A Personal Naming and Directory Service for Mobile Internet Users

Alain Macaire and David Carlier
Gemplus Research Lab

A Personal Naming and Directory Service for Mobile Internet Users

Alain Macaire & David Carlier
Gemplus Research Lab.
BP 100 - 13881 Gemenos - FRANCE
cameleon@research.gemplus.com

Abstract

This paper proposes a new approach for the role of smartcards into distributed and mobile service environments. It is based on the naming and directory service architecture. We present a naming and directory service architecture which is based on a new component we named *Personal Naming and Directory Service* (PNDS), which is embedded on a smartcard. In section two, after a short introduction, we present PNDS concept and list advantages to have it stored on a smartcard. Section three gives an overview and limits of current smartcards applications for mobile users. Section four presents PNDS features more precisely, and shows how it has been integrated into a federated architecture of naming servers (PNDS has been prototyped using a GemXpresso JavaCard platform). To demonstrate the PNDS concept, an example of a PNDS-based application is presented in section five.

1 Internet Services and User Mobility

With the growth and spread of the Internet, vast information resources and services are making available to anyone, at any time, from anywhere in the world. People and businesses are becoming increasingly dependent on rapid and easy access to information drawn from both local and global sources. As more and more people and places become "connected", technological needs and market forces continue to change at an increasing pace.

The Internet community currently focuses part of its forces on the convergence of fixed and mobile networks to provide access to the Internet from wireless terminals (e.g., cellular phones, pagers, in-car computers, palm-top computers). Internet Engineering Task Force (IETF) is chartered to develop or adopt architectures and protocols to support mobility within the Internet [1]; World Wide Web Consortium (W3C) is working towards making information on the World Wide Web accessible to mobile devices [2]; WAP Forum (Wireless Application Protocols) [3] is defining de-facto world standard for wireless information and telephony services on digital mobile phones and other wireless terminals; Co-operation between W3C and WAP Forum has started around a common test bed [4].

This convergence of system and network infrastructures leads to offer on-line access to mobile users regardless their physical location and the serving network, and through various types of terminal devices having different capabilities and interfaces. Therefore, mobile users will need intelligent information handling to easily access information and services and customize terminals and applications according to their own preferences (user profiles).

This paper proposes a new approach for the role of smartcards into these distributed and mobile service environments. This approach is based on the naming and directory service architecture. We present a naming and directory service architecture which is based on a new component we named *Personal Naming* *and Directory Service* (PNDS), which is embedded on a smartcard. In section two, after a short introduction, we present PNDS concept and list advantages to have it stored on a smartcard. Section three gives an overview and limits of current smartcards applications for mobile users. Section four presents PNDS features more precisely, and shows how it has been integrated into a federated architecture of naming servers (PNDS has been prototyped using a GemXpresso JavaCard platform). To demonstrate the PNDS concept, an example of a PNDS-based application is presented in section five. Finally, we conclude with future directions.

2 Naming Services

With the emergence of the Internet and distributed object technologies, naming services have become essential elements in distributed system architectures. *Naming Services* make possible communication, data exchange and co-operation among different distributed objects by providing name-to-object resolution. Moreover, naming services provide foundation for more evolved services such as *Directory Services* and *Trading Services.*

2.1 Naming & Directory Services

Naming services provide objects from a request which has a name as an argument. A naming server manages a hierarchical structure of objects, and provide navigation facilities over a logical graph naming of contexts (figure 1).

In addition, a directory server manages a collection of attributes for each registered object. Attributes hold characteristics of objects and allow servers to provide client with powerful search and filtering mechanisms on attributes, hence on objects. Clients specify search criteria with their requests, and get in return a list of objects which match those criteria.

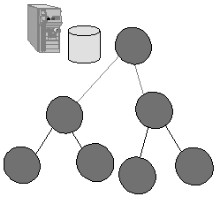

Figure 1: Graph of Naming Contexts

Naming and directory services can be viewed as special address books which are distributed across the network and which provide information on distributed objects. Objects may be of different types such as for example IP adresses from the Domain Name Service (DNS) [5], CORBA Interoperable Object References (IOR) [6], corporate directory entries from an LDAP database (Lightweight Directory Access Protocol) [7], or personal directory entries from a personal address book.

