MAIPAN – Middleware for Application Interconnection in Personal Area Networks

Miklós Aurél Rónai, Kristóf Fodor, Gergely Biczók, Zoltán Turányi, András Valkó
Ericsson Hungary, Traffic Lab
{Miklos.Ronai, Kristof.Fodor, Gergely.Biczok, Zoltan.Turanyi,
Andras.Valko}@ericsson.com

Abstract

This paper proposes the Middleware for Application Interconnection in Personal Area Networks (MAIPAN), a middleware that provides a uniform computing environment for creating dynamically changing personal area networks (PANs). The middleware hides the device configuration and physical scatteredness of the PAN and presents its capabilities as a single computer to the applications. The solution provides easy set-up of PAN-wide applications utilizing multiple devices and allows transparent redirection of ongoing data flows, whenever the configuration of the PAN changes. The proposed middleware interconnects services offered by applications running on different devices by creating virtual channels between the input and output outlets of the applications. Channels can be reconfigured when configuration or user needs change. In contrast to the approaches found in the literature, this paper presents a solution where session transfer, dynamic session management are tightly integrated with strong and intuitive access control security.

1 Introduction

The ever-growing number of wireless terminals, such as smart phones, personal digital assistants (PDAs) and laptops, raises the need to set up, configure and reconfigure personal area networks (PANs) in an easy and ergonomic way.

This paper addresses the goal of creating a dynamically changeable, but uniform computing environment for PANs, which integrates wireless and wired, stationary and mobile devices that are connected to each other. The paper presents the Middleware for Application Interconnection in Personal Area Networks (MAIPAN) solution–which is situated below the applications and above the network layer–that hides the individual devices participating in the PAN and presents the capabilities of applications running on the devices as if they were located on a single computer. This provides a standard "PAN programming platform", which allows the easy set-up of personal area networks and dynamic connection and disconnection of applications running in the PAN. Application programmers using the uniform application programming interface (API) offered by the middleware can develop software without taking care of the various PAN configurations or PAN dynamics. They can assume certain capabilities, but disregard whether these capabilities are provided by one application running on one device or by a set of applications running on several devices. They only have to register the inputs and outputs of their applications at the middleware, and they do not have to take care of which kind of devices or applications will be connected to these outlets and will use their programs.

The presented middleware contains access control, flexible session management and transferable session control solutions. The middleware contains some intelligent

functions, as well, which helps the user to control the PAN and improves human computer interaction (HCI). In theory all kind of solutions for service discovery, physical, link or networking layers can be used with MAIPAN.

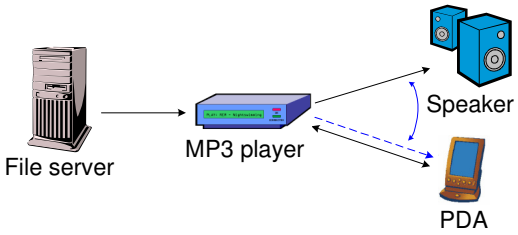

Figure 1: Listening to music

An example for a PAN application can be seen in Figure 1. Assume a MAIPAN-aware PDA user, who enters a MAIPAN-aware room where speakers are placed in the corners. The user decides to listen to music, but she neither has mp3 files nor an mp3-player. She turns on her PDA to check what kind of devices and services are available in the room. First, independently from MAIPAN the PDA's networking protocol establishes network connections to the devices located in the room, and the service discovery protocol (SDP) looks for services offered in the environment. After these steps she can configure a PAN with the MAIPAN dispatcher, which contains her PDA, the speakers in the corner, a fileserver and an mp3 player that are available through the network. MAIPAN establishes the necessary connections, that is, it connects the fileserver to the mp3 player's file input, the sound output of the mp3 player to the speakers and the control input of the mp3 player to the PDA. From now on she can select the mp3 file on the fileserver she wants to listen to, and instruct the mp3 player via her PDA to play the music on the speakers. In the figure the arrows show the data flow between the devices.

The paper is organized as follows. Section 2 is about related work, in section 3 the basic concepts and the middleware's architecture is introduced. Finally, section 4 concludes the paper.

