Role-Based Access Control for Wireless Information Systems

by

Bjørnar Brekka and Flemming Kramer

Master Thesis in
Information and Communication Technology

Agder University College

Grimstad, June 2004

Abstract

Role Based Access Control (RBAC) is an effective access control method for protecting information and resources in large information systems. It is used as a basis for security in medium and large organizations, where the network structure often becomes large and complex. RBAC has got increased attention the last couple of years, because of its reduced complexity and costs for security management. When using RBAC, one control the access to data and resources based on the organizational activities and responsibilities users perform in the organization. This means that a user's access to data and resources are limited by the users authorized roles in the organization.

In general access control models have been developed for fixed information systems. When moving on to wireless information systems, which enable employees to access information and resources on an organizations network, through a mobile terminal like a PDA, it is necessary to adapt the RBAC model, so it can handle such flexible and dynamic environments. The model should be extended with location. Then it is possible to handle access requests based on the physical location in which the user is situated. However, considerations must be taken concerning the mobile devices, which suffer in performance, memory, and battery limitations compared to ordinary computers systems.

This thesis gives an overview of RBAC, its functionality and its advantages. We describe an extended RBAC model that is better suited for mobile environments. This model is called Spatial RBAC and is extended to cope with location information.

In addition an architecture and a prototype application has been developed for estimating a users location. This application uses the concepts in RBAC and together with a location sensing technology, it is possible to grant and deny access to resources based on a users location. By using the Bluetooth technology, we were able to define wireless zones and to obtain a user's location. In these zones it is possible to configure which roles, with their assigned permissions, that can be activated in the different zones. When a user moves from one zone to another the system dynamically updates the user's permissions. In cooperation with our supervisors, we were also able to implement our prototype application into a framework for teleservices, called ActorFrame. Through this framework we were able to offer SMS messaging as an available service. GSM localization is also used in addition to Bluetooth for checking a user's actual location.

The main conclusion is that RBAC can be adapted for use in a wireless information system, and that it is possible to use location aware technologies for resolving a users position, and update his or hers permissions accordingly. By leaving the most advanced functionality on a fixed infrastructure and the light part of the system on the mobile devices it is possible to develop advanced access control systems for mobile environments. We have showed that RBAC is an effective access control mechanism for handling access to data and resources in large information systems as well as to reduce the administration tasks. A proposed RBAC standard will also provide an increase in standardised RBAC components in future access control products.

Preface

This thesis concludes the stage of a Master of Science degree in Information and Communication Technology at Agder University College, Faculty of Engineering and Science, Grimstad Norway. The thesis represents 10 credits (one term fulltime).

The project is part of the AVANTEL (Advanced Telecom services) research project, which is related to the PATS program (Program for Advanced Telecom Services). The thesis is given by Agder University College in relation with the establishment of a AVANTEL research lab that will provide research projects in teleservices. The work was carried out from January2004 till May 2004.

We would like to thank our supervisors PhD student Frode Ørbeck Hansen and PhD student Fritjof Boger Engelhardtsen at Agder University College, for great support and for obtaining the necessary equipment needed.

We would also like to thank Geir Melby at Ericsson Norway for great help and support in the development of a Role Based Application Architecture using ActorFrame. We would also like to thank the Director of studies at Agder University College, Stein Bergsmark.

Grimstad, Norway 2004

Bjørnar Brekka *Flemming Kramer*

Table of contents

List of figures

List of tables

Abbreviations

3GPP = the 3rd Generation Partnership Project
AAA = Authentication, Authorization and Accounting
AP = Access Point
API = Application Interface
AVANTEL = A project in Advanced Telecom Services
CLDC = Connected Limited Device Configuration
EJB = Enterprise Java Beans
GPRS = General Packet Radio Service
GSM = Global System for Mobile Communications
HLR = Home Location Register
ICT = Information and Communication Technology
LAN = Local Area Network
MIDP = Mobile Information Device Profile
PDA = Personal Digital Assistant
RBAC = Role Based Access Control
RF = Radio Frequency
RFID = Radio Frequency Identification
SRBAC = Spatial Role Based Access Control
UML = Unified Modelling Language
UMTS = Universal Mobile Telecommunications System
VPN = Virtual Private Network
WAP = Wireless application protocol, enables internet access for mobile devices
WAP Push = WAP Push allows a content server to push content to a WAP enabled handset
Wi-Fi = wireless fidelity, and is meant to be used generically when referring of any type of 802.11 network
UNIX = an operating system developed by Bell Laboratories in the 1970's

1 Introduction

The introduction of computers in the early 1960's led to a major success, and today computers and other advanced technologies, such as mobile devices, has reached nearly every corner of our lives. Companies, organizations and society have become dependent on the digital computer technology.

Information technology has enabled businesses to increase the productivity of their employees, to integrate their supply chains, and to automate and improve their interactions with customers. To achieve this, organizations must attend great importance to information sharing. This means that information should be centralized and distributed among the users of computer networks. This is referred to as, Distributed computing. This has been the main focus in the 21 century.

Traditional computer networks are fixed networks where access is dependent of location. However, in the last couple of years there has been a fast development in wireless information systems, enabling an employee to connect to the company network independent of location. This could be technologies like wireless LANs, Bluetooth connections or telecom services enabled by GSM, GPRS and UMTS. These extended methods for gaining access to the network, the increased use of the Internet, Intranets, mobile terminals, and the increase in information offered on internal and external networks, have made information security an important part of an organization's security policy [1].

Information security enables organizations to protect their data and their resources for unauthorized users, and is essential for maintaining competitive advantage, financial stability and organizational integrity. Still, this is one of the most difficult challenges confronting organizations today [1]. The increase in information offered on internal and external networks, becomes more difficult and expensive. However, security failures can be even more costly by disrupting an organization's operations and can have financial, legal, human safety, personal privacy, and public confidence impacts.

Important aspects for protecting information and resources for unauthorised users are authentication, confidentiality, and integrity. Authentication will verify the machine or the person in the other end of the network connection. If you are sending sensitive data over the network, only authorized people should be able to see that information. This is confidentiality. Moreover, integrity protects your data from being changed or corrupted in any way. Since computer systems often tend to be big and complex, there has been an increased interest in methods that protect the resources on the computer network and makes the IT management easier to maintain.

Various types of access control methods have been developed for the purpose, such as User Based Access Control or Role Based Access Control [13]. The latter is emphasized in this thesis. In Role Based Access Control, access privileges are based on roles. The different roles in an organization are mapped into an organizational hierarchy. Users are then assigned to these roles based on their working position and their need for access

privileges. RBAC has achieved increased attention the last couple of years, because of the reduced complexity and costs for security management. RBAC is implemented in applications ranging from the health sector to military applications.

However, these access control methods are developed for fixed network structures. When moving into the mobile world, new questions arise, that needs to be considered when implementing access control methods into a mobile environment. In contrast to ordinary computers, mobile devices offer a limited set of resources, like performance, memory, and battery limitations.

With the migration of role based access control into mobile devices, new requirements appear. One such requirement is location. The system should be able to handle access decisions based on the spatial dimension in which the user is situated [14]. Since the traditional RBAC model is designed for fixed network structures, new or extended models must be introduced. One such extended model is the Spatial Role Based Access Control model (SRBAC) [14]. SRBAC has extended the RBAC model in order to cope with the spatial requirements needed in mobile environments.

When users are mobile, the access control system must be able to obtain the position of the mobile user in order to handle the access request. There have been much research on such location aware technologies in recent years, but there is not until now that complete systems are released on the market. One such system is the Cricket indoor location system developed by MIT which will be commercially available 2004 [11].

As stated on the Cricket indoor location system homepage "Location Services will bring huge benefits to emergency response, public safety, public transportation, critical infrastructure protection and disaster management. Location services also offer huge potential for commerce involving people on the move, bringing convenience to consumers, helping businesses track assets, reducing miles travelled and helping buyers and sellers in the same vicinity to find each other" [11].

This project is part of the AVANTEL (Advanced Telecom services) research project, which is related to the PATS program (Program for Advanced Telecom Services) [25]. This is a bigger project in mobile research. A new Tele service lab is established at Agder University College for this and enables further research projects, in faculty of Information Technology.

1.1 Services

Telecom services have traditionally been offered by the telecom operators. Examples of popular services are SMS which was introduced in Europe in 1991 [27], and the possibility to download email. Today, with the new mobile technology and the possibilities internet may give, new and attractive services can be developed. The new technology built into mobile phones and handhelds, as well as new network technologies, makes new services possible to realise. Such services will be integrated in a structure where server's offers some service based on a request from clients. With the mobile devices it is possible to offer a set of services based on the physical location which the user is situated.

3GPP has specified a set of APIs called the Open Service Access (OSA) [27]. Through these APIs, 3rd party application providers can access new resources and services on the telecom networks. The different telecom vendors which are part of the ICT (Information and Communication Technology) industry have specified a new network architecture for service implementation [27]. This new network architecture consists of three layers: an access layer, a control layer and a service layer. All of these layers are connected through a backbone network.

The access layer provides different access systems such as mobile networks (GSM, UMTS) and wireless networks. All the different access systems are then connected to the same backbone network. Terminals that make use of one or more of these access technologies must be connected to the access network. The control layer takes care of all call setup and traffic control. The service layer provides the services to the end user. The layer is also responsible for managing user data in databases (HLR) and for billing (AAA). Special servers called application servers contains the applications necessary for providing a specific service [27].

With the introduction of UMTS, it will be important to introduce new services that users are willing to pay for in addition to traditional speech. It is predicted that telecom operators cannot rely solely on speech, but will need to provide additional services for profit. Location aware services may be an important part of this.

To meet these new requirements, Ericsson, Telenor Resarch, and NTNU are cooperating for making new services for the service network [27]. This is done through the AVANTEL and the ARTS projects. Through this cooperation Ericsson in Norway has developed a prototype of an application server, called ServiceFrame [27].

1.2 ServiceFrame – an application server

ServiceFrame [27] is a framework for developing service applications in Java. It was developed by Ericsson NorArc. The framework is implemented in an application server in the service network and enables users to communicate through different types of terminals, like mobile phones and PDAs. ServiceFrame provide the support for service creation, service deployment and service execution.

In Figure 1 [27], we get an overview of the idea behind ServiceFrame. This figure is divided into different layers to facilitate the development of teleservices applications. Thus, Service providers are relieved from the advanced technology implementations on the underlying layers, and they only concentrate on the modelling of the wanted service functionality. The UML 2.0 specification is used for this purpose [48]. When the modelling phase is done, they can deliver the specified model to a software company for implementing the desired functionality into an application. This application is then ready to be used by the end user.

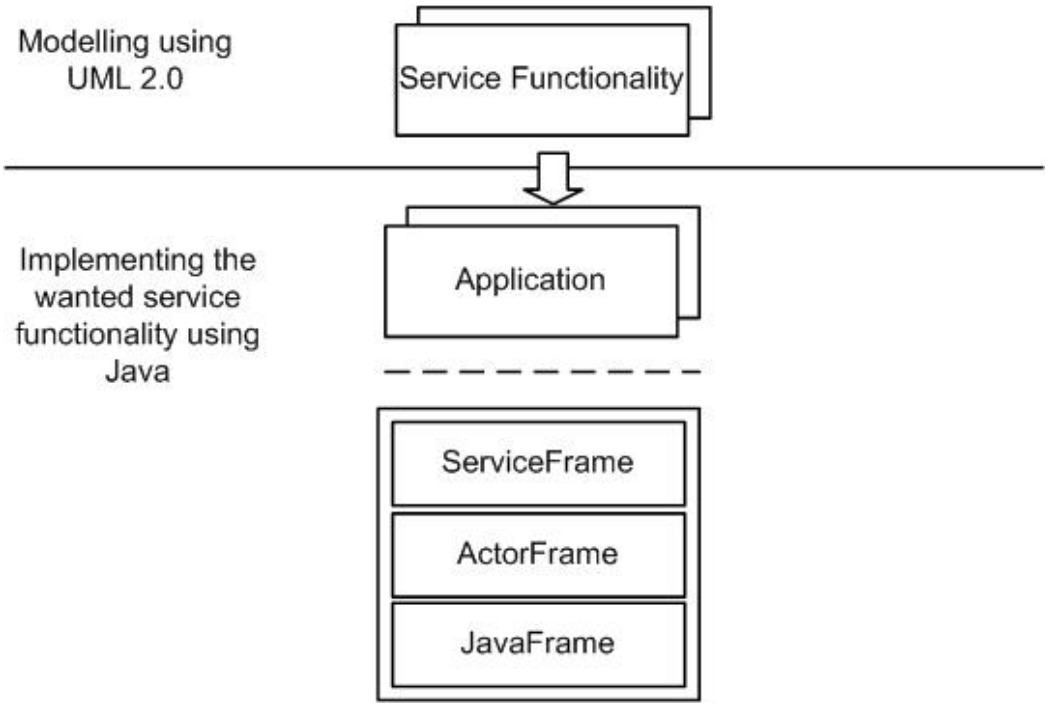

Figure 1 – Implementation of an application into ServiceFrame

An application consists of three layers. ServiceFrame is on top with ActorFrame and JavaFrame at the underlying layers. ActorFrame is a framework that uses the concept of Actors and roles [27]. Actors can play different roles. In ActorFrame an actor is a class that has one in port and one out port. Through these ports the actor can send and receive messages with other actors. One actor has its own behaviour and is described by using a state machine [27]. A state contains the functionality needed for doing specific operations. It then changes states based on the input or the current operation that is requested. JavaFrame [27] contains the necessary library classes used for supporting the implementation of state machines. It is also an execution environment.

The ServiceFrame can be used as an application server, but it offer a limited set of functionality in contrast to commercial application servers [27].

1.3 Thesis Definition

In this subsection we first present our exact thesis definition, before we present the thesis research questions.

"In recent years, Role Based Access Control (RBAC) has been established as a method for handling access to data and resources in large information systems. In such models, one has abstracted the association between users and privileges by the use of role-relations. When using RBAC, one control the access to data and resources based on the organizational activities and responsibilities users perform in the organization. This means that a user's access to data and resources are limited by the users authorized roles in the organization. These roles have associated privileges that grant access to operations on data and resources.

This thesis will convey a study on RBAC and issues relevant to an implementation in a mobile environment.

Current security models and implementations (e.g. ORACLE, CORBA, COM+, .NET, Java, etc.) that exist for RBAC have been developed for fixed network structures and the models may not be suited for mobile environments. Therefore the students will study existing solutions of RBAC and discuss their relevance with regards to an implementation in a mobile environment.

New questions will arise, that will have to be considered when implementing RBAC in a mobile environment. For example the location of a mobile device has to be detected in order to decide the right privileges for the user. A major question will concern the mobile devices. In contrast to ordinary computers they offer a limited set of resources, like performance, memory, and battery limitations. The students must therefore consider these limitations when developing their own implementation.

They will try to find out how Location Based RBAC can be integrated in a system for telecom - services. The students should use one or more of the following technologies for location estimation:

- *Bluetooth Based – For indoor localization.*
- *GSM/Parlay/Parlay-X based: For indoor or outdoor localization.*
- *GPS – For outdoor localization*

The students are going to develop an architecture and a prototype application where the above technology are used to estimate the position of a mobile device together with RBAC to determine a user's privileges to data and resources based on location. The application can show how a mobile device can control embedded devices based on roles and location."

The importance of security in today's computer networks and the growth in wireless information systems, like accessing a company's network through a wireless device and getting access to resources and services as needed, have motivated a study in Role Based Access Control. Because there is a need for higher security, it is important to choose methods that increases security and reduces the tasks for IT management and the administration costs.

Problem 1: Where are RBAC used today, and what sort of implementations exists on the market today?

This will give an overview of some applications and systems that have implemented role based access control. The implementation of features specified in the different Role Based Access Control models varies from product to product and therefore an organization must first consider their needs before implementing a RBAC solution.

Problem 2: What is it that makes RBAC an efficient method for implementing security in information systems?

The interest in RBAC has increased in the recent years, due to its scalability in large organizations and for lightening the administrative tasks. We will therefore investigate RBAC as a method and find out why this is a good solution for achieving better security.

Problem 3: Issues concerning RBAC implementations in a mobile environment?

Since RBAC is designed for fixed networks it may not be suitable for mobile devices in a wireless information system. When moving RBAC principles and mechanisms over to a mobile platform, there are several issues that must be considered. This will affect the fixed RBAC model in a way that suits mobile environments.

Problem 4: How to implement RBAC in an application that is location aware and use telecom services?

We have to investigate technology that is available. First we have to find a technology that can be used for localization of a user indoors. Second, we will find a suited programming language to use. Before the implementation phase starts we must develop an architecture. Then we must find out how it is possible to implement teleservices such as SMS and call forwarding. We must also implement some components of the RBAC model as an access control method.

Problem 5: How is it possible to implement transparent security?

Transparent security should be implemented in a location aware system for example with role based access control mechanisms. This means that the system automatically handles the permissions a user can utilize based on location. For example, a user in one location may not utilize the same set of permissions in another location. This variation of permissions should be invisible for the user, until he or she tries to perform an operation that is denied in the current location.

1.4 The thesis Work

The approach taken in this thesis is to first get an overview of various forms of access control methods and then find out on which technologies to be used for developing a role based wireless localization system. Then we will go deeper into role based access control mechanism and find out how it is possible to transfer the concepts for fixed network structures into a mobile role based localization application.

To verify the problem statements 3, 4 and 5 an architecture for a mobile localization system based on RBAC will be defined and a prototype will be developed for testing purposes. The prototype will be developed in two stages. In the first stage we will start to develop an application that matches the architecture based on J2EE [34] technologies. Then with co-operation from our supervisors and Geir Melby from Ericsson, the prototype will be implemented in a framework for teleservices called ActorFrame. The result of the study in RBAC and the process in developing the prototype will be discussed, summarized and used to make conclusions.

1.5 Reader's guide

The J2EE technology used for developing our prototype consists of several advanced technologies, such as Servlets, EJB and MIDP. In this report we will only explain those concepts that are needed for understanding the prototype developed.

Chapter 2 gives an introduction to different access control methods that are used for network security. These methods are for example used in operating systems, for file access, and in database applications.

In chapter 3 we dive deeper into role based access control and explains the model mostly defined in the NIST standard [13]. At the end we give examples of RBAC support in real life applications.

In chapter 4 we introduce the concept of location aware computing and give some examples of research in the field.

In chapter 5 we introduce the idea of moving role based access control concepts into a mobile environment. Then we will look at a RBAC model that is adapted for use in mobile environments.

In chapter 6 we explain our proposed architecture a wireless system and describes our prototype. We also give an overview of the implementation into ActorFrame.

In chapter 7 the solution is discussed in relation to the problem statements give above, and this discussion is summarised in a conclusion at the end of the paper.

2 Access Control Methods

2.1 Introduction

In according to [2] Access control is about network security. It is used to grant access to users when appropriate and deny access when inappropriate. Almost every network has a form of Access Control. Nevertheless, many networks only use the tools and security settings provided by the operating system. Through different tools, it is possible to manage different access control resources. Having the right tool is essential. A network can make use of many tools to protect different resources. These tools enforce security policy and, or users privileges by protecting mail servers, web applications, database systems, fileservers, applications, or some combination of these resources. Today there exist many non-standardized terms and access control methods.

The most basic method is the Access Control Matrix and optimizations of it, like the Access Control Lists and Capabilities. Then more recent methods like, User Based Access Control, Role Based Access Control, Policy Based Access Control, Context Based Access Control, Mandatory Access Control and Discretionary Access Control are described.

Whatever solution a company chooses, there exist only a few approaches to configure access permissions. Through these approaches, an administrator gives users only the necessary access permissions that the user needs. It is not desirable to grant unnecessary access or to deny necessary access. It is desirable to find the accurate privileges for a user, to determine its least privileges [2]. To achieve this goal it is important use the right access control model and to have the right configuration tools. These tools automate the administration process, since manual configuration prevents the common goals of access control [2].

As a basis we use [2] and [19] for the description of each access control presented below. We will give a brief overview of some solutions that implement access control components, with weight on role based access control. We also give a small overview of the development process, for implementing such a model into real life access control systems.

2.2 Access Control Matrix

The Access Control Matrix is the simplest framework for describing a protection system and it is used as an abstraction mechanism when talking about computer security. The model describes the rights of processes or users over resources, such as files, in a matrix [19].

When using the Access Control Matrix in a system all protected entities are called the set of objects O. The set of subjects S is the set of active objects, such as processes and users. The relationship between the entities is represented by a matrix A. The different relations that describe the object access privileges exist of objects, subjects and rights. The available rights in each entry are specified in a set R. The set of protection states is represented by

the triple (S, O, A). For example, Table 1 [19] shows the protection state in a system. It shows two processes that each own a file. Each process owns itself and the file with the same number as the process. Process 1 can read or write to file 1, and read file 2. Process 2 can append to file 1 and read file 2. Process 1 can communicate with Process 2 by writing to it, and process 2 can read form process 1. The set of rights R in matrix A are {read, write, execute, append, own}. The method lists what processes or users are allowed to do with the resources in a system.

Table 1 – An access control matrix named A.

	File 1	**File 2**	**Process 1**	**Process 2**
Process 1	read, write, own	Read	read, write, execute, own	write
Process 2	append	read, own	read	read, write, execute, own

The processes are treated as both subjects (rows) and objects (columns). This enables a process to be handled by an operation or an operator.

The interpretation of the different rights vary from system to system, but the rights "read", "write", and "append" are usually implemented in the same way. However, these rights can also have different meanings depending on the object involved. Therefore, when talking about the meaning of a particular access control matrix, one must talk with respect to one particular implementation or system.

The objects involved in a matrix are normally thought of as files, devices and processes, but they can also be messages sent between processes and even particular systems or functions. The functions could then determine the set of rights at any particular state based on other data, such as a history of prior accesses, the time of day and so forth.

The method can be used to describe the current protection state in a system. A state is a collection of the current values of all memory locations, like secondary storage and other components. The subset of this collection that concerns security is the protection state of the system. For example [19], if P is the set of possible protection states, and Q a subset of P that is used in particular system, the system is secure when in state Q. Security mechanisms are then used to prevent the system from entering an unsecured state P- Q. Thus, the method can be used to change the protection state of the system.

In theory the access control matrix is ideal for providing a mechanism for controlling access to objects. However, implementing a straightforward implementation in a system, for example an operative system, with a huge collection of objects and subjects, would make the matrix use significant amount of storage. Thus, several optimizations of the model enable systems to use more convenient and simpler versions of the access control matrix [19]. Two of these models are the Access Control List and Capabilities.

2.2.1 Access Control Lists

As mentioned Access Control Lists is a variant of the access control matrix. Most commercial operating systems with security models base their permissions on an Access Control List model of some type. This model stores each column with the object it represents. Each object represented has a set of pairs associated, and each pair contains a subject and a set of rights. The subject can then access the associated object using any of those rights. Access Control Lists bind the data that controls access to a object [19].

Let us use an example from [19]. See Table 1 on the previous page. The set of subjects are process 1 and process 2. The set of objects are file 1, file 2, process 1 and process 2. The access control list for this matrix shows the rights a process has on an object. In this example the access control list would look like this

acl(file 1) = { (process 1, {read, write, own}), (process 2, {append}) }
acl(file 2) = { (process 1, {read }), (process 2, {read, own) }
acl(process 1) = { (process 1, {read, write, execute, own}), (process 2, {read}) }
acl(process 2) = { (process 1, { write }), (process 2, { read, write, execute, own}) }

As we see each object and subject has an associated ACL. Thus, process 1 owns file 1, while process 2 can only append to file 1. Similarly, process 1 and process 2 can read to file 2, but it is process 2 which owns file 2. As with the access control matrix process 1 and process 2 owns itself and the file with the same number. If a subject is not named in the ACL, it has no rights over the associated object.