The list of attributes along with their types depend on the type of registered objects. For example, in case of an address book, `e-mail` and `phone-number` are attributes of a `person` entry; in case of a network-printer directory service, `printing-quality` (laser vs. dot-matrix) is an attribute for a `printer` entry; in case of a user profile, `prefered-colors` and `languages` are attributes for a particular user service.

Combining objects with attributes allows servers to provide each time an adapted service. *Trading Services* benefit from these features and provide users with features to discover and access to new services according to their types and characteristics.

2.2 LDAP

Even if many naming servers have already been implemented for a while such as Domain Naming Service (DNS), Network Information System (NIS), or CORBA naming service (COS), LDAP is a new emerging naming and directory service.

The main interest of LDAP consists of flexibility. All current naming services can be implemented with this protocol. The structure of an LDAP service is based on a hierarchy

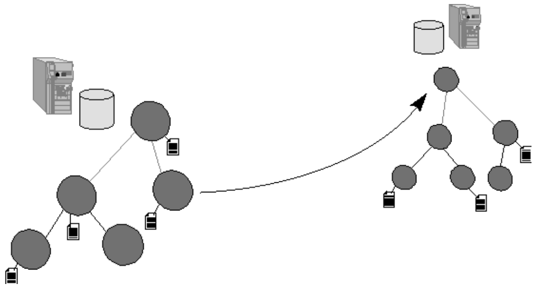

Figure 2: Referral vs. Referred Context

of entries made of attributes and bound objects. The lookup of objects and the search according to filters on attributes provide with convenient accesses [8]. Access controls are supplied by identification and authentication.

Interesting features of LDAP include the support of *Referrals Contexts*. This special type of entry is used to forward requests to other naming servers on the network when the current server cannot provide with the requested object. With referrals, different naming spaces from different naming servers can be linked together (figure 2). Referral entries also allow to share data among several users and make easier global updates on distributed databases.

We have chosen LDAP protocol as a reference for the *Personal Naming and Directory Service* (PNDS).

3 PNDS

Naming and directory services are traditionally supported by network servers and are provided to users as part of their network and service provider subscription.

However, on-line connections and services evolve to become more personalized to users and available at anytime from anywhere. The concept of *Personal Naming and Directory Service* (PNDS) was developed to provide mobile users with the part of naming and directory service that may be private and personalized. PNDS is implemented on a smartcard and is fully integrated in the overall naming and directory architecture through referrals (figure 3).

PNDS is a generic component which is able to store a hierarchical directory of bound objects along with pairs of attribute-value. Therefore, PNDS is perfectly suited to store various kind of users' or network related data, such as for example :

- object references (e.g., network addresses) which allow the system and network to bind to remote services,
- user service profile entries, which personalize services the user has subscribed to,
- Users' personal applications such as for example a personal address book.

3.1 Three Modes of Operation

The PNDS leverages the LDAP concept of referrals by handling three modes of operation.

1. When set in the *Referral Ignore* mode, PNDS ignores every referral, and directory lookups are perfomed locally in the smartcard. This is especially useful when the network is unreachable, or if the user does not want to open a network connection.
2. When set in the *Referral Throw* mode, PNDS throws an exception at destination to the client application as soon as it traverses an object bound to a referral. The client application can choose to open a network connection, and request from the PNDS the remaining part of the query to complete the lookup, as well as the address to contact the server.
3. When set in the *Referral Follow* mode, PNDS is able to follow referrals on its own. Without informing the client application that the requested object is located on a remote server, PNDS requests the hosting terminal to open a network connection and forward the request.

An example of using such a feature is when the user wishes to access a specific service. As the required service information may already be stored on the smartcard (service profile), the first lookup to the PNDS can be performed using the *Referral Ignore* mode. Depending on the result, a second attempt will be issued using *Referral Throw* or *Referral Follow* modes, to link to the network and retrieve service profile information from the referred server.

Data from the PNDS can be updated either by service providers/administrators from the network, or directly by users themselves from the client application on the terminal. A security model for access controls will have to be provided (see section 7). Therefore, it will be possible to bookmark the result of queries locally on the PNDS smartcard for next uses.

3.2 Remote Attributes

Due to their tiny size, smartcards have inherent limitations in term of memory capacity (see section 4). Thus, we have introduced the concept of *Remote Attribute* to reference object attributes which are located remotely on external content servers (figure 3). Commonly, a reference attribute will be stored as a URL, but any other addressing schemes can be supported (e.g. phone number[1]).

