Peer-to-Peer Networks for Virtual Home Environments

Chris Loeser, Wolfgang Mueller
Paderborn University/C-LAB
Fuerstenallee 11
33102 Paderborn, Germany

Frank Berger, Heinz-Josef Eikerling
Siemens SBS/C-LAB
Fuerstenallee 11
33102 Paderborn, Germany

Abstract

This paper presents an approach to extend OSGi server based home networks with JXTA's peer-to-peer (P2P) technology to a multi-server home environment connecting multiple private home networks to one true virtual environment for communication, media sharing/exchange, and distributed device control. We elaborate on the basic integration methodology and extend it to a general, security preserving, architectural concept based on a two level integration concept for inhome *and* interhome *networks. The integration is achieved through additional OSGi services for the management of virtual devices, peer-to-peer communication, and authentication. The example of a Quality of Service based video streaming management demonstrates one of the potential advanced application for which peer-to-peer can be used in addition.*

1 Introduction

With the wide availability of computer-based networks, the interconnection and collaboration between small, mobile devices has gained significant interest in the business domain. In contrast, computer-based networks are not well established in today's home environments. Only very few households afford the luxury of a home network for the control of window shutters, heating, and air conditioning by EIB or LON, for instance. Nevertheless, due to the convergence of PCs and consumer electronics, households currently move from single-PC to multi-PC environments. With the decrease of costs for Ethernet-based network components, a significant market for home networks based on PC technologies is predicted so that even pessimistic studies see it to double in the next 3 years. This market covers the fields smart homes, intelligent homes, virtual homes, ubiquitous computing, and ambient intelligence and even includes car environments and partly mobile phone applications which may temporarily connect to home networks. Several projects and research groups investigate that areas like 3rd Generation Partnership Project, Georgia Tech Aware House, MIT House_n, Microsoft Concept Home MIRA, MIT Oxygen Project, and Philips HomeLab [1, 2].

Typical home networks are currently established by a single server architecture which serves as a home gateway (residential gateway) controlling all main inhome services and giving access to the wide area network. In this context, *entertainment*, *communication*, and *security control* are seen as the three main applications in the next generation home networks which will drive the market. Establishing a homogeneous home network with heterogeneous gadgets, will be the next step and is mandatory to get it widely accepted by the consumers. Here, on the one hand, the support of various protocols has to be managed like standard infraread, IrDA, Bluetooth, SWAP, (wireless) Ethernet, USB, Firewire, powerline etc. and partly, when connecting over wider distances, GSM and GPRS. On the other hand, multiple devices like mobile phones, PDAs, Webpads, cameras, standard remote controls, set-top boxes, notebooks, and PCs have to be seamlessly integrated considering different properties and limitations.

For the implementation of home-based services, we can currently identify two software frameworks: *Embedded Windows XP by Microsoft* and the *Open Services Gateway Initiative (OSGi)* standard [11]. The OSGi approach is to control devices through a central server, the OSGi gateway, in the house or embedded in a car. Nevertheless, *Embedded Windows* and *OSGi* both lack support in distributed multi-server *home environments* and includes no means for advanced communication. Additionally, there were some investigations to apply Jini for advanced resource management for multi-server home environments. Reported problems referred to security and

Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03)
0-7695-1874-5/03 $17.00 © 2002 IEEE

the high footprint for embedded computers and mobile devices. More recently, Sun's Project JXTA [15] was introduced to efficiently set up peer-to-peer (P2P) networks running on devices with small footprints like the Siemens SL45i mobile phone.

In research, we currently find only little work for advanced multi-server home networks. Roussaki et al. present an architecture for virtual home environments in [12] covering two access models: multi-terminal and multi-network access. They mainly investigate UI technologies based on single server architectures. Lloyd et al. [6] introduced a virtual home environment architecture based on multi agent systems classifying communication w.r.t. the distance of the user (in-building, neighborhood, suburban, global). Their focus is on UMTS rather than on in-building communication. [17] gives a very brief and general investigation on applying peer-to-peer in home networks for security, access control, and communication. All of these approaches present a very specific architecture. Most of the approaches seem to be rather limited since they are based on single server architectures and hence do not consider the existing variety of communication devices and their short range protocols like IrDA and Bluetooth.

