INTERNATIONAL
HELLENIC
UNIVERSITY

Enterprise Monitoring and Security Compliance in Mobile Devices

Papadopoulos Konstantinos

SID: 3301110009

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Information and Communication Systems

OCTOBER 2012

THESSALONIKI – GREECE

Enterprise Monitoring and Security Compliance in Mobile Devices

Papadopoulos Konstantinos

SID: 3301110009

Supervisor:	Prof. Vasilis Katos
Supervising Committee Members:	Assoc. Prof. Name Surname
	Assist. Prof. Name Surname

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Information and Communication Systems

OCTOBER 2012

THESSALONIKI – GREECE

Abstract

One of the most important assets that modern companies do have is information. In order to make sure that it is secure from threats, managers need to take decisive steps of great importance. Those steps cover all levels of businesses, from top management to operational level.

This thesis focuses on studying certain steps that are being considered key points in implementing a full security plan in business environment. The analysis is focused on Security policies, Access Control models and mobile devices that interact with corporate information. Even though the first two parameters are not new to information security science, they are deeply affected by modern technologies and trends; more specifically smartphones. As a result smart devices (iOS) are also examined along with the threats that arise by using them. Finally, a prototype management system is proposed that allows companies follow an Access Control model (Role-Based AC) and easily implement and enforce a security policy using mobile devices and modern tools.

This thesis could have not been completed and provide a high quality system without the significant assistance and contribution from Mr. George Mavroudis & Mr. Sotiris Karagiannis from Space Hellas S.A. Furthermore the guidance and advice of the supervisor, Dr. Vasilis Katos was of great importance. Without their help, this project would have not been completed.

Papadopoulos Konstantinos

Friday, October 26, 2012

1 Introduction

The subject of this thesis is focused on Information Security in relation to mobile devices. What is Information Security? There is no formal definition explicitly stated in a dictionary or a handbook however interesting comments have been made on its nature.

- ..When systems fail gracefully.
- ..Things that should happen do happen, and things that shouldn't happen don't. (Sasse, et al., 2007)

They sound a little unconvincing when trying to persuade someone about the importance of Information Security and justify it as an investment however they do reflect true attributes. Information Security in both business and technical level proves to be a rather complicated subject that requires deep analysis and understanding of all parameters in order to be applied successfully. It needs to be mentioned and considered as an axiom that no system is 100% secure.

> *"The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards – and even then I have my doubts"* - Dr. Eugene Spafford, Purdue University.

These words clearly state the true security state of information systems no matter their purpose, technology or context. This thesis will analyze, present and discuss specific issues that are related to Information Security. Furthermore a system prototype has been developed that demonstrates the implementation and enforcement of a security policy in mobile environment along with an access control model. A presentation of the way the report is organized follows.

The first chapter that follows is focused on studying about the theoretical foundation that this report is based on. It is divided in many subsections depending on the subject. Initially, modern enterprise environments and security investments (section 2.1) start the discussion about security. Next a list of threats making special mention to the inside threat follows (section 2.2).

The next section has as main studies security policies (section 2.3). A company that desires to implement certain security policies or control mechanics needs to consider all stakeholders that will be involved and most of all the end-user that will be interacting with them on a regular basic. The importance of security investments will be presented along with numbers and figures based on surveys and questionnaires. Furthermore Security policies will be analyzed; their parameters and attributes are discussed in order to help the reader

familiarize himself with concepts and models. A policy as we will see is not a static document that remains hidden in a drawer. It needs to be assessed on regular time frames, be aligned with both business needs and technological advancements so as to remain valid and useful. Additionally, it will be stated many times that managerial support and dedication is considered mandatory so as to keep employees aware and dedicated to the cause.

However apart from a policy, the implementation of a security strategy requires to follow certain established practices. Access Control models (section 2.4) have been around for many years and represent various technics and theories in order to enforce control mechanics. They can be applied in a wide range of systems and environments depending on the model. Many AC models will be presented, some more important and known than others but all of them have their advantages and disadvantages.

Finally, since the subject is focused on mobile environment, the next section (2.5) is analyzing the relationship between modern enterprises and mobile phones. Finally iPhone has been chosen as a device of study and the security characteristics and functionalities that it offers justify the reason it was picked as a development environment.

During the development phase (chapter 3) a prototype system has been designed and implemented that is aligned with the Role-Based Access Control model. It has been developed satisfying a list of requirements (section 3.2) created in cooperation with Mr. George Mavroudis and Mr. Sotiris Karagiannis from Space Hellas S.A. The system is based on two major components (web platform and mobile application) along with the integration of Amazon Web Services – Simple Storage Service (section 3.3.5). The system allows the manipulation of policies, users and files from web application point of view. The same time the mobile application is responsible to communicate with the server, download and enforce the policy that it is requested to, managing files, logging a wide range of information and performing back-up operations.

2 Literature Review

In order to understand, examine and propose a model or a research methodology that focuses around mobile devices in enterprise environment, first it is mandatory to establish the theoretical foundation, mention past researches and narrow the scope of study. An enterprise is not a single-cell organism that it is easy to monitor and manage. It is comprised of various components, sometimes located globally and the strategic goals are not always clear throughout the organizational hierarchy. It is unavoidable that this blur image is also affecting the Information Security initiatives, goals and actions.

2.1 Modern Enterprise Environment

Every company today in order to function and work in the global economy is establishing some strategic goals and targets. Those targets usually reflect the policy of the company in various fields and furthermore they are defining the way the enterprise is doing business in operational and middle-management level. If those goals are met, then the company can be characterized as successful and usually profitable. However, due to the digital economy and technology, information is becoming one of the most important assets companies have and need to protect. In fact, the National Institute of Standards and Technology has published a series of documents giving guidelines in order to establish security mechanisms and controls (N.I.S.T.). Similar guides and frameworks are designed by both public and private bodies; one of them, SABSA, will be presented later on.

However it is not easy to setup a security mechanism that will be successful. In order to do so, certain principles should be followed (Booker, 2006). First, the senior management of the company should always be aware of the risks that the company is exposed at. Keeping this in mind will provide them the necessary motivation to support security investments. Secondly, security procedures should be established where it is necessary and nowhere else. Creating extra complexity and wasting resources will just mitigate the purpose of security itself. Last, awareness and motivation should always be communicated among company personnel in order to support and accept induced changes. Nevertheless, it is nearly impossible to predict and always monitor the state of a company. This ecosystem is constantly evolving due to its nature. People, organizations themselves and technology are all factors that are both non-static and can lead to unanticipated outcomes; the reasons for these results however differ.

As mentioned earlier, one important factor is that senior management must always be aware about the goals that need to be met implementing security procedures. However this is

not always possible; transferring non-quantitative knowledge requires open mind and willingness to do so. Furthermore management is meant to make decisions regarding the available budget to be spent on security investments. The single most efficient way to do so is literally to put a number in such investments (Booker, 2006) (Choobineh, et al., 2008). If we compare operational with management level regarding security decisions, we observe that at lower level the decisions focus on optimal spending of monetary, human and technological resources. However, deciding strategic goals and funding amount is something totally different.

2.1.1 Quantitating Security Investments

How do we put a number on security investments? How is the amount of Euros that a company is willing to invest in order to prevent future losses determined? According to a recent survey (Computer Security Institute, 2010/2011)(Figure 1), more than 50% of business managers take decisions regarding security budgets based on their return on investments; other metrics are used too.

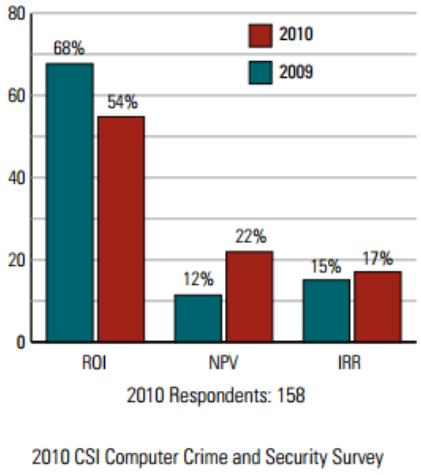

Figure 1 – Percentage of respondents using ROI, NPV and IRR[1]

However it is not easy to always quantify such expenditure because prevented losses and expenses need to be taken under consideration, those are not included in the aforementioned metrics. In order to do such quantification, the time factor plays an important role meaning that the longer a period during which security investments are taking place is, more resources are prevented from being compromised or unavailable. On the other hand the good news is that only 16.5% of the managers state that they do not use any means to evaluate their investments. Even though there is no a definite answer on how to manage financially IT security, the decision making process has been compared to a game theory problem, the

[1] Return On Investment, Net Present Value, Internal Rate of Return

company and the attackers being the players (Cavusoglu, et al., 2004). In this game, the decisions of one side affect the options offered to the opposite. The bottom line is that the actions taken are highly correlated.

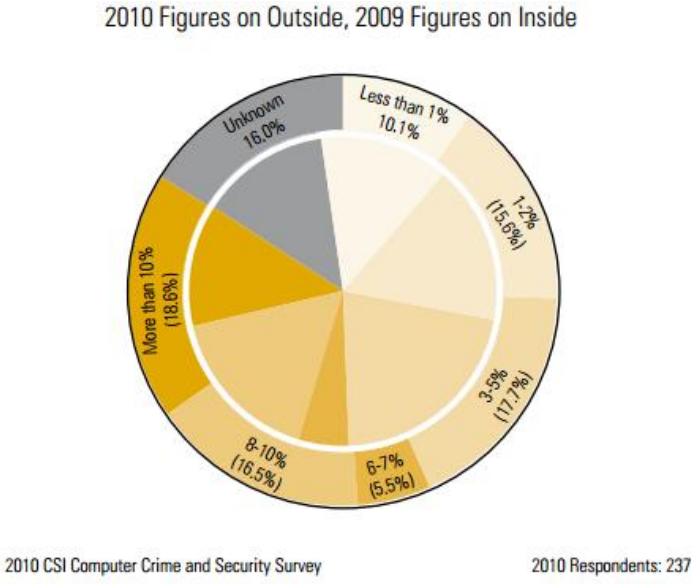

Figure 2 – Percentage of IT budget spent on security

A different opinion states that the enterprise might follow two different ways of decision making depending on its attitude towards risk; either be risk neutral or risk averse. Based on this along with other criteria such as the existence of maximum boundary of accepted loss, management can decide the amount of security mechanisms implementation. An important final note however is that a lower limit on losses exists and most of the time cannot be crossed (Choobineh, et al., 2008).

2.1.2 Implementation of security strategies

After strategic security scope and goals are established based on financial metrics and forecasts, decisions need to be taken in order to design the appropriate model that fits the company. How easy is it to do so? Is there a step-by-step guide on how to implement security architecture?

As a starting point we can say that some guides do exist. One can divide the domain of our problem in two sections; using either holistic or partial models (Shariati, et al., 2011). Even though partial models include security policies that we will analyze later on, at the moment we need to focus on abstractions that span throughout the enterprise. According to the authors, not all Enterprise Information Security Architectures are the optimal solution to be implemented due to the incompatibility between the rapid-changing companies and security models. Those terms are in conflict by definition and the optimum framework needs to be used. Among the most promising choices are SABSA, RISE, Gartner, AGM-based

model and service-oriented EISA. However SABSA (Figure 3) seems the most promising due to its layered architecture (Sherwood, et al., 2009).

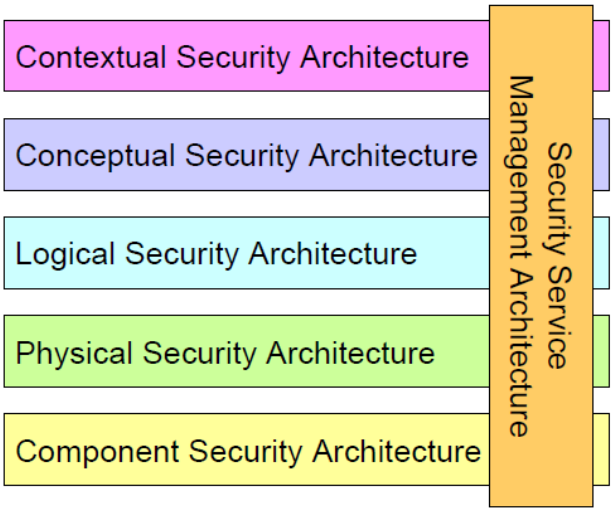

Figure 3 – SABSA framework layered architecture

Key characteristics of this model are that it is aligned to strategic business goals and it is not solely based on technical solutions and implementations, according to the authors.

2.2 Security Issues in Enterprises

Enterprises do not face the same kind of threats and security issues. Depending on their sector and operations type, threats are not perceived the same way by their managers.

2.2.1 Type of threats and Enterprise sectors

Seven types of security threats and four major types of companies are perceived (Yeh, et al., 2007). The last paper is a survey conducted in Taiwan questioning over 100 major enterprises; 1000 questionnaires were distributed but almost 90% of them were not returned due to the sensitive nature of the subject. Based on their answers, managers categorize threats based on the following assets: software, hardware, data, network, physical, personnel and regulation. The four types of industries are: general manufacturing, high-tech, banking & finance and retailing & service. Furthermore, the same survey categorizes threat importance based on managers' perception. It is estimated that the most important aspect is the network on the high-tech and finance sector. Additionally, data and regulation follow as the next most important assets.

2.2.2 The inside threat

With respect to the list of assets mentioned before, personnel have a special role that sometimes is neglected. People are those that initiate attacks taking advantage of existing vulnerabilities. However people can be divided in two basic categories, those that do want to acquire access to corporate data and information and those that already possess that kind of access due to the nature of their job, named "insiders". Contrary to common belief, "insiders" have the means to perform a successful attack to a company much easier than a malicious external entity (Colwill, 2010). The employees in a company do not all have the same privileges and access rights. Even though low-level employees usually are under monitored, senior managers are those that enjoy full trust and access. Most of the time, those are the ones that perform malicious attempts due to the nature of their job role (Sasse, et al., 2007).

Another important factor that is related to performing inside attacks is the so called "Bring your own device – BYOD" trend. "Security's scariest Acronym" (Antonopoulos, 2011) is introducing new risks and problems that include the access of corporate data from a personal and uncontrolled environment. However it should be mentioned that BYOD is also beneficial for employees since it improves their productivity and satisfaction; they are allowed to use devices they personally did choose and they are motivated to keep in good state, updated, they are small powerful computer devices and most of all owners are paying for them – not the company (Flinders, 2010). In order to meet basic security requirements, IT departments must enforce policies and control mechanisms so as to minimize company's' exposure to threats.

A final note is that according to an online survey (Cyber-Ark Software, 2011) that was conducted on IT people, more than 70% of the US participants have used an unauthorized admin password to access sensitive data and nearly 45% of the EU participants accessed information that was not supposed to due to the different work role they possess. This indicates that people are able to go through information that they allowed to. Even though this is done sometimes due to curiosity, the implications cannot be neglected.

In order to mitigate, minimize and finally – if possible – nullify this kind of threats, Chief Information Security Officers need to make better use of internal security policies and procedures. This kind of security threat is too important to be left aside and ignored.

2.3 Security Policies

In order to have a successful company, we need to make sure that we are sufficiently protected and most – if not all – threats have been considered and evaluated. A company's security awareness, education and precaution measures can be compared to the well-known

chain proverb: *it is as strong as its weakest link*. And one of the most important links is the corporate-wide implementation of a security policy that clearly reflects the business's strategic orientation. Furthermore, information security has also been called "business security" (von Solms, et al., 2005). This is due to the major role that information plays as an asset in creating and maintaining a competitive advantage; furthermore it emphasizes the importance of security breach events that target such information. Security policy is given the role of establishing the framework under which all company's departments and employees should work.

2.3.1 Overview

What is a security policy? How can we define it? As a start we should mention that a policy is a document, a well written document that explicitly states and refers to everything that is affected by it. But it is not a unique one across the company. Depending on the corporate level it refers to (senior, middle and operational level) it applies to different resources (human, money, IT equipment, knowledge etc.). Its content is not directly related to technology but to general guidelines that must be followed at all times. Of course these refer to IT issues too but not to specific technologies.

The definition that is provided by S.A.N.S. is: "*A policy is typically a document that outlines specific requirements or rules that must be met. In the information/network security realm, policies are usually point-specific, covering a single area"*. Another interesting explanation of policy is given by (Whitman, et al., 2011) that compares it with a *law established inside a company that also includes penalties for violations and appeal processes*.

2.3.2 Designing a policy

In order to create a policy that is successful – it must be successful in order to justify itself as an investment – business attributes need to be taken under consideration. Policies' single most important aspect, from business point of view, that needs to be examined is that it needs to be aligned to business's strategic goals, understand the business model and do not restrict business functions. Furthermore a security policy implementation process should always – as any security related decision – has the senior management explicit and clearly stated support as well as commitment. The last key aspect is that the purpose, scope and methodology that the policy proposes must be clearly defined (Höne, et al., 2002b).

Furthermore, a security policy should include four different entities: people, policies, procedures and technology (Slay, et al., 2006). Additionally, it has been stated the creation of a security policy is difficult since one must meet the following standards: be law abiding,

have legal support and, last but not least, be properly distributed and accepted (Whitman, et al., 2011). The authors of the last book support and propose a general structure that allows us to study and present various types of policies and key characteristics; this structure was chosen to be presented and analyzed.

While trying to gather information about the nature of guidelines and procedures that this sort of documents include, it is not explicitly stated what a policy should mention (explicitly refers to a step-by-step guide). There is no golden rule on how to write a policy. Even international standards exist that propose a framework but the actual answer should originate from the company that has decided to implement the policy (Höne, et al., 2002b). In general and based on the source that the reader is referring to, different opinions and organizational proposals can be mentioned. In this thesis, the opinions mentioned by (Whitman, et al., 2011) will be mostly presented and analyzed as they were found to be more organized and well-documented than any other similar report.

2.3.3 Policy structure

According to the National Institute of Standards and Technology, policies are categorized in three major groups (Swanson, et al., 1996):

- Program policy
- Issue-specific and
- System specific

A more comprehensive taxonomy of this subject addresses the same structure using a slightly different naming convention (Whitman, et al., 2011):

- Enterprise Information Security Policy (EISP)
- Issue-Specific Security Policy (ISSP)
- System-Specific Policy (SysSP)

The second categorization will be analyzed in the next section. During this analysis, the basic differences between the three types are presented along with the list of components that they are comprised of. We need to clarify that a policy is not a stand-alone entity. It is a part of a layered model that includes additional notions that co-operate with policies; those are standards and guidelines. In Figure 4 there is a visualization of the hierarchy. The colors in each layer represent the importance (red being the most important) that the respective documents have in establishing a reliable security framework in an enterprise. It is clear that policies are placed on top of the pyramid and act the head of the model. Depending on "head's" philosophy, the rest of the layers are designed and implemented accordingly. As we move down the pyramid, entities are becoming less flexible in terms strictness, updated more often and affect smaller parts of a company. This means that the content of a procedure or a

guideline is mandated by the upper levels' directives and statements. Since lower levels are more specific about technology and have a narrow scope, a slight change in the technology or the current business model requires an update and modification of the document. This does not apply to policies because they are not bound by specific details and implementations.

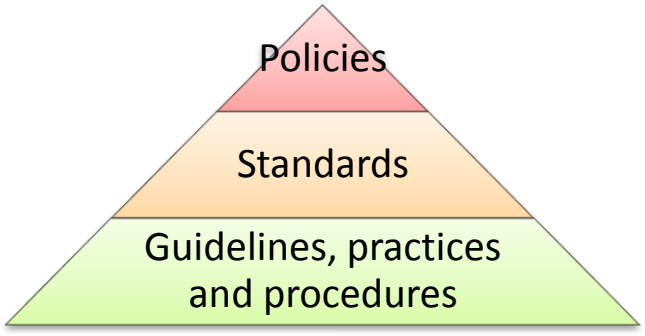

Figure 4 – Policy structure pyramid (Whitman, et al., 2011)

2.3.4 Policy Analysis

Based on the latest presentations, a deeper examination of the types of policies follows.

Enterprise Information Security Policy (EISP)

An EISP is enterprise-wide document that establishes general security guidelines and goals. It is aligned with the business's goals and strategic mission. This type of document describes briefly the overall security purposes within a company that the Chief Information Security Officer is responsible to implement. Policies at this level might not affect directly all departments or employees but certainly the mentality is transferred everywhere in the corporation.

It is comprised – usually but it is not mandatory - of the following sections:

- Purpose
- Definitions and terms used
- Reasons for implementing the existing policy
- Roles and responsibilities
- Connection with other relevant documents

At the appendix (section 6.1) we can see an example of the way an EISP is structured (Georgia Technology Authority, 2002). It covers a wide range of sectors, issues, assets and threats. A different example is presented at 6.2. This is a more compact document that presents many good practices that an EISP and a policy in general should follow (Kennesaw State University, GA, USA, 2006). A few elements that were not mentioned before but are

included in the last document are dates. Dates are very important so as employees and affected personnel be aware about the validity of the document. Date importance will be referred again later.

Issue-Specific Security Policy (ISSP)

Some generic guidelines can be formed about the subjects that are important to be covered by an ISSP. For example we can examine the process of keeping back up files of a private server. Keep in mind that such a process is based on the technical characteristics and O.S.'s functionalities of the server. Therefore the details are case specific. In table 1 we can see an overview of the basic issues that it should address and solve. This list is constructed based on the recommendations stated by (Höne, et al., 2002b) (Walter, 2001) (Wills, 2002) and (Bowden, 2003).

Risk Assessment	Password Policies	Administrative Responsibilities	User Responsibilities	E-mail Policies
Internet Policies	Disaster Recovery and Backups	Intrusion Detection and prevention	Anti-virus and firewall usage	Suppliers of consumables
Network connectivity	Network Security Architecture	Remote Access and mobile devices	Staff training, awareness and education	Violations and legal implications
Disclaimers	Photocopy, fax and phone use			

Table 1 – Indicative functional areas of a security policy

The basic structure of an ISSP follows**:**

- Scope and definition
- Authorized users and assets
- Forbidden or illegal use
- System's management
- Implications in case of unauthorized use
- Review and update schedule
- Enterprise's liability

The authors proposing this structure accompany it with three different design methods – different ISSP for each issue, a company-wide ISSP document and an intermediate

approach that is supposed to offer more flexibility and effectiveness. However this issue is subject to evaluate.

Systems-Specific Policy (SysSP)

The last type of policy is the most specialized type and as a result is highly technical. It includes guidelines and instructions that apply to specific systems, programs or applications. This might include a reporting tool or an instant messaging application. Even though the scope of a SysSP is very narrow a managerial element exists and defines it. Sometimes it is divided in two different sections or documents, one that is originating from management and a second one that includes technical specifications. The first one includes the guidelines that are established i.e. by the IT department and should be "somehow" implemented i.e. only "admin users can delete entries. The second document describes the technical details, what steps should be followed in order to enforce the policy; all steps are described in detail.

For example, during the installation process of a new router every step is described explicitly such as factory-settings modification and new password setups. Even though these steps might sound trivial, it is important to be documented in order to avoid arguments about the optimum order of actions or follow well defined steps in case of emergencies.

2.3.5 Additional Characteristics and implementation

Apart from the main categorization of policies in modern enterprises, it is important to mention a few additional characteristics (Whitman, et al., 2011).

As mentioned earlier dates are an important aspect of a policy. An employee that is working under a specific policy needs to know – it is good for his own protection too – about the validity of the document. Knowing when the document was created or revised helps him understand whether or not it is updated, valid or not. Outdated policies might conflict with a newer version but the luck of date attributes would probably confuse users and make them question management's support for such operations.

Furthermore, dates should also be used to establish a time-table for future revisions or termination of policies. Keeping a policy updated with respect to latest events, technologies, trends and issues is necessary. A policy that refers to an outdated technology is useless and creates unnecessary complexity. Additionally it might not address important aspects of new implementations. For example what would be the usefulness of a policy regarding mobile devices that was referring to old-fashioned phones and not modern smart-phones full with possibilities and applications?