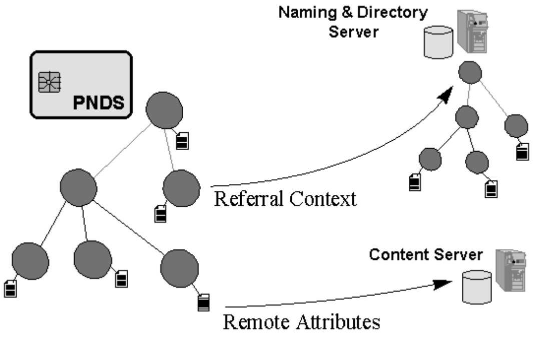

Figure 3: A Personal Naming & Directory Service

4 SmartCards for Service Personalization

A smartcard is a plastic card with an embedded microprocessor and memory which allows it to store data and execute code. The main concerns about smartcard include data confidentiality, secure authentication and high computing mobility (due to its small and convenient size). The current limits are restricted memory (up to 32Kb).

Current types of smartcards are based on either a file system [9], a small SQL-based database [10], or a virtual machine based operating system such as the JavaCard [11]. This last type of smartcards allows service downloading and is well-adapted for the development and deployment of new applications. Development and integration of services in such a JavaCard can be full object-oriented, hence the integration in distributed systems is made easier. Therefore we have chosen the Gemplus GemXpresso JavaCard, which provides full object-oriented design and programming model [12].

4.1 Current Applications for Mobile Users

As far as network access is concerned, the SIM card [13] is certainly today the most widespread smartcard. SIM cards allow mobile users to access the GSM network [14]. Upon entering a PIN code, user is identified and authenticated, and access is granted whatever the GSM terminal used. Moreover, the SIM card is used to store and provide users and terminals data such as personal address books and small interactive applications [15].

More recently, smartcards which support public-key algorithms are being deployed to provide Internet security to users [16]. These type of cards generate the private and public keys on their own. The public key can then be exported as a certificate (e.g., X500), while the private key will never be released outside of the card.

[1] It is possible to use URL addressing scheme to reference any content and services, as for example in WTA telephony services specification from the WAP Forum.

4.2 Limits of Current Network-Oriented SmartCards

Current smartcard applications allow terminal personalization. When the card is inserted, an anonymous terminal can become a personalized terminal. However, this personalization capability is still not widely used. Main smartcards concern is limited to security.

Smartcards provide data storage, but currently only the file and directory structures for binary data is deployed. This reduces the role of the card as a simple binary data server and therefore such a card cannot act in a full co-operation within architectures of open terminals, networks, systems and services.

The naming approach applied to the smartcards makes them more adapted to distributed environments. A naming environment provides to the terminal all personalized information with a better interface than a file system. Powerful searches and referral entries make easier the integration of such a smartcard into distributed systems.

4.3 Benefits for Mobile Users

PNDS extends users' mobility because the part of users' personal and private information is easily and securely carried-on from terminal to terminal. The benefits for mobile users are at least threefold.

- **Access from different access points and terminals**: Users can access services from different terminals and locations (e.g., network computers), and keep each time their own personalized features. Also, services can be adapted according to resources available locally.
- **Access in stand-alone :** Users can access part of their private and personal information securely, even in stand-alone mode, without any network elements or data involved.
- **Security:** The potential threat to individual privacy makes end-users wary about sharing personal information [17]. Storing personal and private information into a secure and tamper resistant device (i.e., a smartcard), allows the control of information exchange by means of user, application, and/or system level authentication.

5 A PNDS Implementation

The PNDS is an individual naming server with a similar approach to LDAP embedded into a smartcard[2]. This service must be supplied in any circumstances. The smartcard is an appropriate support to provide such a personal naming server for mobile users, as it contains the personalization part of mobile users' services.

A directory structure like LDAP appears to be a solution to propose different naming spaces to different services. A naming space is defined by a directory entry.

5.1 GemXpresso JavaCard

The PNDS has been prototyped on a GemXpresso: the Gemplus' 32-bit RISC JavaCard. This card allows one to easily write a card applet in Java language and to invoke it through a generated Java proxy. The client application invokes Java object methods without being aware of the specific smartcard commands.