We present an architecture which integrates OSGi with JXTA's peer-to-peer technology for communication, media sharing/exchange, and distributed device control in virtual home environments [5, 8]. We elaborate on the basic integration and extend it to a general, privacy preserving, architectural concept based two level integration with *inhome* and *interhome* connections where inhome application are based on a single OSGi-server architecture. We integrate peers and peer groups as advanced services which are applied to combine devices to groups and integrate a group as one virtual device. On this basis, peer-to-peer is also applied to connect multiple homes (interhome connections) and combine them to a real virtual home environment establishing a flexible and open multi-server architecture. As an additional advantage, peer-to-peer technology also easily supports access from wider distances through GSM, for instance.

The remainder of this paper is structured as follows. The next section introduces the general idea of virtual home environments. Thereafter, the two integrated frameworks, i.e., OSGi and JXTA are outlined. Section 3, introduces our inhome and interhome concepts combining JXTA and OSGi. That section also outlines an advanced inhome application and introduces a peer-to-peer based video service network with Quality of Service support and considers security issues with focus on peer authentication. Finally, the paper closes with some conclusions.

2 Virtual Home Environments

The idea of virtual home environments has been investigated in the VHE Middleware project [5] in order to provide an infrastructure for full network-based control and access to various home devices and sensors based on an OSGi-oriented architecture. Virtual Home Environments combine and extend concepts known as smart home, intelligent home, residential gateway, smart car, etc. It concerns the integration of several private networks in house, car and office to one virtual environment. For the user, transparency is of great importance: the "look and feel" has to be location and device independent.

As given in Fig 1, VHE middleware considers connecting devices, protocols, and services meaning that different services are accessed by multiple devices through various protocols. That figure gives examples for a potential set of services to be controlled in the context of a home network. Within the private home network we distinguish

- control devices, like mobile phone, PDA, WebPad, standard remote control etc. and
- controllable devices, like TV, VCR, light, alarm, radiator, aircondition, gas meter etc.

for different services like security monitoring, room control, consumer electronics, communication etc. Devices which combine both properties are of particular interest for P2P-based communication and control applications.

3 Communication and Service Frameworks

For establishing a virtual home environment, we combine two frameworks:

(i) peer-to-peer for communication and data exchange and

(ii) OSGi for residential services for control (Java Embedded Server).

For their implementation we use JES and JXTA which are introduced in the next subsections before we present our approach combining both in the next section.

3.1 OSGi and JES

The Open Service Gateway initiative (OSGi) started its work in March 1999 with a focus on the definition and standardization around the concept of the (residential) gateway. The gateway, also denoted as *service framework*, is designed to facilitate the development and use of services that are dynamically deployed. The framework is platform independent and can manage broadband services as well as networks in homes, cars, and comparable environments. The idea of OSGi is to provide a software gateway that interfaces the external Internet with internal (domestic) devices and identifies requirements for interoperability and general usability. For the interface, developers design an application as a set of bundles that contain services. Figure 1 gives examples of components of an OSGi application.

The central component of the OSGi framework is the residential service gateway which functions as an application server for various services such as consumer electronics (audio, video), energy management and control (gas meter), safety and security services (security monitor), healthcare monitoring services etc. The application of such a service gateways is not limited to home environments.

Within a home networking environment, a service provider offers various services. The delivery of such services is mainly enabled through the download of software applications into the service gateway. It is also possible to stage different location based services or aggregations of services, e.g., automatic upload of electricity, gas, and water meters provided by the power company. For car related services like diagnostics, emergency assistance, mobile commerce, navigation support are possible.

Figure 2 shows the architecture of the OSGi service gateway with basic services and with our VHE services which are introduced in the next sections.

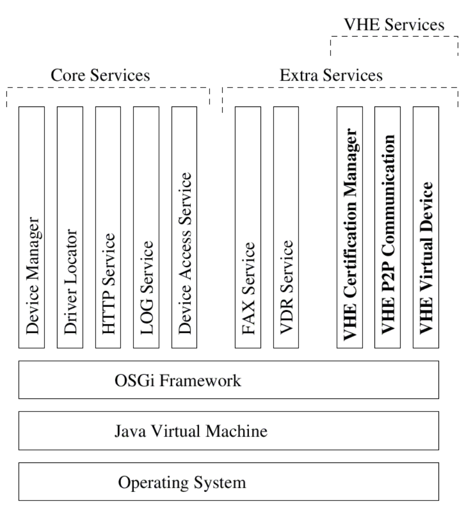

Figure 2: Basic OSGi Services with VHE Extensions

Presently, there are two OSGi implementations freely available: IBM's *Service Management Framework (SMF)* and Sun's *Java Embedded Server (JES)*. Our implementation utilises JES since it combines well with Sun's *Project JXTA* [14].