2 Related Work

Mark Weiser presented the concept of *ubiquitous computing* in the early 90s [1][2]. Ten years after Weiser's publication, Satyanarayanan discussed the vision and challenges of ubiquitous computing in [3], where he introduced the expression *pervasive computing*, which is the end result of mixing distributed systems, mobile computing, smart spaces, invisibility, localized scalability and uneven conditioning [26].

Middleware are essential parts of pervasive and mobile computing environments. Mascolo et al. in [4] among others discussed why traditional middleware (such as CORBA [5]) is not well suited for mobile environments and how a mobile computing middleware should be designed. Another key component of ubiquitous computing is security. Chandrasiri et al. [6] proposes the concept of Personal Security Domains (PSDs)–where a PSD consists of a user's personal devices, and investigates security issues regarding PANs. Nowadays several projects are running in these research topics, some of them are presented in the followings and in our previous papers [26][27].

The goal of the AURA project [7] is to provide each user with an invisible aura of computing and information services that persists regardless of location. The project Gaia [8] designs a middleware infrastructure to enable active spaces in which data and tasks are always accessible and are mapped dynamically to convenient resources present at the current location of the user. The Oxygen project [9] aims to develop very intelligent, user-friendly and easy-to-use mobile devices enabling users to communicate with the system naturally, using speech and gestures that describe their intent. The Portolano project [10] focuses on highly adaptive user interfaces, an intelligent, data-centric network infrastructure and a new environment for developing distributed services. In the frame of this project the one.world architecture [11] is designed, which is a comprehensive framework for building pervasive applications. It includes services like service discovery, checkpointing, and migration in order to support programmers by creating applications for dynamically changing environments. The extrovert-Gadgets (e-Gadgets) project [12] investigated architectures for the composition of ubiquitous computing applications and proposed the GAS-OS middleware to interconnect and control sensors and actuators with more intelligent devices. The 2WEAR [13] project investigated the vision of a personal wearable system that can be dynamically composed out of different devices that are heterogeneous in terms of both form and function. The Cortex project [14] addresses the emergence of a new class of applications that operate independently of human control. The key objective of the EasyLiving [15] project is to create an intelligent home and work environment, which is based on the InConcert middleware solution. The Speakeasy approach [16] focuses on the specification of minimal interfaces between devices using mobile agents and mobile code. MobiDesk [17] is a mobile virtual desktop computing hosting infrastructure that leverages improvements in network speed, cost, and ubiquity to address the complexity, cost, and mobility limitations of today's personal computing infrastructure. Some concepts [18][19][27] are creating a big virtual device from the devices surrounding the users.

Similar to some of the above approaches, MAIPAN represents an entire personal area network as a single device to applications. In MAIPAN, users can easily reconfigure the PAN transparently to applications via a simple channel management mechanism. Application designs therefore do not have to take PAN configuration, its changes or even the notion of the PAN into account. The basic ideas of MAIPAN were introduced in [20].

3 The MAIPAN Platform

3.1 Basic concepts and definitions

MAIPAN distinguishes among *devices*, *applications* and *services*. The word "device" refers to the physical device and by "application" we refer to the software that offer the "services". For example, using these abstractions in case of a mouse we can say, that the mouse is a "device" where a "mouse application" is running, which offers a "mouse service". Or another example is the mp3 player: there is a device where the mp3 player application is running, which offers the mp3 player service. The distinction between application and service is necessary, because users want to use services, they do not want to care about the application offering the service.

MAIPAN is based on three concepts (Figure 2): *pins*, *channels* and *sessions*. Applications offering the services have input and output outlets, which are called pins–borrowing the expression from the integrated circuit world. Pins are the connection points of the applications to the middleware, so the middleware sees the applications in the PAN as a set of input and output pins. A pin has a pre-defined type, which shows the type of data that the pin can emit or absorb, that is, the type of information the application can handle (e.g., mouse movements, keystrokes). According to the needs new types can be defined any time. The dispatcher application is responsible to connect the pins with appropriate types.

