Veikko Pankakoski

Experimental Design for a Next Generation Residential Gateway

Faculty of Electronics, Communications and Automation

Thesis submitted for examination for the degree of Master of Science in Technology.

Helsinki 29.11.2010

Thesis supervisor:

Prof. Jörg Ott

Thesis instructor:

Dr. Henrik Lundgren

Aalto University
School of Science
and Technology

Aalto University
School of Science and Technology

Abstract of the
Master's Thesis

Author: Veikko Pankakoski

Title: Experimental Design for a Next Generation Residential Gateway

Date: 29.11.2010 Language: English Number of pages: 20+90

Faculty of Electronics, Communications and Automation
Department of Communications and Networking
Professorship: Networking Technology Code: S-38

Supervisor: Prof. Jörg Ott
Instructor: Dr. Henrik Lundgren

Today, over half of the European homes have a broadband Internet connection. Typically, this connection is enabled through a residential gateway device at the users' premises. In addition to facilitating triple play services, this gateway also forms the core of users' home networks by connecting their network-enabled devices. While the number and the size of such home networks keep on increasing, three major problems can be identified in current systems. First, home network management is getting increasingly complex, and a growing number of networking technologies and connected devices must be supported and managed. Second, content management has become difficult. Users are generating an increasing amount of content and this content is stored (and sometimes shared) in an almost anarchical manner across different home network devices as well as online. Third, new network-enabled services, such as e-health systems, are emerging, but are typically poorly integrated into existing home networks. There is a clear need for home networking solutions that address these problems.

In this thesis, we adopt a gateway-centric approach to address these problems in a unified manner. We concretise the requirements for a next generation residential gateway by analysing a set of future home networking use cases. These requirements serve as input to our gateway system design. In summary, our design includes the following main components. (i) A residential gateway architecture based on virtualization. This enables new features and new ways to implement the other components of our design. (ii) A gateway-based mechanism to set up community networks between different home networks. (iii) A distributed file system to establish community networks and to enable improved content management and sharing. (iv) Mechanisms for visiting gateway users to utilize other users' gateway resources. We implement these core functionalities and develop a proof-of-concept prototype. We successfully validate our prototype through use case driven testbed experiments. Finally, we believe that the insights gained from this study and the prototype implementation are important overall contributions that can be used in the future research to further explore the limitations and opportunities of this gateway-centric approach.

Keywords: residential gateways, virtualization, peer-to-peer

Aalto-yliopisto
Teknillinen korkeakoulu

diplomityön
tiivistelmä

Tekijä: Veikko Pankakoski

Työn nimi: Experimental Design for a Next Generation Residential Gateway

Päivämäärä: 29.11.2010 Kieli: Englanti Sivumäärä:20+90

Elektroniikan, tietoliikenteen ja automaation tiedekunta

Tietoliikenne- ja tietoverkkotekniikan laitos

Professuuri: Tietoverkkotekniikka Koodi: S-38

Valvoja: Prof. Jörg Ott

Ohjaaja: TkT Henrik Lundgren

Puolella eurooppalaisista kotitalouksista on laajakaistaliittymä. Yleensä käyttäjä kytkeytyy ulkoiseen verkkoon kotireitittimen avulla (residential gateway). Internet-yhteyden ja IP-perustaisten palveluiden kuten VoIP- ja IPTV-palveluiden lisäksi kotireititin muodostaa kotiverkon ytimen kodinverkkolaitteiden liittyessä siihen. Kotiverkkojen lukumäärän ja koon kasvun seurauksena kotiverkoissa voidaan tunnistaa kolme ongelmaa. Ensinnäkin kotiverkkojen hallinta on haastavaa kotiverkossa tuettavien verkkotekniikoiden ja laitteiden määrän kasvaessa. Toiseksi sisällönhallinta on monimutkaistunut käyttäjien luodessa ja kuluttaessa yhä enemmän sisältöä. Kolmanneksi uudet verkkoperustaiset tekniikat kuten sähköisen terveydenhuollon ratkaisut (e-health) integroituvat usein heikosti olemassa olevien kotiverkkolaitteiden kanssa.

Tässä diplomityössä edellä mainittuihin ongelmiin pyritään löytämään yhtenäinen ratkaisu kotireititintä apuna käyttäen. Työssä analysoidaan uudentyyppisen kotireitittimen vaatimuksia käyttämällä hyväksi joukkoa käyttötapauksia. Vaatimusanalyysin perusteella luodaan malli, joka sisältää seuraavat pääkomponentit. (i) Virtualisointitekniikkaan pohjautuva kotireititinarkkitehtuuri. (ii) kotireititinperustainen mekanismi yhteisöverkostoiden pystyttämiseen kotiverkkojen välillä. (iii) Hajautettu tiedostojärjestelmä yhteisöverkkojen pystyttämiseksi ja parannetun sisällönhallinnan ja sisällön jakamisen mahdollistamiseksi. (iv) Mekanismeja, joiden avulla vierailevat käyttäjät voivat hyödyntää muiden käyttäjien kotireitittimien resursseja. Työssä toteutetaan em. ydintoimintoja laaditun mallin perusteella ja toteutuksen toimivuus verifioidaan käyttötapauksiin perustuvalla testauksella.

Avainsanat: kotireititin, virtualisointi, vertaisverkot

Acknowledgements

The work presented in this thesis was performed in Technicolor Paris Research Lab during 1.3. – 31.8.2010. I want to thank Henrik Lundgren, my supervisor, for being an excellent instructor, sacrificing a lot of time to my work, asking many questions and giving honest feedback. In addition I want to thank Martin May, Fabio Picconi, Augustin Soule and Theodoros Salonidis for commenting and giving ideas for my work. Thanks to Pascal Le Guyadec for giving me the hardware and software that were essential for my work. I also want to thank Marianna Carrera for helping me with the Wi-Fi issues.

I want to thank my parents and sisters for their support during my work. I also want to thank the friends from Eurecom: Thanks to Tobias, Lu, Teemu, Henri, the Norwegians and the Italians and all others. Thanks to the Geneva team, Raquel, Agnetha, Eva, Celia, Airi and others. Thanks to the friends in Finland as well. Thank you Christian and Atte for visiting me during my stay in Paris.

Finally, I want to thank my beloved fiancée, Lauriane.

Helsinki, November 29, 2010.

Veikko Pankakoski

Contents

Abbreviations

ACS	Auto Configuration Server
ADSL	Asymmetric Digital Subscriber Line
AES	Advanced Encryption Standard
API	Application Programming Interface
BACnet	Building Automation and Control Networking Protocol
CA	Certificate Authority
CPE	Customer Premises Equipment
CPU	Central Processing Unit
CWMP	CPE WAN Management Protocol
DALI	Digital Addressable Lighting Interface
DECT	Digital Enhanced Cordless Telecommunications
DHCP	Dynamic Host Configuration Protocol
DLNA	Digital Living Network Alliance
DNS	Domain Name System
DPWS	Device Profile for Web Services
DRY	Don't Repeat Yourself
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer
ETH	Ethernet
FIGARO	Future Internet Gateway-based Architecture of Residential netwOrks
FP7	Seventh Framework Programme
FTP	File Transfer Protocol
FUSE	Filesystem in Userspace
FXS	Foreign eXchange Subscriber
HD	High Definition
HGI	Home Gateway Initiative
HomePNA	Home Phoneline Networking Alliance
HTTP	Hypertext Transfer Protocol
HTTPMU	HTTP Multicast over UDP
I/O	Input/Output
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPTV	IP Television
ISP	Internet Service Provider
ITU	International Telecommunication Union
KNX	Konnex Networks
LAN	Local Area Network
LED	Light-Emitting Diode
LRU	Least Recently Used
MAC	Media Access Control
MIMO	Multiple-Input, Multiple-Output
MoCA	Multimedia over Coax Alliance
MPLS	Multiprotocol Label Switching

MROW	Multiple Read, One Writer
NaDa	Nanodatacenters project
NAS	Network Attached Storage
NAT	Network Address Translation
NIC	Network Interface Card
OASIS	Organization for the Advancement of Structured Information Standards
oBIX	Open Building Information Xchange
OS	Operating System
OSGi	Open Services Gateway Initiative
OSI	Open Systems Interconnection
P2P	Peer-to-Peer
PAN	Personal Area Network
PCMCIA	Personal Computer Memory Card International Association
PGM	Physical Gateway Manager
PHY	Physical Layer
PKI	Public Key Infrastructure
POTS	Plain Old Telephone Service
QoS	Quality of Service
RAM	Random Access Memory
RGW	Residential Gateway
RJ	Registered Jack
SCTP	Stream Control Transmission Protocol
SFP	Small Form-factor Pluggable
SOAP	Simple Object Access Protocol
SIP	Session Initiation Protocol
SLP	Service Location Protocol
SSH	Secure Shell
SSID	Service Set Identifier
SSL	Secure Sockets Layel
STB	Set-Top Box
TAP	Network tap
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TR	Technical Report
TTP	Trusted Third Party
TUN	TUNnel
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
UPnP AV	UPnP Audio and Video
UPnP IGD	UPnP Internet Gateway Device Protocol
URL	Uniform Resource Locator
USB	Universal Serial Bus
VAP	Virtual Access Point
VGM	Virtual Gateway Manager
VIF	Virtualized Interface

VM	Virtual Machine
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
WAN	Wide Area Network
WSDL	Web Service Description Language
WLAN	Wireless Local Area Network
WORM	Write Once Read Many
XML	eXtensible Markup Language

1 Introduction

"The Internet lives where anyone can access it"[96]. Nowadays the Internet is accessed from homes. Often, a broadband connection is used to connect the Internet. In 2009, 56% of the households in the EU area connected to the Internet with a broadband subscription [17]. At the same time, 19.5 million people had a broadband subscription in France, which covered 30% of the country's inhabitants [59]. A broadband connection is typically provided by using a residential gateway, which besides the Internet connectivity, provides users with a home network. Home networks are now utilized by countless of appliances including mobile phones, media centers and even coffee makers [42]. The use of broadband has made it easier and faster for individuals to share, contribute, create and consume information. In addition to broadband, the social aspects of the so called Web 2.0 have expanded the amount of user-generated content shared in the Internet. A popular Web 2.0 site Facebook claims [18] that more than 30 billion pieces of user-generated content is shared each month in their system. The improved household connectivity has not only helped the Web 2.0 services to bloom but has also brought new services to the home domain. So called emerging services, such as e-health and home automation are introduced at homes at increasing pace [13, 6, 61]. For example, the home automation systems market revenues are expected to exceed $11.8 billion in 2015 [3].

Despite the popularity of the home networks, these systems are far from being perfect. Since an increasing number of network devices and services are introduced at homes, the *home network is becoming more and more complex* to manage. When a growing number of devices are dependent on the network, regular home users often find themselves in the position of a network system administrator. According to a study [43], significant household effort is required to coordinate, setup and maintain modern home networks. In many cases "the teenager in the family would have to become full-time unpaid tech support" [8].

At the home domain, the side-effects caused by the invasion of information technology are not limited to the home network management. Modern households are struggling to keep their digitalised content such as music, photos and videos, in order. Photos, for example, can be kept in a memory card of a digital camera or in a hard drive of a laptop. Often the same photos are uploaded to online storage services, such as Flickr [101] and Picasa [26], in order to share and backup them. This has led to the situation where users have to search for the photos in various sources, instead of just browsing a traditional photo album.

While the content and network management are a headache, the emerging services do not ease the situation. On the contrary, these services often increase the complexity by introducing separate devices with their own protocols, user interfaces and networks. Users have to learn how to use a new system that works in an isolation from the existing home network. In recent years, sectors such as health care, energy and building industry have started to introduce Internet Protocol (IP) based services at homes [13, 47, 61]. Despite the interoperability possibilities that the IP

technology provides, separate devices and networks are typically introduced for a single purpose. Furthermore, the compatibility with other sector services remains largely non-existent.

Some solutions have been introduced in order to address some of the previously mentioned problems in the home network. For example, HomeMaestro [87] is a distributed system to manage instrumentation and monitoring of home networks. It automatically identifies competition in the network and allows connection and application coordination in collaborative manner across hosts. When it comes to content management and interoperability between different devices, many of the network-enabled devices now hold an interoperability certificate provided by the Digital Living Network Alliance (DLNA) [14]. Although these solutions are clearly a step in the right direction, they have certain limitations: each home network device has to either fulfill specific interoperability requirements or connect to a dedicated connector. In addition, these solutions address a specific problem. HomeMaestro inproves the home network management and DLNA addresses the content management. In other words, none of these solutions addresses the set of problems in a unified manner. In the home network domain where the devices are typically managed by a single entity, such as a family, a centralized home network coordination architecture can be more flexible and easier to adopt. One potential platform for the centralized system is the residential gateway.

The residential gateway (RGW) is a device situated between the Internet and the home network. The main purpose of the residential gateway is to give the users an access to the Internet. However, it has several unique properties due to its strategic location. For example, a residential gateway interconnects the residential network with the Internet, and aggregates multiple devices and services within the home network. It acts as a natural control point where Internet-based devices pass through connecting with most of the Internet-devices used at home. Furthermore, the residential gateway is nearly as powerful as a PC and one of the few always-on devices at home.

As more than half of the households in Europe have a broadband connection, the residential gateway has become a basic appliance at home, such as the microwave, the television and the computer. While the current residential gateway concentrates on providing households with Internet connectivity, the entity has a lot more potential, which should be explored.

1.1 Problem Statement

In this thesis we aim to find an answer to the following question: what kind of design is required for a residential gateway in order to support

(1) improved home network management,

(2) improved content management and

(3) integration of emerging services.

We investigate how a residential gateway should be designed so that it can be better used for supporting the management of various home network devices. We define how the gateway device can be designed in order to solve problems in the content management, such as how content can be more easily located and shared with others. We also study how a gateway device can be exploited to ease the integration of various emerging services in the home network.

1.2 Objectives

To provide an answer to our problem statement, we set the following four objectives for this thesis. In this thesis we will

1. identify the conceptual requirements for a next generation residential gateway,
2. draft an experimental design for the core components of a next generation residential gateway,
3. implement a proof-of-concept prototype and
4. validate the design and the prototype through use case based tests

1.3 Contributions of This Thesis

We summarize the work and the contributions made in this thesis as follows. This thesis uses the FIGARO project [69] as a reference when defining the concept of the next generation RGW. In essence, the goal of FIGARO is to design a future Internet gateway-based architecture of residential networks. FIGARO plans to introduce RGW-based solutions for content management, network optimization, network management and emerging sector integration.

In this thesis, we concretise the requirements for a next generation residential gateway by analysing FIGARO's vision and future home networking use cases. In this process, we identify several functional and non-functional requirements for a next generation gateway. These includes, but are not limited to, a content and resource sharing system, a unified content management system and the security infrastructure. All identified requirements serve as input to our gateway system design.

The goal of this thesis is to understand the overall concept and essential requirements of a next generation gateway. Thus, we need to identify and analyze a large number of sub-systems. The depth of our analysis depends on the role of the sub-system. The focal sub-systems to the overall system are analyzed in detailed level while other sub-systems are examined in less detail.

In summary, our design includes the following main components.

1. A residential gateway architecture based on hardware virtualization. Although the use of such virtualization of RGW's is not new per se, this enables new features and new ways to implement the other components of our design.

2. A gateway-based mechanism to set up community networks between different home networks.

3. A distributed file system to establish community networks and to enable improved content management and sharing.

4. Mechanisms for visiting gateway users to utilize other users' gateway resources.

In addition, our design enables users to execute and install arbitrary applications at the gateway device.

We implement these core functionalities and develop a proof-of-concept prototype. We successfully validate our prototype through use case driven testbed experiments. Finally, we believe that both the prototype and the insights gained from this study are important overall contributions that can be used in the future research to further explore the limitations and opportunities of this gateway-centric approach.

1.4 Related Work

The services the current residential gateway devices provide are relatively modest. A modern residential gateways typically supported services such as voice and video over IP. On top of this, some of the gateways support functions like third-party Wi-Fi sharing and peer-to-peer applications [22]. However, these applications depend on a particular gateway device and typically users do not have the possibility to choose which services the RGW provides. In other words, the current home networks could benefit from a more flexible RGW device. For example, centralized home network management and content management applications could be introduced in RGW device. Also, because of its strategic location, RGW can be seen as a potential platform candidate for emerging services like e-health and home automation.

To address the problem of inflexibility in RGW devices, an idea of the so called the open service gateway has been proposed [73, 31, 36]. The OSGi framework is introduced as a solution to manage Java based applications on the gateway devices. The OSGi framework provides resource isolation and life cycle management for third-party applications in residential gateways. Recently, the Home Gateway Initiative (HGI), which aims to publish specifications for RGWs, introduced an OSGi-based software execution environment in their Residential Profile Release 3 [66]. Although OSGi provides a way to manage software in the residential gateway, in this thesis, we do not use OSGi. Our goal is to design a flexible system that is not limited to the means that the Java sandboxing provides.

Moore's law has implications for the RGW devices as well, which makes it possible to imagine solutions beyond Java based operating system-level virtualization. The

Nanodatacenters project (NaDa) [56] aims to provide a technology allowing service providers, such as content providers, to get virtual slices from end user's RGW devices. Unlike OSGi, which operates on OS-level virtualization, NaDa provides service providers with a hardware level virtualization. The benefit of the hardware virtualization is that the services are no longer dependent on a single programming language but have the resources of a complete, virtualized hardware device. Whereas NaDa concentrates on providing residential gateway based services to service providers, such as broadcasting companies, in this thesis our target are the home users. Thus, we study the potential of the hardware virtualization in order to address our problem statement.

When it comes to the home network management and monitoring, HomeMaestro can be a potential solution [87]. It relies on the assumption that home networks have a relatively small number of devices. The HomeMaestro system uses advanced algorithms to enforce desired performance in the home network. It defines whether the detected network problems are related to competing traffic flows or the to the applications. Based on this, HomeMaestro and assignes the available bandwidth to the applications through priority-based mechanisms and traffic shaping. Another effort to better address home network management is the Digital Living Network Alliance (DLNA)[14]. The alliance grants certificates to devices that follow DLNA's guidelines and fulfill the interoperability requirements for home network devices. DLNA utilizes Universal Plug and Play (UPnP) for control devices and various other standards in order to set up data transmissions. These standards include MoCA and MPEG4 among others. Even though these solutions solve issues related to device management and interoperability, they do not address the broader set of problems in a unfied manner. In this thesis we try to design a system that can be used to solve issues in multiple domains. We try to address the home network management, the content management and the integration of emerging services.

1.5 Outline

The rest of this thesis is structured as follows. In Chapter 2 we survey the relevant home networking technologies as well as current typical residential gateway characteristics. We introduce the FIGARO project in Chapter 3. The FIGARO project is used as a reference in our requirement analysis, which is performed in Chapter 4. Based on the requirement analysis, an experimental design for a next generation residential gateway is introduced in Chapter 5. The design is evaluated by using a prototype implementation, which is introduced in Chapter 6. The correct functionality of the prototype is verified in Chapter 7 through use case-based testing. Finally, we conclude this thesis in Chapter 8.

2 Background

The home networks contain a large number of new technologies. Transmission media, that have been traditionally used for different purposes, such as electricity lines and TV cables, are now often used for transmitting IP packets. In addition to wired systems, several new wireless standards are being introduced in order to better address the varying needs of the new network devices. As the number of technologies increases, different device discovery and control mechanisms are being introduced. Several mechanisms exist that aim to address the issues related to the device and content management at homes. While more and more home network technologies and services are being introduced, residential gateway manufacturers have a hard task to keep up with the new technologies.

In this chapter, we first provide an overview of different home networking technologies and then discuss the device discovery and control mechanisms. Various current and emerging home network technologies are discussed, which is followed by an insight into the latest features of the residential gateway devices.

2.1 Home Networking

Home networks are implemented using both wire and wireless technologies. The number of different available technologies is large. This number is expected to increase, when network solutions enabling emerging services, such as e-health, are being introduced at homes. In this section, an overview of different home networking technologies is given. Residential gateways are discussed separately in Section 2.2.

2.1.1 Link Layer Technologies on Wires

In modern home networks data is not only transmitted over dedicated twisted pair cables but other transmission paths, which are traditionally used for other purposes, are exploited more and more often. This covers for example coaxial cables, which are normally used for TV and radio transmissions and power lines.

IEEE 802.3, or commonly know as Ethernet is undoubtedly the most adopted packet based link layer technology in local area networks. The origin of Ethernet technology goes back to the 1970s when the development stared in Xerox PARC. The seminar paper representing the technology was published in 1976 [54]. The technology was adopted in a wider scale during the 80s when the standard enabled a data rate of 10 Mbps. Later, the technology has evolved to much higher data rates the latest standards enabling rates as high as 10 Gbps. Various physical transmissions paths including optical fibre are supported by Ethernet standards. However at home, Ethernet is typically used over a twisted pair cable. The broadly used standards are 10Base-T, 100Base-TX and 1000BASE-T that support 10Mbps, 100 Mbps and 1 Gbps rates respectively [81].

Telephone and coaxial cables can be utilized as an alternative physical medium

to enable data transfer at home. The prominent technology in this field is the Ethernet-based HomePNA (Home Phoneline Networking Alliance, also known as HPNA). HomePNA enables networking over existing coaxial cables and telephone wiring and because of that, it is typically used inside apartment houses to share a single broadband access between several households, each having a phone line. The technology is ITU standardized and the latest version of the standard, version 3.1, supports up to 320 Mbps data rates. This makes the technology suitable for entertainment applications, such as IPTV. The frequency band of HomePNA is located between 5.5 to 9.5 MHz, which is above DSL frequencies and below broadcast TV frequencies. Thus, HomePNA can coexist on the same phone or coaxial wires with these services [25].

In addition to HomePNA, MoCA standard provides data transfers over existing coaxial cables. Multimedia over Coax Alliance (MoCA) is an industry group defining specifications for home networking over coaxial cables. MoCA is not an open standard. The standard uses 1 GHz band and the version 1.1 of the technology enables data rates up to 175 Mbps. The new MoCA 2.0 standard promises throughputs from 400 Mbps to 800 Mbps. The standard supports 16 devices in the same network. Both technologies, HomePNA and MoCA are adopted in by the industry. However, according to their annual report, MoCA has increased its market share. Both technologies are present in some of the residential gateways available on markets [55, 51, 1, 35, 2].

The idea of using power wires for data transfer has been around of some decades. In fact, power line communication was used for remote relay control in the 1950s. Nowadays, several options exist for communicating over electricity wires at home. The most adopted technology is provided by HomePlug Alliance. HomePlug defines specifications and standards and provides certificates for networking over electrical wires. The current specification, HomePlug AV, delivers data rates up to 200 Mbps. The specification also defines distribution and data encryption techniques. The next version of the specification, HomePlug AV2 is expected to provide 600 Mbps data rates on MAC layer. Some residential gateways support HomePNA, such as the Freebox of the French operator Free [25, 34].

In addition to the industry originated HomePlug, two major standardization instances, IEEE and ITU, have introduced a standard for data communication over electrical wires. IEEE P1901 supports over 100 Mbps data rates at the physical layer by using transmission frequencies below 100 MHz. The standard is limited to the physical layer and the medium access sub-layer of the data link layer (OSI reference model). The technology has been defined to work with other network protocols, such as bridging via 802.1. Some interoperability with G.hn has been defined as well. The technology has been criticized because of the dual-PHY proposal that can cause interoperability problems. According to the standard's web page the "P1901 will be submitted for approval as an IEEE Standard, effective 30 September 2010" [38, 37].

The ITU standard G.9960, commonly known as G.hn, aims to provide a combined

standard for power lines, coaxial cables and phone lines. The data rate can go up to 1 Gbps. The promoters believe G.hn to become the future universal wired home networking standard, which could be embedded in devices such as television, set-top box and residential gateway. AES encryption provides confidentiality and the authentication and key exchange is done using X.1035 recommendation (a password-authenticated key agreement protocol based on Diffie-Hellman key exchange). Currently, it is believed that G.hn-compliant chips are available during 2010. The G.hn technology is promoted by HomeGrid Forum. Recently, the Broadband Forum and the HomeGrid Forum have created a collaboration agreement in order to support conformance and interoperability. The time will show how IEEE P1091 and G.hn will succeed against HomePlug, and against each other [60, 33, 91].

2.1.2 Wireless Link Layer Technologies

In addition to the link layer technologies on different types of wires, wireless technologies are now a commonplace at homes. Wireless systems do not only cover the popular Wi-Fi standard, but new standards that enable services such as home automation and sensor networks are emerging at homes.

IEEE 802.11, being the wireless correspondent for Ethernet, is the primary wireless local area network (WLAN) standard used at homes. Several versions of the standard have be introduced during its lifetime. The first successful version was 802.11b standard, which operates on 2.4 GHz frequency and provides bit rates up to 11 Mbps. Nowadays, the most used version is 802.11g, which has largely replaced the older 802.11b standard. 802.11g works on 2,4 GHz frequency providing 54 Mbps bit rate. It is fully compatible with the older 802.11b standards. 802.11 standard family has been extended with various amendments such as 802.11i, which introduces a replacement for the earlier WEP-based security specification that is broken. One of the latest amendments is the 802.11n standard, which provides improved performance compared with 802.11g standard. In practice, 802.11g will provide bit rates between 100 and 200 Mbps. The amendment also has a support for MIMO-technology (multiple-input, multiple-output) where multiple antennas are channels are used simultaneously in order to improve the throughput. 802.11n is present in some of the modern residential gateways [100, 10, 23].

Another so called home area network standard is the IEEE 802.15.4, also known as ZigBee. The goal of ZigBee is to provide a low-power short distance standard to connect small devices. The maximum distance between devices is approximately 100 meters and the throughput is between 20 to 250 kbps. A ZigBee creates a mesh network that can support several thousand devices. ZigBee saves energy by utilizing a protocol that provides long sleeping times and short connectivity delay. The IEEE 802.15.4 standard was published in 2003 as a solution to some issues that IEEE 802.11 and BlueTooth could not properly address. One of the uses for IEEE 802.15.4 is home automation. IEEE 802.15.4 can be used for controlling lighting, heating and air conditioning. Another domain for the technology is sensor networks. ZigBee is utilized for example in automatic meter reading systems [25, 79].

Z-Wave is a closed protocol very similar to ZigBee. As ZigBee, Z-Wave is suitable for home automation, such as lighting. Z-Wave technology is supported by Z-Wave Alliance, which has over 160 manufacturer members. Z-Wave has a connection limit up to 30 meters and the throughput is between 9.6 to 40 kbps. Z-Wave is reaching the status of a de-facto standard. The Z-Wave application covers, for example, remote home management, energy conservation, home safety and entertainment. As in the case of previously discussed IEEE P1091 and G.hn, it is hard to say, which one of these standards, Z-Wave of ZigBee will eventually dominate the markets. Nevertheless, it seems reasonable to investigate the possibility of adding an option for these technologies in a residential gateway in order to better support emerging sectors [79, 103].

The most common Personal Area Network standard (PAN) is BlueTooth. BlueTooth is an open standard for wireless short distance communication. It operates on 2.4 GHz frequency and provides 432.6 kbps bit rates in symmetric transfer and 721 Kpbs and 57.6 kbps in asymmetric transfer. Bluetooth technology supports device authentication and data encryption. The Bluetooth standard supports a network of eight different devices. However, the specification allows network chaining that enables larger networks. The Bluetooth system is divided into three parts which are radio, link controller and link manager. BlueTooth technology is widely adopted in various home devices, such as laptops and mobile phones. Being a technology present in a large amount of consumer devices, it is likely that some services could benefit from a BlueTooth-enabled residential gateway [81].

2.1.3 Device Discovery and Control

The problem related to device and content management at homes has been addressed by introducing device and content discovery protocols [93, 5, 28, 80, 58]. For example, Universal-Plug-and-Play (UPnP) and Bonjour provide mechanisms to make home appliances communicate with each other. Service Location Protocol (SLP), Java-based Jini and Devices Profile for Web Services (DPWS) are other protocols which attempt to solve the same challenges. Often at the home domain the challenges are not only limited to the device and content discovery, but mechanisms are required to make two devices to use right protocols for actual data transmission after the device discovery. To solve this, industry-based standardization efforts, such as Digital Living Network Alliance (DLNA), have been established [14].

UPnP protocol family enables communication between UPnP enabled appliances. UPnP is a standardized application layer communication protocol set which supports the so called zero-configuration where devices can enter and leave the network without a separate installation process. A UPnP-enabled device is aware of its capabilities and able to informing other devices about its own features. UPnP uses Web Services over multicased UDP (HTTPMU). The standard defines two high level instances: devices and control points. Devices send events, support a web interface and respond to action requests invoked by control points. Control points discover devices, invoke actions and subscribe to event notifications of devices [94].