Some systems abbreviate Access Control Lists [19]. For example in the UNIX [49] operating system, an abbreviate access control list is the basis for file access control. UNIX systems divide the set of users into three categories: the *owner*, the *group owner* of the file, and all *other users*. Each category has its own set of rights. However, the abbreviations of Access Control Lists suffer from a loss of granularity. Since UNIX only uses three set of permissions it would be troublesome if five users would want to access a user's files, because this would need five desired arrangements of rights, while UNIX only supports three [19]. Therefore, the user who owns the file must compromise and give someone more or fewer rights than desired.

There are other disadvantages too. If the model misses a check anywhere in the system, the security system can be broken. If you manage to break into any program with super user permission you get complete control of the system. You cannot give out a precise permission without reworking your own security model. Because of this people tends to give out broad permissions.

2.2.2 Capabilities

Capability lists are like a row of an access control matrix. Each subject has associating pairs, and each pair contains an object and a set of rights. The subject can access the object in the ways defined by the rights for that particular object [19].

Again we use Table 1 for making an example [19] of a capability list. The set of subjects is process 1 and process 2. Then we get a capability list like this:

cap(process 1) = { (file 1, { read, write, own }), (file 2, { read }), (process 1, { read, write, execute, own}), (process 2, { write }) }
cap(process 2) = { (file 1, { append }), (file 2, { read, own }), (process 1, { read }), (process 2, { read, write, execute, own}) }

Each subject has an associated capability list. Thus, process 1 owns file 1 and can read or write to it. Process 1 can read file 2, process 1 owns itself and can read, write and execute. Process 1 can write to process 2. Similarly, process 2 can append to file 1, process 2 can read to file 2 and owns file 2. Process 2 can read to process 1 and process 2 owns itself and can read, write or execute.

Capabilities encapsulate an objects identity [19]. When a user tries to access a file the process that owns that file would present a capability on behalf of the user. The operating system would then examine the capability to determine both the object and the access to which the process is entitled. In contrast to Access Control Lists where the processes are under control by the operating system, Capabilities must be identified by a process in order to use it [19]. The process must have some control over the capabilities. Thus, one should believe that Capabilities are the most security mechanism in use, however it turns out that Access Control Lists that are mostly used. The reason may be because questions about the access rights for subjects on objects are asked more frequently, which Access Control Lists, handles better than Capabilities.

2.3 Mandatory and Discretionary Access Control

Traditional access control models include Mandatory Access Control (MAC) and Discretionary Access Control (DAC). The acronyms also mean other things, but in relation to security they stand for Mandatory and Discretionary Access Control [12].

Mandatory Access Control [2] is obligatory, meaning that an operation should be permitted or denied without letting a user override the policy. In MAC, decisions are made beyond the control of the individual owner of an object [2]. This can increase the level of security, since it is based on a policy that does not allow any operation not specifically authorized.

MAC implements Multi Level Security (MLS) [2]. MLS systems used in military environments, implements an extra security layer for each object by using labels, such as “top secret”, “secret”, “confidential”, and “unclassified”. Users can only access those objects that have the same or a lower level assigned. This works on a “need to know

basis", known as the principal of least privileges [2]. In this way users can only access the objects they need to be able to do their job.

MAC is specially constructed to incorporate with the policy for one-directional information flows [2]. Mail servers for example, implement some kind of MAC. Most servers have defined a max size limit for the email messages. Also many servers reject incoming messages which are suspected for containing computer viruses. Both these methods are MAC, because they cannot be overridden by the end user. The policy is used to control the access to the mailboxes managed by the mail server.

In contrast to MAC, DAC policies leave the final decision to the user [2]. It is the most common type of access control mechanism implemented in computer file systems. In such a system each user can grant or deny access to files for other users. But, as long as the file system is discretionary, so will the adherence to the company policy. Then it will be up to every user to grant or deny access to his or her files in relation to the company policy.

There is also possible to combine MAC and DAC (MDAC), forming a policy which implements both methods [2]. An example could be a mail account owner. The owner can specify rules to incoming mail such as rejecting a mail or delete it when received. Such a system would be a combination of the MAC policy and the DAC policy. The MAC policy will need to override the DAC policy in such systems.

MAC is usually not used because they are almost impossible to implement with today's operating systems. It is most used when strict security is needed, such as in military environments.

In theory DAC can be very finely grained, but actually DAC is extremely labour intensive. Every user must define permissions for all users to every resource he or she owns. All users have to cooperate, else the administrator needs to configure separately for all owners of the resources. Therefore users tend to give users more access privileges than needed.

DAC restricts access to objects based on the identity of the subject which are trying to access them. Any program which runs on behalf of a user inherits the DAC access rights of that user. This basic principle of access control makes it vulnerable to Trojan horses. So the DAC mechanism may provide illusory security to users who are not aware of Trojan horses [45] .

For example consider Bob and Alice who is users in a DAC system. Bob is an honest user while Alice is a dishonest user. Bob has a file which contains highly sensitive data, called *Bobs_File*. Bob has set the ACL to only allow him to read that file, and he is confident in that no other users can get access to the content.

However, Alice wants access to *Bobs_File*. She has legitimate access to the system that enables her to install a utility program. In this program, Alice hides a function that copies the contents of *Bobs_File* into a file called *Alices_File*. *Alices_File* has an ACL associated

with it that enables processes to execute the file on Bob's behalf, while Alice's process can read it.

Alice convinces Bob to execute this program without telling him about the Trojan. Bob executes the corrupted program and it appears to be performing perfectly. However, the program is now operating on Bob's behalf and the system assumes Bob's identity. Now the program copies the contents of *Bobs_File* to *Alices_File*. This copying process is completely within the DAC rules, and thus Bob is unaware of what has happened [45].

2.4 User Based Access Control (UBAC)

In User Based Access Control [2], an administrator must define permissions for each user. Each user may have different individual needs and the security management is therefore labour intensive. It is not possible to know precisely what access permissions each user may need [2]. To make a system effective it is necessary to update each user's permissions daily. UBAC specify permissions to files instead of a group. This makes the job intensive.

For example, two persons each working at different departments, need different access to files and documents. Then security management must assign permissions to those files that they need that day. Next day they may need access to other files and documents, so new permissions must be assigned. This makes the permissions assignment intensive. It would be much easier to assign users to groups, where each group has different permissions, than to individuals.

2.5 Role Based Access Control (RBAC)

Role Based Access Control is solid established as a base for today's security administration needs [3]. RBAC is used as a basis for security in medium and large organizations. In RBAC, security permissions are mapped on different "roles" in an organizational hierarchy. A "role" is a user group with access to a specific group of resources [2]. Each role has different permissions mapped to it, and users are added to the different roles. It is possible for a user to be assigned to one or more roles, according to their access permission needs. It is possible to specify a super-user who has access in every role in an organization, and limit the participation to one or two roles to other users. RBAC has potential for refining user privileges, by assigning specific types of privileges to specific resources. It is possible to make pre-defined roles and assign users to them. A problem can arise when the same role are specified at different departments. Even if it is the same role, say nurse, the role can have different privileges at each department, making the need for adjustments, when users switch to another department. Another aspect is the management of all the administrators who assign privileges. For example there can be over 100 administrators in one large organisation. A convenient way to managing multiple users is by using roles. This approach is referred to as Administrative Role-Based Access Control [3].

2.6 Policy Based Access Control (PBAC)

Policy Based Access Control uses a set of rules, which forms a policy, to determine the user's access rights to different resources within a network. This can be files, directories, web pages and so on. Policies can be used in various technologies. It is mostly used for large networks but there exits proposals for using PBAC with web technologies [4], or for distributed firewall architecture [5]. PBAC can also be used to form policies for other access control models, such as the Role Based Access Control model.

One mechanism to enforce enterprise policy is the use of Access Control Lists (ACLs) as described in chapter 2.2. When using ACLs, there are usually no distinctions between the policy description and the enforcement mechanism. The policy is defined by the set of ACLs associated with the resources in the network. Defining a policy based on this set of ACLs, makes the management of the policies ineffective. It also implies a high risk for errors and it is hardly scalable up to large organizations with a high level of employees and resources [2]. In fact, if an employee just changes his or her role within the company, a new policy must be defined from all the ACLs, to give the employee the right user privileges. This makes PBAC ineffective for enforcing policies, and will add a lot of work to IT management. Also, the policies are defined with a very low granularity, and tend to give the user more privileges than necessary.

On the other side, PBAC makes a strict distinction between the formal statement of the policy and its enforcements. Coupled with a policy description language it is easy to grant and deny access for unauthorised users [2].

PBAC alone, as a security system with its ACLs is not designed with fine granularity or with management simplification in mind. Therefore there are possibilities to use other control tools for implementing good security system.

2.7 Content Dependent Access Control (CDAC)

As the name indicates, this method grants and denies access to users with associated resources, based on the content of the resources. CDAC [2], is primarily used to protect databases containing potentially sensitive data. An example could be at a hospital where a nurse has access to patient records. She could have access to all records containing blood test, unless those tests that are HIV tests. Or she could have access to patients, unless those with cancer. Only certain people, such as doctors, should access such records.

CDAC could be seen as an easy way to manage access. However, it results in a lot of overhead because the need of scanning records based on the constraints in the security policy. This could really slow down the system, and is not desirable. There is also difficult to achieve high levels of granularity, without extremely labour-intensive configuration and management.

2.8 Context Based Access Control (CBAC)

The CBAC model is not suited for files or other applications. This method is for example used in firewalls for protecting traffic. Cisco [6] uses the method to provide an elementary form of packet inspection, in order to protect and preserve the internal security of a network. In CBAC [2], an internal stateful table is used to determine whether or not to permit network access. In CBAC the access to resources does not only depend on who the user is or which resource it is, or even the resource content, but also in the sequence of events that preceded the access attempts [2].

Content and Context Based Access Control are often confused, but these are two different methods to access control. While Content Dependent Access Control grants or denies access based on the content of the resources, CBAC uses strict security policies, contexts, which are the basis for permitting or denying access. This context may be comprised of factors like location, time, temperature, or different states, and controls access based on the relation in which the user makes the request. CBAC does not configure permissions for specific users. For example, in a wireless information system, one could define policies in a context, and dependent of location one could dynamically change the access privileges to resources by reassigning roles.

An example of a CBAC system would be a system which allows the user to access a resource no more than 50 times. The system will then count the number of accesses performed by that user and denies access beyond the first 50 accesses.

Another example would be a quota control system which may be used by fileservers. Such systems can be used by administrators to limit the space a user can upload to the server. A system like this can be used where many users share a common resource, for example in a college, where each student has its own space for saving documents or other related school work. When a student has reached the quota limit, he or she will not be permitted to save any additional data. Then a warning could be sent, stating that there is no more space available. In this way one can prevent students, wanting a lot of hard disk space, from being greedy.

2.9 View Based Access Control (VBAC)

View Based Access Control is usually a method for protecting database systems. [2]. As an example it is used in Simple Network Management Protocol (SNMP) [7]. In contrast to other access models, which grants access to objects like files, documents or printers, VBAC divides a resource into sub-resources [2]. Each user can then be assigned to different sub-resources of that resource. As an example a patient's record can be seen as a resource. The information could then be divided into sub-resources, where users could get access to different sub-resources depending on their role, the need for information in an organization or in which relation the resources are accessed. For example, consider the two roles, nurse and doctor. A nurse may only need access to records concerning drugs and information about blood type, while a doctor needs access to all the sub-resources. Both users access the same record, but different sub-resources are viewable by different

users. It is also possible to update the access privileges based on location or time of the day and so on.

View Based Access Control method is useful when organizational information are stored in databases. Then the access control policy could be defined in a set of predefined interfaces. The only way to interact with the resources would then be through those interfaces. This could be useful when wanting to restrict modification rights on the payroll table for example.

The granularity of VBAC can be very fine, but the configuration is very complex and labour intensive. The fine granularity configuration requires knowledge of data structures and the relationship between users and data.

2.10 Summary

This chapter has given an overview of different access control methods. The Access Control Matrix is the most basic method. However, several optimizations of the model exist, such as the Access Control Lists and Capabilities, for providing more convenient and simpler versions. However the most used access control method, for example in operative systems, is DAC. MAC is only used when strict security is needed such as in the military. A disadvantage with DAC is that it is vulnerable to Trojan horses. There are also methods that are suited for database systems such as VBAC and CDAC. Each access control model has its strengths and weaknesses. However, it is possible to build an access control system by using different layers of security and the using the proper access control method on each layer. Then Role Based Access Control could be used on a higher level.

3 Role Based Access Control

3.1 Introduction

In this section we will go deeper into Role Based Access Control and at the end of this section we will introduce some RBAC implementations that exist in the market today. RBAC models have received much support for their approach to access control. They are known of their many advantages when large-scale management is needed. However, still there does not exist any authoritative standardization of RBAC. To change this, NIST have made a proposal for a RBAC standard [13], and we will use this proposed standard and the paper [15], when explaining RBAC.

3.2 Background

The increase in information and resources offered in an organizations network, have made a demand in access control mechanisms. The access control mechanisms presented earlier has all different methods for controlling access to information and resources. Most of them are also dependent on the discretion of the network administrator. This worked fine for small local area networks, but this becomes cumbersome when the network structure becomes large and complex. A relatively new technology that can handle such a complex structure is called Role Based Access Control (RBAC). The older concept of RBAC exists from UNIX and other operating systems and privilege groupings in database systems. The lack of a standardised model caused the industry to believe that this was the reason for the absence of advanced access control products. The industry had to know that RBAC systems could operate across a wide range of network communication operating systems [13].

Today RBAC is a complete model, but not yet fully standardized. Because of this, there exist different implementations of the RBAC model. There has been up to the company development team to implement a suitable RBAC model and its features in their applications. The different RBAC models are relatively similar on fundamental RBAC concepts. Then one should believe that there were easy to separate them. This is not the case, it is actually difficult to point out similarities and differences between them. This is because many models use different terminology and because products come from different commercial and academic backgrounds.

The RBAC model embodies terms like roles and role hierarchies, role activation, constraints on user and role membership [13]. RBAC controls access to computer networks based on roles. Permissions are then mapped to roles and users are assigned to roles based on their responsibilities and qualifications. This simplifies the management of permissions and roles, since it is easy to assign or reassign a role to a user or change the permissions assigned to roles.

Operating systems like Novell's Netware, Microsoft Windows NT and Solaris had some kind of administrating roles, but they provided little support for application-level use of RBAC [15]. However, today operating systems implement more elements of the RBAC

model. NIST also provides reference implementations for an RBAC web server for both UNIX and Windows NT systems [24].

With RBAC, an organization maps its specific structure, roles and permissions into the Role Based Access Control model. The advantage of using roles is several. Firstly, it simplifies authorization administration because a security administrator only needs to revoke and assign new roles to a user if he changes his or her job function. Furthermore RBAC has shown to be policy neutral and supports security policy objectives as least privilege and static and dynamic separation of duty constraints [13].

In order to prevent misuse of assigned permissions, there is possible to define constraints. A typical authorization constraint is Separation of Duty (SoD). This mechanism reduces the risk for fraud by not allowing a user to have sufficient authority to perform an operation. This can be done easily in RBAC by using the SoD model on roles, user-role assignments, and role-permission assignments [13]. There is also possible to use SoD when a user activates a role. When a user has signed in, with SoD, the role is activated only with the least privileges needed to perform an operation.

3.3 The RBAC Reference Model

To address the issues concerning the different RBAC models and the problems mentioned, NIST [50] have made a proposal for a standardized RBAC Reference model [13]. This model defines a collection of basic RBAC components and features. The reference model, Figure 2, defines a core set of features that must be implemented in all RBAC systems. The model also provides a precise and persistent language. This will address the problem concerning the different terminology used by RBAC models today. Two criteria were set [13], when the proposed RBAC standard was specified. The authors states that the "features must be well understood and well represented in the RBAC literature and established RBAC models. The RBAC features that are included in the reference model should exist in at least one commercial example or reference implementation".

The reference model consists of core RBAC, hierarchical RBAC and constrained RBAC which includes Static Separation of Duty relations and Dynamic Separation of Duty relations. $RBAC_0$ is the base model and represents core RBAC. $RBAC_1$ represents role hierarchies and $RBAC_2$ represents constraints. $RBAC_3$ represents a model that includes both $RBAC_1$ and $RBAC_2$, and implicit RBAC.

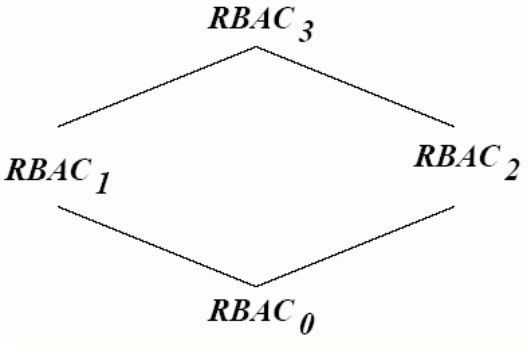

Figure 2 - The relationship between the RBAC models

Since vendors develop for different environments they have different needs for RBAC components. The RBAC reference model provides a package feature, which can be used to choose between those functional components in Figure 2 [15] and features needed for a particular implementation [13]. The standard defines a core set of RBAC functions, $RBAC_0$, that must be included in all RBAC implementations and other sets of optional components that may be implemented. These are $RBAC_1$, role hierarchies and $RBAC_2$, static constraints (static separation of duty) and dynamic constraints (dynamic separation of duty).

The specification specifies a system and administrative functional specification that defines the features required in a RBAC system. There are three categories of features: administrative operations, administrative review functions, and system level functions [13]. The first two are dependent on an administrative interface for performing its functions. The administrative operations are defined for administrative tasks. These are operations for creating, deleting, and maintaining RBAC elements and to create and delete user role assignments. The administrative review features defines query-operations that can be performed on RBAC elements and relations. For example it is possible to return the set of users assigned to a specified role or return the set of permissions assigned to a specific role. The system level functions define features for creating user sessions for enabling role activation or deactivation, enforcement of constraints on role activation, and calculating for an access decision.

The benefits of providing a standardized set of RBAC features are to provide a standard of reference guidelines for vendors developing RBAC products and as an evaluation by their prospective customers. It will also give IT consumers a basis for making purchasing decisions, as well as to provide researchers with new and innovative access control and authorization management models and techniques [13]. The proposed RBAC reference model provides a basis for further standardization in the development of new APIs. We will now describe the RBAC components used in the proposed NIST standard.

3.4 RBAC Components

To describe the different RBAC components the reference model has been divided into two parts, a reference model and a functional specification. To describe the dimensions of RBAC we will use a conceptual model that shows the available feature in RBAC, as showed in Figure 2. The first part will explain the reference model and describe RBAC sets and relations, to provide a common vocabulary for specifications and for giving an overview of the features included in the standard [13]. The second part describes the functional specification that defines administrative operations required.

3.4.1 Core RBAC ($RBAC_0$)

Core RBAC is shown in Figure 3 [13], and defines a minimum set of elements, element sets, and relations that must be implemented for achieving a RBAC system. When looking at Figure 2, $RBAC_0$ is placed at the bottom enabling other components to be implemented

on top. The core elements represent users (USERS), roles (ROLES), objects (OBS), operations (OPS) and permissions (PRMS). Figure 3 also shows a collection of sessions (SESSIONS) where each session is a mapping between a user and an activated subset of roles that are assigned to the user.

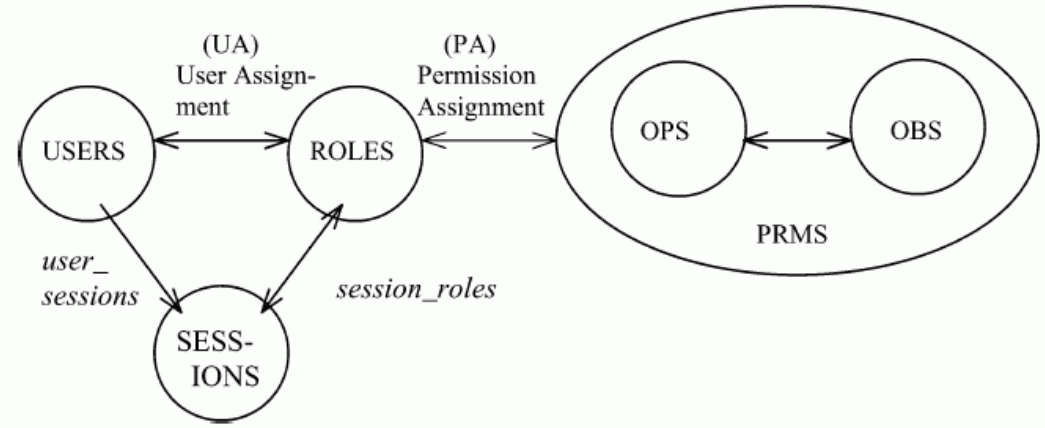

Figure 3 - Elements of core RBAC ($RBAC_0$)

A user is often referred to as a human being, but can also be generalised to include intelligent autonomous agents such as robots, mobile computers, or even network of components. A role is a job function or a job title within an organization that describes the authority and responsibility associated with a particular job function. Permissions consist of operations and objects. Permissions are an approval to perform an operation on objects that is protected through RBAC. Some models includes both positive and negative permissions, but in this model negative permissions is seen as a more advanced feature and is therefore used in form of constraints in $RBAC_2$ [15]. An operation is a specific function that is used on objects, and they are dependent of the system where they are implemented. For example, in operating system environments, operations may be *read*, *write* and *execute*, while in a database system operations may be specific database operations such as *insert*, *update* and *delete*. Objects are entities that contain or receive information and may represent files or directories in a file system or rows, columns, tables and views within a database system. Objects can also represent system resources like printers, disk space and CPU cycles [15]. All objects that are covered by RBAC are listed in permissions that are assigned to roles.

Central to RBAC is the concept of role relations [13]. Figure 3 shows user assignment (UA) and permission assignment (PA). The arrows between users and roles and roles and permissions indicate a many-to-many relationship. This means that a user can be assigned to one or more roles, and a role can be assigned to one or more users. Likewise, permissions can be assigned to one or more roles and roles assigned to one or more permissions. This way of controlling access to resources provides great flexibility and granularity of assignment of permissions to roles and users to roles. In this way a user may be assigned just the permissions that he or she needs for a particular task, which is referred to as the principle of least privilege.

The relation from users and roles to sessions enables a user to be mapped to many roles. A user establishes a session when activating a role that the user is assigned to. Each

session is associated with a single user and each user is associated with one or more sessions. The function *user_sessions* returns the number of sessions associated with a particular user and the *session_roles* return the roles activated within a current session. The permissions that will be available for a current user are the union of permissions from all roles activated in a session. Each session may have a different set of active roles. The model assumes that each user is assigned to at least one role, and a role is associated to at least one permission. There is required that permissions apply to data and resource objects and not to the components of RBAC itself. Permissions to modify users, roles, permissions and UA and PA are called administrative permissions.

Some proposals of RBAC models include duties for core RBAC, in addition to permissions as an attribute of roles. A duty is an obligation for a user to perform one or more tasks that may be essential for an organization. However, this is a seen as an advanced function and is more suited for $RBAC_2$ [15].