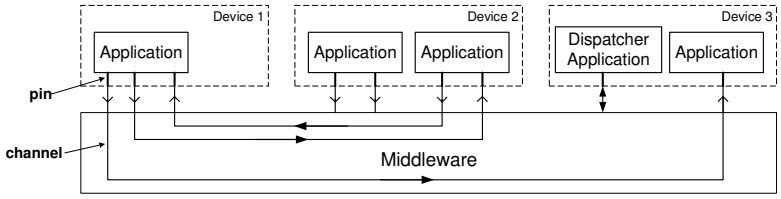

Figure 2: MAIPAN session

To enable communication between pins, the middleware creates and reconfigures channels, which are point-to-point links that interconnect pins. The set of channels that are necessary to use a PAN service is called session. For example, the channels between the entities that can be seen in the mp3 player scenario (Figure 1) compose the mp3 player session: there is a channel between the fileserver and the mp3 player, another channel between the mp3 player and the speakers, and a control channel between the mp3 player and the PDA. The PDA plays only the role of the *control entity*, it is the owner of the session, but it does not participate in it. Applications are neither aware of channels nor sessions, they only know about their pins.

3.2 Security and access control

Security and access control functions are defined and handled on device level. This means that access to services (i.e., access to application pins) are granted for devices, thus if a device gets the right to use a given service, then all applications running on this device will be able to access the service. If we assume that in the PAN there are small devices that offer one or two simple services (e.g., mouse, mp3 player), then in this case it is simpler to grant the access of services to devices instead to each application.

The dispatcher application running on a device, which plays the role of the control entity, can set up and reconfigure sessions. The control entity has to check and ask for the necessary access rights to enable the usage of a given service for the user. For instance, this main control entity can be a PDA, which has enough computing power to manage a PAN. All other devices participating in the PAN are called *participants*. In special cases, participants may delegate the access control rights to other devices, which will be referred to as *manager*.

In this paper we do not go into details regarding authentication and authorization, however, the solutions described in [6] can be used with MAIPAN to ensure secure communication between devices.

3.3 Transferring sessions

In the PAN at least one device playing the role of the controller entity is needed. In case this device disappears all concerned sessions will be automatically torn down. To keep up such session MAIPAN offers the possibility to transfer a running session from the current control entity to another one.

For example, to make some music in a meeting room one of the users creates an mp3 playing session. This way the user's device becomes the control entity of the session. After a while, when the user wants to leave, she can transfer the session by telling to MAIPAN the identity of the new control entity, which can be for example another user's PDA.

3.4 Reconfiguring sessions

In case a participant disappears (e.g., the user leaves the room, or the device's battery is depleted), the concerned channels are automatically disconnected and the sessions have to be reconfigured. In the first step MAIPAN notifies the corresponding dispatcher(s) about the event. In the second step the dispatcher application(s) can decide which services to use instead of the disappeared ones. The dispatcher application can ask for user involvement, if there are multiple possibilities to replace the disappeared service(s), or it can decide on its own, if there is no or only one choice, or the user preferences are known. In the third step the dispatcher builds up the new channels or tears down the sessions concerned.

3.5 Architecture

Based on the concepts above we designed MAIPAN, whose architecture is depicted in Figure 3. Applications run over the middleware, while layers under the middleware provide end-to-end routing and data transmission functions. As it can be seen in the figure, the protocol stack is vertically divided into two parts. The aim of the *data plane* is to provide effective and secure data transport between applications. The *control plane* is responsible for managing pins, channels, sessions and for handling security and access control.

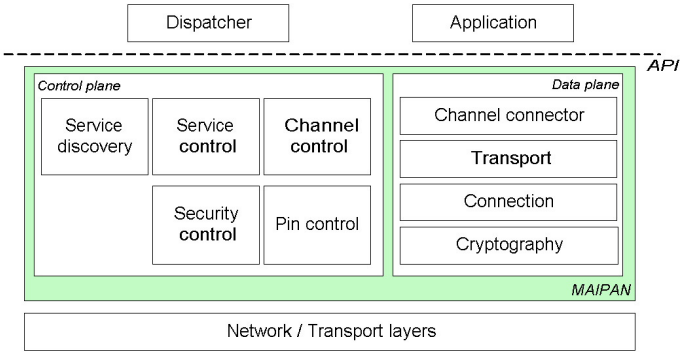

Figure 3: MAIPAN's architecture

3.5.1 Data plane

The application sends data through a pin to the middleware, where the *channel connector layer* redirects the data to the corresponding channel. The *transport layer* creates packets and provides functions such as flow control, reordering, automatic re-transmission, quality of service, etc., if these functions are not supported in the layers below the middleware, but are necessary for the application. The *connection layer* adds information to the packet, which is needed for the delivery: address of the source and destination device and the identifier of the channel. Finally, the *cryptography layer* calculates a message integrity check (MIC) value and encrypts the packet if necessary.

