MyNet: a Platform for Secure P2P Personal and Social Networking Services

D. N. Kalofonos[1], Z. Antoniou[1], F. D. Reynolds[1], M. Van-Kleek[2], J. Strauss[2], and P. Wisner[1]
[1]Nokia Research Center Cambridge (NRCC), Cambridge, MA 02142
{dimitris.kalofonos, zoe.antoniou, franklin.reynolds, paul.wisner}@nokia.com
[2]MIT Computer Science and Artificial Intelligence Lab (CSAIL), Cambridge, MA 02142
{emax, jastr}@csail.mit.edu

Abstract

Recent advances in peer-to-peer (P2P) technologies will enable users to establish ubiquitous connectivity among their personal networked devices and those of others. Building on top of such technologies, we propose a platform of middleware and user interaction tools, called MyNet, that allows everyday users to easily and securely access and share with others their devices, services, and content, without requiring expertise or centralized service support. MyNet offers a more immediate and responsive alternative to the current web-based paradigm of personal and social networking, because it allows users' distributed services and content to be accessed and shared in real-time as they are produced, directly from their personal devices. In this paper, we describe the MyNet system architecture, including secure resource discovery, service management, security framework, and the user interaction tools for building personal and social networks and sharing resources over them. We also present our proof-of-concept implementation, including mobile devices and our tests with real users.

1. Introduction

We are currently experiencing an explosion of personal digital content and services residing on digital consumer devices, such as mobile phones, cameras, portable music players, and game consoles. Today, despite their network capabilities, these devices are mainly used as stand-alone, because taking advantage of networking features still requires significant expertise and effort. At the same time, "Web 2.0" technologies are giving unprecedented control of the Internet to non-expert users. Despite its success, the current paradigm of web-based social networks has some limitations. For example, content needs to be first uploaded, making sharing more cumbersome and less interactive, and personal services hosted on users' devices are difficult or impossible to share "live".

Recently, a number of advances in peer-to-peer (P2P) systems (e.g. UIA [1], JXTA [2]) have enabled seamless connectivity among users' devices. These technologies create network overlays to address ubiquitous connectivity and management of device groups. These elements could provide the basis of a platform for decentralized P2P networking that shares users' resources directly where they reside. The success of such a platform, however, depends on the creation of easy-to-use tools that will enable non-expert consumers to manage their networks and share their resources. Furthermore, since devices are personal, the platform must make users confident not only that it offers comprehensive security and privacy protection, but, as importantly, that it offers them the means to make the right security decisions to protect themselves.

This paper presents the design and a proof-of-concept implementation of MyNet, a platform for secure P2P personal and social networking services. MyNet is built on top of the Unmanaged Internet Architecture (UIA) [1], although other P2P overlays could also be used. MyNet enables non-expert users to easily organize and share their resources within their social neighborhood. A wizard-like interface, part of a MyNet UI tool called MyNetBook, first guides the user to imprint his/her identity on a new device. Devices of the same owner are joined to create a Personal Network using MyNet's introduction process, which may be as simple as a gesture. Personal Networks can be linked to create Social Networks. Users can choose to share access to the resources they own through Passlets, real-world metaphors resembling "passes" or "tickets". The security framework permits unlimited access to the owner of devices without requiring any further user action. Social contacts, on the other hand, can only use those resources they have been granted rights to. Global connectivity and network navigation become as simple as selecting another icon on the screen, while complex configuration for service discovery, network access, and security remain hidden from the end-user.

The primary MyNet contributions are in the areas of user interaction, resource discovery, and security: (a) intuitive UI tools and modalities for P2P personal and social network management and sharing; (b) a secure P2P resource discovery mechanism independent from “native” service discovery mechanisms, that allows users to specify which of their resources can be discovered by each of their social contacts; and (c) an intuitive security framework that allows fine-grained access control, not only to services written using the MyNet security API, but even to legacy distributed applications that may not incorporate security at all.

The rest of this paper is organized as follows: in Section 2 we motivate our work; in Sections 3 and 4 we present basic design concepts and system architecture and design; in Sections 5 and 6 we give details about the security framework and the user interaction model; in Sections 7 and 8 we present our experience implementing and using our prototype; in Section 9 we discuss related work; finally we conclude in Section 10.

2. Motivation

2.1. Use cases

Alice has a Personal Network which consists of her mobile phone, laptop, home PC, and a wi-fi security camera at home. While at work, she uses her laptop to retrieve the proposal she was working on over the weekend from her home PC. She also uses her phone to check on her new pet cat through the security camera.

James’ Personal Network includes his mobile phone and his home media server. While on a trip he meets his old colleague George. Over dinner, James talks about his new baby daughter and wants to share some photos. He TAPs George’s phone with his mobile phone to give him access to the baby photos at his home media server. George can see the photos on his phone, but chooses to display them on his TV screen.

Chris is about to have another meeting with his team and clients. While waiting, he is using his laptop to access his desktop PC to set up a Wiki page for notes and to move all project documents in a directory. When the meeting starts, he sends two Passlets with his laptop to the entire group, enabling everyone to post notes on the Wiki and download project documents.

2.2. Problem statement

The use-cases described above remain challenging even today, mainly because available technology for managing pervasive access to personal devices, content and services is too complex for non-expert users.

For example, consider the case of James above. Assuming he subscribes to some dynamic DNS service to enable global addressing, he would have to login remotely to his router/firewall and open a hole to expose his PC’s web-server, which would make it accessible to everyone, not only George. He would then have to enable security on his web-server (e.g. creating certificates), password-protect his photos directory, edit the web-server Access Control List (ACL) to add George as a user and give him access to his baby’s photos directory. Revoking his access would imply repeating much of the process in reverse order. P2P technologies e.g. [1]-[6] could address the issues of connectivity, but James would still need expertise and a lot of effort to manage and secure his resources.

The only realistic alternative today for non-expert users would be sharing with email and USB sticks when possible, or subscribing to a multitude of web-based services. These methods require users to take several extra steps and assume expenses, even though their content or services are already available in their personal devices. For example, James could share his pictures after uploading them to a social networking site, while Alice could access her content by signing up to some online storage service and her wi-fi camera by signing up to the manufacturer’s remote access service.

3. Basic design concepts

In this section we present the basic concepts behind MyNet. These decisions not only affect the system design, but also to a large extent the user experience.

3.1. UIA background

MyNet uses UIA [1] as its base communication platform. UIA provides strong permanent location-independent device identifiers, and allows users to securely bind personal names to devices. Each device creates a unique public/private keypair, and hashes the public key to create an endpoint identifier (EID), which acts as the permanent device address. UIA constructs an overlay network and offers a traditional socket API to establish connections. The UIA router forwards connections over the authenticated and encrypted overlay network to the destination.