The PNDS is made of directory entries in a hierarchical structure. Each entry contains a list of attributes. The design was made with the concern to save memory in the card and to have a better execution speed. Thus, a special attribute is referenced as the entry name, and binding an object to an entry is perfomed

[2] Even if the PNDS is similar to a smartcard-embedded LDAP server, the set of commands is specific. Due to today's smartcard memory capacitylimitation, a real LDAP implementation into a smartcard is impossible. Furthermore smart card communication protocol is different (actually it is not TCP/IP). Thus, the integration of such a server appears to be an impossible challenge.

by adding an attribute. An attribute consists of a name-value pair. For example, an entry corresponding to a person description could be as it follows.

```
cn = Durand /* common name (entry name) */
gn = Pierre /* given name               */
l  = Paris  /* location                 */
pn = +33 4 12 34 56 78 /* phone number  */
m  = pdurand@gemplus.com /* e-mail      */
```

Searching an entry in a directory is carried out by a search engine implemented in the smartcard. This search engine provides a list of entries matching attribute criteria.

A request to get, add or modify an entry is allowed only after being authenticated by the user's password.

5.2 PNDS Integration into Distributed Systems

A common interface for Java objects has been defined by several international companies to unify the access to different types of naming and directory services. This interface is named JNDI [18] (Java Naming and Directory Interface). JNDI supports are available for major naming and directory servers (e.g., X500, LDAP, NIS, COS).

We decided to choose JNDI to realize the PNDS integration into distributed systems because this model proposes a way to federate all naming servers. PNDS appears to be just a new naming and directory server to be integrated within this framework (figure 4).

In addition to the PNDS embedded-smartcard server, and like the other naming services, we have developed a JNDI SPI (Service Provider Interface) [19], outside the smartcard, to request the PNDS. A client application invokes JNDI methods to the PNDS like other naming servers without being aware of PNDS smartcard requests.

For example, a client application can call the `lookup()`, the `getAttributes()`, or the `list()` methods, to lookup, retrieve attributes, or list sub-entries of a particulary entry. Also, by calling the `createSubcontext()` or the `modifyAttributes()` methods, the client application can create new objects/entries in the graph, or create, modify or remove attributes of an entry.

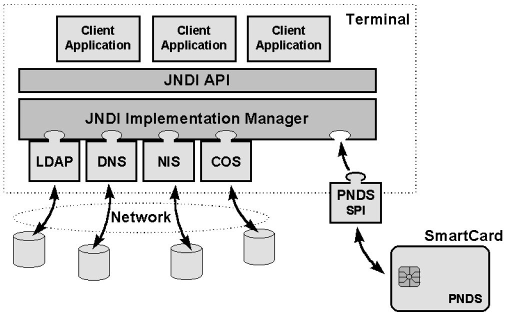

Figure 4: PNDS Integration into JNDI

The `DirContext` Java class, which has been implemented, converts standard JNDI commands to PNDS requests to a GemXpresso smartcard proxy. Implementation of the `InitialContextFactory` JNDI interface provides PNDS initial contexts, that is to say the way to access and send requests to the PNDS.

5.3 Federating Naming Services

As explained previously, implementing a JNDI interface to the PNDS supplies client applications with a unique interface. However the JNDI API allows the management of referrals. A JNDI referral context references another context. This context may be on the same server or on a different one. This other server may host a different type of naming service (e.g. COS, LDAP, ...).

A PNDS entry is a referral entry when it contains a referral attribute with a reserved name and an address to the referred context. When a request needs to explore an entry bound to a referral entry, the behaviour of PNDS depends on the mode of operation that is currently set[3] :

[3] A different operation mode can be set at each request.

- `REFERRAL_THROW` or `REFERRAL_FOLLOW`[4]: an exception is raised to the PNDS-SPI interface. This interface generates a JNDI `ReferralException` to the client application references and data delivered by the PNDS. The client can invoke the `getReferralContext()` of this exception to easily get the referred context.
- `REFERRAL_IGNORE`: PNDS ignores the referral and continues to descend the directory hierarchy, trying to find the requested entry locally. Depending on the result, the requested object or `NOT_FOUND` is returned.

The PNDS is not only a naming and directory server, but also an access point to other naming servers. The PNDS contains both personal named objects and links to other named objects managed by other servers outside the smartcard.