3.1.1 Java Embedded Server (JES)

The Java Embedded Server 2.0 (JES) architecture complies with the OSGi Service Gateway 1.0 [4] and requires a persistent storage, TCP/IP network, and a Java Virtual Machine.

The JES framework manages service life cycles, service registration, bundles, service dependency tracking, and event notification. In order to manage bundles and services, JES creates a BundleContext object, which permits the framework and the bundles to interact. Different core services, e.g., HTTP service, log service, device access service, servlet service, already come with JES. New services can easily access and cooperate with the core services.

A **service** is a group of Java classes and interfaces that makes up a specific feature, like the HTTP service that creates a tiny web server. All services are packaged into bundles before deploying it onto the framework. A **bundle** is basically a JAR file that contains the service and other files, images, or resources. A bundle can also contain several services that operate as an integrated unit.

The OSGi device access architecture provides a framework to allow bundles within an OSGi environment to access physical devices. The framework requires a device manager and a driver locator to connect different devices (e.g., PC, printer, TV, A/V, lighting, security devices, utility meters, etc.). The **Device Manager** coordinates the different drivers so that they can incaranate multiple representations of the same device, and initiates the process of downloading the appropriate drivers. The **Driver Locator** services provide the mechanism for dynamically downloading appropriate drivers in an OSGi device. The service implements an interface which contains two different methods to find, load, install, and start a driver.

As given in Fig. 2, we have extended OSGi by three additional services. The virtual device service is introduced in Subsection 4.1. The P2P communication manager and the certification manager are presented in Subsection 4.2.

3.2 Peer-to-Peer Networks and JXTA

The computation model on the Web is primarily based on a client/server model where information and

Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03)
0-7695-1874-5/03 $17.00 © 2002 IEEE

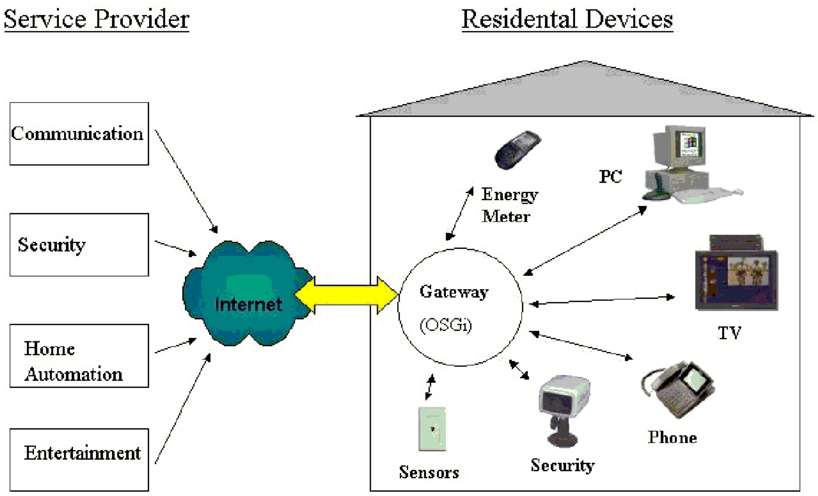

Figure 1: OSGi Residential Gateway

services are published at and accessed through well-known and fixed locations (URLs). In contrast, peer-to-peer computing avoids addressing along with centralized Web sites which has created single points of failure and bandwidth bottlenecks. In peer-to-peer computing each connected device acts as a server and a client at the same time.

Peer-to-Peer applications are typically very specific implementation mainly supporting file exchange, e.g., Gnutella, Napster, Aimster, KaZa, Freenet etc. and instant messaging, e.g., AOL Instant messenger, Yahoo Messenger, MSN Messenger, ICQ etc. [9]. Those decentralized computing systems present the experience of a community environment where members offer and share various resources.

The main idea of peer-to-peer computing is that each peer is client and server at the same time. For an application, each peer may release and allocate

- **processing power**: like for *Seti at home* a bunch of peers performs a distributed computation.