Although MyNet could be built on top of other P2P technologies, UIA provides features in two areas that meet MyNet overlay network requirements: ubiquitous connectivity and distributed device group management. UIA’s routing overlay supports IP mobility along with seamless operation though NATs and most firewalls. UIA’s authenticated mappings from device EIDs to group and user identifiers (SIDs) provides the basis of authorization queries in MyNetSec. A conventional approach of an authenticatable User ID derived from a user private/public key pair could also be followed [7].

3.2. Devices and users

A MyNet device is a routable and authenticatable overlay network endpoint, uniquely identified by its EID. It can host services and content. For devices that support multiple-user accounts, a unique EID identifies both the device and the user logged-in.

MyNet follows the UIA approach to identify users (and groups) as a set of EIDs [1]. This is possible through UIA's trustworthy distributed group membership protocol. MyNet builds authorization at the (more intuitive) user-level, as opposed to the (more common) device-level, on top of this protocol.

3.3. Imprinting

A new device becomes a MyNet device through the Imprinting process, which imprints the owner identity, profile and secret (e.g. PIN) [8]. The owner secret offers protection against misuse for critical tasks, i.e. adding/deleting a personal device. The user can also set preferences about which other actions, e.g. adding a social contact or granting permissions, are protected by this secret. The imprinting process uses the available platform UI modalities (e.g. GUI, RFID).

3.4. Personal device clusters

After a device is imprinted, it can be merged with other devices to create a Personal Device Cluster (PDC), through the MyNet Introduction process. Merging requires owner authentication on both sides and is reversible. Next, UIA and MyNet gossip merges the PDC sub-spaces known to the two devices into one.

The PDC is the basic cell of a MyNet network. It is the Personal Network of a user and all devices in it can authenticate themselves as being part of the same PDC, thus allowing privileged access to each other. This is an implicit result of the Imprinting and Introduction process and does not require further user interaction.

3.5. Social contacts

PDCs of different users can be linked through the MyNet Introduction process. The result is the addition of a social contact to both users' PDCs. Linking requires mutual consent from both sides and it is reversible. During linking, the UIA layers of the two devices exchange routing information, SIDs and EIDs [1]. Once linked, UIA and MyNet gossip takes place and the linked PDC devices know how to route overlay traffic among them. The creation of social contacts establishes long-lived trust relationships allowing users to share resources at any time. For ephemeral sharing scenarios, users may use an out-of-band mechanism to grant Passlets that allow temporary access.

3.6. Services and content

Each PDC device can run one or more user-services (services perceived by the user). Each user-service may be implemented by one or more distributed elementary services, about which the user is in general oblivious.

Services can be distinguished into MyNet-"aware", i.e. written using the public MyNet APIs, or legacy, i.e. services that were not written with MyNet in mind. Legacy services can be further distinguished into MyNet-"enabled", i.e. services for which support has been added in MyNet, and MyNet-"transparent", i.e. services about which MyNet is unaware.

User content includes files stored in any device in a user's PDC. The user content can be shared with other users though services running on devices. It is possible for the same content to be shared though the use of more than one different services. It is important to note that how the content is shared (e.g. downloaded vs. streamed) and what happens to it after it is shared, is determined by these services and not MyNet.

3.7. Groups

A user can create groups of users or devices. User groups can be used to define the recipient of access control privileges. Device groups can be used to define the target of access privileges. MyNet also defines a number of built-in user groups, such as the "world", "my direct contacts", and "my extended contacts".

3.8. Security

The MyNet security framework, called *MyNetSec*, provides authentication, authorization, privacy, and fine-grained access control. Access control is based on Passlets, which resemble the real-world metaphor of "passes" or "tickets". A Passlet contains user-level permissions for its recipient. There are Device Passlets (full access to all services hosted in a device) and Service Passlets (access to selective functionality exposed by only one service). It is also possible to give "PDC-wide" Passlets for a service, i.e. grant access to all instances of the specific service running on any PDC device. In the absence of a Passlet, access to all services over the MyNet overlay is blocked by default.

A user can use the MyNetBook Passlet Manager tool on any personal device to create Passlets that will allow their recipient to access any resources in the PDC. The recipient can be any social contact or group, or in an ephemeral interaction scenario, an unknown recipient determined through an out-of-band mechanism (e.g. NFC [9]). Similarly, a user can use a received Passlet to access the prescribed resources from any personal device in the PDC. At any point, a user can revoke a granted Passlet with any PDC device.

4. System design

In this section we present the MyNet system (Figure 1). The security framework and the user interaction model are presented in more detail in Sections 5 and 6.

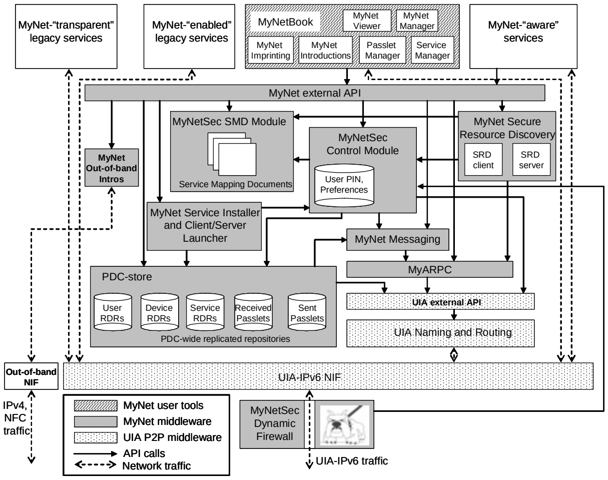


Figure 1: MyNet architecture.

4.1. Architecture overview

Adding devices and contacts to a MyNet PDC is implemented by the *Out-of-Band Introductions* module. MyNet peer device discovery is initiated through mechanisms that do not use UIA-IPv6, such as NFC [9] and Bonjour [10]. During the introduction process, MyNet peer discovery records including overlay routing information are exchanged, which bootstrap the MyNet resource discovery. The Out-of-Band module is an adaptation layer which allows the implementation of the introduction process through a variety of plug-in technologies. Introductions take place between devices in close physical proximity, thus leveraging human communication to establish trust.

MyNet system components and MyNet-“aware” applications can use an asynchronous remote procedure call (RPC) layer, called *MyARPC*, to exchange messages between processes within a single or different machines. Different existing systems could be used as a basis for MyARPC, e.g. XML-RPC, SOAP. For convenience, MyNet uses the RPC library of UIA, which is based on Sun RPC. All MyARPC messages are transmitted over secure SSL connections.

Recognizing that many MyNet devices will not be online all the time, MyNet also introduces a persistent *Messaging* service that guarantees that a one-way message will be delivered to a destination EID whenever that device comes online. The messaging module achieves this by queuing messages in persistent storage if a delivery attempt fails and periodically attempting to send queued messages. The duration of queuing is in general long (e.g. days), during which the device can be rebooted several times. Upon expiration of the queuing period the sender receives an error.

MyNet Messaging uses the API exposed by MyARPC. In turn, its API is used by MyNetSec to send Passlets, by the PDC-store for PDC replication updates, and by MyNet-“aware” applications.

