A GATEWAY SOLUTION FOR ACCESSING NETWORKED APPLIANCES

ARSHAD MUHAMMAD M.SC., B.SC.

PH.D. THESIS 2009

A GATEWAY SOLUTION FOR ACCESSING NETWORKED APPLIANCES

By

Arshad Muhammad M.Sc., B.Sc.

A thesis submitted in partial fulfilment of the requirements of the Liverpool John Moores University for the degree of Doctor of Philosophy

Supervised by

Prof. Madjid Merabti and Dr. Bob Askwith

Networked Appliances Laboratory
School of Computing and Mathematical Sciences
Liverpool John Moores University

May 2009

AT THE REQUEST OF THE UNIVERSITY,THE FOLLOWING FIGURES AND TABLES HAVE NOT BEEN INCLUDED IN THIS DIGITAL COPY;

This thesis is dedicated to my
Uncle Arbab Muhammad Ayub Khan
(1929-2005) & my parents

ACKNOWLEDGEMENTS

I am really thankful to Allah the most gracious and most merciful. I would like to express my deep felt gratitude to both my supervisors Prof. Madjid Merabti and Dr. Robert J. Askwith for their patience and guidance. I would also take this opportunity to thank my late uncle, parents, and my school and college teachers without whose guidance and support I would not have reached this stage of my life. I am also very thankful to my brothers Arbab Shahid Mehmood, Arbab Muhammad Touseef and sisters for their continuous love and support through a difficult and trying period of my life. I am also very thankful to my cousin Arbab Muhammad Tariq for his continuous support throughout my career. I would also like to thanks my Brothers-in law Syed Tofique Shah and Ibrar khan. I would also like to thanks my nephew Ezaz Shah and nieces Laviza Shah, Amara Shah, Alina Shah and Zobia Abrar for my making me laugh when I needed it. I also want to acknowledge my office mate Dr. David Llewellyn-Jones for his encouragement, discussions and support throughout my PhD. I want to thank Dr. Paul Fergus for those long discussions which helped me a lot in my difficult moments in my studies. I would also take this opportunity to thank my fellow student Amjad Shaheed for his support throughout my stay in Liverpool. I also want to acknowledge the cooperation of my house mates Farooq Alam, Khanzada Naveed Ullah and Kamal Orakzai. I also want to acknowledge the cooperation of Dr. Huma Javed, Jasim Saeed, Dr. Humayun Bakht, Dr. Gurleen Arora, Dr Fayçal Bouhafs, Dr. Henry Chang, Muhammad Zahid Khan, Sohail Abbas, Haseeb Ur Rahman, Prof. Azzelarabe Taleb-Bendiab, and Dr. Kashif Kifayat. I would like to extend a special thanks to the academic, administrative, my fellow researchers and technical staff of School of Computing and Mathematical Sciences .I am really grateful to all these people mentioned above in particular and all others that I have not mentioned but have helped me a lot with their sincere and continuous support throughout the course of my PhD.

I want to dedicate this thesis to my late uncle Arbab Muhammad Ayub Khan (May his soul rest in peace) and parents who has been a source of inspiration, selfless love and devotion all my life.

ABSTRACT

Today modern technology has reached the point whereby electronic devices are commonplace in every facet of our life. We encounter numerous electronic devices that surround us within the home and office environment, including devices in shopping centres and other public spaces. All these devices are an essential part of our daily life, and we use them to perform numerous functions, for example, to remotely control and monitor home utility services, to integrate and share content with friends and family, and access online services such as banking and shopping.

Now imagine if all these electronic devices could be seamlessly integrated to enable intercommunication between the functions they provide. For example, if you left your MP3 player at work on a Friday night and the following day you wished to listen to the latest Rock album stored on it – while at home you could discover the MP3 player and play the audio stream on your Hi-Fi without knowing where the device is located.

In order to achieve this seamless interaction we need an ad hoc gateway service to combine the devices we own into a personalised configuration enabling any device, irrespective of where it is located, to be discovered and used. Typically, if devices want to access the services offered by other devices they must go through a centralised gateway. If this gateway fails then all the devices using that gateway and the services provided by it will become unavailable within the network. However a better solution would be to enable the devices to dynamically discover an alternative gateway and reconfigure themselves into the user's personalised configuration.

In this thesis we propose a novel approach to addressing these challenges, using our Ad Hoc Gateway Service Framework for accessing the services of Networked Appliances using Peer-to-Peer (P2P) network technology. Our framework allows devices to be seamlessly interconnected and operated with little human intervention in distributed Peer-to-Peer network. We explore how P2P technologies can be used to implement an ad hoc gateway service that enables devices at different locations to be combined into a personalised configuration. It provides mechanisms that enable zero configuration, automatic service discovery, service management and device capability matching at a management level. The failure of a gateway in our framework does not result in failure to access all devices/services, instead an alternative gateway is located and connected into the existing configuration. We have successfully developed a real world prototype that implements an Estate Agent Framework, which is used to evaluate our framework.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND TERMS

ADSL	Asymmetric Digital Subscriber Line
ARPA	Advanced Research Projects Agency
ARPANET	Advanced Research Project Agency Network
CAN	Content Addressable Network
DHCP	Dynamic Host Configuration Protocol
DHT	Distributed Hash Table
DLNA	Digital Living Network Alliance
DNS	Domain Name Service
DPWS	Devices Profile for Web Services
DSL	Digital Subscriber Line
ebXML	Electronic Business eXtensible Markup Language
ENIAC	Electrical Numerical Integrator and Calculator
FTP	File Transfer Protocol
GENA	General Event Notification Architecture
GPS	Global Positioning System
GSM	Global System for Mobile communications
HAVi	Home Audio-Video Interoperability
HES	Home Electronics System
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
IEEE	Institute of Electrical and Electronics Engineers
IPTV	Internet Protocol Television
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider
LAN	Local Area Network
MANET	Mobile Ad hoc NETwork
MMOG	Massively Multiplayer Online Gaming
MODEM	Modulator-DEModulator
NA	Networked Appliances
NAT	Network Address Translation
OASIS	Organization for the Advancement of Structured Information Standards
OSGi	Open Services Gateway Initiative
OSI	Open Systems Interconnection
P2P	Peer-to-Peer
PAN	Personal Area Network
PC	Personal Computer
PCI	Peripheral Component Interface
PCMCIA	Personal Computer Memory Card International Ass.
PDA	Personal Digital Assistant
POTS	Plain Old Telephone System
PPP	Point-to-Point Protocol
PS3	Play Station 3
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RFID	Radio-Frequency Identification

RIAA	Recording Industry Association of America
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TTL	Time to Live
UDDI	Universal Description Discovery and Integration
UDP	User Datagram Protocol
UML	Unified Modelling Language
UOPF	Ubiquitous Open Platform Forum
UPnP	Universal Plug n Play
URL	Uniform Resource Locator
USB	Universal Serial Bus
WAN	Wide Area Network
WSN	Wireless Sensor Network
WWW	World-Wide Web
XML	eXtensible Markup Language (XML)

CHAPTER 1

1 INTRODUCTION

1.1 Preamble

Modern technology has reached the point whereby electronic devices are commonplace in every facet of our life. We encounter numerous electronic devices that surround us within the home and office environment, including devices in shopping centres and other public spaces. All these devices are an essential part of our daily life, and we use them to perform numerous functions, for example, the DVD player in your home is used to play your favourite movies; the TV is used to watch your favourite programs, and your personal PC is used to perform a range of tasks including online banking and emailing friends, colleagues and family members. Imagine if all these electronic devices could be seamlessly integrated to enable intercommunication between the functions they provide. In this context, we would call such devices "Networked Appliances" (NA) [Merabti 2008; Moyer 2002].

For example, if you left your MP3 player at work on a Friday night and the following day you wished to listen to the latest Rock album stored on it – while at home you could discover the MP3 player and play the audio stream on your Hi-Fi without knowing where the device is located. In order to achieve this seamless interaction we need a middleware to combine the devices we own into a personalised configuration enabling any device, irrespective of where it is located, to be discovered and used.

A number of industries have tried to create internetworking solutions such as Universal Plug n Play (UPnP) [Kim 2006; UPnP July 2006] and the Open Services Gateway Initiative (OSGi) [Marples 2001; Zebin 2007]. Moreover some important research efforts have been developed to discover services offered by NAs within the home environment [Bhatti 2002; Evans 2001; Minoh 2001]. However these solutions do not provide any means for the discovery of the services outside the dedicated

network configuration. This problem has been overcome, in part, using Peer-to-Peer (P2P) technologies whereby digital content can be distributed and discovered using global communications [Li 2002; Yeager 2002].

Typically, if devices want to access the services offered by other devices they must go through a centralised gateway. If this gateway fails then all the devices using that gateway and the services they provide will become unavailable within the network. However a better solution would be to enable the devices to dynamically discover an alternative gateway and reconfigure themselves into the user's personalised configuration.

In this introduction chapter we provide an overview of our research area, which involves Networked Appliances and middleware for home networking. This chapter then details our proposed novel framework that addresses these limitations to enable a device to advertise its services, discover other devices, integrate these services together, provides middleware such as a gateway service and provides an alternative service in case of failure of one service within composition.

1.2 Project aims and objectives

With the introduction of home networking technology, a framework is required to interconnect devices using P2P networks to share services. This framework provides operations such as service recovery, service registration, service sharing and provides gateway services to make communication possible not only within the same network but with other remote networks as well. The framework enables the discovery of alternative gateway services in case of failure without losing any communication or requiring user intervention.

To achieve this, it is necessary to fulfil the following objectives:

1. Understand current P2P technologies and their functionalities.
2. Review compositions of NAs in P2P networks via studying current techniques in this domain.
3. Define the requirements of an Ad Hoc Gateway Service for this research in relations to P2P networks.
4. Develop a solution to ensure availability of gateway services to seamlessly interconnect devices regardless of their location.

5. Develop a solution to ensure secure access to the services available in the P2P network.
6. Develop a solution to ensure allocation of best services available in the P2P network.
7. Develop a solution to ensure availability of an alternative gateway service in case of failure.
8. Develop a working prototype of our proposed framework to demonstrate how we implement different components of the proposed framework.
9. Evaluate and compare our framework against existing solutions.

Hence to achieve these objectives, the understanding of the subject area includes P2P technologies, P2P networking, service advertising, service discovery, content sharing, security techniques and Service-Oriented Architectures (SOA). The focus of this research will be service discovery and composition in P2P networks (described in chapter 2). In existing solutions, where devices connect together via black box devices and discover other services in the network all nodes rely on a single black box device, but as part of this research the gateway will only exist when a user requests it, i.e. it is Ad Hoc.

1.3 Novel aspects of Proposed Framework

The contributions to knowledge through this research are:

- This work presents the design and prototype of ad hoc gateway service (AdHocGS) framework that ensures the availability of gateway services in a distributed P2P network. When a device first connects to the P2P network it discovers a gateway service by discovering the gateway advertisement. Any device in the network that offers a gateway service may respond to the gateway service request, allowing the device access to the gateway, which enables it to discover the available services within the P2P network. This result does not only ease the restrictions associated with the use of special hardware, but also those found in centralised operations, i.e. single point of failure [Muhammad 2005].

- The gateway service itself may be composed of individual services that may either reside locally or remotely within the P2P network, that allow the gateway to perform security management, Quality of Service analysis and device capability matching. If the gateway service fails then all the services it offers fail as well. However, if one or more of the core services used within the gateway service fails then only the failed services will be lost and as a result alternative services can be discovered. However if one or more of the core service used within the gateway service fail then only the failed service will be lost and as a result an alternative service will need to be discovered. Our published paper [Muhammad 2007] demonstrates a working prototype of these components.

- We created an overlay network of gateways where all gateways can communicate with each other by creating a P2P network. Using this overlay network, devices or services available in one network can communicate with other remote devices or services. Using a gateway a numbers of peers can be connected and if the requested service does not exist locally, the gateway can request these services from the overlay network with other gateways. Using this technique, it not only enables us to create a personalised gateway by connecting our home or office devices and accessing them via the Internet but can also enable specialised gateways only offering a specific set of services e.g. video or audio services. By using a specialised gateway the user can ask and pay for using particular services i.e. if a particular peer wants to use a service they may need to pay.

- The proposed framework offers services such as Service-Oriented Networking, Service Advertisement and Discovery, Service Registration, Ad Hoc Gateway Service and Secure Access to the services. Framework components are not fully dependent on each other, allowing flexibility and components can be extended to add further functionality to the framework. For example, Security Manager can be developed according to user requirements.

1.4 Project Achievements

Our Ad hoc Gateway service framework proposes a solution for gateway services within a P2P network. Other research did not address this issue before. We have not only successfully implemented our framework but we have also demonstrated this using our prototype discussed in the implementation section, which could be easily extendable. Our research has produced the following conference papers.

- A. Muhammad, M. Merabti, and B. Askwith, "An Ad Hoc Gateway Service for Discovering and Composing Networked Appliances," In Proceedings of Sixth Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet 2005), Liverpool John Moores University, UK, 27-28 June 2005, pp. 377-382
- A. Muhammad, M. Merabti, B. Askwith and P. Fergus, "Ad Hoc Gateway Service for Automatic Package Delivery using Networked Appliances," IEEE Wireless Communications and Networking Conference, Hong Kong, 11-15 March 2007, pp. 2576-2581.
- A. Muhammad, M. Merabti, and B. Askwith, "An Ad Hoc Gateway Service for Flexible Access to Networked Appliances," In Proceedings of The 8th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, Liverpool John Moores University, Liverpool, UK, 28th-29th June 2007

1.5 Summary and Thesis Structure

In this chapter we highlight some of the related issues important to our approach. This chapter defines the scope of our research project, our novel contributions and thesis structure.

In chapter 2 we present background work. We start with a brief history of computer networks, from the early history of the Internet. We then move to the network architecture such as different network topologies and wired/wireless networking. We then move to Ubiquitous/Pervasive Computing, explain it and then discuss some work done and challenges in this field. We also discuss Networked Appliances in relation to home networking. Some notable research work done in

seamlessly interconnecting Networked Appliances within home networks is reviewed. The chapter also discusses P2P networks, their merits and limitations and some challenges in this field. We discuss some P2P models and how integration is being performed using P2P techniques. We also discuss some well known P2P applications such as Napster, Gnutella, and KaZaA. Each P2P model is discussed in terms of its functionality, limitations, structure, discovery and failure of a particular service or device. We conclude with a discussion of security issues in P2P networks, the importance of security in networks and how it is possible to achieve it.

Chapter 3, related work, is a continuation of chapter 2 where we mainly discuss about Service-Oriented Architecture middleware used to seamlessly interconnect devices within home networks. We discuss some well known SOA architecture middleware such as OSGi, UPnP, and DPWS. We discuss these middleware in terms of their architecture, functionalities and address their limitations. We found in these middleware solutions that discovery services are very limited as they are based on proprietary descriptions of how services must be advertised and discovered.

Chapter 4 presents our novel Ad Hoc Gateway Service (AdHocGS) framework. The chapter outlines a number of challenges that need to be addressed with service discovery and distribution in P2P environments and their impact on this research. In the beginning we determine the requirements of the proposed framework resulting from the analysis in chapters 2 and 3. We conclude from our background chapters the limitations within current middleware solutions not only require ad hoc gateways which enable services to be advertised and discovered within global networks but also provide an alternative gateway service in case of failure. This chapter discusses the requirements for a system that allows Networked Appliances to be advertised and discovered in a P2P environment. This chapter also includes the concepts and models developed to fulfil the requirements and address issues which are raised. We also conclude that in current middleware solutions the failure of a particular service in composition, results in the failure of the whole composition. So we also focus to provide an alternative service in case of failure of any service. We present two scenarios to explain our idea, which we later implement in chapter 6. We discuss how to communicate with NAs with and without middleware. This chapter also discusses some design challenges for the proposed framework; we conclude the chapter by

providing an overview of the main components of our design based on the project requirements.

In chapter 5 we discuss the main components of our framework in more detail and explain how communication is achieved between them with the help of UML diagrams. Using our design we explain the novelty of our framework and how the various design issues have been addressed and how these impact on the overall design processes.

Chapter 6 presents our implementation of the proposed framework. In chapter 4, which is our design chapter, we discussed the components of our AdHocGS framework. In this chapter, we discuss how we implement these components to achieve our objectives. In this chapter the presented case study shows how we implement our framework. This chapter also includes the testing of our framework. We present our prototype and show how it is capable of discovering gateway services within P2P networks and rediscovering alternative gateways when failure occurs. In this chapter we talk about the tools used in designing our prototype.

Chapter 7 demonstrates the application of our framework to the Estate Agent case study and its evaluation. In this chapter each component of the system, described in chapter 4, is evaluated against related work. This chapter also discusses other application areas where our framework may be utilised. We discuss a number of cases which help us to identify limitations and short comings of our implementation.

Finally, Chapter 8 presents the concluding remarks of this research thesis and summarises the finding of this thesis. This chapter concludes our PhD project by providing an overall summary, contribution to knowledge and future plans. Further it lists future work for framework enhancement, followed by the Appendices which include detailed design notes.

CHAPTER 2

2 BACKGROUND

Despite many years of invention, computers were initially designed to work alone, in the form of mainframe computing. In the 1960s sets of computers were connected together to interchange information and allow remote access to computer resources. This change was the start of a new era of computing and networking – known as Internet more recently. In this chapter, we introduce the history of computer including networks, TCP/IP which enables various computers and networks to communicate with each other. Also we introduce the idea given by Mark Weiser [Weiser 2002] of Ubiquitous Computing. This chapter provides an overview of the work carried out in the relevant research related to this thesis, which includes Ubiquitous/Pervasive Computing, Networked Appliances, Peer-to-Peer networking, Quality of Service and security.

2.1 Brief history of computer networks

The history of the Internet dates back to the early development of communication networks. The purpose of the computer network was to allow information exchange among users of various computers. Internet is the worldwide accessible network of interconnected computer networks used to send and receive data via the Internet Protocol (IP) [Tanenbaum 2003]. It consists of thousands of smaller networks such as academic, business and government, which exchange information such as e-mail, chat, file transfer etc.

A computer network is used to interconnect different computers or devices by using transmission technology [Tanenbaum 2003] in order to exchange information. In the early days, computers were huge and took up the space of a whole room. At that time, computers were usually used only in research labs. On June 26, 1946, John Mauchly and J Presper Eckert developed the first electronic general purpose computer

called ENIAC I (Electrical Numerical Integrator And Calculator) [Wilkes 2006]. This computer's abilities were limited and it took a long time to process a program.

In 1960s, ARPA (Advanced Research Projects Agency) initiated a project, with the main objective to connect researchers' computers [Salus 1995] and enable researchers to remotely access computer resources. Prior to the introduction of the Internet, most networks were limited to only allow communications between the stations on the network. This usually meant connecting to the central mainframe computer allowing its connecting stations to store, retrieve and exchange information via directly connected network links.

The Advanced Research Project Agency Network (**ARPANET**) [Chandra 2007; Hauben 2001] developed by ARPA of the US Department of Defence was the first operational distributed network, the first step towards Internet. The first ARPANET link was established between the University of California and Stanford Research Institute in 1969 and became the prototype Internet and brought a change from centralised to distributed computing. Before ARPANET, many hardware and software technologies were used for networking such as wired, wireless etc. The development of TCP/IP [Cameroon 2006; Tanenbaum 2003] in 1970's, connects two different technologies using routers [Casad 2008; Comer 2005]. In 1983 ARPANET switched over to TCP/IP, an important next step towards Internet. Due to growth of Internet many big companies like Cisco [Cisco 2006], International Business Machines Corporation (IBM) [IBM 2009] and Microsoft [Microsoft 2009] introduced advancements in networking hardware, systems and software. Today, the Internet has become an important part of our daily lives. We can do most of our work over the Internet like shopping, banking, distance learning and socialising. Using social networks such as MySpace and Facebook, we can make friends around the world.

2.2 Network Architecture

Networks can be categorised by network protocol layer, e.g. application layer or by scale, e.g. Personal Area Network (PAN), Local Area Network (LAN), Metropolitan Area Network (MAN), Wide Area Network (WAN) or by connection method, for example, HomePNA, Ethernet, Wi-Fi or by functional relationship

Client-Server, Workgroup, Peer-to-Peer or by topology Bus, Star, Ring, Mesh, Tree, Star-bus, or by physical connection, Wired or Wireless.

Wired: when we use physical media such as copper cable, the media used depends on how much information needs to be transmitted at a specific time. Many techniques such as Modems, Integrated Services Digital Network (ISDN), Digital Subscriber Line (DSL), Ethernet, RS-232, RS-485 and Optical fibre are available.

Wireless: is a more recent way of networking and rather than using cables allows users mobility, i.e. to move around in different locations. In wireless different techniques are used depending upon the range. For short range Bluetooth and Infrared (IrDA), medium range Wi-Fi (IEEE 802.11), WiMax (IEEE 802.16) and for long range Satellite and Mobile phones such as Global System for Mobile communications (GSM) [Garg 2007].

Wireless networks use radio waves to transmit data between computers or devices [Garg 2007]. The GSM (Global System for Mobile Communication) network is divided into three systems: Switching System, the Base Station, and the Operation and Support System [Garg 2007]. The cell phones connect to the Base Station which connects to the Operation and Support Station then to the Switching Station where the call is transferred similarly to the destination phone. Wi-Fi is one other example which enables connection to the Internet or other devices that have wireless functionalities; it broadcasts radio waves that can be intercepted by the Wi-Fi receivers attached to the different devices. Wi-Fi is commonly used to extend existing Ethernet LANs.

The OSI Reference Model: Open Systems Interconnection (OSI) model was originally developed to provide a framework for building networking protocols on. The OSI model consist of seven distinct layers, each contains a separate abstraction of networking. Figure 2.1 illustrates the OSI 7-Layer model. Consider the Application Layer as layer 7 and Physical Layer as layer 1. The *Application layer* encapsulates application data, where applications communicate with each other. Communication is done using the application's own language specified by application protocols, such as Hypertext Transfer Protocol (HTTP) used to send and receive web content. HTTP

passes data from the web server to web browser using HTTP headers. All communication between server and client are performed at the application layer. Other examples are File Transport Protocol (FTP) [Chandra 2007], Domain Name Service (DNS) [Chandra 2007] and Telnet. The *Presentation layer* controls the presentation of data contents. The main role of this layer in the OSI model is to ensure the presentation of data is handled correctly between applications. The *Session layer* controls session between two systems, which is important to many communications for networking. The *Transport layer* provides control communication between hosts. Two types of Internet transport service are commonly available first connection-oriented Transmission Control Protocol (TCP), works as a transport layer for reliable delivery of data between computers [Cameroon 2006; Tanenbaum 2003] and connectionless User Datagram Protocol (UDP) [Garg 2007]. The *Network layer* is responsible for the transportation of packets between two hosts in the network using Internet Protocol (IP). Network layer determines the path and direction to allow communications between two hosts and this is called routing. The *Data Link layer* determines how to transmit data between stations. It formats data into frames and delivers it using a network interface. Ethernet, Point-to-Point Protocol (PPP) [Cameroon 2006; Walls 2006] Frame Relay [Chandra 2007] function at the data link layer. The last layer *Physical layer* connects hosts physically. This includes network hardware such as Cat 5 cable, Ethernet and wireless and the signal encoding schemes used.

Figure 2.1 : OSI 7-Layer Model [Tanenbaum 2003]

Ad Hoc Network: or MANET (Mobile Ad hoc NETwork) [DROPS 2007] is a network connection more associated with wireless devices as shown in Figure 2.2. Ad hoc network is a new approach of wireless networking for mobile users; it does not rely on fixed infrastructure. Mobile devices within each other's radio range can communicate via wireless links; devices outside radio range use other devices for transmission such as routers. Connections are established for the duration of one session and devices discover others within range to form a network. Devices can search for another device by broadcasting messages they receive and forwarding via each node. Ad hoc networks are mainly used in military fields where units equipped with wireless devices could form an ad hoc network. They are also used for emergency and rescue missions. An ad hoc network has the ability to make communications possible even between two nodes that are not in direct range of each other; information can be exchanged between devices via intermediate nodes as shown in Figure 2.2 where node A can communicate with node D via nodes B and C and vice versa.

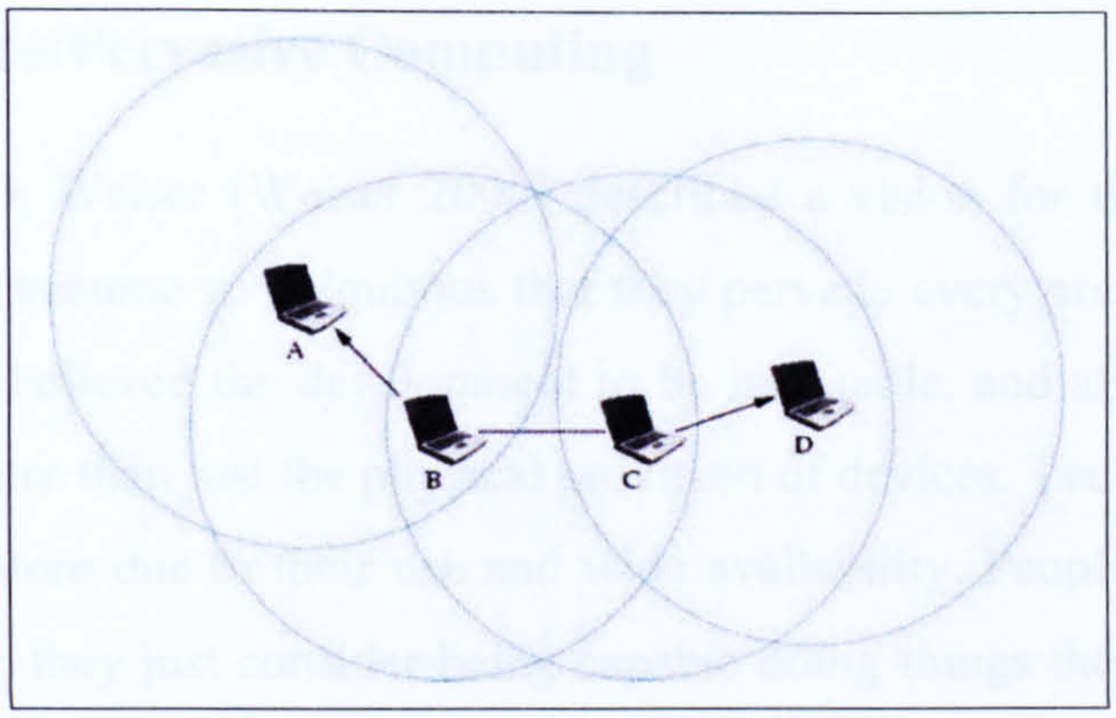

Figure 2.2: An Ad Hoc Network

Wireless sensor network: A Wireless Sensor Network (WSN) [Garg 2007] contains large numbers of tiny sensor nodes deployed in a geographical area to sense or monitor physical conditions such as temperature or sound. Sensors are not only used to monitor movement of an enemy in a battlefield but also in healthcare, home automation etc. Figure 2.3 illustrates a typical WSN, in which sensors are deployed to transmit data to each other which are then collected by the main gateway sensor node to transmit to the user.

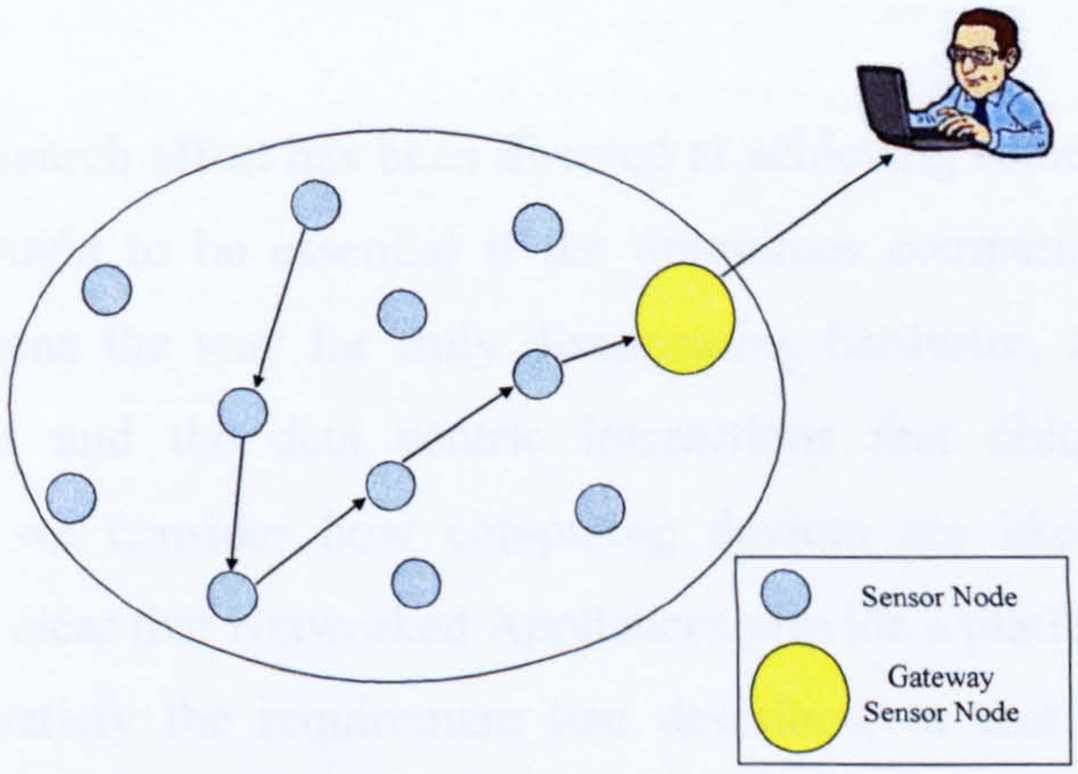

Figure 2.3: Wireless Sensor Network

Each tiny node is equipped with an onboard processor to process raw data by locally carrying out some computations before transmitting the required data. The deployment of sensors may not be predetermined and random deployment means that sensor network protocols and algorithms must have self-organising capabilities. Connections between sensors are ad hoc in nature and require wireless ad hoc networking techniques [Yick 2008].

2.3 Ubiquitous/Pervasive Computing

In 1991 Mark Weiser [Weiser 2002] described a vision for the 21st century in which computers become so ubiquitous that they pervade every area of our lives and environment. He believed the development to be inevitable, and also that the effects would be far greater than just the physical provision of devices. Technologies become human second nature due to their use and wide availability. People stop thinking of using technology; they just consider being capable doing things themselves whatever the technology enables them to do.

If we look around, we see the use of computer technology in many areas, from mobile phones to mainframe computers, and in our home appliances e.g. TV, stereo player. The main idea of Ubiquitous Computing is that computers become embedded in our daily life in such way that we interact with them more naturally and it enables devices to sense changes in their environment and automatically adapt and act based on user needs. One example of such assisting technology includes GPS (Global Positioning System) which allows geographic coordinate location sensing.

Significant research effort has been directed at achieving some of the computing characteristics thought to be essential if the ubiquitous computing vision is to be achieved. This opens the way for truly disappearing hardware, zero-configuration, context awareness and the data centric interactions that ubiquitous computing embodies. When we consider how computing devices are likely to merge with surroundings, it is clear that Networked Appliances provide a platform to build such a future on. They satisfy the requirement just described, in that they can be used naturally without awareness from the user that they are interacting with a computer, and they provide an obvious medium for much of the data, multimedia entertainment and context data, that is currently of importance within the home. However to provide a seamless experience, such devices will need to be robust and operate seamlessly, independent of their geographical location.

2.4 Networked Appliances

Currently there are a range of digital devices and the trend is moving us closer to an increasingly interconnected world. Different household devices are capable of

networked communication. Current advances in technology and high speed broadband are the basis of the development of Networked Appliances for the home. This provides the ability for an Internet connected microwave to download recipes from the Internet, check what is in the refrigerator and place an order in case of any shortage of ingredients [JETRO 2005]. Also using RFID [JETRO 2005] technology, we can gather information from RFID tags in supermarket, to access our refrigerator to check what food we have.

When we talk about Networked Appliances we also need to look at Internet Appliances [Gillett 2001] which are products which access services on the Internet such as WWW or IP telephony. These include intelligent home devices, PDA's and other Internet connected devices, enabling users to operate home devices from a distance. The main idea behind Internet Appliances is to make cheaper devices than PC's dedicated to performing single or a series of functions. Alongside Internet Appliances, the term Information Appliances refers to a device that is used to handle particular information and perform related tasks. Both terminologies Internet Appliances and Information Appliances are often used interchangeably. Common examples of Internet appliances are Internet Tablet, Nokia 770 & N800 [Nokia 2009]. With the rise in home networking, a new range of devices have been introduced such as Vonage Internet phones [Vonage 2006], Internet radio and IPTV (Internet Protocol Television) [Christian 2006]. There is no clear definition of Internet Appliances, e.g. [Gillett 2001] states that Internet appliances are a consumer device that are not PC, but does connect to Internet. The idea behind Internet appliance is to reduce the complexity of the PC for people who want to use the Internet only [Gillett 2001]. On the other hand, Networked Appliances have a network interface and functions well within LAN and do not need Internet access. But Networked Appliances could gain access to the Internet via its network connection. Internet Appliances can only connect to the Internet but cannot interact within the LAN. In our research we are using the NA definition presented above "*a dedicated function consumer device with an embedded processor and a network connection*" [Moyer 2002]. In other words a device that publishes their functionality as services which can be discovered and used by other Networked Appliances, which extend the functionality beyond what they otherwise provide.

2.5 Peer to Peer Networking

The term P2P has been around for a while, but the traditional model of the Internet was Client-Server, where servers host and clients request information.

P2P refers to a network that enables two or more peers to share or exchange information without relying on any central coordination. File-sharing systems are considered as popular P2P, for example, KaZaA [KaZaA 2008] and Napster [Nagaraja 2006], which enable different peers to exchange files such as mp3 songs. Another example of P2P is Instant Messaging. P2P offers advantages in contrast to other architectures such as client/server on the basis of technical and economic criteria, such as performance, persistence, cost [Schoder 2003]. As they operates independently of any central coordination, they enable users to share or exchange information without any boundaries, without worrying about the location of the peer. For example, in the case of KaZaA which is mainly used for music file sharing, users search for a particular song and on the successful search users can download without knowing the location of the host peer.

Table 2.1: P2P limitations and advantages

Limitations	Advantages
➢System complexity ➢Difficult to implement security ➢Cant guarantee QoS ➢Network control ➢Interoperability	➢Sharing ➢High availability ➢Improved performance ➢Scalability ➢Robustness ➢Cost sharing ➢No single point of failure ➢Data persistence

On the other hand P2P introduces some challenges such as network control, security, interoperability and cost sharing [Schoder 2003]. Table 2.1 shows some limitations and advantages of P2P networks.

- **Network Control:** Due to the distributed nature of P2P networks and the absence of a central controller, it is difficult to control the communication among the peers.

- **Security:** To implement security mechanisms in P2P networks is difficult as they frequently access third party systems. In such case the use of firewalls is not enough. An example is instant messenger in which communication mostly happens without any encryption. Organisations interested in P2P must implement methods for authentication, authorisation, availability and trust [Schoder 2003].
- **Interoperability:** in P2P peers may use different protocols. Mechanisms should be implemented to enable peers using different protocols to communicate with each other. Therefore it is important that the communication should be independent of any specific protocol implementation, for example, in the network one device may use TCP/IP while the destination device may use X.25 [Mohan 2004]– mechanisms to support protocol translations must be defined [Abuelma'atti 2002].
- **Cost Sharing:** in P2P sometimes just a few peers use the available resources without sharing their own resources, which violates the spirit of P2P such as information availability, resource sharing and performance. Possible solutions to accountability introduce cost mechanisms that peers have to pay for the information request [Arora 2005]

P2P does not rely on any central controller as in the case of Client-Server network. It relies on the computing power and bandwidth of the other peers or computers in the network. P2P is usually used to connect a large number of devices via ad hoc connections which is mainly for purposes such as content/file sharing. The earliest famous P2P network was Usenet (USEr NETwork) [Fisher 2006] news server system, in which peers communicate with one another propagating articles using the network via emailing other users. However news servers also acted in a Client-Server form when individual users accessed local news servers to access a particular article.

Some networks such as Napster, Limewire use Client-Server for some purposes such as searching while using P2P for others. Networks such as Gnutella or FreeNet [Samsudin 2008] use P2P for all purposes and are referred as true P2P networks. All peers in the P2P provide resources such as bandwidth, storage space, computing power; when peers arrive in the system, resources increase, the total system capacity

also increases. This is not true for Client-Server systems where including more clients could mean slower data transfer for all clients. The distributed nature of the P2P network also increases robustness in the network as data is often replicated on different peers.

Another similar term used for P2P networks is P2P overlay networks [Eng Keong Lua 2005; Li 2007; Lua 2005] . *Overlay networks* are built on top of another network. Nodes in overlay networks act as connected via virtual links i.e. each corresponds to a path in the physical network in the underlying network. The overlay has no control of how packets are routed in the underlying network, P2P networks are usually overlay because they run on top of the traditional IP layer. Another example is dial-up Internet which is an overlay over the telephone network. Protocols used in overlay networks include JXTA [Antoniu 2007], Gnutella, FreeNet [Samsudin 2008] and Distributed Hash Table (DHT) [Takeda 2008].

2.5.1 P2P Models

P2P can be classified into three types according to the degree of centralization. They can be classified on the basis of the connection and routing methodologies they use. Three models are: the Centralised P2P model, Pure P2P model and Hybrid P2P model.

In the Centralised P2P model peers connect to one or more servers to locate other peers. Once the other peers have been discovered the communication between peers are carried out without use of the central server. Instant Messaging (IM) is one of the example of such model, where peers are retrieved using a main server but connections directly maintained by the peers. One of the advantages of such a model is that resources can be located quickly and efficiently; which is also beneficial to monitor users. On the other hand, this system is prone to failure which affects the rest of the network plus sometime information in a server might be out of date.

A Pure P2P does not have the notion of client or server and does not use any centralised server to assist in peer discovery. Instead it relies on the cooperation between peers by exchanging location information between them. Peers connect to the network and discover other peers using location information gathered from previous

connections and also informing the rest of the network about its existence. Gnutella is a good example of use of such model. One advantage of such model being decentralised, is that it avoids a single point of access and is fault-tolerant. On the other hand, they tend to be inefficient to the lookup process as it can be slow as well as traffic intensive. Also there is an issue with the lookup horizon, where the resource may be available on the network but the lookup process cannot find it. This model works best for systems where resource location is not of highest importance such as Instant Messaging (IM).

The Hybrid P2P model gains location information by cooperation between peers as well as previous knowledge of the resource location. They do not rely on central indexing servers but on the knowledge gained from previous participation in the network, e.g. systems like PAST [Druschel 2001] and Scribe [Castro 2002]. As the location of the resource is related to resource name, regular querying of the network results in regular updates in the routing tables. These systems overcome the issue of limited lookup because of the nature of the underlying routing protocols which guarantee the discovery of resources. These routing protocols include Content Addressable Network (CAN) [Ratnasamy 2001], Chord [Stoica 2003], Kademlia [Maymounkov 2002] and Viceroy [Malkhi 2002], which rely on the use of the a Distributed Hash Table [Takeda 2008] abstraction as a method for lookup and data location. This model works best for system where resource location is of highest importance.

2.5.2 Name based classification

P2P network [O'Mahony 2003] consists of all peers as network nodes. There are links between any two nodes in the network; a participating peer knows the location of another peer in the P2P network. Based on how nodes are connected in the overlay network we can classify into two types: Structured P2P networks [Eng Keong Lua 2005; Hsiao 2003; Hung-Chang Hsiao 2003; Lua 2005] and Unstructured P2P networks.

Unstructured P2P networks are formed when the overlay links are established randomly. Such networks can be established easily when a new peer joins the network, copies links of another node and then forms its own links over time. In

unstructured P2P networks if a peer wants to find data over the network, the query is flooded through the network. However, this technique cannot guarantee of finding objects, for example, if a peer wants to find data which is rare (i.e. as shared by few peers) then there is a greater chance that the search will be unsuccessful. Flooding also causes high traffic in the network so such networks may have very poor search efficiency. Popular P2P networks such as Gnutella [Chawathe 2003] and KaZaa [KaZaA 2008] are good examples of unstructured P2P networks.

Structured P2P networks overcome the problem in unstructured network by using the Distributed Hash Table (DHT) [Balakrishnan 2003; Takeda 2008] technique and allowing each peer in the network to be responsible for a specific part of the content. These networks use a hash function to assign values to every piece of content and then follow a global protocol identifying which peer is responsible for which content. This way when a peer wants to search for specific content it first determines using a global protocol which peers are responsible for that data and then directs the search towards these peers. Chord [Stoica 2003], CAN [Lua 2005] Pastry [Rowstron 2001] are well known examples of structured P2P networks.

2.5.3 P2P Applications

In this section we discuss some of applications of P2P networks.

- Academic search engine the Sciencenet [Liebel-Lab 2008] provides a free and open search engine for scientific knowledge. Sciencenet is based on YaCy technology [Linux 2008], which is free distributed search engine based on P2P network principles, written in Java. Educational institutes can download free java software and contribute their own peers. The idea is to encourage educational institutes to contribute to the scientific network.
- In education and academia, many organisations are trying to apply P2P networks for educational and academic purpose due to fast distributions and large storage capacity features. A project called LionShare [University 2006], enables academic users to search and retrieve academic contents from other LionShare users and many other academic networks across the globe. LionShare disallows any anonymous sharing of files, by only allowing login with university access accounts.

- Military: The US military Department of Defense has started a research project on P2P networks for its modern network warfare strategy. This project first started back in November 2001 [Walker 2001]. Due to security reasons details regarding this project and other similar projects are kept classified.
- Business: P2P networks are widely used in business areas. Business is not only interested in file sharing but also in distributed computing, eMarketPlace [Ghenniwa 2005] and office automation via P2P networks. Features such as real-time collaboration, scalability, storage capacity of contents, running high bandwidth applications motivates using P2P in businesses.
- TV: a number of P2P applications are used to deliver TV content over the Internet such as P2PTV.
- Online gaming: Due to increased demand in online gaming, the number of players' increases and running game servers become costly. In order to reduce cost, Massively Multiplayer Online Gaming (MMOG) [MMORPG 2008] is a type of video game infrastructure which is capable of supporting thousands of players simultaneously. The games are usually played via Internet, many new games for consoles such as Xbox, PlayStation etc can play these kinds of games developed using communication middleware based on P2P system [Rieche 2007; Schiele 2007].
- Telecommunications: P2P networks are also used in telecommunication [Jennings 2006]. Demand of voice and video conference in real-time has increased significantly in recent years. For instance, Skype [Skype 2008], one of the most well known used internet phone applications is based on P2P technology.

In the reminder of this section, we discuss some well known examples of P2P applications. P2P is mainly used in file sharing such as music files, but also in P2P TV to watch TV channels online.

Napster

Napster is an online music service, which was originally file sharing software created by Shawn Fanning [Nagaraja 2006]. Napster was the first widely used P2P music sharing system, which changed the way people used the Internet. It allowed

users to share and download MP3 music files. It violated copyright to download music songs. Napster faced some tough legal copyright cases filed by a number of music bands and singers. The original service was shut down by court order but other decentralised P2P programs such as Kaaza, Limewire appeared based on such idea. A Napster user needs to download and install client software, which connects users to the centralised Napster server. Users can then search and share MP3 files stored on other users local hard drives. Napster servers index all files that reside locally on the hard drives on client machines. Clients submit queries to search for audio files on the Napster server which then lists files that match, including other information such as username, IP and port address, used by the client to connect to the peer that has that the file. Once a client gets all this information they can connect to the target peer and download the file. At this point the Napster server is no longer used.

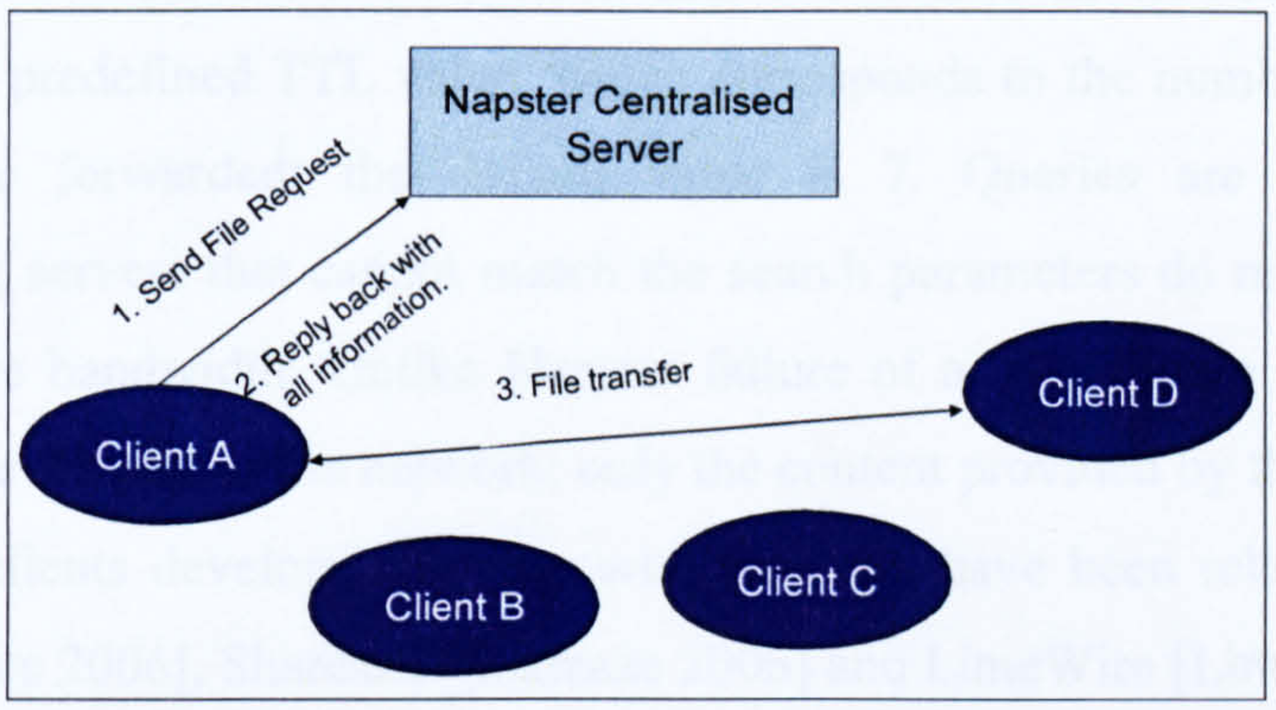

Figure 2.4: Napster in action

Figure 2.4 shows how four clients find and download an MP3 file on the P2P network. Napster is considered as being the first P2P program used to download MP3 music files but has some limitations as it can only be used to share MP3 files. It relied on the Hybrid model, on a Client-Server structure that means if the server failed it caused the entire network to fail.

Gnutella

Gnutella is a simple file sharing program, using the principles of P2P to share files [Gnutella 2008]. Gnutella did not gain that much popularity due to legal problems as faced by Napster. Gnutella uses a protocol for distributed search but also supports traditional Client-Server search. Unlike Napster, Gnutella is different by not being reliant on any central controller to manage contents within the network.

Figure 2.5: Gnutella [Science 2000]

Figure 2.5 shows how a client finds a server by trying to connect to any of a list of known servers that are likely to be available. The advertisement packets also known as Ping or Init [Gnutella 2008] comprise the number of files the client can share and size in kilobytes. In response the server sends the same information and once connected a client knows how much data is available on the network. Gnutella packets have a predefined TTL value, which corresponds to the number of hops that packet will be forwarded; the default value is 7. Queries are propagated as advertisements; servers that cannot match the search parameters do not send a reply, in order to save bandwidth. Unlike Napster failure of a single node in the network does not disable the rest of the network; only the content provided by that node is lost. A number of clients developed for use with Gnutella have been released including BearShare [Peers 2006], Shareaza [Shareaze 2006] and LimeWire [LimeWire 2006].

BitTorrent

BitTorrent is a P2P file sharing protocol used for distributing and downloading large amounts of data. BitTorrent is one of the few P2P file sharing protocol that has attracted millions of users. As opposed to other P2P systems such as Gnutella, KaZaA in which peers sharing different files are organised together, BitTorrent organises peers sharing the same file in P2P network [Guo 2007]. The protocol in Bit Torrent works initially when a provider splits up large files into a group of small files available for the other peers on the network. This process called is *seeding* which allows other users to connect and download the file. Each peer that downloads a part of the file makes it available for other peers to become additional seeders. An increase in number of seeders results in fast downloading and availability of data on the

network. Programmer Bran Cohen designed this protocol in April 2001 and released the first implementation in July 2001 [Cohen 2008] and is currently maintained by BitTorrent Inc. [BitTorrent 2008]. To share files, peers first create a small file called **.torrent* which contains metadata about the files to be shared and *tracker* i.e. is a server that assists in the communication between peers using BitTorrent protocol. Peers that want to download the file must obtain a torrent file using a torrent file peer that can connect to the specified tracker which also gives information about others peers to download the pieces of files. A user usually first locates a torrent file by browsing web pages such as mininova [Mininova 2008], isoHunt [isoHunt 2008] and open it with BitTorrent clients such as BitTorrent [BitTorrent 2008], BitLord [BitLord 2008], BitComet [BitComet 2008] etc.

Bit Torrent clients allow downloads and uploads of torrents using BitTorrent protocols available for the different computing platforms. The clients connect to the trackers specified in the torrent file and obtain a list of peers sharing pieces of a file and client connects to the peer to download pieces of a file. This BitTorrent protocols works as *tit-for-tat* i.e. to encourage fair trading clients prefers to send data to peers who sends data back but this strict policy results in new peers not receiving any data back because they do not have any file to share yet [Cohen 2003]. This principle of fair trading prevents free-riders who only download [Mol 2008]. In BitTorrent failure of the tracker results in other peers unable to locate files anymore. A number of researchers have been working in the area of the effectiveness of BitTorrent systems [Cheng 2008; Dongyu 2008; Pouwelse 2005]. BitTorrent has made a deal with Hollywood to help Warner Bros. to see its films and TV shows [Helm 2006] and Sub Pop [POP 2008] Records releases tracks and video via BitTorrent Inc. to distribute its albums and Blizzard Entertainment [Downloader 2008] use BitTorrent to distribute contents of World of Warcraft game.

The main drawback of BitTorrent is that the use must have a source file i.e. .torrent file and at least one person sharing the complete file. This results that unless a dedicated server continues to offer a file users will be unable to download it and also if peers who have the pieces of a file stop *seeding* it will result in unsuccessful or incomplete downloads for users. When each chunk of a file is downloaded, a hash function checks it against the listed torrent file, if there is a mismatch the chunk is

deleted and the client tries again which protects BitTorrent from many forms of attacks.

JXTA

JXTA is a set of open, generalised P2P protocols that allow any connected device on the network – from cell phone to PDA, from PC to server – to communicate and collaborate as peers [Microsystems 2005]. JXTA is similar to other implementations such as Chord in the sense that it is using DHT, however differs in a way in which the table is managed. When JXTA was first introduced appeared similar like a JINI but the key difference between JXTA and JINI is that it is primarily for the local network while JXTA is for the Internet. In JINI there are bridges to another local network but a path is usually to a specific service in the network, but JXTA applications are less concerned with network boundaries and are less likely to target a specific device or computer [Brookshier 2002].

The JXTA architecture consists of three layers: the core layer, the service layer and the application layer. The core layer provides the main services required for P2P computing such as peer discovery, peer creations, groups, security, groups and mechanism for mobile devices such as Personal Digital Assistant (PDA) [Antoniu 2007; Microsystems 2005; Oaks 2002]. The service layer provides services such as file sharing, protocol translation and authentication. The application layer contains P2P applications built on top service layer such as file sharing. The JXTA protocols are independent of any programming language. The following provides few advantages of JXTA.

- *Interoperability* – JXTA is designed to enable peers with various P2P services to locate and communicate with each other.
- *Platform Independence* – JXTA technology is designed independent of any programming languages, protocols and platform.
- *Ubiquity* – JXTA is designed to access any digital device i.e. cell phone, PC

JXTA provides a common set of protocols for the development of P2P networks i.e. discover peers, peer group creation and organisation, share and discover network services, and communication. Peers can organise into peer groups, depending upon

the services provided by the peers. Peers within groups are identified by unique peer group ID. JXTA provides peers ways to publish, join, discover, create delete and monitor peer groups. There are several benefits of creating peer groups which are:

- Secure environment – creating local peer groups make it easy to enforce security policy. Security may be as simple as username/password – or as complicated as public key cryptography.

- Scoping Environment - to put all the peers offering the same services such as file sharing together in one peer group, which makes it easy for other peers to locate particular services in the P2P network.

- Monitoring Environment – is easy to monitor particular group instead of individual peer.

JXTA uses pipes to send and receive messages among peers. On the features of JXTA is that it uses Secure Unicast Pipes [Microsystems 2005], which is a type of point-to-point pipe that provides a secure communication channel. JXTA itself contains many built-in security features that can be used to build applications. These security features are not enough to build full-fledged secure applications [Brookshier 2002] but they still provide a potential secure base. JXTA uses PureTLS [Systems 2005], Cryptix 3 [Abuelma'atti 2006], Cryptix ASN.1 kit [Abuelma'atti 2006] and Bouncy Castle Crypto[Castle 2005] APIs packages as its security base.

Currently, JXTA provides the following features:

- Transport Layer Security (TLS)[T. Dierks 1999] – also known as Secure Sockets Layer (SSL) is based on public key technology. JXTA provides TLS as medium of secure communications, applications can use these capabilities by using secure pipes, using TLS to guarantee safety against passive attacks [Microsystems 2005].

- Peer Certificates – TLS layer uses certificates to enable its functionalities. Each peer generates its own certificate and therefore acts as its own Certificate Authority, this certificate is also called a root certificate. This certificate is used to sign service certificates that the peer issues for each service that it provides. The root certificate is distributed within the advertisement, which helps the other peer to verify that it is from the peer that claims to issue it.

- Personal Security Environment – every peer is protected by a peer ID and password. This is used to decrypt the private key. This act as first line of defence against users who have physical access to the machine running JXTA peer.

JXTA peers advertise their services via XML documents called Advertisements [Oaks 2002], which enable other peers to interconnect with a peer. JXTA peers use pipes to send and receive messages to each other. Pipe endpoints are referred as input pipe or receiving end and the output pipe or sending pipe. It dynamically binds to peer endpoints, which connect peers that do not have a direct physical link. It provides three modes of communication: Point-to-Point pipes, Propagate pipes and Secure Unicast pipes. All JXTA network resources such as peers, peer groups, pipes; services etc are represented by advertisements. Advertisements are represented as XML [Walsh 1998] documents. Peers discover resources by searching corresponding advertisements. In order to use a discovered advertisement in future, a peer may cache it locally. JXTA is open source and the APIs are readily available for modifications. However, it is a new technology and difficult to find developers material. A most common problem in the release of new JXTA libraries is deprecation errors with existing code and therefore there is a need to rewrite code to benefit from these new libraries.

2.6 Standard Gateways

In this section, we discuss hardware based gateways that connect home computers to the global network i.e. Internet. Some of these standard gateways work with a single computer while some are used to connect computers via a single gateway to the Internet but also allow sharing different devices in a network e.g. printers.

2.6.1 Modem

Modem, or MOdulator-DEModulator, is a type of residential gateway that modulates an analogue signal to encode digital information and vice versa [Walls 2006]. The main goal is to convert digital signals so that they can be transmitted over analogue telephone lines and to reproduce the original digital data at the receiving end as shown in Figure 2.6.

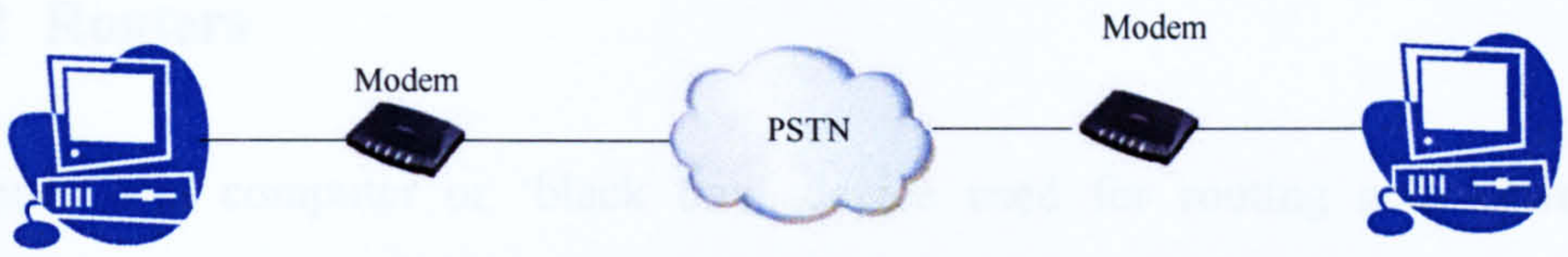

Figure 2.6 : Modem connecting two PC's via PSTN

Traditionally modems are used to connect personal computers with the internet to send and receive data over the telephone lines of Plain Old Telephone System (POTS) [Terashima 2001]. Classifications of modem are generally made by the amount of data they can send in a given time, measured in bits per second (bps). Baud is also used to classify modems, the number of times the modem changes its signal state per second.

If we go back a few years, dial-up internet usually involved an internal or dial up modem installed on a PCI slot to connect to the internet. These usually provide bandwidth up to 28kbps (kilo bits per second) which later increased to 56kbps. Table 2.2 shows some modem standards. With the introduction of broadband faster modems are used to connect called ADSL (Asymetric Digital Subscriber Line)/DSL [Kester 2003] modems ((Asymetric Digital Subscriber Line) modems are used to connect a single computer to a DSL (Digital Subscriber Line) to use ADSL service) or cable modems (used to connect computer to cable television network that uses radio signals transmitted via fixed optical fibre or coaxial cable) [Kester 2003].

2.6.2 **Routers**

A router is a computer or 'black box' device used for routing and forwarding information in the network. Routers are generally equipped with a specialised operating system such as Cisco IOS (the software used in Cisco system routers). Cisco IOS is a package of routing, switching and telecommunications integrated together with a multitasking operating system [Cisco 1992]. Juniper Networks similarly produce an OS called JUNOS [Networks 2008]. Routers used range from small office or home to enterprise network, which may contain many processors and other functions.

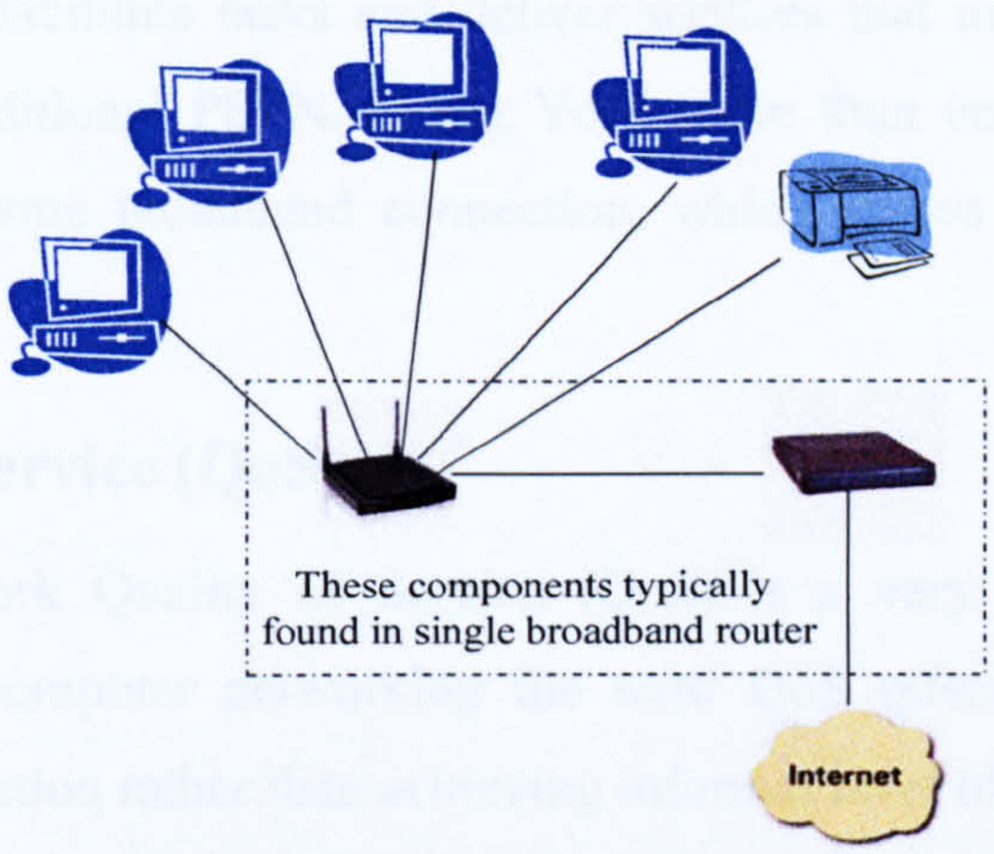

Figure 2.7 : Router in small network

A router connects two or more subnets which are not otherwise connected with each other as shown in Figure 2.7. Routers are usually referred to as layer 3 routers as they operate at layer 3 of the OSI model. As in the above section, we discussed modems that connect single PCs to the internet. With the introduction of high speed broadband, routers are used in homes or small office networks to share one internet connection. Routers not only share internet connections on different PCs but also other devices or services on the network such as the printer in Figure 2.7. Nowadays most routers come with built-in modems called ADSL/Cable Modem Router. Many routers also have functions to provide wireless access usually labelled as wireless routers.

2.6.3 VoIP Gateway

VoIP (Voice over IP) gateway [Gobel 2004] is a device which converts telephony traffic into IP for transmission over a Internet. It allows us to make voice calls using a broadband Internet connection instead of analog phone line. Calls can be made via a VoIP service provider by routing calls via the Internet. These gateways are available as units or as cards. VoIP gateway contains connectors for the IP network and one or more ports for connecting telephone lines. VoIP gateways further divide into two types: Analog Unit: used to connect analog phone lines to it and available for between 2-24 lines. Digital Unit: used to connect digital lines either one or more BRI ISDN lines (Europe), one or more PRI/E1 (Europe) or one or more T1 lines (USA). Voice over IP (VoIP) can facilitate tasks and deliver services that might be expensive to implement using traditional PSTN. Using VoIP more than one phone call can be transmitted on the same broadband connection, which allows additional telephone lines for businesses.