- **data**: data is not owned by a particular member or server, but is passed around and flows freely towards the subscribers. When a member demands for some data increase, more copies are propagated and replicated within the community. Otherwise, fewer copies will be available as existing copies slowly disappear and are replaced by more popular data.

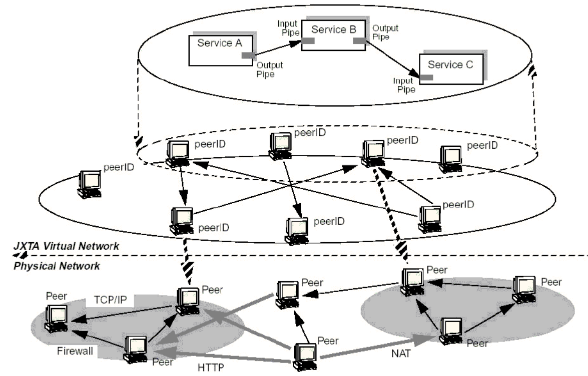

Figure 3: JXTA Virtual Network [16]

- **control**: each peer can offer the possibility of being controlled or illustrating monitored data. This is especially dedicated to sensors or small embedded devices.

3.2.1 Project JXTA

One major disadvantage in existing peer-to-peer systems is that they are mainly proprietary solutions, delivering a single type of service, and employing protocols incompatible with other services. None of the existing P2P applications are interoperable. JXTA [15] is an initiative which calls for standardization of a peer-to-peer system API and overcomes this limitations.

JXTA protocols have been designed to be implementable on any device, including sensors, consumer electronics, PDAs, appliances, network routers, desktop computers, data-center servers, and storage systems. These protocols create a virtual network on top of the existing physical network infrastructure based on which services and applications are built (Fig. 3). This virtual network layer is designed to be as simple as possible with powerful primitives in order to maintain interoperability.

The main purpose of JXTA virtual network is to hide the complexity of the underlying physical network topology, and provide a uniform addressable network for all peers in the network. The virtual network allows a peer to exchange messages with any other peers independently of its network location (firewalls, NATs, or non-IP networks). Messages are transparently routed, potentially traversing firewalls or NATs, and can use different transport/transfer protocols (TCP/IP, HTTP).

JXTA standardizes the way in which peers discover each other, self-organize into peergroups, advertise and discover network resources, communicate with each other, and monitor each other. The network transport layer is built of a uniform peer addressing scheme based on peer ID, relay peers that relay messages between peers, and a binary message format to transport binary and XML data.

A peer is uniquely identified by its peer ID (128 bit UUID), even if it uses different network addresses even when the device may have different IPs at different times (e.g., a notebook). Similarly, a device supporting multiple network interfaces (e.g., Ethernet, WLAN, Bluetooth, IrDA) can be addressed as a single peer which greatly supports our home application. The peer ID abstraction allows a peer to encapsulate not just physical transports, but also logical transfer protocols like HTTP and/or secure protocols like TLS and SSL.

JXTA runs on PDAs, mobile phones (with MIDP), web pads, and on embedded Java processors so that it can be fully integrated into a home network for various applications and may even provide facilities to connect a multi server network.

4 Combining Peer-to-Peer and OSGi Services

Our approach seamlessly combines OSGi and JXTA and presents an architecture which extends the classical single server based system by additional services like communication, media sharing/exchange, and distributed device control. We first elaborate on the basic integration and extend it to a general, privacy preserving architectural concept based on two level integration with *inhome* and *interhome* application:

- **Inhome**: several devices are connected by JXTA and offer their services to each other as far as they are not security critical. Other Device interaction is still controlled by the OSGi DeviceAccessService and does require authentification by the DeviceAccessService. Furthermore, the JXTA groupware is used to address Devices as one virtual device establishing a real extension of the single server based OSGi architecture.
- **Interhome**: multiple inhome networks are connected to a multi server network in order to establish real virtual home environments. For interconnection we introduce a JXTA-based VHE P2P Service (Fig. 5). Individual devices are still controlled by the local DeviceAccessService but the control signals of JXTA peers are additionally distributed over a P2P communication instance over the global network. For secure environments, an additional CertificationManager controls authentication.

4.1 Inhome

For inhome appliances, our approach is based on the classical single server OSGi architecture (cf. Fig.1). The OSGi server allows to easily plug various devices into the private control network and additionally serves as a central authentication database and a gateway to the Internet.