4.2. PDC-wide state replication

MyNet maintains a persistent and replicated hash-table data structure in each host, called *PDC-store*, that is designed to enable MyNet applications and services to share state across instances running on the various devices in a user's PDC. Currently, the PDC-store is used to store Resource Discovery Records (RDRs) for users, devices, and services and all Passlets.

To propagate update messages, the PDC-store relies on MyNet's messaging layer, which guarantees eventual delivery. MyNet PDC-stores, by default, follow an optimistic [11] strategy for replication, in order to provide functionality that most closely resembles simple persistent local databases to applications using them. However, the problem of achieving global consistency in partitioned (disconnected) distributed database systems under optimistic strategies requires tackling complex issues with how to detect and reconcile conflicts [12].

While many such methods have been proposed (e.g. Ivy [13], Ficus [14], Coda [15], Bayou [16], Footloose [17]), we chose a two-fold solution. We first implemented a simple strategy for reconciling updates based on timestamps. While this approach has known limitations (global time synchronization to guarantee consistency), it requires very little overhead, and may be adequate for many applications. For example, in many situations a consistency guarantee may not be required, or values might only be written by one host. Similarly, updates may be sufficiently infrequent that the probability of conflicts may be very low.

The other optimistic PDC-replication strategy, which is still work-in-progress, is derived from [18] and requires transaction histories for each read and update of a value in the table. These transaction histories make it possible to reliably detect when conflicts occur and when reconciliation is necessary.

4.3. Secure P2P resource discovery

Current service discovery frameworks either enable service discovery though some centralized directory service (e.g. UDDI [19]) or define the scope of discovery either administratively (e.g. SLP [20]) or based on local-network boundaries (e.g. Bonjour [10], SSDP [21]). In contrast, MyNet introduces a P2P resource discovery mechanism whose scope is based on users' social relationships. Also, while most discovery frameworks do not incorporate security mechanisms, MyNet uses the Passlet mechanism to allow a user to specify what resources she would like to make visible by her social contacts, thus controlling how different persons view her PDC. Finally, MyNet resource discovery is widely applicable, since it does not depend on "native" service discovery protocols.

MyNet achieves all the above by introducing its own Resource Discovery Records (RDR) corresponding to devices, services, content and contacts (users), a *Service Installer and Launcher* module, and by implementing resource discovery as a "PDC-wide" MyNet-"aware" service. The *Secure Resource Discovery* (SRD) module is responsible for creating an RDR for the device itself, for each new user this device is introduced to, and for each of the MyNet-"aware" and MyNet-"enabled" services hosted by it. New RDRs are stored/replicated in the PDC-store.

The discovery process consists of four steps that take place in sequence but at different points in time:

Step 1 – resource registration: New RDRs are created due to external events. For contacts and devices, this is the imprinting or introduction process. For services and content, MyNet introduces a server-side MyNet Service Installer and Launcher. The server-side installer installs the service and an XML document called Service Mapping Document (SMD, Section 5), which is used by the resource discovery module to create a service RDR. The server-side launcher also starts the server-service automatically or on demand.

Step 2 – user sets discovery permissions: The user uses MyNetBook to create a Resource Discovery Passlet (RD Passlet) that specifies which of the user's own RDR records may be revealed to a specific user.

Step 3 – resource discovery/browsing: All devices in a PDC receive the RDRs owned by the PDC owner through the PDC-store replication mechanism. In order to see into another PDC, the Secure Resource Discovery client (SRD-client) sends a MyARPC request that is received by the SRD-server on any of the devices of the target PDC. The SRD-server asks its MyNetSec module for the RDR records to send, which returns the RDRs authorized by granted RD-Passlets.

Step 4 – service launching: The client-side launcher for services is activated when the user selects a service RDR on MyNetBook. It maps the RDR to the corresponding client-side application and "customizes" the client-side application's configuration at launch-time based on previously received Passlets.

5. Security

One of the main contributions of MyNet is a comprehensive security framework that offers (a) ease-of-use, by exposing user-level permissions while hiding complex system-level security settings with the Passlet mechanism, and (b) fine-grained access control to a large class of legacy distributed applications that may not implement security. Besides MyNet, this framework can be applied to a wide range of other existing distributed request/response applications, such as Web Services, Web Servers, UPnP devices, XML-RPC or SOAP-based services. We believe the above make the proposed security framework unique, although research exists in areas related to security usability of distributed systems (e.g. [7], [22], [23]) and some of its individual components (e.g. [24]-[30]).

In the rest of this section, we present in detail the MyNet security framework. Additional details on its implementation and examples can be found in [31].

5.1. Overview

In order to enforce fine-grained access control, MyNetSec introduces in each PDC device a *Dynamic Firewall*, which intercepts all overlay traffic before it reaches the servers hosted on that device. The firewall is controlled by the *MyNetSec Control* module, which makes the decisions on whether captured traffic will be allowed or rejected, based on the security policies expressed by the user with Passlets. The control module calculates these policies by using information contained in the Passlet repositories in the PDC-store.

The user uses the *Passlet Manager* UI tool, to create, send, revoke, and view Passlets. Once a Passlet is created, the MyNetSec Control module sends it to its recipient in the other PDC and stores it in the 'Sent Passlets' repository of its own PDC-store. Likewise, received Passlets are stored in the 'Received Passlets' repository. The PDC-store's replication mechanism then replicates the Passlets across all PDC devices, thus automatically "programming" all PDC device firewalls.

Support for legacy services is added through the use of *Service Mapping Documents* (SMD), i.e. XML documents that contain the mappings from user-level permissions to the RPCs/actions of the known distributed interfaces. SMDs are available only for MyNet-"enabled" legacy and MyNet-"aware" services. MyNet-"transparent" legacy services can still be access-controlled using Device Passlets.

5.2. Passlets

Passlets define user-level permissions, i.e. permissions that are meaningful to humans. Each

Passlet permission has a user-friendly part which is exposed to the user (e.g. a text description with a tick-box or a pull-down menu of options), and a system representation (i.e. a permission parameter of a certain type). Passlet permission parameters can be of the following types: 'boolean', 'enumeration', 'number', 'list of numbers', 'string', 'list of strings'. System-level permissions are derived from the Passlet's user-level permissions based on the mappings found in the user-service's SMD. The SMD describes for each elementary-service which RPCs/actions should be allowed, depending on Passlet permission parameters.

Common fields in all Passlets include the information about *who* is giving permission, *to whom*, *for what*, and *for how long*. They also include a PDC-wide boolean flag, which is 'true' only if the permission is for all PDC instances of a service, and a unique 128-bit Passlet ID. Device Passlets have only one boolean permission parameter ('allow_all'), which is true only if the user wants to give full-access to the recipient. Service Passlets have at least one 'allow_all' Boolean permission parameter that gives "all-or-nothing" access to the specific user-service. This ensures that (at least) "on/off" access control is possible with Service Passlets for all user-services, even for services where fine-grained access control is not possible because of lack of information or access to their RPC interfaces.