2.7 Quality of Service (QoS)

In any network Quality of Service (QoS) is a very important constraint [Gmach 2008]. In computer networking the term QoS refers to formal resource reservation and allocation rather than achieving informal level of service quality. QoS is the ability to provide different priority to different applications, users or data flows. Flows can be defined as the combination of source and destination addresses and/or socket numbers. It can be defined as packets from certain application [Cisco 2003]. QoS is the ability to guarantee a certain level of performance of data flow. QoS enables provision of better services to certain flows such as either by raising priority of one flow over another. QoS may also try to guarantee that packets will not be delayed or dropped during communication.

In P2P network congestion control and priority based scheduling are very important [Choi 2005]. [Núñez 2006] proposed Extended Service Discovery Protocol (ESDP) which allows discovery of services through queries to the network, propagating using —Sensible Routing. ESDP allows better performance in respect to search time, high probability of success, minimum overhead and improves received

QoS. [Magharei 2007] proposed a solution for streaming of services in live P2P to residential users.

In P2P network protocols that support QoS, two parties that participate in communication may agree on data communication or traffic flow and reserve capacity in the network and during a session it may monitor the achieved level of performance such as data rate and delay. Properties to consider in QoS include latency, jitter and packet loss. Latency is a factor in delay in the network or certain sessions in delivering packets from source to destination, e.g. because a packet takes less a direct route, Jitter is variance in packet delay, e.g. because of variable network traffic or effect of different routes. Packet loss is due to excess network traffic at routers which may drop packets before they arrive at a destination. QoS guarantees are much more important in some networks than others especially for real-time applications such as VoIP [VOIP-Info.org 2003], online gaming, Internet TV [Tanaka 2007] and video conferencing. Since these applications require higher data rates longer delays can cause bad reception in multimedia content. VoIP may require strict limits on jitter and delay while video conferencing require low jitter and latency.

Most P2P network implements web-based services. The goal of Web Services is to make these services accessible over standard Internet protocols, independent of any platform and programming languages [Kontogiannis 2008]. The basic concept of web services is to simulate everything as services by assuming that providers offer available functionality as a service [Milanovic 2004; Wan Nurhayati Wan Ab. Rahman 2008]. QoS is main issue in web services as a number of services available and most of these services offering same functionality but might be from different providers. With the increase in use of web services as a business solution to enterprise application integration, this increases the importance of QoS for web services to service providers [W3C 2003]. However, due to the dynamic nature of web services, it is not an easy task to achieve the desired QoS requirements such as bandwidth and processing time. To provide a better QoS, it is first necessary to identify all the possible QoS requirements for web services. Following we discuss some QoS issues in web services may include performance, reliability, scalability, capacity, robustness, exception handling, accuracy, integrity, accessibility, availability, interoperability,

security, and network-related QoS requirements:[Sivashanmugam 2004; Wan Nurhayati Wan Ab. Rahman 2008].

- **Performance**: The performance of a web service represents how fast a service request can be completed. It can be measured in terms of throughput, response time, latency, execution time, and transaction time. Ideally high quality web service should provide high throughput, low latency, lower execution time and faster transaction time [Anbazhagan 2002].
- **Reliability**: Web services should be provided with high reliability. Reliability is the ability of a web service to perform its required functions under i.e. video, audio. The reliability is the overall measure of a web service to maintain its service quality and it is related to the number of failures per day, week, month, or year [Burstein 2005].
- **Scalability**: Scalability is another issue in P2P networks. Web services should be provided with high scalability. It represents the capability of increasing the computing capacity of service provider's computer system and system's ability to process more users' requests in a given time interval, which is also related to performance [Shuping 2003]. Web services should be scalable in terms of the number of operations or transactions supported. Scalability can be achieved by replicating web services [Bravetti 2008], which also results in increasing the performance.
- **Availability**: is the quality aspect that whether the Web service is available for immediate use. It represents the probability that a service is available. Larger values represent that the service is always ready for use while smaller values represent whether the service will be available at a particular time. Also associated with availability is time-to-repair (TTR) [Anbazhagan 2002]. *TTR* represents the time it takes to repair a service that has failed. Ideally smaller values of TTR are desirable. Availability is the probability that the system is up and is related to reliability i.e. larger values of availability represents high reliability [W3C 2003].
- **Robustness**: Web services should be provided with high robustness. Robustness represents the degree to which a web service can function correctly even in the presence of invalid, incomplete or conflicting

inputs [Shuping 2003]. Generally, web services should still work even if incomplete parameters are provided to the service request invocation.

- **Capacity**: Web services should be provided with the required capacity. It is the limit of the number of simultaneous requests web service can action, which should be provided with guaranteed performance [Shuping 2003], web services should support the required number of simultaneous connections.
- **Integrity**: Integrity for web service to prevent unauthorised access or modification to computer programs or data. There can be two types of integrity: data integrity and transactional integrity. Data integrity relates to modification of transferred data in transit while transactional integrity refers to a procedure or set of procedures to guarantee to preserve database integrity in a transaction [Jiang 2008]. All the activities have to be completed to make the transaction successful. In case of failure, all the changes made are rolled back [Anbazhagan 2002].
- **Security**: as web services are delivered over the public Internet, there is a growing concern about security. The web service provider may apply different approaches and levels of providing security policy depending on the service requestor [D'Ambrogio 2007]. Security in web service to provide confidentiality by authentication both parties i.e. service requester and service provider and can be achieved with data encryption and access control [Vroonhoven 2006].

Some notable research done in managing QoS in web services are as follows:

[Shuping 2003] that web services (WS) didn't gain much attention or slow adoption due to the fact that areas like WS-Transaction [IBM 2005], WS-Security [Thompson 2003] and WS-coordination [IBM 2005] is still yet to be seen. Web services still need to address questions such as will web service meet my performance requirement? Will web service be reliable? Will my requested web service be available? The author suggested a new service discovery model considering QoS constraints when searching for web services. However, current UDDI [Blake 2007] model is that it limits the service discovery to functional requirements only, there is a possibility these may be more than one web service available that can meet the functional requirements with different QoS attributes. The proposed framework is a regulated model that can co-

exist with the current de-regulated UDDI registries [Blake 2007]. The current de-regulated registries can offer services to people to whom the QoS is not important while the regulated registries based on the model can serve to the applications needing QoS assurance. The proposed new registry differs from the current UDDI model by having information about the functional description of the web service as well as its associated QoS registered in the repository. Lookup could be made by functional description of the desired Web service, with the required QoS attributes as lookup constraints. The new role in this model is the web service QoS certifier that does not exist in the original UDDI model. The certifier verifies the claims of QoS for a web service before its registration.

With the increasing number of web services on the web, the service consumers may be presented with a group of services offering the similar functions, may afford different QoS and it is difficult to find out the appropriate one among the large numbers of web services. On the basis of the analyses of the known models, a QoS-aware model for web services discovery by introducing QoS Broker is proposed by [Gang 2009]. Different service providers offering similar functions will require sophisticated patterns to select appropriate web service. For example, the tradeoffs between quality and cost, or invocation of another trade service determining the QoS of various service providers. The author proposed a new model in which QoS is taken as constraints when searching for web services, which would give some confidence to the service consumers about the availability and efficiency of services. This model does not modify the standard UDDI interface and the client side software can transparently plug on to it. This model introduces the monitoring and valuation mechanism to collect continuously the feedback reports to keep QoS information always updated. When the QoS Broker (including QoS Database, Publish, Lookup, Monitor and Valuate module) receives an inquiry from the WS Consumer, it searches the local UDDI registry for related results, if result is insufficient, then it will search with remote QoS Brokers. The QoS Broker then filters and merges all these results. If the inquiry is service related, the server sorts the service results according to the QoS summaries and then sends the results back to the WS Consumer.

With introduction of more and more web services on the web, web service's discovery mechanism becomes essential. UDDI is an online registry standard to

facilitate the discovery of business partners and services but the current UDDI is lacking of ability to predict the quality of service. To address this problem, author proposed UX (UDDI eXtension), a system allowing service consumer to discover services with good qualities [Chen 2003]. In each domain, the service consumer QoS feedback is received and then stored in a local database. By sharing these experiences from the entire service consumer in the local domain, the system can predict the service's further performance. The UX architecture is extended to incorporate the semantic service descriptions in the registry for precisely matching out of the available service capabilities in the matching procedure as current keyword matching can't provide precise and flexible matching result. It is designed for several template QoS metric classes for different kinds of web services and provides better granularity for brokers to predict the service's performance.

2.8 **Security**

Security is an important concern in any computer system [Cameroon 2006; Curbera 2006]. P2P networks are gaining considerable attention today so security is one of the most important concerns [Palomar 2006] In this section, we discuss security issues within P2P networks. Organisations implement different levels of security depending upon their requirements [David 2004]. Implementing different security levels depends on what we need to protect and against what. Figure 2.8 shows a secure home network with a main firewall [Cameroon 2006] and then each computer in the network with a personal firewall. Important security features to achieve are authentication, access control, and data confidentially for communication and stored data, connection control and protection against malware.

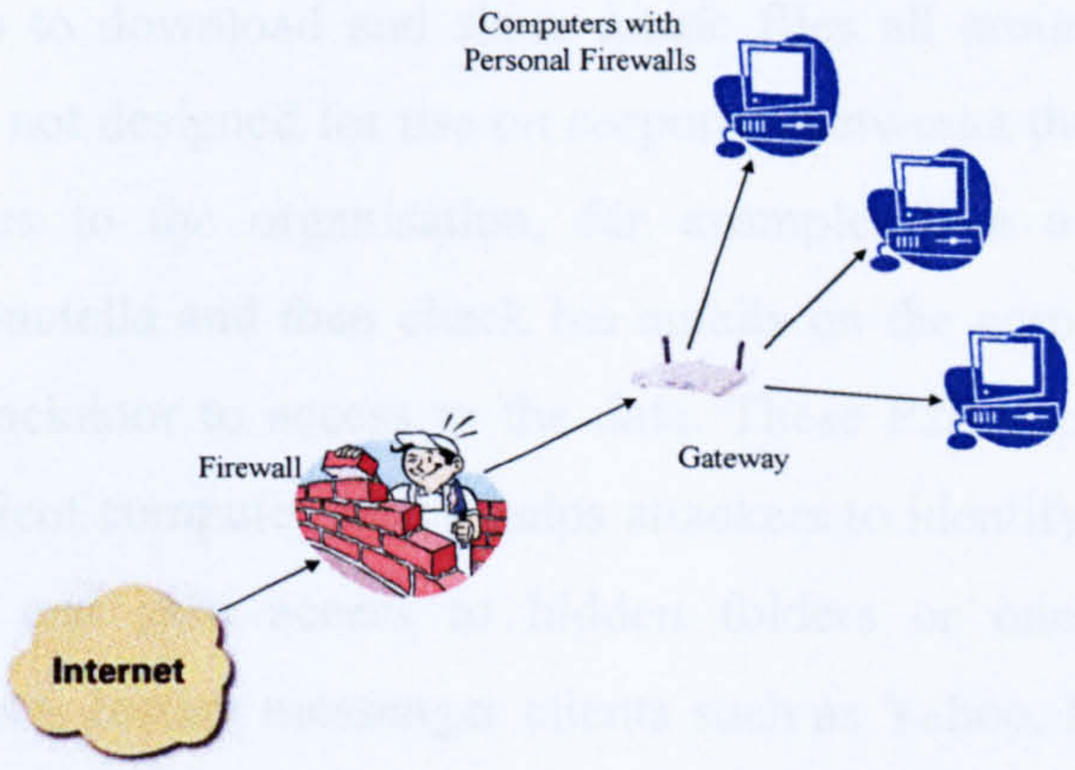

Figure 2.8: Secure network

Connection controls are of key importance as if we can keep our connection secured, it makes it difficult for hackers to damage or steal the data. Access control, can be dealt with by implementing security policies [Curbera 2006; Kocbek 2007], for example, by grouping together users who can access particular types of information. In Client- Server, security policies can be forced to restrict users to a certain level of access i.e. user, administrator, power user etc. P2P networks are open to various forms of threats such as break-in, espionage and Denial of Service [Haggerty 2005]. P2P networks such as KaZaA, LimeWire that allow users in an organisation to download and use copyright material and share files which violates an organisation security policy. This brings numerous problems for an organising network such as using bandwidth and subject to virus attack via downloading virus infected files. Due to the nature of P2P networks decentralised security administration and data storage, users can install and configure their own P2P clients so server-based security policies may be of no use.

P2P users could also possibly download and install Trojans that cause serious damage. For example, a file that looks like Instant Messenger or an mp3 file could allow access to the user's computer. An attacker would then be able to do serious damage or steal more information from the user which could be an organisation's confidential information. A P2P application is installed on the trusted device, which allows communication through a firewall with other P2P users. Once a connection is established, an attacker can gain remote access to the trusted devices for the purpose of stealing information, launching denial of service attacks [Haggerty 2005] or gaining access to network resources. P2P applications such as KaZaA, Napster or Gnutella enable users to download and share music files all around the world. As these applications are not designed for use on corporate networks they may introduce serious security issues to the organisation, for example, if a user starts a P2P application such as Gnutella and then check his emails on the corporate intranet, an attacker may use a backdoor to access to the data. These P2P applications provide direct access to the client computer which helps attackers to identify which operating system a client has and gain access to hidden folders or ones which contain confidential information. Instant messenger clients such as Yahoo, MSN also expose information threats to the company if users use it to discuss sensitive information;

attackers can read all the information as these applications do not widely use encryption.

Different options are available to help build secure web service depending on different organisation criteria [Singhal 2008]. Number of elements are involved when considering security for web services such as choosing between message and transport layer security, authentication, data confidentiality, authorisation, accountability, data encryption, traceability and auditability [Rot 2008]. In the following we briefly discuss some of these elements:

- **Authentication**: users or services that access services and data should be authenticated. It can be either *direct authentication* [Microsoft 2008] where the service validates credentials directly with an identity store, such as a database or service directory or *brokered authentication* where a trusted authority is used to broker authentication services between a client and a service [Microsoft 2008]. For example, a large electronic distribution company uses web service providing catalogue information to the merchants that provide online shopping services. Using web service from their web application merchants display current items available from the distributor. In this scenario, merchant accessing distributor web service using their web application must be authenticated. Simple solution is username and password i.e. to allocate all merchants with a username and password to use every time they access a merchant web service. *Direct authentication* in this scenario, authentication can be done directly i.e. requires the presentation of credentials typically a username and password. The service uses these credentials to authenticate the request. *Broker authentication* in this scenario, authenticate can be done via broker i.e. credentials are used to authenticate with the broker, which issues a security token. The security token is then used to authenticate with services [Youxiang 2008].
- **Authorisation:** Authorization is the process of determining whether an authenticated client is allowed to access a resource or perform a task within a security domain. In above scenario, it is necessary to authorise every merchant to check if they can access a particular web service. For example, it might be the case that some merchants can only allow access the stock information

about certain items. This can be achieved either using role-based or resource-based authorisation [Hu 2008]. In *role-based authorisation* a distributor can associate clients and groups with the permissions that they need to perform particular functions or access resources. Distributor can add a user or group to a role, the user or group automatically inherits the various security permissions. *Resource-based authorisation* is performed directly on a resource, depending on the type of the resource and the mechanism used to perform authorization. Distributor can allow or deny any merchant to access particular or group of resources.[Cameroon 2006]. Resource-based authorization can be based on Access Control Lists (ACLs) or URLs.

- **Parameter Manipulation:** refers to unauthorised modification of the data transfer between web services i.e. an attacker can intercept messages during transmission and modify them before sending to the destination. This usually occurs when messages are not signed or encrypted. Web services need a mechanism to check if data or a message is not changed on arrival at destination and also verify the origin of the message. This can be achieve using digitally signed messages [Song 2009].
- **Data Confidentiality:** an attacker can see messages transmitted between services i.e. an attacker can monitor messages using network monitoring software and steal sensitive data in it which might be credential information due to network eavesdropping [Chen 2007]. This usually occurs when credentials are passed in plaintext or no message level encryption is used. Message replies travel through a number of intermediate points, can be captured by an attacker who can copy messages and reply to a web service pretending to be the client. This usually occurs due to no encryption or when messages are not signed digitally [Vroonhoven 2006]. This can be achieved using encryption/decryption of the messages [Kojiro 2008].
- **Transport layer security vs. Message layer security:** in transport layer security the underlying operating system handles security. For example, for data confidentiality, Secure Sockets Layer (SSL) is a common transport layer approach that is used to provide encryption [Tanenbaum 2003]. If a message needs to go through multiple nodes to reach its destination, each intermediate node must forward the message over a new SSL connection. In this model, the

original message from the origin is not cryptographically protected on each intermediary because it traverses intermediate servers and additional computationally expensive cryptographic operations are performed for every new SSL connection that is established [Mabanza 2007; Tanenbaum 2003]. In message layer security all the information related to security is encapsulated in the message. Securing the message using message layer security instead of using transport layer security has several advantages such as increased flexibility, support for auditing and multiple protocols [Microsoft 2008].

2.9 Summary

In this chapter, we presented the background work. We started with a brief history of computer networks, from the early history of the Internet. A number of topologies that can be used to interconnect computers in small or large networks were outlined. Every topology has some advantages and disadvantages. We also discussed Ubiquitous/Pervasive Computing with some work done and challenges in this field. We also discussed Networked Appliances in relation to home networks. Some notable research work done in seamlessly interconnecting Networked Appliances within home networks were presented. This chapter discussed P2P networks, its merits/demerits and some challenges in this field. We also discussed some well known P2P applications such as Napster, Gnutella, and KaZaA etc. Each P2P model was discussed in terms of functionality, limitations, structure, discovery and failure of particular service or device. The chapter concluded on security in P2P networks. This section discusses the importance of security in networks and how it is possible to achieve it.

In the next chapter we mainly discuss network gateways that are used to seamlessly interconnect devices in the network. We discuss some notable research and point out their shortcomings. This review enables us to specify our novel idea of building a framework that enable to access Networked Appliances in P2P network in Chapter 4.

CHAPTER 3

3 RELATED WORK

With recent advancements in technology home networking is gaining more popularity. Home networking enables users to share services within the home. As the number of companies involved in making network devices increases, prices become affordable. Small business and home networks equipped with dialup Internet access using modem and with little investment may require high bandwidth broadband such as Digital Subscriber Line (DSL) such as AOL, BT or cable. Most providers give initial installation free of cost or with little charge. Most of these devices are very user friendly and with very technical knowledge a user can easily manage it. Many home networks are setup to share an Internet connection and networking devices are usually equipped with firewall/security features or can be using operating system features. In the following section we discuss a number of ways to connect computers in a small office or home network internally and then connect them to the Internet.

In this chapter we provide an overview of the work carried out in the research area of gateways relevant to this thesis. We also discuss Service Oriented Architecture which is a software architecture which allows different applications to exchange data with each other and separate functions into units which can be accessible over network and can be combined together and reused.

3.1 Service-Oriented Architecture (SOA)

In terms of software engineering, SOA is a software architecture which allows different applications to exchange data with one another within business processes. SOA separates functions into a number of units or services, which can be accessible over the network so they can be combined together and reused. A number of business applications can be built reusing the same functionality [Krafzig 2004], SOA emphasises reusing the functionalities instead of rebuilding them again. Technically this is termed 'Loose Coupling', reducing dependencies among systems without

affecting any necessary dependencies. In this case *service* is a unit of work done by a service provider to achieve a desirable result for a consumer or user, for example, a payroll system for one consumer business which could be used for another consumer with some or no change. Figure 3.1 shows the evolution of SOA gateways, other types are discussed later in this chapter.

OAS

[OASIS 2007] defines SOA as "*A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.*"

SOA can be defined as a group of services that communicate with each other. Communication either involves simple data communication or could be two or more services performing some activity. SOA is used in many online applications; for example a CD player which can play any CD. In this scenario the CD player is offering a CD playing service. The CD player does not bind with a particular CD, so the same player can be used to play any CD. Similarly in the case of software services they can be reused or amended as the user demands. This kind of technology is ideal for implementing a flexible gateway service. A majority of services a home user might want to access such as CD player service are naturally described and implemented in terms of services. To explain in more details, take an example of an ecommerce website where users can shop online. The same interface is used by a number of websites or may be with some minor changes.

Web Services [Booth 2004] are used to implement a service-oriented architecture. The goal of Web Services is to make these services accessible over standard Internet protocols, independent of any platform and programming languages. There are three main building blocks in SOA:

- *Service Provider* creates a web service, publishes it and provides information to a service registry [Krafzig 2004] It is up to the provider to decide how to publish the service, what category it should be listed, its security and cost.
- *Service Broker* also known as service registry, is responsible for making web service interfaces available for potential service requestor. Universal Description Discovery and Integration (UDDI) [Blake 2007; OASIS 2004] specification is used to publish and discover information about web services, other service broker technologies include Electronic Business using eXtensible Markup Language (ebXML) [Yohan 2007].
- *Service Requester* or web service client locate entries in broker service registry to bind to the service in order to invoke or use one of its web services.

3.2 Gateways

According to the Oxford dictionary a gateway is defined as a passage that is or may be closed by a gate; an opening through a fence or wall [Oxford 2009]. The term gateway is used by webmasters and search engine optimizers as a webpage designed to attract visitors. Gateway also defines a link between two computers that acts as a portal between two programs to enable them to share information and translate protocols. In the generic terms a gateway can be considered as an entry point. In networking terminology, network gateways interconnect networks with different, incompatible communication protocols [Sunshine 1990]. In networking a gateway is commonly used to transfer data between different networks or one network and the Internet. The computers that are used to control traffic flow within local network or Internet Service Provider (ISP) are gateways. In networking gateway is usually associated with router or switch, which knows where and how to direct packets in the network. Gateway also enables connection from a LAN to a WAN; connecting LAN via local server and then to the Internet. The traditional gateway is used to interconnect devices within a LAN, for example, home or office. The gateway has a

central register holding information about the devices in the network [Wils 2002]. All the devices register when first connected to the network and this register is used to locate the device within the network. When a search is conducted for a particular device this central register is queried. In addition to the registering of devices, the gateway may be responsible for naming and addressing, security, protocol translation and Quality of Service [Jiang 2008]. All the devices that connect to the gateway obtain a unique name and address, which is used for future communication. As the gateway is central, it can maintain a high level of security by authorising username and password and encrypt/decrypt data accordingly. Some of these gateways are designed in such a way that devices from different manufacturers can communicate through it. On the other hand, if the gateway fails, the whole network may fail or become partitioned.

The network gateway can be implemented in hardware, software or as a combination of both [Jiang 2008]. Within the network, one computer is designated as a Gateway, which is used to make communication possible between devices. The major task of the networking gateway is protocol translation; it receives the data from one machine, does the relevant translation and sends data packets to the destination devices as shown in Figure 3.2. The gateway may also be responsible for NAT (Network Address Translation) [Garg 2007] to translate IP addresses between a private and the public Internet.

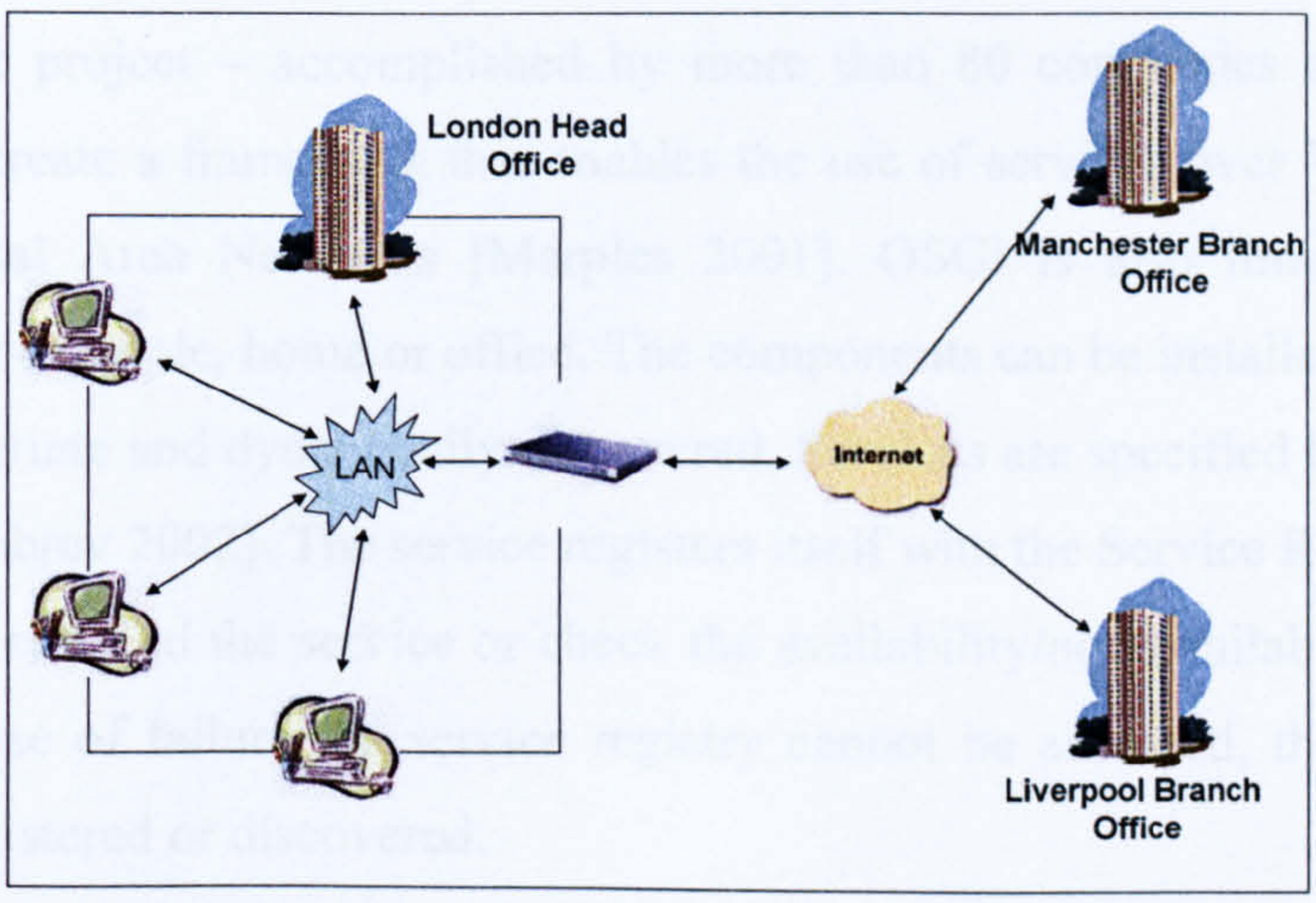

Figure 3.2: Gateway in network

A home or residential gateway is hardware device connecting a home network with the Internet. The residential gateway uses NAT to provide network access to all computers in a home network to share one IP address and Internet connection. The residential gateway acts as a bridge and interacts between DSL or cable modem and the internal network.

Residential Gateways [Bull 2002], are an intelligent network interface device used to provide services to access devices from remote locations across the Internet. Most of these residential gateways are used to access different devices within the home environment. The basic function of the home gateway is to do bridging, protocol and address translation [Hartog 2004] between external broadband and internal home networks. It allows the user to use their home networks and control devices based on e.g. OSGi [Dobrev 2002] or UPnP [Microsoft 2004]. These gateways are external to the consumer premises or located in the network itself [Bull 2002]. Traditionally there is a single service provider (the service aggregator), which delivers the services via a single access route [Bull 2002]. The security resides on the servers of the service provider that controls the gateway [Hartog 2004]. As these gateways are centralised, in case of failure, the whole network becomes unavailable. In the following subsections we discuss some examples of residential gateways.

3.2.1 Open Service Gateway Initiative (OSGi)

The Open Gateway Services Initiative (OSGi) [Marples 2001] was founded in March 1999. The project – accomplished by more than 80 companies around the world – aims to create a framework that enables the use of services over Wide Area Networks to Local Area Networks [Marples 2001]. OSGi is also limited to the specific range, for example, home or office. The components can be installed, updated or removed at anytime and dynamically discovered. Services are specified by use of a Java interface [Dobrev 2002]. The service registers itself with the Service Registry, by which the clients can find the service or check the availability/non-availability of the service. In the case of failure the service registry cannot be accessed, therefore no service can be registered or discovered.

OSGi provides a managed Java framework that supports the deployment of service applications known as bundles. It supports automatic detection of attached hardware and can automatically download and start device drivers. Devices plug and unplug at anytime and it can respond to it immediately. It provides supports to given devices as well as dynamic discovery and downloading of device drivers.

Three main components:

- Drivers: used to do registration of the services.
- Device Manager: coordinates the relationship between certain devices so that they can present multiple representations of the same device, and initiates the process of downloading new drivers.
- DriverLocator: where vendor specific knowledge about the location of drivers is located. Device manager uses this service to identify and download new drivers when they are needed.

Applications within OSGi are called bundles. Devices can download bundles on demand and remove them when they are no longer required. When a bundle is installed, it can register a number of services to be shared with other bundles under the framework. Bundles can register new services, receive notifications about the state of services, or look up existing services to adapt to the current capabilities of the device. New bundles can be installed for added features or existing bundles can be modified and updated without requiring the system to be restarted. These bundles can be remotely installed, started, stopped, updated and uninstalled without requiring a reboot. Figure 3.3 shows the OSGi system design. In the OSGi Service Platform, bundles are the only entities that allow the deployment of Java-based applications. A bundle is comprised of Java classes and other resources which together can provide functions to end users and components called services to other bundles. A bundle is deployed as a Java-Archive (JAR) file.

Figure 3.3: OSGi System Diagram [Marples 2001]

Configuring an OSGi framework is human centric but in most cases managed and controlled via centralised controller by service providers. Service discovery and composition is based on proprietary communication and middleware protocols, which is somewhat restrictive as distributed computing and service models are becoming more pervasive. As devices and services are more heterogeneous, this makes management of such framework more complex. As technologies become more sophisticated, control placed on devices and service integration become more difficult. Due to this device and service providers will use different communication standards, therefore interoperability is a problem and requires a more efficient solution. New architectures need to be developed to overcome these restrictions on current OSGi standard.

3.2.2 Universal Plug n Play (UPnP)

For the last few years, the idea of 'plug and play' has become very familiar. Devices are connected to the computer, which instantly starts working as they are automatically detected by the operating system. Microsoft along with other companies are working on the idea called UPnP [Microsoft 2004], which uses TCP/IP and HTTP to automatically discover, configure and control services [Bull 2002]. It is a set of protocols that are used by devices to advertise their services over the network, which can then be discovered by other devices in the network [Microsoft 2004]. One aspect of UPnP is that the current specification does not address security [Bull 2002;

Microsoft 2004]. Users cannot prevent access to the devices, which restricts its use to low risk environments such as the home or office. The UPnP allows P2P networking of PC's, Networked Appliances (NA) and wireless devices. UPnP supports zero-configuration networking; any vendor device can dynamically join the network, obtain an IP address, and broadcast its capabilities. Devices not only broadcast their services but also discover other devices. There are no restrictions on the devices so devices can leave a network at any time. The main limitation of UPnP is that one cannot access service outside a local area network. All the communication in UPnP happens over Internet Protocol (IP) [Lee 2007], a target must obtain an IP address before it can join a UPnP network and by using IP addresses, a control point can contact other UPnP devices within same subnet [UPnP 2006]. Messages within UPnP are sent using SOAP [Louridas 2006].

The first step in UPnP is discovery, when a device joins the network the UPnP discovery protocol allows the device to advertise its service to a control point in the network. Discovery messages contain information about devices such as name, services offered etc. UPnP uses Simple Service Discovery Protocol (SSDP) [Wu 2007] for this task. The next step is description to enable the control point to know about the device and its services. The control point can retrieve this information via a URL in the discovery message provided by the device during the discovery stage. This device description is expressed in XML [Knauth 2007] and includes vendor name, serial number etc. The next step is Control; after the control point retrieves the description of the device it then sends actions to the device services. Control point uses the control URL for the service provided in the description step to send suitable control messages. These control messages are also expressed in XML using Simple Object Access Protocol (SOAP) [Louridas 2006].

Next is Event notification; UPnP description includes a list of actions services respond to and the state of variables that model the state of the service at runtime. The service publishes updates by sending event messages whenever these variables change. These messages are also expressed in XML and formatted using General Event Notification Architecture (GENA) [Chih-Lin 2007]. Control point may subscribe to receive these messages and events are designed to keep the control point updated about the effects of any actions. The final step is Presentation, which allows

the control point to retrieve a page into a web browser, if the device has a URL for presentation. This allows a user to control the device and view its status.

The main limitation in UPnP is that it is human centric and so does not provide any mechanism for automatic discovery and composition of services without any human intervention. Attribute-value pair matching is used for discovery which is very restrictive. Devices can only conform to the specifications which may isolate a number of other Networked Appliances using different standards.

3.2.3 Devices Profile for Web Services (DPWS)

Devices Profile for Web Services is another notable research that defines set of implementation constraints to enable secure web service messaging, discovery and eventing on resource-constraint devices [DPWS 2006; Jammes 2007]. DPWS was developed by Microsoft and printer manufactures allowing sending secure messages to and from web services, dynamic discovery, describing, subscribing to and receiving events from a web service. DPWS is a type of SOA targeting device-to-device communication such as Open Services Gateway Initiative (OSGi), Home Audio/Video Interoperability (HAVi), and Universal Plug n Play (UPnP). DPWS's objectives are similar as UPnP but DWPS is fully aligned with Web Services technology to allow seamless integration of device provided services. Its specifications was first published in 2004 and defines an architecture in which devices run two types of services: *hosting services* associated directly to a device and *hosted services* are mostly functional and depend on the hosting device for discovery. The DPWS focuses on IP-capable devices, many of these are still resource-constrained by desktop and server standards but are ready to contribute to general web services scenarios involving services already deployed in the home, office network [Microsoft 2008].

In addition to hosted services, DPWS also specifies a set of built-in services: *Discovery service* used by the device connected to a network to advertise its services and discover other services, *Metadata exchange services* provides access to the device hosted services and their metadata and *Published eventing services* allows other devices to subscribe to event messages by a given service. The DPWS protocol stack is shown in Figure 3.4.

DPWS builds on core Web Services standards such as WSDL 1.1, XML schema, SOAP 1.2, WS-Addressing. DPWS gained attention from manufactures recently after successful demonstration of automation system in Consumer Electronic Show [CES 2008]. In DWPS discovery is usually done by sending probe messages over UDP multicast, indicating a client is looking for a particular service i.e. print service defined in WS-Discovery as part of a multicast discovery protocol [Zeeb 2007]. The client device listens to the probe messages, e.g. print service, and responds with a Probe Match message defined in WS-Discovery directly to the client. A Probe Match includes three pieces of information; the address for the device, transports where the device may be reached and security requirements. If a client requires security, the next message is to setup a secure channel between client and device. This channel protects the confidentiality and integrity of all messages between client and device. Each device uses a device certificate as authentication; a device may use self-signed certificates that require the user to enter a device-specific PIN into the client. If a client wants to find out more about a device it may send *GetMetaData* messages directly to the device, in response the device returns information such as

manufacturer, serial number etc. Client can send a control message to start using the service i.e. to start a print job. The service sends an event to the client i.e. about the print job such as print job, number of pages printed etc. One successful project is SIRENA, based on DPWS, which intends to create a service-oriented framework for specifying developing distributed applications in diverse real-time embedded computing environment [SIRENA 2005].

In DWPS discovery take place in few steps, clients usually first discover a service and then in later steps obtain service description. In some cases after obtaining service description, a device might not contain services desired by client. Device discovery defined in DPWS may cause interoperability problems, may lead client to be unable to locate all requested services. Length restrictions for message fields defined in the message section may lead to interoperability issues as the client side considers restrictions sending messages while the device side could reject messages that exceed the restrictions. There is no mechanism in DPWS to rediscover alternative services as service configurations are manually created. Composition remains operational as long as all services within a composition remain operational, any fault need to be corrected manually.

3.2.4 Digital Living Network Alliance (DLNA)

Digital Living Network Alliance (DLNA) [DLNA 2006] formerly known as Digital Home Working Group (DHWG) [DLNA 2006] is an alliance of leading companies in Consumer Electronics (CE), mobile and Personal Computing (PC). DLNA aims to align companies to have industry standards which will allow products from different companies to be compatible with each other.

DLNA aims to create a framework that enables interoperability between devices within or out of three domains such as CE, mobile and PC. In order to give the user facility to interconnect devices seamlessly within three domains, they must address some challenges as follows:

- Products designed for the home should be easy to install;
- Must be affordable;
- These products must interoperate with all other devices;

- Current open industry standards are often too flexible leading to different vendor's products failing to interoperate so they need to design better industry standards to achieve better interoperability.

Another objective of DLNA is to create wired and wireless interoperable network of Consumer Electronics (CE), mobile and Personal Computer (PC), which enable devices to seamlessly connect to each other for sharing information. To deliver interoperability DLNA emphasises three key elements namely industrial collaboration, standards-based interoperability and compelling product.

Different manufacturers are trying to address the interoperability issue within their products and to develop standards that solve interoperability issues. A number of leading companies joined this alliance such as Motorola, Philips, Samsung, Nokia, Microsoft, HP, Sony and Intel. Different vendors are trying to manufacture devices that enforce standards of DLNA. It will enable different vendor devices to be interoperable. Due to rapid advancement in these domains; these standard keep changing to address new devices interoperability.

DLNA published some requirements in order to deliver interoperability within the home; which allow different vendors to participate. These requirements are mainly based on interoperability between networked entertainment and media devices. In future they are going to broaden these requirements in order to accommodate new technologies. These requirements concerns:

- Media formats
- Device discovery, control and media management
- Media transport
- Network stack
- Network connectivity

In the case of device connectivity whether wired or wireless it uses Ethernet, IEEE 802.11a/b/g and Bluetooth. Currently this is based on IPv4 for networking but future specifications will include IPv6. UPnP is used to achieve device discovery and control. HTTP is used for media transport and supports a number of media formats, categorised as required or optional. Required formats are JPEG, LPCM, and MPEG2

while optional formats are PNG, GIF, TIFF, MP3, WMA9, AC-3, AAC, ATRAC3plus, MPEG1, MPEG4 and WMV9. Future implementations will include MPEG 4 and JPEG2K. Interoperability guidelines include that technology should be based on standard bodies, SIGs (Special Interest Groups) [Machinery 2004], and industry forums. It also includes that in case of multiple DLNA-approved technologies are specified, they should bridge or translate as required between any of two technologies.

DLNA uses IPv4 for connectivity as IP allows applications running over different media to communicate easily. Device and service discovery enables devices to automatically discover other devices and their capabilities, through which devices can share different services offered by these device. DLNA uses UPnP™ Device Control Protocol Framework (DCP Framework) [UPnP 2006], which address all these needs to discover, control and share services among devices. DLNA incorporates OSGi and inherits the limitation with OSGi as discussed above.

3.2.5 Home Audio/Video Interoperability (HAVi)

HAVi [HAVi 2004] is another approach that provides interoperability between audio/video devices within home networks. Audio and video devices within the home network can interact with each other and allow devices to interact via another device. Devices from different manufacturers can interact in HAVi regardless of network configuration. HAVi is open, platform independent and language neutral; which provides CE manufacturers the freedom to develop interoperable devices. These can be connected using HAVi, can share their resources and can build more applications such as having two VCRs connected to two tuners with either VCR able to record the signal from either tuner. Within HAVi there is no single master controlling device. Any device within HAVi can control other devices. Controlling devices and controlled devices can be located anywhere within the network. Any device within HAVi can act as controlling and controlled at the same time. Currently HAVi uses digital IEEE-1394 network. IEEE-1394 provides bandwidth up to 800Mb/s; which enables isochronous communication and simultaneously handles multiple real-time digital AV streams. The software elements comprising HAVi are 1394 Communication Media Manager, Registry, Event Manager, Messaging System,

Resource Manager, Stream Manager, Device Control Module, Functional Component Module, Device Control Module Manager and Application.

HAVi provides inter-relationship between other networking standards in respect of audio/video prospective. The main benefit of such inter-relationship is to build bridges with other networking standards as it provides additional benefits to consumer. Irrespective of underlying hardware or implementation details using HAVi the software API and the HAVi bridges, CE manufacturers can allow audio/video devices to interoperate within and across different network. This specification is designed to address interoperability for audio and video systems, which does not address wider interoperability issues.

3.2.6 ePerSpace

ePerSpace is a project under the EU 6th Framework programme for the development of personalised communication services within home networks. ePerSpace [ePerSpace 2005] addresses key requirements for Networked Appliances and home networks. It aims to provide abilities within home devices such as TV, smart phones, PC's etc to exchange data and access external services provided by these devices; which increases user acceptability of such a system. It also provides a solution for interoperability problems within home devices. The ePerSpace provides distributed services that can be accessed via Open Access Network (OAN), which enables the user to access personalised services from anywhere. The approach of ePerSpace is to create a trusted integrated framework to seamlessly interconnect audio and video devices.

The ePerSpace framework provides Global Network Integration and Interoperability which allows interconnecting audio and video to exchange its content between distributed services in a secure manner. Using this framework home and personal devices can build a personal environment that can be controlled using tools provided by Rich Media Object Management. This standard is mainly used to build a dynamic personalised network within the home network. This standard attempts to move one step further than standards discussed above, by adding a level of intelligence that provides context adaption mechanisms based on user profiles. But

again it is choreographed solutions which will be difficult to implement in pervasive ad hoc environments. New devices, standards or services have to conform to the ePerSpace specifications in order to be integrated within the environment.

3.2.7 Networked Appliance Service Utilisation Framework (NASUF)

NASUF (Networked Appliance Service Utilisation Framework) framework allows the operational functions provided by different appliances to be dispersed within the network and used to create high-level application [Merabti 2008]. NASUF allows the services provided by devices to be automatically composed to produce value-added services. NASUF combines advances made in P2P networking, ontology, semantic web services and signature matching, which allows for hosting and discovering unstructured services [Merabti 2008]; enabling semantic interoperability by evolving knowledge structures between different vocabularies; and publishing functions offered by complex devices as individual services. The framework used a service-oriented middleware to discover and combine devices using machine-processable descriptions that allow devices and functions to be selected based on application requirement. This framework addresses a number of issues relating to service-oriented networking, networked appliances, service discovery, dynamic service composition and self-adaption. This framework allows complex devices to publish their functions as independent services so that they can be discovered by other devices within the network. It allows devices and the services they provide to be offered to other devices and services without registering with centralised authorities. Functionality offered by devices can be discovered, composed and used by other devices within the environment. This framework provides a mechanism to enable devices to determine what services are offered by other devices in the network, it can use. Services discovered are used based on their capabilities and service interfaces that match required service capabilities.

]

Figure 3.5 shows the NASUF framework consists of a number of components. DiSUS allows devices to host and discover unstructured services in P2P networks. DistrES allows ontological structures to be evolved within P2P networks based on general consensus and resolves terminology differences between concepts that are syntactically distinct but semantically equivalent. Also developed, the Device Capability Service determines if the device providing the service has the required hardware/software capabilities to execute the service request. The SISM Service performs dynamic service composition between service enabled devices in a P2P network based on device and capability matching. This work forms part of a bigger research initiative within the Networked Appliances Laboratory at Liverpool John Moores University. However the major limitation is that NASUF does not prove high-level marshalling, workflow management which affects the overall Quality of Service. NASUF does not provide any mechanisms to create personalised device configurations that transcend beyond localised networked environments.

3.3 Summary

There are a number of solutions that allow devices to be interconnected in the home environment. However, these solutions are very complex. Due to advancements in technology, new devices are much more complicated and need strong technical knowledge. Due to complexity in devices, it is very difficult for users to configure and use these devices. Research in the area of home networking and service-oriented

architecture has failed to produce convincing results for seamless integration between devices.

Service-oriented frameworks such as OSGi [Forum 2005], UPnP [Lee 2007], DPWS [DPWS 2006], DLNA [DLNA 2006], HAVi [HAVi 2004] and ePerSpace [ePerSpace 2005] are used for integrating home Networked Appliances. However, user need to configure these devices and in some solutions are managed via centralised providers. Services are usually discovered and composed using middleware protocols and interoperability issues are addressed using agreed standards but it is not clear if a single standard is capable of addressing all these issues. The solutions described in chapter 3 do not provide any mechanism to automatically discover and compose devices and services. Compositions are based on application based serialisation. Such services become more heterogeneous and managing such a framework will be more complex where the amount of control placed on device and service integration becomes more difficult. Different service providers use different communication standards and middleware, due to which interoperability becomes a main problem requiring more sophisticated solutions. There is a need for a new architecture to overcome the restrictive proprietary aspect of existing middleware. The solutions described in this chapter do not offer any mechanism for discovery and composition of devices and services. Some solutions require separate hardware adaptors for conversion of appliances into networked appliances, which is somewhat restrictive as distributed computing and service models are becoming more pervasive. As such devices and services are more heterogeneous; this makes management of such framework more complex. As technologies become more sophisticated, control placed on devices and service integration become more difficult.

This problem has been overcome, in part, using peer-to-peer (P2P) technologies whereby not only digital content can be distributed and discovered using global communications [Li 2008] but also enable devices to share and discover services provided by other devices in the P2P network. As such P2P networking is attracting a great deal of interest within a number of key industries as a possible solution for deploying services and overcoming the inherent centralisation problems associated with classic network configurations. Like home networking middleware, P2P also supports a number of techniques that have several advantages and disadvantages. For

example an earlier P2P technique such as Napster allowing content sharing relies on client-server technology, depending on a central server for content sharing. Failure of the central servers in effect disables search mechanism and content cannot be shared or discovered.

DHT-based P2P implementations adopted a more distributed model. Pure P2P models unlike Napster are difficult to control due to the absence of a central server. It is expensive to maintain DHT-based solutions because more time is spent updating indices. DHT-based solutions provide efficient data access but exponential cost as the number of peers joining and leaving the network increases. If a DHT approach is not used then computational costs are reduced however it required an exhaustive traversal of the network which causes network flooding. These solutions work well in structured networks where control can be placed on network topology as opposed to unstructured networks such as global P2P network where devices continually come and go. Using distributed computing model such as P2P and service-oriented architecture needs a new approach to be used to enable ad hoc services to be shared and discovered within global network without or with less human intervention. The new approach provides such as service discovery, service registration, service sharing and provides gateway services to make communication possible not only in same the network but also with other remote networks.

In Chapter 3 we discussed how Service-Oriented Architecture is a software architecture which allows different applications to exchange data with one another within business processes. We also presented the definition of gateways and discussed some notable research in gateways that can seamlessly internet connect devices, which allows other devices to share their own services and discover other shared services such as OSGi, UPnP, and DPWS etc. We mainly discuss these middleware in terms of their architecture, functionalities and their limitations. We found that these middleware solutions that discovery services are very limited as they are based on proprietary descriptions of how services must be advertised and discovered.

In the following chapter we present our requirements analysis based on our literature reviews discussed in chapter 2 and 3. We outline our main requirements to be implemented in order to achieve our research goals. We clearly set out which

requirements are important and which are optional. Based on these requirements we propose our framework for ad hoc gateway services.

CHAPTER 4

4 APPLIANCE GATEWAY SERVICES

4.1 Introduction

Chapter 2 and 3 described P2P technologies, gateways and some related research areas in these fields. It concluded that using the power of P2P technologies can provide an excellent medium for the service distribution and discovery for NAs. However, a number of challenges need to be addressed with service discovery and distribution in P2P environments. This chapter discusses the requirements for a system that allows Networked Appliances to advertise and discover in P2P environments. This chapter also includes the concepts and models developed to fulfil the requirements and address issues which are raised. These concepts provide a foundation to construct a framework which will satisfy all the objectives of this work.

4.2 Problems in Composing Networked Appliances Application

In order to explain our work better, we present an interesting scenario of an estate agency owning a number of houses in different parts of the country. Every house requires monthly maintenance such as utility meter reading. In some cases tenants might have problems with their home security system, home central heating system etc. In such situations the agent needs to send someone to visit and fix the problem. For example, if one tenant requires a change to the central heating settings e.g. increase or decrease heating at midnight but he may not have any access to the central heating controller. In this case, the agent may make an appointment for someone to visit the home at midnight. But if the maintenance person lives far from the tenant's house, he might not be able to pay a visit until the next day.

Instead of sending an engineer or visiting himself a preferable situation would be when the tenant requests a heating setting change the agent can remotely change the setting. In Figure 4.1, a home network consists of a number of appliances offering different services. The agent can setup a gateway, connecting all central heating appliances to the gateway i.e. *HeatingGateway.* Upon reception of the tenant request, the agent can gain remote access via his gateway. This gateway not only allows the agent to control the central heating but also helps in meter reading. Gateways should be flexible to add or remove appliances. In Figure 4.2 devices in the home make a peer network and can communicate with each other or may be connected as Client-Server. All the agents' houses connect with *HeatingGateway* which make a Gateway Peer Overlay Network along with other gateways connected in a P2P fashion. Using this gateway service the agents can not only manage their houses but also communicate with other gateways in the network. This gateway can also locate other services offered by other gateways and use them whenever needed. This allows the agent a flexible gateway, not only to manage some services but also to locate services on request.

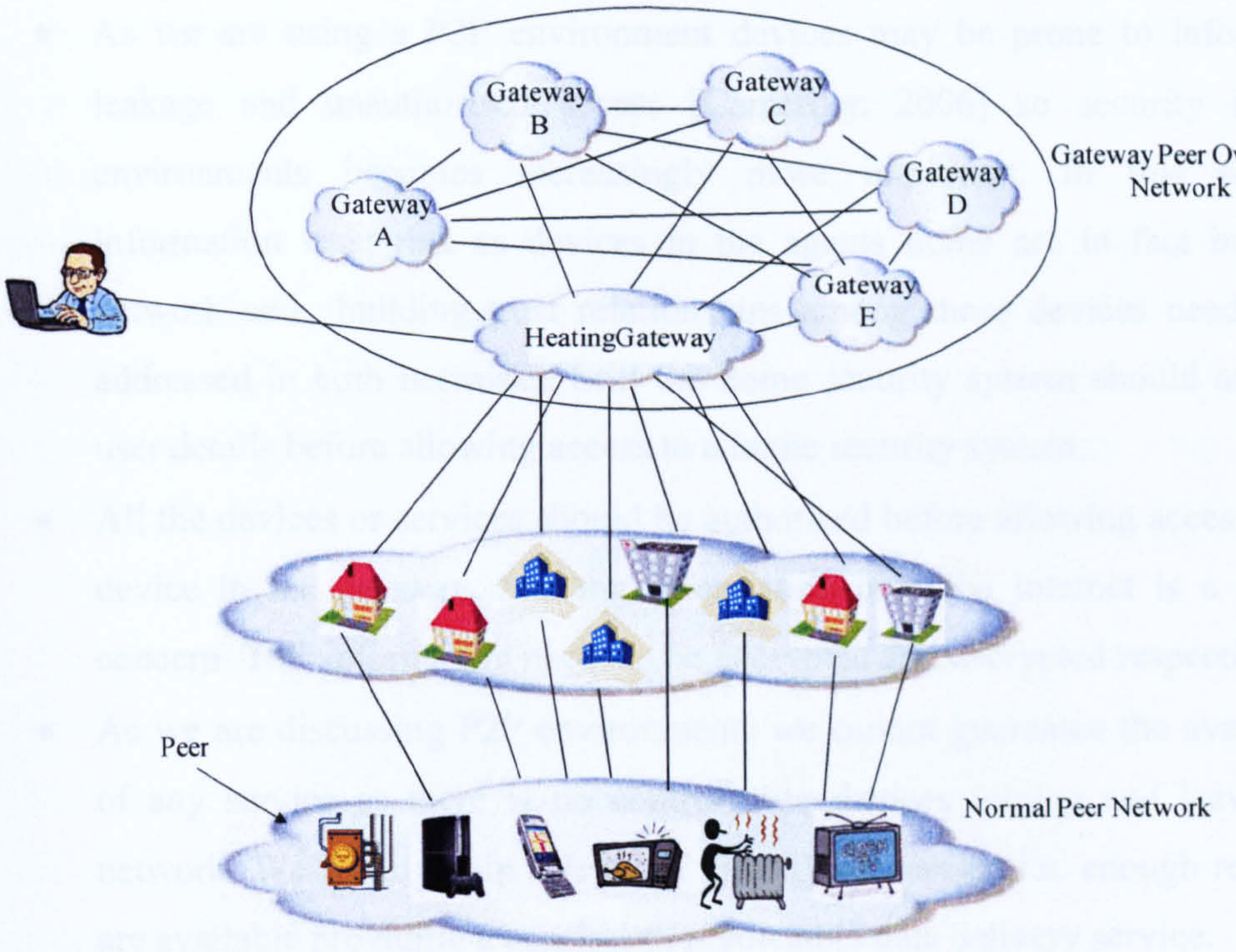

Figure 4.1 : Heating Gateway

In Figure 4.1 in order to implement this scenario in a P2P environment we can immediately identify some problems, which need to be addressed to implement this scenario:

- As we discussed earlier in this chapter and in previous chapters there are some disadvantages to centralised systems especially single point of failure. All devices should be decentralised and other devices need to locate and discover other services without knowing their locations, which improves robustness.
- In a P2P environment, the peers need to be uniquely identified due to the potential size of the P2P network. Every device in the network should be uniquely identified by giving a peer ID, which identifies peers in the system for location and discovery purposes.
- A number of devices operating within an office and home network may be running different operating systems i.e. a PC in the office might be running on Microsoft Windows™ while other device is running on Microsoft Mobile Windows™ or Symbian OS. There is a need to implement a system that can run on any machine. This allows different devices from different vendors to communicate, which improves interoperability.

- As we are using a P2P environment devices may be prone to information leakage and unauthorised access [Cameroon 2006] so security in such environments becomes increasingly more important. In this scenario, information is at risk as devices in the agents home are in fact in a P2P network and building trust relationships among these devices needs to be addressed in both networks. Still the home security system should authorise user details before allowing access to a home security system.
- All the devices or services should be authorised before allowing access to any device in the gateway. Sending information over the internet is a security concern. This information needs to be encrypted and decrypted respectively.
- As we are discussing P2P environments we cannot guarantee the availability of any service as there is no control over devices joining and leaving the network. We try to attain a level of Quality of Service i.e. enough resources are available providing a consistent, predictable data delivery service.

4.3 Discussion of Proposed Solution

In this section we discuss our system requirements on the basis of the prior literature study and challenges discussed earlier.

4.3.1 A Solution for Appliances Gateway Services

We design our framework using SOA due to the fact that distributed resources can be used by peers of the network. We design a framework which provides a gateway service allowing discovery and composition of Networked Appliances (NA's) in a P2P environment. Our framework not only provides this gateway service but also some other services discussed later in this section. The reason behind calling our framework *Ad Hoc* is because a gateway only exists when peers request it. As with P2P networking we cannot guarantee a particular service or device in the network. All operations carried out by our framework exist as a service in the network and many devices in the P2P network may be offering this service. In any situation or at any point in time devices offering this service may leave the network without notice and we have no control over other devices joining or leaving the P2P network.

In an ideal situation, all services exist in the P2P network but if one or more services do not exist our *ad hoc* gateway service framework will still work. Figure 4.3 shows our proposed framework. It consists of four sub-components on the basis of the functionality it provides. All these components may run on different devices offering these services. One of the reasons for implementing the framework is to understand how different services can be integrated together and to demonstrate flexibility for future changes depending on user requirements, as well as seamless integration of functionalities while remaining robust to one or more service failures.

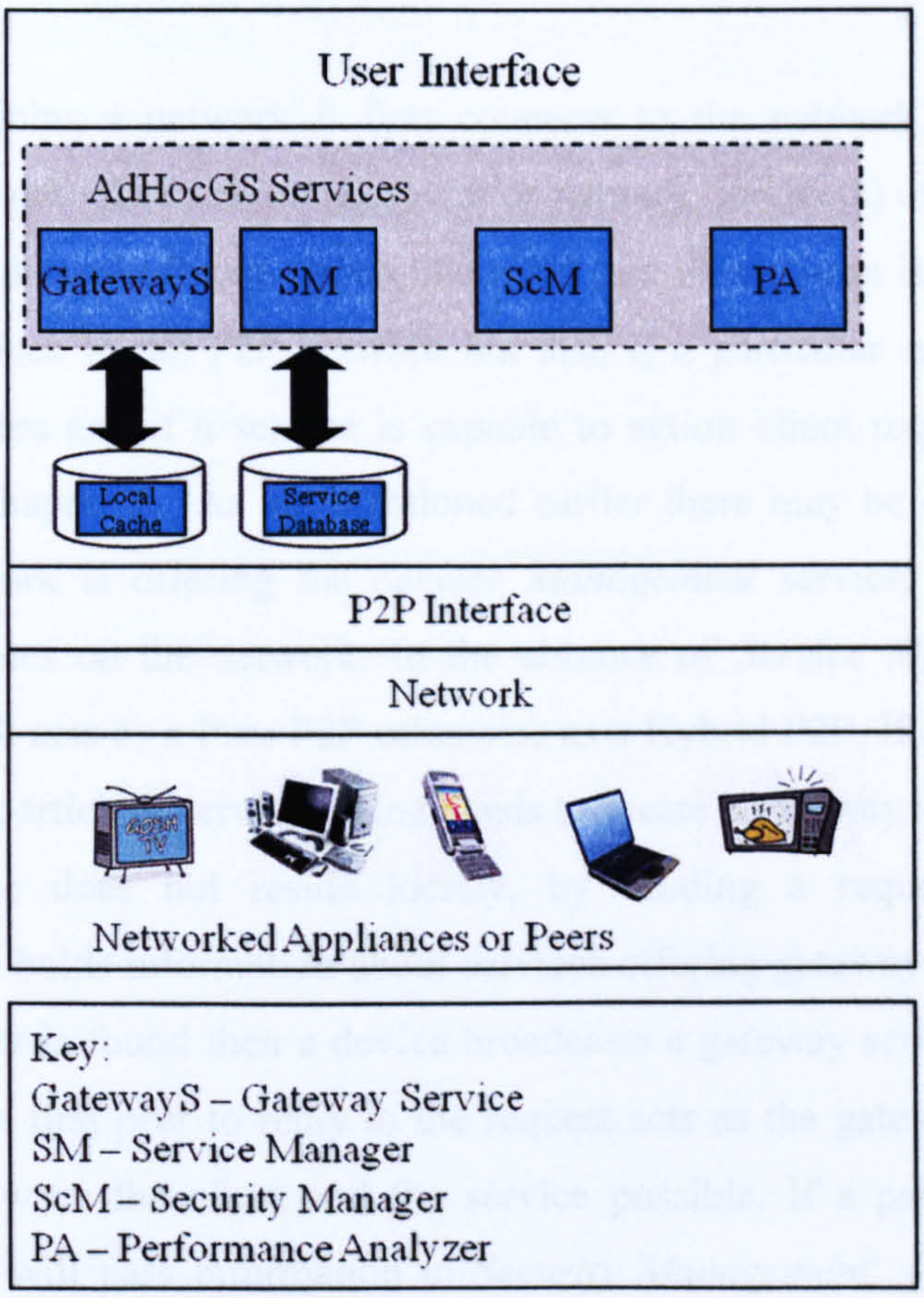

Figure 4.2 : Proposed Framework

In the following section, we briefly discussed all the components that constitute our proposed framework:

- *Client interface:* allows users to interact with the framework i.e. via service requests.

- *Service Management:* provides management services in the P2P network. A device may offer a number of services in the P2P network and needs to keep information about these services such as peer ID, service offered etc.
- *Performance Analyzer Management:* provides management resources in the P2P networks to carry out service requests by the user i.e. bandwidth, data rate etc.
- *Security Management:* provides security for the framework, such as authorisation and authentication.

When a peer joins a network it first connects to the network, and registers information such as peer ID, location (i.e. local or remote), service(s) offered, security constraints, and hardware/software capabilities. All this information is later used not only to locate services in the P2P network but also if a particular service requires security before access and if a service is capable to action client request details as discussed later in chapter 5. As we mentioned earlier there may be cases when no device in the network is offering the *Service Management* service, then a device broadcasts its services on the network. In the absence of *Service Management* the proposed framework acts as a Pure P2P otherwise as a Hybrid P2P. If any peer in the network requires a particular service it first needs to locate a gateway service, in case a requested service does not reside locally, by sending a request to *Service Management* which holds information about services offering gateway services. If no *Service Management* is found then a device broadcasts a gateway service request on the network and the first peer to reply to the request acts as the gateway and makes communication between the client and the service possible. If a particular service requires security it will pass information to *Security Management.* This process is discussed in detail in chapter 5 with the help of some UML diagrams.