The problem with UPnP is that the interaction between devices occurs in isolation via the control point. The control point manages devices separately and individual devices do not interact directly with each other. The Universal Plug-and-Play Audio and Video (UPnP AV) makes direct data transfers between two devices possible. The control point configures the devices and triggers the flow content between the devices. Any protocol and data type understood by both devices can be used when transmitting data. A typical user scenario could be a video player transmitting video content to a remote display. In UPnP AV the source of media content is called MediaServer and the sink of the content is called MediaRenderer. These instances do not communicate directly together using UPnP but via the control point [94].

Even though UPnP AV solves the issue of direct communication between devices, interoperability is not necessarily guaranteed. For example, the devices might not use the same sound of video codecs. DLNA aims to solve this issue. The alliance grants certificates to devices that follow DLNA's guidelines and fulfill the interoperability requirements. The DLNA guidelines define the usage of a large set of standards, such as UPnP AV, MoCA and MPEG4 [14].

In addition to UPnP, which is a Microsoft-originated protocol family, Apple Inc. has developed a similar device and service discovery protocol called Bonjour. As pure UPnP, Bonjour works only in a single broadcast domain, so only the services available in the same LAN can be discovered. Bonjour is used, for example, in content sharing and printer discovery [5].

SLP is another zero-configuration type of protocol for LAN environments. The protocol is defined in RFC 2608 [28] and 3224 [27] documents. SLP, which use multicast messaging over UDP or TCP, is used in LAN printers, for example. The protocol defines three types of roles for a device: user agent, service agent and directory agent. User agent is a device searching for services in a LAN. Service agent is a device that provides services for other devices. Directory agent is used in large networks to cache services.

Jini [80] is a Java programming language based middleware that provides means to discover and utilize services provided by other Jini-enabled instances in the network. In Jini each application exposes its services through an API which other application can use. A device is described as a Java object and it is used through Java's Remote Method Invocation (RMI) mechanism which is Java's correspondance to the remote procedure call. Jini provides two type of services: lookup services and discovery services. A lookup service is used by clients to learn about Jini services present at the network. A device uploads a serialized Java-object to the lookup service which describes the provided services. Discovery service lets devices to get contact to the right lookup service which can satisfy the client's request. Jini, which was originally developed by Sun, is now under the control of Apache's River project [4].

DPWS is a web services-based mechanism to enable interoperation between different network devices [58]. It is orgiginally developed by Microsoft and if now a standard of OASIS [104]. It defines the requirements that a device must implement in order

to guarantee the compability with other web services-based systems. In DPWS the devices announce their services by multicasting hello messages periodically. Clients can also search services by sending probing messages which define the type and scope of the service being searched. A service description, which contains details such as a name and a manufacturer of the device, is send on request to clients. DPWS has also a mechanism that allows services to subscribe to events.

There exists many different service discovery and zero-configuration protocols. When it comes to consumer electronic, UPnP/DLNA seems to have a prominent influence on home appliance markets. An increasing number of high-end entertainment devices are DLNA certified. Thus, in a residential gateway based home network management, the adaptation of DLNA's guidelines is justified. Some of the residential gateways available on the markets hold a DLNA certificate [88]. The Bonjour technology, SLP, Jini and DPWS have their users and there certainly exists use cases where these technologies could be utilized in the context of a residential gateway. However, in this work the focus is on UPnP.

2.1.4 Emerging Services

The so called emerging services, such as health care and home automation, have started to introduce services at homes. These services typically utilize their own proprietary protocols. Recently, there has been a movement towards open and standardized technologies. For example, Web Services are being used in building automation. In health care, IEEE is acting as a standardization body for the so called e-health systems [61, 13].

Various standards are present in the building automation domain. These include KNX, BACnet and oBIX. Some other standards exist as well, such as LonWorks, DALI and OpenTerm. KNX is an OSI-based protocol designed for building automation which supports several data transmission paths including twisted pair wiring, power line networking, radio, infrared and Ethernet (also known as KNXnet/IP). KNX supports also IP tunneling. Building Automation and Control Networks (BACnet) is a network building automation protocol supporting e.g. ZigBee, IEEE 802.3, UDP/IP and Web Services. Several features, such as security (AES ciphering) are included in the protocol. OBIX is an OASIS standard for Web Services based interfaces to building control systems. The standard contains an extensible XML specification for building-based control systems, such as access control, lighting and security [6, 44, 61, 50, 105].

It is difficult to estimate, at what scale building automation will be adopted at homes in the future. Even harder is to estimate which standards turn out to be the most prominent. When designing a residential gateway that addresses the needs of the emerging sectors the system architecture should be designed to be as flexible as possible. In other words, the architecture should allow an easy adaptation of current and future protocols.

Health care represents a sector that sees potential in the home network domain.

While the so called Post-World War II baby boom generation gets older, the health care sector is expected to face economic pressure. E-health is seen as an opportunity that could reduce costs. One instance working on this field is the Continua Health Alliance. It was founded in 2006 to improve the interoperability of measurement devices. The alliance now has over 200 organization members. Continua aims to define guidelines and standards and arrange test events for new products. The standards developed by Continua are defined under IEEE 11703 Personal Health Device Work Group. These standards include the description of the overview of the protocol family, device specifications and the description of the Optimized Exchange Protocol, which is used between measure devices and managers [13].

The members of Continua introduced IEEE 11703 based solutions for home users. In these settings, sensor data is first sent to a dedicated home hub box device that is connected to the Internet. A doctor or other analysing instance gets the e-health data via this device. Short distance protocols such as ZigBee are utilized between the home hub and sensor devices. It is possible to imagine that an advanced and flexible residential gateway architecture could remove the need for a dedicated home hub box [67].

2.2 Residential Gateway

A residential gateway (RGW), also known as a home gateway and Customer-premises equipment (CPE), is a device that connects two networks together: the home network to the Internet. A RGW has typically several local network interfaces (LAN) and a single Internet interface (WAN). The behavior of a residential gateway is similar to a router. A RGW makes it possible for several home network devices to share a single Internet access [79].

During the recent years, several other functions have been included in RGW devices to make the usage of home networks easier. In addition, RGWs have enabled for ISPs to emerge in new business areas, such as TV broadcasting and telephony. Often, the RGW device is rented to the user by an ISP, who uses the device to bundle various services under a single contract.

2.2.1 Standardization

Home Gateway Initiative (HGI) is an industrial organization which publishes hardware and software requirements for connected home devices, such as for residential gateways. The members of HGI include many major operators and device manufacturers. According to HGI's web site, "the HGI helps to consolidate the understanding of what is needed, brings additional vendors into the ecosystem, and speeds the evolution from services concept to deployment reality" [30]. The work of HGI is divided into three parts. First, the organization analysis business needs by looking at operator needs and industry trends. Second, HGI defines guidelines. Third, the organization defines test specifications and scenarios in order to match the needs of

the operators [32].

Another actor in the residential gateway domain is the Broadband Forum. Broadband Forum is an organization developing broadband network specifications. It is a coalition of ISPs and device manufacturers. Founded in 1994 as ADSL Forum it later changed its name to DSL Forum and now the organization is known as Broadband Forum. In 2009, IP/MPLS Forum merged with the organization. The TR-069 specification is the main document from the home broadband sector of the organization and it has been adopted for the use with several devices including Set-Top Boxes (STBs) and Network Attached Storage (NAS) units. Several other technical reports are also provided. For example, TR-111 defines how to apply remote management of home networking devices and TR-135 defines a data model for TR-069 enabled set-top box [82].

On the cable modem side, CableLabs is a major consortium which publishes specifications. CableLabs, which was founded in 1988, is a non-profit organization which has several industrial members. The consortium does the same tasks than HGI. It publishes requirement specifications for cable technology. In important specification effort of CableLabs is the Data Over Cable Service Interface Specification (DOCSIS), which defines how an Internet access can be provided over hybrid fibre-coaxial cable [9].

2.2.2 Hardware

So far the requirements for the hardware in a residential gateway have been relatively modest [62]. However, the new services and the increasing amount of different interfaces force the RGW manufacturers to introduce more and more powerful systems [40]. The constraints of a RGW, such as size and noise, makes performance improvement somewhat challenging. This is due to the fact that components typical for a normal PC system, such as fans, cannot be utilized.

A block diagram of a residential gateway is illustrated in Figure 1. The figure, which is based on the view of HGI, presents the main hardware modules present at a typical RGW nowadays. These include:

- *FXS interfaces.* Foreign eXchange Subscriber is a common interface to a standard plain old telephone service (POTS) phone.
- *DECT interface.* Digital Enhanced Cordless Telecommunications interface is used by short-range wireless communication devices, such as cordless telephones and fax.
- *WLAN interface.* WLAN interface provides an access point for IEEE 802.11-based communication devices.
- *Ethernet interfaces.* The IEEE 802.3 interface enables Ethernet devices to access the Internet and communicate with each other.

- *USB interfaces.* Universal Serial Bus interface can be used for Internet access or it can be additionally used for other services, such for printing servers or NAS applications.
- *DSL interface.* The Digital Subscriber Line interface is used for connecting the RGW to a nearest DSL Access Multiplexer (DSLAM), which typically provides the RGW an Internet access.
- *LED Row.* Light-Emitting Diode row is used for providing feedback to the user.
- *Internal power supply.* The internal power supply uses a voltage regulator to regulate the voltage levels coming from a power-distribution network.
- *Memory.* RAM and Flash memory technologies are used to enable the gateway services.
- *CPU Subsystem.* The CPU subsystem contains the logical hardware unit of the residential gateway.

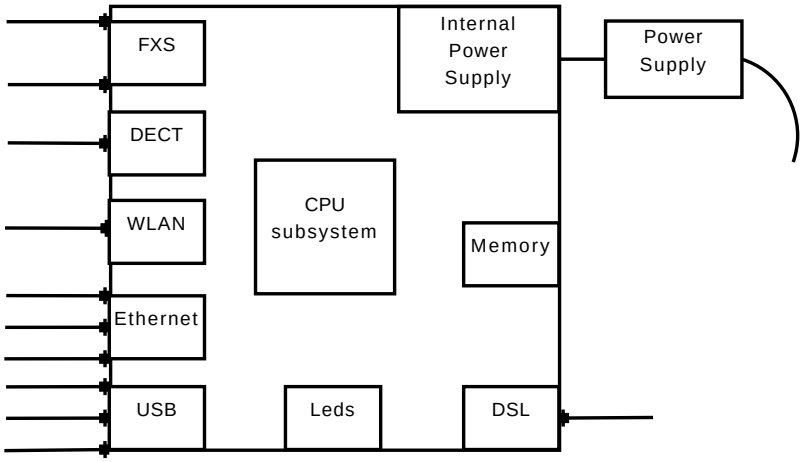

Figure 1: Residential Gateway Architecture

In addition to these, The HGI Phase 3 and 3+ define a large set of requirements for a residential gateway that extend the amount of hardware modules. These widen the system to support for example dual WAN and fiber connections. Also, support for power line communications is included.

The technical report 124 (TR-124) from the Broadband Forum defines functional requirements for broadband residential gateway devices. It contains a superset of the requirements that enable the support for a full suite of voice, data, broadcast video, video on demand and two-way video application in broadband networks. In addition, TR-124 defines the general requirements that assume at least one embedded WAN interface, routing, bridging, a firewall, one or multiple LAN interfaces and home networking functionality [21].

2.2.3 Services

The service palette of a typical RGW is nowadays large. Previously the primary services were a *DHCP service*, which provided each home network device with a dynamic IP address, and the network address translation (NAT), which enabled the sharing of a single Internet connection between all home users. At the same time, NAT functioned as a firewall that protected home users against attacks coming from the public network. To provide a mechanism to configure the gateway manually, a web interface was introduced.

The amount of applications has increased gradually thereafter. *Internet Gateway Device Protocol* (IGD) was added to RGWs to allow the server and peer-to-peer applications to function properly behind NAT [41]. With the aid of IGD, applications, such as Skype, can configure the residential gateway's firewall with port forwarding rules to direct traffic from gateway's public port to a private IP address and port at the home network. UPnP protocol is used for communication [78].

Running servers behind a residential gateway led to the need to have static URL for the gateway device. Since in many cases, ISPs allocate dynamic IP addresses to their clients, a mechanism was required to update DNS servers dynamically. To solve this, a *dynamic DNS update* functionality to provide DNS servers with the latest IP was introduced. A typical RGW supports nowadays IP address updates with services such as DynDNS [16].

Another server application for remote access is *VPN*. Virtual Private Network technology is used for connecting hosts to a local area network over the public network. VPN can be also used for bridging two LANs together over the Internet. Many of the current RGW include VPN software, such as OpenVPN, to provide users with an access to their local network.

Quality-of-Service mechanisms (QoS) were also introduced on RGWs to address issues related to increased amounts of traffic. For example, RGW can be configured to shape traffic to better support online gaming, Voice-over-IP (VoIP) and video streaming. RGWs can be also used for connecting devices to the network, that has only and *USB interface*. This makes it possible, for example, to share printing services and external hard drive devices.

Some of the residential gateways provide an Internet access point for external users. One of the popular actors in this area is the Spanish FON Technology S.L. which aims to create a worldwide Wi-Fi access point network. A FON access point directs a visiting user to a web based *captive portal*, where authentication and possible payment is performed in order to use the Internet access. Similar mechanisms are implemented inside many residential gateway devices [22].

Several other functions can be found in modern gateways as well. Some gateways include a *BitTorrent client* which enables constant peer-to-peer downloading. Since a gateway device is typically always on, this saves the user from keeping her computer open during the download process. Also, some gateways are capable of posting *Twitter updates* each time a visiting user utilizes a shared access point [22].

2.2.4 Gateway Firmwares

As the previous section indicates, the amount of different services at RGWs has expanded alongside with the evolution of the Internet. To get the best out of the gateway hardware, some people have changed the original firmware to another one, which supports better the hardware and gives users a wider set of functions. Such firmware is for example OpenWRT, which is a Linux-based distribution supporting a large set of open services [65, 24].

2.2.5 Remote Management

Currently, ISPs are struggling with large amounts of expensive customer center calls. In order to cut the costs, human intervention in residential gateways must be reduced. Modern RGWs are very complex. The functions include multiple WAN interfaces, virtual ports, queue structures, NATting and Wi-Fi with multiple Service Set Identifier (SSID) just to name a few. A configuration process of such device can vary with customers in time and a default configuration file for all cannot be stored in the devices. If the customer modifies the configuration settings locally, service provider needs to be informed in order to keep the consistent view of the settings.

A common standard for managing RGWs is the technical report 69 (TR-069) from the Broadband Forum. TR-069 describes Customer-Premises Equipment (CPE) Wide Area Network (WAN) Management Protocol (CWMP). The remote management is performed by using Auto Configuration Servers (ACS). TR-069 is an application layer protocol using Web Services. SOAP messages are used for performing remote procedure calls on CPEs. SSL/TLS provides authentication, integrity, confidentiality and non-repetition for the application messages [52].

TR-069 is used for auto-configuration and dynamic service provisioning. These functions take place during the initial connection of CPE to the network and during possible re-configuration operations. ACS can for example find out the CPEs device vendor and current software version. Optional tools exist for improved security [52].

Software/firmware images can be downloaded using the protocol. A certain file format for downloaded files is provided. This format provides a digital signature for the file to support integrity for the data transmitted over the network [52].

TR-069 has tools for status and performance monitoring. CPE can publish performance and status related information which ACS can use for monitoring. In addition, CPE can notify the ACS about a change in its status. ACS can also diagnose CPE's connectivity and service issues by using the provided remote procedure calls [52].

2.2.6 Recent Research

Several reseach papers have been published regarding the usage of OSGi technology in residential gateways. Royon et al. suggest using OSGi to virtualize service

gateways in a residential gateway [74]. The authors propose to use OSGi-based software virtualization to decrease the number of gateway boxes that different service providers introduce at homes. By using Java-based OSGi, various service providers can implement their services on a single gateway device. OSGi provides resource isolation and life-cycle management for the service provider's software in a gateway.

Laoutaris et al. introduce a similar idea to provide resources for service providers at the residential gateway through resource isolation [46]. Authors propose to use the residential gateway as an entity in a content distribution network. Instead of using content distribution systems that are centralized, multiple residential gateways, which are located to the edge of the Internet, are used for delivering content. This P2P-based system benefits from the fact that the residential gateway is often owned by an ISP who rents the device to the end-user. An ISP can control and manage the residential gateway remotely and provide a distributed platform for third-party service providers, which they can use to introduce services. One such a service could be a video service that transfers its most popular content closer to the consumers, in this case to the close-by gateways. The authors suggest to use virtualization in resource isolation. NaDa project, which attempts design the system proposed in the paper, is using hardware virtualization in residential gateways [57].

Some ideas to connect home networks to cloud computing systems have been presented as well. These papers [97, 102] introduce a conceptual model that could be used in order to connect the home devices with cloud services, such as Amazon EC2 and Blue Cloud by IBM. In this model each home network device is virtualized on the residential gateway device and then controlled through the gateway. Tasks such as device management and coordination, remote control and media translation are performed at the gateway by using these virtualized devices. Services located to external clouds can be utilized through the same interface. These ideas are on early state and thus presented on a relatively high level.

Some researchers see the residential gateway as a platform that could be utilized to extend single hop protocols, such as UPnP, to function over the Internet. Haber et al. [29] propose that devices that exist in remote LAN networks could be virtualized at the residential gateway and then accessed from the local LAN by using a single hop protocol, such as UPnP. The remote devices appear as local network devices to the users of the local network. SIP protocol is utilized to enable the communication over the public network.

2.3 Summary

The number of different technologies in home networks is getting bigger and bigger. The network technologies found at homes are not limited to the traditional Ethernet and Wi-Fi standards, but expand to cover other technologies, such as ZigBee and HomePlug. In addition to this, new sectors, such as health care and home automation are emerging in homes.

A residential gateway, originally being a single network translation device, has

evolved to a complex network device. The amount of supported services is increasing which has driven some users to update their gateways in order to better utilize the hidden potential.

3 Residential Gateway Based Future Internet Architecture

Against the previously discussed current state of the home networks and residential gateways, we provide an outlook for the potential future. This prediction is performed through the vision of the FIGARO project, which aims to address the challenges of the current home networks. The content of this chapter is based on the project proposal of the FIGARO project [69].

Future Internet Gateway-based Architecture of Residential Networks (FIGARO) is a large-scale integrating project which aims to design a Future Internet gateway-based architecture centered on federated residential networking. The project is funded by the Seventh Framework Programme (FP7)) of the European Union (EU). "(FP7) is the European Union's main instrument for funding research in Europe" [76]. FIGARO has 12 partners from industry and academic sectors.

According to the project proposal of FIGARO, the project aims to:

1. "design a novel content management architecture that enables distributed content backup, search and access",
2. "develop a network optimization framework, leveraging community networks and heterogeneous networks",
3. "deliver a network management artchitecture which includes new network monitorking and real-time troubleshooting techniques" and
4. "explore novel Internet-based communication and service solutions for emerging sectors, such as energy management and e-health care".

The FIGARO project aims to design a Future Internet architecture, which unlike in the current cloud-based centralised solutions, builds upon a distributed architecture where, according to the project's project proposal, "millons of residential networks to deliver content and services of the future". In order to make this possible, several subsystems will be designed.

The FIGARO project will design a residential gateway-based network management framework, which provides network monitoring and troubleshooting mechanisms. This framework will cover both wired and wireless networks. The project will study the utilization of neighborhood federation to provide multiple access network in order to improve Internet communication. In addition, a gateway-based distributed content management systems will be introduced, which enables transparent storage, search and access functions. A mechanism to provide ubiquitous access to the content regardless of the location will be designed. The project will also design solutions that enable integration of emerging sectors to the residential gateway.

3.1 Future Home Network Vision

There are several motivations for the FIGARO project. The success of electronic content and the popularity of social networks of the Internet has led to the situation where a large portion of the content is created at households located at the edge of the Internet. The trend is expected to continue. Data amounts and, as a consequence, the storage capacity requirements, keep increasing. In addition, wireless access to the Internet is expected to become a norm. The Internet will move from the core and technology-centric model to an edge and user centric architecture.

The FIGARO project will design solutions on various domains. To better understand the concept, the project proposal introduces a set of use cases. The following use cases are used in the project proposal for illustrating the project's vision of the Future Internet Architecture. Note that these use cases are utilized in the requirement analysis in Chapter 4.

Use Case 1: Managing Content

> "Alice records a movie of her daughter's performance at a classical music concert. Before she arrives at home, the movie has already been transferred from the digital camera to the family's common storage system."
>
> "Later, when Alice's family is gathered at home, they want to watch the recorded movie together on their TV. The movie can be easily found on their storage system (without knowing exactly on which device it is stored). This movie is streamed wirelessly to the high-definition (HD) TV and automatically adapted to the network performance to maximize the Quality of Experience of the viewers. In the background and completely transparent to Alice's family, the gateway is performing network monitoring and wireless network optimization to ensure the quality of the bandwidth-demanding HD movie stream."

Use Case 2: Visiting Friends

> "In the evening, Alice is visiting some friends and wants to show them the video of her daughter. Since her friends are part of the same Internet community (or social network), she can easily access and view the movie through her friend's gateway. To reduce the download time for Alice, her gateway collaborates with a few neighboring gateways which help to transfer the video using their spare Internet uplink capacity."

Use Case 3: Sharing Content

> "Some friends of Alice's daughter were performing in the same concert. Together with her friends, Alice's daughter is going to prepare

a personal TV program that gathers the best moments of the concert. This program will be made available to the friends and family through their social network."

Use Case 4: Remote Home Access

"Alice is about to leave on a vacation trip to her cottage. She connects to her cottage's home gateway over the Internet and through a simple interface she announces her arrival and starts the heating system of the house."

These use cases are explained and analyzed in detail in Chapter 4 where the requirements analysis is performed.

3.2 Conceptual System Architecture

FIGARO will address the challenges related to the previously represented vision by designing a future Internet architecture based on residential gateways. The project proposal introduces a conceptual system architecture to structure the different divisions of the concept. Figure 2 illustrates the proposed high level conceptual gateway-centric system architecture. The six different subdomains are discussed in detail in the next following sections.

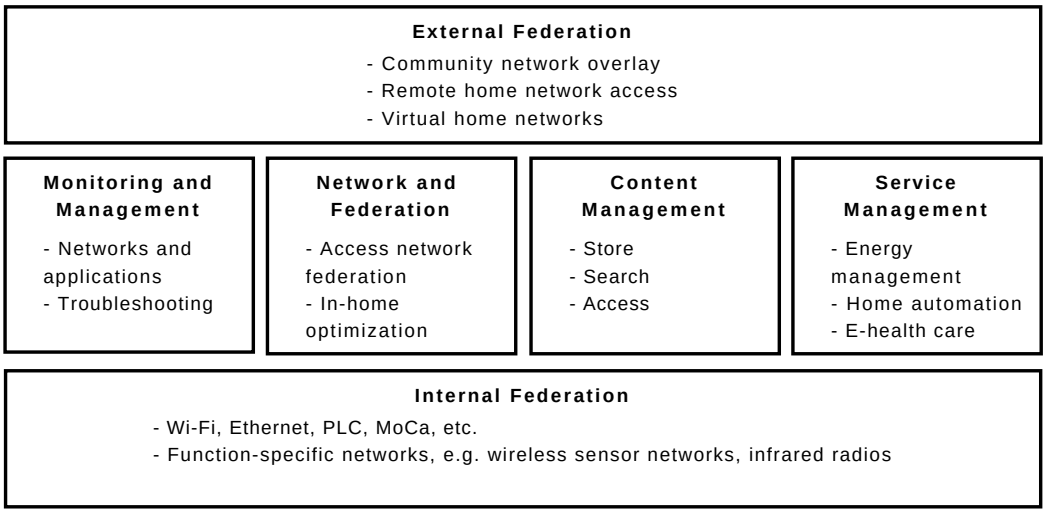

Figure 2: FIGARO architecture.

3.2.1 Federation Approach

FIGARO addresses the user centricity by utilizing the concept of federations. *A federation of networks* is defined as "Two or more independent networks that are

interconnected and operate; at least partly, in a coordinated fashion compose a federation of networks". In addition to this, the term is divided into two sub areas. In *an internal federation*, "a gateway interconnects (and federates) two or more networks within a single residence". To benefit from internal federations, different types of services are accessed in an unified fashion. The second part of a federation of networks concept is *an external federation*, where "multiple gateways interconnect multiple independent residential networks". Ubiquitous access to shared data, content, services and resources will be provided.

The concept of a federation of networks, including external federation (overlay) and internal federation is illustrated in Figure 2.

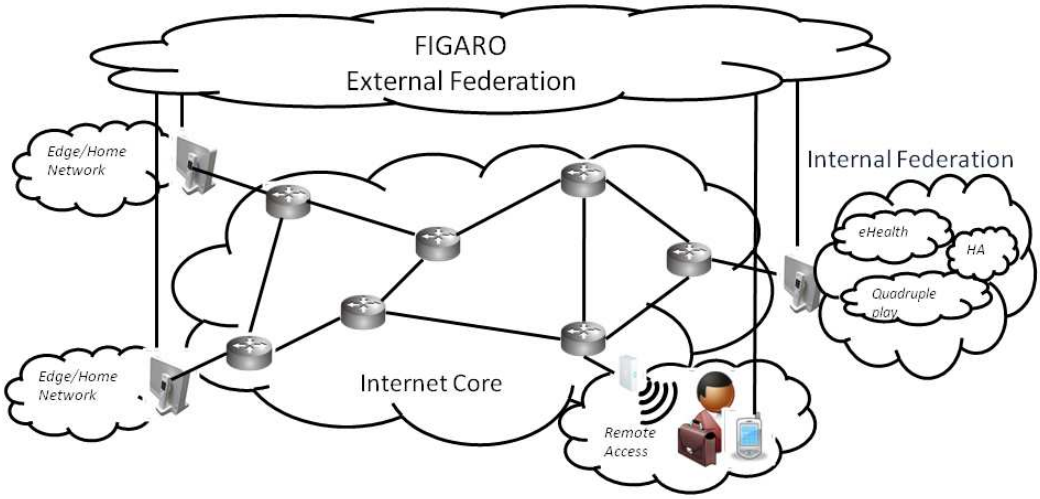

Figure 3: FIGARO overlay network.

In this thesis, the requirements of the federation concept are being analysed in Sections 4 and 5. Note than when we use the term 'federation' later in this thesis, we refer to the external federation.

3.2.2 Network monitoring framework

The FIGARO project introduces a network monitoring framework. Different monitoring techniques are used to provide a satisfying Quality of Experience for the home network users. The network and application performance is characterized at home and Internet-based uplink and downlink traffic is measured in order to optimize the link usage. In addition to the monitoring of wired networks, wireless protocols will be monitored to characterize properties such as bandwidth, delay and jitter. The Internet paths will be monitored as well by the gateway device. Passive monitoring is utilized for application traffic measurements. A packet sniffer will be included in the gateway design to make this possible.

This thesis does not cover details related to the network monitoring framework part.

However, the goal of this thesis is to present an experimental design that does not exclude the possibility of including such component in the system.

3.2.3 Federation-oriented Networking

Four different areas are approached in the network and federation domain. Those are the wireless optimization framework, the handover mechanisms, the neighborhood bandwidth sharing system and the adaptive streaming framework. The wireless optimization framework will use network access bundling techniques to optimize Internet access. This covers, for example, multiple network access technologies. The optimization will be performed by using content classification, unified interface representation and handover mechanisms.

The handover mechanisms make it possible to switch from one transmission technology to another in a smooth way. Media-independent handover protocol (IEEE 802.21) will be used in this context.

Multilink networking techniques will be used to enable neighborhood bandwidth sharing. Wireless channel bonding is performed and a single radio interface will be utilized to connect to multiple close-by wireless access points. The federation overlay is used in order to perform multilink networking with the gateways of the neighborhood.

An adaptive streaming framework will utilize SCTP protocol to provide concurrent and multi-path data transfers. By doing so, a certain quality of service can be guaranteed for each application flow. Possible, a SCTP proxy system will be implemented to enable NAT traversal.

This thesis does not study the usage of wireless optimization, handover mechanisms or multilink networking. Nor requirements analysis or design for an adaptive streaming framework is provided.

3.2.4 Content management

The FIGARO project is planning to design a residential gateway-based content management system. This system will enable distributed content backup, search and access. An unified access provides access to any type of content regardless of its physical location. Since content will largely be user-centric, social networking principles will be used in the context of federations. In addition, the publish-and-subscribe paradigm will be utilized in content sharing. The content management system will also be designed to support wireless ad-hoc sharing.

A caching system will be introduced. The system will use location-aware mechanisms, where content is stored on devices that are physically as close as possible to the most likely consumed content. Also, content replication will be optimized to provide better reliability and accessibility.