5.3. Cumulative Passlets

Cumulative Passlets (cPasslets) are internal structures created and maintained by MyNetSec that express the overall cumulative effect of all individual Passlets granted or received by the user. cPasslets are calculated based on the contents of the PDC-store 'Sent' and 'Received' Passlet repositories, referred to as sent-cPasslets and received-cPasslets respectively. A sent-cPasslet expresses the overall permissions that the owner of the local PDC has granted to a remote user or user-group; when a device in that user's PDC attempts to access any resources, the device receiving the access request looks for the existence and the contents of a sent-cPasslet for that remote user in order to be able to make an access control decision. Similarly, a received-cPasslet expresses overall permissions that a social contact has granted to the owner of the local PDC.

cPasslets provide a snapshot of the cumulative effect of permissions granted to a user or by a user though a series of Passlets. As such, cPasslets are updated continuously and do not have a duration or expiration time. At initialization, MyNetSec scans all received and sent Passlets in the PDC-store that have not been revoked or expired and creates cPasslets for each of the users or groups for which a Passlet exists. During regular operation, MyNetSec continuously creates new and updates existing cPasslets based on the following events: new Passlet received, new Passlet sent, existing Passlet revoked, existing Passlet expired.

Finally, it is important to note that the creation and maintenance of sent-cPasslets is a trusted operation. The PDC-store content replication is a secure (authenticated, encrypted, integrity checked) operation that only devices in the same PDC can perform. Therefore, each device can trust the information in the 'Sent Passlets' repository in its local PDC-store to create and maintain sent-cPasslets, and use them to make access control decisions.

5.4. Dynamic firewall

The MyNetSec Dynamic Firewall assembles traffic into Capture Traffic Units (CTUs), i.e. units of traffic each destined to exactly one elementary service as defined in one of the SMDs. In order to know which incoming traffic to capture and how to assemble it into CTUs, the firewall receives from the control module an ordered list of SMD filters. Each SMD filter is defined as a hierarchical stack of values from well-known protocol layers: (1) *IP layer* (only UIA-IPv6), (2) *transport layer* (e.g. TCP, UDP), (3) *service transport* layer (e.g. HTTP, RTP), (4) service invocation layer (e.g. SOAP), and (5) service ID layer. Each SMD filter starts from layer 1 and ends on the highest layer that is required by the firewall to unambiguously demultiplex incoming traffic. The last element of this SMD filter list is always the "MyNet Default SMD filter", which captures any overlay traffic not captured by any of the previous SMD filters in the list. The list of SMD filters is updated continuously as new SMDs are installed. The firewall inspects all incoming UIA-IPv6 traffic and processes the list of SMD filters in order until it finds the first filter that matches it. The MyNet Default filter captures traffic not captured by the remaining filters.

Once the firewall has assembled a CTU with traffic captured by one of its SMD filters, it passes related information to the MyNetSec control module (Section 5.6). If a decision to accept is made, the firewall allows the CTU to reach the server; if a decision to reject is made, the control module instructs the firewall to reject the CTU either silently or by returning an error.

5.5. SMD module

The SMD Module is responsible for parsing the SMD documents of the services installed in each device and providing information about these services to the MyNet discovery (SRD) and security modules.

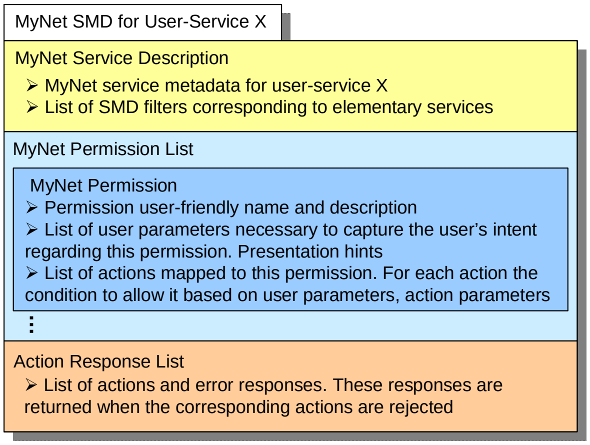

MyNet SMD for User-Service X
MyNet Service Description
➢ MyNet service metadata for user-service X
➢ List of SMD filters corresponding to elementary services
MyNet Permission List
MyNet Permission
➢ Permission user-friendly name and description
➢ List of user parameters necessary to capture the user's intent regarding this permission. Presentation hints
➢ List of actions mapped to this permission. For each action the condition to allow it based on user parameters, action parameters
⋮
Action Response List
➢ List of actions and error responses. These responses are returned when the corresponding actions are rejected

Figure 2: SMD document structure.

SMDs are XML documents containing: (a) the user-service description used to create the RDR records in the PDC-store, (b) descriptions of one or more elementary distributed services that the system uses to implement the user-service, (c) a list of user-level permissions and their mapping to arguments of RPCs/actions, and (d) a list of error codes. The main sections of an SMD are shown in Figure 2. More details and examples of SMDs can be found in [31].

The SMD Module extracts information from each section by parsing the SMD from top to bottom. The SMD Module scans all MyNet Permissions in the Permission List section and, for each SMD filter encountered, it constructs an "RPC Conditions" table indexed by RPC/actions and containing the overall conditions incoming RPC/action arguments must satisfy in order to be allowed. In those cases where an "allow" condition for an RPC/action is found under multiple MyNet Permissions in an SMD, the overall condition is constructed by creating an OR expression, starting from left to right, following the order with which permissions appear in the SMD. These expressions are evaluated by the control module.

5.6. MyNetSec control module

The MyNetSec control module controls how the MyNet dynamic firewall enforces the access control policy defined by the user through Passlets. It accesses the PDC-store Sent and Received Passlet repositories to calculate and maintain cPasslets. Access to the user-PIN and preferences allows it to protect certain critical MyNet operations. Also, the control module uses the information from the SMD module to create the ordered list of SMD filters for the firewall.

Once the firewall has assembled a CTU with traffic captured by one of its SMD filters, it passes information to the MyNetSec control module about which SMD filter captured the CTU, the source and destination EID of the CTU, the protocol PRC/action and the list of received arguments. The control module uses this information, looks up the user permissions from the corresponding cPasslet, starts evaluating the corresponding OR expression found in the "RPC Conditions" table, and accepts the RPC/action when the first condition in this expression is satisfied. If no condition is satisfied, the RPC/action is rejected.

6. MyNetBook

The MyNet user experience was a fundamental system design goal. MyNet takes advantage of familiar metaphors from everyday life and social relationships to connect devices in a user's immediate social neighborhood: the user's own devices and those of one or two social hops away. This is a significant enhancement not found in today's popular approaches for social P2P interaction (e.g. [32], [33], [34]). In addition, rather than requiring new models of behavior, we embrace intuitive human activities, such as pointing and touching, as part of the user interaction model for the purposes of building trust, establishing social contacts, sharing and simplifying lengthy configuration processes. The resulting design is a new network navigation model based on social relationships.