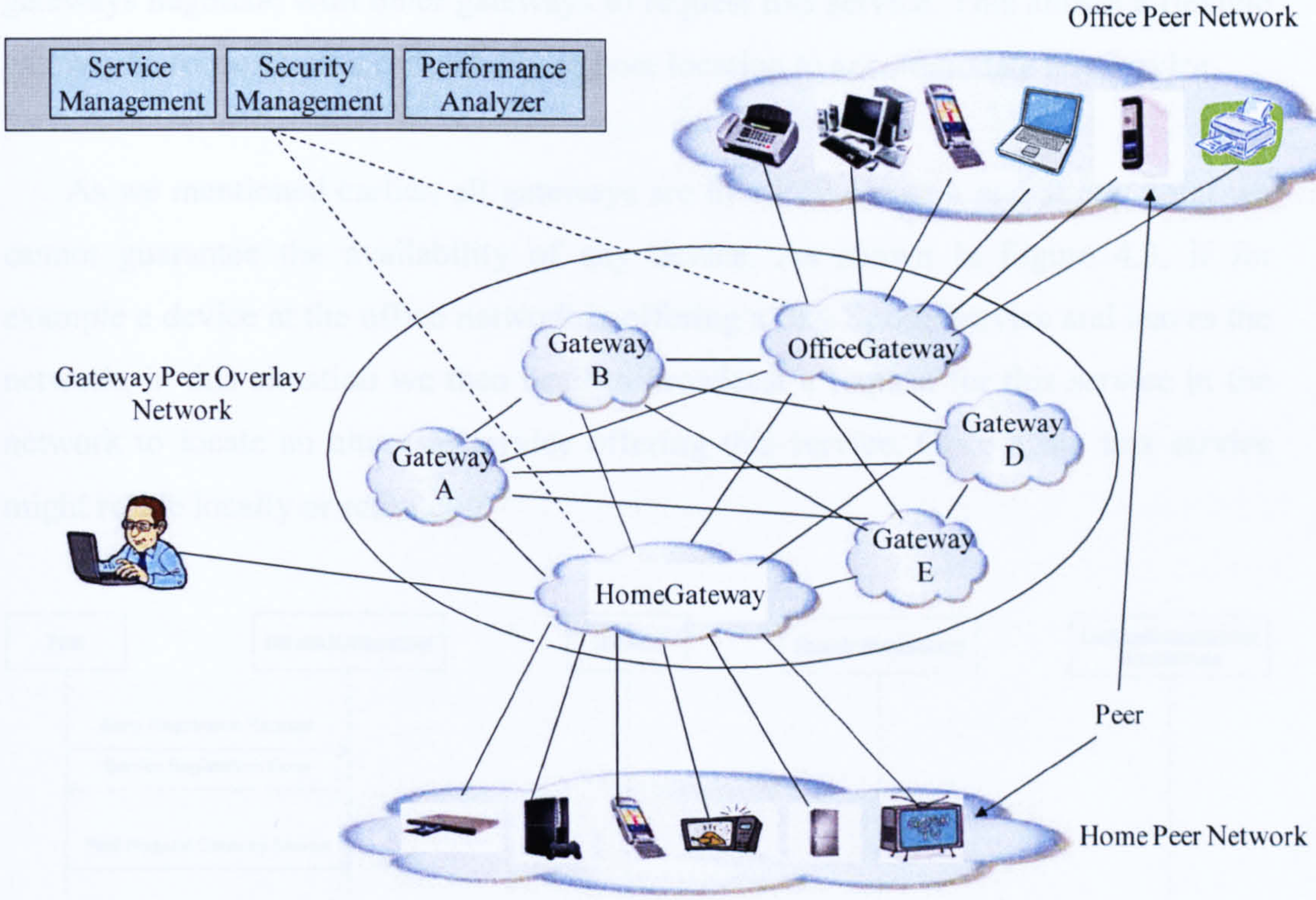

Figure 4.3 : Proposed Framework

In order to access these home Networked Appliances in a global network, these devices need to be connected in a more personalised way by combining them together via a gateway i.e. the *HomeGateway*. All devices in one gateway offer a set of services, such as service management, security management and performance analyser - these are discussed in more detail in later chapters. *HomeGateway* not only allows these services to transcend beyond the localised environment into the global environment but also communicate with other gateways in a Gateway Overlay Network and allow discovering services offered by other gateways in the overlay network. For example, if a device at a home network requests a TV channel e.g. Sky Sports, which is not offered by the home set-top box, the *HomeGateway* will broadcast a request to locate a gateway offering Sky Sport channel. A media centre at an office network connected with *OfficeGateway* might be offering this service is shown in Figure 4.3. Both *HomeGateway* and *OfficeGateway* create a P2P network with other gateways in an overlay network as discussed in the next section. *HomeGateway* can request this service from *OfficeGateway* by sending a request to start offering this service. The user is unaware of how the service is discovered, as

gateways negotiate with other gateways to request this service. This allows a flexible gateway to request services from any remote location to accommodate any service.

As we mentioned earlier, all gateways are in a P2P network and at any point we cannot guarantee the availability of any device. As shown in Figure 4.3, if for example a device at the office network is offering a Sky Sports service and leaves the network, in this situation we then need to broadcast a request for this service in the network to locate an alternate service offering this service. Once again this service might reside locally or remotely.

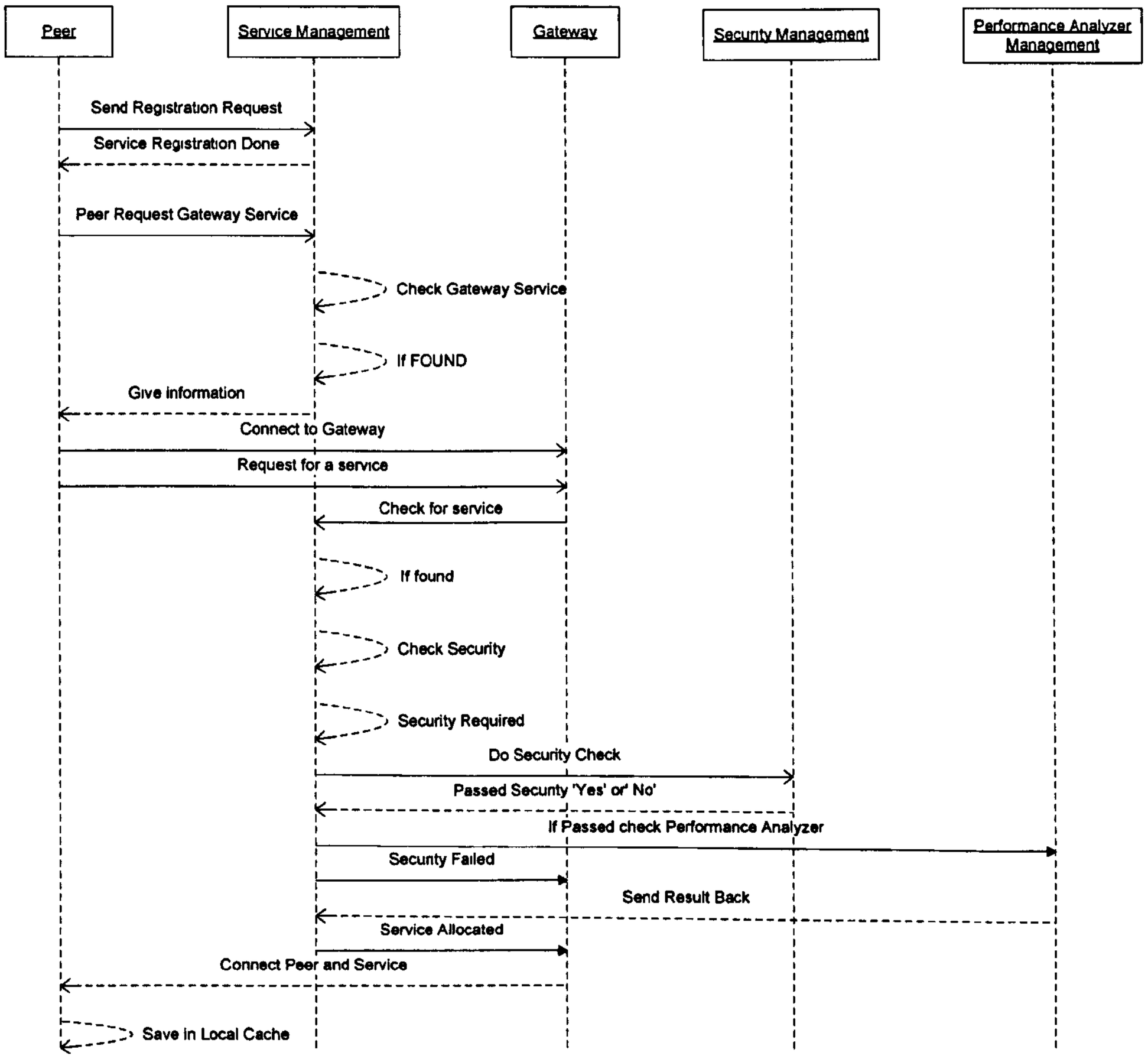

Figure 4.4 : Sequence Diagram for Proposed Framework

Figure 4.4 shows sequence diagram of our proposed framework. When a device first connects to the network, it needs to register its services with *Service Management.* Once it has registered its services, it allows peers to share its services

but also discover services offered by other devices. When a peer needs to locate other services in the network, it first needs to locate a *Gateway* service by sending request to *Service Management. Service Management* first checks its *Service Database* to locate gateway service. If found, information is passed to peer to connect to the gateway, otherwise broadcast request on the Gateway Peer Overlay Network. Once connected to the gateway, a peer then sends its request directly to the gateway. Gateway check services not only with the *Service Management* but also on the Gateway Peer Overlay Network. If any peer needs authentication, gateway can check with *Security Management.* Before allocating any service, gateway checks with *Performance Analyser Management* to check for resources availability. These components are discussed in details in rest of this chapter. Appendix-D contain complete Sequence Diagrams for AdHocGS Framework.

4.3.2 Our Overlay Network of Gateways

A P2P network forms a logical layer over the Internet called an *overlay,* the underlying physical connections between Internet nodes are not necessarily the actual structure of the P2P network. Routing mechanisms used by these peer systems utilise the Internet as a transport medium but may have their own routing protocols independent of or working over the Domain Name System (DNS). Using Internet as base transport and communication protocols the current P2P networks assist in communication across platforms and devices with different capabilities. Super-nodes are nodes in the P2P network that provide extra functionality than the rest of the nodes in the network. KaZaA file-sharing networks use super-nodes for assisting indexing of frequent requests to enable faster search; formation of an overlay network of super-nodes which function within KaZaA networks provide benefits to the entire system. Our concept of a gateway overlay network is inspired by the concept of super-nodes and overlay networks. These overlays to provide an extra functionality to the P2P system without changing the underlying layers. The main functionality of the overlay network proposed in this thesis is to act as a service-offering or service-sharing network. In our case, a gateway is not only offering services but also sharing services offered by other gateways. Our gateway not only allows users to compose different services in a gateway but also offering some core services. Our gateway can also request services offered by other gateways in the overlay network.

When managing services in the network, any system needs to address the issues of managing services, security and QoS. In a Client-Server all these functionalities are managed by some central server, in the P2P environment there is a need to manage them as distributed services. As the nodes in the overlay network would be acting as intermediate between peer network and other services able to log the details such as services id, service locations etc and also any service requested from other gateways. This information can be replicated on any other gateway available in the Gateway Peer Overlay Network at intervals and can act as backup gateway. For the purpose of this thesis *Gateway* is a peer acting as an intermediate between normal peer network and other gateways in gateway overlay network. Hence a Gateway peer is a type of super-node as it provides an additional functionality to peers who can only share and use services.

In Figure 4.5, Peers A-L and Peers R-V are members of the normal peer network, while peers M-Q are acting as Gateway Peer Overlay Network for the peers. As the state of the system is dynamic and constantly changing, it is possible later that this might grow considerably and one of the peers in P2P network becomes part of Gateway Peer Overlay Network to provide 'gateway' functionality to a group of peers who wish to share and offer ad hoc services. As mentioned earlier, the overlay network forms a logical layer on the top of the Internet utilising the Internet base transport protocols. In this context, our overlay network of gateway peers belongs to the same level as a peer but their functionality act as a logical layer above the normal peer network.

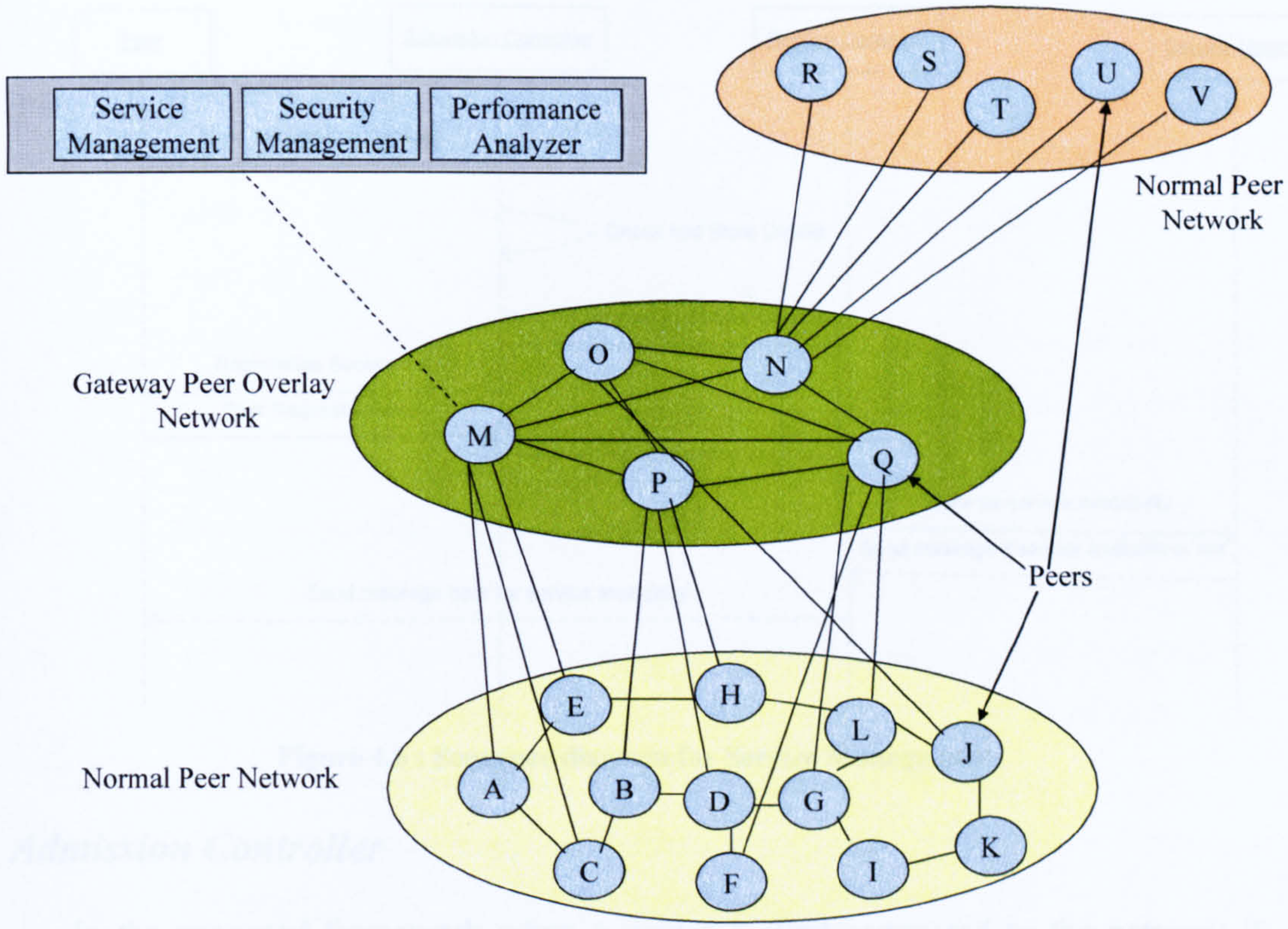

Figure 4.5 : Gateway Peer Overlay Network

4.3.3 Service Management Requirements

The proposed framework allows NAs to not only advertise services but also discover services offered by other NAs. There are a number of services running in a P2P network offering different services, located remotely or locally, different hardware/software capabilities, different security requirements etc. In order to manage these services we need Service Management. Service Management acts as a registry where other peers not only register services but also obtain information about other services. In order to manage services effectively we further divided our Service Management into three further components as shown in Figure 4.6. Appendix-D contain complete Sequence Diagrams of our AdHocGS Framework.

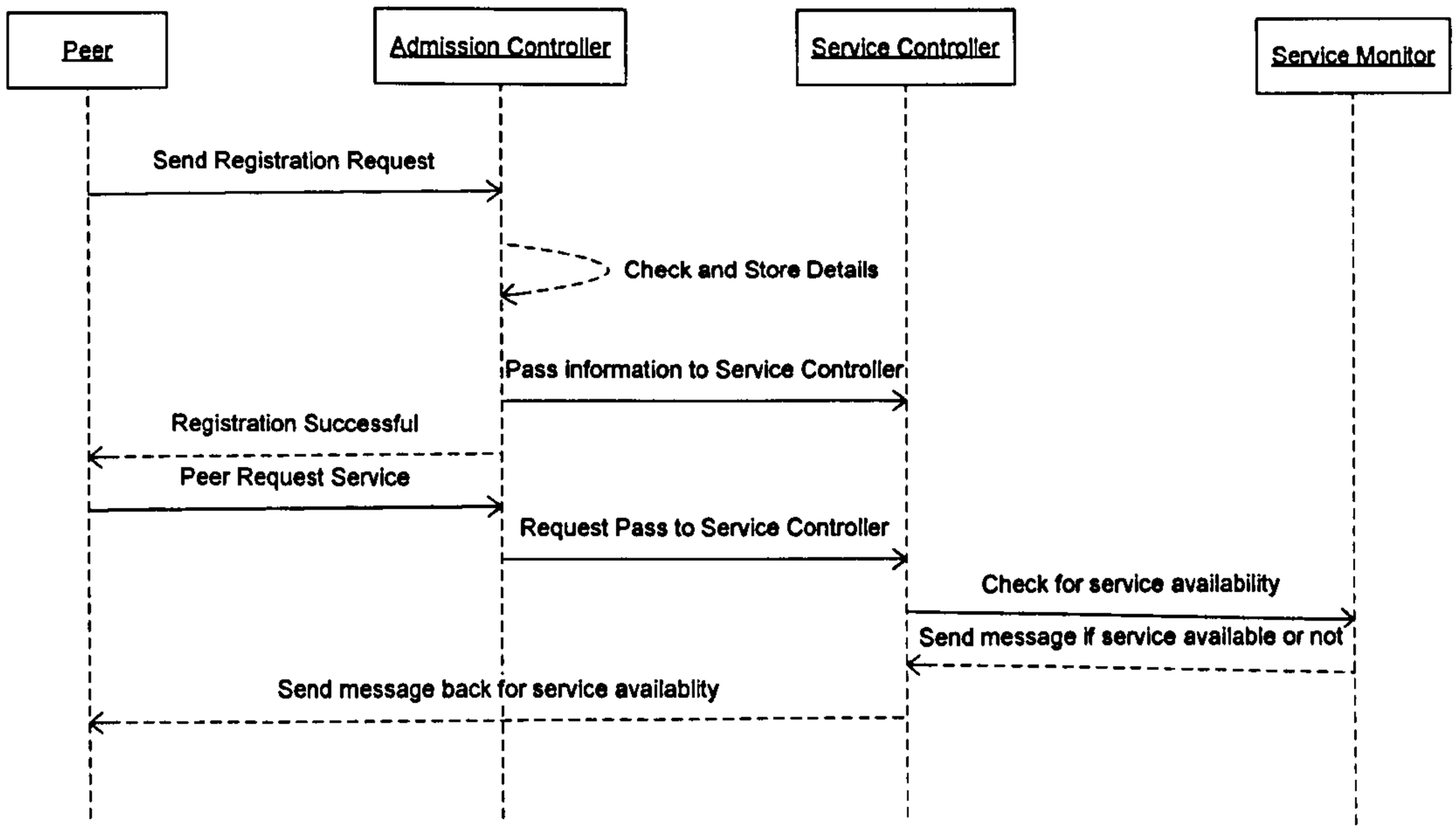

Figure 4.6 : Sequence diagram for Service Management

Admission Controller

In the proposed framework when a device is first connected to the network it needs to register its services. It is worth mentioning about bootstrapping [Mirko 2007], before any new peer joins an existing network it must obtain contact information of at least one node in existing P2P network. There are a number of methods to obtain information about bootstrapping nodes [Gauthier 2008; GauthierDickey 2008; Mirko 2007] i.e. Brute force, random probing, peer caches, obtaining hotlist server. The objective of bootstrapping node is to provide enough configuration information to a new peer so that it may join the network successfully and access other services. In our framework, new peer obtains bootstrapping node information from its local cache stored from previous session before leaving the overlay and try to connect to these peers. In service management Admission Controller is responsible for storing this information such as peer ID, name, location, software/hardware capabilities, if it is offering gateway service, or not, and security information.

- *Peer ID* is used to uniquely identify a peer in the network. At this point it is important to differentiate between User ID and Peer ID as most P2P systems allocate User ID i.e. username, so it is the user rather than peer uniquely identified in the system. It is Peer ID that identifies the peer in

the network for the purpose of discovery. Peer ID is allocated automatically by network i.e. router.

- *Location* is where the peer is relatively located. As we are discussing P2P network, a peer can be located locally or remotely. For example, requested peer might be located in the local network or in the remote network. Admission Controller marked services as "L", if peer resides locally or "R" if peer resides remotely i.e. offered by other gateway in Gateway Peer Overlay Network.
- *Software/hardware capabilities,* to know some more information about peer capabilities, such as memory size, and the software it is running. This information helps the Admission Controller to allocate the best available service if more than one peer is offering the same service. This information can also be used for device capability matching.
- *Offering gateway service,* to know if a particular device is also offering a gateway service.
- *Security,* if any service requires secure access i.e. need authorisation before accessing services or information sent and received should be encrypted etc.

Service Controller

Admission Controller only receives registration information. Service Controller keeps record of all the available services. Service Controller receives request from the network for a particular service. When Admission Controller receives all information at registration stage, it forwards this information to Service Controller, to store it in the service database. The reason behind keeping the Admission Controller and Service Controller separate is to share load in the network i.e. it is costly for Admission Controller to do registration as well as control services. Upon receiving a request for a service, Service Controller checks if the requested service is available in its database. If the requested service is offered by a number of peers, Service Controller checks with the Service Monitor for its availability. If the requested service is not available then it needs to broadcast a request on the Gateway Peer Overlay Network.

Service Monitor

Service Monitor actually controls services. When a service request is sent to Service Management, Service Controller keeps a record about the current status of the requested service such as *availability* if the requested service is available or when it will be available. In order to utilise available services more, every service is allocated for a specific time and can be renewed if no request is made for the service. Service Controller also checks if the requested service is available and if not it will broadcast it on the network.

4.3.4 Gateway Requirements

As discussed earlier we need a service which acts as a middleware between the peer and service. We called it a gateway as it acts as an intermediate peer between two different entities physically separated i.e. the requested service might be located somewhere remote. The main purpose of the proposed framework is to seamlessly interconnect devices independent of their locations. When a peer requests a service, Service Management checks if the service is available locally and if it exists information is passed to the requesting peer. In case the requested service does not exist in the local network, Service Management then needs to locate it outside of the network. Service Management checks the service databases for the peer(s) offering a gateway service and sends a request to the peer(s) to determine if they can still offer a gateway service. The first peer to reply to a Service Management request for gateway service is promoted to main gateway. This is discussed in further detail in chapter 5 and 6. When a peer wants to locate a service in the network it first needs to locate a gateway service. In the presence of Service Management, the peer sends a request to Service Management, which then locates gateway service from its database. In absence of Service Management the peer broadcasts an advertisement for a gateway service in the normal peer network. If Service Management finds that more than one peer offers a gateway service, the first service reply to the request acts as a main gateway service in First in First Out (FIFO). If a peer broadcasts a gateway service advertisement in the absence of Service Management, the first peer to respond to the request acts as a gateway for that peer.

As discussed earlier one novel contribution of our proposed framework is not only to offer gateway services but also to rediscover an alternative gateway service in case of failure, with minimal user intervention. Solutions discussed in chapter 3 do not address this issue. When a gateway fails, services connected to it fail as well. In order to overcome this issue instead of running one gateway in the P2P network, we run more than one gateway service. Out of these gateway services, one acts as the main gateway, or active gateway, while others act as passive or backup gateways. Backup gateways replicate the active gateway at regular intervals. Peers have a list of all available gateways in the network i.e. active and backup gateways. Service Management shares information about peers offering a gateway service such as Peer ID, and location. Figure 4.7 provides an overview of this process with one active gateway and two backup gateways running at the same time i.e. backup gateways copying all the information from the active gateway.

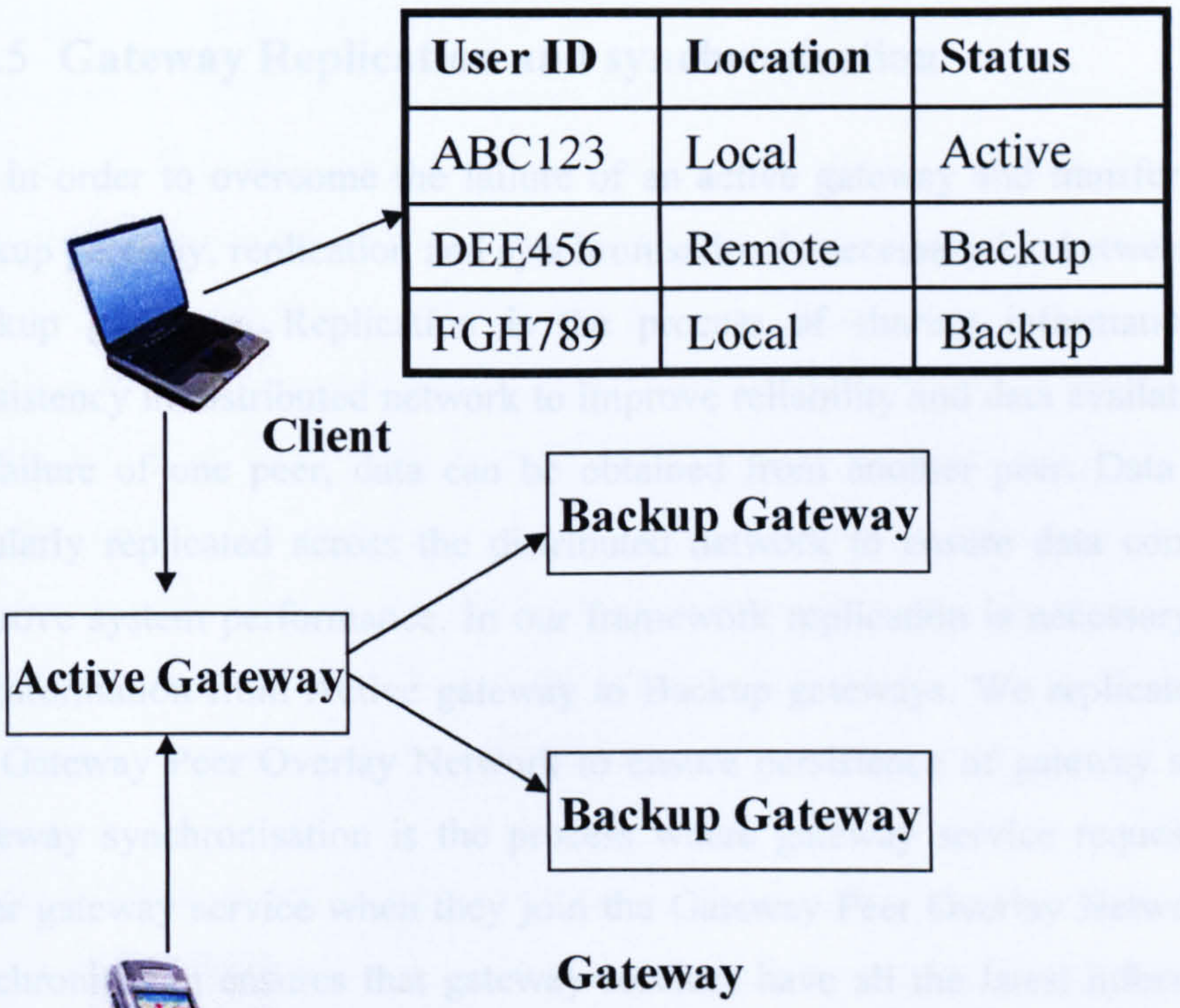

Figure 4.7 : Gateway Service

In case of failure of the main active gateway, the next backup gateway switches to active gateway and starts communication where the active gateway has failed. There is no pre-defined procedure for how gateways are ordered; it is done as it arrives, i.e. when Service Management or peer sent the request for gateway service. As discussed we cannot guarantee any service availability at anytime as a peer may leave the network without any notice. If the next backup gateway is not available, the second available backup gateway becomes the active gateway. Service Manager keeps checking if the backup gateway is still available by sending control messages i.e. Ping. This is explained in more detail in chapter 6. A selection of a number of backup gateways may vary depending on services available in the network. But at any moment in time, we need to keep at least two backup gateways. When a peer switches to the next backup gateway, a request is also sent to Service Management for a gateway service to act as a backup gateway. By using backup gateways in the network, if a particular gateway fails it would not result in loss of all services or data.

4.3.5 Gateway Replication and synchronisation

In order to overcome the failure of an active gateway and transfer control to a backup gateway, replication and synchronisation is necessary i.e. between Active and Backup gateways. Replication is the process of sharing information to ensure consistency in distributed network to improve reliability and data availability. In case of failure of one peer, data can be obtained from another peer. Data needs to be regularly replicated across the distributed network to ensure data consistency and improve system performance. In our framework replication is necessary to replicate all information from Active gateway to Backup gateways. We replicate data across the Gateway Peer Overlay Network to ensure persistence of gateway services data. Gateway synchronisation is the process where gateway service requests data from other gateway service when they join the Gateway Peer Overlay Network. Gateway synchronisation ensures that gateway services have all the latest information in the Gateway Peer Overlay Network. Gateway synchronisation reduces broadcasting service requests over the Gateway Peer Overlay Network. Data replication overcomes the failure of a single gateway i.e. in case of failure of an Active gateway, any Backup gateway promoted to an Active gateway communication may be restarted from the

point where last replication took place or in case of live streaming will resume on new gateway discovery.

4.3.6 Device Capability Management

As the number and variety of devices connected to the Internet grows there is an increase in need to locate the best possible device that is capable of meeting user service requests. A CC/PP (Composite Capabilities/Preferences Profile) profile is a description of device capabilities and user preferences that can be used to guide the adaptation of content presented to that device [W3C 2007]. A CC/PP profile contains number of CC/PP attribute names and associated values that are used to determine the most appropriate resource to deliver to a client. It allows a client to describe its capabilities by reference to a standard profile, accessible to an origin server. The CC/PP profile describes 3 major components (TerminalHardware, TerminalSoftware and TerminalBrowser) of the client such as hardware platform on which software is executing, the software platform on which all applications are hosted and an application. A CC/PP profile describes client capabilities and preferences in terms of a number of "CC/PP attributes" for each component. The description of each component is a sub-tree whose branches are the capabilities or preferences associated with that component. A capability can often be described using a small number of CC/PP attributes, each having a simple, atomic value. Where more complex values are needed, these can be constructed as RDF [Manola 2004] subgraphs. One useful case for complex attribute values is to represent alternative values; e.g. a browser may support multiple versions of HTML. For example, TerminalHardware can be further subdivided into width and height related to video output. TerminalSoftware can be subdivided into name, version and vendor i.e. Internet Explorer, 5.0 and Microsoft.

In a P2P network, a number of peers may be offering the same service. When Service Controller receives a request for a particular service, it needs to search for the best possible service. As we discussed earlier that on registration along with other information, a device also registers their software/hardware capabilities with Admission Controller. Our proposed framework only captures basic information such as memory, CPU speed and screen resolution. Each device publishes these capabilities in case no service management exists. This allows other peers to first determine if it

can effectively execute requested services. This feature can only be implemented on specialised Networked Appliances because simple Networked Appliances [Merabti 2008] do not offer this service. When a peer requests a particular service, Service Controller checks for the best possible service that can execute the peer's request.

When Service Management receives a request for a service from a peer it may result in several services offering the same functionality, which make it difficult for a user to select the best service. For example, a computer monitor or HD TV might be offering video service. It may be possible to stream video on the computer monitor but the best solution is HD TV. In another situation, when a device requests a video service, HD TV might be not available in the network and the computer monitor is the best available service but once HD TV becomes available, the video could be streamed to a new service. In our framework, we developed a mechanism of Device Capability Matcher that allows peers to automatically determine the best device to execute services [Merabti 2008; Muhammad 2007]. In the above case, Service Management matches the device capability requirements using Device Capability Matcher to find the best video service to stream the video data. Our framework searches for services within the network, locates available services and uses the best one available. We designed an algorithm for Device Capability Management as shown in Figure 4.8. When a device sends a request for a particular service to the gateway along with capability requirements, the gateway service checks with the Service Management for a service availability, along with device capability requirement (Figure B.11, Appendix B). Upon discovery of services offering video service, gateway service then checks with Device Capability Matcher which extracts device information such as hardware/software capabilities to check whether the device can execute the requested functionality. On the basis of these capabilities Device Capability Matcher then chooses the best device out of the available ones.

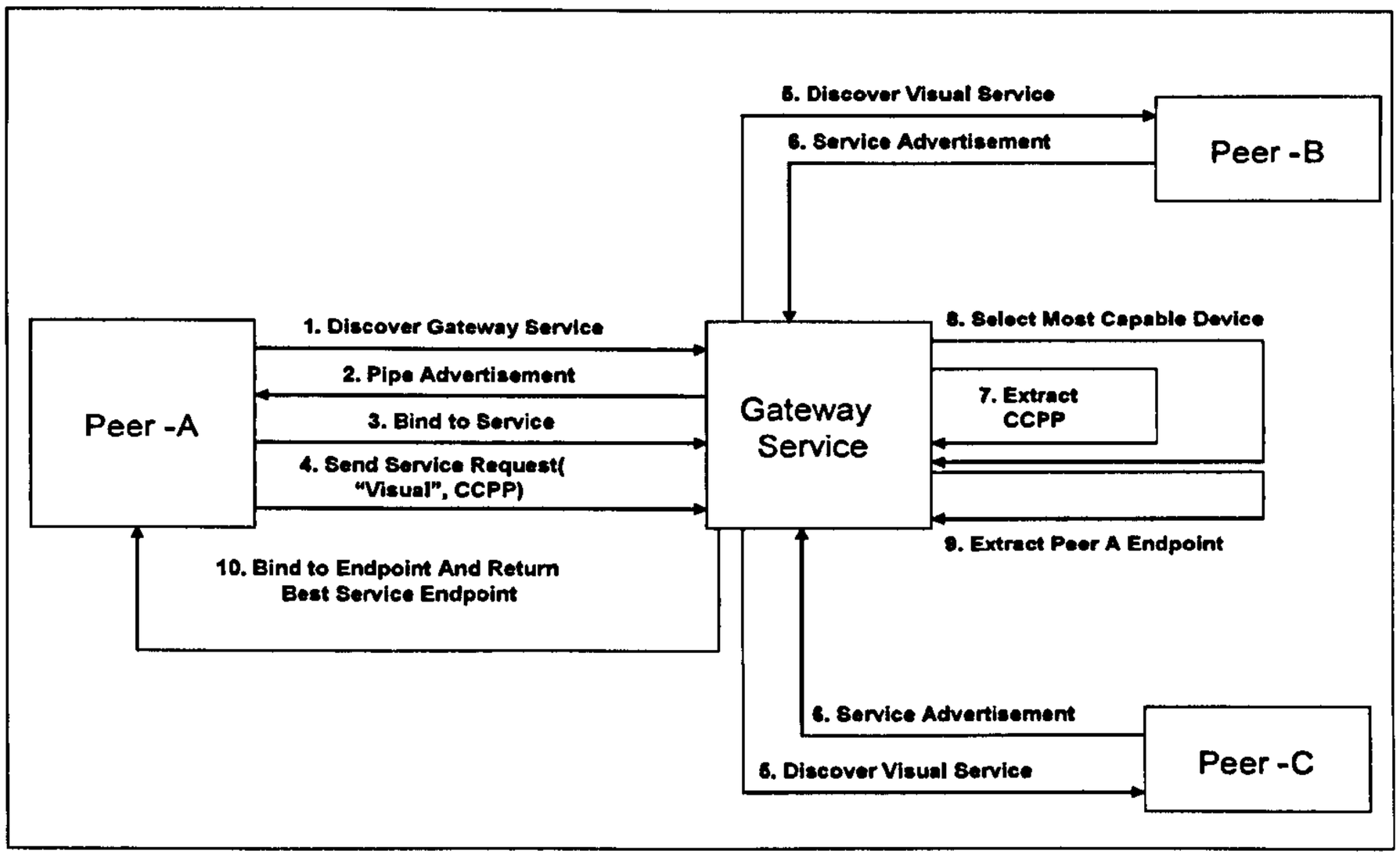

Figure 4.8: Device Capability Matching Algorithm

4.4 Design challenges

After investigating the system requirements, we have established some design challenges for the proposed framework. These challenges are listed below and need to be addressed for implementing our proposed framework.

4.4.1 Naming and Addressing

Since the location of the physical device may continually change, we need to develop mechanisms that support unique naming and addressing functions. All devices working in the network should be distinguished from the rest of the devices in the network, i.e. we need to assign a globally unique ID to each device so they are distinguished from the rest of the devices in the network. It should also make it possible to search for devices for particular capabilities and help to identify those that posses such capabilities. It also helps not to restrict NAs within a particular domain such as Home or Office but mechanisms should be support to access to devices if they cross different domains. We need mechanisms that allocate unique address to devices joining P2P network which allow accessing these peer remotely.

4.4.2 Platform Independence

As there are a number of manufactures creating electronic devices, it may not be possible for Networked Appliances [Moyer 2002] to know about the characteristics of

a target device; therefore it is important that the implementation should be platform independent. Platform independence is also referred to as cross-platform or multi-platform. The idea behind platform independence is to run software built for one platform on different machines with different software and hardware specifications. For example, programs written in Java [Arnold 2005] can run on machines running Microsoft Windows™ Operating System (OS) [Wolf 2007] or Linux [Linux 2007]. We need to implement a system through which devices from the various vendors can communicate i.e. enable interoperability.

4.4.3 Decentralisation

With a central server all the devices controlled by such a single master controller within network benefits cost and security. But as we are using P2P network devices we do not know about the location of other devices, as there is no centralised control within the network. Therefore ad hoc service discovery mechanisms need to be developed that do not necessarily know the service interface *a priori*.

One of the main drawbacks in centralisation is the single point of failure i.e. central machine can bring down the whole network to a failure. We also mentioned at the beginning of this chapter that current solutions offered central gateway services which means in case of failure of a gateway all communications among devices will be lost. We are implementing our gateways as decentralised services in the network, which overcomes failure of a single gateway and also in case of the failure of first gateway will rediscover an alternative gateway with little or no loss of communication (details in next chapter).

4.4.4 Device Capability Matching

Device Capability Matching is needed in order to check the hardware and software capabilities of a device, which are used to determine how effectively the device can execute the services it requests. Typical parameters could be screen resolution, available memory and the software the device has installed. In a P2P network it may be possible that one service could be offered by a number of devices but with different capabilities, for example, different devices may be offering video capabilities such as computer monitor and television. When a device first searches for

a video service a number of peers offering these services may be located with different parameters. Service Management will select the best available service but will still keep track of the other devices offering the same service with best capabilities to run the requested video.

In addition to the basic requirements listed above, other requirements – as listed below – are beneficial in order to implement an ideal system. We address these requirements in our design.

4.4.5 Security

As with all computer-based systems, security is a major issue. Devices are exposed to information leakage, unauthorised access, eavesdropping and message tampering. Mechanisms are therefore needed to provide an efficient security model. Also from the user point of view security is one of the major concerns in such systems. Users want to make sure that no unauthorised access to their services or devices occurs. Security should work on two levels: one when someone accesses the service to ensure only authorised access to the service. Second, information exchange between the devices requires confidentiality and can be achieved by encrypting and decrypting data respectively.

4.4.6 Quality of Services (QoS)

QoS refers to the set of parameters and mechanism used to specify and guarantee the performance of the network [Chauvet 2004; Gmach 2008], including the transit delay expected to deliver data packets to the destination device, device and service protection such as unauthorized access, the cost associated with requests, error management, and priority mandating. Before carrying out any operation within the network we need to make sure whether it is going to be feasible or not. For example, if we need to exchange video data which requires more bandwidth, QoS makes sure that enough bandwidth is available and if not, postpones the request until enough resources available.

4.4.7 Trust Relationship

Trust is arguably the most important bond between human beings. We are aware of the importance of trust [Almenarez 2008]. In decentralisation trust is one of most important parameters because every person has his own point of view. In a decentralised network such as P2P, peers can see where the information is coming from. When peers request information from a particular source it is based on trust because there is not a single point of authorisation. On the basis of trust, peers create a network to exchange information, which create decentralised, personalised "webs of Trust". In the P2P network, peer groups and peer memberships are dynamic and typically they do not implement centralised security mechanisms, therefore a level of trust cannot initially be determined [Chen 2005; Runfang 2007]. Trust relationships need to be implemented in P2P networks, which enable peers to trust each other including transitive trust, for example, peer A trusts peer B, peer B trusts peer C, so peer A trusts peer C.

4.5 An Ad Hoc Gateway Services for Accessing Networked Appliances

On the basis of the above discussion we propose the implementation of a framework which provides platform independence and enables devices from different vendors to communicate via this gateway. Our framework is called An Ad Hoc Gateway Service (AdHocGS) Framework, mainly because it provides a gateway service allowing discovery and composition of Networked Appliances (NA's) in P2P environment. Our gateway service also supports security and performance in regard to QoS and Device Capability Matching. It also provides the mechanism to rediscover alternative gateway services in case of failure of the active gateway.

In the following section we discuss the core components of our system on the basis of above discussion. We group components together according to their functionality as shown in Figure 4.9. It enables us to better explain our framework and also helps us to design and implement a prototype. This will also help us or other researchers to extend this framework in future.

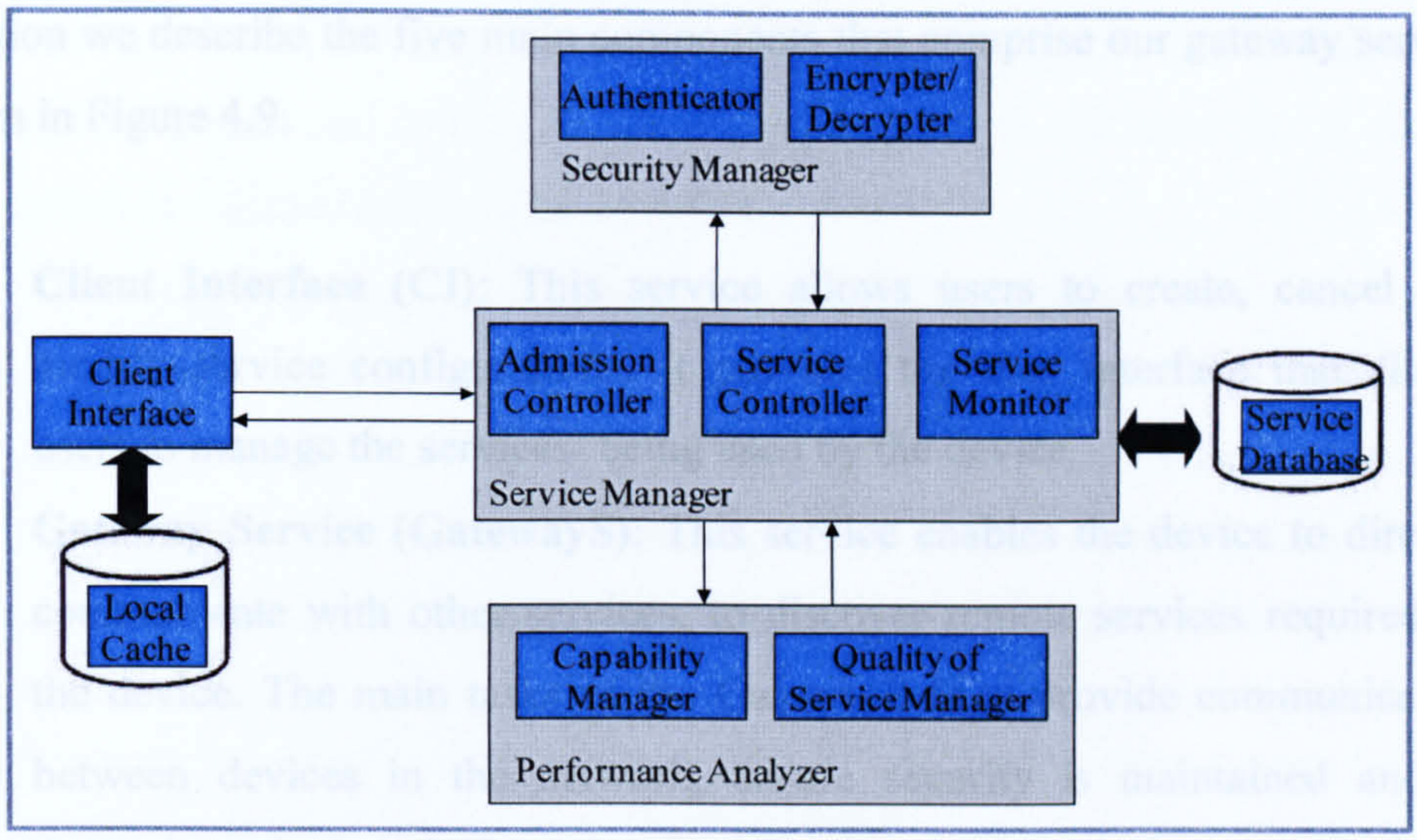

Figure 4.9 : Ad Hoc Gateway Service Framework

In our system we ensure the availability of gateway services in distributed P2P networks. When a device first connects to the network, it locates a Service Manager to register its services. Service Manager stores all the information related to the peer such as peer name, and service(s) offered. It also records information about devices that offer gateway services. When device requests for a particular service, Service Manager first checks if it is available locally, if not it needs to locate it outside the local network. First it needs to locate a gateway service which enables it to locate a service outside the local network i.e. globally. Devices discover a gateway service by discovering the gateway advertisement – an advertisement is an XML message that describes the fundamental metadata associated with specific features of services, such as endpoint binding information, Quality of Service parameters and service capability descriptions. Any device in the network that offers a gateway service may respond to the gateway service request, allowing the device access to the gateway, which enables it to discover personalised services within the network. The gateway service itself may compose of individual services which may either reside locally on the device itself or be provided remotely by other devices within the network, allow the gateway to perform security management, Quality of Service, device capability matching and service discovery. When all the required core services are discovered and bound together the gateway service becomes available to be used. If the gateway service fails then all the core services it offers fail as well. However if one or more of the core service used within the gateway service fail then only the failed service will be lost and as a result an alternative service will need to be discovered. In the remainder of

this section we describe the five main components that comprise our gateway service as shown in Figure 4.9.

- **Client Interface (CI)**: This service allows users to create, cancel and modify service configurations. It provides the user interface that allows users to manage the services' being used by the device.
- **Gateway Service (GatewayS)**: This service enables the device to directly communicate with other services, to discover remote services required by the device. The main tasks of the GatewayS is to provide communication between devices in the network, ensure security is maintained and to determine whether devices are capable of executing services based on the hardware and software capabilities it supports. A gateway communicates with other gateways in the overlay network.
- **Security Manager (ScM):** This service ensures only authorised devices access the services provided by devices. When devices find candidate services, the security manager authorises, authenticates, and encrypts and or decrypts data transferred between devices.
- **Service Manager (SM)**: This service is responsible for the management of services. It contains the list of the services available locally and remotely in order to complete the tasks required of the device. It periodically maintains the service dependencies, manages service bindings at all times and ensures that the time-to-live (TTL) values associated with service advertisements are current.
- **Performance Analyzer (PA):** This service is responsible for checking the hardware, software and network capabilities of the device. For example, the PA may check the performance of the network before binding the service to the device e.g. the available bandwidth. It will also ensure that the device has enough resources available to carry out operations, such as processing power, screen real-estate and the required software drivers such as video codecs.

Some core services are essential for all peers participating in the P2P network, which enable communication between peers, discovery and routing.

- Advertisement Service (AdvertS): This service allows peers to advertise their services and can be located by other peers in the network. It allows peers to register their service with SM.
- Discovery Service (DiscoveryS): This service enables peers to locate available services in the P2P network.
- Lookup Service (LookupS): This service enables communication over a network; the peer can use a lookup protocol to get information on remote machines and establish communication using this information.

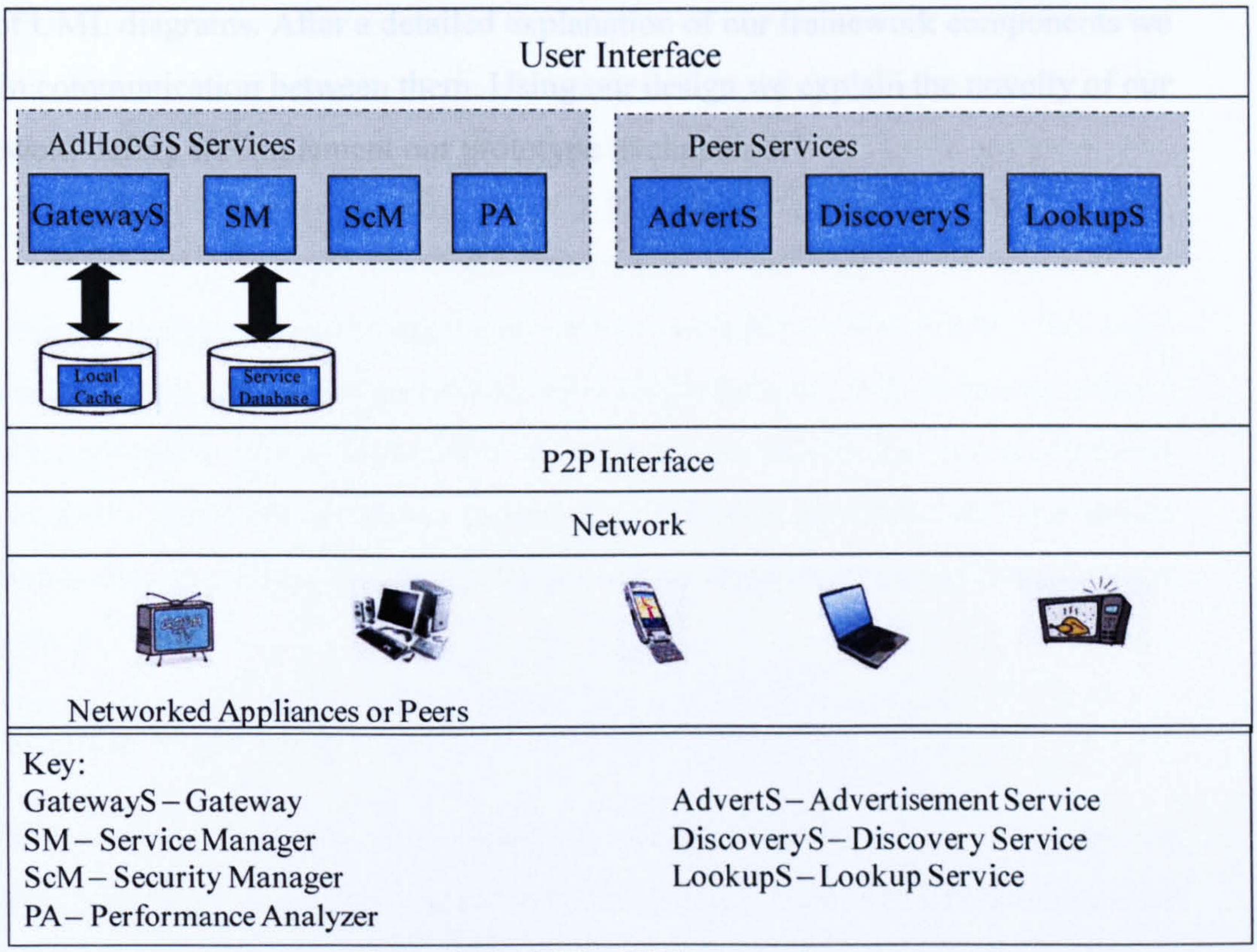

Figure 4.10 : AdHocGS Framework

4.6 Summary

Chapter 4 introduces our Ad Hoc Gateway Service Framework. In the beginning we discussed the novelty of our research based on the results of chapter 2 and 3. We conclude from our background chapter the limitations within current middleware solutions that we not only require ad hoc gateways which enable services to be advertised and discovered within a global network but also to provide an alternative gateway service in case of failure. We also conclude that in current middleware

solutions failure of particular services in composition, results in failure of the whole composition. We also provide an alternative service in case of failure of any service. We presented a scenario to explain our idea and design a prototype presented in chapter 6.

In the next chapter we discuss our AdHocGS Framework components in more detail and explain how they communicate with each other. The various design issues have been addressed and how this impacts the overall design process will be discussed. We discuss the main components of our framework in more detail with help of UML diagrams. After a detailed explanation of our framework components we explain communication between them. Using our design we explain the novelty of our framework before we implement our prototype in chapter 6.

CHAPTER 5

5 ADHOCGS: A FRAMEWORK FOR GATEWAY SERVICES FOR ACCESSING NETWORKED APPLIANCES

The previous chapter highlighted the requirements for the overall research and provided high level specifications for an ad hoc gateway service framework. This chapter discusses the main components in more detail and how they communicate with each other. The various design issues that have been addressed and how they impact the overall design process will be discussed. In this chapter, we start with an overview of design considerations and system modelling. We discuss the main components of our framework in more detail with the help of UML diagrams. After a detailed explanation of our framework components we explain the communications between them. Using our design we explain the novelty of our framework introduced in previous chapters. Using this design we present an implementation of our prototype in chapter 6.

5.1 System Modelling

This section discusses the modelling methodology used to address the requirements and system behaviour for our framework. Important issues to consider before designing a framework are:

- Design a framework with maximum flexibility and extensibility in mind.
- To ensure any changes made to the framework do not heavily impact on other applications that are based on the framework.

As all P2P networks are ad hoc in nature, the topology of the network is very dynamic, these networks are scalable and utilise the resources of the other peers to accomplish tasks. In order to fully utilise this environment, we chose to use Service Oriented Architecture (SOA). In P2P, a SOA would be slightly different in that the

location and discovery procedures will not be centralised whereas the concept of SOA would still be applicable. Services still have to be registered and discovered in order to be utilised. Service registration would be done via advertising the services to the other peers in the network to keep the network truly decentralised as opposed to the traditional centralised method.

We have used UML (Unified Modelling Language) to illustrate our framework in more detail. UML is a standard general purpose modelling language for object-oriented software using graphical notations such as activity diagrams, sequence diagrams [Booch 2005]. The system requirements were captured with the help of UML use case diagrams. This allowed us to define the roles of the participants in the system. Our design specification describes the system requirements in the form of a Use Case Model (Appendix A), mainly used to obtain the functional requirements. The Class Diagrams (Appendix B) describes the data which is used to support the entities in the system. The Activity Diagrams (Appendix C) and Sequence Diagrams (Appendix D) which captures the behaviour of the components and illustrates communication between them.

In our framework the basic functionality of a generic peer is identified as being able to:

- Join P2P network
- Advertise service
- Discover service
- Make a query
- Accept a query
- Respond to a query, and
- Leave the network

The functionality of the peer is included within the peer interface to the P2P network. These operations are encapsulated within a P2P interface and considered a core requirement for any peer to function in our framework.

5.2 AdHocGS Framework

On the basis of the discussion in our requirement analysis, we need a middleware i.e. software framework, through which NAs can communicate with each other. One of the main goals is to design a framework to provide a solution for devices to communicate in a P2P network. We designed a framework *Ad Hoc Gateway Service (AdHocGS)* which enable NAs to advertise and discover services in a P2P environment. Our framework is a grouping of a number of components serving specific tasks. These components can not only work as a group but also perform individual tasks. These components are not totally dependent on each other i.e. absence or failure of one component won't affect the main task of our framework i.e. gateway service. Nonetheless, the fact the framework incorporates functionalities needed for an ideal system means that services will integrate themselves together tightly when they are available. Later in this chapter we discuss components of our framework and how they are designed to work together seamlessly. Our reason for designing components is to allow flexibility for future changes i.e. one component can be redesigned and implemented to improve the performance of our framework. One of the major strengths of the framework is seamless integration of functionalities while remaining robust to individual service failure.

For better explanation, we are going to discuss the requirements mentioned in our scenario. First we need a mechanism which allocates every peer in the network a unique peer ID. Using this unique peer ID, a peer can not only advertise itself but also allows other peers to discover it. This allows peers not to worry about the underlying routing mechanism i.e. discovery of the peer in the P2P network. Secondly, the framework should address the issue of platform independence since different vendor devices may be participating in the P2P network which requires interoperability.

Thirdly, decentralisation in the proposed framework removes reliance on a central server, i.e. which overcomes the problem of single point of failure. Our proposed framework can be implemented as either Pure P2P or hybrid P2P, which we discuss in later chapters. One of the main issues in P2P network is security, as devices are operating in an open network with no central control. Devices are prone to information leakage and unauthorised access, so the framework addresses this issue,

i.e. to authorise peers in the network and make sure data moving in or out of the network should remain safe. In P2P network, as services are limited we need to ensure enough services are available to carry out operations.

On the basis of the above system requirement, we need to implement a decentralised gateway service, which provides platform independence and enables devices from different vendors to communicate through it. Such a gateway service should also support security and performance in regard to QoS and Device Capability Matching. Finally, it should also provide a mechanism to rediscover alternative gateway services in case of failure of the active gateway. In the following sections we discuss the different components of the framework developed using UML diagrams.

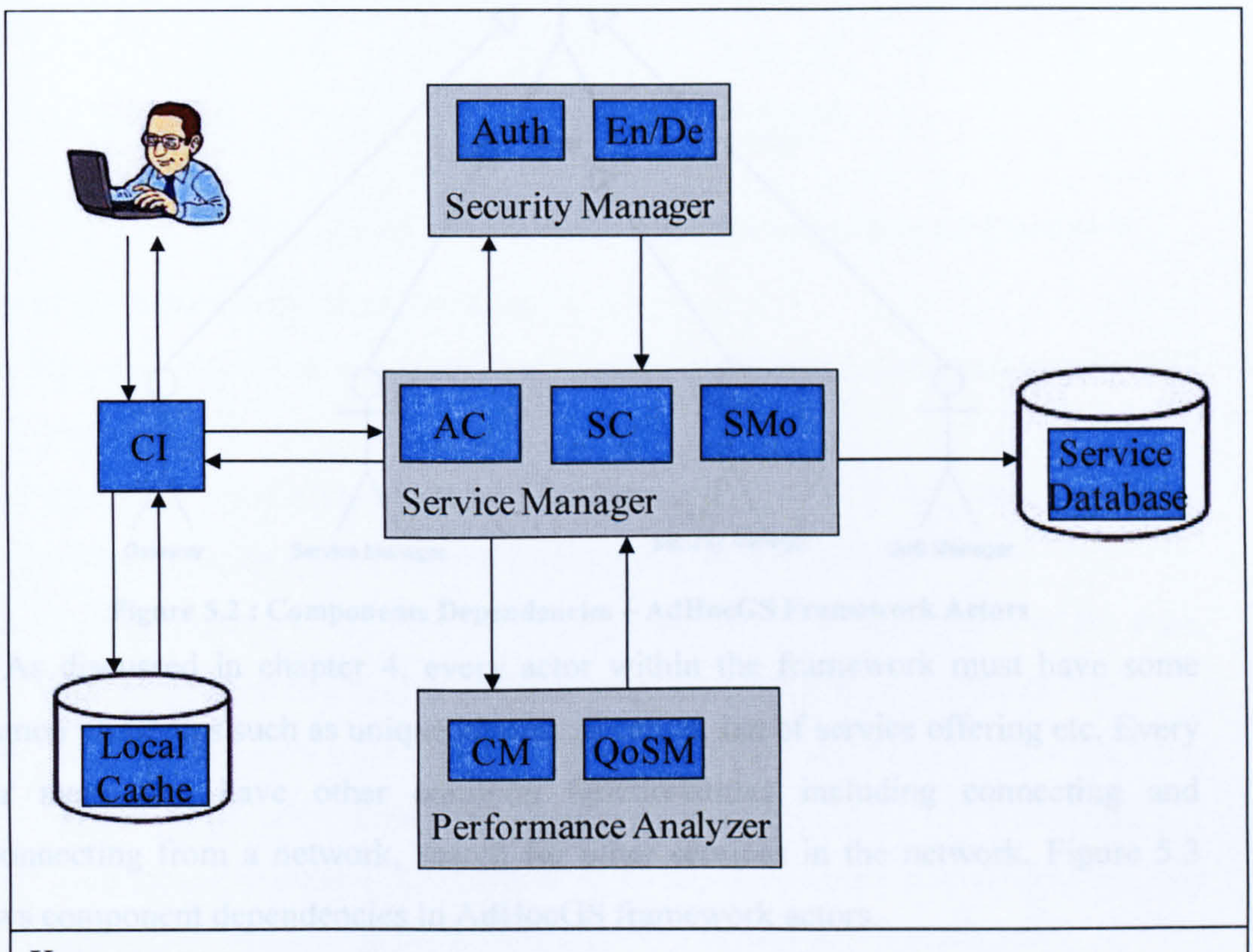

Figure 5.1: AdHocGS Framework

5.3 System Actors

We first identify the key roles of the objects in the system i.e. different roles of peers in the system as shown in Figure 5.2. A peer can act as a normal *Typical* peer in the network, which is one that just participates in the network. From system modelling it becomes evident that Gateway, Security Manager, Service Manager and QoS Manager are completely different from the typical peer as other roles of peers need to locate and communicate with other peers in the network. For example, Gateway peers need to locate other Gateway peers in the network and communicate with them.

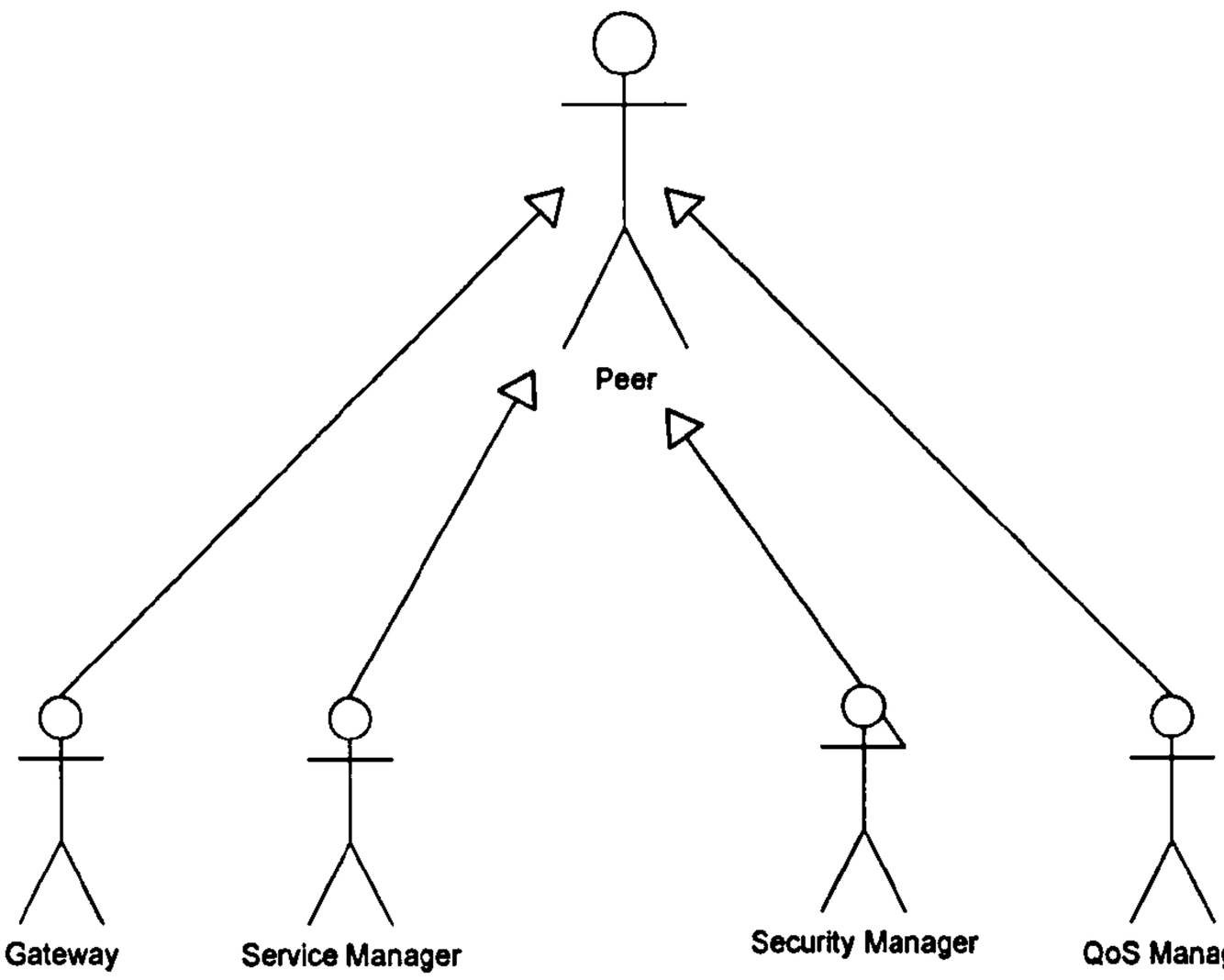

Figure 5.2 : Components Dependencies – AdHocGS Framework Actors

As discussed in chapter 4, every actor within the framework must have some common properties such as unique identity, location, list of service offering etc. Every actor must also have other common functionalities including connecting and disconnecting from a network, search for other services in the network. Figure 5.3 shows component dependencies in AdHocGS framework actors.