6 Example of PNDS Application

To demonstrate the PNDS features and capabilities, we have developed a personal address book as an example of an application (figure 5). This PNDS-based application contains two parts.

1. The first part, the address book itself, supplies mobile users with a set of person information, classified in a hierarchical structure composed of person entries and directory entries. Entries which are private to the user are stored locally on the PNDS smartcard, while shared or public entries are bound to referrals.
2. The other part provides user profile information dedicated to personalize the address book user interface on the terminal according to user's preferences. (Note that in our example, we did not use any referral or remote attribute in user profiles, but of course this can be possible.)

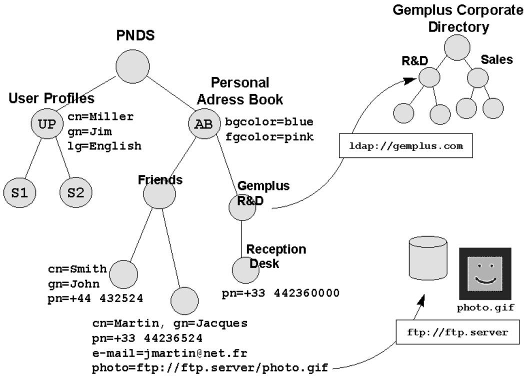

Figure 5: the Personal Address Book

This application is an appropriate PNDS demonstration. It must be available from anywhere whatever the terminal used. As opposed to the address book application code, address book data are personal. This requires features showing the advantages of PNDS and smartcards.

Via the PNDS, the address book application accesses data located on external naming server without being aware of requests to remote servers. Actually, as the address book data are distributed over the network, the PNDS opens the door to a worldwide personal address book.

6.1 PNDS as a Personalized and Private Secure Data Server

The PNDS provides information which enables users to browse their address book over a hierarchical structure of entries, from a directory entry to another. Users can select a person entry (a leaf of the structure), to retrieve its related information. A person entry consists of a set of attributes such as the common name (`cn`), given name (`gn`), phone number (`pn`), e-mail (`mail`) and photo (`jpegphoto`).

The PNDS can be viewed as an opened and

[4] `REFERRAL_FOLLOW` has not yet been implemented. Smart cards such as those supporting the SIM Toolkit API [15] are appropriate candidates for implementing this type of card-driven action towards the terminal and the network.

powerful secure data server, not only as a simple and closed secure data provider such as in a traditional file system. The PNDS is a full naming and directory server.

In addition to the lookup(), the PNDS provides a search() command to find out person(s) into the directory structure according to conditions on attributes. For example: gn=Pierre or pn=+33 4 42*.

Furthermore, the semantics of some entries may be different. For instance, Gemplus corporate directory has not to be stored inside the smartcard because it is provided and supported as part of the Gemplus' information system on the network. Thus in our case, just a referral entry is stored in the PNDS for Gemplus R&D directory entry. When this entry is selected, PNDS provides all necessary information to the JNDI naming manager to transparently forward requests to Gemplus' server (REFERRAL_THROW). If the mode is set to REFERRAL_IGNORE, PNDS provides the Reception Desk entry (pn).

Finally, some attributes of a person entry may be too large with respect to smartcard features. For example, attributes such as pictures or home-pages cannot be stored on the smartcard. However a picture attribute is bound as a remote attribute to the picture provided by a content server on the Internet.

The PNDS smartcard is one part of a personal address book distributed on the Internet. PNDS acts as a federation component.

6.2 User Profile Management

PNDS provides also an appropriate support to personalize an anonymous terminal with user profiles. We have decided to manage two levels of user profiles:

- a general user profile, which contains information related to user's general description and preferences,
- a service-specific user profile, which contains information to personalize a particular service.

In our example, the general profile is implemented by a directory entry created as a subdirectory of the PNDS root. This entry named UserProfiles (UP) provides information attributes related to the PNDS holder such as his/her common name (cn), given name (gn), and preferred language (lg) (figure 5). All these data describe the user.

We chose to store user profile information related to the address book directly as attributes to the PersonalAddressBook entry (AB). These attributes consist of the information to personalize the address book user interface according to the user's preferences such as foreground color (fgcolor), and background color (bgcolor) (figure 5).