3.4.2 Hierarchical RBAC ($RBAC_1$)

The model $RBAC_1$ introduces role hierarchies (RH), as shown in Figure 4. A role-hierarchy is mathematically of partial order. A partial order is a binary relation that is reflexive, transitive and antisymmetric [13]. Role hierarchies are used to represent an organization authority and responsibility. They define senior roles and junior roles, where senior, the most powerful roles, are shown at the top of the graphs in Figure 5, and the less powerful roles are showed towards the bottom. In $RBAC_1$, user membership is inherited top-down, and role permissions are inherited bottom-up. Role hierarchies are a well wanted feature in addition to core RBAC, for improving efficiency and to support organizations structure. Imagine an organization where many users perform the same set of operations. This means that every role must be assigned with those permissions, which would be ineffective and administratively cumbersome. Role hierarchies, such as in Figure 5 [13], allows persons with a senior role to inherit all the access rights of their inferiors. This ensures that occupants of inferior positions inherit any constraints that apply to their superior.

RBAC supports two types of hierarchies, General Hierarchical RBAC and Limited Hierarchical RBAC. The first one has no restrictions on the size of the hierarchy. It supports multiple inheritances of users and permissions among roles. It makes it possible to compose new role from sub-roles, and it provides uniform treatment of user/role assignment relations and role/role inheritance relations [13]. Limited Hierarchical RBAC imposes restrictions on the role hierarchies and they are often limited to simple structures such as trees or inverted trees [13].

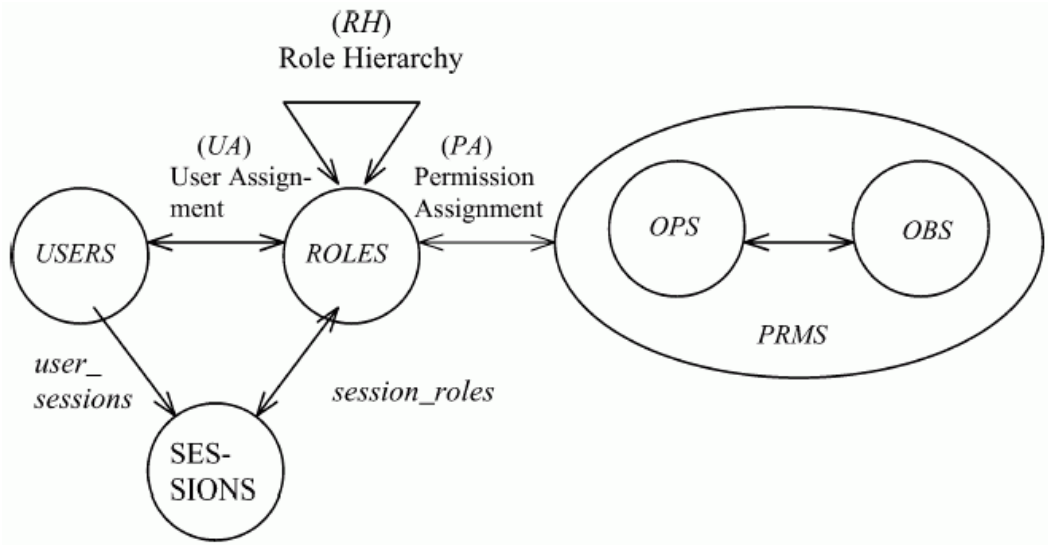

Figure 4 - Hierarchical RBAC (RBAC$_1$)

In Figure 5 (a), an example of an inverted role hierarchy is shown, with the less powerful role as health-care provider. The physician is senior to health-care provider, and inherits all the permissions from that role. The physician role will also inherit any prohibitions that apply to the physician's seniors, namely primary-care physician and the specialist physician. In addition to those permissions inherited by the health-care provider role, the physician role can have own permissions. So the total set of permissions for the physician role would be the specialized assigned permission set, those inherited by the health-care provider, and any constraints from the roles seniors. Since inheritance of permissions is transitive the primary-care physician and the specialist physician inherits permissions from the physician and the health-care-provider roles. In addition to permissions inherited, the two senior roles also have directly permission assigned. This way of inheriting permissions makes it easier for administrators to assign roles. For example, if Bob was hired in as a specialist physician at a hospital, an administrator could just assign Bob to the role specialist physician, and thus Bob is assigned the necessary permissions needed for doing his job. Bob has then implicit inherited permission from the health-care provider and the Physician role as well as the specialized permission in the role specialist physician. The other way of doing it would be to manually assign all the permissions to Bob from the scratch. It would be a really hard-working and time consumption task. UBAC have to be configured in this manner.

Figure 5(b), illustrates multiple inheritance of permissions, where the project supervisor inherits from both the test engineer role and the programmer role. A user can establish a session with his assigned roles, and any combination junior to those the user is a member of. The permissions a user can utilize in a session are those directly assigned to the roles active in a session as well as those assigned to roles junior to these.

In Figure 5(b), the project supervisor inherits all permissions from both the test engineer and programmer as mentioned. Suppose the test engineers want to keep some permission private and prevent the inheritance of those permissions to the project supervisors. This can be done through limited role hierarchies as in Figure 5(c). By defining a new role test engineer′ and relate it to test engineer. By doing this, the test engineer′ and the project

supervisor will inherit permissions from test engineer. New permissions can then be assigned to the test engineer' role. The project supervisor will only inherit permissions from test engineer, thus limiting the scope of inheritance. Roles like test engineer', are called *private roles*. The figure also shows a private role of programmer'. This situation can exist for legitimate reasons, such as preventing the project supervisors from accessing incomplete work in progress. In some systems, the effect of private roles is achieved by denying upward inheritance. In such systems, permissions are not distributed accurately. It is preferred to use private roles. This keeps the meaning of the hierarchical relationship among roles intact [13].

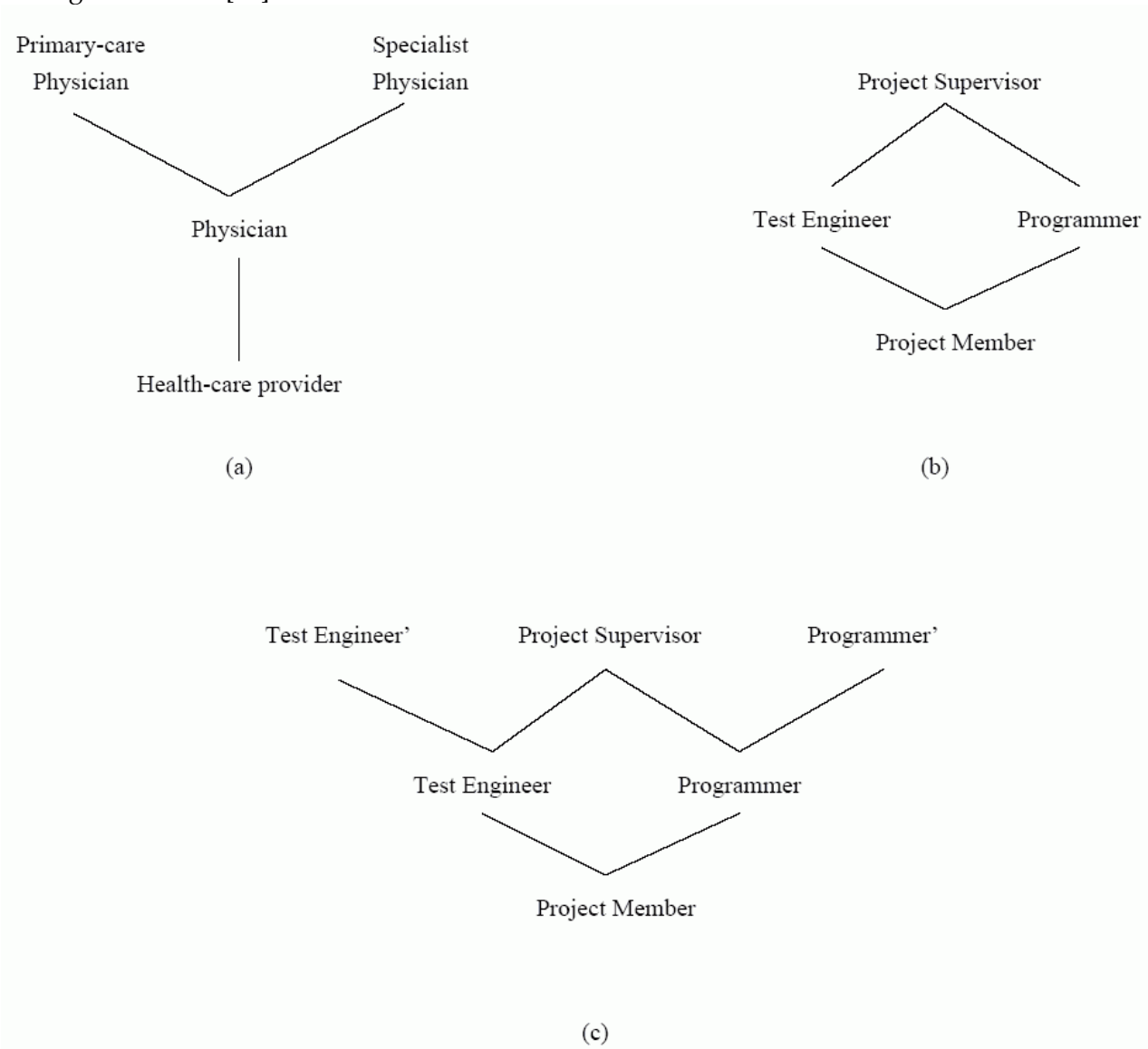

Figure 5 - Example of role hierarchies

3.4.3 Constraints (RBAC$_2$)

The model RBAC$_2$ introduces the concept of constraints, and adds Separation of Suty relations (SoD) to the RBAC model. SoD relations are used to enforce conflict of interest policies [15]. This can be used in organizations that want to prevent users from exceeding their authority of their positions. This means that a user will not be permitted to be a member of roles that have the possibility for committing fraud. Fraud and other major

errors shall not occur without co-operation between multiple users. Constraints are argued to be one important feature for the motivation of implementing RBAC and have long been recognized for its wide application in business, industry and government [15]. The RBAC standard allows for both Static Separation of Duties and Dynamic Separation of Duties.

Static Separation of Duty relations (SSD)

Conflict of interest may arise when a user is gaining access to permissions that extends the authority of his or her position [13]. This can happen when a user is assigned conflicting roles. With SSD this is solved by enforcing constraints on the assignment of users to roles. With SSD the number of permissions available to a user is restricted, by placing constraints on the user assignment of roles. There exists various forms of SoD policies and is discussed in various papers like [16].

Today, RBAC models and policy specifications have grown well beyond simple relations. However, there are not jet known commercial products that implement these advanced static constraint relations [13]. There exists different definitions of SSD relations, and some include assigning SSD for both UA and PA relations. As shown in Figure 6 [13], and according to the NIST standard, SSD relations are only used on the assignment of restrictions on set of roles. This are applied to the UA relations between the set of available roles and the set of users. Then, no user can be simultaneously assigned to both roles specified in the SSD policy if the two roles share a SSD restriction. Imagine an extreme case with a doctor and a patient. If the patient gets sick then the patient can go to a doctor. But, when a doctor gets ill, he can not be assigned the role patient. SSD relations would not be a good solution, DSD solutions should be applied in this case, see the DSD subsection below. So, in real world examples, this definition is overly restrictive because the size of the set of roles in the SSD and the combination of roles in the set for which user assignment is restricted. Therefore the NIST standard defines SSD with two arguments [13]: a role set that includes two or more roles *(rs)*, and a number greater that one *(n)*. The set *(rs, n)* indicates that no user is assigned to more roles indicated in *(n)* from the set of roles in *(rs)* which both are elements of the SSD. SSD relations may exist within hierarchical RBAC, but then precautions must be done to make sure that user inheritance does not undermine SSD policies.

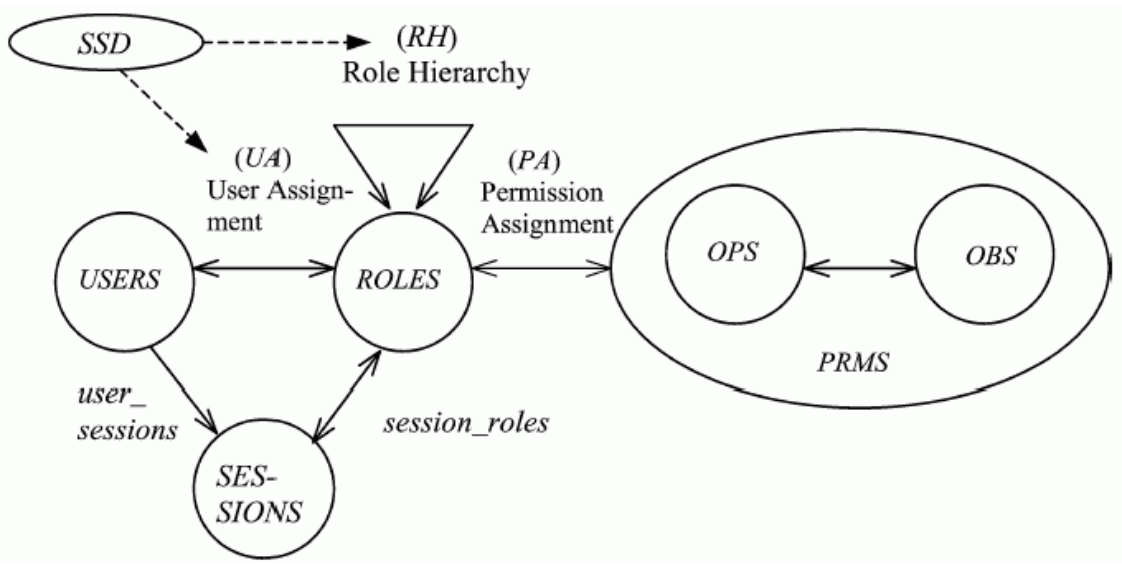

Figure 6 - Static Separation of Duty relations

Dynamic Separation of Duty Relations (DSD)

Dynamic separation of duty relations is also intended to restrict the total number of permissions that is available to a user, just like SSD relations. SSD relations provide the capability to handle conflict-of-interest issues, when users are assigned to roles. DSD relations offer the capability for a user to be authorised for two or more roles that do not create a conflict of interest, when activated independently, but when acted simultaneously in a session policy concerns are initiated. For example, the doctor and the patient example are shown in Figure 7. A user, Bob, who is currently assigned the role doctor, can also be assigned the role patient. With DSD, Bob, can act as a doctor or a patient when those roles are activated independently, but when trying to activate them simultaneously, restrictions are initiated. The DSD relation between user Bob and his possible roles express this. If Bob could activate the roles simultaneously, Bob could prescribe medicine for himself, which would cause conflict of interest. In this example the constraint would require Bob to drop the role doctor before activating the role patient. Bob may prescribe prescriptions as long as he operates as a doctor, but if he gets sick he should not be able to prescribe prescriptions to himself. The role doctor should change to patient. He must now go and see another doctor for prescribing prescriptions.

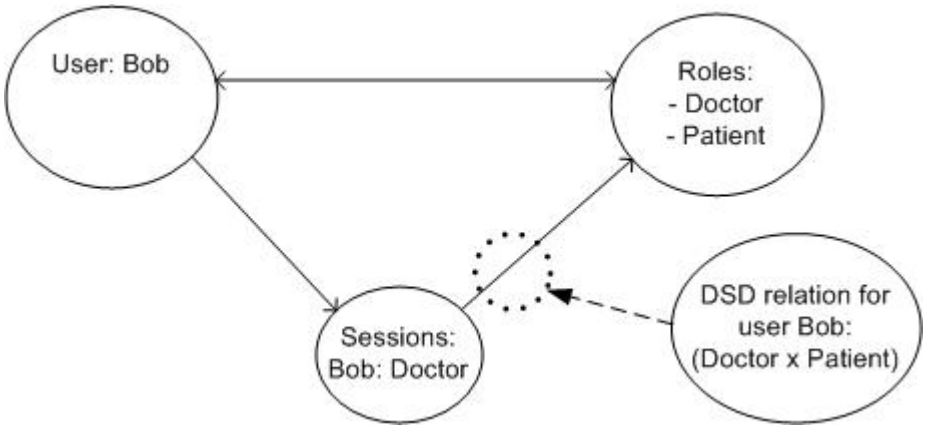

Figure 7 - DSD relation for a user Bob

DSD relations, Figure 8, extend the support for the principle of least privilege, in that different users have different levels of permissions available to them at different times, depending on the role being performed. For example, a programmer may need permissions from the role Test Engineer to do a specific job. With DSD he or she gets access to these permissions via a new session, until the job is done, and then the session is removed. In this way, permissions only exist the time when they are required for performance of duty. DSD relations are able to give organizations with greater operational flexibility.

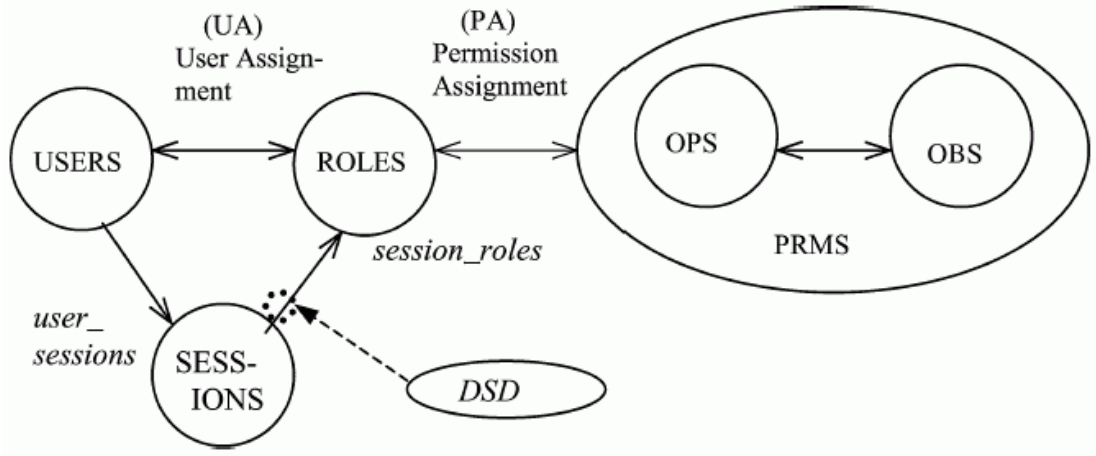

Figure 8 - DSD relations

3.4.4 A Combined model (RBAC$_3$)

RBAC$_3$ consist of both RBAC$_1$ and RBAC$_2$. By implementing both role hierarchies and constraints at the same time, some issues arise. The first issue, partly mentioned under SSD relations, is that constraints may apply to role hierarchies as indicated in Figure 6. Role hierarchies are required to be a partial order, but constraints can limit the number of junior and senior roles that a given role may have. Two or more roles may also be constrained to have no common senior or junior role. These constraints may be useful when the authority to change the role hierarchies has been decentralised, but the chief security officer want to restrict the possibilities in which these changes can be made.

The second issue concerns the assignment of roles. Suppose the programmer role and the test engineer role in Figure 5 (a), are mutually exclusive, meaning that those roles can not be used at the same time or assigned to a user. This is a problem, because the project supervisor inherits permissions from both, independent if they are mutually exclusive or not, and thus violates this restriction. But, in Figure 5 (c), the mutual exclusive restriction may apply to private roles as the programmer' and test engineer role'. This would work, since the project supervisor is the maximal element in the hierarchy, and thus does not inherit anything from those roles. Mutual exclusion can always apply to private roles without raising any conflict [13].

3.4.5 The Functional Specification

This section will give an overview of the functional requirements that are needed for the components described earlier. For implementation, the detailed specification can be found in the NIST standard [13]. The functional requirements embody administrative operations, session management, and administrative review. The RBAC functional specification includes various functions that are required for maintenance of the RBAC model components and supporting system functions. There are three categories of functions [13]:

- Administrative Functions
- Supporting System Functions
- Review Functions

Each of the subchapters explaining the different functions will also give an overview of the core functions that are needed. These functions must be implemented as a minimum.

Administrative Functions

Administrative functions, Figure 9, are needed for creation and maintenance of element sets and relations in the different RBAC models [13]. The basic elements of core RBAC are USERS, ROLES, OPS and OBS. The element sets OPS and OBS are considered to be predefined in the underlying information system, such as a banking system. Creating or deleting users and roles and relations between the roles and operations and objects must be covered by the administrative functions. In addition there must be functions for maintaining relations between users and roles (UA), and between roles and permissions (PA).

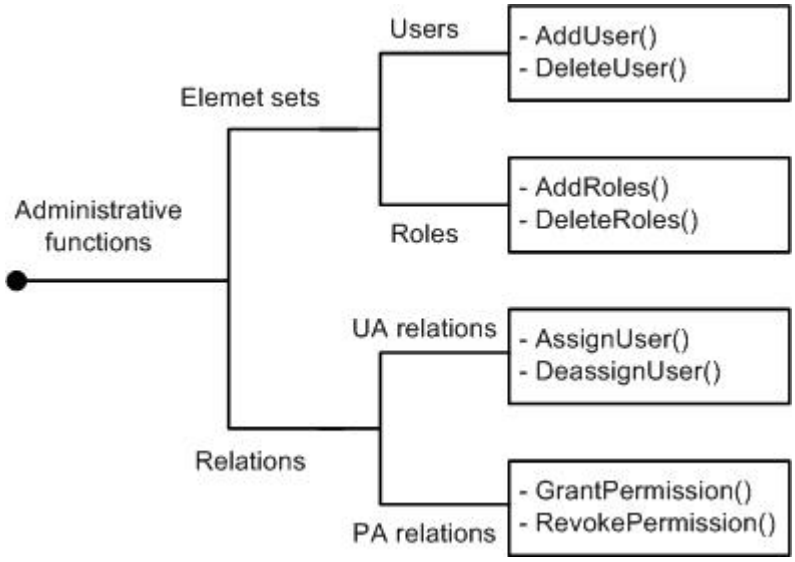

Figure 9 - Administrative Functions for Core RBAC

Supporting System Functions

System functions, Figure 10, are required for session management and for making access control decisions. A user has a set of roles that can be activated and deactivated during a session. A session is initiated for a current user when he or she activates a set of roles from the total space of roles available. When trying to perform an operation, further access control must be done to check if the session object has the requested permissions.

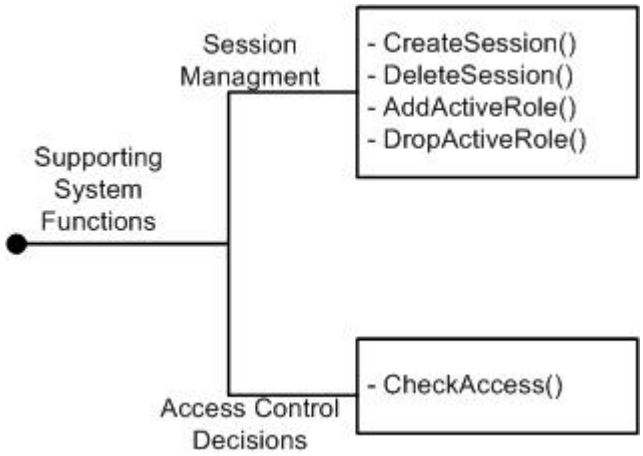

Figure 10 - Supporting System Functions for Core RBAC

Review Functions

With the review functions, Figure 11, which apply to the UA and PA relations, it is possible for an administrator to view the contents of the UA and PA relations, from both the user perspective and the role perspective. For example, an administrator should have the possibility to view the number of users assigned to a given role as well as to view all the roles assigned to a given user. There should also be possible to view the number of active roles for a user in a particular session, *SessionRoles()*, and the total permissions for a given session, *SessionPermissions()* as well as other functions. These functions have descriptive function names so it should be relative easy to understand their functionality.

Not all RBAC implementations have support for all of these functions, and therefore marked O or M [13], indicating if a function is optional or mandatory.