3.5.2 Control plane

The control plane contains the control functions which are necessary to manage the PAN. The *service control* part registers local services offered by the applications, handles their access rights and communicates with the service discovery protocol. In theory, any kind of SDP can be attached to MAIPAN, such as SLP, UPnP or Salutation [21][22]. The *channel control* creates and reconfigures sessions initiated by the dispatcher application. According to the needs of the dispatcher it asks the *pin control* parts of the participating devices to build up the channel between the pins. The pin control part instructs the channel connector layer to create a channel, activates the necessary transport functions for the given channel in the transport layer and sets the destination of the given channel in the connection layer. The *security control* part initiates and co-ordinates the authentication procedure between devices, manages the service access rights, and stores the necessary information for communication (e.g., security keys).

3.6 Implementation

An earlier version of MAIPAN is implemented in C on Linux [23]. We also created some applications (e.g., mp3 player, fileserver) and a dispatcher application in order to study the behavior of the middleware [24][25]. The lessons learnt from these are also incorporated in the design of the currently presented version of the middleware.

4 Conclusion

This paper proposes MAIPAN, a middleware for application interconnection in personal area networks. The essence of the middleware is to create a PAN programming platform, whereby hardware and software resources are interconnected, and the scatteredness of the PAN is hidden from the services. Using the proposed architecture, developers creating distributed applications for PANs do not have to take care of PAN configuration or dynamics, furthermore, they can use the uniform application programming interface (API) offered by the system.

MAIPAN represents a novel approach in its secure access control mechanism and the use of a central control entity. MAIPAN access control ensures 1) seamless interworking of various devices of the same user, 2) protection of one user's devices from devices of another user, 3) still enabling controlled communication and lending between devices of different users. MAIPAN manages device access and configuration via a convenient

central control entity, the dispatcher. MAIPAN is also unique in enabling the change of the dispatcher role, that is, the session control rights can be transferred between devices, from the old dispatcher to a new one.

In the future we plan to finalize the implementation of the current version of MAIPAN and to perform performance analysis of the middleware.

References

[1] Mark Weiser: *"The Computer for the 21st Century"*, Scientific American, September 1991

[2] Mark Weiser: *"Some Computer Science Issues in Ubiquitous Computing"*, Communications of the ACM, July 1993

[3] M. Satyanarayanan: "*Pervasive Computing: Vision and Challenges"*, IEEE Personal Communications, August 2001

[4] Cecilia Mascolo, Licia Capra and Wolfgang Emmerich: "*Middleware for Mobile Computing (A Survey)"*, In Advanced Lectures in Networking. Editors E. Gregori, G. Anastasi, S. Basagni. Springer. LNCS 2497. 2002

[5] A. Pope: *"The Corba Reference Guide : Understanding the Common Object Request Broker Architecture"*, Addison-Wesley, January 1998

[6] Pubudu Chandrasiri, Ozgur Gurleyen, Yashar Shahabi, Christian Gehrmann, Annika Jonsson, Mats Naslund: *"Personal Security Domains"*, Contribution to the 10th WWRF Meeting, New York, October 27-28, 2003

[7] D. Garlan, D. P. Siewiorek, A. Smailagic, P. Steenkiste: *"Aura: Toward Distraction-Free Pervasive Computing"*, IEEE Pervasive Computing, 2002., http://www-2.cs.cmu.edu/ aura/

[8] *"Gaia Project: Active Spaces for Ubiquitous computing"*; http://gaia.cs.uiuc.edu/index.html

[9] *"MIT Project Oxygen"*, Online Documentation, http://oxygen.lcs.mit.edu/publications/Oxygen.pdf

[10] M. Esler, J. Hightower, T. Anderson, G. Borriello: Next Century Challenges: *"Data-Centric Networking for Invisible Computing: The Portolano Project at the University of Washington"*, Mobicom '99., http://portolano.cs.washington.edu/proposal/