We have chosen the single server architecture since it reduces administration overhead and avoids the management of distributed data for user access and profile management, for instance. Considering a distributed P2P architecture for Smart Homes exclusively does not make sense, since granting and mainly revoking access rights causes the mentioned overhead. So high-level P2P applications in VHEs just add up if not security critical areas are considered. Due to this we used JXTA in a different way:

P2P also extends the basic OSGI architecture by peer group control and peer resource sharing by introducing the additional VHE Virtual Device service as given in Fig. 2. Thus, devices can be divided into several groups with respect to function, location, and users. Each of these groups is plugged into the OSGi-server as one *Virtual Device Service*. Via the access through virtual devices, the user is able to control all devices of one group as one service. Possible groups for virtual services are:

- rooms (kitchen, living room, bath room etc.),
- age (kids, teens, adults etc.)
- monitoring devices (aircondition, house security etc.)
- communication devices (mobile phone, fax, email, pager etc.)
- entertainment devices (TV, VCR, HIFI, DVD etc.)
- security management (camera, lights, sensors, communication etc.)
- energy management (gas, electricity etc.)

Each device typically belongs to more than one group so that groups can overlap. The VHE Virtual Device service also includes the adaption of user interfaces with respect to the individual device profile. User Interfaces are individually generated for each class of devices with respect to display size and input/output properties [12].

In addition to the above mechanisms, P2P services allow additional advanced applications due to their extremely high flexibility in management of distributed resources. Consider, for instance, a Home Video Storage Network with Quality of Service support [7]. The individual distributed video Peers are used for streaming and storing videos where single video processing computers are connected to one virtual server via routers and switches as it is shown in Fig. 4.

The management for the virtual server can be implemented by JXTA-based middleware which ensures (a) *Quality of Service* for video streaming and manages (b) the (redundant) storage and distribution of recorded videos. Each peer offers the download of videos and play out of videos by streaming simultaneously. Due to the limited bandwidth, simultaneous streams can easily lead to network overloads so that additional request may not be provided with the required Quality of Service. Whereas the bandwidth assignment is managed by an additional admission control, the existing P2P rendezvous server can take over the flexible distribution of streams with respect to the available bandwidth so that the rendezvous server (with QoS-Admission Control instance) can offer the peers to reroute the video stream on demand to a destination in a subnetwork with more available bandwidth. Additional caching and distribution of copies is possible for videos with high frequent access rates. The virtual video server can be accessed as one service through an OSGI sever which may run on top of the routers or on one of the peers. Considering the underlying advanced streaming technology, the user has to deal with one virtual video server only and does not need to consider any network details.

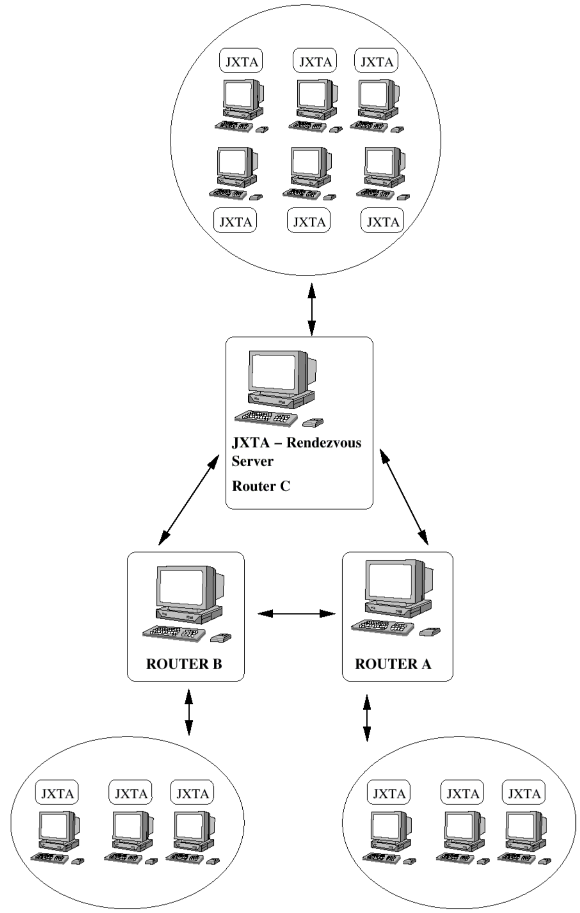

Figure 4: Video Storage Home Network

4.2 Interhome

Interhome networks cover networks between two or more parties (e.g., families or buildings) for distributed control, monitoring, resource sharing and communication. However, in home applications the focus is more often on messaging and entertainment, e.g., children of same age of different families join for video gaming or chat.