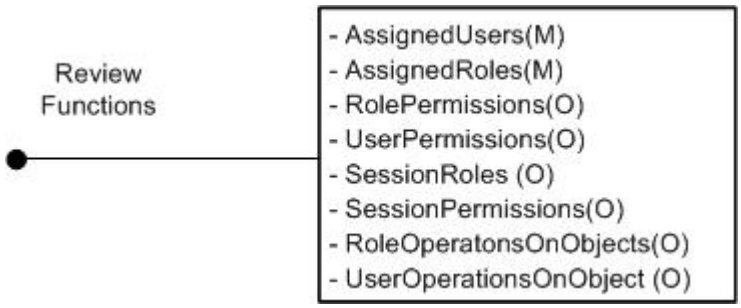

Figure 11 - Review Functions for Core RBAC

3.4.6 The Package Utility

As mentioned earlier the NIST standard provides a package utility. This enables developers to include just those elements that are needed for a particular access control system, for example in an organization. However, the core RBAC component is mandatory and must be implemented in all implementations. Then one can choose to include hierarchical RBAC, SSD relations or DSD relations. Figure 12, gives an overview of the different RBAC elements that can be included. The optional packages are marked with dashed lines. For example, a package can implement core RBAC, limited hierarchies and hierarchical SSD relations. The package will, with the core RBAC, offer the minimum set of elements, element sets, and relations that must be implemented for achieving a RBAC system. Adding role hierarchies, the implementation can represent an organizations authority and responsibility. And finally by adding SSD relations, conflict of interest can be avoided.

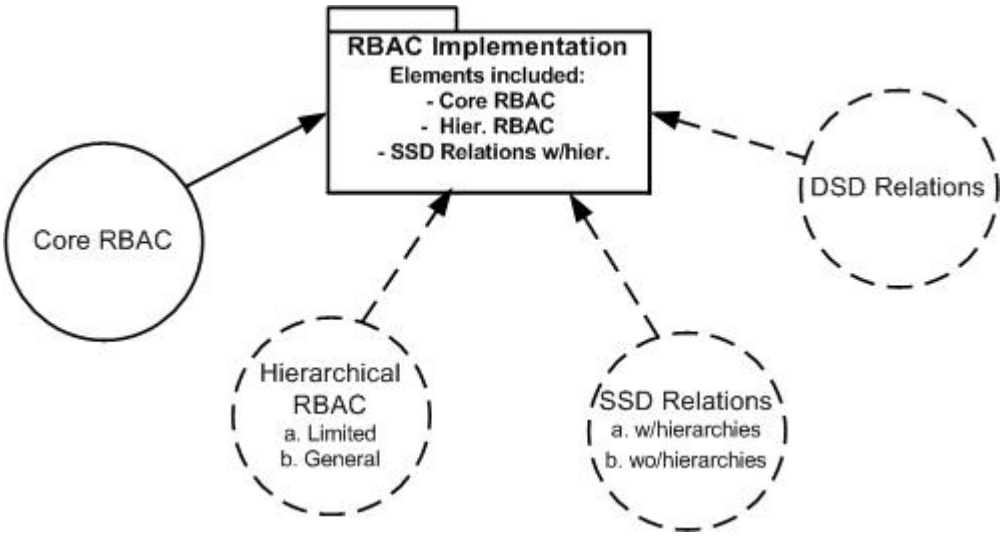

Figure 12 – A possible RBAC implementation

3.5 Some RBAC implementations

There exist many different access control tools. From those which are simple and free to those which are complex and with a high price. The industry spends over billions of dollars every year for products and services. In according to [21] this was about 16.3 billion dollars in 2001 through 2002. In the recent year many buyers decided to reallocate their spending on security products and services with larger technology and service providers, thus sending the signal that security has never been more important for protecting data and information. Some of the largest suppliers of security systems and services in 2002 [21] was Computer Associates, IBM, Network Associates, Nokia, Siemens, Sun Microsystems, Symantec and VeriSign.

It is few solutions that implement all the RBAC components described in the NIST standard [13], thus the development companies adapts the futures in RBAC and implements it into their own specific products. Examples of implementations would be in operating systems, DBMS (database management systems) and other system specific implementations.

As mentioned in the introduction to this chapter, we will give a brief overview of some systems where RBAC are implemented. We will first investigate three database management systems, Informix, Sybase and Oracle [23]. We will then look at a product called the Cisco administration server user interface [22].

3.5.1 RBAC in Database Management Systems

Some of the most used access control methods are MAC and DAC, but RBAC are being supported in more applications and systems. DBMS have taken the lead in implementing role based access control [23]. DBMS provide access control at several levels of granularity including provision for content-dependent controls. DBMS can contain a large amount of data with highly differentiated access permissions for different users depending upon their function or role within the organization. Thus, DBMS needs mechanisms for management of authorizations or privileges. Some DBMS systems [23] that make use of RBAC aspects are

- INFORMIX Online Dynamic Server Version 7.2,
- Sybase Adaptive Server release 11.5
- Oracle Enterprise Server Version 8.0 and Oracle Enterprise Manager 10g release 1

The RBAC features that are supported in these products are categorized based on three RBAC features, the support for user role assignment, support for role relationships and constraints and assignable privileges.

Informix Online Dynamic Server

User role assignment

Informix Online Dynamic Server supports user role assignment [23]. A role can be granted to a single user, a role, a list of users or all users. Informix use the GRANT and DROP commands for assigning users to roles. A user may be assigned more roles, but can

only activate one role during a period in time. It only supports the roles NULL and NONE. These roles are assigned when a user signs on to a database. The user can activate an authorised role by using a SET ROLE command. However, a user can only be active in only one authorised role at a time.

Support for role relations and constraints

A user granted a role may grant this role to another role, thus supporting the ability to build role hierarchies. It has no support for mutually exclusive roles which prevent conflict of interest situations, nor does it support static separation of duties [23]. However, it does support dynamic separation of duties. However, this is a side effect of only allowing one role to be activated at a time.

Assignable privileges

The set of all privileges are assigned into three categories: database level privileges, table-level privileges and execute privileges. The database level privileges refer to those privileges which allow a user to connect to the database, add new objects, like tables, and perform administrative functions including security management. The table-level privileges allow a user to do operations on tables, like the INSERT, UPDATE, and DELETE privileges. The execute privileges are only relevant to database stored procedures, and allows a user to execute a procedure. In Informix only the table-level and execute privileges can be granted to roles. The database-level privileges cannot be granted to roles.

Sybase Adaptive Server

User role assignment

Sybase [23] comes with a set of predefined roles called system roles. There are three system roles. The first role type, sa-role (System Administrator), are meant for a system administrator for managing and maintaining all databases in Sybase as well as controlling physical resources of the server. The second role type, sso-role (System-Security Officer) , are used for performing all security-related tasks, such as creating roles, assigning them to users, groups, or other roles. The third and last role type, oper-role (Operator), has the responsibility for backup operations.

In Sybase roles can be granted to users and users can be assigned many roles. Only a user with the sso-role has the privilege for granting roles to users. Users granted user-defined roles can not assign these roles to other users. This user constraint, and the fact that there will only be allowed for a few system roles, the control of role assignments are under stronger control.

Sybase supports sessions, and users granted a set of roles can activate not only one, but multiple roles from the assigned role set. However, this activation process is only needed for user-defined roles. System roles are automatically activated, but only if they do not have passwords associated with it. It is also possible for a user to delete a role from the default active role set.

Another nice feature in Sybase, is the possibility to define a predefined list of roles to be activated when a user logs on [23]. However, this automatically activation will only work on those roles that do not have passwords associated with them. These roles must be activated one at a time.

Support for role relations and constraints

Just as with the Informix system, Sybase supports granting roles to other roles [23], thus supporting role hierarchies. In contrast to Informix, Sybase support the ability to define mutually exclusive roles, both static and dynamically. With a static constraint on roles, two roles can not be granted both roles. With dynamic constraint on roles, a user can not activate two roles at the same time, thus supporting static and dynamic separation of duties policies as explained earlier. However, Sybase does not limit the number of role assignments a user can perform, but it can limit the number of active roles that can be active at a time, and the total number of roles that can be defined for the system.

Assignable privileges

Sybase operates with two different privilege groups, object access permissions and object creation permissions [23]. The object creation permissions are used for regulating commands for accessing database objects. Some of these commands are SELECT, UPDATE, INSERT and DELETE. The object creation permissions are used for regulating the commands that creates objects, as databases, tables, views, rules and stored procedures. Both permission types can be assigned to roles. However, the object creation permissions cannot be propagated to other roles or users. As mentioned, the assignment of users to roles or roles to roles is only allowed to the System Officer. However, users are allowed to assign privileges of objects they own to other roles.

Oracle Enterprise Server Version

User role assignment

Like Informix and Sybase, Oracle [23] supports the many-to many user and role relationships, thus supporting role hierarchies. Oracle supports two different commands when assigning users to roles, a PUBLIC role and an ADMIN role. By using PUBLIC it is possible to assign a role to multiple users, however users can not grant that role to other users. In contrast, the ADMIN command allows for granting that role to other users or roles, as well as using alter and drop.

Oracle supports sessions and users assigned a set of roles must use a SET ROLE command for enabling or disabling roles for the current user session [23]. In contrast to Informix and Sybase, Oracle supports activating more that one role. Then it is possible to set up a list of roles that will be activated at the time of user login. However, this will only work on roles that do not have a password associated with it.

Oracle also supports additional parameters for the SET ROLE command [23]. These are ALL , EXPECT and NONE. By using ALL, a user can activate all the assigned roles in the roleset. Using EXCEPT it is possible to activate all roles, but exclude certain roles. By

using the NONE statement a user can deactivate or disable all roles for the current session.

Support for role relations and constraints

As with Informix, Oracle also support assigning roles to roles [23], thus it is possible to create a role hierarchy. However, it is not possible to define relations or constraints between roles. Oracle, as with Informix, does not support Dynamic Separation of Duties and nor does it support creating basic user rules for membership in roles.

Assignable privileges

Oracle supports system privileges and object privileges [23]. The system privileges allow for executing commands like, CREATE TABLE and CREATE SESSION. Object privileges are used to perform operations on tables, views and stored procedures. Example of commands are SELECT, INSERT and DELETE.

Both categories of privileges can be granted to roles. However, system privileges can only be granted by the database administrator. Object privileges can be granted by the owner of the object or by a user who has been granted that privilege.

Summary

Table 2 [23], shows an overview of the features in the three different databases. Entry 1, 2 and 3 belongs to user role assignment. Entry 4, 5, 6 and 7 belongs to role relations and constraints, and 8 and 9 belongs to assignable privileges.

Table 2 - overview of the features of Informix, Sybase and Oracle

Item	Feature	Informix	Sybase	Oracle
1	Ability for a role with the grant privilege to grant that role to other users	Yes	No	Yes
2	Multiple active roles for a user session	No	Yes	Yes
3	Specify a default active role set for a user session	No	Yes	Yes
4	Build a role hierarchy	Yes	Yes	Yes
5	Supporting Static Separation on Duties (SSD)	No	Yes	No
6	Supporting Dynamic Separation of Duties (DSD)	(Yes)	Yes	No
7	Specify maximum or minimum rules for role memberships	No	No	No
8	Grant DBMS System Privileges to a Role	No	Yes	Yes
9	Grant DBMS Object Privileges to a Role	Yes	Yes	Yes

The features of user role assignment are supported by Informix and Oracle. However, Sybase does not allow users to assign roles to other users. In Sybase, it is only the system security officer that can assign users to roles, thus Sybase contains tighter control over user-role assignment [23]. Informix does not support enabling multiple roles in a session. This is also the reason why it does not support the activation of role sets for user sessions.

This may be cumbersome since a user has to deactivate and activate roles based on the privileges needed for a particular job function.

All database solutions support role relations, thus the possibility to build a role hierarchy is present. RBAC provides Separation of Duty relations (SoD), which includes both Static Separation of Duty (SSD) and Dynamic Separation of Duty (DSD). As we see of Table 2, Sybase is the only database that supports SSD and DSD relations. However, Informix does support DSD, but this is a side effect of the restriction that only one role can be activated at a time [23].

In assignable privileges, Informix is the only database that does not support granting DBMS System privileges to roles. However, like Sybase and Oracle, Informix does support granting DBMS object privileges to roles. As we see both Sybase and Oracle have support for more RBAC elements than Informix, when speaking about user role assignment and the assignable privileges.

3.5.2 Using RBAC in the World Wide Web

Being a part of the World Wide Web is an important strategic aspect of marketing and sales. A company with a well designed web site can have positive effect of their profit. A key aspect, especially for large organizations, is maintaining the security in multimedia environments such as the World Wide Web. Role Based Access Control suits perfectly for the Internet and for Intranets. It provides an effective and secure way to manage access to an organization's web information. NIST have made a proposal for a possible RBAC implementation that is suited for such environments. This is called RBAC/Web, which is Role Based Access Control adapted for use on the World Wide Web [24]. This is a reference implementation and can be downloaded for both UNIX and Windows NT.

3.5.3 A System specific implementation

A product has been made by Cisco, which have developed an administration server user interface [26] for accessing the Cisco Security Policy Engine (SPA) using the Cisco Broadband Access Center (BAC) application. The Cisco SPE security service provides an authentication and authorization framework based upon the Role Based Access Control (RBAC) model [14], in which user access permissions are associated with roles and users are made occupants of a role. The application can either be accessed through the Cisco Broadband Access Center (BAC) application or via a web browser.

3.6 Access control development phase

When access control tools are to be developed, the development process is carried out in different phases based on three important aspects. These aspects are a security policy, which defines the access control rules, a security model, which provides a formal representation of the access control security policy, and security mechanisms, which define the low level (software and hardware) functions that are needed for implementing the security policy [20].

These three aspects correspond to three different abstraction levels and thus implementation phases. The separation between policies and mechanism introduces

independence between the protection requirements needed in a system, and the mechanisms that enforce these requirements. This separation makes it possible to *i)* discuss protection requirements independently of their implementation, *ii)* compare different access control policies as well as different mechanisms that enforce the same policy, and *iii)* design mechanisms able to enforce multiple policies [20]. The latter is very important because it is not tied to a specific policy, and because of this a change in the policy does not require changes in the access control system.

The formalization between the policy definition and its implementation as a mechanism makes it possible to define a formal model that represents the policy and its formal working [20]. Then, there is possible to define and prove security properties. Therefore, arguing that the model is secure and that the mechanisms have correctly implemented the model, we can argue that the system is secure [20].

The implementation of a correct security mechanism may be complicated. One has to consider the security weaknesses due to the implementation itself and the difficulty of mapping the access control model of real world example into a computer system. In according to [20], the major difficulty lies in the interpretation of, often complex and sometimes ambiguous, real world security policies and the translation of rules enforceable to a computer system.

3.7 Summary

This section has described Role Based Access Control. The model described here is based on the proposed NIST standard. The standard will help organizations and developers to use a common set of terms and definitions. The RBAC model also provides a package utility that enables organizations to choose the right capabilities for their security model. RBAC models are known of their many advantages, especially for handling complex systems and when large-scale management is needed. RBAC are policy neutral and it supports least privileges. This section has also given an overview of some RBAC implementations that exists on the market today, such as the DBMS systems.

RBAC is designed for fixed networks and may not fit in a wireless information system. Because of the enormous growth in wireless technologies and the need for better security RBAC must be adapted into a model that suits a mobile environment and that handles requests based on location. For enabling RBAC to be location aware, it must use some sort of location technologies. In the following chapter we explain what location awareness is and what sort of technologies that exist. We also give an overview of which location technologies that are likely to survive in the future and point to some systems that can be developed for everyday scenarios.

4 Location Aware Computing

4.1 Introduction

Context-aware computing is the concept of sensing and reacting to dynamic environments and activities for example based on a users location. This requires automatic tailoring of information and services based on the current context of the user [10]. The context of the user typically consists of a set of user-specific parameters including his or her location, time, the characteristics of the access device and interface, and the user's requests. The technology needed to realize context-aware computing exists today, and can be found in the increased capabilities of mobile devices and the increasing in wireless connectivity. However, the deployment of such services has been hindered because until now each manufacture has implemented its own systems into the mobile devices, thus an application has to be specially suited for that system. Nowadays there is focus on system wide integrations, such as support for Java in mobile devices and components that that scales with large user populations [9].

Much research has been done and this research has focused on location-sensing technologies and location-aware application support. There exist examples such as conference assistants, support systems for the elderly, tour guides, mobile desktop control and more [9]. For further research in location-aware computing researchers came together at the 2003 workshop on location-aware computing, held as part of UbiComp 2003 in Seattle [10].

Such location aware technologies may be combined with security models such as role based access control, for granting or denying access to services and resources in a wireless location-aware information system.

This section firstly aims to describe location sensing computing techniques and technologies that can be used for making a system location aware. Then we will give an overview of the possibilities for implementing role based access control for determining access to resources based on a user's physical location.

4.2 Location sensing technologies

For making a system location aware, several technologies are possible. Applications that are aimed for outdoor areas typically make use of the popular Global Positioning System (GPS) [33]. A GPS receiver estimates position by measuring data that are received from satellites. Since GPS is based on information from satellites, it offers almost worldwide coverage. However, its performance degrades indoors and in high-rise urban areas. It is only recently, year 2000, that the technology has been accuracy enough to integrate it into small devices like mobile phones and PDAs. This is because USA decided to turn of Selective Availability (SA), a system that only enabled the military to gain the needed accuracy. With SA turned off it now offers accuracies within 5 to 10 meters, compared to 100 meters or so when SA was turned on [33].

GPS offers new services, for example showing the nearest hotel or restaurant when on a business trip. GPS trackers can also be standalone components, without a visual screen or keyboard that make use of another technology, such as Bluetooth, for sending the information to a terminal such as a mobile phone placed in a car. An application installed on the mobile device can then use the information to show the position on a map [8]. The disadvantage of a GPS module is its relative high costs compared with mobile elements such as Bluetooth, and the price of the mobile devices will therefore increase. GPS is therefore not suited for low cost mobile devices.

Indoor location sensing systems can use infrared or radio signals. An example would be the infrared Badge System [8]. This system uses wall mounted infrared sensors that pick up an infrared ID received by the tags worn by the users of the system. In systems using radio, it is possible to locate Wi-Fi (wireless fidelity) enabled devices with accuracies from several meters to tens of meters by using base station visibility and signal strength. Bluetooth can give a more accurate position than Wi-Fi, but it requires more fixed base stations to provide the coverage demand. It is also possible to use RFID [37] identification tags for location determination as well, by placing RFID readers at doorways or walls for detecting the passage of people or objects. This RFID technology is becoming more and more popular and the market is really beginning to understand the utility value. This has made large organizations like Microsoft and Ericsson to establish research projects for developing RFID solutions for the mass market. RFID solutions will be offered for a variety of industries, including manufacturing, healthcare, hospitality, retail, security, and warehousing [8].

Other RF (radio frequency) infrastructures can be used as well, such as those in mobile devices or TV broadcasts. These can be deployed in a wide area with relative ease, in contrast to technologies like RFID that have limited transmission range [8].

The Cambridge Position Systems [8], with the use of mobile phones, and Rosum [8], with the use of TV signals, are projects that have demonstrated location accuracies from 20 meters with mobile phones and from 3 to 25 meters with digital TV signals.

Many of the systems above are based on technologies that are not developed with location sensing in mind. However, there exists at least three types of technologies that are specially designed to provide fine-grained location-sensing, and they are able to achieve accuracies in the order of centimetres [8].

The first system are based on ultrasound, which can be used to estimate known points in the environment, by using mobile tags, and then by using a process like triangulation to derive the location estimate of the tags. The Cricket indoor location system developed by MIT [11] uses ceiling or wall mounted devices for periodically sending out signals, “active beacons”, for detecting devices. Passive listeners, connects to the host devices (handheld, laptop etc.), and estimates the location [8]. The system will be commercially available in early 2004.

The second is vision based systems which often comprises a camera for acquisition and digitalization of images. Those images are then processed and decides which actions to be triggered. A system like this does not require users to carry any sort of tag for detection. However, such systems have difficulties when identifying many objects at the same time. Vision-based systems using some sort of tags tend to be more robust.

Another system, the Cordis RadioEye [40], from Radionor Communications is designed for use in wireless networks, and defines different zones. It is then possible to configure individually conditions for each user, based on their location.

4.3 Future Deployment of location aware technologies

Figure 13 [8], shows the current and predicted deployment of location-sensing technologies within the next two or three years. The width of each box shows the range of accuracies the technology covers. The bottom boundary represents current deployment, while the top boundary shows predicted deployment over the next several years [8]. As we can see, the widest existing deployments are based on GPS, which are specially suited for outdoor localization.

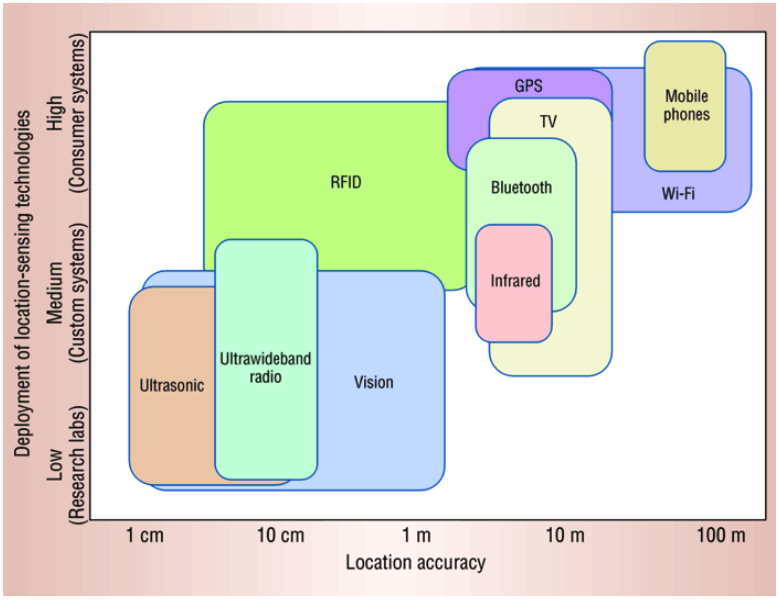

Figure 13 - Location sensing technologies.

Examples of GPS applications can be service applications suited for vehicles such as route planning and fleet tracking, as well as applications integrated in handheld GPS units.

Other examples could be specific applications aimed for specific purposes, implemented in an appropriate location-sensing hardware with a custom software platform. Military training, human-body motion capture, asset tracking and supply chain management are examples of such applications [8]. Future applications may also be included into mobile device such as "dating systems", for example through the use of Java.

Numerous factors are affecting the growth within location sensing technologies. The recent explosion of Wi-Fi, Bluetooth, and other wireless networking technologies has

pushed the implementation of RF technologies into mobile devices. These technologies can then be used for location sensing. Examples of use could be emergency services. By implementing GPS into cars and mobile devices they will be easier to locate and to offer the appropriate service. Another example deals with RFID tracking. The US retail giant Wal-Mart, requires that their top 100 suppliers include RFID tags in their products by 2005 [8].

In addition to deployment of sensing technologies, increased hardware and software support are available for location-aware applications. For example mobile phones have mutated into small computers capable of downloading and running location-aware applications. Microsoft also recently revealed that it's next PC operating system, code-name Longhorn, will include location aware software components.

4.4 Abstracting Location

To be able to support a variety of sensing technologies, researchers are working on techniques that are able to combine data from multiple sensors called fusion, methods for representing location data and on drawing high level contextual information from location data [8].