Another important aspect that might not be considered at first place is styling and presentation. Since we are referring to a document originating from the management of an

enterprise, it should be consistent with the attitude and styling options followed in the past. This tactic makes the readers feel confident that the policy is official and legitimate (Höne, et al., 2002a). Otherwise, using unanticipated context, themes etc. would make the employees feel confused and wondering about its importance.

Finally, in order to create a meaningful and useful policy document, feedback is required. Since a policy is a document that most of the time is affecting more than one person, it cannot be designed based on the decisions of one manager. Recommendations are useful especially from people that are actively involved in the field covered by the scope of the document. This is logical due to their everyday involvement; it is obvious that they do know better the requirements, problems and issues of the field. On the other hand, highly technical educated people cannot be solely responsible for a policy. Even though they are very well aware of a system's capabilities and how to modify one, they are not always trained to keep in mind that non-technical users are involved and they are not fond of reading documents that they do not understand (Höne, et al., 2002a).

Implementation notes

But how do enterprises inform their customers and employees about current policies that should be enforced? The obvious answer is a central repository. One typical example is (President and Fellows of Harvard College). At this site[2] we can see a number of policies that are active and gathered together. However it is unlikely that a user will visit regularly or more than one times this repository (Wood, 2000). Therefore it is a good practice to "make" users read and acknowledge that they agree with a certain policy. This can be done with various technics. Two of the most common examples are login and registration forms. When a user is about to register or login to a web site, especially those ones that involve financial transactions, it is required to verify that he/she has read the policy of web site. Depending on the nature of the site, the policy might include password usage, terms of use of the site, responsibilities of the company towards the clients etc. Using this method the users of the site are verifying that they have read the policy. However, the level of understanding the terms is a totally different issue.

For example in Figure 5 we can observe the method mentioned before. It is a registration form that a bank uses (Lloyds TSB Bank plc). A user is required to check the checkbox regarding terms and conditions. He is also offered a link in order to find out more about them.

[2] http://security.harvard.edu/enterprise-security-policy/

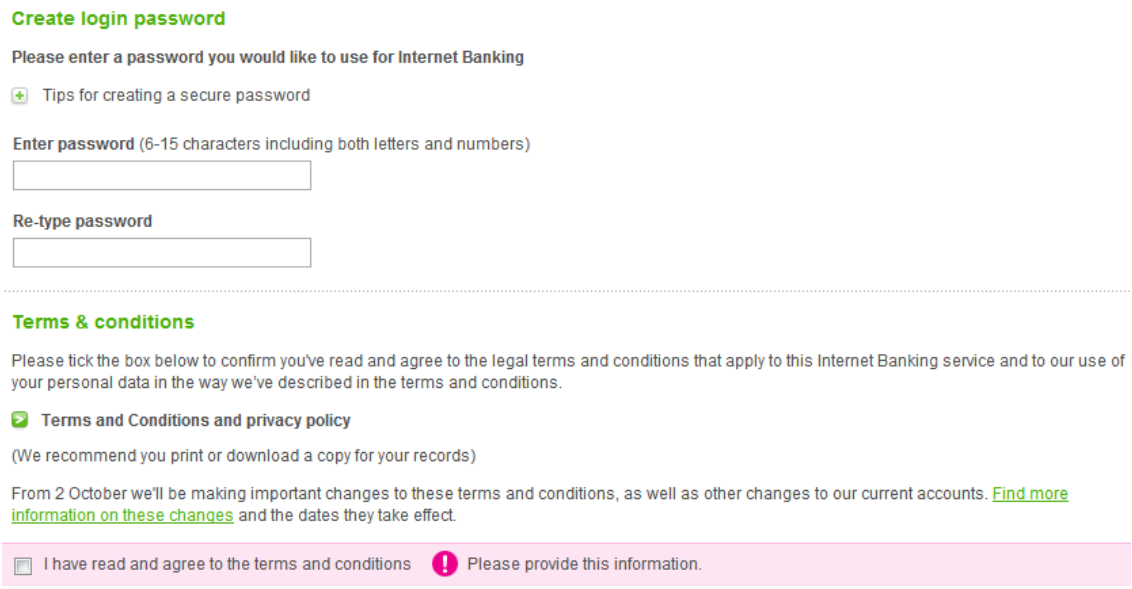

Figure 5 – Registration form and policy acceptance

2.4 Access Control Models

A security policy is the framework in order to establish the desired security level. As it has already been stated, designing one is a demanding task and significant amount of effort is required in order to be successful. However a policy itself is not enough in order to guarantee that the assets of a corporation are safe. That is because policies are just documents that express guidelines and instructions. For that reasons access control models have been proposed, that are established and continue to evolve based on current technology and needs. Access control models are the next step after the strategic and operational policies have been established. Following a model that fits the company's needs, the security policy can be implemented and enforced.

In this chapter we will analyze some[3] of the existing models that are established throughout years along with some that are less popular. Additionally some comments will be made in case that something is related to the prototype of the thesis.

2.4.1 Overview

Security controls can be categorized in two different taxonomies. The first one is based on the nature of the control (administrative, technical or physical). In this thesis and due to the nature of the development phase, we are most interested in the technical nature of security controls. The second taxonomy is related to the way the control is functioning; that is

[3] The selection of the less popular models was a matter of personal preference.

detective, preventive, corrective and recovery (Purcell). The former definitions are considered self-explanatory and no further analysis is done. In general an access control is placed in between a subject (i.e. user, process, application) that is trying to perform some actions on an object (i.e. database, files, applications).

2.4.2 DAC

Discretionary Access Controls is the simplest type. It is based on the notions of individuals and resources. The administrator of the system is responsible for allocating privileges (Own, Read and Write) to each user individually for each resource. With respect to small applications, it is easy to manage and handle since the amount of users and resources tends to be manageable by a person (i.e. <10 users, 3-4 personal computers and 1-2 servers). The organization of users and resources is done as it is presented in Figure 6. It is clear the amount of work required in order to setup access privileges for each user. Furthermore this process has to be repeated each time we want to modify or re-evaluate it. However, as the size of system increases the complexity is becoming significantly important.

One of the major drawbacks that this model has is that it does not make distinctions between users and processes that run on behalf of them (Samarati, et al., 2001). For that reason the model is taking as granted the reliability of a task that has been initiated – not always being aware of it – by a "certified" user. This creates a vulnerability that can be easily exploited by injecting malicious code into applications that trusted users execute. Furthermore, since access control rights are defined for each resource separately by an individual (Wenliang, 2011), it is easy to create inconsistencies relative to the global policy that shall be followed.

An access matrix can be implemented using three different methods: authorization table, access control list and capability. They are not going to be analyzed since they do not contribute neither to the theoretical part nor the development phase.

	File 1	File 2	File 3	File 4	Account 1	Account 2
User A	Own R W		Own R W		Inquiry Credit	
User B	R	Own R W	W	R	Inquiry debit	Inquiry Credit
User C	R W	R		Own R W		Inquiry debit

Figure 6 – Access matrix[4]

[4] Source: http://pic.pimg.tw/nixchun/1186492320.jpg

2.4.3 MAC

Mandatory Access control enables a central regulation authority to be responsible, monitor and modify access rights based on a policy. The MAC model is using the notions of user, subjects, objects and operations. An object is a process or application that runs on behalf of a user and it has a security clearance that is inherited by the security clearance of the corresponding user. On the other hand, objects are resources (i.e. databases, printers) that users want to access; objects are labeled with a security classification (i.e. Secret, Top Secret, Confidential and Unclassified); those values are ordered. Same applies for clearance level. The system examines the labels, compares them with the rules applied by the policy and based on the outcome it provides (or not) the corresponding privileges to the subject for the particular object.

For example, in the Bell-LaPadula model (Bell, et al., 1973) it is defined that an object cannot be read by a subject that has lower clearance. This control definition is established by the creator of this model and individual users are not allowed to break it because this constitutes violation of the model itself. Similar rules exist in Biba Integrity Model (Biba, 1977) but the in the opposite way; a subject cannot read a resource that has lower security classification than its own security clearance.

A good example of MAC model can be easily understood in a database. In a DBMS we can distinguish users (it refers to clearance level i.e. visitor, registered user, staff and admin) and based on their clearance, provide them a different view[5] of the database. Furthermore each user can have different access rights in different tables of the database (Figure 7). The main authority that regulates those rights is the web administrator that has access to the admin panel and enforces the policy. However additional actions need to be taken from scripting point of view. This means the use of the corresponding PHP commands that assist in implementing the controls (6.3).

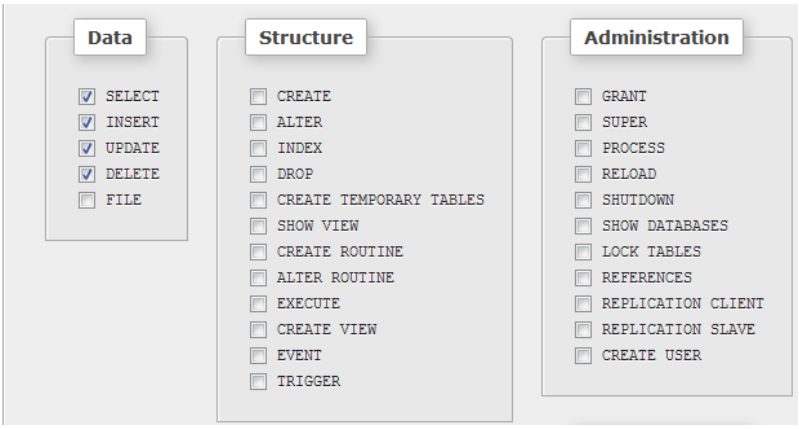

Figure 7 – User privileges in a MySQL database

[5] MySQL documentation for view creation:https://dev.mysql.com/doc/refman/5.0/en/create-view.html

In MAC model, only the central authority that manages the system can modify privileges unlike DAC where the owner of each resource is allowed to modify access rights.

2.4.4 Location-Based MAC

A very interesting variation of the previous model is a location-based mandatory access control model (Indrakshi, et al., 2006) that is mainly focused on securing military applications. It is using the location characteristics of the major elements that MAC has and integrates them as an additional security control.

As seen in Figure 8, each element of MAC (object, subject, operation and user) apart from their respective security level (clearance and classification) have a location attribute. The authors propose certain restriction rules that must be enforced in order to guarantee the security compliance of the architecture. For example, a user must be in the same location as the subject (i.e. device) that acts on his behalf. Additionally, a subject and an object should be in allowable locations in order operations (read & write) to be perform on the first one. The model is using numerous controls in order to verify the security quality of the system that implements it.

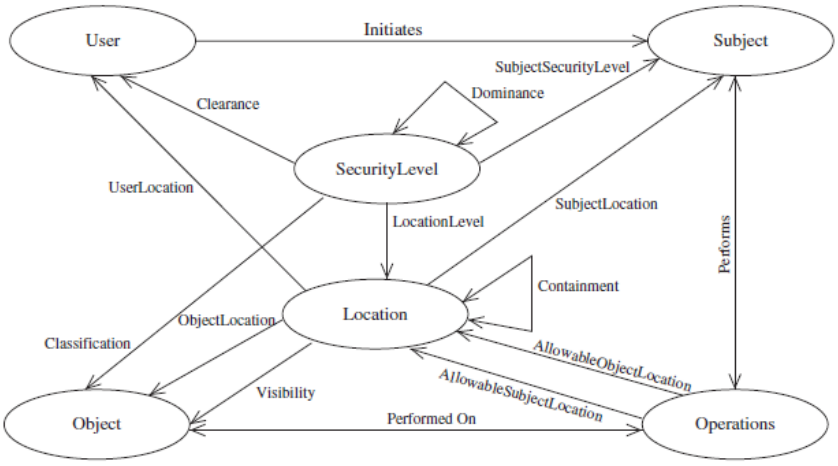

Figure 8 – Relationship of MAC components with location[6]

It is offering an additional degree of security however there are certain implications that must be mentioned. First and most important is the location attributes. They constitute personal information regarding users and extremely important to be confidential regarding the physical assets. That means that unauthorized access to such kind of information must be prohibited. For that reason "location visibility" is used – a term that describes the "accepted" information about location that is permitted to be displayed. Additionally the concept of time is not integrated up until now but authors note that will be included in future research.

[6] Original image source: (Indrakshi, et al., 2006)

2.4.5 RBAC

Role-based access control is a more commercial-friendly type of model that is greatly used. It allows the mapping of users based on the roles they have inside the corporate environment. Using this method the administration of the system becomes easier compared to the more traditional MAC and DAC models; even in cases of huge number of users. That is because users are grouped under a label that characterizes them i.e. administrators or regular users. In case of a change in the access privileges is required then the system administrator can just modify the group and automatically the rights are transferred to each entity that belongs to that group. On the other hand, if a new user is added or an existing needs to be modified, then by changing his group all his access rights are updated based on the new group. Figure 9 is depicting a simple representation of the model. Each entity is linked to a label (access right group) based on their job role. Automatically they have access to different resources.

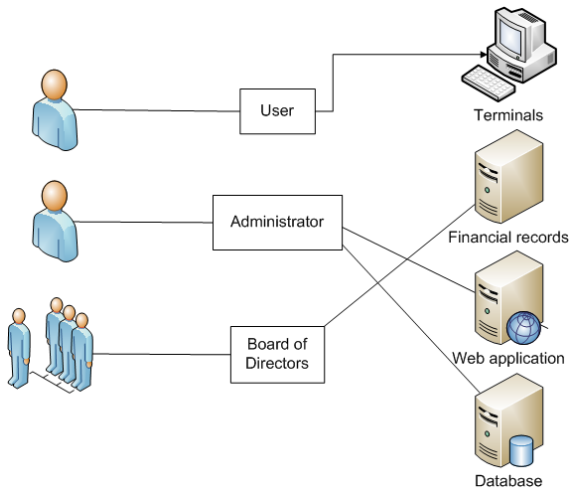

Figure 9 – Role-based access control example

Furthermore, RBAC model supports not only access to different resources but also different type of access to the same resource i.e. read, write.

Generally the advantages that RBAC introduces summarize to (Samarati, et al., 2001):

- Authorization management: As mentioned earlier adding new users or modifying existing roles is greatly simplified.
- Hierarchical roles: As job roles are hierarchically positioned (operational employees are sub ordinaries of executives) in a similar manner the roles of the security model can be described in a hierarchy that allows easy comprehension.

- Least privilege: RBAC and role-nature allows the implementation of this core security principle. Users have the ability use resources with the minimum privileges required in order to perform their duties and tasks.
- Separation of duties: Another important aspect is employed; that is different roles exist (as an extension different users) to perform different tasks. Separation of duties enhances security, monitoring and guaranties higher degree of compliance.
- Constraints enforcement: The use of roles allows the use of specialized and targeted controls and constraints on the system.

RBAC model is used in the development phase of this thesis in order to design and implement the system prototype.

2.4.6 Chinese Wall

Chinese wall was originally proposed (Brewer, et al., 1989) in order to solve situations that were not sufficiently addressed by military-oriented models such as the Bell and LaPadula. The decision that is made in order to allow an entity to access some resources is based neither on labels nor privilege rights given by a central administrative authority. The main difference is that each entity that is about to begin using the system is able to be granted access to any resource available. Furthermore, future requests for access is based solely on past choices. This model can be applied at its full potential in accounting or consulting firms.

Key concept in the Chinese wall paradigm is the “conflict of interest classes”. Under each class are grouped companies that are competitors in a specific business area. Consider the example shown in Figure 10. If a business consultant is about to start providing his service to a company, the first time he is allowed to pick anyone. If he picks Coca cola, then due to the definition of the “simple security rule” of the model, he is not allowed to offer his services in Pepsi or any other potential company that belongs to the same class. The reason for this is that he is aware of confidential information of that particular company and he should not be able to disseminate at direct competitors. Nevertheless he is allowed to pick another company that belongs to a different conflict of interest class, i.e. Mobile phones manufacturers (Google).

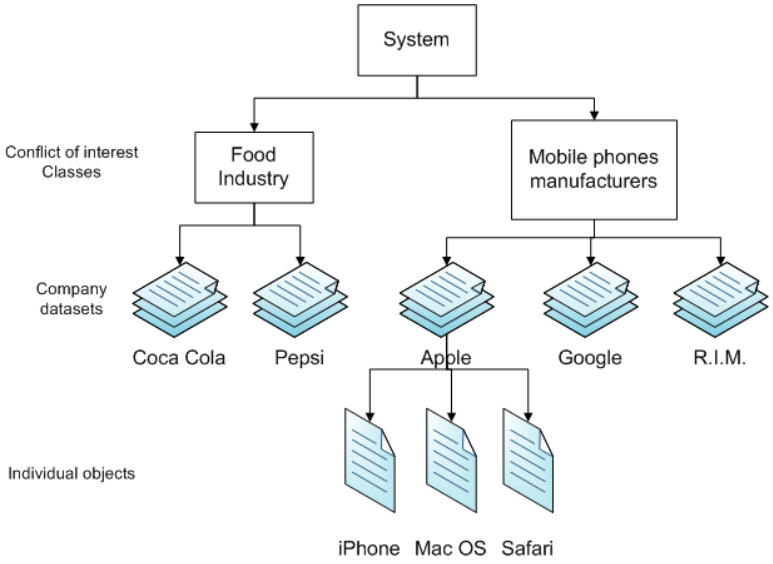

Figure 10 – Chinese Wall architecture

On the other hand, someone that has access to a particular asset of a company (i.e. iPhone) he is allowed to request access rights to a different asset of the same firm since those two pieces of information belong to the same company and no conflicts arise.

Additionally, the definition of the model allows free distribution of information throughout the system – even between competitors. However, information needs to be sanitized, not traceable back to the original source and needs to be considered harmless to the company's interests.

The authors provide with a whole set of formal rules that facilitate the implementation of the system and it provides a very business-friendly framework for information security in the modern world.

2.4.7 DACA

Dynamic Access Control Architecture (Chuchang, et al., 2011) is an extension of an existing model (International Organization for Standardization, 1996) that was originally proposed by ISO/IEC[7] at 1996. It is an interesting concept and compared to the established models (MAC, DAC RBAC) it introduces a different aspect on decision making with respect to allowing entities access resources. Traditional models use a Boolean variable (YES or NO) in order to answer the question: Should user A read the contents of the database X? Based on the access rights that he has, the decision is taken. However DACA induces a mechanism that incorporates logic rules in order to perform such actions. For that reason, the outcome of the decision is not always granted.

The basic components of the model are:

[7] http://www.standardsinfo.net/info/index.html

- Initiator: is the user or the process that acts on behalf of one and is making the initial request for accessing, reading, writing etc. a resource.
- Targets are the resources (databases, printers, files, applications) that are being requested.
- Access Control Enforcement Function (ACEF) is the module that receives all incoming request for resources and promotes them to ACDF. Based on the returning message, ACEF acts accordingly and either allows or prohibits access to the targets.
- Access Control Decision Function (ACDF) is gathering information from both the policy generator and the database. After the evaluation process is finished the result is returned to the ACEF.
- Database for authentication purposes that includes information such as user names, passwords etc.
- Policy generator, or just policy, is the local policy document that describes the conditions based on which the decision is being taken. An important note is that the local policy should be aligned with both the global policy that exists and the overall security requirements. The last one is not checked by the model and it is considered true.

Below there is a graphical representation of the way the components are connected and interact with each other (Figure 11).

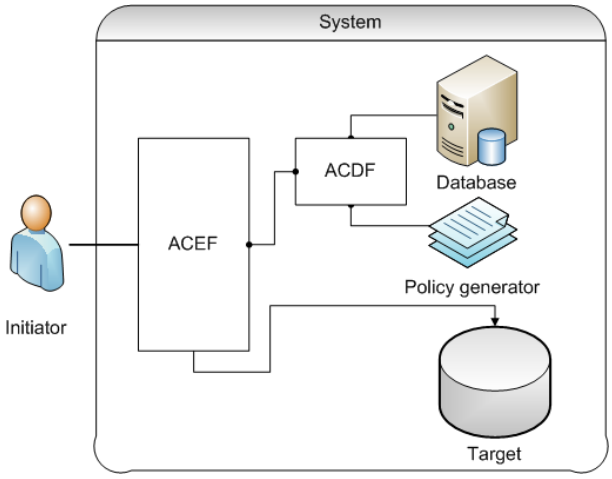

Figure 11 – DACA architecture

Key components of the system are the policy generator along with the Access control functions. Additionally, ACDF includes the observation vector on which additional information is stored and lead to re-evaluation of the access rights given up until that moment.

2.4.8 Clark-Wilson

This is another integrity-based access control model (Clark, et al., 1987) as Biba. However it aims in being implemented in business and enterprise environments rather than in military. Key concepts that form the model are:

- Transactions and
- Separation of duties

Transactions refer to all operations that occur between the existing entities that form the system and transform it from one valid state to another valid state. Separation of duties is one of core principles of Information Security; different entities are given different privileges and access rights so as no one gathers full and uncontrolled authority. Based on those concepts the Clark-Wilson model is establishing two kinds of rules:

- Certification
- Enforcement

And four components:

- Constrained Data Items
- Unconstrained data Items
- Integrity Verification Procedures
- Transformation Procedures

in order to maintain the integrity in a business-oriented system.

2.5 Mobile environment

As it has already been mentioned (0), one of the major threats that modern enterprises face is related to the human factor and the way they act. Employees that have been granted access to company's assets are supposed to behave according to rules and policies but this is not always the case. Due to either malicious motivations or accidental misuse unexpected and undesired situations might occur, should risks are exploited. On top of that, the large penetration of mobile devices (Mlot, 2012) is driving companies to a paradigm shift that mobile phones and tablets act as an extension of their workforce network that enables higher productivity and better user experience.

But do companies take seriously the threats that arise from this shift? Is there a reason to worry or are these risks an exaggeration?

2.5.1 Corporate point of view

As a starting point it should be mentioned that companies are looking positively the BYOD trend. The reasons for this vary but at the end they meet at the same point; money

(better profits or less expenses). When an employee is given a corporate device (mobile phone, tablet or PDA), it is explicitly stated that he does not own the device. This means that he is not allowed to modify it at his own will or performing tasks irrelevant to business operations. This includes installing third party applications, i.e. social media or photo sharing apps. An even more extreme scenario would include the device being locked by administrators so that users would not be allowed to do anything not approved.

Even though this sounds as a very secure model that enforces certain corporate policy that is distributed from top to bottom of the hierarchy, it induces several drawbacks that shall not be ignored. First of all, mobile phones are treated by users as their personal asset. It is a tool that is necessary for everyday tasks and is a major facilitator in business operations. Due to their close interaction with such devices, users need to feel comfortable using them. But forcing them use devices that by definition do not belong to them, makes employees keep a more indifferent or even negative stance towards them. As a result, phones are left unmaintained; firmware is not updated to include latest security patches introduced by the manufacturer. Even worse some might buy their own devices of their personal preference in order to complete corporate tasks and access assets of high sensitivity; devices which are not monitored at all. The bottom line is that companies spend resources in order to provide portable access for their employees, but they either bypass or mitigate the security measures enforced.

On the other hand, accepting BYOD as a valid business practice introduces both positive aspects and emerging security vulnerabilities. Giving someone the possibility to “bring his own device” to work and perform his job, provides a degree of personal satisfaction. This occurs due to the fact that his is personally involved in the process of acquiring the device. As a result there is greater degree of taking care of the device (upgrading operating system, not leaving unattended the device, making sure it is correctly maintained). Furthermore daily duties are performed via a device that satisfies certain personal criteria and not via one that was mandated by senior management or by the head of the department. As employees are allowed to use their own mobile phones, they are also paying for them. Therefore certain expenses are mitigated or even nullified. Even though this is not the major driving force of this change, it is shall be taken under consideration.