For an interhome architecture we have to combine the single server-based inhome networks to a multi-server network. The connection between those networks is established by the VHE P2P Communication Service captured by an OSGi service which additionally has to host the P2P rendezvous server. Security is provided by an additional VHE Certificate Manager service.

4.2.1 Interhome Communication

The *VHE P2P Communication service* provides P2P-based communication and control between OSGi servers for P2P enabled devices based on JXTA technology (cf. Fig. 5). Whereas local devices are still managed via the DeviceAccessService, signals and resources of P2P enabled devices are made available via the P2P communication service. Privacy and Security are ensured through information and signal filtering inn conjunction with the CertificateManager service for authentification.

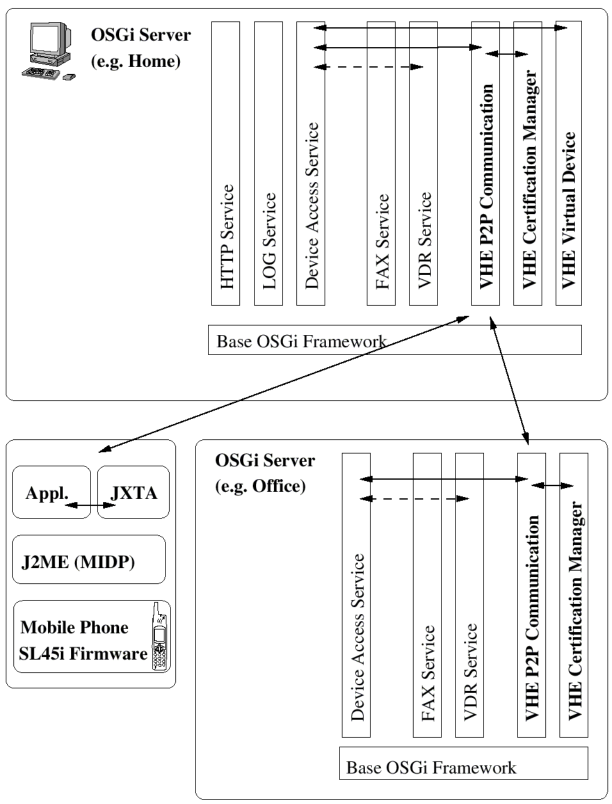

Figure 5: OSGi Architecture with Integrated JXTA

For interhome connections through P2P we can seamlessly migrate from inhome to interhome application, i.e., developing from a home environment to a true virtual home environment such as it is given in Fig. 6. Provided the user has permission to access through the VHE Certificate Manger, all device connected peers can easily access all services from external and internal connections as it will be outlined in the next subsection. This architecture additionally enables full control of inhouse equipment over wider distances through mobile phones via GSM.

4.2.2 Interhome Security

In network-based home environments, security is of utmost importance. We focus for interhome security on the authentication of other servers as peers. We have to consider additionally high level encryption. Low-level protocols like Bluetooth are not considered since most of them comprise their own security mechanisms, e.g. WEP, known MAC addresses, etc.

Security for local and wide area home networks has to include a meaningful combination of symetric/asymetric encryption, message-hashing, and digital signatures. The most important security issues are:

- **confidentiality**: nobody else is permitted to listen to data transmissions. This is necessary since confidential and private data might be exchanged between the peers. Secure transmission can be achieved by channel encryption. Provided transmission is based on IP, latest encryption standard is TLSv1 which is supported by the JXTA API.

- **data integrity**: it ensures that nobody has modified the transmitted data. This is achieved by hash functions and CRC-codes creating message fingerprints.

- **authentification**: is provided by the VHE Certificate Manager service in our framework and ensures that a peer is the one it identifies itself. We implement the authentification by a classical challenge-response transaction: Peer A sends a random number to peer B; peer B signs this number and sends back the signature. Peer A can now verify the signature using peer B's public key which is content of the certificate. Since the certificate has been signed by a trustful person peer A can now trust peer B.