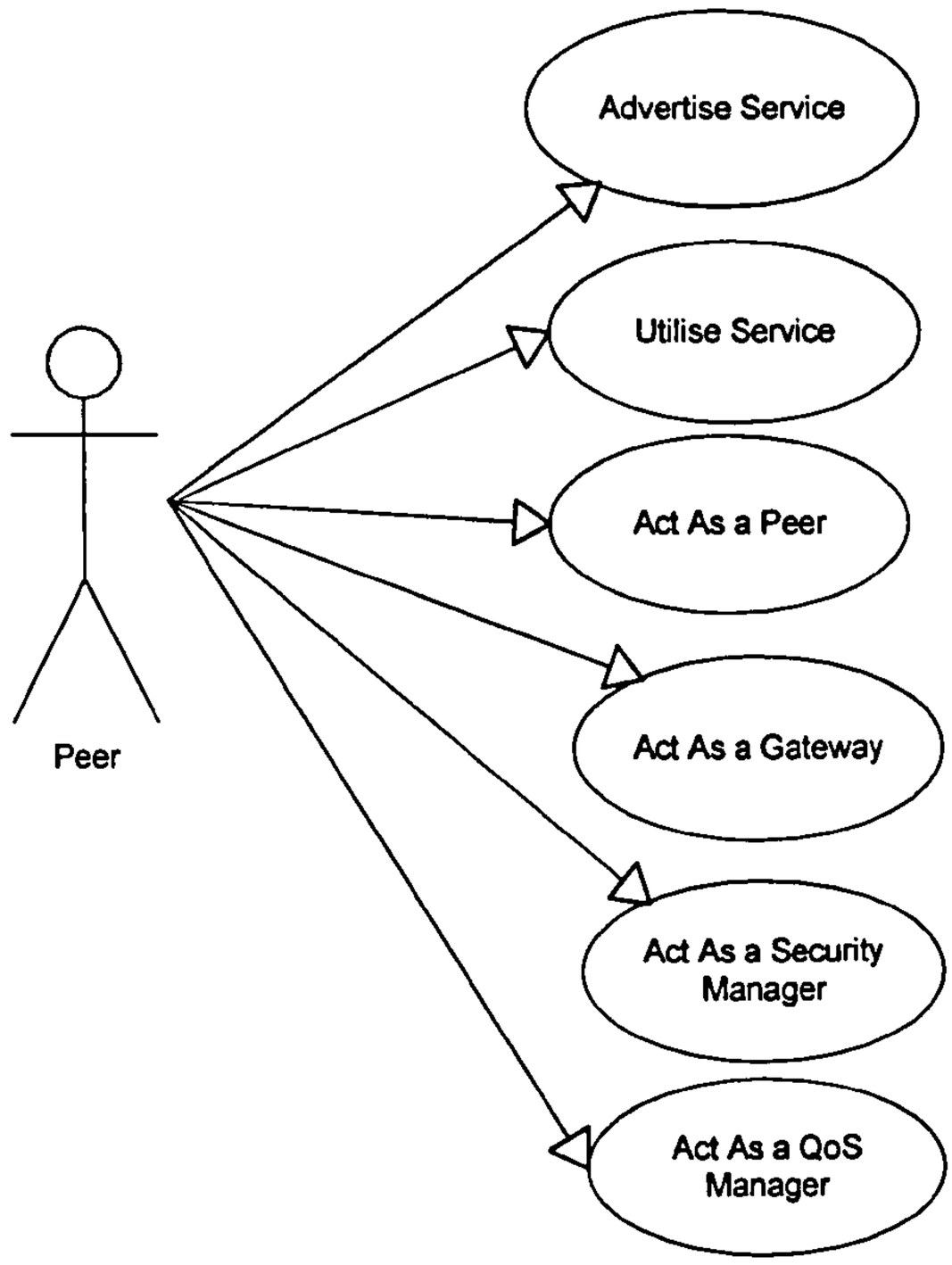

Figure 5.3 : Use Case - Peer roles in the AdHocGS Framework

Roles of the Actors in the framework are:

- *Typical Peer* – it represents any device capable of participating in a P2P network.
- *Gateway* – the Gateway actor is a specialised role of the Peer, can act as Gateway in the system apart from basic P2P functionality.
- *Service Manager* - the Service Manager actor is a specialised role of the Peer, can act as Service Manager in the system apart from basic P2P functionality.
- *Security Manager* - the Security Manger actor is a specialised role of the Peer, can act as Security Manager in the system apart from basic P2P functionality.
- *QoS Manager* - the QoS Manager actor is a specialised role of the Peer, can act as QoS Manager in the system apart from basic P2P functionality.

All the actors mentioned above belong to our AdHocGS Framework. This framework is designed to work in the P2P domain; therefore this framework can be viewed as a sub-system of any P2P network. One of our framework aims is to enable devices to share their services and discover other services. All the peers in the network at least act as Typical peer which reflects the role of peer in any P2P

network. In P2P network, any number of devices may be present in the network. Some of these devices are not resource rich i.e. less processing capacity; low memory, battery power etc, usually referred as thin peers [Starner 2002] (also as lean or slim peer) in networking and thin peers e.g. mobile phones, iPods in P2P network [Arora 2005]. These peers have very limited functionality and cannot act as Gateway peers. On the other hand, some devices are resource rich i.e. greater processing capacity, more storage capacity etc usually refer as thick peers (also as fat or rich peer) in networking and thick peers (laptops, PC's) in P2P network [Arora 2005]. These thick peers can act as Gateway peers.

The actors may change their role e.g. when a device first joins the network it may act as Typical Peer but if the device chooses to act as a Gateway service for other peers it becomes a Gateway peer. In our work, Gateways can be of two types. Users can create their own personalised gateways connecting their home or office devices in order to access them in a global P2P network or a gateway can be specialised for offering services from specific domains such as a TV channel gateway offering a number of channels across the globe, which can charge other users to use the services. Appendices-A contain the complete Use Case Model for AdHocGS Framework.

Personalised gateways allow personal users to access and manage their devices remotely in a P2P network. Using personalised gateways users can not only offer their services in a P2P network but also discover and use services offered by other gateways. For example, if a user needs to access any TV channel it can request a TV channels gateway. As in Figure 5.4, *TVChannelGateway* is offering TV channels from different networks. *HomeGateway* can request *TVChannelGateway* for a particular channel e.g. Sky Sports is not offered by any device in the *HomeGateway*. This is usually done by sending a request to the gateway peer overlay network and any gateway offering this service responds to the request and allows *HomeGateway* to use the service. This allows flexibility in our AdHocGS framework to not only offer a set of services but also request and use services offered by other remote gateways. As mentioned in previous chapters we do not consider the underlying networks, it could be P2P, Client-Server etc. Gateway Peer Overlay Network itself is P2P, so gateways can communicate directly with any other gateway(s) in the network. Peers request

services from the gateway without knowing the location of the service i.e. local or remote.

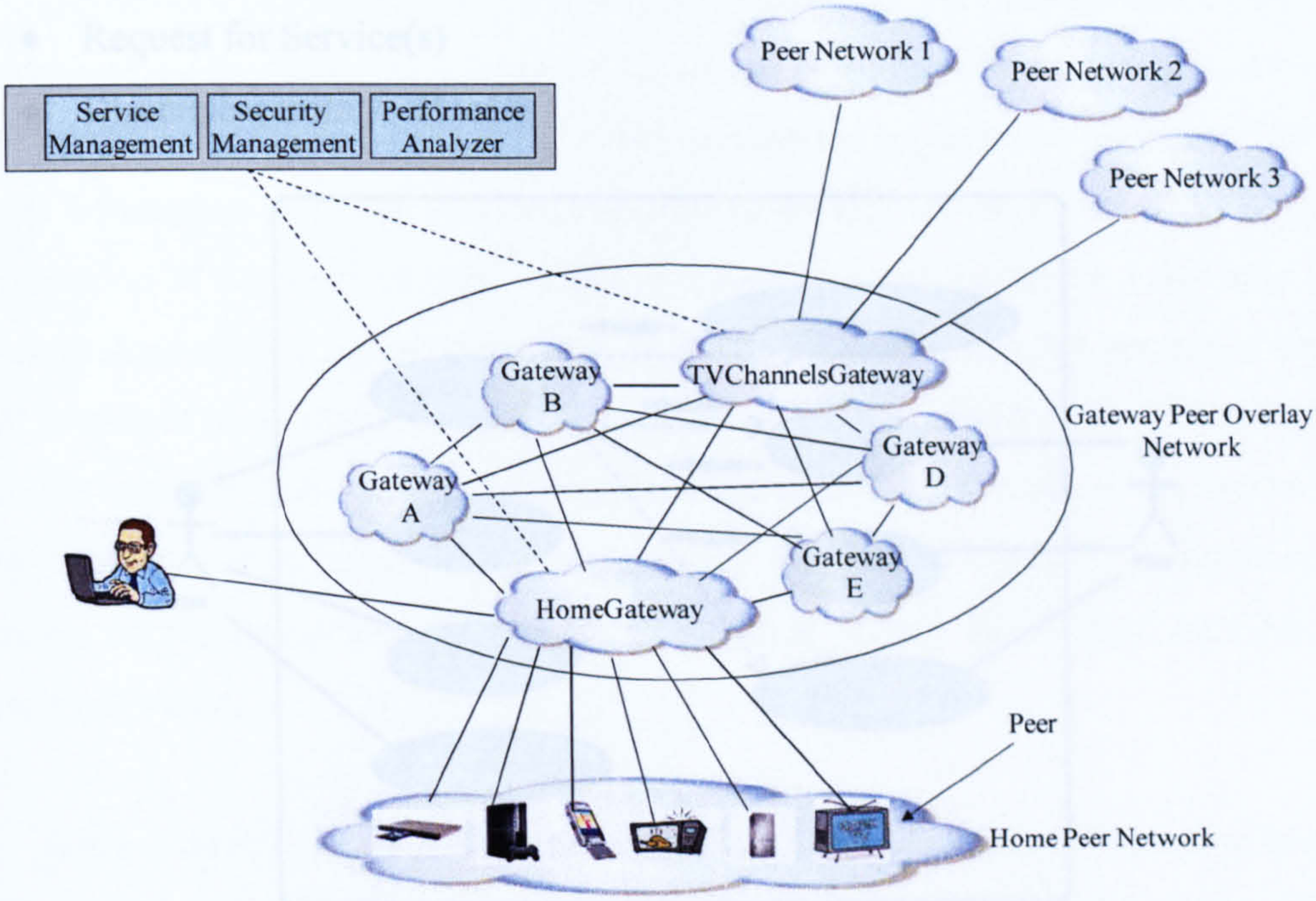

Figure 5.4 : AdHocGS Framework

Our AdHocGS framework marshals and controls compositions and performs interoperations between device and services in a controlled way with less user involvement. On the basis of information gathered at the requirements phase, the main responsibilities of an AdHocGS framework are:

- Joining the gateway peer network
- Manage services
- Manage key services such as Service Manager (SM), Security Manager (ScM) and Quality of Service Manager (QoSM)
- Accepting Gateway service requests
- Managing communication between services as well as with other Gateway peers
- Discovery of services, locally or remotely
- Discovery of alternate service following failure
- Discovery of alternate gateway service following failure
- Transfer control to alternate gateway following failure.

On the basis of our requirement, the functionalities required in our framework can be further divided into the use cases shown in Figure 5.5. These are:

- Connect to AdHocGS
- Offer Service(s)
- Request for Service(s)
- Disconnect from AdHocGS

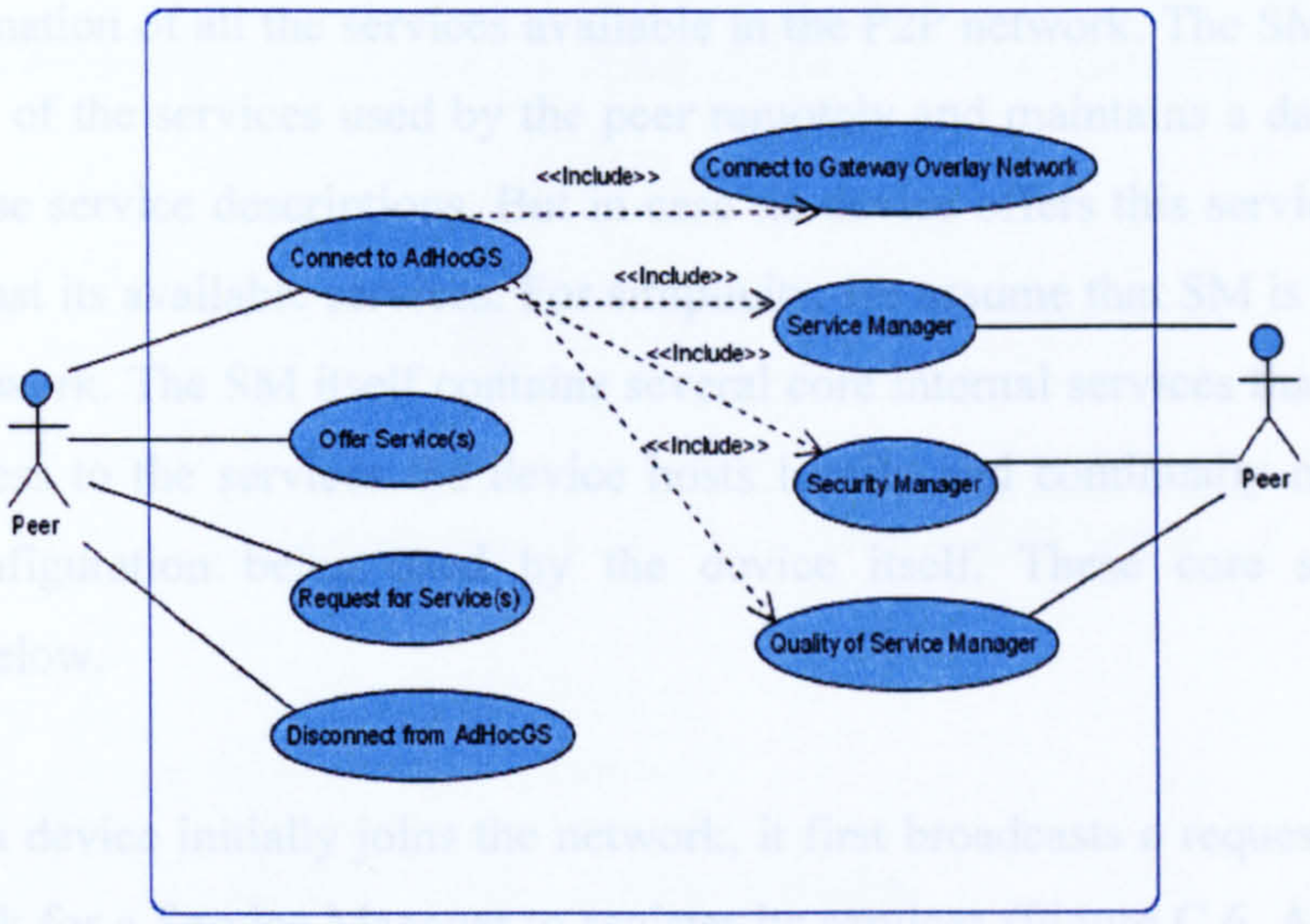

Figure 5.5: P2P Gateway Service Framework

5.4 AdHocGS Services Framework

In this section, we present further the design of the different components of the AdHocGS framework including individual components and how the components communicate with each other. This gives clear understanding of the framework and data flow between different components. We mentioned earlier that components in the framework run as services, which may be offered by different devices available in the network. The idea behind dividing our framework into different components is to provide more flexibility.

The services required to enable AdHocGS framework are the Gateway Service (GatewayS), Service Manager (SM) service, Security Manager (ScM) service, Quality of Service Manager (QoSM) service, Lookup service, Discovery service and Advertisement service. In following section we discuss these services in detail.

5.4.1 Service Manager (SM)

The Service Manager (SM) service is responsible for the management of services being used by the peer, which may be locally or remotely hosted. When a device is initially switched on, it searches for a SM service and registers its services. The SM holds information of all the services available in the P2P network. The SM also holds information of the services used by the peer remotely and maintains a database used to store these service descriptions. But in case no device offers this service, a device will broadcast its available services. For simplicity, we assume that SM is available in the P2P network. The SM itself contains several core internal services that allow it to control access to the services the device hosts locally and continually monitors the service configuration being used by the device itself. These core services are described below.

When a device initially joins the network, it first broadcasts a request within the P2P network for a Service Manager to register its services (Figure C.6, Appendix C). Any device in the network that offers a SM service may respond to the gateway service request, allowing the device access to the SM, which enables it to discover personalised services within the network. The Service Controller (SC) within the Service Manager registers services including properties such as peer name, IP address, services offered, it offering gateway service and any security constraints (if service need to be authenticated before allow other peers to use its services). Information relating to authentication is passed to Security Manager (ScM) which discuss later in this chapter. A device also registers the capabilities it supports, how well the device that provides the service can execute it given the hardware, software and network capabilities the device has. Only if the device has enough resources and exhibits adequate capabilities will the service be used within the user's personalised configuration. Service registration is shown in Figure 5.6.

Admission Controller (AC)

Admission Controller (AC) within the SM is responsible for service registration, as illustrated in Figure 5.6. When a device joins the network, it first obtains information about bootstrapping which allows the device to obtain information about the network. As discussed in chapter 4 there are different methods of obtaining

information about bootstrapping. The AC service is responsible for processing service requests received from the gateway. It tries to match the details defined in the service request with the details defined in each service description it has in the database. If a candidate service is found the service description is passed to the Service Controller (SCo). Admission Controller marks services as "L" if a peer resides locally or "R" if a peer resides remotely i.e. offered by other gateway in the Gateway Peer Overlay Network.

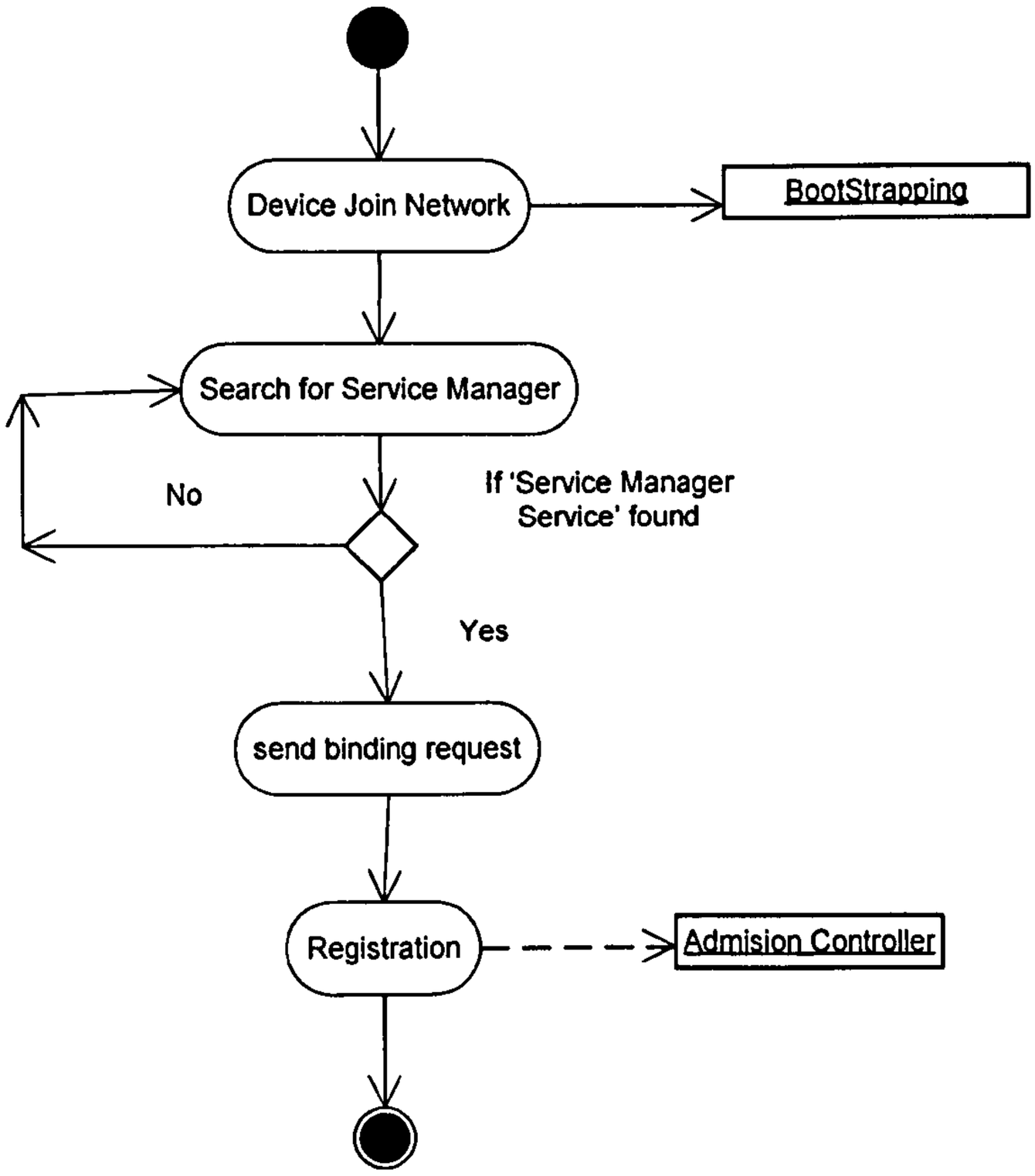

Figure 5.6 : Service Registration Activity Diagram

Service Controller (SCo)

Service Controller (SCo) receives data from AC. AC only receives information when a device first registers its services and also passes information received from the gateway as shown in Figure 5.7. SCo keeps record of all the services available, locations and other parameters. SCo manages its local cache as well, where it keeps record of information regarding the most recently used services and locations. SCo searches its local cache whenever it receives requests for a service. If it find matching information, it first checks if the device offering this service is still connected. If the

device is not available it broadcasts a service advertisement on the Gateway Peer Overlay Network to locate the requested service. SCo also exchanges information with the Service Monitor (SMo) to determine if a requested service is available and if not when it will be. If the requested service is available, SCo also checks if it requires authentication before access. If this is the case, control is passed over to SM, which then authenticates the service. Once all the information is found, it is then passed over to the Gateway service.

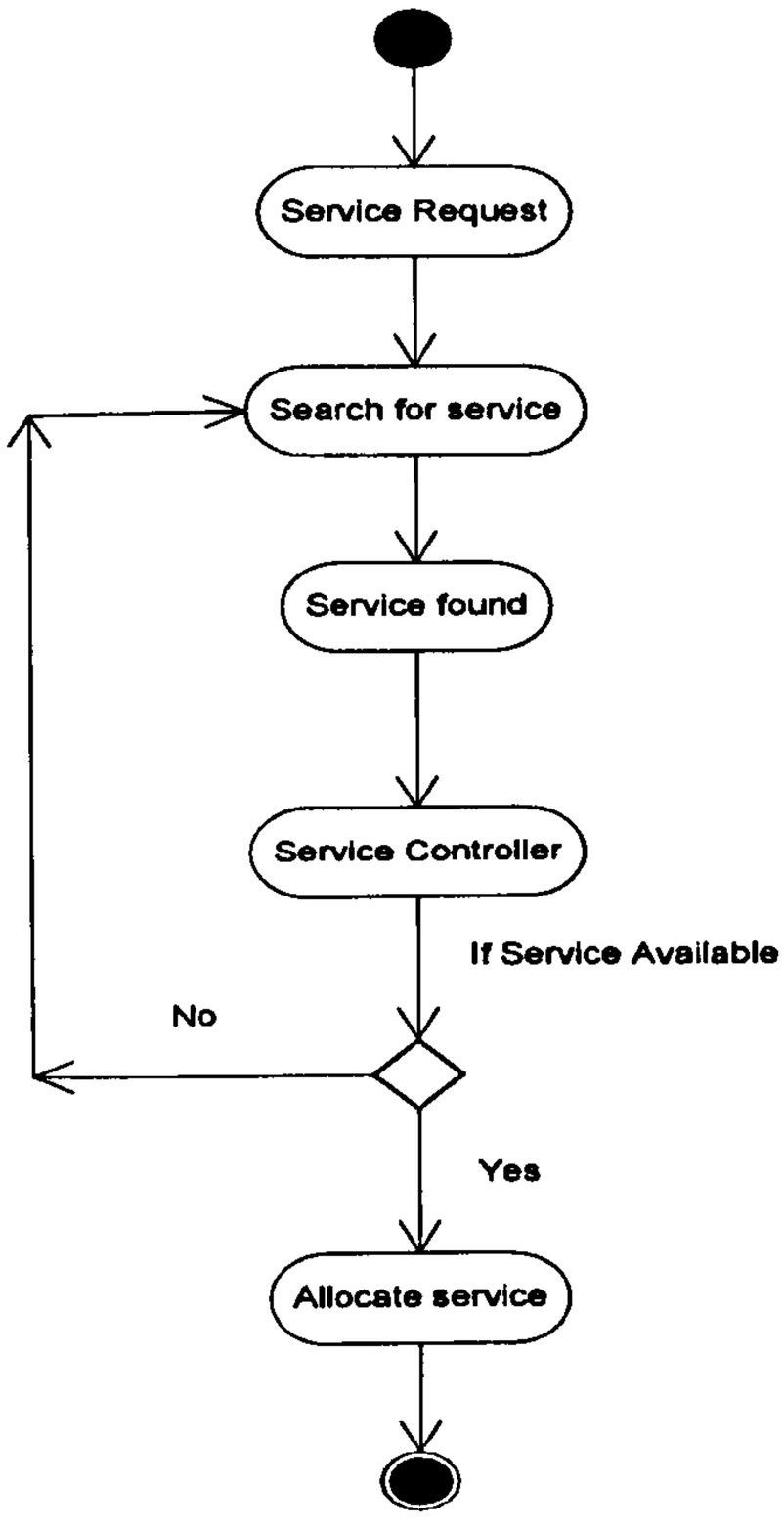

Figure 5.7 : Service Controller Activity Diagram

Service Monitor (SMo)

The Service Monitor (SMo) service is responsible for monitoring services, for example, how many peers are currently using the service. This allows the device to manage its performance to ensure that its own Quality of Service does not deteriorate. If any particular service request needs to be prioritised in the network this is implemented via SMo. When SMo receives a request from different peers for the same service, SMo can check if a particular service has priority over others. SMo also

keeps record if a service is available locally or remotely. SMo exchanges information with SCo such as if a service available or when it will be available.

5.4.2 Gateway Service (GatewayS)

The Gateway Service (GatewayS) allows peers to communicate with each other (Appendix B). As we discussed earlier we need a middleware that enables services from one location to communicate with services in the same or remote locations. GatewayS act as gateway in the network i.e. it connects two networks that are not connected physically. Devices in one network send data to another network device without worrying about underlying technology. As we discussed earlier we are not using any black box device to offer this service but any peer in the network can offer this service. We discussed earlier in this chapter that some peers in the network use *fat* or *thick* peers, i.e. ones with more resources can that act as a gateway e.g. Laptop or desktop PC.

At registration stage devices also register with the SM if they can offer a gateway service or not. This enables SM later to locate devices offering GatewayS. Our GatewayS is ad hoc in nature i.e. it only exists when needed. For instance, if a service requested by devices resides in the local network, SM can action these requests. If a requested service does not reside locally, it needs to locate service remotely. In such situation, we require a new gateway service.

SM sends messages to the peers offering gateway service to check if these devices are able to act as a gateway service (Figure B.7, Appendix B). If the peer still exists, the first peer to reply to the request becomes the active gateway and any subsequent peers will act as backup gateway(s). In case a peer is not available it will then broadcast a gateway advertisement on the P2P network (Figure B.3, Appendix B). Any device in the network that offers a gateway service may respond to the gateway service request, allowing the device access to the gateway, which enables it to discover personalised services within the network. This process is illustrated in Figure 5.8. In the presence of a gateway, it is a task of the gateway to find services requested by any device.

In some cases a number of peers may be offering gateway services at the same time. In such case, when a device requests for a gateway service there may be a number of devices responding to the request or SM knows of a number of devices offering a gateway service. In this case, we need to make a selection for the main gateway. This is usually done by SM assigning a peer as main or active gateway and further as backup or passive gateways on basis of first come first served. Active gateway acts as the gateway carrying out current operation while backup gateways backup all state information of active gateway; this process is illustrated in Figure 5.8.

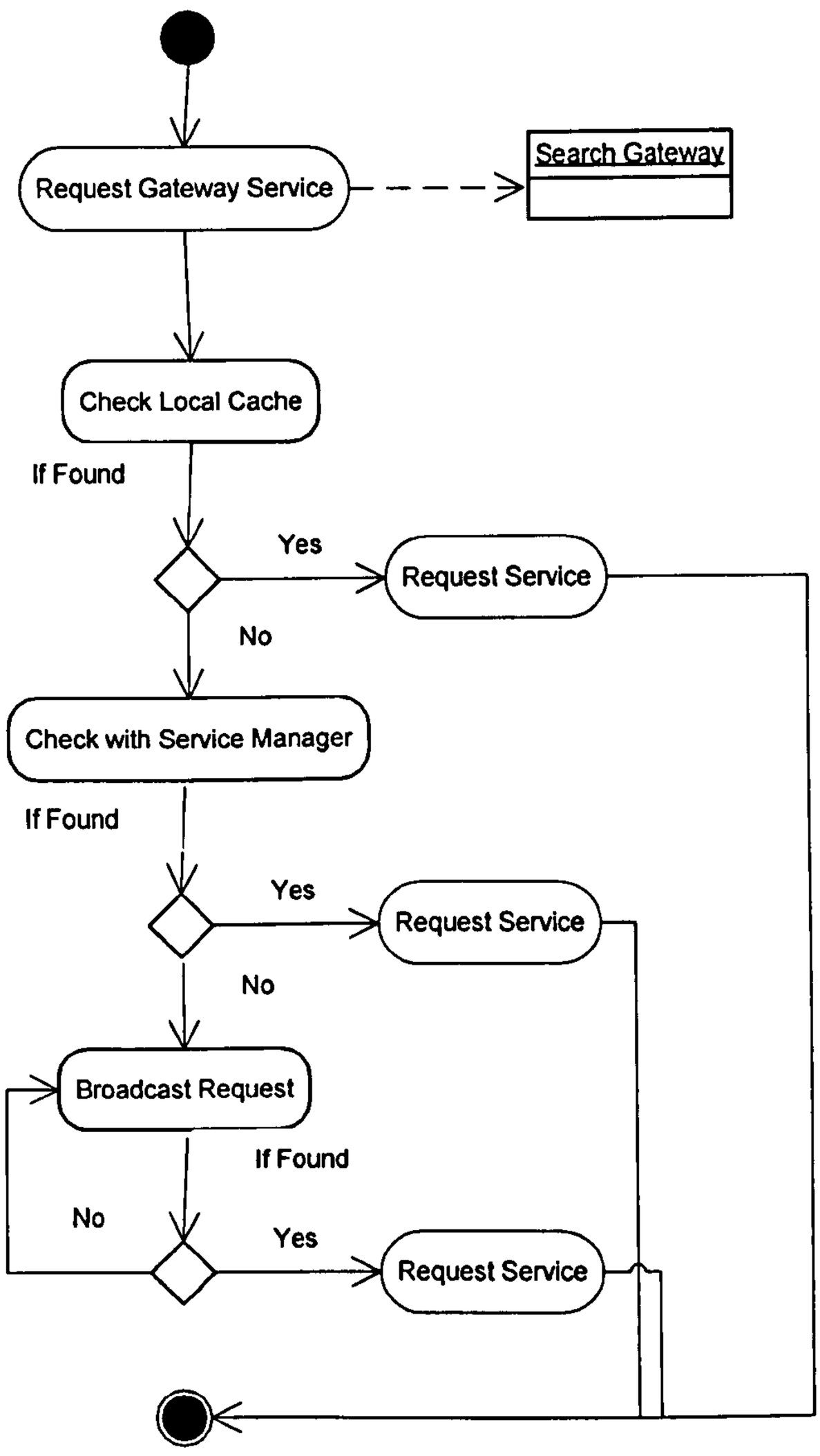

Figure 5.8 : Gateway Discovery Activity Diagram

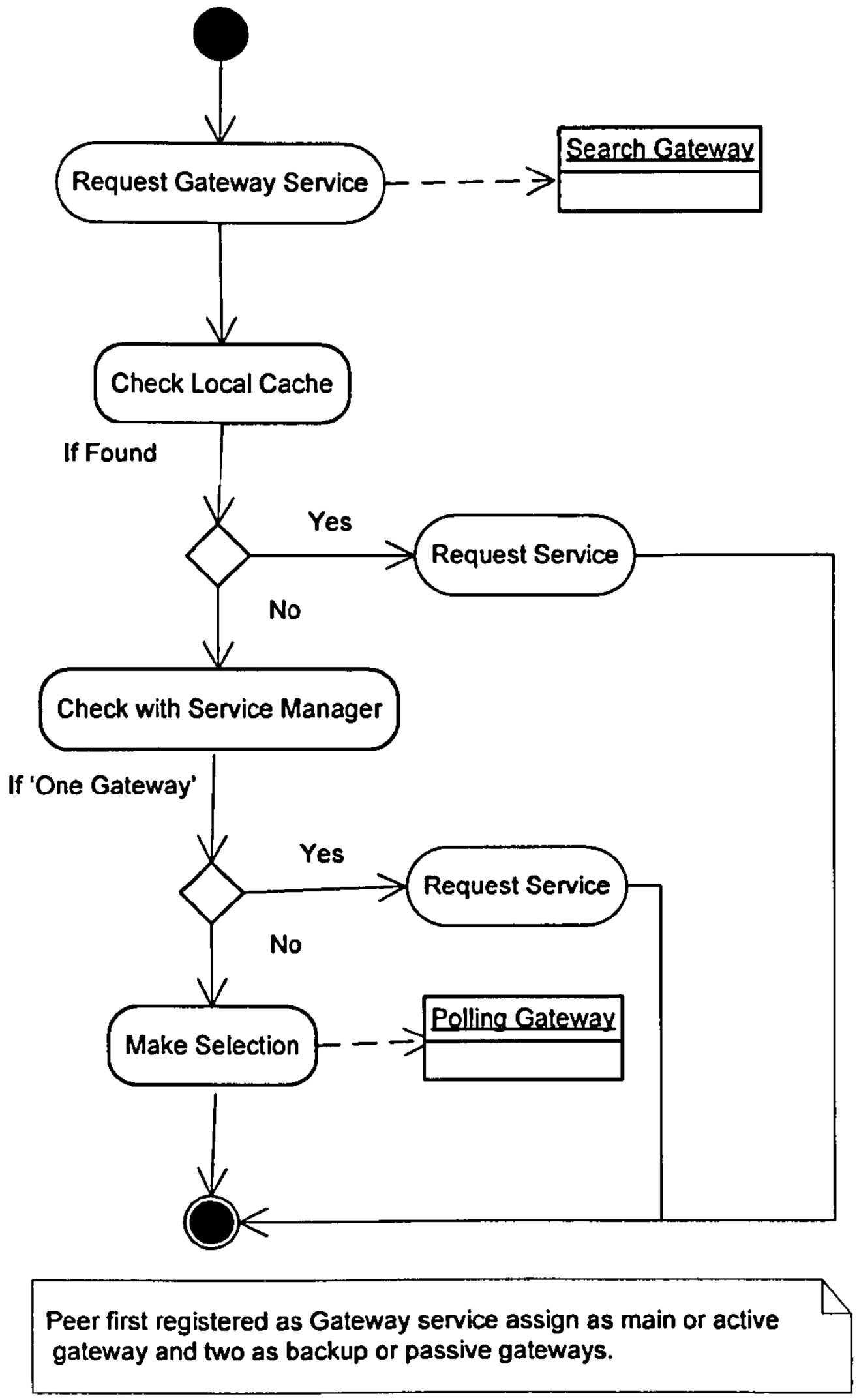

Figure 5.9 : Gateway Selection Activity Diagram

5.4.3 Gateway Replication and synchronisation

In our proposed framework, replication and synchronisation is necessary i.e. between Active and Backup gateways. Replication is the process of sharing information to ensure consistency in a distributed network, to improve reliability. Data needs to be regularly duplicated across the distributed network to ensure data consistency and improve system performance. Replication in P2P is necessary for more data availability, so in case of failure of one peer, data can be obtained from another peer.

As discussed earlier when more than one peer replies to a SM GatewayS peer request, one becomes Active while others becomes Backup gateways. We replicate data across the Gateway Peer Overlay Network to ensure persistence of GatewayS data. Data replication is a method where the Active gateway regularly sends data to its Backup gateways. Gateway synchronisation is the process where GatewayS peer requests data from other GatewayS peers when they join the Gateway Peer Overlay Network. Gateway synchronisation ensures that GatewayS peers have all the latest information in the Gateway Peer Overlay Network. Gateway synchronisation reduces broadcasting service requests over the Gateway Peer Overlay Network. Data replication overcomes the failure of single gateway i.e. in case of failure of an Active gateway, either of the Backup gateways can be promoted to Active gateway and communication can be started from the point where the last replication was performed.

5.4.4 Security Manager (ScM)

As with all computer-based systems, security is a major issue. Devices are exposed to information leakage, unauthorised access, eavesdropping and message tampering. Mechanisms are therefore needed to provide an efficient security model. ScM service ensures only authorised peers can access the services provided by other peers. When peers find candidate services, the ScM authorises, authenticates and encrypt/decrypt data transferred between peers and between GatewayS on Gateway Peer Overlay Network. The ScM service ensures that only authorised devices can access the device and the services it provides. Security can be implemented on the peer level such as Firewalls, which monitor communication coming in and out of the network. At registration a services along with other information informs the SM if it requires secure access i.e. so only authorised devices can access a particular service, this is done via Security Manager (ScM) (Figure B.9, Appendix B). When a peer successfully discovers a requested service, before service allocation if a particular service requires authorisation it will be prompted for credentials i.e. username and password. ScM can authorise these credentials as ScM stores this information during registration. Access will only be granted if the correct credentials are given otherwise access won't be granted and the device needs to search for another service. This process is illustrated in Figure 5.10.

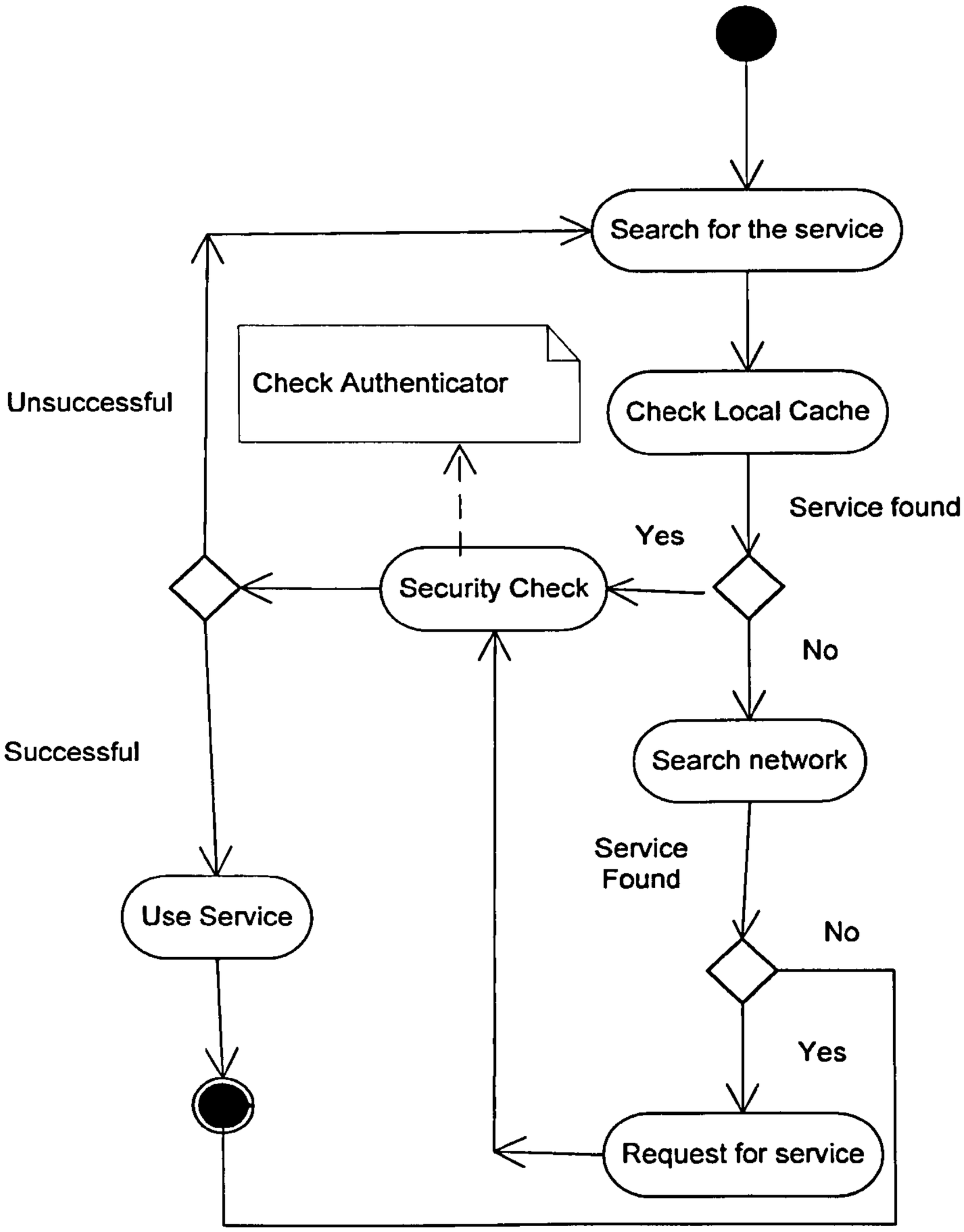

Figure 5.10 : Check Security Activity Diagram

P2P introduces a number of security issues [Cameroon 2006] such as privacy, access control, authentication, trust/reputation. In P2P there is no central controller, so it is difficult to implement any security policies or to authorise any particular peer and check data authentication. [Vroonhoven 2006] discussed a number of security issues in P2P networks such as malicious nodes can easily join the network as a non-malicious peer. This malicious peer easily damage the network in many ways i.e. by spreading virus or Trojan by sharing a virus infected file or make resources unavailable by attempting a denial-of-service attack on peers. A number of solutions have been designed to make P2P networks more secure such as authentication, access control, trust [Detsch 2006; Kumar 2006; Locasto 2005]. In our framework we

include ScM but we implemented controls at very basic level as discussed in chapter 6. We further divide ScM component into: Authenticator and Encrypter/Decrypter.

Authenticator (Auth)

The Authenticator is responsible for authenticating peers before providing access and to only allow authorised peers. If any peer has not been authenticated by Auth, the service request is declined by the SM. The basic method of verifying in our framework is username and password. When devices try to access a service they might be prompted to enter a valid username and password. This can only be implemented to secure a users own personal devices in the network i.e. home or office network. This can be setup by a user when configuring a personal gateway. When a device registers its services with SM, it also registers if a particular service requires authentication to access its services and enters the credentials required to be authenticated with. SM passes this information to Auth which checks it before allowing access to the requested services.

As we are working in a P2P network so devices joining the network might reside locally or remotely. Peers reside in the local network and might need no authentication. If services marked as "L" do not need to do security check they can bypass ScM. Otherwise if devices are marked as "R", security checks might be performed.

The level of Authentication depends upon user requirements or the network where we are using our system. There are many techniques for verification such as Username/Password, which is the basic level of security mostly used by Windows™ based applications. Security can also be implemented at peer level by building a trust relationship i.e. a peer has ensured that interaction with other peers will be fair and secure. Some notable research in this area includes [Almenarez 2008; Bo 2006]. The Trusted Computing Group (TCG) is an organisation that defines and develops open standards for trust computing and security technologies across different platforms and devices [Group 2008], TCG specifications enable secure computing without compromising integrity and privacy.

Encrypter/Decrypter (En/De)

The Encryptor/Decryptor is responsible for encrypting and decrypting data transfers between peers in order to avoid eavesdropping e.g. locally data transfer among devices should be encrypted. The main idea behind cryptography is to hide information from hackers or unauthorised users. Whenever devices try to access services they might go through security checks first. Even after authentication there is still a risk of eavesdropping, this component makes sure to encrypt and decrypt information to provide more secure communication. A number of encryption techniques are available such as Symmetric-key, Public-key cryptography. Cryptography ranges from simple to complex depending upon nature of the network or level of data security necessary in particular network.

5.4.5 Performance Analyzer (PA)

The PA ensures that enough resources are available to carry out the operations performed by devices and the services they provide. In order to achieve this it takes into account a number of Quality of Service parameters such as bandwidth capabilities and network speed, including hardware and software capabilities. The PA consists of several core internal services that enable it to determine the Quality of Service parameters described above in including the capabilities supported by devices providing particular services. These internal core services are described below. Figure 5.11 shows activity diagram for Performance Analyzer.

QoS Manager (QoSM)

In the simplest sense, Quality of Service (QoS) means providing a consistent, predictable data delivery service, in other words, satisfying user application requirements. QoS concerns the ability of a network element (for example, an application, host or router) to have some level of assurance that its traffic and service requirements can be satisfied. QoSM enables provision of better services to certain flows such as raising the priority of one flow over another. QoS may also try to guarantee that packets will not be delayed or dropped during communication. QoSM ensures necessary services are available in the network to carry out certain operation i.e. make sure enough bandwidth available for video streaming. QoSM ensures peers participate in the communication agree on data flows at certain intervals of time.

QoSM also ensures that peers participating in the network but not using services are also sharing their services, which is one of main issue in the P2P network.

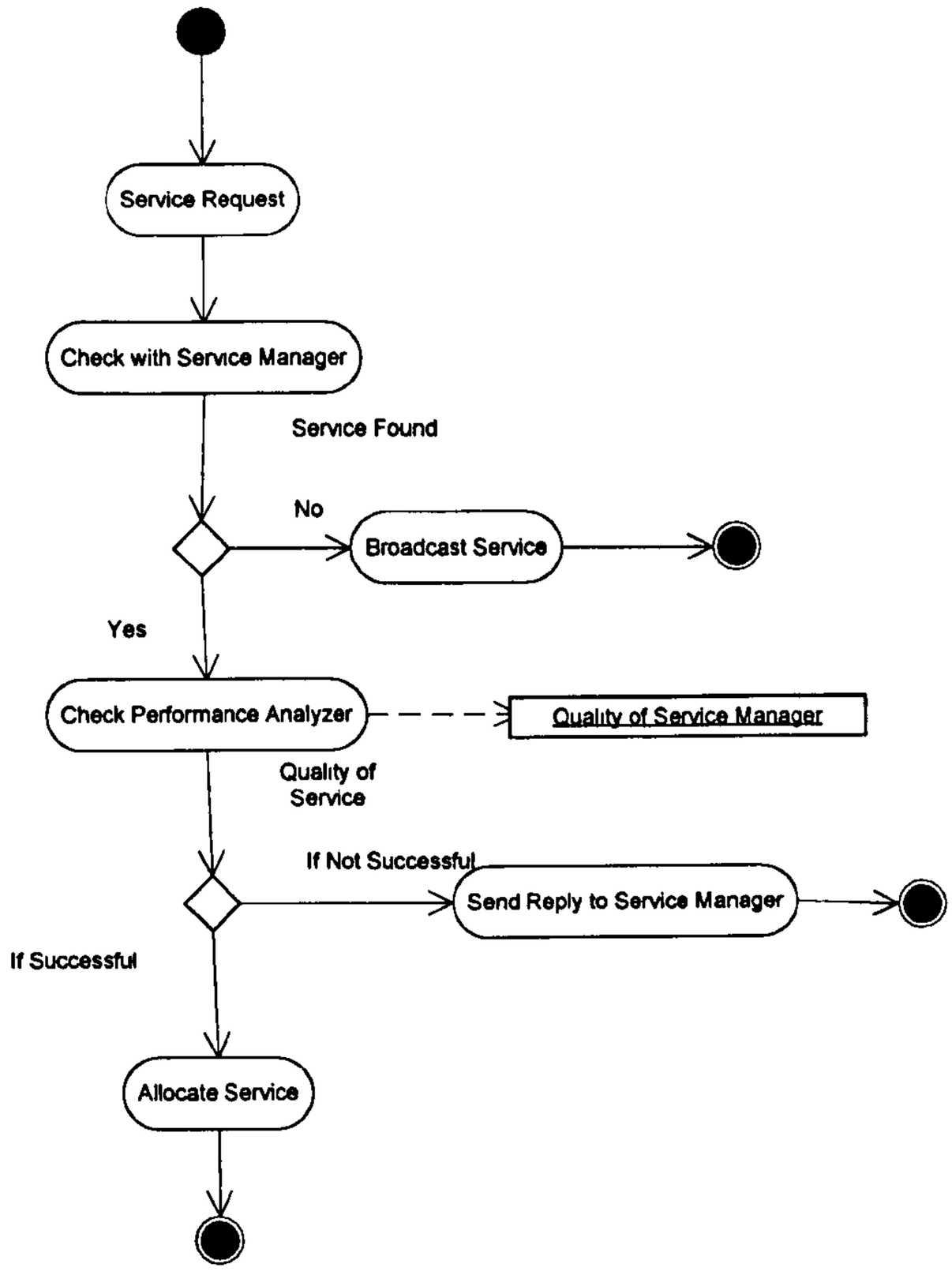

Figure 5.11: Performance Analyzer Activity Diagram

Quality of Services in distributed P2P network as a service might be accessed by a number of peers at one time and can cause network blockage or implementing priorities in the network. In P2P network congestion control and priority based scheduling are very important, as addressed by [Choi 2005]. [Núñez 2006] proposed Extended Service Discovery Protocol (ESDP) that allows discovery of services through queries to the network, propagating using "Sensible Routing". ESDP allows better performance in respect to search time, high probability of success, minimum overhead and improves received QoS. [Magharei 2007] proposed a solution for streaming of services in live P2P to residential users. A number of other researches in providing QoS in P2P networks are discussed in chapter 3. QoS is also discussed in our evaluation chapter.

Capability Manager (CM)

This service works in conjunction with the QoSM service to check the hardware and software capabilities of the device to perform the requested service, for example, screen resolution, memory and the software installed [Mingkhwan 2005]. When a device first registers its service the SM also registers its hardware capabilities which help SM to allocate the best available service. CM enables our framework to look for the most capable service within the network to carry out an operation, but also looks for the best alternatives on the network. Once a better service becomes available it can automatically stream data to the new service. When a service is discovered and matched, there may be several candidate services that offer the same functionality. CM is needed in order to check the hardware and software capabilities of a device, which are used to determine how effectively the device can execute the services it provides. For example, in the network different devices may be offering video capabilities such as a computer monitor and television.

On the basis of the response provided by the SMo, the AC will update its database that contains the list of services available locally and remotely. If the SMo service provides a positive response such as the service is not allocated to another peer it will be passed to the AC, which provides access to the services, otherwise the request is kept in the queue. If the requested service is available, before binding to the service, it must meet a certain level of Quality of Service, for example, bandwidth, the amount of data to be sent, hardware and software capabilities. The QoSM directly communicates with the CM to check the capability of the hardware to perform the service, for example, if a DVD player requests a video service, the CM must ensure that the right device is selected by checking its hardware capabilities. The results from CM and QoSM make a decision as to whether the device should be used; this being the case a connection between the peer and the service is established. Figure 5.12 illustrates the complete structure of our system, which describes how services communicate with each other.

Figure 5.12 describes the interaction between the user and the peer services. In the P2P network when the user selects services from a list of services available the device first checks the local cache on the source peer to find local advertisements. If

the required advertisements are found then the device binds to the local services otherwise a query is propagated to all peers on the P2P network requesting the required services needed. The GatewayS peer receives a message and extracts the query from the eXtensible Markup Language (XML) message and passes it to the ScM service for authorisation which first decrypts the data before the authorisation process is performed. If this is successful, the service request is passed to the SM service to check the availability of the service – if the service is available it passes the service request to the QoSM service which checks the Quality of Service attributes to determine if the requesting device is capable of executing the service.

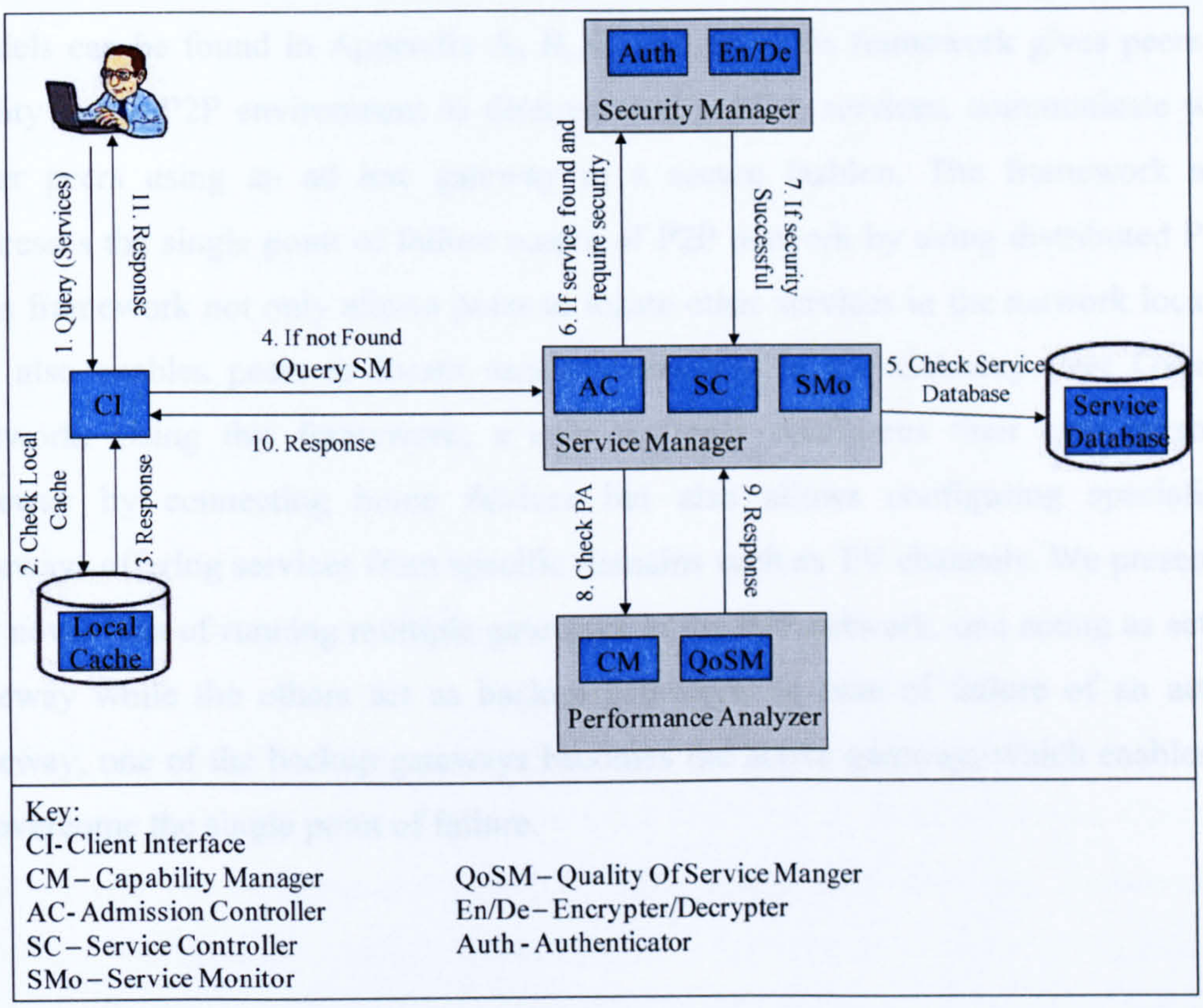

Figure 5.12: Framework in operation

In our proposed design we have created a system with a high level of flexibility, with one of the most novel aspects of our work being the ability to distribute and discover gateways as independent services that provide several fundamental features for discovering and composing services. Once the device is connected in the P2P network it will automatically discover the gateway service, without having to know the location of the gateway. Again the high flexibility evident in our system ensures that we are not dependent on a particular protocol because the communication layer itself is abstracted above via standardised interfaces.

Appendices A-D contains the complete system design notes illustrating the AdHocGS Framework services and their components in detail.

5.5 Summary

We gained a clear understanding about P2P technologies from our background related work (chapter 2) which helped us to derive the requirements (chapter 4) for a framework of ad hoc gateway services in distributed P2P environments. Chapter 4 discussed high-level design requirement and model design for a framework for service-oriented network and ad hoc gateway service framework (detailed UML models can be found in Appendix A, B, C, and D). This framework gives peers an ability in the P2P environment to discover and publish services, communicate with other peers using an ad hoc gateway in a secure fashion. The framework also addresses the single point of failure nature of P2P network by using distributed P2P. This framework not only allows peers to locate other services in the network locally, but also enables peers to locate services remotely in the Gateway Peer Overlay Network. Using this framework, a user not only configures their own personal gateway by connecting home devices but also allows configuring specialised gateways offering services from specific domains such as TV channels. We presented the novel idea of running multiple gateways in the P2P network, one acting as active gateway while the others act as backup gateways. In case of failure of an active gateway, one of the backup gateways becomes the active gateway, which enables us to overcome the single point of failure.

CHAPTER 6

6 IMPLEMENTATION AND CASE STUDY

6.1 Introduction

In this chapter, we discuss our AdHocGS framework implementation. In Chapter 5, we discussed the components of our AdHocGS framework. In this chapter, we discuss how we implement these components to achieve our objectives. The previous chapter discussed the novelty of our AdHocGS Framework, which provides Service Manager (SM), Security Manager (ScM), Quality of Service Manager (QoSM), and gateway service in P2P network and in case of failure of the main gateway provides an alternative gateway service. In this chapter we discuss our implementation with the help of a prototype and present a case study to show implementation of our framework in real world application.

6.2 Implementation Consideration

P2P applications such as Gnutella, Napster and Chord connect users in large networks to share information and resources available in these networks. Gnutella, Napster and Chord protocols mainly provide lookup and discovery functionality. Since our system is designed to address the need for Service Manager (SM), Security Manager (ScM), Quality of Service Manager (QoSM) and provide gateway service in P2P network and rediscover gateway service in case of failure we need more than routing and lookup for our implementation. The .Net framework is ideal for our scenario because it allows developers to extend its functionalities. This allows us to implement our prototype without worrying about underlying low level implementations such as connection, routing etc. .NET handles the management of peers and their connections in the network. .NET provides a number of APIs and namespace for developing P2P applications. For example, one of our design challenges is *Naming and Addressing,* i.e. how to uniquely identify peers in the

network. A Peer Name Resolution Protocol (PRNP) creates an identifier called peer name associated with an endpoint (IP address, port number, communication protocol) and publishes it for other peers to be able to resolve. Using an endpoint, a peer can either send data directly to another peer or send data to all available peers. Using *System.Net.PeerToPeer* we can create a peer name and publish it for other peers to resolve.

6.2.1 P2P Application

In writing any distributed application, communication is a main element. The most common model is Client-Server, where clients send requests and a server responds to requests. In this model, a client only knows how to request while a server knows how to respond to client's requests. A browser talking to a web server is a typical example of this model. Browsers can send information to the Web Server and the server responds back to each request by sending web pages. On the other hand, P2P applications act both as client and server. P2P not only requests information from other peers but also responds to other peers. In a typical P2P application the key features are:

- *Discovering peers* must be able to find other peers or applications that are sharing information. In Hybrid P2P this can be done by contacting a central server that contains list of all peers in the network as every peer registers itself with the server.
- This can also be done via a *broadcasting* or *discovery* mechanism. In some P2P applications a peer broadcasts itself, which can then be discovered by other peers in the network.
- *Querying peers* after discovery, for content sharing.
- *Sharing content* peers can share content, once a query is resolved.

6.2.2 About .Net

In order to create our AdHocGS framework, we did the following:

- Create a user interface;
- When a device arrives in the P2P network, register its services;
- Advertise peer services;
- Discovery and lookup services;

- Provide GatewayS in response to peer requests;
- Locate particular services in a P2P network using a gateway;
- Checking security;
- Connect peers using gateway;
- Make communication possible between two devices;
- In case of failure of active gateway, connect to a backup gateway without losing session data;

The .NET framework has a wide capability for creating P2P applications; following are some models in .NET framework to program P2P applications.