4.4.1 Fusion – The combination of location data

Today, location-aware systems typically use one type of sensors for a particular application. But, in the near future, applications may use any type of sensors available to a system, combine the data and represent them for the user. The task of making sense of these data for the user is known as sensor fusion [8]. This is the major challenge for future applications, and researchers tries to find solutions by borrowing techniques from the field of robotics. They have developed methods for handling sensor uncertainty as well as method for handling speed and travel paths [8]. The combination of these methods and the use of multiple sensors form a location system that uses data from these sensors for measuring the location. Thus, more accuracy can be provided. The University of Washington has demonstrated such a system [8]. The system gathers location data from multiple sources, including infrared and ultrasonic sensors, and uses a filter for calculating the destination. In addition the system learns typical walking paths, for aiding in location estimation.

4.4.2 Representation of location data

A problem when dealing with location data is to find a way to represent the data received by various sensors. Such representations must consider storage space, communication and interpretation, especially when dealing with mobile devices. Sensors tends to report location as a representation of (x, y, z), but hierarchical representation such as (building, floor, room) or (country, state, city) is more effective for application-level reasoning about locations [8]. It is also up to the application itself to translate the location data into more logical data.

Some types of applications need both types of representations. For example applications such as travel planners use map-like coordinates for example received by via GPS, to measure the position on a digital map. Other information about terrain, traffic pattern,

custom procedures and other factors are important for achieving the accurate travel time. Because of the various types of measurements from different sensors and because of the different applications there have not yet emerged a dominant standard [8].

Place and context

Saying that a person is at a specified location, "at home, "in my office", "at a meeting" is often sufficient for location-aware applications to carry out predetermined actions in a given situation, such as turning off a cell phone during a film or concert. In this case the person's interaction or relationship with the place is more important than the actual physical location. However, relying on an undetermined location can be a leak in security [8]. Imagine a student getting access to an employee's permissions and resources, just telling the system he is at a specific location without actually being there. Therefore systems must be able to determine the exact physical location. For example, University of Washington has tested a GPS system to infer a person's mode of transportation by foot, bus or car. Also MIT have developed a location-aware reality museum that classifies the movement pattern of visitors, as "greedy", "busy" or "selective" and adapts the contents accordingly [8].

Some applications types

In addition to those applications mentioned here, such as military and asset tracking, location-aware applications have been developed for a number of everyday scenarios [8]:

- Office application such as nearest-printer services and mobile desktop control can increase workplace productivity.
- Tour and museum guides can help people navigate an unfamiliar space.
- "Locate my friends" or "dating" utilities can be linked with instant messaging for social or business purposes. This allows mobile human users to interact with each other and with the localised environment.
- Conference aids can track presentation attendance and facilitate note taking and discussion.
- Medical facilities can track staff and monitor patients for emergency response.
- Home applications can help with household management and home entertainment, as well as aid the aged and disabled in performing everyday tasks.

Some of the applications will need coarse-grained location-systems while others will need fine-grained location-systems. The deployment of these applications will vary, depending on the return of investment [8]. However, the fine-grained systems do not have a "killer app", a service that is popular, for the economic benefits, but the combination of such systems may lead to benefits that again will lead to widespread adoption.

The advances in location-sensing techniques will soon make coarse-grained location information widely available. The recent achievements in sensor fusion techniques, location representation and software support will increase the development of applications in coarse-grained and by time fine-grained applications will arrive [8].

4.5 Summary

This chapter has given an overview of location aware computing. The key concept is that such systems must be able to sense and react to dynamic environments for example based on a user's location. To achieve this, various sensing technologies can be used for detecting users. This information can be used in addition to predefined information, for example stored in a database, for determining a user's access privileges. The Cricket indoor location system and the Cordis RadioEye, are examples of such systems. We have also given an overview of which sensing technologies that is likely to survive in the near future. Location aware applications will offer new and exiting services, such as "Locate my friends" utilities.

To achieve access control, location sensing systems should use access control methods, such as RBAC. However, RBAC is not suited for wireless networks. In the next chapter issues concerning mobile devices and the adoption of RBAC into mobile environments are presented.

5 Implementing RBAC into mobile environments

5.1 Introduction to RBAC for wireless environments

RBAC models have been developed for fixed network structures. They are also becoming more and more complex. In such computer systems, system resources like processing power, battery limitations, and available space for storing data are not limited in the same way as in mobile environments. In contrast to fixed networks, where an employee gets access to network resources only through fixed terminals, mobile devices makes it possible to be on the move and still get access to his or hers resources anywhere at anytime.

Mobile devices can move constantly, resulting in that location information must be continuously updated. This makes it necessary to adapt the RBAC models into models that can handle more flexible and dynamic requirements. Such a requirement would be to let the RBAC system making authorization decisions based on the spatial dimension in which the user is situated [14]. Therefore, implementing RBAC concepts into mobile environments will first focus on the mobile issues, and then propose an extended RBAC model that is better suited for these environments.

We will first mention some issues concerning mobile devices, before we present a modified RBAC model that better suits into mobile environments. We will use [17] and [14] as a background for how RBAC can be adapted into a wireless environment.

5.2 Mobile issues

The development within mobile technology has increased rapidly in the last couple of years. Today there exist a broad range of mobile devices, from the simplest ones, offering basic services like standard calling and SMS, to the more advanced devices implementing advanced features like PDA functionality, Java and the support for multimedia, like video and audio. The new and improved mobile devices enable more support for advanced programs. Until recently, software has been developed vertically. Thus, it is the manufacturer of the phone that decided the functionality, leaving no room for the user to independently adapt the features. However, the integration of Java and by using common operating systems, such as the Symbian [41] operating system, has resulted in horizontally integration of programs and services. This enables users to adapt their device by letting them add or remove features that suit their needs.

The integration of java enables developers all over the world to take part of the development of new and exiting applications. The advanced features enable new and exiting services to be developed.

In general, mobile computing is characterised by four constraints [18]:

- Mobile elements are resource-poor compared to static elements

The elements included in mobile devices are considerations of costs and the level of technology. There exist various forms of mobile devices with different weight, power, size and technology. Because of this computational resources like processor speed, memory size, and disk capacity will suffer. Mobile elements will improve, but they will be resource-poor compared to static elements.

- Mobility is risky concerning security

Users, with devices like laptops or PDAs would likely to bring them on the street or on business trips. The risk of losing it or getting it stolen is relatively high. Then important and sensitive information could be lost. In addition to security concerns, portable computers are more vulnerable to loss or damage.

- Connectivity are variable in performance and reliability

Mobile devices using wireless connectivity may gain high bandwidth in one building and low bandwidth in another. Outdoors a mobile client may have to relay on a low-bandwidth wireless network with gaps in coverage.

- Mobile devices rely on a finite energy source

While battery technology will improve over time, the need to be sensitive to power consumption will not diminish. Therefore, power consumption must be taken in consideration when developing hardware and software for mobile devices.

In Table1, some constraints concerning hardware are listed. We give a brief description and some examples of the latest in technology currently available.

Table 3 - Limitations in mobile devices

Limitation	Description
CPU – Central Processing unit	The processing power of mobile devices is rather poor compared to computers. For example, the latest Intel CPU is 3.4 GHz, while just 156 MHz in SE P900 and up to 400 MHz in most high end PDAs. Thus, Intel recently released a new CPU with up to 624 MHz with greater support for multimedia and 3D as well ass power saving technology.
Memory and Storage	Mobile devices suffer from low memory capabilities, resulting in slowness and unwanted delays. The latest in storage is the possibility for expansion cards in addition to built in storage. For example, today P900 supports Memory Stick Duo up to 512 MB.
Battery	The Battery is always an important aspect of mobile phones. Today it is common with a week on one charge, however this is likely to improve with time.
Display	The display has changed from small on-line two colours

	displays to big colour touch screens. The latest products for the professional marked have large and clear screens for supporting the advanced functions like office tools or multimedia. The P900 supports a 65,000 color touch screen.
Keyboard	Today there is possible to expand the usual small keypad with a full size keyboard. With touch screens and the possibility for voice activation, the traditional keyboard can be replaced. In addition it brings new functionality.
Programs	There are more and more programs integrated. For example, office tools, calendars, navigation, camera and video, multimedia content, web browsers, editing tools and more. There is also possible to download new programs and to upgrade operative system from the web.
Security	Security has mostly been intergraded in the SIM-card and by the telecom vendor. But, the emergence of the web and the possibilities for developing and downloading new programs, security must be implemented in offered applications as well. The security level must be increased, to hinder viruses and other getting access to sensitive data.

As mentioned the RBAC model is developed for computers which doesn't suffer form the limitations given above, except maybe from the security aspect. Important aspects to consider when adapting a RBAC model into a mobile environment are especially the performance (CPU), the amount of memory and storage. Even if the RBAC model itself can be implemented on a server inside an organizations network there must be a service on the phone that makes it possible to give the user the right privileges and operations based on the location. Location data will be an important part of a mobile RBAC solution. It is possible to adapt the RBAC model so it will be suited for mobile devices, however it is usually not the mobile device which makes access control decisions, when accessing resources on an organizations network. It is usually a server situated in the organizations network, which must take the access decision and thus must implement RBAC.

Figure 14 shows the classic hexagonal mesh model, which is used to visualize cell coverage in a geographical area. For traditional mobile coverage the cell size have shrunken considerably the last couple of years. With the introduction to 3G, the cells must be even smaller for offering the high bandwidth to each user. Cell sizes may now be 100m or less in urban areas. Each cell makes use of a frequency set. A neighbouring cell can not use the same set of frequencies, due to interference and the number of users that can be handled.

Figure 14 - Cell structure with users situated at different locations

Mobile computer users can be in different locations and may want different behaviour of their mobile devices as well as getting dynamically access to resources. However, this requires an access control system that makes use of location for determining the available permissions and operations a user can access. The traditional RBAC model cannot be used for making decisions based on location. However, an extended model may be used.

5.3 A possible RBAC model suited for mobile environments

In a mobile environment, where users can be at different locations, the traditional RBAC model must be extended. The system should be able to base its access decisions depending on the spatial dimension in which the user is situated [14]. Therefore RBAC must use location as a parameter for access decision. There have been discussed several solutions for spatial security, but it is only one model that have extended the RBAC model with spatial security. A proposed model that copes with this requirement is Spatial Role Based Access Control (SRBAC) [14].

With this model it is possible to specify spatial constraints on enabling and disabling of roles [14]. SRBAC enables user access to organizations network resources, anywhere at anytime, through their mobile terminal and it controls the available set of permissions based on where the user is situated. As an example [14], consider a doctor that has permissions to access a patient's electronic record (EPR). However, due to the sensitive information in the EPR, the doctor is only authorised to access the EPR in designated areas. So, if the doctor tries to access the EPR, for example in the cafeteria, where he does not have the right permissions, he will be rejected to do so. The access request is denied.

With the traditional RBAC model in a mobile environment one would need to define roles for each location in an organization. In large organizations with many location areas, the amount of roles that must be specified becomes considerable. Also many roles may have a lot of the same permissions assigned to them [14].

In the SRBAC model, which is suited for mobile environments, it is possible to achieve more flexibility by dynamically change the permissions in a role based on a user's current location. Thus, the role is dynamic, meaning that it may have different permissions assigned on two different locations. This reduces the number of roles that are needed to specify in the system and therefore security administration is simplified [14].

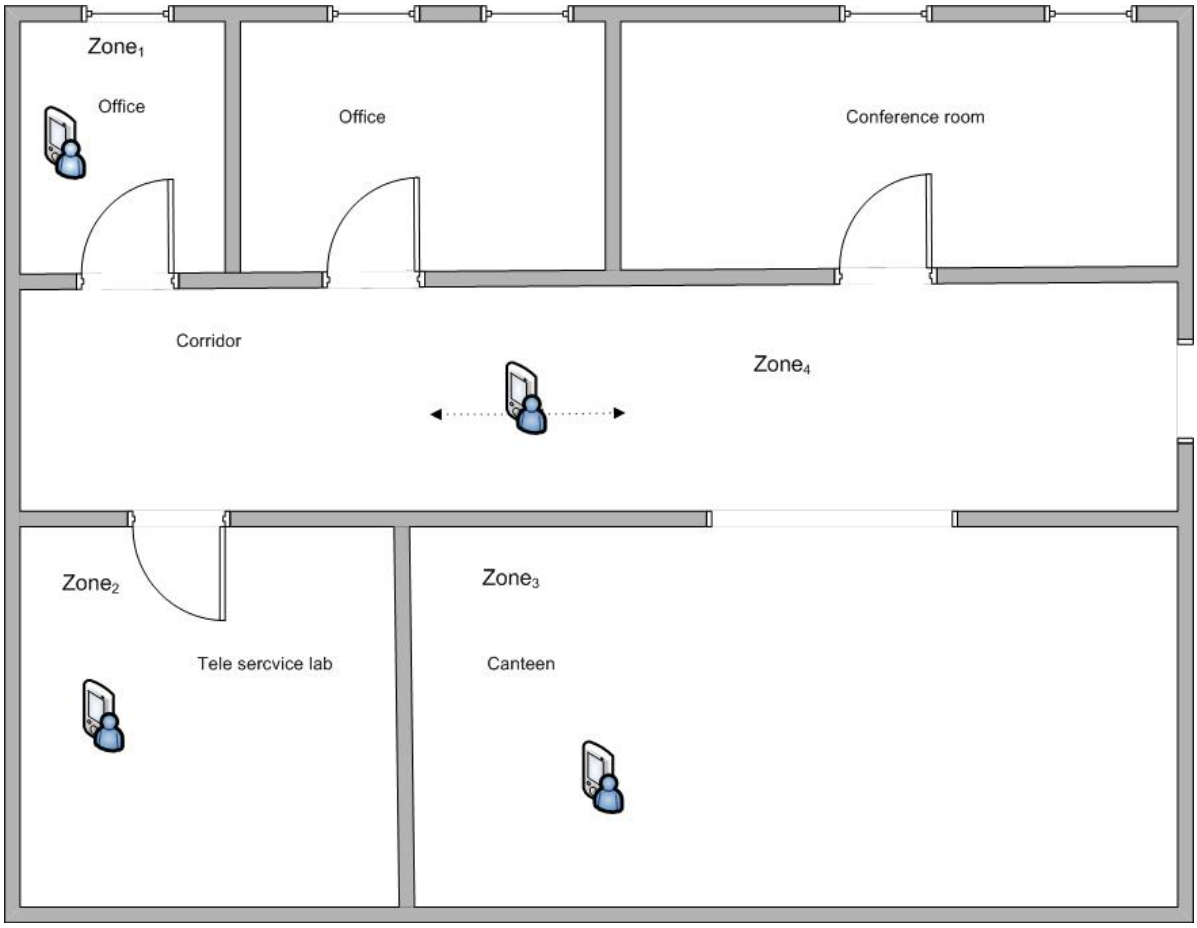

Figure 15 - A building representing users at different Zones

For example, consider Figure 15 which is a fictitious representation of a wireless environment in a university building. The building consists of different location areas or zones, $Zone_1$ at the office, $Zone_2$ at the Teleservices lab, $Zone_3$ at the canteen and $Zone_4$ in the corridor. Each role defined in the system have different permissions associated dependent of the zone the user is situated.

Consider for example a user Bob, an engineer of department. He is assigned to the role *dept_engineer_role*. This role has a set of predefined permissions that is relevant to his working position. When Bob is located at his office, he has access to all resources on his office, including printing at a shared printer. He has also access to resources on the network, which is related to his work. However, the full permission set is only available at his office. When he leaves his office and enters the corridor the location system changes his permission dynamically. For example, he may no longer access classified resources on the network, such as documents and files. When he moves on to the teleservices lab he

has the same role *dept_engineer_role*, but again the system are dynamically changing his permissions, according to his job functions at this location. Table 4 gives an example of a Location Permission Assignment List [14], which defines Bob's available permissions at the different zones.

Table 4 - LPAL - Location Permission Assignment List

ROLES	LOCATIONS	PERMISSIONS
dept_engineer_role	Zone1	p1,p2,p3
dept_engineer_role	Zone2	p1,p2
dept_engineer_role	Zone3	p3
dept_engineer_role	Zone4	Ø

Zone1 may have the permissions p1, p2 and p3 associated with it. Zone2 may have p1 and p2 while Zone3 may only have p1. Zone4, the corridor may have no permissions associated with it (denoted by Ø).

The SRBAC model is based on the same model components as RBAC. However, location is now added as a basic component. Therefore we shortly introduce those components with respect to SRBAC.

5.3.1 Core SRBAC

Core SRBAC extends the existing Core RBAC model [13] to be able to allow for location. SRBAC consists of the following basic components, Users, Roles, Permissions, Sessions and Locations. Users are considered to be mobile users. This is different from the RBAC definition of a user, which is restricted to fixed computers or terminals. A SRBAC user can establish wireless system communication with system resources to perform some operations, based on their role in the organization. Roles hold a set of permissions. Permissions are approvals for executing operations and vary depending on the role and on the location in which the user is situated. Location is represented by symbolic expressions called location expressions [14] that describe location domains. SRBAC assumes that the underlying network structure can identify and verify location of a user.

LOCATIONS consist of many location domains, or zones, and it is possible to divide it into sub areas, called primary location cells. Because it can be unpractical to operate with primary location cells SRBAC introduces logical location domains that reflect organizational location infrastructure and organizational security policy [14]. It is then possible to define locations such as offices and laboratories, like in Figure 15. These logical location domains can be defined by using the primary location cells.

5.3.2 Hierarchical SRBAC

As with RBAC, SRBAC also supports hierarchies. RBAC define hierarchies as an inheritance relationship between roles. For example [14], one role r1, can inherit permissions from another role r2, only if the permissions in r1 are part of r2. However, in SRBAC, the permissions vary with location and therefore it is necessary that the role hierarchies also depend on location. So, role r1 can inherit permissions form another role r2, only if the permissions in r1 are part of r2 and both r1 and r2 are activated in the same

location belonging to a location domain. Thus, the current location must be part of the LOC set.

5.3.3 Constrained SRBAC

SRBAC supports separation of duties to be enforced on roles. However in contrast to RBAC, SRBAC allows for different constraints on roles at different locations. For example, in Figure 15, a user at his office may activate two roles at the same time, while in the canteen the user can only activate one at a time. Thus, the permissions changes based on the location. In SRBAC SSD and DSD relations are called SSSD (Spatial SSD) and SDSD (spatial DSD) relations.

Spatial Static Separation of Duty relations

Spatial SSD enforce constraint on two roles that cannot be activated at the same time, because of conflict, on a current location, but they may be activated on another location. Thus, a user may never activate two roles with a SSSD relation for a special location [14]. This is a stronger separation of duty relation than in RBAC, but if the SSSD relation in SRBAC was valid for the entire location space, it would be similar to the traditional RBAC constraint [14].

Spatial Dynamic Separation of Duty relations

In RBAC, DSD relations offer the capability for a user to be authorised for two or more roles that do not create a conflict of interest, when activated independently, but when acted simultaneously policy concerns are initiated. In addition to this SDSD relations allow users to be assigned to two or more roles that do not make a conflict of interest when activated simultaneously, at a special location. However, at another location those roles could make a conflict of interest situation. This offers a great advantage in contrast to RBAC, where DSD constraints are forced on the entire organization [14]. With SDSD, constraints can be specified for specified locations, thus a user may not activate conflicting roles at that location. However, the user can activate those roles at a different location.

5.4 Summary

This chapter have given an overview of an modified RBAC model suited for mobile environments. It is important to implement the more advanced functionality for example on the fixed network within an organization, while the light functionality is implemented in the mobile devices. Since SRBAC has been extended from RBAC, it provides all the functionality of RBAC, in addition to location.

We have developed a prototype application that works in a wireless environment. This application uses location aware technologies as well as RBAC and SRBAC concepts.

6 Prototype Implementation

6.1 Introduction - The Implementation phase

The implementation process started by defining an architecture as shown in Figure 16. We decided to use Java as the programming language, because of its broad horizontal support and because of our experience in java.

We decided to divide the application scenario into four programs. The first program should be located on the Access Point (AP), scanning for devices. The second, on the server for handling Access Point messages. The third program was developed for the mobile phone and the fourth also on the server for handling requests from the mobile phone and for checking access requests against the database.

This basic application was then going to be implemented in a framework for teleservices called ActorFrame from Ericsson. By implementing our application into ActorFrame we were able to use GSM services in addition to Bluetooth for positioning. The migration from our application into ActorFrame was implemented with the help from our supervisor Fritjof Boger Engelhardtsen and Geir Melby at Ericsson in Asker. The development project has lead to two applications. The first is our basic application. The second application has been further developed by our supervisors and use ActorFrame as a basis. It uses teleservices like SMS and GSM localization. The source code can be found on the enclosed CD.

In this section we will focus on our first application and then give a short description of the second application.

6.2 Our basic application

As mentioned our basic application consists of four programs. Figure 16 gives an overview of the architecture. As you can see we use Access Points for retrieving the location of devices. This information is forwarded to the Access Point server which updates the database with location information. Then we have the mobile devices which send access requests to the Servlet server, which in turn makes a response based upon the data in the database.

The application is divided into two parts. The first part exists of two programs, the Access Points and the Access Point server. The last part consists of a MIDLet, on the mobile device, and the Servlet server. The server receives access requests from the device and makes a decision based on the parameters received and the information in the database.

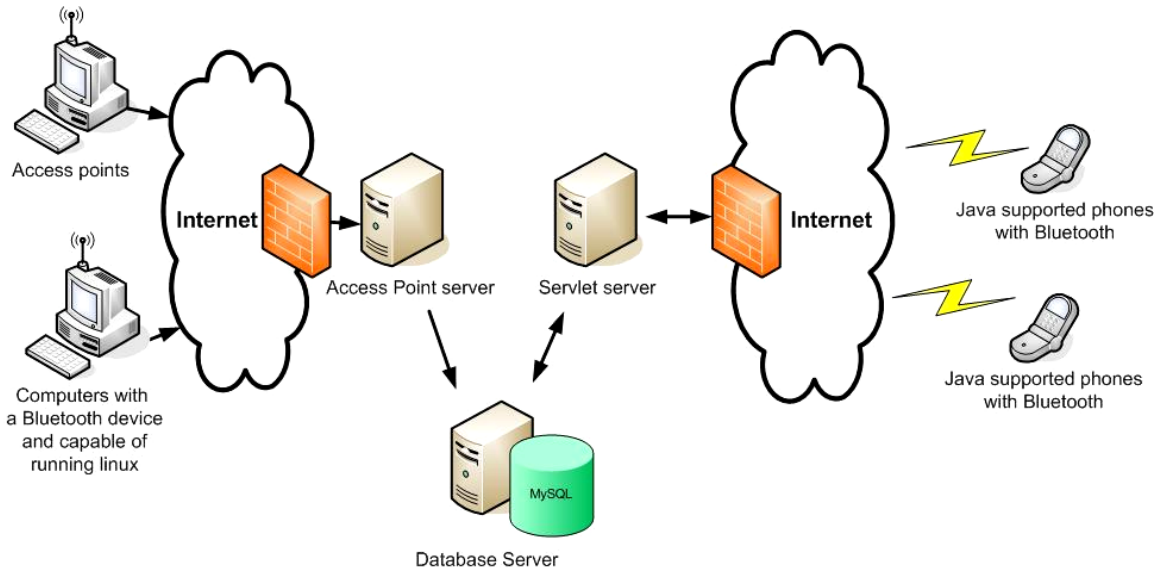

Figure 16 - Architectural design

Starting from left in Figure 16, we have the Access Points, which is a computer with a Bluetooth device attached to it. These machines run Linux, and do not need to be very powerful.

In the center, we have the various servers which are protected by a firewall, the Access Point server, the Servlet server, and the database server. For the server we use a HP Intel Xeon 3 GHz computer, which is more than powerful enough for handling this system. However, when the amount of mobile devices increases, the information load increases and, more performance is needed.