This section presents the system aspects of MyNetBook, an example instantiation of the MyNet UI, and briefly describes the user experience. More details about the MyNet UI model can be found in [35].

6.1. UI tools system overview

MyNetBook is a set of UI tools that use the underlying MyNet external API to create an intuitive visualization of MyNet entities and enable easy interaction between the end-user and his/her PDC. As shown in Figure 1, MyNetBook consists of the following tools: (a) *MyNet Imprinting*, which passes the user data during imprinting to the PDC-store and MyNetSec Control modules; (b) *MyNet Manager*, which uses API calls from the PDC-store and SRD modules to retrieve discovery records; (c) *MyNet Service Manager*, which allows the user to install and launch MyNet-"enabled" and MyNet-"aware" services; (d) *MyNet Viewer*, which is the front-end GUI application; (e) *Passlet Manager*, which uses API calls from the PDC-store, MyNetSec SMD and Control modules to create, edit, browse, and revoke Passlets; finally, (f) *Introduction Manager*, which uses API calls from the Out-of-Band Introductions module for the process of adding new devices and social contacts.

When MyNet is launched on a device for the first time, the Imprinting module guides the user through a wizard-like interface to imprint the user's identity and establish ownership of the PDC. The user enters a name for the PDC, profile information that can be used in the context of social networking and an owner secret (e.g. PIN). The result of the imprinting process is the creation of a PDC containing the imprinted device.

The *MyNet Manager* uses API calls exposed by the local PDC-store to collect the RDRs of the owner's PDC. It can also discover RDRs from other PDCs by using API calls from the SRD-client to send resource discovery requests. The results are passed to Viewer.

The *MyNet Service Manager* coordinates the installation and launching of services by using API calls from the SRD, SMD, Installer and Launcher modules. The user can launch a service by double-clicking on the service representation, e.g. an icon. API calls exposed by the MyNet client-side launcher (a) match the service_id from the service RDR to the appropriate client application installed locally and (b) use its launcher component to launch the client application with the correct configuration, so that it points to the local or remote distributed service.

Finally, the Passlet Manager uses information from the SMDs and RDRs in the PDC-store to (a) collect sent and received Passlets and (b) compose a set of user-level permissions based on the service attributes. Both the Passlets and the permissions are passed to the Viewer for rendering.

6.2. Personal network navigation

The key objective of the user experience is ease-of-use and intuitive interaction based on social behavior paradigms. To this end, the user interaction model exploits concepts such as personal device clusters, social networking, sharing and touch [36], [37]. At the system level, MyNet leverages human-to-human interaction to establish trust and social relationships to route traffic over the overlay network.

MyNetBook uses the well-established metaphor of the hierarchical tree structure to visualize on any device all the devices, services, content, and social contacts in the user's Personal Network. All the devices owned by a user are logically grouped together. Likewise, all services hosted by a device are logically grouped together. Devices are shown as the children of the user and services are the children of devices. Similarly, social contacts appear as children of the PDC owner. Resources shared by a contact through Passlets are shown as the children of that contact.

6.3. MyNet introductions

The MyNet Introductions process replaces lengthy manual configuration procedures with a simple point-and-click gesture. Touching or pointing to an object is a very familiar gesture both in real life and in traditional GUI interfaces. As a result, physical space becomes an extension to the traditional 2-D GUI display. There have been uses of wireless proximity technologies (e.g. IrDA, RFID) to identify and interact objects in the immediate physical proximity (e.g. [38]). However, the MyNet Introductions module makes it possible to interpret the same touch-based gesture differently based on context. For instance, it can express a user's intent to add a personal device, add a social contact, bootstrap network connectivity, invoke service discovery, give access rights, share, launch an application and so on.

MyNet uses Near Field Communication (NFC) [9], a short range wireless communication technology, to realize this interaction model. MyNet Introductions uses API calls exposed by the Out-of-Band Intros module to invoke NFC-based MyNet peer discovery, also known as TAPing. By default, devices are in the polling mode. When triggered by the user, the initiator device goes into sending mode and exchanges peer discovery records with the target (if one is in close proximity). When the Introduction process completes an RDR is created and stored in the PDC-store for the new contact/device. MyNet Introductions also uses Bonjour [10], a local area multicast discovery protocol, for devices that are not portable or NFC–enabled.

6.4. User interaction with security

While research efforts over the years have produced strong security algorithms and protocols, today most systems are not designed with usability in mind, thus making the end-users the weak link in overall security [39], [7]. For this reason, security usability was a fundamental requirement in MyNet system design.

MyNet exposes only intuitive concepts to end-users, while taking care of complex settings at the system level. The MyNet Introductions tools reduce the task of building trust with personal devices and contacts to a simple point-and-click gesture. Likewise, the Passlet Manager simplifies sharing to three, easy to make high-level decisions for *what* to share, with *whom* and for *how long*. Amending previous Passlet decisions and revoking them is equally simple. The result of completing these tasks is promptly reflected in the GUI, where users also get a clear picture of the overall access granted to each of their social contacts.

7. Implementation

MyNet was initially developed for PCs and laptops running Debian Linux, using C, C++, shell scripts and

Python. Use of Sun RPC over the wire allows MyNet to interoperate across different platforms.

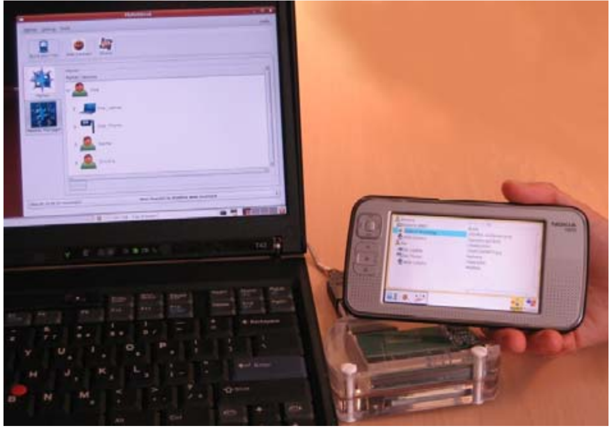


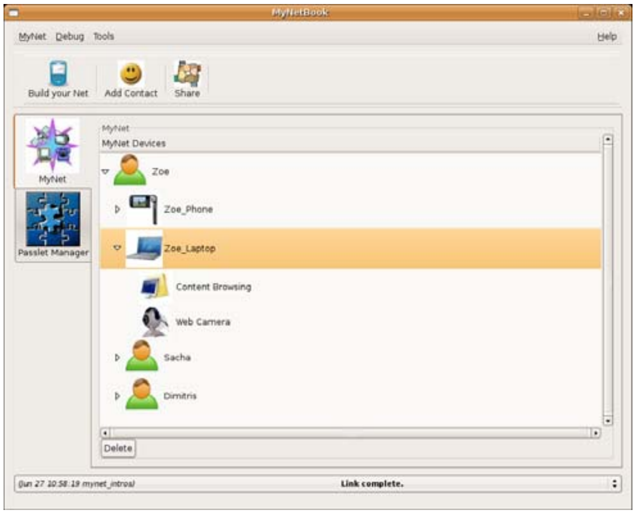
Figure 3: MyNet experimental prototype.