Web Services comprise of the following main aspects that are included as shown in figure 6.1:

- XML Web Services, the communication between services and the application via standard format called XML, universal and accepted on any platform.
- WSDL (Web Service Description Language), contains descriptions about the Web Services including the information about the namespace of the xml file, it also hold the description of the elements that the service consists of.
- SOAP (Simple Object Access Protocol) is a protocol which enables the Web Services inter-communication.
- UDDI (Universal Description, Discovery and Integration) that monitors publication and discovery of the Web Services implementation in respect to message communication between the application and Web Services.

The web service model follows publish, find and bind paradigm as shown in figure 6.2. First, a service provider publishes a web service in a web service registry. In the second step, a web service requester searches for a web service that meet its requirement, and may find multiple matches, and so it chooses a service. In the third step, web service requester then downloads the service description and binds with it to invoke and use the service.

handling registration, discovery and content searching in P2P application. It allows P2P applications to listen to incoming requests, process them and send back information in the form of objects. A Web Service in the .NET framework is accessible via the Simple Object Access Protocol (SOAP) [Louridas 2006] that uses HTTP as a transport and XML for data description to receive and transmit application data. We do not have to know about the platform, object model and programming

language to implement these services. Figure 6.3 shows sample code for P2PService web service.

```
<%@ WebService Language="C#" Class="P2PService" %>

public class P2PService : WebService
{
        public static ArrayList PeerFiles;
        private static String MyLock = "lock";
        P2PService()
        {
            if( null == PeerFiles)
            {
                lock(MyLock)
                {
                    if( null == PeerFiles )
                        PeerFiles = new ArrayList();
                }
            }
        }
}
```

Figure 6.3: C# code for web service

Creating a web service in the .NET framework is as easy as creating a class in a page on the server and Web Service can be called from the peer application by calling a method on a proxy class that is created with .NET. *Windows Forms* found in *System.WinForms* namespace, in .NET framework is used for writing windows-based GUI applications that enable peers to log in, request/share content.

Web Forms found in *System.Web* namespace, makes possible returning HTML content to a peer application. In P2P application start-up it registers with the web service and can call a *Web Forms* application in order to get latest HTML content from the server. *Service Process* found in *System.Service.Process* namespace is used to discover services. As mentioned earlier Web Service could be used if the discovery mechanism is using HTTP protocol, but the service process could listen for other protocols as well.

When building secure web services, there are a number of security threats associated with it such as unauthorised access, parameter manipulation, network eavesdropping and message replaying, as shown in Figure 6.4. *Unauthorised Access* to only provide sensitive information to authorised users can be done using username and password. *Parameter Manipulation* refers to unauthorised modification of the data transfer between web services i.e. an attacker can intercepts messages during transmission and modify it before sending to the destination. This usually occurs when messages are not signed or encrypted. With *Network Eavesdropping* an attacker can see messages transmitted between services i.e. an attacker can monitor messages using network monitoring software and steal sensitive data in it which might be credential information. This usually occurs when credentials are passed in plain text or no message level encryption is used.

Security in .NET framework is achieved using security namespaces such as *System.Security*, *System.Web.Security*, *System.Security.Cryptography* as shown in Figure 6.5. Web services use *System.Web.Security* which contains classes to manage web applications authentications and authorisation. This includes Windows, Forms, URL's and file authorisation controlled by *UrlAuthorizationModule* and *FileAuthorizationModule* classes [Microsoft 2008].

In order to build secure web applications we mostly use *FormsAuthentication*, *FormIdentify* and *PassportIdentify*. *FormsAuthentication* helps with form authentication and authentication ticket manipulation, *FormIdentify* is used to

encapsulate the user identity that is authenticated by *FormsAuthentication* and *PassportIdentify* used to encapsulate the user identity that is authenticated by Passport authentication. In .NET framework different types of authentication token can be used such as:

- User name and password.
- Kerberos ticket
- X.509 certificate
- Custom token

In .NET framework username and password credentials can be send in the SOAP header as shown in Figure 6.6; however they are sent as plaintext this approach can only be used in conjunction with SSL.

```
<wsse:Security

xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/12/secext">
  <wsse:UsernameToken>
    <wsse:Username>Bob</wsse:Username>
    <wsse:Password>YourStr0ngPassWord</wsse:Password>
  </wsse:UsernameToken>
</wsse:Security>
```

Figure 6.6 : Code for Security in Web Services

In the following section we discuss our implementation for our AdHocGS framework in detail. We carry out experiments with our prototype on a number of machines acting as a peer offering different services such as GatewayS, video service, audio service. For our prototype, three peers offering gateway service, locate a video service to play video. We dropped our main gateway and transferred control to the next gateway to resume video.

In section 4.2, we discussed a scenario of an estate agent owing a portfolio of houses across the country and controlling their heating systems remotely. We now discuss a prototype scenario for an estate agent as a proof of concept for our framework. In this section we have implemented it via our AdHocGS framework.

Consider an estate agent with more than 300 houses in his portfolio across Northwest England e.g. Liverpool, Manchester, Chester, Huyton, and Warrington etc. All the

houses require monthly maintenance such as meter readings for gas and electricity. All these houses in different locations require monthly maintenance on specific dates which might be different from location to location. The estate agent has to pay visits to the houses or pay his local agent to do this job, which is time consuming and costly as well. Apart from heating systems, there are a number of devices which may also require periodic maintenance, controlling and monitoring i.e. heating, gas, electricity, broadband, satellite box. Also the estate agent keeps a record of heating, electric, gas consumption on these properties to cut down on his bills, which is again time consuming to monitor and calling providers to switch to different plans.

Imagine a situation, the tenants have problems with their home security system, home central heating system etc. In such situations the agent needs to send someone to visit and fix the problem. For example, one tenant requires a change to central heating settings e.g. increase or decrease heating at midnight but he may not have any access to the central heating controller. In this case, the agent may appoint someone to visit the home at midnight. But if the maintenance person lives remotely from the tenant house he might not be able to pay a visit until the next day. Instead of sending an engineer or visiting himself a preferable situation would be when a tenant requests a heating setting change the agent can remotely change the setting.

In order to allow the above capability in all scenarios, we have to provide various services and functionalities.

- Gateway Service: through which we can communicate.
- Service Manager: to manage all the peers connected to the gateways, i.e. management of peer connected to *Liverpool* gateway.
- Security Manager: All the devices or services should be authorised before allowing access to any device in the gateway and also allowing authorised person to access gateways i.e. estate agent. Sending information over the internet is a security concern.
- Performance Analyser: As we are discussing P2P environments we cannot guarantee the availability of any service as there is no control over devices joining and leaving the network. We try to attain a level of Quality of Service, i.e. enough resources are available to provide a consistent, predictable data delivery service.

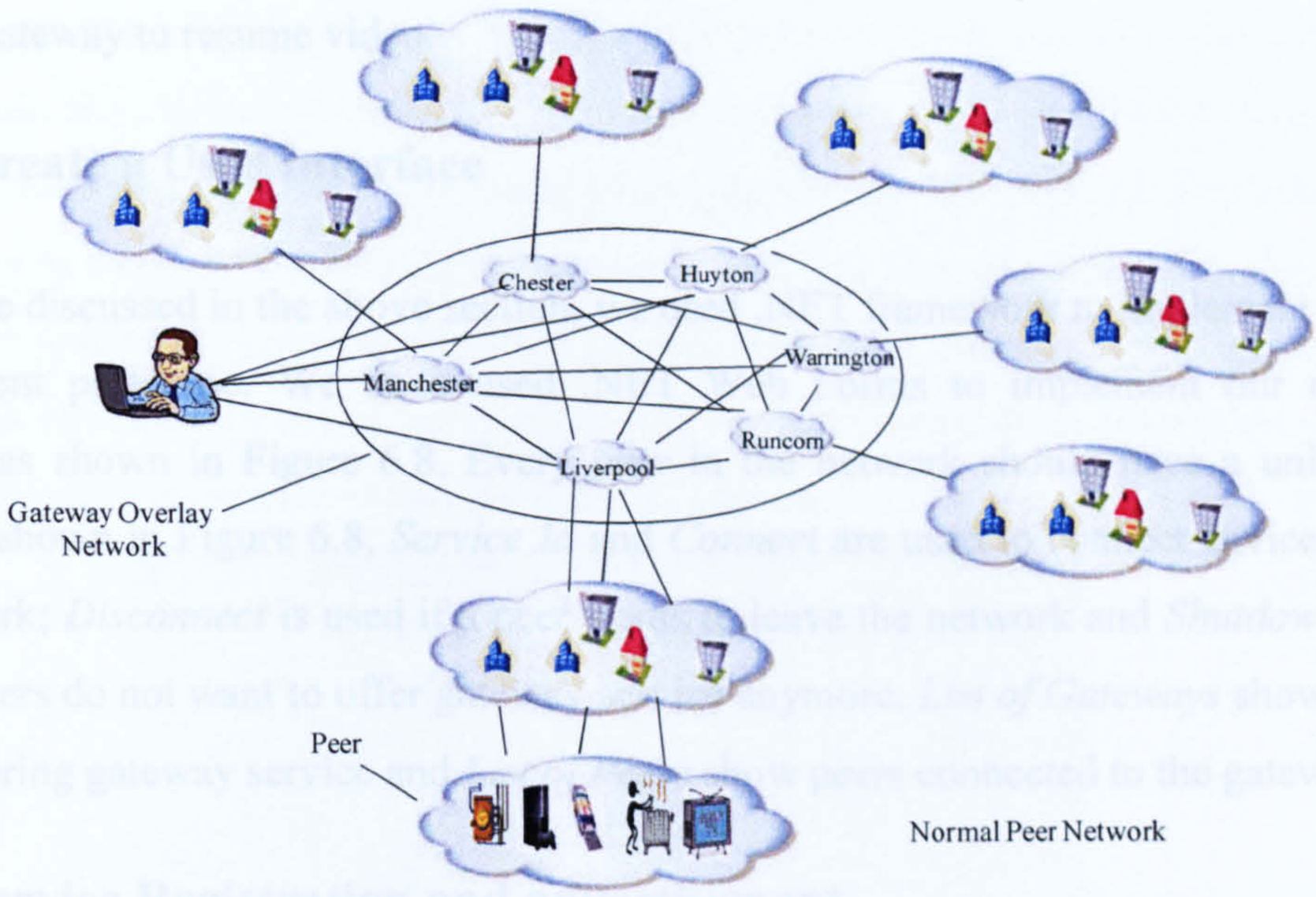

Figure 6.7 : Estate Agent Scenario

All the devices in home compose together in a gateway e.g. *Liverpool* or *Manchester*. The agent can setup a gateway, connecting home appliances to the gateway. As in Figure 6.7 houses in different locations compose together into different gateways, results in creating a Gateway Peer Overlay Network along with other gateways connected in a P2P fashion. This gateway not only allows the agent to control the central heating but also helps in meter reading. Gateways are flexible and devices can be added or removed. In figure 6.7 devices at a home make a peer network and can communicate with each other or may be connected as Client-Server. Using this gateway service the agents can not only manage their houses but also communicate with other gateways in the network. This gateway can also locate other services offered by other gateways and use them whenever needed.

In the following section we discuss the implementation of our AdHocGS framework in detail and use the estate agent scenario as a case study. In this prototype we show how we can implement the estate agent scenario based on our AdHocGS framework. We carry out experiments with our prototype on a number of machines acting as a peer offering different services such as GatewayS, house heating system service and audio service. For our prototype, three peers offer a gateway service, locate a house heating system, a video service to play video from security camera to

allow access to an engineer. We dropped our main gateway and transferred control to the next gateway to resume video.

6.2.3 Create a User Interface

As we discussed in the above section, we used .NET framework to implement our estate agent prototype. We have used .NET Web Forms to implement our user interface as shown in Figure 6.8. Every peer in the network should have a unique name, as shown in Figure 6.8, *Service Id* and *Connect* are used to connect devices in the network; *Disconnect* is used if a peer wants to leave the network and *Shutdown* is used if peers do not want to offer gateway service anymore. *List of Gateways* showing peers offering gateway service and *List of Peers* show peers connected to the gateway

6.2.4 Service Registration and advertisement

When a device first arrives in the network it needs to register its services. Service Manager (SM) is responsible for the service registration and management of services being used by the peers (Appendix B). When a device first arrives it needs to register its information with SM. Information such as peer name, peer ID, services offered, offering gateway service and security constraints. Security refers to whether authorisation is required to access that particular service. An advertisement is an XML message that describes the fundamental metadata associated with specific features of services, such as endpoint binding information, Quality of Service parameters and service capability descriptions. When a device first connects to the network it can advertise its services in the P2P network. Usually a device registers its services with the SM but in case no device offers SM services then a device can broadcast its services.

6.2.5 Discovery and lookup

When a device first connects to the network, it registers its services with Service Manager such as IP Address, peer name, services offered and service location. It will also register other details such as if it is offering a gateway service and any security constraints. Using location information the SM knows services that reside locally or remotely, in case a requested service does not reside locally a remote service can be

used. Figure 6.8 shows the main screen of our AdHocGS Framework prototype. This is the user interface for the device to join the P2P network. First, before connecting to the network, the device enters its name, in this case *peer 4*. As mentioned in earlier chapters other peers in the network could offer gateway service, depending whether it want to act as a GatewayS peer. In this case *peer 4* wants to act as a gateway, the box is ticked '*want to be gateway for other peers'*. Once the peer is connected to the network, it can obtain a list of peers connected in the network by querying SM. The peer can refresh the list every time *Refresh List* clicked.

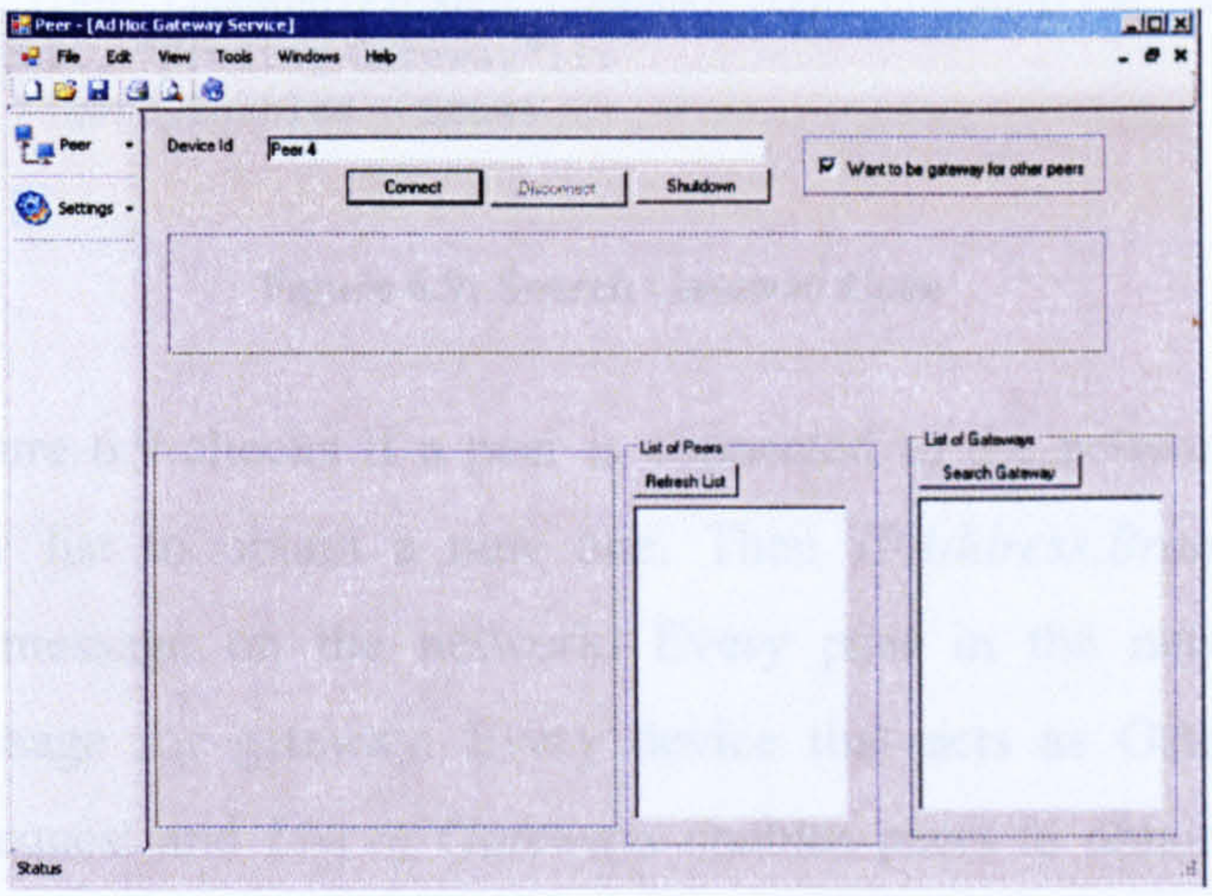

Figure 6.8: AdHocGS Framework

6.2.6 Discovery of Gateway

As mentioned in earlier sections, a peer can act as a gateway in network. When a peer connects to the network it obtains a list of peers attached in the network. In order to find another service in the network, it first needs to find and connect to a gateway. In our case before locating any particular house in estate agent portfolio, we first need to find a gateway (Appendix B), which correspond to estate agent portfolio i.e. *Liverpool, Warrington, and Chester* etc. The estate agent can obtain this list by clicking *List of Gateways* button as shown in Figure 6.8, which allows an estate agent to select location to gain access to that particular gateway. For example, an estate agent wants to access devices connected in the *Liverpool* location. In order to do that the estate agent needs to connect to the gateway and can check available gateways by clicking *List of Gateways.* Figure 6.9 is showing code for how our framework finds a GatewayS peer in the Gateway Peer Overlay Network. In Figure 6.8, peers can search for the gateway using *Search Gateway*. A gateway request is broadcast within the

network. Figure 6.9 is showing code for how our framework finds a GatewayS peer in the network.

```
if (this.client.Connected)
            {
                lblMessage.Text = "Searching for gateway.........";
                this.client.CommandReceived += new
Proshot.CommandClient.CommandReceivedEventHandler(client_CommandReceived);
                lstClients.Items.Clear();
                gateways.Clear();
                this.client.SendCommand(new
Proshot.CommandClient.Command(Proshot.CommandClient.CommandType.GateWay,
IPAddress.Broadcast, "Finding Gateway"));
                timer2.Enabled = true;
            }
```

Figure 6.9: Search Gateway Code

Code in Figure 6.9 checks if a peer is connected to the network and clears the existing gateway list to obtain a new one. Then *IPAddress.Broadcast* command broadcasts this message on the network. Every peer in the network receives a broadcasted message for gateway. Every device that acts as GatewayS peer will respond to the request and *List of Gateways* enables peers to obtain the list of peer offering GatewayS, Figure 6.11 shows two peers in the network offering gateway service. It is possible that more than one peer is offering GatewayS peer service within the network. The first peer to respond to the GatewayS request becomes the *active gateway* while the other becomes *backup gateway*(s), the code in Figure 6.10 shows *PollingGateway()*. We set a timer during which a peer waits for gateway request replies. When a timer expires, devices already replied to the request act as gateways. *Active gateway* is responsible for service discovery, service request, connecting peers and monitoring communication between peers while *backup gateway*(s) mirrors it. As in Figure 6.11, we have 3 available peers and 2 gateways. In this case *150.204.50.203* is our active gateway and 150.204.49.201 backup gateway.

```
switch (e.Command.CommandType)
            {
           ……..
            case (Proshot.CommandClient.CommandType.PollingGatway):
pollingGatway();  break;
          ……..
}
private void pollingGatway()
        {
            if(this.client.Connected && chkIWantTobeGateway.Checked)
            {
                this.client.SendCommand(new
Proshot.CommandClient.Command(Proshot.CommandClient.CommandType.IWantToBeGatway,
this.client.ClientIP, "9999"));                }
```

Figure 6.10: Polling Gateway code

When there is an SM in the network, when services register SM also records if a particular device is offering this service shown as the ticked box in Figure 6.8, so a device needs to retrieve gateway information from the SM.

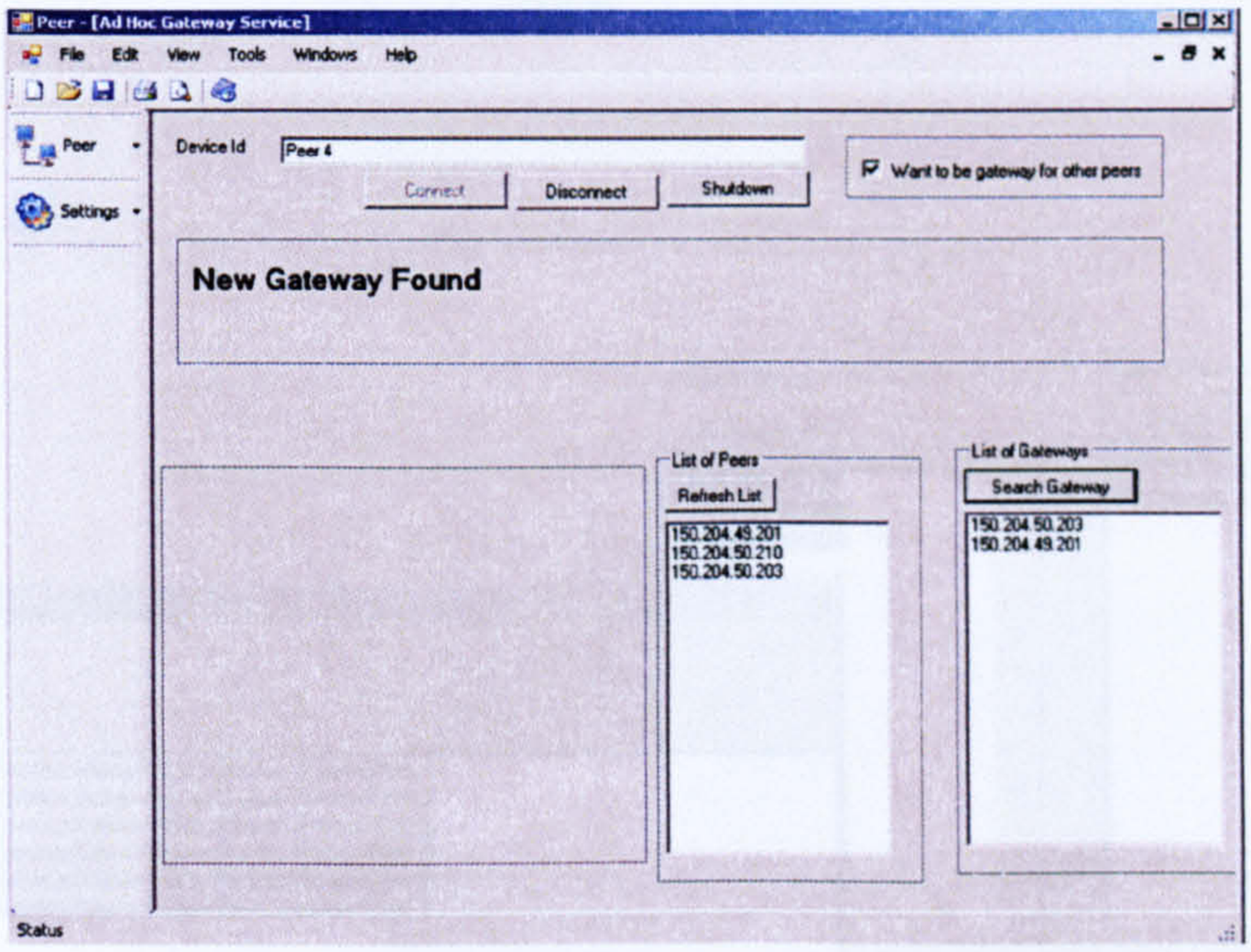

Figure 6.11: Gateway search and list

6.2.7 Discovery of connected service

When a peer requests for GatewayS peer, the list of available service that offering GatewayS peer appear as in Figure 6.11. In our prototype, we need to locate a service connected to the *Liverpool* gateway and change its heating settings. On successful discovery of a gateways service, estate agent can obtain list of peers (houses) connected with the gateway. In our prototype, 150.204.50.203 corresponds to *Liverpool* gateway. When the estate agent selects this gateway, can obtain a list of peers connected to this gateway by clicking *List of Peers.* As our GatewayS peer

communicates with other gateways in the overlay network, it consists of a number of gateways offering range of services in P2P network. Figure 6.13 shows the coding for *ServiceList* when gateway checks service connected with Liverpool gateway, a timer is set for devices to respond to the request. When the time expires, services are displayed as shown in Figure 6.12. Peers can obtain this information by querying SM. In our case, a peer corresponds to the house, so when estate agent selects a peer can obtain the list of devices connected to that peer in the particular house. For our prototype we are only interested in the heating system of the house, so the estate agent can select heating system from the services list.

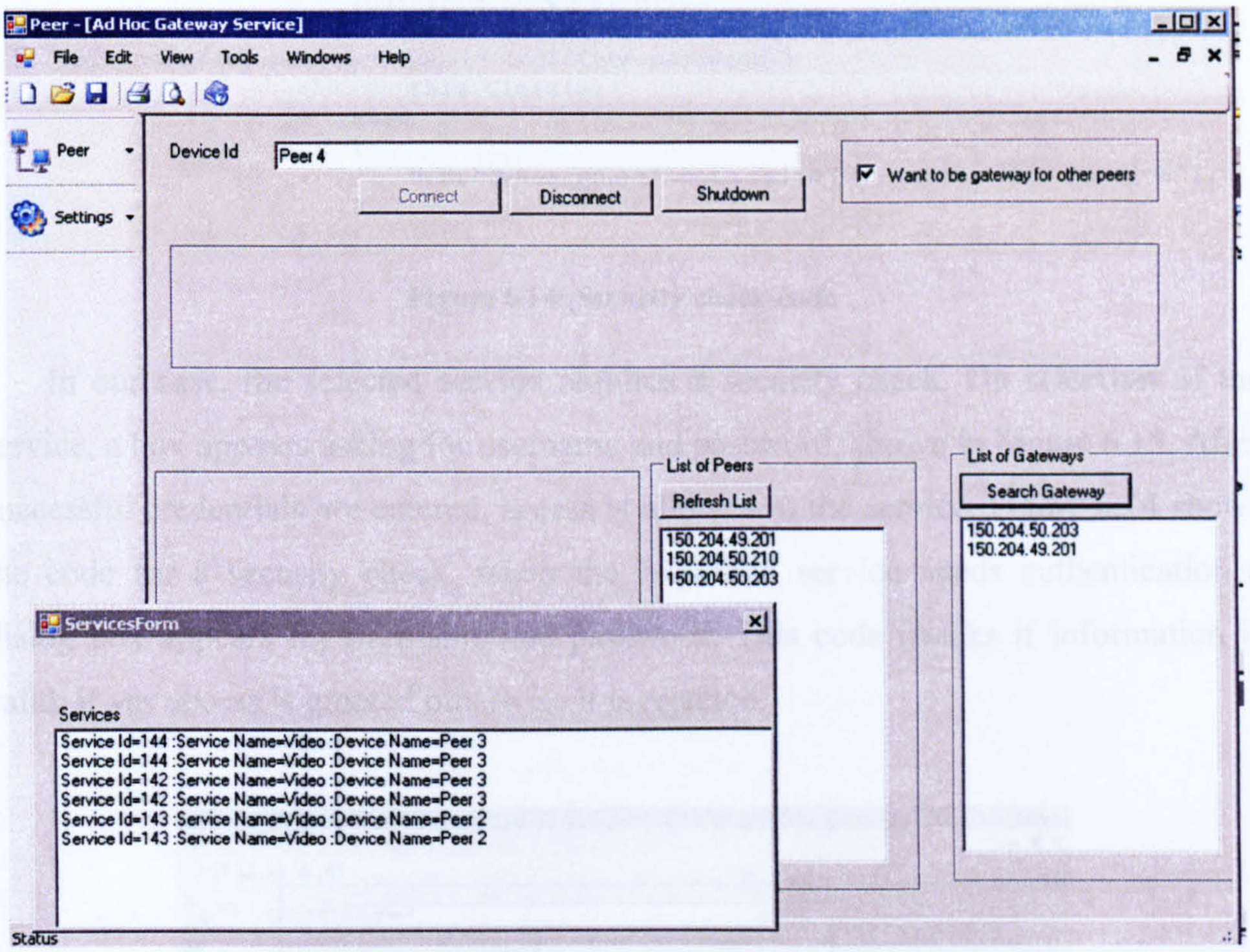

Figure 6.12: Services connect with peer

```
switch (e.Command.CommandType)
            {
          ……….
               case (Proshot.CommandClient.CommandType.ServiceList):
                    AddServiceToList(e.Command.MetaData);
                    break;
                case (Proshot.CommandClient.CommandType.EndOfServiceList):
                    ShowListToUser();
                    break;
          ……..
            }
```

Figure 6.13: Service List Request Code

6.2.8 **Security Check**

As mentioned in chapter 4, our design also includes a Security Manager (ScM) responsible for secure connection with the service, in case a particular service requires it. Also mentioned above when a device registers its service it also includes security constraints (if service needs to be authenticated, it will send the information needed requested before allow other peers to use its services) such as username and password. We only implemented the basic security in our system of username and password.

```
if (serviceEntity.AthenticationRequired)
                {
                    loginform.ShowDialog();
                    if (serviceEntity.userName.Equals(loginform.userName) &&
serviceEntity.password.Equals(loginform.password))
                        this.Close();
                    else
                    {
                        MessageBox.Show("Login failed to access this service");
                    }
                }
```

Figure 6.14: Security check code

In our case, the selected service requires a security check. On selection of the service, a box appears asking for username and password, shown in Figure 6.15. After successful credentials are entered, access is allowed to the service. Figure 6.14 shows the code for a security check, when the requested service needs authentication a dialog box appears for username and password. This code checks if information is valid, if yes access is granted otherwise it is rejected.

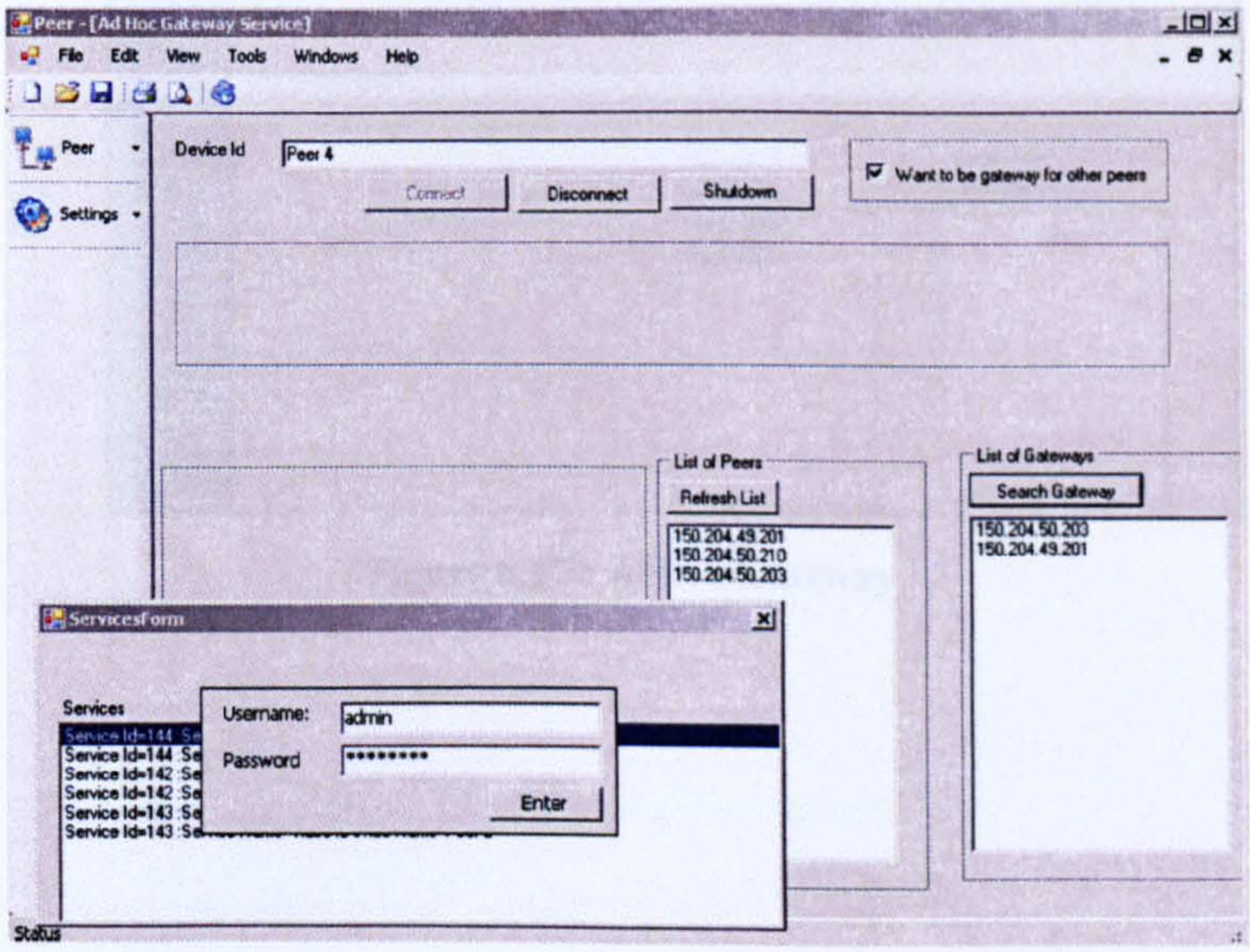

Figure 6.15: Service Security

After a successful security check, the estate agent can then access and change heating settings in the house as shown in figure 6.16.

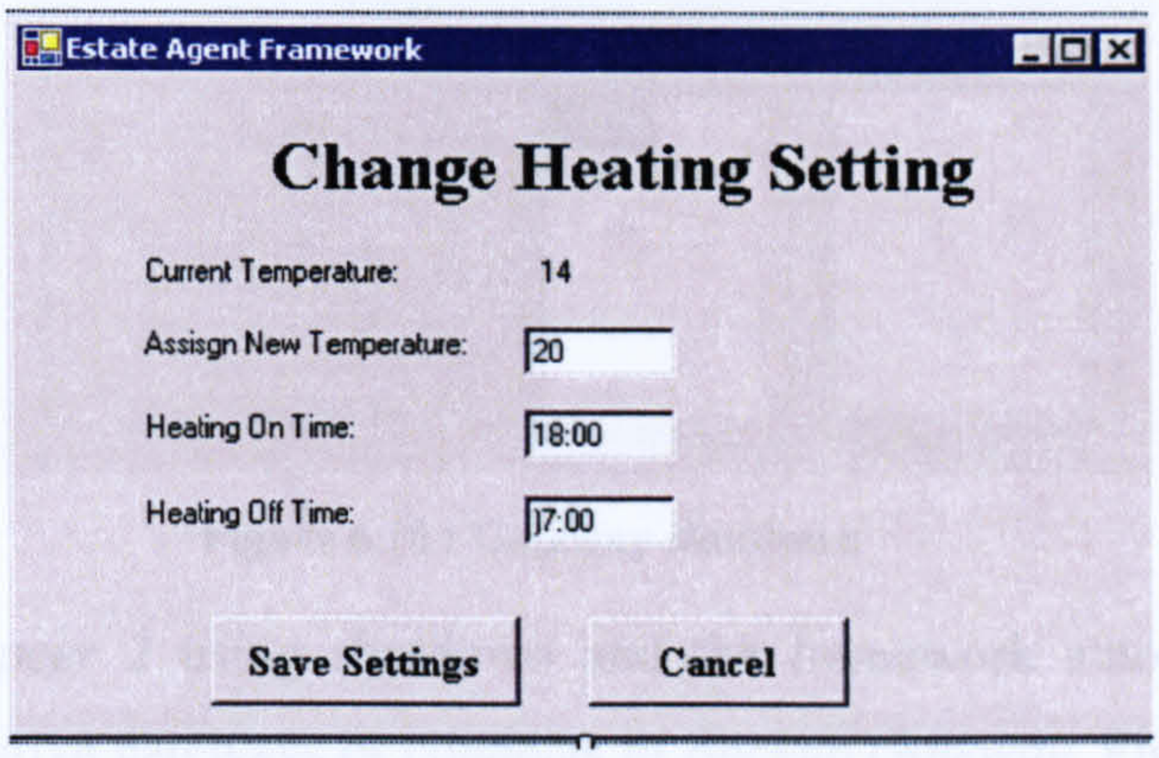

Figure 6.16 : Heating Settings Change

6.2.9 Gateway Failure

As we mentioned in chapter 4, a novel aspect of our research is to provide an alternative gateway in case of failure of the main gateway without losing any session data. In our prototype, *peer 2* is acting as *Active Gateway* and we also have *Backup Gateway* such as 150.204.49.201 as shown in Figure 6.17.

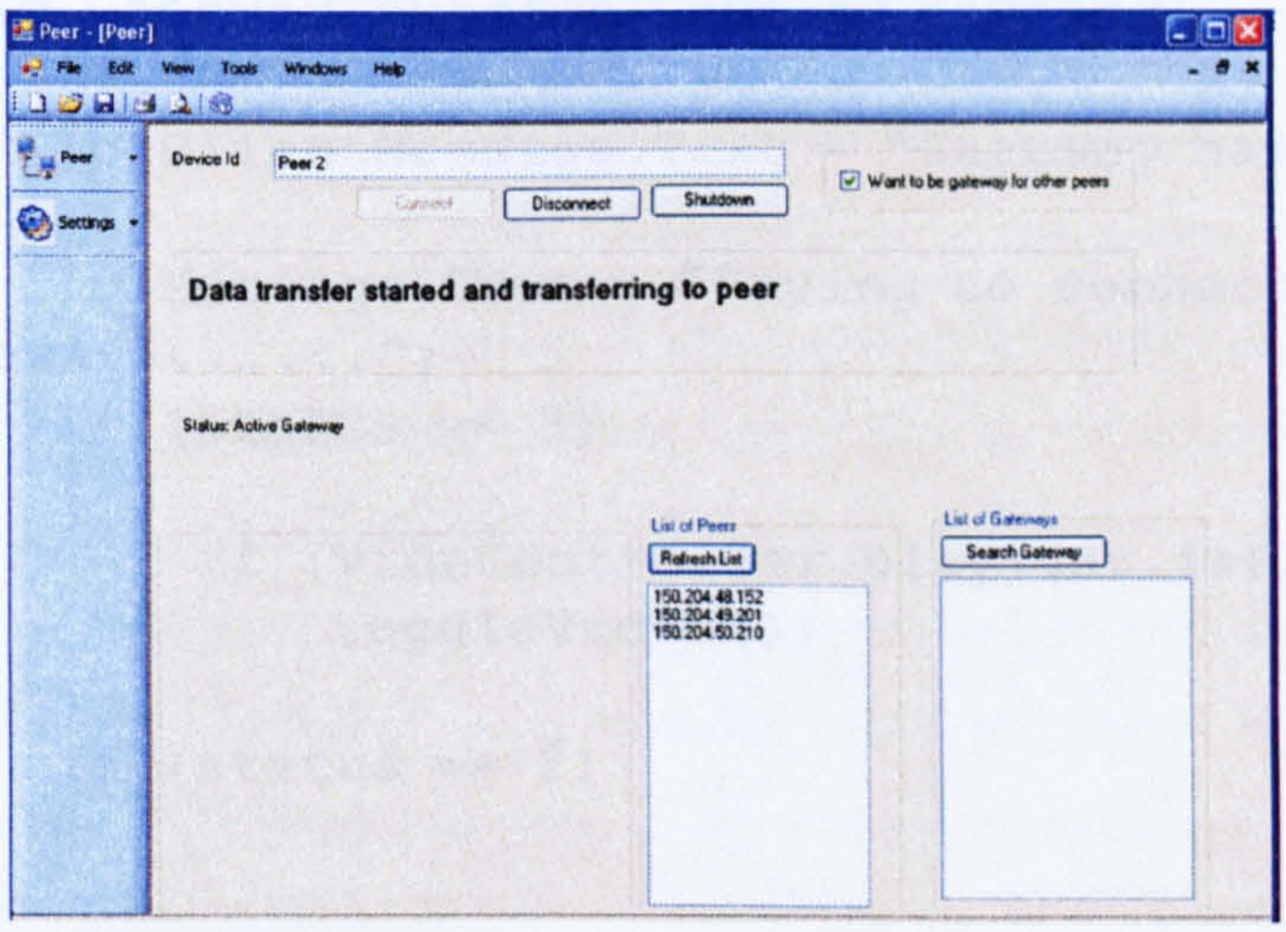

Figure 6.17 : Active Gateway

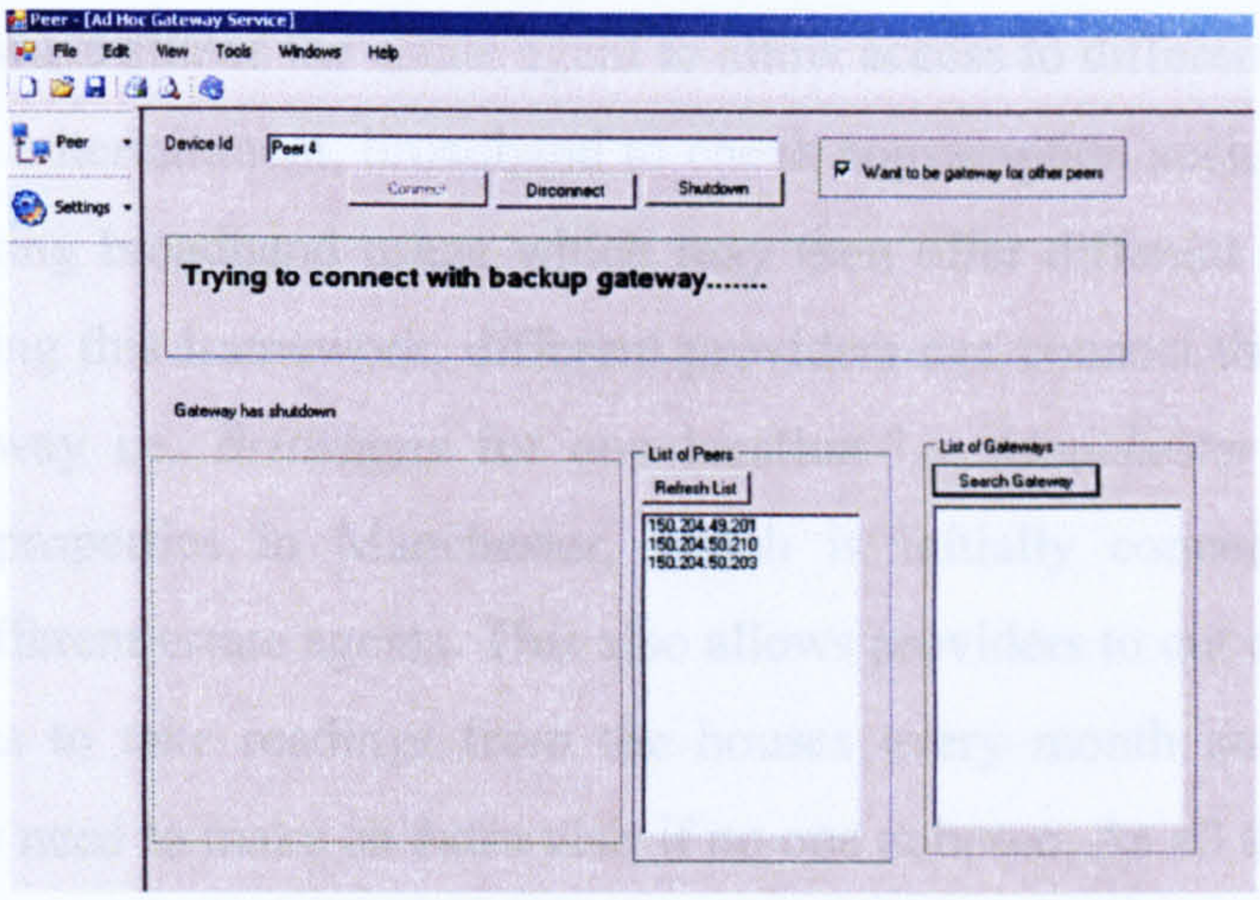

Figure 6.18 : Gateway shutdown

We shutdown *peer 2* using shutdown and the framework automatically starts searching for alternative gateways or connects to *Backup Gateway,* Figure 6.19 shows code for finding the backup gateway.

```
case
(Proshot.CommandClient.CommandType.GatewayShutDown):
                    GatewayShutdown(e.Command);
                    break;
.............

private void GatewayShutdown(Command command)
        {
            lblClientMessage.Text = "Gateway has
shutdown";
            lblMessage.Text = "Trying to connect with
backup gateway.......";
            if (status == 3)
            {
                if (VideController.playlist.isPlaying)
                 toggleVideo();
            }
            if (status == 2)
            {

                toggleBackupTimer(true);
            }
    }
```

Figure 6.19 : Finding backup gateway

Using gateways allows the estate agent to allow access to different providers such as gas, electric, entertainment, broadband to check consumption such as taking meter readings, checking broadband usage which may then offer different packages to cut down bills. Using this framework, different providers can connect these gateways to one main gateway i.e. *Britishgas* for one location i.e. *Manchester* allowing them access to all properties in Manchester, which is initially connected with other gateways by different estate agents. This also allows providers to cut down the cost of sending persons to take readings from the houses every month and safe times as sometimes they need to make an extra visit if no one at home. As all the houses in the estate agent's portfolio are connected to one gateway it can easily implement simple security access. When providers need to access this gateway as other devices may be connected to the gateway but an estate agent can only allow access to specific devices e.g. gas meter. This can be achieved using our Security Manager which authenticates any access to the peers. This can be done by allocating username/password to the providers and local agents.

To explain further, consider a situation when an estate agent needs to send an engineer to one of the properties in Liverpool for some maintenance work. The estate agent is not sure about the visit timings so it might be possible that no one is at home to open a door and to rearrange the visit might cost money and be very time consuming. A preferable situation would be if, when the engineer arrives at your doorstep, a message were sent to your mobile phone or messenger indicating its arrival. Using the audio and video system at the property the estate agent could then talk to and watch the engineer. Using a secure connection the estate agent can verify the engineer by asking for his credentials e.g. Person ID. By composing services such as audio and visual in this way, the estate agent can communicate with the engineer. If we consider the home environment we have audio, visual and image devices such as speakers, TV, and video/security cameras. Using gateway services we can compose services in order to allow audio-video conversation. In the case of office spaces, we have a range of devices at our disposal such as desktop computers with audio/video devices. In the case of public spaces, we have mobile phones and PDAs. Camera and touch screen phones are now commonplace.

Consider the entire home devices connected to *Liverpool* gateway Figure 6.7. When the engineer arrives, the home security system can check for any expected engineer visit which the user feeds into the security system and also registers that he should be informed about engineer arrival. To do this the user would send a message on his or her mobile phone or via his or her PC. As we mentioned in chapter 5, all devices register with the SM. As the user expects the engineer, so the home security system sends a message to the user. The user then logs in to the home security system using the *Liverpool* gateway to check the home security camera in order to verify the engineer. When the engineer arrives at the address, the GatewayS is discovered i.e. *Liverpool*. Using the engineer's Pocket PC as a user interface he can communicate with the *Liverpool* gateway and the rest of network. When the engineer arrives, information is sent to the estate agent on his mobile phone. This is achieved using communication between the *Liverpool* gateway and estate agent's mobile phone. On successful authorisation of the engineer, the estate agent can then login to home security system of the property to deactivate and allow access.

In our prototype, after peers have successfully connected to the gateway as shown in Figure 6.17, the gateway broadcasts a message for the video service locates *peer 1* offering this service. As in Figure 6.12, the lists of services are displayed that offer a video service. Figure 6.12, shows *peer 2* acting as an *Active Gateway* which is transferring data from source *peer 4*. The video stream starts playing on *peer 1* as shown in Figure 6.20.

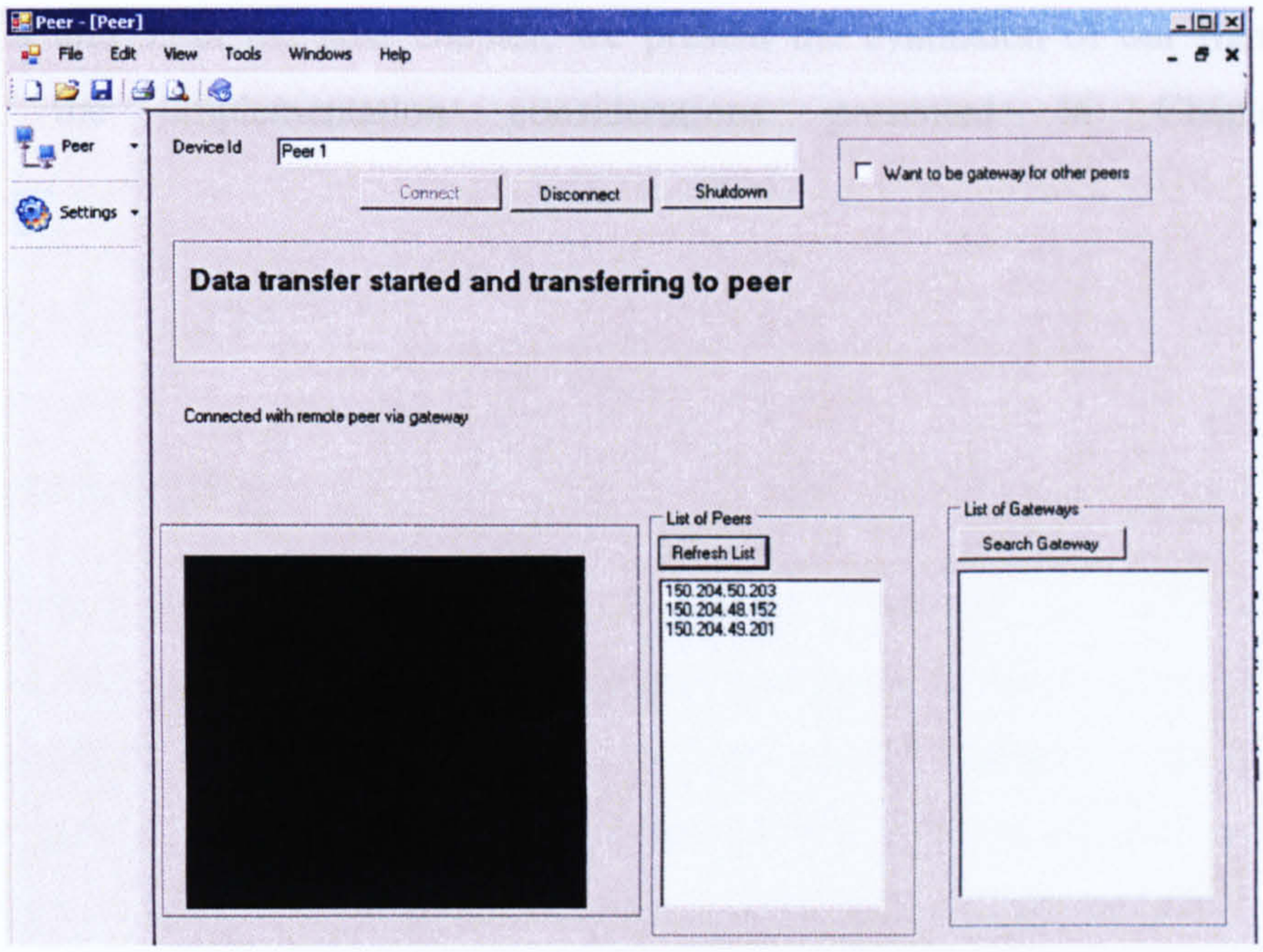

Figure 6.20: Running Video Service

Another interesting scenario was also implemented, called Package Delivery Framework, which we used to show our results [Muhammad 2007].

6.3 Summary

In this chapter we described the implementation of the AdHocGS framework prototype to demonstrate the working of our framework in a distributed P2P environment. As we discussed in chapter 5, we designed our framework assuming Service Manager is available in the network. In our prototype we used .NET framework to illustrate our prototype in distributed P2P by using web services in the .NET framework. The prototype implementation gave us a proof of concept of our framework design. This prototype also helped us to evaluate our framework design explained in detail in the following chapter.

We created two prototype test environments – the AdHocGS Framework to proof concept of our framework design and the estate agent scenario to asses our test scenarios. These test scenarios helped us to illustrate our framework and integration testing of our overall working of the prototype. These test scenarios also involve various smaller tests to prove our concept and working of our system designed in

chapter 4 and 5. In the next chapter, we present the evaluation of our framework against the implementation considerations presented in Chapter 4.

CHAPTER 7

7 EVALUATION

7.1 Introduction

Chapter 4 describes the requirements needed to address the issues we have identified with current Networked Appliances and home network approaches. The requirements detail what is needed to enable Networked Appliances to automatically discover gateway services and in case of failure of main gateway, alternative gateway discovery to resume sessions without losing any information. Gateways offer services such as Service Manager (SM), Security Manager (ScM) and Quality of Service Manager (QoSM). Gateway not only allows peers to locate other services in the local network but also communicate with gateways in the Gateway Peer Overlay Network discovering services not offered by any device in the local network.

7.2 AdHocGS Framework

The key requirements to provide in the framework are open standards, robustness and zero configurations in terms of ease of system. Our framework ensures that functionality is readily available in the network via service replication. As in standard P2P services such as file sharing where files are distributed, shared and discovered within P2P network our framework adopted the same principle but in our case gateway services are replicated. This means that in case of failure of main gateway or service alternative service may be available within the network that can be discovered and used. This makes our framework more robust and fault tolerant, ensures availability of alternative GatewayS Peer within P2P network.

The level of flexibility ensures that our framework allows peers to discover other services within the P2P network offered by other peers. Other research like OSGi [Redondo 2007], UPnP [UPnP 2006], DLNA [Venkitaraman 2007], HAVi [HAVi 2004] are mainly using standards such as Web Services Flow Language (WSFL)

[Leymann 2001] and Business Process Execution Language for Web Services (BPEL4WS) [Curbera 2006]. As long as all the services available within operations reside in same location they remain reliable, but if any service changes in any way such as becoming unavailable or moves location then all the operations have to start from the beginning and all session data is lost.

In our case an alternative gateway or any other service would be automatically discovered without losing any session data. Our framework differs from others in its ability not only to discover and use secondary GatewayS peers which are pre-determined but also keep track of any alternative service leaving the network. Our framework keeps up to date in regards to GatewayS peers available in the network and mirror information in an *active gateway*, in the presence of SM in the network services will be updated if a gateway leaves or joins the network. In the absence of SM, if any gateway leaves the network, messages are sent to all the peers in the network. This function is important as our framework keeps a cache of the GatewayS peers used before and a service needs a gateway it checks its cache. When a gateway leaves the network it broadcasts this message so that peers remove gateway information from their caches. As mentioned in chapter 4, all components of AdHocGS framework running as service such as any peer within network might offering SM, others offering ScM and QoSM which makes our framework more reliable by utilising existing services. But the downside of this environment is the more ad hoc nature and therefore no control can be placed over how and what services are hosted in P2P network. A unique feature supported by our framework which we demonstrated in the implementation is the ability to automatically discover gateway or other services with the network without any or less human intervention. We have also extended the concept around P2P; we not only focus on content sharing but also distribution and sharing services. We have clearly made a novel contribution within this area and demonstrated how P2P can be used to extend NAs and home/office networks.

To our knowledge our framework is the first to use P2P techniques to discover and use ad hoc gateway services. Our GatewayS peer can communicate with other gateways in a Gateway Overlay Network. Gateways in the overlay network create a cloud of gateways directly connected with each other in a P2P fashion. We have

demonstrated via our prototype, our key contribution described in this thesis. As mentioned in earlier chapters all components of our framework run as services. In the rest of this chapter we discuss to what degree we achieved our project requirements.

7.3 Our Overlay Network of Gateways

P2P network forms a logical layer over the Internet called an *overlay,* the underlying physical connections between Internet nodes are not necessarily the actual structure of the P2P network. Routing mechanisms used by these peer systems utilise the Internet as a transport medium but may have their own routing protocols independent of or working over the Domain Name System (DNS). KaZaA is file-sharing networks that uses super-nodes for assisting indexing of frequent request to enable faster search; formation of an overlay network of super-nodes which function within KaZaA network provide benefits to the entire system. Our concept of a Gateway Peer Overlay Network is inspired by the concept of super-nodes and overlay networks. These overlays to provide an extra functionality to the P2P system without changing the underlying layers. The main functionality of the overlay network proposed in this thesis is to act as a service-offering or service-sharing network. In our case, a gateway is not only offering services but also sharing services offered by other gateways. Our gateway not only allows users to compose different services in a gateway but also offering some core services. Our gateway can also request services offered by other gateways in the overlay network. When requested services are not offered by any peer in the local network, GatewayS peer then broadcasts a request for the service. Any gateway offering the requested services can be used by requesting services. By using Gateway Overlay Network, gateways can locate services by broadcasting requests on the overlay network.

7.4 AdHoc Gateway Service

A gateway itself is composed of different services such as Security Manager (ScM), Service Manager (SM) and Performance Analyzer (PA). These are known as the core services that allow the gateway to perform security management, Quality of Service analysis and device capability matching. When all the required core services are discovered and bound together the gateway can be used. All these services may be offered by different devices in the network. If any of components fail, alternate

service can be discovered and composed into a gateway. As a number of devices may be offering a gateway service, one gateway is assigned as active gateway while others can be assigned as backup gateways. All backup gateways replicate active gateways i.e. services connected to the gateway, data transferred etc. In case of failure of active gateway alternate gateway service can be discovered without user intervention and start communication where the active gateway fails. All the gateways also synchronise on the Gateway Peer Overlay Network, enabling gateways on the overlay network to easily locate services and also if any gateway stopped, all gateways on the overlay network are also updated.

7.5 Device Capability Matching

In a P2P network, a number of peers may be offering the same service. When Service Controller receives a request for a particular service, it searches for the best possible service. As we discussed earlier, on registration along with other information, it also registers its software/hardware capabilities with Admission Controller. The proposed framework only captures basic information such as memory, CPU speed and screen resolution. Each device publishes these capabilities in case no service management exists. This allows other peers to first determine if it can effectively execute requested services, and this information added to the Service Controller. This feature can only be implemented on specialised Networked Appliances because simple Networked Appliances do not offer this service. When a peer requests a particular service, Device Capability Matcher checks for the best possible service that can execute peer request. When Service Manager receives a request for a service from a peer, it may result in several services offering the same functionality. It is possible that a number of devices in the network are offering same services i.e. video service. A computer monitor or HD TV might be offering a video service. It may be possible to stream video on a computer monitor but the best solution is HD TV. In this situation, when a device requests a video service, HD TV might be not available in the network and the computer monitor is the best available service but once HD TV becomes available, the video should be streamed to the new service. In our framework, we developed a mechanism called Device Capability Matcher that allows devices to automatically determine the best device to execute services. In the above case, Service Manager matches the device capability requirements using Device

Capability Matcher to find the best video service to stream the video data. On the basis of device capability requirements, Service Manager checks for the best service in case more than one service is offering the same functionality. Our framework searches for services within the network, locates available services and uses the best one while our Service Manager will still keep track of the other devices offering the same service and once the best service becomes available automatically i.e. route all the information.

7.6 Evaluation of design challenges

In this section, we discuss the design challenges mentioned in chapter 4 and how we met them.

7.6.1 Naming and Address

Since the location of the physical device may continually change, we assigned every device a unique name and address. When the device first connects to the P2P network, it registers its service with Service Manager(SM). As Figure 7.1, SM itself consists of three further components. One of the components Admission Controller (AC) assigns unique name and address. These details are unique to a particular device, so even if a device changed its physical location within the network it can still be discoverable by other devices. The name assigned to the device helps peers to know the device but at the network level address is used to route information. When a peer requests a list of peers within the network, a list of IP addresses is displayed. These IP addresses are used to route data to and from the device.

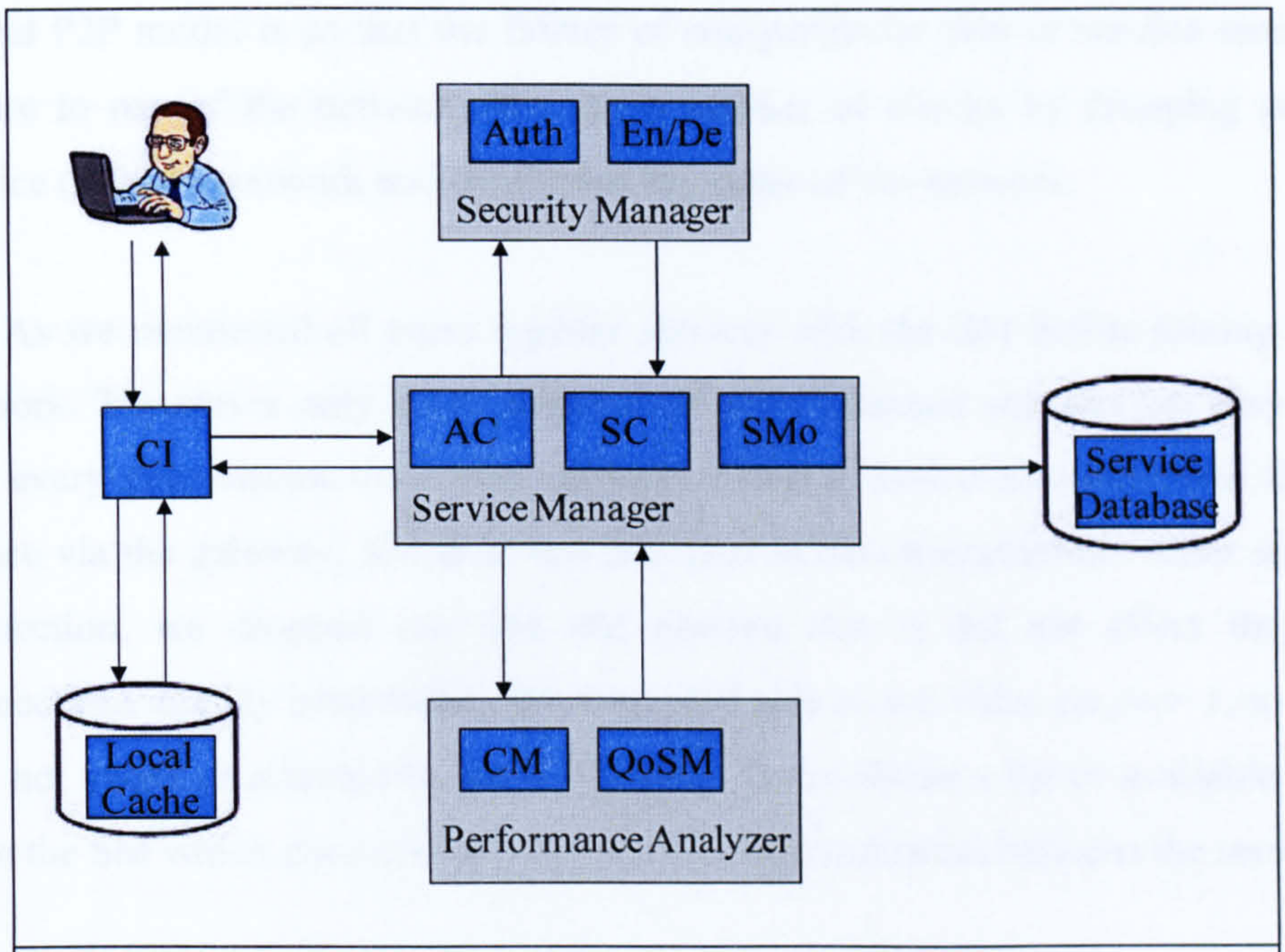

Figure 7.1: AdHocGS Framework

7.6.2 Decentralisation

As we mentioned there are different types of P2P network according to degree of centralisation, i.e. Centralised, Pure and Hybrid P2P model.