6.2.1 The Access Point client

The Access Point client is a standard Java program developed with the Java 2 Platform, Standard Edition [35] . This program runs on Linux computers. The Java program makes use of a Linux program that scans for devices, "hcitool" [36]. The Java program uses a while loop, making the program continuously searching for devices. The discovered devices, those within range, are then sent to the Access Point server and stored in a database. The APs are used to set up wireless zones, which enable us to define wireless zones. These zones can then be configured with user privileges and operations. Then the system can determine the assigned privileges for a user in a current zone.

There are several reasons for choosing Linux for the clients. Most importantly, we use a Linux program called "hcitool", which scans the local Bluetooth network and displays the result on the screen. Another important issue is that all the clients are located outside a firewall, and Linux computers are less vulnerable to hackers. Linux computers are also more stable and they can run for days, without the need for restart. This is an advantage since the APs do not use a monitor or any input devices like a keyboard or a mouse.

As mentioned the AP scans for devices and sends that information to the AP server. The AP server can then display the devices within range at a current location. The information includes a timestamp, the current location, the device MAC address and the device name.

The timestamp is the current time when the AP discovered the device. The location is the current location zone, defined by an AP, where the device is located. As shown in Figure 16, we have two APs, one placed at the Teleservices lab and the other placed at the Siving lab.

Each mobile device with Bluetooth has its own unique MAC address. This address is stored in the Bluetooth chip in the mobile device. Since this number is unique on each Bluetooth chip, we are able to uniquely identify a device, which again are used for authorization. A mobile device does also have a device name. This is a string which is more logical to human beings. This name is editable, not unique, and can be whatever a user wants, but often it reflects the mobile device or a user's nick. The example below shows how this information is shown at the AP servers. (The timestamp is leaved out)

```
Location:         Device MAC:                Device name:
Teleservices lab  12:E3:61:0A:8D:12          P900
Siving lab        00:03:56:7B:F1:3A          IBM-12983
```

The AP makes one UDP packet for each device it finds and sends them to the Access Point server. The UDP packet contains all the information above, and the server passes that information into our database. For each scan with the "hcitool" we also send a packet with MAC address = 00:00:00:00:00:00. This invalid MAC address is used on the server side to separate a scan with the "hcitool". Even if the tool doesn't discover any devices it will send the zero packet. It is only used for timeouts, by comparing the timestamp from when the device was detected with the present time. If it has passed about 20 seconds and the device isn't rediscovered, the device is deleted from the database. Because a device may be out of range for a short time, maybe 2 seconds or so, and then suddenly within range, we prevent the device from being registered in the database all over again.

One scan by the "hcitool" takes roughly 2 seconds and can return anything from 0 to 255 devices. However, the time varies on the amount of devices within range. Below, is the while loop that is used when scanning for devices:

```
while (true)
        Vector = getAllDevicesFromHcitool( );
        For each device
                send UDP packet containing device information to server
        send zero-packet to server
```

We use a vector, a self resizable table, for temporal storage of the scanned devices. Then we send an UDP packet for each device and at the end we send the zero packet. By using UDP we have no guarantee that the packets arrives successfully or in order. However, UDP is much faster than if we had used TCP. TCP needs to establish a three way handshake, and thus increases in delay. It also supports retransmission of packets which, with enough devices, could overload the system due to the amount of data sent with TCP. For example, HTTP is based on TCP and causes reliability, but unwanted delays. It is referred to as the "World Wide Wait" [28].

The APs do not get a reply from the server if a packet is received. This one-way communication is sufficient, since the APs are only meant for continuously sending updated information to the server. Packets out of order or packets not received do not affect the functionality of our program, because the same information is sent multiple times.

6.2.2 The Access Point Server

The Access Point server is also implemented using the Java 2 Platform, Standard Edition. It receives all packets from the AP clients. Each packet is handled individually and if new devices have been discovered, the devices are added in the database. The Access Point server is only used to receive device information, like the MAC address, location and timestamp, from the different APs.

There are only two types of packets that the server receives. The first is the zero-packet which is the last packet sent when one scan is finished. The second is the packet containing the devices. Devices that already exist in the database are updated. This is necessary to prevent devices to be removed because of the timeout mechanism.

As described earlier the zero packets are used for timeouts. The algorithm that handles the received packets is shown below:

```
for each USP packet received
        if packet = zero-packet
                do timeout test for all devices in database located on this access point
                and remove any device that is idle
        else
                if device is already in the database
                        update the database with the new information
                else
                        insert new row containing the new device
```

For each UDP packet that is received, that is not a zero-packet, and if the device already exists in the database, it is updated with the new information, such as the timestamp. If the device does not exist in the database, add it. When the packet is a zero packet, a timeout test it done with every device registered at a current location. If a device has been registered in the database, but is not within coverage, it is deleted. The information received from the AP is stored in the database as shown in Table 5. Here we see three rows, meaning that three devices are registered at the current time. Here we can see the timestamps, which is a number used for timeouts, the LocationMAC, which is the MAC address of the AP so it is possible to find the current location of a device, the LocationName, which is the name of the zone or coverage area, The DeviceMAC, which is the MAC of the mobile device, and the DeviceName.

Table 5 - Storage of device information

	TimeStamp	LocationMac	LocationName	DeviceMac	DeviceName
1	12897598	00:0E:45:32:E3:02	Siv. Ing. Lab. HiA Grimstad	12:E3:61:0A:8D:12	P900
2	12897543	00:20:E0:39:5C:2D	Teleservices lab	00:03:56:7B:F1:3A	HP-62376
3	12897543	00:20:E0:39:5C:2D	Teleservices lab	34:3E:67:D1:8A:09	Thor's Phone

In Figure 17 we see a live capture of mobile devices discovered at two different locations, in the teleservices lab and in the student lab (Siv. ing lab). This capture is taken from the server, thus the two APs have sent information to the server. We see that there are two devices in the teleservices lab and two devices in the student lab. The two APs also send a null packet at the end of each scanning.

```
recived packet --- /10.5.1.7 - Teleservice lab - 0
recived packet --- /10.5.1.7 - Teleservice lab - P900
recived packet --- /10.5.1.7 - Teleservice lab - 0
recived packet --- /10.5.1.7 - Teleservice lab - P900
recived packet --- /10.5.1.7 - Teleservice lab - Ericsson t68
recived packet --- /10.5.1.7 - Teleservice lab - 0
recived packet --- /128.39.202.126 - Siv. ing. lab. HiA Grimstad - Nokia 6600
recived packet --- /128.39.202.126 - Siv. ing. lab. HiA Grimstad - HEATER
recived packet --- /128.39.202.126 - Siv. ing. lab. HiA Grimstad - 0
recived packet --- /10.5.1.7 - Teleservice lab - P900
recived packet --- /10.5.1.7 - Teleservice lab - Ericsson t68
recived packet --- /10.5.1.7 - Teleservice lab - 0
recived packet --- /10.5.1.7 - Teleservice lab - P900
recived packet --- /10.5.1.7 - Teleservice lab - Ericsson t68
```

Figure 17 – Receiving device information from different locations

6.2.3 The Servlet server – handling the mobile requests

The Servlet server, Figure 16 and Figure 18, is developed using the Java 2 Platform, Enterprise Edition [34]. As the name implies it is an HTTP servlet that handles requests from the mobile devices. Servlets makes it possible to build Web-based applications with component-based, platform-independency. Servlets are capable of accessing all the regular Java APIs along with the JDBC API [46] to access enterprise databases.

Our servlet has a connection to the same database as the AP server, and uses that information for carrying out the correct response, based on the request. The data sent between the mobile device and the server is done through URLs. Below two URLs are shown:

http://ourserver.hia.no/rbacservlet?Login=Flemming&Passw=1234
http://ourserver.hia.no/rbacservlet?Action=getpermissions

The first URL sends the login name and user password to the server for authentication. The servlet then makes a random cookie for this HTTP session, it will also add this cookie to the database. Later we can check the cookie rather than constantly sending the login name and password back and forth. Finally, when the user is authenticated, the assigned

permissions for this current user are requested. By asking for permissions we mean that the current user requests the server for what he or she is allowed to do in his or hers current location. When the assigned permissions are shown, a user may attempt to do an operation or log off, for the current user.

When a user is logged in, we use the cookie to find out if it is the authorized user that makes requests. In our application we use the MAC address to distinguish between devices, however, this MAC address is read from the database and not directly from the mobile device. This means that we have to register those MAC addresses in the database in advance.

The Permissions consists of objects, example a file, a light switch or whatever the system has implemented, and an operation associated with the object. Permission examples would include, "unlock front door", "Send SMS" or "turn on lights at 3rd floor".

The user may also attempt to perform one of the operations shown in Figure 20. The user data is evaluated by the server. If the user has the right permission privileges it is performed. For the server to be able to perform an operation, a manually inserted code is necessary. It is advised to make a separate class extending "Thread" [47] so it is less likely to interfere with the Servlet server.

Figure 18 shows a flow diagram of the Servlet server, which describes the request/response scheme. There are four states in addition to the "*idle*" state. The "*idle*" state is where the program only listens, thus no operations are performed. These are, *"Do the operation"*, *"Make cookie and add session to database"*, *"Send permissions"* and *"Remove session from database"*. The user sends a request, and the server responds. After an operation the server returns to the "*idle*" state.

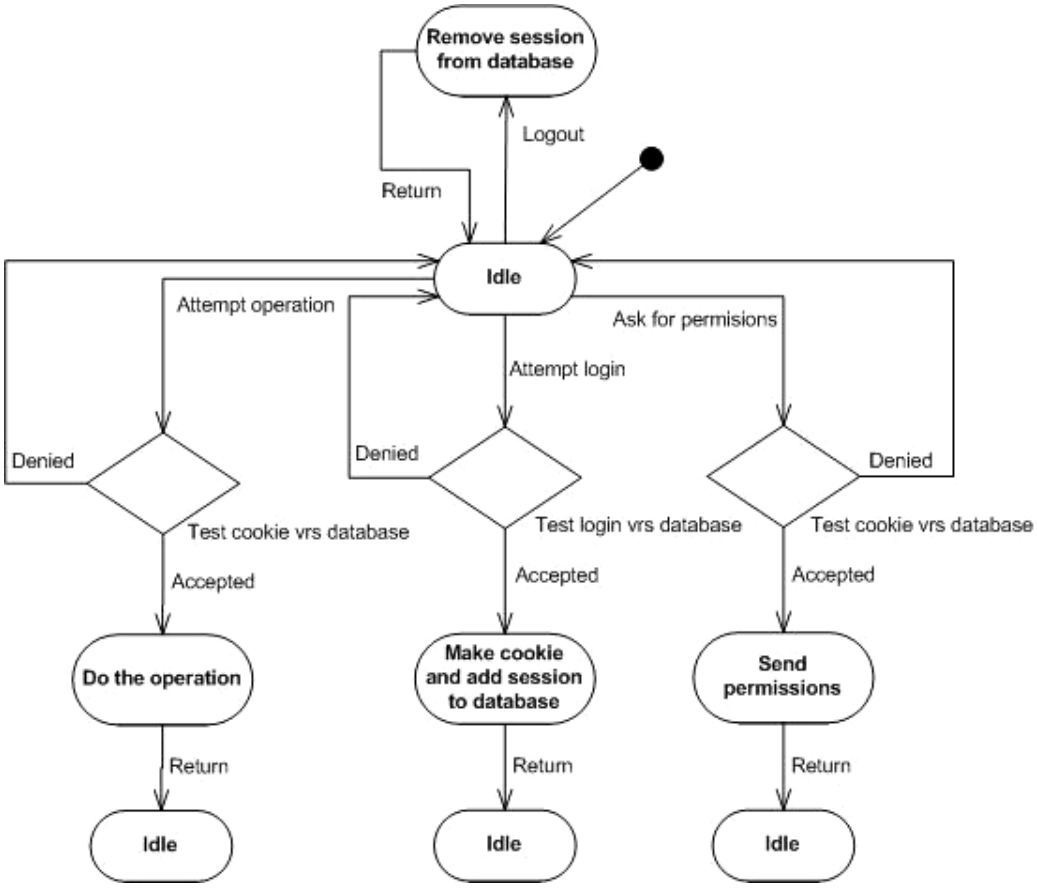

Figure 18 - Servlet flow diagram

A pseudo code of the flow diagram, Figure 18, is shown below.

```
Receive request
if request does not contain cookie
        get login and password from request
        if login and password are correct
                make session-cookie and add to database
        else
                access is denied
else
        if cookie = valid
                if action = ask for permissions
                        send permissions
                else if action = do operation
                        perform the operation
                        and inform the user
                else if action = logout
                        do logout
                        and cleanup for the current user
                else
                        action parameter is invalid
        else
```

cookie is false and access is denied

We will take a closer look at the message scheme in chapter 6.2.4, " The MIDP application".

6.2.4 The MIDP application

MIDP (Mobile Information Device Profile) [38] is an API for developing Java based applications for mobile devices such as cell phones and mainstream PDAs. MIDP adds functionality to the CLDC (Connected Limited Device Configuration) [39], which is the basic programming interface with a virtual machine for resource-constrained devices.

With Java for small devices we were able to make an application that would run on a mobile device. The application should run fine on all Java devices with support for MIDP 1.0 and CLDC 1.0. The phone also needs Bluetooth for being discovered by the AP's and GPRS support for sending the information requests to the server. In our case we used the Sony Ericsson P900 and it worked out successfully. There has also been stated that the SE P900 is more stable and error free when running Java applications, than in comparison to other phones like Nokia.

Figure 19 – The Sony Ericsson P900 mobile phone

The MIDP application allows a user to log in to the system, depending on where the user is situated. A user is associated with one or more roles, these roles have assigned permissions which a user assigned that role can perform. For example, in one location a user is assigned two roles. He can then access permissions assigned to these roles. However, in a second location, the user is only assigned to one of the roles and thus he has only access to the permissions assigned to that role.

The user enters his or her login name and password. If the login attempt was successful, and the user is associated with his assigned roles, a list of assigned permission is shown. Then the user can perform one or more of those operations. The permission list is dynamically updated dependent on location, so if the user leaves the current zone and enters another the list is updated and the name of the zone is shown at the top, as shown in Figure 20.

All communication is performed over HTTP. In Figure 20, a picture of the MIDP application is shown. For example, let us consider a user that is located at the Teleservices lab. He starts the application, assuming he is within reach and detected by the AP. In part 1 the user will enter his login name and password and attempt to log in. If the login was successful, the user's permissions are shown on the screen, as shown in part 2. The user is then allowed to select an operation that he wants to perform, in this case turning on the coffee machine, Make Coffee. In part 3, a confirmation screen appears to inform the user with appropriate information of what has been done. Then the user may want to perform more operations or exit the application. If he decides to go back and do more operations, he will retrieve his permissions as before, but now the actual operation that he just performed is in progress.

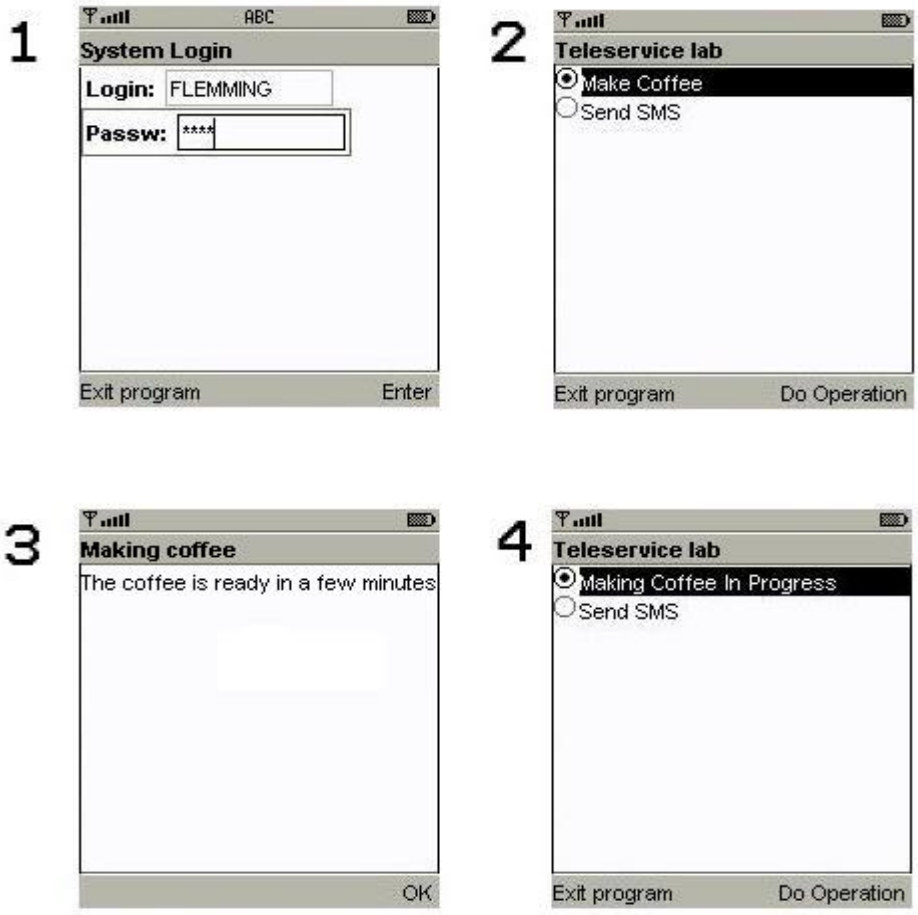

Figure 20 - MIDP interface for a current user

6.2.5 Message sequence

Figure 21, shows a message sequence diagram which depicts the interaction between the user and the different applications. This scheme shows the messages that are sent when a user logs on, performs operations and logs out.

When a user starts the program on his mobile device, the first screen that appears is the login screen. When the user has entered his login name and password, the MIDP application transmits the login name and the password to the servlet. Then this information is checked against the database where the system administrator has added the user and his role assignments. A cookie is made from a random generator and added to the database and to the HTTP session, so future requests are verified through the cookie rather than resending login name and password multiple times. After a valid login, the assigned permissions are requested.

Then a user can execute an operation. When he chooses an operation, this information is sent to the servlet server and again the user is validated. This is to prevent a malicious hacker to abuse the system. After the revalidation is done the Servlet can perform the specified operation. When performed the user is presented with a screen informing the user if the operation was successful or not. Then, if the user does not exit, but presses OK as in part 3 in Figure 20, the updated list of permissions will be displayed again.

When the user wants to exit he executes the exit command which sends a logoff command to the Servlet. The Servlet then removes the HTTP session cookie and the cookie information from the database.

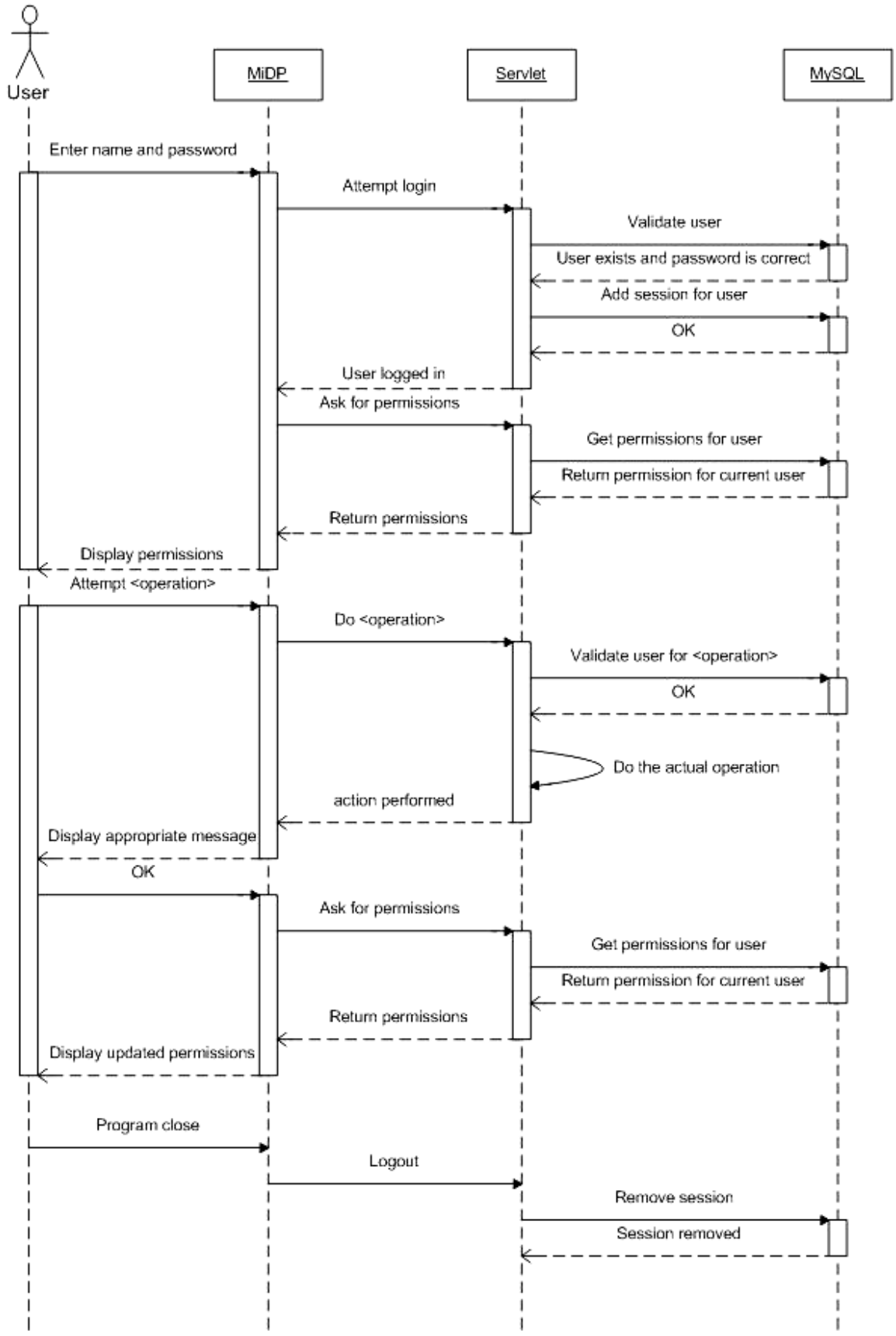

Figure 21 - Sequence diagram for the MIDP application

6.2.6 The database, MySQL

Our database is implemented using MySQL [44]. It is an open source database available to many different platforms including, Linux, MAC OS X and Windows. We selected MySQL based on our previous experience, and because of its simplicity. However, MySQL is no a relational database, therefore relations must be handled with the SQL queries, when performing queries against the database.

Our database contains, in comparison to the RBAC specification [13], main tables for Users, Roles, and Permissions. We also use elements from the SRBAC model [14] such as location. We have relations, which are established at runtime, between Users, Roles, and Permissions. This reflects the RBAC model and allows a user to activate one or more roles in a session.

Because this application only is a prototype none of the more advanced components of RBAC are implemented, such as support for constrains. With the implementation of locations we decided to do this in Roles rather than in Permissions, which is in according to the RBAC principles, but with a relative small editing of the source it could just as easily be implemented in Permissions, such as in the SRBAC model [14].

