the iBICOOP middleware provide base functionalities needed for more advanced applications:

1) Local File Manager: To support synchronization, replication, and sharing along with traditional file managing tasks, the iBICOOP Local File Manager has extra capabilities. The Local File Manager gives user clear cues to the files that have been replicated across multiple devices or shared among different users by using different icons. Users can also see if a conflict has resulted because of independent modifications on different devices.

The Local File Manager is also responsible for access control - on shared devices, users only see files, they have access to. Access control also goes to the operations that are allowed on a file by a specific user.

2) Communication Manager: The iBICOOP Communication Manager is aimed to come over the constraints of different network characteristics that different devices may have. The Communication Manager provides mechanisms to communicate over different available network interfaces of a device - Bluetooth, WiFi, Cellular - and also using different technologies e.g., Web services, http/tcp sockets, ad-hoc mode. The communication between two devices is always secured with SSL. Users can define criteria, such as low-cost or high-speed, to influence the decision of choosing a network interface over another or they can explicitly choose a particular interface [15]. It will not always be possible to have all the network interfaces available for communication, in those case user choices will be limited.

We are using a proxy-based solution to overcome the problem of IP-reachability - especially in cellular (3G) or private networks. In those cases, iBICOOP asks users to register on a proxy server. When users can not be reached directly, their data is cached on the proxy and forwarded to their registered device. Unlike other available solutions such as OVI or MobileME, users can choose or deploy their own proxy server.

3) Security Manager: The iBICOOP Security Manager uses well-established techniques of cryptography and secure communication to provide necessary security. Presently, we are providing RSA, AES, DES, and Diffie-Hellman for generating keys that can be used for encryption/decryption of data for storing on device or sending over the net. Communication over the net is always secured using SSL.

Symmetric Key algorithms are faster so we advocate their use between users, once the key has been exchanged. Secure exchange of keys is done using RSA. In some cases users can also use Diffie-Hellman to generate a symmetric secure key. However as the protocol is vulnerable to “man in the middle” attack, its use is at the discretion of users depending on the context of interaction.

The GUI of the Security Manager provides users a simple interface to create keys, using different algorithms that are available on that device, and also select files to encrypt or decrypt.

4) Partnership Manager: To overcome the challenges of device heterogeneity and provide context-aware content sharing and trusted interactions, it is necessary to have information about devices and users involved in the activity. The iBICOOP Partnership Manager provides this information in form of *profiles* - device profiles and user profiles. Profiles are stored as XML files. This allows us to easily integrate our profiles with the IMS architecture using XDMS (XML Data Management Server). We have defined XML schema for both device profile and user profile. The iBICOOP Partnership Manager provides a simple GUI to create and edit a profile.

A *device profile* is a set of information about the capabilities of the device. It contains information such as available network interfaces, reachability to/from Internet, available memory etc.

Every device in the iBICOOP middleware has a profile. The Partnership Manager provides tools to generate profile for a device which can then be edited/updated by the user. In case of mobile devices, users can transfer the profile XML file to a PC or laptop and can edit it there and then transfer back to the device - XML schemata ensure the integrity of profiles.

A *user profile* is a collection of information about the user, devices owned by user, and available security mechanisms. To use any of the service of the iBICOOP middleware, users start by creating a basic profile about them - much like user account on a Web-site. Using XML schema, they can create it using any XML editor of their choice or they can use the GUI tool provided by the Partnership Manager. This profile contains only high-level basic information - e.g., emails, id, contact information. Later as user adds a device, the corresponding device profile is added to the user profile. Creation of security keys will also show up in the user profile as available security mechanisms. This profile grows to include other information as users use iBICOOP.

In the iBICOOP middleware, profiles are used by core modules to make a service available for user. For example user profiles will need to be exchanged before starting any collaboration between users. The profiles can be exchanged over several means - email, Bluetooth OBEX. Then the Communication Manager modules on both side can extract the embedded device profile in it and use it to establish a communication.

To make our solution privacy-aware, the Partnership Manager module provides filters to control the information exchanged in a profile. To start with we define three types of filters: public, trusted or private. A filter declares what field will be exchanged, e.g., a public filter is limited to share only the type of network interfaces (e.g., WiFi, Bluetooth) while a trusted filter can share the IP addresses of the interfaces too. A user can choose a filter at the beginning of an exchange and only the information corresponding to that filter will be shared. The iBICOOP Partnership Manager module provides GUI tools for users to override the fields, defined by a filter, thus allowing them to create their own filters with any relationship as they deem appropriate.

As a collaboration happens, user profiles will be updated to include the information about the partnership. Users can also manage partnerships in groups if they wish so; in this case new partners share their profiles, using public filter at first, and then can manage the relationship as they wish. A user profile gets updated with every status of partnership that takes place or gets terminated by user. We also aim to make use of IMS presence service to provide real-time information about the partners of a user.