Assuming that BYOD is not just a trend but tends to be a core aspect of modern business models, corporations need to design and enforce security plans that mitigate the increasing number of threats that occur. However there is evidence that display how much aware (or unaware) managers are. Many interesting results can be drawn out of a recent study

(Goode, 2010)[8]. The study was conducted on various business sectors and key personnel were included, such as CISOs, security analysts and consultants. The single most important number that was derived out of this research was that *"almost 56% of the respondents do not have a specific documented security policy that covers mobile phones"*. This number can be considered high, even scary given the importance of information mobile phones can access.

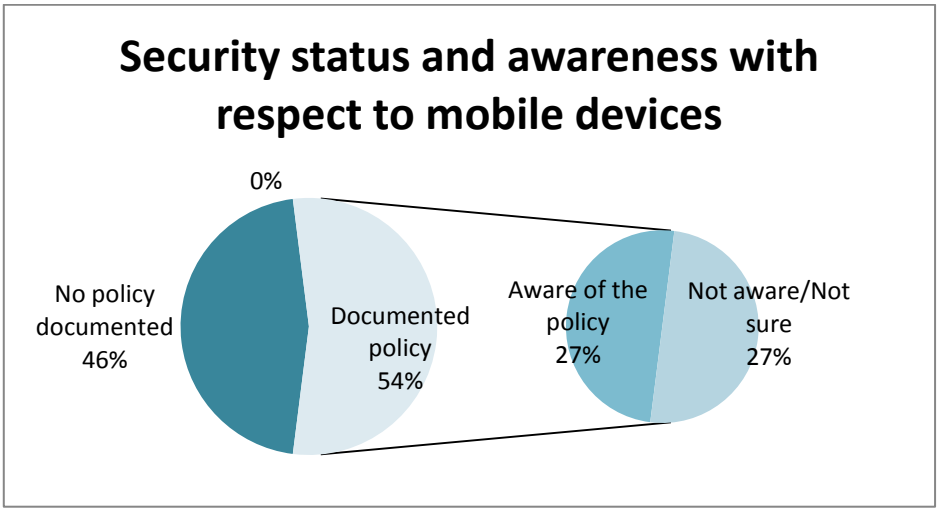

Table 2 – Security awareness and policy documentation

Additionally, Table 2 allows even more in depth analysis. More specifically out those that have indeed a documented policy regarding mobile devices, there is a significantly high percent that presents uncertainty about the awareness status of their employees. This has a great impact on mitigating the purpose of the security policy and allows us to speculate that the actual number of a documented security policy that is actually enforced and used as supposed to drops to just 27%. Even though this is a risky and oversimplified call, nevertheless the actual figures might not vary much.

Furthermore, Table 3 displays the importance of securing mobile applications conceived by the same respondents. It is encouraging that more than 75% consider important the aspect of securing mobile devices and applications against threats that might target their companies.

[8] Goode Intelligence(GI) specializes in market intelligence and research and analysis for the Information Security and Mobile Phone Security sectors (http://www.goodeintelligence.com)

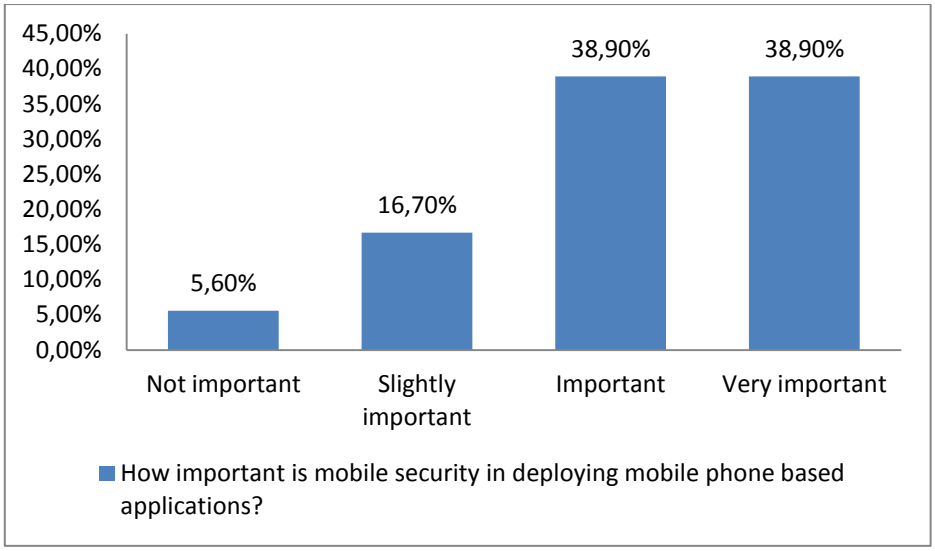

Table 3 – Importance of level of security

The last important figure that derives from the last survey is related with access to corporate information and assets. Only 42% of companies allow remote access to their corporate networks. Main reasons for that are both security policies and the lack of motivation to do so.

By examining the tables that were exported out this survey, one can draw interesting conclusions regarding current status in the field of our study. It seems like that most of seniors and executives that were questioned take seriously the threats that exist with respect to mobile phones but they are not taking appropriate measures. There is either lack of proper actions or over-enforcement of measures. There is certainly the need to take actions that are better aligned with business needs and functions i.e. do allow remote access but use proper access control models, educate and inform employees about the threats, risks and measures.

2.5.2 Modern Platforms

The last survey depicted the current trends with respect to mobile devices and information security. However there is not a single platform or a single manufacturer therefore many questions and security issues arise. Do all platforms share common vulnerabilities? Does a unique way of protecting corporate assets exist?

As a start we need to mention that at the moment five major mobile platforms exist but two of them top the market with more than 80% of total market share. More specifically and based on a research conducted in U.S. that involved more than 30,000 mobile subscribers (comScore, 2012)[9] and lasted for 3 months, half of the market share belongs to Google Android and Apple iOS follows (Figure 12).

[9] comScore is specializing in digital marketing intelligence (http://www.comscore.com/)

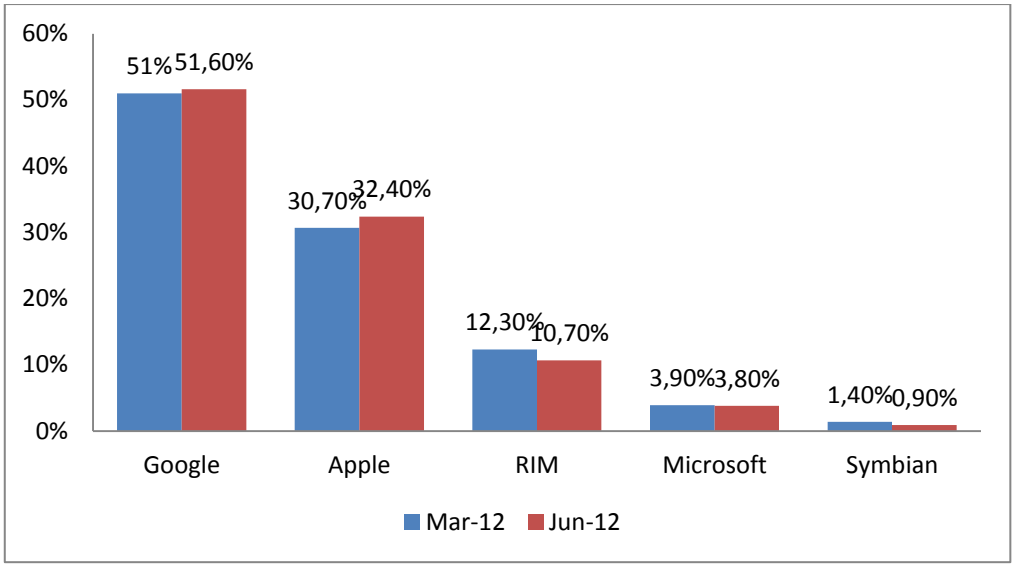

Figure 12 – Market share distribution per mobile platform

Just by looking the last figure, we understand that the wide variety of technologies that exists in modern environment. Furthermore each device manufacturer (i.e. Samsung and LG, both running Android) is implementing his own technologies and hardware specifications that introduce an even more difficulties in enforcing a universal solution.

2.5.3 Motivation for choosing iOS

For those reasons, the scope of this thesis is narrowed and focused on the iOS operating system which is used at iPhone, iPad and iPod touch devices. Additionally this platform was chosen due to its bigger business impact of the prototype system. The role and the purpose of the prototype application are focused on business devices that are used in corporate environment. In this context, Apple products present higher penetration compared to the competition. Even governments seem to adopt them and integrate them into their internal procedures by either developing dedicated applications (Clover, 2011) or even creating app stores for government agencies (Apple Inc.). A federal body, namely NSA, approves and considers Apple platform as one of the most secure available right now in the market (Garfinkel, 2012) that incorporates many advanced features; among those is the use of Advanced Encryption System (AES-256). AES was approved by NSA as a standard encryption algorithm. However this does not mean that the device is full proof. There are always ways to exploit technology by taking advantage of its flaws. A recent example of exploiting SMS spoofing in iOS indicates the need to be cautious not matter the technology (Kovach, 2012). Also the last exploit can take place in the last iOS version that is about to be launched in fall 2012. The remaining parts of the thesis will focus on analyzing security vulnerabilities, issues, exploits and development for the iOS SDK.

2.5.4 IPhone technical security controls / data storage

IPhone is one of the most popular mobile devices due to not only its style and design but also its very user friendly and attractive operating system that offers a very smooth and "eye-friendly" user experience. But apart from the package, is iOS secure enough? Do vulnerabilities exist that threaten corporate information stored inside those devices?

Apple has established a well-organized framework and a series of controls to provide as much monitoring and protection as possible to the ecosystem that has been created around its products. Four major categories of controls are deployed throughout the life-cycle of the applications that are being distributed by the Apple store:

- Address Space Layout Randomization (ASLR)
- Code Signing
- Sandboxing
- Data Encryption

A more detailed analysis follows based on a report published by a consulting firm (MDSec Consulting Ltd, 2012) and a white paper that was presented at Blackhat[10] 2011, Las Vegas, US (Dai Zovi, 2011). These controls are not completely fool-proof but they contribute in the overall security level of the platform.

Address Space Layout Randomization

In short ASLR is one of the techniques used to protect applications at run-time by hiding information regarding the memory layout of processes that run. Therefore, each time a process is being executed various libraries, data, stacks and binaries are loaded at random memory addresses making hard for a hacker to inject malicious code, i.e. hook onto the methods. Without ASLR, injected executable code is relative easy to exploit running applications.

Particularly for iOS, ASLR was first introduced at version 4.3 and it is divided in two categories: limited support for ASLR and full ASLR. The difference in these two types lies on the way an application is compiled. In case of limited ASLR, not every file is initialized in a random location; this includes the main thread's stack. In order to have full support, the application will have to be compiled using Position Independent Executables (PIE). In that case everything that is used during the run-time is loaded at random memory addresses which are not known in advance. This combined with "W^X" policy[11] makes the use of malicious techniques such as return oriented programming (Buchanan, et al., 2008) much harder.

[10] The Black Hat Briefings are a series of highly technical information security conferences that bring together thought leaders from all facets of the InfoSec world (http://blackhat.com/).
[11] A memory page is flagged as either readable or executable but never both.

Code Signing

In order to further elevate the quality of security controls in place, Apple enforces the use of signing certificates. Those are used to verify the origin of each code that is being executed. This helps trace the source of potential problems. An example of code signing includes the developer certificate that is obtained from Apple on request and it is used as a personal identification of the developer in case of malicious applications. Furthermore, applications are signed by Apple during the submission process in App store in order to prohibit the execution of non-approved apps in iOS devices; additionally a trusted app is not allowed to access or execute "untrusted" resources. This check enables to make sure that no security breaches are possible through backdoors. In general the MAC model that is followed in signing includes many steps until an application is verified, installed and allowed to be executed at individual phones.

Sandboxing

During the execution of applications, the operating system enforces restrictions to them. One of them is the file system and the directories that they can access. An isolated environment is created and communication with the "outside world" is prohibited apart from certain exceptions, such as network tasks and contacts. As a result the application is able to view the app specific folder that is assigned to it by the O.S. while on the other hand the root directory of the system is not accessible along with sensitive and private information.

Data Encryption

In order to prevent the confidentiality of information stored inside an iPhone, an entire encryption and protection system is in place that provides a wide range of functionalities. First the device has integrated dedicated hardware that allows quick encryption using Advanced Encryption Standard (AES). Various keys are used in this process such as the device unique ID key (UID), a global shared key (GID) and the password that is provided by the user to unlock the phone on boot. More specifically, by the time the user is enabling the PIN to protect his device, file encryption is automatically enabled and it is enforced based on the parameters provided. Furthermore, high level API (Security Framework[12]) exists that allows the use of encryption and protection classes in order to perform such operations. Table 4 presents a couple of the major classes used for file handling that include the ability to protect their instances along with the corresponding data. This table represents example values that can be used during the corresponding operations. Based on those values, the O.S. will treat data accordingly and appropriate protection measures will be taken. For example, NSDataWritingFileProtectionComplete value will result in encrypting a file and making it inaccessible while the phone is locked or disabled.

[12] http://developer.apple.com/library/mac/#documentation/security/Reference/SecurityFrameworkReference/_index.html

Class	Operation/Attribute	Example Values
NSData	NSDataWritingOptions	NSDataWritingFileProtectionComplete
NSFileManager	NSFileProtectionKey	NSFileProtectionCompleteUnlessOpen NSFileProtectionCompleteUntilFirstUserAuthentication

Table 4 – Examples of encryption-related API operations

The use of such operations allows the implementation of a number of protection mechanisms and the input of the user's password is required most of the time in order for files to be accessible again. However it should be noted that the password is – usually - four characters long; a user-friendly but the same time bad password practice. The security level increases as the length of the passphrase goes higher. Even though brute-force methods can be applied in all kinds of passwords, the iOS security model is combining passwords and the device's hard-coded unique ID. Therefore those operations should be performed on the phone. However Apple has integrated a small time-delay in such operations making time-consuming brute-force attacks to crack long passwords (Apple Inc., 2012). For example a six-character alphanumeric pass would take more than five and a half years to break. Furthermore, since the UID of the device is used during the encryption process, files that are extracted from the phone are still in un-readable format and considered safe.

Finally, iOS is providing an API to use keychain services[13]. A keychain is an encrypted storage that allows the safekeeping of various credentials that users need such as credit card password and card number, Wi-Fi passwords or credentials to remotely connect to other systems. The keychain allows defining different types of passwords; based on the type a specific policy regarding the level of accessibility is enforced. For example Safari passwords are available by the time the device is unlocked while the SIM's PIN is available all the time (encrypted) in order to be able to unlock the device. Additionally the keychain assists in avoiding situations where the user needs to constantly provide his credentials. Each time he is trying to access i.e. his mail, his username and password are retrieved from the keychain.

2.5.5 Transport security

Establishing a good and secure environment to run applications is not enough. Usually information is not created and just stored into the device. An application that is allowed to access corporate (or any kind of) data that are of important value should be able to

[13] https://developer.apple.com/library/mac/#documentation/security/Conceptual/keychainServConcepts/02concepts/concepts.html#//apple_ref/doc/uid/TP30000897-CH204-TP9

guarantee safe transmit of it. Once again certain protocols and classes provide the necessary functionality.

The iOS SDK allows the use of TLS 1.2 protocol as of version 5.0 and it includes 37 different cipher suites (Figure 13). This achieved by the NSURL class[14] that is responsible for creating HTTP(S) requests. Even though some of those suites are characterized as weak they implement various well-known methodologies such as Diffie-Hellman for key exchange and RSA for asymmetric encryption. Additionally an alternative option instead of NSURL exists by using the CFNetwork framework[15] that allows performing similar operations. However the last choice has a major drawback; it is not very user-friendly to implement. To solve this issue a high level library exists that acts as wrapper and is implementing the desired functionality (Asynchronous/synchronous requests, HTTP or HTTPS, certificates etc.). ASIHTTPRequest[16] is free and was maintained by individuals up until May, 2011. Even though it is not completely secure, nevertheless acts as a starting point in using CFNetwork framework. For that reason ASIHTTPRequest was used in the development process of the prototype application. Furthermore iOS devices support major Wi-Fi protocols along with WPA2 Enterprise which uses AES 128-bit.

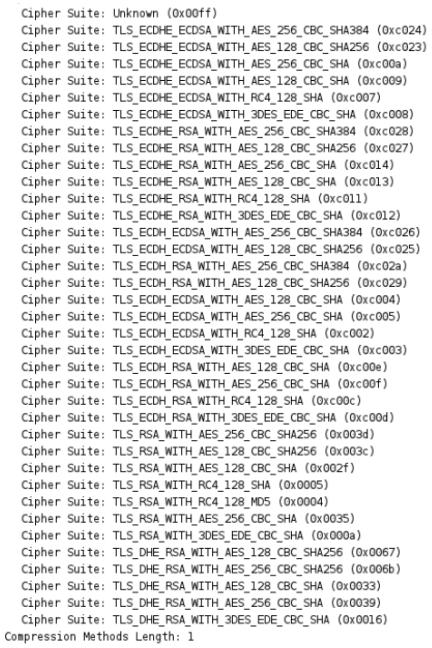

Figure 13 – List of cipher suites (MDSec Consulting Ltd, 2012)

[14] https://developer.apple.com/library/mac/#documentation/Cocoa/Reference/Foundation/Classes/NSURL_Class/Reference/Reference.html
[15] https://developer.apple.com/library/mac/#documentation/Networking/Conceptual/CFNetwork/Introduction/Introduction.html
[16] http://allseeing-i.com/ASIHTTPRequest/

2.6 Additional functionalities and measures

However the functionality offered by the iPhone device is neither the golden rule nor the only solution in order to increase the level of protection in the information stored in mobile devices. A standard mobile device (phone, laptop or tablet) offers users the possibility to lock the device using Personal Identification Numbers (PIN) so as to prohibit using them without their authorization. PINs are part of the common practice that today is used; two factor authentication (2FA). The second part of this process usually is the Subscriber Identification Module (SIM card). In order to identify and authenticate users three different mechanisms have been established (Wood, 1977):

- Something that he has.
- Something that he knows.
- Something that he is.

Based on the last list, current 2FA technics use "has" and "knows" aspect by integrating the SIM card and the PIN. Up until now this was an accepted and generally a secure process to protect mobile assets.

But do users implement and take full advantage of the protection that passwords offer? Studies indicate that this is not the case (Clark, et al., 2005). It seems that more than 85% of users do not change their PIN more than 1 time while the same time 36% of them use the same PIN for multiple purposes. This practice sounds logical to an everyday user since he does not want to be bothered with remembering lots of complicated passwords. Additionally using the same password over multiple platforms means only one thing; someone that obtains a single password can gain access to multiple systems. However these bad habits threaten the foundation of the security model that applies to mobile devices. They completely mitigate the purpose of PINs, a fact that does not align with the scheme proposed by Wood; a password should be difficult to guess, easy for the owner to remember, frequently changed and well-protected.

In order to keep security level at an acceptable level, the third aspect of identification/authorization needs to be integrated at a greater degree; that is Biometrics - "something user is". Up until now the most common and known technic is fingerprint recognition that has been mainly integrated into laptops[17]. However several additional methods exist in identifying the user of a device (Table 5). Based on the same survey, methods that provide high accuracy also suffer from high intrusiveness (iris scanning, fingerprint). Users feel uncomfortable when such technics are being employed in order to perform simple tasks such as accessing mails.

[17]NEC fingerprint technology: http://www.nec.com/en/global/solutions/security/technologies/fingerprint_identification.html

Ear shape recognition	Facial recognition	Fingerprint recognition	Hand geometry
Iris scanning	Keystroke analysis	Service Utilization	Voiceprint recognition
Handwriting recognition			

Table 5 – Biometric technics

On the other hand, identifications such as face or voice recognition that can be implemented during casual use of the phone seem to be more preferred.

All these technics sound promising in order to make mobile devices more secure. However various issues exist such as the lack of standardization (Farnworth, 2008). Furthermore the final choice of a particular method will not be left on the end-user no matter his preferences. Business needs of the company that desires to implement such technics must be taken under consideration since those requirements will justify the costs.

3 Prototype development

This section of the thesis is focused on the development phase of a prototype that is implementing some of the core functionalities an enterprise mobile device should include along with a web-based policy management system that allows a centralized and user-friendly control. Those two components constitute the system that allows both a policy development stage and the enforcement aspect in a user-friendly and easy manner.

3.1 Overview

The system is comprised, as it was already mentioned, of two major parts. The first one is a mobile application that has been developed for the iOS platform. The second is a web-based management platform. Its main purpose is to enable system administrators an easy and user-friendly way of creating an XML-based document that reflects corporate policies. More details about each component will follow. In "Figure 14 – System overview" the overall structure of the prototype is presented. It is clearly depicted the way that the components interact, the roles that exist into the system and the basic operations that occur between them. The system administrator is responsible for the policy creation, label and device handling. On the other hand the user is unable to modify either the policy that applies to his device or his corresponding job role.

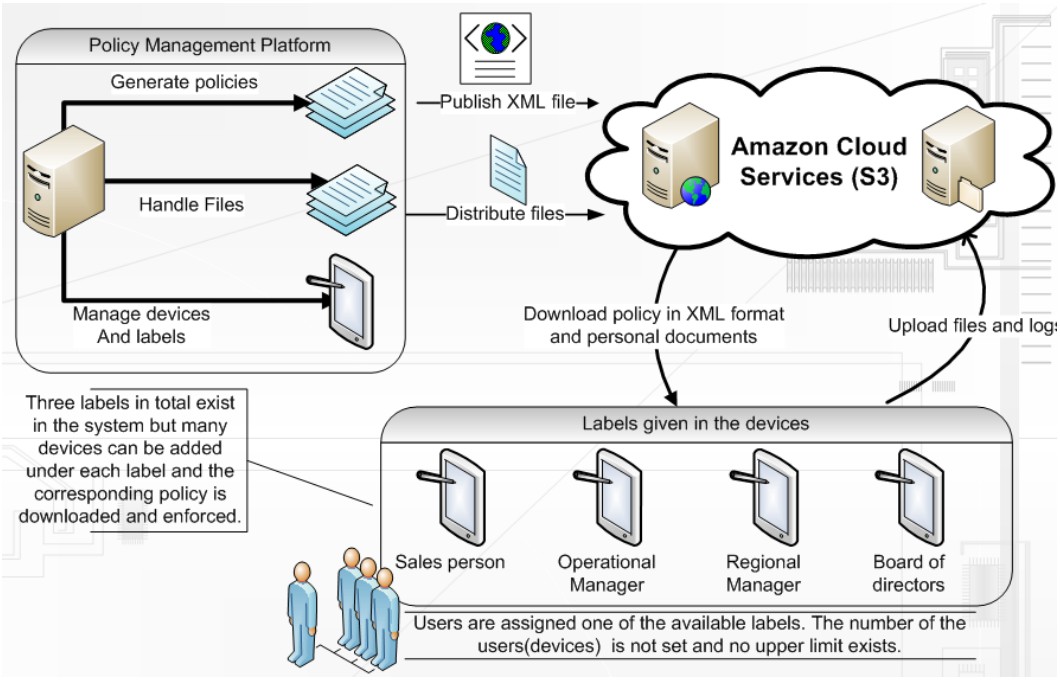

Figure 14 – System overview

3.2 Requirements Analysis

The prototype system that was developed had as final goal to protect enterprise's assets from threats caused by both malicious and accidental use. It is a control mechanism based on the Role Based Access Control model (section 2.4.5). The reason it was designed this way was mainly to deal with the need that employees have different privileges based on their job role. For example a senior employee might not need access business documents while he is away from his office however he is able to do that no matter what time is. On the other hand an employee who work as a sales man and performs door-to-door visits would most likely to need access no matter what his location is. Therefore there is a need to categorize job roles and using this as a starting point start creating the policy that should be enforced. An important note must be mentioned; job roles and access privileges differ from company to company even though some generic roles might exist.