MyNet has also been ported to the Nokia N800 (Linux OS), and MacOS X. Nokia’s development environment for the N800 is based on Scratchbox. It includes a cross platform tool-chain that allows developers to build ARM binaries for the N800. The N800 has 128mb of RAM with up to 8GB of persistent storage and does not include many libraries normally found on a Linux system. This leads to occasional conflicts that require the adjustment of the SW components. It, also, requires its own GUI design to fit the smaller N800 screen and layout concepts. Figure 3 shows MyNet running on the current experimental setup. Both the laptop and the N800 are fitted with an NFC smart sleeve attached through the USB port [40].

We recently started a prototyping effort in native Symbian C++ for Nokia S60 phones which will enable large user trials. The Symbian port presents many challenges because (a) UIA uses Standard Template Libraries and Boost which are unavailable in Symbian, (b) MyNet is written in Python which is not space efficient for always-running background processes, (c) MyNetBook uses the GTK toolkit which is unavailable in Symbian and (d) the Symbian "idioms" require a different software architecture.

7.1. The MyNet prototype

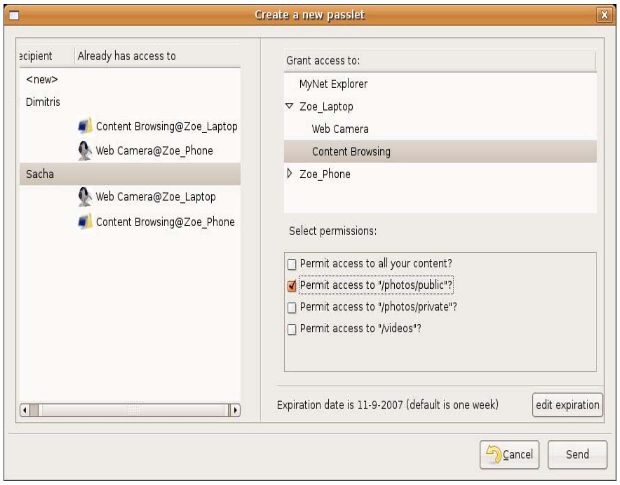
This section gives an overview of the current prototype. In our prototype, the main GUI widget is a notebook with two tabs, a toolbar at the top and a status bar at the bottom (Figure 4). By selecting the first tab the user can browse and manage his/her Personal Network. By selecting the second one, the user can manage Passlets and access rights. Interactive tasks that require the user’s attention such as imprinting, adding personal devices and contacts, sharing and error notifications are displayed in popup windows. The toolbar buttons are “Build your Net” for adding personal devices to the PDC, “Add contact” for adding social contacts to the PDC, and “Share” for creating Passlets. The status bar displays status messages.



- Zoe
  - Zoe_Phone
  - Zoe_Laptop
    - Content Browsing
    - Web Camera
  - Sacha
  - Dimitris

(Jun 27 10:58:19 mynet_intros) Link complete.

Figure 4: The MyNet Viewer



ecipient	Already has access to
<new>	
Dimitris	
	Content Browsing@Zoe_Laptop
	Web Camera@Zoe_Phone
Sacha	
	Web Camera@Zoe_Laptop
	Content Browsing@Zoe_Phone

Grant access to:

- MyNet Explorer
- Zoe_Laptop
  - Web Camera
  - Content Browsing
- Zoe_Phone

Select permissions:

□ Permit access to all your content?
☑ Permit access to "/photos/public"?
□ Permit access to "/photos/private"?
□ Permit access to "/videos"?

Expiration date is 11-9-2007 (default is one week)

Figure 5: The MyNet Passlet Manager tool.

For example, in Figure 4, Zoe owns a Personal Network with two devices, a laptop and a mobile phone, and has two contacts Sacha and Dimitris. She can browse her laptop’s services from the MyNetBook application running on her mobile phone. Zoe can launch a service (hers or that of a contact who has given her a Passlet) simply by double-clicking on the service icon, e.g. she would double-click on the Web Camera listed under Zoe_Laptop in order to launch it.

The “Share” tool button invokes the Passlet Manager (Figure 5), which has a recipient box with all existing contacts and previously issued Passlets. The user can select an existing contact for which to issue a new Passlet. Then the user needs to mark the personal device, service and piece(s) of content to share. Clicking on “Send” completes the transaction. The user can, also, select an existing Passlet in order to view or edit it. Creating a Passlet for a new contact, is as simple as selecting <new> from the recipient box, setting the target resource, pressing “Send” and TAPing the new contact’s device. This adds the recipient as a contact and grants the Passlet without any further user action.

Figure 6 shows a work-in-progress concept for the S60 MyNetBook, with two tabs for “My Contacts” and “My Devices”. The *open lock* and *keys* icons indicate that Passlets have been granted to and received from that contact respectively.

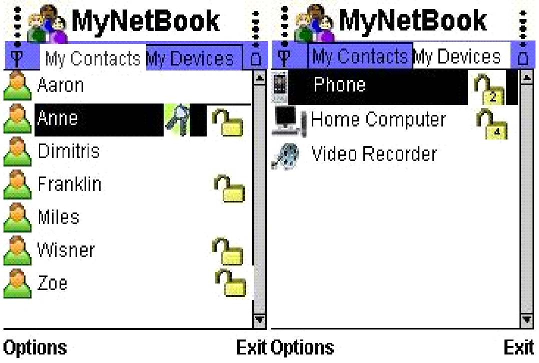


Figure 6: MyNetBook on S60 phones.

7.2. Example applications and services

We implemented two examples of secure distributed applications over a private network that highlight the social networking aspects of MyNet. The first service is web-based Content Browsing. Users can browse the content in their PDC and create Passlets for selected items. The client we used is an unmodified Firefox browser. The second application is a is a simple remote Web camera service. Other possible distributed applications could include P2P VoIP, Video over IP (e.g. see-what-I-see), Gaming, and Blogging.

8. Evaluation

This section presents results from a preliminary user study and evaluation based on the current prototype.

8.1. User study

The first MyNet usability study took place in May 2007. The objective was to assess (a) the usability of the interaction paradigms and GUI and (b) the users’ perception of MyNet concepts. There were 13 testers, ages 18-60, with no strong technical background. They were asked to perform the following tasks: set up a PDC on your mobile phone, add a laptop to your PDC, use the laptop’s web camera from your phone, using your phone share the public photos on the laptop with a friend who has a PDA, access these photos from the PDA, and using the phone revoke access to the photos.

Table 1: Key usability test results.