5) Naming & Discovery: As a general rule a naming scheme should be scalable and provide unique reference to a resource. We also insist that names are human-readable and users should be able to make a quick reference to the resource by just looking at the name. This brought us to not use auto-generated GUID (Globally Unique ID).

To ensure these goals, iBICOOP's naming scheme is based on hierarchical names in the pattern of URI scheme[6]; A similar approach has been advocated in [16], [17]. We combine names of protocol, devices, services, email etc. to give a human-readable name to a resource. For example an iBICOOP user is identifiable uniquely by using email of a user as a central part of ID; devices that a user owns are identifiable through user id and then combining the unique name of the device within that user's domain; similarly names are derived for services available from a particular device. An example for a service is : "ibic://john@work.com:ipaqpda/axis2/services/exchange".

We feel that a user can comprehend clearly about a resource by just looking at the name. As emails are unique, users will be uniquely identifiable, which will ensure that every resource related to them are also identifiable.

Most service discovery protocols for pervasive computing require IP multicast support (e.g., SLP, SSDP) either for distributed advertisement/request, or for discovering a centralized repository in the local network. While this scheme is effective in WiFi or intranets, it cannot be used over the Internet or in Telecoms networks. For such environments, UDDI-like repositories, with complex API and well-known addresses are usually available. While context-aware service discovery service can be devised for pervasive environments[18], they require powerful nodes to forward discovery messages across networks in the environment, and raise energy and security problems. As for iBICOOP, we rely on SLP to find nearby services on currently active network interfaces that support IP multicast. The search syntax is however extended to discover all resources for a specific user or a specific device in a network. As for the communication's proxy server, we consider all users have a dedicated repository where they register their resources. If a user knows the repository's address of another user *via* his profile, the repository can be queried to discover resources of another user. Finally, users may dynamically register resources to any repository given they provide the appropriate credentials (e.g., temporary repository service set up for a conference or trade show). Setting up partnerships between users (bootstrapping a relationship) may also use the discovery service, and may happen after a multicast-based local discovery (e.g., over WiFi) for any Partnership Manager service, or by searching for such services on a public repository,

B. Collaboration Services

The goal of the iBICOOP middleware is to provide abstractions that makes it easier to build services dealing with heterogeneous networks, different devices and security. We propose two collaboration services built on iBICOOP core modules: the *Replication Service* and the *Exchange & Sharing service*.

1) Replication Service: With more and more devices per user, easy replication and synchronization of files has become a key requirement for many users. The iBICOOP Replication Service keeps track of the files that are replicated over the user

[6]http://tools.ietf.org/html/rfc3986

workspace. A workspace is the total space available on all the user devices. User are made aware of conflicts if they try to synchronize with a copy that has been modified on another device. Users can add or remove files from replication service when they wish so. A multi-criteria algorithm is used to choose the best way to transfer data during replication.

2) Exchange & Sharing Service: While the Replication Service allows data management and synchronization between heterogeneous devices of the same user, using the concept of workspace, the Exchange and Sharing Service takes the abstraction to a higher level by allowing several users to interact and collaborate in various scenarios. No matter where data is located, users can transfer and share data between their workspaces. Let us consider two users, Alice and Bob. Alice has a phone with only 3G access and Bob has a PDA connected to Internet through WiFi. Alice wants to transfer a file to Bob. She knows that this file is in her workspace but she can not recall where it is exactly located. The Exchange & Sharing Service can locate all the copies of the file and present all the possibilities to transfer the file between one of Alice's devices and Bob's device.

The service allows users to choose the best way of exchanging data considering the context and their preferences. The aim of this service is to make long-term collaborations or impromptu exchanges an easier and secure task. Using this service, users can set partnerships with other users using different security mechanisms that are available to them. The iBICOOP modules provides support for different network interfaces and help overcome many of the challenges described in III-B. Once a partnership is established, connection between two devices can become immediate except for cases when device or protocol-based restrictions has to be dealt by user. Sharing a document can notify user of modifications and ability to synchronize with latest version.

C. Implementation Status

We currently implement the iBICOOP middleware in Java, using IBM J9's JVM with CDC 1.1 for Windows mobile PDA and smartphones, as well as native CLDC/MIDP for other phones and smartphones. On the infrastructure side, services (e.g., Communication proxy service and Discovery service) are developed in Java 1.6, and deployed on Apache Tomcat and on the Alcatel 5350 IMS application server (for the IMS infrastructure). A number of core modules have already been implemented (Partnership Manager, Security Manager, and Communication Manager) on specific devices and are being ported. We plan to implement iBICOOP as a tight integration between services on the Internet, local networks, and in the IMS architecture thus providing a solution for both the Telecoms and Internet world.