The back-up system is part of the content management module. The back-up system

uses peer-to-peer technology and can be based on several different models. First, the back-up can be federation-based. Second, cloud-services can be used. Third, local back-up mechanisms can be utilized.

In addition to the back-up system, gateway users are provided with a possibility for a remote home access. Home networks can be accessed from other gateways that enable the usage of home services outside the home network.

The content management system will be build upon a distributed file system. There is no need for a new file system, but existing file systems can be utilized. A virtual file system is build on top of the individual file systems residing on home network devices.

This thesis tries to address the challenges of distributed content management. An experimental design will be introduced in Chapter 5, which provides distributed file sharing.

3.2.5 Service management

The final system module of the FIGARO project is the service management system. New management and control modules will be provided that integrate services from other sectors into the residential gateway. These sectors cover home automation, energy management and e-health care. However, the design will not exclude other sectors. For example, in the case of energy management, the federation paradigm could be utilized by s smart grid system for energy negotiation and distribution in a neighborhood. The gateway architecture will make it possible to access these services anywhere from the Internet. The design aims to support protocols such as ZigBee and Continua v1. Existing control and management mechanisms are used, including TR-069 and UPnP.

3.3 Summary

In this chapter, the FIGARO project was introduced. The project will provide a gateway centric future Internet architecture. The gateway will be supporting various novel technologies and utilizes the federation approach in order to address content and user centricity, present in the nowadays Internet.

The ideas of the FIGARO project will be used as an input in the requirement analysis that is represented in the following chapter. The analysis is be limited to the concepts of internal and external federation and content and service management. Later on in this thesis we use these identified requirements to develop an experimental design for a gateway and verify the design with a prototype implementation.

4 Requirements Analysis

In this chapter, we adopt a use case based approach and perform an analysis for the functional requirements using the use cases introduced in Section 3.1. The use case analysis is followed by an analysis of the non-functional requirements.

4.1 Functional Requirements

In this section, we analyse requirements for a system described in Chapter 3. The analysis is performed by examining the use cases introduced in Section 3.1.

We limit the analysis based on the conceptual gateway-centric system architecture introduced in Section 3.2. The analysis is limited to the following subdomains:

(1) Internal and External Federation

(2) Content Management

(3) Service Management

The emphasis is on the requirements that are related to the internal and external federation concept. The content management intersects with the federation concept since content is shared and stored among federation members. Thus, content management is analysed. In the service management analysis, we study how to better support emerging services at the residential gateway. The network monitoring domain and the federation-oriented networking domain such as the wireless optimization frameworks and the handover mechanisms are not discussed in the analysis.

4.1.1 Use Case 1

In the first part of the use case 1, "managing content", a movie is transferred from a digital camera to the user's home storage system while the user is outside her home. In the second part, the movie is easily found in the common storage system and watched using TV. We omit the network monitoring part in this analysis since it is not part of our scope.

Before proceeding to the analysis, it is necessary to make some assumptions. Since the system is based on residential gateways, we assume that the user is visiting another gateway while being outside her home. Also, we assume that the gateway takes an active role in transferring the move to the common storage system. In other words, the gateway has some logic that communicates with the storage system and makes the transmission possible. We assume that the common storage system is not necessary a single system, but could be a network of devices located in the home network, or possible outside home. There must be a mechanism that transfers the

movie from the camera to the gateway. For example, Wi-Fi technology could be utilized. Finally, in the home network, we assume that there is a protocol that both the TV and the gateway use for communication, and that the gateway is used for managing content.

Based on use case 1 and the assumptions above, we derive the following functional requirements. First, in order to utilize other users' home gateways (foreign gateways) to transfer content to the storage system, a mechanism is required to provide visitors with an access to the gateway. We name this as *a visitor access*. In addition, it is reasonable that not anyone can utilize a user's home gateway, but that the visitor access is controlled through *an authentication mechanism*.

Since we assumed that gateway implements some logic that makes the content transmission possible, it must be possible to execute software on the gateway. There are two options related to this: either the logic is installed in the gateway beforehand, for example by the gateway manufacturer, or there is a mechanism that makes it possible for the gateway users to install applications to the gateway. In order to provide a flexible system that can better address the requirements of the future, we assume that the user has the possibility to install applications to the gateway. This leads to the requirement of a user-controlled *application execution environment*.

In order to transfer the video from the camera to the storage system, *a content management mechanism* is required. This system should make the transfer operation easy for the user. Since the video can be easily found in the home network without knowing the exact device where the video is located, the content management mechanism should provide a *unified view of the content* regardless of the location of the content.

Finally, there is a requirement for connectivity between the home gateway, the common storage system and to the TV. In other words, the residential gateway must be able to use the protocols that the storage system and the TV understand.

4.1.2 Use Cases 2 and 3

The use case 2, "visiting friends", and the use case 3, "sharing content", are very similar to each other. In the use case 2, the user visits a friend and shows the friend a video through the friend's residential gateway. The gateways close to the visiting user's home gateway are utilized in order to increase uplink bandwidth. In the use case 3, a personal TV program is prepared together with a group of friends and this video is made available through a social network system.

We assume that the user has shared some of her home network's content with a federation before visiting the other user. In order to provide an Internet community-based access to the home network content, there must be a mechanism to *set up and use community networks* between residential gateway devices. Thus, *a federation management mechanism* must be provided. This system allows users to add and remove users from the community and define which resources and what content is shared among the federation. Hence, community members are provided with an

access to the shared storage system.

The uplink bandwidth sharing among neighboring gateways requires that the neighboring gateways are able to share data locally using fast link layer mechanisms, such as, Wi-Fi. Then, the uplink bandwidth of several gateways can be utilized in order to deliver the content faster to the destination. Hence, the system must provide a mechanism for autonomous communication between gateway devices that belong to the same federation.

4.1.3 Use Case 4

In the use case 4, "remote home access", the user connects to another gateway over the Internet and manages a device located in the remote home network. The gateway must therefore support *remote access* that provides means for the user to control devices behind the gateway. In the use case, the controlled device is a heating system. This sets a condition that the gateway must be able to communicate with the heating system using appropriate medium and protocols. In addition, unauthorized access to the remote home network must be prevented and the communication channel should be encrypted, or at least the origin of the messages should be verified. Thus a *security mechanism* is also required.

4.1.4 Identified Requirements

To summarize the use case analysis, we identified the following requirements:

1. *Content and Resource Sharing.* A system that makes it possible to share home network content and resources using residential gateways is required. This functionality can be understood to be part of the content management and external federation functionality.

2. *Backup System.* Means to backup content is required. This can be understood to be part of the federation concept; the mechanisms of the content and resource sharing can be exploited. The backup system can be considered as part of the content management and external federation functionality.

3. *Unified Content Management.* A mechanism that provides a unified access to the home network content is required. The unified access is independent on the location of the content. Hence the files can be located on any device connected to the federation. Unified content management is part of the content management and external federation functionality.

4. *Execution Environment.* An execution environment is required in order to support arbitrary services at a gateway device. This requirement is part of the service management functionality.

5. *Visitor Access.* Visiting users must be able to utilize other gateways that are part of the same federation. This requirement is related to the external federation functionality.

6. *Security Infrastructure.* An entity that controls the security in the system, including authentication and confidentiality is required. This system can be understood as part of the federation management functionality.

7. *Federation Membership Management System.* A system that controls how users join and leave a federation is required. It is part of the federation management functionality.

4.2 Addressing Functional Requirements

In this section, we discuss how the functional requirements can be addressed. Our goal is to provide an overview of possible solutions. Later, we will use some of the solutions discussed in our experimental design.

4.2.1 Content and Resource Sharing and Backup

A content and resource sharing mechanism is required in order to share home network content and resources between other home networks. We assume that the same mechanism can be used for backing up content. As discussed in Chapter 3, we assume that content and resource sharing is done by using the federation concept. This means that gateway users can establish arbitrary content and resource sharing overlays between homes.

The number of established federations can potentially be large. Thus, the system must be designed so that it scales to support an arbitrary number of users and federations. Since our purpose is to provide a content and resource sharing system that uses residential gateways, we do not consider a centralized storage system. Our goal is to provide a distributed content sharing system.

The peer-to-peer technology (P2P) has been proved to scale well. P2P systems are used as an option to client-server systems when sharing resources and content. Unlike in the client-server approach, in a P2P system all nodes are equal by having the same functionality. In server-based systems, high availability is required from the server whereas in P2P, some of the peers may be unavailable. Availability problems are resolved by duplicating the resources to several peers. The system assumes that there are enough peers in the network so that some peers with the desired resources are online [15, 72, 68].

There are several different peer-to-peer architecture categories. In centralized P2P networks, the system is dependent on a centralized directory server. Peers inform the directory server about their current resources. To find a resource, a peer sends a resource query to the server and the server responds with the address of a peer

owning the desired resource. Napster, among others, implements this architecture [53, 48].

In decentralized and structured systems, a centralized directory server is not used. However the network topology of a peer-to-peer system is more or less controlled. Every resource that is added to the system has a location which is calculated using a hash algorithm. The structured peer-to-peer systems can be furthermore divided into loosely structured and tightly structured systems. In a loosely structured system, the placement is based on hints whereas in a tightly structured system (e.g. Pastry) the exact location is always determined using a distributed hash table [72, 53]

In addition to the architectures mentioned above, in decentralized and unstructured peer-to-peer systems neither a centralized server nor a controlled topology is used. Peers join the network using loose rules. Resource queries are flooded to the network to find the desired resources [53, 48]. Because of flooded queries, decentralized and unstructured peer-to-peer systems are suitable for networks where peers frequently enter and leave the system. The downside is the increased network traffic and bounded scalability [53].

The features of the residential gateway affect the selection of a P2P-based content and resource sharing protocol. The residential gateways are most of the time online. We can also assume that new residential gateways join a federation relatively rarely. In other words, the topology changes of the federation overlay are likely to be fairly static. In addition, we assume that the number of file directory queries is large and they are often performed. Hence, the peers should be able to query and locate the content fast. Based on these observations, the features of the centralized P2P systems and the decentralized and tightly structured P2P systems appear suitable for federation based content and resource sharing in the gateway context. These type of P2P protocols provide fast mechanisms to query and locate content.

4.2.2 Unified Content Management

Several technologies can be utilized in order to provide a system that addresses the unified content management. These include, among others, UPnP/DLNA and Bonjour. These technologies do not automatically provide the user with a unified content management system, however. For example, in the case of UPnP, all devices must support the UPnP protocol in order to discover and manage content in a uniform manner inside home. In addition, the multicast-based protocols such as UPnP and Bonjour work currently only in a single LAN domain. Hence, these technologies cannot be utilized as such if we want to make the unified content management system that covers several home networks.

Since we want to provide similar type of access to the content irrespective of whether it is in the home network or in the federation, we need to design a new content management system. The unified content management system should be able to communicate with various different content management protocols that are used in the home network. For example, UPnP and Bonjour can potentially be utilized as a

part of this content management system. In addition, the system should communicate with the content and resource sharing system discussed in the previous section, in order to provide a unified content management view over the home network and the federations. The unified content management system interconnects the different content management technologies within the home network with P2P-based federation overlays.

4.2.3 Execution Environment

We identified the requirement for a software execution environment in the residential gateway. This requirement has already been addressed by the OSGi framework [86]. OSGi is a Java sandboxing-based technology which separates different application processes and provides application life cycle management. Application can be, among others, installed, started and uninstalled using the framework. Being a well-defined framework, OSGi provides a mature solution for an execution environment inside the residential gateway. However, the OSGi framework is limited to Java sandboxing. Some existing system that have been implemented using other means than Java might still benefit from an execution environment inside a residential gateway. This might be the case, for example, with the services provided by the emerging sectors. Thus, in this thesis we have decided to study the usage of hardware virtualization in a residential gateway. Hardware virtualization provides user applications with a complete operating system platform.

In hardware virtualization, a software layer behaves as a virtualized hardware system. This layer is called the hypervisor and it makes it possible to run simultaneously multiple operating systems on the same hardware. The operating system running directly on the hardware is called host. The virtual instances that run on top of the operating system and interact indirectly with hardware are called guests [77].

The virtualization technologies can be divided into three categories: full virtualization, paravirtualization and software virtualization. As stated earlier, we omit the discussion about the software virtualization. In full virtualization, the underlying hardware is completely simulated. The benefit is that any operating system can be run on top of this system. Respectively in paravirtualization, the underlying hardware is not completely virtualized, but an API is provided for the guest operating systems to benefit from the virtualization. The drawback is that the guest operating systems have to be modified. However, paravirtualization provides better performance compared with full virtualization. Two popular virtual systems providing full virtualization are VMWare and VirtualBox. The most used paravirtualization systems include Xen, KVM and OpenVZ [77, 11, 45, 64].

In the context of the residential gateway, the selection of the virtualization system is affected by the cost and performance constraints and the flexibility requirements. If we want to support arbitrary services, full virtualization is required. For example, some specific building automation system might require the presence of a particular operating system that cannot be modified so that it would function on a paravir-

tualzed system. However, full virtualization requires more performance, which in turn, increases the cost. In the context of the residential gateway, paravirtualization might be a suitable compromise between the performance and the flexibility. A large number of operating systems have been modified to work on paravirtualized systems, such as Xen.

4.2.4 Visitor Access

In order to provide visiting users with access to their federations and home gateways, we identify two requirements that have to be addressed. First, there must be a mechanism that prevents visiting users from utilized the foreign gateway before they have been authenticated. Second, the visiting users must have access to the gateway though a technology such as Wi-Fi or Ethernet.

In many existing gateways, the first requirement is being addressed by using a mechanism called captive portal [99]. A captive portal works as follows. When a new user connects to the access point, the network traffic originating from the user is blocked by using a MAC address based filter. When the user opens a web browser, the DNS queries are resolved normally. However, when the user's browser sends an HTTP query to the resolved address, firewall will redirect the query to a redirect server which answers the client with a specific HTTP status code. This answer message will command the web browser to redirect the query to the authentication page. On the authentication page, user credentials are provided over a SSL connection.

The problem with captive portals is that after the authentication, users are identified using their MAC address. A MAC address is easy to spoof, which makes it possible to get an unauthorized access to the gateway. In order to better support security, the security mechanisms provided with the Wi-Fi standards, such as 802.11i, could be used for authenticating the users at the gateway. However, these mechanisms are often considered as cumbersome, especially when providing users with a temporary Internet access. An evidence of this is the popularity of the captive portal-based authentication systems in public Wi-Fi access points.

In addition to the mechanism that prevents visiting users from utilized the foreign gateway before authenticated, there must be a mechanism to connect to foreign gateways. To address this, the residential gateway can support multiple Wi-Fi access points. Some of the access points are used by the home network users and others are used by the visiting users. The previously discussed captive portal can be used on one of these visiting access points. There are two ways to provide multiple Wi-Fi access points on a residential gateway device. The first way is to add multiple Wi-Fi cards to the gateway device. The second way is to use a single Wi-Fi card for providing multiple access points. In this case, the so called virtual access point (VAP) method is used. For example, a driver called MadWiFi [85] supports up to 4 virtual access point on a single Atheros-based Wi-Fi card. In addition, the Click Modular Router Project [90] provides means to configure Wi-Fi drivers so that VAP can be supported.

The virtual access point technology can be seen as a potential solution to provide multiple Wi-Fi access points on a single residential gateway device. The main benefit is the cost savings compared with multiple Wi-Fi cards. However, the performance of a single Wi-Fi card is shared among several virtual Wi-Fi networks, which can harm the home users' quality of experience.

4.2.5 Security Infrastructure

While the previously discussed captive portal is used to prevent visiting users from accessing the Internet and services before authentication, a security infrastructure is required in order to provide the actual system-wide authentication. The same security infrastructure is required for providing data encryption between the gateways. Public Key Infrastructure (PKI) is a standard way to address security in the computer networks. In PKI, asymmetric encryption is used in order to authenticate users and share secret keys that are then used in symmetric encryption to secure the communication channel. In PKI, certificates are used in order to authenticate users securely. An entity called Trusted Third Party (TTP) assigns certificates to the users of the security system. Several systems.

Several PKI software exist. For example, companies such as RSA Security and VeriSign provide PKI solutions. In the federation context, each gateway and possibly each user must be authenticated. Thus, if a PKI system is used, an entity that assigns certificates to the gateway users is required. In addition, there must be a secure way to verify the different federation memberships of the users.

4.2.6 Federation Membership Management System

Whereas the security infrastructure, among other functions, is used for authenticating federation members, the federation membership management system is used for joining and leaving a federation. This entity is used to maintain the access control list of a federation. The system could also be used for inviting users to a federation. Similar principles present in many of the popular social network sites, such as Facebook, could be utilized when designing a federation membership management system. Due to the scope of this work, we decided to omit the federation membership management system from our design.

4.3 Non-functional Requirements

In this section, we analyse non-functional requirements. Instead of analysing how the system should function, we analyse features that are required in order to support these functions. Thus, the non-functional requirement analysis has to take into account the required functions discussed earlier in this chapter. In addition, we provide an overview of how non-functional requirements could be addressed. This is performed by analysing the current and future features in residential gateways. We

classify the non-functional requirements into four areas; the physical requirements, performance requirements, the I/O requirements and the cost requirements. Peripherals, such as hard drives, are not discussed but the focus is on the core features of a residential gateway device.

4.3.1 Physical Requirements

We identify three main physical requirements for a residential gateway. First, the size of the device is limited. Because residential gateways can be introduced in various type of home environments the device should typically be smaller than a laptop and it should fit into a bookshelf, TV furniture or on a worktable. Second, the device should use as less energy as possible since it is kept always on. Third, the residential gateway should be as quiet as possible. For example, fans are not typically included in a gateway due to the noise. The physical requirements have an impact on the performance. For example, the CPU cannot be cooled using fans. In addition, the device size limits the number of interfaces that a gateway device can be equipped with.

4.3.2 Performance Requirements

We limit the performance requirement analysis to two principal components; the CPU and memory. Thus, the requirements of other components such as computer buses and I/O device performance are not covered in this section.

Currently, the performance provided by typical residential gateway devices is fairly modest. For example, a popular residential gateway in France, SFR NeufBox 4 (NB5), contains a 300 MHz MIPS processor with 32 bit based instruction set. In addition, the device has 32 MB of RAM memory [62]. Typical applications running inside a residential gateway, such as a DHCP server, do not require much from the hardware. However, the landscape changes when the performance features of the set-top box devices are being evaluated. The set-top box is a device that converts the television signal to a proper format for the television. The physical requirements of a set-top box are similar to the residential gateway (the size, energy and noise constraints). Thus, the performance requirements can be analysed through the characteristics of the current set-top boxes.

When an increasing number of demanding applications are introduced on residential gateways and set-top boxes, higher performance is required from the hardware. At the moment, the boom of the 3D technology is one of the driving factors for the performance improvement. For example, Orange is planning to include an Intel Atom CE4100 processor in their forthcoming 3D-enabled set-top box. According to Intel, CE4100 will speed up to 1.2 GHz clock rate [40, 39]. Another French operator, Free, is expected to introduce a set-top box that will have a processor equivalent to the Intel's CE4100 [92].

In the functional requirements, we stated the need for hardware virtualization in the

gateway device. When several virtual machines are run in the same device, more CPU power and especially RAM is required from the hardware compared with the non-virtualized systems. One possible way to better support virtualization is to support it on the hardware level. Several modern CPUs provide instruction extensions that support virtualization and many of the existing virtualization systems utilize these instructions if they are available.

We assume that, due to the Moore's law, the hardware manufactures keep introducing more and more powerful and energy efficient chips for the residential gateway type of household devices. Thus, it is probable that in the near future, a residential gateway device can be composed of hardware equivalent to a setup found in today's low-price laptops. This could mean, for example, a device with 1 to 2 GHz processor having 1 to 2 GB of RAM memory. This setup is capable of providing hardware level virtualization.

4.3.3 I/O Requirements

The design decisions are affected by the number and type of available interfaces on a residential gateway device. Currently, a typical residential gateway contains a large variety of different types of interfaces. For example, the previously discussed NeufBox has the following interfaces [62]:

- 2 USB ports
- 4 Ethernet ports
- 1 SFP optical port
- 1 PCMCIA port
- 2 RJ11 Phone port
- Wi-Fi 802.11g

Moreover, another operator, Free, extends the interface list with the Wi-Fi MIMO and HomePlug AV interface in their FreeBox device [83]. The number of different interfaces is expected to increase along with the new link layer technologies at home. As stated in the background section, Home Gateway Initiative's Phase 3 and 3+ residential gateway descriptions extend the list of current interfaces.

The interfaces that are currently provided by the residential gateways are sufficient to fulfill the functional requirements introduced earlier. The communication with the home network can be performed using Ethernet and Wi-Fi interfaces and the communication towards the Internet is done through a wide area network interface (WAN). However, as stated in the chapter 2, various other network technologies are emerging at homes. Thus, one non-functional requirement for the design is to introduce an architecture that can be easily modified to support future technologies in the gateway device.

4.3.4 Cost Requirements

The design decisions are also dependent on the cost factor. The cost impacts directly to the performance and I/O requirements of a device. The best solution from a technical point of view is rarely selected because of the high price. It is likely that residential gateway manufacturers tend to select hardware configurations that are compromises between the performance, the supported I/O interfaces and the price. Because our functional requirements affect the performance requirements, the cost is affected as well. However, since our target is next generation gateways, and increased hardware cost is exptected and accepted. Eventually, the cost will decrease and hardware components become mainstream.

4.4 Summary

This chapter provided a requirement analysis for a next generation gateway. Since the concept of the next generation gateway is broad, it is important to clearly identify the context and scope of such an analysis. Our point of reference is the FIGARO project and we performed the requirement analysis based on its vision and use cases. As a result of the analysis, seven main functional requirement were identified: (i) the content and resource sharing, (ii) the backup system, (iii) the unified content management, (iv) the execution environment, (v) the visitor access, (vi) the security infrastructure and (vii) the federation membership management system. Each of these requirements were discussed and some alternatives to address these requirements were introduced. Based on the functional requirements, non-functional requirements were analysed.

We believe that if we manage to properly address these identified requirements, we are able to improve the home network and content management, and better support the integration of emerging services at homes. In the following chapter, we introduce an experimental design for a next generation gateway that aims to address the requirements identified in this chapter.

5 Experimental Design

In this section, we will introduce an experimental design for a next generation residential gateway. The design is based on the requirements identified in the previous chapter (Section 4.1.4). Our goal is to address these requirements in a concrete manner. Thus, we aim to provide a design that can be used for implementing a residential gateway that addresses the problem statement discussed in the section 1.1. The following requirements will be addressed:

1. *Content and Resource Sharing.* We will design a distributed file system which can be used for sharing content among a federation
2. *Backup System.* The distributed file system can be used for backing up content.
3. *Unified Content Management.* The distributed file system will provide a unified view of the content irrespective of its physical location.
4. *Execution Environment.* We will design a system where all applications are run inside a virtual machine. In addition, we will introduce a system for installing application into these virtual machines.
5. *Visitor Access.* Our design will support visiting users by providing connectivity to foreign gateways.

In order to limit the scope, we decided not to design a security infrastructure or a federation membership management system. However, our design will take into account the presence of these two requirements. We will also discuss both of them briefly by pointing out ideas that are worth considering when addressing these requirements. More emphasis is put on the above-listed requirements 1 and 3 which are addressed through a distributed file system. In addition, the requirement 5 is concentrated more closely by designing a system based on the hardware virtualization.

We organize this chapter as follows. The distributed file system, being a large and essential component in our design, is introduced first. This is followed by the sections about the application management, security infrastructure and membership management. These sections discuss the design decisions of each sub-system. Once these topics are discussed separately, we give an overview of the overall system design. Each system component is then discussed separately. The chapter concludes with a summary.

The experimental design will be used as a reference design for the prototype implementation that we introduce in the following chapter. As said, the experimental design does not try to provide a complete system that addresses all the requirements of a next generation gateway. Rather, our goal is to identify and design the essential system modules that can be used in order to address some of the requirements identified in the chapter 4.

5.1 Distributed File System

To address the three identified requirements; content and resource sharing, backup system and unified content management, we introduce a distributed file system. For each federation, a separate distributed file system is provided. In other words, each federation has its own overlay network.

5.1.1 Partially Centralized Peer to Peer System

Different P2P mechanisms were introduced in Section 4.2.1. We decided to design a distributed file system that is based on a partially centralized P2P system. Thus, we chose a BitTorrent type of protocol. The advantage of a partially centralized P2P system is that the content can be searched and accessed fast. However, the weakness of such a system is the centralized components which can be a potential single point of failure. Other P2P techniques could have also been used, for example, decentralized and tightly structures P2P systems. While this work considers a single type of P2P technique, in the future work, several different P2P protocols could be utilized and the selection of the protocol could be based on the content type of the file being transferred. For example, chunks of a video file should be downloaded in a chronological order, while some other file types, such as application binary files, could be downloaded in an arbitrary order.

Since our design is based on a partially centralized P2P system, the file directory of the federation is kept in a centralized place. The file directory describes which files are shared in the P2P system, in this case, among a federation. Details related to the files are kept in this directory, including a file name and a size. In our design, the centralized place where the directory is kept is a server instance that is hosted in a single or several residential gateways. When a peer wants know which files are shared among the federation, it queries the file directory server.

In addition to the file directory, one or several tracker instances are required in the P2P system. The tracker keeps track of each file in the file system. When a new file is added to the system, a peer informs the tracker. Respectively, when a peer contacts the tracker and queries a particular file, the tracker provides the peer with the addresses of all other peers in the P2P network who have chunks of the queried file. Then the peer who made the query can contact to these peers individually to download pieces of the file [12].

BitTorrent uses hash entries called torrents to identify files in the P2P network. For each content file a separate torrent file is created. In our design, we use the directory server as a place to store a torrent file for each file shared among the federation. Thus, when a peer wants to download a file, it first downloads the corresponding torrent file from the directory server and then send the hash value inside the torrent file to the tracker in order to get the addresses of the peers that have chunks of the file.

When a peer wants to add a new file to the distributed file system, it first creates

a torrent file for the content file. After that the peer notifies the directory server about the new file which causes the directory server to update the federations file directory with a new entry. Then the peer uploads the torrent file to the directory server. Finally, the peer notifies the tracker about the new file. This is done by sending the hash value inside the corresponding torrent file to the tracker.

In our design, the file directory server and the tracker are placed in the gateway that created the federation. We name the founder of the federation as a *founder peer*. In order to provide durability and redundancy, the file directory server and tracker are replicated to several gateways in addition to the founder peer. Thus, a synchronizing mechanism is required in order to keep the index servers and trackers consistent. Figure 4 illustrates this mechanism. Our experimental design has a single file directory and tracker per federation.

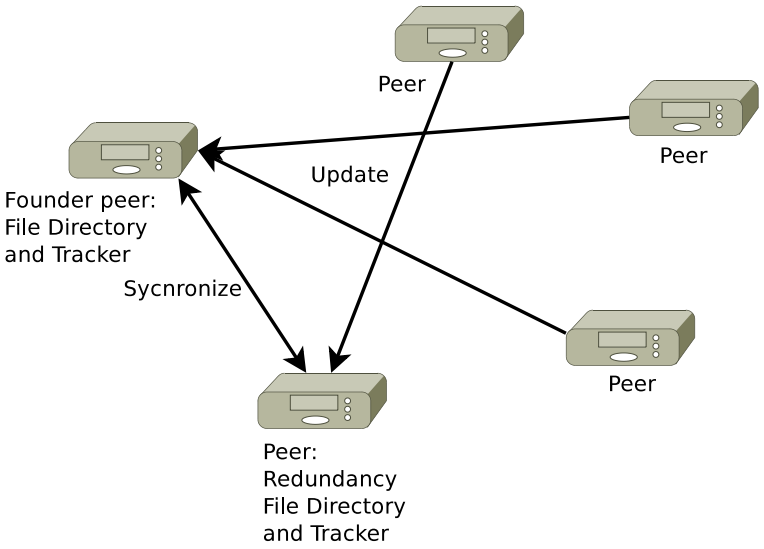

Figure 4: Index updating in the distributed file system

Each time a user searches files, a query is sent to the directory server. Thus network traffic is generated every time the file system is used. Also, the directory servers have to serve a potentially large number of queries. In such a case, the founder peer can potentially become overloaded with file directory queries. In addition to the directory server, the same issue is faced with the tracker. Different mechanisms could be utilized to ease this problem. For example, each peer could have a local copy of the file directory. These peers could update their local copy once in a while by connecting to centralized directory and tracker servers. Updates related to new files would be performed in the same manner. Relatively long intervals between updates could be utilized. This would naturally prevent the users from having immediate access to the latest content once its been added to the federation. Futher exploration of the most suitable file directory distribution system for the federation content sharing purposes is outside the scope of this thesis.

