International Journal of Engineering &Technology, 7 (1.7) (2018) 87-90

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Location based security architecture evaluated using ATAM

Almas Begum [1] *, V. Cyril Raj [2]

[1] *Assistant Professor, Department of Computer Science and Engineering, School of Computing, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai-62, TamilNadu, India*
[2] *Professor, Department of Computer Science and Engineering, Dr. MGR Educational & Research Institute, Chennai-, TamilNadu, India*
**Corresponding author E-mail: almasbegum@gmail.com*

Abstract

The usage of internet along with growing number of smart phones let public to access and store delicate data , Mobile banking, Contact lists, information on Location(EG: GPS), Etc. regularly. So it is becoming a challenging task to provide Security to smart phones for protecting data. The most important step in the early stage of software development is architecture representation and analysis. The objective of this paper is to propose an architecture for smart phones, represent using MVC model and evaluated using ATAM to extract security quality attributes. ATAM list the quality attributes in the following categories that is Risk, Non-Risk, Tradeoff and sensitivity points.

Keywords: *ATAM; Mobile Architecture; Software Architecture; MVC.*

1. Introduction

In this new era, it is endorsed smart phones are globally used. These smart phones, with the help of internet, are able to store personal data, perform Mobile banking, etc where ever they are and whenever they want. These smart phones are also used to obtain information on social events as well as information on places based on the locations.

Software Architecture Analysis in recent years is becoming a vital domain to make Qualitative and Quantitative analysis of the software. So, to analyze and to determine the software architecture, functional and non-functional specifications are needed. "The composite characteristics of software that determine the degree to which the software in use will meet the expectations of the customer"(IEEE Std 729-1983) [3].There are ample of scenario based software architectural evaluation methods are available. They are SAAM (Software Architecture Analysis Method), ATAM (Architecture Trade-off Analysis Method), CBAM (Cost Benefit Architecture Method), ALMA (Architecture-Level Modifiability Analysis) developed by SEI to identify risk of any software at the earliest.

In this paper, Security quality attribute is taken with few aspects and implemented with ATAM. The aspects that are taken are (i) Integrity, (ii) Confidentiality, (iii) Availability, (iv) Performance, (v) Modifiability (vi) Portability.

2. Related work

Analysis and Evaluation of software architecture is becoming a complete practice. There are different methods for this practice in software society. They are SAAM, ATAM, CBAM, ALMA, FAAM [2]. Life cycle of every software development process requires recognition of potential risks, sensitivity points and tradeoffs. This goal is accomplished with the help of architectural evaluation method called ATAM (Architectural Trade-off Analysis Method). Also using ATAM, business goals and quality goals of any architecture can be designed and extracted.

3. ATAM

ATAM is a scenario-based risk-mitigation technique to study software architecture with quality attributes. ATAM has nine steps [1]. As ATAM is considered a mature approach, it has been validated in different domains [5]. The following are the steps used in evaluation of ATAM.

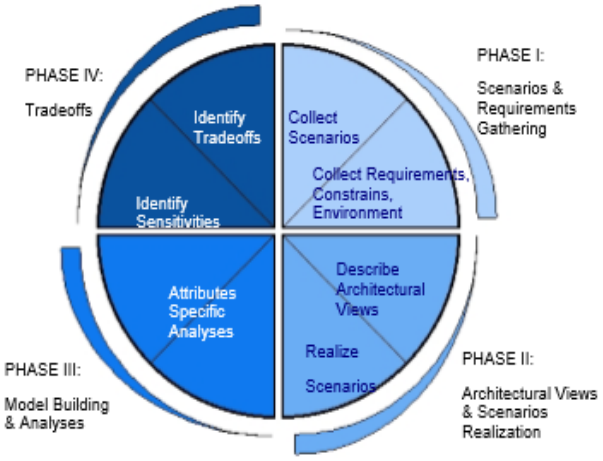

Fig. 1: Architectural Tradeoff Analysis Method Steps & Phases.

Step 1: Present ATAM - gives outline of steps, techniques and out comes from the steps of ATAM.
Step 2: Present Business Drivers – describes the interpretation of different process and their objectives.
Step 3: Present Architecture – describes the outline of architecture.

Step 4: Identify Architectural Approaches – list and specify the observed architectural decision from step 3.
Step 5: Generate Utility Tree – identify, prioritize and refine the better quality attribte in utility tree form.
Step 6: Analyze Architectural Approaches – evaluate risks, sensitivity and tradeoff points.
Step 7: Brainstorm and Prioritize scenarios – develop and evaluate the scenarios identified for better understanding of quality attributes and their importance.
Step 8: Analyze Architectural Approaches – extends step 6 to find risks, sensitivity points and tradeoff.
Step 9: Present Results – reiterate ATAM steps and present the outcomes.