[11] Robert Grimm, Janet Davis, Eric Lemar, Adam MacBeth, Steven Swanson, Thomas Anderson, Brian Bershad, Gaetano Borriello, Steven Gribble, and David Wetherall: *"System support for pervasive applications"*, ACM Transactions on Computer Systems, 22(4):421-486, November 2004

[12] Achilles Kameas, Stephen Bellis, Irene Mavrommati, Kieran Delaney, Martin Colley, Anthony Pounds-Cornish: *"An Architecture that Treats Everyday Objects as Communicating Tangible Components"*, in proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom '03), Fort Worth, Texas, USA, March 23-26, 2003, p. 115

[13] 2WEAR project: *"A Runtime for Adaptive and Extensible Wireless Wearables"*; http://2wear.ics.forth.gr

[14] *"CORTEX Project: CO-operating Real-time sentient objects: architecture and EXperimental evaluation"*; http://cortex.di.fc.ul.pt/index.htm

[15] Barry Brumitt, Brian Meyers, John Krumm, Amanda Kern, Steven Shafer: *"EasyLiving: Technologies for Intelligent Environments"*, in Proc. of Handheld and Ubiquitous Computing Symposium, (Bristol, England), 2000

[16] W. K. Edwards, M.W. Newman, J. Sedivy, T. Smith: *"Challenge: Recombinant Computing and the Speakeasy Approach"*, MobiCom'02, September 23-28, 2002, Atlanta, Georgia, USA.

[17] Ricardo Baratto, Shaya Potter, Gong Su, and Jason Nieh: *"MobiDesk: Mobile Virtual Desktop Computing"*, Proceedings of the Tenth Annual ACM International Conference on Mobile Computing and Networking (MobiCom 2004), Philadelphia, PA, September 26-October 1, 2004

[18] Jonvik, T.E., Engelstad, P.E., Thanh, D.V.: *"Building a Virtual Device on Personal Area Network"*, Proceedings of 2003 International Conference on Software, Telecommunications and Computer Networks (SoftCOM'2003), Dubrovnik (Croatia) / Ancona, Venice (Italy), October 7-10, 2003

[19] Jonvik, T.E., Engelstad, P.E., Thanh, D.V.: *"Dynamic PAN Based Virtual Device"*, Proceedings of 2nd IASTED International Conference on Communications, Internet and Information Technology (CIIT'2003), 17-19 Nov 2003

[20] Miklós Aurél Rónai, Kristóf Fodor, Gergely Biczók, Zoltán Turányi, András Valkó: *"MAIPAN: Middleware for Application Interconnection in Personal Area Networks"*, poster at Mobiquitous 2005 conference, San Diego, CA, USA, July 17-21 2005

[21] Rekesh John: *"UPnP, Jini and Salutation – A look at some popular coordination frameworks for future networked devices"*, California Software Laboratories Inc., Technical Report, June 17 1999

[22] Feng Zhu, Matt Mutka, and Lionel Ni: *"Classification of Service Discovery in Pervasive Computing Environments"* MSU-CSE-02-24, Michigan State University, EastLansing, 2002

[23] Kristóf Fodor: *"Implementation of a Protocol Stack for Personal Area Networks"*, Master's Thesis, Budapest University of Technology and Economics, June 2003

[24] Fodor Kristóf, Kovács Balázs: *"A Blown-up rendszer megvalósítása"*, HTE-BME Student conference, Budapest, Hungary, May 2003

[25] Balázs Kovács: *"Design and Implementation of Distributed Applications in Ad Hoc Network Environment"*, Master's Thesis, Budapest University of Technology and Economics, May 2003

[26] Biczók Gergely, Fodor Kristóf, Kovács Balázs, Szabó Ágoston: "*Pervasive computing – rejtett számítástechnika"*, Híradástechnika journal, Budapest, Hungary, March 2003

[27] Biczók Gergely, Fodor Kristóf, Kovács Balázs, Szabó Ágoston: "Blown-up rendszer tervezése és megvalósítása", Students' National Scientific Conference, first prise, (első helyezett TDK/OTDK dolgozat), Győr, Hungary, November 2002 / April 2003