5.1.2 Writing Policies

One major problem with the distributed file system is the simultaneous writing. When several users try to modify a file at the same time, the distributed file system must make sure that only one of the users modifies the file. The modifications made by the other users should be prevented. To coordinate this, several messages needs to be exchanged between the peers. When the federation becomes large, this incurs significant overhead and complexity.

A major part of the content inside home networks is static. For example, videos, music and photos are rarely modified after they have been created. Thus, a file systems based on Write Once Read Many principle (WORM) or Multiple Read, One Writer principle (MROW) can be sufficient for our system. In WORM, a file can be written only once to the file system. In MROW, only the creator or owner of the file is allowed to modify it. The advantage of these two approaches is the simplicity of the design in proportion to a system where everyone is allowed to modify every file.

Our BitTorrent based distributed file system uses the WORM principle. However, when considering the flexibility and complexity, the MROW principle appears to be a suitable compromise for the federation content sharing purpose. Hence in the future versions, MROW model could be used.

5.1.3 Sharing vs. Isolation

In our design, each file which is added to a distributed file system is visible and accessible to all members of the federation. This is due to a simplicity of the design. However, in order to provide more flexible usage of federations, the possibility for private directories should be supported. By doing this, users could, for example, make remote back-up copies of their private content. The content placement system (see 5.1.8) could be utilized to make remote copies of the files to other user's gateways. In this case the access to these files would be restricted only to the owner of the files. Each user could then allocate some amount of storage capacity that would be used to store back-up copies of other user's content. The users would then get the same amount of remote back-up capacity from the federation. A design for such system is not considered in this thesis.

5.1.4 File System Abstraction

An overview of the distributed file system architecture is provided in Figure 5. Because the distributed file system can be used by arbitrary client applications, the interface to the distributed file system should be abstracted so that these client applications are able to use the file system in an easy manner. In our experimental design, the distributed file system is abstracted to the level of a normal operating system file system. In other words, the P2P overlay can be used through standard file system commands provided by the operating system. For example, if Linux

operating system is used, the files in a distributed file system directory can be listed using the "ls" command.

The file system abstraction provides users with a unified file system view where the physical location of the content is hidden. In other words, the content that is accessed through the file system abstraction layer can be located in the local home network or in some of the other home networks that are members of the same federation.

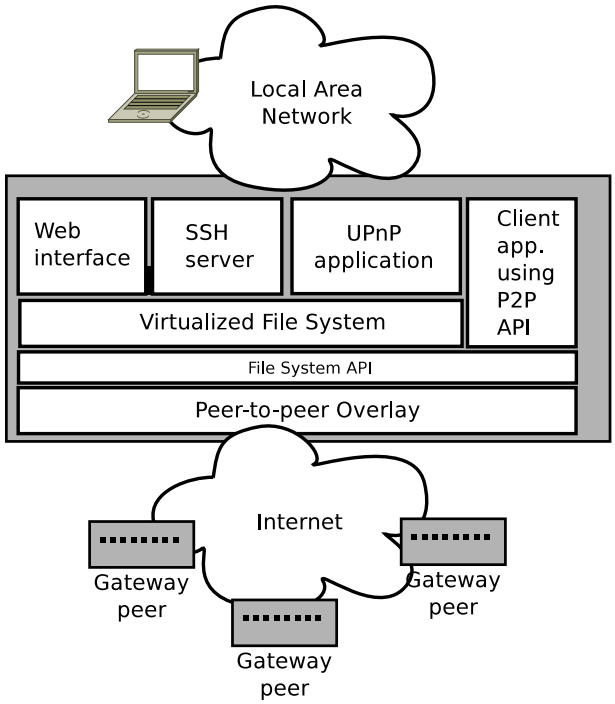

Figure 5: Overview of the distributed file system architecture

Applications of various kinds can be imagined on top of the abstracted distributed file system. Some possible applications as listed in Figure 5. For example, a web interface can be used for browsing the files of a file system directory, which in this case are shared among a federation. This provides users with a convenient way to browse the content of a federation by using a web browser. Also, a SSH server can be installed in the gateway as well in order to access the distributed file system.

To share content among UPnP-enabled devices, a UPnP client application can be installed at the gateway. The UPnP client sees all UPnP content available in the LAN to which the gateway is connected. Once a UPnP client is run inside the gateway, it is possible to transfer the UPnP content of the home network to the distributed file system. In this case, the UPnP client application first copies a file from an UPnP device to the gateway and then this file is moved to the distributed file system by using an additional mechanism, such as a web interface. An example of this kind of setup is given in Section 6.4.2.

Instead of immediately copying the file in the UPnP network device to the gateway,

it is also possible to store only an entry of the file to the distributed file system. This entry describes file details such as the file name, date and size but does not contain the actual file data. At the first time when a federation member outside the home network wants to read the file, it is copied from the UPnP network device to the gateway. After that, the file is transferred over the Internet to the home network of the federation member who wanted to read the file. This is done by using the P2P overlay described in Section 5.1.1. Content management and caching mechanisms are discussed in detail in Section 5.1.8.

In addition to sharing UPnP files located to the home network, the files located to the distributed file system can be accessed using UPnP. This can be done by using a UPnP Media Server at the gateway. UPnP Media Server is used for sharing files inside a LAN. Thus, in order to share the files of the distributed file system, some directories of the distributed file system or even the whole file system can be shared using a UPnP Media Server. In this setup, the content of a federation appears as a UPnP Media Server present in the LAN.

5.1.5 File System API

Even though the file system abstraction described in Section 5.1.4 provides a simple and clear way to utilize the underlying P2P overlay, it has some drawbacks. The file system does not provide any feedback regarding the data transmission process. In case of a large file, the application using the file system must wait until the content is downloaded to the gateway. It is of course possible to read and write file chunks periodically. For example, a video decoder can read a part of the file to its internal buffer while the P2P overlay keeps downloading and writing the rest of the file. However, in some cases, client applications would benefit from a system that provides more feedback and control. For example, sometimes knowing the download rate of the P2P system can be useful. Occasionally, it can also be beneficial to be able to define the order in which the file chunks are being downloaded.

To provide more fine-grained control over the P2P overlay, an API for the P2P system is exposed. In our experimental design, the API provides the following functions: *start new file system, make directory, list files, download file, get download percentages* and *upload file*. Each of these methods communicates directly with the P2P overlay. The file system abstraction uses these methods to utilize the P2P overlay. In future work, more methods could be provided in order to give applications more control.

5.1.6 Subscription Mechanism

In the case of media content, download times can be long. To reduce the duration when accessing the content first time, a part of the file or even the whole file can be downloaded to the home network before the user accesses it. In our design, we introduced a simple *subscription mechanism* in order to control the content

caching. In our subscription system, users select directories whose content they want to be automatically downloaded to their home network. When a user adds content to the subscribed directory, the files are automatically downloaded to the home networks of the users who have subscribed to that directory. In our design, the files are completely downloaded. In future versions, more advanced systems could be designed that provide users with the possibility to define which parts of the file are initially being downloaded.

5.1.7 Content Placement in the Home Network

The content can be placed on various physical devices in the home network. For example, files can be stored directly into a hard drive attached to the gateway or into an external NAS device.

Our design addresses the content placement at homes as follows. When a user stores a file to the distributed file system, the physical location of the file is first recorded to an entity called *link list*. This is illustrated in Figure 6. The link list is a list of *location entries*. These location entries have two data structures, the *file description structure* and the *location structure*. The file description structure describes the details of the file such as the name, the logical directory and the size. In addition to these, the file description structure includes a hash value of the whole file that is used by the BitTorrent protocol. Respectively, the location structure describes the physical location of the file inside the home network. A location can be for example a directory in a NAS device or in a UPnP Media Server.

Once the location of the file in the home network is recorded in the link list, the P2P operations are performed: the file is registered to the file directory server and to the BitTorrent tracker. These operations make it possible for other federation members to find and download the file.

Users can create directories into the directory server and storge files into these logical directories. Before a user can create a directory or add a file to the directory server, the system checks whether an entry with the same name exists in the system. In case so, the operation fails and the user must define another name for the new file or directory. Thus, each file and directory must have a unique name in a directory. The name and the logical directory of a file, which are stored to the file description structure, are kept consistent with the directory server. Files and directories cannot be moved once they are added to the directory server.

When a user wants to read a file that is located in the distributed file system, the system first checks whether the file exists locally in the home network. In order to do so, the link list is used. If the file is not found in the home network (the location entry is not on the link list), the file is downloaded from other locations over the Internet by using the P2P protocol.

Respectively, the following operations are performed when a federation member wants to download a file that is located in other federation member's home network. We name these two type of users as *distant member* and *home member*, respectively.

First, the file system of the distant member contacts to the file system of the home member to ask the file. The file system of the home member uses the link list in order to find the location of the file in the home network. Then, the file system copies the file from the home network device to the gateway and sends it to the file system of the distant member.

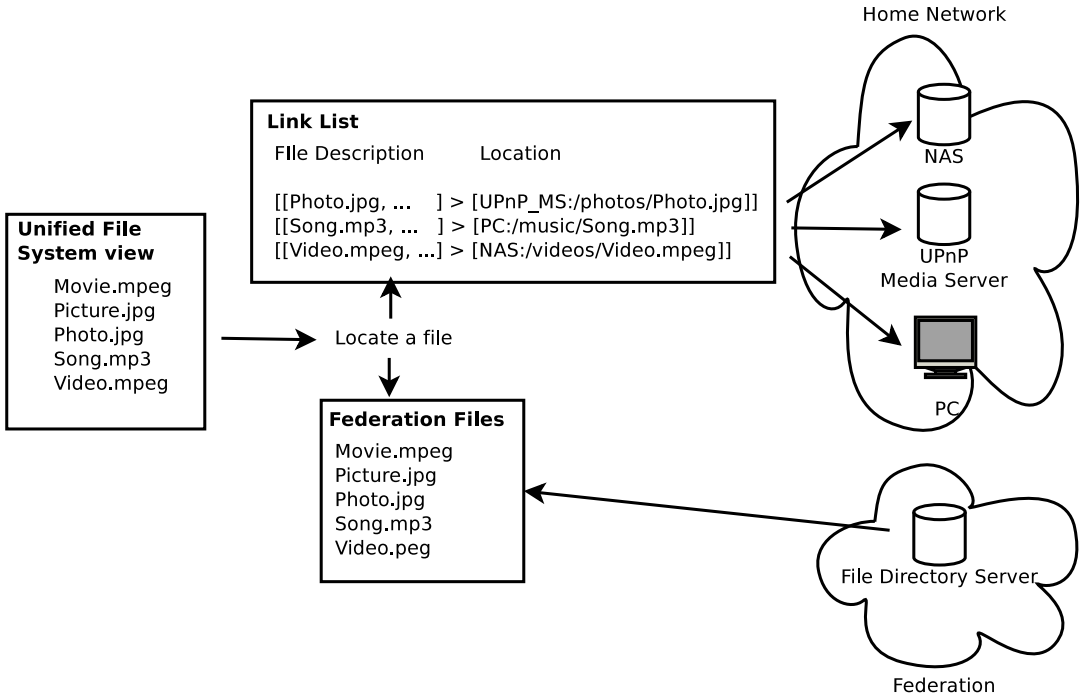

Figure 6: Managing physical locations of the content

In case a federation member wants to delete a local file, the following phases are performed. First, the peer contacts to the BitTorrent tracker. The BitTorrent tracker modifies the entry that has the same hash value than the file being deleted: the IP address of the federation member who wants to delete the file is removed from the entry. As a result of the deletion operation, the BitTorrent tracker returns an updated list of the IP addresses having the file. In case the returned list is empty, in other words no one has the file or pieces of the file, the peer contacts to the file directory server and asks the directory server to remove the entry of the deleted file. After the removal of the address from the tracker and the potential removal of the file directory server entry, the file system deletes the file from the storage device. Finally, the link list entry is removed.

The deletion operation described above has a weakness. In case the peer, which deletes a file, has the only complete copy of the file, an entry of the file is preserved in the distributed file system although the file is not complete. Since some peers may be offline during the deletion operation, it is difficult to verify does the whole file exist in the distributed file system after the deletion operation is performed at a single peer. To address this, for each file the peer can periodically contact the other peers and ask which parts of the file they have. In case the peer finds out, after

performing a certain number of queries, that the complete file is missing in the file system, it can decide to remove the file by following the same procedure that was described above.

5.1.8 Content Caching in the Home Network

The approaches described above introduce network traffic to the local network each time a remote federation user accesses files present at the home network. The traffic can have an effect on the quality of experience of the home user, especially if a Wi-Fi link is used for data transmissions.

In order to address this, files that are often accessed should be located as close to the gateway as possible. A dedicated storage device can be attached to the gateway to store temporary copies of popular content. Which content is stored and deleted from the gateway is based on the caching algorithm being used.

When content is downloaded to the residential gateway, it must be physically placed somewhere in the home network. To make the downloading process straightforward, users have to specify the location of the storage device before using the federation services. A caching system can also be introduced here. The recently downloaded content can be first stored into the cache device and then at a proper moment, when the network traffic level is low, the content can be copied to the targeted storage device and finally deleted from the cache. Several storage devices located in the home network can be allocated for the caching purposes. The detailed design of the caching system is outside of the scope of this thesis.

5.2 Managing Applications

As stated in Chapter 4, we need to provide the home users and the visiting users with a mechanism that makes the application management possible in the gateway. Our experimental design includes an application manager that has the following functions:

- Copy the installation file to the gateway
- Install application
- Start application
- Stop application
- Provide access point address, e.g. TCP/UDP port, to the application
- Remove application

In our design, single compressed installation files are used. This is due to simplicity of the design, usability and mobility of the applications inside different networks.

An installation file follows a predefined format by containing the source or binary codes of the application, installation instructions, starting and stopping instructions and the access point address. The application manager uses these details in order to perform the previously mentioned functions. The access point address is the address to which the user should connect in order to use the application. This can be for example a TCP/UDP port address in the gateway.

The installation files can be located in a distributed file system, in home network devices or on the Internet. In the future, these applications could also be downloaded from dedicated stores. One potential idea is the *Gateway Application Store* that all gateway users could utilize in order to buy and download new software to their gateways.

5.2.1 Application Management Issues

There are some requirements that the current version of our application manager does not address. Several matters needs to be taken into account when designing such a system for commercial use. First, the installation process should be done with a single click. Second, the installed application should work in "out of the box" manner. Third, the installed application should not break or influence the behavior of existing software. Fourth, the installed application should not behave maliciously in the local network. Fifth, users should be able to revert to the previous state of the system that existed before the application was installed.

The first and the second matter requires that the application binaries are included in the installation file. This can introduce overhead when each installation file contains all required libraries, even when the library would already exist on the gateway. To avoid this, a more sophisticated mechanism must be designed that takes into account the dependency requirements.

The third matter requires that there is a verification process for new applications before they are being deployed. This verification process must guarantee that the installed application work well with the existing setup.

The fourth matter is trickier. It is possible to do a network and application level monitoring on the physical gateway domain to detect anomalies. Policy-based rules could be applied in order to block applications that behave suspiciously.

Several options exist for the fifth matter. A virtualization approach bounds the possible damage to the domain of a single virtual gateway. The virtualization inside a gateway device will be introduced in Section 5.5. Users can always delete a broken virtual machine and start a fresh one. Existing applications can be reinstalled in the system. Another option is to back up the system images of the virtual machines periodically to off-site locations. The distributed file system could be used to implement the off-site copying. In a case of a failure, the previously stored image could be downloaded over the Internet and launched on a new virtual machine.

A robust and easy-to-use application management system is a large research topic.

This thesis does not try to evaluate which of these solutions is the most suitable for the next generation gateway architecture.

5.3 Security Infrastructure

As stated in earlier, a design for the security infrastructure is outside the scope of this work. However, in this section we briefly discuss the basic mechanisms that a security infrastructure for federation services requires.

In order to enable federation services, an authentication infrastructure needs to be built. The current design requires that all members of a federation are able to authenticate each other mutually. In addition, data encryption is required when transmitting packets over the Internet.

A certificate based public key infrastructure (PKI) must be implemented. This requires that there is a trusted third party (TTP) that assigns certificates for the gateway users. In the gateway based infrastructure, an ISP would be a logical instance for providing certificates for its clients.

In addition to gateway user authentication, there must be a way to securely map users to federations. In other words, there must exist a mechanism that allows each federation member to verify the membership of any fellow federation member. Clearly, there is a need for a Certificate Authority (CA) that assigns federation membership certificates to the users.

Since there can be an arbitrary number of federations there is a need for an arbitrary number of federation membership certificates. In order to address the scalability issues, these certificates could be assigned locally, on a federation basis. One possibility is to place a CA on one or several gateways that belong to the federation. The possible CA concept is illustrated in the Figure 7. Redundancy challenges of the similar kind as in the case of the file directory servers are also faced when it comes to distributed CAs.

Session key management becomes problematic in a file system where several hundred gateways can participate in the peer-to-peer file sharing. If data has to be transmitted over the Internet in an encrypted format, symmetrical encryption is required on these transmission links. Thus, a separate session key is required for each connection. In a traditional setup, each peer pair would require its own session key and PKI handshake. In the case of a relatively small federation, this can probably be tolerated.

One way to circumvent the problem of large amount of session keys is to encrypt sessions only between peers and index servers when querying federation content. In this case, the media content is sent over the Internet unencrypted and only the data origin is verified by signing the file chunks before sending them. This removes the handshake process, but endangers the confidentiality. In some cases, such as in the e-health systems, confidentiality is essential. Then, files can be encrypted locally before storing them to the file system. This, however, makes the content viewing

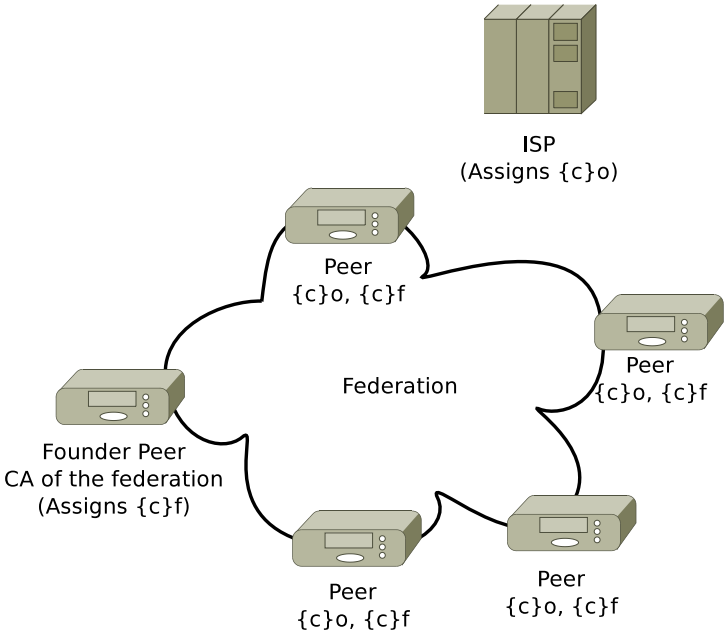

Figure 7: ISP and federation certificates

troublesome, when encryption keys need to be delivered between peers.

5.4 Federation Membership Management System

A federation membership management system is required in order to implement the federation concept. The system keeps record of the members in the federation. Thus, for each federation an access control list is created and maintained. The membership management is dependent on the membership policy of the federation, which respectively is dependent on the federation characteristics, such as the size and purpose. There are several possible federation joining policies. The first option is to let everyone join the federation. The second option is that a new member can join the federation after she gets an invitation from any of the existing members. The third option is that each new member must be accepted by an administrator before they are allowed to join the federation. Other policies may exist too.

In addition, there is a similar problem of leaving a federation. This becomes a policy question in the case of misbehaving members. One policy could be for example that an administrator can exclude a member after receiving a report from another member. Another kind of policy could be that a member is excluded after a certain number of different members have reported about misbehavior.

The Federation Membership Management System needs to be introduced in the context of each federation. In practice, it is a system that communicates with the Certificate Authority of the federation and makes it possible for members to commit federation membership actions. These include sending federation membership invitations and joining and leaving federations. This thesis does not present a design

for a membership management system. However, the design must take into account the existence of such a system.

5.5 Conceptual Overview

In this section we introduce the system design. The conceptual overview of our experimental design is illustrated in Figure 8. One physical residential gateway device implements several logical gateways. These logical gateways are implemented by using hardware virtualization on the gateway device. Hardware virtualization was discussed in Section 4.2.3. Hardware virtualization is selected in order to better isolate different processes from each other and to provide a flexible execution environment for the applications. We also wanted to address the needs of the emerging services. We believe that hardware virtualization provides better means to integrate emerging services within the residential gateway.

In Figure 8, three different type of *gateway profiles* are provided: the *Home Gateway Profile*, the *Visitor Gateway Profile* and the *Federation Gateway Profile*. Each of these profiles is implemented on a separate virtual machine (VM). We name these VMs as *virtual gateways*. A gateway profile defines the configurations and services that a virtual gateway has. In this case, a *configuration* can for example define, how many interfaces the virtual gateway has and how these interfaces are connected together. When we discuss about *services*, we refer to the different applications installed on the virtual gateway. These can be, for example, a DHCP server, a UPnP Media Server or a video transcoding software.

Since the configurations and services can vary, the experimental design is not limited to these three profiles. Arbitrary profiles can be supported. However, we assume that the home gateway profile, which is dedicated to home network users, is present on every gateway device by default. The different profiles are covered in detail in Section 5.5.2.

5.5.1 Domains

An overview of the experimental gateway architecture is provided in Figure 9. Our architecture contains two domain categories: the *physical gateway* and the *virtual gateway*. The physical gateway is a system started at the boot time of the gateway. It has access to the network interface cards (e.g. Wi-Fi, ZigBee and Ethernet) present at the gateway device. In Figure 9, the physical interfaces start with a letter *P*, for example, "Path1". In addition, the physical gateway has access to the virtualized interfaces. These virtual interfaces are each connected to one virtual Ethernet interface at a virtual gateway. In Figure 9, virtual interfaces at the physical gateway are named as *Vif*, for example "Vif1" and virtual Ethernet interfaces at the virtual gateways are named as *Veth*, for example "Veth1".

The system that manages the physical gateway domain is called the *Physical Gateway Manager (PGM)*. The PGM is an application instance running continuously on

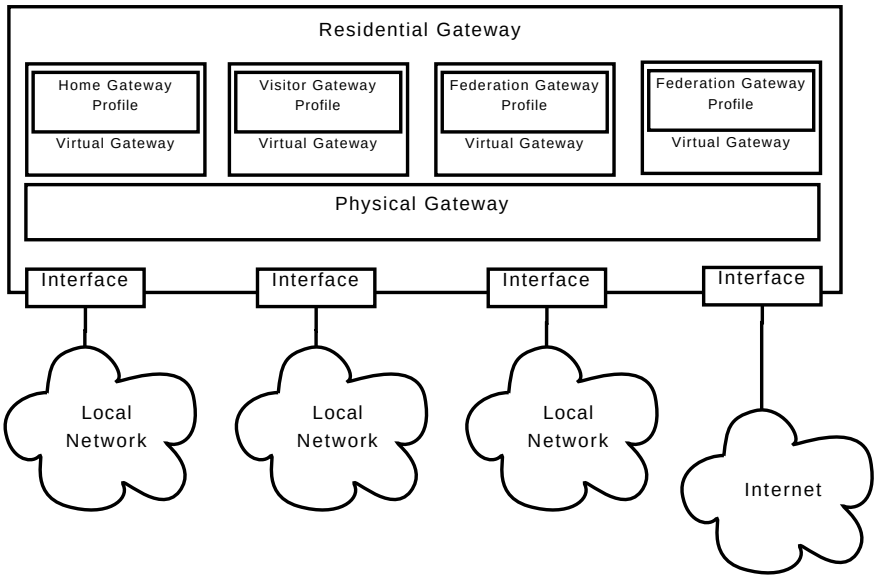

Figure 8: Virtualized Residential Gateway

the top of the gateway's operating system. It performs life-cycle management for the virtual domains, controls network interface cards (NIC), configures the network inside the physical gateway and does security and federation membership management. In addition, the PGM exposes a remotely accessible Application Protocol Interface (API) for controlling purposes. The PGM is described in more detail in Section 5.6.

A virtual gateway is a virtualized execution environment. Our design assumes hardware level virtualization. A virtual gateway has one or more virtualized Ethernet interfaces that are each connected to one virtualized interface at the physical gateway. In Figure 9, for example, the virtual Ethernet interface "Veth0" inside virtual gateway is connected to the virtual interface "Vif1" at the physical gateway. The communication between the physical gateway and a virtual gateway is performed through these interfaces.

The *Virtual Gateway Manager* (VGM) manages a virtual gateway. It is an application instance running continuously on the top of the virtual gateway's operating system. The VGM is used to implement a gateway profile, in other words, the configurations and services that are present at the virtual gateway. After the gateway profile is installed and the virtual gateway is running, the VGM is used for modifying the behavior of the virtual gateway. This means that the VGM is capable of configuring the network inside a virtual gateway and to install services in it. It can also be used for installing distributed file systems in a virtual gateway such as the one described in Section 5.1. The VGM is also responsible for the federation membership management and the security. However, as stated earlier, these two features were omitted. Similar to the PGM, the VGM exposes an API that is used for controlling the virtual gateway.

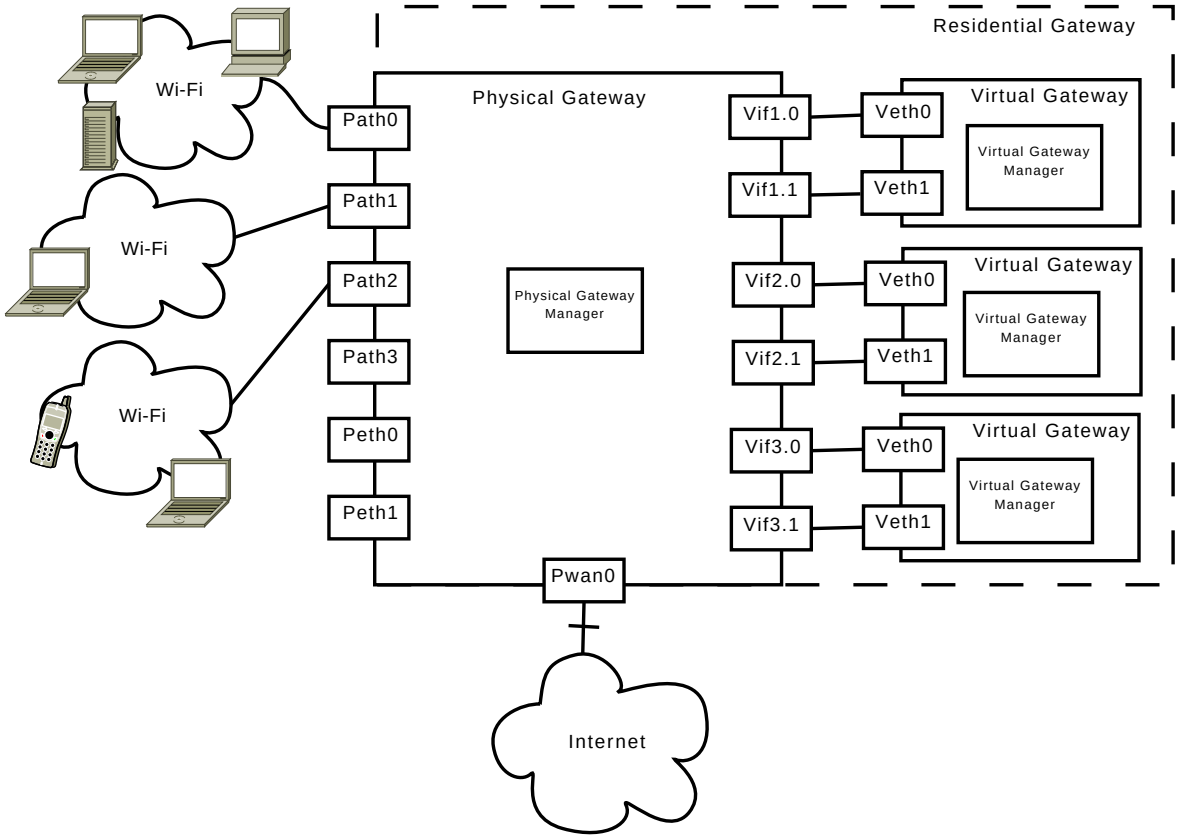

Figure 9: Overview of the experimental gateway architecture.