4. MVC representation &ATAM evaluation

The architecture for smart phonesis represented using Model-View -Controland security attributes are elicited using an architectural analysis method – ATAM (Architectural Tradeoff Analysis Method). The purpose of this architectural analysis is to propose Security for trustworthiness of information where ever it is needed. In the following section, architectural analysis is performed with the ATAM Phases.
Phase 1: Presentation Phase:
Step 1 ~ Step 3: The first three steps of ATAM [9] are combined in this phase. The first step is to present architecture outline for providing security for mobile devices while using in location based services. The second stepexplains about the interpretation on the business objectives of the architecture. The main objective is to provide security for the smartphones when accessing important information, thereby making the architecture flexible from various locations. The third step explains about the evaluation and communication that occurs in the architecture. The communication reveals the architectural decisions made that are based on quality attributes and design of the architecture.
The following section illustrates quality attributes and its architectural decisions that are made.The achievement of any architecture is reliant on its quality attributes and its architectural decision [6]. Every quality attribute is described in to three categories: external stimuli, architectural decisions, and responses [4]. The architecture in this paper is evaluated with quality attributes such as Security, Integrity, Availability, Performance, Modifiability and Portability.
The security attribute aims in providing less chance of using unauthorized and unauthenticated usage of information thereby preventing from confession or loss of information. The next attribute is Integrity, refers to the accuracy, completeness and validity of information without corruption. Availability attribute refers to provide information and all vital services for the users whenever and where ever needed. The success of security attributes is based on Integrity and Availability. The Performance attribute is based on the receptiveness of a request taken and responding back within a short interval of time. Performance attribute is always based on how fast interaction takes place between different applications. The next attribute is Modifiability. It refers to the how a system can be modified efficiently and effectively without any interruptions. The last attribute taken is portability. It refers to the capacity of a system to run under different situations.

4.1. Architectural decisions

The accomplishment of all quality attributes taken are interdependable. The following are the architectural decisions considered for the quality attributes.

4.1.1. Architectural decisions to support security

The security attribute must provide strong authentication and authorization to the information being accessed from various locations when using Location based Services (LBS).
AD1: Model-View-Controller (MVC)
This is the most common architectural design used in software architecture for communication between system components. The following diagram figure2 shows the basic model of MVC Model. The main objective of using MVC model is to fetch the information from the database and to update the user interface with the needed information based on user requests. It contains three interrelated parts for the illustration of information.

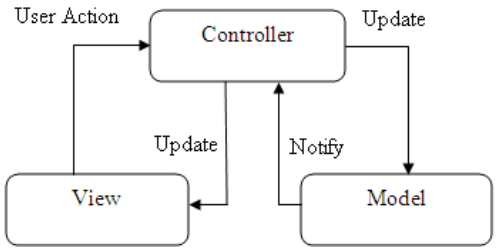

Fig. 2: Common MVC MODEL.

The following diagram figure 3 illustrate the same concept for using smartphones in LBS. Here the Model is used to fetch the saved information from the database and pass notification to the controller for update of information. The view is used for displaying the information to the end-user on the screen. There are different views that show information in different ways based on the screen display. It also takes the responsibility of collecting information from Model and presents it to the end-user. The last is Controller, used to update the model, collect the user-actions in the form of inputs based on the interface and to update the change in view of displaying information when needed.

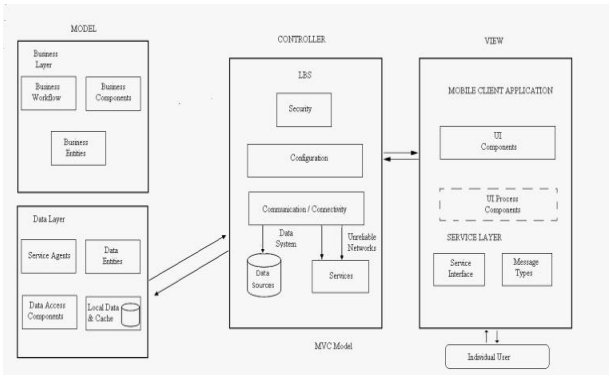

Fig. 3: MVC Model for Security in Smartphones.

AD2: Data Access
The second architectural decision considered is Data Access. Smartphone users always have access to sensitive data with their devices. There are a lot of concerns that are to be recognized to secure sensitive information from getting attacked. The main concern to secure data access is to handle connection timeouts, disconnections properly, selecting incorrect data format. When doing any kind of transaction, failing to verify data fields, non-filtering of false characters, granting access to stay on transaction page for a long time, mishandling of data access exceptions. Security must be provided for accessing data in order to prevent loss of information during device theft or while doing online transactions.
AD 6: Connections & Communications
The next architectural decision considered is Connections & Communication that are needed for any kind of uploading and downloading of data. All kind of connections are done in the data layer. Nowadays all communications done with Smartphone are based on over-the-air. So security of sensitive data becomes a concern that has to be taken care. Authentication and authorization has to be verified for any kind of connection such as over-the-air, wired, Bluetooth, so that security can be provided to the data that will be accessed from any location. When a communication has to

be made from a different location, user has to make sure there is a two way communication.