Authentication is the most important but also one of the most difficult issues when dealing with peer-to-peer computing. Here it is necessary that every peer gets a *certificate* which allows the verification of the identification. For virtual home applications low level authentification by individual protocols like in Bluetooth is not sufficient since additional peer dependent and peer group dependent access rights have to be managed [3, 10].

Since we are dealing with highly distributed applications with high security requirements, the VHE

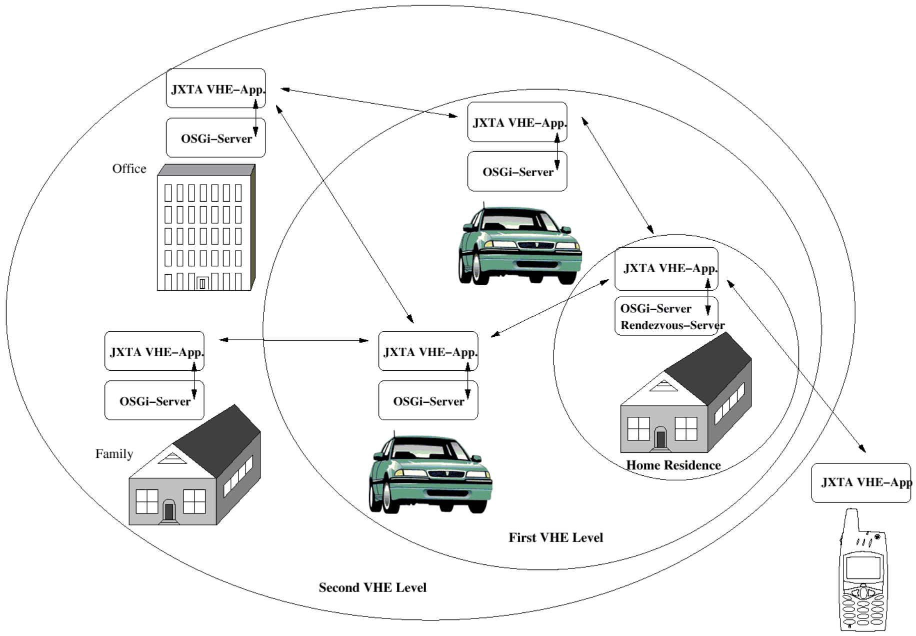

Figure 6: Different Access Levels in Virtual Home Environments

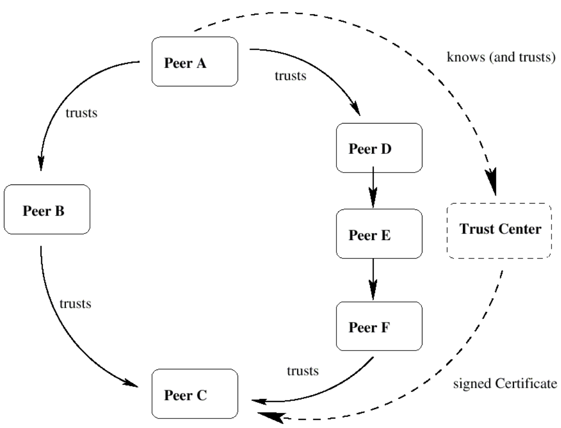

Figure 7: Web of Trust for Multiple Peers

Certificate Manager combines a distributed *web of trust* certification, which is similar to the PGP/GPG structure, with hierarchical certificate structures via using *Certification Authorities (CA)* as they are known from the SSL-protocol.

For a Web of Trust authentication [13] for multiple peers, we assume that each peer has a unique peer ID created by JXTA and each peer has a public certificate which contains the peer ID and its public key. In a distributed *web of trust* several people with PDAs, mobile phones or laptops occasionally meet and exchange data. Some people/devices know each other, some do not. Devices who trust each other, confirm this confidence by signing the other persons peer-to-peer devices digitally.

Fig. 7 gives an example with 6 peers and a virtual trust center. It shows, for example, two interconnections from peer A to peer C. When A can trust B and B trust C, then A can also trust C. Through semi-transitive transformation a validation chain can be created. The route between two parties should be as short as possible. The longer the route from peer A to peer X (of a personally unknown user) the more unsecure the ID of device X might be.