ISSUE	YES	IN PART	NO
Users understand the end result of imprinting	77%	15%	8%
Users can create, navigate and access a PDC	100%	-	-
Users associate sharing with new contacts	54%	39%	7%
From the GUI, users deduce the Passlet metaphor	64%	-	36%
Users can issue and revoke Passlets	100%	-	-
Adding contacts and sharing raises privacy issues	75%	25%	-
Users prefer TAPing over other wireless proximity modalities for portable devices	78%	7%	15%

Though the GUI implementation was sub-optimal at this prototype version, the results were very positive (Table 1). Testers were able to complete the tasks successfully and the majority deduced the results of their actions correctly. Though new, both the Passlet metaphor and the TAPing modality were well received. Adding new contacts and sharing raised strong privacy concerns. Through this preliminary usability test we gained valuable insight into the users’ needs and we are taking these lessons into account for the next version.

Table 2: Feature availability.

Properties: Services:	Service Sharing	Access Control	Remote Access	F2F	Scalability	Identity Management	Content Limitations
Email	no	yes	yes	no	limited	service[5]	limited[4]
VPN	difficult	no[2]	yes	no	yes	enterprise[6]	no
USB	no	yes	no	yes	limited	user	limited[4]
BTH	no	yes	no	yes	limited	user	limited[4]
UPNP	yes	no	no	yes	limited	no	no
DFS	no	difficult	difficult	yes	yes	enterprise[6]	limited[4]
Httpd	difficult[3]	difficult[1]	difficult	yes	yes	no[7]	no
Twango	no	yes	yes	no	limited	service[5]	limited[4]
Flickr	no	no	yes	no	yes	service[5]	limited[4]
FaceBook	no	limited	yes	no	yes	service[5]	limited[4]
MyNet	yes	yes	yes	yes	yes	user	no

1. Some http server implementation provide tools for authenticating users and authorizing access to content and others do not. Configuring the systems to provide such security is almost always a challenge for the typical user.
2. VPNs are common tools for enabling remote access. Although users must authenticate themselves before they can connect, VPNs rarely provide any ability to limit which services or content an authenticated user can access.
3. We make a distinction between web-based services such as Twango, Flickr, Google and Facebook based on httpd and hosting an http-based service on your home network and securely sharing access to it with your friends.
4. Services that share copies of files cannot easily share dynamic content, content streams, or any content that does not reside in a file, e.g. db records. DFS share access to files, not merely copies, but it is not easy to share non-file content.
5. Modern services usually provide reasonably easy user identity management, but users have to be willing to register. Some people object to registering with services they do not know and trust (they might be afraid of spam or identity thief).
6. Enterprise identity management systems are usually dictated. The typical user has limited control or influence.
7. Http does not mandate identity management, but it is possible using https to integrate a variety of security frameworks.

8.2. Comparison with other sharing systems

Table 2 summarizes key properties of a wide variety of technologies that enable shared access to devices, services or content and compares them to MyNet. The left column lists selected technologies. The comparison criteria (top row) include: “Service Sharing” (technology provides the means for sharing access to services), “Access Control” (technology offers access control to the resources), “Remote Access” (ability to traverse multiple NATs and Firewalls), “F2F” (“face-to-face” sharing through only local connectivity), “Scalability” (supports large numbers of users, large numbers of files, large sized content etc), “Identity Management” (captures identity issues from a user's perspective, e.g. who owns the identities used), and “Content Limitations” (limitations in the type of content that can be shared).

9. Related work

Most of the popular Social Networking systems rely on web-based centralized interfaces (e.g. Facebook [32], Myspace [33], and Flickr [34]). While these systems allow users to easily describe social links, unlike MyNet, they do not extend to user’s devices or services running on those devices, and they require a centralized infrastructure where content needs to be uploaded. A study of user practices in Flickr [41] found that some users favor a centralized system where content is viewable by all, whereas others wished to share only with social contacts. The former population is well served by Flickr and similar, whereas MyNet may be more appropriate for the latter group.

In addition, there exist a number of peer-to-peer social data-sharing systems, such as Turtle [3], SPROUT [4], F2F [5], Tribler [6]. MyNet extends capabilities offered by such P2P systems to include easy-to-use strong authentication-authorization, and support for sharing general services in addition to just content. The study in [42] examines local P2P sharing with iTunes. MyNet could allow the same sort of social interactions, without imposing any location limitations.

Usability studies of content sharing [43], [44] discuss features and limitations of existing systems. MyNet adds several new dimensions to the set of possible methods, as it is not limited to pre-existing administrative domains, does not require centralized servers, nor focus solely on sharing files. Flipper [45] is a system designed to allow easy photo sharing. Like MyNet, Flipper lets users specify the person, rather than the device, that they wish to send photos to. However, in Flipper all user identities and content are stored in a central database, hence only registered users with active database connections can share content.

Current service discovery frameworks either enable service discovery though some centralized directory service (e.g. UDDI [19]) or define the scope of discovery either administratively (e.g. SLP [20]) or based on local-network boundaries (e.g. Bonjour [10], SSDP [21]). VIA [46] describes a service discovery framework that covers multiple domains, though unlike MyNet’s domains that correspond to users, VIA instead covers topological partitions. Frank and Karl [47] examine service discovery combined with ad-hoc routing messages. Unlike all these discovery systems, MyNet introduces a P2P discovery mechanism whose scope is based on users’ social relationships, regardless of network topology and administrative boundaries. Furthermore, MyNet allows access control on which resources can be discovered through Passlets.

MyNet security is related to projects in a number of areas. Several systems attempt to address the ease of use in configuring firewalls [25], [26], [29]; these systems do not match MyNet’s fine-grained access policies as they cannot identify individual users, while in some cases [26] security is traded-off in the process. A discussion of integrating policy-driven access control with distributed firewalls in [28] does not offer a comprehensive method for creating user and service identities. Polymer [27] and Firmato [48] each provide a language and compiler to enforce security policies, though unlike MyNet, Polymer only works for specially compiled Java programs and Firmato requires the compiler to know the full network topology. Connection Conditioning [24] discusses separating web servers from their security policies, though MyNet extends this approach to cover distributed applications beyond simple web servers. Automatic management of policies on the inter-corporation scale are discussed in [30], as opposed to individual users as in MyNet. Finally, an approach which embeds extensible policies into applications is found in Alpaca [49], at the cost of pervasive application modifications.

10. Conclusions and future work

The current MyNet design and proof-of-concept implementations provide a platform for secure P2P personal and social networking services, which enables non-expert users to easily organize and share their resources. In the future we plan to enhance the PDC-store replication algorithm, introduce virtual MyNet devices and remote introductions, complete the implementation of the SMD module, and focus on new MyNet-“aware” distributed applications for personal and social networking that leverage MyNet’s P2P features. We are also planning to extend the social networking aspects in the system, e.g. interaction with friends-of-friends, search in one’s social neighborhood.

Acknowledgments

The authors would like to thank John Ankcorn of NRCC for his valuable contributions and the MIT UIA team for their close collaboration during this project.

References

[1] B. Ford *et al.*, “Persistent Personal Names for Globally Connected Mobile Devices”. in *OSDI*, November 2006.