For that reason, the prototype has implemented some basic business roles that possibly exist inside a company. However this is not restricting at all and can be modified on demand. In Table 6 there is a list that presents the roles used in the prototype. According to the role that is given to each employee, a certain policy is downloaded.

List of roles used
Sales Person
Operational Manager
Regional Manager
Board of Directors

Table 6 – Job roles used

In order to enforce the labeling system, each user is associated with a mobile device using its UID. Each time the mobile application is requesting from the system the policy document, it is not aware beforehand which will be the XML file. It only transmits its own UID. The server side of the system is responsible to decide which file to send back as a response. Furthermore, the system administrator is able to modify the role assigned to each user (this action affects the associated device), edit or delete existing users, create new documents and view existing policies. This is the core set of the functionalities. In the following sections an extensive analysis of each part (mobile & web aspect) follows. In order to design an efficient requirements analysis document, the designer needs to contact the stakeholders that are interested in the application. Not every company is interested in the same restrictions or information. For example one might want to know the GPS coordinates throughout the day or others might just want to be informed in case of violations in the policy. That role was fulfilled by Mr. Karagiannis Sotiris & Mr. George Mavroudis from Space Hellas S.A that assisted in establishing some key restrictions that the application should

monitor. Finally, those requirements were set based on the developer's experience, available resources and scope of the project within the available time frame.

3.2.1 Mobile application

From the mobile device point of view, the application acts both proactively and reactively based on information it retrieves from its sensors (i.e. GPS, time) or the policy document. Furthermore it acts as a logging application that tracks and stores information about the usage of the device (i.e. carrier network, battery levels, potential IP address). After such information has been stored, they are uploaded to the Amazon cloud service. The frequency of both updating the log file and uploading files to the server is based on the policy – it might not backup files at all if it is not set. This assists in monitoring and verifying the compliance of the device. The *core concept* of the mobile application can be summarized in the next two steps:

1. **Download, parse and store the policy that is defined in XML format.**

In order to be able to enforce a policy, a certain document needs to be downloaded to the device. The chain of events begins by the client side. The mobile application is sending a POST request to the server including its own UID number. This number is already stored in the system database by the administrator. At this point, the web application is trying to define the necessary document name to send back. This step is being done by evaluating the statement: <<device-user-role-policy>>. Starting from the device, at the end the system is determining which the correct document to send back is (Figure 15).

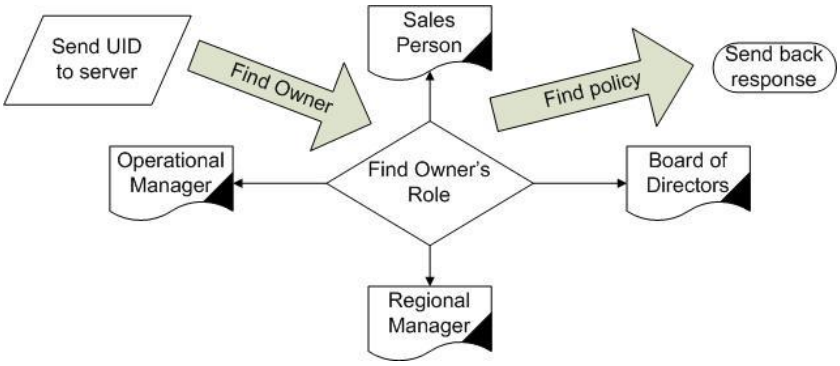

Figure 15 – Steps to define XML file to send

After the document name has been defined, the device is responsible to retrieve it from Amazon Simple Storage Service. In the next image (Figure 16) the sequence diagram provides the steps required to initialize the application. In this figure the process to find and retrieve the policy document is depicted too.

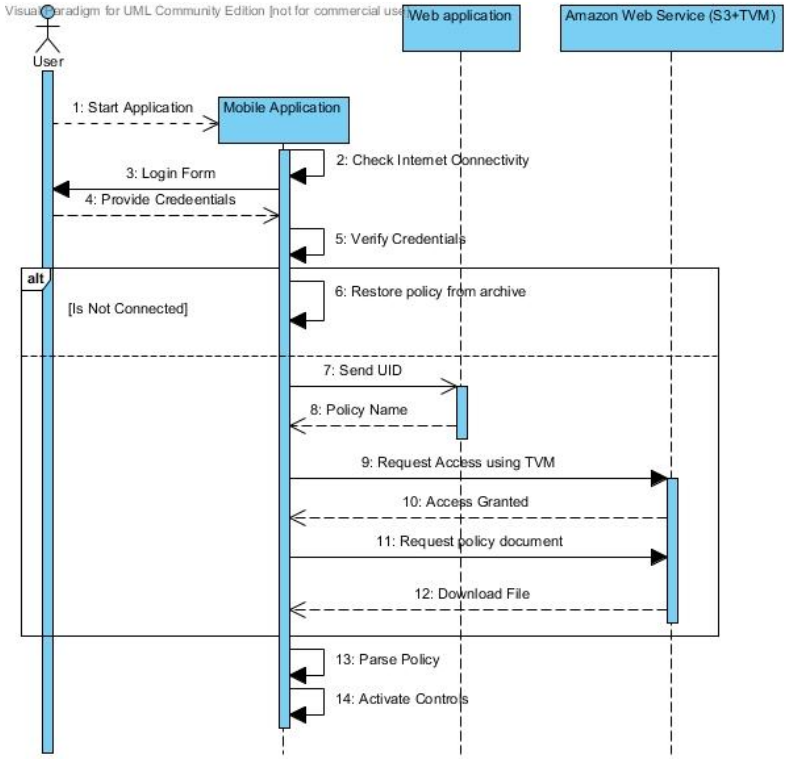

Figure 16 – Application initialization

2. Perform actions based on the elements and attributes declared in the XML

After the document is downloaded to the device, the application first needs to parse the XML document. In case this is not possible, an error message is provided and the latest version that is stored is used. On the other hand, if parsing is possible then the operations that are stated are initiated along with the corresponding attributes that are included. A typical XML document can include both elements and attributes. For the purposes of this project, elements represent different controls or operations while the attributes are used to store specific values that are needed for performing checks i.e. minimum accepted longitude.

Performing operations

A set of operations needed to be implemented in the prototype mobile application. The requirements list includes:

1. Tracking GPS coordinates of the device
2. Displaying GPS coordinates and trail path
3. Update and store policy document in XML format
4. Display active policy
5. Perform data wipe in case of policy violations

6. Restoring user's documents files from Amazon S3
7. Recording vital statistics
8. Uploading files and logs to Amazon S3 as a back-up
9. Encryption and decryption of files stored in the device
10. Display vital information of the device (i.e. battery level, carrier info etc.)

All this operations are contributing in the overall scope of the system; that is to monitor and verify compliance of the device's owner using some of the established practices with regards to security. Of course these operations are subject to change based on the company's preferences and development capabilities.

3.2.2 Web platform

Apart from the mobile application, the system requires a web platform to be created in order to provide easy administration of the policies, devices and users. Since it is neither practical nor appealing to modify manually the code of the application in order to download the necessary XML file, this was a necessary step. The live edition of the platform can be found at: http://konpapadopoulos.kiwedevelopment.eu/thesis/; Figure 17 is providing with compact list of operations performed by the web manager.

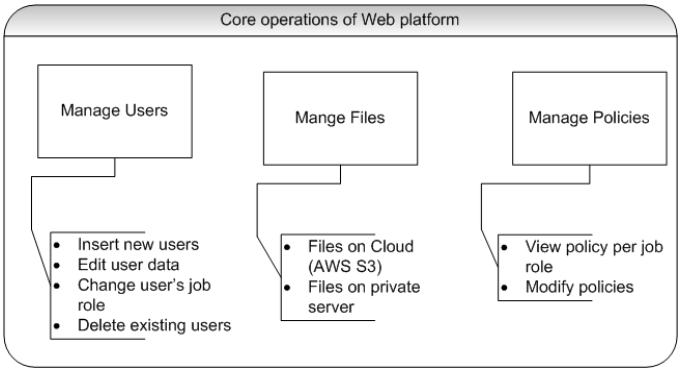

Figure 17 – Web management system, operations overview

A complete list with the required functionality follows:

- A very appealing and interesting landing page that attracts user (presentation purposes).
- Display of all registered users along with their device's UID based on their job roles that have been assigned to them.
- Presentation of all possible characteristics that can be included in the policy.
- Generation of the policy in a user-friendly manner via graphics and simple input forms (i.e. use of Google Maps API).

- Registration of new users along with UID of their device.
- Assignment and modification of job roles.
- Management and modification of existing policies.

Those characteristics were chosen so as to provide an easy way of managing policies and devices. That is because most users don't know and/or don't want to type XML documents, verify their compliance with an XML schema and debug it in case on errors. Using this system, the user has only to worry about setting the correct parameters during the form submission.

While these operations are being performed at the front-end, the system must store and recall all necessary information from a database. For that reason, a typical MySQL database is used which enables common read and write operations to occur.

3.3 Implementation Details / Technologies Used

In this section a more detailed analysis of the two components that were created. Technologies used and frameworks included will be mentioned along with their corresponding documentations and links will be given. The latest version of the core files that are used in the project and presented in the following sections can be found at: https://github.com/papadopoulosk/PyC_project.git

3.3.1 Web platform

As already mentioned, the live version of policy manager web platform can be found at http://konpapadopoulos.kiwedevelopment.eu/thesis/; it is a web-based management system. In order to develop it core web technologies were used. The list includes HTML, CSS, XML, JavaScript and PHP. The latest version of HTML is 5 and the standard is maintained and documented by the World Wide Web Consortium[18]. CSS is used for the presentation purposes of the site; latest version is 3. Again, W3C is responsible for standardizing the technology[19]. XML is another version of the Markup Languages family and it is mainly used for transferring information and storing data that needs to be saved in a machine-friendly format. XML fulfills this purpose. Since it is highly related to HTML, once again W3C is responsible[20]. PHP stands for Hypertext Preprocessor[21] and it is server-side scripting language that is highly correlated with HTML since it is able to dynamically generate HTML content. The version of PHP used was 5.3.13. Finally a database was necessary in order to store

[18] http://www.w3.org/TR/2011/WD-html5-20110525/
[19] http://www.w3.org/Style/CSS/Overview.en.html
[20] http://www.w3.org/XML/
[21] http://php.net/

necessary information and retrieve it at a future time. MySQL[22] 5.0.77 was chosen as is one of the most popular RDBM systems, mainly due to its open source nature.

Furthermore a set of software was used in order to develop and test the system. The editor that greatly assisted in developing the prototype was NetBeans Integrated Development Environment v7.1[23]. NetBeans is a java-based platform that facilitates development for various programming and scripting languages, including HTML and PHP. Additionally the editor is capable of connecting remotely to the RDBMS and uploading automatically files to the server providing a complete environment.

Landing Page

Apart from the tools used, the site is divided in two major parts at the moment this thesis was written. First one is the landing page of the site which, as mentioned before, is used to introduce the system named "Protect your Company – PyC" and provide a nice presentation layer. In Figure 18 a snapshot is displayed that also contained the visualization of the system, also shown at Figure 14. Furthermore, a video presentation is included that is upload at http://www.youtube.com/watch?v=7DOPHVt1bbU.

Figure 18 – Landing page

[22] http://www.mysql.com/
[23] http://netbeans.org/

Management Page

However the most important part of this component is the management page. This is where all major operations occur. They are divided in two parts, asset management and policy handling. In Figure 19 we can see the user interface that is responsible for the first part. Through the options that are available, the administrator is able to modify users easily and quickly.

The second part can be viewed in Figure 20, as we can see there is a list with available parameters that are later used during the generation of the XML document. In order to both produce the visual presentation and perform the necessary operations, two core files are needed: The class file policy.php and the file manager.php. The first one is responsible for handling policies while the second is what the name implies; it is the handler that coordinates all operations. More details about the source code of those files can be found at appendix section 6.4.3.

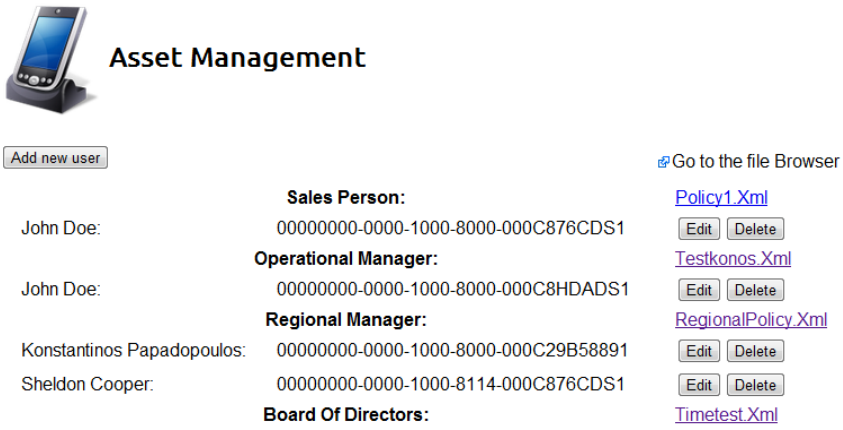

Figure 19 – Policies, users and roles

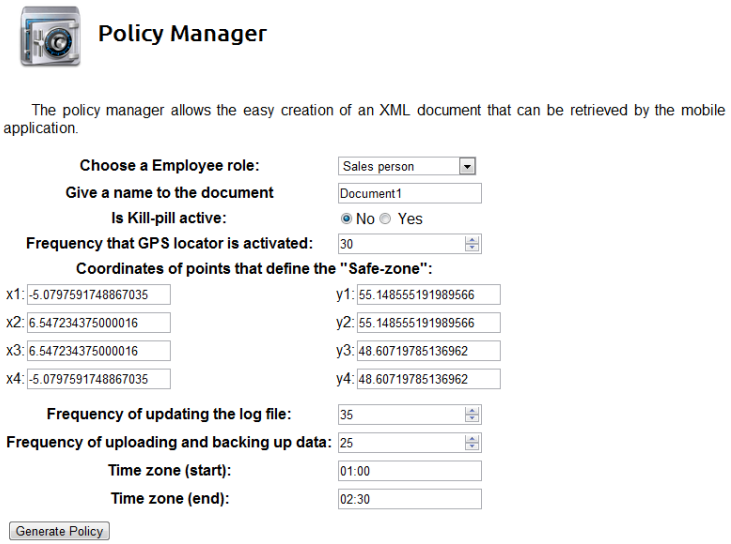

Figure 20 – Policy creation for each role

However the manager section includes an additional functionality that is important in order to be able to set the restrictions about the location of the device, which is the GPS module. Instead of forcing the user providing the desired coordinates that define the "safe" zone and allow accessing the files, the API of Google Maps[24] has been used (Figure 21). The user is able to define the coordinates by dragging a shape and automatically latitude and longitude values are used in the XML file. This is a very user-friendly way to implement a rather complicated and important aspect of the prototype.

Figure 21 – Google Maps

Additionally the possibility is given to the system administrator to view each XML file on his browser by clicking the link next to each job role (Figure 22).

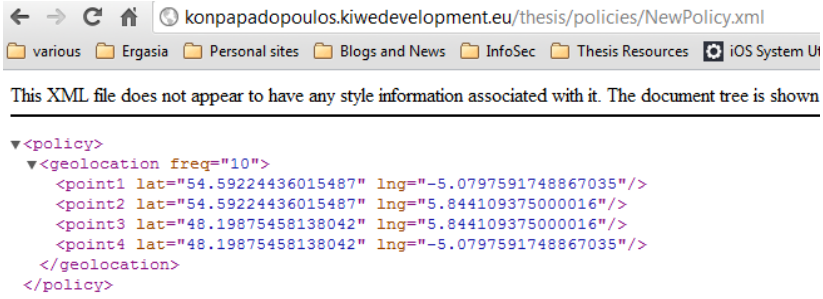

Figure 22 – Viewing a policy

Even though a policy might not contain all elements since it is not mandatory to check and provide values to every field, an example below exists that includes all possible restrictions and values. As we can see, a full document contains initially the date that it was created along with the role attribute with corresponds to the role that the owner of the device possesses inside the company. Next follow the GPS restrictions. In order to include the minimum required information only the coordinates of two points are defined, those are the North-East

[24] https://developers.google.com/maps/

and South-West points of the square area. Using those coordinates, the square "safe zone" can be manipulated by the mobile application. Furthermore the geolocation element is accompanied by the "freq" attribute that its purpose is to instruct the application how often to check the location of the device. Next follows the "kill-pill" which in practice means that by the moment the device downloads the policy, all files need to be deleted. The rest of the elements become obsolete when this restriction is applied. The following two elements provide information about the frequency of updating the log and backing-up to the server the files of the device. The last restriction that the XML file introduces is the time frame. Since not all employees should be able to access corporate documents throughout the day, based on the role different time frames are provided. For example an employee that is registered to be working from 8.00am to 4.00pm, should be only have access during those hours.

```
<policy created="August 16, 2012, 7:06 pm" role="3">
        <geolocation SWlat="49.64244842513996" SWlng="-5.0797591748867035"
                NElat="54.59224436015487" NElng="3.2073906250000164" freq="12"/>
        <killpill status="1"/>
        <logfreq value="30"/>
        <uploadfreq value="90"/>
        <timeframe start="08:00" end="16:00"/>
</policy>
```

Next, AJAX[25] (Asynchronous JavaScript And XHTML) technology and the JQuery library have been used to perform minor operations.

```
$.ajax({
        type: "POST",
        data: "delusr=" + 10,
        url: "manager.php",
        success: function(msg){
        $("#rowID10").hide("slow");
        }
});
```

AJAX is used to perform operations related to databases (select, insert, delete or update statements) without refreshing the web page. This way the visitor/user of the specific web site can have better navigation experience without having to load new pages all the time. AJAX is implemented during the delete operation of users in the Policy Manager page. Even though AJAX uses simple routines written in JavaScript, JQuery provides an AJAX API[26] for easy manipulation and execution. The last example of AJAX is sending using POST method[27] to the script "manager.php" including the variable 10 under the name "delusr". Using this code is similar to saying that the ID of the user that is going to be deleted is 10 and it is stored in a variable named "delusr" in order to send it to manager.php. Finally, if the operation is

[25] http://en.wikipedia.org/wiki/Ajax_(programming)
[26] http://api.jquery.com/jQuery.ajax/
[27] POST method definition: http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html

completed successfully a JQuery function is used in order to “hide slowly” the element that has ID equal to “rowID10”.

Finally a plug-in[28] has been used to provide a web-based interface (Figure 23) so as the administrator of the system will be able to manage the files that users upload. However personal files of the users are encrypted (more details in the mobile application section about encryption) so they are un-viewable. Additionally the admin can distribute new files to employees based on their job roles. Using this operation, distribution of documents becomes much easier and secure.

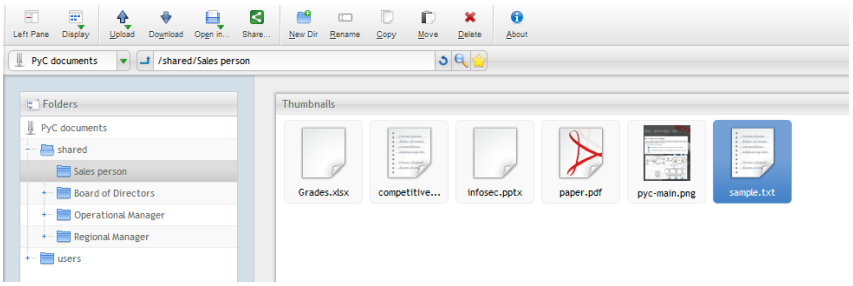

Figure 23 – Web-based file browser

The last functionality of the web manager is a direct link to the Amazon Simple Storage Service console. That practically is a web-based file browser (Figure 24) that allows manipulating all files created or uploaded to the cloud service and is added in order to replace the file browser of the private server.

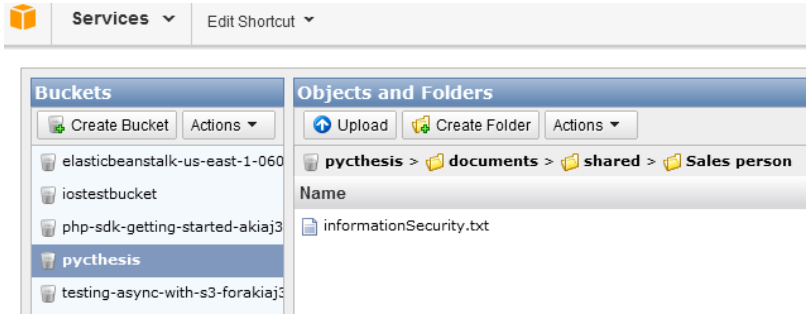

Figure 24 – Amazon S3 console

3.3.2 Mobile Application

In order to create the mobile software required to perform the operations certain bibliography was heavily studied (Dalrymple, et al., 2009), (Kochan, 2011) and (Conway, et al., 2012). Furthermore there was extensive assistance and help from online forums and communities[29]. Core technology that was used was Objective-C programming language; a superset of the C language. The development phase included the use of software tools that are

[28] http://encode-explorer.siineiolekala.net/
[29] http://stackoverflow.com/, http://www.codeproject.com/

distributed by Apple. More specifically, the version of the Software Development Kit (SDK) was iOS 5.1[30] (at the moment Apple is ready to release version 6), the operating system was Mac OS X Lion[31] ver.10.7.3 and the editor used was XCode 4.3.3[32]. The application was developed and tested on the iPhone 5.1 simulator that the SDK ships with. For that reason there were limitations in the functionalities that were developed; jail breaking[33] and the use of closed API[34] was not possible.

The developed application is using various frameworks that Apple is providing along with additional libraries that are available to use without charge. Those packages greatly assisted in implementing the desired functionality. A presentation follows that explains step by step how each operation out of those mentioned in the relative requirements list was implemented. The application is supposed to be installed on the mobile device each employee has. Furthermore, the employee should be registered in the web management system in order to be able to download the XML file and be able to perform successfully all operations that the app provides.

Overview

A detailed class diagram can be found at Figure 25. It is depicting the most important classes in the mobile application displaying the relationships that exist between them along with the role, i.e. class, framework or protocol. Since this is a rather complicated and meaningless figure to the average reader, the analysis will focus on the major functionalities that are related to the aim of this prototype: develop a system that enables the implementation of an access control model that enforces a certain policy.

Furthermore, in the appendix (sections 6.4.1 & 6.4.2) UML diagrams are provided and help the reader get a deeper understanding of the processes that occur and the way they collaborate. This is rather important point so as to eliminate any black boxes that the reader might have. The first diagram is a system-wide use case diagram that displays most of the actions and operations that occur between the components along with important internal functions. Also two sequence diagrams are available depicting the "Restore files" and "Checking violations" operations.

[30] http://www.apple.com/ios/
[31] http://www.apple.com/osx/
[32] https://developer.apple.com/xcode/
[33] http://en.wikipedia.org/wiki/IOS_jailbreaking
[34] Apple does not approve the use of low-level API such as file system management, call and SMS handling.