The tables and the relations

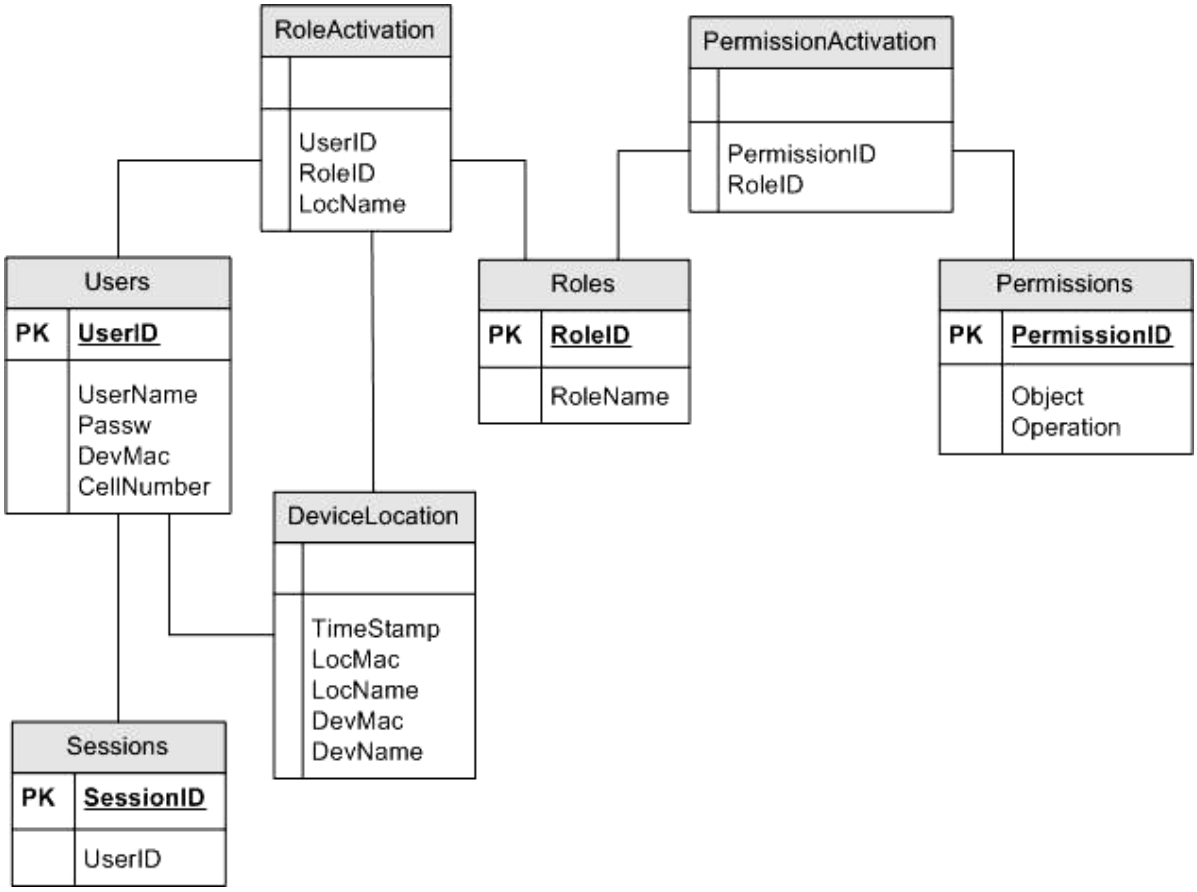

Figure 22 - Database structure

Users of the system are represented in the User table. Each user has a username, which also is referred to as the login name, and a password. We mean that a mobile device is personal and therefore a user is also associated with a mobile device, with its corresponding MAC address and the phone number. The MAC address limits a user to use only one device for a login. The user's phone number is stored in order to use services like sending SMS. We thought of sending a SMS message to a user entering a zone, but this is not implemented in our prototype. If it was possible to get the phone number without manually register it in the database, we could use it as an information message to guests of the system. However, if we had the time, we could use Bluetooth or WAP Push for sending messages like this. The information messages could be used to inform a guest that there exist wireless zones where the user can get access to resources. For example, it could be possible for a guest to use the local wireless LAN for surfing on the internet or use some related service.

DeviceLocation is a table containing all the location information stored by the Access Point server. This table is only for devices that are detected by the Access Points, even devices not recognized or previously stored by the server are stored here. As described earlier, in 6.2.2 the Access Point Server, the database contains a Timestamp, LocMac, LocName, DevMac and DevName.

We see that the user table is connected to the session table, The Role Activation table and the Device location table. Because a user can activate a role at a give location only with an authorized device, this relation is necessary. The Session table holds Session data. Sessions keep track of the session cookie described earlier in chapter, 6.2.3, and which the user cookie belongs to. The roles table contains all the roles in the system. These roles can map the roles of an organization. A role is identified with a Role ID and a Role name. The Permissions table maps the permissions that are available to the different roles in a system. In compliance with the RBAC model [13], Permissions are a combination of objects and operations. Permissions also have an ID so they can be identified. The User and Role table are connected through the Role Activation table. This corresponds to the UA relation in RBAC. In the same way, the Role table and the Permission table are connected through the Permission Activation table, which correspond PA relation. With this relation it is possible to define which permissions a specified role may use or not. Thus, with the assigned permissions a role can perform one ore more operations on objects, such as turning in the coffee machine.

6.3 The Second application with the integration of ActorFrame

ServiceFrame is presented in chapter 1.2. It consists of three layers as shown in Figure 23 [30], JavaFrame, ActorFrame and ServiceFrame, which is a framework for developing service applications. By using asynchronous communication through mediators, active objects are able to communicate with each other.

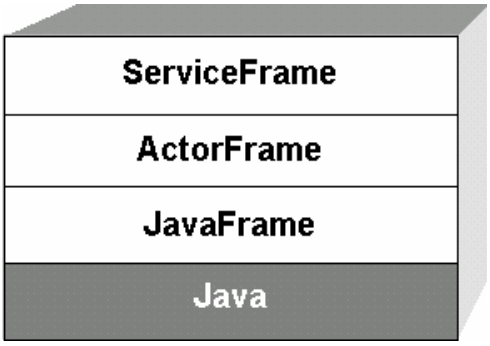

Figure 23 - Ericsson's framework overview

We have been involved in the start up phase of the project, but the implementation is carried out by our supervisors, Fritjof Boger Engelhardtsen, Frode Hansen, and Geir Melby from Ericsson, which is the founder of ActorFrame. The only part of our architecture that has been implemented into ActorFrame is the Access Point server. Thus the Access Point clients are exactly the same as in our basic application. This version of the application is not enclosed on the CD. ActorFrame is a framework developed by Ericsson, and it is confidential. However, they let us using it, in connection with our student project.

With the Access Point clients being unaltered from the previous version, we will take a look at the changes made to the new Access Point server.

6.3.1 The new Access Point Server

Since ActorFrame have built in support for databases, through JBOSS [32], our MYSQL database is now leaved out and implemented in ActorFrame. Because of this, there is no need to update the database any longer. The new Access Point server has an internal mechanism to detect changes in the Access Point clients. This means that only new devices that enter a zone are being updated. The information is sent further into the system. Then the Access Point server makes what ServiceFrame calls ActorMessages. ActorMessages are messages that are used for requests and responses. These messages are then sent to the ServiceFrame process described below.

6.3.2 The ServiceFrame process

ServiceFrame makes use of actors. An actor represents a user in our case. The actor object is capable of knowing its location through our APs and by using GSM localisation. GSM localization data in addition to data registered by the APs will prevent a user from logging on to the system from another location. This combination of data is called sensor fusion. It is performed a double check on the location where the user is situated. The GSM location service is gained through a service server at the lab and through a connection with a services provider, in our case through Telenor. It is a specialized command from Telenor that can determine, with the accuracy of 10m, where a device is located by latitude and longitude. This information is used by the actors to determine what the user is allowed to do. Below is the UML 2.0 [48], a modelling language, diagram that represents the new architecture of the AP server. All modelling in ServiceFrame is done with UML 2.0 and enables the visualization of the prototype.

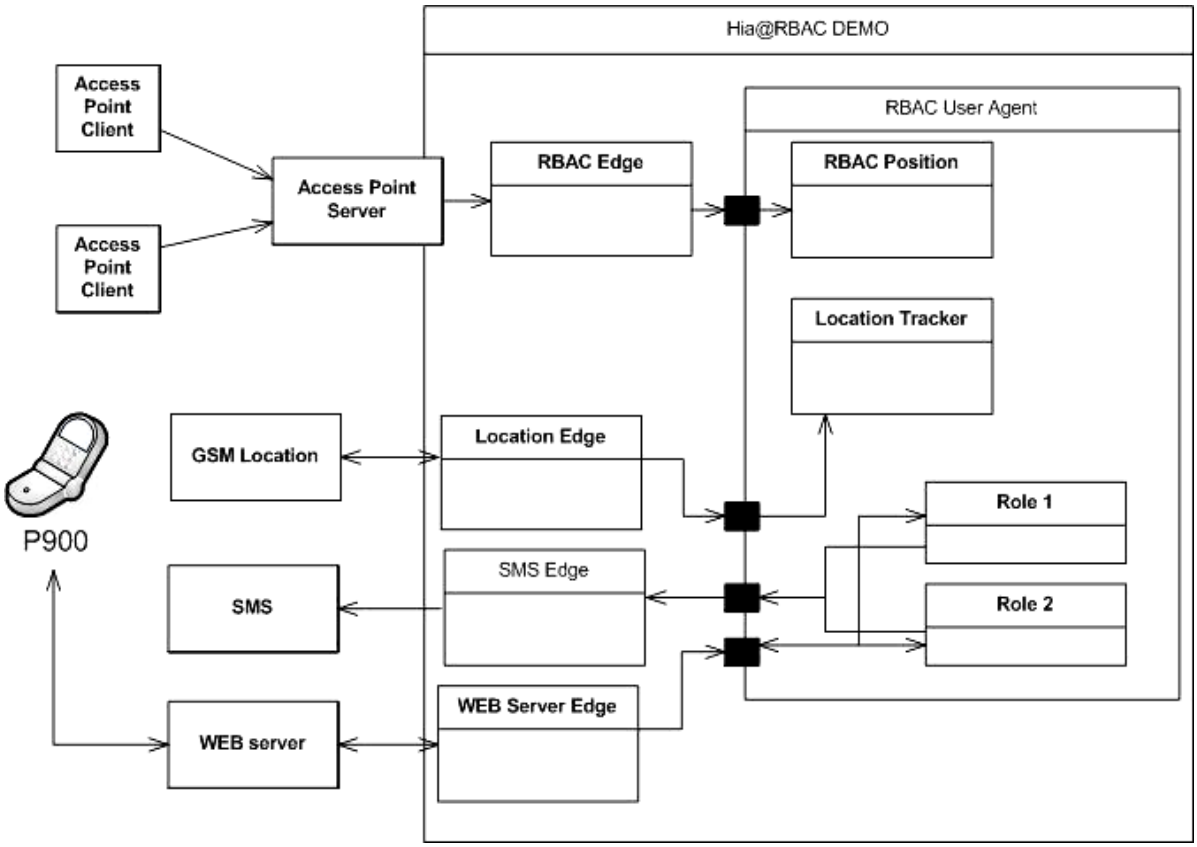

Figure 24 - The RBAC architecture implemented in ServiceFrame

We see that the AP clients are connected to the new AP server. The AP server then forwards messages to the RBAC User Agent based on the operation that is requested. The AP Server is connected to the RBAC Edge which again is connected to the RBAC user agent. All the messages are sent through the RBAC edge and to the RBAC User Agent. The RBAC User Agent keeps track of the position of a current user. Then requests from these users are forwarded to the edge that is capable of doing the operation that is requested. The mobile device communicates with the RBAC user agent through the Web Server Edge. The SMS Edge and the Location Edge is used to provide teleservices.

The new application is tested, and the only thing that is necessary to set up is the AP server. The AP clients are unchanged. Also the MIDLet and the Servlet server are unchanged.

6.4 Summary

This chapter has described our prototype application as well as the integration with ActorFrame. It is designed to work in a wireless environment and by using Access Points wireless zones can be defined. Then different permissions can be assigned based on the location of the user. The new established teleservices lab and ActorFrame has also enabled us implement Tele services. Our application does support access control and implements concepts from the RBAC model in addition to the location concept from the SRBAC model.

7 Discussion

7.1 Introduction

In this section we will discuss RBAC and how it can be implemented and adapted into a mobile environment. We will also discuss our prototypes and we give proposals and improvements for further development. We will also discuss the improved implementation, developed by our supervisors, and its benefits in according to our prototype. Since we only have been partly involved in this development process we do not have enough insight to go deep into the discussion.

7.2 RBAC

We have found that RBAC is an efficient method when access control is important and when information and resources increases. It forms a basis for access control in large information systems and when large-scale management is needed. Access control is only a small part of an organizations security policy. In addition security terms like confidentiality and integrity must be paid attention too.

In contrast to other access control methods such as UBAC, where an administrator must assign permissions to each user, and PBAC, which uses policies for access definitions, RBAC assigns permissions to roles which can reflect the different job functions in an organization. It then uses these roles to grant access to resources and information on computer networks.

UBAC and PBAC, among others access control methods are dependent on the discretion of the network administrator. However, in RBAC the permissions are assigned to predefined roles and therefore the discretion is only done when the roles are defined, and not when the role is assigned to the employee. In addition RBAC is policy neutral and supports the principle of least privilege. By also using constraints, which is argued as an advantage for RBAC, it is possible to define which roles that can be activated at the same time, hindering conflict of interest situations. For example, an employee Bob could change his working position. Then an administrator only needs to reassign Bob to the new role. If UBAC was used in this example, all the privileges in Bob's job function had to be disabled, and the new privileges in his new job function had to be assigned. This is inconvenient and a time consuming task especially when the number of users grows.

Since RBAC supports role hierarchies it is possible to map an organizational structure, and enables roles to inherit permissions and prohibitions from other roles. This is not supported in any other access control model, mentioned in this paper.

In addition, RBAC has a reference model, Figure 2, which enables one to choose the desired functionality that is needed in a system. Then it is possible to only buy products that support those functions that are needed, or by using the package utility to predefine components needed for a system.

In contrast to early methods for security like the Access Control Matrix, Access Control Lists and Capabilities which is used in low level systems for access on resources such as files or processes, RBAC is can be used on a higher level. RBAC defines operations that can be defined for files, resources or higher level operations such as turning on or off objects, like a coffee machine or the light. In theory these operations can be widely defined. RBAC can also be combined with other access control mechanisms like the traditionally mandatory and Discretionary Access Control methods. Concepts from CBAC could also be used to implement some sort of quota control.

RBAC exists in implementations in everything from operating systems, database management systems and healthcare systems. RBAC are also used in firewall architectures, and for controlling access in mobile environments such as a wireless networks. It is also suited for use in multimedia environments such as in the World Wide Web. Because of the growth in mobile commerce it is very important to determine whether an identity is permitted to access a resource or not. RBAC can be used for this purpose. However, the standard RBAC model must be changed in according to the environment it is supposed to be integrated in.

All of the more traditional access control methods have required manual configuration. This has not resulted in less finely grained permissions, nor has it reduced the amount of workload needed for defining such permissions. "At present, most access to network information is not controlled on a fine-grained basis. There is a very real danger that by accommodating all of the needs for fine-grained access management into the basic management mechanisms we will produce a system that is too complex and costly to see widespread implementation anytime soon [2]", which is used as a good argument for using RBAC in such implantations.

To fully implement RBAC we mean it is important to provide an administrative user interface, which is needed for implementing the administrative functions and operations defined in the RBAC functional specification.

We mean that RBAC is the best access control model that exists today, which offers a broadly set of features, increases the security and in addition easies the administration tasks. RBAC can also be used together with other access control models and concepts. Adapted RBAC models may also be implanted in future location aware applications.

7.3 Our Prototype application

In according to our thesis definition, we should develop an application prototype. This prototype should implement some concepts from the RBAC model. In addition, the application should work in a mobile environment and make use of a technology for location estimation as well as implementing teleservices.

To cope with this problem definition, we started to design an architecture, as shown in Figure 16, of the system. The system should work in a mobile environment, meaning that a user can move freely and his or her access privileges are being dynamically updated, which is an important element of transparent security. In the beginning of the

development process we thought of implementing teleservices. However, this was not an easy task, and therefore with the introduction of ActorFrame we decided to implement the Servlet server into this framework for using teleservices. This is an expansion of our prototype. In according to the architectural design, the actual technology used and the programming challenges, several issues occurred which had to be dealt with.

Firstly, we had to decide the programming language. Then how to discover mobile users at different locations and how this information could be sent back and forth between a server and the mobile devices, for further processing. Secondly, we had to think of the mobile devices, their programming support and their constraints. Then we had to decide how to store user information and their access privileges at the different locations. We also had to think of how to implement teleservices and support for RBAC components.

We decided to use Java as the programming language. The reason for this is firstly that our education has focused on the new and growing Java technology. Secondly, Java offers a wide variety of advanced technologies ranging from heavy server side APIs to light horizontally APIs suited for mobile environments. The Java development platform can easily be downloaded from the Internet. Sun and the mobile manufactures also offer required tools and the necessary documentation. Java does not require the latest technologies or computers, thus making it ideal for student projects.

Since Java uses a virtual machine, delays can occur, for example when starting a program on a mobile phone. Since the developer applications supports device simulation, even the mobile devices are not needed in the development phase. However, for a location aware system which we have developed, some sort of Java supported device is required.

As described in chapter 4.2, there are several sensing technologies that we could use for discovering mobile users. Since there are different technologies suited for outdoor localization, such as GPS, and indoor systems, such as Bluetooth, Wi-Fi, and RF technology, we decided to first make a system that works indoors. Then, if we got the time we could extend the system to also include outdoor positioning. However, we spent a lot of time on the development process and to get the system up and running, so outdoor positioning was leaved out.

For the indoor system, we decided to use Bluetooth devices connected to a computer, running Linux, as Access Points. The decision was based on that the devices are relatively cheap and that they are easy to set up. Also more and more mobile phones and PDAs come with Bluetooth integrated. With Bluetooth it is possible to define smaller or bigger coverage areas, since it comes in three different classes. Class1 supports 10cm, class2 supports 10m and class3 supports 100m.

The Bluetooth devices we use in the Access Points and the tool we use for scanning devices causes delays in the system. We have tested different types of Bluetooth devices, and we found out that the scanning procedure had the same delay with all the tested Bluetooth devices. Thus, it seems that it is the scanning program, “hcitool”, which causes delays. One scan with the hcitool may take a long time (10 seconds), a short time (0.5

seconds) or anything in between. This has been observed by watching the scanning process at the APs. Because of this, the scanning of mobile devices within range is slow. This causes delays in the updating of devices that have recently entered one of the APs coverage areas or just leaved a coverage area. This does again affect the updating procedure that sends information with available user data to the server, thus causing slowness and an unreliable system. To cope with this delay problem, we must stay within a coverage area in a certain period of time before we are discovered by the APs. Only then, we can get the assigned permissions on a special location. However, the MIDP can be started before the device is discovered, since this data uses GPRS and sends that information through the internet. When moving out of a covered area, we also have a delay to minimize the chance for dropout. This delay causes the user to be able to access his or hers rights even outside the covered area for a few seconds.

Bluetooth may be too strong to cover small areas. They can propagate through thin walls, and in our case it is actually possible to get access by standing in the hall or outside the door. Thus, it is possible to get access to permissions that should be denied outside the coverage area a user has leaved.

We tested both a class3 and a class2 device. The class3 device covered a too large area in the teleservices lab, enabling us to log in by standing outside the door. Thus, we swapped to class 2. Then we expected coverage only inside the Teleservices lab. However, this was not the case, we could still login by standing outside the door, but the coverage area was still reduced. A solution to hinder this may be to limit the coverage area by using some sort of signal dampening material like aluminium foil. It is also possible to use more Bluetooth APs so triangulation can be used. Then it is possible to measure the signal strength for finding out a user's specific position.

Another disadvantage with Bluetooth is that it uses almost the same frequency range as the Wi-Fi standard (2.5 GHz), and therefore interference could arise. However, since Bluetooth devices changes frequency 1600 times a second, interference problems are a rare problem. We can verify this, since we have placed the Access Points (AP) in places where there is wireless network coverage, more specific 802.11b, and no problems seemed to occur.

As shown in Figure 26, the mobile users are detected by APs. These APs can be equipped with other sensing technologies than Bluetooth. When a mobile device is detected, the APs send the information to the AP server through the fixed network. We decided to use UDP for sending the information over the network, because this is a connection-less protocol. Thus, reducing the traffic on the network and reducing delays. Between the mobile device and the Servlet server we decided to use HTTP, and by sending URLs. This is an easy way to send information, and is easily supported by using a servlet on the server side and GPRS from the mobile device. However, this forces the server to listen at specified intervals, for mobile requests. It has to know when to listen. If the mobile device sends a request without the server listening, the request is not discovered and no response is sent. Therefore this synchronous messaging should be changed to asynchronous messaging by using sockets which also is supported.

Since we decided to use Java as the programming platform, the mobile devices had to support Java. Therefore we used the SE P900 mobile phone for testing purposes. This phone has several advantages. It is a phone based on the Symbian operative system, which is suited for mobile devices. This operating system has support for MIDP, but also a more advanced Java version called Personal Java [42] is supported. It has also support for the C programming language which also offers low level access to the phones specific features, such as the contact list. In contrast to other phones SE P900 is equipped with a large touch-screen which users may find more accessible than a tiny keyboard. Because of the mobile constraints, mobile devices implement a small fraction of possibilities in contrast to the regular java standard. It is important to make sure that no labour intensive tasks are implemented on the phone. This is due to the lack of system performance, only 156 MHz in the P900.

We did not want to store any information on the mobile device. This was because of the available storage, but with the possibility for increasing storage, it could as well be implemented on the phone. However, not all phones have support for additional storage. There could also be a security risk, since a mobile device can easily be stolen or tapped for information through security holes.

When we developed the user interface for the mobile, we had to make it as simple and user friendly as possible. However, not many user interface components are supported in the currently version of MIDP 1.0 and 2.0. Either can components of different types be nested in the same form. A form is shown as one screen on the mobile device. This has resulted in poorer user interface and forced us to use more forms, or screen views, than desirable, making the application bigger and the user interface weaker. As a result the user must make more choices than desirable, and it may exceed a user's patience. However, with further development of the Java support for mobile devices, a better user interface could be developed. It is desirable that the user should do as few selections as possible, and letting the system take care of the rest.

Because of the mobile constraints we decided to store all the user information on the server. We also decided to implement the core RBAC first, with the use of a database and functions in the application. Then more additional functionality could be added at a later point in time.

Our prototype application implements access control by checking a user's identity, his activated roles, the available permissions and the user's location. In addition authentication is implemented by asking for login name and password. The MAC address is also used to limit a user from logging on from other devices at the same time. However, confidentiality is not well implemented since all the messages that are sent through the URLs are sent in plain text. This is a security gap, and enables a hacker to see that information. It is then possible to exploit our system. He can fake a login from an Internet browser by sending that information in an URL. This is possible, and tested, but the attacker must make use of a mobile device within coverage, and he must know the IP address of the server and the exactly command syntaxes that we use in our application.

However, if the attacker knows this information, he can perform the operations that are supported. Then, he could get access to send free SMS messages, which is not desirable. Therefore, the message passing should be encrypted, but it will increase the overload that has to be sent, thus reducing the available bandwidth. By implementing cryptography security threats like this would be much smaller, or even eliminated. This would ensure integrity.

When concerning the Bluetooth technology there has recently been discovered security holes that allow an attacker to get access to users address book or even the possibility to send SMS messages from the victim's mobile phone [43]. This has not yet been a big security threat, however when these threats are becoming more common, attacks will probably increase. Thus, an attacker may exploit sensitive information, like credit card information. To seal these security holes, it may be necessary to install a firewall or security updates from the manufactures.

Transparent security is implemented by letting the system dynamically change the permissions for a user based on the location in which the user is situated. By letting permissions be defined for different location domains, less administration is required.