5.5.2 Profiles

Next we discuss about the three different type of gateway profiles: the home gateway, the visitor gateway and the federation gateway. These gateway profiles define the configuration and services available on a virtual gateway. All of these gateway profiles can be created using the VGM. In addition to these three profiles, arbitrary gateway profiles can be created using the VGM. However, we want to introduce these three profiles separately in order to provide suitable virtual gateway configurations and services to the visitor and home users.

The home gateway profile is dedicated to the usage of the home network. We name a virtual gateway that implements a home gateway profile as *home gateway*. Each application and service present at the home gateway is available for the home network users. In addition, the home gateway connects the home network users with federation services. In practice this means that home gateway can be used to access all distributed file systems belonging to different federations. Different link layer interfaces can be allocated for the usage of a home gateway profile-enabled virtual gateway. For example, users can allocate an Ethernet interface and a Wi-Fi access point for the home gateway users. An example of the home gateway profile is given in the section 6.4.

The visitor gateway profile is used for providing a complete virtual gateway for the usage of a single visiting user – or household. The visitor gateway profile could be used, for example, when friends or relatives are visiting a home. The same services that are present at the visitor's home gateway can be provided by using the visitor gateway profile. We call a virtual gateway that implements a visitor gateway profile as *visitor gateway*. The visitor gateway can be connected to all of the visitor's federation overlay networks. In addition, the visitor can install arbitrary applications to this dedicated gateway. Since the federations are the same as at the visitor's home gateway and the user is free to modify the gateway by installing arbitrary applications, the gateway appears similar to the visitor's home gateway. In order to use home network devices, a VPN application can be used. VPN can be used to provide a site-to-site VPN connection from the visitor gateway to the home gateway. This, among other possibilities, enables a remote control of network-enabled devices at home.

Each visitor gateway is accessed through a dedicated Wi-Fi access point or Ethernet interface. Hence, in order to use a visitor gateway, the visitor has to first authenticate herself on an *authentication interface* that is accessible through an authentication Wi-Fi access point, then leave the shared Wi-Fi access point and finally connect to the dedicated Wi-Fi access point or Ethernet interface. For details, we refer the reader to Section 6.5 where a detailed example of the visitor gateway profile is given.

While the visitor gateway is allocated to a single visitor and provides the visitor with all the federations she is a member of, *the federation gateway profile* is allocated for a usage of a single federation. A federation gateway profile connects to a single federation overlay network in order to provide access to the content of the federation. The owner of the physical gateway can decide whether she wants to allocate one or more virtual gateways for the usage of a federation. More virtual gateways are allocated to federations, more resources are used at the gateway. Thus, a gateway owner (home user) needs to select a subset of federations instead of allocating virtual gateways for each federations she belongs to.

All the visiting users who are members of the federation that the federation gateway profile implements can use this gateway simultaneously. Thus, the visitor who are close to the physical gateway can utilize this virtual gateway by using e.g. Wi-Fi. For example, the members of the "Paris Food" federation could access to the federation services through all the gateways that implement a federation gateway profile for that specific federation. The number of simultanous connections to the virtual gateway could be restricted to a certain number in order to restrict the size of provided resources in densely populated areas. An example of the federation gateway profile is provided in Section 6.6. We name a virtual gateway that implements a federation gateway profile as *federation gateway*.

In the case of the visitor gateway profile, we briefly introduced the authentication Wi-Fi access point. Access to the federation gateways is provided though this same access point. Users are able to use their federation gateways immediately after the authentication. Thus, unlike in the case of the visitor gateway, where the visitor

gets a private Wi-Fi access point after authentication, federation gateway users do not have to disassociate from the authentication Wi-Fi access point in order to use their federation gateways.

5.5.3 Interfaces

Three different types of interfaces, wireless, wired and virtual, are included in the experimental architecture. Other interfaces common to the residential gateways such as RJ11, SFP and USB ports are not considered, but are assumed to exist. In Figure 9 wireless interface are named as "Path", wired interfaces are named as "Peth" and virtual interfaces are named as "Vif".

The system functionality requires at least two separate *Wi-Fi access points* on the gateway. [1] The first Wi-Fi access point is dedicated to the home gateway users and it is connected to the home gateway. The second Wi-Fi access point is allocated for the visitors. The second Wi-Fi access point has two different functions. First, it is used for authentication. All new visitors connect to this access point and provide their credential. Second, this Wi-Fi access point is used to connect to federation gateways. After the visitor is authenticated, an access to one or several federation gateways can be provided through the same Wi-Fi access point.

In addition to wireless interfaces, an array of *Ethernet interfaces* is provided. As discussed in Section 4.3.3, a typical residential gateway contains a switch of 4 or more Ethernet ports. Our design does not set requirements for the number of Ethernet interfaces present at the gateway.

In order to provide the gateway device with an Internet access, a *Wide Area Network interface* (WAN) is required. We assume that in the near future, a majority of the homes connect the Internet using a single link. Hence, our design assumes a single WAN interface present at the gateway even though, as stated in the section 2.2.2, some residential gateway may support dual WAN interfaces is the near future.

In addition to physical interfaces, our experimental design uses virtual interfaces. These interfaces are provided by the virtualization system. For each virtual interface at a virtual gateway (Veth interface) a correspondent interface exists on the physical domain (Vif interface). PGM connects physical interfaces and the virtual interfaces together using different network tools.

5.6 Physical Gateway Manager

Based on the previous discussion regarding the distributed file system, application management, security infrastructure and membership management, we introduce an experimental design for the Physical Gateway Manager. As stated earlier, the PGM is the main system implementing the logic on the physical gateway domain.

[1] In case necessary, Wi-Fi access points can be replaced with a set of Ethernet interfaces. This special setup can be useful when no Wi-Fi cards are present at the gateway device.

An overview of the different components of the PGM is shown in Figure 10.

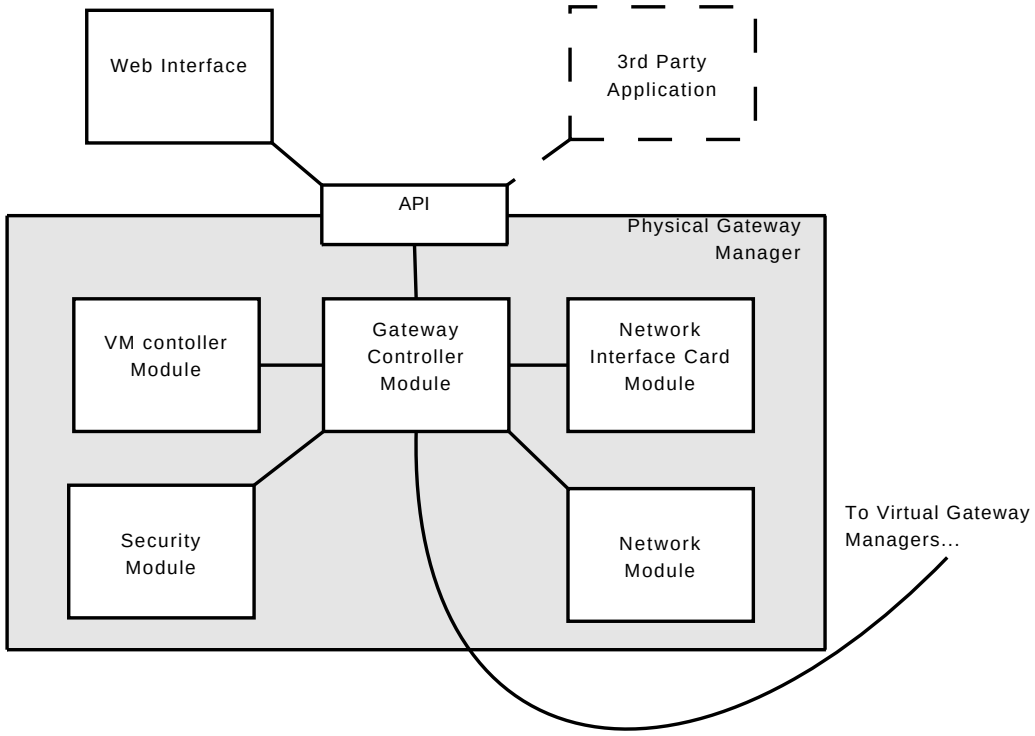

Figure 10: Physical Gateway Manager Architecture

5.6.1 Gateway Controller Module

The Gateway Controller Module is the core part of the PGM implementing the logical operations on the physical domain. It takes commands through the API and controls other modules of the PGM. The Gateway Controller Module communicates with the VGM instances located in different virtual gateways in order to install new gateway profiles in fresh virtual gateways and to modify existing ones. The Gateway Controller Module exposes an API that provides the following functions:

- Create a virtual gateway based on a virtual gateway profile
- Configure the network of the physical gateway for a virtual gateway
- Start, pause and delete a virtual gateway
- Authenticate to a federation

5.6.2 Network Interface Card Module

The Network Interface Card (NIC) Module controls the NIC devices present at the gateway device. This can include, among others, Ethernet, Wi-Fi, ZigBee,

HomePNA and MoCa interfaces. The NIC Module aggregates different NIC devices under the control of a single management unit and communicates with various device drivers.

5.6.3 Network Module

The Network Module handles the internal network configuration of the gateway device. It connects physical network interfaces to virtual ones in order to provide virtual gateways with an access to the local network and the Internet. The Network Module utilizes software bridge and firewall tools and is also responsible for providing DHCP services for visiting federation users. Based on the virtual gateway profiles present at the gateway, different network configurations can exist at the physical gateway. A set of different network configurations will be presented in Chapter 6.

5.6.4 Virtual Machine Controller Module

The Virtual Machine (VM) Controller Module controls the virtual machine hypervisor. This includes creating, starting, pausing, stopping and deleting virtual machines. By default, the same operating system is used on each gateway profile as discussed earlier. However, in order to retain the flexibility and assure the support for emerging services, other operating systems should also be supported. In order to enable this, the VM Controller Module can be used for installing arbitrary operating system images in the virtual gateways.

5.6.5 Security Module

The Security Module is used for authenticating visiting users who attempt to use the services of the residential gateway. In addition to authenticate visitors, the gateway must be able to determine, which federations a visitor belongs to. The different aspects of the security infrastructure were discussed in Section 5.3.

5.7 Virtual Gateway Manager

We proceed to describe the Virtual Gateway Manager, located on each virtual gateway. The VGM is responsible for controlling the virtual gateway. Thus, it is the main gateway application, on top of the operating system. Different modules of the VGM are presented in Figure 11. Each of the modules is discussed in detail in the following sections.

5.7.1 Gateway Controller Module

The Gateway Controller Module is the core component of the virtual gateway manager. It manages other modules and takes orders from the Web Services API it

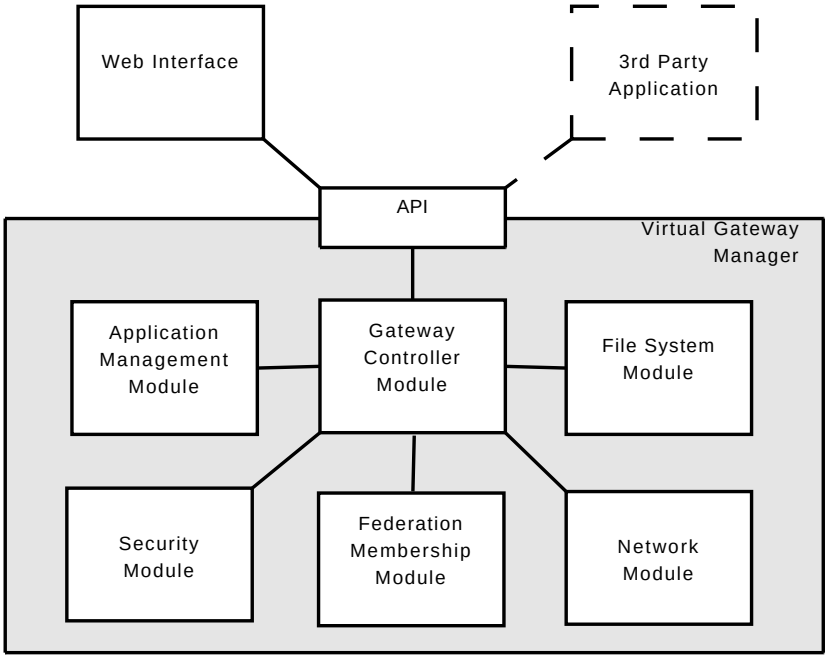

Figure 11: Virtual Gateway Manager Architecture

exposes. This API exposes the following functions:

- Configure the network of the virtual gateway
- Create a federation
- Invite users to a federation
- Connect and disconnect to/from a federation
- Install, start, stop and delete an application

When a new virtual gateway is created, a virtual gateway profile is loaded in a fresh system using the API. The profile defines how the network is configured, which federations the gateway belongs to and which applications are installed. The communication between the virtual gateway controller and the physical gateway controller is done using the Web Services API. Other mechanisms could be used as well, such as an internal communication system of the hypervisor. However, in order to keep the design hypervisor independent and easily portable, a Web Services-based solution is chosen.

5.7.2 Network Module

The Network Module is similar to the one in the PGM. It manages the internal network configuration of the virtual gateway device. It connects virtual network

interfaces to each other. The Network Module utilizes software bridge and firewall tools and is also responsible for providing dynamic IP addresses for the gateway users.

The network configuration depends on the used virtual gateway profile. The network configuration of a virtual gateway is performed in the following manner. First, the PGM connects to the VGM by using a Web Services interface. A SOAP message is sent, which describes the network configuration. Then the Network Module performs the network configuration based on the description. This includes, among others, setting up IP forwarding between two interfaces, or bridging interfaces together. Different example network configurations are presented in the chapter 6.

5.7.3 File System Module

The File System Module manages distributed federation file systems. When a virtual gateway joins a federation, the Gateway Controller Module makes the File System Module connect the federation's distributed file system. File System Module retrieves the federation certificate from Security Management Module. The address of the gateway implementing the file directory server and tracker server is provided to File System Module which then retrieves the file index from its local database. The cache devices where the federation content is stored must be defined as well. Once all this is done, the distributed file system is mounted in a directory of the operating system.

The file system is being accessed using applications, such as web interface, UPnP Media Server or SSH. The distributed file system was discussed in detail in the section 5.1.

5.7.4 Application Management Module

The Application Management Module is designed for application management. The purpose of the Application Management Module is to provide an unambiguous way to manage federation application installation and removal in virtual gateways. As the distributed file system, an application manager is introduced on each virtual gateway.

The application installation files can be stored to the distributed file system, to the Internet or to a home network device. For each application a single installation file is stored. The Application Management Module is controlled via Gateway Controller Module using a Web Services interface. The location of the installation file is provided for the Application Management Module in order to download and install a new application. Since all federations are independent on each other, the same application can be installed in several virtual gateways that are located to the same physical gateway. The details related to application management were discussed in 5.2.

5.7.5 Federation Membership Management Module

The Federation Membership Management Module manages the memberships of a federation. In case the virtual gateway is a normal member of the federation, federation membership module is used for joining and leaving federations and sending federation invitations.

5.7.6 Security Management Module

The Security Management Module implements the security functions of the virtual gateway. This covers authentication with individual users and with the federation. As in the case of the Federation Membership Management Module, the Security Management Module can have two roles. It can behave as a certificate authority in its federation, or it provides basic security functions, such as authentication and encryption. The Security Management Module is used by the Federation Membership Management Module and File System Module.

5.8 Summary

In this chapter, the core components of our gateway design were presented. First the distributed file system, the application management and the security system were introduced. Then two major components were described: the Physical Gateway Manager and the Virtual Gateway Manager. The Physical Gateway Manager was created to control the life cycle management of virtual gateways. In addition, the Physical Gateway Manager is used for the access control, when visiting users try to utilize the gateway. The Virtual Gateway Manager was designed in order to handle the gateway configurations and management. The Virtual Gateway Manager makes it possible to connect a virtual gateway to federation file systems and enables application management.

The design presented here is the first effort to address a portion the requirements that were identified in Chapter 4. More emphasis was put on the distributed file system and virtualization part. These two subareas were considered essential in order to address the federation-based content and resource sharing and the visitor access domains. We believe that these two domains of the future home network vision have a dominant impact to the gateway design. The other design parts were discussed in less detail.

The large goal of this thesis is to better understand the concept of the future home network. Thus, in order to see the general view clearly, we decided to design a relatively large amount of different components. Therefore, for most of the components a fairly light approach was taken with respect to the level of details. Several underlying requirements and research problems are related to each component that we did not address in this chapter. These are outside the scope of the thesis.

The final outcome of this chapter is a design that is experimental and targeted to

research use. It can be used as a reference when deciding which research problem to address in a greater detail. It can also be used as a concrete starting point for the next phases of the design. Overall we believe that the design is a next step from the home network vision towards a concrete gateway design that addresses that vision. In the next section, we will implement a prototype in order to evaluate the design.

6 Prototype Implementation

Based on the experimental design presented in the section 5, we implemented a prototype. The purpose of the prototype is to evaluate the feasibility of the overall concept and design decisions as well as to provide directions for the future work.

6.1 General Framework

The components of the implementation are presented in the figure 12. Our prototype assumes four network interface cards to be present: two Ethernet cards and two Wi-Fi cards. We used Linux operating system with Xen-enabled kernel. The system had four Xen domains. The first domain implemented the physical gateway domain and the rest of the domains were dedicated to virtual gateways. We implemented three different profiles: a home gateway profile, a visitor profile and a federation profile.

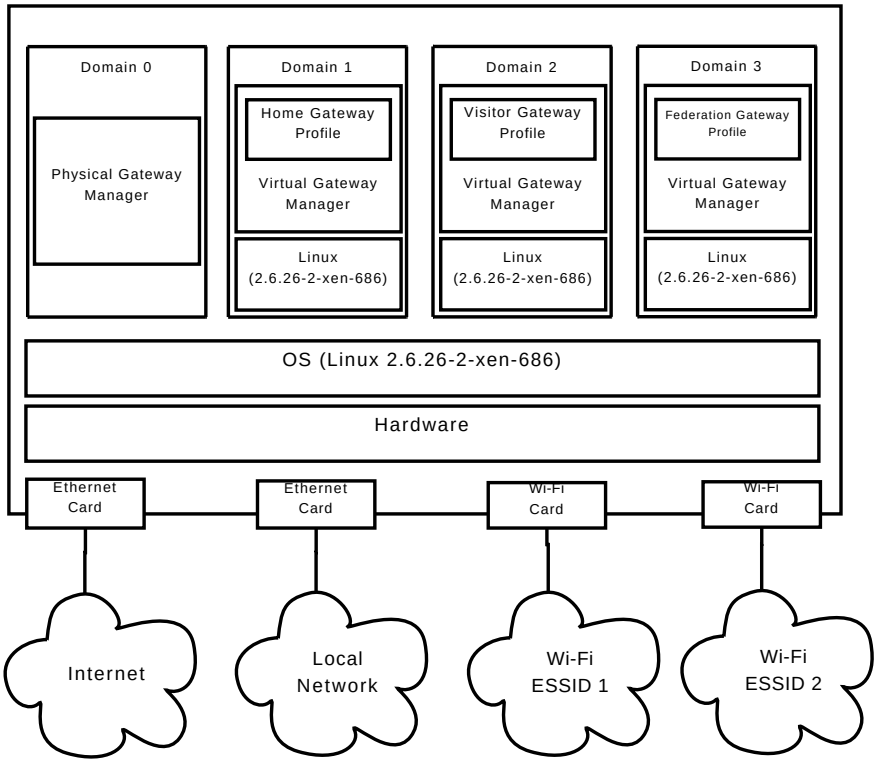

Figure 12: The components of the prototype implementation

The implementation is portable in a sense that it can be deployed on any hardware that is capable of running a Linux-based operating system and has the required interfaces. In addition, the virtualization technique can be changed easily, in case needed. The prototype implementation builds on existing technology. There technologies are discussed briefly in the following sections.

6.1.1 Python

We selected Python language to implement all the system components of the prototype. Python is an interpreted and object-oriented programming language which has simple syntax that makes it possible to create applications fast. Thus, Python fits well for prototyping purposes. Other attractive features of Python are its large standard library and the extensive third-party library base. Python supports countless of library bindings for various kind third-party systems which make it easy to manage diverse services under a single application [71].

6.1.2 Xen

Our prototype uses Xen hypervisor[11]. We selected Xen because it is open source and that it is a proven and mature virtualization technology. However, lighter virtualization systems exist that could be also suitable for the required purpose. For example, OpenVZ is a popular and light container-based virtualization technology [64]. The advantage of Xen over the lighter systems is its capability to run various type of operating systems. The lighter solutions, such as OpenVZ, require that each virtual domain has the same operating system. In the case of the prototype, a lighter solution could have fitted as well. In future versions, the needs of emerging services have to be taken into account. For example, a requirement to use a specific OS would require the use of a virtualzation technique such as Xen.

6.1.3 Libvirt

We used LibVirt toolkit for controlling various different virtualization systems [49]. LibVirt is capable of using the majority of the commands provided by the most common virtualization platforms. This includes, among others, creating, starting and stopping virtual machines. The toolkit provides APIs for various different programming languages including C, Java, C#, Ruby and Python. Libvirt gives modularity over the tools bundled to the different virtualization solutions. In practice, Libvirt makes it possible to switch between the different hypervisor system by changing one line one in the application code.

6.1.4 Django

The user interface parts of the system are implemented using Django web framework. “Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design” [84]. Django forces the developer to follow model-view-controller architectural pattern when designing web systems. Model-view-control paradigm separates the logic, data and presentation from each other which makes the application code modular. Django follows also Don’t Repeat Yourself (DRY) principle which forces the developer to program application code without unnecessary duplication. Since the prototype is programmed with Python, Django

was chosen. Several other web frameworks for Python exist such as Pylons, TurboGears and web2py. Django framework was chosen because of it provides a fast way to develop web sites and because it is well documented.

6.2 Physical Gateway Manager Implementation

The Physical Gateway Manager is implemented according to the design described in Section 5.6. An overview of the implemented system is provided in Figure 13. All different modules of PGM are discussed separately in the following sections.

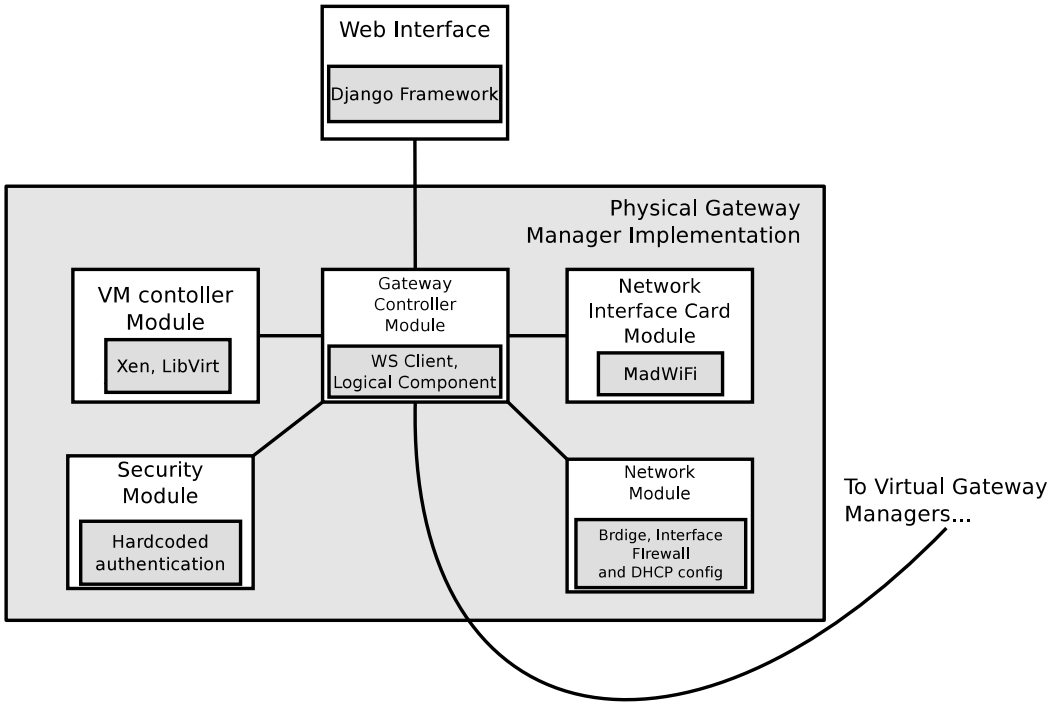

Figure 13: Physical Gateway Manager implementation

6.2.1 Network Module Implementation

The Network Module was described in Section 5.6.3. In the implementation, we divide the module into four separate sub-modules. These are the Interface Configurator, the Bridge Configurator, the Firewall Configurator and the DHCP Configurator.

The Interface Configurator calls *ifconfig* command to configure network interfaces. For example, the Interface Configurator can be used for creating and deleting interfaces, allocating IP addresses interfaces and defining the interface operation model, such as the promiscuous mode. The PGM uses the Interface Controller functions to manage virtual and physical interfaces.

The Bridge Configurator is used for managing software bridges. It is a similar system command wrapper as the Interface Configurator. The Bridge Configurator calls the *Bridge-utils* tool to create Linux Ethernet bridges and to add interfaces to them.

The Firewall Configurator configures firewall settings. It uses *Iptables* application and *Netfilter* Linux module. A set of pre-configurations were implemented such a script that configures a NAT between two interfaces and a script that maps ports from one interface to another.

The DHCP Configurator implements a DHCP service by calling *dhcpd* application. A separate DHCP service description file is used for defining how IP addresses are allocated. The Physical Gateway Controller uses the DHCP Configurator to provide IP addresses for visiting users.

6.2.2 NIC Module Implementation

The NIC Module was described in Section 5.6.2. In our implementation, the NIC Module is used for controlling Wi-Fi cards. MadWifi drivers are used to control Atheros type of Wi-Fi cards. NIC Module uses *wlanconfig* and *iwconfig* tools for configurations. NIC Module provides methods for establishing Wi-Fi access points. The MadWiFi driver provides a mechanism to establish up to 4 separate Wi-Fi access points using a single IEEE 802.11 card.

6.2.3 Virtual Machine Controller Module Implementation

The description of the Virtual Machine Controller is provided in Section 5.6.4. The Virtual Machine Controller uses *LibVirt* library to control virtual machines. The Virtual Machine Configuration is defined in a separate XML file. This XML file describes the features of the virtual machine, such as the used kernel, where the disk and swap images are located and how many interfaces are provided.

In the prototype, OS images used in the virtual gateways are stored to the system beforehand. In the future systems, this could be done dynamically so that OS images are downloaded to the gateway over the Internet. The XML file also defines a static IP address for the virtual machine interface. This makes it possible to locate the virtual gateways without additional discovery mechanisms. Because the number of virtual gateways is small in our prototype (typically less than 10), it is reasonable to configure the IP addresses of the virtual gateways statically at Virtual Machine Controller Module.

6.2.4 Security Module Implementation

The Security Module was briefly discussed in Section 5.6.5. The implementation of the Security Module provides a simple authentication mechanism. The Security Module compares user names and passwords with a user profile database. The

user profile database contains the *user name, password, full name* and *the list of federations*.

In the prototype, authentication data is hardcoded. In the future, this module would communicate with the security infrastructure in order to authenticate the user and learn her gateway's address. Also, the security infrastructure would be utilized to learn which federations the visitor belongs to. The security infrastructure was briefly discussed in Section 5.3.

6.2.5 Gateway Controller Module Implementation

The Gateway Controller Module uses the previously introduced system modules in order to provide the functions of the PGM. The Gateway Controller Module was introduced in Section 5.7.1. Its implementation is divided into two sub-components. The first component is called *Web Services Client Component* which is used for communicating with Web Servces interfaces. The second component is called *Logical Component* which implements the functions that the PGM provides. These functions were introduced in Section 5.7.1.

In the implementation, the Web Services client component is used for communicating with the VGMs. It exploits the methods of the VGMs that are exposed using Web Service Description Language (WSDL). The Web Services framework is established using ZSI toolkit [75].

The logical component exposes a set of methods for external usage and calls other modules in order to implement the functionalities of these methods. The functions can be initiated by two instances. The first instance is when the start-up script of the gateway, which calls different methods in order to initialize the gateway. The second instance is the web interface, which provides users with various functions. The start-up operation is described in Section 6.2.6 and the web interface operations are described in Section 6.2.8.

6.2.6 System Operation at the Start-up Phase

When the gateway is started up, the following operations are executed by using the Gateway Controller Module:

1. Configure a shared Internet access for the virtual gateways
2. Start a home gateway
3. Configure a local network for the home gateway
4. Start federation gateways
5. Start DHCP service on authentication interface
6. Start the web interface on the authentication interface

The figure 14 summarizes the interaction between the modules during the start up. These phases are discussed next in detail.