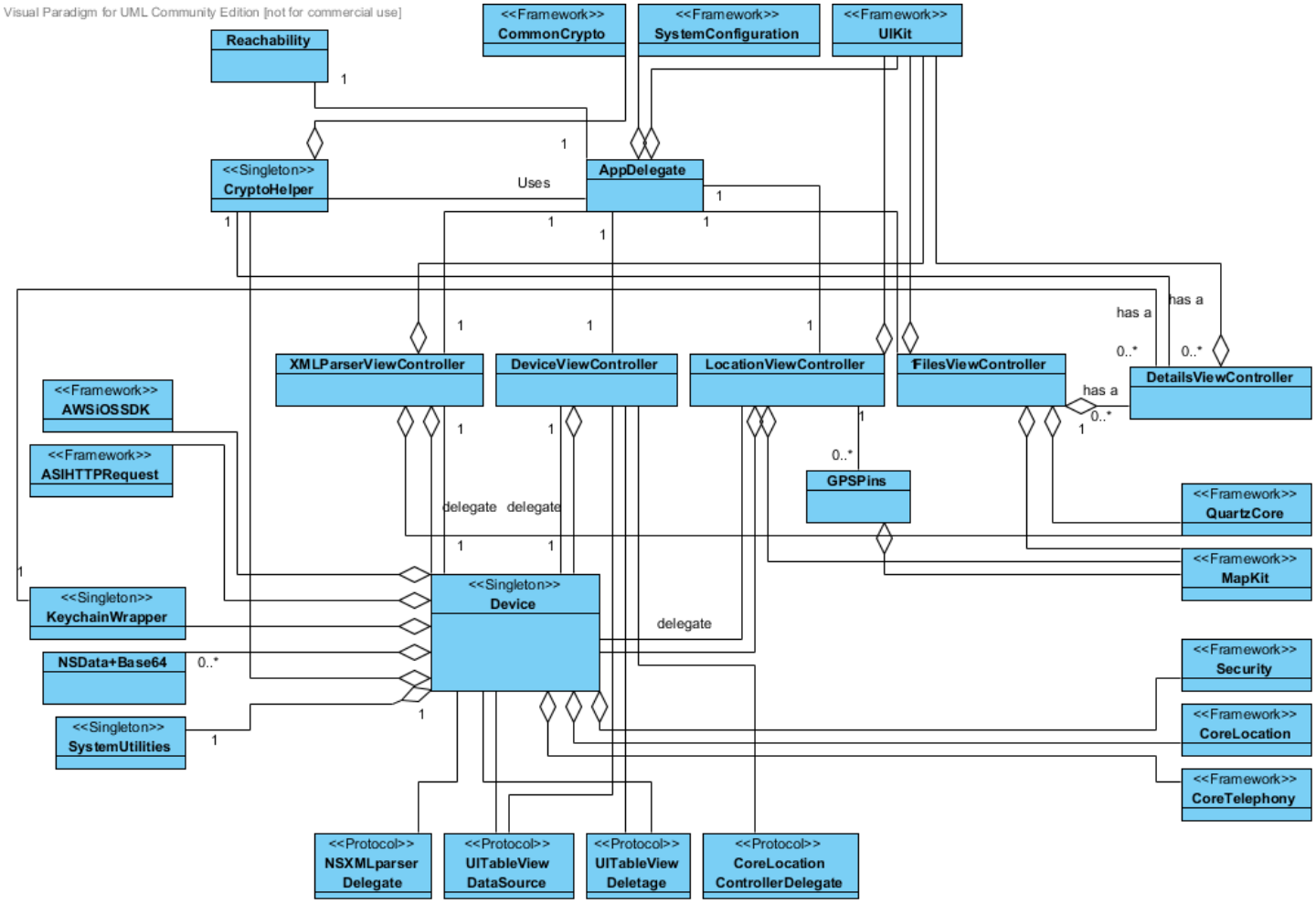

Figure 25 – Classes and frameworks diagram overview

The application is composed of the following major classes:

- AppDelegate
- Device
- XMLParserViewController
- DeviceViewController
- LocationViewController
- FilesViewController
- DetailsViewController
- KeychainWrapper
- Reachability
- CryptoHelper

These are not the only classes of the system, nevertheless they implement most of the basic functionality and they are responsible for handing and manipulating the rest of the classes. Furthermore, during the development phase three major software design patterns

were used, delegation[35], singleton class[36] and Model-View-Controller (MVC[37]). Delegation and MVC are heavily used in both iOS and MAC OS development.

Delegation design pattern is based on the concept of transferring the duty of performing an operation to a different object. In order for this pattern to work properly (in iOS) usually various protocols are being declared. They are used as a guide to what methods a delegate object should implement. If a class wants to act as a delegate, then it is necessary to conform to this protocol by defining those methods. For example in Figure 26 UIScrollViewDelegate is defined as the delegate of UIScrollView. The latter is storing the memory address of his delegate in order to send messages to it. In case an event occurs or an operation starts/finishes (Figure 27) then UIScrollView invokes the necessary method of UIScrollViewDelegate. Most likely this method decleration is predefined in the corresponding protocol. At that point, the delegate is able to perform other operations based on the returned results. If delegation hadn't been used in this case, then UIScrollView had to implement all operations, in this example "Download next image" as shown in figure below.

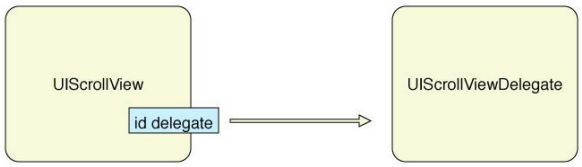

Figure 26 – Delegation design pattern, initialization[38]

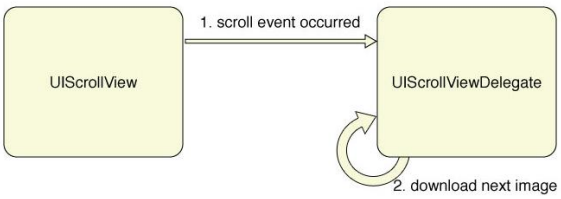

Figure 27 – Delegation design pattern, event[38]

MVC model is a very powerful technic that allows better management, testing, debugging and maintenance of software products. The main idea behind MVC (Figure 28) is that different objects should be responsible for different operations. Therefore, as the name implies, a separate class should be responsible for storing the information of a table (Model), another class should be managing the way that the table is being presented (View) and a last class must be the manager that is handling all operations (Controller).

[35] http://developer.apple.com/library/ios/#DOCUMENTATION/General/Conceptual/DevPedia-CocoaCore/Delegation.html
[36] http://en.wikipedia.org/wiki/Singleton_pattern
[37] https://developer.apple.com/library/mac/#documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html
[38] Source: http://thinkvitamin.com/code/ios/ios-design-patterns-delegation-part-2/

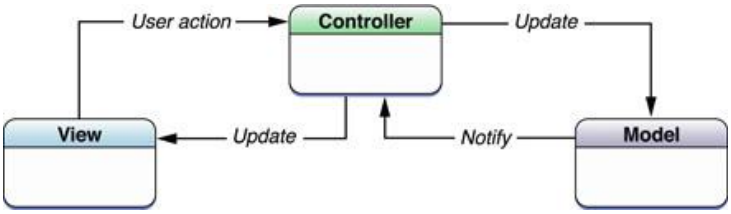

Figure 28 – Model View Controller design pattern[39]

Singleton design pattern lies on the fact that a class should have only one instance created and no more. In case an object tries to initialize a new instance of that class, then the implementation should return the already existing object. Using this pattern, no matter how many times we try to create new instances, every occurrence will refer to exactly the same object. This is useful in cases that the developer wants to keep the consistency of the project and removing the overhead of synchronizing many objects that actually refer to the same entity.

An analysis of all major classes follows. The extension .h/.m refers to the header file (.h) that the class is declared and the implementation file (.m) on which each method is defined. This is a standard naming convention used in Objective-C programming language.

AppDelegate.h/.m

This is the first class that is called by default during the initialization of the application. As the name implies it is a delegate that is responsible for managing the way the application is supposed to run. During the execution all other instances are being created. This includes the rest of the classes that follow.

Initially the AppDelegate is performing an identity check of the user. Each time the application is launched or resumed, the user is prompted to provide a password (Figure 29).

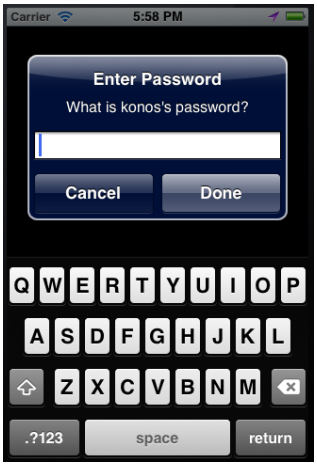

Figure 29 – Login screen

[39] Source: https://developer.apple.com/

The password is cross-checked with the already stored credentials. Those credentials were established during the first execution of the application. The credentials are stored in the keychain of the iOS as a hash value. The value that the user provides is hashed and compared to the existing value. In case of a successful authentication the navigation continues to the actual content of the app. Otherwise the user is requested to re-submit his password. At the moment there is no max limit of attempts.

A new UITabBarController[40] is initialized (the tabs that appear in the application screen at the bottom) and four views are created, one for each tab, and initialized with the desired values. Then they are added to the main tab bar controller. Each tab is displaying the corresponding information that is provided by the ViewController classes.

```
UIViewController *xmlvc =     [[XMLparserViewController alloc] init];
UIViewController *devicevc =  [[DeviceViewController alloc] init];
UIViewController *locationvc= [[LocationViewController alloc] init];
UIViewController *filesvc = [[FilesViewController alloc]
initWithNavigationBar:navBar];
```

One more important task that AppDelegate is performing is checking constantly the network status (connected to the Internet or not) in order to perform updates of the policy. This operation is performed using the Reachability class[41], which is a similar to the official one Apple is providing. Every time the network status changes, Reachability class is submitting a note to the NSNotificationCenter[42]. The AppDelegate is designed to "subscribe" to this center listening to specific notes and being able to receive those changes in connectivity.

Device.h/.m

This is a model class that its main purpose is to keep information related to the physical mobile device. This includes the UID of the device, IP address, battery levels etc. In order to be able to provide the same information regarding the device, this class has been designed to comply with the singleton design pattern. Therefore every time a new object is created, it contains the same data throughout the application. This particularly useful in cases in which different parts of the application want to retrieve some data related to the device such as the carrier name. All it has to be done is create and initialize a Device object. Some core information that is contains can be seen below.

```
[information setObject:[SystemUtilities getUniqueIdentifier] forKey:@"Unique Identifier"];
[information setObject:[SystemUtilities getSystemUptime] forKey:@"System Uptime"];
[information setObject:[SystemUtilities getModel] forKey:@"Model"];
```

[40] http://developer.apple.com/library/ios/#DOCUMENTATION/UIKit/Reference/UITabBarController_Class/Reference/Reference.html
[41] https://github.com/tonymillion/Reachability/
[42] https://developer.apple.com/library/mac/#documentation/Cocoa/Reference/Foundation/Classes/nsnotificationcenter_Class/Reference/Reference.html

```
[information setObject:[SystemUtilities getIPAddress] forKey:@"IP Address"];
[information setObject:[SystemUtilities netmaskForWifi] forKey:@"Netmask For Wifi"];
[information setObject:[SystemUtilities getBatteryLevelInfo] forKey:@"Battery Level Info"];
[information setObject:[NSString stringWithFormat:@"%f", self.locMgr.location.coordinate.latitude]
forKey:@"Latitude"];
[information setObject:[NSString stringWithFormat:@"%f", self.locMgr.location.coordinate.longitude]
forKey:@"Longitute"];
[information setObject:[SystemUtilities getRealDeviceType ] forKey:@"Real device type"];
[information setObject:[NSString stringWithString:@"Vodafone"] forKey:@"Carrier name"];
```

This is a part of the source code that displays the process of retrieving some of the vital information of the device using the SystemUtilities class. The latter is a class that was developed by Merkelsoft Inc.[43] and was obtained without charge from Binpress[44]. The reason SystemUtilities was used is that it is designed as a wrapper that provides a number of user-friendly methods in order to obtain information about the battery, the carrier, network, CPU, memory etc. Next this information is stored at an NSMutableDictionary[45] using an appropriate key. The "information" dictionary can be later retrieved by any object that creates an instance of the Device class. For example, DeviceViewController is retrieving the data of this dictionary and displays it in a table-like format.

Another important operation of the Device is to act as a delegate of the location manager that is responsible for handling the GPS of the phone. Each time the coordinates change, a message is send to the Device and the information contained in the "information" dictionary is updated. This is being performed so as all other objects that need this piece of information will be able to receive updated data. Additional tasks include keeping log of the vital stats of the device into a file, upload the log and the user's documents to the web server, restore files upon request and request from the web server the policy document that is associated with the device. In order to perform all necessary requests to the web platform, two major options exist: either use NSURLRequest or CFNetwork. During the implementation of the application, the second option was used by integrating a free library which is taking advantage of the CFNetwork framework. ASIHTTPRequest[46] is a popular wrapper that helps perform HTTP requests to a web server. As a result, the implementation process became much faster and developer-friendly. Furthermore there is a set of methods that enables the manipulation of the responses, such as files that were downloaded. Generally, the Device class is responsible for all operations than include sending and receiving data from the web server.

[43] http://www.markelsoft.com/
[44] http://www.binpress.com/app/ios-system-utilities/908
[45] https://developer.apple.com/library/mac/#documentation/cocoa/reference/foundation/classes/nsmutabledictionary_class/Reference/Reference.html
[46] http://allseeing-i.com/ASIHTTPRequest/

The last core operation that is performed by the Device class is the encryption of the shared files that are downloaded from the web application. As it has already been mentioned, the web application is allowing the administrator to distribute files to employees based on their job role. However these files are unencrypted. In order to secure the assets that are stored to the phone symmetric encryption is implemented. Even though the actual encryption process is handled by the CryptoHelper class, Device.h is invoking the methods each time a file is downloaded from the "shared" folder of the server. The encryption algorithm used is Advanced Encryption Standard (AES) using 128 bits as encryption block that operates in cipher-block chaining mode. AES-128 is considered secure enough for the time being (Seagate Technology LLC, 2008).

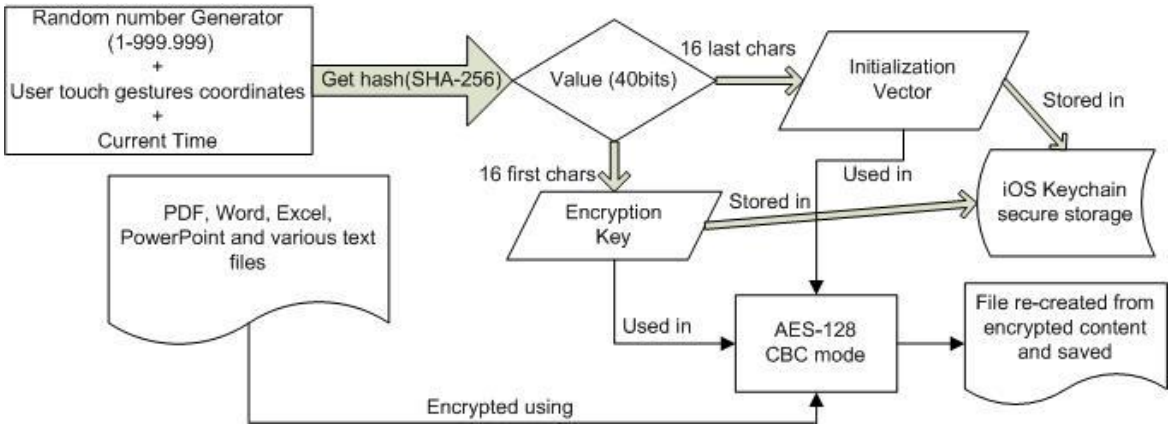

Figure 30 – Encryption process

In order to start the encryption process the key and an initialization vector is required. Every time an encryption process is about to start, a long string is generated. Its value is the concatenation of the current time plus the value of a random number generator and the coordinates of any possible touch gestures the user performed. The final string value is hashed using SHA-256. The output value is forty characters long. Then the sixteen first characters are used as encryption key while the last sixteen are used as initialization vector. Then the file is encrypted symmetrically using the generated key and IV. After the process is complete, both the key and the IV are stored into the keychain of the iOS. The next time, the encrypted documented is required to be accessed, therefore decrypted, the file name is used as a search parameter and the key and the IV are retrieved and the file decrypted.

XMLParserViewController.h/.m

The next class is a view class; XMLParserViewController that has as main responsibility to present in a meaningful and useful manner the XML policy document. In

order to do so during the initialization method, a pointer is created that refers to the unique instance of the Device class. Therefore, each time an update on the policy is requested an appropriate method of the Device is invoked and the result is return to the XMLParserViewController. At that point the policy is stored in a file from which it can be retrieved on demand or in case of no Internet connection.

An additional key operation of the class is the actual parsing of the XML that is downloaded. There are various solutions[47] on how to parse an XML document in iOS. However, the developer has chosen to integrate the official parser[48] provided by the SDK since compatibility issues and bugs will never arise as problems. Again delegation has been used and our class conforms to the NSXMLParserDelegate protocol and acts as a delegate of an NSXMLParser instance. That instance is performing the actual task of parsing while the results are passed back to the delegate. Various messages are sent back each time new elements, attributes, end of elements, end of parsing and error events in parsing occur. In general, NSXMLParser is a valuable and easy to implement class.

After the parsing is complete and all information is parsed, the class is responsible to present the outcome in a meaningful and user-friendly manner to the user of the application (Figure 31). Each rule of the policy that is enforced is presented in different table section along with the corresponding values. For example in the following figure, the allowed time frame to access the files is between 8:00am and 4:00pm.

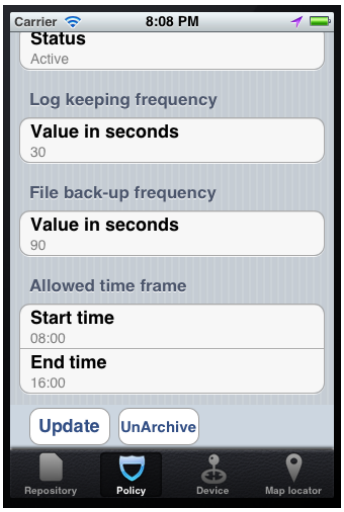

Figure 31 – Policy viewer

[47] http://www.raywenderlich.com/553/how-to-chose-the-best-xml-parser-for-your-iphone-project

[48] https://developer.apple.com/library/mac/#documentation/Cocoa/Reference/Foundation/Classes/NSXMLParser_Class/Reference/Reference.html

DeviceViewController.h/.m

After the application has been initialized and the Device class has filled the Information dictionary with the appropriate data, a presentation is necessary to occur. This role is fulfilled by DeviceViewController. The class is using a table-like view (UITableView[49]) to create the desired output (Figure 32). This view offers a wide range of effects and methods that are inherited by the UITableView, such as sliding the interface up or down. Furthermore in the same figure additional operations exist purely for demonstration purposes that perform logging and uploading the log to the web server. These events can be triggered either on schedule based on the security policy that was downloaded or on user's demand by clicking the corresponding buttons.

Figure 32 – Table view of DeviceViewController

LocationViewController.h/.m

The next class that needs to be analyzed is responsible for providing the user information regarding his position (Figure 33). This is important because some of the major parameters that a policy includes are the latitude and longitude that define a "safe zone". Inside that safe zone the user can access his documents, otherwise they are deleted. Only after he has entered back inside the safe zone, he is able to view his documents again. The iOS 5.1 SDK includes a framework that allows the use of Google Maps API for the iPhone. Map Kit framework[50] is very easy to implement and manage. As it is depicted in the code example below, each time the location coordinates are updated, the new latitude and longitude values are retrieved from the Device class since that is the storage repository. Then those coordinates are stored in "zoomLocation", a variable that will next define the new location in the map. After all parameters have been submitted, the "_mapView" updates the region that is displays.

[49] http://developer.apple.com/library/IOS/#documentation/UIKit/Reference/UITableView_Class/Reference/Reference.html
[50] http://developer.apple.com/library/ios/#documentation/MapKit/Reference/MapKit_Framework_Reference/_index.html

```
zoomLocation.latitude = [myphone getLat];
zoomLocation.longitude= [myphone getLong];
MKCoordinateRegion viewRegion = MKCoordinateRegionMakeWithDistance (zoomLocation,
0.5*METERS_PER_MILE, 0.5*METERS_PER_MILE);
MKCoordinateRegion adjustedRegion = [_mapView regionThatFits:viewRegion];
[_mapView setRegion:adjustedRegion animated:YES];
```

Additional details such as the transition effect from one region to another can be also defined. Finally in the last figure, displays the possibility of keeping track of the path that the device follows on the map. Pins are representing the location of the user the last time the GPS controller was activated and the coordinates were recorded.

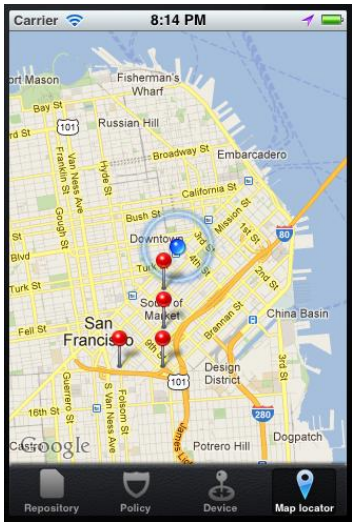

Figure 33 – GPS locator of LocationViewController

FilesViewController.h/m

The next important class is used to provide a complete user experience. It assists in displaying all stored documents that the user/employee has stored in his mobile device. That way he can constantly be aware about his access rights status. In case he is eligible to download the documents; the list displays all files in a table format (Figure 34). Otherwise, if the user is violating the restrictions set by the policy downloaded, the table is empty since the files are deleted from the device. Furthermore, the user is able to delete existing files, create new entries or modify existing text files. Documents that are in PDF or PowerPoint format are only eligible to be displayed without any modification being possible since iOS does not provide a native solution to edit such types of documents.

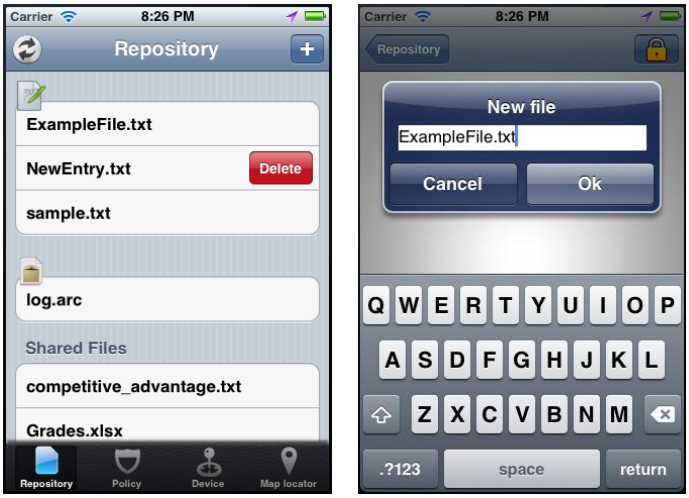

Figure 34 – Files stored in the device / New file creation

The source code example below is depicting the steps required to create a dictionary containing all information about the stored files and Figure 35 is providing a better conceptual understanding of the structure. A dictionary is a structure that allows storage of information in key-value pairs. Therefore if an item is stored under the key "car", in order to obtain it back the key "car" is used.

```
NSString *rootPath = [NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES)
objectAtIndex:0];
NSString *filePath = [NSString stringWithString:[rootPath stringByAppendingString:@"/Files/"]];
NSFileManager *fileManager = [NSFileManager defaultManager];
NSArray *arrayFromFile = [fileManager contentsOfDirectoryAtPath:filePath error:nil];
for (NSMutableString *file in arrayFromFile){
    if ( [fileTypes valueForKey:[[file lastPathComponent] pathExtension]]==nil){
      NSMutableArray *specificFileType = [[NSMutableArray arrayWithObject:file] retain];
      [fileTypes setObject:specificFileType forKey:[[file lastPathComponent] pathExtension]];
    } else {
      [[fileTypes objectForKey:[[file lastPathComponent] pathExtension]] addObject:file];
    }
  }
```

In order to get the files and be able to use them a certain format is necessary. This requirement is defined by class that constructs the table (UITableVIew[51]). The solution employed to solve the problem is as: First the application is scanning the designated folder file by file. Each time a file is being processed, the file extension (i.e .txt) is checked. The loop checks whether or not the specific extension is already used as a key in the "FilesTypes" dictionary. If the key exists, the value of that key is retrieved. Since objective-C is using pointers to store variables, tables etc., the value that is retrieved is the address of an

[51] http://developer.apple.com/library/IOS/#documentation/UIKit/Reference/UITableView_Class/Reference/Reference.html

NSMutableArray (dynamic array) that is containing the names of files under the specific extension.