[2] JXTA Community Projects, https://jxta.dev.java.net/

[3] B. Popescu, B. Crispo, and A. Tanenbaum, “Safe and private data sharing with Turtle: Friends Team-Up and Beat the System,” in *Proc. 12th SPW*, 2004.

[4] S. Marti, P. Ganesan, and H. Garcia-Molina, “SPROUT: P2P routing with social networks,” in *P2P&DB*, 2004.

[5] J. Li and F. Dabek, “F2F: Reliable storage in open networks,” in *5th IPTPS*, Santa Barbara, CA, Feb. 2006.

[6] J. Pouwelse *et al.*, “Tribler: A social-based peer-to-peer system,” in *5th IPTPS*, Feb. 2006.

[7] D. Kalofonos and S. Shakhshir, “IntuiSec: a Framework for Intuitive User Interaction with Smart Home Security Using Mobile Devices”. in *IEEE PIMRC*, 2007.

[8] F. Stajano and R. Anderson, “The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks”. in *Proc. SPW*, 1999.

[9] NFC Forum, http://www/nfc-forum.org

[10] Apple Computer, Inc., “Bonjour,” http://developer.apple.com/networking/bonjour/

[11] H. T. Kung and J. R. Robinson, "On optimistic methods for concurrency control," *ACM TODS*, vol. 6, pp. 213-226, June 1981.

[12] Parker *et al.*, “Detection of Mutual Inconsistency in Distributed Databases.” IEEE Transactions on Software Engineering Issue 3. May 1983 pp. 240-247.

[13] A. Muthitacharoen, *et al.*, “Ivy: A read/write peer-to-peer file system,” in *Proc. 5th OSDI*, 2002.

[14] R. G. Guy *et al.*, “Implementation of the Ficus replicated file system,” in *USENIX Summer*, 1990.

[15] J. J. Kistler and M. Satyanarayanan, “Disconnected operation in the Coda file system,” in *13th SOSP*, 1991.

[16] D. B. Terry *et al.*, “Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System,” in *15th SOSP*, 1995.

[17] J. Paluska et al., “Footloose: A case for physical eventual consistency and selective conflict resolution,” in *Proc. 5th WMCSA*, 2003.

[18] Davidson, S. B. 1984. Optimism and consistency in partitioned distributed database systems. *ACM Trans. Database Syst.* 9, 3 (Sep. 1984), 456-481.

[19] UDDI project, http://www.uddi.org/

[20] IETF, RFC 2165: Service Location Protocol

[21] UPnP Forum, Simple Service Discovery Protocol, “UPnP Device Architecture 1.0.1”, Dec 2003.

[22] J. Wang et al. Secure smart environments: Security requirements, challenges and experiences in pervasive computing. In *Experience Workshop on Pervasive Computing*, July 2005.

[23] D. Balfanz, G. Durfee, and D. K. Smetters. In search of usable security: Five lessons from the field. In *IEEE Security & Privacy*, pp 19–24, Sept/Oct 2004.

[24] K. Park and V. S. Pai, “Connection conditioning: Architecture-independent support for simple, robust servers,” in *Proc. NSDI*, San Jose, CA, May 2006.

[25] “Firewall builder,” http://www.fwbuilder.org

[26] S. Mizuno *et al.*, “A new remote configurable firewall system for home use gateways,” in *Proc. CCNC*, 2005.

[27] L. Bauer *et al.*, “Composing security policies with polymer,” in *Proc. PLDI*, 2005.

[28] T. Dimitrakos *et al.*, “Policy-driven access control over a distributed firewall architecture,” in POLICY’02.

[29] G. Munz *et al.* “DIADEM firewall: Web server overload attack detection and response,” in Broadband Europe (BBEurope) 2005, Bordeaux, France, December 2005.

[30] J. Burns *et al.*, “Automatic Management of Network Security Policy by Self-securing Networks,” in Proceedings of DISCEX II, 2001.

[31] D.N. Kalofonos, "MyNetSec: Intuitive Security for Peer-to-Peer (P2P) Personal and Social Networking Services". Published in Nokia Research Center Technical Report (NRC-TR-2007-014), Nov. 2007, http://research.nokia.com/files/NRCTR2007014.pdf

[32] “Facebook”, http://www.facebook.com/.

[33] “Myspace”, http://www.myspace.com/.

[34] “Flickr”, http://www.flickr.com/

[35] Z. Antoniou & D.N. Kalofonos, “User-Centered Design of a Secure P2P Personal and Social Networking Platform”, *In Proc.* IASTED Intl. Conf. on Human Computer Interaction (IASTED-HCI’08), March, 2008.

[36] Z. Antoniou & D.N. Kalofonos, "NFC-based Mobile Middleware for Intuitive User Interaction with Security in Smart Homes", *Proc. of* IASTED CSN, Aug. 2006.

[37] Z. Antoniou, & S. Varadan, “Intuitive Mobile User Interaction in Smart Spaces via NFC-enhanced devices”, *Proc. IEEE ICCGI*, 2007.

[38] Siegemund F. and. Florkemeier C.. “Interaction in Pervasive Computing Settings using Bluetooth-enabled Active Tags and Passive RFID Technology together with Mobile Phones”. *In Proc.* IEEE PERCOM’03.

[39] A. Whitten and J. D. Tygar. Why johnny can’t encrypt: A usability evaluation of PGP 5.0. *In Proceedings o*f the 8th Usenix Security Symposium, pages 169–184, 1999.

[40] M. Michalakis, D.N. Kalofonos, Shafai B., An experimental hardware extension platform for mobile devices in smart spaces, *in Proc. PSC’06*, 2006.

[41] A. Miller, W. Edwards, “Give and take: a study of consumer photo-sharing culture and practice,” CHI ’07.

[42] A. Voida *et al.* “Listening in: practices surrounding iTunes music sharing,” in CHI ’05, 2005.

[43] S. Voida *et al.*, “Share and share alike: exploring the user interface affordances of file sharing,” in CHI, 2006.

[44] J. Lee, “An end-user perspective on file-sharing systems,” Commun. ACM, vol. 46, no. 2, 2003.

[45] S. Counts and E. Fellheimer, “Supporting social presence through lightweight photo sharing on and off the desktop,”. *In Proc* CHI ’04, 2004.

[46] P. Castro, *et al.*, “Locating application data across service discovery domains,” in *MobiCom*, 2001.

[47] C. Frank, H. Karl, “Consistency Challenges of Service Discovery in Mobile Ad Hoc Networks”, *MSWiM*’04.

[48] Y. Bartal *et al.* "Firmato: A novel firewall management toolkit," ACM Trans. Comput. Syst. 22, 4, Nov. 2004.

[49] C. Lesniewski-Laas, B. Ford, J. Strauss, R. Morris, and M. F. Kaashoek, “Alpaca: Extensible authorization for distributed services,” in *Proc. 14th ACM CCS*, 2007.