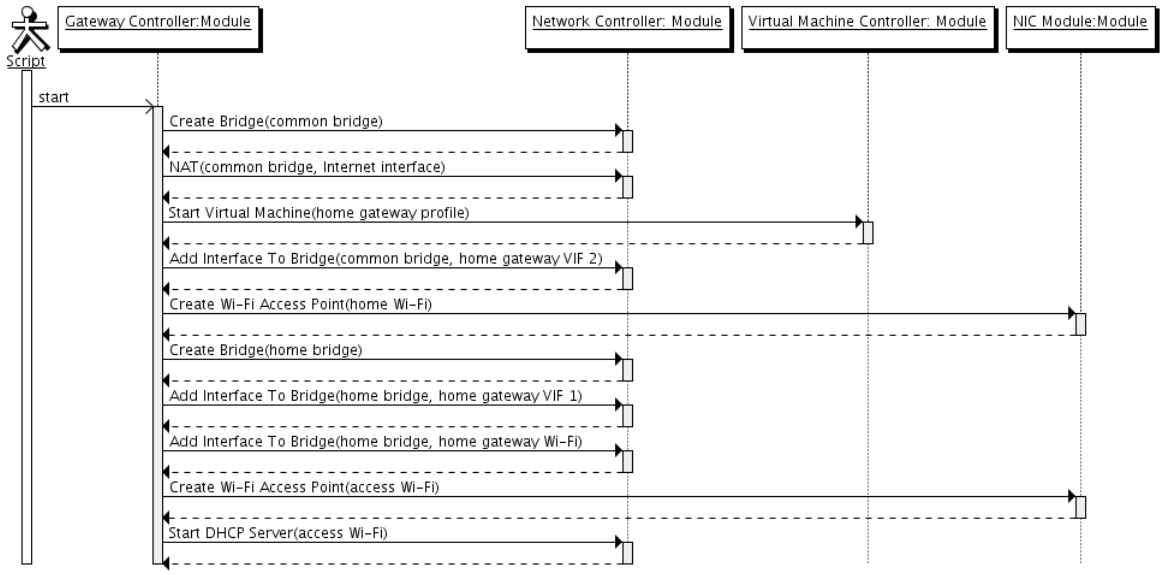

Figure 14: The start-up operation

Since our design assumes only a single IP address for the residential gateway, the Internet access have to be shared among the different virtual gateways. To provide a shared Internet access, the Gateway Controller Module first creates a network bridge by calling the Network Module (phase 1). The bridge (br0) is illustrated in Figure 15. One interface from each virtual gateway is attached to this bridge by using the Network Module. Then, IP forwarding is configured between the bridge (br0) and the Internet interface (wan0) by using the Network Module. This provides every virtual gateway with an Internet access. Since the virtual gateways are behind the NAT, port forwarding from the Internet interface (wan0) towards the virtual gateway interfaces (vif interfaces) must be configured in case virtual gateways run server applications. In our implementation, this is done by the Network Module and each port forwarding rule is customized manually for each server application that is run inside the virtual gateways. In the future systems, mechanism such as UPnP Internet Gateway Device Protocol (UPnP IGD) could be utilized [41] in order to let the applications configure port forwarding on the physical gateway domain.

A home gateway is created by calling the Virtual Machine Controller Module which creates the virtual gateways using Xen (phase 2). The Virtual Machine Controller Module uses an XML configuration description that defines the properties of the virtual gateway. The XML configuration description was discussed in 6.2.3. In the case of the home gateway, two network interfaces are provided. After the home gateway is up and running, the first network interface of the virtual gateway (vif1.1) is attached to the shared bridge (br0 in Figure 15).

To provide a local network for the home gateway, the Gateway Controller Module uses the NIC Module to establish a Wi-Fi access point for the home users. This is

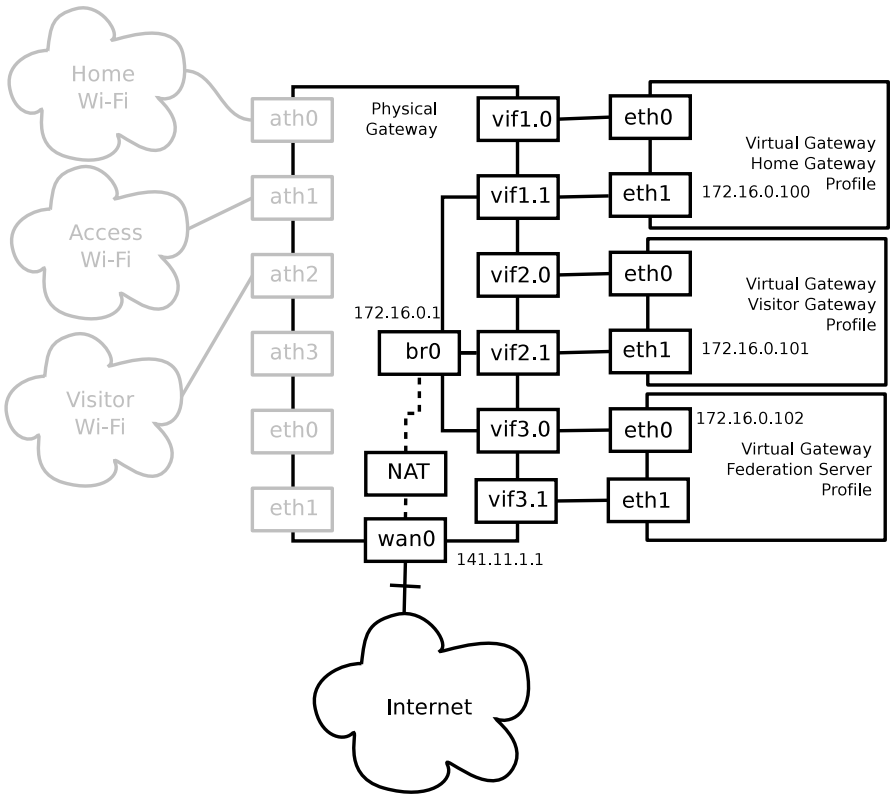

Figure 15: Network address translation between virtual gateways

illustrated in Figure 16. Once a Wi-Fi access point is up, the Gateway Manager Module calls the Network Module to create a bridge (home br) between the home gateway interface (vif1.0) and the new Wi-Fi access point interface (ath0).

In addition to the access point that is allocated to the home gateway ("Home Wi-Fi" in Figure 16), another access point is created for the visiting users ("Access Wi-Fi" in Figure 16). As described in Section 5.5.2, this Wi-Fi access point is used to authenticate all visiting users but also to access the federation gateways. (Remember that some visitors can be provided with a visitor gateway that has its own Wi-Fi access point). After the interfaces are up, the Gateway Controller Module starts a DHCP service on the visiting interface (ath1) by calling the Network Module.

6.2.7 Web Interface

The web interface is used by the visiting gateway users. As stated in the previous section, the web interface is started during the start-up phase and it can be accessed through the visiting user interface ("Access Wi-Fi" in Figure 16). Unlike our design (Section 5.7.1), the prototype does not implement a Web Services API at the PGC. This functionality was considered less critical in the context of a proof-of-concept implementation. Thus, the web interface calls the methods of the Gateway

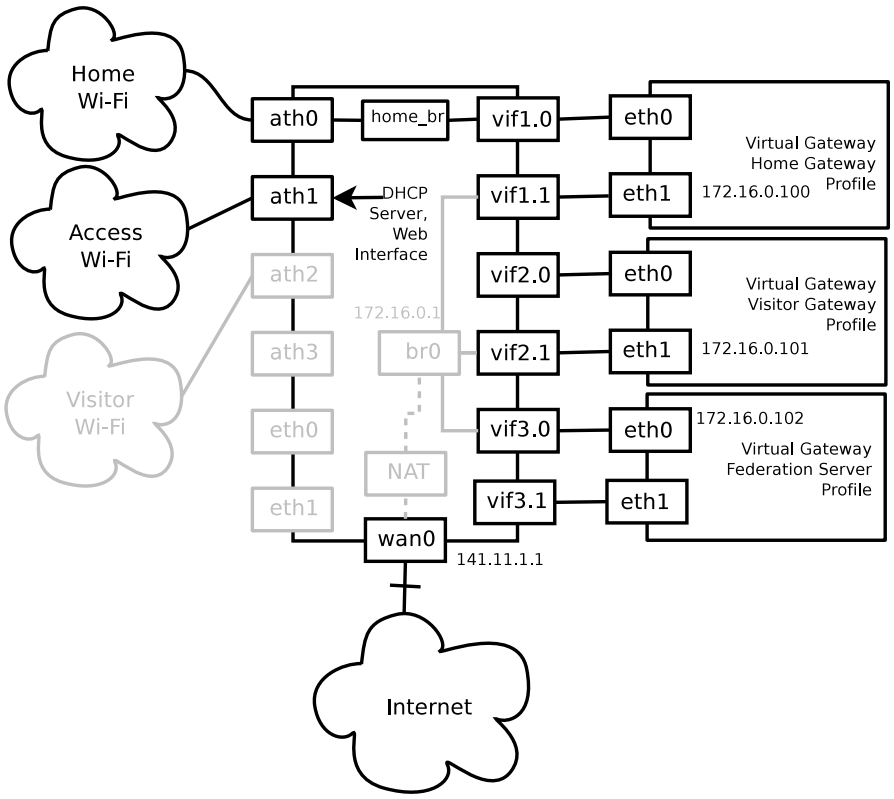

Figure 16: Bridging and DHCP server in the physical gateway

Controller Module directly (by using Python methods).

The web interface is illustrated in Figure 28 and Figure 29 in Appendix A.1. The first view allows a visiting user to authenticate herself to the system. The second view provides an interface that has the following functions:

1. Use Internet
2. Use VPN
3. Use federation gateways

The function 1 provides the user with an Internet access. The function 2 starts a visitor gateway that runs a VPN client application. Finally, the function 3 makes it possible to access different federation gateways. These different function are discussed in detail next.

6.2.8 System Operation Initiated by the Web Interface

The web interface uses the methods of the Gateway Controller Module to provide services for visiting users. The following operations are initiated in the PGM in order to address the high-level functions provided by the web interface:

1. Authenticate visiting users
2. Provide Internet access to visiting users
3. Provide access to federation gateways
4. Start visitor gateways
5. Configure Internet access for visitor gateways
6. Configure local network for visitor gateways

Each of these phases are discussed next in detail.

Visiting users have to authenticate themselves before accessing the gateway services (function 1). To make this possible, an authentication web page is displayed (see appendix A.1). In the prototype implementation, users need to type a local IP address (192.168.100.1) to their browser in order to view the web interface. In future versions, a captive portal could be provided which makes it possible to force a web browser to access a specific web page when the browser is launched. The captive portal was discussed in Section 4.2.4.

The authentication process is illustrated in Figure 17. When a user provides her credentials, the Gateway Controller Module queries the Security Module which consults a local database. As mentioned in Section 6.2.4, the credientials are hardcoded in our prototype. In case the user name exists and the password is correct, the corresponding user profile is returned. In case the user name or the password is incorrect, an error code is returned and the web interface asks the user to authenticate again. In the case of a successful authentication, a cookie is stored to the user's browser to enable a web session. After that, the user is directed to another web page (see A.1). The functions of this page were described in Section 6.2.7.

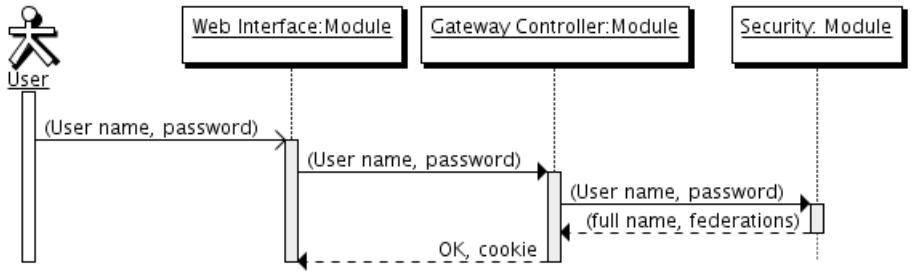

Figure 17: The module interaction in authentication process

In the prototype, each visitor is provided with an Internet access (function 2) and each visitor has access to every federation gateway (function 3). A captive portal solution could be used in order to control to which interfaces the visitors can access. However, this functionality was not considered important in the context of a proof-of-concept prototype.

When a visiting user clicks the "use VPN" link on the web page, the Gateway Controller Module calls the Virtual Machine Controller Module. The related interaction is described in Figure 18. A dedicated XML configuration description of the virtual gateway is used when a visitor gateway is launched (function 4). In the prototype implementation, this means that the virtual machine has two network interfaces. Also, the system image of the virtual gateway contains a VGM instance, which is launched at the boot time. In the prototype implementation, a specific VPN application is also started at the boot time. This configuration will be described in detail in Section 6.5. The visitor gateway is attached to the shared bridge (function 5) which was described earlier in Figure 16. This provides the visitor gateway with an Internet access.

Once the visitor gateway has an Internet access, the Gateway Controller Module uses the NIC Module in order to allocate a Wi-Fi access point for the visitor gateway. A bridge is created between the access point and the interface of the visitor gateway (function 6). This configuration is identical with the local network configuration of the home gateway, which was described in Section 6.2.6. After the interfaces are connected together using the bridge, a new web page is displayed to the visiting user. This page informs the user about the access point (SSID) that is associated to the freshly created visitor gateway.

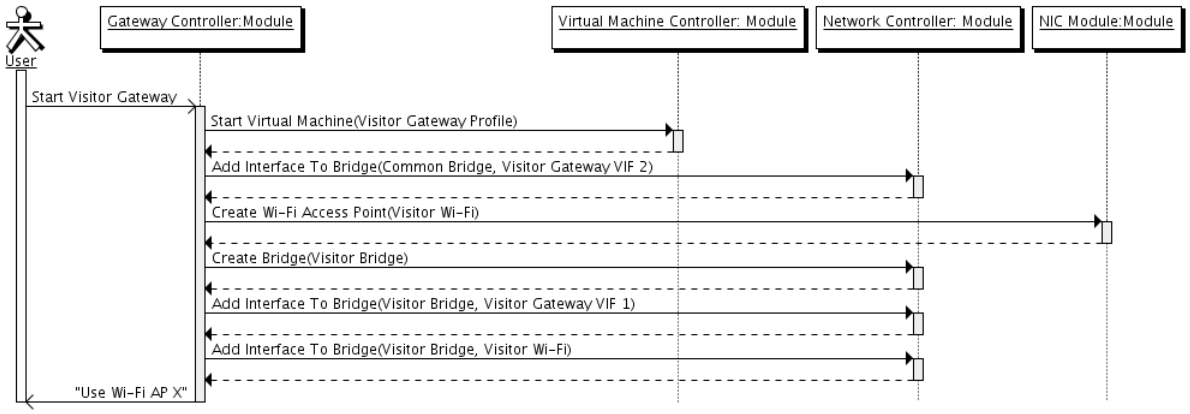

Figure 18: The module interaction in a visitor gateway start up

This concludes the Physical Gateway Manager implementation and operation. We proceed to describe the Virtual Gateway Manager implementation.

6.3 Virtual Gateway Manager Implementation

The Virtual Gateway Manager is a daemon process running inside each virtual gateway. It bundles all the logic that is required to implement different types of virtual gateways. When a fresh virtual gateway is created, a gateway profile, such as a home gateway or a federation gateway is created by using the VGM. Different profiles were introduced in Section 5.5.2. The VGM connects the virtual gateway to federations, configures the internal network of the virtual gateway and installs

applications into the system. As stated in the design chapter, the VGM consists of several sub-components. An overview of the Virtual Gateway Manager is given in figure 19. Each component is introduced in the following sections.

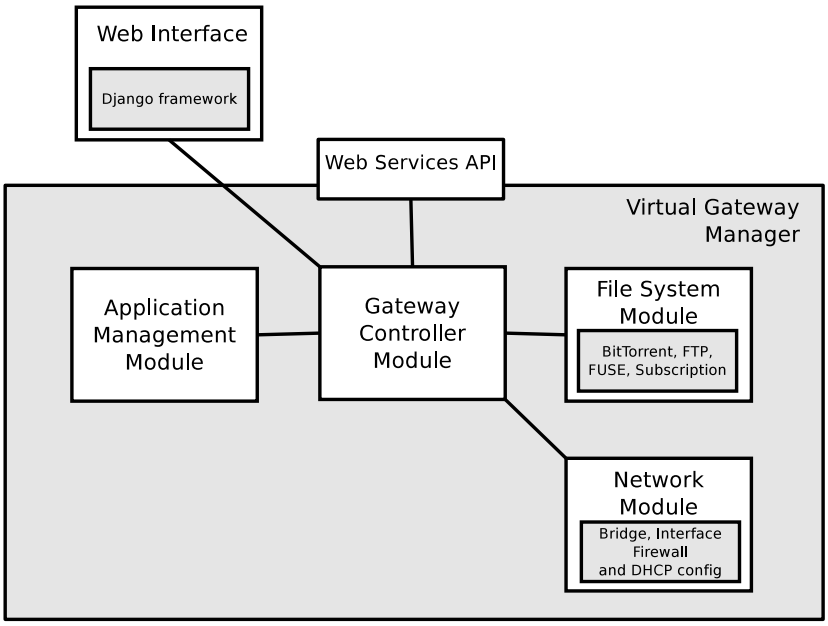

Figure 19: Virtual Gateway Manager implementation

6.3.1 Network Module Implementation

The Network Module used in the VGM is the same implementation as used in the PGM. As described in Section 5.7.2, the Network Module configures the software bridges and the firewalls inside a virtual gateway. If required, it can be also used for providing a DHCP service.

6.3.2 File System Module Implementation

The file system module consists of five sub-modules, which are the BitTorrent module, the FTP module, the peer-to-peer control module, the FUSE module and the subscription module.

BitTorrent Module is implemented using the existing Python modules provided by Bram Cohen in his BitTorrent implementation [7]. These modules are used for implementing the trackers and peers of the peer-to-peer overlay. In addition, the torrent files are created using the BitTorrent modules.

The FTP Module consists of several sub-modules. It is used to implement the file directory server. On each founder peer, a separate FTP server is used. This FTP server was implemented using pyftpdlib library [70]. Torrent files are created locally and uploaded to the FTP server. Each time peers want a list of files, the FTP server

is contacted and the list of torrents are returned. When a user wants to download a file, she will first download the torrent file from the FTP server and then contact the BitTorrent tracker with the hash value inside the torrent file. Section 5.1.1 describes the file directory server design.

P2P Control Module implements methods for managing the distributed file system. In practice it contains the API methods of the distributed file system methods, which were described in Section 5.1.5.

FUSE Module (Filesystem in Userspace) [20] is used to implement the file system abstraction functionality (described in Section 5.1.4). The FUSE Module connects the functions of the P2P Control Module to the Linux file system. Similar relation was described in the design chapter in Figure 5. The FUSE Module is called each time an operation, such as read a file or make a node, is executed at a certain Linux directory. When the FUSE Module is started, it will attach itself to a Linux directory. The operations that the operating system would normally perform are done by the FUSE Module.

Subscription Module design was introduced in Section 5.1.6. This module periodically monitors a distributed file system directory. In case the module notices that a new file appears in the directory, it will execute the "touch" command. In the prototype, this command is used for making the distributed file system to download the new files to the local directory. This functionality is implemented inside the FUSE Module. In other words, the new file, which is actually just an entry in the FTP directory, is downloaded to the peer's local hard drive. In the prototype, this hard drive is located physically to the gateway device. The purpose of this operation is to download interesting content automatically to the home network, before actually consuming it.

6.3.3 File System Operation

The file system functionality is illustrated in Figure 20. When a user stores a file to the file system, it is first stored to a local directory (1). In this case, the FUSE Module calls the upload function provided by the P2P Control Module. The P2P Control Module stores the file to a local directory which corresponds to the cache system described in Section 5.1.8. Then the P2P Control Module creates a torrent file of the file by using BitTorrent Module (2). Once the torrent file is done, the P2P Control Module uploads it to to the FTP server by calling the FTP Module (3). After that the P2P Control Module calls the BitTorrent Module to register the torrent to the tracker (4). In case user wants to create a directory first the FUSE Module is called. The FUSE Module calls the P2P Control Module which first creates the directory into the cache system (5) and then to the FTP server by calling the FTP Module (6).

The file reading operation is described in Figure 21. When a user wants to read a file, the FUSE Module calls the P2P Control Module which downloads the corresponding torrent file from the FTP server by calling the FTP Module (1). Once the torrent

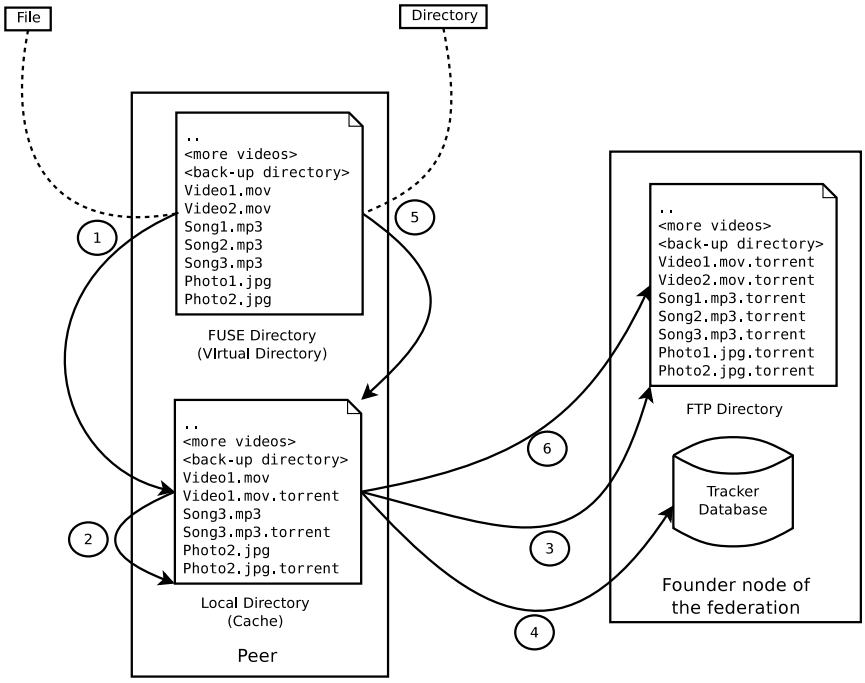

Figure 20: Storing a file and creating a directory

is downloaded the P2P Control Module sends the hash value inside the torrent file to the tracker by using the BitTorrent Module (2). The tracker answers with the addresses of the peers having chunks of the file (3). The P2P Module then uses the BitTorrent Module to download the file chunks from these peers (4). Once the file is downloaded completely, the P2P Module returns it to the FUSE Module (5). The file can then be accessed using the file system.

When a user wants to get the listing of the files in a directory, the file list is fetched from the FTP server by the P2P Module which uses the FTP Module. In case some of the files exist locally on the cache system, the P2P Module provides the FUSE Module with the file size and the date information available in the cache system. Otherwise, the size and date data is fetched from the FTP server, which in this case, however, refers to the torrent file details.

The current file system implementation has some limitations. For example, the cache directory keeps copies of all the local files that are being shared. Also, when a file is not downloaded to the local directory, the size and date information is based on the torrent file in the FTP directory. In addition, the FTP directory is queried each time a file system operation is executed on the FUSE directory, which can cause delay, scalability and performance issues. However, the main purpose of the implementation is to create a system that can be used for evaluating the concept as a whole. As stated in the design chapter, the distributed file system is a too large research problem to be addressed completely in this thesis that attempts to evaluate several aspects of the next generation residential gateway concept.

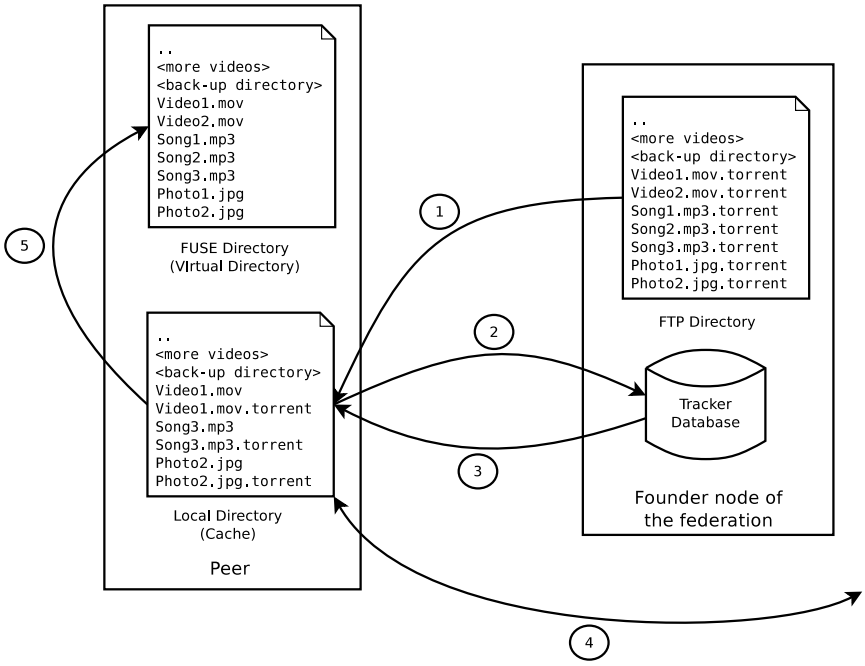

Figure 21: Reading a file

6.3.4 Federation Membership Management Module Implementation

The purpose of the Federation Membership Management Module was described in Section 5.7.5. However, the prototype implementation does not provide this module. The Federation Membership Management Module is not in the scope of this thesis.

6.3.5 Security Module Implementation

The purpose of the Security Module was described in Section 5.7.6. For the same reason as in the case of the Federation Membership Management Module, this module was left as unimplemented.

6.3.6 Application Management Module Implementation

The Application Management Module is used for installing applications to a virtual gateway. This module was introduced in Section 5.7.4. A specific installation file format is used when handling the applications inside a federation. An installation file is a group of files compressed into a single *tar.gz* archive. In addition to the binaries or source code, the archive contains a text file that has the instructions to install, run and access the application. In practice, the installation and execution instructions are Linux command line scripts. The access point address that describes the address where the application can be accessed is a TCP/UDP port that the application is listening – once its running.

The Application Management Module operates as follows. First it copies an installa-

tion file from the distributed file system to a temporary directory. Unlike described in Section 5.2, the prototype does not support downloading the installation file from the Internet or the home network devices. Once the installation file is downloaded, the module runs the installation command and the running commands written into the instruction file. Finally, the Application Management Module returns to the Gateway Controller Module with an access point address so that application can be utilized.

6.3.7 Gateway Controller Module Implementation

The Gateway Controller Module is the core component of the virtual gateway manager. The design for the Gateway Controller Module was introduced in Section 5.7.1. It uses other modules in order to control the virtual gateway. Some of the methods of the Gateway Controller Module are exposed using the Web Services technology.

The following functions are provided:

- *Found a Federation.* In order to host a federation, the following input must be provided: credentials, local FTP directory, FTP port, BitTorrent tracker port.
- *Connect to Federation.* The following parameters are required in the federation connection procedure: address of the founder, credentials, FTP port, BitTorrent tracker port, local directory where the federation directory is mounted
- *Configure Network.* In order to configure the network, a description of the network configuration is required.
- *Install Application.* The input needed for installing an application is the location of the installation file in the federation directory.

In the following sections, we describe three different virtual gateway profiles that were implemented in the prototype; the home gateway profile, the visitor gateway profile and the federation gateway profile. A set of network configurations and applications specific to these profiles are introduced.

6.4 Home Gateway Profile

A home gateway profile configuration instance is implemented in order to illustrate the concept. The home gateway profile is dedicated to home users. It implements some functions of a typical residential gateway. In addition, a home gateway profile-enabled virtual gateway connects the home users to their federations.