Hybrid P2P has central servers to keep information about peers and respond to certain information requests. Peers are responsible for hosting their own services, because the central server does not keep this information. Servers only know what services they want to share and to make these services sharable/available for other peers that request them. In our prototype SM only registers services on their arrival in the P2P network. SM itself does not perform any role in transmission of information to and from peers. In our case, when a peer requests a gateway service, SM checks for peers offering this service. If found, a list is given to the peer to communicate with that particular peer, if not found the peer will then broadcast an advertisement. Upon successful connection with gateway service, SM is not responsible for the data transmission between other peers within the network. One of the reasons we used a

hybrid P2P model is so that the failure of one particular peer or service won't bring failure to rest of the network. We ran a number of checks by dropping particular service (s) in the network and monitored the status of the network.

As we mentioned all peers register services with the SM before joining the P2P network. The server only keeps a record of peers attached and services they offered and every peer shares their own services. Once a device connects with the other device via the gateway, SM does not take part in data transmission. After successful connection, we dropped our SM and noticed that it did not affect the rest of connections already established. We were still able to see video on *peer 1*, so the user was not aware or affected by the SM failure. Peers obtain a list of available services from the SM which does not take any part in communication between the services.

In the second case, we tested that the SM is working in the network, both devices connected via an active gateway and users were able to continue receiving video. We dropped the active gateway and transferred all control to a backup gateway. In this case, the end user will notice some pause in the video because of control transferred to the new gateway. In the third case, all devices are connected via a gateway service. While the video was streaming from *peer 3* to *peer 1*, we dropped our SM which did not affect connections already established but no more devices can connect to the network, even though currently connected users would not notice the failure. In order to check flexibility of our framework we dropped the active gateway and control automatically transferred to the backup gateway. As no SM is available in the network, any devices that request for a gateway service need then to broadcast their request over the network. Devices that cache information about the gateway used are able to access a gateway as long as these peers are still connected to the device, which cannot be guaranteed due to nature of P2P network as we don't have any control when devices connect or leave the network.

In the fourth case, we again dropped our SM as well as our active gateway. In this case it did not affect an end user watching a video but there were pauses in the video. We dropped our backup gateway as well, which transfers control over to last gateway available. When the first active gateway is shutdown and the backup starts working, a

request is sent to the SM to the find next gateway. As the SM is also shutdown, the request needs to broadcast a request for gateway service.

In order to overcome a failure of SM, we introduce alternative SM within P2P network, which replicates the active SM as we did in the case of gateway services. We are replicating SM, which takes over control in case the active SM failed. Due to amount of data SM has to store, we use peers with more storage i.e. fat peer. In our implementation, we only cover to provide first alternative SM.

7.6.3 Platform Independence

In a P2P network we do not know in advance what operating system peers are using such as Microsoft Windows, Mac OS or Linux. However we implemented our prototype use .NET that is based on Microsoft Windows™ XP.

In order to run this prototype on different platforms or peers within the network using different platforms, we used *Mono* [Novell 2008] which provides the necessary software to develop and run .NET client and server applications on Linux [Babcock 2007; Linux 2007], Mac [Solaris 2008], Microsoft Windows™ and Unix[Unix 2008]. Features of Mono includes:

- Multi-platform compilation
- Based on ECMA/ISO standards[ECMA 2006]
- Can run .NET, Java, Python, ASP.NET and Winsforms applications
- Open source, free and commercially supported

Mono is an open development by Novell to develop UNIX version of the Microsoft .NET development platform. Using Mono, UNIX developers can build and deploy cross platform .NET applications. Mono contains a number of components for building new software.

- A Common Language Infrastructure (CLI) [Libby 2007] virtual machine contains a class loader, Just-in-time complier (JIT) [El-Kadri 2006];

- A class library that can work with any language, which works on Common Language Runtime (CLR) [Schmied 2007]. Both .NET and Mono-provided compatible class libraries are included;
- A complier for the C# language.

7.6.4 Device Capability Matching

We used a Device Capability Matching [Matsubara 2007; Muhammad 2007] service in order to check the hardware and software capabilities of a device. This service is used to determine how effectively we can use the available service to perform an operation. In a P2P network there is always a possibility that one service can be offered by a number of devices. For example, different devices such as monitor, television etc, may be offering a video service. Our framework searches for services within the P2P network, locates available services and uses the best one. Figure 4.9 show our algorithm for implementation of device capability matching. In our implementation, when a peer requests a video service a number of devices reply and the gateway checks the available video service for best performance, such as screen resolution. In case all services are the same, the first available is used.

As we discussed in Chapter 4 our Performance Analyser (PA) ensures that enough resources are available before carrying out any operations. The main component of PA is Capability Manager (CM), working in conjunction with the Quality of Service Manager (QoSM). CM checks the hardware and software capabilities of the device to perform the requested service, for example, screen resolution, memory and the software installed.

In addition to the basic requirements listed above, a few other requirements – as listed below – are beneficial in order to implement an ideal system. We addressed these requirements in our design, though we did not implement them fully. However we have been careful to ensure that it would not affect operation of the rest of framework.

7.6.5 Security

Security is an important component of any computer system. P2P networks are gaining considerable attention today so security is one of the most important concerns. In this section, we discuss security issues within P2P networks. Organisations implement different levels of security depending upon requirements. Important factors in implementing security are determined by what we need to protect and against. Important considerations are connection control, access control, antivirus and most important data stored on the computers. Connection controls are of key importance as if we can keep our connection secure it makes it difficult for hackers to damage or steal the data [Palomar 2006]. Access control, can be dealt with by implementing security policies, for example, by grouping together persons who can access these type of information.

As mentioned in chapter 4, we have proposed a module to implement security within our framework. Levels of security will vary depending on user requirements. We mentioned earlier that security is not the main focus of our research but we did implement basic security within our framework such as to authorise user access to a particular service within P2P network and encrypt/decrypt data. A number of solutions designed to make P2P network more secure such as authentication, access control, trust etc, some notable researchers are [Detsch 2006; Kumar 2006; Locasto 2005].

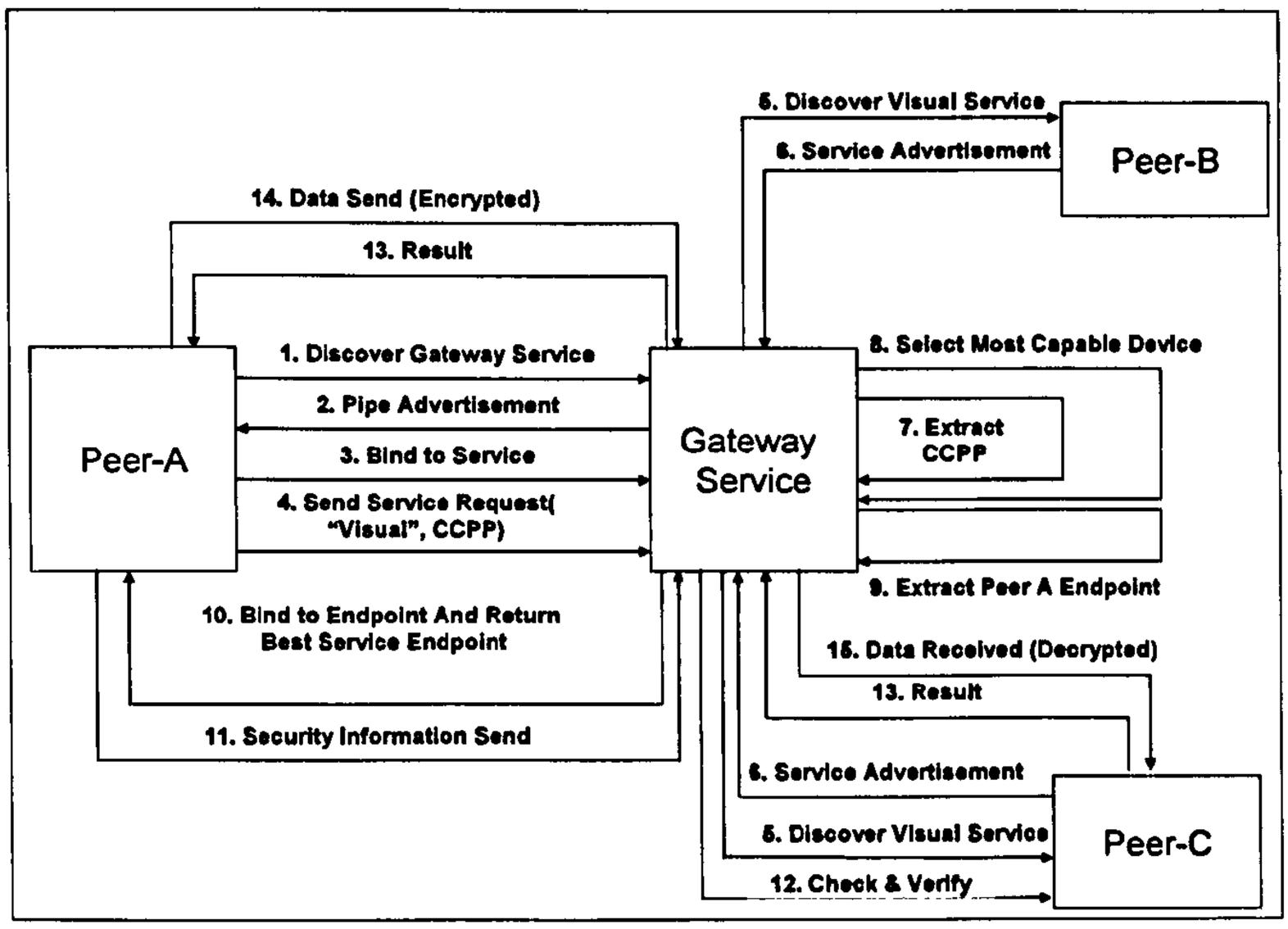

Figure 7.2: Security Algorithm

As mentioned in chapter 5, services can request secure login so that only authorised peers can use this service, which peers need to provide SM when registering its services. Whenever any peers request this particular service they should authorise using username and password. Upon successful entry for username and password access will be granted. Figure 7.2 shows our algorithm for security implementation.

7.6.6 Quality of Services

QoS refers to the set of parameters used to specify the resource requirements in the network [Gmach 2008], including the delay or time taken for packets to arrive at a destination. It may also include security of the network. Before carrying out any operation, we need to check if all the services are available and able which may also include the cost to carry out an operation.

QoS is the ability to provide priorities to different application or operations, which guarantees certain levels of performance in the network. For example, QoS guarantees are important if the network is running on insufficient capacity, especially for real-time applications such as VoIP, online games [Rieche 2007] etc. Before carrying out any operation within the network we need to make sure whether it going to be feasible or not, for example, if we need to exchange a video which requires more

bandwidth, QoS tries to ensure that enough bandwidth is available for it, and if not, postpones the request until enough resources are available. A network or protocol that supports QoS negotiates with an application about data transmission such as amount and type of data and may reserve capacity in the node. During transmission it monitors the achieved level of performance such as data rate. In a distributed P2P network a service might be accessed by a number of peers at any one time, which can cause network blockage or implementing priorities in the network. In our P2P network congestion control and priority based scheduling is needed for QoS.

As we discussed in Chapter 4 our Performance Analyser (PA) in Chapter 4, ensures that enough resources are available before carrying out any operations. One of the main components of PA is QoS Manager, which is responsible for providing a consistent, predictable data delivery service.

We included this module in our design in order to make an ideal system, but in the above definition of QoS, it is beyond our research scope. This module can be implemented with user requirements such as level of data rate per sec. QoS very much to do with the network side of OSI model, which is not our project scope. A number of solutions been designed such as [Núñez 2006] who proposed Extended Service Discovery Protocol (ESDP) which allows discovery of services through queries to the network, propagating using "Sensible Routing". ESDP allows better performance in respect to search time, high probability of success, minimum overhead and improves received QoS. [Magharei 2007] proposed a solution for streaming of services in live P2P to residential users.

7.6.7 Trust Relationship

Trust is one of the most important bondings between human beings, we are all aware of its importance. Due to nature of decentralised, self organised network such as P2P or ad hoc network, it is difficult to establish trust [Yajun 2007]. One of the main reasons is that these networks do not follow any fixed infrastructure to communicate and do not have enough information about nodes in advance. On the basis of trust, peers create a network to exchange information, which create decentralised, personalised "webs of Trust" [Chen 2005]. In the P2P network, peer

groups and peer memberships are dynamic and typically they do not implement centralised security mechanisms, therefore a level of trust cannot initially be determined [Chen 2005].

In P2P or ad hoc network, it is not feasible to use identity certificate [Funabiki 2007] to establish trust relationships as peers do not know each other and use different security policies. As in traditional networks, system security depends on trusting the third party to provide authentication and key management [Funabiki 2007]. There are a number of researches on building trust relationship in such networks [Hoffman 2006; Pirzada 2006; Sun 2006; Yan Lindsay 2006]. Building trust relationship within such networks enables secure communication among peers.

7.7 Comparison with existing Approaches

In this section we compare our framework with some existing Networked Appliances and home networking approaches. We use our novel contributions as a basis for our comparison such as Service-Oriented Networking, Service Discovery, Device Capability Matching, Dynamic Service Composition and Service Failure as shown in Table 7.1.

7.7.1 Universal Plug and Play (UPnP)

Service-Oriented Networking: UPnP is a Service-Oriented Architecture and provides mechanism to disperse device functions in the same way our framework does [Jakab 2007]. However the main limitation of UPnP is that one cannot access service outside a local area network. As our implementation shows our framework can communicate across a global network as well as locally in a home or office. All the communication in UPnP happens over Internet Protocol (IP) [Lee 2007], a target must obtain an IP address before it can join a UPnP network and by using IP addresses, a control point can contact other UPnP devices within same subnet [UPnP 2006]. Messages within UPnP are sent using SOAP [Louridas 2006].

Service Discovery: UPnP uses Simple Service Discovery Protocol (SSDP) [Wu 2007] to discover services in the network. That allows pre-determined services such as printers and scanners to be discovered by using matching attribute-value pair. In

UPnP specifications SSDP does not accept advanced querying, so it is a major limitation of UPnP that service descriptions and service requests must be pre-determined and in a format defined by SSDP specifications. If attribute-value pairs differ syntactically but are semantically the same then service discovery fails. In our framework, we don't follow any pre-determined service description. When peers require any service they send a service request, including device capability if looking for any particular device description, otherwise service name such as audio, video. Usually this method allows more services matched within our framework than UPnP.

Device Capability Matching: In UPnP devices provide a URL [Matsubara 2007] which points to a UPnP description used to describe the device and services it provides, information about the actions and state variables. When a device receives a request for device description information it replies with device description which can be extracted by control points upon device discovery using this URL. This description usually describes high-level information about the device rather than the individual properties. Due to this it is not possible to determine how resourceful the device in terms of memory and processing power. It makes it difficult to automatically determine the best device or service available in the network. In our framework we used Composite Capabilities/Preference Profiles (CC/PP) specifications [Mahmoud 2007; Matsubara 2007], which allows devices to select the best devices available is in the network. A UPnP specification does not provide this feature.

Dynamic Service Composition: In UPnP there is no mechanism to address dynamic service composition, services are manually discovered using a user interface, and no mechanism that allows devices to automatically discover services and compose them. In our framework we overcome this limitation by allowing devices to query the network for services and automatically compose with other devices within our network. This feature not supported in UPnP.

Service Failure: In UPnP there is no mechanism to discover alternative service in case of failure of the first one. If a service fails then the whole solution may fail. Also if a control point within a UPnP specification fails, the whole network may fail. In our framework, if one service fails, an alternative service can be discovered and

composed with other services to maintain the best Quality of Service. Also if our main gateway fails, an alternative gateway is discovered.

Table 7.1 : Comparison Table

	Service-Oriented Networking	Service Discovery	Device Capability Matching	Dynamic Service Composition	Service Failure
UPnP	☑	☑			
OSGi	☑	☑			
DPWS	☑	☑			
AdHocGS Framework	☑	☑	☑	☑	☑

7.7.2 Open Service Gateway Initiative (OSGi)

Service-Oriented Networking: OSGi is another example of Service-Oriented Architecture [Kumaran 2007]. OSGi service providers host services in service containers controlled by a service operator. These services are then served via the Internet to home networks using OSGi gateway. OSGi uses a traditional centralised approach similar to the set-top box solution. In case of failure of the central service provider the whole network will fail. In our framework, we use P2P technologies to use Service Manager to register services but always provide backup a Service Manager in case of failure of a Service Manager.

Service Discovery: OSGi provides service discovery mechanisms that allow services residing within the OSGi Service Platform to be discovered. Discovery is based on searching for services with pre-determined properties. A Simple query language is used to select required services. UPnP services need to be described using predetermined vocabularies. As such discovering services that are syntactically distinct but semantically the same results in failure. Our framework provides a more advanced service discovery mechanism that allows devices to discover service within global network using P2P technologies.

Device Capability Matching: The OSGi specification does not address capability matching. Within OSGi service usage is manually performed by the service provider,

service operator and the user. In our framework compositions are created on basis of best solution available within network. In our framework, Capability Manager provides the service that enables peers to determine the hardware and software capability of the service before the peer uses it. If a particular service is not available, the device uses the next best service available. This feature is not provided by OSGi, which is an important feature for services that reside within P2P.

Dynamic Service Composition: The OSGi specification does not provide any mechanism to dynamically compose services without any human intervention. Our framework provides zero-configuration or at least low human intervention. When a peer first requests a service, they can select available services in the network but if any particular service fails or leaves the network, it can automatically rediscover alternative service without human intervention. OSGi specification does not provide this feature.

Service Failure: There is no mechanism in OSGi to rediscover alternative service as service configurations are manually created. Composition remains operational as long as all services within composition remain operational, any faults need to be corrected manually. In our framework alternative service is automatically discovered in case of failure in composition and plug in without any human intervention. Again an OSGi specification does not provide this feature.

7.7.3 Devices Profile for Web Services (DWPS)

Service-Oriented Networking: A Devices Profile for Web Services (DWPS) is an another example of Service-Oriented Architecture [Microsoft 2008], and defines architecture similar to UPnP but is fully aligned with Web services technology and includes numerous extension points which allows for seamless integration of device-provided services. DWPS was initially developed by Microsoft and some printer manufacturers to send and receive secure messages from web services. Similar to our framework devices run two types of services, hosting services and hosted services, allowing them not only to publish services but discover other available services.

Device Capability Matching: In DPWS service discovery uses several steps to communicate with the client. In DPWS if client wants to obtain metadata/description about device, is that it needs to send an extra message. In our framework, devices register all information with the Service Manager (SM) which peers obtain with service advertisement. In our framework, SM makes sure that the requested service can match the capabilities required. Capability Manager provides the service that enables peers to determine hardware and software capability of the requested service before the device uses it. If a particular service is not available, a device uses the next best service available. This feature is not provided by DPWS, which is an important feature for services that resides within P2P.

Service Discovery: In DPWS messages are protected using message level signatures and secure channels. Authentication is usually done using certificates or PIN/password exchange but there is a possibility of network eavesdropping through which an attacker can steal this information. Device discovery defined in DPWS may cause interoperability problems, may lead to clients being unable to locate all requested services [Zeeb 2007]. Length restrictions for message fields defined in message section may lead to interoperability issues as the client side considers restrictions sending message while device side could reject message that exceed the restrictions.

Service Failure: There is no mechanism in DPWS to rediscover alternative service as service configurations are manually created. Composition remains operational as long as all services within composition remain operational, any fault needs to be corrected manually. In our framework alternative service is automatically discovered in case of failure in composition and plug in without any human intervention. Again DWPS specification does not provide this feature.

7.8 Summary

Our framework has performed as expected and has demonstrated challenges discussed in chapter 1. The overall performance of our framework needs to be improved; however we successfully achieved enough to demonstrate our idea.

Our evaluation shows that our framework addressed challenges within networked applications and home networking. A number of approaches have been adopted by many researchers, most of these approaches use configuration that requires technical knowledge, lacked by most home users. If these devices are configured by providers this is not always cost-effective. Our framework gives independence to non technical home users to not worry about configuring devices and overcome failures of gateway service or any other service. The framework we developed gives zero-configuration, system robustness to home users.

CHAPTER 8

8 CONCLUSION AND FUTURE WORK

In this chapter, we discuss our achievements in designing the AdHocGS framework. We proposed the framework of a decentralised gateway service, which provides platform independence and enables devices from different vendors to communicate via this gateway. Our gateway service also supports security and performance in regard to QoS and Device Capability Matching. It also provides the mechanism to rediscover alternative gateway services in case of failure of active gateway. We demonstrated our idea by implementing a real world example of Estate Agent discussed in chapter 6. In the rest of this chapter, we present a thesis summary, state our novel contributions to the research area and suggested future work. We include the difficulties faced in studying this research area and suggest further improvements in the framework. We concluded with final remarks at the end of the chapter.

8.1 Thesis Summary

Chapter 1 provided an overview of our research area, which involves Networked Appliances and middleware for home networking. We identified research carried out in ad hoc home networking, which includes how to integrate devices and configure them together. We mainly pinpoint research carried out in the fields of service discovery, composition and human intervention in it. The chapter discussed some background work such as Ubiquitous Computing, Networked Appliances, Gateways, P2P networks. The chapter then briefly introduced our results. Our research is mainly focused on providing an AdHocGS within P2P networks and provides an alternate gateway services in case of failure of the first one. The chapter concluded by outlining our research project aims and objectives and the novel contributions made.

In chapter 2, background and related work was presented. We started with some history of computer networks, from early history of Internet. We then moved to

network architecture such as different network topologies, wired/wireless networking. We then moved to Ubiquitous/Pervasive Computing and discussed some results and challenges in this field. The chapter also discussed Networked Appliances in relation to home networks. Some notable research work done in seamlessly interconnecting Networked Appliances within home networks was presented. The chapter discussed P2P networks, their merits/demerits and some challenges in this field. In the chapter we discussed some P2P models and how this integration is being performed using P2P techniques. We also discussed some well known P2P applications such as Napster, Gnutella, and KaZaA etc. Each P2P model was discussed in terms of functionality, limitations, structure, discovery and failure of particular service or device. The chapter concluded by examining security in P2P networks, the importance of security in P2P networks and how it is possible to achieve it.

Chapter 3, Gateways, is a further literature review chapter but we mainly focused service-oriented architecture middleware that is used to seamlessly interconnect devices within home networks. We discussed some well known SOA architecture middleware such as OSGi, UPnP, and DPWS etc. We mainly discussed these middleware in terms of their architecture, functionalities and address limitations in them. We found in these middleware solutions that discovery services are very limited as they are based on proprietary description of how services must be advertised and discovered. As our research is focused on providing AdHocGS in such environment, we also presented some definition about gateways.

Chapter 4 presented our AdHocGS framework requirements. To begin with we discussed the novelty of our research which arose from the analysis from chapter 2 and 3. We concluded from our literature review and the limitations within current middleware solutions, that we not only required ad hoc gateways which enable services to be advertised and discovered within global network but also provide an alternative gateway service in case of failure. We also concluded that in current middleware solutions failure of a particular service in composition resulted in failure of the whole composition. We presented two scenarios to explain our idea, which we later implemented in chapter 6. We discussed how to communicate with NAs with and without middleware. Later in the chapter we presented the project requirements.

At end of the chapter we discussed the main components of our design based on the project requirements.

In chapter 5 we discuss in detail the main components of our framework with the help of some UML diagrams. After explanation of our framework components we explain the communication between them. Using our design we conclude by explaining the novelty of our framework.

In chapter 6 we discuss our implementation, including how we implement these components to achieve our objectives. In previous chapters we discussed the novelty of our AdHocGS Framework, which provides Service Manager (SM), Security Manager (ScM), Quality of Service Manager (QoSM), gateway service in P2P network and in case of failure of main gateway provides alternative gateway service. In chapter 6, we presented a case study showing how we implement our framework. This chapter also includes the testing of our framework. We present a prototype of how to discover gateway service within P2P network and rediscovery of alternative gateway in case of failure of first one. In this chapter we talk about the tools used in designing our prototype.

Chapter 7 is concerned with the system evaluation, including application case study. We discuss the performance of our framework and usage of our framework in different situations. These help us to identify limitations and short comings of our AdHocGS framework.

8.2 Contribution to knowledge

In this thesis a solution we have named AdHocGS framework has been presented for providing gateway service to access networked appliances in P2P network. The challenges we have overcome in order to achieve this include: service-oriented networking, service advertisement, service discovery and composition, gateway creation composing number of services. We have addressed these challenges using our AdHocGS framework and made several novel contributions [Muhammad 2005; Muhammad 2007; Muhammad 2007]. Our framework provides services that discover and interconnect devices within the network, allowing devices to advertise their services and discover available services, compose different services into AdHocGS,

make secure access to the services and provide an alternative gateway service in case of failure of a main gateway. This section discusses our research contribution to the knowledge.

8.2.1 AdHocGS Framework

We developed the AdHocGS framework to offer open standards, robustness and zero configurations in terms of ease of system. Our framework ensures that functionality is readily available in the network via service replication. As in standard P2P services such as file sharing where files are distributed, shared and discovered within P2P network our framework adopted the same principle but in our case gateway services are replicated. This means that in case of the failure of a main gateway, or service, alternative gateway services may be available within the network that can be discovered and used. This makes our framework more robust and fault tolerant, and ensures availability of alternative GatewayS Peer within P2P network. One of the reasons to implement as a framework is to understand how different services can be integrated together and also allows flexibility for future changes depending on user requirements, as well as seamless integration of functionalities while remaining robust to one or more service failures.

8.2.2 Overlay Network of Gateways

Our concept of a Gateway Peer Overlay Network is inspired by the concept of super-nodes and overlay networks. Using these overlays we provide an extra functionality to the P2P system without changing the underlying layers. The main functionality of the overlay network proposed in this thesis is to act as a service-offering or service-sharing network. In our case, a gateway is not only offering services but also sharing services offered by other gateways. Our gateway not only allows users to compose different services in a gateway but also offers some core services. Our gateway can also request services offered by other gateways in the overlay network. When a requested service is not offered by any peer in the local network, GatewayS peer then broadcasts request for the service. Any gateway offering requested services can be used by requesting services. By using Gateway Peer Overlay Network, gateways can locate services by broadcasting requests on the overlay network. Our gateway is therefore not only offering its own services but can

also request and use other services offered by other gateways in overlay network. Using this technique, it not only enables us to create a personalised gateway by connecting our home or office devices and access them via the Internet but can also enable specialised gateways only offering a specific set of services e.g. video or audio services.

8.2.3 AdHoc Gateway Service

A gateway itself composes different services such as Security Manager (ScM), Service Manager (SM) and Performance Analyzer (PA). These are known as the core services that allow the gateway to perform security management, Quality of Service analysis and device capability matching. When all the required core services are discovered and bound together the gateway can be used. All these services may be offered by different devices in the network. If any of the components fail, alternate service can be discovered and composed into a gateway. As a number of devices may be offering a gateway service, one gateway is assigned as active gateway while others can be assigned as backup gateways. All backup gateways replicate active gateway i.e. service connected to the gateway, data transferred etc. In case of failure of an active gateway alternate gateway service can be discovered without user intervention and start communication where the active gateway failed. All the gateways also synchronise on the Gateway Peer Overlay Network enabling gateways on the overlay network to easily locate services and also if any gateway stopped, all gateways on the overlay network are also updated. This features enables our AdHocGS Framework to overcome the single point failure problem in existing solutions such as OSGi, UPnP, DPWS etc by recovering alternative gateway service in the P2P network with less or no data loss. In P2P network, a number of peers may be offering the same service. When Service Controller receives a request for a particular service, it searches for the best possible service. This allows other peers to first determine if it can effectively execute requested services, this information is added to the Service Controller. In our proposed framework, when a peer requests for particular service, Service Controller check for the best possible service that can execute the peer request as Service Manager receives request for a service from a peer, it may result in several services offering same functionality.

8.2.4 Gateway Replication and Synchronisation

In our proposed framework we replicated our active gateway and synchronised all gateways on Gateway Peer Overlay Network between Active and Backup gateways. Data needs to be regularly duplicated across the distributed network to ensure data consistency and improve system performance. Replication in P2P is necessary for more data availability, so in case of failure of one peer, data can be obtained from another peer. As discussed earlier when more than one peer replies to SM GatewayS peer requests, one becomes an Active gateway while others become Backup gateways. Replication is necessary to replicate all information from Active gateway to Backup gateways. We replicate data across the Gateway Peer Overlay Network to ensure persistence of GatewayS data. Data replication is a method where Active gateway regularly sends data to its Backup gateways. Gateway synchronisation is the process where GatewayS peers request data from other GatewayS peers when they join the Gateway Peer Overlay Network. Gateway synchronisation ensures that GatewayS peers have all the latest information in the Gateway Peer Overlay Network. Gateway synchronisation reduces broadcasting service requests over the Gateway Peer Overlay Network. Data replication overcomes the failure of a single gateway i.e. in case of failure of Active gateway, one of the Backup gateways is promoted to Active gateway and communication can be started from point where last replication done.

8.2.5 Networked Appliances Utilisation

Recent advances in home networking devices and the increase in users connected to the internet, has allowed "home automation" to gain more attention. Home automation is the process of accessing and controlling home devices from remote locations across the Internet. In our work, we explore novel ways to utilise Networked Appliances. We explore how these appliances can be connected in a P2P network, benefitting single user to big suppliers. We explore in this research how Networked Appliances can be utilised as Gateways allowing Networked Appliances to be accessed irrespective of their location. We are using these appliances in our daily lives such as in home, office, public places ranging home appliances such as TVs, PCs, and Audio/Video devices to office appliances such as PDAs, and printers. Our framework seamlessly integrates these devices to enable intercommunication between the

functions they provide irrespective of where you are at any given time. Using appliances as a 'gateway' eliminates the use of special purpose gateways to allow devices and overcome the single point of failure. Our results demonstrate, a move towards the vision suggested by Mark Weiser of the "computer everywhere," and increasingly use computers as an essential part of our daily lives. Our results demonstrate how it can help in arranging daily home visit jobs such as engineer visits, receiving postal deliveries and so on, enabling the user to communicate with an engineer or postman while they are away from home.

8.3 Further Work

The implementation and case study evaluation demonstrate our contribution to knowledge has been made and research carried out addresses several research problems. However, in our research many challenges were raised and a number of interesting questions need further research to answer them. This section provides details of some of the questions raised, which provides an interesting research focus for future researchers in Liverpool John Moores University or anyone interested within the networking community.

8.3.1 Security

One of the key functions that AdHocGS framework does not fully address is that of security. For pragmatic reasons the framework developed only addressed very basic security. Due to the *ad hoc* nature of P2P, middleware must ensure that only authorised peer(s) can access services. The middleware must also ensure that content sent and received between two peers must be authenticated. It must ensure that data streams are not intercepted or altered during transmission. In this way trust may be maintained between services, as there is no central controller in P2P.

To address this challenge, a smart authentication mechanism needs to be developed. This mechanism also needs to guarantee that data transmitted between services has not been altered or intercepted during transmission. In case of transmission of sensitive data such as payment details we must encrypt each packet. In case of streaming video or audio data might need less encryption where one packet

needs to be encrypted after certain number of packets i.e. 50th packet. This mechanism must be lightweight and can be installed on any devices.

8.3.2 Quality of Service

Another key requirement that needs to be further addressed is Quality of Service. In case of P2P, resources are very limited and due to its ad hoc nature cannot guarantee service availability at any point. In most cases of data transmission one needs to pay for bandwidth usage or sometimes limits to bandwidth allocated. In order improve service provision we need to know in advance any costs involved in particular data transmission and availability of services. In some cases, to prioritise some services than other must ensure availability of bandwidth.

In our design, we discussed the Quality of Service Manager (QoSM) which ensures QoS in the framework. But we do not implement it as part of our evaluation. We clearly illustrate how QoSM can communicate with our modules in the framework, which gives clear understanding for future researchers. We already mentioned some of the interesting QoS mechanisms which can incorporate with our framework or new mechanisms could be developed which requires further research.

8.3.3 Device Capability Matching

Another key requirement that needs to be further addressed is Device Capability Matching. When a service is discovered and matched, several candidate services that offer same functionality may be present. Device capability matching is needed in order to check the hardware and software capabilities of a device, which are used to determine how effectively the device can execute the services it provides. For example, in the network different devices may be offering video capabilities such as computer monitor, television. Our framework only addresses this issue when service is discovered for the first time to check best available service. This requires further research in order to develop a mechanism where it checks the peers arriving in the network can provide better service than the one in use. In order to achieve this, a mechanism needs to be developed to keep track of the other devices offering the same service so that where that device becomes available it automatically routes all the information to it or it lets the device know that a better service is available now.

8.3.4 Protocol Independence

As there are many manufacturers creating electronic devices it may not be possible for Networked Appliances [Moyer 2002] to know about the characteristics of a target device; therefore it is important that the communication should be independent of any specific protocol implementation. For example, in the network one device may use TCP/IP while the destination device may use X.25 [Mohan 2004] – mechanisms to support protocol translations must therefore be defined [Abuelma'atti 2002]. Mechanism need to be developed through which devices from the various vendors can communicate i.e. to improve interoperability.

8.4 Concluding Remarks

In this thesis we have stated that configuration and composing of home appliances is very difficult for the ordinary home user. Nowadays home appliances available in the market such as TV, DVD players, mobile phones etc are very complicated. It is difficult for a user with less technical knowledge to install these devices and use them. Examples would be connecting your mobile phone with the laptop or connecting DVD player with TV. Lots of companies do offer free of charge installation upon purchasing item for first time but less so for any problems in future. The costs for an engineer call out and the wait at home may be too high. As discussed earlier in this thesis, we need to provide zero configurations or less user intervention in configuring and composing devices within a home environment. There are a number of devices available in home or office environments, which can be composed together, for example, in home environment such as video/audio services, the video service of a video player can be combined together with any video output device such as TV or computer monitor. Available audio devices can be used for audio output. But as we said earlier, we want to remove the complexity of this integration and configuration from the ordinary user. In our research we tried to understand how we can seamlessly interconnect the devices we own, independent of location. In order to achieve this independence of location, we need a gateway. Using gateways, we allow devices to be discovered globally such as at home and in office networks. Our approach is novel, which is reflected in the number of papers we have published (a full list can be found in Appendix E).

In addition to achieving our challenges after our literature survey, which included service discovery, naming and addressing, platform independence, dynamic service composition and device capability matching This thesis presents our AdHocGS framework, which discovers service, combined them together and provides alternative services. We argued that our framework allows devices to advertise services and discover other services across a P2P network of Gateways and integrate them together automatically. We presented a case study and developed a prototype to implement our framework.

Networked Appliances need a mechanism to create personalised device configurations that transcend beyond localised networked environments, allowing services offered by devices to be dynamically discovered and composed within *ad hoc* networks, which include home and wide area networks, freeing the user from the constraints imposed by machines to use services offered by home appliances without worrying about their configurations and integration. We believe this framework makes an important contribute to the vision for future Networked Appliances.

REFERENCES

[Abuelma'atti 2002] O. Abuelma'atti, Merabti, M. Askwith, B., "Interworking the Wireless Domain", Proceedings of the *Third International Symposium in Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, Staffordshire, UK, pp., (July 2002)

[Abuelma'atti 2006] O. Abuelma'atti, A. Mingkhwan, M. Merabti and B. Askwith, "A bridging architecture for wireless networked appliances", Proceedings of the *1st International Symposium on Wireless Pervasive Computing*, Phuket, Thailand, pp. 1-6, (January 2006)

[Almenarez 2008] F. Almenarez, A. Marin, D. Diaz, A. Cortes, C. Campo and C. Garcia-Rubio, "A trust-based middleware for providing security to ad-hoc peer-to-peer applications", Proceedings of the *6th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2008)*, Hong Kong, pp. 531-536, (17-21 March 2008)

[Anbazhagan 2002] M. Anbazhagan and A. Nagarajan, "Understanding Quality of Service for Web Services" http://www.ibm.com/developerworks/library/ws-quality.html, (Accessed:2009).

[Antoniu 2007] G. Antoniu, L. Cudennec, M. Jan and M. Duigou, "Performance scalability of the JXTA P2P framework", Proceedings of the *21st International Parallel and Distributed Processing Symposium (IPDPS'07)*, Long Beach, CA, United States, pp. 1-10, (26-30 March 2007)

[Arnold 2005] K. Arnold, James Gosling and D. Holmes, "The Java™ Programming Language", Prentice Hall, 4th edition. Isbn:0-321-34980-6 (2005).

[Arora 2005] G. Arora, M. Hanneghan and M. Merabti, "P2P commercial digital content exchange," *Electronic Commerce Research and Applications*, vol. 4 (3), pp. 250-263, (2005).

[Babcock 2007] C. Babcock, "Linux looks ahead," *InformationWEEK*, vol. 143 (1), pp. 27-8, (2007).

[Balakrishnan 2003] H. K. Balakrishnan, M. Frans; Karger, David; Morris, Robert; Stoica, Ion "Looking up data in P2P systems," *Communications of the ACM*, vol. 46 (2), pp. 43-48, (2003).

[Bhatti 2002] G. Bhatti, Z. Sahinoglu, K. A. Peker, J. Guo and F. Matsubara, "A TV-centric home network to provide a unified access to UPnP and PLC domains",

Proceedings of the *IEEE fourth International Workshop on Networked Appliances*, Gaithersburg, MD, USA, pp. 234-42, (15-16 Jan 2002)

[BitComet 2008] BitComet, "BitComet - A free C++ BitTorrent/HTTP/FTP Download Client", http://www.bitcomet.com/, (Accessed:2008).

[BitLord 2008] BitLord, "BitLord - The Ultimate Torrent Downloader", http://www.bitlord.com/, (Accessed:2008).

[BitTorrent 2008] BitTorrent, "Bit Torrent", http://www.bittorrent.com/company/, (Accessed:2008).

[Blackwell 2006] L. Blackwell, "Instant Messengers grow up and go to work," *PC World (San Francisco, CA)*, vol. 24 (2), pp. 66, (2006).

[Blake 2007] M. B. Blake, A. L. Sliva, M. Zur Muehlen and J. V. Nickerson, "Binding now or binding later: The performance of UDDI registries", Proceedings of the *40th Annual Hawaii International Conference on System Sciences (HICSS 2007)*, Waikoloa, HI, USA, pp. 171-178, (3-6th January 2007)

[Bo 2006] Z. Bo, J. Sushil and M. S. Kankanhalli, "Building trust in peer-to-peer systems: a review," *International Journal of Security and Networks*, vol. 1 (1-2), pp. 103-12, (2006).

[Booch 2005] G. Booch, J. Rumbaugh and I. Jacobson, "The unified modeling language user guide", Addison-Wesley, 2nd edition. Isbn:0321267974 (2005).

[Booth 2004] D. Booth and H. Hugo, "Web Services Architecture", http://www.w3.org/TR/ws-arch/wsa.pdf, (Accessed:2008).

[Bravetti 2008] M. Bravetti, S. Gilmore, C. Guidi and M. Tribastone, "Replicating Web Services for Scalability", Proceedings of the *3rd International Symposium on Trustworthy Global Computing (TGC 2007)*, Sophia-Antipolis, France, pp. 204-221, (5-6th November 2008)

[Brookshier 2002] D. Brookshier, Navaneeth Krishnan, Darren Govoni and J. C. Soto, "JXTA: Java P2P Programming", SAMS, 1st edition. Isbn:0672323664 (2002).

[Bull 2002] P. M. Bull, P. R. Benyon and P. R. Limb, "Residential gateways," *BT Technology Journal*, vol. 20 (2), pp. 73-81, (2002).

[Burstein 2005] M. Burstein, C. Bussler, T. Finin, M. N. Huhns, M. Paolucci, A. P. Sheth, S. Williams and M. Zaremba, "A semantic Web Services Architecture," *IEEE Internet Computing*, vol. 9 (5), pp. 72-81, (2005).

[Cameroon 2006] R. Cameroon, B. Woodberg, M. K. Madwachar, M. Swarm, N. R. Wyler, M. Albers and R. Bonnell, "Networking, Security, and the Firewall," in *Configuring Juniper Networks NetScreen and SSG Firewalls*, Syngress, 1st edition. Isbn:1597491187 (2006).

[Cameroon 2006] R. Cameroon, B. Woodberg, M. K. Madwachar, M. Swarm, N. R. Wyler, M. Albers and R. Bonnell, "Routing," in *Configuring Juniper Networks NetScreen and SSG Firewalls*, Syngress, 1st edition. Isbn:1597491187 (2006).

[Casad 2008] J. Casad, "Sams Teach Yourself TCP/IP in 24 Hours", Sams, 4th edition. Isbn:0672329964 (2008).

[Castle 2005] B. Castle, "The Legion of the Bouncy Castle", http://www.bouncycastle.org/, (Accessed:2005).

[Castro 2002] M. D. Castro, P.; Kermarrec, A.-M.; Rowstron, A.I.T, "Scribe: a large-scale and decentralized application-level multicast infrastructure," *IEEE Journal on Selected Areas in Communications*, vol. 20 (8), pp. 1489-1499, (2002).

[CES 2008] CES, "Consumer Electronic Show", http://www.cesweb.org/, (Accessed:2008).

[Chandra 2007] P. Chandra and D. Lide, "The Data World," in *Wi-Fi Telephony: Challenges and Solutions for Voice over WLANs*, Newnes, 1st edition. Isbn:0750679719 (2007).

[Chauvet 2004] A. Chauvet, "OpenMaster Quality of Service", http://www.evidian.com, (Accessed:January 2004).

[Chawathe 2003] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham and S. Shenker, "Making Gnutella-like P2P systems scalable", Proceedings of the *ACM SIGCOMM 2003: Conference on Computers*, Karlsruhe, Germany, pp. 407-418, (25 - 29 August 2003)

[Chen 2005] R. Chen and W. Yeager, "Poblano: A Distributed Trust Model for Peer-to-Peer Networks", Sun Microsystems, http://www.jxta.org/docs/trust.pdf, pp. 1-26, (2005)

[Chen 2007] S. Chen, J. Zic, K. Tang and D. Levy, "Performance evaluation and modeling of web services security", Proceedings of the *IEEE International Conference on Web Services (ICWS)*, Salt Lake City, UT, United states, pp. 431-438, (9-13th July 2007)

[Chen 2003] Z. Chen, C. Liang-Tien, B. Silverajan and L. Bu-Sung, "UX - An Architecture Providing QoS-Aware and Federated Support for UDDI", Proceedings of the *International Conference on Web Services (ICWS)*, Las Vegas, NV, United states, pp. 171-176, (23-26th June 2003)

[Cheng 2008] W.-q. Cheng, J. Gong and W. Ding, "Identifying file-sharing P2P traffic based on traffic characteristics," *Journal of China Universities of Posts and Telecommunications*, vol. 15 (4), pp. 112-120, (2008).

[Chih-Lin 2007] H. Chih-Lin, H. Yen-Ju and L. Wei-Shun, "Multicast complement for efficient UPnP eventing in home computing network", Proceedings of the *IEEE International Conference of Portable Information Devices*, Orlando, FL, USA, pp. 361-365, (25-29th March 2007)

[Choi 2005] K. H. Choi, H. J. Shin and D. R. Shin, "Service discovery supporting QoS in P2P network", Proceedings of the *7th International Conference on Advanced Communication Technology (ICACT)*. pp. 1241-1246, (21-23rd Feburary 2005)

[Christian 2006] P. Christian, "Let a thousand TV channels bloom [Internet protocol television]," *Engineering and Technology*, vol. 1 (7), pp. 28-31, (2006).

[Cisco 1992] Cisco, "Cisco Systems Inc." http://www.cisco.com/, (Accessed:2008).

[Cisco 2003] Cisco, "Quality of Service Networking," in *Internetworking Technologies Handbook*, Cisco Press, 4th edition. Isbn:157051192 (2003).

[Cisco 2006] Cisco, "Cisco Systems Inc." http://www.cisco.com/, (Accessed:2006).

[Cohen 2003] B. Cohen, "Incentives build Robustness in BitTorrent", Proceedings of the *1st Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, USA, pp., (5-6th June 2003)

[Cohen 2008] B. Cohen, "Bram Cohen Home Page", http://bramcohen.com/, (Accessed:2008).

[Comer 2005] D. E. Comer, "Internetworking with TCP/IP", Prentice Hall, 5th edition. Isbn:0131876716 (2005).

[Curbera 2006] F. Curbera, R. Khalaf, W. A. Nagy and S. Weerawarana, "Implementing BPEL4WS: The architecture of a BPEL4WS implementation," *Concurrency Computation Practice and Experience*, vol. 18 (10), pp. 1219-1228, (2006).

[D'Ambrogio 2007] A. D'Ambrogio and B. Paolo, "A model-driven approach to describe and predict the performance of composite services", Proceedings of the *6th international workshop on Software and performance*, Buenes Aires, Argentina, pp. 78-89, (5-8th Febrary 2007)

[David 2004] L.-j. David, M. Madjid, S. Qi and A. Bob, "Security in a Ubiquitous Computing Environment", Proceedings of the *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 2158-2163, (29th Nov - 3rd December 2004)

[Detsch 2006] A. Detsch, L. P. Gaspary, M. P. Barcellos and R. N. Sanchez, "Flexible security configuration & deployment in peer-to-peer applications", Proceedings of the *IEEE/FIP Network Operations and Management Symposium*, Vancouver, BC, Canada, pp. 209-219, (3-7th April 2006)

[DLNA 2006] DLNA, "DLNA Overview and Vision Whitepaper 2006", http://www.dlna.org/en/industry/about/dlna_white_paper_2006.pdf, (Accessed:2008).

[Dobrev 2002]P. Dobrev, D. Famolari, C. Kurzke and B. A. Miller, "Device and service discovery in home networks with OSGi," *IEEE Communications Magazine*, vol. 40 (8), pp. 86-93, (2002).

[Dongyu 2008] Q. Dongyu and S. Weiqian, "Global stability of peer-to-peer file sharing systems," *Computer Communications*, vol. 31 (2), pp. 212-19, (2008).

[Downloader 2008] B. Downloader, "Blizzard Downloader - WoWWiki - Your guide to the World of Warcraft", http://www.wowwiki.com/Blizzard_Downloader, (Accessed:2008).

[DPWS 2006] DPWS, "The Devices Profile for Web Service specification ", http://specs.xmlsoap.org/ws/2006/02/devprof/devicesprofile.pdf, (Accessed:2009).

[DROPS 2007] DROPS, "P2P, Ad Hoc and Sensor Networks – All the Different or All the Same?" http://drops.dagstuhl.de/opus/volltexte/2007/951/pdf/06431.JanacikPeter.Paper.951.pdf, (Accessed:2009).

[Druschel 2001] P. R. Druschel, A, "PAST: a large-scale, persistent peer-to-peer storage utility", Proceedings of the *Proceedings of the Eighth Workshop on*

Hot Topics in Operating Systems, Schoss Elmau, Germany, pp. 75-80, (20-22 May 2001)

[ECMA 2006] ECMA, "Standard ECMA-370", http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-370.pdf, (Accessed:2009).

[El-Kadri 2006] M. El-Kadri, V. Groza, R. Abielmona and M. Assaf, "A just-in-time compiler for a reconfigurable testing platform", Proceedings of the *IEEE Instrumentation and Measurement Technology Conference*, Sorrento, Italy, pp. 628-632, (24-27th April 2006)

[Eng Keong Lua 2005] J. C. Eng Keong Lua, Marcelo Pias, Ravi Sharma and Steven Lim, "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes," *IEEE COMMUNICATIONS SURVEY AND TUTORIAL*, vol. 7 (2), pp. 72-93, (2005).

[ePerSpace 2005] ePerSpace, "Towards the era of personal services at home and everywhere", http://www.ist-eperspace.org/, (Accessed:2005).

[Evans 2001] D. Evans, "In-home wireless networking: an entertainment perspective," *Electronics & Communication Engineering Journal*, vol. 13 (5), pp. 213-19, (2001).

[Fisher 2006] D. Fisher, M. Smith and H. T. Welser, "You are who you talk to: Detecting Roles in usenet newsgroups", Proceedings of the, Kauai, HI, United States, pp. 59, 2006)

[Forum 2005] O. Forum, "The OSGi Service Platform - Dynamic services for networked devices", www.osgi.org, (Accessed:2005).

[Funabiki 2007] S. Funabiki, T. Isohara, Y. Kitada, K. Takemori and I. Sasase, "Self-organized public key management with certificate management nodes for wireless ad hoc networks," *Transactions of the Information Processing Society of Japan*, vol. 48 (8), pp. 2835-45, (2007).

[Gang 2009] Y. Gang, W. Chanle, Y. Jun, C. Shi and W. Chanle, "A QoS-aware model for Web services discovery", Proceedings of the *First International Workshop on Education Technology and Computer Science (ETCS)*, Piscataway, NJ, USA, pp. 740-4, (7-8th March 2009)

[Garg 2007] V. K. Garg, "Mobile Network and Transport Layer," in *Wireless Communications and Networking*, Elsevier Morgan Kaufmann, 1st edition. Isbn:0123735807 (2007).

[Garg 2007] V. K. Garg, "An Overview of Wireless Systems," in *Wireless Communications Networking*, Morgan Kaufmann, 1st edition. Isbn:0123735807 (2007).

[Garg 2007] V. K. Garg, "Wide-Area Wireless Networks (WANs) — GSM Evolution," in *Wireless Communications Networking*, Morgan Kaufmann, 1st edition. Isbn:0123735807 (2007).

[Garg 2007] V. K. Garg, "Wireless Personal Area Networks:Low Rate and High Rate," in *Wireless Communications and Networking*, Elsevier Morgan Kaufmann, edition. Isbn:0123735807 (2007).

[Gauthier 2008] C. G. Gauthier and C. Grothoff, "Bootstrapping of peer-to-peer networks", Proceedings of the *International Symposium on Applications and the Internet (SAINT)*, Turku, Finland, pp. 205-208, (28th July - 1st August 2008)

[GauthierDickey 2008] C. GauthierDickey and C. Grothoff, "Bootstrapping of peer-to-peer networks", Proceedings of the *International Symposium on Applications and the Internet (SAINT)*, Turku, Finland, pp. 205-208, 2008)

[Ghenniwa 2005] H. Ghenniwa, M. N. Huhns and W. Shen, "EMarketplaces for enterprise and cross enterprise integration," *Data and Knowledge Engineering*, vol. 52 (1), pp. 33-59, (2005).

[Gillett 2001] S. E. Gillett, W. H. Lehr, J. T. Wroclawski and D. D. Clark, "Do appliances threaten Internet innovation?," *IEEE Communications Magazine*, vol. 39 (10), pp. 46-51, (2001).

[Gmach 2008] D. Gmach, S. Krompass, A. Scholz, M. Wimmer and A. Kemper, "Adaptive quality of service management for enterprise services," *ACM Transactions on the Web*, vol. 2 (1), pp. 8, (2008).

[Gnutella 2008] Gnutella, "The Gnutella Protocol Specification v0.4", http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf, (Accessed:2009).

[Gobel 2004] M. Gobel, "VoIP Basic", Schlembach Fachverlag, edition. Isbn:3935340281 (2004).

[Group 2008] A. Group, "Trusted Computing: Tune In, Turn It On", https://www.trustedcomputinggroup.org/news/Industry_Data/Aberdeen_Report_TC_TuneIn_TurnItOn.pdf, (Accessed:2008).

[Guo 2007] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding and X. Zhang, "A performance study of BitTorrent-like peer-to-peer systems," *IEEE Journal on Selected Areas in Communications*, vol. 25 (1), pp. 155-169, (2007).

[Haggerty 2005] J. Haggerty, Q. Shi and M. Merabti, "Early Detection and Prevention of Denial-of-Service Attacks: A Novel Mechanism with Propagated Traced-Back Attack Blocking," *IEEE Journal on Selected Areas in Communications (J-SAC)*, vol. 23 (10), pp. 1994-2002, (2005).

[Hartog 2004] D. Hartog, M. Balm, De Jong and J. J. B. Kwaaitaal, "Convergence of residential gateway technology," *IEEE Communications Magazine*, vol. 42 (5), pp. 138-143, (2004).

[Hauben 2001] R. Hauben, "From the ARPANET to the Internet", http://www.columbia.edu/~rh120/other/tcpdigest_paper.txt, (Accessed:2008).

[HAVi 2004] HAVi, "The HAVi Specification", http://www.havi.org/HAVi_1.1.pdf, (Accessed:2006).

[HAVi 2004] HAVi, "HAVi, the A/V digital network revolution", http://www.havi.org/pdf/white.pdf, (Accessed:2004).

[Helm 2006] B. Helm, "BitTorrent Goes Hollywood " http://www.businessweek.com/technology/content/may2006/tc20060508_693082.htm, (Accessed:2009).

[Hoffman 2006] L. J. Hoffman, K. Lawson-Jenkins and J. Blum, "Trust beyond security: An expanded trust model," *Communications of the ACM*, vol. 49 (7), pp. 95-101, (2006).

[Hsiao 2003] H.-C. Hsiao and C.-T. King, "A Tree Model for Structured Peer-to-Peer Protocols", Proceedings of the *3rd International Symposium on Cluster Computing and the Grid* Tokyo, Japan, pp. 336-343, (12-15th May 2003)

[Hu 2008] X.-h. Hu, Y.-j. Fu and Z.-p. Zhang, "Research of Web services security solution based on policy," *Microcomputer Information*, vol. 32 (15), pp. 93-4, (2008).

[Hung-Chang Hsiao 2003] C.-T. K. Hung-Chang Hsiao, "A Tree Model for Structured Peer-to-Peer Protocols", Proceedings of the *3rd International Symposium on Cluster Computing and the Grid* Tokyo, Japan, pp. 336-343, 2003)

[IBM 2009] IBM, "IBM", www.ibm.com, (Accessed:2009).

[IBM 2005] IBM and BEA, "Web Services Transactions specifications", http://www-106.ibm.com/developerworks/webservices/library/ws-transpec/?dwzone=webservices, (Accessed:2009).

[isoHunt 2008] isoHunt, "isoHunt - the BitTorrent and P2P search engine", http://isohunt.com, (Accessed:2008).

[Jakab 2007] M. Jakab, M. Kropfberger, M. Ofner, R. Tusch, H. Hellwagner and L. Boszormenyi, "Metadata integration and media transcoding in Universal-Plug-and-Play (UPnP) enabled networks", Proceedings of the *15th EUROMICRO International Conference on Parallel, Distributed and Network-Based Processing*, Naples, Italy, pp. 363-369, (7-9th Febrary 2007)

[Jammes 2007] F. Jammes, A. Mensch and H. Smit, "Service-oriented device communications using the devices profile for Web services", Proceedings of the *21st International Conference on Advanced Information Networking and Applications Workshops/Symposia (AINAW)*, Niagara Falls, ON, Canada, pp. 987-995, (21-23rd May 2007)

[Jennings 2006] C. Jennings and D. A. Bryan, "P2P for communications: beyond file sharing," *Business Communications Review*, vol. 36 (2), pp. 36-40, (2006).

[JETRO 2005] JETRO, "Japan External Trade Organisation - Ubiquitous Networks", http://www.jetro.org/index.php?option=com_content&task=view&id=237, (Accessed:2006).

[Jiang 2008] D. Jiang and M. Li, "Quality of service in the home network", Proceedings of the *Second International Conference on Future Generation Communication and Networking (FGCN)*, Piscataway, NJ, USA, pp. 473-476, (13-15 December 2008)

[KaZaA 2008] KaZaA, "KaZaA", http://www.kazaa.com/us/index.htm, (Accessed:2009).

[Kester 2003] W. Kester, "DSP Applications," in *Mixed-signal and DSP Design Techniques*, Elsevier Newnes, 1st edition. Isbn:0750676116 (2003).

[Kim 2006] K.-S. Kim, C. Park and J. Lee, "Internet home network electrical appliance control on the Internet with the UPnP expansion", Proceedings of the *International Conference on Hybrid Information Technology*, Cheju Island, South Korea, pp. 629-634, (12-14th Febrary 2006)

[Knauth 2007] S. Knauth, R. Kistler, D. Kaslin and A. Klapproth, "UPnP compression implementation for building automation devices", Proceedings of the *5th IEEE International Conference on Industrial Informatics*, Vienna, Austria, pp. 75-79, (23-27th July 2007)

[Kocbek 2007] S. Kocbek and M. B. Juric, "Influence of security mechanisms on Web services interoperability," *Elektrotehniski Vestnik*, vol. 74 (3), pp. 113-18, (2007).

[Kojiro 2008] N. Kojiro, O. Michiko, A. Michitaka and K. Norihisa, "Processing methods for partially encrypted data in multihop Web services," *Electronics and Communications in Japan*, vol. 91 (5), pp. 26-32, (2008).

[Kontogiannis 2008] K. Kontogiannis, "Challenges and opportunities related to the design, deployment and, operation of web services", Proceedings of the *16th Frontiers of Software Maintenance (FoSM)*, Beijing, China, pp. 11-20, (30th Sept - 2nd Oct 2008)

[Krafzig 2004]D. Krafzig, K. Banke and D. Slama, "Enterprise SOA: Service-Oriented Architecture Best Practices ", Prentice Hall PTR, 1st edition. Isbn:0131465759 (2004).

[Kumar 2006] V. Kumar, "Trust and Security in Peer-to-Peer System", Proceedings of the *17th International Conference on Database and Expert Systems Application (DEXA)*, Krakow, Poland, pp. 703-707, (4-8th Septemeber 2006)

[Kumaran 2007] S. Kumaran, P. Bishop, T. Chao, P. Dhoolia, P. Jain, R. Jaluka, H. Ludwig, A. Moyer and A. Nigam, "Using a model-driven transformational approach and service-oriented architecture for service delivery management," *IBM Systems Journal*, vol. 46 (3), pp. 513-529, (2007).

[Lee 2007] J.-J. Lee, C.-Y. Huang, L.-Y. Lee and C.-L. Lei, "Design and implementation of secure communication channels over UPnP networks", Proceedings of the *International Conference on Multimedia and Ubiquitous Engineering (MUE)*, Seoul, South Korea, pp. 307-312, 2007)

[Leymann 2001] F. Leymann, "Web Services Flow Language (WSFL 1.0)" http://xml.coverpages.org/WSFL-Guide-200110.pdf, (Accessed:2009).

[Li 2008] J. Li, "On peer-to-peer (P2P) content delivery," *Peer-to-Peer Networking and Applications*, vol. 1 (1), pp. 45-63, (2008).

[Li 2002] Z. Li, D. Huang, L. Zhuang and J. Huang, "Research of peer discovery method in peer-to-peer network", Proceedings of the *IEEE Conference on*

Computers, Communications, Control and Power Engineering, Beijing, China, pp. 383-386, (28-31 Oct 2002)

[Li 2007] Z. Li and P. Mohapatra, "On investigating overlay service topologies," *Computer Networks*, vol. 51 (1), pp. 54-68, (2007).

[Libby 2007] J. C. Libby and K. B. Kent, "An embedded implementation of the Microsoft common language infrastructure", Proceedings of the *10th Euromicro Conference on Digital System Design: Architectures, Methods and Tools*, Lubeck, Germany, pp. 155-62, (29-31st August 2007)

[Liebel-Lab 2008] K. K. I. o. T.-. Liebel-Lab, "KIT Search Engine Initiative", http://sciencenet.fzk.de/, (Accessed:2008).

[LimeWire 2006] LimeWire, "LimeWire", http://www.limewire.com/english/content/home.shtml, (Accessed:2008).

[Linux 2007] Linux, "The Linux Home Page", www.linux.org, (Accessed:2009).

[Linux 2008] Linux, "YaCy Distributed Web Search", http://yacy.net/index.html, (Accessed:2008).

[Locasto 2005] M. E. Locasto, J. J. Parekh, A. D. Keromytis and S. J. Stolfo, "Towards collaborative security and P2P intrusion detection", Proceedings of the *6th Annual IEEE System, Man and Cybernetics Information Assurance Workshop (IAW '05)*, West Point, NY, USa, pp. 333-339, (15-17th June 2005)

[Louridas 2006] P. Louridas, "SOAP and web services," *IEEE Software*, vol. 23 (6), pp. 62-67, (2006).

[Lua 2005] K. Lua, J. Crowcroft, C. Pias, R. Sharma and S. Lim, "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes," *IEEE Communications Survey and Tutorial*, vol. 7 (2), pp. 72-93, (2005).

[Mabanza 2007] N. Mabanza, "A comparison of m-commerce and e-commerce transport layer security protocols", Proceedings of the *4ht IASTED Asian Conference on Communication Systems and Networks (AsiaCSN'07)*, Anaheim, CA, USA, pp. 207-211, (2-4th August 2007)

[Machinery 2004] A. f. C. Machinery, "ACM Special Interest Groups Guide", http://www.acm.org/sigs/guide98.html, (Accessed:2008).