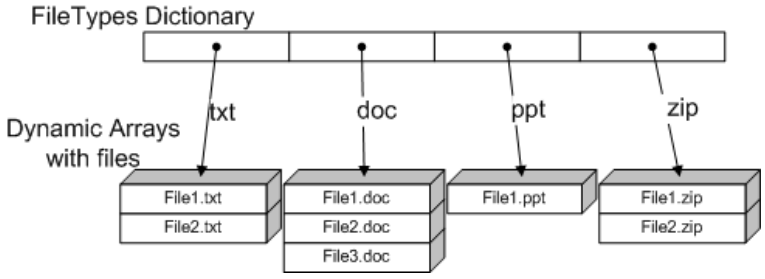

Figure 35 – Conceptual structure of files

On the other hand, if the key (extension) does not exist in the dictionary, a new array is created that will be storing the names of the files that correspond to this file type. Furthermore, the pointer of the newly created array is kept at the dictionary under this key.

Finally, FilesViewController is responsible for the initialization of DetailsViewController object that is being analyzed next.

DetailsViewController.h/.m

This class handles the way that each file is being displayed in the screen of the device. The outcome depends on the type of file. If the file has extension that refers to plain text such as .txt, .css or .js then the file is presented inside a text editor that can be unlocked, edited and then saved. On the other hand, if the file is "static" such as PDF, as mentioned before, can be only viewed. However, the important aspect of this class is that every time an edited file is closed; the file is encrypted again using the methodology presented at Device.h/.m section. Once again new encryption key and initialization vector are created, used and stored in the keychain while the old values (if any) are discarded.

KeychainWrapper.h/.m

Even though this wrapper was not developed by the author of this thesis, it assists greatly in performing important tasks in the mobile application. As the name indicates, it acts as a helper class that provides an easy-to-use interface in order to perform operations related to the iOS Keychain.

Reachability.h/.m

This is a modification of the official Reachability class that apple provides in order to check constantly the connectivity of the mobile device. It was taken as-is and it is being mentioned due to the nature of the task it performs.

CryptoHelper.h/.m

In order to perform all encryption operations, the use of Common Crypto framework is required. It provides a number of functions that are responsible for various encryption processes such as both symmetric and asymmetric encryption or decryption of files, key generation. There is also the capability to choose different modes of encryption such as Electronic codebook (ECB), Cipher-block chain (CBC) etc. In fact it is a full encryption framework. However the drawback is that the functions and the API required good knowledge of the C programming language. In order to overcome this obstacle, a wrapper is used and it is modified according to the needs of the system.

3.3.3 Frameworks used

During the development process various frameworks have been used that allow implementing various operations and using a number of classes. For completeness purposes an overview follows even though some of them have been already mentioned in previous analysis. Detailed information about all of them can be found at http://developer.apple.com/library/ios/navigation/.

- **SystemConfiguration** provides the interface that allows checking the internet connectivity and examining if remote hosts are reachable. Reachability class is using it.
- **MapKit** is necessary to implement the functionality that calculates and displays the current position of the user on a map that LocationViewController is responsible for. It uses services offered by Google Maps API.
- **CoreLocation** is necessary to manager the GPS hardware controller of the device. Input parameters are defining the way that the controller will operate and report back its results. The results are then processed based on the application requirements. Some of them feed the MapKit back with information so the current location can be displayed on a map.
- **CoreTelephony** is responsible to perform operations related to calls, carriers and cellular networks. For example it possible to obtain the name of the carrier network or call information. However since the application was tested only on a simulator, these operations were simulated rather than actually tested.
- **UIKit** is one of the most important frameworks since it is including all classes and protocols that are responsible for the creation and management of the User Interface in an iOS application.
- **Foundation** is a core part in iOS development. It includes all the classes that allow the use basic datatypes and protocols (i.e. NSArray, NSData). Without it, it is not

possible to implement basic types and operations that are related with the application logic.

- **CoreGraphics & QuartzCore** frameworks enable the implementation of various effects and graphics that enhance the user experience in the iOS. Example of such an effect is the rotation effect the refresh button in the Files tab.
- **CFNetwork** is a low-level API that allows the implementation of network connection tasks such communicating over HTTP or FTP. CFNetwork is mandatory in using the ASIHTTPRequest framework that acts as a high-level wrapper.
- **MobileCoreServices** includes the definitions of various Uniform Types Identifiers. This refers to the various types of files along with their extensions that are used and allows an easy manipulation and processing of objects such as files and directories.
- **CommonCrypto** is the library that Apple uses as the official encryption package that is able to perform all kind of encryption-related operations. It is not very user-friendly and requires knowledge of the C programming language.
- **Security** includes basic type declaration of structures and data types that are used in interacting with the Keychain, encryption operations etc.
- **AWSiOSSDK** It is the officially provided framework by Amazon in order to use the API for the Amazon Web Services. It has been integrated into the application so as to be able to access the cloud. Without it, it would not be possible to both upload and download files that are stored in Simple Storage Service (S3).

3.3.4 Encryption Overview

Even though all parts of the encryption and decryption that occur during the execution of the application have been referred in previous sections, it is considered wise to provide a complete overview to assist the reader understand better this important aspect of the system.

First of all, the type of encryption applied to files and documents is the symmetric algorithm AES-128 (advanced encryption system 128 bits per block of encryption). The security of this algorithm has already been established and is considered more than adequate (Garfinkel, 2012). The mode that the algorithm is operating is CBC and the length of key and initialization vector (IV) is 16 characters long. In order to provide a secure encryption procedure, keys and IVs are produced using very high randomness coefficient. Their values depend on three independent factors that increase the complexity and the randomness. First is the current time that the file is being encrypted, second is various touch gestures of the users during the edit period of the files and third is a value provided by a random number generator.

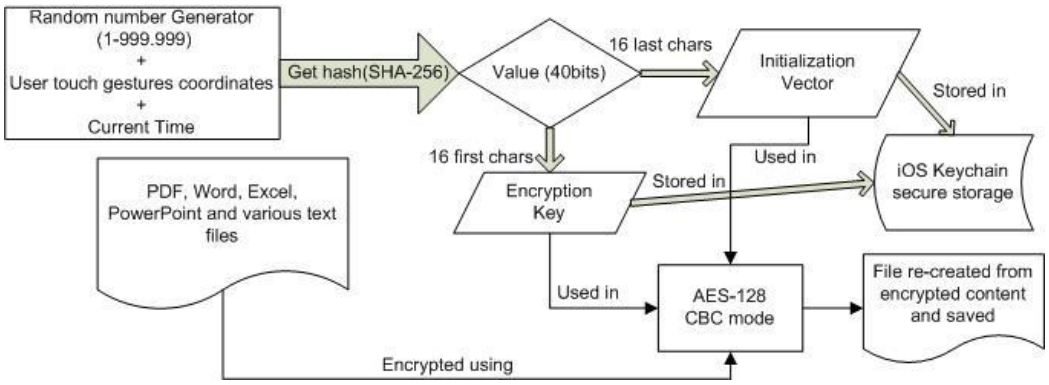

Figure 36 – Encryption overview

All together produce a string value that is then hashed using SHA-256. The first sixteen characters are used as key and the next sixteen as IV. Finally those values are stored in the iOS keychain so as to be able to retrieve them upon decryption. At this point the encryption can start. A final note, the first time a file that is unencrypted is downloaded to the device, the gesture parameter is ignored since there is no user interaction at that point.

3.3.5 Amazon Simple Storage Service (S3)

In order to deliver a professional system that is able to offer modern services to customers, the integration of cloud services has been considered imperative. Modern cloud operators offer a variety of services, packages and platforms in order to integrate them in business solutions. Amazon was chosen to fulfill this role due to the ease-of-use of its console and provided SKDs. The activation of the account was fast and requires a credit or debit card. However there are no charges for one year, if data transfer and HTTP requests remain below a certain threshold.

The first step that is required to take is creating an account at http://aws.amazon.com/. After the account has been created and credit card details have been provided, the user can activate various services by navigating to https://console.aws.amazon.com/console/home which is the console for managing all available services from Amazon. For this project two different services are required:

- Simple Storage Service (S3)
- Elastic Beanstalk

The first is storage service that allows managing files, uploading, downloading, distributing etc. by providing a simple and easy to use web interface (Figure 37). However the web console is not the optimum way to integrate S3 in the project. Therefore the appropriate SDKs that provide a programming API were used. Since the project is a two-part system (web

and mobile application) two different SDKs are included. The iOS SDK[52] is full framework with classes and protocols that greatly assists in communicating with the cloud. Similarly there is an SDK to use with PHP[53] so as to cover the requirements of the web management system of the project.

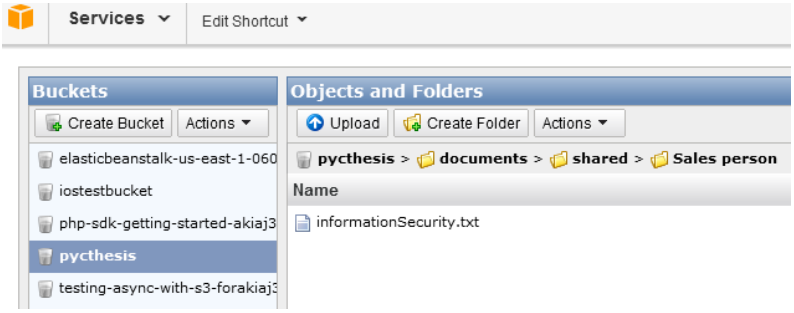

Figure 37 – S3 Console manager

The SDK that is related to PHP is offering similar functionalities with the iOS one. However due to the nature of the mobile devices, it is not safe to store credentials in order to connect to cloud. The reason is that we cannot be certain about the environment of the application and device. It might fall to malicious user that will reverse engineer the app and acquire the username and password of the cloud service. This is where the second service of Amazon, Elastic Beanstalk, is performing its duties.

In general Elastic Beanstalk is a cloud service that allows to runs various applications of different programming languages (PHP, .NET, Python etc.) without the need to worry about infrastructure and performance. In our case the application that is being executed is java-based and officially provided by Amazon. It's called Token Vending Machine (TVM)[54] and acts as an intermediate between the mobile application and the S3 service. The TVM is configured to work with S3 and create temporary credentials which mobile applications and can use and connect with the cloud. The credentials expire after a predefined time period, usually 12 hours. Then the mobile application can refresh the credentials. Using this application an extra layer of security is added to the system. Finally, Amazon is also offering encryption of files at rest so as to ensure that files cannot be accessed by illegitimate users. Furthermore the possibility of SSL is present and constant monitoring and logging settings exist.

[52] http://docs.amazonwebservices.com/mobile/sdkforios/gsg/Welcome.html
[53] http://aws.amazon.com/php/
[54] http://aws.amazon.com/code/8872061742402990

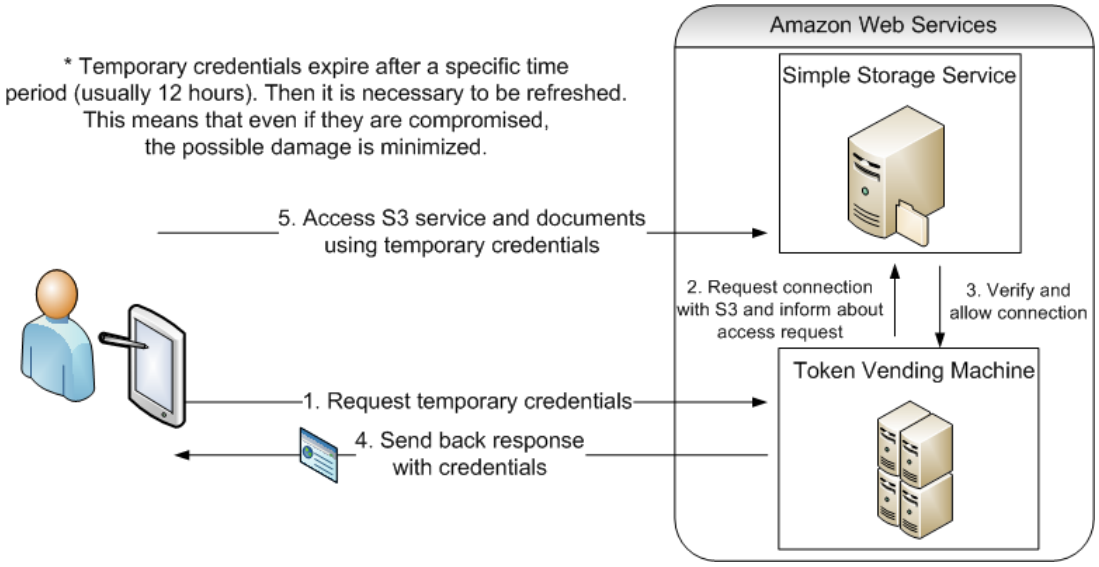

Figure 38 – Amazon Web Service integration

3.4 Future Proposals and recommendations

The prototype system is able to implement and demonstrate core operations of a functional system that is assisting in corporate compliance with regards to policies that are distributed from a central authority. The most important operations refer to finding the appropriate policy, downloading and enforcing it. The way this is done depends on the implementation of the system and a matter of personal taste or experience. In this system XML format has been chosen as a policy document type. JSON could also have been used. However in order to offer much better quality of service to a company that desires to implement such a system, the operations of the mobile application need to be extended.

Mobile Application

The restrictions that currently apply to the system are mostly focused on GPS readings and time frames while extra parameters are included such as the frequency of uploading files to the server. There are pretty useful and cover a significant amount of requirements but on the other side they are trivial and do not offer great flexibility. Up until now there was no specific mention on how to manage the complete file system, Bluetooth services, SMS and call handling or operations related to the Internet. The main reason for the lack of such mention is the restrictions applied by Apple itself. While working in the environment of the iPhone simulator there is no way to access private API that is responsible for such services. Even on normal devices jail breaking is required to do so.

Taking as granted that a jail broken device is available, there is a wide range of possibilities on how to further improve the system. The way to jailbreak a device will not be discussed here since there are a number of solutions available online and it is not inside the

scope of the thesis to do so. As a starting point the application could be transformed into a background daemon that runs seamlessly on the device without the user being aware of it. The daemon will be able to be initialized on start-up of the device. There is a full documentation[55] and a technical report[56] provided by Apple that explains what a daemon is, differences and similarities with agents etc. It can be used as a starting point to implement a daemon. Additionally a blog was found[57] and according to the author of this thesis is believed to be one of the single best guides that it was discovered during the research and explains how to implement a daemon. Furthermore daemons and agents require the creation of certain property list (.plist) files that are read during start-up of the O.S. and contain parameters regarding run-time execution. Property list files are in fact flat XML documents that Apple uses for various reasons. By defining the appropriate settings[58] in a .plist file the execution of the background program can be greatly altered. For example one of the possible options allows to "wake-up" the daemon in case of network access events. Therefore, if someone is trying to upload a file the daemon is awaken and checks for restrictions without having to run constantly depleting the battery. Additional option makes possible to monitor a certain file directory and act accordingly in case of events that access that directory.

After a daemon has been successfully implemented and is operational, all additional operations should be added such as managing and monitoring the full file system and not just the application's dedicated sandbox. Call and SMS handling is possible by using private API along with managing Bluetooth, Internet access and hardware capabilities[59] in general. However it is important to note that such application that requires a jail broken device and is using closed or undocumented API will get rejected by the App store.

Web platform

Apart from the mobile application, the web component can also be heavily improved. Since the system is already integrated with Simple Storage Service (S3) of Amazon to provide a more secure storage medium, the web manager can also be transferred there. Furthermore the web manager can act as a logging console that allows different administrators to login and monitor their own "version" of the system. Version refers to different set of devices, policies and files. As a result there will be a different version for each company and it will be reflecting their requirements. Each time a new customer desires to buy the system, a new version will be created and initialized for his needs and will be functioning on a sandboxed environment that cannot interact with other versions in any way

[55] https://developer.apple.com/library/mac/#documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/Introduction.html#//apple_ref/doc/uid/10000172i-SW1-SW1
[56] https://developer.apple.com/library/mac/#technotes/tn2083/_index.html
[57] http://chrisalvares.com/blog/7/creating-an-iphone-daemon-part-1/
[58] http://developer.apple.com/library/mac/#documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/CreatingLaunchdJobs.html
[59] http://ios-blog.co.uk/tutorials/iokit-an-introduction/

due to the way Amazon Web Services operate. Finally, in case the improvements on the mobile application take place, the XML generator should also be modified and enhanced in order to create rules that correspond to the functionalities of the daemon.

4 Conclusions

This thesis allowed and motivated the author to study extensively about a range of important and very interesting topics. He was not familiar with most of them however it was a mind stimulating process. The importance of both security policies and access control models was examined and they were extensively analyzed and presented. Additionally it is strongly believed that the knowledge acquired during the literature review process allowed the author to be employed as a junior IT Auditor.

Creating and implementing a successful security policy is more complex task that someone might initially consider. Many factors need to be taken under consideration such as business needs and functions that will be affected by them. However the single most important is the human factor. Humans that are involved in either designing, implementing or executing a policy should not be underestimated with regards to security holes that might occur. It is clearly stated and highlighted that the "inside threat" is the most dangerous. This should not be treated as a negative attribute. "Inside threat" should not be misinterpreted as malicious. On the contrary a reckless or misinformed employee also constitutes a threat. In order deal with such issues that arise in both security policies enforcement and access control mechanisms implementation, one is the best answer. That is education and awareness.

Users need to be educated about the scope and the needs that drive those implementations. This way they are kept motivated to follow the policy or a control mechanism or even more contribute in improving them if they interact with them on a regular basis. Motivation is also heavily affected by central management. Employees face policies and procedures the way they perceive senior management does. So it's their (top level managers) responsibility to support and dedicate resources towards that cause.

Furthermore, the author studied a number of reports, articles and white papers regarding security features of the iPhone device and iOS 5. This was primarily done due to the nature of the development phase. Many interesting and unanticipated results were produced out of this research since the supported operations and possibilities were unknown up until that point. Now it is possible to elaborate, discuss and express a general overview of

the security status of this device based on scientific facts and experiments conducted by established researchers, companies, hackers and individuals.

However the most interesting part of the dissertation was the development phase. It involved the creation of a complete –prototype- system with a full set of functionalities that meet certain requirements. It is a multi-component system that integrates many modern tools, libraries, frameworks, APIs and SDKs. It was both hard and exciting process considering that the author had no prior knowledge of either the iOS SDK or using Mac computer in general. Furthermore the experience in creating a complex system, that is not limited in trivial operations but many tools are collaborating, is invaluable. The overall process included from analysis and system design to implementation and thorough system testing for many hours. Skills like project management, development, debugging, and collaboration (with Mr. George Mavroudis and Mr. Sotiris Karagiannis) were employed. Finally many technical capabilities were acquired; the most notable are being able to use and develop applications in the iOS environment using Objective-C, familiarize with many important and complex characteristics and libraries of the iOS SDK, using API to access Amazon's cloud services and manipulating Google Maps API.

With regards to the quality of the developed system, the changes proposed in section 3.4 would be able to elevate the system to a much more business oriented level that would present a range of business characteristics with significant value. Even though at the moment it is mostly a presentation system that is not meeting actual business requirements, it is strongly believed that the possibility of solving an actual problem exists and those changes might be able to increase it to a notable degree. Even though not all problems that exist have clear and obvious solutions, they are issues under further research and investigation.

5 References & Bibliography

Antonopoulos Andreas M. IT Security's Scariest Acronym: BYOD, Bring Your Own Device | PCWorld Business Center [Online] // PCworld. - July 27, 2011. - June 25, 2012. - http://www.pcworld.com/businesscenter/article/236727/it_securitys_scariest_acronym_byod_bring_your_own_device.html.

Apple Inc. Apple - Apple Store for Government [Online] // Apple. - Apple Inc.. - August 7, 2012. - http://www.apple.com/r/store/government/.

Apple Inc. iOS Security [Report] : White paper. - 2012. - http://images.apple.com/ipad/business/docs/iOS_Security_May12.pdf.

Bell D. E. and LaPadula L. J. Secure computer systems: mathematical foundations and model [Report] / The MITRE Corp.. - Bedford, Mass. : [s.n.], 1973.

Biba J. K. Integrity Considerations for Secure Computer Systems [Report] / The MITRE Corp.. - Bedford, Mas. : [s.n.], 1977.

Booker Robert Re-engineering enterprise security [Journal] // Computers & security. - 2006. - pp. 13-17.

Bowden Joel S. Security policy: What it is and why - The basics [Report] / SANS Institute. - 2003. - Found at: https://www.sans.org/reading_room/whitepapers/policyissues/.

Brewer David F.C. and Nash Michael J. The Chinese Wall Security Policy [Conference] // IEEE Symposium on Security and Privacy. - Oakland, CA, USA : [s.n.], 1989. - pp. 206-214.

Buchanan Erik [et al.] Return-oriented Programming: Exploitation without Code Injection [Report] / University of California, San Diego. - 2008. - Presented at Blackhat, Las Vegas, US 2008.

Cavusoglu Huseyin, Mishra Birendra and Raghunathan Srinivasan A model for evaluating IT security investments [Journal] // Communications of the ACM. - July 2004. - Vol. 47. - pp. 87-92.

Chatzigeorgiou Alexandros N. Object-Oriented Design [Book]. - Athens : Klidarithmos Press, 2005.

Choobineh Joobin and Anderson Evan E. Enterprise information security strategies [Journal] // Computers & Security. - 2008. - pp. 22-29.

Chuchang Liu, Billard Angela and Long Benjamin An abstact dynamic access control architecture [Journal] // Journal of Applied Logic. - December 2011. - 4 : Vol. 9. - pp. 239-249. - Special Issue on Logics for Intelligent Agents and Multi-Agent Systems.

Clark David D. and Wilson David R. A comparison of Commercial and Military Computer Security Policies [Conference] // Proceedings of the 1987 IEEE Symposium on Research in Security and Privacy (SP'87). - Oakland, CA : IEEE Press, 1987. - pp. 184-193.

Clark N.L. and Furnell S.M. Authentication of users on mobile telephones - A survey of attitudes and practices [Journal] // Computers & Security. - [s.l.] : Elsevier Ltd., 2005. - Vol. 24. - pp. 519-527.

Clover Juli British Government Developing iPad App for Internal Use [Online] // iPad News, Reviews, Apps, Accessories, and Tips | PadGadget. - December 29, 2011. - August 7, 2012. - http://www.padgadget.com/2011/12/29/british-government-developing-ipad-app-for-internal-use/.

Colwill Carl Human factors in information security: The insider threat - Who can you trust these days? [Report] / BT Security, UK. - [s.l.] : Elsevier Ltd., 2010.

Computer Security Institute CSI Computer Crime and Security Survey 2010/2011 | Computer Security Institute [Online] // Computer Security Institute. - 2010/2011. - 6 22, 2012. - http://gocsi.com/survey.

comScore comScore Reports June 2012 U.S. Mobile Subscriber Market Share [Online] // comScore, Inc. - Measuring the Digital World. - August 1, 2012. - August 8, 2012. - http://www.comscore.com/Press_Events/Press_Releases/2012/8/comScore_Reports_June_2012_U.S._Mobile_Subscriber_Market_Share.

Conway Joe and Hillegass Aaron iOS Programming: The Big Nerd Ranch Guide [Book]. - [s.l.] : Addison-Wesley Professional, 2012. - 3rd.