V. Conclusion

In iBICOOP, we are developing a middleware to allow ubiquitous access of user data from a multitude of devices with heterogeneous capabilities and running on different platforms. The services that we are building on top of iBICOOP aim to show viability of iBICOOP as a standard-based platform for future advance services. Leveraging Telecoms and Internet world, integration with IMS architecture, is one of the salient feature of iBICOOP that we have not found in other proposed solutions. To conclude iBICOOP offers a complete solution for end-user with regards to content-sharing and data management in emerging pervasive computing environments.

References

[1] V. Issarny, D. Sacchetti, F. Tartanoglu, F. Sailhan, R. Chibout, N. Lévy, and A. Talamona, "Developing ambient intelligence systems: A solution based on web services," *Autom. Softw. Eng.*, vol. 12, no. 1, pp. 101–137, 2005.

[2] *http://www-rocq.inria.fr/arles/download/ozone/index.htm*.

[3] B. D. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, and K. R. Walker, "Agile application-aware adaptation for mobility," *SIGOPS Oper. Syst. Rev.*, vol. 31, no. 5, pp. 276–287, 1997.

[4] D. B. Terry, M. M. Theimer, K. Petersen, A. J. Demers, M. J. Spreitzer, and C. H. Hauser, "Managing update conflicts in bayou, a weakly connected replicated storage system," *SIGOPS Oper. Syst. Rev.*, vol. 29, no. 5, pp. 172–182, 1995.

[5] J. Osrael, L. Froihofer, and K. M. Goeschka, "What service replication middleware can learn from object replication middleware," in *MW4SOC '06: Proceedings of the 1st workshop on Middleware for Service Oriented Computing (MW4SOC 2006)*. New York, NY, USA: ACM, 2006, pp. 18–23.

[6] S.-B. Lee, G.-M. Muntean, and A. Smeaton, "User-centric utility-based data replication in heterogeneous networks," *Communications Workshops, 2008. ICC Workshops '08. IEEE International Conference on*, pp. 290–294, May 2008.

[7] R. Geambasu, M. Balazinska, S. D. Gribble, and H. M. Levy, "Homeviews: peer-to-peer middleware for personal data sharing applications," in *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2007, pp. 235–246.

[8] C. Mascolo, L. Capra, S. Zachariadis, and W. Emmerich, "Xmiddle: A data-sharing middleware for mobile computing," *Wirel. Pers. Commun.*, vol. 21, no. 1, pp. 77–103, 2002.

[9] B. G. Christensen, "Lightpeers: A lightweight mobile p2p platform," *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, pp. 132–136, March 2007.

[10] M. Boulkenafed and V. Issarny, "Adhocfs: Sharing files in wlans," in *NCA*, 2003, pp. 156–164.

[11] M. Boulkenafed, D. Sacchetti, and V. Issarny, "Using group management to tame mobile ad hoc networks," in *MOBIS*, 2004, pp. 245–260.

[12] M. Boulkenafed and V. Issarny, "A middleware service for mobile ad hoc data sharing, enhancing data availability," in *Middleware*, 2003, pp. 493–511.

[13] S. Chetan, J. Al-Muhtadi, R. Campbell, and M. Mickunas, "Mobile gaia: a middleware for ad-hoc pervasive computing," *Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE*, pp. 223–228, Jan. 2005.

[14] E. Chan, J. Bresler, J. Al-Muhtadi, and R. Campbell, "Gaia microserver: an extendable mobile middleware platform," *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, pp. 309–313, March 2005.

[15] L. Rong, M. Fredj, V. Issarny, and N. Georgantas, "Mobility management in b3g networks: a middleware-based approach," in *ESSPE '07: International workshop on Engineering of software services for pervasive environments*. New York, NY, USA: ACM, 2007, pp. 41–45.

[16] Y. C. Chung and D. Lee, "Non-anchored unified naming for ubiquitous computing environments," *Pervasive Computing and Communications, IEEE International Conference on*, vol. 0, pp. 260–263, 2008.

[17] B. Ford, J. Strauss, C. Lesniewski-laas, S. Rhea, F. Kaashoek, and R. Morris, "User-relative names for globally connected personal devices," in *Proceedings of the 5th International Workshop on Peer-to-Peer Systems (IPTPS06*, 2006.

[18] P. Raverdy, O. Riva, A. de La Chapelle, R. Chibout, and V. Issarny, "Efficient context-aware service discovery in multi-protocol pervasive environments," *Mobile Data Management, 2006. MDM 2006. 7th International Conference on*, pp. 3–3, May 2006.