[Magharei 2007] N. Magharei, Y. Guo and R. Rejaie, "Issues in offering live P2P streaming service to residential users", Proceedings of the *4th Annual IEEE Consumer Communications and Networking Conference (CCNC'07)*, Las Vegas, NV, United states, pp. 757-762, (11-13 January 2007)

[Mahmoud 2007] Q. H. Mahmoud and Z. Wang, "Customizing and delivering mobile services using software agents and CC/PP", Proceedings of the *4th IEEE Consumer Communications and Networking Conference*, Las Vegas, NV, USA, pp., (11-13 January 2007)

[Malkhi 2002] D. N. Malkhi, Moni; Ratajczak, David "Viceroy: A scalable and dynamic emulation of the butterfly", Proceedings of the *21st Annual ACM Symposium on Principles of Distributed Computing (PODC'02)*, Monterey, CA, United States, pp. 183-192, (21-24 July 2002)

[Manola 2004]F. Manola and E. Miller, "RDF Primer - W3C Recommendation", http://www.w3.org/TR/2004/REC-rdf-primer-20040210/, (Accessed:2009).

[Marples 2001] D. Marples and P. Kriens, "The open services gateway initiative: An introductory overview," *IEEE Communications Magazine*, vol. 39 (12), pp. 110-114, (2001).

[Marples 2001] D. Marples, Kriens, P., "The open services gateway initiative: An introductory overview," *IEEE Communications Magazine*, vol. 39 (12), pp. 110-114, (2001).

[Matsubara 2007] F. M. Matsubara, T. Hanada, S. Imai, S. Miura and S. Akatsu, "Networked device capability and content media format matching scheme for multimedia access," *IEEE Transactions on Consumer Electronics*, vol. 53 (1), pp. 145-149, (2007).

[Maymounkov 2002] P. Maymounkov and D. Mazi`eres, "Kademlia: A Peer-to-peer Information System Based on the XOR Metric", Springer Berlin / Heidelberg, edition. MIT, Isbn:978-3-540-44179-3 (2002).

[Merabti 2008] M. Merabti, P. Fergus, O. Abuelma'atti, H. Yu and C. Judice, "Managing Distributed Networked Appliances in Home Networks," *Proceedings of the IEEE*, vol. 96 (1), pp. 166-185, (2008).

[Merabti 2008] M. Merabti, P. Fergus, O. Abuelma'atti, H. Yu and C. Judice, "Managing distributed networked appliances in home networks," *Proceedings of the IEEE*, vol. 96 (1), pp. 166-85, (2008).

[Microsoft 2004] Microsoft, "Understanding Universal Plug and Play", http://www.upnp.org/, (Accessed:20099).

[Microsoft 2008] Microsoft, "Security - MSDN", http://msdn.microsoft.com, (Accessed:2009).

[Microsoft 2008] Microsoft, "A Technical Introduction to the Devices Profile for Web Services - MSDN", http://msdn.microsoft.com, (Accessed:2009).

[Microsoft 2009] Microsoft, "Microsoft", www.microsoft.com, (Accessed:2009).

[Microsystems 2005] S. Microsystems, "JXTA v2.3.x: Java Programmer's Guide", http://www.jxta.org/docs/JxtaProgGuide_v2.3.pdf, (Accessed:2008).

[Milanovic 2004] N. Milanovic and M. Malek, "Current solutions for Web service composition," *IEEE Internet Computing*, vol. 8 (6), pp. 51-59, (2004).

[Mingkhwan 2005] A. Mingkhwan, P. Fergus, O. Abuelma'atti, M. Merabti, B. Askwith and M. Hanneghan, "Dynamic Service Composition in Home Appliance Networks," *To appear in Multimedia Tools and Applications: A Special Issue on Advances in Consumer Communications and Networking*, (2005).

[Mininova 2008] Mininova, "Mininova : The ultimate BitTorrent source!" http://www.mininova.org, (Accessed:2009).

[Minoh 2001] M. Minoh and T. Kamae, "Networked appliances and their peer-to-peer architecture AMIDEN," *IEEE Communications Magazine*, vol. 39 (10), pp. 80-84, (2001).

[Mirko 2007] K. Mirko, W. Arno, S. Gregor and W. Torben, "Decentralized Bootstrapping in Pervasive Applications", Proceedings of the *5th IEEE International Conference on Pervasive Computing and Communications Workshops*, White Plains, NY, USA, pp. 589-592, (19-23 March 2007)

[MMORPG 2008] MMORPG, "MMORPG.com - Your Headquarters for Massive Multiplayer Online Role-Playing Games!" http://www.mmorpg.com/index.cfm?bhcp=1, (Accessed:2008).

[Mohan 2004] N. Mohan, "X.25 protocol," *Telecommunications*, vol. 42 (3), pp. 51-63, (2004).

[Mol 2008] J. J. D. Mol, J. A. Pouwelse, D. H. J. Epema and H. J. Sips, "Free-riding, fairness, and firewalls in P2P file-sharing ", Proceedings of the *Eighth International Conference on Peer-to-Peer Computing (P2P)*, Aachen, Germany, pp. 301-310, (8-11 September 2008)

[Moyer 2002] S. Moyer, D. Maples, S. Tsang and A. Ghosh, "Service portability of networked appliances," *IEEE Communications Magazine*, vol. 40 (1), pp. 116-121, (2002).

[Muhammad 2005] A. Muhammad, M. Merabti and B. Askwith, "An Ad Hoc Gateway Service for Discovering and Composing Networked Appliances", Proceedings of the *sixth annual postgraduate symposium on the convergence of telecommunications, networking and broadcasting (PGNet 2005)*, Liverpool John Moores University,UK, pp. 377-382, (27-28 June 2005)

[Muhammad 2007] A. Muhammad, M. Merabti, B. Askwith and P. Fergus, "Ad Hoc Gateway Service for Automatic Package Delivery using Networked Appliances", Proceedings of the *IEEE Wireless Communications and Networking Conference (WCNC'07)*, Kowloon, China, pp. 2578-2583, (11-15 March 2007)

[Muhammad 2007] A. Muhammad, M. Merabti, B. Askwith and P. Fergus, "Ad Hoc Gateway Service for Automatic Package Delivery using Networked Appliances", Proceedings of the *IEEE Wireless Communications and Networking Conference*, Hong Kong, pp. 2576-2581, (11-15 March 2007)

[Muhammad 2007] A. Muhammad, M. Merabti. and B. Askwith, "An Ad Hoc Gateway Service for Flexible Access to Networked Appliances", Proceedings of the *8th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, Liverpool John Moores University,Liverpool, UK, pp. 141-145, (28-29 June 2007)

[Nagaraja 2006] K. Nagaraja, S. Rollins and M. Khambatti, "From the editors: Peer-to-peer community: Looking beyond the legacy of napster and gnutella," *IEEE Distributed Systems Online*, vol. 7 (3), pp. 59-65, (2006).

[Networks 2008] J. Networks, "Networking Security Solutions:Juniper Networks", http://www.juniper.net/, (Accessed:2009).

[Nokia 2009] Nokia, "Nokia", www.nokia.com, (Accessed:2009).

[Novell 2008] Novell, "Mono", http://www.mono-project.com, (Accessed:2009).

[Núñez 2006] A. Núñez, "Sensible policies for QoS-based service discovery protocols in P2P networks", Proceedings of the *5th IASTED International Conference on Communication Systems and Networks (CSN'06)*

Palma de Mallorca, Spain, pp. 189-194, (28-30 August 2006)

[O'Mahony 2003] D. D. D. O'Mahony, "Overlay Networks: A Scalable Alternative for P2P", http://www.dynamicobjects.com/papers/w4spot.pdf, (Accessed:2008).

[Oaks 2002] S. Oaks, B. Traversat and L. Gong, "JXTA in a Nutshell", O'REILLY, 1st edition. Sebastopol, Isbn:0-596-00236-X (2002).

[OASIS 2004] OASIS, "Introduction to UDDI: Important Features and Functional Concepts", http://xml.coverpages.org/UDDI-TechnicalWhitePaperOct28.pdf, (Accessed:2009).

[OASIS 2007] OASIS, "OASIS: Organization for the Advancement of Structured Information Standards", http://www.oasis-open.org/home/index.php, (Accessed:2008).

[Oxford 2009] Oxford, "Oxford English Dictionary", http://www.askoxford.com/concise_oed/gateway?view=uk, (Accessed:2009).

[Palomar 2006] E. Palomar, J. M. Estevez-Tapiador, J. C. Hernandez-Castro and A. Ribagorda, "Security in P2P networks: survey and research directions", Proceedings of the *Emerging Directions in Embedded and Ubiquitous Computing (EUC'06)*, Seoul, South Korea, pp. 183-192, (1-4 August 2006)

[Peers 2006] F. Peers, "BearShare", http://www.bearshare.com/, (Accessed:2006).

[Pirzada 2006] A. A. Pirzada and C. McDonald, "Trust establishment in pure ad-hoc networks," *Wireless Personal Communications*, vol. 37 (1-2), pp. 139-168, (2006).

[POP 2008] S. POP, "Sub Pop Records", http://www.subpop.com/, (Accessed:2009).

[Pouwelse 2005] J. Pouwelse, P. Garbacki, D. Epema and H. Sips, "The Bittorrent P2P file-sharing system: Measurements and analysis", Proceedings of the, Ithaca, NY, United states, pp. 205-216, 2005)

[Ratnasamy 2001] S. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Shenker, "A scalable content-addressable network", Proceedings of the *Conference. Applications, Technologies, Architectures, and Protocols for Computer Communications (ACMSIGCOMM'01)*, San Diego, CA, USA, pp. 161-172, (27-31 August 2001)

[Redondo 2007] R. P. D. Redondo, A. F. Vilas, M. R. Cabrer, J. J. P. Arias and M. R. Lopez, "Enhancing Residential Gateways: OSGi Services Composition", Proceedings of the *Digest of Technical Papers. International Conference on Consumer Electronics (ICCE'07)*, Las Vegas, NV, USA, pp. 1-2, (10-14 January 2007)

[Rieche 2007] S. Rieche, K. Wehrle, M. Fouquet, H. Niedermayer, L. Petrak and G. Carle, "Peer-to-peer-based infrastructure support for massively multiplayer online games", Proceedings of the *4th Annual IEEE Consumer Communications and Networking Conference (CCNC'07)*, Las Vegas, NV, USA, pp. 763-767, (11-13 January 2007)

[Rot 2008] A. Rot and L. Ziora, "Selected problems of web services security", Proceedings of the *International Conference on Semantic Web Web Services (SWWS'08)*, Las Vegas, NV, USA, pp. 210-214, (14-17 July 2008)

[Rowstron 2001] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems," *IEEE Journal on Selected Areas in Communications*, vol. 20 (8), pp. 1489-1499, (2001).

[Runfang 2007] Z. Runfang and K. Hwang, "PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18 (4), pp. 460-473, (2007).

[Salus 1995] P. H. Salus, "Casting the Net: From Arpanet to Internet and Beyond ", Addison Wesley, edition. Isbn:0201876744 (1995).

[Samsudin 2008] A. T. Samsudin, N. S. Herman, H. Unger and M. K. Awang, "The searching scalability of peer-to-peer system", Proceedings of the *International Conference on Advanced Communication Technology (ICACT'08)*, Phoenix Park, South Korea, pp. 1891-1896, (17-20 Febrary 2008)

[Schiele 2007] G. Schiele, R. Suselbeck, A. Wacker, J. Hahner, C. Becker and T. Weis, "Requirements of peer-to-peer-based massively multiplayer online gaming", Proceedings of the *7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07)*, Rio de Janeiro, Brazil, pp. 738-743, (14-17 May 2007)

[Schmied 2007] F. Schmied and A. Cyment, "Aspect-oriented weaving and the.NET common language runtime," *IET Software*, vol. 1 (6), pp. 251-262, (2007).

[Schoder 2003] D. Schoder and K. Fischbach, "Peer-to-peer prospects," *Communications of the ACM*, vol. 46 (2), pp. 27-29, (2003).

[Science 2006] F. C. Science, "P2P", http://ww2.cs.fsu.edu/~jungkkim/P2P.html, (Accessed:2008).

[Shareaze 2006] Shareaze, "Shareaze", http://www.shareaza.com/, (Accessed:2009).

[Shuping 2003] R. Shuping, "A model for web services discovery with QoS," *SIGecom Exch.*, vol. 4 (1), pp. 1-10, (2003).

[Singhal 2008]A. Singhal, "Web services security: Techniques and challenges (extended abstract)", Proceedings of the, Berlin, Germany, pp. 158, 2008)

[SIRENA 2005] SIRENA, "SIRENA consortium 2003", http://www.sirena-itea.org/Sirena/Home.htm, (Accessed:2009).

[Sivashanmugam 2004] K. Sivashanmugam, J. A. Miller, A. P. Sheth and K. Verma, "Framework for semantic Web process composition," *International Journal of Electronic Commerce*, vol. 9 (2), pp. 71-106, (2004).

[Skype 2008] Skype, "Skype Offical Website", http://www.skype.com/intl/en-gb/, (Accessed:2009).

[Solaris 2008] Solaris, "Solaris Operating System", http://www.sun.com/software/solaris/index.jsp, (Accessed:2009).

[Song 2009] H. Song, D. Tharam, C. Elizabeth and T. Biming, "Secure web services using two-way authentication and three-party key establishment for service delivery," *Journal of Systems Architecture*, vol. 55 (4), pp. 233-242, (2009).

[Starner 2002] T. Starner, "Thick clients for personal wireless devices," *Computer*, vol. 35 (1), pp. 133-5, (2002).

[Stoica 2003] I. M. Stoica, Robert; Liben-Nowell, David; Karger, David R.; Kaashoek, M. Frans; Dabek, Frank; Balakrishnan, Hari "Chord: A scalable peer-to-peer lookup protocol for Internet applications," *IEEE/ACM Transactions on Networking*, vol. 11 (1), pp. 17-32, (2003).

[Sun 2006] L. Y. Sun, Y. Wei, H. Zhu and K. J. R. Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24 (2), pp. 305-17, (2006).

[Sunshine 1990] C. A. Sunshine, "Network interconnection and gateways," *IEEE Journal on Selected Areas in Communications*, vol. 8 (1), pp. 4-11, (1990).

[Systems 2005] C. Systems, "PureTLS", http://www.rtfm.com/puretls/, (Accessed:2009).

[T. Dierks 1999] C. A. T. Dierks, "The TLS Protocol", http://www.ietf.org/rfc/rfc2246.txt?number=2246, (Accessed:2009).

[Takeda 2008] A. Takeda, K. Hashimoto, G. Kitagata, S. M. S. Zabir, T. Kinoshita and N. Shiratori, "A new authentication method with distributed hash table for P2P network", Proceedings of the *22nd International Conference on Advanced Information Networking and Applications (AINA'08)*, Okinawa, Japan, pp. 483-488, (25-28 March 2008)

[Tanaka 2007] K. Tanaka, "Research on fusion of the web and TV broadcasting", Proceedings of the *International Conference on Informatics Research for Development of Knowledge Society*, Kyoto, Japan, pp. 129-136, (29th January 2007)

[Tanenbaum 2003] A. S. Tanenbaum, "Computer Networks", Prentice Hall, 4th edition. Isbn:0-13-066102-3 (2003).

[Terashima 2001] N. Terashima, "Design Methodology for Telecommunication Services," in *The Intelligent Communication System: Toward Constructing a Human Friendly Communication Environment*, Elsevier Academic Press, edition. Isbn:0126853517 (2001).

[Thompson 2003] S. Thompson, "Implementing WS-Security" http://www.ibm.com/developerworks/webservices/library/ws-security.html, (Accessed:2009).

[Trivedi 2009] R. Trivedi, "Web Services Tutorial: Understanding XML and XML Schema", http://www.developer.com/services/article.php/2195981, (Accessed:2009).

[University 2006] P. S. University, "LionShare", http://lionshare.psu.edu/, (Accessed:2009).

[Unix 2008] Unix, "The Unix System", http://www.unix.org/, (Accessed:2008).

[UPnP 2006] UPnP, "UPnP Technology - The Simple, Seamless Home Network", http://www.upnp.org/specs/arch/UPnP-DeviceArchitecture-v1.0-20060720.pdf, (Accessed:2009).

[UPnP 2006] UPnP, "UPnP™ Standards", http://www.upnp.org/standardizeddcps/basic.asp, (Accessed:2009).

[UPnP July 2006] UPnP, "UPnP Technology - The Simple, Seamless Home Network", UPnP, http://www.upnp.org/specs/arch/UPnP-DeviceArchitecture-v1.0-20060720.pdf, pp. 8, (July 2006) (Accessed:2008).

[Venkitaraman 2007] N. Venkitaraman, "Wide-area media sharing with UPnP/DLNA", Proceedings of the *5th IEEE Consumer Communications and*

Networking Conference, Las Vegas, NV, USA, pp. 294-298, (10-12 January 2007)

[VOIP-Info.org 2003] VOIP-Info.org, "Voip-Info.org - A reference guide to all things VOIP", http://www.voip-info.org/wiki/view/QoS, (Accessed:2009).

[Vonage 2006] Vonage, "Vonage - A better phone service for less", www.vonage.com, (Accessed:2006).

[Vroonhoven 2006] J. v. Vroonhoven, "Peer to Peer Security", http://referaat.cs.utwente.nl/documents/2006_04_A-Broadband_for_All/2006_04_A_Vroonhoven,J.van-Peer_to_Peer_security.pdf, (Accessed:2009).

[W3C 2003] W3C, "QoS for Web Services: Requirements and Possible Approaches" http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/, (Accessed:2009).

[W3C 2007] W3C, "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 2.0", http://www.w3.org/TR/2007/WD-CCPP-struct-vocab2-20070430/, (Accessed:2009).

[Walker 2001] L. Walker, "Uncle Sam Wants Napster!" http://www.washingtonpost.com/ac2/wp-dyn?pagename=article&node=washtech/techthursday/columns/dotcom&contentId=A59099-2001Nov7, (Accessed:2008).

[Walls 2006] C. Walls, "Chapter 8: Networking," in *Embedded Softwar: The Works*, Newnes, edition. Isbn:0750679549 (2006).

[Walls 2006] C. Walls, "Networking," in *Embedded Software: The Works*, Butterworth Heinemann, 1st edition. Isbn:0750679549 (2006).

[Walsh 1998] N. Walsh, "XML", http://www.xml.com/pub/a/98/10/guide1.html#AEN58, (Accessed:2009).

[Wan Nurhayati Wan Ab. Rahman 2008] Wan Nurhayati Wan Ab. Rahman and D. F. Meziane, "Challenges to Describe QoS Requirements for Web Services Quality Prediction to Support Web Services Interoperability in Electronic Commerce," *Communications of the IBIMA*, vol. 4 (6), pp. 50-58, (2008).

[Weiser 2002] M. Weiser, "The computer for the 21st Century," *IEEE Pervasive Computing*, vol. 1 (1), pp. 19-25, (2002).

[Wendell 2004] O. Wendell, "CCNA INTRO Exam Certification Guide: CCNA Self-study", Cisco Press, 2nd edition. Isbn:1587200945 (2004).

[Wilkes 2006] M. Wilkes, "What I remember of the ENIAC," *IEEE Annals of the History of Computing*, vol. 28 (2), pp. 30-31, (2006).

[Wils 2002] A. Wils, F. Matthijs, Y. Berbers, T. Holvoet and K. De Vlaminck, "Device discovery via residential gateways", Proceedings of the *2002 Digest of Technical Papers. International Conference on Consumer Electronics, 18-20 June 2002*, Los Angeles, CA, USA, pp. 96-7, (2002// 2002)

[Wolf 2007] W. Wolf, "Chapter 4: Processes and Operating Systems," in *High-Performance Embedded Computing: Architectures, Applications, and Methodologies*, Morgan Kaufmann, 1st edition. Isbn:012369485X (2007).

[Wu 2007] J. Wu and J. Dong, "Simple service discovery and configuration protocol for embedded devices", Proceedings of the *International Conference on Communication Technology (ICCT '06)*, Guilin, China, pp. 201-204, (27-30 November 2007)

[Yajun 2007] G. Yajun and W. Yulin, "Establishing trust relationship in mobile ad-hoc network", Proceedings of the *International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'07)*, Shanghai, China, pp. 1562-1564, (21-25 September 2007)

[Yan Lindsay 2006] S. Yan Lindsay, Y. Wei, H. Zhu and K. J. R. Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24 (2), pp. 305-17, (2006).

[Yeager 2002] W. Yeager and J. Williams, "Secure peer-to-peer networking: the JXTA example," *IT Professional*, vol. 4 (2), pp. 53-57, (2002).

[Yick 2008] J. Yick, B. Mukherjee and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52 (12), pp. 2292-2330, (2008).

[Yohan 2007] R. Yohan, K. Hangkyu, K. Hak Soo, H. Myoung and S. Jin Hyun, "Semantic business registry information model", Proceedings of the *International Conference on Convergence Information Technology (ICCIT '07)*, Gyeongju, South Korea, pp. 2142-2145, (21-23 November 2007)

[Youxiang 2008] D. Youxiang, B. Yongtang, P. Lijiang, Y. Beibei, X. Jiuyun and S. Nianyun, "A secure Web services model based on the combination of SOAP registration info and token proxy", Proceedings of the *International Symposium on Computer Science and Computational Technology (ISCSCT'08)*, Shanghai, China, pp. 15-20, (20-22 December 2008)

[Zebin 2007] C. Zebin and S. Fickas, "Do no harm: model checking eHome applications", Proceedings of the *First International Workshop on Software Engineering for Pervasive Computing Applications, Systems, and Environments (SEPCASE '07)*, Minneapolis, MN, USA, pp. 8, (20-26 May 2007)

[Zeeb 2007] E. Zeeb, A. Bobek, H. Bohn and F. Golatowski, "Service-oriented architectures for embedded systems using devices profile for Web services", Proceedings of the *21st International Conference on Advanced Information Networking and Applications Workshops/Symposia,*, Niagara Falls, ON, Canada, pp. 956-963, (21-23 May 2007)

[Zeeb 2007] E. Zeeb, A. Bobek, H. Bonn and F. Golatowski, "Lessons learned from implementing the devices profile for Web services", Proceedings of the *Inaugural IEEE International Conference on Digital Ecosystems and Technologies*, Cairns, Australia, pp. 229-232, (21-23 Febrary 2007)

APPENDICES

APPENDIX A: ADHOCGS FRAMEWORK USE CASE MODEL

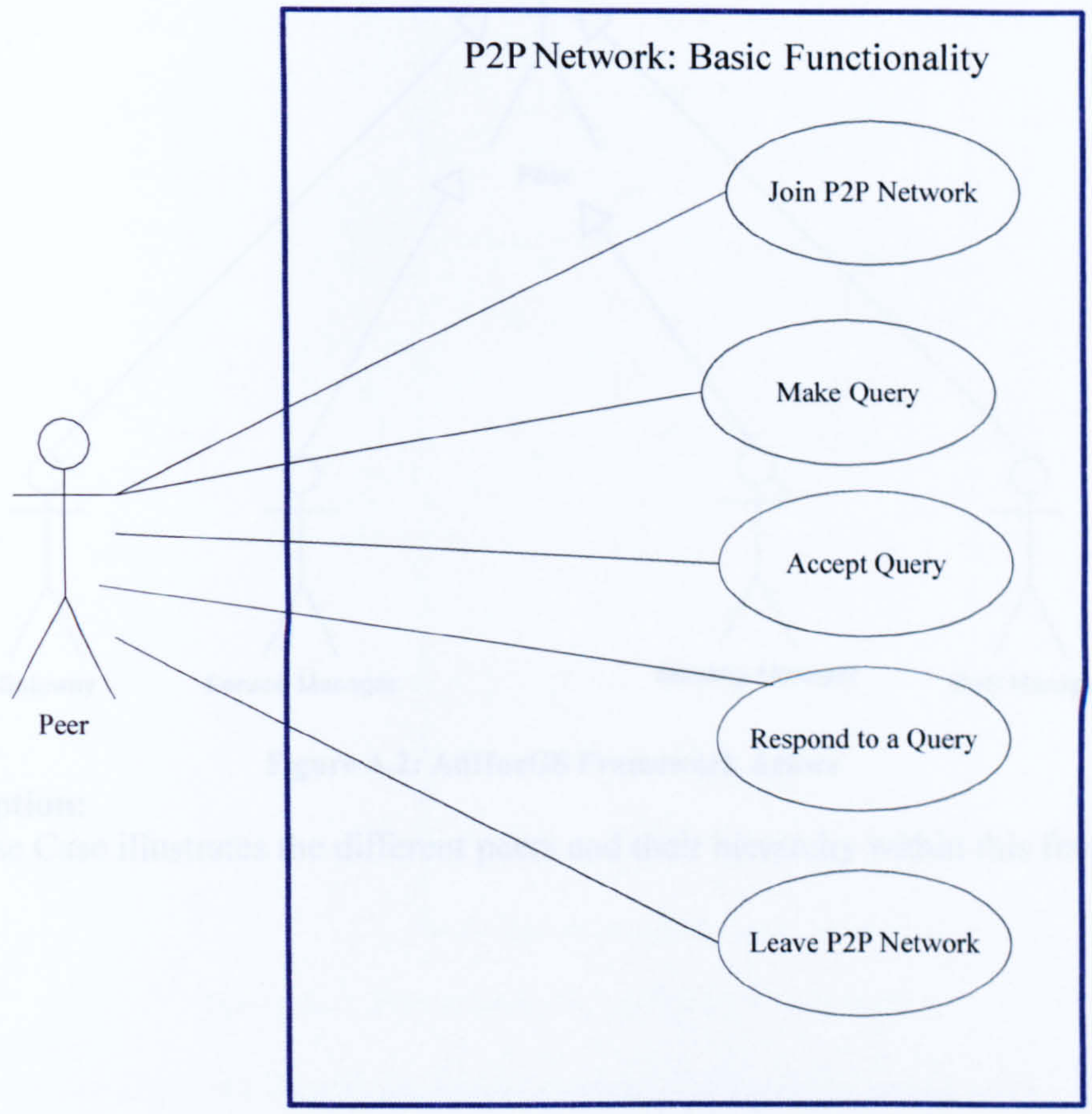

Figure A.1: P2P System Basic Functionality

Description:
This Use Case illustrates the typical functionality of a Typical peer within a P2P system.

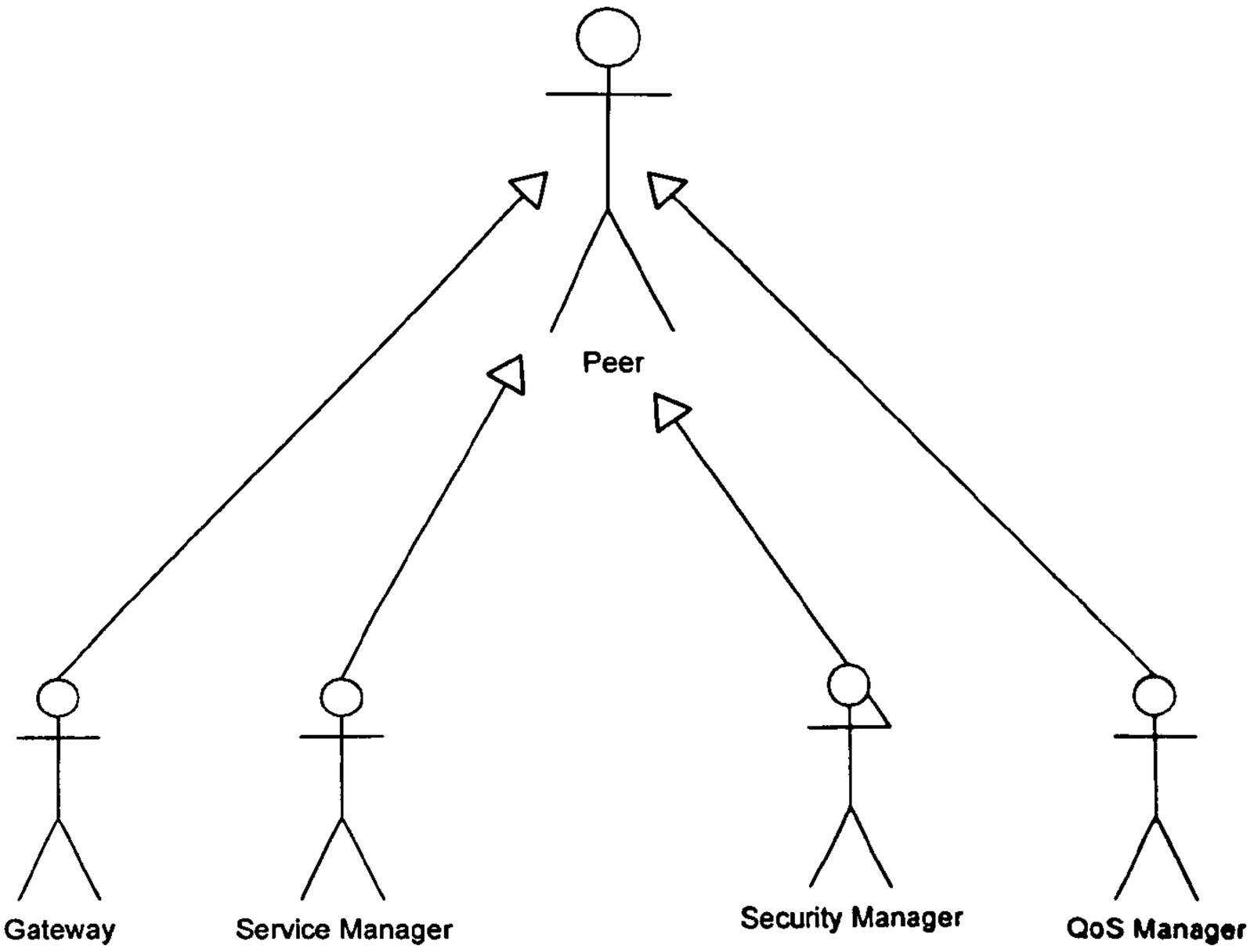

Figure A.2: AdHocGS Framework Actors

Description:

This Use Case illustrates the different peers and their hierarchy within this framework.

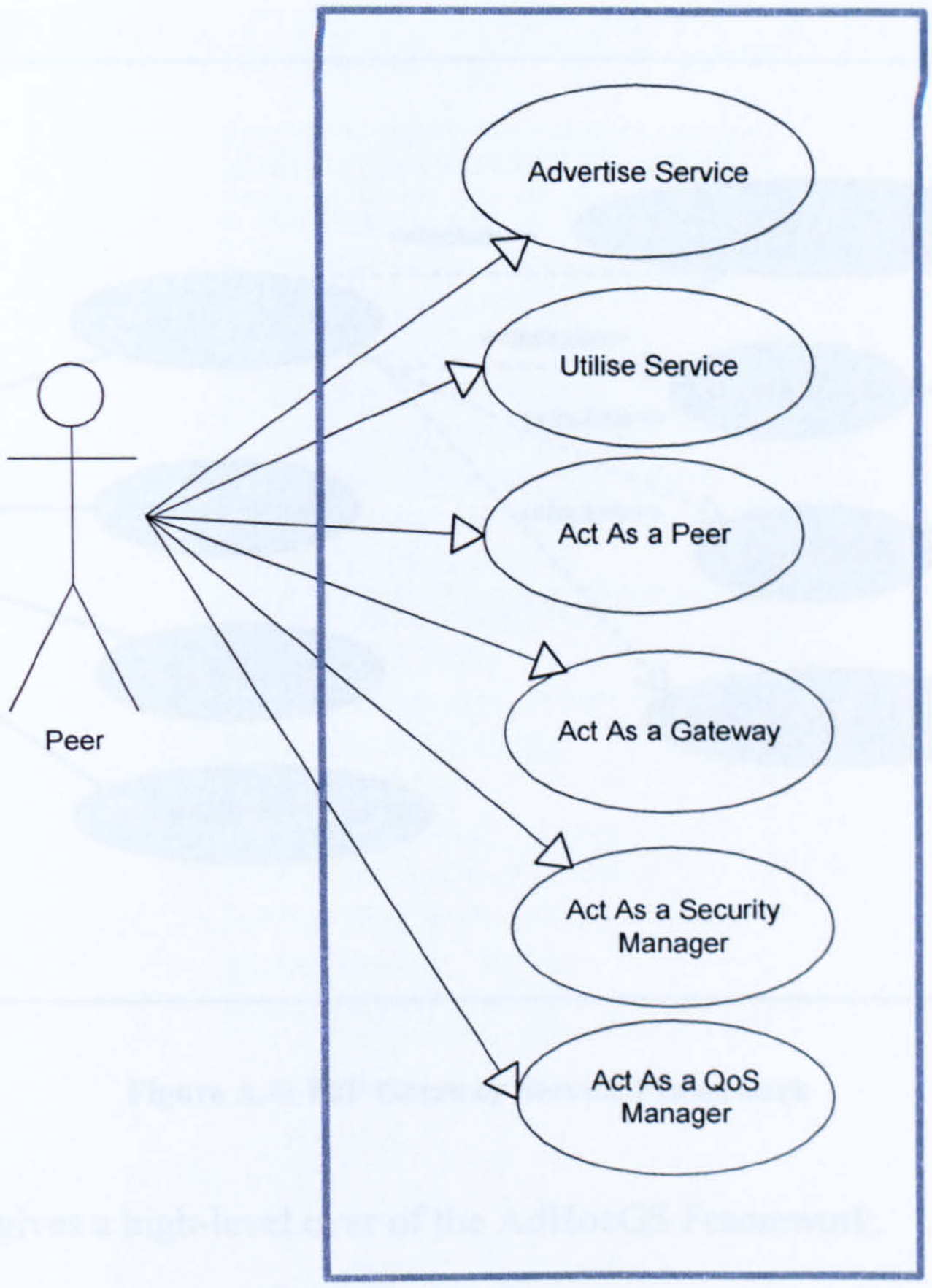

Figure A.3: Peer roles in the AdHocGS Framework

Description:

This Use Case Diagram illustrates different roles of peer in this framework.

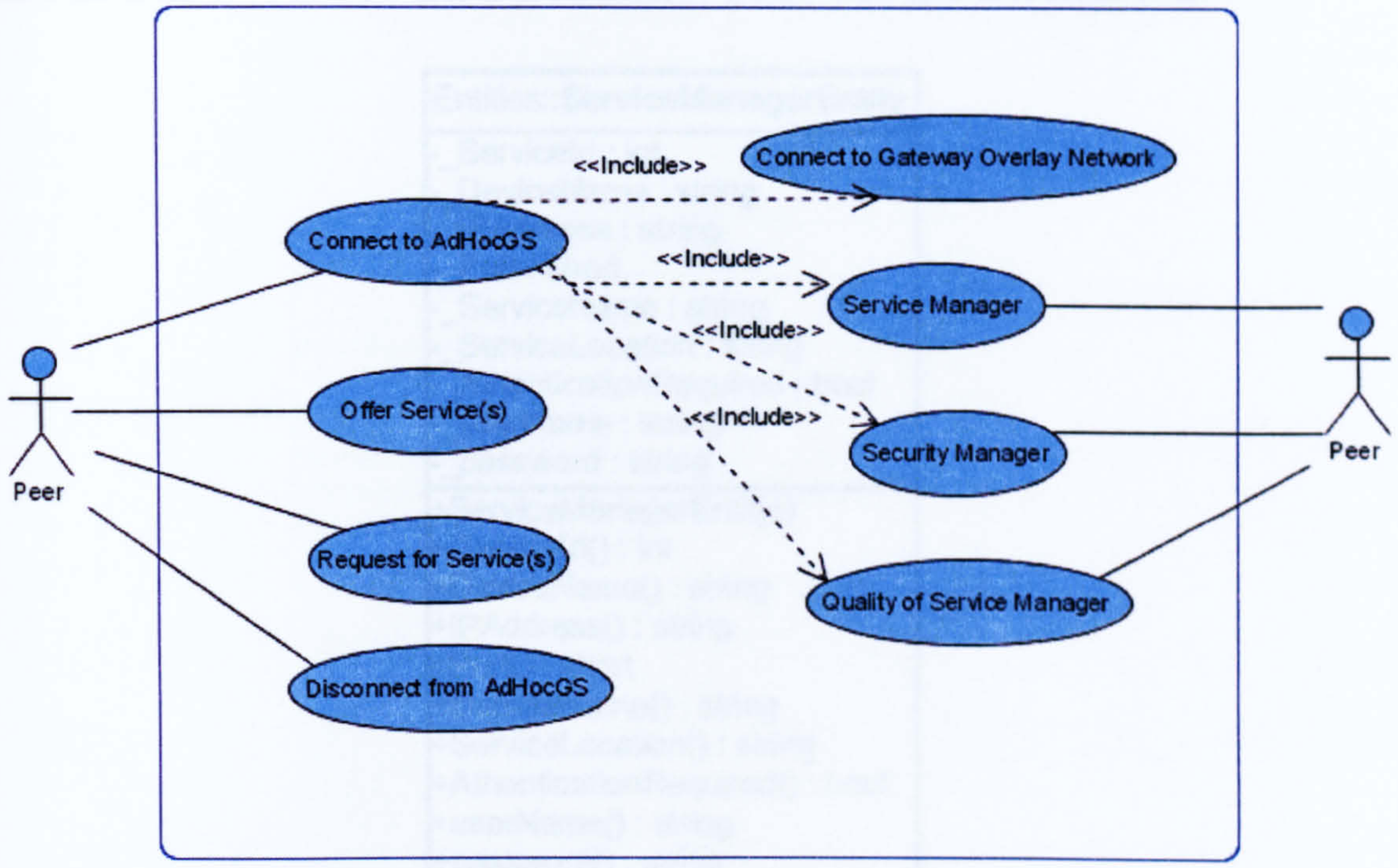

Figure A.4: P2P Gateway Service Framework

Description:

This Use Case gives a high-level over of the AdHocGS Framework.

APPENDIX B: ADHOCGS FRAMEWORK CLASS DIAGRAMS

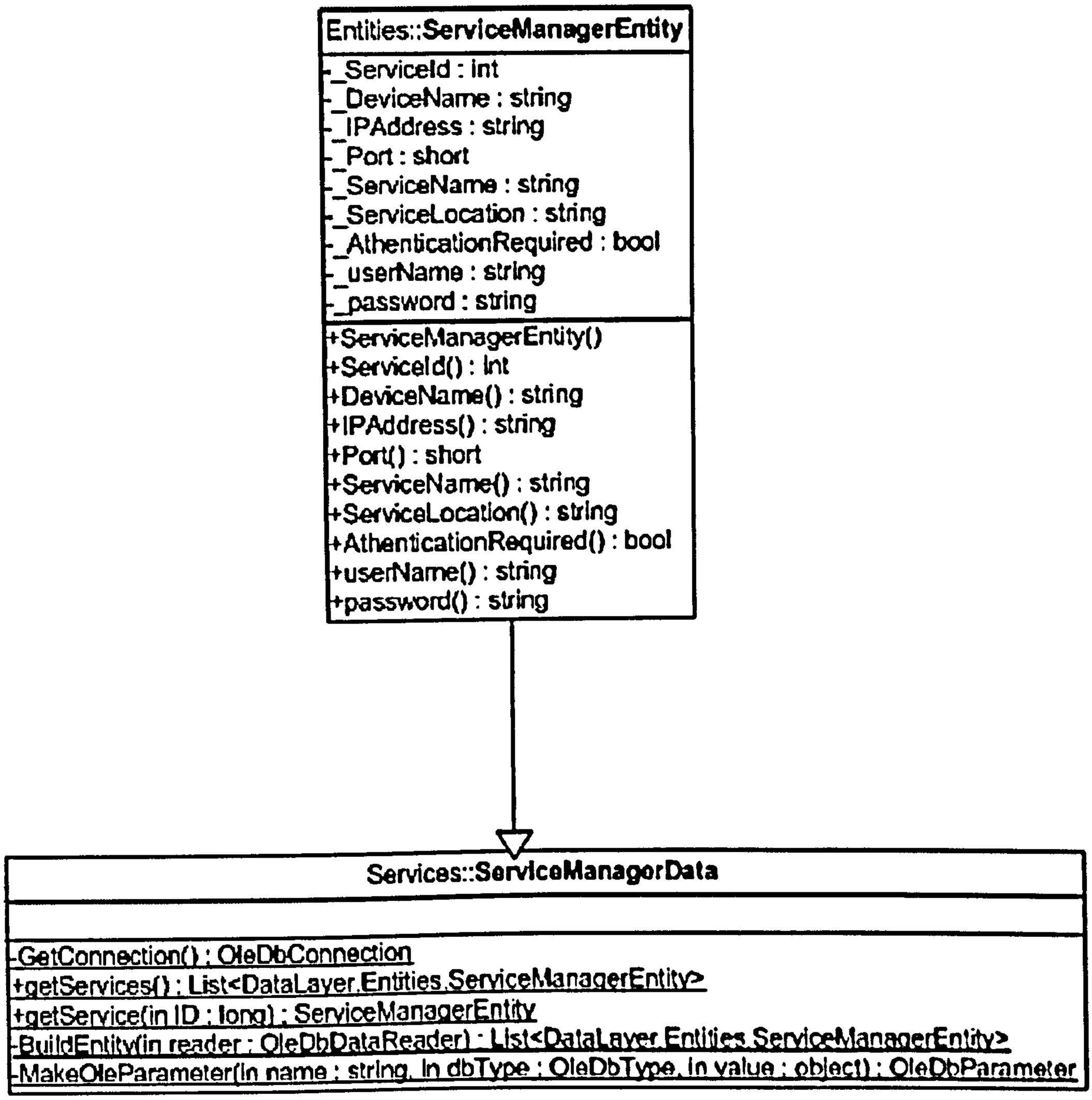

Figure B.1: AdHocGS Framework Service Manager

Description:
This Class Diagram illustrates the classes used to implement Service Manager within this framework

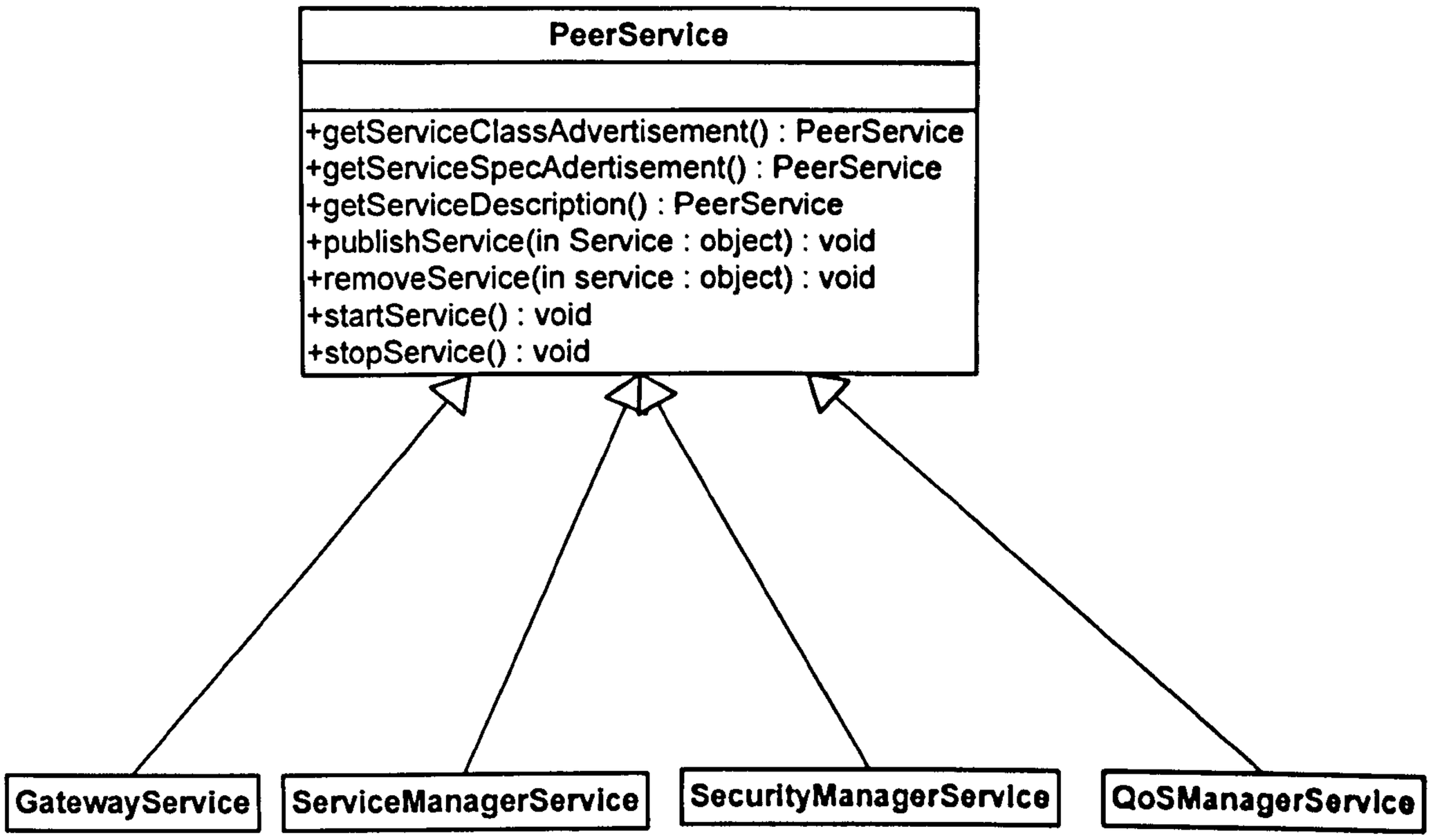

Figure B.2: Peer Services

Description:
This Class Diagram illustrates the relationships between the various services within this framework

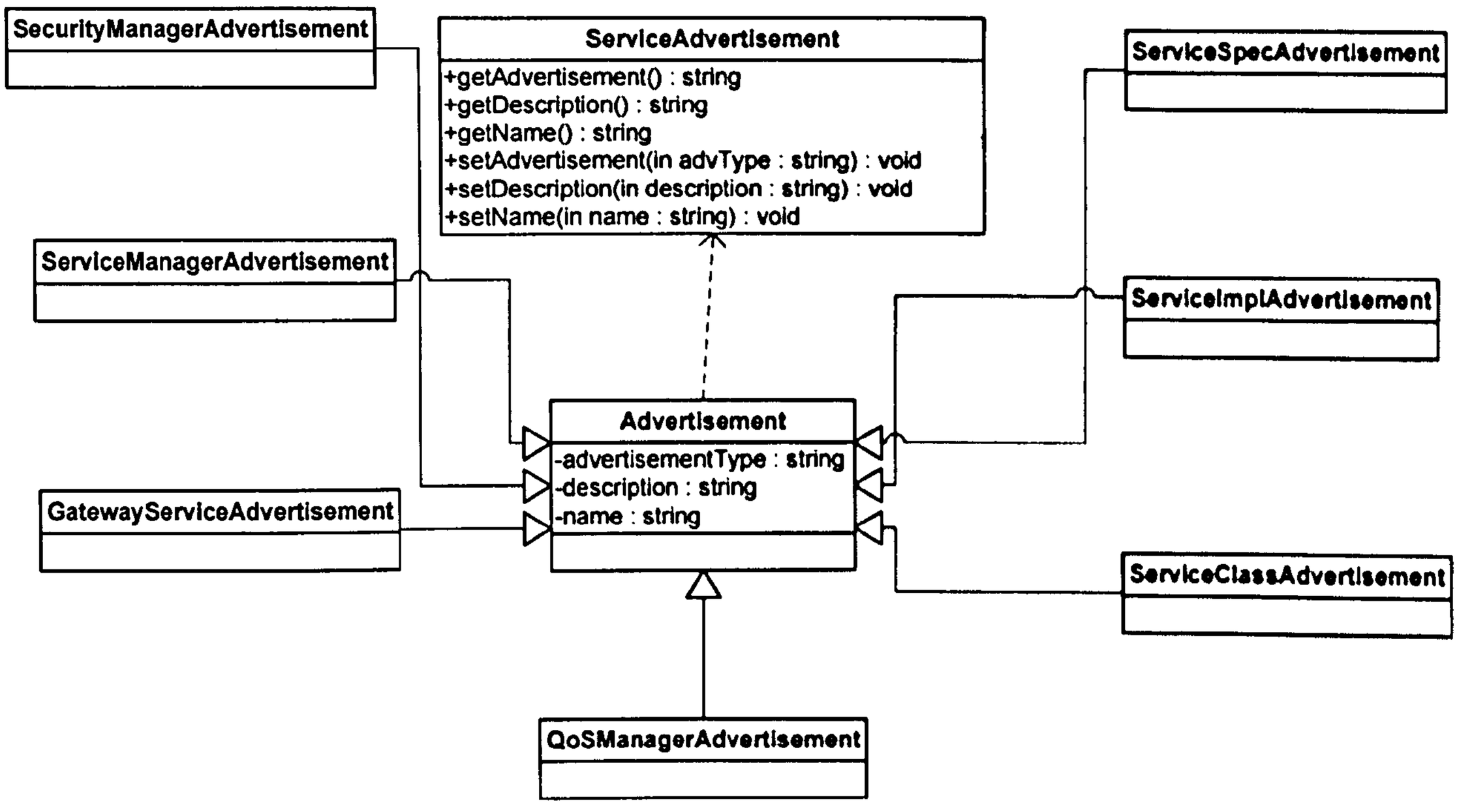

Figure B.3: Service Advertisement

Description:
This Class Diagram illustrates the required classes to create Service Advertisements within this framework

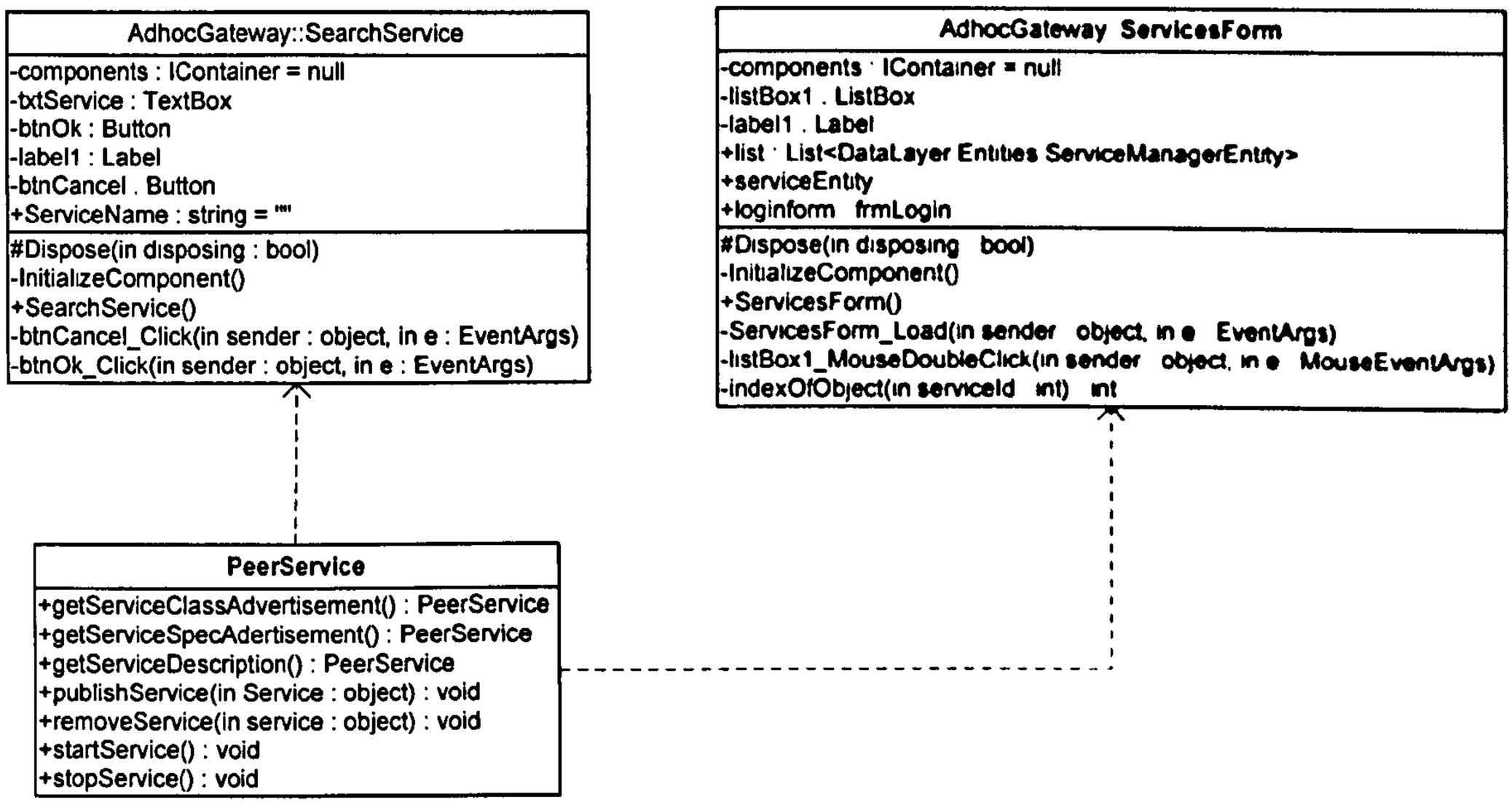

Figure B.4: Peer searching for service

Description:

This Class Diagram illustrates the classes required for the peer searching for a service in this framework.

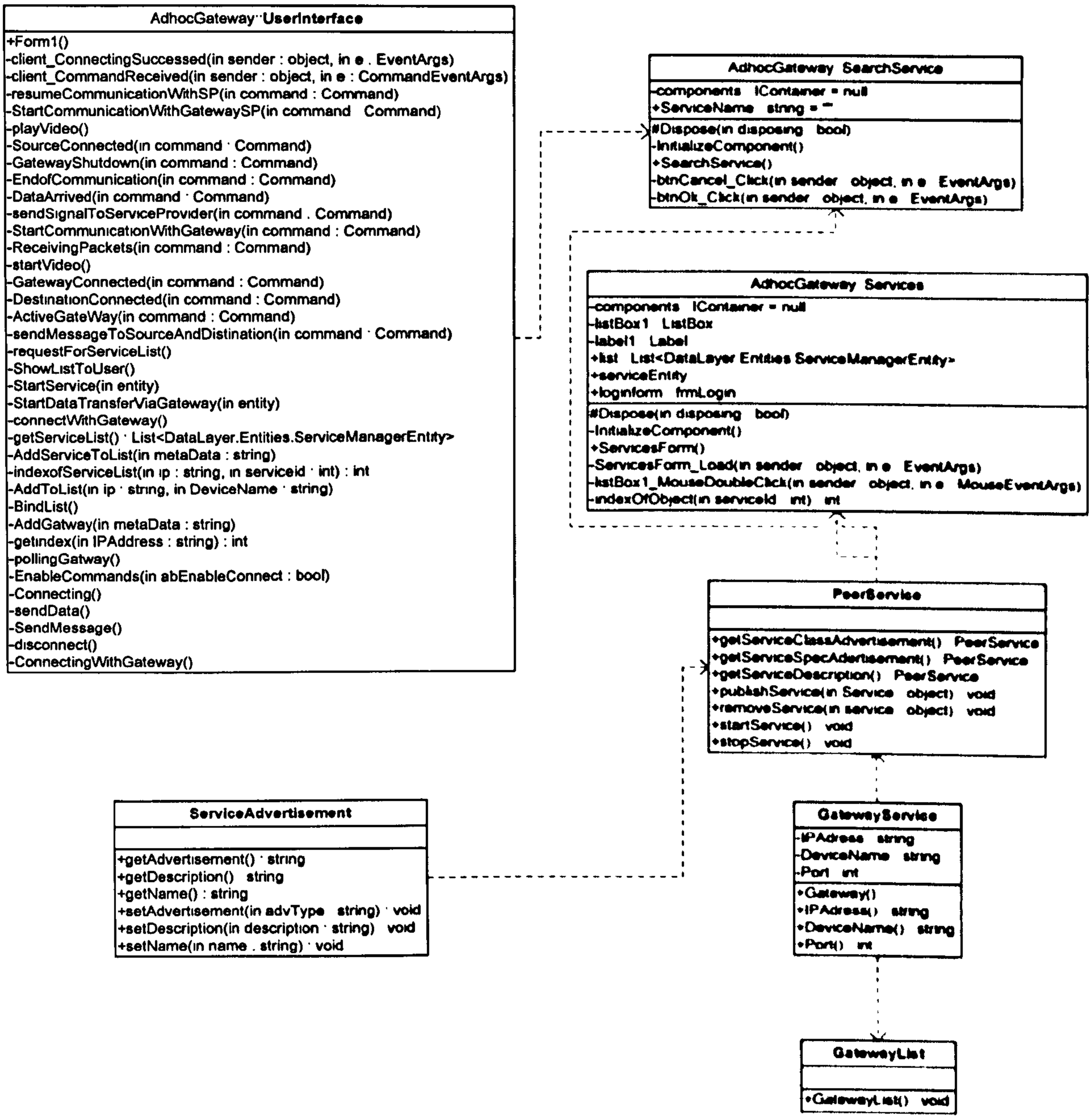

Figure B.5: Service Interface Model

Description:

This Class Diagram illustrates the classes required to create Service Interface Model within this framework.

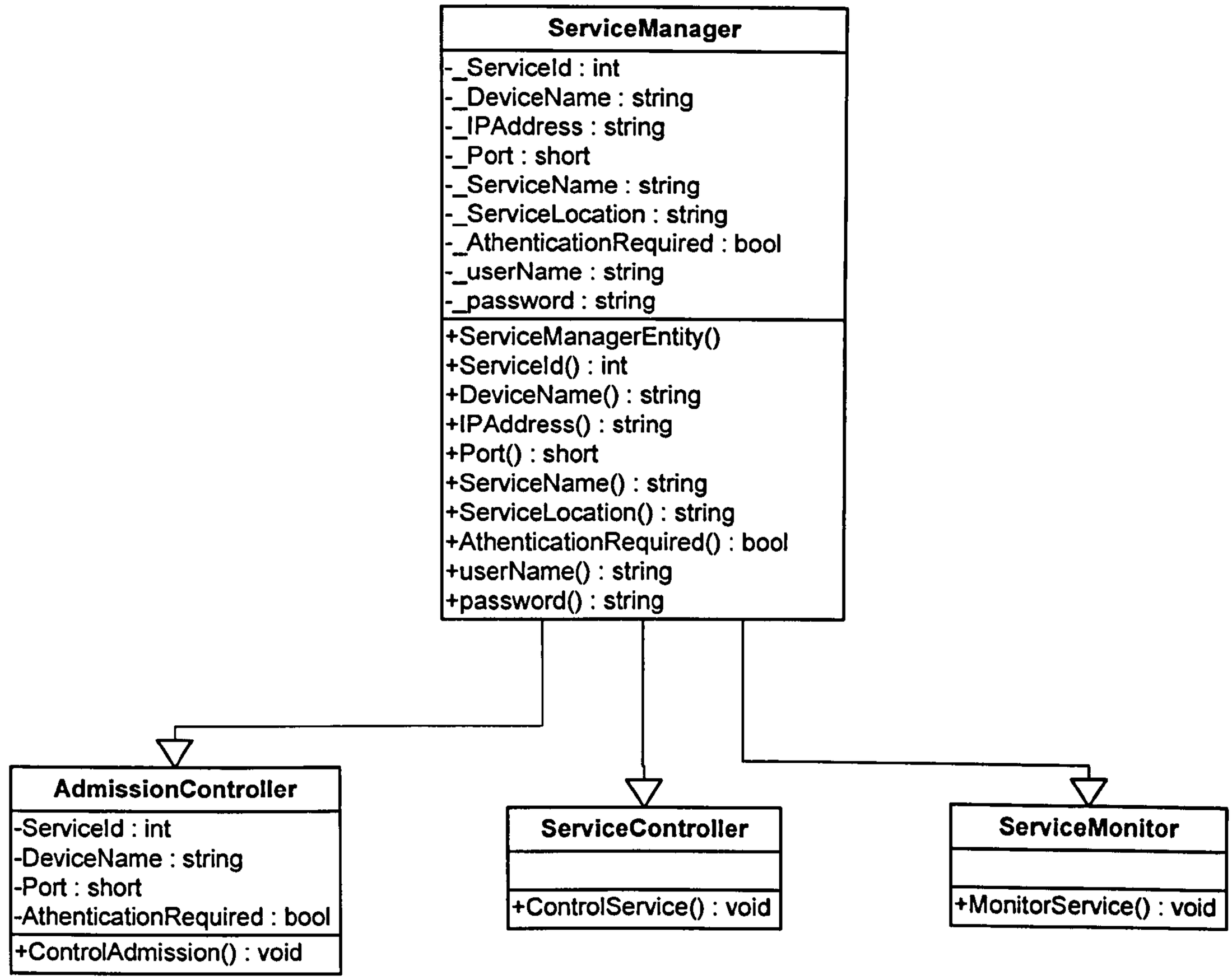

Figure B.6: Service Manager

Description:
This Class Diagram illustrates the classes required to create Service Manager within this framework.