Our system is only tested in a scenario with few users. Therefore it may arise problems when testing it in a multiple user scenario. Just think of the way we transfer messages. It was desirable to use threads [47] for handling multiple users. Then one thread would be used to listen for user requests, while the actual user request could be handled in a separate thread. In this way our server could handle several requests simultaneously. However, in our present version we use the same thread for listening and for performing the request. Even though a buffer is used for incoming requests, it could cause huge delays if multiple users were accessing the system.

7.3.1 RBAC support and expansions

Role Based Access Control is meant for medium or large information systems with thousands of users, or more [13]. In such systems RBAC easies the administration tasks greatly. Our application has only been tested in an environment with few users. This is due to the lack of users and amount of available mobile devices that is used to access the system. Because of this, the RBAC advantage does not appear that clearly. However, the system could be tested in a larger user scenario.

We have used the RBAC reference model [13] when we were to implement RBAC in our application. The model defines a collection of basic RBAC components and features that must be implemented in all RBAC systems. Optional components that may be implemented are role hierarchies, static constraints or dynamic constraints. In our model we have only implemented Core RBAC, or $RBAC_0$ as shown in Figure 2. The core elements represent users, roles, objects, operations and permissions. It also defines sessions, which is used to activate a set of roles for a current user. Our implementation of core RBAC is shown in Figure 25. It corresponds to Figure 3 except for the relation between sessions and roles. Since we focused on getting the basic elements on place first we leaved this out since it would be a heavy task to implement.

As we see of Figure 25 we have implemented users, roles, permissions and location. Permissions consist of operations and objects. There also exist UA and PA relations. This is done through the database through the tables RoleActivation, a relation between users and roles, and PermissionActivation, a relation between roles and permissions. Then it is possible to assign users to one or more role, or to assign roles to one or more users. It is also possible to assign permissions to one or more roles and roles assigned to one or more permissions. This way of controlling access to resources provides great flexibility and granularity of assignment of permissions to roles and users to roles. This enables an administrator to assign just the permissions a user needs to carry out a particular job. Thus, the principle of least privilege is implemented.

We have also a relation form users to a sessions. We activate a session for every user. This session object encapsulates information about the current user and it is used for authorization of user requests. However, since we do not have a relation between sessions and roles it is not possible for a user to activate roles in a session, nor to implement SSD and DSD relations.

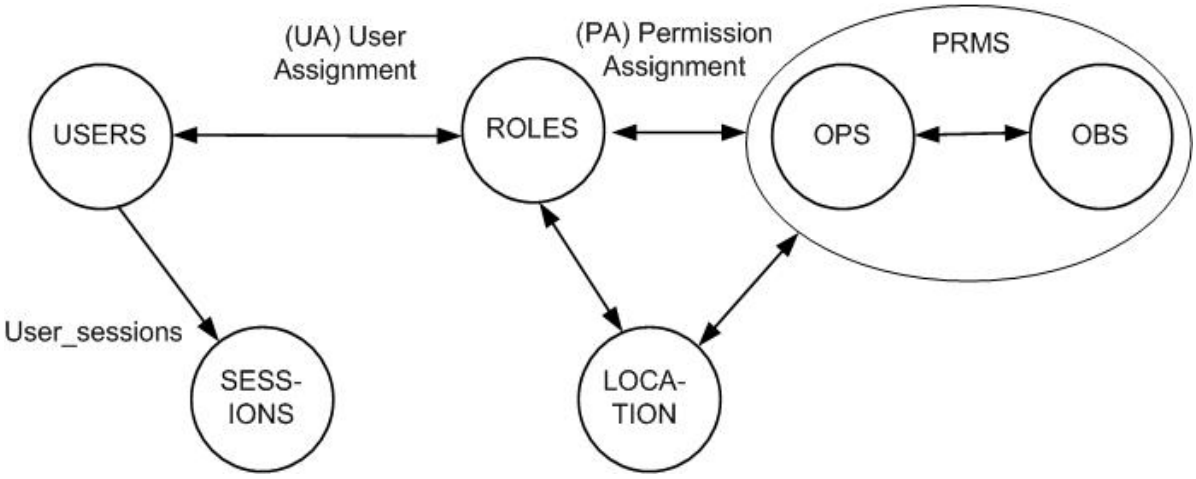

Figure 25 - Core RBAC support in our application

Since our system is designed to work in a wireless environment, where users can get access to resources based on location, we have extended the context with location as indicated in Figure 25. We have used concepts from the SRBAC model [14]. It is then possible for users to activate different roles at different locations. In our application the assignment of roles is done dynamically, and the users can not choose which roles to activate or deactivate, because of the missing relation between sessions and roles in Figure 25. Instead it is the administrator that specifies which roles that a user activates at a current location. This feature makes it possible to specify spatial constraints on the amount of roles assigned to a user. These roles do again have permission assigned to them which will be listed. In accordance to the SRBAC model, it would be more flexible to specify which permissions that can be activated at a given location than specifying roles. This would reduce the number of roles that are needed to be specified in the system. Thus, security administration would be simplified [14].

The specification [13] defines both System specific functions and administrative specific functions. The latter is dependent on an administrative interface for performing its functions. These are functions for creating, deleting and maintaining RBAC elements and to create and delete user role assignments. Since we have not developed an administrative interface these administrative specific functions are not easily to manage. In our case an administrator have to add, delete or change entries directly in the database which is a cumbersome task. However, such an interface is not developed in this version of the application since we do not need it to demonstrate our idea behind the application.

As mentioned in chapter 3, the system level functions defines features for creating user sessions for enabling role activation or deactivation, enforcement of constraints on role activation, and calculating for an access decision. These functions are implemented in our application. However, user sessions are not the same as described in RBAC. We have not either implemented constraints on role activation. Instead we use a constraint on the permissions assigned to the roles. A user may only perform ono operation at a time. This may also hinder conflict of interest situations on operations.

7.3.2 Integration into ActorFrame

With the integration of our Access Point server into ActorFrame it is now possible to use teleservices. In addition to use Bluetooth for localisation it is now possible to use GSM localisation to verify that a user is at the location given by the AP clients.

In addition to localization techniques, a SMS service is implemented. This enables a user with the correct access privileges to send a SMS. It could also be possible to use call forwarding. For example, consider an employee leaving his office. When leaving his office he brings with him his mobile device. When the system discovers the change in location, his office phone is automatically forwarded to his mobile phone. Then a message could be sent to his mobile phone telling him that his office phone is forwarded. When he is back at his office, the call forwarding could be automatically turned off. In this way the employee will always be available.

ActorFrame makes use of Actors. These Actors can have different roles, depending on the behaviour. This could be mapped to a user scenario, where an Actor could represent a user. Then users could be assigned different roles depending on their current tasks, or location.

The new AP server is not depended on the MySQL database that we used in the first application, thus storing all the information in the same application. The new integration gives advantages. By using state machines and asynchronous messages it can handle far more user requests than our version of the server. It also has the ability to only update those devices that are new, in contrast to our server which updates even older devices. There would also be a difficult task to make an administrative user interface for this implementation.

There are not implemented more support for RBAC components. In fact, since it is not depended on the database, the RBAC structure is now more diffused since it must be

implemented somewhere in the framework. There has not been focused on using RBAC components in this version.

ActorFrame is a complex framework and advanced help is needed when implementing an application into it. In our case, we got help form Geir Melby which has designed the system. The system is also large, and thus needs computers with large capacity, especially with a serious amount of users. It is also more robust and a more serious application than our version. However, the purpose of this project was to make a prototype that could be used to give users different privileges based on location.

7.3.3 Future work

Figure 26 shows an overview of our architecture with the use of Bluetooth for discovering mobile devices. However, it also shows other possible technologies. As we see there are defined three zones, or coverage areas. In zone 1 and 2, APs with Bluetooth are used. But zone 3 shows future possibility for detection. GPS can be used outdoor, and GSM positioning can be used both inside an outside. GSM positioning can be used together with Bluetooth detection to double check if a current user is situated at a current location. This figure shows that a user can move freely, but it is not until he is within coverage that he can perform some sort of operations. For example, when using this system in an organization it is possible to define a guest role, which is appropriate for visitors. The servers and the database are protected by a firewall for preventing attacks from the internet.

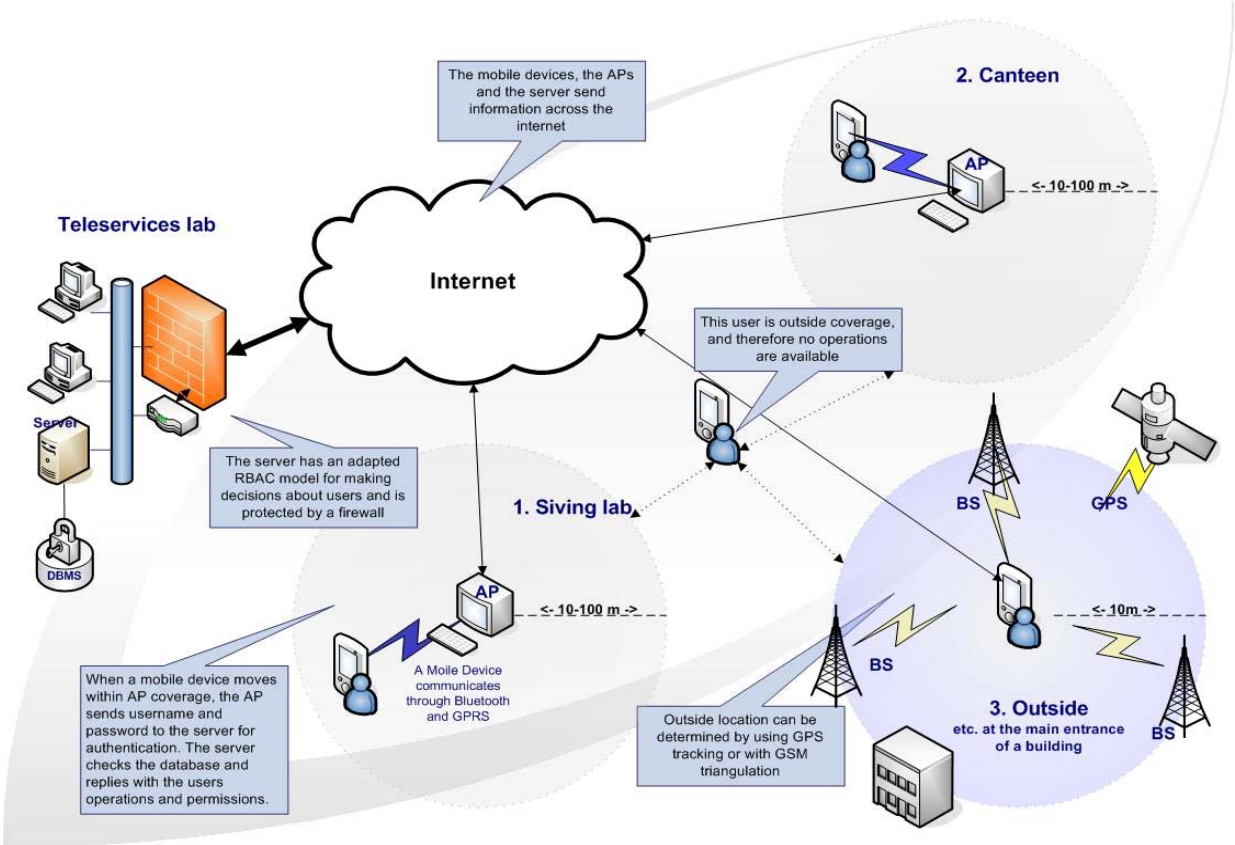

Figure 26 – System Architecture and possible discovery scenarios

The currently implementation of our AP does nothing else than to scan for devices and send that information to the server. Even if no devices are within range it sends information to the server, which in turn updates the database. However, more intelligent functions could be implemented in AP. This would make the AP more robust and lower the amount of information, thus the amount of packets, sent to the server. This would also make it more scalable when implementing it into bigger environments with many Access Points. If the existing AP program was used in such large environments, it would consume a lot of bandwidth and the change for taking down the server is present, due to the amount of packets it must process. Therefore, a future version should only send updated information about the devices.

Our management of the database is difficult, since we have not been able to finish an admin interface, however we started to develop it (enclosed CD). Because of this an administrator must have full control of the various parameters in our prototype. The job will be work intensive when the number of users increases, thus increases the information in the database. A further improvement would be to make an interface that eases the administration of the various aspects, such as assigning roles to users and permissions to roles. Then also review functions and administrative functions could be implemented.

In our application one has to define roles for each zone in a system. When the amount of users grows, the administration tasks will be a heavy burden. Therefore, it is wisely to adapt the idea from the SRBAC model, where permissions are used instead of roles. Then the administration would be further reduced.

Our application does not have a relation between sessions and roles. In a future application this should be implemented. Then users and roles are part of a session, and both SSD and DSD relations could be implemented. Support for more advanced RBAC components, such as role hierarchies could also be implemented.

Before we started to program we thought of security in the way of encryption. Because we use HTTP to transfer messages in plaint text between the mobile device and the server, an attacker could just send a URL from any computer and thus confuse the system. If an attacker knows the password and the username it is possible to fake the MAC address, and thus it is possible to get unauthorized access to the system. However, this is only possible when the exploited device is within coverage.

Encryption can be implemented through the bouncy castle crypto package [29]. However, this is a time consuming task, therefore it is not implemented in this version of our prototype.

7.4 Summary

This discussion has found that RBAC is an effective access control method for use in medium and large environments where large scale management is needed. It reduces the administration tasks and by using roles and support for hierarchies it can map an organizations internal job hierarchy. We have also discussed prototype and its support for RBAC. The implementation into ActorFrame has proven to give us an effective way for

implementing teleservices. Since our application is a prototype future improvements are proposed and can be used in future research projects.

8 Conclusion

This thesis has focused on a study in RBAC and issues relevant to an implementation in a mobile environment. An architecture and a prototype application has been developed where RBAC has been extended to handle access control based on the location in which the user is situated. By also implementing our application into ActorFrame, we have showed that it is possible to develop an application that has support for teleservices.

We have showed that RBAC is an effective access control mechanism for handling access to data and resources in large information systems as well as to reduce the administration tasks. The support for role hierarchies and inheritance of both permissions and prohibitions makes RBAC ideal for mapping an organizations authority and responsibility.

There exist various RBAC models, each with a different terminology. This confusion has made application developers to adapt their own versions. However, the proposed NIST standard will try to address these problems, by defining a collection of basic RBAC components and features in addition to a precise and persistent language.

The main conclusion is that RBAC can be adapted for use in mobile environments. The SRBAC model is an example of an adapted RBAC model that can be used in such environments. We have also shown that it is possible to use location aware technology, such as Bluetooth and GSM localization, for resolving a user's actual position. By implementing the advanced functionality on a fixed network structure, it is possible to develop light applications that are within the constraints of the mobile devices, like performance and memory limitations.

In addition, this work forms a basis for further work in this area. The goal is to find more precise ways for determining location of users and to make a better application environment, which can be used in commercial wireless location aware systems. This project has also been a part of the AVANTEL research project, which aims to increase the expertise in heterogeneous services and rapid development of services.

9 References

[1] T. Guerin and R. Lord, "RBAC identity management", portalsmag.com, August 19, 2003, viewed: Mars 16, 2004

[2] Camelot Information Technologies Ltd., "Differentiating Between Access Control Terms," 2001, http://www.secinf.net/uplarticle/2/Access_Control_WP.pdf, Viewed: February 11, 2004

[3] E. Ferari and D. Ferraiolo, in *Proceedings of the Eight ACM Symposium on Access Control Models and Technologies*, June 2-3, 2003, Italy, ISBN: 1-58113-681-1

[4] V. Ungureanu, F. Vesuna, N. H. Minsky, "A Policy-Based Access Control Mechanism for the corporate web", in *Annual Computer Security Applications Conference*, 2000, pp. 150-158

[5] T. Dimitrakos, I. Djordjevic, B. Matthews, J. Bicarregui, C. Phillips, "Policy-Driven Access Control over a Distributed Firewall Architecture", In *3rd International Workshop on Policies for Distributed Systems and Networks*, June 05-07, 2002, pp. 228-231

[6] J. Wiggins, "Cisco Context Based Access Control (CBAC)", January, 2002, http://www.giac.org/practical/Jim_Wiggins_GSEC.doc Viewed: February 18, 2004

[7] B. Wijnen, R. Presuhn, K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3415, December 2002

[8] M. Hazas, J. Scott, J. Krumm, "Location-Aware Computing Comes of Age", *IEEE Computer Society*, Vol. 37, pp. 95-97, February 2004

[9] T. Nadeem et al., "Implementation of a Scalable Context-Aware Computing System", Department of Computer Science University of Maryland, USA, pp. 364 – 374, October 2003

[10] M. Hazas, J. Scott and J. Krumm, in *Proceedings of the 2003 Workshop on Location-Aware Computing*, Seattle, Washington, USA, October 2003

[11] The Cricket Indoor Location System, web page, http://nms.lcs.mit.edu/projects/cricket/ Viewed: February 16, 2004

[12] K. Bauknecht, "INFORMATION SECURITY", http://www.ifi.unizh.ch/ikm/sec02/autho.pdf, 2000, Viewed: February 16, 2004

[13] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuh and R. chandramouli, "Proposed NIST Standard for Role-Based Access Control"

[14] F. Hansen and V. Oleshchuk,"Spatial Role-Based Access Control Model for Mobile Systems", Agder University College, Department of Information and Communication Technology

[15] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein and Charles E. Youman, "Role-Based Access Control Models", IEEE Computer, Volume 29, Number 2, pp. 38-47, February 1996

[16] Virgil D. Gligor, S I. Gavrila, David Ferraiolo, "On the Formal Foundations of Separation-of-Duty Policies and their Composition", in *Proceedings of IEEE Symposium on Security and Privacy*, May 1998

[17] F. Cuppens and A. Miege, "Modelling Context in the Or-BAC Model", in *Annual Computer Security Applications Conference*, 2003

[18] M. Satyanarayanan, "Fundamental Challenges in Mobile Computing", School of Computer Science, Carnegie Mellon University, http://www-2.cs.cmu.edu/afs/cs/project/coda/Web/docdir/podc95.pdf Viewed: January 26, 2004

[19] M. Bishop, Addison-Wesley, *Computer Security: Art and Science*, University of California - Davis, December 2, 2002, ISBN: 0-201-44099-7

[20] "Access Control: Policies, Models, and Mechanisms," Tutorial Lectures given during the Foundations of Security Analysis and Design, Bertinoro, Italy, 2001, ISBN: 3-540-42896-8

[21] Aberdeen Group Inc., "Security, Privacy, and Risk Management", July 2003, http://www3.ca.com/Files/IndustryAnalystReports/Security_Top_10.pdf, Viewed: April 19, 2004

[22] Cisco Systems, *Cisco Security Policy Engine Administration Server User Interface*, http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/bacbba/bacbba25/usguid/apaspe.pdf Viewed: April 19, 2004

[23] C. Ramaswamy and R. Sandhu, "Role-Based Access Control Features in Commercial Database Management Systems", Computer Security Division and Software Eng. Dept.,

http://csrc.nist.gov/rbac/RBAC_DBMS_Comparison.pdf
Viewed: April 19, 2004

[24] John F. Barkley, D. Richard Kuhn, Lynne S. Rosenthal, Mark W. Skall, and Anthony V. Cincotta, "Role-Based Access Control for the Web", National Institute of Standards and Technology Gaithersburg, Maryland, http://www.itl.nist.gov/div897/staff/barkley/cals-paper/cals-paper.html
Viewed: April 22, 2004

[25] PATS - Program for Advanced Telecom Services, http://www.item.ntnu.no/avantel
Viewed: Mars 4, 2004

[26] Cisco Systems, *Cisco Security Policy Engine Administration Server User Interface*, http://www.cisco.com/en/US/products/sw/netmgtsw/ps5117/products_user_guide_chapter09186a008018705f.html
Viewed: April 19, 2004

[27] Geir Melby, Master thesis, "Using J2EE Technologies for Implementation of ActorFrame Based UML 2.0 Models", Agder University College, Grimstad, Norway, May 2004

[28] WC3 - The World Wide Web Consortium, "W3C Recommendations Reduce World Wide Wait", http://www.w3.org/Protocols/NL-PerfNote.html
Viewed: Mars 24, 2004

[29] Legion of the Bouncy Castle, The Bouncy castle crypto Package, http://www.bouncycastle.org/documentation.html
Viewed: February 12, 2004

[30] F. Boger Engelhardtsen and T. Gagnes, " Using Jini and JavaSpaces with Ericsson NorARC's technologies for service creation", Masters Thesis, Agder University College, Norway, May 2002

[31] A.Hawick, H.A.James,"Middelware for Context Sensitive Mobile Applications", Computer Science Division, School of Informatics, University of Wales

[32] JBOSS, an Application Server, http://www.jboss.org
Viewed: April 7, 2004

[33] Ahmed El-Rabbany, "Introduction to GPS: The Global Positioning System". Artech House, Inc, ISBN 1-58053-183-0, 2002

[34] "The Java Technology, Java 2 Platform, Enterprise Edition (J2EE)", The source for Developers, A Sun Developer Network Site,
http://java.sun.com/j2ee/
Viewed: April 7, 2004

[35] "The Java Technology, Java 2 Platform, Standard Edition (J2SE)", The source for Developers, A Sun Developer Network Site,
http://java.sun.com/j2se/
Viewed: April 7, 2004

[36] Online Manual for "Hcitool",
http://www.linuxforum.com/man/hcitool.1.php
Viewed: February 5, 2004

[37] Klaus Finkenzeller, "RFID Handbook, Fundamentals and Applications in Contactless Smart Cards and Identification, Second Edition" , John Wiley and Sons Ltd., ISBN 0-470-84402-7, 2003

[38] "J2ME, Mobile Information Device Profile (MIDP)", The source for Developers, A Sun Developer Network Site,
http://java.sun.com/products/midp/index.jsp
Viewed: January 20, 2004

[39] "J2ME, Connected Limited Device Configuration (CLDC)", The source for Developers, A Sun Developer Network Site,
http://java.sun.com/products/cldc/
Viewed: January 20, 2004

[40] Radionor Communications, "Technology", the Cordis RadioEye,
http://www.radionor.no/
Viewed: Mai 25, 2004

[41] The Symbian Operating System,
http://www.symbian.com/
Viewed: Mai 5, 2004

[42] "J2ME, PersonalJava", The source for Developers, A Sun Developer Network Site,
http://java.sun.com/products/personaljava/
Viewed: January 20, 2004

[43] "Expert: Gaps still pain Bluetooth security", News.com,
http://news.com.com/Expert%3A+Gaps+still+pain+Bluetooth+security/2100-1009_3-5197200.html
Viewed: April 23, 2004

[44] MySQL Home Page,"MySQL The World's Most Popular Open Source Database", http://www.mysql.com/
Viewed: January 15, 2004

[45] National Computer Security Center, "*A guide to understanding Discretionary Access Control in trusted Systems*" , September 30, 1987, http://www.radium.ncsc.mil/tpep/library/rainbow/NCSC-TG-003.html
Viewed: February 16, 2004

[46] "J2EE, JDBC Technology", source for Developers, A Sun Developer Network Site, http://java.sun.com/products/jdbc/
Viewed: April 23, 2004

[47] "Threads: Doing Two or More Tasks At Once", Lesson, http://java.sun.com/docs/books/tutorial/essential/threads/
Viewed: April 23, 2004

[48] UML, Unified Modelling Language, "UML Resource Page" http://www.uml.org/
Viewed: April 23, 2004

[49] The Open Group, "History and Timeline, UNIX Past" http://www.unix-systems.org/what_is_unix/history_timeline.html
Viewed: February 12, 2004

[50] NIST, "National Institute of Standards and Technology", http://www.nist.gov/
Viewed: February 12, 2004