Also, when services themselves are not part of the PNDS smartcard, user service profiles can be specified by creating new UserProfile subdirectory entries (e.g., S1, S2 on figure 5).

6.3 Perspective of this Application

As an extension to our demonstration, we plan to integrate our personal address book in an e-mail manager such as Netscape Communicator which supports access to LDAP directory servers on the Internet. In the latest 4.5 version, this application supports the notion of *Roaming Access* for roaming users to retrieve user profile information from any place on the network.

At each new connection from an anonymous terminal (e.g. a public network computer), users must manually enter an LDAP server address along with a distinguished name (dn), in order for the application to retrieve their profile and set their preferences accordingly. However, since relatively few users know their distinguished name, and probably fewer can type it correctly, this manual configuration may be tedious and can easily be replaced by just inserting a smartcard into a smartcard reader.

Also, accessing the Internet from anywhere at anytime should allow users not only to retrieve their personal preferences but also to benefit from features and services which are available locally, provided as part of the local network or service provider.

A plugin can allow Netscape Mail to access the address book service from the PNDS, and update the current configuration. Thus, when users insert their card into a terminal, the Netscape mail address book will be personalized from the PNDS.

7 Perspectives on Security

Security should play a central role in the Personal Naming and Directory Service. However, scope of this paper is only limited to decribe PNDS itself and its integration into distributed systems. Therefore, we focuse our discussion on presenting only some possible mecanisms to deploy security within the PNDS. We consider the following security concerns :

- Controling the accesses to the PNDS and its data,
- The role of the PNDS in the overall security architecture of a distributed application.

7.1 Access Controls

Access to the PNDS information is currently permitted after typing the right PIN code, nothing is supplied otherwise. However a PNDS may consist of several services for various external applications with different types of accessing users. Access to pieces of information may require a specific authorisation.

A first approach of this problem may lead to identify two kinds of users, each one having a different level of access privileges to read/write parts of the PNDS :

1. a **cardholder level**, which allow users to modify entries from their personal profiles and applications (e.g., Personal Address Book),

2. an **administrator level** (i.e., network and/or service providers), which allow service/network providers to remotely manage (update) service profile entries.

Different PIN codes can be assigned to different privilege levels, and access conditions have to be set at the context level.

7.2 Security Architecture

The other perspective concerns the overall security of distributed applications. Extensive security can be implemented for naming and directory services. PNDS can act as a keys and certificates provider, and is able to use cryptographic features provided as part of the smartcard operating system.

Possible roles of PNDS in the security of distributed application over the Internet are illustrated on figure 6. The *Secure Socket Layer* (SSL) is used to authenticate users to other naming servers on the network (i.e., referrals), while the *Remote Keys Encryption Protocol* (RKEP) [20] is used to secure content (i.e., cipher/decipher mail folders). Part of such a security architecture has already been demonstrated by Gemplus in the Vault prototype [21].

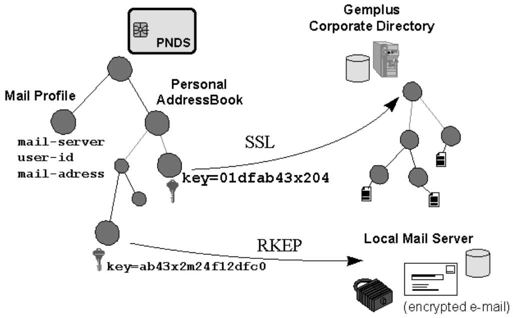

Figure 6: Example of PNDS-based Security Architecture

8 Conclusions

Providing an adapted and personalized service is not limited to just only taking into account users' preferences. With the convergence of different network infrastructures, systems and terminals, this problem encompasses network profiles, terminal profiles, and of course smartcard profiles. These include hardware and software profiles and finally users' profiles, which can be partly carried-on within the smartcard. Providing a service matching the devices capalities and users' preferences at each connection, is one of the today's challenge on network convergence (figure 7).

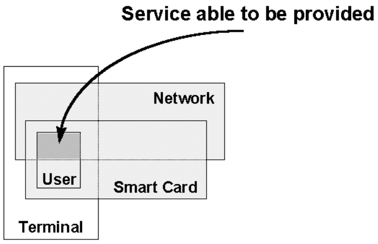

Figure 7: Adaption of Services

PNDS-based smartcard is an interesting concept for profiling aspects. It proposes a flexible structure based on a naming and directory service for applications and systems on the terminal. PNDS allows the integration of user profiles in an anonymous terminal, and personalizes the terminal and its applications with both information stored inside the card and references to personal data on the network.