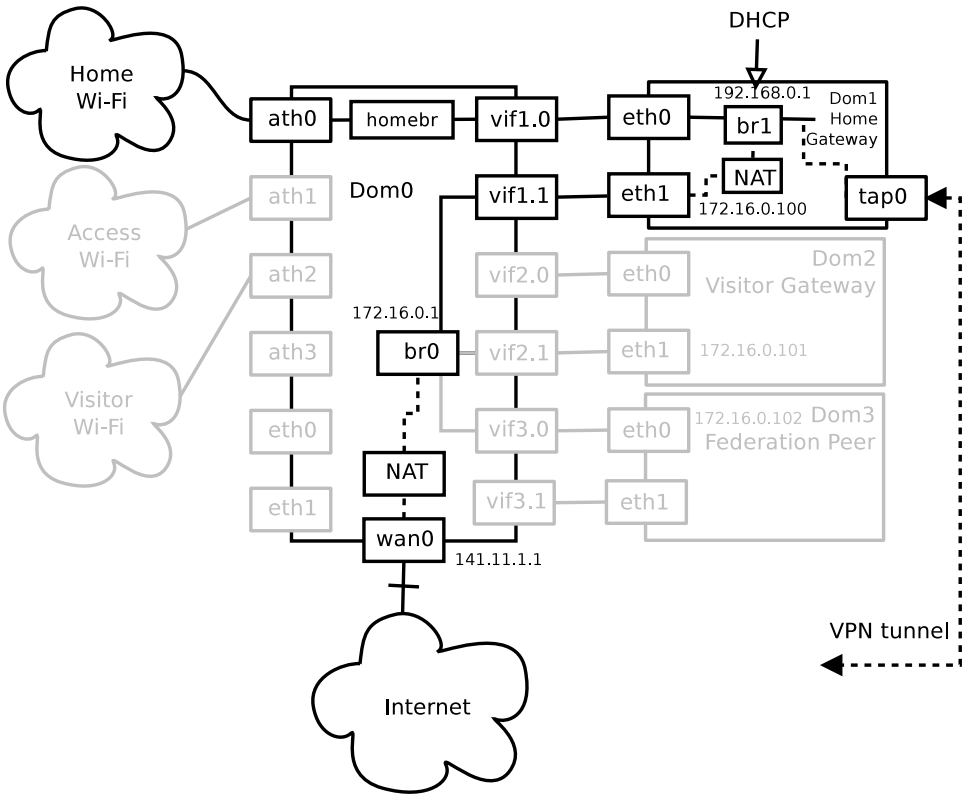

Figure 22: The network configuration of the home gateway profile

6.4.1 Network Configuration

The network configuration of the home gateway is presented in Figure 22.

The first interface (eth0) is connected to a bridge (br1). In order to provide remote home users with a VPN access to the home network a TAP interface (tap0) is attached to the bridge as well. This TAP interface is a virtual interface which the VPN application listens to. Everything sent to TAP interface is encapsulated into a VPN message and sent to the other end of the VPN tunnel via the second interface (eth1). In order to provide the remote users with an access to the VPN server, port mapping needs to be configured from the Internet interface (wan0) towards the home gateway's public interface (vif1.1). As discussed earlier in Section 6.2.6, this configuration is done manually at physical gateway domain. Once the network is configured inside the virtualized home gateway, a DHCP service is started at the bridge interface (br1). In order to provide home users with an Internet access, IP forwarding is performed between the bridge (br1) and the second interface (eth1).

6.4.2 Applications

In addition to the DHCP service, other applications are installed in the home gateway. All applications are installed in the home gateway manually. However, in

future versions these applications could be installed in the gateway using the application installer module. The installed applications include an UPnP client and a web application which provides file browsing and application installation functions.

A UPnP client application called Djmount [89] is used to mount the UPnP content available at the home network to a federation directory. Djmount "mounts as a Linux filesystem the media content of compatible UPnP AV devices". In the federation context, Djmount makes it possible to share home network content within federations.

A web interface for the federation file system was implemented and deployed on the home gateway. In practice, it is a web application that makes it possible to browse Linux file directories using a web browser. The web interface is used to browse distributed file system directories and add content to them. In addition, the web interface supports the subscription functionality. Users can select distributed file system directories and all new content that is added to these directories are immediately downloaded to the hard drive of the local gateway. The design for the subscription mechanism was described in Section 5.1.6. The web interface can also be used for installing applications to the gateway. Screenshot pictures of the web interface are shown in the section A.2 in Figures 30 and 31.

6.5 Visitor Gateway Profile

The visitor gateway profile provides a single virtual gateway for the exclusive usage of a visiting user or household. In theory, a copy of the visitor's home gateway could be implemented on a visitor gateway profile. However, in the prototype system, the visitor gateway provides the visitor with a VPN application that creates a tunnel to the visitor's home gateway.

6.5.1 Network Configuration

The network configuration of the virtual gateway profile is provided in Figure 23. The first interface (eth0) and a TAP interface (tap0) are attached to a bridge (br0). The TAP interface behavior was explained in the section 6.4.1. IP forwarding is performed between the bridge (br0) and the second interface (eth1) in order to provide the visitor user with the Internet access. Note that DHCP service is not used at the visitor gateway but it is provided via the VPN tunnel. In practice, the TAP interface is a single hop distance from the other end of the VPN tunnel. This makes it possible to utilize services that work only on a single LAN. Such services include, among others, UPnP and Bonjour.

6.5.2 Applications

OpenVPN [63] application is selected for the VPN application. OpenVPN is an open-source application which provides a VPN system utilizing either layer 2 (TAP)

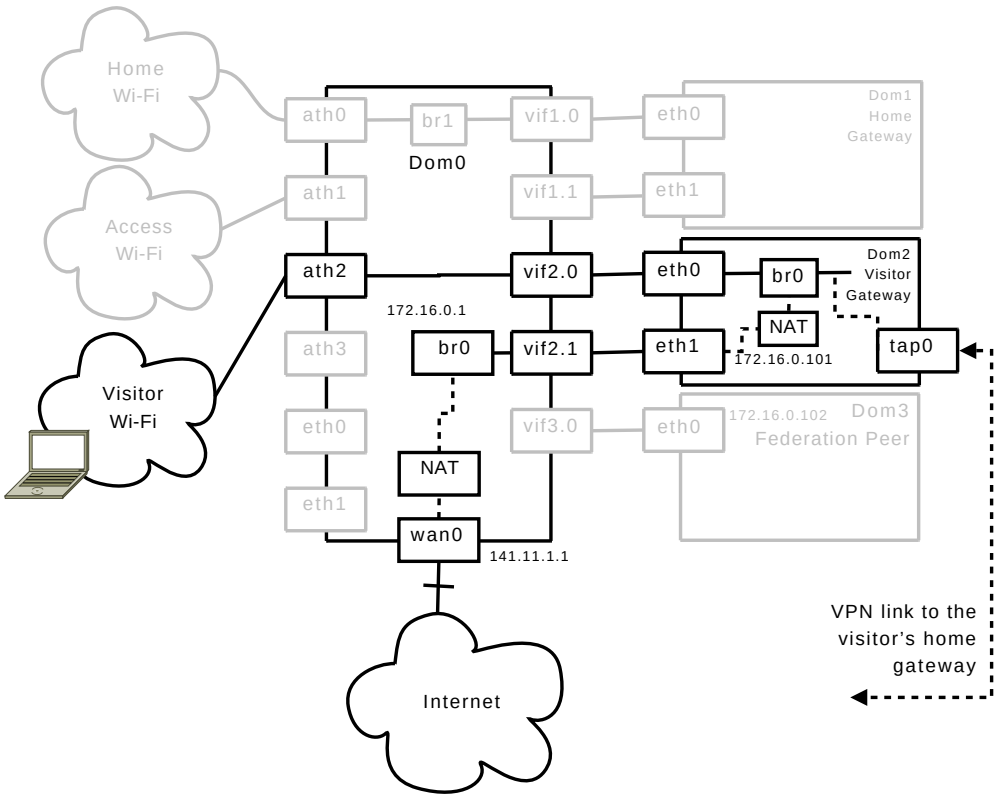

Figure 23: The network configuration of the visitor gateway profile

or layer 3 (TUN) interfaces in order to use the VPN service. The software encrypts the packets using SSL.

6.6 Federation Gateway Profile

The federation gateway profile is used at a virtual gateway to provide the visiting users with the services of a specific federation. In the prototype implementation, the federation gateway is attached to a single federation.

The network configuration of the virtual gateway profile is provided in the figure 24. The visiting users access the gateway through the access Wi-Fi interface. The firewall is configured so that the users can access the external virtual interface (vif3.1) and then the second internal interface (eth1) of the federation gateway. The prototype does not have a captive portal. Hence, each user can access the federation gateways. We consider this limitation insignificant in respect to the objectives of the prototype implementation.

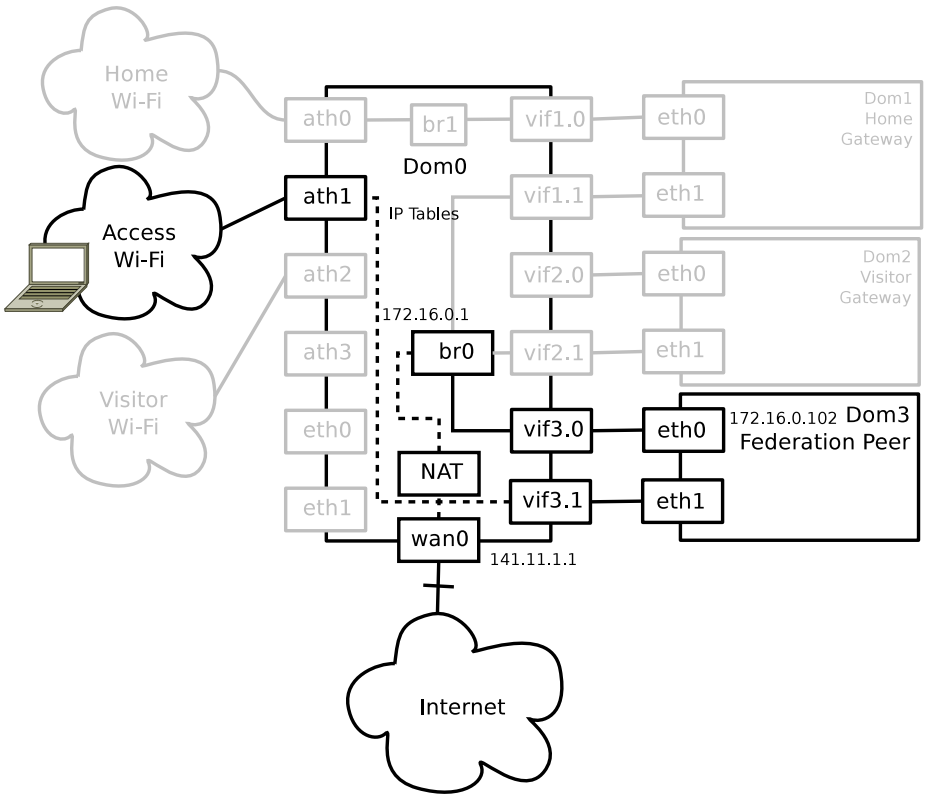

Figure 24: The network configuration of the federation gateway profile

6.6.1 Network Configuration

6.6.2 Applications

A federation file system is mounted in the federation gateway. Then, a web based file manager application is started at the internal interface (eth1). It is the same application, described in Section 6.4.2, that is utilized for managing files and installing applications.

A video transcoding application was implemented in order to demonstrate the concept of the application installer. The transcoder application contains the source code of a transcoding program called FFmpeg [19] and a web interface for utilizing FFmpeg via web browsers. The web interface first asks the user to upload a file to the gateway from her client device, such as laptop. Then, the application transcodes the video into a compact-sized format. After the transcoding, the user must specify where the transcoded video is stored in the federation file system. See further details in Section 7.4

6.7 Summary

In the chapter, a prototype implementation was presented. The prototype was created using existing technologies, such as Xen hypervisor and Django web framework. Two major components, the Physical Gateway Manager and the Virtual Gateway Manager were implemented. The prototype implemented three different virtual gateway profiles. These were the home gateway profile, the visitor gateway profile and the federation gateway profile.

The implemented prototype is in line with the design. It has a modular architecture and we believe that it provides a flexible framework for the usage of future research. The architecture aims to support gradual development. Hence, some modules can be re-implemented in a more optimized manner and integrated into the existing prototype. In addition, missing modules such as the Security Module can be integrated into the existing prototype afterwards. The prototype is evaluated through these profiles in the next chapter.

7 Evaluation

The main purpose of this chapter is to verify that the implemented prototype functions correctly. We conduct three use case tests. The first test evaluates the functionality of the visitor gateway profile introduced in the implementation chapter. In the first test, VPN application is used in order to connect the home network remotely. The second test concentrates on evaluating the federation overlay concept by utilizing the distributed file system for video sharing. The third test verifies the correct functionality of the application manager. We describe the experimental set up and then we step through the three test cases.

7.1 Evaluation Environment

The evaluation environment consists of two desktop computers which act as the residential gateway devices and two laptop machines which acts as clients for the residential gateway devices. We emulate DSL link by using a LAN. This approach was chosen because of the time constraints set for the work.

The gateway devices has the following setup:

- 1.73 GHz Intel Pentium processor
- 0.5 GB of RAM
- OS in all domains: Linux 2.6.26-2-xen-686 (Debian Lenny)
- Two Ethernet cards
- Two Atheros Wi-Fi cards

Both laptops have one Wi-Fi and one Ethernet card. The first laptop had Linux OS and the second one had Windows OS.

7.2 Test 1: One-hop Video Streaming

The test case is illustrated with a use case:

> *"Alice is visiting some friends and wants to show them the video of her daughter. Alice can easily access and view the movie through her friend's gateway".*

The purpose of this test is to evaluate the behavior of the visitor gateway. The setup for the test is described in the figure 25. The laptop 1 runs a UPnP media server called MediaTomb which contains a video file. The laptop 1 is connected to a Wi-Fi access point of the home network via its home gateway. The home gateway profile

is used in the gateway device 1. The laptop 2 runs an application called XBMC [98] which implements UPnP media renderer and controller, thus its able to view content from UPnP media servers. The laptop 2 is visiting a virtual gateway in the gateway 2. The visitor gateway profile described in the section 6.5 is utilized. This means that a VPN tunnel is created between the two gateways so that they are logically in the same LAN. In the prototype, the OpenVPN configuration is hardcoded. Thus, the required security certificates are pre-installed and the destination address of the home gateway is pre-configured in order to provide the VPN conncetion. The goal of this test is to stream the video from the home network device to the visiting user's device.

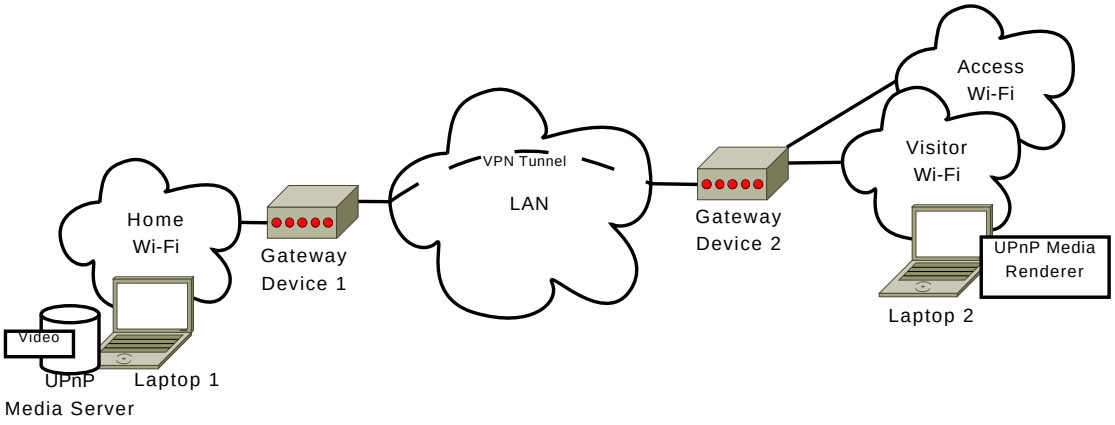

Figure 25: The test case 1.

7.2.1 Phases

We assume that the gateways and the laptops are up and running before the test. Each phase of the test is described next.

1. Start the home gateway at the gateway device 1 (OpenVPN is launched during the start-up).
2. Associate the laptop 1 to the home network Wi-Fi access point of the gateway device 1
3. Start UPnP media server on the laptop 1
4. Associate the laptop 2 to the access Wi-Fi AP of the gateway device 2
5. Using the laptop 2, open the authentication web page on the gateway device 2
6. Start a visitor gateway using the access web page on the gateway device 2
7. Using laptop 2 leave the access Wi-Fi access point on the gateway device 2

8. Associate the laptop 2 to the visitor Wi-Fi access point of the gateway device 2
9. Start UPnP media renderer and controller application at the laptop 2
10. Start viewing the video file using the UPnP media renderer and controller application at the laptop 2

7.2.2 Analysis

Once the laptop 2 associated to the Wi-Fi AP in phase 8 the IP address was correctly received over the VPN tunnel from the DHCP server running in the home gateway. The laptop 2 was able to ping the laptop 1 and vice versa. Once the UPnP media renderer and controller application was launched on the laptop 2, the application could see the UPnP devices at the home network and also the content of these devices (phase 9). Thus, UPnP protocol messages were delivered over the VPN tunnel. Finally, in the phase 10 the stream from the UPnP media server to the UPnP media renderer was successfully initiated over the Internet. The streaming worked without problems.

7.2.3 Discussion

The video streaming over VPN seems to be working well. However, in the test setup, LAN was used instead of DSL links which gives a biased view of the latency and bandwidth related issues. The DHCP service seems to be working over the VPN connection. Nevertheless, it took rather long before the laptop got an IP address. This can be related to issues in the bridge configurations.

The VPN application can be considered as a special case on the context of the next generation gateway. In fact, it does not provide any new features compared to existing gateway solutions. From the design perspective, the VPN application is a single application installed to the gateway using the application installer. Of course, some modifications to the existing VPN application needs to be done in order to make it compatible and usable in the context of the experimental design. For example, the application must communicate with the security infrastructure in order to manage the authentication and it must provide a user interface that can be accessed remotely.

Since VPN provides the user with a single hop to her home network, it can be useful in some occasions. For example, as done in the test, UPnP-enabled devices can be used over the Internet. However, the community overlay provides a more flexible access methods between home networks.

7.3 Test 2: Federation Based File Sharing

The second test case is illustrated with the following use case:

"Alice records a movie of her daughter's performance at a classsical music concert. This program will be made available to the friends and family through their federation."

In this test, federation-based file sharing is tested. The home user shares the video file within a federation using the web interface. The test setup is illustrated in the figure 26. As in the test 1, the laptop 1 runs a UPnP media server that contains video file. The laptop 1 is connected to the home network's residential gateway via Wi-Fi access point. Home network profile is used. At the boot time of the gateway, Djmount application is started, as described in the section 6.4.2. The web interface makes it possible for the home users to select UPnP content that is shared among the federation. The home gateway of the gateway device 1 and the federation gateway of the gateway device 2 belong to the same federation.

The laptop 2 has an application called VLC [95] for watching videos and it is connected to a federation gateway on the gateway device 2. Thus, the federation gateway profile described in the section 6.6 is utilized. The laptop 2 accesses to the federation file system using a web browser, downloads the video and watches it.

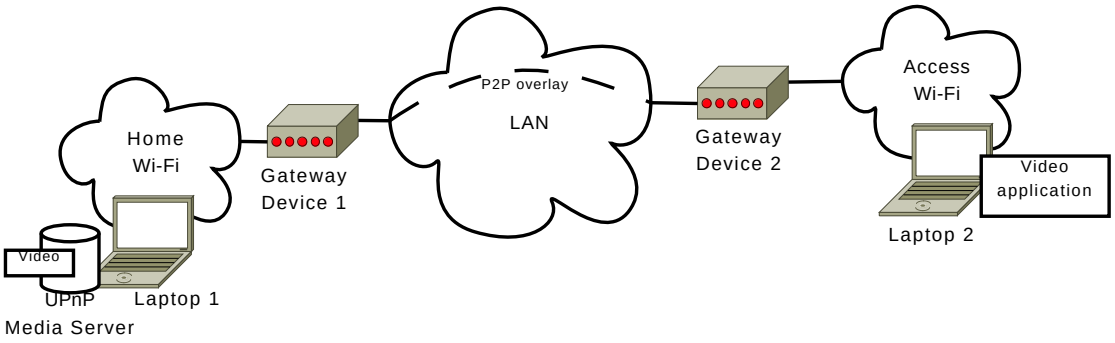

Figure 26: The test case 2.

7.3.1 Phases

Below, each phase of the test is described in detail.

1. Start the virtual home gateway on the gateway device 1 (Djmount mounts the UPnP content to the virtual home gateway)
2. Associate the laptop 1 to the home network Wi-Fi access point of the gateway device 1
3. Start UPnP media server at the laptop 1
4. Using the laptop 1, open a web browser and access the web interfaces running on the virtual home gateway (using a static IP address)

5. Open the “share UPnP content” page on the web interface
6. Using the laptop 1, select the video on the web interface and copy it to the federation file system directory
7. Associate the laptop 2 to the access Wi-Fi access point of the gateway device 2
8. Using the laptop 2, open a web browser and access the web interfaces running on the federation gateway (using a static IP address)
9. Open “browse federation files” page on the web interface
10. Click the file to open it on the VLC application.

7.3.2 Analysis

During the phase 5 we could see the UPnP media server on the laptop 1 by using the web interface. In addition, we could see all the content available on that UPnP media server, including the video that we wanted to share. Once we switched to the second laptop in the phase 9, the video shared in the home network was listed on the web interface. Thus, the file directory mechanism of the distributed file system seemed to be working well. Finally once we opened the video on the VLC application in the phase 10, it was successfully downloaded over the Internet and shown on the screen. However, the downloading time was rather long, thus we had to wait for a while before we could watch the video.

7.3.3 Discussion

The file system is designed in a way, that the whole file is downloaded before it is released for the usage of applications. Because the test file is fairly large (88.7 MB) and the uplink bandwidth of the peer-to-peer application is defined to be small, it takes some time before the video file can be viewed. If there would be more members in the federation having the same file, the downloading would be done faster. To cope with the long download delay when accessing the file for the first time, the subscription mechanism could be used here in order to download the video automatically once it appears in the federation.

In the test case, a web browser is required in order to access the video file at the federation gateway. Another option would be to mount the distributed file system in a UPnP media server. However, this is a potential security issue, since everybody on the same federation LAN are able to receive UPnP protocols multicast messages.

7.4 Test 3: Third-party Software on a Federation Gateway

The test case 3 is described using the following use case:

"Alice records a movie of her daughter's performance at a classsical music concert. Before she arrives at home, a compact version of the movie has already been transferred from the digital camera to a federation shared among her friends and family."

This test aims to install an application to a federation gateway and use it for transcoding a video file in to a smaller format. The idea is to provide the federation members with a small-sized version of a video before downloading the large and better quality video over the Internet. In addition, the subscription mechanism is used to download the file automatically to the gateway. The setup for the test is described in the figure 27.

The laptop 1 is connected to a Wi-Fi access point of the home gateway on gateway 1. A video transcoding application is shared with the federation at the home gateway. In addition, the laptop 1 makes the gateway 1 to subscribe to a directory in the distributed file system. The laptop 2 connects to the federation gateway of the gateway device 2. The laptop 2 makes the federation gateway to download the transcoding application from the distributed file system and installs it to the federation gateway. The laptop 2 uses the transcoding application to transcode a video. Once the video is transcoded, it is uploaded to the same directory of the distributed file system where the home gateway subscribed earlier. Thus, the transcoded video is downloaded to the home network immediately onces its added to the directory.

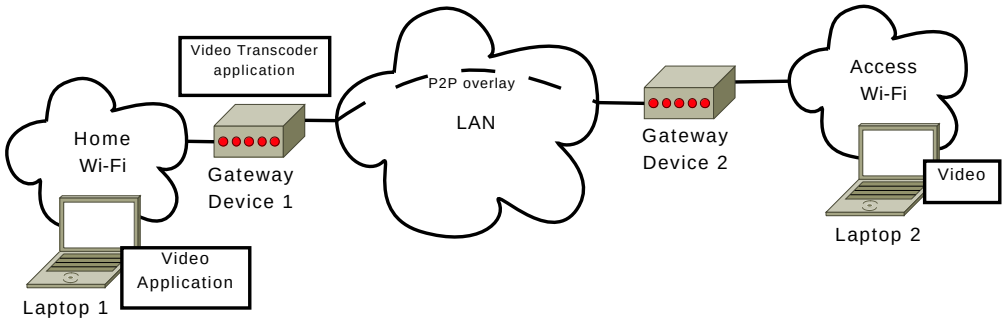

Figure 27: The test case 3.

7.4.1 Phases

The phases of the test case 3 are listed next in detail.

1. Start the home gateway on the gateway device 1
2. Associate the laptop 1 to the home network Wi-Fi access point of the gateway device 1
3. Using the laptop 1, open a web browser and access the web interfaces running on the home gateway (using a static IP address)

4. Open the “share local content” page on the web interface
5. Using the laptop 1, locate the transcoding application installation file on the local hard drive and upload it to a distributed file system directory
6. Using the laptop 1, subscribe to a distributed file system directory called “videos”
7. Associate the laptop 2 to the access Wi-Fi AP of the gateway device 2
8. Using the laptop 2, open a web browser and access the web interfaces running on the federation gateway (using a static IP address)
9. Open the “install application” page on the web interface
10. Select the installation file shared by the laptop 1
11. Install the application to the virtual federation gateway on the gateway device 2
12. Start the application on the virtual federation gateway
13. Using the laptop 2, upload a video file to the transcoding application and use the application to convert the uploaded video
14. Store the converted video file to a distributed file system directory called “videos”
15. Using the laptop 1, Open the “browse content” page on the web interface
16. Using the laptop 1, Open the federation directory called “videos”
17. Click the file to open it on the VLC application.

7.4.2 Analysis

During the phase 13 we could access the web interface of the transcoding application without problems. Once the video was converted and uploaded we could see the compact file version in the distributed file system directory where in subscribed in the phase 16. Finally, in the phase 17 we watched the file. Unlike in the test case 2, the video could be viewed immediately. Thus, the subscription mechanism seemed to be working well.

7.4.3 Discussion

One major issue noticed during the test was that the downloading and installation of the application takes a long time. One benefit of uploading a video through a federation gateway is to share the a video fast with others. Video can be quickly uploaded to the gateway using Wi-Fi and the transcoding and storing to the federation can be done without any intervention from the visiting user. In order to support this, either the application downloading and installation should be fast or the application is already installed to the gateway.

The idea of the video transcoding and the implemented subscription mechanism is to provide federation members with a local copy of each video in compact format. Thus, videos can be viewed immediately and if a video seems to be interesting, the better-quality version can be downloaded from the federation. An alternative approach would be to download the beginning of each video to the local storage system, and once the video is being watched, rest of the file is being downloaded.

7.5 Summary

In this chapter, we tested the prototype through three different use cases. The first test tested the visitor gateway functionality. The VPN application was successfully used in a virtualized gateway to create a site-to-site link between two gateways. The user was able to control UPnP applications remotely despite the fact that UPnP is a single hop protocol working only in a single LAN.

In the second test files were shared using the distributed file system. The content of a UPnP Media Server was shared with the federation and then viewed from a remote gateway. A separate application was used to locate the UPnP-enabled files in the home network and then a dedicated mechanism was used to share the files with the federation.

In the third test a video file was transcoded on a federation gateway and then transferred to the home gateway automatically using a subscription mechanism. The transcoding application was installed during the test and used for transcoding some content.

Our goal was to verify the correct functionality of the prototype, and these three tests verified the functionality of the core parts of the implementation: the distributed file system and the different virtual gateway profiles. In addition, the application manager was also tested.

8 Summary and Conclusions

Like running water and electricity, the Internet connectivity has become a commodity in modern homes. The Internet connection has led to the introduction of home networks. Unlike in the case of electricity where the usage is simple – just connect the plug to the socket – the way home networks are used varies based on the used home network technology. The diversity of the available technologies and services has made the management of the home network a complex task.

In addition to the complexity caused by the variety of network technologies at homes, content management causes trouble. During recent years, the amount of user-generated content present at homes and in the Internet has increased dramatically. To address this, some technologies have been introduced. DLNA, for example, provides certificates for devices that interoperate with other DLNA certified devices. However, this solution works only inside a single home network and does not address the fact that much of the content is now shared over the Internet.

In this thesis, we introduced a design for an experimental residential gateway that attempts to address some of the issues discussed above. In order to understand requirements for such system, we used the vision of the FIGARO project as a reference. The FIGARO project attempts to design a residential gateway-based architecture for the future Internet. The project describes its vision through a set of use cases which were analysed in this thesis. As a result of the requirement analysis, we constructed a requirement list for a next generation gateway. These were the *content and resource sharing sharing system, backup system, unified content management, execution environment, visitor access, security infrastructure* and *federation membership management system*. We analysed each of these requirements separately and discussed some of the possible solutions.

Based on these identified requirements we created an experimental design for the gateway. This design covered a rather large number of system components. Thus, several design decisions had to be taken. As a result of the work, a virtualized gateway system was introduced. In our design, a single gateway device supported several virtualized gateway instances. Two main components of the design were the Physical Gateway Manager, which manages different virtual gateway instances at a gateway device, and the Virtual Gateway Manager, which manages a single virtualized gateway. To provide different types of gateways, such as a gateway for home users and a gateway for visiting users we introduced the concept of the Virtual Gateway Profile. In order to address the content management, we introduced a distributed file system based on BitTorrent protocol.