To increase security, additional *Certification Authorities (CA)* also called *Trust Center (TC)* are integrated. In that case, a peer receives a signed certificate by a CA. Other peers have a list of certificates of several CAs. Every peer certificate signed by a CA is considered to be reliable. In our scenario the Home Residence Area just allows Peer Access of Peers with Certificates signed by the Home itself.

In Fig. 7 the dashed lines describe the authentication architecture with the use of a TC. The applicability of that heavily relies on the computation of digital signatures with large numbers (i.e., 2048 - 4096 bit). The main bottleneck for application is the limited computing power of portable devices like mobile phones, though next generation devices easily overcome that problem.

5 Summary and Conclusions

We have introduced an approach to combine single server OSGi-based smart homes to multi-server virtual home environments (VHE) with JXTA's peer-to-peer for inhome and interhome applications. For the seamless integration, we have introduced two additional OSGi services: VHE Virtual Device, VHE P2P Communication. The latter manages communication and access between servers where the first one provides advanced management of devices and combines them to a virtual device. An additional VHE Certificate Manager service manages authentication for the individual connection.

First implementations have shown that P2P technology can not only be applied to resource sharing, device control, and communication. Really attractive applications can be implemented by extending the concept of a rendezvous server as it was shown by the advanced management of resources for video streaming.

Our current experience with the presented technology and users is that security is of utmost importance in order to achieve a wider acceptance. Though our present framework already integrates security by authentication, we see the necessity for additional advanced access control by means of smart card technology. Smart cards can be applied here not just as a security token but may additionally identify the user through biometrics as the owner of the card. Moreover, multi-functional smart cards can be used for storing user profiles and keeping the current state of the home environment.

5.1 Acknowledgements

The work described herein is funded by German Ministry of Technology and Research through the DLR as the *Virtual Home Environments Middleware* project.

References

[1] 3rd Generation Partnership Project (3GPP). *The Virtual Home Environment*, Oct 2000.

[2] E. Aarts, editor. *Ambient Intelligence in Homelab.* Royal Philips Electronics, 2002.

[3] C. Candolin. *Security Issues for Wearable Computing and Bluetooth Technology.* Telecommunications Software and Multimedia Laboratory, Helsinki University of Technology, Finland, 2000.

[4] K. Chen. *Programming Open Service Gateways with Java Embedded Server Technology.* Addison-Wesley, 2001.

[5] Information Technology for European Advancement (ITEA). *Middleware for Virtual Home Environments.* `http://www.vhe-middleware.org`.

[6] Lloyd, Hayzelden, and Cuthbert. Virtual home environments to be negotiated by a multi-agent system. In *Impact of Agent Technology on Telecommunication Workshop (IMPACT99), Seattle*, Dec 1999.

[7] C. Loeser, P. Altenbernd, M. Ditze, and W. Mueller. Distributed video on demand services on peer to peer basis. In *Proceedings of the First International Workshop on Real-Time LANs in the Internet Age (RTLIA2002)*, June 2002.

[8] C. Loeser and W. Mueller. Resource sharing management with peer-to-peer connections in home environments. In *Proceedings of the International ITEA Workshop on Virtual Home Environments*, feb 2002.

[9] M. Miller. *Discovering P2P.* Sybex, 2001.

[10] R. Morrow. *Bluetooth Operation and Use.* McGraw-Hill Professional, 2002.

[11] Open Services Gateway Initiative. *OSGi.* `http://www.osgi.org`.

[12] I. Roussaki, H. Jormakka, S. Xynogalas, A. Laikari, M. Chantzara, and M. Anagnostou. Multi-terminal and multi-network access to virtual home environment. In *Proceedings of IST Mobile and Wireless Telecommunications*, June 2002.

[13] W. Stallings. *Cryptography and Network Security.* Prentice Hall, 2003. ISBN: 0-13-091429-0.

[14] Sun Microsystems. *Java Embedded Server (JES).* `http://wwws.sun.com/software/embeddedserver`.

[15] Sun Microsystems. *Project JXTA.* `http://www.jxta.org`.

[16] B. Traversat, M. Abdelaziz, M. Duigou, J.-C. Hugley, E. Pouyoul, and B. Yeager. *Project JXTA Virtual Network.* Sun Microsystems, Feb 2002.

[17] E. Turcan. Peering the smart homes. In *Proceedings of the First International Conference on Peer-to-Peer Computing(P2P'01)*, Aug 2001.