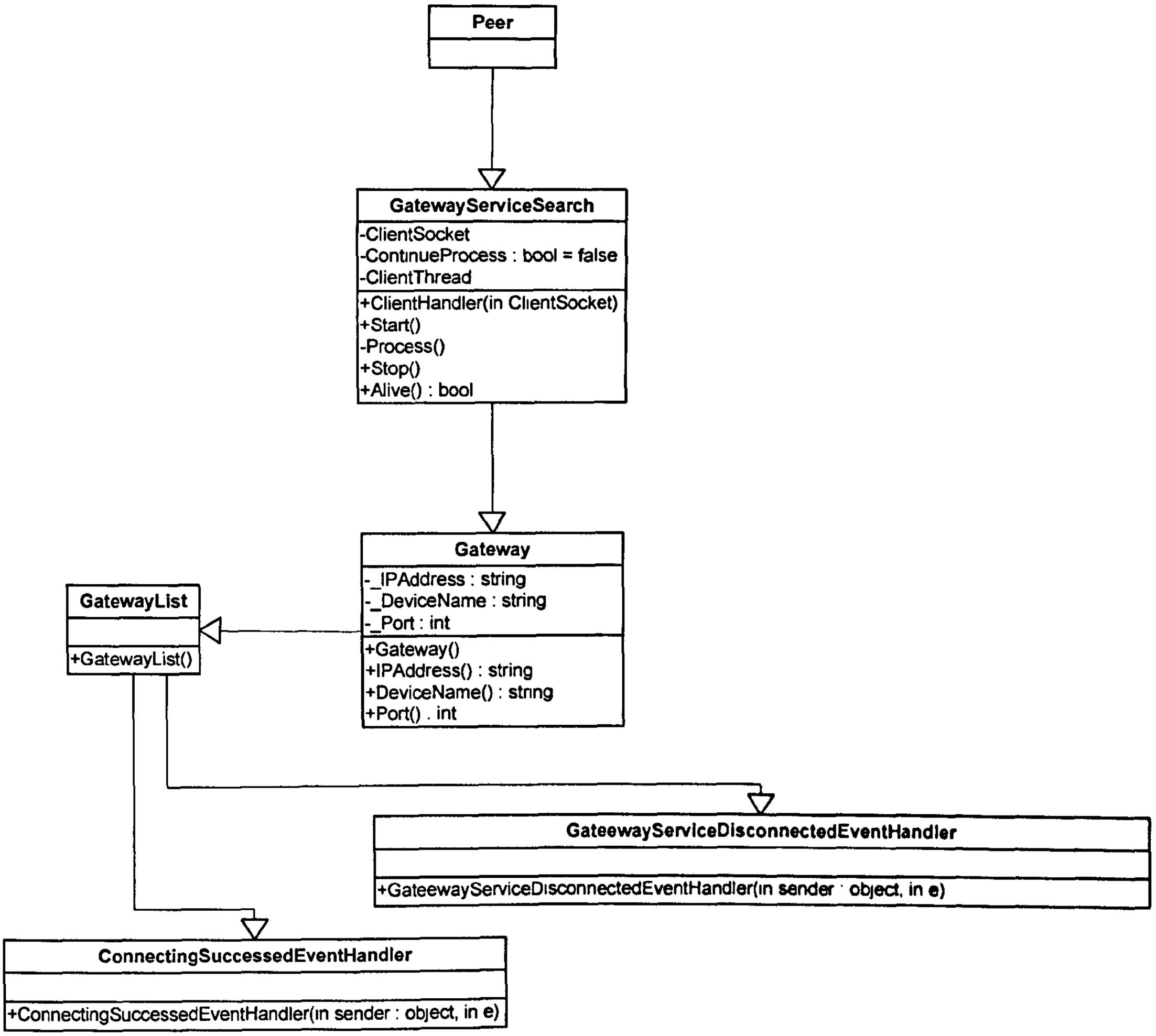

Figure B.7: Gateway Service

Description:
This Class Diagram illustrates the classes required to search for Gateway Service within this framework.

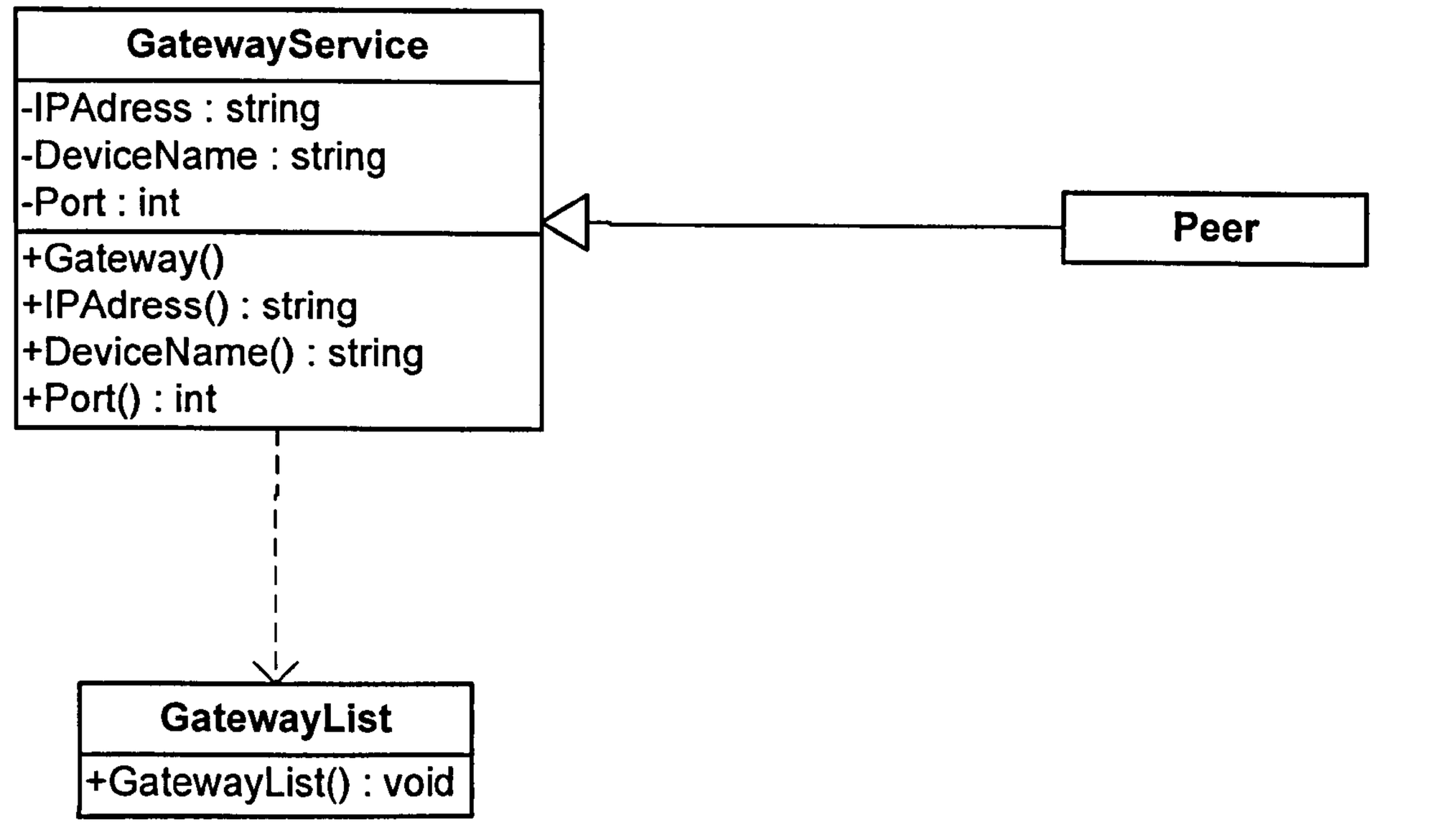

Figure B.8: Gateway *searching*

Description:
This Class Diagram illustrates the classes required for peer searching for Gateway Service in this framework

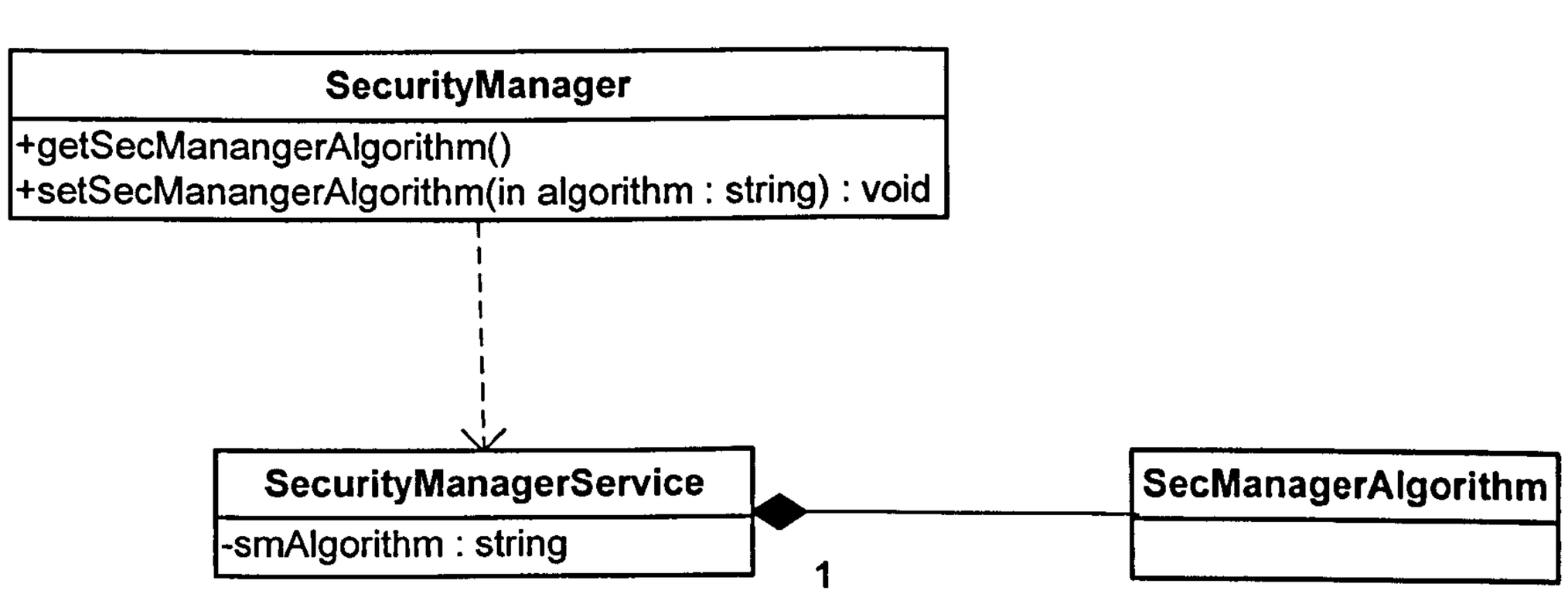

Figure B.9: Security Manager

Description:
This Class Diagram illustrates the classes required to create the Security Manager Service within this framework.

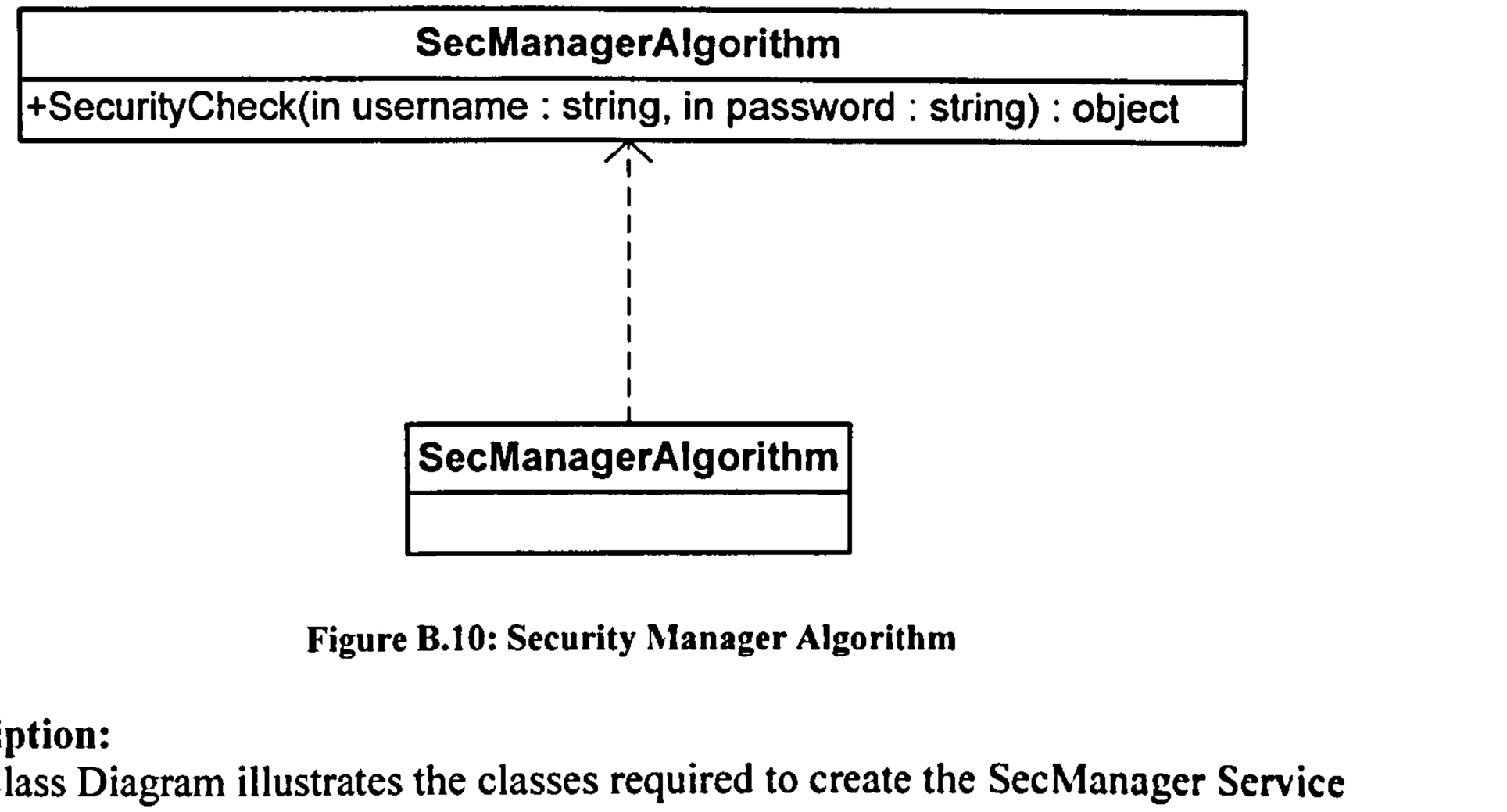

Figure B.10: Security Manager Algorithm

Description:
This Class Diagram illustrates the classes required to create the SecManager Service within this framework.

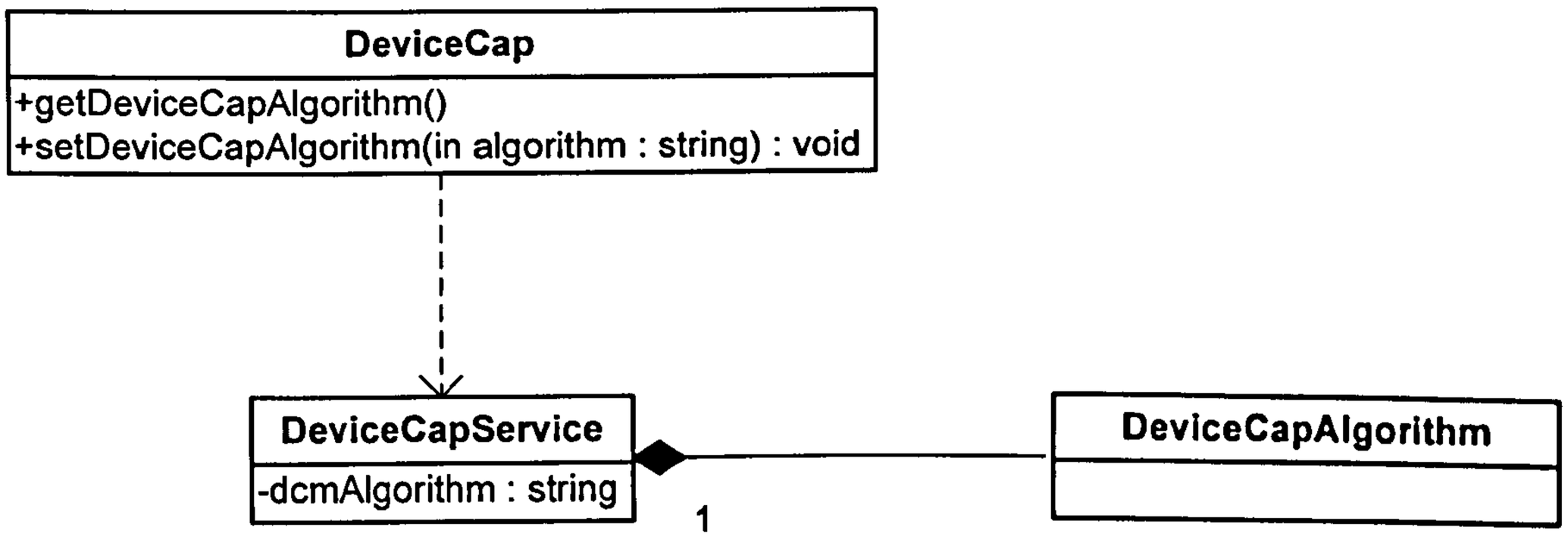

Figure B.11: Device Capability Service

Description:
This Class Diagram illustrates the classes required to create the DeviceCap Service within this framework.

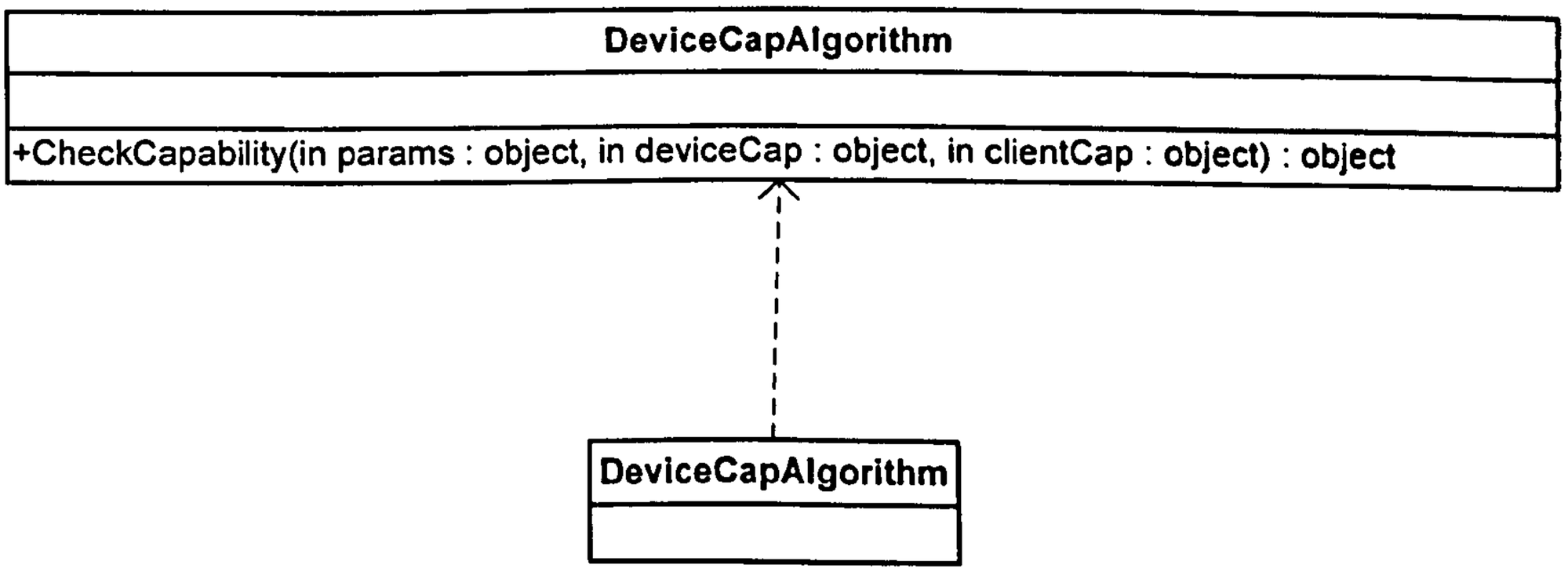

Figure B.12: Device Capability Algorithm

Description:
This Class Diagram illustrates the classes required to create the DeviceCap Algorithm within this framework.

APPENDIX C: ADHOCGS FRAMEWORK ACTIVITY DIAGRAMS

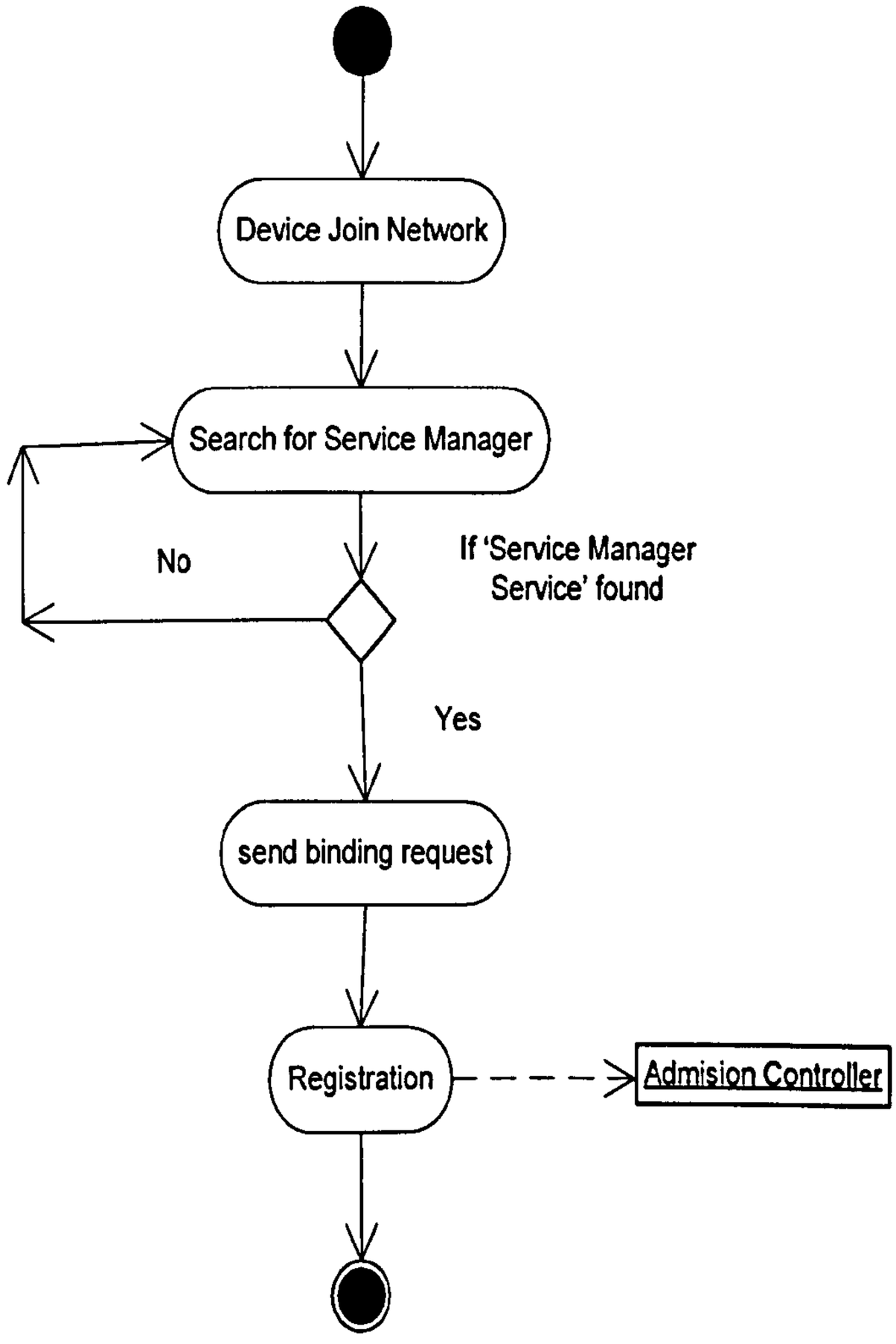

Figure C.1: Service Registration Activity Diagram

Description:

This Activity Diagram illustrates Service Registration within this framework.

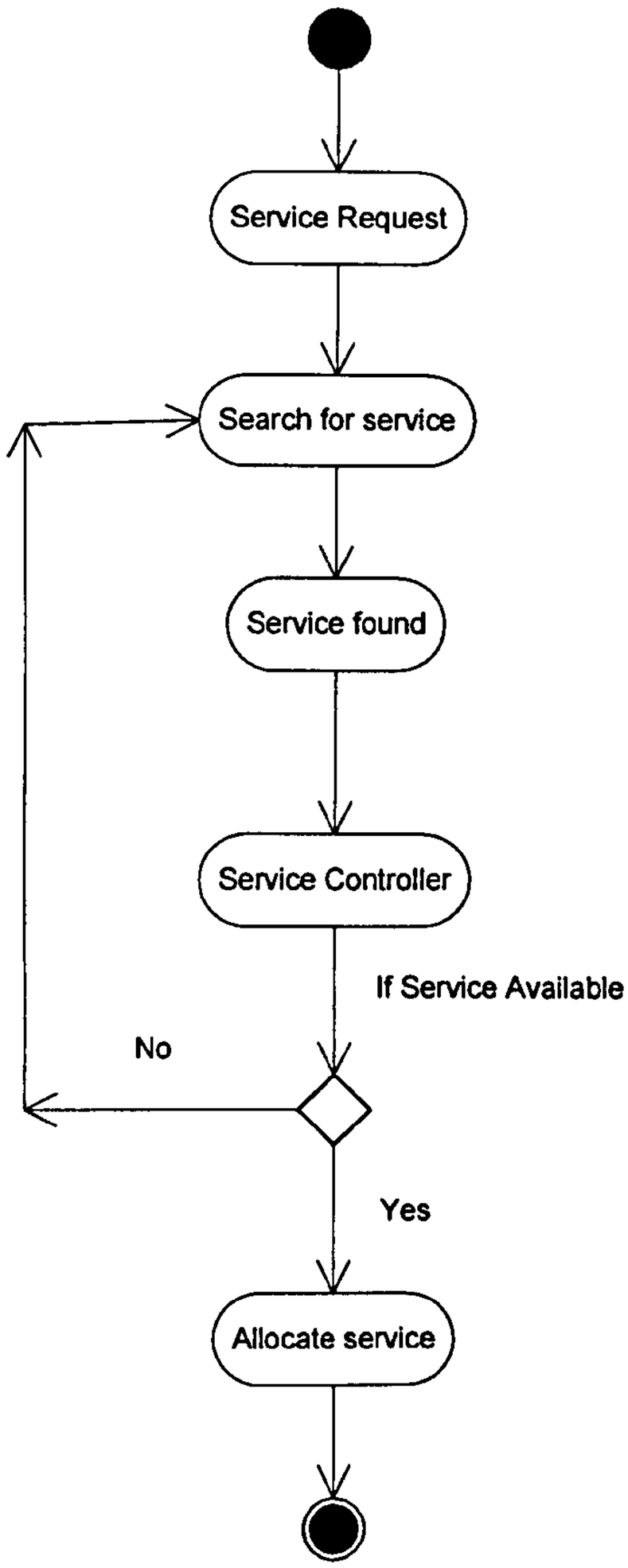

Figure C.2: Service Controller Activity Diagram

Description:

This Activity Diagram illustrates Service Controller within this framework.

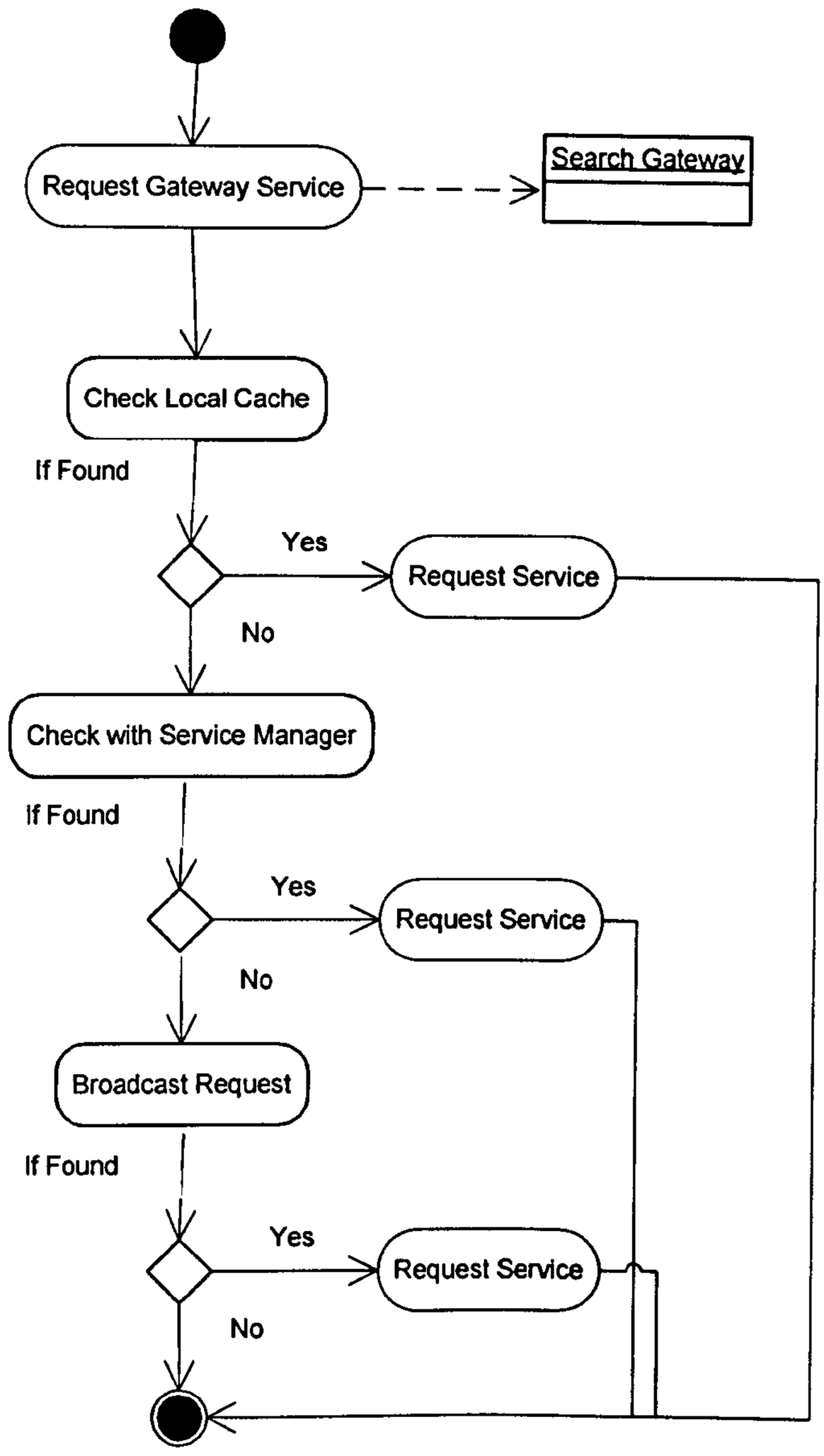

Figure C.3: Gateway Discovery Activity Diagram

Description:

This Activity Diagram illustrates Gateway Discovery within this framework.

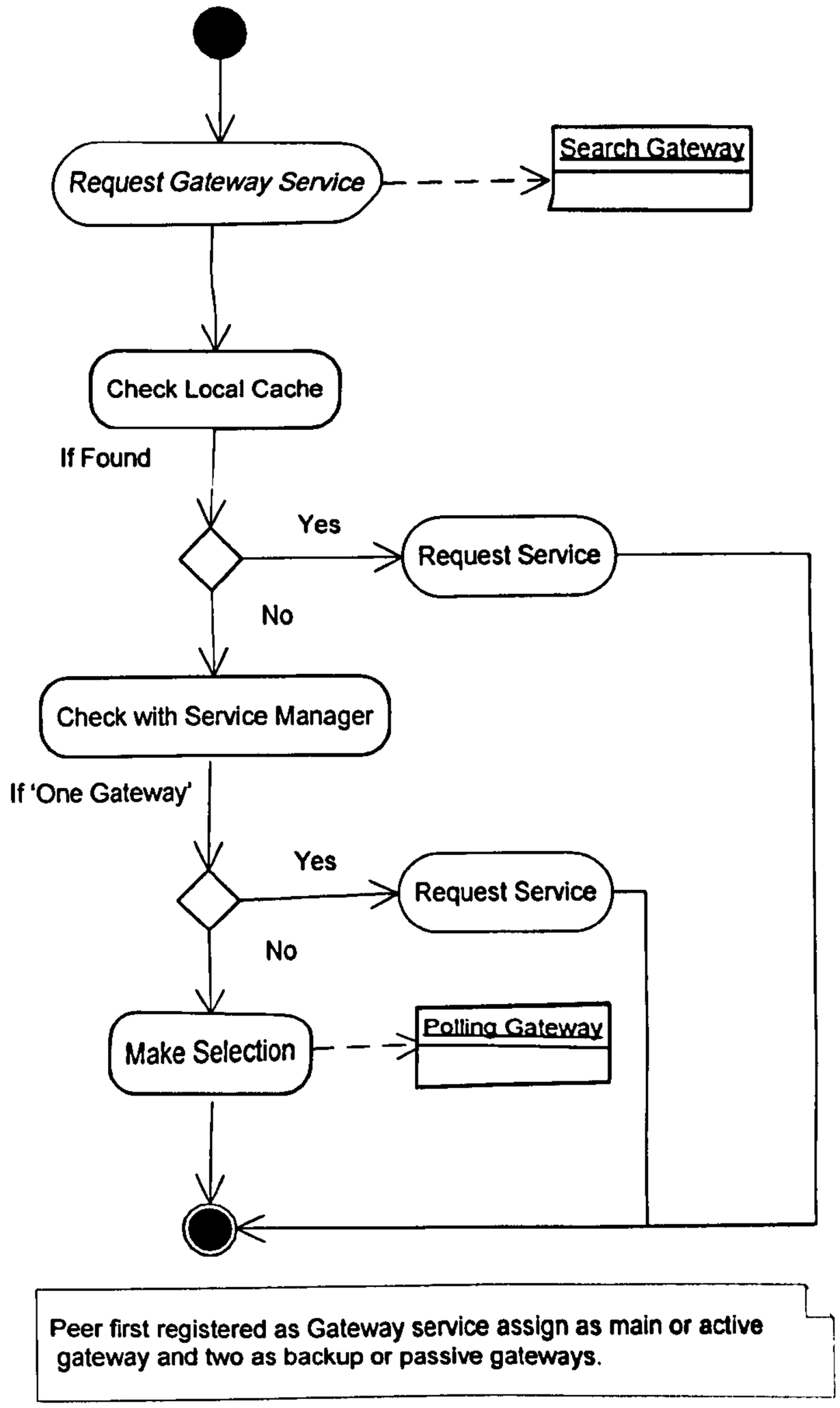

Figure C.4: Gateway Selection Activity Diagram

Description:

This Activity Diagram illustrates Gateway Selection within this framework.

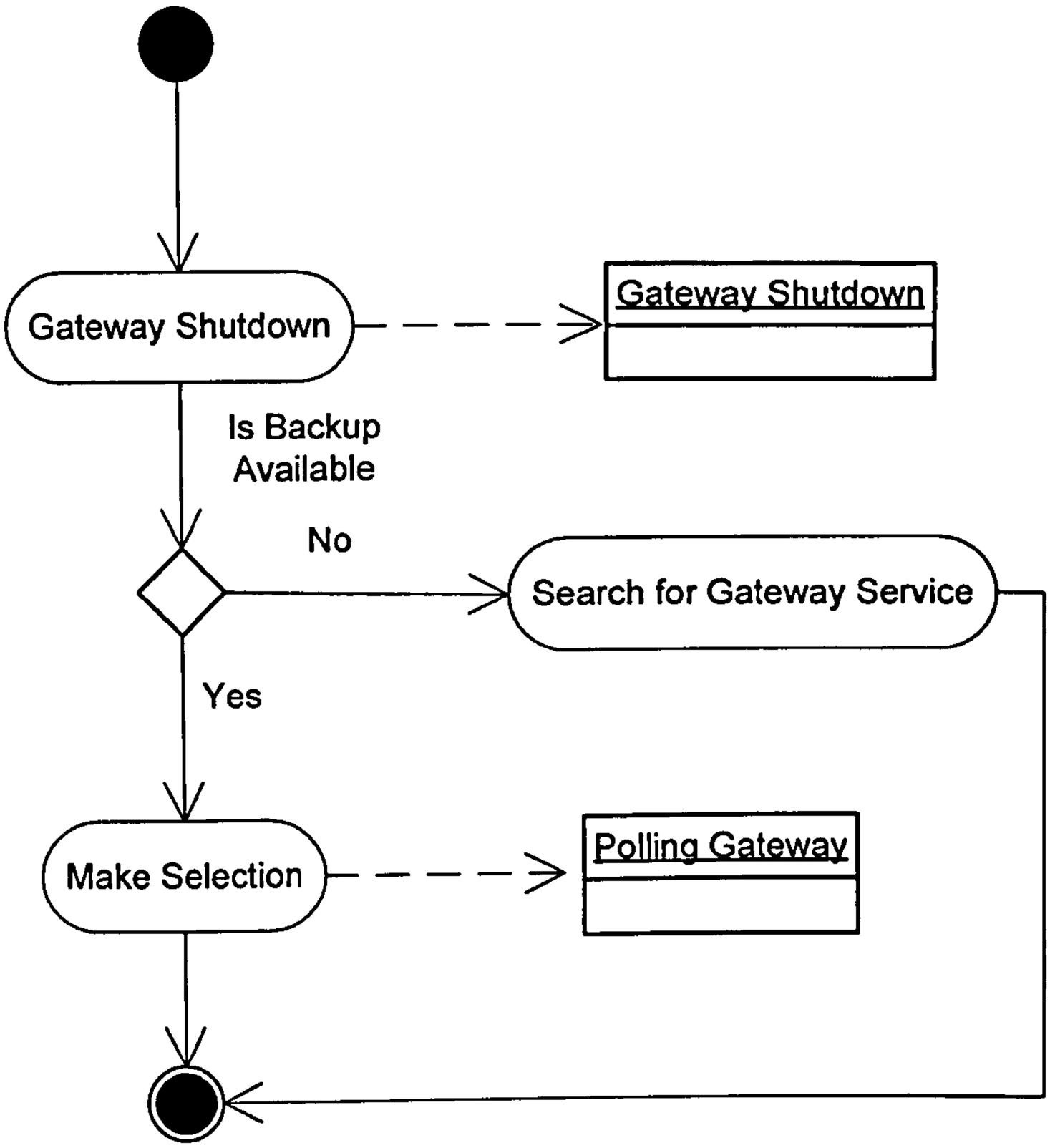

Figure C.5: Alternative Gateway Search Activity Diagram

Description:

This Activity Diagram illustrates Alternative Gateway Search within this framework.

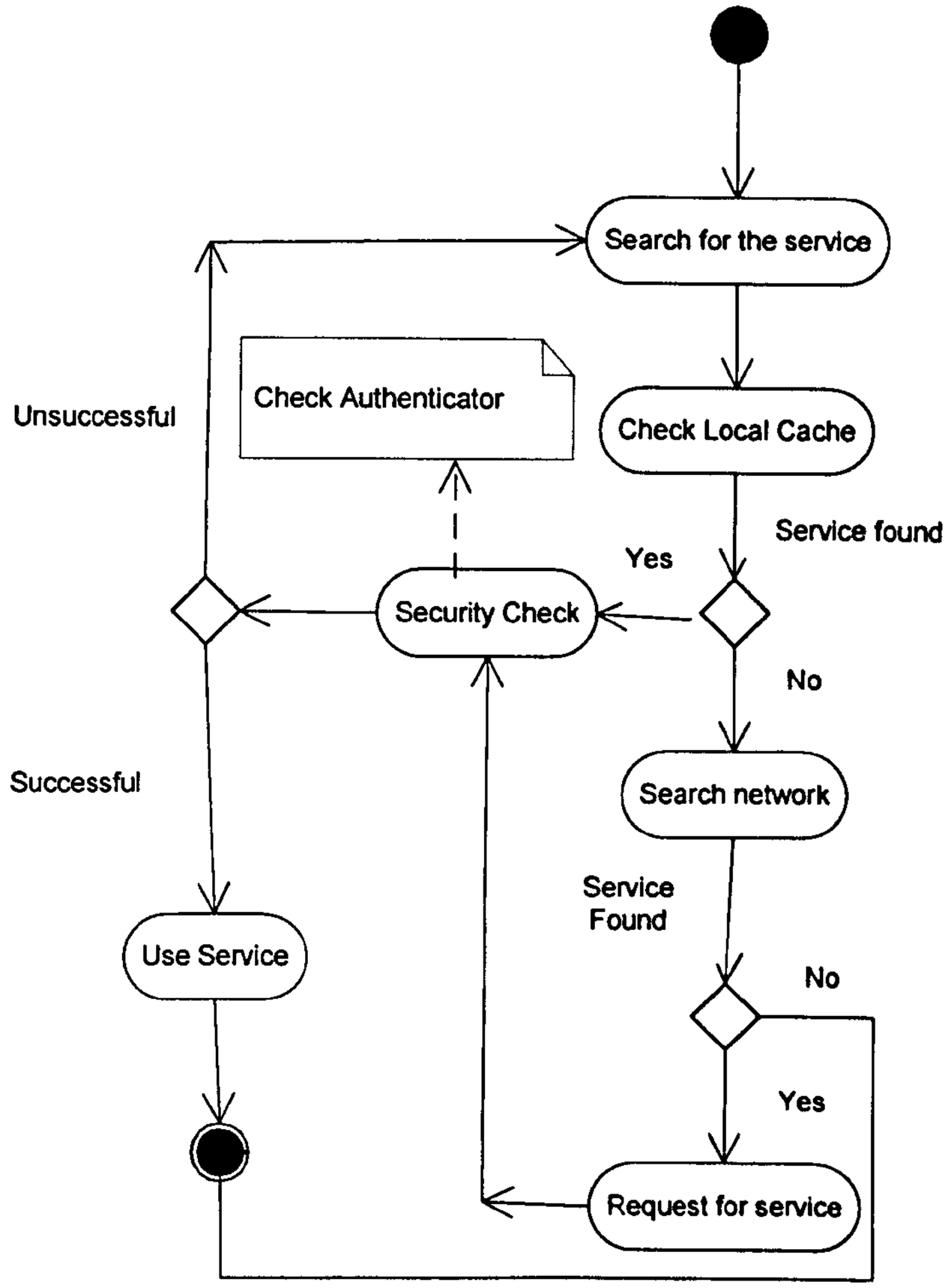

Figure C.6: Security Check Activity Diagram

Description:

This Activity Diagram illustrates Security Check within this framework.

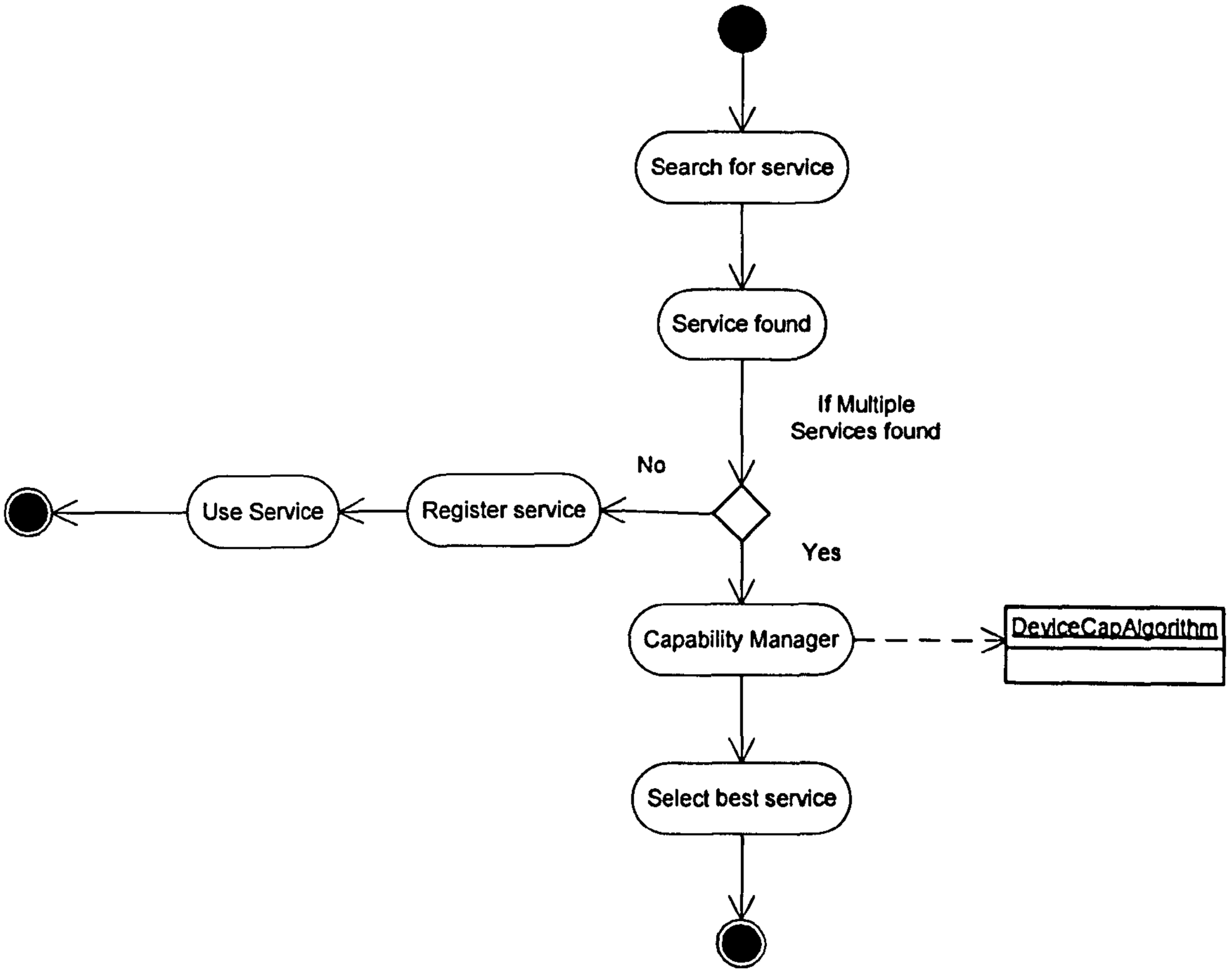

Figure C.7: Device Capability Matching Activity Diagram

Description:

This Activity Diagram illustrates Device Capability Matching within this framework.

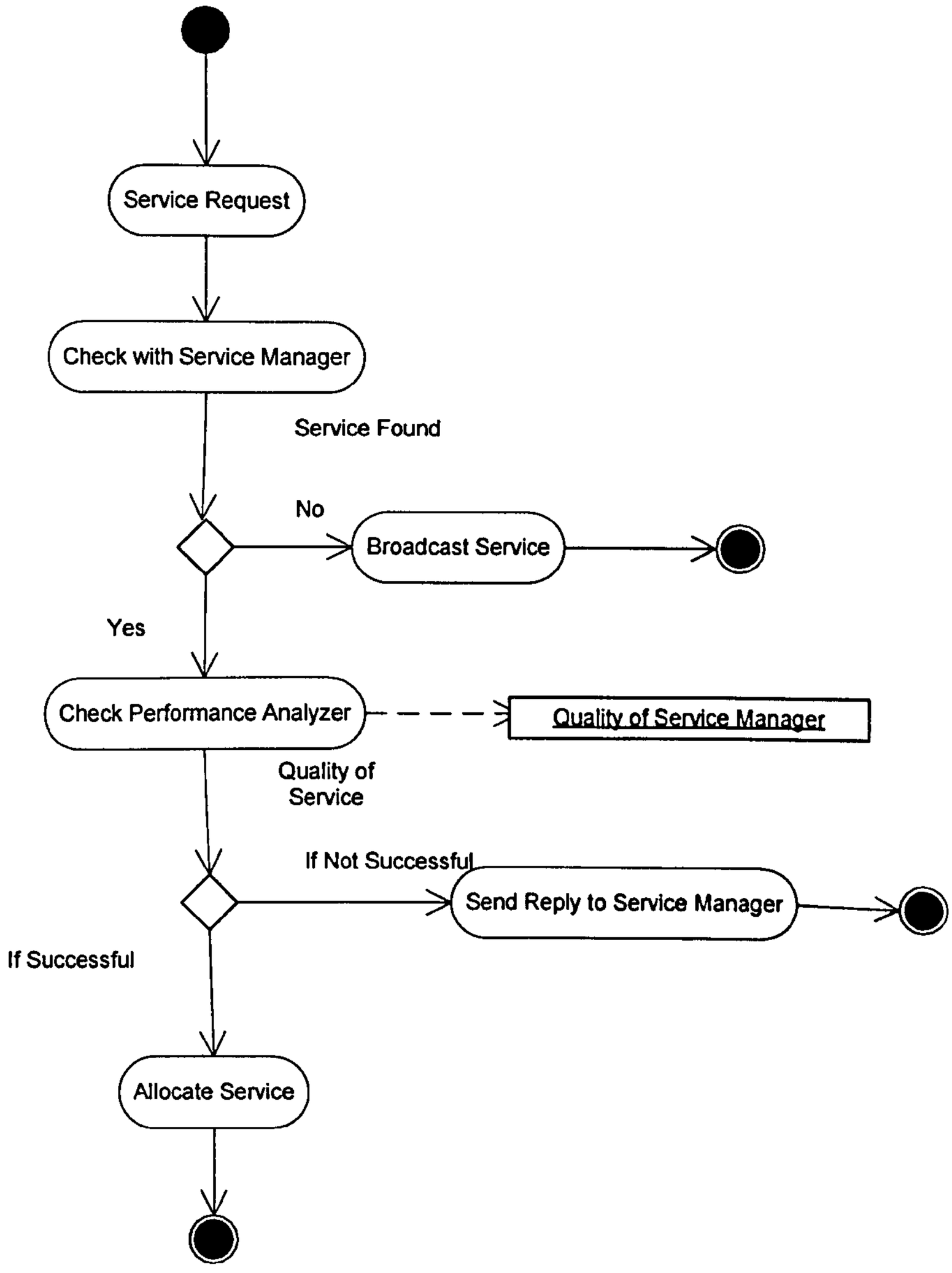

Figure C.8: Performance Analyser Activity Diagram

Description:

This Activity Diagram illustrates Performance Analyser within this framework.

APPENDIX D: SEQUENCE AND STATE TRANSITION DIAGRAMS

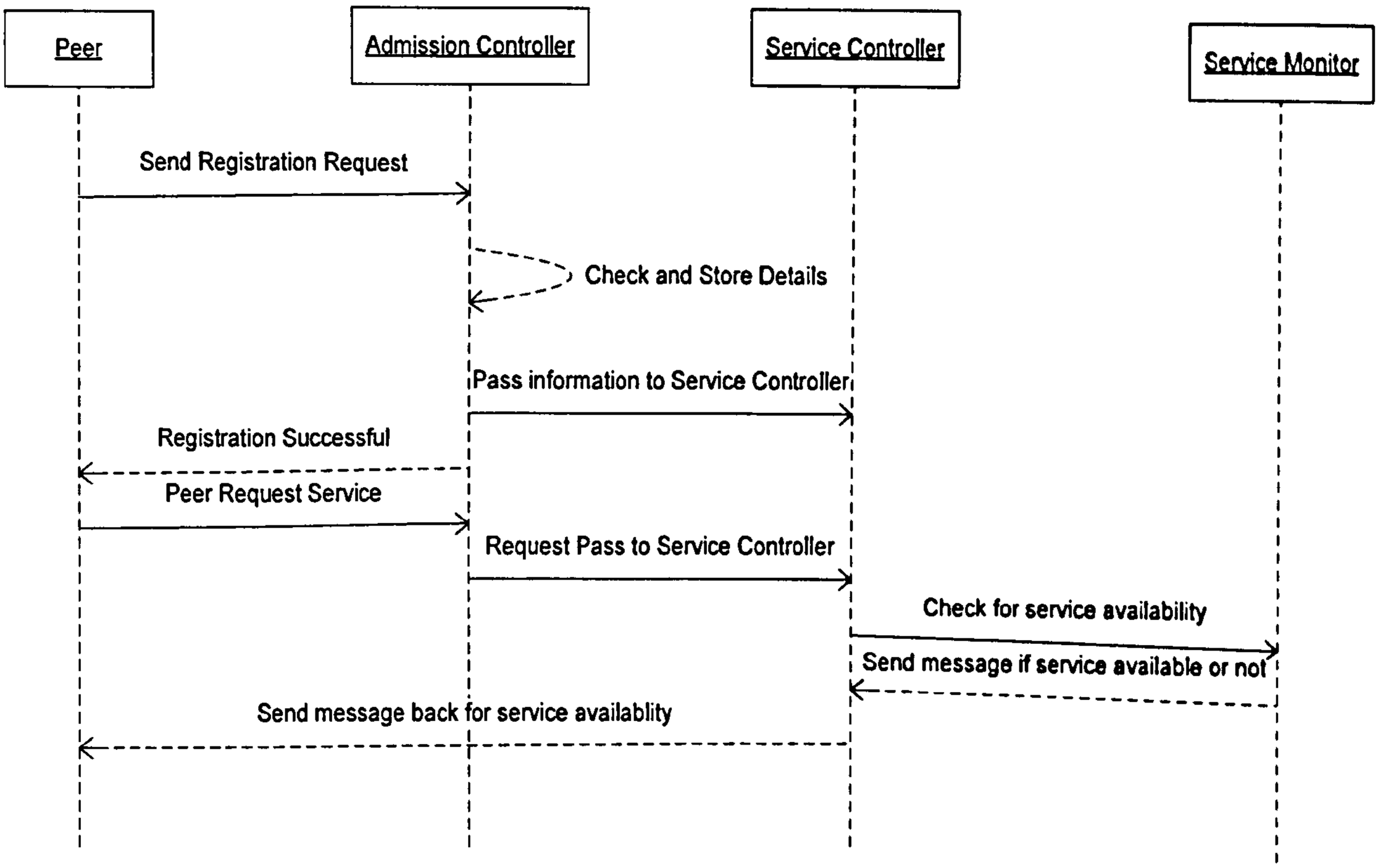

Figure D.1: Service Manager Sequence Diagram

Description:

This Sequence Diagram illustrates Service Manager within this framework.

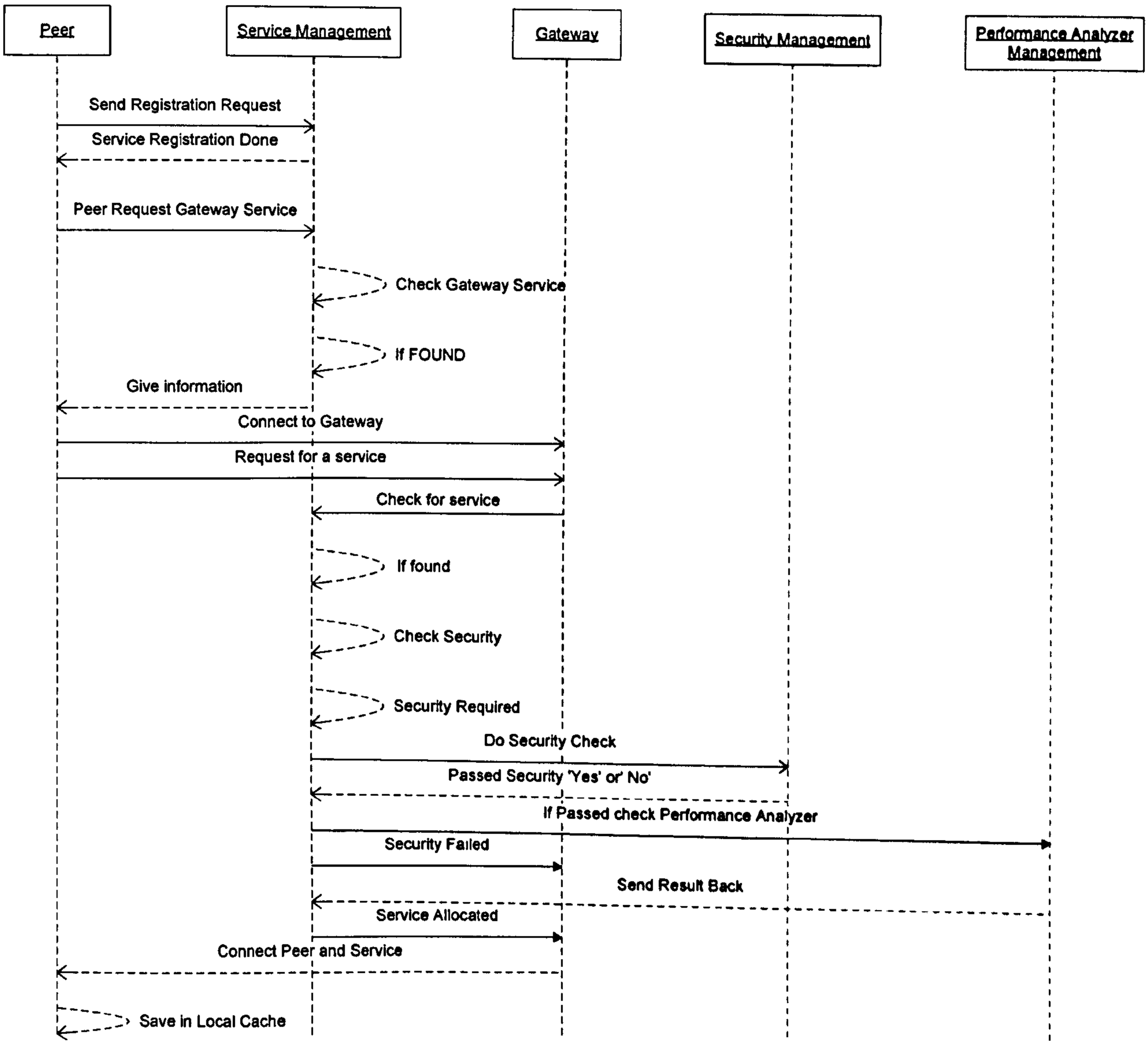

Figure D.2: AdHocGS Framework Sequence Diagram

Description:

This Sequence Diagram illustrates AdHocGS Framework.

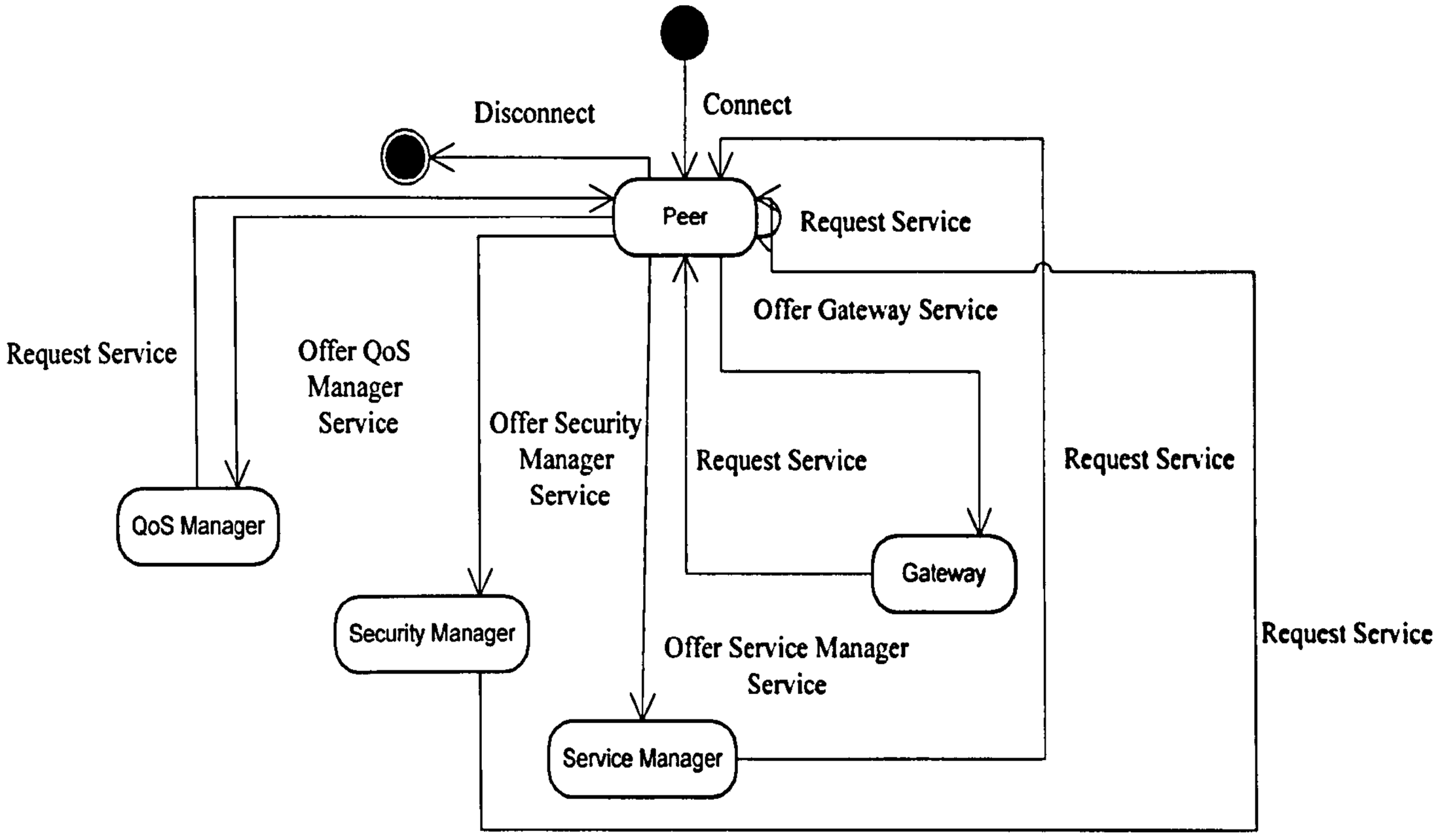

Figure D.3: Peer roles State Transition Diagram

Description:

This State Transition Diagram illustrates the transitions between the various peer roles within this framework.

APPENDIX E: PUBLICATION RESULTING FROM THIS THESIS

A.Muhammad, M. Merabti and B. Askwith, "An Ad Hoc Gateway Service for Discovering and Composing Networked Appliances", Proceedings of the *sixth annual postgraduate symposium on the convergence of telecommunications, networking and broadcasting (PGNet 2005)*, Liverpool John Moores University, Liverpool, UK, pp. 377-382, (27-28 June 2005)

A.Muhammad, M. Merabti, B. Askwith and P. Fergus, "Ad Hoc Gateway Service for Automatic Package Delivery using Networked Appliances", Proceedings of the *IEEE Wireless Communications and Networking Conference (WCNC'07)*, Kowloon, China, pp. 2578-2583, (11-15 March 2007)

A.Muhammad, M. Merabti. and B. Askwith, "An Ad Hoc Gateway Service for Flexible Access to Networked Appliances", Proceedings of the *8th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, Liverpool John Moores University, Liverpool, UK, pp. 141-145, (28-29 June 2007)

A.Muhammad, M. Merabti and B. Askwith, "E-System: Package Delivery Framework", Proceedings of the *2nd International Conferences on Developments in eSystems Engineering (DeSE '09)*, Abu Dhabi, UAE, pp. 196-201, (14-16 December 2009)

Paper accepted A.Muhammad, P. Fergus, M.Merabti and B. Askwith, "Peer-to-Peer Overlay Gateway Services for Home Automation and Management", in *Fourth International Workshop on Telecommunication Networking, Applications and Systems,* Perth, Australia, (20-23 April 2010)