The design was evaluated through a prototype implementation which was created using Python programming language and Xen virtualization. The core functions of the Physical Gateway Manager and Virtual Gateway Manager were implemented and their functionality were validated through user case-based testing. In future work, this prototype will be deployed to home environments. It will be utilized to evaluate the concept and to learn more about its possibilities and limitations.

At the beginning of this thesis, we defined the goal. Our task was to find an answer to the question: what kind of design is required for a residential gateway in order to support improved home network management, content management and the integration of emerging services. Next we analyse what was achieved.

Our design did not explicitly improve home network management. We did not introduce a home network management system that takes into account all the different home network technologies present at homes nowadays. However, we introduced a design that provides a platform for introducing services that support better home network management. Two features of our design endorse this statement. First, the Physical Gateway Manager provides a component (NIC Module) that can be expanded to support various home network technologies, such as ZigBee and MoCA. Second, the virtualized gateway paradigm makes it possible to install arbitrary configurations in the virtual gateway that can take into account the presence of new technologies. Once the underlying infrastructure is set up by using a system such as the one introduced by our design, a home network management service can be developed into a virtual gateway. The same mechanism also supports introduction of emerging services on the residential gateway device.

Our design provides means to perform content management. However, due to the lack of analysis, we cannot argue that our system is an improvement compared with existing solutions. We addressed content management with a distributed file system that used BitTorrent protocol for P2P-based data transmissions. This system acted also as a basis for the federation concept. The system provided a unified content view over a federation and the home network. Hence, the home network content and the content shared among the federation could be accessed in the same manner. To test the functionality, we implemented a simplified version of the distributed file system. The correct functionality of this implementation was tested through use case-based testing. After the implementation and tests we concluded that it is feasible to use a residential gateway-based distributed file system for content management between home networks.

The design of the distributed file system is not optimal. The FTP-based directory server, for example, represents a single-point-of failure. In addition, our system performs a query over the Internet each time the file system is used. Clearly, more optimized approaches exist. Also, several sub-components, such as the one managing content placement at the home network or the one that controls content management between federation members were designed on a high level. Several details that are required in order to design an optimized system were not discussed. Thus, we do not argue that our design is optimal in any means.

A next generation gateway has dozens of sub-systems which interact with each other. During our work, the first task was to identify what are those sub-systems. Then we examined how those systems interact with each other. These phases were revisited until a consistent map of the different sub-systems was build. After that we studied how these sub-systems could be designed. In many cases, we limited our examination to the conceptual level and concentrated on a small number of the sub-system's

most important interfaces or functions. For example, the file system had functions to store and read files but security mechanisms considering private file access were omitted. In the case of the application manager, simple application installation and running functions were designed but features such as application library dependency management were not considered. We tried to limit our focus on the sub-system details that had an impact to the overall system design. As a consequence, the deepness of the examination varied between sub-systems. For example, application manager uses the distributed file system to download an installation file but is by itself a relative autonomous entity and does not have much effect on the design of other components. On the contrary, the distributed file system can be seen as a central component for the federation concept. Thus, we decided to study the distributed file system in a greater detail in comparison to the application manager. All in all, one can argue that for each of the sub-system a separate Master's Thesis could be written in order to sufficiently cover the topic.

Another kind of approach could have been taken. We could have tried to identify only a small number of necessary components and then carefully design some of them. For example, the file directory mechanism of the distributed file system could have been designed extensively. Now we designed the whole file system infrastructure but the study was performed on a high level. Alternatively, we could have done an analysis for different virtualization techniques and then, for example based on measurements, we could have suggested the most suitable system for a residential gateway. Now we simply selected a well-known virtualization system Xen, and moved forward to study how it should be configured.

This thesis works as an input to the upcoming large-scale system project, FIGARO. It was the first proof-of-concept study that tried to evaluate the feasibility of the visions the project has. The approach we took, working on a large number of domains, had to be taken in order to sufficiently address this demand. The design described in this thesis must be revisited during the FIGARO project and each sub-domain must be studied and designed in a more precise manner than was done in this thesis. However, several sub-systems and their requirements are now identified. In addition, some potential ways to address these requirements are now being presented. We consider this to be an important contribution of this thesis.

During the work, we found out that many parts of the vision of the FIGARO project are feasible. There are major challenges, however, one of the most crucial is related to the usability of the system. As stated in the introduction, more than half of the Europeans are potential users of the gateway device. In addition, the usability has become a competitive edge in consumer electronics during the recent years when more and more device models are available on the markets. Traditionally, the residential gateway has been a device that is configured only once, the first time its used at home. A gateway, such as the one introduced in this thesis, requires more attention and interaction from its users than the present systems. Users have to be taught how to use the new type of residential gateway. Understanding the concept such as "federation" and "visitor user" can take time. Providing an easy way to manage all of the features introduced in this thesis is likely be one of the

most challenging parts of the future design.

The large number of new features does not only make the usability challenging. The remote troubleshooting becomes a complex task when the system goes out of order. Remote troubleshooting on residential gateways is currently performed using TR-69 protocol. Currently, help-desk calls cover a large portion of the ISPs expenditure. Once the gateway device starts to provide federation overlays and virtualization, the remote management might become a nightmare for the ISPs. The remote management features of a residential gateway might turn out to be the most significant factor when ISPs select the technology to be deployed to their clients homes. A feature-rich residential gateway may become a double-edged sword for ISPs: at the same time such a system can work as a competitive edge but also as a troubleshooting nightmare. How to design a system that satisfies the troubleshooting requirements is likely to be the key challenge for the project.

The usability and remote troubleshooting represent only a portion of the challenges that future research needs to solve. There is still a long way to go, before that dream becomes reality, where home network devices and content are easily managed, and where new services work in harmony with these devices. We believe that in this thesis we took a step towards the realization of that dream.

References

[1] MoCA Annual Report, 2009.

[2] 2Wire. 2Wire HomePortal Intelligent Gateways. Referred 31.8.2010. Web page. http://www.2wire.com/index.php?p=479.

[3] Allied Business Intelligence". Home automation systems revenue to approach $12 billion worldwide in 2015. Web Document. Referred 31.8.2010. http://www.abiresearch.com/press/1633-Home+Automation+Systems+Reve/nue+to+Approach+$12+Billion+Worldwide+in+2015.

[4] Apache Software Foundation. Apache River. Web Document. Referred 21.11.2010. http://incubator.apache.org/river/index.html.

[5] Apple Inc. Bonjour, may 2006. http://developer.apple.com/documentation/Cocoa/Conceptual/NetServices/NetServices.pdf.

[6] BACnet. BACnet Website. Web Document. Referred 31.8.2010. http://www.bacnet.org/.

[7] BitTorrent Inc. Web Document. Referred 31.8.2010. http://www.bittorrent.com/.

[8] Bly, S., Schilit, B., McDonald, D. W., Rosario, B., and Saint-Hilaire, Y. Broken expectations in the digital home. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems* (New York, NY, USA, 2006), ACM, pp. 568–573.

[9] CableLabs. CableLabs, Revolutionizing Cable Technology. Web Document. Referred 21.11.2010. http://www.cablelabs.com.

[10] Chen, J.-C., Jiang, M.-C., and wen Liu, Y. Wireless lan security and ieee 802.11i. *Wireless Communications, IEEE 12*, 1 (feb. 2005), 27 – 36.

[11] Citrix Systems. Xen Hypervisor. Web Document. Accessed on 31 August 2010. http://www.xen.org/.

[12] Cohen, B. Incentives build robustness in bittorrent. Tech. rep., bittorrent.org, 2003.

[13] Continua Health Alliance. Web Document. Referred 31.8.2010. http://www.continuaalliance.org/.

[14] Digital Living Network Alliance. Web Document. Referred 31.8.2010. http://www.dlna.org/.

[15] Dollimore, J., Kindberg, T., and Coulouris, G. *Distributed Systems. Concepts and Design*, vol. 3. Addison Wesley, 2005.

[16] Dynamic DNS service. Web Document. Referred 31.8.2010. http://www.dyndns.com/.

[17] Eurostat. Web Document. Referred 31.8.2010. http://epp.eurostat.ec.europa.eu/portal/page/portal/eurostat/home/.

[18] Facebook. Statistics. Web Document. Referred 31.8.2010. http://www.facebook.com/press/info.php?statistics.

[19] FFmpeg. Web Document. Referred 31.8.2010. http://ffmpeg.org/.

[20] Filesystem in Userspace. Web Document. Referred 31.8.2010. http://fuse.sourceforge.net/.

[21] Fink, J., and Manbeck, J. Tr-124 functional requirements for broadband residential gateway devices. Tech. rep., Broadband Forum, 2006.

[22] FON Technology S.L. Web Document. Accessed on 30 August 2010. http://www.fon.com.

[23] Free. WiFi MiMo. Web Document. Accessed on 31 August 2010. http://www.free.fr/adsl/pages/internet/connexion.html#/wifi-mimo.

[24] FreeWRT. Web Document. Accessed on 30 August 2010. http://freewrt.org/.

[25] Goleniewski, L., and Jarrett, K. W. *Telecommunications Essentials*, vol. 2. Addison-Wesley Professional, 2006.

[26] Google. Picasa. Web Document. Referred 31.8.2010. http://picasaweb.google.com/.

[27] Guttman, E. RFC 3224, Vendor Extensions for Service Location Protocol, Version 2, Jan. 2002.

[28] Guttman, E., Perkins, C., Veizades, J., and Day, M. Rfc 2608, service location protocol, version 2, 1999.

[29] Haber, A., De Mier, J., and Reichert, F. Virtualization of remote devices and services in residential networks. In *Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST '09. Third International Conference on* (2009), pp. 182 –186.

[30] HGI Mission. Web Document. Accessed on 31 August 2010. http://www.homegatewayinitiative.org/about/mission.asp.

[31] Hofrichter, K. The residential gateway as service platform. In *Consumer Electronics, 2001. ICCE. International Conference on* (2001), pp. 304 –305.

[32] Home Gateway Initiative. Web Document. Accessed on 30 August 2010. http://www.homegatewayinitiative.org.

[33] HomeGrid Forum. Referred 31.8.2010. Web page. http://www.homegridforum.org/.

[34] HomePlug Alliance. Press Kit. http://www.homeplug.org/news/press_kit/HomePlug_Electronic_Press_Kit.pdf.

[35] HomePNA. HomePNA Products. Referred 31.8.2010. Web page. http://www.homepna.org/products/product_types/.

[36] Ibanez, M., Madrid, N., and Seepold, R. Security management with virtual gateway platforms. In *Emerging Security Information, Systems and Technologies, 2009. SECURWARE '09. Third International Conference on* (18-23 2009), pp. 70 –75.

[37] IEEE. IEEE P1901 Project Authorization Request. http://grouper.ieee.org/groups/1901/1901_PAR.pdf.

[38] IEEE. IEEE P1901 Working Group. Referred 31.8.2010. Web page. http://grouper.ieee.org/groups/1901/.

[39] Intel Corporation. Intel at the 2010 International Consumer Electronics Show. Web Document. Accessed on 31 August 2010., 2009. http://download.intel.com/pressroom/kits/events/ces2010/pdfs/CES2010_FactSheet.pdf.

[40] Intel Corporation. Intel News Release: Intel Unveils 45nm System-on-Chip for Internet TV. Web Document. Accessed on 31 August 2010., 2009. http://www.intel.com/pressroom/archive/releases/2009/20090924comp_b.htm.

[41] Internet Gateway Device (IGD) V 1.0. Web Document. Accessed on 31 August 2010. http://upnp.org/specs/gw/igd/.

[42] Jura Impressa F90. Web Document. Accessed on 31 August 2010. http://www.jura.com/home_x/products_home_use/f_line/impressa_f90.htm.

[43] Keith, R. G., Edwards, W. K., Newman, M. W., and Ducheneaut, N. The work to make a home network work. In *Proc. ECSCW* (2005), pp. 469–488.

[44] KNX Association. Web Document. Referred 31.8.2010. http://www.knx.org/.

[45] KVM: Kernel Based Virtual Machine. Web Document. Referred 31.8.2010. http://www.linux-kvm.org/.

[46] Laoutaris, N., Rodriguez, P., and Massoulie, L. Echos: edge capacity hosting overlays of nano data centers. *SIGCOMM Comput. Commun. Rev. 38* (January 2008), 51–54.

[47] Li, L., Hu, X., Huang, J., and He, K. Research on the architecture of automatic meter reading in next generation network. In *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on* (13-16 2008), pp. 92 –97.

[48] Li, Z., Huang, D., Liu, Z., and Huang, J. Research of peer-to-peer network architecture. In *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on* (apr. 2003), vol. 1, pp. 312 – 315 vol.1.

[49] LibVirt: The virtualization API. Web Document. Accessed on 31 August 2010. http://libvirt.org/.

[50] LonMark International. Web Document. Referred 31.8.2010. http://www.lonmark.org/.

[51] Lookabaugh, T. MoCA Home Networking Makes Sense for Cable Operators Everywhere, 2010. http://www.mocalliance.org/industry/presentations/MoCA_presentation_at_ANGA.pdf.

[52] Lupton, W., Blackford, J., Digdon, M., and Spets, T. Tr-069 cpe wan management protocol v1.1. Tech. rep., Broadband Forum, 2007.

[53] Lv, Q., Cao, P., Cohen, E., Li, K., and Shenker, S. Search and replication in unstructured peer-to-peer networks. In *ICS '02: Proceedings of the 16th international conference on Supercomputing* (New York, NY, USA, 2002), ACM, pp. 84–95.

[54] Metcalfe, R. M., and Boggs, D. R. Ethernet: distributed packet switching for local computer networks. *Commun. ACM 19*, 7 (1976), 395–404.

[55] Multimedia over Coax Alliance. Referred 31.8.2010. Web page. http://www.mocalliance.org/.

[56] Nanodatacenters Project. Web Document. Accessed on 31 August 2010. http://www.nanodatacenters.eu/.

[57] no. 223850, P. Nanodatacenters, combined deliverable: D1.1 (system design and decomposition) and d3.1 (draft architecture specification of security, privacy, and incentive mechanisms). Tech. rep., European Comission Seventh Framework Programme, april 2009.

[58] OASIS. Devices Profile for Web Services Version 1.1. Web Document. Referred 21.11.2010. http://docs.oasis-open.org/ws-dd/dpws/wsdd-dpws-1.1-spec.html.

[59] OECD. Web Document. Referred 31.8.2010. http://www.oecd.org/home/0,2987,en_2649_201185_1_1_1_1_1,00.html.

[60] Oksman, V., and Galli, S. G.hn: The new itu-t home networking standard. *Communications Magazine, IEEE 47*, 10 (oct. 2009), 138 –145.

[61] Open Building Information Xchange. Web Document. Accessed on 31 August 2010. http://www.obix.org/.

[62] OpenBox4 - Bringing Open Source to the Neuf Box 4. Description du hardware de la Neuf Box 4. Web Document. Accessed on 31 August 2010., 2010. http://www.neufbox4.org/wiki/index.php?title=Description_du_hardware_de_la_Neuf_Box_4.

[63] OpenVPN Technologies, Inc. Web Document. Referred 31.8.2010. http://openvpn.net/.

[64] OpenVZ. Web Document. Accessed on 31 August 2010. http://wiki.openvz.org/.

[65] OpenWrt, a Linux distribution for embedded devices. Web Document. Accessed on 30 August 2010. http://openwrt.org.

[66] OSGI. The Role of OSGi Technology in the Home Gateway Initiative (HGI) and End to End Connectivity and Service Provisioning. http://www.osgi.org/wiki/uploads/Congress2005/1014_0930_pastorino.pdf.

[67] Philips. Philips innovation facilitates development of wireless home healthcare devices. Web Document. Accessed on 31 August 2010. http://www.apptech.philips.com/newscenter/index.php?fID=112.

[68] Pouwelse, J., Garbacki, P., Epema, D., and Sips, H. The bittorrent p2p file-sharing system: Measurements and analysis. In *Peer-to-Peer Systems IV*, M. Castro and R. van Renesse, Eds., vol. 3640 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2005, pp. 205–216. $10.1007/11558989_19$.

[69] Project Proposal: Future Internet Gateway-based Architecture of Residential Networks (FIGARO), 2009. http://cordis.europa.eu/fp7/.

[70] Python FTP server library. Web Document. Referred 31.8.2010. http://code.google.com/p/pyftpdlib/.

[71] Python Software Foundation. Web Document. Accessed on 31 August 2010. http://www.python.org/.

[72] Rowstron, A., and Druschel, P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, 2001.

[73] Royon, Y., Frénot, S., and Le Mouël, F. Virtualization of service gateways in multi-provider environments. In *Component-Based Software Engineering*, I. Gorton, G. Heineman, I. Crnkovic, H. Schmidt, J. Stafford, C. Szyperski, and K. Wallnau, Eds., vol. 4063 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2006, pp. 385–392. $10.1007/11783565_31$.

[74] Royon, Y., Frénot, S., and Mouël, F. L. Virtualization of service gateways in multi-provider environments. In *Component-Based Software Engineering* (2006), vol. 4063 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 385–392.

[75] Salz, R. ZSI: The Zolera Soap Infrastructure. Web Document. Referred 31.8.2010., 2005. http://pywebsvcs.sourceforge.net/zsi.html.

[76] Seventh Framework Programme (FP7). Web Document. Accessed on 30 August 2010. http://cordis.europa.eu/fp7/.

[77] Siever, E., Figgins, S., Love, R., and Robbins, A. *Linux in a Nutshell*, vol. 6. O'Reilly Media, 2009.

[78] Skype Limited. UPnP in Skype 3.8 for Windows. Referred 31.8.2010. Web page. http://blogs.skype.com/garage/2008/04/upnp_in_skype_38_for_windows_b.html.

[79] Sosinsky, B. *Networking Bible*, vol. 1. Wiley, 2009.

[80] Sun. Jini. Web Document. Referred 21.11.2010. http://www.jini.org/wiki/Main_Page.

[81] Tanenbaum, A. *Computer Networks*, vol. 4. Prentice Hall Professional Technical Reference, 2002.

[82] The Broadband Forum. Referred 31.8.2010. Web page. http://www.broadband-forum.org/.

[83] The Digital TV Consultancy. Freeplug: Freebox HD, the first box with power line communication (PLC) technology at no additional cost. Web Document. Accessed on 31 August 2010., 2008. http://www.digitaltvnews.net/content/?p=2345.

[84] The Django Framework. Web Document. Referred 31.8.2010. http://www.djangoproject.com/.

[85] The MadWiFi Project. Web Document. Accessed on 31 August 2010. http://madwifi-project.org/.

[86] The OSGi Alliance. Web Document. Accessed on 31 August 2010. http://www.osgi.org/.

[87] Thomas Karagiannis, Elias Athanasopoulos, C. G. P. K. Homemaestro: Order from chaos in home networks. Tech. rep., Microsoft Corp., may 2008.

[88] Thomson Achieves Industry First DLNA Certification for Residential Voice over IP Gateway. Web Document. Accessed on 31 August 2010. http://www.dlna.org/news/pr/view?item_key=d2aabc78ed960921e08f34cac19a0a6a2470a396.

[89] Turboult, R. Djmount. Web Document. Referred 31.8.2010. http://djmount.sourceforge.net/.

[90] UCLA. Web Document. Referred 31.8.2010. http://read.cs.ucla.edu/click/.

[91] UK, I. Broadband Forum and HomeGrid Forum to collaborate on G.hn. Referred 31.8.2010. Web page., 2010. http://www.iptv-news.com/iptv_news/june_2010_2/broadband_forum_and_homegrid_forum_to_collaborate_on_g.hn.

[92] Univers Freebox, Olivier Viaggi. Une Freebox V6 intégrant un Atom CE 4100 ? Pourquoi pas ... Web Document. Accessed on 31 August 2010., 2009. http://www.universfreebox.com/article11470.html.

[93] UPnP Forum. Web Document. Accessed on 31 August 2010. http://www.upnp.org/.

[94] UPnP Forum. UPnP Device Architecture 1.0.

[95] VideoLAN. Web Document. Referred 31.8.2010. http://www.videolan.org/vlc/.

[96] Vinton Gray Cerf. Web Document. Referred 31.8.2010. http://www.brainyquote.com/quotes/quotes/v/vintoncerf404539.html.

[97] Wei, Z., Li, J., Yang, Y., and Jia, D. A residential gateway architecture based on cloud computing. In *Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on* (2010), pp. 245 –248.

[98] XBMC. Web Document. Referred 31.8.2010. http://xbmc.org/.

[99] Xia, H., and Brustoloni, J. Detecting and blocking unauthorized access in wi-fi networks. In *NETWORKING 2004, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications* (2004), vol. 3042 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 795–806.

[100] Xiao, Y. Ieee 802.11n: enhancements for higher throughput in wireless lans. *Wireless Communications, IEEE 12*, 6 (dec. 2005), 82 – 91.

[101] Yahoo. Flickr. Web Document. Referred 31.8.2010. http://www.flickr.com/.

[102] Yang, Y., Wei, Z., Jia, D., Cong, Y., and Shan, R. A cloud architecture based on smart home. In *Education Technology and Computer Science (ETCS), 2010 Second International Workshop on* (2010), vol. 2, pp. 440 –443.

[103] Z-Wave Alliance. Referred 31.8.2010. Web page. http://www.z-wavealliance.org/.

[104] Zeeb, E., Bobek, A., Bonn, H., and Golatowski, F. Lessons learned from implementing the devices profile for web services. In *Digital EcoSystems and Technologies Conference, 2007. DEST '07. Inaugural IEEE-IES* (2007), pp. 229 –232.

[105] ZVEI, Division Luminaires. Digital Addressable Lighting Interface Activity Group Overview. Web Document. Referred 31.8.2010., 2001. http://www.dali-ag.org/c/manual_gb.pdf.

A Screenshots

A.1 Physical Gateway Controller

Figure 28: The authentication interface

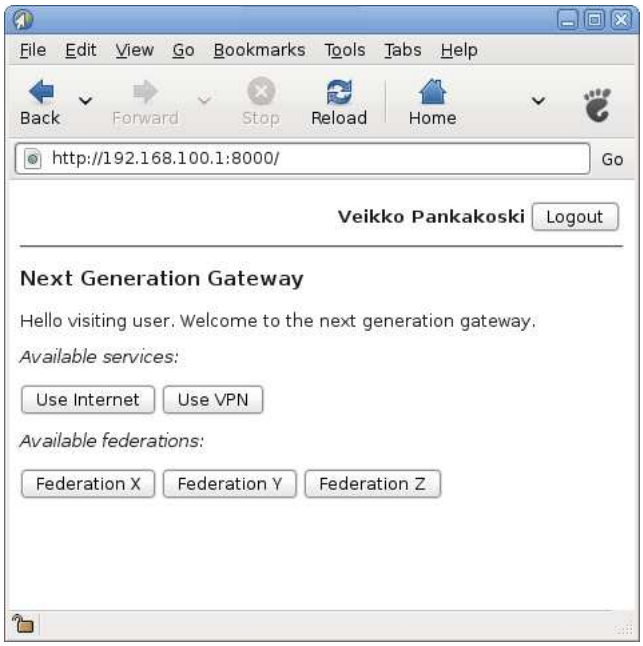

Figure 29: The physical gateway interface

A.2 Virtual Gateway Controller

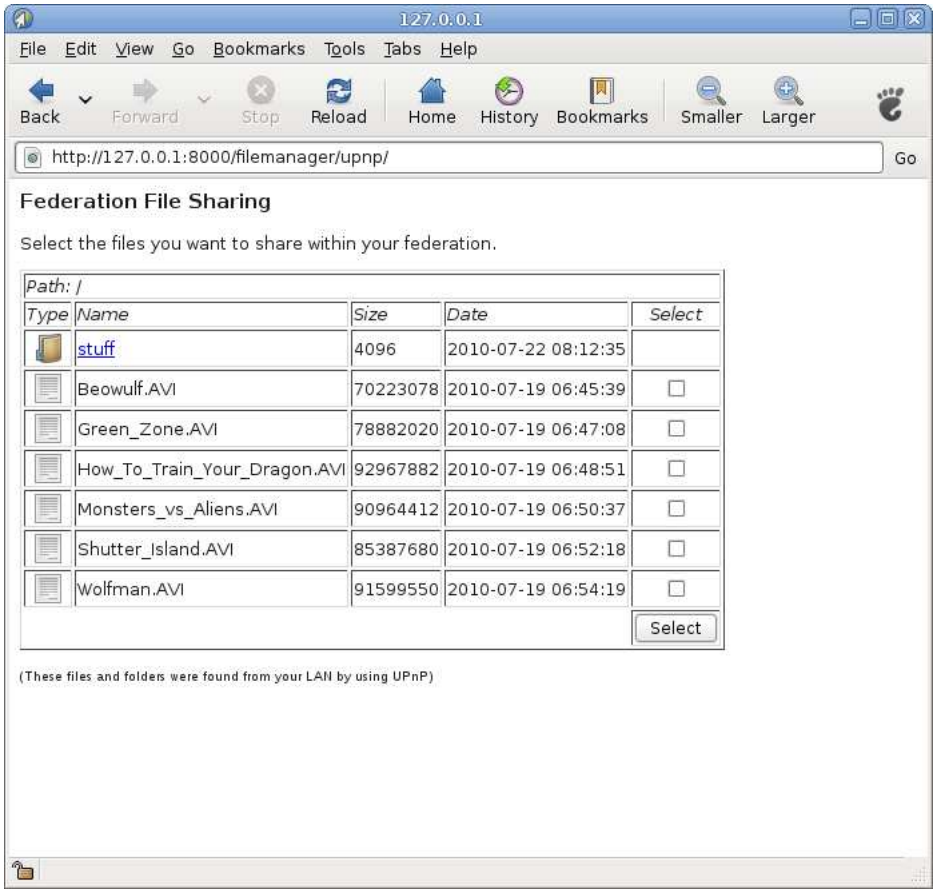

Figure 30: The web interface for the federation file system

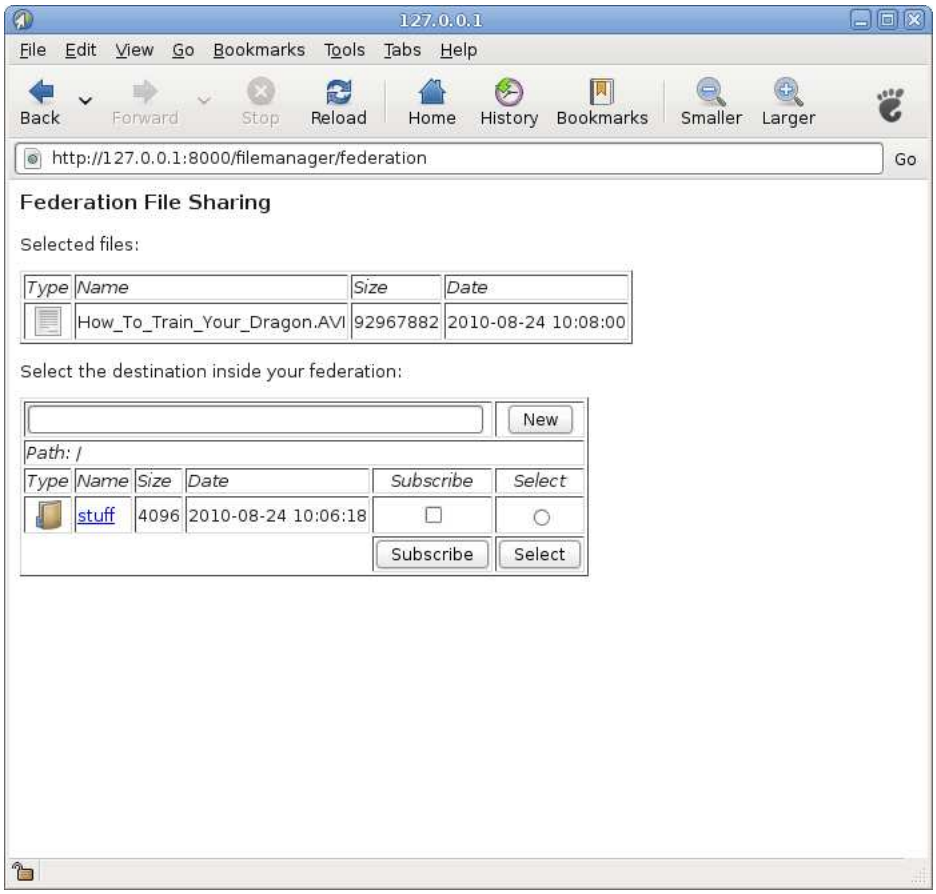

Figure 31: The web interface for the federation file system