PNDS is a generic component that is able to store any kind of objects, and referrals to objects accessible on the network are implemented both to get an infinite data memory capacity and to share data between several people.

Furthermore, PNDS has been integrated in a global framework and architecture based on a unified application programming interface (API). This means that client applications invoke PNDS as any other servers, without being aware of PNDS smartcard specific commands.

In this prototype, access control has been limited to PIN code authentication. Next step would be to refine the security, and define a security model for accessing and modifying PNDS local and remote objects.

9 Acknowledgements

This work was partly supported by the Commission of the European Community (CEC) as a part of the CAMELEON ACTS research project (AC341).

References

[1] Routing Area Working Groups, *IP Routing for Wireless/Mobile Hosts (mobileip),* Internet Engineering Task Force (IETF), http://www.ietf.org/html.charters/mobileip-charter.html.

[2] Mobile Access Interest Group, *Working towards seamless Web access from mobile devices,* World Wide Web Consortium (W3C), http://www.w3.org/Mobile/.

[3] Wireless Application Protocols Forum, http://www.wapforum.org.

[4] J. Hjelm, B. Martin, P. King, *WAP Forum - W3C Cooperation White Paper,* W3C, http://www.w3.org/TR/NOTE-WAP, October 1998.

[5] P. Mockapetris, *RFC-1034: Domain Names - Concepts and Facilities*, Internet Network Information, November 1987,

[6] The Common Object Request Broker Architecture - version 2.1, *Corba Services Specification - Naming Services,* Object Management Group, December 1997.

[7] W. Teong, T. Howes, F. Kille, *RFC-1777: Lightweight Directory Access Protocol,* Network Working Group, March 1995.

[8] T. Howes, *RFC-2254: The String Representation of LDAP Search Filters,* Network Working Group, December 1997.

[9] Identification Cards - Integrated Circuit(s) Cards with Contacts - Part 9, *7816-4: Inter-Industry Commands for Interchange,* International Standard Organisation (ISO), 1998.

[10] Identification Cards - Integrated Circuit(s) Cards with Contacts - Part 9, *7816-9: Enhanced Inter-Industry Commands,* International Standard Organisation (ISO), 1998.

[11] JavaCard 2.0, *Language Subset and Virtual Machine Specification,* Sun Microsystems Inc., JavaCard Forum, October 1997.

[12] J.J. Vandewalle, E. Vetillard, *Developing Smart Card based Applications Using JavaCard,* Cardis'98, in proceedings of Third Smartcard Research and Advanced Application Conference, Springer-Verlag, Louvain-la-Neuve, Belgium, September 1998.

[13] Digital Cellular Telecommunication System, *Specification of the Subscriber Identity Module (SIM)*, European Telecommunications Standards Institute (ETSI), July 1998.

[14] Digital Cellular Telecommunication System, *GSM Public Land Mobile Network (PLMN),* European Telecommunications Standards Institute (ETSI), October 1993.

[15] Digital Cellular Telecommunication System, *Specification of the SIM Application programming Interface (SIM Toolkit)*, European Telecommunications Standards Institute (ETSI), July 1998.

[16] Gemplus, *Understanding fundamentals of smartcard enabled security for Web and e-mail,* Gemplus White Paper, http://www.gemplus.com, September 1998.

[17] *Platform for Privacy Preference (P3P),* World Wide Web Consortium (W3C), July 1998.

[18] Java Naming and Directory Interface, *Interface Specification,* JavaSoft, Sun Microsystems Inc., January 1998.

[19] Java naming and Directory services, *Service Provider Interface Specification,* JavaSoft, Sun Microsystems Inc., January 1998.

[20] M. Blaze, *High-Bandwith Encryption with Low-Bandwith Smartcards,* ftp://ftp.research.att.com/dist/mab/card_cipher.ps.

[21] P. Biget, *The Vault, an Architecture for Smartcards to Gain Infinite Memory,* Cardis'98, in proceedings of Third Smartcard Research and Advanced Application Conference, Springer-Verlag, Louvain-la-Neuve, Belgium, September 1998.