4.1.2. Architectural decisions to support integrity

The next quality attribute taken is Integrity which represents the state of being firmness in good design and also delivering the information and services as intended. When users use smartphones to access any information, for example opening a document, the document has to be intended to the user according to the design of the device with no loss in the information. The following are the decisions made to locate these concerns.

AD3: Synchronization

Synchronization ("sync") is one of the most important architectural decision that must be considered to ensure the consistency of data when uploaded or downloaded from device to device. Synchronization can be over-the-air or cradled or both types [7]. Every time an application is installed by the user, sensitive data gets involved in the sync process. While sync is being processed, interruptions may occur. Therefore an exquisite way to deal with interruptions that must allow the process to get resumed once the connection is available. Lack of connection and communication (AD6), synchronization cannot be performed with the data when end users intends.

4.1.3. Architectural decisions to support availability

When supporting security and synchronization, it is also an important role for architecture to make data available when required to the end users. Availability can be defined as the property of making the data and services to be available whenever the end user requires. Availability should be as high as possible by reducing the downtime, making the data & services available by resolving automatically any kind of fault occurrences. The following are the decisions to be made for making the information available to the end users.

AD5: API

To make the information available to the end users from any location, API also plays a vital role as all applications are API-centric, were API acts in front end to connect with rich clients and gets combined with internal components at the backend. In case of any interruptions occur, connections & communications (AD6) also gets interrupted.

AD7: Power

Consumption of power in Smartphones are becoming challenging task as all applications has utility towards the battery. As long as battery life is good, all tasks can be performed better without interruptions. All applications must be enhanced to reduce the usage of power to increase its performance mainly during power is very less. In case the device has very less power and user performs any kind of transactions from a different location, software performance has to be reduced so that transactions can be done with no data integrity loss.

AD8: Coupling & Cohesion

To improve the performance among the components, ensure it is always loosely coupled and highly cohesive between the layers. When any sensitive information is accessed by smartphone user, the processing of information should be between the access layer and the data layer at back end.

4.1.4. Architectural decisions to support performance

The next important quality attribute taken is Performance. To have a better performance, goals has to be identified at the earliest. There are many constraints that has to be considered while designing applications for Performance. Some of them are Device (AD4), Power (AD7) and Connections & Communications (AD6). AD6 is interrelated to both Device and Power. The following are the decisions to be considered to have a better and balanced Performance.

AD4: Device

Each device differs in architecture and its performance. There are certain features that are unique to the device such as camera, GPS, GPU, etc that will increase the performance of the device. While designing the device, consideration for battery life, screen resolution, memory size, etc must be made. Device also plays a vital role in authenticating the sensitive information that is stored in itself when there is detached database.

AD7: Power

The next decision to support Performance is Power which is the most narrow factor of the device. The amount of power consumed while using accessing applications specifies the battery life. Resources such as speed of processor,wireless connections &communications , reading , writing & storing information in to the memory, etc may reduce the battery life. Restriction has to be provided for any kind of transaction to be made when applications are running on power saver mode so that data integrity is not lost. Failing to provide restriction may lead to data integrity issues. Different types of communication protocols can be used to increase the usage of power.

4.1.5. Architectural decisions to support modifiability

The next important quality attribute taken is Modifiability. This attribute is needed when there is more development in requirements. Modifiability depends on factors such as changes made to the system and cost of changes that are made. Cohesion & Coupling (AD8) can be used to explicit modifiability [10]. To improve modifiability, make sure there is an increased cohesion and reduced coupling is being implanted.

4.1.6. Architectural decisions to support portability

The last quality attribute taken is portability. Portability is a type of modifiability which helps software to get adapted in different environment.

Phase 2: Investigation & Analysis:

Step 4 ~ Step 6: In this phase next 3 steps are combined. They are Identifying Architectural Decisionsand analyzing architectural decisions, to elicit the importance of architectural decisions made and to identify sensitivity, tradeoff points and risks.A combination of different architectural styles is followed to explore information and services either locally or remotely. There are eight architectural decisions made and elicited in the above phase.

The refinement of quality attribute with scenarios are mapped to a scale rating as High (H), Medium (M), Low (L) with parameters as importance and difficulty. These parameters are used to rate the success of scenarios and what is the severity in achieving the scenarios.

Different types of refinements are extracted from the architecture to explain and justify various quality attributes that supports security. Every quality attribute elicited is depicted in three ways. They are

- Stimuli – an event that make architecture to respond or change.
- Responses – measurable answer for stimuli that is generated.
- Architectural decisions – are those conditions of an architecture that has an explicit impact on concluding attribute responses [5].