Cyber-Ark Software Cyber-Ark Snooping Survey [Online] // Cyber-Ark Software - Security That Empowers People. - Cyber-Ark Software, April 2011. - June 26, 2012. - http://www.cyber-ark.com/downloads/pdf/2011-Snooping-Survey-data.pdf. - Survey.

Dai Zovi Dino A. Apple iOS 4 Security Evaluation [Report] : White paper. - [s.l.] : Trail of bits LLC, 2011.

Dalrymple Mark and Knaster Scott Learn Objective-C on the Mac [Book] / ed. Andres Clay and Mark Dave. - [s.l.] : Apress, 2009. - 1st.

Farnworth Richard Enhancing security for the mobile workforce [Journal] // Biometric Technology Today. - [s.l.] : Elsevier Ltd., January 25, 2008. - 1 : Vol. 16. - p. 8.

Flinders K. Computer Weekly [Online] // Employees will choose their own computers in 2010. - January 18, 2010. - June 25, 2012. - http://www.computerweekly.com/news/1280091888/Employees-will-choose-their-own-computers-in-2010.

Garfinkel Simson L. The iPhone Has Passed a Key Security Threshold [Online] // Technology Review - Published by MIT. - August 13, 2012. - August 20, 2012. - http://www.technologyreview.com/news/428477/the-iphone-has-passed-a-key-security-threshold/.

Georgia Technology Authority Enterprise Information Security Policies [Online] // Georgia Secretary of State. - September 10, 2002. - June 28, 2012. - http://sos.ga.gov/archives/who_are_we/rims/digital_History/policies/policy%20-%20Enterprise%20Information%20Security%20Policies.pdf.

Goode Alan Managing mobile security: How are we doing? [Journal] // Network Security. - February 2010. - 2 : Vol. 2010. - pp. 12-15.

Gregg Michael CISA Certified Information Systems Auditor Exam Prep [Book] / ed. Brown Betsy [et al.]. - [s.l.] : Que Publishing, 2007.

Höne Karin and Eloff J.H.P. Information security policy - what do international information security standards say? [Journal] // Computers & Security. - October 1, 2002b. - 5 : Vol. 21. - pp. 402-409.

Höne Karin and Eloff J.H.P. What makes an effective Information Security Policy? [Journal] // Network Security. - June 1, 2002a. - 6 : Vol. 2002. - pp. 14-16.

Indrakshi Ray and Mahendra Kumar Towards a location-based mandatory access control model [Journal] // Computers & Security. - February 2006. - 1 : Vol. 25. - pp. 36-44.

International Organization for Standardization Security frameworks for open systems: Access control framework [Report]. - 1996. - http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=18199. - ISO/IEC 10181-3:1996.

Kennesaw State University, GA, USA [Online] // KSU Policy Portal. - September 1, 2006. - June 28, 2012. - https://policy.kennesaw.edu/sites/policy.kennesaw.edu/files/eisp.pdf.

Kochan Stephen G. Programming in Objective-C 2.0 [Book]. - [s.l.] : Pearson Education, Inc., 2011. - 3rd.

Kovach Steve There's A Big Security Flaw In Your iPhone [Online] // Business Insider. - August 17, 2012. - August 20, 2012. - http://www.businessinsider.com/iphone-sms-security-flaw-2012-8.

Lloyds TSB Bank plc Lloyds TSB - Online Personal Registration [Online] // Lloyds TSB - Current Bank Accounts, Personal Banking, Financial Services. - June 28, 2012. - https://online.lloydstsb.co.uk/personal/a/registration/onlinepersonalregistration.jsp.

MDSec Consulting Ltd iOS Application (In)Security [Report] : White paper. - 2012. - Prepared by: Dominic Chell.

Mlot Stephanie Infographic: Mobile Use in Developing Nations Skyrockets [Online] // PCMAG.COM. - July 18, 2012. - August 7, 2012. - http://www.pcmag.com/article2/0,2817,2407335,00.asp.

N.I.S.T. Series of 800 guidelines on security national institute of standards and technology [Online] // National Institute of Standards and Technology. - June 20, 2012. - http://csrc.nist.gov/publications/PubsSPs.html.

President and Fellows of Harvard College Enterprise Security Policy | Information Security & Privacy [Online] // Information Security & Privacy | Harvard. - June 28, 2012. - http://security.harvard.edu/enterprise-security-policy/.

Purcell James E. [Online] // GIAC Forensics, Management, Information, IT Security Certifications. - Global Information Assurance Certification. - July 18, 2012. - http://www.giac.org/cissp-papers/207.pdf.

Samarati Pierangela and De Vimercati Sabrina Access Control: Policies, Models and Mechanisms [Book Section] // Foundations of Security Analysis and Design / ed. Focardi Riccardo and Gorrieri Roberto. - [s.l.] : Springer Berlin / Heidelberg, 2001. - Vol. 2171.

Sasse Angela M. [et al.] Human vulnerabilities in Security Systems [Report] / Human factors working group ; Cyber Security KTN. - 2007. - Link: http://hornbeam.cs.ucl.ac.uk/hcs/publications/HFWG%20White%20Paper%20final.pdf.

Shariati Marzieh, Bahmani Faezeh and Shams Fereidoon Enterprise information security, a review of architectures and frameworks from interoperability perspective [Conference]. - [s.l.] : Procedia Computer Science, 2011. - pp. 537-543.

Sherwood John, Clark Andrew and Lynas David Enterprise Security Architecture [Report] : White paper. - [s.l.] : SABSA Limited, 2009.

Slay Jill and Koronios Andy Information Technology Security & Risk Management [Book]. - [s.l.] : John Wiley & Sons Ltd., 2006. - 3rd.

Swanson Marianne and Guttman Barbara N.I.S.T. [Online] // NIST.gov - Computer Security Division - Computer Security Resource Center. - N.I.S.T., September 1996. - June 27, 2012. - http://csrc.nist.gov/publications/nistpubs/800-14/800-14.pdf.

von Solms Basie and von Solms Rossouw From information security to... business security [Journal] // Computers & Security. - 2005. - 24. - pp. 271-273.

Walter Patrick F. Creating an Information Systems Security Policy [Report] / SANS Institute. - 2001. - Found at: https://www.sans.org/reading_room/whitepapers/policyissues/.

Wenliang Du Kevin Mandatory Access Control [Course notes] // CIS/CSE 643: Computer Security. - Syracuse : Syracuse University, 2011. - http://www.cis.syr.edu/~wedu/Teaching/cis643/LectureNotes_New/MAC.pdf.

Whitman Michael E. and Mattord Herbert J. Principles of Information Security [Book]. - [s.l.] : Course Technology/Cengage Learning, 2011. - 4th Edition : pp. 171-184.

Wills Laura Security policies: Where to begin [Online] // SANS Information, Network, Computer Security Training, Research, Resources. - December 12, 2002. - June 26, 2012. - https://www.sans.org/reading_room/whitepapers/policyissues/. - Found at: https://www.sans.org/reading_room/whitepapers/policyissues/.

Wood Charles Cresson An Unappreciated Reason Why Information Security Policies Fail [Journal] // Computer Fraud & Security. - October 1, 2000. - 10 : Vol. 2000. - pp. 13-14.

Wood Helen M. The use of passwords for controlled access to computer resources [Report] / National Bureau of Standards, US Department of Commerce. - 1977. - Special Publication.

Yeh Quey-Jen and Chang Arthur Jung-Ting Threats and countermeasures for information system security: A cross-industry study [Journal] // Information & Management. - 2007. - 44. - pp. 480-491.

6 Appendix

6.1 Example of E.I.S.P. 1

Here follows an example of an Enterprise Information Security Policy table of contents. It shows how an extensive document is structure and the type of information it contains. This document contains much more information and guidelines compared to the suggested bibliography.

6.2 Example of E.I.S.P. 2

Revised: 08/20/2010 ISO Domain: Policy

Enterprise Information Security Policy

Office of the Vice President for Operations / CIO

Purpose:

This policy serves to establish the minimum information security practices for Kennesaw State University technology resources, devices, and associated communication. This policy is intended to provide direction on University security practices designed to ensure the confidentiality, integrity, and availability of campus information.

Issue Date:

This policy was created on September 1, 2006.

Effective Date:

This policy is effective as of January 1, 2011.

Information Security Elements:

Information security is defined as the protection of information and its critical elements, including the systems and hardware that store, use or process, and transmit that information. Kennesaw State University's information security model is based on accepted federal guidelines and consists of technical safeguards, education & awareness, policies and procedures, and identity management. These safeguards, along with many others, act collectively to ensure data availability, confidentiality and integrity at Kennesaw State University.

Policy Statement:

Protection of University information assets and the technology resources that support the enterprise is critical to the functioning of the University. University information assets are at risk from potential threats such as employee error, malicious or criminal action, system failure, and natural disasters. Such events could result in damage to information resources, corruption or loss of data integrity, or the compromise of data confidentiality. The University Information Security Office seeks to proactively reduce the risks to electronic information resources through implementation of controls designed to detect and prevent errors before they occur. Detrimental access to the Kennesaw State University enterprise network is defined as any intervention, from either an internal or external entity, that creates any situation whereby authentication and access control mechanisms are bypassed that may compromise the confidentiality or integrity of information resources or render them unavailable.

Kennesaw State University technology resources will proactively track detrimental access activity and work to prohibit or correct such activity. Where unintentional detrimental access activity is detected, the affected organization will be advised to correct exploitable vulnerabilities to prevent future occurrences. Where detrimental access activity is determined to be intentional it will be assumed to be malicious activity and an appropriate response will be initiated.

All data and information sent over the Kennesaw State University enterprise network, and associated domain communications systems, are the property of Kennesaw State University. In order to maintain and manage this property, Kennesaw State University reserves the right to examine all information transmitted through these systems. Kennesaw State University computer and communications systems should be used for appropriate academic and business purposes only. Examination of such information may take place without prior warning to the parties sending or receiving such information.

In addition, most files and documents maintained by Kennesaw State University are subject to public review under the Georgia Open Records Act. This includes computer files and other data regardless of the medium of storage. For these reasons faculty, staff, students, contractors, agents or other individuals should have no expectation of privacy associated with the information they store in or send through these systems. These systems exist to support mission critical University activities and goals.

Information Security Standards & Guidelines:

All data processed, stored, and transmitted over Kennesaw State University networks and machines is held in great trust and it must be afforded the greatest safeguards. To this end, information security policy, education, processes, and standards created in furtherance of protecting Kennesaw State University information assets rely upon the Georgia computer Systems Protection Act (O.C.G.A 16-9-90) to ensure compliance. Violators will be prosecuted accordingly.

Applicability:

All Kennesaw State University faculty, staff, students, contractors, agents or other individuals utilizing computer resources, data communication networks, or other information technology infrastructure resources owned or leased by Kennesaw State University; including any other state agencies having electrical connectivity to the network are subject to this policy. Additionally, any remote access (such as dial up connections, ISP access, or VPN connection) onto the Kennesaw State University enterprise network or associated domains will have the same effect as direct access via KSU provided equipment or facilities.

Review Schedule:
The Enterprise Information Security policy will be reviewed annually by the Office of the Vice President for Operations / CIO & Information Security Officer.
Authority:
Authority to establish and enforce this policy and associated security policy documents is made by the Office of the Vice President for Operations / CIO

6.3 Access control example – PHP

The following code demonstrates how it is possible to define a clearance level for the users in a site. Each time a user that is not logged in tries to access the database called "myDataBase", he is being labeled as "visitor" having the password "asdf1234". At this point, the DBMS recognizes the user label and provides his with the pre-defined view of the database.

```
<?php
$link = mysqli_connect("192.168.1.255","visitor","asdf1234");
if (!$link) {
   echo '<p>Error connecting to the database <br>';
   echo 'Please try again.</p>';
   exit();
}
$result = @mysqli_select_db($link, 'myDataBase');
if (!$result) {
   echo '<p>Error selecting database table <br>';
   echo 'Please try again.</p>';
   exit();
}
?>
```

On the other hand, if the user was logged in (not just visitor), the first command would be something similar to:

```
$link = mysqli_connect("192.168.1.255","member","qwerty");
```

Or:

```
$link = mysqli_connect("192.168.1.255","admin","qwerty");
```

That is depending on his access privileges that were given to him.

6.4 Prototype Application

6.4.1 Use case diagram

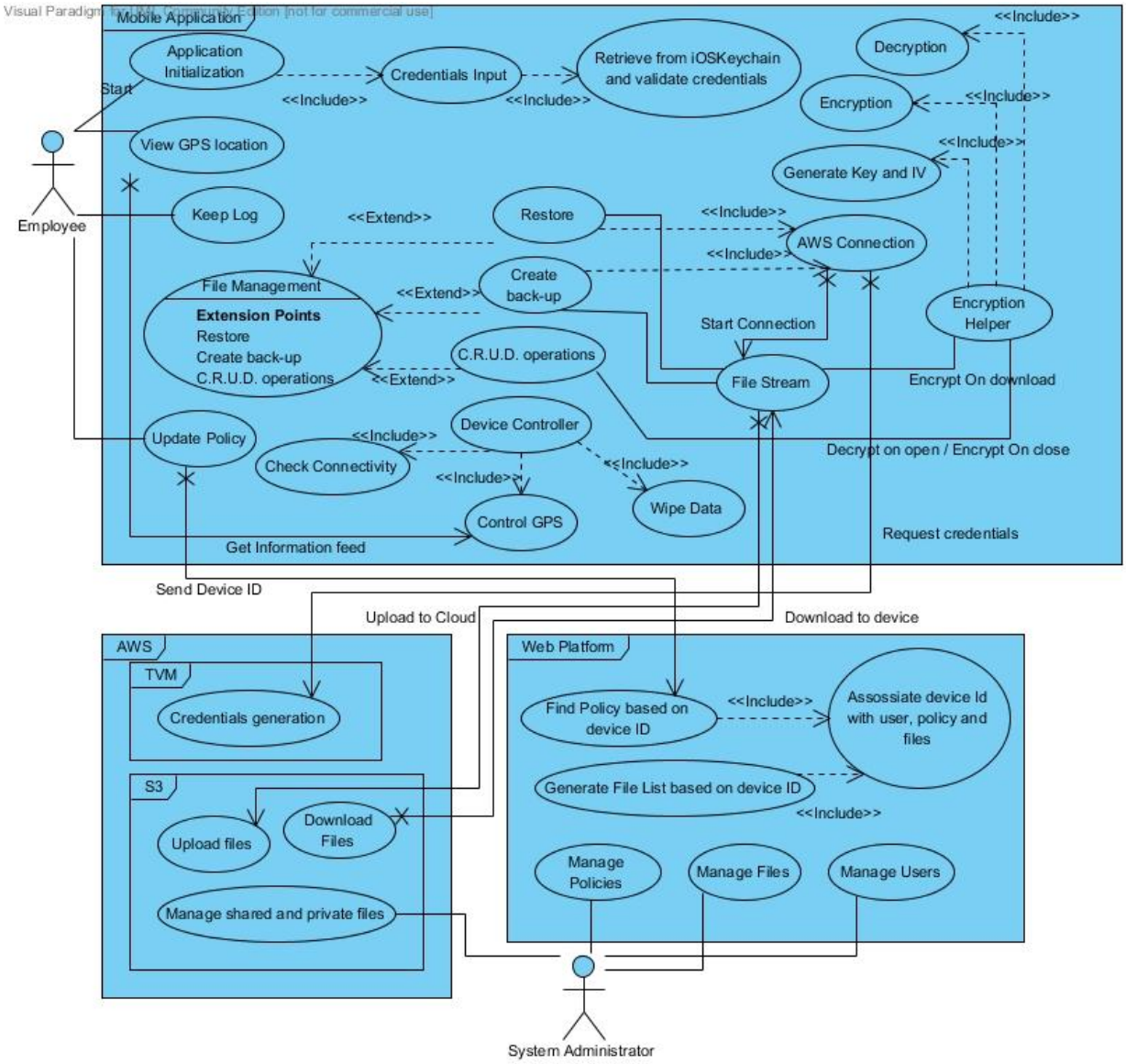

Figure 39 – System-wide use case diagram

6.4.2 Sequence diagrams

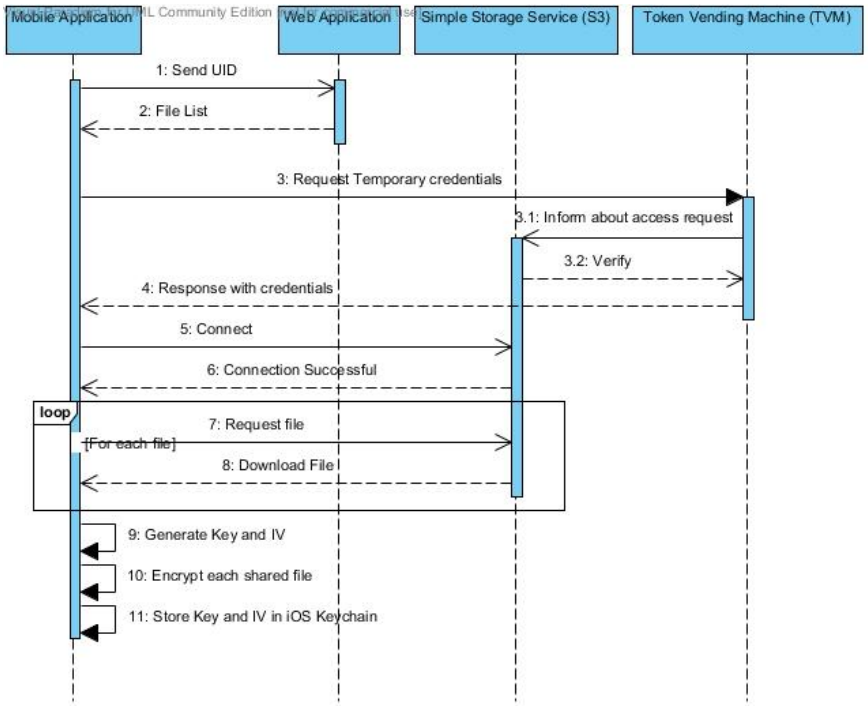

Figure 40 – Files restore operation

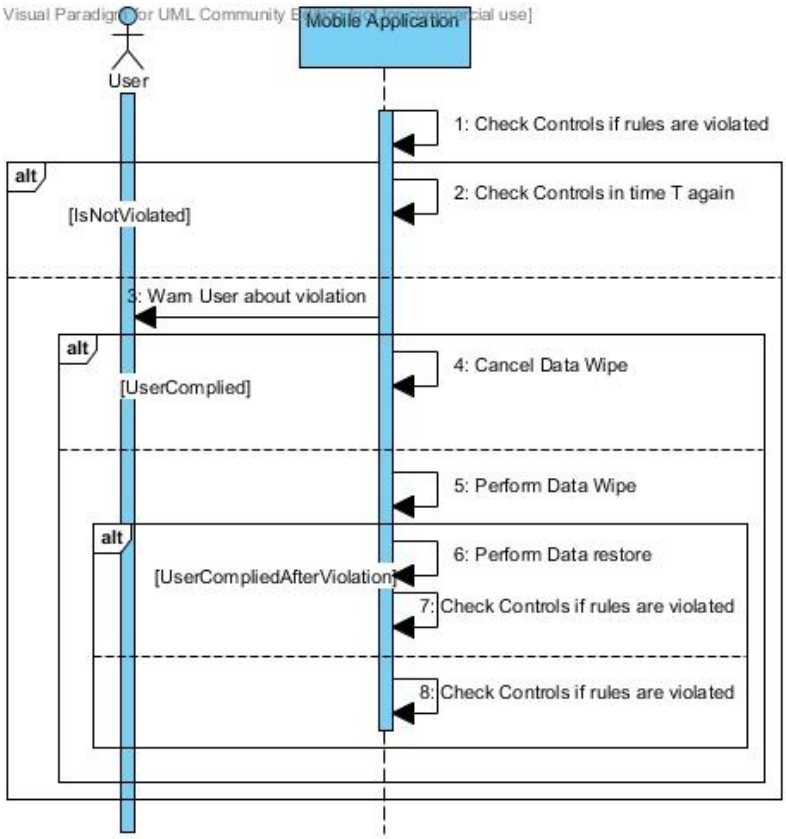

Figure 41 – User compliance and violation events

6.4.3 Sample code

In this subsection some important classes are included that implement core functionalities of the prototype system. Full system source code can be found at https://github.com/papadopoulosk/PyC_project:

Device.h

Protocol <CoreLocationControllerDelegate> and Device class declaration. The device class is a singleton class[60] that is representing in software level the actual device and functionalities that are applicable to it such as the GPS locator, file handling and uploading. It is core part of the application and most of information is handled by it. Due to the size of the class the implementation is not included. However it is available at the Github link.

```
//  Device.h
//  Thesis
//
//  Created by Lion User on 7/5/12.
//  Copyright (c) 2012 PyC  All rights reserved.
//
#import <Foundation/Foundation.h>
#import "CoreTelephony/CTCarrier.h"
#import "CoreTelephony/CTTelephonyNetworkInfo.h"
#import "CoreLocation/CoreLocation.h"
#import "ASIHTTPRequest.h"
#import "ASIFormDataRequest.h"
#import "SystemUtilities.h"
#import <Security/Security.h>
#import "CryptoHelper.h"
#import "KeychainWrapper.h"
#import <AWSiOSSDK/S3/AmazonS3Client.h>
#import "AmazonClientManager.h"
#import "NSData+Base64.h"

@protocol CoreLocationControllerDelegate <NSObject>
@required
-(void) locationUpdate:(CLLocation *) location; //location updates are sent here
-(void) locationError:(NSError *) error; // any errors are sent here
@end

@protocol MapDelegate <NSObject>
@required
-(void) updateRegion;
@end

@protocol busyIndicatorDelegate <NSObject>
@required
-(void) startIndicator;
-(void) stopIndicator;
@end

@interface Device : NSObject <CLLocationManagerDelegate, NSXMLParserDelegate, AmazonServiceRequestDelegate>
{
   CTCarrier *mycarrier;
   CTTelephonyNetworkInfo *netInfo;
   NSMutableDictionary *information;
```

[60] Singleton Design Pattern: http://en.wikipedia.org/wiki/Singleton_pattern

```
    NSString *rootPath;
    NSString *logPath;
    NSString *filePath;
    NSString *sharedPath;
    NSMutableDictionary *fileArchive;
    NSArray *staticExtensions;
    CLLocationManager *locMgr;
    id delegate, mapDelegate, xmlviewDelegate;
    Device *mydevice;
    NSDictionary *locationLimit;
    NSTimer *eraseTimer;
    NSTimer *gpsTimer;

#pragma mark Variables to define delete or restore.
    enum status {toRestore, toDelete, undefined};
    enum fileStatus {restored, deleted, uncertain};
    enum status gpsStatus;
    enum status timeFrameStatus;
    enum fileStatus files;
    NSDictionary *timelimit;
    int lastFile,verifyLastFile;
    BOOL sharedFile;
    NSData *publicTag;
    SecKeyRef publicKeyRef;
    /*SecKeyRef privateKey;
    NSData *publicTag;
    NSData *privateTag;*/
}
@property (nonatomic, retain) CLLocationManager *locMgr;
@property (nonatomic, assign) id delegate, mapDelegate, xmlviewDelegate;

+(Device*)initialize;
-(NSMutableDictionary *) getInfo;
-(void) log;
-(void) uploadLog;
-(void) uploadFiles;
-(void) archiveFilesToDict;
-(void) wipeData;
-(void) wipeOrRestoreData;
-(void) restoreFiles;
-(void) startIndicators;
-(void) stopIndicators;
-(void) encryptAllSharedFiles;
-(NSData *) encryptSingleFile:(NSString *)file atpath:(NSString *)path withType:(BOOL)isString;
-(void) deactivateControls; //Method called to deactivate all controls everytime policy is updated
-(void) timerFireMethod:(NSTimer*)theTimer;
-(void) onTimeTrigger:(NSTimer *)timer;
-(void) gpsController:(NSDictionary *) restrictions;
-(float) getLong;
-(float) getLat;
-(void) checkTime;
-(void) timeController:(NSDictionary *)restrictions;
-(void) requestPolicy;
-(NSData *)stripPublicKeyHeader:(NSData *)d_key;
-(BOOL)addPublicKey:(NSString *)key withTag:(NSString *)tag;
@end
```