ATAM has three types of scenarios elicitation [8]. They are use case scenarios, growth scenarios, exploratory scenarios. Use case scenarios are used for information elicitation, growth scenarios are used for expected changes to the system, and exploratory scenarios aimed for stress that changes the system. The following are the different types of scenarios elicited for different quality attributes.

The first scenario for Security attribute taken is "Longer complicated paascode". The rating given is (H, L) which illustrate that, to have security for any kind of sensitive information, it is highly important to have longer complicated passcode so that it will be less in difficulty to break passcode.

The next quality attribute taken is Integrity. One of the scenario taken is "Signed and validated profile content". This will be useful when the user has some business scenarios to be authenticated

along with a digital signature, which will ensure that the information has not been intercepted without the user's observation. So the rating defined is (H, L), that is, it is highly important to provide integrity for the information with less difficulty of getting corrupted.

The next scenario taken is for Availability quality attribute. One of the scenarios considered is "Loss access to trusted device". The rating given is (H, L). It is highly important for any user to access sensitive information from trusted device which will be less in difficulty to provide sensitive information to be available.

The next scenario taken is, "user's voice recognition" for performance quality attribute. This is a biometric feature that helps to provide authentication once the user speaks to open the lock or to make any kind of financial transaction. So rating given is (H, M) that is it is highly important to recognize the user's voice by the device when authenticated user speaks, and medium difficulty in performing the order.

The next quality attribute is Modifiability. The scenario taken is "should be open for changes". When the system is getting upgraded, it should be open for any kind of authenticated changes made. The rating given is (H, M).

The last quality attribute is Portability with scenario "easy adaptation in different environment". The rating given is (H, H). When system is getting moved in different environment, it is highly important to get adapted with the environment. At the same time it must be highly difficult to make adaptation of the system with unauthenticated user. The last step in this phase is to analyze the architectural decisions taken to identify their interactions with the quality attributes.

Phase 3: Testing

Step 7~ step 8: Based on scenarios taken in above phase, testing is carried out in two steps: brainstorming scenarios and analyzing architectural decisions. This step is explained with the help of bottom up approach and utility tree is explained with the help of top down approach [6]. The scenarios used are elicited in two ways as use case scenarios and change scenarios. Change scenarios elicited generally sub divided into two categories: growth and exploratory scenarios.These elicited scenarios, in this case study are based on exploratory scenarios. Then architectural decision is carried out similar to phase 2.

Phase 4: Reporting

Step 9: The last step is to summarize the outcomes in the form of a document which include analysis, architectural decisions, scenario refinements, priorities given to the refinements, identifying risks and non-risks, tradeoff and sensitivity points.

5. Conclusion

Software Architecture Analysis in recent years, is becoming a vital domain to make Qualitative and Quantitative analysis of the software. Mobile architecture becomes a part of everyone life. Representing and analyzing the architecture is one of the important step in the software development. In this paper proposed mobile architecture represented using MVC model and evaluated using ATAM. Main quality concern in this paper focused on is Security.Security quality attribute identified with the help of ATAM. ATAM list the quality attributes in the following categories that is Risk, Non-Risk, Tradeoff and sensitivity points. Finally based on the analysis the extracted scenarios are rated into High Medium and Low based on the important. Further to improve the extraction of quality attributes, utility tree can be generated and mapped with scenarios in future.

References

[1] Rick Kazman, MarkKlein, Paul Clements, ATAM: Method for Architecture Evaluation, August 2000, Carnegie Mellon, SEI.

[2] "Scenario-Based Software Architecture Evaluation Methods: An Overview ", Mugurel T. Ionita1, Dieter K. Hammer, HenkObbink

[3] "Software Quality: Definitions and Strategic Issues", Ronan Fitzpatrick Staffordshire University, School of Computing Report, April 1996.

[4] Novel Architecture for improving Security using LBS in Mobile Devices.

[5] "Architectural Analysis for Improving Security using LBS with ATAM" Almas Begum and V. Cyrilraj.

[6] Understanding Quality Attributes, from Understanding Quality Attributes, from Software Architecture in Practice by Felix Bachmann, Mark Klein.

[7] The essential components of software architecture design and analysis RickKazman, Len Bass, Mark Klein, 2006.

[8] Scenario based Software Architecture evaluation methods: An Overview, MugurelT. Ionita, Dieter K. Hammer, HenkObbink.

[9] Architectural Quality in Development Processes: A Case Study, Anna Grimán and Maria Pérez, Journal Of Object Technology, Vol. 2, No. 2, March-April 2003.

[10] Using ATAM to Evaluate a Game-based Architecture Ahmed Bin-Subaih1, Steve Maddock1.