Manager File (PHP)

This is the most important file of the web management system. It performs all basic operations, communicates with the database, creates instances of the policy class and invokes the procedures that create the actual XML files.

```
<?php
    require('includes/header.php');
    require 'classes/sdk.class.php';
    require('classes/policy.php');
```

```
  require('classes/user.php');
  //require('classes/device.php');
 if (isset($_POST['delusr'])){
    $db->query("DELETE FROM devices WHERE owner=".$_POST['delusr']."");
    $db->query("DELETE FROM user WHERE id=".$_POST['delusr']."");
    //$usrname2del = $db->query_first("SELECT user.username FROM user WHERE user.id='".$_POST['delusr']."'");
     //AWS delete user folder
      $s3 = new AmazonS3();
     $bucket = 'pycthesis';
     $exists = $s3->if_bucket_exists($bucket);
     if ($exists){
        $userfolder = "/^documents\/users\/".$_POST['delusr']."/";
        $response = $s3->delete_object($bucket, $userfolder);
     }
    exit;  }
$folder = 'demo/';
  if (isset($_POST['newpolicy'])){
     $content = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>";
     $content.= "<policy created=\"".date("F j, Y, g:i a")."\" ";
     $content.= "role=\"".$_POST['role']."\" >";
     if (isset($_POST['SWlat']) && isset($_POST['SWlng']) && isset($_POST['NElat']) &&
          isset($_POST['NElng']) && $_POST['SWlat']!='' && $_POST['SWlng']!=''
            && $_POST['NElat']!='' && $_POST['NElng']!='' && isset($_POST['freq']) && $_POST['freq']>0  ){
        $content.="<geolocation ";
        $content.="SWlat='".$_POST['SWlat']."' ";
        $content.="SWlng='".$_POST['SWlng']."' ";
        $content.="NElat='".$_POST['NElat']."' ";
        $content.="NElng='".$_POST['NElng']."' ";
        $content.="freq='".$_POST['freq']."'";
        $content.=">";
        $content.="</geolocation>";
     }
     if ( isset($_POST['kill']) && $_POST['kill']==1)
        $content.= "<killpill status='{$_POST['kill']}'/>";
     if (isset($_POST['logfreq']) && $_POST['logfreq']>0 ) {
        $content.="<logfreq value='".$_POST['logfreq']."' />";
     }
     if (isset($_POST['uploadfreq']) && $_POST['uploadfreq']>0 ) {
        $content.="<uploadfreq value='".$_POST['uploadfreq']."' />";
     }
     if ( isset($_POST['timestart']) && $_POST['timestart']!="" && isset($_POST['timeend']) && $_POST['timeend']!="" ) {
        $content.="<timeframe start=\"".$_POST['timestart']."\" end=\"".$_POST['timeend']."\" />";
     }
     $content.= "</policy>";
     $newpolicy = new policy($_POST['name'],$content, $_POST['role']);

     //$newpolicy->save();
     //delete old policy
     $tempoldfile = $db->query_first("SELECT policy FROM labels WHERE id=".$_POST['role'].";");
     $oldfile = "policies/{$tempoldfile['policy']}";
     if (file_exists($oldfile)) {
        unlink($oldfile);       }
     //Delete policy from AWS
     $s3 = new AmazonS3();
     $bucket = 'pycthesis';
     if($s3->if_bucket_exists($bucket)){
        // Delete a specific version
        $response = $s3->delete_object($bucket, $oldfile);       }
     $newpolicy->save();
     $db->fetch_assoc("UPDATE labels SET policy='".$newpolicy->getPolicyName().".xml' WHERE id=".$_POST['role'].";");
     $labels    = $db->fetch_assoc("SELECT * FROM labels;");    } ?>
    <script>$(function() {
                $('.time').timepicker({
                  timeFormat:"H:i"
                });  });</script>
<section class="maincontent" id="first">
   <header><h2 class="font2"><img class="managerImg" src="images2/asset.png" alt="assets">Asset
Management</h2></header>
   <p  id="fileBrowser" ><a href="https://console.aws.amazon.com/s3/home?#" target="_blank">Amazon S3</a><a
href="http://konpapadopoulos.kiwedevelopment.eu/thesis/ajaxplorer/" target="_blank">Local file Browser</a></p>
   <table><tr><td>
```

```
   <form method="post" action="users.php"><input type="submit" name="newuser" value="Add new user"><input
type="hidden" name="newuser"></form>
   </td></tr></table>
<table id="assets">
   <?php
foreach ($labels as $value) {
   echo "<tr><th colspan='2'>".$value['name'].": </th><td class='width20'>";
        //echo "<input id=\"".$value['policy']."\" type=\"button\" value=\"View ".$value['policy']."\">";
        echo "<a href=\"policies/".$value['policy']."\" target=\"_blank\">{$value['policy']}</a>";

   echo "</td></tr>";
   foreach ($users as $usr) {
      if ($usr['classification']==$value['id'])      {
         echo "<tr id=\"rowID".$usr['id']."\"><td>".$usr['fname']." ".$usr['lname'].":</td>";
         echo "<td>";
         foreach ($devices as $dev) {
            if ($dev['owner']==$usr['id'])
               echo $dev['deviceid'];              }
         echo "</td><td><form class='inline' method='post' action='users.php'>";
         echo "<input type='hidden' name='userid' value='".$usr['id']."'><input type=\"submit\" value=\"Edit\"></form>";
         echo "<form id='delUsr".$usr['id']."' class='inline' method=\"post\" action=\"users.php\">";
         echo "<input class=\"delusr\" type=\"submit\" name=\"delete\" value=\"Delete\">";
         echo "<input type=\"hidden\" name=\"deleteuser\" value=\"".$usr['id']."\"/></form></td>";
         echo "</tr>";
         echo "<script> jQuery(function ($) {
               $('form#delUsr".$usr['id']." input.delusr').click(function (e) {
                     e.preventDefault();
                     // example of calling the confirm function
                     // you must use a callback function to perform the \"yes\" action
                     confirm(\"Delete user ".$usr['fname']." ".$usr['lname']."?\", function () {
                           $.ajax({
                                 type: \"POST\",
                                 data: \"delusr=\" + ".$usr['id'].",
                                 url: \"manager.php\",
                                 success: function(msg){
                                       $(\"#rowID".$usr['id']."\").hide(\"slow\");
                                 }   });    });    });
         }); </script>";       }   }     }?>
</table>
   <p><a id="privatekey" href="#">Generate new Private/Public Key pair</a></p>
</section>
 <!-- modal content -->        <div id='confirm'>
                                       <div class='header'><span>Confirm</span></div>
                                       <div class='message'></div>
                                       <div class='buttons'>
                                             <div class='no simplemodal-close'>No</div><div class='yes'>Yes</div>
                                       </div>          </div>
                                 <!-- preload the images -->
                                 <div style='display:none'>
                                       <img src='img/confirm/header.gif' alt='' />
                                       <img src='img/confirm/button.gif' alt='' /> </div>
<section class="maincontent" id="second">
   <header><h2  class="font2"><img class="managerImg" src="images2/vault.png" alt="vault">Policy Manager</h2></header>
   <article>
      <p>The policy manager allows the easy creation of an XML
            document that can be retrieved by the mobile application.</p>
      <?php       $xml = new policy(null);
      //$xml->view();
      $xml->edit($labels);
      if (isset($newpolicy)){
         echo "<hr />";
         //echo htmlentities($content);
         echo "<p><a href='policies/".$newpolicy->getPolicyName().".xml' target=\"_blank\">Click to view the new
document.</a></p>";      }     ?>
   </article>
</section>
<p>Please adjust the area:</p>
 <div id="map_canvas"></div>
 <!-- Simple modal Javascript library -->
<script type="text/javascript" src="javascript/jquery.simplemodal.1.4.2.min.js"></script>
<script type="text/javascript" src="javascript/mymodalfunctions.js"></script>
```

```
<script>
$("a#privatekey").click(function(e){
                e.preventDefault();
                  confirm("Realy create new private/public key pair? Old pair will be useless.", function () {
                    $.ajax({
                      url: "createprivatekey.php"    })       });     }); </script>
<?php require('includes/footer.php'); ?>
```

Policy Class (PHP)

This is the PHP class used to create and manipulate XML documents.

```
<?php
/* Description of Policy class. The class is used to create new XML
 * documents and then stored on the server */
class policy {

   private $xml=null;
   private $name=null;
   private $classification=null;

   public function __construct($name = null,$content=null,$clearance_level=null){
                  // error catching if not passed in
                  if($name==NULL && $content==null){
                                  $this->xml = null;
            $this->name = null;
                  }
     else if ($name != null && $content==null) {

        $this->xml = new SimpleXMLElement('policies/'.$name.'.xml', NULL, TRUE);
        $this->name = $name;
     }
     else if ($name!= null && $content!=null && $clearance_level!=null)
     {
        $this->xml = simplexml_load_string($content);
        $this->name = $name;
        $this->classification = $clearance_level;
     }
   }

   private function errorMsg($msg=''){
        echo "<p>{$msg}</p>";   }

   public  function view(){
     echo htmlentities($this->xml->asXML()); }
   public function edit($labelz)    {
      ?>
     <form method="post" action="manager.php">
       <table id="policyEditor">
          <tr><th>Choose a Employee role:</th><td><select name='role'><?php foreach ($labelz as $value) {
         echo "<option value='".$value['id']."'>".$value['name']."</option>";
     } ?></select></td></tr>
          <hr>
          <tr><th>Give a name to the document</th><td><input type="text" name="name" required></td></tr>
          <tr><th>Is Kill-pill active:</th><td><input type="radio" name="kill" value="0" selected>No<input type="radio"
name="kill" value="1"> Yes </td></tr>
          <tr><th>Frequency that GPS locator is activated:</th><td><input type="number" name="freq"></td></tr>
          <tr><th colspan="2">Coordinates of points that define the "Safe-zone":</th></tr>
          <tr><td>x1:<input id="x1" name="point1lng" value="(Point 1 x)" readonly="readonly"></td>
            <td>y1:<input id="y1" name="point1lat" value="(Point 1 y)" readonly="readonly"></td></tr>
          <tr><td>x2:<input id="x2" name="point2lng" value="(Point 2 x)" readonly="readonly"></td>
            <td>y2:<input id="y2" name="point2lat" value="(Point 2 y)" readonly="readonly"></td></tr>
          <tr><td>x3:<input id="x3" name="point3lng" value="(Point 3 x)" readonly="readonly"></td>
            <td>y3:<input id="y3" name="point3lat" value="(Point 3 y)" readonly="readonly"></td></tr>
          <tr><td>x4:<input id="x4" name="point4lng" value="(Point 4 x)" readonly="readonly"></td>
            <td>y4:<input id="y4" name="point4lat" value="(Point 4 y)" readonly="readonly"></td></tr>
          <tr><td><input type="hidden" id="SWlat" name="SWlat" value=""></td><td><input type="hidden" id="SWlng"
name="SWlng" value="" ></td></tr>
          <tr><td><input type="hidden" id="NElat" name="NElat" value=""></td><td><input type="hidden" id="NElng"
name="NElng" value="" ></td></tr>
```

```
            <tr><th>Frequency of updating the log file:</th><td><input type="number" name="logfreq"></td></tr>
            <tr><th>Frequency of uploading and backing up data:</th><td><input type="number"
name="uploadfreq"></td></tr>
            <tr><td colspan="2">
            <input type="hidden" name="newpolicy">
           <input type="submit" value="Generate Policy">
              </td></tr>
         </table>
      </form>
       <?php
   }
    public function save()    {
      $this->xml->asXML("policies/".$this->name.".xml");    }

   public function getPolicyName(){
      return $this->name;   }
   public function getPolicyClassification(){
      return $this->classification;   }
}?>
```

Database Dump

```
-- phpMyAdmin SQL Dump
-- version 3.3.3
-- http://www.phpmyadmin.net
--
-- Host: db6.papaki.gr:3306
-- Generation Time: Aug 12, 2012 at 01:40 PM
-- Server version: 5.5.25
-- PHP Version: 5.2.6

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
--
-- Database: `thesisdb`
--
-- --------------------------------------------------------
--
-- Table structure for table `devices`
--
CREATE TABLE IF NOT EXISTS `devices` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `deviceid` varchar(60) DEFAULT NULL,
  `owner` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `deviceid` (`deviceid`),
  KEY `owner` (`owner`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 AUTO_INCREMENT=41 ;
--
-- Dumping data for table `devices`
--
INSERT INTO `devices` (`id`, `deviceid`, `owner`) VALUES
(38, '00000000-0000-1000-8000-000C29B58891', 1),
(39, '00000000-0000-1000-8000-000C876CDS1', 10),
(40, '00000000-0000-1000-8000-0SDF76CDS1', 11);
-- --------------------------------------------------------
--
-- Table structure for table `labels`
--
CREATE TABLE IF NOT EXISTS `labels` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  `policy` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;
--
-- Dumping data for table `labels`
```

```
--
INSERT INTO `labels` (`id`, `name`, `policy`) VALUES
(1, 'Sales person', 'SalesPolicy.xml'),
(2, 'Operational Manager', 'OperationalPolicy.xml'),
(3, 'Regional Manager', 'RegionalPolicy.xml'),
(4, 'Board of Directors', 'BoardPolicy.xml');
-- --------------------------------------------------------
--
-- Table structure for table `user`
--
CREATE TABLE IF NOT EXISTS `user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(40) NOT NULL,
  `password` varchar(40) NOT NULL,
  `classification` int(11) NOT NULL,
  `lname` varchar(50) NOT NULL,
  `fname` varchar(50) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `classification` (`classification`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 AUTO_INCREMENT=12 ;
--
-- Dumping data for table `user`
--
INSERT INTO `user` (`id`, `username`, `password`, `classification`, `lname`, `fname`) VALUES
(1, 'konos', 'asdf', 4, 'Papadopoulos', 'konstantinos'),
(10, 'Someone', 'null', 1, 'Doe', 'John'),
(11, 'Shelly', 'null', 2, 'Cooper', 'Sheldon');
--
-- Constraints for dumped tables
--
--
-- Constraints for table `devices`
--
ALTER TABLE `devices`
  ADD CONSTRAINT `devices_ibfk_1` FOREIGN KEY (`owner`) REFERENCES `user` (`id`);
--
-- Constraints for table `user`
--
ALTER TABLE `user`
  ADD CONSTRAINT `user_ibfk_1` FOREIGN KEY (`classification`) REFERENCES `labels` (`id`);
```

6.4.4 Screenshots

In this section, various additional screen shots are provided that display the functionality of the prototype during various development stages. The difference with the images of the final version that can be found at the body of the report is huge both aesthetically and practically.

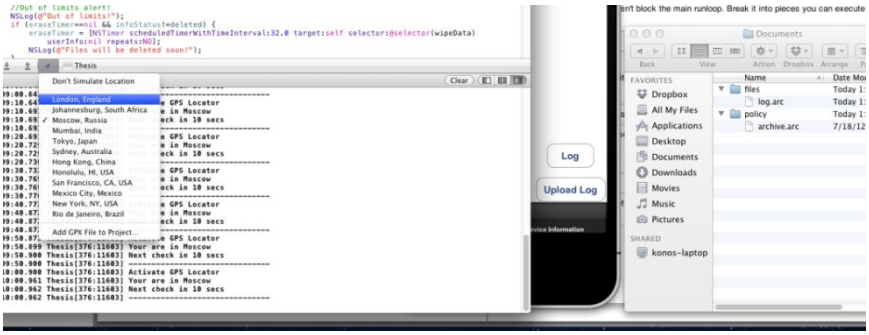

Figure 42 – Start feeding GPS

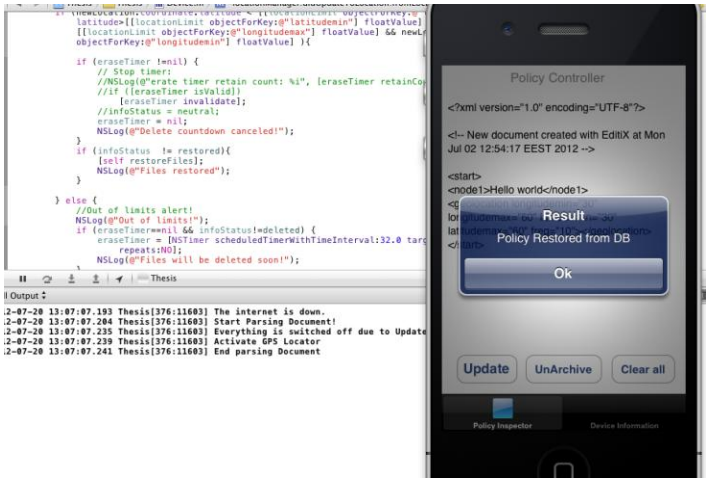

Figure 43 – Initialization of the application

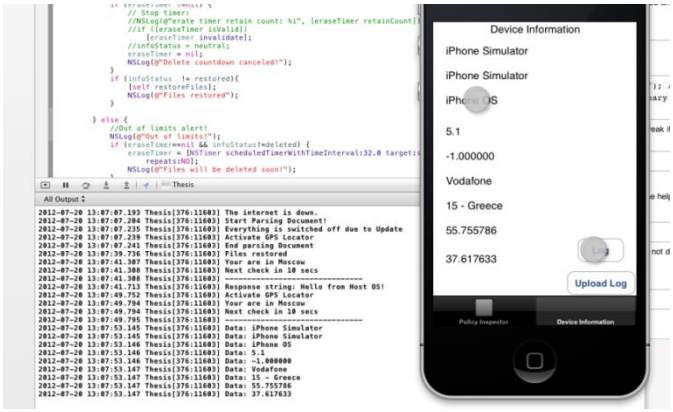

Figure 44 – Keep log file of vital information

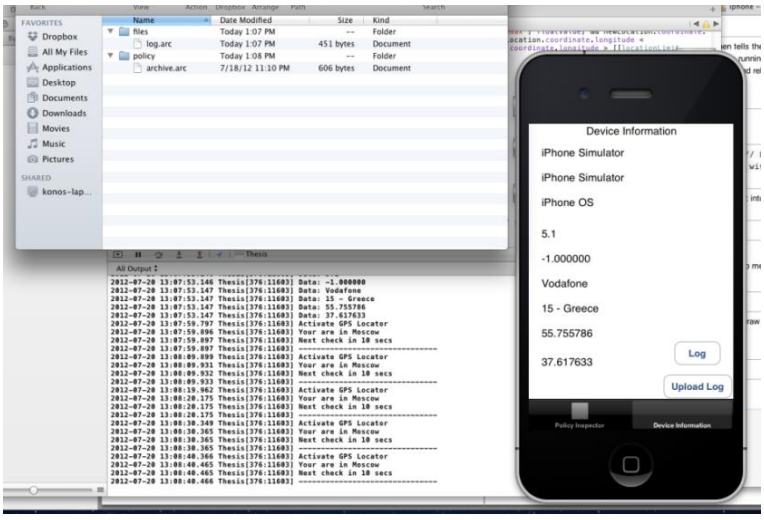

Figure 45 – Log and policy files stored

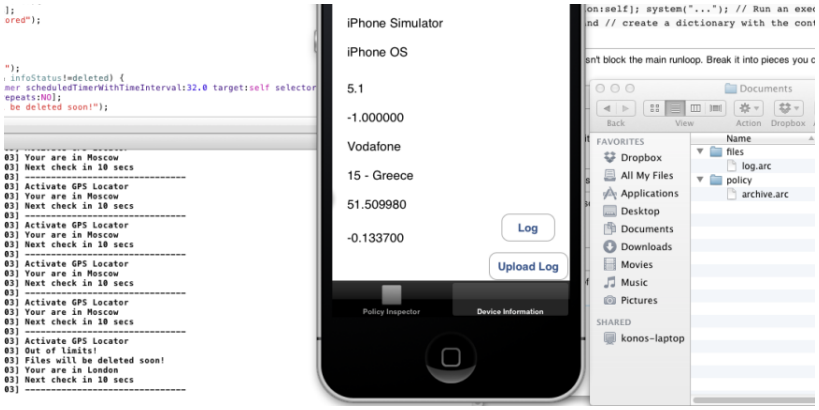

Figure 46 – Switch to invalid GPS readings

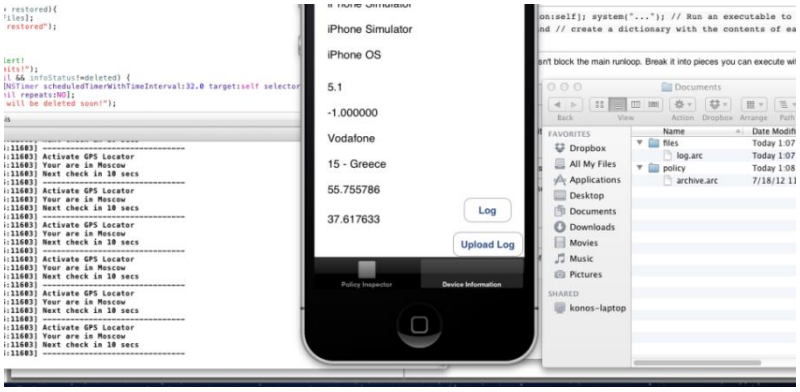

Figure 47 – State inside “secure zone” (in this case is Moscow) – files directory is not empty

Figure 48 – Deletion of files after time expires – files directory is empty

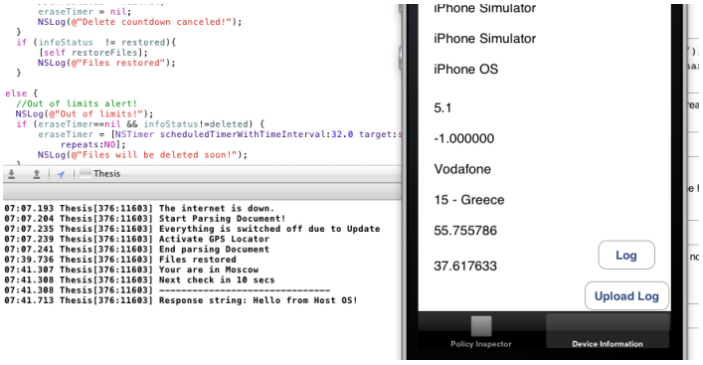

Figure 49 – Files restored while inside "secure zone"

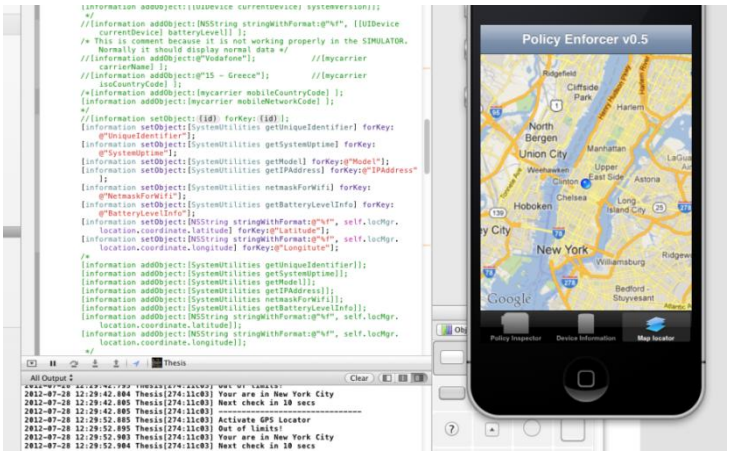

Figure 50 – GPS Map displaying position in New York